

# **MARKETING FOR FUTURE**

## **Eddmi – Pass Platform**

**Luís Manuel Martins Ribeiro**

**Dissertation for the Master Degree in**

**Engenharia Informática**

**Area of expertise Arquiteturas, Sistemas e Redes**

Supervisor: António Cardoso Costa

Co-supervisor: Daniel da Silva Nogueira

**Jury:**

President:

Doutor Maria de Fátima Coutinho Rodrigues

Vowels:

Doutor Maria Goreti Carvalho Marreiros

Doutor António Manuel Cardoso da Costa



# Acknowledgments

This thesis represents the accomplishment of a big goal for me. I must admit that I was not alone because I had family and friends supporting me since the beginning, and also because along the way I crossed myself with people who in one way or another contributed to this work, and without whom it would not be possible.

So, I dedicate this space to those who gave their contribution to what I could accomplish my goal, expressing my special thanks

To Prof. António Costa, for his guidance, availability and constructive contribution in all the process.

To Eng. Daniel Nogueira for his availability and generosity revealed throughout these months of work. Thanks for the healthy discussions.

To Código Irreverente for allowing me to develop this project, especially to Filipe Lemos e César Brandão for all the support.

To my family for their unconditional support. In particular to my Mother for the constant support and encouragement.

To my daughter Matilde for always make me laugh at the end of a hard working day.

Finally, I would like to express my immense gratitude to Isabel, my wife, who is my main support and the main driver of my dreams. I thank her all the strength, patience and dedication.



# Abstract

In a highly competitive market companies know that having quality products or provide good services is not enough to keep customers "faithful". Currently, quality of products/services, location and price are fundamental aspects customers expect to get on every purchase, so they look for ways to distinguish companies. This can happen either in a strictly materialistic way or by evaluation of intangible metrics such as having his opinion appreciated or being part of a selected group of "premium" customers. Therefore, companies must find ways to value and reward its customers in order to keep them "faithful" to their products or services. Loyalty systems are one means to achieve this goal, however, due to its nature and how they are implemented, often companies end up having low acceptance, without achieving intended objectives.

In an era of technological revolution, where global average adoption of smartphones and tablets is 74% and 40% [Our Mobile Planet, 2014], the opportunity to reinvent loyalty systems reappears. Throughout this thesis a new tool, relying on the latest technologies and aiming to fulfill this market opportunity, will be presented. The main idea is to use ancient loyalty concepts, such as stamps or points cards, and transforms them into digital cards, to be used in digital wallets, introducing an innovative technology component based on Apple's Passbook technology.

The main goal is to create a platform for managing the card's life cycle, allowing anyone to create, edit, distribute and analyze the data, and also create a new communication channel with customers, improving the customer-supplier relationship and enhancing the mobile-marketing.

**Keywords:** Digital Cards, Loyalty, Passbook, Passwallet



# Resumo

Num mercado altamente competitivo, as empresas sabem que ter produtos de qualidade e prestar um bom serviço não é suficiente para manter os clientes “fiéis”. Atualmente, a qualidade dos produtos/serviços, a localização e o preço são aspetos básicos que o cliente espera obter em cada compra, de tal forma que procura maneiras de distinguir as empresas. Isto pode acontecer tanto do ponto de vista estritamente material, como através da avaliação de aspetos intangíveis tais como ver a sua opinião valorizada ou sentir que faz parte de um grupo restrito de clientes "premium". Assim sendo, as empresas devem encontrar novas formas de valorizar e compensar o cliente no sentido de o manter “fiel” aos seus produtos ou serviços. Os sistemas de fidelização são uma das armas usadas para atingir este objetivo, no entanto, devido à natureza desses sistemas ou à forma como são implementados, muitas vezes acabam por ter fraca adesão dos clientes e não atingem os objetivos pretendidos.

Numa era marcada pela revolução tecnológica, em que a média global de adoção de smartphones e tablets é 74% e 40% [Our Mobile Planet, 2014], surge a oportunidade de reinventar os sistemas de fidelização. Ao longo desta tese será apresentada uma nova ferramenta que assenta nas mais recentes tecnologias e visa preencher essa janela de oportunidade criada pelo mercado. A ideia principal é usar conceitos de fidelização antigos, tais como o cartão de carimbos ou o cartão de pontos, e transformá-los em cartões digitais, a usar em carteiras digitais, suportados por uma plataforma tecnologicamente inovadora assente na tecnologia Passbook da Apple.

O objetivo último é criar a plataforma para gestão do ciclo de vida dos cartões, permitindo criar, editar, distribuir e analisar os dados obtidos, possibilitando ainda criar um novo canal de comunicação com o cliente, tornando a relação entre ambos mais transparente e potenciando o *mobile-marketing*.

**Palavras-chave:** Cartões Digitais, Fidelização, Passbook, Passwallet



# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Scope .....	1
1.2	Project Presentation.....	2
1.3	Definitions .....	4
1.4	Organization Presentation .....	6
1.5	Thesis Organization .....	6
1.6	Problem Statement .....	7
1.7	Objectives.....	9
1.8	State of the Art .....	9
1.8.1	Passes and Passbook .....	9
1.8.2	Web Development.....	17
1.8.3	Security .....	18
1.8.4	Development Methodologies .....	19
1.9	Conclusion .....	20
<b>2</b>	<b>Analysis.....</b>	<b>21</b>
2.1	Requirements and Features .....	21
2.1.1	Functional .....	21
2.1.2	Non Functional.....	25
2.2	Specification .....	26
2.2.1	Processes and Models .....	26
2.2.2	Logical structure .....	28
2.2.3	Physical Structure .....	29
<b>3</b>	<b>Development .....</b>	<b>31</b>
3.1	Tools and Technologies .....	31
3.1.1	Technologies.....	31
3.1.2	Tools .....	32

3.2	Environment .....	39
3.2.1	Development .....	39
3.2.2	Staging .....	39
3.2.3	Production .....	40
3.3	Implementation .....	41
3.3.1	Documentation .....	41
3.3.2	Web platform .....	42
3.3.3	Push notification API .....	45
3.3.4	Web service API .....	47
3.3.5	Security .....	49
<b>4</b>	<b>Testing and Deployment .....</b>	<b>53</b>
4.1	Development .....	53
4.1.1	Unit .....	53
4.1.2	Integration .....	56
4.2	The solution .....	58
4.2.1	Functional .....	58
4.2.2	Acceptance .....	68
4.2.3	Usability .....	69
4.2.4	Security .....	70
<b>5</b>	<b>Conclusion .....</b>	<b>73</b>
5.1	Thesis Summary .....	73
5.2	Completed Objectives .....	73
5.3	Other Works .....	74
5.4	Limitations .....	76
5.5	Future Work .....	76
<b>6</b>	<b>References .....</b>	<b>77</b>

# List of Figures

Figure 1 - Pass Life Cycle .....	2
Figure 2 - Interactions between the client and the Pass Platform [Apple Inc. 2013a] .....	8
Figure 3 - Scrum development activities [Scrum Reference Card, 2014] .....	20
Figure 4 - Templates Use Cases .....	22
Figure 5 - Passes Use Cases.....	23
Figure 6 - Client/Server Model.....	27
Figure 7 - Platforms Structure Overview .....	27
Figure 8 - Typical Yii application workflow [Yii Framework, 2014b] .....	43
Figure 9 - Interactions between the client and the Push Notification API [Apple Inc. 2013a]...	46
Figure 10 - Web Service’s API Structure Overview .....	47
Figure 11 - Access rules example .....	50
Figure 12 - Login unit test .....	53
Figure 13 - XHProf output for the request contact .....	59
Figure 14 - XHProf output for the request sign up.....	60
Figure 15 - XHProf output for the request sign in.....	61
Figure 16 - XHProf output for the request create template .....	62
Figure 17 - XHProf output for the request update template .....	64
Figure 18 - XHProf output for the request create pass.....	65
Figure 19 - XHProf output for the request update pass.....	66
Figure 20 - XHProf Output for the request create issue .....	67
Figure 21 - XHProf Cookie, Get and Post outputs .....	67
Figure 22 - Nessus Vulnerability Scan Resume .....	70
Figure 23 – Screenshots of the mobile applications .....	75



# List of Tables

Table 1 - Passbook Supported Pass Types [Apple Inc, 2013b] .....	10
Table 2 - Passes Package Structure [Apple Inc, 2013b] .....	11
Table 3 - Top-Level Keys [Apple Inc, 2013b] .....	12
Table 4 - Lower-Level Keys [Apple Inc, 2013b] .....	12
Table 5- Pass services provider evaluation .....	14
Table 6 - Extensions for Chrome used on the staging environment.....	40
Table 7 - Pass's implemented features .....	42
Table 8 - Template's implemented features .....	42
Table 9 - Web Service API implemented methods .....	48
Table 10 - Features tested .....	54
Table 11 - Features not yet tested.....	54
Table 12 - Sign Up test cases.....	55
Table 13 - Login test cases .....	55
Table 14 - Create Pass test cases .....	56
Table 15 - 1 <sup>st</sup> Stage data for the integration test cases .....	57
Table 16 - 2 <sup>nd</sup> Stage data for the integration test cases.....	57
Table 17 - Contact Function Test Inputs .....	58
Table 18 - Sing Up Functional Test Inputs.....	59
Table 19 - Sign In Functional Test Inputs .....	60
Table 20 - Create Template Functional Test Inputs .....	61
Table 21 - Update Template Functional Test Inputs.....	63
Table 22 - Create Pass Functional Test Inputs .....	64
Table 23 - Update Pass Functional Test Inputs .....	65
Table 24 - Create Issue Functional Test Inputs .....	66
Table 25 - Template's implemented features updated .....	68



# List of Abbreviations

<b>AMP</b>	Apache MySQL and PHP
<b>API</b>	Application Programming Interface
<b>APNs</b>	Apple Push Notification Service
<b>CMS</b>	Content Management System
<b>CRUD</b>	Create Read Update and Delete
<b>CSS</b>	Cascading Style Sheets
<b>CSRF</b>	Cross-site Request Forgery Prevention
<b>DNS</b>	Domain Name Service
<b>DOS</b>	Denial of Service
<b>FTP</b>	File Transfer Protocol
<b>GPS</b>	Global Positioning System
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IDE</b>	Integrated Development Environment
<b>iOS</b>	iPhone Operating System
<b>IP</b>	Internet Protocol
<b>JSON</b>	JavaScript Object Notation
<b>LAMP</b>	Linux Apache MySQL and PHP
<b>MAMP</b>	Mac Apache MySQL and PHP
<b>MVC</b>	Model–View–Controller
<b>ORM</b>	Object-Relational Mapping

<b>OS</b>	Operating System
<b>PHP</b>	PHP Hypertext Preprocessor
<b>PKCS</b>	Public-Key Cryptography Standards
<b>QR</b>	Quick Response
<b>RAD</b>	Rapid Application Develop
<b>REST</b>	Representational State Transfer
<b>RBAC</b>	Role-Based Access Control
<b>SHA</b>	Secure Hash Algorithm
<b>SQL</b>	Structured Query Language
<b>SSL</b>	Secure Scket Layer
<b>UI</b>	User Interface
<b>URL</b>	Uniform Resource Locator
<b>WAMP</b>	Windows Apache MySql and PHP
<b>XSS</b>	Cross-site scripting

# 1 Introduction

## 1.1 Scope

In a highly competitive market, the image that is projected to customers is very important for all companies. Preserve the good customers is crucial to the financial health of a company, and for that it is necessary to use a marketing tool in order to keep the customers informed about the company latest news. But not all marketing should be made with the intention of selling, a big part of it should be made with the intention of keeping the customers "faithful to its origins" and make them loyal.

The relationship established with customers is one of the most effective forms of loyalty, because is very difficult to break, especially when the business relationship is maintained for some time and there are special conditions that a new supplier can hardly offer. Another way to achieve customer retention is by giving them a differentiating factor, which can be done in several ways. Innovating and introducing new products with different features, or giving new uses to old products, or providing advantages comparatively to competing products, is a good way to do it. However, the cost of innovation has to be justified, because otherwise the company risks of not penetrating the market. The difference can also be marked by quality, which is increasingly important due to the prices approach imposed by concurrency on the market. A customer who is satisfied and acknowledges that is served with high quality standards tends to become loyal and is more difficult for him to change his supplier, with the possibility of being less well served than is accustomed to.

The customer loyalty is not an easy or cheap process, but with the right tools that process can be a good experience for companies and customers. The scope of this thesis is to present this

tool as a web-based platform that uses a comprehensive set of tools, made available by Apple and other companies.

## 1.2 Project Presentation

The project describes a web-based platform to create, update, distribute and delete passes with Apple’s Passbook and Passkit framework, which are then distributed to mobile devices with iOS or Android operating system. In other words, the project consists in a platform able to manage the pass life cycle.

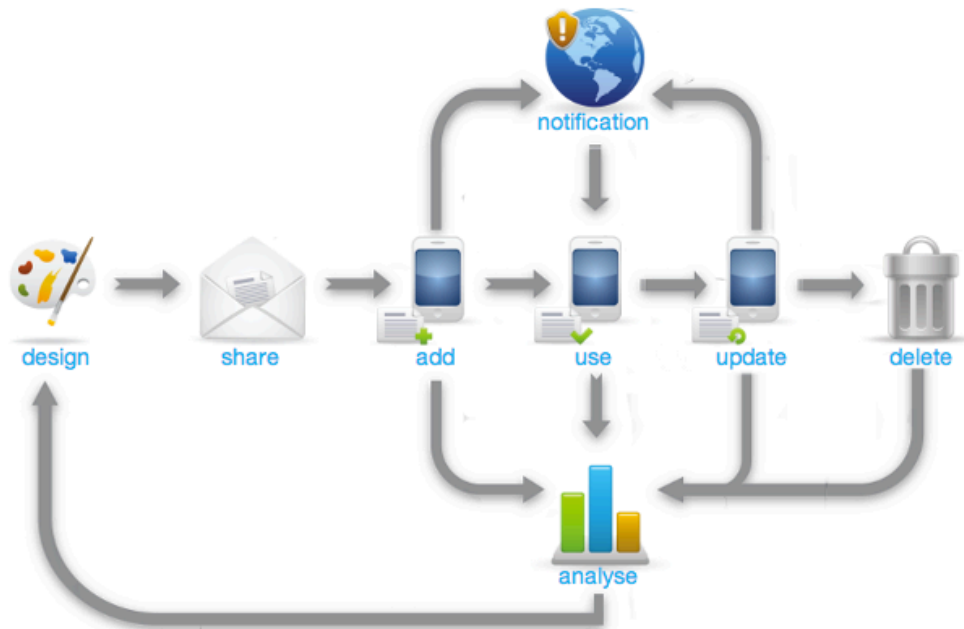


Figure 1 - Pass Life Cycle

The Passbook is an API, and also an App, that pretends to be a simple way for users to store all their passes. Since Passbook is time and location enabled, everytime the user wakes its device Passbook will check the relevance of passes regarding the time and user location to know if there is any relevant pass. If it exists, it will be displayed on the Lock Screen. In this way the user can receive useful and personalized messages base on the time or location. [Apple Inc, 2012]

The pass front is used to present the most important information. Normally it also includes a barcode that can store an encoded message with information such as a membership id, an event ticket number or even a location. On the other hand, the pass back is used to present

other information that is too extensive to be presented at the front, such as addresses. Passes also can be updated dynamically by way of the Apple Push Notification service, which lets service providers easily notify their users of important updates. [Apple Inc, 2012]

The passes are stored and used on Passbook app and can be distributed via email, the web or directly from an app. Passes can be extremely personalized with specific information to a single user, or generic information for any user. The passes design also can be personalized, being possible to change the background, color, text color, images, and all the contents.

Taking into account the Passbook and passes specifications, the project must be developed divided of three main parts:

- A web platform where users can create, update, distribute and delete passes, as well as get statistical information about their passes;
- A push notification Application Programming Interface (API) to notify all users when passes are updated;
- A web service to update all distributed passes available to all programmers who want to develop applications that use Pass Platform remotely.

The web platform must include a template designer with a visual preview of the templates, and all the required tools to manage the pass life cycle. It is intended that the platform use the state of the art web technologies to achieve the best user experience.

The push notification API will be used to update the passes to their latest version, and must be ready to support two Mobile Operating Systems: iOS and Android.

Regarding the web service for the integration of the Pass Platform with third part services, several technologies that can be used and will be analyzed in terms of best options.

The main goal of the Pass Platform is to become the main tool for small and medium companies producing digital marketing and interact with their customers.

## 1.3 Definitions

Along this document will be used a set of terms and concepts specific to the areas involved in the project, which for those not technically savvy, can be difficult to understand. Anyone who reads this report will have a basic understanding of what is presented, so in the section will be described definitions of some terms, and the remaining will be explained when concepts appear for the first time.

**2D Barcode** - 2-dimensional barcode uses geometrical patterns to store information. QR codes and Aztec codes are examples of 2D barcodes that Passbook uses.

**API** – API or Application Programming Interface is a set of tools for building software applications. APIs are designed for programmers to write applications that are consistent with the interface.

**Beacons** – A beacon is a small, wireless device, sometimes also called a 'mote'. When placed in a physical space, it broadcasts tiny radio signals to smart devices. [Estimote, 2014]

**Geo-fencing** - Through the use of GPS, geo-fencing sets the perimeter or boundaries of a certain location. For example, a notification can appear in a user's device if he is within 20 meters to the store.

**Geo-location** - The actual location or the positioning of a device.

**HTTP** - HTTP, or Hypertext Transfer Protocol, is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol that can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers. [The Internet Society, 1999]

**JSON** - JSON, or JavaScript Object Notation, is a lightweight, text-based, language-independent data interchange format. JSON defines a small set of formatting rules for the portable representation of structured data. [The Internet Society, 2006]

**Passbook** - Apple's application for the iOS (6 and 7) that allows users to store coupons, tickets, cards, and passes on their phones.

**Passes** - Digital representation of information that might otherwise be printed on small pieces of paper or plastic. Can contain images and a barcode, and can be updated using push notifications. Passes can be loyalty, visit and store cards, event tickets or coupons.

**Push Notifications** - Messages or alerts that are sent in real time directly to users devices through Internet connectivity. Users are notified of new information that is controlled by the app and receive messages that are pushed to their devices. In Passbook, push notifications include receiving alerts when some of the fields change.

**REST** – REST, or Representational state transfer, is an architectural style consisting of a coordinated set of constraints applied to components, connectors, and data elements, within a distributed hypermedia system. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements. When applied to Web Services, takes the designation of RESTful API's. [Fielding and Taylor, 2002]

**Server Redundancy** - A redundant server is a backup server running on the network that instantly can take over if the primary server fails. Other possible scenario is that both are working and sharing duties. In this case if one of them fails, the other takes over full duties until the both are back on line.

**UI** – UI stands for User interface and is a platform where users essentially interact with a machine. Apple's iOS 7 is an example of a user interface as it serves as the operating system for its devices.

## 1.4 Organization Presentation

*Código Irreverente* is a web based company that offers a comprehensive service of knowledge and fostering for business based on a cross platform, with the main goal of enhancement each client individually by taking companies marketing solutions to the next level. Although it is a recent startup, the company has professionals with extensive experience and capable of adapting the most recent technology and carry out high complexity projects.

Considering there has been a drastic shift from desktop to mobile, it makes sense for businesses to offer mobile solutions to their customers, and that is where *Código Irreverente* is focused, with the slogan “welcome to the mobile marketing revolution” and a focus on the development of digital marketing platforms that rely on backend and frontend development for the various existing mobile operating systems.

## 1.5 Thesis Organization

This thesis is divided into six main chapters, divided by a logical sequence of problem presentation, context, analysis and implementation.

### **Introduction**

In this initial chapter is presented the project, the organization and the definition of a set of terms or technologies that will be used in this thesis. In is also is described the scope and the main contributions of this thesis.

### **Context**

This chapter describes the problem, framed in a broader context, with the description of all aspects related to this work. It is also presented the state of art for this type of platform. The goals of the project are also presented in this section.

### **Analysis**

This chapter presents the project requirements, both functional and non-functional, and main solution features. It is also presented the project specification, including the processes and models and the logical/physical structures.

## **Development**

Divided into three parts, this chapter describes all the technical details related with project implementation. In the first and second sections are presented the development environment and the tools/technologies used. In the third section the effective implementation during development is detailed.

## **Testing and Deployment**

This chapter presents the tests performed on the platform, both in terms of development and of effective use. Among others, the results from integration, security and acceptance tests are described and explained.

## **Conclusion**

Finally, the conclusion chapter starts by a summary of the solution, as well as the achievements and limitations detected. Additionally, it describes some ideas on future development.

# **1.6 Problem Statement**

The exponential growth of mobile devices, combined with their increasing processing power, presents an opportunity for companies to produce more and better mobile marketing solutions. This is an opportunity most technological companies do not want to let go, as such have developed tools that improve the mobile marketing.

Apple presented one off those tools in iOS 6 that is a combination of an app and a framework. The app is called Passbook, and basically is a digital wallet for passes, coupons, tickets, and other types of intelligent cards. The framework is named Passkit and provides access to the user's pass library on iOS devices.

For consumers it is the Apple's usual deal, nice and smooth to use, with a good user interface (UI) and user experience. After the user imports a pass to the Passbook app, which can be done by clicking a link or scanning a QR code on a poster or flyer, the pass is ready to use. There are a couple of great features, such as geo-fencing, which allows passes to present a notification pop up on the lock screen when it is within a certain distance from the venue where the pass can be used, or the possibility to limit the pass utilization adding an expiring date, very useful for event tickets or coupons. So, with Passbook the consumers will be

themselves free of plastic cards and coupons that most of the time end up in trash, and can have of a digital wallet on their smartphone.

