Design Document

for

PAYS

**Version 1.0**

**Prepared by**

**Group #: 08 Group Name: *Techfanatics***

| **NAME**  Chitte Charan Kumar Reddy | **ROLL NUMBER**  210293 | **MAIL ID**  cckreddy21@iitk.ac.in |
| --- | --- | --- |
| Jaya Siva Shankar | 210707 | jssp21@iitk.ac.in |
| Akshay Narayan O | 210097 | akshayn21@iitk.ac.in |
| O Deeven Kumar | 210681 | deevenk21@iitk.ac.in |
| Manjusree Nayak | 210587 | manjusree21@iitk.ac.in |
| N.Hema Sai | 210650 | nhemasai21@iitk.ac.in |
| Mourya Reddy.N | 210662 | mourya21@iitk.ac.in |
| Chetan | 210281 | chetan21@iitk.ac.in |
| P.Lokesh Nayak | 210710 | lokeshn21@iitk.ac.in |
| Sai Sreeja | 210168 | saisreeja21@iitk.ac.in |
| Aryan Thada | 210205 | aryant21@iitk.ac.in |
|  |  |  |
|  |  |  |

| **Course:** | **CS253** |
| --- | --- |
| **Mentor TA:** | ***Archi*** |
| **Date:** | **10/02/2023** |

Content

**Contents ii**

**Revisions ii**

**1 Context Design 1**

1.1 Context model 5

1.2 human Interface Design 5

**2 Architecture design 8**

**3** **object-oriented design 9**

3.1 Use case diagram 9

3.2 Class diagram 14

3.3 sequence diagram 15

3.4 State diagram 17

**4 Project Plan 4**

**5** **Other Details 5**

**Appendix A - Group Log 6**

**Revisions**

| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| Draft Type and Number | Chitte Charan Kumar Reddy  Jaya Siva Shankar  Akshay Narayan O  O Deeven Kumar  Manjusree Nayak  N.Hema Sai  Mourya Reddy N.  Chetan  P.Lokesh Nayak  Sai Sreeja  Aryan Thada | First draft | 10/02/2023 |

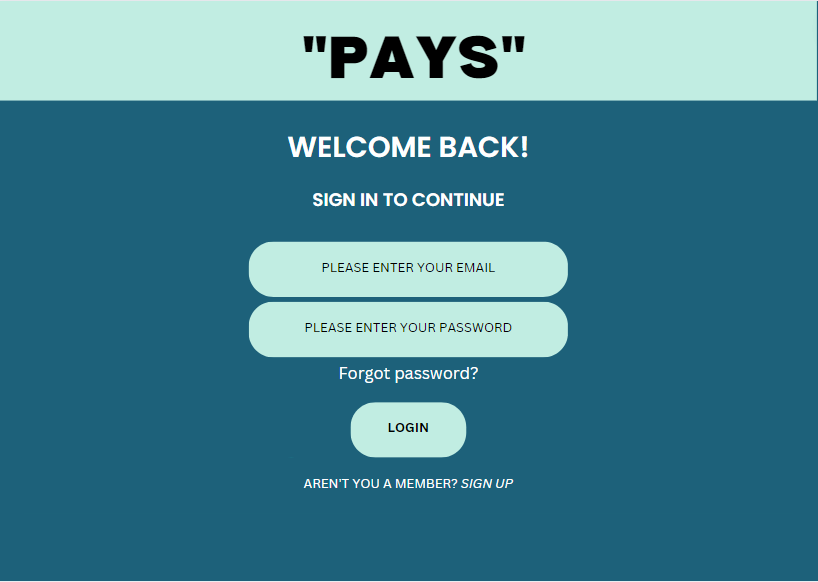
# Context Design

## Context Model

## 

## Human Interface Design

* **LOGIN PAGE :**

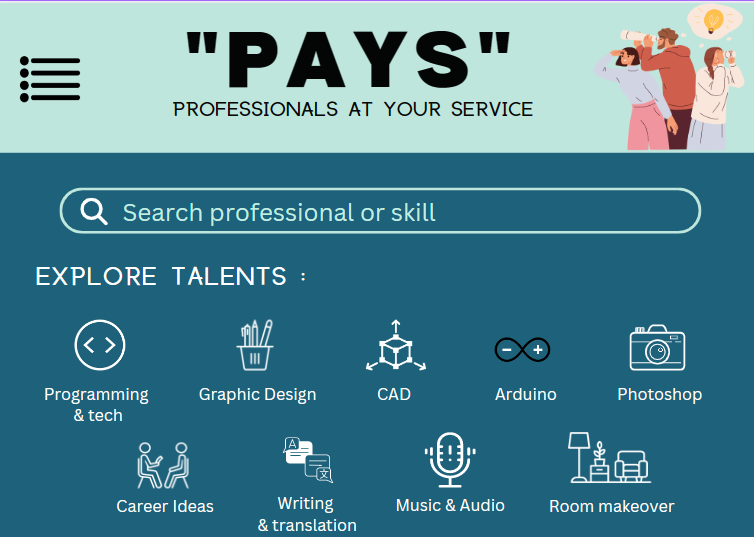


## 

Login page : All users who are registered in “PAYS” shall login using their IITK credentials. If they are not registered, they shall use the sign up option.

USER : Both client and professional.

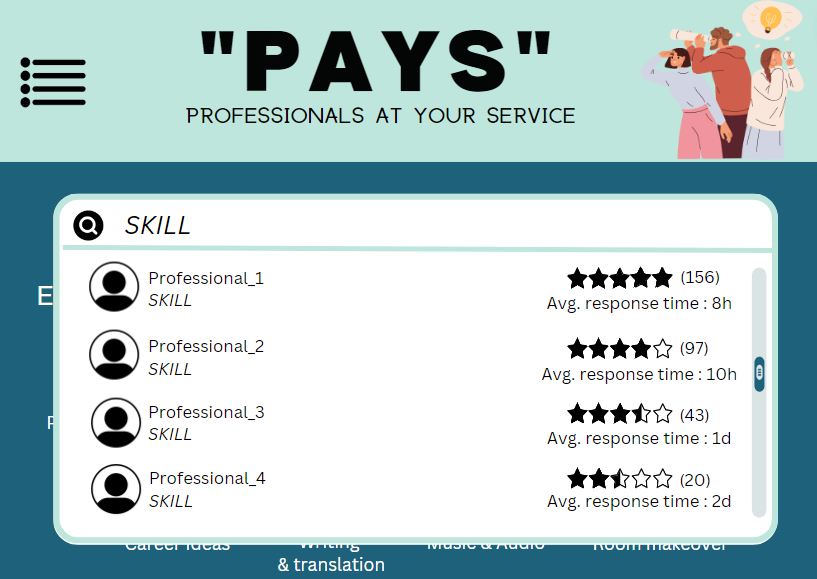
* **HOMEPAGE :**



Home page : Clients can search the professionals by specifying their name or the particular skill.Both users can access the dashboard.

