INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Internship Program Management App with SAP Cloud Application Programming Model

NUNO MIGUEL CARVALHO DINIS
Junho de 2023

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Internship Program Management App with SAP Cloud Application Programming Model

Nuno Dinis

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Information and Knowledge Systems

ISEP Supervisor: Dr. Paulo Oliveira
KonkConsulting Supervisor: Tiago Mota

Evaluation Committee:
President:

Members:

Porto, June 30, 2023

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore the work presented in this document is original and authored by me, having not previously been used for any other end.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P.PORTO.

ISEP, Porto, June 30, 2023

Dedicatory

This dissertation is dedicated to my family, since they gave me the energy and will to conclude this work, specially to my grandfather.

Abstract

Internship programs are frequently used as part of recruiting on companies and provide a valuable onboarding options for organizations which have high demand of workers. For this, companies need to submit project proposals to entities which can provide the interns for the job. Examples of these entities are schools and universities.

Human Resource Management Systems provide great advantages for the management of Human Resource related data. Many times they offer a centralized solution which helps management saving time during the process of managing their employees, payroll and even their onboarding and recruiting.

Saving time on management by automating manual operations performed by the Human Resources personnel can benefit the companies on their budget. A real company which had a manual process of managing their internship programs proposals were on the origin of this work. Since they already have great part of their Human Resource management automated by an Human Resource Information System, thus, this dissertation is focused on the development of a new application for the submission and further management of internship proposals at the organization.

After a study of the existent solutions, it was concluded that the SAP Cloud Application Programming model would be used to develop an end-to-end cloud application for managing the internship proposals of the company.

An internship proposal application was developed using this same framework. It allows the centralization and automation of the process of submitting internship proposals and the maintenance of internship related data and it is integrated with the HRIS system of the company keeping the data about the enterprise resources up to date. It also allows the approval of projects and their management. The application also permits the creation, update and deletion of candidate data, cycles and institutions, including filtering capabilities.

The quality and efficiency of the implementation was evaluated by the final users, as well as the main use cases were tested by end-to-end testing, concluding a satisfactory implementation which met the requirements.

Keywords: SAP SuccessFactors, Human Resource Information System, Human Capital Management, SAP Cloud Integration Platform, SAP, SAP Cloud Application Programming Model

Resumo

Programas de estágio são frequentemente utilizados como forma de recrutamento nas empresas e fornecem importantes formas de *onboarding* para as organizações que têm uma alta demanda de trabalhadores. Para tomarem benefício dos programas de estágio, as empresas precisam de submeter propostas de estágio a entidades que podem fornecer estagiários. Exemplos destas entidades serão escolas e universidades.

Sistemas de gestão de recursos humanos oferecem vantagens para a gestão do capital humano da empresa bem como dos seus dados. Muitas vezes oferecem um forma centralizada de gerir, analisar e proceder com transações que ajudam a poupar tempo durante a gestão dos seus fucnionários, do seu pagamento e até dos ciclos de *onboarding* e recrutamento.

Poupar tempo na gestão de recrusos humanos com a automatização de processos manuais, pode ajudar as empresas a reduzirem gastos no seu orçamento. Uma empresa real que tem o seu processo manual de gestão de propostas de estágio está na génese deste trablho. Esta empresa já tem grande parte dos seus processos automatizados por uma solução de gestão de recursos humanos, um sistema de informação de recursos humanos, mais especificamente. Este trabalho vai apresentar a implementação de uma nova aplicação para a submissão e gestão de propostas de estágio dentro da organização.

Depois de um estudo inicial do estado da arte neste ramo, foi concluido utilizar a framework SAP Cloud Application Programming model para desenvolver uma aplicação final que visa automatizar a gestão de propostas de estágio na empresa.

Uma aplicação para a submissão e gestão de propostas de estágio foi desenvolvida utilizando a framework suprarreferida. Esta aplicação permite a centralização e automação do processo de submissão de propsotas de estágio, assim como a aprovação de propostas. Também permite a gestão de dados relacionados com as propostas de estágio, como candidatos, ciclos e instituições. Esta solução estará também integrada com o sistema de gestão de recursos humanos da empresa de forma a manter a correta atualização de dados relativos aos recursos da empresa.

A qualidade e eficiencia da implementação foi avaliada pelos utilizadores finais bem como a maior parte dos cenários de testes foram concluidos com sucesso, concluindo assim satisfatóriamente a dissertação.

Acknowledgement

I would like to start by thanking my family, specially my grandparents, my parents and my step-dad and step-mom which never let me down and helped me throughout this process.

To my friends, specially to Gabriel Azevedo, which stood up by my side during 7 years of studying, from the bachelor degree until the time of this dissertation.

My colleagues at the Tuna Académica do Instituto Superior de Engenharia do Porto by the good times provided which so much helped going through the writing of this work, thank you for the moments, for the music and for the trips.

To my co-workers which were always willing to help me at any time, including my supervisor, Tiago Mota.

To my supervisor at ISEP, Dr. Paulo Oliveira, which was always present to help and was indefatigable, always helping when needed and showing high interest on this work. Without any doubt the help given and the feedback obtained helped during the development of this work and helped developing a better version of this dissertation.

Contents

List of Figures

List of Tables

List of Source Code

List of Acronyms

API	Application Programming Interface
BTP	Business Technology Platform
CAP	Cloud Application Programming Model
CDS	Core Data Services
COTS	Commercial Off-The-Shelf
CPI	Cloud Platform Integration
CPU	Central Processing Unit
CRUD	Create, read, update and delete
CSV	Comma-separated values
DCL	Data Control Language
DSR	Design Science Research
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
ESS	Employee Self Service
FFE	Fuzzy Front End
FTP	File Transfer Protocol
FURPS	Functionality, Usability, Reliability, Performance, and Supportability
HANA	High Performance Analytics Appliance
HCM	Human Capital Management
HR	Human Resources
HRIS	Human Resource Information System
HRM	Human Resource Management
HRMS	Human Resource Management Systems
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
iPaas	Integration Platform as a Service
NCD	New Concept Development
ODATA	Open Data Protocol
QFC	Quality Function Deploymen
QL	Query Language
RAM	Random Access Memory
REST	Representational State Transfer
SFTP	SSH File Transfer Protocol
SOAP	Simple Object Access Protocol
UI	User Interface
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locators

Chapter 1

Introduction

The goal of this chapter is to provide a summary of the project that is covered in this master's dissertation on Informatics Engineering at the Polytechnic Superior Institute of Engineering in Porto.

What problem is solved throughout this dissertation, the definition of objectives, as well presenting the ways of achieving those objectives, are some topics that are clarified. Lastly, it covers the contents of the following chapters as well as the document structure.

1.1 Context

Human Resource Information Systems (HRIS) can serve as a vital important tool since they provide the administration with crucial information related to enrollment and maintenance systems that can be aligned with the general corporate technique for comprehending the hierarchical development goals. An organization can also determine the overall costs incurred per employee and their effects on the firm as a whole by using Human Resources (HR) tools [1].

The use of HRIS systems for administrative functions helps organizations save time and money while achieving their goals. HRIS systems have benefits such as the ability to produce reports that would often take several hours and the addition of analytical analysis for decision support [2].

Companies frequently use internship programs as part of their recruiting efforts since they assist on obtaining skilled talent and mold future employees, which is advantageous given the rising requirement for qualified and competent labor on their team. Apart from companies, both students and colleges themselves understand the significance of this [3]. Companies now find themselves developing internship programs to capitalize on the mentioned benefits.

Before leaving the company and being sent to academic institutions, company internship proposals typically go through an internal clearance procedure that includes a few steps, which include:

- The proposition of projects by the employees of the company, specially from the technical team.
- Review of the same projects and storage of the projects on a file.
- After approval, send the project proposals to the institution.
- If any candidate from that same institution is selected, onboard them into the company.

1.2 Problem Description

A real organization that operates in the Information Technology sector and currently uses a manual method for the submission and approval of internship proposals, which are many times done by the technical department of the company, served as the inspiration for this Master's Dissertation work.

The submission process is usually cyclical, since every year the company has new projects to submit for universities and other entities when the projects and internships phases are opened.

Currently, every year the HR department of the organization has to send an email to its teams in order to advise the submission of internship proposals. This can lead to delays on the submissions, if, per example, the responsible for the HR department is on leave and the warning of the new internship cycle being opened gets delayed.

This procedure is being carried out by email, which could result in a breakdown in communication and mistakes being made when creating internship bids as well as when approving them. There is no other procedure or automation for this rather than the email exchange, which frequently causes a delay in submitting these suggestions to the appropriate body.

In order to handle its HR data, the company also uses an HRIS solution, which results in highly automated reports and centralized data for the entire organization. The company currently does not manage their internship proposal procedure using the HRIS's capabilities.

This company has expertise in SAP systems, more focused on the Human Resource information, and makes use of many of the capabilities for its own internal use and management. In this specific case, the solution which the company has doesn't offer a proper solution for the management of internship proposals.

These systems can typically be expanded, but there are some restrictions. One of the challenges that corporations have is building bespoke extensions to these systems in order to have capabilities which are not configurable nor incorporated into the original system. This could result in the creation of an entirely new full stack application, which would be expensive, complex, and take a long time to construct. For this development, a large developer team might be required, with knowledge for security, database management, back-end development, front-end development and even architectural design, increasing significantly the costs of the bespoke application.

1.3 Objectives

Taking into account the factors listed on Section 1.1, this work will have the objective to propose some resolution for the problems faced. The list of final objectives according with the described problem is:

- Prepare a solution for a better centralization of information in regards to internship proposals, which is the primary goal of this solution.
- Include the automation of this process, providing a new method of submitting these artifacts which should include the integration with the primary HRIS system of the company.

- The solution should also allow the stakeholders of the internship program proposals to be aware of the submission process as well as the deadlines.
- The solution should have an user-friendly interface for end-users and should also include security features and authorization management for the data associated to internship proposals.
- Finally, the users should be able to access this data through the HRIS system and receive notifications from it, further automating the workflow and enabling easier reporting on related information.

Apart from designing and implementing a solution, this project's secondary objectives, which derive from the objectives listed, includes studying how HRIS systems can be expanded and evaluating and contrasting methods of integrating HRIS systems with external data and applications. These objectives also include reviewing methods of developing HRIS integrated applications and study existent solutions for the internship project portfolio management. With this, there is also the objective of investigating the application development paradigm for HRIS systems and how can their capabilities be used in order to downsize the development costs of developing such applications and extensions.

1.4 Methodology

The methodology of this work consisted on researching the relevant articles for the topics which were addressed. Initially, the main purpose was to better understand the current state about Human Resource Management Systems (HRMS). It also included understanding and researching the most recent technologies which could help on the development and resolution of issues related with the problems described on section 1.3.

Design Science Research (DSR) was used along this thesis. DSR is a paradigm for problem-solving that aims to advance human knowledge by producing original artifacts. Simply said, DSR aims to advance the knowledge bases of technology and science by developing novel artifacts that address issues and better the environments in which they are implemented [4].

There are 6 main activities which are part of the DSR [4]:

- Problem Identification/Motivation: This task clarifies the unique research issue and supports the necessity of a remedy. In addition to inspiring the researcher and the research's audience to pursue the solution, justifying the worth of a solution also enables the audience to appreciate the researcher's comprehension of the issue at hand. Knowledge of the situation and the significance of the issue's resolution are resources needed for this task. In this particular case, the problem identification can be found on the Introduction chapter as well as on the Value Analysis section, on Chapter 3.
- Define the solution objectives: The definition of the problem and knowledge of what is realistic and possible can be used to infer the goals of a solution. The objectives can be qualitative, such as a description of how a new artifact is intended to enable answers to problems not previously addressed, or quantitative, such as terms in which a desirable solution would be better than present ones. The goals should be logically deduced from the problem description. During this work, the objectives of the overall research are stated during the section 1.3. During the Chapter 3, Value Analysis, and

Chapter 4,Solution Analysis, these kinds of objectives are addressed and turned into value on the project.

- The artifact is designed and developed: A DSR artifact can theoretically be any created thing that incorporates a research input into the design. In this process, the required functionality and architecture of the artifact are determined before the actual artifact is built. These activities are described during the Chapter 4 Analysis, Chapter 5 Design and Chapter 6 Implementation.
- Demonstration: This exercise shows how to apply the artifact to an issue in one or more instances. This could entail using it for an experiment, simulation, case study, demonstration, or other applicable task. This activity is actively demonstrated during Chapter 6 Implementation and 7 Evaluation and result analysis.
- Evaluation: The evaluation gauges how well the artifact contributes to a fix for the issue. Comparing the goals of a solution to the actual outcomes that were seen after using the artifact in context is what this activity entails. This is stated during Chapter 7 - Evaluation and result analysis.
- Communication: Here, all pertinent stakeholders are informed of the problem's details as well as the intended artifact. Depending on the audience, such as working professionals, and the research goals, the appropriate means of communication are used. This will be done alongside Chapter 7 when the users test the system in order to provide feedback.

1.5 Document Structure

This report contains overall 8 chapters, including the present introduction, which gives an overall context and explains the primary problem which originated this work and its final objectives. As a second chapter, there is the State of the Art which includes detailed context of the project and an overview of the main concepts which support this dissertation, specifying the main existent solutions of extending HRIS and what kind of tools are used to do so. This analysis also includes a comparison between the ways of implementing such extensions and ways of integrating HRIS system's data. Finally it presents already existent solutions on the market.

Following this, the third chapter, Value Analysis, details what is the added value of this project when executed.

On a fourth chapter is presented the analysis of the solution, this means that the main business description and functionalities are described during this section.

Chapter 5 reflects the analysis made on previous chapters and provides the design for the solution. This design includes

Implementation is the sixth chapter, which illustrates the implementation and explains how the development was done. It is also in this section that some of the details of the development logic are considered and compared on how they impact the final solutions. During this chapter an analysis of the issues faced during the implementation is also done, as well as presenting the steps to overcome those problems.

The seventh chapter contains the evaluation of the designed and implemented solution as well as the detail of testing and acceptance testing by the end user.

The last chapter is the conclusion on which the final remarks of the project are made and where it is mentioned the future work for the project.

Chapter 2

Context and State of the Art

This chapter will provide an overview of the business field under study and provide clarifications for the reader's understanding of the fundamental ideas surrounding human resource information systems. This State-of-the-Art chapter also offers a thorough examination of the most recent trends, advancements, and best practices in Human Resource Information Systems (HRIS) and Human Resource Management Systems (HRMS), as well as how to take advantage of them to obtain a competitive edge in the market. It will also looks at the difficulties that businesses encounter when putting these technologies into use and provides workable ideas to aid them. In order to choose the best tools to development an extension to HRIS systems in the given scenario, it will be presented, compared and evaluated the most adequate technologies to use for this development.

2.1 Human Resource Information - Business Context

Human resources (HR) information refers to the data and records that a company keeps on their employees. This can include personal information such as name, address, and contact information, as well as employment information such as job title, department, and salary. HR information can also include information related to benefits, time off, and performance evaluations. It is important for companies to keep accurate and up-to-date HR information in order to effectively manage their workforce and comply with legal requirements [5]. This section will board some of the main concepts regarding HR information at organizations and its importance.

2.1.1 Personal Data

This type of information is related directly to the employees existent on the HRIS system. Information like name, address, dependent information and contacts are examples of personal data which are usually maintained as part of Human Resource Information for Personal Data.

Some of the data present in this section is related with Organizational Management data, acting like a bridge on the data representation between the employee and the organization. Section 2.1.2 explains on a more detailed how the information of Organizational Management is usually stored in solutions brought by SAP [6] which is one of the world leaders for software like enterprise resource planning (ERP) systems and other types of management systems.

2.1.2 Organizational Management

Organizational Management corresponds to information relative to the organization itself. Departments, stores, positions and jobs are examples of objects which are relative to this type

of data. The connection of the employee data and organizational information is needed in order to represent structures inside the company. In figure 2.1, we can see one representation of the object oriented structure of Organizational Management data as per SAP definition.

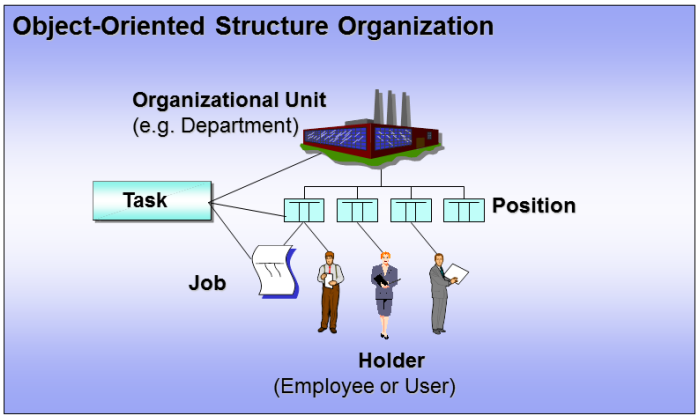


Figure 2.1: Organization management structure as represented by SAP [6]

2.1.3 Talent management

In Human Resource Management (HRM), talent management refers to a set of activities aimed at retaining and attracting human resources to a business or organization. From the perspective of HRIS, the automation of talent management activities can include [7] [8]:

1. Recruiting and hiring - Staffing is a key component of HRM. Staffing is significant because it provides a supply of personnel needed to complete roles inside a company that are required to fulfill business objectives. Per example, after HR professionals have completed new job analysis, a job description can be created. This type of job description is utilized while recruiting new employees. E-recruiting, often known as Internet recruiting, is one of the approaches available to HR professionals that can be connected with HRIS.

2. Performance management - Such web accessible applications play a significant role in reducing the effort required to manage performance evaluation. Staff members' performance is usually reviewed on a regular basis. Supervisors, colleagues, clients, and others who are involved have immediate access to performance reviews and relevant data relative to the employee.

3. Training and development - It enables HR to track employees' education, qualifications, and skills, as well as outline what training courses, books, Web-based learning, or resources are available to acquire specific talents. Courses can therefore be delivered in timed sessions, with delegates and training resources mapped and managed inside the same system.

2.1.4 Benefits administration

Benefits can be represented by general company policies for employees' overall benefits, like Health Insurance, Allowances, etc. Management of general staff benefit policies in large organizations necessitates a massive amount of written information, which can be performed more efficiently if HRIS are used. The system can track benefit eligibility dates, trigger reports to remind HR to notify employees, allow benefit choices to be quickly inputted, and deductions can be triggered on the payroll side of things, all of which reduces communication and paper flow between HR and payroll staff, as well as the likelihood of errors occurring at any point in the process [8].

2.1.5 Payroll and time tracking

The payroll module automates the pay process by collecting data on employee time and attendance, calculating various deductions and taxes, and generating pay cheques and employee tax reports on a regular basis. This module can house the entire staff-related business, as well as connect with the finance administrative units established before a company adopted HRIS. Traditional payroll administration was a tedious and time-consuming task that was prone to error due to the numerous details required, such as the original wage minus or plus different payments [8]. An HRIS can help to streamline this process; normally, the payroll staff only must provide the hours worked (or potentially not even that if the corporation uses an automated time clock incorporated with the HRIS), and the system will do all of the calculations for the company using a series of steps and procedures. Pay cheques are then quickly printed and distributed [8].

2.2 HRIS Systems - Features

The earliest employee information systems first debuted in the 1940s and were powered by sorting and tabulating machines into the 1950s. The computer system back then was only used for fairly basic tasks like keeping track of employee records and payroll procedures. 40% of all Fortune 500 companies had such systems in place by the 1970s. The majority of businesses with more than a few thousand employees created some kind of personnel data system during the 1970s [9].

The information mentioned through Section 2.1 is used in HRIS context, in an automated way. There are many features and capabilities of HRIS systems which are going to be revised during this section, which is more focused on their features.

2.2.1 Employee self-service

Employee self-service (ESS) is an HRIS feature that allows employees to see and update their own HR-related information without having to go via a human resources specialist [10]. Staff may request attendance reports directly from the system rather than from HR departments. The tool enables administrators to endorse and request from their subordinates via the system without burdening the HR department. Many organizations have gone beyond the traditional functions and developed HRM information systems to support recruitment, selection, hiring, job placement, performance appraisals, employee benefit analysis, health, safety, and security, while others have integrated an outsourced applicant tracking system that includes a subset of the above.

A corporation can delegate responsibilities through employee communication in order to facilitate the completion of functions traditionally anticipated of HR through direct individual access to the Web. Allowing employees to enter data on time and work, check their pension condition, manage deposits, design their contributions, build a yearly performance strategy, and monitor staff information are additional acceptable possibilities for direct access. A Web portal allows for two-way communication, which helps to improve the interaction between individual employees and the larger corporation [8].

As a brief summary, the key features of ESS capabilities of HRIS systems are [10]:

1. Personal information, such as contact information, emergency contacts, and tax information, can be seen and updated.
2. Enrolling in or changing existing benefits
3. Requesting time off or keeping track of vacation time
4. Viewing and printing job descriptions and performance evaluations
5. Requesting training or opportunity for professional development
6. Searching for and applying for vacant openings within the firm

As its advantage is the time efficiency which is a result of allowing employees to access and update their own information without having to go via a third party. Furthermore, because employees are directly entering their own information, ESS can lessen the possibility of mistakes. Finally, because workers can access their own HR-related information and view the status of requests they've made, ESS can increase communication and openness.

2.2.2 Strategical decision making

Human resource strategic decision making has improved and capability has increased as a result of highly consolidated and organized data provided by HRIS systems.

Studies like [11] confirm on the existence of a practice and preference. For managers which have decision making tasks regarding their staff, HRIS tools makes their life easier [12].

HRIS data provides managers with a decision-making tool rather than just a robust database, which is due to its high standardization and being so highly configured. Even the most basic Commercial Off-The-Shelf (COTS) human resource software solution may give precise data in a range of areas. Firms are now able to do calculations that have an impact on the business as a whole thanks to proper HR data management. Health-care expenditures per employee, pay benefits as a percentage of operational expense, cost per hire, return on training, attrition rates and costs, time required to fill certain occupations, return on human capital invested, and human value added are some examples of such calculations [13].

2.3 Human Capital Management

There are many definitions given throughout the time for Human Capital Management (HCM).

Table 2.1 will go over some of the most common definitions given to this context:

Table 2.1: Examples of HCM definitions and correspondent authors [14]

Author	Year	HCM Definition
Finn [15]	2003	HCM is all about ensuring that the enormous potential provided by people is aligned with the mission and strategic objectives of the business, to maximise their value on behalf of the stakeholders
Hayton [16]	2003	The measurement and analysis of human resource 'metrics' such as cost per hire, turnover costs, the effectiveness of training interventions, and indicators of overall HRM system effectiveness such as human capital return on investment
Nalbantian [17]	2004	HCM involves putting into place the metrics to measure the value of HC attributes (accumulated knowledge, skills, experience, creativity and other relevant workforce attributes) and using that knowledge to effectively manage the organisation
Baron and Armstrong [18]	2007	HCM provides the basis for strategic HRM through measurement and analysis leading to evaluation, diagnosis and action
Lawler [19]	2009	Making human capital a source of competitive advantage requires much more than making some quick fixes to a control-focused organisation. It requires attracting and retaining the right people as well as organising and managing them effectively. . . . It requires the right managerial behaviours as well as the right design of most of an organisation's major operating systems in order to create a Human Capital centric (HC-centric) organisation

HCM is a people management strategy that focuses on procedures including hiring, performance management, remuneration, and benefits administration. To promote corporate success, HCM aims to acquire, nurture, and retain talent.

Managing benefits, keeping track of vacation time, and maintaining employee data are among the administrative duties of HR that are primarily the emphasis of HRIS. On the other hand, HCM is a broader phrase that includes not just the HRIS administrative tasks but also the strategic management of HR, including hiring, development, and performance management. HRIS is a subset of HCM, which has more restricted functionality and functions as a foundational tool for HR procedures. HCM offers a more complete solution for managing the whole employee lifecycle. It is safe to say that both therms are related.

2.4 Market Solutions

This section will include a quick description and introduction to the most popular HRIS systems on the market, as well as a final comparison of them.

The following systems will be considered for this analysis:

- SAP's Success Factors
- Oracle Cloud HCM
- Sage people

- Workday

This will concentrate on some comparison criteria. The following will be the criteria selected for this:

1. Cloud based or On Premise based
2. Integration capabilities
3. Existent modules

2.4.1 SAP's Success Factors

SAP SuccessFactors solutions are cloud-based HR and payroll software tools that also enable talent management, HR analytics and workforce planning, and employee experience management. SAP SuccessFactors products, which were launched in 2001 and have been part of the SAP implementations since 2012, are utilized by over 235 million users in over 200 countries and territories worldwide [20].

SuccessFactors is also a multi-tenant solution that offers a separate application instance for each customer, improving security and preventing customer data from mixing. The software is built on a hybrid database strategy that allows for configurability and scalability - at least two SuccessFactors customers have more than a million employees. While all customers use the same database schema, customer data is segmented at the database level, allowing configuration by redefining the relationship between different schema elements using a combination of XML objects and the Meta Data Framework [21].

For integration purposes, SAP delivers some possible solutions.

SuccessFactors' Integration Center

SAP SuccessFactors Integration Center is a built-in solution within SuccessFactors for developing, testing, and monitoring inbound and outbound integrations. Multiple output file types are available (from simple Comma-Separated values, or CSV, to XML or JSON). The output can be stored securely on SFTP servers, and other scheduling options can be applied (from fixed time scheduling to event-driven triggers through Intelligent Services). Integration Center also includes a collection of pre-packaged integrations that can be deployed and used in the customer instance [22].

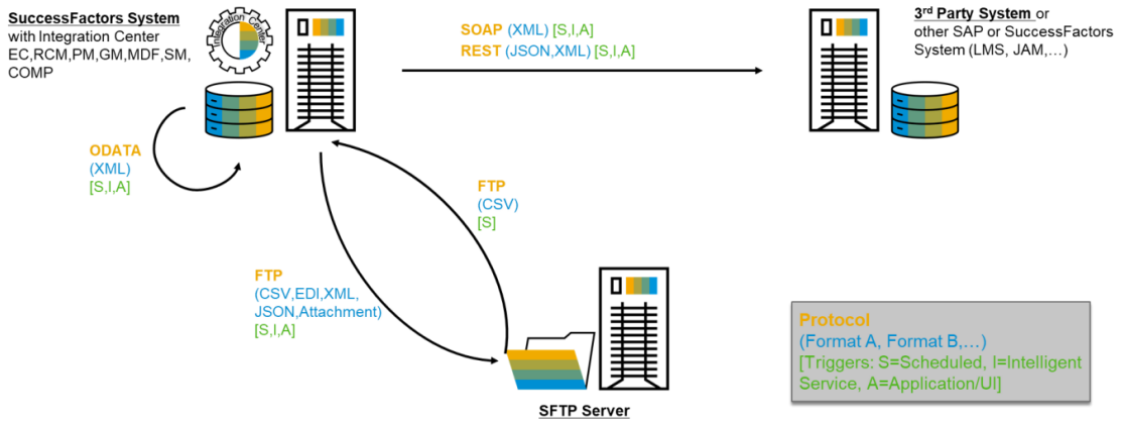


Figure 2.2: Integration types supported by the Integration Center [22].

On figure 2.2 the arrows indicate that Integration Types support only one source and one target system. The source system is at the beginning of the arrow, and the destination system is at the end. Open Data Protocol (ODATA), Simple Object Access Protocol (SOAP), Representational State Transfer (REST), and File Transfer Protocol (FTP) are supported protocols, and the formats for those protocols (CSV, XML, and JSON) are supported formats. The color green shows if the integration type can be scheduled (S), triggered by an application event (A), or intelligent service event (I)

SuccessFactors' APIs

SAP Success Factors offers a set of Application Programming Interfaces (APIs) in order to expose data contained in their systems.

There are 2 main web services which can be used:

- REST/OData APIs
- Compound Employee APIs which are SOAP based APIs

SuccessFactor's APIs with middleware - SAP Cloud Platform

Cloud Integration (CI) is a cloud middleware solution that allows integrations to be created, deployed, and monitored. These integrations can take place between SAP On-Premise, SAP Cloud, and third-party applications. It provides a wide range of connectivity, message transformation, and authentication options, as well as pre-packaged integrations to interface SAP solutions with other SAP solutions and third-party products [22].

The type of connectivity with SAP Success Factors from a Middleware standpoint can be made via OData and SOAP APIs.

SuccessFactors' SFTP server

For inbound and outbound integration SSH File Transfer Protocol (SFTP) servers can be used in Integration Center. This can either be a customer owned SFTP server or and SAP provided one. [22]

After this, the customer can get the data needed using any software of their choice and consume the files existent on the SFTP server.

2.4.2 Workday

The workday HCM module was created to provide management activities from the ground up and is built on the base of cloud services. Workday HCM Suite provides a unified solution for all HR tasks, including planning, human resource management, performance management, financial management, etc. It can be used by businesses of almost any size, especially those with many locations. It serves as a one-stop shop for analytics, payroll, vacation time, benefits, learning & development, onboarding, hiring, and salary benchmarking. Customers can access the most recent version with updated capabilities using a browser or a mobile device thanks to Workday HCM software, which is a very user-friendly, worldwide system application [23]. Workday HCM also has Employee self-service capabilities for both employees and managers [24].

Workday - APIs and Middleware

Workday offers a variety of connections and integration options, including direct use of Workday Web Services or REST-based APIs. Alternatively, you can use the Workday Integration Cloud, which offers a full set of integration toolsets including an Enterprise Service Bus (ESB) integrated into Workday Studio, an eclipse-based IDE for building custom connections, and related tools for managing and monitoring integrations. The Workday application surfaces all integrations, whether they are offered by Workday or one of authorized partners, and allows you to use them directly in your business process setups. This gives the flexibility needed for a dynamic HCM environment [25].

Cloud Connectors

Connectors are pre-built integrations that cover one to several use cases and are designed, documented, and supported by Workday. By providing the processing logic, data transformation and error handling needed to integrate a third-party system with Workday, they significantly minimize the deployment time necessary to finish an end-to-end integration. All connections operate in the Workday Cloud and can be customized to fit various client circumstances (such as customization of your third-party system), relieving the load of having to manage a separate integration architecture [26].

2.4.3 Oracle Cloud HCM

Oracle HCM is the Human Capital Management, cloud-based solution provided by Oracle. It offers a complete solution which has multiple modules [27]:

- Oracle ME - It is a platform oriented to employee experience that directs workers through their personal and professional activities, provides effective HR service support, and simplifies communication throughout the company. It's a more self service oriented module.
- Oracle HR - Provides tools for managing personnel, positions, and tasks effectively, including global assignments. These tools also make it easier to manage special industry, union, collective work, and employee agreements through the use of policy-driven procedures.
- Talent Management - Module focused around talent management oriented tasks. Recruiting, learning cpabilities (help giving recommendations for skills which employees would like and benefit from learning), carreer development (by providing employees ways to help on carrer mobility, thus increasing employee retention, by per example suggesting new jobs and positions within the same company), performance management, compensation management and sucession planning.
- WorkForce Management - Includes health and safety aspects promoting workplace safety and employee overall health, time recording and labour management, absence management.
- Oracle payroll - Automatically process payroll on many locations supported. This can many times reduce the implementation of third party solutions which run payroll separately, reducing also integration necessities.
- Oracle HCM analytics - Allows running analytical reports supporting easier decision making by management and promoting a better workforce leverage.

