



**CS5002NI Software Engineering**  
**McGregor Institute**  
**20% Group Coursework**  
**AY 2023-2024**

Group Name:			
S.N.	Student Name (Section name: Compulsory)	College ID	University ID
1.	Aashraya Shrestha (L2C21)	NP01CP4S230101	22085481
2.	Anjan Giri (L2C21)	NP01CP4S230104	22085496
3.	Sanskar Karki (L2C21)	NP01CP4S230118	22085610
4.	Shamel Rai (L2C21)	NP01CP4S230135	22085617
5.	Sulav Maharjan (L2C21)	NP01CP4S230112	22085635

*I confirm that I understand my coursework needs to be submitted online via MySecondTeacher 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

1. Introduction .....	1
2. Project Charter .....	2
2.1. Problem Statement .....	2
2.2. Business Case .....	2
2.3. Goal Statement .....	3
2.4. Timeline .....	3
2.5. Scope .....	3
2.6. Team Members .....	4
3. SRS .....	5
3.1. Functional Requirements .....	5
3.2. Non-Functional Requirements .....	7
3.2.1. Design and Implementation Constraints .....	7
3.2.2. External Interfaces Required .....	7
3.2.3. Other non-functional requirements .....	8
3.3. Goals of Implementation .....	10
4. Group Task .....	11
4.1. Environmental Model Specification .....	11
4.1.1. Context Diagram .....	11
4.1.2. Data Flow Diagram Level 1 .....	12
4.1.3. Data Flow Diagram Level 2 .....	13
4.2. Internal Model Specification for the System .....	15
4.2.1. Entity Relationship Diagram (ERD) .....	15
4.2.2. Data Dictionary .....	17
4.2.3. Process Specifications (Pspecs) .....	19
4.3. Design Specification .....	24
4.3.1. Structure Chart .....	24
4.4. Assignment Diary .....	28
4.4.1. Assumptions .....	28
4.4.2. Member Responsibilities .....	31
4.4.3. Meeting Minute (Group Meetings) .....	32
5. Individual Tasks .....	33

5.1.	Make Payment .....	33
5.1.1.	Environmental Model Specification.....	33
5.1.2.	Internal Model Specification.....	34
5.1.3.	Design Specification .....	36
5.2.	Purchase Plant.....	39
5.2.1.	Environmental Model Specification.....	39
5.2.2.	Internal Model Specification.....	40
5.2.3.	Design Specification .....	42
5.3.	Report Preparation.....	45
5.3.1.	Environmental Model Specification.....	45
5.3.2.	Internal Model Specification.....	46
5.3.3.	Design Specification .....	48
5.4.	Join the Program.....	50
5.4.1.	Environmental Model Specification.....	50
5.4.2.	Internal Model Specification.....	51
5.4.3.	Design Specification .....	53
5.5.	Take Certification Exam .....	56
5.5.1.	Environmental Model Specification.....	56
5.5.2.	Internal Model Specification.....	57
5.5.3.	Design Specification .....	59
6.	Summary.....	62
7.	References.....	63

## Table of Figures

Figure 1: Context Level Diagram.....	11
Figure 2: Level 1 DFD .....	12
Figure 3: Level 2 DFD 1 .....	13
Figure 4: Level 2 DFD 2 .....	13
Figure 5: Level 2 DFD 3 .....	14
Figure 6: Entity Relationship Diagram .....	16
Figure 7: Structure Chart Part 1 .....	25
Figure 8: Structure Chart Part 2 .....	26
Figure 9: Structure Chart Part 3 .....	27
Figure 10: Structure Chart Full .....	27
Figure 11: Make Payment Context Level Diagram:.....	33
Figure 12: Make Payment Level 1 DFD .....	34
Figure 13: Make Payment Level 2 DFD .....	35
Figure 14: Make Payment Structure Chart.....	36
Figure 15: Context Level Diagram.....	39
Figure 16: Purchase Plant Level 1 DFD .....	40
Figure 17: Purchase Plant Level 2 DFD .....	41
Figure 18: Purchase Plant Structure Chart.....	42
Figure 19: Report Preparation Context Level Diagram.....	45
Figure 20: Report Preparation DFD Level 1 .....	46
Figure 21: Report Preparation DFD Level 2 .....	47
Figure 22: Report Preparation Structure Chart.....	48
Figure 23: Join the Program Context Diagram .....	50
Figure 24: Join the Program Level 1 DFD .....	51
Figure 25: Join the Program Level 2 DFD .....	52
Figure 26: Join the Program Structure Chart.....	53
Figure 27: Take Certification Exam Context Diagram .....	56
Figure 28: Take Certification Exam Level 1 DFD .....	57
Figure 29: Take Certification Exam Level 2 DFD .....	58
Figure 30: Take Certification Exam Structure Chart.....	59

Table of Tables

Table 1: Timeline..... 3

Table 2: Process A ..... 19

Table 3: Process B ..... 20

Table 4: Process C..... 21

Table 5: Process D..... 22

Table 6: Process E ..... 23

Table 7: Individual Contribution..... 31

Table 8: Meeting Log..... 32

## 1. Introduction

The provided coursework is the first coursework of the module Software Engineering (Module code: CS5002NI) which accounts for 20% of our module grades. The coursework requires us to make a system as per the requirements and needs based on the provided scenario. For this, the coursework is assigned to group of 4-5 members. The main purpose of this coursework is to provide us the experience of working in group to come up with the solution for a particular problem and to help us learn various aspects of Software Engineering.

Based on the provided scenario, we know that McGregor Institute of Botanical Training has been operating for almost 7 years in Nepal. It has been providing courses specializing in agriculture and horticulture. But in recent years, the sudden increase in people showing interest towards agriculture has led the institute to make plans for introducing short-term certification courses related to horticulture. Along with this, the institute is planning to sell different varieties of plants for minimal fee and even is free in some cases. The institute also wants to build a community of people with interest in plants and also create a platform where such individuals can organize programs to share their ideas and to protect endangered plants and forests and can also be used to post queries which will be answered by the experts.

In accordance with the scenario, a system is to be made fulfilling all the given requirements along with proper and simple management to make the institute run smoothly and to provide efficient learning and interacting platform for the plant enthusiasts. Different operations should be carried out by the system in order to meet the required goal. For this purpose, different tasks are divided among the group members to complete the task in certain provided time.

Different aspects of software engineering are used for completing this coursework which includes Context level Diagram, Data Flow Diagrams (DFDs) which provide the environmental model specifications of the system. For the internal model specification of the system, Entity Relationship Diagram (ERD) should be made which is necessary to complete the system specifications along with data dictionary. An Upper-Level Structure Chart should also be made to design the entire required system.

## **2. Project Charter**

### **2.1. Problem Statement**

McGregor Institute of Botanical Training has been providing courses for past 7 years and now wants to expand the scale of courses they provide. They also want to sell various plants while also looking to provide a platform for plant enthusiasts to express their view and discuss it among fellow plant enthusiasts.

As the institute now wants to do much more than they have been doing for past 7 years, the system used by them all these years is not sufficient to meet the demand of the institute and is unable to function or carry out the operations according to the new plans of the institute. The lack of proper system of the institute to carry out different operations may lead to failure of the plans made and might as well affect the whole institute.

### **2.2. Business Case**

The Ireland based “McGregor Institute of Botanical Training” located at Godawari, Lalitpur which has been operating for almost 7 years. This institute has been providing different postgraduate and undergraduate courses affiliated to Dublin City University that specializes in agriculture and horticulture. With the number of people growing interested in the field of agriculture and horticulture this institute has planned to initiate short term certification associated with agriculture and horticulture to satisfy the increasing stipulation.

The McGregor institute aspires to sell varieties of plants at minimal cost or in some cases, for free. Not only that, the institute also wants to encourage the individuals to participate in this field and create a vibrant community of people who are interested in agriculture and horticulture by developing an online platform. The platform will perform as a bridge where individuals can discuss, exchange ideas and knowledge with other enthusiasts. By providing features such as registration for new applicants, joining into the programs, taking certification exams, purchasing plants etc., they are encouraging everyone to join the platform. The individuals can also seek advice from the expert, post queries, and collaborate with their fellow colleagues. This will help the McGregor institute reach out more to plant enthusiast and also achieve their goals.

### 2.3. Goal Statement

The main goal is to introduce and connect individuals who are plant enthusiasts for the purpose of discussing ideas and planning to protect rare plants and forests. Alongside providing people with a range of short-term certifications, the goal is to sell varieties of plants at a minimal fee and even for free during special occasions. The firm also intends on being a platform that is open for any queries that people may have which is directly answered by experts.

### 2.4. Timeline

Task	Completion Date
Project Charter, Corrections	December 10, 2023
DFD	December 15, 2023
ERD	December 20, 2023
Individual Tasks	December 23, 2023
Assignment Diary	December 28, 2023
Process Specification	January 2, 2024
Correction and Finalization	January 4, 2024

Table 1: Timeline

### 2.5. Scope

The development of an online platform for horticulture has to provide many opportunities some of them are listed below as scope: -

- i. Source to reach-out more interested individuals.
- ii. A community for plant enthusiasts will be formed where people can unite and discuss ideas.
- iii. People will know more about domain of culture.
- iv. Chance for interested individuals to get certification related floriculture.
- v. People can post their queries to be answered by experts which will make the platform more knowledgeable and relevant.
- vi. Programs can be organized under supervision to protect rare plants.
- vii. Individual could also acquire varieties of plants at low price or for free.



## **2.6. Team Members**

The group of people involved in this project for proper execution are: -

- i. Aashraya Shrestha
- ii. Anjan Giri
- iii. Sanskar Karki
- iv. Shamel Rai
- v. Sulav Maharjan

### 3. SRS

The Software Requirement Specification (SRS) Format is a comprehensive document that defines the precise requirements crucial for the successful development of a software system. These requirements comprise of both functional and non-functional aspects, prepared according to the specific software's scope, and intended environment. Effective communication between clients and contractors is imperative, ensuring a comprehensive grasp of customer needs. (GeeksforGeeks, 2023)

An SRS outlines the specifics of a software product or program, detailing its functionalities within a designated context. Its purpose varies based on the given requirements. When composed by a client, the SRS articulates user needs and expectations. Conversely, when crafted by a developer, it assumes a contractual role, aligning the understanding between the customer and developer for successful project execution. (Javatpoint, 2021)

#### 3.1. Functional Requirements

- a. **Registration:** Admin as well as the user will have the ability to use this feature to register an account for themselves and submit their personal data and information to create a profile agreeing to all the legal terms and conditions of the platform.
- b. **Enrollment Function:** After the registration is complete, users will have the ability to choose whether they want to register for undergraduate program, postgraduate program or for short-term certification according to their requirements. The system will also integrate a secure payment system and will also provide financial aids or scholarship options. After enrollment, the functionality will provide the user with the course materials from the university and send a confirmation email or notification to the user for successful enrollment.
- c. **Feature Control:** Different individuals will get different features according to their roles in the institution. For example: A student can register their account, take an online certification examination, purchase varieties of plants, ask for different queries, etc., whereas an employee can check the examination answer, provide feedback to the queries and many more.

**d. Purchase and Invoice:** When a user purchase any plant, he/she can have different methods of payment whether it is cash, card, or digital method of payment. When the transaction has been made an invoice would be generated and stored in a database. So, even if an issue occurs in the transaction period, the transaction history can be checked.

**e. Recommendation Functionality:** Users will be able to submit their request for recommendation from the experts in this functionality. It will provide a form where users can provide specific detail about their problem. Many horticulture experts will be available here to solve their problem.

**f. Report generation:** Here, the administrator can input the employee data, user data and financial data. It will check the validation of the data and ensure the accuracy of the data and notify the administrator if any errors may have occurred to generate a final report. It will also provide visual graphical representation of the data enhancing the visual appearance of the report.

**g. Take Certification Exam:** The system will provide users with the option to take mock test practice before the actual examination. The functionality will also provide the feedback on the performance in the mock test which will help the user to improve in the areas that they are weak at. It will also implement a system to verify whether the user has fulfilled the necessary prerequisites before giving the certification examination. It will also provide the feature to retake the examination if the user fails to pass the examination.

**h. Form System:** Users will be able to communicate with each other about the plants. The system will allow the user to create a new post where they can start discussion. Users can also comment on other posts and contribute to ongoing discussion. Voting system will also be implemented to let the users upvote and downvote the post and comments. Search functionality will also be provided so that users can access specific topics quickly.

**i. Notification:** Users are allowed to set notifications preference on what activity or events they want to be notified. The system will implement a real time notification for important events like messages, updates or mentions. The system will also include a notification badge that will indicate if the notification has been read or not.

## **3.2. Non-Functional Requirements**

### **3.2.1. Design and Implementation Constraints**

- i. With the power of technologies and innovative minds, the system can be designed using advanced web development tools like Python, C#, JavaScript, etc.
- ii. There should be a separate system for Mobile Phone compatibility, developed using Swift and Kotlin.
- iii. The system should implement secure user authentication with authorization mechanisms while also using HTTPS for secure data transmission.
- iv. There should be backup servers on the go for any mishaps while also providing a regular system backup.
- v. The system should have a simpler, clean UI focusing more on user satisfaction with a good response time by proper code optimization.

### **3.2.2. External Interfaces Required**

#### **Hardware Interface**

- i. The system should be compatible across various devices such as desktops, laptops, phones, tablets, etc.
- ii. The system should not take more resource as it may lag in some device resulting in poor user experience.

#### **Software Interface**

- i. The system should have a feature to chat with examinees for any sort of confusions or tips during the exam. They can also use it to talk with any other plan enthusiast.
- ii. The system should hold the record for the type of plants present in the garden with their location in section which will be easier for not only tourists but for everyone to locate the plant.

**User Interfaces**

- i. The system should be integrated with proper security feature. As leaking of any user data might be dangerous, they all should be encrypted properly.
- ii. The system should have a feedback section where users can comment on about how well the system is working.

**Communication Interfaces**

- i. The system should ensure secure data transfer between client and server using HTTPS.
- ii. The system should feature a notification bar which alerts users about on-going events, exam schedule, visiting schedule for tourists, etc.

**3.2.3. Other non-functional requirements****Customer Support**

- i. The user can contact customer support at any time for proper guidelines or any other difficulties.
- ii. The users can contact customer support when they have difficulty finding any specific plant or if they get lost.

**Language Support**

The system offers multiple Languages Support which helps people have the app in their native language for better understanding

**Balancing Load**

The system should be able to distribute incoming clout to multiple servers which makes app stable also ensuring optimal performance with resource utilization.

**Error Handling**

The system should provide a clear error message on encounter with also a solution to solve those errors.

**Maintenance**

The system should have a maintenance break at least once in a week to ensure everything is working fine while also informing users beforehand about maintenance with its start and end time.

### **3.3. Goals of Implementation**

- i. Gather and help plant enthusiasts connect with each other.
- ii. Discuss the ideas and create programs to save the rare and endangered plants and forest.
- iii. Teach courses related to horticulture.
- iv. Sell a variety of plants at a minimal fee or even for free during special occasions.
- v. Provide individuals with short-term certification courses related to a variety of plants.
- vi. To create a platform where people can post their queries and confusions which are then answered directly by experts.