USER- Client and professional.

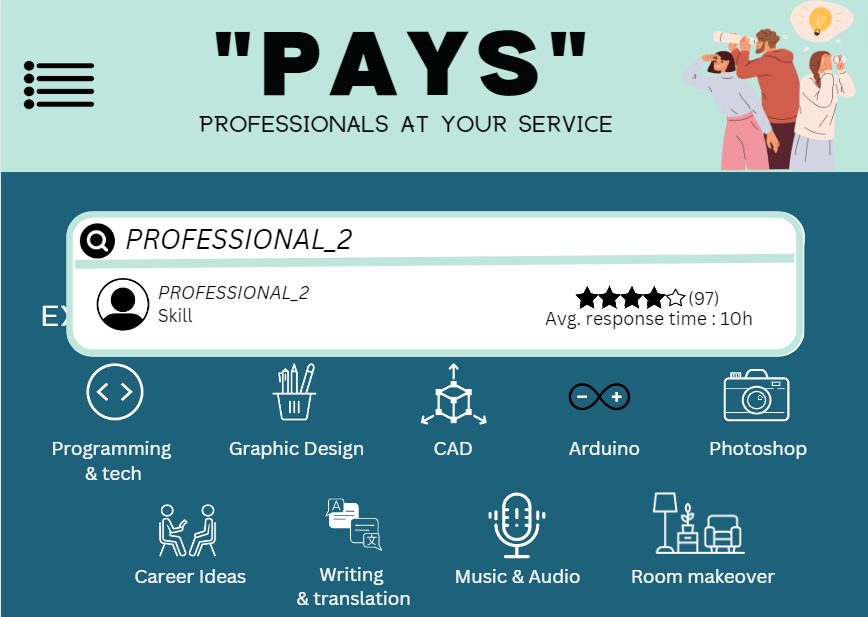
* **SEARCH BY SKILL :**



Search by skill : Users can search various skills and professionals with that skill will be displayed in the decreasing order of their ratings.

USER : Client and professional.

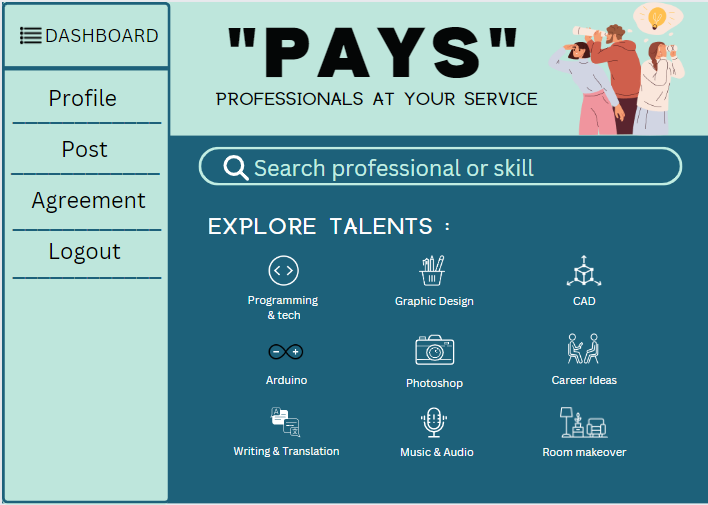
* **SEARCH BY PROFESSIONAL NAME :**



Search by professional name : Users can search a particular professional and that professional will be displayed.

USER : Client and professional.

* **DASHBOARD :**



Dashboard : Users can access their profiles and the post and agreement sections through the dashboard. Users also have an option to logout from their account in the dashboard.

Users : Client and professional.

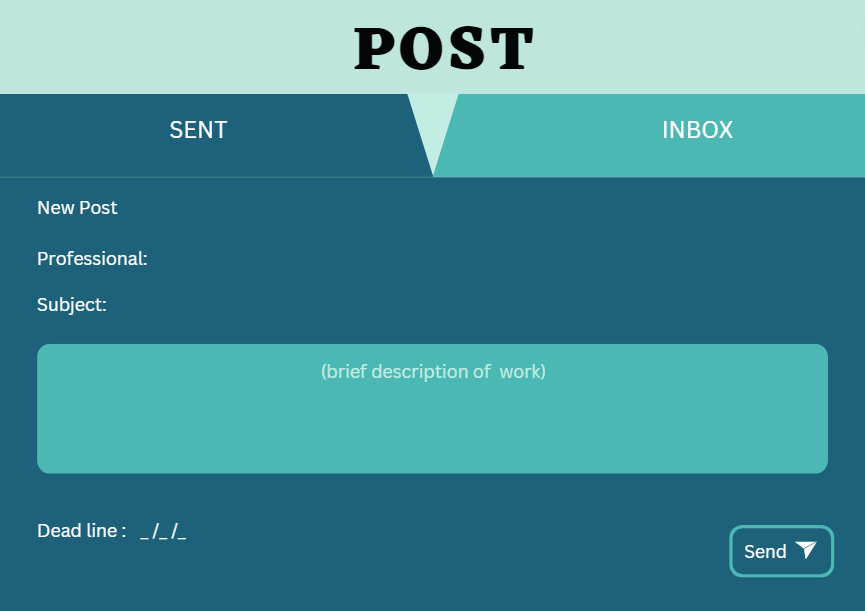
* **PROFILE :**



Profile page : Profile page of the professional showing their skills,rating, response time and their contact details.

USER : Professional.

