

SPES

(STUDENT PERFORMANCE EVALUATION SYSTEM)



BY
CAPT EHTESHAM UL HASSAN
CAPT MOHSIN HUSSAIN
CAPT MEHDI HUSSAIN
CAPT IRFAN HAIDER

SUPERVISOR
LT COL HUMAYUN ZUBAIR KHAN, PHD

Submitted to the Faculty of Software Department
National University of Sciences and Technology, Islamabad
in partial fulfillment for the requirements of a B.E Degree in
Computer Software Engineering

May 2022

CERTIFICATE OF CORRECTION & APPROVAL

This is to officially state that the thesis work contained in this report titled “**SPES**” (**Student Performance Evaluation System**) Is carried out by: **EHTESHAM UL HASSAN, MOHSIN HUSSAIN, MEHDI HUSSAIN, IRFAN HAIDER** under my supervision and that in my judgment, it is fully ample, in scope and excellence, for the degree of Bachelor of Computer Software Engineering from National University of Sciences and Technology (NUST).

Approved By:

Signature: _____

Supervisor: Lt Col *Humayun Zubair Khan, phd*
MCS, NUST Rawalpindi

DECLARATION OF ORIGINALITY

We hereby declare that the work contained in this report and the intellectual content of this report are the product of our work. This thesis report has not been formerly published in any structure nor does it include any verbatim of the published resources which could be treated as violation of the international copyright decree. We also affirm that we do recognize the terms 'plagiarism' and 'copyright' and that in case of any copyright infringement or plagiarism established in this thesis, we will be held fully accountable of the consequences of any such violation.

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached at the end of the document.

Signature: _____

Supervisor: Lt Col *Humayun Zubair Khan, phd*

ACKNOWLEDGEMENTS

In the name of Allah, the most merciful and the most Beneficent. Who led us to this extent. May all glory, honor and Adoration be unto Thy Name.

Our special thanks go to our supervisor **Lt Col Humayun Zubair Khan, phd** for guiding us throughout the process that resulted in the successful completion of our project. We would also like to thanks to the faculty of Software Department for their guidance in the building of the project.

At last, we are most obliged to our Parents, their support contributed immensely to the success of this project.

*Dedicated to my exceptional parents and adored siblings whose
tremendous support and cooperation led me to this wonderful
accomplishment*

ABSTRACT

Nowadays most of the education institutes encourage the use of technology in teaching mechanism rather than using the traditional teacher centered teaching mechanism to enhance the learning ability of the students by making a student-centered learning mechanism. The teachers must evaluate the students' performance no matter what their learning mechanism is and there are ways of doing that i.e., formal evaluation and informal evaluation. That means by giving exam papers and giving feedbacks based on the students' behavior.

Teachers want to improve student performance, and technology can help them accomplish this aim. To mitigate the challenges, administrators should help teachers gain the competencies needed to enhance learning for students through technology. Additionally, technology in the classroom should make teachers' jobs easier without adding extra time to their day.

Table of Contents

1	Introduction.....	3
1.1	Intended Audience and Reading Suggestions.....	3
1.2	Motivation.....	3
1.3	Project Vision.....	3
1.4	Project Objective.....	4
1.4.1	Primary Objectives:	4
1.4.2	Academic objectives:	4
1.4.3	Application / End – goal objectives:	4
1.5	Deliverables	4
	Chapter 2.....	5
2	Literature Review.....	6
2.1	Introduction.....	6
2.2	Problem Domain	6
2.3	Related Work	6
2.4	Shortcomings/issues.....	6
2.5	Proposed Project	7
2.6	Deliverables	7
2.6.1	Software Requirement Specification (SRS).....	7
2.6.2	Software Architecture Document	7
2.6.3	Software Design document	7
2.6.4	Implementation code Document	8
2.6.5	Software Testing Document	8
2.6.6	Final Project Report	8
2.6.7	User Manual.....	8
2.7	Technological Requirements	8
2.7.1	Software Interfaces	8
2.7.2	Programming Interface	9
	Chapter 3.....	2
3	Overall Description.....	3
3.1	Product Perspective.....	3
3.2	Product Functions	3
3.2.1	User Profiles.....	3

3.2.2	Login Access.....	3
3.2.3	Creation of Subjects.....	3
3.2.4	Import structures.....	3
3.2.5	Access Right.....	3
3.2.6	Students.....	3
3.3	User Classes and Characteristics.....	4
3.4	Operating Environment.....	4
3.4.1	Technology Platform:.....	4
3.5	User Documentation.....	5
	Chapter 4.....	6
4	Software Requirements Specification.....	7
4.1	System Features.....	7
4.1.1	Use Case UC1: Register.....	10
4.1.2	Use Case UC2: Login.....	10
4.1.3	Use Case UC3: View Course.....	10
4.1.4	Use Case UC4: View Subjects.....	10
4.1.5	Use Case UC5: Add Subject.....	11
4.1.6	Use Case UC6: Import Subjects.....	11
4.1.7	Use Case UC7: View Students.....	11
4.1.8	Use Case UC8: Add Student.....	11
4.1.9	Use Case UC10: Import Students.....	11
4.1.10	Use Case UC10: View Course Result.....	11
4.1.11	Use Case UC11: Import Course Result.....	11
4.1.12	Use Case UC12: View Course Performance.....	11
4.1.13	Use Case UC13: View User Profile.....	11
4.1.14	Use Case UC15: Logout.....	11
4.2	Other Non-functional Requirements.....	11
4.2.1	Performance Requirements.....	12
4.2.2	Safety Requirements.....	12
4.2.3	Security Requirements.....	12
4.2.4	Software Quality Attributes.....	12
4.2.5	Business rules:.....	13
4.3	Software Quality Attributes.....	13
4.3.1	Runtime System Qualities.....	13
	Chapter 5.....	14

5	System Design Specifications.....	15
5.1	System Architectural Design	15
5.2	Class Diagram.....	16
5.3	Use Case Diagram.....	18
5.3.1	Use Case UC1: Register.....	18
5.3.2	Use Case UC2: Login	19
5.3.3	Use Case UC3: View Course.....	21
5.3.4	Use Case UC10: Update user Profile.....	21
5.3.5	Use Case UC12: View User Profile.....	22
5.3.6	Use Case UC15: Logout	23
5.4	Sequence Diagrams.....	23
5.4.1	User Registration	23
5.5	Activity Diagrams	25
5.6	Design Rationale.....	25
5.7	Component Design.....	27
	Chapter 6.....	30
6	System Implementation	31
6.1	Technology Used	31
6.1.1	Programming Language Used.....	31
6.1.2	Development Tools.....	31
6.1.3	Database.....	31
6.1.4	Operating System.....	31
6.2	Complete System Implementation.....	31
6.2.1	Login Module.....	31
6.2.2	Register Dialog	33
6.2.3	Chief Instructor Registration Form.....	34
6.2.4	Instructor Registration	35
6.2.5	Student Registration.....	36
	Chapter 7.....	37
7	System Implementation	43
7.1	Overview.....	43
7.1	Unit Testing	43
7.1.1	Login Feature Testing	43
7.1.2	Registration	44
7.1.3	Import Subjects	45

Chapter 8.....	46
8 Conclusion and Future Work.....	47
8.1 Conclusion.....	47
8.2 Future work.....	47
Glossary.....	48
Bibliography.....	49

Chapter 1

1 Introduction

This document completely describes the system at the architecture level, including subsystems and their services, hardware mapping, data management, access control, global software control structure, and boundary conditions. The document is organized into nine major sections. Each section provides detailed sub-sections relevant to the major section. Charts, tables, and graphics have been inserted to explain or clarify content.

1.1 Intended Audience and Reading Suggestions

This document is meant for all the stake holders.

- **Project Supervisor:** It will assist to supervise the project and guide the group available by better manner.
- **Development Team:** It will assist the developer to develop the application and to discover back the functional requirements.
- **Testing Team:** It yearns for assistance to testers to comprehend the constraints.
- **Complainers (Users):** The likely participants of the product, who are interested in academic analytics like teachers, professors etc.
- **UG Project Evaluation Team:** Assessment board which will evaluate the assessment and evolution of UG Projects.
- **Staff:** The intended audience is both higher management and local officials of the different departments, who will update the complaint-status after its resolution.

1.2 Motivation

With the modern technology the traditional academic evaluation system can be optimized in different levels. The software systems can be developed to gather analytics and facts about the rise and falls in the academic trends. These analyses can be used to improve the lacking areas. As a software engineer such systems are intriguing and can be used for long term. Such systems can also help in future predictions and can be used on existing valuable data to analyze past trends. Areas where system can suggest the improvement can be lagging students, subjects that have lower average grades, proficiency comparisons of a student with past academic records.

