INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA INFORMÁTICA

Desenvolvimento de uma aplicação para ajudar pessoas com transtornos de ansiedade

CLÁUDIO COELHO VASCONCELOS

Outubro de 2020

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Development of an application to help anxiety disorders

Cláudio Coelho Vasconcelos

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

Supervisor: António Constantino Martins

Porto, October 15, 2020

Dedicatory

"To my family and my girlfriend without whom I wouldn't be motivated and committed to this project. Thanks to my family for supporting me all throughout my studies. Thanks to my girlfriend for always being there for me. Thanks to my supervisor for providing great insight and useful advises to make this dissertation possible."

Abstract

This report documents the conception of the project Development of an application to help anxiety disorders. In this dissertation it is documented the analysis, design, development and evaluation of a mobile application and recommendation system that have the objective of helping people with anxiety.

In recent years, recommendation systems have grown in popularity and appreciation with users searching for ease and convenience when receiving predictions and recommendations based on their preferences and characteristics. With anxiety disorders affecting a considerable percentage of the population, tools for anxiety self-management can help people who aren't being accompanied by a therapist.

The mobile application will have a variety of exercises that will provide ways for users to relieve their anxiety and provide tools for learning how to self-manage anxiety. The mobile application will be connected to a recommendation system which will be able to suggest anxiety relieving exercises to users by taking into account exercise ratings and user characteristics.

To evaluate the general functionality of developed components tests were developed, with positive results. Additionally, the recommendation techniques were also tested to analyze the prediction error of each one. Overall the results were also positive with considerably low errors.

Keywords: Anxiety Disorder, Mental Health, Recommendation System, Microsoft Xamarin

Resumo

Este relatório documenta a concepção do projeto Desenvolvimento de uma aplicação para ajudar pessoas com transtornos de ansiedade. Nesta dissertação é documentado a análise, desenho, desenvolvimento e avaliação de uma aplicação móvel e sistema de recomendação que têm como objetivo ajudar pessoas com ansiedade.

Nos últimos anos, os sistemas de recomendação cresceram em popularidade e apreciação com os utilizadores procurando facilidade e conveniência ao receber previsões e recomendações com base nas suas preferências e características. Com os transtornos de ansiedade afetando uma percentagem considerável da população, as ferramentas para gerir a ansiedade podem ajudar pessoas que não estão a ser acompanhadas por um psicólogo.

A aplicação móvel terá uma variedade de exercícios que fornecerão maneiras para os utilizadores aliviarem sua ansiedade e fornecerão ferramentas para aprender como geri-la. A aplicação móvel estará conectada a um sistema de recomendação que poderá sugerir exercícios para alíviar a ansiedade aos utilizadores, tendo em consideração as classificações dos mesmos e as características do utilizador.

Para avaliar a funcionalidade geral dos componentes desenvolvidos foram desenvolvidos testes, sendo os resultados positivo. Além disso, as técnicas de recomendação também foram testadas para analisar o erro de previsão de cada uma. Em geral os resultados também foram positivos com erros consideravelmente baixos.

Contents

List of Figures **xiii**

List of Tables **xv**

List of Acronyms **xvii**

1 Introduction **1**

1.1 Context 1
1.2 Problem 1
1.3 Objectives 2
1.4 Motivation 3
1.5 Contribution 3
1.6 Dissertation structure 3

2 State of the art **5**

2.1 Recommendation Systems 5
2.1.1 Techniques 6
Content-based filtering 6
Collaborative filtering 7
Hybrid 9
2.1.2 Comparison of different recommendation techniques 9
2.1.3 Examples of Recommendation Systems 10
Tiktok 10
Netflix 10
Google News 10
Amazon 11
Facebook 11
2.2 Anxiety Disorders 11
2.2.1 Generalized Anxiety Disorder (GAD) 11
2.2.2 Panic Disorder 12
2.2.3 Phobia-related disorders 12
Specific Phobias 13
Social Anxiety Disorder (SAD) 13
Agoraphobia 13
Separation anxiety disorder 14
2.2.4 Post-Traumatic Stress Disorder (PTSD) 14
2.2.5 Obsessive-Compulsive Disorder (OCD) 14
2.3 Depression 15
2.4 Schizophrenia 15
2.5 Existing solutions 16
2.5.1 Communicating with therapists 17

Betterhelp 18
Talkspace 18
LARKR 18
2.5.2 Cognitive Behavioral Therapy (CBT) 18
Mindshift CBT 19
Anxiety Coach 19
2.5.3 Mindfullness 19
Headspace 19
2.5.4 Others 20
IntelliCare 20
2.6 Methods, Techniques and Technologies 20
2.6.1 Mobile applications 20
React Native 20
Ionic 21
Xamarin 21
Flutter 21
2.6.2 Web applications 22
Angular 22
React 22
Vue 22
Ember 23
Node.Js 23
2.7 Summary 23

3 Value Analysis 25
3.1 Value for customer 25
3.2 Perceived value 25
3.3 New Concept Development 26
3.3.1 Opportunity Identification 28
3.3.2 Opportunity Analysis 28
3.3.3 Idea Generation and Enrichment 28
3.3.4 Idea Selection 28
3.3.5 Concept Definition 28
3.4 Analytic Hierarchy Process 29
3.4.1 Phase 1 — Construction of the hierarchical decision tree 29
3.4.2 Phase 2 — Comparison of criteria 30
3.4.3 Phase 3 — Relative priority of each criteria 31
3.4.4 Phase 4 — Evaluating the consistence of the relative priorities 31
3.4.5 Phase 5 — Comparison Matrix for each criteria 32
3.4.6 Phase 6 — Priority for each idea 34
3.4.7 Phase 7 — Idea selection 34
3.5 Value proposition 34
3.6 Business Model Canvas 35
3.7 Summary 36

4 Design 37
4.1 System actors 37
4.2 Functional and non functional requirements 37
4.2.1 Functional requirements 38

FR01 - Request exercise recommendations . . . 40
FR02 - Receive recommendations with the use of machine learning . . 40
FR03 - Notification with recommendation . . . 40
FR04 - Recommend exercise to patient . . . 40
FA01 - Exercise with video . . . 40
FA02 - Exercise with audio . . . 40
FA03 - Exercise with text . . . 41
FA04 - Filter exercises . . . 41
FA05 - Exercise history . . . 41
FA06 - Rate exercises . . . 41
FI01 - Chat communication with other users . . . 41
FI02 - Send photo/video/audio in chat . . . 41
FI03 - Report symptoms . . . 41
FP01 - Register . . . 41
FP02 - Login . . . 41
FP03 - Setup profile . . . 41
FP04 - Define user role . . . 41
FP05 - Add information to patient profile . . . 42
4.2.2 Non-functional requirements . . . 42
Usability . . . 42
Reliability . . . 42
Performance . . . 42
Supportability . . . 42
Design constraints . . . 43
Implementation constraints . . . 43
Interface constraints . . . 43
Physical constraints . . . 43
4.3 Architecture . . . 43
4.3.1 Component Diagram . . . 43
4.3.2 Recommendation System Component Diagram . . . 44
4.3.3 Database . . . 45
4.3.4 Deploy Diagram . . . 46
Alternative 1 . . . 46
Alternative 2 . . . 47
Deployment Alternative Selection . . . 47
4.4 Summary . . . 49

5 Implementation **51**
5.1 Database and Authentication . . . 51
5.2 Recommendation System . . . 52
5.2.1 Class Diagram . . . 53
5.2.2 Recommenders . . . 54
5.2.3 Comparers . . . 61
5.2.4 Flow . . . 64
5.3 Mobile Application . . . 65
FR01 - Request exercise recommendations . . . 65
FR02 - Receive recommendations with the use of machine learning . . 66
FA01 - Exercise with video . . . 67
FA02 - Exercise with audio . . . 70

FA03 - Exercise with text 70
FA04 - Filter exercises 71
FA05 - Exercise history 72
FA06 - Rate exercises 73
FP01 - Register 74
FP02 - Login 75
FP03 - Setup profile 75

6 Evaluation **77**
6.1 Development, Testing and Evaluation Cycles 77
6.1.1 Alpha Cycle 77
6.1.2 Beta Cycle 77
6.1.3 Release Cycle 78
6.2 Quantitative Evaluation Framework (QEF) 78
6.2.1 Functionality 78
6.2.2 Adaptability 78
6.2.3 Efficiency 79
6.2.4 Results 79
6.3 Tests 79
6.3.1 Unit tests 80
6.3.2 Acceptance Tests 81
6.3.3 Response time 82
6.3.4 Prediction Errors 82
Prediction Error of Each Recommendation Technique 83
Similarity Prediction Error 84
6.4 Summary 84

7 Conclusions **85**
7.1 Achieved objectives 85
7.2 Future Work 87
7.2.1 Mobile Application 87
7.2.2 Recommendation System 87

Bibliography **89**

A QEF **97**

List of Figures

2.1 Types of Recommendation Systems (Isinkaye et al., 2015) 6
2.2 Content-based filtering recommendation system (Grimaldi, 2018) 7
2.3 Collaborative filtering recommendation system (dataaspirant, 2015) 8
2.4 Google Trends comparison of Betterhelp, Talkspace and LARKR ("Betterhelp, Talkspace, LARKR - Google Trends", n.d.) 17

3.1 Innovation process 26
3.2 New Concept Development 27
3.3 Hierarchical decision tree divided in three criteria 30
3.4 Updated hierarchical decision tree divided in three criteria 34

4.1 Use Case Diagram (Part 1) 38
4.2 Use Case Diagram (Part 2) 39
4.3 Use Case Diagram (Part 3) 39
4.4 Component Diagram 44
4.5 Components Diagram 45
4.6 Entity Relationship Diagram (ERD) 45
4.7 Database structure 46
4.8 Deployment Diagram 1 47
4.9 Deployment Diagram 2 47
4.10 Updated deploy diagram 48

5.1 Connection to Firebase Sequence Diagram 51
5.2 Sign In Sequence Diagram 52
5.3 Firestore example 52
5.4 Recommendation System Class Diagram 53
5.5 Table transformation 54
5.6 *UserItemRatingsTable* Class 55
5.7 *UserItemRatings* class 55
5.8 Rating Matrix with Users, Items and Latent Features 56
5.9 *Initialize* method 57
5.10 *FactorizeMatrix* method 58
5.11 *GetSuggestions* method from *UserCollaborativeFilterRecommender* 61
5.12 Cosine Similarity Method 62
5.13 Co-Rated Cosine Similarity Method 62
5.14 Pearson Correlation Similarity Method 63
5.15 Root Mean Square Similarity Method 64
5.16 Activity Diagram of the Recommendation System 64
5.17 Recommendation System Sequence Diagram 65
5.18 Activity with various recommendations 66
5.19 Exercise Recommendation in the Main Activity 67
5.20 Exercise List 68

5.21 Video Exercise 69
5.22 Audio Exercise 70
5.23 Text Exercise 71
5.24 Filter Floating Action Menu 71
5.25 *FilterExercises* method 72
5.26 Exercise History Interface 73
5.27 Rate modal window 74
5.28 Method for rate button click 74
5.29 Sign In Intent 75
5.30 Edit Profile Activity 76

6.1 Import Logic for MovieLens Ratings 80
6.2 Similarity Unit Test 81

List of Tables

2.1 The conceptual goals of content-based and collaborative recommendation systems (Aggarwal, 2016) . . . 9
2.2 Recommendation techniques advantages and disadvantages (Burke, 2002) (Shokeen & Rana, 2019) (Agarwal et al., 2017) . . . 10

3.1 Benefits and Sacrifices . . . 26
3.2 Fundamental Scale . . . 31
3.3 Criteria comparison matrix . . . 31
3.4 Criteria Relative Priority . . . 31
3.5 Random Index Table . . . 32
3.6 Time and Effort of Development Comparison Matrix . . . 33
3.7 Anxiety Alleviation Comparison Matrix . . . 33
3.8 Usability/Convenience Comparison Matrix . . . 33
3.9 Canvas Model . . . 36

4.1 FURPS+ classification system . . . 38
4.2 Use cases . . . 40

6.1 Acceptance Tests . . . 82
6.2 Response time of recommendation system . . . 82
6.3 Prediction Error of the Different Recommendation Techniques . . . 83
6.4 Prediction Error of the Similarity Somparers . . . 84

7.1 Dissertation Objectives . . . 86
7.2 Use Cases Status . . . 86

List of Acronyms

AAA	Arrange, Act and Assert
AI	Artificial Intelligence
AHP	Analytic Hierarchy Process
BaaS	Backend-as-a-Service
CBT	Cognitive Behavioral Therapy
CI	Consistency Index
CR	Consistency Ratio
CSS	Cascading Style Sheets
EU	European Union
ERD	Entity Relationship Diagram
FFE	Fuzzy Front End
FURPS	Functionality Usability Reliability Performance Supportability
GAD	Generalized Anxiety Disorder
GAD-7	Generalized Anxiety Disorder 7 Questionnaire
GDP	Gross Domestic Product
HP	Hewlett-Packard
HTML	HyperText Markup Language
IaaS	Infrastructure as a Service
ISEP	Instituto Superior de Engenharia do Porto
JIT	Just-In-Time
JS	JavaScript
JSX	Javascript-XML
MAE	Mean Absolute Error
MVC	Model View Controller
NCD	New Concept Development
NPM	Node Package Manager
NPD	New Product Development

OCD	Obsessive-Compulsive Disorder
PaaS	Platform as a Service
PHQ-9	Patient Health Questionnaire-9
PTSD	Post-Traumatic Stress Disorder
QEF	Quantitative Evaluation Framework
RCT	Randomized Controlled Trial
RI	Random Index
RMSE	Root Mean Square Error
SaaS	Software as a Service
SAD	Social Anxiety Disorder
SDK	Software Development Kit
SEO	Search Engine Optimization
SPA	Single Page Application
SVD	Singular Value Decomposition
UI	User Interface

Chapter 1

Introduction

The purpose of this chapter is to contextualize the reader on the subject of mental illnesses, explain the problem, tell what are the objective of this project and how they're going to contribute to solving the problem. Next, the contributions and the motivational factors that led to the development of this project are indicated. Lastly, the structure of the document is identified, in order to describe the content of each chapter.

1.1 Context

Mental illness comprehends many clinical conditions associated with several changes that include limitations related with thinking capacity and reality understanding, deficits in communication skills and difficulties in developing appropriate emotional and behavioral responses. One of its core features are cognitive impairments which are present at illness onset and remain relatively stable over the course of illness, jeopardizing psychosocial functionality (Paz et al., 2017). Anxiety disorders are one of the mental health problems that are the most common therefore having a big impact on our society (ADAA, n.d.-b).

Mental illness self-management is an approach designed to involve people with mental health problems as active agents in their own treatment and recovery, teaching them to monitor their condition and to use appropriate coping strategies.

Information technology can be an adequate tool for mental illness self-management, which has been shown to be effective (Lorig et al., 2006), and applied for example to smoking cessation (Bricker et al., 2014) and stress management (Ly et al., 2014). Mobile applications also seem to contribute and help their users to engage in health promoting behaviors outside the clinical context or in other activities such as therapeutic homework and it generally improves mental health symptoms (Webb et al., 2017)(Howells et al., 2016)(Chandrashekar, 2018).

Mobile apps are a great choice for psychological treatment delivery compared to other platforms due to various reasons such as ease of habit, low effort expectancy, and high motivation. Also, Artificial Intelligence (AI) technology has the potential to transform mental healthcare by providing tools that could allow the identification of mental health problems at an earlier stage when interventions may be more effective, and it may also allow the recommendation of certain treatments based on an individual's unique characteristics (Chandrashekar, 2018).

1.2 Problem

Mental health issues affect about 84 million people across European Union (EU) countries. The mental health problem most common across EU countries is anxiety disorder, with an estimated 25 million people, that is 5.4% of the population living with anxiety disorders (OECD, 2018). Because of this the costs of mental health problems on EU economies are immense. In 2015, the overall costs related to mental ill-health[1] are estimated to have exceeded 4% of Gross Domestic Product (GDP) across the 28 EU countries. This means that more than EUR 600 billion are being spent in helping people with mental illness or mental problems (OECD, 2018). There's only so much that a country's health care system can do to help people manage their mental ill-health, as such there's a need for people to self-manage their anxiety problems.

There is a scope of opportunities to use mobile applications in engaging intervention or pedagogical strategies for mental disease self-management, and to promote practices that aim to improve mental health.

1.3 Objectives

The aim of this dissertation is to design and develop an application prototype, with the objective of improving self-management of anxiety disorders. This objective will be achieved by pursuing the following tasks:

- Investigation and analysis of various mental health mobile applications;
- Investigation and analysis of various recommendation systems;
- Identify solutions to the problem described in section 1.2;
- Design an architecture that satisfies the identified requirements;
- Implementation of a mobile application that provides exercises to alleviate anxiety;
- Implementation of a recommendation system that suggests exercises to users through the use of machine learning;
- Integration of caregivers and therapists;
- Conduct a study that demonstrates the utility of the developed application, followed by its analysis.

The use of mobile devices seems to be the logical path for self-management. Self-management applications in mobile devices may help users perform anxiety relief exercises on the go and it can also help alerting users, caregivers and therapists in a faster way than traditional methods.

The main focus of the project will be the implementation of exercises that alleviate anxiety and a recommendation system that is able to suggest those exercises to users. These exercises and recommendations will help the user better adjust lifestyle habits, it will also promote the reduction of risk factors of anxiety disorders and promote independent living. The recommendations will vary from user to user, depending from the anxiety disorder that they have to the ratings that certain exercises have. This project also has the objective of implementing a day-to-day log in which the user can report symptoms by describing an anxiety experience or an event that triggered a social problem. This feature also works as a record of user mood over time and allows the review of past entries, this can be also be useful for the caregiver and therapist.

[1]Mental ill-health is an umbrella term that includes both mental illness and mental health problems

1.4 Motivation

The idea of this project was born because anxiety illness affects so much of us and those close to us, and since it is different from person to person it means that the treatment is different too, two of the most popular types of treatments are therapy and medication. There are a lot of different ways that therapy can be performed. It can be performed one-on-one, as a group, or through the use of technology which allows people with anxiety to talk with therapists through the phone or text them or just give help in general by providing coping or meditation skills (ADAA, n.d.-a).

Many people seek technology to help their anxiety, there are those who don't want medication and those who don't feel comfortable with one-on-one or group therapy sessions or simply don't have the time or energy to do so. This project will be developed with those people in mind.

1.5 Contribution

There's already several applications to help people with anxiety but there's not many that have a recommendation component, this project will try to fill the current gap that exists on the market of mental health applications, more specifically, the market of applications that focus on anxiety. This application prototype will be named WEBECOOL and the plan for it is to have exercises that will help users manage their anxiety, contain a recommendation system, communication with therapists and participation of caregivers. All of this will help users with anxiety and will set it apart from other applications.

This dissertation will also provide insight for future developers who want to build an application related to mental illness and with a recommendation system.

1.6 Dissertation structure

This document is divided in seven chapters:

1. Introduction — The current chapter, a brief introduction to the dissertation theme, the problem and what are the objectives.
2. State of the art — Information related to the theme of this project is gathered in this chapter. Recommendation systems and anxiety disorders are discussed in detail and examples of existing solutions are shown.
3. Value Analysis — The project is studied and an analysis is performed on this chapter. The value of this project is discussed and the Canvas Business Model is presented.
4. Design — In this chapter the function and non-functional requirements are determined, as well as the actors and the use cases. Different design solutions are analyzed but one is chosen and it is then explained in more detail.
5. Implementation — The entire development process of the application taking into account functional and non-functional objectives and the requirements.
6. Evaluation — Evaluation and quality control plan are described in this chapter alongside with tests and the calculation of the prediction error of the recommendation system.

7. Conclusions — This last chapter presents the conclusions about the project, as well as the future work that could be developed for the continuation of the project and some personal considerations will also be shared.

Chapter 2

State of the art

This chapter has the objective of contextualizing the reader on recommendation system techniques, anxiety disorders and other mental illness. There will also be present an overview of various existing solutions as well as an overview of technologies of mobile and web software development.

2.1 Recommendation Systems

Whether it be professionally or personally, in our day-to-day lives we have to make decisions without often having enough personal experience or knowledge about the different options. Therefore, we often base our decisions on the recommendations of others. A recommendation system assists the decision-making process based on the experience of others (Resnick & Varian, 1997).

We can define a recommendation system, as a system that is able to make pertinent suggestions that are adjusted to the needs of someone.

One example of a recommendation system in mental health software is the IntelliCare suite of apps mentioned in the sub section 2.5.4 below. Using a centralized app approach, it has Hub app that provided app recommendations aiming to increase user engagement and provide the ideal exercises for each type of difficulty the user is experiencing.

A study which collected the records of 8057 users over 12 IntelliCare apps measured overall engagement and app-specific usage longitudinally by the number of weekly app sessions (loyalty) as well as the number of days with app usage (regularity) over 16 weeks. The results showed that among Hub users, Hub recommendations increased app-specific loyalty and regularity across all 12 apps. Centralized app recommendations overall increased user engagement of the apps and app-specific usage. (Cheung et al., 2018)

However, most software solutions to help users manage anxiety disorders are not a suite of apps, usually they are a single application. Possible recommendation systems inside these applications don't recommend other apps, instead they recommend different types of exercises or procedures. This is the case of a certain smartphone-based personalized activity recommender system for the recognition and self-management of negative emotions. This recommender system considers the users and similar subjects' preferences of activity and histories of choices simultaneously in the current context to make the best recommendations. In a study, it verified that the recommender system was able to provide useful suggestions, which indeed alleviated users' negative emotions. (Hung et al., 2015)

2.1.1 Techniques

There are three main techniques of recommendation systems: Content-based filtering; Collaborative filtering; Hybrid.

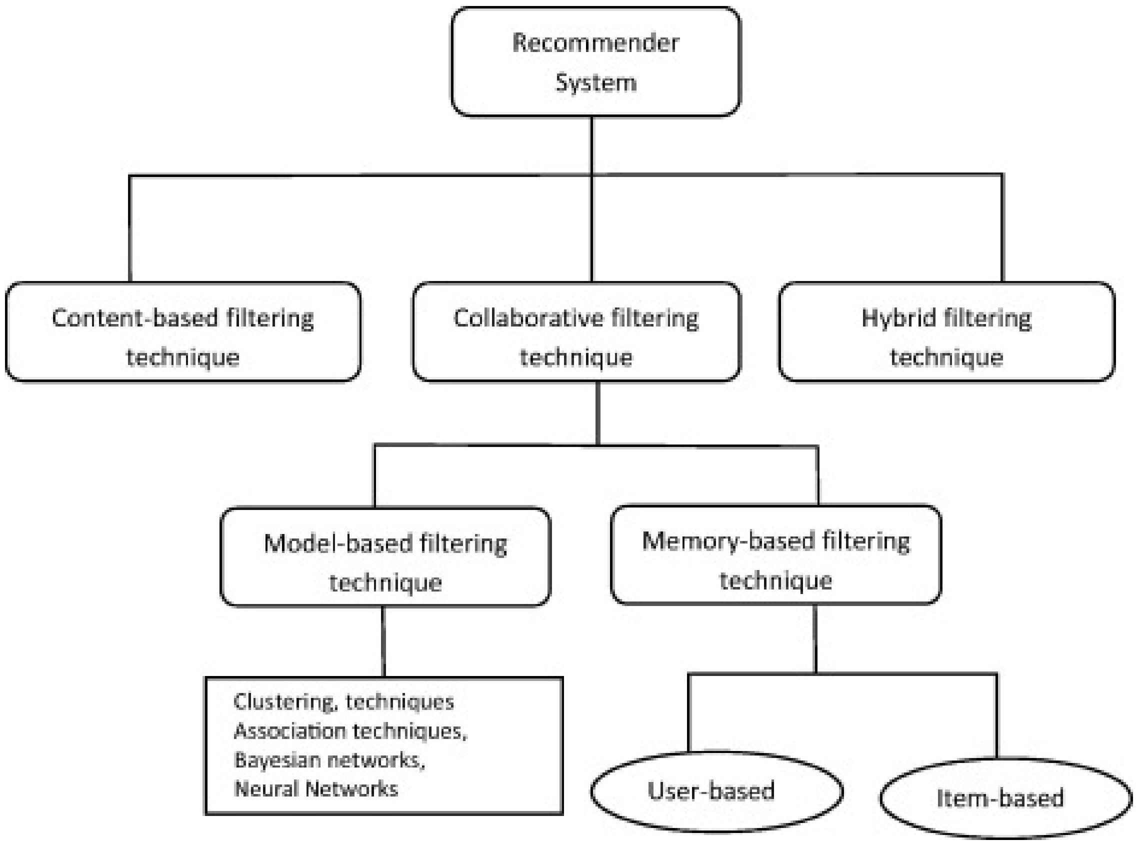

Recommender System
Content-based filtering technique
Collaborative filtering technique
Hybrid filtering technique
Model-based filtering technique
Memory-based filtering technique
Clustering, techniques
Association techniques,
Bayesian networks,
Neural Networks
User-based
Item-based

FIGURE 2.1: Types of Recommendation Systems (Isinkaye et al., 2015)

The names of these types of recommendation systems are pretty self explanatory, a content-based filtering recommendation system recommends with the content in mind while a collaborative takes into account the actions and preferences of other users. A brief explanation of how each type works can be seen in the figures 2.2 2.3 below.

Content-based filtering

Content-based filtering is based on a description of the item and a profile of the users preference. In a content-based recommendation system, keywords are used to describe the items and a user profile is built to indicate the type of item this user likes. It then recommends items that are similar to those that a user liked in the past, or is examining in the present (Kordík, 2018) (Agarwal et al., 2017).

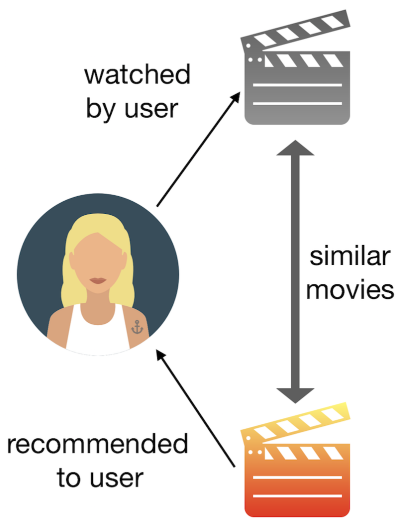


FIGURE 2.2: Content-based filtering recommendation system (Grimaldi, 2018)

In short, it uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback. In the figure 2.2 there's an example of a simple content-based filtering recommendation system, a user watches a movie and other similar movies are recommended, for example, a user likes and mostly watches horror movies, the recommendation system will recommend other horror movies (Kordík, 2018).

Content-based recommendation have advantages, it's great at recommending new items when there's not a lot of information about the user. But it also has some disadvantages, the recommendations can be overspecialized, using the last example, perhaps the user has already watched most horror movies and the recommendation system keeps recommending horror movies that the user already knows about, or it can recommend always the same horror movies. Another issue with content-based recommendation systems is that while they're good at recommending new content, they have difficulty recommending to new users due to the fact that the system doesn't know what type of content that he likes, this issue is called cold start (Kordík, 2018).

Collaborative filtering

Collaborative filtering identifies interactions between users and the items whether it is explicit feedback such as a star rating or thumb-up/thumb-down, or implicit feedback such as the number of episodes watched in a TV show. Recommendation systems that use Collaborative filtering aggregate ratings or recommendations of objects, recognize commonalities between the users on the basis of their ratings, and generate new recommendations based on inter-user comparisons (Burke et al., 2011) (Agarwal et al., 2017).

This type of recommendation system is very common in online stores. For example, a variation of the collaborative recommendation system algorithm is currently used on Amazon. Actions such as the user adding an item to a wishlist, buying or positively rating a similar item, searching for a specific type of product are actions that will help a collaborative filtering recommendation system give better recommendations for the user and other similar users (Burke et al., 2011).

In the figure 2.3 below a simple example of a collaborative filtering recommendation system can be examined, there's two similar users (e.g. both male, both are cyclists) and they both previously bought two food items but one of the users also bought another food item, that item will then be recommended to the similar user. This example is exactly what happens when you see a "Customers Who Bought This Item Also Bought" recommendation on a website.

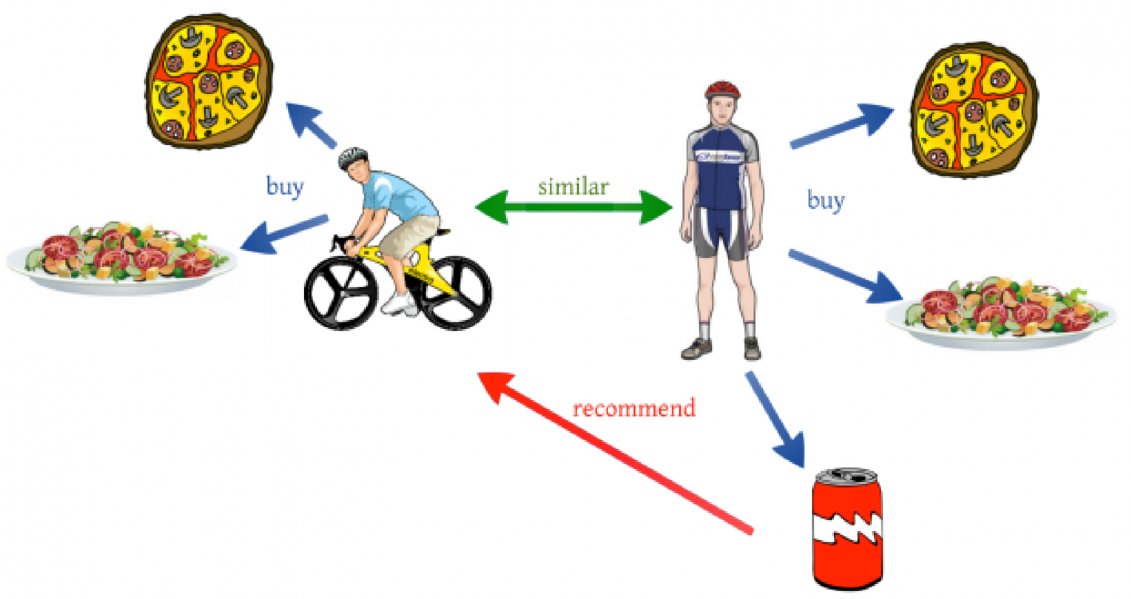


FIGURE 2.3: Collaborative filtering recommendation system (dataaspirant, 2015)

Within collaborative filtering there different types of recommendation algorithms such as:

- **Model-based filtering** — It uses machine learning or data mining techniques to create a model capable of recommending based on the interests of users. These techniques can quickly recommend a set of items because of the the fact that they use pre-computed model.
- **Memory-based filtering** — It uses the data available in the system to establish correlations between products or users, based on similarity.

Collaborative filtering has various advantages, they are simple and they don't need to understand the items themselves to recommend and they also have the potential to solve the cold start issue that content-based filtering has, if basic information about the user is known the system can start recommending items that other similar used liked. But there are also disadvantages like the data sparsity problem, not all items are classified by the users and the recommendation system will not recommend items that weren't classified, resulting in a lot of items that will never get recommended. The cold start problem can also occur, not having information about the items or users will cause it. Lastly there's the gray sheep problem where a specific user has no other similar users and because of that the system has difficulty finding recommendations (Burke et al., 2011) (Agarwal et al., 2017).

Hybrid

Hybrid recommendation systems allow combining content-based filtering and collaborative filtering in a manner that often suits a particular industry. It provides the strengths of more two recommendation systems and also has fewer of the drawbacks of any individual system (Çano & Morisio, 2017) (Agarwal et al., 2017). There are several ways in which the systems can be combined, such as:

- **Weighted Hybrid** — Combines the results of the different recommendation techniques, assigning different weights to each of them. The weights assigned to each of the techniques can then be adjusted according to user feedback.
- **Switching Hybrid** — Switches between the recommendation techniques depending on the context.
- **Mixed Hybrid** — Recommendations from more different techniques are presented together, creating a wide range of recommendations.

As previously mentioned, one of the biggest advantages of a hybrid recommendation system is that it solves the problems that content-based or collaborative filtering systems have. When there's not much information about the user, recommendations can still occur based on similar content, when there's no similar users the system can use similar users, this means that when there's a cold start and the system doesn't know about what type of content the user likes it can still provide recommendations. The only disadvantage is that two types of recommendations systems are being used and there's slightly more complexity and resources being used (Çano & Morisio, 2017).

2.1.2 Comparison of different recommendation techniques

The content-based and collaborative techniques mentioned in the subsection above differ in several aspects, such as the objective, the inputs, the general process, the advantages, the disadvantages, among others. In the table 2.1, the content-based and collaborative techniques are compared, comparing their main conceptual goal and the different forms of input.

Approach	Conceptual Goal	Input
Content-based	Give me recommendations based on the content (attributes) I have favored in my past ratings and actions	User ratings + item attributes
Collaborative	Give me recommendations based on a collaborative approach that leverages the rating and actions of my peers/myself	User ratings + community ratings

Table 2.1: The conceptual goals of content-based and collaborative recommendation systems (Aggarwal, 2016)

The table 2.2 provides a summary of the advantages and disadvantages for each technique discussed in Techniques subsection.

Type	Advantages	Disadvantages
Content-based filtering	Doesn't have item cold start problem Quality improves over time Implicit feedback is sufficient	User cold start problem Quality is dependent on high amounts of data Overspecialization
Collaborative filtering	Domain knowledge not needed Quality improves over time Implicit feedback is sufficient	Cold start problem Gray sheep problem Data Sparsity problem Quality is dependent on high amounts of data
Hybrid	Robust Mitigates the disadvantages of content-based and collaborative	Knowledge of combining techniques

TABLE 2.2: Recommendation techniques advantages and disadvantages (Burke, 2002) (Shokeen & Rana, 2019) (Agarwal et al., 2017)

2.1.3 Examples of Recommendation Systems

In this subsection some applications of recommendation systems will be presented. The recommendations in these applications are a huge reason for their success since they're an integral part of their system.