## 4. Group Task

### 4.1. Environmental Model Specification

#### 4.1.1. Context Diagram

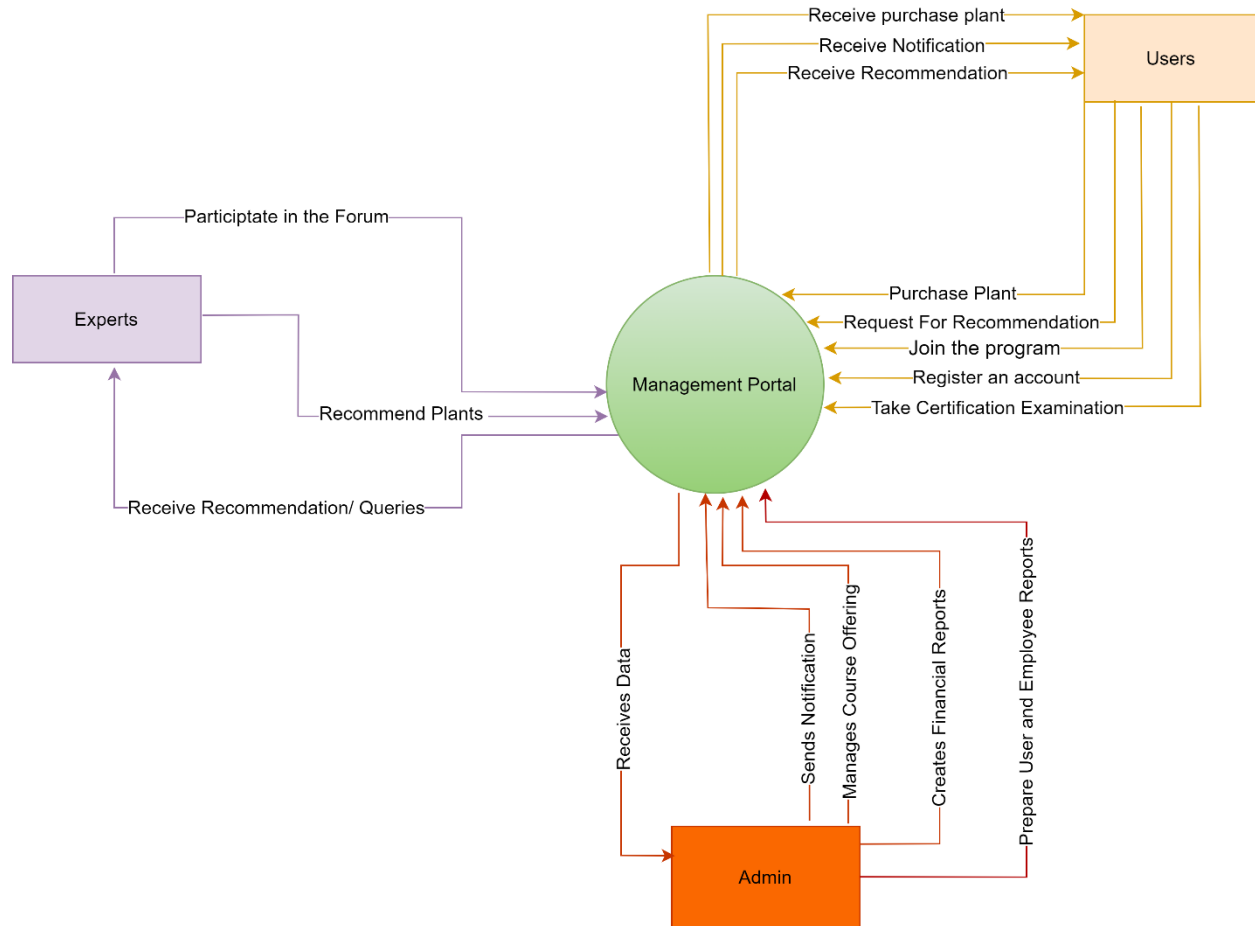


Figure 1: Context Level Diagram



## 4.1.2. Data Flow Diagram Level 1

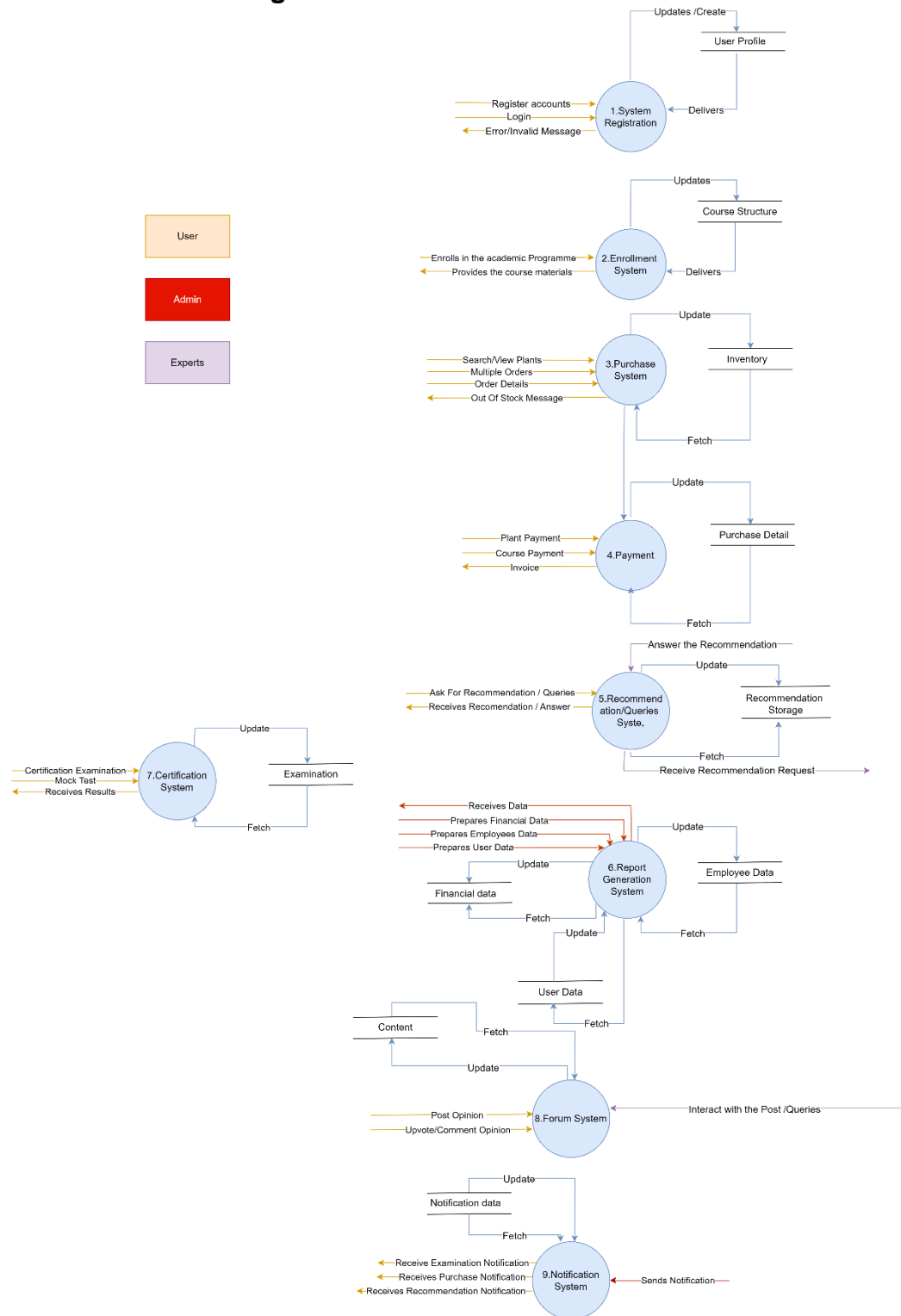


Figure 2: Level 1 DFD

### 4.1.3. Data Flow Diagram Level 2

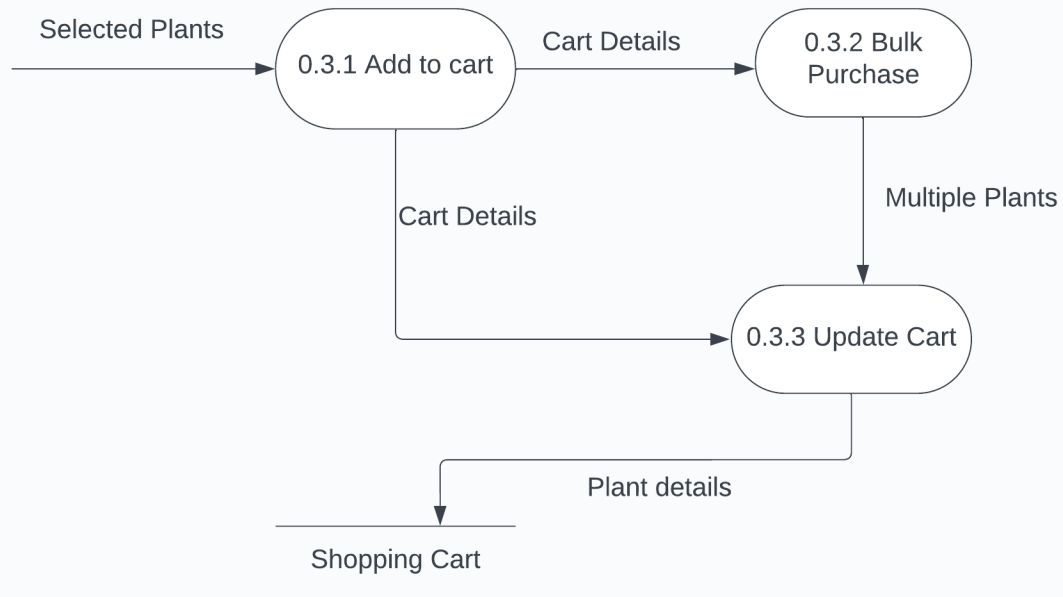


Figure 3: Level 2 DFD: 1

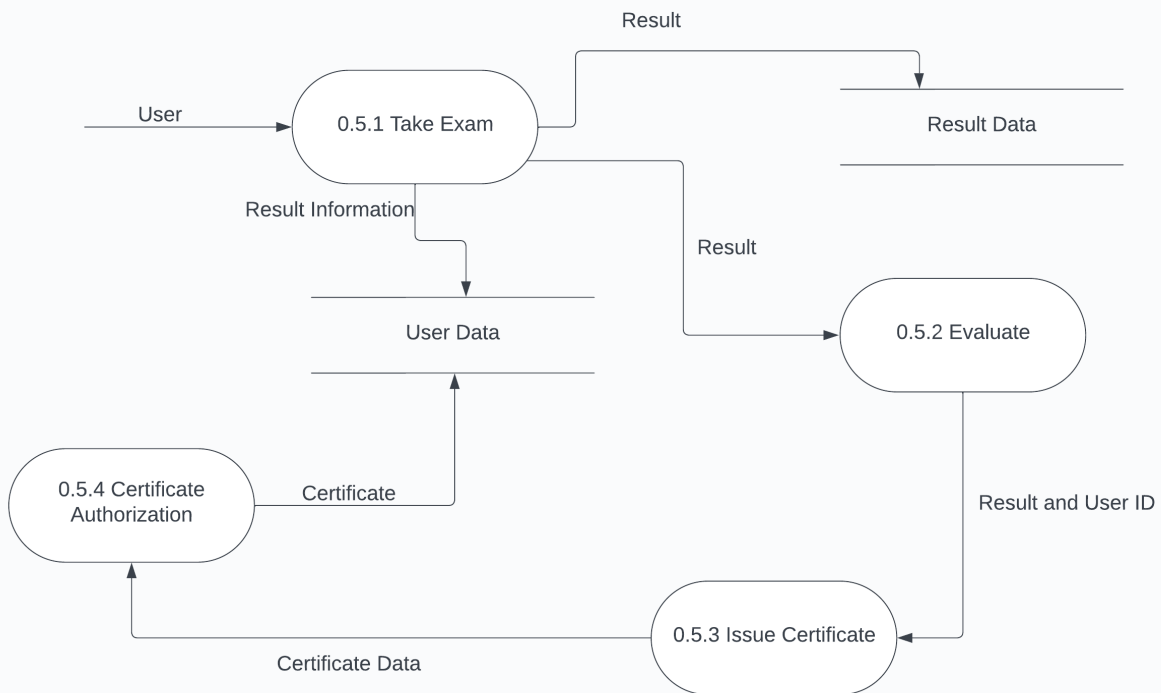


Figure 4: Level 2 DFD: 2

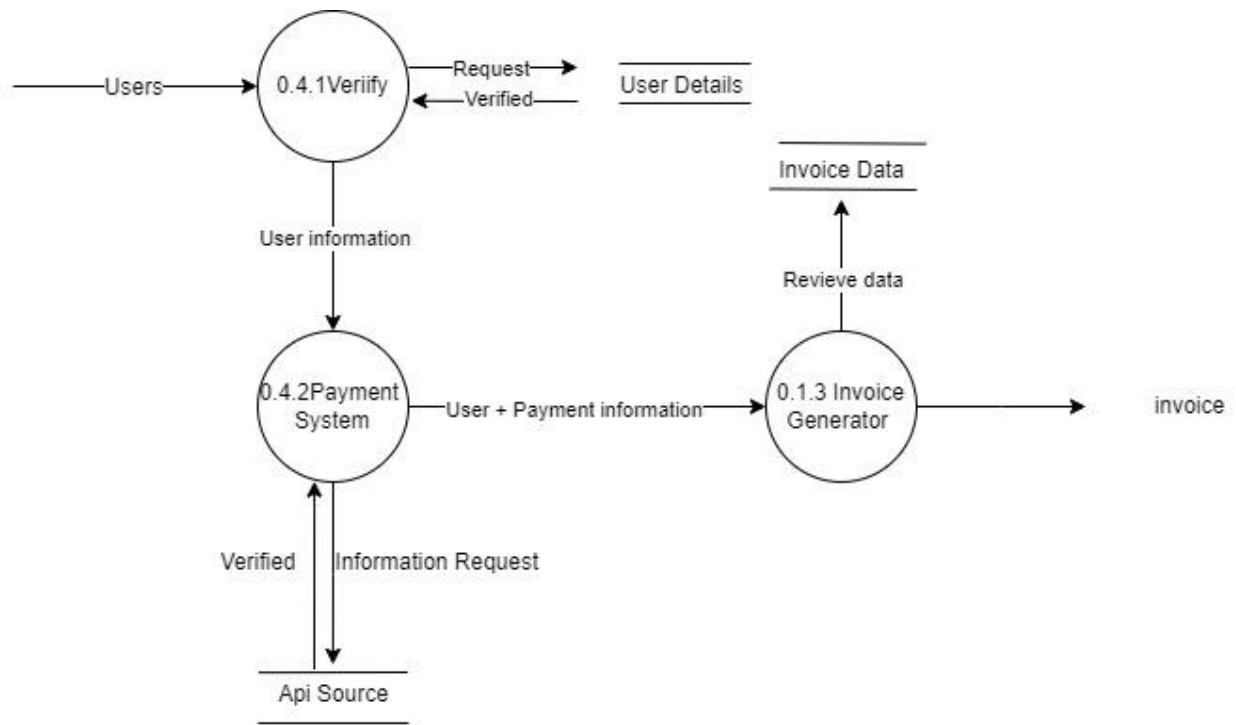


Figure 5: Level 2 DFD: 3

## 4.2. Internal Model Specification for the System

### 4.2.1. Entity Relationship Diagram (ERD)

An Information Technology (IT) system's relationship between individuals, things, locations, ideas, and events are shown graphically in an entity relationship diagram (ERD), often called an entity relationship model. A relational database's basis and a means of defining business processes are both provided by the data modeling approaches employed in an ERD. (Techtarget, 2019)

