





**Faculty of computers and artificial intelligence  
Information Systems Department.**

**Supervised by:  
DR. Ahmed Yakoup**

**Team Members**



**Rania  
Ramadan**



**Ashrakt  
Mohamed**



**Israa  
Sami**



**Eman  
Mohamed**



**Nehal  
Salama**



**Kholoud  
Mostafa**

## Abstract

Our project is primarily about Start-ups, it's a website tries to connect entrepreneurs with the market through the way of posting, commenting and interacting with each other.

In reality there are a lot of youth want to start their own businesses but unfortunately after getting the idea they suddenly stop their efforts because they don't have the idea about expressing their thinking and delivering their business to people and market and how to bring someone invests in it.

But let us think about entrepreneurs who try to get over these obstacles and start to search, absolutely they should allocate time to this, what about time consuming? What about the effort? what about if there weren't information? Our system comes to solve these issues and let entrepreneurs, investors and even job seekers to be connected to each other to get what each type wants, and make it an easy way to encourage them all.

We have different types of users such as investors, entrepreneurs, job seekers, and business partner, so we try to connect them all to each other.

## Dedication

This project is wholeheartedly dedicated to our beloved parents, who have been our source of inspiration and gave us strength when we thought of giving up, who continually provide their moral, spiritual, emotional, and financial support.

To our brothers, sisters, mentor, friends, and classmates who shared their words of advice and encouragements to finish this project.

# Acknowledgement

First and foremost, we would express our deepest gratitude and appreciation to our advisor **Dr.Ahmed Yakoup** for his support, outstanding guidance and encouragement throughout our graduation project and his continuous support and help as he was always there for us providing more than the needed from him since the very beginning of our project.

We would also like to thank our families, especially our parents, for their encouragement, patience, and assistance over the years.

We are forever indebted to our parents, who have always kept us in their prayers and pushed us to Success.

Finally, for our faculty for providing the suitable environment that leaded us to represent the best image that computer science graduates of Helwan University are supposed to represent.

**THANK YOU ALL**

# Table of Contents

<b>Chapter 1: Introduction</b>	<b>12</b>
1.1 Problem Statement	13
1.2 Objectives	13
1.3 Scope Of Work	14
1.3.1 Planning	14
1.3.2 Designing	14
1.3.3 Coding	14
1.3.4 Testing	15
1.3.5 Documentation	16
1.4 Overview Of Remainder Of Document	16
<b>Chapter 2 : Background/Literature Review</b>	<b>18</b>
2.1 General Constraints	19
2.1.1 Time	19
2.1.2 Assign Tasks	19
2.1.3 Gathering Data	20
2.1.4 Manager Behavior Work	20
2.2 Project Description	20

2.2.1 Goal And Definition	20
2.2.2 Problems To Be Faced	21
2.2.3 People And Organizations	21
2.3 Assumptions And Constrains	21
2.3.1 Project Assumptions	21
2.3.2 Project Constrains	22
2.3.2.1 Business Constraints	22
2.3.2.2 Technical Constrains	22
2.4 Context And Background	23
2.4.1 Context	23
2.4.1.1 Political	23
2.4.1.2 Sociological	23
2.4.1.3 Legal	24
2.4.1.4 Environment Factors	24
2.4.2 Background	24
2.4.2.1 Expected Value	24
2.4.2.2 Pre-requisite Knowledge	25
2.5 Project Benefits	25

<b>Chapter 3: METHODOLOGY</b>	<b>27</b>
3.1 Development Approach And Methodology	28
3.1.1 Approach	28
3.1.2 Methodology	30
3.2 Hardware And Software Tools	38
3.2.1 Software Tools	38
3.2.2 Hardware Tools	41
3.3 Planning And Analysis	42
3.3.1 Project Planning	42
3.3.1.1 Feasibility Study	42
3.3.1.2 Estimated Cost	46
3.3.1.3 Gantt Chart	46
3.3.2 Analysis And Limitation Of Existing System	48
3.3.2.1 Assumptions And Constraints	48
3.3.2.2 Risk And Risk Management	54
3.4 Requirements	58
3.4.1 User Requirements	58
3.4.1.1 Mandatory Requirements	58

3.4.1.2 Desirable Requirements	58
3.4.2 System Requirements	58
3.4.3 Domain Requirements	59
3.4.4 Functional Requirements	59
3.4.5 Non-Functional Requirements	71
3.5 Software Design	75
3.5.1 Class Diagram	75
3.5.2 Use Cases Diagram And Scenarios	76
3.5.3 ERD Diagram	78
3.5.4 Sequence Diagram	78
3.5.5 Activity Diagram	91
3.5.6 System Architecture	92
<b>Chapter 4: Result and Discussion</b>	<b>94</b>
4.1 Results	95
4.1.1 Expected Results	95
4.1.2 Actual Results	96
4.2 Discussion and Testing	97
<b>Chapter 5: Conclusion And Discussions</b>	<b>107</b>

5.1 Conclusion _____	108
5.2 Recommendations For Future Work _____	109

## List of Figures

Incremental Approach_____	29
Agile Approach _____	30
Gantt Chart _____	47

## List of Tables

Dividing time for the whole project_____	19
Competitors _____	48
Schedule Risk _____	54
Budget Risk _____	55

Operational Risk	56
Technical Risk	57
Registration and log in test cases	98
Profile test cases	102
Post test cases	103
Admin and category test cases	104
Comment test cases	105

## 1.introduction

In this chapter, we will discuss the project's objectives, scope of the project, work needed to build the project, and overview of the whole document.

## 1.1 Problem Statement

This website is to solve two common problems, the first is the problem of wasting time of searching for investors, marketing resources and technologies, and the second is the rare of this information because it isn't a widespread businesses, we will solve these problems by creating a website to make a miniature market.

## 1.2 Project Objectives

In general, the project objective is to make it easy for the entrepreneurs to find what helps them to market and grow up their businesses.

The main objectives in the following:

- Connecting entrepreneurs to the market.
- Contributing in decreasing time consumption.
- Contributing in decreasing unemployment.
- Providing support to entrepreneurs by any needs the need.
- The website shows the available posts for all the projects on it.
- The entrepreneur (project owner) shows his details about the project through uploading images, videos.
- The investor or donator search for the posts to view the projects, he can search by location or the latest updates or the under constructed projects.
- The website connects the co-founders to the investors with no need to any broker, the website will be the only broker.

- The job seeker goes to search for the start-ups which offers an opportunity for work.
- The website tracks the user to represent the posts that might have interest.

### 1.3 Scope Of work

The approximate work involved to finish the project is divided into These five phases:

#### 1) 1.3.1 Planning

- Collecting data about the project and the lack that made us in a need of the website.
- Making surveys in different ways (video, questionnaires, forms ...etc.).
- Determining the functional and non-functional requirements.
- Setting a Gantt chart for the project.
- Determining the resources of the team.
- Dividing the tasks on team using Trello.

#### 2) 1.3.2 Designing

- Determining the diagrams to be carried out within the project: -
  - ERD Diagram.
  - Activity Diagram.
  - Class Diagram.
  - Use-case Diagram.
  - Sequence Diagram.
  - SW Architecture.
  - Wireframe diagram.

#### 3) 1.3.3 Coding

- The main supposed functions to be coded in this App are:

- Sign Up.
- Login.
- Send email.
- Accept & Reject post.
- Chat.
- Add post.
- Update post.
- Delete post.
- Add Comment.
- Update Comment.
- Delete Comment.
- Rate.
- Feedback.
- Send a message.
- Search.
- Log out.
- Contact.
- Filter.
- Add job.
- Track user.
- Semantic search.
- Add category.
- Delete category.
- Edit category.
- Make admin.
- Block user.
- Edit profile.
- Edit/Delete profile picture.

#### 4) 1.3.4 Testing

- Functional Testing
  - Unit Testing.
  - Regression Testing.
  - Integration Testing.
  - System Testing.
  - Acceptance Testing.
- Non-functional Testing.
  - Performance Testing.
  - Stress Testing.
  - Load Testing.
  - Security testing.
- Test Cases

### 5) 1.3.5 Documentation

The documentation should mainly include these main chapters:

- Project Introduction : includes an overview of the project and limitations.
- Project Background/Literature Review : includes the tools and technologies as well as tasks and timeline plan.
- Project Methodology: includes the functional and nonfunctional requirements, the diagrams, software tools ,analysis of the project and the development approach.
- Project Testing : includes testing types.

## 1.4 Overview Of Remainder Of Document

The remainder of the document will take us through the details of the project and will cover the following:

- The general constraints and limitation of the project.
- Project benefits.
- Project's development approach.
- Software tools.
- Planning and analysis of the idea of this project.
- Risk management.
- Project Requirements.
- Project' Software diagrams
- Project results.

## 2. Background/Literature Review

In this chapter includes citation of theoretical background, related work and general constraints should be included. We're going to discuss and go deeper in how we plan the project and the project benefits.

## 2.1 General Constraints

### ➤ 2.1.1 Time

- Limitations in the time of delivering a specific sprint of project in each meeting of our advisor.
- Limitations in the estimated time to deliver the whole project
- Limitations of finding the resource in an appropriate time
- Learning new technology take long time.
- Training a model took a long time to work, consuming the resources of the device and its large area.

So we have put this schedule before beginning in developing project

Tasks	Time
<b>Time for finding ideas and study them for choosing</b>	1 month
<b>Time for making software diagram</b>	2 months
<b>Time for learning needed technology</b>	2 months
<b>Time for each sprint</b>	2 weeks
<b>Time for development process</b>	7 months
<b>Editing and testing</b>	1 month

Table 1-Dividing time for the whole project

### ➤ 2.1.2 Assign Tasks

We have considered these constraints in dividing tasks:

- The physical ability of each team member.
- The knowledge and background of team members.
- The ability of each team member to learn new technology.
- The skills of each team member in searching for information.
- Limitation in putting specific format for variables and functions to tackle errors.

We use trello website to manage our work. It divides the project into Multi sprints, each sprint contains 3 categories of tasks (to do,

doing and done tasks). Every task has a deadline and every member has mentioned a specific task.

#### ➤ **2.1.3 Gathering Data**

- The resources of data aren't available.
- The resources of data require to pay for.
- The different formats for APIs make problems in integrating them.
- Limitation in finding tutorials which explain what we exactly need for new technologies particularly.

#### ➤ **2.1.4 Manager Behavior training**

- Limitations in understanding and getting the information and points quickly.
- Limitations in selecting a meeting time appropriate to advisor.

## **2.2 Project Description**

#### ➤ **2.2.1 Goal And Definition**

- This website was built upon solving young entrepreneurs problems that tackle them from begging their businesses.
- So the website contains different users such as investors, entrepreneurs, job seekers, and business partner.
- The website tries to enable the multiple users to connect each other through some features like uploading profile information, uploading posts of what they obviously need, commenting and sending feedback on the content they like or not, chatting with the desired destination they want, applying for job and sending emails if they want.
- Also we found that unemployment had been increased so we have added a feature for entrepreneurs if they want employees

they will post that and the job seekers can search in the category they want to find jobs.

- Then it is a collection of features solve some problems.

#### ➤ **2.2.2 Problems to be faced**

- We may face a problem in time we have to deliver the project in.
- Also the budget we need to make every feature efficiently.
- Anything would happen to any team member so the work load will be on the rest of the team.
- Problems in equipment/s or software editors.

#### ➤ **2.2.3 People and Organizations**

- Investors.
- Entrepreneurs.
- Job seekers.
- Donators.
- Incubators.
- Funding organizations.
- Marketing organizations.

### **2.3 Assumptions AND Constraints**

#### ➤ **2.3.1 Assumptions for project**

- We will get all the courses we need to learn technology.
- We will have the required time to finish this project which is 11 months.
- We will find the users for this project and it will be active.
- We will get the paid API we want to make search efficient.

- We will have contacts of professional people for any enquiries.
- It is applicable and can be developed.
- Your team members have all the required skills.
- All of the equipment/s and software editors is in good condition.
- The project will be delivered in an efficient time.

### ➤ **2.3.2 Constraints for project**

#### ➤ **2.3.2.1 Business constraints**

- The project should have specific categories.
- The project have only one admin.
- No money available to pay for paid resource.
- We have only one tester.
- We will work on agile environment.
- The physical ability for each team member.

#### ➤ **2.3.2.2 Technical constraints**

- We have a lot of errors.
- We will use github for uploading the source code so all should learn it.
- We have only one knows the technology and the other will learn.
- Everything might be approved firstly.
- Account should be secure for not being hacked.
- The ability of solving errors.

## 2.4 Context AND Background

### 2.4.1 Context

There are some issues would affect any project but in our project we don't have almost of them

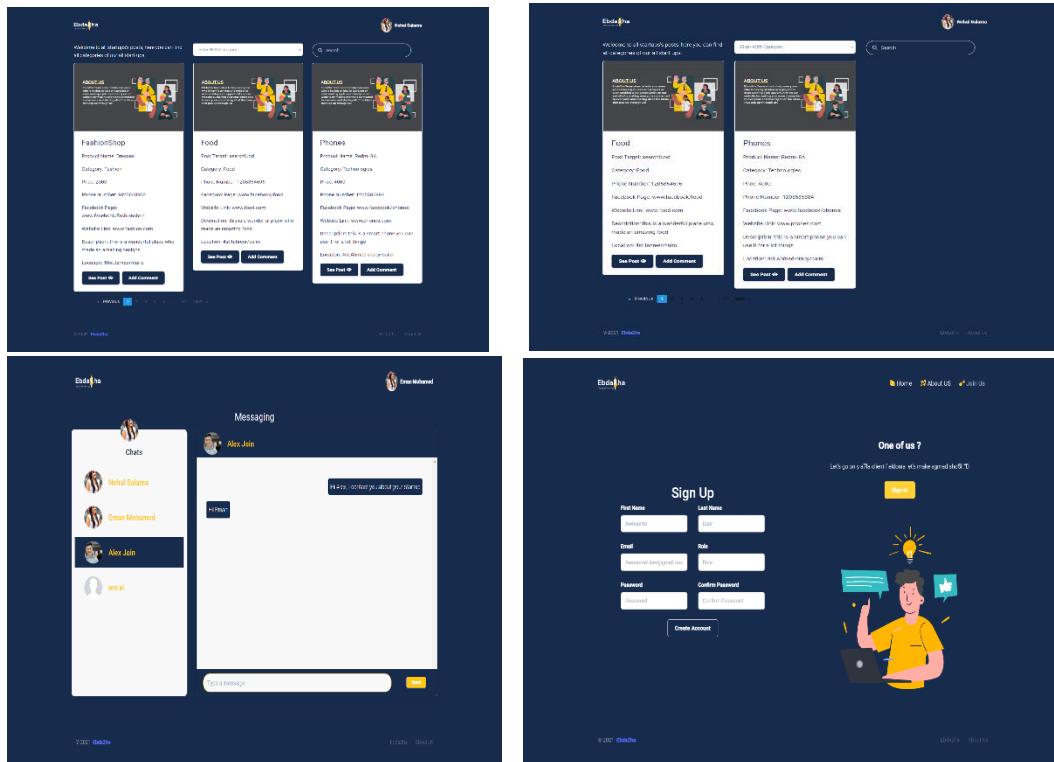
#### 1) 2.4.1.1 Political

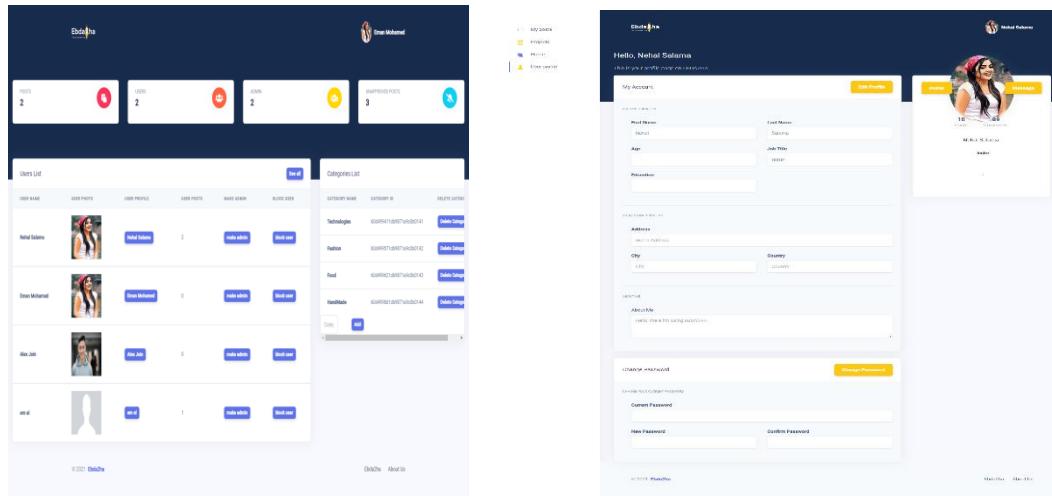
Our project isn't fast changing, it is only can be updated and adding some new features, so our project will be easily accepted nationally and locally.

Also it isn't for some type of the country, it is open to any one inside Egypt and have an idea for start-up.

#### 2) 2.4.1.2 Sociological

Our project designed to be easy to use for the sector of entrepreneurs, investors and job seekers, and designed to keep up the thinking of these types.





### 3) 2.4.1.3 Legal

In our project we considered that the time of work is appropriate to the work assigned.

We set the sprint takes 2 weeks, and the sprint have three simple tasks.

### 4) 2.4.1.4 Environment Factors

It doesn't have any impact on physical environment, so we don't have risks of any damages.

On the other hand, it has an impact on the social environment because it is a source of improving start-ups and finding jobs for the related categories of start-ups.

## 2.4.2 Background

### ➤ 2.4.2.1 Expected value

We expected a website for connecting entrepreneurs with investors helping them in reaching their goals and also decreasing the unemployment problem.

It also supports different categories and containing different types of chatting and contacting.

#### ➤ 2.4.2.2 Pre-requisite knowledge

- Team members were divided to these categories according to background of them
  - ✓ Some knows angular framework and node js as the backend.
  - ✓ Some knows PHP language.
  - ✓ Some knows ASP.net language.
  - ✓ All team members have a good background about structured query language (SQL) and software diagrams.
  - ✓ All knows how to work in agile environment.
  - ✓ All worked with trello tool.
  - ✓ Some knows how to deal with github

## 2.5 Project Benefits

The project brings some benefits such as:

- Free platform of connecting to the sector of services.
- A huge collection of ideas and variety of categories.
- Encouraging to youth for thinking in start-ups and start to plan how to implement them.
- Decreasing unemployment problem.
- Decreasing consuming cost.

And we have get some soft skills such as:

- 1) Team Work.
- 2) Leadership Skill.
- 3) Communication Skills.

4) Negotiation Skills.

We also have got also technical skills and experience such as :

- 1) Learning node JS as backend.
- 2) Learning angular and react as framework.
- 3) Learning testing.
- 4) Learning APIs.

### 3. Methodology

In this chapter, we will go deeper in Ebda2ha web application's implementation approach and methodology, analysis and risk management, requirements, design, and present its diagrams and database.

## 3.1 Development Approach and Methodology

### ➤ 3.1.1 Approach

We have studied the approaches of developing a software, and also studied the idea of the project.

And we have decided to set the incremental approach for our project for some reasons will be discussed.

#### Incremental Approach

Determines user needs and defines the overall architecture, but then delivers the system in a series of increments ("software builds").

The first build incorporates a part of the total planned capabilities, the next build adds more capabilities, and so on, until the entire system is complete.

- ✓ It is used to improve the technical manageability of projects and avoid the risk of a "big bang" integration at the end of a project.
- ✓ The system is built in a series of steps or *increments*.
- ✓ An architecture for the entire system is developed at the beginning of the project, it ensures that all increments can be realized and integrated into the entire system.
- ✓ Each increment creates an extension of the system, building on the previous step.
- ✓ Each single step is realized in an independent sequential process, often called a **mini waterfall**.
- ✓ The requirements are written in detail, a design is created, and new classes are implemented and tested in unit tests for each increment.

- ✓ Subsequently, the new classes are integrated into the entire system, followed by integration and system tests.
- ✓ All tests, from the unit test to the system test, run at the end of each increment.

And this figure explains how the incremental approach works:

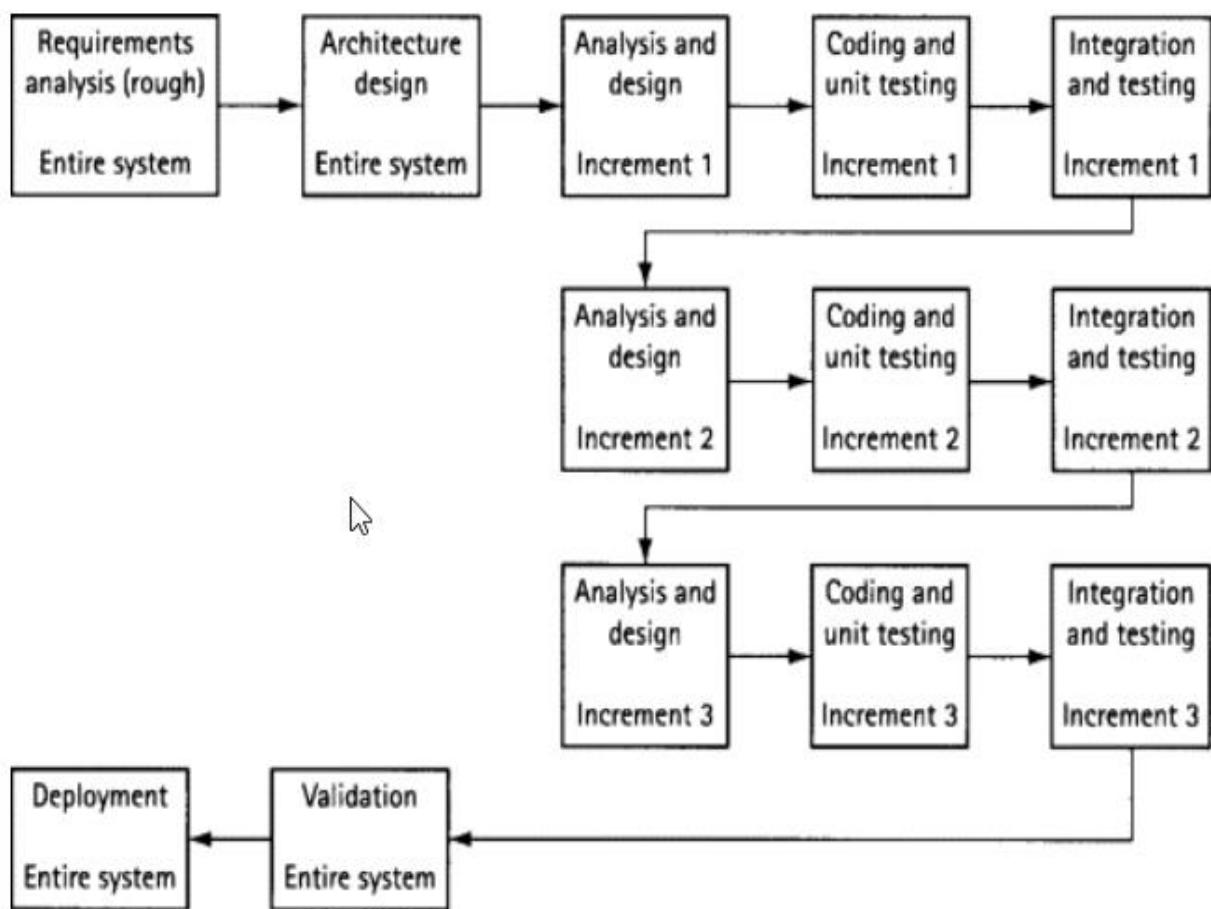


Figure 1-incremental approach

So this approach is appropriate for our idea of project, and to the teamwork it helps in decreasing load on them.