Tiktok

TikTok has a main stream of content called the "For You" feed, its personalized to the individual user. The "For You" feed is powered by a recommendation system that makes a different "For you" feed for each user because the videos that are shown depend on the videos that the user likes or shares, the accounts followed, the comments posted and the content the user creates to help determine your interests. In addition, the recommendation system will factor in video information like the captions, sounds and hashtags associated with the content the user likes. Other signals contribute to the recommendation system understanding of what a user likes, for example, if a user watches a longer video from beginning to end, its considered a strong indicator of interest. Users can also give give explicit feedback by liking, add a video to their favorites or mark it Not interested. (Perez, 2020).

Netflix

Netflix provides its users the ability to rate movies and television series on a 5-point scale, it also saves the user actions in terms of watching there items. It then is able to recommend films and TV series based on the user actions and ratings assigned to them. The user profile is also taken into account, prioritizing, for example, favorite categories. In addition to this, Netflix explicitly provides examples of recommendations based on the films that were watched by the user. For example, if the user watched Titanic, a section with the title "Because you watched Titanic" will appear, it would contain recommended movies based on Titanic. Presenting explanations of the recommendations made is important to provide the user with an understanding of how the items were suggested, and may, in an easier way, originate a possible interest in them (Aggarwal, 2016).

Google News

Google News features a collaborative filtering system, it recommends news to the user based on their history of clicks. These clicks are associated with the user through identification mechanisms enabled by Gmail accounts. In this case, news articles are treated as items, where the act of clicking and viewing the news articles can be seen as a positive rating. In this scenario, ratings are obtained through user actions, rather than be explicitly assigned by him (Aggarwal, 2016).

Amazon

Amazon recommends products (e.g. books, games, electronics, among others) on its own website taking into account the 5-point scale ratings, buying behavior and the browsing behavior of users, in order to personalize the shopping experience of these users. This information is stored in a specific database relating it to the user who is authenticated. Similarly to 2.1.3, Amazon in many cases provides explanations for recommendations. For example, the relationship of a recommended item to previously purchased items may be included in the Amazon interface (Aggarwal, 2016).

Facebook

Facebook recommends potential friends to users in order to increase the number of social connections, based on structural relationships and not on rating data. Therefore, the nature of the underlying algorithms is different compared to the recommendation systems described above. Likewise, the objectives are also different. While a product recommendation directly increases the merchant's profit by facilitating product sales, an increase in the number of social connections improves a user's experience on a social network (Aggarwal, 2016).

2.2 Anxiety Disorders

While everyone can feel anxiety and it serves us a purpose, there are anxiety disorders that involve more than temporary worry or fear. The anxiety does not go away and can get worse over time for those with anxiety disorders. It can affect negatively daily activities such as job performance, school work, and relationships.

Research suggests that those who suffer from an anxiety disorder may run a higher risk of experiencing physical health problems, too. ("ADAA Managing Stress and Anxiety", n.d.)

An estimated 264 million people worldwide have an anxiety disorder (Ritchie & Roser, 2018).

An estimated 31.1% of U.S. adults experience any anxiety disorder at some time in their lives ("NIMH Any Anxiety Disorder", n.d.).

An estimated 19.1% of U.S. adults had any anxiety disorder in the past year ("NIMH Any Anxiety Disorder", n.d.).

There are several types of anxiety disorders, including Generalized Anxiety Disorder (GAD), panic disorder, various phobia-related disorders, Post-Traumatic Stress Disorder (PTSD) and Obsessive-Compulsive Disorder (OCD) ("NIMH Anxiety Disorders", n.d.) ("NIMH Any Anxiety Disorder", n.d.):

2.2.1 Generalized Anxiety Disorder (GAD)

People suffering from GAD display an excessive amount of anxiety or worry, most days for at least 6 months, about numerous things such as work, personal health, social interactions, and everyday routine life. The fear and anxiety can cause significant issues in areas of their life, such as social interactions, school, and work. ("NIMH Anxiety Disorders", n.d.)

Generalized anxiety disorder symptoms include:

- Feeling restless, wound-up, or on-edge
- Difficulty controlling feelings of worry

- Being easily fatigued
- Being irritable
- Having difficulty concentrating; mind going blank
- Having muscle tension
- Having sleep problems, such as difficulty falling or staying asleep, restlessness, or unsatisfying sleep

2.2.2 Panic Disorder

People with panic disorder have recurrent unexpected panic attacks. Panic attacks are sudden periods of intense fear that come on quickly and reach their peak within minutes. Panic attacks can be unexpected or may be expected, such as a response to a feared object, or unexpected, apparently occurring for no reason. ("What Are Anxiety Disorders?", n.d.)

During a panic attack, these symptoms may occur:

- Heart palpitations, a pounding heartbeat, or an accelerated heartrate
- Sweating
- Trembling or shaking
- Sensations of shortness of breath, smothering, or choking
- Feelings of impending doom
- Feelings of being out of control
- Chills or hot flashes
- Feeling dizzy, light-headed or faint
- Feeling detached
- Nausea or abdominal pains

Those with panic disorder know their fear is excessive, but they cant overcome it. These fears cause such distress that some people go to extreme lengths to avoid what they fear by avoiding places, situations, or behaviors they associate with panic attacks. Worry about panic attacks, and the effort spent trying to avoid attacks, cause significant problems in various areas of the persons life, including the development of Agoraphobia ("NIMH Anxiety Disorders", n.d.).

2.2.3 Phobia-related disorders

A phobia is an intense fear of/or aversion to specific objects or situations. Even though it can be realistic to be anxious in some circumstances, the fear or aversion people with phobias feel is out of proportion to the actual danger caused by the situation or object. ("NIMH Anxiety Disorders", n.d.)

People with a phobia:

- May have an irrational or excessive worry about encountering the feared object or situation
- Take active steps to avoid the feared object or situation

- Experience immediate intense anxiety upon encountering the feared object or situation
- Endure unavoidable objects and situations with intense anxiety

There are several types of phobias and phobia-related disorders such as:

Specific Phobias

As the name suggests, people who have a specific phobia have an intense fear of, or feel intense anxiety about, certain types of objects, situations or activities that are generally not harmful. ("NIMH Anxiety Disorders", n.d.)

Some examples of specific phobias include the fear of:

- Flying
- Heights
- Specific animals, such as spiders, dogs, or snakes
- Receiving injections
- Blood

Social Anxiety Disorder (SAD)

People with Social Anxiety Disorder (SAD) have a general intense fear of, or anxiety toward, social or performance situations. They worry that actions or behaviors associated with their anxiety will be negatively evaluated by others, leading them to feel embarrassed. People with this disorder will try to avoid social situations or endure them with great anxiety. Social anxiety disorder can manifest in a range of situations, such as within the workplace or the school environment. Common examples are extreme fear of public speaking, meeting new people or eating/drinking in public. ("NIMH Anxiety Disorders", n.d.)

Agoraphobia

Agoraphobia is the fear of being in situations where escape may be difficult or embarrassing, or help might not be available in the event of panic symptoms. The fear is out of proportion to the actual situation ("What Are Anxiety Disorders?", n.d.).

A person with agoraphobia experiences this fear in two or more of the following situations:

- Using public transportation
- Being in open spaces
- Being in enclosed spaces
- Standing in line or being in a crowd
- Being outside of the home alone

The individual actively avoids the situation, in part, because they think being able to leave might be difficult or impossible in the event they have panic-like reactions or other embarrassing symptoms. Untreated agoraphobia can become so serious that a person may be unable to leave the house. A person can only be diagnosed with agoraphobia if the fear is intensely upsetting, or if it significantly interferes with normal daily activities ("What Are Anxiety Disorders?", n.d.).

Separation anxiety disorder

Separation anxiety is often thought of as something that only children deal with; however, this is not completly true since adults can also be diagnosed with separation anxiety disorder. A person with separation anxiety disorder is excessively fearful or anxious about separation from those with whom he or she is attached. They often worry that some sort of harm or something untoward will happen to their attachment figures while they are separated. This fear leads them to avoid being separated from their attachment figures and to avoid being alone. People with separation anxiety may also have nightmares about being separated from attachment figures or experience physical symptoms when separation occurs or is anticipated. ("NIMH Anxiety Disorders", n.d.)

2.2.4 Post-Traumatic Stress Disorder (PTSD)

PTSD is a disorder that can occur in people who have experienced or witnessed a traumatic event such as a natural disaster, a serious accident, a terrorist act, war/combat, rape or other violent personal assault.

In the past this disorder has been known by many names, such as "shell shock" during World War I and "combat fatigue" after World War II. Despite popular belief PTSD does not just happen to combat veterans, It can occur in all people of any ethnicity, nationality or culture, and any age. PTSD affects approximately 3.5 percent of U.S. adults, and an estimated one in 11 people will be diagnosed PTSD in their lifetime. Women are twice as likely as men to have PTSD ("What Is PTSD?", n.d.).

Those struggling with PTSD have intense, disturbing thoughts and feelings related to their experience that last long after the traumatic event has ended. They may relive the event through flashbacks or nightmares; they may feel sadness, fear or anger; and they may feel detached or estranged from other people. Because of this, those with PTSD often avoid situations or people that remind them of the traumatic event, and they may have strong negative reactions to something as ordinary as a loud noise or an accidental touch.

It is completely natural to feel afraid during and after a traumatic situation. Fear triggers many split-second changes in the body to help protect against danger or to avoid it. This fight-or-flight response is a typical reaction designed to protect a person from harm. Nearly everyone will experience a range of reactions after trauma, yet most people recover from initial symptoms naturally. However people who have PTSD may feel stressed or frightened, even when they are not in danger. Those who continue to experience difficulties may be diagnosed with PTSD ("NIMH Post-Traumatic Stress Disorder", n.d.).

2.2.5 Obsessive-Compulsive Disorder (OCD)

OCD is a chronic mental disorder in which a person has uncontrollable, reoccurring thoughts (obsessions) and/or behaviors (compulsions) that he or she feels the urge to repeat over and over again. The repetitive behaviors, such as hand washing, checking on things or cleaning, can significantly interfere with a persons everyday tasks and social interactions ("NIMH Obsessive-Compulsive Disorder", n.d.).

Many people have focused thoughts or repetitive behaviors, but these do not interfere with the everyday life and can add structure or make tasks easier. For those with OCD, thoughts are persistent and unwanted routines and behaviors are rigid and failure to do so causes great distress or anxiety. Many people with OCD know or suspect their obsessions are not true however there are others that may think that their obsessions could be legitimate. Even if they realize that their obsessions are not true, it is difficult for people with OCD to hold their focus away from obsessions or avoid compulsive behavior.

A diagnosis of OCD requires the presence of obsession and/or compulsions that are time-consuming (more than one hour a day), cause major distress, and affect work, social or other important function. Approximately 1.2 percent of Americans have OCD and among adults slightly more women than man are affected. OCD often begins in childhood, puberty or early adulthood; the average age symptoms appear is 19 years old (“What Is Obsessive Compulsive Disorder?”, n.d.).

2.3 Depression

Depression (major depressive disorder) is a common and serious medical illness that has a negative impact on how you feel, think, and act. Fortunately, it is also treatable. Depression causes feelings of sadness and/or a loss of interest in activities once enjoyed. It can contribute to a wide range of emotional and physical problems and can reduce the ability of a person to function at work and at home (“What Is Depression?”, n.d.).

Depression symptoms can vary from mild to severe and can include:

- Feeling sad or having a depressed mood
- Loss of interest or pleasure in activities once enjoyed
- Changes in appetite weight loss or gain unrelated to dieting
- Trouble sleeping or sleeping too much
- Loss of energy or increased fatigue
- Increase in purposeless physical activity (e.g., hand-wringing or pacing) or slowed movements and speech (actions observable by others)
- Feeling worthless or guilty
- Difficulty thinking, concentrating or making decisions
- Thoughts of death or suicide

Symptoms must last at least two weeks for a diagnosis of depression.

Also, medical conditions (e.g., thyroid problems, a brain tumor or vitamin deficiency) can resemble symptoms of depression so it is important to rule out general medical causes.

An estimated one in 15 adults (6.7%) in any given year is affected by depression. And at some point in their lives, one in six people (16.6%) will experience depression. Depression can strike at any time, but on average, first appears during the late teens to mid-20s. Women are more likely than men to have depression. Some studies show that one-third of women in their lifetime will experience a major depressive episode. (“What Is Depression?”, n.d.)

2.4 Schizophrenia

Schizophrenia is a chronic and severe mental disorder that affects a person’s way of thinking, feeling, and acting. People with schizophrenia may appear to have lost touch with reality.

Although schizophrenia is not as severe as other mental disorders, the symptoms may be very disabling.

Schizophrenia symptoms usually start between the ages of 16 and 30 but in some uncommon cases, children have schizophrenia too.

Schizophrenia symptoms fall into three categories: positive, negative, and cognitive.

"Positive" symptoms are psychotic behaviors not generally seen in healthy people. People with positive symptoms may lose touch with some aspects of reality. Symptoms include: hallucinations, delusions, unusual or dysfunctional ways of thinking, agitated body movements.

"Negative" symptoms are associated with disruptions to normal emotions and behaviors. Symptoms include: reduced expression of emotions via facial expression or voice tone, reduced feelings of pleasure in everyday life, difficulty beginning and sustaining activities, reduced speaking.

The cognitive symptoms of schizophrenia are mild for some patients, but for others they are more severe and patients may experience changes in their memory or other aspects of thinking. Symptoms include: poor ability to understand information and use it to make decisions, trouble focusing or paying attention, problems with the ability to use information immediately after learning it. ("NIMH Schizophrenia", n.d.)

2.5 Existing solutions

The number of mobile health apps rapidly increased in the past few years, these apps claim to help users deal with various mental health issues such as GAD, SAD, Depression, Addiciton, PTSD and even schizophrenia (Anthes, 2016). Mental health combining with computer technology isn't new to psychology, there's studies showing that internet-based Cognitive Behavioral Therapy (CBT), a therapeutic approach with the objective of changing problematic thoughts and behaviors, can be effective for treating conditions such as depression, GAD and eating disorders (Webb et al., 2017)(Howells et al., 2016)(Chandrashekar, 2018). But many of these online therapeutic programs are designed to be completed in lengthy sessions in front of a conventional computer screen. Smartphone apps, on the other hand can be good choice for psychological treatment delivery compared to other platforms due to ease of habit, can be used on the go and low effort expectancy (Yuan et al., 2015).

Depression A meta-analysis of 18 Randomized Controlled Trials (RCTs) covering two mobile apps revealed that using apps to alleviate symptoms and self-manage depression significantly reduced patients depressive symptoms compared to control conditions. Smartphone-based therapies were also found to yield the greatest benefits for individuals with mild to moderate, rather than major, depression (Firth et al., 2017).

Anxiety A meta-analysis of 9 RCTs that evaluated the effects of mobile apps interventions on symptoms of anxiety disorders revealed that users experienced reductions in anxiety after using anxiety treatment apps. Additionally, anxiety-focused mobile apps delivered the greatest reductions in anxiety symptoms when paired with face-to-face or internet-based therapies. In fact, using a mobile app resulted in no significant loss of treatment efficacy when compared to sessions with therapists (Ly et al., 2015).

Schizophrenia A review of 5 studies focused on using smartphone apps for treating symptoms of schizophrenia demonstrated app retention was 92%. Self-reported patient experience survey results revealed high adherence, positive user experience, and broad-ranging clinical benefits (Firth & Torous, 2015).

In this section it will be presented a list of some of the most popular or well researched Mobile Health software solutions that help managing mental health, specifically depression and anxiety.

The web page "ADAA Reviewed Mental Health Apps", n.d. contains a list of reviews for various applications, the applications that best coincide with managing anxiety disorders will be presented below and the ones that have a specific sole focus such as managing OCD will not be discussed.

Some other applications were obtained from a website that appeared as the first result of google searching "app to talk with therapist". This way we'll discuss the apps that people looking for this kind of solution will initially find after a quick search, these applications are Betterhelp, Talkspace and LARKR and they'll be discussed in the sub-section below (Mazzo, n.d.).

It will also be presented a different type of software solution that helps with anxiety, IntelliCare is a suite of apps, it contains various apps.

2.5.1 Communicating with therapists

Betterhelp, Talkspace and LARKR are three platforms which allow users to talk with therapists in order to manage their anxiety, this kind of services can greatly improve the reduction of anxiety symptoms as mentioned previously.

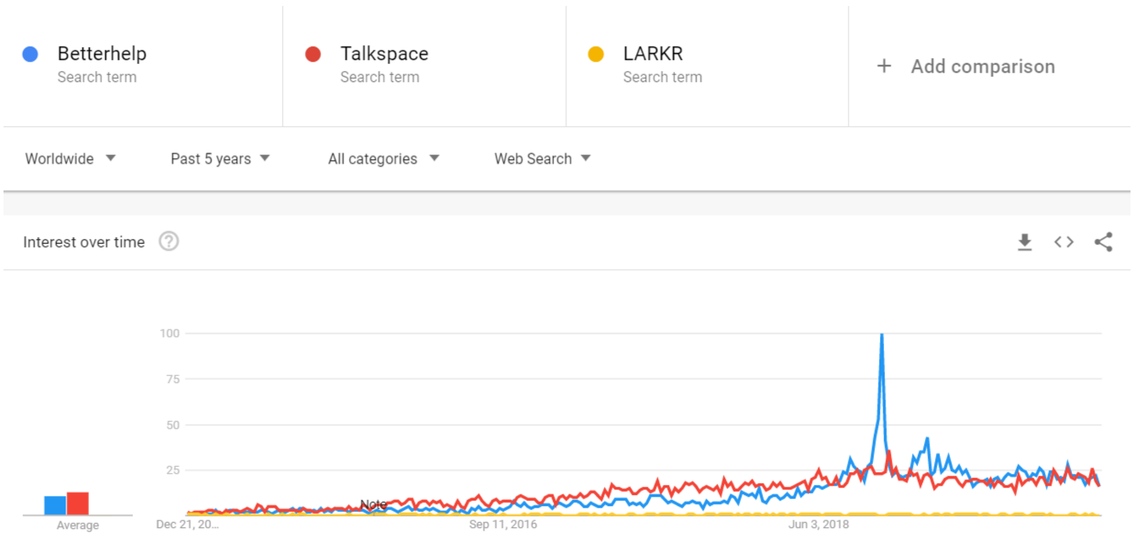


FIGURE 2.4: Google Trends comparison of Betterhelp, Talkspace and LARKR ("Betterhelp, Talkspace, LARKR - Google Trends", n.d.)

From the three apps the two most popular are Betterhelp and Talkspace which have more or less the same amount of interest and LARKR with substancionally less interest as shown in the figure 2.4 above.

Betterhelp

BetterHelp is an online counseling platform available worldwide that provides access to a licensed therapists. This service is not free, it ranges from $40 to $70 per week and it provides various methods of communicating with a counselor, such as exchanging messages, live chatting, phone call or video call. (*BetterHelp Website*, n.d.).

The *BetterHelp Website*, n.d. affirms that "Counselors on BetterHelp are licensed, trained, experienced, and accredited psychologists (PhD/PsyD), marriage and family therapists (LMFT), clinical social workers (LCSW/LMSW), or licensed professional counselors (LPC). All of them have a Masters Degree or a Doctorate Degree in their field. They have been qualified and certified by their state's professional board after successfully completing the necessary education, exams, training and practice. While their experience, expertise and background vary, they all possess at least 3 years and 2,000 hours of hands-on experience."

A study from Marcelle and Davis, 2017 supports the claim that online counseling platforms (BetterHelp in this case) can significantly reduce depression symptoms. BetterHelp users depression symptoms were evaluated using the Patient Health Questionnaire-9 (PHQ-9) industry standard of clinically verified depression assessment and the results showed 78% of BetterHelp members classified as having Severe Depression before using BetterHelp were no longer classified as having Severe Depression after use.

Talkspace

Similar to Betterhelp, Talkspace is an online counseling platform. For a subscription it allows users to send therapists text, audio, pictures and videos messages in a private, text-based chat room. To provide this service Talkspace requires the user to subscribe to paid plans that start at $65 per week and go up to $99 per week.

Talkspace have an licensed therapists, which means they don't use interns or therapists who are not fully licensed and still under supervision. All Talkspace therapists have over 3,000 hours of clinical experience. (*Talkspace Website*, 2018).

LARKR

LARKR is available for iOS and Android, unlike Betterhelp and Talkspace, this app solely focuses on communicating with therapists by video, emulating the classic therapy experience ("LARKR On-Demand Mental Health", n.d.). Each 50 minute video session costs 85$ and features a licensed therapist chosen by the user from a list of therapists that match the answers of a set of questions responded by the user in the beginning. It's also possible to cancel appointments or change therapist. ("Home Page", n.d.).

2.5.2 Cognitive Behavioral Therapy (CBT)

CBT is an effective combination of talk therapy and behavioral therapy. You work with a psychotherapist or therapist in a structured way, attending a limited number of sessions. This psychotherapy trains patients to re-frame negative thinking patterns into positive thoughts. Transforming ones thoughts will ultimately result in positive actions and behaviors which allows patients to react more effectively to challenging situations. ("Cognitive Behavioral Therapy - Mayo Clinic", n.d.)("What Is CBT?", n.d.)

Mindshift CBT

This application's target is adolescents, teens, and young adults and it has the objective of teaching about anxiety, helping users engage in healthy thinking and to take action. Users check in each day to track their anxiety and work with CBT-tools in the app. It gives the users opportunity to acquire basic skills to manage their symptoms of anxiety disorders, including GAD, specific phobias, and panic attacks. Its also useful for managing worry, social anxiety, and perfectionism (*MindShift ADAA Review*, n.d.)(*Anxiety Canada - MindShift CBT*, n.d.).

This application provides lists of active coping strategies tailored to the type of anxiety reported by the user and a variety of methods to manage anxiety. "Chill out" tools such as breathing exercises, mental imagery, and mindfulness strategies are provided in text and audio. "Quick Tips" are included to assist with anxiety in the moment (*MindShift ADAA Review*, n.d.).

Anxiety Coach

Anxiety Coach, developed by Mayo Clinic, is a paid self-help iOS app designed to reduce wide variety of fears and worries from shyness to compulsions and obsessions. With CBT strategies, it claims to help users manage, track, make plans and view progress to better deal with anxiety ("AnxietyCoach", n.d.). This application was developed by two clinical psychologists known in the area of anxiety disorders. Dr. Stephen P. H. Whiteside is the director of the Pediatric Anxiety Disorders Program at Mayo Clinic. Dr. Jonathan Abramowitz is an internationally known expert on adult anxiety disorders at the University of North Carolina.

2.5.3 Mindfullness

Mindfulness is the quality of being present and fully engaged with whatever oneself is doing at the moment — free from distraction or judgment, and aware of one's thoughts and feelings without getting caught up in them. It trains this moment-to-moment awareness through meditation, allowing to build the skill of mindfulness so that one can then apply it to their everyday life. By teaching the mind to be present, it's teaching to live more mindfully — in the present, taking a breath, not beholden to reactive thoughts and feelings — which is particularly helpful when faced with challenging circumstances or difficult situations. ("The Science-Backed Benefits of Mindfulness", n.d.)

Headspace

Headspace is one of the most popular apps that provide users with themed sessions that are focused on stress, sleep, focus and anxiety. It is a free mobile application available for iOS and Android devices and has over ten million downloads only on Google Play ("Headspace Google Play Page", n.d.). While free to download only some sessions are free while others would require the user to subscribe to the Headspace Plus plan which gives the user full access to the library of sessions. ("Headspace Website", n.d.).

Economides et al., 2018 performed a study with the headspace app which results suggest that mindfullness training is beneficial to stress and that smartphones are a an effective medium for mindfulness training. This study is backed up by other studies which show that mindfullness-based interventions have demonstrated significant beneficial impact on stress, anxiety, depression, and well-being (Spijkerman et al., 2016) (Khoury et al., 2015).

2.5.4 Others

IntelliCare

IntelliCare is a suite of apps, it contains various apps that among others some are Worry Knot, Boost Me, Social Force, iCope, Purple Chill and MoveMe which focus on worrying, stress, social interaction, inspirational messages, meditation and mood, respectively.

Mohr et al., 2017 performed a study with the objective of evaluating the efficiency of IntelliCare on reducing symptoms of depression and anxiety and the results concluded that it showed substantial reductions in the outcomes of PHQ-9 for depression and the Generalized Anxiety Disorder 7 Questionnaire (GAD-7) for anxiety.

2.6 Methods, Techniques and Technologies

In this section of the state of the art a number of technologies that are relevant to the project will be presented. Technologies for mobile and web applications will be shown to give the reader a better understanding on what are the current technologies in these fields.

2.6.1 Mobile applications

In this part it will be shown cross-platform mobile app development frameworks that allow applications to be available in different platforms such as iOS and Android either as a native or hybrid app[1].

React Native

React Native is a Javascript-based, open-source cross-platform app development framework designed and first lauched in 2013 by Facebook for building mobile apps. It makes use of actual Android or iOS components when building user interfaces. It has a syntax extension to JavaScript, called Javascript-XML (JSX) used for building the User Interface (UI) (*Introducing JSX - React*, n.d.). React Native then invokes the native rendering APIs in the platform-specific language i.e Swift (iOS) and Java (Android) to render the app on the screen (DashMagazine, 2019).

React Native has become one of the most popular cross-platform development frameworks, here it will be presented some characteristics that set it apart from other frameworks:

- Hot Reloading - Allows developers to immediately see the changes made in the code on the phone (Bigio, 2016).
- High number of libraries/third-party packages that can be used for development.
- React Native apps are fast and show similar performance when compared to native applications such as Swift applications on iOS (Calderaio, 2017)(Chrzanowska, 2017).

[1]Similarly to native apps, hybrid apps run on the device. Hybrid apps are written with web technologies (e.g., HTML5, CSS and JavaScript). Hybrid apps run inside a native container, and leverage the devices browser engine (but not the browser) to render the HTML and process the JavaScript locally. A web-to-native abstraction layer enables access to device capabilities that are not accessible in Mobile Web applications, such as the accelerometer, camera and local storage. (Cowart, n.d.)

Ionic

Ionic is a free, open-source, cross-platform framework that allows developers to build mobile and desktop apps using web technologies (HTML, CSS and JavaScript)(*What Is Ionic Framework?*, 2019). The first release was in 2013 and it was built on top of AngularJS. With the release of Ionic 4 in 2019 the framework was re-worked to be based on Web Components, allowing the user to choose any user interface framework, such as Angular, React or Vue. (Lynch, 2019).

Xamarin

In 2016, Microsoft acquired Xamarin from its founders and set out to make it open source. Xamarin is a cross-platform framework for building applications in iOS, Android, and Windows with the use of the .NET framework (Britch & Johnson, 2019). Since it is based on C# it means that developers have to take a different approach to developing cross-platform apps. Unlike hybrid apps that make use of web technologies, it compiles down to native code for the individual platforms (DashMagazine, 2019).

According to DashMagazine, some of the advantages of Xamarin are:

- Xamarin apps can achieve near-native app performance.
- Xamarin allows you to create rich experiences using native UI elements.
- Xamarin also lets you share up to 90% of the codebase across all platforms with you only having to design the UI for each platform separately.
- You can design a uniform UI across all platforms using Xamarin.Forms.

Flutter

Flutter is an open-source created by Google Software Development Kit (SDK) for developing Android, iOS and Web applications. Flutter apps are written in the object-oriented Dart language, which set Flutter apart from other SDKs with the following Dart particularities (Google, 2019):

- Stateful hot reload - Flutter uses a Just-In-Time (JIT) compiler which allow most changes to source code to be reflected immediately in the running app without requiring a restart or any loss of state.
- Flutter avoids the need for a separate declarative layout language like JSX or XML, or separate visual interface builders, because Darts declarative, programmatic layout is easy to read and visualize.
- Due to features that are familiar to developers of both static and dynamic languages some developers might find it easy to learn the language.
- Dart makes it easier to create animations and transitions that run at 60fps or even 120fps (Google, n.d.). Dart can do object allocation and garbage collection without locks. And like JavaScript, Dart avoids preemptive scheduling and shared memory (and thus locks). Because Flutter applications are compiled to native code, they do not require a slow bridge between JavaScript and native.

Unlike hybrid apps that make use of web views, or the previously mentioned React Native apps that use native components, Flutter apps fully compile down to native code (DashMagazine, 2019).

2.6.2 Web applications

An application can be a website, making it accessible to anyone with a connection to an internet and with a device that has a browser, the key advantage of the use of web applications is that it's widely accessibly. In this subsection we'll focus on front-end frameworks for developing web applications.

Front-end or the client-side involves building of web pages and user interfaces for web-applications. It usually implements the structure, design, behavior, animations for everything a person can see on screen while using website or web application. The client-side is a visual part that helps users visualize and interact with this part of an application. Thanks to a variety of programming tools, this interaction is possible. Client-facing web apps are usually built using a combination of JavaScript or Typescript, HTML, and CSS.

Frontend technologies are packages with prewritten, standardized code organized in files and folders. They provide developers with a foundation of functional code to build on along with the ability to change the final design. As a consequence, it helps developers save time as they don't need to write from scratch every single piece of code. ("How to Choose a Technology Stack for Web App Development", 2019)

Angular

Angular is a front-end framework that focuses on building rich Single Page Applications (SPAs). Its a front-end framework able to build an entire client-side application. In the beginning Angular used Javascript, but later releases adopted Typescript, which is a superset of Javascript. Angulars key drawbacks are its size compared to other frameworks, and the fact its not Search Engine Optimization (SEO) friendly by nature, although it can be SEO optimized. Google developed angular, and many companies use it such as Microsoft, Autodesk, Apple, Adobe, Paypal and Google itself. (Goel, 2019)

React

React is not a framework, its a front-end library, but many developers consider it a framework and its usually compared in that context. React was the first to implement the component-based architecture that Angular and Vue, and many other frameworks started to adopt later on. Reacts virtual dom makes the dom-manipulation much faster, and its easy to pick up, thanks mainly to its JSX syntax. React can be used either on the server side or on the client side. It was developed and is maintained by Facebook, and Facebook and Instagram use it for example. (Goel, 2019)

Vue

Vue.js is a recent popular framework, it started as an individual project and quickly grew into becoming one of the most trending JS frameworks. Its a progressive framework, which means that if you have an existing project, you can adopt Vue for one portion of the project, and all would work well. It also brings offers the component architecture like React does, and the Vue ecosystem can help you build full front-end applications. Companies are starting to invest in Vue, companies like Facebook, Netflix, Adobe, Xiaomi and Gitlab trusted Vue and used it. (Goel, 2019)

Ember

Ember was named the best Javascript framework in 2015. Today, the Ember community is massive, and it is constantly expanding, with new features, and releases introduced regularly. Ember possesses the two-way data binding like Angular, and it provides a lot of features and components that you can use out of the box. This framework is often used by Google, Microsoft, Heroku, and Netflix. Ember revolves around the productivity of the developer and tries to maximize it either by eliminating the need for time-wasting activities or by adopting certain JS best practices in its core design. (Goel, 2019)

Node.Js

Node.Js is a lightweight and efficient JavaScript runtime environment on the server side, powered by the Chrome V8 JavaScript engine, that uses a non-blocking I/O model. Its event-driven model enables developers to create fast and scalable network applications. Some of the advantages of this technology are that it has a vast ecosystem of package modules thanks to its Node Package Manager (NPM), it was designed to read and manage large streams of data, it shares the same code on the client and on the server-side, and it enables developers to write real-time, server-side applications in JavaScript (Capan, 2013) (Gawron, 2018).

2.7 Summary

In this chapter concepts like recommendation systems and brief overviews about anxiety disorders and other mental illness were presented. After contextualizing the reader with these subjects it's then possible to better understand the project in general and the choices made during the analysis and design. Techniques and technologies that are relevant to the project were also presented as well as existing solutions of recommendation systems and applications for managing and helping anxiety disorders.

Chapter 3

Value Analysis

In 1961, in his book "Techniques of Value Analysis and Engineering", engineer Lawrence D. Miles stated: "Value analysis is a problem-solving system implemented by the use of a specific set of techniques, a body of knowledge, and a group of learned skills. It is an organized creative approach whose purpose is the efficient identification of unnecessary costs, i.e. cost that provides neither quality nor use nor tool life nor appearance nor customer features."

In summary, the value analysis has the objective of achieving an increased value with the lowest cost for a certain product or service, as such in the following sections the value for the customer and the perceived value will be analyzed and a value propostion and the business model canvas will be created.

3.1 Value for customer

The value for the customer is the value that the customer gives to the product or service according to his personal perception. The same product or service may have different values for different customers depending on their use and experience. A user who gets a good experience with a product or service tends to want to repeat the experience as well as tends to consider that the value of that product or service as higher.

Applying this concept to the project, the value for the customer is the alleviation of anxiety, learning how to better manage anxiety and being able to contact a therapist with the ease that a mobile phone offers.

3.2 Perceived value

Zeithaml, 1988 explained perceived value as "the consumers overall assessment of the utility of a product based on perceptions of what is received and what is given. This means that the perceived value is the result of the difference between benefits and sacrifices for the client.

Within the scope of the project, for people with anxiety it is possible to consider the benefits of the solution as: The mental health benefits of self managing anxiety and there's also the convenience of communicating with a therapist by using a mobile application. On the other hand, it is also possible to consider as sacrifices the effort that the user needs to learn on how to use the app and the time used for performing the exercises. Nevertheless the perceived value varies from customer to customer as the weights of benefits and sacrifices are measured in a personal way. Regarding caregivers and therapists, they have similar benefits and sacrifices, they're benefited by being able to help people with anxiety disorders and they would sacrifice time and effort to do so. These benefits and sacrifices are possible to observe on the table 3.1 below.

Benefits	**Sacrifices**
Users with anxiety	
Mental health benefits	Time
Convenience	Effort
Therapists and Caregivers	
Helping Users	Time
	Effort

Table 3.1: Benefits and Sacrifices

3.3 New Concept Development

The process of innovation goes through three key phases: the Fuzzy Front End, the New Product Development process, and commercialization. (P. Koen et al., 2001).

