McGregor-Institute-Project

Introduction:

McGregor Institute of Botanical Training is a training institution established in Godawari, Lalitpur. It has been successfully doing business in Nepal for over 7 years. It is associated with Dublin City University and delivers a variety of undergraduate and graduate programs with a particular emphasis on agriculture and horticulture. Due to the increase in number of people interested in horticulture, the McGregor institute aims to establish an environment or online platform where people may interact, exchange ideas and plan events with the goal of preserving plants. This platform will act as a gathering place for people with similar interests to collaborate and exchange knowledge.

1.1. Aims and Objectives

The McGregor Institute of Botanical Training provides undergraduate, postgraduate, and short-term certification programs with the goal of providing thorough instruction in horticulture and agriculture. The project seeks to create a thriving community of plant enthusiasts by creating a place for discussions, information exchange, and professional guidance. To promote biodiversity and sustainable gardening methods, the institute intends to sell a variety of plant species, with a focus on accessibility. The following objectives were achieved to attain the aims of the course and the system:

- To satisfy the increasing interest in horticulture by providing top-notch undergraduate and postgraduate courses in agriculture and horticulture, and short-term certification programs.
- To outline the functional requirements, their features, and their implementation, the development team prepares a comprehensive System Requirement Specification document.
- The internal and external environmental model specifications were illustrated with a data flow diagram, an entity relationship diagram, a data dictionary, structure charts, and process specifications in order to sequence.

- After completing the group project assigned to each member, each member is given an individual assignment to thoroughly describe one of the procedures listed.

1.2. Business Modality

The business modality of McGregor Institute of Botanical Training is a holistic approach that combines community engagement, commerce, and education. By offering both paid and unpaid courses in horticulture and agriculture, the institute generates revenue and makes education more accessible. The institute also sells plant types at little or no cost, which not only creates another source of income but also promotes inclusivity and accessibility. The platform's forum feature enables plant enthusiasts to exchange ideas, plan events, and consult professionals, fostering community formation. Expert advice tailored to users' locales and soil conditions enhances the platform's value and user experience. Mock tests and certification exams promote lifelong learning and increase revenue. The notification system guarantees user involvement by delivering information on time. The business model also enables strategic decision-making and potential partnerships by gathering and analyzing user data. The McGregor Institute's comprehensive approach reflects its educational objective and promotes a thriving and informed community of plant enthusiasts.

1.3. Problems and Consequences

The McGregor Institute has never had an application specifically curated for their institution before, so they may be unaware of their application necessities. This lack of awareness could potentially result in the project not meeting the institution's expectations. If such a mishap were to occur, it would lead to both parties wasting time and money, which would undoubtedly pose a significant consequence.

2. Project Charter

A project charter document is a high-level document that outlines the purpose, goals, and scope statement of a project. It serves as a formal authorization for the project to begin and sets the foundation for project planning and execution. It provides a mutual understanding and agreement among all main stakeholders and sponsors about the project's purpose, scope, and expected outcomes. It also serves as a reference point throughout the project management process to help ensure that the project remains on track and meets your objectives (ATLASSIAN, 2023).

Project Name	McGregor Plant Application
Project Description	The McGregor Institute of Botanical Training is a training facility situated in Godawari, Lalitpur, Ireland. They have been conducting business in Nepal for almost 7 years. However, due to the recent unexpected increase in interest in the field of agriculture, a variety of short-term certification programs in horticulture will be made available to those who are interested. They plan to provide a variety of plant kinds for sale at a reasonable cost, and in certain instances, for free, in addition to the certification classes. They aim to unite people who have a passion for plants, establish a forum where those with an interest may exchange ideas, and coordinate initiatives to preserve rare plants and forests.
Project Objective	The main objective of this project is to provide a forum or online platform where horticultural enthusiasts can interact, exchange ideas, and talk about a range of plant related subjects.

<table>
<tr><td><b>Problem Statement</b></td><td colspan="2">The main problem of this project is there is no centralized location where people who are passionate about plants can go to expand their knowledge, exchange ideas, and get involved in environmental conservation initiatives.</td></tr>
<tr><td><b>Business Case</b></td><td colspan="2">Establish an application in response to the growing interest in horticulture and agriculture and become a comprehensive center for plant education, accessibility, and community building.</td></tr>
<tr><td><b>Goal Statement</b></td><td colspan="2">The main goal of this project is to establish a whole ecosystem that offers instruction, encourages social responsibility, and advances environmental preservation in Nepal floral and agricultural fields.</td></tr>
<tr><td><b>Timeline</b></td><td colspan="2">After approval, the project will begin within one month and be completed within eight months.</td></tr>
<tr><td><b>Scope</b></td><td colspan="2">The scope of this project is to motivate experts in the field to contribute to conversations and impart their knowledge and putting together community based programmed aimed at preserving rare plants and forest ecosystems along with the support for sustainable development and preservation of the environment.</td></tr>
<tr><td rowspan="6"><b>Team Members</b></td><td>Title Role</td><td>Name of members</td></tr>
<tr><td>Project Manager</td><td>Aayushree Dahal</td></tr>
<tr><td>Developer</td><td>Apreksha Gurung</td></tr>
<tr><td>Finance Manager</td><td>Neha Pokhrel</td></tr>
<tr><td>Resource Manager</td><td>Pritishma Kunwar</td></tr>
<tr><td>Tester</td><td>Sajiya Khatoon</td></tr>
</table>

Potential Risk	The potential risk may be not being able to complete the project in the stated time because of unforeseen issues. The team may exceed the given budget due to the extension of the time frame of the project.

Table 1: Project Chater

3. Software Requirement Specification:

Software Requirement Specification (SRS) is an in-depth definition and description of the software requirements that must be achieved for the software system to be developed successfully. Depending on the sort of requirement, these could be non-functional or functional. It is vital to completely understand the needs of customers, and that is why various customers and contractors connect. SRS, which specifies software requirements and may include modifications and changes required to improve product quality and meet customer demands, is generated based on information obtained following interaction (geeksforgeeks, SRS Format, 2024).

3.1. Functional Requirements:

This fully clarifies the impact that the program's functioning may have on the software system. Every functional requirement, which might include data processing, calculations, etc., is arranged in a prioritized manner. The planned operation of the system which outputs should be generated from the supplied inputs is specified by the functional requirements. They explain how the system's input and output are related to one another. A thorough explanation of all the data inputs, their sources, the units of measurement, and the range of acceptable inputs must be provided for each functional requirement (geeksforgeeks, 2024).

3.1.2. Sign-up

As long as the user corresponds to the requirements, this application permits registration for any user, including teachers, users, customers, and providers which gets monitored by Admin.

3.1.3. Register

In order to log in, new users can generate a unique username and password through the program, which must be approved by the admin.

3.1.4. Enroll

Undergraduate and postgraduate students can use the system to explore and register for courses, as well as to enroll in classes to get certification and extend their botanical knowledge.

3.1.5. Staff Induction

Staff members engaged in non-academic activities shall be able to sell plants through the system with the system administrator's approval.

3.1.6. Teacher Induction

Any user should be allowed to upload their resume and access a list of free information on enrollment through the system.

3.1.7. Preparing questions paper

The system should only give teacher's entry to a platform where they can compile a list of test questions that are appropriate for their students and generate them at exam time.

3.1.8. Payment

There are two ways to purchase the plants: cash on delivery or online banking.

3.1.9. Monitoring

The system list of plants and the data needed to care for each plant are visible to every user who visits the system store section.

3.1.10. Notice

Every announcement should be managed by the administrator, who should also make sure that it is posted according with their requests.

3.1.11. Announcements and Updates

Only students and teacher staff should be able to access the Notices site, which should contain all changes and notices.

3.1.12. Result

With the permission of the teacher, the system should create the results of every exam that the students have taken and notify the students via their special login.

3.1.13. Teacher Vacancy

Any user of the system can access an area specifically especially for job announcements for teachers, which includes the number of teachers needed and the subjects they teach.

3.1.14. Conducting Exams

Exams should be offered online by the system in order to promote students from beginner to intermediate and from intermediate to advanced levels.

3.1.15. Video Streaming

All tutorial videos which are created by teachers have to be managed by the system, which would define them as either academic or practical.

3.1.16. Final Assessment

In order to improve their levels, students take written and practical tests at the end of the year to assess their knowledge.

3.2 Non-Functional Requirements

The restrictions or requirements set for the system are known as non-functional requirements. They detail the software's quality attribute. Factors like scalability, maintainability, performance, portability, security, dependability, and many more are covered by non-functional requirements. Non-Functional Requirements deal with critical software system quality issues. A vital aspect of software engineering are non-functional requirements, or NFRs. They discuss aspects of the software system that have nothing to do with its functional requirements. Performance, dependability, maintainability, usability, security, and many more attributes are among them. A software system's ability to satisfy the needs and expectations of its users and stakeholders depends on its non-functional requirements. In a nutshell non-functional requirement, which go beyond functional requirements to describe the attributes and features of the software system, are an essential component of software engineering. These specifications guarantee that the software system operates effectively and dependably under various loads and conditions while also meeting the expectations of its customers and other stakeholders (geeksgorgeeks, 2024).

3.3 Design and Implementation constraints

Specific limitations or circumstances that influence the creation and design of software systems are known as design and implementation constraints. These limitations may be the result of resource limitations, such as memory or processing capacity restrictions, or technical limitations, such as compatibility with an assembly or technology. Budgetary considerations may also play a role. These barriers are also shaped by regulatory elements like accessibility requirements or industry-specific laws. The numerous limitations that direct design and execution are further exacerbated by time restrictions, collaboration requirements, and implementation considerations. An official workplace's overall performance and smooth functioning depend on the software product meeting quality standards, project objectives, and applicable regulations; all of which depend on the proper management of these restrictions.

3.3.1. Challenges in hardware and software

To operate the program, a minimum of 2 GB of RAM is needed on a phone and a minimum of 4 GB of RAM on a PC.

3.3.2. Limitation of the Developer

The system must work on all platforms and devices that are accessible, including Windows, Linux, iOS, Android, and MacOS.

3.3.3. Operating System and Database Provision

Since Oracle is more compatible with older OS generations, it will be chosen as the database management system.

3.3.4. External Interface is required

These requirements include screen layouts, buttons, features on each screen, hardware interfaces (where a team describes the devices the program is intended for), user interfaces (the logic of how the software interacts with the user), and other relevant information.

3.3.4.1. Hardware

The system needs to work with a wide range of hardware, including desktop, laptop, and mobile devices.

3.3.4.2. Software

To develop the project, Swift and Python should be utilized. Swift is used for iOS and MacOS, and Python is used for Android, Linux, and Windows. Through the Django Framework, Python is utilized.

3.3.4.3. User interface

When the screen size varies, the system should be designed so that it doesn't modify the user interface.

3.3.4.4. Registration

You can use your phone number, Twitter, Facebook, or Google throughout the signup process.

3.3.4.5. Payment

Software applications such as Esewa, Khalti, IMEpay, Fonepay, or any other type of mobile banking system can be utilized for online course payments or plant purchases**.**

3.3.4.6. Networking Devices

In addition to supporting cloud, the system should be able to communicate via network protocols and communication.

3.4 Other non-functional requirements

In addition to the functional requirements, the system needs the following requirements in order to perform properly.

3.4.1. Security

The system should monitor all network traffic monitored by administrator and provide secured sites for all its users.

3.4.2. Reliability

It should be simple to use and reliable.

3.4.3. Flexibility

The system should handle thousands of users without lagging or crashing the server.

3.4.4. Performance

The system needs to operate without any latency or malfunctions when a user utilizes it.

3.4.5. Maintainability

An administrator should be able to quickly manage the system, enhance its functionality, and correct any problems or malfunctions.

3.4.6. Storage

All user, instructor, and student data must be stored by the system and uploaded to cloud storage so that records can be kept in a virtual setting.

3.5 Goals Implementation

- Code needs to be effective and reusable in the future as well.
- Using object-oriented programming is an effective idea.
- To make code more understandable for developers, comments should be included in the code.
- The code should be compiled and processed using the most recent hardware and language processor versions.

4. Group Task:

4.1. Environment Model for Specification

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business operation through data movement. Rather than using textual description, depicting business requirements of applications in a diagrammatic format by representing the sequence of process steps and flow of information is easier to understand. When used through an entire development process, the results of business analysis are firstly documented. Then, the representation is refined to show how information moves through and gets changed by an application. Both automated and manual processes and represented (Nolle, 2023).

Data Flow Diagrams represent the functions or process which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The DFD structure allows us to start from an overview and extend it to a hierarchy of detailed diagrams. Level 0 of the DFD diagram; a context diagram often served as the starting point, provides a basic overview of the entire system. From there it digs down to level 1 diagram that breaks down the system primary functions into smaller, more manageable functions. This could develop further into level 2 diagram if more research is done.

Further, the Level 0 DFD captures all of system functionality and presents it as a single, continuous process. It also provides a spotlight on the main data flows that occur between the system and these outside entities this level of representation usually leaves out data stores where information may be kept to accessed (Security, 2023).

