UNIVERSITÉ IBN ZOHR
Faculté des Sciences Agadir

Département d'Informatique

Filière Sciences Mathématiques et Informatique

PROJECT DE FIN D'ÉTUDE

Présenté Par: Taha Bouhsine

Pour l'obtention de la
Licence en Sciences Mathématiques et Informatique

Design And Full Stack Development Of A Crowdfunding Platform Sahem

---

Encadré par : Dr. BELAQZIZ Salwa

Année universitaire : 2019 - 2020

Dedication

In memoriam Hanya Laaiba, my dear grandmother,
her last words of encouragement are living with me to this day,
feeding my motivation, and always urging me to pursue my dreams, and my studies,
never to be forgetten, never to be erased,
may your soul rest in peace,
may your memory guide me trough my future steps to the unknown

To your beautiful soul,
I dedicate this work.

Taha BOUHSINE

Thanks

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec ultricies varius velit non accumsan. Aliquam erat volutpat. Suspendisse potenti. Proin pretium nibh eu urna molestie, ut eleifend ligula maximus. Sed rutrum, sapien eu laoreet semper, libero arcu tempor felis, vitae aliquet mauris sem vitae lectus. Ut vulputate elementum lacus, sit amet gravida lacus placerat vel. Fusce sed aliquet felis. Integer vitae sodales felis. Phasellus a turpis elementum, interdum mauris sed, ullamcorper orci. Vivamus blandit mauris enim, a mollis quam cursus et. Nulla volutpat dapibus eros, ut rutrum orci bibendum iaculis. Curabitur orci elit, venenatis at neque sit amet, venenatis molestie erat. Maecenas luctus sit amet arcu in ornare. Curabitur venenatis, tellus congue sodales finibus, sem nulla sodales nunc, et luctus quam lacus vitae odio.

Contents

List of Figures

List of Tables

Acronyms

Introduction

One of the biggest topics discussed around the business community for the past few years is the idea of crowdfunding using an online platform. This is a great tool to use for building capital for a startup, funding growth for your company or development of services or products to further your business. New crowdfunding platforms and websites are popping up regularly to meet the needs of the expanding market, with plenty of room to benefit from the advantages of this technology by creating a crowdfunding platform.

Project Presentation

Problem

Our mission is to help bring creative projects to life. privide a platform for people to create fundraisers. funding platform for creative projects. Everything from films, games, and music to art, design, and technology. Kickstarter is full of ambitious, innovative, and imaginative ideas that are brought to life through the direct support of others. backers pledge to projects to help them come to life and support a creative process. To thank their backers for their support, project creators offer unique rewards that speak to the spirit of what they're hoping to create. A platform will allow for gatekeeping that will monitor and create symbiotic relations with other algorithms and information online. This can be done by using cloud-based solutions for better

access. we will be able to coordinate multiple campaigns easier. we will be able to find people who are willing to invest with little equity involved. our platform can help leap the hurdle of lack of experience in each field. Letting the barrier of entry be lessened for everyone who has a dream. funding Model. “Keep It All”

Objectifs

build a crowdfunding platform whilist keeping in mind the following aspects.

1. User Experience Making sure that our crowdfunding software solutions are easy to navigate for the end-user is a crucial part of the development process. If our customers are befuddled by how to navigate our application, then it is highly likely that they will leave and find another crowdfunding platform. Not only will we want a functional interface, but one that is eye-catching as well. One thing to take note of is the importance of giving them the terms and conditions within the first few steps, so they fully understand what to expect. This shows a dedication to transparency that many startups and entrepreneurs will appreciate.
2. Account Management Our customers will want to know what is going on, and making it easy on them will help make our platform successful. That means setting up systems that make it very clear what’s going on with their project. We will want them to be able to access who has been investing, how much money they have, how far they are from their goal, and any other metric they need to run a successful campaign. This could also include reports for recharges, withdrawals all available via a simple to navigate dashboard.
3. Report Generation As the platform owner, you need a way to benefit from your time spent on creating this site that will help so many people. So, ensuring you also have access to backed reports like rewards, investors, and such can help you help them, as they say. This means creating a dashboard that, just like for the actual campaign creators, is easy to navigate and gives you access to reports you can use to course-correct and upgrade the systems.
4. Payment Gateway & Marketing When starting a crowdfunding platform you will want to set it up with access to the right payment gateways. Each gateway has its features, and

so doing some in-depth research into them will allow you to choose one or several that works for the largest number of potential customers.

Document Organization

Crowdfunding

Crowdfunding is an increasingly popular alternative method of raising finance. But what is crowdfunding? In this chapter we explain what crowdfunding is, how it works, the risks and rewards and Morocco's regulations on it.

Definition

Crowdfunding is the practice of raising money from a large number of individuals for the purposes of financing a project, venture, business or cause. Traditionally, crowdfunding has been carried out via subscriptions, benefit events and door-to-door fundraising. However, today the term is typically associated with raising money through website platforms, which allows crowdfunding to reach a larger pool of potential funders.

History

Crowdfunding has a long history with several roots. Books have been crowdfunded for centuries: authors and publishers would advertise book projects in praenumeration or subscription schemes. The book would be written and published if enough subscribers signaled their readiness to buy the book once it was out. The subscription business model is not exactly crowdfunding, since the actual flow of money only begins with the arrival of the product. The list of subscribers has, though, the power to create the necessary confidence among investors that is needed to risk the publication.

