



Comunicações Transacionais para Melhorar o Relacionamento com o Cliente

JOSÉ BERNARDO SOUSA MAIA ROCHA FERREIRA

Junho de 2022

Transactional Communications to Improve Customer Relationship

José Bernardo Sousa Maia Rocha Ferreira

**A dissertation submitted in partial fulfillment of
the requirements for the degree of Master of Science,
Specialization Area of Software Engineering**

Internal Supervisor: Isabel de Fátima Silva Azevedo

Enterprise Supervisor: Ivo André Soares Pereira

Abstract

Transactional communications are messages that are sent as soon as an action is completed by a user typically using a Web application. Therefore, it consists of a relationship strategy whose objective is to strengthen the ties between the organization and its customers, as well as provide relevant information.

Usually, transactional communications work automatically, and messages can be scheduled to be sent whenever an action is completed. However, small and medium organizations may not have the necessary funds and means to create and maintain a Web application or to integrate it with a transactional communication channel.

In this dissertation, a transactional communication tool called QuickMessage is analyzed and improved, which sends Short Message Service (SMS) transactional messages in a non-automated way, with users having to manually trigger their sending at an exact moment. Since the QuickMessage tool only had one SMS channel, it was intended to implement other transactional channels, such as the Email channel, and also allow the scheduling of SMS messages and Emails.

Evaluation methods based on software tests, usability questionnaires, and the Quantitative Evaluation Framework (QEF) are used to evaluate the final solution.

Keywords: Transactional Communications, Customer Relationship, SMS, Email.

Resumo

As comunicações transacionais são mensagens que são enviadas logo que uma ação é concluída por um utilizador, normalmente numa aplicação de Web. Portanto, consiste numa estratégia de relacionamento, cujo objetivo é melhorar a relação da empresa com os seus clientes, bem como fornecer informações relevantes.

Normalmente, as comunicações transacionais funcionam automaticamente e as mensagens podem ser agendadas para serem enviadas sempre que uma ação for concluída. No entanto, pequenas e médias empresas podem não ter os fundos e os meios necessários para criar e manter uma aplicação de Web ou para integrá-la com um canal de comunicação transacional.

Nesta dissertação, é analisada e aprimorada uma ferramenta de comunicação transacional denominada QuickMessage, que envia mensagens de Short Message Service (SMS) transacionais de forma não automatizada, devendo o utilizador acionar manualmente o seu envio no momento exato. Uma vez que a ferramenta QuickMessage possuía apenas um canal de SMS, pretendeu-se implementar outros canais transacionais, como o canal de Email, e também permitir o agendamento de mensagens de SMS e Emails.

Métodos de avaliação baseados em testes de software, questionários de usabilidade e na Quantitative Evaluation Framework (QEF) são utilizados para avaliar a solução final.

Palavras-chave: Comunicações Transacionais, Relacionamento com Cliente, SMS, Email.

Acknowledgment

The internship and the elaboration of the report would not be possible without the support of some stakeholders. Therefore, I want to thank all those who supported me and contributed to the realization and achievement of this final stage in my education, the degree of Master of Science, specialization area of Software Engineering.

Thus, I thank:

My family, because all this was possible thanks to the effort and dedication they always had, and thanks to the motivation they gave me whenever I was unmotivated.

Isabel Azevedo, for accepting to be my advisor in this project, for her availability and understanding, helping in the most difficult times, always manifesting her enriching insights for the internship, the completion of the report, and enrichment of my training.

Ivo Pereira, the internship's coordinator, for being my supervisor at E-goi, Lda., for making available part of his time and helping in the most difficult times, always expressing his enriching opinions for the internship, the elaboration of the report, and enrichment of my training.

All the teachers who contributed to my training throughout my degree, for all the knowledge, dedication, and help for my personal and educational growth.

All the employees of the company where the internship took place, for all their attention, sympathy, understanding, and knowledge imparted during my stay.

Thank you all very much!

Table of Contents

1	Introduction	1
1.1	Context	1
1.2	Organization Presentation	2
1.3	Problem.....	2
1.4	Objectives.....	3
1.5	Methodology	4
1.6	Document Structure	6
2	Background	7
2.1	Context	7
2.2	E-Commerce	7
2.3	Marketing.....	8
2.3.1	Digital Marketing	9
2.3.2	SMS Marketing.....	11
2.3.3	Email Marketing	12
3	State of Art	15
3.1	Transactional Communications.....	15
3.1.1	Marketing Communications vs Transactional Communications	16
3.2	Tools	17
3.2.1	QuickMessage	17
3.2.2	MailChimp	17
3.2.3	MailJet.....	17
3.2.4	Sendinblue	18
3.2.5	SMSAPI	18
3.2.6	uSendit.....	18
3.2.7	Comparative Analysis	18
4	Value Analysis.....	21
4.1	Innovation Process	21
4.2	New Concept Development	22
4.2.1	Opportunity Identification	23
4.2.2	Opportunity Analysis.....	23
4.2.3	Ideas Generation	27
4.2.4	Ideas Selection.....	28
4.2.5	Concept Definition	34
4.3	Solution Value	34
4.3.1	Value, Customer Value, and Perceived Value	34
4.3.2	Value Proposition and Canvas Business Model.....	35

5	Analysis.....	37
5.1	Domain Model.....	37
5.2	Functional Requirements	39
5.3	Non-Functional Requirements.....	40
5.3.1	Quality Attributes	40
5.3.2	Constraints	41
5.3.3	Concerns.....	41
5.4	Use Cases.....	41
5.4.1	Send Quick Email	42
5.4.2	Send Advanced Email	43
5.4.3	Create Email Template.....	44
5.4.4	View Email Templates	45
5.4.5	Delete Email Template.....	45
5.4.6	Edit Email Template	46
5.4.7	View Email Reports	47
5.4.8	View Email Senders	47
5.4.9	Schedule SMS Message	48
5.4.10	Schedule Email	48
5.4.11	View Scheduled SMS Messages.....	49
5.4.12	View Scheduled Emails	49
5.4.13	Delete Scheduled SMS Message.....	50
5.4.14	Edit Scheduled SMS Message	51
5.4.15	Delete Scheduled Email	51
5.4.16	Edit Scheduled Email	52
6	Design.....	53
6.1	Technology Stack.....	53
6.2	Overall Software Architecture	53
6.3	Alternative Overall Software Architecture	54
6.4	Attribute-Driven Design	55
6.4.1	Iteration 1	57
6.4.2	Iteration 2.....	65
7	Implementation	71
7.1	Development Methodology.....	71
7.2	Solution Development	71
7.2.1	Email Senders.....	72
7.2.2	Email Templates	72
7.2.3	Emails Sending.....	75
7.2.4	Email Reports	80
7.2.5	Scheduling.....	82
7.2.6	Permissions.....	82
7.2.7	Documentation	84
7.2.8	Response Time.....	84
7.2.9	Notifications	85

8	Evaluation	87
8.1	Evaluation Indicators	87
8.2	Definition of Hypotheses	87
8.3	Evaluation Methodology	88
8.3.1	Software Testing	88
8.3.2	QEF Model	89
8.3.3	Usability Questionnaire.....	92
8.4	Results	93
8.4.1	Software Testing	93
8.4.2	QEF Model	97
8.4.3	Usability Questionnaire.....	98
9	Conclusions.....	101
9.1	Achieved Goals and Results.....	101
9.2	Limitations and Future Work	102

List of Figures

Figure 1 - DSRM Process Model according to Peffers [1]	5
Figure 2 - E-Commerce Business Models [3].....	8
Figure 3 - Evolution of Digital Marketing [6].....	10
Figure 4 - Differences between Digital Marketing and Classical Marketing [6].....	10
Figure 5 - Example of SMS Marketing	12
Figure 6 - Example of Email Marketing	13
Figure 7 - Open Rate and Unique Clicks in Standard and Transactional Emails [14]	16
Figure 8 - Innovation Process Steps [25].....	22
Figure 9 – NCD Model [25].....	23
Figure 10 - SWOT Analysis.....	24
Figure 11 - Interest Over Time of "Transactional Email" and "Transactional SMS" between 2004 and 2022.....	26
Figure 12 - Interest Over Time of MailJet between 2004 and 2022	26
Figure 13 - Interest Over Time of Sendinblue between 2004 and 2022	26
Figure 14 - Hierarchical Structure of AHP	29
Figure 15 - Canvas Business Model.....	35
Figure 16 - Domain Model	37
Figure 17 - Use Cases Diagram.....	42
Figure 18 - Send Quick Email SSD.....	43
Figure 19 - Send Advanced Email SSD	44
Figure 20 - Create Email Template SSD.....	45
Figure 21 - View Email Templates SSD	45
Figure 22 - Delete Email Template SSD.....	46
Figure 23 - Edit Email Template SSD	46
Figure 24 - View Email Reports SSD	47
Figure 25 - View Email Senders SSD	47
Figure 26 - Schedule SMS Message SSD.....	48
Figure 27 - Schedule Email SSD	49
Figure 28 - View Scheduled SMS Messages SSD	49
Figure 29 - View Scheduled Emails SSD	50
Figure 30 - Delete Scheduled SMS Message SSD	50
Figure 31 - Edit Scheduled SMS Message SSD	51
Figure 32 - Delete Scheduled Email SSD	52
Figure 33 - Edit Scheduled Email SSD.....	52
Figure 34 - Components Diagram of the Current Overall Architecture	54
Figure 35 - Components Diagram of the Alternative Overall Architecture	55
Figure 36 - Steps of the Attribute Driven Design method (adapted) [31]	56
Figure 37 – Components Diagram of the Primary Use Cases on Layers	61
Figure 38 - Send Quick Email SD	62
Figure 39 - Send Advanced Email SD.....	63

Figure 40 - Create Email Template SD	63
Figure 41 - View Email Templates SD	63
Figure 42 - Delete Email Template SD	64
Figure 43 - Edit Email Template SD	64
Figure 44 - View Email Senders SD	64
Figure 45 - Components Diagram of the New Elements' Dependencies	68
Figure 46 - User Interface of the Feature of Visualizing Email Senders	72
Figure 47 - User Interface of the Feature of Visualizing Email Templates	73
Figure 48 - User Interface of the Feature of Creating Email Templates	74
Figure 49 - User Interface of the Feature of Deleting Email Templates.....	74
Figure 50 - User Interface of the Feature of Editing Email Templates.....	75
Figure 51 - User Interface of the First Step of the Feature of Sending Quick Email	76
Figure 52 - User Interface of the Second Step of the Feature of Sending Quick Email.....	76
Figure 53 - Email Section while Sending Quick Email.....	77
Figure 54 - User Interface of the Third Step of the Feature of Sending Quick Email	77
Figure 55 - Schedule Section while Sending Quick Email	78
Figure 56 - User Interface of the First Step of the Feature of Sending Advanced Email.....	78
Figure 57 - User Interface of the Second Step of the Feature of Sending Advanced Email	79
Figure 58 - Email Section while Sending Advanced Email	79
Figure 59 - User Interface of the Third Step of the Feature of Sending Advanced Email	80
Figure 60 - Schedule Section while Sending Advanced Email	80
Figure 61 - User Interface of the Feature of Visualizing Email Reports (graph).....	81
Figure 62 - User Interface of the Feature of Visualizing Email Reports (table).....	81
Figure 63 - Custom Annotation of the Method to Send a Quick Email.....	82
Figure 64 – Authorization Class	82
Figure 65 - Controller's Authorization Handler Method	83
Figure 66 - User Interface of the Feature of Updating Role.....	83
Figure 67 - Swagger Documentation Example	84
Figure 68 - Code that Allows Sending Advanced Emails Asynchronously	84
Figure 69 - Spinner when Loading the Email Templates	85
Figure 70 - Notifications when Changing User Status	85
Figure 71 - V-Model Architecture [33]	88
Figure 72 - Three-dimensional quality space [34].....	89
Figure 73 - Dimensions, Factors, Requirements, and respective Weights of the QEF Model ...	90
Figure 74 - Email Channel Factor Requirements	90
Figure 75 - Schedule SMS Messages and Emails Factor Requirements	91
Figure 76 - Maintenance Factor Requirements.....	91
Figure 77 - Performance and Usability Factors Requirements	91
Figure 78 – Results of the Fulfillment of Requirements and Overall System Quality, using the QEF Model	98

List of Tables

Table 1 - Differences between Marketing and Transactional Communications [15]	16
Table 2 - Comparative Analysis Between Platforms	19
Table 3 - Saaty Scale.....	30
Table 4 – Second Level Criteria Comparison Matrix	30
Table 5 - Second Level Criteria Normalized Matrix.....	30
Table 6 - Random Index Table.....	31
Table 7 - Parity Comparison Matrix of Development Time	32
Table 8 - Parity Comparison Matrix of Complexity	32
Table 9 - Parity Comparison Matrix of Technological Resources.....	32
Table 10 - Parity Comparison Normalized Matrix of Development Time.....	33
Table 11 - Parity Comparison Normalized Matrix of Complexity.....	33
Table 12 - Parity Comparison Normalized Matrix of Technological Resources	33
Table 13 - Solutions Relative Priorities Matrix for each Criterion	33
Table 14 - Criteria Relative Priorities Matrix.....	33
Table 15 - Composite Priority Matrix.....	34
Table 16 - Elements and Responsibilities of the Presentation Layer	58
Table 17 - Elements and Responsibilities of the Business Layer.....	59
Table 18 - Methods of the EmailTemplatesService	59
Table 19 - Methods of the SendEmailService	59
Table 20 - Methods of the SendersService	59
Table 21 - Methods of the EmailTemplatesController.....	60
Table 22 - Methods of the SendEmailController.....	60
Table 23 - Methods of the SendersController	60
Table 24 - Methods of the EmailTemplatesDispatcher.....	60
Table 25 - Methods of the EmailDispatcher	61
Table 26 - Methods of the SendersDispatcher	61
Table 27 - Design Kanban Board - Iteration 1	65
Table 28 - New Elements and Responsibilities of the Presentation Layer.....	67
Table 29 - New Elements and Responsibilities of the Business Layer	67
Table 30 - Design Kanban Board - Iteration 2	69
Table 31 - Association between evaluation indicators and evaluation methods.....	88
Table 32 - Items/Questions of the SUS Questionnaire [32].....	92
Table 33 - Scale and Description of the SUS Questionnaire Answers [32]	92
Table 34 – Interpretation of SUS Score [35]	93
Table 35 - Criteria and Results of the UC08 called View Email Senders	95
Table 36 - Criteria and Results of the UC03 called Create Email Template	95
Table 37 - Criteria and Results of the UC04 called View Email Templates	96
Table 38 - Criteria and Results of the UC05 called Delete Email Template	96
Table 39 - Criteria and Results of the UC06 called Edit Email Template.....	96
Table 40 - Criteria and Results of the UC01 called Send Quick Email	96

Table 41 - Criteria and Results of the UC02 called Send Advanced Email	97
Table 42 - Criteria and Results of the UC07 called View Email Reports.....	97
Table 43 - Answers of the Users to the SUS Questionnaire	99
Table 44 - Average of the Answers of the Users and Normalized Results to the SUS Questionnaire.....	100

Acronyms

ADD	Attribute-Driven Design
AHP	Analytic Hierarchy Process
AMA	American Marketing Association
API	Application Programming Interface
B2B	Business-to-Business
B2C	Business-to-Consumer
CI	Consistency Index
CR	Consistency Ratio
CRUD	Create, Read, Update and Delete
DBMS	Database Management System
DSRM	Design Science Research Methodology
FR	Functional Requirement
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IS	Ideal System
JSON	JavaScript Object Notation
NCD	New Concept Development
QEF	Quantitative Evaluation Framework
REST	Representational State Transfer
RI	Random Index
RS	Real System
SD	Sequence Diagram
SMS	Short Message Service
SSD	System Sequence Diagram

SUS	System Usability Scale
UML	Unified Modeling Language
URL	Uniform Resource Locator

1 Introduction

This chapter is dedicated to describing not only the current context of this work but also the problem this dissertation is based on and its objectives. Furthermore, it also defines the methodology used and the summary of the structure of the document with an overview of all its chapters.

1.1 Context

Nowadays, organizations need a way to send messages to their customers, either related to transactions or just marketing. These messages are called transactional communications, and they can be sent as soon as an action is completed by a user typically using a Web application. Therefore, it consists of a relationship strategy whose objective is to strengthen the ties between the organization and its customers, as well as provide relevant information.

Usually, transactional communications work automatically, and messages can be scheduled to be sent whenever an action is completed. However, small and medium organizations may not have the necessary funds and means to create and maintain a Web application or to integrate it with a transactional communication channel.

When this is the case, organizations deal with their business more commonly, by receiving and storing the customer and transaction data in Excel. With this data, they send messages to customers manually, which requires a lot of time and it isn't practical. By using a transactional communication tool, small and medium organizations can import their Excel with data, and send messages to a group of customers with little effort and in a more practical way.

The platform QuickMessage is an example of a transactional communication tool, developed by E-goi, to respond to the needs of small and medium organizations, and to be simple and easy to use. It allows the sending of SMS messages to a group of customers, the creation and visualization of SMS messages templates, and the visualization of reports/statistics related to the sending of these messages.

Bearing in mind the opportunity presented to the organization by the current utility of the transactional communication tools, it is important to focus on developing new features to add to the current platform QuickMessage. This must be done to be able to differentiate itself from or get closer to, other organizations and platforms that operate in the same sector of activity.

1.2 Organization Presentation

E-goi is a digital marketing organization, founded by Miguel Gonçalves, in 2008, based in Matosinhos, Portugal. At the time of writing this document, the company has more than 100 employees spread across several countries, between Europe and Latin America. Currently, the company has a strong presence in European and Latin American markets, with emphasis on the Portuguese and Spanish markets in Europe, and Brazilian and Colombian in Latin America.

E-goi's mission is to create effective digital communication solutions, with intuitive usability. Regarding its vision, E-goi intends to be the partner that manages to obtain from consumers the recognition of the value of the brands. They defend values such as innovation, creativity, dynamism, relevance, and sustainability.

E-goi provides an automated multi-channel marketing tool called E-goi, that offers several channels, such as email, SMS, push notifications, and publications on social networks such as Facebook and Twitter. To complement the sending of messages through these channels, E-goi allows editing the body of messages to send, managing recipients, defining behavior flows according to the status of messages, among other features. The main objective of the platform is to help its customers to achieve their professional goals, facilitating interaction between them and their contacts.

In addition, E-goi provides a recent new platform called QuickMessage, which is the platform analyzed and improved in this document, that allows the sending of transactional messages in the SMS channel.

1.3 Problem

In this dissertation, although there are many transactional communications tools, it will be analyzed and improved one called QuickMessage, a solution provided by E-goi.

This platform allows the sending of SMS transactional messages in a non-automated way, with users having to manually trigger their sending at an exact moment. When sending SMS transactional messages, it is possible to import an Excel with custom data of the customers, including the recipient phone, that can be used as the list of recipients of the SMS message. In addition, the platform allows the users to create and visualize SMS messages templates, so that users may select a template when sending an SMS transactional message. Finally, it is possible to visualize reports/statistics related to the sending of these messages, so that users can detect if their messages are being sent, delivered, or rejected.

Although the platform is functional and has customers who use it, it has some limitations and problems:

- **Only One Transactional Channel** – The platform only has one transactional channel, the SMS transactional channel, which limits the usage of the application. This is a

problem because some customers may not want to receive messages by SMS but they may prefer to receive them in another channel, such as the Email channel.

- **Not Possible Scheduling SMS Messages** – The platform doesn't allow the scheduling of SMS messages and users have to manually trigger their sending at an exact moment. This is a problem because the user must be present at the exact moment they want the SMS message to be sent, and if the user is busy at that moment, it won't be possible to send the message.
- **Competitors Have More Features** – The competitor platforms that operate in the same sector of activity have more features than QuickMessage, as can be seen in Table 2. This is a problem because the platform will be considered inferior and incomplete in comparison with the other competitor platforms, and may lose new and old customers that want to take advantage of a better platform.
- **Customers Are Not Fully Satisfied** – The platform has been used by several customers, but they have not been fully satisfied, mainly because of the lack of features, so they have made several improvements proposals. These improvements proposals include the implementation of the new Email channel and feature to allow the scheduling of SMS messages and Emails. This is a problem because if the customers are not fully satisfied, they may stop using the platform and start using one of the competitor platforms.
- **Usability Problems** – The platform has some usability problems since some buttons are not working, some areas have poor visibility and there is inconsistency regarding the design of some areas. This is a problem because the interface won't be intuitive and the customer may have a bad experience using the platform. This may cause the customer to stop using the platform and start using one of the competitor platforms.
- **Performance Problems** – The platform has some performance problems when importing Excel files with too many rows of data and then using it to send SMS messages. In addition, when visualizing the list of SMS messages templates, the full list is loaded, which may cause a delay showing it when there are too many templates. This is a problem because it will cause customer dissatisfaction when using the application.

1.4 Objectives

The main objective is to analyze and improve the transactional communication tool called QuickMessage, by adding new features and making some improvements related to usability and performance, a request made by E-goi.

Since the QuickMessage tool has only one SMS channel, the implementation of other transactional channels is intended, such as the Email channel. The implementation of this channel includes the feature to allow sending transactional Emails where it is possible to import an Excel with custom data of the customers, including the recipient email, that can be used as the list of recipients of the Email. In addition, the feature to allow users to create and visualize Email templates is also included, so that users may select a template when sending a

transactional email. Finally, the implementation of this channel includes the feature to visualize reports/statistics related to the sending of these emails, so that users can detect if their emails are being sent, delivered, clicked, opened, or rejected.

Since the tool only sends SMS messages manually, and users have to manually trigger their sending at an exact moment, the implementation of a feature to allow scheduling SMS messages and Emails is intended.

To solve the problems of the competitors having more features and customers not being fully satisfied, the implementation of the new Email channel and the feature to allow the schedule of SMS messages and Emails are intended, to enrich the platform.

To solve the usability problems and performance problems, an analysis of the platform QuickMessage is intended, to find the majority of these problems and fix them the correct and best way.

These objectives will be accomplished by following these steps:

1. Analysis of the concepts of marketing and transactional communication;
2. Study and analysis of the state of the art on the subject;
3. Analysis of the current software solution (in aspects such as features, usability, and performance);
4. Consultation of the architectural documentation (new architectural drivers may be identified);
5. Evaluation of the different approaches;
6. Design of the solution;
7. Implementation of the solution;
8. Evaluation of the designed/implemented solution.

1.5 Methodology

This dissertation uses an adapted and customized version of the Design Science Research Methodology (DSRM), created by Ken Peffers. It covers the traditional phases of Software Engineering and is compatible with software development activities.

The methodology consists of six steps, which were defined by Peffers [1]:

1. **Problem Identification and Motivation** – It is the definition of the problem and the justification of how the proposed solution will be valuable in this context. For the definition of the problem, it is necessary sufficient knowledge of the problem. This is represented by the study of the current state of the art and analysis of related work. In this document, the problem is described in section 1.3, and the value of the proposed solution is in sections 2, 3, and 4.
2. **Objectives of Solution Definition** – It is the definition of the objectives based on the problem's constraints. These objectives can be quantitative or qualitative. Quantitative

objectives measure how the new solution is better than current solutions while qualitative objectives measure how the new solution solves problems that haven't been addressed. In this document, the objectives are defined in section 1.4.

3. **Design and Development** – It is the creation of design artifacts and implementation of the desired solution to the problem. The design is performed after studying the current state of the art and analyzing related work and the current application. Subsequently, the new solution will be developed, according to the design. In this document, the analysis and design are defined in sections 5 and 6, while the development is defined in section 7.
4. **Demonstration** – It is the description of how effectively the solution solves the problem. It is necessary to demonstrate the ability of the solution to solve one or more of the issues defined in the problem identification. In this document, the demonstration is defined in section 8.
5. **Evaluation** – It is the measurement of how the implementation fulfills the defined objectives for the problem. In this document, the evaluation is defined in section 8.
6. **Communication** – It is the final step, that communicates the solution and shows all the achieved outcomes and results. The dissertation itself communicates the solution and the achieved outcomes and results are defined in the final section 9.

In a more general view, the six steps of the DSRM can be observed in Figure 1.

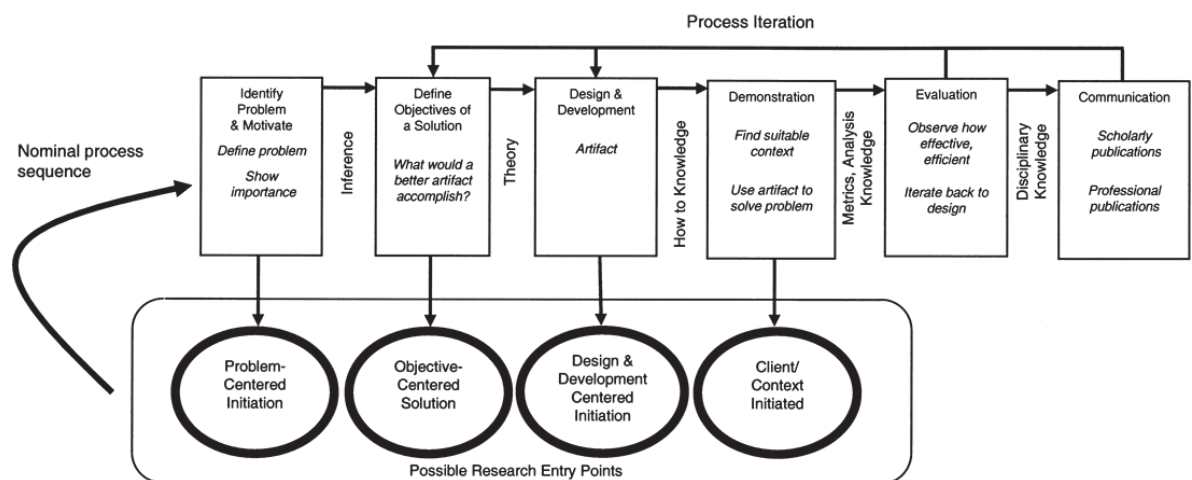


Figure 1 - DSRM Process Model according to Peffers [1]

The difference between the adapted and customized version of DSRM, and the original DSRM is that this methodology used doesn't require knowledge generation for a class of problems. This means that it isn't required, for a specific situation, to present a solution in a generic way that can be used in other situations.

1.6 Document Structure

The structure of this document consists of 9 chapters: Introduction, Background, State of the Art, Value Analysis, Analysis, Design, Implementation, Evaluation, and finally, Conclusions.

The present chapter, Chapter 1, denominated Introduction, describes the current context of this work, the problem this dissertation is based on, and its objectives. Furthermore, it also defines the methodology used and the summary of the structure of the document with an overview of all its chapters.

Chapter 2, denominated Background, provides an analysis of the concepts related to E-Commerce and Marketing.

Chapter 3, denominated State of the Art, provides an overview of the study of the current state of transactional communications. Furthermore, it provides a summary of related work, where the applications that exist in the area under study, in which the project work fits, are identified.

Chapter 4, denominated Value Analysis, describes the value analysis of the work, starting with a section related to business process and innovation, where the New Concept Development (NCD) model is applied. In addition, it defines and explains the usage of the Analytic Hierarchy Process (AHP) for multicriteria decisions to aid solve the problem of this work. To conclude, it demonstrates the customer value and the value proposition with the inclusion of the Canvas business model.

Chapter 5, denominated Analysis, demonstrates the requirements engineering, that englobes the identification of the domain model, functional and non-functional requirements, and use cases.

Chapter 6, denominated Design, presents the design of the solution, which includes the technology stack, architecture alternatives, and detailed software architecture. In addition, it depicts the application of the method Attribute-Driven Design (ADD) for architectural design, with the use of many drivers previously identified and many design concepts.

Chapter 7, denominated Implementation, provides a detailed overview of the implementation. It demonstrates how the framing and implementation of the solution were done according to the analysis and design of the solution defined in Chapters 5 and 6.

Chapter 8, denominated Evaluation, demonstrates how the solution was evaluated using evaluation indicators, the hypotheses defined for them and the evaluation framework applied called Quantitative Evaluation Framework (QEF). In addition, the tests performed on the implemented solution, based on some quality attributes, are explained.

The final chapter, Chapter 9, denominated Conclusions, describes the achieved goals and results and identifies some limitations and future work.

2 Background

This chapter is dedicated to providing an analysis of the concepts related to E-Commerce and Marketing.