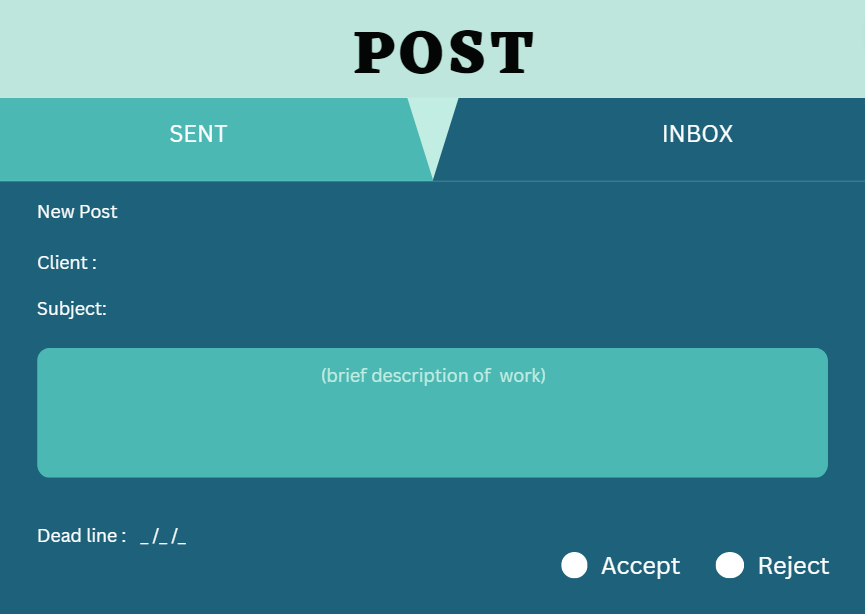
* **POST :**



Sent section in posts : Clients will request professionals by making a post with a work description and tagging the professionals.

USER : Client.

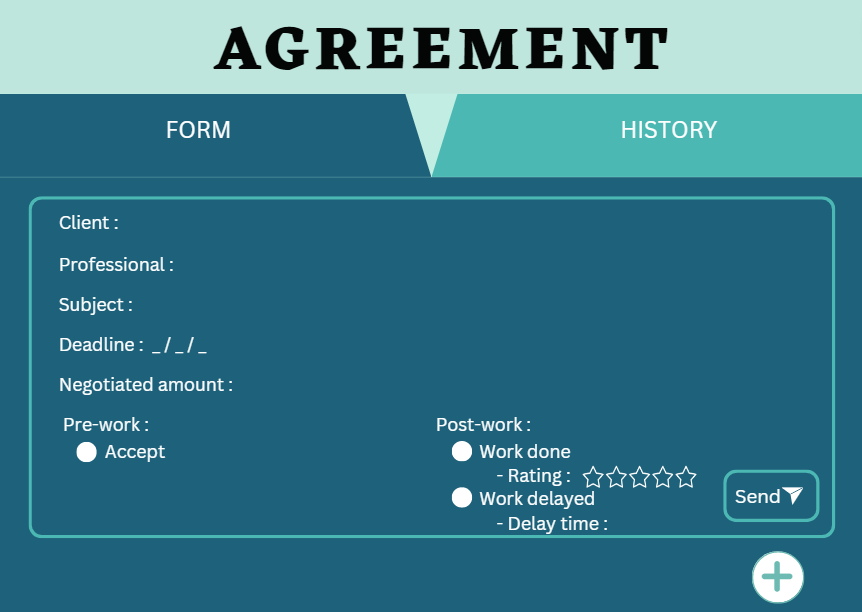
* **POST :**



Inbox section of post : In this section professionals can accept or reject the request sent by the client. The reaction of the professional shall be displayed in the client's inbox section.

User : Client and professional.

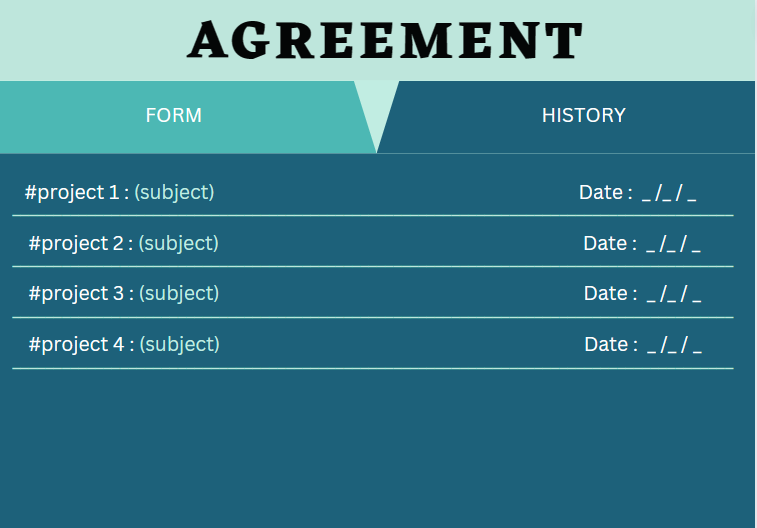
* **AGREEMENT :**



Form section in Agreement : Professional can create an agreement form for the work given by client and can send the agreement to client. Professionals also have the option to let the client know whether the work is done or delayed through the post work options. This agreement shall be displayed in the form section of the client.

Users- Client and Professional.