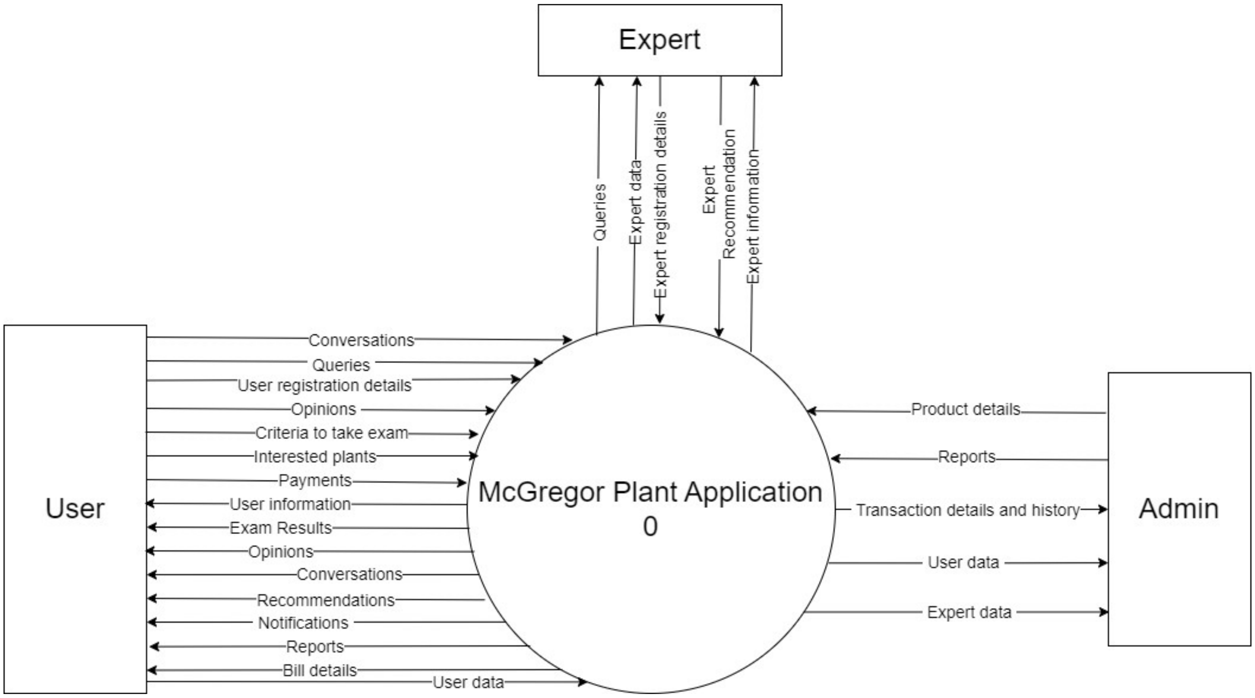

Figure 1: McGregor Application: Context level diagram.

4.2. Internal Model Specification

4.2.1. Level 1 Diagram

The goal of Level 1 data flow diagram is to provide a complete overview of the system. Level 1 data flow diagram examines the system in more detail. The single process component from the context diagram is divided into smaller processes in level 1 DFD. The primary process data storage is identified by level 1 DFD as well. The Context Level DFD must be examined before it can begin to design a Level 1 DFD. The single process needs to be divided into its component parts. All external entities, important databases, and multiple process components should be included in level 1 data flow diagram (lucidchart, 2023).

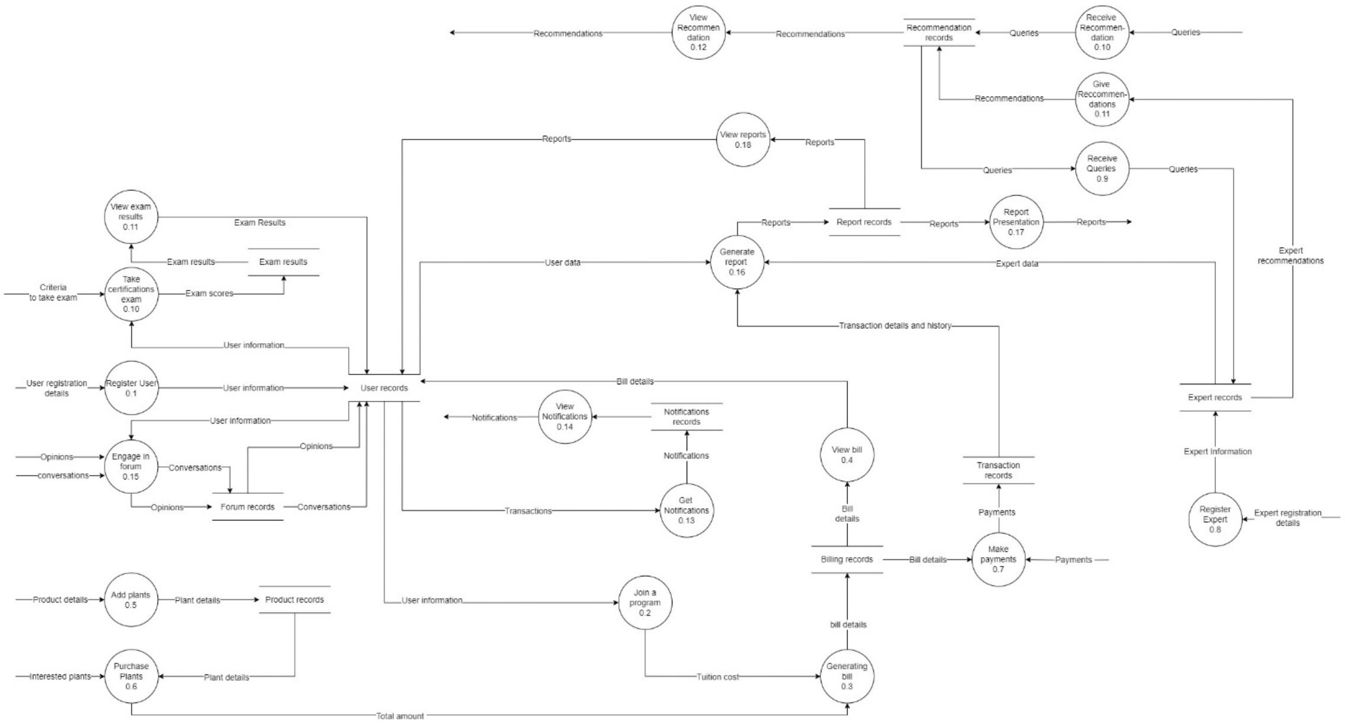

Figure 2: McGregor Plant Application: Level 1 data flow diagram.

4.2.2 Level 2 Diagram

The Level 2 data flow diagram divides the sub process found in the level 1 DFD into even more sub processes, giving an even more detailed perspective of the system. The level 2 DFD goes further deeper into parts which can be used to organize or record specific and important information regarding how the system operates (GeeksforGeeks, 2023).

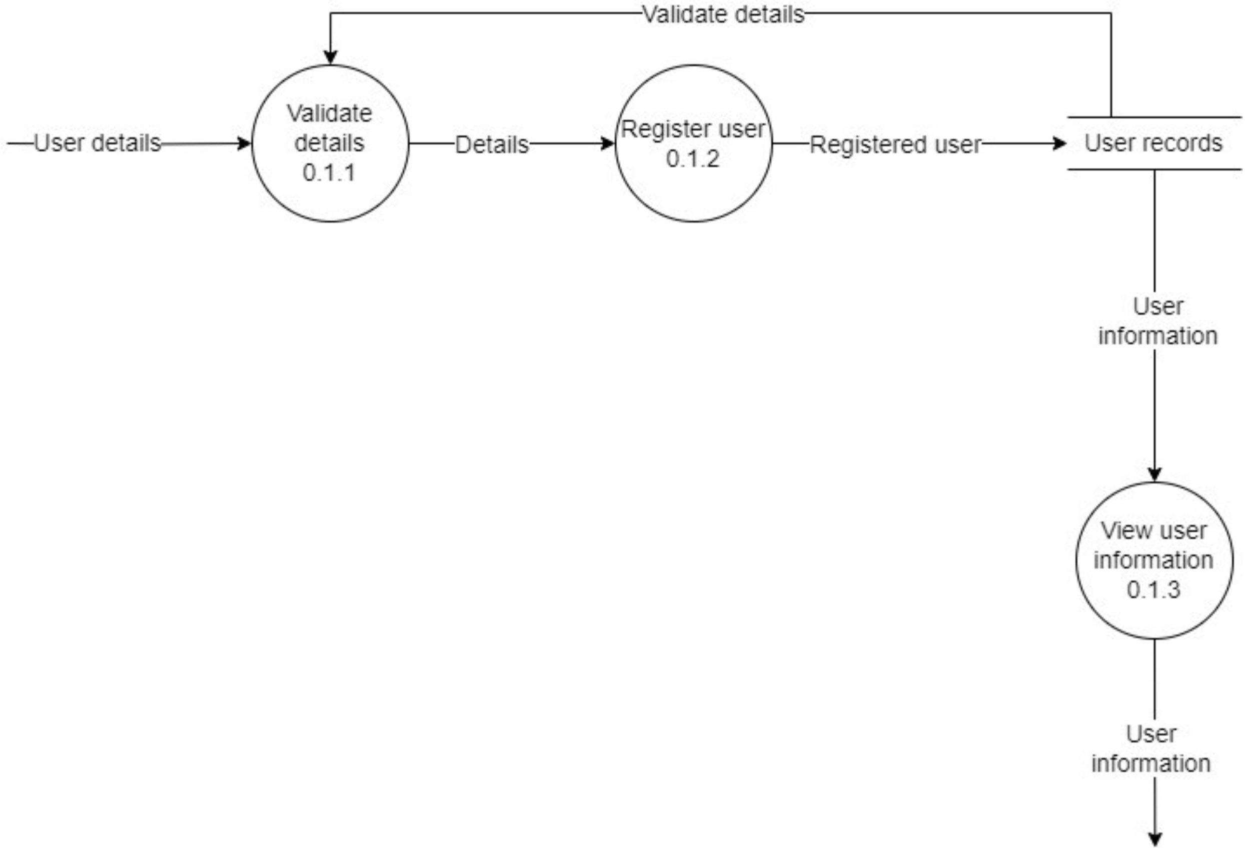

Figure 3: McGregor Plant Application: Level 2 data flow diagram of registering user.

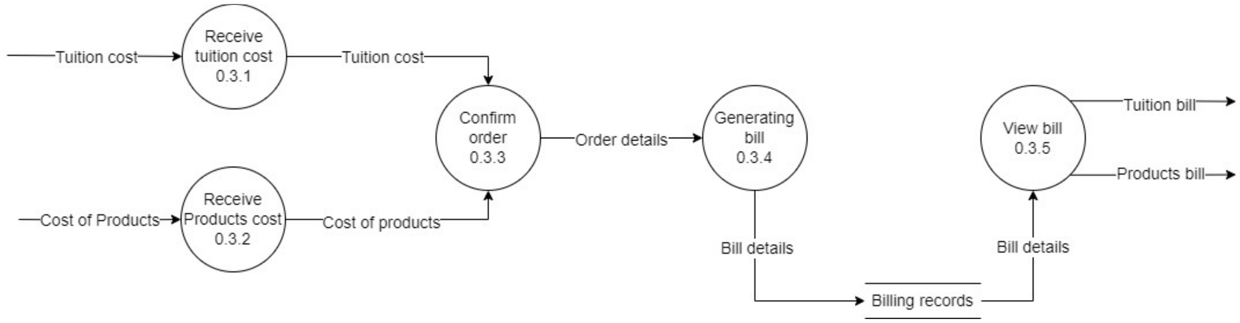

Figure 4: McGregor Plant Application: Level 2 data flow diagram of generating bill.

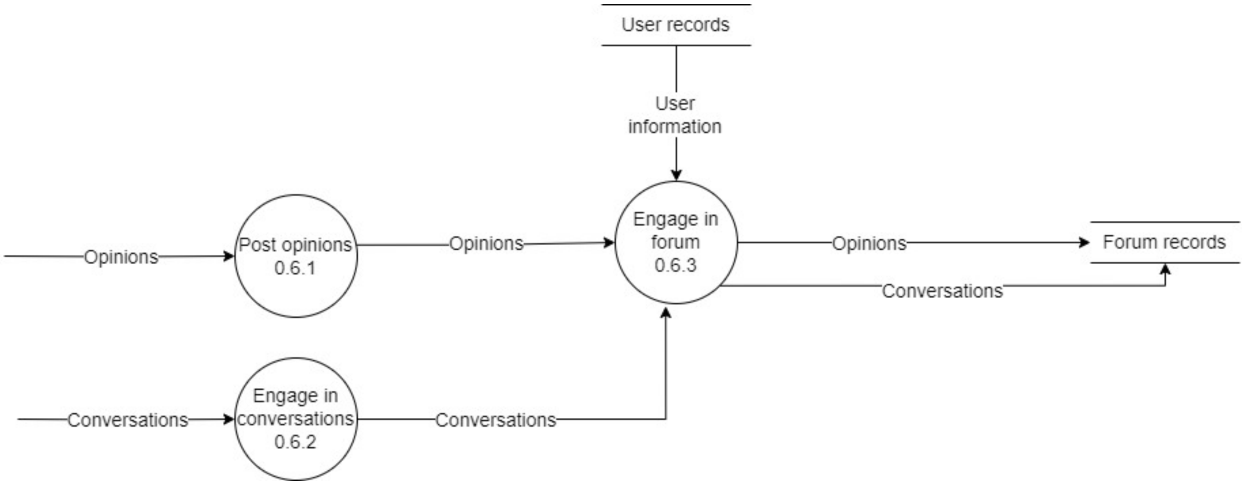

Figure 5: McGregor Plant Application: Level 2 data flow diagram of engaging in forum.