An ERD contains:

- i. **Entity:** Any unique, distinguishable, and independent thing is an entity. It refers to entities that are noteworthy in and of themselves, such as people, groups, systems, data points, or even specific system components. (Techopedia, 2014)
- ii. **Attribute:** An entity's property or characteristic is called an attribute. Any quantity of attributes can be present in an entity. One of the attributes is considered to be the primary key. Simple, Composite, Single-valued, Multi-valued, and Derived attributes are the six categories of an attributes. Another attribute is their which is rarely utilized i.e. Complex Attribute. (Geeksforgeeks, 2023)
- iii. **Relationship:** Establishing links between tables is crucial for maintaining data integrity and organizing data while building a database schema. The links between tables that define the relationship between the data in one table and the data in another are called relationships. Within a database model, relationships can be classified as one-to-one, one-to-many, or many-to-many. (Datensen, 2023)

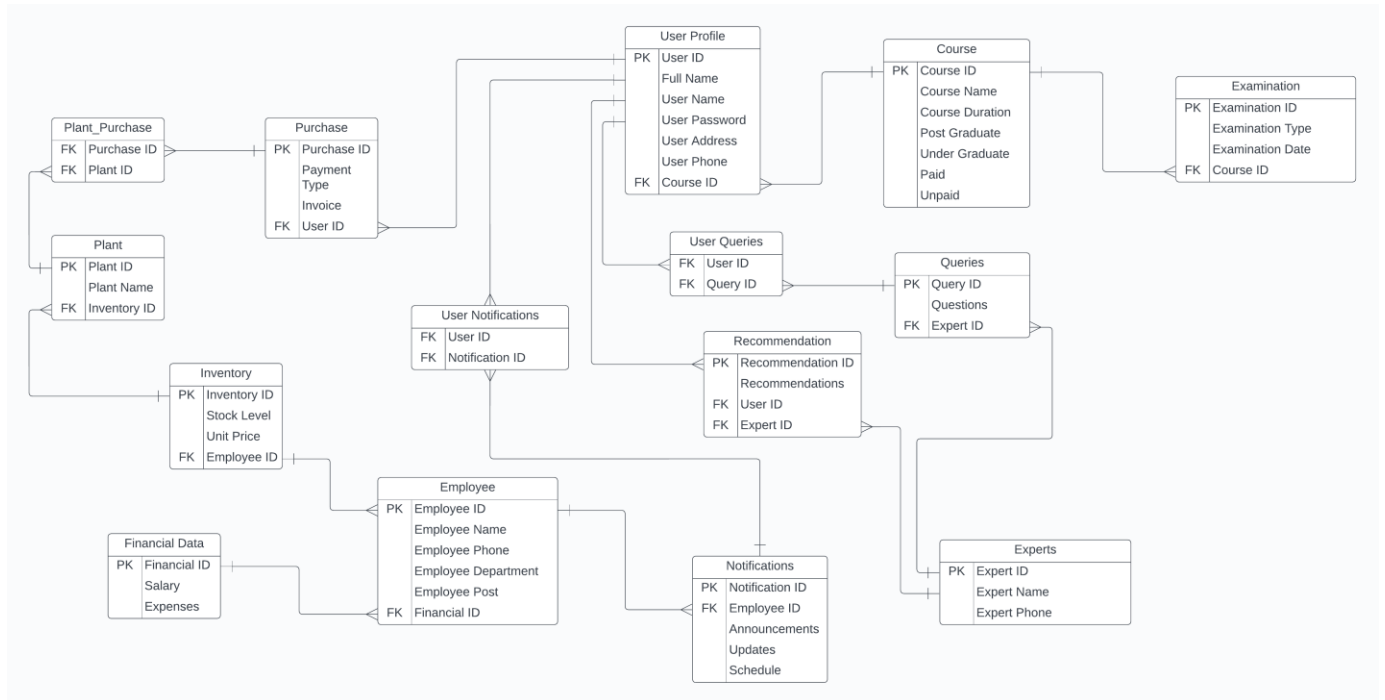


Figure 6: Entity Relationship Diagram

#### 4.2.2. Data Dictionary

The concept of Data Dictionary is used to describe the names, definitions, and attributes of the data elements that have been captured or used in databases, information systems, or research projects. It offers guidelines on interpretation, acceptable interpretations, and representation in addition to describing the goals and implications of data items inside the framework of a project. Metadata on data items is also provided via a data dictionary. The scope, properties, and application and usage guidelines of data items may all be defined with the help of the metadata that is provided in a data dictionary. Data dictionaries help to prevent inconsistent data in the project, establish project-wide conventions, Provide easy access to data for analysis etc. (UC Merced Library, 2023)

Register admin = Command

Login = Command

Register Users = Command

Register experts = Command

Purchase = Command

Join Program = Command

Generate Report = Command

Post Form = Command

Take Examination =Command

Send Notification = Command

User Profile = Full Name + User Name + User Password + User Address + User Phone  
+ [Course ID]

Course = Course Name + Course Duration +[Postgraduate/ Undergraduate]

Plant Purchase = Purchase ID + Plant ID

Purchase = Payment Type + Invoice + [Paid/Unpaid] + User ID

Plant = Plant Name + Inventory ID

Inventory = Stock Level + Unit Price + Employee ID

Employee = Employee Name + Employee Phone + Employee Department + Employee Post

Financial Data = Salary + Expenses + Employee ID

User Queries = {User ID + Query ID}

Query = Questions + Recommendations

Examination = Examination Type + Examination Date + {Course ID + Expert ID}

User Notifications = {User ID + Notification ID}

Notifications = Announcements + Updates + Schedule

Experts = Expert Name + Expert Phone

### 4.2.3. Process Specifications (Pspecs)

#### 4.2.3.1. Process A

<b>Number</b>	0.3.1
<b>Name</b>	Add to cart
<b>Description</b>	This process prepares information about the different items that are to be purchased by the user in a single purchase
<b>Input data flow</b>	Selected plant
<b>Output data:</b>	Stores the data in the cart and displays to the user
<b>Process logic</b>	User can have one or multiple plant orders User can also delete the unwanted order/s

Table 2: Process A



**4.2.3.2. Process B**

<b>Number</b>	0.4.1
<b>Name</b>	Enroll
<b>Description</b>	This process prepares the information about the enrollment of the users
<b>Input data flow</b>	User information
<b>Output data flow</b>	Valid user information, user enrolled
<b>Process logic</b>	Ask for user credentials Checks for the validation of user credentials Stores in the enrollment data, student data

*Table 3: Process B*

**4.2.3.3. Process C**

<b>Number</b>	0.5.3
<b>Name</b>	Issue Certificate
<b>Description</b>	This process prepares the information on whether to issue the certificate for the user or not
<b>Input data flow</b>	User information, Examination
<b>Output data flow</b>	Result, certificate
<b>Process logic</b>	Takes user credentials Checks the user credential to know if they are qualified or not If qualified, they can take examination and grant certificate

*Table 4: Process C*

**4.2.3.4. Process D**

<b>Number</b>	0.2.3
<b>Name</b>	Proceed to verify payment and order
<b>Description</b>	This process prepares the information about the verification of the payment process and the order
<b>Input data flow</b>	Order detail
<b>Output data flow</b>	Notification
<b>Process logic</b>	<p>The system takes the order details from the user.</p> <p>The system checks if the item is available or not in the inventory.</p> <p>If the item is available, the system updates the inventory and if it's not available a notification is sent to the user</p>

*Table 5: Process D*

**4.2.3.5. Process E**

<b>Number</b>	0.3.1
<b>Name</b>	Report Preparation
<b>Description</b>	This process prepares the information about how the report is generated by the system
<b>Input Data flow</b>	User detail, request
<b>Output Data flow</b>	Report structure
<b>Process logic</b>	The user provides the raw data The data is the sent for verification After verification the data is sent in the report generator and is stored in report Admin sends the report to the requested user.

*Table 6: Process E*

### **4.3. Design Specification**

#### **4.3.1. Structure Chart**

Structure Chart represent hierarchical structure of modules. It breaks down the entire system into lowest functional modules, describe functions and sub-functions of each module of a system to a greater detail. It partitions the system into black boxes (functionality of the system is known to the users but inner details are unknown). Inputs are given to the black boxes and appropriate outputs are generated. Modules at top level called modules at low level. Components in structure chart are read from top to bottom and left to right. When a module calls another, it views the called module as black box, passes necessary parameters and receives results. (Geeksforgeeks, 2024)

Structure chart aims to visually represent the logical process identified in the design. The exact notation may vary, but typically while making a structure chart, an inverted tree structure is used, with boxes representing major logical actions. (Sciencedirect, 2003) It is a visual hierarchy of the modules or components of a software system and their relationships. Its main purpose is to provide a clear and visual representation of the interaction of the various elements within the software, making it easier to understand the composition and connections of the system.