1.3 Project Vision

We propose a system that will provide core functionalities i.e., visual representation of academic analysis. This system can provide following core functionalities to cater analytics needs:

- Student Performance.
- Course Performance.
- Representation of facts and figures by charts and graphs.

The project consists of a web application that can be plugged in to an existing system, and its data can be utilized for above mentioned features.

1.4 Project Objective

1.4.1 Primary Objectives:

1. Web platform that can be run as standalone system.
2. The application can be connected to different management systems.
3. Maximize the information gathering on data trends.
4. A secure login system to different roles.

1.4.2 Academic objectives:

1. Development of a Web application hosted along the existing system.
2. To have a good hand on using the plug and play system using Microsoft technologies i.e., C# and LINQ to SQL.
3. To automate the traditional evaluation system.
4. Introducing a shared medium for instructors and professors, academic institutes and universities visualizing academic trends.

1.4.3 Application / End – goal objectives:

1. To facilitate the end users of the application in terms of secured access and modular analysis features.
2. To enable university to improve and optimize academic processes.
3. To save the time and resources of manual work.
4. To enable the institutes to keep a track of the current academic trends.

1.5 Deliverables

1. Complete working project
2. Web application
3. Documentation
4. Video of working of Project

Chapter 2

2 Literature Review

2.1 Introduction

This document completely describes the system at the architecture level, including subsystems and their services, hardware mapping, data management, access control, global software control structure, and boundary conditions. The document is organized into nine major sections. Each section provides detailed sub-sections relevant to the major section. Charts, tables, and graphics have been inserted to explain or clarify content.

2.2 Problem Domain

The web application is a performance evaluation system designed to help instructors to evaluate results more efficiently, help draw comparisons, show graphical representation of student's performance and current standing in the course. It is designed to make the job of instructors easy by filling up the data forms in real-time and showing the graphical representation of the result. It is designed to show the performance of the student in a graphical representation and highlighting rise and fall in the course performance and to calculate the average current standing in the course. Grade comparison with the previous 4 x courses will also be available. Access based student performance updates/ graphs assist course advisor in preparation of performance briefs.

2.3 Related Work

Currently there is no system working that provides the facts and figures about the academic data. The instructors compile the lacking areas by hand. This system can not necessarily eliminate the problem but greatly ease the relevant work process.

2.4 Shortcomings/issues

1. An existing comprehensive data base is required to give deep analysis.
2. The application is web-based application only.
3. Can always be enhanced with different types of queries.
4. For earlier versions, the contents of the application will be in English language only.

2.5 Proposed Project

Student Performance Evaluation System (SPES) provides a comprehensive interface for maintenance and grading student performance over the MIMS (RMS). It can be used by the faculty to maintain the records and performance analysis of students easily.

The creation and management of accurate, up-to-date information regarding a students' academics without having a fear of data loss and managing all the records and evaluation over the MIMS without having to use any other platform. It tracks all the details of a student from the day one to the end of the course which can be used follow: For all reporting purpose, reports communicate information which has been compiled as a result of research and analysis of data and of issues.

Reports can cover a wide range of topics, but usually focus on transmitting information with a clear purpose, tracking of weekly test scores, entered by the teacher for a particular class. Class ranking of student, is a measure of how a student's performance compares to other students in his or her class. Student Performance Evaluation System will enhance the existing system by its platform which enables the faculty to grade students by automatically generating results.

2.6 Deliverables

2.6.1 Software Requirement Specification (SRS)

The purpose of the document is towards the introduction of a comprehensive picture of the SPES. It describes the persistence and characteristics of the system, the interfaces, the boundaries of the system, whatever the product will do, its processes and workings, noted that the constraints by which it ought to operate and how the system is going to act in response to exterior stimuli. This paper is intended for developers and the participants of the system. It shall describe how the system will primarily aid concerned groups to team up and cooperate with each other.

2.6.2 Software Architecture Document

In this document the overall architecture of the system is discoursed, also including the introduction of various components and subsystems. It is chiefly braced by system Architecture diagram which depicts an insider's viewpoint of the system by unfolding the high-level software components that execute the major functions to make the system operational.

2.6.3 Software Design document

The Software Design Document is a record to give documentation that shall be utilized to help in programming advancement by giving the subtleties to how the product ought to be fabricated. Inside the Software Design Document are narratives and graphical documentation of the product plan for the task. It covers every single practical

prerequisite and shows how they communicate with one another adroitly. The low-level design additionally appears with respect to how really, we have been executing how we are going to actualize these requirements.

2.6.4 Implementation code Document

The implementation code document provides details about the pseudo code for the application and project prototype.

2.6.5 Software Testing Document

This document has testing modules in which there are certain test cases which depicts the correctness and accuracy of the project.

2.6.6 Final Project Report

This is the thesis report which compiles all the previous and current working for the project. Thesis report provides the whole summary for the project and gives details about every aspect of the project starting from introduction of the project, literature review, requirements leading to design discussions then testing and lastly future work and conclusion.

2.6.7 User Manual

User Manual gives details about the use of the product. It contains details as how to use the product. Its functionalities and details of every aspect as how that works and how to use it. User Manual is for users to get to know the product.

2.7 Technological Requirements

SPES entails subsequent software and hardware requirements specifications.

2.7.1 Software Interfaces

1. The system requires access to an existing data base, and login access for the users.
2. SPES shall work on SQL Server database management system.
3. SPEC app shall be able to execute on all windows systems with basic hardware requirements.
4. The app will require IIS to run.

2.7.2 Programming Interface

Programming interfaces for project are:

1. SQL Server Management Studio
2. Visual Studio

Chapter 3

3 Overall Description

3.1 Product Perspective

The current evaluation system is based on the cumbersome process in which users must track data figures and information like scores, averages and peaks and descents manually.

In this project we develop a comprehensive solution that shall not only maximize the speed of the process but also shall provide supportive measures like charts and graphs.

3.2 Product Functions

Following are the key functions of the Student Performance Evaluation System (SPES):

3.2.1 User Profiles

SPES registers every user and maintain their records.

3.2.2 Login Access

This application provides access to the existing created users i.e., Students, Instructors and Chief Instructors logins and roles.

- User launches app and logs in as a specified role.
- Students can view their performances.
- Instructors and Chief Instructors can view the course level performances.

3.2.3 Creation of Subjects

The system allows the creation of course and subjects hierarchy along with the subject total marks, teaching instructor and type of subject i.e., practical theory.

3.2.4 Import structures

The system will cater the resolve for the issue of large course structures by the feature if importing data through Excel files. The Excel file will be structured according to the needs and data structure i.e., Subject Type, Subject Category, Subcategory and the course it belongs to.

3.2.5 Access Right

SPES will allow users to login, based on their roles. User of the application can either sign-in as Students, Instructors or Chief Instructors.

3.2.6 Students

The system also provides the feature of course enrollment. Where a student user can be created and enrolled into a referred course.

3.3 User Classes and Characteristics

The software has four types of users i.e., Chief Instructor, Instructor, Student and Admin. These two categories of users shall be provided with different access level to the system and its data and can perform functions assigned to their respective roles.

Chief Instructor: - Chief Instructor role has the access to overall system. This role can view all the courses currently being offered and view course level and student level performances.

Instructor: - Instructor level can access the performances of the courses that are assigned to it. Instructor can enroll the students and create course results.

Student: - Student role can view their performance and course level performance for courses in which they are enrolled.

Admin: - This role has access on overall system, it can manage create, edit and delete the artifacts of system.

3.4 Operating Environment

- This web application can be deployed on MIMS and will be able to process datasheets or results stored in SQL servers.
- Minimum RAM 2 GB
- Reasonable Storage Space.
- Intel i3 Processor.
- PC or Laptop with MIMS access.
- This application can be accessed by user through a machine having any web browser compatible with MIMS.

3.4.1 Technology Platform:

3.4.1.1 Web based Front End:

SPESs front-end would be developed for web browser, providing the users with the interface to get registered on the server, and view performance details. .Net development tools would be used as the development environment.

