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: Asmita Basnet

London Met ID: 22085764

College ID: np01ai4s230019@islingtoncollege.edu.np

Assignment Due Date: May 7

Assignment Submission Date: May 7

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. Introduction

The McGregor Institute of Botanical Training, an Irish institution based in Godawari, Lalitpur, has been operating in Nepal for around seven years. It offers a variety of undergraduate and graduate courses with a focus on horticulture and agriculture and is associated with Dublin City University.

Nestled in Nepal's scenic surroundings, the McGregor Institute of Botanical Training has served as a knowledge hub for the region's diverse flora and fauna. The institute, which is celebrating seven years of cultivating a passion for horticulture and agriculture, recognizes that innovation is required to meet the growing needs of botanical education. The institution seeks to meet the expanding demand for botanical study while also developing a close-knit community based on a common love of plants. The McGregor Institute aims to provide a multifaceted platform that merges education, plant trade, and community engagement beyond typical academic bounds.

Rather than just a transactional platform, this platform is meant to function as a vibrant canvas for fostering a community of plant aficionados in the heart of Nepal. The McGregor Institute embarks on a revolutionary mission to develop a software system that can adapt to the institute's changing needs throughout time. This effort aims to provide the groundwork for the future of Nepal's botanical education sector. The software solution will expedite enrollment and plant purchases while also encouraging expert discussions, debates, and idea sharing. This paper conveys the spirit of the Institute's pursuit for knowledge, community, and sustainability by detailing the project's nuances from start to finish, including expected milestones.

1.2. Project overview

The McGregor Institute plans to offer a variety of short-term horticulture certification programs for individuals from all backgrounds who desire to expand their knowledge and skills in this industry. In order to foster a better relationship between the community and the environment, the institution intends to democratize access to plants by offering a variety of plant varieties for sale, some of which will be distributed for free.

This Institute intends to develop a feature-rich software system that will include expert recommendations, report generation, certification testing, plant purchasing, program enrollment, user registration, payment processing, and a community interaction forum. The project will follow a methodical approach, beginning with the gathering and analysis of requirements and progressing to system design, implementation, testing, and deployment. The Agile technique will be used throughout the development process, allowing for incremental improvements and alterations to meet changing requirements.

The McGregor Institute aspires to be Nepal's foremost center for botany training and community involvement, while also improving its operations and user experience.

1.2. Scope

- **Inclusion**

1. User Registration: The institute's platform can be seamlessly integrated by community members with the help of an easy-to-use registration process.

2. Enrollment in Courses: Anyone interested in short-term certification courses can do so with ease due to an easy-to-use and accessible system.

3. Buying Plants: An intuitive interface makes the process of buying plants easier and provides a wide selection of kinds to suit the preferences of buyers.

4. Payment Processing: By integrating a secure payment gateway, transactions for course fees and plant purchases are made safe and easy.

5. Expert Suggestions: Providing users with the option to consult experts according to their region and soil type improves community awareness and involvement in plant choosing.

6. Participation in the Forum: The forum provides a lively setting for community members to exchange ideas and have in-depth conversations about subjects pertaining to plants.

7. Notification System: Users are kept informed and involved with personalized notifications, which promote a sense of community and connectedness.

8. Report Preparation: To support responsible and informed decision-making, administrators possess the capacity to produce detailed reports that include financial statements, personnel information, and user-specific data.

9. Take Certification examinations: By offering a systematic pathway for skill development and accreditation, users may easily access practice tests, review their results, and take certification examinations.

- **Exclusions**

There are no exclusions for the system.

1.3. Intended Audience

The target audience for the McGregor Institute of Botanical Training's software system is:

• Aspiring students: Students interested in horticultural and agricultural programs, including undergraduate, graduate, and short-term certifications.

• Plant enthusiasts: They are community members who enjoy gardening, plants, and sustainable agriculture. They want to buy plants and talk about anything relating to plants.

• Experts: Experts in agriculture, horticulture, and botany provide advice and guidance on plant cultivation and selection.

• Community members engage in conservation discussions, organize activities, and seek support from professionals and enthusiasts.

• Candidates for consideration: Students seeking accreditation in horticulture and related areas, including sample tests, results, and preparation for certification exams.

• Target audience: Those interested in botanical education, conservation, and sustainable agriculture techniques.

2. System Requirement

In order to understand the needs and expectations of clients and stakeholders, direct contact was necessary during the requirements collecting process.

2.1. Functional Requirements

- **User Registration:**

1. Those who have a working email address can sign up.
2. An account verification procedure is available.

- **Course Enrollment:**

1. In addition to enrolling, users can browse and examine course details.
2. The system delivers confirmation and keeps track of enrollments.

- **Plant Purchasing:**

1. Details of the available plants are displayed in the catalog.
2. Plants can be added to shopping carts, checked out, and securely paid for by users.

- **Payment Processing:**

1. Integration with safe payment gateways such as "Khalti," "Esewa," and so on.
2. Payment records kept in an administrative file.

- **Expert Recommendations:**

1. Users submit inquiries regarding the location and the state of the soil.
2. Plant guidance and perspectives are offered by experts.

- **Forum Engagement:**

1. Members of the community can upvote, remark, and post.
2. Users are informed about activity on their postings by the notification system.

- **Notification System:**

1. Notifications are sent to users according to their activities.
2. Notifications can be about expert responses, discussions in the forum, modifications to the course, etc.

- **Report Preparation:**

1. Reports pertaining to users, employees, and finances can be generated with administrative access.
2. Analyzable visual data is supplied.

- **Take Certification Exams:**

1. After applying and meeting the conditions, users are permitted to take practice exams, examine their results, and take part in certification exams.

2.2. Non-functional Requirements

- **Performance:**

1. The system must effectively manage concurrent user interactions.

- **Security:**

1. Payment information, user data, and personal information must be securely stored and encrypted.

- **Scalability:**

1. Future growth to support a growing user base should be accommodated by the architecture.

- **Usability:**

1. For a wide range of users, the user interface must be clear and easy to use.

3. Project planning

3.1. Work Breakdown Structure

A work breakdown structure (WBS) visibly divides project deliverables into layers depending on dependencies. It's simply project plan in graphic form, with the project goal at the top and dependents and sub-dependencies underneath. (Raeburn, 2024)

Work Breakdown System For Proposed System Development:

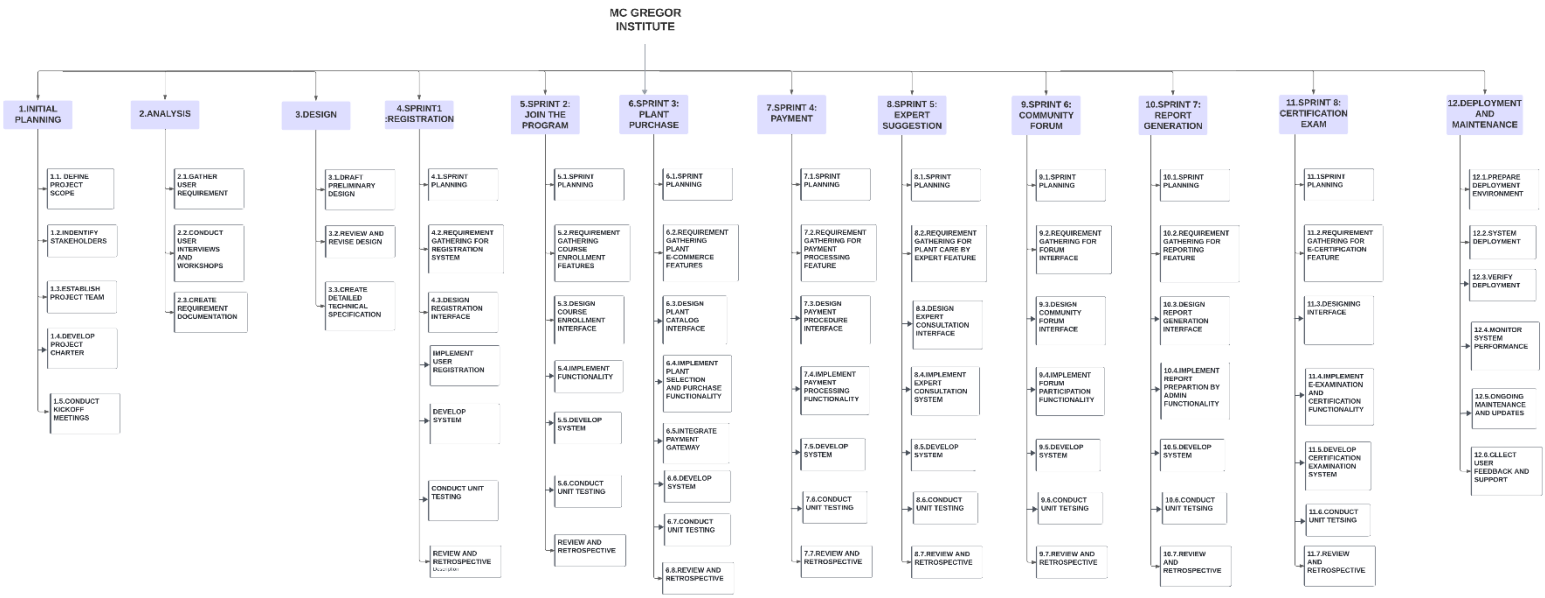


Figure 1:Work Breakdown Structure

4. GANNT Chart

A Gantt chart is a horizontal bar chart that depicts the progression of a project and its tasks. This provides your team with a visual overview of project details such as the project schedule, impending milestones, and overall project timetable.