War bonds are theoretically a form of crowdfunding military conflicts. London's mercantile community saved the Bank of England in the 1730s when customers demanded their pounds to

be converted into gold - they supported the currency until confidence in the pound was restored, thus crowdfunded their own money. A clearer case of modern crowdfunding is Auguste Comte's scheme to issue notes for the public support of his further work as a philosopher. The "Première Circulaire Annuelle adressée par l'auteur du Système de Philosophie Positive" was published on 14 March 1850, and several of these notes, blank and with sums have survived. The cooperative movement of the 19th and 20th centuries is a broader precursor. It generated collective groups, such as community or interest-based groups, pooling subscribed funds to develop new concepts, products, and means of distribution and production, particularly in rural areas of Western Europe and North America. In 1885, when government sources failed to provide funding to build a monumental base for the Statue of Liberty, a newspaper-led campaign attracted small donations from 160,000 donors.

The late 19th century saw the creation of one of the world's most recognisable landmarks with the gift of the Statue of Liberty by the French to the US. While the French paid for the construction and shipping of the statue it was down to the US to fund the base upon which it would stand. With the statue ready to leave France, the Americans were still well short of the $300,0001 needed to build the base and erect the statue. Running short of time the American Committee (responsible for raising the funds) teamed up with newspaper owner Joseph Pulitzer to launch a campaign to invite citizens to donate even small amounts to help in the funding of the pedestal, offering donors miniature replicas of the statue in return. This 19th century crowdfunding campaign raised $100,000 in just five months, contributed to one of the most popular attractions in the world and illustrated the financing power of a large crowd when tapped for funding. While the American Committee were lucky to have Mr Pulitzer and his paper to publicise their plea for donations, others wishing to access so many people would have had no such help. This, however, has changed in recent years with the rise of social media and the new ease with which communities can form and interact online.

Crowdfunding on the internet first gained popular and mainstream use in the arts and music communities. The first noteworthy instance of online crowdfunding in the music industry was in 1997, when fans of the British rock band Marillion raised US$60,000 in donations by means of an Internet campaign to underwrite an entire U.S. tour. The band subsequently used this method

to fund their studio albums. This built on the success of crowdfunding via magazines, such as the 1992 campaign by the Vegan Society that crowdfunded the production of the ”Truth or Dairy” video documentary. In the film industry, independent writer/director Mark Tapio Kines designed a website in 1997 for his then-unfinished first feature film Foreign Correspondents. By early 1999, he had raised more than US$125,000 on the Internet from at least 25 fans, providing him with the funds to complete his film. In 2002, the ”Free Blender” campaign was an early software crowdfunding precursor. The campaign aimed for open-sourcing the Blender 3D computer graphics software by collecting $100,000 from the community while offering additional benefits for donating members.

The first company to engage in this business model was the U.S. website ArtistShare (2001). As the model matured, more crowdfunding sites started to appear on the web such as Kiva (2005), IndieGoGo (2008), Kickstarter (2009), GoFundMe (2010), Microventures (2010), and YouCaring (2011).

The phenomenon of crowdfunding is older than the term ”crowdfunding”. According to wordspy.com, the earliest recorded use of the word was in August 2006.

Before Crowdfunding, the Peer-To-Peer Lending Era

Peer-to-peer lending, also abbreviated as P2P lending, is the practice of lending money to individuals or businesses through online services that match lenders with borrowers. Peer-to-peer lending companies often offer their services online, and attempt to operate with lower overhead and provide their services more cheaply than traditional financial institutions.[citation needed] As a result, lenders can earn higher returns compared to savings and investment products offered by banks, while borrowers can borrow money at lower interest rates,[1][2][3] even after the P2P lending company has taken a fee for providing the match-making platform and credit checking the borrower.

Functionality

The platforms operate by allowing those seeking finance to make a pitch on the site outlining how much money they need, what they need it for and what, if anything, you get in return for contributing. Potential funders can then view pitches on the platform, interact with both those looking for finance and other potential funders and then decide whether or not they want to back the campaign. The majority of platforms operate the all–or–nothing model where, if the target amount is not raised within a given timeframe, contributions are returned to funders and no financing goes ahead.

Crowdfunding Types

Crowdfunding facilitates the raising of capital for a variety of purposes, using numerous variations of the model. Below is a typology of how the operators in the market can potentially be segregated. The majority of platforms can be categorised under these four types, but there are several variations, such as hybrid models and those platforms that define themselves in a sectoral vertical rather than by the type of finance they provide.

1. Donation Model: The donation model of crowdfunding is a means for charities, or those who raise money for social or charitable projects, to gather a community online and to enable them to donate to a project. While most established charities facilitate this through their own website, crowdfunding is popular for small organisations and people raising money for personal or specific charitable causes. Popular sites include Crowdrise and Causes.
2. Lending Model: Crowdfunded lending is largely an evolution of the peer–to–peer model of lending, pioneered by firms such as Lendingclub and Zopa. Projects or businesses seeking debt apply through the platform uploading their pitch, with members of the crowd taking small chunks of the overall loan. Some platforms focused on social causes offer interest–free loans such as micro–lending site Kiva. Others operate more as an investment, where interest rates are decided either by those seeking the loan or using a market for loan parts, such as that used by UK platform FundingCircle.

3. Reward Model: The most popular form of crowdfunding to date has been the reward model which has grown significantly in the funding of creative, social and entrepreneurial projects. The model allows people to contribute to projects and receive non–financial rewards in return, usually operating a tiered system where the more you donate the better the reward you receive. The model often closely resembles philanthropy with the donation far exceeding the monetary value of the reward or the reward costing the fundraiser little, such as experience or recognition–related rewards. For some projects the model is similar to a presale agreement. In these cases entrepreneurs or artists crowdfund the production cost of their record, movie, game or product and allow the donors to be the first recipients once the production is complete. Popular platforms operating the reward model are Kickstarter and Indiegogo in the US and Peoplefund.it in the UK.
4. Investment Model: The final type is the application of crowdfunding to investing for equity, or profit/revenue sharing in businesses or projects. This form of the model has been the slowest to grow due to regulatory restrictions that relate to this type of activity. Some European platforms have been pioneers of the equity crowdfunding model, allowing anyone to take a small stake in an unlisted or private business through crowdfunding. The most popular sites in offering this model are CrowdCube in the UK and Symbid in the Netherlands. Others such as Quirky offer a revenue or profit–sharing model allowing you to capitalise on the success of the projects you back.