3.4.1.2 Programming languages:

- C#

3.4.1.3 Programming Environment

- .Net framework
- Visual Studio

3.4.1.4 Database

- SQL Server Database

3.5 User Documentation

UD- version 1.0: Ultimate publication shall be complemented with a user manual to enlighten users how to use SPES. User documentation that would be delivered along with the final product

- User manual

Chapter 4

4 Software Requirements Specification

4.1 System Features

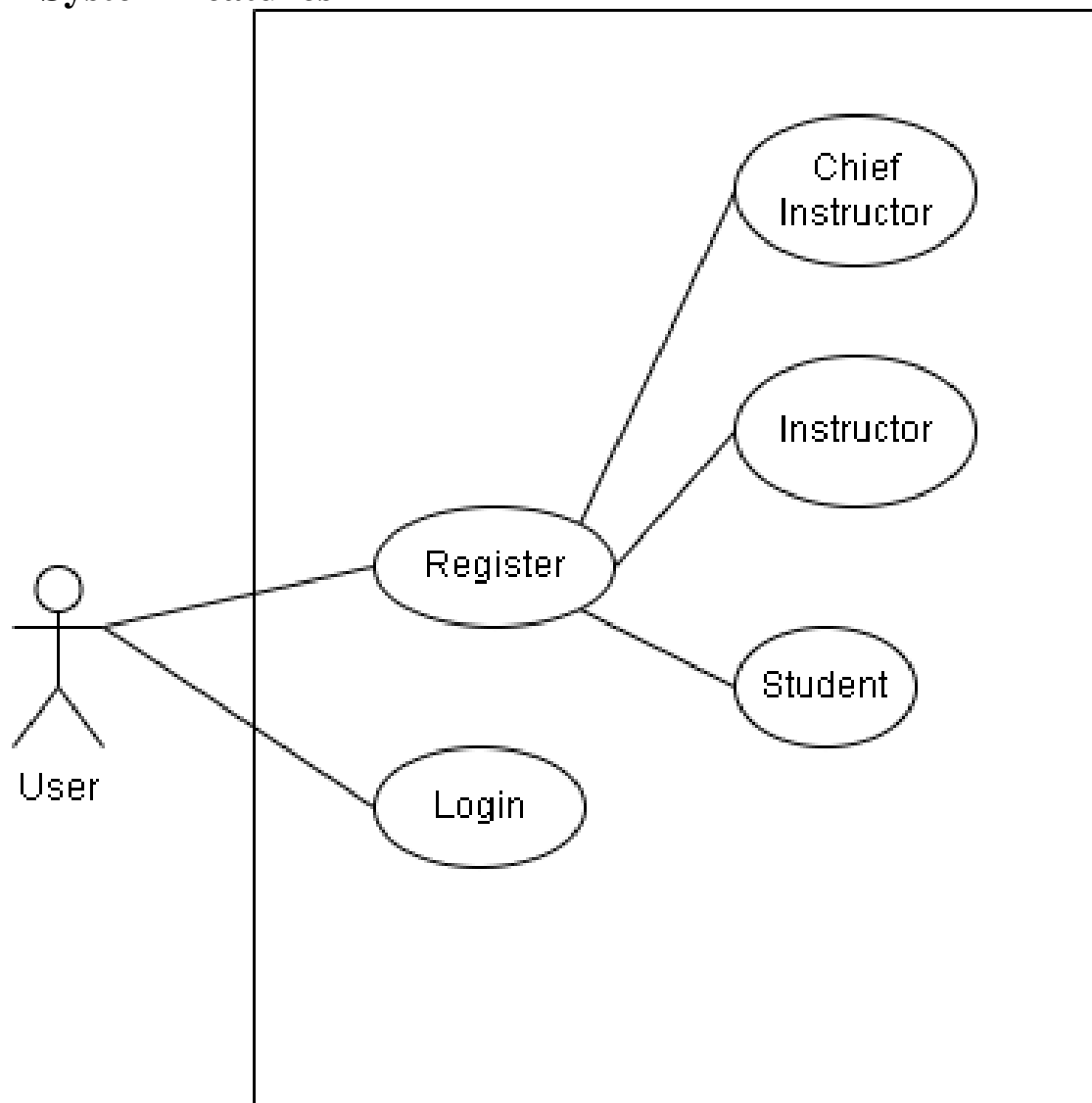


Figure 01-Use Case Diagram

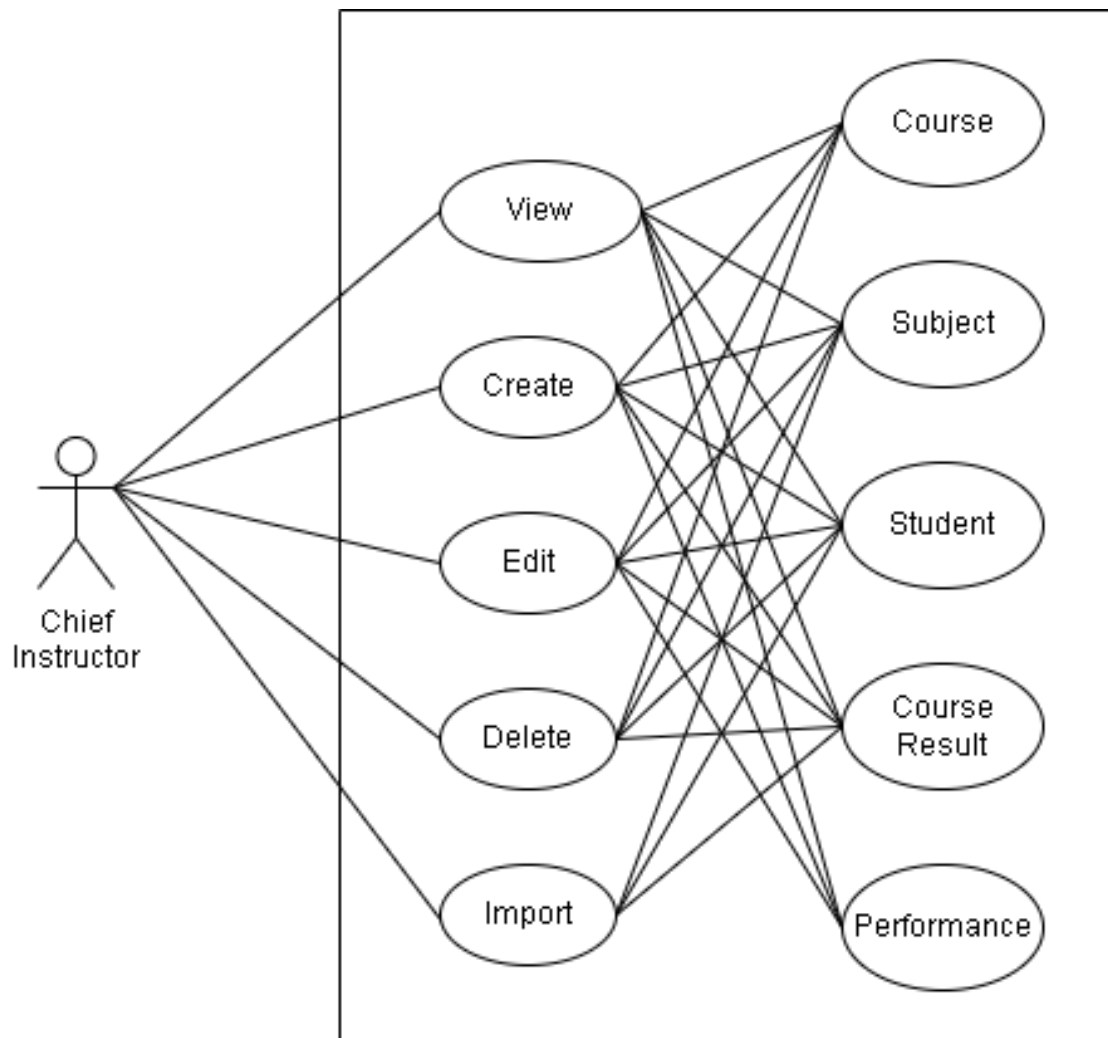


Figure 02 Use Case Chief Instructor

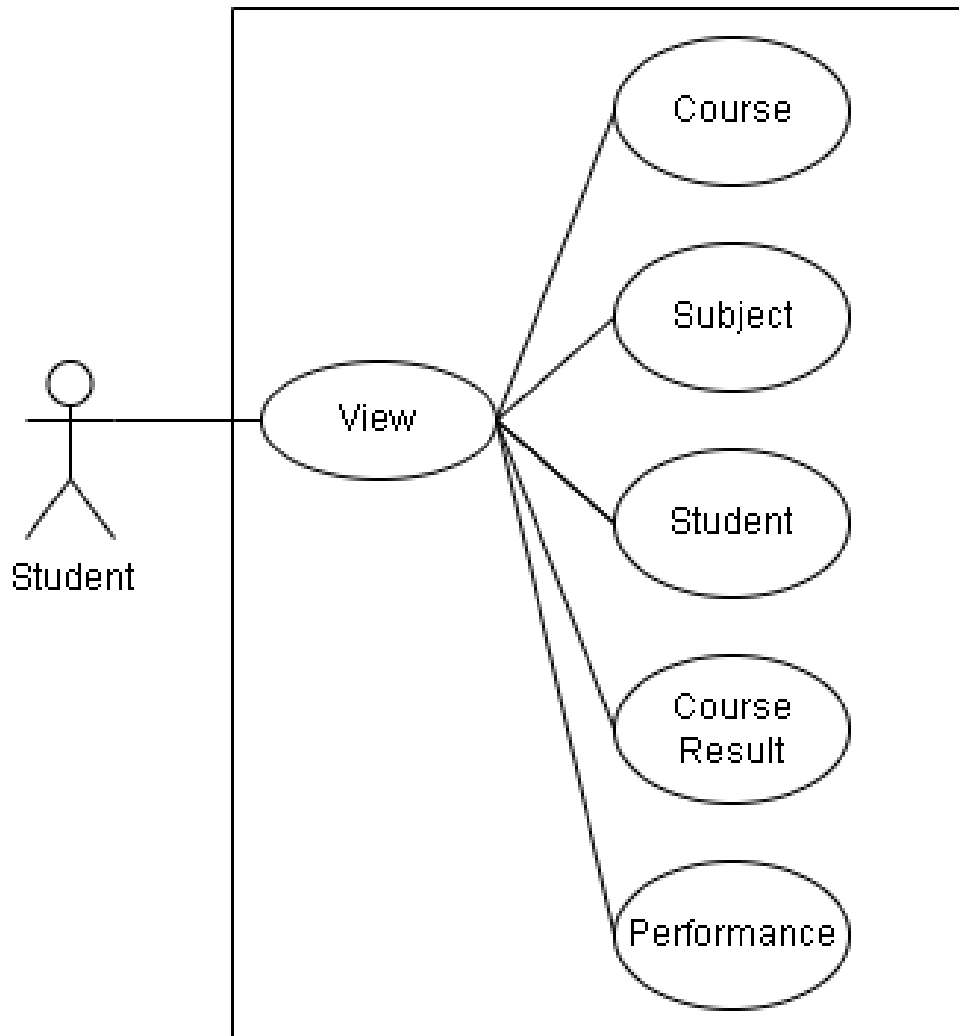


Figure 03 Use Case Student

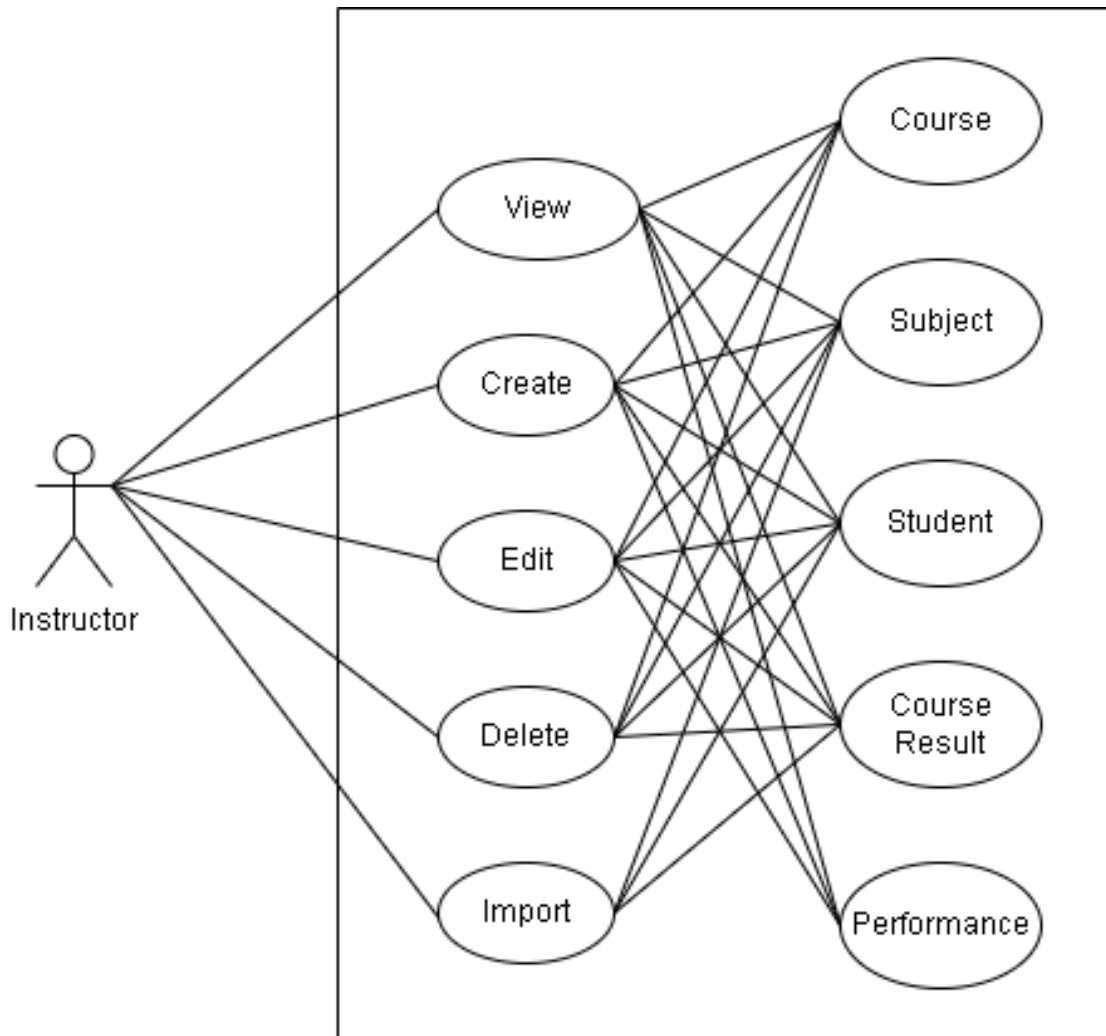


Figure 04 Use Case Diagram Instructor

4.1.1 Use Case UC1: Register

User Can sign up to the application. Every user is registered in the central database and credentials are maintained for the registered users

4.1.2 Use Case UC2: Login

Users of SPES can Login to the system only if they are already registered that way its provides the authentication of credentials of legitimate users.

4.1.3 Use Case UC3: View Course

User can view courses which are related to its role. Admin and Chief Instructors can view all the current offered courses. While Instructors and Students can only view the courses which are assigned to them or enrolled in.

4.1.4 Use Case UC4: View Subjects

User can view the details of subjects in the course.

4.1.5 Use Case UC5: Add Subject

The subjects can be created in a particular course from UI by the authorized user.

4.1.6 Use Case UC6: Import Subjects

User of application can import a bulk of subjects by the help of excel sheet.

4.1.7 Use Case UC7: View Students

User of the application can view the list of students enrolled in the course.

4.1.8 Use Case UC8: Add Student.

Students can be created by Admin, Chief Instructor and Instructor roles. The students would be created and enrolled in a particular course.

4.1.9 Use Case UC10: Import Students

Bulk of student records can be imported in the system through excel sheets.

4.1.10 Use Case UC10: View Course Result

Users of the application can view the course result of the students.

4.1.11 Use Case UC11: Import Course Result

Admins, Chief Instructors and Instructors can import the course result for the course.