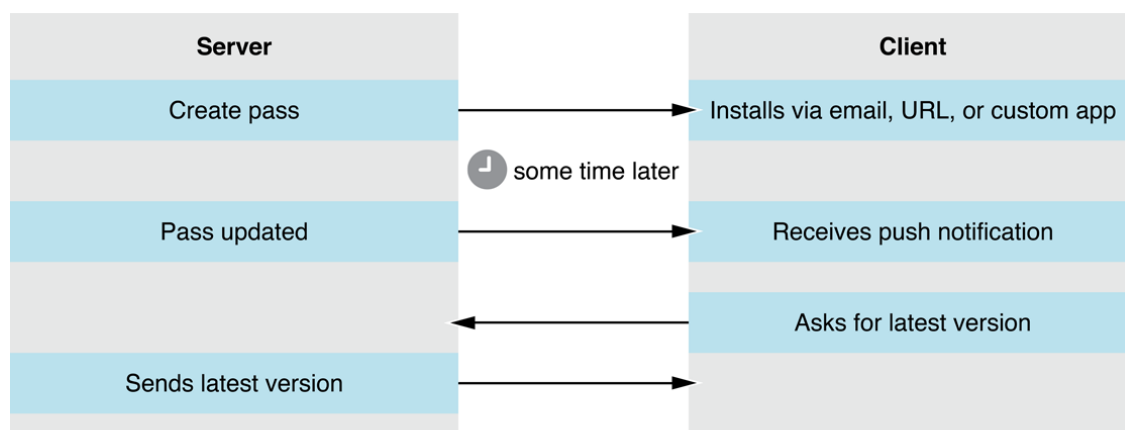


Figure 2 - Interactions between the client and the Pass Platform [Apple Inc. 2013a]

For consumers is easy to use and it brings a great set of benefits, but for the business side the process is not so simple, and it is where the problem stands. Some of the difficulties with Passbook derive from the need to be a developer to create, distribute and manage passes, and unlike most apps and platforms focused on the end user, the Passbook platform is mostly directed to merchants (business owners, marketing officers and others) that typically are not developers.

The business side is quite complex because, although Apple handles some parts, such as the phone identity and the pass updates with push notifications, the companies wanting to use this technology must take care of the rest, such as the pass design, with certain restrictions imposed by Apple, the content and updates, the passes redemption and also the signature and validation with a valid pass type ID. To do this, companies need to have a team of developers or they need to outsource it, which can be relatively costly. For most Portuguese companies, or even worldwide, these are two scenarios that are not even considered.

Considering the presented scenario, one may find a reason or two for low adoption rates on this technology, and that is where the Eddmi Pass Platform has an opportunity. Eddmi aims to be the bridge between Apple and the companies that want to use passes, offering a tool with the functionalities required for creating, distributing and updating passes, which can be used by anyone without programming skills.

It is also important to mention one of the benefits of Passbook is that the company does not need an app. If you are a restaurant owner, or a theatre owner, or a shop owner, but do not want to invest time, money and effort building your own app, and then convincing consumers to install it, you can simply use Eddmi and create passes that will fit in the standard app in every iPhone with iOS6 or later. Besides that, with the rise of Passbook acceptance, applications are emerging capable of using Passbook passes for all major mobile platforms, such as Android, Blackberry and Windows Phone, so you do not need to worry if consumers have an iPhone.

At this point, Passbook is in an accelerated adoption phase, and the companies that integrate it with their business, besides being at the front of mobile marketing trends, are already getting benefits.

## **1.7 Objectives**

With the completion of the master's thesis it will be implemented a secure prototype almost 100% functional, excluding minor aspects such as final layout, details of interaction and design. Additionally, other important goal is to conduct usability testing with various types of target users, and the correct use of good practices in safe software development, as well a testing process oriented to information security.

The main goal of the Eddmi Pass Platform is to become the main tool for small and medium companies producing digital marketing and interacting with their customers.





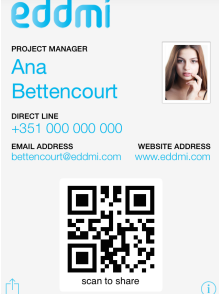
## **1.8 State of the Art**

### **1.8.1 Passes and Passbook**

The Passbook and the passbook API were announced with iOS 6 on June 12 at the 2012 Apple Worldwide Developers Conference and released on September 19 of the same year.

The Passbook API and passes have a set of constraints that must be respected by developers. For start, the card types to be used are limited to five, and each of these types has a defined structure that must be respected, otherwise the pass will not be validated. [Apple Inc, 2012] In Table 1 we can see the types of existing passes and how they can be used.

Table 1 - Passbook Supported Pass Types [Apple Inc, 2013b]

Pass Type	Description	Example
Boarding Pass	Can be used to represent a ticket for boarding in some type of transportation.	 <p>An example of a boarding pass for Oceanic Airlines, flight C13 from Sydney (SYD) to Los Angeles (LAX). It includes flight details like departure time (2:25 PM), flight number (815), class (Coach), and date (7/22). The passenger name is Benjamin Linus. A QR code is visible at the bottom.</p>
Coupon	Can be used for merchants offering a % discount, a monetary value or promotional item.	 <p>An example of a coupon for Fashion Boutique offering a 25% discount. The promotion is valid from Primavera/Verão until 31/03/2014. A QR code and the number 03100 are shown at the bottom.</p>
Event Ticket	Can be used as a ticket to give access to an event in an particular time and venue.	 <p>An example of an event ticket for Web Wednesday. The ticket is for Percy PassKit, status Unpaid, dated 11 September, 2013, from 6:30pm to 9:30pm. A QR code and the number #579557969868115 are shown at the bottom.</p>
Store Card	Can be used to represent a user's account at a store. Can be used as points card, pre-paid card or stamps card.	 <p>An example of a store card for Coffee. It shows a progress bar for coffee cups (8/10) and identifies the client as Ana Bettencourt. A QR code and the number 01105 are shown at the bottom.</p>
Generic	Can be used for any purpose not fitting above categories, such as a membership or business cards.	 <p>An example of a generic business card for eddmi. It lists Ana Bettencourt as the Project Manager, along with contact information: Direct Line (+351 000 000 000), Email Address (bettencourt@eddmi.com), and Website Address (www.eddmi.com). A QR code and the text 'scan to share' are shown at the bottom.</p>

All types of passes have these important aspects in common:

1. Top header: A top strip containing the company name and logo. When users have a full deck of passes in Passbook, this header area is the part of the pass they will see. Although it includes essential info to the user, because they can easily find the pass they need, the logo and the logo text are not required info, however are recommended.
2. Main content area: This section contains the main information of the pass. Usually shows the most important information of the pass, and appears once the pass is open. Unlike the top header, each pass type has different styling for this section.
3. Additional info: The third section is important because it is used to present additional information, usually not as significant as the main content.
4. Barcode: A two-dimensional barcode, (Azetec, QR or PDF417) which is easier to read for image-based scanning devices (like the iPhone). It contains encoded information, such as the pass number, or an URL to download, which can be easily transferred to other systems by scanning it with a barcode reader.

Table 2 - Passes Package Structure [Apple Inc, 2013b]

File	Description
background.png	Image displayed as the background of the front of the pass. (Just used by Event Ticket Passes).
footer.png	Image displayed on the front of the pass near the barcode. (Just used by Boarding Passes).
icon.png	The pass' icon. This is displayed in notifications and in emails that have a pass attached, and on the lock screen.
logo.png	Image displayed on the front of the pass in the top left.
manifest.json	This file describes the list of files inside the pass and the SHA1 checksums of each of those files. Each key is the path to a file, relative to the top level of the bundle. All the files must appear in the manifest, except for the manifest itself and the signature.
pass.json	A JSON dictionary with Top-Level Keys that defines the pass.
signature	The manifest.json file PKCS#7 signature. It is generated using an Apple-provided certificate.
strip.png	Image displayed behind the primary fields on the front of the pass. (Used by Coupon and Store Cards Passes).
thumbnail.png	An additional image displayed on the front of the pass, for example, on a membership card. (Used by Generic and Event Ticket Passes).

Focusing on the more technical aspects, the passes are stored on disk as a zipped package with the pkpass file extension, and the top level of the package contains the files show in Table 2. [Apple Inc, 2013b]

With the description in above table, it is easy to understand the meaning and function of each file that makes up the Package Structure. However, the file pass.json is just a JSON dictionary with Top-Level Keys that defines the pass. So, in order to better understand the function of this file, Table 3 describes all the Top-Level Keys and its meanings.

Table 3 - Top-Level Keys [Apple Inc, 2013b]

Key	Description
Standard Keys	Information that is required for all passes.
Associated App Keys	Information about an app that is associated with a pass.
Companion App Keys	Information about a pass provided for a companion app.
Expiration Keys	Information about when a pass expires or if it still valid.
Relevance Keys	Information about where and when a pass is relevant.
Style Keys	Indicates the pass' type.
Visual Appearance Keys	Visual styling and appearance of the pass.
Web Service Keys	Information about the web service used to update passes.

Of all keys presented, only the Standard keys are required, being all the others of optional use, however If the web service is provided, is also required an authentication token to communicate. Regarding the signature, possible it is the most interesting and challenging part of creating a pass. The signature is generated with a certificate in PEM format, which developers must get with his Apple Developer ID, and with OpenSSL. Each pass type requires a specific certificate.

Table 4 - Lower-Level Keys [Apple Inc, 2013b]

Keys	Description
Pass Structure Dictionary Keys	Keys that define the structure of the pass. These keys are used for all pass styles and partition the fields into the various parts of the pass.
Beacon Dictionary Keys	Information about a location beacon. Available in iOS7/8.
Location Dictionary Keys	Information about a location.
Barcode Dictionary Keys	Information about a pass' barcode.

Beyond the Top-Level Keys, the file pass.json also includes Lower-Level Keys and Keys Field Dictionary. While the Top-Level Keys are more relevant for developers, the Lower-Level Keys

are the most important elements for the end user. Although some are the Top-Level Keys that set the graphic style of the pass, it is the Lower-Level Keys that contain the information that will be presented. [Apple Inc, 2013b] Regarding the Field Dictionary Keys, they are used by the Lower-Level Keys to define data types. For example, if a Lower-Level Key is an auxiliaryField, among others, it will have a key, label and value. However, if the value is of the type date or time, it may have associated the Field Dictionary Keys short, medium or long. Table 4 presents the Lower-Level Keys.

As mentioned before, some of the information is of optional use, however there is a minimum amount of meta-information that must be provided.

1. `formatVersion`: The file format version. For now there is only one version, so it will be used the version 1.
2. `passTypeIdentifier`: The pass type identifier. It is similar to the bundle identifier of an iOS app. Exp: `pass.com.codigoirreverente.generic`. The last part identifies the pass type and the middle part the company that issued it.
3. `serialNumber`: The pass serial number. It can be generated in many ways, and it can be numeric, such as 1000, or a combination of numbers and letters, such as "CPS61E35".
4. `teamIdentifier`: Is a unique 10-character identifier that Apple assigns to iOS developers.
5. `organizationName`: The name of the entity that issues the pass.
6. `description`: A short description of the pass.

With the release of the iOS 7, the passbook concept underwent some changes, mostly related with the design, although some also very important such as the incorporation of the protocol iBeacons, which uses the Bluetooth Low Energy (4.0) and beacons to communicate with user devices, the option of void passes and the auto expiration date. However, at the official keynote presentation none of these features were mentioned.

More recently, with the launch of iOS 8, Apple introduced new features for Passbook, including the support for a new type of pass.

Passbook already supported five types of passes: store cards, boarding passes, tickets, coupons, and generic cards that could be used for membership cards or business cards, but from this point Passbook also supported credit cards. [Apple Inc, 2014a]

For safety reason, this type of pass requires extra steps, which the old types did not need. As such, users must add the credit or debit card from their iTunes account to Passbook by entering the card security code, after which adding a new card on Passbook is allowed. Users need only to use their iPhone camera to capture the card information, or simply manually type it. The first card added becomes the default payment card, but at any time users can go to Passbook and change the default settings, or simply select the card they want to use. [Apple Inc, 2014a] This Passbook new feature and the Passkit framework provide the support APIs to Apple’s payment service called Apple Pay, also launched in iOS 8. [Apple Inc, 2014b]

Table 5- Pass services provider evaluation

	<b>Captain Pass</b>	<b>Pass-dock</b>	<b>Pass-source</b>	<b>PassSlot</b>	<b>PassKit</b>	<b>Urban Airship</b>
API	3	3	2	4	4	5
Documentation	2	3	3	3	4	5
Code Samples	0	0	0	3	3	3
Design Preview	5	3	0	3	4	3
Design Wizard	0	3	0	3	0	0
Overall Design	5	4	3	4	4	3
Overall Usability	5	4	4	4	4	3
iOS 6	5	5	5	5	5	5
iOS 7	5	5	0	5	5	5
Android	5	5	5	5	5	5
iBeacons	5	5	0	5	5	5
Geo-fencing	4	3	0	4	5	3
Allows Tests	3	0	0	0	0	3
Multilanguage	0	0	0	0	5	0
Pass Types	5	5	5	5	5	5
<b>TOTAL</b>	<b>52</b>	<b>48</b>	<b>27</b>	<b>53</b>	<b>58</b>	<b>53</b>

There are many companies that provide platforms for the design and distribution of digital passes for the Passbook application. Over time, some of them do not update its platform, as when changes were released with iOS 7, but most keep upgrading its platforms with major features such as geo-location and geo-fencing with GPS and Beacons. Table 5 presents the worldwide most relevant service providers for the Passbook platform, including an evaluation of some of the most important features (0-5 scale).

There are more service providers available, but the ones here presented are the most important, due to the amount of services offered or because they were quick to develop platforms using the Passbook API.

The evaluation was made using the 0-5 scale, where 0 means the service does not provide the feature in question, and 5 means that the service supports the feature, being one of the best. The evaluation is based in the author's personal review and may be different for other analysts.

### **Captain Pass**

Captain Pass his the most recent pass designer released. It is a good service with a nice pass preview, being the best in the evaluation. This service has a user-friendly interface and a good dashboard providing useful information.

In terms of the supported features, this service provides all the Passbook features, including the ones presented in iOS 7. The feature deserving to be highlighted is the pass test. This feature allows the user to send an email with the template that is being developed, in order to see the template in a real device. It also provides a useful API for developers, but with some lack of documentation, presenting only some common features.

### **Pass-dock**

Pass-dock is easy to use, with arrows that guide the user along the pass creation process. Everything is segmented into parts and it also has a real time simulator showing how the pass would look.

This service also has an API for developers, but the documentation is very limited, both for normal users and developers. The feature to highlight most in this service is the wizard, through which the user can build a template in a very simple way.

### **Pass-source**

Pass-source was one of the first pass services provider on the market. It has a lot of templates for users to choose, but the features are limited. In terms of designing a pass, it is difficult for the average user to understand, mostly because there is no visual editor.

This service also has an API for developers, but at this time only produces passes for the Passbook for iOS6 and Android, both on the web platform and by API calls.

### **PassSlot**

PassSlot was designed for developers and requires an iOS developer account in order to use it in all its potential. For developers it is a good tool for implementation, but for regular customers and companies without developer knowledge it is not as good.

This service has scale good API that supports all the features available on the web platform, but without a corresponding documentation. Although the platform has been designed for developers, this service also has a wizard to simplify template design.

### **PassKit**

PassKit is the service that received the best rating in the evaluation. The PassKit designer is easy to use, with everything organized and with a real time preview of what the Passbook pass would look like on the desired device, so it is one of the most appealing and user friendly creators available.

This service also has an API for developers, with relatively good documentation, and supports all the Passbook features, including beacons and notification messages. Regarding the documentation, it is adequate both for normal users and developers.

One of the features to deserve more attention is the multi-language support. This feature allows the user to create passes in several languages, with texts in several languages, as well as with different images for each of them. Of all the services evaluated, this is the only one with such functionality.

### **Urban Airship**

Urban Airship Wallet, formerly known as PassTools, is an easy to use service, with concise descriptions of the relevant pass creation steps. Like PassSlot, it seems to be a service more oriented for developers and its simulator is not as accurate as some of the other services, even so still a good choice to help users visualize their passes.

In terms of support, this service supports all major features and has the best API and documentation, including lots of support, both for normal users and developers. Besides the API and documentation, another feature deserving to be highlighted is the pass test. Like in Captain Pass, this feature allows the user to send an email with the template in development, in order to be seen in the real device.

### **1.8.2 Web Development**

With regard to web development, it can be divided into two parts: frontend and backend development.

Regarding the frontend development, there seems to be a global consensus that JavaScript is the best client-side scripting language. Although JavaScript also can be used as a server side language with CommonJs or Node.js, JavaScript has been designed to be used as a client-side script and to allow easy integration with browsers to manipulate DOM elements.

Besides JavaScript, HTML and CSS are the other technologies that all web applications use. HTML version 5 is used as the application markup language. CSS version 3 is used for styling the application. The release of the latest versions has introduced some substantial changes in the way those technologies are used. HTML has acquired some new tags and media functions, while CSS also got new features, including transitions that can be used to replace JavaScript functionalities.

If about frontend things are standardized, in the backend side it is not the same due to a lack of consensus about the best language to use. Thus, there is a wide range of server-side languages used in web applications development, including Java, PHP, .NET, Python and Ruby. It is such a dynamic domain that all those languages constantly appear as the top most rated languages, as well as much required on job offers. [Tiobe Software, 2014]

Concluding, one can say that frontend development is more normalized, and that backend development is still in a stage where several technologies coexist, with no consensus on what is the best solution.

### 1.8.3 Security

Although current frameworks already have security mechanisms against major threats, at the end all comes down to developer's implementation. To ensure high levels of web application security it is essential to apply good practices during development. Confidentiality, integrity, availability, authenticity, accountability, non-repudiation, access control and fault-tolerance are requirements that a solution must implement in order to be secure, even when attacked. On the other hand, there is also a set of common errors/practices to be avoided. With the purpose of being significant in this area, in 2014 the IEEE Computer Society created the IEEE Center for Secure Design, aiming to change the focus in security from finding bugs to identifying common design flaws. [Arce, Clark-Fisher, Daswani, DelGrosso, Dhillon, Kern, Kohno, Landwehr, McGraw, Schoenfield, Seltzer, Spinellis, Tarandach, and West, 2014]

Security experts from various areas contributing to the Center concluded there are 10 common security flaws that must be avoided in all software developments, giving relevant suggestions on how to avoid each of them. The flaws are the following:

1. Flaw: Assume trust.  
Trust must be earned, so never assume it.
2. Flaw: Bad authentication mechanism.  
Use an authentication system that cannot be bypassed or corrupted.
3. Flaw: Bad authorization system.  
Make sure that you only authorize after the authentication process is concluded.
4. Flaw: Mix data and control instructions.  
Strictly separate data and control instructions, and never process control instructions received from untrusted sources.
5. Flaw: Non-validated data.  
Define an approach that ensures all data are explicitly validated.
6. Flaw: Bad use of cryptography.  
Apply cryptography correctly using well-known algorithms, not created by you.
7. Flaw: Bad handling of sensitive data.  
Identify sensitive data and how they should be handled.

8. Flaw: Do not consider the users.

Always consider the users and design a system that can be configured and used in a secure manner, with easy-to-use, intuitive interfaces and sufficiently expressive, but not excessive, security controls.

9. Flaw: Bad integration of external components.

Assume that incoming external components are not to be trusted until appropriate security controls have been applied, and understand how integrating external components changes your attack surface.

10. Flaw: Do not foresee future changes.

Keep in mind that all applications have changed to objects or actors one time or another. The design must be flexible enough to allow these changes.

Summarizing, any application being developed must take into account these safety rules, in order to be considered minimally secure and safe.

#### **1.8.4 Development Methodologies**

The development methodologies currently in vogue are “agile”, proposing alternatives to traditional project management. Basically an agile methodology is a collection of methods for software development that promotes adaptive planning, continuous development, early delivery, and supports rapid and flexible response to change. The concept is based on making small increments to software and delivering it to the customer as soon as possible. This allows the team to obtain more feedback during the development process, hopefully making the software more robust. Moreover, it also promotes interaction between stakeholders in the development process, allowing the requirements to be changed according to client’s actual needs. [Scrum Reference Card, 2014]

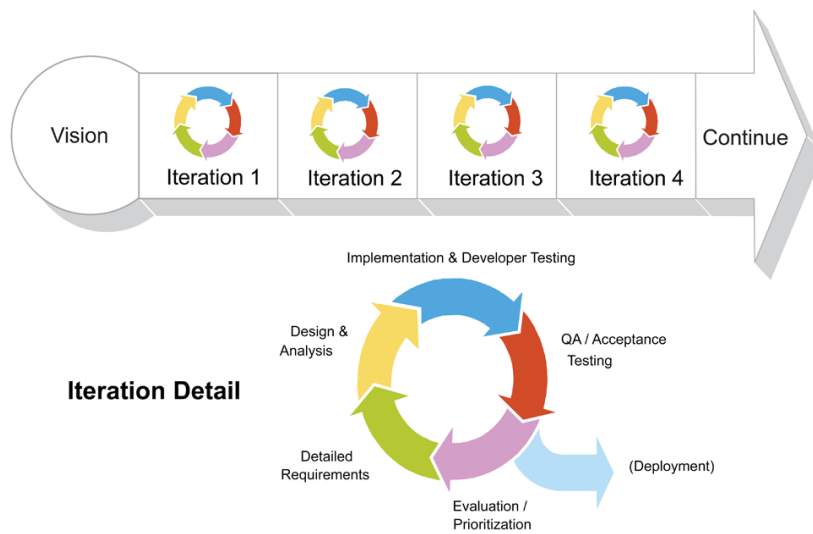


Figure 3 - Scrum development activities [Scrum Reference Card, 2014]

Within agile methodologies, Scrum is probably the most used one. Scrum provides an alternative to traditional Waterfall development process, which relies on a perfect understanding of the product, requiring a complete framework with structured roles, meetings, rules, and artifacts for teams to adapt their development processes. [Scrum Reference Card, 2014] Figure 3 presents the Scrum approach, aiming to replace the Waterfall one.

In Scrum there are only three roles: Product Owner, Team and Scrum Master. The responsibilities of the traditional project manager role are split up among these three.

Agile is considered a good methodology because it allows teams to gather requirements while developing the solution, making it less likely to disturb the development progress. As the team's work cycle is limited to two weeks, the stakeholders have chances to adjust the releases in order to be successful.

## 1.9 Conclusion

Although there is a wide range of Passbook services available, none of them are truly designed for the end user, who has no development knowledge. Some of them are exclusively targeted to developers, while others have a very complex creation process that may confuse users. It may be concluded that there is room for a service intending to bridge this gap. This is the challenge that Eddmi aims to solve.