HCM extracts

An HCM extract is a tool for reporting, archiving, and outward interface development. Customers can create custom defined data extracts using HCM Extract to export business data kept in cloud apps. It has a dedicated interface for specyfing the records and data to be extracted [28]. With HCM extract it is possible to:

- Use complex extracting and selection criteria to extract the needed recrods.
- Using quick formula database items and rules, define the data elements in an HCM extract.

Web-services

In order to integrate web applications with HCM Cloud there is the possibility to use some web based services. Web services use open standards-based technologies to expose business objects and processes to other applications [29]. Examples of this is SOAP Web Services for Oracle HCM Cloud and REST API provided by Oracle Fusion HCM.

2.4.4 SAGE People

Sage People is a cloud-based HR and people management system for mid-sized international enterprises. Built on Salesforce, the top enterprise platform in the world, it automates hiring, management, and professional development of the workforce while supporting new working styles, relieving HR of manual administration and low-value duties. It is a single system of record for HR and employee data, and it includes extensive reporting, dashboards, and analytics that offer useful information for guiding strategy, plans, and operations [30].

It offers capabilities on areas like talent acquisition, performance and talent, reporting and analytic features, benefits and well-being, reward planning, leave and attendance reporting, and experience and communications [30].

SAGE People APIs

Sage people also offers web services and APIs for data transfer and extraction. It's possible to use some entities like *employees* and *absences* which retrieves data for employees and time off respectively [31].

2.4.5 Overall Comparison

Using a simple therm of comparison, the following table 2.2 will provide some briefing of the information about the Human Resource systems that were previously analyzed. On the column "Modules", it will be stated how many modules are covered from the following:

- On-boarding and Recruiting
- Talent Management (Performance management, training, etc)
- Benefits
- Payroll
- Time and Absence recording

For the ESS column, it's taken on consideration the capabilities of Employee Self-Service. On the other columns, it is self explanatory, the existence of APIs for data integration and the possibility of extracting mass data or usage of reporting to extract data are respectively taken into account.

Table 2.2: Human Capital Management systems comparison

System	ESS	Modules	API	Data extraction/Reporting
SAP Success Factors	✓	5 out 5	✓	✓
Workday HCM	✓	5 out 5	✓	✓
Oracle HCM	✓	5 out 5	✓	✓
SAGE People	✓	5 out 5	✓	✓

Concluding the analysis on the above systems, it is possible to notice that all the systems offer similar capabilities across all the topics considered. Many times the preference of clients takes into consideration the market value of the solutions and the pricing, which is often difficult to know without having a proper budget estimate from the developing company. Other important aspect for companies selecting an HRIS/HCM system is the existence of other systems at the time they want to adopt a new HRIS system. Per example, one company which already uses ERP - HR solutions from SAP, per example the S/4HANA ERP system, may prefer to use SAP SuccessFactors once it may offer pre-packaged integration and migration tools as well as it might be important to maintain the same support team.

After this analysis, since the real company which is the primary object of study of this dissertation already has a SAP SuccessFactors solution in use, that will be the system used as base for HR data in this project.

2.5 Application Programming Interfaces

A set of established guidelines known as an API, or application programming interface, allows various applications to communicate with one another. Companies are able to open their application data and functionality to external third-party developers, business partners, and internal departments within their companies thanks to this intermediary layer, which handles data transfers between systems [32].

Connectivity interfaces to an application have been collectively referred to as "APIs" in the past. But as time went on, the modern API developed some distinctive qualities that have genuinely changed the technological landscape. First off, modern APIs follow particular standards (typically Hypertext Transfer Protocol or HTTP and REST), which make them developer-friendly, self-descriptive, simple to use, and widely understood [33].

APIs act like a bridge between applications. It's comon that in some architectures, separated concerns are written in different languages, in these cases, APIs work as translators, enabling the communication between components which are developed at different ways, by different people on different programming languages.

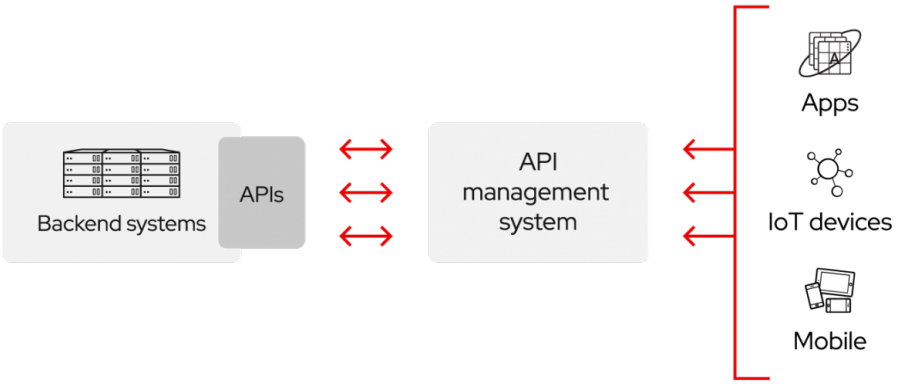


Figure 2.3: API representation by Red Hat [34]

By analyzing Figure 2.3 it's possible to get a better representation of the API purpose. On this context, APIs are important to transfer data relative to the HRIS systems in scope for an extension.

As mentioned on Section 2.4.1, SuccessFactors presents some alternatives for exposing data via API, being those alternatives REST and OData based as well as SOAP based.

2.5.1 Representational State Transfer

Representational State Transfer is an architectural style for the distribution of hypermedia systems [35].

The concept was initially originated by Roy Fielding at [35]. It presents a few constraints to follow this architecture at its principle:

- Client-Server: It is a common architectural style for network-based applications. This design pattern depicts distributed systems with several clients that communicate with the server components to request services. Clients start REST requests, which are then processed by an origin server, which further provides services. REST places a lot of emphasis on separating the client and server, but in some architectures it might be difficult to decide which modules will function as clients or servers. The key advantages of the client-server architecture are maintainability, independent evolution, and responsibility separation. According to coherence and the independent evolution of each, evaluators should scrutinize the specification of the client-server boundary [36].
- Stateless: This means that each request sent by the client to the server must be complete and cannot draw on any context that has been previously kept on the server. As a result, the client alone maintains the state of the session. Visibility, dependability, and scalability are caused by this constraint. This brings a drawback which is the possibility of reducing network performance, since the requests will need to be more repetitive [35].
- Cache: To improve network efficiency, cache limitations are taken into account. Data in a response to a request must be explicitly or implicitly marked as cacheable or non-cacheable in order to comply with cache limitations. If a response is cacheable, a client cache may use the response data for subsequent, identical queries. The benefit of imposing cache limits is that they may be able to remove some transactions entirely

or partially, reducing the average latency of a sequence of contacts and enhancing efficiency, scalability, and user perception of performance [35].

- Uniform Interface: The emphasis on a consistent interface between components in the REST architectural style sets it apart from other network-based designs. Multiple architectural constraints are required to direct the behavior of components in order to achieve a unified interface. The four interface restrictions that make up REST are: resource identification; resource manipulation using representations; self-descriptive messages; and hypermedia as the vehicle for application state [35]. Any significant idea in the system domain that we want to make accessible via a standardized interface is referred to as a resource. Through a representation and a uniform resource identifier (URI), a service consumer addresses resources. A resource's representation is a hypermedia document that includes links to related resources and any relevant information about the resource's status. Finally, a straightforward interface that specifies a resource's identification and common access methods must be used to access the representation of a resource. The uniform interface is this one. Interoperability and discoverability quality attributes are positively impacted by uniform interface constraints [36].
- Layered System: A layered system is set up hierarchically, with each layer acting as both a client and a service provider to the layers above and below it. With the exception of a few functions that have been specifically chosen for export, some layered systems' inner layers are hidden from everyone but the outer layer next to them. As a result, at some level of the hierarchy in these systems, the components implement a virtual machine [37]. The layered-client-server style is enhanced by the addition of a proxy and gateway [35].
- A client component using the code-on-demand model [38] has access to a set of resources but lacks the knowledge to use them. It requests the code that embodies that knowledge from a distant server, obtains that code, and executes it locally [35].

There are a few concepts which are also important to the REST architecture which are going be explained in further detail on this section.

Resources and Uniform Resource Identifier

At RFC 2396 [39], there is a brief overview of the three main concepts and definitions around the URI:

- Uniform - There are several benefits of uniformity, from allowing different types of resource identifiers to be used in the same context, even though that the mechanisms to access them may differ. It also allows uniform semantic interpretation of common syntactic conventions among various types of resource identifiers. It permits the creation of new types of resource identifiers without any interference with the existent ones. Finally it also gives the possibility to re-use already existent identifiers in different contexts and applications.
- Resource - A resource can be anything that has identity. There are many examples like electronic documents, or in the specific case of this dissertation, it can be a representation of an employee through data. The resource is the conceptual mapping to an entity or set of entities, not always the entity that at any given time corresponds to that mapping. So, even if the entities to which a resource now corresponds change

over time, the resource can still exist as-is as long as the conceptual mapping does not change.

- Identifier - An identifier is an item that can be used to refer to something that is uniquely identifiable. The object in the case of URI is a string of characters with a specific syntax.

An exmaple of an URI on the API from Success Factors is:

`https://<API-Server>/odata/v2/EmpInfo`

In the above example, the EmpInfo, represents the Employment Information entity, which is the representation of the information related with the Employment of a given Employee, per example the Hire Date.

Hypertext Transfer Protocol

HTTP is the protocol used on a REST architecture. There are a few methods which are synthesized on table 2.3 which are important to understand:

Table 2.3: HTTP Methods [40]

Method	Description
Get	Retrieves the representation of a resource.
Head	Retrieves metadata for the representation and resource.
POST	In the strict sense, POST creates a resource. In the real world,however, POST is typically used to create, update, and even delete a resource. It is normal to use REST services that support only GET and POST.
PUT	Updates a resource. More often than not, you will not see this method used in the real world but instead will see POST used to perform the actions.
DELETE	Deletes a resource.

Each of these methods offers particular functionalities that are frequently explained using the database acronym CRUD. Create, Read, Update, and Delete are referred to as CRUD [40].

2.5.2 Open Data Protocol

OData makes it possible to build REST-based data services, which let Web clients publish and modify resources that are identifiable by Uniform Resource Locators (URLs) and described in a data model using straightforward HTTP messages [41].

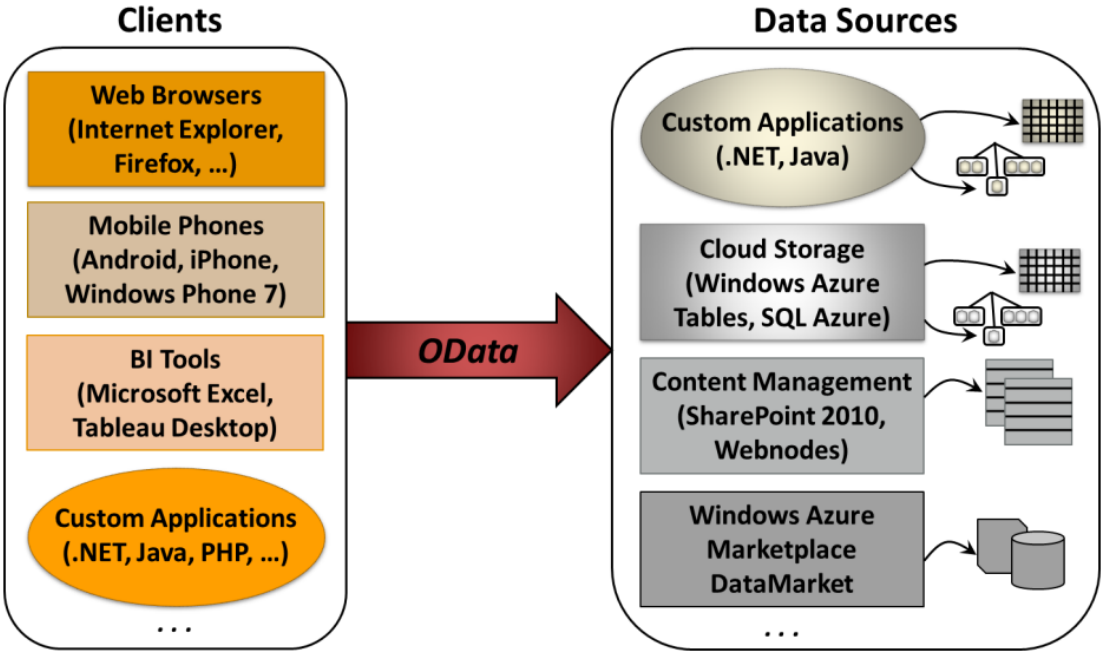


Figure 2.4: Any OData client is capable of accessing data provided by an OData source [42]

Figure 2.4 represents examples of OData clients and sources which are capable of communicating using the protocol. It is stated that any OData client can access the data provided by an OData source [42].

OData services are capable to provide the description of data models through the metadata which is exposed. OData also allows complex data querying capabilities, through applying filters on the URI during a client-server communication. These OData queryable options are introduced by an $ symbol, as listed below:

- $filter: this query operator is used to retrieve data filtered by one or more conditions. You can use it, per example, to filter the Employees from a data set which belong to Country Portugal or Spain.
- $top: it's used to bring the top N results of a given query.
- $skip: this operator allows to skip a certain number of results on a query. Per example, it's possible to Skip the first 10 Employees from a result set.
- $select: as a database select statement, it allows to specify which properties of a certain entity should be retrieved. This can be used for getting the First and Last names of an employee, per example, instead of bringing all the information.
- $expand: allows to fetch multiple related entities in a single request using the relationships between entities. Per example, if the employee has an Address Information, which is a different entity from the employee itself, we can use an $expand query to also retrieve the Address information from the employee on a single query.
- $format: formats the data according to the type specified, we can use JSON or XML for data querying.

Another capability of these types of APIs is the operations, which allow custom logic to be invoked.

2.5.3 Simple Object Access Protocol

SOAP is a simple protocol which is used to communicate data in a distributed, decentralized setting. It is a three-part XML-based protocol that includes an envelope that establishes a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for expressing remote procedure calls and responses. SOAP can theoretically be used in combination with a variety of different protocols, although the most common are HTTP based [43].

There are four fundamental components on a SOAP message [44]:

- Envelope: The SOAP message format's envelope is a deciding factor in identifying a given document as SOAP and defining the beginning and end of the message format. The SOAP message format requires it.
- Header: This element includes optional data, like authorization and verification information.
- Body: It is a required element that includes XML information. It has data that is sent to the program as well as request and response information.
- Fault: This element communicates information regarding faults that occurred during message transmission, such as version mismatch, client fault when the message was created incorrectly, and server fault when the server experienced a difficulty. Errors and status data are also included.

The figure 2.5 contains the representation of the typical SOAP message:

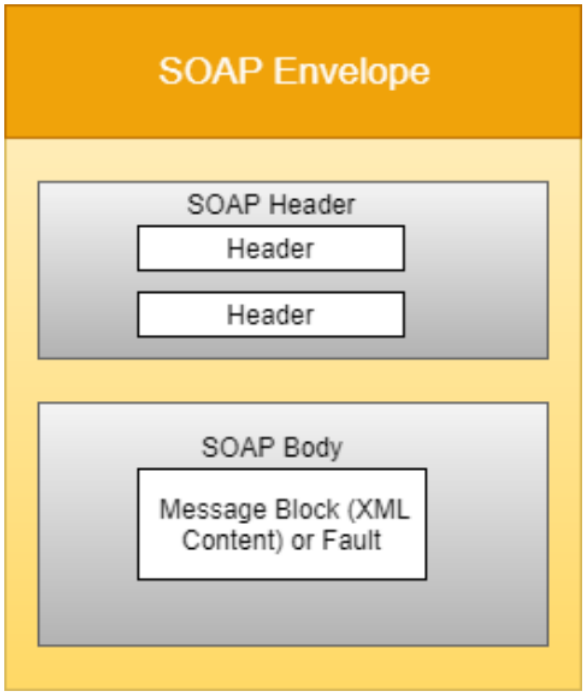


Figure 2.5: API SOAP message representation [44]

2.6 Middlewares

An overview of integration platforms and a quick comparison of the ones that are currently available on the market are included in this section.

2.6.1 Integration Platform as a Service (iPaas)

Integration platform as a service is a collection of cloud services designed to solve a variety of cloud, Business-to-Business, and On-premise integration and management scenarios. It is a set of cloud services that enables the creation, execution, and management of integration flows that connect any combination of on-premises and cloud-based processes, services, applications, and data inside a single business or across many companies [45]. Cloud services for protocol bridging, message transports, transformation, routing, service virtualization, adapters, orchestration, partner community management, MFT, registry/repository, development tools, and others are typically combined in iPaaS packages. It consists of a collection of cloud-based services that provide a multi-tenant and scalable platform to handle a wide range of integration scenarios [45].

2.6.2 SAP Cloud Platform Integration (SAP CPI)

SAP Cloud Platform Integration is an iPaaS solution, which helps connecting SAP, non-SAP, cloud or on-premise solutions through its integration capabilities [46].

SAP's integration solution presents the following integration content and capabilities:

- Predefined content - SAP provides out of the box integrations which contains many pre-built integration flows and elements, which can be either re-used or even only configured [46]. One of the examples is a pre-made integration between SuccessFactors and SAP ERP. These integrations already have configured artifacts like data mappings and value mappings.
- Develop and edit integration content - SAP CPI also allows the creation of fully custom developed integration flows. It's possible to use these integration flows to configure messages' senders and receivers for integration scenarios. It also allows to manipulate these messages during the execution. In this way, the integration flows allow the connectivity of the data sources to the CPI tenant and allows the gathering of integration steps in order to process the data.
- Transport integration content - It is possible to move these integration flows between tenants using Change and Transport Systems, SAP Cloud Transport management or even a manual export/import via files.

For connectivity purposes, SAP CPI has a great variety of connectors which are pre-packaged, apart from the overall API, file system and databases connectivity. There are connectors and adapters like Dropbox and Facebook which allows CPI to connect to the respective systems.

There are also many message transformation capabilities, from Data Mapping capabilities, which allow the data transformation to different formats according to the data source and target. There are also script capabilities for more complex transformations as well as encryption and decryption mechanisms. It also allows the storage of data variables for future use and it also gives access to CPI's data stores.

2.6.3 Mulesoft - Anypoint Platform

Mulesoft's solution called Anypoint Platform is focused on API and integration capabilities.

The Mule Enterprise Service Bus is a middleware solution that links the enterprise fast, easily, and securely. Mule as an ESB is a Java-based middleware solution that is simple to use and simple to scale, in contrast to standard middleware technologies [47].

Mulesoft Anypoint Platform contains the following capabilities:

- Anypoint Flex Gateway - made to manage and secure anywhere-running APIs. Anypoint Flex Gateway provides the performance needed for the most demanding applications while delivering enterprise security and manageability across any environment. It was designed to connect effortlessly with DevOps and CI/CD workflows [48].
- Anypoint API Governance - While developers want to minimize the complexity of conformance review cycles, architects want to guarantee uniform quality and security while dealing with a variety of APIs across numerous teams and design tools. With self-service tools, Anypoint API Governance guarantees constant API quality and security [49].
- Anypoint API Manager - Gain crucial insights into API programs, manage client access, secure APIs with policies, and group APIs as products with just a few easy steps. No matter where they are located, manage all the APIs and microservices from a single location [50].

From an integration perspective there are other relevant capabilities in Anypoint Platform [51]:

- Anypoint API Designer - A visual or coding-based guided experience is offered by API Designer for designing, describing, and testing APIs in any language. Use Anypoint Exchange, a library of APIs, templates, examples, and connectors, along with a single-click mimicking service, to quickly involve API customers at various phases of the design process.
- Anypoint Studio - Querying backend systems, routing events, business transformation logic, and error handling are just a few of the frequent integration requirements that have prebuilt modules in an IDE for integration and API development.
- Anypoint Connectors - Utilize a desktop IDE or a web interface to integrate systems. Utilize pre-built connectors or one using SDK. Visual error management helps designers find and correct mistakes.
- Anypoint Data Graph - Combines several API requests using Anypoint DataGraph. Without adding new code, enterprise architects may quickly combine many APIs into a single data service. In a single GraphQL request, developers can use numerous APIs from the data service.

2.6.4 Selected Solution

Due to its pre-built SAP connectors, as well as the SAP already pre-packaged integration and taking advantage of the experience covered on SAP CPI by the company, as well as the experience with all the SAP ecosystem, SAP CPI is going to be used.

2.7 HCM and HRIS extensions for project portfolio management

This section will give an overview of the tools available which may fit for the requirements needed. It will also approach possible ways of developing extensions for HRIS and HCM systems in order to fulfill the requirements. It will also take advantage of previous mentioned points like REST APIs and how it's possible to use them to extend functionalities of HRIS systems or even have custom integrated solutions.

2.7.1 SAP Enterprise Portfolio and Project Management

This section will give a brief overview of the possible solution found on the SAP marked for the project management solutions.

SAP Enterprise Portfolio and Project management is the SAP solution for managing data regarding projects, their management and portfolio. It allows the following points [52]:

- Enable help on project management - from proposal to development and closure.
- Help analyzing, during the initial phase, possible project anomalies and risks.
- Reporting costs and accounting data.
- Integrate with enterprise information.

This makes cross-functional, creative business processes possible [52]. In order to provide information of the projects and the project portfolio with drill-down and details portfolio management integrates data from project management, HR and Financial systems [53].

The main feature which is related with this specific dissertation is the portfolio management which allows submitting project proposals, prioritize them with current projects, and monitor and review their progress.

As per SAP the costing of the platform starts on 136 US Dollars per person/user per month (professional version) [54].

Although this is a specialized solution from SAP perspective it is more focused on project management and its adjacent tasks, thus providing more complexity than what is really needed for this project. Its cost is also being considered, since all the employees of the organization would need the access to the application, this can translate in higher costs for the company for what it needs.

2.7.2 SAP Cloud Application Programming Model

Creating a custom application can also be considered for the use case needed. During this section, a framework provided by SAP to develop end-to-end applications will be covered.

A framework of languages, libraries, and tools called the SAP Cloud Application Programming Model (CAP) is used to create enterprise-level services and applications. It directs programmers through a "golden road" of tried-and-true best practices and a variety of original answers to frequent problems [55]. It allows building conceptual data models and service descriptions using Core Data Services (CDS). The data, service, and User Interface (UI) layers all use these CDS models as inputs. Following translation into native artifacts, such as SQL database schemes, they are then evaluated to swiftly fulfill run time demands. In

conclusion, CDS is utilized to create the artifacts at the persistence layer and as a source for business level data description. It is used to specify the data's visual elements, and those specifications (annotations) specify the UI layer. Additionally, the application service layer is then generated using it [55].

In-memory and Column based databases

For a better understanding of the types of databases which can be used on SAP CAP there are two therms which are meaningful to understand, which is In-Memory computing and databases and Column Based Databases.

In-memory computing is a therm which is described by the use of Random Access Memory (RAM) as a means of storage, instead of its traditional use of cache of memory. In this way, data retrieval and processing performance is close to the Central Processing Unit (CPU) [56].

From a database perspective, there are two main ways of storing the data, either by the use of row storage or by the use of columnar storage. As it's presented on Figure 2.6, column on both row storage and column storage.

When using column storage, a query must visit all blocks in order to select all attributes of a record (a row operation); whereas, a query only has to visit one block to select all values of an attribute (a column operation). On the other hand, with row storage, a query only needs to visit one block to select all attributes of a record (a row operation); otherwise, it must visit all blocks to get all values of an attribute [57].

On figure 2.6 it's possible to analyze how the different types of storage behave at a physical level, which is directly going to influence the querying capabilities, depending on the type of query used.

Case ID	Amount	Agent
0098878130	65,732.76	Daniel
0102522689	7,846.98	Vincent
0112883423	874,987.87	Tiffany

Row-oriented Storage

Block 1	0098878130
	65,732.76
	Daniel
Block 2	0102522689
	7,846.98
	Vincent
Block 3	0112883423
	874,987.87
	Tiffany

Columnar Storage

Block 1	0098878130
	0102522689
	0112883423
Block 2	65,732.76
	7,846.98
	874,987.87
Block 3	Daniel
	Vincent
	Tiffany

Figure 2.6: Column Storage vs Row storage database [57]

The ability to eliminate the requirement for the index, which is prevalent in the world of disk-based databases, is made possible by the usage of row and columnar data storage. One important area for compression is the absence of the index, and columnar storage enables even more compression [56].

SAP High Performance Analytics Appliance (HANA)

SAP HANA (High-performance Analytic Appliance) is a multi-model database keeps data in memory rather than on a disk. Execute powerful analytics and fast transactions on the same system thanks to the column-oriented in-memory database design are some of the perks which SAP HANA has taking advantage of the topics mentioned on Section 2.7.2, [58].

SAP Fiori

Based on the HTML5 foundation, Fiori was created with a focus on user-centric apps and a common UI paradigm. This made it easier for SAP to create a responsive User Interface (UI) that works on any type of device with an HTML5-compliant web browser and is also compatible with all screen sizes [59].

Together with having applications for individual duties, the customer has access to a line of apps based on the duties that specific user handles through Fiori. Consider an employee who manages multiple tasks in a business: the user is a staff member but also has managerial permissions to handle financial matters. As a manager, the user needs access to specific approval-related apps; as a staff member, the user needs to request time off, update timesheets, submit claims for travel expenses, and so forth. The Fiori launchpad serves as the end user's single point of access to all apps relevant to their roles in Fiori [59].

On the particular case of this dissertation, a Fiori user interface would provide the user capabilities of submitting, approve and manage internship projects as well as providing an intuitive interface experience.

Core Data Services

Predictive analytics [1] and text mining [2] are all made possible by the availability of multi-model databases that support complicated data structures like spatial, text, graph, and time series data. These capabilities serve a variety of application patterns, from decision and analytical support systems to transactional systems, in real-time and on live data [62].

Moving data, however, becomes the bottleneck as data size increases, and the expense of doing so becomes unaffordable. Nevertheless, the data can frequently be analyzed and processed without using conventional application layer, processing it directly on the data store [63].

[1] A subset of advanced analytics called predictive analytics uses historical data along with statistical modeling, data mining, and machine learning to forecast future results. Utilizing trends in this data, businesses use predictive analytics to spot dangers and opportunities. Big data and data science are frequently linked with predictive analytics [60].

[2] The process of converting unstructured text into a structured format with the purpose of identifying significant patterns and fresh insights is known as text mining, also known as text data mining. Companies are able to investigate and find hidden links within their unstructured data by utilizing advanced analytical approaches like Naive Bayes, Support Vector Machines (SVM), and other deep learning algorithms[61].

Having as example a table which has millions of values representing different currency exchange rates which need to be calculated and processed. On a normal, row storage application, the ideal was to pull the data from the database onto an application server in order to process it. Then, the residual data would be disposed after the calculations are done. In this type of cases, it would be advantageous to use the data processing logic on data storage so that there is no unnecessary movement of such a big amount of data. This type of processing is called code push down or code-to-Data, which improves the performance of some applications by implementing some logic at the database level [63] [64].

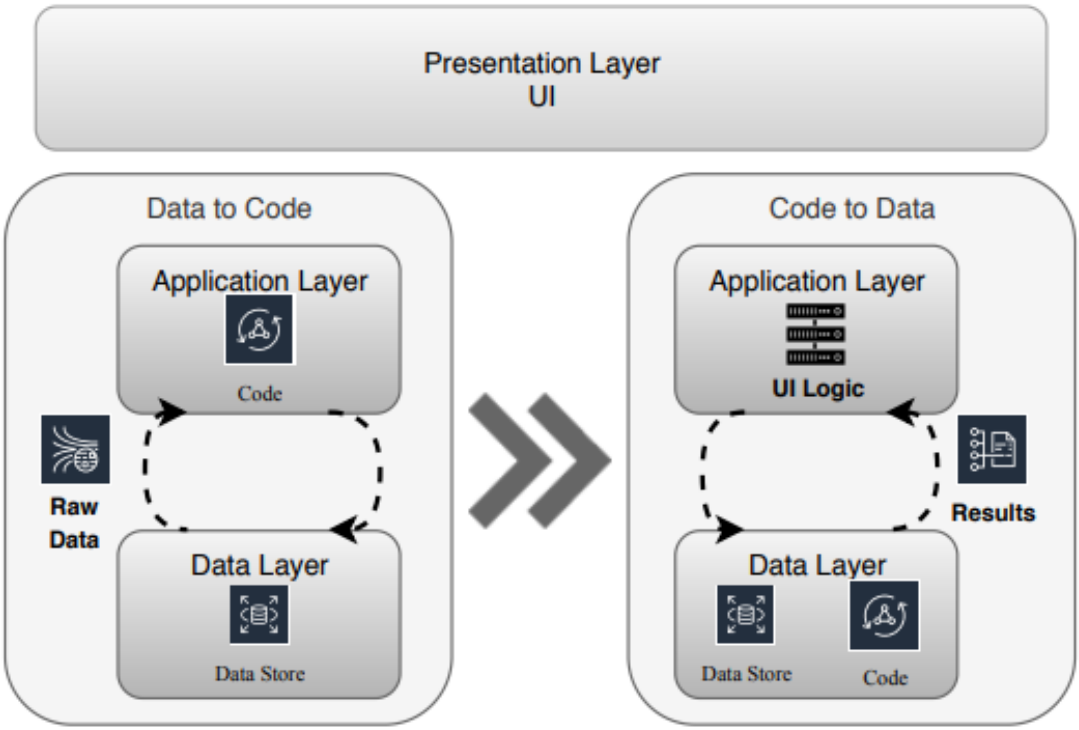


Figure 2.7: Code-to-data and data-to-code comparison [63]

The figure 2.7 is an illustration which presents the comparison between both approaches. It is to note that the code to data approach presents an application layer which mainly focus on the user interface logic.

CDS models are expressed in a Data Definition Language (DDL). DDL is built with traditional SQL as a base and includes various improvements like associations, modifications, and annotations. To consume CDS data, SQL is extended with the CDS Query Language (QL). The QL provides improvements like views within the CDS data model definition. The use of associations outlined by the DDL is also introduced. The Data Control Language (DCL) is used to define instance-based authorization to CDS entities. For a greater improvement of the performance, DCL uses literal conditions which provides comparison between CDS entities as a base for authorization restriction, pushing further down the authorization logic to the database [63].

There are two variants on CDS, as explained on [65]:

- ABAP CDS: This variant can be used even if the database is not a HANA, this means that ABAP CDS is an open DDL language which can be used across many traditional databases as well.
- HANA CDS: It's a database language which allows creating tables views and other structures on a HANA databases.