4.2.3. Entity Relationship Diagram

An Entity Relationship Diagram (ERD) serves as a visual representation of relational databases, offering a design blueprint or a means to comprehend existing databases. ERDs, also known as entity relationship models or graphs, depict the relationships among entity sets within a database, with entities representing objects or data components and entity sets comprising similar entities. These entities possess attributes defining their properties, and by delineating entities, their attributes, and showcasing relationships, ERDs elucidate the logical structure of databases (Secoda, 2023).

ERDs comprise three primary elements: entities, attributes, and relationships. Entities equate to tables in relational databases (rows), attributes are the properties associated with entities (columns), and relationships indicate interactions between entities .

Explaining Entity Relationship Diagrams (ERD):

ERDs are data models visually portraying an organization's entities and their interrelationships. Entities represent values corresponding to organizational elements like customers, products, or vendors, while relationships define how entities interact (e.g., purchases or sales). The graphical representation in an entity relationship diagram illustrates these entities and relationships.

The ERD stands out as a crucial tool for systems analysts, offering a graphical view of an organization's functional requirements. It is pivotal in designing relational databases and plays a fundamental role in modeling various system types.

The ERD, typically depicted as rectangles connected by lines, is employed to document an organization's current data architecture, and guide its future development. It serves as a blueprint for implementing data in specific software applications, especially in the realm of software engineering (Secoda, 2023).

Components of an ERD:

The primary components of ER diagrams include entities (which translate into tables) and the relationships between them.

Entities are the objects or concepts for which information is stored, such as customers, products, or orders. Entity categories offer a more structured description of entity attributes.

Attributes function as identifiers for entities, distinguishing different pieces of data. For instance, a customer entity may have a purchase attribute.

Relationships elucidate how entities interact. In a 1:M (one-to-many) relationship, one customer can have multiple sales orders, with each sales order belonging to a single customer.

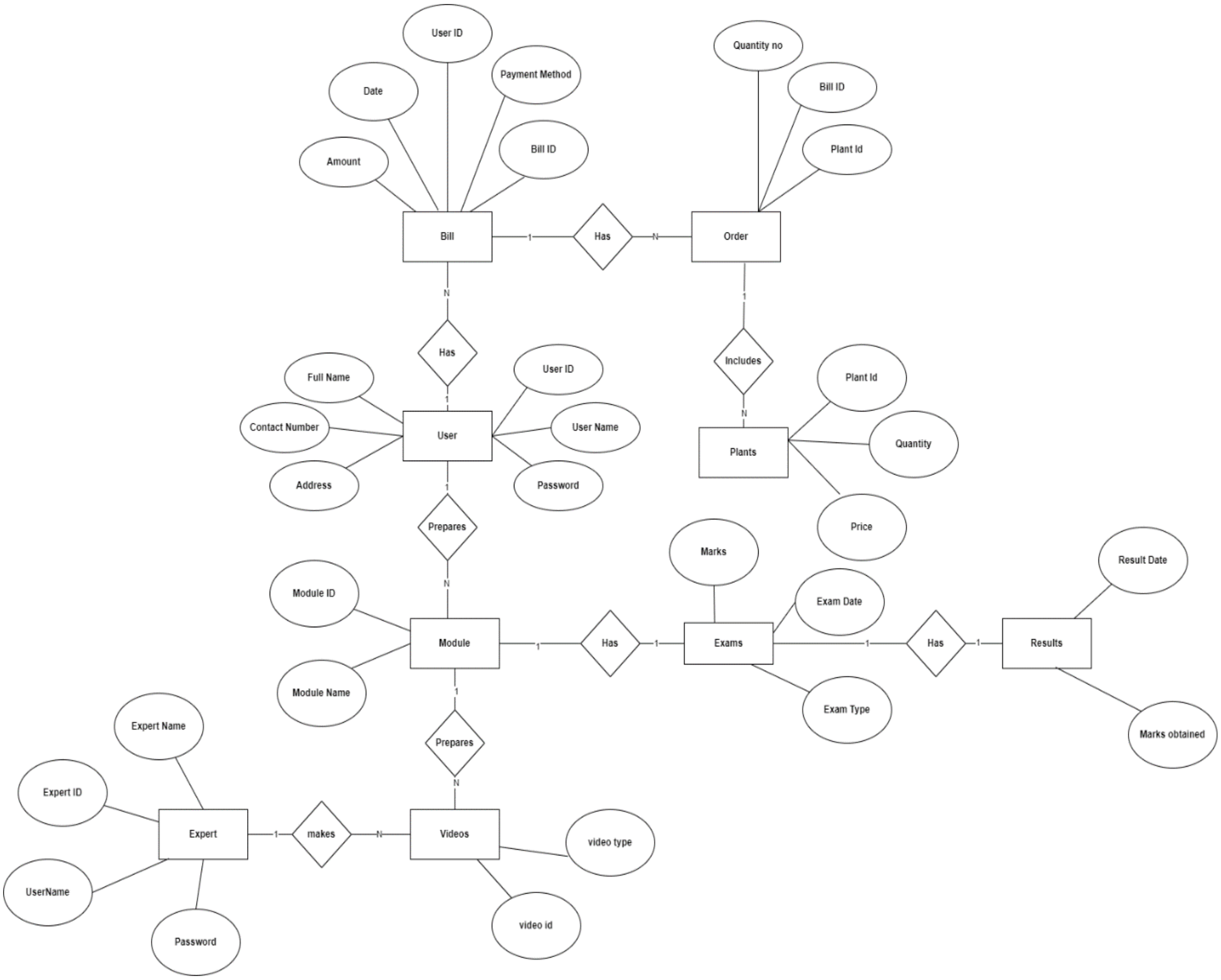

Figure 6 : Entity Diagram Relationship

The entire system can be seen using the shown entity relationship diagram. It utilizes each of the chosen data stores as an entity, considering their cardinality as well as the necessary links between them. First, cardinality functions are defined for Users, staff, and Experts. These are described below for each cardinality function:

- A single Bill only shows a single User and their transaction data, even though a User may make many purchases and hence generate multiple Bills reflecting a single User.

- A Bill could include several orders, and since each order has a unique description and can be purchased by several Users several times, it indicates a many-to-many relationship that influences the data credentials.

- A bridge entity of order line is evolved to represent the specific order in each Bills in order to minimize data redundancy.

- A bridge entity is necessary because a User may take more than one exam, whether it be a simulated exam, a physical exam, or a mental agility test, and more than one User may appear on each exam. This could lead to data redundancy in the system.

- An Experts oversees planning and producing several exam papers or videos, although each one may only have one owner.

- One exam can be designed to produce a single report containing the classified details since it includes information about the Experts and the user who is representing them.

4.2.4. Data Dictionary

A data dictionary is a collection of terms, descriptions, and attributes for data elements that are gathered for a study project, database, or information system. It is a centralized source full information regarding the content application of data pieces within a particular dataset or database. Data dictionary is important as it provides details on the contents of the database. It is often specifically used by database administrators. It provides instructions on interpretation, acceptable meanings, and representation in addition to describing the meanings and goals of data items within the overall structure of a project. It also provides metadata about data elements within a database (Library, 2022).

User Registration = Command

Expert Registration = Command

Report generation = Command

Sign Up = Username + Full Name + Password + Address + Age + Contact number

Sign In = [Gmail, Facebook, Phone number, User Credentials]

User Credentials = [Username, Email] + Password + Two Way Authentication

User Table

User Details = Full Name + Username + Password + Location

User Info = {User ID + User Details}

User_ID = integer

User_Name = String

Full Name = String

User Name = String

Password = String

Address = String

Expert Table

Expert Details = Expert Name + UserName + Password + Expert ID

Expert_ID = integer

Expert_Name = String

UserName = String

Password = String

Plants Table

Plant Details = Plant_Id + Quantity + Price

Plant Info = {Plant_Id + Plant Details}

Plant_Id = integer

Quantity = integer

Price = integer

Bill Table

Bill Details = Bill_Id + User_ID + Date + Amount

Bill_ID = integer

User_ID = integer

Amount = integer

Exams Table

Exam Details = Exam_Date + Exam_Type + Marks

Exam_Date = Date

Exam_Type = String

Marks = integer

Results Table

Results Details = Result_Date + Marks_Obtained + Result_Status

Result_Date = Date

Marks_Obtained = integer

Result_Status = String

Videos Table

Videos Details = Video_type + Video_Id

Videos_Id = integer

Video_type = String

4.2.5. Process Specification:

A process specification is a method used to record, examine, and clarify the formulas and logic procedures that are utilized to produce output data from entered input data. Its goal is to outline and specify engineering and regulatory requirements and processes. Accurate and comprehensive process requirements are necessary for reliable and consistent data (Rouse, 2012).

Process Name: Validate details

Process Number: 0.1.1

Input Parameters: User details

Output Parameters: Validated details

Process Logic: A new user that needs to be added to the system provides their details. These details are then checked to ensure that they are correct.

Process Name: Register User

Process Number: 0.1.2

Input Parameters: Validated details

Output Parameters: Registered user

Process Logic: After the details pass the validation, the user gets registered to the system and is stored in the User records datastore.

Process Name: Post opinions

Process Number: 0.6.1

Input Parameters: Opinions

Output Parameters: Opinions

Process Logic: The users can post their opinions in the forum and these opinions can be viewed by other users as well.

Process Name: Engage in forum

Process Number: 0.6.3

Input Parameters: Opinions, Conversations

Output Parameters: Opinions, conversations

Process Logic: The registered users can engage in the forum by posting their opinions and by having conversations with one another.

Process Name: Generating bill

Process Number: 0.3.4

Input Parameters: Order details

Output Parameters: Bill details

Process Logic: A bill gets generated for tuition cost and cost of products to be purchased by the user once the order gets confirmed.

4.3. Design Specification:

Structure chart is a chart derived from data flow diagram. It depicts the system in more detail than DFD. It breaks down the entire system into lowest functional modules, describes functions and sub functions of each module of the system to a greater detail than DFD. It provided visual illustrations of the logical processes that have been discovered in the design. It represents hierarchical structure of modules. At each layer a specific task is performed. Structure charts include a variety of diagrams, but they all have an inverted tree structure with boxes indicating each of the primary logical steps (point, 2023).

