Final Year Project Documentation

Project Name: Display Notification System

Project advisor: **Mr. Waqas Asghar**

Project supervisor: **Mr. Fahad Sabah**

Presented by:

BSCS 14226 Talha Riaz

BSCS 14258 Umair Shahbaz

BSCS 14245 M. Adil Amin

BSCS 14252 Shoaib Arshad

DECLARATION

We hereby declare that the project work entitled **"Digital Notification System"** submitted to the Superior University, is a record of an original work done by my group members under the guidance of **Mr. Waqas** Asghar Professor of CS&IT Department of Superior University, this project work is submitted in the partial fulfilment of the requirements for the award of the degree of Bachelors in Computer Science& Information Technology. The results embodied in this document have not been submitted to any other University or Institute for the award of any degree or diploma.

Project advisor: Mr.Waqas Asghar

Signature: _____

Abstract

There are many departments of administration for the maintenance of university information and student databases in any institution.
All these departments provide various records regarding students, classes and announcements. Most of these track record is needed to maintain. This information could be the general details like student name, result, daily and important announcements and special class details.
All the modules in university administration are interdependent. They are maintained manually. So they need to be automated and centralized as, Information from one module will be needed by other modules.
For example if a student wants to see if his class has been relocated to some other room he need not come to coordinator every time he faces a problem the coordinator will display that information every time when it is needed by students on a digital board.

This project is useful for easy user interface. The system utilizes the powerful database management, data retrieval and data manipulation. This project provides more ease for managing the data than manually maintaining in the documents. The project is useful for saving valuable time and reduces the huge paper work.

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to **Prof.** Program Manager of Department of Computer Science, Superior University, for providing us an opportunity to do our project work on Digital Notification System. This project bears an imprint of many people. We sincerely thank our project guide **Mr. Waqas Asghar** a faculty member of Superior University Lahore for guidance and encouragement in carrying out this project work. We also wish to express our gratitude to the officials and other staff members of Superior University Lahore who rendered their help during the period of our project. Last but not least. We wish to avail ourselves this opportunity to express a sense of gratitude and love to our friends and our beloved parents for their support, strength, help and everything else.

Contents

List of Figure

Chapter No: 1

Introduction

1. Introduction

1.1 Problem

Now, what is the problem and why we've made this system? Well as you can see that whenever a result is displayed on notification wall or an announcement is displayed there comes a rush around it and is has happened number of times that when the result gets updated there are always some errors that do not gets updated or the result is pulled from the wall by some student. There is also a problem that when a class needs to know that at what time there is a class and with whom and it causes a lot of headache to class CR but by using this system every student can know all the information regarding their next class. So, by now you should've know that this system can solve many daily academic problems a student or a faculty are suffering.

1.2 Challenges in current system

Order of Data: Notice can get out of order in traditional notice board system. If someone accidentally puts some data in the wrong place, it can lead to lost data. Automated notice management systems allow users to quickly check whether information already exists somewhere in the system, which helps avoid problems like redundant data.

Complexity: Automated system is less complex than manual system of handling notices, which can make it easier for untrained people to access and manipulate data. Anyone having the basic knowledge of computers can work on the automated system.

Inconsistency of data: There will be an unavailability for future use, since notice might get misplaced during manual notices management. So notice won't be preserved properly for future use.

Damage: Manual notices stack are vulnerable to damage, destruction and theft in ways that digital databases are not. A company may back up its digital data both on site and at offsite locations, ensuring its security if the office building suffered a fire or similar disaster. A manual database, however, may only exist in one place without any copies. As a result, a manual database would be very vulnerable to a fire or other natural disaster. In addition, while access time in a manual database system, information must be found by hand rather than electronically. While a digital database will typically allow users to search the entire database for specific information in seconds, someone looking for information in a manual system may have to spend hours searching for a particular piece of data.

Editing and Communication: Manual notices do not allow users to easily edit data or information. Manual notices often cannot be edited directly, forcing users to make new copies. To circulate notice on paper, users must require peons and other staff. Digital Notification System app allow users to edit information fields directly, and because data is stored digitally, it is already in a form that can be easily transmitted.

1.3 Purpose

Our purpose that we are trying to achieve is to make an application that helps students and faculty members to connect with information easily. We want to help reduce crowd situations whenever an information is displayed on notice board, we want to be looked professionally moderate society, we want people to show that yes Pakistan is a digitally connected country.

1.4 Introduction:

Digital Notification System App helps you access notices on just a screen. It is an online notice board maker where a group of people can easily post announcements or any other type of notices on a digital board. These notes can have text, images or include online videos such as from YouTube.

The notice board has always been the place where staff/students gathers to get their latest release of corporate news and this application brings the notice board to a virtual location where staff/students can read notices. With this electronic notice and announcement system, notification alerts may be sent out notifying staff and students that a new notice has been posted, where staff may know if it concerns him directly.

These are the features that this application has:

• An electronic dashboard board for disseminating information out to staff and students.

• Notices can be posted, with response obtained instantly.

• Notice administrator may push important notices in to selected category.

• Notice administrator may create any notice category. The interface of this application is straight forward and takes you roughly a minute to get started. Adding notes to a digital board is easy, just click on the post notice button and enter the text. Users can view the post on the spot.

This application creates interactive notice boards. You can customize boards with a custom background, title and background images. It allows students to post notices with the permission from HOD, with text, images and videos. You can set preferences on who can edit and view the board.

1.5 **Identification of Need**

1. As we discussed earlier that manual maintenance of a notices is a tedious job. So to enhance the ease of working, we go for this package.
2. Giving the facility to convey messages to all students anytime
3. Making students updated about all the events and activities going on in the university.
4. The students will not require to stand in the crowd to see the notice. There will be no issue of fighting in order to see the notice first.
5. The least but most important it saves time.
6. Utilizing less man power. As there are many persons involved in circulating the message. With this application, only one person is required to post the notice. Rest of the man power is saved in the entire process.

1.6 **Motivation**

The motivation for this idea came when we saw a lot of troubles in handling the notices. Whenever a result is displayed it also brings a lot of headache for students and faculty members. For example whenever a result is displayed there sometime remains an error that shows marks gained by a particular student wrong and that who suspects that result is incorrect goes to teacher and then there stars a tour system for that student to correct the mistakes after a lot of headache the result finally gets updates but what also happens is that a lot of time is wasted and the whole page that is showing the marks is also pulled of and a lot of people who has their eyes on that result page get disturbed now to end this headache we have created this system. Now this system would speed-up the process of correcting any mistake, the coordinator would have to just correct that result on the application that we have given it will display that result on digital board in few minutes in this the headache of faculty and students will get removed.

1.7 **Objective**

Our objective is to make a system through which academic information is easily available for students. So, a lot of information can be displayed in just a small area i.e. on a digital board so there would be no confusion or fuss regarding that information.

1.8 **Scope**

The system will have to interact with a university environment. Therefore we are interested in keeping under control the adjacent systems such as the database, the university department software, the mailing system, or the computerized environment. Regarding the human environment, we have to consider the program manager above all and the indulgency for the rules of the faculty. This system should not be considered an error free system, there should be some tolerance regarding this system.

Chapter No: 2

Existing and Proposed solutions

2. Project solution and Limitation

2.1 Existing solution

Currently our university has manual system of putting notices on noticeboard. It's out dated now. As nobody has a time to stand in rush in order to read the notices on noticeboard.

2.2 Limitations of Existing Solution

2.2.1 Order of Data

Notice can get out of order in traditional notice board system. If someone accidentally puts some data in the wrong place, it can lead to lost data. Automated notice management systems allow users to quickly check whether information already exists somewhere in the system, which helps avoid problems like redundant data.

2.2.2 Complexity

Automated system is less complex than manual system of handling notices, which can make it easier for untrained people to access and manipulate data. Anyone having the basic knowledge of computers can work on the automated system.

2.2.3 Inconsistency of data

There will be an unavailability for future use, since notice might get misplaced during manual notices management. So notice won't be preserved properly for future use.

2.2.4 Damage

Manual notices stack are vulnerable to damage, destruction and theft in ways that digital databases are not. A company may back up its digital data both on site and at offsite locations, ensuring its security if the office building suffered a fire or similar disaster. A manual database, however, may only exist in one place without any copies. As a result, a manual database would be very vulnerable to a fire or other natural disaster. In addition, while access time in a manual database system, information must be found by hand rather than electronically. While a digital database will typically allow users to search the entire database for specific information in seconds, someone looking for information in a manual system may have to spend hours searching for a particular piece of data.

2.2.5 Editing and Communication

Manual notices do not allow users to easily edit data or information. Manual notices often cannot be edited directly, forcing users to make new copies. To circulate notice on paper, users must require peons and other staff. Digital Notification System app allow users to edit information fields directly, and because data is stored digitally, it is already in a form that can be easily transmitted.

2.3 Proposed Solution

2.3.1 To eliminate wastage of time and energy

Digital Notification System will be able to save lot of paper and time. It directs both teacher and student's energy and attention to one thing at a time by placing proper persons at their proper places at the proper time. Everything will be instantaneous**.**

2.3.2 To avoid duplication and overlapping

This application will help to remove the duplicity of notices. Only one person, who is admin can post the notice. No one else would be able to do so. So student and staff will be given correct information all the time.

2.3.3 To ensure due attention of student to each and every notice

Digital Notification System ensures that everyone has kind attention to every notice and updates going on in university. There will be a buzz at each and every notice to drive the attention of student to check it once. In this way, students will be well informed about their college activities.

2.3.4 To bring system in to college life

It would be a dire need of all universities as it is an easy and shortcut method to inform all the students. As in the absence of proper notification system it will be very difficult to inform students at right time.

2.3.5 Searching a particular Notice

This application allows you search the notice very easily through title of notice. If anyone forgets about the notice details, he can search it out very easily.

2.3.6 Prevent Crowd in College

As you can see, there is always a crowd at notice board. As notice board is one, and people to see notice are more. With this application, there will be no more crowd. Everyone will be well informed. So their time will be saved for other work.

2.3.7 Automatically Updated Dashboard

The dashboard of notice board is automatically updated when a new message arrives. The user can himself refresh the dashboard to see any new notice**.**

2.3.8 Anytime Service

With this application, notices will be delivered anytime. There is no restriction of time to send a notice.

2.3.9 Keeping Notices at one place

This application allow you to have notices in one place only. If there is an attachment with that, all will be placed in a separate folder dedicated to that application. So there will be no here and there of notices.

Chapter No: 3

Analysis

3. Analysis Phase

3.1 Software Requirement Specification

3.1.1 Data Requirements

Data requirement is meant to be the data that will be used in our application. Data required in this project is all notices that need to be conveyed to the user. This application also require the username and passwords of persons in order to register them and sending notification about updates. So two main requirements are:

- Notice Details
- User Details

3.1.2 Functional Requirements:

In order to make this application functional, we require the following:

User registration: Given that a user has a computer network, then the user should be able to register through the web process. The user must provide user-name, password and e-mail address. The user can choose to provide a regularly used phone number too.

User Login: Given that a user has registered, then the user should be able to log in to this service. The log-in information will be stored on the computer and server so for the future use the user should be logged in automatically.

Password: Given that a user has registered, then the user should be able to retrieve his/her password by e-mail.

Dash Board: Given that a user is logged in to the website, then the first page that is shown should be the dashboard page. The user should be able to see all the options from which he/she can post notices.

Search Notice: The user should be able to search for a notice by its title. For example, if a user types fee, all the notices having fee in their content get displayed.

Deleting Notices: The user should have the option to delete the unnecessary notices from the server and computer, by ticking them one by one and then deleting those in one go. This way, user can save the computer's memory from unrequired notices.

Posting Notices: The admin of this application should be able to post the notices. He should be able to add a picture within notices. That picture can be taken either from gallery or by using the camera of the mobile phone.

3.2 Performance Requirements:

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

Prominent search feature: The search feature should be prominent and easy to find for the user.

Fault Tolerance: The fault tolerance of the system is very good. If the system loses the connection to the Internet or the system gets some strange input, the user should be informed.

3.2.1 System Dependability

Following are the requirements that an application require from the computer on which it is installed.

Internet Permission: Application developed, require full internet permissions of computer and Digital Board so that it can fetch notices from the server. At the same time, it should be able to receive buzz or notification tone whenever a new notice is posted by admin.

System Tools: This application require various system tools to be used. For example, it requires Camera of a mobile in order to click the image and post in into notice. It also require system tool that prevents it from sleeping.

3.2.2 Maintainability Requirements

Following are the maintainability requirement of Digital Notification System:

Application extendibility: The application should be easy to extend. The code is written in a way that it favors implementation of new functions. It is required in order for future functions to be implemented easily to the system.

Application test ability: Test environments should be built for the application to allow testing of the application's different functions.

3.2.3 Security Requirements

Communication Security: We have provided security for the communication between the system and server. The messages are encrypted for log-in communications, so others cannot get user-name and password from those messages.

Admin Login Account Security: If an admin tries to log in to the web portal with a non-existing account then the admin should not be logged in. The admin should be notified about log-in failure.

Admin Account Security: There is also security of admin accounts. An admin and IP address cannot be able to log-in to the web portal for a certain time period after three times of failed log-in attempts.

3.2.4 Look and Feel Requirements

Regarding look and feel, we are straight forward. We believe in simplicity. So these are the requirements:

- **Simple and Light**: The user interface is very simple and lightly colored. It gives a relaxing effect on looking at its GUI. No colors are used that could trouble the viewer and user.
- **Easy to Use**: The application is easy to use. If any user is doing something wrong, he/she should be informed correctly, that what is going wrong behind the scene. There are proper instructions for the user to use this application.
- **Soft Sound Notification**: The sound for notification are very soft. It will not disturb the people with a loud note. Everything is sober in this application.

3.2 Software Quality Assurance Plan

3.2.1 Purpose

The purpose of this document is to specify how the software quality assurance plan is handled in the software development life-cycle of the project “Digital Notification System”. Now we have used Spiral model to bring this project to life. The intended use of this software project is to enhance the pre-existing Notification system to a digital level where all the paper work is finished and converted on digital board. This document is based on the IEEE standard for Software Quality Assurance Plan, IEEE Std 730.1-1995. This document is intended to be used in partial fulfillment of the requirements for the Bachelors in Computer Science. Furthermore,

this document will be reviewed and evaluated by the major professor and the supervisory committee.

3.3 Reference

- Software Requirements Specification, Kansas State University, March 2003, (http://www.cis.ksu.edu/~cme6556/software_requirements_specification_1.0.pdf)
- IEEE Guide for Software Quality Assurance Planning
- IEEE Standard for Software Quality Assurance Planning, IEEE Std 730.1-1995
- Software Project Management: A Unified Framework

3.4 **Management:**

- Advisor: Mr. Waqas Asghar
- Supervisor: Mr. Fahad Sabah
- Developers: Umair Shahbaz and Adil Amin

3.5 **Responsibilities**

Advisor

Primary responsibilities include reviewing each milestone deliverable at the requirements, architecture, and implementation phases. After reviewing, Advisor provided feedback and suggestions to the software developer

Supervisor

In addition to the responsibilities as the supervisor, the supervisor has supervised and evaluated all artifacts submitted by developer. Reviews and walkthroughs of related materials has been conducted.

3.6 **Software Developers**

Since the project is developed by two people, developers are responsible for ensuring quality of the project. The software developers are also responsible for producing the required artifacts for this project.

Furthermore, the following tasks has been conducted in order to ensure quality assurance:

Design: The developers and major professors have conduct reviews and analyses of the construction of the software. Strengths and weaknesses of various design techniques has been discussed and scrutinized.

Implementation: Informal code reviews has been conducted by the developers on a regular basis to ensure consistency with the design and the detection of any error. Also, developers will be there for purpose of maintainability and future work.

