LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS5002NI SOFTWARE ENGINEERING

Assessment Weightage & Type

35% Individual Coursework

Year and Semester

2022-23 Spring

Student Name: Himanshu Joshi

London Met ID: 22068759

College ID: NP01CP4A220266

Assignment Due Date: May 7, 2024

Assignment Submission Date: May 7, 2024

Title (Where Required):

Word Count (Where Required): 2684

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents:

Table of Figures:

Table of Tables:

1) Work Breakdown Structure and Gantt Chart:

A work breakdown structure (WBS) is a project management approach that divides larger projects into smaller, more manageable components or jobs. The primary purposes of a work breakdown structure (WBS) are to provide a clear understanding of the project scope, identify all work that needs to be done, and fuel effective project planning and management (Team, 2023).

The WBS of the given scenario is given below:

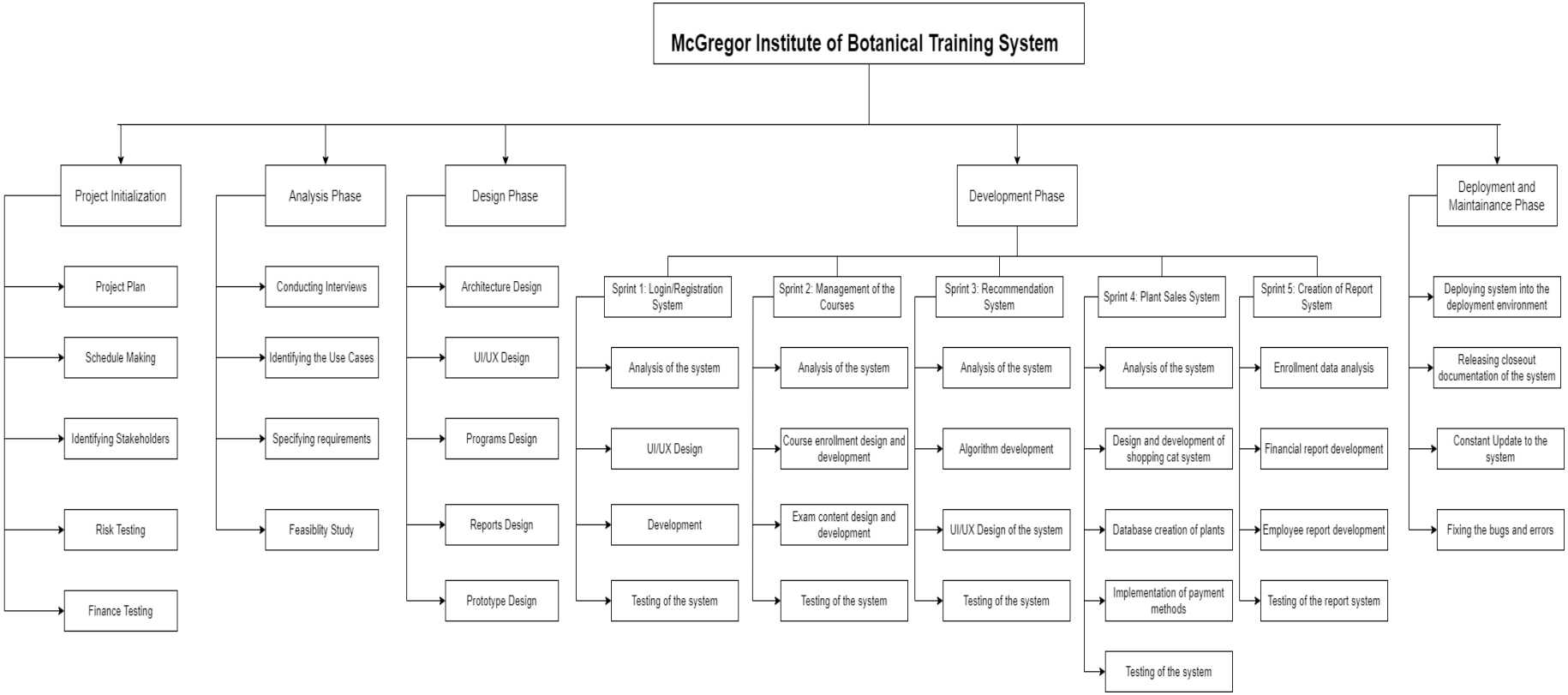

Figure 1: WBS of the system.

The Work Breakdown Structure that is created above, is created by using a scrum methodology. This methodology includes different sections known as sprints, and each sprint is divided into the sub tasks of it. Here, there are overall five major steps or tasks related to the system. Each phase consists of its subtask, and the major phase is the development phase, where it is further divided into five sprints, and those five sprints are also further divided into major sub sprints .Each sprint has the duration of minimum of 2 weeks, and maximum of 4 weeks. So, the scrum methodology helps a lot while creating the work breakdown structure for the given scenario. By following the above work breakdown structure, a software application can be made very efficiently with proper testing and deployment.

A Gantt chart is a project management tool used to plan, schedule, and monitor a project. Gantt charts can help with planning and scheduling, remote work collaboration, resource allocation, and task delegation. A Gantt chart visualizes all information using a horizontal bar graph. The chart allows project managers and team members to quickly evaluate job schedules, dependencies, and progress. Planning all tasks ahead of time and keeping them visible in one place enables teams to deliver on time (Shweta, 2024). From the above work breakdown structure, the Gantt Chart is made accordingly. The scrum methodology has been used to develop the Gantt Chart, where it is divided into multiple sprints, and each sprint has the minimum duration of 2 weeks, and the maximum duration of 4 weeks.

