INTRODUCTION

1. INTRODUCTION

1.1 System Overview

The project titled by "**FARSA**" is the part of our Sixth semester. This is project which is done by B.Voc software Development student. So I, selected the topic FARSA (Fire and Rescue Service Academy, Thrissur) Website.

Kerala fire and rescue services academy is situated in Ramavarmapuram in Thrissur city. To ensure community safety by creating basic awareness regarding fire safety, life safety and Disaster Management among the people and thereby mitigate the fire loss and improve effective and timely rescue and life saving activities. The academy offers basic training for fire and rescue officer, in addition there are practical courses also. The academy can train and accommodate 350 trainees in the institute and the hostel . An Olympic type pool is here for training.

This project is mainly consider the working of fire and rescue academy Thrissur. The Director of the institution will be act as the admin of the web application. By this web application all the daily activities are monitored and also control the duties of fire officers.

1.2 Objectives of the System

The main objective of the project is to make the whole working academy smoothly. Here this project is developed for the client requirement. Basically, the whole of the academy is as manually. But this may cause some difficulties. Through this project the daily routine and duties can be monitored by the admin and also by the director of the academy. Each staff can post their attendance daily. This will help director to understand the how many of staffs are present on that day. Also can assign duties to staffs. So this will help the whole working of the academy smooth and digitally accessible to everyone.

Here also academy provide course for the public and students. We have the registration And course selection in the site. And other common service also providing. The user can understand all about the academy.

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

System analysis is the reduction of the entire system by studying the various operations performed and the relationship with the system and the requirement of its successor. It is the most essential part of the development of a project. System will be analyzed in terms of their objectives, and the input, processes and output required achieving these objectives. The aims are to identify the boundaries of the system, the subsystems and the interfaces between subsystems. A system can be defined as an orderly grouping of independent component linked together according to a plan to achieve a specific objective.

System analysis is the term used to describe the process of collecting and analyzing facts in respect of existing operation of the situation prevailing so that an effective computerized system may be designed and implemented of proved feasible. To analyze a system means to study the system in detail. The analyst has to understand the functioning and concept of the system, before design the appropriate computer based system that will meet all the requirements of the existing system. The system analyst has to carry out a customary approach to use the computer for problem solving.

The system study phase involves the initial investigation of the structure of the system, which is currently in use, with the objective of identifying the problem and difficulties with the existing system. The major steps involved in the phase included defining the user requirements and studying the present system to verify the problem. The performance expected by the new system was also defined in this phase in order to meet the user requirements. The information gathered from various documents were analyzed and evaluated and the findings reviewed in order to establish specific system objectives.

2.1 Existing System

Currently there is no existing system is used in this academy. Here all the working and enquires are done through the official site of fire and rescue service academy, kerala. For the smooth working of all the process happening in the academy they preferred to build a site for access their all process happening in the daily duties in the academy.

The existing system is not developed yet. If user have any enquiry they have to communicate through the official. But there will not be all details of this particular academy. In this academy all delay activities are stored as manually. And attendance also noted in register. Suppose the director want to check who were leave in a day, he have to analyze the whole register. To make it is easy we need a digital system. Here they can access daily routine and works. And also verify it.

2.2 Proposed System

In the proposed system all disadvantages can be overcome. Here currently no existing system is used in this academy. All staff can post their daily attendance and also know their daily works by login into their account. At the beginning director or admin of this application should approve each profile of staff. Then only they login into their site. Director can view the daily attendance of the staff and there by assign work to each staff. They can communicate through the notification section provided in the each profile.

User can also get information about all details related to the academy. Student also apply for the different course that is added by the admin or director.

2.3 Advantages of Proposed System

- The proposed system will make all current activities of the academy more easy.
- Fast access of information
- User Friendly

2.4 Requirement Analysis

Requirement analysis is the first technical step in the software engineering process. It is at this point that a general statement of software scope is refined into a concrete specification that becomes a foundation for all the software engineering activities that follow. Analysis must focus on the informational, functional and behavioral domains of a problem. To better understand what is required, models are created, the problem is partitioned, and representation that depict the essence of requirements and later, implementation detail, are developed.

➢ Functional Requirements

- Admin can login into the site and approve director. Then only all registered director can login approve staffs.
- Users can login using their email and password and can get information an access their services

2.4.1 Identification of Need

Currently there is no existing system is used in this academy. Here all the working and enquires are done through the official site of fire and rescue service academy, kerala. For the smooth working of all the process happening in the academy they preferred to build a site for access their all process happening in the daily duties in the academy.

2.4.2 Preliminary Investigation

The investigation method used in this project is direct. The data is collected by direct contact with academy. Investigated all current activities in the academy. And collected their requirements and the needed information for development of the site.

2.5 Feasibility Study

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test technical operation and economic feasibility, for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in feasibility study portion of the preliminary investigation.

2.6 Project Planning

As per the initial plan the project has to be completed within three months. For the system study I a meet Vismaya our project leader . The PL is the one who give us the information about the activities. The technical assistance for the project is provided by Ms.Vismaya, my guide and one of the Senior Software Developer at Fab studioz. According to the plan, hardware requirements of this project are monitor, keyboard, mouse, Intel dual core processor, 4 GB RAM and 160 GB hard disk etc. The software used were, Python as

development environment for Flask, Sql alchemy as server and Google chrome as browser. The deadline of project is April last.

In the project "FARSA", the operating system used is windows 10 which is a personal computer operating systems developed by Microsoft, a version of windows NT. WindowNT. Windows 10

Chooses because of its window management features. Live Tiles and Lock screen. The apps We use in windows 10 can feed you information without you even having to open them.

SQL Alchemy consists of two distinct components, known as the **Core** and the **ORM**. The Core is itself a fully featured SQL abstraction toolkit, providing a smooth layer of abstraction over a wide variety of DBAPI implementations and behaviors, as well as a SQL Expression Language which allows expression of the SQL language via generative Python expressions. A schema representation system that can both emit DDL statements as well as introspect existing schemas, and a type system that allows any mapping of Python types to database types, rounds out the system. The Object Relational Mapper is then an optional package which builds upon the Core. Many applications are built strictly on the Core, using the SQL expression system to provide succinct and exact control over database interactions..

Mature and high performing architecture: Over seven years of constant development, profiling, and refactoring has led to a toolkit that is high performing and accurate, well covered in tests, and deployed in thousands of environments. With virtually every major component in its second or third full iteration, SQL Alchemy 0.6 is roughly twice the speed of older 0.4 versions from just a few years ago, and versions 0.7 and 0.8 continue to improve. Its raw execution speed is competitive with comparable tools, and advanced ORM features like its unit of work, in-memory collections, eager loading of collections via joins or secondary sub selects, and other optimizations allow SQL Alchemy's ORM to emit fewer and more efficient queries than in any previous version simplify certain steps, such as inserting or deleting a data record. For client programmers this has the advantages that

they do not have to process the tables directly, but can rely on SPs. Like views, SPs help in the administration of large database projects. SPs can also increase efficiency. MySQL has supported SPs since version 5.0.

2.7 Project Scheduling

Pert Chart

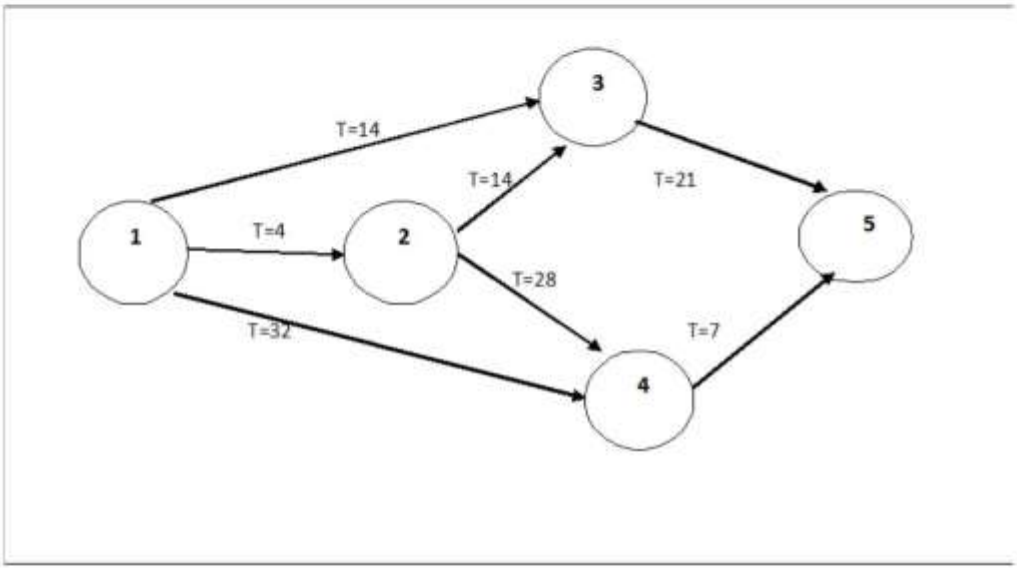


Fig 1.1**:** PERT Chart

PERT chart represents the project in a graphical form. It is used to schedule, organize and coordinate tasks within the project. It provides information about expected completion time of project & parallel works. The milestones are numbered 1, 2, 3, 4 and 5. 'T' is the time (in weeks) between milestones and it is the maximum time taken to complete the activities. The milestones are,

Milestone 1: System analysis and planning

Milestone 2: Design

Milestone 3: Coding

Milestone 4: Testing

Milestone 5: implementation maintenance

According to the project FARSA, after the system analysis, the time taken for system study is 2 week. If system study is excluded, then the time taken for database/interface design is 4 weeks. After design process, the time taken to coding process is again 3 weeks. The testing can be performing parallel with or after coding. The time taken for testing is 2 weeks. The time taken for implementation and maintenance is near 2 months.

Gantt Chart (17/12/2021 to 15/04/2022)

The Gantt chart shows the exact time taken for complete a particular process or task. The project starts at first. System analysis is completed after 2 week. After two week of system study, system design starts. System design includes interface design, database design etc. after 2 week of system study, system design started. System design includes interface design, database design etc. after 3 week of system design, coding started. It takes near 4 weeks of time. Validation process performs parallel with coding. After testing, implementation is performed. The last 2 weeks was for testing and implementation at user's site. Following is the Gantt chart which shows the track of project. The darkened portion indicates the completion of process. Following is the Gantt chart which shows the track project.

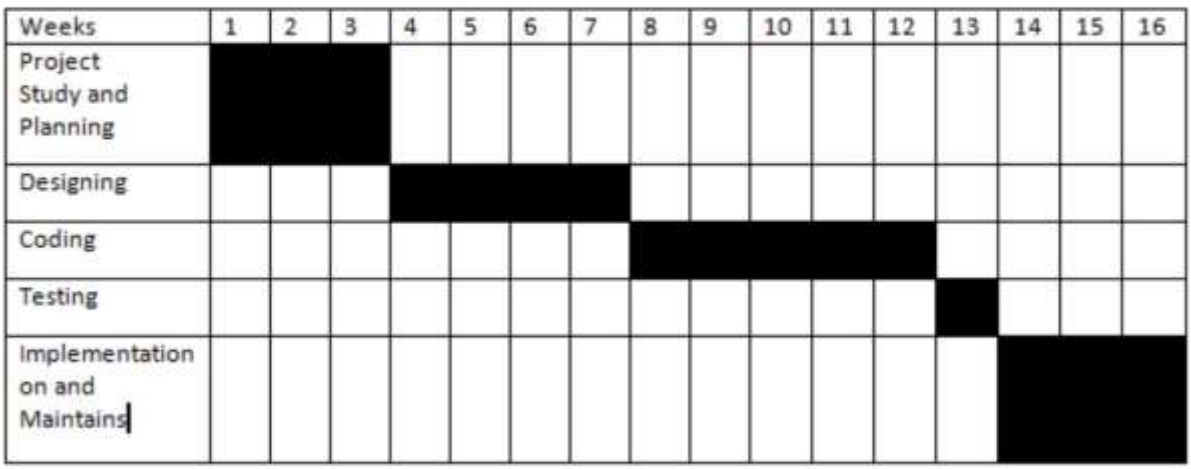


Fig 1.2**:** Gantt Chart

2.8 Software requirement specification (SRS)

1. Introduction

Computer and information technology has a major influence on the society. And the society is becoming more and more dependent on technology. Going on is an area of simplifying almost all complicated works using computer. The last few years witnessed a tremendous increase in the capabilities and use of computers. Manual processing makes the process slow and the other problems such as inconsistency and ambiguity on operations. The proposed system intends user-friendly operations which may resolve ambiguity. By considering all this factors, the application produced, which performs the social services simply and effectively.

1.1 Purpose

All staffs can post their daily attendance and also know their daily works by login into their account. At the beginning director or admin of this application should approve each profile of staff. Then only they login into their site. Director can view the daily attendance of the staff and there by assign work to each staff. They can communicate through the notification section provided in the each profile.

1.2 Scope

To ensure community safety by creating basic awareness regarding fire safety, life safety and Disaster Management among the people and thereby mitigate the fire loss and improve effective and timely rescue and life saving activities. The academy offers basic training for fire and rescue officer, in addition there are practical courses also. The academy can train and accommodate 350 trainees in the institute and the hostel . An Olympic type pool is here for training.

This project is mainly consider the working of fire and rescue academy Thrissur. The Director of the institution will be act as the admin of the web application. By this web application all the daily activities are monitored and also control the duties of fire officers.

1.3 Definitions, Acronyms and Abbreviations

DB – Database`

IDE – Integrated Development Environment

1.4 References

The technical assistance for the project is provided by Ms. Vismaya, Software Developer of Software The senior software Developer of FAB STUDIOZ

2. Overall Description

Kerala fire and rescue services academy is situated in Ramavarmapuram in Thrissur city. To ensure community safety by creating basic awareness regarding fire safety, life safety and Disaster Management among the people and thereby mitigate the fire loss and improve effective and timely rescue and life saving activities. The academy offers basic training for fire and rescue officer, in addition there are practical courses also. The academy can train and accommodate 350 trainees in the institute and the hostel . An Olympic type pool is here for training.

This project is mainly consider the working of fire and rescue academy Thrissur. The Director of the institution will be act as the admin of the web application.

2.1 Product Perspective

The project titled by **FARSA.**I enquired to the academy, and collected their requirements by direct contact

2.1.1 System Interface

FARSA is a website get all and information about the Fire and Rescue service Service academy, Thrissur. And smooth working daily activities.

2.1.2 User Section

Software **FARSA** contains 5 main modules based on the services provided. That are,

- Admin Login
- Director Login
- Staff Login
- Students Login
- User Login

2.1.3 Hardware interface

Processor : Intel Pentium
RAM : 4GB
HDD : 160 GB or Higher
System bus : 64 BIT
Monitor : Hp monitor
Keyboard : Logitech (104keys)
Mouse : Logitech

2.1.4 Software interface

Platform : Windows 10
Language : Python
Backend : SQL Alchemy
Browser :Chrome

2.1.5 Communication interface

The user can communicate with the system through Graphical User Interface.

2.1.6 Memory constraints

I3 Intel Pentium processor or higher with a minimum of 8GB RAM and 2 TB of hard disk space will be required so that the software performs its functions in an optimum manner.

2.1.7 Operations

The operations of general user are limited, but the registered users can view the projects and import project.

2.1.8 Site adaptations requirements

The user must careful when they provide personal details such as username, password, email id etc.

2.2 Product Functions

The system will allow access only to authorized users with correct email id and password. Following are some important functions of FARSA.

- Login facility for authorized users.
- Highly secure system.
- Perform modification only by administrator.
- The administrator can see all registration details .

2.3 User characteristics

All public users must have email and a password. And also the administrator has One constant email id and password for login purpose.

2.4 Constraints

Only registered user can enter in the site.

2.5 Apportioning requirements

Not required.

3. Specific requirements

This section describes the information required by the developer of the system.

3.1 External interface

The external interface of FARSA system must be a representation of data storing. Used a standard design for security tool.

3.2 Functions

Implement all the user requirements as per requirement specification.

3.3 Performance requirements

FARSA is a security tool. It must be secured and cannot be share personal details.

3.4 Software system attributes

3.4.1 Security

For the account security and privacy there is username and password entry before login process. For ensuring the reliability of a user, all users have to provide their basic details and security requirements such as name, and email id etc. Administrator will also have to enter his username and password.

3.4.2 Maintainability

Maintenance of system is not more difficult. The information is updated by authorized user. Overall maintenance is handled by system administrator.

3.4.3 Portability

It is a web application which is portable at any system having a browser for opens the website. It supports any windows operating system such as windows XP, vista etc. The language used in this project is JAVA which is platform independent.

3.4.4 Change management process

Overall maintenance is handled by system administrator.

3.4.5 Document approvals

When the requirements are gathered according to the user, SRS is then finally reviewed, approved and signed by the developer and user (the administrator of the site). This SRS serves as a contract for software development activities between developer and user.

2.10 Software engineering paradigm applied

The system FARSA is developed according to waterfall model. Each step of development is done in sequential order. I choose this method of development for proper management of limited time. According to Avails system, water fall model is more simple and easy to implement. The sequential order of execution of development helps me to finish the project fast. This model is also less expensive to handle.

After collecting the data, prepared a rough design of interface and database. The repeated analysis of design is finally reached at a perfect and 6 errorless design. The diagrams showing the data flow, actions of system, and relationship between entities will prove the correctness of design.

After the phase of design, I moved to coding section. The first step in coding was creating all forms using front end language, Python. The scripting in this project is enabled using flask. All remaining coding including validation are done in python.

After coding, testing is performed. Prepared each test case and test each unit of code according to the test case. After modular testing, integrate all tested module and perform system testing. Finally the completed software is submitted. Here the system administrator is considered as who upload the content, who has the control over the property they upload. The updates can do only from permitted administrator.

2.11 Data Models

2.11.1 Data Flow Diagram

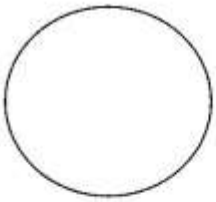
A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. It's a structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

Data flow diagram symbols :-

- Source or Destination of data

- Data Flow



- Process

- Storage

Steps to Construct Data Flow Diagrams

Four steps are commonly used to construct a DFD

- Process should be named and numbered for easy reference. Each name should be representative of the process.
- The destination of flow is from top to bottom and from left to right.
- When a process is exploded in to lower level details they are numbered.
- The names of data stores, sources and destinations are written in capital letters.

Rules for constructing a Data Flow Diagram

- Arrows should not cross each other.
- Squares, circles and files must bear
- Decomposed data flow squares and circles can have same names.
- Draw all data flow around the outside of the diagram.

2.11.2Activity Diagram

Activity diagrams are graphical representations of workflows of step wise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Initial node

Final node

Decision node

Control Flow

Activity

Fork.

Join.

Fork is a black bar with one flow going into it and several leaving it. This denotes the beginning of parallel activity. Join is a black bar with several flows entering it and one leaving it. All flows going into the join must reach it before processing may continue. This denotes the end of parallel processing.

2.11.3 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represent the whole system.

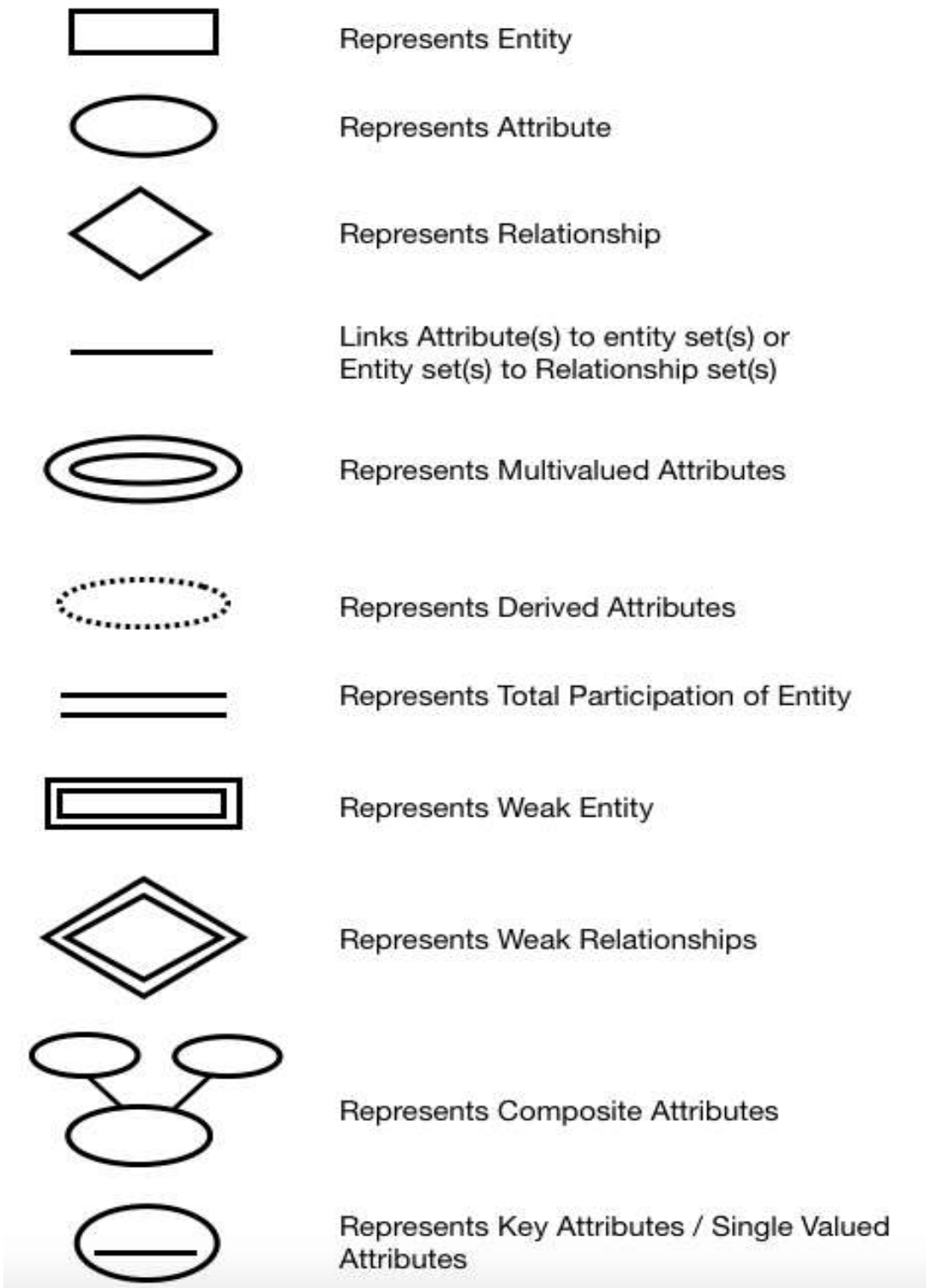
2.11.4 E-R Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems. Uses of entity relationship diagrams.