Testing: Developers has conducted tests as presented in this document later to ensure the requirement satisfaction and reliability of the software

3.7 Use Case Model

3.7.1 HOD Control Notification Use Case

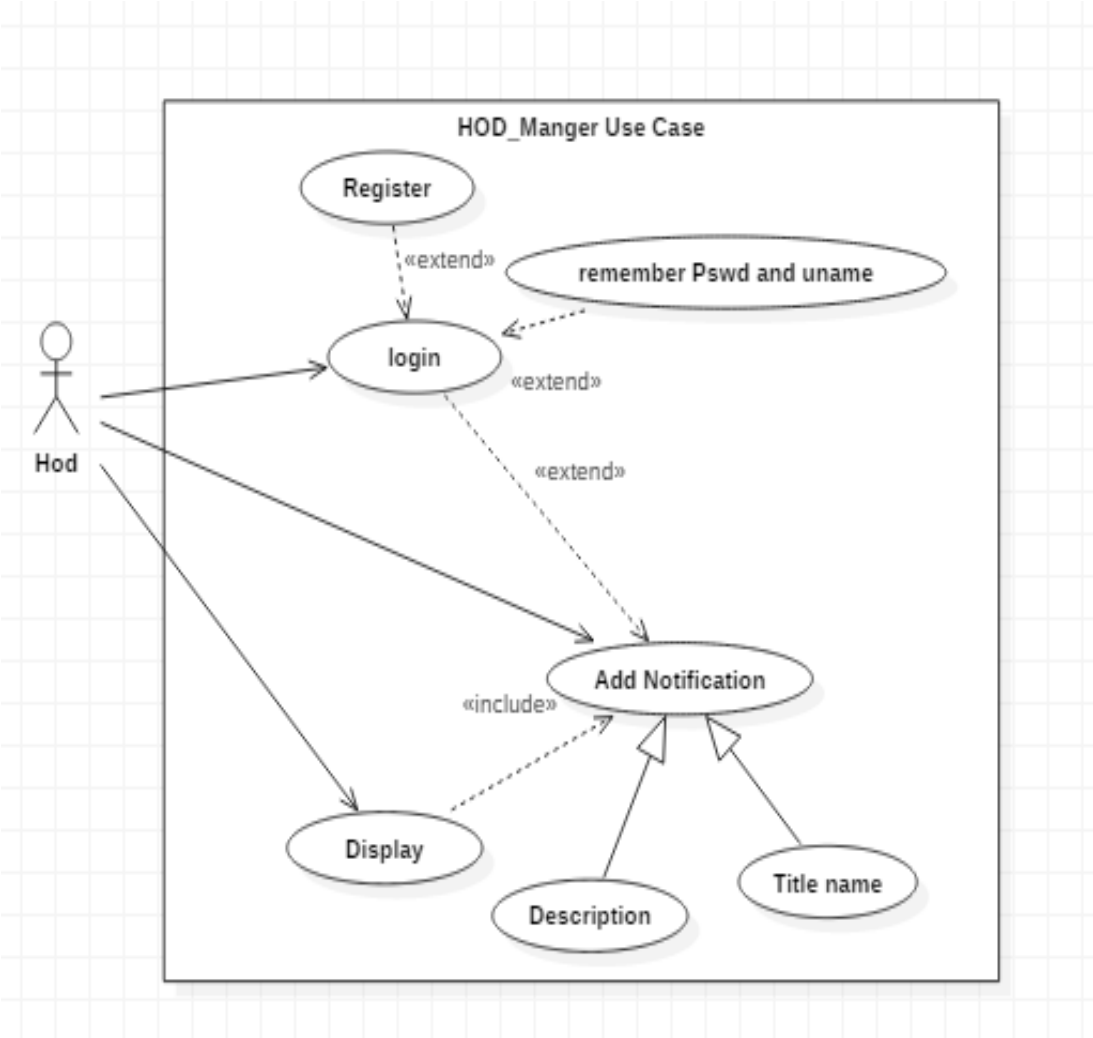

Figure 1: Hod Use Case Manager

3.7.2 Coordinator Use case

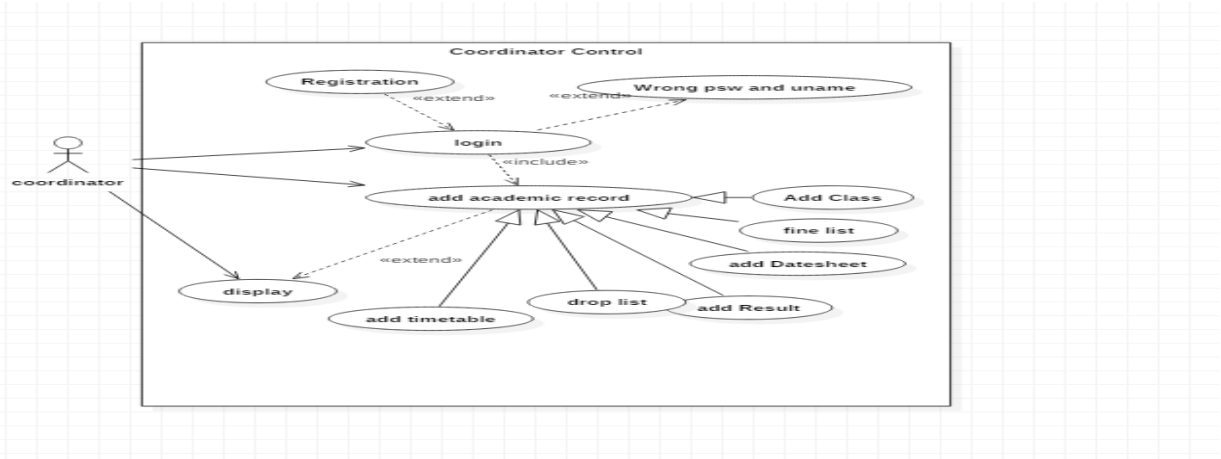

Figure 2: Coordinator Manager Use Case

3.7.3 Event Manager Use Case

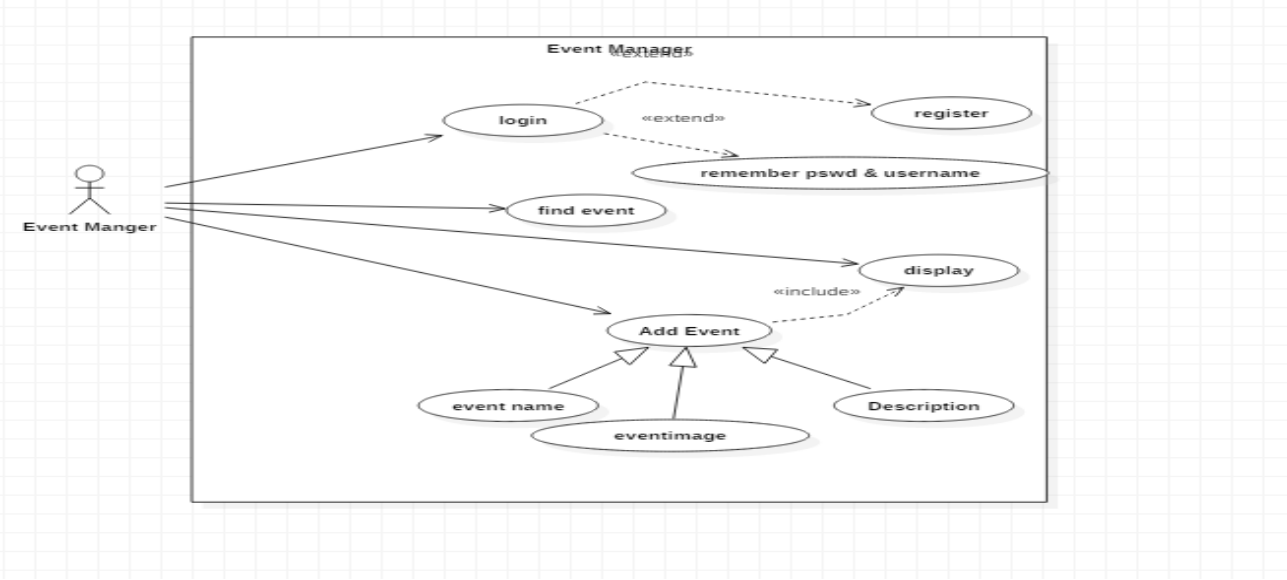

Figure 3: Event Manager Use Case

3.7.4 Admin Examination Control Use Case

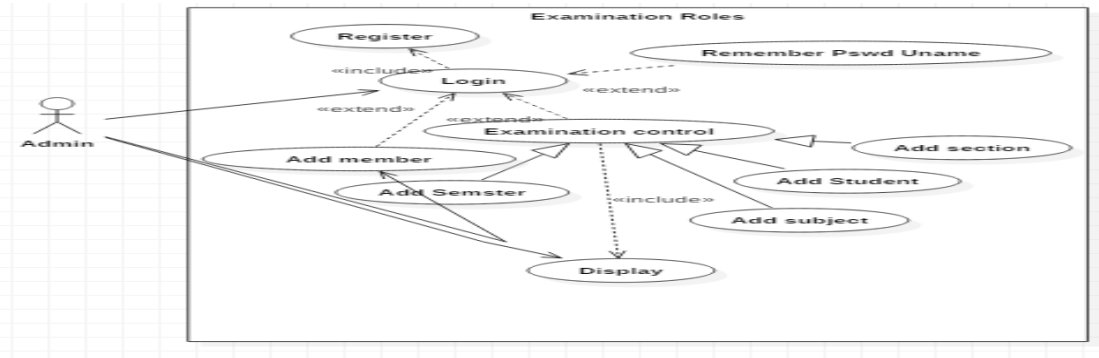

Figure 4: Admin as Examination control Use Case

3.7.5 Fully Dressed-Up Use Case:

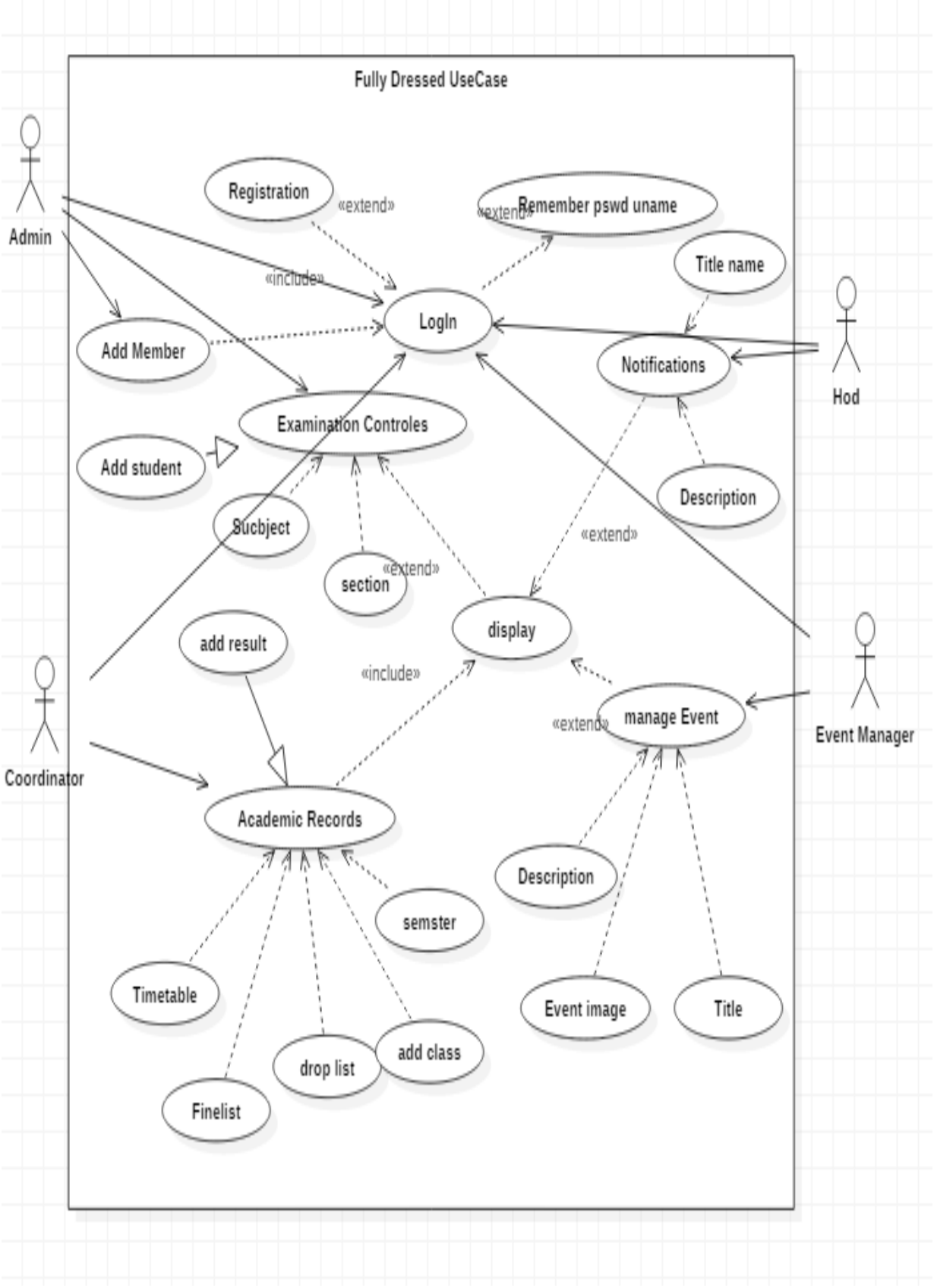

Figure 5: Fully Dressed Up Use Case

3.8 Tables of Use Cases

3.8.1 Create Account

Identifier	Create an account
Purpose	Use case enables the actors to create an account for using services provided by the application.
Priority	High
Pre-conditions	Application page must be open
Post-conditions	After the successful execution of use case request will send to the administrator for verification.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User provides a valid username, valid email address and password.	System validates the information, records information submitted by the user and sends the request to admin for verification.

Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	User attempts to submit invalid information e.g. invalid email address, invalid profile information, two different password, invalid password length etc.	System upon performing validation checks returns appropriate error messages and request for retry.

3.8.2 Login (Admin, Coordinator, HOD, Event Manager)

Identifier	Login
Purpose	Use case enables user to login.
Priority	High
Pre-conditions	User must be registered as admin, HOD, Coordinator and Event manager.
Post-conditions	If user provides valid username or password then user has logged in and is able to access the system.
	If user does not provide valid username and password then system displays an error message.

Typical Course of Action

S#	Actor Action	System Response
1	User provides valid username and password.	
		System authenticates the user and logs the user in.

Alternate Course of Action

S#	Actor Action	System Response
1	User has provided invalid username and password.	System displays an error message.
2	User left any password or username field empty.	System displays error message and request to fill that space.

3.8.3 Logout (Admin, Coordinator, HOD, Event Manager)

Identifier	Logout
Purpose	Use case enables the user to logout
Priority	Medium
Pre-conditions	User Must be login
Post-conditions	After the successful execution of use case, the user will logout

Typical Course of Action

S#	Actor Action	System Response
1	User clicks on logout.	System will logout.

3.8.4 Add Admin

Identifier	Add Admin
Purpose	Use case enables the Administrator to add an Admin.
Priority	High
Pre-conditions	Administrator Must Login
Post-conditions	After the successful execution of use case, Admin is added.

Typical Course of Action

S#	Actor Action	System Response
1	User clicks o Add Admin button.	
		System presents the user with Admin entry form.
2	User enters Admin details and submits.	
		System creates a new Admin

Alternate Course of Action

S#	Actor Action	System Response
1	User enters the details of the Admin which is already present.	System displays error message that Admin is already present.
2	User left some empty field in the entry form.	System displays error message and request to fill that space.

Add Course

Identifier	Add course
Purpose	Use case enables the user to add course.
Priority	Medium
Pre-conditions	User Must be login as Administrator
Post-conditions	After the successful execution of use case, the user will add the course.

Typical Course of Action

S#	Actor Action	System Response
1	User clicks on add course.	
		Systems will open the add course menu.
2	User enters the required information and submit it.	
		System will add the course.

3.8.5 Delete Course

Identifier	Delete course	
Purpose	Use case enables the user to delete course.	
Priority	Medium	
Pre-conditions	User Must be login as Administrator	
Post-conditions	After the successful execution of use case, the user will delete the course.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User clicks on view courses.	
		Systems open the view courses page.
2	User clicks on delete button.	
		System will display the confirmation message that you want to delete that particular course.
3	User will click on Ok to delete that course	
		System will delete that course.

3.8.6 Update Course

Identifier	Update course
Purpose	Use case enables the admin to update course.
Priority	Medium
Pre-conditions	User Must be login as administrator
Post-conditions	After the successful execution of use case, the admin will able to update that particular course.

Typical Course of Action

S#	Actor Action	System Response
1	Admin clicks on view courses.	
		System will open the course page.
2	Admin clicks on update button to update the particular course.	
		System will open the course page.
3	Admin update the course and clicks on update button.	
		System will save the updated course.

3.8.7 Add Student

Identifier	Add Student	
Purpose	Use case enables the Administrator to add a student.	
Priority	High	
Pre-conditions	Administrator Must Login	
Post-conditions	After the successful execution of use case, student is added.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Admin clicks on Add a Student button.	System presents the admin with student entry form.
2	Admin enters student details and submits.	System creates a new student.
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Admin enters the details of the student which is already present and registered.	System displays error message that Student is already exists.
2	Admin left some empty fields in the registration form.	System displays error message and request to fill that space.

3.8.8 Update Student

Identifier	Update student
Purpose	Use case enables the admin to update student.
Priority	Medium
Pre-conditions	User Must be login as administrator
Post-conditions	After the successful execution of use case, the admin will able to update that particular student profile.