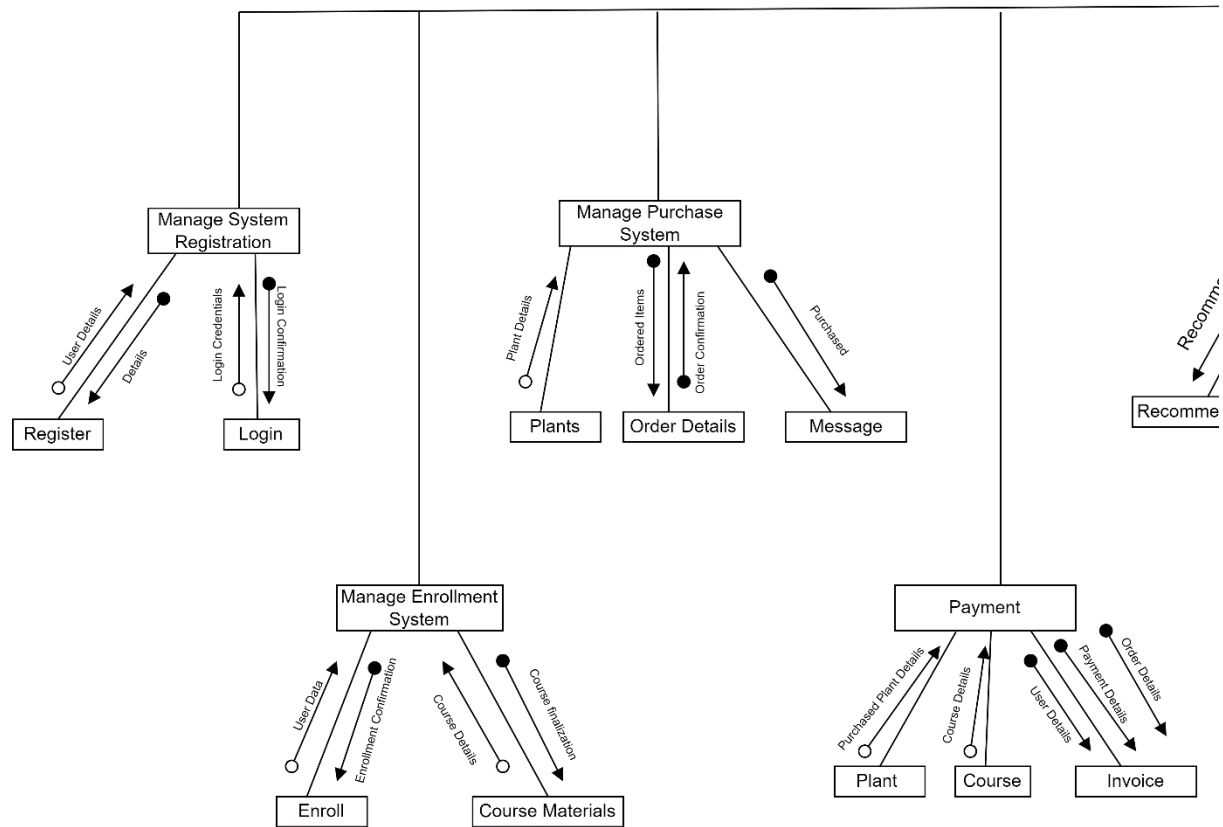


Figure 7: Structure Chart Part 1

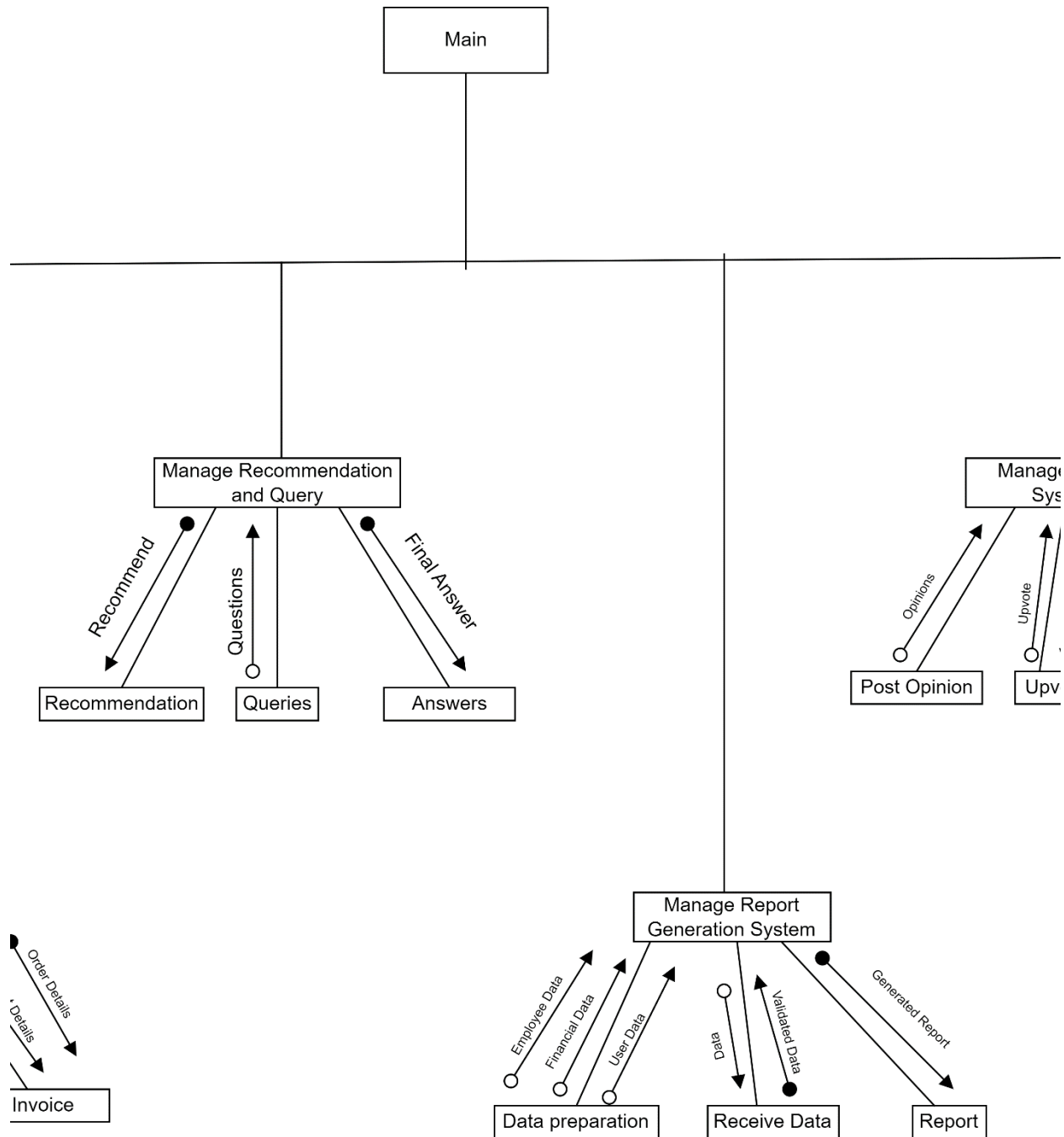


Figure 8: Structure Chart Part 2

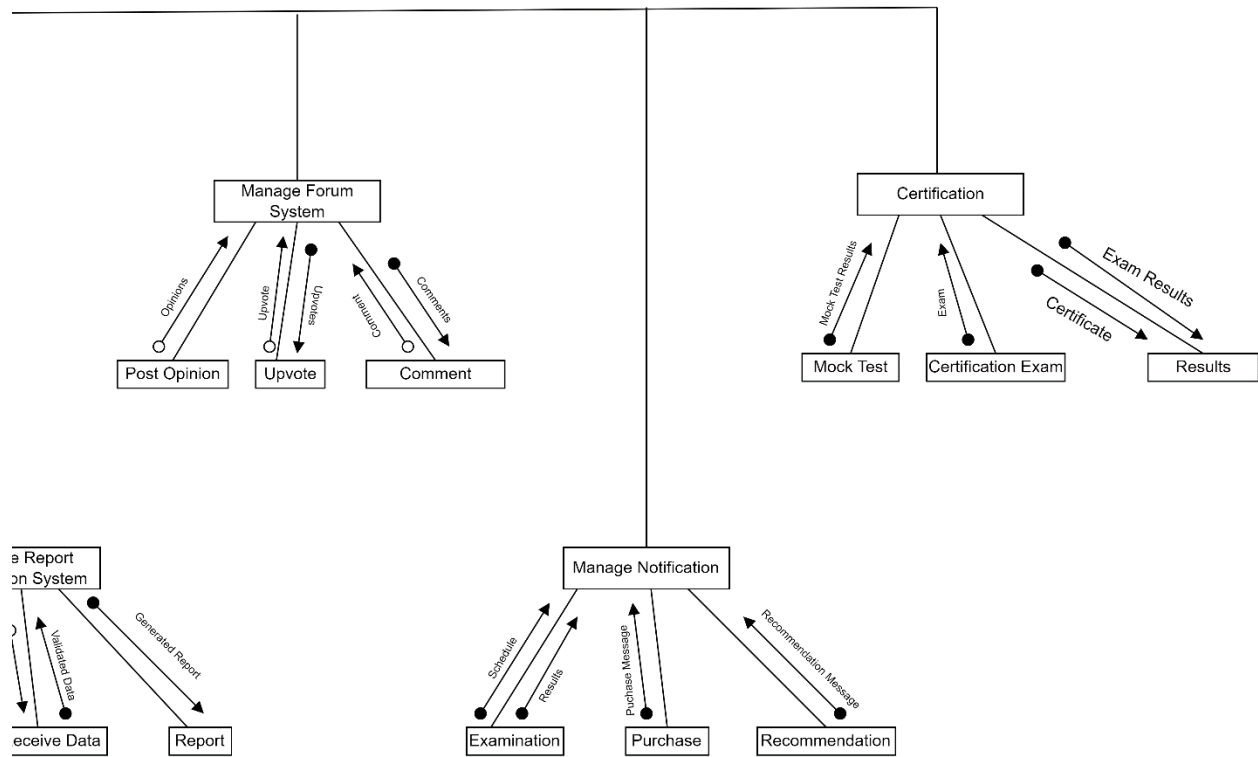


Figure 9: Structure Chart Part 3

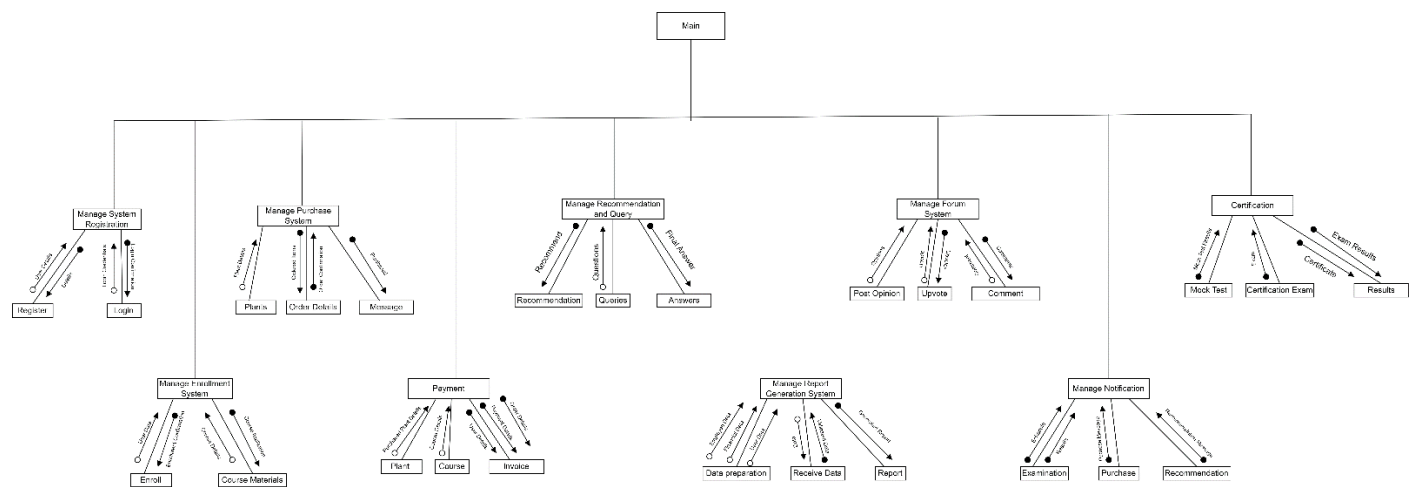


Figure 10: Structure Chart Full



## **4.4. Assignment Diary**

### **4.4.1. Assumptions**

It is crucial to establish various specifications to complete the system design of the McGregor Institute of Botanical Garden. For this, many assumptions are to be made by all the members of the group, which meets the requirements of the system.

Stated below are the assumptions we made for storing data: -

1. User Profile: This database stores the details of the user which includes User ID, Full Name, Address, Contact, Username, and Password.
2. Course Details: This database stores the details about the course. It provides Course ID, Course Name, Course Duration, Post Graduate, Under Graduate, Paid and Unpaid courses.
3. Examination: This section of the database contains details of the examination which provides the Exam ID, Exam Type, and Examination Date.
4. Queries: The questions and doubts of the users are recorded within this system along with the recommendations of the experts provided to the user.
5. Experts: Following the queries, this database stores the details of the experts. It gives Expert ID, Expert Name, and Expert Phone.
6. Employee Details: This database includes the Employee ID, Employee name, Contact Information, Department and Post.
7. Financial Data: The Financial ID, Salary of the employee, and the expenses of the organizations are recorded within this database.
8. Inventory: This database stores the Inventory ID, Stock Level and Unit Price.
9. Purchase Details: This database contains Purchase ID, Payment Type, Invoice, and the Total Amount.
10. Plant: The plant section of the database includes the Plant ID, Plant Name, and Plant Type.
11. Notification: This database contains Notification ID, Announcements, Updates, and Schedules.

To develop the above database, the following assumptions are made from the provided specifications listed below: -

1. A user can enroll in only one course at a time.
2. A single course can have multiple students.
3. The Full Name and the Username (name displayed in the system) of the user can differ as per the preference.
4. The user can enroll in either under-graduate or post-graduate course.
5. The user can also get enrolled for a paid or an unpaid course depending on the situation.
6. The user must be enrolled to a course in-order-to attend the examination.
7. A course can have multiple examinations.
8. A user can post their queries for the experts to answer.
9. An expert can answer multiple queries.
10. The experts can provide recommendations to the users based on the requirements.
11. A user can get multiple notifications and a single notification can be received by multiple users.
12. An employee can send multiple announcements and updates whereas; a single notification is issued by an employee.
13. An individual only has one salary. On the other hand, multiple employees can have the same salary.
14. Organization contains multiple inventories to store all the plants.
15. Multiple employees can manage one inventory whereas an employee is assigned to only one inventory.
16. An inventory can contain multiple plants but only one plant can be assigned to a single inventory.
17. Plants can be of multiple types.
18. A user can purchase multiple times.
19. A single purchase can have only one customer.
20. An invoice is generated when the purchase is made.

- 21. A user can choose the method of payment.
- 22. A user can receive multiple recommendations while, one recommendation is sent to only one user.
- 23. One expert can send multiple recommendations whereas one recommendation is associated with only one expert.

#### 4.4.2. Member Responsibilities

Member Name	Responsibilities
Aashraya Shrestha	<ul style="list-style-type: none"> <li>- Goal Statement</li> <li>- Goals of Implementation</li> <li>- SRS and DFD explanation</li> <li>- ERD, Structure Chart, Assumption</li> <li>- Purchase Plant (Individual Task)</li> </ul>
Anjan Giri	<ul style="list-style-type: none"> <li>- Introduction</li> <li>- Problem Statement</li> <li>- Structure chart explanation</li> <li>- ERD, Structure Chart, Assumption</li> <li>- Join the Program (Individual Task)</li> </ul>
Sanskar Karki	<ul style="list-style-type: none"> <li>- Non-Functional Requirements</li> <li>- ERD, Structure Chart, Assumption</li> <li>- DFD Level 2</li> <li>- Take Certification Exam (Individual Task)</li> </ul>
Shamel Rai	<ul style="list-style-type: none"> <li>- Business Case</li> <li>- Context Level Diagram and DFD Level 1</li> <li>- Data Dictionary</li> <li>- Process Specification</li> <li>- Functional Requirements</li> <li>- Make Payment (Individual Task)</li> </ul>
Sulav Maharjan	<ul style="list-style-type: none"> <li>- Scope</li> <li>- ERD and Module Specification explanation</li> <li>- Summary</li> <li>- Report Preparation (Individual Task)</li> </ul>

Table 7: Individual Contribution

**4.4.3. Meeting Minute (Group Meetings)**

Date	Time	Location	Discussion
December 6, 2023	10:30 A.M. - 12:00 P.M.	Alumni Block	Analyzed the problem and assigned a few topics to the team members.
December 10, 2023	11:00 A.M. - 12:00 P.M.	Discord	Made sure that all the members completed the assigned tasks properly and made some corrections.
December 20, 2023	2:30 P.M. - 4:00 P.M.	Study Room	Members were divided into two groups where, team 1 completed the ERD and team 2 completed the DFD.
December 21, 2023	11:00 A.M. - 2:00 P.M.	Kumari Hall	Reviewed our DFD and ERD while also making corrections. Also completed Data Dictionary.
December 28, 2023	11:00 A.M. - 12:00 P.M.	London Block	Assigned individual tasks to all members and worked on Structure chart and Project Specifications.
December 31, 2023	11:00 A.M. – 1:15 P.M.	Study Room	Worked on writing assumptions, members responsibilities and completed assignment diary.
January 4, 2024	1:00 P.M. - 2:00 P.M.	Alumni Block	Make the final correction and finalized the documentation.

*Table 8: Meeting Log*

## 5. Individual Tasks

### 5.1. Make Payment

#### 5.1.1. Environmental Model Specification

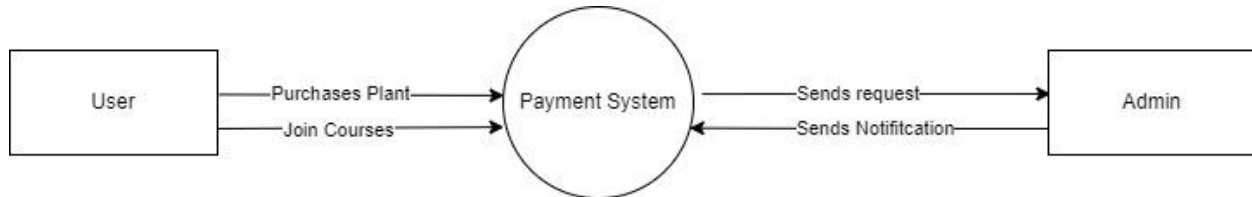


Figure 11: Make Payment Context Level Diagram:

In the system above, the user can join the course which can be both paid/unpaid courses and can also buy plants. When the user buys any plant, it goes through the payment system. The system sends request to the admin and after receiving the request, the admin will provide the user with the notification.

### 5.1.2. Internal Model Specification

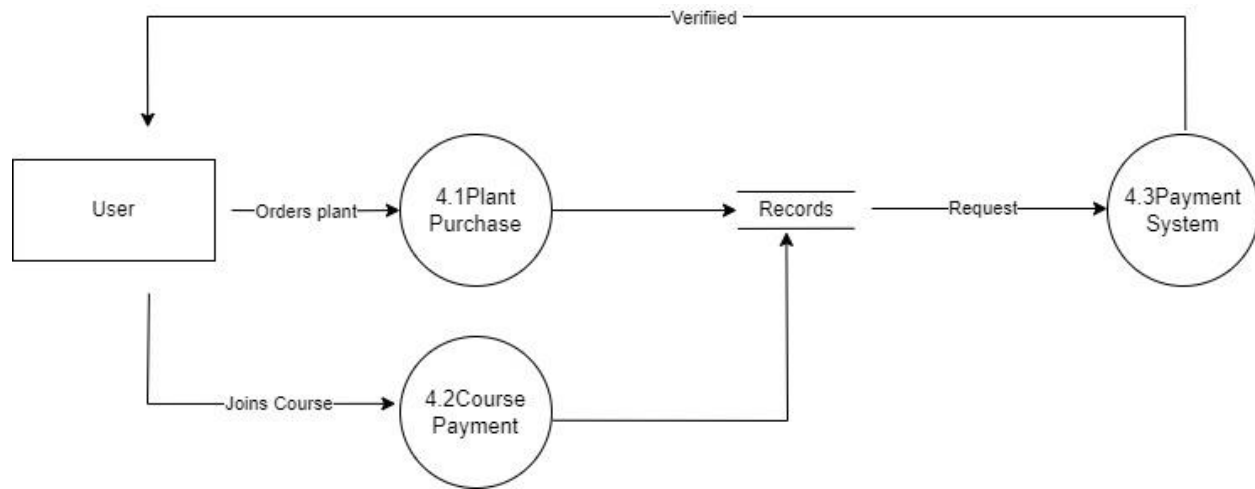


Figure 12: Make Payment Level 1 DFD

In the DFD diagram above, the user orders a plant which is processed through the purchase plant system, which then is stored in the records. The record will request the payment system for the payment process and if the payment verification is completed, a notification is sent to the user.