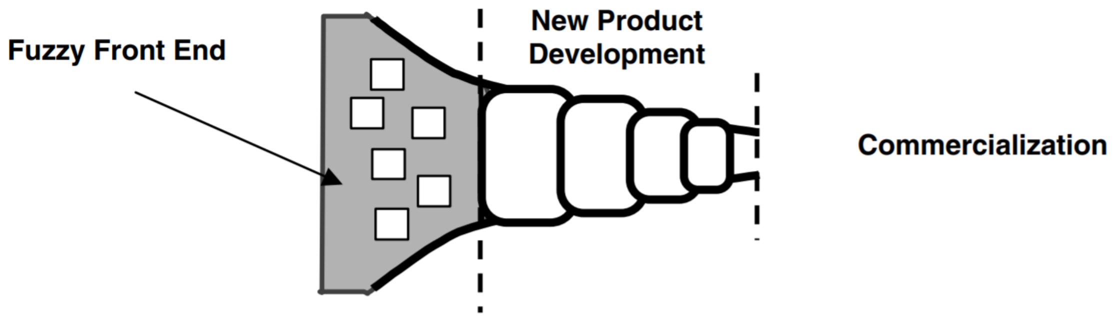


Figure 3.1: Innovation process

The Fuzzy Front End (FFE) provides the most opportunities for improvement of the overall innovation process, . After the FFE the more structured part of the process begins, the New Product Development (NPD).

In order to optimize the FFE, a model called New Concept Development was developed whose purpose is to provide a common language and definition of the key components of the FFE. (P. A. Koen et al., 2002)

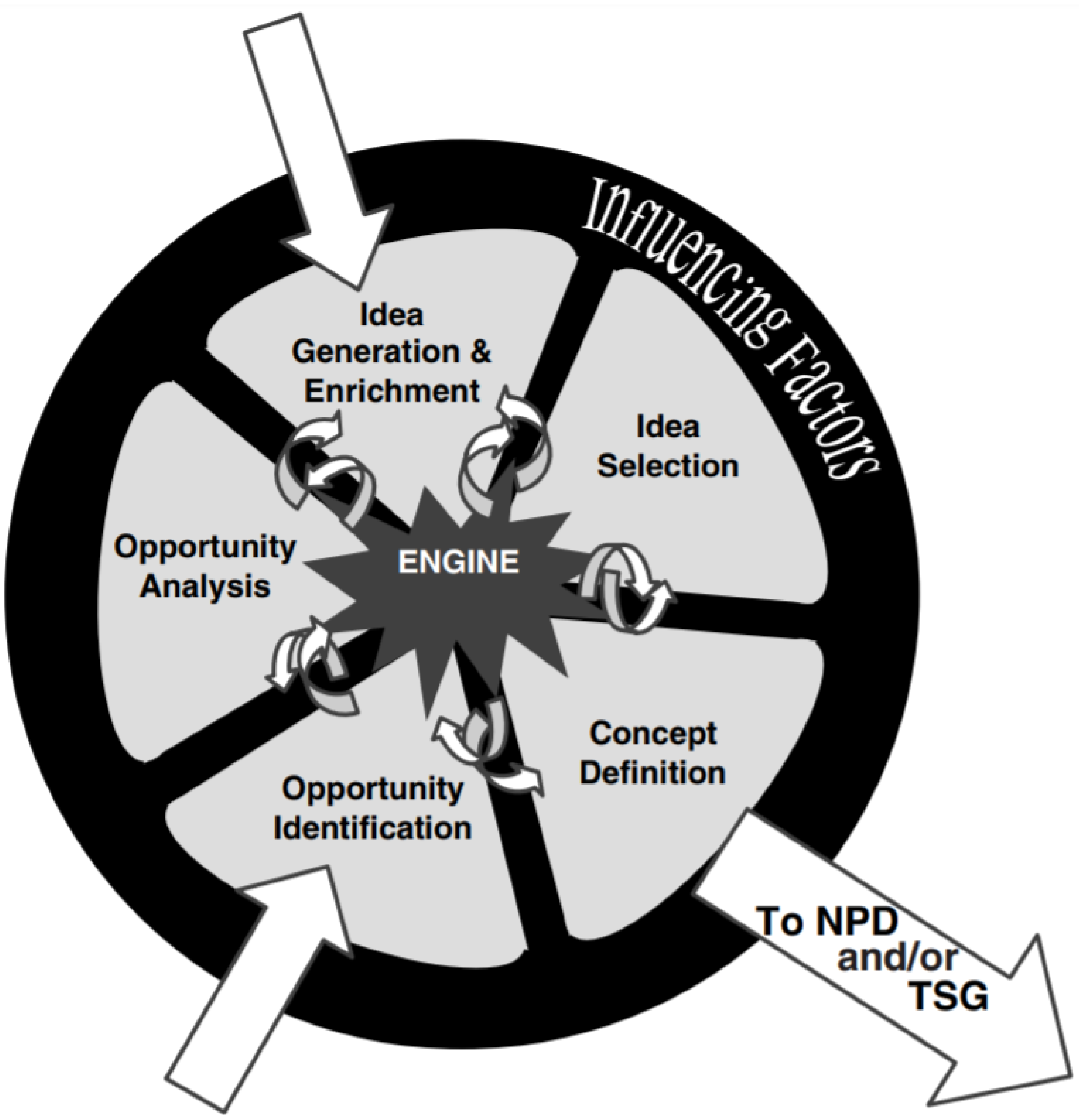


FIGURE 3.2: New Concept Development

The New Concept Development (NCD) model shown above in the Figure 3.2 consists of three key parts:

- The engine is the leadership, culture and business strategy of the organization that drives the five key elements. (P. A. Koen et al., 2002)
- The activity five elements (opportunity identification, opportunity analysis, idea generation and enrichment, idea selection and concept definition) of the FFE (P. A. Koen et al., 2002)
- The influencing factors consist of organization capabilities, the outside world and the enabling sciences that may be involved. These factors affect the innovation process from the FFE all the way to commercialization. These influencing factors are usually uncontrollable by the organization. (P. A. Koen et al., 2002)

In the figure 3.2 it's possible to see that it has a circular shape, it means to suggest that ideas and concepts are expected to iterate across the five elements. The arrows pointing into the model represent starting points and indicate that projects begin at either opportunity identification or idea generation and enrichment. The exiting arrow represents how concepts leave the model and enter the NPD. (P. A. Koen et al., 2002)

The five elements:

3.3.1 Opportunity Identification

In this element the organization identifies problems and opportunities that it might want to pursue. The Opportunity identification defines the market or technology that the organization may want to participate in. (P. A. Koen et al., 2002)

For this project, it was identified the opportunity of creating software to help people with anxiety disorders.

3.3.2 Opportunity Analysis

In the previous element the opportunity was identified but not all opportunities are worth pursuing, in this element the organization confirms if an opportunity is indeed worth pursuing. (P. A. Koen et al., 2002)

The opportunity identified has the potential of exploring new ideas that most existing anxiety-related software doesn't have (e.g. recommendation system or the integration of caregivers and therapists). If successfully done it would help those with anxiety disorders, as such this opportunity was deemed worth pursuing.

3.3.3 Idea Generation and Enrichment

The element of idea generation and enrichment concerns the birth, development, and maturation of a concrete idea. Idea generation is evolutionary. Ideas are built up, discarded, combined, reshaped, altered, and upgraded. An idea may go through many iterations and changes as it is evaluated, studied and discussed (P. A. Koen et al., 2002).

Some ideas for this project were generated, the first one that came to mind was create a mobile application. A second possible idea was to create a mobile application for users and caregivers but therapists have a website. And lastly, a third idea, to create a website.

3.3.4 Idea Selection

Ideas are generally easy to generate but it's hard to decide the best idea that will generate the most value at the lowest cost. In order to make this selection the Analytic Hierarchy Process (AHP) method will be used based on three factors/criteria: Time and Effort of Development;Anxiety Alleviation and Usability/Convenience. The execution of the AHP method can be seen below in the section 3.4. The result was that it is recommended to choose the idea of creating a mobile application.

3.3.5 Concept Definition

The final step of the NCD model, from this step it exits into the NPD and then Commercialization. Here the concept must be defined in more detail and usually a compelling case for investment must be made, also refered by organizations as a "win statement" (P. A. Koen et al., 2002). This concept was already explained in detail but in the section Value proposition below there's the equivalent of a "win statement".

3.4 Analytic Hierarchy Process

Introduced by Thomas Saaty, the AHP method is an efficient tool to deal with complex decisions, allowing to define priorities through qualitative as well as quantitative criteria. By reducing complex decisions in a series of paired comparisons, AHP helps capturing the objective and subjective aspects of a decision. Furthermore, the AHP method incorporates a useful technique to verify consistency in the evaluation of decisions, thus reducing the bias in the decision process (Saaty, 1980).

In the hierarchical process, as presented by Saaty, the following steps are present (Saaty, 1980):

- Definition of objectives, alternatives and criteria relevant to the problem of decision;
- Evaluation of alternatives in relation to the criteria and evaluation of the relative importance in relation to each criteria;
- Result of the overall assessment of each alternative, with all the elements hierarchically structured.

About obtaining priorities in the AHP method, a relationship is established between the elements of each level of hierarchy with equal comparison of criteria or sub-criteria, that is, the comparison between attributes occurs at the same level as decision structure. The values determined according to the scale proposed by Thomas Saaty that ranges from 1 to 9, where 1 represents a comparison where the elements are equally important, 5 points to great or essential importance and 9 indicates absolute importance value of one element in comparison to other (Saaty, 1980).

Using logical consistency, the AHP method incorporates both quantitative and qualitative aspects of human thought, this being the fact of defining the problem, the hierarchy and the quantitative aspects to express preferences effectively (Saaty, 1980).

3.4.1 Phase 1 — Construction of the hierarchical decision tree

At this stage, the construction of the hierarchical decision tree is performed and within there will be the objectives, criteria and viable alternatives. In order to choose the best project idea, through a tree hierarchical, it is necessary that the criteria are well structured. Figure 3.3 shows the tree structured in order to select the best business idea.

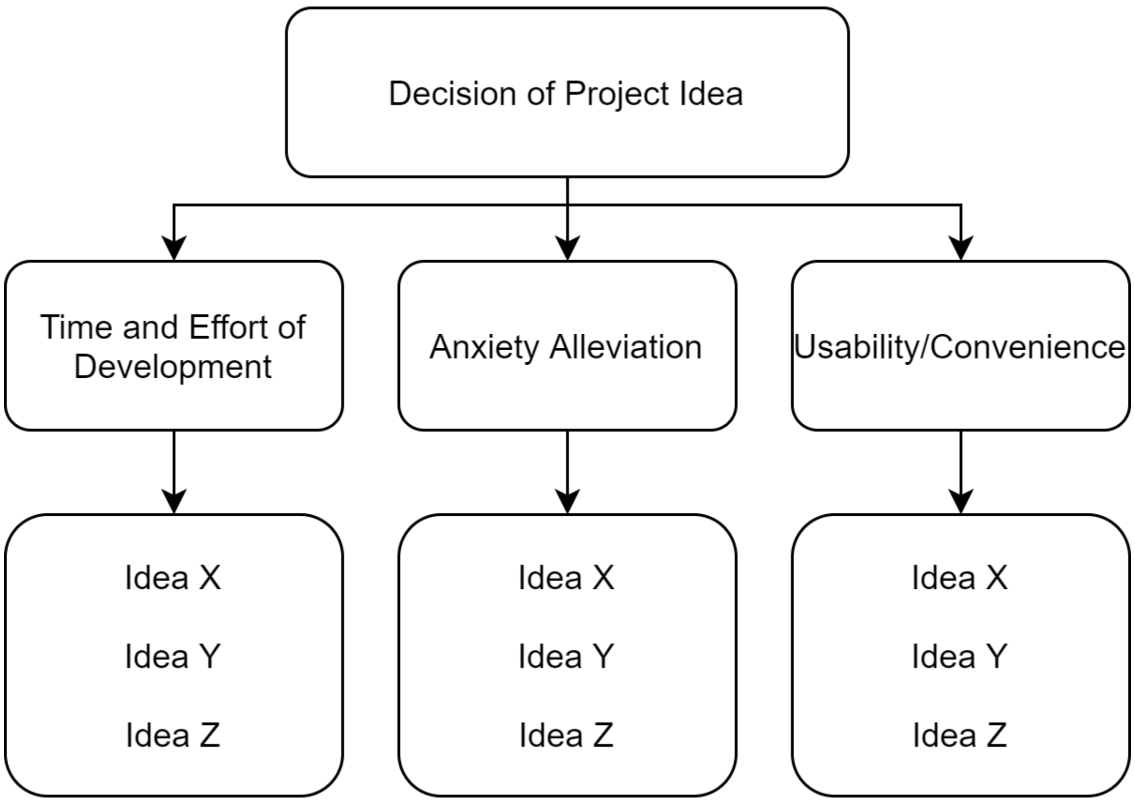


FIGURE 3.3: Hierarchical decision tree divided in three criteria

The criteria used (Time and Effort of Development, Anxiety Alleviation and Usability/Convenience) was chosen in order to help the decision of one of three ideas initially generated.

- **Time and Effort of Development** — Like any other project, the time and effort of development is important, a long development time is undesired. Short time and low effort of development will allow time and effort to be directed to other elements that aren't crucial such as testing and non-functional requirements and other minor functionalities.
- **Anxiety Alleviation** — The software needs to have the potential of alleviating the user's anxiety. For example, an application for alleviating anxiety in a smart fridge built in screen doesn't have the same potential of alleviating anxiety as a mobile application because it's not available everywhere and there's probably no internet connection and no sound.
- **Usability/Convenience** — Two of the key characteristics of this project, it needs to have a good usability and must be convinient to the user otherwise he won't use the application as much as it should be used. In order to the application help the user it needs his participation, it needs the user to talk to the therapist, to report symptoms and perform the exercises recommended.

3.4.2 Phase 2 — Comparison of criteria

In this phase, the priorities between elements of each level of the hierarchy are established through the use of a comparison matrix.

The key point to be considered is the determination of a scale of values for this comparison, which should not exceed a total of nine factors, in order to maintain the matrix consistent. This scale, defined by Thomas Saaty, is called the Fundamental Scale and can be seen in more detail in the table 3.2 below (Saaty, 1980).

Importance level	**Definition**	**Description**
1	Equal importance	The two activities contribute equally to the objective
3	Moderate Importance	Experience and judgement slightly favor one activity over another
5	Strong importance	Experience and judgement strongly favor one activity over another
7	Very strong importance	One activity is strongly favored over another; it's dominance demonstrated in practice
9	Extreme importance	The evidence is in favor of one activity over another, to the greatest extend possible
2,4,6,8	Intermediate values	Used between the importance levels above

Table 3.2: Fundamental Scale

For this particular problem where the objective is to choose one of three ideas, using the criteria defined in the hierarchical division, the following comparison matrix was developed.

Criteria	**Time and Effort of Development**	**Anxiety Alleviation**	**Usability/Convenience**
Time and Effort of Development	1	1/2	1/3
Anxiety Alleviation	2	1	1/3
Usability/Convenience	3	3	1

Table 3.3: Criteria comparison matrix

3.4.3 Phase 3 — Relative priority of each criteria

After some calculations, the results the relative priority of each criteria can be seen below in the table 3.4.

Criteria	**Relative Priority**
Time and Effort of Development	0,16
Anxiety Alleviation	0,25
Usability/Convenience	0,59

Table 3.4: Criteria Relative Priority

The criteria Usability/Convenience is the most important followed by Anxiety Alleviation and lastly Time and Effort of Development.

3.4.4 Phase 4 — Evaluating the consistence of the relative priorities

The point of this phase is to calculate the Consistency Ratio (CR) to see if the criteria has been consistent in comparison to large samples of completely random judgments

To perform the CR calculation, it is necessary to first obtain the Consistency Index (CI) and the Random Index (RI), demonstrated in the following formula.

$$CR = \frac{CI}{RI}$$

The CI is obtained from the following equation:

$$CI = \frac{\lambda_{max} - n}{n - 1}$$

The n is the number of criteria and the λ_{max} can be obtained from the calculations below.

$$\begin{bmatrix} 1 & 1/2 & 1/3 \\ 2 & 1 & 1/3 \\ 3 & 3 & 1 \end{bmatrix} \times \begin{bmatrix} 0,16 \\ 0,25 \\ 0,59 \end{bmatrix} = \begin{bmatrix} 3.02 \\ 3.04 \\ 3.10 \end{bmatrix}$$

$$\lambda_{max} = average \left\{ \frac{3{,}02}{0{,}16} \quad \frac{3.04}{0.25} \quad \frac{3.10}{0.59} \right\} = 3,05$$

After determining the λ_{max} value, the CI can be calculated:

$$CI = \frac{3.05 - 3}{3 - 1} = 0,025$$

Finally, the Consistency Ratio will be calculated by obtaining the Random Index corresponding to $n = 3$ from the Thomas Saaty index table 3.5 below.

N	1	2	3	4	5	6	7	8	9	10
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

TABLE 3.5: Random Index Table

Now that we have the values of the CI and RI we can calculate the CR:

$$CR = \frac{0,025}{0,58} \approx 0,04$$

In conclusion, the Consistency Ratio, whose value is ~ 0.04 and less than 0.10, we can say that the values assigned to relative priorities are consistent and reliable.

3.4.5 Phase 5 — Comparison Matrix for each criteria

The ideas generated in the section 3.3.3 are as follows:

- **X** — Mobile application.
- **Y** — Mobile application for users and caregivers but therapists have a website.
- **Z** — Website.

This ideas will be analyzed and attributed a value from the fundamental scale for each criteria.

In the table 3.6 below is a comparison matrix for the criteria Time and Effort of Development. The idea Y is the only one with a low result because the time and effort to develop for two platforms is significantly higher than the time and effort needed to develop for just one platform, either mobile or web.

	X	**Y**	**Z**	**Relative Priority**
X	1	3	1	0,42
Y	1/3	1	1/2	0,14
Z	1	3	1	0,42

TABLE 3.6: Time and Effort of Development Comparison Matrix

In the table 3.7 below is a comparison matrix for the criteria Anxiety Alleviation. The idea Z scored worse than the other two because the potencial of alleviating anxiety is higher on a mobile device because the user can interact with the application everywhere, if the user has a panic attack when he's out of the house and has no connection to the internet he can still use the application while to do offline exercises.

	X	**Y**	**Z**	**Relative Priority**
X	1	1	3	0,42
Y	1	1	3	0,42
Z	1/3	1/3	1	0,14

TABLE 3.7: Anxiety Alleviation Comparison Matrix

In the table 3.8 below is a comparison matrix for the criteria Usability/Convenience. Similarly to the previous criteria, X and Y scored better than Z. This is because a mobile application have far better usability (Nielsen, 2012) and is more convenient for users due to the fact that it's possible to use offline and it's quicker to launch an app than to type in the website address on the browser.

	X	**Y**	**Z**	**Relative Priority**
X	1	1	5	0,45
Y	1	1	5	0,45
Z	1/5	1/5	1	0,09

TABLE 3.8: Usability/Convenience Comparison Matrix

To conclude, an updated diagram was built, as it is possible to see in the figure 3.4. It summarizes the entire information obtained and allows the preparation of the next phase of the AHP.

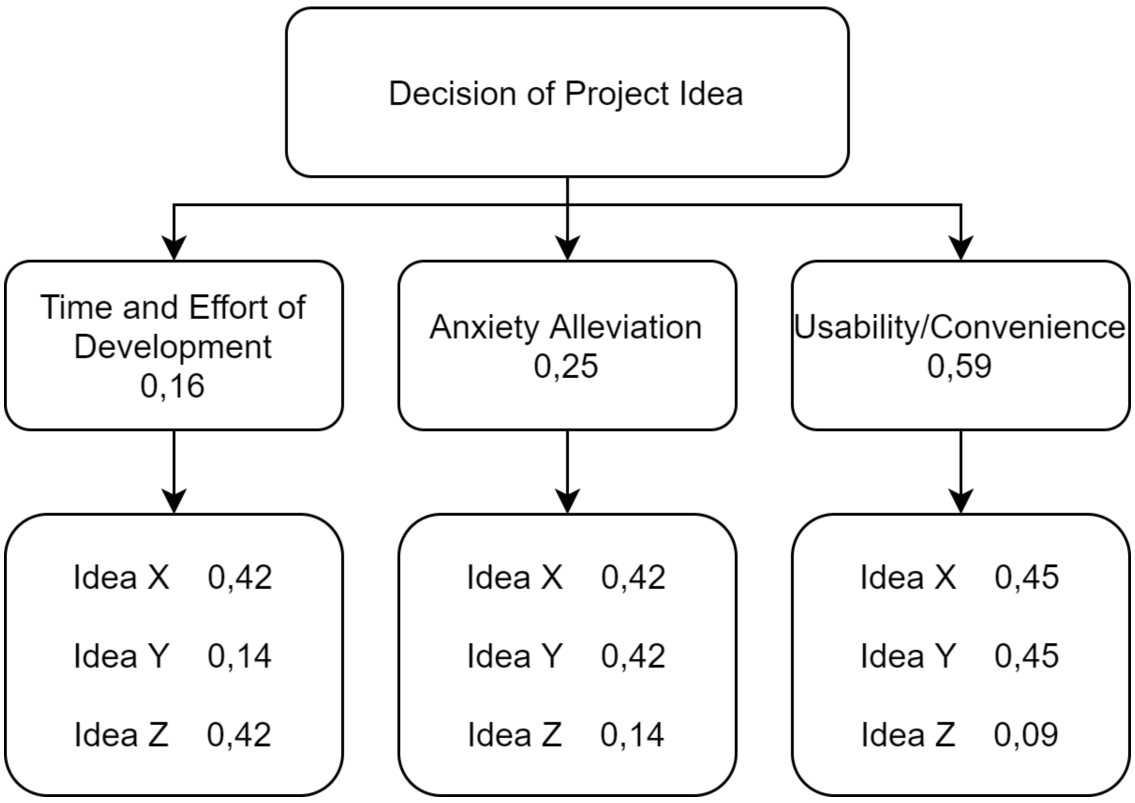


FIGURE 3.4: Updated hierarchical decision tree divided in three criteria

3.4.6 Phase 6 — Priority for each idea

This is the last step that has calculations. The values obtained from the priority matrix calculated in phase 5, when multiplied with the priority vector of the criteria results in the final values of each idea. The highest value being the idea that is recommended to be chosen.

$$\begin{bmatrix} 0,42 & 0,42 & 0,45 \\ 0,14 & 0,42 & 0,45 \\ 0,42 & 0,14 & 0,09 \end{bmatrix} \times \begin{bmatrix} 0,16 \\ 0,25 \\ 0,59 \end{bmatrix} = \begin{bmatrix} 0,43 \\ 0,39 \\ 0,15 \end{bmatrix}$$

3.4.7 Phase 7 — Idea selection

In short, observing the values obtained in the previous phase and taking into account the criteria defined and their respective importance, we can say that option X is the best idea for the opportunity discussed in the section 3.3.1. Using this analytical hierarchy method, the choice of this project has the final weight of 0,43 and it is to create a mobile application.

3.5 Value proposition

Osterwalder, 2004 said "A Value Proposition is an overall view of a company's bundle of products and services that are of value to the customer."

In the case of this dissertation, instead of a company's bundle of products and services the focus will be on the project's features that are of value to the user.

A Value proposition has to objective of showing the users why they should acquire or use a certain product or service by showing them the benefits that they'll receive by using said product or service.

In conclusion, the value proposition of WEBECOOL is: A hybrid mobile app that provides anxiety self-management for those with anxiety disorders by including communication between the user, therapist and caregiver, and exercises that alleviate anxiety symptoms that are suggested to the user by a recommendation system.

3.6 Business Model Canvas

A business model described how a organization created and captures value. The Business Model Canvas is a simple graphic model but it helps focusing on the more crucial aspects of the organization business model. It can be divided in four main sections, the left one is about the infrastructure, the middle one shows what can be offered to the customer, the right one is related to the customers and lastly, the bottom one is about the finances. However, all nine blocks are related and communicate with each other. (Osterwalder et al., 2010)

- **Value Propositions** — The value proposition explains what are the benefits that would motivate a client to use a certain product or service. In the previous section 3.5 the value proposition was already created and is now being added to the canvas model.
- **Key Partners** — All of those who will contribute to this project. GECAD proposed this project and will assist the development of it.
- **Key Activities** — The most relevant activities in executing the value proposition, that is, researching and developing the desired solution.
- **Key Resources** — Resources or assets that are necessary to create value. This project needs hardware and software so that development is possible, it also needs servers for databases or recommendation system algorithms.
- **Cost Structure** — The intrinsic costs of the project, mainly related to the key resources, namely, the cost of development and maintenance of servers.
- **Customer Relationship** — How the users will interact with the project development, during the development the application will be used by third parties that can give feedback on it. Direct feedback can also be possible through e-mail.
- **Channels** — How the app will be distributed to users, since it's a mobile application then it should be available on the mobile app stores. It is possible that some health care institutions will recommend and distribute the application to it's users.
- **Customer Segments** — The main customer segment is people with anxiety disorders, those are the users that will benefit the most of this application but there are other relevant segments such as the caregivers that can assist the anxiety management of the users and there's the therapist who can communicate and help the user.
- **Revenue Streams** — The purpose of the WEBECOOL is not tied to revenue and profit, instead of economic causes it is focused on social causes, specifically mental health and anxiety self-management. As such there are no revenue streams.

Key Partners	Key Activities	Value Propositions	Customer Relationship	Customer Segments
▪ GECAD	▪ Development of WEBECOOL ▪ Research	A hybrid mobile app that provides anxiety self-management for those with anxiety disorders by including communication between the patient, therapist and caregiver, and exercices or procedures that alleviate anxiety symptoms that are suggested to the user by a recommendation system.	▪ Testing ▪ Email	▪ Patients with anxiety disorder ▪ Caregivers ▪ Therapists
	Key Resources		Channels	
	▪ Hardware and software for development ▪ Server		▪ Mobile app stores ▪ Health care institutions	
Cost Structure			Revenue Streams	
▪ Development ▪ Maintenance				

TABLE 3.9: Canvas Model

3.7 Summary

In this chapter the value analysis was performed. The value for the customer and the perceived value were discussed and a value proposition and business model canvas were created.

Chapter 4

Design

All good software goes through the process of analysis and design, it is the process of preparing the plan for a software application while satisfying a problems functional requirements and not violating its non-functional constraints. It defines software methods, functions, objects, and the overall structure and interaction of the code so that the resulting functionality satisfies the requirements.

4.1 System actors

System actors are external entities that interact with the application, these entities can be individuals or systems (Keene, n.d.). In this project only individuals were identified:

- **Non-authenticated user:** Is not registered or hasn't logged in. This actor has a limited set of features.
- **User (Patient):** Is the main user of the application, this user is someone who wants to use the application to relieve his anxiety.
- **Caregiver:** Is chosen by the user, this actor can help the user and can also provide insight to a therapist about the user.
- **Therapist:** Is responsible for helping the user improve his mental health with features such as a chat or recommending exercises for example.

4.2 Functional and non functional requirements

The main purpose of this section is to capture and present the functional and non-functional requirements of this project. The requirements will be classified in the different categories of the FURPS+ requirements classification system. This system is an evolution of the FURPS model that was developed by Robert Grady at Hewlett-Packard (HP) and divides the requirements into the following categories: Functionality, Usability, Reliability, Performance and Supportability (Eeles, 2005).

The requirements of the Functionality category are part of the functional requirements. The requirements of the remaining categories correspond to non-functional requirements. In the evolution from FURPS to FURPS+, the '+' sign was added, which aims to identify additional categories belonging to non-functional requirements that generally represent restrictions or limitations of the system (Eeles, 2004).

	FURPS	**+**
Functional Requirements	**F**unctionality	
Non-functional Requirements	**U**sability	Design constraints
	Reliability	Implementation constraints
	Performance	Interface constraints
	Supportability	Physical constraints

TABLE 4.1: FURPS+ classification system

Functional requirements describe what the system should be capable of, that is, the functionalities of the software. Non-functional requirements describe how a system should work, thus representing the technical conditions for carrying out the project with the objective of achieving user satisfaction.

To carry out the requirements engineering of this project, the use of the FURPS+ classification system was adopted due to its worldwide recognition in software development.

4.2.1 Functional requirements

There's only one category of functional requirements within the FURPS+ classification system, the functionality category.

The system will have various functionalities and some may be only applicable to some user roles (Patient, Caregiver or Therapist) instead of being available for all users. In the Figures 4.1, 4.2 and 4.3 represented below we can see an overview of the actors and the use cases in the application.

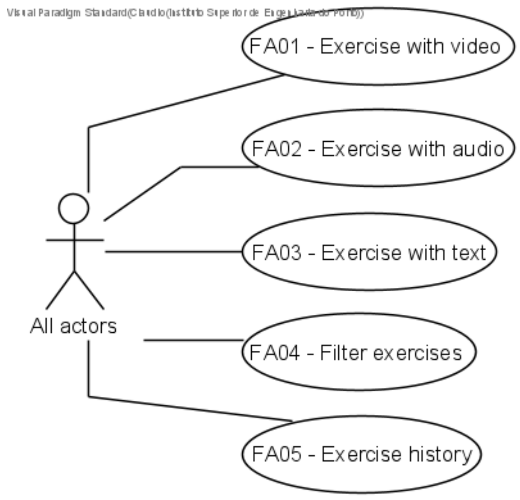

All actors
FA01 - Exercise with video
FA02 - Exercise with audio
FA03 - Exercise with text
FA04 - Filter exercises
FA05 - Exercise history

FIGURE 4.1: Use Case Diagram (Part 1)

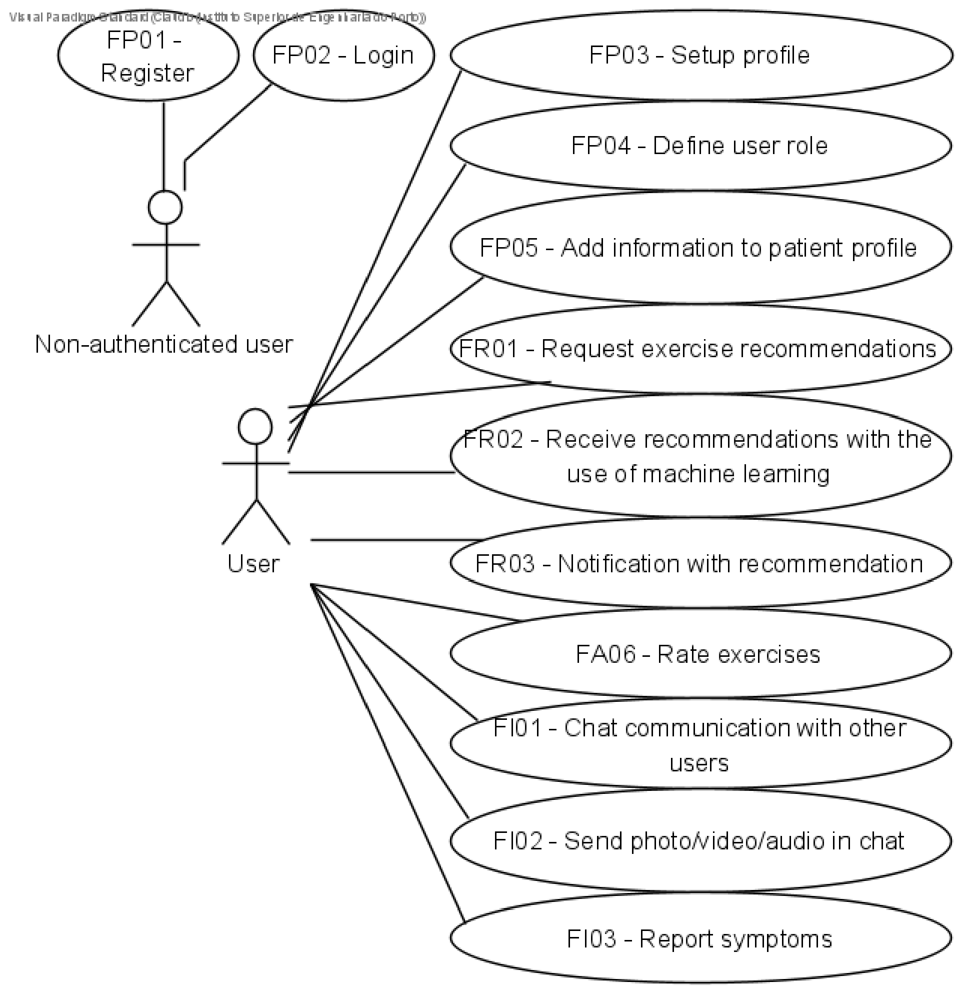

Visual Paradigm Standard (Cláudio (Instituto Superior de Engenharia do Porto))
FP01 - Register
FP02 - Login
FP03 - Setup profile
FP04 - Define user role
FP05 - Add information to patient profile
FR01 - Request exercise recommendations
FR02 - Receive recommendations with the use of machine learning
FR03 - Notification with recommendation
FA06 - Rate exercises
FI01 - Chat communication with other users
FI02 - Send photo/video/audio in chat
FI03 - Report symptoms
Non-authenticated user
User

FIGURE 4.2: Use Case Diagram (Part 2)

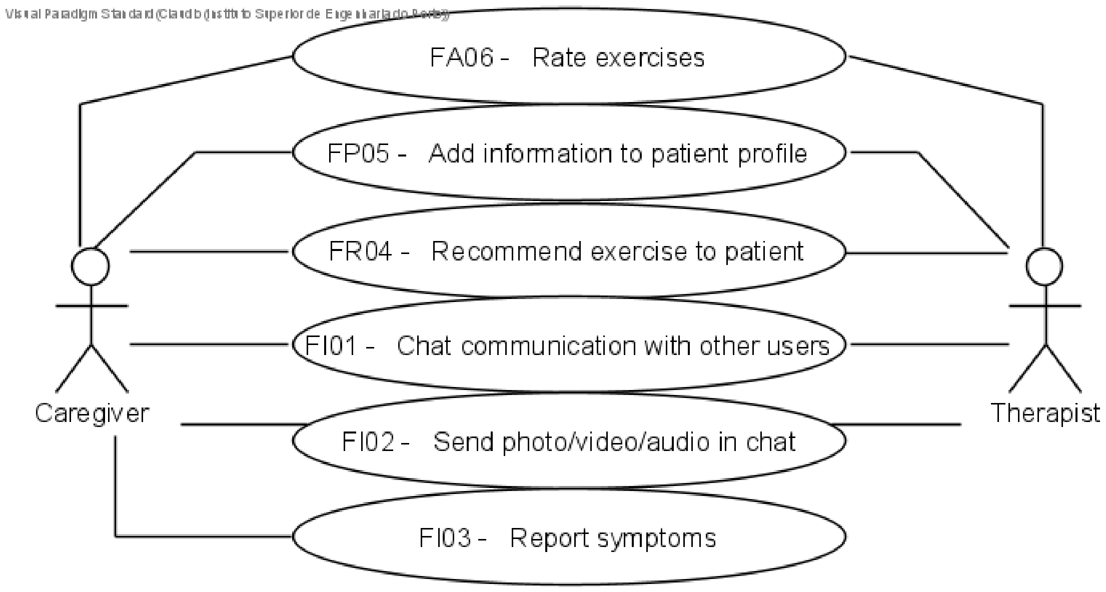

Visual Paradigm Standard (Cláudio (Instituto Superior de Engenharia do Porto))
FA06 - Rate exercises
FP05 - Add information to patient profile
FR04 - Recommend exercise to patient
FI01 - Chat communication with other users
FI02 - Send photo/video/audio in chat
FI03 - Report symptoms
Caregiver
Therapist

FIGURE 4.3: Use Case Diagram (Part 3)

Below, in the table 4.2 we can see all the use cases seen in the figures above with their corresponding priority. These requirements have an unique identifier that originates from the QEF that is present on the Evaluation chapter.