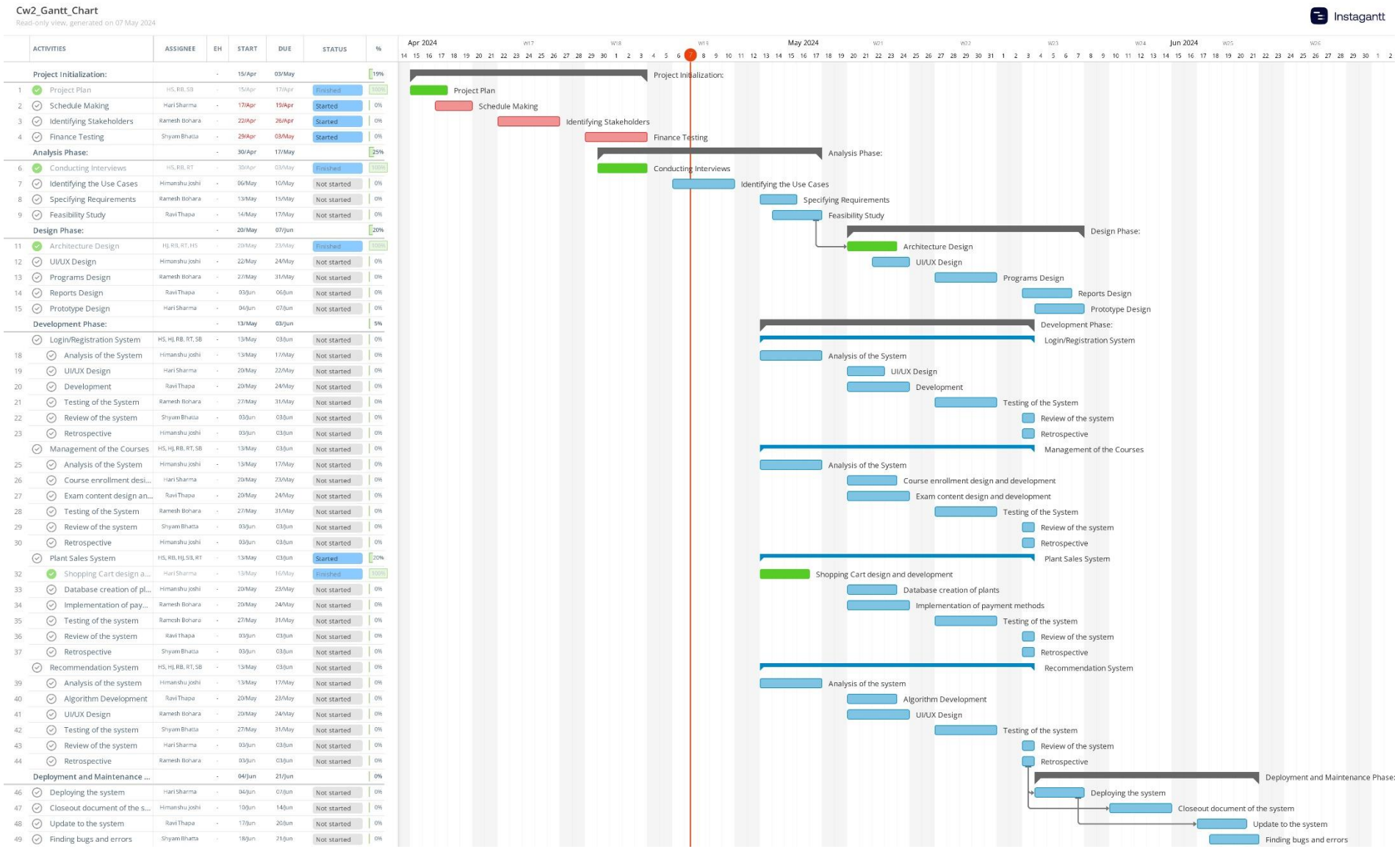

Figure 2: Gantt Chart of the system.

2) Use Case Diagram:

A use case diagram is defined as an important tool in system design as it visually represents how uses interact with a system. It serves as a blueprint for understanding a system's functional requirements from the perspective of its users, facilitating communication among stakeholders, and guiding the development process (geeksforgeeks, 2024).

The use case diagram for the given scenario is given below:

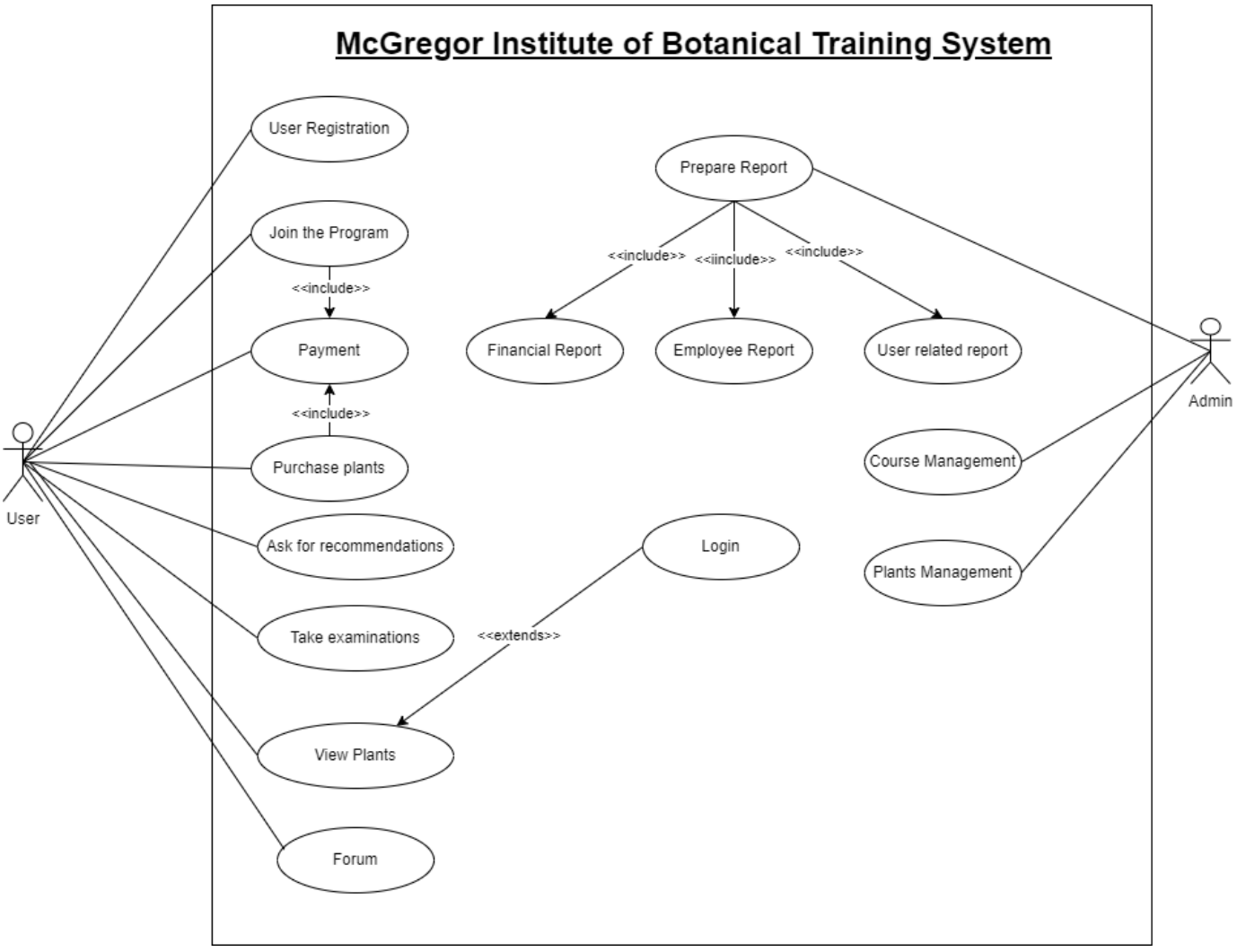

Figure 3: Use case diagram.

3) High Level Use Case:

The high-level use case is a brief, paragraph, or unstructured description of the activity, whose goal is to assist you father related use cases for development in the Elaboration phase by giving you just enough material to get a sense of its complexity (Anon., 2022).

Below is the high-level description of the use cases in the given scenario:

3.1) User Registration:

- **Name:** User Registration
- **Actor(s):** User
- **Description:** A new user who is wishing to use this system or go through it can register himself/herself by providing the appropriate details of them.

3.2) Join the Program:

- **Name:** Join the Program
- **Actor(s):** User
- **Description:** A new user who is wishing to join any program that the system provides, can go through it, for which payment should be done firstly. The program contains different undergraduate, graduate programs offered by McGregor Institute.

3.3) Payment:

- **Name:** Payment
- **Actor(s):** User
- **Description:** A new user who is wishing to buy any plants or to go through any undergraduate or graduate programs, for which payment is the first thing to do for.

3.4) Purchase Plants:

- **Name:** Purchase Plants
- **Actor(s):** User
- **Description:** A user who is wishing to purchase plants, can purchase it by visiting the plants store, but for it payment should be done first.

3.5) Ask for Recommendations:

- **Name:** Ask for Recommendations
- **Actor(s):** User
- **Description:** A new user who is wishing to use ask for recommendations, can have it by asking with experts. Users can locate their site on map, if possible, also provide photos of the soil's state, so that experts can respond with the best plans or the crops to grow in that specific spot.

