



CS5002NI Software  **slington college** Engineering

T-14 Training Academy (इस्लिटन कलेज)

20% Group Coursework

AY 2021-2022

Group Name: T-14 Training Program Software			
SN	Student Name (Section name)	College ID	University ID
1	Aasutosh Rajopadhyaya (L2C13)	NP01CP4S210319	20048802
2	Avigya Aryal (L2C13)	NP01CP4S210167	20048838
3	Mubson Karki (L2C13)	NP01CP4S210231	20048923
4	Shreesu Thapa (L2C13)	NP01CP4S210226	20049043
5	Shishir Subedi (L2C13)	NP01CP4S210320	20049039

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

Table Of Context:

1. Introduction.....	7
1.1. Scenario	8
2. Project Charter	10
3. Software Requirement Specification (SRS)	12
3.1. Functional Requirements	12
3.2. Non – Functional Requirements	13
4. Detailed Specification of Group Task.....	15
4.1. Environmental Model Specification.....	15
4.1.1. Context Level Diagram.....	15
4.1.2. Data Flow Diagram (DFD).....	17
4.2. Internal Model Specification.....	24
4.2.1. Entity Relationship Diagram (ERD)	24
4.2.2 Data Dictionary.....	1
4.3. Process specification	6
4.4. Design Specification	9
4.4.1. Structure Chart.....	9
4.5. Assignment Diary.....	1
5. Individual Task.....	4
5.1. Register Membership.....	4
Introduction to module.....	4
Environmental Model Specification	4
Internal Model Specification	5
Design Specification.....	9
Module Specification (Mspecs)	11
5.2. Enrol Staff	13
Introduction to module.....	13
Environmental Model Specification	13
Internal Model Specification	14
Design Specification.....	16

5.3. Purchase Football Kit.....	18
Introduction	18
Design Specification.....	20
Module specification.....	21
5.4. Report Preparation	23
Introduction	23
Environmental Model Specification	23
Internal Model Specification	24
Design Specification.....	26
Module specification (Mspecs)	27
5.5. Take a Mock Exam	1
Environmental Model Specification	1
Internal Model Specification	1
Design Specification.....	3
Module Specification	4
6. Summary	6
7. References	7

Table of Figure

Figure 1 Logo of MS Word	8
Figure 2 Logo of Draw.io	8
Figure 3 Context Level Diagram (Training Academy).....	16
Figure 4 Data Flow Diagram - Level 1	19
Figure 5 Data Flow Diagram Level 2 Request Registration.....	21
Figure 6 Data Flow Diagram Level 2 Upload Video	22
Figure 7 Data Flow Diagram Level 2 Pay Bill	23
Figure 8: Entity	24
Figure 9: Attribute	25
Figure 10: Relationship	25
Figure 11: Entity Relationship Diagram for T-14 Training Academy.....	1
Figure 12 Structure Chart (Training Academy).....	1
Figure 13 Group Coursework Discussion Meeting	3
Figure 14 Context Level Diagram (Register Membership)	5
Figure 15 Data Flow Diagram Level1 (Register Membership).....	6
Figure 16 Data Flow Diagram Level2 (Register Membership) for 0.1 - request membership	7
Figure 17 Data Flow Diagram Level2 (Register Membership) for 0.2 - validation request	8
Figure 18: Structure chart of registered membership	10
Figure 19 Context Level Diagram enroll staff	14
Figure 20 Data flow diagram level1 enroll staff	14
Figure 21 Data flow diagram level2 enroll staff	15
Figure 22 Structure chart enroll staff	17
Figure 23: Context Level Diagram of Purchase Football Kits	18
Figure 24: Level 1 DFD of Purchase Football Kits	19
Figure 25: Level 2 DFD of Purchase Football Kits	20
Figure 26: Structure Chart of Purchase Football Kits	21
Figure 27 Context Level Diagram: Report Generation	24
Figure 28 Data Flow Diagram Level1: Report Preparation.....	24

Figure 29 Data Flow Diagram Level2: Report Preparation	25
Figure 30 Structure Chart: Report Preparation.....	26
Figure 31: Context Level Diagram for Taking a Mock Test	1
Figure 32: Level 1 DFD for Taking a mock exam	2
Figure 33: Level 2 DFD for Taking a Mock Test.....	3
Figure 34: Structure Chart for Taking a Mock Exam	4

Table of Table:

Table 1 Pspecs table: Registration and Deregistration of Staff	6
Table 2 Pspecs table: Reservation of Schedule	7
Table 3 Pspecs table: Take Mock Exam	7
Table 4 Pspecs table: Generating Report	8
Table 5 Pspecs table: Payment.....	8
Table 6 Task Division	3

1. Introduction

The coursework provided is designed to help students understand software engineering concepts and the implementation of those concepts practically. A system has to be developed to help T-14 Training Academy manage data and organizational activities.

T-14 Training Academy is a fresh start for a group of retired players that specialize in football. They are looking to register players, enroll staff, design exam papers, purchase football kits, prepare reports, and take mock exams. While designing this system, it should be noted that the academy's data are to be handled in an organized way. Also, the system should be error-free and user-friendly.

So, for the development of this system, the overall development will be divided into different phases. First and foremost, a project charter will be prepared for the detailed analysis of the project. A system requirement specification document will be prepared after the project charter, which will describe what the software will do and how the needs are to be met.

Various models are drawn for showing the data flow in the system. The functions involved in the system and how the data are processed in those functions are properly presented through Data Flow Diagram (DFD). Also, the data stores are presented and related with the help of an entity-relationship diagram (ERD).

Aims and Objective

- To have a brief idea about 'Structured Software Engineering'.
- To understand the process of modeling context level diagrams, DFD, ERD, and structure charts.
- To collect various requirements of the training academy.
- To work successfully in a small group to a given time scale.

Methodology

For the completion of this coursework, tools like notepad, draw.io were used. Lecture slides, teachers, and the internet's help were taken for gathering the information. Reports, web documents, journals were used as a secondary sources while doing this coursework.

Tools Used

- **MS Word:** After the completion of the project, its description was shown in the form of a report. MS Word, which is a word processor designed by Microsoft was used to write the report.



Figure 1 Logo of MS Word

- **Draw.io:** Draw.io is proprietary software that is designed to make diagrams and charts (Computer Hope, 2020). To represent classes, different methods, and a whole bunch of variables within a diagram, this software was very useful.



Figure 2 Logo of Draw.io

1.1. Scenario

T14 is a football training academy established as an initiative of retired members of the national football team. It offers programs for a variety of age groups and offers two levels of training, namely, basic and intermediate, to its participants. Although there are no prerequisites for basic training, there are some requirements for intermediate training that must be met before the instruction may begin. Additionally, T14 Training Academy sells accessories at a reasonable rate. Specifically, the client needs software developed that will allow users to register, enroll, design exam and practice test papers, acquire football uniforms, prepare and submit reports, make notices and announcements, and take a mock exam, among other things. The following are some of the fundamental needs provided by the client:

- 1. User Registration:** The system will be able to register users in T14 Academy online, which will be the first step in the process.
- 2. Enroll Staff:** New staff can be enrolled in the training academy.
- 3. Design of exam test papers and practice test papers:** Both exam test papers and practice test papers should be able to be designed in the system.
- 4. Purchase football kits:** Football-related accessories should be able to be purchased at a discounted rate through the system. Jerseys, boots, socks, footballs, leg protectors, and other related items are available for purchase.
- 5. Take mock exams:** Both the practice exams and the online exams should be able to be held online.
- 6. Post notices and announcements:** Notices and announcements should be able to be posted by system administrators with the help of the system.
- 7. Report preparation:** A report on a player's performance and health should be generated online.
- 8. Make and save a schedule:** The staff has the right to provide their training time in the system. The user also can reserve the schedule as per their will.

2. Project Charter

Problem Statement

- The registration of new players has decreased at the rate of 20%.
- The rise of Covid-19 has decreased the economic condition of Training Academy at the rate of 50%.
- The pandemic has a direct impact on the transaction of training kits.
- Employees have difficulties arriving at work which causes hindrance and delays in work.

Business Case

- The training academy has set the target to increase the economic condition by 50% within one year.
- The training academy has kept the vision to enroll double the players in the upcoming years.

Goal Statement

- To increase the transaction of players kit 5 times in number.
- To increase the advertisement of the training academy.
- To establish a strong team of players for external exposure.
- Training academy would double the internal games.

Timeline

The training academy has set the date 10th January 2022 as the deadline for the new registration of the players. The training would start from 15th January 2022. The further training process would start according to the player's performance.

Scope

The scope of the Project Management System is divided into Project perspective and Product perspective:

Project Perspective:

- To collect the required information.
- To design the project.
- To develop a plan.
- To perform the testing.

Product Perspective:

- The application deals with online and offline payment.
- The application keeps the record of the users.

Team Member

The Project Management System has the following team members and their assigned work:

Project Manager: Avigya Aryal

The Project Manager is responsible for the overall charge of the planning and execution of a project.

Developer: Mubson Karki

The developer is assigned to develop an application based on the customer's requirement.

Designer: Shreesu Thapa

The designer is responsible for implementing a plan and work of the project.

Tester: Shishir Subedi

The tester is responsible for testing the program developed by the developer and overall project functionality.

Debugger: Aasutosh Rajopadhyaya

The debugger identifies the errors in the code and makes the program error-free.

3. Software Requirement Specification (SRS)

A software requirements specification (SRS) is a document that details how and what the application will perform. It specifies the features and functionality that the product must have to meet the needs of all stakeholders (business and users). The SRS outlines in detail what the software will do and how it will be expected to perform, as well as how an application will interact with system hardware, other programs, and human users in a broad variety of real-world settings. An SRS provides sufficient information to enable developers to complete the described program.

3.1. Functional Requirements

1. Register Membership: The students should fulfill all the specific information asked by the system to be online members of the academy.

2. Test Papers: The system should allow trainers to design and generate exams and practice test papers for the players.

3. Enroll Staff members:

3.1 Vacancy Forms: To recruit new staff members, the system should provide a job advertisement on several social media platforms.

3.2 Staff Appointment: The system should verify all of the applied staff members' caliber points and qualifications to finalize their appointment.

4. Purchase football kits: All the academy members can purchase kits and equipment provided by the system at a reasonable price.

5. Report preparation: The system should enable the trainer to generate the final report of the players regarding their performance in the tests.

6. Exam notices and announcements: The admin should notify players regarding their examinations, results, and any further notices via their assigned email addresses and the software's news portal.

7. Take a Mock Exam: The members are required to pass a mock test to join the intermediate level.

7.1. Tutorial Videos: The system provides a set of different tutorials guiding the players on different playing aspects along with the knowledge required for performing better in the game.

7.2. Mock Exams: Physical and written exams are taken once the player completes the particular section of the videos to examine their knowledge and proficiency.

3.2. Non – Functional Requirements

1. Design and Implementation constraints

1.1. Limitations of the developer: The developer is required to code on almost every accessible device and platform like ios, android, mac, Linux, and windows.

1.2. Limitations in hardware: A minimum of 2 GB of RAM is required to run the application fluently.

1.3. Database: The database management system to be used is Oracle as requested from customers technique represented.

2. External Interfaces

2.1. Hardware: It needs to support every hardware like laptops, pcs, and mobile phones.

2.2. Software

2.2.1. User interface: The code should be implemented in such a way that the presentation is not affected by the change in screen size.

2.2.2. Payment: For the online payment procedure different software like eSewa or khalti is optimized.

2.2.3. Registrations: Google or Facebook can be utilized during registration processes.

2.3. Network

2.3.1. Interaction with protocols and servers: The system should successfully interact with the network protocol, communication protocol and should be supported by the application server.

3. Other Non-Functional Requirements

3.1 Security: It needs to track all the network traffic incoming and outgoing through firewalls and provide a secured platform for all its users.

3.2. Flexibility: It needs to support thousands of users at a time consecutively.

3.3. Performance: It needs to be smooth and should operate without any lags or bugs.

3.4. Storage: It needs to store all the details of the users, staff, and players in a reliable oracle database or cloud computing can also be utilized for storing the records in a virtual environment.

3.5. Accessibility: It needs to be readily available and accessible to every user with minimum delay time.

3.6. Maintenance: Updates should be prioritized as per the user's review and improvements should be brought to prevent any bugs or faulty activities.

Goals of Implementation

1. The code should be written in a properly structured manner in such a way that it can be re-used and maintained in the future.
2. Code refactoring should be emphasized.
3. Latest hardware and updated drivers are mandatory before any process is carried out.
4. Object-oriented programming language should be strictly used for writing the codes.

4. Detailed Specification of Group Task

4.1. Environmental Model Specification

4.1.1. Context Level Diagram

Also known for the top-level or level 0 data flow diagram, a context level diagram is a basic overview of the whole system. This diagram presents the scope and boundaries of a system. It contains a single process node that generalizes how every function of the entire system is related to external entities. Thus, at-a-glance view, the context level diagram shows the entire system as a high-level process.

The context level diagram of this football training academy, T-14 Training Academy is presented below. This diagram shows and reflects how the external entities are interacting with the software. Various functions that are carried out by the system are shown in this context level diagram. Also, this diagram helps in understanding the diagrammatical ideas on how data would flow between the system and external entities.

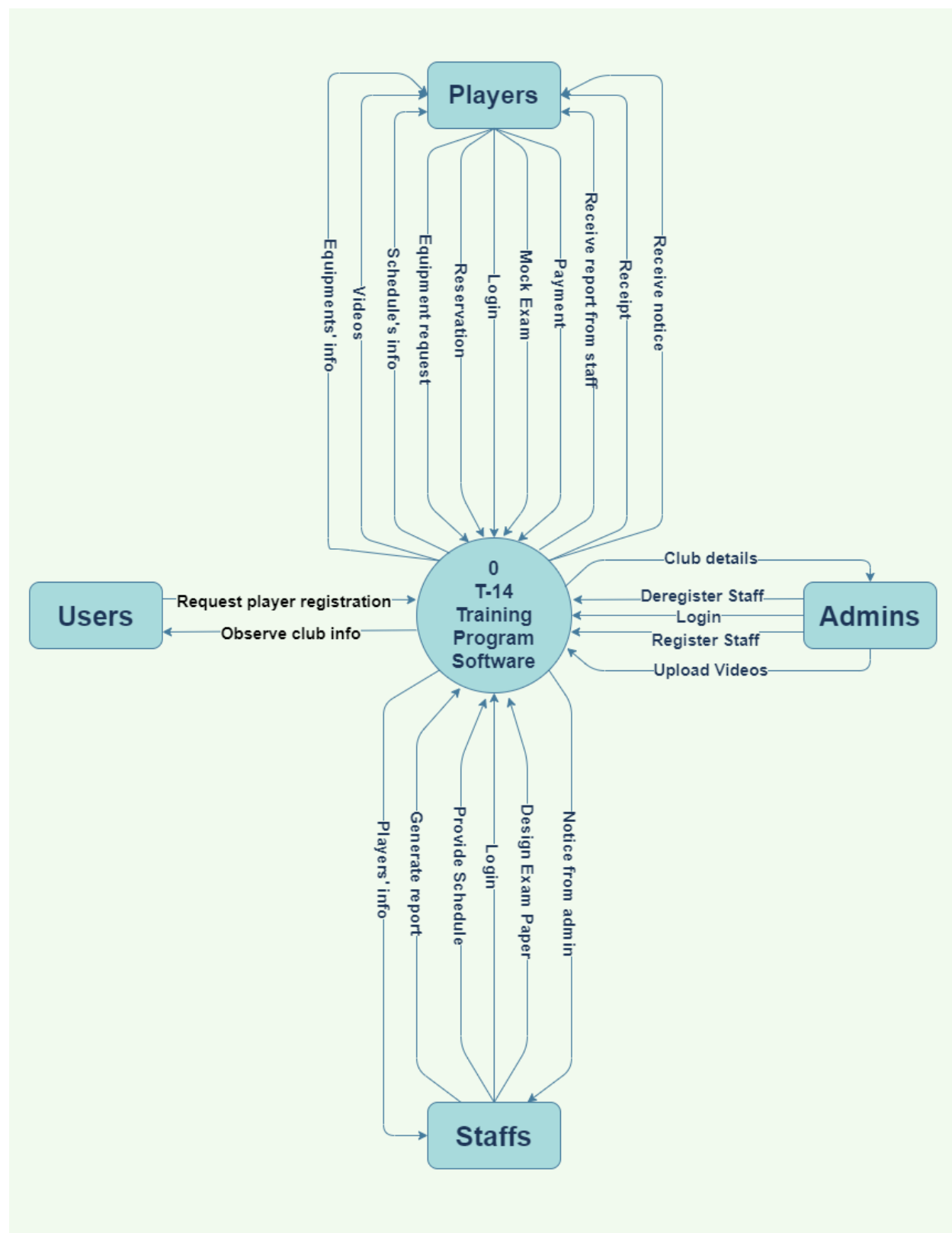


Figure 3 Context Level Diagram (Training Academy)

From the given level 0 data flow diagram, a brief idea of how data is being processed and flowed in the T-14 Training Program Software is developed. A total of four entities – Admin, Staff, Player, and User are interacting with the software. For every external entity, data flow is bi-directional. Thus, it can be known that the system is user interactive.