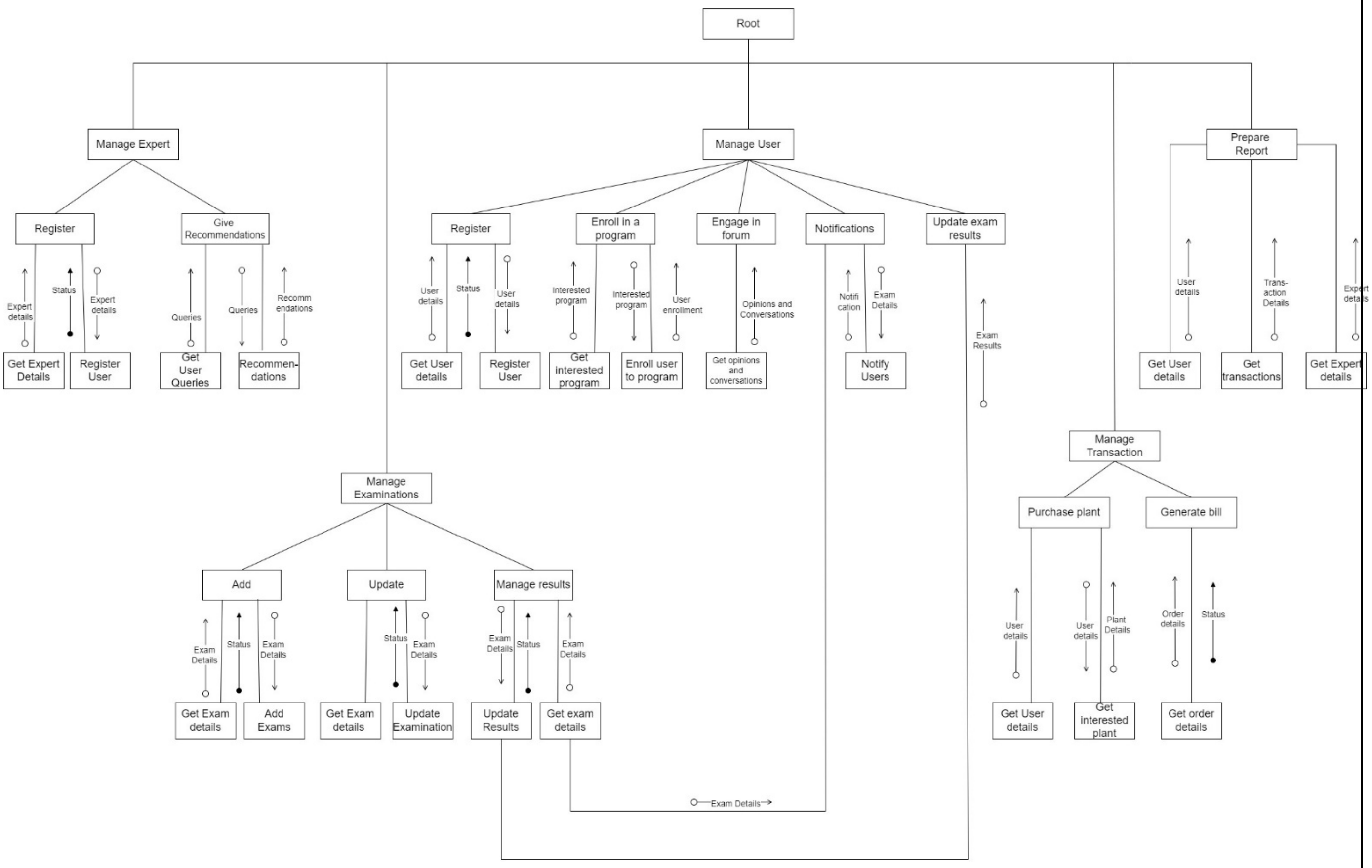

Figure 7: McGregor Plant Application: Structure chart (Design Specification)

4.4. Assignment Diary

4.4.1. Assumptions

The specifications submitted for the McGregor Institute Application are examined and adjusted to build a system that satisfies the required goals. To enhance the logic and design of the system, several assumptions are made throughout the design phase. Now, the following entities are derived from the design:

- **Expert**: To use the system, the expert needs to register in the system. The expert must enter their credentials like full name, username, date of birth, etc. Each expert will have a unique Id and this data will be saved on the database.

- **User**: To use the system, user need to register in the system by entering their credentials like full name, username, date of birth etc. Each user will have a unique Id and their data will be saved on the database.

- **Bill**: A bill will be generated after the user finalizes to purchase the plant or enroll in the course that they are interested in joining. The bill generated will have details like user Id, purchased product details, purchase quantity, unit price and the total amount. The system will give its own unique number and this data will also be stored in the database.

- **Plant**: Information like plant ID, description, name, quality of plant which are all related to plants will be stored in the database.

- **Payment**: This stores the information about the payment made by the user. After the bill has been successfully generated, payment can be done. Amount received, method of payment.

- **Exam**: After the students have been enrolled in a course, the students can take exams. Student details like exam id, full marks, exam type, and pass marks are stored in a database.

- **Result**: The number of passed and failed students, results of the exams, marks obtained are stored by this database.

Based on the above-mentioned assumptions, the team collected the following needs from the given specifications:

⇒ To finalize the test results the expert can design exam paper, mock test question papers and can check the answer of the user. Also, the expert can give recommendation to the users.

⇒ The overall system is monitored by the system admin and deals with the hiring process of experts and staffs.

⇒ To enroll in the courses, user must register.

⇒ To determine the knowledge of a user, mock test created by the expert can be used.

⇒ To generate the bill of the plant sold, a staff is assigned for the transaction process.

4.4.2. Omissions

There were some omissions left in the project despite taking all the client's requirements into consideration. A few are outlined below:

- There is a threat of sensitive information being due to the absence of addressed security protocols, encryption methods and access control.
- The user interface aspect of the application is absent, potentially causing users difficulty in adapting to the lack of UI.
- Data backup and system backups are not specified.
- Details regarding the application's compatibility with multiple operating systems have not been mentioned.

4.4.3. Group Member Responsibilities

Collaboration was a key strength within our team as we tackled the design of the system. Each group member actively engaged in cooperative efforts, offering support through task reviews, and participating in discussions that lasted the whole project. These interactions ensured a comprehensive understanding of the complete system process among all team members. Our cooperative efforts were successful because of our dedication to supporting one another and exchanging ideas, which created a dedicated team atmosphere throughout the project. The responsibilities assigned to each group member are as follows:

Group member	**Responsibilities**
Sajiya Khatoon	Complete the individual task of **Make Payment.** Create the Entity Relation Diagram. Assist in documentation

Pritishma Kunwar	Complete the individual task of **Purchase Plant.** Create the Environmental model specification. Make the internal model specification. Finish the structural charts and assist in report documentation.
Neha Pokhrel	Complete the individual task of **Report presenting**. Create specifications for the software requirements. Create the data dictionary and assignment diary. Contribute to the report documentation.
Aayushree Dahal	Complete the individual task of **Joining the program.** Finish the project charter. Create the Process specification. Contribute to the report documentation.
Apreksha Gurung	Complete the individual task of **Taking Certification Exam**. Assist in making the environment specification model. Engaged in the project assignment diary section. Contributed to the report documentation.

Table 2: Group Member Responsibilities.

Group Meetings

Date	Time	Location	Discussion
20/12/2023	8:00 AM To 9:00 AM	London Block	On the first day, we discussed the assignment and issues of the coursework and divided the task among the group members.
21/12/2023	10:00 AM To 11:30 AM	Canteen	We went over the requirements of the McGregor Institute of Botanical Training in great depth, created a work plan, and conducted a requirements analysis.
22/12/2023	11:00 AM To 1:00 PM	Canteen	We discussed the functional specifications for the system, the System Required Specification (SRS), and the Project charter.
25/12/2023	8:30 AM To 10:00 AM	Coffee Station	We design the DFD and Data Dictionary for the entire system. We met our module lecturer and dispelled any questions we had regarding the requirements.
27/12/2023	8:00 AM To 11:00 AM	Kumari Block	We discussed the mistakes in the design and solved them. We also began working on a Structure Chart for the overall system.
28/12/2023	11:00 AM To 1:00 PM	Coffee Station	We discussed with our module tutor about the difficulties we were having understanding the concepts of DFD and Structure Chart. Also, we discussed the report formatting and structure.
29/12/2023	11:00 AM To 1:30 PM	Canteen	We discussed and examined the Entity Relationship Diagram (ERD), Structure Chart and Data Flow Diagram (DFD).
31/12/2023	11:30 AM To 2:00 PM	Library	Each of the members worked on their individual task. We discussed and corrected each other’s design implementations after reviewing them.

2/12/2024	10:00 AM To 1:30 PM	Kumari Block	We met with the module lecture and talked about the issues and problems faced.
3/12/2024	10:00 AM To 11:00 AM	Kumari Block	We designed the charts and diagrams, and we combined individual tasks to assemble the whole report.
4/12/2024	10:00 AM To 12:00 PM	Nepal Block	Before submitting the work, we did a full inspection, in-depth evaluation, and report formatting.

Table 3: Meeting Log.

5. Individual Task

5.1 Make Payment

Name: Sajiya Khatoon

London Met ID: 22067685

The system can take register information according to this module function. The member provides the system with the details for this module function. Every member's detail is verified and entered the Information Data.

5.1.1. Environmental Model Specification

Context level diagram is a different term for the Level-0 DFD model. The system is depicted as a single process with external entities because it is meant to be an abstraction perspective. The system shows up as a single bubble with incoming and outgoing arrows signifying the data input and output.

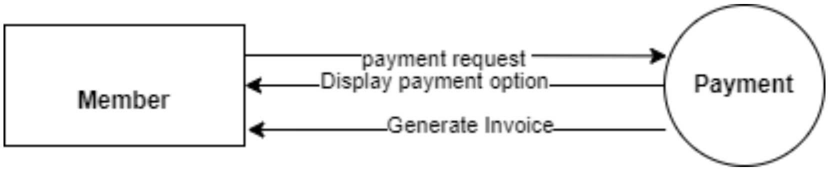

Figure 8: Payment: Level 0 Data flow diagram

The context level diagram of the primary process represents the environmental model specification of a register in system function. A member must enter their information into the system to register. The system creates membership credentials after a member registers. Using the provided membership credentials, the member logs into the system.

5.1.2. Internal Model Specification

Level 1 and Level 2 data flow diagrams (DFD) depict a member making payment function's internal model specification. All the data flows that take place during the process between the external entities, the process, and the data repositories are represented by these diagrams. The DFDs for levels 1 and 2 are as follows:

5.1.2.1. Level 1 Diagram

In 1-level DFD, the context diagram is divided into numerous bubbles or processes. At this level, the primary features of the system are highlighted, and the high-level 0-level DFD process is divided into smaller processes.

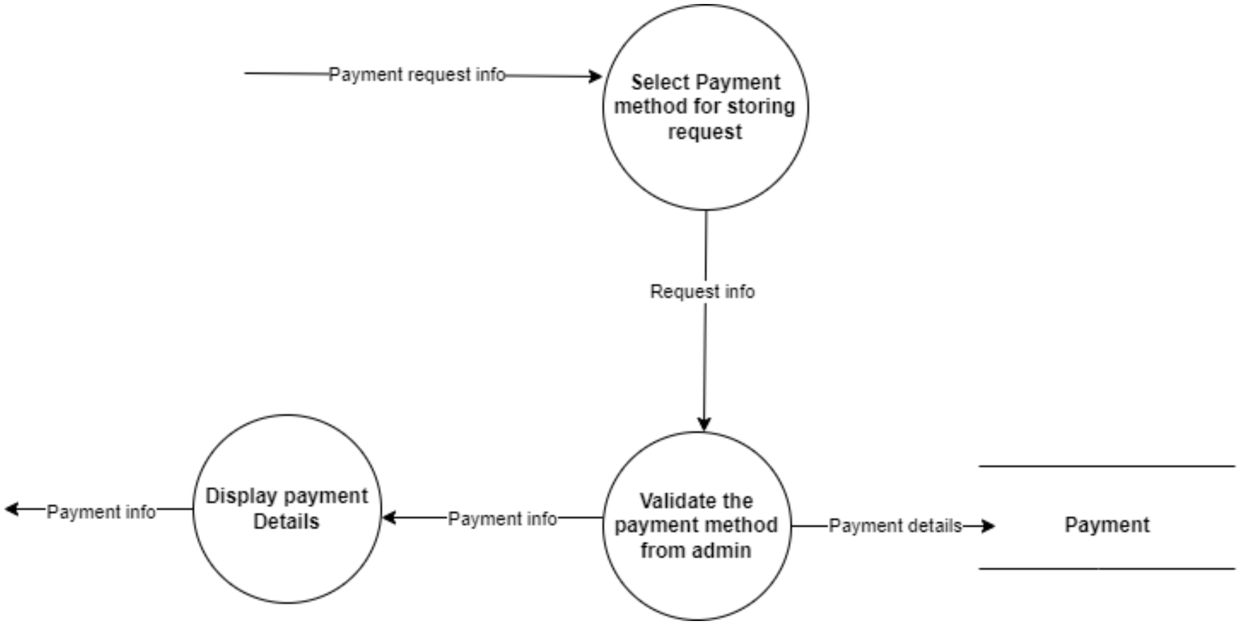

Figure 9 : Level 1 Data Flow diagram

In Level 1 DFD, a member provides their details to the system. The detail of the member is validated and register into the system. Members' login credentials are generated by the database, which also retains their personal information.

5.1.2.2. Level 2 Diagram

Parts of 1-level DFD are further examined in 2-level DFD. It can be used to plan or document every detail that is particular to or required to understand how the system operates.