3.6) Take Examinations:

- **Name:** Take Examinations
- **Actor(s):** User
- **Description:** A user who is wishing to take examination of the programs, can take it, for which they should register and give their appropriate details. They may also give mock tests according to their convenience. They can check results as well.

3.7) Forum:

- **Name:** Forum
- **Actor(s):** User
- **Description:** A user who is wishing to participate in discussion on plants, can share their thoughts through postings and are also able to upvote and remark on other people's views.

3.8) Prepare Report:

- **Name:** Prepare Report
- **Actor(s):** Admin
- **Description:** This is the task that is done by the administrator of the system. They have the facility to prepare the financial report, employees report, and also user related reports.

3.9) Course Management:

- **Name:** Course Management
- **Actor(s):** Admin
- **Description:** This is the task that is done by the administrator. They can see all the courses available and can also add some new if the system provides any new courses.

3.10) Plants Management:

- **Name:** Plants Management
- **Actor(s):** Admin
- **Description:** This us the task that is done by the administrator. They can view all plants that is available, and can also add new plants, if they are available. Also, they can delete any existing plants if they are out of stock.

4) Expanded Use Case:

Expanded Use Case is a comprehensive written depiction of a system's capabilities that offers a methodical description of how system interacts with other systems. It generally consists of two primary flows: an alternate flow that covers exceptions or variations, and the main flow that outlies the procedures to take when everything goes according to plan (Ebrary, 2023).

Some of the expanded use cases of the given scenario are given below:

4.1) Join the Program:

- **Name:** Join the Program
- **Actor(s):** User
- **Purpose:** To join different training programs provided by McGregor Institute.
- **Overview:** A new user who takes interest in joining the programs, initiates his process. He/she does the payment first, and then are eligible to join the programs.
- **Type:** Primary, Essential.
- **Action Steps:**

Actor Action	System Response
1) New student shows the interest in joining McGregor Institute's program.	
2) User logins and selects the option of join the program.	
	3) Then the system provides a registration form asking the student to fill the personal details and credentials.
	4) After filling the registration form, the admin successfully approves the student.
5) After getting approved by the admin, the user goes into the payment process.	
	6) Admin checks the payment method of the user, and if the payment is successful, user becomes eligible to access the program resources.

Table 1: Expanded Use Case of Join the Program.

- **Alternative Course of Action:**

Line-3:

If there is the service of giving self-enrollment service, then after the user fills out the registration form, there is no need of any approval from the administrator. The system instantly provides access to program resources to the user after form submission.

Line-5:

If there is the service of giving partial access to the program resources to the user after the registration, then the scenario becomes different. Then the users will do the payment according to their satisfaction with the resources. If they are satisfied, then they will do the payment, otherwise no more action would be required.

4.2) Purchase Plants:

- **Name:** Purchase Plants
- **Actor(s):** User
- **Purpose:** To purchase different plants from the shopping section.
- **Overview:** A new user who takes interest in purchasing the plants, can purchase it by browsing the catalog of available plants in the system.
- **Type:** Primary, Essential.
- **Action Steps:**

Actor Action	System Response
1) New user shows the interest in purchasing the plants from the system.	
2) User logins to his/her respective account.	
	3) The system confirms the successful login of the user.
4) Then the user goes in the shopping cart section of plants and goes through different plants that are available.	
5) After visiting into different plants, the user selects the plants that he/she wants to purchase.	
6) After selecting the plants, the user goes in the payment process, and enters the required payment details.	.
	7) The system then checks the payment method and successfully confirms the payment.
	8) Then the system provides the necessary plant delivery details to the respective user.

Table 2: Expanded Use Case of Purchase Plants.

- **Alternative Course of Action:**

Line-7:

If there arise any payment issues during the payment of the plants purchase, the system would automatically alert/notify the users. The payment issue might be due to the reasons like out of balance, expiration of the card, etc. The user then gets to select a different option for payment.

5) Collaboration Diagram:

The collaboration diagram represents the interaction of the objects to carry out the behavior of a certain use case or a portion of a use case. The collaboration diagram is used by the designers to specify and make clear the functions of the objects that carry out a specific sequence of events in a use case (geeksfoegeeks, 2023).

From the use case model, a specific task is chosen, and the collaboration diagram is made from it. In this case, the “Purchase Plants” activity was chosen, and the tasks were mentioned that how plants are purchased by the user, what users had to do for it. Different messages, which were functions were communicated from one object to another, and finally the collaboration diagram was made from it.

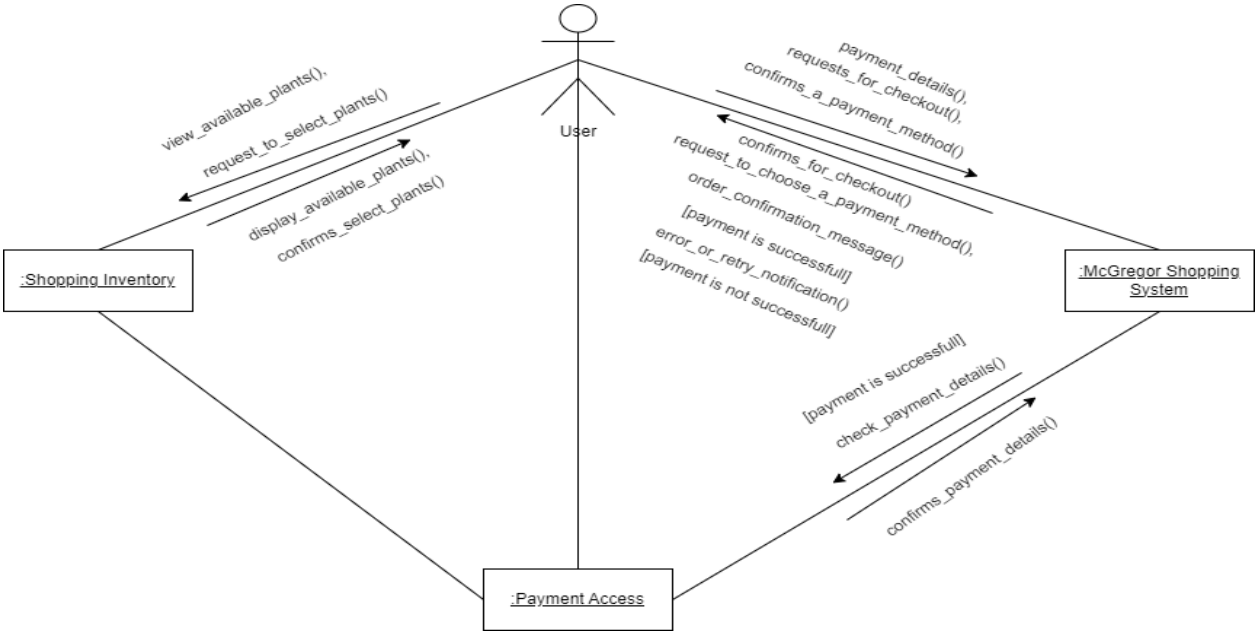

Figure 4: Collaboration Diagram.

6) Sequence Diagram:

In software engineering and systems design, a sequence diagram is a form of UML (Unified Modeling Language) diagram that is used to illustrate the interactions and communication between distinct items or components inside a system (VanZandt, 2023). For creating the sequence diagram, Purchase Plants activity was chosen, and firstly objects we found out. Different messages and information are passed from objects to objects, which defines how the purchase plants activity is done by the user and, also defines the conditions that should be satisfied for it.