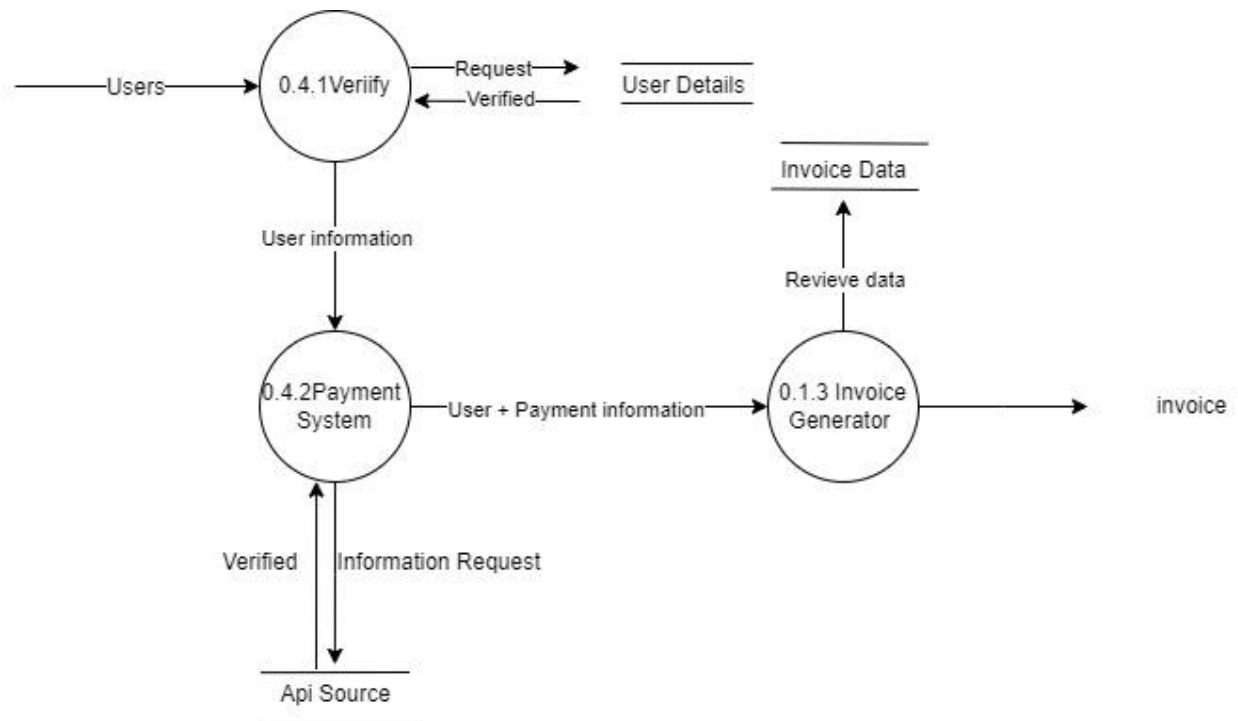


Figure 13: Make Payment Level 2 DFD

In the level 2 DFD diagram, user sends their user credentials for verification process then sending a request to the data stored in the user detail. After the verification is complete, user information is sent to the payment system. The payment system will send the information to the required API source, if the source is verified and the payment system is notified. The payment system will send the user and payment information to the invoice generating system which will store the invoice in the invoice data and sends it to the invoice to the user.



### 5.1.3. Design Specification

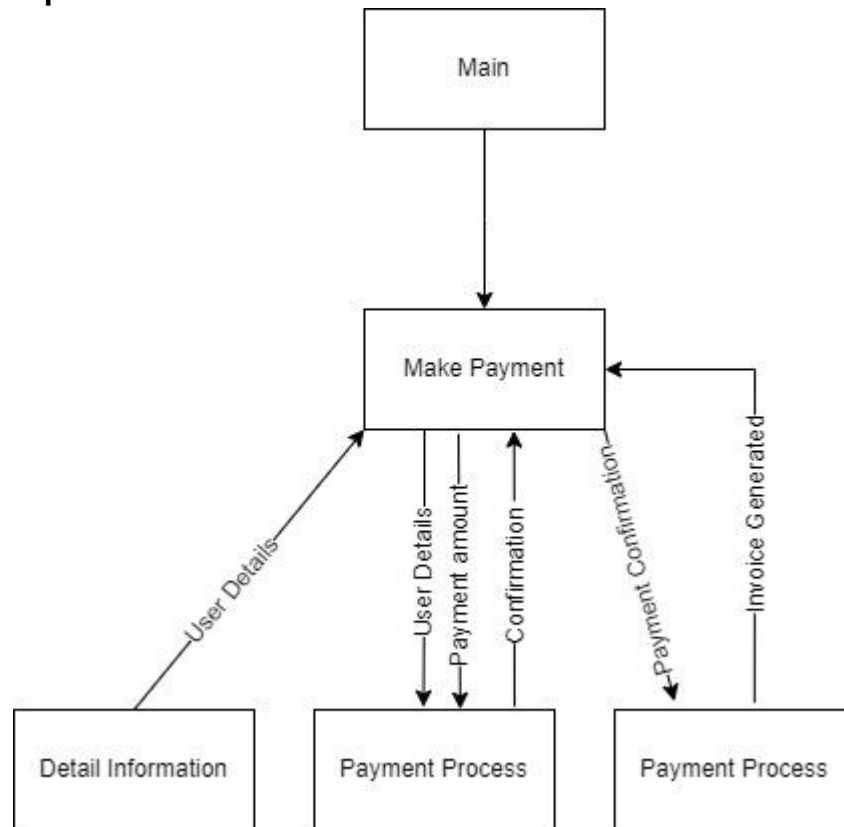


Figure 14: Make Payment Structure Chart

**Module Specification**

Purpose: Handles user data verification payment processing, Api calls and invoice generation

**PSEUDOCODE**

```
Function processUserPurchases(UserData)
Items = UserData.items

Amount =userData.amount

paymentMethod = userData.paymentMethod

Username = userData.username
Email = userData.email

phoneNumber = userData.phoneNumber

VerificationStatus = call UserDetailDataModule(Username,email, phonenumber)

If verificationStatus == "Successful"
    paymentStatus = Call PaymentSystemModule( item, amount, paymentMethod)
    If paymentStatus ==="Successful"
        If paymentMethod == "CreditCard" // or anyother method
            ApiResponse = Call APIModule(userData)

            If apiResponse == "Successful"
                Invoice = call invoiceGeneratorModule(userData, paymentStatus,
                    apiResponse)
                sendInvoiceToEmail(email,invoice)
            Else:
                HandleApiError(ApiResponse)
        Else:
            handlePaymentError(paymentStatus)
    Else:
        handleUserDetailVerificationError(verificationStatus)
```

**INPUT PARAMETER:-** User purchase data

**OUTPUT PARAMETER :-** invoice

**GLOBAL Variable :** NONE

**CALLS:-** User detail data module, payment system, module , Invoice generator , API

**CALLED BY:** invoice module

## 5.2. Purchase Plant

The program includes the facility of purchasing variety of plants where the price varies from minimal amount to even free for some cases. For this, the customer must request the specific plant species. Then the system needs to make sure if the desired plant is available or not by surfing through the present data. After which the plant is handed over to the customer by one of the staff members.

### 5.2.1. Environmental Model Specification

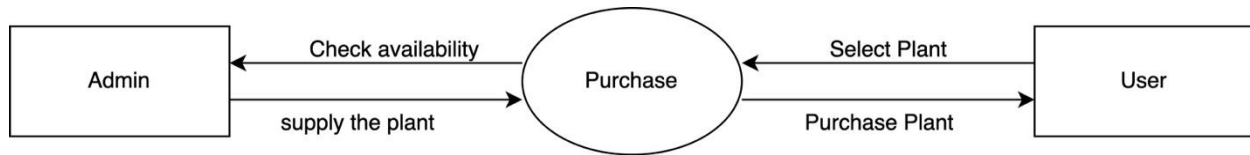


Figure 15: Context Level Diagram

Here, the user selects the desired plant which is checked by the admin through the purchase system. The admin then supplies the plant which the customer pays for before receiving.

### 5.2.2. Internal Model Specification

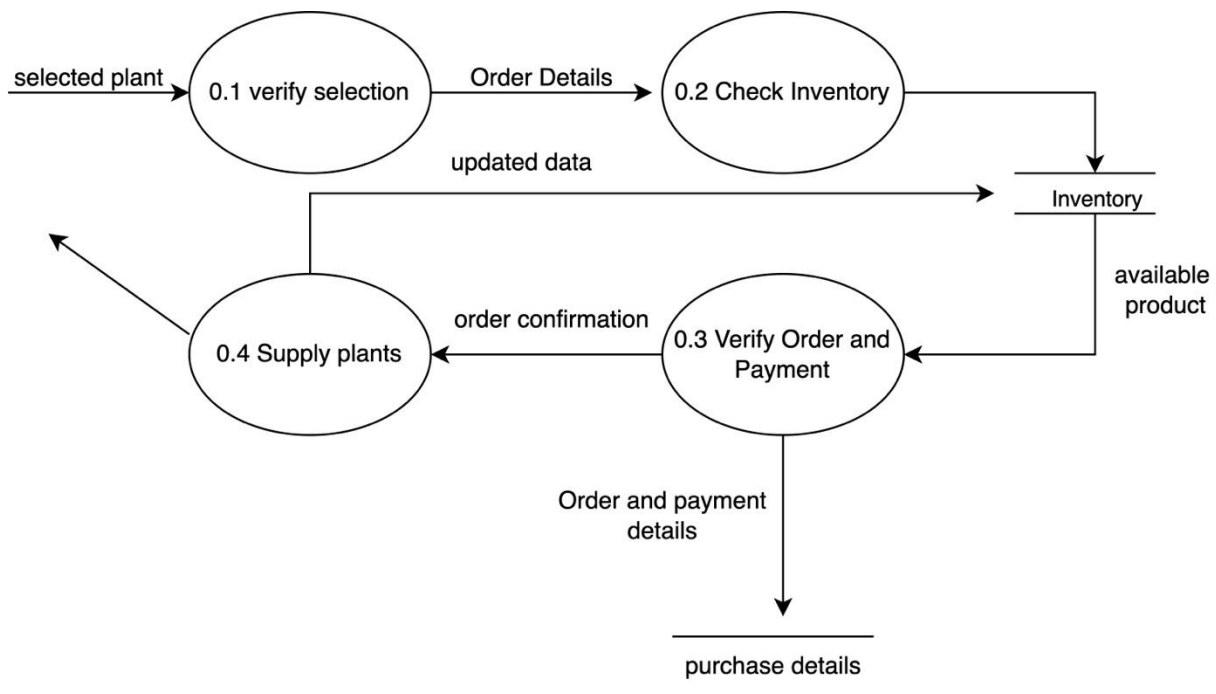


Figure 16: Purchase Plant Level 1 DFD

Here, the selected plant is verified by the system and the order details are sent where the availability of the plant is confirmed. If the plant is available within the stock, the payment and order is verified. The details of the order and the payment is stored within the system. After the verification, the user is supplied with the plant and the inventory is updated.

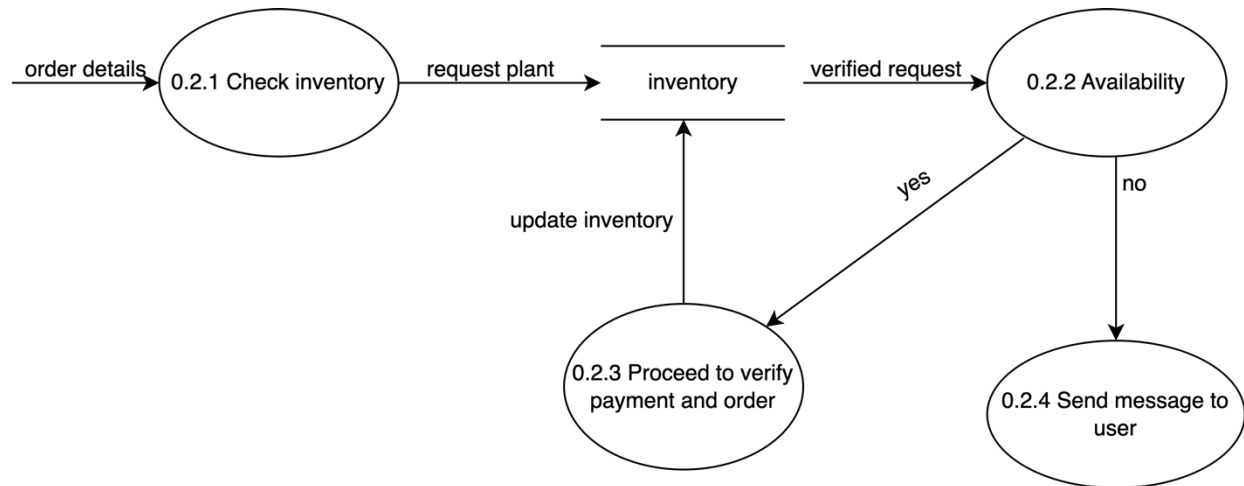


Figure 17: Purchase Plant Level 2 DFD

This diagram shows the process of inventory verifying the users selected plant and how the inventory is managed. When the order details are received, the requested plant is checked within the inventory and confirmed if it is available or not.

If the selected plant is available, then the payment and order is verified, and the inventory is updated accordingly.

On the other, if the selected plant is out of stock, then a message is sent to the user informing them about the situation.