Use cases	Actors	Priority
FR01 - Request exercise recommendations	User	5
FR02 - Receive recommendations with the use of machine learning	User	5
FR03 - Notification with recommendation	User	4
FR04 - Recommend exercise to patient	Caregiver, Therapist	3
FA01 - Exercise with video	All actors	5
FA02 - Exercise with audio	All actors	5
FA03 - Exercise with text	All actors	5
FA04 - Filter exercises	All actors	4
FA05 - Exercise history	All actors	5
FA06 - Rate exercises	User, Caregiver, Therapist	5
FI01 - Chat communication with other users	User, Caregiver, Therapist	3
FI02 - Send photo/video/audio in chat	User, Caregiver, Therapist	1
FI03 - Report symptoms	User, Caregiver	3
FP01 - Register	Non-authenticated user	5
FP02 - Login	Non-authenticated user	5
FP03 - Setup profile	User	5
FP04 - Define user role	User	3
FP05 - Add information to patient profile	User, Caregiver, Therapist	2

TABLE 4.2: Use cases

FR01 - Request exercise recommendations

The user can request various exercise recommendations. Here, the algorithm will recommend certain exercises taking into account ratings by the user and other users.

FR02 - Receive recommendations with the use of machine learning

In the homepage of the application there should be an exercise recommendation. This recommendation must use machine learning.

FR03 - Notification with recommendation

The application can create a notification for the user with a recommendation.

FR04 - Recommend exercise to patient

The caregiver and therapist actors shall be able to recommend specific exercises to patients.

FA01 - Exercise with video

The application should contain exercises in the video format. This format will mostly contain videos of mindfulness meditation.

FA02 - Exercise with audio

The application should contain exercises in the audio format. Similar to video exercises but only in audio form.

FA03 - Exercise with text

The application should contain exercises in the text format. This means that the user can read about anxiety and ways to relieve it.

FA04 - Filter exercises

The user should be able to filter exercises. This means that the user could choose to only see audio exercises if he wishes to.

FA05 - Exercise history

The user should be able to check his exercise history that will contain the list of exercises previously done, when they were performed and how much the user rated them.

FA06 - Rate exercises

After doing an exercise the user should be able to rate that exercise between one and five stars.

FI01 - Chat communication with other users

The application should provide a chat between users and caregivers and therapists. Each user could have a chat with his caregiver or his therapist.

FI02 - Send photo/video/audio in chat

In the chat previously mentioned the user should be able to send photos, video or audio.

FI03 - Report symptoms

Day-to-day log in which the user can report symptoms by describing an anxiety experience or an event that triggered a social problem. This feature works as a record of user mood over time and allows the review of past entries.

FP01 - Register

The application should provide an option for the user to create an account.

FP02 - Login

With a registered account, a user can login into that account and have all profile data accessible.

FP03 - Setup profile

The user should be able to setup profile information like age, gender and what type of anxiety he has. This information will then provide more accurate recommendations.

FP04 - Define user role

The user should be able to register as User, Caregiver or Therapist.

FP05 - Add information to patient profile

The user, caregiver and therapist should be able to add information to a user profile.

4.2.2 Non-functional requirements

In the FURPS+ classification system there are several categories that are part of the non-functional requirements: usability, reliability, performance, supportability and constraints. In software development, non-functional requirements are also important, since they ensure the quality of the system, thus obtaining high user satisfaction.

Usability

Usability is related to the graphical interface that's presented to the user and how it behaves when interacted. Requirements such as accessibility, aesthetics and consistency belong inside the usability requirement (Eeles, 2004). In the context of this project, different usability requirements were identified, namely:

- textbfIntuitive interface: the application should have an easy to understand interface with good navigation;
- **Appropriate content:** all content inside the app should be easy to understand and related to mental health.

Reliability

Reliability includes concepts like availability, accuracy, and recoverability, for example, recoverability of the system from shut-down failure (Eeles, 2004). For this application, the following requirements were identified:

- **Clear errors:** In event of failure, show clear error to the user;
- **Offline mode:** If a server is unavailable, some app features should still be possible to use.

Performance

Performance involves things such as system response time, recovery time, and startup time (Eeles, 2004). This project has the following performance requirements:

- **Quick start-up time:** the application shouldn't take too much time to start;
- **Responsiveness:** all user interaction should have immediate feedback.

Supportability

The last letter of the Functionality Usability Reliability Performance Supportability (FURPS)+ acronym, supportability, where it is specified a number of other requirements such as testability, adaptability, maintainability, compatibility, configurability, installability, scalability and localizability (Eeles, 2004). This project contains the following supportability requirements:

- **Scalable backend:** the server resources should be capable of growing;
- **Cross-platform:** the system should allow the implementation in multiple platforms;
- **Testability:** the system should be tested in order to be free of errors.

Design constraints

As implied by the name, design constraints limit the design of the software (e.g. required to use a relational database) (Eeles, 2004). There isn't any major design constraint for this project.

Implementation constraints

Implementation constraints is everything that limits the process of coding or implementing (e.g. required standards, platform or implementation language) (Eeles, 2004). One of the requirements of this project is that it should be available for android and iOS, because of that there's one implementation constraint that has been found:

- Cross-platform programming language that allows development for iOS and Android

Interface constraints

Interface constraints consist of requirements about the interaction with an external item. In other words, these requirements are about the protocols or the nature of the information that is passed across that interface between the system in question and external system (Eeles, 2004). For this project only one interface requirement has been discovered:

- The application should access an external server that contains a database and a recommendation system

Physical constraints

Lastly, physical restrictions are all those that limit the hardware that houses the system, for example the shape, size, and weight (Eeles, 2004). However, in this project there is no physical restriction.

4.3 Architecture

In this section several artifacts about the architecture of the Webecool system will be shown. For the deployment two alternatives were found and discussed and one will be selected.

4.3.1 Component Diagram

The logical architecture mainly supports functional requirements - what the system should provide in terms of services to its users. The system is decomposed into a set of key abstractions, this decomposition is not only for the sake of functional analysis, but it is also useful for identifying common mechanisms and design elements across different parts of the system (Kruchten, 1995). The logical view is intended to provide a basis for understanding the structure and organization of the system design, illustrating the main components and their relationships, which encompass architecturally significant behaviors.

Below, in the figure 4.4 it's possible to see illustrated a high-level component diagram of the webecool system. The mobile applications will interact with three components, the authentication, the database and the recommendation system. The authentication component allows the user to register, login and if the user is authenticated then it can store data on the database. The application also communicates with the recommendation system which will provide recommendations. In order to provide relevant recommendations, the recommendation system also communicates with the database to retrieve it's data and through the use of machine learning it

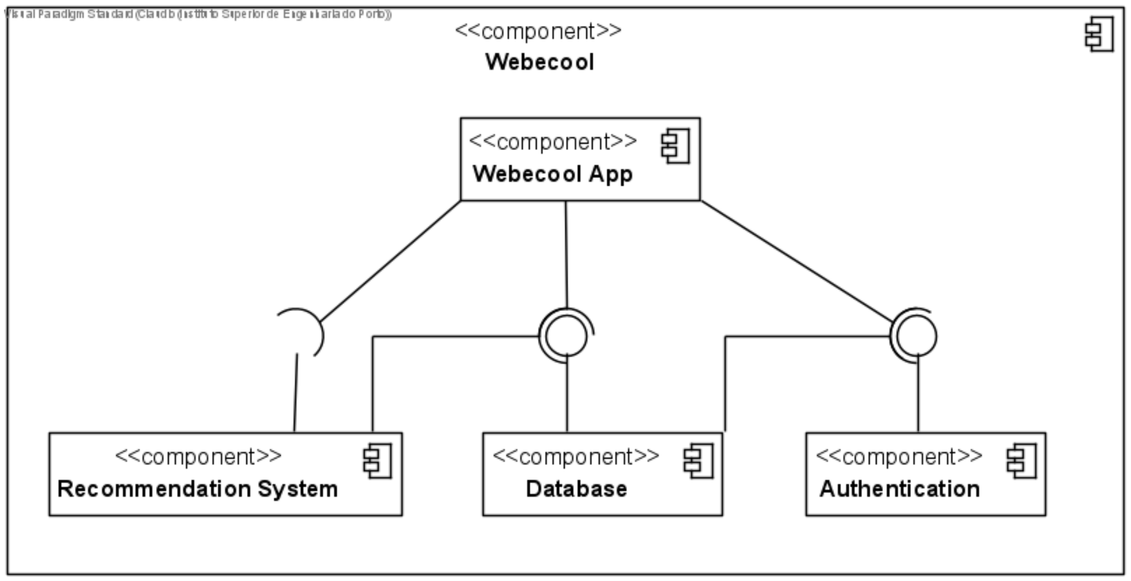


FIGURE 4.4: Component Diagram

4.3.2 Recommendation System Component Diagram

From the figure 4.4 we can identify one component named Recommendation system. Below, in the figure 4.5 we can see that component of the webecool system in more detail. It has the following components:

- Controllers: responsible for handling API requests;
- Recommender: has the main logic inherent to the recommendation;
- Mathematics: responsible for providing mathematical calculations such as matrix multiplication and factorization;
- Comparers: responsible for having the different methods for obtaining the degree similarity between users;
- Models: responsible for all the domain classes;
- Parsers: responsible for importing data from files and mapping those data for generating recommendations;
- Datasets: data to be used for the recommendations.

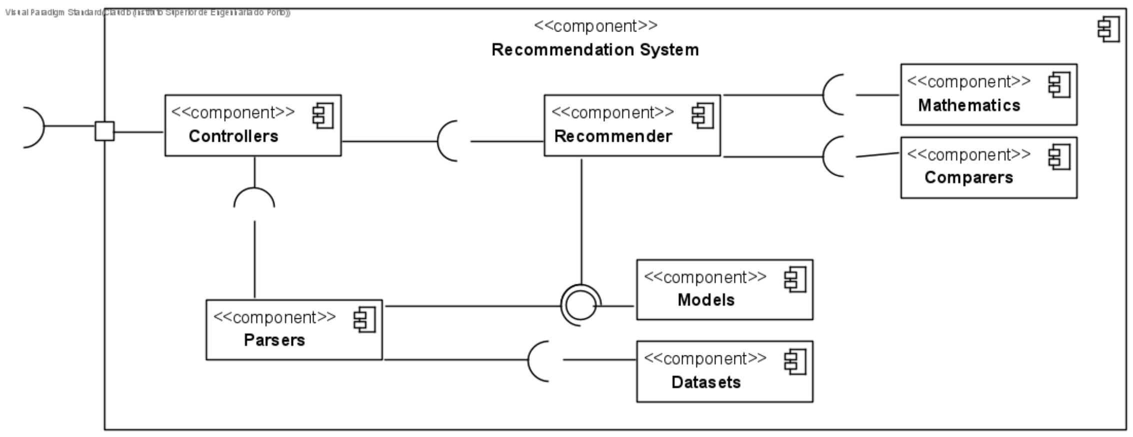


FIGURE 4.5: Components Diagram

4.3.3 Database

The Firestore from Google Firebase is a horizontally scaling document-model NoSQL database in the cloud. In the figure 4.7 it is possible to see the documents and collections created for the webecool system. Each user can only read and write on his user document due to the authentication and the security rules created, on the other hand every user can read the Exercises collection.

The Entity Relationship Diagram (ERD) present on the figure 4.6 shows how the *Users* collection contains a subcollection *Ratings* however there's no direct connection between the *Exercises* collection and the *Ratings* collection due to the way that the Firestore document-model NoSQL database works. So, each rating will have an *exerciseId* which would act as a foreign key.

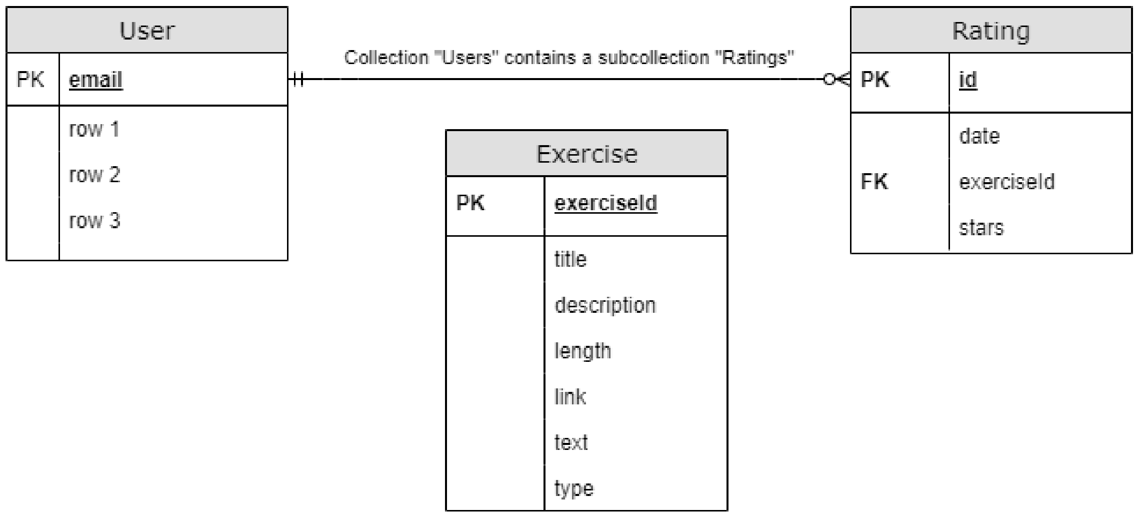


FIGURE 4.6: Entity Relationship Diagram (ERD)

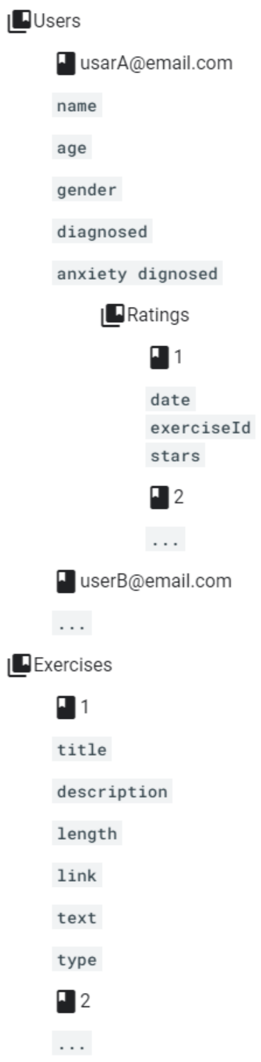


FIGURE 4.7: Database structure

4.3.4 Deploy Diagram

In this subsection, two proposals for the deployment of application will be presented and discussed and one will be selected.

Alternative 1

The deployment diagram shown on the figure 4.8 has two servers, one server handles the recommendation system and other minor application components while the other server handles the database. The application, either on iOS or Android, interacts with the database to store and retrieve information. The application also uses the recommendation system to get recommendation of exercises/procedures, this recommendation system needs information about the users and the exercises that is stored in the database in order to offer decent recommendations, as such the recommendation system also interacts with the server that contains the database.

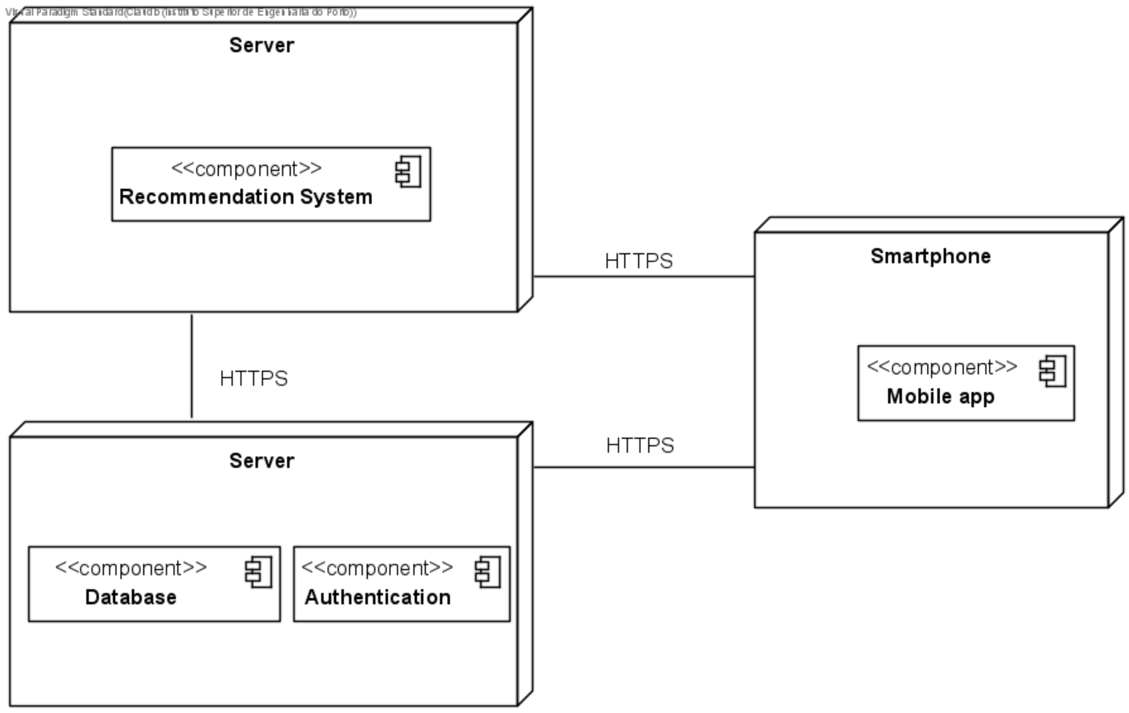


FIGURE 4.8: Deployment Diagram 1

Alternative 2

Unlike the previously deployment proposal, this alternative (Figure 4.9)only contains one servers, it contains both the recommendation system the database. The application, either on iOS or Android, interacts with this server to either use the database or the recommendation system to get recommendation of exercises/procedures.

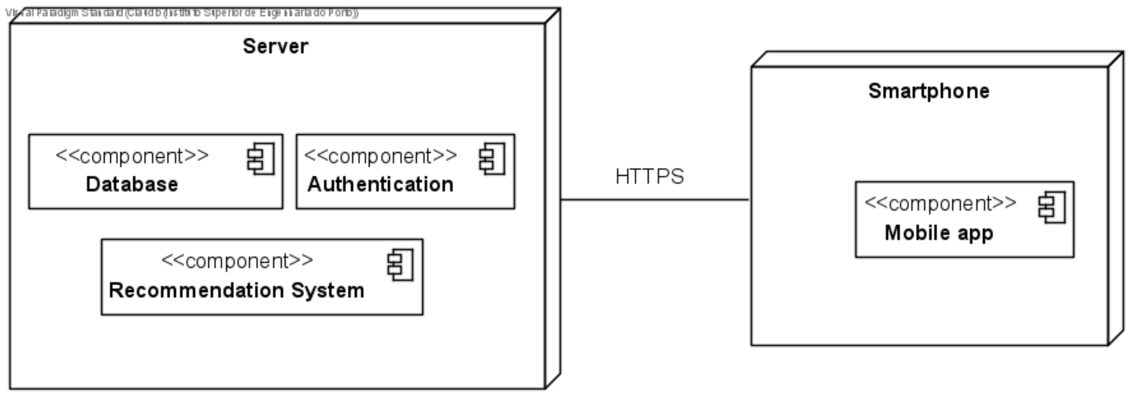


FIGURE 4.9: Deployment Diagram 2

Deployment Alternative Selection

The only difference between the two alternatives shown above in the figures 4.8 and 4.9 is that one contains only one server and the other one contains two servers, one for database and the other one for the recommendation system.

Platforms like Azure and Firebase offer a variety of services which can make the first approach worthwhile due to free plans and low costs on smaller projects (Firebase, n.d.) (Azure, n.d.).

Having the recommendation system and the database on the same server offers one immediate advantage, the communication between recommendation system and database will be faster however it's not a problem if the recommendation system doesn't receive data from the database instantly. The only component of the project that should have a quick response time is the mobile application.

Firebase is a Backend-as-a-Service (BaaS) that has a NoSQL database called Firestore, it is suitable for small-scale applications but it is also scalable. One of the key aspects of this database is the access method for mobile apps. There are libraries for Android and iOS and even other technologies like .NET which offer a simple way of connecting the application to the back-end (Sagar, 2019) ("Google.Cloud.Firestore .NET Client Library for the Firestore API.", n.d.).

The implementation of a recommendation system can be done as a ASP.NET core web application which can then be deployed to Microsoft Azure, this can not be deployed to Firebase for example, as such, the database can remain on Firebase and the recommendation system can be deployed to Azure.

For the reasons mentioned, the first alternative (Figure 4.9) was chosen and an updated version of the diagram is shown below in the figure 4.10. The recommendation system will be deployed to Microsoft Azure. This approach was used due to the simplicity of the deployment process of a ASP.NET core web application. For the database and authentication the Google Firebase platform was chosen due to it's ease of use for mobile applications. The mobile application will be developed using Microsoft Xamarin which is a technology that allows development of Android and iOS apps.

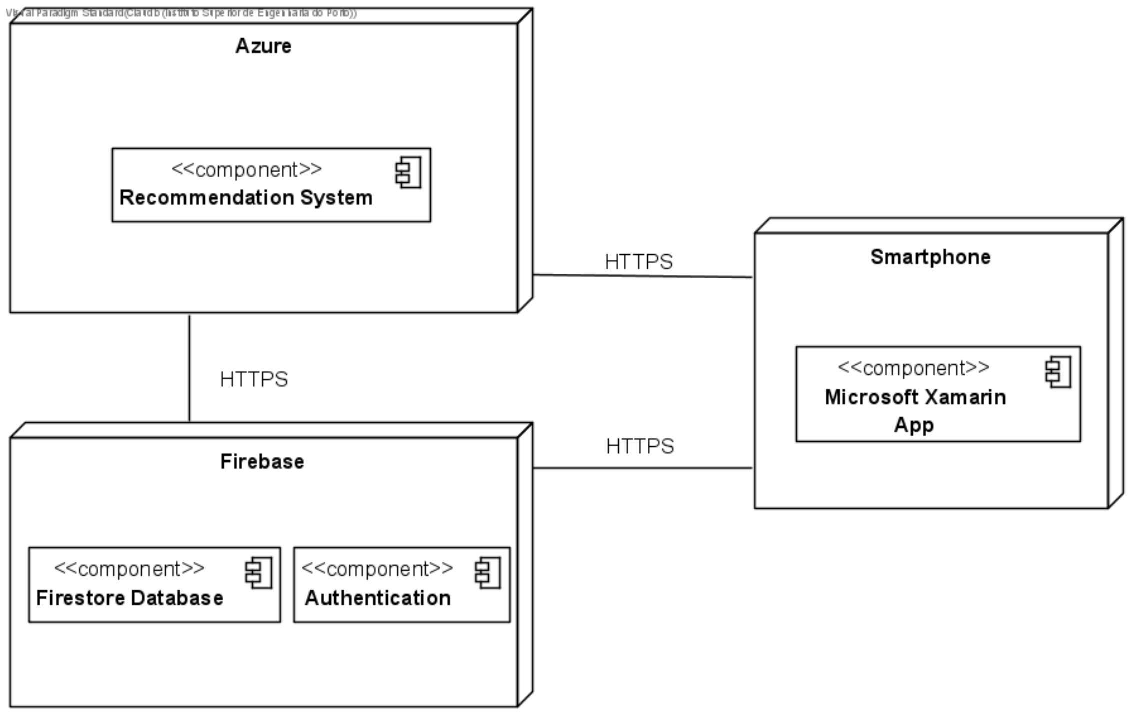


FIGURE 4.10: Updated deploy diagram

4.4 Summary

In this chapter, the system actors were recognized and the functional and non-functional requirements were identified and divided in the categories present in the FURPS+ classification system. The architecture of the Webecool system was shown in detail and two options for the deployment were discussed.

Chapter 5

Implementation

In this chapter, the development process carried out for the implementation of the different components of the system will be presented and discussed. For this, the development process for each of these components will be presented by providing details about the implementation and by showing artifacts that are considered relevant.

5.1 Database and Authentication

Before authenticating or using the database the application must first establish a connection with Firebase. This connection can be seen in the sequence diagram present on the figure 5.1. Two classes were created to handle the Authentication and the Firestore database, these classes are called *AuthService* and *FirestoreService*, respectively. The *AuthService* class establishes a connection with Firebase by calling *GoogleApiClient* class, after this connection is established the *FirebaseApp* is initialized and an instance of *FirebaseAuth* is obtained which will be useful for signing in. After the *AuthService* is initialized then the *FirestoreService* can be initialized too. It receives an instance of *FirebaseApp* as a parameter which can then be used to obtain an instance of *FirebaseFirestore* which is the class that allows interaction with the database.

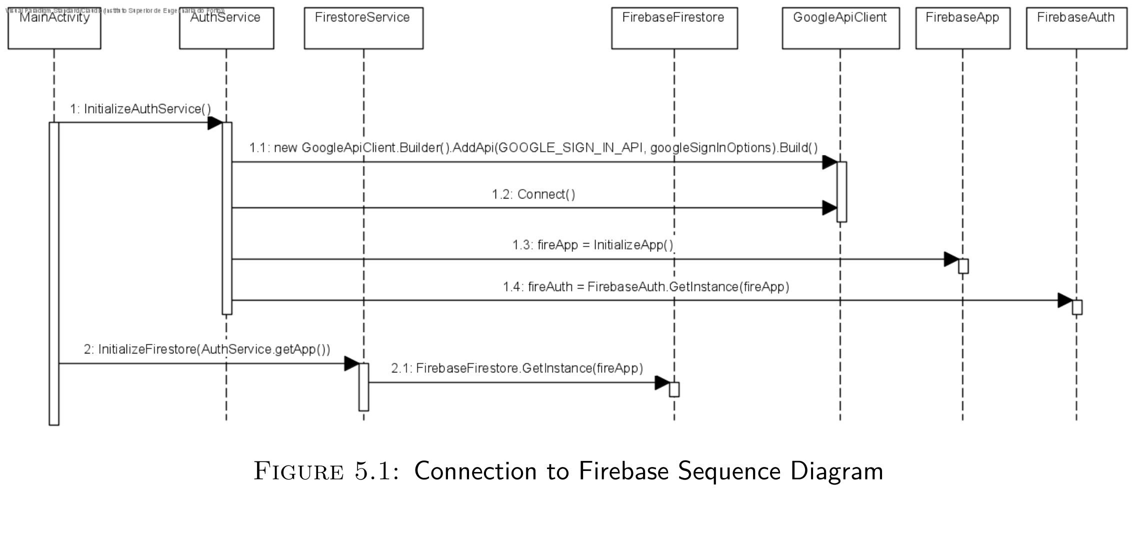

MainActivity
AuthService
FirestoreService
FirebaseFirestore
GoogleApiClient
FirebaseApp
FirebaseAuth
1: InitializeAuthService()
1.1: new GoogleApiClient.Builder().AddApi(GOOGLE_SIGN_IN_API, googleSignInOptions).Build()
1.2: Connect()
1.3: fireApp = InitializeApp()
1.4: fireAuth = FirebaseAuth.GetInstance(fireApp)
2: InitializeFirestore(AuthService.getApp())
2.1: FirebaseFirestore.GetInstance(fireApp)

FIGURE 5.1: Connection to Firebase Sequence Diagram

In order to sign in the user must press a button. After the button is pressed the activity gets an intent which allows the user to select his Google Account (Figure 5.29). When a Google Account is selected the method *OnActivityResult()* is called which will then send the account to the *AuthService* class which will attempt to sign in into Firebase by using the previously mentioned instance of *FirebaseAuth* that was obtained during the connection to firebase in figure 5.1. This sign in can result in a success or a failure, either way the *AuthActivty* handles what happens next. The process described above can be seen in figure 5.2.

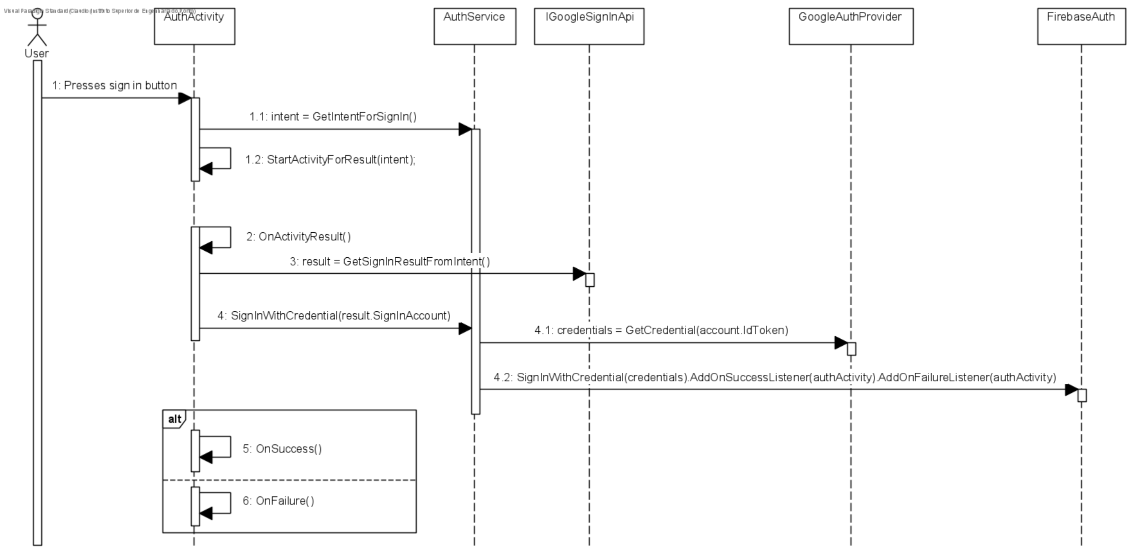
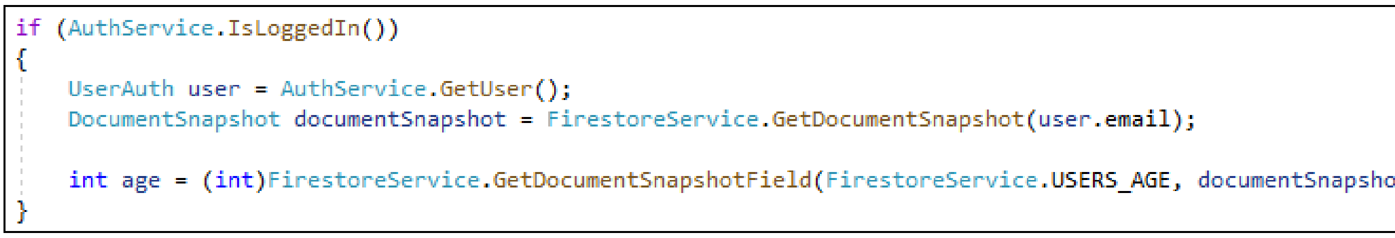

User
AuthActivity
AuthService
IGoogleSignInApi
GoogleAuthProvider
FirebaseAuth
1: Presses sign in button
1.1: intent = GetIntentForSignIn()
1.2: StartActivityForResult(intent);
2: OnActivityResult()
3: result = GetSignInResultFromIntent()
4: SignInWithCredential(result.SignInAccount)
4.1: credentials = GetCredential(account.IdToken)
4.2: SignInWithCredential(credentials).AddOnSuccessListener(authActivity).AddOnFailureListener(authActivity)
alt
5: OnSuccess()
6: OnFailure()

FIGURE 5.2: Sign In Sequence Diagram

Now that the application established a connection to Firebase and Firestore and a user is signed in, the application can interact with the document-model NoSQL Firestore database. An example of that interaction can be seen in the code snippet 5.3.



```
if (AuthService.IsLoggedIn())
{
    UserAuth user = AuthService.GetUser();
    DocumentSnapshot documentSnapshot = FirestoreService.GetDocumentSnapshot(user.email);

    int age = (int)FirestoreService.GetDocumentSnapshotField(FirestoreService.USERS_AGE, documentSnapsho
}
```

CODE SNIPPET 5.3: Firestore example

5.2 Recommendation System

With the objective of predicting anxiety relieving exercises that will have a positive impact on the user, it was decided to use an intelligent recommendation system that would assist in the user's decision process. A system is considered intelligent if it has the ability to suggest recommendations based on a created model and has the ability to adapt itself when existing data changes (Katakam, 2019).