### ➤ 3.1.2 Methodology

To cope up with incremental approach we have selected a methodology would help us in developing software.

We selected agile Development methodology.

#### **Agile Development Methodology**

It is a methodology of developing a software and contains some advantages, so we selected this methodology.

##### ➤ Pros

- ✓ It minimizes risks such as bugs, cost overruns, and changing requirements.
- ✓ It allows software to be released in iterations.
- ✓ Iterative releases improve efficiency by allowing teams to find and fix defects and align expectation early on.
- ✓ It also allows users to realize software benefits earlier, with frequent incremental improvements.
- ✓ It makes clients involved in the process to get feedbacks from them.

And this figure represent the idea of agile:

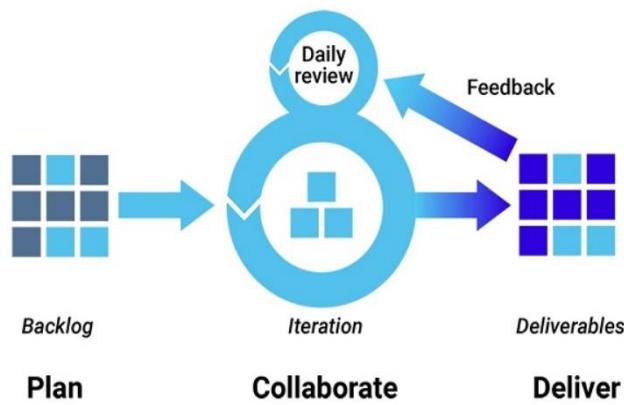


Figure 2-Agile approach

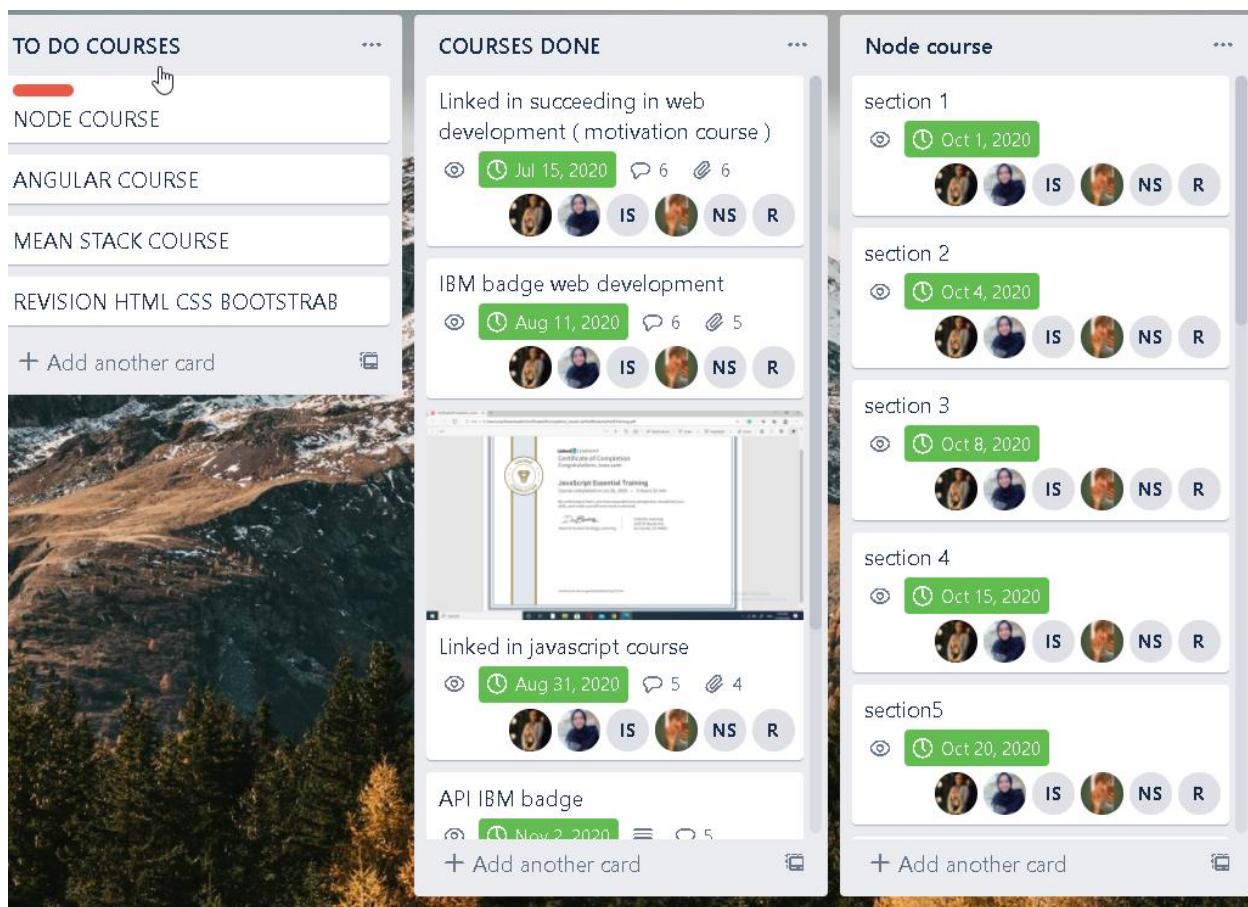
Then there are a lot of types inside agile, but we selected Scrum methodology and it consists of:

- Knowing all features and putting them in backlog.
- Then divide the project into number of sprints.
- Then dividing each sprint into a number of user stories.
- Each sprint contains three columns To Do , Doing, Done.
- Putting tasks in each Sprint and then in each column.
- Assign each tasks to only one team member or to a group of them.
- Set a deadline for each sprint.
- Then set a deadline for each task.
- Choose a repository to upload tasks on.

Then we used trello tool to divide the work and to implement the agile methodology.

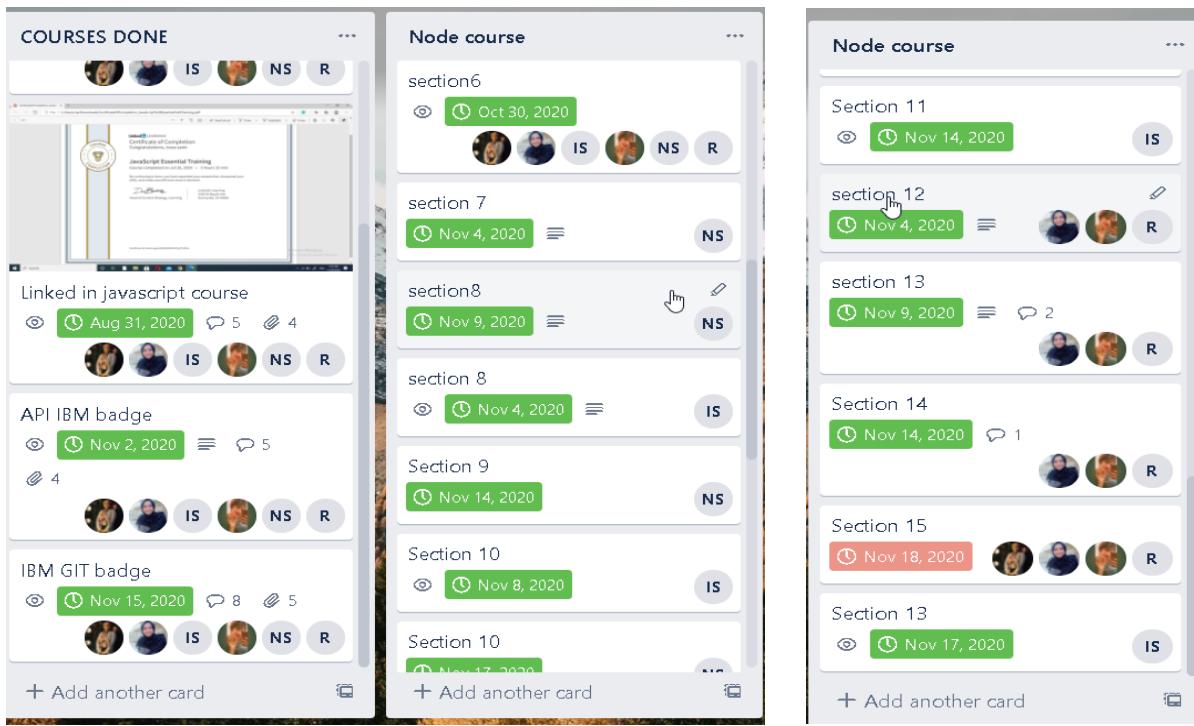
And this is our board on trello:

The learning section :



The image shows a digital workspace interface with three main sections:

- TO DO COURSES**: A list of courses to be completed, including "NODE COURSE", "ANGULAR COURSE", "MEAN STACK COURSE", and "REVISION HTML CSS BOOTSTRAB". A button "+ Add another card" is available.
- COURSES DONE**: A list of completed courses, including "Linked in succeeding in web development (motivation course)" (date: Jul 15, 2020), "IBM badge web development" (date: Aug 11, 2020), "Linked in javascript course" (date: Aug 31, 2020), and "API IBM badge" (date: Nov 2, 2020). Each entry includes a timestamp, a comment count, a share icon, and a circular profile icon for each user.
- Node course**: A detailed view of a Node.js course, showing five sections:
  - section 1 (Oct 1, 2020): Includes a timestamp, a comment count, a share icon, and circular profile icons for users IS, NS, and R.
  - section 2 (Oct 4, 2020): Includes a timestamp, a comment count, a share icon, and circular profile icons for users IS, NS, and R.
  - section 3 (Oct 8, 2020): Includes a timestamp, a comment count, a share icon, and circular profile icons for users IS, NS, and R.
  - section 4 (Oct 15, 2020): Includes a timestamp, a comment count, a share icon, and circular profile icons for users IS, NS, and R.
  - section 5 (Oct 20, 2020): Includes a timestamp, a comment count, a share icon, and circular profile icons for users IS, NS, and R.A "+ Add another card" button is also present at the bottom of this section.



The screenshots illustrate the user interface of the Ebda2ha web application, specifically focusing on course management and progress tracking.

**Screenshot 1: Courses Done**

- Linked in javascript course:** Completed on Aug 31, 2020, with 5 comments and 4 attachments. It includes a certificate image and participant icons.
- API IBM badge:** Completed on Nov 2, 2020, with 5 comments and 4 attachments.
- IBM GIT badge:** Completed on Nov 15, 2020, with 8 comments and 5 attachments.

**Screenshot 2: Node course**

This section shows a list of course sections:

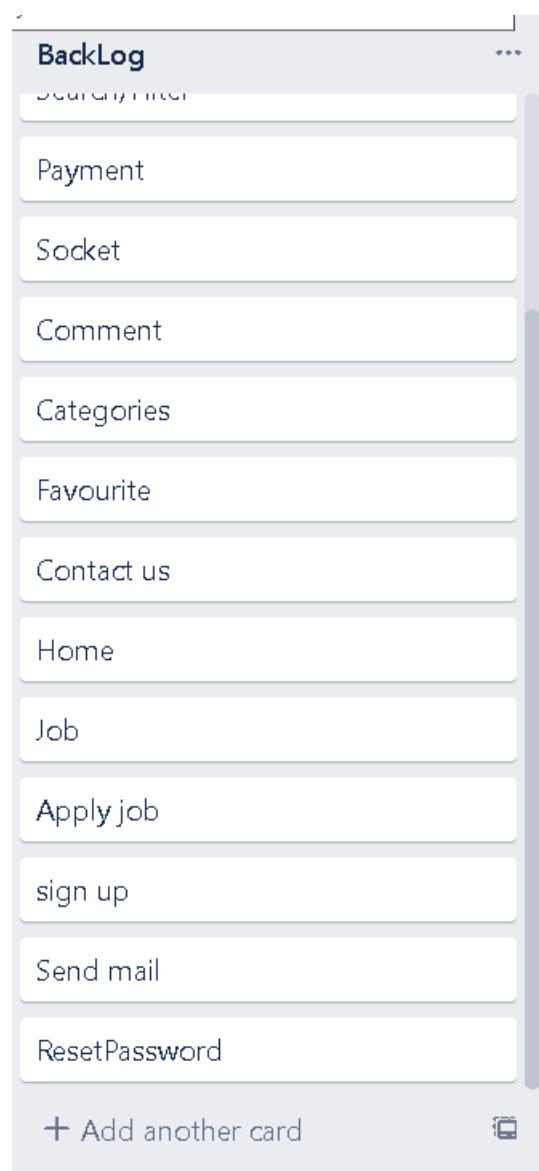
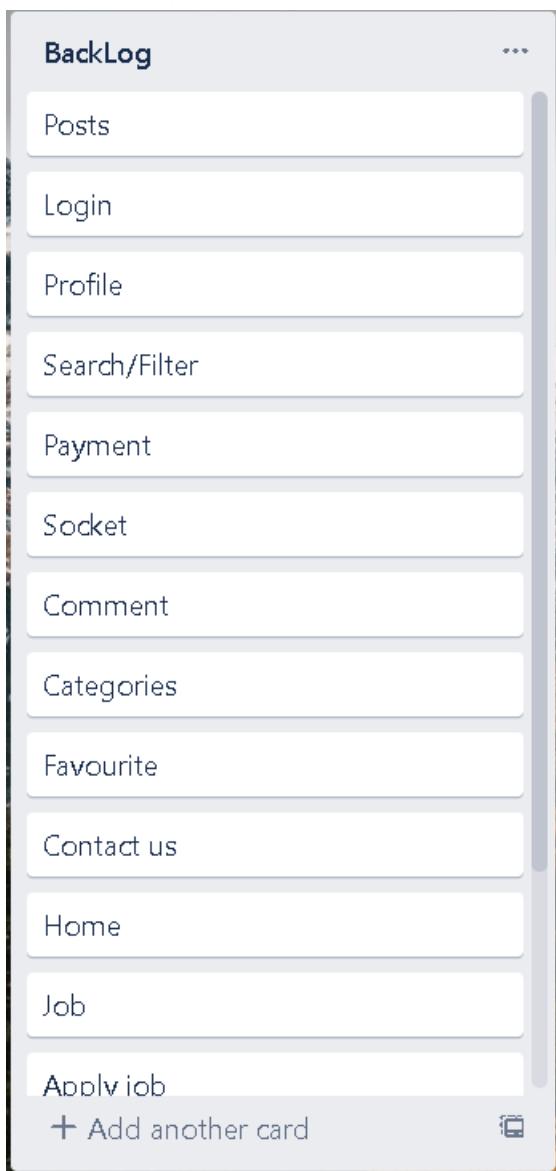
- section6: Completed on Oct 30, 2020, by participants IS, NS, and R.
- section 7: Completed on Nov 4, 2020, by participant NS.
- section8: In progress, last updated on Nov 9, 2020, by participant NS.
- Section 9: In progress, last updated on Nov 14, 2020, by participant NS.
- Section 10: In progress, last updated on Nov 8, 2020, by participant IS.
- Section 10: In progress, last updated on Nov 17, 2020, by participant IS.

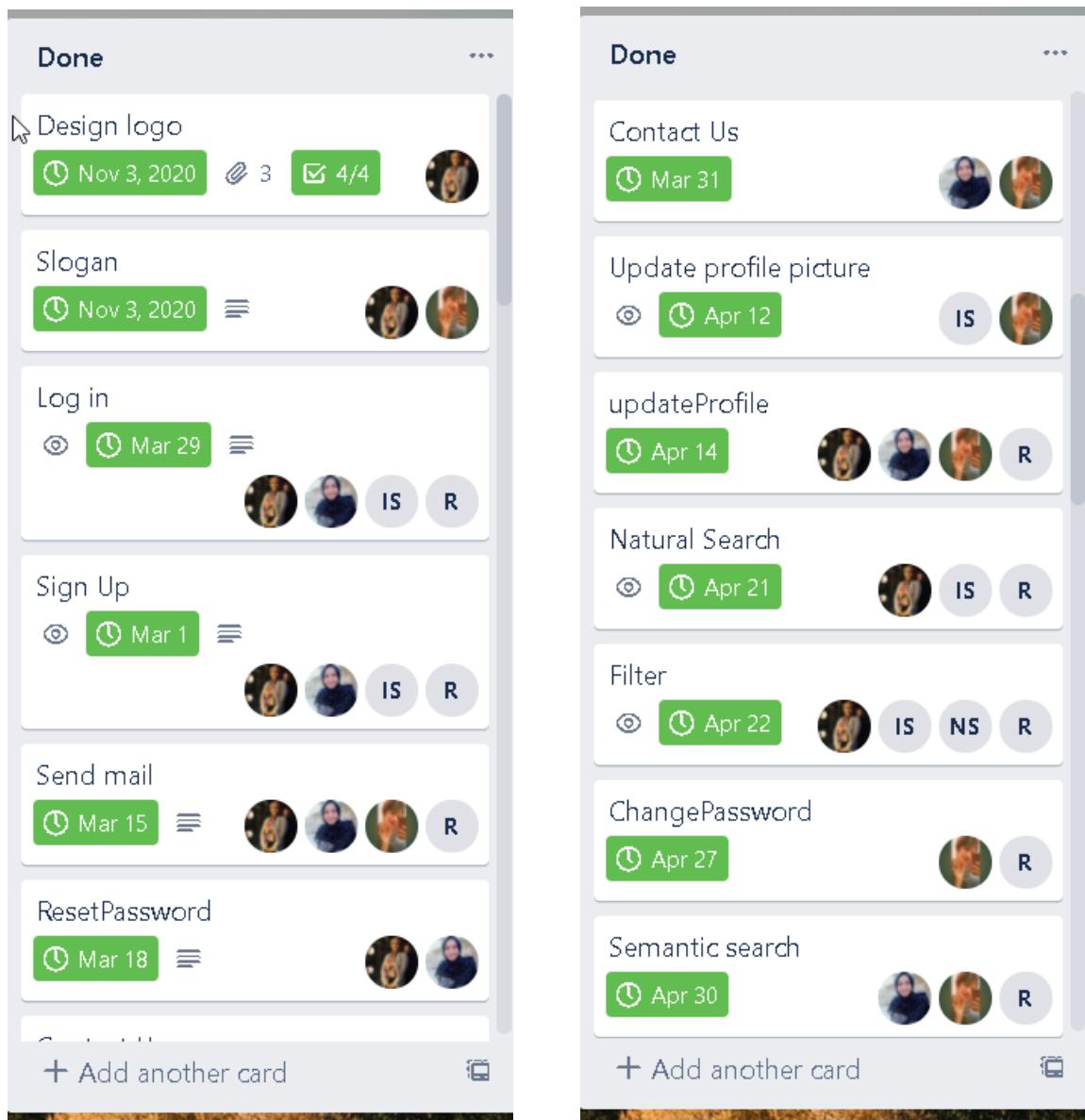
**Screenshot 3: Node course**

This section shows a list of course sections:

- Section 11: In progress, last updated on Nov 14, 2020, by participant IS.
- Section 12: In progress, last updated on Nov 4, 2020, by participants R and IS.
- Section 13: In progress, last updated on Nov 9, 2020, by participants R and IS.
- Section 14: In progress, last updated on Nov 14, 2020, by participants R and IS.
- Section 15: In progress, last updated on Nov 18, 2020, by participants R and IS.
- Section 13: In progress, last updated on Nov 17, 2020, by participant IS.

The working section:





The image displays two side-by-side screenshots of a Kanban board from a web application. Both boards have a header labeled "Done" and a "..." button in the top right corner.

**Left Board (Top Row):**

- Design logo: Status: Done (green), Date: Nov 3, 2020, Comments: 3, Attachments: 4/4, Participants: 1 (profile picture).
- Slogan: Status: In Progress (yellow), Date: Nov 3, 2020, Comments: 3, Participants: 2 (profile pictures).

**Left Board (Second Row):**

- Log in: Status: In Progress (yellow), Date: Mar 29, Comments: 1, Participants: 4 (profile pictures, including "IS" and "R").
- Sign Up: Status: In Progress (yellow), Date: Mar 1, Comments: 1, Participants: 4 (profile pictures, including "IS" and "R").

**Left Board (Third Row):**

- Send mail: Status: In Progress (yellow), Date: Mar 15, Comments: 1, Participants: 4 (profile pictures, including "R").
- ResetPassword: Status: In Progress (yellow), Date: Mar 18, Comments: 1, Participants: 2 (profile pictures).

**Left Board (Bottom Row):**

- + Add another card

**Right Board (Top Row):**

- Contact Us: Status: Done (green), Date: Mar 31, Participants: 2 (profile pictures).
- Update profile picture: Status: In Progress (yellow), Date: Apr 12, Participants: 2 (profile pictures, including "IS").