Each horizontal bar in the graphic represents a task, and the length of each bar indicates how long that step or work will take. When you zoom out to see the big picture, these charts give project managers and teams an overview of what work needs to be done, who is doing it, and when. (Martins, 2024)

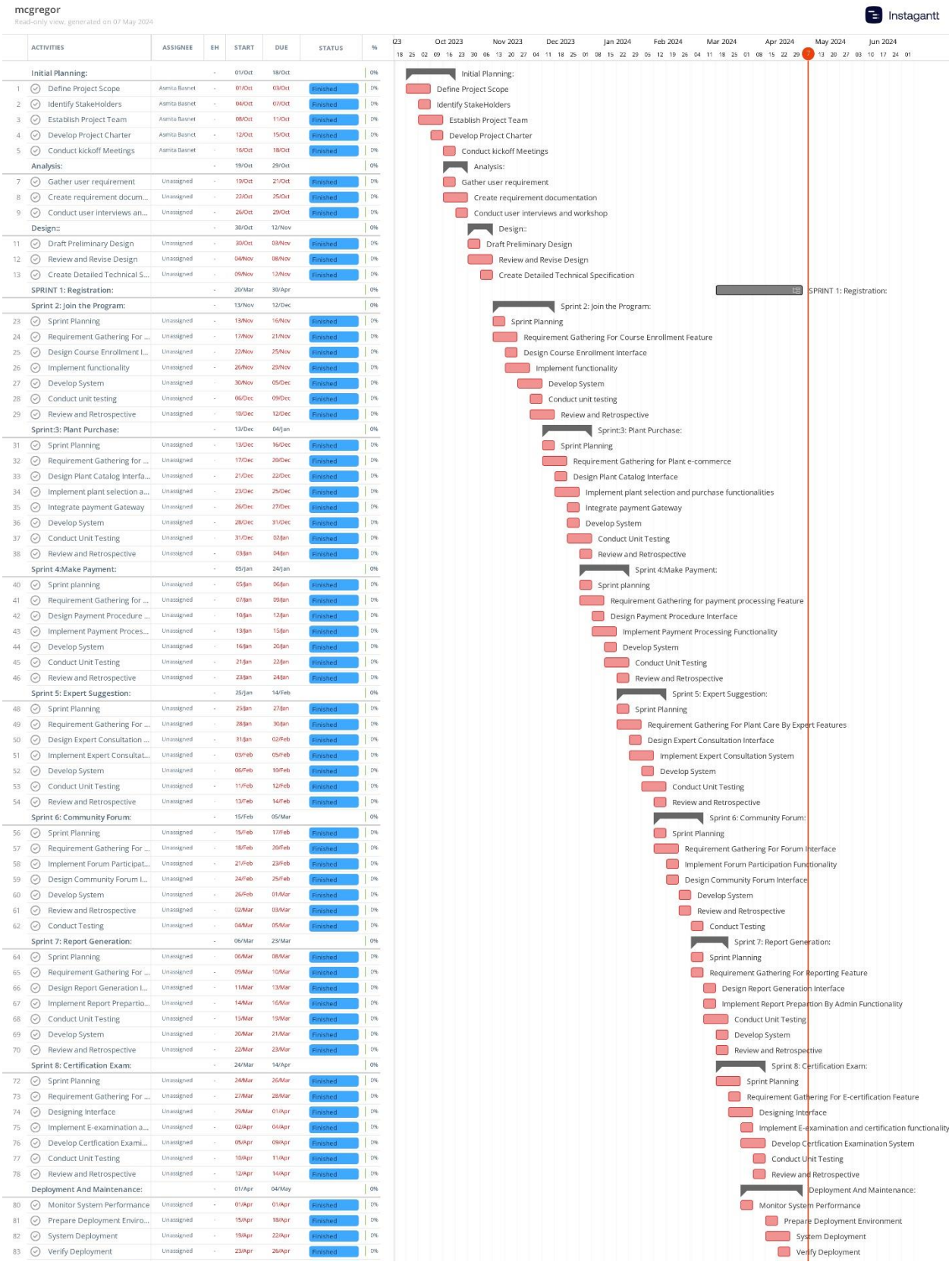


Figure 2:Gannt Chart

5. System Analysis

5.1. Use-Case Diagram

A Use Case Diagram is a sort of Unified Modeling Language (UML) diagram that depicts the interaction between actors (users or external systems) with the system under examination in order to achieve certain goals. It provides a high-level overview of the system's functioning by demonstrating the numerous ways people can interact with it.

UML notations provide a visual language that allows software developers, designers, and other stakeholders to communicate and document system designs, architectures, and behaviors in a consistent and intelligible way. (geeksforgeeks, 2024)

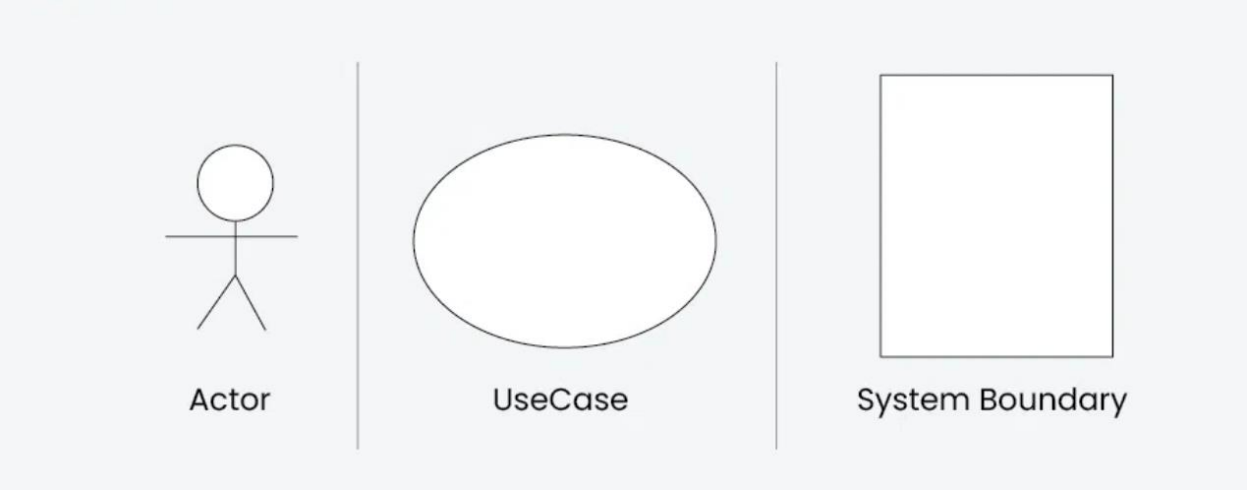


Figure 3: Use Case Diagram Notation

5.1.1. Actors

Actors are external entities that communicate with the system. They initiate use cases and receive results. The proper identification and understanding of actors is critical for effectively modeling system behavior. (geeksforgeeks, 2024)

5.1.2. Use Case

Use cases function similarly to scenes in a play. They reflect the precise functions that your system can perform. The use case is represented by ovals. (geeksforgeeks, 2024)

5.1.3. System Boundary

The system boundary is a visual depiction of the scope or bounds of the system being modeled. It determines what is internal and external to the system. The system boundary is often portrayed as a rectangular box that encompasses all of the system's use cases. (geeksforgeeks, 2024)

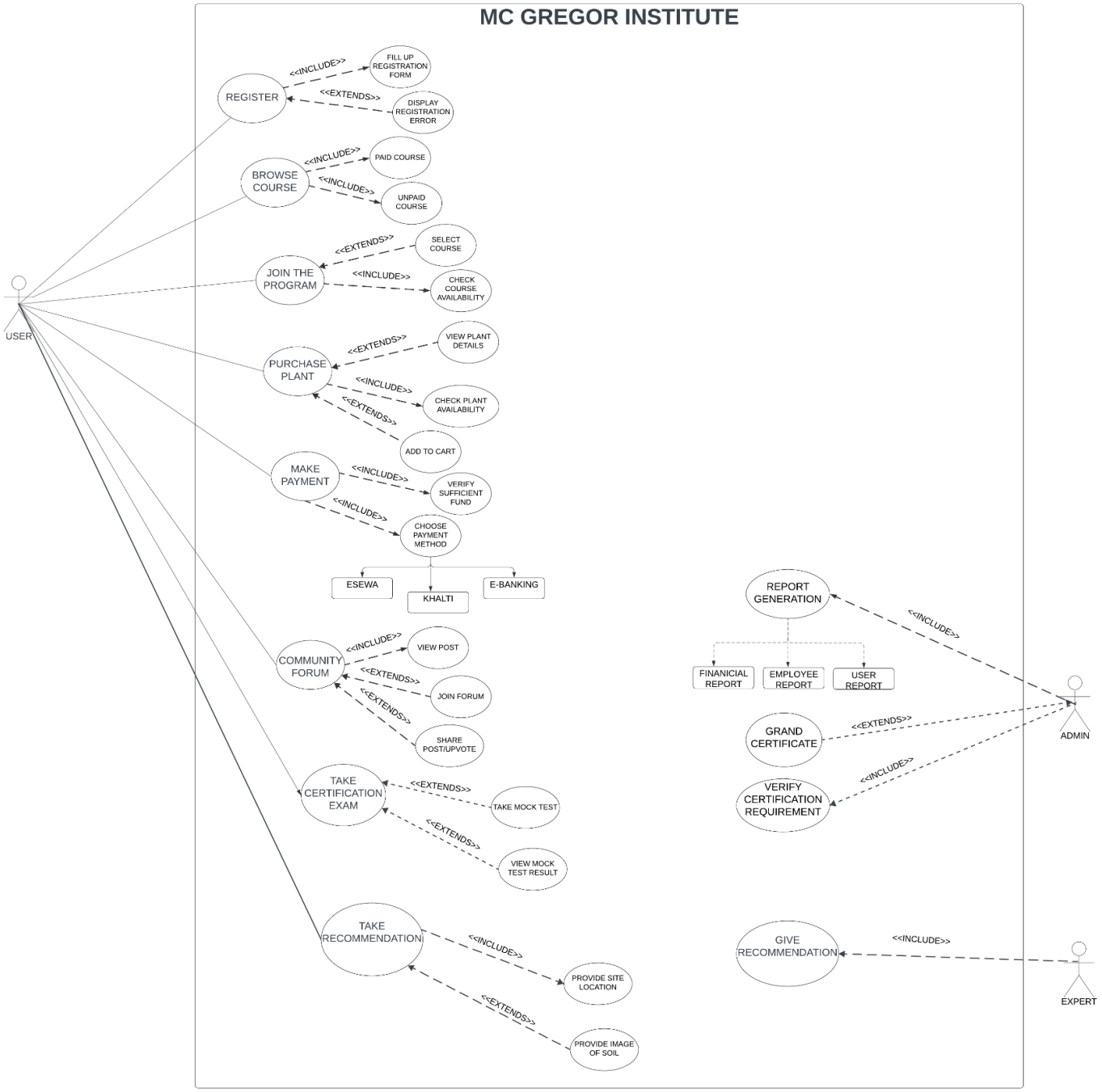


Figure 4: Use-Case Diagram

5.2. High Level Use Case

1. Register

Use Case	**Login**
Actor	User
Description	Users who have registered can log in by entering a valid username and password. If data entered is invalid, an error message is displayed.