How big is crowdfunding?

In 2015, crowdfunding raised $34 Billion worldwide (Source: Massolutions) and the total had doubled every year for the previous four years! Many crowdfunding campaigns are for quite small sums but some are very large. The model of crowdfunding which raised the most money is the lending model. In the UK crowdfunding in 2015 totalled £1,112 million and this figure is expected to continue to rise.

Why is crowdfunding growing?

Making a public call to fundraise from the crowd is not a new idea but crowdfunding as we now understand it has grown very quickly for a number of reasons since its emergence in the 1990s. These include:

1. Technological developments : The emergence of wide and low-cost access to the internet and communication tools like social media mean it is easier and cheaper for us to reach out to much more widely dispersed and larger groups of people.
2. Societal changes : These technical changes have also empowered us to take on new activities which were once controlled by gatekeepers. We can see this in the way people write and publish books, publish music, writing blogs, and the general sharing of our lives online. Crowdfunding is just the financial manifestation of this sense of empowerment and the ability to take "ownership" of a process. At the same time, we are also increasingly comfortable with transacting financially online be it shopping and e-commerce or checking our bank balances. This confidence to use money online is essential for the growth of crowdfunding.
3. Economic factors : The other key factor in the extraordinary growth of crowdfunding is that after the financial crisis of 2008 access to funding has been more problematic and so people are exploring alternatives to the traditional sources. At the same time interest rates have fallen to historically low levels and so "retail investors" are looking for better places to put their investments and some crowdfunding campaigns seem to offer better returns than would be available on the high street.

Principe

The other important variation in crowdfunding is the distinction between what are known as the "Keep It All" and "All Or Nothing" models.

In a "Keep It All" campaign, you keep everything you raise regardless of whether you reach your target or not.

In an "All Or Nothing" campaign you only get to keep what you raise if you succeed in reaching

your target.

Crowdfunding can be undertaken by both individuals and organisations.

Downsides of Crowdfunding

1. It is not easy – to be a success takes a lot of time and effort to carry out continued promotion of your campaign.
2. Many campaigns are unsuccessful and the successful ones take work, preparation and effort to make them happen – there is no guarantee of success but work, preparation and effort can increase the chances of success.
3. It is also a very public process and so you must be prepared to be open and honest in a public arena and expect brickbats and bouquets in equal measure – it is important to accept that there may be public scrutiny of your project and prepare for this.
4. Reputation considerations: there are a number of aspects to this, including your appearance on the platform in the first place. Might customers and others view it negatively? There are also significant expectations from investors if your crowdfunding campaign is successful. If it meets delays or runs into problems, the reputation of your company might become damaged. Finally, by putting your idea onto a crowdfunding platform before you're ready to bring it to the wider market (i.e. before it's fully developed and tested), you are leaving your company open to public criticism of your plans, idea, or business.

Benefits of Crowdfunding

1. It is a very accessible process which is open to all and can be carried out on your terms
2. you decide on the amount you want to raise and the timescale to raise the funds.
3. You are in control – the promotion and selling of the project is the responsibility of your group.
4. It can bring much more than money – it can attract new people and support (nonfinancial) to your group.

5. What money it does bring can be very different from traditional investment – funds raised through crowdfunding are unrestricted and can be used for all elements of your project.
6. It can be quick – as you are in control you are able to work quickly to start raising funds.
7. You will also produce a very valuable asset for you or your organisation as you run a campaign and that "crowd asset" can be a very useful and enduring resource – people who support your project are also likely to support your group.
8. It's a quick and easy way to get a lot of exposure for your brand and for your new idea or initiative
9. You can engage directly with investors who could also become your customers
10. There is a low barrier to entry – often much lower than with other forms of investment
11. It's an easy way to get feedback on your idea

What makes crowdfunding different from other funding?

Crowdfunding reaches widely by using technology and reduces the size of funding each individual contributor has to come up with. This means that more people can take part. Making it easy for a wider group of people to support a business or project introduces a wider range of motivations for people to back a campaign. This means that there is a range of reasons why people might support you, and not simply for a financial return. The idea of many small contributions making a difference (as opposed to a small number of large contributions) is a concept underpinning lots of online activities which have disrupted traditional industries and it is called "The Long Tail". Because the process of crowdfunding is a very public one and involves many people it can, and does, bring many more advantages than simply money. It can be a powerful campaigning tool, it can build networks, validate an idea, build awareness and many other things, all of which can be very valuable and useful. A good crowdfunding campaign will recognise and target these additionalities. For some entrepreneurial projects, it has the advantage of accelerating the process of setting up a business by allowing the entrepreneur to run in parallel a series of processes like market research, publicity and marketing and fundraising, which have often traditionally been seen as sequential.

Motivation And Rewards

Crowdfunding Actors

in crowdfunding there is two main actors, the project owner that created the fundraiser, and the actor that provide the funds.