2.1 Context

At a time marked by technological advances and changes in consumer habits, where E-Commerce starts to win, digital marketing is a key piece of communication between companies and their customers. This communication, maintained through different approaches, such as contact by email, needs to be well structured, personalized, and always with a converter or reader focus on a buyer [2].

2.2 E-Commerce

E-Commerce has become a useful business model that is used by a huge number of organizations all over the world. In the year 1979, Michael Aldrich introduced and invented online shopping to provide the opportunity of enabling online transactions between businesses and consumers or between one business and another. This concept and technique were later called Electronic Commerce (E-Commerce) in which businesses took the use of the internet to conduct transactions [3].

So, E-Commerce is the act of buying and selling products and services through the Internet. The business model is designed in a way that provides its offerings online and allows consumers to select and purchase the desired products [3].

To function online, companies must select the most adequate technology to carry out the desired tasks and activities. In this regard, the considerations need to be made by the nature and type of customers along with the capabilities, resources, and kind of business that is involved in the process. The technology usage may differ for varied businesses and their specific target audiences and critical analysis in this regard could be conducted to ensure optimum results [3].

In Figure 2, it is possible to observe the many existing E-Commerce Business Models.

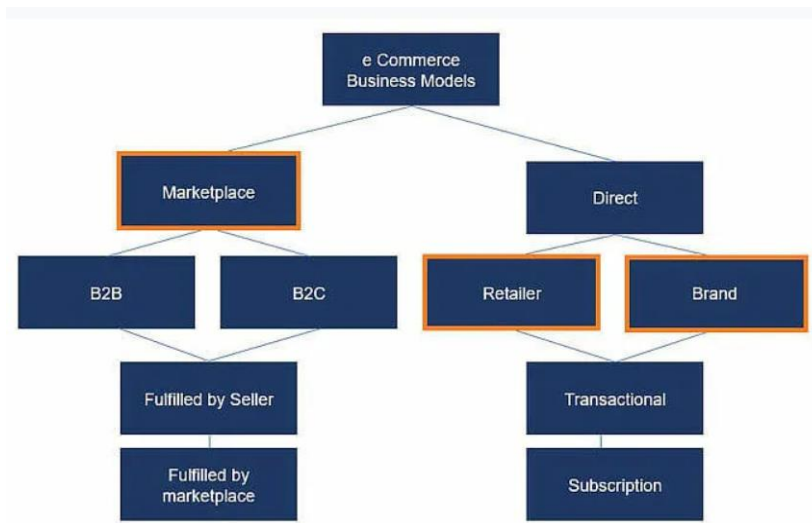


Figure 2 - E-Commerce Business Models [3]

E-Commerce emerges as one of the most imperative innovations in contemporary times environment. Businesses around the world running in many industries have inclined towards the usage of E-commerce to optimize the opportunities and improve their overall performance [3].

E-Commerce further has led to an increase in the efficiency of production and operations management. With an online presence, it has become more feasible for firms to take up orders from the customers, and get them produced and delivered effectively. Businesses working upon the E-Commerce models have been allowed to get the products delivered to customers within no time frame. Using E-Commerce, businesses have been allowed to enter new markets and lead toward internationalization. With the use of online platforms organizations have been allowed to reach out to customers all over the world [3].

In addition, E-Commerce has led to an increase in the efficiency of production and operations management. With an online presence, it has become more feasible for companies to take orders from customers, produce them and deliver them effectively. The companies working on the E-Commerce models were allowed to deliver the products to the customers in a short time. Using E-Commerce, companies were allowed to enter new markets and lead in internationalization. With the use of online platforms, organizations were empowered to reach customers across the world [3].

2.3 Marketing

For most people, the first thing that comes to mind when the subject of marketing comes up is advertising, in forms such as television ads, newspaper entries, or roadside billboards. Modern society has become so accustomed to these advertisements that it isn't difficult to find countless examples of them. However, there is more to the marketing field than simply finding new ways to attract consumer attention to certain products. At its core, marketing is simply a

way of managing relationships between suppliers of products and services and those who buy or consume those products [4].

The concept of marketing as it is understood was defined during the time of the Industrial Revolution. This period spanned the end of the 18th century and lasted until the 19th century. It was a time of rapid social change driven by innovations in the scientific and technological industries [4].

It was during the Industrial Revolution that buying goods became easier for the consumer than doing things on their own. Mass production created many industries engaged in the same effort to meet the needs of a growing consumer market. The transport infrastructure, as well as the mass media, took over, which created the need for producers to find better ways to develop the products customers needed and a more sophisticated approach to informing them about these commodities [4].

According to American Marketing Association (AMA), "Marketing is the activity, set of institutions, and processes for creating, communicating, delivering, and exchanging offerings that have value for customers, clients, partners, and society at large" [5]. It is an area focused on generating value for a product or service of a business, to win and retain customers.

There are many types of marketing, however, the most predominant one is digital marketing, which is applied to electronic media such as websites, blogs, social networks, and emails [5].

Some of the marketing objectives are as follows:

- Increase sales, since it's the main objective of an organization;
- Retain customers to make them come back and buy the product or service again;
- Increase visibility to attract the target audience;
- Build good relations to strengthen the brand ;
- Engage collaborators which are essential for the business's success.

2.3.1 Digital Marketing

With the coming of digital technology and the decline of some forms of traditional media, businesses have been forced to develop new means of connecting with the public and promoting products and services to potential customers.

Digital marketing is the marketing of products or services carried out through digital technology such as the Internet. The use of digital marketing strategies has allowed many businesses to diversify their marketing efforts, as traditional forms of marketing have become less effective.

As it can be observed in Figure 3, digital marketing has come a long way. The concept started in 1971 when the first email was sent by Tomlinson. It greatly expanded in 1998 when Google was launched, in 2004 when Facebook was started, and in 2005 when Youtube was created. It

continues to exist and expand nowadays, where there is programmatic advertising, in-app advertising, GeoFencing, and Remarketing.

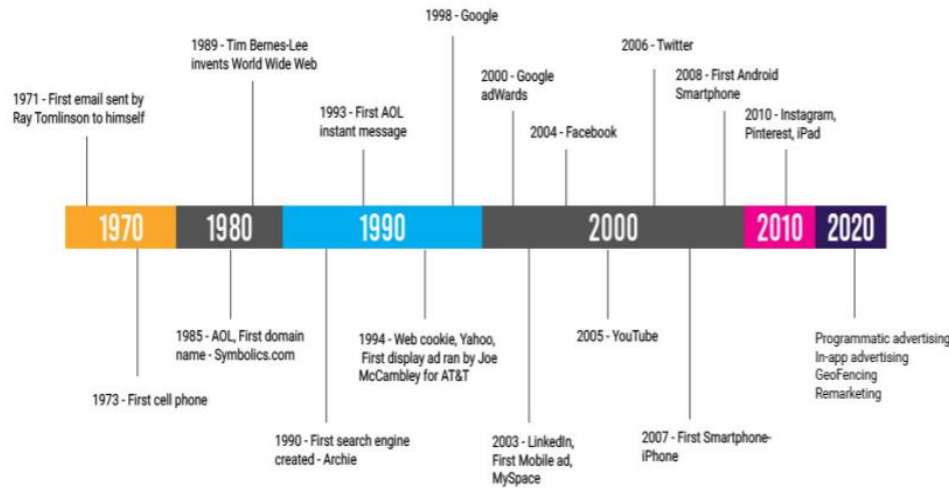


Figure 3 - Evolution of Digital Marketing [6]

In Figure 4, it is possible to identify the differences between digital marketing and classical marketing. In classical marketing there is limited interaction with the target group, the focus is on the product, the marketing is paid and there is no participation of the customer, while in digital marketing there is a continuous dialogue with the target group, the focus is on the consumer, the marketing is free and there is communication with the participating consumers. In addition, digital marketing is an open system and transparent while classical marketing is not. Finally, while classical marketing has mass and formal communication, no interaction, and medium-term control of results, digital marketing is one-to-one marketing, it has informal and social communication and it has a one-day test with a speed control of results.

	DIGITAL MARKETING	CLASSICAL MARKETING
	Computer, smartphones, tablets, game consoles	Press, the electronic device (TV, Radio), Direct Marketing, Phone
1	Continuous dialogue with the target group	Limited interaction
2	Focus on consumer	Focus on product
3	Free	Paid
4	Communicating with the participating consumers	No participation of the customer
5	Open system	No open system
6	Transparent	Not Transparent
7	One to one marketing	Mass communication
8	Informal communication	Formal communication
9	Social communication	No interaction
10	One-day test (speed control of results)	Medium-term control of results

Figure 4 - Differences between Digital Marketing and Classical Marketing [6]

There are many specializations within digital marketing such as the following [7]:

- **Search Engine Optimization** – It is the art and science of making web pages attractive to search engines;
- **Content Marketing** – It is a strategy based on the distribution of relevant and valuable content to a target audience;
- **Social Media Marketing** – This means driving traffic and brand awareness by engaging people in discussion online. The most popular platforms for it are Facebook, Twitter, and Instagram;
- **Pay-Per-Click Marketing** – It is posting an ad on a platform and paying every time a user clicks on it;
- **Affiliate Marketing** – It lets someone make money by promoting another person's business;
- **Native Advertising** – It is marketing in disguise. Its goal is to blend in with its surrounding content so that it's less blatantly obvious as advertising;
- **Marketing Automation** – It uses software to power digital marketing campaigns, improving the efficiency and relevance of advertising;
- **Email Marketing** – It is sending a promotional message by email and hoping that the user clicks on it (this is the most effective lead generator).

2.3.2 SMS Marketing

Over the years, the cell phone has become an increasingly attractive product with additional features. SMS marketing is an effort to effectively and successfully use this capacity for the introduction of goods and services [8].

SMS, known as text messaging, is a store-and-forward communication system for the cell phone, while SMS marketing, according to Dickinger, is defined as “using interactive wireless media to provide customers with time and location sensitive, personalized information that promotes goods, services and ideas, thereby generating value for all stakeholders” [8].

Text message ads were found to increase consumers' inclination to buy and this explains why their popularity is growing (more than 10 billion SMS messages each month). Organizations have effectively implemented SMS advertising to update their customers and track people and parcels [8].

As a media channel, mobile is unique, since it has a mass-market reach, it is a highly capable target and it is a tracking capable interactive media. In addition, SMS advertising is also commonly used to reinforce other traditional media such as broadcast and print media. Studies have found that SMS advertising campaigns generate higher response rates compared to email and internet advertising campaigns [8].

Because of special SMS marketing characteristics such as ubiquity, sensitivity to time, personalization, and interactivity, the attractiveness of SMS advertising is growing rapidly with the potential for a new communication channel [9].

SMS marketing messages are easy to create and execute and can be delivered to recipients in a few seconds. The downside is the limitation of 160 characters. Unlike Email ads, the delivery of SMS ads is guaranteed, since there are no spam filters that can catch advertising messages. Since cell phones are personal devices and people take them with them wherever they go, customers will get messages anytime and anywhere. Innovation and interactivity make messages more personalized and encourage potential customers to respond right away [9].

In Figure 5, it is possible to observe an example of an SMS Marketing message, where an organization wishes a person a happy birthday and gives a discount.

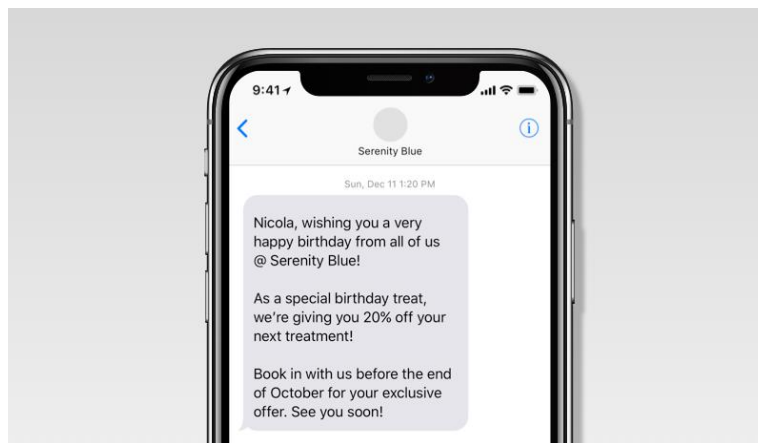


Figure 5 - Example of SMS Marketing

There are two types of messages that it is possible to send with SMS marketing [10]:

- **Campaigns** – They are the classic one-to-many bulk text messages, which are typically used to communicate promotions or general information;
- **Transactional** – They are the one-to-one messages that are triggered by a certain event or behavior like an order confirmation or shipping notification for an E-Commerce purchase. These messages communicate relevant and timely information to an individual.

2.3.3 Email Marketing

The first email was sent in 1971 by a computer engineer named Ray Tomlinson, which was just a string of numbers and letters. In 1978, a marketing manager at Digital Equipment Corp named Gary Thuerk used this new method of direct communication to send the first commercial email to inform people about a new product. In the 1990s, the internet became commercially available to the masses and the way people communicated began to change dramatically, which made marketers discover that email could be an effective way to advertise [11].

An email was one of the first widely used communication tools in the early stages of E-Commerce, and despite the emergence of several complementary communication channels, it continues to play an important role for many multi-channel retailers [12].

In addition, emails also have different characteristics compared to other digital channels. For example, when it comes to online advertising, email provides marketers with more control in terms of delivery time and the context in which the message is displayed [12].

Email marketing is a “communication tool where the sender uses an electronic mail as a communication tool to communicate messages to people or groups of customers”. The role of Email marketing includes the sending of organization announcements and any email communication intended to build loyalty, trust, or brand awareness [13]. It is a powerful marketing channel, that uses email to promote an organization’s products or services, by making customers aware of the latest items or offers. It can also play a key role in marketing strategy with lead generation, brand awareness, relationship building, or keeping customers engaged between purchases through different types of marketing emails [11].

There are different types of marketing emails, however, these are the predominant ones [11]:

- **Promotional Emails** – Used to promote special offers, new product releases, and the brand at large;
- **Informational Emails** – Used to share news related to the organization.

In Figure 6, it is possible to observe an example of an Email Marketing message, where an organization promotes a special offer.

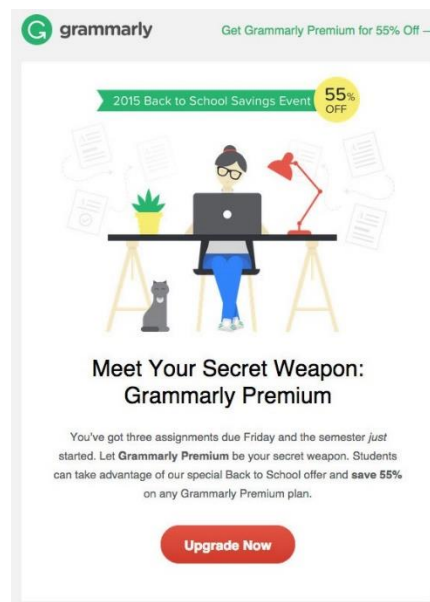


Figure 6 - Example of Email Marketing

3 State of Art

This chapter is dedicated to providing an overview of the study of the current state of transactional communications. Furthermore, it provides a summary of related work, where the applications that exist in the area under study, in which the project work fits, are identified.

3.1 Transactional Communications

Transactional communications or triggered communications are messages that are sent as soon as an action is completed by a user typically using a Web application. Therefore, it consists of a relationship strategy whose objective is to strengthen the ties between the organization and its customers, as well as provide relevant information.

Transactional communications, often in the form of emails or text messages, are a tactic for developing deeper relationships with customers in all stages of the customer life cycle. With this communication, it is possible to welcome new customers or subscribers with a discount or incentive, celebrate a customer's birthday with a special offer, and remind customers of upcoming or missed appointments [14].

Transactional communications are the most relevant, engaging, and timely messages because they are informed by specific consumer behaviors, milestones, and events within a defined time frame or on a specific date [14].

There are two types of transactional communications [14]:

- **Event-Based** – Messages that are triggered by customer actions performed within a defined timeframe (for example when a customer makes a purchase and receives an order confirmation email);
- **Segment-Based** – Messages that are sent when customers fit into a defined set of conditions (for example when a customer enrolls in a Premium Plan and only receives emails related to that plan).

In 2017, according to Lifecycle Marketing, transactional emails had more than double the open rate of standard emails. In addition, it also had more than triple the unique clicks of standard emails [14]. In Figure 7, it is possible to observe the open rate and unique clicks of both the standard emails and transactional emails in 2017.

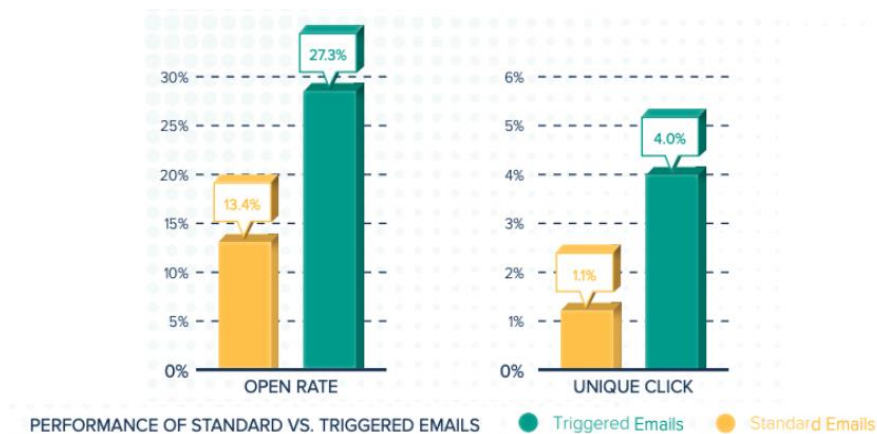


Figure 7 - Open Rate and Unique Clicks in Standard and Transactional Emails [14]

3.1.1 Marketing Communications vs Transactional Communications

Marketing is defined as the set of activities and processes of an organization to create and communicate offers that have value for customers and partners, while Transactional, in the context of communications, comes from the term transaction, which describes an interaction between stakeholders to exchange resources [15].

Table 1 describes some of the main differences between Marketing Communications and Transactional Communications. The main difference between the two is the goal they achieve. Marketing messages seek to exploit the interest of recipients in their offer and try to convert this interest into sales and profit for the organization, while Transactional Messages look to increase the recipient's trust in the organization and its respective offer, to gain loyalty from the customer after a positive interaction between them [15].

Table 1 - Differences between Marketing and Transactional Communications [15]

	Marketing Communications	Transactional Communications
Focus	Single sale	Customer retention
Orientation	Product features	Product benefits
Time	Short time-scales	Long time-scales
Customer Service	Limited	High
Customer Commitment	Limited	High
Customer Contact	Moderate	High
Concern for Quality	Limited	High

The differences between the marketing messages and transactional messages reflect the purpose of each. It exists a transfer in the importance of the organization's offer, where it stops selling the product features at the time of sale to sell the benefits to customers after-sale [15].

3.2 Tools

There are currently several platforms that, like QuickMessage, work in the area of Marketing and send marketing messages and/or transactional messages. To understand the limitations and differentiation factors of each one of the platforms, it will be done a comparative analysis between different platforms, based on the features available in each one of them.

In this analysis, the International platforms **QuickMessage** [16], **MailChimp** [17], **MailJet** [18] and **SMSAPI** [19], and the Portuguese platforms **Sendinblue** [20] and **uSendit** [21], are compared.

3.2.1 QuickMessage

QuickMessage is a platform specialized in Transactional SMS, and therefore, sends transactional SMS messages, however in a non-automated way, with users having to manually trigger their sending in an exact moment. In addition, it is possible to view statistics/reports related to the sending of these messages (how many messages were sent, delivered, and rejected). Finally, users can also create and view SMS templates, which can be used when sending transactional SMS messages.

With this project, the platform is intended to have the feature to send transactional emails, to view statistics/reports related to the sending of these emails, and to create and view email templates. In addition, the possibility to schedule transactional SMS messages and emails is also pretended.

3.2.2 MailChimp

MailChimp is a platform specialized in Email Marketing and Marketing Automation, and therefore, it is possible to create and send email campaigns and automate them. Besides the possibility to create and send email campaigns, the platform allows the sending of transactional emails. In addition, it is possible to view statistics/reports related to the sending of these campaigns (open rate, click rate, and unsubscribes). Finally, users can also create and view email templates, which can be used when creating an email campaign or sending a transactional email.

3.2.3 MailJet

MailJet is a platform specialized in Email Marketing and Marketing Automation, and therefore, it is possible to create and send email campaigns and automate them. Besides the possibility to create and send email campaigns, the platform allows the sending of transactional SMS messages and emails. In addition, it is possible to view statistics/reports related to the sending of email campaigns (open rate, click rate, and unsubscribes) and transactional SMS messages (how many messages were sent, delivered, and rejected). Finally, users can also create and view

email templates, which can be used when creating an email campaign or sending a transactional email.

3.2.4 Sendinblue

Sendinblue is a platform specialized in SMS Marketing, Email Marketing, and Marketing Automation, and therefore, it is possible to create and send email/SMS campaigns and automate them. Besides the possibility to create and send email/SMS campaigns, the platform allows the sending of transactional SMS messages and emails. In addition, it is possible to view statistics/reports related to the sending of email/SMS campaigns (open rate, click rate, and unsubscribes). Finally, users can also create and view email templates, which can be used when creating an email campaign or sending a transactional email.

3.2.5 SMSAPI

SMSAPI is a platform specialized in SMS Marketing, and therefore, it is possible to create and send SMS campaigns. Besides the possibility to create and send SMS campaigns, the platform allows the sending of transactional SMS messages. In addition, it is possible to view statistics/reports related to the sending of these campaigns (open rate, click rate, and unsubscribes) and schedule the sending of these campaigns and transactional SMS messages. Finally, users can also create and view SMS templates, which can be used when creating an SMS campaign or sending a transactional SMS message.

3.2.6 uSendit

uSendit is a platform specialized in SMS Marketing, and therefore, it is possible to create and send SMS campaigns. Besides the possibility to create and send SMS campaigns, the platform allows the sending of transactional SMS messages. In addition, it is possible to schedule the sending of these campaigns and transactional SMS messages. Finally, users can also create and view SMS templates, which can be used when creating an SMS campaign or sending a transactional SMS message.

3.2.7 Comparative Analysis

After analyzing the features of the different platforms, it is possible to observe that the platforms under analysis have different offers in terms of features available to users. While some platforms have a broad spectrum of functionalities, to offer a wide range of features related to marketing, others continue to focus on a single channel (email or SMS).

Table 2 presents the most important and relevant features offered by the different platforms. A green checkmark means that the platform has the feature, a red X means that the platform

doesn't have the feature and an orange exclamation means that the platform doesn't have the feature but it will have it at the end of the project.

Table 2 - Comparative Analysis Between Platforms

	QuickMessage	MailChimp	MailJet	Sendinblue	SMSAPI	uSendit
SMS Marketing	✗	✗	✗	✓	✓	✓
Email Marketing	✗	✓	✓	✓	✗	✗
Transactional SMS	✓	✗	✓	✓	✓	✓
Transactional Email	!	✓	✓	✓	✗	✗
Schedule SMS	!	✗	✗	✗	✓	✓
Schedule Email	!	✗	✗	✗	✗	✗
SMS Reports	✓	✗	✓	✓	✓	✗
Email Reports	!	✓	✓	✓	✗	✗
SMS Templates	✓	✗	✗	✗	✓	✓
Email Templates	!	✓	✓	✓	✗	✗
Marketing Automation	✗	✓	✓	✓	✗	✗

By analyzing Table 2, it is possible to conclude, that among the analyzed platforms, QuickMessage has the least offer of features, since it is still in an early stage of development. However, after the end of this project, it will be able to compete with the rival platforms because it will have a better offer of features.

Regarding SMSAPI and uSendit, it is possible to observe that these platforms are focused on SMS Marketing, while MailChimp is focused on Email Marketing and Marketing Automation.

Regarding the other platforms, MailJet and Sendinblue, it is possible to observe that they have the most complete offer of features. Unlike SMSAPI, uSendit, and MailChimp which only focus on marketing in one channel, MailJet and Sendinblue focus on two channels of marketing (Email and SMS).

4 Value Analysis

This chapter is dedicated to describing the value analysis of the work. It starts with a section related to business process and innovation, where the NCD model is applied. Then, there is a section where the usage of the AHP for multicriteria decisions to aid solve the problem of this work is defined and explained. Finally, it has a section where the customer value and the value proposition are demonstrated with the Canvas business model.

4.1 Innovation Process

Currently, innovation is one of the main strategic drivers of organizations. While the strategic focus of organizations was based on practices of good engineering, currently extra effort is required to create a differentiating factor, due to the standardization of good engineering practices by organizations. Consequently, innovation emerges as a new differentiating factor, and as such, as a new focus of an organization's strategies [22].

In the business concept, innovation can be translated into the creation of a new product or update/maintenance of current products, and the main objective is to add value to the organization and customers [23].

In the creation of a new product or update/maintenance of current products, it is necessary to take into account the need and value. The need is related to the needs and expectations of the customer concerning the product in question. The definition of value varies by context and person, however, given the context of the current project, one can define value as a quantitative value, related to customer satisfaction with the product presented [24].

The innovation process tries to define the life cycle of an idea from its creation to the commercialization of a product. The process is sequential, so only ideas and products whose value exists are expected to reach the commercialization step. Each step of the process has specific and dependent goals on the previous steps [24].

As it can be seen in Figure 8, the innovation process is divided into three distinct steps [25]:

- **Fuzzy Front End** – It is the first step of the innovation process. The main objective involves developing a new feature or developing/updating a new product. This is based on an identified current opportunity, opportunity analysis, generation of ideas, and selection of the idea to be implemented in the next step;
- **New Product Development** – Step of the innovation process responsible for the development and testing of the product. The objective is to transform the concept into a product, carry out market analyses, understand how to reach customers, test the product and make possible changes;

- **Commercialization** – Last step of the innovation process, responsible for promoting and commercializing the developed product. The objective is to sell the created/changed product. In this step, the ideas have already been transformed into products that have been tested and approved.

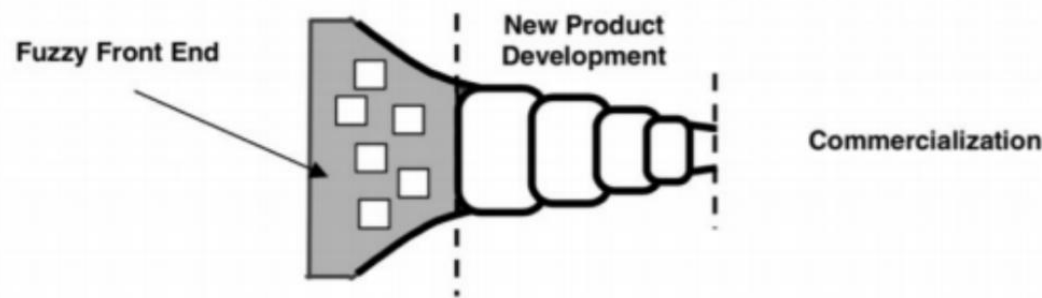


Figure 8 - Innovation Process Steps [25]

4.2 New Concept Development

New Concept Development (NCD) was created in 2002, by Peter Koen, and it is a model whose objective is to create a standard for the first step of the innovation process, Fuzzy Front End. According to Peter Koen, it has been given more attention to the Fuzzy Front End, to optimize the probability of success of new products before they begin their development and commercialization process [25].

To achieve this concept, this methodology defines a sequence of steps that must be performed. Firstly, the existing opportunities must be identified and analyzed which of the existing opportunities represent opportunities to create value for the organization and its customers. After identifying and analyzing opportunities, ideas must be created that can create the previously identified value. These ideas must be analyzed and the best idea must be selected as the basis for defining the concept to be implemented in the next step, New Product Development [25].

In Figure 9, it is possible to observe the five steps of the NCD Model:

- Opportunity Identification;
- Opportunity Analysis;
- Ideas Generation;
- Ideas Selection;
- Concept Definition.