Table 1:High level Use case for register

2. Browse Course

Use Case	**Browse Course**
Actor	User
Description	The user peruses the list of programs, taking note of their titles, descriptions, and enrollment specifics. Certain courses are offered free of charge, while others require payment based on the specific course requirements.

Table 2:High level Use Case For Browse Course

3. Join the program

Use Case	**Join the program**
Actor	User

Description	The user can choose a particular program based on its suitability for their interests, degree of competence, and career objectives.

Table 3:High level Use Case For Join The Program

4. Purchase plants

Use Case	**Purchase plants**
Actor	User
Description	The user peruses the plant catalog, which provides comprehensive details about every plant, including common name, scientific name, description, requirements for growing, cost, and availability.

Table 4:High Level Use Case For Purchase Plant

5. Make payment

Use Case	**Make payment**
Actor	User
Description	This use case describes how a user can pay for a purchase or enroll using the online platform of system. In order to ensure smooth transactions for users enrolling in classes, buying plants, or utilizing other services provided by the institution, it provides a number of payment alternatives.

Table 5:High Level Use Case For Make Payment

6. Community Forum

Use Case	**Community Forum**
Actor	User
Description	This use case describes the features of the Institute's community forum. The forum provides a venue for plant enthusiasts and students to converse, exchange knowledge, look for guidance, and build a sense of community around botanical studies and related subjects.

Table 6:High Level Use Case For Community Forum

7. Take Certification Exam

Use Case	**Take Certification Exam**
Actor	User
Description	This use case describes how a user can take a certification exam offered by McGregor Institute of Botanical Training. McGregor Institute's training programs include certification tests, which allow users to assess their knowledge, skills, and proficiency in various areas of botanical sciences and horticulture.

Table 7:High level Use Case For Take Certification Exam

8. Report Generation

Use Case	**Report Generation**

Actor	Admin
Description	The admin chooses the appropriate report type or category from the selections that are available, such as financial reports, employee reports, user activity reports, program enrollment reports, or custom reports based on specified criteria or parameters.

Table 8:High Level Use Case For Report Generation

9. Take Recommendations

Use Case	**Give Recommendations**
Actor	Experts
Description	The expert sends the plant recommendations to the user via email, notification, or direct message inside the system, and includes full information about each recommended plant, such as its common name, scientific name, growth needs, care tips, and potential advantages.

Table 9:High Level Use Case For Take Recommendation

5.3. Expanded Use Case

5.3.1. User Registration

Use Case: User Registration

Actor: User

Description: The user completes the registration form, providing needed information such as their full name, email address, desired username, password, and any other optional fields or

profile information asked by the institute. When the process is complete, the system confirms the user's registration and activates their account, giving them access to the platform's features and services.

Actor Action	**System Response**
1. User navigates through McGregor Institute registration page.	
2. Fills out all the necessary details required to fulfill registration process.	
3. They go through all the agreement and policies and agrees it.	
4. The user submits the registration form after which system process the request.	5. The system accepts the registration form submitted by the user.
	6. The system sends verification link to user to confirm their email.
7. The user confirms their email by clicking on the link	8. The system authorizes the account
	9. The system allows user to access the resources and features after successful registration.

Table 10:Expanded Use Case For Registration

Alternative Cases:

Line 4: If the user submits registration form with incomplete or wrong details, the system ask user to correct the error before moving towards further processing.

Line 2: If the user registers with the existing email or username that is already in database, the system notifies user about duplication and ask to enter another username or email.

Line 6: If user does not confirm the verification link sent by the system to confirm their email, an error message is displayed for unsuccessful registration.

5.3.2. Join the program

Use Case: Purchase Plant

Actor: User

Description: The "Join the Program" use case demonstrates how a user can enroll in a program offered by the McGregor Institute of Botanical Training. Users interested in expanding their knowledge and skills in botanical studies or related subjects can look through the institute's program offerings and enroll in courses that match their educational objectives and interests.

Actor Action	System Response
1. User navigates through course enrollment section.	2. Present available programs to user, including price and details of program
3. Browses through available courses varieties and select the program of interest	4. Receives user’s request to join the program
5. User processed to enrollment process	6. System processes enrollment request and verify eligibility
7. Choose payment method and complete transaction if payment is required	8. Process transactions securely with choose method.
	9. System confirms user’s enrollment and provide all the necessary details related to program.

Table 11:Exapanded Use Case For Join the Program

Alternative Case

Line 2: If user selects the course that has reached its maximum capacity, system displays message informing for waiting list until further opening.

Line 6: If the transaction fails, The user is notified of the payment failure and is asked to retry the transaction or use a different payment method.

6. Sequence Diagram

A sequence diagram simply illustrates the interaction between items in a sequential manner, i.e. the order in which these interactions take place. Sequence diagrams show how and in what order the objects in a system operate. These diagrams are commonly used by businesspeople and software developers to document and comprehend requirements for new and current systems. (geeksforgeeks, 2024)

Here are the notations for sequence diagram:

1. Actors: An actor in a UML diagram depicts a type of role that interacts with the system and its objects.

Figure 5: UML Diagram Actor

2. Lifelines: A lifeline is a named element that represents an individual participant in a sequence diagram. So, each incident in a sequence diagram is symbolized by a lifeline.

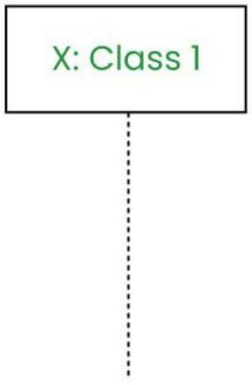


Figure 6:LifeLines for UML Diagram

3. Messages: Messages are used to show the communication between items. On the lifeline, the messages show in sequential succession.

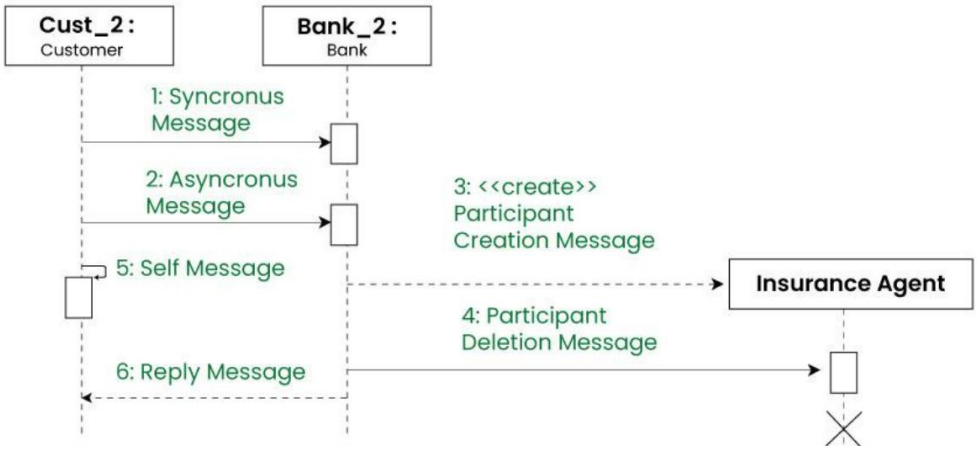


Figure 7:Messanger For Uml Diagram

4. Self-Message: Certain conditions may need the object to deliver a message to itself. Such messages are known as Self Messages, and they are represented by a U-shaped arrow.

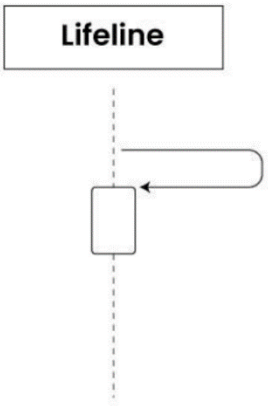


Figure 8:Self Message for Uml Diagram

(geeksforgeeks, 2024)

Here is the sequence diagram for Join the program Feature:

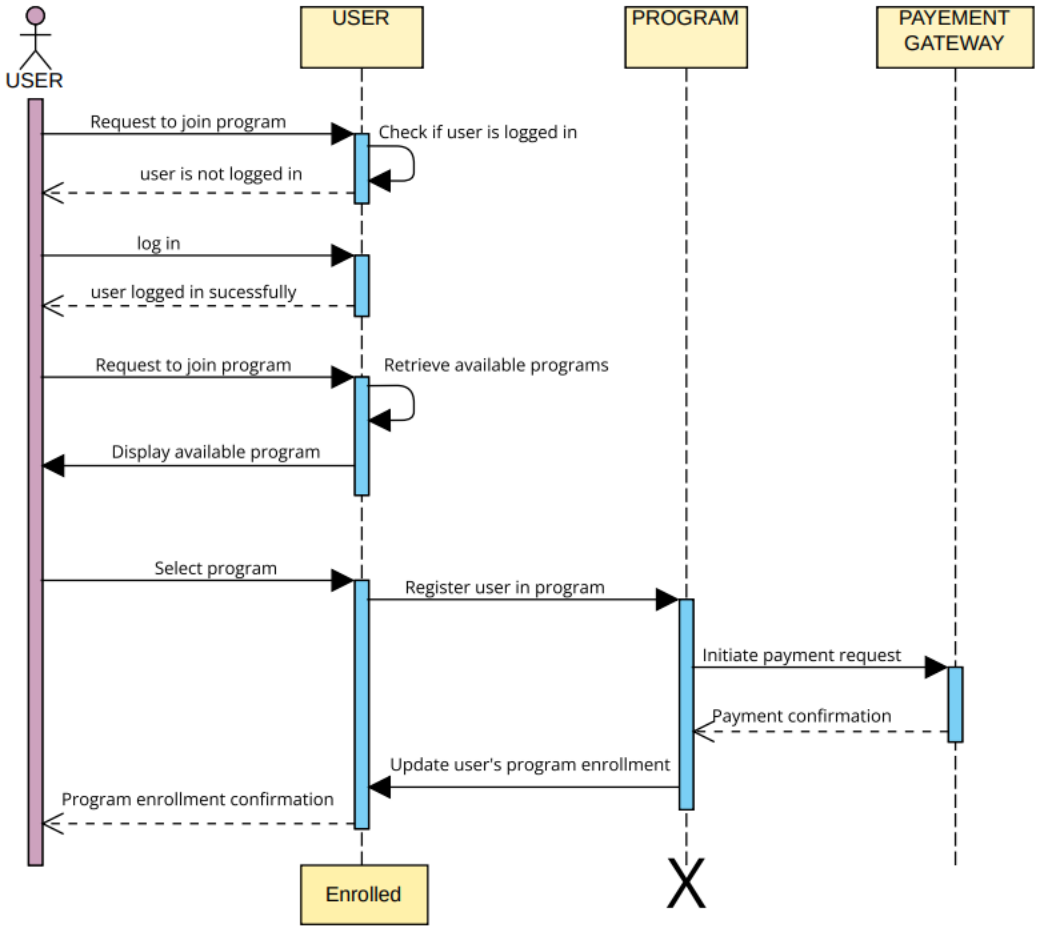


Figure 9:Sequence Diagram for the System

7. Communication Diagram/ Collaboration Diagram

A Collaboration / Communication Diagram is a form of Interaction Diagram that depicts the interactions and relationships among objects in a system. It demonstrates how objects work together to accomplish a given goal or action. They are used to depict a system's dynamic behavior and show how messages flow between components throughout a certain scenario or use case. They are important in system development because they help with understanding, communication, design, analysis, and documenting of the system's architecture and behavior. (geeksforgeeks, 2024)

Here are few main notations used in Communication Diagram:

1. Object: Objects are symbolized by rectangles with their names on top. Each object involved in the interaction is depicted as a distinct rectangle in the diagram.

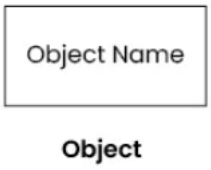


Figure 10:Object For Communication Diagram

2. Actor: They are typically displayed at the top or side of the diagram, highlighting their role in interactions with the system's objects or components. They remain linked to things via messages, demonstrating communication with the system.

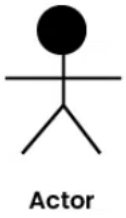


Figure 11:Actor for Communication Diagram

3. Link: Links indicate linkages or relationships between objects. Links are displayed as lines that connect items, with option labels to reflect the nature of the relationship. Links can be unidirectional or bidirectional, determined by the type of relationship.

Figure 12:Link for Communication Diagram

4. Message: Messages reflect how objects interact with each other. Messages are presented as arrows connecting objects, corresponding to the flow of communication.

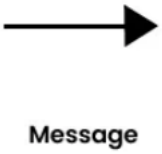


Figure 13:Message For Communication Diagram

(geeksforgeeks., 2024)

Here is the Communication Diagram for Join the program Feature:

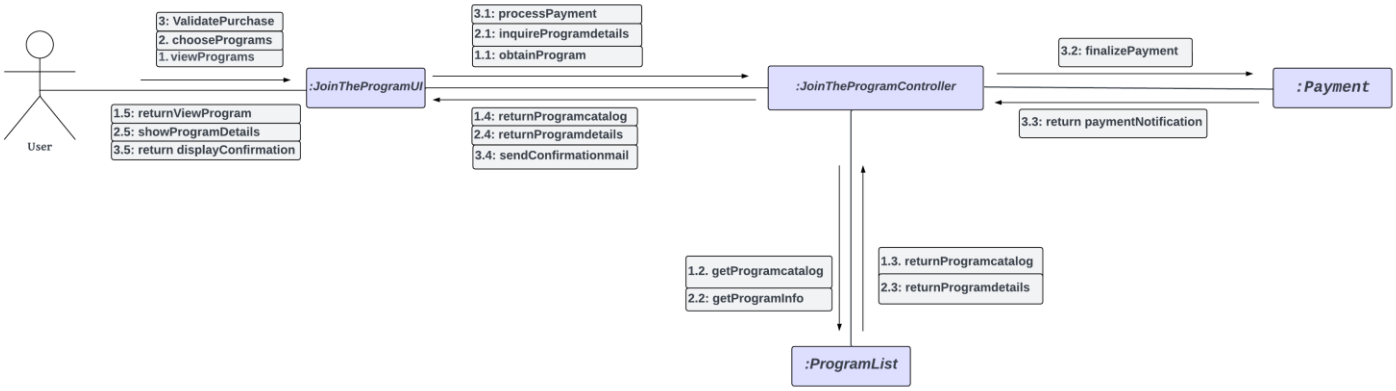


Figure 14:Communication Diagram For The System

8. Class Diagram

Class diagrams are a sort of UML (Unified Modeling Language) diagram that is used in software engineering to visually describe the structure and interactions of classes inside a system, or to build and visualize object-oriented systems. Classes are displayed as boxes, with three compartments for the class name, properties, and methods. Lines connecting classes represent associations, such as one-to-one or one-to-many. Class diagrams provide a high-level overview of a system's design, making it easier to describe and document the software's structure. They are an essential tool in object-oriented design and play an important role throughout the software development lifecycle. (geeksforgeeks, 2024)

8.1. Domain Classes

Here is representation of Domain Classes based on High Level Description:

Use Case	Domain Classes
Registration	Student, Customer, User
Enrollment	Student, Course
Purchase Plants	Customer, Plant, Transaction, Receipt
Payments	Student, Customer, Transaction, Receipt
Get Recommendations	User, Recommendation
Certifications Exam	User, Exam, Certificate
Forum	User, Forum, Post, Discussion

Table 12:Domain classes

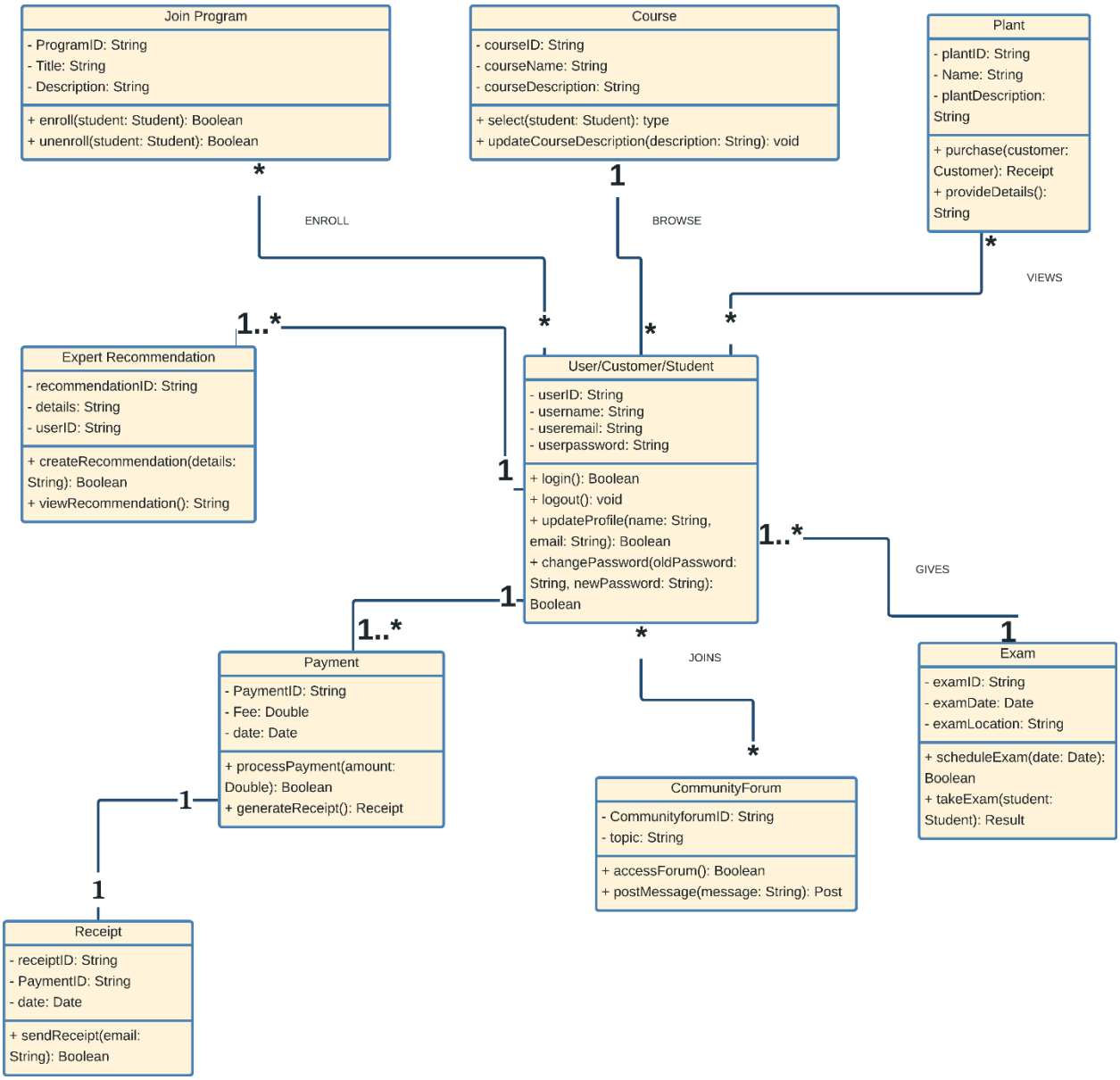


Figure 15:Class Diagram

9. Further Development

For my project, I've used **Agile Methodology** considering careful evaluation of the fundamental flexibility of online systems, as well as the possibility of growing user needs, It is focused on taking gradual and iterative stages to complete undertakings. The incremental portions of a project are completed in short development cycles. The strategy promotes speedy delivery and adaptability.