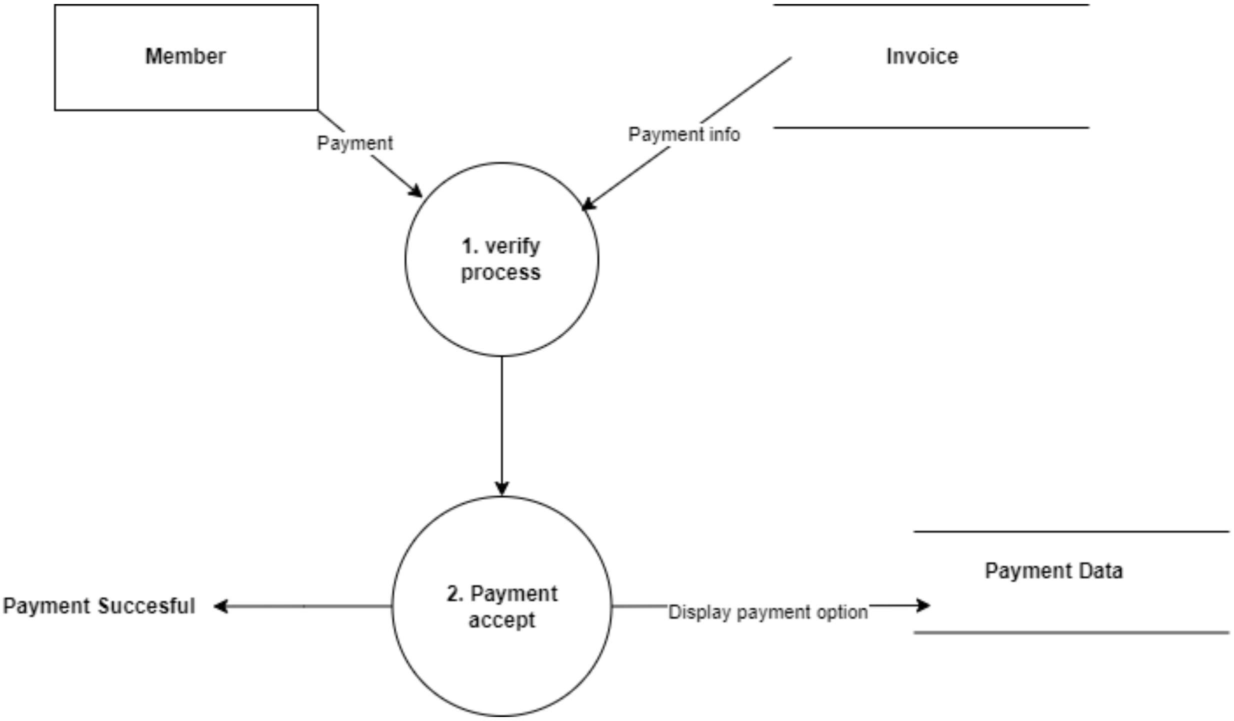

Figure 10: Level 2 Data Flow Diagram.

The payment process is then processed by further deconstruction so that the member data are passed. After payment validation, it is sent to the member information data store with valid member details passed. It can now be processed further to show the member that has been registered.

5.1.3. Design Specification

5.1.3.1. Structure Chart

The Structure Chart illustrates the modules' hierarchical structure. It breaks down the entire system into its most basic functional modules and provides a more thorough analysis of the main functions and sub-functions of each module.

The primary functions in the structure chart are member payment method. In order to verify a member's details and payment method, the function offers member details. The purpose of the cheque detail is to validate the member details.

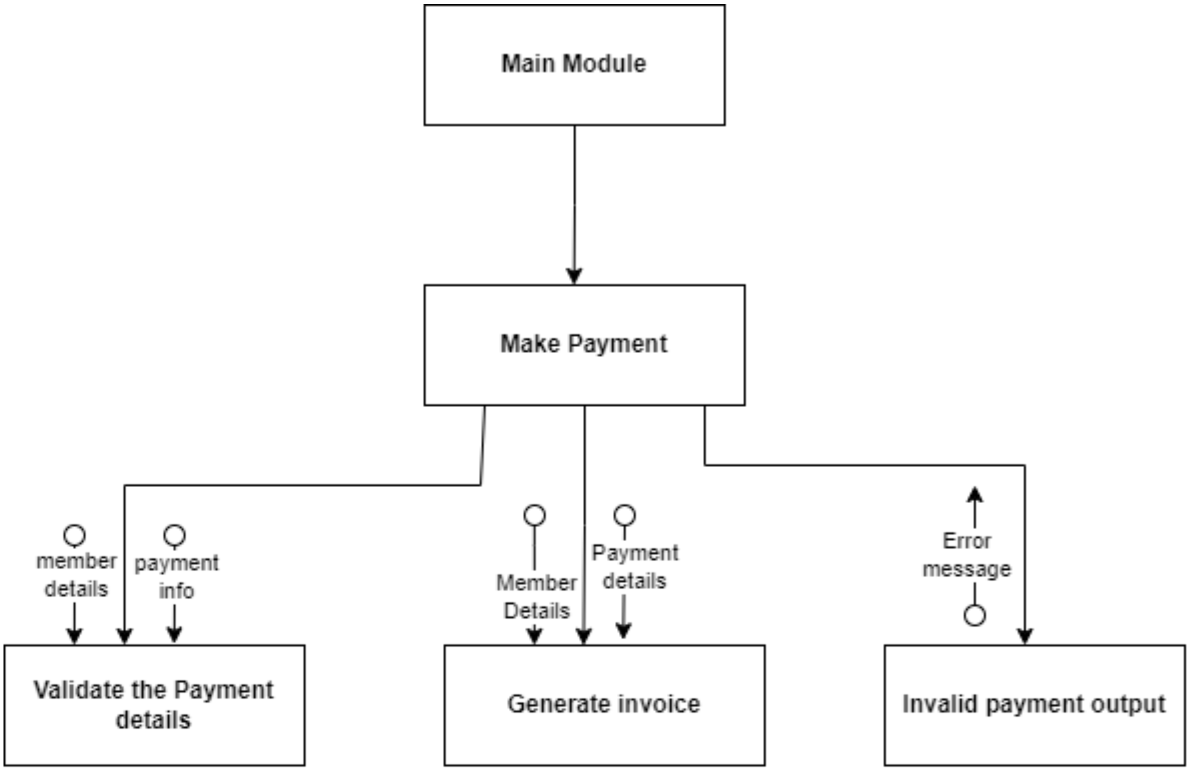

Figure 11:Structure Chart

Validate Payment and Invalid Payment are the two modules in this diagram that specify the data flow of member details to and from each of the submodules. They were therefore either removed or validated.

5.1.3.2. Module Specification

MODULE NAME: Make Payment

PURPOSE: The following module proceeds with the transactional information about the kits being purchased by the members which is finalized by the invoice generation.

PSEUDOCODE

```
DO
    DECLARE Invoice Details
    INITIALIZE Member ID
    IF Member ID MATCHES Member ID FROM Member Details
        IF Plant MATCHES Plant Details FROM Plant AND Quantity GREATER THAN 0
            SET Invoice Details EQUAL TO {Invoice ID, Plant Info, Plant Description, Order Quantity, Total Amount}
        ELSE DISPLAY "Invalid payment please try again."
    ELSE DISPLAY "Invalid Member ID"
END DO
```

INPUT PARAMETER: Member ID, Plant Details, Plant description, Total Quantity

OUTPUT PARAMETER: Invoice ID, Invoice Details, Invoice Info

GLOBAL VARIABLES: Plant Info Database, Invoice Database, **Member Database**

LOCAL VARIABLES: Invoice Details, Member ID, Plant Description, Quantity

CALLED BY: Main Module

5.2. Purchase Plant

Name: Pritishma Kunwar

London Met ID: 22067976

In this module function of the application, the users can make purchase requests to purchase the plants that they are interested in. In return, the system generates a bill and presents it to the user.

5.2.1. Environmental Model Specification:

In an environmental model specification, a concise representation of the overall system is provided. It depicts the interaction of the system and the external entity. It comprises three elements: a circle representing the process, arrows denoting data flow, and a rectangle symbolizing an external entity. This diagram portrays the whole system as a solitary process and does not give details of its internal structure (Mark von Rosing, 2015).

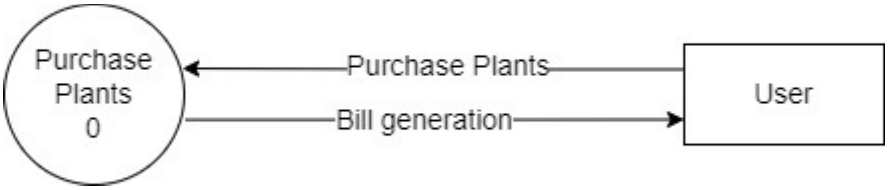

Figure 12: Context level diagram of Purchase Plants.

In this case, the external entity is a **User** who inputs details for purchasing plants. In response, the process/system generates a bill and sends it as output. This is just an overview of the whole system and will be further expanded in the internal model specification.

5.2.2. Internal Model Specification

5.2.2.1. Level 1 Diagram

The Level 1 data flow diagram provides a broader overview of the system compared to the context diagram but goes into more detail. It breaks down the single process depicted in the context diagram into sub-processes and introduces a datastore to store the output data generated by these processes. As more processes are included, the diagram will require more data flows and data stores to connect them (Lucidchart, 2023).

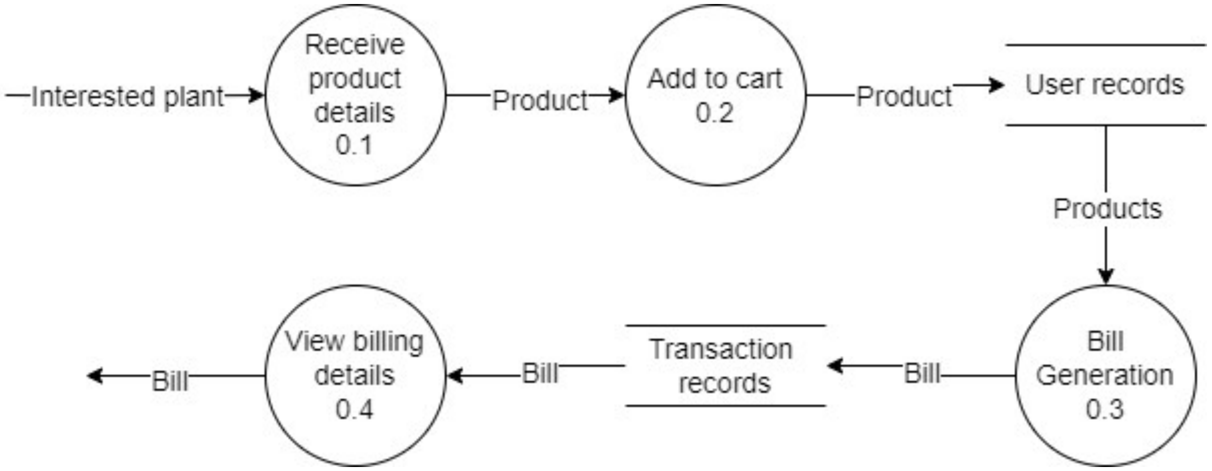

Figure 13 13: Purchase Plants: Level 1 Data Flow Diagram.

Here, the user inputs the plant they are interested in and adds it to the cart. The system receives the input (interested plant), includes the product in the cart, and stores this information in the **User records** datastore. Subsequently, the product details are retrieved from the datastore and forwarded to another process to generate the bill. This bill also gets stored in a datastore named **Transaction records**. Additionally, the user has the option to view the billing details.

5.2.2.2. Level 2 Diagram

Level 2 data flow diagram provides an even more in-depth view of the system as the subprocesses of the Level 1 data flow diagram are further broken down. Each of these broken-down pieces are presented independently along with the association of data flows and data stores (geeksforgeeks, Levels in Data Flow Diagrams (DFD), 2023).

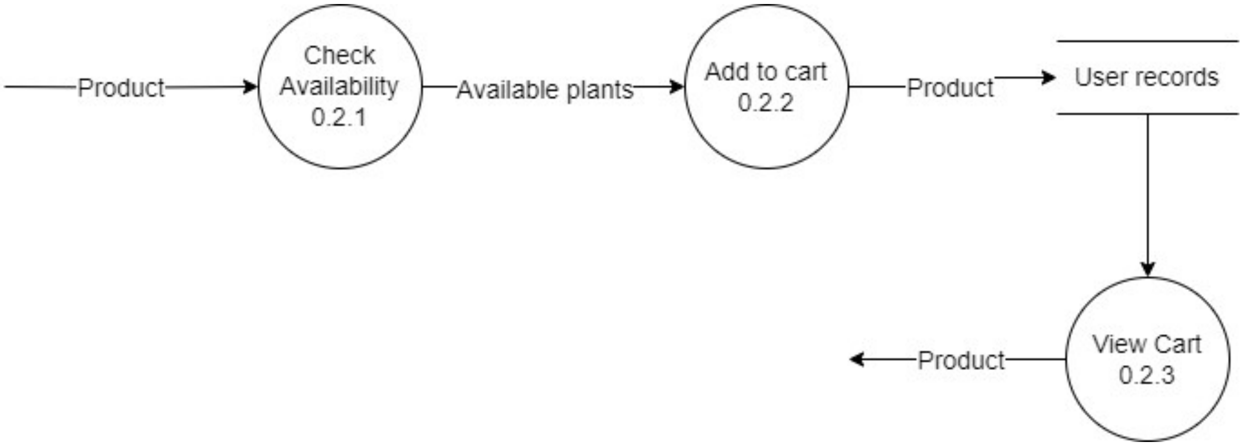

Figure 14 14: Purchase Plants: Level 2 Data Flow Diagram .