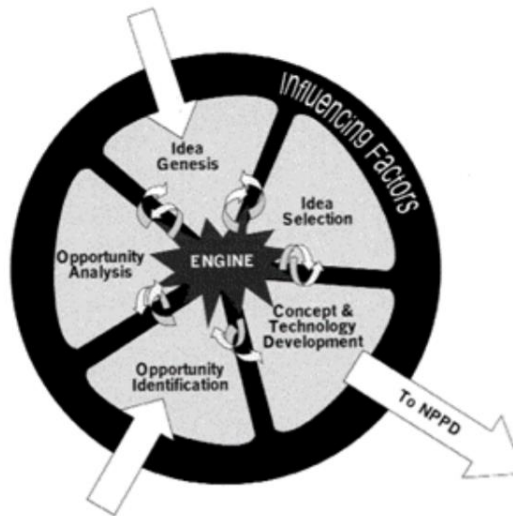


Figure 9 – NCD Model [25]

4.2.1 Opportunity Identification

Opportunity identification is the first step of NCD and is usually motivated by the organization and business objectives, where new opportunities may be related to internal or external factors.

In this project, opportunity identification is motivated by the user needs, prioritizing the constant improvement of products. As such, the problems identified in section 1.3 of Chapter 1, represent the existing opportunity to improve the product QuickMessage and make the platform more competitive. This improvement consists mainly of the implementation of a new Email channel (transactional emails, email reports, and emails templates) and the implementation of a feature to allow the scheduling of the SMS messages and Emails.

After the opportunity has been identified, it is necessary to analyze its value to confirm that it is advantageous to pursue the opportunity.

4.2.2 Opportunity Analysis

To identify the value that the opportunity can add to the product, additional information is needed, which involves planning carried out through market studies and scientific experiments. Opportunity analysis should be an integral part of a formal and incremental process, whose objective is to identify the strengths and weaknesses of the opportunity and draw conclusions regarding the value it adds to the product as well as the cost to pay for its development [25].

A complete opportunity analysis should include [25]:

- **Strategic Framework** – Determining whether the opportunity fits the strengths, weaknesses, and threats of the market where the organization operates;

- **Market Segment Evaluation** – Market description and analysis of driving factors, such as economic, technological, and cultural factors;
- **Competitors Analysis** – Determine the biggest competitors present in the market segment identified;
- **Customer Evaluation** – Determining what customer needs are intended to be satisfied by the creation of the new product.

4.2.2.1 Strategic Framework

Regarding the strategic framework, to determine the strengths, weaknesses, opportunities, and threats, a SWOT analysis is carried out.

SWOT analysis is a tool used to assess the competitive position of a company, its products, and its ideas, based on internal and external factors.

A visual representation of the scheme of a SWOT analysis contains the following components:

- **Strengths** – Represent internal factors that approve the idea being evaluated;
- **Weaknesses** – Represent internal factors that disapprove of the idea being evaluated;
- **Opportunities** – Represent external factors that approve the idea being evaluated;
- **Threats** – Represent external factors that disapprove of the idea being evaluated.

In Figure 10, it is possible to observe a SWOT Analysis related to the opportunity to improve the product QuickMessage and make the platform more competitive, by implementing a new Email channel and a feature to allow the scheduling of SMS messages and Emails. This analysis aims to identify strengths, weaknesses, opportunities, and threats associated with the opportunity under analysis, to understand whether the opportunity can generate value for the company and its users.



Figure 10 - SWOT Analysis

With the SWOT Analysis, it was identified three distinct strengths related to the opportunity under analysis. The first strength is the **direct response to requests from current platform users**. Customers have not been fully satisfied with the actual platform, mainly because of the lack of

features, so they have made several requests, which include the features mentioned before, that will be met with this improvement of the product. The second strength is the **offer of a wider range of communications channels available**. With the implementation of the new Email Channel, not only will it be possible to send transactional SMS messages, but it will also be possible to send transactional Emails, which makes the platform omnichannel instead of uni-channel. The third strength is related to **making the platform richer in terms of features**. With the implementation of the new Email Channel and the feature to allow the scheduling of the SMS messages and Emails, the platform will be richer and it will be able to compete with rival platforms.

Regarding the weaknesses, it was identified two weaknesses related to the opportunity. The first weakness is the **costs and time of production of the new features**. Since it will be required someone to develop, implement and test the solution with the new features, it will be necessary to cover those costs. The second weakness is the **increased costs of usage and maintenance of the platform**. Since the usage of the platform will increase and it will be more difficult to maintain the application with the new features, it will be more expensive the usage and maintenance of the platform.

In terms of opportunities, it was identified four opportunities, the **increased usage of the platform**, the possibility to **retain more customers** and **attract new customers**, and the fact that **existing competitors don't have a detailed feature to schedule SMS messages and Emails**.

Finally, regarding the threats, it was identified two threats related to the opportunity under analysis. The first threat is the appearance of **new competitors**. These new competitors might have new, better, and more detailed features related to transactional communications and can be a threat to the business. The second threat is the fact that **existing competitors are more well-known and already have the feature to send transactional emails**. Since there are already some well-known competitors specialized in the area of sending transactional emails, it will be hard to compete with them.

In short, through the SWOT analysis, is concluded that the **opportunity is valid and that it should be chased**. This happens because it was identified that it is an opportunity to answer direct customer requests on the subject, bringing value to them and the platform. Also, weaknesses and threats are not impediments to the implementation of the project associated with the opportunity.

4.2.2.2 Market Segment Evaluation

The market segment considered is the market of transactional SMS and transactional Email. This market has been on the rise since 2004, especially the transactional Email market, as shown in Figure 11, which represents the Google Trends trend graph.

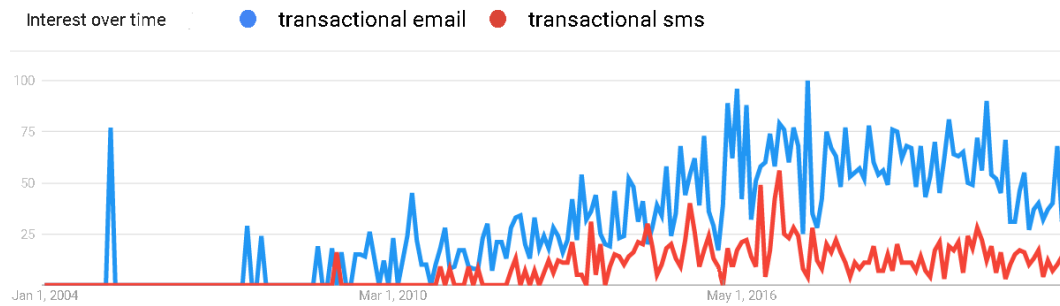


Figure 11 - Interest Over Time of "Transactional Email" and "Transactional SMS" between 2004 and 2022

To present an assessment of the market segment, it is necessary to study the competition in the market in question. Through the analysis presented in section 3.2.7 of Chapter 3, it was identified two main competitors: MailJet and Sendinblue.

MailJet was created in 2010 and since 2012, it shows a rise in terms of the number of users and interest over time, as is possible to observe in Figure 12. Currently, MailJet serves customers in more than 150 countries and sends 2 billion emails per month [18].



Figure 12 - Interest Over Time of MailJet between 2004 and 2022

Sendinblue was created in 2007 and since 2016, it shows a rise in terms of the number of users and interest over time. Currently, Sendinblue serves 80.000 customers in 160 countries and sends 100 million emails per day [20].

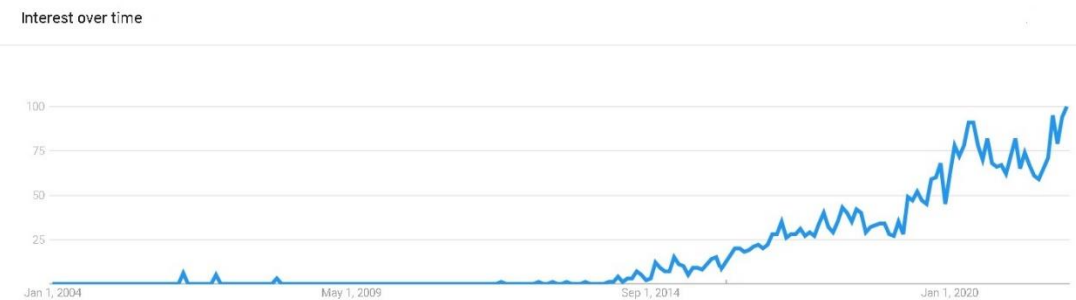


Figure 13 - Interest Over Time of Sendinblue between 2004 and 2022

4.2.2.3 Competitors Analysis

The main competitors were identified in the market segment evaluation section, and are MailJet and Sendinblue.

The main advantage of the opportunity under analysis is the fact that the main competitors do not have a feature to schedule SMS messages and Emails. With the implementation of this feature, the platform will have a feature that differentiates itself from the rival platforms. In addition, with the implementation of the Email Channel, the platform will be able to compete with the main competitors, because instead of having only one channel, it will have the same number of channels as the competitors (SMS and Email).

4.2.2.4 Customer Evaluation

The opportunity is aimed at customers who need a platform that not only allows the sending of transactional SMS messages but also the sending of transactional Emails. In addition, with the development of the opportunity, it will be suppressed the problem of the customers that have not been fully satisfied with the platform, mainly because of the lack of features.

4.2.3 Ideas Generation

The ideas generation consists of its birth, development, and maturation. The process is evolutionary and ideas are raised, rejected, combined, modified, and evolved [25].

After evaluating the identified opportunity, it is necessary to generate possible solutions that can be implemented to achieve the objectives associated with the identified opportunity. To generate possible solutions, it is necessary to analyze the platform and have discussions with elements of the work team, to identify possible ways to achieve the goals set [25].

Thus, after analyzing the platform and having a discussion with elements of the work team, three different approaches were raised:

- Implementation of the new features (Email channel and scheduling of SMS messages and Emails) in the existing QuickMessage application;
- Decomposition of the existent QuickMessage monolithic application into microservices (SMS Management, Email Management, and User Management) and implementation of the new features in the respective microservice;
- Creation of a new application similar to QuickMessage, but only for the Email channel, and implementation of the feature of scheduling SMS messages and Emails in the respective application.

The creation of a new application similar to QuickMessage is an approach identified however it would require changing the name of the existent QuickMessage to QuickMessageSMS and the new QuickMessage to QuickMessageEmail.

Although the proposed approaches have significant differences between them, they allow reaching the outlined goals. As such, it is necessary to evaluate these ideas, to understand which is most suitable for the project at hand. Given the importance of the decision to be made, it becomes necessary to base the decision on a complex analysis, based on key criteria for the execution of the project.

4.2.4 Ideas Selection

After generating ideas, it is necessary to evaluate them and select which idea will serve as the base for the solution to be implemented. As this is a crucial decision in the innovation process, organizations use methods to support decision-making at this step of the process.

A decision-making support method is the Analytic Hierarchy Process (AHP) method. This method was created in the 1970s by Thomas Saaty and it consists of deconstructing the idea in parts as small as possible, to facilitate their analysis. As such, the method AHP is divided into five distinct parts: Hierarchical Division, Establish Relative Priorities, Evaluate Consistency of Relative Priorities, Build Parity Comparison Matrix, and Choice of Alternative [26].

4.2.4.1 Hierarchical Division

Hierarchical Division consists of identifying the final objective, the comparison criteria, and the implementation alternatives in a hierarchy. The objective is at the top of the hierarchy, being considered the main element of the hierarchy, and the decisions made must not jeopardize the defined objective. At the next level of the hierarchy is the comparison criteria, which may include restrictions imposed on the implementation of the project, such as budget constraints and technology constraints. These criteria should influence the selection of the option from among the possibilities. The last level of the hierarchy is the different approaches, that are being analyzed by the method [26].

Taking into account the project described in this document, the objective is to improve the product QuickMessage and make the platform more competitive, by implementing a new Email channel and a feature to allow the scheduling of the SMS messages and Emails. The options to be evaluated were defined previously, as such, it is necessary to define the criteria comparison, considering the project and its conditions:

- **Development Time** – Time required to spend in the development of the solution;
- **Complexity** – Level of difficulty related to the implementation and maintenance of the solution;
- **Technological Resources** – Technological elements needed for the solution, such as the processor, memory, disk, bandwidth, number of servers, and number of databases.

Thus, the three elements of the first part of the AHP method are defined (Objective, Comparison Criteria, and Solutions). As such, it is possible to make a representation of the hierarchical structure of the AHP method, as shown in Figure 14.

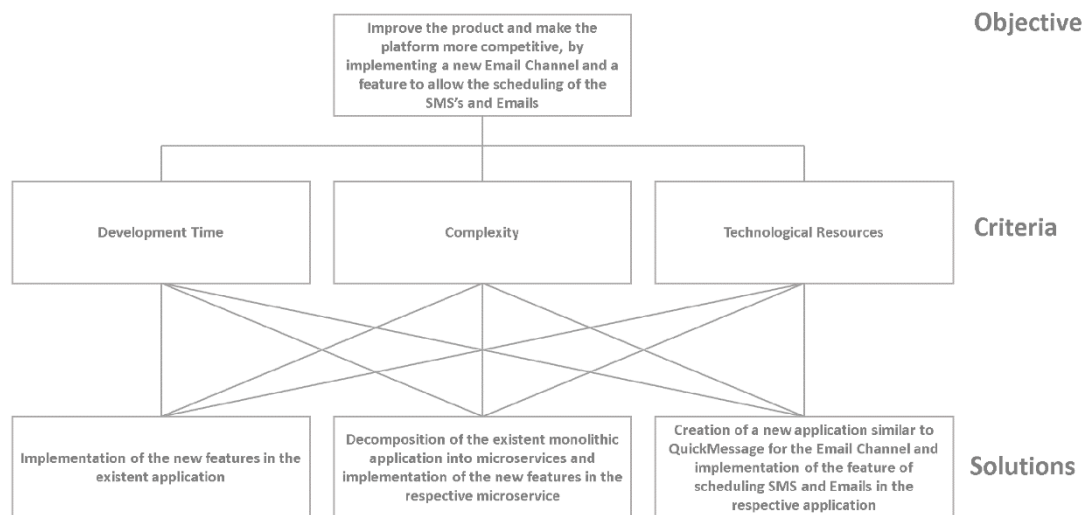


Figure 14 - Hierarchical Structure of AHP

4.2.4.2 Establish Relative Priorities

The second part of the AHP method is to Establish Relative Priorities. In this part, the relative priorities of each of the existing comparison criteria are defined using the Saaty fundamental scale.

The Saaty Scale, introduced in 1985 by Thomas Saaty and Kevin Kearns, is a pair comparison scale used to compare alternatives of the AHP [27].

When using the Saaty scale to compare alternatives, a matrix must be created for comparing these alternatives. These alternatives must be listed in the same order, horizontally and vertically in the matrix. This way, it is possible to fill the diagonal line of the matrix, which starts at the top-left cell and ends at the bottom-right cell, with the value 1. This happens because according to Saaty, “the relative importance of any element compared to itself is 1” [27].

In that same matrix, the remaining cells are filled with the comparison between criteria. It should be noted that reciprocal values are entered for reverse comparisons. Comparing criterion A and criterion B, if criterion A is considered to have weak importance concerning criterion B (equivalent to a rating of 3 on the scale of Saaty), then criterion B is assumed to have a reciprocal evaluation (equivalent to a value assessment 1/3, on the Saaty scale) [27].

In Table 3, it is possible to observe a representation of the Saaty scale.

Table 3 - Saaty Scale

Importance Level	Definition	Explanation
1	Equal Importance	Both criteria contribute equally to the objectives
3	Weak Importance	Experience and judgment slightly favor one criterion over the other
5	Strong Importance	Experience and judgment heavily favor one criterion over the other
7	Very Strong Importance	One criterion is very strongly favored over another
9	Absolute Importance	Evidence favors one criterion over another with the highest degree of certainty
2, 4, 6, 8	Intermediate Values	When looking for a compromise condition between two definitions

In Table 4 and Table 5, the comparison between the criteria previously defined is shown, considering the following:

- **A:** Development Time
- **B:** Complexity
- **C:** Technological Resources

Table 4 – Second Level Criteria Comparison Matrix

	A	B	C
A	1	0.25	0.5
B	4	1	2
C	2	0.5	1

To obtain the relative priorities, it is necessary to normalize the second level criteria comparison matrix and calculate the arithmetic mean of the values of each row of the normalized matrix.

Table 5 - Second Level Criteria Normalized Matrix

	A	B	C	Relative Priority
A	1/7	1/7	1/7	0.1428
B	4/7	4/7	4/7	0.5714
C	2/7	2/7	2/7	0.2857

According to the values of relative priorities presented in Table 5, it is possible to affirm that complexity is the most important criterion, followed by the technological resources and development time, respectively.

After prioritizing the criteria, it is necessary to verify their consistency, since prioritization is subjective.

4.2.4.3 Evaluate Consistency of Relative Priorities

To evaluate the consistency of relative priorities, it is necessary to calculate the Consistency Ratio (CR) of the criteria.

The first step is to calculate the Consistency Index (CI), whose formula is:

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

To calculate $\lambda \max$, it is necessary to multiply the second level criteria comparison matrix with the relative priorities (self vector). With the values of the resulting matrix, it will be done an average of the division of the resulting matrix values with the relative priorities.

$$\begin{aligned} \begin{bmatrix} 1 & 0.25 & 0.5 \\ 4 & 1 & 2 \\ 2 & 0.5 & 1 \end{bmatrix} \times \begin{bmatrix} 0.14 \\ 0.57 \\ 0.29 \end{bmatrix} &\cong \lambda \max \begin{bmatrix} 0.14 \\ 0.57 \\ 0.29 \end{bmatrix} \Leftrightarrow \\ \Leftrightarrow \begin{bmatrix} 0.43 \\ 1.71 \\ 0.86 \end{bmatrix} &\cong \lambda \max \begin{bmatrix} 0.14 \\ 0.57 \\ 0.29 \end{bmatrix} \\ \Leftrightarrow \lambda \max = \text{average} \left\{ \frac{0.43}{0.14}, \frac{1.71}{0.57}, \frac{0.86}{0.29} \right\} &\simeq 3.01 \end{aligned}$$

Considering the value of $\lambda \max$ being 3.01 and n , that is the number of criteria, being 3, the CI can be calculated.

$$CI = \frac{3.01 - 3}{3 - 1} \simeq 0.005$$

Now that the CI is calculated, the second step is to obtain the Random Index (RI), which is obtained through a table depending on the value of n . This table is presented in Table 6.

Table 6 - Random Index Table

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Since the value of n is 3, the RI obtained from Table 6 is 0.58.

Now that it was obtained the CI and the RI, it is possible to calculate the CR.

$$CR = \frac{CI}{RI} = \frac{0.005}{0.58} \simeq 0.0086 < 0.1$$

Since $CR < 0.1$, the data is considered consistent and the final result of the process is valid.

4.2.4.4 Build Parity Comparison Matrix

After evaluating the consistency of the relative priorities, performing the prioritization of the solutions for each criterion is needed.

From Table 7 to Table 12, the comparison between the solutions for each criterion is shown, considering the following:

- **X:** Implementation of the new features (Email channel and scheduling of SMS messages and Emails) in the existing QuickMessage application.
- **Y:** Decomposition of the existing QuickMessage monolithic application into microservices (SMS Management, Email Management, and User Management) and implementation of the new features in the respective microservice.
- **Z:** Creation of a new application similar to QuickMessage, but only for the Email channel, and implementation of the feature of scheduling SMS messages and Emails in the respective application.

Table 7 - Parity Comparison Matrix of Development Time

	X	Y	Z
X	1	3	5
Y	0.33	1	2
Z	0.20	0.5	1

Table 8 - Parity Comparison Matrix of Complexity

	X	Y	Z
X	1	2	1
Y	0.5	1	0.5
Z	1	2	1

Table 9 - Parity Comparison Matrix of Technological Resources

	X	Y	Z
X	1	3	2
Y	0.33	1	0.5
Z	0.5	2	1

To obtain the relative priorities, it is necessary to normalize the parity comparison matrixes and calculate the arithmetic mean of the values of each row of the normalized matrix.

Table 10 - Parity Comparison Normalized Matrix of Development Time

	X	Y	Z	Relative Priority
X	15/23	2/3	5/8	0.6479
Y	5/23	2/9	1/4	0.2299
Z	3/23	1/9	1/8	0.1222

Table 11 - Parity Comparison Normalized Matrix of Complexity

	X	Y	Z	Relative Priority
X	2/5	2/5	2/5	0.4
Y	1/5	1/5	1/5	0.2
Z	2/5	2/5	2/5	0.4

Table 12 - Parity Comparison Normalized Matrix of Technological Resources

	X	Y	Z	Relative Priority
X	6/11	1/2	4/7	0.539
Y	2/11	1/6	1/7	0.2114
Z	3/11	1/3	2/7	0.2973

Through the analysis of the three tables with relative priorities, it is possible to conclude that for the criteria of development time and technological resources, **Solution X** is the most convincing by a great margin. Regarding the criterion of complexity, there is an equal benefit value for **Solution X** and **Solution Z**.

4.2.4.5 Choice of Alternative

After completing the previous steps, it is necessary to choose one of the solutions. This will be done by using a matrix with the relative priorities of the solutions for each criterion presented in Table 13, and a matrix with the relative priorities of the criteria presented in Table 14.

Table 13 - Solutions Relative Priorities Matrix for each Criterion

	A	B	C
X	0.65	0.4	0.54
Y	0.23	0.2	0.21
Z	0.12	0.4	0.30

Table 14 - Criteria Relative Priorities Matrix

	Relative Priority
X	0.14
Y	0.57
Z	0.29

Multiplying the matrix presented in Table 13 with the matrix presented in Table 14 gives rise to the composite priority matrix, presented in Table 15.

Table 15 - Composite Priority Matrix

	Composite Priority
X	0.4756
Y	0.2071
Z	0.3318

Through the analysis of Table 15, it appears that the solution that obtained the greatest result was **Solution X**. In conclusion, the implementation of the new features (Email channel and scheduling of SMS messages and Emails) in the existing QuickMessage application, is the best solution to solving the problem.

4.2.5 Concept Definition

After using AHP to decide which of the three solutions to implement, it is important to define the concept. The concept of this project consists of a system that allows mainly the creation and sending of transactional emails by a HyperText Markup Language (HTML) Editor. In addition, it should exist an area where it is possible to create and view existing email templates, as well as an area where reports related to the sending of the emails are seen. Finally, it should be possible to schedule the sending of transactional SMS messages and Emails. Regarding the interface, it must be simple and intuitive for the user.

4.3 Solution Value

After identifying the opportunity and choosing the solution to implement, it is important to understand how this solution will contribute to adding value to the organization and its customers. As such, key concepts related to value will be analyzed, and a value proposition of the solution will be presented, along with the Canvas business model.

4.3.1 Value, Customer Value, and Perceived Value

The concept of value is used in several ways, in many cases, the value to the organization is different from the value to the customer. However, it is important to define the value of a product, which consists of the customer's view on the relationship between the benefits that the product offers and the cost of the product. There must be a constant concern within an organization to offer the customer a product with value [28].

For the customer, value is related to their perception or experience with a product or service, generating a psychological or emotional connection. Once it has a subjective base, the value to the customer is different for each individual. Regarding this project, the fact that it offers a

platform with a simple and intuitive interface, inside the QuickMessage product, can generate value for the customer because the customer will have a pleasant experience in its use [28].

Perceived value is considered the most important indicator when referring to a customer's intention to repurchase a product. It consists of the result of the difference between benefits and sacrifices for the customer. The perceived value varies from client to client as the weights of benefits and sacrifices are measured in a personal way. Within the scope of the project, it is possible to consider the benefits of the solution as the increase in the number of transactional channels and the possibility to schedule SMS messages and Emails. On the other hand, it is also possible to consider the learning difficulty to use the HTML Editor to send Emails and the effort to create new Email templates as customer sacrifices [29].

4.3.2 Value Proposition and Canvas Business Model

A value proposition can be understood as the benefit statements that are delivered by an organization to its external constituents. It aims to present the reasons why a customer should purchase a product or service and what benefits it will add to the customer. It must be explicit what the product or service consists of, the value obtained by the customer as well as what the product or service simplifies the customer's life [30].

Within the scope of the project, the value proposition consists of an offering of a platform that allows the sending of transactional SMS messages and Emails, which stands out for the possibility to schedule the sending of such messages, with a simple and intuitive interface, differentiating itself from the competitors. This solution will increase the number of transactional channels available in the QuickMessage platform and will help satisfy customers' needs, since they have not been fully satisfied with the current platform, mainly because of the lack of features.

The Canvas Business Model is a model composed of nine interconnected blocks that allow discussion and analysis of new business proposals. The analysis allows verifying how the organization generates value for customers. The Canvas business model of the project of this dissertation is described in Figure 15.

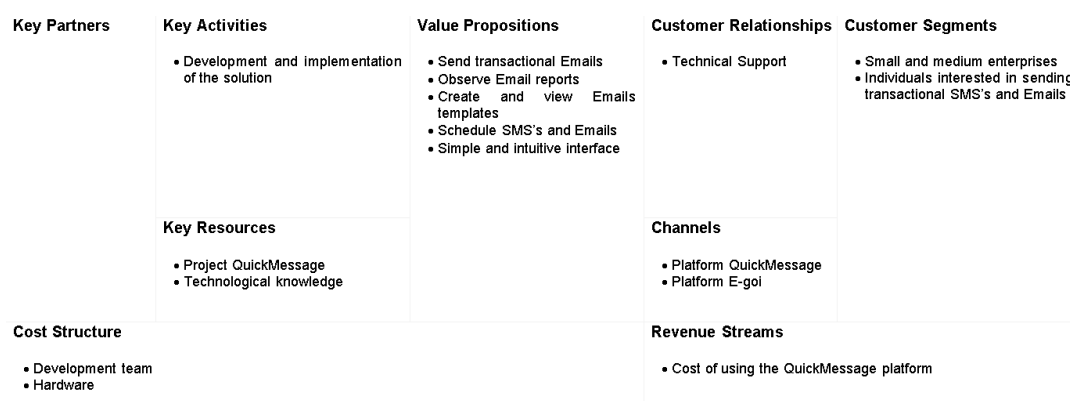


Figure 15 - Canvas Business Model

The nine blocks represented in Figure 15 are:

- **Key Partners** – This block aims to list all external partners of the organization that are essential for the realization of the idea in a product or service. Within the scope of the project, there is no supplier network to consider.
- **Key Activities** – This block aims to understand the activities that will give rise to the product or service and that are essential to achieving this objective. Within the scope of the project, the activities include the development and implementation of the defined solution.
- **Key Resources** – This block aims to list all the resources which without it, it is impossible to materialize the idea in a product or service. Within the scope of this project, the necessary resources are the basis of the QuickMessage project, as well as the underlying technological knowledge.
- **Value Propositions** – This block consists of the presentation of the reasons why the customer chooses to use the product or service over others. Within the scope of the project, the value proposition consists of a modified platform with a simple and intuitive interface that allows the user to have a pleasant experience. The modified platform includes the features to send transactional emails, observe email reports, create and view email templates and schedule SMS messages and Emails.
- **Customer Relationships** – This block explains how the relationship between the organization and the customer will be to the product in question. Within the scope of the project, technical support is always provided when necessary.
- **Channels** – This block defines the communication and sales channels of the product or service. Within the scope of the project, the channels to be used are the communication channels of the QuickMessage product and the E-goi platform to keep the customer always up to date and informed.
- **Customer Segments** – This block explains the types of customers that are expected to buy a product or service. Within the scope of the project, the expected customers are small and medium-sized enterprises, as well as individuals, that need to send transactional SMS messages and Emails to communicate with their customers.
- **Cost Structure** – This block aims to understand the costs inherent to the implementation of the idea. Within the scope of the project, the costs involved are the need for hardware, such as servers, and development teams to implement the solution.
- **Revenue Streams** – This block explains all the sources of return for a product or service. Within the scope of this project, all returns are expected to consist of the customer paying for the use of the product.

5 Analysis

This chapter is dedicated to demonstrating the requirements engineering, that englobes the identification of the domain model, functional and non-functional requirements, and use cases.

5.1 Domain Model

To understand the problem under analysis, it is important to know and understand the domain concepts and the way they are related. As such a domain model is used, which the main purpose is to show, through a structured visual representation, the business concepts alluding to the project.

Figure 16 shows the domain model defined using the Unified Modeling Language (UML) notation, in which the conceptual classes are organized according to their interconnections and respective cardinalities.