CDS views give the possibility of creating simple code push down reports and applications. Example of algorithm 2.1 is the example of a code snippet which allows, from a table called SPFLI, which has data relative to airline flights information. This simple piece of code defines a new view, which has the columns stated as its content. On the top of the code there are some notations which start the @ sign, these types of notation allow some simple configuration, like for example setting authorizations for accessing the created view.

```
@AbapCatalog.sqlViewName: 'Z_CDS_VIEW'
@AbapCatalog.compiler.compareFilter: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
define view Z_CDS_VIEW as select from SPFLI {
  key SPFLI.CARRID as CarrierID,
  SPFLI.CONNID as ConnectionID,
  SPFLI.CITYFROM as CityFrom,
  SPFLI.CITYTO as CityTo,
  SPFLI.COUNTRYFR as CountryFrom,
  SPFLI.COUNTRYTO as CountryTo,
  SPFLI.FLIGHTTIME as FlightTime,
  SPFLI.DISTANCE as Distance
}
```

Listing 2.1: ABAP simple CDS view

The following example already applies some logic to the view, on this example it's possible to see some transformation being made using a case statement. It allows to have a new column on the view which is called FlightDurationCategory. This column takes conditional values which are calculated during the view definition. In this case, if the flight time value is less than 120 minutes, it sets the value of the column FlightDurationCategory as short, else if the value is 360 minutes, it categorizes it as Medium, finally, if bigger than that, the category is set as long.

```
@AbapCatalog.sqlViewName: 'Z_CDS_VIEW2'
define view Z_CDS_VIEW2 as select from SPFLI {
  key SPFLI.CARRID as CarrierID,
  SPFLI.CONNID as ConnectionID,
  SPFLI.CITYFROM as CityFrom,
  SPFLI.CITYTO as CityTo,
  SPFLI.COUNTRYFR as CountryFrom,
  SPFLI.COUNTRYTO as CountryTo,
  SPFLI.FLIGHTTIME as FlightTime,
  SPFLI.DISTANCE as Distance,
  case when SPFLI.FLIGHTTIME < 120 then 'Short'
       when SPFLI.FLIGHTTIME < 360 then 'Medium'
       else 'Long'
  end as FlightDurationCategory
}
```

Listing 2.2: ABAP CDS view with data transformation

Core Data Services - APIs, Web services and UI

There is another important functionality related with CDS view which is the fast and efficient way of deploying web services which expose an entity as a form of a CDS entity.

Below there is a listing which includes the code for exposing an entity via service, it includes the Bookshop service definition which contains 2 entities: Books and Authors [66].

```
service BookshopService {

  entity Books {
    key ID : UUID;
    title  : String;
    author : Association to Authors;
  }

  entity Authors {
    key ID : UUID;
    name   : String;
    books  : Association to many Books on books.author = $self;
  }

  action submitOrder (book : Books:ID, quantity : Integer);

}
```

Listing 2.3: CDS with service

Apart from exposing the web services, CDS also allows to use annotations in order to build an automatically generated UI in SAP Fiori, provided by the SAP CAP framework. Below is another example given by SAP for generating an UI for the book catalog [67].

```
annotate CatalogService.Books with @(
  UI: {
    SelectionFields: [ ID, price, currency_code ],
    LineItem: [
      {Value: title},
      {Value: author, Label:'{i18n>Author}'},
      {Value: genre.name},
      {Value: price},
      {Value: currency.symbol, Label:' '},
    ]
  }
);
```

Listing 2.4: CDS with UI

With the code above it's possible to enable the generation of a User Interface which could allow listing of books, per example.

2.7.3 Solution Comparison

This subsection presents a brief comparison between using a custom solution created on purpose for this requirement or using the SAP PPM for the project management and proposal capabilities.

The table below shows the comparison between what can be offered from a custom built solution when compared with the solution provided by SAP:

Table 2.4: Comparison between SAP's PPM and the custom solution which will be provided

	SAP PPM	Custom Solution Proposal
End user complexity	High	Low
Scalability	Medium	High
Integration with HRIS	Complex	Out of the box
Pricing and system maintenance costs	High	Low
Emails Notifications	Not Included	Included

Although the solution provided by SAP PPM has many applicabilities, it also has analytic integration capabilities and does also integrate with SAP HCM Success Factors. This solution is complex and it is oriented for bigger projects, to its overall and cost management, not being internship oriented. Since the organization for which this application is going to be applied also wants low complexity, this can be reached by designing the application together with the future users of the application.

In this way, the solution which is going to be provided in this dissertation is a custom solution in order to meet the customer requirements using SAP Cloud Application Programming model as a platform for development.

Chapter 3

Value Analysis

This project has the goal of evaluating and developing a platform which allows the management of Internship Projects proposals inside an organization. The system will not be ran on its own data alone, but it will need to incorporate data from an Human Resources Information System (HRIS) for further authorization purposes and employee notification. This dissertation also has the objective of analyzing the best solution to develop this application and to integrate the data from the HRIS system.

This chapter will analyze the opportunity and the value of the project which is going to be developed using the New Concept Development (NCD). After that we will analyze the feasibility and applicability of this opportunity. Finally we will decide on many ways of developing this project.

3.1 Innovation Process

Everyday there is a fresh demand for new processes, goods, or services during this technological evolution, which is constantly changing. For these to be successful, they all need to be properly included into an innovation process.

The term "innovation process" refers to the series of actions that businesses, individuals, or other entities take to generate novel concepts, solutions, or procedures that can add value or effect positive change. A problem or opportunity that needs to be solved is typically identified at the beginning of the innovation process. Next, ideas are generated, the most promising ones are chosen, and prototypes are created and tested. The innovation's implementation within a company or on the market is the last phase [68] [69].

3.2 New Concept Development

The first phase of the innovation process is the *Fuzzy Front End* (FFE), this can be separated into two smaller stages. The idea generation and assessment, on which there is an initial brainstorm which originates the idea based on a customer need, it should also frame the technology and cost of the project. On this stage there is also the need of an assessment which can provide overall ideas of the need of the project, as well as the risk of developing it. It should also contemplate all other existing projects. The next stage of the fuzzy front end phase is the concept development and the product planing, on which, after having an idea, there is the need of providing an market analysis for the product, invest on its definition and start the product planning. During the product planning there should be the

consideration around the costs and the timing. This stage also includes defining and writing the specifications of the project as well as presenting an architecture for it [69].

Since the Fuzzy Front End approach can be often misleading and confusing, Koan on [68] presents a way of simplifying and standardizing the mentioned innovation process step as the New Concept Development model.

The NCD model is mainly composed by three key parts:

- The engine: The organization's leadership, culture, and business strategy, which drive the five major variables that the company can influence, constitute the engine.
- The key components: The inner area presents the five elements which are controlled as part of the FFE: opportunity identification, opportunity analysis, idea generation and enrichment, idea selection and concept definition.
- The influencing factors: Organizational capabilities, the outside world (distribution channels, law, governmental policies, clients, competitors, and the political and economic environment) and any relevant enabling sciences, both internal and external, make up the influencing elements. These elements have an impact on innovation at every stage, from conception through commercialization. The firm has little to no control over these affecting elements.

The figure below demonstrates how these key parts interact and the circular module which represents it's cyclic nature.

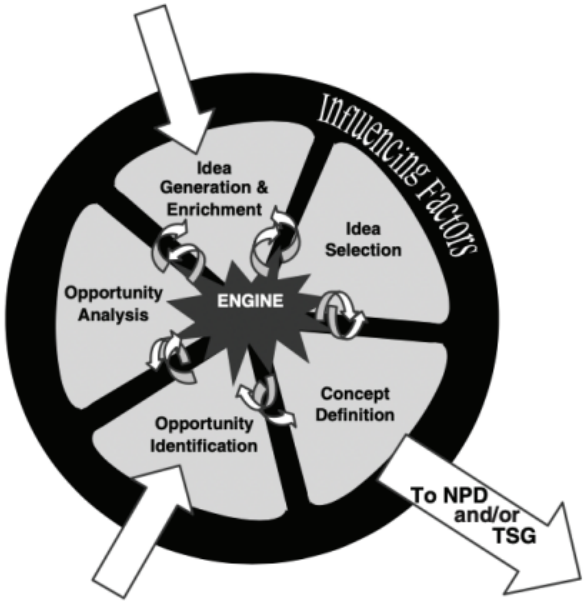


Figure 3.1: NCD Representation by Koen [68]

On the following sections of this chapter it will be described the key-elements of the project on this dissertation context.

3.3 Opportunity Identification

During Opportunity Identification, organizations identify opportunities that for which they might pursue. There can be an shift on resources allocation so that business opportunities are considered. This element is often guided by the business goals [68]. The innovation can also be related with the intern processes of companies, as this is the case on this dissertation theme. In this case, the organization for which this development is thought, intends to search for a value solution for their internship proposal process. It is during these activities that the collaborators of the given company suggest possible technical internship project ideas and submit them to the Human Resources department so that they can be analyzed, accepted (or not) and further submitted to the intended entity (per example schools, universities, governmental programs). After this, the Human Resources department stores the information on a listing file and replies manually to the person which proposed the project. By analyzing this process, there are a some concerns which suggest an innovation on the process:

- The possibility of loosing information through the emails or delaying the process due to a lost or even wrong sent email. This type of information loss is referred as data in motion [70].
- Data in rest, is another factor which may lead to information lost. This is when information rests in file systems and high centralized data storage [70].
- Information is often difficult to find and track for further analysis.
- There is a possibility of input errors while copying the information into an Excel spreadsheet, either due to its lacking of an user friendly interface.
- An additional step is made sending proposals via email and respective acceptance of internship proposals, taking many times more time due to difficulties on tracking emails and information, resulting in an obvious reducing on the productivity.
- Since the internship proposal is often cyclical through schools and other entities, it is possible to miss out opportunities to have internship proposals sent due to lack of notification to the corresponding teams.

3.4 Opportunity Analysis

In this section, an opportunity is evaluated to see whether it merits further investigation. To convert opportunity identification into specific business and technological opportunities, further data is required. In order to do as such, many times it requires to make early technology and marked assessments. It would also require using extra effort in focus groups, studies or scientific experimentation in order to further investigate the opportunity. These type of analysis and the effort given to it results on reducing uncertainties about the opportunity [68].

For this opportunity analysis assessment, it will be used the value proposition map canvas. The value map provides a more organized and thorough description of the characteristics of a certain value proposition in the business model. It divides the value proposition into three major objects: the products and services, pain relievers and gain creators [71].

The list of products and services is just a brief summary of what is provided. It is a list of all the goods and services upon which the value propositions are based. This assists the client

in meeting functional, social, or emotional demands, as well as basic ones. Only when the product is related to a particular customer segment and the jobs, pains, and gains of that customer segment does it produce value in and of itself [71]. The pain reliefs detail the precise methods by which the goods and services relieve particular customer pains. They make it clear how some of the things that irritate the client before, during, or after they try to perform a work or stop them from doing so are meant to be eliminated or reduced [71]. The gain creators explain how the good and service bring benefits to customers. They clearly explain how the product produces results and benefits that the buyer anticipates, wants, or would be pleasantly surprised by, such as practical utility, social gains, pleasant feelings, and cost savings [71].

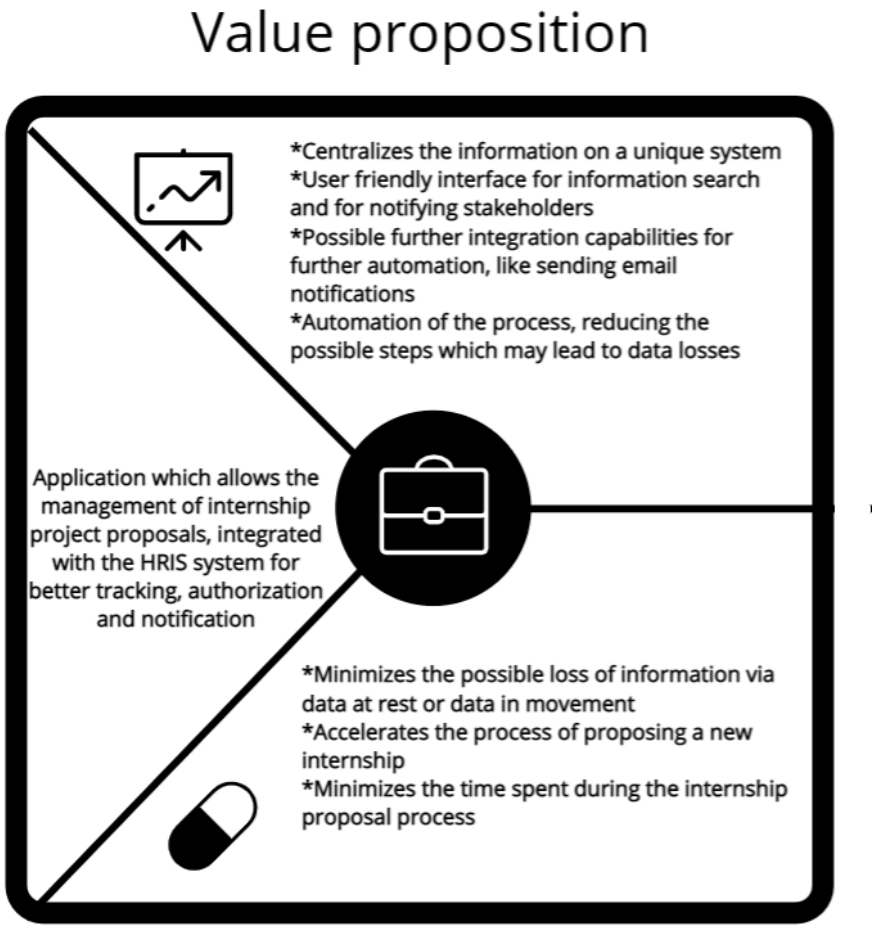


Figure 3.2: Value Map representation, on the left the product or service, on top the gain creator and on the bottom the pain reliefs

From the Value Proposition Map, we noticed that there are some advantages by using an centralized application in order to manage the process of the internships proposals. Among some of the pain killers, as previously mentioned, there can be data loss via human error (by manually copying values from emails to spreadsheets and adapt the internship proposals for further storage). Information can also be lost via *data in motion*, specifically when the proposals are usually done via email and are stored using files. This can also lead to data breaches, due to the high movement of data. The fact of accelerating the process of

proposing new internships often can be seen as a productivity increase, since the employees will have more time to dedicate to other activities.

From the gains perspective, the centralization of information on an unique system will allow better data quality and control. The application will also provide a user friendly interface for data retrieval and further analysis of the internship proposals. The scalability of the system will also allow further integration with new functionalities, like, per example, sending notifications via email automatically regarding the status of some internships. The process automation will also have an impact since it will reduce the human provoked steps leading to less errors and possible data losses.

3.4.1 Comparison with solution proposal

The proposal for this issue is the development of a standalone application which allows users to propose projects for further internships, simplify the storage and search of internship related information. It should also allow different users with different authorizations given the information which is stored on the main Human Capital Management (HCM) system of the organization.

In order to automate the way of notifying the teams of the opening time for the internship proposals, the application should provide a simple notifying mechanism for users about the status of their project proposals.

The current comparison of an actual solution existent on the market for this type of application is made on section 2.7.

3.5 Quality Function Deployment

An overarching concept known as Quality Function Deployment (QFD) offers a way to translate client expectations into the proper technical requirements for each step of product development and manufacturing (i.e., marketing strategies, planning, product design and engineering, prototype evaluation, production process development, production, sales) [72].

To develop the QFD analysis, there was the need to define 2 types of requirements, the client's requirements and the functionalities of the application.

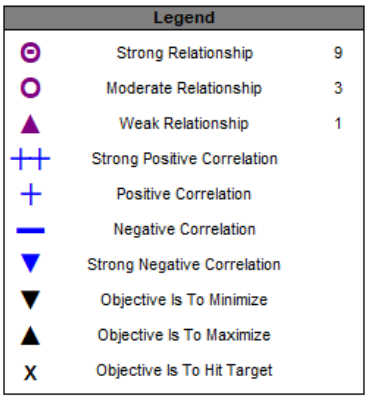
For this application there are the following requirements:

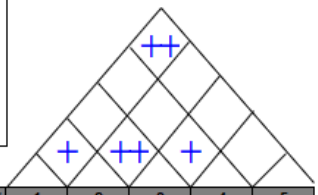
1. Usability and ease of use - this is related to the user experience of the application, how the application should behave visually and how easy it is to handle.
2. Data integrity and centralization - the ability of maintain all the data relative to internships on the same system, with good data quality.
3. Integrated system - this requirement is specifically related with the data integration either from the HRIS system or even with email notification. This requirement will allow further authorization logic to be considered and automatic email notifications using the HRIS centralized data.
4. Notifications - the capability of the system sending email notification to the users either with the status of their proposals or even with the deadlines of project submission for each university.

5. Filtering and ordering capabilities - these types of functionalities will allow users to filter and order the projects existent by a number of characteristics. This will allow management to have better productivity while analyzing the solution proposals.

Figure 3.3 has the representation of the relationships regarding the QFD, and it allows to gather some conclusions:

- The requirement with highest importance is the centralization and the integrity of data.
- There are three customer requirements which are more important which are the centralize information, the requirement which is related with the efficiency of the internship proposals management and data loss prevention.
- The filtering and ordering capability has a strong relationship with the usability/ease of use.
- The data integration and data integrity/centralization are strongly related.





Row #	Max Relationship Value in Row	Relative Weight	Weight / Importance	Demanded Quality (a.k.a. "Customer Requirements" or "Whats") \ Quality Characteristics (a.k.a. "Functional Requirements" or "Hows")	Column # 1 Usability and ease of use	2 Data integrity and centralization	3 Integrated system	4 Notifications	5 Filtering and ordering capabilities
1	9	20,0	4,0	Centralize the information	▲	⊙	⊙		
2	9	20,0	4,0	More efficiency during internship proposal and management	⊙	○		○	⊙
3	9	15,0	3,0	Don't miss oportunity window for project submission	▲		▲	⊙	
4	9	20,0	4,0	Prevent data loss	▲	⊙	○		○
5	9	10,0	2,0	Intuitive way of creating internship proposal	⊙				○
6	9	15,0	3,0	Enable possible integration with HRIS systems	▲	⊙	⊙		
7									
8									
9									
10									
				Target or Limit Value					
				Difficulty (0=Easy to Accomplish, 10=Extremely Difficult)	8	8	7	6	7
				Max Relationship Value in Column	9	9	9	9	9
				Weight / Importance	334,0	501,0	336,0	195,0	270,0
				Relative Weight	20,4	30,6	20,5	11,9	16,5

Figure 3.3: QFD representation, which correlates the Application's Requirements/Functionalities with the Requirements from the customer

Chapter 4

Analysis

The present section's goal is to discuss the implementation's analysis in relation to the solution's functional and non-functional requirements. The proposed solution will then be executed and assessed in the following chapters. This chapter's goal is also to explain the best possible route for project implementation and to make some decisions about how and what to implement during this dissertation. During the chapter, it will be used Unified Modeling Language (UML), which assists in specifying, visualizing and documenting models of systems, helping structuring and designing software artifacts. It helps modeling businesses, rules and both software and non-software systems [73].

4.1 Domain Model

At its worst, business logic can be extremely complicated. Rules and logic define many various circumstances and tilted behavior, and objects were built to operate with this complexity. A Domain Model generates a web of interrelated objects, each of which represents a meaningful individual, whether that individual is as huge as a corporation or as tiny as a single line on an order form [74]. Implementing a Domain Model in an application entails installing an entire layer of objects which model the business in question. These objects imitate the data on real life applications and capture the business rules [74].

As its most important concept regarding the internship project submission there is a central entity which represents the internship proposal or project. These proposals are submitted as projects when universities or other institutions open their cycles. These projects are suggested by employees of the company. After the submission proposal, the team leader reviews the project proposal and accepts it or not. If it is accepted, the project is sent to the institutions and the candidates can apply to the projects. When a given candidate is accepted he is considered as an intern, which is an employee of the organization.

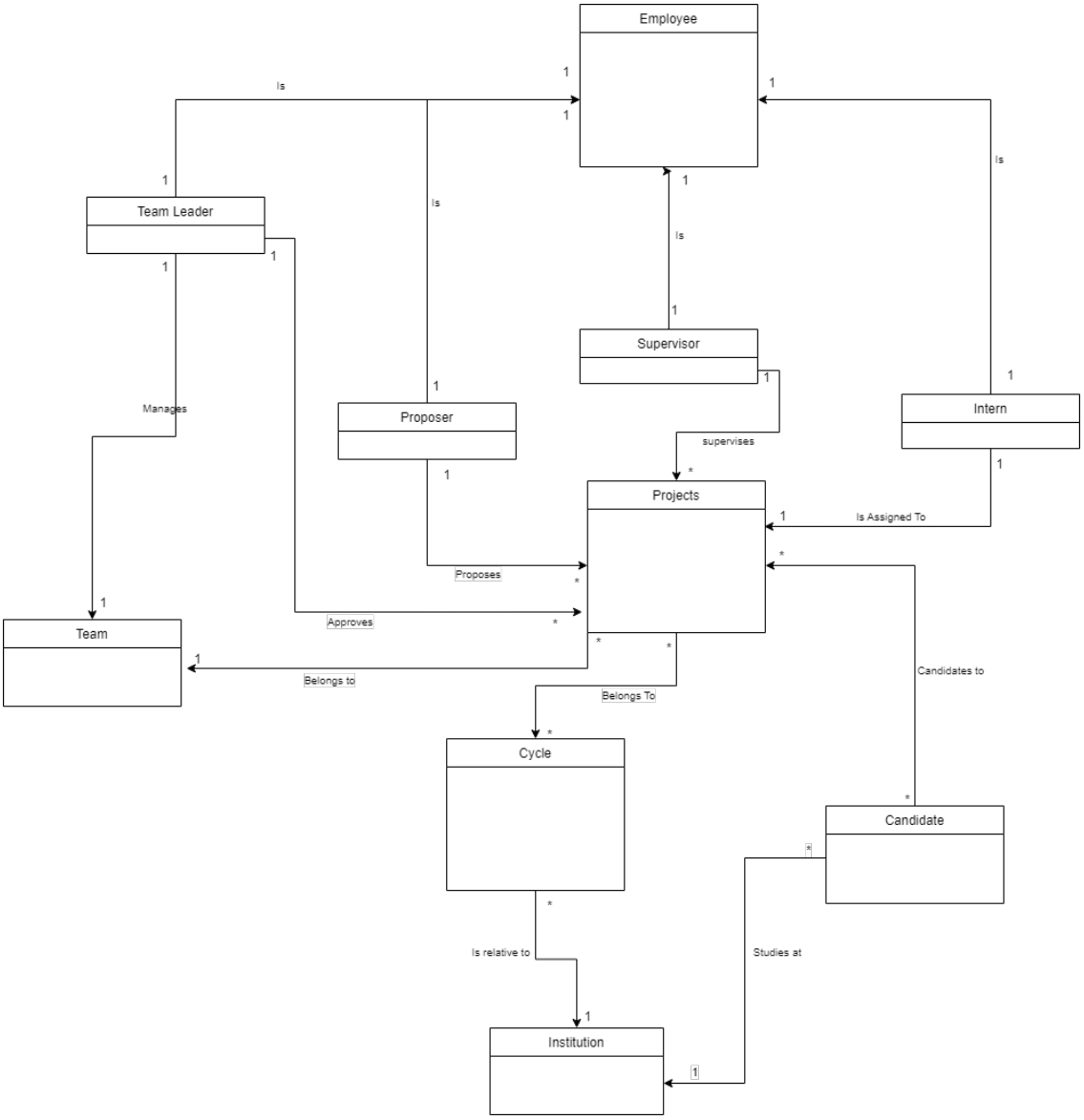


Figure 4.1: Domain Model Diagram

Figure 4.1 represents the state of the Domain design for the application, representing each conceptual class for this specific business and their correspondent relationships. On this particular case, the domain will have a central concept which represents each project and its proposal. These projects are characterized by:

- ID - The unique identifier of the project.
- Name - The name of the project.
- Description - A brief description which gives an overview of the purpose of the project and what is intended.
- StartDate - The begin date of the project, this field should only be populated when the project is assigned to a given candidate.

- EndDate - The end date of the project, as agreed with both the institution and the intern.
- Status - The current status of the project proposal, it represents states like cancelled, submitted to institution, in progress, blocked or closed, per example.
- Document - This is the proposal document which is going to be respectively sent to the institutions.

Each project, will have multiple candidates, which represent students which have interest in a certain internship. These candidates have an unique ID and a name and they can be applicants to many projects.

Another important concept about this specific scenario is the cycle which represent a project cycle within a certain institution. The cycles are described by an unique ID, a name, a correspondent start and end date as well as a due date. The due date has the objective to signalize up until when this cycle is opened to receive project proposals within a certain institution. These cycles can also be further analyzed by the team to better understand if it is worth to cancel or not a project, per example, due to it lacking candidates throughout many cycles. Each institution also represent a model, with an identifier, a name and an area of studies. Each candidate belongs to a given institution, which can be an university, school or even governmental entities, and each cycle is also relative to a given institution.

The company has many employees, which have an ID, a name, an username (which is used on the HRIS system of the organization), an email address, a division and department correspondent to the company and a title.

The company which originated this project also has internal teams, with which each project can be related. These teams are managed by a team leader, which is an employee. The team leader is the person which can approve project proposals so that they can be submitted.

The projects have also an proposer, which is the person which submits the project, a supervisor, which is the person assigned to supervise the intern which the project is allocated to. The proposer, supervisor and intern are all employees.

4.2 Requirement Engineering

This section will gather the main information about the requirements of the application to be developed. Requirements engineering is mainly used to identify what needs to be designed during a project rather than how the application should be designed [75] . FURPS is a strategy for validating prioritized requirements following a thorough grasp of the client's needs and requirements. The abbreviation FURPS stands for Functionality, Usability, Reliability, Performance, and Supportability. Over time, and in response to a pressing need, the FURPS+ was born [76].

4.2.1 Stakeholders

The first part of better understanding the requirements of the project and further provide information about it is identifying the stakeholders, these persons are people which are related to the system in any form and in this way they have a final word on the system, its process and behavior:

- Team leaders - The person which is responsible for a certain team on the organization. This person will have direct impact on the application since it is his responsibility of approving and reviewing the existent and proposed projects.
- Technical team - Persons which belong to the organization and are part of the teams of the company. These persons will be supervisors or will propose projects, thus having a direct interaction with the application.
- Human Resources (HR) Department responsible - Is the person which submits the project to a given entity.

On a brief summary each one of the employees of the company is one of the influencing parts on the system.

Since all the persons above have a certain interaction with the system, they are all considered actors on the system. These persons all will have functionalities on the system which will be specified during the next sections.

4.2.2 Functional Specifications

Functional requirements are represented by the letter F of the FURPS+ model. These requirements are those that specify the software functionality that must be included in the application in order for users to do their duties and meet business objectives. The capabilities of the software product are specified by these functional criteria. They can be translated into a question which is *What the software must do to add value for its stakeholders?* [77].

A use case diagram in the UML can condense the specifics of the system's users (sometimes referred to as actors) and their interactions with it [78].

The functional requirements of this projects are the following:

1. Submit Internship Proposal: the employees of the company should be able to submit project proposals on the application, giving it a name, a brief description and uploading the more detailed documentation about this project proposal.
2. Approve Projects: the team leaders of the company have the capability of approving and cancelling projects, as well as managing their status.
3. Assign the intern: the HR department and the team leaders have the capability of assigning an employee (in this case can be an intern) to a given project.
4. Assign the project supervisor: the team leader updates the project assigning the project supervisor.
5. Accept projects: the project managers have the ability to update the project status.
6. Create, read, update and delete (CRUD) cycles: the HR department employees have the capabilities to completely managing the cycles for which the projects can be submitted.
7. CRUD institutions: the HR department has CRUD capabilities on the institutions for which the projects can be sent. These entities can be created by simply give them a name and a area of action (per example, engineering).
8. Report on all projects: the team leaders and the HR department has the capability to read all projects and to apply filters and analyze them.

9. CRUD the teams: the HR department has the capabilities to create and update teams on the application. If needed they can also delete and report on the team available.
10. CRUD the candidates: the HR department has the capability of creating candidates on the system and associate them to a given project.

Figure 4.2: Use Case Diagram

On figure 4.2 it iss possible to comprehend the different users of the system and their actions:

- Technical Team Member - This user will have the function of submitting internship proposals, it will also be capable of seeing his own projects and check their status. They can also update their project proposals.

- HR Manager - This is the person responsible for the HR department of the organization. The objective is to review all projects, approve projects and update the cycles for internship proposals at intended organizations.
- Team Leader - This user represents the person responsible for the different teams of the organization. This type of user will have the capability of updating the project status and approve projects. They will also have the capability of reviewing all the the projects and also create project proposals.

4.2.3 Non Functional Requirements

These types of requirements will not define any functionality of the system, but they will impact on how the system will behave. They are correspondent to the other letters of the FURPS+ model.

The following requirements will be considered as Non-Functional Requirements:

- Usability and ease of use - Looking at, recording, and defining needs based on user interface issues such as accessibility, interface aesthetics, and consistency within the user interface are all part of usability [76]. In this project, it is defined that the user interface should be intuitive and easy to deal with. As the team is also used to deal with SAP products, the design of the application and its usability must be compilant with the SAP standards. This can be achieved by using the SAP CAP capabilities, since it originates a design on a SAP Fiori standard.
- Reliability - Data integrity is needed on the system, as well as its correct integration. The reliability of data will be achieved by using real time employee data through the APIs provided by the HRIS systems of the company. Another factor that will be relevant is the database modelling and taking advantage of the Data Modeling capabilities of SAP CAP.
- Performance - Requirements related with the start-up time, and the time which certain queries to the data take. This will be approached by using different types of columnar storage and code push-down techniques provided by the framework.
- Supportability - Although there is no client requirement, there are aspects which are going to be considered like scalability and compatibility.
- Design Contraint (+) - Data centralization. The data of the application should be centralized, by this, the database must contain all the relevant data for the characterization of internship project proposals.

4.2.4 Use Cases Analysis

Each use case has its own behaviour. Entities like Institution, Cycles and Candidates have a similar behaviour according with each one of the methods.

Project Creation

The main use case of this system is the internship project proposal by the users, considering all the relevant data for it. The user, which should be an team leader or a member of the technical team, is the employee which is currently logged on to the system. His objective is to submit an internship proposal which will later be reviewed by his manager and, if

approved, it can be further sent to an institution as an Internship Proposal. The submission of the internship proposal is then made manually by the HR department using the document provided during the internship proposal creation.

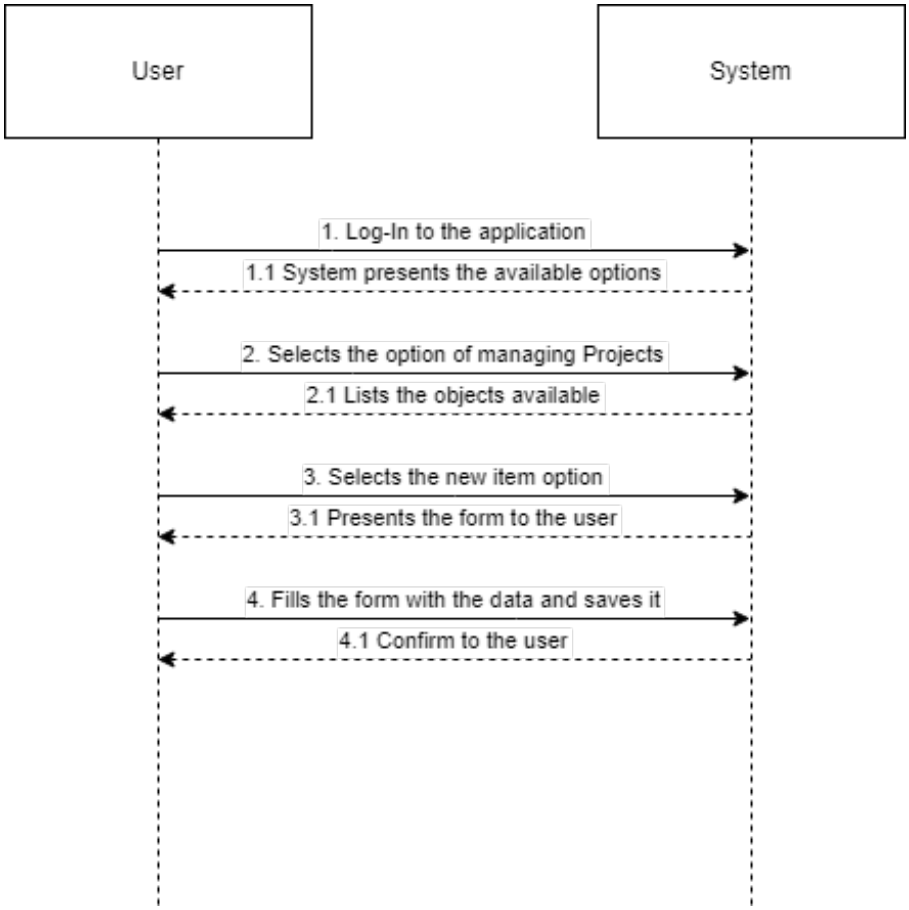


Figure 4.3: Sequence diagram for the use case of creating a new internship project

During the creation of an internship project proposal, the user can not assign a cycle or a candidate to the project, as well as defining its status, since the default status should be set to "Submitted for Manager Approval".