Selection of Agile Methodology was done based on trait:

- It provides instant feedback, which promotes software enhancement in every subsequent increment.
- The Agile approach promotes beneficial personal interaction.
- It successfully handles and accepts regular modifications in project requirements based on stakeholder preferences.
- It is appropriate for its time-saving techniques, that assist in software delivery and reduce time-to-market.

We want to create an innovative online platform that can satisfy the requirements of the McGregor Institute and those who use it, which will ensure flexibility and upgrades as the project advances. Through every step of the development procedure, Following essential Agile practices are implemented.

Sprint: In sprint, it provides a time box for a month or less, and a new sprint instantly begins following the completion of the one preceding it.

Sprint planning: Here, the project manager determines the goals and steps for the subsequent sprint.

Daily Scrum: It is essentially a stand-up meeting where sprints are run on a set timetable and a discussion of completed tasks is conducted. It usually takes no more than 15 minutes.

Sprint Review: If the program has features that are not beneficial or necessary, it is reviewed and then submitted to Sprint Retrospective.

Sprint Retrospective: Here, the product's qualities and condition are evaluated.

9.1. Further Development Strategy

Architectural Choice: Microservices Architecture Pattern

Reason For Selection: I choose this pattern as it correlates with Agile principles, as it emphasizes flexibility, scalability, and adaptability. This architecture breaks down each software program into small individually accessible services and then builds a small new program whenever feature needs to be added. It provides continuous distribution and integration, allowing teams to swiftly release updates and respond to changing requirements, both of which are essential aspects of Agile methodology.

9.2. Design Plans: MVC Architecture

The Model-View-Controller pattern would be the most appropriate design pattern when considering all of the different aspects of our system's design. Between the underlying database system and our microservices' business logic, the MVC paradigm adds a substantial layer of abstraction. This was selected because it has a number of benefits, such as:

- It divides an application's many components, making code simpler to comprehend and modify.
- The development procedure can be sped up by having multiple developers work on different components at once.
- Each component (Model, View, and Controller) may be developed and evaluated independently, which promotes code reuse and scalability.

9.3. Development Plan

The goal of the online system's development strategy is to employ the MVC pattern to create a scalable and reliable microservices architecture. The following is a list of this development plan's main components:

9.3.1. Programming Language:

Python: Python is popular because of its many libraries, ease of use, and readability. Building microservices with it is advantageous, particularly for tasks like web development, machine learning, and data processing.

9.3.2 Database:

PostgreSQL: PostgreSQL is suitable for microservices because it ensures data consistency, which is critical for transaction-processing services. It is adaptable, so it can handle a wide range of data and requests from many services.

9.4. Features to Develop:

These are the primary features used in the system that will include features that improve customer service, deliver useful information, and broaden the scope of offerings available from the institute.

1. **User Administration**: It enables user registration, authentication, and authorization, ensuring secure system access.
2. **Course Administration**: This feature is a crucial component of the online learning platform that manages enrollment, and access of courses.
3. **E-Nursery Service**: This service allows customers to purchase plants according to their choice and with full-fledged services and description creating funding for the institute.
4. **Payment Gateway:** This feature integrates with safe transaction pathways, simplifying the entire buying procedure.
5. **Expert Suggestion**: This service uses customer location info and soil condition and requests to enable experts to suggest plants and resources that are appropriate
6. **Report Generation:** Admins can create a variety of reports to fetch the data, if necessary, in future and to maintain secure transaction details.
7. **Take Certification Exam:** This function manages the operation and delivery of online accreditation tests, which may broaden the institute's appeal.
8. **Community Forum:** This feature allows user to Join the community where plant enthusiasts are engaged and share their experience and idea.

9.5. Testing Levels

A thorough testing strategy is required to assure the accuracy and stability of the system. The below section will provide all the necessary testing required for the system:

- **Unit Testing**: This is significant because microservices frequently contain particular features, and unit tests ensure that each microservice operates properly on its own without interruption.
- **Integration Testing**: Integration testing in microservices ensures that all of the system's components communicate effectively. It tests to see whether operations like delivering messages or transferring data between services operate as expected.
- **System Testing**: System testing for microservices determines whether the overall system, consisting of all the individual services, functions properly. It assesses whether the system performs as expected and acts well in various conditions. This testing ensures that everything fits together properly and runs efficiently as an entire system.
- **Security Testing**: Security testing ensures that private information is secured, and that the system is not vulnerable to assaults.

9.6. Black Box and White Box Testing

- **Black Box Testing**: Black box testing ensures that each microservice functions properly when approached by different applications or users.
- **White Box Testing**: White box testing identifies possible problems with microservices deployment and assures compliance with recommended procedures and codes of conduct.

9.7. Maintenance Plans

A strong maintenance strategy ensures system efficiency and scalability remain intact throughout time. Key features include:

- **Monitoring and Logging**: Implementing extensive monitoring tools to keep updated on system performance, usage of resources, and problems. This will guarantee uptime, effective resource management, and prompt error identification.
- **Version Monitoring and Backups**: For tracking code changes, allow backups, and promote teamwork, use Git or comparable tools. Maintain regular records and program code backups that will help in data recovery in the event of mistakes or catastrophes.

- **Issue Management**: To prevent repetition, clearly identify responsibilities, communication channels, troubleshooting methods, and post-incident evaluation in your response to system problems.
- **Security Updates**: To address risks, keep all software components up to speed with the most recent security patches. Use scanning tools for vulnerabilities to find possible safety risks and setup errors in advance.