**Right Board (Second Row):**

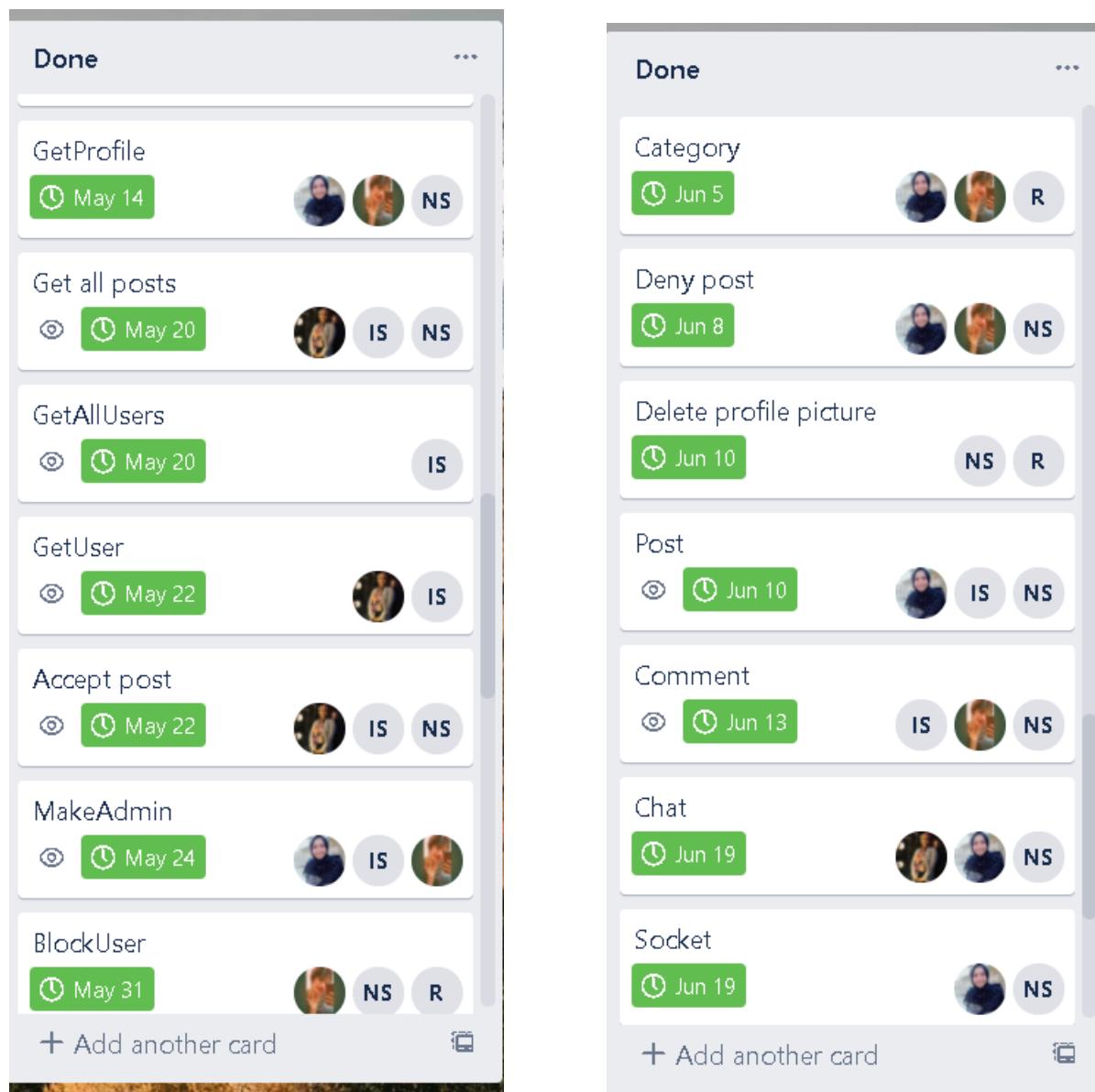
- updateProfile: Status: In Progress (yellow), Date: Apr 14, Participants: 4 (profile pictures, including "R").
- Natural Search: Status: In Progress (yellow), Date: Apr 21, Participants: 4 (profile pictures, including "IS" and "R").

**Right Board (Third Row):**

- Filter: Status: In Progress (yellow), Date: Apr 22, Participants: 4 (profile pictures, including "IS", "NS", and "R").
- ChangePassword: Status: In Progress (yellow), Date: Apr 27, Participants: 2 (profile pictures, including "R").

**Right Board (Bottom Row):**

- Semantic search: Status: In Progress (yellow), Date: Apr 30, Participants: 2 (profile pictures, including "R").
- + Add another card



The image displays two side-by-side screenshots of a Kanban board, likely from a project management application. Both boards have a header labeled "Done" and three columns: "Category", "Deny post", and "Delete profile picture".

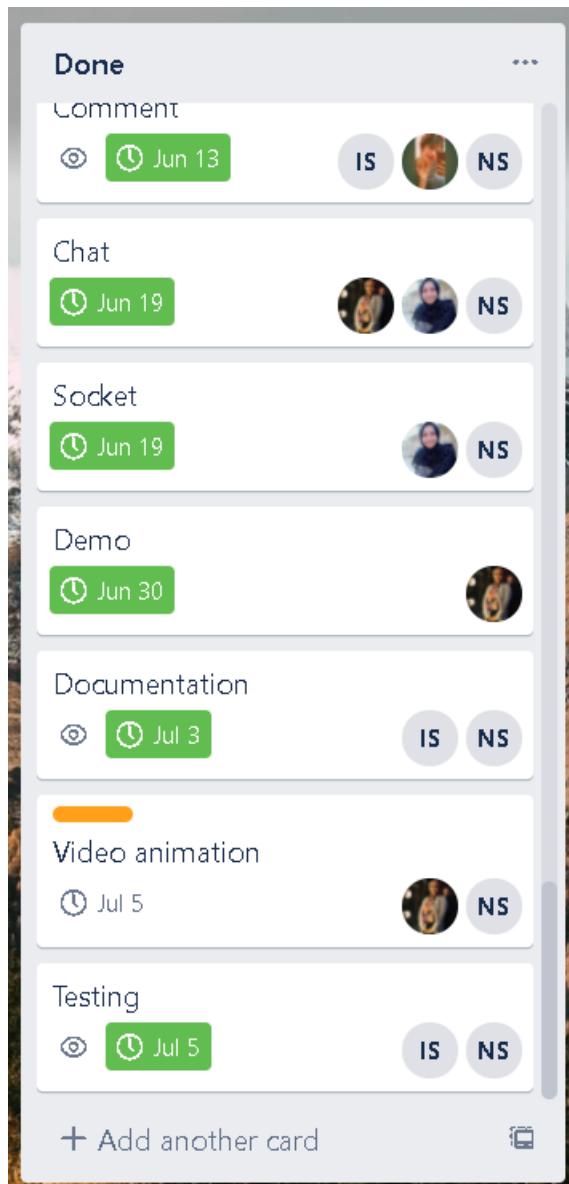
**Left Board (May 14 - June 19):**

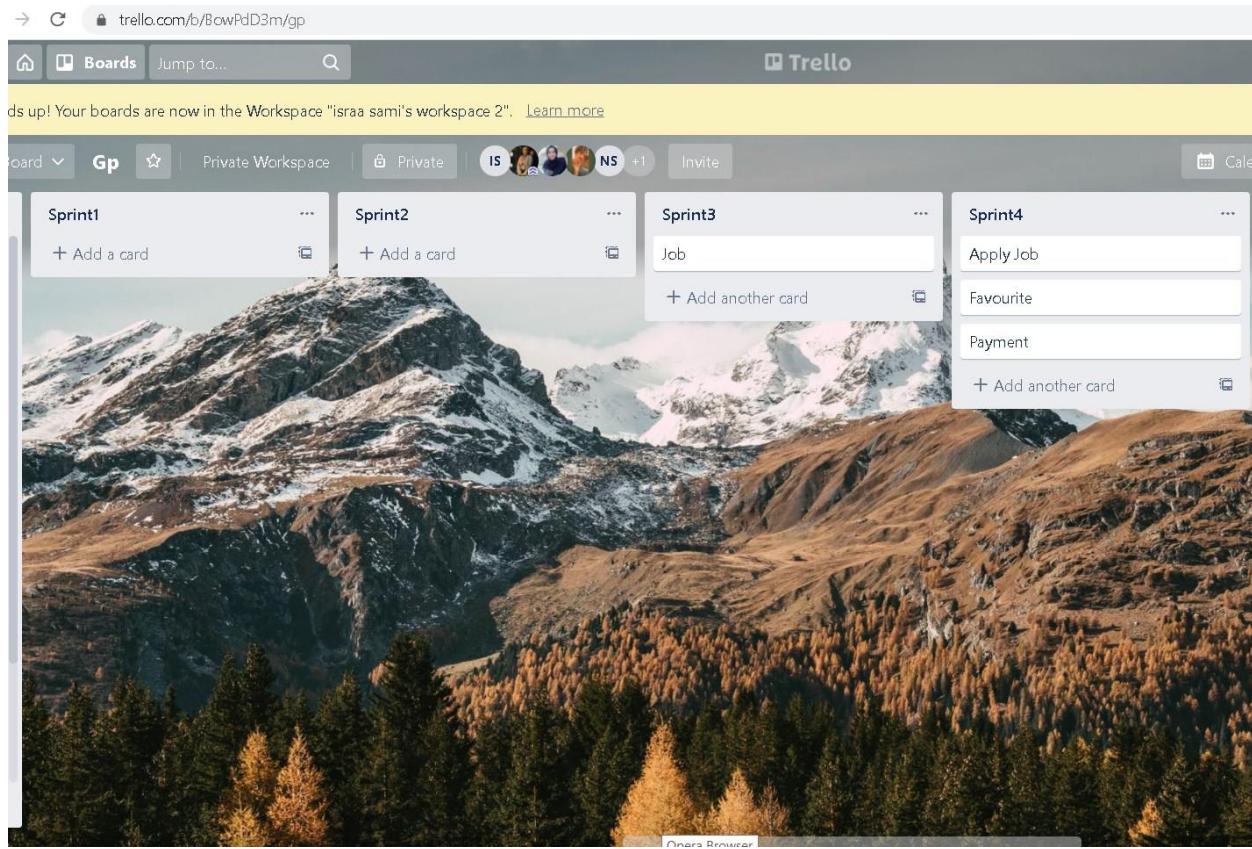
- Category: GetProfile (Status: NS)
- Deny post: Get all posts (Status: IS, NS)
- Delete profile picture: GetAllUsers (Status: IS)
- Post: GetUser (Status: IS, NS)
- Comment: Accept post (Status: IS, NS)
- Chat: MakeAdmin (Status: IS, NS)
- Socket: BlockUser (Status: NS)

**Right Board (May 14 - June 19):**

- Category: Deny post (Status: R)
- Deny post: Delete profile picture (Status: NS, R)
- Delete profile picture: Post (Status: NS, R)
- Post: Comment (Status: IS, NS)
- Comment: Chat (Status: NS)
- Chat: Socket (Status: NS)

Both boards include a footer with "+ Add another card" and a trash bin icon.

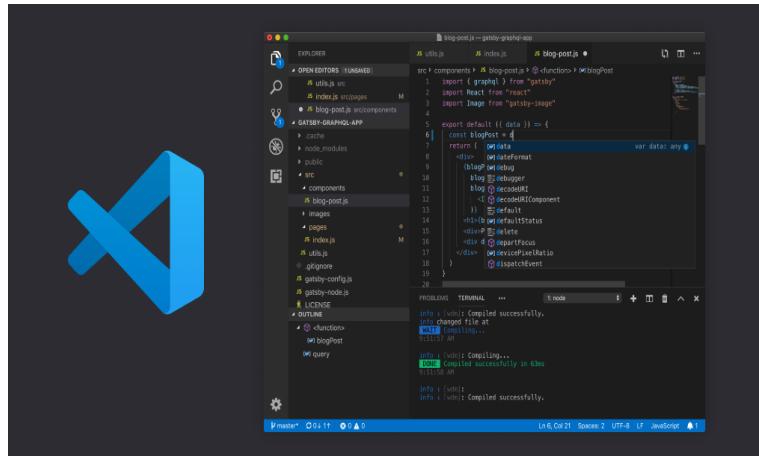




## 3.2 Hardware And Software Tools

### ➤ 3.2.1 Software Tools

- Visual studio code :editor



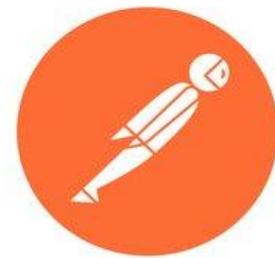
- Robo 3t :for visualizing data

## Robo 3T

mongodb



- Postman : for testing

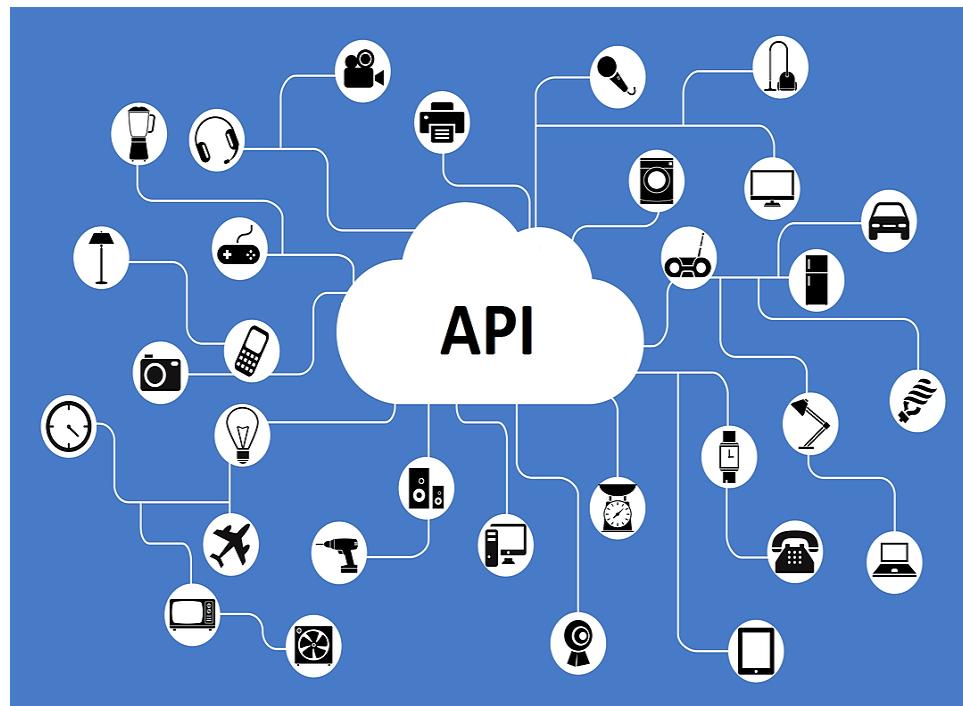


POSTMAN

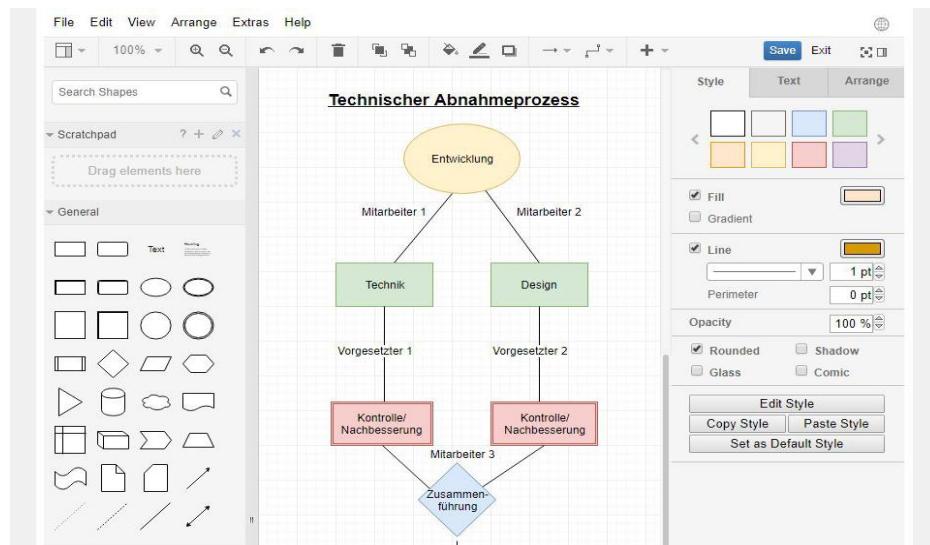
- Mongodb: the local database



- API :for making semantic search



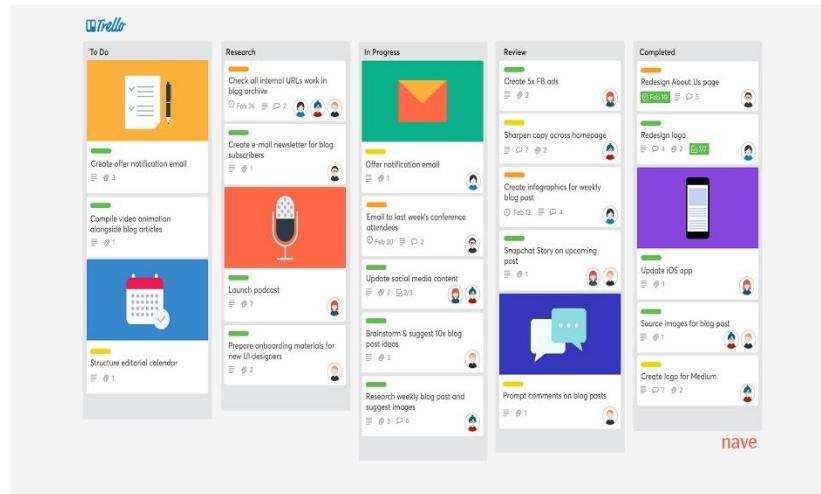
- Draw.io :for designing diagrams



- Photoshop : for wireframe & ui



- Trello: for dividing tasks



### ➤ 3.2.2 Hardware Tools

The project doesn't require any hardware tools.

## 3.3 Planning And Analysis

### 3.3.1 Project Planning

#### ➤ 3.3.1.1 Feasibility Study

A feasibility study used to determine if the project is achievable, so we will show some points to determine it.

##### **Problems that the project solves:**

- ✓ The difficulties of begging a start-up because there aren't a platform to market to in for free.
- ✓ The difficulties of not reaching to investors to get some benefits.
- ✓ The difficulties to enter a market lonely so there are a way to be a partner with someone.
- ✓ The unemployment has been increased and the market isn't available to all sectors anymore.

##### **Whom will the project benefit?**

- ✓ Entrepreneurs.
- ✓ Job seekers.
- ✓ Investors.
- ✓ Incubator.
- ✓ Funding organizations.
- ✓ Marketing organizations.
- ✓ Donators.

##### **Reasons why we chosen this project to work on:**

- ✓ Help entrepreneurs be involved with the market and investors.
- ✓ Help decreasing unemployment.
- ✓ Creating a free stage for entrepreneurs to help them.

- ✓ The country go forward putting efforts in start-ups.

## Market analysis:

### Main competitors:

- PATREON
- KIVA
- SEEDINVEST
- INDIEGOGO
- I FUND WOMEN

### Unique points in our project:

- A communication platform.
- The presence of a section for projects that you have not yet started for the need for any support.
- Saving comments.
- Variety of categories of start-ups.
- Locating the closest incubators, job seekers and buyers.
- Having an option to donate money to Startup but without profit, it is just a donation.
- The borrower can apply for a loan which is posted to the website for leaders to support and later the borrower repays the loan.

### Target Market:

- Age: 15 – 70.
- Gender: Both.
- Place: anywhere inside egypt.
- Income Level: 0- 20000\$ and above.
- Education level: Elementary school education and above.

- Relevant behavioral factor: can use a smart devices and dealing online with different languages.

## **Organizational and operational analysis:**

### **Number of team members this project will need:**

- 5 developers.
- 2 testers.
- 1 Customer services manager.
- 1 designer.
- 2 admins.
- 2 Public relation executives.

### **To keep the work on this project we shall provide:**

- Rules Manual.
- Regular Trainings.
- Safe Environment.
- Modern Management.
- Good work place.
- Efficient work tools.

## **Sales Strategy:**

### **To increase sales we have to take care of important points:**

- Check for and follow up customers' complaints regularly.
- Execute regular testing.
- Upgrade the web application to offer the customers optimized functions.
- Keep the good relations with the best incubators and investors.

- Claim and Maintain high rate on website.
- The web application is completely free but it will contain customized ads on the web application to be benefited from.
- Keep the clients engaged with promotions and offers that will be provided from entrepreneurs.

### **Technical analysis:**

**No Hardware required for the app.**

**Main technologies and programs we're associated with:**

- Web application –browsers.
- Angular Framework.
- Mongo DB Database.
- Microsoft Visual studio code.
- Adobe Photoshop.
- Draw.io.

### **Resource and Time analysis**

**Resources needed:**

- Devices for coding, running and testing – Laptops and mobile phones.
- Coding programs.
- Internet service.
- Design programs.
- API - packages.
- Node JS , angular and mongo DB libraries and packages.

### **Time schedule:**

- 1 month for planning.
- 2 months for training.
- 9 months for developing and testing.

### ➤ 3.3.1.2 Estimated Cost

**The budget of the application as programming and its process includes:**

- Purchasing a node js and angular course = 200\$.
- Marketing for the website = 99\$\year.
- Meeting in a public workspace.
  - Offline meeting every week (for 2-3 hours) in public work space (100 LE).

### ➤ 3.3.1.3 Gantt Chart

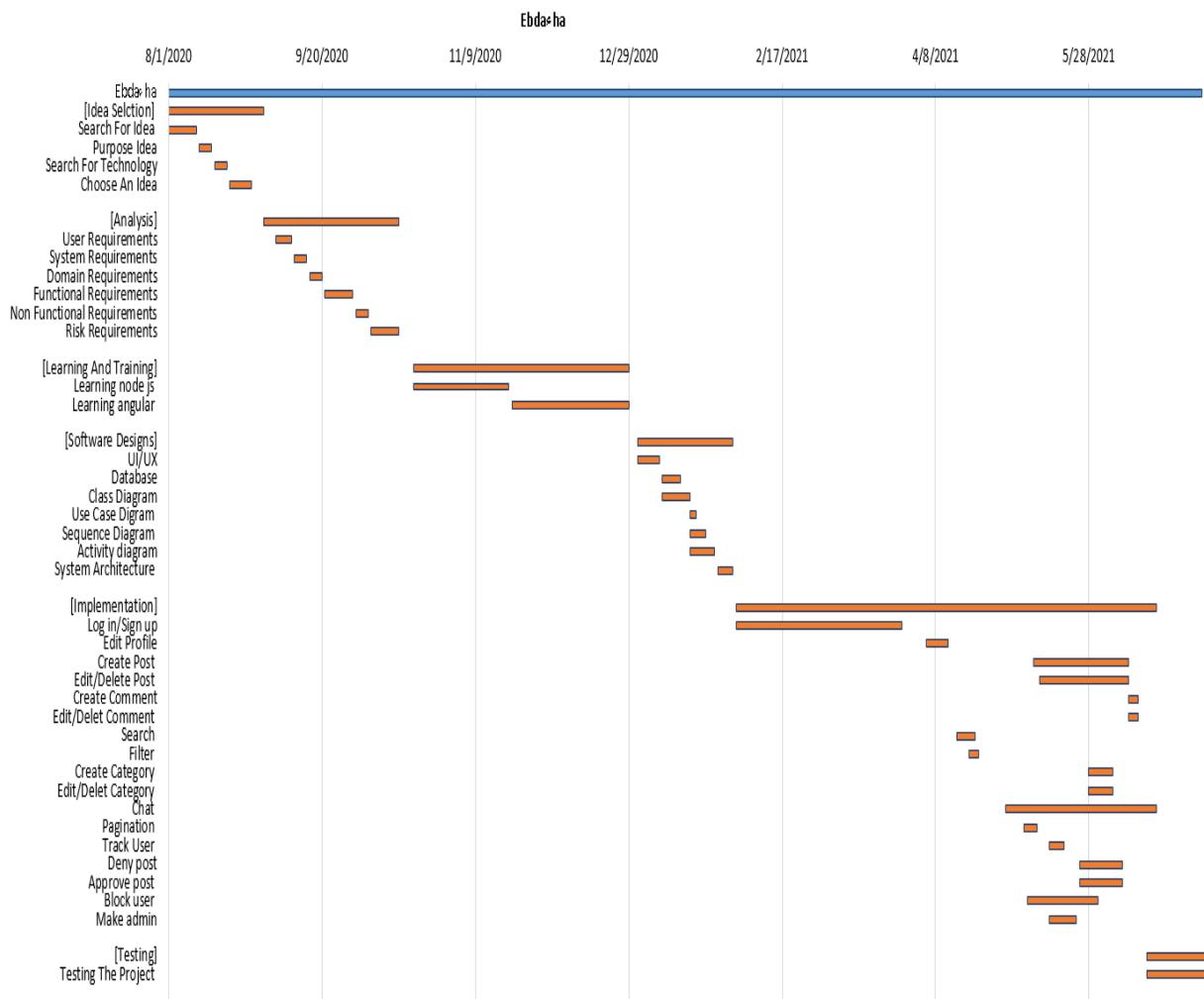


Figure 3-Gannt chart

### 3.3.2 Analysis And Limitation of Existing System

#### ➤ 3.3.2.1 Assumption And Constraints

We have existing competitors but we will discuss the features and what makes our website unique.