1. Creator
2. Funder general public

Existent Platforms

Non-Profit:

1. Fundly: This is probably the best known of the non-profit crowdfunding sites. This platform charges a fee for all campaigns and a small transaction fee for processing. It offers a customizable donation page, with an integrated mobile experience. You can use multiple media formats to promote your cause as well as blogging.
2. Salsa: This p2p platform offers branded marketing materials for every supporter and customizes messages as well. This is a fully integrated system with Salsa CRM. So, if you already use that, software data is transferred easily.
3. Soapbox Engage: This is a social change platform that offers not only the ability to create cause for donation but allows you to create forms, petitions, and events to further your cause.
4. Bonfire: is a crowdfunding platform that allows you to create custom t-shirts to raise money for your projects. You create the t-shirt and then build a custom web page to market and use as a landing spot to drive your social media marketing campaigns.

Business:

1. Kickstarter: This is a crowdfunding platform that allows you to market potential products or businesses for investors to donate to. You create a custom page for your book, game,

or any other creative project and then set a goal and start building funds. Each project will set up designated donations that have rewards attached to them.

2. Indigogo: This is a great sight for startups and creatives that uses similar methods to Kickstarter. You set up a custom page and goals and market your campaign. They have integrated systems to help with the fulfillment of delivery, mobile management, perk options, etc.
3. Seedrs: This crowdfunding platform uses equity investing to raise funds for small businesses and startups. It has three options to invest whether equity, funds or convertible donations and allows other investors a discount in the future.
4. Crowdcube: This platform offers you the chance to invest in business and causes via an easy-to-use website. Creating campaigns via your own customizable page, you will have access to analytics and social media campaigns to help promote your project.

Crowdfunding In Morocco

Expected since April 2018, the examination of the so-called “collaborative financing” bill is finally starting in Morocco. Technically, the concept of crowdfunding (“crowd financing”) is to bring together two needs through a platform on the internet: that of the entrepreneur/idea carrier who lacks liquidity and hopes to find financing, and that of a saver who wants to invest a certain amount of money, even modest, in a project with detailed outlines. These are generally small businesses, but this law will also enable associations to raise funds to finance their projects.

Under the banner of crowdfunding, the law admits three types of financing operations, each time via an electronic platform edited and managed by a collaborative financing company: the financing of projects in the form of a loan (crowdlending, for which Bank Al-Maghrib will control the interest rate or the maximum duration of the loan), a grant (crowdfunding: the donor will have to obtain an authorization if the amount exceeds MAD 500,000), or capital (crowd equity).

In the latter case, for management companies that want to propose capital investment, it will be necessary to obtain a visa from the Moroccan Capital Market Authority (AMMC).

The platforms must have a minimum capital of $31,000 (MAD 300,000) and the collaborative finance companies that will manage them must have a risk prevention and reduction policy.

he majority of respondents 96% do not even know the word crowdfunding. We also noticed that there is only one person who has already contributed to or supported a project through crowdfunding and this has two causes: the first, as we have already mentioned, is that the majority do not know this method of financing and the second is, of course, the absence of a regulation framing this system. We also found that more than half of the respondents are willing to support a project by crowdfunding for a percentage of 50.5%, for those who answered no, it is mainly the lack of information and money that is the reason why they are not ready to support a project via crowdfunding. Regarding the type of project most respondents are ready to support a charitable project, social or solidarity with a percentage of 81.6%, followed by support from a local company with 75.7% and then the development of a business with a high growth potential with a percentage of 59.2%, this shows that Moroccans have the culture of giving and helping, even if the amounts remain low because the majority are ready to support a project for an amount that does not exceed 500 DH. Most respondents plan to have their own projects with a percentage of 62%, while 5% answered I do not know against 33% do not want to have their projects. However, according to the distribution by labor force, we find that this majority consists mainly of the unemployed and students, while more than half of the workers (58.7%) do not want to have their own projects. For those who want to have their own projects, it is mainly the lack of funding (84.7%), and the lack of idea and the risk of failure that prevents them, so crowdfunding can fill this gap is becoming an alternative to financing projects or even a complement. For those who do not want to have their projects they are mainly workers and are satisfied by their work or students who want to finish their studies first. Finally we found that the majority who want to have their own project and who have financing problems for a percentage of 55.5% are interested in crowdfunding to launch or develop their projects, however the lack of information on this system is the first cause for those who are not interested in this new mode of financing, which is normal. So, we can answer the hypothesis that we mentioned in the introduction: What is the future of crowdfunding in Morocco? Would it be a realistic alternative to overcome the lack of funding for startups, very small company - small

and medium company in Morocco? What are the stakes and obstacles that hinder this type of financing? According to the respondents' answers and according to the second part where we have focused on crowdfunding in Morocco, we can say that this new financing mode can find its place in Morocco and its future is certain, and can be an indispensable instrument for the financing of start-up companies and a complement to the traditional financing circuit for very small, medium and small companies, although there are several problems to be solved such as the ignorance or the confusion of the people toward this concept or the absence of regulations governing crowdfunding in Morocco.

According to the respondents's answers and according to the second part where we have focused on crowdfunding in Morocco, we can say that this new financing mode can find its place in Morocco and its future is certain, and can be an indispensable instrument for the financing of start-up companies and a complement to the traditional financing circuit for very small, medium and small companies, although there are several problems to be solved such as the ignorance or the confusion of the people toward this concept or the absence of regulations governing crowdfunding in Morocco.

Project management

Project Development Proccess

Monitoring and planning

Chapter

1 Project General Context

Contents

1.1 Contexte du projet

1.2 Présentation ou description du projet,

1.3 Problématique et Objectifs du projet

1.4 Conduite du projet

1.5 Planification du projet

Chapter 2

Analysis And Design

Contents

2.1 Description générale de la plateforme

2.2 Spécification des besoins

2.3 Analyse fonctionnelle

2.4 Analyse dynamique

2.5 Analyse structurelle

Chapter

3 Technical and environmental study

Contents

3.1 Capture of technical needs

The capture of technical needs identifies all the constraints and the choices dimensioning the design of the system.