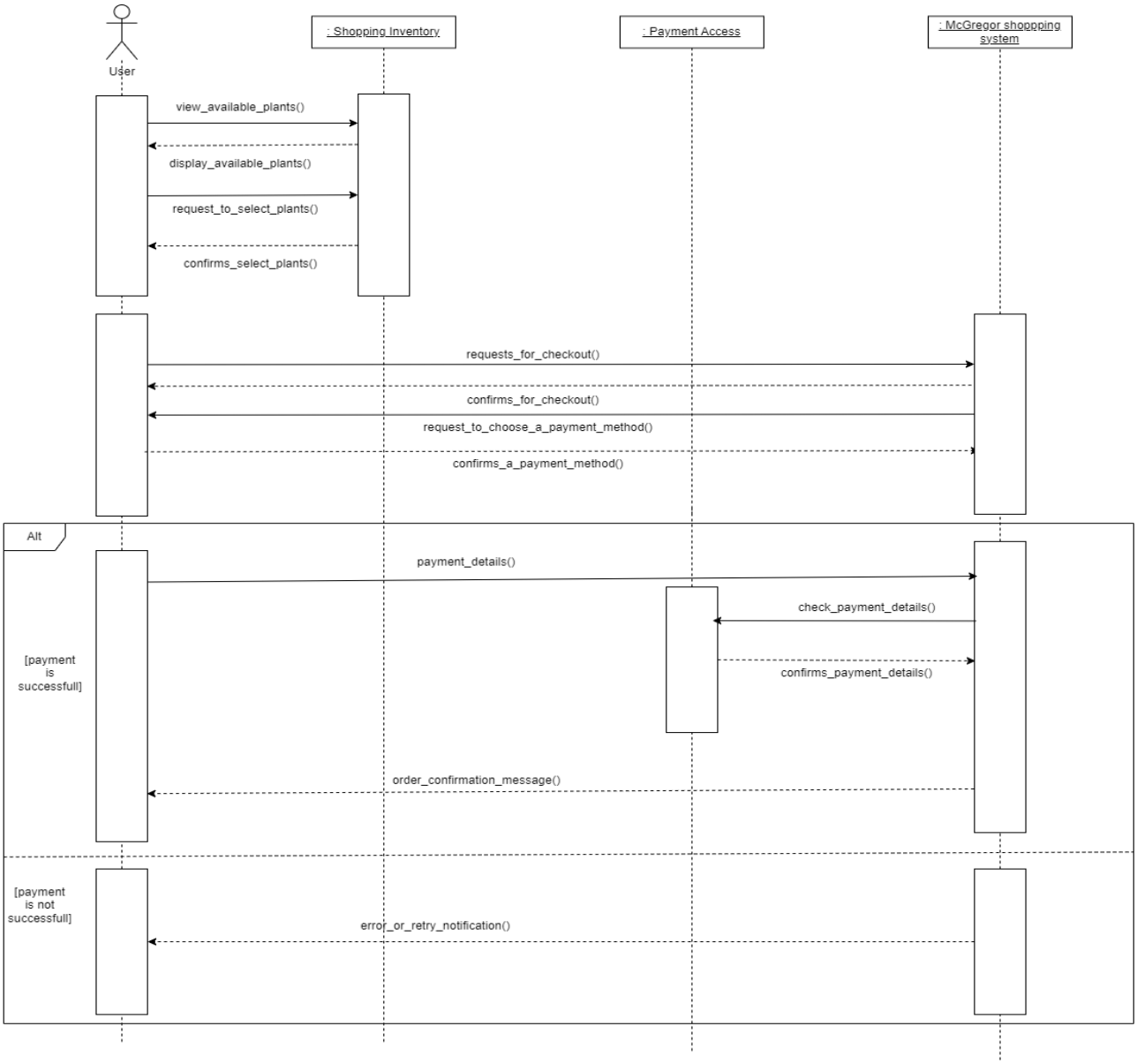

Figure 5: Sequence Diagram.

7) Class Diagram:

A class diagram is a form of UML (Unified Modeling Language) diagram, that shows the organization and connections between classes in an object-oriented software system at a high level. Software engineers and other stakeholders can better grasp the system's architecture and design with the aid of this visual representation (Pandey, 2024).

According to the given scenario, the McGregor Institute system does a variety of tasks which eventually makes it a better system. There are different activities, which are then made classes, and each class contains its own attributes, and own methods as well. When we identify every class that are required for it, we establish different relationships between them and eventually, the combination of those classes work as a single unit of McGregor Institute. The relations can have one-to-one, one-to-many, many-to-many, many-to-one, inheritance, aggregation, composition. So, in this way, we establish the class diagram.