# 2 Analysis

## 2.1 Requirements and Features

### 2.1.1 Functional

#### 2.1.1.1 Overview

The platform must have at least two groups of functionalities: templates and passes. Some of the platform's features will require functionalities that can be implemented by using well-known third party services for a successful implementation.

The platform must also have a responsive design, using state of the art technologies, and include an access control process where users must identify themselves using username/email and password. Only registered and authenticated users will access the system functionalities.

#### 2.1.1.2 Templates

In this section are presented the use cases for the features of templates' group. In each use case is described the feature's goal, the preconditions, the conditions for success or failure and the required actors. In Figure 4 is presented the overview of the use cases.

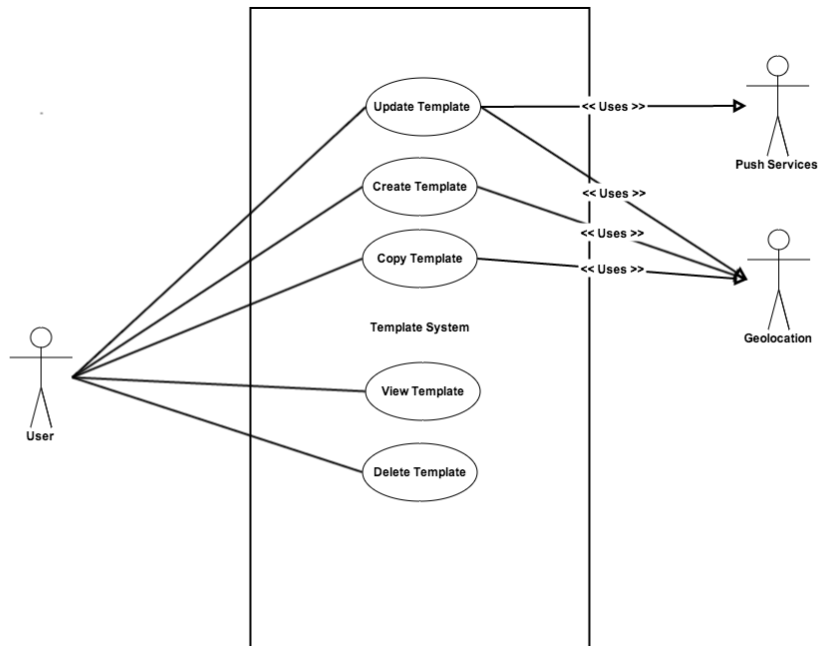


Figure 4 - Templates Use Cases

### Use Case Create Template

Objective: Create a Template.  
 Preconditions: The user must be logged.  
 Success Conditions: The user inserts valid data.  
 Fault conditions: The user inserts invalid data.  
 Actors: User, Platform, Geocoding Service.

### Use Case Update Template

Objective: Update a Template.  
 Preconditions: The user must be logged;  
 Must exist at least one template to update.  
 Success Conditions: The user inserts valid data.  
 Fault conditions: The user inserts invalid data.  
 Actors: User, Platform, Geocoding Service, Push Service.

### Use Case Copy Template

Objective: Copy a Template.  
 Preconditions: The user must be logged;  
 Must exist at least one template to copy.  
 Success Conditions: The user inserts valid data.  
 Fault conditions: The user inserts invalid data.  
 Actors: User, Platform, Geocoding Service, Push Service.

### Use Case View Template

Objective: View a Template.  
Preconditions: The user must be logged;  
Must exist at least one template to view.  
Success Conditions: Exists a template created.  
Fault conditions: No template created.  
Actors: User, Platform.

### Use Case Delete Template

Objective: View a Template.  
Preconditions: The user must be logged;  
Must exist at least one template to delete.  
Success Conditions: Exists a template created.  
Fault conditions: No template created.  
Actors: User, Platform.

### 2.1.1.3 Passes

This section describes the use cases for the features of the passes group. Like in the templates' group, the use cases include the feature's goal, the preconditions, conditions for success or failure, and the required actors. The Figure 5 shows the use case overview.

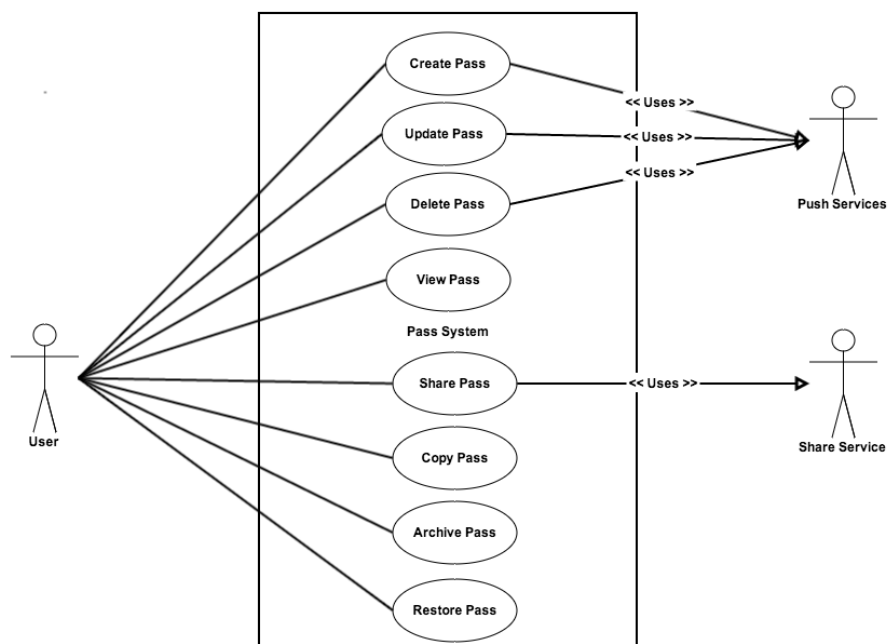


Figure 5 - Passes Use Cases

### **Use Case Create Pass**

Objective: Create a Pass.  
Preconditions: The user must be logged;  
Must exist at least one template.  
Success Conditions: The user inserts valid data.  
Fault conditions: The user inserts invalid data.  
Actors: User, Platform.

### **Use Case Update Pass**

Objective: Update a Pass.  
Preconditions: The user must be logged;  
Must exist at least one pass.  
Success Conditions: The user inserts valid data.  
Fault conditions: The user inserts invalid data.  
Actors: User, Platform.

### **Use Case Delete Pass**

Objective: Delete a Pass.  
Preconditions: The user must be logged;  
Must exist at least one pass archived.  
Success Conditions: Exists a pass archived.  
Fault conditions: No pass to delete.  
Actors: User, Platform, Push Service.

### **Use Case View Pass**

Objective: View a Pass.  
Preconditions: The user must be logged;  
Must exist at least one pass created.  
Success Conditions: Exists a pass created.  
Fault conditions: No passes created.  
Actors: User, Platform.

### **Use Case Share Pass**

Objective: Share a Pass.  
Preconditions: The user must be logged;  
Must exist at least one pass created.  
Success Conditions: Exists a pass created.  
Fault conditions: No passes created.  
Actors: User, Platform, Share Service.

### **Use Case Copy Pass**

Objective: Copy a Pass.  
Preconditions: The user must be logged;  
Must exist at least one pass to copy.  
Success Conditions: The user inserts valid data.  
Fault conditions: The user inserts invalid data.  
Actors: User, Platform.

### **Use Case Archive Pass**

Objective: Archive a Pass.  
Preconditions: The user must be logged;  
Must exist at least one pass to archive.  
Success Conditions: Exists a pass to archive.  
Fault conditions: No pass to archive.  
Actors: User, Platform.

### **Use Case Restore Pass**

Objective: Restore a Pass.  
Preconditions: The user must be logged;  
Must exist at least one pass archived.  
Success Conditions: Exists a pass archived.  
Fault conditions: No pass archived.  
Actors: User, Platform.

## **2.1.2 Non Functional**

### **2.1.2.1 Acceptance requirements**

1. The system usability must be accepted by more than 50% of beta users.
2. The platform must implement at least a feature suggested by users.

### **2.1.2.2 Documentation requirements**

1. The system must provide a helper for the pass designer.
2. The platform must have adequate API documentation.

### **2.1.2.3 Interface requirements**

1. The system must have an intuitive usability.
2. The system must have responsive design in all web pages.

#### **2.1.2.4 Performance requirements**

1. The system must be transparent to users.
2. The system must be independent of all the major systems and browsers.
3. The system must provide pagination when presenting large amounts of data.

#### **2.1.2.5 Security requirements**

1. The system must guarantee data protection from unauthorized access.
2. The system must ensure its integrity in case of accidental or malicious attacks.
3. All communications between system and third party services must be encrypted using the https protocol.

## **2.2 Specification**

### **2.2.1 Processes and Models**

The preferential model to use on this project is the client/server architecture. The client/server model describes the way communications between clients (service consumers) and server (service provider) happens, assuming some principles.

Clients and servers have well defined interfaces and functions and can be implemented by software modules, hardware components, or a combination thereof. Each client/server association is established between two functional modules when the client module initiates a service request and the server model decides to answer the service request.

A crucial feature of the client/server model is the way information is exchanged between clients and server. This exchange is strictly through messages typically interactive, so no information is exchanged through global variables. The client request and additional information is placed into a message sent to the server, and server's response is another message sent back to the client. This is the basic model, however there are some exceptions, such as message queuing systems allowing clients to store messages on a queue to be picked up later asynchronously by the server.

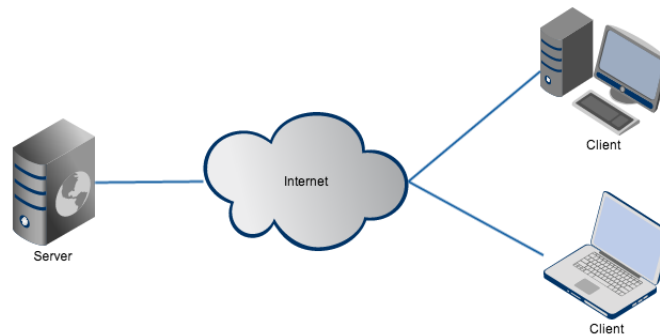


Figure 6 - Client/Server Model

Normally clients and servers are separate machines connected through a network, like presented in Figure 6, however, conceptually clients and servers may also run on the same machine. In this project it is intended to use a distributed client/server system, where clients and server are located in different machines.

Since client/server service requests are real-time messages exchanged through network services, and although this increases the appeal for this model because of features like flexibility and scalability, it also introduces technical issues like portability, interoperability, security, and performance, aspects that must be taken into account in server implementation.

Regarding processes, as mentioned before, the platform will require some processes that can be implemented by using third party services. In Figure 7 is presented the type of services required, followed by a small description of each type of service.

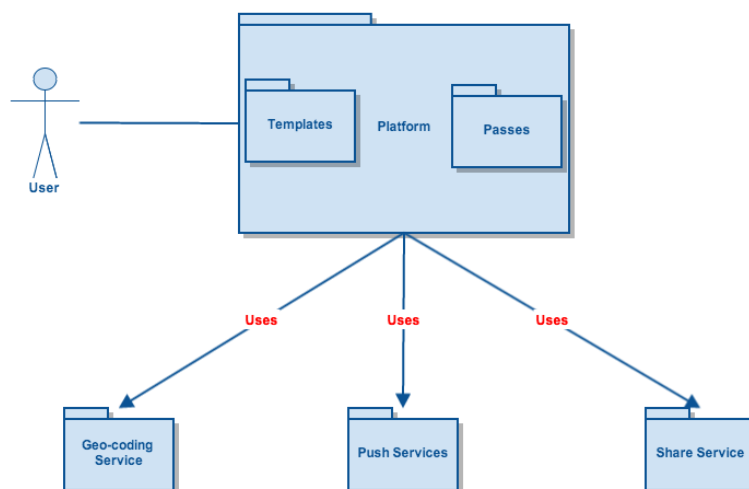


Figure 7 - Platforms Structure Overview

## **Geocoding**

The Geocoding Service will be used to convert addresses into GPS coordinates. These coordinates can be used to activate some pass features such as geo-fencing, geo-location and beacons.

## **Push**

The Push Service will be responsible for pushing updates to the passes every time they experience some change.

## **Share**

The Share Service will allow the end user to share passes in social networks.

The minimum requirements for all services imply they must be well known, successfully used and easy to implement.

### **2.2.2 Logical structure**

The platform's logical structure will be based on the Model-View-Controller (MVC) design pattern, also relevant as software architecture. [Wang, 2011]

The MVC pattern aims to separate the application logic from the user interface, allowing independent development, testing and maintenance of each part. The separation of concepts is recommended so that developers can more easily change each part without affecting others. This makes possible the creation of stable applications, avoids repeated coding, encourages Rapid Application Develop (RAD) and saves developers' time.

The MVC goal is to split the development process in parts, assigning a role to each part. Below can be seen the role each part takes in the process. [Wang, 2011]

**Model:** The Model represents the application information (data) and the business rules (behavior). It is responsible for responding to requests with instructions to change state and to requests for information about its state. Usually, the first ones are controller requests and the second ones view requests.

**View:** The View contains elements of user interface, such as text or form inputs, and usually renders the model into a form suitable for interaction. For different purposes, there can exist

multiple views for a single model, but typically a viewport has a one to one correspondence with a display surface and knows how to render it. The view also can contain static elements.

**Controller:** The Controller is what glues Model and View together. It receives inputs from the view and creates a response involving model objects on the process. In other words, a controller accepts input from user and instructs model and viewport to perform actions based on that input.

MVC is a simple concept, however it can bring benefits for the development process. Among them are the clarity of design, where public methods in the model stand as an API for all commands available to manipulate data and state, and the possibility of having multiple views, allowing to render the model state in different ways.

### **2.2.3 Physical Structure**

Regarding the project's physical structure, at this point (development and staging) is not required any specific solution such as server redundancy. At the time of writing the project was hosted in a virtual server running CentoOS, with 4GB RAM and 20GB ROM. As virtual server, at any time is possible to increase ROM or RAM, making it a viable solution.

Regarding the code repository, besides a local copy, a Bitbucket free account will be used as a lightweight code reviewing system.



# 3 Development

## 3.1 Tools and Technologies

Considering the analysis performed and the requirements presented, a selection of tools and technologies was made for developing the platform and to implement security rules and best practices during software development.

### 3.1.1 Technologies

Since this project takes place in the scope of a startup company, one objective was to use open source and freeware technologies. Being the project a web platform, the best set of technologies to use was HTML5 with JavaScript and CSS3, supported in web platforms of the AMP Stack type.

The AMP Stack is a popular set of technologies used to run dynamic web sites and servers in the main OS, such as Linux (LAMP), Mac OS (MAMP) and Windows (WAMP). It includes Apache, MySQL, and PHP/Python/Perl. It is considered one of the best platforms for development and deployment of high performance web applications requiring a rock-hard and consistent foundation. Some of the advantages of AMP, compared with other types of web platforms, are the performance, uptime and availability. On the other side, a disadvantage of this platform is the weak performance of the MySQL server when dealing with large volumes of data.

In this project AMP stands for Apache, MySQL, and PHP.

PHP is an open source project with the majority of code written to be portable between operating systems. Besides this, when compared to competitors such as Perl, .NET or Ruby, PHP has a large set of strengths, including built-in libraries, scalability, performance and portability.

### **3.1.2 Tools**

During technical development were also used several tools to support the assessment and implementation of the main technologies. Since the ultimate goal was to develop the best platform in the shortest period of time, the company selected all the tools based on its accumulated experience.

#### **3.1.2.1 Tools for Development Support**

The set of tools described are used primarily for development of the solution. Although they continue to be used after the platform's launch, it is most likely that they will not be changed.

##### **ApiGen**

ApiGen is a tool to create good API documentation from PHP source code. [Jaroslav Hanslík, Ondřej Nešpor & Nette Foundation, 2012]

ApiGen has support for PHP 5.3 namespaces/packages and uses Taxy to create useful and easy to read HTML documentation. Furthermore, ApiGen also uses FSHL to create highlighted source code, a Nette Framework system for templating, and TokenReflection to explain the documented source code. [Jaroslav Hanslík, Ondřej Nešpor & Nette Foundation, 2012]

##### **Bootstrap**

Bootstrap is an open source front-end framework for developing web projects, of all dimensions, which includes a responsive mobile first fluid grid system, used for creating page layouts through a series of rows and columns housing the site content and scaling up to 12 columns as the device or viewport size increases.

Bootstrap is built on Less, a preprocessor with additional functionality like variables and mixins, that compiles its CSS and uses media queries to create the key breakpoints in the grid system, as well as for showing and hiding content by device. [Spurlock, 2013]

The main reasons supporting the choice of Bootstrap as front-end framework were grounded in all the previously mentioned features, and because there exists a large set of plugins complementing some JavaScript needs, such as colorpicker, datepicker and bootstrap-tour.

### **Git**

Git is a free and open source distributed version control system, with a tiny footprint and lightning fast performance. [Gajda, 2013]

With features like cheap local branching, convenient staging areas, and multiple workflows, Git allows developers to have multiple branches of the same project, which can be entirely independent of each other. All the branches' functionalities such as create, merge, and delete are simple to execute in a command prompt. This way, developers can create new branches for each novel functionality and seamlessly switch back and forth between them, or delete each branch when that feature gets merged into the main line. [Gajda, 2013]

### **jQuery**

jQuery is a Cross-Browser fast and feature-rich JavaScript library that supports CSS3 selectors to find elements, as well as in style property and HTML manipulation, event handling and animations. [Franklin, 2013]

The main reasons to use jQuery are related with the way it makes Ajax much simpler and easy-to-use, with a combination of versatility and extensibility making a framework with a largest set of plugins. In this project some of the plugins to be used are jQueryFlippy, jQueryGeocomplete and jQueryQRCode.

### **NetBeans IDE**

In terms of an integrated development environment (IDE) for this project, the choice was the NetBeans IDE. NetBeans is an open-source project committed to provide free tools for developers in order to produce software more quickly and efficiently, by leveraging the strengths of platforms and code standards of the major programming languages. [NetBeans, 2014]

The reasons for the choice of NetBeans IDE were based on its web development features, such as a dedicated PHP coding environment, full support for HTML5, JavaScript, and CSS3, and a complete integration with web standards.

Another reason for using NetBeans IDE is due to the possibility of debugging PHP code using Xdebug, which is a NetBeans plugin to debug scripts and web pages, either locally or remotely. Among other, Xdebug allows the inspection of local variables, to set breakpoints, to navigate to declarations, types and files using hypertext links, going to shortcuts, and to evaluate live code.

NetBeans also provides command-line debugging, displaying the program output in a command line window within the IDE. This way it is possible to inspect the generated HTML without having to switch to a browser.

### **Phabricator**

Phabricator is an open source set of web applications aiming to help software developers in building better software. Phabricator is mostly written in PHP, with the server running on Linux or Mac OS X distributions, but it can be used on any platform due to his powerful command line client Arcanist, which works on Linux, Mac OS X or Windows. [Phacility Inc, 2014]

Phabricator includes a set of tools to execute the following tasks:

- Review and audit code;
- Host Git/Hg/SVN repositories;
- Browse repositories;
- Track bugs or "features";
- Track tasks;
- Create Wiki for the projects;
- Hide stuff from coworkers.

Besides all the Phabricator features, two of the major requirements that led to use this tool were the fact of being open source and developed in PHP. For now Phabricator is used as wiki repository, tasks tracker and bugs tracker, however a short-term goal is to better explore its potential.

### **PHPUnit**

PHPUnit is a de-facto standard testing framework for PHP, created by Sebastian Bergmann with the philosophy that “the sooner you detect code errors, the quicker you can solve them”. [Sebastian Bergmann, 2014]

PHPUnit is used in unit testing and applies assertions to verify the expected behavior of the code being tested. Its output can be created in several formats such as JUnit XML or JSON.

## **Yii**

As PHP was the server side language adopted, the chosen framework should bring some leverage and significantly reduce the development process. Taking into account requirements such as speed, security and documentation, the adopted framework was Yii. It is a high-performance PHP framework for developing web 2.0 applications coming with lots of features such as MVC, ActiveRecord, authentication and role-based access control. Yii is a framework for overall purpose web programming and can be used in the development of almost all kinds of web-applications. Due to having an advanced caching feature, Yii is especially good for the development of applications with big traffic flows, such as portals, forums, CMS or E-commerce products. Yii is particularly well-known for being flexible, practice and fast. [Yii Framework, 2014a]

The following list shows 15 power features supported by the Yii framework: [DollarFry, 2013]

1. It is fast;
2. Supports the MVC pattern;
3. Includes a great CRUD feature (Gii);
4. Allows an ORM approach (database tables as objects);
5. Includes Easy Form Validation;
6. Supports JQuery and Ajax;
7. Has inbuilt Authentication and Authorization;
8. Supports theming;
9. Provides Web Services support;
10. Supports caching;
11. Contains Error Handling;
12. Addresses security (XSS, CSRF and Cookie Attack Prevention);
13. Allows extensions;
14. Supports testing;
15. Has good documentation.

Besides the features already described, Yii was also selected because it supports all major data base systems. As such, the system's data scalability is guaranteed, because it is always possible to change the DBMS to a different one, if needed.

### **3.1.2.2 Tools for Solution Support**

Although this toolkit is also used during the development process, it is considered to be part of the solution because it will be used in the solution release.

#### **Apache HTTP Server**

The Apache HTTP Server Project, usually known as Apache, is a collaborative software development effort to create a good implementation of an open source HTTP Web server, and is managed by The Apache Software Foundation.

Its first version (Apache 1.0) was released on December 1, 1995, and currently is the world's most popular Web server, mostly because is implemented in almost all operating systems such as Windows, Solaris, Linux, Mac OS, and many more. [The Apache Software Foundation, 2014]

The choice of this tool was based on its scalability, security and additional features supported by means of add-ons.

Scalability - Apache allows load balancing in multiple instances of an application, allowing it to handle more volume, as well as maintain stability in case of instance goes down.

Security – Apache support Secure Sockets Layer (SSL), a cryptographic protocol, designed to provide secure communication over the Internet. Furthermore, with the installation of add-on “modsecurity web application firewall”, it provides extra protection against attacks like Denial of Service (DoS), Cross-site scripting (XSS), SQL injection and others.