Here, the user entity can request player membership to the system. Without membership, the user is limited to just observing club information. Also, staff or trainer has various functions like providing a schedule, generating a report, designing exam paper, and so on. Likewise, players can perform all of the functions described in the above figure. Finally, the admin is the other entity that is like the head of the system which can register staff and can achieve access to all functions of the system. These major functionalities along with other functions of this system are also included in this context level diagram.

4.1.2. Data Flow Diagram (DFD)

Now as the basic overview of the entire system has been designed in the context level diagram, the flow of data within that system is to be designed. This is done by Data Flow Diagram (DFD). The DFD is simply an intuitive way of showing how data is processed by a system. In a data flow diagram, there are three notations; circles represent functional processing, the partial rectangle represents data stores and labeled arrows show the data movements between the functions.

DFD is an important part of software engineering. It helps in tracking and documenting how data of a given function moves in a system. The best thing about the DFD is that it is simple and intuitive. System users who have less idea about the system can understand how their data are being processed. As clear and well-designed DFD can represent the entire system requirements graphically, DFD is one of the best models for analysis.

The data flow diagram of the T-14 training academy is divided into two levels which are shown below:

4.1.2.1. Data Flow Diagram – Level 1

The level 1 DFD presents the system's functions in a more detailed manner. This diagram helps in understanding major functions - how data are being used for the process and what are their outputs. For this T-14 training program software, a total of 11 data stores, 20 functions, and numerous data flows are used. The way system is built and functions can be seen in the level 1 DFD as below:



Figure 4 Data Flow Diagram - Level 1

Here, every functionality is presented in a circle in the above figure. Different data stores are used for storing data. When registering and deregistering staff, a *staff records* datastore is used. This same-store is also used in other various functions like adding schedules. Similarly, *schedule details*, *reservation details*, *notice details*, *equipment details*, *payment details*, and *report details* are used as a data store to store their respective data. Some data stores have two different data stored within them like *video records* store the videos as well as the watch history of the player. Also, *exam records* store both exam papers and the answer papers within it. Thus, every data store has a bi-directional data flow. Data comes as well as goes out.

4.1.1.2. Data Flow Diagram – Level 2

Level 2 is the specific data flow diagram for a particular function. As the functions in level 1 execute other various sub-functions within them, it becomes important to show how the data are being processed within the functions of level 1. So, the level 2 data flow diagram is designed and presented below:

- **Request Register (0.8)**

This is the 0.8th function of the level 1 DFD that needs other various sub-functions to work. This level 2 diagram for the request register contains an external entity – User as shown in the figure below:

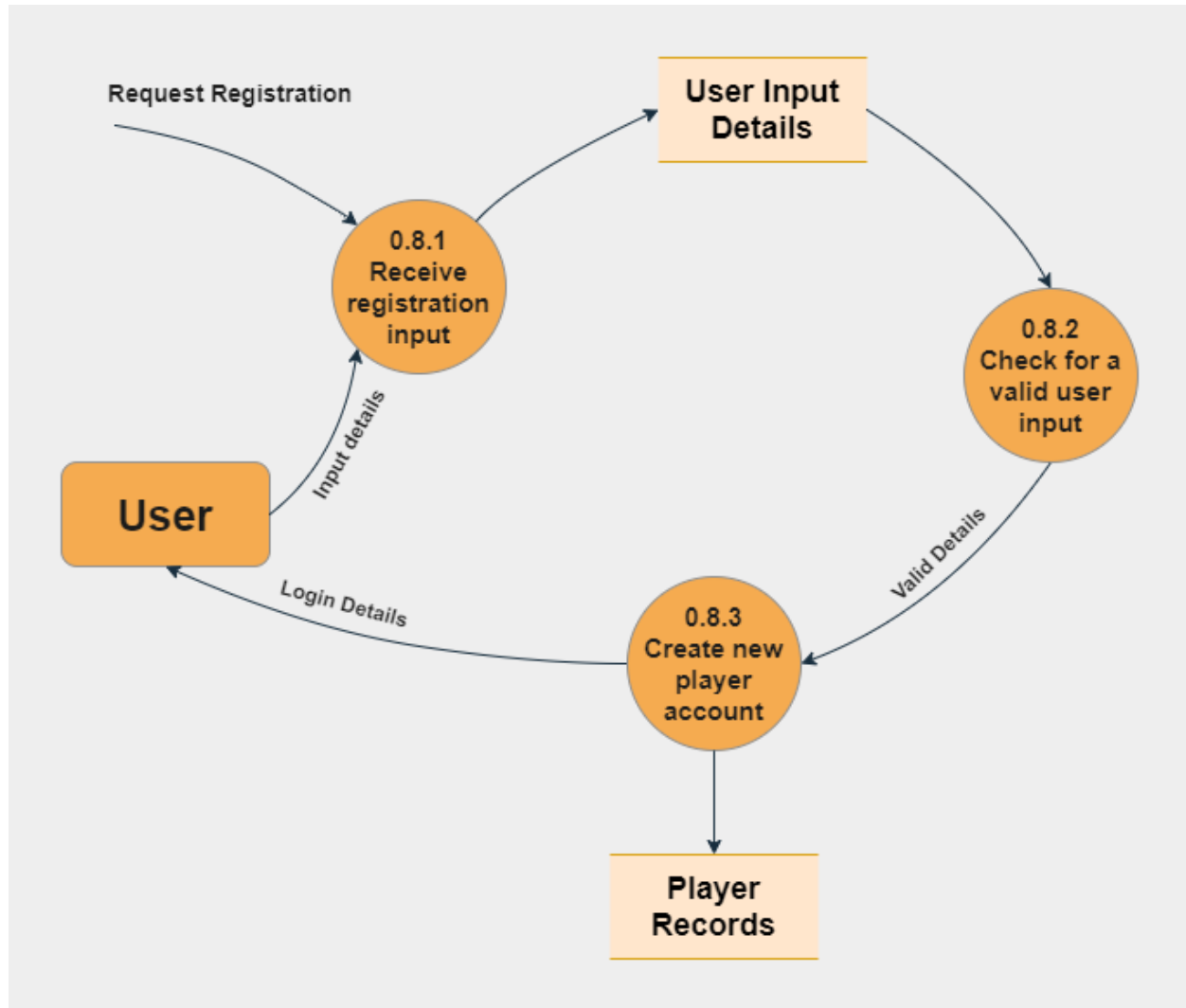


Figure 5 Data Flow Diagram Level 2 Request Registration

From the above diagram for requesting registration, the input details of the user are processed and stored. The system receives the registration input and stores them in the *user input details* datastore. Then, these input details are received by another function (0.8.2) for checking if the user has given valid input details. Finally, the input details are added in the *player records* the login details for a new player account are given to the user.

- **Upload Videos (0.11)**

The below level 2 diagram of the upload video function shows how data are transferred within various sub-functions and a data store – *videos records*.

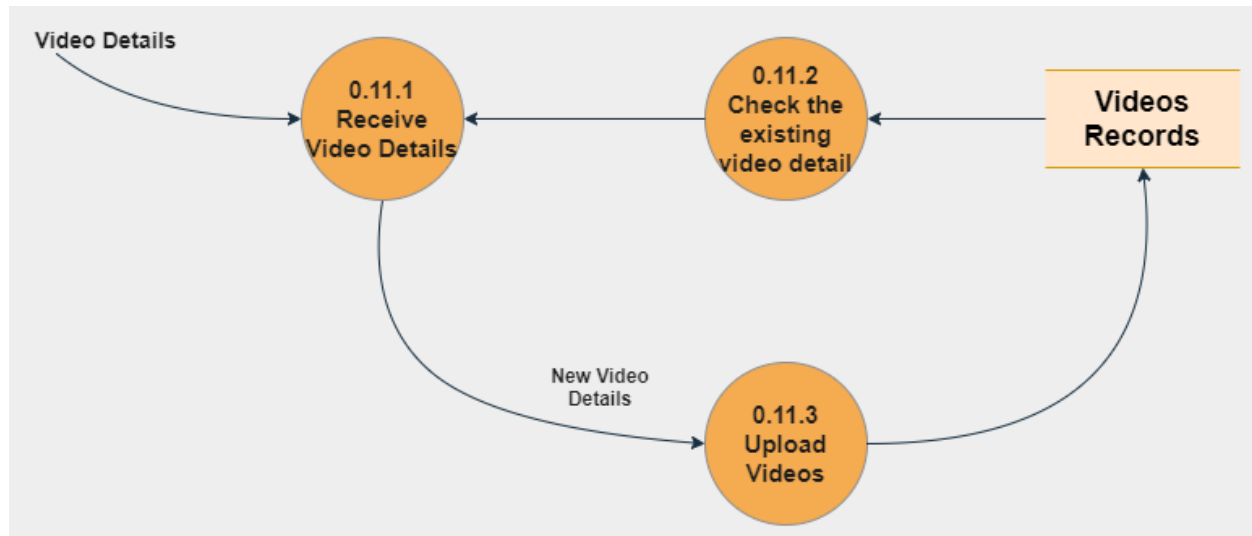


Figure 6 Data Flow Diagram Level 2 Upload Video

From the above diagram, video data is being processed throughout the sub-functions. First, a new video detail is received and is checked if the new video detail is already in the *video records* data store or not. After confirmation, the new video details are then stored in the data store as shown in the figure above. Thus, a total of three sub-functions are being executed for uploading a video in the data store.

- **Pay Bill (0.19)**

Here, the below figure is a level 2 data flow diagram of the pay bill function. This function contains four sub-functions within them where data are being processed at each one of them as shown in the figure below:

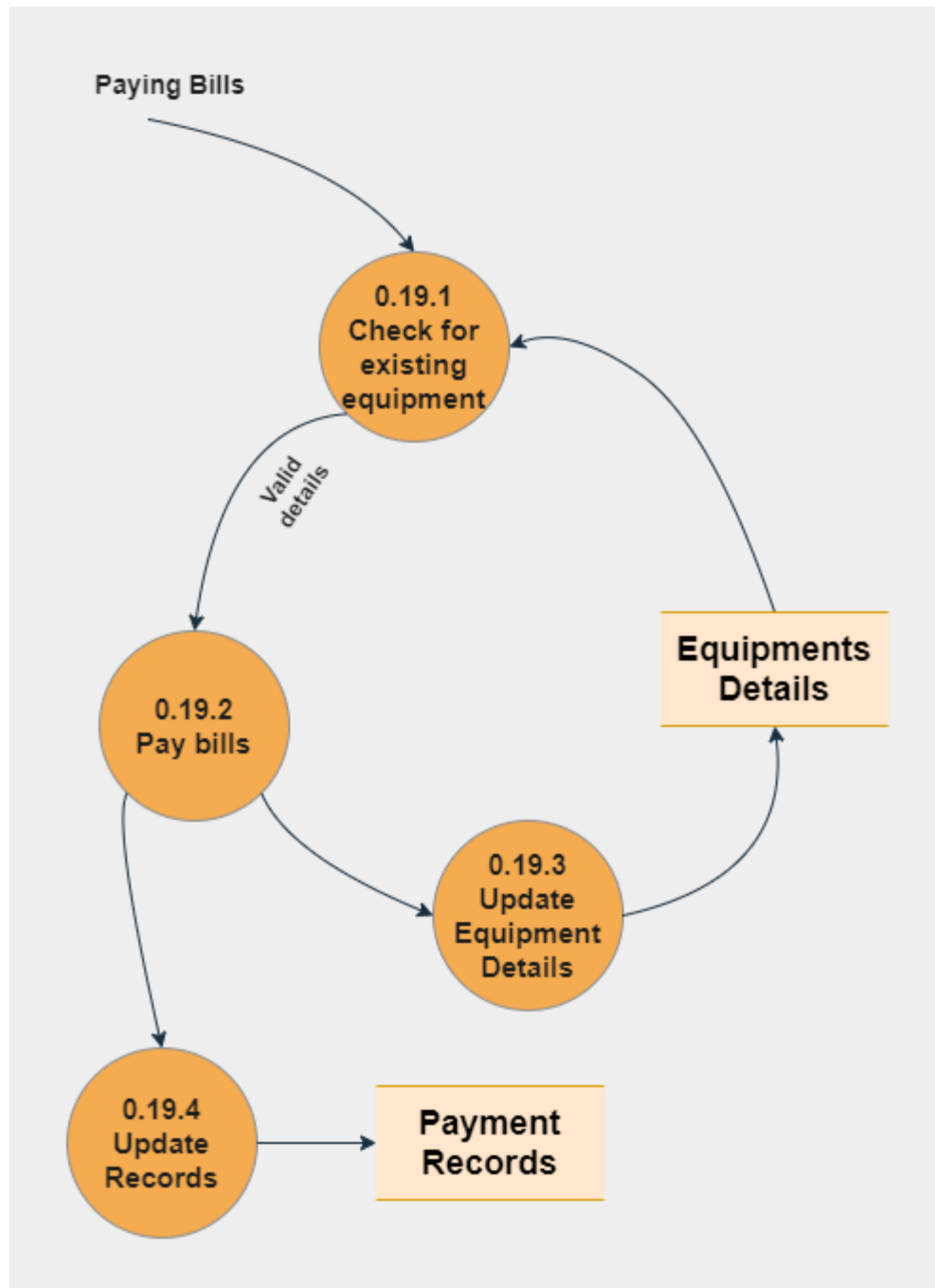


Figure 7 Data Flow Diagram Level 2 Pay Bill

From the above diagram, the payment details are being processed along with the equipment details. During the payment of a bill, first, if the equipment details that are being purchased are valid or not is confirmed. After that, the bill is paid online. So, once the bill has been paid, again, the equipment detail is processed and the purchased

equipment detail is updated in the *equipment's details* data store. Finally, the payment record is also updated/added in the *payment records* data store for any further use.

In this way, various functionalities of the system can be understood through these diagrams. The T-14 Training Program Software contains a total of 20 functions. So, how data are being processed and how these functions are being executed can be easily understood through these context level diagrams and data flow diagrams. Thus, the main purpose for modeling these diagrams i.e., to make potential system user understand their data flow and system functionalities is achieved.

4.2. Internal Model Specification

4.2.1. Entity Relationship Diagram (ERD)

The Entity-Relationship Diagram (ERD) is a visual representation of groups or entities that shows how they interact inside a system. ERD is a direct view of the database that displays how each table is connected to the others, which helps with relational database creation and design. Because database columns are related, the ERD illustrates relationships between entities' components rather than relationships between entities themselves, which is beneficial in databases (Lucidchart, 2019).