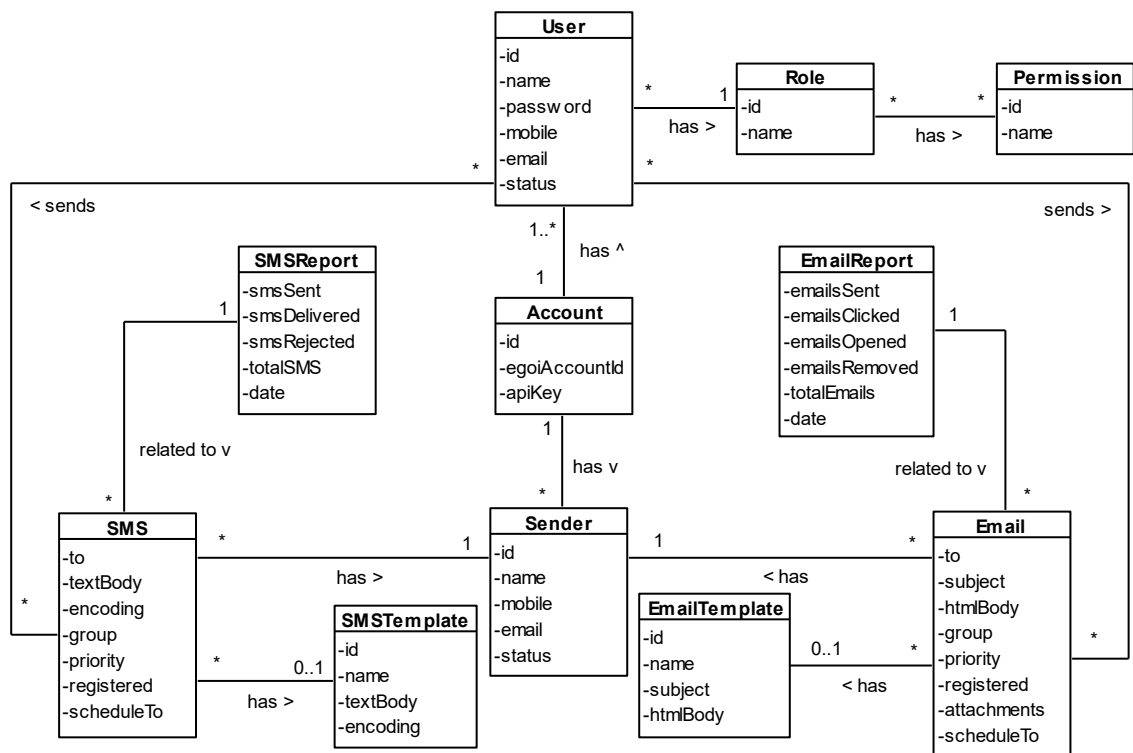


Figure 16 - Domain Model

According to Figure 16, there are 11 distinct domain concepts:

- **Account** – Represents the accounts of the customers on the platform. An account has many senders and many users. Each account comes with a default user, which has an admin role with all permissions.
- **User** – Represents the users in the platform. It is the user who performs actions on the platform, such as sending SMS messages and Emails. Many users can be created and can only have one role.
- **Role** – Represents the roles of the users in the platform. Each role has a name and a list of permissions associated with it.
- **Permission** – Represents the permissions associated with a role. Each permission has a name, which represents the action the user can perform with it.
- **Sender** – Represents the senders created and configured within an account. A sender can have a mobile and an Email configured, which is the contact to be used when sending SMS messages and Emails, respectively. The status of the sender represents if it is validated or not.
- **SMS** – Represents the SMS messages that users can send inside the platform. Each SMS has the contact of the recipient, the sender, the content of the message, the encoding, and if it is a registered SMS message or not. Optionally, it can have the date and time that is pretended to be sent.
- **SMSTemplate** – Represents the SMS message templates that can be created to be used when sending SMS messages.
- **SMSReport** – Represents the SMS messages reports that are generated with the sending of SMS messages on a specific date. It shows how many SMS messages were sent, delivered, rejected, and their total.
- **Email** – Represents the Emails that users can send inside the platform. Each Email has the contact of the recipient, the sender, the subject, the content of the email in HTML, and if it is a registered Email or not. Optionally, it can have the date and time that is pretended to be sent, as well as attachments.
- **EmailTemplate** – Represents the Emails templates that can be created to be used when sending Emails.
- **EmailReport** – Represents the Emails reports that are generated with the sending of Emails on a specific date. It shows how many Emails were sent, clicked, opened, removed, and their total.

From the previously identified domain concepts, only the Email, EmailTemplate, and EmailReport are new, however, some concepts were changed. The SMS concept was changed to support the scheduling of SMS messages, and the Sender concept was changed to support the sending of Emails since the email contact of the recipient is needed. The remaining concepts are concepts that already exist within the QuickMessage platform, and which will form the basis of the solution to solve the identified problem.

5.2 Functional Requirements

Functional requirements are requirements that specify features that the system must be able to perform. They capture the features from the point of view of the user and are related to the problems and needs that must be solved by the solution.

After analyzing the current solution and rival platforms, and having meetings with the product owner, the following new functional requirements were reached:

- FR.1 Send Quick Email** – With this requirement, users must be able to send quick emails with attachments. A quick email, in this project, is considered an email that is sent without the import of an Excel with data.
- FR.2 Send Advanced Email** – With this requirement, users must be able to send advanced emails with attachments. An advanced email, in this project, is considered an email that is sent with the import of an Excel with data.
- FR.3 Create Email Template** – With this requirement, users must be able to create emails templates.
- FR.4 View Email Templates** – With this requirement, users must be able to visualize the list of email templates.
- FR.5 Manage Email Templates (delete/edit)** – With this requirement, users must be able to delete and edit existing email templates.
- FR.6 View Email Reports/Statistics** – With this requirement, users must be able to visualize reports and statistics related to the email sending.
- FR.7 View Email Senders** – With this requirement, users must be able to view the list of email senders created and configured within the account.
- FR.8 Schedule SMS Message** – With this requirement, users must be able to schedule the sending of a quick or advanced SMS message.
- FR.9 Schedule Email** – With this requirement, users must be able to schedule the sending of a quick or advanced email.
- FR.10 View Scheduled SMS Messages** – With this requirement, users must be able to view the list of scheduled SMS messages.
- FR.11 View Scheduled Emails** – With this requirement, users must be able to view the list of scheduled emails.
- FR.12 Manage Scheduled SMS Messages (delete/edit)** – With this requirement, users must be able to delete and edit existing scheduled SMS messages. Only SMS messages that haven't yet been sent can be edited or deleted.
- FR.13 Manage Scheduled Emails (delete/edit)** – With this requirement, users must be able to delete and edit existing scheduled emails. Only emails that haven't yet been sent can be edited or deleted.

All of these functional requirements are tested according to the metrics defined in Chapter 8.

In addition to the new specified functional requirements, the platform already has features that can be performed related to the following old functional requirements:

- **Send Quick SMS Message** – This requirement allows users to send quick SMS messages. A quick SMS message, in this project, is considered an SMS message that is sent without the import of an Excel with data.
- **Send Advanced SMS Message** – This requirement allows users to send advanced SMS messages. An advanced SMS message, in this project, is considered an SMS message that is sent with the import of an Excel with data.
- **Create SMS Messages Template** – This requirement allows users to create SMS messages templates.
- **View SMS Messages Templates** – This requirement allows users to visualize the list of SMS messages templates.
- **Manage SMS Messages Templates (delete/edit)** – This requirement allows users to delete and edit existing SMS messages templates.
- **View SMS Messages Reports/Statistics** – This requirement allows users to visualize reports and statistics related to the SMS message sending.
- **View SMS Messages Senders** – This requirement allows users to view the list of SMS messages senders created and configured within the account.
- **Create Users** – This requirement allows users to create other users, as well as configure their roles.
- **View Users** – This requirement allows users to view the list of users created within the account.
- **Manage Users (delete/edit)** – This requirement allows users to delete and edit existing users.
- **Create Role** – This requirement allows users to create roles, as well as configure their permissions.
- **View Roles** – This requirement allows users to view the list of roles created within the account.
- **Manage Roles (delete/edit)** – This requirement allows users to delete and edit existing roles.

5.3 Non-Functional Requirements

Non-functional requirements are requirements that do not directly interfere with the system. Instead, they establish how the system behaves in specific situations.

In this project, it is possible to identify different types of non-functional requirements, such as quality attributes (measurable characteristics of interest to users and developers), constraints (limitations and restrictions), and concerns (design decisions that should be made).

5.3.1 Quality Attributes

In this project, it is possible to identify five quality attributes, which are associated with all of the new use cases:

- QA-1** The new functional requirements must be secured by a new set of permissions, which must also be possible to add to the user's roles (maintainability).
- QA-2** The application architecture and code must be documented (maintainability).
- QA-3** All features of the system must have an adequate response time, of less than 1 second (efficiency).
- QA-4** The system must deal with all errors and present the unexpected errors to the user with a clear message about the error (usability).
- QA-5** The interaction between users and the system must be simple, intuitive and fully adapted to the action in question (usability).

All of these quality attributes are analyzed in more detail later in Chapter 6.

5.3.2 Constraints

In this project, it is possible to identify four constraints:

- CON-1** The system must use the Transactional API, provided by E-goi, to send transactional messages.
- CON-2** It must be used Java and the Spring Framework for the backend development.
- CON-3** It must be used Typescript and Angular for the frontend development.
- CON-4** The server should provide a Representational State Transfer (REST) Application Programming Interface (API) for client-side communication.

5.3.3 Concerns

In this project, it is possible to identify two concerns:

- CRN-1** The system must be designed to be easily scalable and easy to maintain.
- CRN-2** Should be adopted good practices of object-oriented software analysis and design.

5.4 Use Cases

To be able to satisfy all the functional requirements specified above, several use cases were created. In Figure 17, it is possible to observe all the use cases defined, each one related to one of the functional requirements reached before.

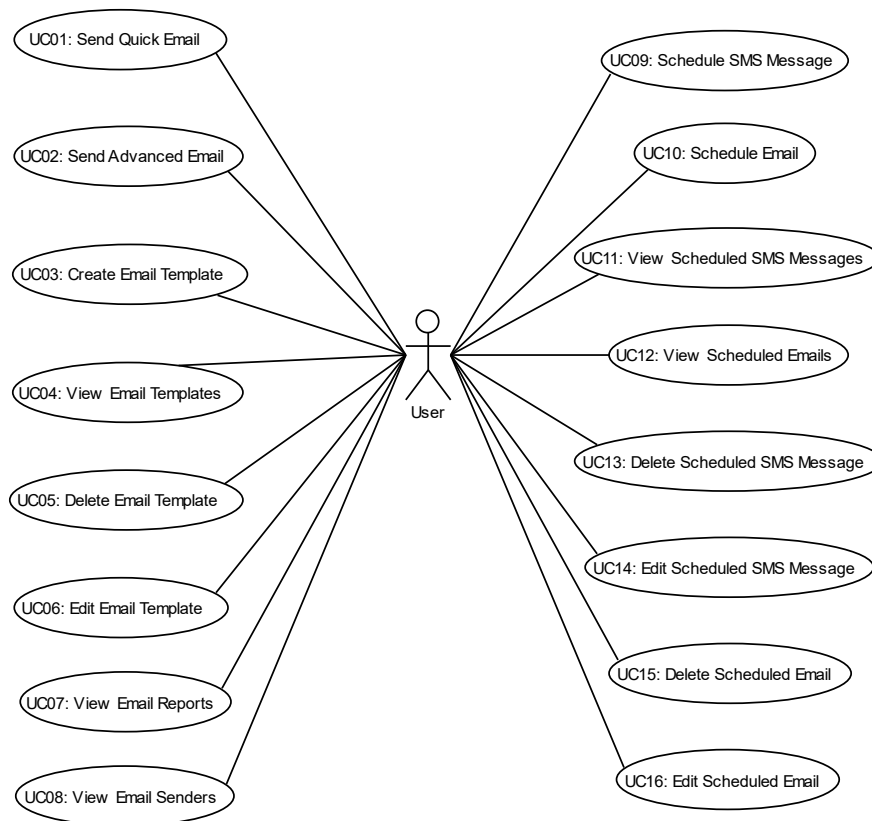


Figure 17 - Use Cases Diagram

In this way, and following Figure 17, for each use case, its functionality will be described, with a System Sequence Diagram (SSD).

5.4.1 Send Quick Email

This use case is related to the **FR1**.

To perform this use case, the user must have the required permission and enter the page that allows the sending of quick emails.

When the user enters the page that allows the sending of quick emails, the system requests the recipient list, which the user enters. After that, the system requests the sender, subject, message, and attachments, which the user enters (the user can also select a template that will fill the required fields). After that, the system requests if it is a registered email or not, which the user also enters. Finally, the system requests the action confirmation, which the user confirms, and then the system informs the success of the action.

In Figure 18, it is possible to observe the SSD related to this use case.

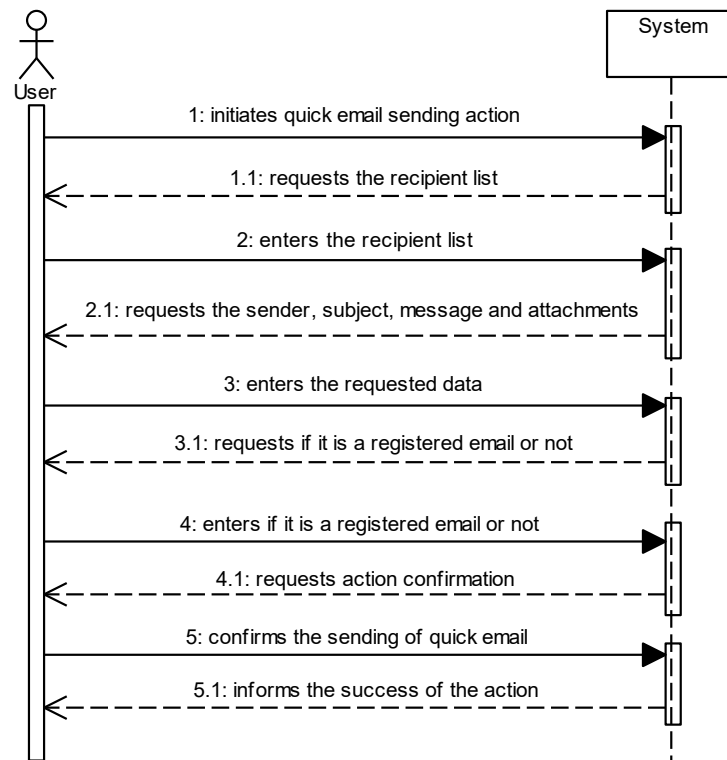


Figure 18 - Send Quick Email SSD

5.4.2 Send Advanced Email

This use case is related to the **FR2**.

To perform this use case, the user must have the required permission and enter the page that allows the sending of advanced emails.

When the users enter the page that allows the sending of advanced emails, the system requests the Excel with data to import, which the user imports. After that, the system requests the column of the Excel that contains the recipient contact, which the user enters. After that, the system requests the sender, subject, message, and attachments, which the user enters (the user can also select a template that will fill the required fields). After that, the system requests if it is a registered email or not, which the user also enters. Finally, the system requests the action confirmation, which the user confirms, and then the system informs the success of the action.

In Figure 19, it is possible to observe the SSD related to this use case.

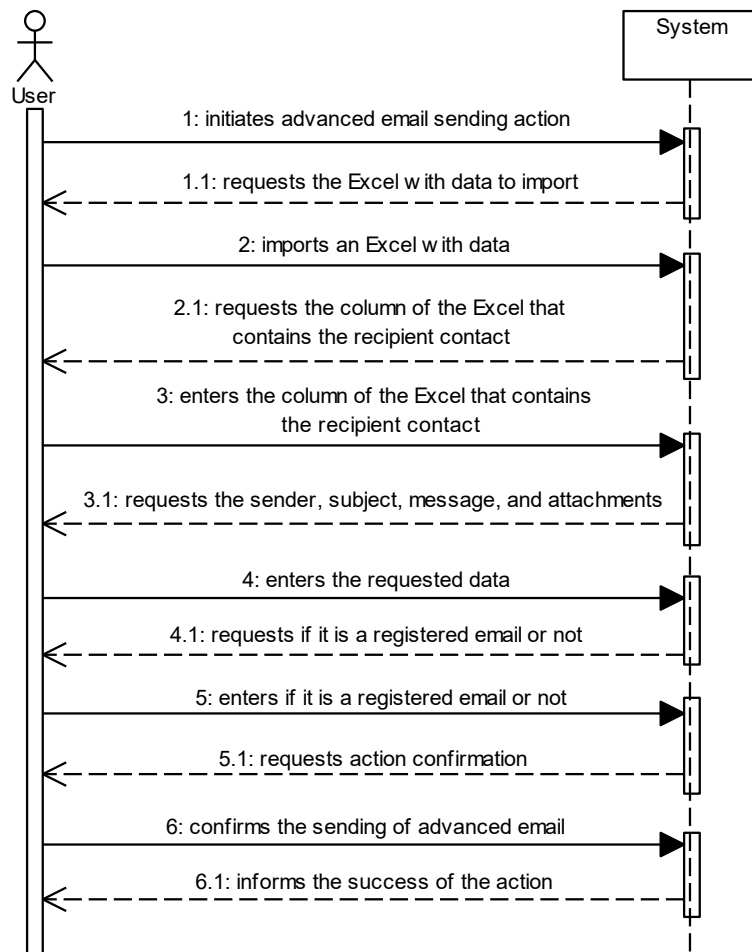


Figure 19 - Send Advanced Email SSD

5.4.3 Create Email Template

This use case is related to the **FR3**.

To perform this use case, the user must have the required permission and enter the page that allows the creation of emails templates.

When the user enters the page that allows the creation of email templates, the system requests the template name, subject, and message, which the user enters. After that, the system informs the success of the action.

In Figure 20, it is possible to observe the SSD related to this use case.

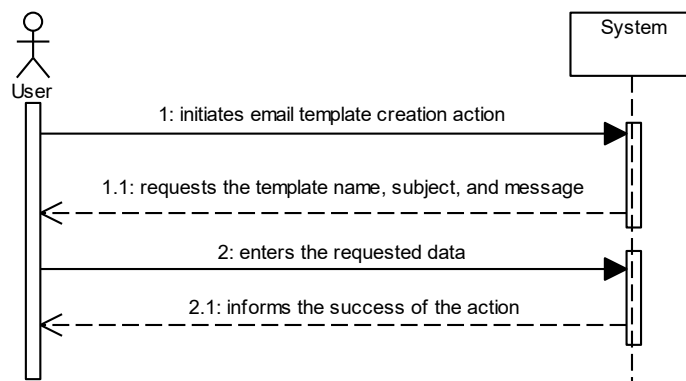


Figure 20 - Create Email Template SSD

5.4.4 View Email Templates

This use case is related to the **FR4**.

To perform this use case, the user must have the required permission and enter the page that allows the visualization of emails templates.

When the user enters the page that allows the creation of email templates, the system shows the list of email templates.

In Figure 21, it is possible to observe the SSD related to this use case.

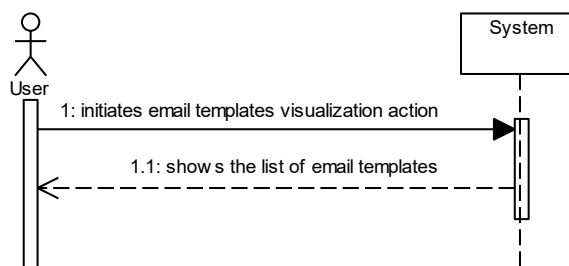


Figure 21 - View Email Templates SSD

5.4.5 Delete Email Template

This use case is related to the **FR5**.

To perform this use case, the user must have the required permission and enter the page that allows the deletion of emails templates.

When the user enters the page that allows the deletion of email templates, the system requests the selection of an email template to delete, which the user selects. After that, the system informs the success of the action.

In Figure 22, it is possible to observe the SSD related to this use case.

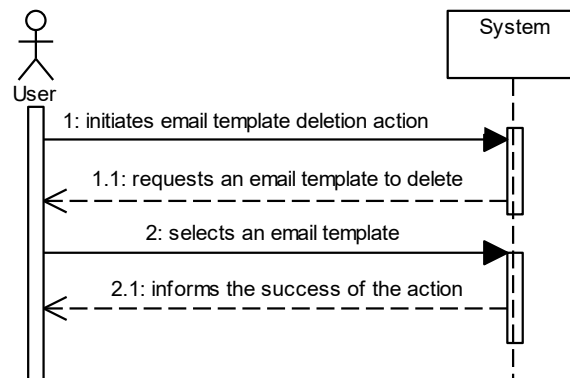


Figure 22 - Delete Email Template SSD

5.4.6 Edit Email Template

This use case is related to the **FR5**.

To perform this use case, the user must have the required permission and enter the page that allows the editing of emails templates.

When the user enters the page that allows the editing of email templates, the system requests the selection of an email template to edit, which the user selects. After that, the system requests the new template name, subject, and message, which the user enters. Finally, the system informs the success of the action.

In Figure 23, it is possible to observe the SSD related to this use case.

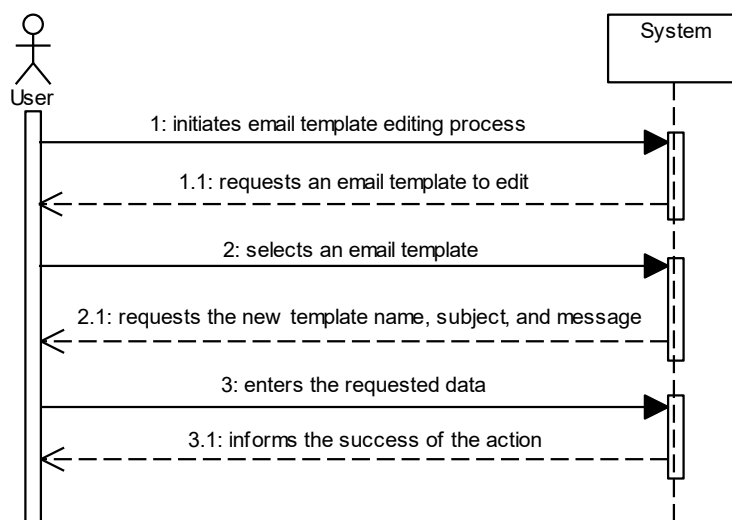


Figure 23 - Edit Email Template SSD

5.4.7 View Email Reports

This use case is related to the **FR6**.

To perform this use case, the user must have the required permission and enter the page that allows the visualization of email reports.

When the user enters the page that allows the visualization of email reports, the system shows the reports related to the sending of emails.

In Figure 24, it is possible to observe the SSD related to this use case.

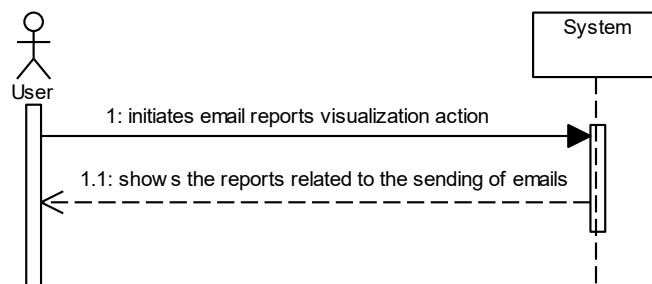


Figure 24 - View Email Reports SSD

5.4.8 View Email Senders

This use case is related to the **FR7**.

To perform this use case, the user must have the required permission and enter the page that allows the visualization of email senders.

When the user enters the page that allows the visualization of email senders, the system shows the list of email senders.

In Figure 25, it is possible to observe the SSD related to this use case.

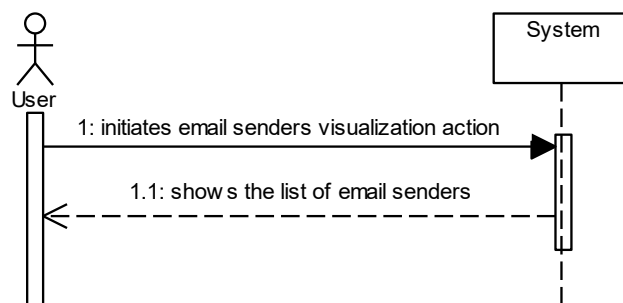


Figure 25 - View Email Senders SSD

5.4.9 Schedule SMS Message

This use case is related to the **FR8**.

To perform this use case, the user must have the required permission and enter the page that allows the sending of quick or advanced SMS messages.

When the user enters the page that allows the sending of quick or advanced SMS messages, in the final step, the system requests the date and time pretended to send the SMS message, which the user enters. Finally, the system requests the action confirmation, which the user confirms, and then the system informs the success of the action.

In Figure 26, it is possible to observe the SSD related to this use case.

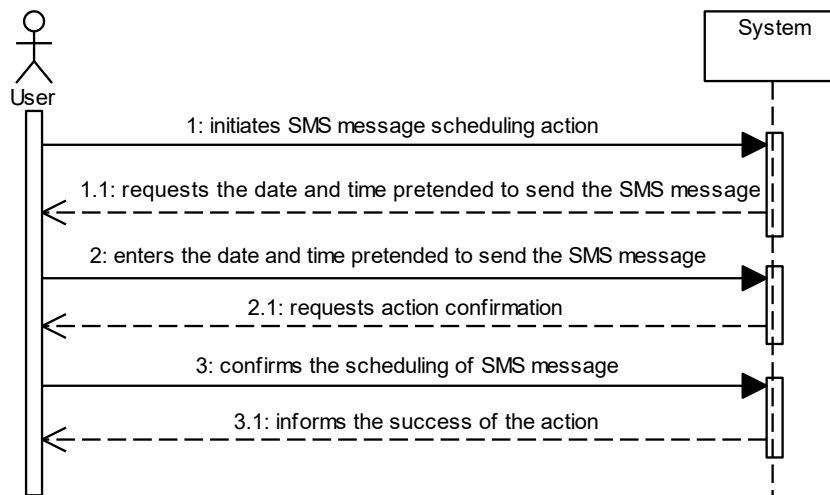


Figure 26 - Schedule SMS Message SSD

5.4.10 Schedule Email

This use case is related to the **FR9**.

To perform this use case, the user must have the required permission and enter the page that allows the sending of quick or advanced Emails.

When the user enters the page that allows the sending of quick or advanced Emails, in the final step, the system requests the date and time pretended to send the Email, which the user enters. Finally, the system requests the action confirmation, which the user confirms, and then the system informs the success of the action.

In Figure 27, it is possible to observe the SSD related to this use case.

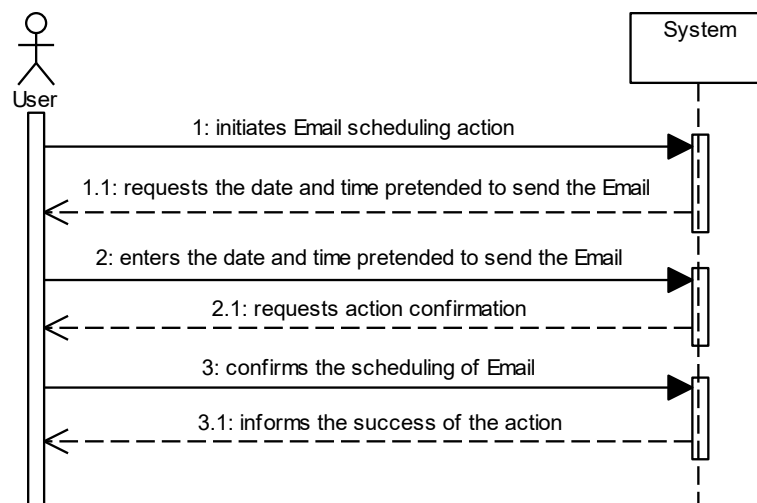


Figure 27 - Schedule Email SSD

5.4.11 View Scheduled SMS Messages

This use case is related to the **FR10**.

To perform this use case, the user must have the required permission and enter the page that allows the visualization of scheduled SMS messages (messages not yet sent).

When the user enters the page that allows the visualization of scheduled SMS messages, the system shows the list of scheduled SMS messages.

In Figure 28, it is possible to observe the SSD related to this use case.

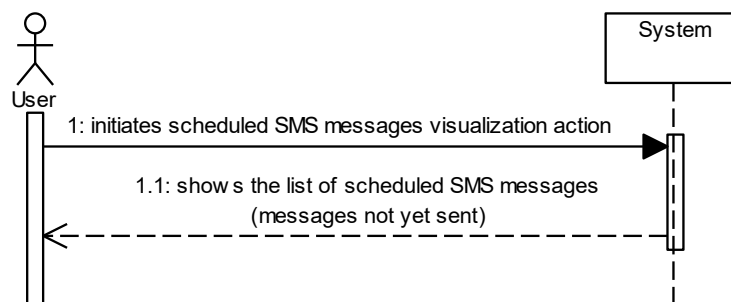


Figure 28 - View Scheduled SMS Messages SSD

5.4.12 View Scheduled Emails

This use case is related to the **FR11**.