### 5.2.3. Design Specification

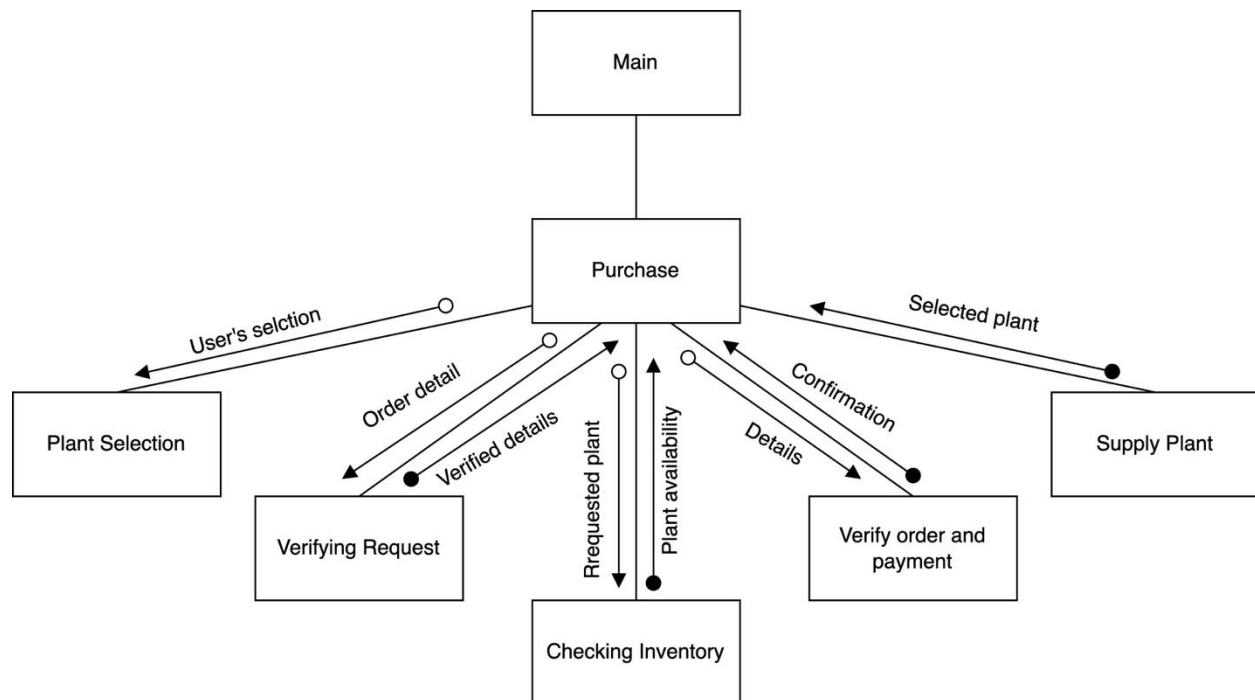


Figure 18: Purchase Plant Structure Chart

**Module Specification:**

**Module Name:** Purchase plant

**Purpose:** The purpose of this report is to make the purchase process of the plants seamless and effortless. This requires the user to select their desired plant which is then verified and confirmed by the system before supplying it to them.

**Pseudocode:****DO**

Var selected\_plant = input ("Enter the plant of your choice:")

**IF** (Validate (user\_detail, order\_detail))

**IF** (plant = available)

Var payment\_type = input ("Enter the payment type:")

**IF** (Validate (Payment))

**Display** ("Payment successful")

**Else IF** (plant = navailable)

**Display** ("Sorry, the requested plant is out of stock!")

**End IF**

**End IF**

**End DO**



**INPUT PARAMETERS:** payment\_type

**OUTPUT PARAMETERS:** Payment-successful message, Out-of-stock message

**GLOBAL VARIABLES:** user\_detail, order\_detail

**LOCAL VARIABLES:** selected\_plant , payment\_type

**CALLS:** validate()

**CALLED BY:** MAIN

### 5.3. Report Preparation

This module's objective is to produce user report. It gathers information from the user data, employee data, and financial data in order to generate the report. The details of the process are stated below through DFD levels:

#### 5.3.1. Environmental Model Specification

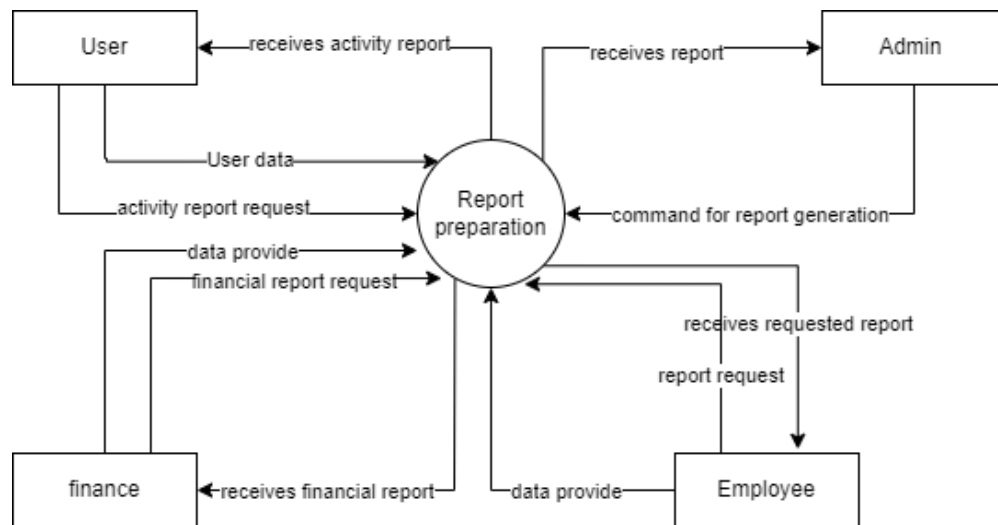


Figure 19: Report Preparation Context Level Diagram

### 5.3.2. Internal Model Specification

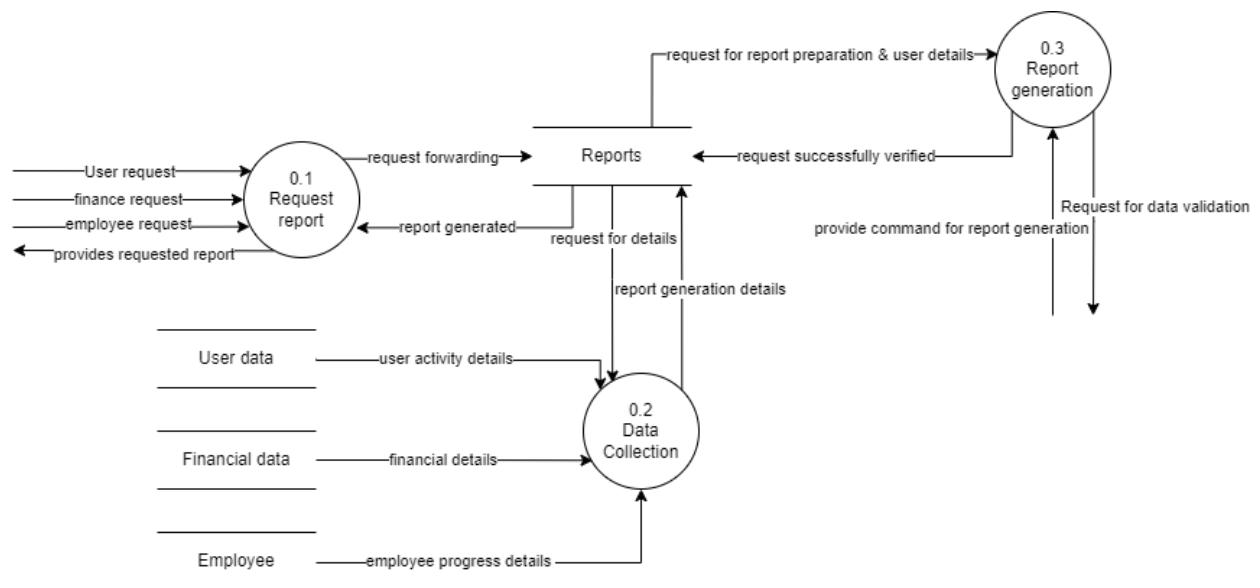


Figure 20: Report Preparation DFD Level 1

In the diagram above, it is shown where report is generated, data collection stores all the information about user, financial data and employee progress. As report generation takes the request of report preparing and user details, it verifies the request and sends it back where report is stored.

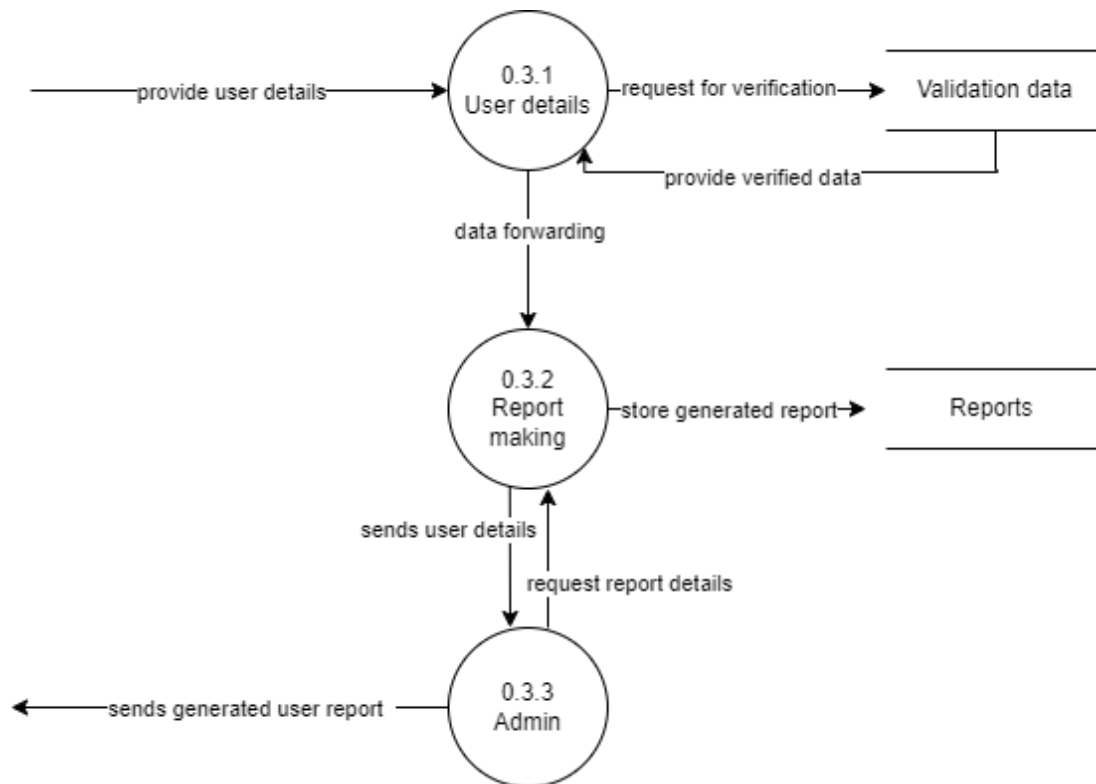


Figure 21: Report Preparation DFD Level 2

In DFD level 2, it is shown that in report generation where the details are sent to verify and generate the report from the given data, user details are verified and provided for report making to the admin and then the report is generated.

### 5.3.3. Design Specification

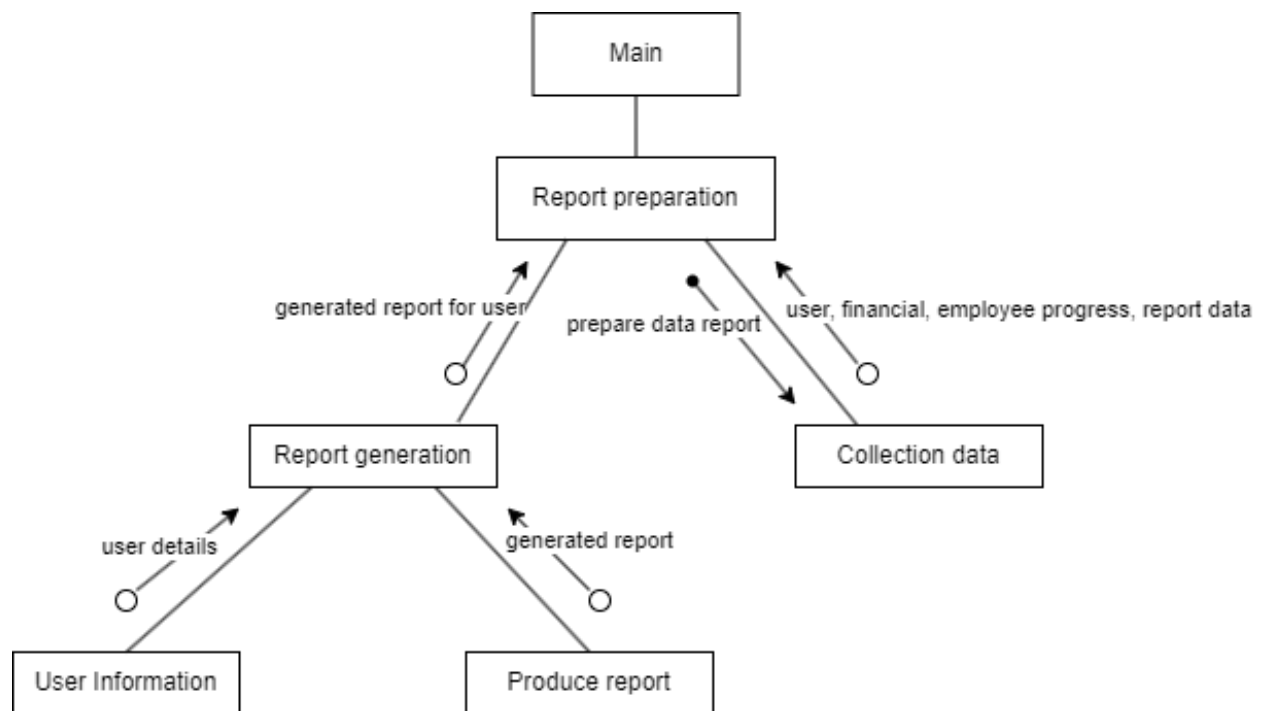


Figure 22: Report Preparation Structure Chart

DO

var user\_data = DB.get\_user\_data()

var financial\_data = DB.get\_financial\_data()

var employee\_data = DB.get\_employee\_data()

var report = generate\_report (user\_data, financial\_data, employee\_data)

DISPLAY (report)

END DO

**Input Parameters:** user\_data

**Output Parameters:** user\_data

**Global Variable:** DB

**Local Variable:** user\_data, financial\_data, employee\_data, report

**Calls:** Get user data, financial data, employee data and generated report of the user

**Called by:** Main

## 5.4. Join the Program

### 5.4.1. Environmental Model Specification

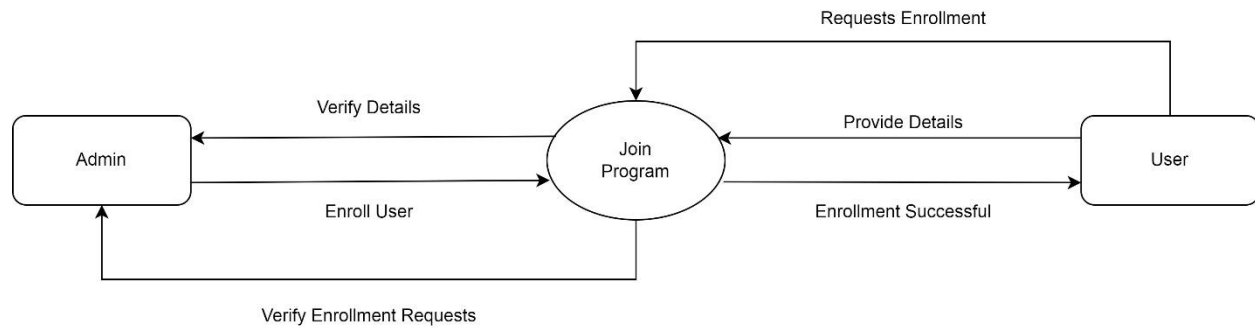


Figure 23: Join the Program Context Diagram

Firstly, user requests enrollment through the system and provides his/her details for enrollment.

Then, staff reviews the enrollment requests and verifies the details provided by the user.

After reviewing and validating all information provided by the user, he/she is enrolled in the program.