Additional Features – Apache has a bunch of modules that can be installed to simplify/improve features such as authentication and URL rewriting, or for interfacing with other programming languages.

#### **Criticue**

Criticue.com is an online tool that helps developers improve their web designs by exchanging feedback with a large community of web designers, usability experts and entrepreneurs. [Criticue, 2013]

One of the best Criticue's features comes from all reviews being moderated, meaning that spam reviews are avoided, and allowing users to only receive valid feedback. Its usage is free of charge. It works as one-for-one exchange: for one user's web page review he gets one review of his own work, and the more feedback is posted, the more feedback is received. [Criticue, 2013]

### **OPcache**

OPcache is a tool of the core PHP distribution that improves PHP performance by storing precompiled script bytecode in shared memory. [The PHP Group, 2014a]

In each request to PHP scripts, the code is parsed and compiled into some opcode, and then executed in the Zend Engine. This allows the PHP machinery to skip the compilation step required in languages like Java or C#, and all changes made to the PHP code can be immediately seen. With OPcache it is possible to skip the script's loading and parsing on each request, reducing execution time and improving the overall site performance. [The PHP Group, 2014a]

### **phpMyAdmin**

PhpMyAdmin is a PHP web application offering a complete web interface for administering MySQL databases, widely recognized as the leading application in this field, and supported by numerous developers and translators worldwide.

The phpMyAdmin core features cover the basic MySQL database and table operations, but it also has an internal system maintaining metadata to support advanced features. Additionally, system administrators can manage users and their access privileges. [Delisle, 2012]

The goal of phpMyAdmin is to provide a comprehensive web-based management of MySQL servers and data, keeping up with MySQL and the worldwide web standards' evolution. Being an open source project, anyone can install its own copy of phpMyAdmin on a web server and personalize it according with its needs.

### **Peek**

Peek is a basic service that gives developers and designers a way to see real people using their website or mobile app. Unlike Criticue, in Peek normal users perform the tests, although both of them are free to use.

With Peek you can “get a peek into the mind of your users super-fast and totally free”. Basically, after submitting a request, the user receives a 5 minute video from a real person using the site or app and responding to three simple questions [Copy Hackers, 2014]:

1. What is your first impression of this site? What is the site for?
2. What would you do next on the page? Describe your experience.
3. What did you like? Not like? Would you return to the site in the future?

With this type of feedback developers can improve their user experience and good sense.

Peek is provide by User Testing, an online tool used by more than 20000 customers, including the top 10 web companies like Google, Facebook and Twitter [User Testing, 2014].

### **XHProf**

XHProf is a light-weight hierarchical and instrumentation based profiler, with a simple HTML based user interface written in PHP. Its raw data collection component is implemented in C as a PHP extension, and the reporting/UI layer browser-based makes it easy to view and analyze results. [The PHP Group, 2014b]

Through the data collection phase, XHProf keeps track of the number of executed calls, counts inclusive metrics for the dynamic call graph, and in the reporting phase computes metrics such as memory usage and CPU time. [The PHP Group, 2014b] Additionally, it supports hierarchical DIFF reports to compare two runs and aggregate results from multiple runs. [The PHP Group, 2014b]

XHProf handles recursive functions by detecting cycles in the call graph, a data collection time itself, and avoiding the cycles by giving unique depth qualified names for the recursive invocations. Its reports can be used to compute some metrics, such as what chain of calls led to a particular function getting called, and can be helpful in understand the structure of the executed code . [The PHP Group, 2014b]

## **3.2 Environment**

Project deployments should be treated as part of a development workflow usually including at least three environments: development, staging and production. That is the case in this project.

Developers work on features in the development environment with a specific branch, and once implemented, they are merged into the staging branch in order to perform all tests. After all tests completed and new features validated, they are merged into the production environment and become ready to use. Below is described in more detail each of the environments.

### **3.2.1 Development**

The development environment is based on a localhost with the MAMP Stack. A Mac OS X operating system is used, running the Apache server with PHP 5.4 and a MySQL instance. The Chrome browser is adopted by default, however Firefox, Opera and Safari are also used. In each browser the newest version is selected, quickly updated whenever possible.

Since the application to develop aims to create passes to be used by the iOS operating system, and those passes must comply with a set of rules and restrictions imposed by Apple, this is the most appropriate development environment, because it allows debugging iOS and Android applications, which in different environments could be complicated because Xcode (the iOS IDE) is only available for Mac.

### **3.2.2 Staging**

Regarding the staging environment, it was used the local machine mentioned before and an environment exactly like the Production environment. This testing environment was located at the same server, but the production environment had a different URL.

Alongside was created an Ubuntu 13.10 virtual machine and a Windows machine with Windows 7 for testing different operating systems with different browsers.

In the Ubuntu virtual machine browsers like Chrome, Firefox and Opera were used, and in the Windows machine the Internet Explorer browser was included.

For the Chrome browser some extensions were installed in order to improve the development and testing processes. In Table 6 are presented the extensions and a description about how they were used.

Table 6 - Extensions for Chrome used on the staging environment

Name	Application
NetBeans Connector	Test the responsiveness of the design
PageSpeed	Test the page loading speed
Postman	Test the REST API simulating API requests on localhost
Xdebug helper	Debug the code on the development environment
YSlow	Test the page loading speed

### 3.2.3 Production

The main specification about the production environment dictates that it will be hosted on a cloud server. The option of using a cloud web server is grounded on several reasons. One of the most important is the system's scalability, being easy to extend the resources once they are needed. Another is the resource configuration, allowing the creation of a test environment to satisfy the effective platform's needs. Relevant is also the set of features such as hosting multiple sites, running multimedia applications and creating DNS, FTP, and mail servers.

As mentioned before, the production environment will be based on a LAMP stack platform with CentOS. Depending on the growth of the platform's usage, is not ruled out the possibility of implementing advanced features like caching systems.

This is an anticipated scenario. As such, by release time the server configuration may be different.

## 3.3 Implementation

As mentioned before, the project aims at a web-based platform to manage the pass life cycle, divided in three main parts:

- A web platform;
- A push notification API;
- A web service/API;

Along with these three major components, there are other important aspects in this project implementation. These are fault tolerance, transparency, heterogeneity, consistency and the effective communication protocols. Below is described how they were implemented.

### 3.3.1 Documentation

Regarding the documentation, two different approaches were used: for internal and for external use.

The “internal” process is to generate documentation for internal use. Basically, this documentation is automatically generated using comments embedded in the source code. This is very important because, if the project were to be developed by another programmer, good documentation would allow him to better understand the project’s code.

Since the internal documentation must be automatically generated, it is necessary to have some tool to do it. Two tools were tested: ApiGen and phpDocumentor. Essentially, these are tools that execute similar processes to collect the partial documentation and both understand code annotations, although they produce documentation in different ways. After testing it was concluded both tools were good to the job, but being ApiGen the Netbeans recommended tool, it turned out to be the selected one.

Regarding external documentation, basically it goes through documenting the API and the currently available methods. Besides the documentation, a test account was created to serve as a Postman collection. In Appendix I it is presented the API documentation.

### 3.3.2 Web platform

#### 3.3.2.1 Overview

Taking into account the project requirements, the web platform was developed with state of the art technologies for web development, such as HTML5, CSS3 and font icons, responsive design, and the template designer allowed a visual preview of the solution. The web platform is also ready to support all features from the Passkit framework, including geo-fencing, geo-location and beacons.

#### 3.3.2.2 Features

The platform features were implemented according to the specified requirements. Two major groups were implemented: templates and passes.

In Table 7 are presented the pass's implemented features, and in Table 8 depict the ones related to templates. For security reasons all passes have a digital signature that is verified by Apple.

Table 7 - Pass's implemented features

Feature	Description
Archive Pass	Archive a pass for not to be used
Copy Pass	Create a pass from an existing one
Create Pass	Create a pass from a template
Delete Pass	Delete a pass from the data base
Restore Pass	Restore a pass so it can be used again
Share Pass	Share a pass in social networks
Update Pass	Update the pass dynamic fields
View Pass	View a pass details and actions

Table 8 - Template's implemented features

Feature	Description
Copy Template	Create a template from an existing one
Create Template	Create a template from scratch
Delete Template	Delete a template from the data base
Update Template	Update an existing template
View Template	View a template details and actions

Other implemented features are related to user and account management. For such purposes methods to register users were implemented, as well as update profile, reset password, etc., in order to access the platform. The platform also allows non-registered users, as such

frontend pages were developed like “about”, “terms of service”, “privacy policy”, etc., where unregistered users can learn about the platform.

Taking into account one of the platform's requirements, design implementation must be responsive as possible in all its dimensions, however, when using a simulator, the design is responsive only to a certain level. As such, when used in smartphones, user experience is not yet the best.

### 3.3.2.3 Logical Structure

Regarding the logical structure, the framework guarantees it. Yii implements the MVC design pattern, however Yii goes a step further, and besides implementing MVC, also introduces the Application concept. Application is a front-controller that encapsulates the execution context and processing of a request. Application collects the information about the user's requests and dispatches it to the appropriate controller. [Yii Framework, 2014b]

The following image shows a typical workflow of a Yii application and how the requests are handled:

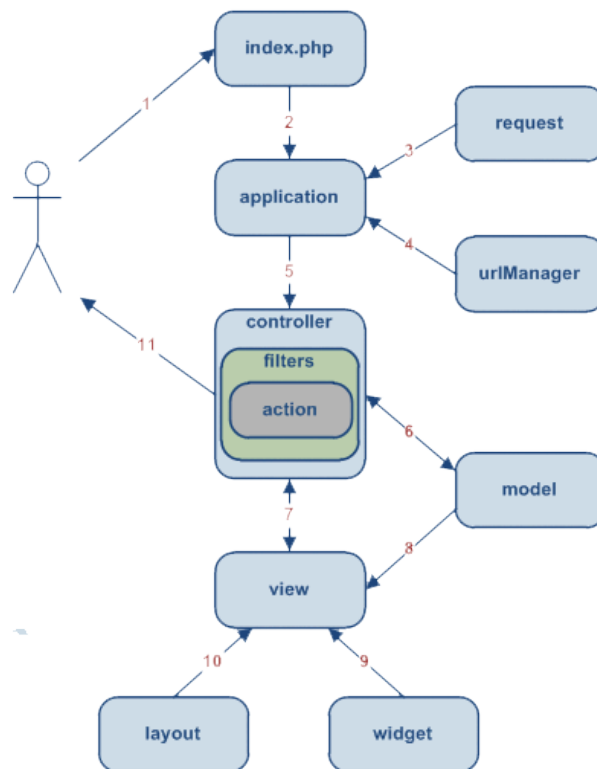


Figure 8 - Typical Yii application workflow [Yii Framework, 2014b]

1. A user makes a request through `http://pass.eddmi.com/index.php/user/view/12` and the server handles the request by executing the bootstrap script `index.php`.
2. The bootstrap script creates an Application instance and runs it.
3. The Application calls an application component named `request` and gets detailed about the request.
4. With the help of an application component (`urlManager`), the application determines the requested controller and action. For this example, the controller is `user`, which refers to the `UserController` class, and the action is `view`, whose actual meaning is determined by the controller.
5. The application creates an instance of the selected controller to handle the request. The controller determines that the action `view` refers to a method named `actionView` in the controller class, so creates and executes filters associated with this action, such as access control or benchmarking. If the filters permit, the action is executed.
6. The action reads a User model whose database ID is 12.
7. The action renders a view named `view` with the User model.
8. The view reads and displays the attributes of the User model.
9. The view executes some widgets.
10. The view rendering result is embedded in a layout.
11. The action completes the view rendering and displays the result to the user.

#### **3.3.2.4 Transparency**

Regarding the system's transparency, it was concluded that it is transparent because platform users cannot tell whether it relies on external services. In other words, for the user all transactions are only processed within the platform.

#### **3.3.2.5 Heterogeneity**

Regarding to the system's heterogeneity, it was concluded that it has been achieved. Since the user interface is a web page, this enables the service to be used on any operating system with any architecture, only being necessary a modern web browser.

#### **3.3.2.6 Consistency**

Regarding the data consistency, all operations are performed within the platform without any relevant third party service, so consistency is guaranteed. Regarding the notifications consistency, the situation is slightly different.

If notification are made by email, once again all operations are executed without third party service and consistency is guaranteed, however, if notification is made by push notification to an user smartphone, it is necessary to use the Apple Push Notification Service or the PassWallet Notification Service, and consistency is no longer guaranteed. In case of a push notification failure, the next one to execute with success will upgrade the pass with data from all transactions.

#### **3.3.2.7 Fault Tolerance**

Concerning the system's fault tolerance, it has been partially implemented. The system is partially fault tolerant because if some third party service fails, the website, which contains the user interface, will still function. The Geo-location and ShareThis services are good examples of this situation. If any of these services fails, part of the processing will stop working, but the platform will not fail completely, thus implementing a secure fail good practice.

#### **3.3.2.8 Communication Protocols**

Communication protocol: HTTPS.

Synchronous messaging: REST.

Asynchronous messaging: REST.

### **3.3.3 Push notification API**

Regarding to the push notification API, it is prepared to support iOS using the Apple Push Notification Service, and Android using and a third part service, which also is a mobile wallet known as Passwallet. The Apple Push Notification Service (APNs) is a service that uses push technology, through a constantly opened IP connection, to forward notifications from servers of third party applications to Apple devices. The Passwallet service, developed by Attido Mobile, is similar to the APNs and uses the same technologies, with a difference when sending notifications to Android devices. [Attido Mobile, 2013]

The push notification API is used to update passes to their latest version. Updating passes is a task shared between the API and the wallet that is using the pass. When information on a pass changes, the API server sends a push notification to the user's device, which makes the device communicate with the server to find out what was changed on the pass and get its latest version. With exception of the pass type ID and serial number, anything in the pass can be updated. [Apple Inc. 2013a]

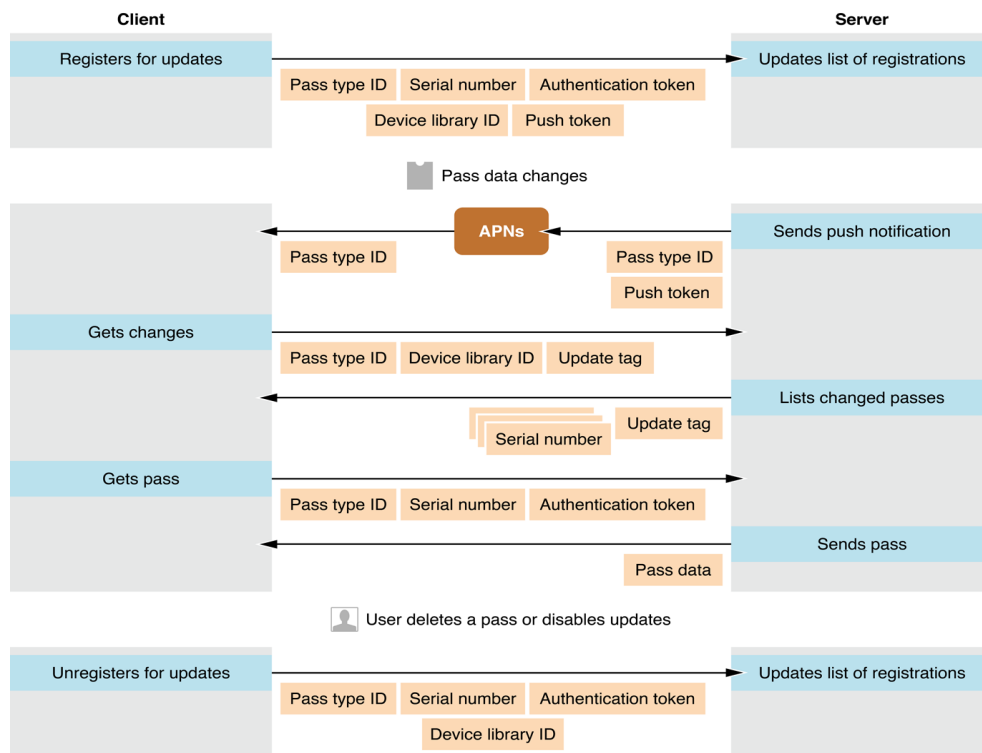


Figure 9 - Interactions between the client and the Push Notification API [Apple Inc. 2013a]

In Figure 9 are shown all the communication between the clients, usually smartphones, and the server. This process goes on in three different situations: when the client first imports a pass, when the pass has some change, and when the client removes the pass from his digital wallet.

The first communication occurs when the user imports a new pass for his digital wallet, and basically registers the client on the server in order to receive push notifications for that given pass. Besides the pass information, it also sends the device library ID and push token. With these two items, associated with the pass serial number, the server can send push notifications to the device. That is what happens on the second communication. When the pass is changed, the server notifies all clients, which then request a pass update. Finally, the third communication process occurs when the client deletes the pass. This process is similar to the first one, but at this time the client is unregistering himself from the pass registration's list.

Since third party services are used for push notifications, the implementation is basically a class with a set of methods or some functions that invoke the APNs or the Attido Mobile API, depending on the device OS.

### 3.3.4 Web service API

#### 3.3.4.1 Overview

The web service is an interface to the pass platform that third party services can use to access some functionalities. It uses the Representational State Transfer (REST) architecture over HTTP protocol for communications, and JSON objects for data transmission. To perform actions using the web service, it is required to comply with the calling convention by sending a request to the endpoint, a method and some arguments. The response will be received and formatted as a JSON file. It is also required an HTTP header specifying the Authorization Token to grant access. This token is automatically generated at the account creation, and is located at the user profile.

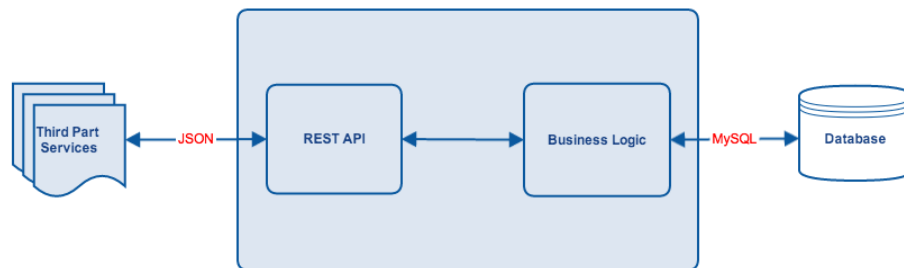


Figure 10 - Web Service's API Structure Overview

The choice for a RESTful web service was based on the simplicity of interfaces, high performance, scalability, portability, reliability, and because it can be accessed from any environment allowing HTTP requests. [Fielding and Taylor, 2002] On the other hand, the choice of JSON for data transmitting is due to its lightweight data-interchange format, easy to read and write by humans, as well as easy to parse and generate by machines.

#### 3.3.4.2 Features

Unlike the web platform, features associated with the web service go further than the initial requirements. For the service to be used at its best, it was necessary to create some additional methods to the platform's leverage.

These additional methods are used both for templates and passes, as well as for a new group: validation. It mainly serves to validate the API access token. This method is relevant because it allows the API user to verify the token validity before executing operations that require more resources. In Table 9 are presented the web service implemented methods.

Table 9 - Web Service API implemented methods

Method	Description
Validate Token	Validate an API access token
Get Templates By Token	Retrieve all templates related with a given token
Get Template Dynamic Fields	Retrieve template dynamic fields
Create Pass	Create a pass with a given template id
Update Pass	Update a pass with a given serial number
Get Pas	Retrieve a pass with a given serial number
Void Pass	Invalidate a pass with a given serial number
Unvoid Pass	Validate a pass with a given serial number

Appendix I contains the printable version of the API documentation, including more detailed implementation and examples.

#### 3.3.4.3 Transparency

Regarding transparency, the web service is transparent because user's cannot know whether it relies on external services or if operations are processed by the service itself.

#### 3.3.4.4 Heterogeneity

Since web service functions use well-defined response codes for all requests, the heterogeneity of the system is guaranteed for all devices. The web service API relying on the HTTP protocol also enforces system's heterogeneity. In the Appendix I is described the web service response codes.

#### 3.3.4.5 Consistency

Since all operations are performed by the platform without third party service, data consistency is guaranteed. Regarding notification consistency, similar to the web platform, the web service cannot guarantee its consistency.

If the notification is done by email, consistency is guaranteed, however, if notification is a push for the user's smartphone, consistency is no longer guaranteed. Once again, if any push notification fails, the next one successfully executed will upgrade the pass to its latest version.

#### 3.3.4.6 Fault Tolerance

Concerning fault tolerance, the service is fault tolerant because it does not use third party services, and all errors are returned with a response code and an error message describing what went wrong.

#### 3.3.4.7 Communication Protocols

Communication protocol: HTTPS.

Synchronous messaging: REST.

### 3.3.5 Security

Regarding security issues two aspects have to be considered: passes' security and platform's security.

#### Passes

The way Apple structured passes assured the necessary security. When a pass is created, it includes a PKCS#7 signature created using a certificate provided by Apple. Whenever the pass is imported to Passbook, or updated, this signature is validated. If any of passes' files was changed without being validated by its signature, the pass import will fail.

#### Platform

Authentication and use of cryptography are two of the most important contributions to a web based application security. On the other hand, bad use of cryptography is one of top 10 software security flaws. [Arce, Clark-Fisher, Daswani, DelGrosso, Dhillon, Kern, Kohno, Landwehr, McGraw, Schoenfield, Seltzer, Spinellis, Tarandach, and West, 2014]

Normally a successful user authentication results in the creation of a session for the user. If such session comes from easy-to-obtain information, such as username and password, forging identities may be a problem. To prevent it, authentication was implemented using Yii's built-in authentication/authorization framework. [Yii Framework, 2014c] Security was one of Yii's concerns since its initial development, its base security includes Cross-site Scripting Prevention (XSS), Cross-site Request Forgery Prevention (CSRF) and Cookie Attack Prevention. [Yii Framework, 2014 d]

The core piece of Yii authentication framework is a pre-declared user application component that implements the IWebUser interface. The user component represents the persistent identity information for the current user, and can be globally accessed using `Yii::app()->user`. With the user component, it is possible to execute a set of authentication operations, such as login or logout of a user, check if a user is logged in or not, or check if the user can perform specific operations. [Yii Framework, 2014c]

A created session will represent the user across the site, however, unlike other implementations, Yii's never stores the user's password. The validation is provided by a SHA256 hash generated from the user inserted password and a salt stored in the database.