At first this table defines the features they have or not:

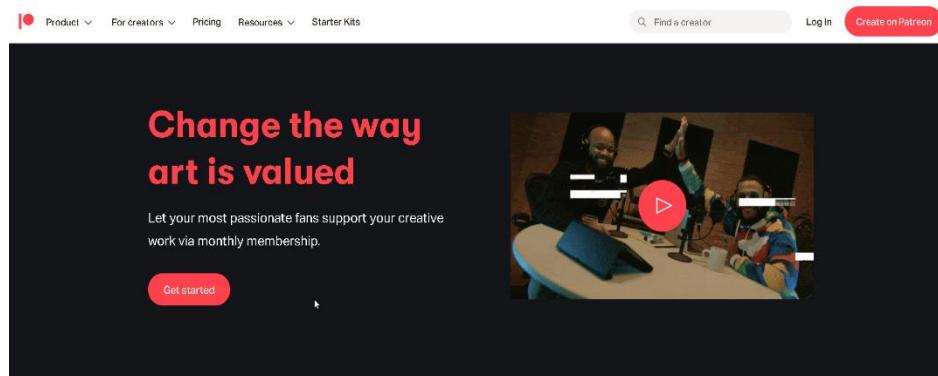
Website Logo	Website Name	Donate money	Invest money	Provide job	Provide loan
	PATREON	T	F	F	F
	KIVA	F	F	F	T
	SEEDINVEST	F	T	F	F

	INDIEGOGO	T	F	F	F
					
	I fund women	F	F	F	T
					
	Ebda2ha	T	T	T	T
					

Table 2-comaptitors

Now we will discuss each one individually:

### 1) PATREON



Search the 200,000+ creators on Patreon

Activate Windows  
Go to Settings to activate Windows.

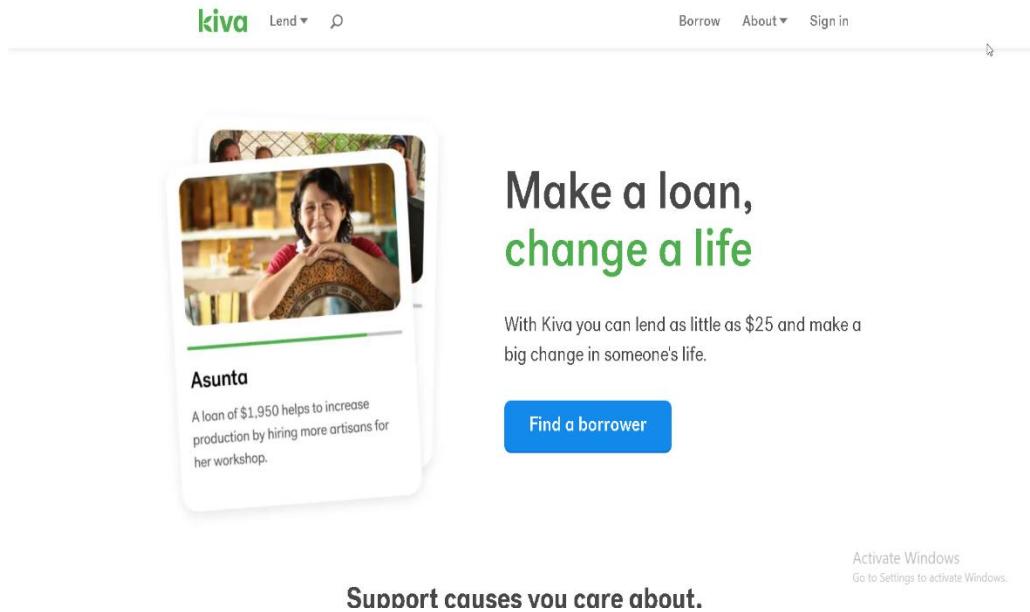
**Description:** Patreon is a way to enable people to support specific content creator by money (webcomics, videos, songs, ..etc).

Fans pay a few bucks per month OR per post you release, and then you get this money every month, or every time you release something new.

### Features:

- Make others donate to the category they want because they classify content creators in categories.
- Gives a variety to the one who wants support to choose which level he is in.
- Gives the one wants to support to choose he wants to pay per/week or per/month, ..etc.

## 2) KIVA



The screenshot shows the Kiva homepage. On the left, there's a profile card for a borrower named Asunta. She is shown smiling, wearing a red shirt, and holding a circular object. Below her photo, her name 'Asunta' is listed, followed by a description: 'A loan of \$1,950 helps to increase production by hiring more artisans for her workshop.' To the right of the profile card, the main headline reads 'Make a loan, change a life' in large, bold, dark gray and green text. Below this, a sub-headline says 'With Kiva you can lend as little as \$25 and make a big change in someone's life.' A blue button labeled 'Find a borrower' is positioned at the bottom of this section. At the very bottom of the page, there's a small note: 'Support causes you care about.' and a link to 'Activate Windows'.

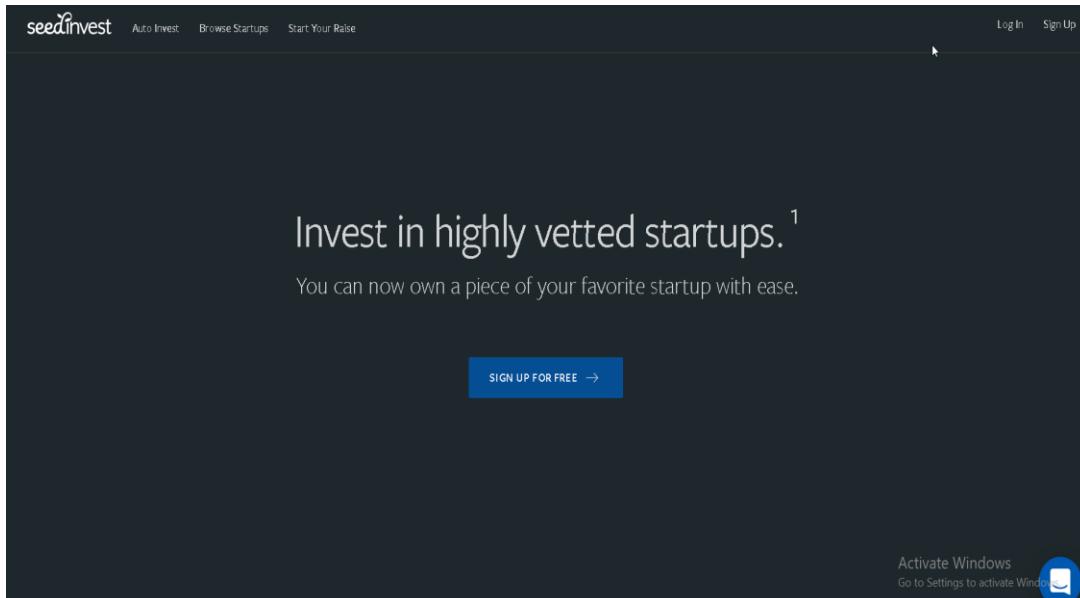
**Description:** Kiva is a non-profit organization with a mission of connecting people through lending to alleviate poverty. The borrower

can apply for a loan which is posted to the website for lenders to support and later the borrower repays the loan.

### **Features:**

- Kiva enables the borrower to apply for loan which is posted to its website
- At Kiva, loans aren't just about money—they're a way to create connection and relationships.

## **3) SEEDINVEST**



**Description:** SeedInvest is a leading equity crowdfunding platform, opening up access to venture capital and angel investing to everyone.

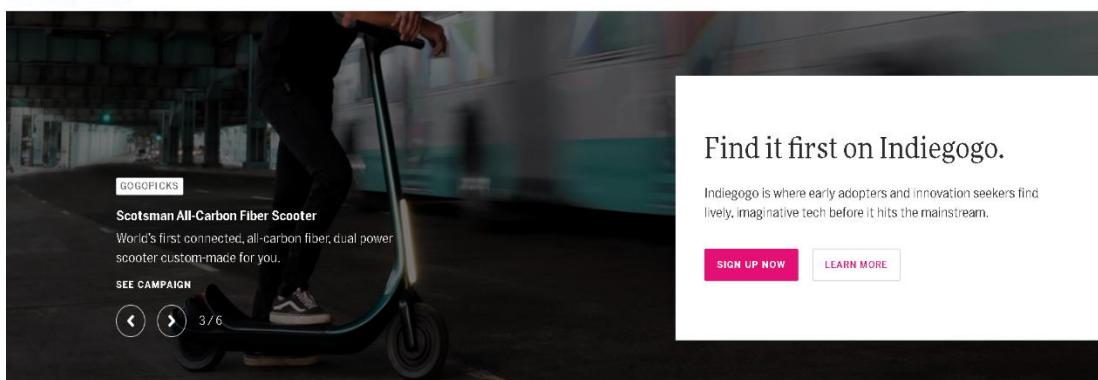
### **Features:**

- This website is used to offer new technologies things or things is creative.
- This website enables the incubators to invest their money in the projects.

## 4) INDIEGOGO

**INDIEGOGO** Explore ▾ What We Do

For Entrepreneurs Start a Campaign Log In Sign Up

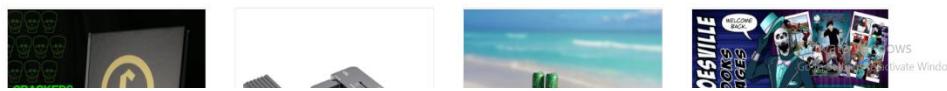


Find it first on Indiegogo.

Indiegogo is where early adopters and innovation seekers find lively, imaginative tech before it hits the mainstream.

[SIGN UP NOW](#) [LEARN MORE](#)

Popular Projects

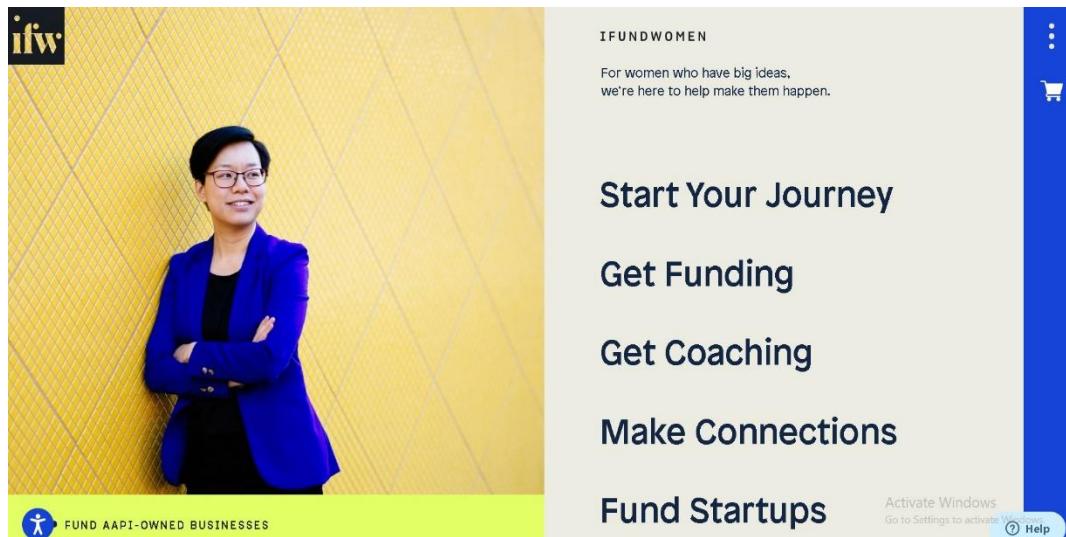


**Description:** Indiegogo's crowdfunding campaigns are where new and groundbreaking products take flight, sometimes long before they hit mainstream availability. With thousands of campaigns launching every week, there's great technology, design, and much more around every corner — often with limited-time perks and pricing for the earliest backers.

### Features:

- This website is used to offer new technologies things or things is creative.
- This website enables the incubators to denote his money in the projects.

## 5) I FUND WOMEN



**Description:** iFundWomen is a startup funding platform that provides female entrepreneurs with access to capital through crowdfunding and grants, expert business coaching and a growing network of women business owners.

**Features:**

- It funds the start-up upon the category.
- Gives woman strt-ups owners coaching courses in the field the want.
- Make some to get a grant or give it.

### ➤ 3.3.2.2 Risk And Risk Management

	Risk	Probability rate	Impact rate
Schedule Risk	Wrong time estimation.	Low likely	High
	Resources like team and their skills are not tracked properly.	Mid likely	High
	Failure to identify time required to develop functions.	Mid Likely	Mid
	Unexpected project growth.	Most likely	High

Table 3-schedule risk

#### Can be avoided by:

- Following up the Gantt chart regularly and reduce the critical paths in the project network.
- Look after the major tasks -specially- that the other tasks depend on.
- Monitor\review the agreed-on time plan.
- Scheduling the risky tasks first.

#### Can be reduced (if happened) by:

- Updating the plan to adapt with any new added requirements without affecting the already existed plan.
- Maximize the rest of resources like team and tools.
- Not wasting time because the problem but move forward.

	Risk	Probability rate	Impact rate
Budget Risk	Wrong budget estimation.	Most likely	Low
	Cost overruns	Low likely	Low
	Project growth	Low Likely	Mid

Table 4-budget risk

**Can be avoided by:**

- Good research.
- Regularly checking the budget spent and needed.
- Never skipping the budget through any new added requirements.

**Can be reduced (if happened) by:**

- Sacrifice not-so-needed resource that consume money.
- Quick adaptation.
- Replace cost-consuming parties with other available-resource consuming parties.

	Risk	Probability rate	Impact rate
<b>Operational Risk</b>	No resource planning.	Low likely	Mid
	No communication in the team.	Low likely	Mid
	No distribution for responsibilities.	Mid Likely	High
	Not enough training.	Mid likely	High

Table 5-operational risk

### Can be avoided by:

- Monitoring and evaluations at regular intervals.
- Cutting the project into very small tasks.
- Full training before proceeding the work.

### Can be reduced (if happened) by:

- Adding pieces of training into the normal work.
- Rotation through the training.
- Changing the responsibilities to achieve the best optimized output.
- Learning and planning again.

	Risk	Probability rate	Impact rate
Technical Risk	Changing requirements.	Most likely	Mid
	Angular and node js are in their initial stages.	Most Likely	High
	The database is not efficient.	Low likely	Mid
	Attacks or application failure	Mid likely	High

Table 6-technical risk

**Can be avoided by:**

- Preparing the threats and actions resulted.
- Understanding the platform\tools used.
- Not rushing the analysis phase.
- Specifying the desired outcome.
- Detailed test cases and regulations.

**Can be reduced (if happened) by:**

- Analyzing the security threats.
- Preparation for facing any attack.
- Quick adaptation to any new requirements.
- Evaluating \monitoring the database from time to time.

## 3.4 Requirements

### ➤ 3.4.1 User requirements

#### 3.4.1.1 Mandatory Requirements:

- The user make an account and enter his data easily in few clicks.
- The user is notified when registering by sending email to his account to insure security.
- The user can posting about his start-up and what it is providing.
- The user can connect to investors to help him.
- The user can chatting with any one he wants.
- The admin can approve or deny posts and comments.

#### 3.4.1.2 Desirable Requirements

- The user can search for investors by location.
- The user can upload a file about the services he can provide immediately.
- The user can share the start-up location if it has a location.
- Notification for users if there is any new service.

### ➤ 3.4.2 System requirements

- **Browser**

Since our web application will be written with node js/angular then it will works on google, firefox, safari browsers.

- **Internet connection**

Since we would need to keep the users data in the cloud, we need an internet connection to store and retrieve the data.

- **Storage space**

In order to keep the user's data in addition to the web application itself, so we need a fine storage.

➤ **3.4.3 Domain Requirements**

- Enough memory should be reserved for on the database server to accommodate for any number of users and their data.
- The necessary browser required to browse the web application.
- Each user must have a unique user-name/password combination.
- The database should be backed up every occasionally in case the original does become corrupt.
- The web application must have update capabilities for future models and accessories.
- The web application must verify all values before making the change in the database.

➤ **3.4.4 Functional Requirements**

- **Sign in**

- Actor : User (any user except admin).
- Pre : having internet connection and opening the website to click on "join us" button.
- Description : This function is to make the user register his account with email and password to be a unique combination.

- Post : the user will be sent to him an email to verify account.

- **Verify Email**

- Actor : User (any user except admin).
- Pre : having internet connection and have been registered on the web site, then opens his email.
- Description : This function is to make the user verify his account to be registered in the system.
- Post : the account will be verified and can log in.

- **Log In**

- Actor : All users
- Pre : The user should be registered in the website and having internet connection to open the website to click on "Log in" button .
- Description : This function is to make the user log in to his account with password and email.
- Post : the user will successfully enter his account.

- **Forget Password**

- Actor : All users
- Pre : the user should be registered in the website and then open the website to click on "forget password" button then enter his email.
- Description : This function is to enable the user to register with new password if he doesn't able to remember it.
- Post : the user password is edited successfully.

- **Edit Profile**

- Actor : User (any user except admin).
- Pre : the user should open the website to click on his profile then “edit” button.
- Description : This function is to enable the user to update his information and entering the data that were missing.
- Post : the user profile is edited successfully.

- **Edit profile picture**

- Actor : User (any user except admin).
- Pre : the user should open the website to click on his profile then “edit photo” button.
- Description : This function is to enable the user to update his profile picture.
- Post : the user’s profile picture is edited successfully.

- **Delete profile picture**

- Actor : User (any user except admin).
- Pre : the user should open the website to click on his profile then “delete photo” button.
- Description : This function is to enable the user to delete his profile picture.
- Post : the user profile picture is deleted successfully.

- **Change Password**

- Actor : All users.
- Pre : the user should log in firstly.

- Description : This function is to enable the user to update his password if he remembers it.
- Post : the user password is updated successfully.

- **Get All Users**

- Actor : All users.
- Pre : the user should log in firstly.
- Description : This function is to enable the user to get all users in the system.
- Post : the admin gets all users successfully.

- **Get User**

- Actor : All users.
- Pre : the user should log in firstly.
- Description : This function is to enable the admin to get specific user in the system.
- Post : the admin gets the user successfully.

- **Make Admin**

- Actor : User (admin only).
- Pre : the user should log in firstly and the user is exist in the system.
- Description : This function is to enable the admin to make another user admin with him.
- Post : the admin make other users admins also successfully.

- **Approve post**

- Actor : User (admin only).

- Pre : the user should log in firstly and the user is exist in the system.
- Description : This function is to enable the admin to make approve posts.
- Post : the admin make posts approved and makes them present in categories.

- **Deny post**

- Actor : User (admin only).
- Pre : the user should log in firstly and the user is exist in the system.
- Description : This function is to enable the admin to deny posts.
- Post : the admin make posts removed and not accepted.

- **Block User**

- Actor : User (admin only).
- Pre : the user should log in firstly and the user is exist in the system.
- Description : This function is to enable the admin to block specific user in the system.
- Post : the admin blocks the user successfully.

- **filter**

- Actor : All users
- Pre : the user should log in firstly.
- Description : This function is to enable the admin to filter users in the system by name.

- Post : the admin gets the users he wants successfully.

- **Search For Post**

- Actor : User (any user).
- Pre : the user should log in firstly.
- Description : This function is to enable the user to search for some relates posts in the system and this is semantic.
- Post : the user gets posts successfully.

- **Filter Posts**

- Actor : User (any user).
- Pre : the user should log in firstly.
- Description : This function is to enable the user to filter posts in the system by category.
- Post : the user gets the posts of the category successfully.

- **Create Post**

- Actor : User (all users except admin).
- Pre : the user should log in firstly.
- Description : This function is to enable the user to create posts in the system.
- Post : the user creates the post successfully.

- **View Post**

- Actor : User (all users except admin).
- Pre : the user should log in firstly.

- Description : This function is to enable the user to view posts in two options if he searched before for specific event then the website will display all the posts related to the search like cookies, or if he didn't do any search before then the website will view random posts.
- Post : some type of posts display successfully.

- **Update Post**

- Actor : User (all users except admin).
- Pre : the user should log in firstly and created a post.
- Description : This function is to enable the user to update the data of his posts in the system.
- Post : the user update the posts successfully.

- **Get All Unapproved Posts**

- Actor : User (admin only).
- Pre : the user should log in firstly.
- Description : This function is to enable the admin to get unapproved posts in the system.
- Post : the admin gets unapproved posts successfully.

- **Get All Posts**

- Actor : User (any user except).
- Pre : the user should log in firstly.
- Description : This function is to enable the user to get all posts in the system.

- Post : the user gets all post successfully.

- **Find Post**

- Actor : User (any user except admin ).
- Pre : the user should log in firstly.
- Description : This function is to enable the user to find specific post in the system.
- Post : the user gets the post successfully.

- **Create Comment**

- Actor : User (all users except admin).
- Pre : the user should log in firstly and there are posts in the system.
- Description : This function is to enable the user to create comments in the system.
- Post : the user creates the comment successfully.

- **Get All Comments**

- Actor : User (any user except admin).
- Pre : the user should log in firstly and added some comments.
- Description : This function is to enable the user to get all his comments in the system.
- Post : the user gets all comments successfully.

- **Update Comment**

- Actor : User (all users except admin).
- Pre : the user should log in firstly and created a Comment.

- Description : This function is to enable the user to update the data of his comments in the system.
- Post : the user update the comments successfully.

- **Get Single Comment**

- Actor : User (any user except admin).
- Pre : the user should log in firstly and added a comment.
- Description : This function is to enable the user to get specific comment in the system.
- Post : the user gets the comment successfully.

- **Delete One Comment**

- Actor : User (any user except admin).
- Pre : the user should log in firstly and added a comment.
- Description : This function is to enable the user to delete specific comment in the system.
- Post : the user deletes the comment successfully.

- **Delete All Comments**

- Actor : User (any user except admin).
- Pre : the user should log in firstly and added some comment.
- Description : This function is to enable the user to delete all comments in the system.
- Post : the user deletes all comments successfully.

- **Get All Mails**

- Actor : User (admin only).
- Pre : the admin should log in firstly and there are some mails, then press on button of get all mails.
- Description : This function is to enable the admin to get all mails in the system.
- Post : the admin gets all mails successfully.

- **Send Contact Mail**

- Actor : User (any user except admin).
- Pre : the user should log in firstly.
- Description : This function is to enable the user to send mail with feedback.
- Post : the email is sent successfully.

- **Get Single Mail**

- Actor : User (admin only).
- Pre : the admin should log in firstly press button of getting single mail.
- Description : This function is to enable the admin to get specific mail in the system.
- Post : the admin gets the mail successfully.

- **Delete Single Mail**

- Actor : User (admin only).