10. User Interface (UI) Design

- User Home Page For The System

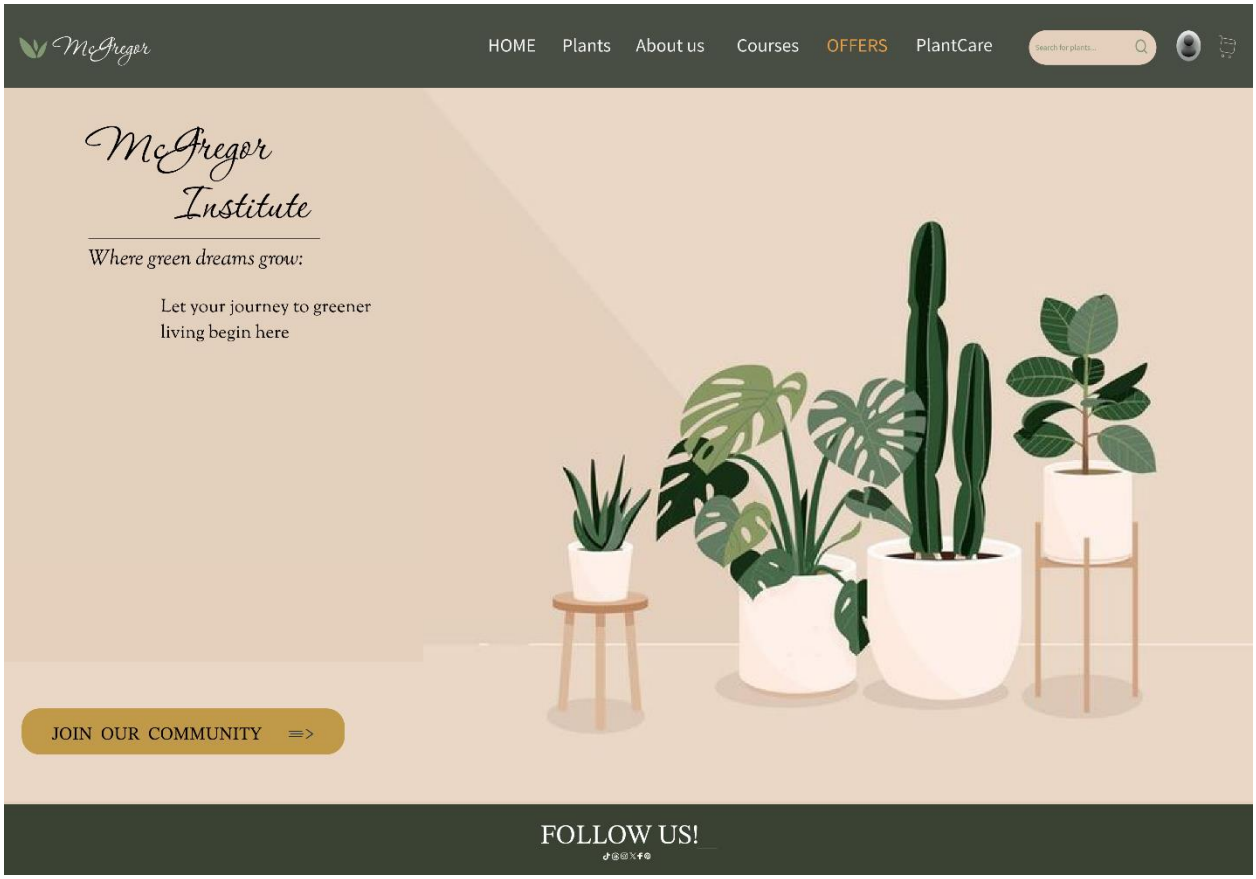


Figure 16:Home page

- Purchase Plant Page For The System

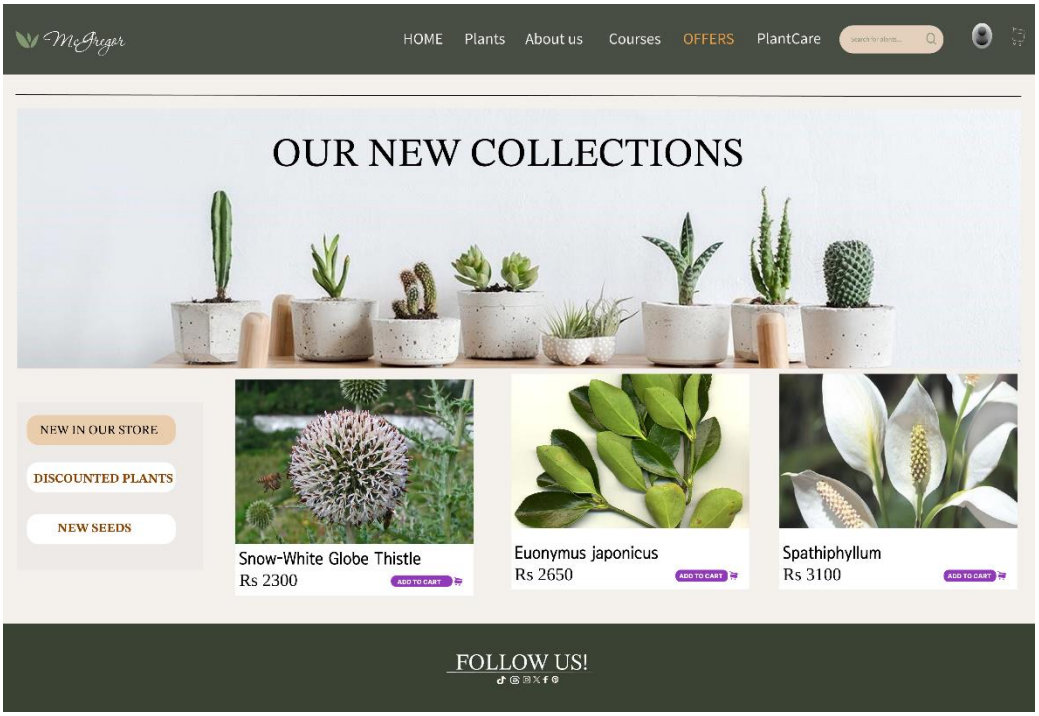


Figure 17:plant purchase page

- Join The Program Page For The System

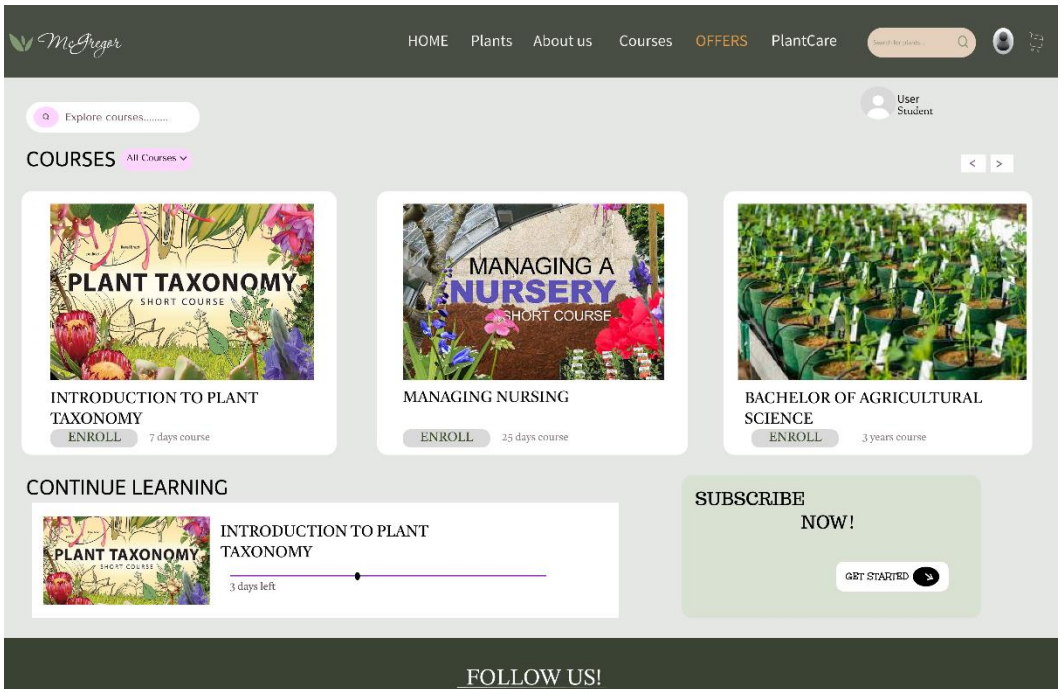


Figure 18:Join the program page