The tools and materials selected as well as the management of integration constraints generally conditioning the prerequisites of general architecture.

The capture of technical needs is as follows:

- Capturing software specifications.
- Capturing specifications related to hardware configuration.

3.1.1 Capturing software specifications

Once the hardware specifications have been determined, we move on to the non-functional specifications, in other words technical or software specifications.

These are technical features that the system will provide to the user regardless of the business or functional term.

We will therefore list a list of technical functionalities that our system will be able to provide and offer to users, which is why we have introduced the concepts of operator and technical use cases:

a) Technical actors

Also called ”technical actors” of the system, these are the actors who benefit from the technical functionalities of the system:

- User: These are the people who use the system functionality in one way or another.
- Administrator: this is the person responsible for managing all the management of the platform as well as that of the users.

b) Identification of technical use cases

Security management	Authentication	UC1-Register
		UC2-Login
		UC3-Reset Password
		UC4-Change password
		UC5-Logout
	Confidentiality	UC6-Role management
	Secure payment	UC7-Send Payment
		UC8-Receive Payment
	Backup and restore	UC9-Save the database
		UC10-Restore the database
Data managment	Dynamic data	UC11-Dynamic data dispaly
	Integrity	UC12-Simultaneous updates
	Interoperability	UC13-Network deployment
	Compatibility	UC14-Run on different browsers and platforms
	History	UC15-Get all the user created projects fundraised
		UC16-Get all the user created projects fundraised
	Statistics	UC17-Display platform statistics

	Fund	UC18-Fund migration from anonymous user
Help management	Gestion des erreurs	UC19-Logging Viewing types of errors
		UC20-Send the error by email to the administrator
	Aide en ligne	UC21-Consult the user help Guide
	Notifications	UC22-Confirmation or error message
	Mails	UC23-Sending emails
Payment management	Multi-currency platform	UC24-Choose a currency for purchase and payment
	Payment method	UC25-Possibility to choose a payment method
Optimization	Navigation	UC26-Nested operations
	Sharing	UC27-Possibility to share your donation
	Theming	UC28-Possibility to switch themes between dark and light themes

Table 1. *Identification of technical objectives and use cases*

c) Technical description use cases

View all the use cases description in the appendix 4.7

3.1.2 Capturing hardware specifications

3.2 Adopted Architecture

3.2.1 Model View Controller Design Pattern

The Model View Controller (MVC) design pattern specifies that an application consist of a data model, presentation information, and control information. The pattern requires that each of these be separated into different objects.

MVC is more of an architectural pattern, but not for complete application. MVC mostly relates to the UI / interaction layer of an application. We're still going to need business logic layer, maybe some service layer and data access layer.

1. Model contains only the pure application data, it contains no logic describing how to present the data to a user.
2. View presents the model's data to the user. The view knows how to access the model's data, but it does not know what this data means or what the user can do to manipulate it.
3. Controller exists between the view and the model. It listens to events triggered by the view (or another external source) and executes the appropriate reaction to these events. In most cases, the reaction is to call a method on the model. Since the view and the model are connected through a notification mechanism, the result of this action is then automatically reflected in the view.

3.2.2 Representational State Transfer Architecture

REST, or REpresentational State Transfer, is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other. REST-compliant systems, often called RESTful systems, are characterized by how they are stateless and separate the concerns of client and server. We will go into what these terms mean and why they are beneficial characteristics for services on the Web.

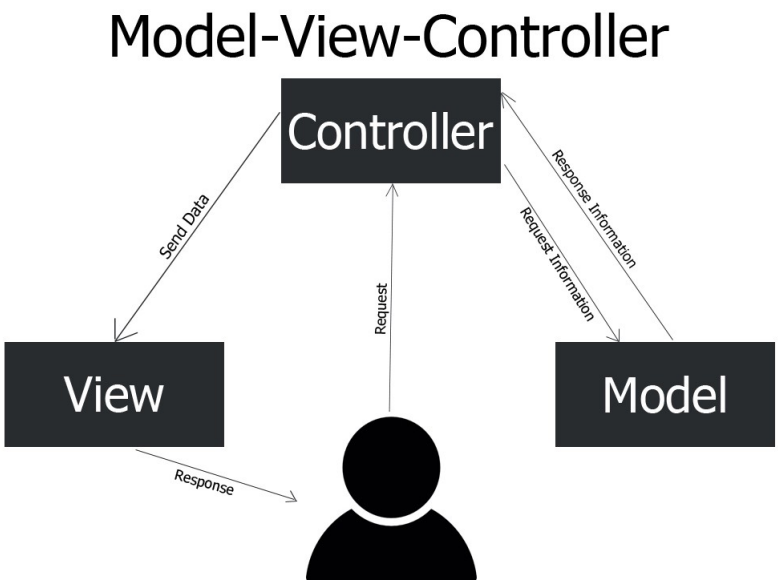


Figure 1. *Model View Controller*

3.2.3 Software as a Service

Software as a Service or SaaS , which allows us to set up services, such as application servers and databases, as needed to create the apps we need. This type of system typically works in a cloud or hybrid cloud environment, which often makes provisioning of servers and software as simple as entering some configurations and clicking a button.

This type of setup is perfect for the MEAN stack, as each piece of the MEAN stack is easily placed into the cloud and thus can be easily provisioned, so they are widely available on SaaS services and can be used quickly when needed. If we are looking for a good development stack, the MEAN stack has much to offer for IT management as well as for developers!

3.2.4 Monolithic Applications

If all the functionalities of a project exists in a single codebase, then that application is known as monolithic application. We all must have designed a monolithic application in our lives in which we were given a problem statement and were asked to design a system with various functionalities. We design our application in various layers like presentation, service and persistence and then deploy that codebase as single jar/war file. This is nothing but a

monolithic application where “mono” represents the single codebase containing all the required functionalities.