4.1.12 Use Case UC12: View Course Performance

Users can view the course performances of a single of previous four courses by the help of graphs and charts.

4.1.13 Use Case UC13: View User Profile

User can view into their profile that provides them the interface to check-in their credentials.

4.1.14 Use Case UC15: Logout

System intends to permit user to logout of the application successfully.

4.2 Other Non-functional Requirements

4.2.1 Performance Requirements

- Application ought to be light weight yet represent the comprehensive information.
- The front-page load time must be no more than 2 seconds for users that access the application.
- High Specs android phones shall be able to run the application
- A blended visual representation of data loads quickly and give more information than the textual facts.
- Application can handle many users at a time.

4.2.2 Safety Requirements

- Application shall handle any user's information safely.
- Users must have to register using original information so that if any mishap occurs service shall provide him as much support as possible.
- User credentials and private info shall not be shared with the rest of the users.

4.2.3 Security Requirements

- Only authorized users can modify their profile and CRUD operations of artifacts.
- The system shall not be accessed by any unauthorized person.

4.2.4 Software Quality Attributes

- **Availability:** System shall be operating on any time during working hours of the service and can manage more than one ride at a time.
- **Reusability:** The components of the system shall be written in a way that they are easy to reuse.
- **Reliability:** The system defect rate shall be as less as possible.

4.2.5 Business rules:

- Application provides the course hierarchy, its results and performance board.
- User must register to the application to use the services.
- Optimizes and helps institutes and universities to catch the lacking areas.

4.3 Software Quality Attributes

Quality attributes of SPED are portrayed underneath. In the wake of these characteristics, the quality of SPES shall be enhanced.

4.3.1 Runtime System Qualities

At execution SPES ought to offer its users with features that they can publish and search for the desired services. Some of the qualities that needs to be counted in the development of SPES are portrayed here.