Create Institution, Cycle, Candidate or Team

The steps behind creating the Institution, Cycle or Candidates are very similar to each other. They present 4 different methods:

- Creation of the entity: the process consists in creating an entity according with what was selected and inserted by the user.
- Reading values: this activity consists in listing and showing the details of given projects.
- Update: corresponds to the change of values on a certain object which was previously created.
- Delete: it is the act of erasing a certain record from the database.

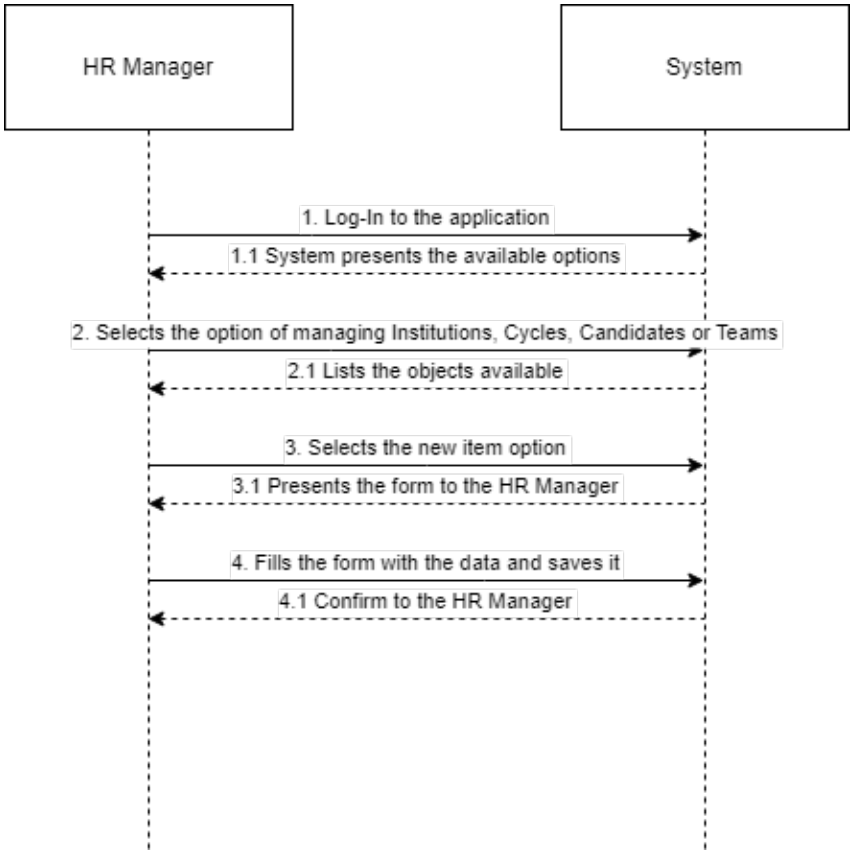


Figure 4.4: Sequence diagram for the use case of creating a new Institute, Cycle, Team or Candidate

Figure 4.4 represents the HR Department user interaction with the system when creating an Institute, Team, Cycle or Candidate.

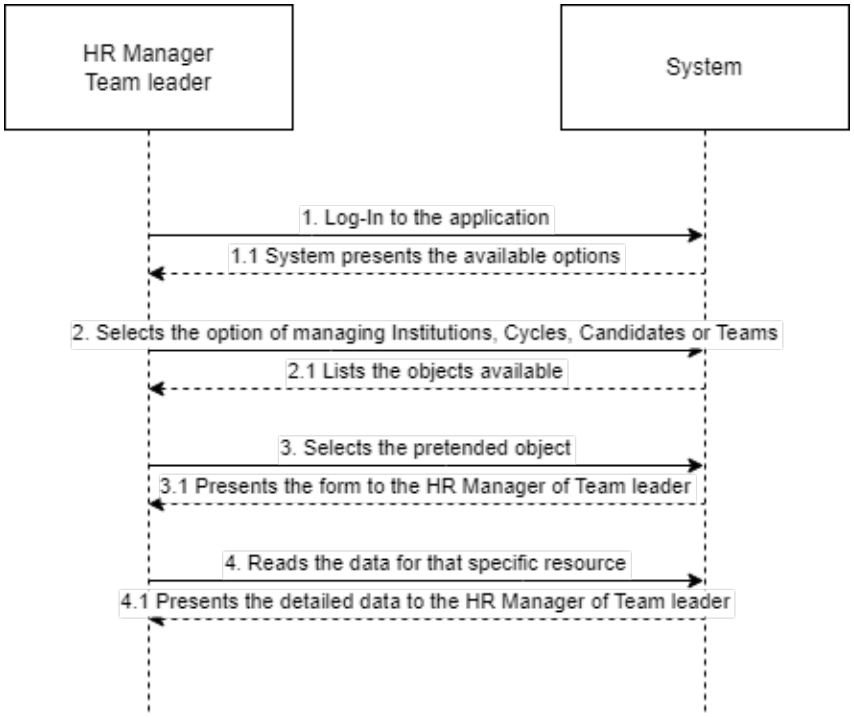


Figure 4.5: Sequence diagram for the reading of Institutes, Cycles, Teams or Candidates

Figure 4.5 represents the capability of the HR department users reading the cycles, teams, candidates and institutions. The points 1 and 2 represent the listing of the projects. On point 3, the user selects a specific entity and opens a more detailed view of the entity, showing more detailed data.

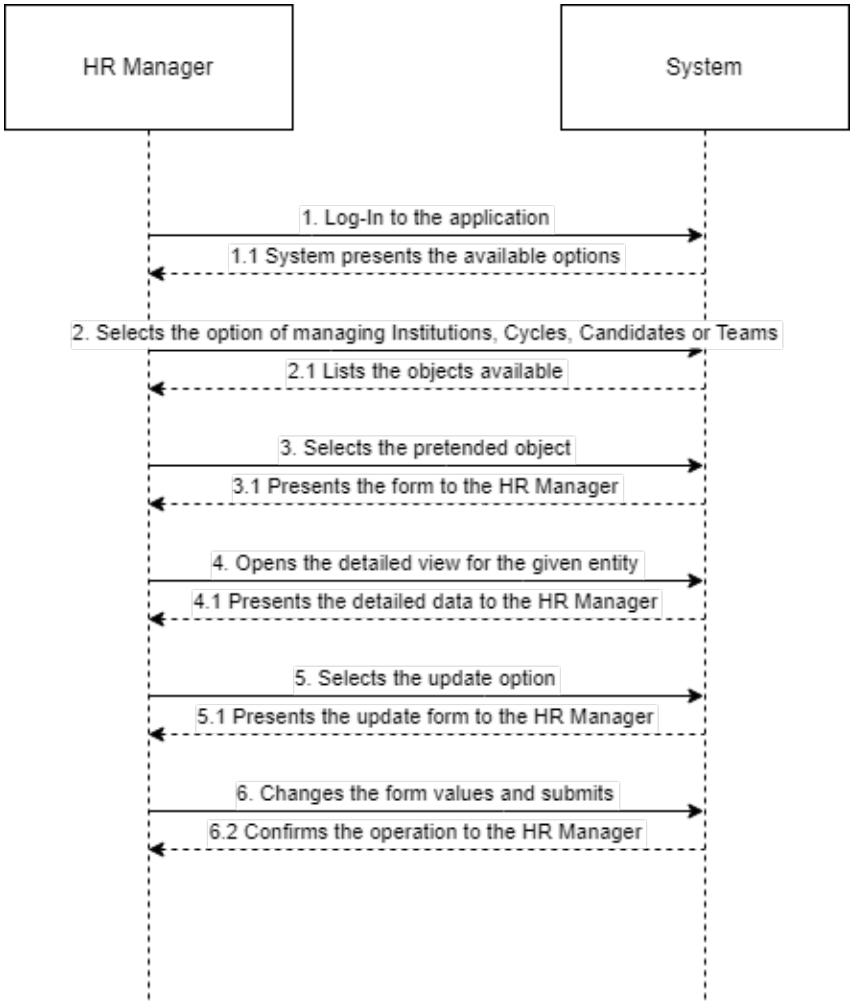


Figure 4.6: Sequence diagram for updating Institute, Cycle, Team or Candidate

The sequence diagram 4.6 presents the sequence actions on the system for updating a given resource from a certain entity.

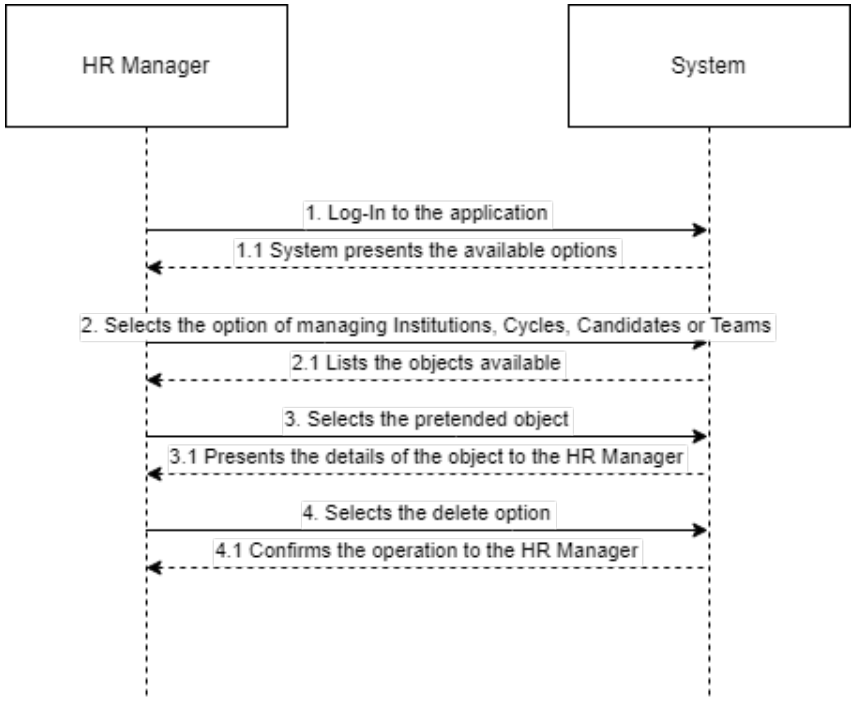


Figure 4.7: Sequence diagram for deleting an Institute, Cycle, Team or Candidate

The system sequence diagram relative to the deletion of records is represented on the diagram 4.7. This diagram represents the initial interaction of the user by logging on to the application. After this step, as in the other operations, the user can select an option regarding which entity he wants to manage, if he really has access to do so. The application lists the objects which are available to the user and he selects the ones which he wants to delete. After this, the user selects the delete option and the system sends an API call to delete the data on the database.

Project Approval

Since the internship project is set to initial status of "Submitted for Manager Approval", the team leaders of the organization can approve the project, setting its status to "Approved by Manager". When the projects are in this status, they are available to be assigned to cycles and they can be sent to institutions accordingly.

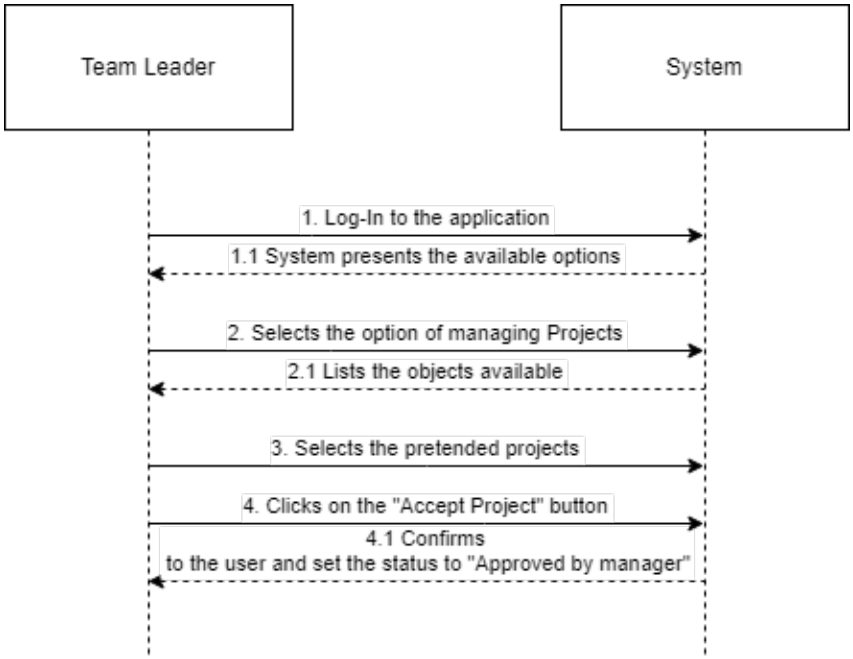


Figure 4.8: Sequence diagram for Approving

On diagram 4.8, it is explained on how the manager approves projects for submission. In this case, the project needs to be in status "Submitted for Manager Approval". The manager lists the projects and selects the intended ones, after this, the manager can select the option of "Accept Projects" and all the selected projects are updated accordingly to a new status.

Assigning internships and interns

Assigning internships to an intern which already exists on SuccessFactors, updating the internship proposal to other status and assign the supervisors of the project are all done through the functionality of updating the project proposal.

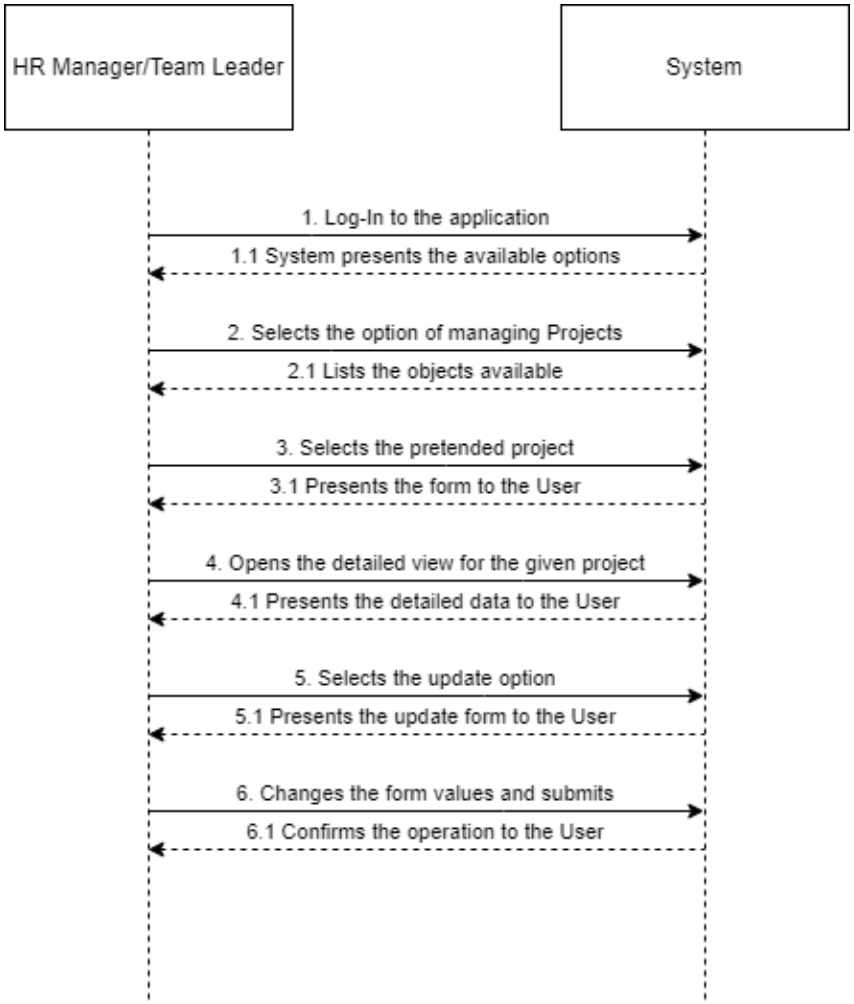


Figure 4.9: Sequence diagram for updating Project

On figure 4.9 it is shown how the updating project works. This functionality is very similar to the update institutions, per example. But it is controlled by the user authorization in this case. Not all fields are updatable, but only the fields for assigning projects to interns and supervisors, as well as assigning candidates to projects and cycles to projects.

Chapter 5

Design

The main goal of this chapter is to present the design and architecture of the solution, needed for fulfilling the requirements of the system. It is during this chapter that it is explained what are the main components of the application, which overall architecture it will have as well as explaining the data structures. It will also include the layers of the system from the User interface to the data maintenance at a database level.

5.1 Overall Architecture

The main purpose of the work is to develop an application which fulfills the specific needs of the real organization. The domain of the project will have a big impact on the development using the framework at hand. The data of this domain will contain two distinct sources, the OData API from SuccessFactors, which will provide information about the employees and the HANA database, which will contain the information regarding the projects and other related information which is not scope in SAP's Human Resource Information System (HRIS).

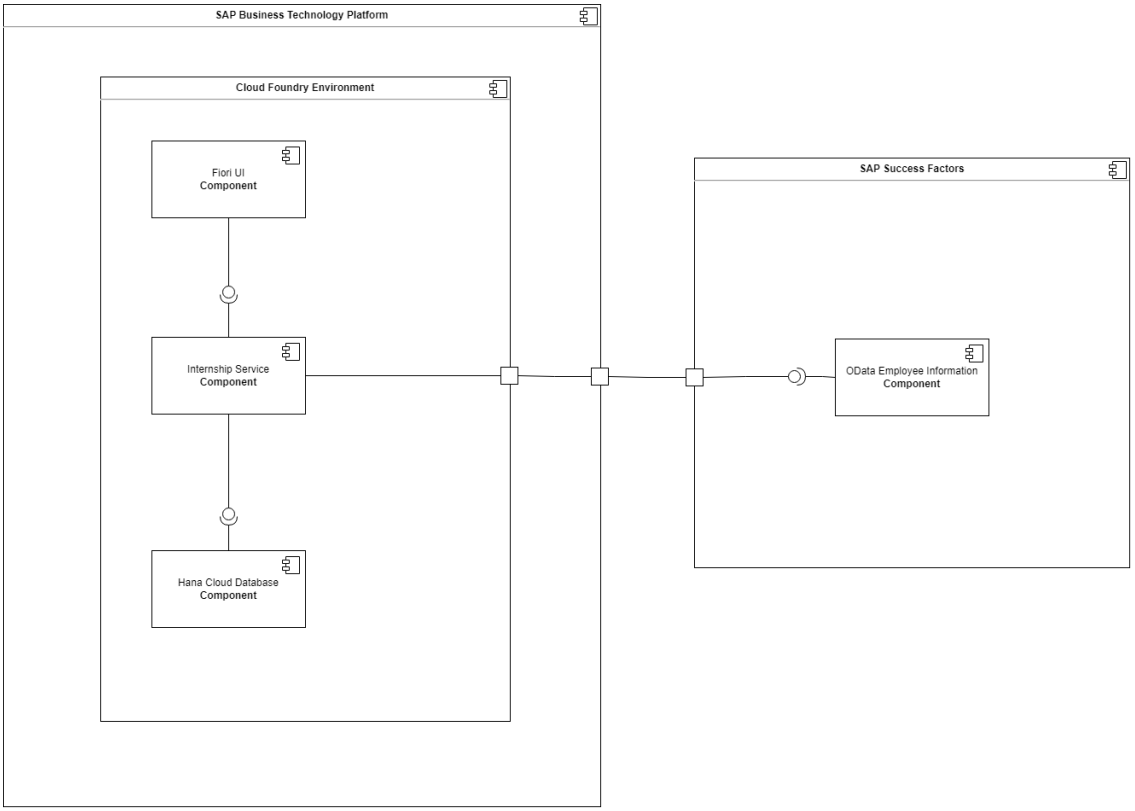


Figure 5.1: Overall Component Diagram

In figure 5.1 it is possible to identify the components which are present on the development. The project's user interface is represented by the Fiori UI application. This user interface will be generated using the framework and its domain driven capabilities using annotations provided by SAP CAP. This application will use the internship service, which is an OData service. This OData service will use an HANA cloud database to store the data. The connection to SuccessFactors will be done using a destination on SAP Business Tehcnology Platform, and this destination will be used to fetch the Employee data which is already present on the HRIS of the company. On the diagram 5.1, this destination is represented by the square gate, on the edge of the Cloud Foundry Environment component. The SuccessFactors component contains the OData entity component which has an interface connection. In this case, this connection is used to query real time data from the systems, which helps achieving the data integrity on the system and improve the accuracy of the data.

5.2 Process View

The process view has the main objective to represent the system behavior throughout the different components. This will be represented by various sequence diagrams which will further explain how the system fully works and the full data flow of the system during its processes. The following subsections will describe how the system behaves for the different scenarios.

5.2.1 Submit Internship Proposal

The project creation represents the moment for which an user creates a new internship project proposal.

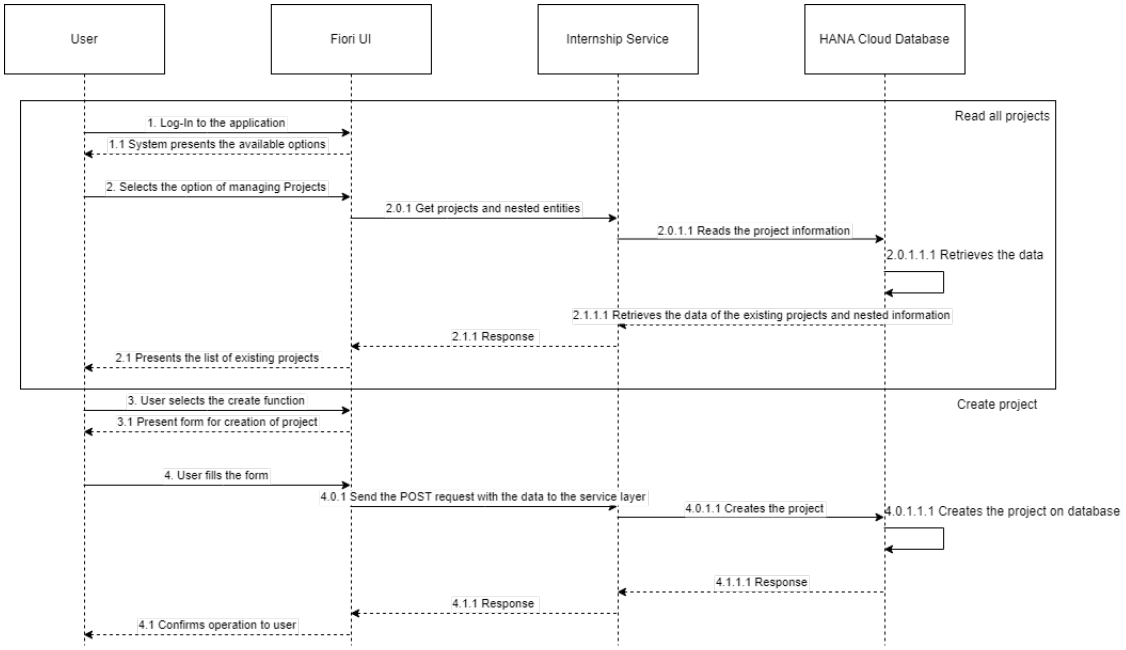


Figure 5.2: Submit Internship Proposal sequence diagram

On figure 5.2 the read all projects behavior is also represented, since all the scenarios will contain this type of sequence.

Once the user selects the create project option, a form is presented with all the relevant information needed for this project creation. After the user confirms and creates the project, the User Interface sends a post request for the service, which sends the create to the database, thus creating the project accordingly.

In this case, the SAP SuccessFactors' OData service is included on the Internship Service, since this is the component responsible for fetching the data of the Human Resources Information System.

5.2.2 Read project

The read project scenario is one of the main scenarios on this application. Its objective is to list all the projects and, if the user intends, retrieve the data relative to that specific project in detail.

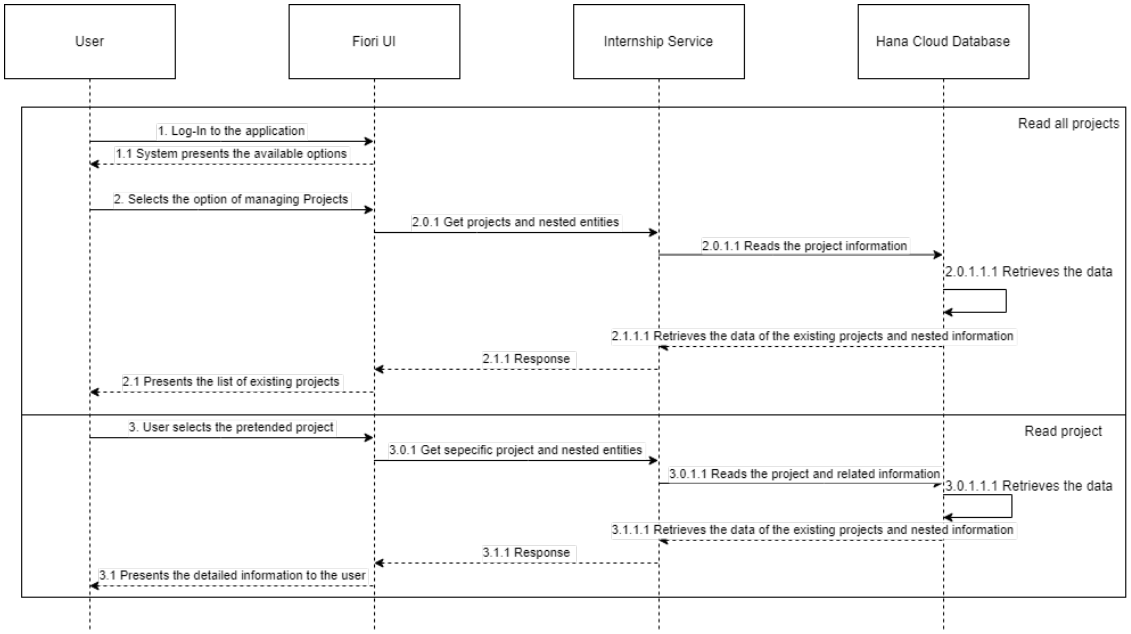


Figure 5.3: Read project sequence diagram

Figure 5.3 represents the behaviour of the system in regards to present to the final user the information of the projects.

Initially, this diagram is separated in 2 distinct parts. As the user logs into the system and chooses the managing project option, he is presented with the list of projects which he has access to (Read all projects). This list only shows the high level information of the project, without detailing the information.

On the second part (Read project), the user is able to choose a specific project and see the content for that project. During both processes, there are nested entities which are relevant to the project which are also fetch. Per example, during the retrieval of data for a given project, it is needed to show the correspondent team information or even the cycles which the project has been assigned to. These nested entities also need to be included throughout the information query.

5.2.3 Update project

The project update allows the user to change any data he is authorized.

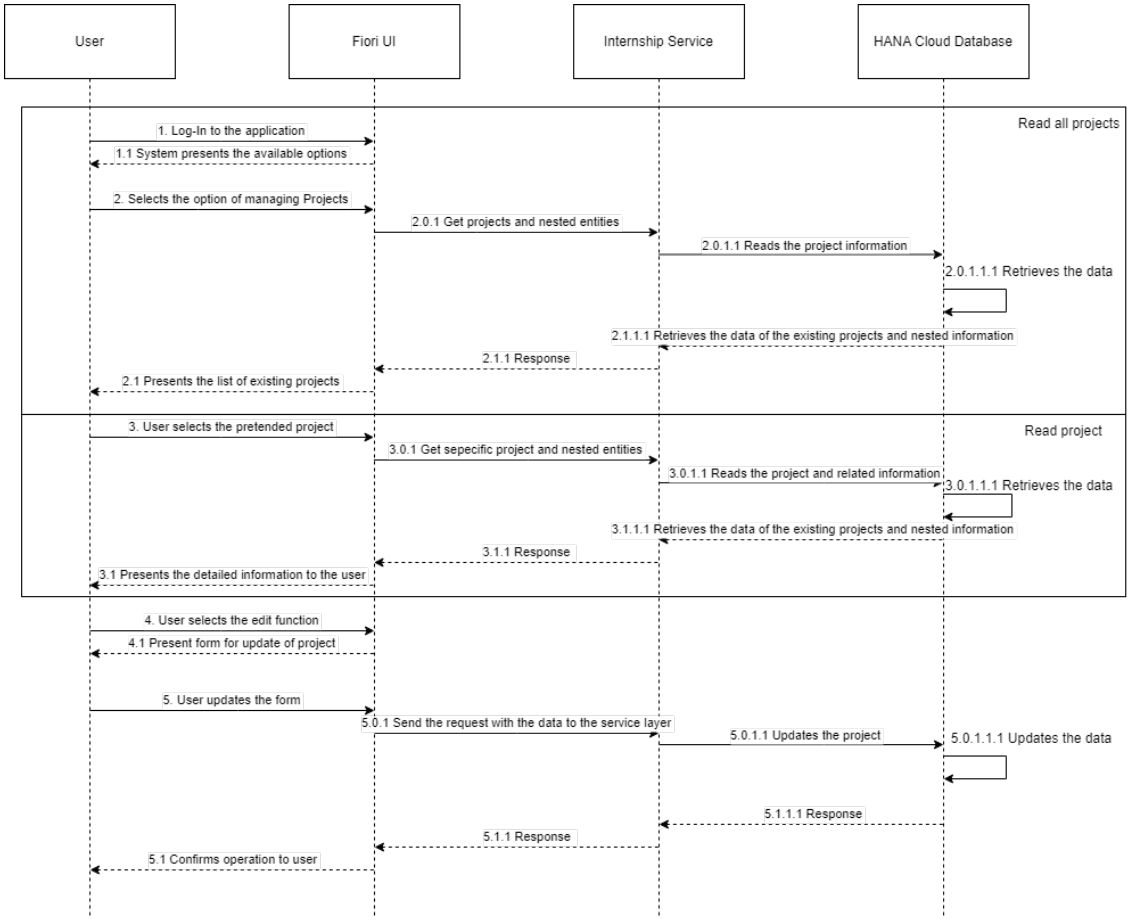


Figure 5.4: Update project sequence diagram

Figure 5.4 represents the process for updating a certain project. This starts with the listing of projects, then the user selects a given project as he wanted to see the details of that project. On that specific part, the user is allowed to choose the update option. When this option is selected, the fields turn editable and the user can update the information accordingly.

Once the user clicks on the "save" button, the process of updating the data begins, by sending the data through the service and updating the data directly on the database.

In this case, the update project status, as well as the intern assignment and the supervisor assignment are done through this same functionality, but only the users with the needed authorization have the access to do it.

5.2.4 Delete project

The project deletion allows the user to delete the data intended.