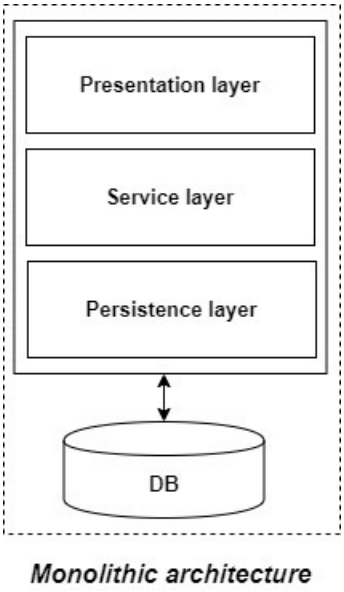


Figure 2. *Monolithic Applications*

Disadvantages of Monolithic applications:

1. It becomes too large in size with time and hence, difficult to manage.
2. We need to redeploy the whole application even for a small change.
3. As the size of the application increases, its start-up and deployment time also increases.
4. For any new developer joining the project, it is very difficult to understand the logic of large Monolithic application even if his responsibility is related to a single functionality.
5. Even if a single part of the application is facing a large load/traffic, we need to deploy the instances of the whole application in multiple servers. It is very inefficient and takes up more resources unnecessarily. Hence, horizontal scaling is not feasible in monolithic applications.
6. It is very difficult to adopt any new technology which is well suited for a particular functionality as it affects the whole application, both in terms of time and cost.
7. It is not very reliable as a single bug in any module can bring down the whole monolithic application.

Advantages of monolithic applications:

1. Simple to develop relative to microservices where skilled developers are required in order to identify and develop the services.
2. Easier to deploy as only a single jar/war file is deployed.
3. Relatively easier and simple to develop in comparison to microservices architecture.
4. The problems of network latency and security are relatively less in comparison to microservices architecture.

3.3 Choice of languages

3.3.1 Conception

a) Unified Modeling Language

b) PlantUml

PlantUML is an open-source tool allowing users to create UML diagrams from a plain text language. The language of PlantUML is an example of a Domain-specific language. It uses Graphviz software to lay out its diagrams. It has been used to allow blind students to work with UML. PlantUML also helps blind software engineers to design and read UML diagrams.

3.3.2 General

a) JSON

b) HTTP

c) JWT

3.3.3 Front-End

a) Typescript

TypeScript is an open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript and adds optional static typing to the language. TypeScript is designed for development of large applications and transcompiles to JavaScript.[5] As TypeScript is a superset of JavaScript, existing JavaScript programs are also valid TypeScript programs.

TypeScript may be used to develop JavaScript applications for both client-side and server-side execution (as with Node.js or Deno). There are multiple options available for transcompilation. Either the default TypeScript Checker can be used,[6] or the Babel compiler can be invoked to convert TypeScript to JavaScript.

TypeScript supports definition files that can contain type information of existing JavaScript libraries, much like C++ header files can describe the structure of existing object files. This enables other programs to use the values defined in the files as if they were statically typed TypeScript entities. There are third-party header files for popular libraries such as jQuery, MongoDB, and D3.js. TypeScript headers for the Node.js basic modules are also available, allowing development of Node.js programs within TypeScript.

The TypeScript compiler is itself written in TypeScript and compiled to JavaScript. It is licensed under the Apache License 2.0. TypeScript is included as a first-class programming language in Microsoft Visual Studio 2013 Update 2 and later, beside C# and other Microsoft

languages. An official extension allows Visual Studio 2012 to support TypeScript as well. Anders Hejlsberg, lead architect of C# and creator of Delphi and Turbo Pascal, has worked on the development of TypeScript.

b) HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

c) CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.[1] CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.[2]

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

3.3.4 Back-End

a) Javascript

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification.[7] JavaScript is high-level, often just-in-time compiled,

and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

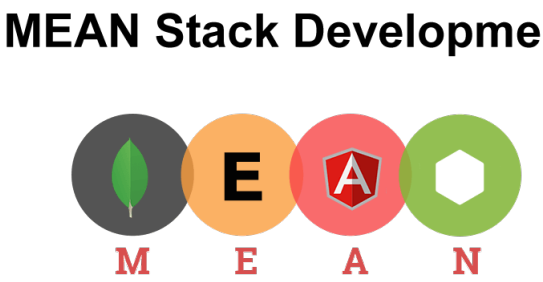
b) ES6

Script. The developers use ECMAScript mostly for client-side scripting of World Wide Web (WWW).

The sixth edition of ECMAScript standard is ECMAScript6 or ES6 and later renamed as ECMAScript 2015. It is a major enhancement to the JavaScript language, which allows us to write programs for complex applications. It adds many features intended to make large-scale software development easier. The most common ES6 web-browsers are Chrome and Firefox. A transpiler converts the ES6 based code into ES5 which is supported many browsers. TypeScript is a transpiler. Grunt, Gulp, and Babel are some other transpilers to compile the modules. Therefore, TypeScript supports ES6.

3.4 Framework used

MEAN stack



While the name sounds like “mean”, it actually stands for the software pieces that are used to create a particular development stack: MongoDB, ExpressJS, Angular, and NodeJS. One of the biggest advantages of using this particular development stack is the ability to allow developers

to use one consistent data model across the stack, using JSON and BSON (for MongoDB). This allows for quick transitions between the various pieces of the stack, especially when a single programmer has to handle more than one portion of the stack.