In this instance, the **Add to Cart** process from the Level 1 data flow diagram is detailed further. To add an item to the cart, the system first validates the product entered by the user. Upon confirming its validity, the item is added to the cart and stored in the **User records** datastore.

5.2.3. Design Specification

5.2.3.1. Structure Chart

Structure chart is the hierarchical representation of modules where the entire system is broken down into the lowest functional modules. It describes the functions of each module in detail. The structure chart has different components such as modules, call line, parameters, and control parameters (geeksforgeeks, Structure Charts, 2024).

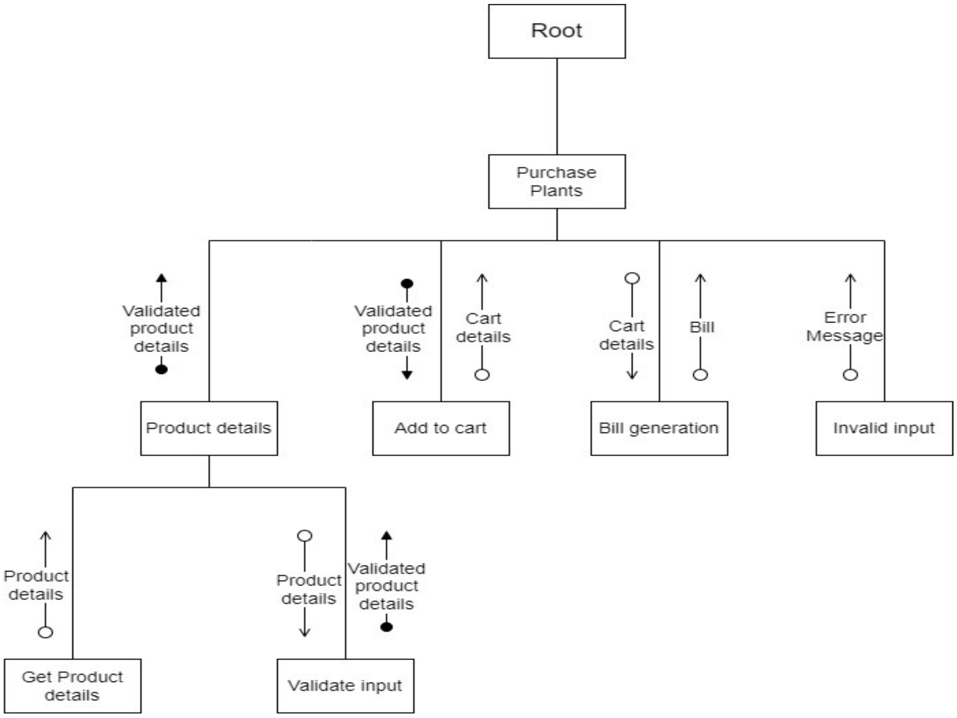

Figure 1515: Structure chart of Purchase Plants.

In this case, the structure chart depicts the overall functioning of a sub-system that allows users to purchase plants. Here, the structure chart of the function **Purchase Plant** is provided by the **Root**. The **Product details** function gets the validated product details through sub-functions **Get Product details** and **Validate input**. The **Add to cart** function gets the cart details once the validated product details are given as input. Finally, the cart details are given as an input to the **Bill Generation** function and a bill gets generated and sent to the **Root** function. If the entered input is invalid the **Invalid input** function shows and error message

5.2.3.2. Module Specification

Module Name: Purchase Plants

Purpose: The purpose of making this module is to enable users to purchase plants and get an order bill in return.

Pseudocode

```
DO
    INPUT interested product
    IF Product is available
        ADD Product TO cart
        IF cart. IS EMPTY
            DISPLAY “Cart is empty”
        ELSE
            SET Bill EQUAL TO Bill details
    ELSE
        DISPLAY “Product unavailable”
END DO
```

INPUT PARAMETER: interested product

OUTPUT PARAMETER: Bill details

GLOBAL VARIABLES: None

LOCAL VARIABLES: cart

CALLED BY: Root

5.3. Report Presentation

Name: Neha Pokhrel

London Met ID: 22068755

PURPOSE: The purpose of this module function is to prepare a report of the members.

5.3.1. Environmental Model Specification

Level-0 Data Flow Diagram shows the complete requirement of a report in one single bubble,marked by connecting lines with input and output data .

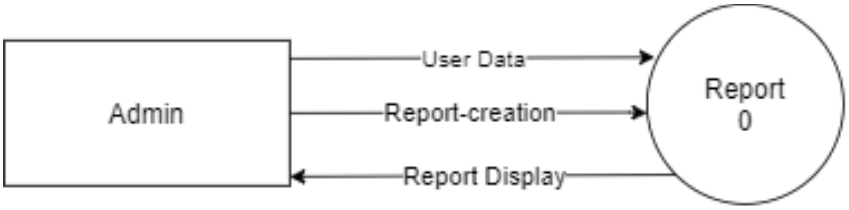

Figure 16: Report Presentation: Level-0 Data Flow Diagram.

The above figure represents the Level-0 Data Flow Diagram of the function “Report Presentation". Here the admin crestes report by using user data. And the system displays the generated report.

5.3.2. Internal Model Specification

Internal Model Specification of “Report Presentation” is represented by Level-1 and Level-2 Data Flow Diagrams (DFD). Following are the DFDs for Level-1 and Level-2 of the module.

5.3.2.1. Level 1 Diagram

Level 1 DFD shows the data flow between Level-0 DFD and is further subdivided. Here the basic processes and information sources are mentioned.

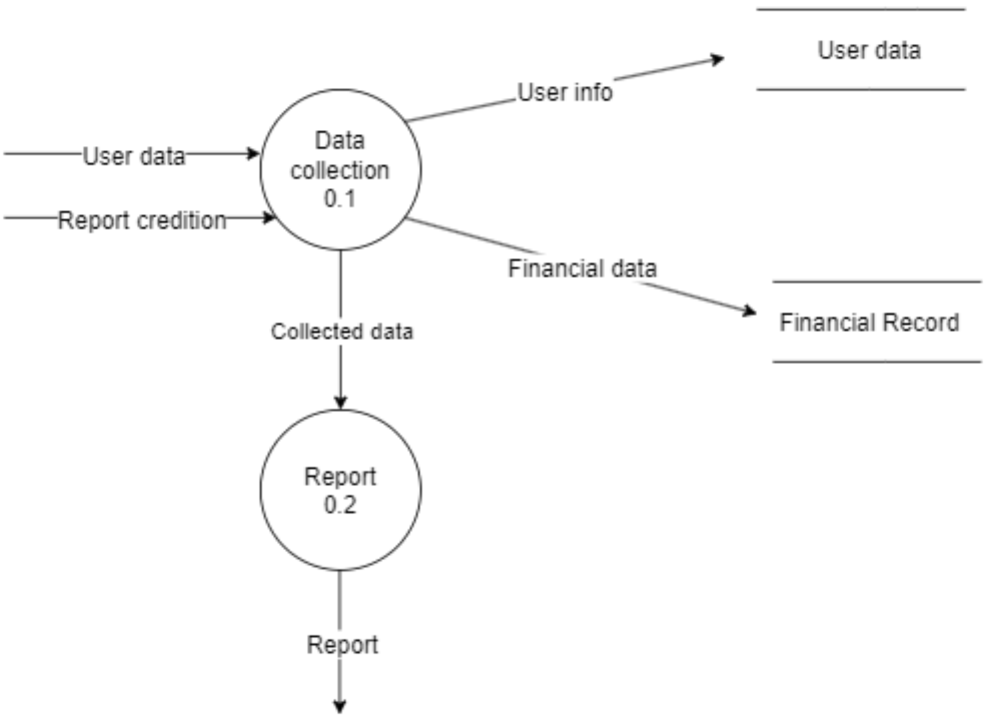

Figure 17 :Report Presentation:Level 1 DFD

The above Level 1 DFD of the function “Report Presentation” consists of two processes (Data collection) and (Report) and two Data Dictionary File name (User data, Financial Record).Firstly we have user data and report credition where it is inserted in the system through validation process. When the user info and financial records are collected, it goes through report preparation process. The Report can be viewed through the system.

5.3.2.2. Level 2 Diagram

It is a furthermore process done after Level-1 DFD. The components of Level 1 DFD are explained in deep in Level 2 DFD.

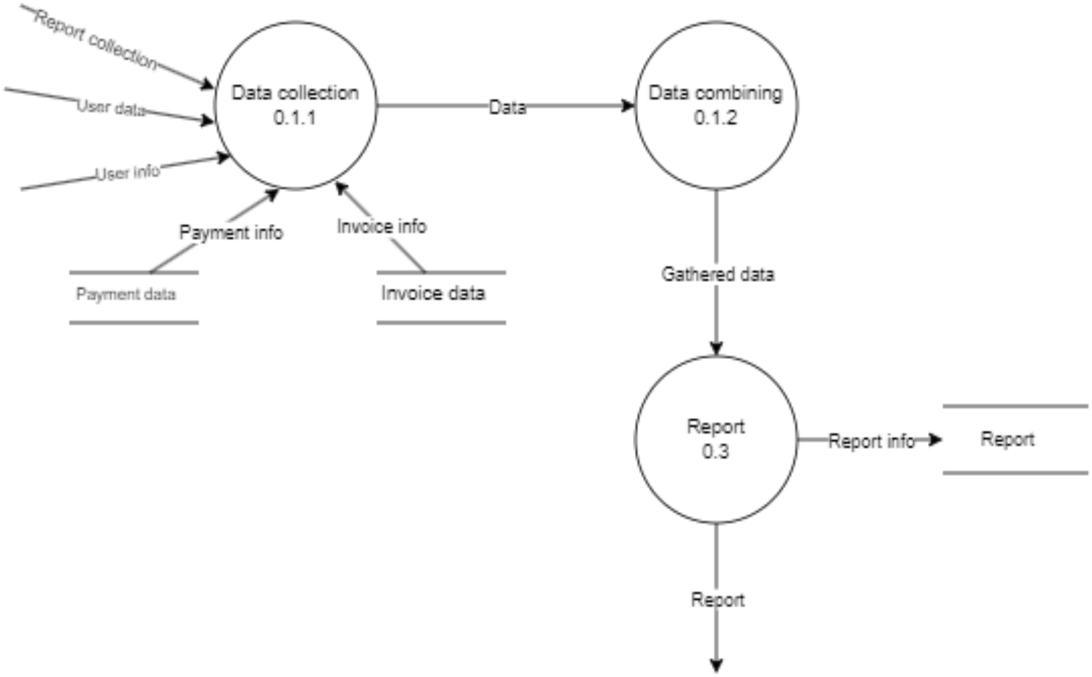

Figure 18: Report Presentation: Level 2 DFD

Level 2 DFD of the function "Report Presentation" consists of three processes (Data collection, Data combining and Report). Report collection, user data and user info are inserted in the Data collection. Through Report the report info can be viewed.

5.3.3. Design Specification

5.3.3.1. Structure Chart

The hierarchical organization of modules is represented by the organization Chart. It deconstructs a system into its most basic functional modules and provides a more thorough explanation of the main and auxiliary functions of each module. The main topic of this essay is a thorough discussion of structure charts (geeksforgeeks, 2024).

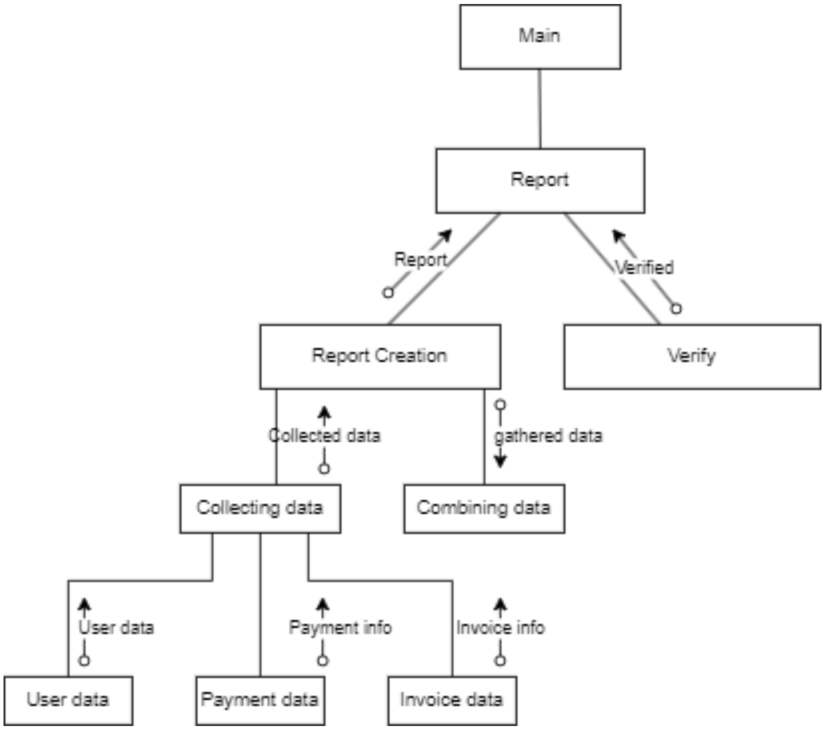

Figure 19: Report Presentation: Structure Chart

The above diagram depicts the Structure Chart of the function “Report Presentation”. It consists of 1 module (Report) and two sub-modules(Report creation and verify).Report Creation furthermore has two modules(Collecting and combining data).Through the Report Presentation we can collect and combine data and also examine user data, payment data and invoice data.