The implementation of this system was based on Scott Clayton's article "Building a Recommendation Engine in C#" (Clayton, 2018). It has a rating of 4.95 / 5 based on ratings from readers of Code Project which is one large community of software developers worldwide. It received two awards: "Best C# Article of March 2018 : Second Prize" and "The Machine Learning and Artificial Intelligence Challenge : First Prize".

Throughout this section the main components of this recommendation system will be explained, such as: the classes used; the various techniques developed; the different similarity calculations used in the collaborative technique; the data set used; and a flow demonstrating the relationship between the various features of the system.

5.2.1 Class Diagram

Before explaining the important parts that make up this system, it is important to present them and show their connections. Thus, in Figure 5.4 there is a class diagram for the recommendation system.

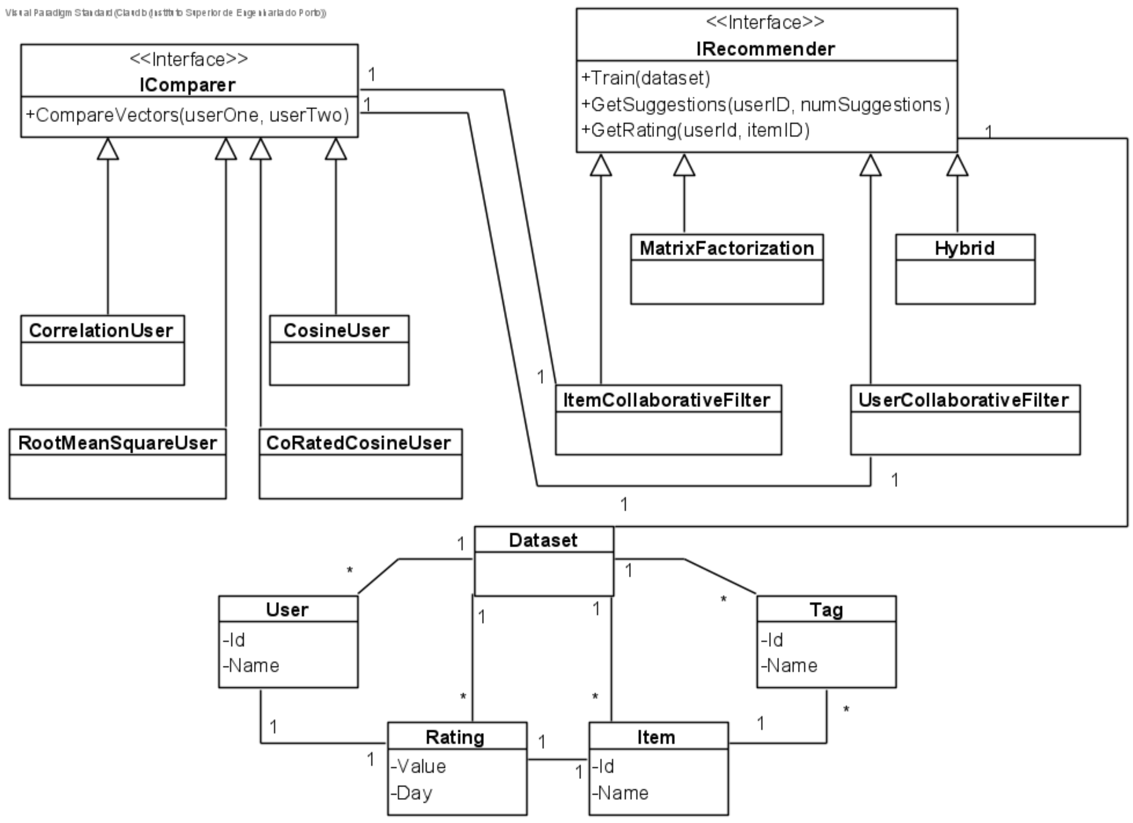


FIGURE 5.4: Recommendation System Class Diagram

It is possible to observe that the Strategy pattern appears two times. The Strategy Pattern makes it so that instead of implementing a single algorithm directly into code, a developer can implement any number of algorithms and choose which to pick. It is considered a good practice when there's multiple algorithms to perform a single task to simplify the code by using this pattern (Ali, 2020). In this case the strategy pattern allows the use of different techniques and different calculations to obtain similarity between users or items.

The *IRecommender* interface has three algorithm that each recommendation technique implements:

- **Train**: training has two purposes: the construction of the table made up of users and items; performing the training itself - iteratively train the model in order to improve predictions and improve the success rate;
- **GetSuggestions**: has the objective of returning the items that obtained the best score for a user;

- **GetRating**: predicts the classification of an item for a user, this means that it predicts the degree of appreciation of the item by the user.

The *IComparer* interface only has one algorithm which is implement differently depending on which of the comparers is chosen:

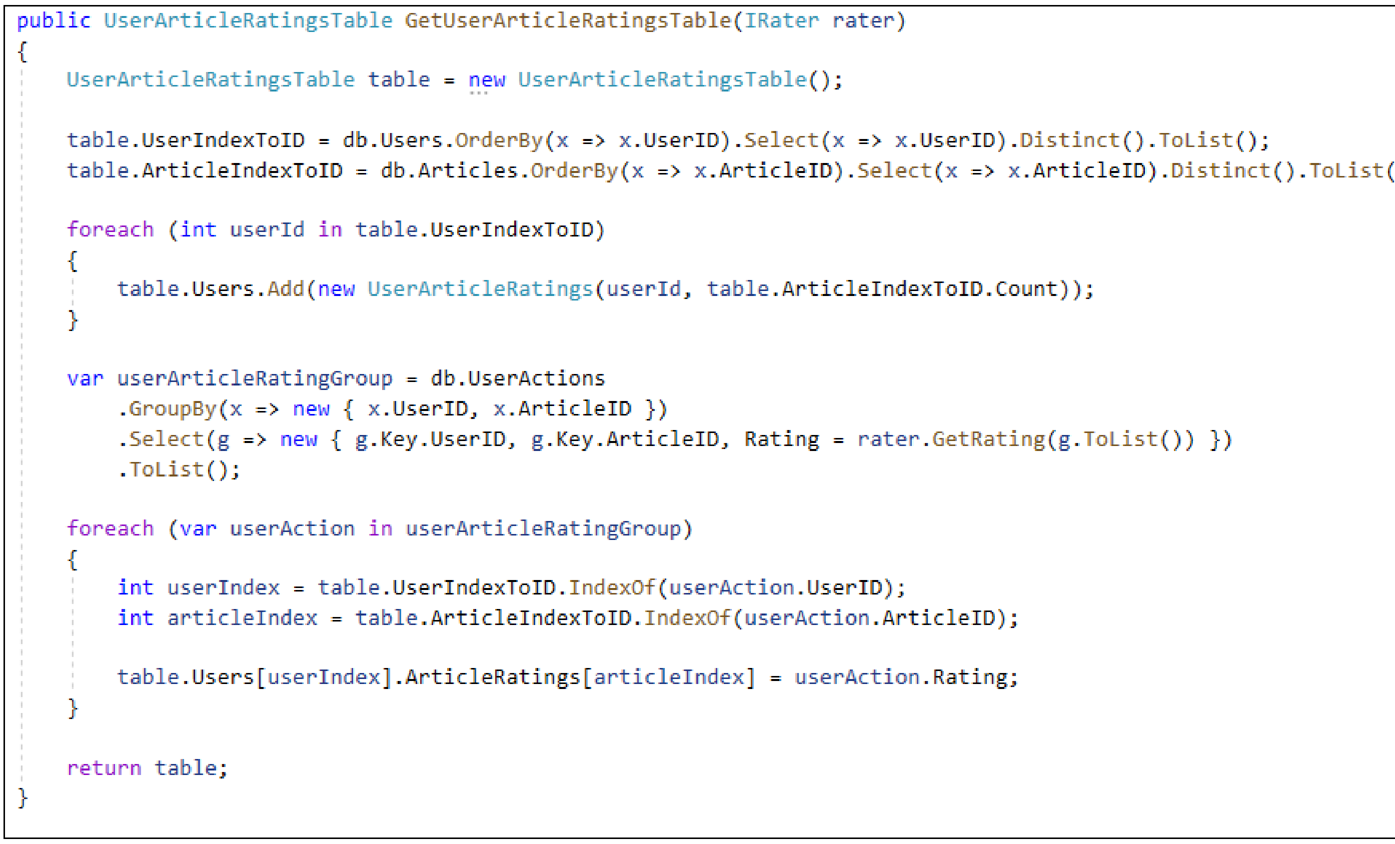
- **CompareVectors**: from two vectors (either two users or two items) it returns a single value indicating how similar or different they are.

5.2.2 Recommenders

As described in the subsection Class Diagram, the Train method is responsible for iteratively training the model. Training is a computationally complex task involving a long flow and several mathematical operations.

First, an instance of a class that implements the *IRecommender* interface class object is created with the number of latent features to be used in the learning process. This number of latent features is one of the two dimensions of the two matrices to be created - users and items.

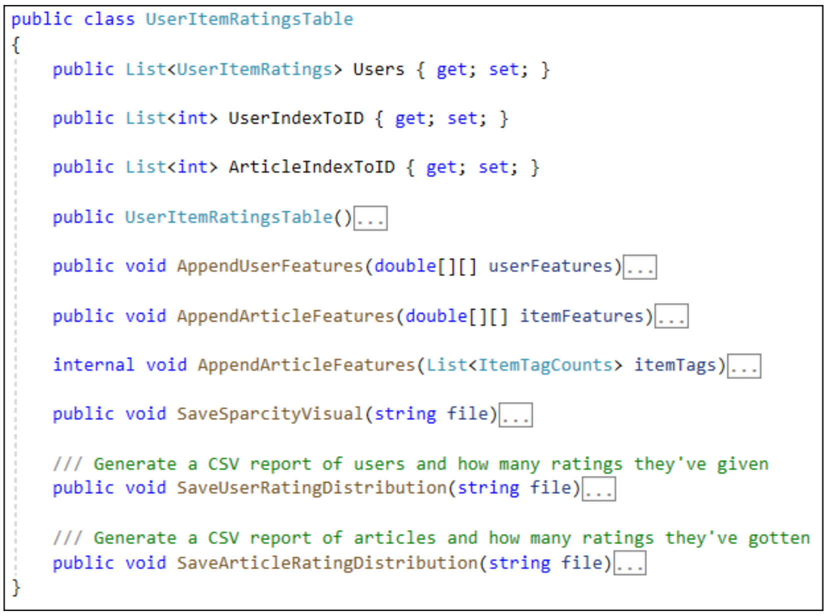
The next step is the construction of the users-items table or rating matrix, consisting of all users, items and ratings. The method responsible for this transformation is found in the code snippet 5.5.



```
public UserArticleRatingsTable GetUserArticleRatingsTable(IRater rater)
{
    UserArticleRatingsTable table = new UserArticleRatingsTable();

    table.UserIndexToID = db.Users.OrderBy(x => x.UserID).Select(x => x.UserID).Distinct().ToList();
    table.ArticleIndexToID = db.Articles.OrderBy(x => x.ArticleID).Select(x => x.ArticleID).Distinct().ToList(

    foreach (int userId in table.UserIndexToID)
    {
        table.Users.Add(new UserArticleRatings(userId, table.ArticleIndexToID.Count));
    }

    var userArticleRatingGroup = db.UserActions
        .GroupBy(x => new { x.UserID, x.ArticleID })
        .Select(g => new { g.Key.UserID, g.Key.ArticleID, Rating = rater.GetRating(g.ToList()) })
        .ToList();

    foreach (var userAction in userArticleRatingGroup)
    {
        int userIndex = table.UserIndexToID.IndexOf(userAction.UserID);
        int articleIndex = table.ArticleIndexToID.IndexOf(userAction.ArticleID);

        table.Users[userIndex].ArticleRatings[articleIndex] = userAction.Rating;
    }

    return table;
}
```

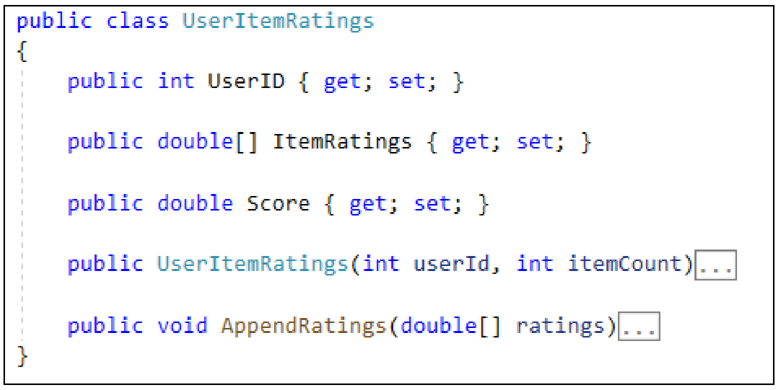
Code Snippet 5.5: Table transformation

The classes used in the code snippet 5.5 can be seen in the code snippets 5.6 and 5.7.



```
public class UserItemRatingsTable
{
    public List<UserItemRatings> Users { get; set; }

    public List<int> UserIndexToID { get; set; }

    public List<int> ArticleIndexToID { get; set; }

    public UserItemRatingsTable()...

    public void AppendUserFeatures(double[][] userFeatures)...

    public void AppendArticleFeatures(double[][] itemFeatures)...

    internal void AppendArticleFeatures(List<ItemTagCounts> itemTags)...

    public void SaveSparcityVisual(string file)...

    /// Generate a CSV report of users and how many ratings they've given
    public void SaveUserRatingDistribution(string file)...

    /// Generate a CSV report of articles and how many ratings they've gotten
    public void SaveArticleRatingDistribution(string file)...
}
```

Code Snippet 5.6: *UserItemRatingsTable* Class

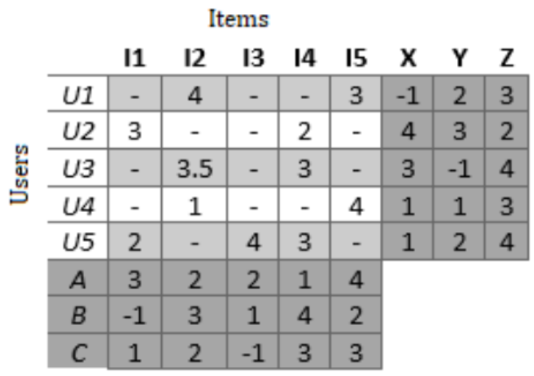


```
public class UserItemRatings
{
    public int UserID { get; set; }

    public double[] ItemRatings { get; set; }

    public double Score { get; set; }

    public UserItemRatings(int userId, int itemCount)...

    public void AppendRatings(double[] ratings)...
}
```

Code Snippet 5.7: *UserItemRatings* class

First, an instance of the UserItemRatingsTable class is initialized so that each user is then initialized (UserItemRatings) containing a array with the user ratings.

Figure 5.8 illustrates an example of a rating matrix with the existence of two Latent Features matrices. Throughout this section latent features will be explored in greater detail.


Items
Users

	I1	I2	I3	I4	I5	X	Y	Z
U1	-	4	-	-	3	-1	2	3
U2	3	-	-	2	-	4	3	2
U3	-	3.5	-	3	-	3	-1	4
U4	-	1	-	-	4	1	1	3
U5	2	-	4	3	-	1	2	4
A	3	2	2	1	4			
B	-1	3	1	4	2			
C	1	2	-1	3	3			

FIGURE 5.8: Rating Matrix with Users, Items and Latent Features

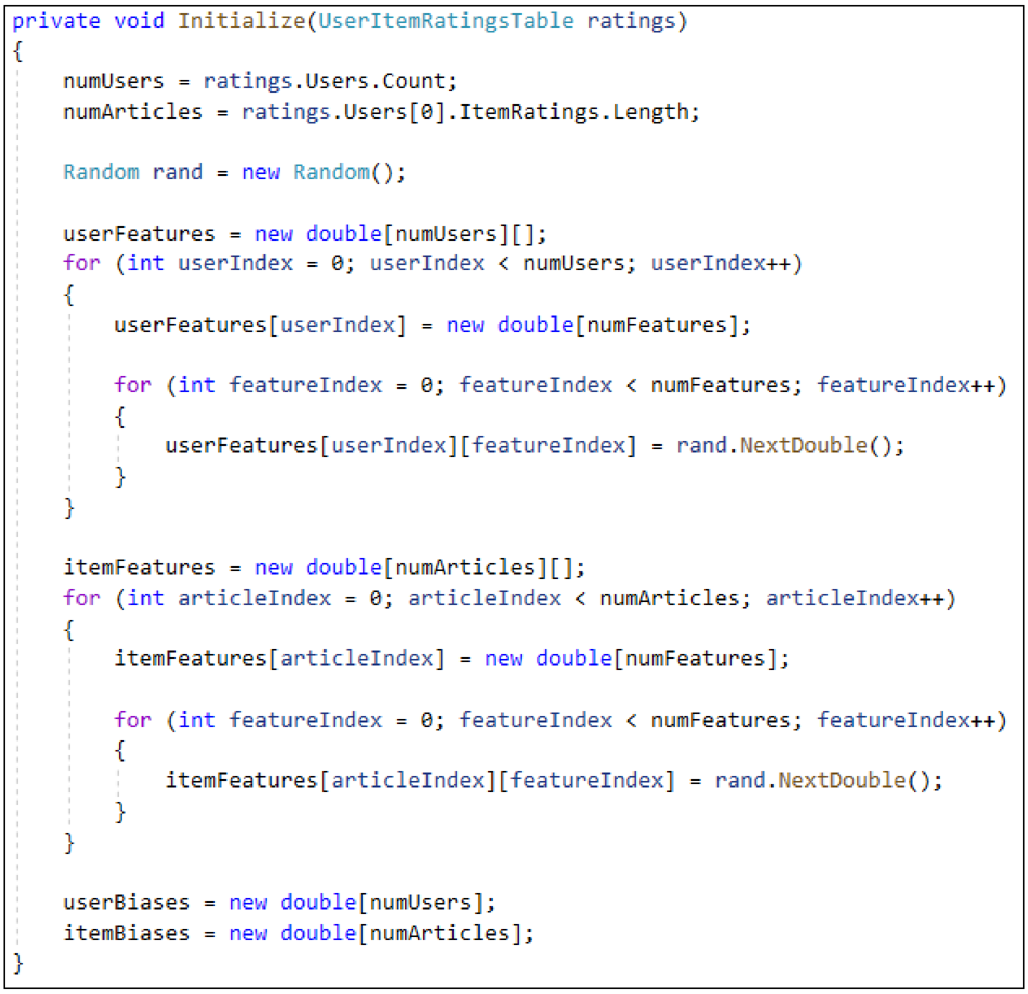
After the rating matrix is created, it is time to apply the Singular Value Decomposition (SVD) technique which factores a matrix of user items into smaller matrices, which can be used to fill in missing ratings. The technique is then initialized with the number of latent features and the number of iterations. This last parameter is the stopping criteria that the algorithm will go through to improve its prevision model.

After creating an instance of the *SingularValueDecomposition* class, the method of factoring the matrix is invoked by passing the rantings matrix as a parameter. It is in this method that the model responsible for making the predictions is built.

Initially, all the data necessary for the construction of the model are initialized: the two matrices of Latent Features (users and items) are initialized with random values so that, after certain adjustments and optimizations, these values approximate the real values; and biases for users and items.

Users behave differently when it comes to item classification. Supposing there's a ratings system that allows each user to rate each item from 1 to 5 stars. Supposing we have two users: User 1, who rates items with an average of 4 stars, and User 2, whose average rating is 1.5 stars. If User 1 rates some new item with 3 stars, it means something very different than if User 2 rates the same item with 3 stars (User 1 really liked the new item but User 2 didn't). This is considered user bias. Likewise, a 3 star rating on an item with low rating average means something different than a 3 star rating on an item with high rating average, this is item bias (Kirwin, 2016). Therefore, it is important to understand that a rating can have several meanings, which leads to the inclusion of biases in the recommendation system.

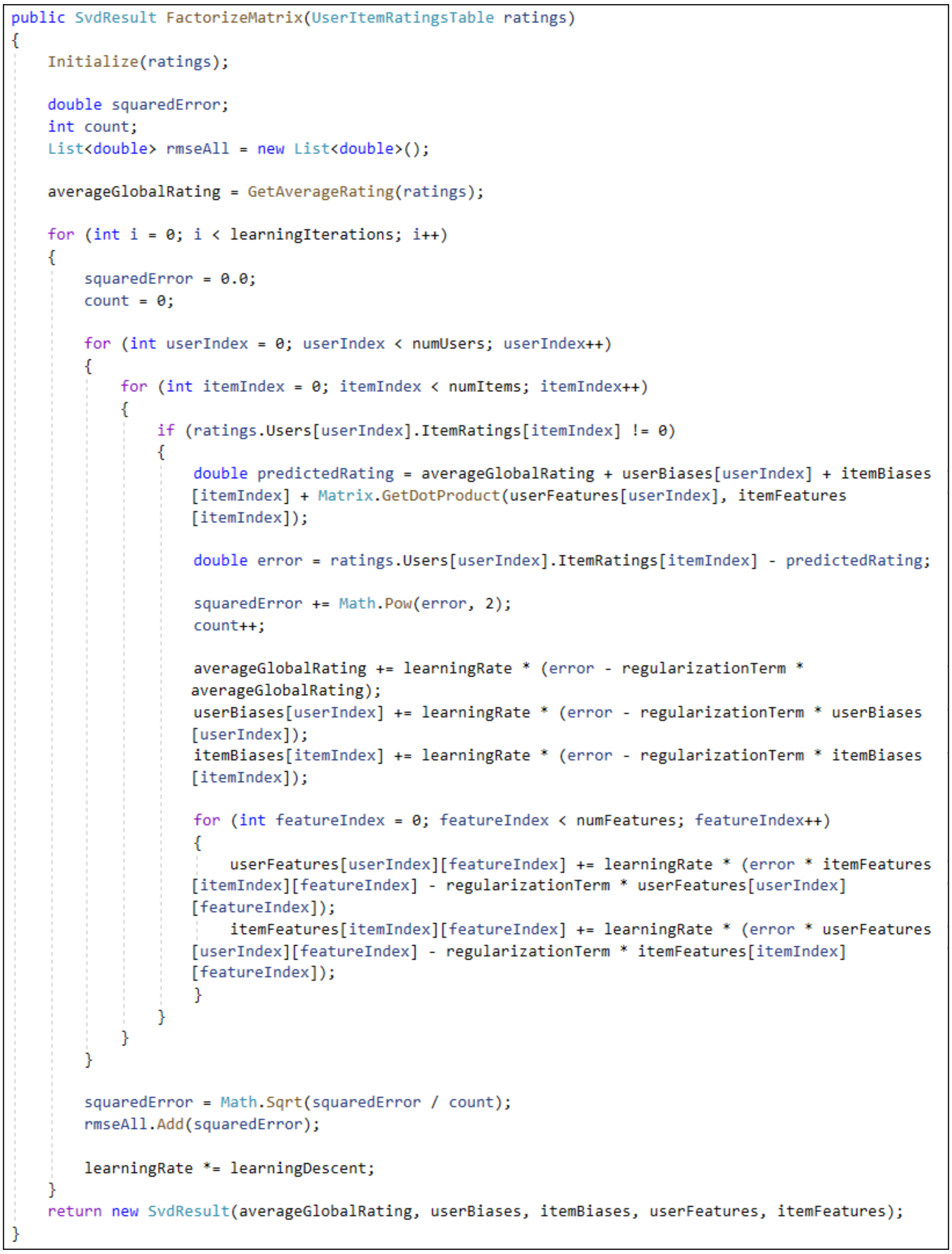
The data initialization method is shown in the code snippet 5.9.



```
private void Initialize(UserItemRatingsTable ratings)
{
    numUsers = ratings.Users.Count;
    numArticles = ratings.Users[0].ItemRatings.Length;

    Random rand = new Random();

    userFeatures = new double[numUsers][];
    for (int userIndex = 0; userIndex < numUsers; userIndex++)
    {
        userFeatures[userIndex] = new double[numFeatures];

        for (int featureIndex = 0; featureIndex < numFeatures; featureIndex++)
        {
            userFeatures[userIndex][featureIndex] = rand.NextDouble();
        }
    }

    itemFeatures = new double[numArticles][];
    for (int articleIndex = 0; articleIndex < numArticles; articleIndex++)
    {
        itemFeatures[articleIndex] = new double[numFeatures];

        for (int featureIndex = 0; featureIndex < numFeatures; featureIndex++)
        {
            itemFeatures[articleIndex][featureIndex] = rand.NextDouble();
        }
    }

    userBiases = new double[numUsers];
    itemBiases = new double[numArticles];
}
```

Code Snippet 5.9: *Initialize* method

After the data is initialized, the average of all the ratings present in the matrix is calculated, which will be used in future mathematical calculations. Now the iterative learning process can start, as it can be seen in the code snippet 5.10.



```
public SvdResult FactorizeMatrix(UserItemRatingsTable ratings)
{
    Initialize(ratings);

    double squaredError;
    int count;
    List<double> rmseAll = new List<double>();

    averageGlobalRating = GetAverageRating(ratings);

    for (int i = 0; i < learningIterations; i++)
    {
        squaredError = 0.0;
        count = 0;

        for (int userIndex = 0; userIndex < numUsers; userIndex++)
        {
            for (int itemIndex = 0; itemIndex < numItems; itemIndex++)
            {
                if (ratings.Users[userIndex].ItemRatings[itemIndex] != 0)
                {
                    double predictedRating = averageGlobalRating + userBiases[userIndex] + itemBiases
                    [itemIndex] + Matrix.GetDotProduct(userFeatures[userIndex], itemFeatures
                    [itemIndex]);

                    double error = ratings.Users[userIndex].ItemRatings[itemIndex] - predictedRating;

                    squaredError += Math.Pow(error, 2);
                    count++;

                    averageGlobalRating += learningRate * (error - regularizationTerm *
                    averageGlobalRating);
                    userBiases[userIndex] += learningRate * (error - regularizationTerm * userBiases
                    [userIndex]);
                    itemBiases[itemIndex] += learningRate * (error - regularizationTerm * itemBiases
                    [itemIndex]);

                    for (int featureIndex = 0; featureIndex < numFeatures; featureIndex++)
                    {
                        userFeatures[userIndex][featureIndex] += learningRate * (error * itemFeatures
                    [itemIndex][featureIndex] - regularizationTerm * userFeatures[userIndex]
                    [featureIndex]);
                        itemFeatures[itemIndex][featureIndex] += learningRate * (error * userFeatures
                    [userIndex][featureIndex] - regularizationTerm * itemFeatures[itemIndex]
                    [featureIndex]);
                    }
                }
            }
        }

        squaredError = Math.Sqrt(squaredError / count);
        rmseAll.Add(squaredError);

        learningRate *= learningDescent;
    }
    return new SvdResult(averageGlobalRating, userBiases, itemBiases, userFeatures, itemFeatures);
}
```

CODE SNIPPET 5.10: *FactorizeMatrix* method

Each iteration of this process has an error and a learning rate associated, which is adjusted at the end of each iteration. The main objective is to minimize the error in order to learn and build a better model.

Regarding the learning rate, it is used to adjust the values in the latent feature matrices, bringing their values closer to the real ones. It is also used to regularize the steps between each iteration, as illustrated in the developed algorithm (learningDescent variable), where the rate is reduced by 1% in each iteration, being an adjustable value.

On each iteration of the learning cycle, two iterations more are carried out - one for users (matrix rows) and another for items (matrix columns). First, it is checked whether the value of the current cell, given by the current row and column, is different from zero, because if it is, it is not possible to learn from it. On the other hand, being a non-zero value, the predicted rating is calculated with the following equation:

$$predictedRating = \bar{a} + b + c + \vec{u} * \vec{i} \tag{5.1}$$

where: $\bar{a}$ = rating average
b = user bias
c = item bias
$\vec{u} * \vec{i}$ = dot product of the latent features vectors

The predicted value could be calculated by only using the dot product between the vectors, however, the additions of bias make the prediction model more flexible and accurate, because it takes into account different variables.

Having calculated the predicted value, the error is calculated by using the *Root Mean Square Error (RMSE)* formula which is used due to the simple calculation and its common use when measuring the error of a model in predicting quantitative data. (Moody, 2019). The equation is as follows:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{5.2}$$

where: $\hat{y}_i$ = predicted value
y_i = observed value
n = number of ratings

After calculating the error, the next step is the optimization and adjustment of all variables: latent features, average of ratings and biases. This procedure is performed using the Gradient Descent algorithm. There are several variations for this algorithm, with the Stochastic Gradient Descent (Pandey, 2019) being chosen for this project. In short, optimization using this algorithm is done individually, one value at a time, instead of all values simultaneously.

In order to update the latent features matrix values, the following Gradient Descent formula (Pandey, 2019) was used:

$$\Delta q_{if} = \lambda(\epsilon p_{uf} - \gamma q_{if}), \Delta p_{uf} = \lambda(\epsilon q_{if} - \gamma p_{uf}) \tag{5.3}$$

where:

$\Delta q_{if}, \Delta p_{uf}$ = changes of values for the current user and item from the latent features matrices
λ = learning rate
ϵ = error associated with the prediction of the current value
γ = regularization term

There's a regularization term so that there are no high values of latent features, not causing the data to overfit, the regularization term multiplies by the value of the feature and is subtracted by the other value of the term.

In addition to updating the values of the feature matrices, there is also a need to optimize other variables: average ratings and biases. The equation is similar to that presented previously, with the exception of removing the error multiplier (p_{uf} and q_{if}).

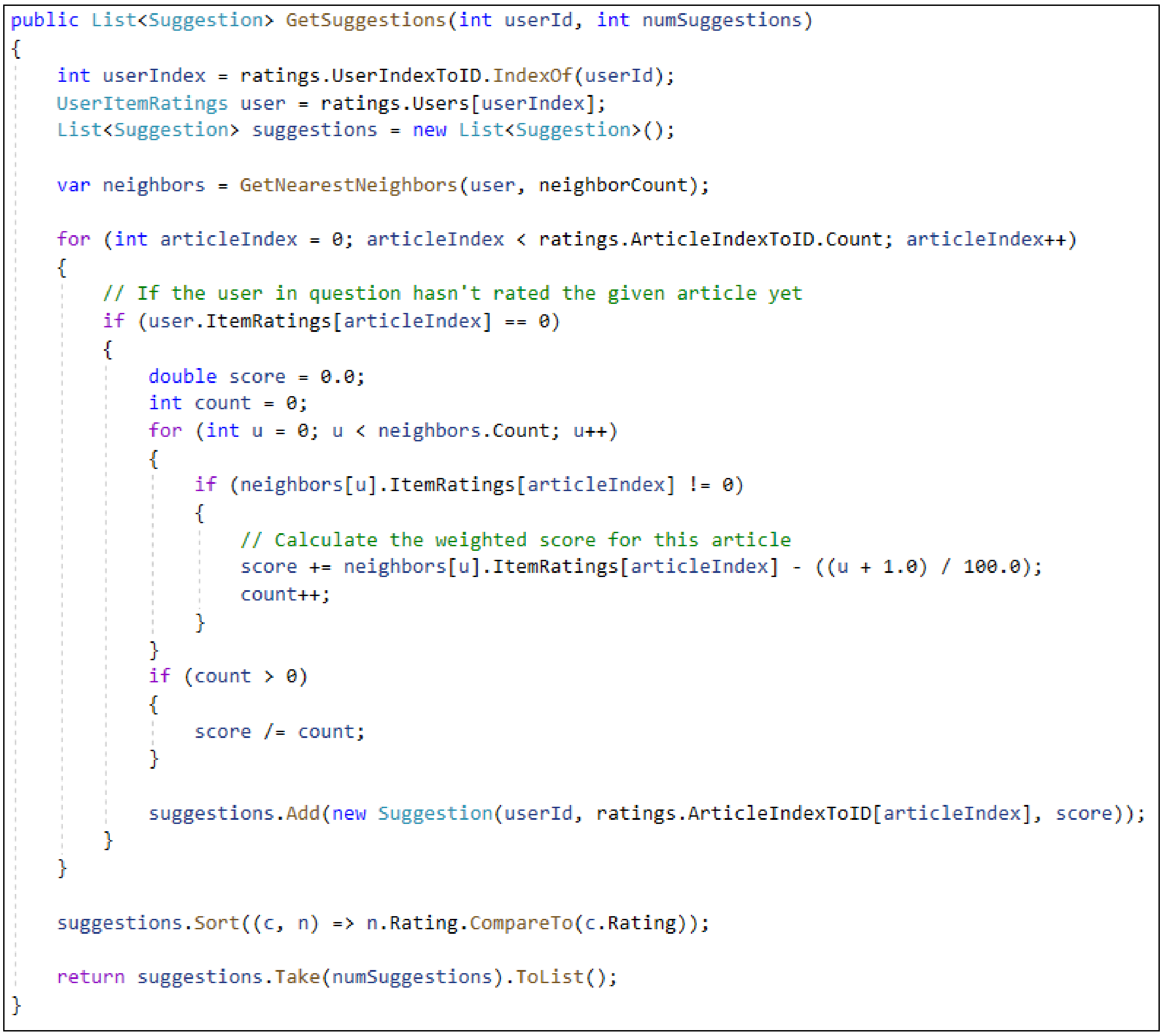
$$\Delta v = \lambda(\epsilon - \gamma v) \tag{5.4}$$

where: Δv = changes in average rating values and user and item biases
λ = learning rate
ϵ = error associated with the prediction of the current value
γ = regularization term

The described process is repeated for all the values in the ratings matrix, being associated, as previously mentioned, an error to each iteration (calculated using the RMSE technique).

Finally, an SVD object containing the average of user ratings, latent features and biases for users and items already optimized is returned, allowing predictions to be made.