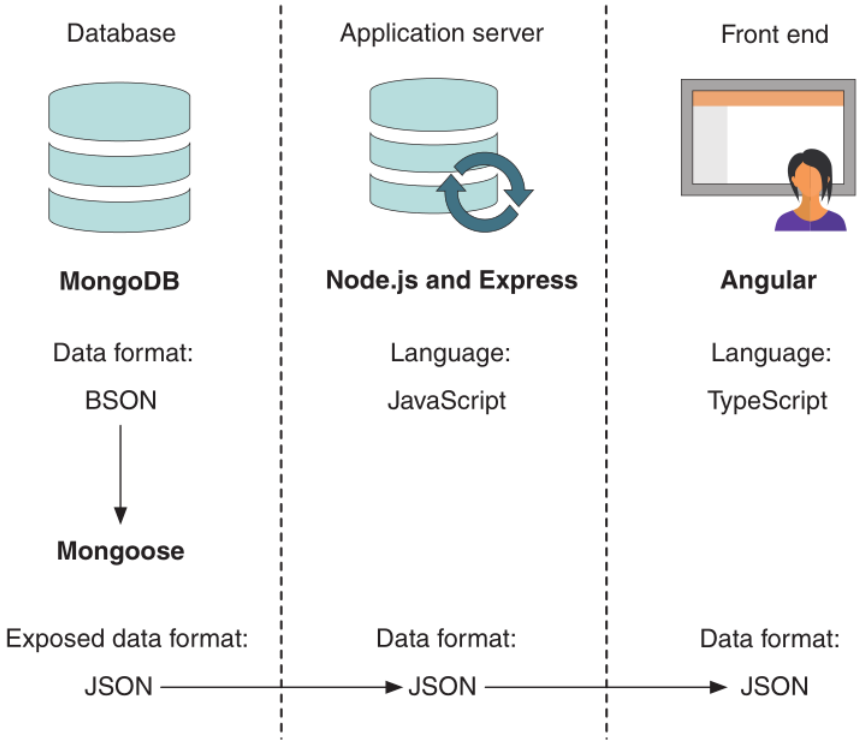


Figure 3. *Mean Stack*

3.4.1 Front-End

a) Angular 9

Angular is an app-design framework and development platform for creating efficient and sophisticated single-page apps in html, css, and Typescript which is a superset of JavaScript. Angular provides built-in features for animation, http service, and materials which in turn have features such as auto-complete, navigation, toolbar, menus, etc. The code is written in Typescript, which compiles to JavaScript and displays the same in the browser.

3.4.2 Back-End

a) Node Js

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Following are some of the important features that make Node.js the first choice of software architects.

1. Asynchronous and Event Driven: All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
2. Very Fast: Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
3. Single Threaded but Highly Scalable: Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
4. No Buffering: Node.js applications never buffer any data. These applications simply output the data in chunks.
5. License: Node.js is released under the MIT license

b) Express Js

ExpressJS is a web application framework that provides us with a simple API to build websites, web apps and back ends. With ExpressJS, we need not worry about low level protocols, processes, etc. Express provides a minimal interface to build our applications. It provides us the tools that are required to build our app. It is flexible as there are numerous modules available on npm, which can be directly plugged into Express. Express was developed by TJ Holowaychuk and is maintained by the Node.js foundation and numerous open source contributors.

c) MongoDB

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

1. Database

   Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.
2. Collection

   Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.
3. Document

   A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

d) Mongoose

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

Chapter 4

Realization, GUI And Tests

Contents

4.1 Hardware environments

4.2 Development environments

4.2.1 Conception

a) Gantt chart

A gantt chart is a horizontal bar chart that visually represents a project plan over time. Modern gantt charts typically show us the status of—as well as who's responsible for—each task in the project.

In other words, a gantt chart is a super-simple way to keep us out of a project pinch!

What are the key parts of a gantt chart? A gantt chart is made up of several different elements:

1. Task list: Runs vertically down the left of the gantt chart to describe project work and may be organized into groups and subgroups
2. Timeline: Runs horizontally across the top of the gantt chart and shows months, weeks, days, and years
3. Dateline: A vertical line that highlights the current date on the gantt chart
4. Bars: Horizontal markers on the right side of the gantt chart that represent tasks and show progress, duration, and start and end dates
5. Milestones: Yellow diamonds that call out major events, dates, decisions, and deliverables
6. Dependencies: Light gray lines that connect tasks that need to happen in a certain order
7. Progress: Shows how far along work is and may be indicated by % Complete and/or bar shading
8. Resource assigned: Indicates the person or team responsible for completing a task

4.2.2 Design

a) Adobe Photoshop

Adobe Photoshop is a software application for image editing and photo retouching for use on Windows or MacOS computers. Photoshop offers users the ability to create, enhance, or otherwise edit images, artwork, and illustrations. Changing backgrounds, simulating a real-life painting, or creating an alternative view of the universe are all possible with Adobe Photoshop. It is the most widely used software tool for photo editing, image manipulation, and retouching for numerous image and video file formats. The tools within Photoshop make it possible to edit both individual images as well as large batches of photos. We used it to prototype and create our platform logo.

b) Adobe XD

Adobe XD is a vector-based user experience design tool for web apps and mobile apps, developed and published by Adobe Inc. It is available for macOS and Windows, although there are versions for iOS and Android to help preview the result of work directly on mobile devices. XD.

4.2.3 Development

a) Visual Studio Code

DescriptionVisual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring.

b) MongoDB Compass Community

MongoDB Compass is the defacto GUI tool for MongoDB much like MySQL Workbench is MySQL's associated tool. It allows us to visually explore our data, run ad hoc queries, interact with our data with full CRUD functionality, as well as view and optimize our queries' performance.

c) Postman

Postman is an interactive and automatic tool for verifying the APIs of our project. Postman is a Google Chrome app for interacting with HTTP APIs. It presents us with a friendly GUI for constructing requests and reading responses. It works on the backend, and makes sure that each API is working as intended.