Typical Course of Action

S#	Actor Action	System Response
1	Admin clicks on view students.	
		System will open the students page.
2	Admin clicks on update button to see the profile of that particular student.	
		System will open the student profile.
3	Admin update the profile and clicks on update button.	
		System will save the updated profile

3.9 Screen Shots

3.9.1 Adding Notification Dashboard

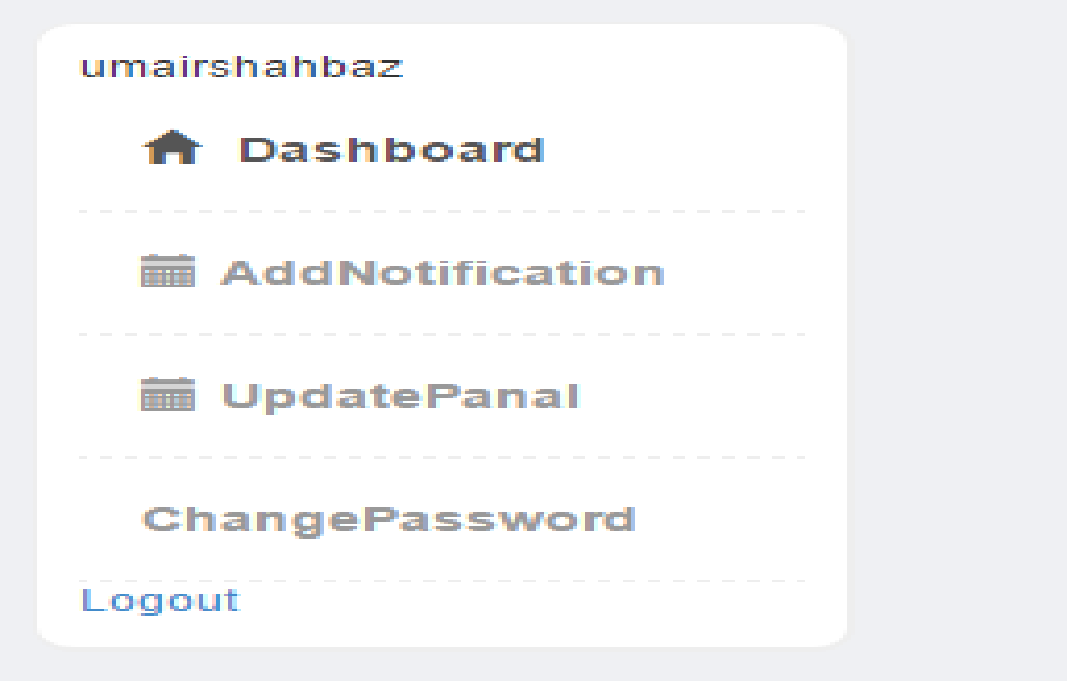

Figure 6: Notification Dashboard

3.9.2 Add Events:

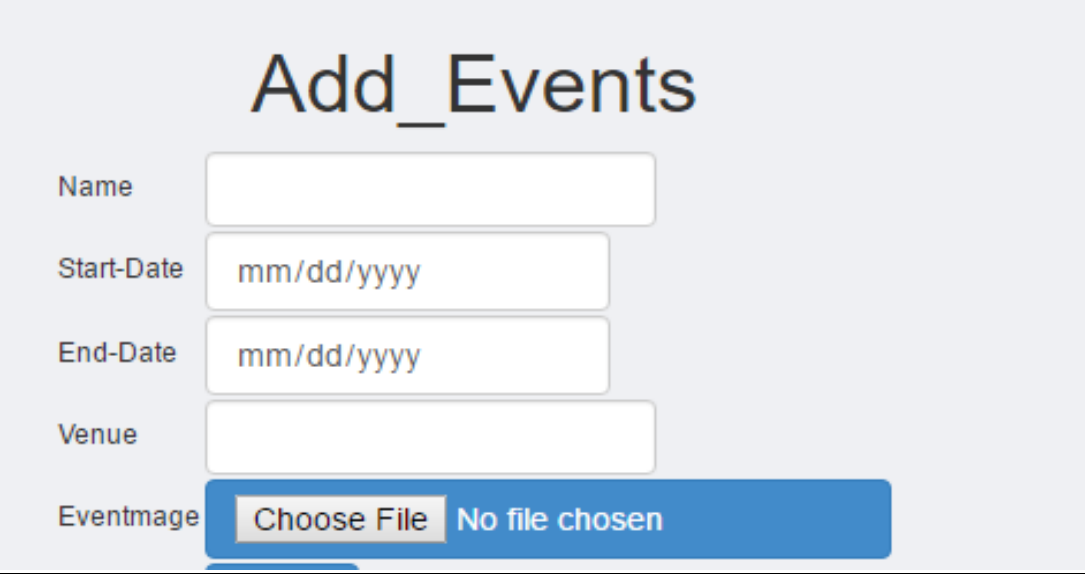

Figure 7:Add Events

3.9.3 Adding Date sheet

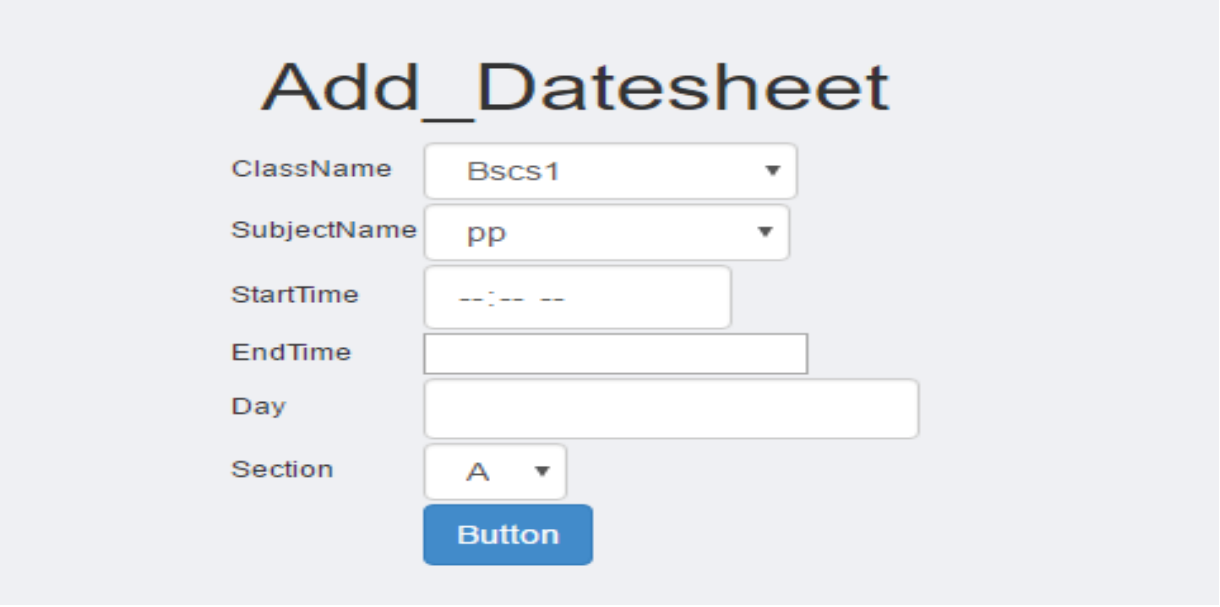

Figure 8: Add Date Sheet

3.9.4 Adding Date Sheet Dashboard

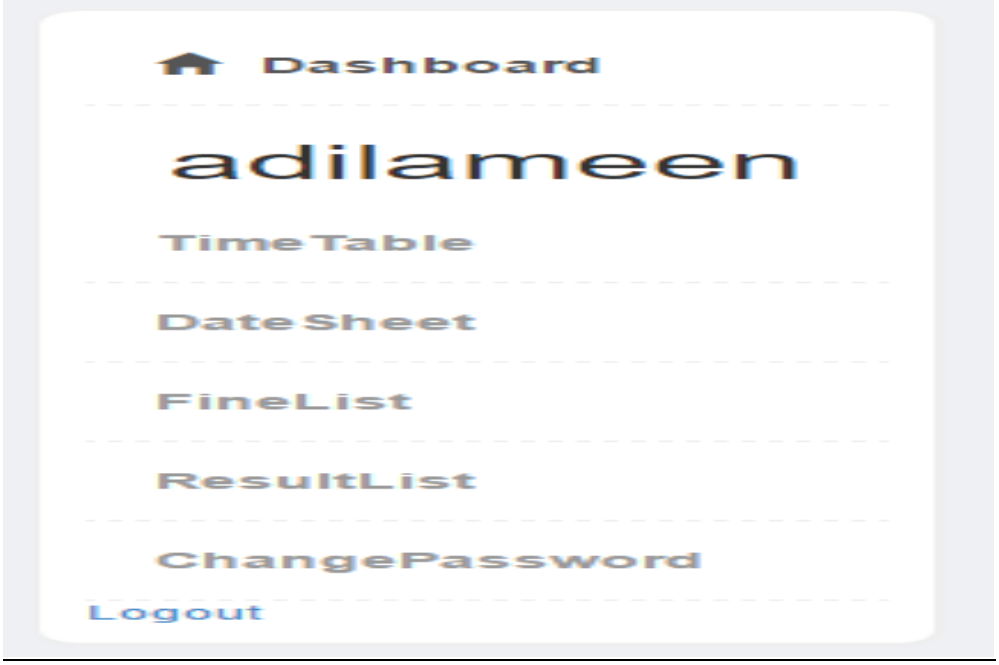

Figure 9: Add Date Sheet Dashboard

3.9.5 Main Dashboard

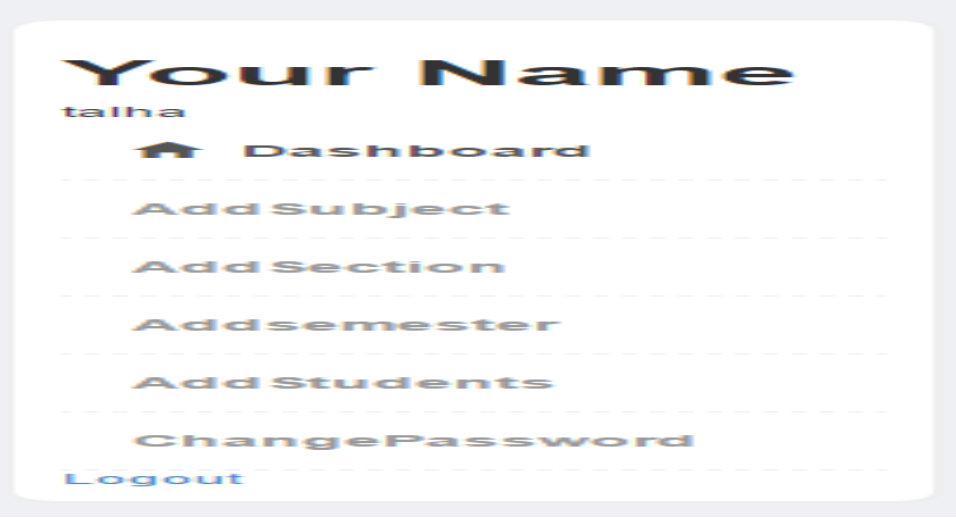

Figure 10 : Main Dashboard

Sign-In Page:

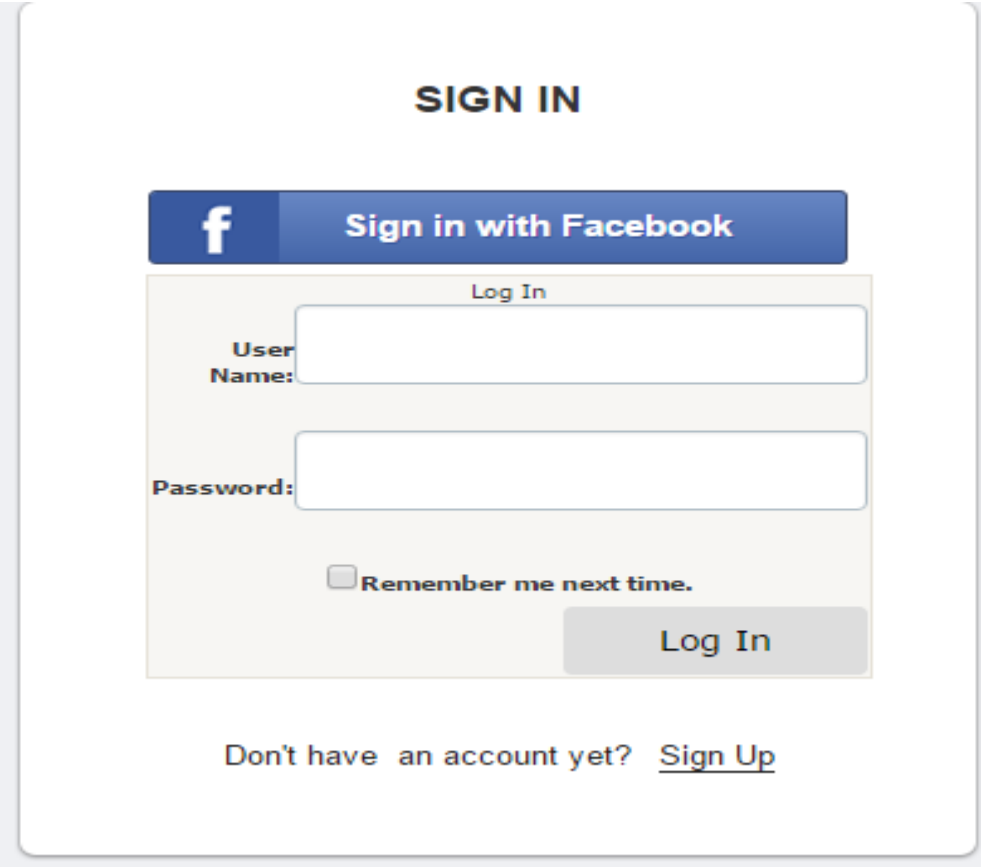

Figure 11: Sign-In/Log In Page

Chapter No: 4

Design

4. Designing Phase

4.1 ER Diagram

Figure 12: Entity Relationship Diagram

4.2 **Database Designing**

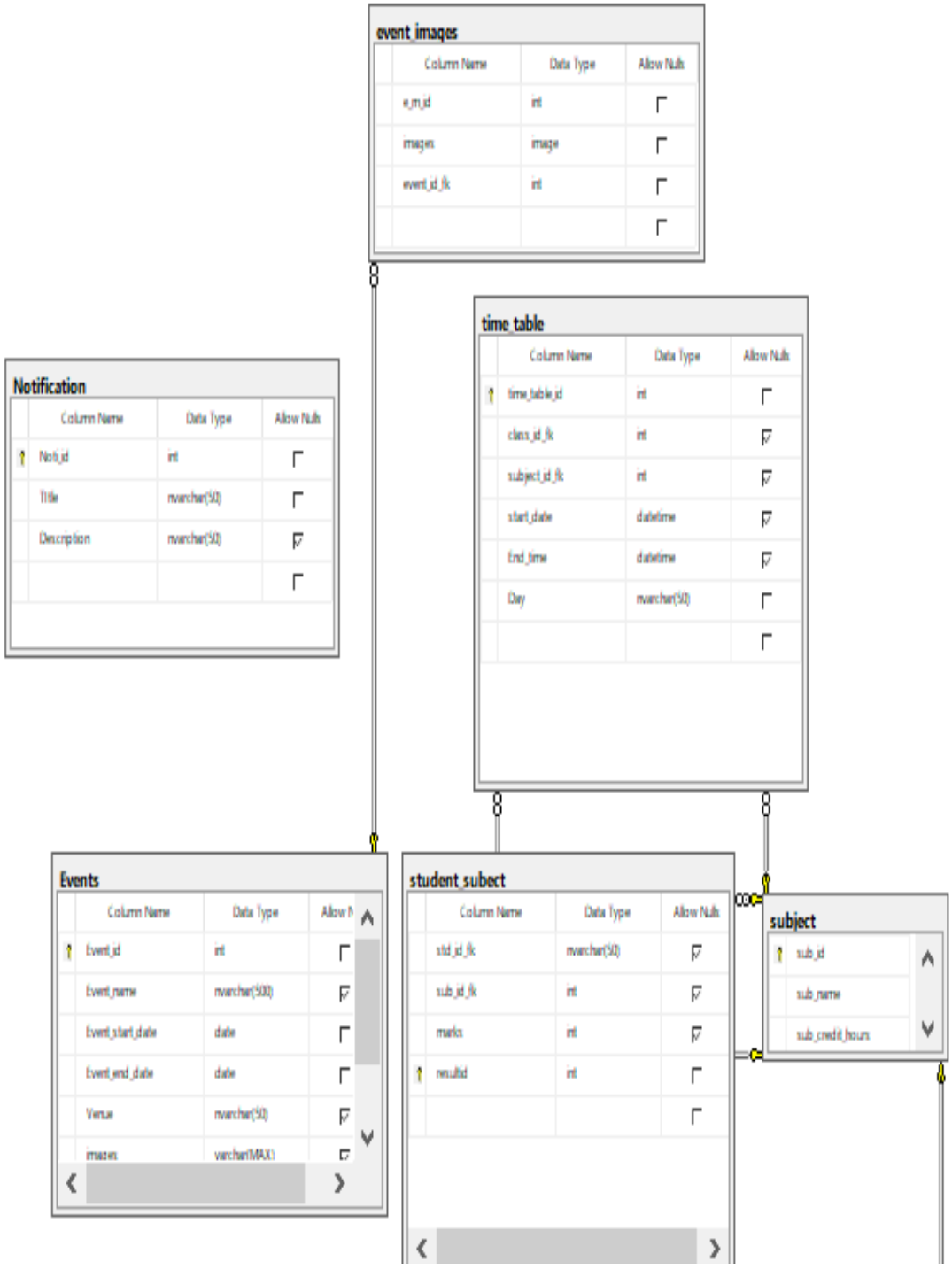

Figure 13: Database Designing

4.3 **Class Diagram**

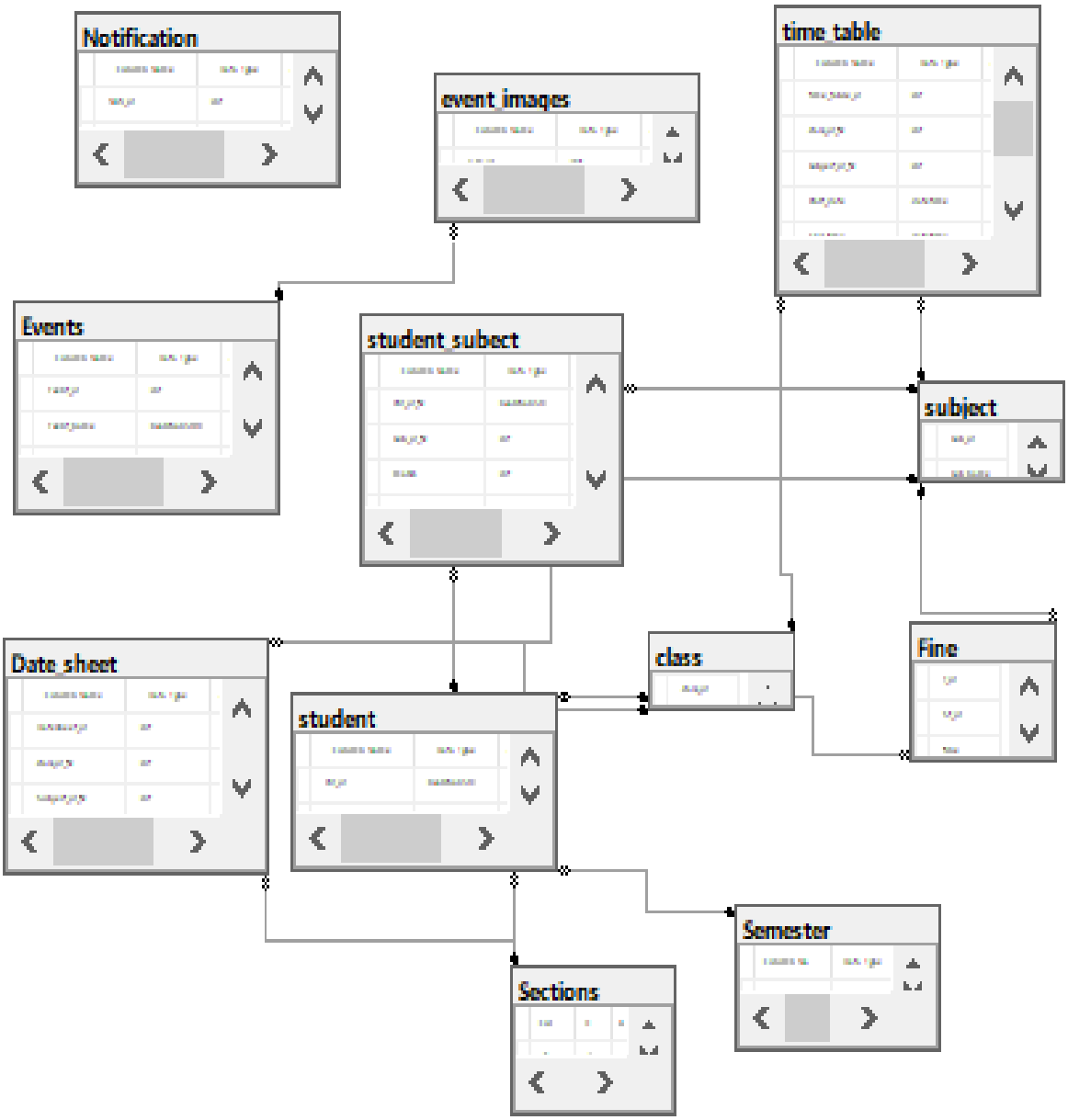

Figure 14: Class Diagram

4.4 Activity Diagrams

4.4.1 Login Activity Diagram

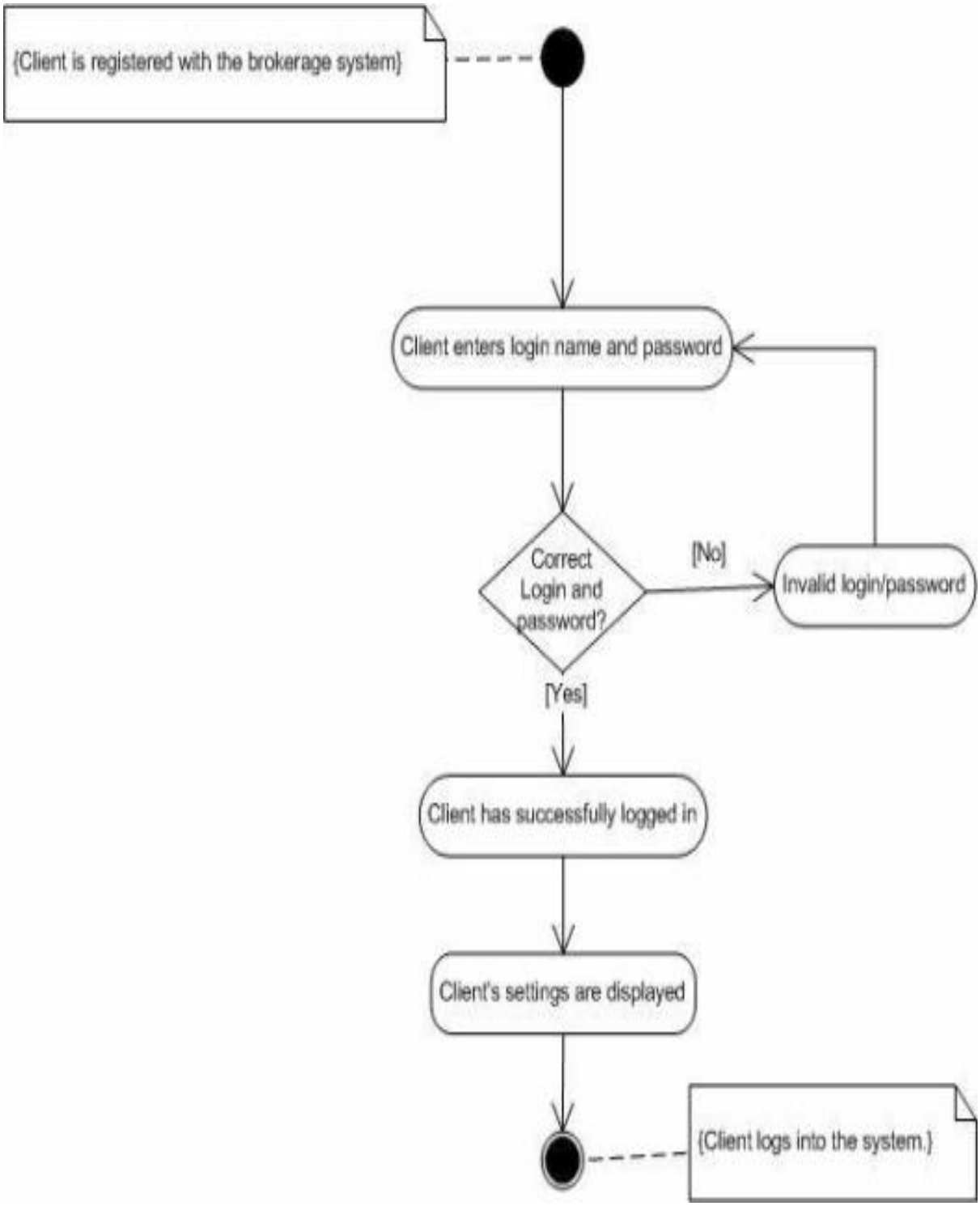

Figure 15: Log In Activity Diagram

4.4.2 Event Manager Diagram

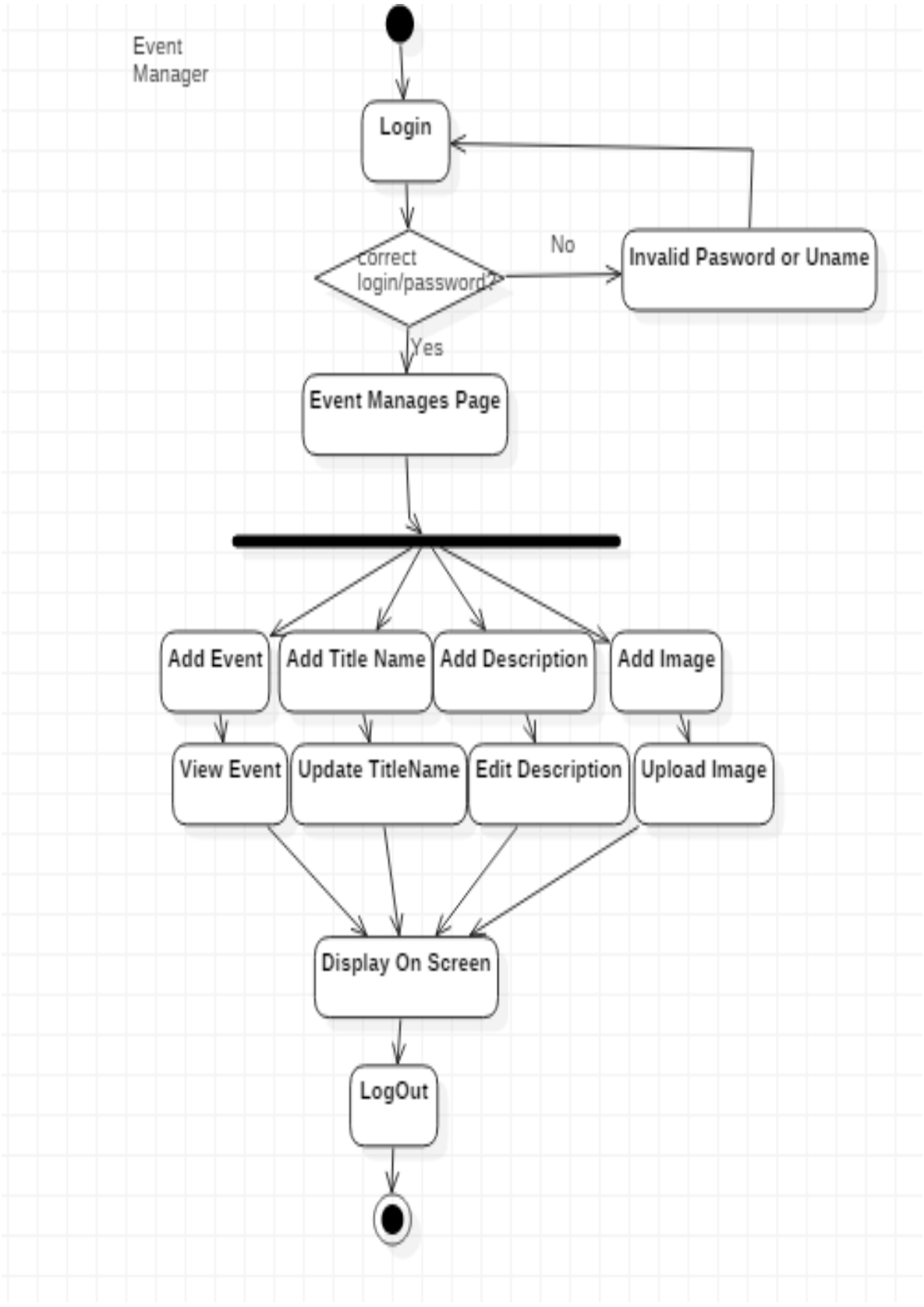

Figure 16: Event Manager Activity Diagram

4.4.3 Coordinator Manager

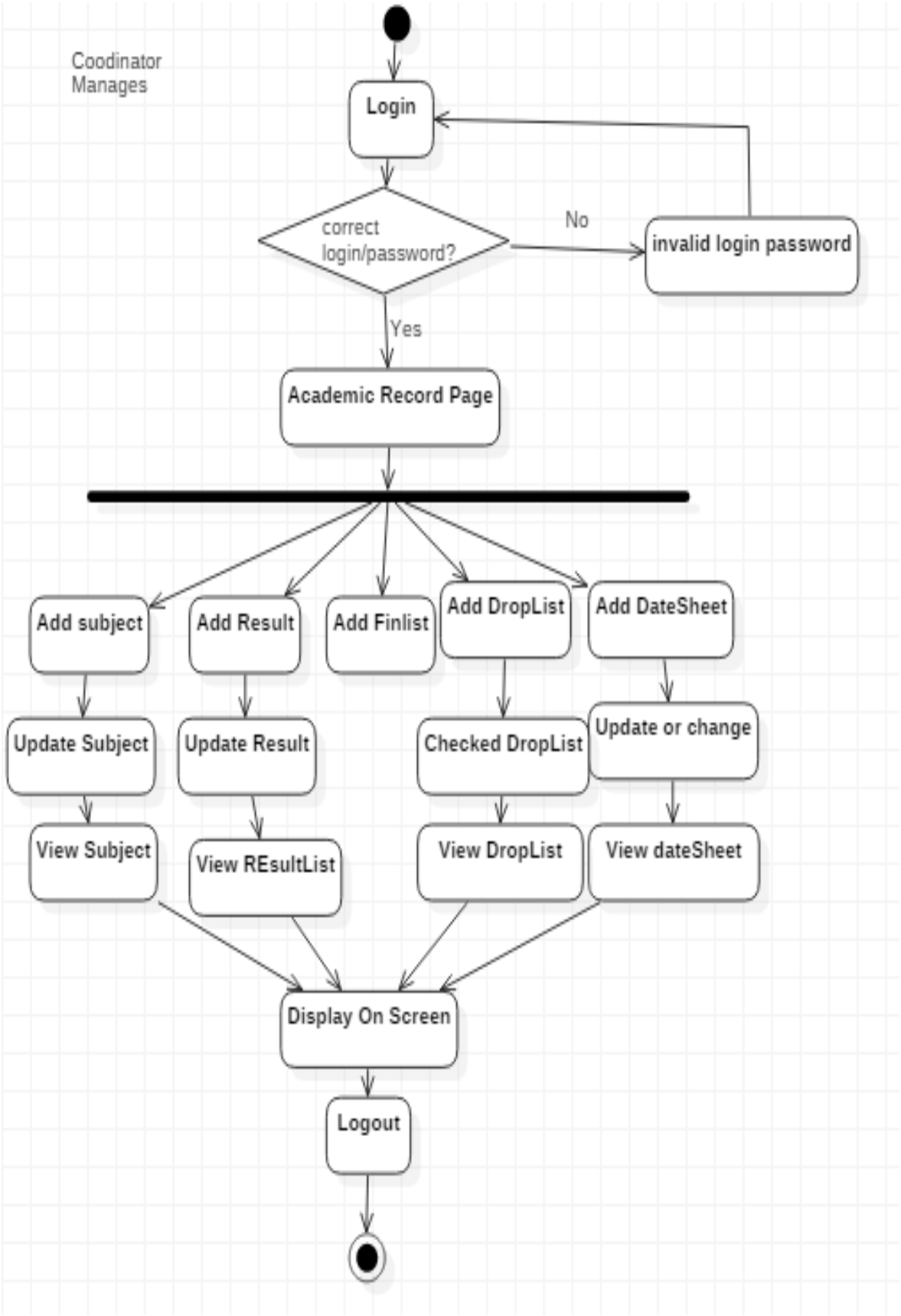

Figure 17: Coordinator Manager Activity Diagram

4.4.4 HOD Manager

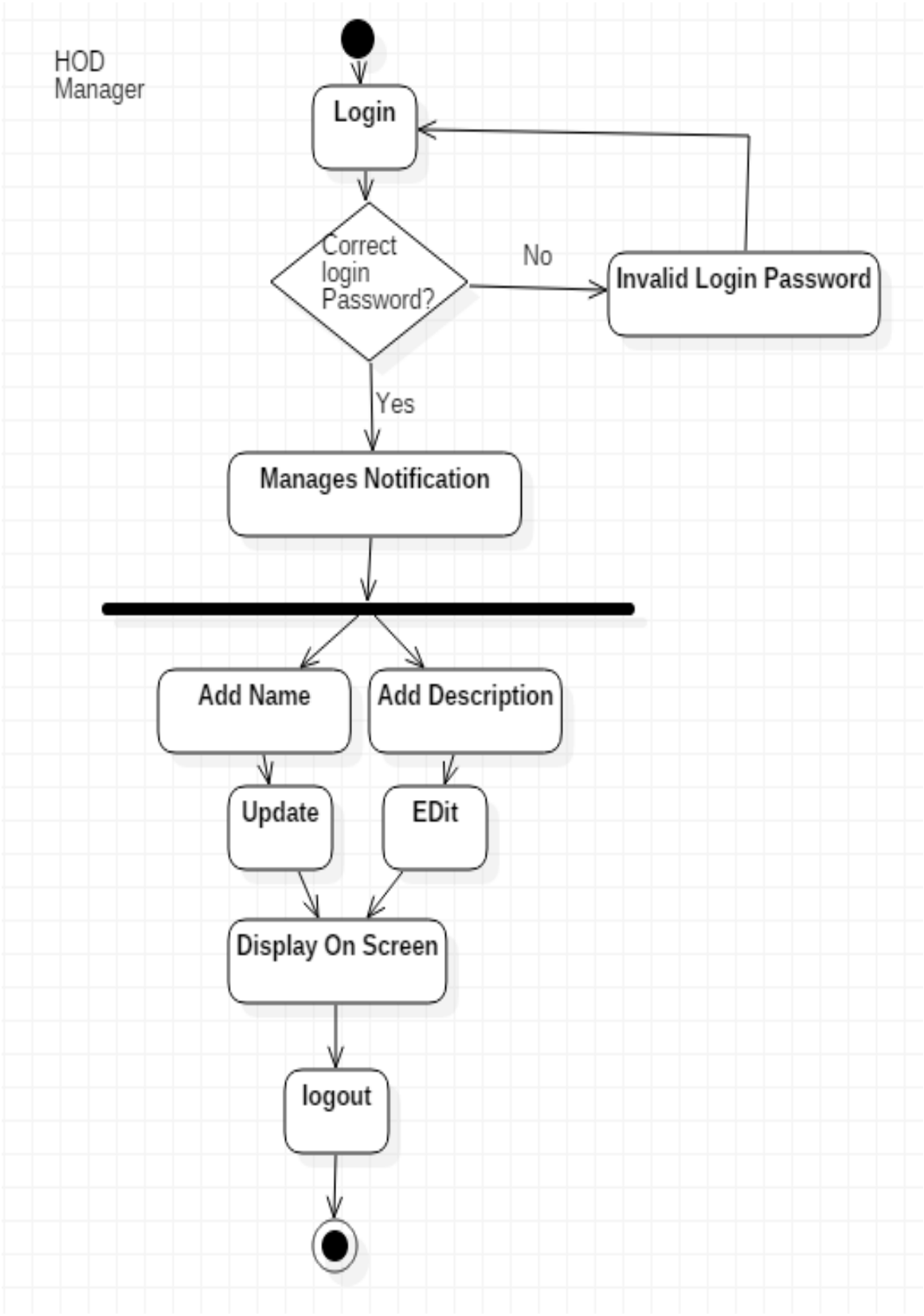

Figure 18: HOD Manager Activity

4.4.5 Admin as Examination Control

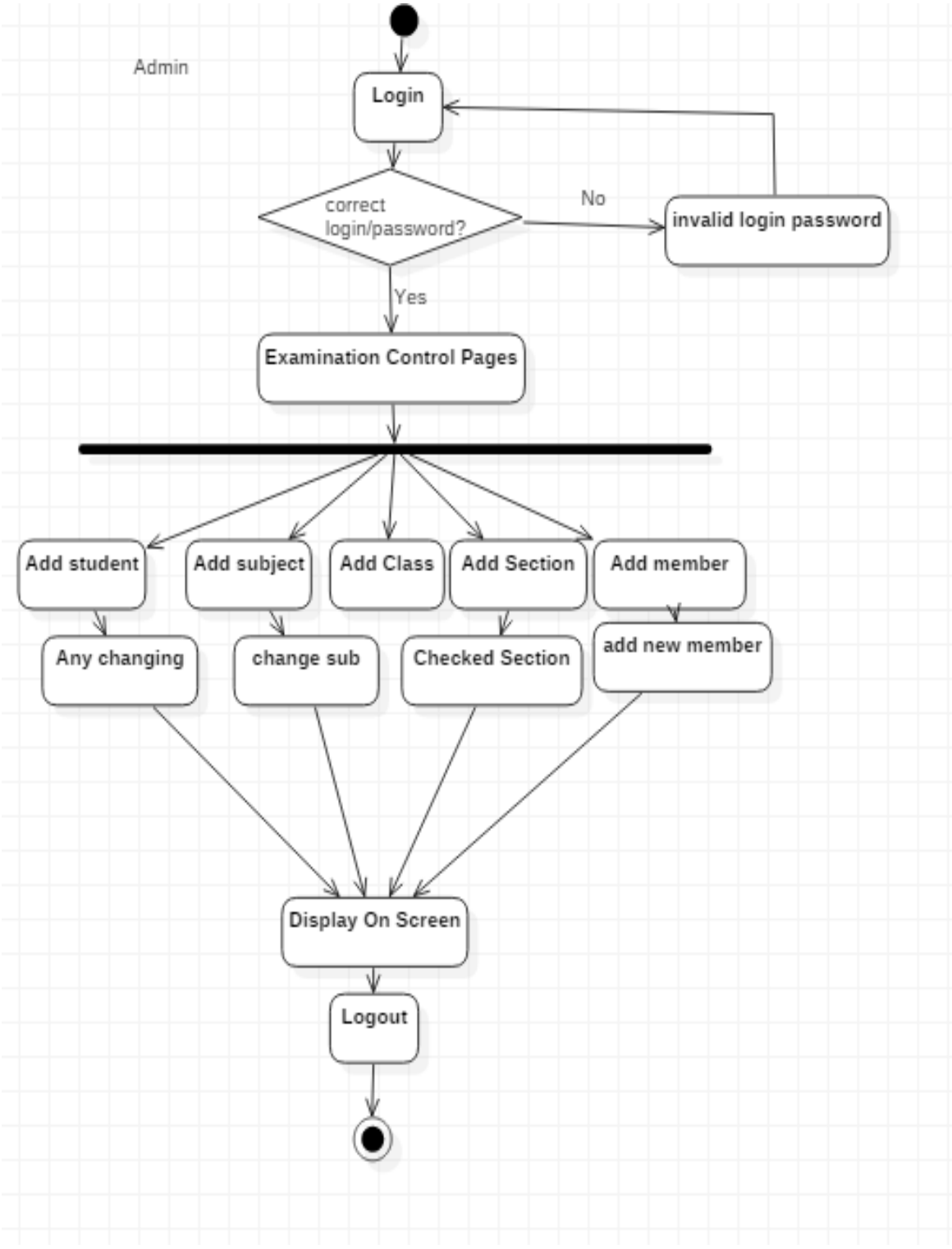