- Pre : the admin should log in firstly and pressed the button of delete mail.
- Description : This function is to enable the admin to delete specific mail in the system.
- Post : the admin deletes the mail successfully.

- **Get Single Category**

- Actor : User (admin only).
- Pre : the admin should log in firstly press button of getting single category.
- Description : This function is to enable the admin to get specific Category in the system.
- Post : the admin gets the Category successfully.

- **Get All Categories**

- Actor : All users
- Pre : the admin should log in firstly and there are some Categories, then press on button of get all categories
- Description : This function is to enable the admin to get all categories in the system.
- Post : the admin gets all categories successfully.

- **Create Category**

- Actor : User (admin only).
- Pre : the user should log in firstly.
- Description : This function is to enable the user to create category in the system.

- Post : the user creates the category successfully.

- **Update Category**

- Actor : User (admin only).
- Pre : the user should log in firstly and created a category.
- Description : This function is to enable the user to update the data of his categories in the system.
- Post : the user updates the category successfully.

- **Delete Category**

- Actor : User (admin only).
- Pre : the user should log in firstly and added a comment.
- Description : This function is to enable the admin delete the category and all posts exist in that category.
- Post : the user deletes the category and its related posts successfully.

- **Chat**

- Actor : Users except admin.
- Pre : the user should log in firstly and added a comment.
- Description : This function is to enable the user to chat with other user.
- Post : the message of specific user is sent successfully to the other user.

### ➤ 3.4.5 Non-Functional Requirements

- **Availability**

- The website will be available to anyone has internet connect, it can be accessed 24 hours in 7 days.
- The website unfortunately will not work if there isn't internet connection and a browser.
- The storage in database is very huge so the user can creating posts as long as he wants.
- Under bad circumstances, the web application shall present a user with notification message informing him that the system is unavailable now.

- **Maintainability**

- If the web application is under maintenance, we'll firstly inform the users via notification if it will affect any vital function, then the existed data of the user shall remain saved aside but the newly added data won't be added till the maintained is over.
- Any new feature must not impact any of the dashboard functions.
- The system shall not be shut down for maintenance more than once in a 48-hour period.

- **Security**

- Any unauthorized access to the web application is denied, no one can use the application without creating an authorized account.
- All the users are verified.

- Authentication and authorization made for every role of registered users.
  - Passwords shall never be viewable at the point of entry or at any other time.
  - The access permissions for any data may only be changed by the web application developers.
  - The data that the user posts can be only changed be the user himself.
- 
- **Usability**
  - The web application will have a page called about us to help the user understand the website and how to deal with.
  - The user can add his data within few clicks, which will take no time to submit.
  - The accessibility of interface is very important since any type of people are likely to use the app, so that we followed Google material design, colors and fonts.
  - The application shall be easy to use by adult members (age 15 to 70).
  - All the main tasks are shown in the dashboard with a different colors, all the buttons are oblivious.
- 
- **Flexibility**
  - No piece of text that might be displayed to a user shall reside in program source code. Every piece of text that a user might see must be modifiable without changing source code.

- The data the user entered even posts can be updated or even deleted.
- Users' data shall be suitable for use by any other type of users in any country.
- The application shall be easy to use by adult members (age 15 to 70).
- All the main tasks are shown in the dashboard with a different colors, all the buttons are oblivious.

- **Scalability**

- The web application will be updated so it can serve more users (that increases gradually), process more data and do more transactions.
- Any added feature or function mustn't affect the storage massively.
- Time to save any data mustn't increase with users' number increasing.
- Functions and roles shall be scaled for the number-growth of the users.
- The admin dashboard shall support unlimited users inquiries.

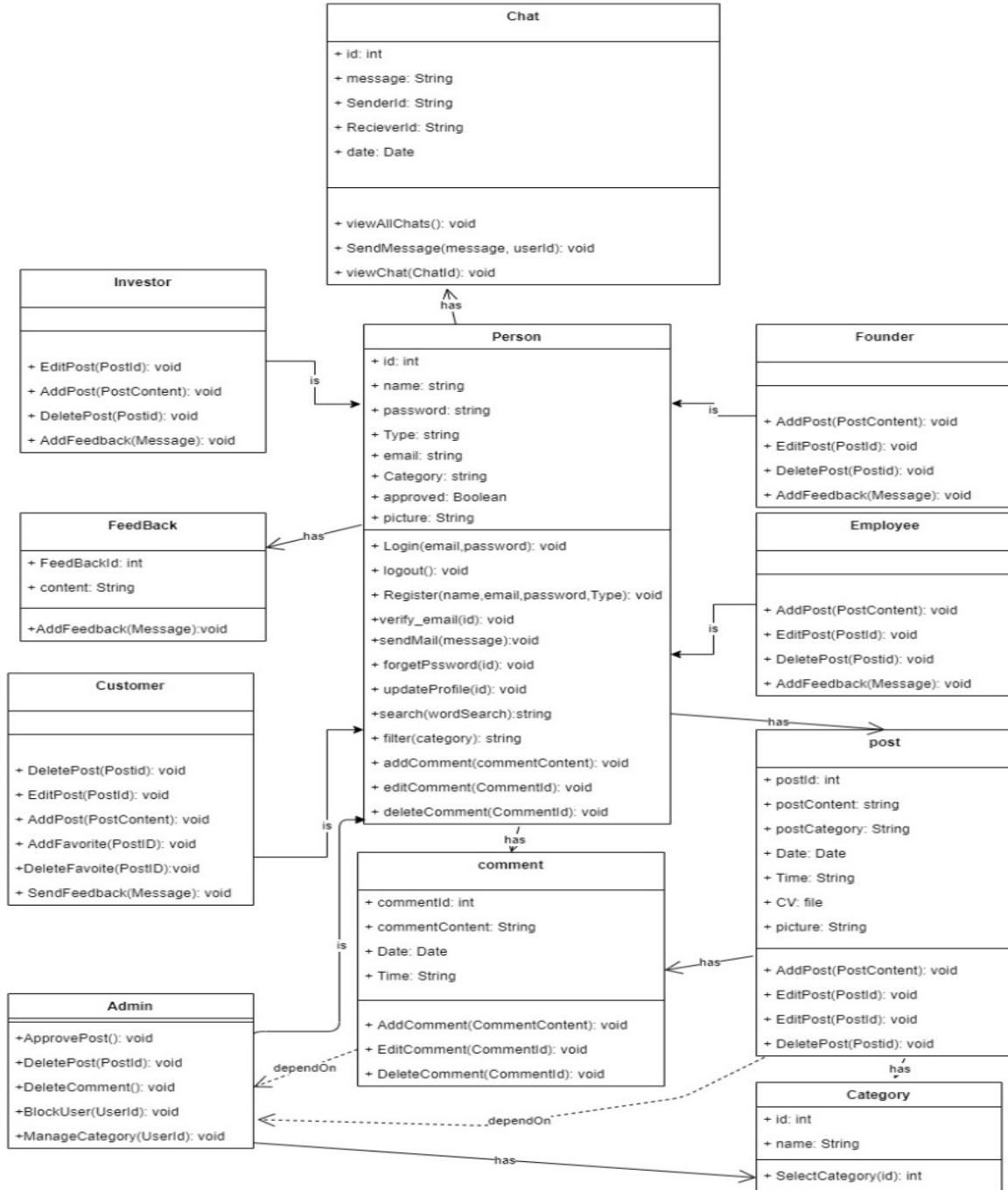
- **Efficiency**

- The web application needs no storage in any smart device, because it social and online platform.
- The web application shall be able to handle the entry of users' data at a minimum rate of 5 per second.

- Any live interaction between users will not take time because the using of socket-io chatting.
- The posts and comments are reviewed and approved within minutes.
- Routine maintenance that is executed while users are active shall not cause any delay of more than 10% from the normal for the users.