In Postman, we create a request, and Postman looks at the response to make sure it has the element we want in it. As it is an automation tool, it drastically improves testing time and quality of the project. It helps in the early detection of bugs that might sprout at later stages and cause more damage to the system.

Postman is the way to streamline the process of API testing. All APIs that we create and deploy first rigorously go through Postman so that any major or show stopper bugs are identified on time and fewer bugs leak through to later stages.

d) Git

Git is a distributed revision control and source code management system that allows several people to work on the same codebase at the same time on different computers and networks. These can be pushed together, with all changes stored and recorded. It's also possible to roll back to an earlier state if necessary.

e) Github

At a high level, GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. To understand exactly what GitHub is, we need to know two connected principles:

1. Version control
2. Git

f) Github Desktop

GitHub Desktop is a fast and easy way to contribute to projects from Windows and OS X, whether we are a seasoned users or new users, GitHub Desktop is designed to simplify all processes and workflow in our GitHub. GitHub Desktop is an open-source Electron-based GitHub app. It is written in TypeScript and uses React.

g) Boost Note

h) Latex

LaTeX is a tool used to create professional-looking documents. It is based on the WYSIWYM (what we see is what we mean) idea, meaning we only have focus on the contents of our document and the computer will take care of the formatting. Instead of spacing out text on a page to control formatting, as with Microsoft Word or LibreOffice Writer, users can enter plain text and let LATEX take care of the rest.

i) MiKTex

j) Tex Live

k) Markdown

Markdown is a lightweight markup language with plain-text-formatting syntax. Its design allows it to be converted to many output formats, but the original tool by the same name only supports HTML. Markdown is often used to format readme files, for writing messages in online discussion forums, and to create rich text using a plain text editor.

l) Marp

Marp is the ecosystem to write our presentation with plain Markdown

4.3 DBMS Choice

4.3.1 MongoDB

4.3.2 Justification of the choice

4.4 Platform security

4.4.1 Physical level

4.4.2 Logical level

4.5 Interfaces

4.6 tests

4.7 Deployment

Conclusion

Contents

And a very interesting conclusion here.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco

laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Appendix

Use cases Description

Use Case #1	Register		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 2. *Use Case Register*

Use Case #2	Login		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 3. *Use Case Login*

Use Case #3	Reset Password		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 4. *Use Case Reset Password*

Use Case #4	Change password		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 5. *Use Case Change password*

Use Case #5	Logout		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 6. *Use Case Logout*

Use Case #6	Role management		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 7. *Use Case Role management*

Use Case #7	Send Payment		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 8. *Use Case Send Payment*

Use Case #8	Withdraw Payment		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 9. *Use Case withdraw Payment*

Use Case #9	Save the database		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 10. *Use Case Save the database*

Use Case #10	Restore the database		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 11. *Use Case Restore the database*

Use Case #11	Dynamic data dispaly		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 12. *Use Case Dynamic data dispaly*

Use Case #12	Simultaneous updates		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 13. *Use Case Simultaneous updates*

Use Case #13	Network deployment		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 14. *Use Case Network deployment*

Use Case #14	Run on different browsers and platforms		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 15. *Use Case Run on different browsers and platforms*

Use Case #15	CRUD Fundraisers		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 16. *Use Case CRUD Fundraisers*

Use Case #16	Read the user's funded fundraisers		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 17. *Use Case Read the user's funded fundraisers*

Use Case #17	Display platform statistics		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 18. *Use Case Display platform statistics*

Use Case #18	Fund migration from anonymous user		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 19. *Use Case Fund migration from anonymous user*

Use Case #19	Logging and Viewing types of errors		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 20. *Use Case Logging and viewing types of errors*

Use Case #20	Send the error by email to the administrator		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 21. *Use Case Send the error by email to the administrator*

Use Case #21	Consult the user help Guide		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 22. *Use Case Consult the user help Guide*

Use Case #22	Confirmation or error message		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 23. *Use Case Confirmation or error message*

Use Case #23	Sending emails		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 24. *Use Case Sending emails*

Use Case #24	Choose a currency for purchase and payment		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 25. *Use Case Choose a currency for purchase and payment*

Use Case #25	Possibility to choose a payment method		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 26. *Use Case Possibility to choose a payment method*

Use Case #26	Nested operations		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 27. *Use Case Nested operations*

Use Case #27	Share to social media		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 28. *Use Case Share to social media*

Use Case #28	Switch UI themes		
Goal in Description	This is a very long line. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		
Stereotype and Package			
Preconditions			
Postconditions			
Primary Actors			
Use Case Relationships:			
Basic Flow			
Alternative Flow			
Exceptions			
Constraints			
User Interface Specifications			
	Metrics	Priority	Status
Notes			

Table 29. *Use Case Switch UI themes*

Architecture

Webography

[1] *Latex @ Wikipedia*. URL: https://en.wikipedia.org/wiki/LaTeX (visited on 0004–2016).

[2] *FSA*. URL: http://www.fsa.ac.ma/ (visited on 0003–2020).

Bibliography

[3] Charles Bazerman et al. *Shaping written knowledge: The genre and activity of the experimental article in science*. Vol. 356. University of Wisconsin Press Madison, 1988.

[4] Ashraf Aboulnaga, Alaa R Alameldeen, and Jeffrey F Naughton. "Estimating the selectivity of XML path expressions for internet scale applications". In: *VLDB*. Vol. 1. 2001, pp. 591–600.

[5] Ashbindu Singh. "Review article digital change detection techniques using remotely-sensed data". In: *International journal of remote sensing* 10.6 (1989), pp. 989–1003.