- Database Design: ER Diagrams are used to model and design relational databases,in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model.) In software engineering, an ER diagram is often an initial step in determining requirements for an information systems project. It’s also later used to model a particular database or databases. A relational database has an equivalent relational table and can potentially be expressed that way as needed.

- Database troubleshooting: ER diagrams are used to analyse existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it’s going wrong.
- Business information systems: The diagrams are used to design or analyse relational databases used in business processes. Any business process that uses fielded data.

ER diagram symbols :-



SYSTEM DESIGN

3. SYSTEM DESIGN

3.1 Fundamental Design Concepts

The most creative and challenging phase of the system lifecycle is the system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementing the candidate system. In system design, we move from the logical to physical aspects of the lifecycle.

The first step is to determine how the output is to be produced and in which format. The input data and master files have to be designed at the next step and finally the impact of the candidate system on the user and organization are documented and evaluated by the management. After identifying the problem and the limitations of the existing system, a detailed design of the proposed system is conducted. Free flow personal interview and reference to previous prepared manually were the only methods taken to collect necessary information. At present, the all organizations are on the path of computerization process.

Design is the phase that indicates the final system. It is the solution, the translation of requirements into ways of meeting them. In this phase the following elements were designed namely dataflow, data stores, processes, procedures. Firstly logical design was done where the outputs, inputs, databases and the procedures were formulated in a manner that meet the project requirements. After logical design physical construction of the system is done.

3.2 Input Design

In the input design, user-oriented inputs are converted into a computer based system format. Online data entry accepts commands and data through a keyboard. The major approach to input design is the menu and the prompt design. In each alternative, the user's options are predefined. The data flow diagram indicates logical data flow, data stores, source and destination. Input data are collected and organized into a group of similar data. Once identified inputs are selected for processing.

In this software, importance is given to develop Graphical User Interface (GUI), which is an important factor in developing efficient and user-friendly software. For inputting user data, attractive forms are designed. User can also select desired options from the menu, which

provides all possible facilities. Also the important input format is designed in such a way that accidental errors are avoided. The user has to input only just the minimum data required, which also helps in avoiding the errors that the users may make. Accurate designing of the input format is very important in developing efficient software. The goal or input design is to make entry as easy, logical and free from errors.

Input Forms are:

The different forms used in the web are:.

- Login Page.
- User registration
- Data information
- Service Adding

3.3 Output Design

In the output design, the emphasis is on producing a hard copy of the information requested or displaying the output on the CRT screen in a predetermined format. Two of the most output media today are printers and the screen. Most users now access their reports from a hard copy or screen display. Computer's output is the most important and direct source of information to the user, efficient, logical, output design should improve the systems relations with the user and help in decision-making.

As the outputs are the most important source of information to the user, better design should improve the system's relation and also should help in decision-making. The output device's capability, print capability, response time requirements etc should also be considered form design elaborates the way output is presented and layout available for capturing information. It's very helpful to produce the clear, accurate and speedy information for end users.

Output Forms are:

- View page.
- View Project
- View details of data

See Appendix A for Screen Shots

3.4 Database Design

In designing a database application you must set up not only the program's routines for maximum performance, but you must pay attention also to the physical layout of the data storage.

A good data base design does the following:

1. Provides minimum search times when locating specific records.
2. Stores the data in the most efficient manner possible to keep the database from growing large.
3. Make data updates as easy as possible.
4. Is flexible enough to allow inclusion of new functions required of program.

Normalization

Data normalization is the process of eliminating redundant data within a database. Taking data normalization fully results in each piece of information in a database appearing only once, although that's always not practical.

Data normalization can handle by creating child tables. A child table is a table in which all the entries share common information that is stored in another table containing the common information is called a parent table, and table containing member's first name is child table. A look up table is another way to store information to prevent data redundancy and to increase the accuracy of data entry functions.

The objectives of database are accuracy and integrity, successful recovery from failure, privacy and security of data and good overall performance database is a collection of interrelated data stored with minimum redundancy to serve many users quick inexpensive and flexible for users. Thus normalization helps one to attain a good database design and thereby ensures efficiency of database.

First Normal Form:

A relation is in first Normal Form (1NF), if and only if all its attributes are based on single domain. The objective of normalizing a table is in to remove its repeating groups and ensure that all entries of the resulting table have at most single value. 1NF is applied to the table named as Login, Notification tables.

Second Normal Form:

A table is said to be in second normal form (2NF), when it is in 1NF and every attribute in the record is functionally dependent upon the whole key, and not just a part of the key.

There are additional normalization levels, such as Third normal form (3NF), Boyce-Codd Normal Form (BCNF), Fourth normal form (4NF), and Fifth normal form (5NF).

Database

A database is similar to a data file in that it is a storage place for data. Like a data file, a database does not present information directly to a user: the user runs an application that accesses data from the database and presents it to the user in an understandable format.

Database systems are more powerful than data files. The data is more highly organized. In a well designed database, there are no duplicate pieces of data that the user or application has to update at the same time. Related pieces of data are grouped together in a single structure or record, and relationships can be defined between these structures and records.

When working with data files, an application must be coded to work with the specific structure of each data file. In contrast, a database contains a catalog that applications use to determine how data is organized. Generic database applications can use the catalog to present users with data from different databases dynamically, without being tied to a specific data format. A database typically has two components: the files holding the physical database and the database management system (DBMS) software that applications use to access data. The DBMS is responsible for enforcing the database structure, including

1) Maintaining the relationships between data in the database

2) Ensuring that data is stored correctly, and that the rules defining data

Relationships are not violated.

3) Recovering all data to point of known consistency in case of system.

Secure Avails system uses a database in second normalized form since all non-key attributes of each table is depend on the key attribute of that table. Database in the web server is managed by SQL Alchemy. The database contains four tables and the details of the tables are specified in the appendix B.

By normalizing the database into 2NF we could remove data redundancy in the tables. Normalization also improved the performance of the application since the complexity of handling a normalized database from the front end is always less. Integrity constraints are implemented in the database by specifying the primary key in tables. Also validations are performed at the front end code.

See Appendix B for Table Design

3.5 Code Design

Index.html

```
<!DOCTYPE html>
<!-- Change the value of lang="en" attribute if your website's language is not English.
You can find the code of your language here -
https://www.w3schools.com/tags/ref_language_codes.asp -->
<html lang="en">

<!-- Mirrored from rhythm.bestlooker.pro/rhythm-original/mp-index-08.html by HTTrack
Website Copier/3.x [XR&CO'2014], Wed, 20 Apr 2022 06:01:33 GMT -->
<head>
    <title>Fire and Rescue Academy,Thrissur</title>
    <meta name="description" content="Rhythm &mdash; One & Multi Page Creative
Template">
    <meta charset="utf-8">
    <meta name="author" content="https://themeforest.net/user/bestlooker/portfolio">
    <!--[if IE]><meta http-equiv='X-UA-Compatible' content='IE=edge,chrome=1'><![endif]-->
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />



    <!-- CSS -->
    <link rel="stylesheet" href="../static/css/bootstrap.min.css">
    <link rel="stylesheet" href="../static/css/style.css">
    <link rel="stylesheet" href="../static/css/style-responsive.css">
    <link rel="stylesheet" href="../static/css/animate.min.css">
    <link rel="stylesheet" href="../static/css/vertical-rhythm.min.css">
    <link rel="stylesheet" href="../static/css/owl.carousel.css">
    <link rel="stylesheet" href="../static/css/magnific-popup.css">
  </head>
  <body class="appear-animate">
```

```
<!-- Navigation panel -->
        <nav class="main-nav dark transparent stick-fixed">
         <div class="full-wrapper relative clearfix">
            <!-- Logo ( * your text or image into link tag *) -->
            <div class="nav-logo-wrap local-scroll">
               <a href="intro.html" class="logo">
                  <img src="../static/images/logo.png" alt="logo" />
               </a>
            </div>
            <div class="mobile-nav" role="button" tabindex="0">
               <i class="fa fa-bars"></i>
               <span class="sr-only">Menu</span>
            </div>

            <!-- Main Menu -->
            <div class="inner-nav desktop-nav">
               <ul class="clearlist">

                  <!-- Item With Sub -->
                  <li>
                     <a href="#" class="mn-has-sub active">Home </a>



                  </li>
                  <li>
                     <a href="#about" class="mn-has-sub active">About Us </a>

                  </li>
                  <!-- End Item With Sub -->

                  <!-- Item With Sub -->
                  <li id="register">
```

```
<li>
                    <a href="/login" class="mn-has-sub active">Login </a>

                </li>
                <li>
                    <a href="/gallery" class="mn-has-sub active">Gallery </a>

                </li>
                <li>
                    <a href="#contact" class="mn-has-sub active">Contact Us </a>

                </li>
                <!-- Item With Sub -->

                <!-- Divider -->
                <li><a> </a></li>
                <!-- End Divider -->

                <!-- Search -->
                <li>
                    <a href="#" class="mn-has-sub"><i class="fa fa-search"></i> Search</a>

                    <ul class="mn-sub">

                        <li>
                            <div class="mn-wrap">
                                <form method="post" class="form">
                                    <div class="search-wrap">
                                        <input type="text" class="form-control search-field"
placeholder="Search...">
                                        <button class="search-button animate" type="submit"
title="Start Search">
```

```
            </ul>
          </div>
          <!-- End Main Menu -->
        </div>
      </nav>
      <!-- End Navigation panel -->

        <main id="main">

          <!-- Fullwidth Slider -->
          <div class="home-section fullwidth-slider-fade bg-dark" id="home">

            <!-- Slide Item -->
            <section class="home-section bg-scroll bg-dark-alfa-50" data-
background="../static/images/hero-bg1.jpg">
              <div class="js-height-full container">

                <!-- Hero Content -->
                <div class="home-content">
                  <div class="home-text">

                    <h1 class="hs-line-8 no-transp font-alt mb-50 mb-xs-30">

                                                                    Suspect and
Inspect,BUT DO NOT NEGLECT
                    </h1>

                    <h2 class="hs-line-14 font-alt mb-50 mb-xs-30">
                      Fire and Rescue  Academy
                    </h2>
```

```
<!-- Slide Item -->
            <section class="home-section bg-scroll bg-dark-alfa-30" data-
background="../static/images/hero-bg2.jpg">
              <div class="js-height-full container">

                <!-- Hero Content -->
                <div class="home-content">
                  <div class="home-text">

                    <h1 class="hs-line-8 no-transp font-alt mb-50 mb-xs-30">

                    </h1>

                    <h2 class="hs-line-14 font-alt mb-50 mb-xs-30">

                    </h2>
                    <div class="local-scroll">
                      <a href="#about" class="btn btn-mod btn-border-w btn-medium btn-
round d-none d-sm-inline-block">know About Us</a>
                      <span class="d-none d-sm-inline-block"> </span>
                      <a href="/login" class="btn btn-mod btn-border-w btn-medium btn-
round lightbox mfp-iframe">Login</a>
                    </div>

                  </div>
                </div>
                <!-- End Hero Content -->

              </div>
            </section>
            <!-- End Slide Item -->
```

```
</div>
    <!-- End Page Wrap -->
    <!-- JS -->
    <script type="text/javascript" src="../static/js/jquery-3.5.1.min.js"></script>
    <script type="text/javascript" src="../static/js/jquery.easing.1.3.js"></script>
    <script type="text/javascript" src="../static/js/bootstrap.bundle.min.js"></script>
    <script type="text/javascript" src="../static/js/SmoothScroll.js"></script>
    <script type="text/javascript" src="../static/js/jquery.scrollTo.min.js"></script>
    <script type="text/javascript" src="../static/js/jquery.localScroll.min.js"></script>
    <script type="text/javascript" src="../static/js/jquery.viewport.mini.js"></script>
    <script type="text/javascript" src="../static/js/jquery.countTo.js"></script>
    <script type="text/javascript" src="../static/js/jquery.appear.js"></script>
    <script type="text/javascript" src="../static/js/jquery.sticky.js"></script>
    <script type="text/javascript" src="../static/js/jquery.parallax-1.1.3.js"></script>
    <script type="text/javascript" src="../static/js/jquery.fitvids.js"></script>
    <script type="text/javascript" src="../static/js/owl.carousel.min.js"></script>
    <script type="text/javascript" src="../static/js/isotope.pkgd.min.js"></script>
    <script type="text/javascript" src="../static/js/imagesloaded.pkgd.min.js"></script>
    <script type="text/javascript" src="../static/js/jquery.magnific-popup.min.js"></script>
    <script type="text/javascript" src="../static/js/wow.min.js"></script>
    <script type="text/javascript" src="../static/js/masonry.pkgd.min.js"></script>
    <script type="text/javascript" src="../static/js/morphext.js"></script>
    <script type="text/javascript" src="../static/js/jquery.lazyload.min.js"></script>
    <script type="text/javascript" src="../static/js/all.js"></script>
    <script type="text/javascript" src="../static/js/contact-form.js"></script>
    <script type="text/javascript" src="../static/js/jquery.ajaxchimp.min.js"></script>
    <!--[if lt IE 10]><script type="text/javascript" src="js/placeholder.js"></script><![endif]-->


  </body>
</html>
```

Login.html

```
<!DOCTYPE html>
<!-- Change the value of lang="en" attribute if your website's language is not English.
You can find the code of your language here -
https://www.w3schools.com/tags/ref_language_codes.asp -->
<html lang="en">

<!-- Mirrored from rhythm.bestlooker.pro/rhythm-original/mp-index-08.html by
HTTrack Website Copier/3.x [XR&CO'2014], Wed, 20 Apr 2022 06:01:33 GMT -->
<head>
    <title>Fire and Rescue Academy,Thrissur</title>
    <meta name="description" content="Rhythm &mdash; One & Multi Page
Creative Template">
    <meta charset="utf-8">
    <meta name="author"
content="https://themeforest.net/user/bestlooker/portfolio">
    <!--[if IE]><meta http-equiv='X-UA-Compatible' content='IE=edge,chrome=1'><![endif]-->
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />



    <!-- CSS -->
    <link rel="stylesheet" href="../static/css/bootstrap.min.css">
    <link rel="stylesheet" href="../static/css/style.css">
    <link rel="stylesheet" href="../static/css/style-responsive.css">
    <link rel="stylesheet" href="../static/css/animate.min.css">
    <link rel="stylesheet" href="../static/css/vertical-rhythm.min.css">
    <link rel="stylesheet" href="../static/css/owl.carousel.css">
    <link rel="stylesheet" href="../static/css/magnific-popup.css">
</head>
<main id="main">

        <!-- Head Section -->
        <section class="small-section bg-dark-lighter">
          <div class="relative container align-left">

            <div class="row">

              <div class="col-md-8">
                <h1 class="hs-line-11 font-alt mb-20 mb-xs-0">My Account</h1>

              </div>

              <div class="col-md-4 mt-30">
                <div class="mod-breadcrumbs font-alt align-right">
                  <a href="/">Home</a> / <a
href="#">Login</a> / <span>My Account</span>
                </div>

              </div>
            </div>
```

```
<!-- Section -->
        <section class="page-section">
          <div class="container relative">

            <!-- Nav Tabs -->
            <div class="align-center mb-40 mb-xxs-30">
              <ul role="tablist" class="nav nav-tabs tpl-minimal-tabs">

                <li>
                  <a href="#mini-one" aria-controls="mini-one" class="nav-link active" data-bs-toggle="tab" role="tab" aria-selected="true">Login</a>
                </li>
              </ul>
            </div>
            <!-- End Nav Tabs -->

            <!-- Tab panes -->
            <div class="tab-content section-text">

              <div role="tabpanel" class="tab-pane fade show active" id="mini-one">

                <!-- Login Form -->
                <div class="row">
                  <div class="col-md-4 offset-md-4">

                    <form class="form contact-form" method="post">
                      <div class="clearfix">

                        <!-- Username -->
                        <div class="form-group">
                          <input type="text" name="email"  class="input-md round form-control" placeholder="Email"  required >
```

```
    </div>
    <!-- End Page Wrap -->
    <!-- JS -->
    <script type="text/javascript" src="../static/js/jquery-3.5.1.min.js"></script>
    <script type="text/javascript" src="../static/js/jquery.easing.1.3.js"></script>
    <script type="text/javascript" src="../static/js/bootstrap.bundle.min.js"></script>
    <script type="text/javascript" src="../static/js/SmoothScroll.js"></script>
    <script type="text/javascript" src="../static/js/jquery.scrollTo.min.js"></script>
    <script type="text/javascript" src="../static/js/jquery.localScroll.min.js"></script>
    <script type="text/javascript" src="../static/js/jquery.viewport.mini.js"></script>
    <script type="text/javascript" src="../static/js/jquery.countTo.js"></script>
    <script type="text/javascript" src="../static/js/jquery.appear.js"></script>
    <script type="text/javascript" src="../static/js/jquery.sticky.js"></script>
    <script type="text/javascript" src="../static/js/jquery.parallax-1.1.3.js"></script>
    <script type="text/javascript" src="../static/js/jquery.fitvids.js"></script>
    <script type="text/javascript" src="../static/js/owl.carousel.min.js"></script>
    <script type="text/javascript" src="../static/js/isotope.pkgd.min.js"></script>
    <script type="text/javascript" src="../static/js/imagesloaded.pkgd.min.js"></script>
    <script type="text/javascript" src="../static/js/jquery.magnific-popup.min.js"></script>
    <script type="text/javascript" src="../static/js/wow.min.js"></script>
    <script type="text/javascript" src="../static/js/masonry.pkgd.min.js"></script>
    <script type="text/javascript" src="../static/js/morphext.js"></script>
    <script type="text/javascript" src="../static/js/jquery.lazyload.min.js"></script>
    <script type="text/javascript" src="../static/js/all.js"></script>
    <script type="text/javascript" src="../static/js/contact-form.js"></script>
    <script type="text/javascript" src="../static/js/jquery.ajaxchimp.min.js"></script>
    <!--[if lt IE 10]><script type="text/javascript" src="js/placeholder.js"></script><![endif]-->
</body>
</html>
```