Once the model is created and trained, obtaining suggestions becomes rather simple. For all items that the user has not yet assigned any rating, their value is predicted through the same mathematical equation present in the matrix factorization process. In the code snippet 5.11 there is an algorithm for getting suggestions from the recommendation system using the User Collaborative Filter technique.



```
public List<Suggestion> GetSuggestions(int userId, int numSuggestions)
{
    int userIndex = ratings.UserIndexToID.IndexOf(userId);
    UserItemRatings user = ratings.Users[userIndex];
    List<Suggestion> suggestions = new List<Suggestion>();

    var neighbors = GetNearestNeighbors(user, neighborCount);

    for (int articleIndex = 0; articleIndex < ratings.ArticleIndexToID.Count; articleIndex++)
    {
        // If the user in question hasn't rated the given article yet
        if (user.ItemRatings[articleIndex] == 0)
        {
            double score = 0.0;
            int count = 0;
            for (int u = 0; u < neighbors.Count; u++)
            {
                if (neighbors[u].ItemRatings[articleIndex] != 0)
                {
                    // Calculate the weighted score for this article
                    score += neighbors[u].ItemRatings[articleIndex] - ((u + 1.0) / 100.0);
                    count++;
                }
            }
            if (count > 0)
            {
                score /= count;
            }

            suggestions.Add(new Suggestion(userId, ratings.ArticleIndexToID[articleIndex], score));
        }
    }

    suggestions.Sort((c, n) => n.Rating.CompareTo(c.Rating));

    return suggestions.Take(numSuggestions).ToList();
}
```

Code Snippet 5.11: *GetSuggestions* method from *UserCollaborativeFilterRecommender*

When invoked, the items with the best ratings for that user are returned.

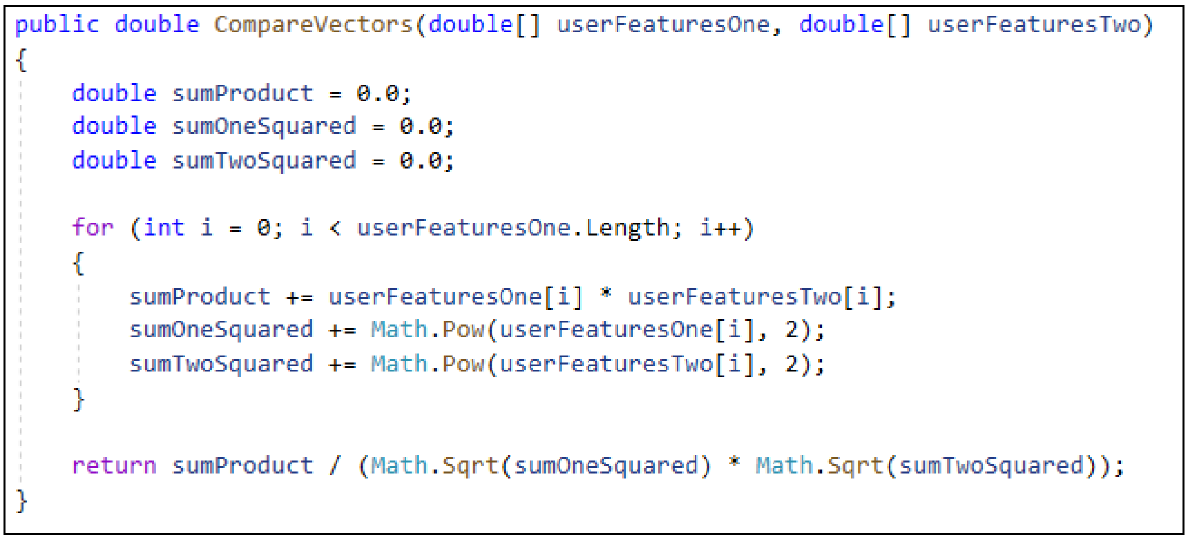
5.2.3 Comparers

One of the fundamental aspects of the collaborative filtering technique is the calculation of the similarity between users or items. There are several ways to calculate the similarity between users or items, the higher the value, the greater the degree of similarity, as such four different comparers algorithms were created for this project. These comparers implement the IComparer interface mentioned in the section 5.2.1. These are the formulas and algorithms used:

- **Cosine Similarity** — two vectors related to items in the user's multidimensional space are considered. The similarity between them is measured by calculating the cosine of the angle between these two vectors (Agarwal et al., 2017). The formula is presented in equation 5.5 and the code can be seen in the code snippet 5.12:

$$CS(a,u) = \frac{\sum_{i=1}^{n} a_i u_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} u_i^2}} \tag{5.5}$$

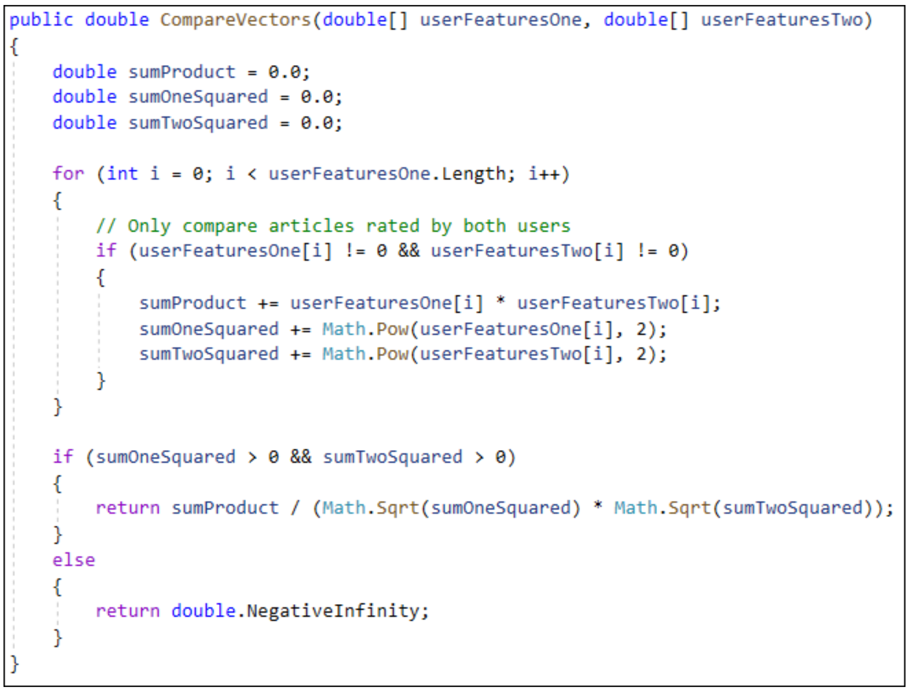
where: a_i = rating of the current item corresponding to user 1
u_i = rating of the current item corresponding to user 2



```
public double CompareVectors(double[] userFeaturesOne, double[] userFeaturesTwo)
{
    double sumProduct = 0.0;
    double sumOneSquared = 0.0;
    double sumTwoSquared = 0.0;

    for (int i = 0; i < userFeaturesOne.Length; i++)
    {
        sumProduct += userFeaturesOne[i] * userFeaturesTwo[i];
        sumOneSquared += Math.Pow(userFeaturesOne[i], 2);
        sumTwoSquared += Math.Pow(userFeaturesTwo[i], 2);
    }

    return sumProduct / (Math.Sqrt(sumOneSquared) * Math.Sqrt(sumTwoSquared));
}
```

Code Snippet 5.12: Cosine Similarity Method

- **Co-Rated Cosine Similarity** — the logic of this comparer is similar to the similarity of cosine with the following exceptions: only items that both users have rated are taken into account; and the division is only applied if the sum of the squares of the two vectors is greater than 0. The code is shown in the code snippet 5.13:



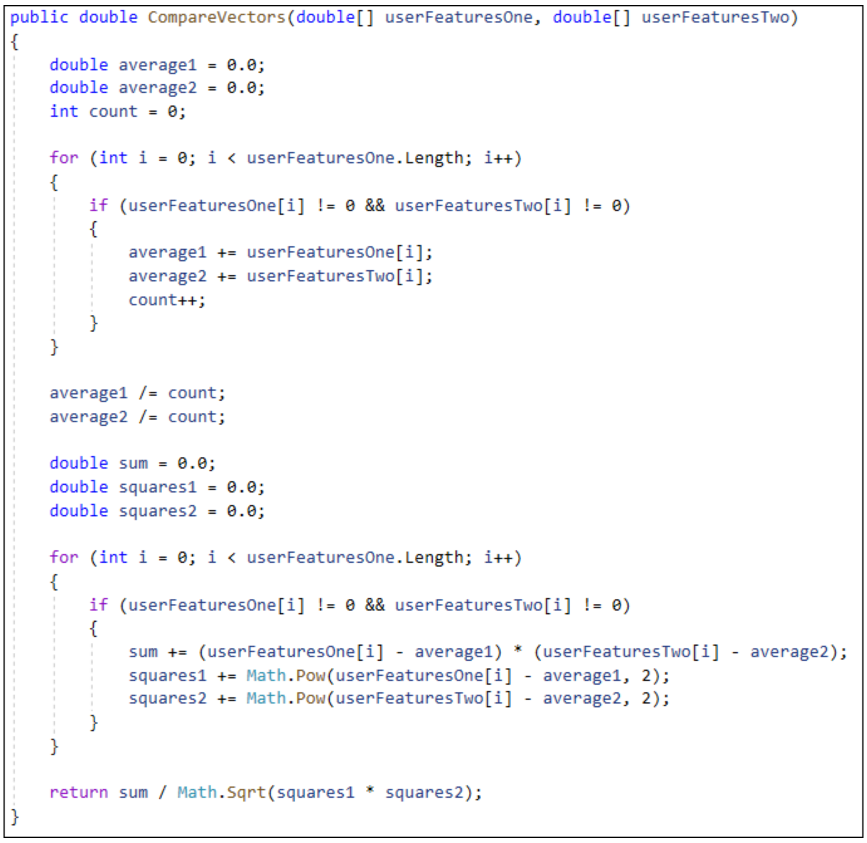
```
public double CompareVectors(double[] userFeaturesOne, double[] userFeaturesTwo)
{
    double sumProduct = 0.0;
    double sumOneSquared = 0.0;
    double sumTwoSquared = 0.0;

    for (int i = 0; i < userFeaturesOne.Length; i++)
    {
        // Only compare articles rated by both users
        if (userFeaturesOne[i] != 0 && userFeaturesTwo[i] != 0)
        {
            sumProduct += userFeaturesOne[i] * userFeaturesTwo[i];
            sumOneSquared += Math.Pow(userFeaturesOne[i], 2);
            sumTwoSquared += Math.Pow(userFeaturesTwo[i], 2);
        }
    }

    if (sumOneSquared > 0 && sumTwoSquared > 0)
    {
        return sumProduct / (Math.Sqrt(sumOneSquared) * Math.Sqrt(sumTwoSquared));
    }
    else
    {
        return double.NegativeInfinity;
    }
}
```

Code Snippet 5.13: Co-Rated Cosine Similarity Method

- **Pearson Correlation Similarity** — measures the degree of correlation between two variables. Assumes values between -1 and 1, where -1 represents a negative correlation, 1 a perfect correlation and 0 the variables do not depend on each other (Agarwal et al., 2017). The calculation formula can e seen below in the equation 5.6 and the code can be seen in the code snippet 5.14:

$$PCS(a,u) = \frac{\sum_{i=1}^{n}(r_{a,i} - \bar{r}_a) * (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{n}(r_{a,i} - \bar{r}_a)^2 * (r_{u,i} - \bar{r}_u)^2}} \tag{5.6}$$

where: $r_{a,i}$ = rating of the current item corresponding to user 1
$\bar{r}_a$ = average of the ratings of user 1
$r_{u,i}$ = rating of the current item corresponding to user 2
$\bar{r}_u$ = average of the ratings of user 2



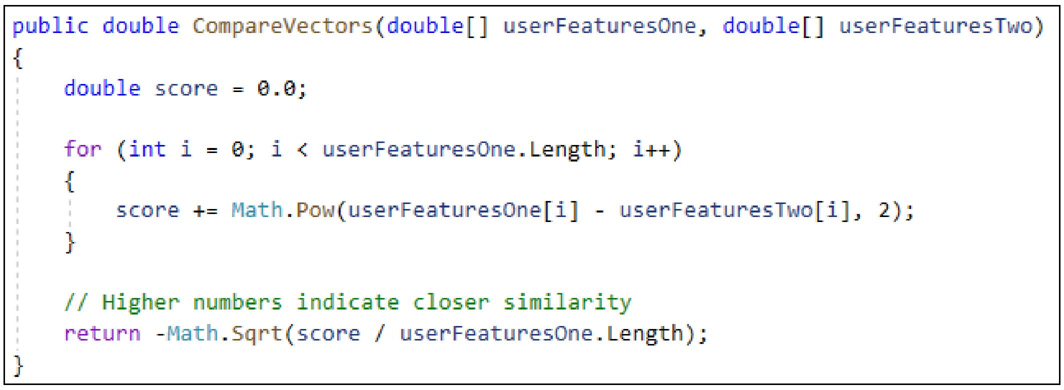
```
public double CompareVectors(double[] userFeaturesOne, double[] userFeaturesTwo)
{
    double average1 = 0.0;
    double average2 = 0.0;
    int count = 0;

    for (int i = 0; i < userFeaturesOne.Length; i++)
    {
        if (userFeaturesOne[i] != 0 && userFeaturesTwo[i] != 0)
        {
            average1 += userFeaturesOne[i];
            average2 += userFeaturesTwo[i];
            count++;
        }
    }

    average1 /= count;
    average2 /= count;

    double sum = 0.0;
    double squares1 = 0.0;
    double squares2 = 0.0;

    for (int i = 0; i < userFeaturesOne.Length; i++)
    {
        if (userFeaturesOne[i] != 0 && userFeaturesTwo[i] != 0)
        {
            sum += (userFeaturesOne[i] - average1) * (userFeaturesTwo[i] - average2);
            squares1 += Math.Pow(userFeaturesOne[i] - average1, 2);
            squares2 += Math.Pow(userFeaturesTwo[i] - average2, 2);
        }
    }

    return sum / Math.Sqrt(squares1 * squares2);
}
```

Code Snippet 5.14: Pearson Correlation Similarity Method

- **Root Mean Square Similarity** — calculates the average distance between items (Coutsias et al., 2004). The formula is presented in equation 5.7 and code can be seen on the code snippet 5.15:

$$RMS(a,u) = \sqrt{\frac{1}{N}\sum(a-u)^2} \tag{5.7}$$

where: a = rating of the current item corresponding to user 1
u = rating of the current item corresponding to user 2
N = number of ratings



```
public double CompareVectors(double[] userFeaturesOne, double[] userFeaturesTwo)
{
    double score = 0.0;

    for (int i = 0; i < userFeaturesOne.Length; i++)
    {
        score += Math.Pow(userFeaturesOne[i] - userFeaturesTwo[i], 2);
    }

    // Higher numbers indicate closer similarity
    return -Math.Sqrt(score / userFeaturesOne.Length);
}
```

Code Snippet 5.15: Root Mean Square Similarity Method

5.2.4 Flow

After explaining the implementation of the recommendation techniques and their functionalities, it is important to describe the flow and how the recommendation system works. Figure 5.16 shows an activity diagram listing all the features of this system.

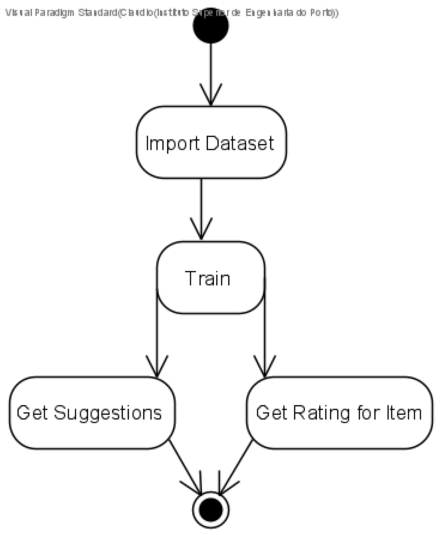

Visual Paradigm Standard(Cláudio(Instituto Superior de Engenharia do Porto))
Import Dataset
Train
Get Suggestions
Get Rating for Item

Figure 5.16: Activity Diagram of the Recommendation System

In figure 5.16, it is possible to observe that only after the dataset has been imported and its training has been carried out, is it possible to obtain suggestions (recommendations) or predict the classification of an item by a specific user.

Within the system, first of all, the recommendation technique to be used is chosen. Then, it is checked if a certain model has already been created. If it has already been created, it is imported into the system allowing the creation of recommendations for a user. On the other hand, if it does not already exist, the dataset is imported from the Firestore database and then a model is created and saved based on the imported data. This flow is shown in the sequence diagram present on figure 5.17.

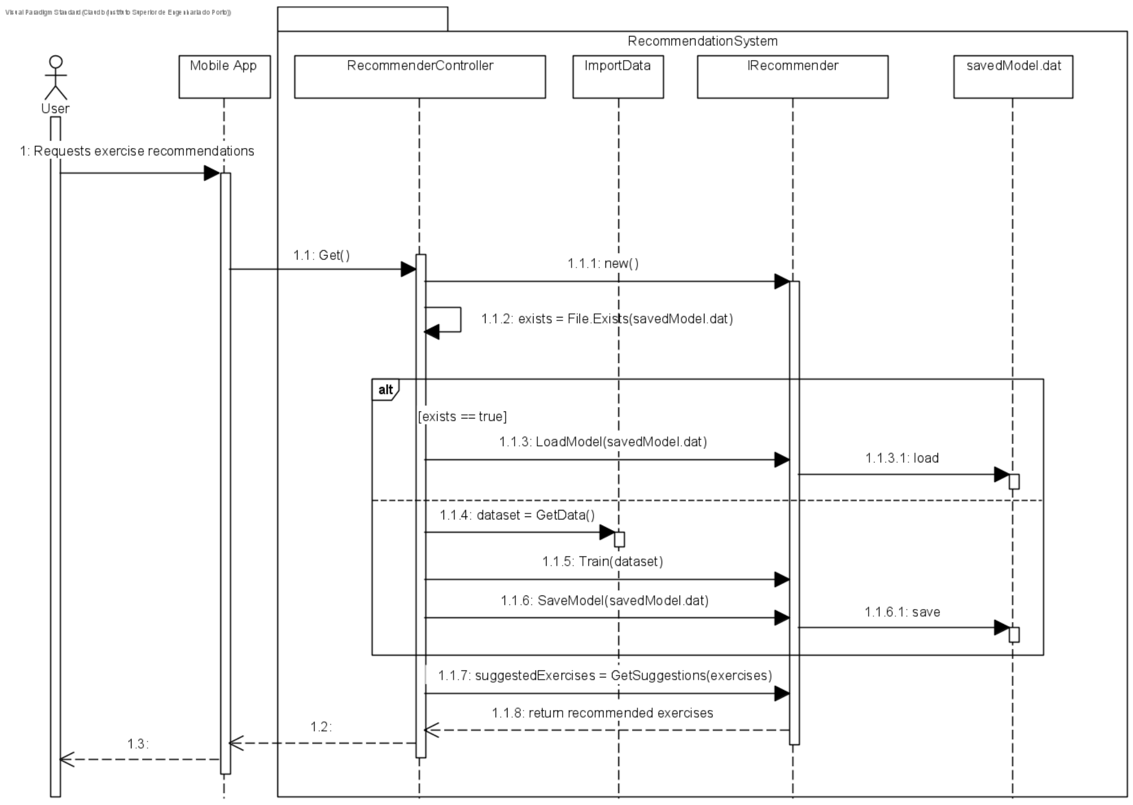

RecommendationSystem
User
Mobile App
RecommenderController
ImportData
IRecommender
savedModel.dat
1: Requests exercise recommendations
1.1: Get()
1.1.1: new()
1.1.2: exists = File.Exists(savedModel.dat)
alt
[exists == true]
1.1.3: LoadModel(savedModel.dat)
1.1.3.1: load
1.1.4: dataset = GetData()
1.1.5: Train(dataset)
1.1.6: SaveModel(savedModel.dat)
1.1.6.1: save
1.1.7: suggestedExercises = GetSuggestions(exercises)
1.1.8: return recommended exercises
1.2:
1.3:

FIGURE 5.17: Recommendation System Sequence Diagram

5.3 Mobile Application

The application was developed in Microsoft Xamarin, a technology which allows development for iOS and Android mobile platforms.

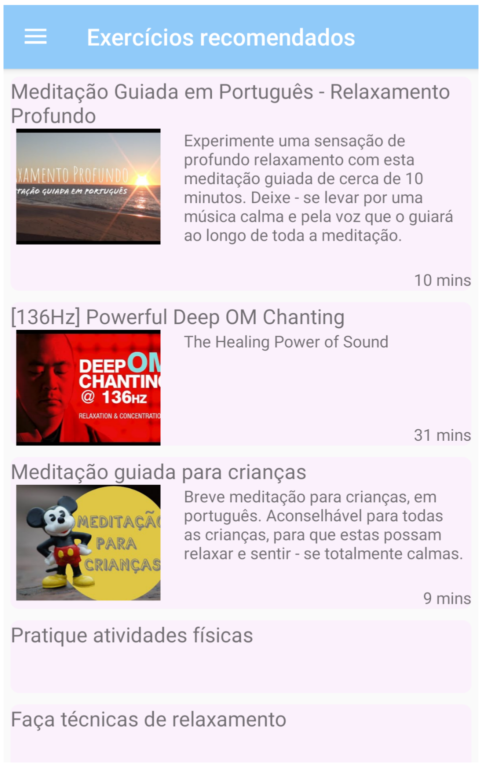
The use cases from the section 4.2.1 that were implemented will be discussed bellow. Artifacts such as code snippets, sequence diagrams and images from the mobile application will be used in this section to describe the details of the implementation of each use case, the options taken and the way they were implemented.

FR01 - Request exercise recommendations

In this use case the user is able to request various exercise recommendations. These recommendations come from data saved in the database by the user and other users (ratings and profile data), more information about the recommendation system can be found at the section 5.2. In order to get a list of recommendations from the recommendation system the application must perform a GET request. This request can be done at following URL that's hosted by Azure:

`https://webecoolbackend.azurewebsites.net/api/Recommendation/1/list/5`

Where 1 is an unique ID that each registered user has that is saved in the database and 5 is the number of recommendations that are going to be returned.



Exercícios recomendados

Meditação Guiada em Português - Relaxamento Profundo

Experimente uma sensação de profundo relaxamento com esta meditação guiada de cerca de 10 minutos. Deixe - se levar por uma música calma e pela voz que o guiará ao longo de toda a meditação.

10 mins

[136Hz] Powerful Deep OM Chanting

The Healing Power of Sound

31 mins

Meditação guiada para crianças

Breve meditação para crianças, em português. Aconselhável para todas as crianças, para que estas possam relaxar e sentir - se totalmente calmas.

9 mins

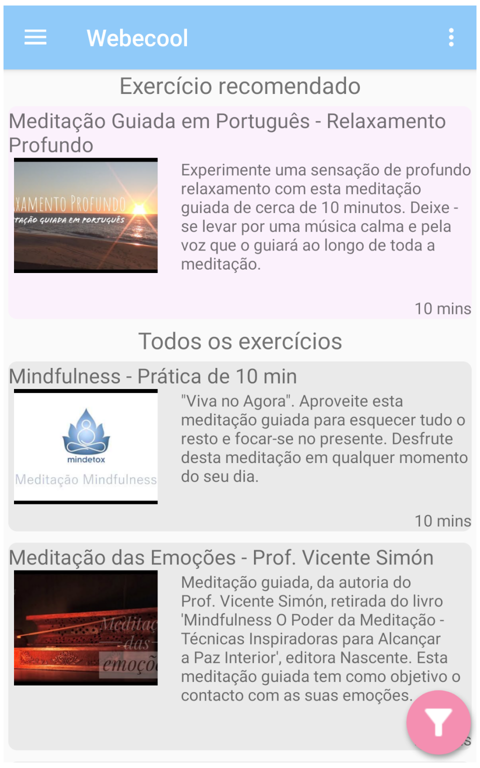
Pratique atividades físicas

Faça técnicas de relaxamento

FIGURE 5.18: Activity with various recommendations

FR02 - Receive recommendations with the use of machine learning

This use case works similarly to the first use case described above. The recommendation in this use case is present on the main page of the application, this will help more casual users of the application get access to recommendations that could improve their anxiety management. As it can be seen in the figure 5.19 the recommendation has a different color to the normal exercises, this was done to give emphasis. The colors of this application are meant to give the user a neat and calming interface, and light blue and light pink are some of the most calming colors according to surveys (Hale, 2019).


Webecool

Exercício recomendado

Meditação Guiada em Português - Relaxamento Profundo

Experimente uma sensação de profundo relaxamento com esta meditação guiada de cerca de 10 minutos. Deixe -se levar por uma música calma e pela voz que o guiará ao longo de toda a meditação.

10 mins

Todos os exercícios

Mindfulness - Prática de 10 min

"Viva no Agora". Aproveite esta meditação guiada para esquecer tudo o resto e focar-se no presente. Desfrute desta meditação em qualquer momento do seu dia.

10 mins

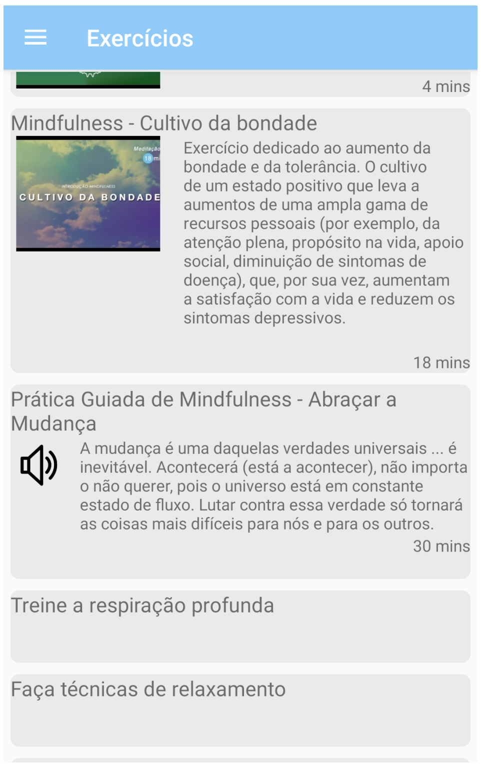
Meditação das Emoções - Prof. Vicente Simón

Meditação guiada, da autoria do Prof. Vicente Simón, retirada do livro 'Mindfulness O Poder da Meditação - Técnicas Inspiradoras para Alcançar a Paz Interior', editora Nascente. Esta meditação guiada tem como objetivo o contacto com as suas emoções.

FIGURE 5.19: Exercise Recommendation in the Main Activity

FA01 - Exercise with video

The application has three types of exercises, each one is presented different in the lists present on the application as it can be seen in the figure 5.20.



Exercícios

4 mins

Mindfulness - Cultivo da bondade

Exercício dedicado ao aumento da bondade e da tolerância. O cultivo de um estado positivo que leva a aumentos de uma ampla gama de recursos pessoais (por exemplo, da atenção plena, propósito na vida, apoio social, diminuição de sintomas de doença), que, por sua vez, aumentam a satisfação com a vida e reduzem os sintomas depressivos.

18 mins

Prática Guiada de Mindfulness - Abraçar a Mudança

A mudança é uma daquelas verdades universais ... é inevitável. Acontecerá (está a acontecer), não importa o não querer, pois o universo está em constante estado de fluxo. Lutar contra essa verdade só tornará as coisas mais difíceis para nós e para os outros.

30 mins

Treine a respiração profunda

Faça técnicas de relaxamento

FIGURE 5.20: Exercise List

One of the types of exercises are in video form, the user can watch a video with the objective of relieving anxiety. The video is from Youtube but the user can watch it inside the application. It's possible to see this type of exercise in the figure 5.21.

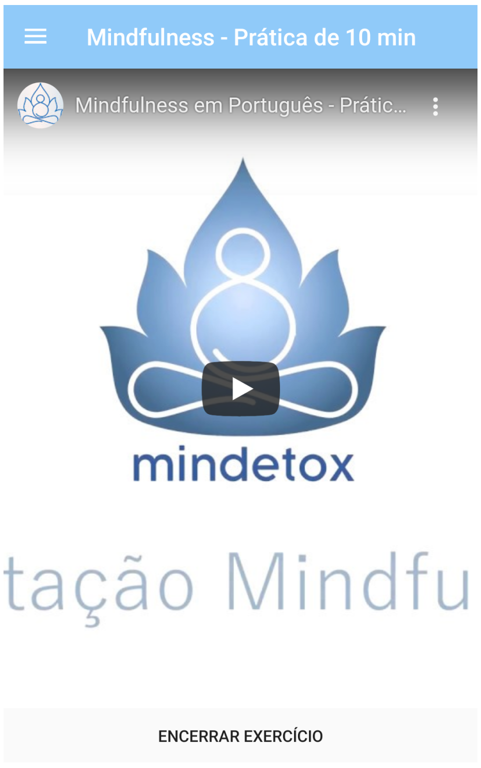

Mindfulness - Prática de 10 min
Mindfulness em Português - Prátic...
mindetox
tação Mindfu
ENCERRAR EXERCÍCIO

FIGURE 5.21: Video Exercise

It's important that these exercises have the potential to relieve anxiety, for this reason the mindfulness technique for meditation was chosen because of it's significant beneficial impact on stress, anxiety, depression, and well-being (Spijkerman et al., 2016) (Khoury et al., 2015).

The videos in question are from three different Youtube channels. The channel "Sociedade Portuguesa de Meditação - Mindfulness Institute" is from a non-profit association that seeks to contribute to a more conscious society through the study and practice of Meditation ("Meditt from Mindfulness Institute." n.d.), it's channel was chosen for having mindfulness videos with people such as Paulo Borges who's responsible for mindfulness courses, workshops and meditation retreats since 1999, he's a Professor at the University of Lisbon, member of the presidency of the Portuguese Buddhist Union ("Meditação Guiada - Mindfulness #1 c/ Paulo Borges", 2014). This channel also has videos from João Palma who teaches mindfulness since 2008 and he is a Certified Breathworks Teacher from Breathworks CIC and a Certified MBSR Teacher from the UC San Diego School of Medicine Mindfulness-Based Professional Training Institute where he is also a MBSR Mentor providing supervision to MBSR teachers on their training pathway ("João Palma Budadharma", n.d.).

Another channel is called "Mindful Healing" who also has videos from Paulo Borges. Lastly there's the Mindetox channel who has some videos from Vicente Simón who is a psychiatrist, researcher and professor of Physiological Psychology at the Faculty of Psychology at the University of Valencia. In the last 15 years he has dedicated himself to the study of consciousness and the practice of meditation, having published several works on the subject. He is the founder and president of the "Asociación de Mindfulness y Salud", an organization dedicated to the promotion and training of health professionals interested in integrating mindfulness into their clinical practice ("Vicente Simón | Nascente", n.d.).

FA02 - Exercise with audio

Similarly to the video exercises this type of exercises also features mindfulness based meditation but via audio. These audios were obtained from videos from the channels mentioned above in the use case FA01 - Exercise with video. The interface of the audio exercises can bee seen in the figure 5.22.

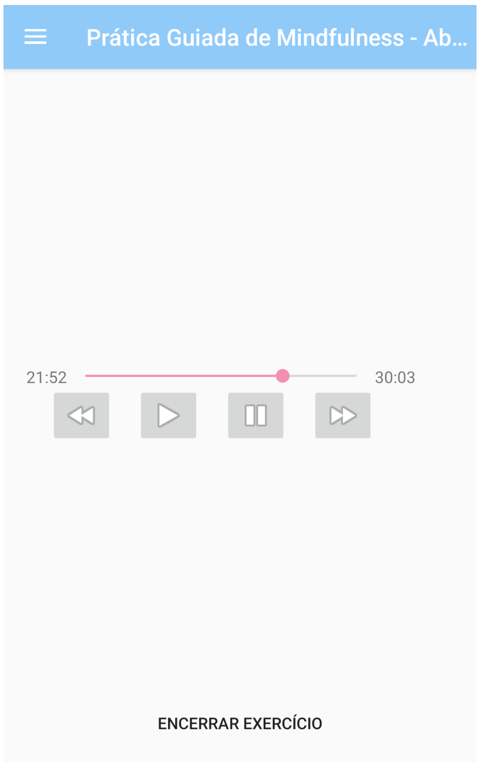
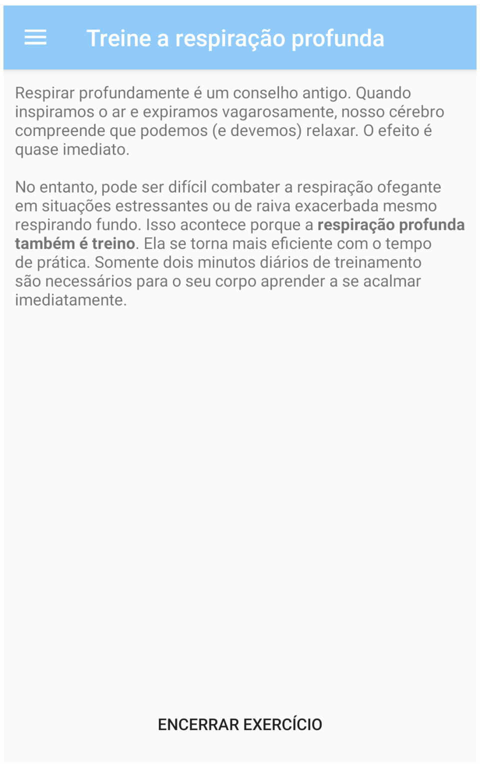


FIGURE 5.22: Audio Exercise

FA03 - Exercise with text

Text-based exercises are from an article written by the CEO and Founder of Vittude who studied The Science of Happiness at the University of California, Berkeley (Pimenta, 2019).

The interface of this type of exercises can be seen in the figure 5.23.


Treine a respiração profunda

Respirar profundamente é um conselho antigo. Quando inspiramos o ar e expiramos vagarosamente, nosso cérebro compreende que podemos (e devemos) relaxar. O efeito é quase imediato.