To perform this use case, the user must have the required permission and enter the page that allows the visualization of scheduled Emails (emails not yet sent).

When the user enters the page that allows the visualization of scheduled Emails, the system shows the list of scheduled Emails.

In Figure 29, it is possible to observe the SSD related to this use case.

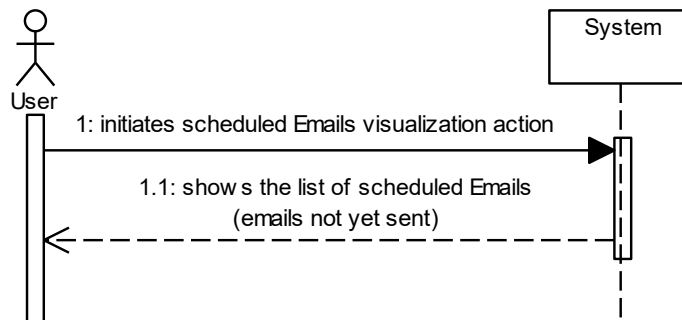


Figure 29 - View Scheduled Emails SSD

5.4.13 Delete Scheduled SMS Message

This use case is related to the **FR12**.

To perform this use case, the user must have the required permission and enter the page that allows the deletion of scheduled SMS messages.

When the user enters the page that allows the deletion of scheduled SMS messages, the system requests the selection of a scheduled SMS message to delete, which the user selects. After that, the system informs the success of the action.

In Figure 30, it is possible to observe the SSD related to this use case.

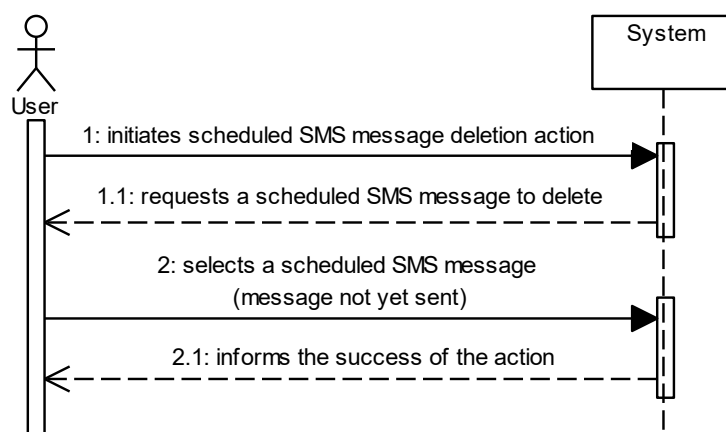


Figure 30 - Delete Scheduled SMS Message SSD

5.4.14 Edit Scheduled SMS Message

This use case is related to the **FR12**.

To perform this use case, the user must have the required permission and enter the page that allows the editing of scheduled SMS messages.

When the user enters the page that allows the editing of scheduled SMS messages, the system requests the selection of a scheduled SMS message to edit, which the user selects. After that, the system requests the new date and time pretended to send the SMS message, which the user enters. Finally, the system informs the success of the action.

In Figure 31, it is possible to observe the SSD related to this use case.

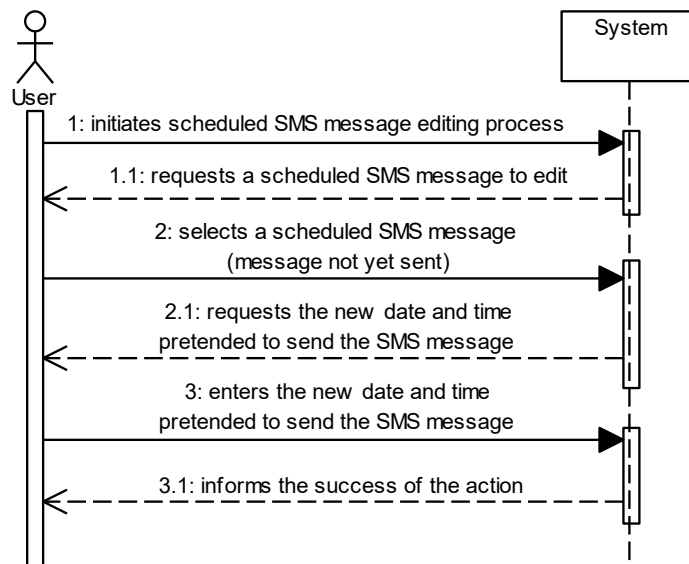


Figure 31 - Edit Scheduled SMS Message SSD

5.4.15 Delete Scheduled Email

This use case is related to the **FR13**.

To perform this use case, the user must have the required permission and enter the page that allows the deletion of scheduled Emails.

When the user enters the page that allows the deletion of scheduled Emails, the system requests the selection of a scheduled Email to delete, which the user selects. After that, the system informs the success of the action.

In Figure 32, it is possible to observe the SSD related to this use case.

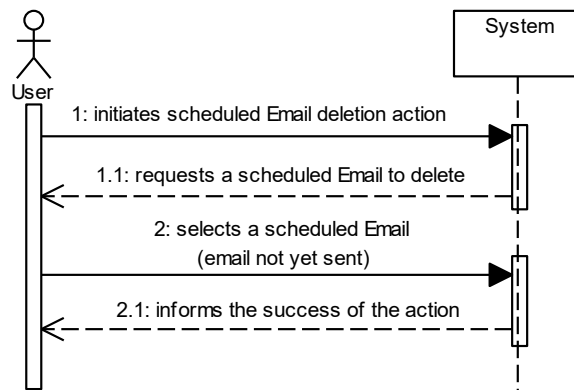


Figure 32 - Delete Scheduled Email SSD

5.4.16 Edit Scheduled Email

This use case is related to the **FR13**.

To perform this use case, the user must have the required permission and enter the page that allows the editing of scheduled Emails.

When the user enters the page that allows the editing of scheduled Emails, the system requests the selection of a scheduled Email to edit, which the user selects. After that, the system requests the new date and time pretended to send the Email, which the user enters. Finally, the system informs the success of the action.

In Figure 33, it is possible to observe the SSD related to this use case.

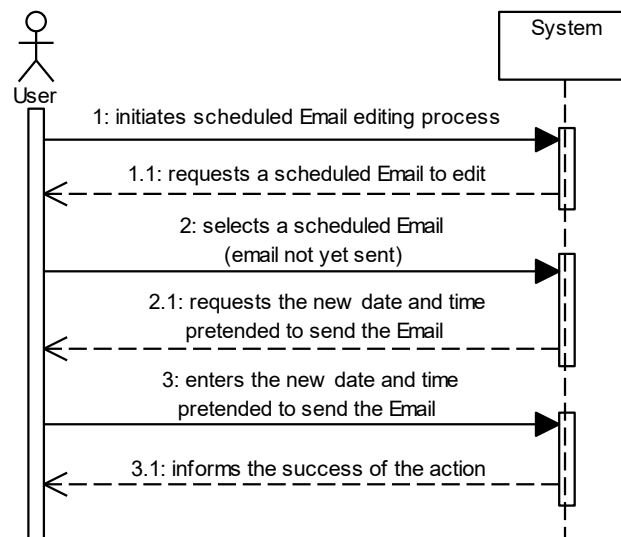


Figure 33 - Edit Scheduled Email SSD

6 Design

This chapter is dedicated to presenting the design of the solution, which includes the technology stack, architecture alternatives, and detailed software architecture. In addition, it depicts the application of the method Attribute-Driven Design (ADD) for architectural design, with the use of many drivers previously identified and many design concepts.

6.1 Technology Stack

The technology stack to be used was already decided when the current QuickMessage application was analyzed.

Since it will be added new features to the existing QuickMessage application, and it will not be created a new application, the technology stack to be used is the same as the current solution.

The backend is developed using Java with Spring Framework, and the frontend using Typescript with Angular. Furthermore, MariaDB is used as the Database Management System (DBMS). Finally, the frontend communicates with the backend using HyperText Transfer Protocol (HTTP), and the request's body and the response body are in JavaScript Object Notation (JSON).

6.2 Overall Software Architecture

The software architecture, like the technology stack, was already decided when the current QuickMessage application was analyzed.

The QuickMessage system consists of three components: the frontend (client), the backend (server), and the database.

The user interacts with the application through the frontend, which provides a graphical interface, that allows the user to log in and perform the many actions/operations that the platform allows. The frontend communicates with the backend through a REST API, and the backend interacts with the database, to obtain information regarding the users and their permissions.

The backend of the QuickMessage system communicates with the Slingshot/Transactional System through a REST API, to be possible to do the Create, Read, Update and Delete (CRUD) operations of the templates, the obtaining of the senders, and the sending of messages.

In Figure 34, the **current overall architecture of the solution** is specified, through a components diagram.

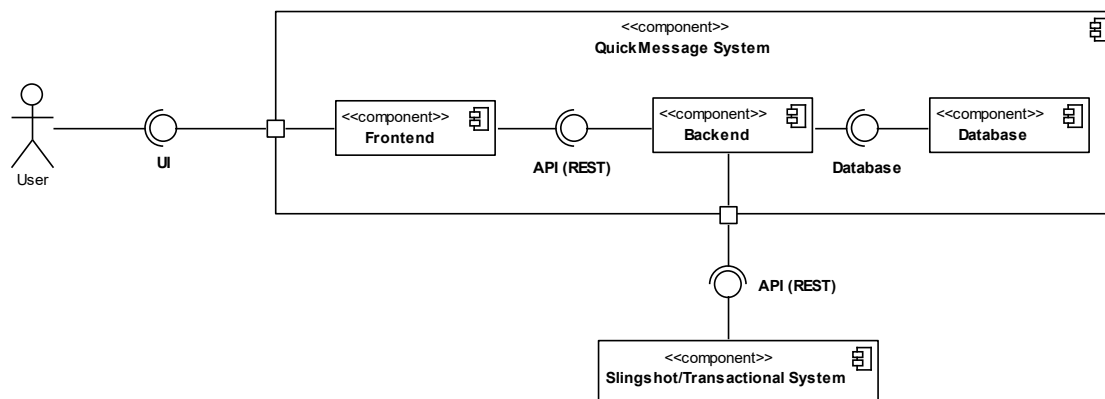


Figure 34 - Components Diagram of the Current Overall Architecture

Since the QuickMessage system is a Web Application, it follows a 3-tier application architecture that consists of a presentation tier (frontend), an application tier (backend), and a data tier (database).

6.3 Alternative Overall Software Architecture

It was decided to implement the new features (Email channel and scheduling of SMS messages and Emails) in the existing QuickMessage application, maintaining the current overall architecture. Although, one alternative would be decomposing the existing QuickMessage monolithic application into microservices (SMS Management, Email Management, and User Management) and implementing the new features in the respective microservice.

This alternative was analyzed by the development team and its implementation was denied. This happened because it would require more development time, it would increase the complexity of the application and it would require more technological resources (this can be seen in more detail in Chapter 4).

However, if it was implemented, the QuickMessage system would consist of five components: the frontend (client), the User Management microservice (server), the SMS Management microservice (server), the Email Management microservice (server), and the database.

The user would interact with the application through the frontend, which would provide a graphical interface, that allows the user to log in and perform the many actions/operations that the platform allows. The frontend would communicate with all the microservices through a REST API.

The User Management microservice would be the only microservice that would interact with the database, to obtain information regarding the users and their permissions.

The SMS Management and Email Management microservices of the QuickMessage system would communicate with the Slingshot/Transactional System through a REST API, to be possible to do the Create, Read, Update and Delete (CRUD) operations of the templates, the obtaining

of the senders, and the sending of messages. Furthermore, these microservices would need to communicate with the User Management microservice through a REST API, to check if the users exist and have the necessary permissions to do the action/operation.

In Figure 35, the **alternative overall architecture of the solution** is specified, through a components diagram.

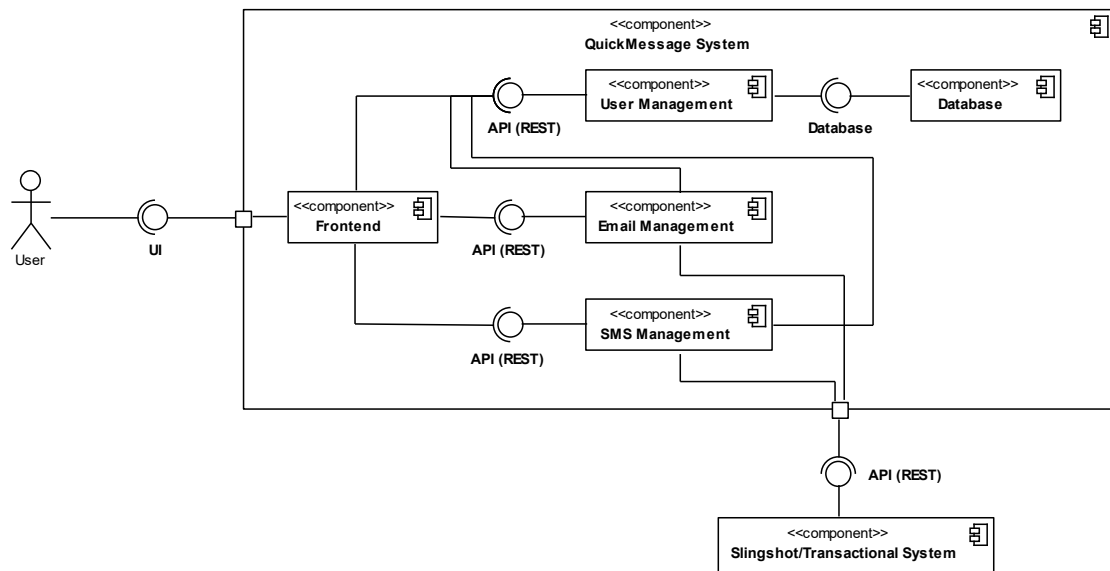


Figure 35 - Components Diagram of the Alternative Overall Architecture

6.4 Attribute-Driven Design

ADD is the method that is used to design the new QuickMessage solution. The use of ADD is necessary to produce a sufficiently detailed design to support the construction of the new features in the system.

The ADD method is an iterative approach for designing software to meet functional requirements as well as non-functional requirements. It uses a recursive decomposition process based on the quality attributes that a system needs to fulfill. At each step, tactics and patterns are chosen to satisfy some requirements [31].

In the **first step, the inputs are reviewed** through the specification of the design purpose, the primary functional requirements (expressed using use cases with short descriptions), the quality attribute scenarios (with the associated use cases), the constraints on the system, and the architectural concerns [31].

In the **second step**, the iterative process can start and **the iteration goal is established by selecting drivers**. The iteration goal typically involves designing to satisfy a subset of the drivers [31].

In the **third step**, **one or more elements of the system are chosen to be refined**. This refinement can be a decomposition into finer-grained elements, a combination of elements into coarser-grained elements, or an improvement of previously identified elements [31].

In the **fourth step**, **one or more design concepts that satisfy the inputs are chosen**. To identify the patterns and tactics, the inputs are used [31].

In the **fifth step**, **architectural elements are instantiated, responsibilities are allocated, and interfaces are defined**. The chosen architectural element is decomposed into children's elements and responsibilities are allocated to the children's elements using use cases [31].

In the **sixth step**, **views are sketched and design decisions are recorded**. The services and properties that are provided and required for each element are defined [31].

Finally, in the **seventh step**, **the current design is analyzed and the iteration goal is reviewed**. The functional requirements and non-functional requirements are verified if they have been addressed or not, and the design decisions are specified in a Kanban Board [31].

After the last step, a new iteration can begin. Each iteration decomposes an architectural element into children's elements. The number of iterations applied is a decision that has to be performed and depends on the abstraction of the architecture [31].

In Figure 36, it is possible to observe the steps of the ADD method.

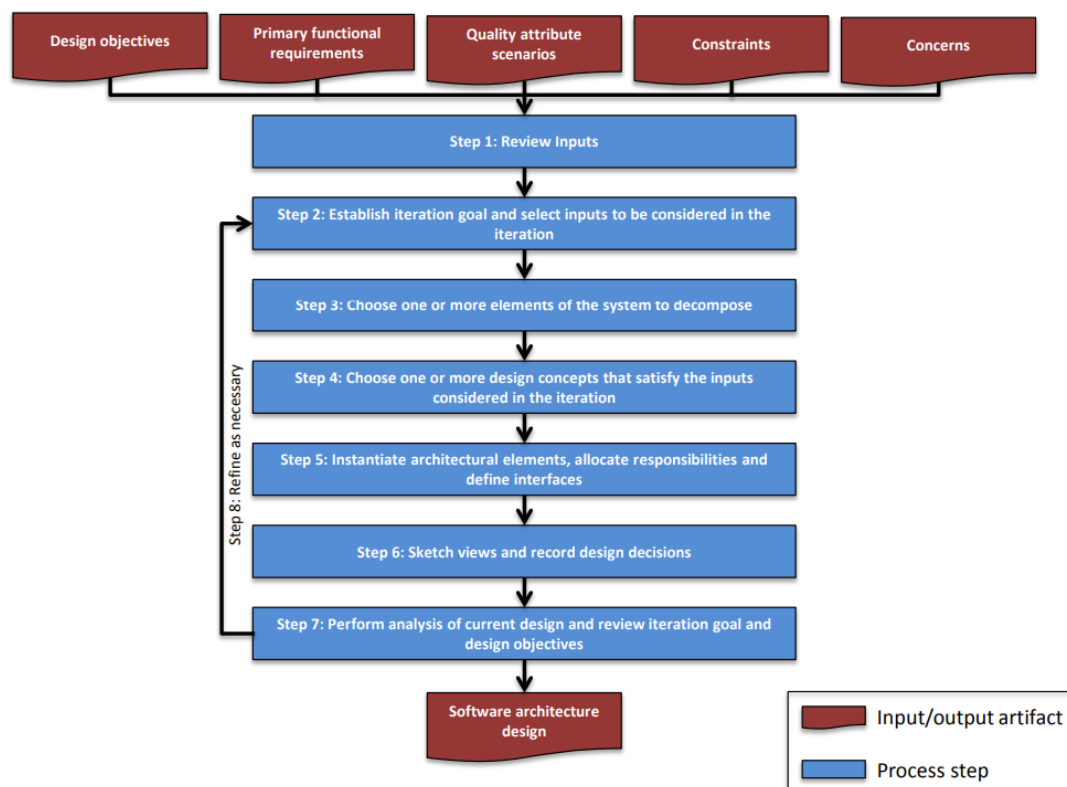


Figure 36 - Steps of the Attribute Driven Design method (adapted) [31]

6.4.1 Iteration 1

Since the functional requirements and non-functional requirements were already specified in Chapter 5, it is possible to initiate the first iteration of the ADD method.

Step 1: Review Inputs

For the first step, it is necessary to review the inputs through the specification of the design purpose, the primary functional requirements, the quality attribute scenarios, the constraints on the system, and the architectural concerns.

The **design purpose** is to develop new features for a brownfield system in a mature domain. It is necessary to produce a sufficiently detailed design to support the construction of the new features in the system.

The **primary functional requirements** are all the use cases from Chapter 5, that are directly related to the sending of Emails. This includes only the following use cases:

- Send Quick Email (UC01)
- Send Advanced Email (UC02)
- Create Email Template (UC03)
- View Email Templates (UC04)
- Delete Email Template (UC05)
- Edit Email Template (UC06)
- View Email Senders (UC08)
- Schedule Email (UC10)

The **quality attribute scenarios**, the **constraints on the system**, and the **architectural concerns** present in Chapter 5 are all included as drivers.

In addition to these drivers, the **existing architecture design** is also included as a driver, since this is a brownfield development.

Step 2: Establish Iteration Goal

The iteration goal is to support the primary functionality by identifying the elements that support the primary functionality.

Step 3: Choose Elements of the System to Refine

This is a brownfield system in a mature domain, with an already specified overall software architecture. It can be decomposed/refined the modules located in the different layers of the 3-tier application architecture.

The support of functionality in this system requires the collaboration of components associated with modules that are in the different layers (presentation layer, business layer, and data layer).

Step 4: Choose Design Concepts that Satisfy the Inputs

The first design concept is to **decompose the domain objects across the layers**. It is necessary to identify layer-specific modules with an explicit interface, to ensure that modules that support all the functionalities are identified. This task is only performed for the primary use cases.

The second design concept is to **use the Transactional API, provided by E-goi, to send transactional messages**, which satisfies **CON-1**.

The third design concept is to **use Java and the Spring Framework for the backend development**, which satisfies **CON-2**.

The fourth design concept is to **use Typescript and Angular for the frontend development**, which satisfies **CON-3**.

The fifth design concept is to **provide a REST API in the server for client-side communication**, which satisfies **CON-4**.

The sixth design concept is to **design the system to be easily scalable and easy to maintain, by adopting good practices of object-oriented software analysis and design**. These concepts satisfy **CRN-1** and **CRN-2**.

Step 5: Instantiate Elements, Allocate Responsibilities and Define Interfaces

In this step, elements of the Presentation and Business Layer are instantiated (the data layer has no elements), their responsibilities are allocated, and then their interfaces are defined.

In Table 16, it is possible to observe the **elements and responsibilities of the Presentation Layer**.

Table 16 - Elements and Responsibilities of the Presentation Layer

Element	Responsibility
CreateEmailTemplateView	Menu for the creation of Email Templates.
ViewEmailTemplatesView	A menu that displays the list of Email Templates.
DeleteEmailTemplateView	Menu for the delete of an Email Template.
EditEmailTemplateView	Menu for the edit of an Email Template.
SendQuickEmailView	A menu that allows the sending of an Email without a CSV file.
SendAdvancedEmailView	A menu that allows the sending of an Email with a CSV file.
ViewEmailSendersView	A menu that displays the list of Email Senders.
EmailTemplatesService	Service responsible for the communication with the server related to Email Templates.
SendEmailService	Service responsible for the communication with the server related to the sending of Emails.
SendersService	Service responsible for the communication with the server related to Email Senders.

In Table 17, it is possible to observe the **elements and responsibilities of the Business Layer**.

Table 17 - Elements and Responsibilities of the Business Layer

Element	Responsibility
EmailTemplatesController	A controller that contains the business logic regarding the Email Templates.
SendEmailController	A controller that contains the business logic regarding the sending of Emails.
SendersController	A controller that contains the business logic regarding the Email Senders.
DomainModel	Domain classes from the domain model.
EmailTemplatesDispatcher	Service responsible for the communication with the Transactional API related to Email Templates.
EmailDispatcher	Service responsible for the communication with the Transactional API related to the sending of Emails.
SendersDispatcher	Service responsible for the communication with the Transactional API related to the Email Senders.

In Table 18, it is possible to observe the **methods of the EmailTemplatesService**.

Table 18 - Methods of the EmailTemplatesService

Method	Description
void createTemplate(EmailTemplate)	Requests the creation of an Email Template to the server.
EmailTemplate[] getTemplates()	Requests the list of all Email Templates to the server.
void updateTemplate(id, EmailTemplate)	Requests the update of an Email Template to the server.
void deleteTemplate(id)	Request the deletion of an Email Template to the server.

In Table 19, it is possible to observe the **methods of the SendEmailService**.

Table 19 - Methods of the SendEmailService

Method	Description
void sendQuickEmail(Email)	Requests the sending of a Quick Email (may not be a single Email, since the list of recipients is specified) to the server.
void sendAdvancedEmail(Email)	Requests the sending of an Advanced Email (may not be a single Email, since the list of recipients is specified) to the server.

In Table 20, it is possible to observe the **methods of the SendersService**.

Table 20 - Methods of the SendersService

Method	Description
Sender[] getSendersEmail()	Requests the list of all Email Senders to the server.

In Table 21, it is possible to observe the **methods of the EmailTemplatesController**.

Table 21 - Methods of the EmailTemplatesController

Method	Description
void createEmailTemplate(EmailTemplate)	Receives a request to create an Email Template. This method is exposed by the controller.
List<EmailTemplate> getAllEmailTemplates()	Receives a request to obtain all the Email Templates. This method is exposed by the controller.
void updateEmailTemplate(id, EmailTemplate)	Receives a request to update an Email Template. This method is exposed by the controller interface.
void deleteEmailTemplate(id)	Receives a request to delete an Email Template. This method is exposed by the controller.

In Table 22, it is possible to observe the **methods of the SendEmailController**.

Table 22 - Methods of the SendEmailController

Method	Description
void sendQuickEmailMessage(Email)	Receives a request to send a Quick Email. This method is exposed by the controller.
void sendAdvancedEmailMessage(Email)	Receives a request to send an Advanced Email. This method is exposed by the controller.

In Table 23, it is possible to observe the **methods of the SendersController**.

Table 23 - Methods of the SendersController

Method	Description
List<Sender> getEmailSenders()	Receives a request to obtain all the Email Senders. This method is exposed by the controller interface.

In Table 24, it is possible to observe the **methods of the EmailTemplatesDispatcher**.

Table 24 - Methods of the EmailTemplatesDispatcher

Method	Description
void createTemplate(apiKey, EmailTemplate)	Requests the creation of an Email Template to the Transactional API.
List<EmailTemplate> listAllTemplates(apiKey)	Requests the list of all Email Templates to the Transactional API.
void putTemplate(apiKey, id, EmailTemplate)	Requests the update of an Email Template to the Transactional API.
void deleteTemplate(apiKey, id)	Request the deletion of an Email Template to the Transactional API.

In Table 25, it is possible to observe the **methods of the EmailDispatcher**.

Table 25 - Methods of the EmailDispatcher

Method	Description
void sendEmail(apiKey, List<Email>)	Requests the sending of a list of Emails to the Transactional API. If it is a Quick Email, the list is a single Email with all recipients. If it is an Advanced Email, the list is a group of Emails for each recipient.

In Table 26, it is possible to observe the **methods of the SendersDispatcher**.

Table 26 - Methods of the SendersDispatcher

Method	Description
List<Sender> listAvailableEmailSenders(apiKey)	Requests the list of all Email Senders to the Transactional API.

Step 6: Sketch Views and Record Design Decisions

In this step, views and recorded design decisions are sketched according to the elements that were instantiated in the previous step.

In Figure 37, it is possible to observe the **primary use cases on layers**, through a **components diagram**.

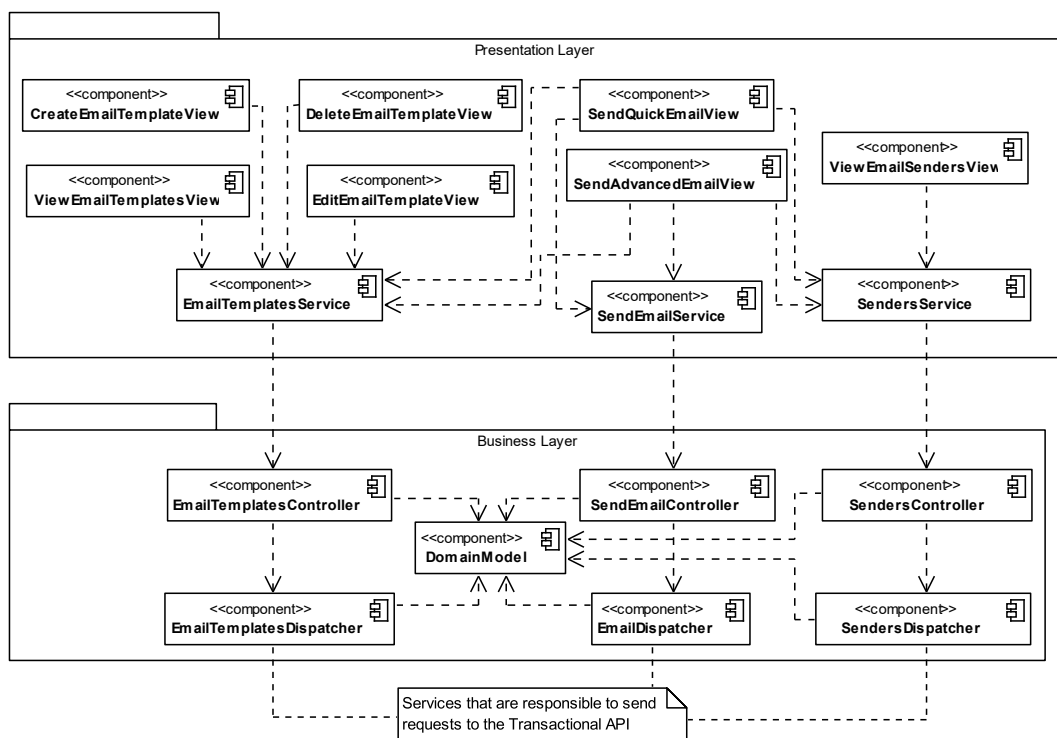


Figure 37 – Components Diagram of the Primary Use Cases on Layers

By looking at Figure 37, it is possible to observe that the Views related to Email Templates have a dependency on the EmailTemplatesService since it is this service that has the responsibility to create, update, list, and delete Email Templates.