Staffindex.html

```
{% extends 'stulayout.html'%}
{% block content %}


        <main id="main">

            <!-- Home Section -->
            <section class="home-section bg-dark" data-
background="../static/video/video.jpg" id="home">
              <div class="js-height-full container">

                <!-- Video BG Init -->
                <!-- Please, replace three video files in folder "video" with your own
ones -->
                <div class="bg-video-wrapper">
                  <video class="bg-video" preload="auto" autoplay loop muted
playsinline data-object-fit="cover">
                    <source src="../static/video/video.mp4" type="video/mp4">

                  </video>
                </div>

                <!-- Hero Content -->
                <div class="home-content">
                  <div class="home-text">

                    <h1 class="hs-line-8 no-transp font-alt mb-50 mb-xs-30">
                      Fire And Rescue Academy
                    </h1>

                    <h2 class="hs-line-14 font-alt mb-50 mb-xs-30">
                       Welcome
                    </h2>
                    <div class="local-scroll">

                      <a href="/view_course_stu" class="btn btn-mod btn-border-w
btn-medium btn-round d-none d-sm-inline-block">View Courses</a>

                    </div>
                  </div>
                </div>
                <!-- End Hero Content -->
```

```
<!-- Scroll Down -->
                <div class="local-scroll">
                    <a href="#about" class="scroll-down"><i class="fa fa-angle-down scroll-down-icon"></i><span class="sr-only">Scroll to the next section</span></a>
                </div>
                <!-- End Scroll Down -->

            </div>
        </section>
        <!-- End Home Section -->

        <!-- About Section -->
        <section class="page-section" id="about">
            <div class="container relative">

                <h2 class="section-title font-alt align-left mb-70 mb-sm-40">
                    About Farsa

                    <a href="#" class="section-more right">More about us <i class="fa fa-angle-right"></i></a>

                </h2>

                <div class="section-text">
                    <div class="row">


                        <div class="col-md-4 col-sm-6 mb-sm-50 mb-xs-30">
                            The Kerala Fire and Rescue Services is the service department of the Government of Kerala whose function is to fight fires and provide relief measures in times of calamities and disasters in Kerala.
```

```
</li>
<div class="col-md-4 col-sm-6 mb-sm-50 mb-xs-30">
                    The Kerala Fire Force department started working as a separate department since 1963. Director of Civil Defence held the position of the Head of the Department till 1967. From 1967 to August 1970 an Inspector General of Police held the charge of the Director for Fire Force.  </div>

                                                    <div class="col-md-4 col-sm-6 mb-sm-50 mb-xs-30">
                    Considering the rescue works under taken by this Department and significance in that area, this department had been renamed as 'Kerala Fire & Rescue Services' in 2002.  </div>

                </div>
            </div>

        </div>
    </section>
    <!-- End About Section -->

    <!-- Divider -->
    <hr class="mt-0 mb-0 "/>
    <!-- End Divider -->

    <!-- Services Section -->
    <section class="page-section" id="services">
        <div class="container relative">

            <h2 class="section-title font-alt mb-70 mb-sm-40">
                Training facilities
            </h2>

            <!-- Nav tabs -->
            <ul role="tablist" class="nav nav-tabs tpl-alt-tabs font-alt pt-30 pt-sm-0 pb-30 pb-sm-0">
```

```
<li>
                    <a href="#service-graphic" class="nav-link" data-bs-toggle="tab"
role="tab" aria-selected="false">


                      <div class="alt-tabs-icon">
                        <span class="icon-layers"></span>
                      </div>


                      3. Accomodation Facility
                    </a>
                  </li>
                  <li>
                    <a href="#service-development" class="nav-link" data-bs-toggle="tab"
role="tab" aria-selected="false">


                      <div class="alt-tabs-icon">
                        <span class="icon-profile-male"></span>
                      </div>


                      4. Existing Staff Strength
                    </a>
                  </li>
                  <li>
                    <a href="#service-photography" class="nav-link" data-bs-toggle="tab"
role="tab" aria-selected="false">


                      <div class="alt-tabs-icon">
                        <span class="icon-notebook"></span>
                      </div>


                      5.Courses
                    </a>
                  </li>
```

By normalizing the database into 2NF we could remove data redundancy in the tables. Normalization also improved the performance of the application since the complexity of handling a normalized database from the front end is always less. Integrity constraints are implemented in the database by specifying the primary key in tables. Also validations a

```
<!-- Service Item -->
        <div role="tabpanel" class="tab-pane fade" id="service-web-design">

          <div class="section-text">
            <div class="row">
              <div class="col-md-4 mb-md-40 mb-xs-30">
                <blockquote class="mb-0">
                  <p>
                    Class Room facility
                  </p>

                </blockquote>
              </div>
              <div class="col-md-4 col-sm-6 mb-sm-50 mb-xs-30">
                Class room facility for the trainees are provided in the first and second floor.
                                                                                Four class rooms with a capacity of 100 each and two other class rooms with 50 each is available in this academy.
                                                                                One classroom is equiped with LCD projector.
              </div>

            </div>
          </div>

        </div>
        <!-- End Service Item -->

        <!-- Service Item -->
        <div role="tabpanel" class="tab-pane fade" id="service-graphic">
```

```
                                    <a
href="mailto:support@bestlooker.pro">adotsr.frs@kerala.gov.in</a>
                                </div>
                            </div>
                        </div>
                        <!-- End Email -->

                    </div>
                </div>

            </div>

        </div>
    </section>
    <!-- End Contact Section -->




</main>



{%endblock%}
```

SYSTEM TESTING

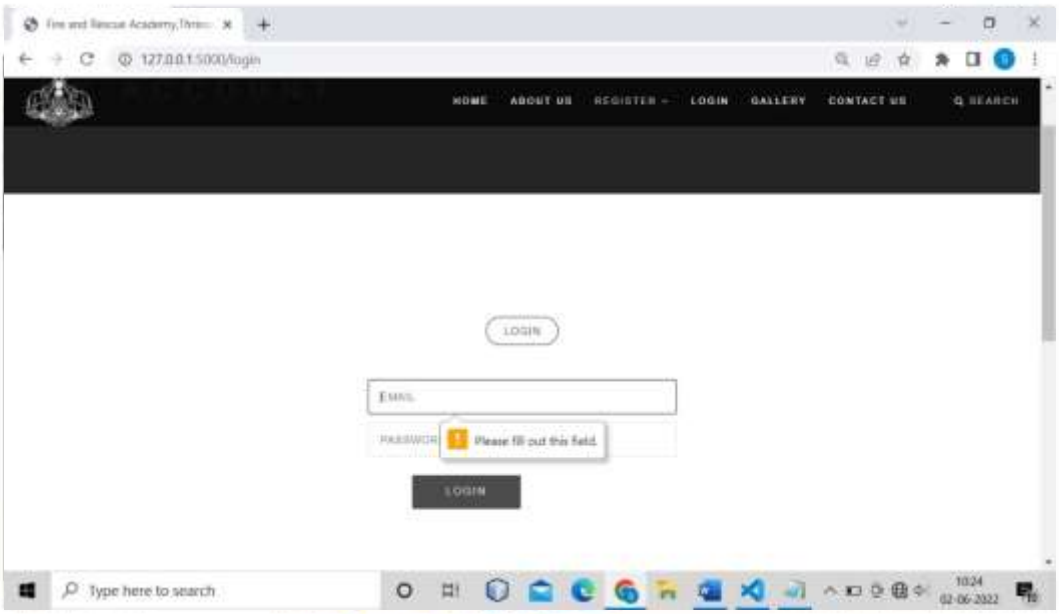
4. SYSTEM TESTING

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding an as yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is the vital to the success of the system. The entire testing process can be divided into different phases:

- Unit testing
- Integration testing
- Validation testing
- System testing

4.1 Unit Testing

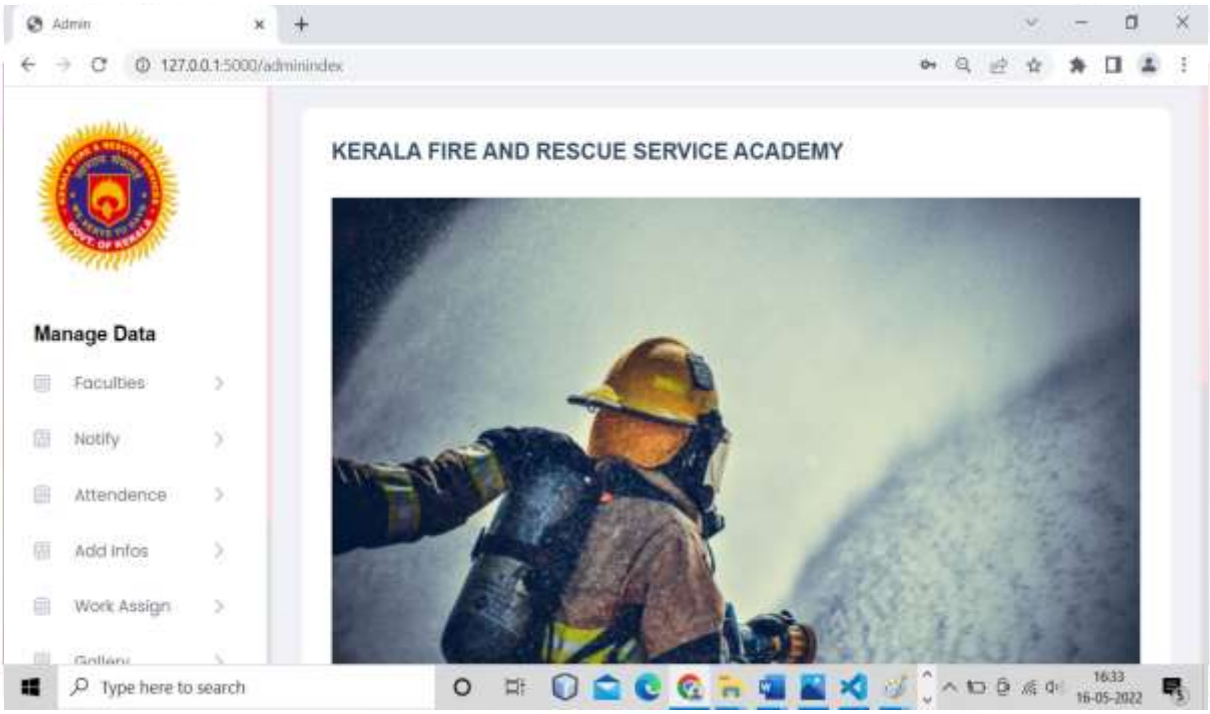
It focuses efforts on the smallest unit of software design. This is known as module testing. Each unit is tested separately for errors and defects found are corrected. The modules are tested separately. In database management, the data and design of tables are tested separately for errors. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output for the module.



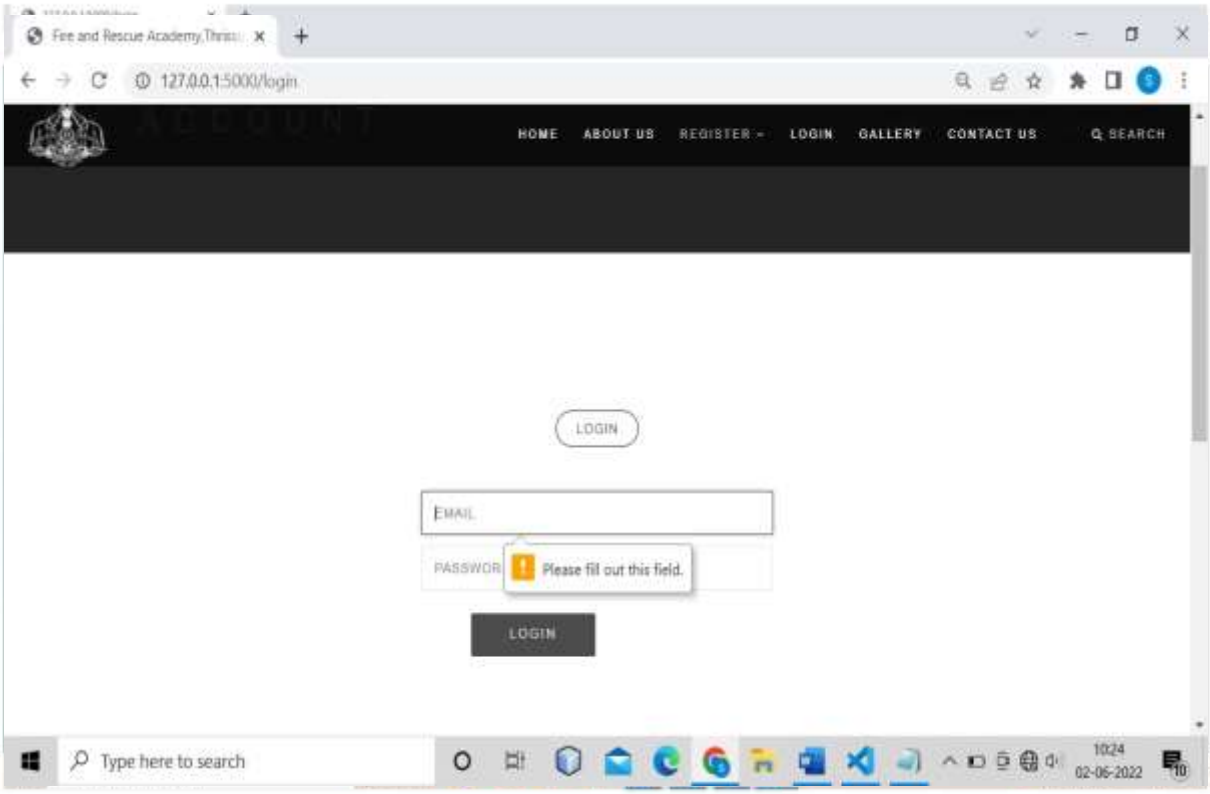
4.2 Integration Testing

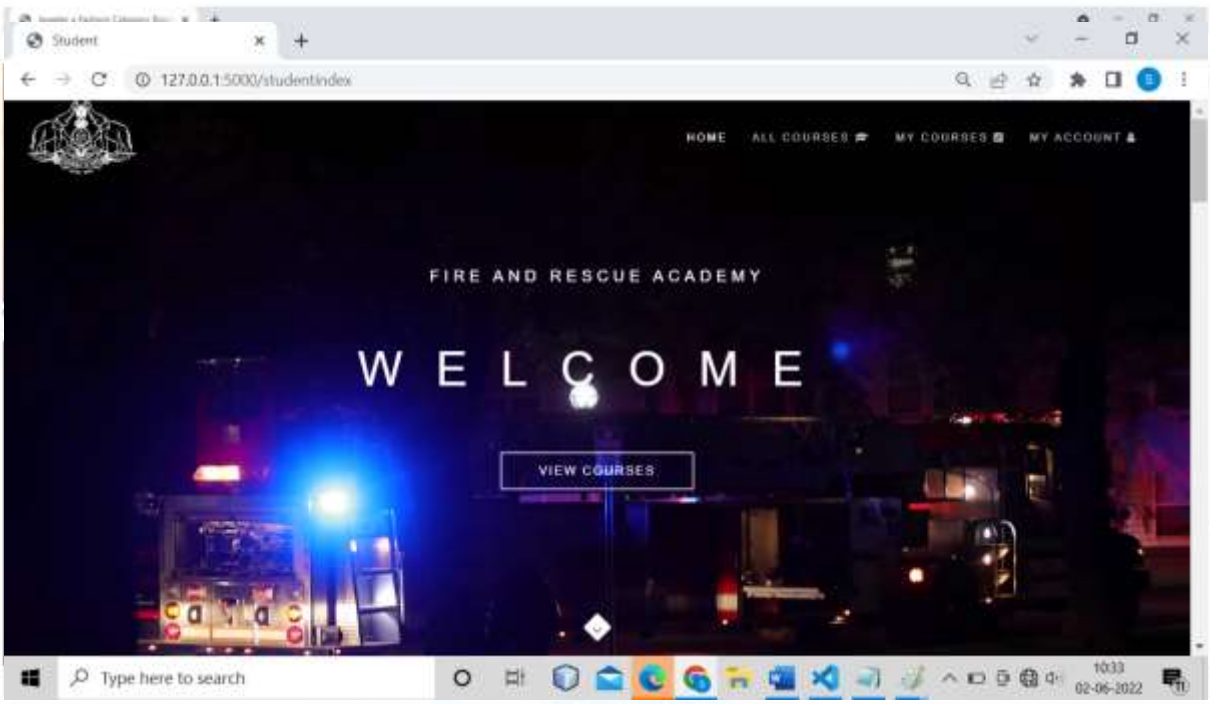
Integration testing is a systematic approach for constructing the program structure, while at the same time constructing test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

Admin

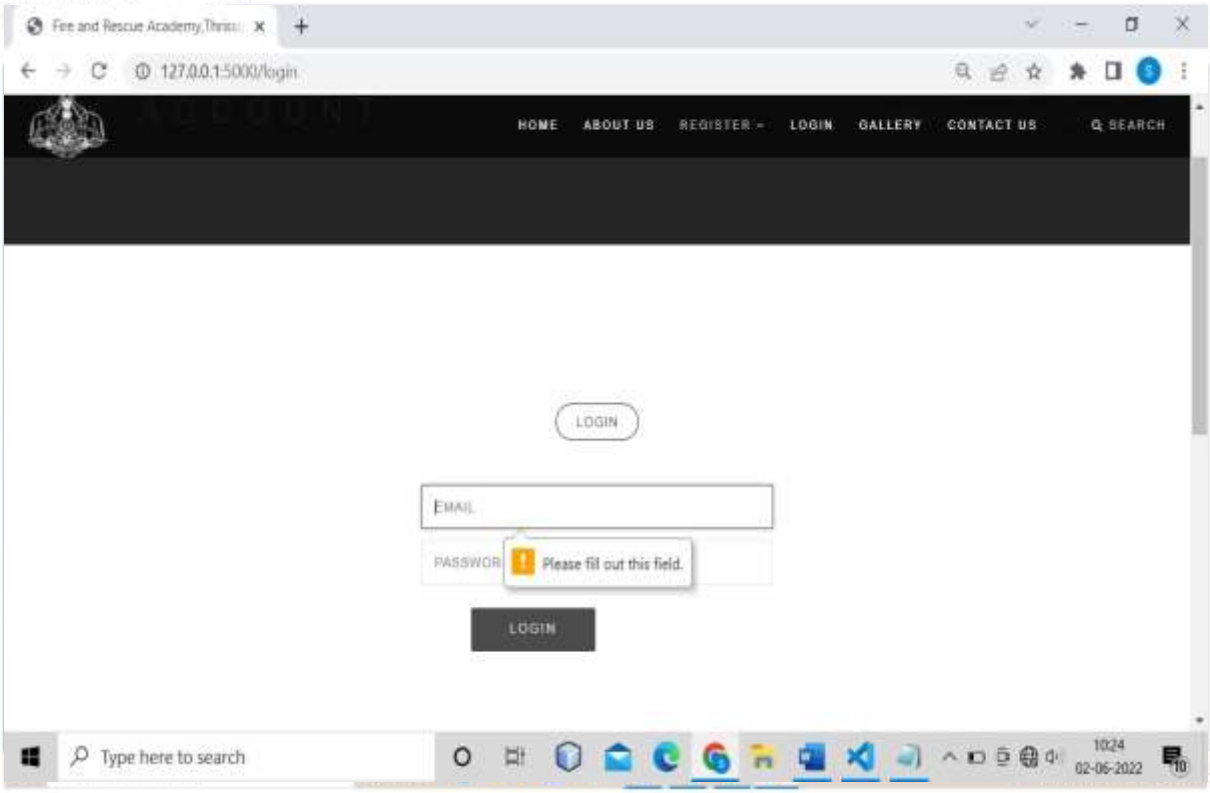


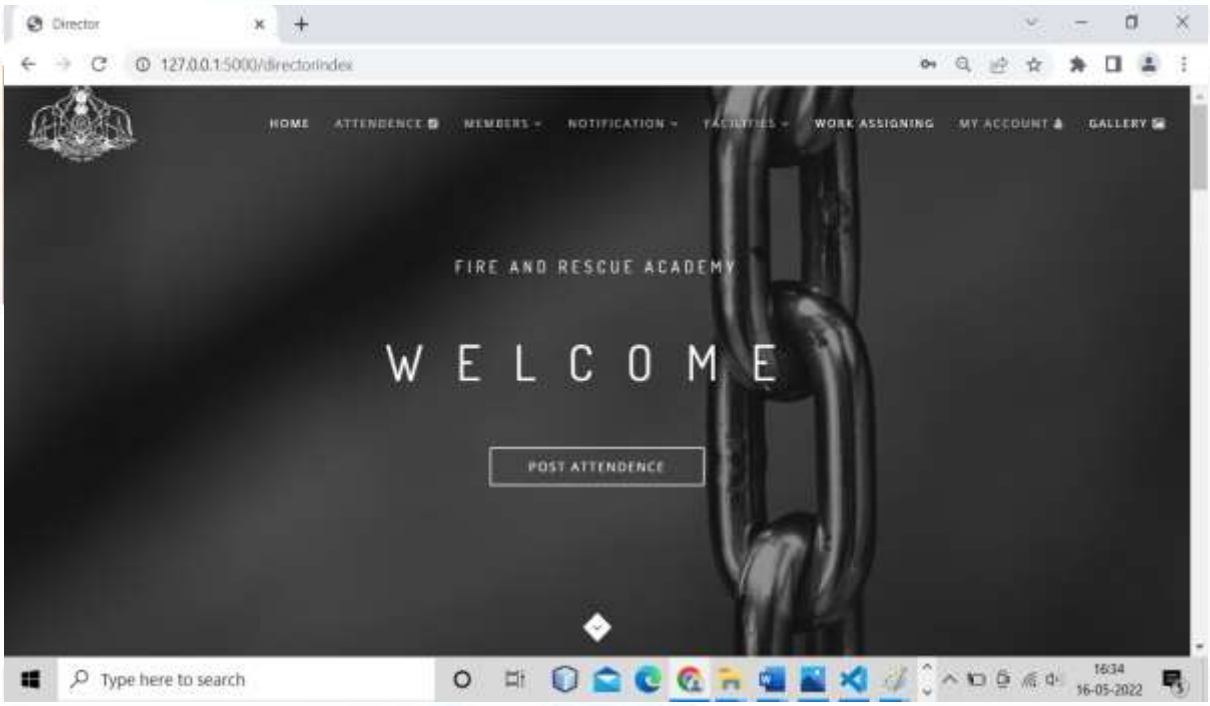
User


Fire and Rescue Academy,Thriss
127.0.0.1:5000/login
HOME
ABOUT US
REGISTER
LOGIN
GALLERY
CONTACT US
SEARCH
LOGIN
EMAIL
PASSWOR
Please fill out this field.
LOGIN
Type here to search
10:24
02-06-2022