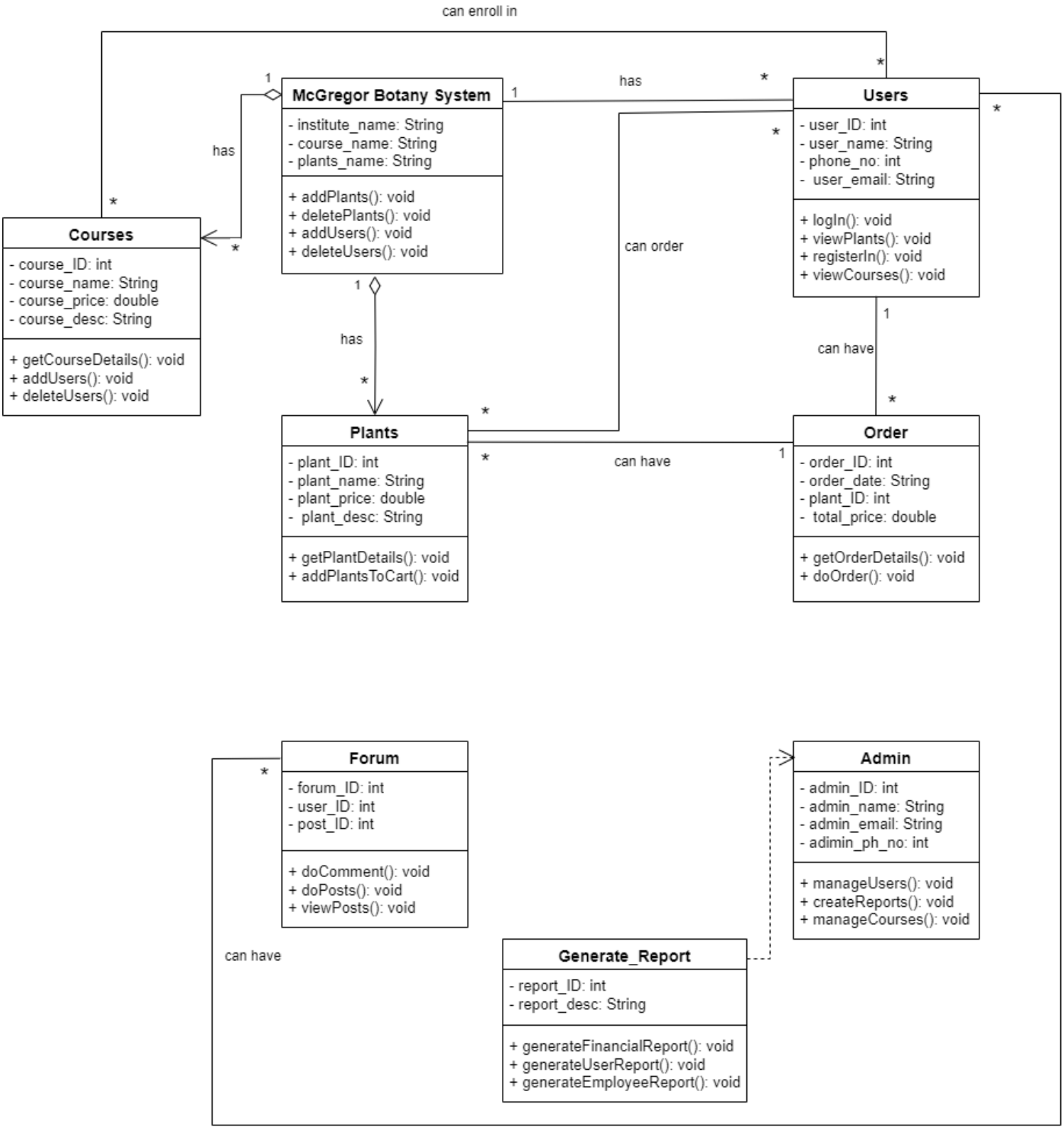

Figure 6: Class diagram for the system.

8) Further Development:

Now, all the diagrams which were essential for the initialization of the project is done. Now, to move for the further development process of this project, there are several factors that we need to be concerned about. The further development requires a methodology, which is essential for the further planning and for the reliability of the project, we need to choose a certain methodology for it. I have chosen scrum methodology for it as this methodology provides me the essential sprints that could be required for the sustainable planning for the project. This methodology is easy to apply and easy to understand as well.

8.1) Architectural Choice:

For all the features that are to be included in the McGregor Institute system, an architecture which is scalable and modular would be the best option. If the system needs any further development or updates, it would be very good choice. For an architectural choice for the system, microservices design choice would be a very effective choice. For every functionality like management of users, courses, recommendations, plant shopping cart, etc., this microservices design could be very effective choice.

8.2) Design Pattern:

Design Pattern could be another factor that is to be considered for the sustainable development of the project. We can use various design patterns for it, but the MVC (Model-View-Controller) design pattern would be a very good choice for it, and advantageous as well. In this pattern, the program can we divided into three different parts. In Model, we can design the data and business logic of the application. In View, we can design the user interfaces that would be easy for the users to interact with, and in Controller, we can application logic, data transformation, etc. So, the design pattern MVC would be very effective for the further development process.

8.3) Development Plan:

For the proper development plan, now there would be proper coding of the system. For it, React could be used for the frontend responsive user interface. Likewise, the frameworks like Django could be used for developing its web application. Git can be used for the management of the codebase. Finally, MySQL language could be used for the database preparation of it. So, these are the major steps and strategies for the further project development of the system.

The priority orders of the features that is planned are given below:

1) User Login and Registration.
2) Shopping Cart.
3) Courses and Examinations.
4) Payment.
5) Report Preparation.
6) Forum.
7) Aks for Recommendations.

8.4) Testing Plan:

Now, during the development process, the testing is also necessary for the system. So, mainly there are three types of testing which could be effective for the system throughout the development cycle. While developing every single module of the systems, the unit tests should be done for the see the result of its functionality. Since, modules are also interconnected with each other, so the integration testing would help to see the result whether the modules are properly working together or not. The stakeholders should also be involved in the testing process as per user requirements since the feedback and advice of them could help in the improvement of the system. So, this plan would be effective in terms of the testing plan.

8.5) Maintenance Plan:

After the testing plan, maintenance plan prioritizes fixing bugs, reviewing new requests on the feature on a regular basis. Planning regular release and update to the system, and introducing new features on a regular basis, can much improve the system. Deploying the system on a good platform also comes insides this maintenance plan. Involving the users and stakeholders on regular basis and getting productive feedback from them would be a good plan for the regular maintenance and enhancements.

9) Prototype:

Prototype is a simulation of the appearance, feel, and functionality of the final project. The prototype lacks the precise logic that that is required in the software development phase. Members of the software development team use it to plan user testing and feedback. Additionally, they differ in sophistication. The process of developing an idea, drawing it, and turning it into a clickable prototype that imitates actual software is known as software prototyping (Sharma, 2022).

For the prototypes of the system, that is required are given below:

9.1) Home Page:

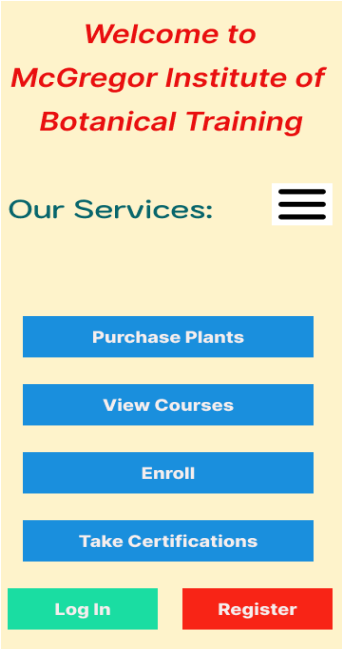

Figure 7: Home Page Design.

9.2) Login Page:

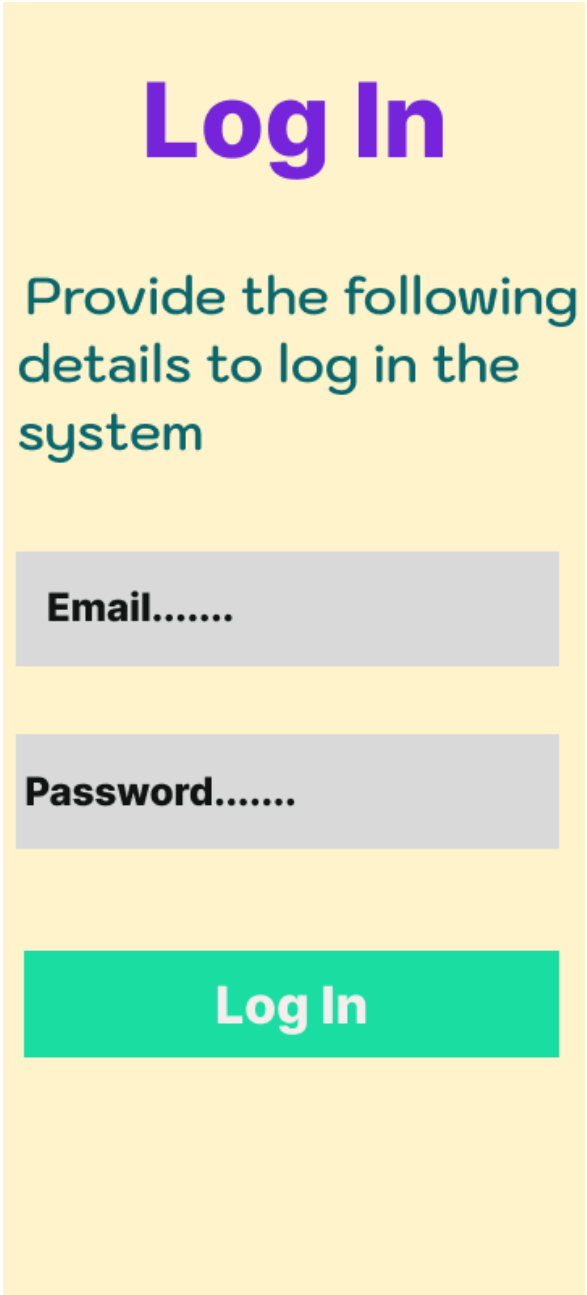

Figure 8: Login Page.