The Views related to sending Emails have a dependency on the SendEmailService since it is this service that has the responsibility to send Quick and Advanced Emails. Furthermore, the Views also have a dependency on the EmailTemplatesService and SendersService since they need the list of Email Templates and Email Senders when sending an Email.

The View related to Email Senders has a dependency on the SendersService since it is this service that has the responsibility to list Email Senders.

The Controller related to Email Templates has a dependency on the EmailTemplatesDispatcher since it is this dispatcher that has the responsibility to communicate with the Transactional API to create, update, list, and delete Email Templates.

The Controller related to sending Emails has a dependency on the EmailDispatcher since it is this dispatcher that has the responsibility to communicate with the Transactional API to send Quick and Advanced Emails.

The Controller related to Email Senders has a dependency on the SendersDispatcher since it is this dispatcher that has the responsibility to communicate with the Transactional API to list Email Senders.

In Figure 38, it is possible to observe the **Sequence Diagram (SD) of the UC01 called Send Quick Email**. The optional scheduling of the Email isn't represented, however, before confirming the sending of the Quick Email, the user can choose to send now or schedule the Email.

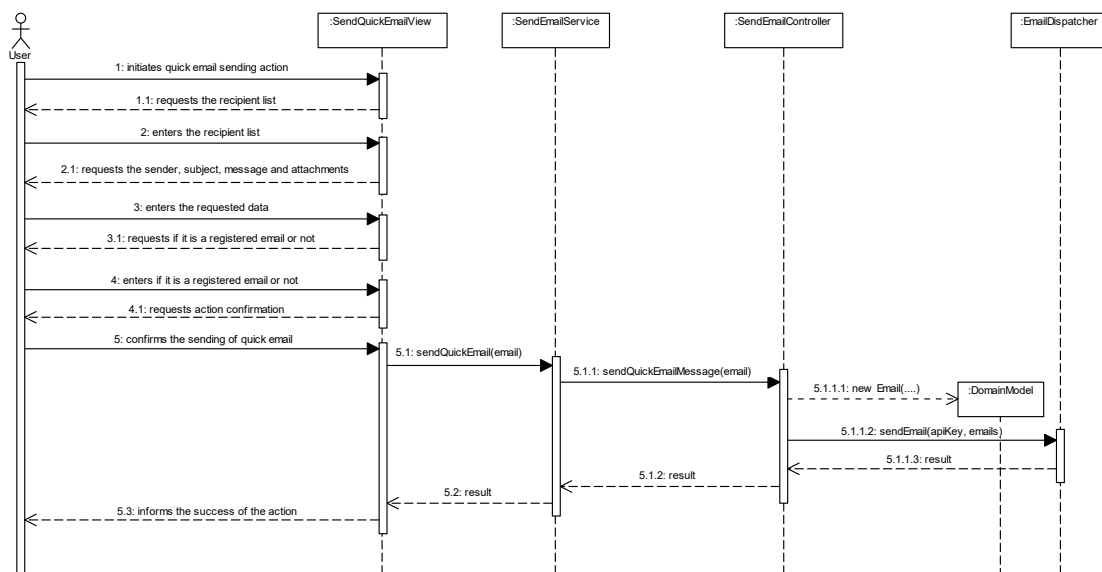


Figure 38 - Send Quick Email SD

In Figure 39, it is possible to observe the **SD of the UC02 called Send Advanced Email**. The optional scheduling of the Email isn't represented, however, before confirming the sending of the Advanced Email, the user can choose to send now or schedule the Email.

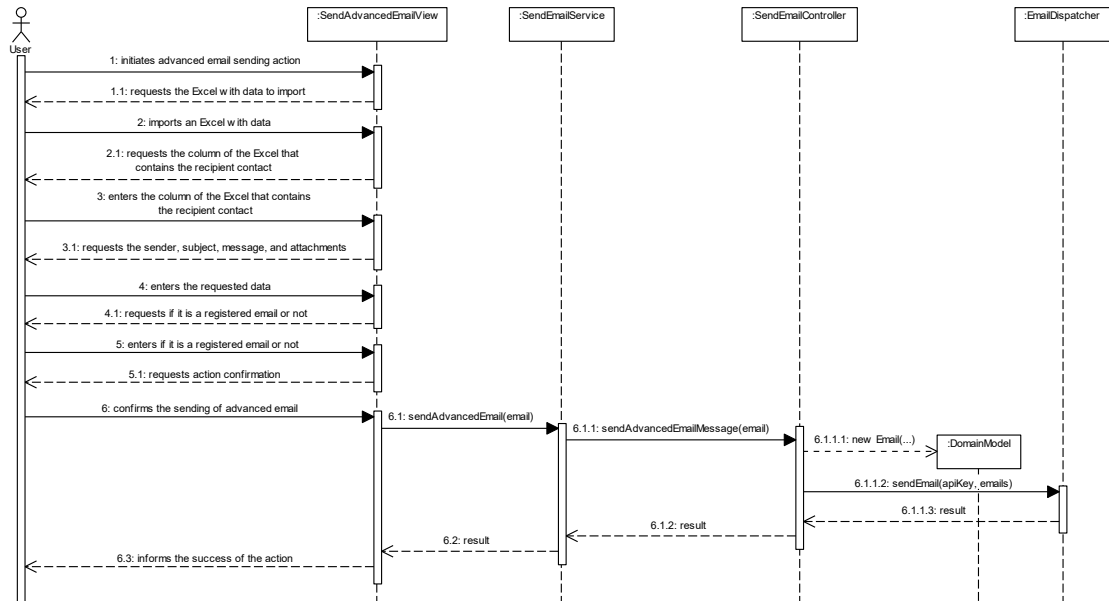


Figure 39 - Send Advanced Email SD

In Figure 40, it is possible to observe the **SD of the UC03 called Create Email Template**.

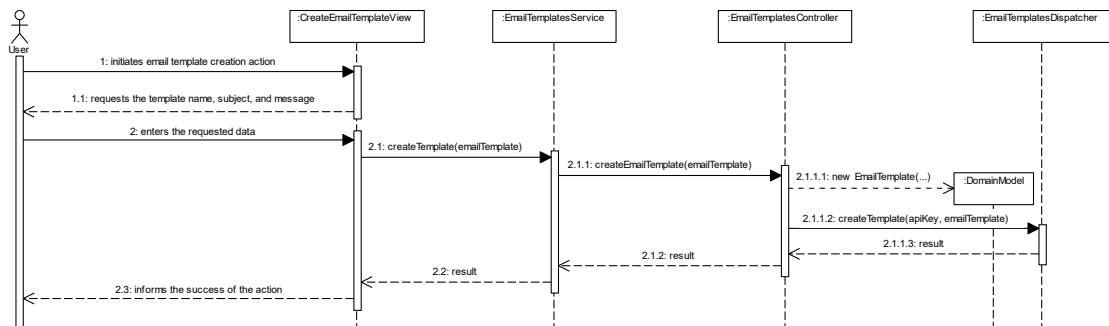


Figure 40 - Create Email Template SD

In Figure 41, it is possible to observe the **SD of the UC04 called View Email Templates**.

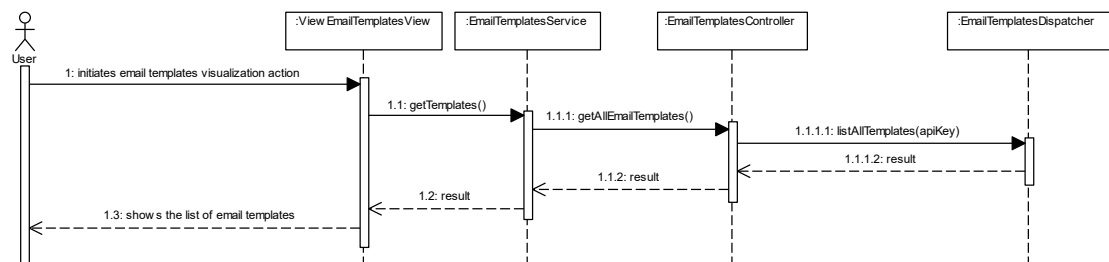


Figure 41 - View Email Templates SD

In Figure 42, it is possible to observe the **SD of the UC05 called Delete Email Template**.

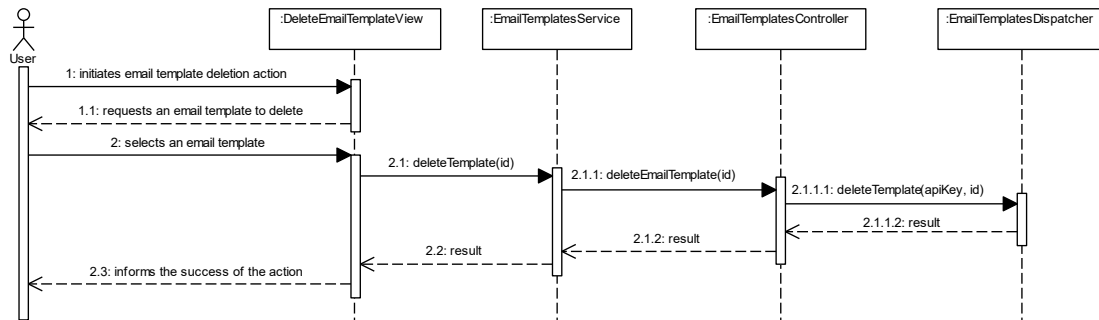


Figure 42 - Delete Email Template SD

In Figure 43, it is possible to observe the **SD of the UC06 called Edit Email Template**.

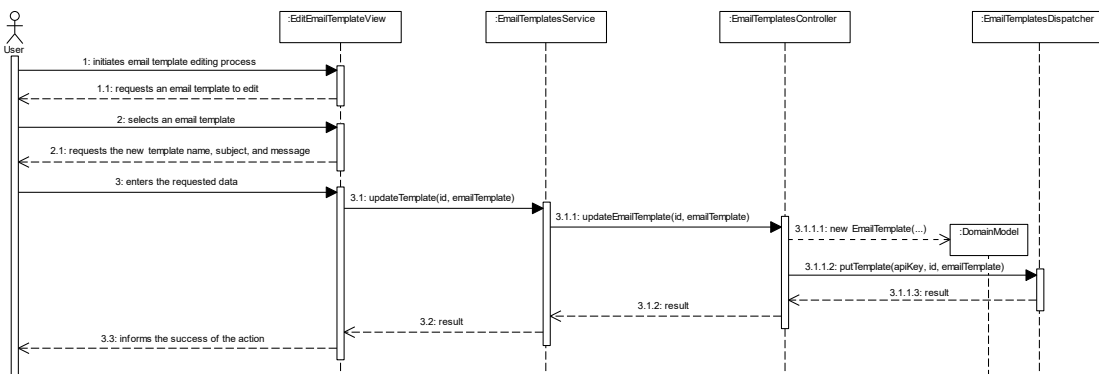


Figure 43 - Edit Email Template SD

In Figure 44, it is possible to observe the **SD of the UC08 called View Email Senders**.

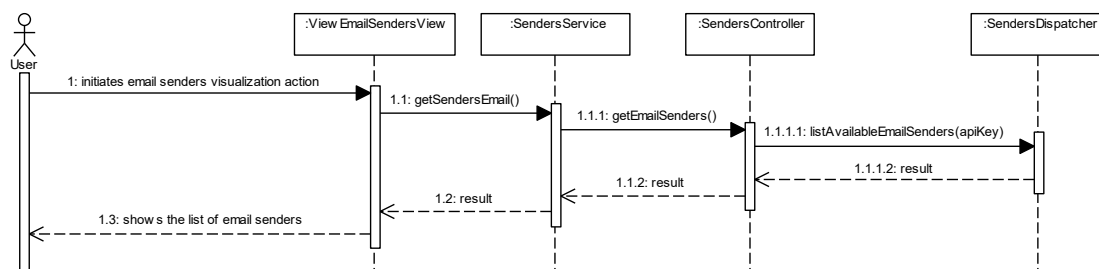


Figure 44 - View Email Senders SD

Step 7: Analyze Current Design and Review Iteration Goal

In Table 27, it is possible to observe the **Design Kanban Board for the Iteration 1**. It specifies which use cases, quality attributes, constraints, and concerns have been addressed, partially addressed, or not addressed.

Table 27 - Design Kanban Board - Iteration 1

Not Addressed	Partially Addressed	Addressed	Design Decisions
		UC01, UC02, UC03, UC04, UC05, UC06, UC08, UC10	Modules across the layers and preliminary interfaces to support these use cases have been identified.
QA-1, QA-2, QA-3, QA-4, QA-5			No relevant decisions were made.
		CON-1	The Transactional API is used to send transactional messages.
		CON-2	Java and the Spring Framework are used for backend development.
		CON-3	Typescript and Angular are used for frontend development.
		CON-4	The server provides a REST API for client-side communication.
		CRN-1, CRN-2	The system is designed to be easily scalable and easy to maintain, by adopting good practices of object-oriented software analysis and design.

By looking at Table 27, it is possible to observe that all the use cases, constraints, and concerns have been addressed. However, none of the quality attributes have been addressed, which means another iteration is needed to support the quality attribute scenarios.

6.4.2 Iteration 2

Since the quality attribute scenarios were not addressed in the previous iteration, it is necessary to initiate a second iteration of the ADD method.

Step 1: Review Inputs

It is not necessary to review the inputs since they were already reviewed in the previous iteration.

Step 2: Establish Iteration Goal

The iteration goal is to support the quality attribute scenarios.

The following quality attributes must be supported:

- The new functional requirements must be secured by a new set of permissions, which must also be possible to add to the user's roles (QA-1).
- The application architecture and code must be documented (QA-2).

- All features of the system must have an adequate response time, of less than 1 second (QA-3).
- The system must deal with all errors and present the unexpected errors to the user with a clear message about the error (QA-4).
- The interaction between users and the system must be simple, intuitive and fully adapted to the action in question (QA-5).

Step 3: Choose Elements of the System to Refine

It can be decomposed/refined the modules in the layers that are related to QA-1, QA-2, QA-3, QA-4, and QA-5.

For the QA-1 quality attribute scenario, the decomposition of both the Presentation Layer and Business Layer is needed. For the QA-2 quality attribute scenario, the decomposition of only the Business Layer is needed. For the QA-3, and QA-4 quality attribute scenarios, the decomposition of only the Presentation Layer is needed.

Step 4: Choose Design Concepts that Satisfy the Inputs

The first design concept is to **secure the new functional requirements through a set of permissions and make it possible to add these permissions to the user's roles**, which satisfies **QA-1**. In the presentation layer, a component is needed, that does not allow the user to access the feature if they do not have the required permissions. In the business layer, a component is needed, that does not allow a request if the user does not have the required permissions. The use cases related to creating roles and editing roles must be adapted to allow adding the new permissions.

The second design concept is to **document the application architecture and code**, which satisfies **QA-2**. In the business layer, the use of Swagger Documentation in the Controllers is needed.

The third design concept is to have an **adequate response time, of less than 1 second, in all the features of the system**, which satisfies **QA-3**. Most features in the system have a response time of less than 1 second, the only feature that may have a response time superior is the sending of SMS messages and Emails since it is done synchronously. The sending of SMS messages and Emails should be done asynchronously, so the user doesn't wait for all the messages to be sent. In the presentation layer, while the user waits for the response of the server, it should be presented a Spinner component that shows that the server is processing the request.

The fourth design concept is to have the **system deal with all errors and present the unexpected errors to the user with a clear message about the error**, which satisfied **QA-4**. In the presentation layer, a component is needed, that presents notifications to the user when an error occurs, when the operation is being done/processed by the server and when the operation ends successfully.

The fifth design concept is to have an **interaction between the users and the system simple, intuitive, and fully adapted to the action in question**, which satisfies **QA-5**. After the development of the solution, a usability questionnaire will be used to evaluate the usability of the QuickMessage application.

Step 5: Instantiate Elements, Allocate Responsibilities and Define Interfaces

In this step, the new elements of the Presentation and Business Layer are instantiated, and their responsibilities are allocated.

In Table 28, it is possible to observe the **new elements and responsibilities of the Presentation Layer**.

Table 28 - New Elements and Responsibilities of the Presentation Layer

Element	Responsibility
PermissionsService	Service responsible for the communication with the server related to the User Permissions. It checks if the user has permission to access the features.
Spinner	The element is responsible to show the user that the server is processing the request.
Notifications	The element is responsible to present to the user a notification when an error occurs, when the operation is being done/processed by the server and when the operation ends successfully.

In Table 29, it is possible to observe the **new elements and responsibilities of the Business Layer**.

Table 29 - New Elements and Responsibilities of the Business Layer

Element	Responsibility
AuthGuard	Service responsible to secure the Controller from unauthorized access. If the user doesn't have the required permissions, the request fails.
SwaggerDocumentation	Service responsible to generate Swagger documentation related to the Controllers.

Step 6: Sketch Views and Record Design Decisions

In this step, views and recorded design decisions are sketched according to the elements that were instantiated in the previous step.

In Figure 45, it is possible to observe the **new elements' dependencies, through a components diagram**.

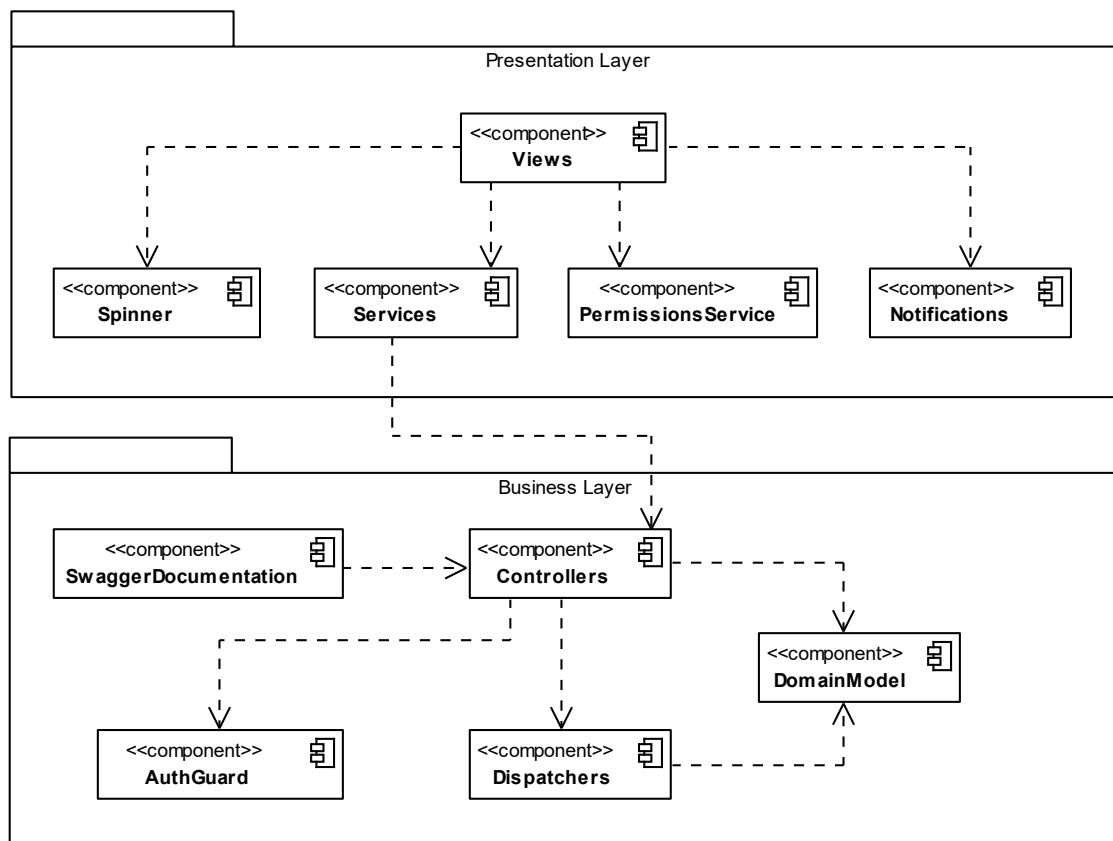


Figure 45 - Components Diagram of the New Elements' Dependencies

By looking at Figure 45, it is possible to observe that the Views of the Presentation Layer have new dependencies. Firstly, they have a dependency on the Spinner component, since it is this element that shows, in the user interface, that the server is processing the request. Secondly, they have a dependency on the PermissionsService, since it is this element that checks if the user has permission to access the feature in the interface. Finally, they have a dependency on the Notifications, since it is this element that presents to the user if an error occurred, if the operation is being done/processed, or has just ended successfully.

The Controllers have also a new dependency on the AuthGuard since it is this element that secures the Controllers from unauthorized access. In addition, the new element SwaggerDocumentation has a dependency on the Controllers since it generates documentation related to them.

Step 7: Analyze Current Design and Review Iteration Goal

In Table 30, it is possible to observe the **Design Kanban Board for the Iteration 2**. It specifies which use cases, quality attributes, constraints, and concerns have been addressed, partially addressed, or not addressed.

Table 30 - Design Kanban Board - Iteration 2

Not Addressed	Partially Addressed	Addressed	Design Decisions
		UC01, UC02, UC03, UC04, UC05, UC06, UC08, UC10, CON-1, CON-2, CON-3, CON-4, CRN-1, CRN-2	No relevant decisions were made.
		QA-1	Functional requirements are secured through a set of permissions and it is possible to add these permissions to the user.
		QA-2	The application architecture and code are documented.
		QA-3	All the features of the system have an adequate response time, of less than 1 second.
		QA-4	The system deals with all errors and presents the unexpected errors to the user with a clear message about the error.
		QA-5	The interaction between users and the system is simple, intuitive, and fully adapted to the action in question.

By looking at Table 30, it is possible to observe that all the use cases, constraints, concerns, and quality attributes have been addressed. This means that there is no need for more iterations of ADD.

7 Implementation

This chapter is dedicated to providing a detailed overview of the implementation. It demonstrates how the framing and implementation of the solution were done according to the analysis and design of the solution defined in Chapters 5 and 6.

7.1 Development Methodology

For a team to be able to work together, there must be a development methodology common to all elements.

E-goi uses the Git tool, as a control system of versions for all their projects and products, and so this tool was also used in the development of the QuickMessage application. The tool that allows the use of Git at E-goi is the GitLab application installed on an internal server.

For the development of the solution, it was created three branches in Git. A branch where it was done the development of the many features (may contain unfinished code), a staging branch where the application would be deployed to be tested (contains finished code), and a production branch where the application would be deployed to the end-user (contains finished and tested code).

The flow consists of the implementation of features in the development branch, followed by testing the features in the staging branch, and making the features available in the production branch.

7.2 Solution Development

Following the analysis and design defined in Chapters 5 and 6, the solution was developed. This solution was developed using Java and Spring Framework for the backend development, and using Typescript and Angular for the frontend development. Furthermore, it was provided a REST API for client-side communication by the server, and this server used the Transactional/Slingshot API to send transactional messages, and handle most of the features. Finally, the system was developed by adopting good practices of object-oriented software analysis and design, to be easily scalable and easy to maintain.

The solution development can be divided into **nine main groups**: Email Senders, Email Templates, Emails Sending, Email Reports, Scheduling, Permissions, Documentation, Response Time, and Notifications.

7.2.1 Email Senders

This group is related only to the feature of visualizing Email Senders in the QuickMessage application. In the solution, it was only implemented this feature, and if it is necessary to add, edit, or delete Email Senders, the E-goi application needs to be used, since the Email Senders are connected to an E-goi account.

The feature of visualizing Email Senders was implemented mainly by creating a new Angular component called **EmailSendersComponent** and using it inside the area where it was possible to visualize the SMS Senders. For this, it was created two tabs, one where the user could visualize the SMS Senders, and the other one where the user could visualize the Email Senders.

In this component, it is possible to visualize the Email Senders in a table with pagination, by requesting the server to list the Email Senders. It is possible to visualize the id, name, email, and status of the sender.

In Figure 46, it is possible to observe the **user interface of the feature of visualizing Email Senders**.

SMS		EMAIL	
Id	Name	Email	Status
1	sysadmin+96@e-goi.com	sysadmin+96@e-goi.com	✓
4	Pedro Pinto	ppinto@hostingatlantic.com	✓
5	Pedro Pinto	ppinto@techinhosting.com	✓
6	Pedro Pinto	ppinto@internethosted.com	✓
7	test@actz.xyz	test@actz.xyz	✗
8	test@base64.fr	test@base64.fr	✗
9	test@00tbox.com	test@00tbox.com	✗

Figure 46 - User Interface of the Feature of Visualizing Email Senders

7.2.2 Email Templates

This group is related to the features of creating, visualizing, deleting, and editing Email Templates in the QuickMessage application.

The feature of visualizing Email Templates was implemented mainly by creating a new Angular component called **EmailTemplatesComponent** and using it inside the area where it was possible to visualize the SMS Templates. For this, it was created two tabs, one where the user

could visualize the SMS Templates, and the other one where the user could visualize the Email Templates.

In this component, it is possible to visualize the Email Templates in a table with pagination, by requesting the server to list the Email Templates. It is possible to visualize the id, name, and subject of the email template. Furthermore, it is possible to create, view (visualize the HTML content of the Email), delete, and edit Email Templates by clicking on the buttons related to that action.

In Figure 47, it is possible to observe the **user interface of the feature of visualizing Email Templates**.

The screenshot shows a web application interface for managing templates. The main heading is 'Templates'. Below it, there are two tabs: 'SMS' and 'EMAIL'. The 'EMAIL' tab is active, displaying a table with the following data:

Id	Name	Subject	Actions
709	CodeSnippet - Personal	CodeSnippets	View Edit Delete
710	Bernardo	Bernardo Friday	View Edit Delete
711	Teste 1	Subject	View Edit Delete
713	HtmlEditorModel1	HtmlEditorSubject1	View Edit Delete
714	carlosmodel	assuntocarlos	View Edit Delete
715	CodeSnippet - Personal	CodeSnippets	View Edit Delete
716	Ola Carlos	Ola	View Edit Delete

At the bottom of the table, there is a pagination control showing 'Items per page: 10' and '1 - 10 of 97'.

Figure 47 - User Interface of the Feature of Visualizing Email Templates

The feature of creating Email Templates was implemented mainly by creating a new Angular component called **CreateEmailTemplateComponent** and using it inside the modal that appears when the user clicks the button to add a new Email Template.

In this component, it is possible to create a new Email Template by filling in the template name, subject, and email body (using an HTML editor). Furthermore, it is also possible to add variables to the template, that can be replaced later when sending the Email.

When all fields are filled, the user can click on the button to complete the action and request the server to create the Email Template.

In Figure 48, it is possible to observe the **user interface of the feature of creating a new Email Template**.

Create Email Template

Template Name *

Subject *

Enter text here...

Add New Variable +

Close Create

Figure 48 - User Interface of the Feature of Creating Email Templates

The feature of deleting Email Templates was implemented mainly by creating a new Angular component called **DeleteEmailTemplateComponent** and using it inside the modal that appears when the user clicks the button to delete an existing Email Template.

In this component, it is possible to confirm the deletion of an Email Template and request the server to delete the Email Template.

In Figure 49, it is possible to observe the **user interface of the feature of deleting an Email Template**.

Delete Email Template?

Are you sure you wish to delete the Email template with Id **709**?

Cancel

Yes, delete template

Figure 49 - User Interface of the Feature of Deleting Email Templates

The feature of editing Email Templates was implemented mainly by creating a new Angular component called **UpdateEmailTemplateComponent** and using it inside the modal that appears when the user clicks the button to edit an existing Email Template.

In this component, it is possible to edit an Email Template by filling in the template name, subject, and email body (using an HTML editor). Furthermore, it is also possible to add variables to the template, that can be replaced later when sending the Email.