Student
127.0.0.1:5000/studentindex
HOME
ALL COURSES
MY COURSES
MY ACCOUNT
FIRE AND RESCUE ACADEMY
WELCOME
VIEW COURSES
Type here to search
10:33
02-06-2022

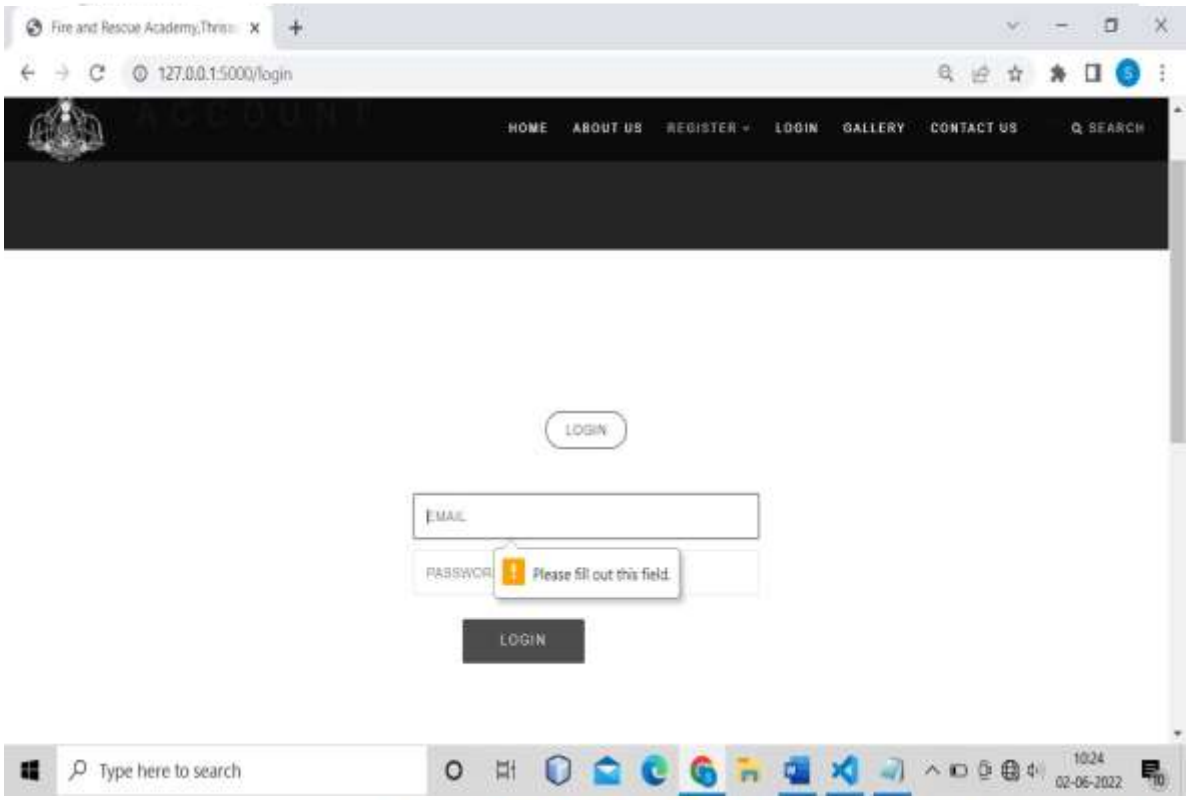
Director


Fire and Rescue Academy,Thriss
127.0.0.1:5000/login
HOME
ABOUT US
REGISTER
LOGIN
GALLERY
CONTACT US
SEARCH
LOGIN
EMAIL
PASSWOR
Please fill out this field.
LOGIN
Type here to search
1024
02-06-2022


Director
127.0.0.1:5000/directorindex
HOME
ATTENDENCE
MEMBERS
NOTIFICATION
FACILITIES
WORK ASSIGNING
MY ACCOUNT
GALLERY
FIRE AND RESCUE ACADEMY
WELCOME
POST ATTENDENCE
Type here to search
16:34
16-05-2022

4.3 Validation Testing

Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that can reasonable expected by the client. That means the input we are entering to the system will be checked whether the entered input is valid or not.



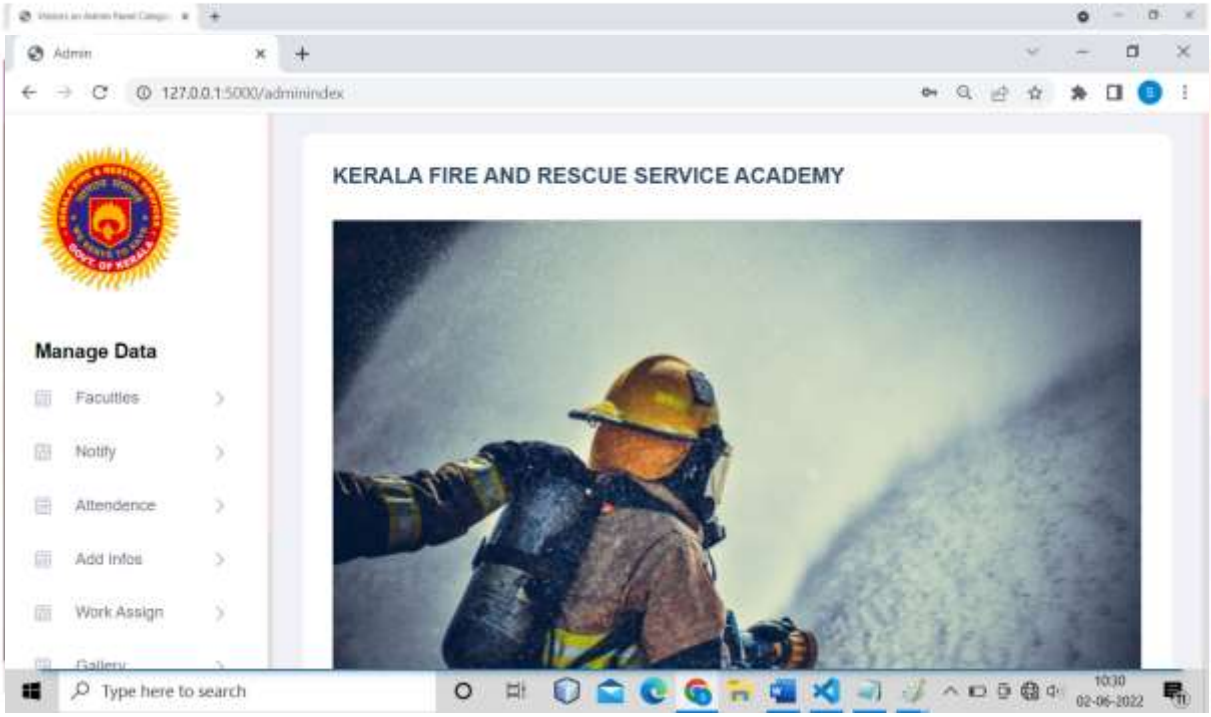
4.4 System Testing

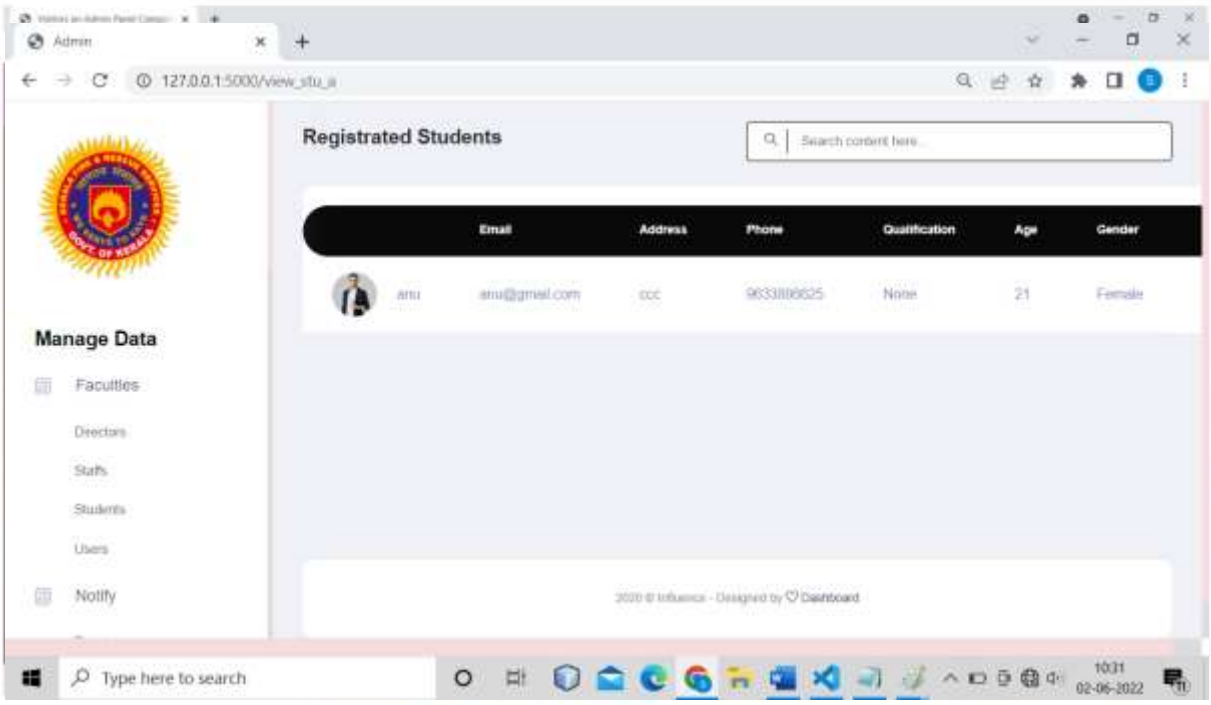
After performing validation testing, the next step is system testing of the proposed system could be useful if it does not produce the required output in the specific format. All the reports which are generating will be tested. The system under consideration is tested by constantly keeping in touch with the system uses at the time of developing and making changes whenever required. This is done with regard to the following points.

- Input screen design
- Output screen design
- Menu Driven System

- Format reports

All the above tests are performed in FARSA and found successful.





SYSTEM IMPLEMENTATION AND MAINTENANCE

5. SYSTEM IMPLEMENTATION AND MAINTENANCE

5.1 System Implementation

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which is operated in a very different manner from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

Implementation Methods

There are several methods for handling the implementation and the consequent conversion from the old to the new computerized system. The most secure method for conversion from the old system to the new system is to run the old and new system in parallel. In this approach, a person may operate in the manual older processing system as well as start operating the new computerized system. This method offers high security, because even if there is a flow in the computerized system, we can depend upon the manual system. However, the cost for maintaining two systems in parallel is very high. This outweighs its benefits.

Another commonly method is a direct cut over from the existing manual system to the computerized system. The change may be within a week or within a day. There are no parallel activities. However, there is no remedy in case of a problem. This strategy requires careful planning.

Implementation Plan

The implementation plan includes a description of all the activities that must occur to implement the new system and to put into operation. It identifies the personnel responsible for the activities and prepares a time chart for implementing the system. The implementation plan consists of the following steps:

- List all files required for implementation
- Identify all data required to build new files during the implementation.
- List all new documents and procedures that go into the new system.

The project is implemented and hence found that it satisfies all the objectives of the system.

5.2 System Maintenance

Software maintenance is the process of modifying a software system or component after its delivery in order to correct faults, improve the performance and the other attributes, or to adapt to the changed environment. Maintenance covers a wide range of activities including correcting the coding and design errors, updating the documentation and test data, and upgrading the user support. There is an aging process that calls for periodic maintenance of hardware and software. Maintenance is always necessary to keep the software usable and useful. Hardware also requires periodic maintenance activities can be classified into

- Corrective Maintenance
- Perceptive Maintenance
- Adaptive Maintenance

Corrective maintenance removes software faults. Perceptive maintenance improves the system without changing its functionality. The objective of perceptive maintenance should be to prevent failures and optimize the software. Adaptive maintenance modifies the software to keep it up-to-date with its operative environment.

We here using Perceptive maintenance. We prefer to use this maintenance method. Because the functionality of the software is constant. If any changes or updations needed it maintained through analysing of the software. Will not be used to change the entire software, only the specified area is focusing. If there is any further updation in the software, they will contact directly and will provide all the maintenance tasks.

CONCLUSION

CONCLUSION

From this project I gained a lot of information about how software development is used to complete projects. The project exercise was mainly to enable me acquire practical skills I have been able to acquire practical skills besides the knowledge that was learned from completing my assigned tasks I gained a lot from talking to my boss and fellow workers. They gave a lot of good advice that I will take with me as I prepare to enter the workforce after graduation. I had a great semester

Sandhya S

SCOPE FOR FUTURE ENHANCEMENT

7. FUTURE ENHANCEMENTS

FARSA project, it describe the structure and working of fire and rescue service academy. In future, there should be include some advanced technology to make every service time efficient and quick responsive. For that I preferred to add Machine learning and AI elements to this site. When receive call from user or other stations automatically track their location and record the call. By the call record , using sound recognition notify the officer that some service requirement is arrived.

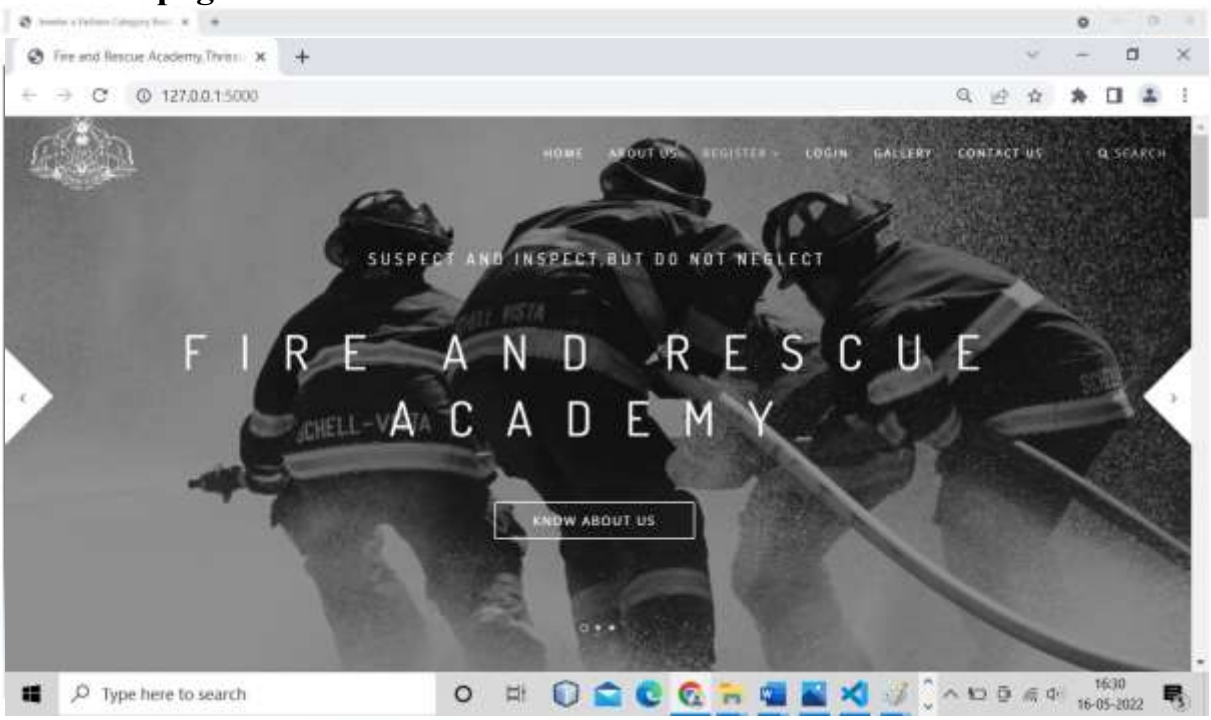
Also inform about the emergency service to the nearest fire stations. Sort monthly attendance of staffs and director and also inform about their absence as text message to their registered mobile number. All students should get every updation in their course as messages. Provide online training and classes. User can get all information of their nearest station in message to the mobile (text message).