An ER-Diagram comprises various components in its design. They are entities, attributes, and relations.

Entity: Entity is the real object or concept that can be defined and have data stored about it. For example, a Customer is a real object that can have its data stored. They are represented by a rectangle.



Figure 8: Entity

Attribute: Attributes are the properties of the entity. Attributes define the entity and get stored in the database. For example, a Customer can have CustomerID and CustomerName as its attribute which defines the entity Customer and can be stored in the database. The attributes are represented by an oval.

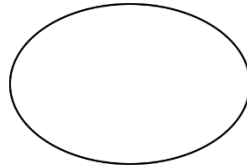


Figure 9: Attribute

Relationship: Relationship is a component of ERD that shows how different entities are linked with each other. It is represented by a diamond.

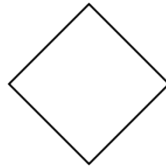


Figure 10: Relationship

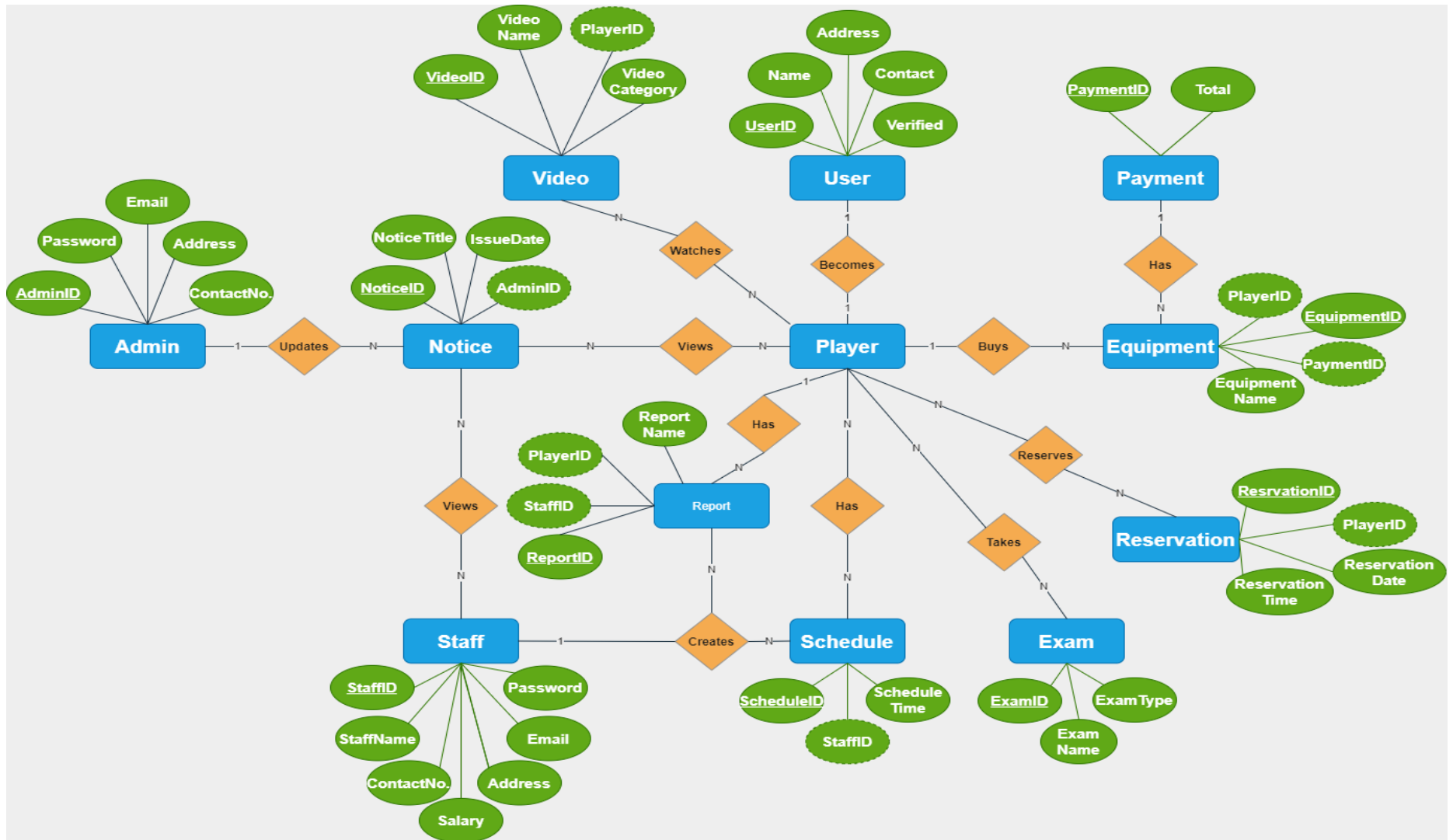


Figure 11: Entity Relationship Diagram for T-14 Training Academy

The ER-Diagram for the T-14 Training Academy's database is shown above. The ERD was built to meet the T-14 Training Academy's online requirements. The entities used in the ERD are Admin, Notice, Staff, Player, User, Report, Schedule, Exam, Equipment, Video, Reservation, and Payment. The Entity-Relationship Diagram (ERD) describes these entities and characteristics that will be kept in the training academy's database. The graphic depicts the relationship between the database's numerous tables.

4.2.2 Data Dictionary

The data dictionary is a document that describes the structure of how data is stored in a database. It includes attributes of the entities along with their data type and constraints. This document contains an explanation of data in a system.

The data dictionary is a crucial component in any system which provides additional information and relationships between entities. It also helps in organizing data and prevents redundancy of data.

The tables represent the data dictionary of entities along with their attributes in the system.

1. User:

The user is someone who uses the online system of the training academy.

User = UserID + UserName + UserContactNo + UserAddress + Verified

UserID = String

UserName = String

UserContactNo. = int

UserAddress = String

Verified = String

2. Payment:

Payment made by players for the purchase of equipment.

PaymentID = String

Total = int

Payment = PaymentID + Total

3. Admin:

Admin is the one who overlooks and manages all the activities in the Training academy.

AdminID = String

AdminName = String

AdminContactNo. = int

AdminEmail = String

Password = String

Admin = AdminID + AdminName + AdminContactNo. + AdminEmail + Password

4. Reservation:

Details of reservation of schedule by player.

ReservationID = String

ReservationDate = Date

ReservationTime = String

Reservation = ReservationID + ReservationDate + ReservationTime

5. Video:

Tutorial videos are provided in the training academy.

VideoID = String

VideoName = String

VideoCategory = String

Video = String + String + String

6. Equipment:

Equipment which can be bought in training facility

EquipmentID = String

PaymentID = String

PlayerID = String

EquipmentID = String

Equipment = EquipmentID + PaymentID + PlayerID + EquipmentID

7. Reservation:

Details of reservation of schedule by player.

ReservationID = String

ReservationDate = DATE

ReservationTime = String

Reservation = ReservationID + ReservationDate + ReservationTime

8. Exam:

Details of exams conducted in the training academy.

ExamID = String

ExamName = String

ExamType = String

Exam = ExamID + ExamName + ExamType

9. Staff:

Details of staff working in the training academy.

StaffID = String

Password = String

Email = String

StaffName = String

StaffAddress = String

ContactNo. = int

Salary = int

Staff = StaffID + Password + Email + StaffName + StaffAddress + ContactNo. +
Salary

10. Report:

Report of players.

ReportID = String

StaffID = String

PlayerID = String

ReportName = String

Report = ReportID + StaffID + PlayerID + ReportName

11. Note:

Notice issued by admin.

NoticeID = String

NoticeTitle = String

IssueDate = DATE

AdminID = String

Notice = NoticeID + NoticeTitle + IssueDate + AdminID

12. Player:

Players who have been registered in Training Academy.

PlayerID = String

UserID = String

Password = String

Email = String

PlayerName = String

ContactNo. = Int

Address = String

Player = PlayerID + UserID + Password + Email + PlayerName + ContactNo. + Address

In this way, a data dictionary contains information that provides details about the contents of a database. This can be quite handy in designing test cases and designing software by technical experts.

4.3. Process specification

Introduction:

A process specification is a method used in software engineering for documenting, analyzing, and explaining the decision-making logic and formulas. The process specification methodology reduces uncertainty. The methodology allows an individual or the organization to obtain a precise description of executed tasks. It accomplishes the task and validates the system design which includes the data dictionary and data flow diagrams. The Training Academy follows the following process specification in the application.

1. Registration and Deregistration of Staff:

Name:	Registration and Deregistration of Staff
Description:	To register the details of staff and de-register the staff by admin.
Process Type:	Online
Inflow data:	Staff Details
Outflow data:	Registration details
Process logic:	<ul style="list-style-type: none"> • Collects the input details of the staff. • Register the staff by updating the database with new details. • Matches the detail of staff from the <i>Staff Records</i> Database. • De-register the staff by admin.

Table 1 Pspecs table: Registration and Deregistration of Staff

2. Reservation of Schedule

Name:	Reservation of Schedule
Description:	To reserve a staff's training schedule by player.
Process Type:	Online
Inflow data:	Schedule details, player details
Outflow data:	Reservation details
Process logic:	<ul style="list-style-type: none"> • Get staff details and time details from the <i>Schedule</i> database. • Player registers for training schedule by providing their details. • Reservation for a player is booked in the <i>Reservation Details</i> database.

Table 2 Pspecs table: Reservation of Schedule

3. Take Mock Exam

Name:	Take Mock Exam
Description:	To give an exam for upgrading a player's training to intermediate.
Process Type:	Online
Inflow data:	Exam details, player details
Outflow data:	Paper details
Process logic:	<ul style="list-style-type: none"> • The staff adds question paper in the <i>Exam Records</i> database. • Player views the question paper and gives the exam. • The answer written by the user is again stored in the <i>Exam Records</i> database.

Table 3 Pspecs table: Take Mock Exam

4. Generating Report

Name:	Generating Report
Description:	To generate an overall report of a player.
Process Type:	Online
Inflow data:	Player's detail, Exam details, Reservation details, Video history details
Outflow data:	Report details
Process logic:	<ul style="list-style-type: none"> • Player's result in exam is viewed from <i>Exam Record</i> database. • Player's performance in training is known from the <i>Reservation Details</i> database. • Player's watch history video is achieved from <i>Video Details</i> database. • All this information is generalized and a report is prepared. • The report is then stored in the <i>Report Details</i> database.

Table 4 Pspecs table: Generating Report

5. Payment

Name:	Payment
Description:	To pay a bill for the player's purchase of equipment.
Process Type:	Online
Inflow data:	Player details, Equipment details
Outflow data:	Payment details
Process logic:	<ul style="list-style-type: none"> • Receive price details of equipment from the <i>Equipment Details</i> database. • Pay the bill of purchased equipment online. • Update the quantity of equipment in its respective database. • Record the payment details in the <i>Payment Records</i> database.

Table 5 Pspecs table: Payment

The Process specification methodology used for the training academy application consists of the above-mentioned functions. The functions mentioned above are the topmost function applied in the application. The functions are registration which includes the details of staff for the registration and deregistration. The payment function includes the payment details of the player for the purchase of the football kit. The reservation of the schedule function sets the detailed timetable information of the players. The mock exam is generated by the staff for the players to upgrade their level. The report is generated for each player which include the player details, payment record, video watched history, exam report, and many more.

4.4. Design Specification

4.4.1. Structure Chart

The Structure Chart illustrates the hierarchical arrangement of modules. It breaks down the whole system into its smallest functional modules and goes into further detail about each module's functions and sub-functions. The system is divided into modules represented by rectangle boxes in the System Structure Chart (functionality of the system is known to the users but inner details are unknown). Inputs are given to the rectangle boxes, and corresponding outputs are produced (Agarwal, 2019).

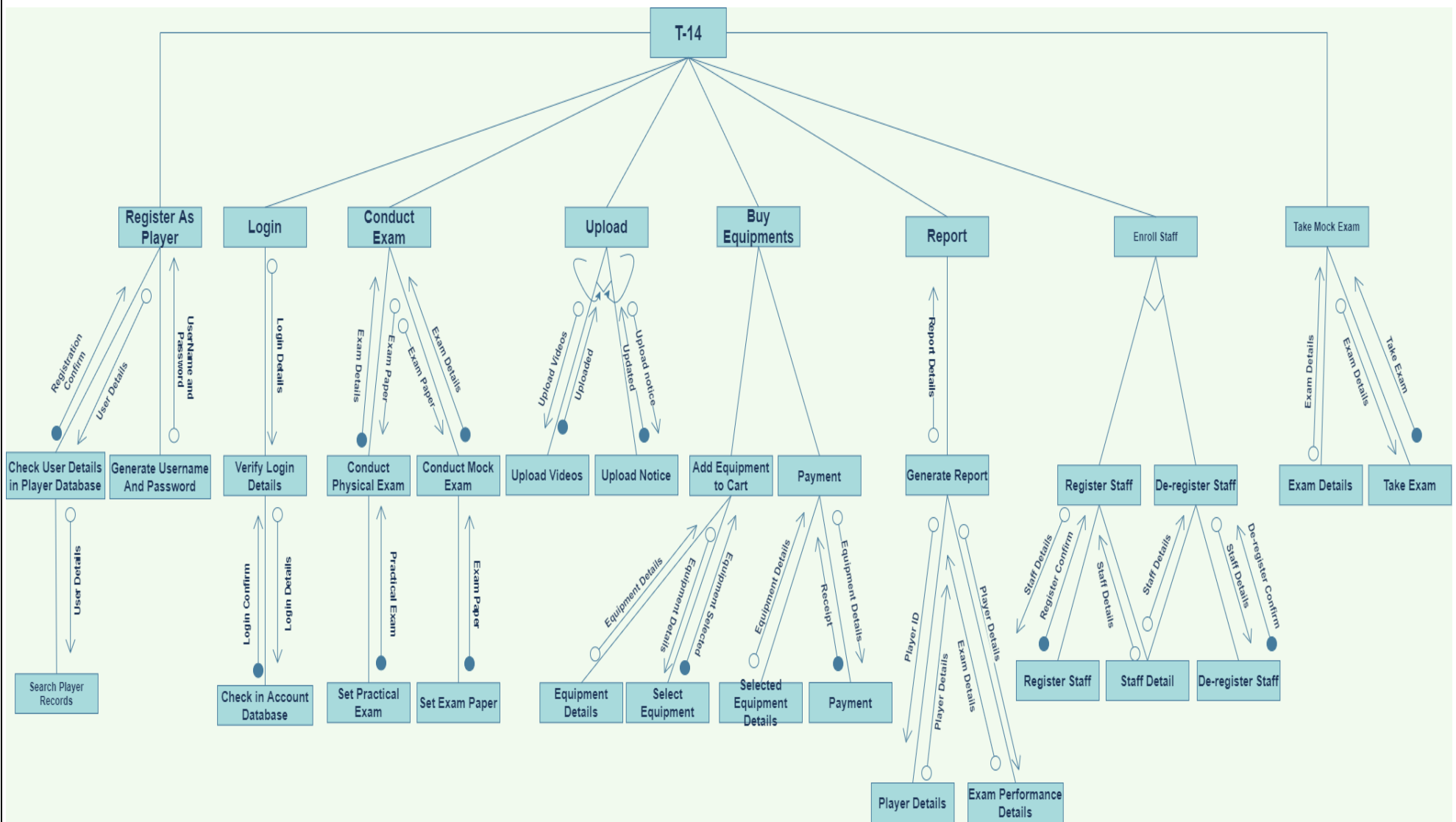


Figure 12 Structure Chart (Training Academy)

In the above structure chart of T-14 Training Academy, T-14 is the main root that performs various functions. The functions of the T-14 are then represented in the box as modules. The main function of the T-14 is considered here which includes registering as a player, login, conducting an exam, upload, buying equipment, reporting, enrolling staff, and taking a mock exam. These modules each have their sub-module or function they operate on to achieve the function. The flow of the chart is indicated by the arrowhead which either provides the details or confirms the entry or request.