9.3) Register Page:

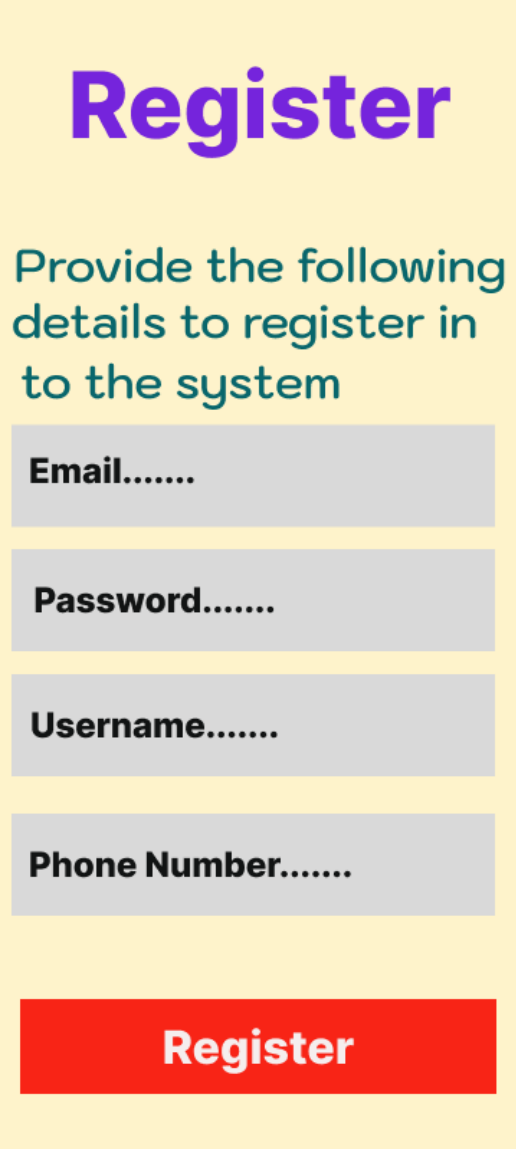

Figure 9: Register Page

9.4) View Plants Page:

Figure 10: View Plants Page

9.5) View Courses Page:

Figure 11: View Courses Page

9.6) Purchase Plants Page:

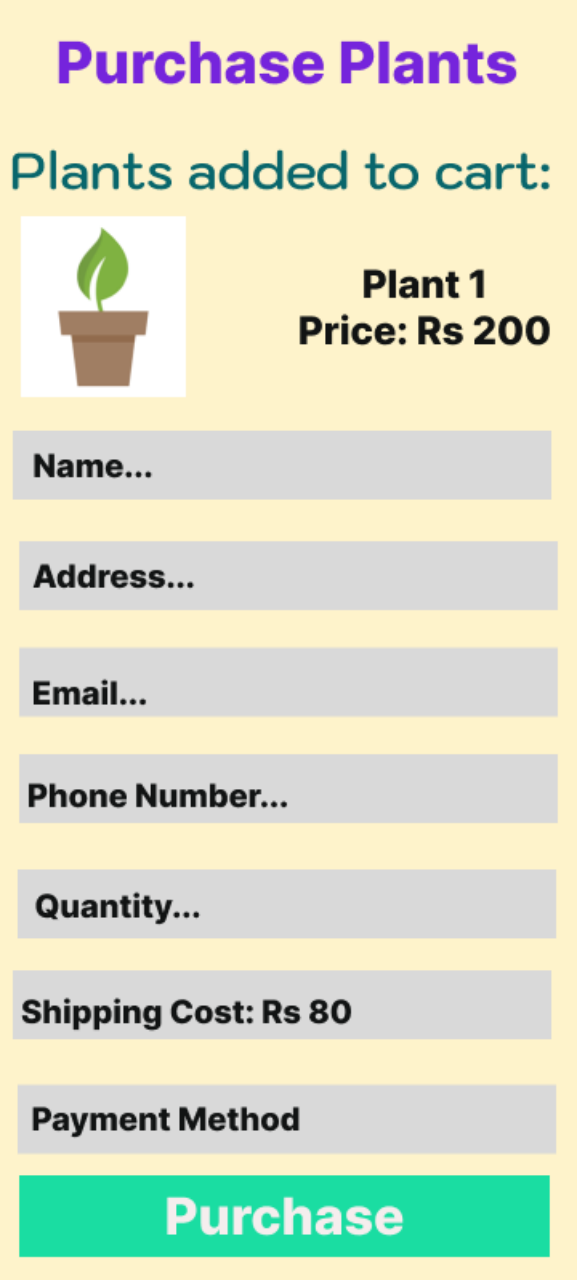

Figure 12: Purchase Plants Page.

9.7) Purchas Course Page:

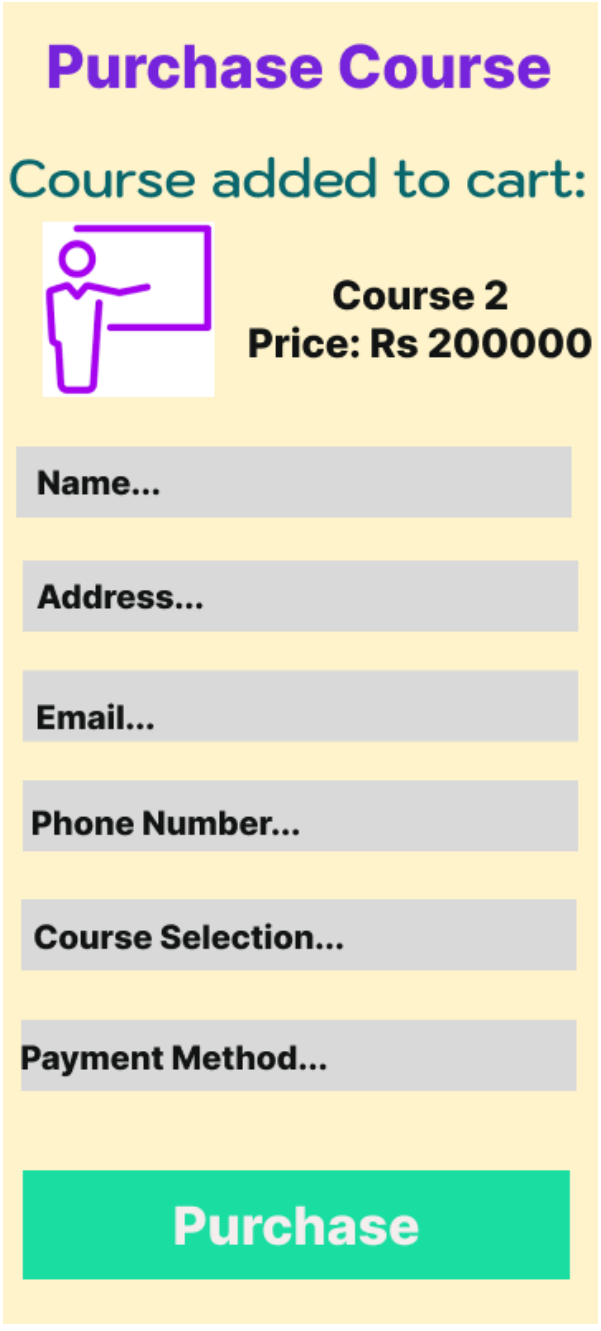

Figure 13: Purchase Course Page.