When all fields are filled, the user can click on the button to complete the action and request the server to update the Email Template.

In Figure 50, it is possible to observe the **user interface of the feature of editing an Email Template**.

Update Email Template

Template Name *
CodeSnippet - Personal

Subject *
CodeSnippets



%DATE%

Informamos que o BNP Paribas Personal Finance, S.A., Sucursal em Portugal, entidade responsável pela gestão do cartão de crédito Banco CTT, procedeu à alteração da redação de algumas das cláusulas que integram as condições gerais do seu contrato de modo a integrar os seguintes pontos:

- Valores máximos de comissões que poderão ser cobrados no âmbito dos pagamentos nas modalidades de Pagamentos Especiais (pagamentos na sua loja CTT e Banco CTT, com e sem juros), e

Add New Variable +

Close **Update**

Figure 50 - User Interface of the Feature of Editing Email Templates

7.2.3 Emails Sending

This group is related to the features of sending a Quick Email (without a csv file) and sending an Advanced Email (with a csv file) in the QuickMessage application.

The feature of sending a Quick Email was implemented mainly by creating a new Angular component called **QuickEmailFormComponent**. For this, it was added, to the sidebar of the application, a new item related to the sending of Quick Emails and Advanced Emails. Two tabs were created, one where the user could send Quick Emails, and the other one where the user could send Advanced Emails.

In this component, it is possible to send a Quick Email by following three steps.

The **first step** is to fill out the recipients of the Email, which will be a list of emails of the people to who the user wants to send an Email. In Figure 51, it is possible to observe the **user interface of the first step of the feature of sending a Quick Email**.

The screenshot shows the 'Send Email' interface with a sidebar on the left containing 'Dashboard', 'Quick SMS', 'Quick Email', and a 'MANAGEMENT' section with 'Instances', 'Users', 'Roles', 'Templates', and 'Senders'. The main area is titled 'Send Email' and has two tabs: 'WITHOUT CSV FILE' (selected) and 'WITH CSV FILE'. A progress bar at the top of the main area shows three steps: '1 Fill out your Destinatary' (active), '2 Fill out your Email', and '3 Done'. Below the progress bar is a large text input field labeled 'Destinatary *' containing the email 'jbernardo@e-goi.com'. A small note at the bottom left of the field says 'One destinatary per line'. A 'Next >' button is at the bottom right. The top header bar includes a back arrow, 'E-GOI Tests', a 'Quick Message' button with a right arrow, and a user profile icon.

Figure 51 - User Interface of the First Step of the Feature of Sending Quick Email

The **second step** is to select the sender of the Email and fill out the subject and the email body (using an HTML editor). Optionally the user can fill both the subject and the email body by selecting an existing Email Template. Furthermore, in this step, the user can also add attachments to the Email, to a certain limit of megabytes. In Figure 52, it is possible to observe the **user interface of the second step of the feature of sending a Quick Email**.

The screenshot shows the 'Send Email' interface at the second step. The progress bar now shows '1 Fill out your Destinatary' as completed and '2 Fill out your Email' as active. The 'WITHOUT CSV FILE' tab is still selected. The 'Destinatary' field is now a dropdown menu showing 'sysadmin+96@e-goi.com'. Below it is an 'Email' dropdown menu. Underneath is an 'Attachments' section with a blue circular icon containing a document symbol and the text 'You have a limit of 5 MB.'. At the bottom, there are '< Previous' and 'Next >' buttons. The top header bar remains the same as in Figure 51.

Figure 52 - User Interface of the Second Step of the Feature of Sending Quick Email

In Figure 53, it is possible to observe the **Email section that allows to select Email Templates and fill in the subject and the email body**.

Figure 53 - Email Section while Sending Quick Email

The **third step** is the last and consists of visualizing a summary with the number of recipients, the sender, the subject, and the email body. Furthermore, it allows the user to choose between sending now the Email or scheduling it to a specific date and time. In the end, the user can click on the button to complete the sending and request the server to send the Quick Email. In Figure 54, it is possible to observe the **user interface of the third step of the feature of sending a Quick Email**.

Figure 54 - User Interface of the Third Step of the Feature of Sending Quick Email

In Figure 55, it is possible to observe the **Schedule** section that allows choosing between sending the Email now or scheduling it to a specific date and time.

Schedule (now or later) ^

☐ Send Now
☒ Send Later

Choose a Date
2022-06-30

Choose a Time
12:00

Figure 55 - Schedule Section while Sending Quick Email

The feature of sending an Advanced Email was implemented mainly by creating a new Angular component called **AdvancedEmailFormComponent**. For this, as was mentioned before, it was added, to the sidebar of the application, a new item related to the sending of Quick Emails and Advanced Emails. Two tabs were created, one where the user could send Quick Emails, and the other one where the user could send Advanced Emails.

In this component, it is possible to send an Advanced Email by following three steps.

The **first step** is to import a csv file, which contains at least a column that represents the email of the recipient (it may contain more columns, such as the date and time to schedule). After importing the csv file, it must be selected which column contains the email of the recipient. In Figure 56, it is possible to observe the **user interface of the first step of the feature of sending an Advanced Email**.

Send Email

WITHOUT CSV FILE WITH CSV FILE

1 Import your File (.CSV) 2 Fill out your Email 3 Done

Change File

Select your csv file to import (Contact Column*)

email

email	name	date	time
jbernardo@e-goi.com	Bernardo	30-06-2022	12:00
acarvalho@e-goi.com	Antonio	30-06-2022	13:00
mazevedo@e-goi.com	Miguel	30-06-2022	14:00

All the rows are imported but only five are shown

Next >

Figure 56 - User Interface of the First Step of the Feature of Sending Advanced Email

The **second step** is to select the sender of the Email and fill out the subject and the email body (using an HTML editor). Optionally the user can fill both the subject and the email body by selecting an existing Email Template. Furthermore, the user can also add new variables to the email body, and replace them by selecting a column of the csv which contains the value pretended for each row (for example replacing the variable “%name%” with the value of the

column “name” for each row). Finally, in this step, the user can also add attachments to the Email, to a certain limit of megabytes. In Figure 57, it is possible to observe the **user interface of the second step of the feature of sending an Advanced Email**.

Figure 57 - User Interface of the Second Step of the Feature of Sending Advanced Email

In Figure 58, it is possible to observe the **Email section that allows selecting Email Templates, filling in the subject and the email body, and adding new variables and replacing them by selecting a column of the csv.**

Figure 58 - Email Section while Sending Advanced Email

The **third step** is the last and consists of visualizing a summary with the number of recipients, the sender, the subject, and the email body. Furthermore, it allows the user to choose between sending now the Email or scheduling it. If the user chooses to schedule, it is necessary to select

a column of the csv which contains the date, and a column of the csv which contains the time. In the end, the user can click on the button to complete the sending and request the server to send the Advanced Email. In Figure 59, it is possible to observe the **user interface of the third step of the feature of sending an Advanced Email**.

Figure 59 - User Interface of the Third Step of the Feature of Sending Advanced Email

In Figure 60, it is possible to observe the **Schedule section that allows choosing between sending the Email now or scheduling it, based on the columns of the csv selected as the date and time**.

Figure 60 - Schedule Section while Sending Advanced Email

7.2.4 Email Reports

This group is related only to the feature of visualizing Email Reports in the QuickMessage application.

The feature of visualizing Email Reports was implemented mainly by creating two new Angular components called **EmailReportsChartComponent** and **EmailsReportsTableComponent** and using them inside the area where it was possible to visualize the SMS Reports. For this, it was created two tabs, one where the user could visualize the SMS Reports, and the other one where the user could visualize the Email Reports.

In the **first component**, it is possible to visualize an Email Report graph of the last 30 days, by requesting the server. It is possible to visualize the date of the report, and how many emails were viewed, clicked, and rejected. In Figure 61, it is possible to observe the **user interface of the feature of visualizing Email Reports (graph)**.

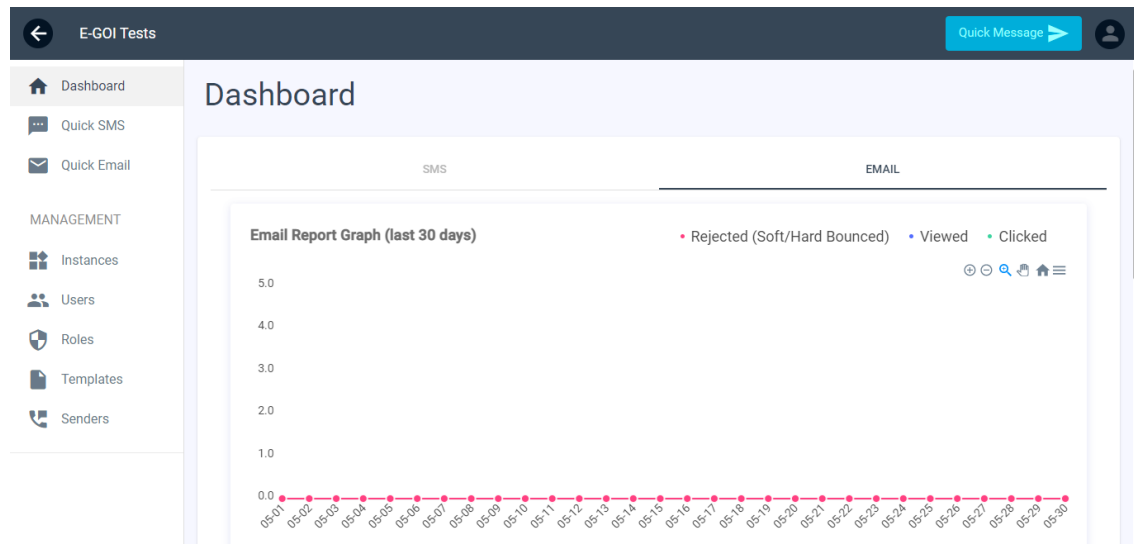


Figure 61 - User Interface of the Feature of Visualizing Email Reports (graph)

In the **second component**, it is possible to visualize an Email Report table of the last 30 days, by requesting the server. It is possible to visualize the date of the report, and how many emails were viewed, clicked, soft bounced (related to temporary errors), hard bounced (related to permanent errors), and their total. In Figure 62, it is possible to observe the **user interface of the feature of visualizing Email Reports (table)**.

The screenshot shows the same 'E-GOI Tests' dashboard as Figure 61, but with the 'EMAIL' tab displaying the 'Email Report Table (last 30 days)'. The table has the following structure:

Date	Viewed	Clicked	Soft Bounced	Hard Bounced	Total
2022-05-30	0	0	0	0	25
2022-05-29	0	0	0	0	24
2022-05-28	0	0	0	0	24
2022-05-27	0	0	0	0	26
2022-05-26	0	0	0	0	31
2022-05-25	0	0	0	0	70

At the bottom of the table, there is a pagination control showing 'Items per page: 7' and '1 - 7 of 30'.

Figure 62 - User Interface of the Feature of Visualizing Email Reports (table)

7.2.5 Scheduling

This group is related to the features of scheduling SMS messages and Emails and visualizing, deleting, and editing scheduled SMS messages and Emails in the QuickMessage application.

The **feature of scheduling Emails** was implemented as can be seen in Section 7.2.3, and the same was done to the **scheduling of SMS messages** since it was only necessary to repeat what was done in the features of Sending Quick Email and Sending Advanced Email.

The **features of visualizing and deleting scheduled SMS messages and Emails** were also implemented. However, these features weren't added to the final solution, since there were some business problems while implementing them, related to restrictions of the Transactional API. It was only possible to list individually each scheduled SMS message or Email, so if a user scheduled an SMS message or Email to 200 recipients, it would only be possible to cancel each one individually. It was pretended that if a user scheduled an Email to 200 recipients, the user could cancel all of them together.

The **features of editing scheduled SMS messages and Emails** weren't implemented since there were also some business problems while trying to implement them, related to restrictions of the Transactional API. It would only be possible to edit individually each scheduled SMS message or Email, and the pretended was to edit all of them together.

7.2.6 Permissions

This group is related to the quality attribute of securing the new functional requirements through a set of permissions and making it possible to add these permissions to the user's roles.

In the **backend**, there is a service that secures the Controller's methods from unauthorized access by checking if the user has the required permissions.

Each method of the Controllers has a custom annotation that allows defining the required permissions to access it. In Figure 63, it is possible to observe **the custom annotation that was added above the method of the Controller to send a Quick Email**.

```
@Authorization(feature = Feature.Send, featureModule = FeatureModule.SendEmail)
```

Figure 63 - Custom Annotation of the Method to Send a Quick Email

The custom annotation called "Authorization" is a **class created as an annotation that allows obtaining the feature and the feature module**, as can be seen in Figure 64.

```
@Target(METHOD)
@Retention(RUNTIME)
public @interface Authorization {
    Feature feature();
    FeatureModule featureModule();
}
```

Figure 64 – Authorization Class

Each time a method of the Controller is called, there is a service that detects it and handles the process of checking if the user has the required permissions. It obtains the annotation called “Authorization” defined in the method called, and then it calls a service that verifies if the user can access the feature by sending as parameters, the feature, the feature module, and the username. In Figure 65, it is possible to observe the **method that handles the process of checking if the user has the required permissions**.

```
public boolean preHandle(HttpServletRequest request, @NotNull HttpServletResponse response, @NotNull Object handler) {
    boolean canAccess = false;
    Principal user = request.getUserPrincipal();

    if (handler instanceof HandlerMethod) {
        Class<Authorization> authorizationClass = Authorization.class;
        Authorization authorization = ((HandlerMethod) handler).getMethod().getDeclaredAnnotation(authorizationClass);
        canAccess = privilegeService.checkIfUserCanAccessThisFeature(authorization.feature(), authorization.featureModule(), user.getName());
    }

    if (!canAccess) {
        LOGGER.info("Unauthorized: " + user.getName() + " | " + request.getContextPath());
        response.setStatus(HttpStatus.UNAUTHORIZED.value());
    }

    return canAccess;
}
```

Figure 65 - Controller's Authorization Handler Method

In the **frontend**, there is also a service that checks if the user has permission to view the feature in the interface, and hides it if not. Furthermore, there is also a feature that allows the user to **add the permissions related to the new features to the roles**, as can be seen in Figure 66.

Update Role

Name *

admin

ACCESS PERMISSION

DATA PERMISSION

Templates Management

SMS Templates

Email Templates

☒ Create

☒ Read

☒ Update

☒ Delete

Senders Management

Users Management

Roles Management

Instances Management

Send Management

Scheduled Management

Close

Update

Figure 66 - User Interface of the Feature of Updating Role

7.2.7 Documentation

This group is related to the quality attribute of documenting the application architecture and code.

The application architecture and code were mostly documented in the current document, however, it was also added Swagger Documentation to the Controllers of the QuickMessage application. In Figure 67, it is possible to observe an **example of the Swagger Documentation above the method of the Controller to send a Quick Email.**

```
@Operation(summary = "Sends a Simple Email.")
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "A Simple Email was sent with success."),
    @ApiResponse(responseCode = "500", description = "Something went wrong due to invalid input values supplied.")
})
@PostMapping(path = "/send-simple", consumes = "application/json", produces = "application/json")
ResponseBody<String> sendSimpleEmailMessage(@RequestBody SimpleEmailSendRequest req, HttpServletRequest request) throws InternalServerErrorException;
```

Figure 67 - Swagger Documentation Example

7.2.8 Response Time

This group is related to the quality attribute of having an adequate response time, of less than 1 second, in all the features of the system.

Most features in the system had a response time of less than 1 second, the only feature that had a response time superior was the sending of SMS messages and Emails since it was done synchronously. To decrease the response time, **it was changed the sending of SMS messages and Emails, to be done asynchronously**, so the user doesn't wait for all the messages to be sent. In Figure 68, it is possible to observe **the code that allows sending Advanced Emails asynchronously, through the use of the CompletableFuture.supplyAsync() method.**

```
for (AdvancedEmailInformation information : req.getInformation()) {
    if (information.getTo() == null || information.getTo().isEmpty()) {
        continue;
    }
    String toMessage = req.getHtmlBody();
    if (information.getVariables() != null) {
        for (AdvancedEmailVariable variable : information.getVariables()) {
            if (variable.getKey() == null || variable.getValue() == null) {
                continue;
            }
            toMessage = toMessage.replace(variable.getKey(), variable.getValue());
        }
    }

    final String finalMessage = toMessage;
    CompletableFuture.supplyAsync(() -> {
        try {
            return EmailDispatcher.sendEmail(apiKey, Collections.singletonList(
                new EmailSendRequest(req.getSenderId(), req.getSenderName(), Collections.singletonList(information.getTo()),
                    req.getSubject(), finalMessage, req.isOpenTracking(), req.isClickTracking(), req.getGroup(),
                    information.getScheduleTo(), req.isRegistered(), req.getAttachment())
            ));
        } catch (InternalServerErrorException | JsonProcessingException e) {
            logger.error("Error in Sending an Email due to: {}", e.getMessage());
        }
        return null;
    });
}
```

Figure 68 - Code that Allows Sending Advanced Emails Asynchronously

Furthermore, a **Spinner** was implemented, which appears when the server is still processing the request, to show the user to wait, as can be seen in Figure 69, where the server is still obtaining the list of the Email Templates.

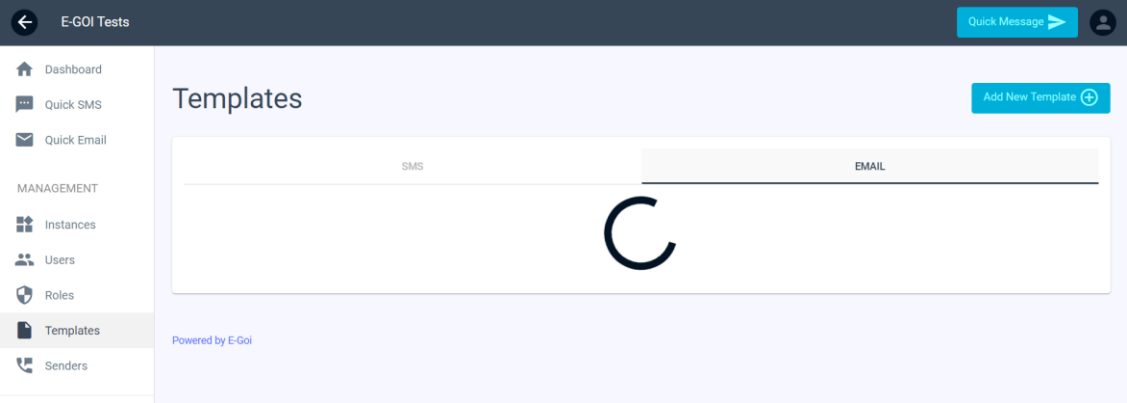


Figure 69 - Spinner when Loading the Email Templates

7.2.9 Notifications

This group is related to the quality attribute of having the system deal with all errors and presenting the unexpected errors to the user with a clear message about the error.

The **Toastr Angular Component** was used to present to the user a notification when an error occurs, when the operation is being done/processed by the server, and when the operation ends successfully, as can be seen in Figure 70.

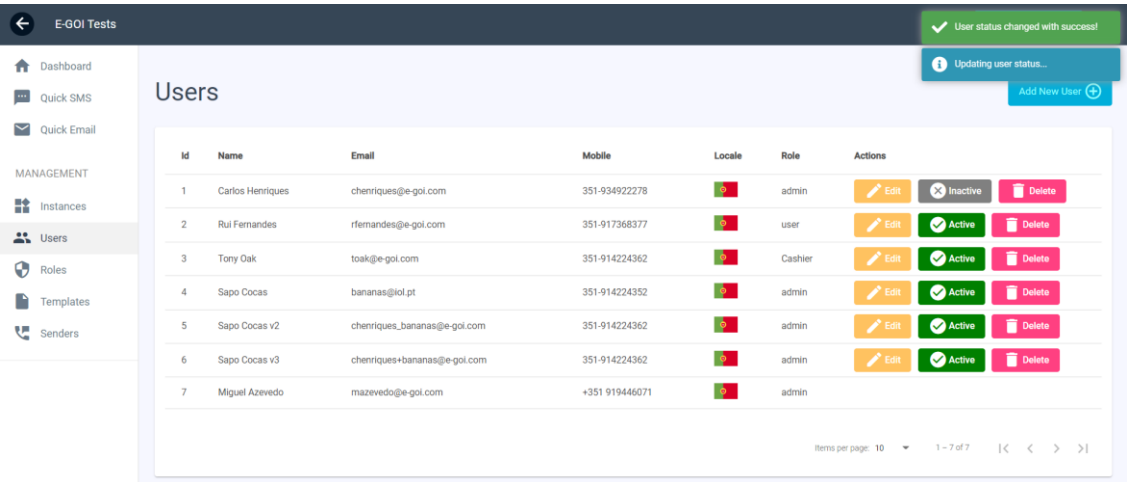


Figure 70 - Notifications when Changing User Status

8 Evaluation

This chapter is dedicated to demonstrating how the solution was evaluated using evaluation indicators, the hypotheses defined for them and the evaluation framework applied called Quantitative Evaluation Framework (QEF). In addition, the tests performed on the implemented solution, based on some quality attributes, are explained.

8.1 Evaluation Indicators

To assist in the evaluation of the system, the following indicators were defined:

- Fulfillment of Functional and Non-Functional Requirements;
- System Quality;
- System Usability.

The first indicator identified allows verifying if the functional and non-functional requirements were achieved. The second indicator is related to the quality of the system and it intends to ensure that the final product meets the needs of the users and decreases the number of existing problems. Finally, the last indicator is related to the usability of the system and allows for assessing whether the final product is efficient, effective, and can be used to bring satisfaction to the users, so they can have a good experience with the platform.

8.2 Definition of Hypotheses

Once the evaluation indicators have been defined, it is required to evaluate them, through the following hypotheses that are defined:

- Fulfillment, in full, of functional and non-functional requirements;
- Solution quality level at 100%;
- Usability degree of the solution greater than 80%.

The first hypothesis is intended to ensure that the requirements have been fully achieved. The second hypothesis is intended to reach the maximum level of quality, and this directly depends on the requirements completion rate. Concerning the last defined hypothesis, it indicates that usability must reach a rate greater than 80%. This value comes from the adoption of the System Usability Scale (SUS) evaluation model, proposed by John Brooke in 1986, based on the use of questionnaires to be answered by users, to assess their experience with a system [32].

8.3.2 QEF Model

The QEF model, proposed by Paula Escudeiro and José Bidarra in 2006, is an evaluation method that aims to improve the quality of a software solution, allowing to detect and fix, at any stage of development, any failures that may occur [34].

QEF model is based on three-dimensional visualization of the qualities associated with the solution to be evaluated, in which each orthogonal dimension is composed of a set of factors that, in turn, are composed of a set of requirements [34].

A factor has a relative weight of importance, which is calculated by the number of requirements associated with it, divided by the total number of requirements within the dimension where the factor is inserted. It has also a relative weight of fulfillment, which is calculated by the sum of the multiplication of each requirement by its fulfillment value, divided by the sum of the weight of the requirements within that factor [34].

A requirement has a specific degree of relevance, which is evaluated on a pre-defined scale, that varies between 2, 4, 6, 8, and 10, with 10 being the highest value. It can be evaluated through the values 0, 25, 50, 75, and 100 that correspond to a specific description based on fulfillment indicators [34].

To calculate the quality in the QEF model, it is necessary to understand two distinct concepts [34]:

- Ideal System (IS) – Represents the ideal state in which the solution to be evaluated must be in (solution with maximum quality);
- Real System (RS) – Represents the current state of the solution.

Therefore, and according to the QEF model, the quality of the system is inversely proportional to the Euclidean distance between the IS and the RS. This means that the smaller the Euclidean distance between the IS and the RS, the higher is the actual quality of the solution [34].

In Figure 72, it is possible to observe the three-dimensional quality space.

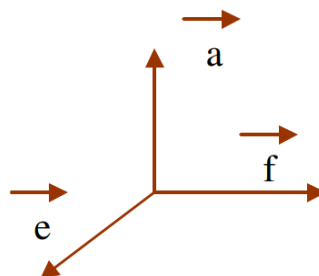


Figure 72 - Three-dimensional quality space [34]

For this project, the dimensions, factors, requirements, and respective weights are represented in Figure 73.

q	D	Qi	Dimension	Q _i	Factor Weight	Factor	Requirement Weight	Requirement	Fulfillment
0%	1.73	0	Functionality	0	0.54	Email Channel	10	FEC01 - Send Quick Email	0
							10	FEC02 - Send Advanced Email	0
							10	FEC03 - Create Email Template	0
							10	FEC04 - View Email Templates	0
							10	FEC05 - Manage Email Templates (delete/edit)	0
							6	FEC06 - View Email Reports/Statistics	0
							10	FEC07 - View Email Senders	0
		0	Functionality	0	0.46	Schedule SMS Messages and Emails	10	FS01 - Schedule SMS Message	0
							10	FS02 - Schedule Email	0
							8	FS03 - View Scheduled SMS Messages	0
							8	FS04 - View Scheduled Emails	0
							6	FS05 - Manage Scheduled SMS Messages (delete/edit)	0
							6	FS06 - Manage Scheduled Emails (delete/edit)	0
		0	Adaptability	0	1.00	Maintenance	10	AM01 - Configure New Permissions (Email Template, Email Senders, Send Email)	0
							8	AM02 - Document Implementation	0
		0	Efficiency	0	0.33	Performance	10	EP01 - System must have an adequate response time	0
				0	0.67	Usability	10	EU01 - System must deal with the errors	0
							10	EU02 - Interface must be simple and intuitive	0

Figure 73 - Dimensions, Factors, Requirements, and respective Weights of the QEF Model

As it can be observed in Figure 73, it was defined three distinct dimensions:

- Functionality;
- Adaptability;
- Efficiency.

For the Functionality dimension, 2 factors were defined (Email Channel and Schedule SMS Messages and Emails), with a total of 13 requirements. The list of requirements associated with this dimension, as well as the requirements metric evaluation can be found in Figure 74 and Figure 75.

Requirement	Metric Evaluation	Wfk - Fullfilment (%)		
		0	50	100
FEC01 - Send Quick Email	Users can send a quick Email (without Excel input) with attachments.	Not Implemented	Implemented Without Attachments Feature	Fully Implemented
FEC02 - Send Advanced Email	Users can send a advanced Email (with Excel input) with attachments.	Not Implemented	Implemented Without Attachments Feature	Fully Implemented
FEC03 - Create Email Template	Users can create Email Templates.	Not Implemented	-	Fully Implemented
FEC04 - View Email Templates	Users can view the list of Email Templates.	Not Implemented	-	Fully Implemented
FEC05 - Manage Email Templates (delete / edit)	Users can delete and edit existing Email Templates.	Not Implemented	-	Fully Implemented
FEC06 - View Email Reports/Statistics	Users can view the reports and statistics related to the Email Sending.	Not Implemented	-	Fully Implemented
FEC07 - View Email Senders	Users can view the list of Email Senders.	Not Implemented	-	Fully Implemented

Figure 74 - Email Channel Factor Requirements

Requirement	Metric Evaluation	Wfk - Fullfilment (%)		
		0	50	100
FS01 - Schedule SMS Message	Users can schedule the sending of a SMS Message.	Not Implemented	-	Fully Implemented
FS02 - Schedule Email	Users can schedule the sending of a Email.	Not Implemented	-	Fully Implemented
FS03 - View Scheduled SMS Messages	Users can view the list of scheduled SMS Messages.	Not Implemented	-	Fully Implemented
FS04 - View Scheduled Emails	Users can view the list of scheduled Emails.	Not Implemented	-	Fully Implemented
FS05 - Manage Scheduled SMS Messages (delete / edit)	Users can delete and edit existing scheduled SMS Messages (only if the SMS message wasn't already sent).	Not Implemented	-	Fully Implemented
FS06 - Manage Scheduled Emails (delete / edit)	Users can delete and edit existing scheduled Emails (only if the Email wasn't already sent).	Not Implemented	-	Fully Implemented

Figure 75 - Schedule SMS Messages and Emails Factor Requirements

For the Adaptability dimension, only 1 factor was defined (Maintenance), with a total of 2 requirements. The list of requirements associated with this dimension, as well as the requirements metric evaluation can be found in Figure 76.