Figure 19: Admin as Examination Control activity

4.5 DFD Diagrams

4.5.1 DFD Diagram Level 0

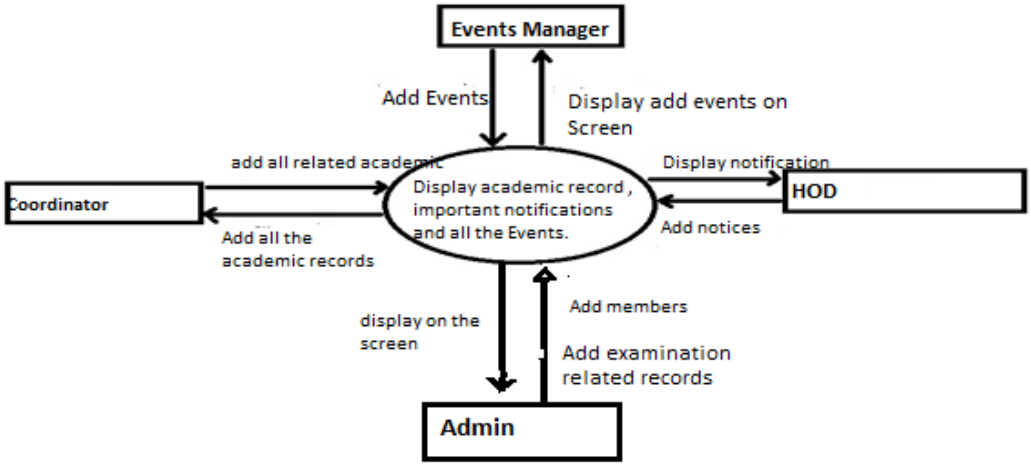

Figure 20: DFD Diagram Level 0

4.5.2 DFD for Coordinator

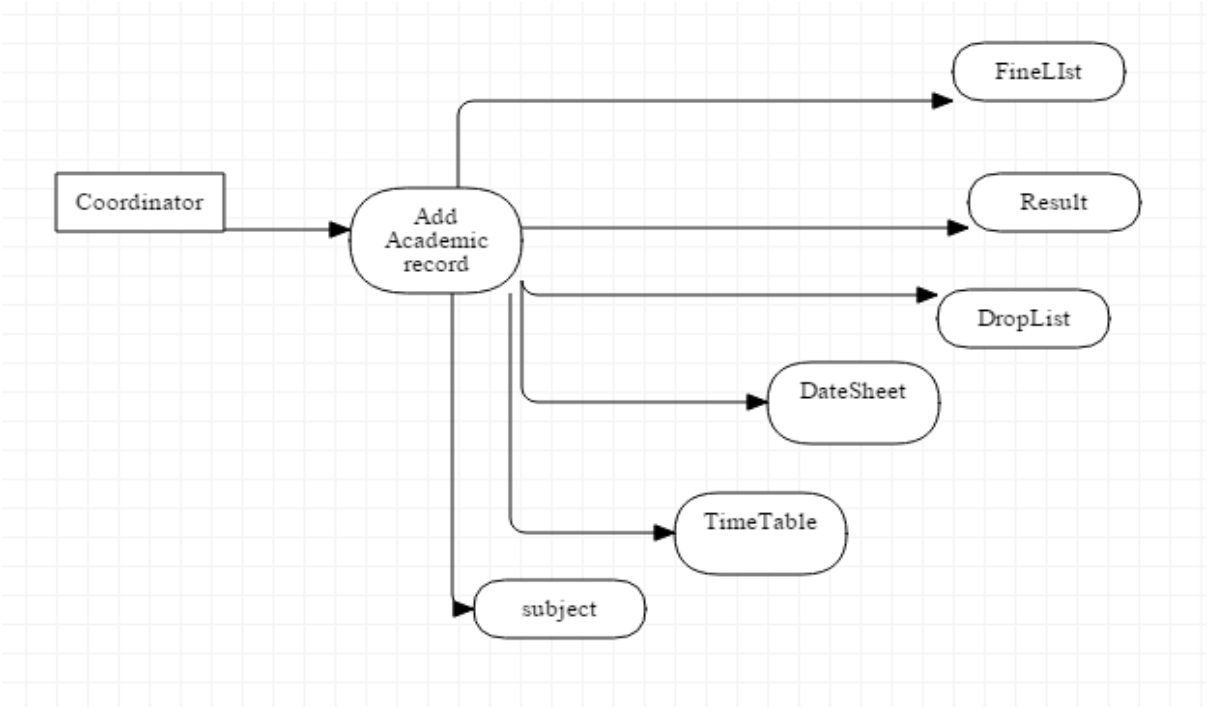

Figure 21 : DFD For Coordinator

4.5.3 DFD for Event Manager

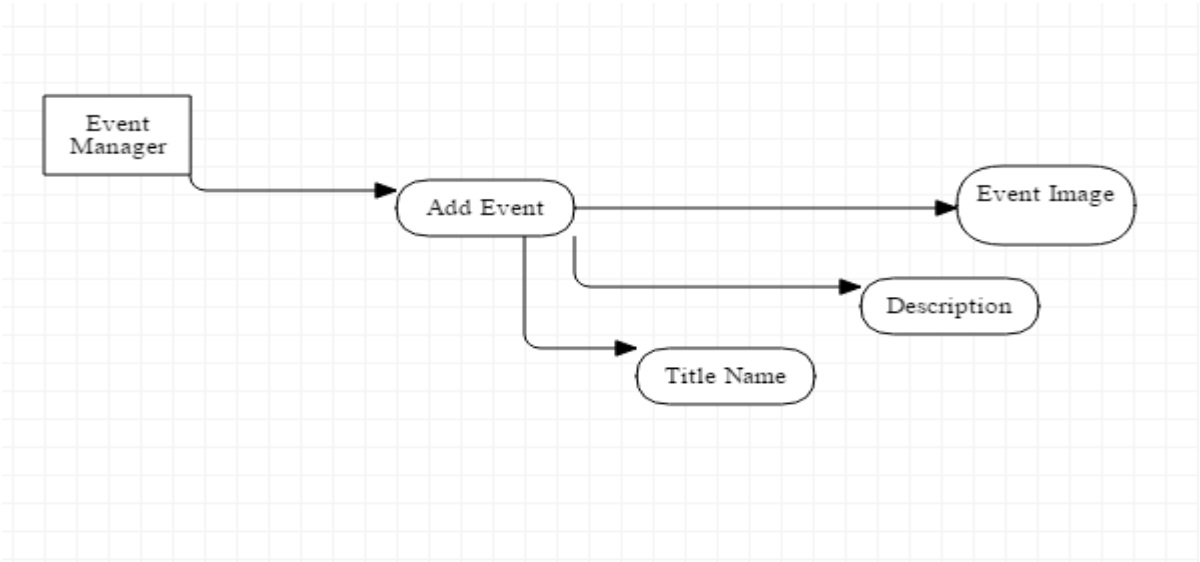

Figure 22 : DFD for Event Manager

4.5.4 DFD for HOD

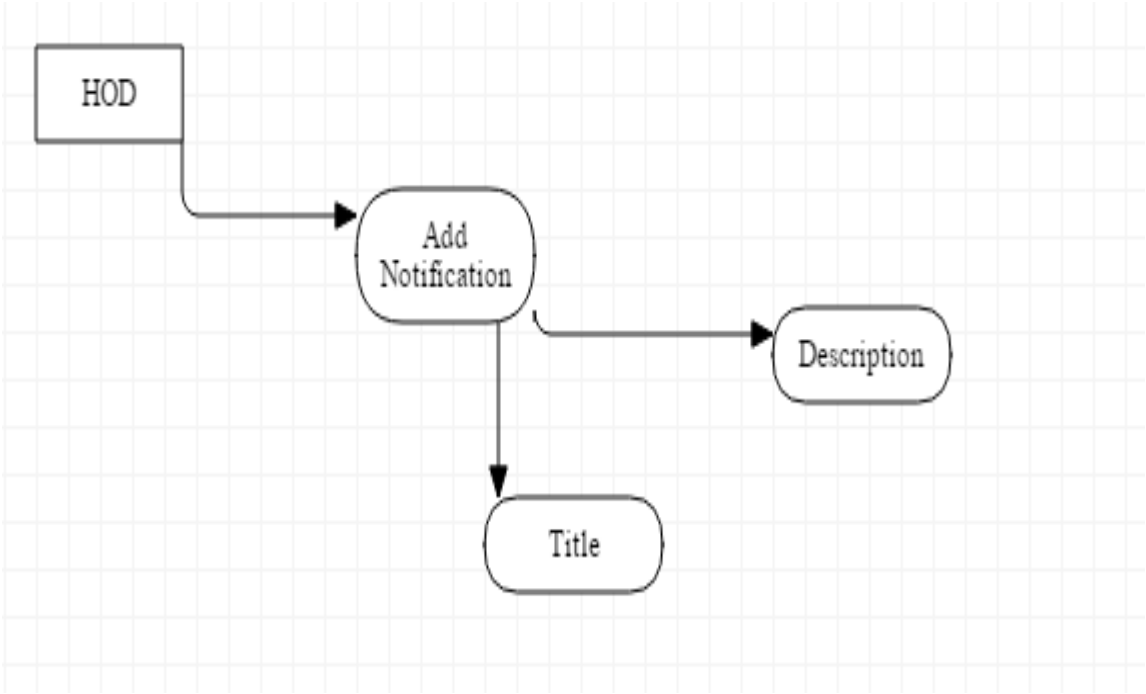

Figure 23: DFD for HOD

4.5.5 Data Flow Diagram Level 1

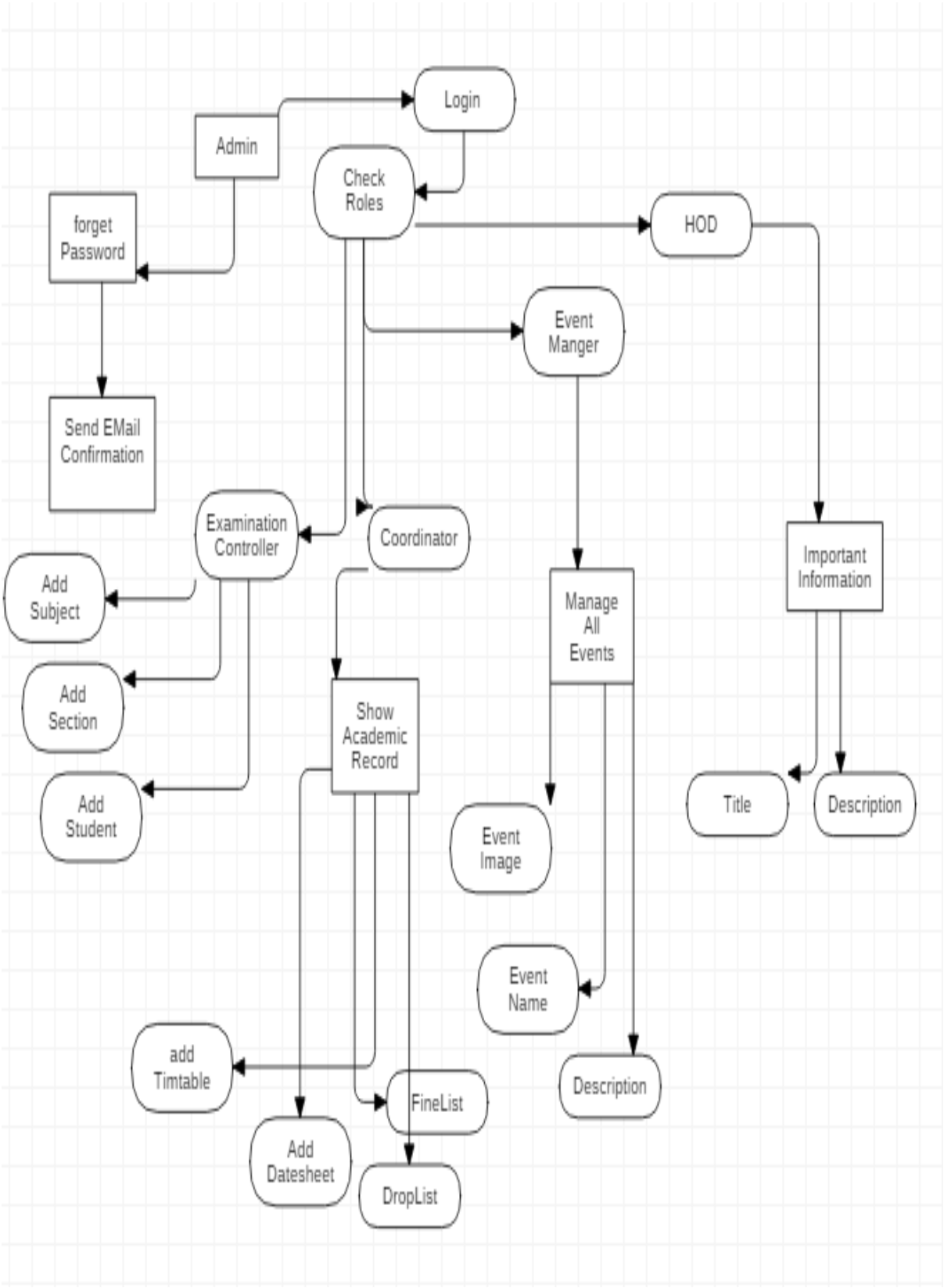

Figure 24: DFD Level 1

4.6 Sequence Diagrams

Coordinator Sequence Diagram

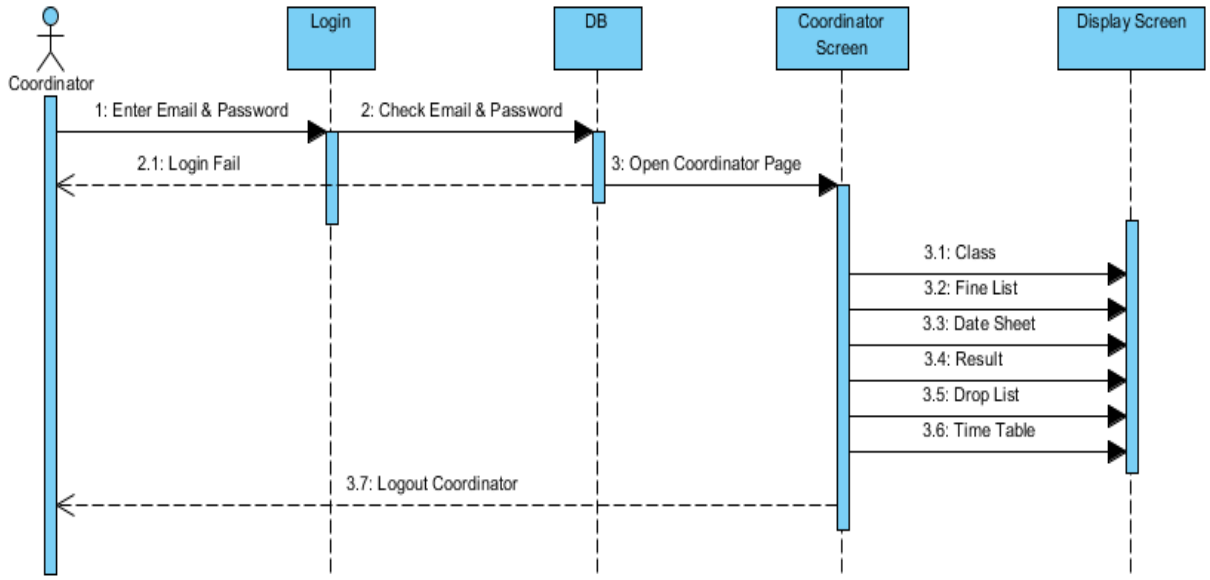

Figure 25 : Coordinator Sequence Diagram

Event Manager Sequence Diagram

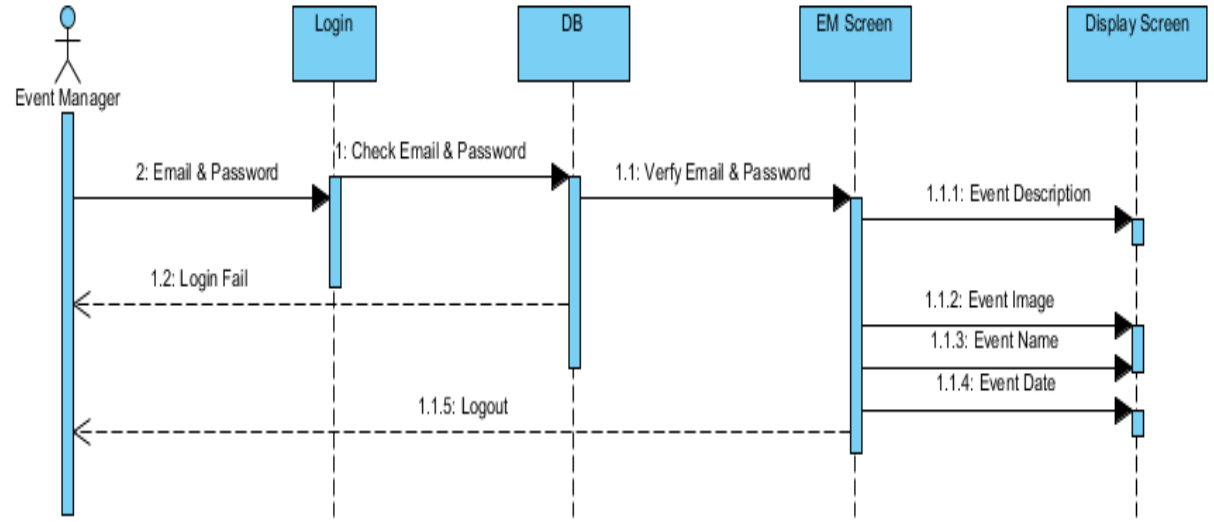

Figure 26 : Event Manager Sequence

4.6.1 Examination Control Sequence Diagram

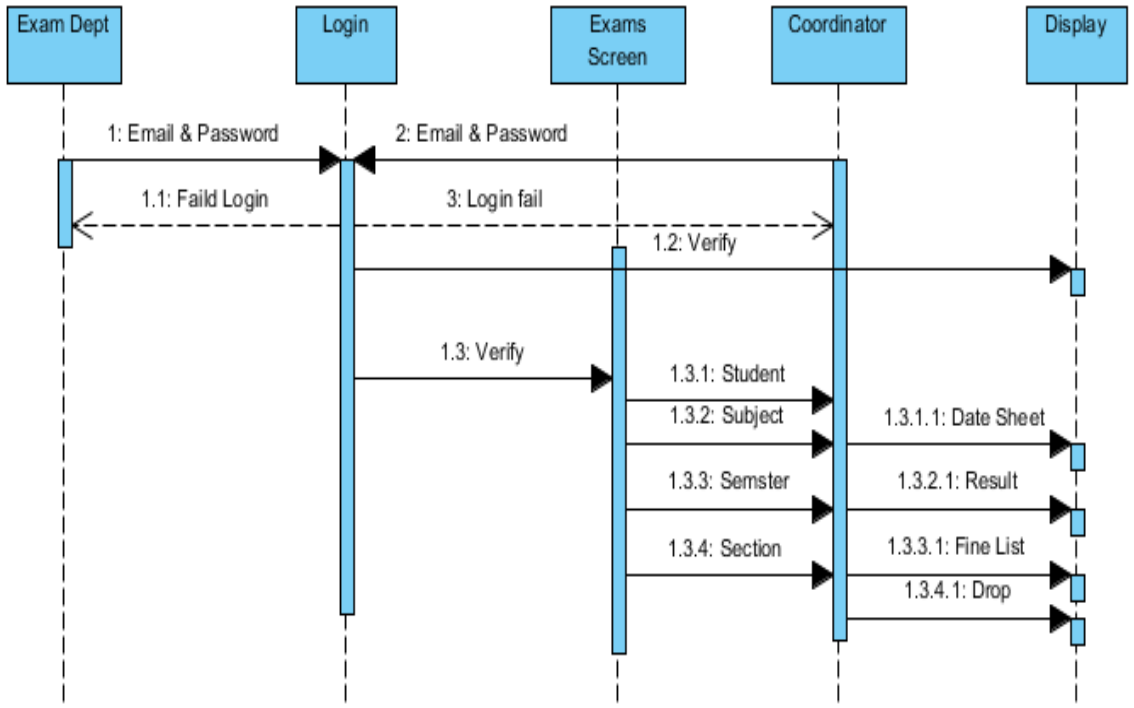

Figure 27 : Examination Control Sequence Diagram