### 5.4.2. Internal Model Specification

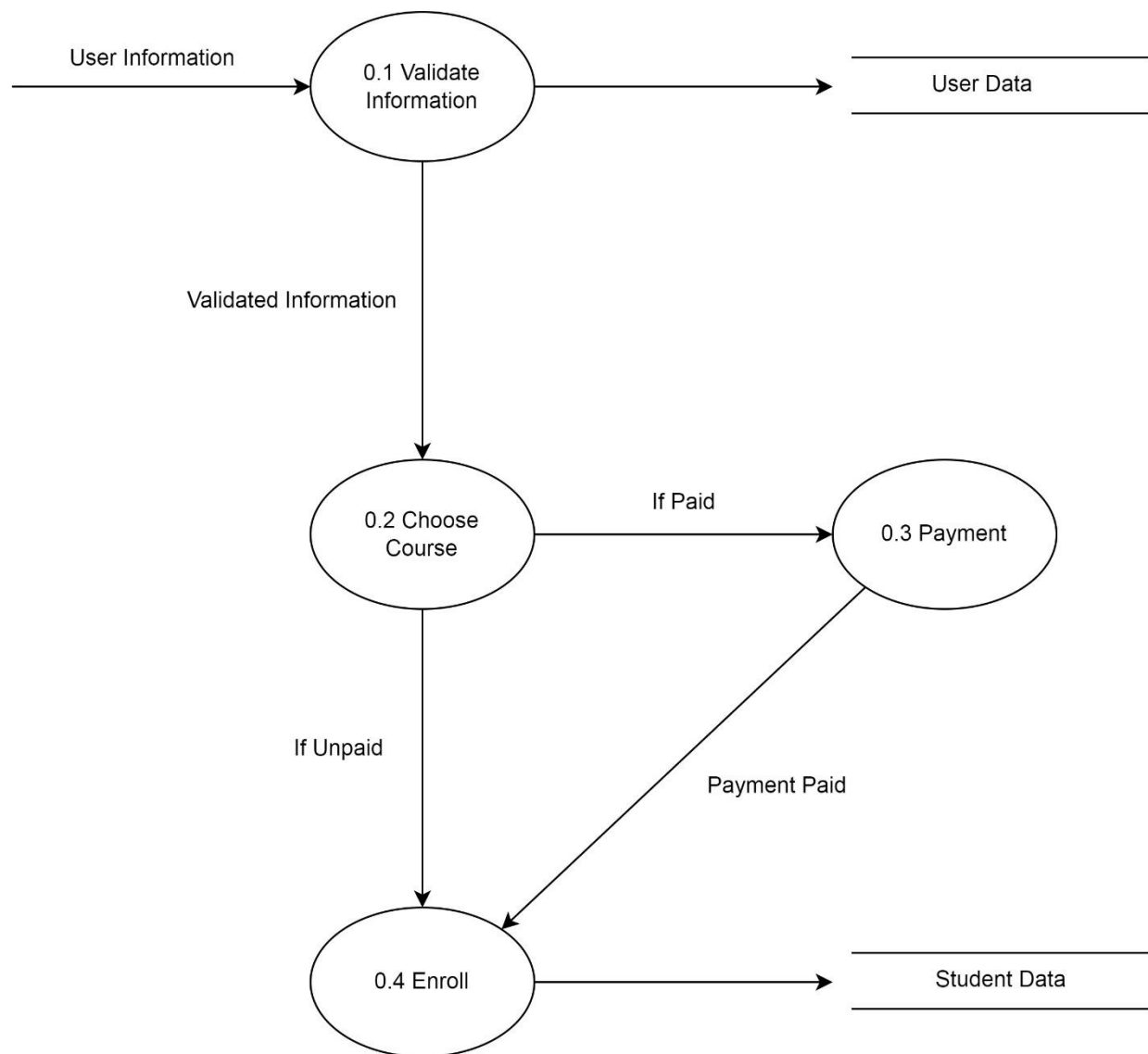


Figure 24: Join the Program Level 1 DFD

The information provided by the user is validated and is stored in user data. After that, course is chosen to join, and course type is checked whether the course is paid or unpaid. After verifying payment, the user is then enrolled in the course and that data is stored in Student Data.



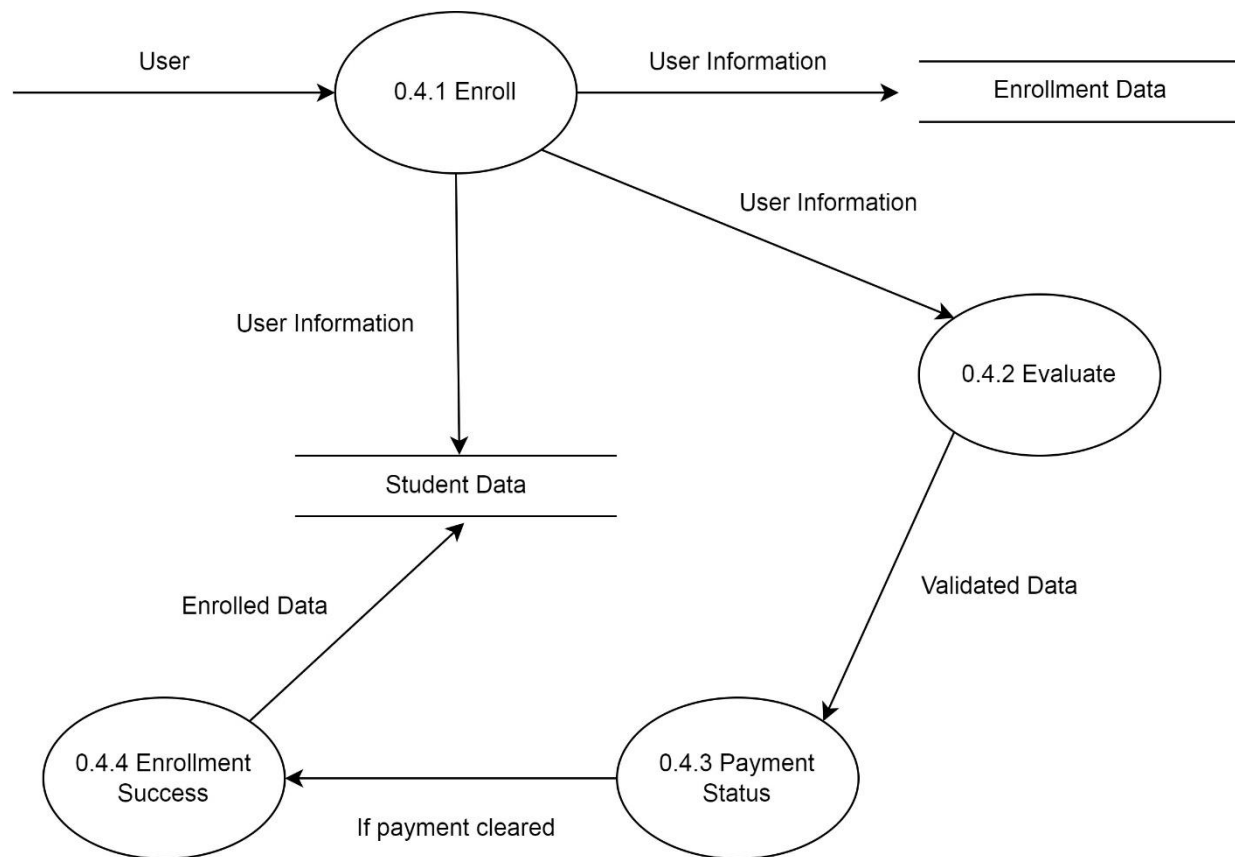


Figure 25: Join the Program Level 2 DFD

For enrollment process, user information is stored in enrollment data.

User information is also validated first and payment status is checked whether it is cleared or not. After payment is cleared of the user, he/she is enrolled in the course successfully and his/her data is stored in the Student Data.

### 5.4.3. Design Specification

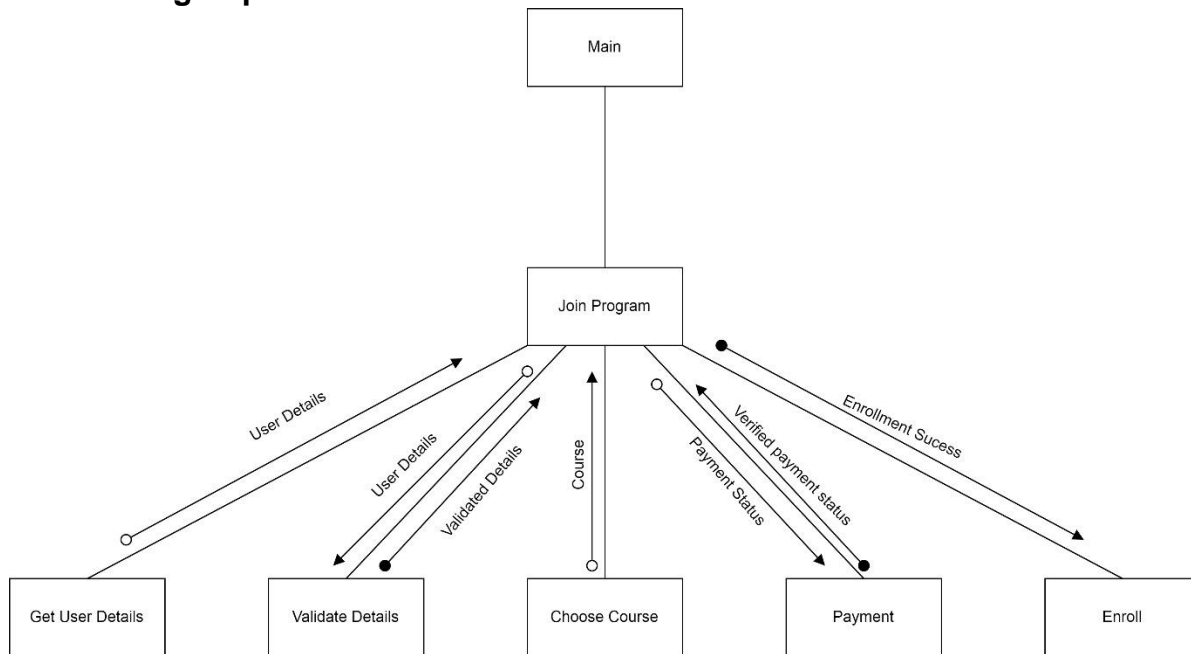


Figure 26: Join the Program Structure Chart

**MODULE NAME:** Join the Program

**PURPOSE:** Join the Program module gets user detail and stores it in the User Data. Then, the data is validated by the admin and after thorough check of the data, the user is enrolled in the course of his/her choice and stores the data in student record.

**PSEUDOCODE:**

DO

```
var FullName = input ("Enter your full name:")
var UserName = input ("Enter your username:")
var contact = input ("Enter your contact information:")
var address = input ("Enter your address:")
var age = input ("Enter your age:")
var DOB = input ("Enter your date of birth:")
var course = input ("Enter the course that you want to join:")
var password = input ("Enter the password:")
if (validate (FullName, UserName, contact, address, age, DOB, course,
password)
//check whether the course selected by the user is paid or unpaid
    if (course = paid)
        var PaymentType = input ("Enter your Payment Method:")
        var PaymentAmount = input ("Enter the amount paid:")
        var PaymentDate = input ("Enter the date in which the payment is
made:")
        if (validate (PaymentType, PaymentAmount, PaymentDate)
//enroll the user and provide the id
        Student_id = DB.User_Data (FullName, UserName, contact,
address, age, DOB, course)
        DB.Student_Data (Student_id, PaymentType, PaymentAmount,
PaymentDate)
        DISPLAY ("Program joined Successfully")
        DISPLAY ("Your student ID is:" + Student_id)
```

```
    else if (course = unpaid)
        Student_id = DB.User_Data (FullName, UserName, contact,
        address, age, DOB, course)
        DISPLAY ("Program joined Successfully")
        DISPLAY ("Your student ID is:" + Student_id)
END DO
```

**INPUT PARAMETERS:** user\_details

**OUTPUT PARAMETERS:** Student\_id

**GLOBAL VARIABLES:** DB

**LOCAL VARIABLES:** FullName, UserName, contact, address, age, DOB, course, password, PaymentType, PaymentAmount, PaymentDate

**CALLS:** get user details, validate data, add user data, get payment details, add payment data, display success message along with student id

**CALLED BY:** Main

## 5.5. Take Certification Exam

### 5.5.1. Environmental Model Specification

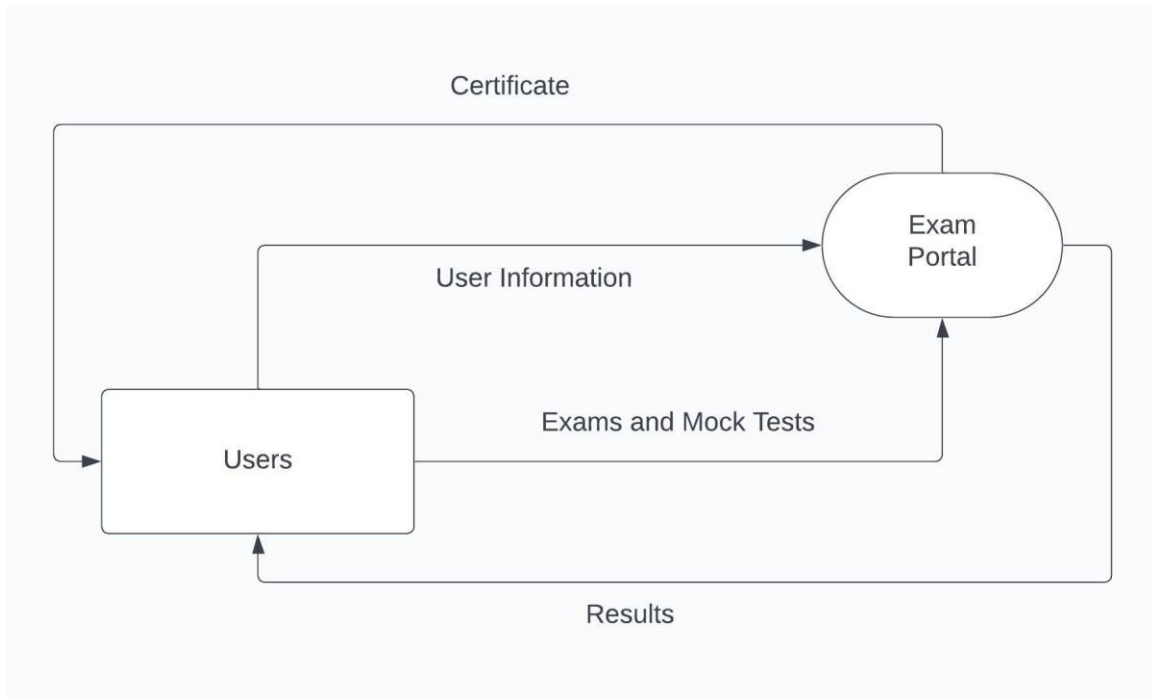


Figure 27: Take Certification Exam Context Diagram

In the diagram above, the user takes exam and mock tests and submits it to the Exam Portal after verifying user's information. When the results are in, they user may be eligible for granting certificate.