Requirement	Metric Evaluation	Wfk - Fullfilment (%)		
		0	50	100
AM01 - Configure New Permissions (Email Template, Email Senders, Send Email)	The new requirements must be secured by a set of permissions. It must be possible to add these permissions to the user's roles.	Not Implemented	-	Fully Implemented.
AM02 - Document Implementation	Application architecture, and code must be documented.	Not Documented	Partially Documented	Fully Documented

Figure 76 - Maintenance Factor Requirements

For the Efficiency dimension, 2 factors were defined (Performance and Usability), with a total of 3 requirements. The list of requirements associated with this dimension, as well as the requirements metric evaluation can be found in Figure 77.

Requirement	Metric Evaluation	Wfk - Fullfilment (%)		
		0	50	100
EP01 - System must have an adequate response time	All features of the system must have a response time inferior than 1 second. If it takes more than 1 second for some specific reason, it must show a progress message to the user.	Not Implemented	-	Fully Implemented
EU01 - System must deal with the errors	The system must deal with all errors and present the unexpected errors to the user with a clear message about the error.	Not Implemented	-	Fully Implemented
EU02 - Interface must be simple and intuitive	The interface must be simple and intuitive, to bring a good experience to the customers.	SUS Score < 67	SUS Score Between 67 and 80	SUS Score > 80

Figure 77 - Performance and Usability Factors Requirements

8.3.3 Usability Questionnaire

The usability evaluation of a solution provides an analysis of how easily the user interacts with that solution, and whether he approves or denies that solution. As such, it is necessary to define a method to evaluate how the user interacts with a system. This evaluation will be based on the System Usability Scale (SUS) evaluation method [32].

The SUS evaluation model, proposed by John Brooke in 1986, is based on the use of questionnaires to be answered by users, to assess their experience with a system. The questionnaire evaluates usability using 10 items/questions with 5 response options for respondents, from Strongly Disagree to Strongly Agree [32].

The items/questions used in this questionnaire and the response scale, are represented in Table 32 and Table 33, respectively.

Table 32 - Items/Questions of the SUS Questionnaire [32]

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

Table 33 - Scale and Description of the SUS Questionnaire Answers [32]

Scale	Description
1	Strongly Disagree
2	Disagree
3	Indifferent/No Opinion
4	Agree
5	Strongly Agree

After completing the questionnaire, it is important to interpret the results and to carry out this interpretation, it is necessary to normalize the results obtained. Therefore, for each questionnaire, the following steps must be followed [35]:

1. Separate questions whose number is even and whose number is odd, forming two distinct groups;

2. For each question in the odd-numbered question group, subtract 1 from the answer value ($X - 1$);
3. For each question in the even-numbered question group, subtract from the value 5 the answer value ($5 - X$);
4. Add all the values obtained, after the previous transformations;
5. Multiply the value obtained by 2.5.

After all these steps, it is necessary to calculate the average score of all the questionnaires answered by the users. This final value, which is between 0 and 100, represents the result of the SUS evaluation method, which can be framed in the evaluation scale presented in Table 34.

Table 34 – Interpretation of SUS Score [35]

SUS Score	Adjective Rating
> 80	Excellent
68 - 80	Good
67	Okay
51-66	Poor
< 51	Awful

8.4 Results

In the analysis of results, the implemented solution is validated, by verifying if the objectives have been achieved and identifying future work related to the solution. Firstly, the fulfillment of the requirements by analyzing the results of the Software Testing is verified. After that, the system quality is verified by analyzing the QEF Model. Finally, the system usability is verified by analyzing the Usability Questionnaire.

8.4.1 Software Testing

The first evaluation method analyzed is related to the indicator of the fulfillment of the solution requirements. To this end, two categories of software tests were analyzed, the Integration Tests and the System Tests, according to the V-Model.

It was decided to analyze only these two categories since it was only considered important to test the communication between the QuickMessage backend and the Transactional API (with Integration Tests) and the QuickMessage application as a whole (with System Tests).

Integration Tests

According to the V-Model, the Integration Tests are the ones that follow the Unit Tests in the validation phase. These tests are used to evaluate the compliance of a system by combining several individual components [33].

The integration tests were developed in a tool called Postman. This tool allows making requests to a server by mainly specifying the server Uniform Resource Locator (URL) and the HTTP method. It can also be optionally provided the request body and request headers.

To this end, it was created a Postman collection that it was used to create many requests to test all the new features endpoints.

Firstly, regarding the **Email Senders**, it was created a GET request with the URL of the QuickMessage endpoint to obtain the Email Senders. This request returned a 200 HTTP status code with all the Email Senders, which was expected.

Secondly, regarding the **Email Templates**, it was created four requests, which were executed in the following order:

1. A GET request with the URL of the QuickMessage endpoint to obtain the Email Templates. This request returned a 200 HTTP status code with all the Email Templates, which was expected.
2. A POST request with the URL of the QuickMessage to create a new Email Template and a request body with the content of the Email Template. This request returned a 200 HTTP status code with the created Email Template, which was expected.
3. A PUT request with the URL of the QuickMessage to update an existing Email Template and a request body with the updated content of the Email Template. This request returned a 200 HTTP status code with the updated Email Template, which was expected.
4. A DELETE request with the URL of the QuickMessage to delete an existing Email Template. This request returned a 200 HTTP status code, which was expected (after deleting the Email Template it was checked if it was deleted).

Thirdly, regarding the **Emails Sending**, it was created four requests:

- A POST request with the URL of the QuickMessage endpoint to send a Quick Email and a request body with the Email content (without scheduling). This request returned a 200 HTTP status code, which was expected (after sending the Email it was checked if the Email was received).
- A POST request with the URL of the QuickMessage endpoint to send an Advanced Email and a request body with the Email content (without scheduling). This request returned a 200 HTTP status code, which was expected (after sending the Email it was checked if the Email was received).
- A POST request with the URL of the QuickMessage endpoint to send a Quick Email and a request body with the Email content (with scheduling). This request returned a 200 HTTP status code, which was expected (after sending the Email it was checked if the Email was received at the specific date and time).
- A POST request with the URL of the QuickMessage endpoint to send an Advanced Email and a request body with the Email content (with scheduling). This request returned a 200 HTTP status code, which was expected (after sending the Email it was checked if the Email was received at the specific date and time).

Fourthly, regarding the **Email Reports**, it was created a GET request with the URL of the QuickMessage endpoint to obtain the Email Reports of the last 30 days. This request returned a 200 HTTP status code with all the Email Reports of the last 30 days, which was expected.

Finally, regarding the **Scheduling**, the feature of scheduling Emails and SMS messages was already tested in the requests related to Emails Sending. The features of visualizing, deleting, and editing scheduled SMS messages and Emails, since there were some business problems while implementing them, no requests were made.

System Tests

According to the V-Model, System Tests follow Integration Tests and their main objective is to test the system as a whole, based on the definition of use cases. The success of a system test is only completed if it does not fail any previously defined criteria according to the use case definition. So, if at least one criterion fails, the test fails [33].

To this end, it was done System Tests for all the new use cases developed.

Firstly, regarding the **Email Senders**, in Table 35, it is possible to observe the criteria and results of the System Test performed on the **UC08 called View Email Senders**.

Table 35 - Criteria and Results of the UC08 called View Email Senders

Criteria	Result
The system allows visualizing Email Senders, only if the user has the required permissions.	Success
The system shows the list of Email Senders.	Success
System Test Completed with Success	

Secondly, regarding the **Email Templates**, in Table 36, Table 37, Table 38, and Table 39 it is possible to observe the criteria and results of the System Tests performed on the use cases. System Tests are performed on the **UC03 called Create Email Template**, **UC04 called View Email Templates**, **UC05 called Delete Email Template**, and **UC06 called Edit Email Template**.

Table 36 - Criteria and Results of the UC03 called Create Email Template

Criteria	Result
The system allows creating Email Templates, only if the user has the required permissions.	Success
The system allows the user to fill in the template name, the subject, and the HTML body.	Success
The system allows the user to insert variables.	Success
The system creates and saves the new Email Template.	Success
System Test Completed with Success	

Table 37 - Criteria and Results of the UC04 called View Email Templates

Criteria	Result
The system allows visualizing Email Templates, only if the user has the required permissions.	Success
The system shows the list of Email Templates.	Success
System Test Completed with Success	

Table 38 - Criteria and Results of the UC05 called Delete Email Template

Criteria	Result
The system allows deleting Email Templates, only if the user has the required permissions.	Success
The system deletes the existing Email Template.	Success
System Test Completed with Success	

Table 39 - Criteria and Results of the UC06 called Edit Email Template

Criteria	Result
The system allows editing Email Templates, only if the user has the required permissions.	Success
The system allows the user to fill in the template name, the subject, and the HTML body.	Success
The system allows the user to insert variables.	Success
The system edits and saves the existing Email Template.	Success
System Test Completed with Success	

Thirdly, regarding the **Emails Sending**, in Table 40, and Table 41 it is possible to observe the criteria and results of the System Tests performed on the use cases. System Tests are performed on the **UC01 called Send Quick Email**, and **UC02 called Send Advanced Email**.

Table 40 - Criteria and Results of the UC01 called Send Quick Email

Criteria	Result
The system allows sending Quick Emails, only if the user has the required permissions.	Success
The system allows the user to fill in the recipients.	Success
The system allows the user to select the sender and templates, and fill in the subject and the HTML body.	Success
The system allows the user to add attachments.	Success
The system allows the user to view a summary at the end.	Success
The system allows the user to send the Email at the moment or schedule it to a specific date and time.	Success
The system sends the Quick Email to the recipients.	Success
System Test Completed with Success	

Table 41 - Criteria and Results of the UC02 called Send Advanced Email

Criteria	Result
The system allows sending Advanced Emails, only if the user has the required permissions.	Success
The system allows the user to import a CSV file and select the column that contains the recipients.	Success
The system allows the user to select the sender and templates, and fill in the subject and the HTML body.	Success
The system allows the user to add variables and replace variables by selecting a column of the CSV.	Success
The system allows the user to add attachments.	Success
The system allows the user to view a summary at the end.	Success
The system allows the user to send the Email at the moment or schedule it to a specific date and time, for each recipient.	Success
The system sends the Advanced Email to the recipients.	Success
System Test Completed with Success	

Fourthly, regarding the **Email Reports**, in Table 42, it is possible to observe the criteria and results of the System Test performed on the **UC07 called View Email Reports**.

Table 42 - Criteria and Results of the UC07 called View Email Reports

Criteria	Result
The system allows visualizing Email Reports.	Success
The system shows the list of Email Reports for the last 30 days.	Success
System Test Completed with Success	

Finally, regarding the **Scheduling**, the feature of scheduling Emails and SMS messages was already tested in the System Tests related to Emails Sending. The features of visualizing, deleting, and editing scheduled SMS messages and Emails, since there were some business problems while implementing them, no System Tests were made.

8.4.2 QEF Model

The second evaluation method analyzed is related to the indicator of the system quality. To this end, the QEF Model is used, in which the main objective is to improve the quality of a software solution. This is done through the detection and consequent correction of any failures that may occur during the entire development process [34].

The results of the fulfillment of the requirements and the overall system quality using the QEF Model are represented in Figure 78.

q	D	Qi	Dimension	Qi	Factor Weight	Factor	Requirement Weight	Requirement	Fulfillment
93%	0.27	73.08	Functionality	100	0.54	Email Channel	10	FEC01 - Send Quick Email	100
							10	FEC02 - Send Advanced Email	100
							10	FEC03 - Create Email Template	100
							10	FEC04 - View Email Templates	100
							10	FEC05 - Manage Email Templates (delete/edit)	100
							6	FEC06 - View Email Reports/Statistics	100
							10	FEC07 - View Email Senders	100
				41.67	0.46	Schedule SMS Messages and Emails	10	FS01 - Schedule SMS Message	100
							10	FS02 - Schedule Email	100
							8	FS03 - View Scheduled SMS Messages	0
							8	FS04 - View Scheduled Emails	0
							6	FS05 - Manage Scheduled SMS Messages (delete/edit)	0
							6	FS06 - Manage Scheduled Emails (delete/edit)	0
		100	Adaptability	100	1.00	Maintenance	10	AM01 - Configure New Permissions (Email Template, Email Senders, Send Email)	100
							8	AM02 - Document Implementation	100
		100	Efficiency	100	0.33	Performance	10	EP01 - System must have an adequate response time	100
							10	EU01 - System must deal with the errors	100
				100	0.67	Usability	10	EU02 - Interface must be simple and intuitive	100

Figure 78 – Results of the Fulfillment of Requirements and Overall System Quality, using the QEF Model

Looking at Figure 78, it is possible to observe that each of the 18 defined requirements was evaluated individually according to the percentage of fulfillment.

Regarding the **Functionality** dimension, of the 13 requirements associated with it, only 9 have a fulfillment percentage of 100%. The 4 requirements that have a fulfillment of 0% are the requirements associated with the visualization and managing of scheduled SMS messages and Emails since these weren't fully implemented (because of some business problems). This way, the **Functionality** dimension is implemented at **73.08%**, when compared to the ideal solution.

Regarding the **Adaptability** dimension, all of the 2 requirements associated with it, have a fulfillment percentage of 100%. This way, the **Adaptability** dimension is the most as close as possible to the ideal solution, with a fulfillment percentage of **100%**.

Regarding the **Efficiency** dimension, all of the 3 requirements associated with it, have a fulfillment percentage of 100%. This way, the **Efficiency** dimension is the most as close as possible to the ideal solution, with a fulfillment percentage of **100%**.

The real solution is at a **Euclidean distance of 0.27** from the ideal solution. Considering the solution under analysis, the conclusion is that the **real solution represents an implementation of 93% of the ideal solution**.

8.4.3 Usability Questionnaire

The third evaluation method analyzed is related to the indicator of the system usability. To this end, a Usability Questionnaire based on the System Usability Scale (SUS) is used. This allows the evaluation of the user experience, by providing the user a guide with steps, and a questionnaire with 10 items/questions with 5 response options for respondents, from Strongly Disagree to Strongly Agree [32].

Regarding the **guide**, it was provided to the user a guide with the following steps:

1. Login into the application with the credentials provided.
2. Click on the menu button to view Email Templates.
3. Click on the button to initiate the creation of an Email Template from scratch.
4. Fill in the template name, subject, and HTML body, and click on the button to create.
5. Click on the menu button to send a Quick Email.
6. Fill in the recipients.
7. Select the sender and the Email Template created, and add attachments to the Email.
8. Schedule the Email to a specific date and time and click on the button to send the Quick Email.

Regarding the **questionnaire**, one was provided with the questions presented in Table 32.

This way, the guide, and questionnaire were provided to **30 different people**, which includes E-goï collaborators, other software engineers, and people with no knowledge in the area (customers were not targeted in the study). In Table 43, it is possible to observe the answers of the users to each question.

Table 43 - Answers of the Users to the SUS Questionnaire

Question	Answer				
	1	2	3	4	5
I think that I would like to use this system frequently	0	0	0	21	9
I found the system unnecessarily complex	25	5	0	0	0
I thought the system was easy to use	0	0	0	3	27
I think that I would need the support of a technical person to be able to use this system	29	1	0	0	0
I found the various functions in this system were well integrated	0	0	10	20	0
I thought there was too much inconsistency in this system	26	4	0	0	0
I would imagine that most people would learn to use this system very quickly	0	0	0	24	6
I found the system very cumbersome to use	28	2	0	0	0
I felt very confident using the system	0	0	7	23	0
I needed to learn a lot of things before I could get going with this system	20	7	3	0	0

Regarding the **first question**, it is possible to observe that 21 users answered with “Agree” and 9 users answered with “Strongly Agree”.

Regarding the **second question**, it is possible to observe that 25 users answered with “Strongly Disagree” and 5 users answered with “Disagree”.

Regarding the **third question**, it is possible to observe that 3 users answered with “Agree” and 27 users answered with “Strongly Agree”.

Regarding the **fourth question**, it is possible to observe that 29 users answered with “Strongly Disagree” and 1 user answered with “Disagree”.

Regarding the **fifth question**, it is possible to observe that 10 users answered with “Indifferent/No Opinion” and 20 users answered with “Agree”.

Regarding the **sixth question**, it is possible to observe that 26 users answered with “Strongly Disagree” and 4 users answered with “Disagree”.

Regarding the **seventh question**, it is possible to observe that 24 users answered with “Agree” and 6 users answered with “Strongly Agree”.

Regarding the **eighth question**, it is possible to observe that 28 users answered with “Strongly Disagree” and 2 users answered with “Disagree”.

Regarding the **ninth question**, it is possible to observe that 7 users answered with “Indifferent/No Opinion” and 23 users answered with “Agree”.

Regarding the **tenth question**, it is possible to observe that 20 users answered with “Strongly Disagree”, 7 users answered with “Disagree”, and 3 users answered with “Indifferent/No Opinion”.

In Table 44, it is possible to observe the average of the answers for each question and the normalized results. The normalized results are obtained by subtracting 1 from the average of the odd questions, and by subtracting from the value 5 the average of the even questions.

Table 44 - Average of the Answers of the Users and Normalized Results to the SUS Questionnaire

Question Number	Average	Normalized Result
1	4.3	$4.3 - 1 = 3.3$
2	1.17	$5 - 1.17 = 3.83$
3	4.9	$4.9 - 1 = 3.9$
4	1.03	$5 - 1.03 = 3.97$
5	3.67	$3.67 - 1 = 2.67$
6	1.13	$5 - 1.13 = 3.87$
7	4.2	$4.2 - 1 = 3.2$
8	1.067	$5 - 1.067 = 3.933$
9	3.77	$3.77 - 1 = 2.77$
10	1.43	$5 - 1.43 = 3.57$

By adding up all the normalized results, the value obtained is 35.015, and then by multiplying this value by 2.5, the **final value of the SUS Questionnaire is 87.53**, which is superior to the 80 pretended.

9 Conclusions

This chapter is dedicated to describing the achieved goals and results and identifying some limitations and future work.

9.1 Achieved Goals and Results

In the initial phase of the project, different objectives were defined, as can be seen in section 1.4. At this point of the document, it is important to evaluate whether the objectives were successfully achieved or not, since the fulfillment of all the objectives means a successful project.

The main objective was to analyze and improve the transactional communication tool called QuickMessage, by adding new features and making some improvements related to usability and performance. It was pretended to:

1. Implement the Email transactional channel;
2. Implement the feature to allow scheduling SMS messages and Emails;
3. Solve the problems of the competitors having more features and customers not being fully satisfied;
4. Solve the usability and performance problems.

The **first objective**, related to the **implementation of the Email transactional channel**, has been successfully achieved. Firstly, it was implemented a feature that allows users to visualize the list of Email Senders created and configured within the account. Secondly, it was implemented some features related to Email Templates, that allow users to visualize, create, edit and delete Email Templates. After that, it was implemented a feature to send Quick Emails, which allows users to send Emails without a CSV file, and a feature to send Advanced Emails, which allows users to send Emails with a CSV file. Finally, it was implemented a feature that allows users to visualize reports/statistics related to the Email sending. Before implementing these features, all of them went through an analysis and design phase, and after implementing them, they went through an evaluation phase, which they passed. In the evaluation phase, all features successfully passed the integration and system tests and were considered 100% fulfilled by the QEF Model.

The **second objective**, related to the **implementation of the feature to allow scheduling SMS messages and Emails**, has also been successfully achieved. When sending an SMS message or Email, an option was added that allows users to send the SMS message or Email at the moment or schedule it to a specific date and time. Furthermore, it was also planned to implement a feature that allows users to visualize, edit, and delete scheduled SMS messages or Emails. However, these features weren't added to the final solution, since there were some business problems while implementing them, related to restrictions of the Transactional API. All of these

features went through an analysis and design phase, however only the feature that allows the scheduling of SMS messages and Emails was implemented and passed the evaluation phase. This feature successfully passed the integration and system tests and was considered 100% fulfilled by the QEF model.

The **third objective**, related to the **resolution of the problems of the competitors having more features and customers not being fully satisfied**, has also been successfully achieved. Firstly, it was done a comparative analysis between different platforms, based on the features available in each one of them, to understand the limitations and differentiation factors of each one of the platforms. After that, it was concluded that to compete with the rival platforms it was necessary to implement the new Email channel, and the feature to allow the scheduling of SMS messages and Emails, which were implemented.

The **final objective**, related to the **resolution of usability and performance problems**, has also been successfully achieved. To this end, the QuickMessage platform was analyzed and some problems were found and fixed, which were related to the quality attributes defined. Firstly, it was changed the sending of SMS messages and Emails, to be done asynchronously, so the user doesn't wait for all the messages to be sent. Secondly, a Spinner was implemented, which appears when the server is still processing the request, to show the user to wait. Thirdly, it was implemented notifications that appear when an error occurs, when the operation is being done/processed by the server, and when the operation ends successfully. Finally, the usability was evaluated by 30 people with a SUS Questionnaire, and it was concluded that the usability level was excellent.

Bearing in mind that all objectives were achieved, it is safe to conclude that the present project was successful.

9.2 Limitations and Future Work

Even if the developed project is considered to have fulfilled all the objectives proposed, this does not mean that it is free from any limitations. Therefore, in this section, some limitations are identified that could be the target of future work.

Regarding the features related to visualizing, editing, and deleting SMS messages or Emails, these features weren't added to the final solution, since there were some business problems while implementing them, related to restrictions of the Transactional API. It was only possible to list individually each scheduled SMS message or Email, so if a user scheduled an SMS message or Email to 200 recipients, it would only be possible to cancel each one individually. It was pretended that if a user scheduled an Email to 200 recipients, the user could cancel all of them together, which is not possible due to this limitation of the Transactional API.

Regarding the usability questionnaire, the sample of 30 users isn't considered to be small. However, the results would be more reliable and complete if the questionnaire was filled in by more users and if all of the users were customers. So the fact that only 30 users filled in the

questionnaire, and no one was a customer of the QuickMessage platform is considered a limitation.

Given all the work carried out, and based on the defined limitations, it becomes relevant to define the future work that comes from the development of this dissertation. Firstly, the features related to visualizing, editing, and deleting SMS messages or Emails should be fully implemented, which will be only possible if the Transactional API is changed to support them. Then, it should be implemented more transactional channels and added more unique features, which allow it to differentiate itself from the rival platforms.

References

- [1] K. Peffers, T. Tuunanen, M. Rothenberger e S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, pp. 44-77, 2007.
- [2] M. Hudák, E. Kianičková e R. Madleňák, "The Importance of E-mail Marketing in E-commerce," *Procedia Engineering*, vol. 192, pp. 342-347, 2017.
- [3] A. Andonov, P. G. Dimitrov e V. Totev, "Impact of E-commerce on Business Performance," *TEM*, vol. 10, nº 4, pp. 1558-1564, 2021.
- [4] J. Hardy, "The History Of Marketing: From Trade to Tech," 14 September 2016. [Online]. Available: <https://historycooperative.org/the-evolution-of-marketing-from-trade-to-tech/>. [Acedido em 5 January 2022].
- [5] American Marketing Association, "Definitions of Marketing," 2017. [Online]. Available: <https://www.ama.org/the-definition-of-marketing-what-is-marketing/>. [Acedido em 10 January 2022].
- [6] A. Riva e L. Pilotti, "One-to-one Marketing. How to Build an Effective Digital Marketing Strategy," vol. 12, nº 3, pp. 329-349, 2021.
- [7] "Digital Marketing," Mail Chimp, [Online]. Available: <https://mailchimp.com/marketing-glossary/digital-marketing/>. [Acedido em 16 January 2022].
- [8] A. Adhami, A. Rabiee e M. Adhami, "A conceptual investigation of variables affecting the success and acceptance of SMS Marketing in Iran," *Journal of Medicine & Life*, vol. 8, nº 2, pp. 92-98, 2015.
- [9] D. McCorkle, V. Jurkus, V. Auruskeviciene e J. Reardon, "The Antecedent Effects of SMS Marketing on Consumer Intentions," *International Journal of Business*, vol. 6, nº 1, pp. 88-105, 2013.
- [10] "What is SMS Marketing," Sendinblue, 26 October 2021. [Online]. Available: <https://www.sendinblue.com/blog/what-is-sms-marketing/>. [Acedido em 17 January 2022].
- [11] "Email Marketing," Mail Chimp, [Online]. Available: <https://mailchimp.com/marketing-glossary/email-marketing/>. [Acedido em 17 January 2022].

- [12] M. Goic, A. Rojas e I. Saavedra, "The Effectiveness of Triggered Email Marketing in Addressing Browse Abandonments," *Journal of Interactive Marketing*, vol. 55, pp. 118-145, 2021.
- [13] A. Bismo, S. Putra e Melysa, "Application of Digital Marketing (social media and email marketing) and its Impact on Customer Engagement in Purchase Intention: a case study at PT. Soltius Indonesia," em *2019 International Conference on Information Management and Technology (ICIMTech)*, Jakarta/Bali, Indonesia, 2019.
- [14] "Why You Should Use Triggered Marketing Communications," GSMarketing, 3 May 2020. [Online]. Available: <http://blog.gsmarketing.com/blog/why-you-should-use-triggered-marketing-communications>. [Acedido em 20 January 2022].
- [15] "Relationship Marketing Strategies – Differentiating Relationship Marketing and Transaction Marketing," Marketing Insider, 15 April 2018. [Online]. Available: <https://marketing-insider.eu/relationship-marketing-strategies/>. [Acedido em 21 January 2022].
- [16] "QuickMessage," E-goi, [Online]. Available: <https://quickmessage.app/#/>. [Acedido em 23 January 2022].
- [17] "MailChimp," [Online]. Available: <https://mailchimp.com/>. [Acedido em 23 January 2022].
- [18] "MailJet," [Online]. Available: <https://www.mailjet.com/>. [Acedido em 23 January 2022].
- [19] "SMSAPI," [Online]. Available: <https://www.smsapi.com/en>. [Acedido em 23 January 2022].
- [20] "Sendinblue," [Online]. Available: <https://pt.sendinblue.com/>. [Acedido em 23 January 2022].
- [21] "uSendit," [Online]. Available: <https://www.usendit.pt/>. [Acedido em 23 January 2022].
- [22] R. Sbragia e L. Bonini, "O Modelo de Design Thinking como Indutor da Inovação nas Empresas: Um Estudo Empírico," *Revista de Gestão e Projetos*, vol. 2, pp. 3-25, 2011.
- [23] M. Jacobides, T. Knudsen and M. Augier, "Benefiting from innovation: Value creation, value appropriation and the role of industry architectures," *Research Policy*, vol. 35, no. 8, pp. 1200-1221, 2006.
- [24] M. Leber, M. Bastič, M. Mavrič e A. Ivanišević, "Value Analysis as an Integral Part of New Product Development," *Procedia Engineering*, vol. 69, pp. 90-98, 2014.

- [25] P. Koen, G. Ajamian, S. Boyce, A. Clamen, E. Fisher, S. Fountoulakis, A. Johnson, P. Puri e R. Seibert, "Fuzzy Front End: Effective Methods, Tools, and Techniques," 2002.
- [26] R. Saaty, "The analytic hierarchy process—what it is and how it is used," *Mathematical Modelling*, vol. 9, nº 3-5, pp. 161-176, 1987,.
- [27] T. Saaty e K. Kearns, *Analytical Planning: The Organization of System*, 1985, pp. 19-62.
- [28] T. Woodall, "Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis," *Academy of Marketing Science Review*, vol. 12, 2003.
- [29] O. Itani, A.-N. Kassar e S. Loureiro, "Value get, value give: The relationships among perceived value, relationship quality, customer engagement, and value consciousness," *International Journal of Hospitality Management*, vol. 80, pp. 78-90, 2019.
- [30] S. Bagchi e B. Tulske, "E-business Models: Integrating Learning from Strategy," *20th Annual International Conference of the Strategic Management Society*, 2000.
- [31] N. Ali e C. S. Pineda, "Exploring How the Attribute Driven Design Method is Perceived," March 2015.
- [32] J. Brooke, "SUS: A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, 1995.
- [33] S. Popic, V. Komadina, R. Arsenovic e M. Stepanovic, "Implementation of the simple domain-specific language for system testing in V-Model development lifecycle," *2020 Zooming Innovation in Consumer Technologies Conference (ZINC)*, pp. 290-294, 2020.
- [34] P. Escudeiro e J. Bidarra, "Quantitative Evaluation Framework (QEF)," *Conselho Editorial/Consejo Editorial*, p. 16, 2008.
- [35] G. Sasmito, L. O. M. Zulfiqar e M. Nishom, "Usability Testing based on System Usability Scale and Net Promoter Score," em *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 540-545.