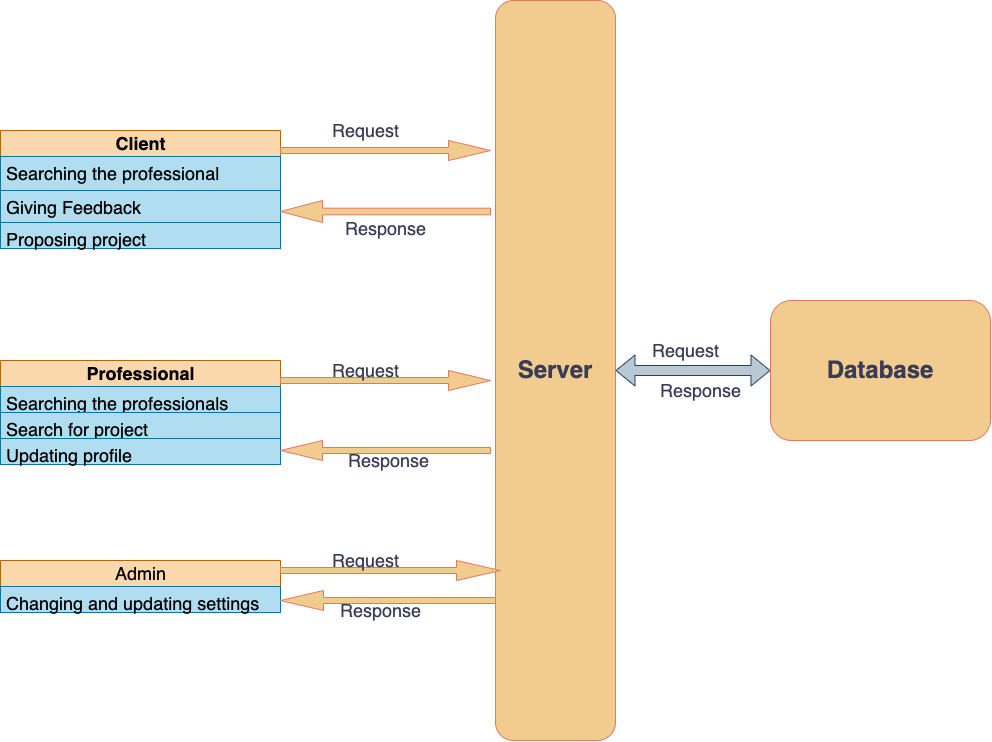
* **AGREEMENT :**



History section in agreement : Users can see their past agreements in the history section.

User : Client and Professional.

# Architecture Design

****

**Why are we using this:** We are using this model since we would be interacting with the data in multiple ways, like checking and receiving requests, storing data of the users, managing agreements and posts. Also, in the near future, one might think of adding some functionalities to it, which makes the Model-View-Controller model the best fit.

**Advantages:** One great advantage of this model is that the data can be represented in various manners, which will prove helpful to us while sorting professionals based on similar talents, or from the same ratings or in some other fashion. Also, it allows the data to change independently; it would be helpful if something runs out of stock.

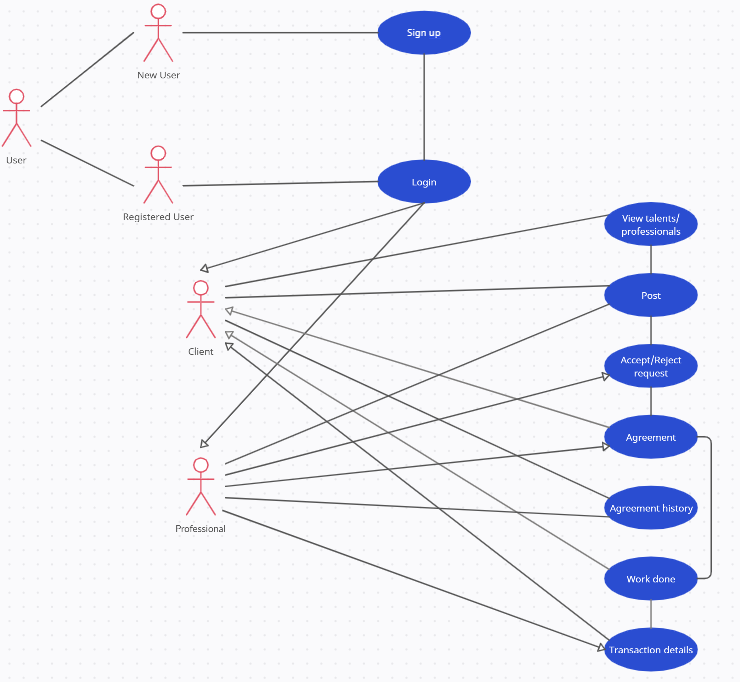
**Disadvantages:** One disadvantage is that we need to write some additional code for full functionality, even if we know that our implementation is simple in the starting stages.

# Object Oriented Design

## Use Case Diagrams

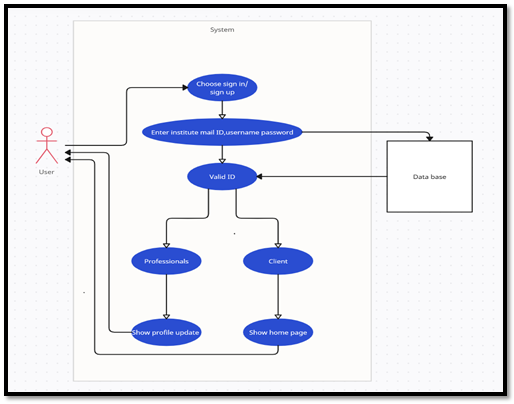
## We have given a consolidate diagram showing different use cases provided:

* Finding professionals for a certain skill.
* user configuration for login and registration.
* Sending a request for professionals through posts and professionals accepting or rejecting the request.
* Agreement form.
* Sending transaction details of the professional to the client. Delivering feedback from the client to the professional.