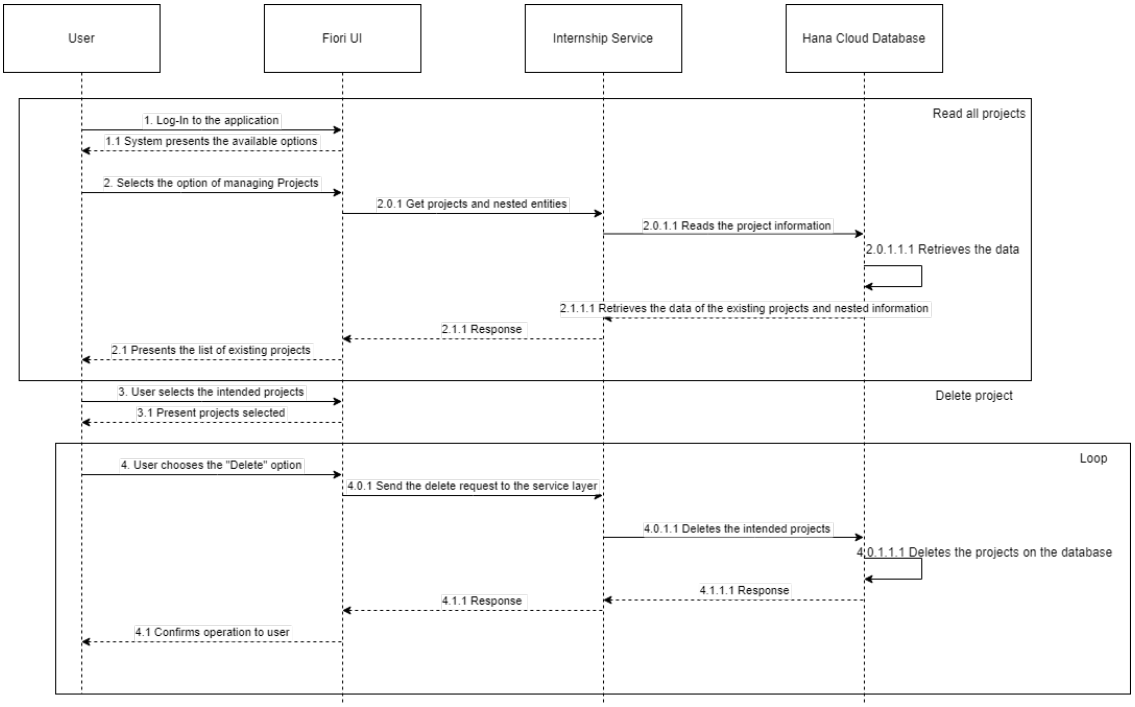


Figure 5.5: Delete project sequence diagram

The illustration 5.5 represents the process for deleting projects. It starts by listing the projects, then the user can select which projects he intends to delete, and by clicking the Delete option the user interface sends the deletion request throughout the service and the database proceeds with the deletion.

5.2.5 Approve project

The approve project functionality will be implemented by using an custom action. This action will update the database with the value which is predefined. In this case, the approval will be made by the managers, to one or more projects at the time, and it will set their status to "Approved by manager".

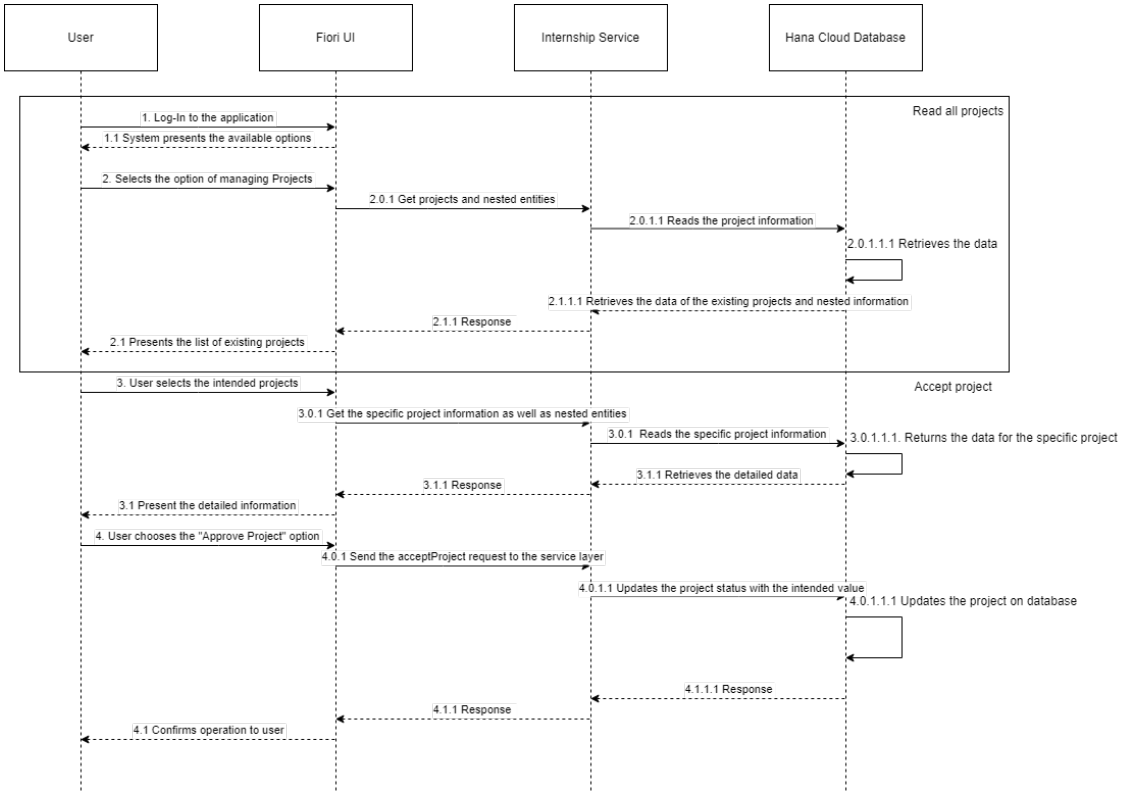


Figure 5.6: Accept project sequence diagram

The figure 5.6 represents the accept project functionality. The user can select one project for approval. After the success of the operation, the user is able to see the project in the "Approved by Manager" status. In this specific case, the user will have the role of a Team Leader.

5.2.6 Other entities

The maintenance of entities like Institutions, Cycles, Candidates and Teams will follow the exact same principle as the designs above. For each specific entities the four operations of Create, List, Read, Delete and Update will work the same way that the project, which was described on the sections above.

5.3 Database Design

The basic ideas and principles that underpin the database architecture of this specific CAP project are succinctly summarized in this chapter. The main objectives of this section is to detail the database tables which are going to be used as part of this specific project as well as how they relate to each other on a relationship model.

5.3.1 Tables

There are 8 tables which will reside on the database for this specific projects. These tables are to serve the purpose of maintaining the structure of the business entities presented and to store the data correctly and accurately.

Team

This table will store the team information which are specific sub-departments of the company for which each project can be related and developed in accordance within a specific team.

Table 5.1: Team table description

Attribute	Description	Data type
ID	Unique team identifier	nvarchar(36) - Universally Unique IDentifier (UUID)
team_name	Name of the team	nvarchar(5000)
team_leader_userId	The user ID of a team leader, which is an employee	nvarchar(100)

Table 5.1 represents the table *Team*. The attribute ID is an UUID which is the primary key of the table. The attribute team_name corresponds to the name of the team, per example, "Web Development team" or "SAP Team". The team_leader_userId establishes the relationship to the user entity of SuccessFactors. It is also a unique value on SuccessFactors and it will represent a company's employee. This field will be used to integrate the SuccessFactors API with the data present in the HANA Database.

Institution

The institution is a simple entity which represents the different schools and entities for which projects can be submitted.

Table 5.2: Institution table description

Attribute	Description	Data type
ID	Unique institution identifier	nvarchar(36) (UUID)
name	Name of the Institution	nvarchar(5000)
area	The area of studies of the given institution	nvarchar(5000)

The database table 5.2, alongside its ID, which is the primary key, has its correspondent name and area.

Candidate

This table is used to store the data of candidates to a given project, which still are not in SuccessFactors, since they are not company's employees.

Table 5.3: Candidate table description

Attribute	Description	Data type
ID	Unique candidate identifier	nvarchar(36) (UUID)
name	Name of the candidate	nvarchar(128)
institution_ID	The institution identifier	nvarchar(36)

The candidate table 5.3 has an unique identifier, the name of the candidate and an Institution ID. The Institution ID is a foreign key to a given institution, which the candidate belongs to.

Cycle

The table cycle captures information regrading cycles of internships within an institution.

Table 5.4: Cycle table description

Attribute	Description	Data type
ID	Unique candidate identifier	nvarchar(36) (UUID)
name	Name of the cycle	nvarchar(256)
institution_ID	The institution identifier	nvarchar(36)
startDate	The start date of the cycle	date
endDate	The end date of the cycle	date
dueDate	The date until which the cycle can receive internship proposals	date

The database table 5.4 alongside an ID which is the primary key, has a name, which has brief information about the timing and institution of the cycle, a start date and end date for which the internships are comprehended, an institution ID which is a foreign key representing an institution and a due date, which is the date until the company can send its internship proposals.

Cycle To Project

This table stores the relationship between the cycles and the projects, since a project can belong to many cycles and a cycle can have many projects.

Table 5.5: Cycle to project table description

Attribute	Description	Data type
ID	Unique institution identifier	nvarchar(36) (UUID)
project_ID	The identifier of a given project	nvarchar(36)
cycle_ID	The identifier of a cycle	nvarchar(36)

The table 5.5 has one identifier, a project ID which is a foreign key to the projects table and a cycle ID which is also a foreign key to the table project.

Candidate To Project

This table stores the relationship between the Candidates and projects. It is representitive of a many to many relationship.

Table 5.6: Candidate to project table description

Attribute	Description	Data type
ID	Unique institution identifier	nvarchar(36) (UUID)
candidate_ID	The identifier of a candidate	nvarchar(36)
project_ID	The identifier of a project	nvarchar(36)

The table 5.6 has one identifier, and two other foreign keys which are related to the *candidate* table and the *project*.

Project

The project represents the main table of this implementation, where the project is represented.

Table 5.7: Project table description

Attribute	Description	Data type
ID	Unique institution identifier	nvarchar(36) (UUID)
name	The name of the project	nvarchar(128)
description	The description of the project	nvarchar(1024)
start date	The start date of the project	date
end date	Planned end date	date
status_ID	Status of the project	integer
team_ID	Team identifier	nvarchar(36)
proposer_userId	The user ID of the project proposer	nvarchar(100)
intern_userId	The user ID of the intern	nvarchar(100)
supervisor_userId	The user ID of the supervisor of the project	nvarchar(100)
document	The pdf document which deeply describes the project	blob

The table 5.7 represents where the project will be stored at a database level. The ID is also an UUID. The name of the project should be brief and highly summarize the project intentions. The description is a text value which stores a brief description of the project and its objectives. The start date is when the project is intended to start and the end date when it is planned to end. The team ID represents a foreign key to the team table. The proposer, intern and supervisor userIds are to store the information which is going to be used to link to the SuccessFactors employee data for those three employees, which have different objectives on the project itself. The document is a media file, PDF, which can be further submitted to the institutions as a project proposal. The Status ID is related to a Status table, which represents the status of the given project.

Status

The table *status* is used in order to store the different status for which a project can be. This information is stored on a table with the objective of being easily maintained and configured if there is a need to extend and have more status.

Table 5.8: Status table description

Attribute	Description	Data type
ID	Status identifier	integer
name	The name of the status	nvarchar(255)
descr	The description of a status	nvarchar(1000)
criticality	The criticality of a status	integer

The table which have the status information is represented on 5.8. The identifier is an integer number, the name represents briefly the status with few words. The description is more detailed than the name. The name is usually more used in screen since it is more user friendly and direct. The criticality is used with the objective to indicate to the user a visual reference of a status. In this case the Fiori UI will use this criticality to represent the status with icons.

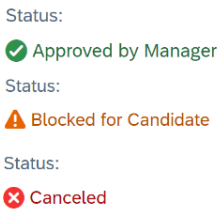


Figure 5.7: Project status icons

The figure 5.7 represent the 3 different icons which are going to be used as the criticality on the user interface. In this case, the name is being shown but it is just indicative and serves as an example in this case. The icons are stored on the table in the criticality, and different icons correspond to different integer numbers. In this case the criticality 1 represents the red icon, the criticality 2 represents the yellow warning icon and the criticality 3 represents the first green icon.

5.4 SuccessFactors OData User entity

In addition to the data stored in the database, there will be a projection of the entity User which exists in the HRIS system and represents the employees of the company. It can also represent an intern since the intern only is assigned to the project when he belongs to the company thus existing on the HRIS system. In this case the user has a few fields and acts like a projection of a table:

- userId - This represents the user ID on the system. It is unique per employee.
- username - This represents the username of the employee on the HRIS system.
- defaultFullName - Represents the full name of the employee in its default form.
- email - The email of the employee.
- department - The department of the employee. This has a relation to the organization structure of the company.
- division - Similar to the department, but on a different level. This also has a relationship with the organizational structure.

- title - The title of the employee, per example Mr. or Mrs.

This data is fetched in real time each time that the application needs it, so it is not stored in a table on the HANA database.

5.5 Entity Relationship

This section gives a brief overview to complement the understanding of the database structure used for the development of the application.

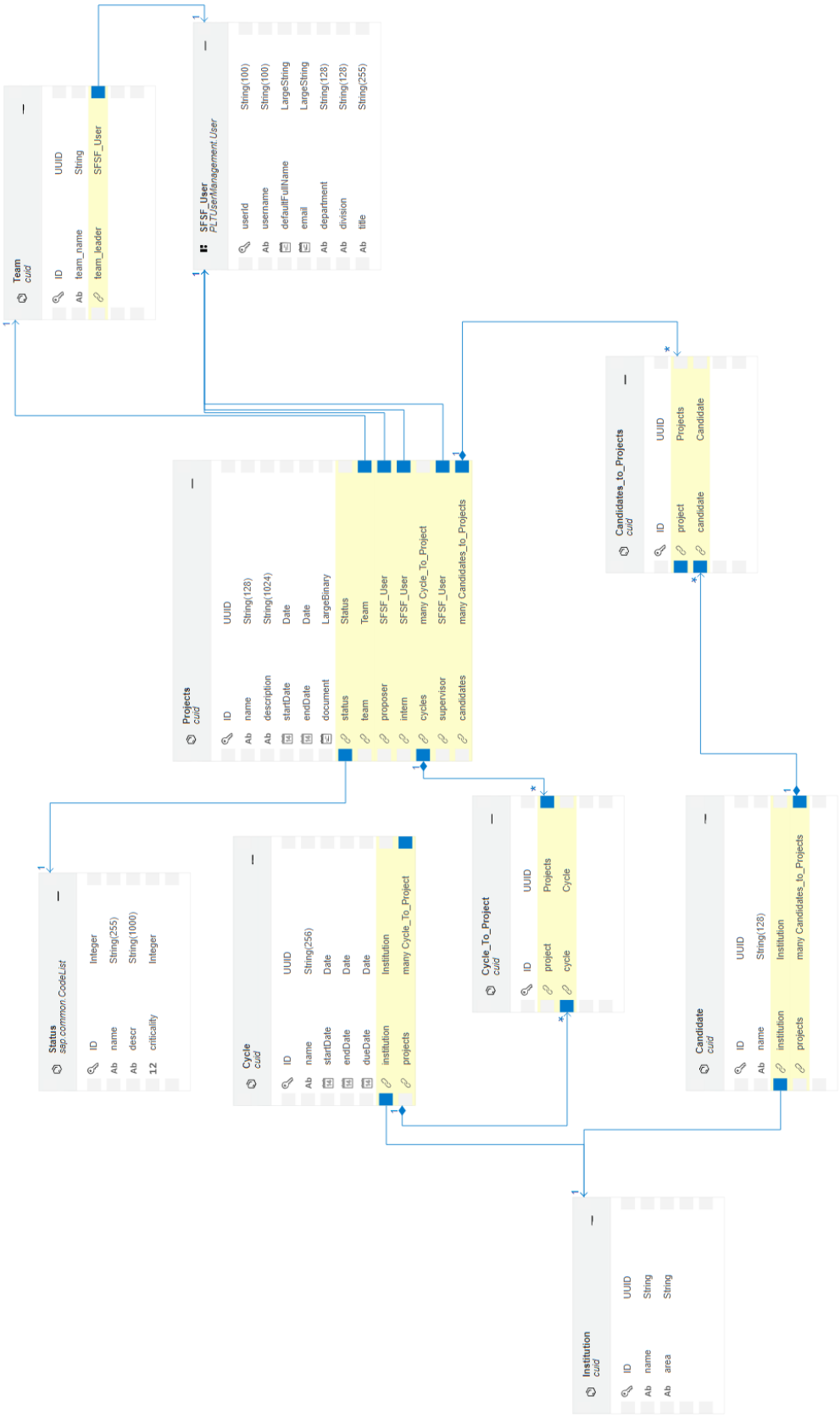


Figure 5.8: Project entity relationship diagram

The figure 5.8 represents the entity relationship. It is to note that SFSF_User is not a database table, as previously mentioned, but represents the employees, so it was included in the diagram. The project is the principal table, and it has two many to many relationships to the table Cycle, this means that one project can belong to many cycles and a cycle can have many projects. The same happens with candidates, since a candidate can be related to multiple projects and one project can have multiple candidates. Tables Cycle_To_Project and Candidates_To_Project are used to establish correctly these many to many relationships.

5.6 Service

The data on the database is accessible through an OData service, which can even be used by external application, or even, if wanted, consumed by a custom user interface built externally from SAP CAP. The entities of the OData services will be:

- Project
- Institution
- Cycle
- Candidate
- Team
- SFSF_User

All the above entities are correspondent to the database tables which were explained on previous sections, except from SFSF_User, which in this case represents the OData entity which exists in SuccessFactors.

Chapter 6

Implementation

This chapter is focused on explaining and detailing the implementation of the solution according to the analysis made in Chapter 4 as well as following the design principles stated in Chapter 5. It will describe the development of the application, detailing some of its steps as well as relating this with the design described on previous chapters.

6.1 SAP Business Technology Platform

The process started with the configuration needed for SAP Business Technology Platform (SAP BTP).

SAP BTP allows the transformation of data into value for the business, and swiftly construct and extend SAP applications.

The SAP BTP services and products are accessible through a variety of cloud infrastructure providers. The multi-cloud foundation supports a variety of programming languages, many locations, and several environments, including Cloud Foundry, ABAP, and Kyma. The SAP BTP cockpit, where the accounts can be accessed and applications and control all of the actions connected to them, is the primary point of entry to the cloud platform [79].

During this section, it is explained how the implementation started and what initial configurations were done in order to start the project.

6.1.1 SAP Business Application Studio

SAP Business Application Studio is a cloud integrated development environment (IDE) which contains many productivity tools oriented for cloud application development. It contains functionalities which allows to speed the development [80].

Initially there is the need to create a new dev space, if not created yet, for starting the developments and accessing the IDE. A dev space is an application development environment that has the resources, tools, and capabilities required.

For the necessary business context, a dev space offers specialized tools and runtimes that are already installed. This makes setting up the development environment easier and faster, and it allows to build, test, and run the solution locally or in the cloud. Each type of development area has a set of predefined extensions that increase the productivity and speed of work [81].

In this case, during the dev space creation, it was used the Full Stack Cloud Application dev space type. This contains the tools which helps the development of SAP CAP applications with higher efficiency. Tools like the Core Data Services' (CDS) Graphical Modeler and the

CAP tools which permits the use of the command-line alongside CDS for easily running the application.

6.1.2 SAP HANA

There was the need of creating a new SAP HANA instance on SAP BTP. This instance is present on the subaccount for dev. This is done by creating the new database.

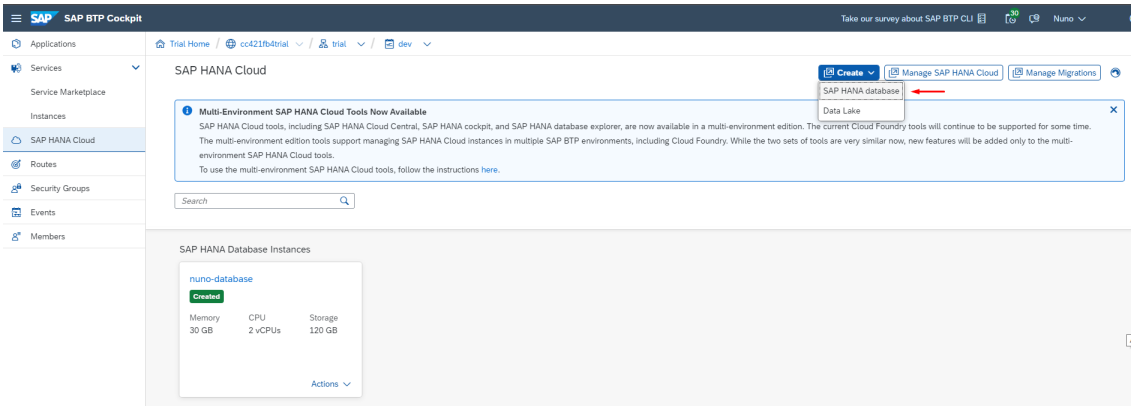


Figure 6.1: HANA Database Creation in dev space on SAP BTP

Figure 6.1 illustrates the step for the creation of the database, which is achieved on the dev space, on SAP HANA Cloud, by selecting create and after "Sap Hana database".

6.1.3 Destination

Destinations allow the connection to remote APIs of any kind of systems. In this case, it was used to connect to SAP SuccessFactors. This configuration is also made in SAP BTP, under the sub account, on destinations [82].

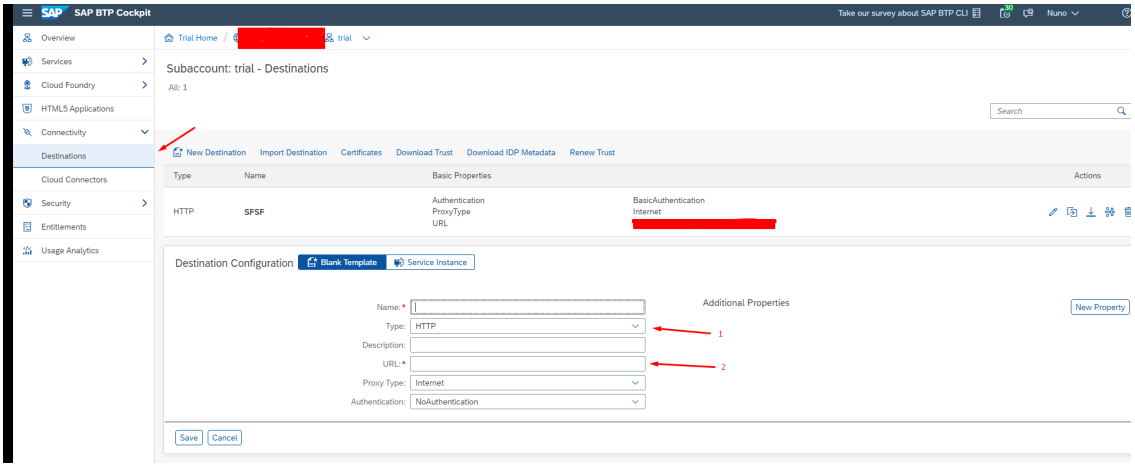


Figure 6.2: Destination creation on SAP BTP

Figure 6.2 demonstrates on how and where a destination can be created. After clicking on new destination, there is the need of creating an unique name so that it can be further used. Then, select the type of protocol, in this case is HTTP, as indicated on point Type of the

figure. The URL relative to the SuccessFactors instance, pointing to the OData V2 service was used. Then it is possible to select the authentication type. For development purposes it was using the basic authentication. In productive scenarios, a technical user in Success Factors should be used. These users, are accounts which have restrictive access and serve the purpose of authenticating APIs.

After this configuration and after saving the destination, it is possible to check if the connection is in fact working, which can be achieved by selecting the function presented on figure 6.3.

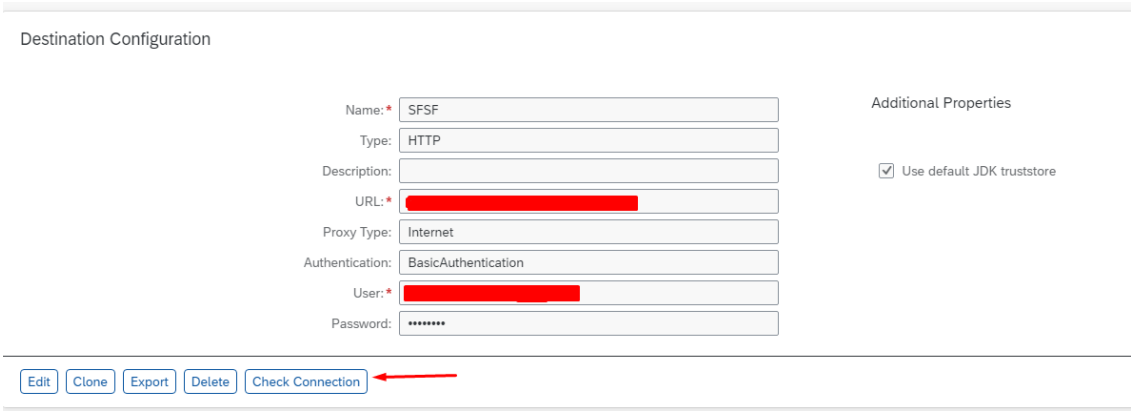


Figure 6.3: Destination creation on SAP BTP

The confirmation message illustrated on figure 6.4 illustrate that the configuration is ready to be used.

Figure 6.4: Destination creation on SAP BTP

6.1.4 Cloud Foundry

The Cloud Foundry environment enables the development of cloud applications. The Cloud Foundry runtime service, which is based on the open-source application platform hosted by the Cloud Foundry Foundation, is contained in it as is the SAP BTP [83].

It is possible to develop new business apps and services using the Cloud Foundry environment, which supports a variety of runtimes, programming languages, libraries, and services. Additionally, it interfaces with SAP HANA XSA, the model of the extended application services.

6.2 Application Development

After the initial setup described on section 6.1, the development can be started on the Business Application Studio.

6.2.1 Initiate the project

Before starting the project, it is needed to login onto the Cloud Foundry environment on the SAP Business Application Studio.

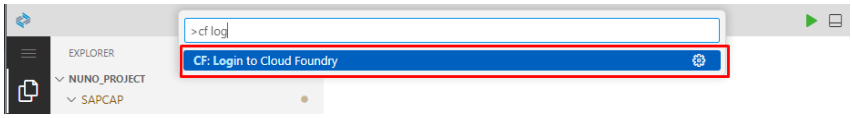


Figure 6.5: Using the command help tool to login to Cloud Foundry

There is a command pallete already provided on the SAP Business Application Studio which allows to use short cuts for commands. Figure 6.5 demonstrates de command used for logging in to the Cloud Foundry environment prior to starting the development.

Then, on the dev space itself, there is a folder which is called projects. This folder exists by default. Inside that folder, command *cds init <project name>* is run.

```
user: projects $ cds init sapcap
Creating new cap project in ./sapcap

Adding feature 'nodejs'...
Done adding features

Continue with 'cd sapcap'
Find samples on https://github.com/SAP-samples/cloud-cap-samples
Learn about next steps at https://cap.cloud.sap
```

Figure 6.6: Command ran to start the CDS project

After running command presented in 6.6, SAP CDS generated a file structure which will be used for this project. This file structure has three folders and 3 documents:

- app - This folder will be used to store documents related with the development of front-end coding.
- db - This folder is to store the documents regarding the database and data model of the application. It will also contain the Comma Separated Value (CSV) files which are going to be used to run the application locally during the development.
- srv - This folder will contain the artifacts used for defining the service and apply business logic accordingly.

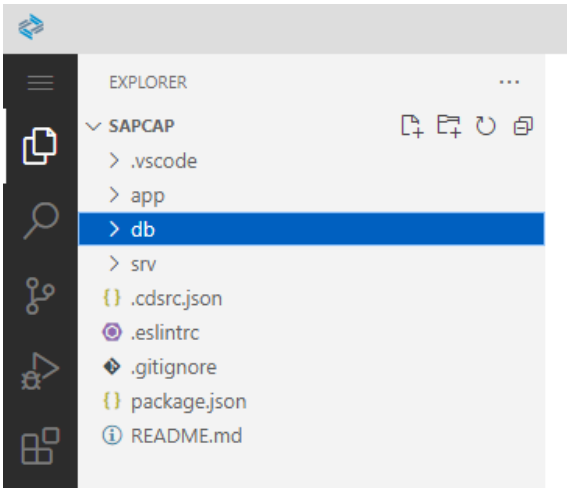


Figure 6.7: Folders generated by running the command

Figure 6.7 represents how the file structure is presented then on the file explorer of the SAP Business Application Studio.

6.2.2 Success Factors import

In order to use the Success Factors entities, there is the need of importing the structure by using an EDMX file. The file can simply be dragged to the folder *srv*. Then, by using the command *cds import srv/ECEmployeeProfile.edmx* it is possible to generate Core Schema Notation files, based on the EDMX files. In this way, we can further use the entities as CDS entities, thus integrating the SuccessFactors entities on the CDS data model.

This import will affect the *package.json* file.

Figure 6.8: Change on package.json file importing the service definition

Figure 6.8 represents the change made on the file under *cds/requires* path. The credentials are further bound using the destination previously considered on section 6.1.3 [84].

6.2.3 Design of the application database using CDS

For starting the development of the application using CDS, it was created a file on the database folder. The file is *projects-model.cds*. In this file we are going to design the database which is going to be used, using the design which was provided in section 5.3. The SAP CAP framework will then use this CDS model to generate the database automatically during the deployment.

Institution

Listing 6.1 is simply instructing the framework the database format which the institution should have. The CUID notation following the *Institution* allows to define the key which is auto-generated.

```
entity Institution : cuid{
    name : String;
    area: String;
}
```

Listing 6.1: CDS Database Institution Definition

SFSF_User

The entity *SFSF_User* will not be an database table, but instead, a projection of the SuccessFactors' OData API. This means that it won't have in fact a table associated to it on the HANA database, but can be referenced through the database model so that entities can relate to the Employees on SuccessFactors.

```
using PLTUserManagement as UM_API from '../srv/external/
    PLTUserManagement.csn';
.
.
.
.
.
@cds.odata.valuelist
entity SFSF_User as projection on UM_API.User {
    key userId,
    username,
    defaultFullName,
    email,
    department,
    division,
    title,
}
```

Listing 6.2: CDS SuccessFactors User Entity definition

As present on listing 6.2, the users can use the *as project on* notation to generate views of certain entities or conjunction of entities.

Status

The *Status* entity will contain the status and the criticality of the projects so that they are used on the project. In this case it has 2 annotations:

- @readonly: this annotation allows to indicate the user that this entity is for read only purposes and can not be updated via UI or OData API.
- @cds.autoexpose: permits the exposure of the entities in services on other entities which have an association to the autoexposed entity.

```
@readonly
@cds.autoexpose
entity Status : sap.common.CodeList {
        key ID          : Integer default 1;
        criticality     : Integer;
}
```

Listing 6.3: CDS SuccessFactors User Entity definition

Listing 6.3 refers to the definition of the *Status* entity. For the initial development, it is being for read only purposes even though that it will change in the future and it will be a maintainable entity for future scalability and configuration.