Once the validation is made, the working session is created with an expiration time and only stores non-critical information.

Regarding the authorization procedures, they were also created based on Yii's implementation. Yii has a powerful role-based access control (RBAC) provided by his authentication/authorization framework, in which is possible to set an authorization hierarchy and user roles. Besides some access rules, this project will use three user roles: guest, authenticated and admin. [Yii Framework, 2014c]

**Guest** - User with access to all pages of the website not requiring authentication, such as home page and contacts. In Yii, guests are represented by the symbol “\*”.

**Authenticated** - An authenticated user, besides sharing the same rights as a guest user, has access to the reserved area for registered users, including all the templates and passes' features. In Yii, authenticated users are represented by the symbol “@”.

**Admin** - An admin user has access to all the website's features, including those authorized to guests and authenticated users. In Yii, admins are represented by “admin”.

These user roles can be associated with tasks, operations or access rules. An access rule is a piece of code that is always executed to validate the user's access.

```
public function accessRules()
{
    return array(
        array('allow',
            'actions' => array('distribution', 'download', 'email'),
            'users' => array('*'),
        ),
        array('allow',
            'actions' => array('index', 'view', 'create', 'update', 'delete'),
            'users' => array('@'),
        ),
        array('allow',
            'actions' => array('admin'),
            'users' => array('admin'),
        ),
        array('deny',
            'users' => array('*'),
        ),
    );
}
```

Figure 11 - Access rules example

In Figure 11 is shown an example of access on a controller. The controller's actions distribution, download and email are accessible to all type of users, but index, view, create, update and

delete are only available for authenticated users'. The action admin can only be accessed by admin users. If a guest user tries to access, for example, the action create, such access will succeed validation and the user will be redirected to the site's home page or other page, depending on configuration.

The use of Apache Server has contributed to the platform security. Apache provides secure communication by using TLS/SSL, and installation of the "mod\_security" add-on, working as web application firewall, also provided extra protection against DoS, XSS and SQL injection attacks.

### **Tools and Third part services**

Besides Apache and Yii, other tools were used in the development and support. Among features mentioned before, security was considered a most relevant concern. The solution is based on current tools, in constant development, within an active community, contributing to reduce the appearance of security breaches. Besides Apache and Yii, Bootstrap, JQuery, and Git are examples of those current tools.

Regarding the use of third party services, was used the same criteria. All services count in a big team and are in constant improvement. Examples are the geocoding service supported by Google and the Push Notification Service supported by Apple.



# 4 Testing and Deployment

## 4.1 Development

### 4.1.1 Unit

The use of unit tests is to monitor each piece of program's code and assure it works correctly. This kind of test is used to detect problems early in the development cycle by describing inputs and outputs each piece of code must comply with. In this project unit tests all data models were tested and results correctly asserted. Tests were essentially made to validate operations in some important features.

```
<?php
class LoginTest extends CDbTestCase
{
    public function testLogin()
    {
        $login = new Login;
        $login->username = "lribeiro@eddm1.com";
        $login->passwd_repeat = $login->passwd = "123456";

        $this->assertTrue($login->save());

        // verify the login is pending activation
        $login = Login::model()->findByPk($login->id);
        $this->assertTrue($login instanceof Login);
        $this->assertEquals($login->active, 0);

        // active the login and and verify again
        $login->active = 1;
        $this->assertEquals($login->active, 1);
    }
}
```

Figure 12 - Login unit test

## Features tested

Table 10 - Features tested

Feature	Complexity	Notes
Sign Up	Low	Required few inputs
Login	Low	Required few inputs
Create Pass	Medium	Requires few inputs and a template

## Features not tested

Table 11 - Features not yet tested

Feature	Complexity	Notes
Create Template	High	Required lot inputs and images
Update Template	High	Required lot inputs and images
Copy Template	High	Required lot inputs and images
Delete Template	Low	Does not requires inputs
View Template	Low	Does not requires inputs
Update Pass	Low	Required few inputs
Delete Pass	Low	Does not requires inputs
Archive Pass	Low	Does not requires inputs
Restore Pass	Low	Does not requires inputs
View Pass	Low	Does not requires inputs
Share Pass	Low	Does not requires inputs
Create Issue	Low	Required few inputs
Contact	Low	Required few inputs

## Notes

The reason to test only three features is due to the fact that untested features have the same degree of complexity as the tested ones. The exception is create template, which was not tested because it required special inputs, such as images, and also the use of a geo-location service, introducing great complexity in the creation of tests.

## Approach

The approach used was Black Box Unit Testing.

## Failure Criterion

Failure occurs when the obtained output is different of expected output, for each test case.

## Sign Up

Table 12 - Sign Up test cases

#	Inputs					Output	
	F. Name	L. Name	Email	Passwd	Repeat	Expected	Obtained
1	Luis	Ribeiro	<a href="mailto:lribeiro@eedmi.com">lribeiro@eedmi.com</a>	passwd	passwd	Success	Success
2	Manuel	Ribeiro	<a href="mailto:mrifeiro@eddm.com">mrifeiro@eddm.com</a>	passwd	nopass	Error	Error
3	Manuel	Martins	<a href="mailto:mrifeiro@eddm.com">mrifeiro@eddm.com</a>	passwd	passwd	Error	Error

#1: Passed because all conditions were fulfilled.

#2: Passed because the passwords mismatched, causing the expected fault.

#3: Passed because the email used was already registered, causing the expected fault.

## Login

Table 13 - Login test cases

#	Inputs		Output	
	Username/Email	Passwd	Expected	Obtained
1	<a href="mailto:lribeiro@eedmi.com">lribeiro@eedmi.com</a>	passwd	Success	Success
2	<a href="mailto:lribeiro@eddm.com">lribeiro@eddm.com</a>	nopass	Error	Error
3	<a href="mailto:mrifeiro@eddm.com">mrifeiro@eddm.com</a>	passwd	Error	Error

#1: Passed because all conditions were fulfilled.

#2: Passed because the inserted password was incorrect, causing the expected fault.

#3: Passed because the email used was not registered, causing the expected fault.

## Create Pass

Table 14 - Create Pass test cases

#	Inputs				Output	
	Template	Name	Email	Contact	Expected	Obtained
1	280	Luís Ribeiro	<a href="mailto:lribeiro@eedmi.com">lribeiro@eedmi.com</a>	919076340	Success	Success
2	2800	Luís Martins	<a href="mailto:lmartins@eddm.com">lmartins@eddm.com</a>	917881580	Error	Error

#1 - Passed because all conditions were fulfilled.

#2 - Passed because the template with the indicated id did not exist, causing the expected fault.

### 4.1.2 Integration

In accordance with performed unit tests, it was possible to execute an integration test. The test integrated sign up and login tests.

#### Approach

To perform a test the same input was used for the unit tests in its multiple combinations. The tests are described in two stages: the first referring to the sign up action, and the second to the login.

#### Failure Criterion

Failure occurs when the combination of both outputs is not equal to the combination of the expected output for each test case.

## 1<sup>st</sup> Stage - Sign Up

Table 15 - 1<sup>st</sup> Stage data for the integration test cases

#	Inputs					Output	
	F. Name	L. Name	Email	Passwd	Repeat	Expected	Obtained
1	Luis	Ribeiro	<a href="mailto:lribeiro@eedmi.com">lribeiro@eedmi.com</a>	passwd	passwd	Success	Success
2	Luis	Ribeiro	<a href="mailto:lribeiro@eedmi.com">lribeiro@eedmi.com</a>	passwd	passwd	Success	Success
3	Luis	Ribeiro	<a href="mailto:lribeiro@eedmi.com">lribeiro@eedmi.com</a>	passwd	nopass	Error	Error

## 2<sup>nd</sup> Stage - Login

Table 16 - 2<sup>nd</sup> Stage data for the integration test cases

#	Inputs		Output	
	Username/Email	Passwd	Expected	Obtained
1	<a href="mailto:lribeiro@eedmi.com">lribeiro@eedmi.com</a>	passwd	Success	Success
2	<a href="mailto:lribeiro@eedmi.com">lribeiro@eedmi.com</a>	nopass	Error	Error
3	<a href="mailto:lribeiro@eedmi.com">lribeiro@eedmi.com</a>	passwd	Error	Error

#1: Passed because all conditions were fulfilled.

#2: Passed because the combination of expected outputs for each stage was fulfilled. The first stage was successfully completed. In the second stage the inserted password was incorrect, causing the expected error.

#3: Passed because the combination of expected outputs for each stage was fulfilled. Since the first stage execution returned an error, caused by the password mismatch, the second stage action also failed because the email was not registered in the platform.

## 4.2 The solution

To better understand how the performed tests in the Appendix V are presented some screenshots for the presented solution.

### 4.2.1 Functional

The main goal of functional testing is to ensure application's complete functionality. Black box tests were selected for this purpose. Since this type of tests requires some inputs, to analyze the outputs only features that required data processing were tested. To analyze and validate outputs it was used XHProf, a PHP profiling tool.

#### Contact

##### Overview

Objective:	Get in touch with the company by the contact form.
Preconditions:	None
Success Conditions:	Enter all the required fields, including a valid email.
Fault conditions:	Required fields, or a valid email missing.

Table 17 - Contact Function Test Inputs

Input	Value
Name	Luís Ribeiro
Email	<a href="mailto:lribeiro@edDMI.com">lribeiro@edDMI.com</a>
Subject	Functional Test
Body	Hi, I'm executing a functional test to ensure this feature works. Great.
Code	xrrhye

**Result:** success

RunID: 541220dddf6a

URL: test-pass.edDMI.com/index.php/site/contacts

Stat	Exact URL	Similar URLs
Count	<u>10</u>	<u>10</u>
Min Wall Time	37.6760 ms	37.6760 ms
Max Wall Time	3.8974 s	3.8974 s
Avg Wall Time	1.1278 s	1.1278 s
95% Wall Time	3.8974 s	3.8974 s
<b>Display run Incl. Wall Time (microsec)</b>	<b>2,043,359 microsecs</b>	
Min CPU Ticks	36.9950 ms	36.9950 ms
Max CPU Ticks	206.9690 ms	206.9690 ms
Avg CPU Ticks	123.3816 ms	123.3816 ms
95% CPU Ticks	206.9690 ms	206.9690 ms
<b>Display run Incl. CPU (microsecs)</b>	<b>204,969 microsecs</b>	
Min Peak Memory Usage	2,945,920 microsecs	2,945,920 microsecs
Max Peak Memory Usage	4,589,192 microsecs	4,589,192 microsecs
Avg Peak Memory Usage	3,331,859 microsecs	3,331,859 microsecs
95% Peak Memory Usage	4,589,192 microsecs	4,589,192 microsecs
<b>Display run Incl. MemUse (bytes)</b>	<b>4,490,344 bytes</b>	
<b>Number of Function Calls:</b>	<b>7,852</b>	

Figure 13 - XHProf output for the request contact

## Sign Up

### Overview

Objective:	Sign Up for access the reserved area.
Preconditions:	None
Success Conditions:	Enter a valid email address that is not yet registered.
Fault conditions:	Email is not valid or is already registered.

Table 18 - Sing Up Functional Test Inputs

Input	Value
First Name	Luís
Last Name	Ribeiro
Email	<a href="mailto:lribeiro@eddm.com">lribeiro@eddm.com</a>
Password	Iribeiro
Repeat Pwd	Iribeiro
Account Type	1 (Free)

**Result:** success

RunID: 541222262c0e5

URL: test-pass.eddmi.com/index.php/site/signup

Stat	Exact URL	Similar URLs
Count	<u>5</u>	<u>5</u>
Min Wall Time	112.1480 ms	112.1480 ms
Max Wall Time	3.1533 s	3.1533 s
Avg Wall Time	1.3671 s	1.3671 s
95% Wall Time	3.1533 s	3.1533 s
<b>Display run Incl. Wall Time (microsec)</b>	<b>3,136,572 microsecs</b>	
Min CPU Ticks	44.9930 ms	44.9930 ms
Max CPU Ticks	164.9750 ms	164.9750 ms
Avg CPU Ticks	93.7858 ms	93.7858 ms
95% CPU Ticks	164.9750 ms	164.9750 ms
<b>Display run Incl. CPU (microsecs)</b>	<b>147,978 microsecs</b>	
Min Peak Memory Usage	3,813,720 microsecs	3,813,720 microsecs
Max Peak Memory Usage	5,963,120 microsecs	5,963,120 microsecs
Avg Peak Memory Usage	4,542,904 microsecs	4,542,904 microsecs
95% Peak Memory Usage	5,963,120 microsecs	5,963,120 microsecs
<b>Display run Incl. MemUse (bytes)</b>	<b>5,793,704 bytes</b>	
<b>Number of Function Calls:</b>	<b>14,112</b>	

Figure 14 - XHProf output for the request sign up

## Sign In

### Overview

Objective:	Sign In into the reserved area.
Preconditions:	Must be a registered user.
Success Conditions:	Valid username and password.
Fault conditions:	Wrong username or password.

Table 19 - Sign In Functional Test Inputs

Input	Value
Username	<a href="mailto:lribeiro@eddm.com">lribeiro@eddm.com</a>
Password	lribeiro

**Result:** success

RunID: 541233df1b201

URL: test-pass.eddmi.com/index.php/user/login

Stat	Exact URL	Similar URLs
Count	<u>7</u>	<u>7</u>
Min Wall Time	46.0650 ms	46.0650 ms
Max Wall Time	328.6780 ms	328.6780 ms
Avg Wall Time	131.3216 ms	131.3216 ms
95% Wall Time	328.6780 ms	328.6780 ms
<b>Display run Incl. Wall Time (microsec)</b>	<b>97,506 microsecs</b>	
Min CPU Ticks	33.9950 ms	33.9950 ms
Max CPU Ticks	54.9910 ms	54.9910 ms
Avg CPU Ticks	45.1361 ms	45.1361 ms
95% CPU Ticks	54.9910 ms	54.9910 ms
<b>Display run Incl. CPU (microsecs)</b>	<b>54,991 microsecs</b>	
Min Peak Memory Usage	3,281,272 microsecs	3,281,272 microsecs
Max Peak Memory Usage	3,748,040 microsecs	3,748,040 microsecs
Avg Peak Memory Usage	3,430,203 microsecs	3,430,203 microsecs
95% Peak Memory Usage	3,748,040 microsecs	3,748,040 microsecs
<b>Display run Incl. MemUse (bytes)</b>	<b>3,720,680 bytes</b>	
<b>Number of Function Calls:</b>	<b>5,402</b>	

Figure 15 - XHProf output for the request sign in

## Create Template

### Overview

Objective:	Create a Template.
Preconditions:	Must have a valid session.
Success Conditions:	Inputs have valid data.
Fault conditions:	Inputs have invalid data.

Table 20 - Create Template Functional Test Inputs

Input	Value
Layout	Generic (Visit Card)
Certificate	pass.com.codigoirreverente.visitcard
Template Name	CV Luís Ribeiro
Organization Name	Código Irreverente, Lda.
Description	Cartão de Visita de Luís Ribeiro
Serial Number	Auto Generated Serial
Time zone	z
Background Color	rgb(245,245,245)
Label Text Color	rgb(82,207,207)
Value Text Color	rgb(0,0,0)
eventTypeldentifier	Strip
stripTypeldentifier	Strip

Input	Value
Header Fields	, , headerFields[Key_0]
Primary Fields	Web Developer, , Nome
Secondary Fields	Email, , Email
Auxiliary Fields	Contacto, , Contacto
Back Fields	, backFields[Key_0],
Barcode	PKBarcodeFormatPDF417, URL to Download, , No message,
Lat	
Lng	
Dynamic	On
Images	logo, thumbnail

**Result:** success

RunID: 540e3c0c3df78

URL: test-pass.eddmi.com/index.php/passtemplate/create

Stat	Exact URL	Similar URLs
Count	<a href="#">19</a>	<a href="#">19</a>
Min Wall Time	54.6030 ms	54.6030 ms
Max Wall Time	3.2933 s	3.2933 s
Avg Wall Time	562.5781 ms	562.5781 ms
95% Wall Time	3.2933 s	3.2933 s
<b>Display run Incl. Wall Time (microsec)</b>	<b>3,221,666 microsecs</b>	
Min CPU Ticks	48.9930 ms	48.9930 ms
Max CPU Ticks	3.9674 s	3.9674 s
Avg CPU Ticks	581.6486 ms	581.6486 ms
95% CPU Ticks	3.9674 s	3.9674 s
<b>Display run Incl. CPU (microsecs)</b>	<b>3,679,441 microsecs</b>	
Min Peak Memory Usage	4,447,768 microsecs	4,447,768 microsecs
Max Peak Memory Usage	6,004,032 microsecs	6,004,032 microsecs
Avg Peak Memory Usage	5,153,349 microsecs	5,153,349 microsecs
95% Peak Memory Usage	6,004,032 microsecs	6,004,032 microsecs
<b>Display run Incl. MemUse (bytes)</b>	<b>5,221,768 bytes</b>	
<b>Number of Function Calls:</b>	<b>6,588</b>	

Figure 16 - XHProf output for the request create template

## Update Template

### Overview

- Objective: Update a Template.
- Preconditions: The template has to previously exist.
- Success Conditions: Inputs have valid data.
- Fault conditions: Inputs have invalid data.

Table 21 - Update Template Functional Test Inputs

Input	Value
Template ID	280
Layout	Generic (Visit Card)
Certificate	pass.com.codigoirreverente.visitcard
Template Name	CV Luís Ribeiro
Organization Name	Código Irreverente, Lda.
Description	Cartão de Visita de Luís Ribeiro
Serial Number	Auto Generated Serial
Time zone	z
Background Color	rgb(255,255,255)
Label Text Color	rgb(82,207,207)
Value Text Color	rgb(0,0,0)
eventTypeId	Strip
stripTypeId	Strip
Header Fields	, , headerFields[Key_0]
Primary Fields	Web Developer, , Nome
Secondary Fields	Email, , Email
Auxiliary Fields	Contacto, , Contacto
Back Fields	, backFields[Key_0],
Barcode	PKBarcodeFormatPDF417, URL to Download, , No message,
Lat	
Lng	
Dynamic	On
Images	logo, thumbnail

**Result:** success

RunID: 540e3cf999ecc

URL: test-pass.eddmi.com/passtemplate/update/279

Stat	Exact URL	Similar URLs
Count	<a href="#">2</a>	<a href="#">2</a>
Min Wall Time	99.1290 ms	99.1290 ms
Max Wall Time	3.3409 s	3.3409 s
Avg Wall Time	1.7200 s	1.7200 s
95% Wall Time	3.3409 s	3.3409 s
<b>Display run Incl. Wall Time (microsec)</b>	<b>3,340,940 microsecs</b>	
Min CPU Ticks	70.9890 ms	70.9890 ms
Max CPU Ticks	3.9424 s	3.9424 s
Avg CPU Ticks	2.0067 s	2.0067 s
95% CPU Ticks	3.9424 s	3.9424 s
<b>Display run Incl. CPU (microsecs)</b>	<b>3,942,401 microsecs</b>	
Min Peak Memory Usage	5,398,560 microsecs	5,398,560 microsecs
Max Peak Memory Usage	5,533,712 microsecs	5,533,712 microsecs
Avg Peak Memory Usage	5,466,136 microsecs	5,466,136 microsecs
95% Peak Memory Usage	5,533,712 microsecs	5,533,712 microsecs
<b>Display run Incl. MemUse (bytes)</b>	<b>5,280,112 bytes</b>	
<b>Number of Function Calls:</b>	<b>11,592</b>	

Figure 17 - XHProf output for the request update template

## Create Pass

### Overview

Objective:	Create a Pass.
Preconditions:	Must have a valid session, and at least a template.
Success Conditions:	Inputs have valid data.
Fault conditions:	Inputs have invalid data.

Table 22 - Create Pass Functional Test Inputs

Input	Value
Template Id	280
Name	Luís Ribeiro CV
Primary Fields	Luís Ribeiro
Secondary Fields	Iribeiro@codigoirreverente.com
Auxiliary Fields	+351 919076340

**Result:** success

RunID: 540f7f87b572f

URL: test-pass.eddmi.com/index.php/pass/create/280

Stat	Exact URL	Similar URLs
Count	<u>6</u>	<u>6</u>
Min Wall Time	59.8270 ms	59.8270 ms
Max Wall Time	256.1720 ms	256.1720 ms
Avg Wall Time	141.4042 ms	141.4042 ms
95% Wall Time	256.1720 ms	256.1720 ms
<b>Display run Incl. Wall Time (microsec)</b>	<b>256,172 microsecs</b>	
Min CPU Ticks	48.9930 ms	48.9930 ms
Max CPU Ticks	102.9840 ms	102.9840 ms
Avg CPU Ticks	67.3232 ms	67.3232 ms
95% CPU Ticks	102.9840 ms	102.9840 ms
<b>Display run Incl. CPU (microsecs)</b>	<b>102,984 microsecs</b>	
Min Peak Memory Usage	5,180,000 microsecs	5,180,000 microsecs
Max Peak Memory Usage	5,270,720 microsecs	5,270,720 microsecs
Avg Peak Memory Usage	5,205,059 microsecs	5,205,059 microsecs
95% Peak Memory Usage	5,270,720 microsecs	5,270,720 microsecs
<b>Display run Incl. MemUse (bytes)</b>	<b>5,241,896 bytes</b>	
<b>Number of Function Calls:</b>	<b>6,972</b>	

Figure 18 - XHProf output for the request create pass

## Update Pass

### Overview

Objective:	Update a Pass.
Preconditions:	Must have a valid session, and the pass to update.
Success Conditions:	Inputs have valid data.
Fault conditions:	Inputs have invalid data.