9.8) Enrollment Page:

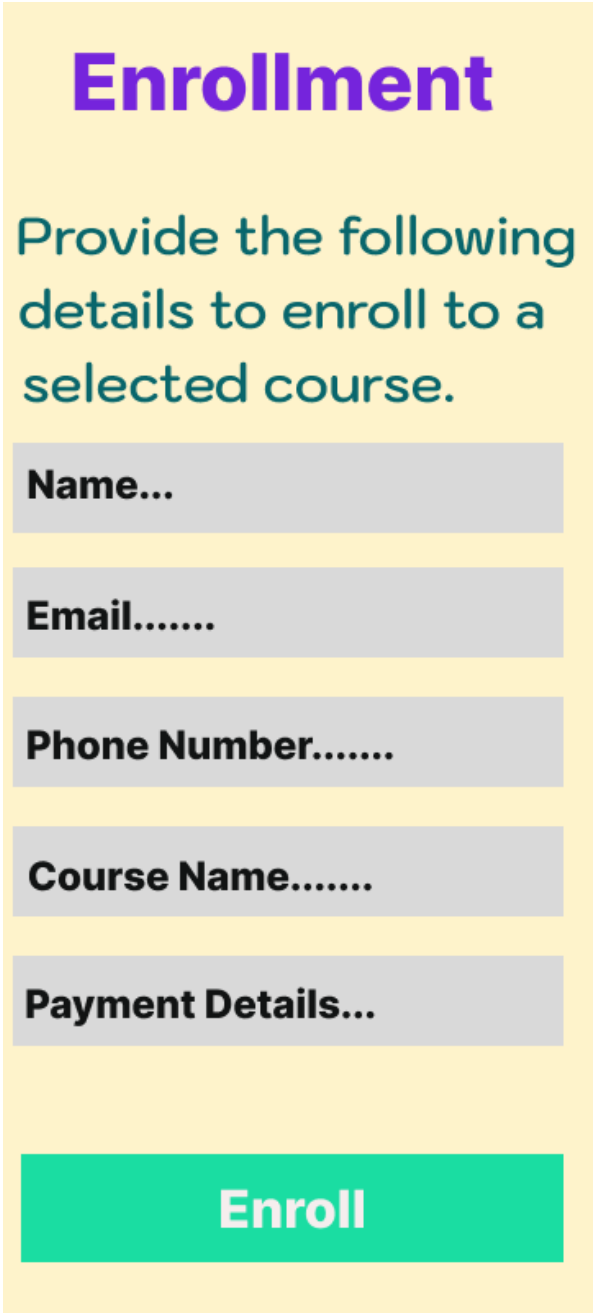

Figure 14: Enrollment Page.

9.9) Take Certification Examination Page:

Figure 15: Take Certification Examination Page.

9.10) Course Management Page:

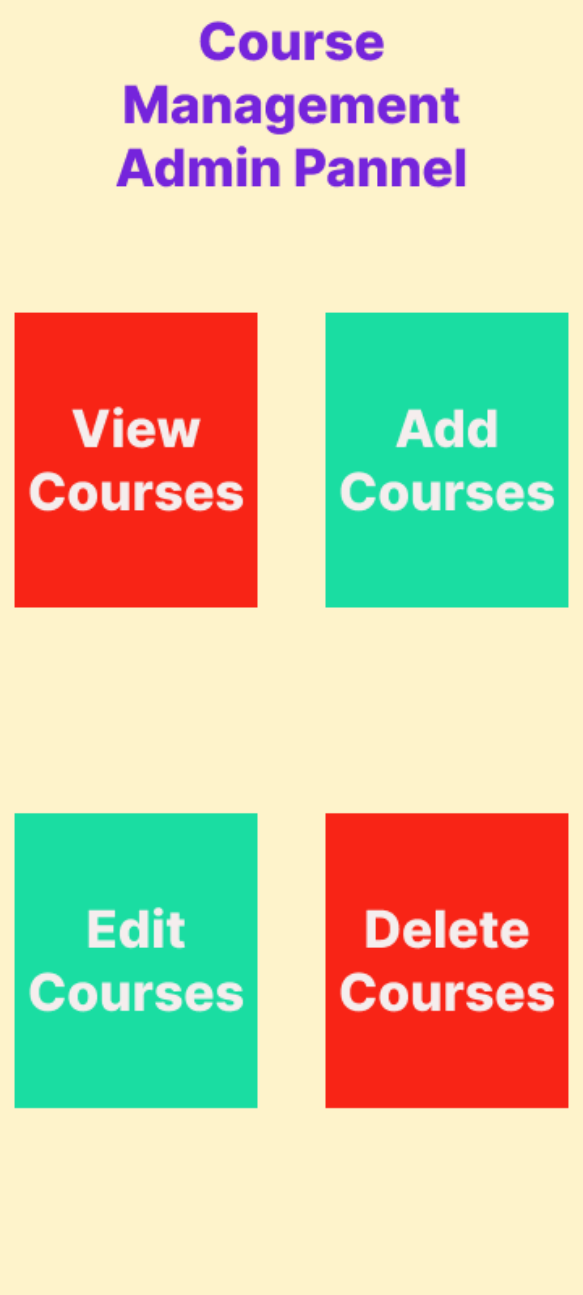

Figure 16: Course Management Page.

9.11) Login Page:

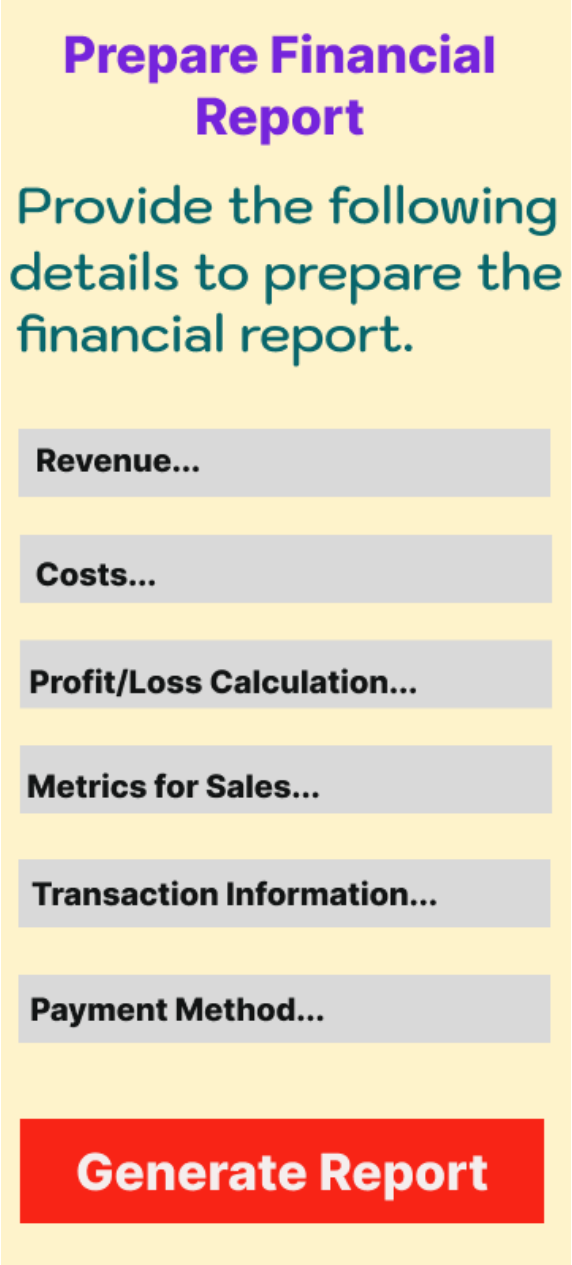

Figure 17: Financial Report Preparation Page.