### 5.5.2. Internal Model Specification

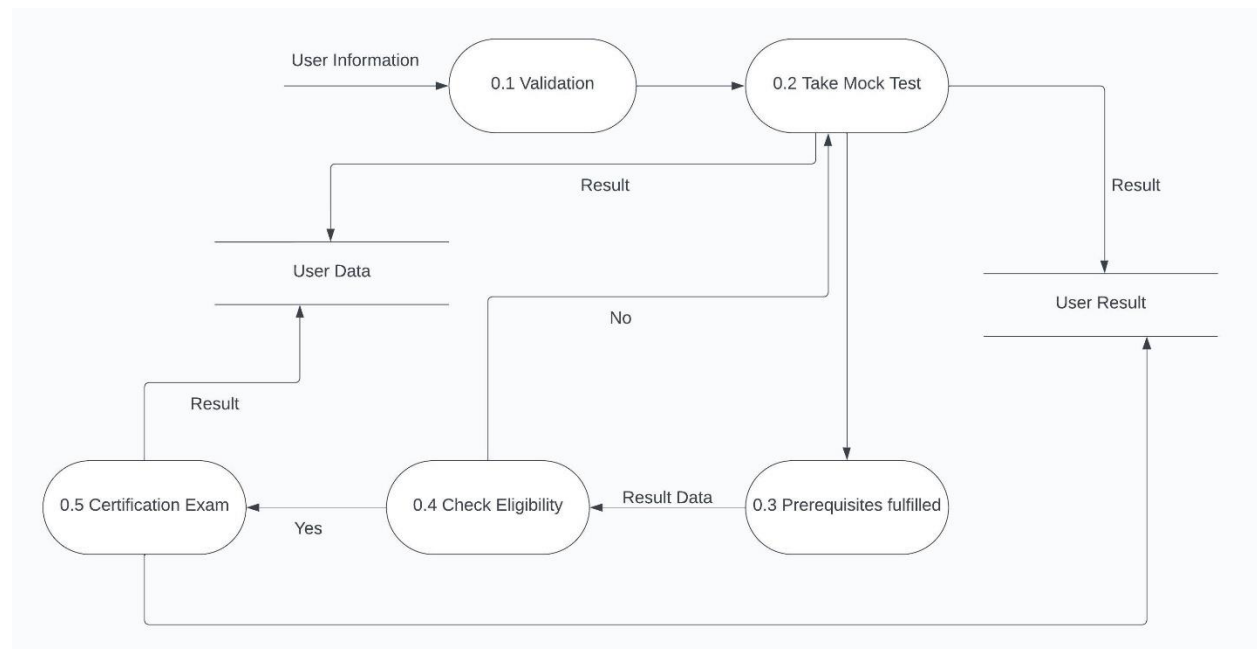


Figure 28: Take Certification Exam Level 1 DFD

In the diagram, user information is sent to be validated and after validation, the student may take mock test to check their capability. If they do well in mock test by getting a passing score, they get confident for certification exam. The results of mock test and certification exam are saved in datastore.

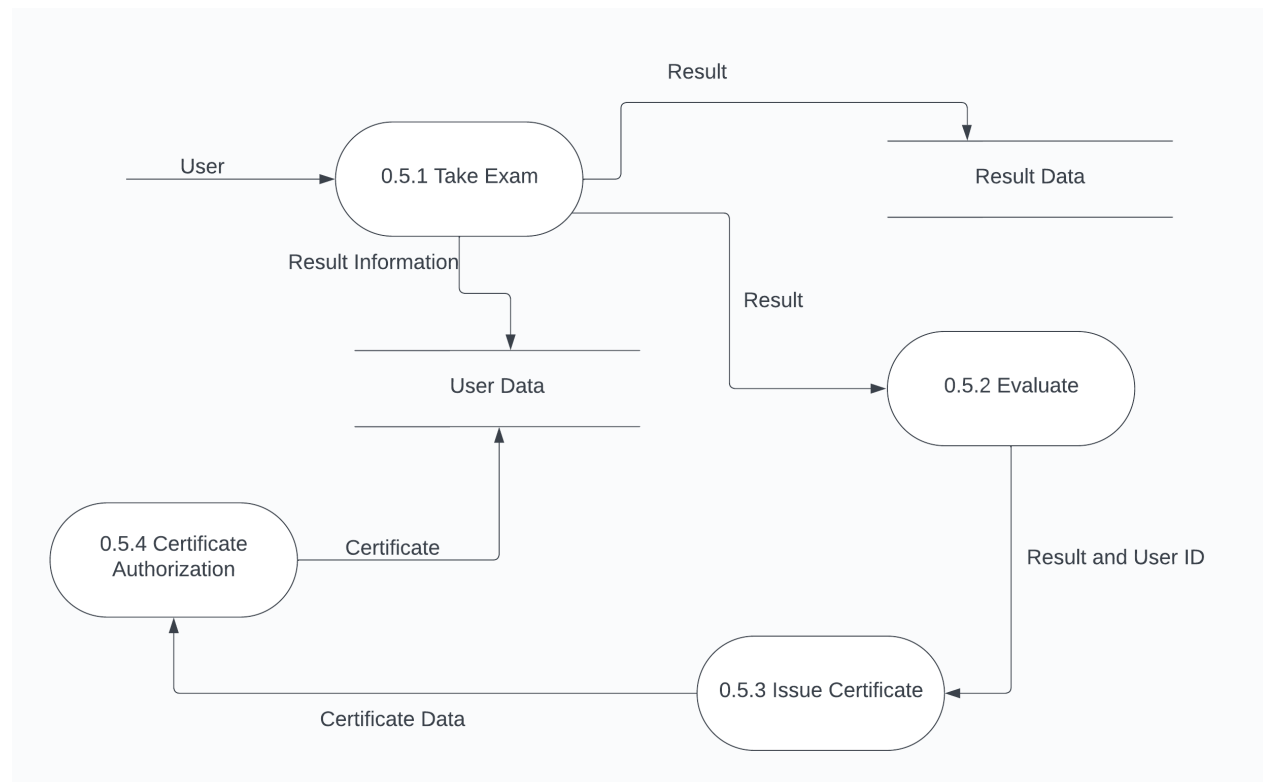


Figure 29: Take Certification Exam Level 2 DFD

Here, when user takes the exam, their results are saved in a datastore. The Result information is also saved in User Data. The results are then sent for evaluation and after comparing results and user ID, the certificates are issued. The certificates are then authorized and saved in user data.

### 5.5.3. Design Specification

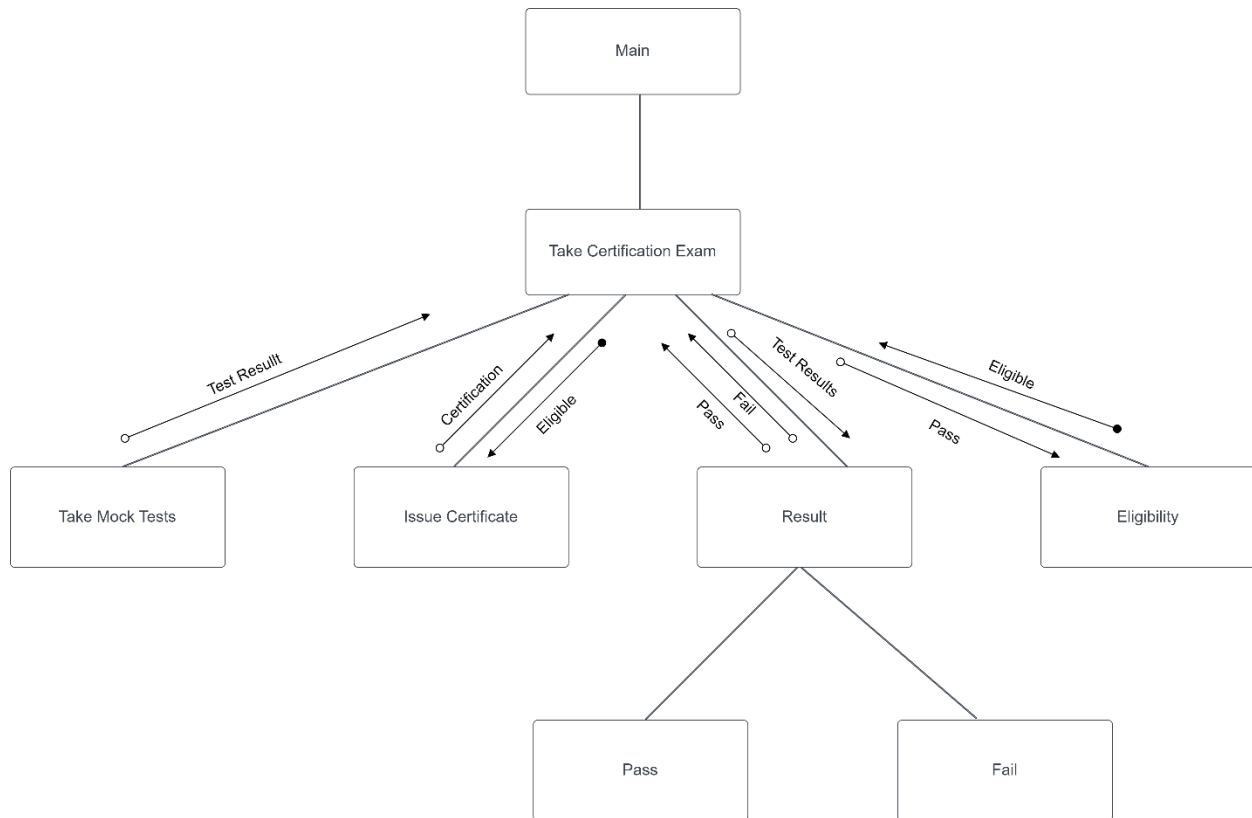


Figure 30: Take Certification Exam Structure Chart



## Module Specifications

**Module Name:** Take Certification Exam

**Purpose:** This module validates user information making them eligible for taking mock tests and exams. Likewise, when the users do well and receive the score required for their certification, they are declared eligible for granting a certificate.

**Pseudocode:**

**DO**

```
var FullName = input("Enter your Full Name: ")
var UserName = input("Enter your Username: ")
var Contact Number = input("Enter your Contact Number: ")
var age = input("Enter your age: ")
var Date of Birth = input("Enter your Date of Birth: ")
var Course = input("Enter your Course: ")

IF (validate (FullName, UserName, Contact Number, age, Date of Birth, Course))

    IF (validate(Mock Test Results))

        Student_ID = DB.User_Data(FullName, UserName, Contact
        Number, Age, Date of Birth, Course)

        DISPLAY ("Prerequisites Fulfilled")

        IF (validate(Eligibility))

            DISPLAY("You are eligible for certification exam")

        END IF

    END IF

    IF (validate(Certification Exam))
```

Student\_ID = DB.User\_Data(FullName, UserName, Contact  
Number, Age, Date of Birth, Course)

**DISPLAY**("You are granted a Certificate")

**END IF**

**END IF**

**END DO**

**INPUT PARAMETERS:** user\_details

**OUTPUT PARAMETERS:** results, certificate

**GLOBAL VARIABLES:** DB

**LOCAL VARIABLES:** FullName, UserName, Contact Number, age, Date of Birth,  
Course

**CALLS:** validate()

**CALLED BY:** Main

## 6. Summary

In summary, the purpose of this coursework was to develop a platform (forum) for an Ireland based training institute located at Godavari, Lalitpur named McGregor Institute of Botanical Training. It specializes in agriculture along with horticulture and are now planning to offer a variety of plants for a minimal cost, or even free in certain circumstances. They hope to establish a group of individuals sharing a passion for plants and wish to get together in a forum or platform where admirers for plants can exchange ideas and arrange campaigns to save endangered species and forests. The forum may also serve as a venue for users to post queries for the experts to respond to. So, we made a project charter and met their software requirement as planned and implemented a significant portion of the system utilizing the structured method and accomplished the objective. As a result of taking part in this process, we were able to demonstrate our practical understanding of structured software engineering and work really well even after having a small group to finish the task on time.

Our project has been successfully completed to design a platform for the McGregor institute of Botanical Training. Every member of our team has contributed a lot in this project. Without the proper coordination and contribution from the team members, completing the project within time was close to impossible. This project involved a number of problems and points of debate, but we discussed them and worked through them in our meetings. In order to finish this job on time, we conducted extensive research on the idea of software engineering through websites, journals, and other sources. The finished forum will surely be an effective tool for the institute in a long run providing them with the facility helping the betterment of the institute and also helps build a community of individuals with interests in horticulture.

We really thank our tutors who guided us throughout this coursework while giving us several ideas and providing instructions which drastically helped us completing it properly. In this coursework we learned about the skills of software engineering and its practical uses which can be utilized in real life scenarios.

## 7. References

- Datensen, 2023. *Relationships in ER diagrams | Datensen*. [Online]  
Available at: <https://www.datensen.com/blog/data-modeling/relationships-in-er-diagrams/#:~:text=Types%20of%20Relationships%20in%20ER%20diagrams&text=In%20a%20one%2Dto%2Dmany,another%20table%2C%20and%20vice%20versa.>  
[Accessed 23 12 2023].
- GeeksforGeeks, 2023. *Software Requirement Specification (SRS) Format - GeeksforGeeks*. [Online]  
Available at: <https://www.geeksforgeeks.org/software-requirement-specification-srs-format/>  
[Accessed 24 12 2023].
- Geeksforgeeks, 2023. *Types of Attributes in ER Model - GeeksforGeeks*. [Online]  
Available at: <https://www.geeksforgeeks.org/types-of-attributes-in-er-model/>  
[Accessed 23 12 2023].
- Geeksforgeeks, 2024. *Structure Charts - Software Engineering - GeeksforGeeks*. [Online]  
Available at: <https://www.geeksforgeeks.org/software-engineering-structure-charts/>  
[Accessed 02 01 2024].
- Javatpoint, 2021. *Software Engineering | Software Requirement Specifications - javatpoint*. [Online]  
Available at: <https://www.javatpoint.com/software-requirement-specifications>  
[Accessed 24 12 2023].
- Sciencedirect, 2003. *Structure Chart - an overview | ScienceDirect Topics*. [Online]  
Available at: <https://www.sciencedirect.com/topics/computer-science/structure-chart#:~:text=1.-,Structure%20Charts,of%20the%20main%20logical%20actions>  
[Accessed 02 01 2024].
- Techopedia, 2014. *What is an Entity? - Definition from Techopedia*. [Online]  
Available at: <https://www.techopedia.com/definition/14360/entity-computing>  
[Accessed 23 12 2023].
- Techtarget, 2019. *What is Entity Relationship Diagram (ERD)? | Definition from TechTarget*. [Online]  
Available at: [A relational database's basis and a means of defining business processes are both provided by the data modeling approaches employed in an ERD.](#)  
[Accessed 23 12 2023].
- UC Merced Library, 2023. *What Is a Data Dictionary? | UC Merced Library*. [Online]  
Available at: <https://library.ucmerced.edu/data-dictionaries#:~:text=A%20Data%20Dictionary%20is%20a,part%20of%20a%20research>

%20project

[Accessed 23 12 2023].