Table 23 - Update Pass Functional Test Inputs

Input	Value
Template Id	280
Name	Luís Ribeiro CV
Primary Fields	Luís Ribeiro
Secondary Fields	lribeiro@edDMI.com
Auxiliary Fields	+351 919076340

**Result:** success

RunID: 54137f230d83a

URL: test-pass.edDMI.com/index.php/pass/update/1263

Stat	Exact URL	Similar URLs
Count	<u>8</u>	<u>15</u>
Min Wall Time	41.3430 ms	41.3430 ms
Max Wall Time	5.4830 s	10.9400 s
Avg Wall Time	1.0020 s	1.3307 s
95% Wall Time	5.4830 s	10.9400 s
<b>Display run Incl. Wall Time (microsec)</b>	<b>43,652 microsecs</b>	
Min CPU Ticks	36.9940 ms	36.9940 ms
Max CPU Ticks	75.9880 ms	75.9880 ms
Avg CPU Ticks	50.3675 ms	51.7923 ms
95% CPU Ticks	75.9880 ms	75.9880 ms
<b>Display run Incl. CPU (microsecs)</b>	<b>36,994 microsecs</b>	
Min Peak Memory Usage	4,310,184 microsecs	4,293,712 microsecs
Max Peak Memory Usage	5,391,632 microsecs	5,391,632 microsecs
Avg Peak Memory Usage	4,675,499 microsecs	4,576,243 microsecs
95% Peak Memory Usage	5,391,632 microsecs	5,391,632 microsecs
<b>Display run Incl. MemUse (bytes)</b>	<b>4,211,240 bytes</b>	
<b>Number of Function Calls:</b>	<b>6,132</b>	

Figure 19 - XHProf output for the request update pass

## Create Issue

### Overview

Objective:	Create an Issue.
Preconditions:	Must have a valid session.
Success Conditions:	Inputs have valid data.
Fault conditions:	Inputs have invalid data.

Table 24 - Create Issue Functional Test Inputs

Input	Value
Title	Functional Test
Platform	Mac OS
Browser	Chrome
Comment	I'm performing a functional test to assure that this feature works well.

**Result:** success

RunID: 541234291e279

URL: test-pass.eddmi.com/index.php/user/issue

Stat	Exact URL	Similar URLs
Count	<u>7</u>	<u>7</u>
Min Wall Time	54.1260 ms	54.1260 ms
Max Wall Time	11.2025 s	11.2025 s
Avg Wall Time	3.5528 s	3.5528 s
95% Wall Time	11.2025 s	11.2025 s
<b>Display run Incl. Wall Time (microsec)</b>	<b>9,342,940 microsecs</b>	
Min CPU Ticks	42.9930 ms	42.9930 ms
Max CPU Ticks	177.9720 ms	177.9720 ms
Avg CPU Ticks	94.4140 ms	94.4140 ms
95% CPU Ticks	177.9720 ms	177.9720 ms
<b>Display run Incl. CPU (microsecs)</b>	<b>177,972 microsecs</b>	
Min Peak Memory Usage	3,798,192 microsecs	3,798,192 microsecs
Max Peak Memory Usage	5,746,976 microsecs	5,746,976 microsecs
Avg Peak Memory Usage	4,550,634 microsecs	4,550,634 microsecs
95% Peak Memory Usage	5,746,976 microsecs	5,746,976 microsecs
<b>Display run Incl. MemUse (bytes)</b>	<b>5,653,072 bytes</b>	
<b>Number of Function Calls:</b>	<b>15,228</b>	

Figure 20 - XHProf Output for the request create issue

## Notes

The times contained in the XHProf outputs have some variation in the min and max values, because outputs include all of the request for that URL. Since the URL is prepared to accept GET and POST requests, values have some variation. For performance analyzes it is considered the average time. To better understand this times, Appendix IV presents two callgraphs related to the aforementioned requests.

For safety reasons forms are submitted using the POST method and the passed data is omitted, as presented in the Figure 21.

Cookie	Results
cc_cookie_decline	null
cc_cookie_accept	cc_cookie_accept
_ga	GA1.2.2079155855.1406733268
PHPSESSID	dlnrg9j4heq8mo2tfj54nlj0k1
__unam	abc4cec-147975d98f4-5ce1f63b-50

Get	Results

Post	Results
Skipped	Post data omitted by rule

Figure 21 - XHProf Cookie, Get and Post outputs

### 4.2.2 Acceptance

Since at this stage the platform is only accepting limited beta users, feedback from users is very important. In order to collect their opinion, a user's form has been created where they are redirected to every time they logout the application reserved area.

The presented questions were mostly related to platform's usability, however, the survey also served to question users about new features they would like to see implemented, and also questioning about the future of digital wallets. On Appendix II are presented the survey questions and also the obtained results. Besides those questions, the survey also requests the user email, in order to validate feedback from a registered user.

In response to the feedback obtained, improvements have been made in the platform. In terms of functionality, a wizard for templates was the one most suggested by beta testers (most of them with basic knowledge about computer science) as being the most useful. As such, it was also implemented. Below is presented the use case for this "future" question and the table with the Template's implemented features updated.

#### Use Case Wizard

- Objective: Create a Template with Wizard.
- Preconditions: The user must be logged;  
Must exist at least one template on Wizard.
- Success Conditions: The user inserts valid data.
- Fault conditions: The user inserts invalid data.
- Actors: User, Platform, Geocoding Service, Push Service.

Table 25 - Template's implemented features updated

Feature	Description
Copy Template	Create a template from an existing one
Create Template	Create a template from scratch
Delete Template	Delete a template from the data base
Update Template	Update an existing template
View Template	View a template details and actions
Wizard	Create a template from a wizard

The feedback system is mostly based on the interface design. It was concluded that would be important for registered users to report technical issues. To fill this gap it was created a

system to report bugs or suggest improvements. Currently it is a simple form where the user can report a problem and indicate operating system and browser used, being the follow-up process processed by email, but intending to evolve into an issue tracker.

### **4.2.3 Usability**

Regarding platform usability, its evaluation was divided in two parts: public part and reserved for registered users. Since these two are very different, two approaches were applied.

In order to test the public site's usability two online tools were used: Criticue.com and Peek. Due to aforementioned characteristics, Criticue.com turned out to be the most useful tool and also the one with more and better feedback.

Although most of feedback did not imply any change to design and usability, some changes were made, so this tool effectively helped to improve site's usability. In Appendix III is presented all the feedback gathered from Criticue.com.

Regarding Peek tool, it was used to get feedback from real users. Although this feedback is more difficult to assess, in the end it was a good complement to Criticue.com because opinions from those who have actually used the platform could be collected.

Concerning the part only accessible to registered users, a different approach was applied and acceptance tests survey were used to collect feedback. Essentially some questions used to validate user acceptance could also be used to test usability. Here are those questions:

- 2. What do you think about the platform design?*
- 3. In your opinion what is the degree of difficulty in creating a template?*
- 4. Do you think a wizard could simplify the template design?*

Although a direct answer was not collected about what was wrong, it allowed to determine what was wrong in some platform components, and motivated developers to reflect about it and make improvements. On Appendix II are presented the survey's questions and obtained results.

#### 4.2.4 Security

Regarding security tests, the method consisted in performing vulnerabilities' scans during development. A tool called Nessus was used, basically a Network Vulnerability Scanner allowing to audit and track the web server, looking for vulnerabilities such as SQL Injection and XSS.

The last scan test was performed on September 20 and allowed the identification of 58 different vulnerabilities, with some of them detected more than once. Although 58 may be a large number of vulnerabilities, none of them was critical or even of high level, not generating a significant risk to the platform.

On Figure 22 is presented the resume of the latest performed scan, as explained below.

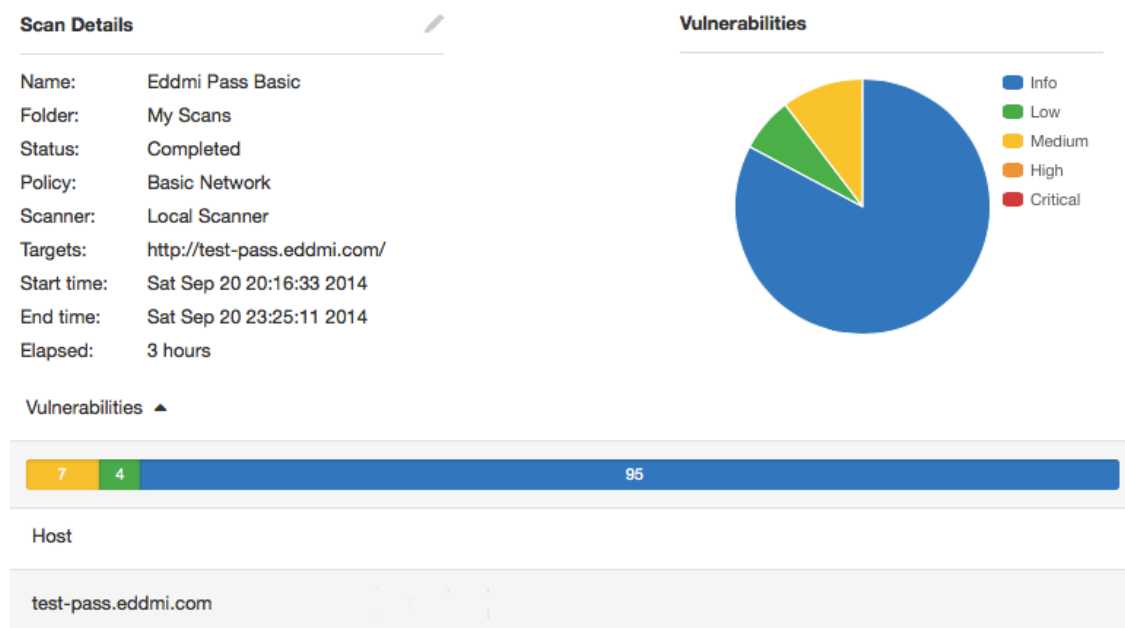


Figure 22 - Nessus Vulnerability Scan Resume

Nessus classifies the vulnerabilities into five levels: Info, Low, Medium, High and Critical. While the vulnerabilities of Info, Low and Medium levels do not represent relevant risk to the platform, high or critical level risks are of real concern and should be fixed as soon as possible.

About the latest test, no vulnerability of critical or high level was found, so it can be concluded the platform does not contain any serious risk.

Regarding the vulnerability of medium level, it refers to the SSL certificate. Since at pre-production stage it was not necessary the use an authentic certificate, the self-signed certificate used was identified as vulnerability.



# **5 Conclusion**

## **5.1 Thesis Summary**

This thesis describes a concept of mobile marketing focused on customer's value and loyalty, providing a cutting edge technology tool to create loyalty systems based on digital cards and enhance direct marketing. Besides the concept, this work also explains the requirements and resources to implement and support the platform, including security concerns and best practices.

This thesis also includes implementation suggestions, including a group of tests to validate the solution in terms of technological and usability levels, detailing the tools and technologies selected to support the implementation.

## **5.2 Completed Objectives**

Regarding the company's goal of turning the Eddmi Pass Platform into a main platform for small and medium business in the area of digital marketing, it has been changed. That goal was transferred to the Eddmi Customer Link Platform, which offers a larger set of features and is already attracting reference customers, however, the project was very important because it delivered the basis for cards to be used in the digital communication with customers. In fact, this project became so relevant that, even with the creation of a new platform, still it will be launched as an independent platform. The site layout will be maintained but the name will be changed to Pass-Link, and the overall image and color pallet also will undergo some changes.

About the project itself, in short all objectives were achieved. The main goal was to implement a secure and almost 100% functional prototype, and now it is possible to conclude this goal was fully achieved, or even surpassed by the development of features not included in initial requirements.

Regarding other goals, they included usability tests with various types of target users, use of good practices in secure software development, as well as an information security oriented testing process.

Concerning usability tests, due to changes in the company's strategy for the project, the number of beta users has been reduced, however, with the use of tools like Critique and Peek, it was feasible to get feedback from different sources and validate design and usability. Regarding security-related objectives, because development of the solution was based in security practices, with tools chosen by their level of support and security, it can be stated that they have been achieved.

## **5.3 Other Works**

Although this project works as a standalone application, it also was used as part of a bigger project called "Eddmi Customer Link".

The Eddmi Customer Link is a loyalty platform using Passbook as one of the ways to communicate with customers. Although the platform supports all the "old loyalty systems" like physical cards, its goal is to improve customer mobile marketing and end user experience by means of digital cards.

The first development approach was to keep two separated platforms. The loyalty platform would support customer's data, campaigns, reports and all required functionalities for loyalty programs, and would use the pass platform to create and update the digital cards.

The communication between both platforms was based on the public API of the pass platform, whose documentation appears in Appendix I, which is based on message exchange using JSON over the HTTP protocol. The communication authentication was assured by sending an authentication token (MD5 hash) in the header's request. Most of API methods were developed to be used on the loyalty platform.

Despite this approach shown a good performance, failing some of the established usability and simplicity metrics for the loyalty platform was not acceptable. Even so, its development was very important because it proved the relevance of the pass platform concept.

Since the first approach did not succeed, a second and final approach was tried. Basically it consisted in merging core functionalities of the pass platform into the loyalty platform. This implied some changes to the database schema and also added some new features. As the tools used in loyalty platform development were the same used in the pass platform, the development process was kept simple.

Although the main goals were to simplify the configurations and remove the need for users to register in two platforms, this approach also improved platform's performance, proving that this approach was a good business/technical decision.

Alongside with loyalty platform development, two mobile applications were also developed. The first one was created to help prove the concept and promote the system as a complete solution, and the second was developed for a customer to be used in beta stage. Both applications are hybrid solutions developed using Ionic and AngularJS Frameworks, and compiled with Apache Cordova. In Figure 23 are presented screenshots of each application.

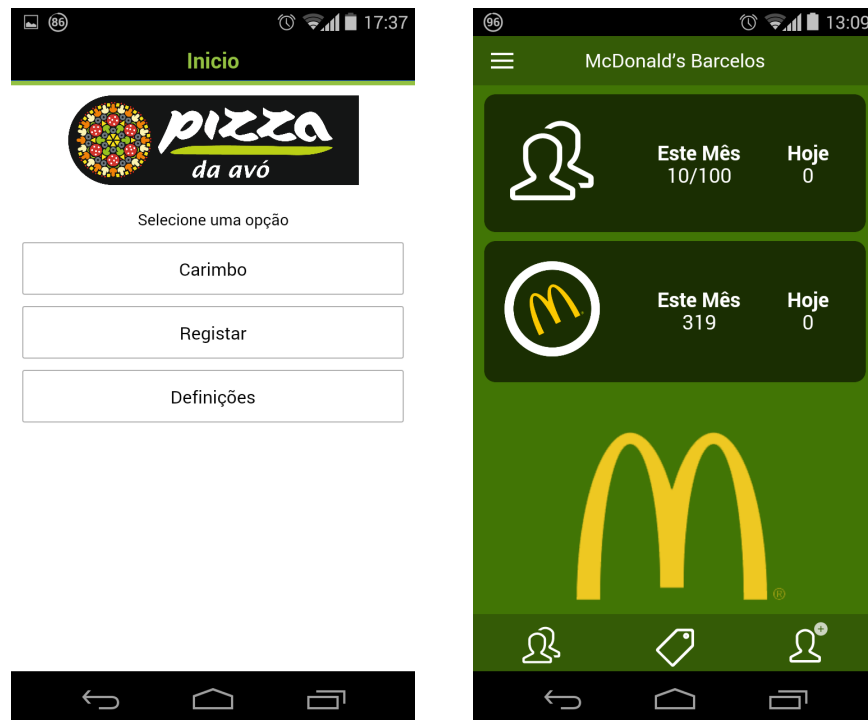


Figure 23 – Screenshots of the mobile applications

## 5.4 Limitations

At this stage of the project (pre-production solution), the physical resources are adjusted, but after application release and with many users accessing it, this may become a limitation. This is a concern, and in the future the system will undergo some changes on its structure (e. g., server redundancy and distributed servers).

Other limitation may be due to the use of a standard MySQL database. If the number of users grows exponentially, it will be necessary to migrate the database to an adequate system. Although this is an improbable scenario, it had impact on the choice of tools to use. This is why Yii was selected as the PHP framework, because it supports different database systems with minimum effort.

## 5.5 Future Work

The work to be undertaken in the future concerns the following topics:

- Improve the design for smartphones;
- Improve the issues feature;
- Implement API methods for uploading images on the templates design;
- Implement sign up with social networks;
- Move the code repository to our Phabricator's repository;
- Improve the solution's physical resources for when going into production.
- Use HHVM over PHP on the server (Tests prove performance improvement).

## 6 References

- [Apple Inc, 2012] Apple Developer, Getting Started with Passbook on iOS 6. Available at: [https://developer.apple.com/Passbook/Getting\\_Started\\_with\\_Passbook.pdf](https://developer.apple.com/Passbook/Getting_Started_with_Passbook.pdf). [Last Access: 2014-02-14].
- [Apple Inc, 2013a] Apple Developer, Passbook Programming Guide. Available at: [https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/PassKit\\_PG/PassKit\\_PG.pdf](https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/PassKit_PG/PassKit_PG.pdf). [Last Access: 2014-02-20].
- [Apple Inc, 2013b] Apple Developer, Passbook Package Format Reference. Available at: [https://developer.apple.com/library/ios/documentation/UserExperience/Reference/PassKit\\_Bundle/PassKit\\_Bundle.pdf](https://developer.apple.com/library/ios/documentation/UserExperience/Reference/PassKit_Bundle/PassKit_Bundle.pdf). [Last Access: 2014-02-20].
- Apple Inc, 2014a] Apple Developer, Getting Started with Apple Pay. Available at: <http://www.apple.com/apple-pay/> [Last Access: 2014-09-19].
- Apple Inc, 2014b] Apple Pay. Available at: <https://developer.apple.com/apple-pay/Getting-Started-with-Apple-Pay.pdf> [Last Access: 2014-09-20].
- [Arce, Clark-Fisher, Daswani, DelGrosso, Dhillon, Kern, Kohno, Landwehr, McGraw, Schoenfield, Seltzer, Spinellis, Tarandach, and West, 2014] Arce I, Clark-Fisher K, Daswani N, DelGrosso J, Dhillon D, Kern C, Kohno T, Landwehr C, McGraw G, Schoenfield B, Seltzer M, Spinellis D, Tarandach I, and West J. Avoiding the top 10 software security design flaws. 2014. IEEE Computer Society. Available at: <http://cybersecurity.ieee.org/images/files/images/pdf/CybersecurityInitiative-online.pdf>. [Last Access: 2014-09-20].
- [Attido Mobile, 2013] PassWallet, PassWallet Developer Guide v0.6. Available at: <http://passwallet.attidomobile.com/PassWallet%20Developer%20Guide.pdf>. [Last Access: 2014-02-20].
- [Boronczyk and Naramore] Timothy Boronczyk, Elizabeth Naramore, Jason Gerner, Yann

- and Gerner and Scouarnec and Stolz and Glass, 2009] Le Scouarnec, Jeremy Stolz, Michael K. Glass. Beginning PHP6, Apache, MySQL Web Development. 2009 Wiley Publishing, Inc. ISBN: 978-0-470-39114-3.
- [Copy Hackers, 2014] Get A “Peek” Into Your Site’s Usability: A Review. Available at: <http://copyhackers.com/2014/03/get-peek-sites-usability-review>. [Last Access: 2014-09-02].
- [Criticue, 2013] Criticue.com. Available at: <http://www.criticue.com> [Last Access: 2014-09-02].
- [Delisle, 2012] Marc Delisle. Mastering phpMyAdmin 3.4 for Effective MySQL Management. 2012 Packt Publishing. ISBN 978-1-84951-778-2.
- [DollarFry, 2013] 14 features that make Yii, the best php framework. Available at: <http://www.dollarfry.com/8-reasons-to-use-yii-frameworkthe-best-php-framework>. [Last Access: 2014-03-12].
- [Estimote, 2014] Estimote Beacons. Available at: <http://estimote.com>. [Last Access: 2014-03-12].
- [Fielding and Taylor, 2002] Fielding, Roy T.; Taylor, Richard N., Principled Design of the Modern Web Architecture, 2002. Available at: <http://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf>. [Last Access: 2014-02-20].
- [Franklin, 2013] Jack Franklin. Beginning jQuery. 2013 Apress. ISBN13: 978-1-4302-4932-0.
- [Gajda, 2013] Wodzimierz Gajda. Git Recipes A Problem-Solution Approach. 2013 Apress. ISBN13: 978-1-4302-6103-2.
- [Jaroslav Hanslík, Ondřej Nešpor & Nette Foundation, 2012] ApiGen. Available at: <http://apigen.org/> [Last Access: 2014-09-15].
- [Laurie and Laurie, 2009] Ben Laurie, Peter Laurie. Apache: The Definitive Guide, 3rd Edition. O'Reilly Media 2009. Ebook ISBN: 978-0-596-55599-3.
- [NetBeans, 2014] NetBeans IDE Features - PHP Development. Available at: <https://netbeans.org/features/php>. [Last Access: 2014-03-20].
- [Our Mobile Planet, 2014] Our Mobile Planet research. Available at: <http://think.withgoogle.com/mobileplanet/en/>. [Last Access: 2014-10-20].
- [Phacility Inc, 2014] phabricator . Available at: <http://phabricator.org/> [Last Access: 2014-09-15].
- [Sebastian Bergmann, 2014] PHPUnit. Available at: <https://phpunit.de/> [Last Access: 2014-09-15].
- [Scrum Reference Card, 2014] About Scrum. Available at: <http://scrumreferencecard.com/scrum-reference-card/> [Last Access: 2014-09-23].