9.12) Login Page:

Figure 18: Employee Report Preparation Page.

9.13) User Report Preparation Page:

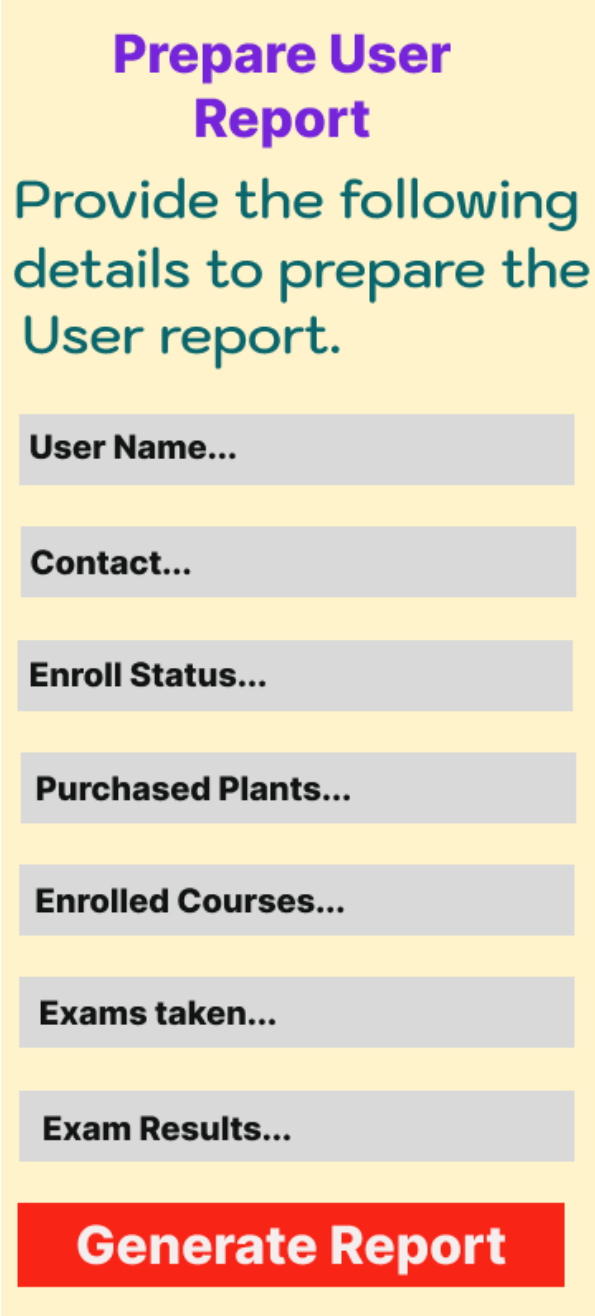

Figure 19: User Report Preparation Page.

9.14) Payment Process Page:

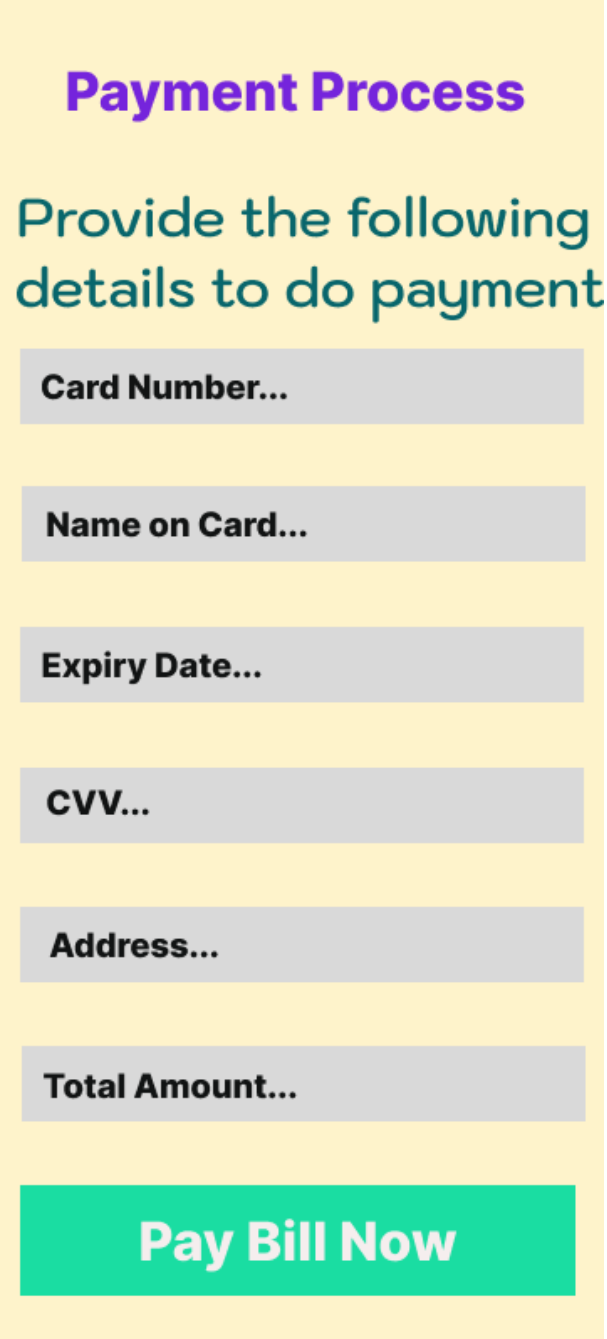

Figure 20: Payment Process Page.

9.15) Ask for Recommendations Page:

Figure 21: Ask for Recommendations Page.

10) References and Bibliography:

Anon., 2022. *what-is-a-use-case-template-how-to-writ.* [Online] Available at: https://blog.logrocket.com/product-management/what-is-a-use-case-template-how-to-write/
[Accessed 16 April 2024].

Ebrary, 2023. *expanded_cases.* [Online] Available at: https://ebrary.net/73317/computer_science/expanded_cases
[Accessed 17 April 2024].

geeksfoegeeks, 2023. *difference-between-sequence-diagram-and-collaboration-diagram.* [Online] Available at: https://geeksforgeeks.org/difference-between-sequence-diagram-and-collaboration-diagram/
[Accessed 18 April 2024].

geeksforgeeks, 2024. *use-case-diagram.* [Online] Available at: https://www.geeksforgeeks.org/use-case-diagram/
[Accessed 16 April 2024].

Pandey, S., 2024. *what-is-class-diagram.* [Online] Available at: https://www.naukri.com/code360/library/what-is-class-diagram
[Accessed 20 April 2024].

Sharma, I., 2022. *what-is-software-prototyping.html.* [Online] Available at: https://www.tatvasoft.com/outsourcing/2022/05/what-is-software-prototyping.html
[Accessed 24 April 2024].

Shweta, C. B., 2024. *what-is-a-gantt-chart.* [Online] Available at: https://www.forbes.com/advisor/business/software/what-is-a-gantt-chart/
[Accessed 15 April 2024].

Team, A. E. C., 2023. *what-is-work-breakdown-structure.* [Online] Available at: https://business.adobe.com/blog/basics/what-is-work-breakdown-structure [Accessed 15 April 2024].

VanZandt, P., 2023. *what-is-sequence-diagram.* [Online] Available at: https://ideascale.com/blog/what-is-sequence-diagram/ [Accessed 20 April 2024].