APPENDIX

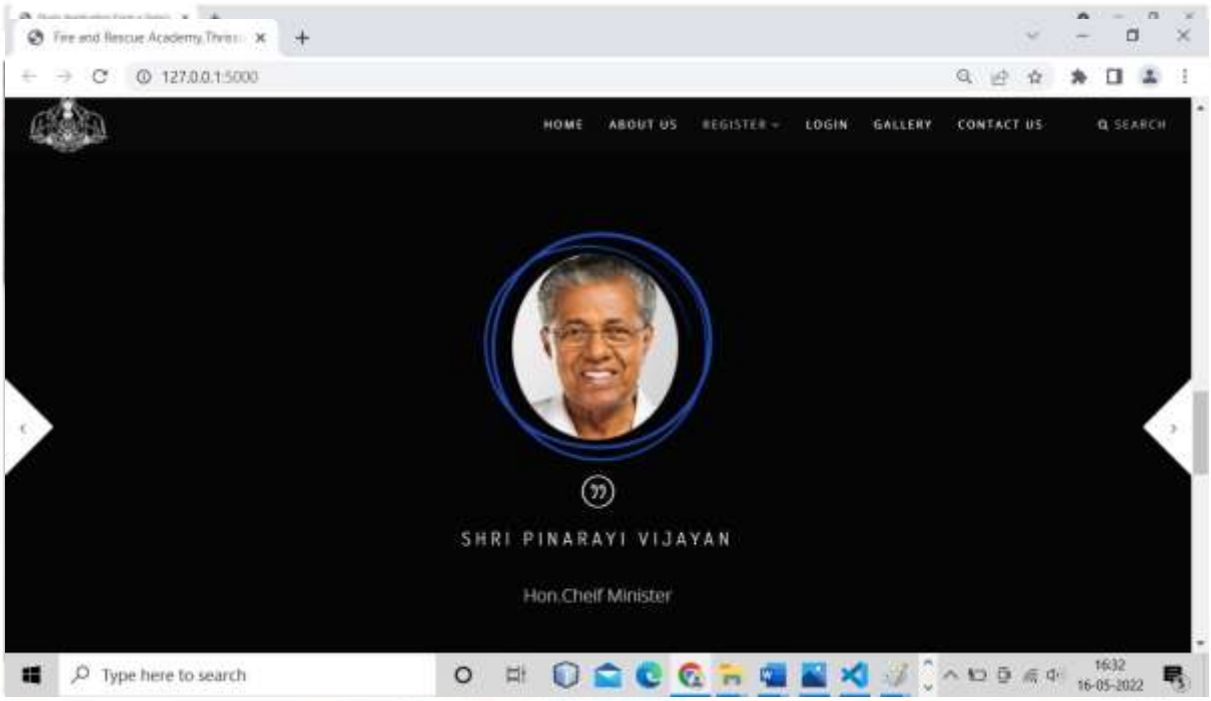
8. APPENDIX

8.1 Screen Shots

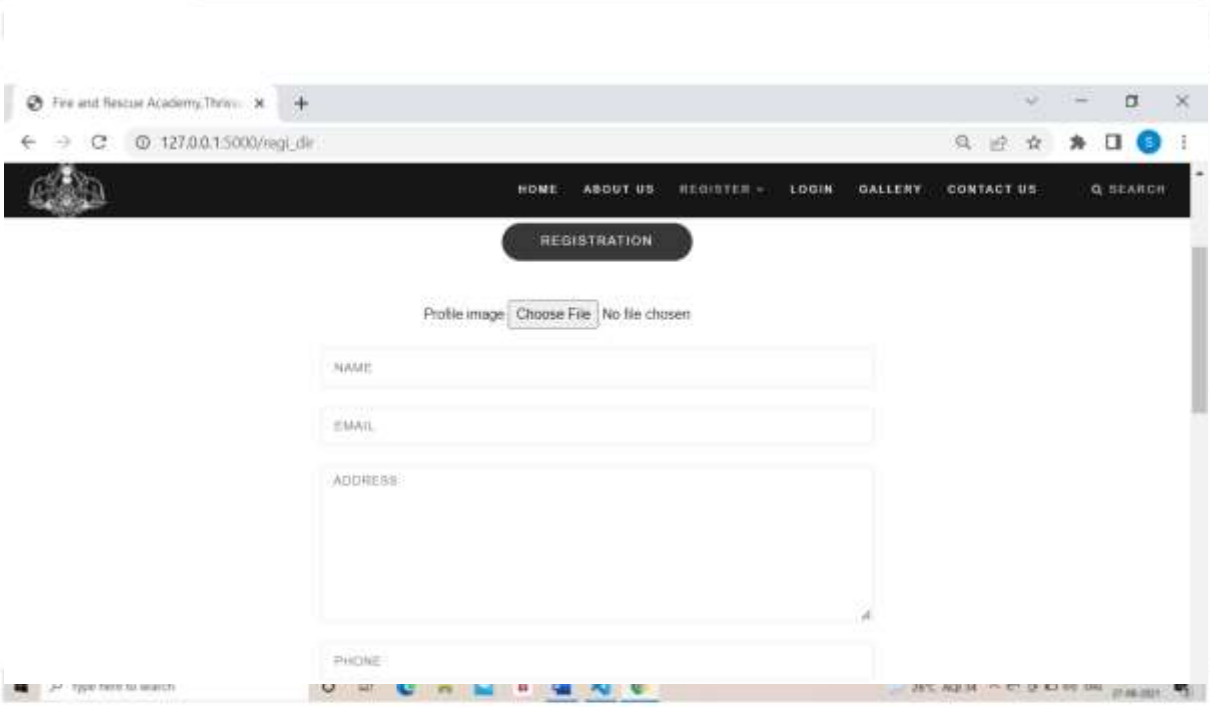
1. Home page



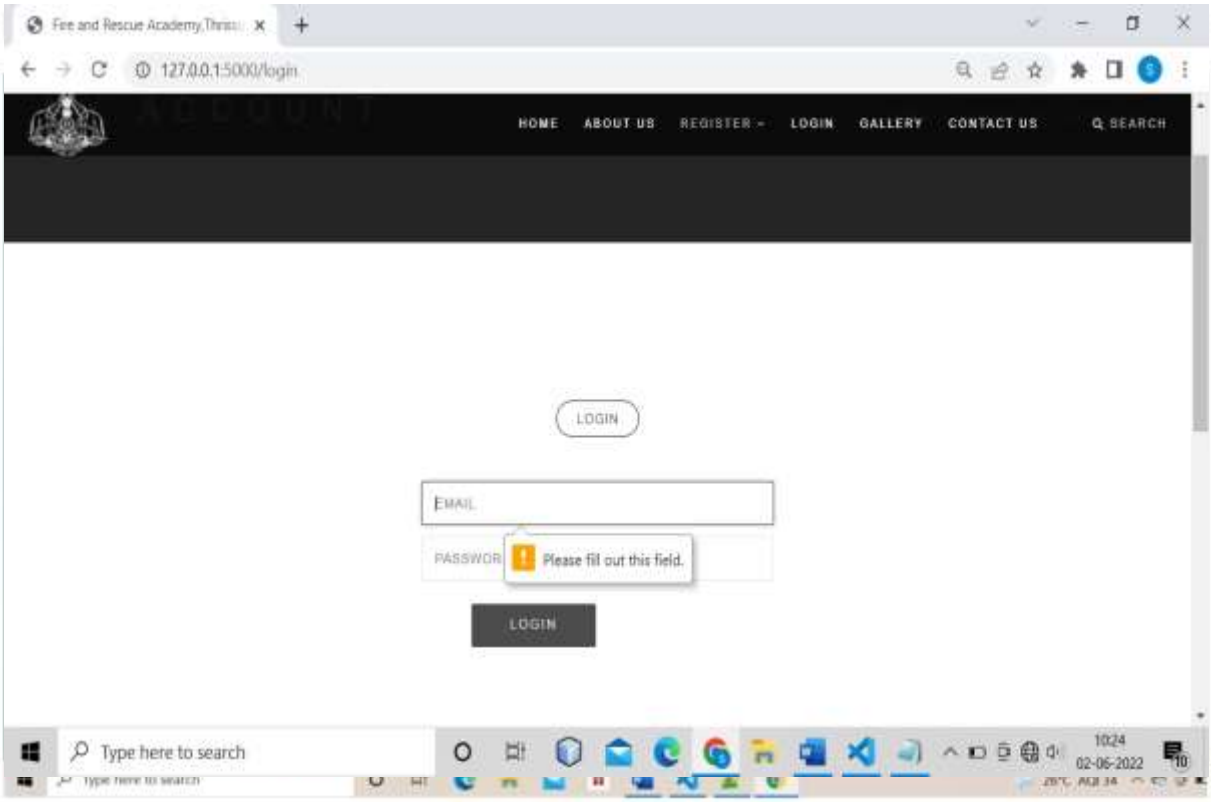
2. About Us



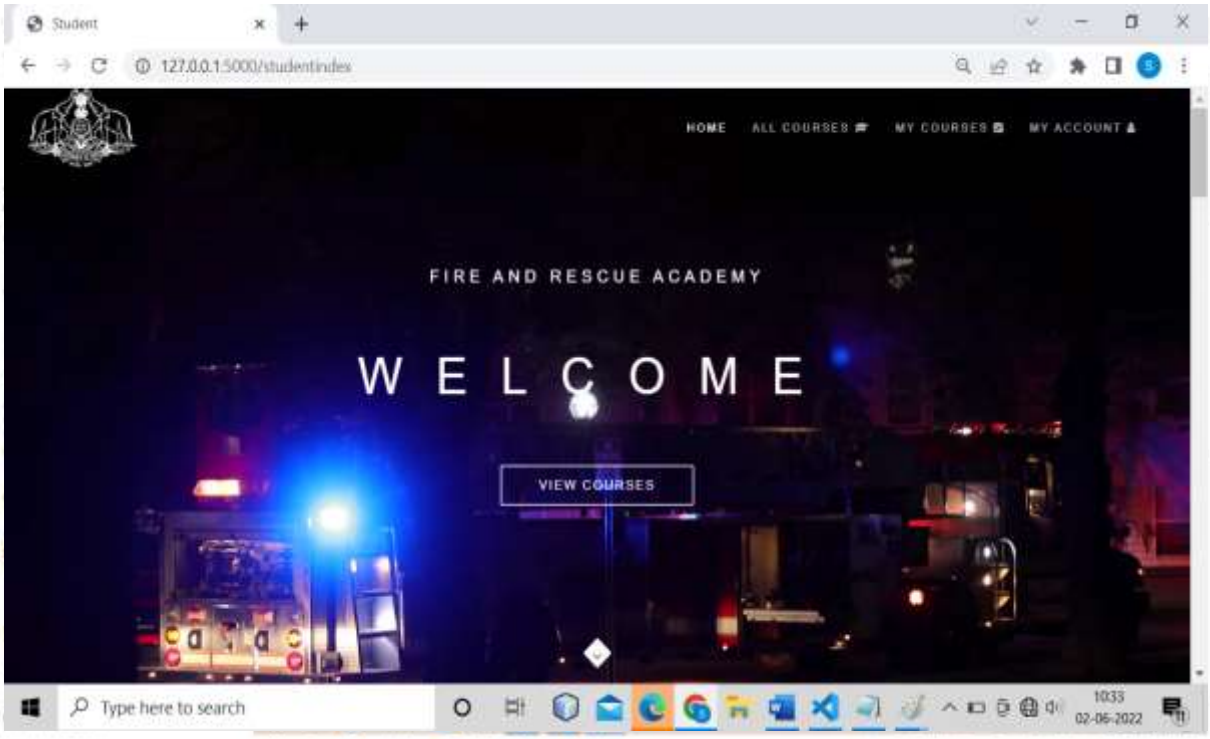
3 Registration



4.Login



5.User


Student
127.0.0.1:5000/studentindex
HOME
ALL COURSES
MY COURSES
MY ACCOUNT
FIRE AND RESCUE ACADEMY
WELCOME
VIEW COURSES
Type here to search
1033
02-06-2022

8.2 Database Design

1 Login

Field Name	Datatype(size)	Description
Id	Integer	Primary key
Image	String (20)	Nullable=false
Name	String (30)	
Email	String (30)	
Address	String (50)	
Phone	integer	
Qualification	String (30)	
Experience	String (30)	
Age	integer	
Designation	String (30)	
Category	String (30)	
Gender	String (30)	
Password	String (30)	
User type	String (30)	
Status	String (30)	

2 Notification

Field Name	Datatype(size)	Description
Id	Integer	Primary key
Dir id	Integer	
Image	String (30)	Nullable=false
Name	String (30)	
Email	String (50)	
Date	String (30)	
Message	String (30)	
User type	String (30)	

3 Gallery

Field Name	Datatype(size)	Description
Id	Integer	Primary key
Image1	String (30)	Nullable=false
Image2	String (30)	Nullable=false
Image3	String (30)	Nullable=false
Date	String (20)	

4 Vehicle

Field Name	Datatype(size)	Description
Id	Integer	Primary key
Image	String (30)	Nullable=false
Station Name	String (30)	
Location	String (50)	
Category	String (30)	
number	String (30)	

5 Station

Field Name	Datatype(size)	Description
Id	Integer	Primary key
Image	String (30)	Nullable=false
Station Name	String (30)	
Incharge	String (50)	
Location	String (30)	
Emergency	String (30)	

6 Call

Field Name	Datatype(size)	Description
Id	Integer	Primary key
Time	String (30)	Nullable=false
Date	String (30)	
Call from	String (50)	
Service	String (30)	
Receiver	String (30)	

7 Course

Field Name	Datatype(size)	Description
Id	Integer	Primary key
Image	String (30)	Nullable=false
Course Name	String (30)	
Purpose	String (50)	
Duration	String (30)	
Fee	String (30)	
Seat	String (30)	
Affiliated	String (30)	
Description	String (30)	

8 Work

Field Name	Datatype(size)	Description
Id	Integer	Primary key
Name	String (30)	
Category	String (30)	
Sta id	String (50)	
Work	String (30)	
Date	String (30)	
Time	String (30)	
Spot	String (30)	

9 Feedback

Field Name	Datatype(size)	Description
Id	Integer	Primary key
Name	String (30)	
Date	String (30)	
Time	String (50)	
Email	String (30)	
Feedback	String (30)	
User type	String (30)	

10 Attendance

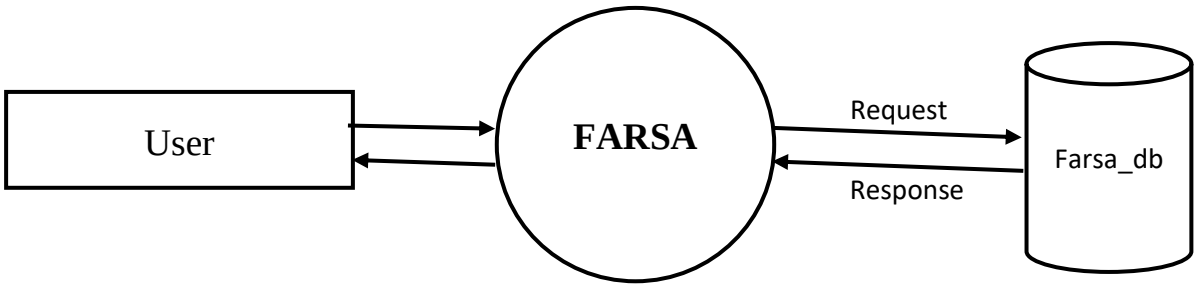
Field Name	Datatype(size)	Description
Id	Integer	Primary key
Name	String (30)	
Email	String (30)	
Att id	String (50)	
Time	String (30)	
User type	String (30)	
Status	String (30)	

11 Apply

Field Name	Datatype(size)	Description
Id	Integer	Primary key
Name	String (30)	
Email	String (30)	
Address	String (50)	
Phone	integer	
Qualification	String (30)	
Age	integer	
Gender	String (30)	
Current course	String (30)	
Backlog	String (30)	
Career gap	String (30)	
Experience	String (30)	
Course id	String (30)	
Stu id	String (30)	
Date	String (30)	