5.3.3.2. Module Specifications

Module Name: Report Presentation

Purpose: The purpose of this module is to create the report, verify it and view result.

Pseudocode:

```
DO
        INPUT User Data
        DO
             IF User Data is Valid
               THEN Create Report
           ELSE
                  Report Creation Cancelled
          END DO
   END DO
```

INPUT PARAMETERS: User Data

OUTPUT PARAMETERS: Report

GLOBAL VARIABLES: None

LOCAL VARIABLES: None

CALLED BY: Main

5.4 Join the Program

Name: Aayushree Dahal

London Met ID: 22067884

As part of Joining the program user can interact with other users and assist the institute to achieve the goals which make it easier for user to register for the joining the program by displaying the following Undergraduate and Postgraduate degree.

5.4.1. Environmental Model Specification

The context level DFD symbolizes the whole process of joining a program through the system. It points out how users engage with one other by following the requirements and providing necessary information while completing the joining process.

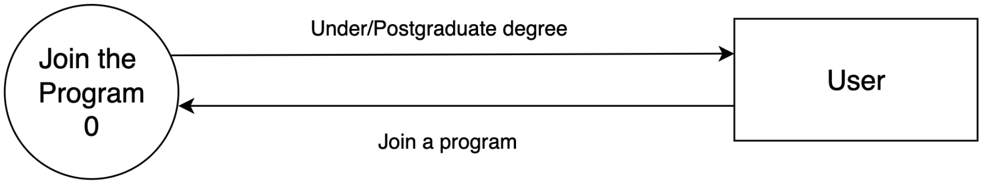

Figure 2017: Context Level Diagram of Join the program.

The following diagram shows that user are requires to comply with specific guidelines to join the program, by providing details about the Undergraduate and Postgraduate degree. This system focuses on a single process and set on both in and out from external entities.

5.4.2 Internal Model Specification

5.4.2.1. Level 1 Diagram

The level 1 DFD shows a more detailed approach than the context diagram it gives an overall understanding of the system. The level 1 DFD start with the single process node from level 0 and breaking it down into subprocesses. It may require more data flows and data storage to connect the processes once they are added.

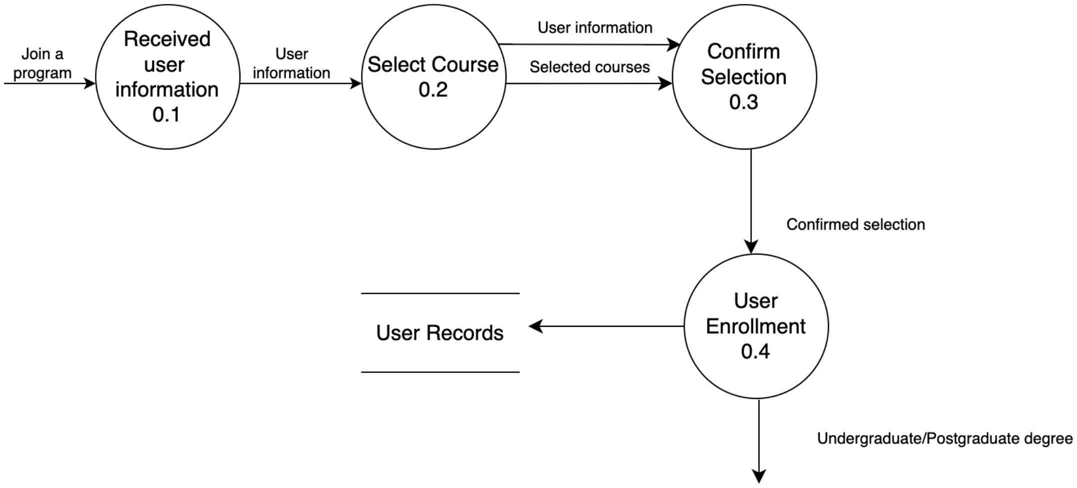

Figure 2118: Level 1 DFD: Join the Program.

The level 1 DFD is a representation process of joining the program. The system received the user information for joining a program in this section and about the selected courses they have taken. The next steps involve confirming the user selection and user enrollment process of undergraduate/postgraduate degree. The following user records is kept in the appropriate databases.

5.4.2.2 Level 2 Diagram

The level 2 DFD includes more details on subprocess and goes further into certain topics. It can be applied to the arrangement or documentation of certain data. The arrangement clarifies how the system works and assists in a better understanding of the individual processes.

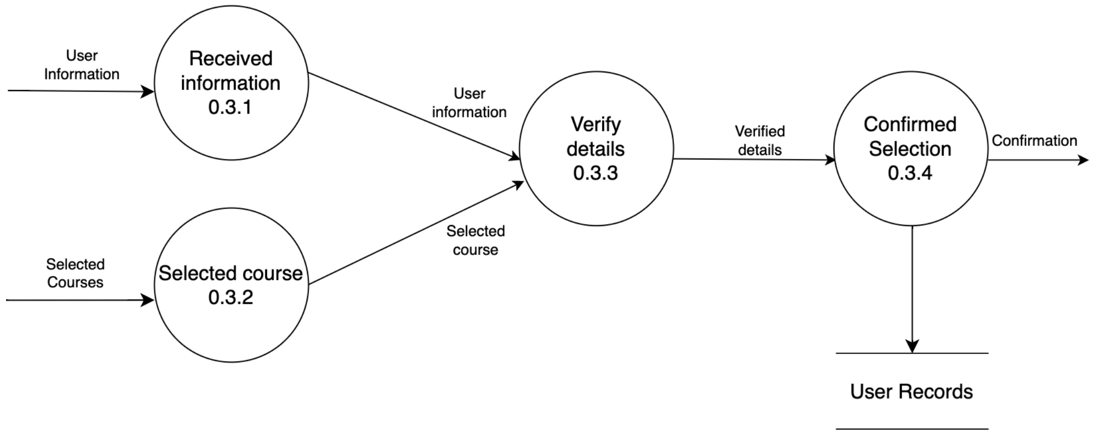

Figure 2219: Level 2 DFD: Join the Program.

Level 2 DFD further breaks down a system in specific details and information to illustrate the data flow between them. The following details is provided for the confirmed selection of user from the level 1 data flow diagram. It includes the required user information and a selected course with a proper Verify details procedure. Following that, the user qualification for joining the program and the confirmed selection. The corresponding databases have the following user records.

5.4.3 Design Specification

5.4.3.1 Structure Chart

The structure chart shows the hierarchical link between the modules of the system. A module is represented by each box and the arrows which show the flow of control between the modules. The structure chart provides a high-level overview of the system and illustrates its various modules to accomplish the ultimate objective of joining the Program.

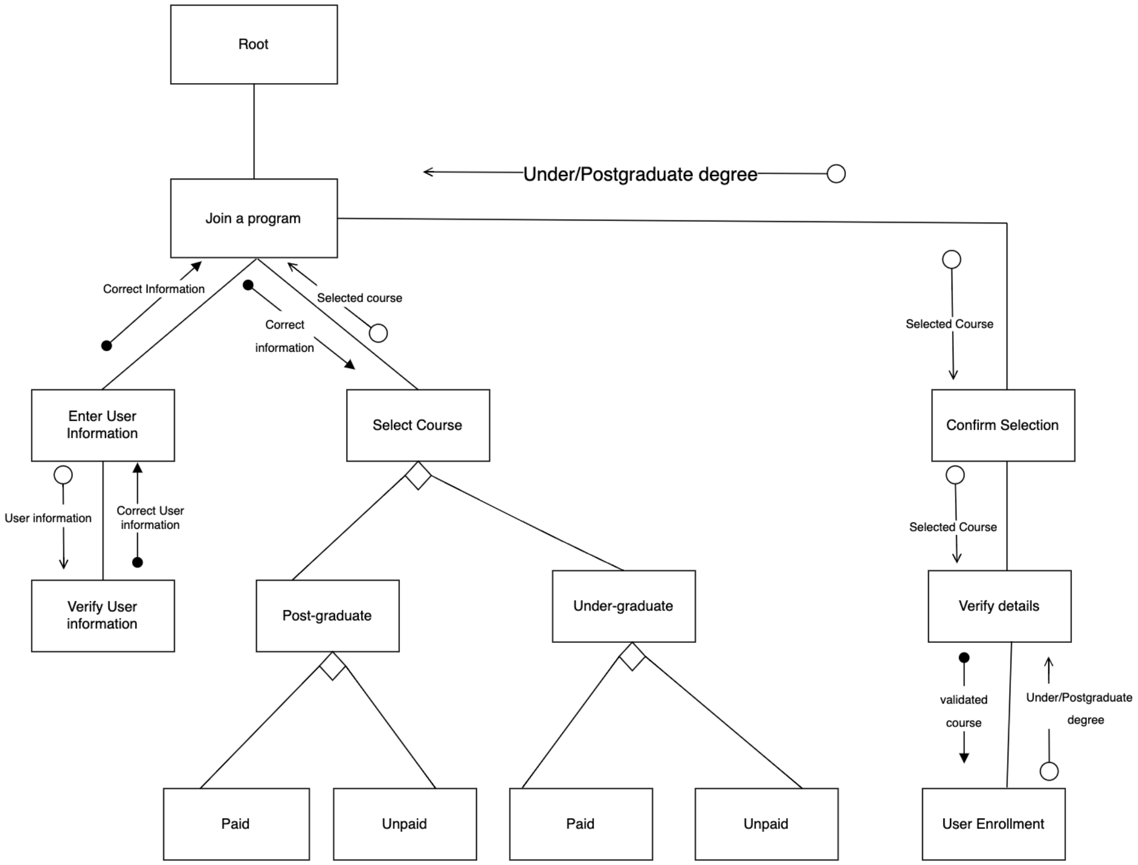

Figure 2320: Structure Chart of Join the Program.

The diagram from root shows the structure chart for joining a program. The Structure chart displays numerous steps such as input of user information and selecting course with accurate information, once the correct information is entered it get verified with the user information. Following, course selection is used to provide postgraduate and undergraduate degrees including paid and unpaid options. The module oversees confirming the selection of course with verifying the details and finishing the final user enrollment process with validate course of undergraduate/postgraduate degree. It basically displays the connection between the system components and its hierarchical arrangement.

5.4.3.2 Module Specification

Module Name: Join the Program

Purpose: The purpose of this module is to assist people in registering in the several education program which the institute offers and records their information and progress.

Pseudocode:

```
DO
    INPUT User information
        DO
            IF User information is valid
                THEN Select Course
                    IF Post-graduate is true
                    ELSE IF Select Under-graduate
                        THEN select paid or unpaid
            ELSE
            DISPLAY “Invalid User information”
        END DO
            IF Select Course is true
                THEN Confirm Selection and Verify Details and User Enrollment
            ELSE
                Enrollment cancelled
END DO
```

INPUT PARAMETERS: User information

OUTPUT PARAMETERS: Confirmation message, Cancellation message.

GLOBAL VARIABLES: None

LOCAL VARIABLES: Postgraduate, Undergraduate, Confirm Selection
CALLED BY: Root

5.5. Take Certification Exam

Name: Apreksha Gurung

London Met ID: 22068145

The primary goal of this Take Certification Exam is to show data flows through the system during the registration, examination, and evaluation phases. This system accepts the details from the user in this module function. It aids in the comprehension of the certification system's overall information flow and interactions between its many parts by users.

5.5.1. Environmental Model Specification

The Context Level Diagram, also known as Level 0 DFD provides a simplified and clear overview of the system and its interactions with the external environment. The system is displayed as a single process with incoming and outgoing arrows signifying the data input and output.

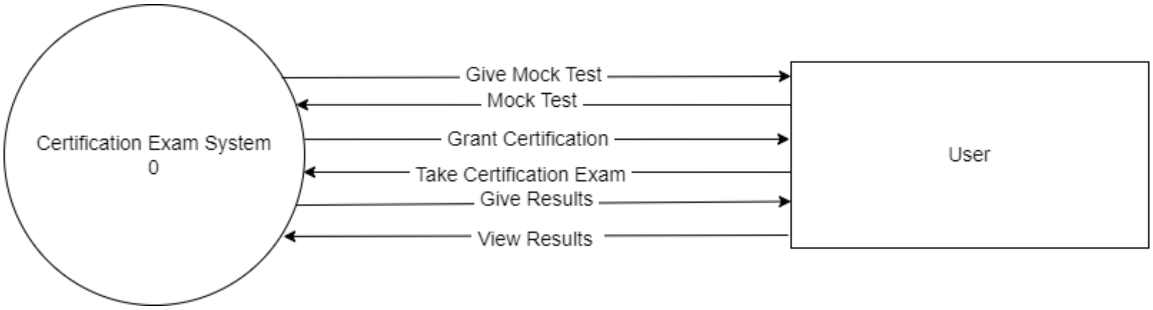

Figure 2421: Context Level Diagram of Take Certification Exam.