4.5. Assignment Diary

Since the given project is broad in scope, a formal work diary was created to keep track of all the tasks that were discussed, assigned, finished, challenges that arose, and so on. With the help of the assignment diary, we were able to acquire an exact sense of the completion percentage and progress of our work, and we were able to make necessary adjustments as a result. It also assisted in keeping track of the items that were eliminated from the project as a result of challenges, as well as the assumptions that were made to keep the project up to the standards required by the client.

Assumptions:

1. A user is someone who gains access to the system after registering as a member. The user is permitted to be a player and has access to player-specific features.
2. Admin is the system's administrator who has the authority to register and de-register staff. Additionally, the administrator verifies the user's membership registration.
3. The staff trains the players at their convenience (schedule). Players must register for the scheduled times to have access to the training facilities.
4. Exam papers are created by staff and are maintained in a separate database. Players can access these papers, which serve as tests. The answer papers are maintained in the same database as the examination papers.

5. The admin uploads the videos into the system, which are then viewed by the players. As a result, these video details, in addition to the player's viewing history, are also maintained in a separate database.

6. The staff prepares a general report on a player based on their exam performance, video viewing history, and training participation.

Group Meeting: While working on this project, the group met regularly to discuss the progress. Five individuals contributed to the completion of various tasks. The meeting, which took place both physically and digitally from time to time, allowed group members to discuss the coursework and its contents with one another. The following are the important meetings scheduled:

1. First Meeting:

Date: 28th December 2021

Time: 8:30 AM – 10:00 AM

Location: Islington College

Discussion: The coursework and its core contents were discussed in detail during the first meeting, and the assignments for the group were split among the group members.

2. Second Meeting:

Date: 1st January 2021

Time: 11:30 AM – 2:30 PM

Location: Online Platform

Discussion: The second meeting was focused on concluding the discussion on how to meet the need for additional coursework. Different chart constructions were split into their constituent parts, and the individual tasks were determined.

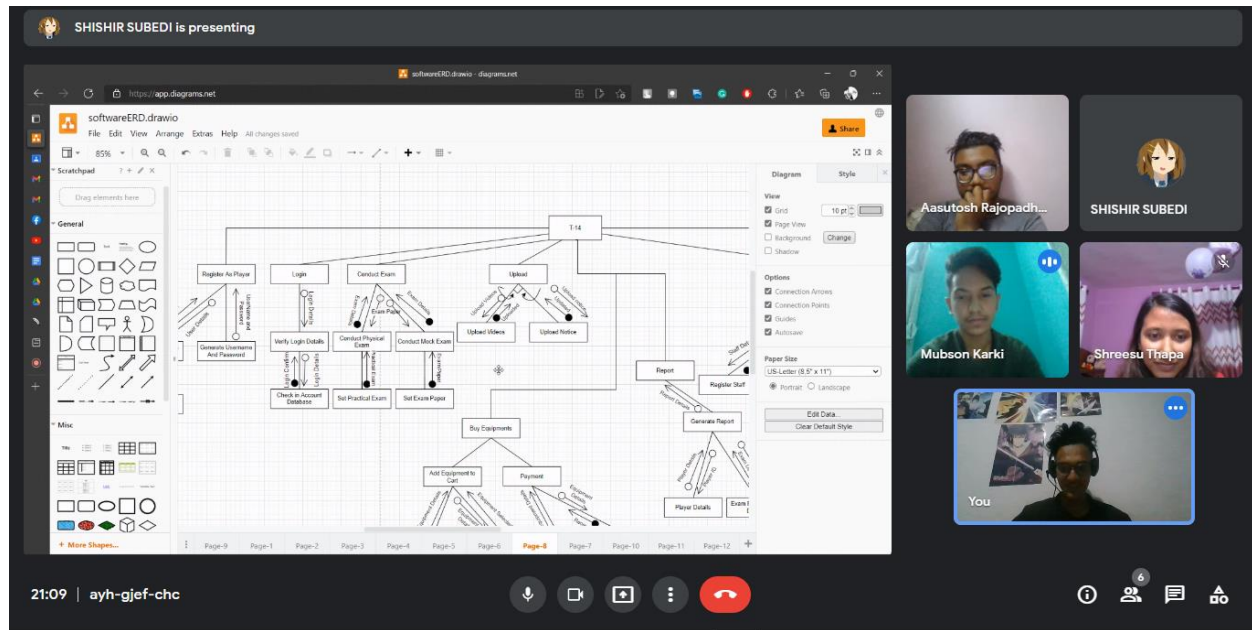


Figure 13 Group Coursework Discussion Meeting

Group task division:

SN	Student Name	Group Task Division
1	Aasutosh Rajopadhyaya	Introduction and Conclusion of the Training Academy.
2	Avigya Aryal	Software Requirements Specification (SRS) of the Training Academy and Assignment Diary.
3	Mubson Karki	Context Level Diagram and Data Flow Diagram (Level1 and Level2) of the training academy.
4	Shreesu Thapa	Project Charter and Process Specification of the Training Academy.
5	Shishir Subedi	ER-Diagram and Structure Chart of the Training Academy.

Table 6 Task Division

5. Individual Task

5.1. Register Membership

Name: Mubson Karki

London Met ID: 20048923

Introduction to module

This module, register membership involves two different entities – User and Admin. Here, users are any individuals who are accessing the software as guests and are not players. But once the users' membership has been achieved, they turn into players. So, for this membership registration, the admin is responsible.

The main function of this module is to register the user as a player. So, to register membership, the user provides their details which get stored in a database called "*User Input Details*". Then, the admin verifies the input details by accessing this database. Once the details of the users are verified, those details get again stored in another database called "*Player Records*". This means that now the user has been upgraded to the player. Finally, log-in details are provided to the user from which they can access the software as a player.

Environmental Model Specification

Context Level Diagram

Here, different flows of information between the system and external entities for registering membership are presented via a context diagram. Context diagram or level 0 data-flow diagram is a diagram drawn to define and clarify the boundaries of the software system (Computer Science Department, University of Cape Town, 2011).

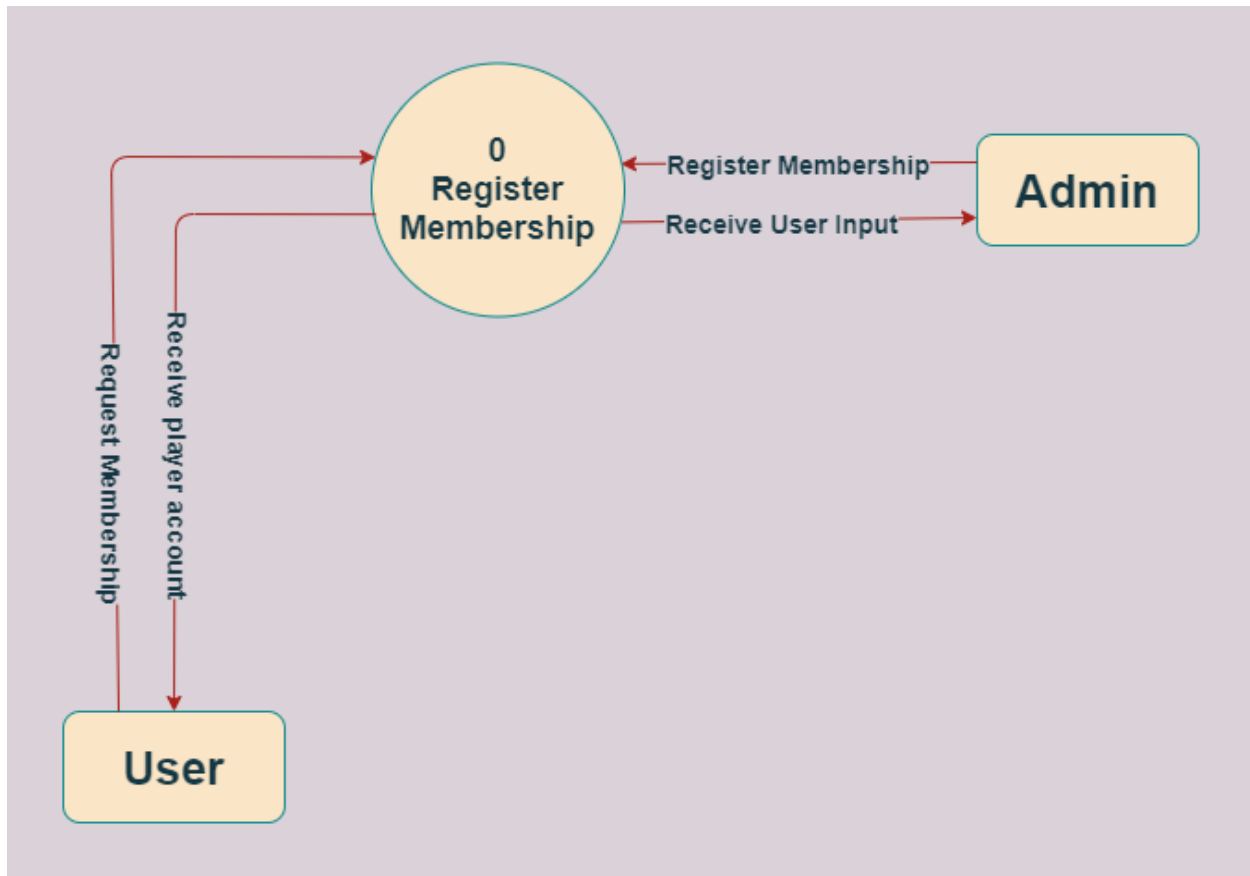


Figure 14 Context Level Diagram (Register Membership)

From the above figure of environmental model specification for registering membership, admin and user are two external entities interacting with the software. The user is the one who provides their details and the admin is the one who on receiving the user input, verifies the details. Finally, once the user inputs are verified, a player account for the user is created and is given to the user.

Internal Model Specification

The internal model specification of registering membership is represented by level 1 and level 2 Data Flow Diagrams (DFD). These DFD are an intuitive way of representing a data flow model that shows how data is processed by a system (Sommerville, 2016).

Here, the diagrams that are presented below show all the data flows between the process, external entities, and data stores that take place in the process.

DFD Level1

The overall functions involved in registering membership are presented in this level 1 DFD. For this function, a total of two data storages are used – User input details and Player records. The user input detail is for storing all the input details given by a user whereas the player record keeps all the records of a player. So, the data flow between external entities and how these storages are used is shown below:

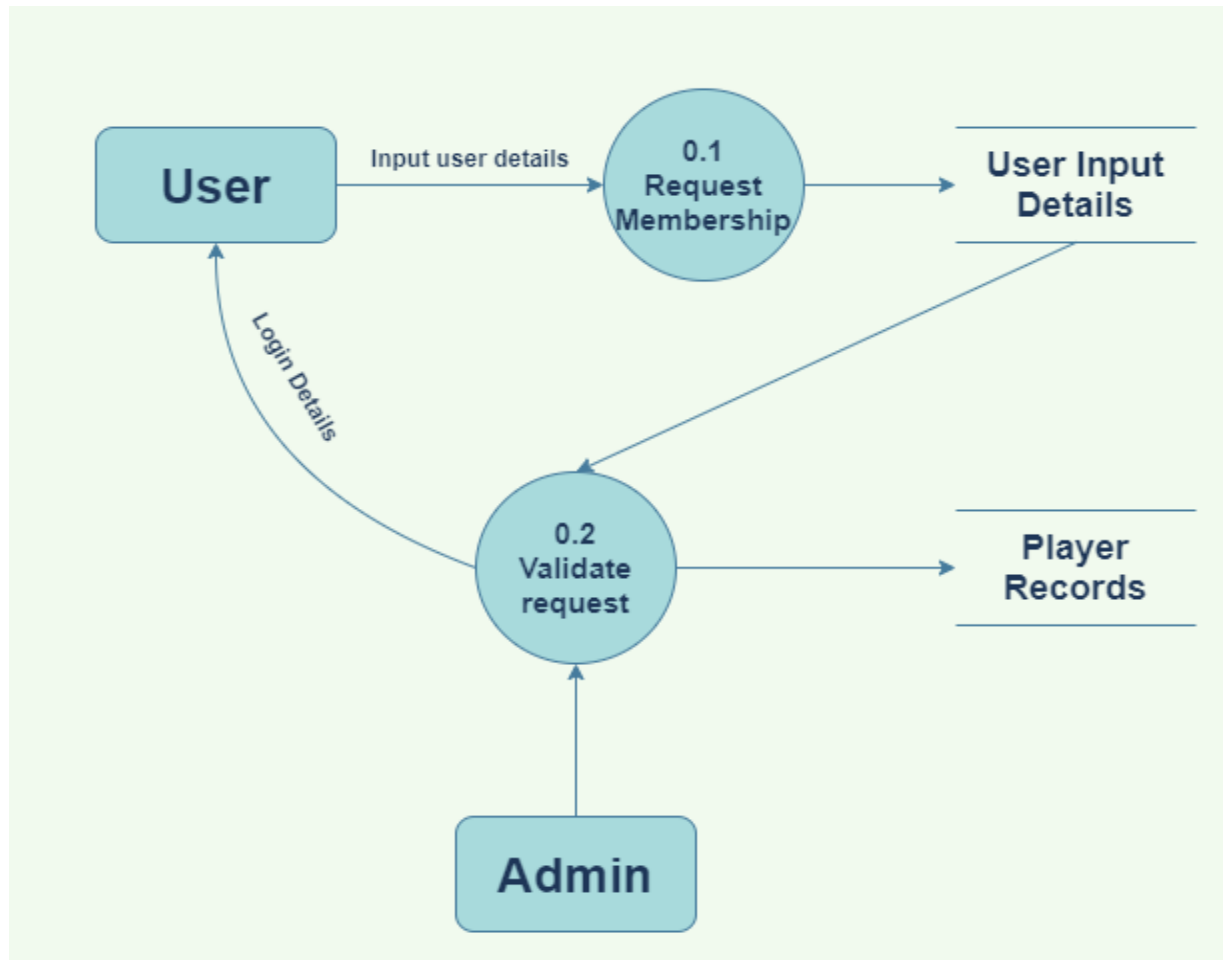


Figure 15 Data Flow Diagram Level1 (Register Membership)

The above figure shows that user details are necessary for requesting membership. So, once the details are provided, it gets stored in the user input details. Now, after the admin validates the request, user details get stored in the player records and login details are given to the user.