- About Us Page For The System

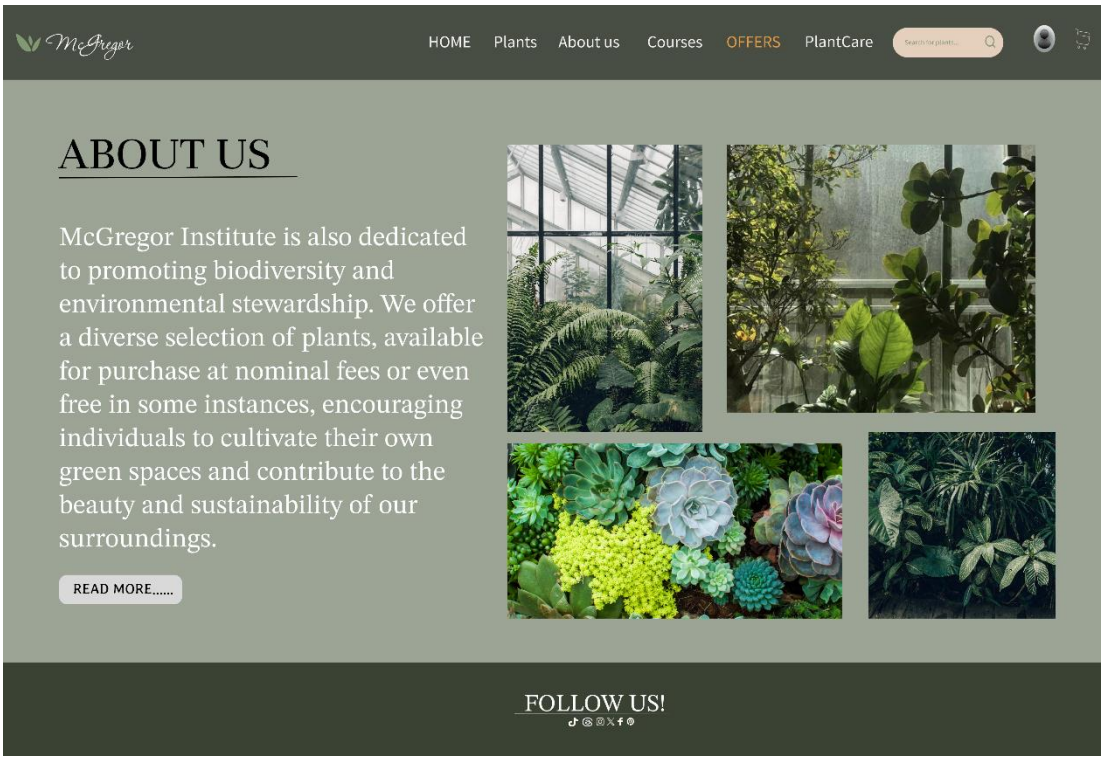


Figure 19:About us page

- Offers Page For The System

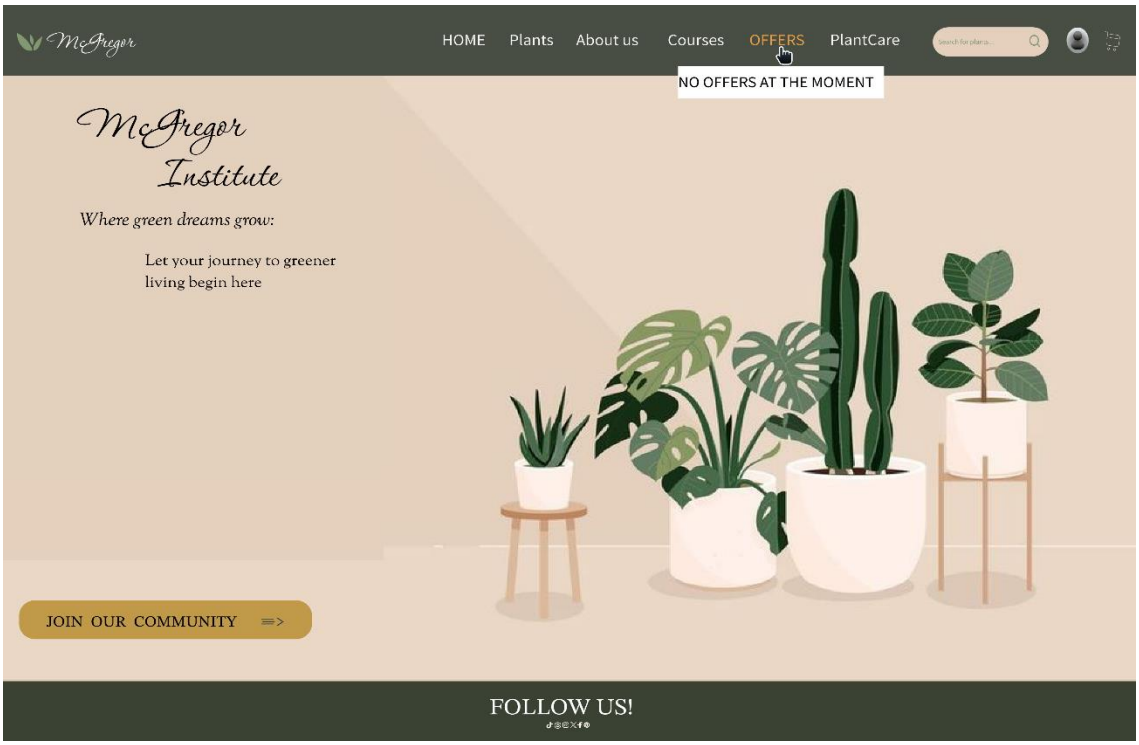


Figure 20:Offer page

- Join Our Community For The System

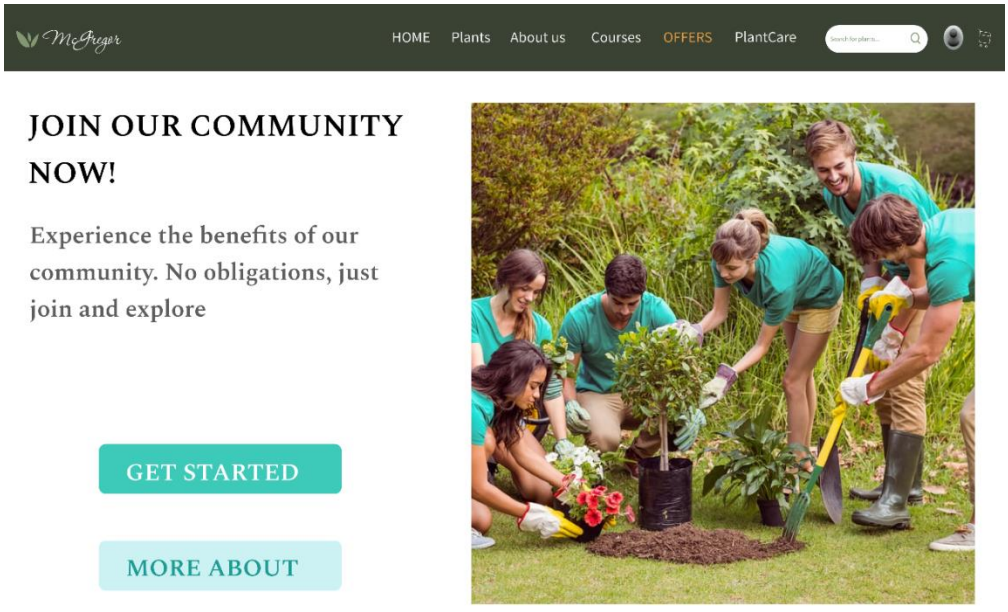
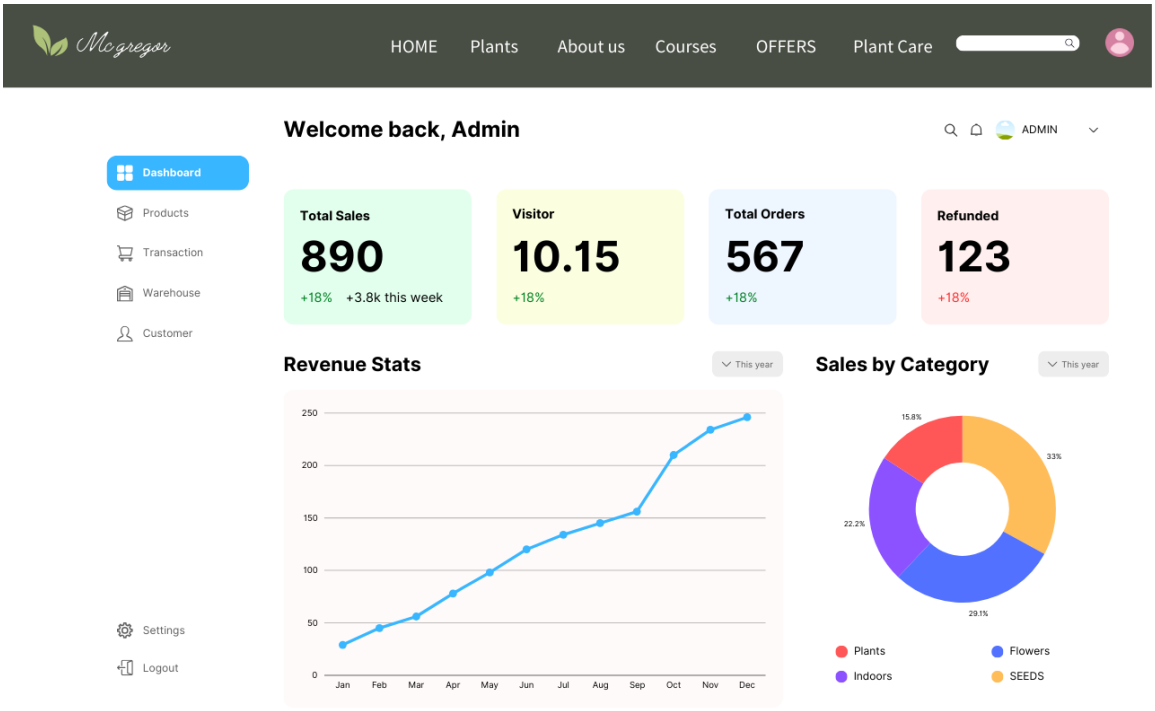
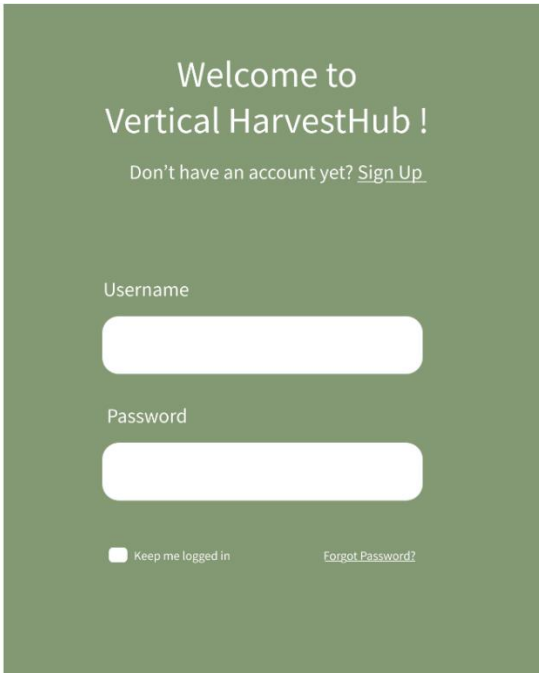