4.6.2 HOD Sequence Diagram

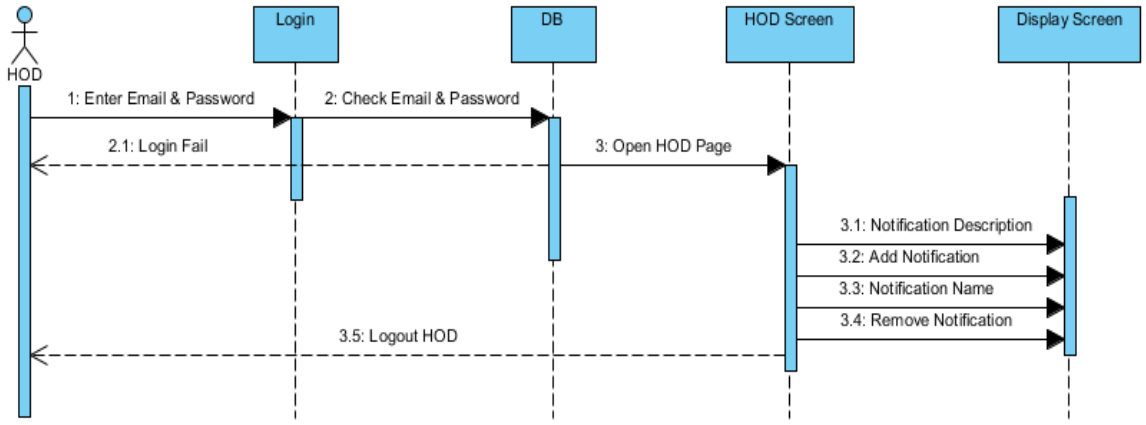

Figure 28 : HOD Sequence Diagram

4.6.3 Full Sequence Diagram

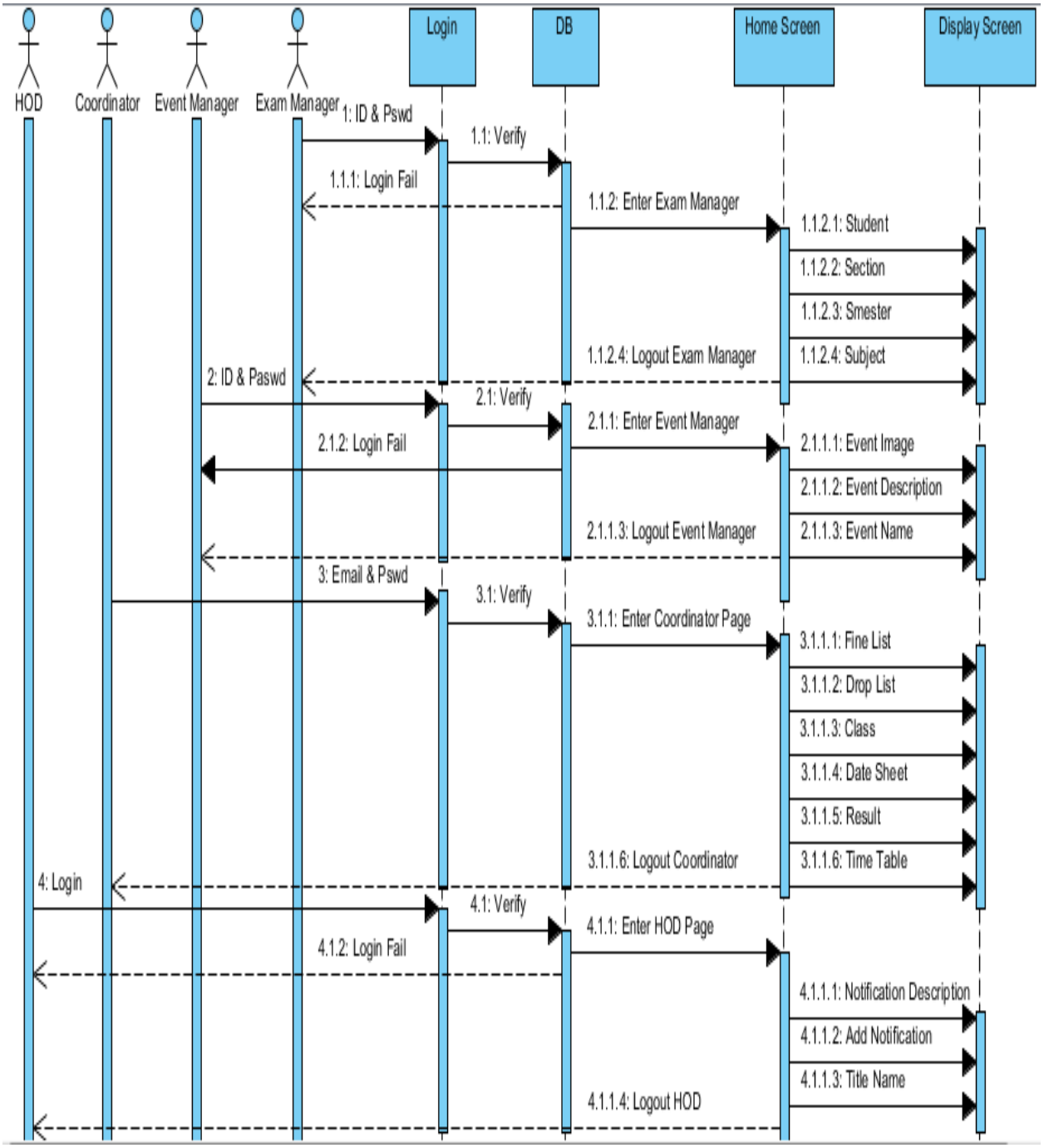

Figure 29 : Full Sequence Diagram

4.7 Communication Diagrams

4.7.1 HOD CD

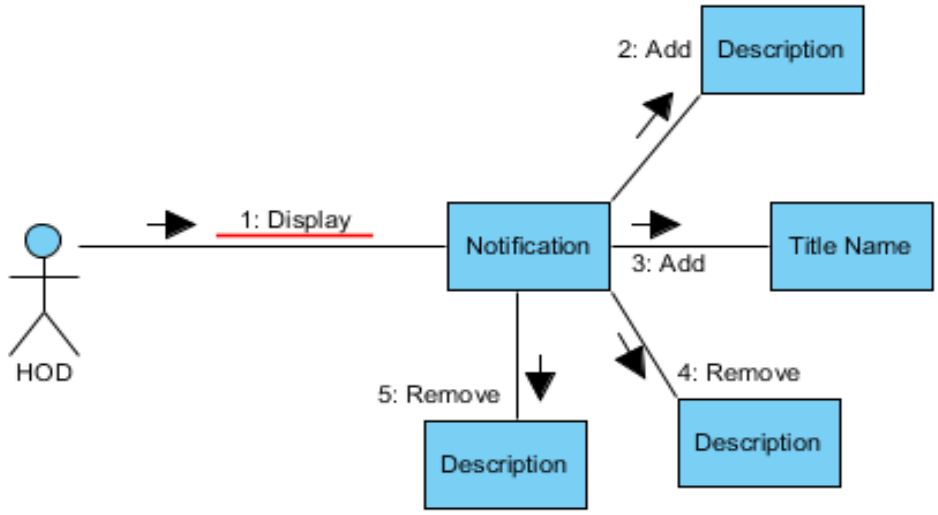

Figure 30 : HOD Communication Diagram

4.7.2 Event Manager CD

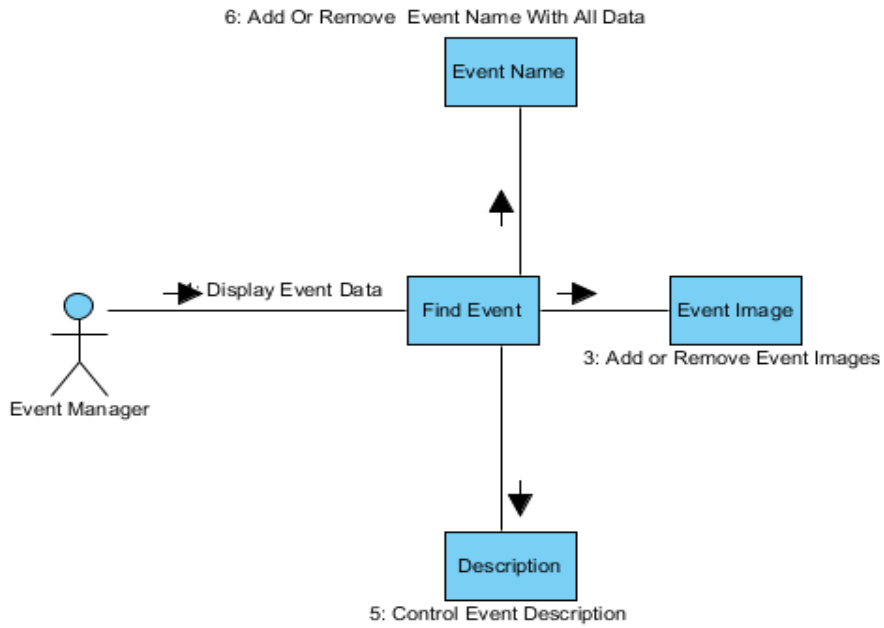

Figure 31 : Event Manager Communication Diagram

4.7.3 Coordinator CD

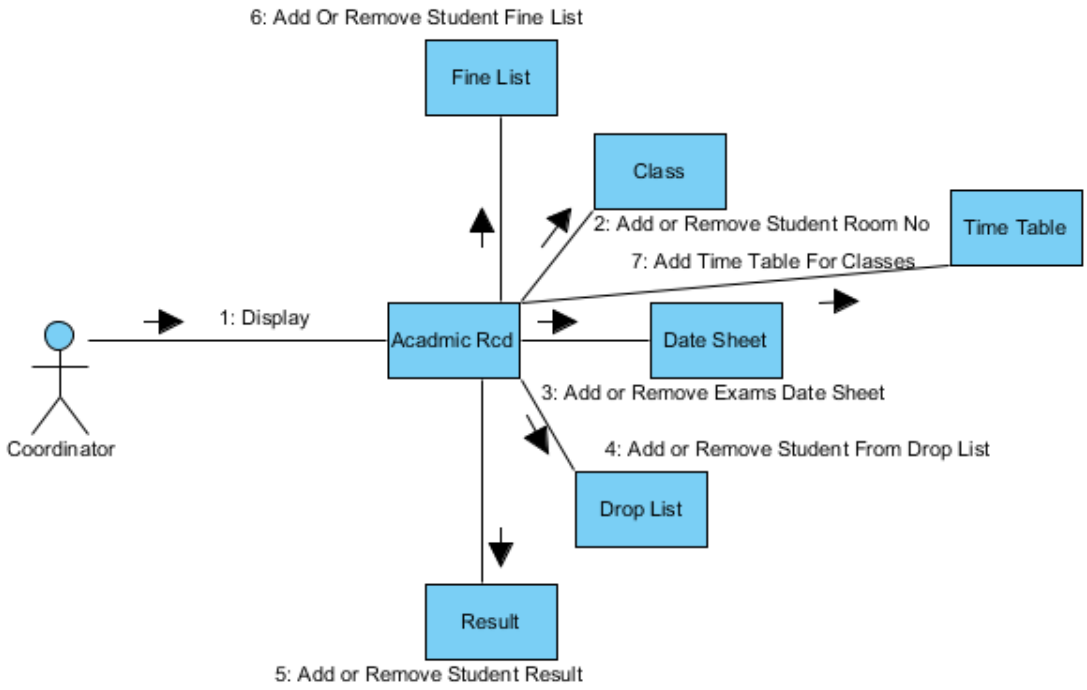

Figure 32 : Coordinator Communication Diagram

4.7.4 Admin CD

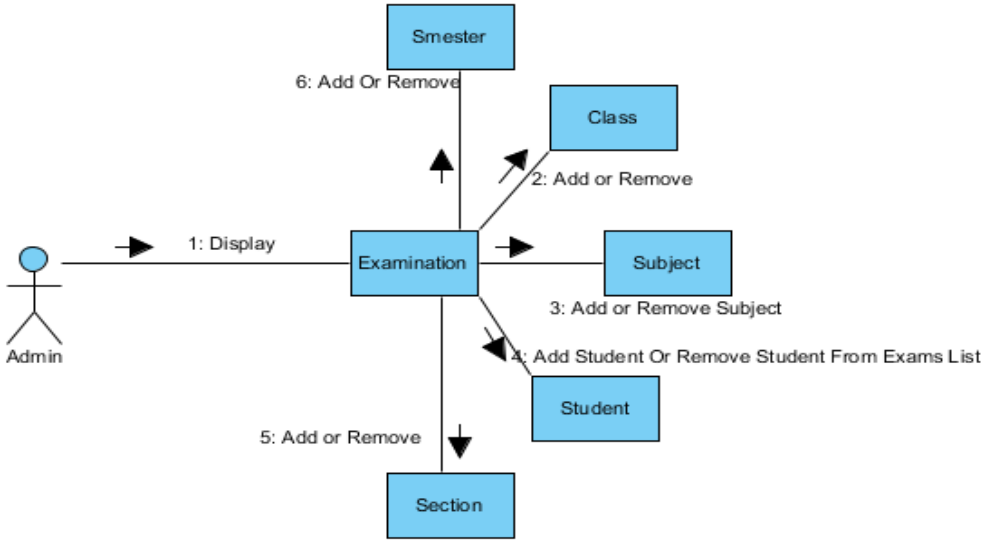

Figure 33 : Admin Communication Diagram

4.8 Component Diagram

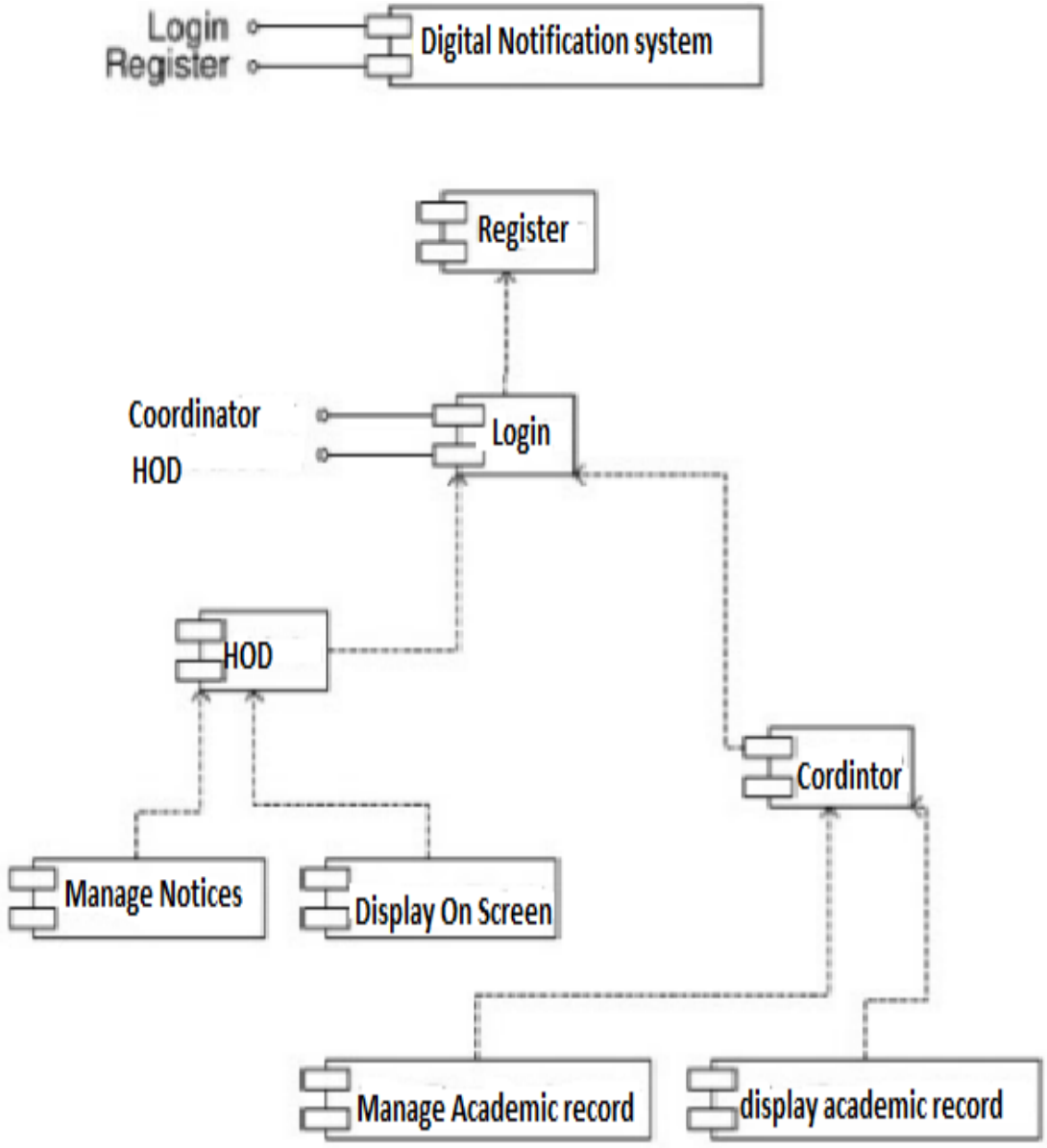

Figure 34 : Component Diagram

4.9 **Deployment Diagram**

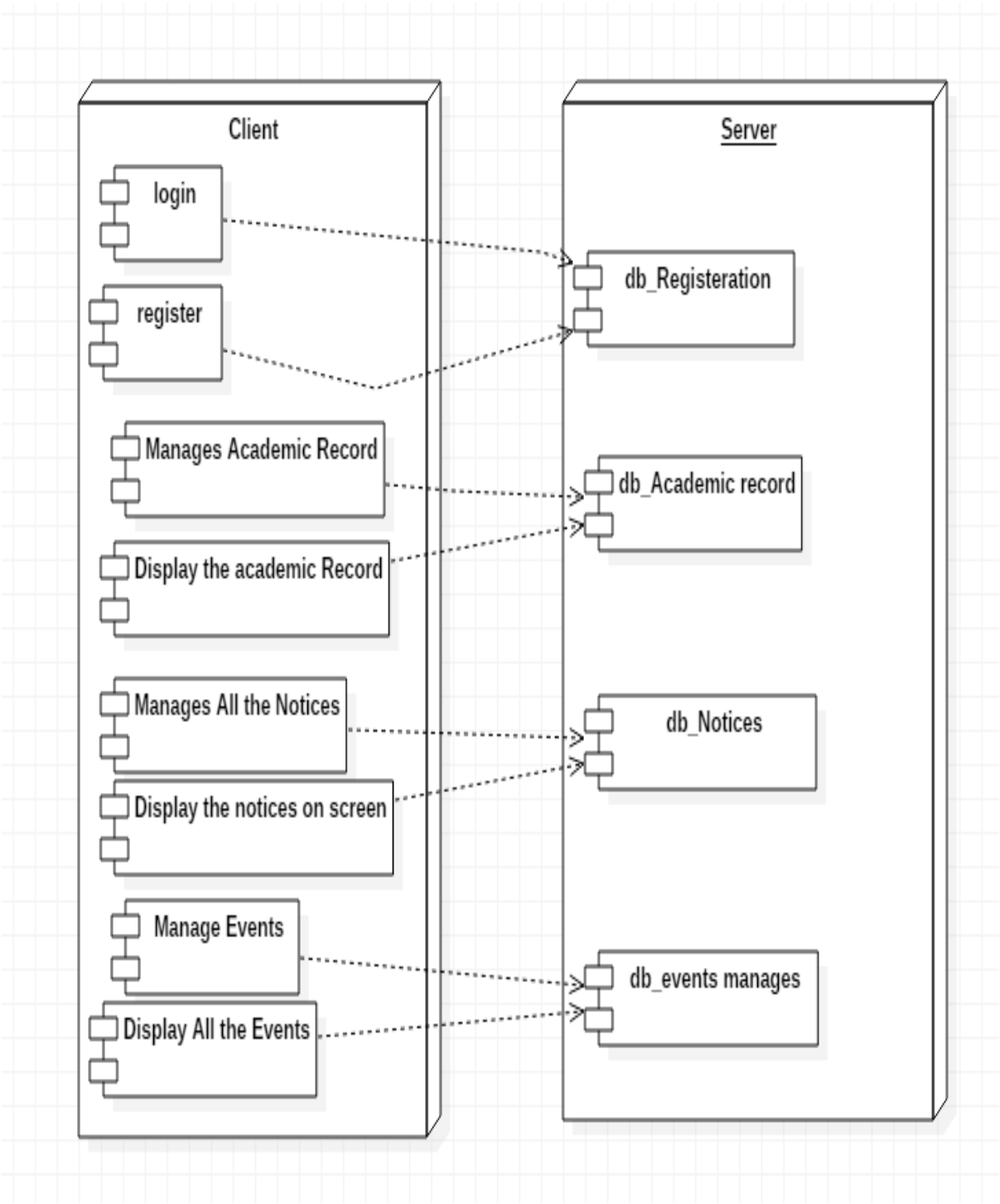

Figure 35 : Deployment Diagram

Chapter No: 5

Testing

5. Testing Phase

5.1 Test Plan

A test plan can be defined as a document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning. In software testing, a test plan gives detailed testing information regarding an upcoming testing effort, including

- Scope of testing
- Schedule
- Test Deliverables
- Release Criteria
- Risks and Contingencies

It is also be described as a detail of how the testing will proceed, who will do the testing, what will be tested, in how much time the test will take place, and to what quality level the test will be performed.

The process of defining a test project so that it can be properly measured and controlled. The test planning process generates a high level test plan document that identifies the software items to be tested, the degree of tester independence, the test environment, the test case design and test measurement techniques to be used, and the rationale for their choice.