4.3.1.1 Functionality

SPES must provide functions to publish and search the different services. SPES ought to offer the feature of authentication of user.

4.3.1.2 Availability

SPES should be available 24/7 since the complaint can be lodged at any time. If at all system is down so the servers will take about 15 minutes to start the SPES again.

4.3.1.3 Usability

Usability is an important criterion in the development of SPES. The system should present all functionalities in such a way that nothing is missed by the user. The graphical user interface of app is to be designed with usability as the priority. The app will be presented and organized in a manner that is both visually appealing and easy for the user to navigate.

4.3.1.4 Non-Runtime System Qualities

These are characteristics of SPES which are required to make this software useful for further enhancements. It will also be helpful in future development as well as extending system to different environments.

4.3.1.5 Modifiability

SPES ought to support modifiability in case any further enhancements or features are effortless to incorporate.

4.3.1.6 Portability

The system should work on WIFI as well as 3G network.

4.3.1.7 Testability

Various quality assessments ought to be executed so that SPES is exempt of flaws and operate agreeing to requirements.

Chapter 5

5 System Design Specifications

5.1 System Architectural Design

Layard architecture will be used with each layer providing a set of functionalities. These layers will be composed of multiple services which will communicate with each other via message passing. Abstractly, the services can be observed as components of the complete solution. Though, on the inside, each facility is made up of software components, exactly as any other application, additionally these components can make use of each other without being aware of the internal implementation

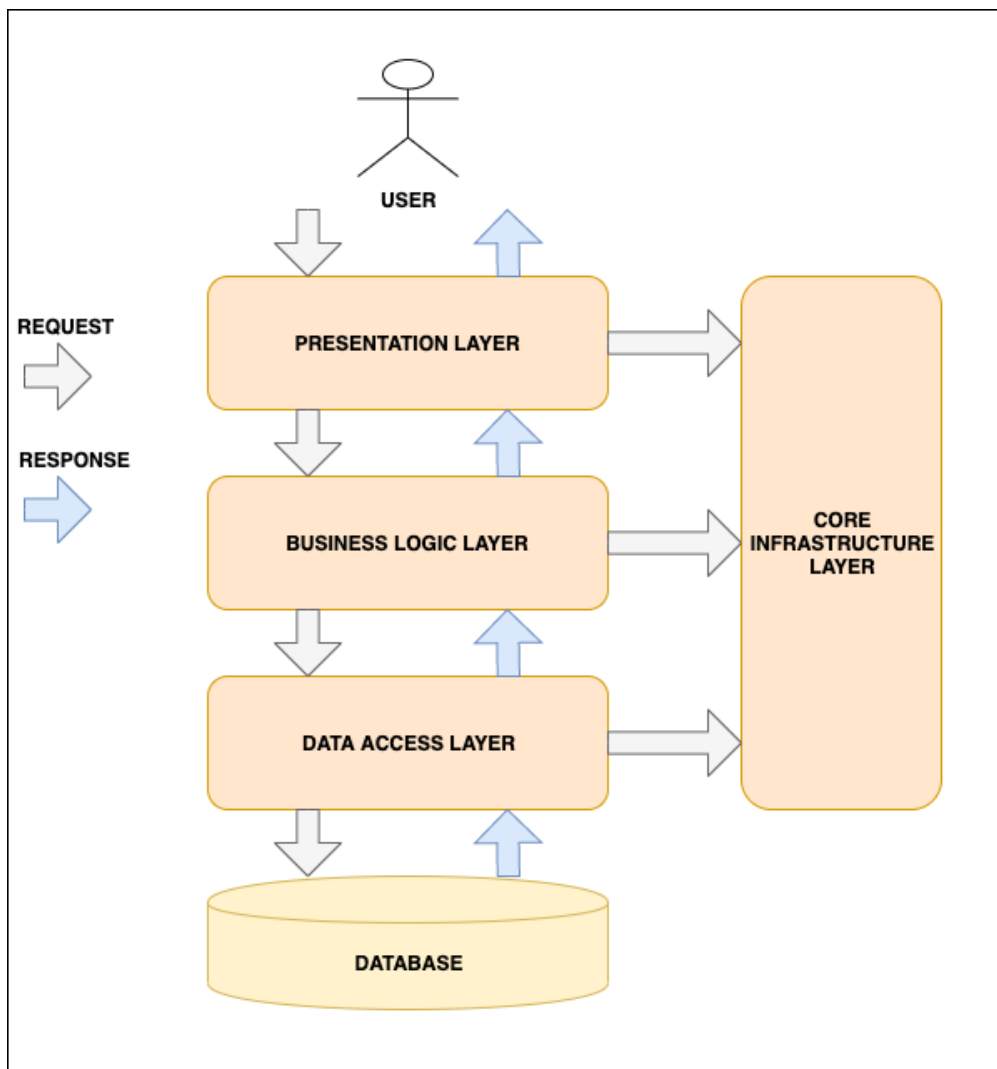


Figure 2 System Architecture

5.1.1 Presentation Layer

This layer act as a manifesto for the communication of the user and system. This layer presents data to the user and recognizes input from the user.

5.1.2 Business Logic

It provides the core functionality to application which a user is required to have from the application.

5.1.3 Data Access Layer

This layer entertains the request from the Service Layer and sends report back data after querying from the database server.

5.2 Class Diagram

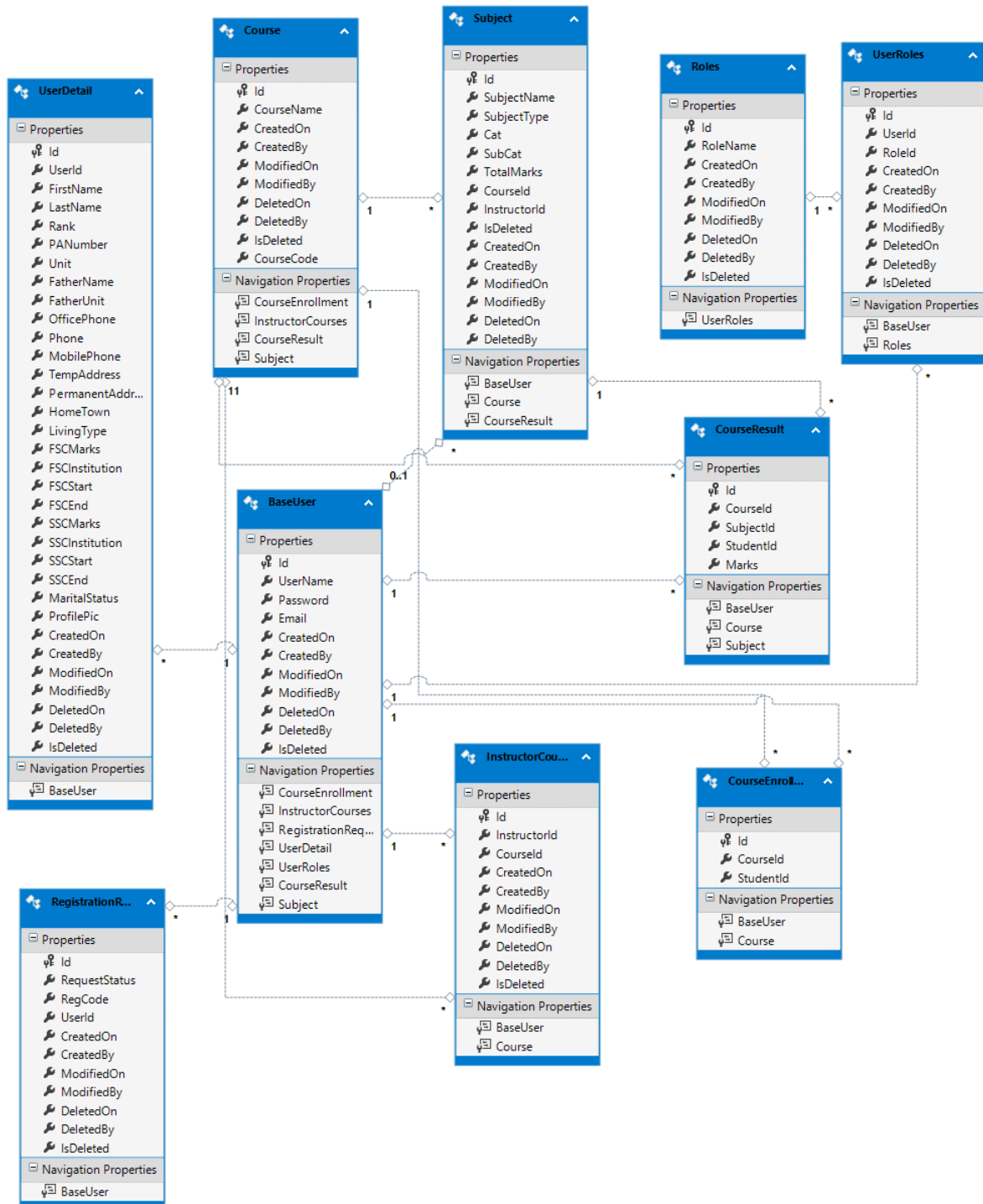


Figure 3 Class Diagram

The description for class diagram is shown on the next page:

Class name	Description
	App class contains all the information that SPES has to perform. It is the main class which will be acting as a gateway to all the other classes
Base User	This class represents the current or existing user object of the application. Its also a parent class and it is being inherited by UserDetails class which has all the details relevant to a user in the system.
Register and Login	This class contains the functions for sign in and sign-up processes for the users.
System interface	It contains all the information to enable user to interact with SPES. It has links to all the functions of different classes that on selection lead to different actions.
Dashboard	It contains the functions with intents to the Profile and Logging out of the application
Registration and Authentication	Registration and verifications are included as the functions in the signup and registration class Registration is to save data in database and verification is to verify the stored data at the time of the login.
User Profile	User profile saves the information including Name Number, Picture, type and phone Number of the User Profile saves Name and phone Number of the user .
Role	This class contains the information about the roles of the users in the system.

Table 1 System Features

5.3 Use Case Diagram

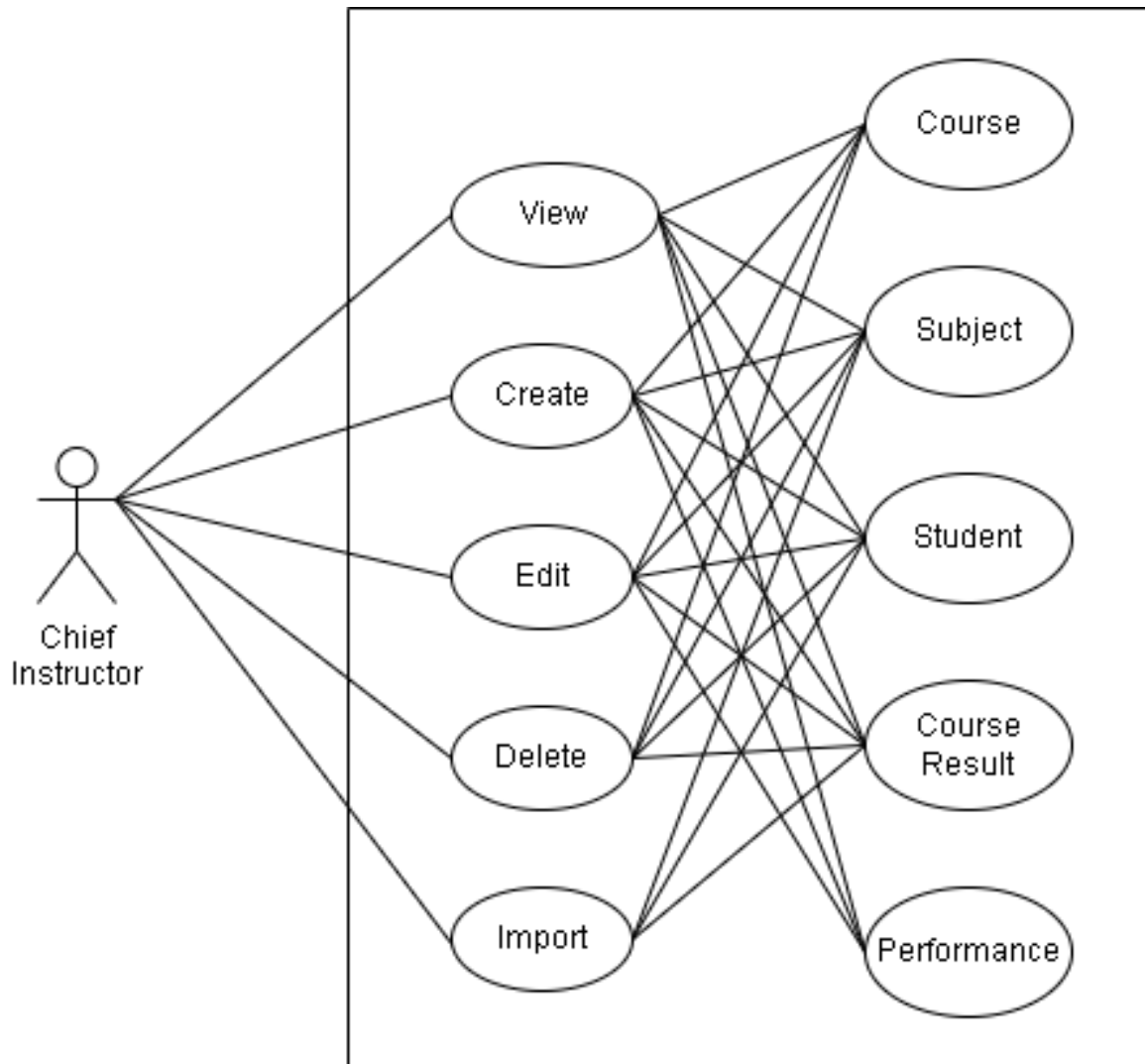


Figure 4 Use Case Diagram

5.3.1 Use Case UC1: Register

Name: Register

Scope: APES

Primary Actor: Unregister User

Description: First, user install the app, after opening the android application the Register and Log In form shall be shown to the user. After filling the required information, the system records the fields of data into database and user gets registered.

Stakeholders and Interests: Unregister user: User must sign up to register their account.

Preconditions: Must have android application to register.

Success Guarantee (Post conditions): Unregister user shall get registered after they provide the valid information.

Main Success Scenario: Following Table 3.1 shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: User requests to create an account click “Create Here “button.</i>	<i>Step 2: The system shall show Sign Up form.</i>
<i>Step 3: User enters the detail e.g. first name, last name, Email, Mobile number and CNIC number click “Register Now button.</i>	<i>Step 4: The system shall redirect to main dashboard.</i>
<i>Step 5: The user shall confirm the information.</i>	<i>Step 6: The system creates the user account.</i>

Table 2 Sign-Up

Extension

In extension we explain the alternative scenarios of use case.

Alternative Flow 1: Username/email already exists

- System displays a message that the email already exists.
- System asks the user to enter another username/email.

Alternative Flow 2: Invalid email ID

- System displays a message that the email is invalid.
- System asks the user to enter a valid email id.

Alternative Flow 3: Mismatched passwords

- System displays a message that the password does not match.
- System asks the user to enter the password again.

Alternative Flow 4: Invalid username/password/email format

- System displays a message that the format is invalid.

5.3.2 Use Case UC2: Login

Name: Login

Scope: SPES

Primary Actor: Registered user

Description: The user opens the application and is directed to the login and signup form then user enters its credentials and system shall successfully login the user.

Stakeholders and Interests: Registered user: User must login to their account.

Preconditions: User must have their account.

Success Guarantee (Post conditions): User can access that application and user can book ambulance now.

Main Success Scenario: Following Table shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: User requests to login to their account.</i>	<i>Step 2: The system asks for details and provides login form.</i>
<i>Step 3: User enters the detail e.g. username and password and press the login button.</i>	<i>Step 4: The system shall confirm the username and password from existing database.</i>
	<i>Step 5: If the username and password matches, then system login to the user account otherwise show message of invalid username/password and try again.</i>

Table 3 Login

Extensions:

In extension we explain the alternative scenarios of use case.

Alternative Flow 1: Username/email doesn't exist

- System shows the message that username you entered does not exist.
- System asks the user to enter the valid username.

Alternative Flow 2: Wrong Password

- System shows the message the password you entered is wrong/incorrect.
- System asks the user to enter the correct password.

Alternative Flow 4: User already login

- User exits in login list so use case end.

Alternative Flow 5: Course doesn't exist

- If the course code specified doesn't exist, the system will prompt with course doesn't exist.

5.3.3 Use Case UC3: View Course

Name: View Course

Scope: SPES

Primary Actor: User

Description: User can view the relevant course, either assigned or enrolled in.

Stakeholders and Interests: User: User can view the list of courses.

Preconditions: User must login to the application.

Success Guarantee (Post conditions): User can access the details of the courses.

Main Success Scenario: Following Table 3.3 shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: The user request the courses list by clicking the courses menu.</i>	<i>Step 2: System shall show the list of courses.</i>

Table 4 Select View Course

Extensions:

In extension we explain the alternative scenario of use case

Alternative Flow 1: Users are not available.

- If courses are not available, system show message courses not available.