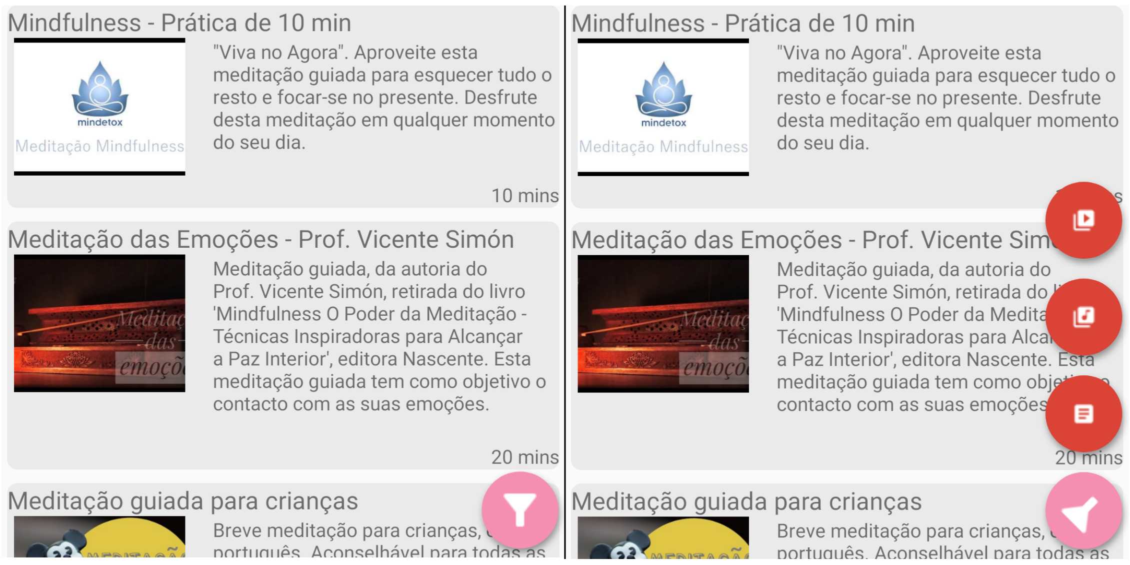
No entanto, pode ser difícil combater a respiração ofegante em situações estressantes ou de raiva exacerbada mesmo respirando fundo. Isso acontece porque a **respiração profunda também é treino**. Ela se torna mais eficiente com o tempo de prática. Somente dois minutos diários de treinamento são necessários para o seu corpo aprender a se acalmar imediatamente.

ENCERRAR EXERCÍCIO

FIGURE 5.23: Text Exercise

FA04 - Filter exercises

For filtering the list of exercises a floating action menu was used and each button represents a type of exercise (video, audio and text). This menu can be seen on figure 5.24.


Mindfulness - Prática de 10 min

"Viva no Agora". Aproveite esta meditação guiada para esquecer tudo o resto e focar-se no presente. Desfrute desta meditação em qualquer momento do seu dia.

10 mins

Meditação das Emoções - Prof. Vicente Simón

Meditação guiada, da autoria do Prof. Vicente Simón, retirada do livro 'Mindfulness O Poder da Meditação - Técnicas Inspiradoras para Alcançar a Paz Interior', editora Nascente. Esta meditação guiada tem como objetivo o contacto com as suas emoções.

20 mins

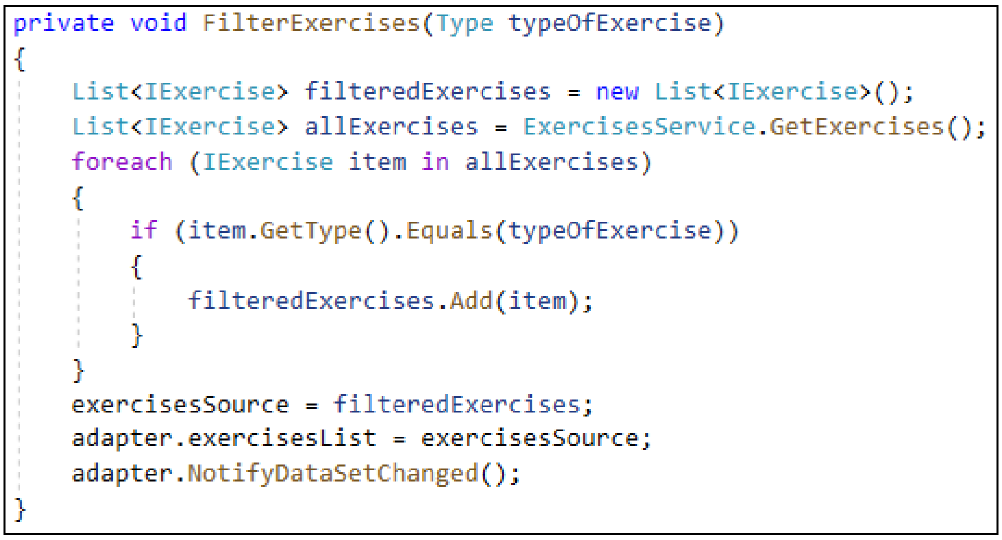
Meditação guiada para crianças

Breve meditação para crianças,

FIGURE 5.24: Filter Floating Action Menu

In order to implement the floating action menu a *NuGeT* package added to the project (“FAB.XamarinAndroid”, 2016), this package is based on the floating action menu created by Tarianyk that has more than 5000 stars (favorites) and 1000 forks (Tarianyk, 2020).

Each floating action button inside the menu is associated with a exercise type, when clicked the buttons will call the *FilterExercises* function that can be seen on figure 5.25 and the type of exercise will be passed as parameter.



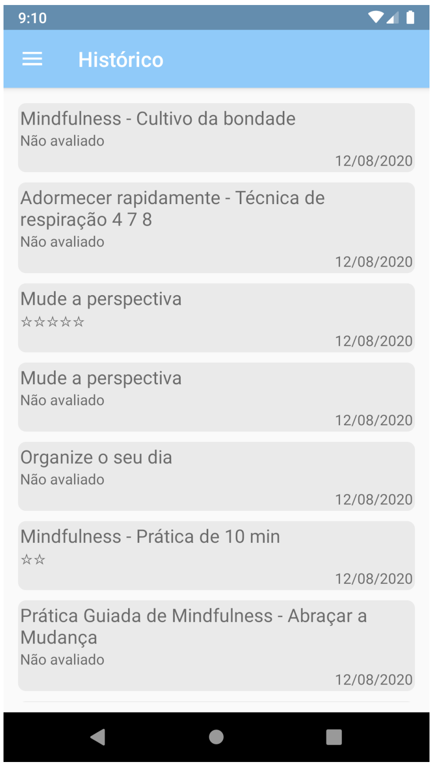
```
private void FilterExercises(Type typeOfExercise)
{
    List<IExercise> filteredExercises = new List<IExercise>();
    List<IExercise> allExercises = ExercisesService.GetExercises();
    foreach (IExercise item in allExercises)
    {
        if (item.GetType().Equals(typeOfExercise))
        {
            filteredExercises.Add(item);
        }
    }
    exercisesSource = filteredExercises;
    adapter.exercisesList = exercisesSource;
    adapter.NotifyDataSetChanged();
}
```

CODE SNIPPET 5.25: *FilterExercises* method

The function iterates all exercises and saves to a list the exercises that correspond to the type passed as a parameter. When all exercises are iterated then the adapter of the *ListView* of exercises receives the new filtered list and it is notified that the data changed. The interface then updates to show the new filtered list.

FA05 - Exercise history

The user can choose to see his exercise history which contains all exercises that the user did. In order for exercises to appear in this list the user must press the button to end the exercise which allows him to rate the exercise (as seen in FA06 - Rate exercises sub section). This list of exercises that the user did can be seen in the figure 5.26, it features the exercise name, the date that the user did the exercise and how many stars were rated, if any.


9:10

Histórico

Mindfulness - Cultivo da bondade
Não avaliado
12/08/2020

Adormecer rapidamente - Técnica de respiração 4 7 8
Não avaliado
12/08/2020

Mude a perspectiva
☆☆☆☆☆
12/08/2020

Mude a perspectiva
Não avaliado
12/08/2020

Organize o seu dia
Não avaliado
12/08/2020

Mindfulness - Prática de 10 min
☆☆
12/08/2020

Prática Guiada de Mindfulness - Abraçar a Mudança
Não avaliado
12/08/2020

FIGURE 5.26: Exercise History Interface

FA06 - Rate exercises

Inside any exercise there's a button at the bottom which allows the user to end and rate the exercise. This rating will be saved to the database and will be used for the recommendation system, these ratings are also used for the history use case mentioned above in the subsection 5.3.

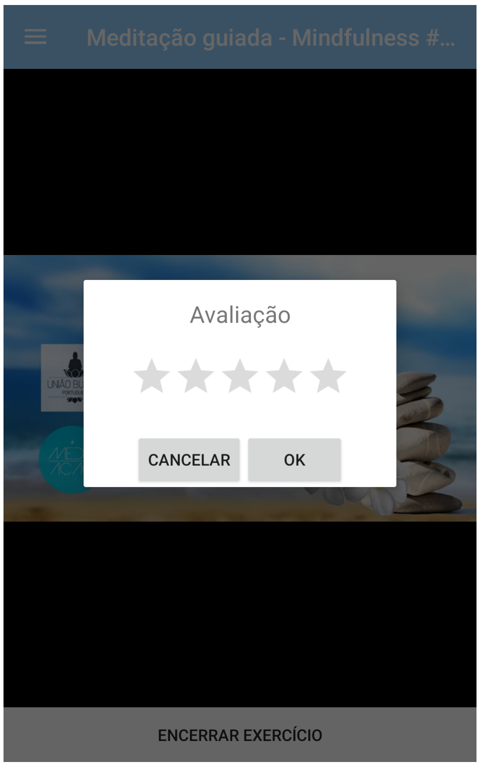
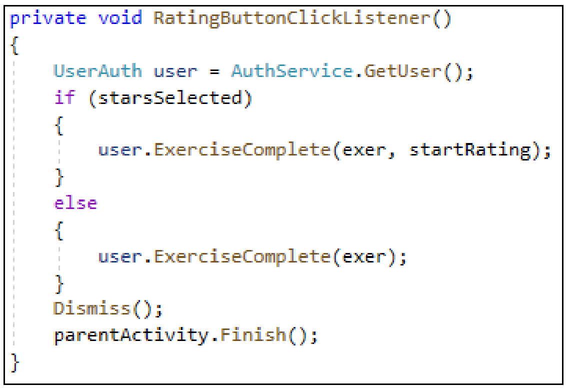


FIGURE 5.27: Rate modal window

In the code snippet 5.28 below it's possible to see the method that is invoked when the rate button inside the modal page is pressed. It's possible to end the exercise with or without stars, and when the rating is saved on the database the modal window is dismissed the parent activity is finished which means that the exercise page is also dismissed.

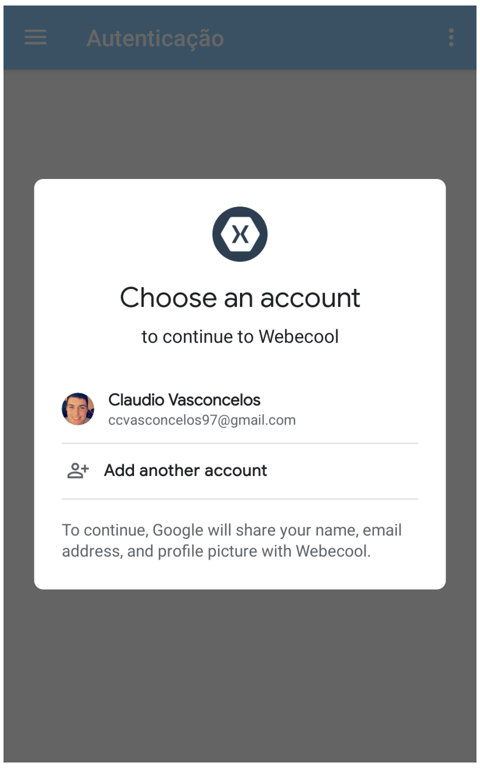


```
private void RatingButtonClickListener()
{
    UserAuth user = AuthService.GetUser();
    if (starsSelected)
    {
        user.ExerciseComplete(exer, startRating);
    }
    else
    {
        user.ExerciseComplete(exer);
    }
    Dismiss();
    parentActivity.Finish();
}
```

CODE SNIPPET 5.28: Method for rate button click

FP01 - Register

To register the user only needs to press the sign in button and if the user wasn't already on the database he will then be added. The sign in sequence diagram is present on figure 5.2, at one point of the sequence diagram an intent for selecting a google account is used, this intent can be observed in the figure 5.29.



To continue, Google will share your name, email address, and profile picture with Webecool.

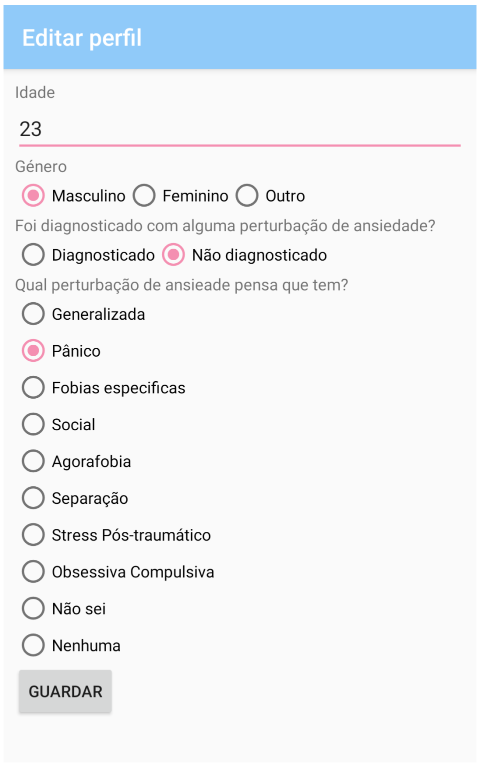
FIGURE 5.29: Sign In Intent

FP02 - Login

If the user is already on the database then when he signs in with the button mentioned in the FP01 - Register use case then the user is signed in and all database data of the user is used. And similarly to the register use case, the sequence diagram of this use case can be seen on figure 5.2.

FP03 - Setup profile

In this use case the user can add some information to his profile which will help the recommendation engine provide better predictions. In the figure 5.30 it's possible to see this use case, the user can provide his age, gender, whether he was diagnosed or not with an anxiety disorder. He then can choose an anxiety disorder from the list (the list contains all anxiety disorders discussed in the section 2.2), in case that the user wasn't diagnosed there's two additional options which are that the user has no anxiety disorder or doesn't know which one he has.


Editar perfil

Idade

23

Género

[✓ Masculino] Feminino Outro

Foi diagnosticado com alguma perturbação de ansiedade?

Diagnosticado [✓ Não diagnosticado]

Qual perturbação de ansieade pensa que tem?

Generalizada

[✓ Pânico]

Fobias especificas

Social

Agorafobia

Separação

Stress Pós-traumático

Obsessiva Compulsiva

Não sei

Nenhuma

FIGURE 5.30: Edit Profile Activity

Chapter 6

Evaluation

Given the inherent complexity in the process of developing software, there is a need to ensure rigorous quality control and evaluation of functionalities, thus allowing to measure the solution's degree of fulfillment of the objectives.

Since there is a need to outline a solid evaluation and quality control plan, both in meeting objectives and in the characteristics of the solution itself, it was decided to use a system of development cycles in the production phase, accompanied by the QEF model.

All software should be tested in order to have low instances of unpredictable behavior and ensure high quality software, as such the last section will show the tests that were evaluated.

6.1 Development, Testing and Evaluation Cycles

The creators of the QEF explained that "the QEF framework is not restricted to measure the final quality instead it allows for the evaluation of systems quality at any moment during is lifecycle." As such there will be various development cycles with the objective of evaluating the project quality, in each cycle the requirements will be measured and the QEF will be filled. (Escudeiro et al., 2006)

6.1.1 Alpha Cycle

The development and implementation of the desired solution starts in this phase, at the end of the phase the **Alpha Tests** occur. Each requirement on the QEF will be evaluated by acceptance tests. The alpha cycle has the objective of checking the prototype quality and show the developer what needs to be improved or developed next by identifying all possible issues/bugs.

In this cycle the mobile application and the database were built. Register, login and setup profile use cases were implemented and then exercises to relieve anxiety were added into de application. These use cases were then tested and some bugs were found and fixed.

6.1.2 Beta Cycle

During this phase the process of development continues with new insight from the previous alpha cycle. It is known what exactly needs to be done. At the end of the cycle the **Beta Tests** occur and unlike Alpha Tests there should be participation from external users that will aid in testing the application. These users will use the application and answer questionnaires that are directly related to specific non functional requirements in the QEF. This phase can be considered a form of external User Acceptance Testing, in short, it reduces product failure risks and provides increased quality of the product through customer validation.

In this cycle the recommendation system was created, some use cases like filter exercises were added and other minor improvements were implemented. In order to test the recommendation system a external data set was used, more details on testing the recommendation system can be found on section 6.3. Unfortunately in this cycle the objective of using external users to test the application and answer questionnaires proved to be a challenge due to the COVID-19 pandemic.

6.1.3 Release Cycle

After the Beta Tests the Release Cycle occurs where small improvements and fixes are made and the development process is finalized. After this cycle future work could be made to add more functionality into the application, more information about that can be found in the section 7.2.

6.2 Quantitative Evaluation Framework (QEF)

To carry out the evaluation of the solution, the use of QEF was adopted.

QEF is a framework that allows quantitative assessment of software quality and it was developed by Instituto Superior de Engenharia do Porto (ISEP) professors (Escudeiro et al., 2006). For this project, the QEF has three dimensions: Functional, Adaptability and Usability that represent our scenario of quality. Each dimension has a set of factors, and each factor has a set of requirements, with everything combined it is possible determine the degree of performance of our project.

6.2.1 Functionality

The dimension of functionality has the five factors and from each one a set of requirements was determined, these requirements are related to the functional requirements discussed in the section 4.2.1.

- Interaction with Therapists
- Anxiety Exercises
- Recommendation System
- Profile
- Content Quality

6.2.2 Adaptability

This dimension has the objective of measuring the level of adaptability of the project, some of the requirements present in this dimension are related to the non-functional requirements discussed in the section 4.2.2. Two factors were identified and will be evaluated.

- Versatility
- Maintenance

6.2.3 Efficiency

The dimension of usability was created to determine the level efficiency and usability. Some of the requirements from this dimension can't be easily determined, so some are measured by questionnaires which will show how users think and feel about the application.

- Navigation
- Support

6.2.4 Results

The results of the QEF can be observed in the QEF. Even with 0% on every requirement related to questionnaires the results of the QEF were positive with the final result of 74% which is a 15 in a 20-point scale. The functionality and adaptability dimensions were the ones with highest scores, mostly due to the lack of questionnaires but despite this the Usability dimension was able to reach 50%.

6.3 Tests

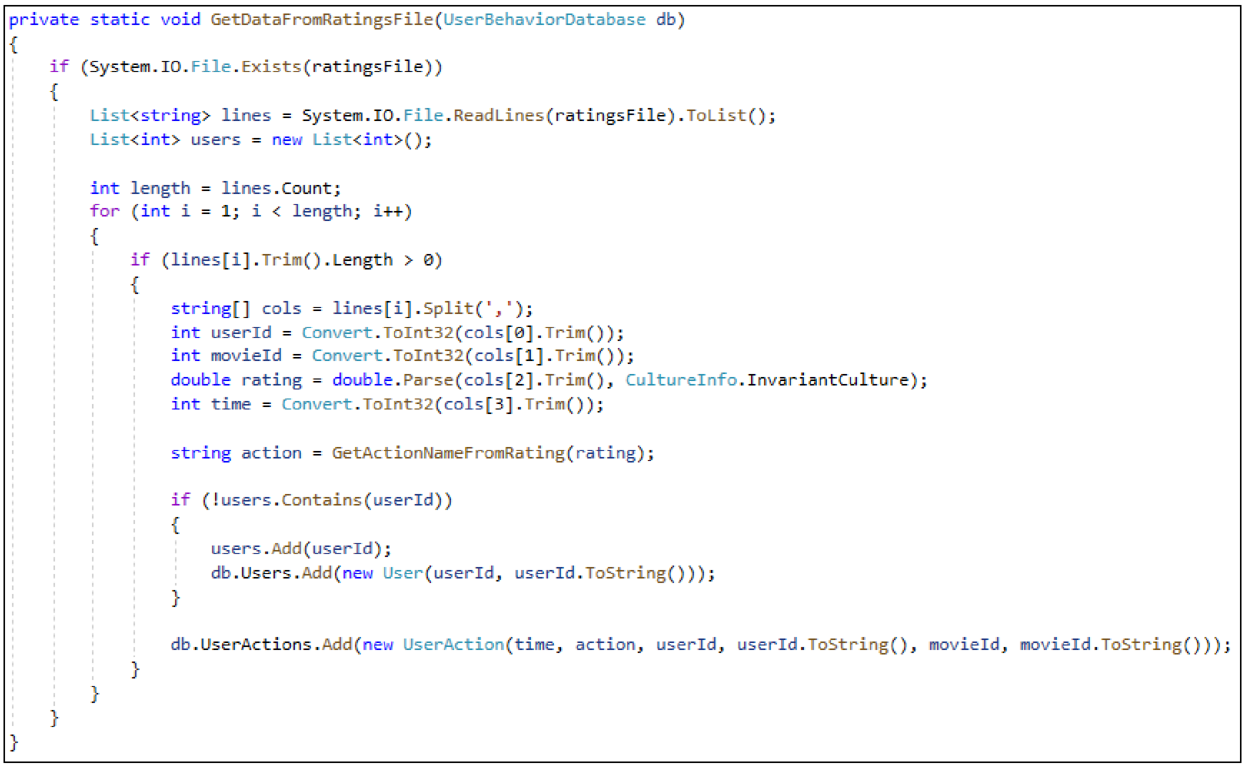
In order to ensure the quality and reliability of this software there will be tests. Testing will help find defects and bugs during development, but it needs to be widespread through all the software.

Testability is a requirement for the software design, the design must allow the existence of unit tests throughout all the code. Unit tests can also be combined and tested as a group with the purpose of exposing faults in the interaction between integrated units, this kind of combined tests are called integration tests and will also be performed. Acceptance tests will also be performed and were already mentioned in the section 6.1.

The project is expected to contain at least one API that will serve as communication between the server and the mobile application, as such it must also be tested. The reliability, and security must be assured by these tests. POSTMAN can be used for these tests since one of it's main features is its ability to run automated tests.

The dataset became a considerable limitation for the present system, since obtaining data regarding exercise ratings by different users, or related data, in a short time, was an inconceivable task, as such the MovieLens Dataset ("MovieLens", 2013) was chosen because of its wide use in education, research and industry. It is downloaded hundreds of thousands of times a year, reflecting its use in popular programming books, traditional and online courses, and in software (Harper & Konstan, 2015). The dataset is characterized by containing more than 27 million ratings from 58 thousand different films by 280 thousand users. It mainly presents two files in .csv: user ratings in the <userID, itemID, rating, timestamp> format and movies in the <itemID, title, tags> format.

Then, a specific import logic was created for each file. Code snippet 6.1 shows a code excerpt for importing the user's ratings file.

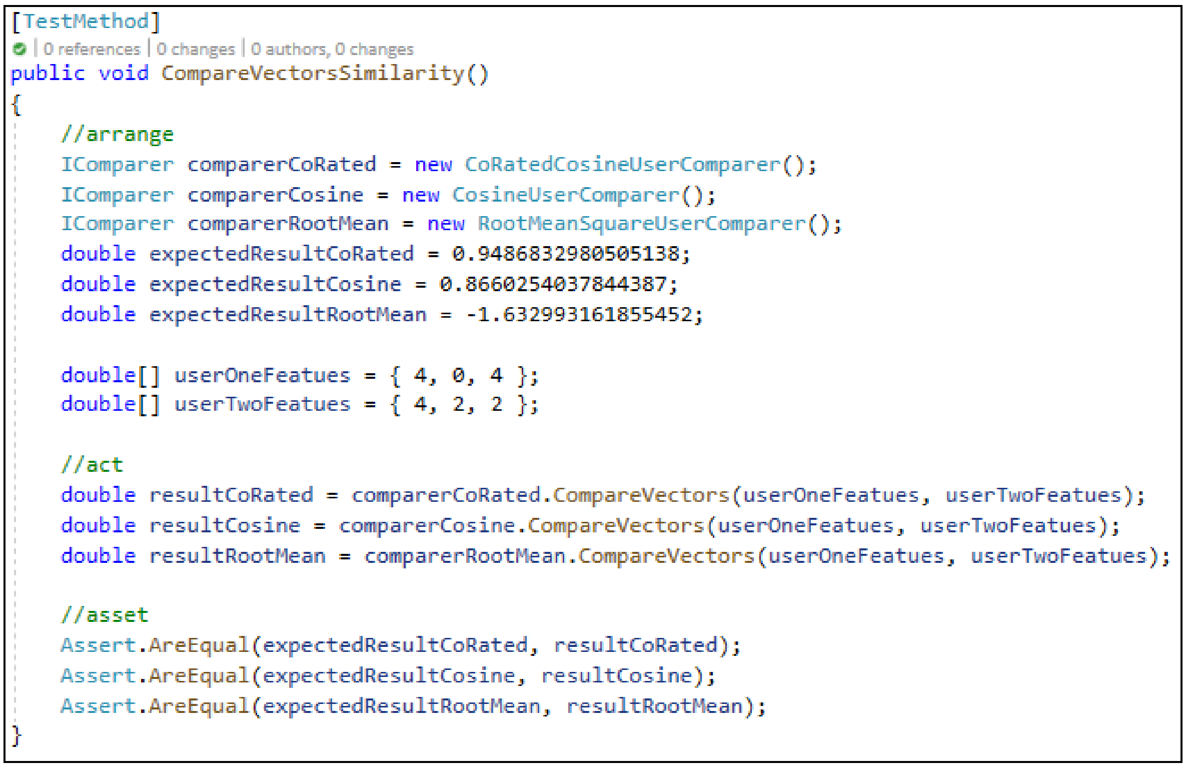


```
private static void GetDataFromRatingsFile(UserBehaviorDatabase db)
{
    if (System.IO.File.Exists(ratingsFile))
    {
        List<string> lines = System.IO.File.ReadLines(ratingsFile).ToList();
        List<int> users = new List<int>();

        int length = lines.Count;
        for (int i = 1; i < length; i++)
        {
            if (lines[i].Trim().Length > 0)
            {
                string[] cols = lines[i].Split(',');
                int userId = Convert.ToInt32(cols[0].Trim());
                int movieId = Convert.ToInt32(cols[1].Trim());
                double rating = double.Parse(cols[2].Trim(), CultureInfo.InvariantCulture);
                int time = Convert.ToInt32(cols[3].Trim());

                string action = GetActionNameFromRating(rating);

                if (!users.Contains(userId))
                {
                    users.Add(userId);
                    db.Users.Add(new User(userId, userId.ToString()));
                }

                db.UserActions.Add(new UserAction(time, action, userId, userId.ToString(), movieId, movieId.ToString()));
            }
        }
    }
}
```

Code Snippet 6.1: Import Logic for MovieLens Ratings

6.3.1 Unit tests

The unit testing framework of Visual Studio was used to prepare the unit tests. Therefore, test classes were defined for each unit tested (class, controller or others). As an example, code snippet 6.2 shows the implementation of a unit test present in this project.



```
[TestMethod]
| 0 references | 0 changes | 0 authors, 0 changes
public void CompareVectorsSimilarity()
{
    //arrange
    IComparer comparerCoRated = new CoRatedCosineUserComparer();
    IComparer comparerCosine = new CosineUserComparer();
    IComparer comparerRootMean = new RootMeanSquareUserComparer();
    double expectedResultCoRated = 0.9486832980505138;
    double expectedResultCosine = 0.8660254037844387;
    double expectedResultRootMean = -1.632993161855452;

    double[] userOneFeatues = { 4, 0, 4 };
    double[] userTwoFeatues = { 4, 2, 2 };

    //act
    double resultCoRated = comparerCoRated.CompareVectors(userOneFeatues, userTwoFeatues);
    double resultCosine = comparerCosine.CompareVectors(userOneFeatues, userTwoFeatues);
    double resultRootMean = comparerRootMean.CompareVectors(userOneFeatues, userTwoFeatues);

    //asset
    Assert.AreEqual(expectedResultCoRated, resultCoRated);
    Assert.AreEqual(expectedResultCosine, resultCosine);
    Assert.AreEqual(expectedResultRootMean, resultRootMean);
}
```

Code Snippet 6.2: Similarity Unit Test

As illustrated in the code snippet, the code is organized according to the Arrange, Act and Assert (AAA) test writing standard (Gomes, 2018). The standard divides the unit testing code into three phases:

1. **Arrange** — prepare all preconditions and inputs;
2. **Act** — invoking the method under test, saving and capturing the resulting state;
3. **Assert** — verify the results through assertions.

This division clearly separates what is being tested from the configuration and verification steps, and organizes the code so that it is easy to understand what is being tested, thus making the tests more intuitive.

6.3.2 Acceptance Tests

Acceptance tests are formal tests to determine whether a system meets the acceptance criteria. These allow the customer to determine whether or not to accept the system. Thus, acceptance tests were developed based on usage scenarios to test aspects of the system's use. Each acceptance test tests a feature - use case - and there may be more than one test for a use case.

In the table 6.1 it's possible to see the acceptance tests of all use cases of this project.

Use cases	Status
FR01 - Request exercise recommendations	Passed
FR02 - Receive recommendations with the use of machine learning	Passed
FR03 - Notification with recommendation	Failed
FR04 - Recommend exercise to patient	Failed
FA01 - Exercise with video	Passed
FA02 - Exercise with audio	Passed
FA03 - Exercise with text	Passed
FA04 - Filter exercises	Passed
FA05 - Exercise history	Passed
FA06 - Rate exercises	Passed
FI01 - Chat communication with other users	Failed
FI02 - Send photo/video/audio in chat	Failed
FI03 - Report symptoms	Failed
FP01 - Register	Passed
FP02 - Login	Passed
FP03 - Setup profile	Passed
FP04 - Define user role	Failed
FP05 - Add information to patient profile	Failed

TABLE 6.1: Acceptance Tests

6.3.3 Response time

A test was performed in order to analyze the execution time of the algorithm responsible for the training and requesting recommendations (collaborative recommendation technique). This test was performed by POSTMAN twenty times on the recommendation system API that's hosted on Microsoft Azure. This was tested using the MovieLens dataset which has over 250 thousand possible items that could be recommended.

The average, the lowest and the highest response time value of results are present in table 6.2.

	Lowest	Highest	Average
Training	81ms	132ms	104ms
Getting recommendations	61ms	87ms	68ms

TABLE 6.2: Response time of recommendation system

In view of a large amount of data, the test result was quite positive, with no significant variation between the times obtained, and all of them respect are below 2 seconds which count as an achieved non functional performance requirement.

6.3.4 Prediction Errors

In order to calculate de prediction the prediction error two metrics will be applied and their results will be presented and analyzed in this sections. These are the two metrics:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{6.1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{6.2}$$

where: n = number of ratings
Y_i = predicted value
$\hat{Y}_i$ = real value

Prediction Error of Each Recommendation Technique

All the implemented recommendation techniques will be evaluated using the metrics MAE and RMSE, these are the techniques:

- **Collaborative filtering** — based on users (UCF) and on items (ICF)
- **Matrix Factorization (MF)**
- **Hybrid** —all possible combinations

For the present test, no variations were made in the variables of the SR techniques algorithms. Therefore, in Matrix Factorization the number of iterations is equal to 100 and the number of latent features is equal to 20. In collaborative filtering, the number of similar or neighbor users/items is equal to 20.

Regarding the similarity calculation metric used in the collaborative technique, the Co-Rated Cosine Similarity was used, since it is the measure that obtained the least prediction error. This subject is discussed in greater detail in the subsection 6.3.4.

With only one iteration performed in order to determine the prediction error of the collaborative and hybrid techniques (UCF + ICF). On the other hand, thirty iterations were performed using Matrix Factorization and hybrid techniques (MF + UCF, MF + ICF and MF + UCF + ICF), and in the end the average was calculated.

Recommendation Technique	MAE	RMSE	RMSE - MAE	$\frac{\text{MAE + RMSE}}{2}$
MF	0,77	1,01	0,24	0,89
UCF	1,80	2,15	0,35	1,98
ICF	0,91	1,11	0,20	1,01
MF and UCF	1,55	1,78	0,23	1,67
MF and ICF	0,78	0,98	0,20	0,88
UCF and ICF	1,79	1,95	0,16	1,87
MF, UCF and ICF	1,42	1,59	0,17	1,51

TABLE 6.3: Prediction Error of the Different Recommendation Techniques

Table 6.3 shows the values resulting from the application of the metrics to each recommendation technique. As it can be seen, the technique with the least prediction error is "MF and ICF", followed by "MF". Both have a common technique — Matrix Factorization. On the other hand, the technique that has a most prediction error is collaborative filtering based on users.

The reasons that justify the difference in the values between the techniques are not evident or easy to deduce. They may be related to details of the algorithms, dataset size, relationships between users and items, among others. However, it can be concluded that under the conditions described, excluding the combinations between the techniques, Matrix Factorization was the technique that obtained the least prediction error.

Analyzing the combinations between the various techniques (hybrid), the one that obtained the least prediction error was the combination between the Matrix Factorization techniques and collaborative filtering based on items. This result is totally understandable, because, as analyzed above, Matrix Factorization and the item-based collaborative filtering were the techniques, excluding the hybrids, that obtained lower errors.

Similarity Prediction Error

For this experiment four tests were performed to calculate the prediction error using the Mean Absolute Error (MAE) and RMSE metrics in the collaborative filtering technique based on users using the four similarity calculation measures.