DFD Level2**For request membership**

The request membership function of the level 1 DFD has various sub-functions that are executed when the user requests membership. So, another level 2 DFD is required that shows the data flow for these sub-functions. For request membership, only external entity – user is used along with one data storage – user input details. The level 2 DFD for requesting membership is presented below:

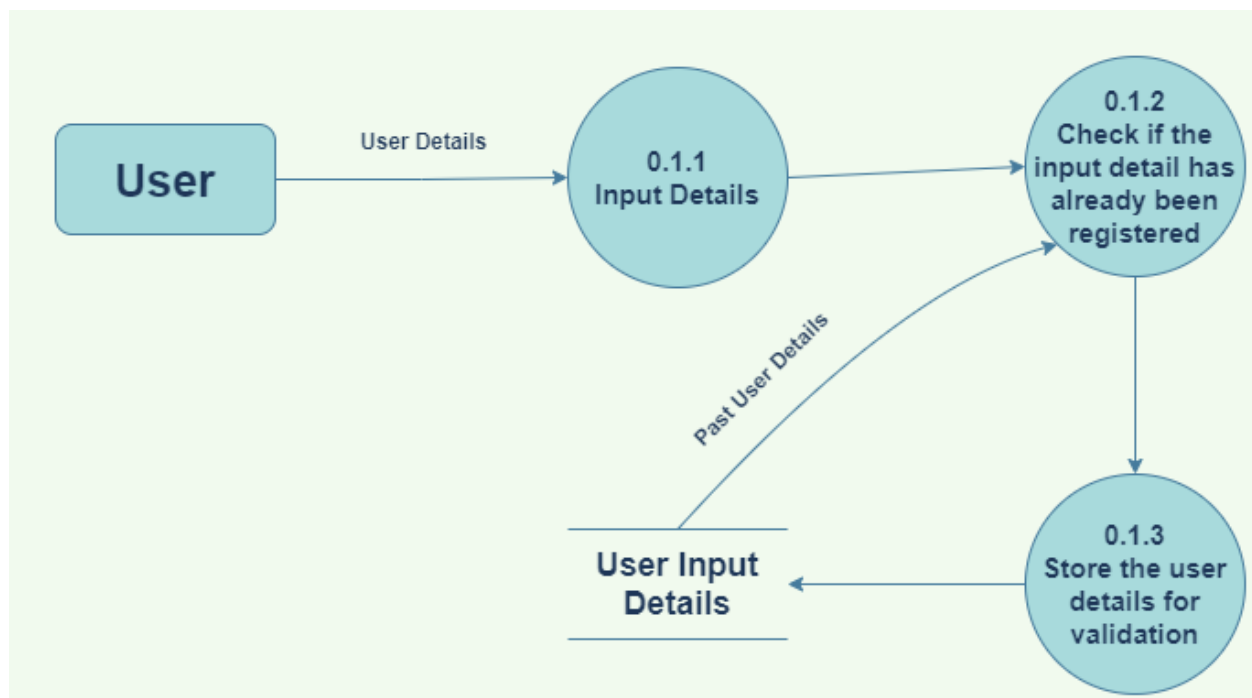


Figure 16 Data Flow Diagram Level2 (Register Membership) for 0.1 - request membership

Here, the user provides their details to the system. So, once the details are provided, the system then checks if that detail provided by the user is already in the data store to avoid data redundancy. Finally, if the user has provided the new details, then it gets stored in the user input details database.

For validating request

Just like the request membership function, this validating request function also has various sub-functions that get executed. For verifying the request, both external entities – admin and user along with their respective data storage are used. The data flow between functions, external entities, and data storage are presented below in a level 2 DFD:

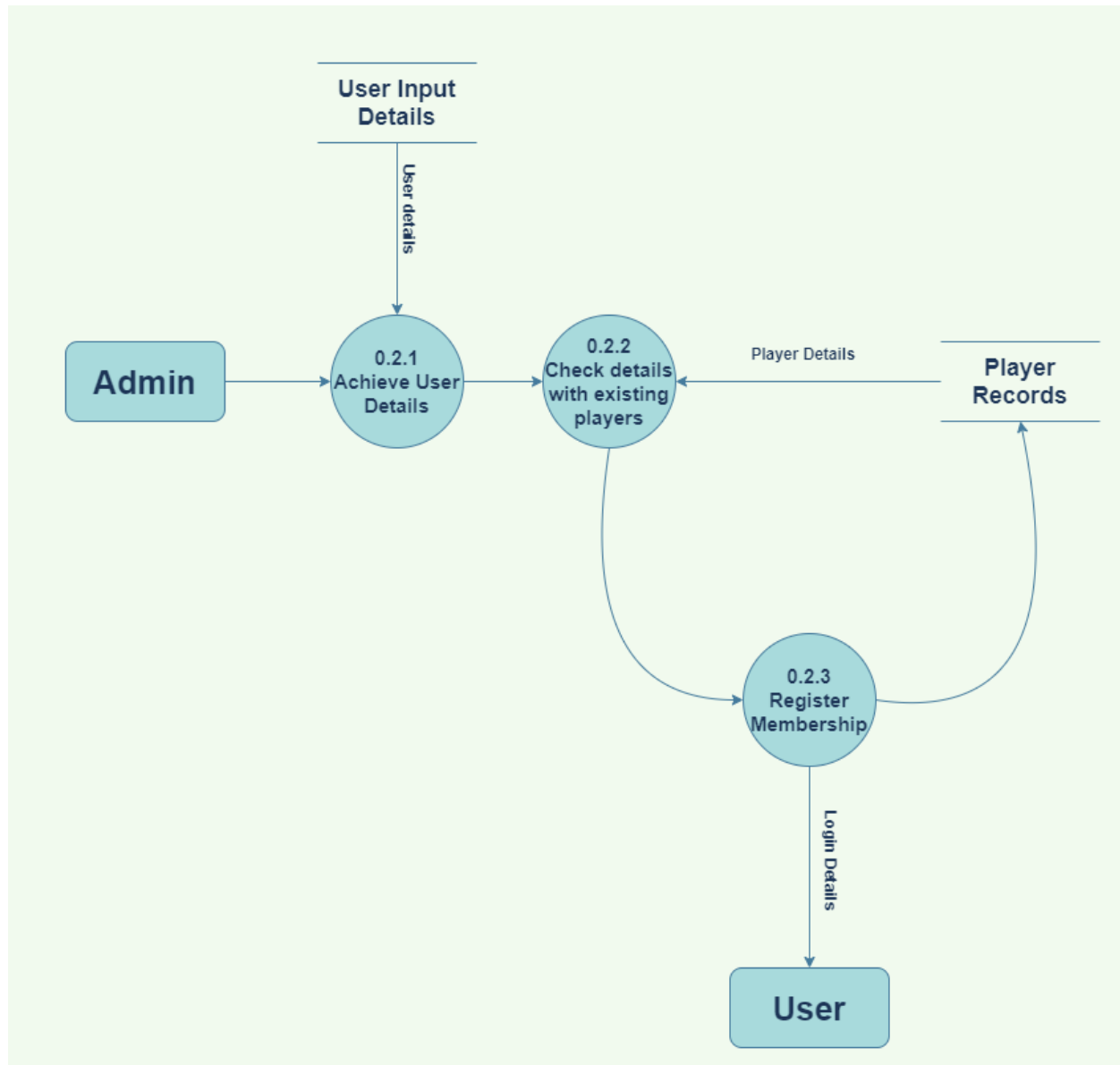


Figure 17 Data Flow Diagram Level2 (Register Membership) for 0.2 - validation request

Here, the admin receives user requests from the user input details database. So, once the request is received, the system then checks if that detail of the user is already in the player records or not. Finally, if the user is not a player, then their details get stored in the player records and they are provided with login details.

Design Specification Structure Chart

The Data Flow Diagram (DFD) itself is not enough for providing detailed information on how different functions work. Thus, a structure chart is designed that shows how a user can request membership. A structure chart is a chart that shows the functionality of a system by breaking down it into its lowest manageable levels.

A structure chart contains a hierarchical structure of modules (process or task of the system). In a structure chart, modules at top-level calls modules at a low level. When a module calls another module, it views the called module as a black box, passing required parameters and receiving results (Agarwal, 2019).

The structure chart for registering membership is presented below:

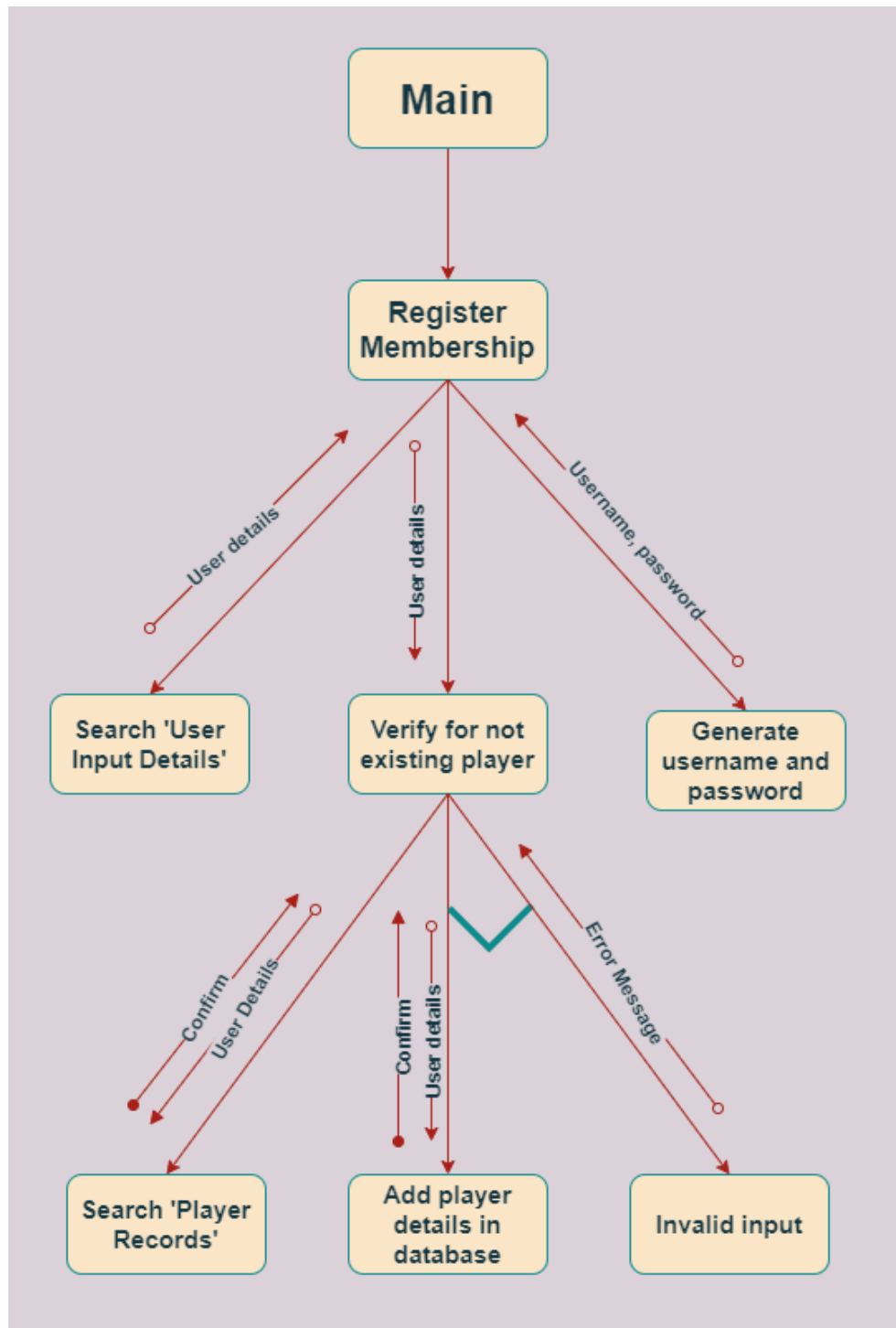


Figure 18: Structure chart of registered membership

Here, all functions in registering membership are presented by the structure chart in a detailed manner.

Module Specification (Mspecs)**Module Name:** Register Membership

Purpose: The purpose of this module is to register users as a player. This module also involves verification of the user details by the admin for their registration as a player.

Pseudocode**START****GET**user_Inputs **FROM** user DATABASE**CHECK****IF** user_Inputs **IN** player DATABASE**DISPLAY** "Invalid Input"**ELSE****ADD** user_Inputs **IN** player DATABASE**CREATE** user_Name, password**END****Input Parameters:** user_Details**Output Parameters:** user_Name, password**Global Variable:** player_ID**Local Variable:** user_Details, user_Name, password

Calls: During the verification of user inputs, details of the user are to be compared with the player records. If the user is already registered as a player, then an error message is shown else, the membership registration is done. Here, for this comparison to take place, the register membership module uses information from the player database. Also, for adding new user details in the player's database, this module calls another module with a player database.

Called by: When the system is installed, a user is limited to registering for membership. Thus, this module occurs first before any other functions could be carried out. This module is called the main module.

Conclusion

Here, a module for registering membership is created that interacts with two external entities – Admin and User. This module also involves verification of the user details by the admin for their registration as a player. In this module, input details provided by the user are stored in a separate database called User Input Details. Also, another database of Player Records is used in this module which is called from another module. Various sub-functions like verifying details, display details are used for creating this module.

5.2. Enrol Staff

Name: Aasutosh Rajopadhyaya

London met ID: 20048802

Introduction to module

For the enrolment of staff, the system first takes information from the individual who wants to get enrolled. The details like name, address, contact number, email, etc. need to be filled according. The details must be verified before registering him/her into the staff database. After, registration staff will get information regarding their enrolment and login details.

Environmental Model Specification

Context level diagram

The environmental model specification of enrolment of staff in the training academy is represented by a context diagram. Context diagrams provide additional insight into system requirements and flow.

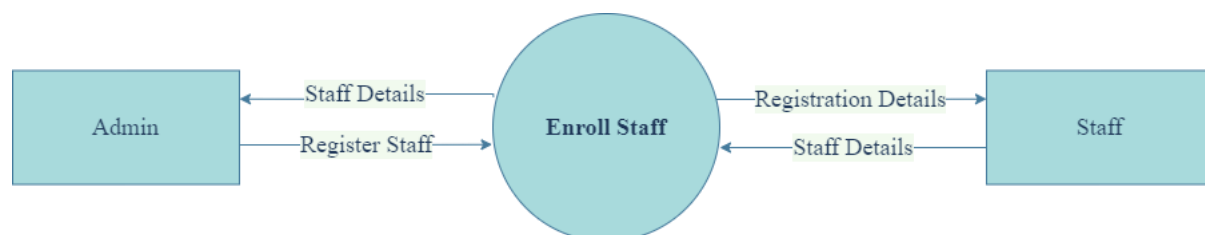


Figure 19 Context Level Diagram enroll staff

Staff will fill their details into the system, admin gets the details and it is an admin who enters staff's information into staff records. The staff gets registration details after successfully enrolling in the training academy.

Internal Model Specification

Internal model specification of staff's enrolment is represented by level 1 and level 2 data flow diagram respectively. DFD's helps in understanding the flow of data in the process of registration.

The Data Flow Diagram is as follows:

Level 1 DFD

In the first data flow diagram, staff fills all the details required for the enrolment of the staff in the training academy. The details are stored in the records of staff.

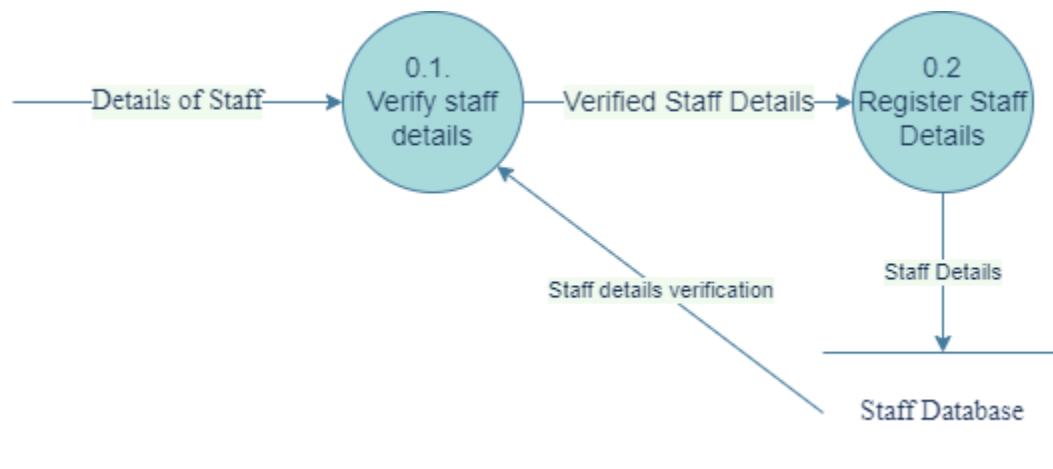


Figure 20 Data flow diagram level1 enroll staff

From the given figure the details of staff are accepted first and those details flow towards staff records for registration or enrolment of staff. Before registration, the details are first verified concerning staff records.