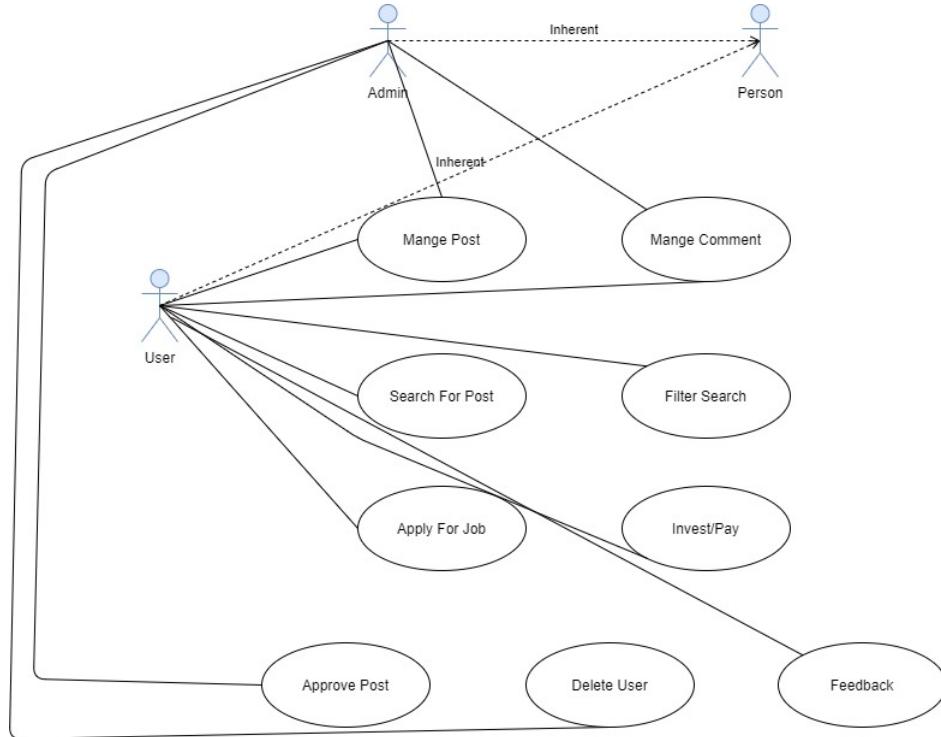
## 3.5 Software Design

- 3.5.1 Class diagram

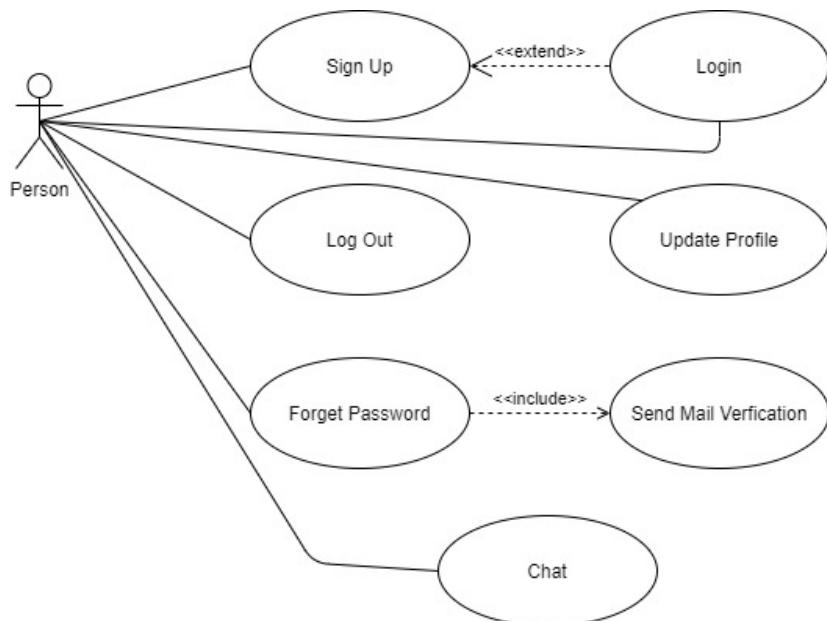


- 3.5.2 Use case diagrams

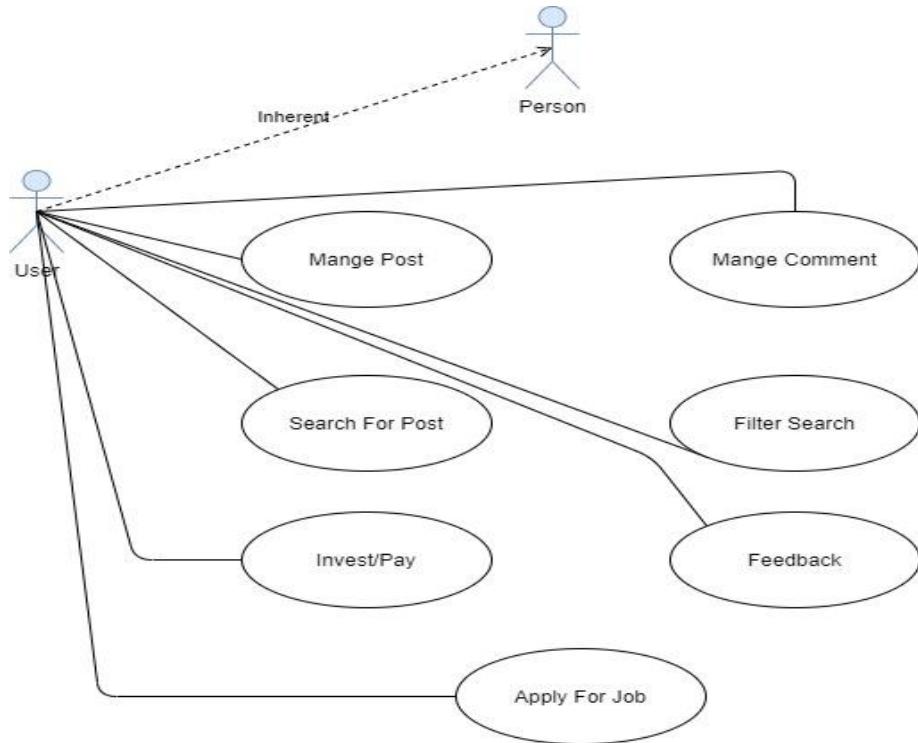
- ✓ The whole system



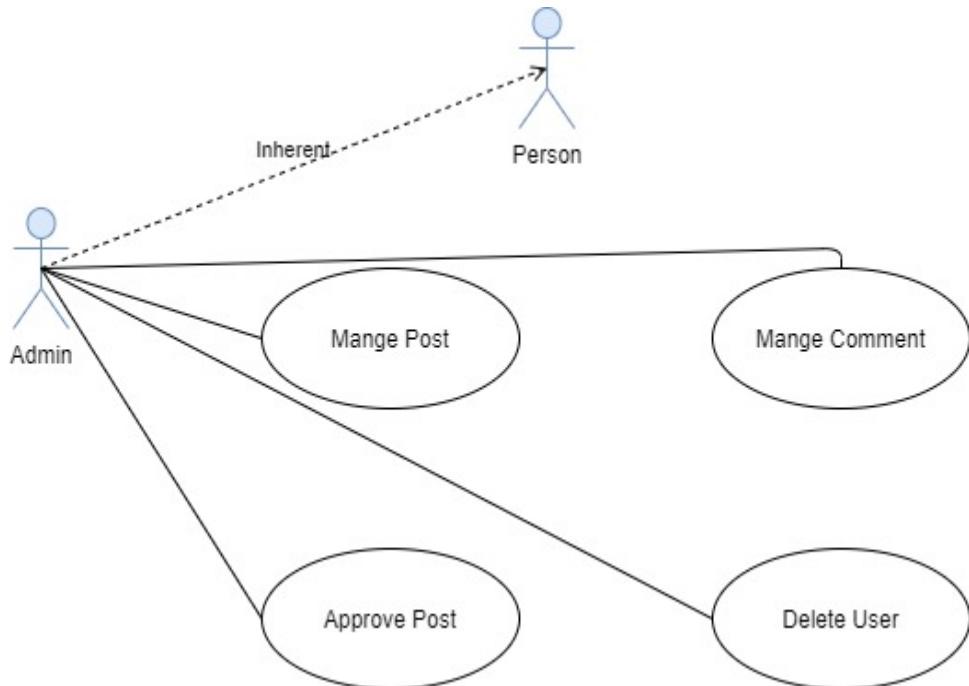
- ✓ Main common features



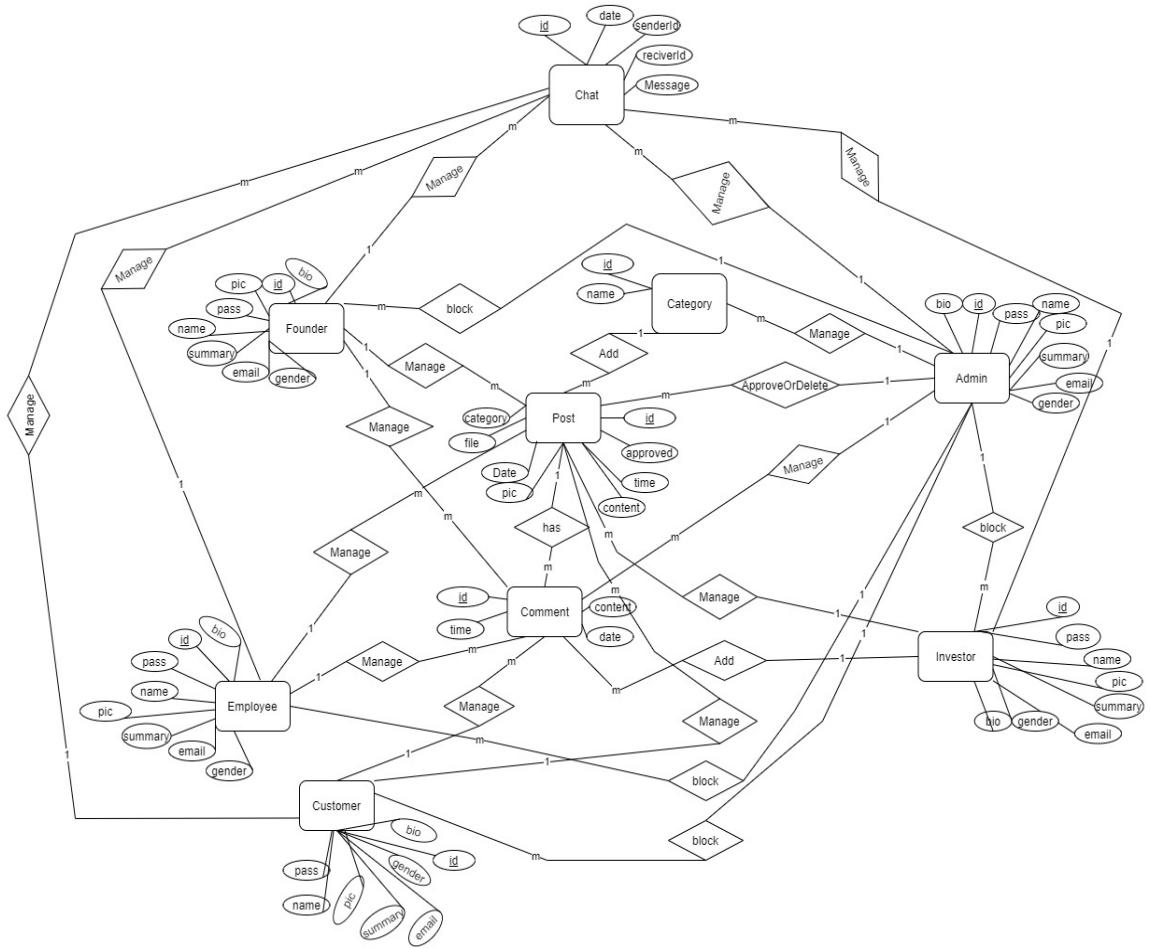
## ✓ User



## ✓ Admin

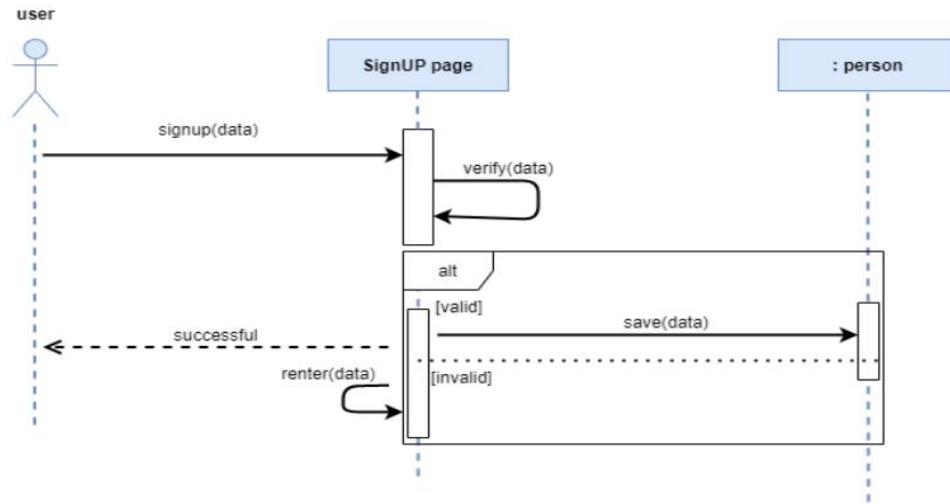


- 3.5.3 Erd diagram

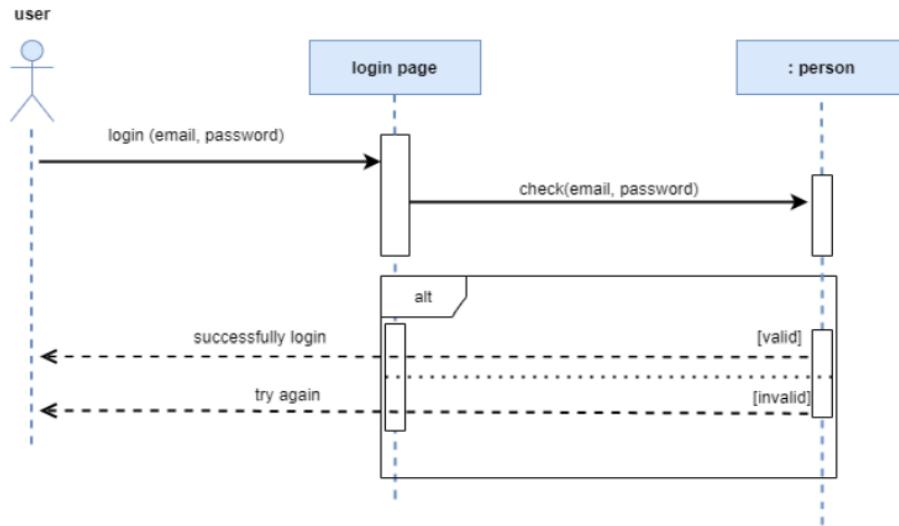


- 3.5.4 Sequence diagrams

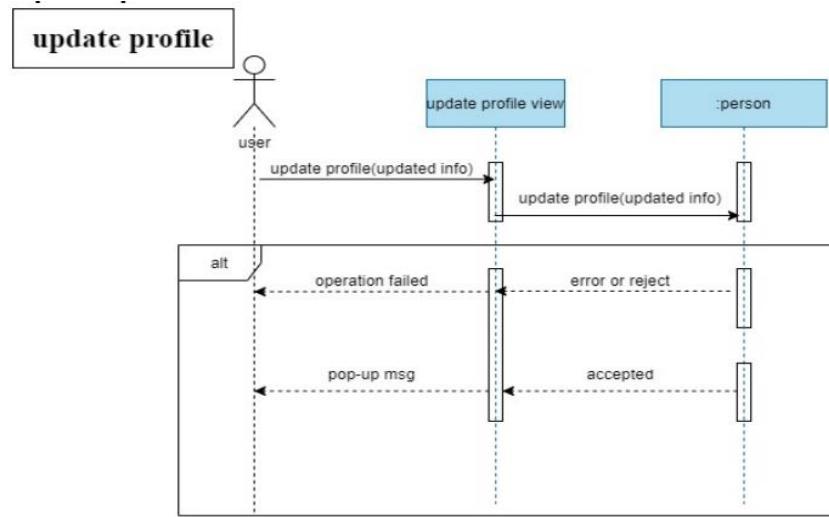
- ✓ Sign up



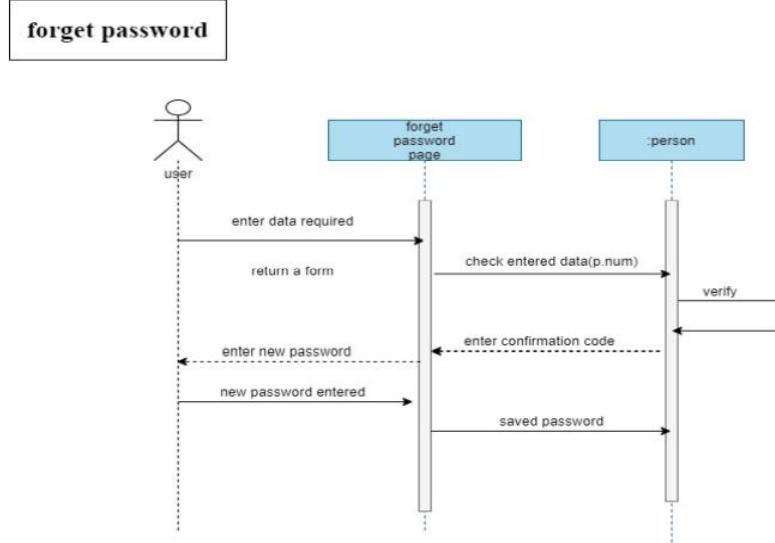
### ✓ Log in



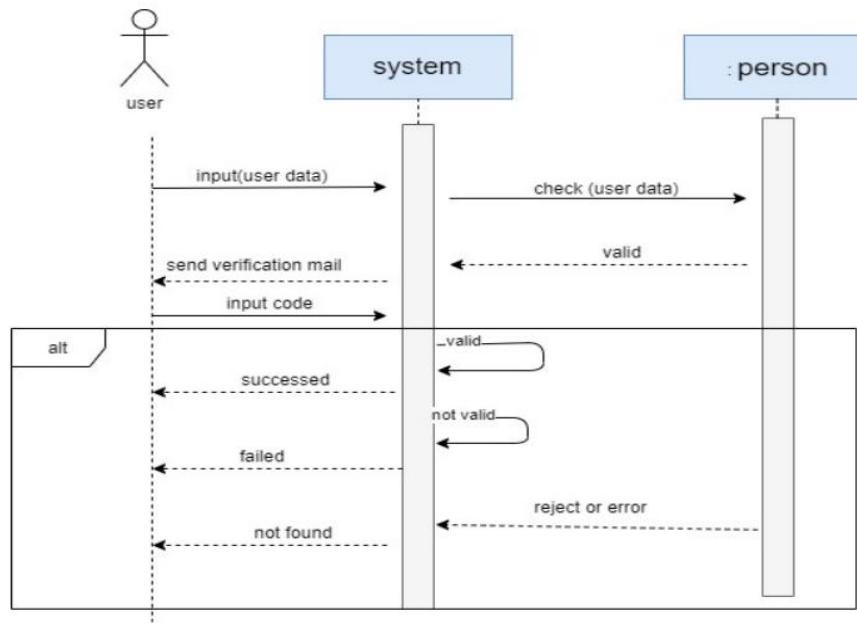
### ✓ Update profile



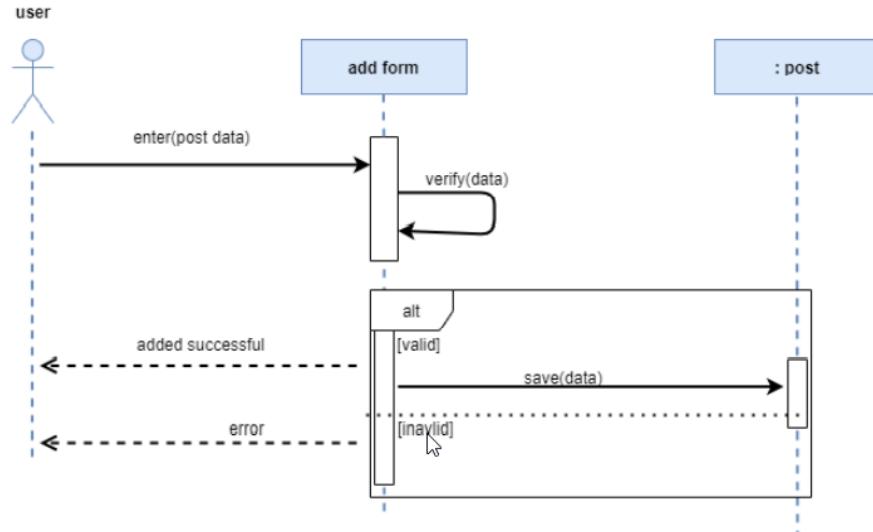
✓ Forget password



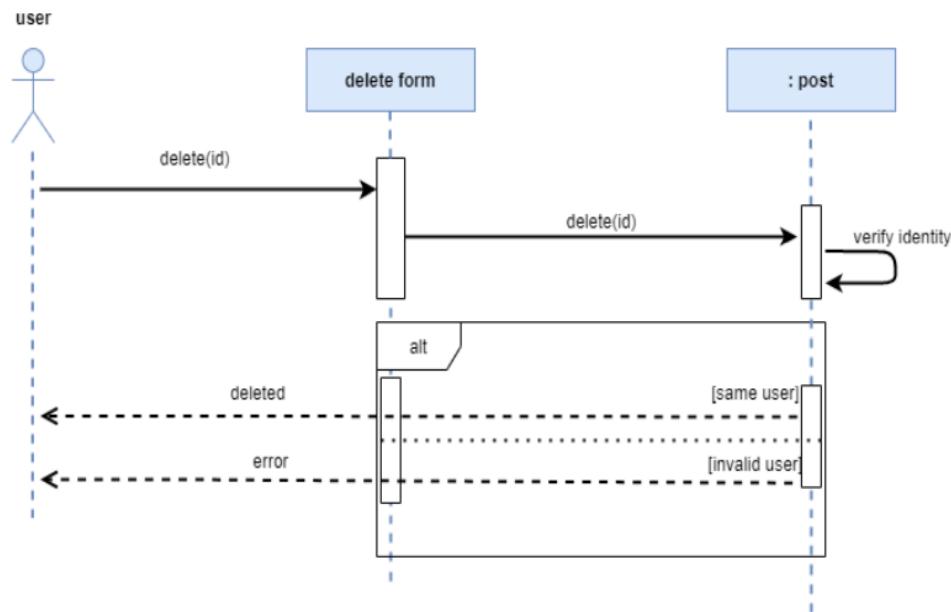
✓ Send mail verification



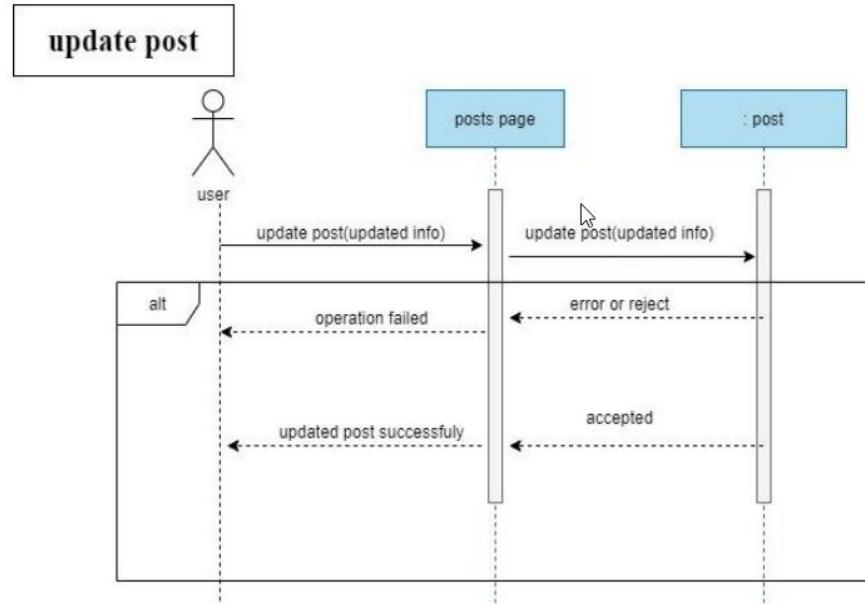
### ✓ Add post



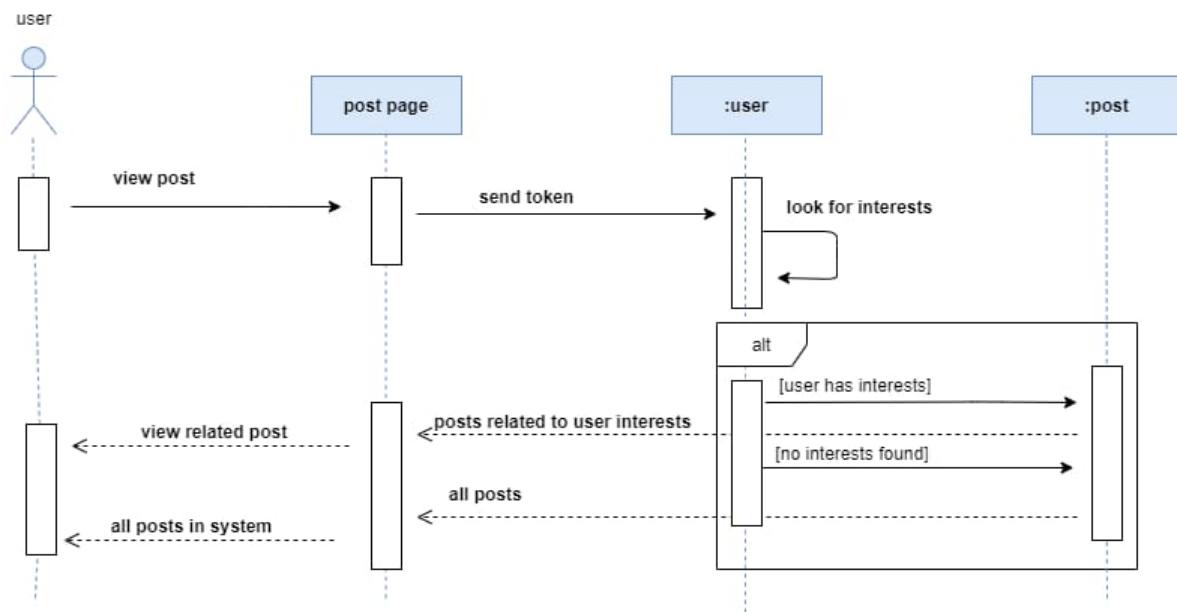
### ✓ Delete post



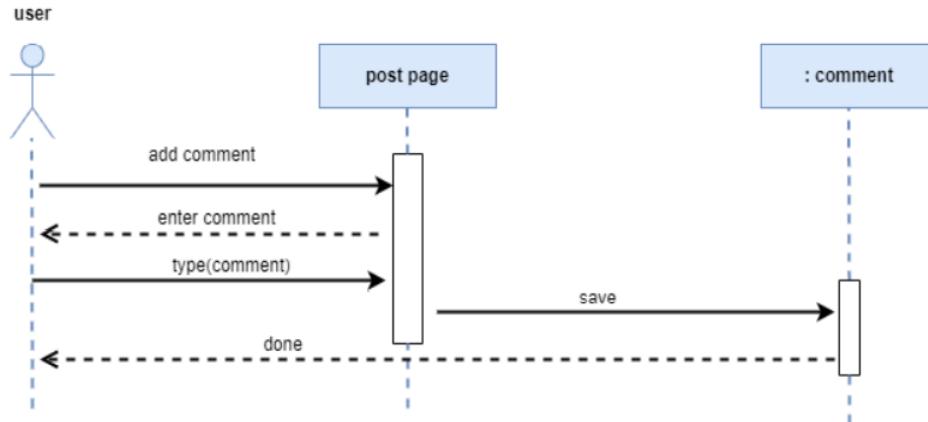
✓ Update post



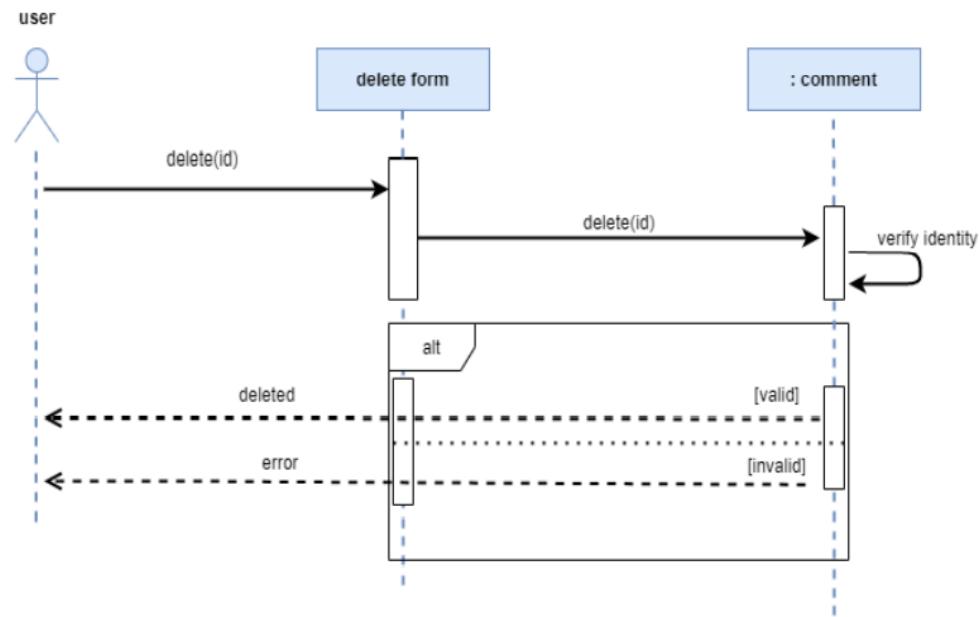
✓ View post



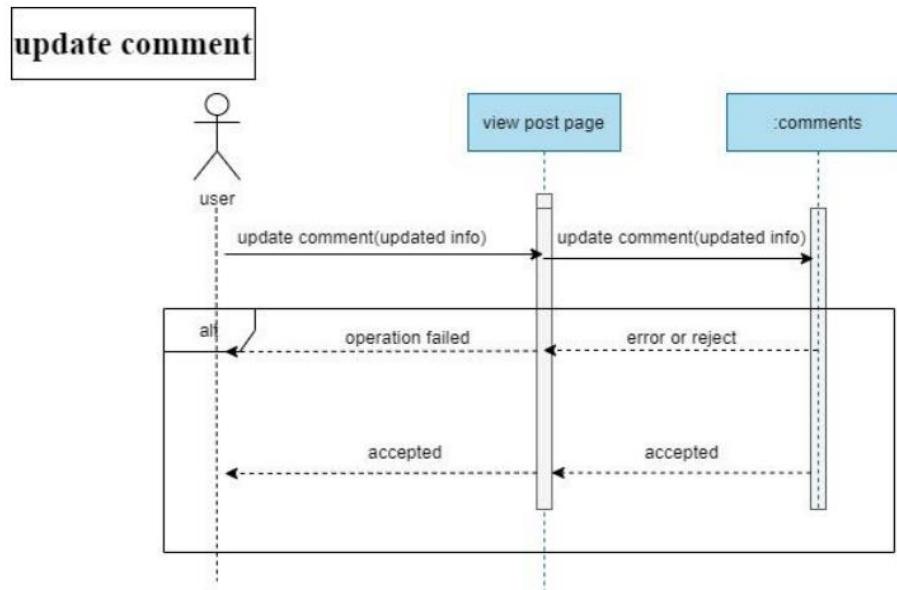
### ✓ Add comment



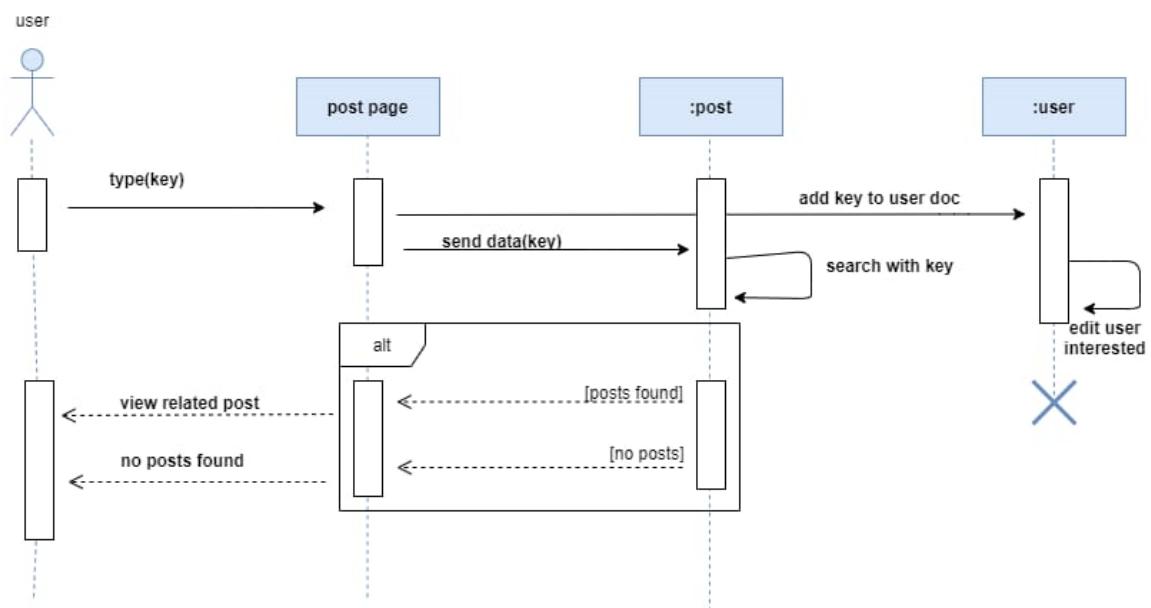
### ✓ Delete comment



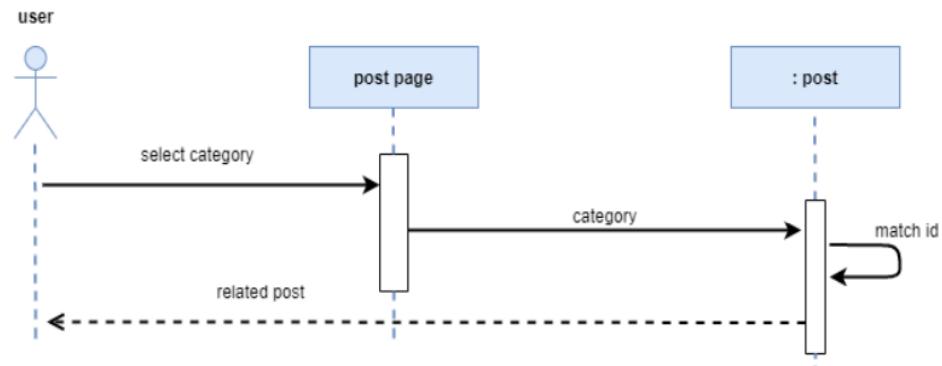
### ✓ Update comment



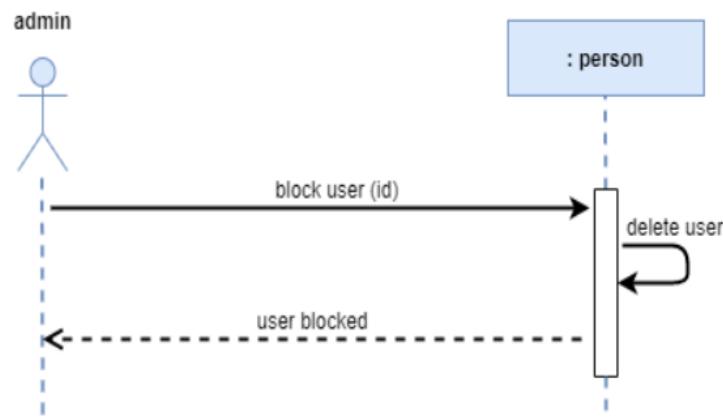
### ✓ Semantic search



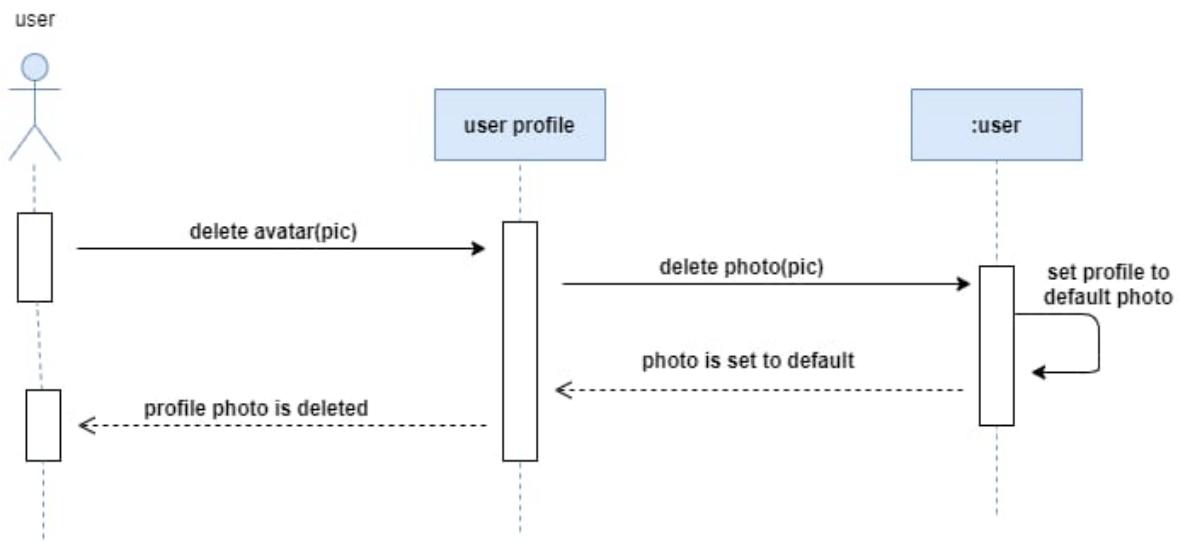
✓ Filter



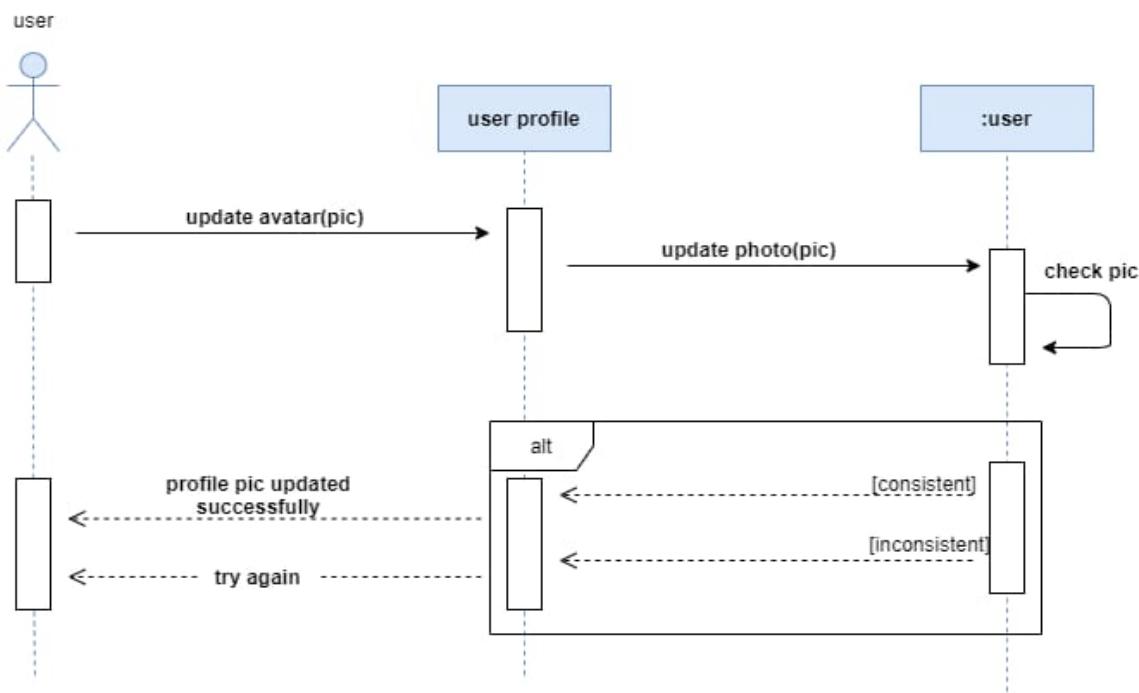
✓ Block user



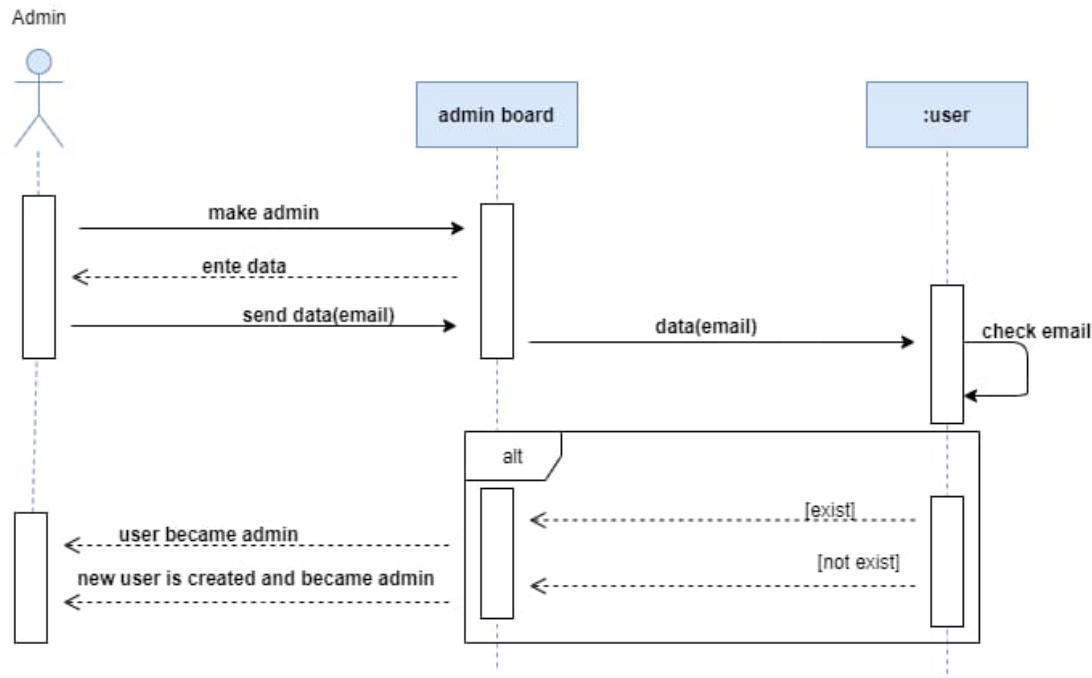
✓ Delete profile picture



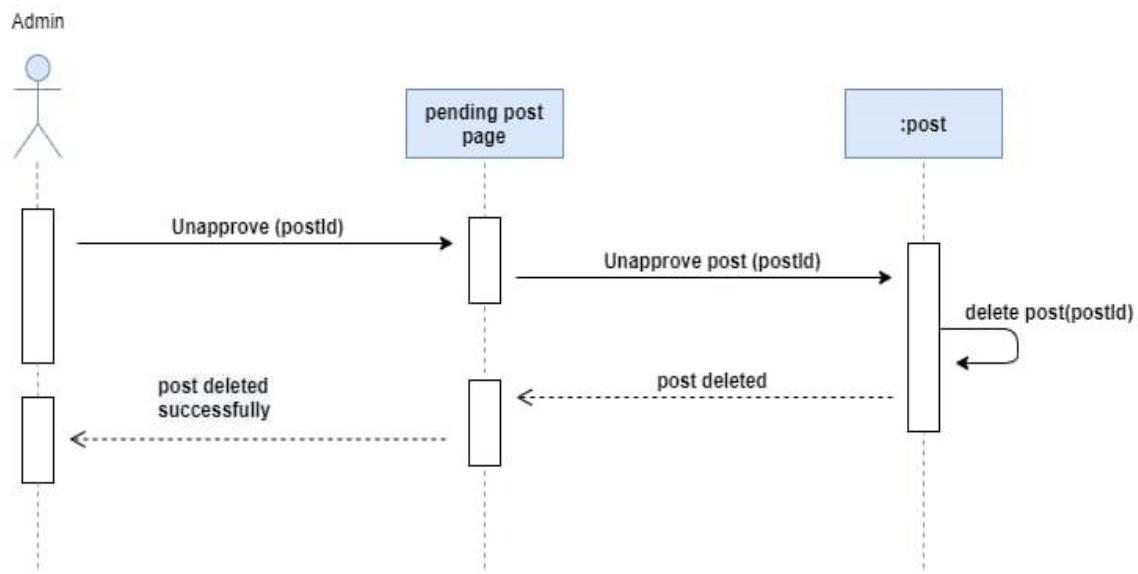
✓ Edit profile picture



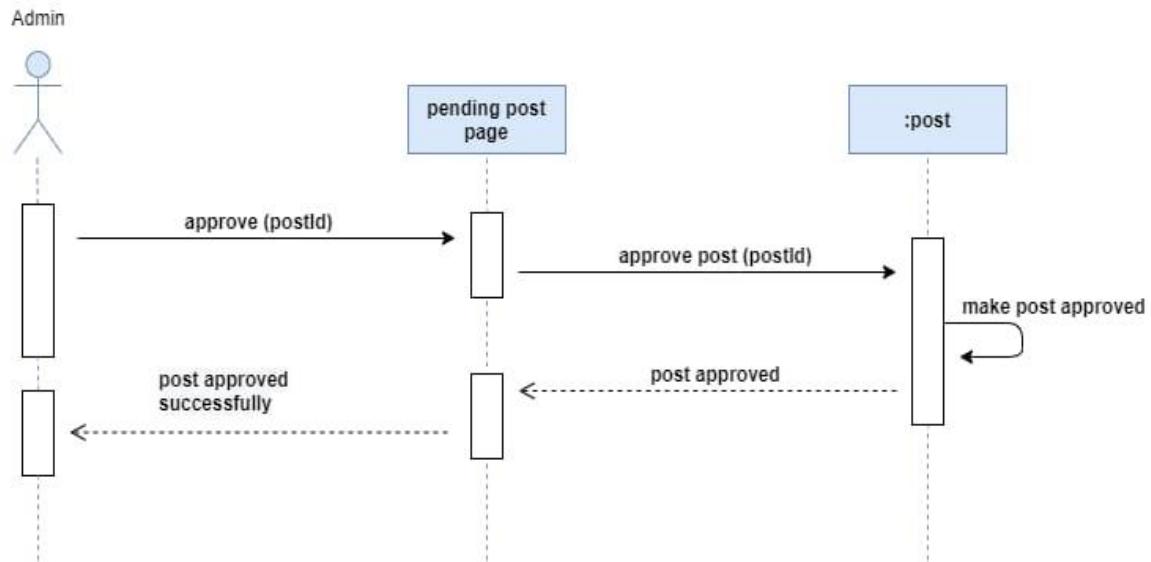
### ✓ Make admin



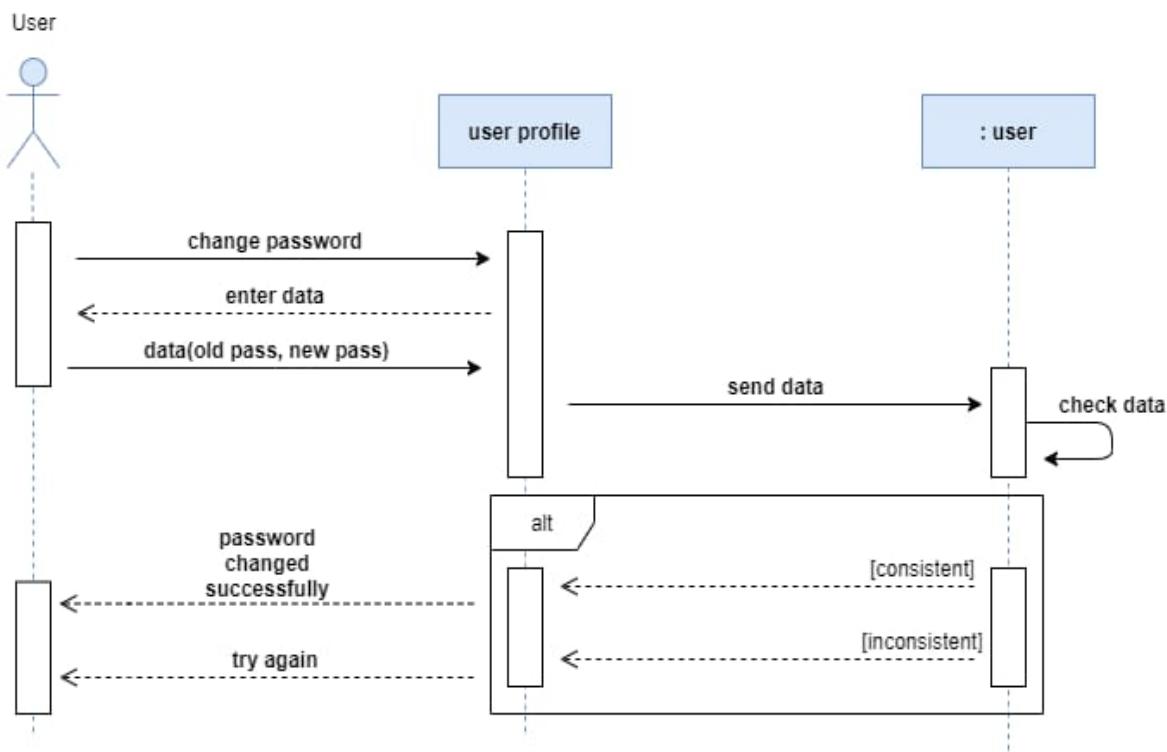
### ✓ Unapproved post



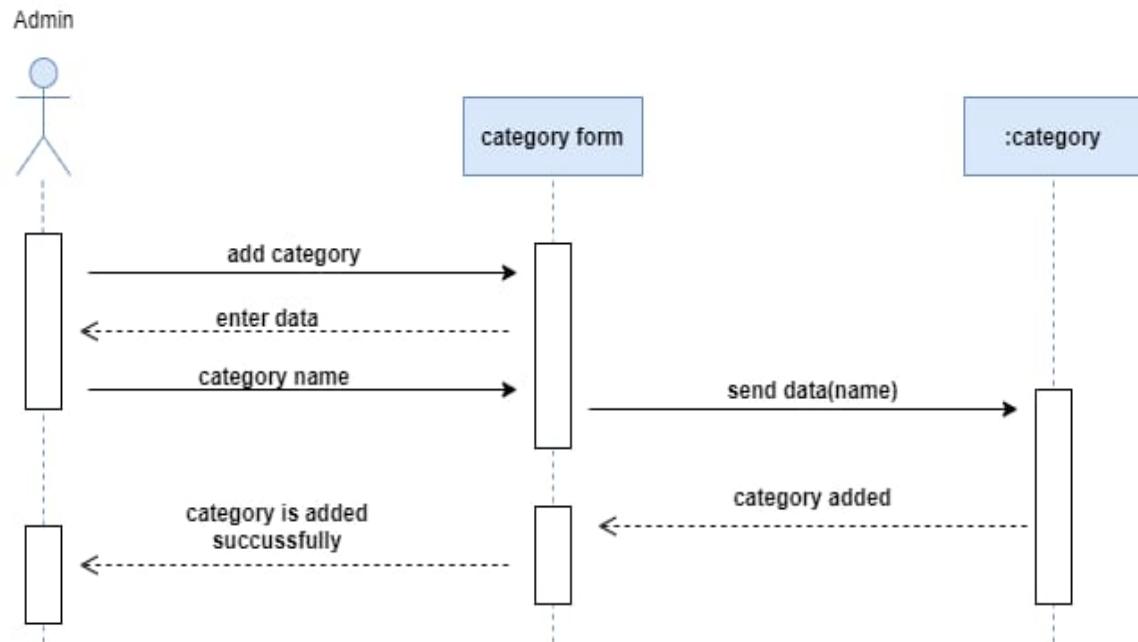
### ✓ Approved post



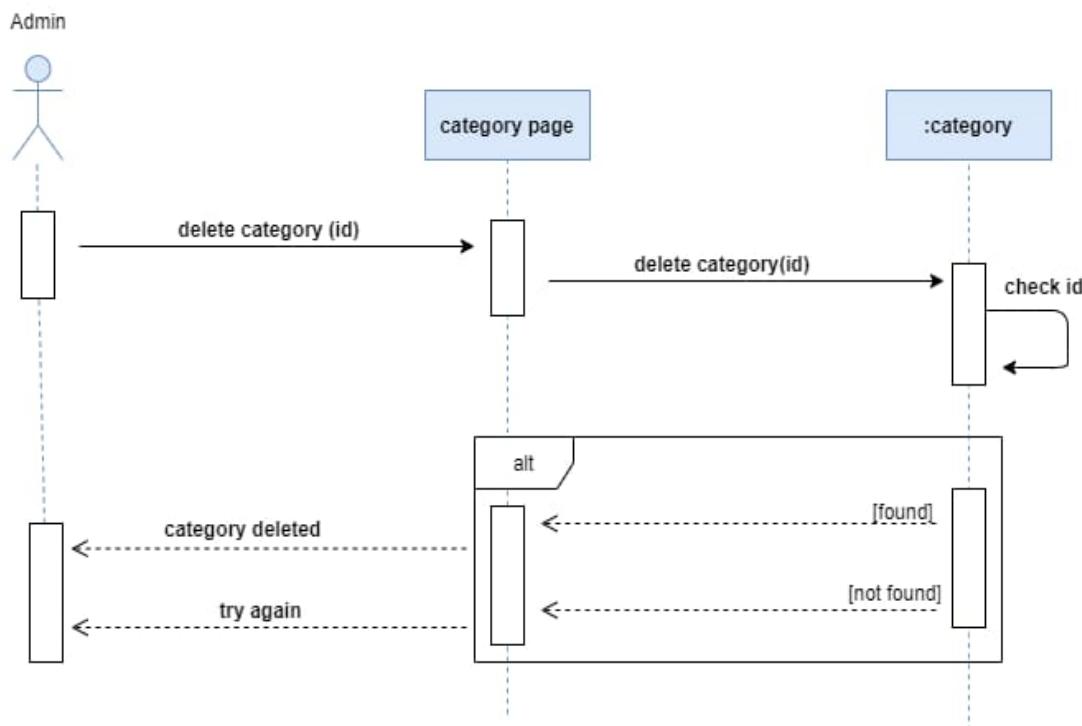
### ✓ Change password



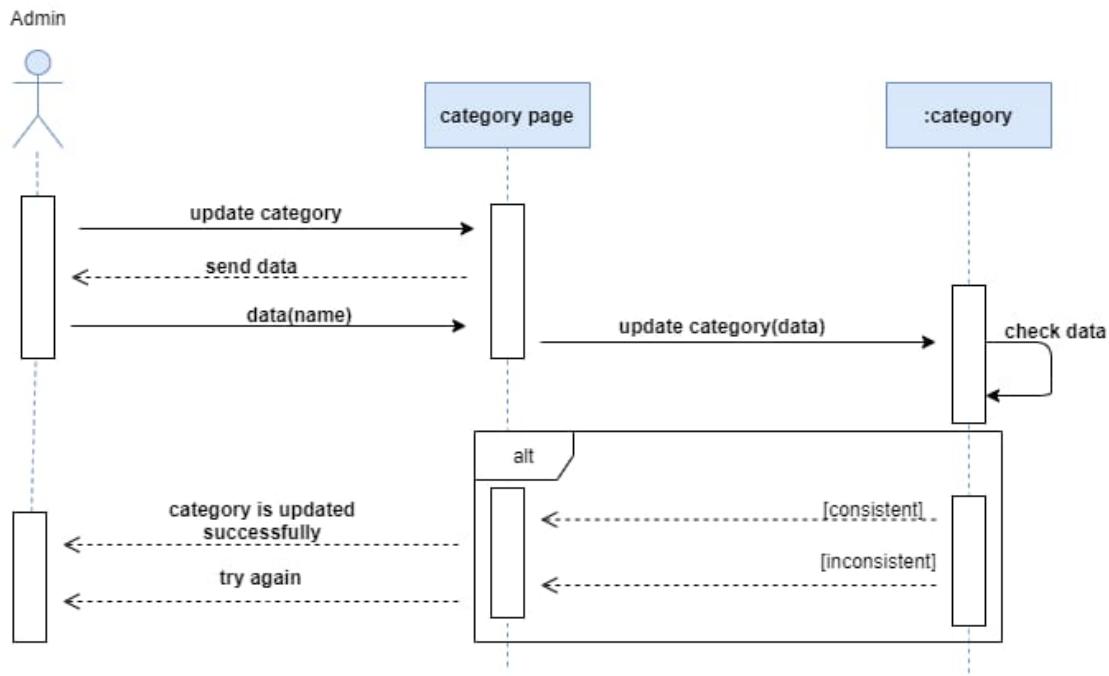
### ✓ Add category



### ✓ Delete category

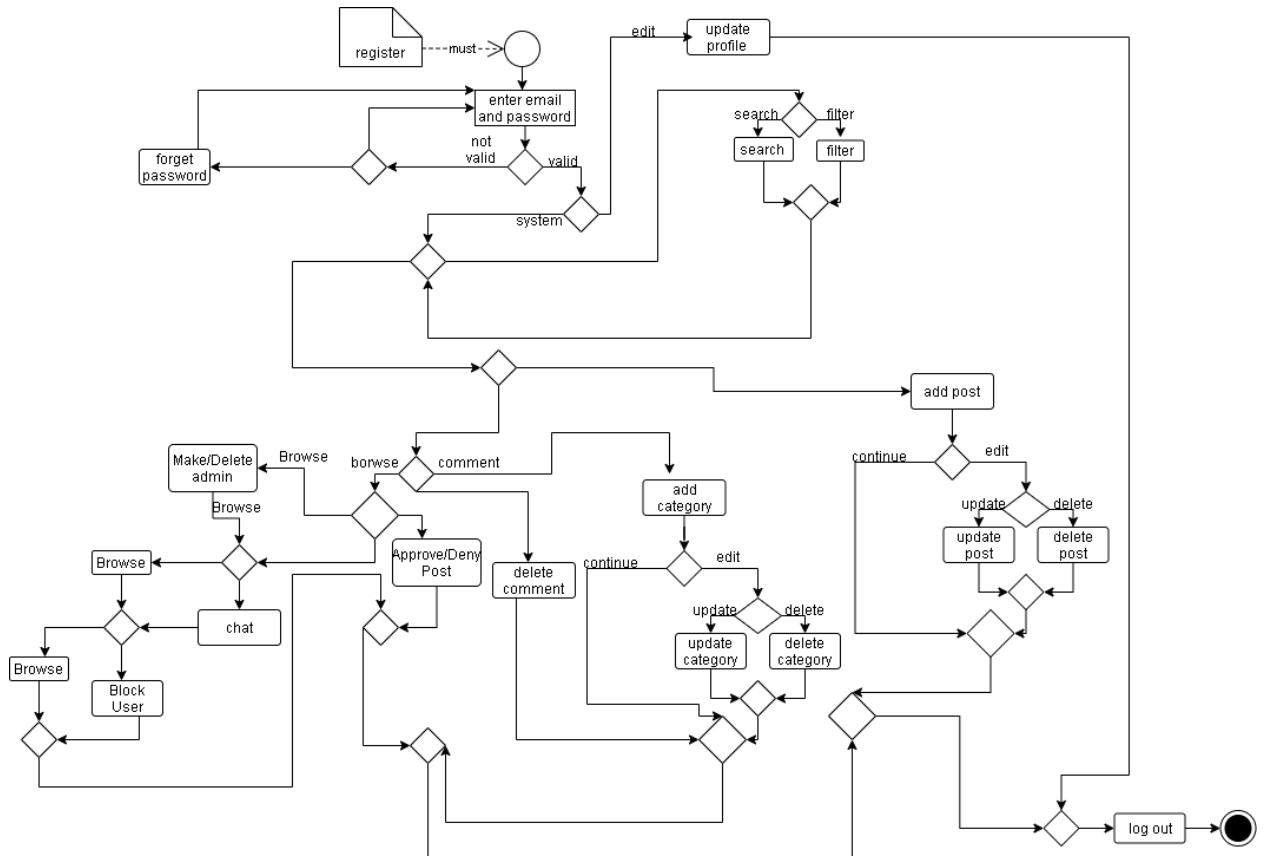


### ✓ Update category

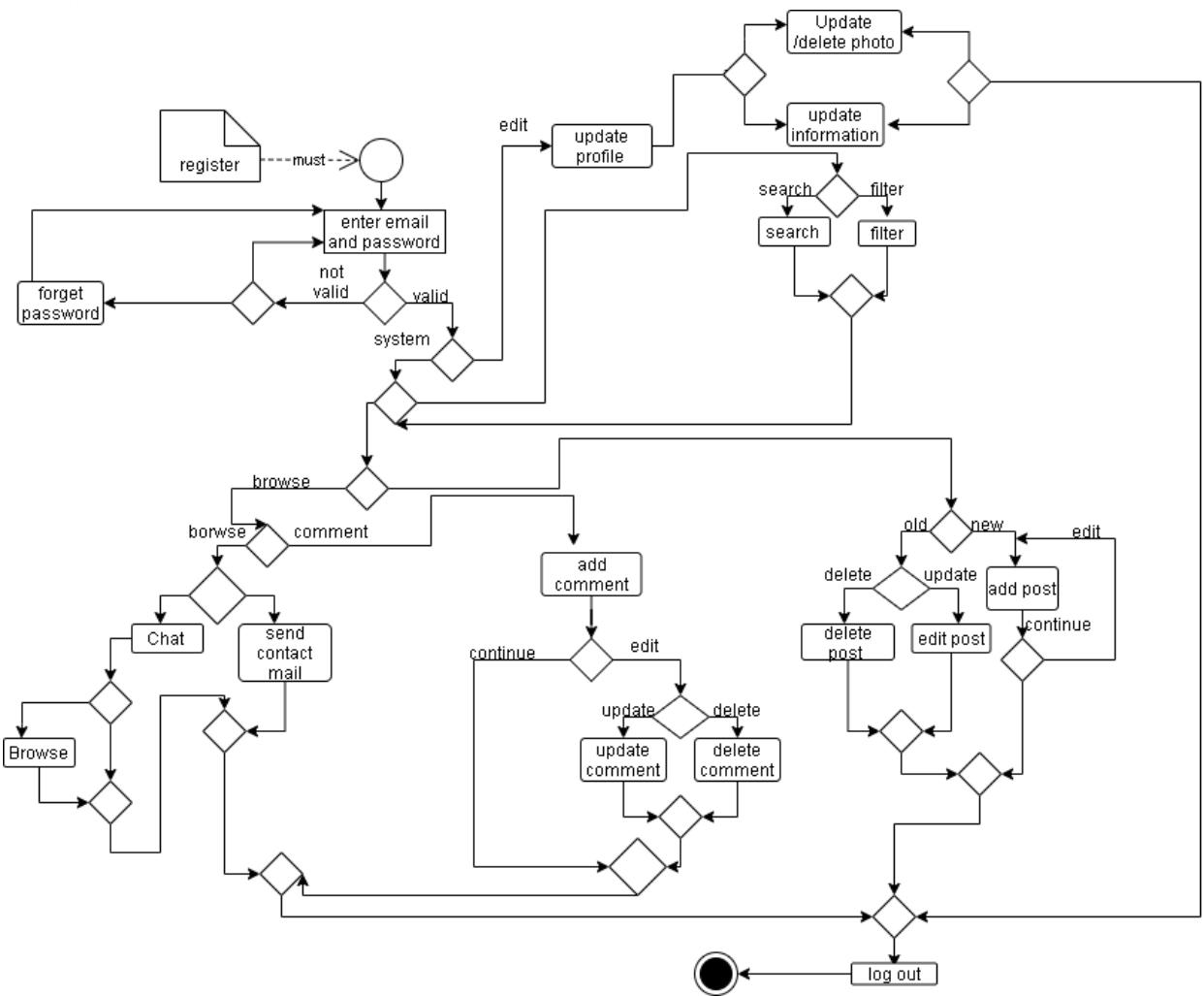


- 3.5.5 Activity diagrams

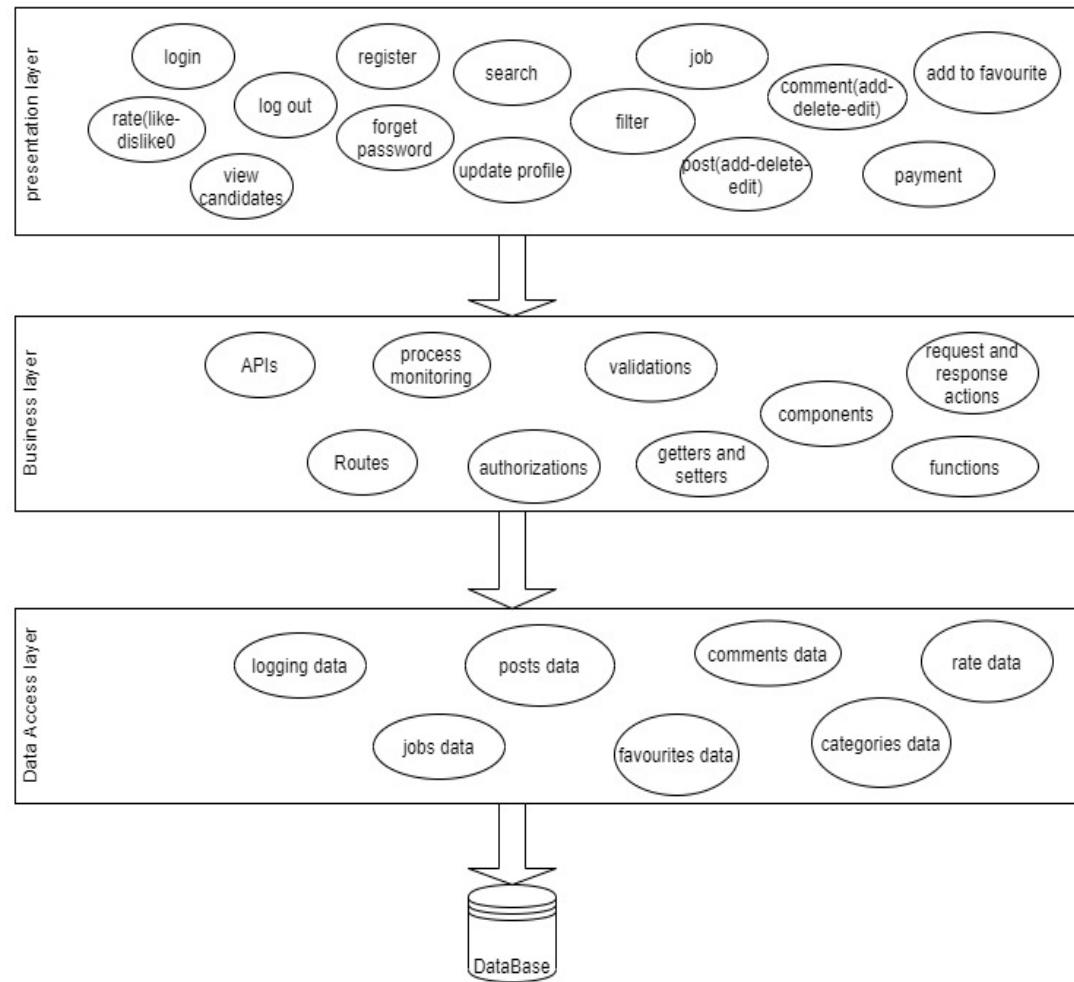
- ✓ Admin



- ✓ User



- o 3.5.6 System architecture



## 4. Results and Discussions

Results and Discussions chapter should provide the reader with the discussions on the collected results and/or achieved final product capabilities as the result of the graduation project. In this chapter we're going to find out the results of the project whether they're achieved or not and also the differences between the desired results and the actual ones.

## 4.1 Results

### ➤ 4.1.1 Expected Results

All users except admin (investors, entrepreneurs, job seeker)

- Different users can sign in, log in and log out.
- Users can edit their profile.
- Users can upload/ delete/ edit posts and write all the requirements they need.
- Users can choose the category they will not upload posts at any random position.
- Users can view other users posts.
- Users can have more than one category such as being investor then entrepreneur.
- Users can request a new password by "forget password".
- Different users can send feedback.
- Different users can make/ delete / edit comments.
- Different users can make a conversation with any other user.
- The website includes an indeed semantic search to make the search more efficient.
- Users can filter what they want to search through.
- The website tracks preferences for each user and recommends posts related to these preferences.
- The website integrates with payment method.
- Users can create a proposal within the website.
- Users can upload proposal without making it within website.
- Users can submit an inquiry through contact us.
- The website provides friendly user interface that allows different users to communicate without issues.
- Users can login with Facebook and google.

- Users can reach their needs by make search by location and this would help in finding needs.
- Users can add job.

#### Admins

- Admin can deny or accept posts.
- Admin can delete comments.
- Admin can block users.
- Admin can make another user an admin for the website.
- Admin can create category.
- Admin can delete/edit category.

#### ➤ 4.1.2 Actual Results

All users except admin (investors, entrepreneurs, job seeker)

- Different users can sign in, log in and log out.
- Users can edit their profile.
- Users can upload/ delete/ edit posts and write all the requirements they need.
- Users can choose the category they will not upload posts at any random position.
- Users can view other users posts.
- Users can have more than one category such as being investor then entrepreneur.
- Users can request a new password by "forget password".
- Different users can send feedback.
- Different users can make/ delete / edit comments.
- Different users can make a conversation with any other user.

- The website includes an indeed semantic search to make the search more efficient.
- Users can filter what they want to search through.
- The website tracks preferences for each user and recommends posts related to these preferences.
- Users can submit an inquiry through contact us.
- The website provides friendly user interface that allows different users to communicate without issues.
- Users can add job.

### Admins

- Admin can deny or accept posts.
- Admin can delete comments.
- Admin can block users.
- Admin can make another user an admin for the website.
- Admin can create category.
- Admin can delete/edit category.

## 4.2 Discussions

- We have managed to get the same expected results except for payment integration, log in with facebook and google, search by location and finally upload and create proposal.
- And also we made some type of testing to check the functionality of our system
- **Functional testing**
  - UNIT TESTING

Testing of individual items (e.g. modules, programs, objects, classes, etc.) As part of the coding phase, in isolation from other development items and the system as a whole.

➤ INTEGRATION TESTING

Testing the interfaces between major (e.g. systems level application modules) and minor (e.g. individual programs or components) items within the application, also, testing the application as whole after committing a file into a branch on GitHub.

➤ SYSTEM TESTING

Testing a system behavior after it is fully integrated, as a whole when development is finished, hence, the system can be tested as complete entity.

○ TEST CASES

- ✓ Model of registration and log in

ID	description	Steps	expected	actual	status
001	User signs up with invalid email format	1. Open web application 2. Chooses Sign up 3. Enter first name 4. Enter last name 5. Enter email with invalid format 6. Enter password 7. Click 'sig up'	An error message telling 'enter a valid email'	As expected	Pass
002	User signs up with email which is already exist	1. Open web application 2. Choose Sign up 3. Enter first name 4. Enter last name 5. Enter valid email 6. Enter password 7. Click 'sig up'  8. Choose Sign up 9. Enter first name 10. Enter last name 11. Enter the same you entered in the last step 12. Enter password 13. Click 'sig up'	An error message telling 'this email is already exists'	As expected	pass

003	User signs up with names informative	1. Open web application 2. Chooses Sign up 3. Enter first name with special characters 4. Enter last name with special characters 5. Enter valid email 6. Enter password 7. Click 'sig up'	An error message 'please enter a valid name'	As expected	Pass