Alternative Flow 2: Data loading error occur

- If data loading error occur, then contact the system admin.
- Show message to user 'Please contact the system admin'.

5.3.4 Use Case UC10: Update user Profile

Name: Update user profile

Scope: SPES

Primary Actor: User following

Description: User shall successfully login in to system can update profile.

Stakeholders and Interests: Registered User: User can update the profile.

Preconditions: User must login successfully .

Success Guarantee (Post conditions): User can access that application and update the profile.

Main Success Scenario: Following Table shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: The user clicks on profile.</i>	<i>Step 2: The system shall show user profile form.</i>
<i>Step 3: The user shall press edit button.</i>	<i>Step 4: System shall edit the record in edit text and show to the user.</i>
<i>Step 5: User shall add the record and press update the button.</i>	<i>Step 6: System shall update the record according to user ID.</i>

Table 11 update user profile

Extensions:

In extension we explain the alternative scenario of use case

Alternative Flow 1: Do not complete profile.

- If user doesn't complete the profile the user shall not be able to use the app.

Alternative Flow 4: Data loading error occur

- If error occur data loading, then check your internet connection.

5.3.5 Use Case UC12: View User Profile

Name: View profile

Scope: SPES

Primary Actor: Registered User

Description: User shall successfully logins to system then the user shall View the profile

Stakeholders and Interests: Registered User: User shall View the profile.

Preconditions: User must login have to complete the profile.

Success Guarantee (Post conditions):User can access that application and update the profile.

Main Success Scenario: Following Table 3.13 shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: The user clicks on view profile</i>	<i>Step 2: The system shall show profile.</i>

Table 13 View User Profile

Extensions:

In extension we explain the alternative scenario of use case

Alternative Flow 1: Data loading error occur

- If error occur data loading, then check your internet connection.

5.3.6 Use Case UC15: Logout

Name: Logout

Scope: SPES

Primary Actor: Registered user

Description: The user must have logged in to the application.

Stakeholders and Interests: Registered user: user must have logged in to the application.

Preconditions: User must have a registered account

Success Guarantee (Post conditions): User shall be logged out of the application

Main Success Scenario: Following Table 3.19 shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: The user requests to logout from their account.</i>	<i>Step 2: The system will remove all listener from firebase.</i>
<i>Step 4: User can check login activity.</i>	<i>Step 3: The system will redirect to login activity.</i>

Table 16 Logout

Extensions:

In extension we explain the alternative scenario of use case

Alternative Flow 1: Data loading error occur

- If error occurs during data loading, then check your internet connection.

5.4 Sequence Diagrams

5.4.1 User Registration

The below diagram defines the sequence of actions that happens when a end User tries to sign Up into the system.

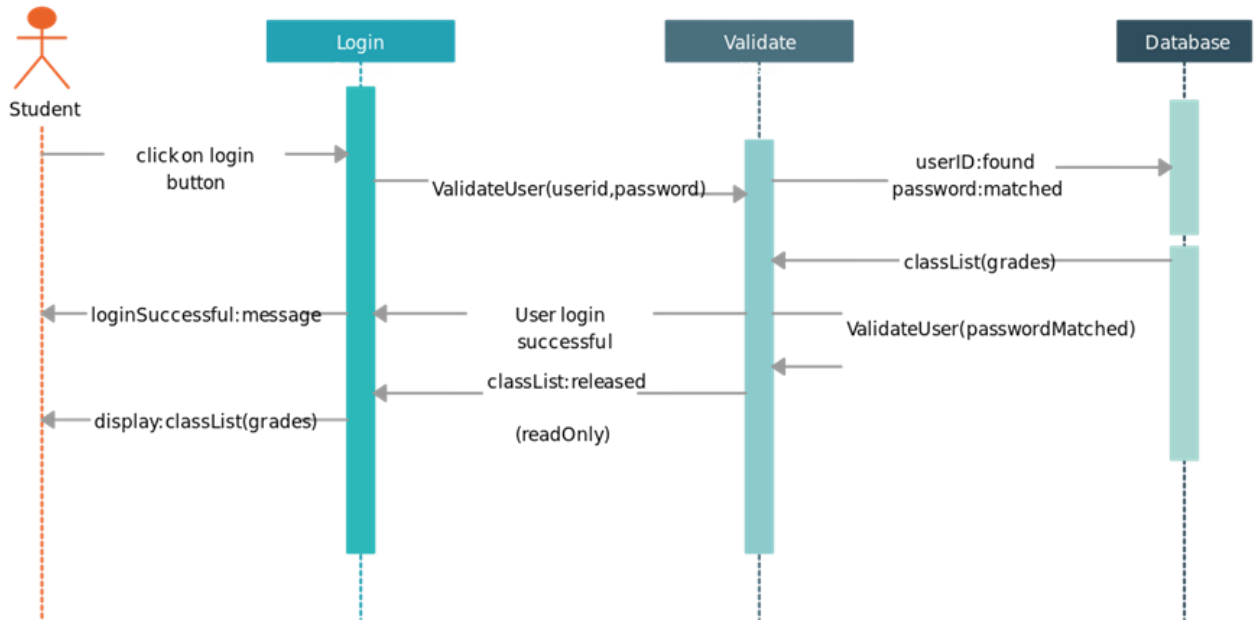


Figure 5 Sequence Diagram Sign-Up

5.5 Activity Diagram

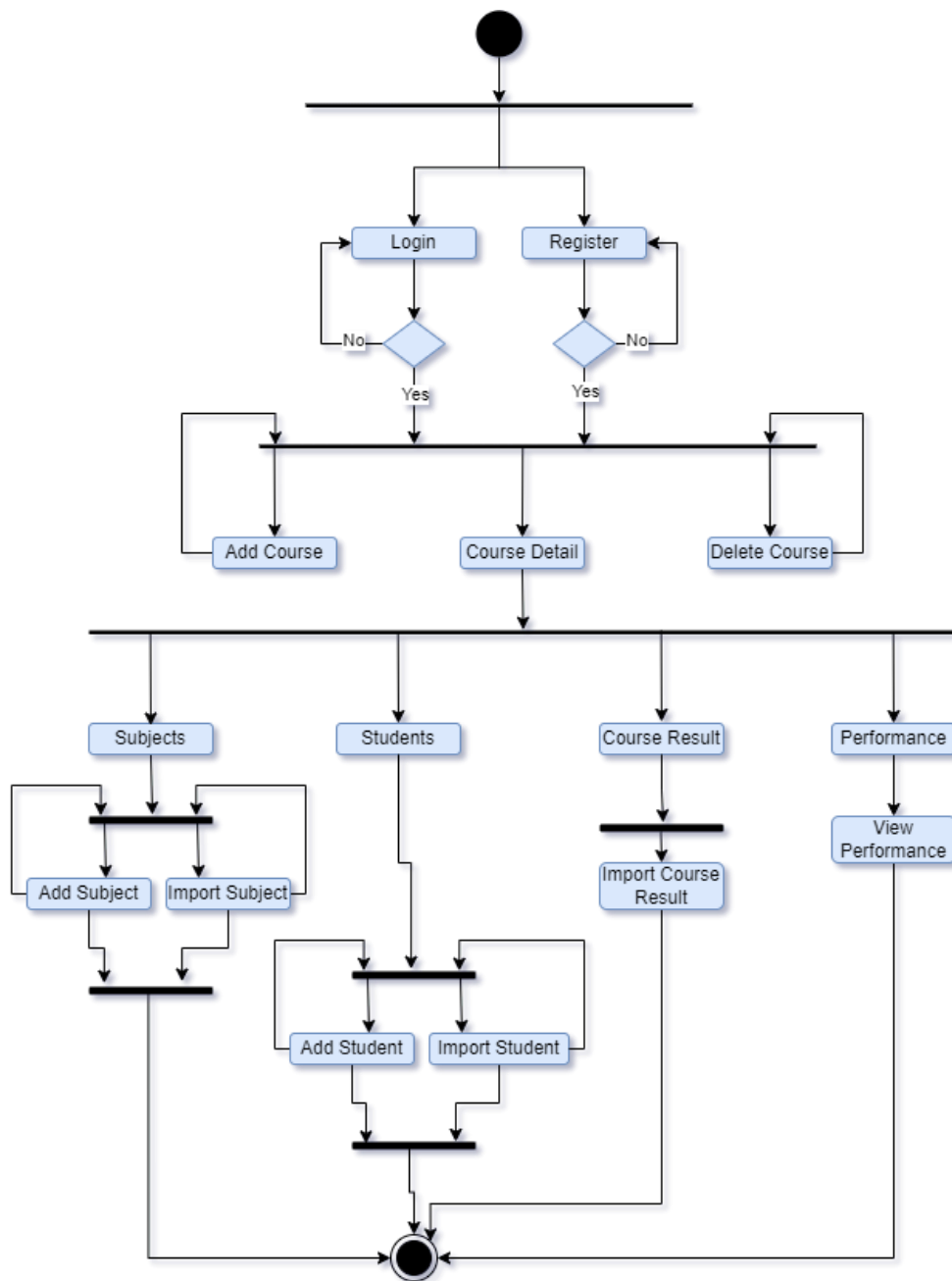


Figure 12 Activity Diagram

5.6 Design Rationale

The layered architecture design is a strong broadly useful example, making it a decent beginning stage for most applications, especially when you don't know what

architecture design is most appropriate for your application. In any case, there are a few interesting points from a architecture design viewpoint while picking the pattern.