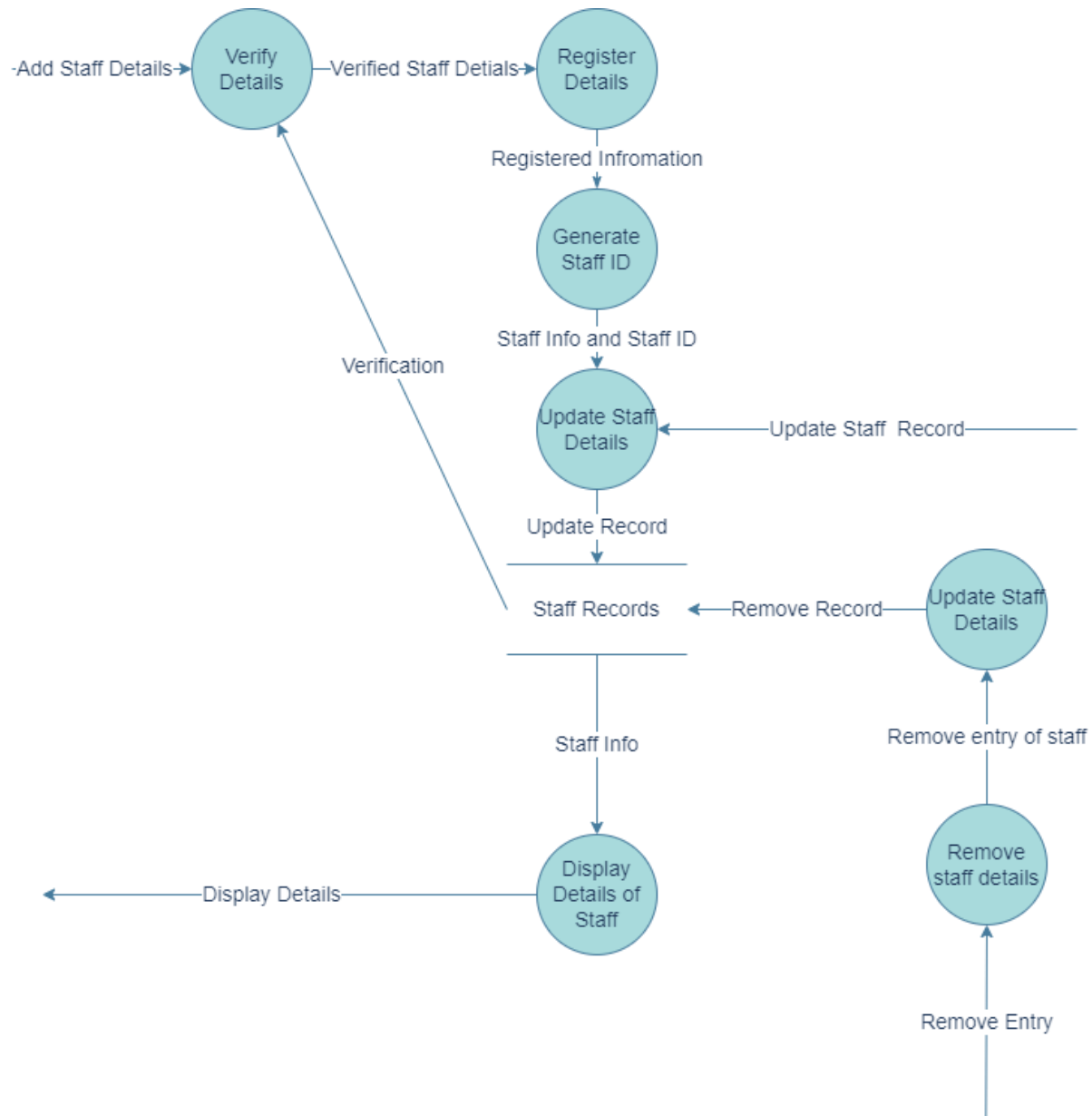
Level 2 DFD

Figure 21 Data flow diagram level2 enroll staff

From the given DFD, the details of staff are accepted first then sent for verification. Details are verified by checking pre-existing records stored in Staff Records. If verified, the details of staff are registered for enrolment. In the process, a staff id is also generated. Details of staff can be accessed by staff which is obtained from staff records.

To update a records, staff updates the required field, and then it is implemented on staff records. For deletion, the details are removed and updated which is implemented on staff records.

Design Specification

Structure Chart:

The design specification for enrolling staff is represented by design specification. Structure Chart is made for design specification which consists of a structure of the model in order. It describes how arguments are passed down and also shows how the function works. The structure chare of staff enrolment is as follows:

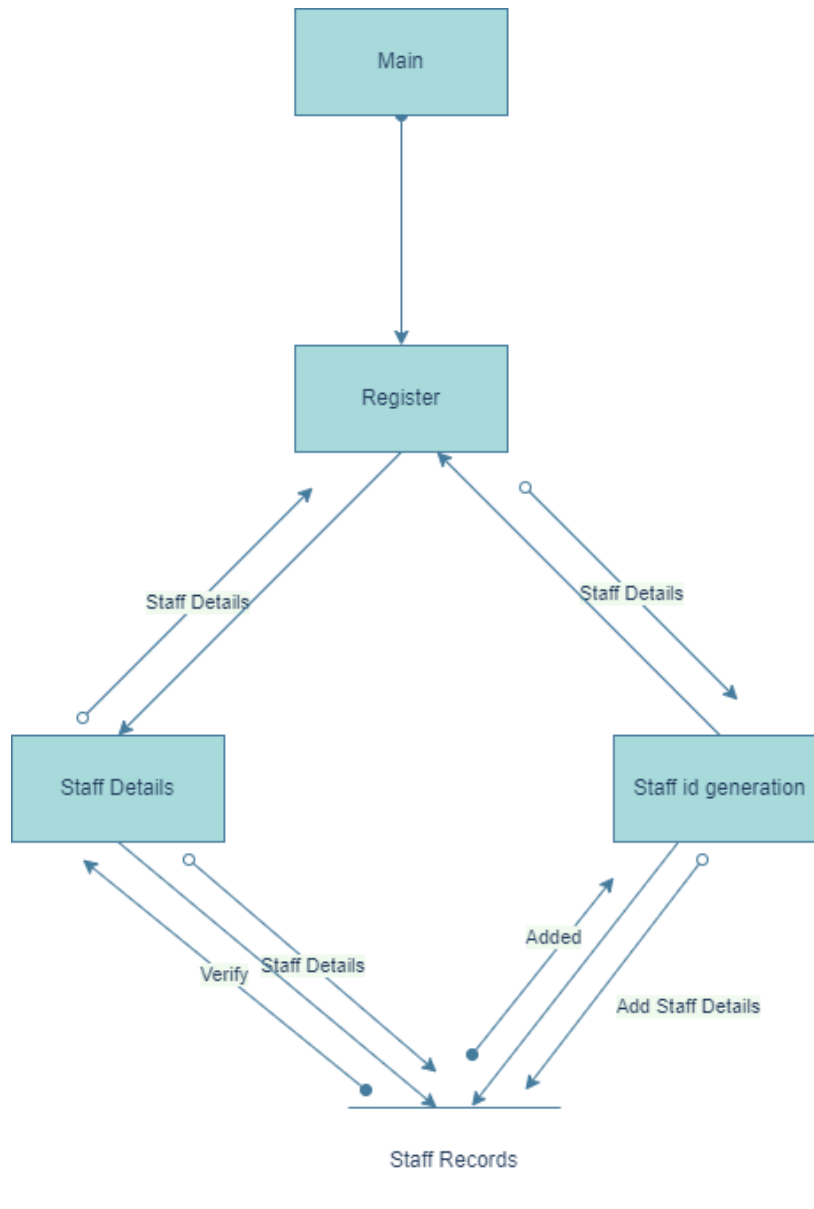


Figure 22 Structure chart enroll staff

From the structure chart, staff details are passed down for registration of staff for enrolment. These details are verified first and details of staff are registered if verification is passed. A staff id is generated and the staff details are sent to staff.

5.3. Purchase Football Kit

Name: Avigya Aryal

London Met ID: 20048838

Introduction

The individual task to define one of the coursework's modules is included in the second portion of the coursework. In this section, the module to purchase football kits is taken into account, and the module requirements are specified on an environmental and internal basis, as well as the definition of the structure chart and a description of the module specification, respectively.

Context Level Diagram

A context diagram is a high-level data flow diagram (referred to as a "Level 0" diagram). It comprises only one process node ("Process 0") that abstracts the entire system's operation about external entities.

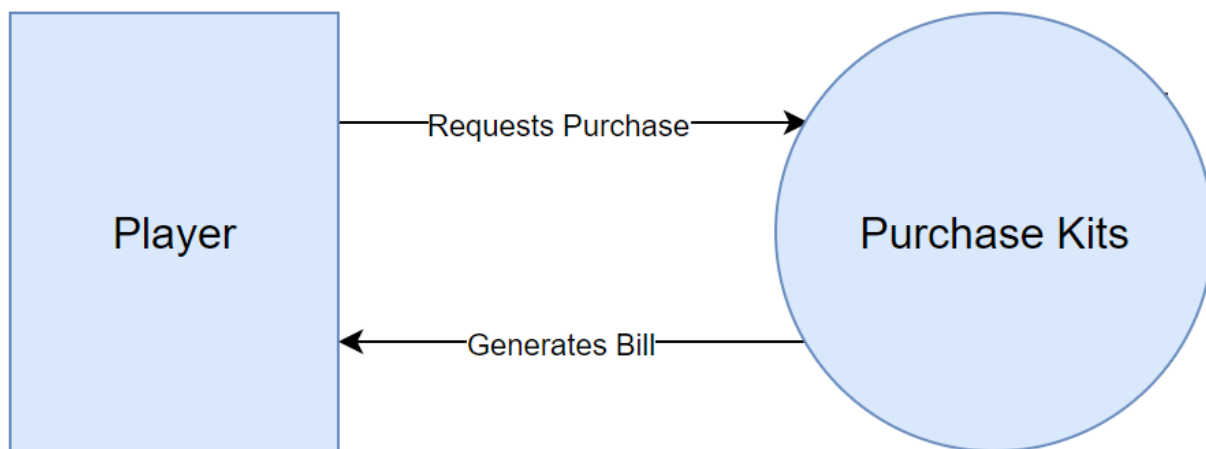


Figure 23: Context Level Diagram of Purchase Football Kits

Level 1 DFD

Data Flow Diagram is the abbreviation for Data Flow Diagram (DBD). DFD is used to model the data flow within a system or process. Additionally, it provides information about the inputs and outputs of each entity and the process as a whole. Data flow diagrams can be constructed in multiple nested layers.

The context diagram is split into numerous bubbles/processes in a 1-level DFD. At this level, we highlight the system's primary functions and decompose the 0-level DFD high-level process into subprocesses. The two-level DFD delves deeper into specific components of the one-level DFD. It can be used to plan for or record specific or important details about how a system operates.

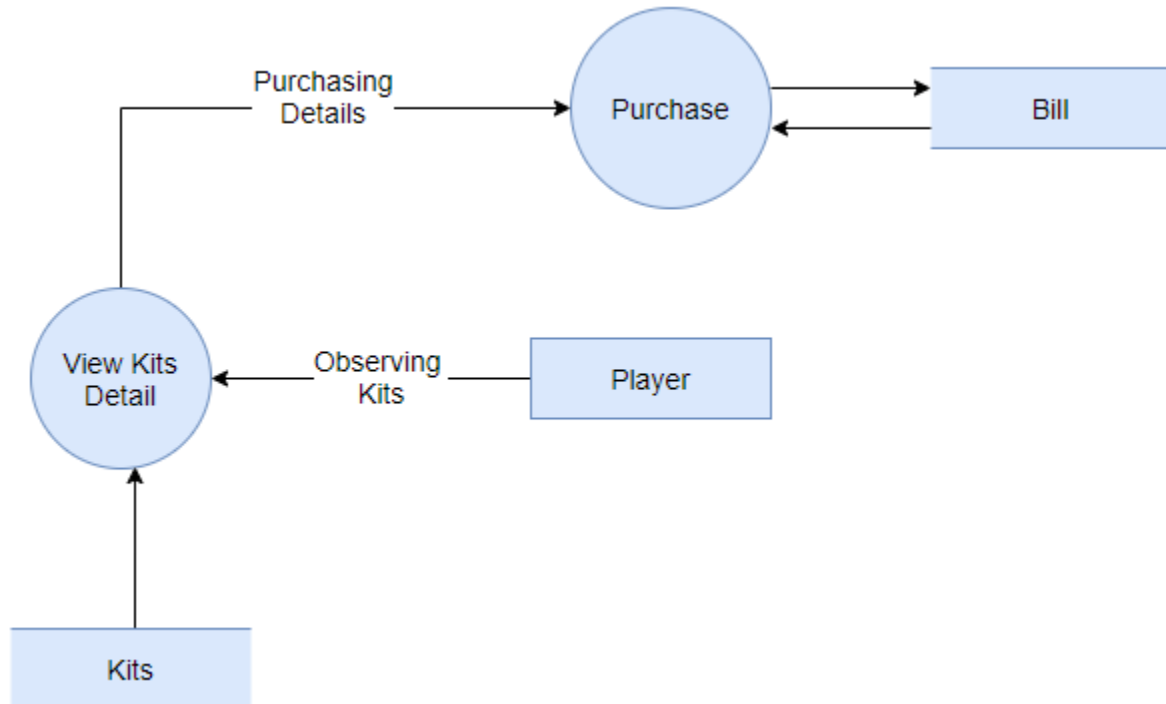


Figure 24: Level 1 DFD of Purchase Football Kits

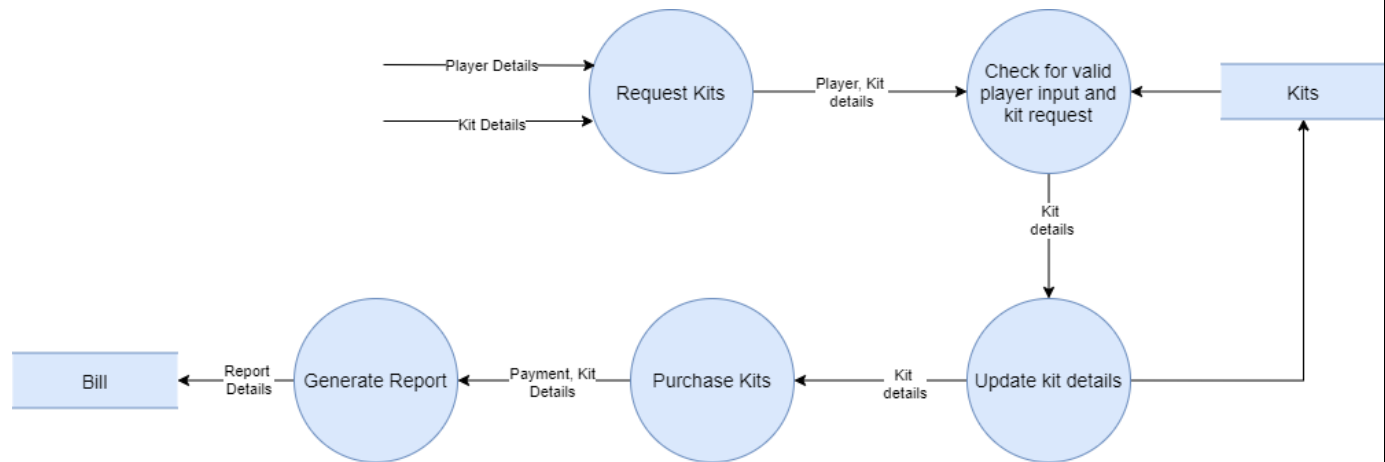
Level 2 DFD

Figure 25: Level 2 DFD of Purchase Football Kits

The player initially requests the purchase of the kits, with which the function generates a bill back to the player.

The player can observe kits from the view kit details function, which receives the information from the kits from their respective data stores. Now the function to purchase the kits is carried out by the incoming purchasing details and outgoing data flow towards the bill data store, which then returns the generated bill.