004	User signs up with blank email	1. Open web application 2. Chooses Sign up 3. Enter first name 4. Enter last name 5. Leave email blank 6. Enter password 7. Click 'sig up'	An error message 'please enter an email'	As expected	Pass
005	User signs up with blank password	1. Open web application 2. Chooses Sign up 3. Enter first name 4. Enter last name 5. Enter valid email 6. Leave password blank 7. Click 'sig up'	An error message 'please enter a password'	As expected	pass
006	User signs up with blank names	1. Open web application 2. Chooses Sign up 3. Leave first name blank 4. Leave last name blank 5. Enter valid email 6. Enter password 7. Click 'sig up'	An error message 'please enter names'	As expected	pass
007	User signs up with valid data	1. Open web application 2. Chooses Sign up 3. Enter first name 4. Enter last name 5. Enter valid email 6. Enter valid password 7. Click 'sig up'	Create a register position for this email and send a verification link through email	As expected	pass
008	User signs up with password not in this format of 8 characters	1. Open web application 2. Chooses Sign up 3. Enter first name 4. Enter last name 5. Enter valid email 6. Enter password less than 8 characters 7. Click 'sig up'	An error message telling 'please enter a password in a format of 8 characters at minimum'	As expected	pass
009	User log in with invalid email format	1. Open web application 2. Chooses log in	An error message telling 'enter a valid email'	As expected	Pass

		3. Enter invalid format of email 4. Enter password 5. Click 'log in'			
010	User log in with valid email and incorrect password	1. Open web application 2. Chooses log in 3. Enter valid email 4. Enter incorrect password 5. Click 'log in'	An error message 'please enter a correct password'	As expected	pass
011	User log in with blank email	1. Open web application 2. Chooses log in 3. Leave email blank 4. Enter password 5. Click 'log in'	An error message telling 'enter an email'	As expected	Pass
012	User log in with blank password	1. Open web application 2. Chooses log in 3. Enter valid email 4. Leave password blank 5. Click 'log in'	An error message telling 'enter a password'	As expected	Pass
013	User log in without making his account verification code	1. Open web application 2. Chooses sign in 3. Enter first name 4. Enter last name 5. Enter email with invalid format 6. Enter password 7. Click 'sig up' 8. Choose log in 9. Enter valid email 10. Enter password 11. Click 'log in'	An error message 'please complete the sign up process'	As expected	Pass
014	User log in with valid data	1. Open web application 2. Chooses log in 3. Enter valid email 4. Enter correct password 5. Click 'log in'	The user log in the website successfully	As expected	pass
015	User access his profile by signing in then verify his account and log in	1. Open web application 2. Chooses sign in 3. Enter first name 4. Enter last name 5. Enter email with invalid format 6. Enter password 7. Click 'sig up' 8. Check email 9. Click on verification link 10. Choose log in 11. Enter valid email 12. Enter password 13. Click 'log in'	The registration and log in process complete successfully	As expected	Pass

<b>016</b>	User log in with invalid email format	1. Open web application 2. Chooses log in 3. Enter invalid format of email 4. Enter password 5. Click 'log in'	An error message telling 'enter a valid email'	As expected	Pass
<b>017</b>	User log in with valid email and incorrect password	1. Open web application 2. Chooses log in 3. Enter email 4. Enter incorrect password 5. Click 'log in'	An error message 'please enter a correct password'	As expected	pass
<b>018</b>	User forget his password	Prereq. User loged into web application 1. Open web application 2. Chooses forget password 3. Enter email 4. Check mail 5. Click on verification link 6. Enter new password 7. Confirm new password 5. Click 'confirm'	The password changed successfully	As expected	pass
<b>019</b>	User forget his password and making new without verify email first	Prereq. User loged into web application 1. Open web application 2. Chooses forget password 3. Enter email	The view of related ui of changing password shouldn't present until user check email and verify this,, also presenting to the user a message telling him check your email	As expected	Pass
<b>020</b>	User forget his password and making with invalid format which is 8 characters	Prereq. User loged into web application 1. Open web application 2. Chooses forget password 3. Enter email 4. Check mail 5. Click on verification link 6. Enter new password with characters less than 8 7. Confirm new password 5. Click 'confirm'	An error message 'please enter a valid password'	As expected	pass
<b>021</b>	User change password with incorrect old password	Prereq. User loged into web application 1. Go to your profile 2. Click 'reset password' 3. Enter incorrect old password 4. Click 'next'	An error message 'please enter a correct password'	As expected	pass
<b>022</b>	User change password with correct old password	Prereq. User loged into web application 1. Go to your profile 2. Click 'reset password'	The password has changed successfully	As expected	pass

		3.Enter correct old password 4.Enter new password 5.Confirm new password 6.Click 'confirm'			
023	User change password and making the confirmPass not equal newPassword	Prereq. User loged into web application 1.Go to your profile 2.Click 'reset password' 3.Enter correct old password 4.Enter new password 5.Confirm password by entering different password 6.Click 'confirm'	An error message telling 'two passwords don't match each others'	As expected	pass

Table7-registration and log in test cases

### ✓ Model of profile

ID	description	Steps	expected	actual	status
001	User update profile with social links with invalid format	Prereq. User loged into web application 1.Go to your profile 2.Click 'update profile' 3.Enter incorrect format of social links 4.Click 'submit'	An error message telling 'please enter a valid URL'	As expected	pass
002	User update gender only in the profile	Prereq. User loged into web application 1.Go to your profile 2.Click 'update profile' 3.Enter gender 4.Click 'submit'	The gender updated successfully	As expected	Pass
003	User update his profile photo without updating all the information	Prereq. User loged into web application 1.Go to your profile 2.Click 'update photo' 3.Upload a photo 4.Click 'submit'	The photo Is being uploaded and updated successfully	As expected	pass
004	User update all the information in one step	Prereq. User loged into web application 1.Go to your profile 2.Click 'update profile' 3.Enter all the new values for the data in your profile 4.Click 'submit'	The data are updated successfully	As expected	pass

Table8-Profile test cases

✓ Model of post

ID	description	Steps	expected	actual	status
001	User add post with blank description	Prereq. User loged into web application 1.Click on 'create post' 2.Enter a valid name 3.Leave description blank 4.Choose category 5.Click 'share post'	An error message 'please enter a description'	As expected	pass
002	User add post with blank name	Prereq. User loged into web application 1.Click on 'create post' 2.Leave name blank 3.Enter description 4.Choose category 5.Click 'share post'	An error message 'please enter a name'	As expected	pass