Table 6.1: Table with the different status

ID	Name	Description	Criticality
1	Submitted for Manager Approval	Submitted for Manager Approval	3
2	Approved by Manager	Approved by Manager	3
3	Submitted to Cycle	Submitted to Cycle	3
4	Blocked for Candidate	Blocked for Candidate	2
5	Started	Project is started	3
6	Canceled	Project is canceled	1
7	Completed	Project is completed	3

Table 6.1 represents the different status which are going to be used to represent the different project status.

Team

The *Team* definition will use the same principle as the *Institution* entity. In this case it will have a CUID defined as primary key.

```
entity Team : cuid  {
    team_leader    : Association to one SFSF_User;
    team_name      : String;
}
```

Listing 6.4: CDS Team definition

On listing 6.4 it is present the association to the entity from SuccessFactors' OData API on the team_leader. On the real database, this will translate to a field on the actual database called *team_leader_ID* which will be used as a foreign key. The date type associated to it will be the same as the key field defined on listing 6.2, the *UserID* field.

Candidate

The *Candidate* entity is defined below. The different point on this entity is that this definition incorporates the *Association to many* notation.

```
entity Candidate : cuid {
    name        : String (128);
    institution : Association to one Institution;
    projects : Association to many Candidates_to_Projects
                    on projects.candidate = $self;
    }
```

Listing 6.5: CDS Candidate definition

As referred on listing 6.5, the association is using a linking entity, which is the *Candidates_to_Projects* association table. This will translate in a many to many relationship with *Projects*. The *Institution* entity is also mentioned as an association.

Candidate_to_Projects

The entity *Candidate_to_Projects* represents the linking between two entities on a many to many relationship.

```
entity Candidates_to_Projects : cuid{
    project   : Association to one Projects;
    candidate : Association to one Candidate;
}
```

Listing 6.6: CDS Candidate to Project definition

Cycle

The *Cycle* will have the same type of relationship to the entity *Project* as the *Candidates* one.

```
entity Cycle : cuid {
    name            : String (256);
    institution     : Association to one Institution;
    startDate       : Date;
    endDate         : Date;
    dueDate         : Date;
    projects        : Association to many Cycle_To_Project
                    on projects.cycle = $self;
}
```

Listing 6.7: CDS Cycle definition

In this case, by analyzing the listing 6.7, it is possible to check the association to many projects using the linking entity *Cycle_To_Project*.

Cycle_To_Project

There is one more linking table which will be the *Cycle_To_Project* table.

```
entity Cycle_To_Project : cuid  {
    project : Association to one Projects;
    cycle : Association to one Cycle;
}
```

Listing 6.8: CDS Cycle_To_Project definition

As in the *Candidates_To_Project* table, the listing 6.8 illustrates the associations to *Projects* and to the *Cycle* entity.

Projects

As stated previously, this is the main entity. It has multiple associations to other tables, either linked or external sources associations.

```
entity Projects : cuid,managed {
    name        : String(128);
    description : String(1024);
    startDate   : Date;
    endDate     : Date;
    status      : Association to one Status ;
    team        : Association to one Team;
    proposer    : Association to one SFSF_User;
    intern      : Association to one SFSF_User;
    cycles      : Composition of many Cycle_To_Project
                      on cycles.project = $self;
    supervisor  : Association to one SFSF_User;
    document    : LargeBinary @Core.MediaType: 'document/pdf' @Core.
    ContentDisposition.Filename: fileName;
    fileName    : String;
    candidates : Composition of many Candidates_to_Projects
                      on candidates.project = $self;
}
```

Listing 6.9: CDS Project definition

On listing 6.9 the main entity is defined. It is possible to see a name, a description, and end and start date simply defined by their data types. The representation of the many to many associations is made via the annotation of *Composition of many*. The snippet for *on columnName.project = $self;* makes reference to the instance of project which is used, then to reference the entity using its ID. In the cycles example, the cycles attribute says that there are many cycles associated to the project.

The document is represented by a *Large Binary* and it has 2 annotations:

- @Core.MediaType - To define which media type is used.
- @Core.ContentDisposition.Filename: fileName - To define where on the structure the file name is stored.

6.2.4 Exposing the OData service using CDS

In this section it will be presented on how it is possible to expose an OData service using the CDS capabilities and annotations.

```
using sfsf.nuno.project.model.db as model from '../db/projects-model';
.
.
.
@odata.draft.enabled
 @(restrict: [
    { grant: ['READ','WRITE'], to: 'HRMANAGER' },
    { grant: ['READ'], to: 'MANAGER' },
    ])
entity Institution as projection on model.Institution;

@odata.draft.enabled
@(restrict: [
    { grant: ['READ','WRITE'], to: 'HRMANAGER' },
    { grant: ['READ'], to: 'MANAGER' },
    ])
entity Cycle as projection on model.Cycle;

@odata.draft.enabled
 @(restrict: [
    { grant: ['READ','WRITE'], to: 'HRMANAGER' },
    { grant: ['READ'], to: 'MANAGER' },
    ])
entity Candidate as projection on model.Candidate;

@odata.draft.enabled
 @(restrict: [
    { grant: ['READ','WRITE'], to: 'HRMANAGER' },
    { grant: ['READ'], to: 'MANAGER' },
    ])
entity Team as projection on model.Team;
```

Listing 6.10: CDS Service for Institutions

The listing 6.10 represents the simpler OData entities and how they are exposed as a service. In this case, we import the database models on the initial *using* statement. Each entity is being exposed as projections on database tables.

Then it is possible to annotate the entities with the required annotations. In this case, it was used the *@restrict* annotation, which allows the definition of certain roles and users for authentication control. In this case, only users which are from the type *HRMANAGER* or *MANAGER* are able to access those entities. Normal users won't be able to access them. The *READ* access was given to the *MANAGER* which represents the team leaders due to the need of accessing those entities during expansions of the project entity itself.

The other annotation, *@odata.draft.enabled* allows the saving of draft versions of the entities, when per example the system shuts down unexpectedly before the user saving or the user doesn't want to save the data immediately on the database.

It is to note that all the $expand, $filter and $select are already implemented by the framework when entities are exposed, so there is no coding required for those functionalities.

Regarding the Project entity, there were some difficulties when exposing the service. This was mainly due to the specific needs of authorization control for this entity. Since there are some specific fields which should only be allowed to edit by some certain users, the CDS annotations cannot be applied directly to certain places.

The most ideal way of implementation the projects functionalities in order to comply with the authorization necessities was by creating two different services which have different types of users assigned:

- *Project* - This entity has the main objective of containing administrative functionalities. Use cases like Accept Projects, associate new candidates to certain projects, assigning the intern and the supervisor are examples of administrative tasks which should be done by this entity.
- *ProjectUser* - The *ProjectUser* is the entity user to submit the initial project proposal, by a normal employee of the company. In this case, the users which are not team managers nor members of the HR Department have access to this entity.

It is important to note that both these OData entities point to the same database table, but have different field selections.

```
1   @odata.draft.enabled
2   @cds.redirection.target
3   entity Project as projection on model.Projects actions {
4   @( cds.odata.bindingparameter.name : '_it',
5   Common.SideEffects :
6   { TargetEntities: [ '_it/status' ], }
7   )
8   action acceptProject();
9   action submitCycle();
10  };
11  annotate Project with @(requires: ['HRMANAGER','MANAGER']);
```

Listing 6.11: CDS Service for Project Managment

The code snippet 6.11 demonstrates in this case how the *Project* entity is represented. There are a few annotations to take into consideration:

- @cds.odata.bindingparameter.name - This annotation allows the application to update immediately the status ID once a certain action runs.
- @cds.redirection.target - In this case, this annotation is mandatory since there are 2 projections on the same database table. In this case, it is needed to know the preferred entity in case of redirection to that same table, since it is used by the *Cycle_To_Project* and *Candidate_To_Project* tables.
- @(requires: ['HRMANAGER','MANAGER']) - For further authorization checks on this specific entities, the annotation *requires* is used.

The *actions* keyword allows the definition of actions. These actions can be further handled to have certain behaviours which are not enabled by the default Create, Read, Update and Delete actions. In this case we have x actions:

- acceptProject - Action to accept the project when it is on the status of *Submitted by User*
- submitCycle - Action to define the status of the project as *Submitted to Cycle*

In this case the actions are going to be further handled by *javascript* code, on the *project-service.js* file.

```
1     @odata.draft.enabled
2     @(restrict: [
3     { grant: ['READ','WRITE'], where: 'createdBy = $user' },
4     ])
5     entity ProjectUser as projection on model.Projects excluding{
      candidates, cycles }
```

Listing 6.12: CDS Service for Project Submission

On the listing 6.12 the *ProjectUser* entity is further defined and configured. In this case, the *restrict* annotation will serve the purpose of each user having access to see and update their own project ideas which were submitted. During the definition of this entity there were some items which were excluded, in this case the *candidates and cycles* are excluded since the users which submit the projects and don't have the necessary authorizations to access the *Project* entity are not due to make changes on those embedded entities.

In this way, the basic implementation of the service is done, the basic CRUD operations are now available and ready to be used.

6.2.5 Custom Logic

In order to execute special transformations and further complex logic apart from the basic capabilities of the CRUD operations, is is possible to implement custom code on each action. It is possible to add actions for each entity before, during and after each operation: *CREATE, DELETE, UPDATE and READ*.

The custom actions, like the *acceptProject* action, should be implemented also using this capability.

```
1     this.on('acceptProject', async (req) => {
2         const n = await UPDATE(Project).set({ status_ID: 2 }).where({ ID
      : req.params[0].ID })
3     });
```

Listing 6.13: Javascript implementation of acceptProject functionality

The listing 6.13 presents how the *acceptProject* is implemented. In this case it is used the UPDATE functionality on the *Project* entity to set the status_ID to 2 which represents the Approved by Manager status. In this way, the managers can analyze the project proposal and accept the project accordingly.

In this case, the same kind of logic is going to be applied to other actions, per example blocking the project for some candidate.

6.2.6 User Interface CDS definition

The user interface configuration is based on the services which were previously defined. The user interface is going to be built using the capabilities of UI annotations of CDS. On the service folder, the file *project-service-ui.cds* is going to be used to make this development.

In this case, the example which is going to be used is the Cycle screen for both maintenance and overall view.

Then the service is annotated accordingly, starting with the @ui annotations. These annotations will allow to design the multiple screens for the application

```
using sfsf.projman.service.ProjectManager as service from './project-service';
```

Listing 6.14: Javascript implementation of acceptProject functionality

In listing 6.14, the project service is imported.

Then the annotations should be made by using the *annotate service.entity with @....* notation.

```
annotate service.Cycle with @(UI : {
    UpdateHidden         : false,
    DeleteHidden         : false,
    CreateHidden         : false,
    Identification       : [{Value : name}],
    HeaderInfo           : {
        $Type            : 'UI.HeaderInfoType',
        TypeName         : 'Cycle',
        TypeNamePlural   : 'Cycles',
        Title            : {
            $Type : 'UI.DataField',
            Label : 'Cycle ID',
            Value : ID
        }
    },
```

Listing 6.15: CDS UI Annotation - options and header

Listing 6.15 represents the top lines of CDS UI notation definition. At these lines, initially it is defined the functionalities of the UI. In this specific case, the update, delete and creation of entities are not hidden, so the UI automatically generates creation and update capabilities for the entity. Then, it is chosen the UI identification, in this case, the value is name so that it can be easier read by the user. After that, the header info is presented, which in this case it is simply the identification of the entity.

After defining the capabilities and header information, it is needed to define the line items. These line items will be used on the initial screen where all the entities which are active are shown.

```
1   LineItem                : [
2       {
3           $Type : 'UI.DataField',
4           Value : ID
5       },
6       {
7           $Type : 'UI.DataField',
8           Value : name,
9           Label : 'Cycle Name'
10      },
11      {
12          $Type : 'UI.DataField',
13          Value : institution.name,
14          ![@Common.FieldControl] : #ReadOnly,
15          Label : 'Institution Name'
16      },
17      {
18          $Type : 'UI.DataField',
19          Value : institution.area,
20          ![@Common.FieldControl] : #ReadOnly,
21          Label : 'Institution Area'
22      },
23      {
24          $Type : 'UI.DataField',
25          Value : startDate,
26          Label : 'Start Date'
27      },
28      {
29          $Type : 'UI.DataField',
30          Value : endDate,
31          Label : 'End Date'
32      },
33      {
34          $Type : 'UI.DataField',
35          Value : dueDate,
36          Label : 'Due Date'
37      }
38  ],
```

Listing 6.16: CDS UI Annotation - Line Item

Listing 6.16 is showing the information which is going to be initially presented to the user for each cycle which exists on the system. In this case, the direct fields are the *ID, name, startDate, endDate and dueDate*. The label information is used in order to show user friendly names to the user. The fields for nested entities are presented for the institution in this case. The values for institution name and institution area are being fetch on lines 13 and 19 of the listing.

For filtering capabilities, the *SelectionFields* annotation is used.

```
    SelectionFields      : [
        ID,
        name,
        institution_ID,
        institution.name,
        institution.area,
        startDate,
        endDate,
        dueDate
    ],
```

Listing 6.17: CDS Facets definition

On listing 6.17, the fields for which this entity can be filtered by are defined. In this way, the fields defined have filters generated automatically for the interface.

Since the line items represent the initial screen showed to the user, then if the user selects one entity, the details are shown. These details can be sectioned on the user interface on the detail screen. For this, facets are used.

```
    Facets                : [{
        $Type   : 'UI.ReferenceFacet',
        ID      : 'CycleDetails',
        Target  : '@UI.FieldGroup#Details',
        Label   : 'Details'
    },
    {
        $Type   : 'UI.ReferenceFacet',
        ID      : 'InstitutionDetailsCyclIe',
        Target  : '@UI.FieldGroup#InstitutionDetails',
        Label   : 'Institution  Details'
    }],
```

Listing 6.18: CDS Facets definition

Code snippet 6.18 shows the information regarding the types of facets which will be used as well as their disposition. In this case the details information will have a section for Cycle details and other for specific institution details.

After defining the facets, , it is needed to specify the fields which are going to be used on the facets.

```
    FieldGroup #InstitutionDetails  : {Data : [
     {
          $Type          : 'UI.DataField',
          Value          : institution_ID,
          Label          : 'Institution ID'
     },
     {
          $Type          : 'UI.DataField',
          Value          : institution.name,
          ![@Common.FieldControl] : #ReadOnly,
          Label          : 'Institution Name'
     }]},
     FieldGroup #Details  : {Data : [
     {
          $Type          : 'UI.DataField',
          Value          : name,
          Label          : 'Cycle Name'
     },
     {
          $Type          : 'UI.DataField',
          Value          :  startDate,
          Label          : 'Start Date'
     },
     {
          $Type          : 'UI.DataField',
          Value          : endDate,
          Label          : 'End Date'
     },
     {
          $Type          : 'UI.DataField',
          Value          : dueDate,
          Label          : 'Due Date'
     }
     ]}
});
```

Listing 6.19: CDS Field Groups

Listing 6.19 represents the facets *InstitutionDetails* and *Details*. In this case the facet institution details will contain the information regarding the institution and the details information will contain the dates and cycle name.

For more specific cases, another two annotations were applied.

```
annotate service.Cycle @(Capabilities : {
     Insertable          : true,
     Deletable           : true,
     Updatable           : true
});
annotate service.Cycle @(Common : {
     SideEffects #InstitutionChanged : {
          SourceProperties : ['institution_ID'],
          TargetEntities   : [institution]
     }
     });
```

Listing 6.20: CDS Field Groups

Listing 6.20 has two new annotations. The *@Capabilities* allows to define which operations are available. The *@Common - Side Effects* annotation is used for when creating a cycle, if the institution_ID field is changed, the other fields are automatically updated accordingly.

Finally, it is needed to add some configuration for the fields which will need value help.

```
annotate service.Cycle with {
    institution      @(
        Common : {
            Text             : institution.name,
            TextArrangement  : #TextOnly,
            FieldControl     : #Mandatory,
            ValueList        : {
                $Type           : 'Common.ValueListType',
                CollectionPath  : 'Institution',
                Parameters      : [
                    {
                        $Type             : 'Common.
    ValueListParameterInOut',
                        LocalDataProperty : 'institution_ID',
                        ValueListProperty : 'ID'
                    },
                    {
                        $Type             : 'Common.
    ValueListParameterDisplayOnly',
                        ValueListProperty : 'name'
                    }
                ],
                Label : 'Institutions'
            }
        },
        title   : 'Name'
    );
}
```

Listing 6.21: CDS Field Groups

Code 6.21 shows the information for the institution to present it as a value list. This allows the user to select the intended institution from a value help section when creating or editing a cycle.

After this the basic screen logic is implemented, the user interface can finally be tested.

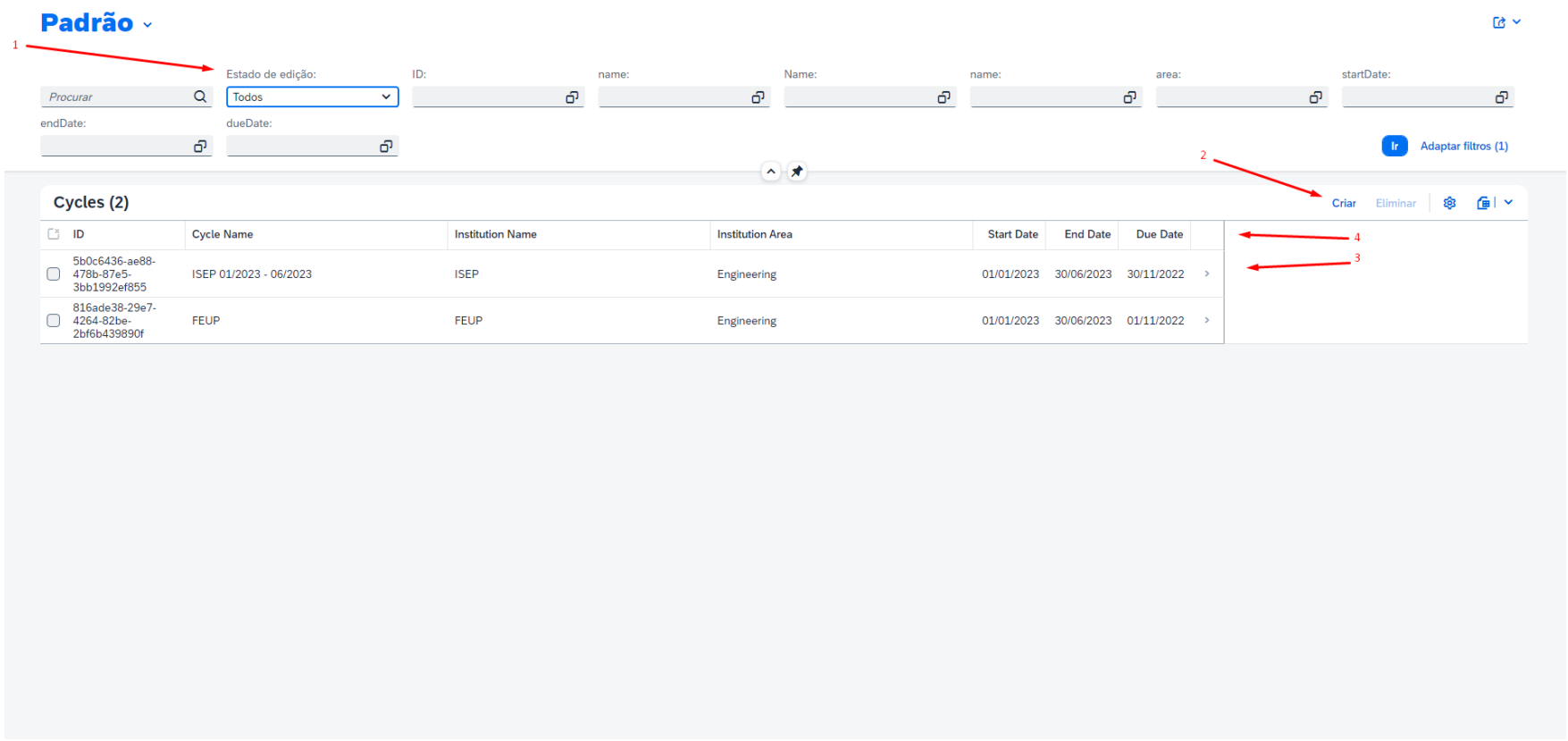


Figure 6.9: Cycle initial screen

Figure 6.9 demonstrates how the interface looks when the initial screen is shown to the user. In this case, the ID is shown so that the user can further use it on value helps if needed, since the names can get very similar.

The following points are regarding the previous shown implementations and they are going to be described as follows:

1. Represents the columns chosen as line items on annotation 6.16
2. Example of one cycle fetch from the database.
3. The creation option, which is not hidden, as described on listing 6.15
4. The filtering capabilities, which are defined from the selection fields on 6.17

By clicking on one of the chosen entities, it is possible to show the details of a given entity.

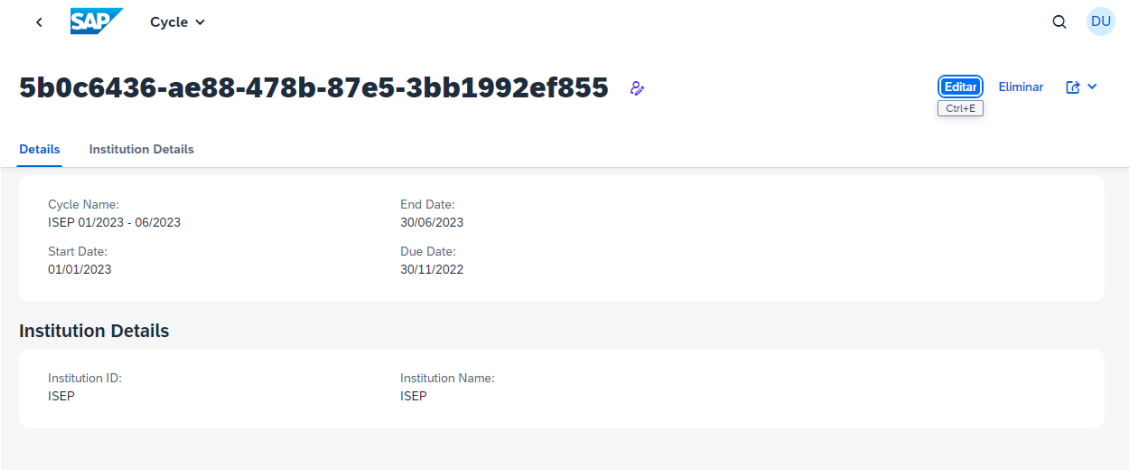


Figure 6.10: Cycle details

On figure 6.10 it is presented the details of the cycle which was chosen. It is to note that there is an edit button on the top right which is available due to the *UpdateHidden : false* which is present on listing 6.15. The button *Delete* is also available.

Then, by clicking on the *Edit* functionality, it is possible to enter on the change mode, where the fields which are not annotated with the *![@Common.FieldControl] : #ReadOnly* annotation. This annotation allows to make fields read only, this is used to inactivate the change on fields which are derived by expansions and navigation properties.

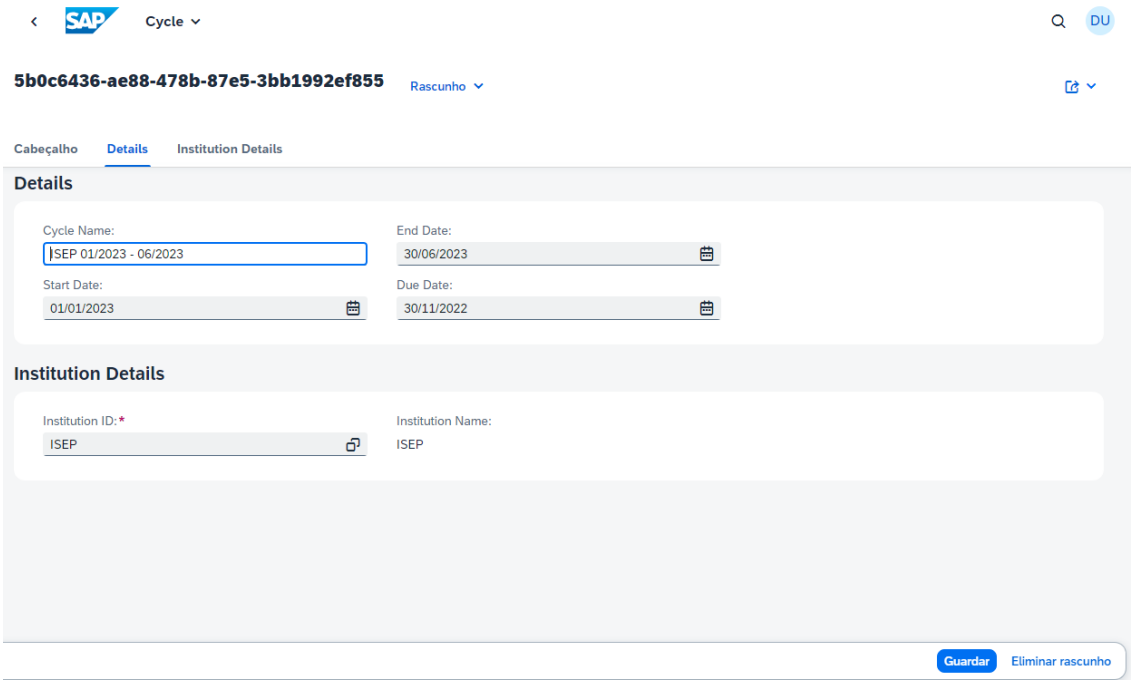


Figure 6.11: Cycle edit screen

On figure 6.12 it is possible to see the screen of the cycle edition. It is also possible to check that the Institution Name field showed in the UI is not editable since it is derived from the Institution Selected on the Institution ID field. On the bottom right corner it is possible to see the save button and the delete draft, which is enabled by the draft annotation.

In this case, the UI automatically identified the date fields and allowed to use the calendars for date insertion.

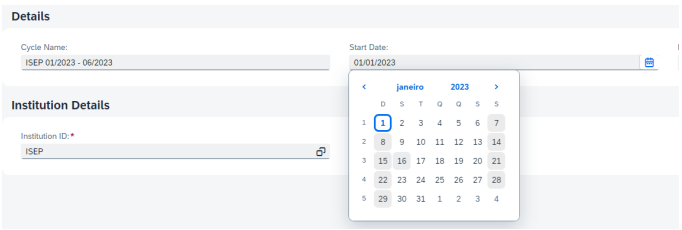


Figure 6.12: Date calendar

The illustration 6.12 shows the calendar display for data selection which is displayed by default.

Regarding the *Institution ID*, it is possible to check that there is a value help available so that the user can chose the right institution.

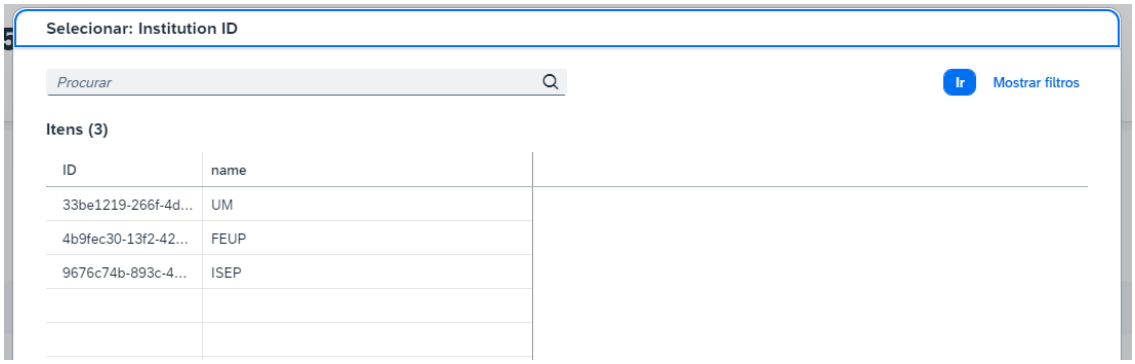


Figure 6.13: Value help selection example for institution

Figure 6.13 represents the value selection for institutions, which are enabled by the code snippet 6.21. It is also possible to verify that there are filters available for institution search.

Regarding the project entity, it is possible to verify that there are two services for the same entity which have two different names. The ProjectUser will be used by the normal users of the system. This entity will not have all the information available for editing. The Project entity will be accessible by the HR administrators and the team leaders which will have the capabilities of assigning projects to candidates, approve projects as well as assigning cycles to projects.

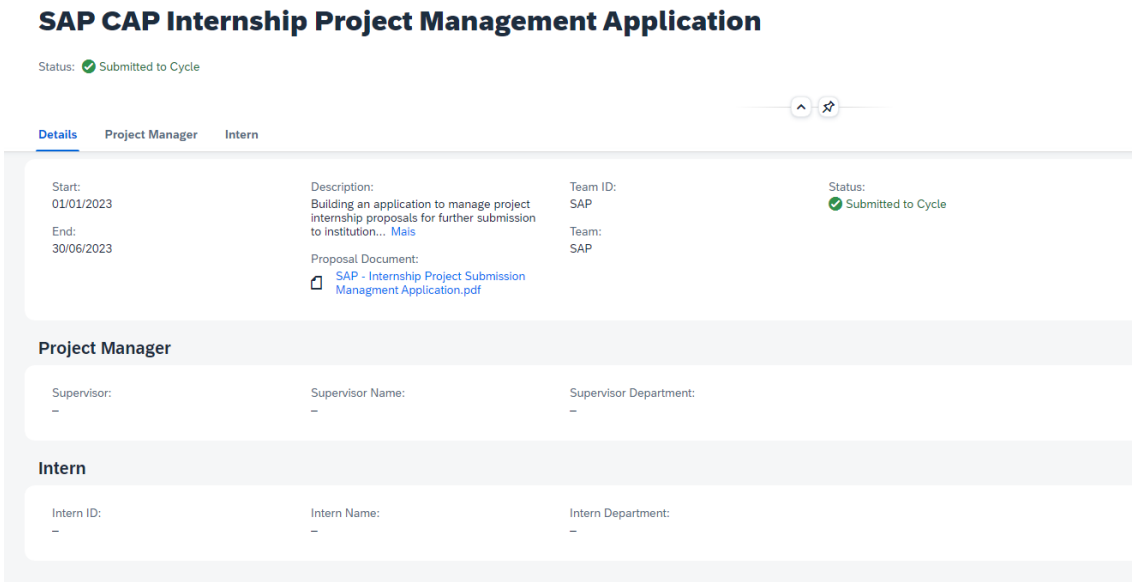


Figure 6.14: Project user entity details

Figure 6.14 shows the user interface for the *ProjectUser* entity, on the details view. In this way it is to note that the fields regarding the Project Manager and Intern are not editable. There are also no fields regarding the cycles and candidates to the project.