Comparer	MAE	RMSE
Cosine Similarity	2,04	2,33
Co-Rated Cosine Similarity	1,80	2,15
Pearson Correlation Similarity	3,10	3,99
Root Mean Square Similarity	3,15	3,37

TABLE 6.4: Prediction Error of the Similarity Somparers

Observing the results of the prediction errors for each calculation present in Table 6.4, it is possible to conclude that the Cosine technique and the Co-Rated Cosine variation, are the ones that present the lowest prediction error, both in the MAE and RMSE metrics. On the other hand, Root Mean Square Similarity has a higher error in the MAE metric and Pearson Correlation Similarity a greater error in the RMSE metric.

6.4 Summary

The plan of evaluation and quality control was presented in this chapter. It will contain three development and evaluation cycles that will be assisted by the use of the Quantitative Evaluation Framework. The last section of this chapter showcases the tests performed on the webecool system.

Chapter 7

Conclusions

This dissertation had the main objective of developing a platform where users could perform exercises to relieve their anxiety, this platform is composed of a an application and a recommendation system who is able to predict exercises that will have a positive impact on the user.

Firstly the theme of this project was contextualized and the objectives were defined, then there was a deeper dive into the state of the art of the recommendation systems and anxiety disorders, then some options for the user application were analyzed and chosen. Then the design was established, the webecool system would have a mobile application written in Microsoft Xamarin, a NoSQL database hosted on Firebase and an API on Azure for the recommendation system. More details on the implementation of those three components was present on the Implementation chapter.

In the first phase of the implementation the mobile application and the database has more focus, the anxiety exercises and the register/login use cases were developed early on, then the recommendation system was developed. This component was the most difficult due to machine learning concepts and algorithms, complicated mathematical equations and the implementation of different recommendation techniques. However there was one main limitation, the lack of data. In order to test the recommendation system and it's different techniques a large data set (MovieLens) was entered into the system and tested in order to fully trust the recommendation even when there's low amount of data.

Below, the objectives will be discussed and the achieved objectives will be identified, as well as some limitations of this system and some future work that could improve the application and the other components.

7.1 Achieved objectives

In this section the objectives of this dissertation and the Webecool system will be discussed. In the table 7.1 there are the objectives present in the chapter Introduction and the corresponding status.

Objective	Status
Investigation and analysis of various mental health mobile applications	Achieved
Investigation and analysis of various recommendation systems	Achieved
Identify solutions to the problem described in the section Problem	Achieved
Design an architecture that satisfies the identified requirements	Achieved
Implementation of a mobile application that provides exercises to alleviate anxiety	Achieved
Implementation of a recommendation system that suggests exercises to users through the use of machine learning	Achieved
Integration of caregivers and therapists	Not achieved
Conduct a study that demonstrates the utility of the developed application, followed by its analysis	Not achieved

Table 7.1: Dissertation Objectives

As shown in the table 7.1, the objectives — Integration of caregivers and therapists; Conduct a study that demonstrates the utility of the developed application, followed by its analysis — were the only ones not to be reached. Despite being considered important parts for the present system, other objectives were given a higher priority, such as the recommendation system for example. Conducting a study to demonstrate the utility of the application proved to be a challenge during the COVID-19 pandemic, the plan was to select a group of subjects at convenience that could use the application for a few days and then report back their experience through a questionnaire. Even though this important objective was not reached it was still possible to evaluate the Webecool system by performing acceptance tests and tests on the response time and the prediction error of the recommendation system.

One of the greatest limitations of this project was related to the lack of data regarding training classifications by different users, or related data. But its possible to consider that this limitation was overcome by the use of an existing dataset (MovieLens), which allowed to recognize the good performance of the developed system.

Use cases	Priority	Status
FR01 - Request exercise recommendations	5	Implemented
FR02 - Receive recommendations with the use of machine learning	5	Implemented
FR03 - Notification with recommendation	4	Not Implemented
FR04 - Recommend exercise to patient	3	Not Implemented
FA01 - Exercise with video	5	Implemented
FA02 - Exercise with audio	5	Implemented
FA03 - Exercise with text	5	Implemented
FA04 - Filter exercises	4	Implemented
FA05 - Exercise history	5	Implemented
FA06 - Rate exercises	5	Implemented
FI01 - Chat communication with other users	3	Not Implemented
FI02 - Send photo/video/audio in chat	1	Not Implemented
FI03 - Report symptoms	3	Not Implemented
FP01 - Register	5	Implemented
FP02 - Login	5	Implemented
FP03 - Setup profile	5	Implemented
FP04 - Define user role	3	Not Implemented
FP05 - Add information to patient profile	2	Not Implemented

Table 7.2: Use Cases Status

In the table 7.2 it's possible to see the list of implemented use cases which are 61% of all use cases. More than half were implemented but it's also important to take into account the implementation rate among the use cases with the highest priority, that is, the use cases with priority 4 or 5. Among 12 use cases with high priority, 11 where implemented, making it 91% implementation rate.

7.2 Future Work

Despite all development efforts to achieve objectives and mitigate their limitations, it is generally not possible to meet all requirements and limitations. In fact, a project can hardly be considered as finished and complete, since there are always improvements and modifications that can be made.

Since the system has several components, the future work to be employed will be indicated for each of them in the sub sections below.

7.2.1 Mobile Application

The first future work improvements would be the implementation of the remaining use cases, this would mean that the application would now have caregivers and therapists which would be able to provide more data about the user to the recommendation engine and they would also be able to manually recommend exercises.

The current amount of exercises to relieve anxiety is low and to increase it one possible solution would be for therapists creating exercises. The application could have a functionality which would allow therapists to create video, audio or text exercises. This would contribute to a more rich application with better resources for the user better self manage their anxiety.

The iOS version of the app could also be created and tested instead of just Android, especially since it is developed in hybrid technology that allows development for iOS and Android (Microsoft Xamarin), this would also create an opportunity to improve the UI in order to provide a better experience to the user.

7.2.2 Recommendation System

As mentioned previously, the lack of data was a limitation and an external dataset had to be used for testing. A future improvement for the recommendation system would be gathering more information about the user in order to gather enough data to have a more complete dataset that can be tested. The recommendation system could have some integration with applications like Google Fit that would provide the recommendation system more data about the user which would generate better predictions.

Another future task that could be performed would the creation of more tests in order to fully comprehend how accurate and intelligent the predictions really are.

If therapists and caregivers were integrated into the webecool system then the recommendation system could take into account their manual recommendations to users, the data they would provide about the user and the exercises created by therapists would also create more exercise diversity and improve the recommendations.

Bibliography

ADAA. (n.d.-a). Therapy | Anxiety and Depression Association of America, ADAA. Retrieved February 15, 2020, from https://adaa.org/finding-help/treatment/therapy

ADAA. (n.d.-b). Understand the Facts | Anxiety and Depression Association of America, ADAA. *ADAA*. Retrieved June 13, 2020, from https://adaa.org/understanding-anxiety

ADAA Managing Stress and Anxiety. (n.d.). Retrieved December 16, 2019, from https://adaa.org/living-with-anxiety/managing-anxiety

ADAA Reviewed Mental Health Apps. (n.d.). Retrieved December 2, 2019, from https://adaa.org/finding-help/mobile-apps

Agarwal, A., Chauhan, M., & Ghaziabad. (2017). *Similarity Measures used in Recommender Systems : A Study*. Retrieved September 30, 2020, from /paper/Similarity-Measures-used-in-Recommender-Systems-%3A-A-Agarwal-Chauhan/943ae455fafc3d36ae4ce68f1a60ae4f85623e

Aggarwal, C. C. (2016). An Introduction to Recommender Systems. In C. C. Aggarwal (Ed.), *Recommender Systems: The Textbook* (pp. 1–28). Springer International Publishing. https://doi.org/10.1007/978-3-319-29659-3_1

Ali, Q. (2020, August 7). *Strategy Design Pattern in Depth*. Retrieved September 27, 2020, from https://medium.com/@iamqamarali/strategy-design-pattern-in-depth-32750ae0ad6

Anthes, E. (2016). Mental health: There's an app for that. *Nature*, *532*(7597), 20–23. https://doi.org/10.1038/532020a

Anxiety Canada - MindShift CBT. (n.d.). Retrieved December 3, 2019, from https://anxietycanada.com/articles/new-mindshift-cbt-app-gives-canadians-free-anxiety-relief/

AnxietyCoach. (n.d.). *App Store*. Retrieved December 10, 2019, from https://apps.apple.com/us/app/anxietycoach/id565943257

Azure, M. (n.d.). *Pricing OverviewHow Azure Pricing Works | Microsoft Azure*. Retrieved September 17, 2020, from https://azure.microsoft.com/en-us/pricing/

BetterHelp Website. (n.d.). Retrieved November 30, 2019, from https://www.betterhelp.com/faq/

Betterhelp, Talkspace, LARKR - Google Trends. (n.d.). Retrieved December 17, 2019, from https://trends.google.com/trends/explore?date=today%205-y&q=Betterhelp,Talkspace,LARKR

Bigio, M. (2016). *Introducing Hot Reloading - React Native*. Retrieved December 4, 2019, from https://facebook.github.io/react-native/blog/2016/03/24/introducing-hot-reloading.html

Bricker, J. B., Mull, K., Kientz, J. A., Vilardaga, R. M., Mercer, L. D., Akioka, K., & Heffner, J. L. (2014). Randomized, Controlled Pilot Trial of a Smartphone App for Smoking Cessation Using Acceptance and Commitment Therapy. *Drug and alcohol dependence*, *143*, 87–94. https://doi.org/10.1016/j.drugalcdep.2014.07.006

Britch, D., & Johnson, J. (2019). *What is Xamarin? - Xamarin | Microsoft Docs*. Retrieved December 3, 2019, from https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, *12*. https://doi.org/10.1023/A:1021240730564

Burke, R., Felfernig, A., & Göker, M. (2011). Recommender Systems: An Overview. *Ai Magazine*, *32*, 13–18. https://doi.org/10.1609/aimag.v32i3.2361

Calderaio, J. A. (2017). *Comparing the Performance between Native iOS (Swift) and React-Native*. Retrieved December 4, 2019, from https://medium.com/the-react-native-log/comparing-the-performance-between-native-ios-swift-and-react-native-7b5490d363e2

Çano, E., & Morisio, M. (2017). Hybrid Recommender Systems: A Systematic Literature Review. *Intelligent Data Analysis*, *21*(6), 1487–1524. https://doi.org/10.3233/IDA-163209

Capan, T. (2013, August). *Why The Hell Would I Use Node.js? A Case-by-Case Tutorial*. Retrieved September 26, 2020, from https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js

Chandrashekar, P. (2018). Do mental health mobile apps work: Evidence and recommendations for designing high-efficacy mental health mobile apps. *mHealth*, *4*, 6–6. https://doi.org/10.21037/mhealth.2018.03.02

Cheung, K., Ling, W., Karr, C. J., Weingardt, K., Schueller, S. M., & Mohr, D. C. (2018). Evaluation of a recommender app for apps for the treatment of depression and anxiety: An analysis of longitudinal user engagement. *Journal of the American Medical Informatics Association*, *25*(8), 955–962. https://doi.org/10.1093/jamia/ocy023

Chrzanowska, N. (2017). *React Native vs Swift - Performance and Development Comparison | Netguru Blog on React Native*. Retrieved December 4, 2019, from https://www.netguru.com/blog/react-native-vs-swift-performance-development-comparison

Clayton, S. (2018, March 22). *Building a Recommendation Engine in C#*. Retrieved October 2, 2020, from https://www.codeproject.com/Articles/1232150/Building-a-Recommendation-Engine-in-Csharp

Cognitive behavioral therapy - Mayo Clinic. (n.d.). Retrieved December 17, 2019, from https://www.mayoclinic.org/tests-procedures/cognitive-behavioral-therapy/about/pac-20384610

Coutsias, E. A., Seok, C., & Dill, K. A. (2004). Using quaternions to calculate RMSD. *Journal of Computational Chemistry*, *25*(15), 1849–1857. https://doi.org/10.1002/jcc.20110

Cowart, J. (n.d.). *What is a Hybrid Mobile App?* Retrieved December 3, 2019, from https://www.telerik.com/blogs/what-is-a-hybrid-mobile-app-

DashMagazine. (2019). *Mobile App Development Frameworks in 2019 - codeburst*. Retrieved December 3, 2019, from https://codeburst.io/mobile-app-development-frameworks-in-2019-f8fb2ece20a8

dataaspirant. (2015). An Introduction to Recommendation Engines. *Dataconomy*. Retrieved February 23, 2020, from https://dataconomy.com/2015/03/an-introduction-to-recommendation-engines/

Economides, M., Martman, J., Bell, M. J., & Sanderson, B. (2018). Improvements in Stress, Affect, and Irritability Following Brief Use of a Mindfulness-Based Smartphone App: A Randomized Controlled Trial. *Mindfulness*, *9*(5), 1584–1593. https://doi.org/10.1007/s12671-018-0905-4

Eeles, P. (2004). What, no supplementary specification? *IBM*. Retrieved February 12, 2020, from http://www.ibm.com/developerworks/rational/library/3975.html

Eeles, P. (2005). Capturing Architectural Requirements. *IBM*. Retrieved February 12, 2020, from http://www.ibm.com/developerworks/rational/library/4706.html

Escudeiro, P., Bidarra, J., & Escudeiro, N. (2006). Evaluating educational software.

FAB.XamarinAndroid. (2016). Retrieved October 4, 2020, from https://www.nuget.org/packages/FAB.XamarinAndroid/

Firebase. (n.d.). *Firebase Pricing*. Retrieved September 17, 2020, from https://firebase.google.com/pricing

Firth, J., & Torous, J. (2015). Smartphone Apps for Schizophrenia: A Systematic Review. *JMIR mHealth and uHealth*, *3*(4), e102. https://doi.org/10.2196/mhealth.4930

Firth, J., Torous, J., Nicholas, J., Carney, R., Pratap, A., Rosenbaum, S., & Sarris, J. (2017). The efficacy of smartphone-based mental health interventions for depressive symptoms: A meta-analysis of randomized controlled trials. *World Psychiatry*, *16*(3), 287–298. https://doi.org/10.1002/wps.20472

Gawron, K. (2018, October 11). *What is Node.js and Why You Should Use It For Enterprise Software?* Retrieved September 26, 2020, from https://www.monterail.com/blog/nodejs-development-enterprises

Goel, A. (2019). Top 10 Web Development Frameworks [Updated]. *Hackr.io*. Retrieved December 17, 2019, from https://hackr.io/blog/top-10-web-development-frameworks-in-2019

Gomes, P. (2018, January 13). *Unit Testing and the Arrange, Act and Assert (AAA) Pattern*. Retrieved October 13, 2020, from https://medium.com/@pjbgf/title-testing-code-ocd-and-the-aaa-pattern-df453975ab80

Google. (n.d.). *Flutter performance profiling - Flutter*. Retrieved December 4, 2019, from https://flutter.dev/docs/perf/rendering/ui-performance

Google. (2019). *Technical Overview: Flutter*. Retrieved December 4, 2019, from https://flutter.dev/docs/resources/technical-overview

Google.Cloud.Firestore .NET client library for the Firestore API. (n.d.). Retrieved September 24, 2020, from https://googleapis.github.io/google-cloud-dotnet/docs/Google.Cloud.Firestore/index.html#authentication

Grimaldi, E. (2018). How to build a content-based movie recommender system with Natural Language Processing. *Medium*. Retrieved February 23, 2020, from https://towardsdatascience.com/how-to-build-from-scratch-a-content-based-movie-recommender-with-natural-language-processing-25ad400eb243

Hale, T. (2019). *What Is The World's Most Relaxing Color? A New Survey Just Found Out*. Retrieved October 10, 2020, from https://www.iflscience.com/editors-blog/what-is-the-worlds-most-relaxing-color-a-new-survey-just-found-out/

Harper, F. M., & Konstan, J. A. (2015). The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, *5*(4), 19:1–19:19. https://doi.org/10.1145/2827872

Headspace Google Play page. (n.d.). Retrieved December 1, 2019, from https://play.google.com/store/apps/details?id=com.getsomeheadspace.android&hl=pt_PT

Headspace website. (n.d.). Retrieved December 1, 2019, from https://www.headspace.com/subscriptions

Home Page. (n.d.). *Larkr: On-Demand Mental Health Care*. Retrieved December 10, 2019, from https://larkr.com/

How to Choose a Technology Stack for Web App Development. (2019). Retrieved December 17, 2019, from https://dev.to/2muchcoffeecom/how-to-choose-a-technology-stack-for-web-app-development-13fg

Howells, A., Ivtzan, I., & Eiroa-Orosa, F. J. (2016). Putting the 'app' in Happiness: A Randomised Controlled Trial of a Smartphone-Based Mindfulness Intervention to Enhance Wellbeing. *Journal of Happiness Studies*, *17*(1), 163–185. https://doi.org/10.1007/s10902-014-9589-1

Hung, G. C.-L., Yang, P.-C., Wang, C.-Y., & Chiang, J.-H. (2015). A Smartphone-Based Personalized Activity Recommender System for Patients with Depression. *EAI Endorsed Trans. Cognitive Communications*, *2*, e5. https://doi.org/10.4108/eai.14-10-2015.2261655

Introducing JSX - React. (n.d.). https://reactjs.org/docs/introducing-jsx.html

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261–273. https://doi.org/10.1016/j.eij.2015.06.005

João Palma Budadharma. (n.d.). Retrieved October 10, 2020, from https://www.budadharma.org/home/quem-somos-2/joao-palma-mindfulness/

Katakam, N. (2019, January 7). *How Can We Design An Intelligent Recommendation Engine?* Retrieved October 2, 2020, from https://uxplanet.org/how-can-we-design-an-intelligent-recommendation-engine-b9bb1db4d050

Keene, L. (n.d.). *Creating Your Software Requirements*. Retrieved September 25, 2020, from https://www.keenesystems.com/blog/creating-your-software-requirements

Khoury, B., Sharma, M., Rush, S. E., & Fournier, C. (2015, June). *Mindfulness-based stress reduction for healthy individuals: A meta-analysis* (Vol. 78). Elsevier Inc. https://doi.org/10.1016/j.jpsychores.2015.03.009

Kirwin, W. (2016, January 11). *Implicit Recommender Systems - Biased Matrix Factorization - Activision Game Science*. Retrieved September 30, 2020, from http://activisiongamescience.github.io/2016/01/11/Implicit-Recommender-Systems-Biased-Matrix-Factorization/

Koen, P., Ajamian, G., Burkart, R., Clamen, A., Davidson, J., D'Amore, R., Elkins, C., Herald, K., Incorvia, M., Johnson, A., Karol, R., Seibert, R., Slavejkov, A., & Wagner, K. (2001). Providing Clarity and A Common Language to the Fuzzy Front End. *Research-Technology Management*, *44*(2), 46–55. https://doi.org/10.1080/08956308.2001.11671418

Koen, P. A., Ajamian, G., Boyce, S. D., Clamen, A., Fisher, E. S., Fountoulakis, S. G., Johnson, A., Puri, P. S., & Seibert, R. (2002). Fuzzy Front End : Effective Methods , Tools , and Techniques. Retrieved February 19, 2020, from https://www.semanticscholar.org/paper/1-Fuzzy-Front-End-%3A-Effective-Methods-%2C-Tools-%2C-and-Koen-Ajamian/e6b921c6125984aede5285e23f23b03efa95b967

Kordík, P. (2018). Machine Learning for Recommender systems Part 1 (algorithms, evaluation and cold start). *Medium*. Retrieved February 23, 2020, from https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1-algorithms-evaluation-and-cold-start-6f696683d0ed

Kruchten, P. (1995). The 4+1 View Model of architecture. *IEEE Software*, *12*(6), 42–50. https://doi.org/10.1109/52.469759

LARKR On-Demand Mental Health. (n.d.). *App Store*. Retrieved December 10, 2019, from https://apps.apple.com/us/app/larkr-on-demand-mental-health/id1253710426

Lorig, K., Ritter, P., Laurent, D., & Plant, K. (2006). Internet-Based Chronic Disease Self-Management: A Randomized Trial. *Medical care*, *44*, 964–71. https://doi.org/10.1097/01.mlr.0000233678.80203.c1

Ly, K. H., Asplund, K., & Andersson, G. (2014). Stress management for middle managers via an acceptance and commitment-based smartphone application: A randomized controlled trial. *Internet Interventions*, *1*(3), 95–101. https://doi.org/10.1016/j.invent.2014.06.003

Ly, K. H., Topooco, N., Cederlund, H., Wallin, A., Bergstrom, J., Molander, O., Carlbring, P., & Andersson, G. (2015). Smartphone-supported versus full behavioural activation for depression: A randomised controlled trial. *PLoS ONE*, *10*(5). https://doi.org/10.1371/journal.pone.0126559

Lynch, M. (2019). *Introducing Ionic 4: Ionic for Everyone | The Ionic Blog*. Retrieved December 4, 2019, from https://blog.ionicframework.com/introducing-ionic-4-ionic-for-everyone/

Marcelle, E., & Davis, T. S. (2017). BetterHelp members experience significant reduction in depression symptoms: Viable alternative to traditional face-to-face counseling. *White Paper*, 1–24. https://www.betterhelp.com/study/Study_of_BetterHelp_eCounseling.pdf

Mazzo, L. (n.d.). The Best Therapy and Mental Health Apps If the Couch-Session Thing Isn't for You. *Shape*. Retrieved December 17, 2019, from https://www.shape.com/lifestyle/mind-and-body/best-therapy-mental-health-apps

Meditação guiada - Mindfulness #1 c/ paulo borges. (2014, August 20). Retrieved October 10, 2020, from https://www.youtube.com/watch?v=ikNXhRpnQ8w

Meditt from Mindfulness Institute. (n.d.). Retrieved October 10, 2020, from https://meditt.space/about

Miles, L. D. (1961). *Techniques of Value: Analysis and Engineering*. McGraw-Hill.

MindShift ADAA review. (n.d.). Retrieved December 2, 2019, from https://adaa.org/node/2558

Mohr, D. C., Tomasino, K. N., Lattie, E. G., Palac, H. L., Kwasny, M. J., Weingardt, K., Karr, C. J., Kaiser, S. M., Rossom, R. C., Bardsley, L. R., Caccamo, L., Stiles-Shields, C., & Schueller, S. M. (2017). Intellicare: An eclectic, skills-based app suite for the treatment of depression and anxiety. *Journal of Medical Internet Research*, *19*(1). https://doi.org/10.2196/jmir.6645

Moody, J. (2019, September 5). *What does RMSE really mean?* Retrieved October 1, 2020, from https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e

MovieLens. (2013, September 6). Retrieved October 13, 2020, from https://grouplens.org/datasets/movielens/

Nielsen, J. (2012). Mobile Sites vs. Apps: The Coming Strategy Shift. *Nielsen Norman Group*. Retrieved February 22, 2020, from https://www.nngroup.com/articles/mobile-sites-vs-apps-strategy-shift/

NIMH Anxiety Disorders. (n.d.). Retrieved December 16, 2019, from https://www.nimh.nih.gov/health/topics/anxiety-disorders/index.shtml

NIMH Any Anxiety Disorder. (n.d.). Retrieved December 16, 2019, from https://www.nimh.nih.gov/health/statistics/any-anxiety-disorder.shtml

NIMH Obsessive-Compulsive Disorder. (n.d.). Retrieved December 16, 2019, from https://www.nimh.nih.gov/health/topics/obsessive-compulsive-disorder-ocd/index.shtml

NIMH Post-Traumatic Stress Disorder. (n.d.). Retrieved December 16, 2019, from https://www.nimh.nih.gov/health/topics/post-traumatic-stress-disorder-ptsd/index.shtml

NIMH Schizophrenia. (n.d.). Retrieved December 16, 2019, from https://www.nimh.nih.gov/health/topics/schizophrenia/index.shtml

OECD. (2018). Health at a Glance: Europe. *State of Health in the EU*. Retrieved February 15, 2020, from https://ec.europa.eu/health/state/glance_en

Osterwalder, A. (2004). The business model ontology a proposition in a design science approach.

Osterwalder, A., Pigneur, Y., Clark, T., & Smith, A. (2010). *Business model generation: A handbook for visionaries, game changers, and challengers*.

Pandey, P. (2019, March 19). *Understanding the Mathematics behind Gradient Descent.* Retrieved October 1, 2020, from https://towardsdatascience.com/understanding-the-mathematics-behind-gradient-descent-dde5dc9be06e

Paz, J. F. D., Julián, V., Villarrubia, G., Marreiros, G., & Novais, P. (2017, June). *Ambient Intelligence Software and Applications 8th International Symposium on Ambient Intelligence (ISAmI 2017)*. Springer.

Perez, S. (2020, June 18). *TikTok explains how the recommendation system behind its For You feed works*. Retrieved September 26, 2020, from https://social.techcrunch.com/2020/06/18/tiktok-explains-how-the-recommendation-system-behind-its-for-you-feed-works/

Pimenta, T. (2019, December). *10 dicas para controlar a ansiedade*. Retrieved October 10, 2020, from https://www.vittude.com/blog/controlar-a-ansiedade/

Resnick, P., & Varian, H. R. (1997). Recommender systems. *CACM*. https://doi.org/10.1145/245108.245121

Ritchie, H., & Roser, M. (2018). Mental Health. *Our World in Data*. Retrieved December 16, 2019, from https://ourworldindata.org/mental-health

Saaty, T. L. (1980). *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill.

Sagar, P. (2019, July 26). *Firebase vs. MongoDB: Which Database to Use for Your App Development - DZone Database*. Retrieved September 24, 2020, from https://dzone.com/articles/firebase-vs-mongodb-which-database-to-use-for-your

The Science-Backed Benefits of Mindfulness. (n.d.). *Headspace*. Retrieved December 17, 2019, from https://www.headspace.com/mindfulness

Shokeen, J., & Rana, C. (2019). A study on features of social recommender systems. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-019-09684-w

Spijkerman, M. P., Pots, W. T., & Bohlmeijer, E. T. (2016, April). *Effectiveness of online mindfulness-based interventions in improving mental health: A review and meta-analysis of randomised controlled trials* (Vol. 45). Elsevier Inc. https://doi.org/10.1016/j.cpr.2016.03.009

Talkspace Website. (2018). Retrieved December 2, 2019, from https://help.talkspace.com/hc/en-us

Tarianyk, D. (2020, October). *Clans/FloatingActionButton*. Retrieved October 4, 2020, from https://github.com/Clans/FloatingActionButton

Vicente Simón | Nascente. (n.d.). Retrieved October 10, 2020, from https://nascente.pt/autores/vicente-simon

Webb, C. A., Rosso, I. M., & Rauch, S. L. (2017). *Internet-based cognitive-behavioral therapy for depression: Current progress and future directions* (Vol. 25). Lippincott Williams and Wilkins. https://doi.org/10.1097/HRP.0000000000000139

What Are Anxiety Disorders? (n.d.). Retrieved December 16, 2019, from https://www.psychiatry.org/patients-families/anxiety-disorders/what-are-anxiety-disorders

What Is CBT? (n.d.). *Psychology Today*. Retrieved December 17, 2019, from https://www.psychologytoday.com/blog/bottoms/201611/what-is-cbt

What Is Depression? (n.d.). Retrieved December 16, 2019, from https://www.psychiatry.org/patients-families/depression/what-is-depression

What is Ionic Framework? (2019). Retrieved December 4, 2019, from https://ionicframework.com/docs/intro

What is Obsessive Compulsive Disorder? (n.d.). Retrieved December 16, 2019, from https://www.psychiatry.org/patients-families/ocd/what-is-obsessive-compulsive-disorder

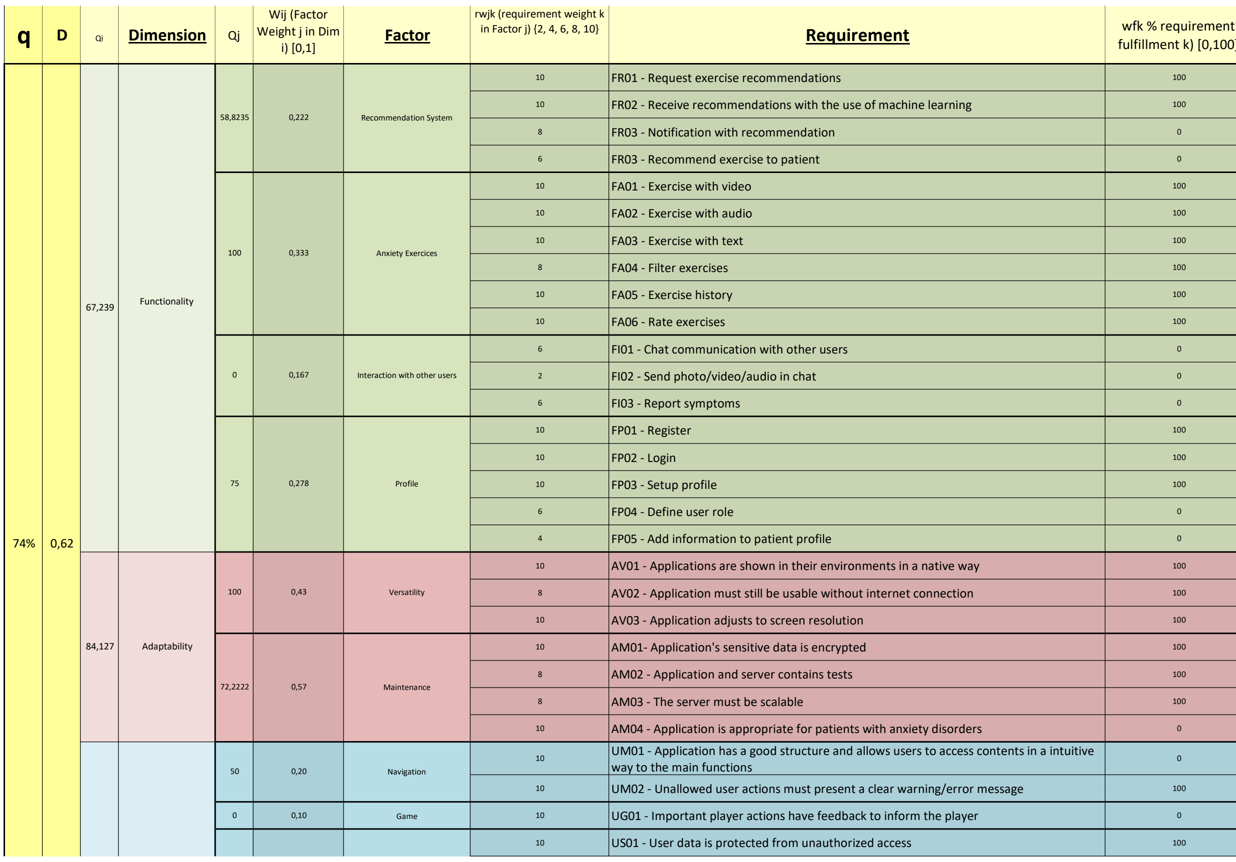
What Is PTSD? (n.d.). Retrieved December 16, 2019, from https://www.psychiatry.org/patients-families/ptsd/what-is-ptsd

Yuan, S., Ma, W., Kanthawala, S., & Peng, W. (2015). Keep Using My Health Apps: Discover Users' Perception of Health and Fitness Apps with the UTAUT2 Model. *Telemedicine and e-Health*, *21*(9), 735–741. https://doi.org/10.1089/tmj.2014.0148

Zeithaml, V. (1988). Consumer Perceptions of Price, Quality and Value: A Means-End Model and Synthesis of Evidence. *Journal of Marketing*, *52*, 2–22. https://doi.org/10.1177/002224298805200302

Appendix A

QEF



q	D	Qi	Dimension	Qj	Wij (Factor Weight j in Dim i) [0,1]	Factor	rwjk (requirement weight k in Factor j) {2, 4, 6, 8, 10}	Requirement	wfk % requirement fulfillment k) [0,100
74%	0,62	67,239	Functionality	58,8235	0,222	Recommendation System	10	FR01 - Request exercise recommendations	100
							10	FR02 - Receive recommendations with the use of machine learning	100
							8	FR03 - Notification with recommendation	0
							6	FR03 - Recommend exercise to patient	0
				100	0,333	Anxiety Exercices	10	FA01 - Exercise with video	100
							10	FA02 - Exercise with audio	100
							10	FA03 - Exercise with text	100
							8	FA04 - Filter exercises	100
							10	FA05 - Exercise history	100
							10	FA06 - Rate exercises	100
				0	0,167	Interaction with other users	6	FI01 - Chat communication with other users	0
							2	FI02 - Send photo/video/audio in chat	0
							6	FI03 - Report symptoms	0
				75	0,278	Profile	10	FP01 - Register	100
							10	FP02 - Login	100
							10	FP03 - Setup profile	100
							6	FP04 - Define user role	0
							4	FP05 - Add information to patient profile	0
		84,127	Adaptability	100	0,43	Versatility	10	AV01 - Applications are shown in their environments in a native way	100
							8	AV02 - Application must still be usable without internet connection	100
							10	AV03 - Application adjusts to screen resolution	100
				72,2222	0,57	Maintenance	10	AM01- Application's sensitive data is encrypted	100
							8	AM02 - Application and server contains tests	100
							8	AM03 - The server must be scalable	100
							10	AM04 - Application is appropriate for patients with anxiety disorders	0
				50	0,20	Navigation	10	UM01 - Application has a good structure and allows users to access contents in a intuitive way to the main functions	0
							10	UM02 - Unallowed user actions must present a clear warning/error message	100
				0	0,10	Game	10	UG01 - Important player actions have feedback to inform the player	0
							10	US01 - User data is protected from unauthorized access	100

		50	Usability	100	0,30	Support	10	US02 - The application is stable and executes systematically without failures	100	
							8	US03 - The application can be implemented in multiple platforms	100	
				25	0,40	Content Quality	10	UCQ01 - Texts are well written and all the sentences make sense	0	*Q
							10	UCQ02 - All the messages are easy to understand	0	*Q
							10	UCQ03 - The application content should not contain racist, xenophobic or anti-semitic content, or any other discriminatory content	0	*Q
							10	UCQ04 - All content is related to mental health	100	

*Q	* = questionnaire related requirement