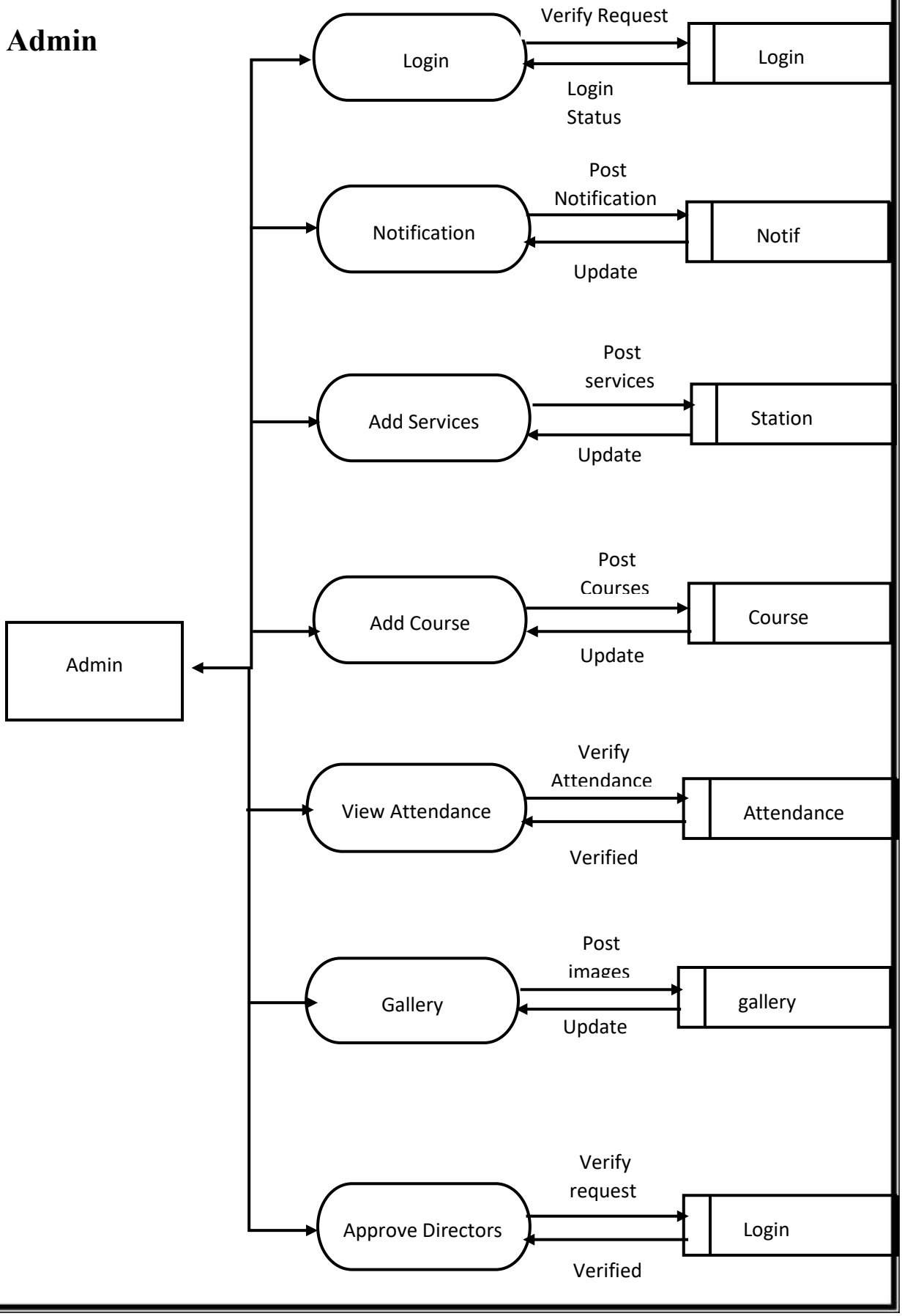
8.3 Data Flow Diagram

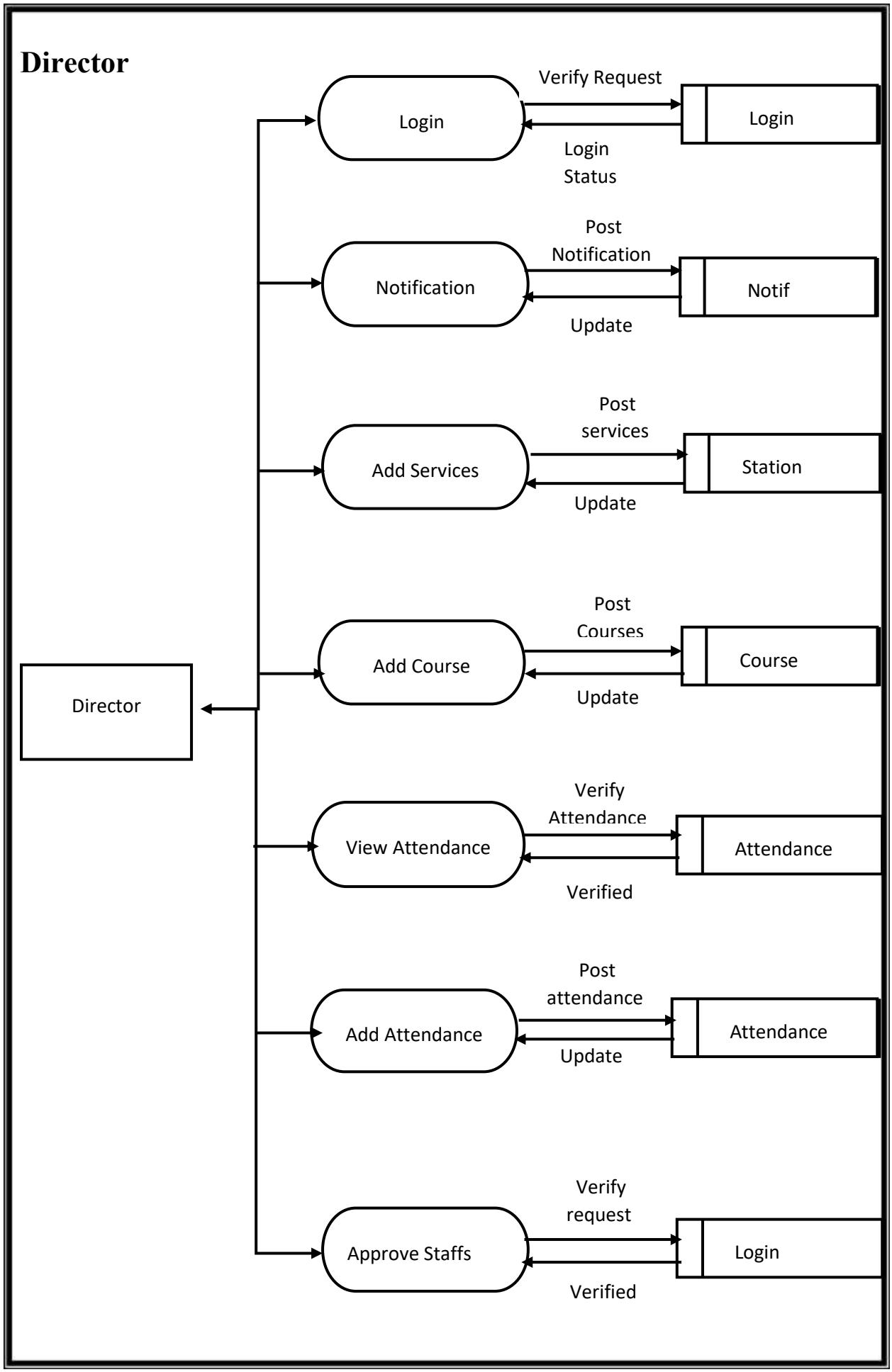
DFD 0th level

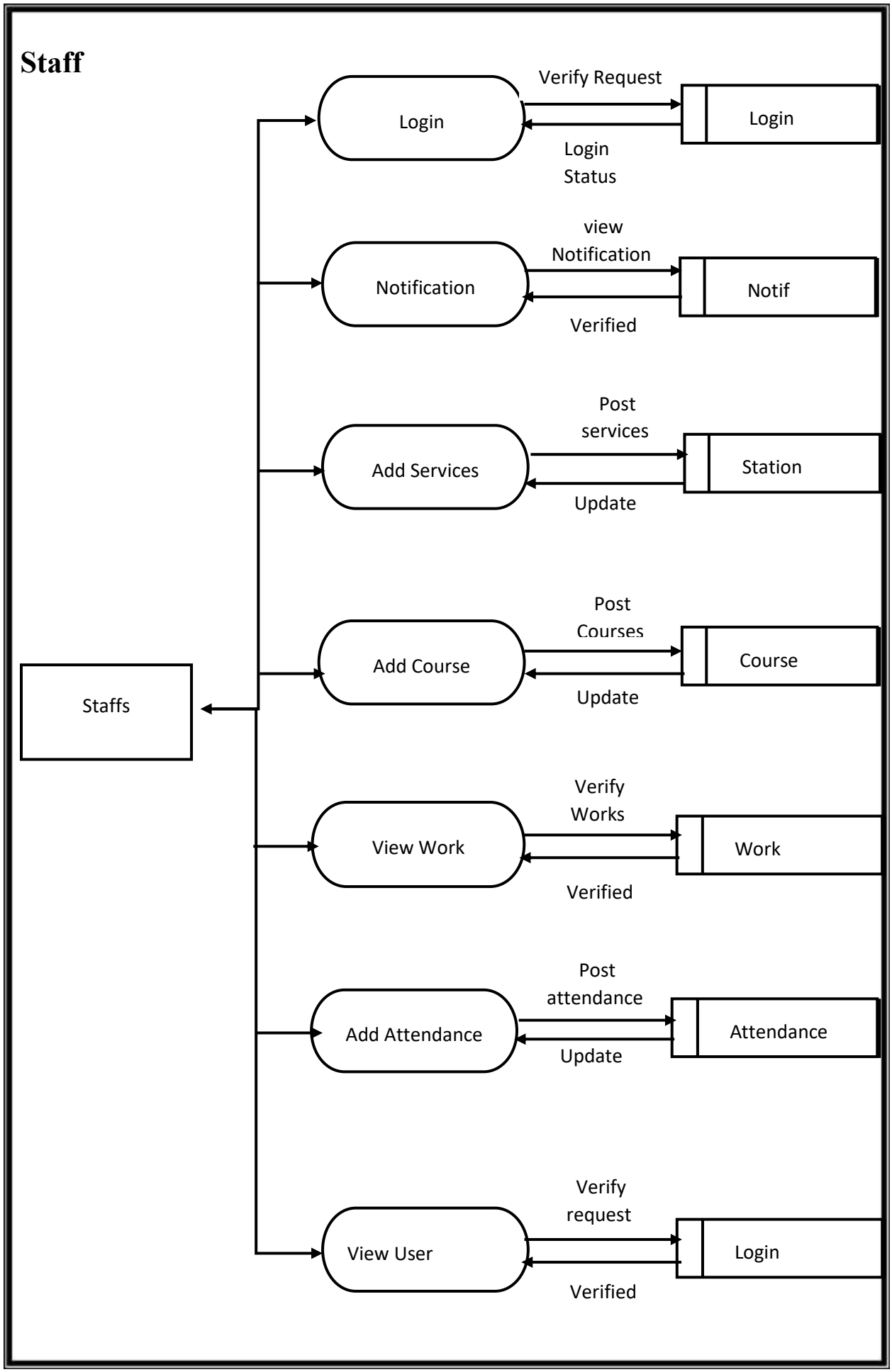


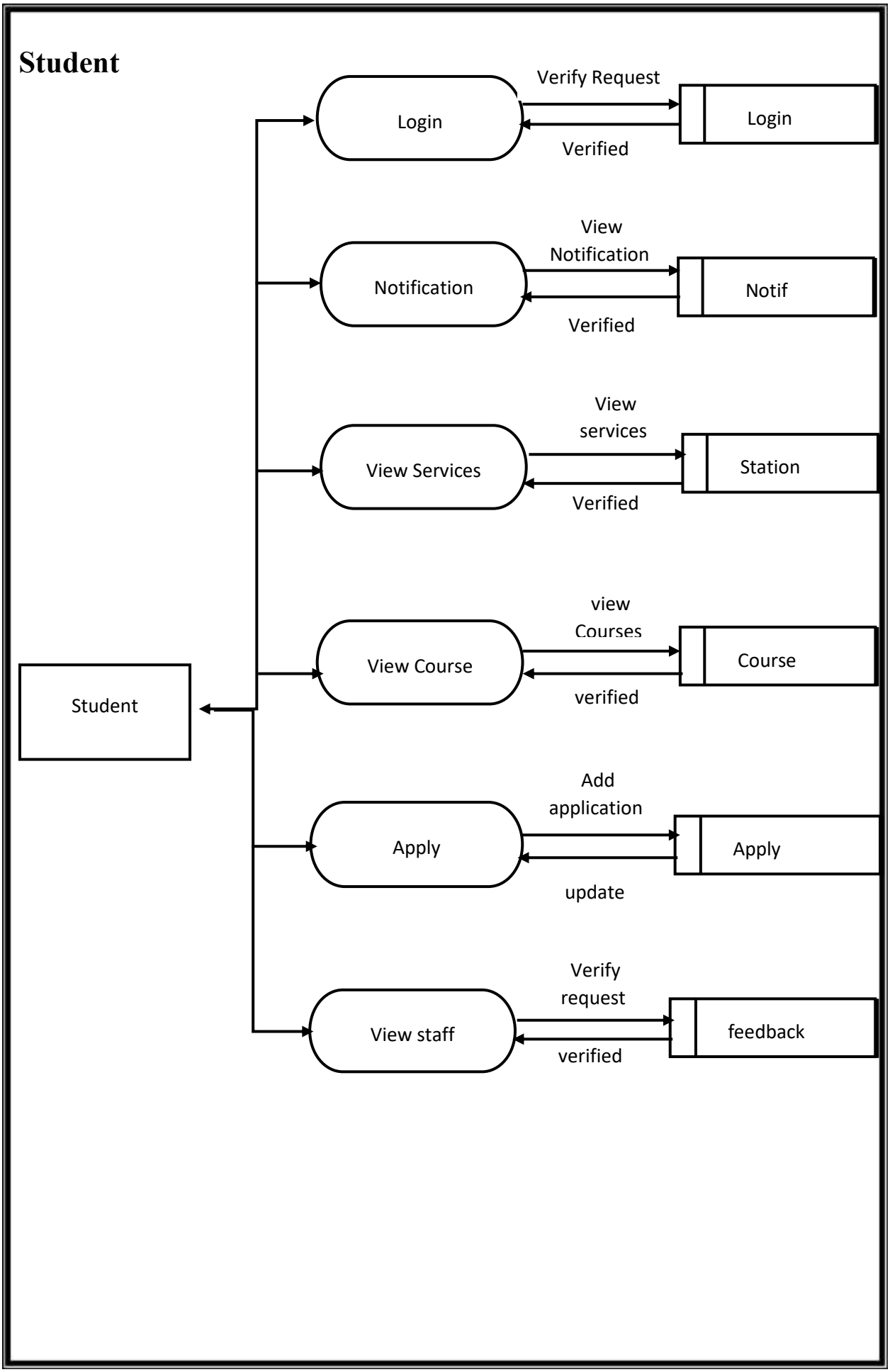
DFD 1th level

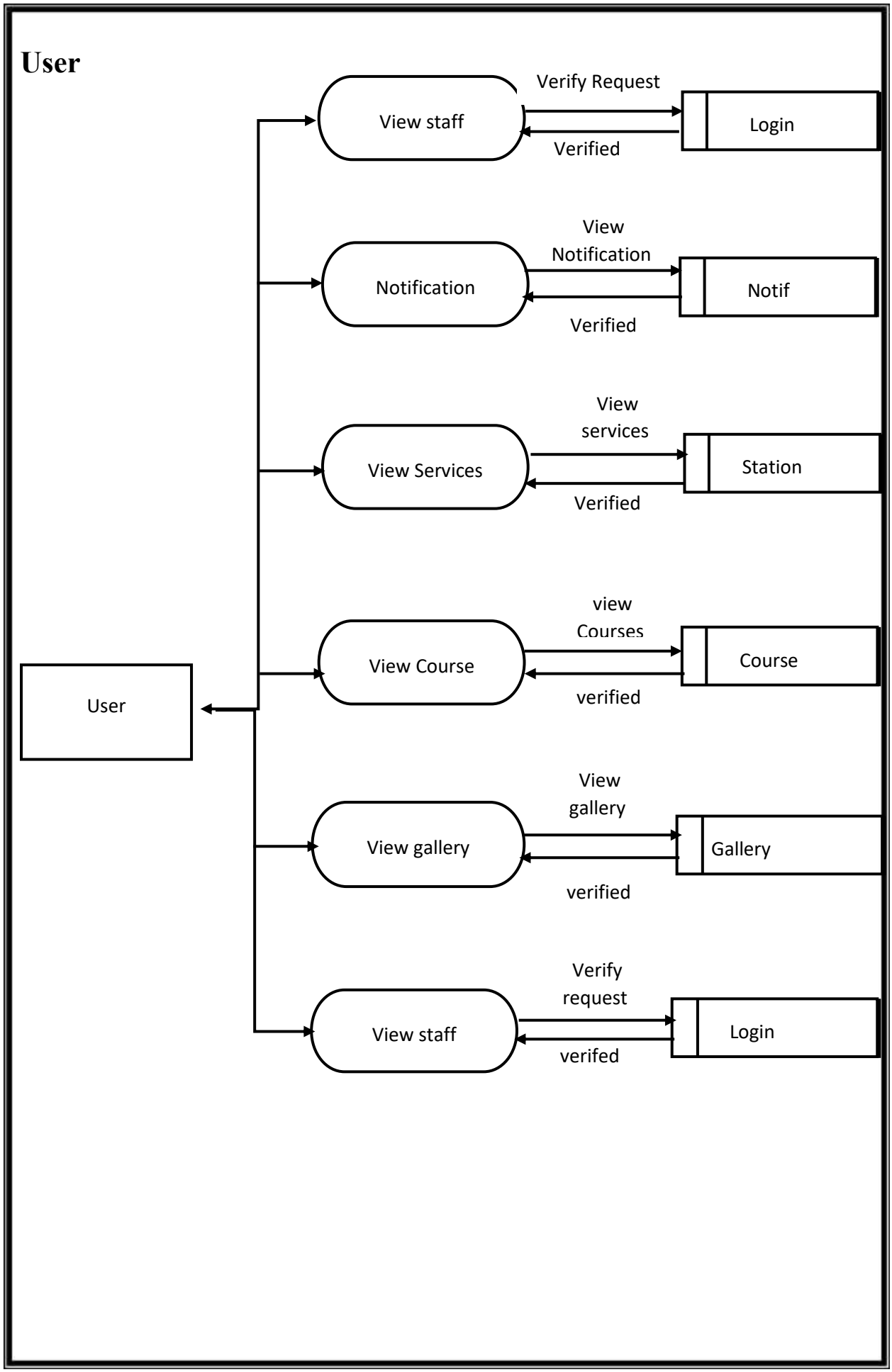
Admin




Director
Verify Request
Login
Login
Login Status
Post Notification
Notification
Notif
Update
Post services
Add Services
Station
Update
Post Courses
Add Course
Course
Update
Director
Verify Attendance
View Attendance
Attendance
Verified
Post attendance
Add Attendance
Attendance
Update
Verify request
Approve Staffs
Login
Verified

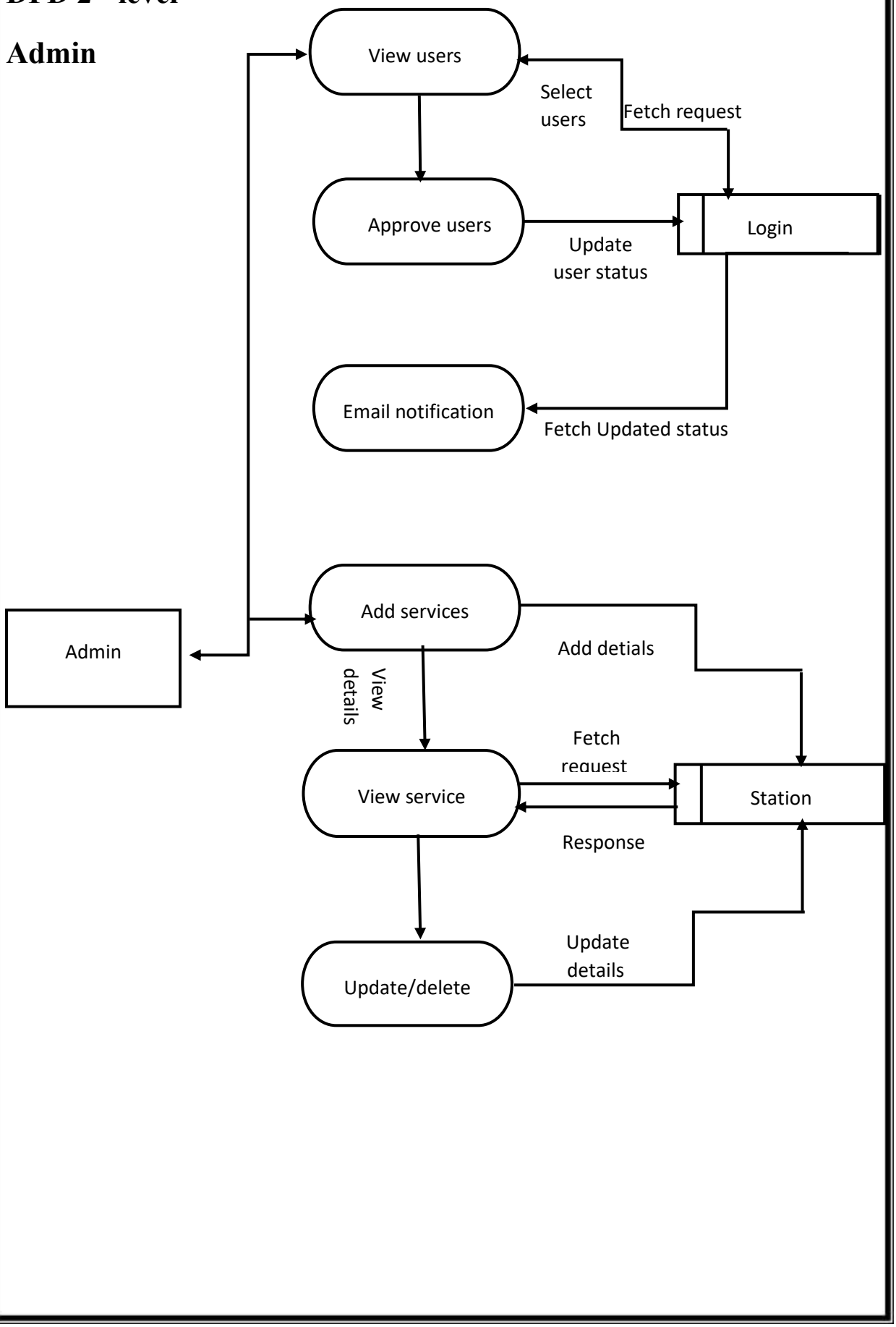

Staff
Staffs
Login
Verify Request
Login
Login Status
Notification
view Notification
Notif
Verified
Add Services
Post services
Station
Update
Add Course
Post Courses
Course
Update
View Work
Verify Works
Work
Verified
Add Attendance
Post attendance
Attendance
Update
View User
Verify request
Login
Verified

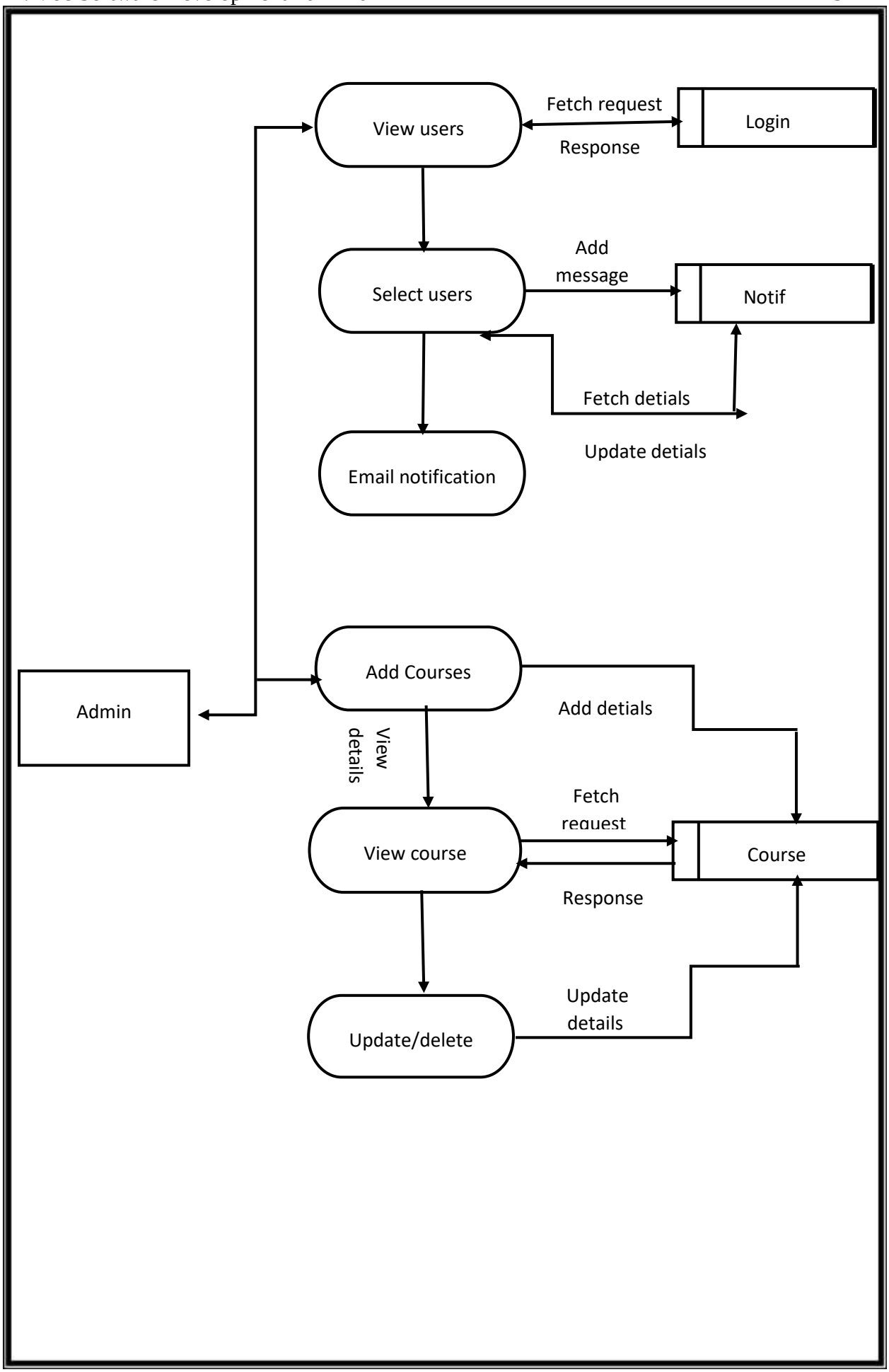

Student
Verify Request
Login
Login
Verified
View
Notification
Notification
Notif
Verified
View
services
View Services
Station
Verified
view
Courses
View Course
Course
Student
verified
Add
application
Apply
Apply
update
Verify
request
View staff
feedback
verified

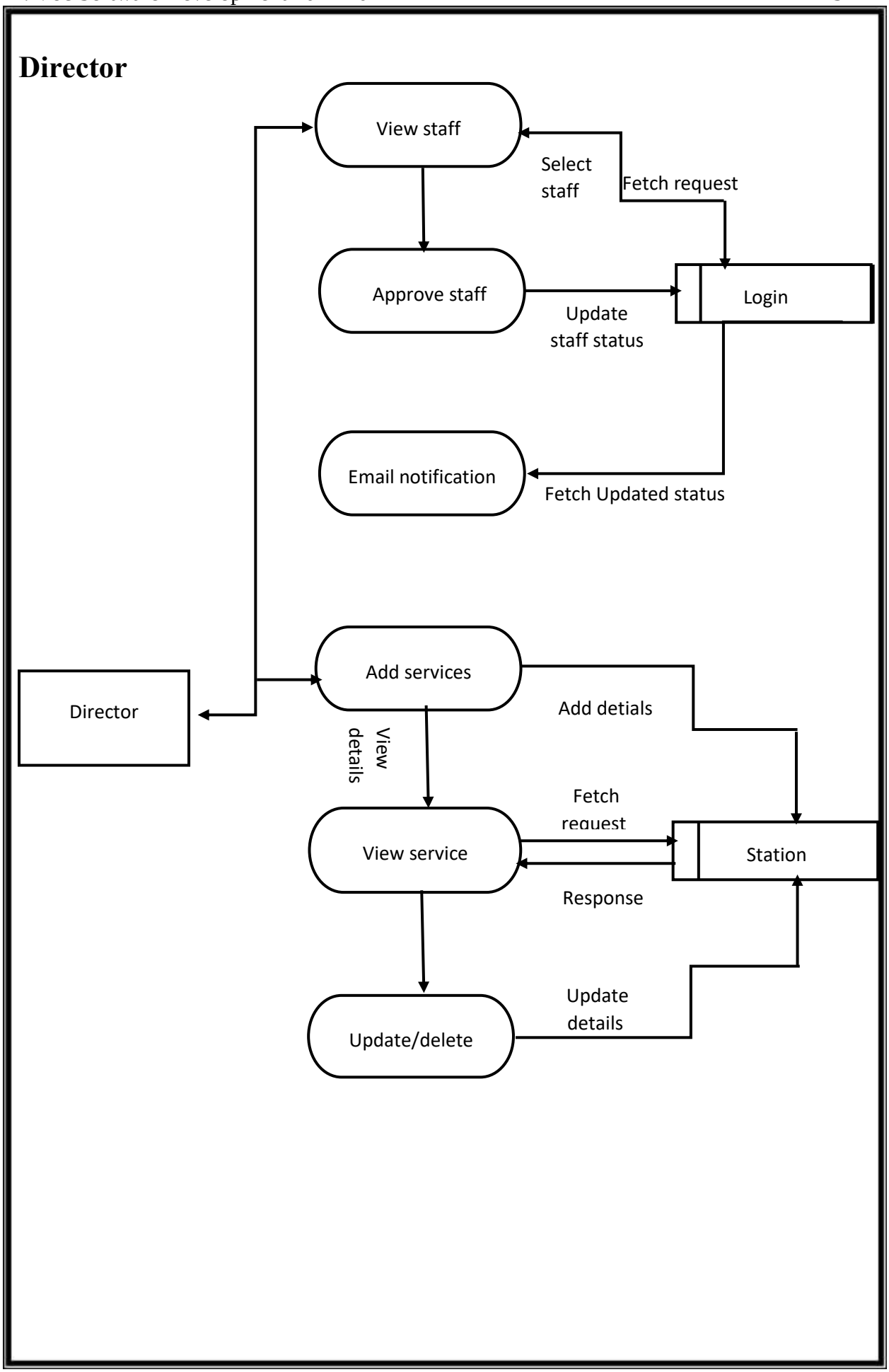

User
View staff
Verify Request
Login
Verified
Notification
View
Notification
Notif
Verified
View Services
View
services
Station
Verified
View Course
view
Courses
Course
verified
User
View gallery
View
gallery
Gallery
verified
View staff
Verify
request
Login
verifed