003	User add post without choosing category	Prereq. User loged into web application 1.Click on 'create post' 2.Enter a valid name 3.Enter description 4.Click 'share post'	An error message 'please choose a category'	As expected	pass
004	User add post with valid data	Prereq. User loged into web application 1.Click on 'create post' 2.Enter a valid name 3.Enter description 4.Choose category 5.Click 'share post'	The post is added successfully	As expected	pass
005	User deletes his post	Prereq. User loged into web application 1.Click on your profile 2.Click on 'all posts' 3.Choose the post you want 4.Click delete 5.Click 'confirm'	The post is deleted successfully	As expected	Pass
006	User updates his posts	Prereq. User loged into web application 1.Click on your profile 2.Click on 'all posts' 3.Choose the post you want 4.Click update 5.Update the information you want 6.Click 'confirm'	The post is updated successfully	As expected	Pass

Table9-Post test cases

✓ Model of admin and category

ID	description	Steps	expected	actual	status
001	Admin get all posts	Prereq. admin loged into web application 1.Click on all posts	List of all posts displays	As expected	pass
002	User get all users	Prereq. admin loged into web application 1.Click on all users	List of all users displays	As expected	pass
003	Admin get specific user	Prereq. admin loged into web application 1.Click on get user 2.Enter specific id	The user is displayed successfully	As expected	pass
004	Approving posts	Prereq. admin loged into web application 1.Click on all posts 2.Choose specific post 3.Enter accept	The post is accepted successfully and appears in the category to all users	As expected	pass
005	Denying post	Prereq. admin loged into web application 1.Click on all posts 2.Choose specific post 3.Deny accept	The post is denied and removed	As expected	pass
006	Admin adds category with blank name	Prereq. admin loged into web application 1.Click on create category 2.Leave name blank 3.Click 'craete'	An error message 'please enter a name'	As expected	pass
007	Admin add category with valid data	Prereq. admin loged into web application 1.Click on create category 2.Enter a valid name 3.Click 'craete'	The category is added successfully	As expected	pass
008	Admin makes an existing user also admin	Prereq. admin loged into web application 1.Click on make admin 2.Enter a valid id 3.Click 'make admin'	The user becomes an admin	As expected	pass

<b>009</b>	Admin makes an unexisting user also admin	Prereq. admin loged into web application 1.Click on make admin 2.Enter an unexisting id 3.Click 'make admin'	An error message telling 'please enter a valid ID'	As expected	pass
------------	---	---	--	-------------	------

Table10-Admin and category test cases

✓ Model of comment

ID	description	Steps	expected	actual	status
<b>001</b>	User add a comment	Prereq. admin loged into web application 1.Click on the post wanted 2.Click on 'comment' 3.Enter a description 4.enter 'submit'	The comment added successfully	As expected	pass
<b>002</b>	User update comment	Prereq. admin loged into web application 1.Click on your profile 2.Choose all comments 3.Choose comment 4.Click on Update comment 5.Enter a description 6.Enter 'submit'	Comment updated successfully	As expected	pass
<b>003</b>	User deletes commnet	Prereq. admin loged into web application 1.Click on your profile 2.Choose all comments 3.Choose comment 4.Click on delete comment 5.Enter 'submit'	The user is deleted successfully	As expected	pass
<b>004</b>	The number of comments if the post is deleted	Prereq. admin loged into web application 1.Click on your profile 2.Choose any post 3.make three comments and these will be the only comments you made in the website 4.Click delete post 5.Go to your profile	The number should be zero	The number still 3 comments	fail

		6.Check the number of comments you made			
--	--	---	--	--	--

Table11-Comment test cases

## 5. Conclusions and Recommendations

The Conclusion and Recommendation chapter conclude the report by stating the task, the difficulties faced, experiences gained, results achieved and final thoughts on the project. Then Recommendations for future work and project enhancement should be included.

## ➤ 5.1 Conclusion

In the end, we have tried to do our best to participate and assist in the advancement of the technological community and seek to work on what serves the people in society and reduce exploitation and increase the excellent service with limited resources.

Finally, populating Start-ups in our society will be more convenient if our web site application is used by targeted users. We must help the society in achieving this to make inventors choose the right start-up to be supported, and job seekers can find a suitable job without wasting time or effort.

We should continue to try to help people make the best choices based on their customized reading without any kind of losing data or non-organized or old-fashioned archives.

We have tried to do this with our limited resources, and we are sure that we will expand this website and improve its performance and popularity if given the right resources.

Finally, realizing the fact that “a few clicks” can do a lot to a society, will definitely make us more aware of the technology role and become better versions of ourselves.

Although we faced some difficulties such as:

- Learning new powerful technologies & tools such as machine learning.
- Some problems & errors and fixing bugs while implementation.
- Finding free materials and tutorials for new technologies and sometimes it doesn't exist.
- Finding people who work in the field and meet them to give us more information we need and help us.

- The project enhancement, it's difficult to reuse or generalize usage of the functions in different aspects of code.
- Time management and deadlines.

But however how many problems we faced we have gained many experiences and we enhance ourselves by gaining good skills such as:

- Self-Study skills.
- Time Management skills.
- Planning and Analysis skills
- Teamwork and Project Management skills.
- Learning new technologies.
- Learning new database such as MongoDB and using Mongo Atlas Cloud.
- Learn and practice on powerful programming language such as javascript.

## ➤ 5.2 Recommendations For Future Work

Using more powerful and more durable equipment and the existence of the necessary time and human capabilities of software engineers, software developers, and security engineers, as well as the presence of test engineers and the existence of a budget appropriate for the work we can get more results and add new features.

Because the equipment used in the implementation of this project can be considered as only prototypes as well as we did not have all knowledge about the best tools that needs learning and work on them.

Therefore, the availability of experienced engineers in the field of software, testing, Security, mechatronics, and electronics, as well as providing the appropriate budget not only for the use of better tools and equipment but

also for marketing the product better consider our main recommendations for this helpful project.

Also adding feature to deal with different languages to spread this website to a wide area and far reaching types of people.



*Thank You*