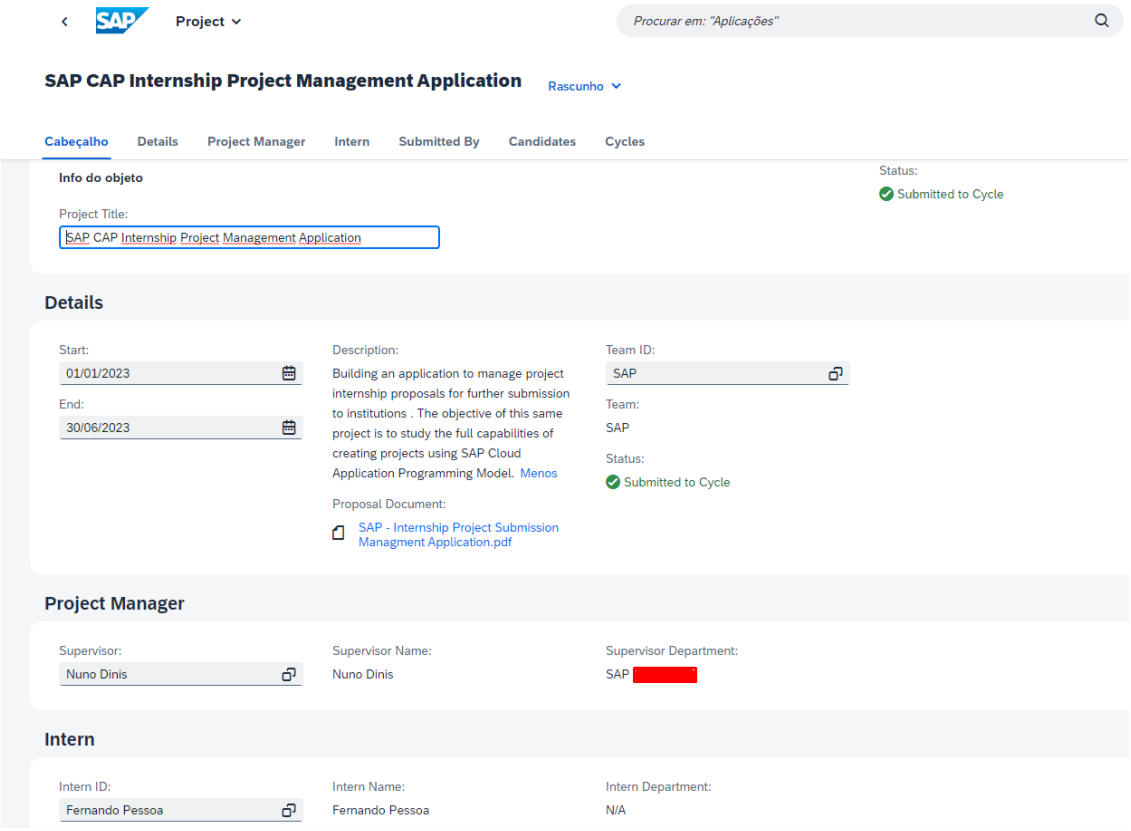


Figure 6.15: Project management entity on edit mode

Figure 6.15 represents the *Project* entity which is destined to the management of the project. In this case it is possible to see that the project manager and intern fields are editable.

Further down on the entity, it is possible to check the candidates and cycles which are related with this project. These are projections of the relationships stored on the *Cycle_To_Project and Candidates_To_Project* entities.

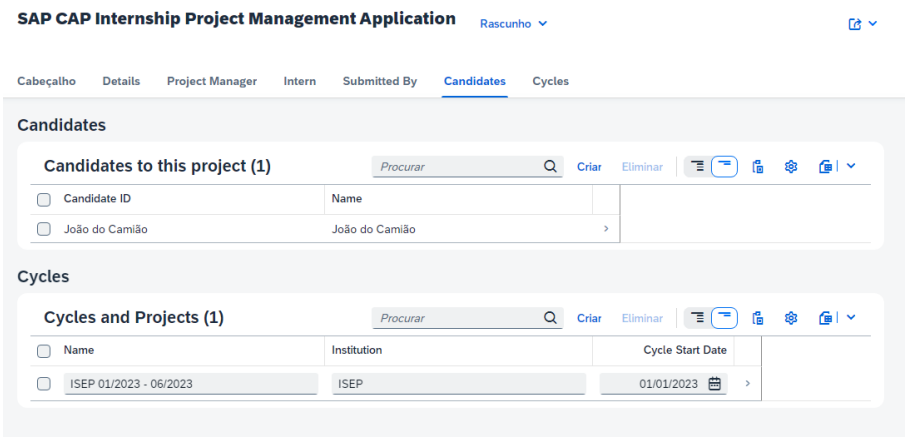


Figure 6.16: Project cycle and project association

On figure 6.16 it is possible to check the data related with these entities embedded on the Project entity.

It is also on this entity which the custom action *acceptProject()* is going to be available.

```
{    $Type   : 'UI.DataFieldForAction',
     Action  : 'sfsf.projman.service.ProjectManager.acceptProject',
     Label   : 'Accept Project',
     ![@UI.Hidden]   : {$edmJson: {$Ne: [{$Path: 'status_ID'}, 1]}}
},
```

Listing 6.22: Accept project UI

For it to be active, it is necessary to implement the UI annotation which activates the button. On listing 6.22 it is possible to check the binding of this UI field button to the Action on the service. The annotation for the availability of the button has a condition using the *edmJson*. This is checking if the status of the project is Submitted to Manager Approval. Only in those cases, the button is available.

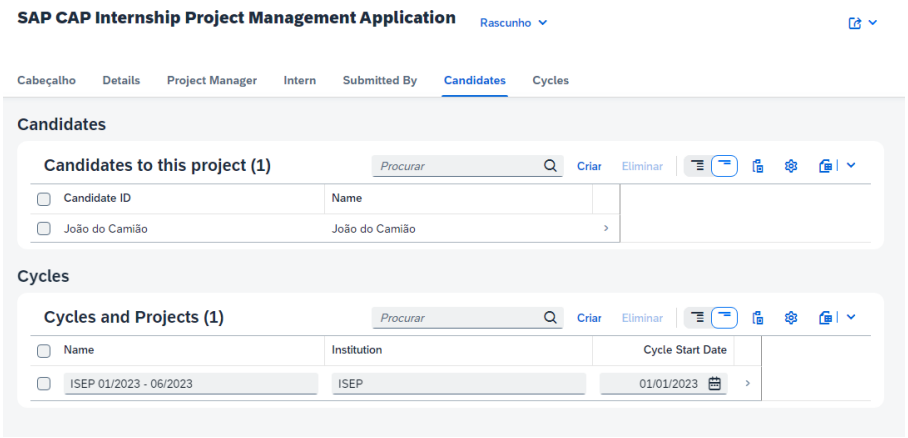


Figure 6.17: Accept Project action

On figure 6.17 it is possible to check the action button which is available in order to execute the approval action.

6.2.7 Security

From an authentication perspective, it is possible to use the *requires : 'authenticated-user'* in order to have the user authentication.

```
service ProjectManager @(requires : 'authenticated-user')
```

Listing 6.23: Authentication on service

Listing 6.23 obligates the user to log on to the system in order to be able to use it.

In this case, mocked authentication is being used, for further development purposes and simplicity. This is configured on the package.json file, like it is possible to check on listing.

```
"auth": {
        "[development]": {
          "strategy": "mock",
          "users": {
            "Nuno.Dinis": {
              "roles": [
                "Manager"
              ]
            },
          }
        }
      },
```

Listing 6.24: Authentication mock on package.json

Further authentication will be configured on SAP BTP by the usage of roles, role collections and users.

6.3 Future Work

One of the functionalities which would increase the value of this work is the implementation of a type of notifications so that team leaders and HR Managers can have reminders of up until when the submission of internship projects needs to be done.

In this case, due to both time constraints and the iterative nature of the application, being that there are still improvements to be done at the data type and model, as well as other new functionalities may arise, it was not possible to implement this functionality.

Despite not being implemented, the author has the knowledge on how this will be implemented.

6.3.1 SAP Cloud Platform Integration

The usage of the SAP CPI integration flows is the main alternative to implement this use case. Integration flows contain the necessary steps to extract the data, transform it and send it to destination systems. In this case it would be used to send email notifications.

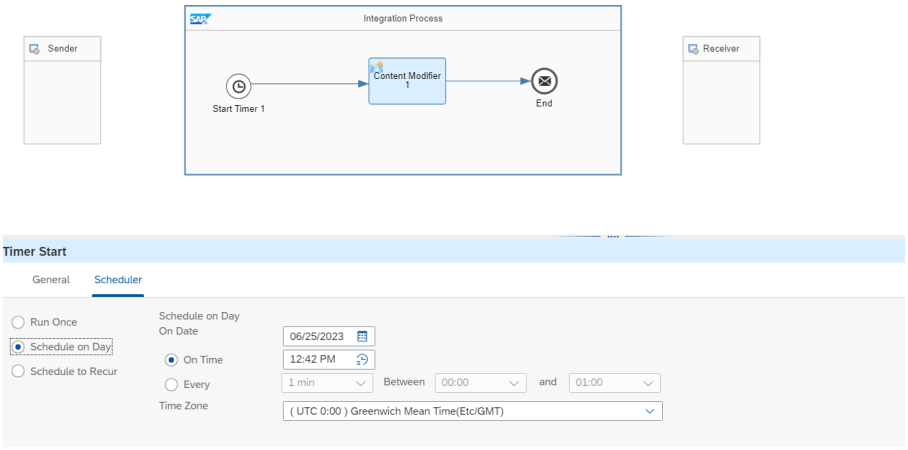


Figure 6.18: SAP CPI timer configuration

For running automatically, it would be needed to use an initial timer, like shown in figure 6.18.

Then it would be needed to fetch the data regarding the Cycle's due date, which is the date until when projects can be submitted. For this, it would be used the OData service which was developed in SAP CAP, which can be used alongside the UI implemented.

For this, an request-reply node would be used with an OData connector, like in figure 6.19.

After that, it would be needed to fetch the email address of the team leaders, in this case it would be possible also by using either the SAP CAP service developer or fetch the data directly using the SuccessFactors OData API.

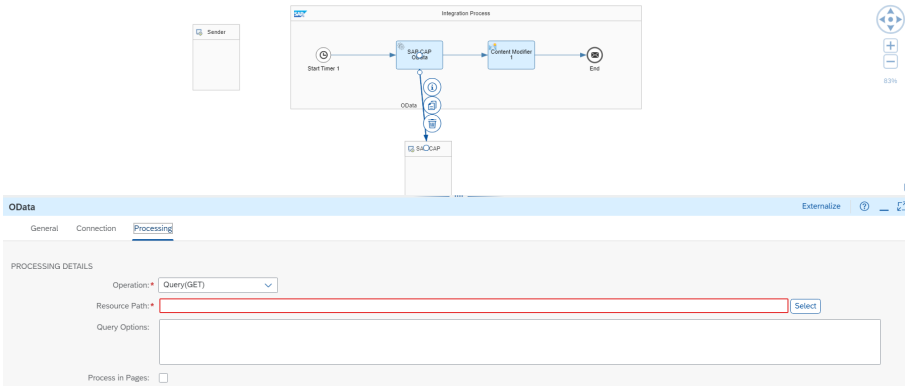


Figure 6.19: SAP CPI OData call

Then, it is possible to configure an email call, using also a request-reply but this time, using the Mail connection offered by SAP CPI, like it is exemplified on figure 6.20.

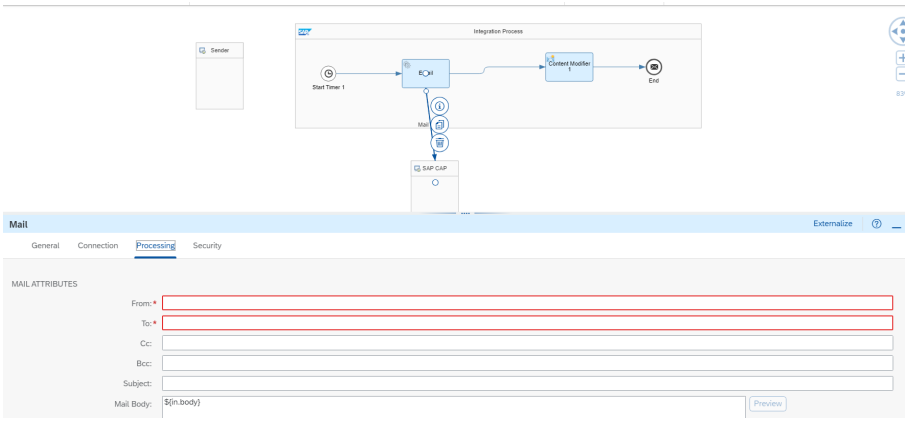


Figure 6.20: SAP CPI Email call

For this it is necessary to configure a sending email, a to email, which would be provided by the OData call to SAP SuccessFactors or the SAP CAP OData service. The body is configurable to include custom text and send the message with the dates which could be brought by the cycles. Other cycle information could be inserted.

The main logic to identify if the email should be sent or not would be the distance from the due data. Per example, if the due data is 30 days from the time the integration is running, an email notification should be sent.

Chapter 7

Evaluation and result analysis

During the current chapter, it will be presented how the evaluation of results and its analysis will be done. It will detail how the testing of the application will occur and what should be evaluated during the result analysis. It will also present what aspects are going to be evaluated and how they are going to be analyzed.

7.1 Result Analysis Indicators

The main objective of this project is to have less human related errors during the internship proposal process on the company as well as optimizing the productivity of the organization's Human Resources (HR).

The perceived increase of the productivity and reduced time spent by the internship management proposal app as a whole is one important factor for evaluation and final conclusions of the project's success.

Apart from this objective, it is also important to test the user's authorizations throughout the application, so that only the authorized users have access to see all the projects and only the team leaders, per example, are able to approve project proposals.

Finally, there should also be an evaluation of the end user's satisfaction, given the objectives previously mentioned.

7.2 Methodology

The developed project will be submitted to both User acceptance/end-to-end and integration testing in a way that each functionality is tested from end to end, from the Human Resource Information System data until the projects which are submitted by the employees.

From a data perspective, there is a need to certify its accuracy. This way, the data will be further analyzed on the database to guarantee its integrity.

7.2.1 Application Testing

From an application testing perspective, there are some scenarios which are due to be tested, and they should be previously scripted:

- For authorization purposes, different types of users will try to perform different types of authorizations, and test if it's not possible to do operations which they are not supposed to.

- There should be end to end testing for all of the application's entities.
- For data querying capabilities, the search of internship projects obeying to different filter criteria should be considered.

Unit Testing

Due to low coding and time purposes, the author disregarded the need of performing unit testing, since there are few lines of code on the java script custom handler functions.

Security Testing

Although security is part of the non-functional requirements, it had some dedicated testing since it is such an important aspect of the implementation.

For security testing purposes, there are a few scenarios which are to be tested. These tests have the objective of certifying that the correct user management is being made, and that the users have access to only what is required giving their role.

The tests which were performed are:

1. The user should be authenticated in order to use the application, so a login screen should appear - *successfully tested*
2. The basic user should not have the access to perform the management of internship proposals, thus not being able to assign supervisors and change the internship proposal status. In this case the service should return a 502 code - *successfully tested*.
3. The normal user and the team leader should not have access to Create, Read, Update and Delete (CRUD) the cycles, teams, candidates and institutions - *successfully tested*.
4. The system should allow the HR department user to CRUD the cycles, teams, candidates and institutions - *successfully tested*.
5. Both the team manager and the HR department should be able to read all the internship proposals - *successfully tested*.

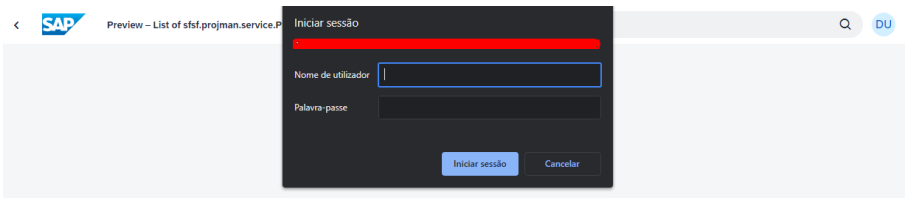


Figure 7.1: Login request for application

Figure 7.1 represents the login in screen which is requested on browser. If the user doesn't login he doesn't have access to the application.

Regarding the test scenario number 2, it can be shown on the figure 7.2 below as successfully tested.

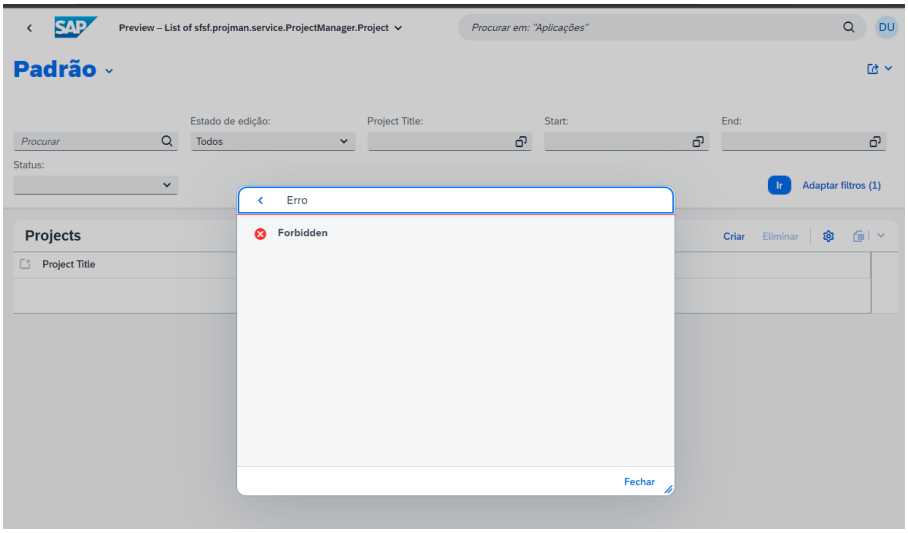


Figure 7.2: Missing authorization to project management screen

Integration and End to End testing

From an integration perspective, it will be tested during the end to end testing scenarios. The specific tests which consider integration testing are the correct saving of data on the database as well as it correct representation from the database, as well as the data being correctly pulled from SuccessFactors using the SuccessFactors OData API for the specified entities.

The following tests were performed from end to end scenarios, as well as the correct maintenance on the database was checked:

1. The basic user can successfully create a project on the database - *successfully tested*.
2. The basic user can successfully update and see his own projects on the database - *successfully tested*.
3. The HR Department user has the capability to assign interns - *successfully tested*.
4. The interns are employees of the company and are present in SuccessFactors - *successfully tested*.
5. The HR department and the team leader can assign the internship supervisor and the change is correctly reflected on the database - *successfully tested*.
6. The HR department and the team leader can assign an intern to the project - *successfully tested*.
7. The HR department can perform the CRUD operations on candidates, cycles, teams and institutions - *successfully tested*.
8. The team manager can accept projects and the project status is successfully updated on the database - *successfully tested*.

In this case, it is possible to say that, since all the use cases were covered by this testing, the system testing was successfully performed.

By analyzing the data which is on the system after creation of projects and assignment of interns, supervisor, candidates and cycles, it is possible to say that the data reliability mentioned on Chapter 4 was successfully delivered.

Since the system is a custom made system, it is possible to say that it is currently scalable, and it is compatible with wider and smaller screens accordingly as well as with mobile screens.

Testing the Create project functionality

For further exemplification, an user creating a project will be tested during this section.

Initially, the user selects the creating project option and is presented with to a form:

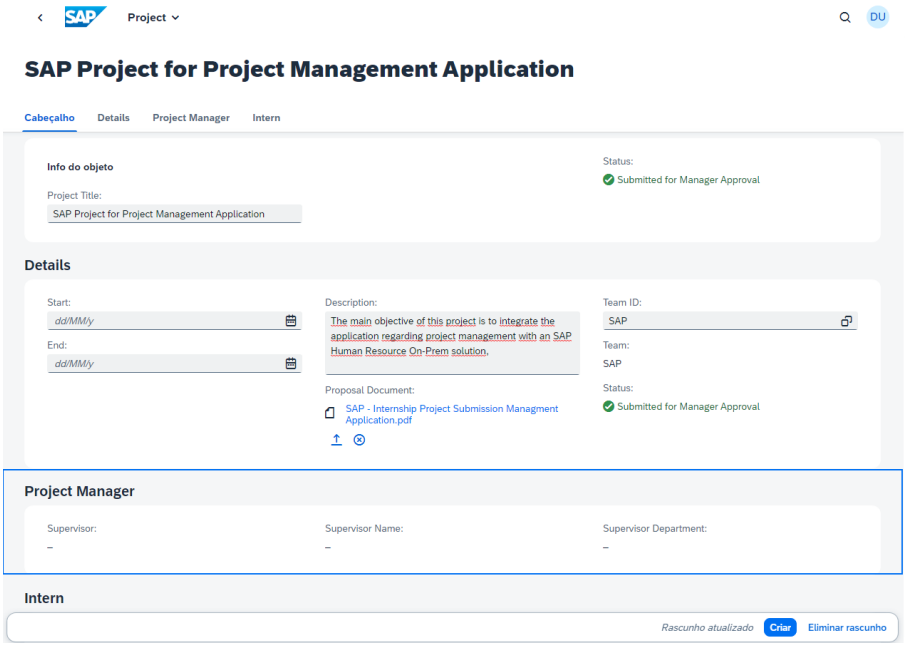


Figure 7.3: Project creation form

The figure 7.3 represents the form presented to the user for further project creation.

If the user clicks on save, the data is maintained correctly on the database and presented to the user after being saved.

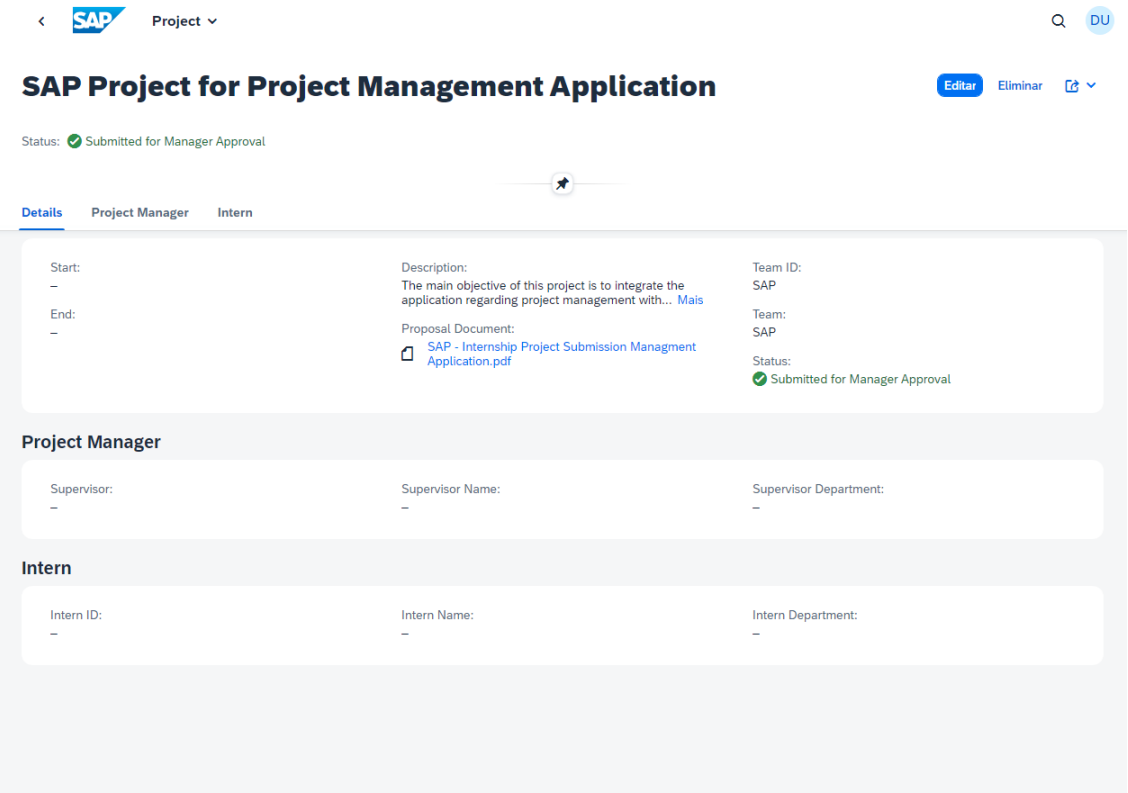


Figure 7.4: Project creation form

Figure 7.4 demonstrates the data on read mode being presented to the user after being saved on the database.

The next step, consists on a different user, in this case the team leader, being presented with the multiple projects created and selecting the recently created record by the normal user.

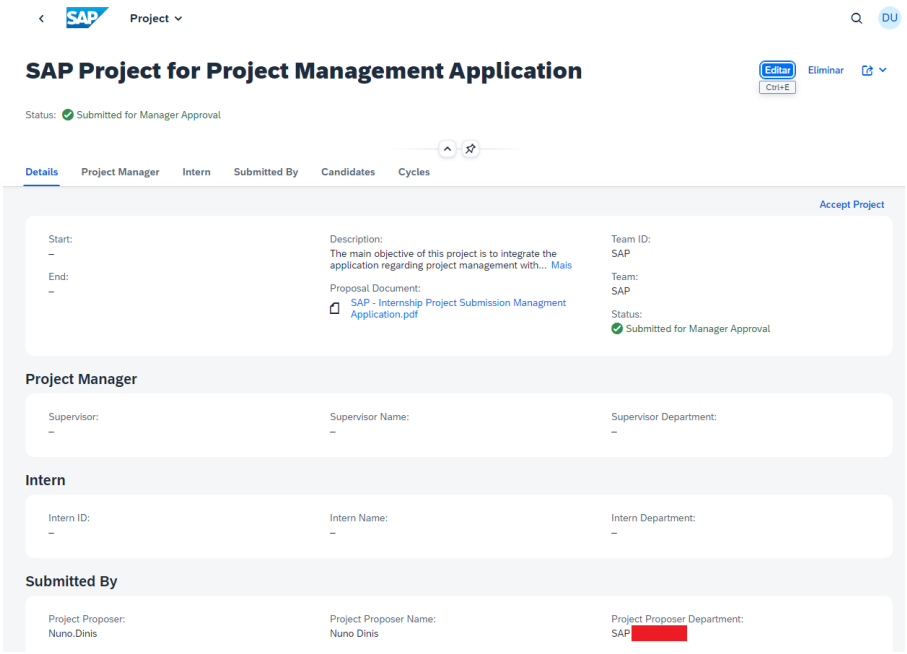


Figure 7.5: Project created is shown to the manager accordingly

On figure 7.5 the project is presented with information about the project and has the accept option.

The next step would be the acceptance of the project by the manager, in this case the manager clicked on the accept project button.

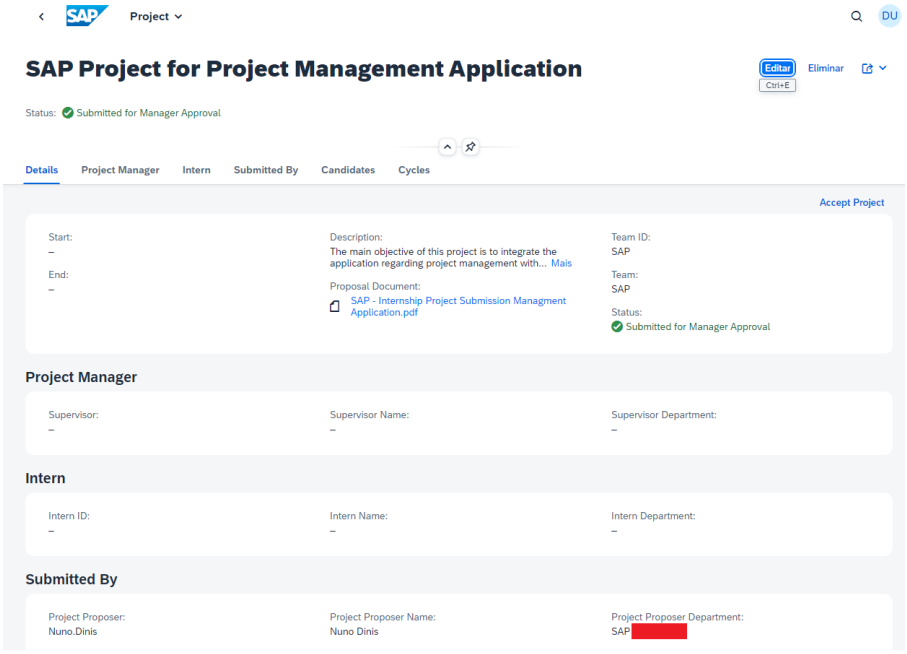


Figure 7.6: Manager accepted the project

As seen in figure 7.6, the status is set to Approved by Manager.

Now the HR Department can submit the project to a certain cycle. For this the person can assign cycles on the project like it is shown below:

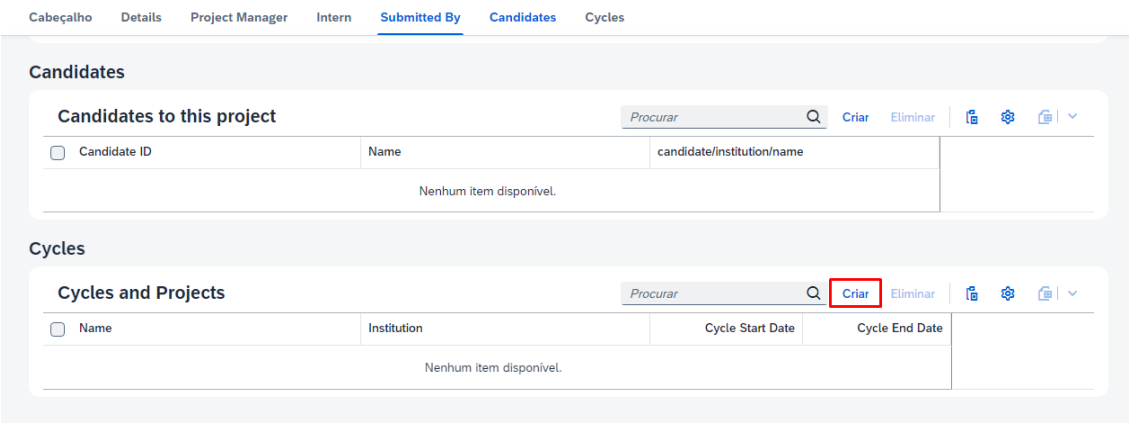


Figure 7.7: HR Manager decides to assign the project to a cycle

Figure 7.7 shows the HR manager capability of creating the new cycle inside a project, thus creating a new association between project and cycle.

Then the user needs to select the cycle and click on save.

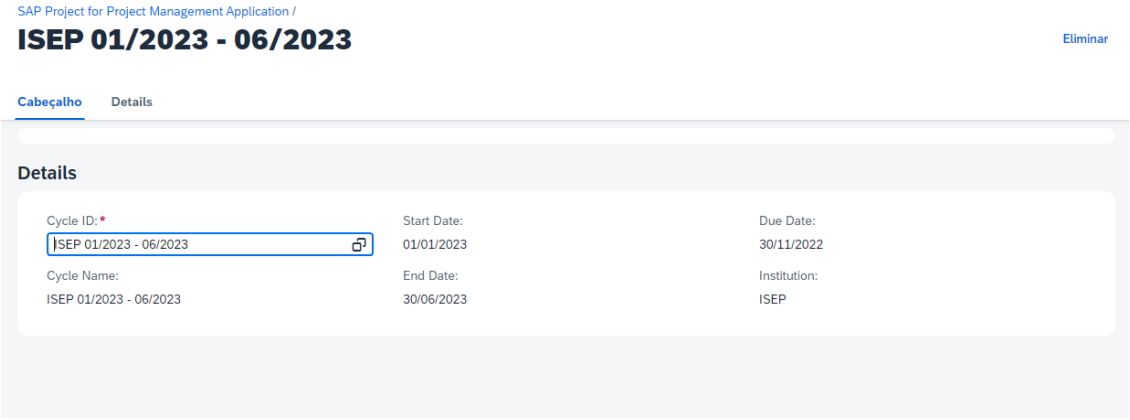


Figure 7.8: The HR manager selected the specific cycle

After the step illustrated on figure 7.8 it is possible to see the cycle on the project.