A testing plan is a methodological and systematic approach to testing a system such as a machine or software. It can be effective in finding errors and flaws in a system. In order to find relevant results, the plan typically contains experiments with a range of operations and values, including an understanding of what the eventual workflow will be.

Test plan is a document which includes, introduction, assumptions, list of test cases, and list of features to be tested, approach, deliverables, resources, risks and scheduling. A test plan is a systematic approach to testing a system such as a machine or software. The plan typically contains a detailed understanding of what the eventual work flow will be. A record of the test planning process detailing the degree of tester independence, the test environment, the test case design techniques and test measurement techniques to be used, and the rationale for their choice.

5.2 Test Activities

Various Testing Activities are as follow:

5.2.1 Black box testing

Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

5.2.2 White box testing

This testing is based on knowledge of the internal logic of an applications code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

5.2.3 Unit testing

Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. May require developing test driver modules or test harnesses.

5.2.4 Incremental integration testing

Bottom up approach for testing i.e. continuous testing of an application as new functionality is added, Application functionality and modules should be independent enough to test separately it should be done by programmers or by testers.

5.2.5 Integration testing

Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

5.2.6 Functional testing

This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black-box type testing geared to functional requirements of an application.

5.2.7 System testing

Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

5.2.8 End-to-end testing

Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

5.2.9 Acceptance testing

Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer do this testing to determine whether to accept application.

5.2.10 Usability testing

User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing.

5.3 Test Cases

5.3.1 Test Case: 1

Identifier	TC-1
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-1
Short description	Case enables the actors to create an account for using services provided by the application.
Pre-condition(s)	Application page must be open.
Input data	Enter valid information.
Detailed steps	• User enter valid information. • Click the "Register" button.
Expected result(s)	Account created and will be verified by administrator.
Post-condition(s)	After the successful execution request will send to the administrator for verification.

Identifier	TC-2
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-2
Short description	Case enables the actors to create an account for using services provided by the application.

Pre-condition(s)	Application page must be open.
Input data	Enter valid information.
Detailed steps	• User enter invalid username or password • Click the "Register" button.
Expected result(s)	System shows the particular message.
Post-condition(s)	No execution.
Identifier	TC-3
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-3
Short description	Case enables the actors to create an account for using services provided by the application.
Pre-condition(s)	Application page must be open.
Input data	Enter valid information.
Detailed steps	• User misses a compulsory field. • Click the "Register" button.
Expected result(s)	System shows the particular message.
Post-condition(s)	No execution.

5.3.2 Test Case: 2

Login Account (Actor: Admin, HOD, Event Manager& Coordinator)

Identifier	TC-4
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-4
Short description	Enables the admin to access his homepage.
Pre-condition(s)	Admin should have a valid account.
Input data	Username & Password
Detailed steps	• Admin enters valid user name and password. • Successful login.
Expected result(s)	Will show the home page.
Post-condition(s)	Will give access to his home page.

Identifier	TC-5
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-5
Short description	Enables the HOD to access his home page.
Pre-condition(s)	HOD should have a valid account.
Input data	Username & Password

Detailed steps	• User enters invalid user name and password.
Expected result(s)	Gives error and prompt for next try.
Post-condition(s)	Re-try login page appears.

Identifier	TC-6
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-6
Short description	Enables the Coordinator to access his home page.
Pre-condition(s)	Coordinator should have a valid account.
Input data	Username & Password
Detailed steps	• User left any username or password field empty
Expected result(s)	Gives error and request to fill that field.
Post-condition(s)	Login page appears again.

5.3.3 Test Case: 3

Logout (Actor: Administrator, HOD and Coordinator)

Identifier	TC-7
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-7
Short description	Enables the actors to logout.
Pre-condition(s)	Actor must be login.
Input data	Actor click on the "Logout" button.
Detailed steps	• Actor click on the "Logout" button.. • System will logout successfully.
Expected result(s)	System will logout successfully.
Post-condition(s)	After execution system will return to the login page.

Test Case: 4

Add Admin

Identifier	TC-8
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-8
Short description	Enables the Main Administrator to add sub admin.
Pre-condition(s)	Main Administrator must login.
Input data	Enter the admin information.
Detailed steps	• Admin enter valid information. • Click the "ADD" button.
Expected result(s)	Account created in database.
Post-condition(s)	System will shows the confirmation message.

Identifier	TC-9
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-9
Short description	Enables the Main Administrator to add sub admin.
Pre-condition(s)	Main Administrator must login.
Input data	Enter the admin information.

Detailed steps	• Enters the details of the admin which is already present and registered. • Click the "ADD" button.
Expected result(s)	System displays error message that admin is already exists.
Post-condition(s)	System retains page.

Identifier	TC-10
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-10
Short description	Enables the Main Administrator to add sub admin.
Pre-condition(s)	Main Administrator must login.
Input data	Enter the admin information.
Detailed steps	• User left some empty field in the registration form. • Click the "ADD" button.
Expected result(s)	System displays error message and request to fill that space.
Post-condition(s)	System retains page.

5.3.4 Test Case: 5

Add Course (Actor: Administrator)

Identifier	TC-11
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-11
Short description	Enables the administrator to add course.
Pre-condition(s)	Administrator must be login.
Input data	Enter the required information of course.
Detailed steps	• Administrator click on “Add" button... • Course will be selected from the menu and submitted.
Expected result(s)	System will show the confirmation message and save course in database.
Post-condition(s)	After execution system will show confirmation message.

5.3.5 Test Case: 6

Delete Course (Actor: Administrator)

Identifier	TC-12
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-12
Short description	Enables the administrator to delete course.
Pre-condition(s)	Administrator must be login.
Input data	Click on the "delete" button.
Detailed steps	• Administrator click on “delete" button... • Course will be deleted from the database.
Expected result(s)	System will show the confirmation message and delete course.
Post-condition(s)	After execution system will show confirmation message.

5.3.6 Test Case: 7

Update Course (Actor: Administrator)

Identifier	TC-13
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-13
Short description	Enables the Administrator to update course.
Pre-condition(s)	Administrator must login.
Input data	Admin view the course list.
Detailed steps	• Admin update the information. • Click the "UPDATE" button.
Expected result(s)	Updated course will be saved in database.
Post-condition(s)	System will shows the confirmation message.

5.3.7 Test Case: 8

Add Student (Actor: Administrator)

Identifier	TC-14
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-14
Short description	Enables the Administrator to add a student.
Pre-condition(s)	Administrator must login.
Input data	Enter the student information.
Detailed steps	• Admin enter valid information. • Click the "ADD" button.
Expected result(s)	Account created in database.
Post-condition(s)	System will shows the confirmation message.

Identifier	TC-15
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-15
Short description	Enables the Administrator to add a student.
Pre-condition(s)	Administrator must login.
Input data	Enter the student information.

Detailed steps	• Enters the details of the student which is already present and registered. • Click the "ADD" button.
Expected result(s)	System displays error message that Student is already exists.
Post-condition(s)	System retains page.

Identifier	TC-16
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-16
Short description	Enables the Administrator to add a student.
Pre-condition(s)	Administrator must login.
Input data	Enter the student information.
Detailed steps	• User left some empty field in the registration form. • Click the "ADD" button.
Expected result(s)	System displays error message and request to fill that space.
Post-condition(s)	System retains page.

5.3.8 Test Case: 9

Update Student (Actor: Administrator)

Identifier	TC-17
Priority	High
Related requirement(s)	REQ-1
Related use-case(s)	UC-17
Short description	Enables the Administrator to update student.
Pre-condition(s)	Administrator must login.
Input data	Admin view the student list.
Detailed steps	• Admin select the student he/she wants to update. • Admin updates the information. • Click the "UPDATE" button.
Expected result(s)	Updated profile will be saved in database.
Post-condition(s)	System will shows the confirmation message.

Chapter No: 6

Summary and Conclusions

6. Summary & Conclusion

6.1 Summary

In this project documentation we have briefed to the supervisor, advisor and to the FYP team that what we have done in our project. First of all we have created of software named Digital Notification System. Now why we have created this project? We created this project to transform our university into digitalization and it's a big step into this. What this system can do? This system will solve many daily academic problems that are faced by students and faculty members of a university i.e. if a result is displayed on a notice board there comes a rush of students around it and it can be caused due to some reasons that are first of all it is displayed on a page, a page of A4 size, now it's a small regarding that rush situation and will be displaying just a one type of information but with our system a lot of information can be displayed on just a digital board hence it will reduce workload of staff members it will reduce paper consumption and it will be updated from time to time to reduce errors. Now in this document we have shown every aspect of this project from Introduction to Conclusion and by reading this document I'm sure that our teachers and staff members will be happy to see our work detailed very beautifully in a good way which is easy to read and easy to understand.

6.2 Conclusion

By doing the documentation and this project me& my team have learned a lot, on how to do project in a given amount of time, how to work together on different aspects of the project, how to make presentations and merge all our efforts that we have learned so far in these four years of BSCS. We thank a lot to our teachers who helped us in this project in a way that we can't forget our entire life and last but not least we thank our parents who have carried us this far that we are about to hold degrees of which we dreamed about our entire lives. At the end we just want to say that this project will help students and faculty members in a lot of ways. It will reduce a lot stress from the Faculty members a lot of paper work will be saved and it will show the information in a very good and precise manner and we think this project will get good attention from all departments of the university.

Chapter No: 7

Lessons Learnt and Future Enhancements

7. Future Enhancements

7.1 Lessons Learnt

We learned a lot by doing this project.

• Team Work

• Languages used: ASP.Net on Visual Studio, C sharap for backend, JAVA

• Database: MySQL

So during this project we learned all the above things. Before this project, we had no idea about MySQL although we had little bit knowledge of this before. But now we learned a lot about ASP.Net and got knowledge of using Android and Java for developing mobile application and PHP for server side scripting. Now we prefer to work on command line rather than graphically. We learned how to work together on different difficult tasks. If we talk about the project, Digital Notification System has reduced lot of manual work. It has made notifying each and every user very easy and that too with no time paper work hurdles.

7.2 Future Enhancements

This application of Digital Notification System can be further extended to include the following features:

- This System can be converted on mobile with just an application that will do all the work like it is being done on a computer.
- We can create an application for user and students who will receive the notices on their mobile with notice not going to everyone just the ones that notice belongs to.
- We can add audio and visual aids as notices.
- We can include different type of files as notices
- This system can also be used not just in universities but offices too.

Chapter No: 8

Bibliography

8. Bibliography

8.1 References

1. Software Requirements Specification, Kansas State University, March 2003, (http://www.cis.ksu.edu/~cme6556/software_requirements_specification_1.0.php)
2. IEEE Guide for Software Quality Assurance Planning
3. IEEE Standard for Software Quality Assurance Planning, IEEE Std 730.1-1995
4. Software Project Management: A Unified Framework
5. US State Department of Energy (energy.gov)
6. 12 Best Software Development Methodologies with Pros & Cons (acodez.in)
7. Spiral Model in Software Development Life Cycle (SDLC): Phases, Explanations, Software Quality Assurance Plan(www.slideshare.net)
8. Methodology (xbsoftware.com)
9. http://searchcrm.techtarget.com/definition/entity-relationship-diagram
10. https://www.tutorialspoint.com/dbms/dbms_data_schemas.html
11. http://whatis.techtarget.com/definition/use-case-diagram
12. https://www.slideshare.net/NurIslam5/organization-and-team-structures
13. https://support.office.com/en-us/article/Delete-or-change-a-header-or-footer-on-a-single-page-a9b6c963-a3e1-4de1-9142-ca1be1dba7ff
14. https://stackoverflow.com/questions/34857212/how-to-create-notification-system-use-case-diagram
15. https://softwareengineering.stackexchange.com/questions/278135/notifications-in-use-case-diagrams
16. http://www.aspforums.net/Threads/982531/How-to-perform-multiple-field-search-in-Entity-Framework/
17. https://forum.axure.com/t/dynamically-resize-individual-repeater-items-based-on-text-length/31057
18. https://www.freshdesignweb.com/jquery-image-slider-slideshow/
19. https://w3layouts.com/admin-templates/
20. https://colorlib.com/wp/free-html5-admin-dashboard-templates/
21. https://www.udemy.com/the-complete-database-modeling-and-design-beginners-tutorial/

8.2 Appendix

a. Team Structure:

The team structure that we used to create this project is Democratic Team Structure. In this structure everyone can contact with everyone at the same time this leads to higher morale and no one is left behind, the diagram will show you better:

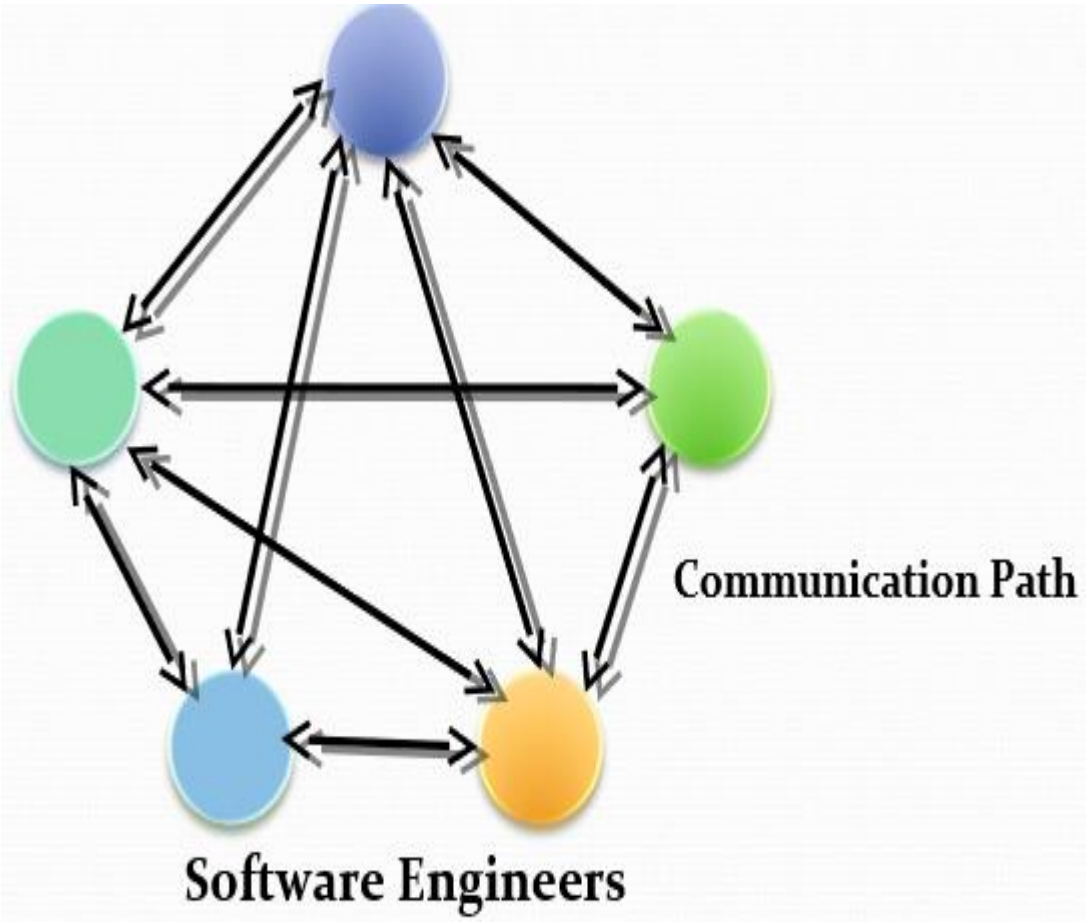

Figure 36: Team Structure Diagram

b. Roles& Responsibilities:

Roles:

Team Leader: Talha Riaz

Data Collector**:** Adil Amin

Data Analyzer: Shoaib Arshad

Requirement Specifier: Umair Shahbaz

Responsibilities

Documentation and Front-End Developer: Talha Riaz

Documentation and Back-End Developer: Adil Amin

Graphic Designer and Front-End Developer: Shoaib Arshad

Back-End Developer and Database Designer: Umair Shahbaz

c. Scope Statement:

The system will have to interact with a university environment. Therefore we are interested in keeping under control the adjacent systems such as the database, the university department software, the mailing system, or the computerized environment. Regarding the human environment, we have to consider the program manager above all and the indulgency for the rules of the faculty. This system should not be considered an error free system, there should be some tolerance regarding this system.

d. Meetings Summary:

Department of CS & IT Superior University

Final Year Project Meetings Performa

Project Title: Digital Notification System

Advisor: Mr. Waqas Asghar

Student Names	Progress	New Tasks	Date	Signature

I, _____, being Project Advisor of this group, certify that above mentioned information is true as per my knowledge and concern. I allow the group to present its Final Year Project to the Panel.

_____ Signature of Supervisor