- [Spurlock, 2013] Jake Spurlock. Bootstrap. 2013 O'Reilly Media. ISBN: 978-1-449-34391-0.
- [The Apache Software Foundation, 2014] Apache HTTP Server Project. Available at: <http://httpd.apache.org>. [Last Access: 2014-03-11].
- [The Internet Society, 1999] Network Working Group, RFC2616 - Hypertext Transfer Protocol -- HTTP/1.1. Available at: <http://tools.ietf.org/html/rfc2616>. [Last Access: 2014-02-20].
- [The Internet Society, 2006] Network Working Group, RFC4627 - The application/json Media Type for JavaScript Object Notation (JSON). Available at: <http://tools.ietf.org/html/rfc4627>. [Last Access: 2014-02-20].
- [The PHP Group, 2014a] OPcache. Available at: <http://php.net/manual/en/intro.opcache.php> [Last Accessed: 2014-09-15].
- [The PHP Group, 2014b] XHProf. Available at: <http://pt1.php.net/manual/en/intro.xhprof.php> [Last Access: 2014-09-09].
- [Tiobe Software, 2014] TIOBE Index for September 2014. Available at: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>. [Last Access: 2014-09-23].
- [User Testing, 2014] User Testing. Available at: <http://www.usertesting.com> [Last Access: 2014-09-02].
- [Yii Framework, 2014a] About Yii. Available at: <http://www.yiiframework.com/about>. [Last Access: 2014-03-12].
- [Yii Framework, 2014b] Yii Fundamental - Model-View-Controller (MVC). Available at: <http://www.yiiframework.com/doc/guide/1.1/en/basics.mvc>. [Last Access: 2014-05-21].
- [Yii Framework, 2014c] Authentication and Authorization. Available at: <http://www.yiiframework.com/doc/guide/1.1/en/topics.auth>. [Last Access: 2014-09-21].
- [Yii Framework, 2014d] Yii Security. Available at: <http://www.yiiframework.com/doc/guide/1.1/en/topics.security>. [Last Access: 2014-09-21].
- [Wang, 2011] Guanhua Wang. Application of lightweight MVC-like structure in PHP. Publish in Business Management and Electronic Information (BMEI), 2011 International Conference on Guangzhou (Volume: 2) - Page(s): 74 - 77. 2011 IEEE. Print ISBN: 978-1-61284-108-3.



# APPENDIX I





Copyright © 2014 Código Irreverente, Lda.

Last revision: 03/12/2014

## API Documentation

---

Version: v1 (draft)

Created: 04/06/2014

Latest Review: 04/06/2014

Root URL: <https://pass.eddmi.com>

---

## Overview

---

The Eddmi Pass API allows you to call methods over the HTTPS protocol and sends the responds in REST JSON style.

To perform an action using the Pass API, you need to select a calling convention, send a request to its endpoint specifying API Token (for some methods), a method and some arguments. The response will be received formatted as a JSON file.

### Authentication

Most requests will need no authentication, because the used ids and serial number required only authorized user have access, but for some methods you will need an API token that is send as an URL parameter. You can get the API token on the user profile at the web platform.

### Encoding

Use UTF-8 encoding when sending arguments to the API methods.

# Authentication

---

## Methods

- Validate Token

## Validate Token

### Description

Check if a given token is valid to access the API.

### Request

GET /api/v1/token/\$api\_token HTTP/1.1

Host: pass.eddmi.com

### URL Parameters

\$api\_token: The token to access the API.

## Example

---

### Request

GET /api/v1/token/247a00c98fb293d7e91bc24f1ef3096d HTTP/1.1

Host: pass.eddmi.com

### Response with Success

```
{  
  "status": "success",  
  "message": "token is valid"  
}
```

## Response with Error

```
{  
  "status": "error",  
  "message": "token is not valid"  
}
```

## Templates

---

### Methods

- Get Templates By Token
- Get Template Dynamic Fields

### Get Templates By Token

#### Description

Retrieve all templates of a given token

#### Request

GET /api/v1/templates/\$api\_token HTTP/1.1

Host: pass.eddmi.com

#### URL Parameters

\$api\_token: The token to access the API.

## Example

---

### Request

GET /api/v1/templates/247a00c98fb293d7e91bc24f1ef308b5 HTTP/1.1  
Host: pass.eddmi.com

### Response with Success

```
{
  "status": "success",
  "message": "templates retrieved",
  "data": [
    {
      "id": "8553394ce8009ae",
      "name": "Points 4",
      "description": "Points 4",
      "passtype_id": "pass.com.codigoirreverente.pointscard",
      "image": "https://pass.eddmi.com/uploads/templates/85/8553394ce8009ae/pass.png"
    },
    ... ..
  ]
}
```

### Response with Error

```
{
  "status": "error",
  "message": "user not found"
}
```

## Get Template Dynamic Fields

### Description

Retrieve all dynamic fields of a given template.

### Request

GET /api/v1/fields/\$template\_id HTTP/1.1

Host: pass.eddmi.com

### URL Parameters

\$template\_id: The template id.

## Example

---

### Request

GET /api/v1/fields/8553394ce8009aq HTTP/1.1

Host: pass.eddmi.com

### Response with Success

```
{
  "status": "success",
  "message": "dynamic fields retrieved",
  "data": [
    "Points",
    "Name"
  ]
}
```

## Response with Error

```
{  
  "status": "error",  
  "message": "template not found"  
}
```

## Passes

---

### Methods

- Create Pass
- Get Pass
- Update Pass
- Void Pass
- Unvoid Pass

## Create Pass

### Description

Create a new pass with a given template id.

### Request

POST /api/v1/pass/\$template\_id HTTP/1.1

Host: pass.eddmi.com

### URL Parameters

\$template\_id: The template id.

## Example

---

### Request

POST /api/v1/pass/8553394ce8009ae HTTP/1.1

Host: pass.eddmi.com

### Payload

```
{  
  "Passname": "API Test",  
  "serialNumber": " ", // do no include it if template haves auto generated serial  
  "pontos": 10,  
  "name": "API Guy"  
}
```

### Response with Success

```
{  
  "status": "success",  
  "message": "pass created",  
  "data": {  
    "serial": "e0c565a36f29748ef0edd04ed47a58fa",  
    "distribution": "https://pass.eddmi.com/index.php/pass/distribution/8553b689e24ed5a",  
    "download": "https://pass.eddmi.com/index.php/pass/download/8553b689e24ed5a"  
  }  
}
```

## Response with Error

```
{
  "status": "error",
  "message": "template not found"
}
```

## Get Pass

### Description

Retrieve the links for distribution and download of the pass with the given serial number.

### Request

GET /api/v1/pass/\$passtype /\$serial TTP/1.1

Host: pass.eddmi.com

### URL Parameters

\$passtype: The pass type id.

\$serial: The pass serial number.

## Example

---

### Request

GET  
/api/v1/pass/pass.com.codigoirreverente.pointscard/42cf0d23dfe4070e95332438dd7ed0  
HTTP/1.1

Host: pass.eddmi.com

## Response with Success

```
{
  "status": "success",
  "message": "pass retrieved ",
  "data": {
    "distribution": "https://pass.eddmi.com/index.php/pass/distribution/8553b582e5950c2",
    "download": "https://pass.eddmi.com/index.php/pass/download/8553b582e5950c2"
  }
}
```

## Response with Error

```
{
  "status": "error",
  "message": "pass not found"
}
```

## Update Pass

### Description

Update a pass with the given serial number.

Request

PUT /api/v1/pass/\$passtype/\$serial HTTP/1.1

Host: pass.eddmi.com

### URL Parameters

\$passtype: The pass type id.

\$serial: The pass serial number.

## Example

---

### Request

PUT

/api/v1/pass/pass.com.codigoirreverente.pointscard/42cf0d55dfe4070e95332438d69d7e

HTTP/1.1

Host: pass.eddmi.com

### Payload

```
{  
  "Passname": "API Test",  
  "serialNumber": " ", //do no include it if the template haves auto generated serial  
  "pontos": 100,  
  "name": "API Guy"  
}
```

### Response with Success

```
{  
  "status": "success",  
  "message": "pass updated",  
  "data": {  
    "passtype": "pass.com.codigoirreverente.pointscard",  
    "serial": "42cf0d55dfe4070e95332438d69d7ed0",  
    "distribution": "https://pass.eddmi.com/index.php/pass/distribution/8553b582e5950c2",  
    "download": "https://pass.eddmi.com/index.php/pass/download/8553b582e5950c2"  
  }  
}
```

## Response with Error

```
{  
  "status": "error",  
  "message": "pass not found"  
}
```

## Void Pass

### Description

Void the pass with the given serial number.

### Request

PUT /api/v1/void/\$passtype/\$serial HTTP/1.1

Host: pass.eddmi.com

### URL Parameters

\$passtype: The pass type id.

\$serial: The pass serial number.

## Example

---

### Request

PUT  
/api/v1/void/pass.com.codigoirreverente.pointscard/dbfb9b4cff1d5a033cd9b258e1216f4  
HTTP/1.1

Host: pass.eddmi.com

## Payload

```
{  
  "voided": "true" // Optional because is the default value  
  "voidInfo": "API Test"  
}
```

## Response with Success

```
{  
  "status": "success",  
  "message": "pass has been voided because API Test"  
}
```

## Response with Error

```
{  
  "status": "error",  
  "message": "pass not found"  
}
```

## Response with Not Modified

```
{  
  "status": "not modified",  
  "message": "pass is voided"  
}
```

## Unvoid Pass

### Description

Unvoid the pass with the given serial number.

### Request

```
PUT /api/v1/void/$passtype /$serial HTTP/1.1
```

Host: pass.eddmi.com

### URL Parameters

\$passtype: The pass type id.

\$serial: The pass serial number.

## Example

---

### Request

```
PUT  
/api/v1/void/pass.com.codigoirreverente.pointscard/dbfb9b4cff1d5a033cd9b258e121f4a  
HTTP/1.1
```

Host: pass.eddmi.com

### Payload

```
{  
  "voided": "false"  
  "voidInfo": "Unvoid API Test"  
}
```

### Response with Success

```
{  
  "status": "success",  
  "message": "pass has been unvoided"  
}
```

### Response with Error

```
{  
  "status": "error",  
  "message": "pass not found"  
}
```

### Response with Not Modified

```
{  
  "status": "not modified",  
  "message": "pass is not voided"  
}
```

## Responses

---

### Codes

Our response codes use the HTTP format.

Code	Message
100	Continue
101	Continue
200	OK

201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
305	Use Proxy
306	(Unused)
307	Temporary Redirect
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required'

412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Long
415	Unsupported Media Type
416	Requested Range Not Satisfiable
417	Expectation Failed
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	HTTP Version Not Supported
600	Insufficient funds

### **Format**

All the responses have at least two fields: status and message. All the responses that have data to retrieve also include the field data. Below you can see some examples.

### **Response without data to retrieve**

```
{  
  "status": "error",  
  "message": "pass not found"  
}
```

## Response with data to retrieve

```
{
  "status": "success",
  "message": "dynamic fields retrieved",
  "data": [
    "Points",
    "Name"
  ]
}
```

## Tests

---

### Web Platform

The best way to test our platform is to create an account. This way you can test all the functionalities and get total control of the data.

Sign Up: <https://pass.eddmi.com>

### Postman Collection

For those who don't have the time to create the platform account we provide a collection to use on the Postman – REST Client extension for Chrome using a given API Token.

If you are not familiarized with Postman you can download the plugin and make some tests with the available collection.

Postman: <http://www.getpostman.com>

API Token: 247a00c98fb293d7e91bc24f1ef308b5

Collection: <https://www.getpostman.com/collections/d60b9e0f0c5b4be84fab>

All the other required data, such as ids and serial numbers, it will be obtained after execute the first operations.

## **Feedback**

Note that the platform is in a development phase, so some of the functionalities and the data for test purposes can be change at any time without previous notice.

If you find some error during the tests we appreciate that you send us a reports with as much detail as possible to the email [support@edmi.com](mailto:support@edmi.com).

## **Change Log**

---

**Version 1.0 (draft)**

# APPENDIX II



## Users Survey

### 1. Do you work on IT or Computer Science?

This question is to evaluate your IT expertise.

#### Results

Options	Value	%
1 - No	21	84
2 - Yes	4	16

### 2. What do you think about the platform design?

Consider the frontend and the backend.

#### Results

Options	Value	%
1 - Very Bad	0	0
2 - Bad	1	4
3 - Acceptable	6	24
4 - Good	12	48
5 - Very Good	6	24

### 3. In your opinion what is the degree of difficulty in creating a template?

Did it take too long to understand the template designer?

#### Results

Options	Value	%
1 - Very Hard	5	20
2 - Hard	3	12
3 - Normal	3	12
4 - Easy	7	28
5 - Very Easy	7	28

#### 4. Do you think that a wizard could simplify the template design?

Did you miss a wizard to create the design?

##### Results

Options	Value	%
1 - Not Much	0	0
2 - A little	0	0
3 - Some	3	12
4 - Much	19	76
5 - Very Much	3	12

#### 5. What new feature would you like to see implemented?

What do you miss?

##### Results

Options	Value	%
1 - API Kits	1	4
2 - A Knowledge Base	2	8
3 - Create vCards	5	20
4 - Design Wizard	17	68

#### 6. What is your opinion about the platform in general?

What is your overview?

##### Results

Options	Value	%
1 - Very Bad	0	0
2 - Bad	0	0
3 - Acceptable	3	12
4 - Good	14	56
5 - Design Wizard	8	32

## 7. How much would you pay for this service?

Monthly value in €.

### Results

Options	Value	%
10 - 20	13	52
20 - 30	4	16
30 - 40	4	16
40 - 50	2	8
50 or +	2	8

## 8. Do you think that with the right apps the digital wallets can replace the traditional ones?

This question is not related with the platform, but can influence our next step.

### Results

Options	Value	%
1 - No	4	16
2 - Maybe	8	32
3 - Yes	13	56



# APPENDIX III



**CRITICUE** 

<http://www.criticue.com>

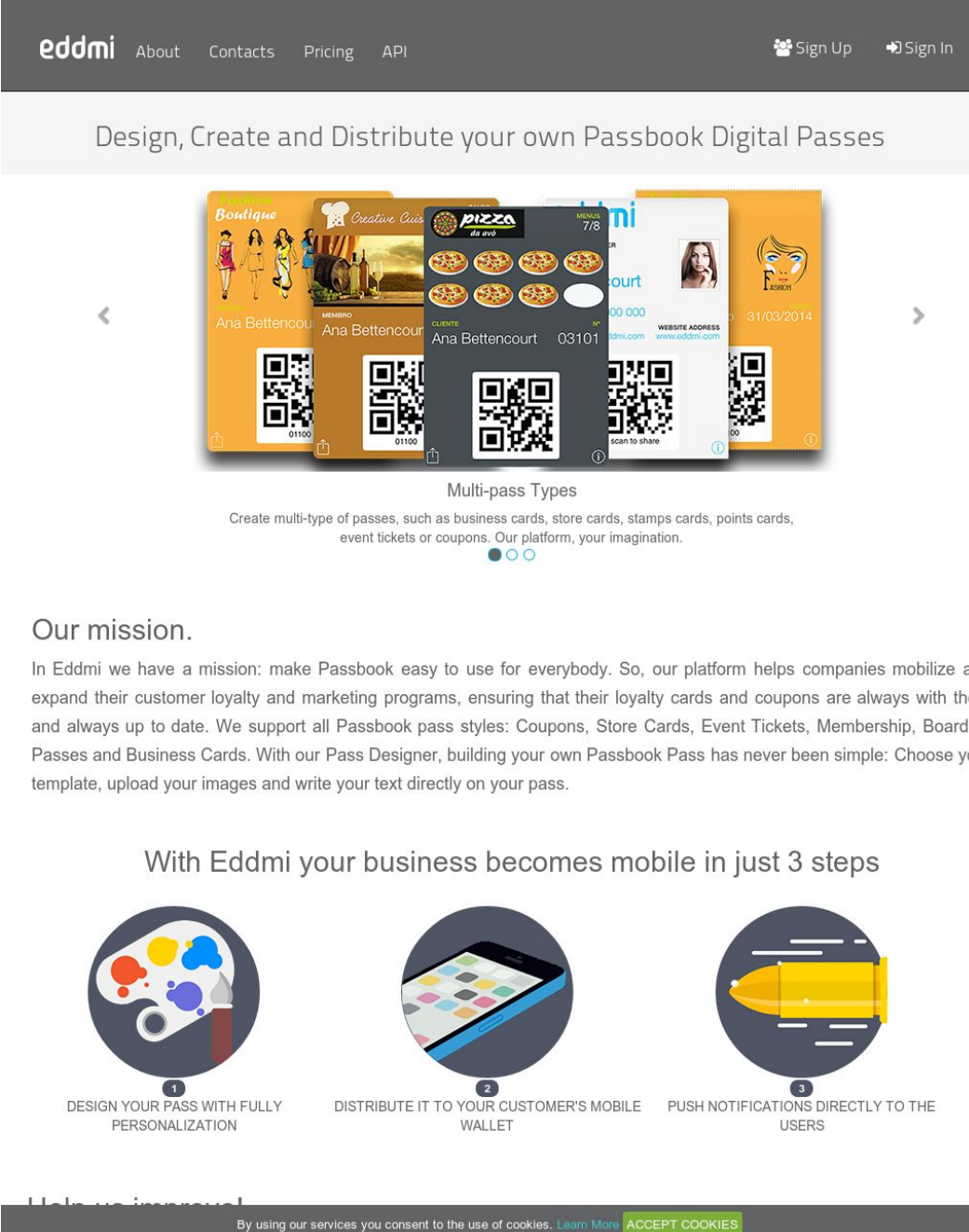
## Resume

This document presents the Criticue's users reviews made to the presented site during the period presented below.

Application URL: <http://test-pass.eddmi.com>

Submission date: 2014-06-01

Latest Review: 2014-08-28



The screenshot shows the Eddmi website interface. At the top, there is a navigation bar with the Eddmi logo and links for About, Contacts, Pricing, and API. On the right side of the navigation bar, there are links for Sign Up and Sign In. Below the navigation bar, a main heading reads "Design, Create and Distribute your own Passbook Digital Passes". Underneath this heading, there is a carousel of five different digital pass designs: a boutique membership card, a creative cuisine membership card, a pizza menu card, a business card, and a fashion membership card. Below the carousel, the text "Multi-pass Types" is displayed, followed by a sub-heading "Create multi-type of passes, such as business cards, store cards, stamps cards, points cards, event tickets or coupons. Our platform, your imagination." and three small circles. Below this, the text "Our mission." is followed by a paragraph explaining Eddmi's mission to make Passbook easy to use for everybody. Below the mission statement, the text "With Eddmi your business becomes mobile in just 3 steps" is followed by a three-step process diagram. The first step is "DESIGN YOUR PASS WITH FULLY PERSONALIZATION" represented by a paint palette icon. The second step is "DISTRIBUTE IT TO YOUR CUSTOMER'S MOBILE WALLET" represented by a smartphone icon. The third step is "PUSH NOTIFICATIONS DIRECTLY TO THE USERS" represented by a yellow pushpin icon. At the bottom of the screenshot, there is a footer with a cookie consent message: "By using our services you consent to the use of cookies. Learn More ACCEPT COOKIES".

## Reviews

The site takes a while to load. There were words over the main image but I could not read them. The bullet in the #3 I'd no5t completely inside the circle like the other pictures.



**ktyler8**

175 reputation

Review time: 2m 49s | 1 month ago

---

I think the homepage could do with more colour. The section with the image slideshow could have a coloured background such as the light blue in the logo. it would make the page more appealing

Instead of **you are not a developer**, you could have '**you're** not a developer' as it is less of a mouthful to read.

I like the footer with social links

The pricing page is really nice with the different colours, but i think the price should be bigger as it is very important. I thin it should be the same size and font style as the decrepition of the plan ( For Personal Use).

Clear navigation and neat animations make the webs pleasurable to use.



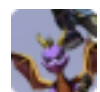
**adam.marsh.3386**

64 reputation

Review time: 9m 27s | 1 month ago

---

The site features a new orange look...I think your doing a really great job...the site looks pretty cool...I think you should however, move the logo from too close of the header links



**ceo\_kirk**

58 reputation

Review time: 1m 12s | 1 month ago

---

I love this concept. I wish more companies offered Passbook style offerings.

Although am reasonably confident that, as a business owner, I would be willing to go into your production site and play around until I came up with something I liked, I know many more who are just plain scared of even trying. Is your company able to offer design assistance? If so, that might be a hook to use to get those less comfortable to try.

I like the graphics, the easy to read font and color design.



**fark**

114 reputation

Review time: 5m 28s | 1 month ago

---

First thing, the space is quite far between Hello, Passes! paragraph and Help us improve! Add a bit more image to the front page would be more eye catching. The gray background beneath the dark gray bar where Home is settled could be thinner. Over expression is simple and coupled with a clean color theme.



**casktdirect**

122 reputation

Review time: 9m 49s | 1 month ago

---

The design is quite nice but I'd put some more time into the content and layout. It's not obvious from the image at the top what it actually is. The titles are not very clear either. Maybe include one line that says EXACTLY what it is in a large font? I'd also suggest reviewing the size of your breadcrumb text. Looks nice overall though :)



**owenr88**

128 reputation

Review time: 3m 51s | 1 month ago

---

overall a nice website but there are a few things that do not look right to me <http://test-pass.eddmi.com/img/nav-logo-b.png> is better than the white and needs some serious size reduction - forget hover and stick to blue version. Check the sign in and sign up buttons in other browsers - in Firefox looks as if they need some padding added.



**brygee**

508 reputation

Review time: 5m 32s | 1 month ago

---

Seems straight and concise. I would however recommend the changing of the TOP grey bar, as it clashes with everything else around it. Otherwise, it seems straight to the point and works wonders for the product that you're trying to advertise.



**guest**

4 reputation

Review time: 1m 15s | 1 month ago

---

Though i am not conversant with the site language, but i think the design is unique and i have not seen. is it built on wordpress?



**bonpriya**

34 reputation

Review time: 1m 23s | 1 month ago

---

First of all I am not a developer. That being said I think your message on your homepage needs to be stronger. The image itself (to me) dosent convey your product application. Maybe move your message above the image to give better context to the images your using.



**tova**

121 reputation

Review time: 2m 58s | 1 month ago

---

The first thing I noticed was the images with cookies :) I thought 'yammy!', then I opened your site. There's too much gray - there's nothing wrong with it, but its kind of default color. Try adding another, bolder color to evoke visual interest. It's not clear what you and your website offers - use simple words and sentences to describe it in as fewer words as possible. Slider - it should pause when user hovers the images, the first picture is OK, the rest could be higher quality. Try using different, more accentuated font for headings to make them more visually intriguing. Light grey footer needs more spacing at the bottom. There's horizontal scroll when website is at 320px width. You use different font size for similar elements when comparing Home and other pages, use similar or same ones.



**kate-an**

121 reputation

Review time: 26m 56s | 1 month ago

---

I think the 'home' on top is a bit awkward there. You generally don't have breadcrumbs on the homepage. Try reducing the font size on navigation a bit more. Work on creating a style guide for the website. It looks a bit dull and could be nicer.