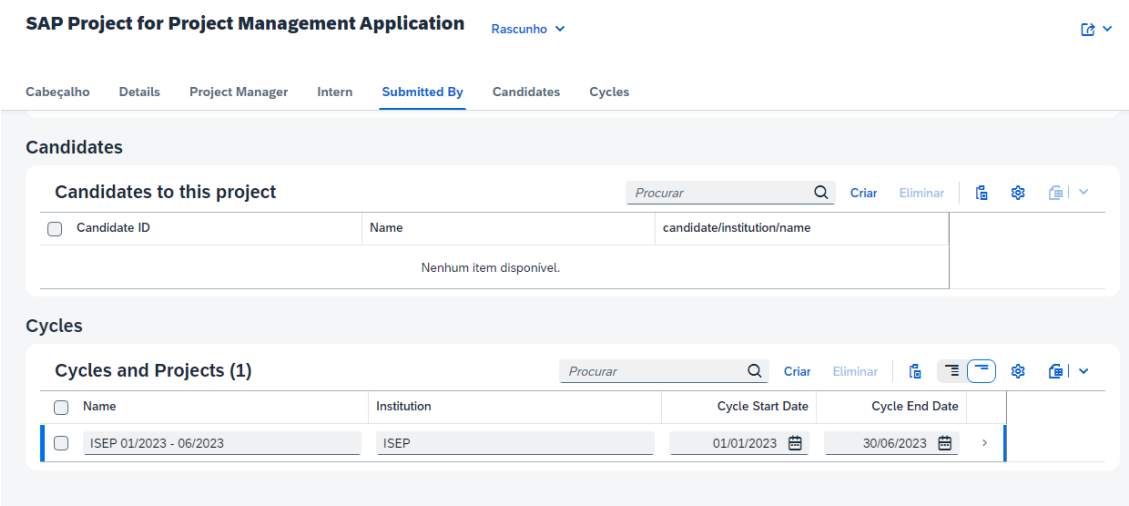


Figure 7.9: The cycle is presented on the project

On figure 7.9 it is possible to check the created cycle inside a given project.

The next step, would be to assign candidates to a project, and then block the project to a candidate when a certain candidate is interviewed and is further on boarded to the company.

After this step, the team leader can assign the project intern and team manager.

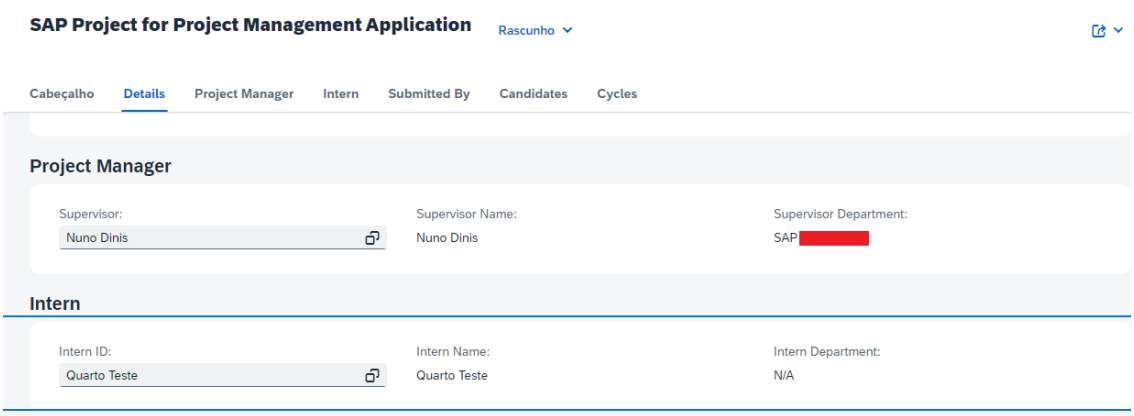


Figure 7.10: Project with intern and supervisor information

The figure 7.10 illustrates the capability of the information contained on Success Factors being presented to the user on the cycle entity when the project supervisor and intern are assigned.

Regarding the data filtering capabilities, it was also successfully implemented. All the entries are filterable

This demonstration of the usability of the system allows to ensure that the implementation was successfully done with which was pretended on a higher level.

7.2.2 System Usability Scale

Brooke (1996) created the System Usability Scale (SUS) as a "quick and dirty" survey scale that would allow a usability practitioner to quickly and simply assess the usability of a given product or service [85].

This was the survey utilized once it is compatible with a wide range of applications and technologies. Aside from that, both research participants and administrators did find the survey to be relatively short and simple to utilize. Third, the survey delivers a single score on a scale that is easily comprehended by a wide range of persons (from project managers to computer programmers) who are typically involved in product and service development and may have little or no experience with human factors and usability [85].

The survey was composed by the standard SUS questions:

1. I think that I would like to use this system frequently
2. I found the system unnecessarily complex
3. I thought the system was easy to use
4. I think that I would need the support of a technical person to be able to use this system
5. I found that the various functions in this system were well integrated
6. I thought that there was too much inconsistency in this system
7. I would imaging that most people would learn to use this system very quickly
8. I found the system very cumbersome to use
9. I felt very confident using the system
10. I needed to learn a lot of things before I could get going with this system

The answers for the surveys' affirmations will be the following, considering the numeration its value:

1. I strongly disagree
2. I disagree
3. I don't agree and I don't disagree
4. I agree
5. I strongly agree

The results of these questionnaires will determine the success of the application from an usability perspective.

Questionnaire results

The questionnaire was answered by 4 persons, two normal users, one team leader and one HR Manager, having the whole spectrum of types of users of the system. It is to note that the questions 1,3,5,7 and 9 have better results when the answers are closer to 5. Questions 2, 4, 6, 8 and 10 have better results when closer to 1.

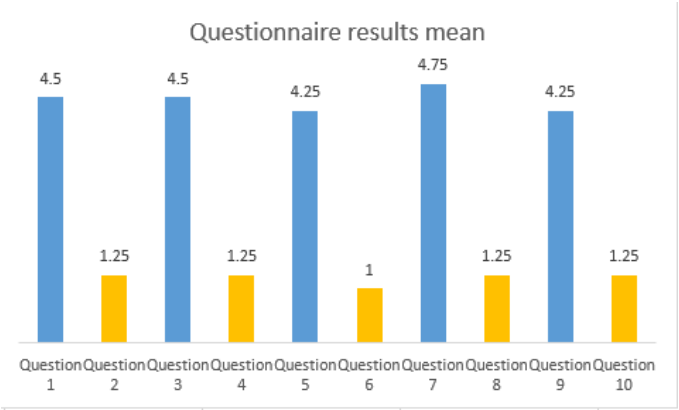


Figure 7.11: Questionnaire answers results' mean

The figure 7.11 has the mean of the answers of the questionnaire. The blue bars represent the even questions, the orange bars represent the odd questions.

In this case, for calculation of the questionnaires' result there are two steps to do first:

- Even questions: regarding the odd questions, their value should be subtracted to 5.
- Odd questions: the results of odd questions should have 1 subtracted.

Even values result:

$$(4.5 - 1) + (4.5 - 1) + (4.25 - 1) + (4.75 - 1) + (4.25 - 1) = 17.25$$

Odd values result:

$$(5 - 1.25) + (5 - 1.25) + (5 - 1) + (5 - 1.25) + (5 - 1.25) = 19$$

Then, add the values and multiply by 2.5.

$$Result = (17.25 + 19) * 2.5 = 90.625$$

The obtained result is then 90.625 which is an excellent result of the survey. In this case it is also important to mention that the number of participants is very low, this is due to the embryonary stage of the development. More users will test the system in the future, when the application is on a more advanced state. It is also to note that, although the low number of users, it covered all type of users.

There were additional comments added by the users:

- *'This application will be very useful for daily activities within the company, with regard to internship projects. It will undoubtedly centralize and streamline all interactions within the team.'*
- *'The usability is fine. Additional information will be helpful.'*
- *'Forms are intuitive, easy to use and elements/concepts listed are self-explanatory.'*

7.2.3 Non Functional Requirements Results

For the non functional requirements, the results are positive. As previously mentioned, the usability and ease of use was evaluated by the usability survey, which had very positive results. The reliability of data is also good, being this supported by the data being well stored on the database and well integrated with the data which is present in Success Factors, presenting consistent data, meaning that the data represents what is on the system.

The performance requirements were not raised by the organization for this work, it is not possible to yet analyze the impacts of the implementation in regards to the time efficiency of the application due to its early phase, since little data is yet being used and the application is not fully deployed on the cloud nor its final environment.

Just like performance, supportability was not raised as a need for the application, but the SAP CAP framework already applies the User Interface flexibility to use different screen sizes as well as cell phones devices.

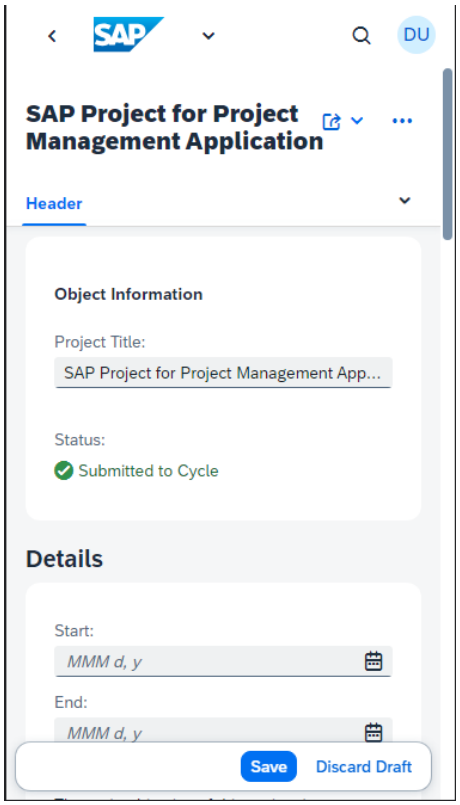


Figure 7.12: Mobile phone view of the application

Figure 7.12 shows the implementation being tested as part of a simulation of a smartphone.

Finally, for the design constraints, the data is now centralized on a unique database and can be accessed through an unique access point, dropping the manual interventions and increasing automation.

7.3 Evaluation Results

After the analysis of the work done through this chapter, it is possible to say that the implementation was successful on its main requirements. The usability is excellent considering the User Testing and the functionalities are working fine both from security perspective as well as data integrity. The user feedback was also very good, overall, stating that the usability of the system is within the acceptable standards.

The data is now centralized on a unique database, enhanced with the real time data from the main Human Resource Information System from the company, thus in agreement with the integrity of data and its centralization.

It is possible to say that both functional and non-functional requirements were successfuly implemented.

Chapter 8

Conclusion

Having the work for this project finished, this final chapter has the main objective of mentioning the author considerations as well as the objectives which were reached. Besides this, it also will have the summary of the limitations of the system which will be resolved by future work.

8.1 Objectives reached

The great majority of objectives listed in section 1.3 were successfully reached. The main goal of this work was to present a solution which could help the organization centralizing and organizing the data relative to internship projects. This objective was reached successfully. The solution implemented was already evaluated by some key users and the perceived increase of productivity is recognized. In comparison, having a solution for the intended purpose of submitting project proposals and accepting them on the same application, is more centralized and direct than having a manual process of maintaining the information manually in files. There are also steps on this workflow which avoid the need of trading emails for approvals and corrections on the proposals.

The information kept by the solution now allows the users to better know the deadlines of submission proposal cycles and to avoid missing opportunities to send internship proposals accordingly. With this the solution allows the reduction of human induced errors and problems introduced by data at rest as well as data in motion.

The development of a solution for resolving the issues with the already existent process was successfully achieved, since the main users were completely satisfied with the solution proposed.

Since the process shifted from an manual emailing process in conjunction with manual data recording on excel spreadsheets, it is safe to say that the solution implemented the automation of the process successfully.

Regarding the user interface, it is intuitive and easy to use, and the authorization and safety features were successfully implemented, as it was possible to verify during the chapter 7. This was also one of the initial important requirements, so that the users do not have to spend extra time on learning how to use the platform and it does not get confusing in order to introduce more issues.

The data is not accessible throughout the HRIS system of the company since this requirement was dropped further on the developments due to time constraints. It will be considered on the next section as part of the future work, alongside with the notifications.

8.2 Limitations

There are a few limitations of this work, which are going to be further explained during this section.

The integration with the Human Resource Information System of the organization is lacking a notification mechanism which would inform the users of the system about the status of projects. Since the acceptance of projects by the team manager currently does not trigger any notification to the person which submitted the internship proposal this could cause further delays on the workflow. This limitation is actually for all the status of the project, since the user which needs to take action next considering the status is not notified.

Another limitation is the lack of a reporting tool. Although the application has filtering of information, there is no extraction of reports to files, to further analyze the projects, per example. This extraction could include information relative to how many cycles did the project went through and what are the projects with more candidates. This could lead to further analysis of data using data mining or text mining capabilities so that the organization would know which area of projects is more appealing to candidates.

Finally, the capability of taking actions on the HRIS of the organization is also lacking on the implementation, since the users need to use the application in order to do so.

8.3 Future work

There are a few functionalities which need to be thought on future work. The further implementation of the notification system on the SAP Cloud Platform Integration is one of the future objectives of this development. This would be one of the initial objectives. The notification to the HRIS system is also one of the capabilities is going to be considered.

Alongside this, there is also the need of implementing the SAP Launchpad, since it will be the portal where the Fiori applications of the organization would be centralized.

Future work also includes improvements on the data model, like per example adding the different sub areas on the institutions for different courses. It would also imply the deployment of the application to the cloud to be widely accessible to all users. For data security and General Data Protection Regulation, there should be a mechanism to purge data which is older than the established time by the regulation, specially for candidates information.

The limitations which were considered during section 8.2 are also being considered to be addressed in the future. The integration with artificial intelligence is one of the future objectives of the tool in order to further automate the process and to take advantage of further reporting capabilities.

Finally, a presentation for the whole organization would need to be prepared in order to give an overview and knowledge transfer to the future users.

8.4 Author Comments

Although having a few years of experience in the SAP area, the author never dominated the SAP Cloud Application Programming Model since it is a relatively recent framework. The main purpose of SAP CAP is developing either model based OData services or end-to-end applications with a SAP standardized user interface's look and feel. The author has

considered a good experience developing and applying concepts of web development on a different paradigm when developing this application using SAP CAP.

From a theoretical perspective, the author is satisfied with the results reached, once it was possible to analyze different protocols and development standards across all the scope of end-to-end applications.

The technologies used to develop the project were both exciting and interesting for the experience of the author. The multiple possibilities of development using CAP gives the author the confidence that this framework will be highly present on the future of custom made SAP developments and extensions.

Bibliography

[1] M Anupa. "Role of Human Resources Information System (Hris) in Accelerating Organizational Effectiveness – It Companies Perspective". In: *International Journal of Management and Humanities* 5.6 (2021), pp. 22–25. issn: 2394-0913. doi: 10.35940/ijmh.f1245.025621.

[2] Ali Albouti and K D Balaji. "IS HRIS PLAYING A ROLE IN HRM TODAY?" In: *International Journal of Research Publication and Reviews Journal homepage: www.ijrpr.com* 3.7 (2022), pp. 2964–2968. url: www.ijrpr.com.

[3] H Ziya Ozek. "The Eurasia Proceedings of Educational Social Sciences (EPESS) The Eurasia Proceedings of Educational Impact of Internship Programme in Engineering Education". In: *Social Sciences (EPESS)* 9 (2018). issn: 2587-1730. url: www.isres.org.

[4] Jan vom Brocke, Alan Hevner, and Alexander Maedche. "Introduction to Design Science Research". In: Sept. 2020, pp. 1–13. isbn: 978-3-030-46780-7. doi: 10.1007/978-3-030-46781-4_1.

[5] Priyanka Wandhe. "A Role of Effectiveness of Human Resource Information System (HRIS) in 21st Century". In: *SSRN Electronic Journal* (2020). doi: 10.2139/ssrn.3718247. url: https://ssrn.com/abstract=3718247.

[6] SAP AG. "Organizational Management". In: (). url: https://help.sap.com/doc/saphelp_nw73ehp1/7.31.19/en-US/4e/ebb6fb0d1857dee10000000a421bc1/content.htm?no_cache=true.

[7] SAP. *What is a talent management system (TMS)? | SAP Insights*. url: https://www.sap.com/insights/what-is-a-talent-management-system-tms.html (visited on 01/28/2023).

[8] Rand Aldmour et al. "The Practice of HRIS Applications in Business Organizations in Jordan: An Empirical Study". In: Sept. 2015.

[9] Gerardine Desanctis. "Human Resource Information Systems: A Current Assessment." In: *MIS Quarterly* 10.1 (1986), pp. 15–27. issn: 0276-7783. doi: 10.2307/248875. url: https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=4679481&site=eds-live.

[10] Veronica. *An Evaluation of The Employee Self-Service Feature of SAP System with Technology Acceptance Model (TAM): A Case Study*. 2020. doi: 10.1109/ICIMTech50083.2020.9211291. url: https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.9211291&site=eds-live.

[11] Hassan Rangriz, Javad Mehrabi, and Alireza Azadegan. "The Impact of Human Resource Information System on Strategic Decisions in Iran". In: *Computer and Information Science* 4.2 (2011). url: www.ccsenet.org/cis.

[12] Indrit Troshani, Cate Jerram, and Sally Rao Hill. "Exploring the public sector adoption of HRIS". In: *Industrial Management Data Systems* 111.3 (Jan. 2011), pp. 470–488. issn: 0263-5577. doi: 10.1108/02635571111118314. url: https://doi.org/10.1108/02635571111118314.

[13] Kenneth A. Kovach et al. "Administrative and Strategic Advantages of HRIS". In: *Employment Relations Today* 29.2 (June 2002), pp. 43–48. issn: 1520-6459. doi: 10.1002/ERT.10039. url: https://onlinelibrary.wiley.com/doi/full/10.1002/ert.10039%20https://onlinelibrary.wiley.com/doi/abs/10.1002/ert.10039%20https://onlinelibrary.wiley.com/doi/10.1002/ert.10039.

[14] Fida Afiouni. "Human capital management: A new name for HRM?" In: *Int. J. of Learning and Intellectual Capital* 10 (Jan. 2013), pp. 18–34. doi: 10.1504/IJLIC.2013.052081.

[15] Richard Finn. "Human-capital management (HCM): A three-letter acronym too far for HR or the way to get the people agenda on to that of the board?" English. In: *Human Resource Management International Digest* 11.5 (2003). Copyright - Copyright MCB UP Limited (MCB) 2003, pp. 2–4. url: https://www.proquest.com/scholarly-journals/human-capital-management-hcm-three-letter-acronym/docview/214892504/se-2.

[16] James C. Hayton. "Strategic human capital management in SMEs: An empirical study of entrepreneurial performance". In: *Human Resource Management* 42.4 (Dec. 2003), pp. 375–391. issn: 1099-050X. doi: 10.1002/HRM.10096. url: https://onlinelibrary.wiley.com/doi/full/10.1002/hrm.10096%20https://onlinelibrary.wiley.com/doi/abs/10.1002/hrm.10096%20https://onlinelibrary.wiley.com/doi/10.1002/hrm.10096.

[17] Jeanne Carsten. "Play to Your Strengths: Managing Your Internal Labor Markets for Lasting Competitive Advantage". English. In: *Personnel Psychology* 57.3 (Autumn 2004). Copyright - Copyright Personnel Psychology, Inc. Autumn 2004. url: https://www.proquest.com/scholarly-journals/play-your-strengths-managing-internal-labor/docview/220141968/se-2.

[18] M. Armstrong and Angela Baron. "Human Capital Management: Achieving Added Value Through People". In: 2007.

[19] Edward Lawler. "Make Human Capital A Source of Competitive Advantage". In: *Organizational Dynamics* 38 (Oct. 2008). doi: 10.2139/ssrn.1311431.

[20] SAP. *Human Capital Management (HCM) | Cloud HR Software Systems | SAP*. url: https://www.sap.com/products/hcm.html (visited on 01/29/2023).

[21] Pedro Hernandez. *Why Microsoft Picked SAP SuccessFactors for Its HCM Needs.* 2016. url: https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=119655373&site=eds-live.

[22] SAP, Andreea Bejenaru, and Clarissa Anjani. "Implementation Design Principle SAP SuccessFactors Integration: Integration Center and Cloud Integration". In: (2020).

[23] Sai Manikanth. *Workday HCM Modules | Human Capital Management Modules*. url: https://hkrtrainings.com/workday-hcm-modules (visited on 02/02/2023).

[24] Workday. "Workday Human Capital Management Suite". In: (2014).

[25] But Workday and With Workday. "Workday Integration Cloud Connectors for Human Capital Management". In: (2018). url: https://www.workday.com/content/dam/web/hk/documents/datasheets/workday-integration-cloud-connectors-hcml-datasheet.pdf.

[26] Workday. "Workday Cloud Connect packages". In: (2016).

[27] Oracle. "Oracle Cloud HCM Solution Overview Brochure". In: (2022).

[28] Oracle. "Oracle Fusion Cloud HCM HCM Extracts 22D". In: (2011).

[29] Oracle. "Oracle Fusion Cloud HCM - HCM Integrations". In: (2011).

[30] Sage. "Sage People global HR and People management system". In: (2022).

[31] Sage. *Sate People API Get started*. 2021. url: https://developer.sage.com/api/people/get-started/ (visited on 02/05/2023).

[32] IBM. *What is an Application Programming Interface (API)? | IBM*. url: https://www.ibm.com/topics/api (visited on 02/06/2023).

[33] Ma-Keba Frye. *What is an API? (Application Programming Interface) | MuleSoft*. url: https://www.mulesoft.com/resources/api/what-is-an-api (visited on 02/06/2023).

[34] Red Hat. *O que é API? Guia de APIs para iniciantes*. 2019. url: https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces (visited on 02/06/2023).

[35] Roy Thomas Fielding. "REST: Architectural Styles and the Design of Network-based Software Architectures". Doctoral dissertation. University of California, Irvine, 2000. url: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

[36] Bruno Costa et al. "Evaluating a Representational State Transfer (REST) Architecture: What is the Impact of REST in My Architecture?" In: *2014 IEEE/IFIP Conference on Software Architecture*. Apr. 2014, pp. 105–114. doi: 10.1109/WICSA.2014.29.

[37] David Garlan and Mary Shaw. "An Introduction to Software Architecture". In: (1994).

[38] A. Fuggetta, G.P. Picco, and G. Vigna. "Understanding code mobility". In: *IEEE Transactions on Software Engineering* 24.5 (1998), pp. 342–361. doi: 10.1109/32.685258.

[39] Tim Berners-Lee et al. "Uniform Resource Identifiers (URI): Generic Syntax. Request for Comments: 2396". In: 2002.2002-10-11 (1998). url: http://www.ietf.org/rfc/rfc2396.txt.

[40] Robert Richards. *Pro PHP XML and Web Services*. 2006, pp. 633–672. isbn: 9781430201397. doi: 10.1007/978-1-4302-0139-7.

[41] IEC ISO. *ISO - ISO/IEC 20802-1:2016 - Information technology — Open data protocol (OData) v4.0 — Part 1: Core*. url: https://www.iso.org/standard/69208.html (visited on 02/12/2023).

[42] Chappell Associates. "INTRODUCING ODATA Sponsored by Microsoft Corporation". In: (2011).

[43] Don Box et al. *Simple Object Access Protocol (SOAP) 1.1*. May 2000. url: https://www.w3.org/TR/2000/NOTE-SOAP-20000508/ (visited on 02/12/2023).

[44] Virender Ranga and Anshu Soni. "API Features Individualizing of Web Services: REST and SOAP Hybrid Framework for FANets View project Intrusion Detection and Prevention Systems for IoT and CPS. View project". In: *Article in International Journal of Innovative Technology and Exploring Engineering* 8 (2019), pp. 2278–3075. doi: 10.35940/ijitee.I1107.0789S19. url: https://www.researchgate.net/publication/335419384.

[45] Matei Marian. "Emerging Markets Queries in Finance and Business iPaaS: Different Ways of Thinking-review under responsibility of the Emerging Markets Queries in Finance and Business local organization". In: *Procedia Economics and Finance* 3 (2012), pp. 1093–1098. issn: 2212-6716. doi: 10.1016/S2212-5671(12)00279-1. url: www.sciencedirect.com.

[46] "Feature Scope Description for SAP Integration Suite THE BEST RUN Content". In: (2023).

[47] Mulesoft. *Middleware technology for integration: Open source ESB*. url: https://www.mulesoft.com/resources/esb/integration-middleware-technology (visited on 05/06/2023).

[48] Mulesoft. url: https://www.mulesoft.com/platform/api/flex-api-gateway (visited on 05/06/2023).
[49] Mulesoft. url: https://www.mulesoft.com/platform/api/governance-anypoint (visited on 05/06/2023).
[50] Mulesoft. url: https://www.mulesoft.com/platform/api/manager (visited on 05/06/2023).
[51] *Anypoint Platform: A powerful and innovative feature set for APIs and integrations*. url: https://www.mulesoft.com/platform/anypoint-platform-features (visited on 05/06/2023).
[52] SAP. "SAP Portfolio and Project Management for SAP S/4HANA Master Guide for SAP Portfolio and Project Management 1.0 for SAP S/4HANA THE BEST RUN Content". In: (2018). url: http://help.sap.com..
[53] SAP. "Master Guide Portfolio and Project Management - Master Guide for SAP Portfolio and Project Management 6.1 ". In: (May 2017). url: http://help.sap.com..
[54] SAP. *SAP Enterprise Portfolio and Project Management Pricing*. url: https://www.sap.com/products/scm/project-portfolio-management/pricing.html (visited on 02/18/2023).
[55] SAP. *Developing with the SAP Cloud Application Programming Model | SAP Help Portal*. url: https://help.sap.com/docs/btp/sap-business-technology-platform/developing-with-sap-cloud-application-programming-model (visited on 02/12/2023).
[56] Timur Mirzoev and Craig Brockman. *SAP HANA and its performance benefits*. 2014. url: http://arxiv.org/abs/1404.2160.
[57] Yunsen Wang and Alexander Kogan. "Cloud-Based In-Memory Columnar Database Architecture for Continuous Audit Analytics." In: *Journal of Information Systems* 34.2 (2020), pp. 87–107. issn: 08887985. url: https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib&db=bth&AN=145694401&lang=pt-pt&site=eds-live&scope=site.
[58] SAP. *What is SAP HANA?* url: https://www.sap.com/products/technology-platform/hana/what-is-sap-hana.html (visited on 02/14/2023).
[59] Yeshwanth Valaboju. "A Study on SAP Fiori Apps and Fiori Design Principles". In: *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET) | An ISO* 9001 (2008). issn: 2319-8753. url: www.ijirset.com.
[60] Nishchol Mishra and Sanjay Silakari. "Predictive analytics: a survey, trends, applications, oppurtunities & challenges". In: *International Journal of Computer Science and Information Technologies* 3.3 (2012), pp. 4434–4438.
[61] Said A Salloum et al. "A survey of Arabic text mining". In: *Intelligent natural language processing: Trends and applications* (2018), pp. 417–431.
[62] Franz Färber et al. "SAP HANA Database-Data Management for Modern Business Applications". In: (2011).
[63] Add Belati and Firas Alomari. "An Overview of SAP Core Data Services". In: Feb. 2020.
[64] Hasso Plattner. "The Impact of Columnar In-Memory Databases on Enterprise Systems Implications of Eliminating Transaction-Maintained Aggregates". In: (2150).
[65] "Don't Try Coding ABAP Core Data Services Without Reading This First". In: (Mar. 2017). url: https://itpfed.com/dont-try-coding-abap-core-data-services-without-reading-this-first-2/.

[66] SAP. *capire - Providing Services*. url: https://cap.cloud.sap/docs/guides/providing-services#defining-services (visited on 02/14/2023).

[67] SAP. *capire - Serving Fiori UIs*. url: https://cap.cloud.sap/docs/advanced/fiori (visited on 02/14/2023).

[68] Peter A. Koen et al. "1 Fuzzy Front End : Effective Methods , Tools , and Techniques". In: 2002.

[69] Herstatt / Verworn et al. "The "FuzzyFront End" of Innovation The "Fuzzy Front End" of Innovation The "Fuzzy Front End" of Innovation". In: (2001). url: www.tu-harburg.de/tim.

[70] Simon Liu and Rick Kuhn. "Data Loss Prevention". In: *IT Professional* 12.2 (2010), pp. 10–13. doi: 10.1109/MITP.2010.52.

[71] Alex Osterwalder et al. "Value Proposition Design". In: (2014).

[72] Lai-Kow Chan and Ming-Lu Wu. "Quality function deployment: A literature review". In: *European Journal of Operational Research* 143.3 (2002), pp. 463–497. issn: 0377-2217. doi: https://doi.org/10.1016/S0377-2217(02)00178-9. url: https://www.sciencedirect.com/science/article/pii/S0377221702001789.

[73] *What is UML | Unified Modeling Language*. url: http://www.uml.org/what-is-uml.htm (visited on 02/21/2023).

[74] Martin Fowler et al. "Patterns Of Enterprise Application Architecture". In: (2002).

[75] Linda A. Macaulay. "Requirements Engineering". In: Applied Computing (1996). doi: 10.1007/978-1-4471-1005-7. url: http://link.springer.com/10.1007/978-1-4471-1005-7.

[76] *What is FURPS+? – Business Analyst Training in Hyderabad – COEPD*. url: https://businessanalysttraininghyderabad.wordpress.com/2014/08/05/what-is-furps/ (visited on 02/25/2023).

[77] Linda Westfall. "Software Requirements Engineering: What, Why, Who, When, and How". In: (2005).

[78] *UML Use Case Diagram Tutorial | Lucidchart*. url: https://www.lucidchart.com/pages/uml-use-case-diagram (visited on 02/22/2023).

[79] SAP. *SAP Business Technology Platform - Basic Concepts*. url: https://help.sap.com/docs/btp/sap-business-technology-platform/btp-basic-platform-concepts?locale=en-US (visited on 06/18/2023).

[80] SAP. *Business Application Studio*. url: https://www.sap.com/portugal/products/technology-platform/business-application-studio.html#technical-information (visited on 06/22/2023).

[81] SAP. *Dev spaces in SAP Business Application Studio*. url: https://help.sap.com/docs/bas/sap-business-application-%20studio/dev-spaces-in-sap-business-application-studio (visited on 06/22/2023).

[82] SAP. *Destinations*. url: https://help.sap.com/docs/SAP_BUILD_CLASSIC/acb8ee53fa0e448f9b4a4125591f5d5a/4be99e779fcb4ff9a8e5179b74bd72cc.html?locale=en-US (visited on 06/22/2023).

[83] SAP. *Cloud Foundry Environment*. url: https://help.sap.com/docs/btp/sap-business-technology-platform/cloud-foundry-environment (visited on 06/22/2023).

[84] SAP. *Using Services*. url: https://cap.cloud.sap/docs/guides/using-services (visited on 06/22/2023).

[85] Aaron Bangor, Philip T. Kortum, and James T. Miller. "An empirical evaluation of the system usability scale". In: *International Journal of Human-Computer Interaction* 24.6 (Aug. 2008), pp. 574–594. issn: 10447318. doi: 10.1080/10447310802205776.