Figure 21:Join the Community page

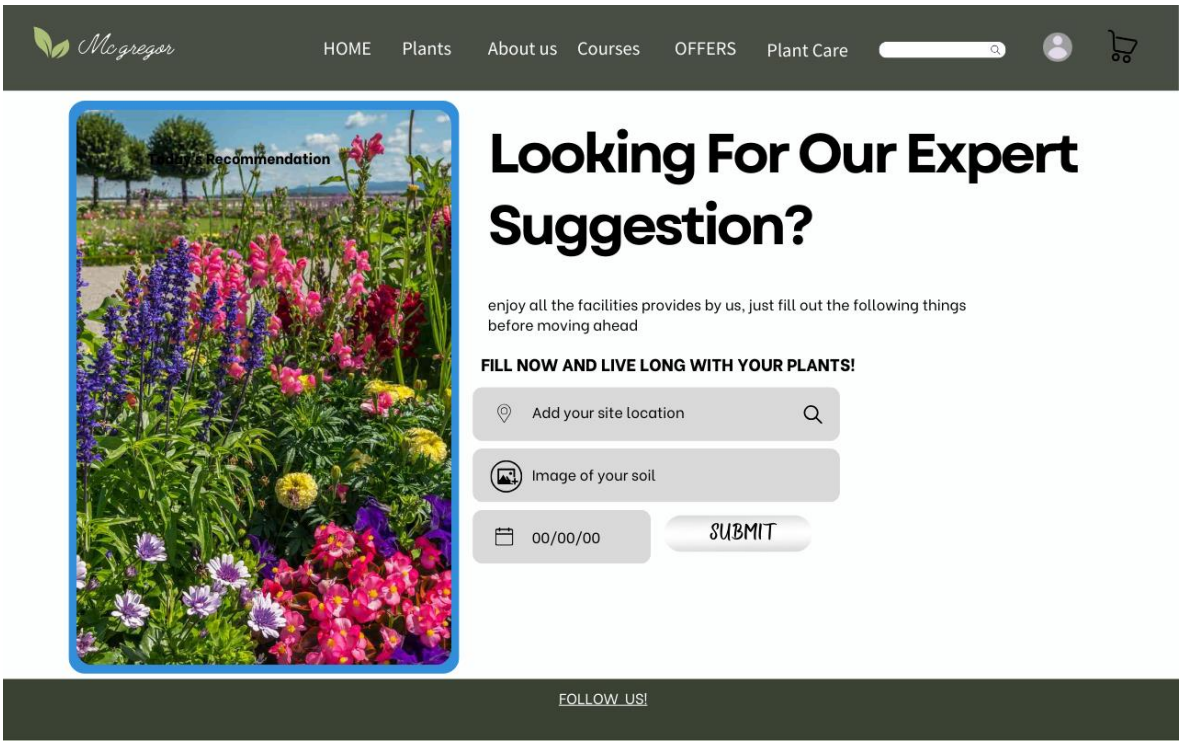
- Admin Home Page For The System



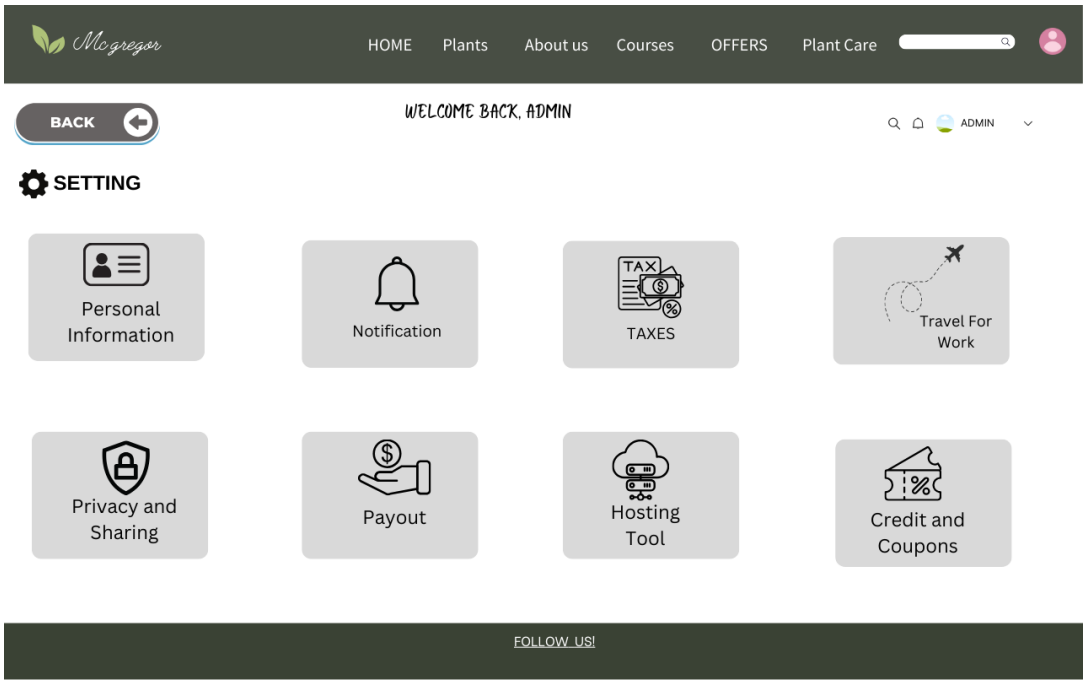
- Login Page For The System



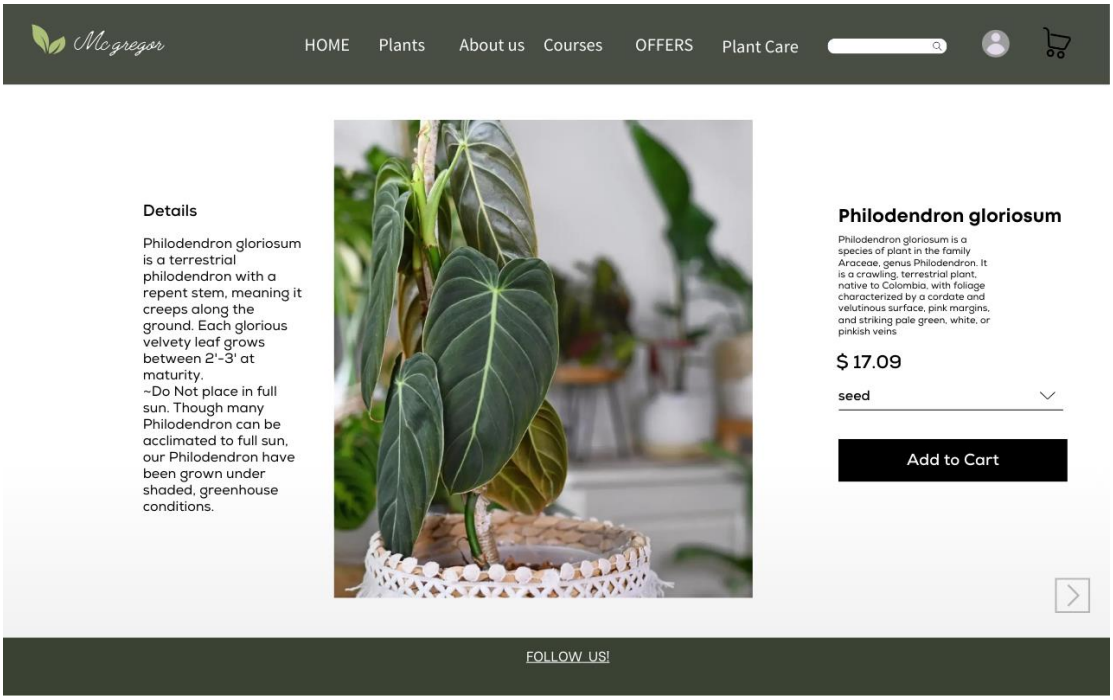
- Plant Care Page For The System
-



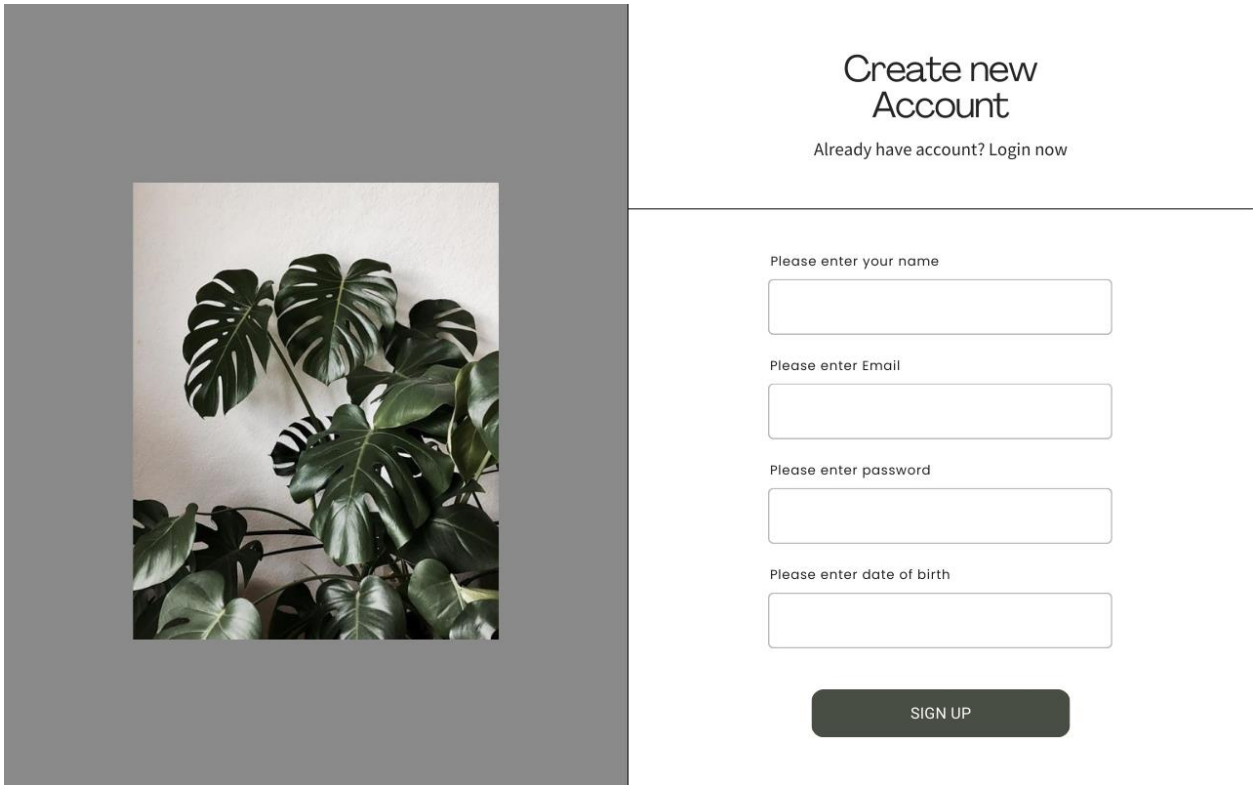
- ADMIN Setting Page For The System



- View Plants Individually



- Register Page Of The System



11. Conclusion

In the end, the McGregor Institute of Botanical Training's endeavor to create an all-inclusive software solution represents a critical turning point in its pursuit of excellence and innovation in botanical teaching and community outreach. The Institute's commitment to creating a dynamic platform that breaks down conventional barriers is clear from its project summary, which includes a wide range of features like community forums and expert suggestions.

In addition to providing short-term certification programs and democratizing access to plant varieties, the Institute seeks to enhance people's knowledge and abilities while fostering a stronger bond between the environment and the community. The Institute guarantees flexibility and response to changing needs by adhering to agile methodology and accepting small adjustments and enhancements throughout the development process.

The McGregor Institute's software project essentially incorporates the organization's three main values: sustainability, community building, and information distribution. The Institute is paving the path for a bright future in Nepal's botanical education sector as it sets new benchmarks for innovation and excellence as it sets out on this transformative journey.

APPENDIX

Bibliography And CITATIONS

geeksforgeeks., 2024. *Collaboration Diagrams | Unified Modeling Language(UML).* [Online]
Available at: https://www.geeksforgeeks.org/collaboration-diagrams-unified-modeling-languageuml/
[Accessed 6 May 2024].

geeksforgeeks., 2024. *What is Agile Methodology?.* [Online]
Available at: https://www.geeksforgeeks.org/what-is-agile-methodology/
[Accessed 6 May 2024].

geeksforgeeks, 2024. *Class Diagram | Unified Modeling Language (UML).* [Online]
Available at: https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams/
[Accessed 5 May 2024].

geeksforgeeks, 2024. *Collaboration Diagrams | Unified Modeling Language(UML).* [Online]
Available at: https://www.geeksforgeeks.org/collaboration-diagrams-unified-modeling-languageuml/
[Accessed 5 May 2024].

geeksforgeeks, 2024. *Sequence Diagrams | Unified Modeling Language (UML).* [Online]
Available at: https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/
[Accessed 5 may 2024].

geeksforgeeks, 2024. *Sequence Diagrams | Unified Modeling Language (UML).* [Online]
Available at: https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/
[Accessed 6 May 2024].

geeksforgeeks, 2024. *Use Case Diagrams | Unified Modeling Language (UML).* [Online]
Available at: https://www.geeksforgeeks.org/use-case-diagram/
[Accessed 5 may 2024].

Martins, J., 2024. *What is a Gantt chart? A guide to mapping your project timelines.* [Online]
Available at: https://asana.com/resources/gantt-chart-basics
[Accessed 5 May 2024].

Raeburn, A., 2024. *Work Breakdown Structure (WBS): What Is It? [2024] • Asana.* [Online]
Available at: https://asana.com/resources/work-breakdown-structure
[Accessed 20 March 2024].