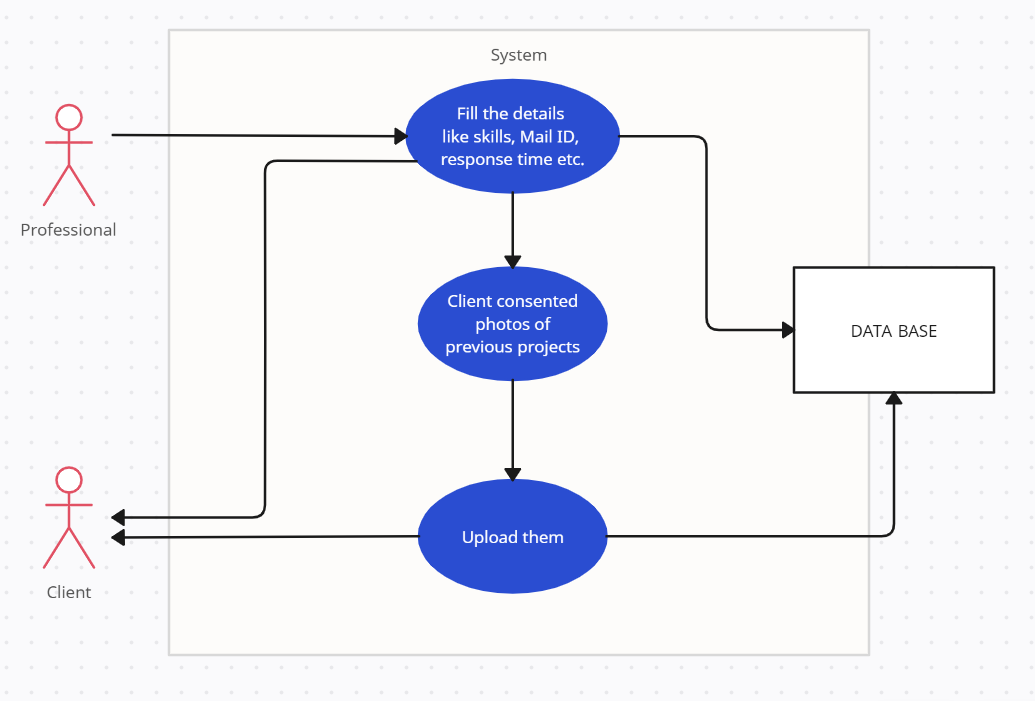
**3.1.1 User Configuration**

To show how the professional/client will sign up & log in



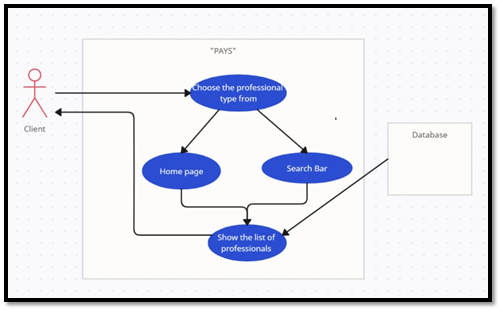
**3.2.2 Updating Profile**

To show the previous work projects of the professional along with their ratings.



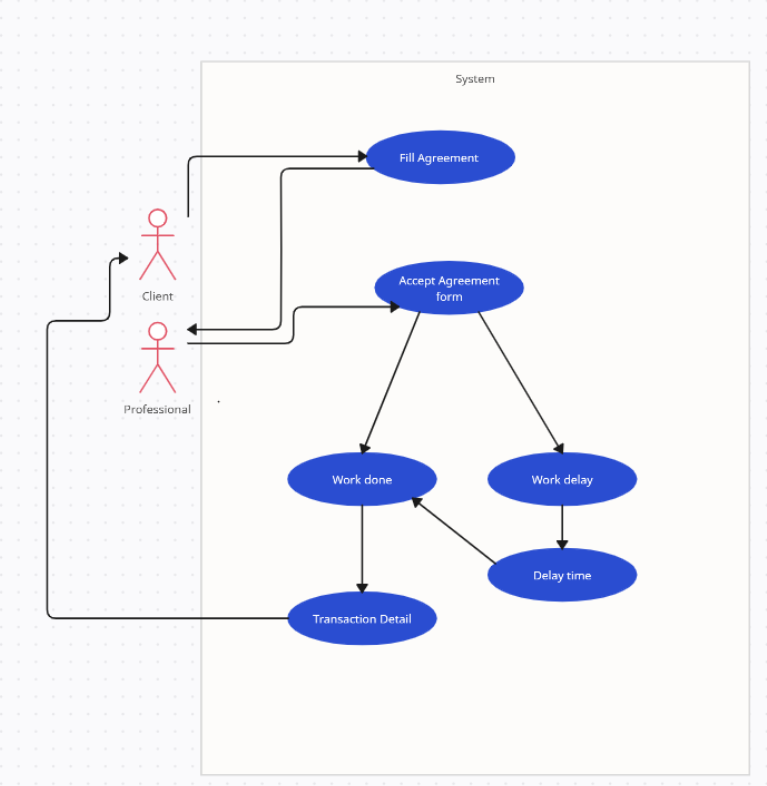
**3.3.3** **Finding the Professional for a Project**

Demonstrate how the clients will find their required professional.



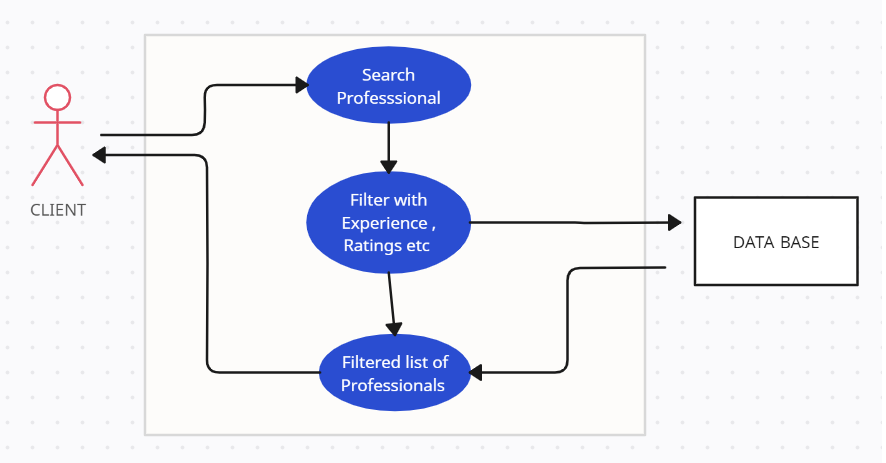
**3.3.4. Making a send request in Post**

Enables the client to send a work request to their desired professional.