The kit request is then processed by the player, and the kit details are then validated by examining the input values. Lastly, the purchase transaction functions so that it sends the purchase details to the process the generation of a report finalized by the bill generator.

Design Specification

Structure Chart

The structure chart illustrates the modules' hierarchical structure. It breaks down the entire system into its lowest functional modules and describes in further depth the functions and sub-functions of each module (GeeksForGeeks, 2021).

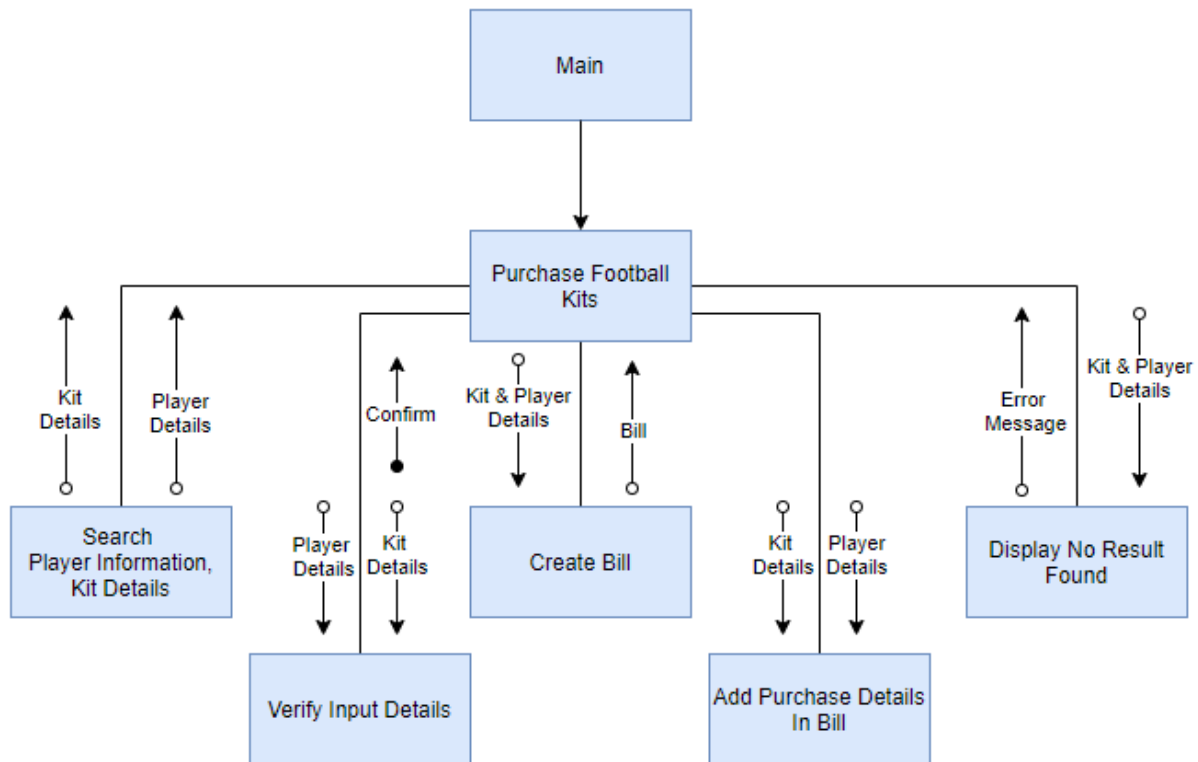


Figure 26: Structure Chart of Purchase Football Kits

The main module is sub-divided into one of its modules to purchase football kits. Now it is further divided into the functions of searching for player information and kit details, verifying input details, creating a bill, purchasing a transaction on the bill, and a data flow for invalid results. The module to search for the player's information and kit details sends the respective data and is received to verify the inputted details. Once the verification is approved, a control flow allows the generation of a bill that also receives the kit and player details functionality, in addition, to purchase details in the bill.

Module specification

MODULE NAME: Purchase Football Kits

PURPOSE: The following module details the transactional information regarding the kits purchased by the players on the generated bill.

Pseudocode:

DO

DECLARE Bill_Details

INITIALIZE Player ID

IF Player ID **MATCHES** Player ID FROM Player Information

IF Kit Details **MATCHES** Kit Details FROM Kit AND Quantity **GREATER THAN 0**

SET BILL_Details **EQUAL TO** {Bill ID, Kit Details, Quantity, Total amount}

ELSE Display "Enter correct Kit Details or Quantity"

ELSE DISPLAY "No any result found"

END DO

Input Parameters: Player ID, Kit Details, Quantity

Output Parameters: Bill ID, Bill Details

Global Variables: Kits Database, Bill Database, Player Database

Local Variables: Bill Details, Kit Details, Quantity

Called By: Main Module

5.4. Report Preparation

Name: Shreesu Thapa

London Met ID: 20049043

Introduction

The training academy has recently been established which has developed an application. The application generates different reports based on the training academy. The information of the player, payment details of the player, and the timetable generated by the training academy are generated as a final report. The player details include the detailed information of the player. The information player has registered while filling the registration form and the data player detail database is stored. The payment detail has the data of each player who has purchased the kit. The timetable detail stores the schedule of the player in a timetable database which is later exported as a report.

Environmental Model Specification**Context Level Diagram**

The context level diagram lists out the flow of information between the system and external components. The conceptual boundary of the context level diagram of this task is report generation which interconnects the players and staff. The flow of data in the player includes the player's detail stored in the player database. The player receives the report generated by the staff. The flow of data in staff includes the report management. The staff later views the report.

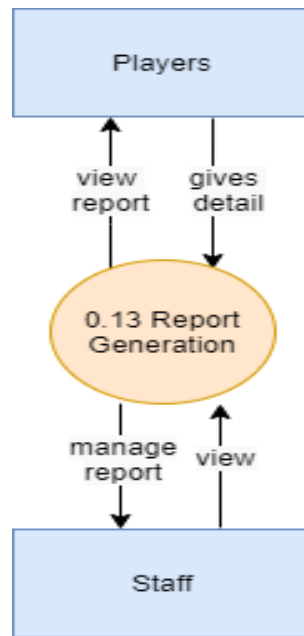


Figure 27 Context Level Diagram: Report Generation

Internal Model Specification

The internal model specification of the generated report is represented by level 1 and level2 Data Flow Diagrams.

Data Flow Diagram

Data Flow Diagram maps out the way information flows through a system or the application. The Data Flow Diagram consists of level1 and level2 diagrams. The process is represented by using circle, rectangle, and partial rectangle shapes for flowing the data through a system.

Level 1 DFD



Figure 28 Data Flow Diagram Level1: Report Preparation

The level 1 diagram consists of Report Data (database) which stores the data generated from the generated report process. The report details are viewed by the players, staff, and admin.

Level 2 DFD

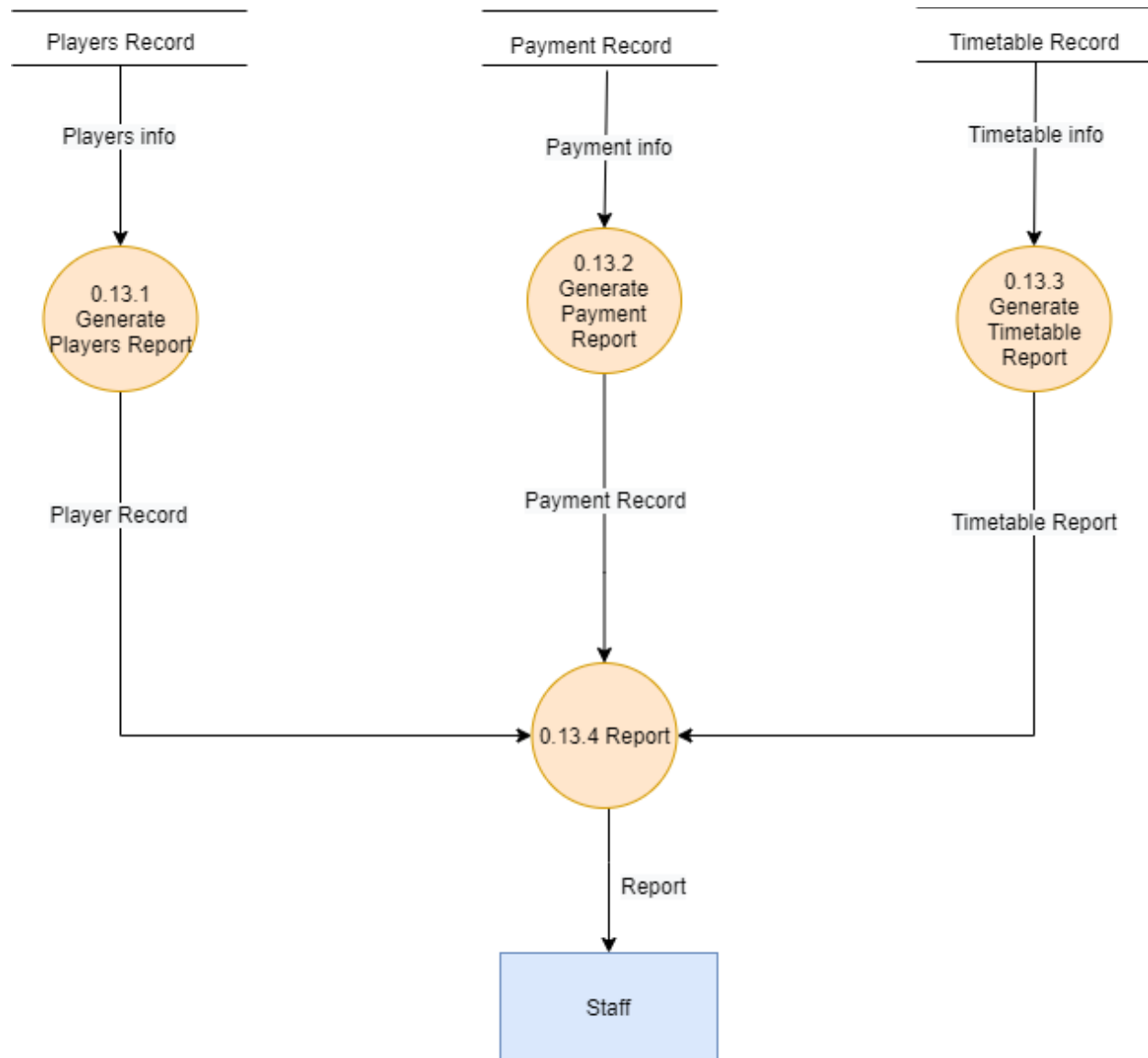


Figure 29 Data Flow Diagram Level2: Report Preparation

The Level2 Data Flow Diagram consists of three databases that store the three different data. The data are players' detail, payment details, and timetable details. The player's detail generates the report of the player, the payment detail generates the detail of the

purchased kit payment and the timetable detail generates the schedule. The report generated is merged into a final report which is checked by the staff.

Design Specification

Structure Chart

A structure chart is the breaking down of the steps performed in the application to the lowest manageable levels. The structure chart for the report preparation is shown below:

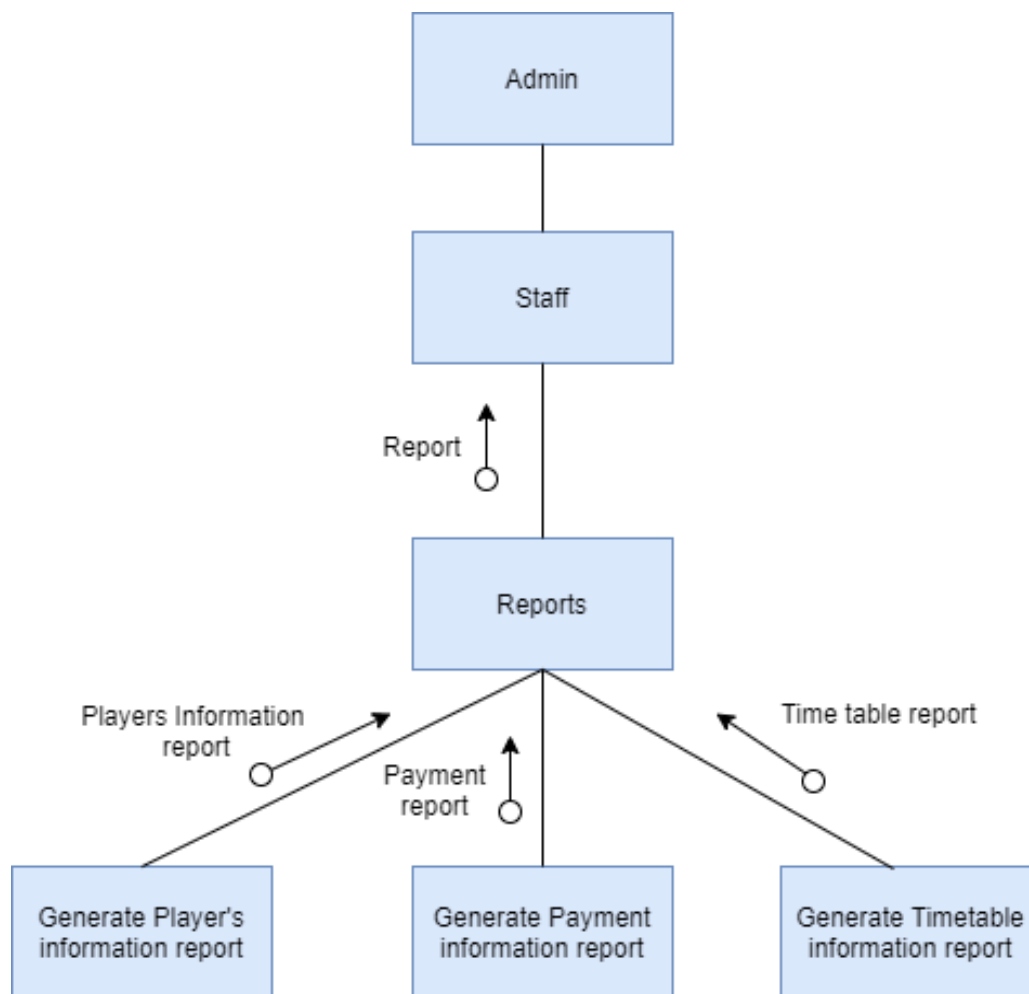


Figure 30 Structure Chart: Report Preparation

The above diagram, Structure Chart shows the modules and sub-modules involved in report preparation. The Player's information report, Payment report, and timetable report are merged into a file called report which is checked by staff and further by admin.

Module specification (Mspeccs)

Module specification is the information about an individual module which includes the module name, the purpose of the module, pseudo code, input parameter, output parameter, global variables, local variables, a method called, and method which calls the other variables.

1. Player's Information Report

MODULE NAME: Generate Player's Information Report

PURPOSE: The module Generate Player's Information Report produces a report of Player. The report is generated from the Player's Record tracked in the training academy.

PSEUDOCODE:

START

GET Player's information **FROM** player's record

GENERATE players_report

RETURN players_report

END

INPUT PARAMETERS: players_detail

OUTPUT PARAMETERS: players_report

GLOBAL VARIABLES: players_id

LOCAL VARIABLES: players_report

CALLS: The module calls 'Player's Record'.

CALLED BY: The module is called 'Reports'

2. Payment Report

MODULE NAME: Generate Player's Payment

PURPOSE: This module Generate Player's Payment produces a report of payment to be paid by the Players in the Training Academy. The report is generated from the Player's Payment Record.

PSEUDOCODE:

START

GET payment details **FROM** payment record

GENERATE payment_report

RETURN payment _report**END**

INPUT PARAMETERS: players_detail

OUTPUT PARAMETERS: payment_report

GLOBAL VARIABLES: players_id

LOCAL VARIABLES: payment_report

CALLS: The module calls 'Payment Record'.

CALLED BY: The module is called 'Reports'.