**rowbert**

110 reputation

Review time: 2m 41s | 1 month ago

---

The color scheme looks great, but the carousel at the top can be a bit awkward to use. The lights don't seem to light up fast enough (under the carousel). The footer bar also covers up the bottom of your page and leaves it unreadable. The page is also small which makes the resizing of the top bar feel really weird.



**zlawrence**

4 reputation

Review time: 1m 58s | 2 month ago

---

I think this looks pretty awesome. The menu looks nice, and the font really fits. It's a sort of elegance. I'm not sure how it becomes after you've signed in but it looks pretty good from here.



**potchi**

15 reputation

Review time: 56s | 2 month ago

---

My first impression was that I had no idea what the site was about. Was it selling pizza? Websites? I clicked "About," and while it did make it more clear, it was a struggle. The text sounds in places to have been written by someone who doesn't speak English as their primary language ("Our platform will manager our customers clients" should be "Our platform will manage our customer's clients"), and in some places, it's overly technical/wordy ("the cost of marketing dissemination (emails, sms, mms), is astronomical and unbearable"). I'd focus on making this content more simple to understand.



**chrishuff**

447 reputation

Review time: 5m 56s | 2 month ago

---

It should be written "It's ok" :) but I like the page design and theme



**monika64**

12 reputation

Review time: 1m 46s | 2 month ago

---

The overall site is very good, although I think you could be more clear on exactly what the business hopes to achieve as it wasn't 100% clear to me, especially on the about screen - some of it doesn't read correctly.

From a design perspective I would fix the footer to the bottom of the screen, as this is where it appears on the home and about screens, but it floats up the screen on the contacts section.



**stephen01**

74 reputation

Review time: 9m 10s | 2 month ago

---

Design is quite nice! But the transition especially between the slideshow and "Hello, Passes!" seems a bit awkward to me. The blocks don't look like they belong on the same page.



**hubguitar.com**

24 reputation

Review time: 42s | 2 month ago

---

The gray with white text on gray text within gray message areas lead to everything blending together. I do however like your image in the slider as it gives a great example of what the cards could look like.



**linkibl**

88 reputation

Review time: 2m 25s | 2 month ago

---

First of all, you have to add here backlight of current category, it will improve navigation at the site and it will look a lot better and more professional. Second one, you should invent here a professional logo, it will be a lot more friendly for viewers and your site will look more modern.



**dawergo**

14494 reputation

Review time: 1m 1s | 2 month ago

---

does not look very mobile friendly so far... I think you should consider adding SVI and more vectorial graphics to get the visitors interested. and if you consider yourself a UI and UX developer, some fancy models might help.



**cosmin.dina.73**

141 reputation

Review time: 8m 14s | 3 month ago

---

i could not get the content of the website. but seams you used a carrousel. so it's a bit nice.. the colors are also great too.. i just think where did you used "hello passes" its not ok.



**macheidan**

20 reputation

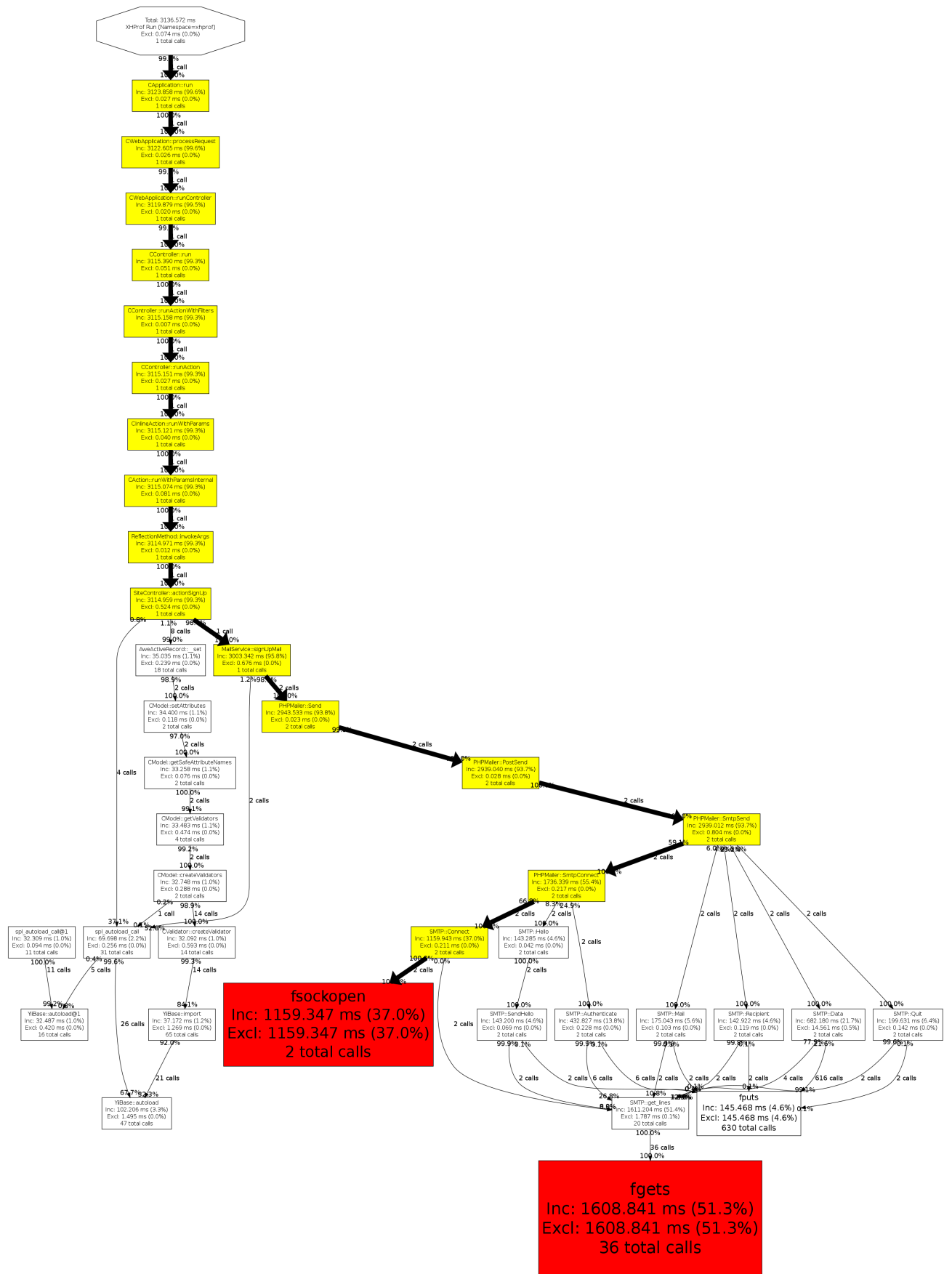
Review time: 1m 36s | 3 month ago

---

## APPENDIX IV







# APPENDIX V



eddm | Home

test-pass.eddmi.com

Apps ISEP Code Bits Web Tools Eddmi | CI Tese Utis TQ Cheat Sheets Dopelists 3 New JavaScript API PeetHaa/OpCacheGUI amnuts/opcache-gui Other Bookmarks

eddm About Contacts Pricing API Sign Up Sign In

## Design, Create and Distribute your own Passbook Digital Passes

Multi-pass Types

Create multi-type of passes, such as business cards, store cards, stamps cards, points cards, event tickets or coupons. Our platform, your imagination.

Why work with us?

In Eddmi we have a mission: make Passbook easy to use for everybody. So, our platform helps companies mobilize and expand their customer loyalty and marketing programs, ensuring that their loyalty cards and coupons are always with them and always up to date.

We support all Passbook pass styles: Coupons, Store Cards, Event Tickets, Membership, Boarding Passes and Business Cards. With our Pass Designer, building your own Passbook Pass has never been simple: Choose your template, upload your images and write your text directly on your pass.

eddm | Home

test-pass.eddmi.com

Apps ISEP Code Bits Web Tools Eddmi | CI Tese Utis TQ Cheat Sheets Dopelists 3 New JavaScript API PeetHaa/OpCacheGUI amnuts/opcache-gui Other Bookmarks

eddm About Contacts Pricing API Sign Up Sign In

## Why work with us?

In Eddmi we have a mission: make Passbook easy to use for everybody. So, our platform helps companies mobilize and expand their customer loyalty and marketing programs, ensuring that their loyalty cards and coupons are always with them and always up to date.

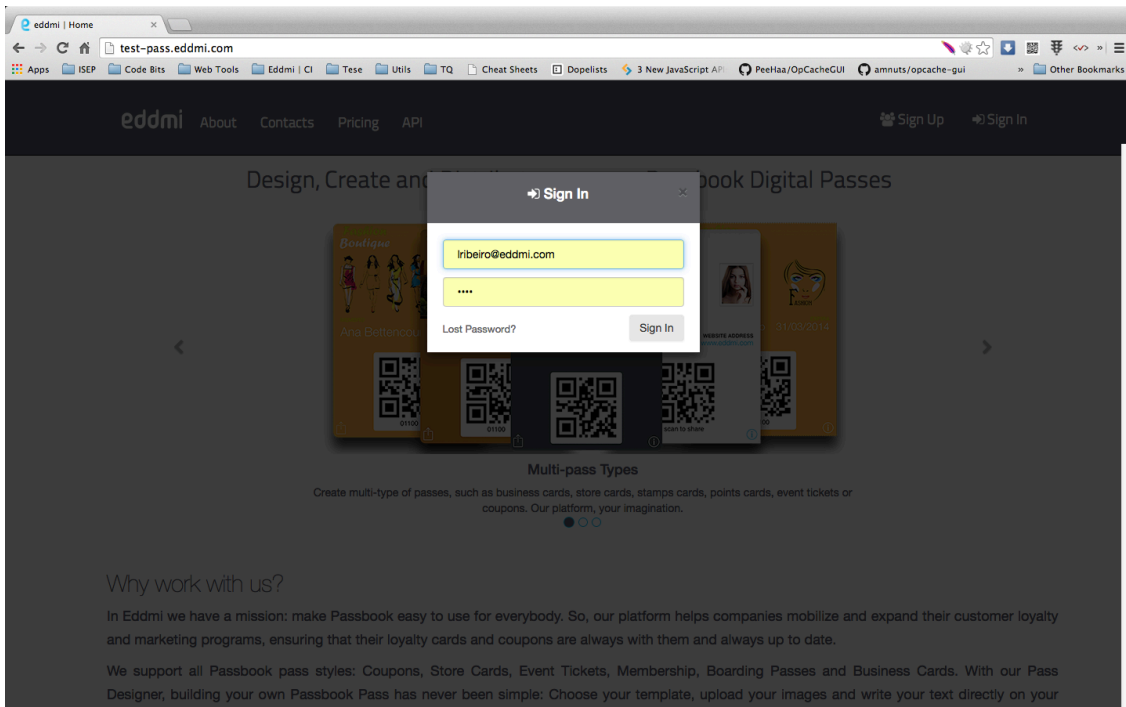
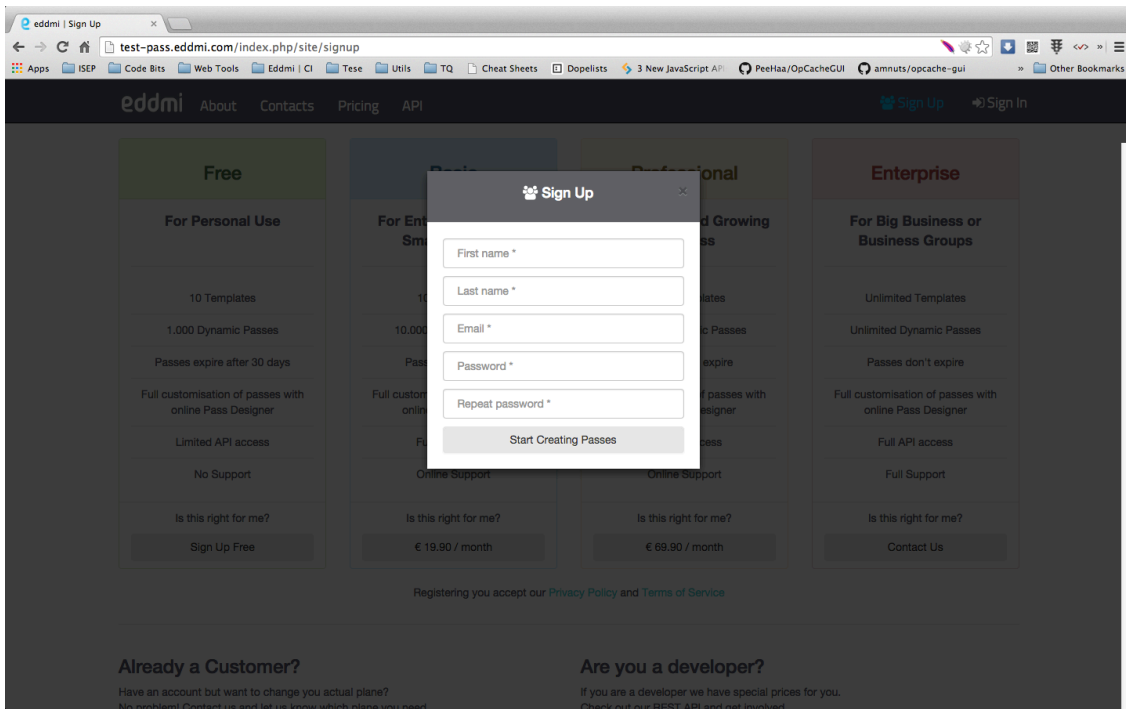
We support all Passbook pass styles: Coupons, Store Cards, Event Tickets, Membership, Boarding Passes and Business Cards. With our Pass Designer, building your own Passbook Pass has never been simple: Choose your template, upload your images and write your text directly on your pass.

### With Eddmi your business becomes mobile in just 3 steps

- Design your pass with fully personalization
- Distribute it to your customer's mobile wallet
- Push notifications directly to the phone

Help us improve!

We are searching for beta testers for our platform. Want to help? Sign up, give it a try, and give us your feedback. It's free.



eddm | Templates

test-pass.eddmi.com/index.php/passtemplate/index

Apps ISEP Code Bits Web Tools Edm | CI Tese Utis TQ Cheat Sheets Dopelists 3 New JavaScript AP PeetHaa/OpCacheGUI amnuts/opcache-gui Other Bookmarks

eddm Dashboard Templates Passes Issues

Luís Ribeiro

## Templates

Dashboard / Templates

Wizard New

10

Name	Pass Type	Pass Layout	Creation Date	Update Date	Serial Number Option	Actions
Business Card	Visit Card	Visit Card	2014-10-22 17:45:16	2014-10-22 17:54:16	Auto generated serial	👁️📄🗑️
CI Business Card	Visit Card	Visit Card	2014-10-22 18:02:41	2014-10-22 18:03:34	Auto generated serial	👁️📄🗑️

Showing 1 to 2 of 2 entries

1

eddm | Templates

test-pass.eddmi.com/index.php/passtemplate/index

Apps ISEP Code Bits Web Tools Edm | CI Tese Utis TQ Cheat Sheets Dopelists 3 New JavaScript AP PeetHaa/OpCacheGUI amnuts/opcache-gui Other Bookmarks

Template successfully updated. 1 passes have been updated.

eddm Dashboard Templates Passes Issues

LUIS Ribeiro

## Templates

Dashboard / Templates

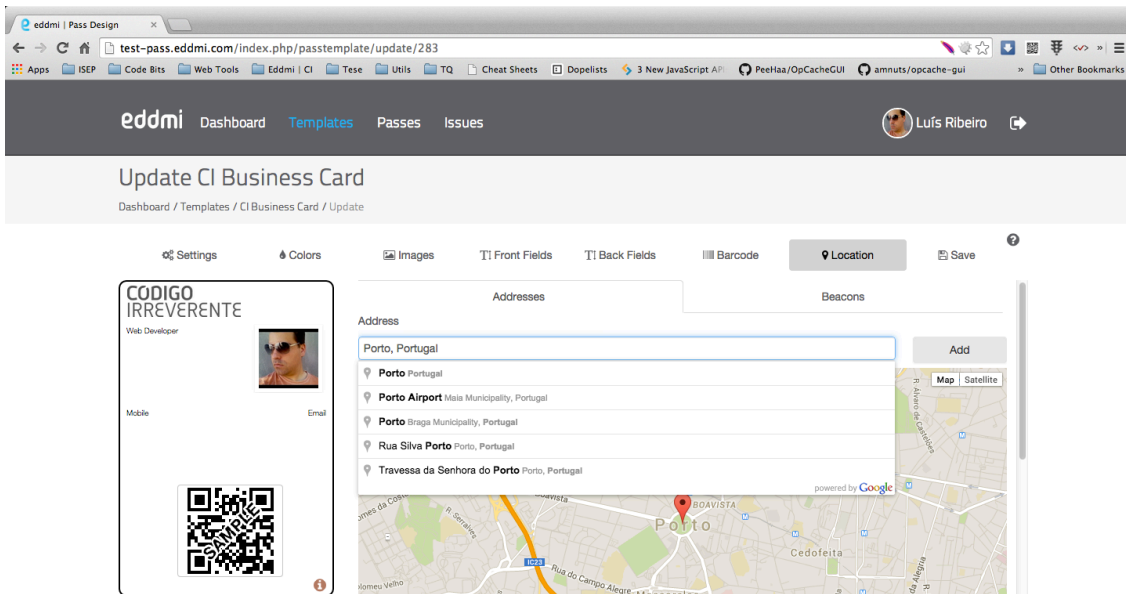
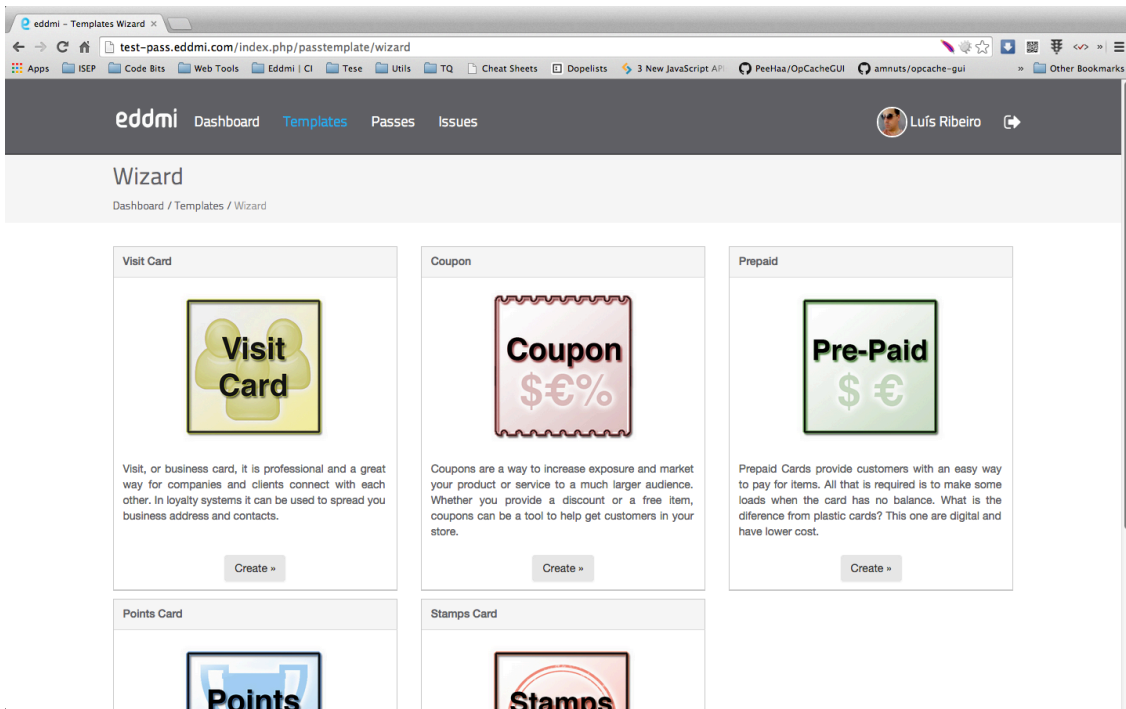
Wizard New

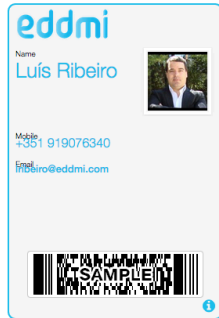
10

Name	Pass Type	Pass Layout	Creation Date	Update Date	Serial Number Option	Actions
Business Card	Visit Card	Visit Card	2014-10-22 17:45:16	2014-10-22 19:39:12	Auto generated serial	👁️📄🗑️
CI Business Card	Visit Card	Visit Card	2014-10-22 18:02:41	2014-10-22 18:03:34	Auto generated serial	👁️📄🗑️

Showing 1 to 2 of 2 entries

1





### General Informations

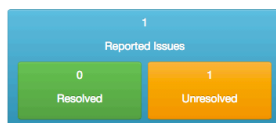
Name	Business Card
Id	925447e061b0e3c
Serial Number	4abd8b35033945878578183db89d0a33
Organization	Eddmi
Template	Luís Ribeiro Eddmi Business Card
Pass Type	Visit Card
<b>Dates</b>	
Creation Date	2014-10-22 17:50:41
Update Date	2014-10-22 19:39:17
Expire Date	--

### Statistics

Downloads  
**8**

Linked Devices  
**1**

### Statistics



### Latest Report

**Title:** Functional Test  
**Date:** 2014-09-12 00:45:35  
**Browser:** Chrome on Mac OS  
**Comment:** I'm performing a functional test to assure that this feature works good.  
**Status:** Unresolved

### New Issue

**Title \***

**Platform \***

**Browser \***

**Comment \***



### Luís Ribeiro

**Username/Email** lribeiro@eddmi.com

**Joined** 11-09-2014 23:28

**Last Update** 22-10-2014 17:32

**Last Login** 22-10-2014 19:37

### Free Account

**API Token** c6964f7061238b7e0dc0ee6b6c1be947

**Last Created Pass** CI Business Card

**Last Updated Pass** CI Business Card

**Last Archived Pass** --

2 Active Passes

0 Archived Passes

2 Linked Devices



Should be at least 150x150 px

### Luís Ribeiro

### Account

Free

Basic

Professional

Enterprise

Developer

Update

Account changes will only become effective after our commercial team analysis