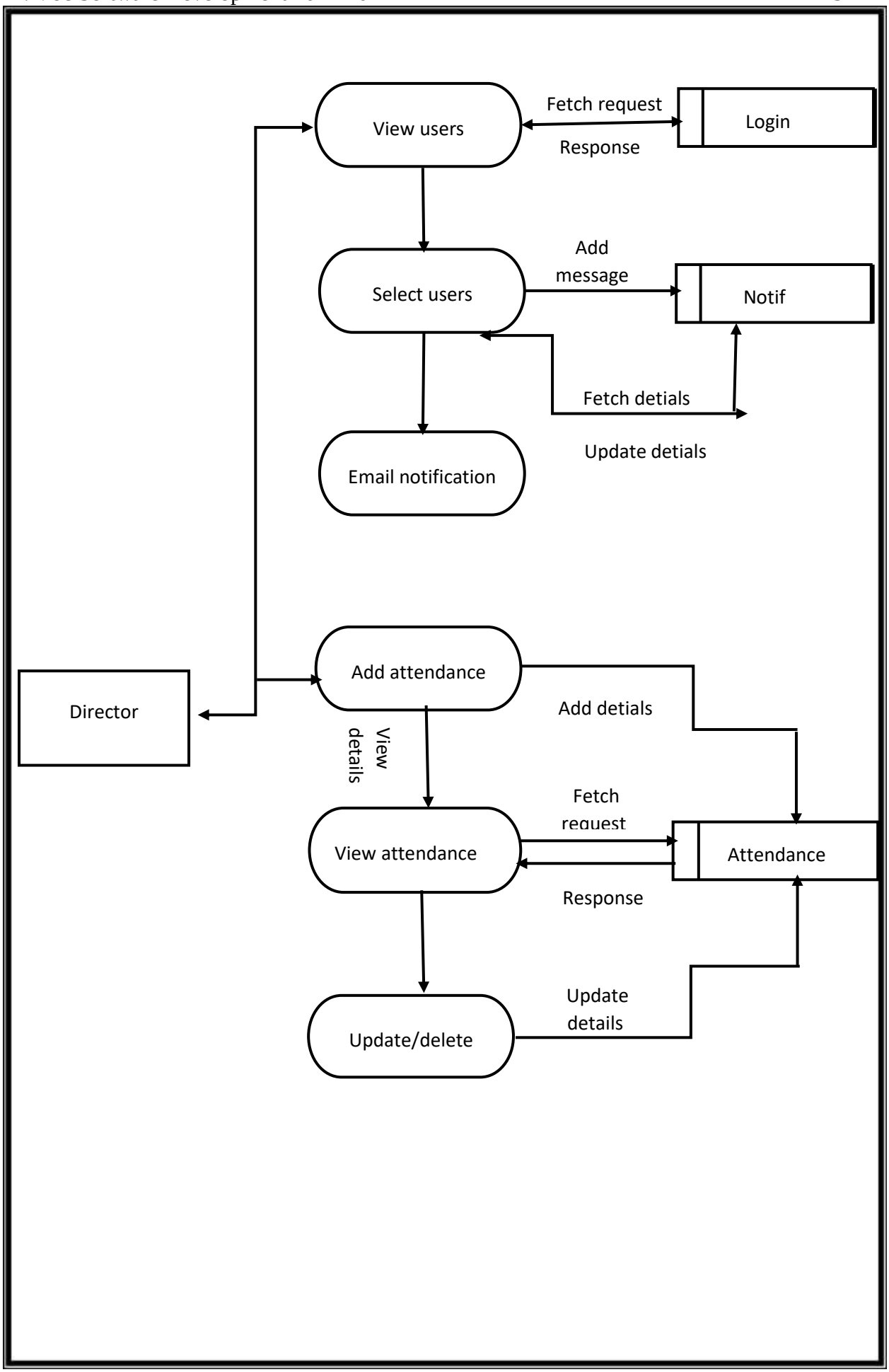
DFD 2th level

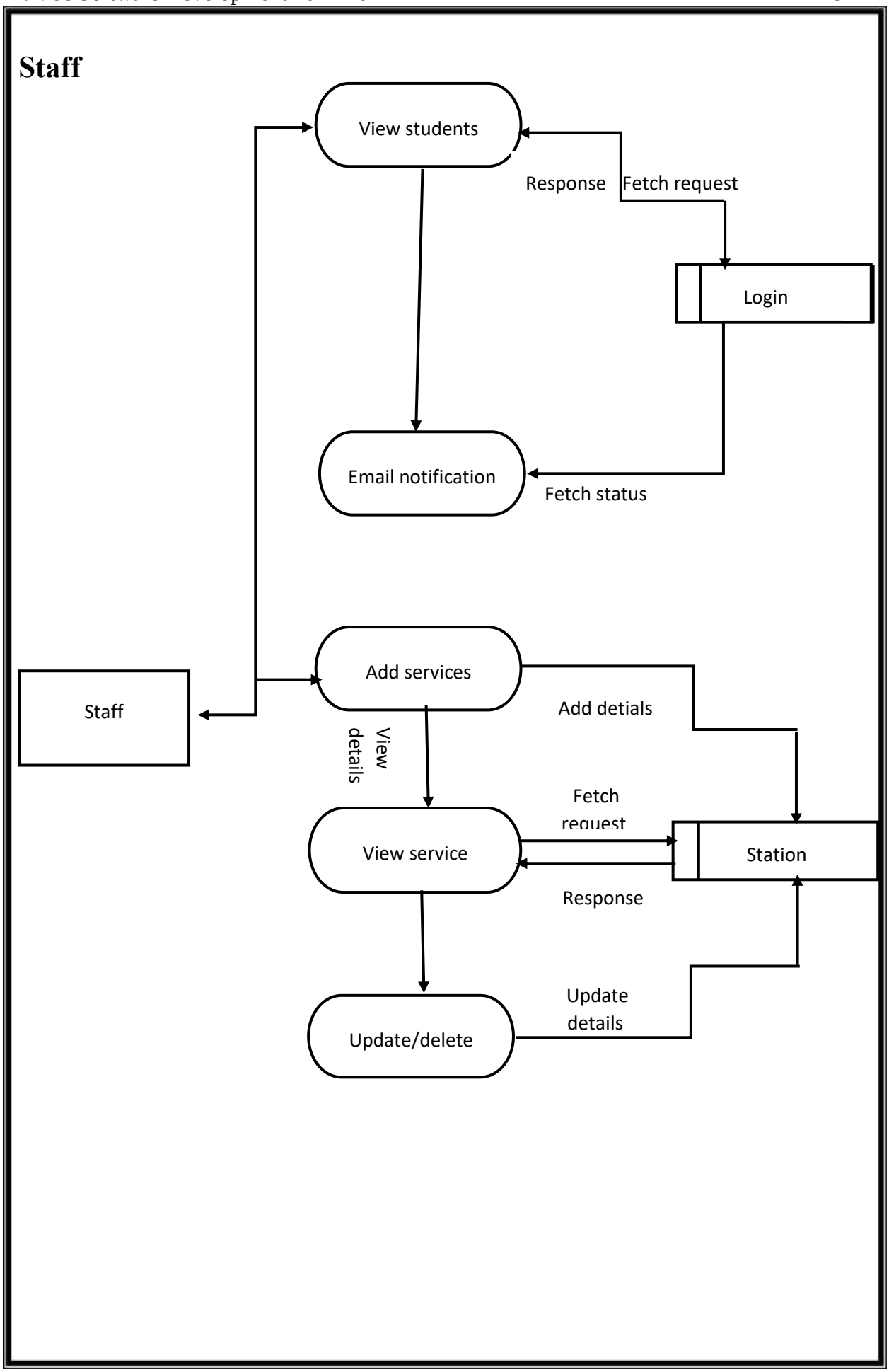
Admin

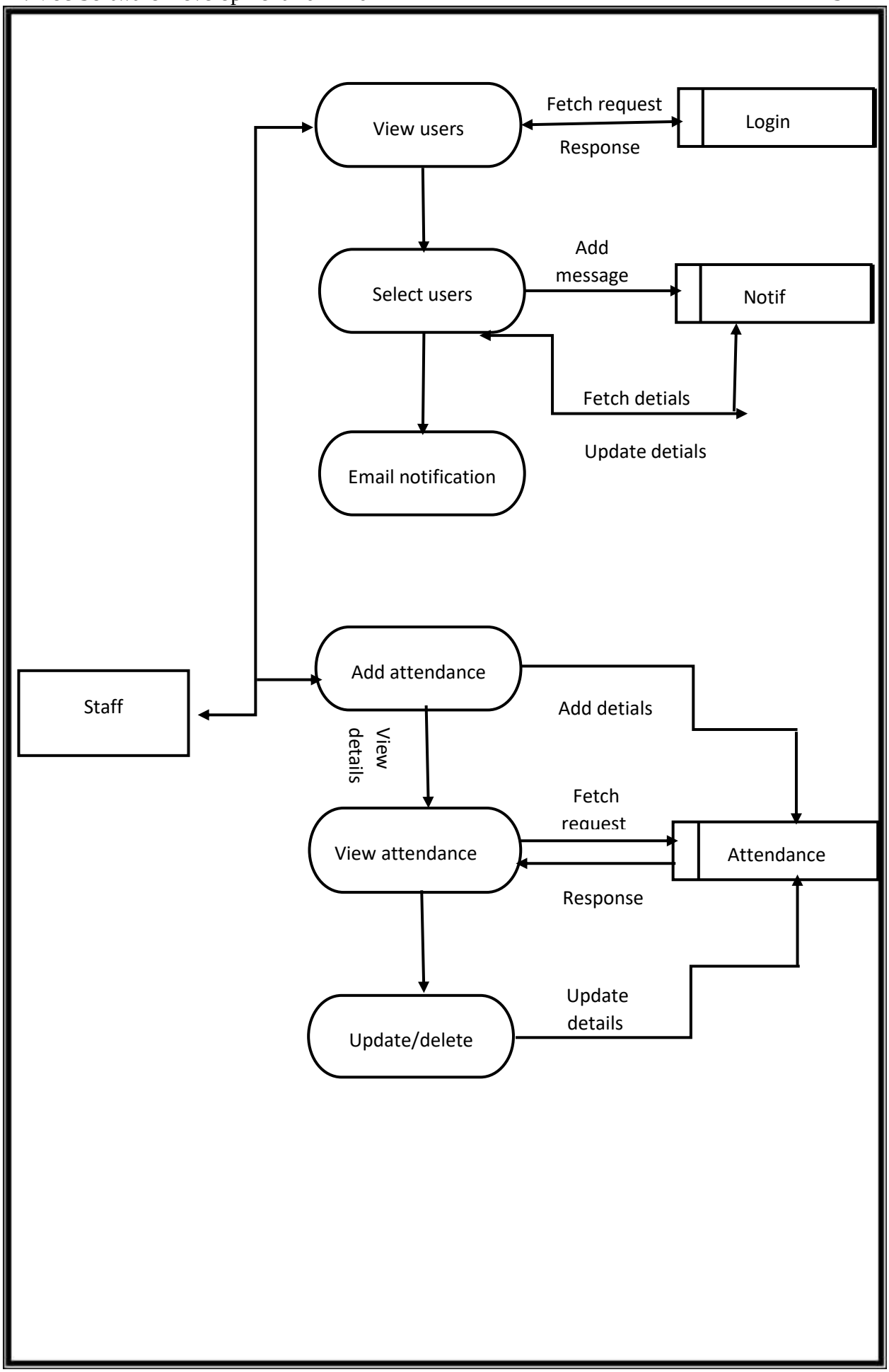


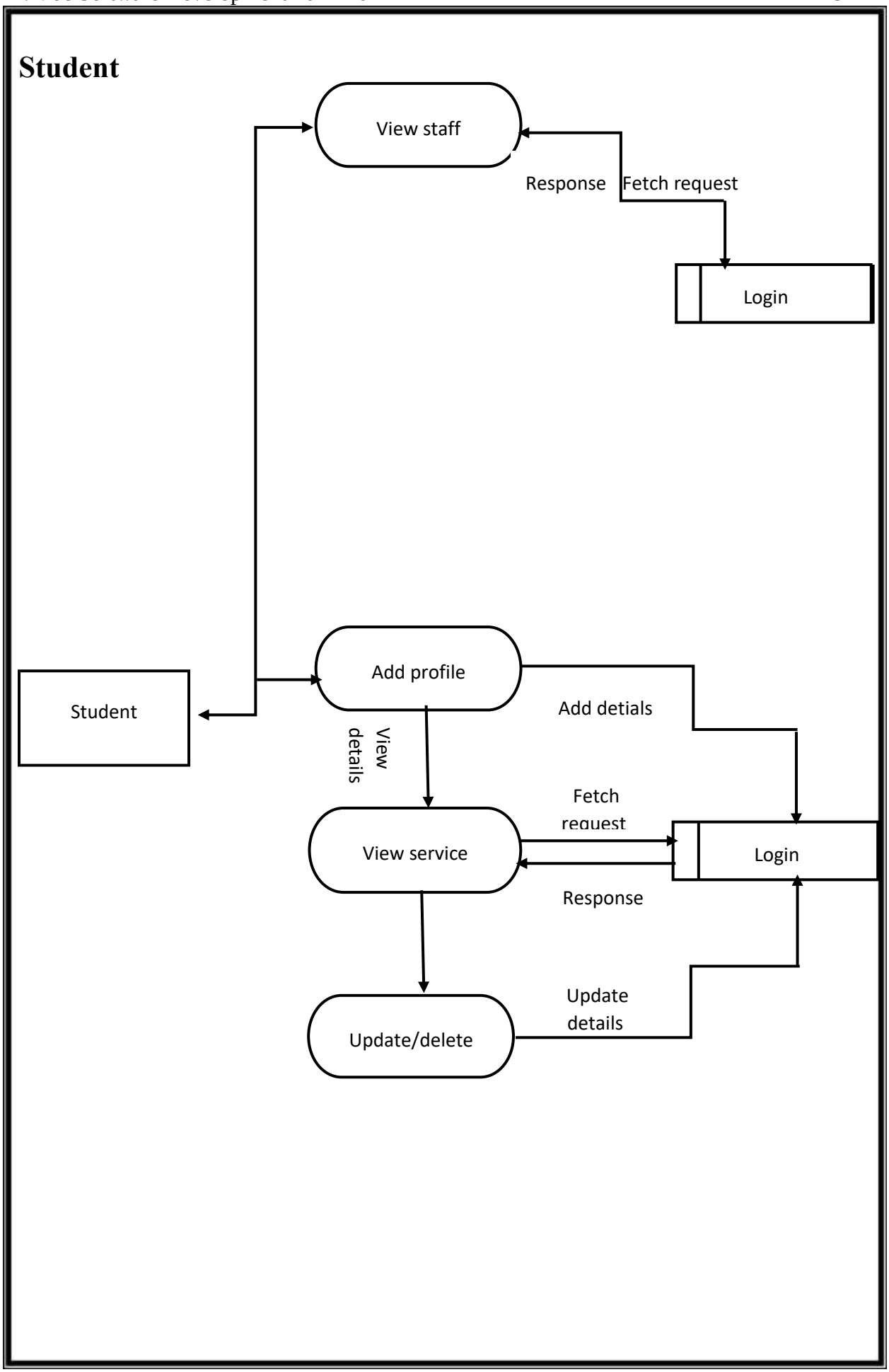

View users
Fetch request
Response
Login
Select users
Add
message
Notif
Fetch detials
Update detials
Email notification
Admin
Add Courses
Add detials
View
details
Fetch
request
View course
Course
Response
Update
details
Update/delete

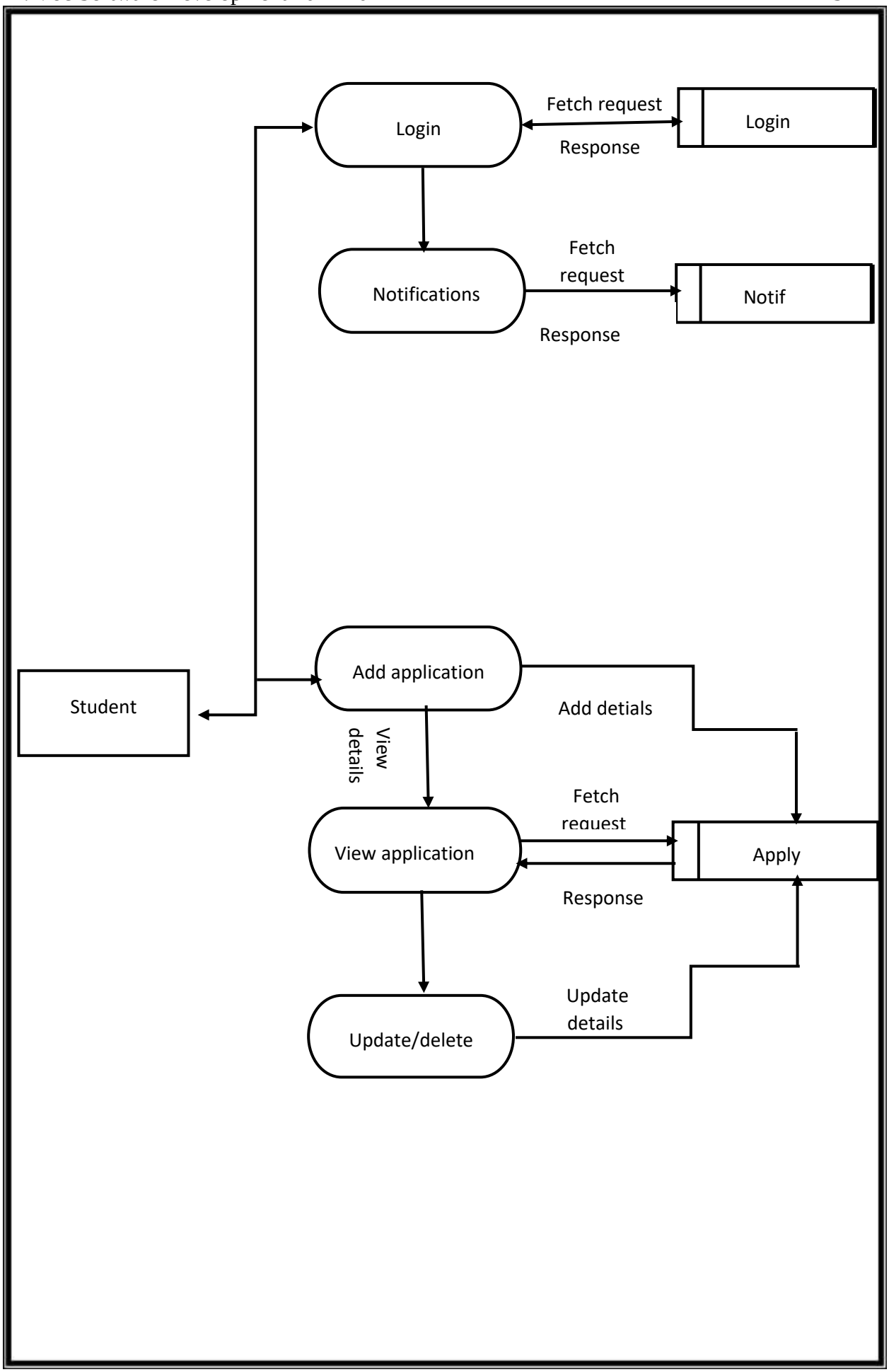

Director
View staff
Select staff
Fetch request
Approve staff
Login
Update staff status
Email notification
Fetch Updated status
Add services
Director
Add detials
View details
Fetch request
View service
Station
Response
Update details
Update/delete