3. Timetable Report

MODULE NAME: Generate Timetable Report

PURPOSE: This module Generate Timetable Report produces a report of timetable prepared by the Training Academy for each player. The report is generated from the Timetable Information Record.

PSEUDOCODE:

START

GET timetable information **FROM** timetable record

GENERATE timetable _report

RETURN timetable_report

END

INPUT PARAMETERS: players_detail

OUTPUT PARAMETERS: timetable_report

GLOBAL VARIABLES: players_id

LOCAL VARIABLES: timetable_report

CALLS: The module calls 'Timetable Record'.

CALLED BY: The module is called 'Reports'.

4. Generate Report

MODULE NAME: Generate Report

PURPOSE: The main aim of this module Generate Report is to generate a report based on the relevant aspects of the application prepared for the training academy.

PSEUDOCODE:

START

GET players_report **FROM** Generate Players Report

GET payment_report **FROM** Generate Payment Report

GET timetable_report **FROM** Generate Timetable Report

COMBINE players_report **ON** payment_report **AND** timetable_report

GENERATE final_report

RETURN final_report

END

INPUT PARAMETERS: players_detail, payment_details, timetable_details

OUTPUT PARAMETERS: final_report

GLOBAL VARIABLES:

LOCAL VARIABLES: final_report

CALLS: The module requires reports from players' information, payment details, and Timetable details. Therefore, the module calls the three sub-modules in the method.

CALLED BY: The module is called by the staffing entity and admin entity.

Conclusion

The report preparation of the application contains detailed information of players, payment, and timetable schedule. The final report was generated. The individual task to prepare the report of topic Report Preparation was completed on time. The coursework was difficult to perform though, consulting the module teachers and tutors the individual task was completed on time so was the group coursework. All the topic was covered in the report of individual task. Therefore, the report was generated for the Report Generation topic.

5.5. Take a Mock Exam

Name: Shishir Subedi

London Met ID: 20049039

Module Name: Take a mock exam

Environmental Model Specification Context Level Diagram

The Context Level Diagram is the diagram that illustrates the whole system or process being modeled or analyzed. It's designed to provide you with a rapid overview of the system, showing it as a single high-level process with external links. It is written in such a way that it can be understood by a wide variety of individuals, including stakeholders, business analysts, data analysts, and developers.

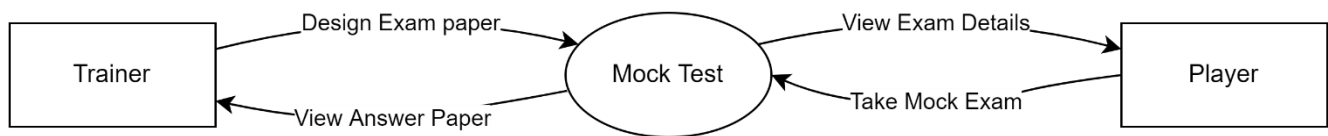


Figure 31: Context Level Diagram for Taking a Mock Test

The mock test is given by the player to upgrade to an intermediate training level. To take the mock test the trainer should first design the exam paper and submit it to the system. The trainer can later check the answer paper submitted by the player. The player can view exam details such as exam time, exam date, exam paper, and so on. The players then can take the mock exam and submit the paper for review.

Internal Model Specification Level 1 DFD

The Level 1 DFD is a more in-depth examination of the Context Level Diagram's components. It breaks down the high-level process of the Context Diagram into its subprocesses, demonstrating the system's basic operations.

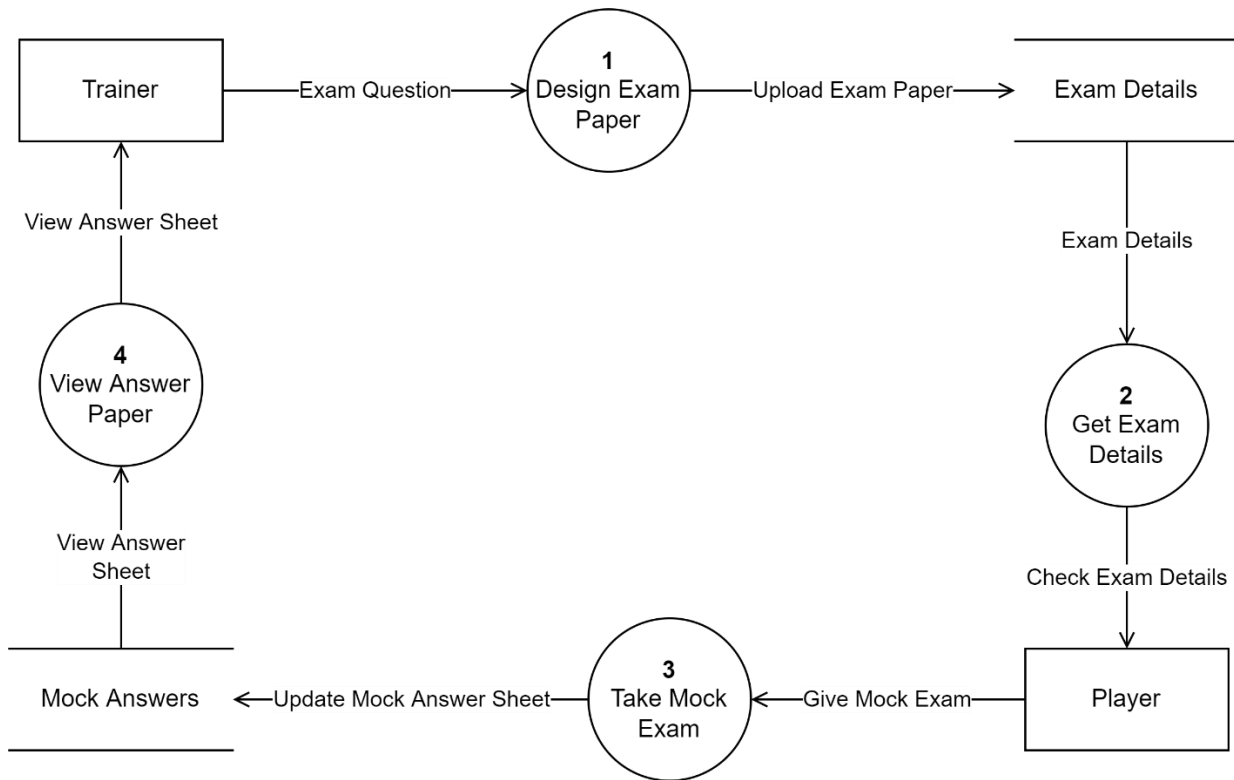


Figure 32: Level 1 DFD for Taking a mock exam

The Level 1 DFD for taking a mock exam clarifies the process taken for a player to take the test. The trainer initially designs the paper which is stored in the exam details database. The exam details can be accessed by the player and the player then take the mock exam. The mock exam answer is then submitted and stored in the Mock Answers database which can be accessed by the trainer for review.

Level 2 DFD

Level 2 DFD expands on the topics introduced in Level 1. It goes over the Level 1 DFD procedures in greater detail and explains the subprocesses involved. It describes every step necessary to accomplish a function to show how the system works.

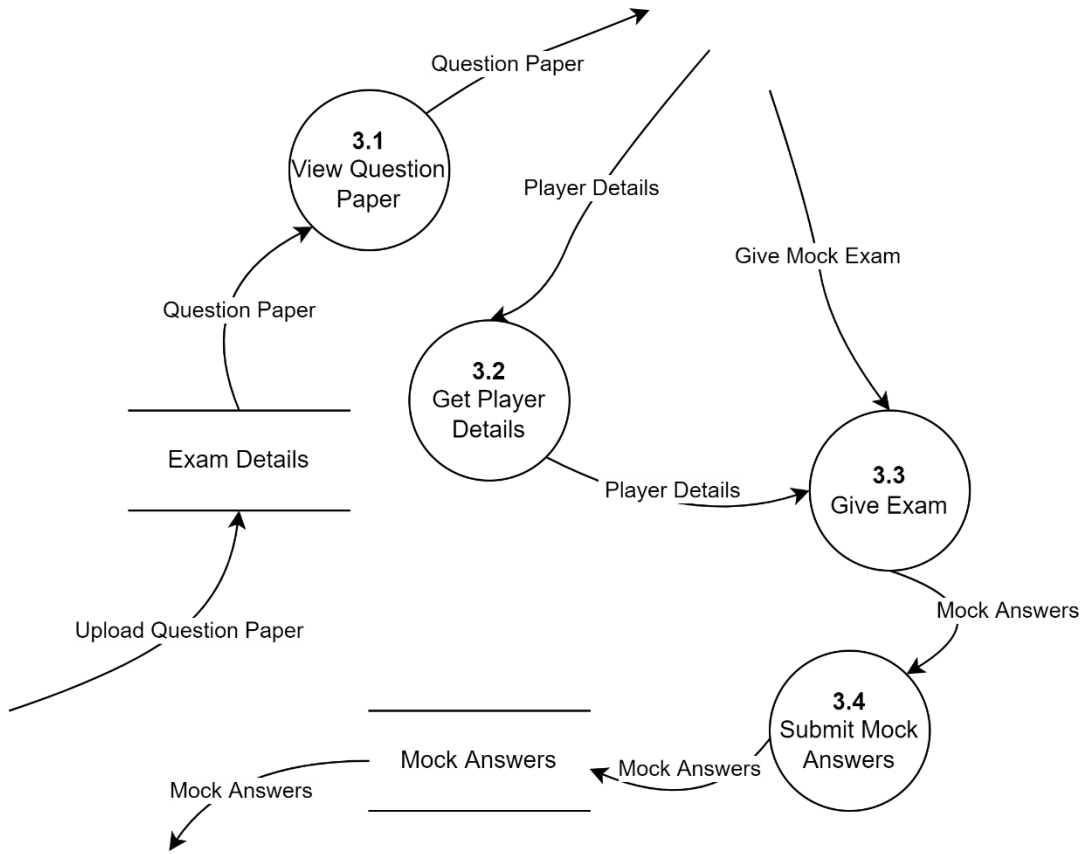


Figure 33: Level 2 DFD for Taking a Mock Test

The level 2 DFD further explains how the players take the mock test. The trainer first uploads the question paper in the Exam Details database which can be viewed by the player. Then the player detail is accessed from the Player Details database which provides Player ID. The player then gives the mock exam along with the player details and submits the mock answers. The answers are then stored in the Mock Answers database along with player details which are later viewed by the trainer for review.

Design Specification Structure Chart

A structure chart is a diagram that illustrates a program's hierarchical structure. The diagram breaks down the whole system into its smallest functional modules and goes into further depth on the functions and sub-functions of each module.

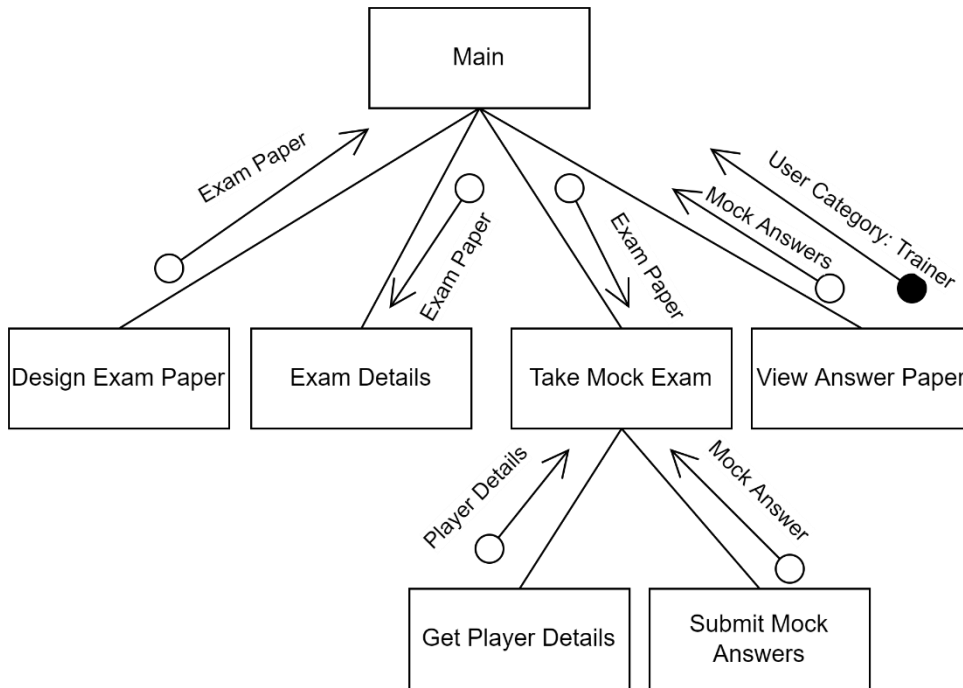


Figure 34: Structure Chart for Taking a Mock Exam

Module Specification

Module Name: Take a mock exam

Purpose: This module obtains the Exam details which include exam date, exam time, exam question paper, Player details like Player ID, Player Name and enable the players to take the mock exam. Players then can submit the mock test answers to be reviewed by the trainer.

Pseudocode:

DO

DECLARE Exam Details, Player Details

UPDATE Exam Paper

INITIALIZE ExamDate, ExamTime, ExamPaper, PlayerID, PlayerName

INPUT PlayerID, PlayerName, MockAnswer

DISPLAY MockAnswer

END DO

Input Parameters: PlayerID, PlayerName, MockAnswer

Output Parameters: Mock Answer

Global Variable: PlayerID

Local Variable: PlayerName, Mock Answer

Conclusion:

The report describes and explains the process involved when players of T-14 Training Academy take a mock exam to upgrade their training module. The mock exam upgrades the player's training module to intermediate. The report has a detailed explanation of the process from designing a question paper for the mock exam, enlisting the question paper in the exam details database, to viewing the question paper by the player, and taking the mock exam and submitting it to the trainer for review. The process is explained using Context Level Diagram, Level 1 and Level 2 DFD, and Structure Chart for the Take mock exam module of the main system.

6. Summary

The application for the Training Academy (T-14) has been successfully built. The Training Academy application fulfills all the requirements from the scenario. The report has been prepared for the application which includes the Introduction, SRS, Data flow diagram, Structure Chart, Process Specification, and other topics required for the project management system. The project was developed professionally, which includes all the aspects of the Project Management System.

However, the project consists of different diagrams which define the project diagrammatically. The diagram consists of a context level diagram, data flow diagram level1, data flow diagram level2 and structure chart. The diagram makes the project clearer to the viewer reading the document.

The group task has been completed by the enormous effort of all the group members. The task was divided among the group member and was completed on time. Different problems were faced regarding the coursework which was solved by conducting different meetings mentioned in the assignment diary. The unsolved problems were solved by taking the help of the module teacher and by researching on different websites. The overall coursework was completed on time and the individual task was also divided among the team members randomly. The coursework was merged and submitted to the portal on time.

The group project developed different skills of software engineering which includes requirement analysis, project management, structured software engineering and problem-solving skills.

7. References

- Agarwal, A. (2019). *GeeksforGeeks*. Retrieved January 1, 2022, from <https://www.geeksforgeeks.org/software-engineering-structure-charts/>
- Agarwal, A. (2019). Software Engineering | Structure Chart. *Software Engineering / Structure Chart*.
- Computer Science Department, University of Cape Town. (2011). *University of Cape Town*. Retrieved January 1, 2022, from https://www.cs.uct.ac.za/mit_notes/software/htmls/ch06s06.html
- GeeksForGeeks. (2021). *Software Engineering / Structure Charts*. Retrieved January 1, 2021, from <https://www.geeksforgeeks.org/software-engineering-structure-charts/>
- Lucidchart. (2019). *ER Diagram (ERD) - Definition & Overview / Lucidchart*. Retrieved 12 23, 2021, from https://www.lucidchart.com/pages/er-diagrams#section_0
- Sommerville, I. (2016). *Software Engineering* (10 ed.). Singapore: Pearson Education South Asia Pte Ltd.