A context level diagram is used by the primary process to show the environmental model specification. Here, the User is the only entity, and the Certification Exam System is the only process. Data flow from users to mock test, take certification exam, and Check result make up the process. The user can access Give mock test, Grant Certification, and Give results from the Certification Exam System.

5.5.2. Internal Model Specification

A certification exam system's internal model specification is expressed by level 1 and level 2 Data Flow Diagram (DFD). All the data flows that take place during the process between the external entities, the process, and the data repositories are represented by these diagrams. The DFDs for level 1 and level 2 are as follows:

5.5.2.1. Level 1 Diagram

The context diagram's primary process is broken down into its primary sub processes or functions in this diagram. The data flows between various sub processes are displayed, and each is depicted as a distinct process bubble.

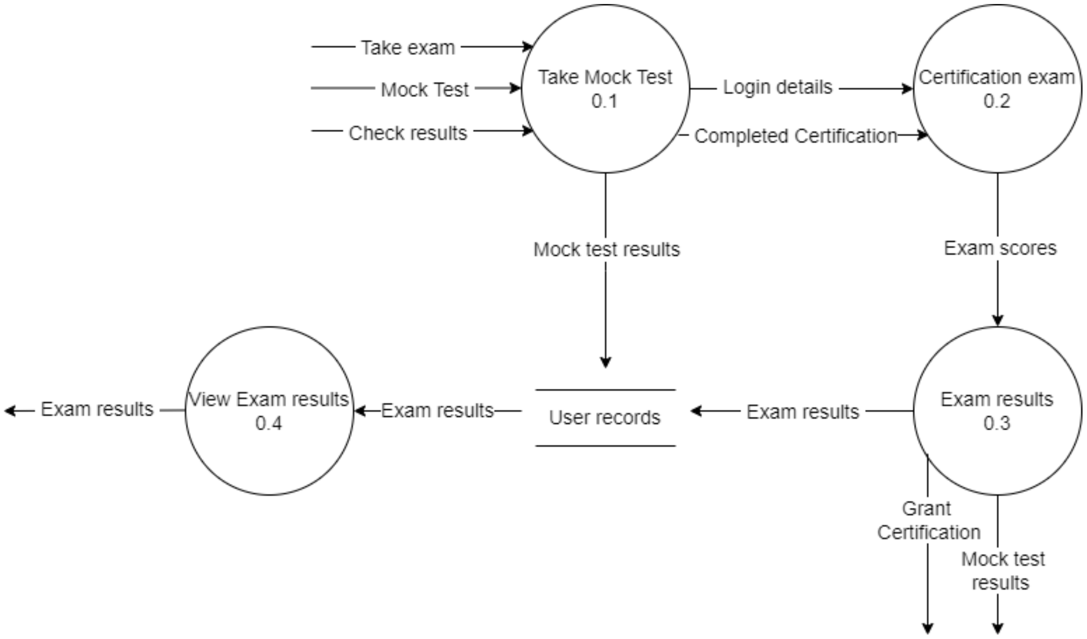

Figure 2522: Level 1 DFD: Take Certification Exam.

In Level 1 DFD, Take Mock Test, Certification Exam, Exam Results, and View Exam Results are the four processes shown in the above diagram. Mock Test and Exam results are stored in a database called user records, which is depicted in the diagram. User information is retrieved into the mock test system and subsequently goes through the full certification procedure. Exam results are accessible to mock test users via the user records database, procedure and the user details that are fetched in.

5.2.2.2. Level 2 Diagram

Level 2 DFD provides a more detailed look in the process while a Level 1 DFD represents a particular process or sub process. A level 2 DFD focuses on one of the primary processes from level 1 and shows the detailed steps or sub processes involved, along with the specific data flows between them.

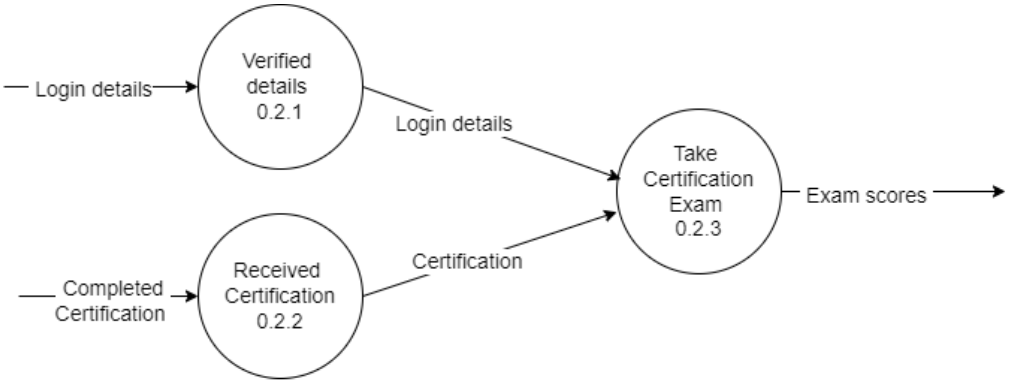

Figure 2623: Level 2 DFD: Take Certification Exam.

The user's details are then passed to process the user's creation through additional deconstruct. Following certification verifies and receipt exam details, the information is given to the users as they take the certification exam and are successful based on their exam scores. The certification exam can now be displayed after additional processing.

5.5.3. Design Specification

5.5.3.1. Structure Chart

A structure chart shows the hierarchical structure of system components and how they interact. The Structure chart provides an outline for system design and development by dissecting the entire system into its most basic modular components and illustrating their linkages and hierarchy. It also analyzes the functions and sub- functions of each module in detail.

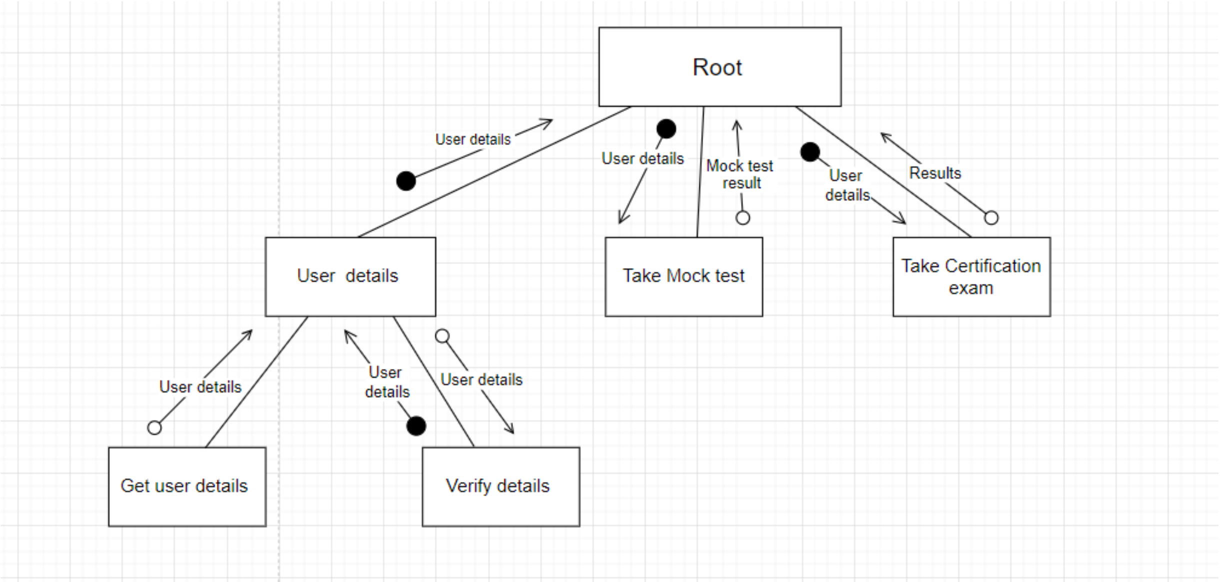

Figure 2724: Structure chart of Take Certification Exam.

This structure chart outlines the key components and their interactions when a user decides to take a certification exam. Verification of user details is incorporated into the structure chart by gathering and verifying data. It is possible for users to participate in a mock test that includes the preparation materials. Finally, the "View Exam Results" module allows users to review their performance in both mock and certification exams. This structured approach ensures a systematic and user-friendly experience throughout the certification process.

5.5.3.2. Module Specification

Module Name: Take a Certification Exam

Purpose: The purpose of this module is to show data flows through the system during the registration, examination, and evaluation phases.

Pseudocode:

```
DO
   INPUT User details
   DO
        IF User details is valid
        THEN
              take Mock test and Certification exam
                  DISPLAY Exam Results
        ELSE
              DISPLAY "INVALID INPUT"
   END DO
END DO
```

Input Parameters: User details

Output Parameters: Exam results or (Exam Cancelled message)

Global Variables: None

Local Variables: User details, Exam results

Called By: Root

5. Summary

Software Engineering is a process of designing, developing, and implementing software solutions to meet user demands. It is necessary for handling the constantly changing nature of software and enable scalability. It involves using practical and theoretical knowledge to create software by detailed instructions.

This is an overall system surface layer design that has been customized and modified for specific needs. Working on each group member's design, internal, and environmental specifications was one of the other individual's jobs. We learned about the practical implementation of the comprehensive understanding of botany, gardening, and the larger facets of community involvement in plant-related efforts through this project. Even though all the Botanical Training ideas were useful and effective, it was usual for us, as new students, to run into obstacles and problems. There were times when we were clueless as to what was wrong. We were able to solve these issues with the help of our lecture and tutorial teacher and internet resources.

We learned a lot about the intricate process of designing and implementing a multifunctional platform. This involves understanding user needs and incorporating diverse features such as user registration, course enrollment, plant purchasing, payment processing, expert recommendations, forum engagement, certification exams, and notification systems. The project was a success because a small group of members worked well together, effectively managed the project, and met the deadline.

6. References

ATLASSIAN. (2023, 12 20). *ATLASSIAN*. Retrieved 12 27, 2023, from ATLASSIAN: https://www.atlassian.com/work-management/project-management/project-planning/project-charter

GeeksforGeeks. (2023, 12 21). *GeeksforGeeks*. Retrieved 12 28, 2023, from GeeksforGeeks: https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/

geeksforgeeks. (2023). *Levels in Data Flow Diagrams (DFD)*. Retrieved 01 03, 2024, from https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/

geeksforgeeks. (2024, January 02). *SRS Format*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/software-requirement-specification-srs-format/

geeksforgeeks. (2024, January 02). *SRS Format*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/software-requirement-specification-srs-format/

geeksforgeeks. (2024). *Structure Charts*. Retrieved 1 3, 2024, from https://geeksforgeeks.org/software-engineering-structure-charts/

geeksforgeeks. (2024, January 03). *Structure Charts*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/software-engineering-structure-charts/

geeksgorgeeks. (2024, January 02). *Non-functional Requirements in SRS*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/non-functional-requirements-in-software-engineering/

Library, U. M. (2022, 12 24). *UC Merced Library*. Retrieved 01 01, 2024, from UC Merced Library: https://library.ucmerced.edu/data-dictionaries

Lucidchart. (2023). *Data flow diagram examples, symbols, types, and tips*. Retrieved 01 03, 2024, from https://www.lucidchart.com/blog/data-flow-diagram-tutorial#:~:text=Level%201%20DFDs%20are%20still,stores%20to%20link%20them%20together.

lucidchart. (2023, 12 21). *lucidchart*. Retrieved 12 28, 2023, from lucifchart: https://www.lucidchart.com/blog/data-flow-diagram-tutorial

Mark von Rosing, A.-W. S. (2015). Phase 2: Process Concept Evolution. In *The Complete Business Process Handbook Body of Knowledge from Process Modeling to BPM, Volume I* (pp. 11-35). Morgan Kaufmann.

Nolle, T. (2023, 12 23). *TechTarget*. Retrieved 01 02, 2024, from TechTarget: https://www.techtarget.com/searchdatamanagement/definition/data-flow-diagram-DFD

point, t. (2023, 12 22). *tutorials point*. Retrieved 12 29, 2023, from tutorials point: https://www.tutorialspoint.com/software_engineering/software_analysis_design_tools.htm

Rouse, M. (2012). *Techopedia*. Retrieved 01 03, 2024, from https://www.techopedia.com/definition/28636/process-specification

Secoda. (2023). *Entity Relationship Diagram* . Retrieved 1 1, 2024, from https://www.secoda.co/glossary/entity-relationship-diagram

Security, D. (2023, 12 20). *Dig Security*. Retrieved 01 02, 2024, from Dig Security: https://www.dig.security/glossary/data-flow-diagram