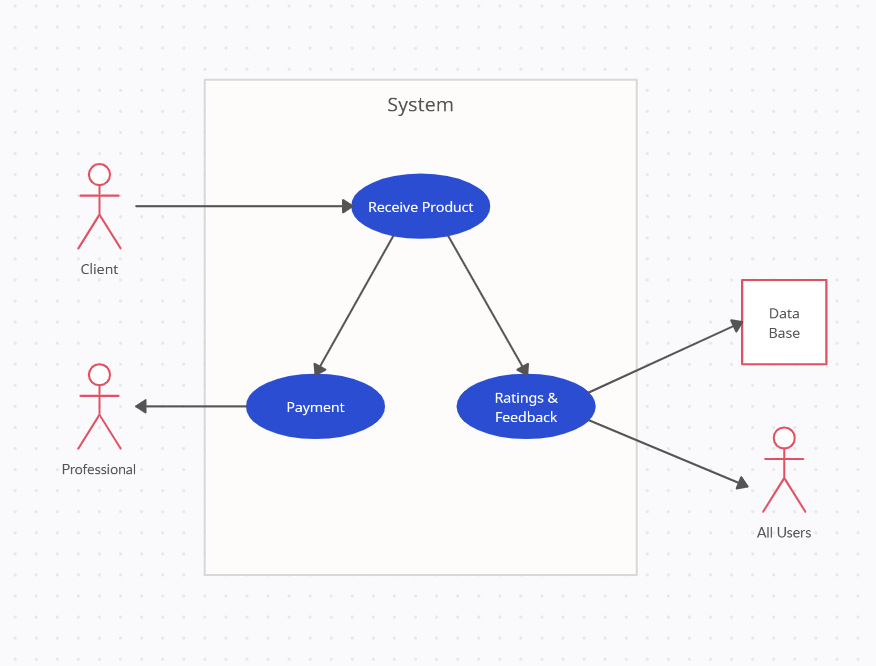
**3.3.6. Filter Search**

To demonstrate how clients can filter their search.

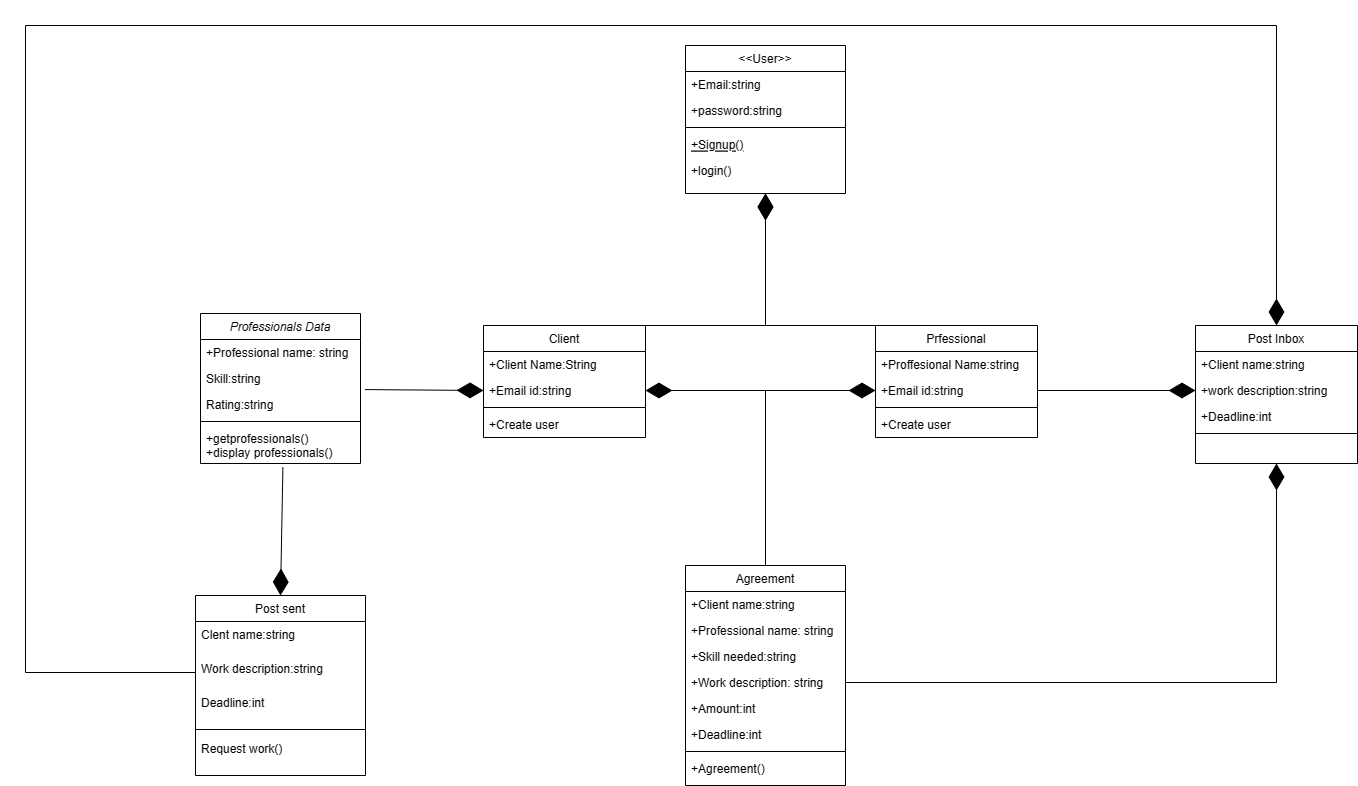


**3.3.7. Feedback**

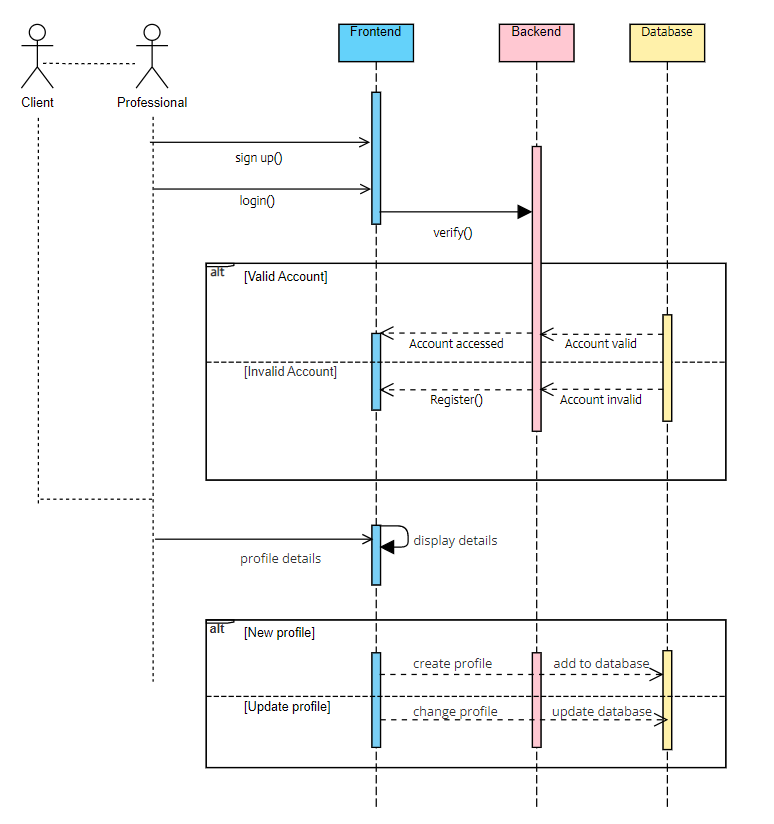
To show the feedback system of the platform.