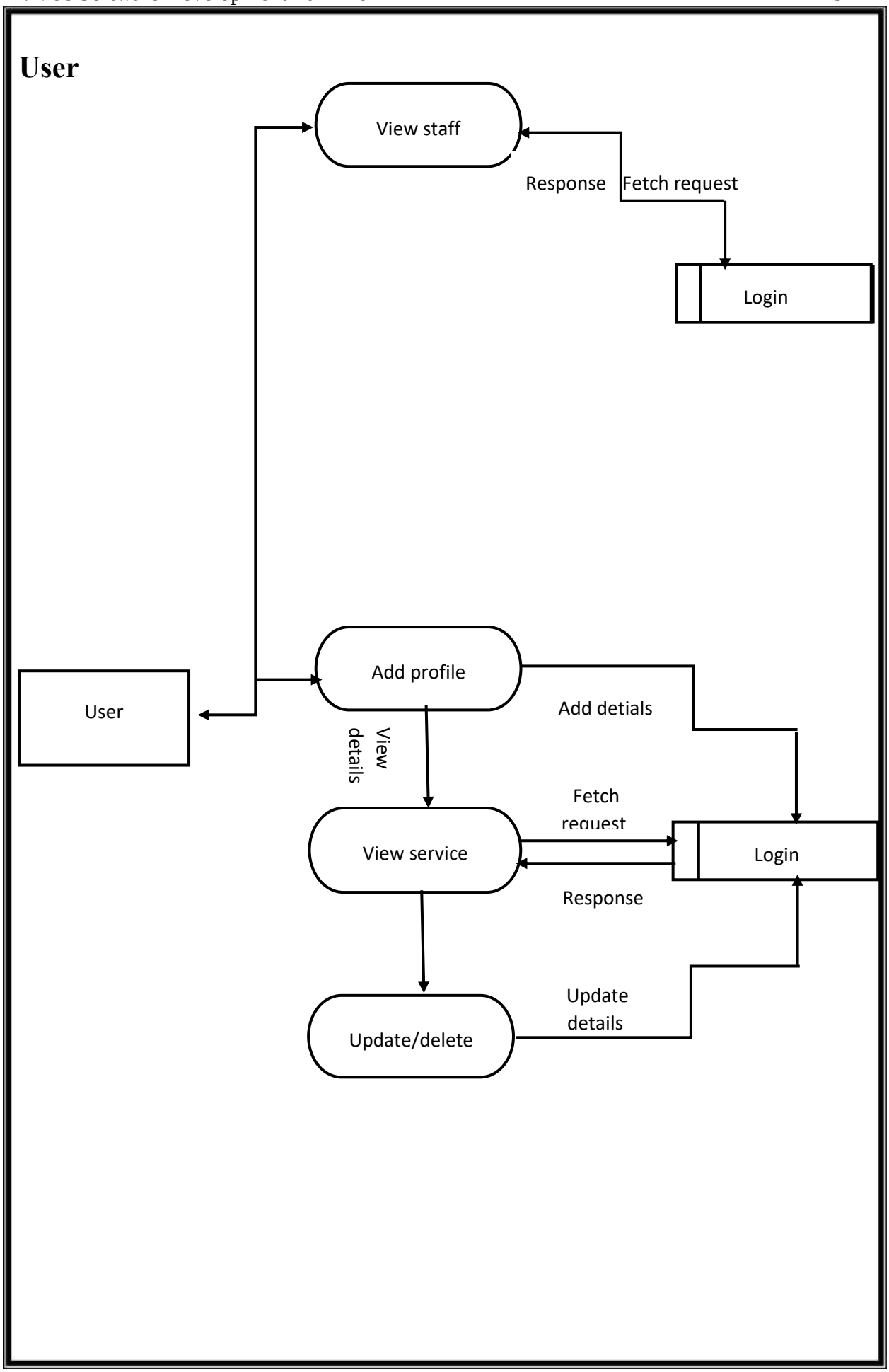

View users
Fetch request
Response
Login
Select users
Add
message
Notif
Fetch detials
Update detials
Email notification
Director
Add attendance
Add detials
View
details
Fetch
request
View attendance
Attendance
Response
Update
details
Update/delete

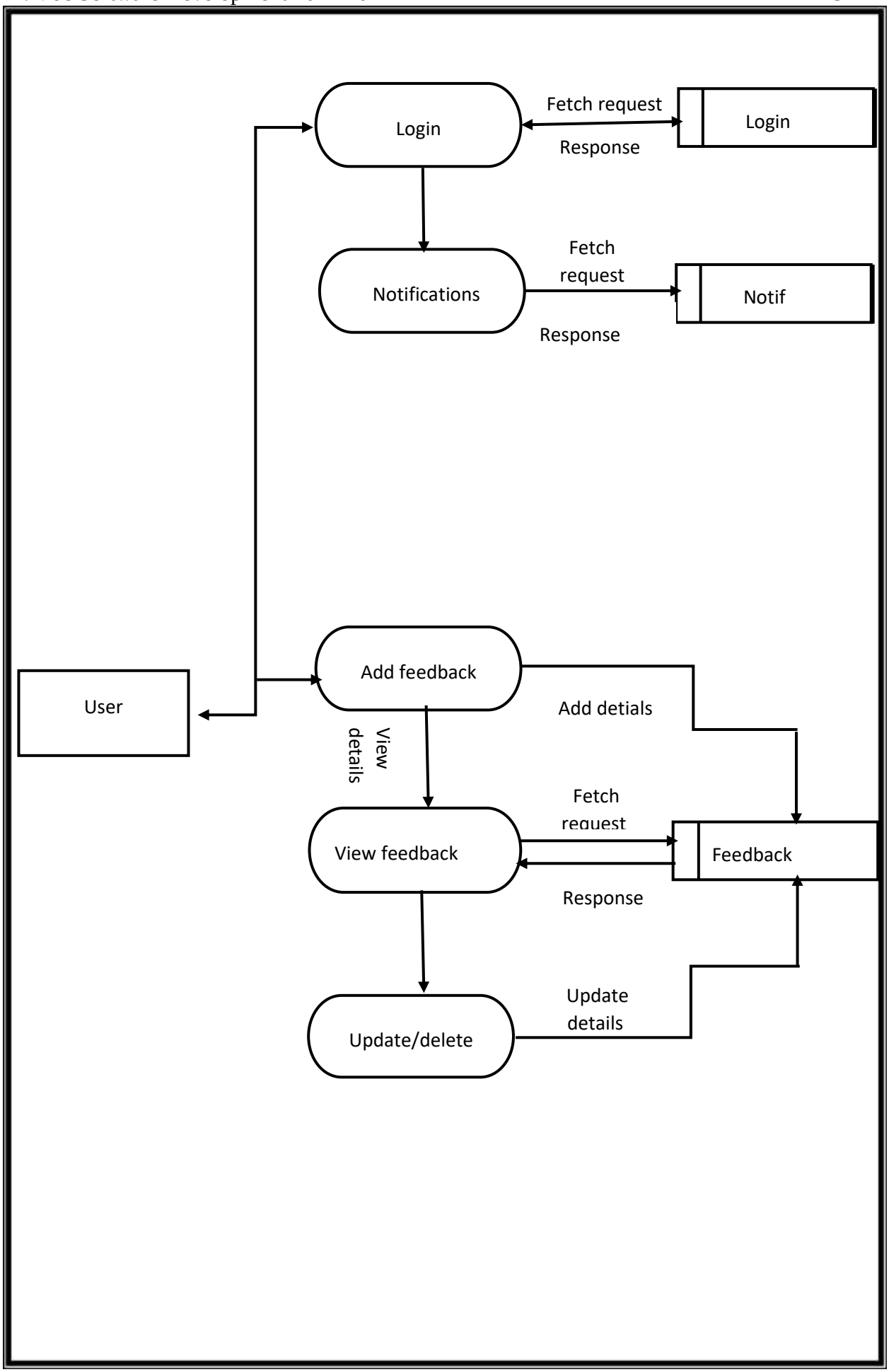

Staff
View students
Response
Fetch request
Login
Email notification
Fetch status
Staff
Add services
Add detials
View details
Fetch request
View service
Station
Response
Update details
Update/delete


View users
Fetch request
Response
Login
Select users
Add
message
Notif
Fetch detials
Update detials
Email notification
Staff
Add attendance
Add detials
View
details
Fetch
request
View attendance
Attendance
Response
Update
details
Update/delete

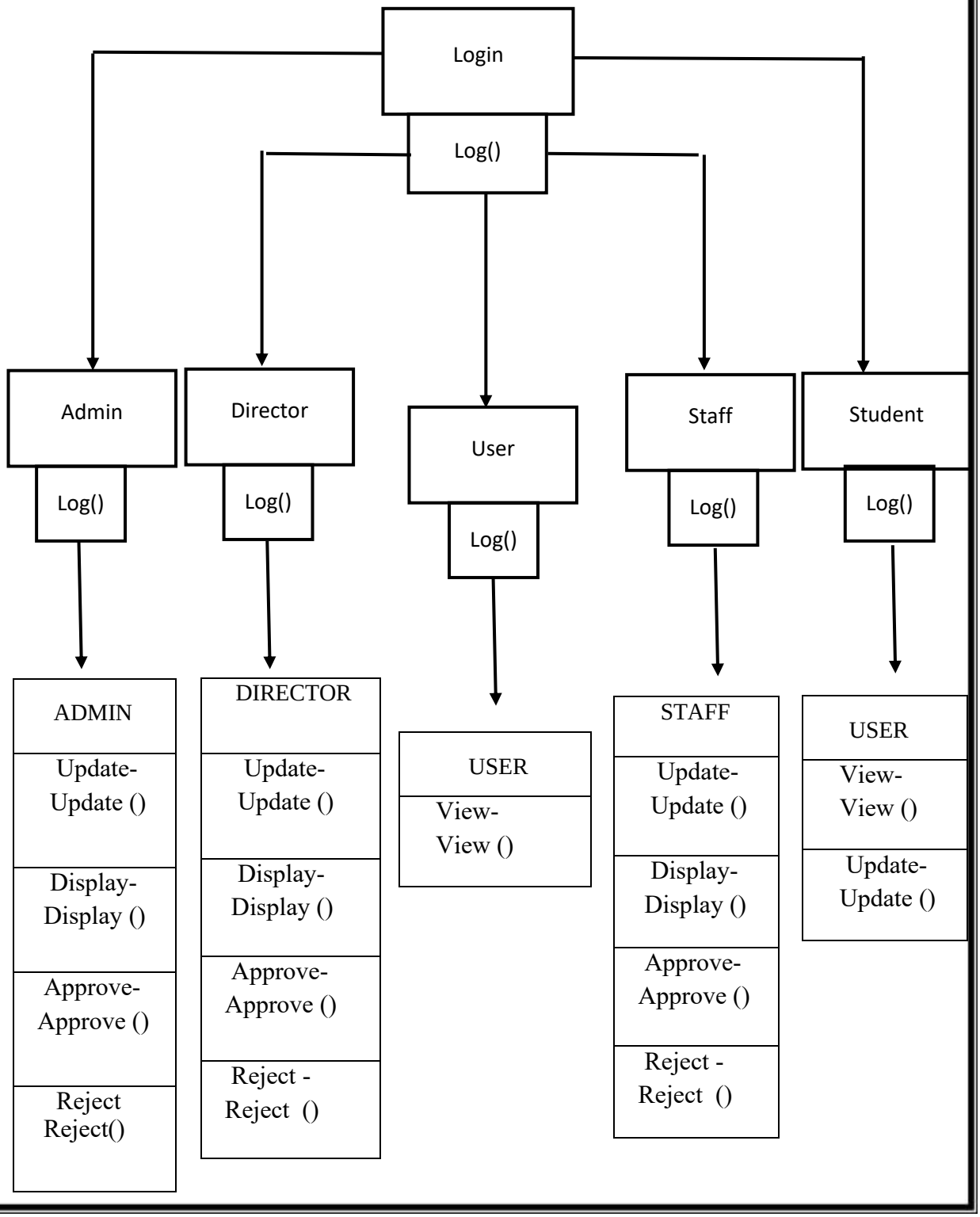

Student
View staff
Response
Fetch request
Login
Add profile
Student
Add detials
View details
Fetch request
View service
Login
Response
Update details
Update/delete


Login
Fetch request
Response
Login
Notifications
Fetch request
Response
Notif
Student
Add application
Add detials
View details
Fetch request
View application
Apply
Response
Update details
Update/delete

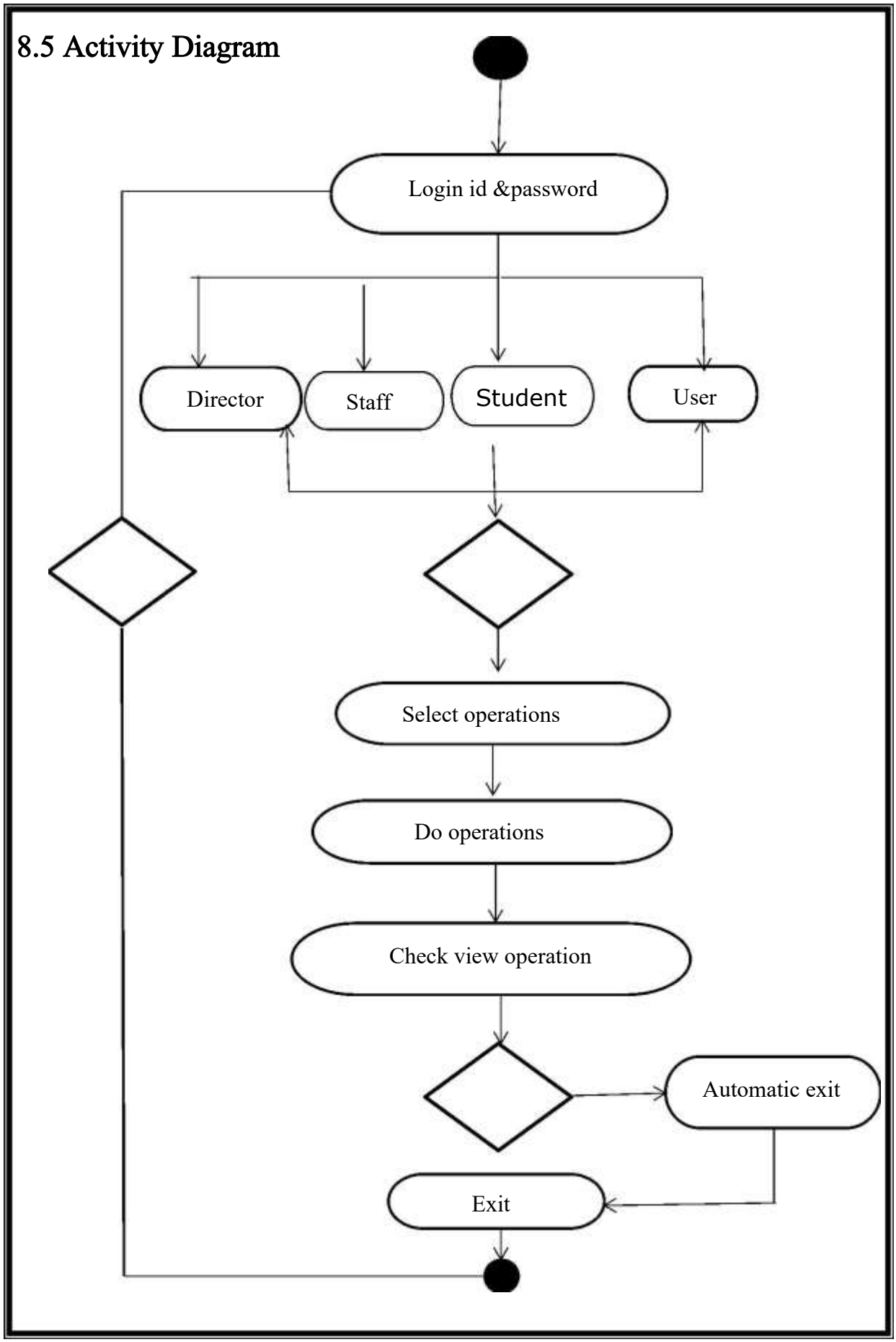

User
View staff
Response
Fetch request
Login
Add profile
User
Add detials
View details
Fetch request
View service
Login
Response
Update details
Update/delete


Login
Fetch request
Response
Login
Notifications
Fetch request
Response
Notif
User
Add feedback
Add detials
View details
Fetch request
View feedback
Feedback
Response
Update/delete
Update details

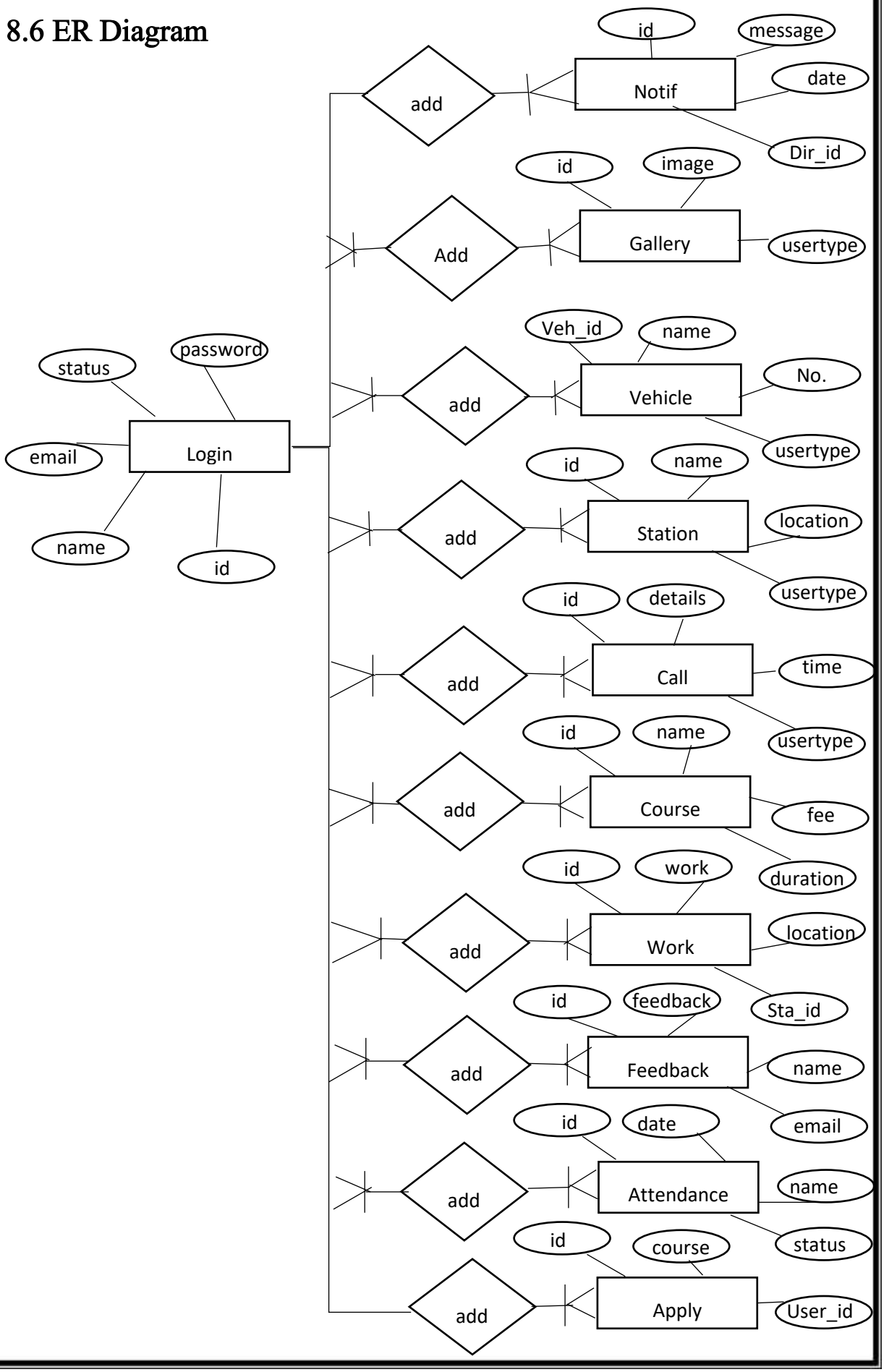
8.4 Class Diagram



8.5 Activity Diagram



8.6 ER Diagram



BIBLIOGRAPHY

Bibliography

BOOKS

- Learning Python Learn to code like a professional with python author: FABRIZIO ROMANO
- Python 101 Author MIKE DISCROLL
- Software Engineering: A practitioner's Approach Author : ROGER S PRESSMAN

WEBSITES

- www.pythontutorialpoint.com
- www.tutorialspoint.edu
- www.stackoverflow.com