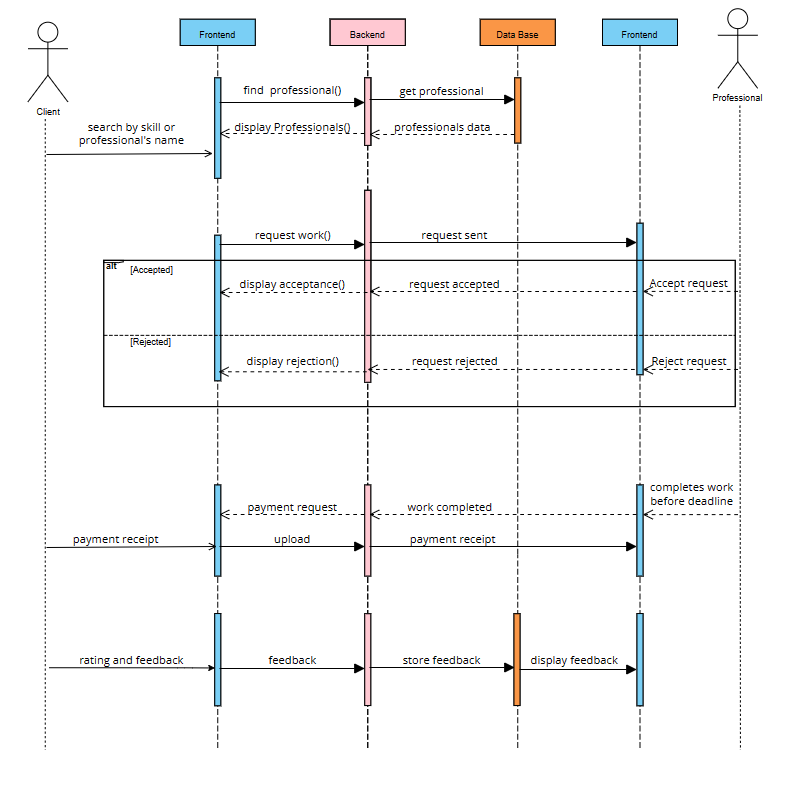


## Class Diagrams

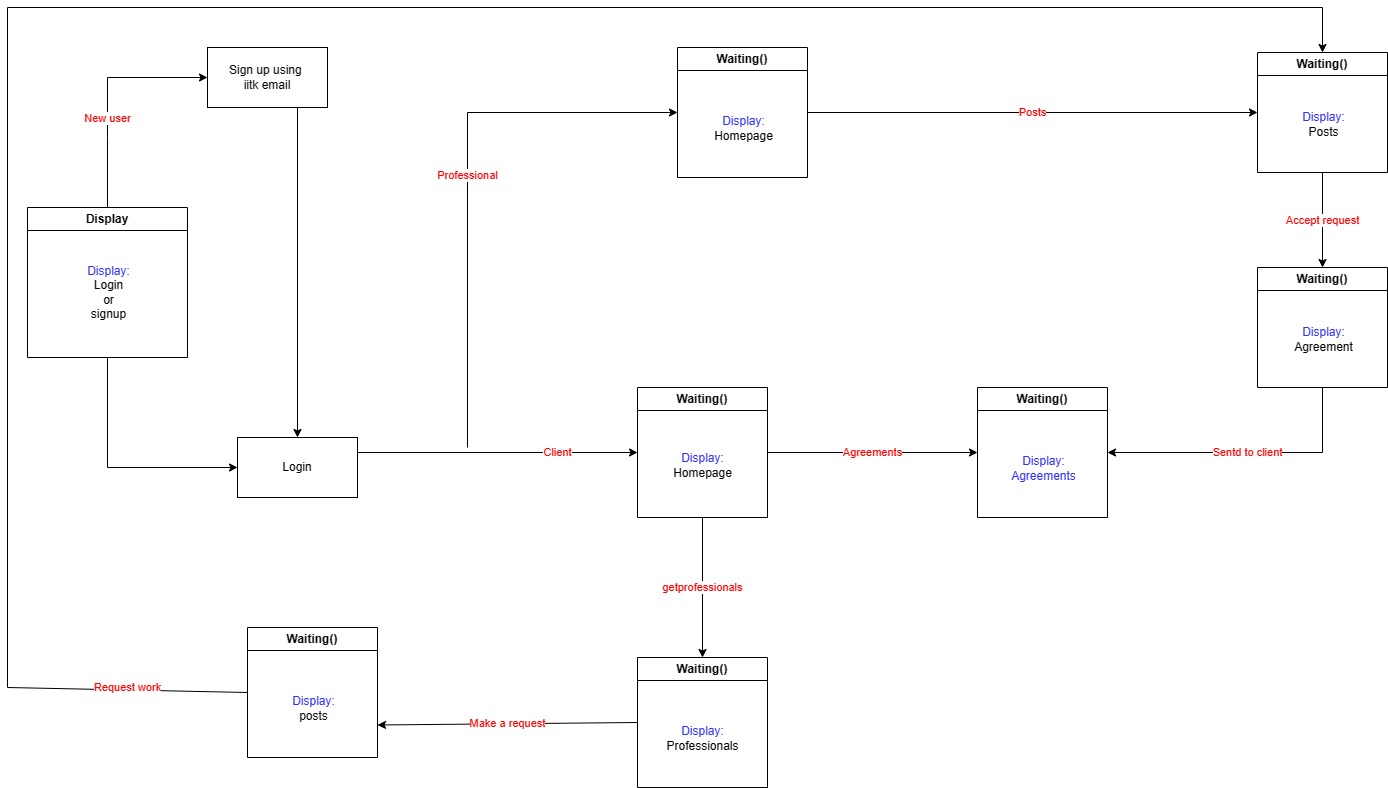


## Sequence Diagrams





## State Diagrams



## 

# Project Plan

**4.1 Communication -**

* The team communicates internally via a Discord Server both via messaging and voice calls.
* The team also has weekly offline meets to discuss the details among ourselves.
* The server is divided into channels based on the scope of work so that discussions are organized based on their context.



**4.2 Code Collaboration -**

* An organization has been set up on GitHub for storing and collaborating on the source code of this project.
* Currently, the organization hosts three repositories (all private access):

○ back-end\_pays

○ front-end\_pays

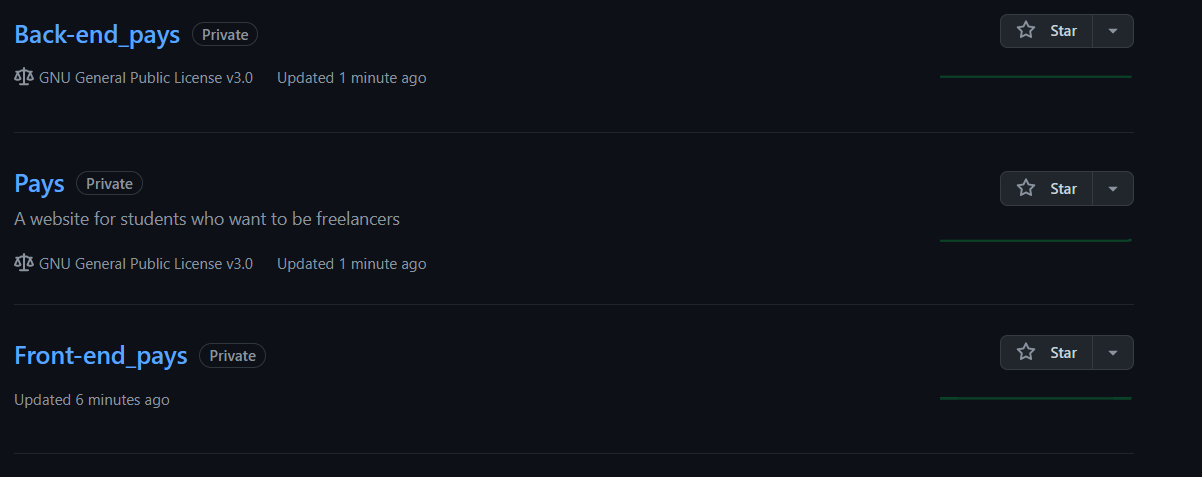
○ pays

* We will maintain two environments (dev and prod) for the implementation.

○ Dev: where development happens

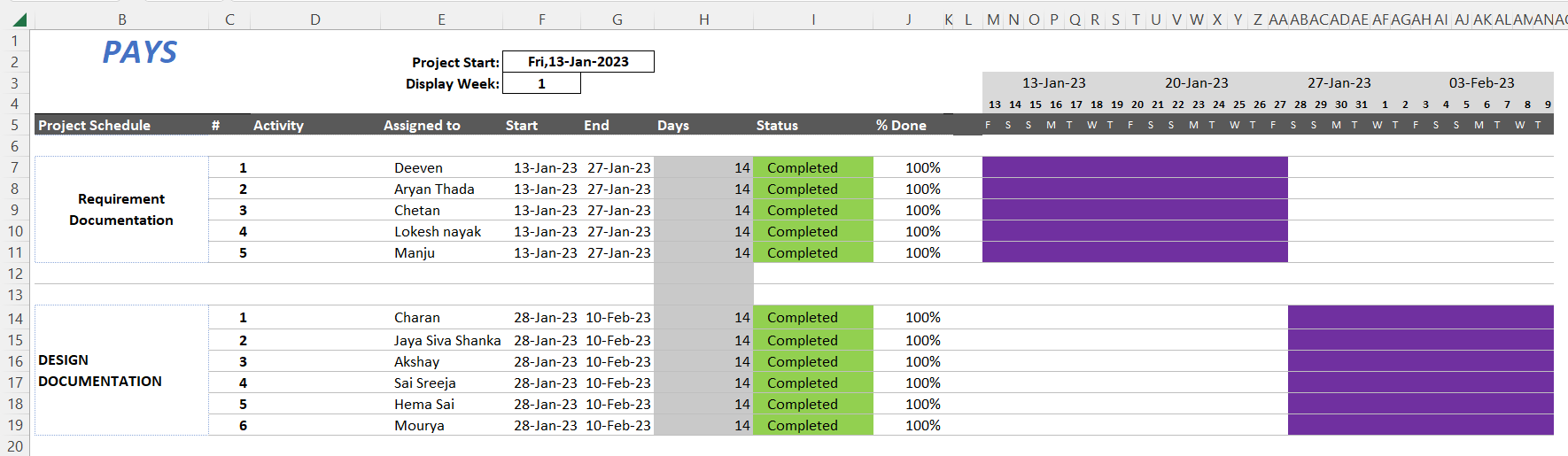
○ Prod: the hosted version for public use (latest release)

* We aim to keep high code coverage of the codebase via unit tests.



**4.2 Project Planning-**

* We are using Excel as of now to implement the Gantt Chart.
* Our Gantt chart looks like this now-



**Appendix A - Group Log**

| **Meeting Minutes** | **Agenda** |
| --- | --- |
| 31st Jan 2023 6 pm - 9 pm.  Offline Meet at RM 4rth Floor. | Started working on a draft for Software Design Template |
| 2nd Feb 2023 12 pm - 1 pm  Offline meet. | Met with TA to discuss ideas and also cleared our doubts and also showed the work pogress. |
| 4rth Feb 2023 2 pm - 4 pm  Offline meet at RM 4rth Floor | Divided the work among various members according to their strengths and preferences. |
| 5th Feb 2023 2 pm - 4 pm  Online meet at Discord | Discussed progress and cleared doubts. |
| 9th Feb 2023 2pm - 8 pm  Offline meet | Finalised and gave finishing touch to the Document. |