The main thing to keep an eye out for is what is known as the architecture sinkhole anti-pattern. This anti-pattern depicts the circumstance where solicitations course through different layers of the design as straightforward go through preparing with practically zero rationale performed inside each layer. For instance, accept the introduction layer reacts to a solicitation from the client to recover client information. The introduction layer passes the solicitation to the business layer, which essentially passes the solicitation to the constancy layer, which at that point makes a basic SQL call to the database layer to recover the client information. The information is then passed right back up the stack with no extra handling or rationale to total, compute, or change the information.

Each layered architecture will have probably a few situations that fall into the design sinkhole hostile to design. The key, be that as it may, is to examine the level of solicitations that fall into this classification. The 80-20 standard is normally a decent practice to follow to decide if you are encountering the design sinkhole against design. It is commonplace to have around 20 percent of the solicitations as straightforward go through preparing and 80 percent of the solicitations having some business rationale related with the solicitation. In any case, in the event that you find that this proportion is switched and a dominant part of your solicitations are basic go through handling, you should consider making a portion of the design layers open, remembering that it will be progressively hard to control change because of the absence of layer confinement.

Another thought with the layered design is that it will in general loan itself toward solid applications, regardless of whether you split the introduction layer and business layers into isolated deployable units. While this may not be a worry for certain

applications, it represents some expected issues as far as sending, general strength and dependability, execution, and versatility.

5.7 Component Design

We gave an Object-Oriented description in section 5.7 and summarized each object member function in all the objects created.

5.7.1 CreatAccount ()

Begin

IF User is an End User

Enter the User Credentials

Else

Enter Driver Credentials and Vehicle Credentials

End

5.7.2 UserCredentials ()

Begin

Enter First name

Enter Last name

Enter Email

Enter Password

Enter Phone Number

End

5.7.3 Login Activity

5.7.3.1 Login ()

Begin

Enter Credentials

Authenticate credentials

IF Authenticate=yes

Access Dashboard

Else

Access Denied

End

5.7.4.1 EnterCredentilals ()

Begin

Enter Email

Enter Phone Number

Enter Password

End

5.7.5 Profile

5.7.5.1 getInfo ()

Begin

Get User Credentials from Sign Up Process

Store User Credentials

End

5.7.5.2 UpdateInfo ()

Begin

Get Changed Information

Update Previous Information

End

5.7.5.3 ShowFeedBack ()

Begin

Take User Feed Back

Take Driver Feed Back

Store Feed Back

Show Feed Back to the User and Driver

End

5.7.5.4 ViewUserProfile ()

Begin

IF UserID=End User's Id

Show User's Stored Information

Else

Show Driver's Stored Information

End

5.7.8 Registration

5.7.8.1 RegisterUser ()

Begin

Get credentials from UserCredentials ()

Store them in Database

End

5.7.9 Authentication

5.7.9.1 AuthenticateCredenmtials ()

Begin

Get Credentials from EnterCredentials ()

Compare them with DataBase

IF CredentialsMatch=Yes

Access DashBoard

Else

Access Denied

End

5.7.10 Verification

5.7.10.1 CheckAlreadyRegisterdUser ()

Begin

Get User Credentials from UserCredentials ()

Get Driver Credentials from DriverCredentials ()

IF Match=yes

Enter Credentials again

Else

Store Credentials

End

Chapter 6

6 System Implementation

6.1 Technology Used

6.1.1 Programming Language Used

SQL Server database has been used for handling all the data storing, retrieval and fetching that is used to operate the application. The asp.net application for the project was written using C#.

6.1.2 Development Tools

Application is developed using Visual Studio development Toolkit. Database access is provided by SSMS of Microsoft.

6.1.3 Database

The systems Database shall be designed and maintained using SQL Server.

6.1.4 Operating System

Web application developed shall be able to run all devices with Windows operating system.

6.2 Complete System Implementation

The system is composed of web-based application with the backend data store. The data store keeps the records for all the logged in sessions. A logged in user can be a student or a teacher. The web pages displayed are customized for these roles. The system understands the role-based access of the resources. The teacher role being the higher role will have more access as compared to the student role.

Furthermore, the system is capable of authorization of its users. A secure and detailed login system takes care of the valid user. The system is capable of profiling and formulation of the existing data. The comprehensive data formulation is optimized by the latest technologies like .NET framework and lightweight yet mature frontend frameworks like jQuery.

6.2.1 Login Module

This is the main module, which greets the user following successful login. This is used to access all the different functionality of the application and can be considered as a main menu.

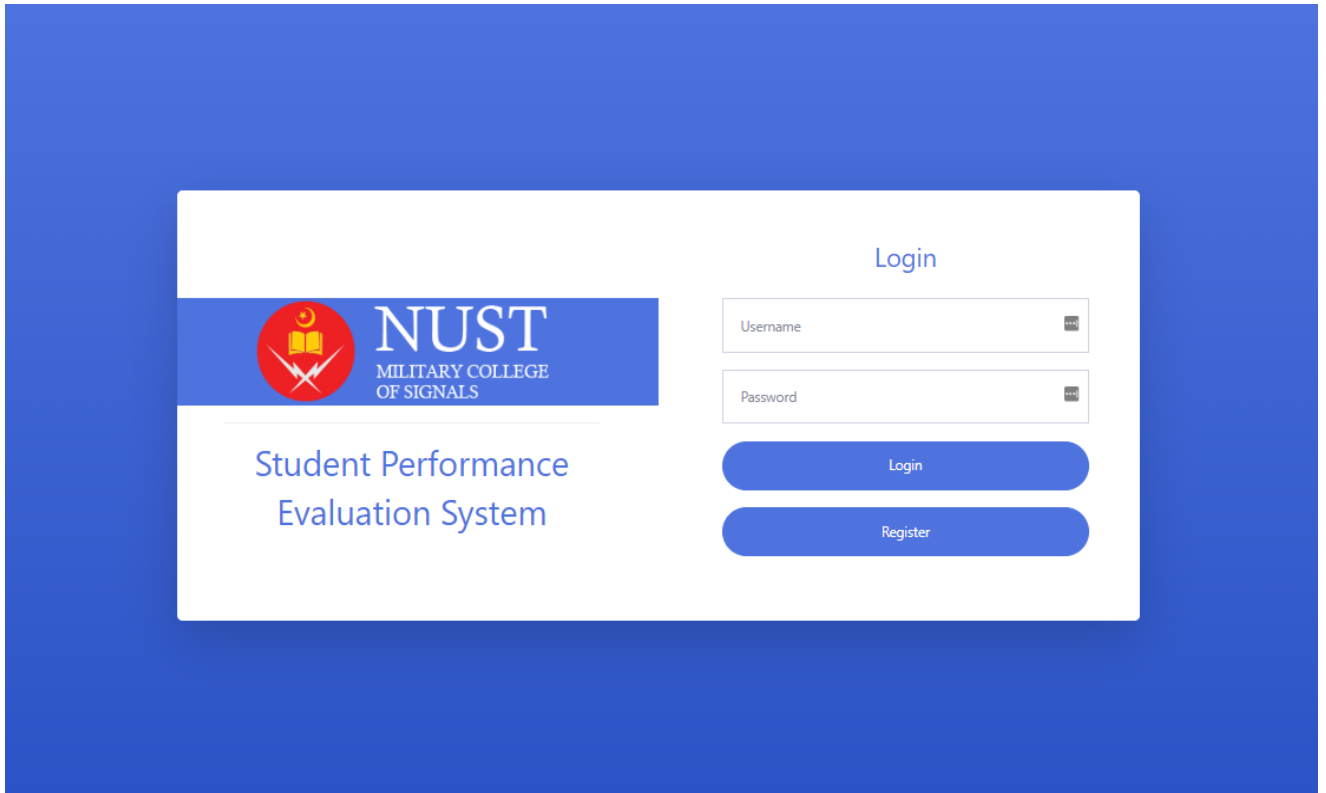


Figure 20 Login Module

6.2.2 Register Dialog

This dialog lets the user to pick the role through it want to register as.

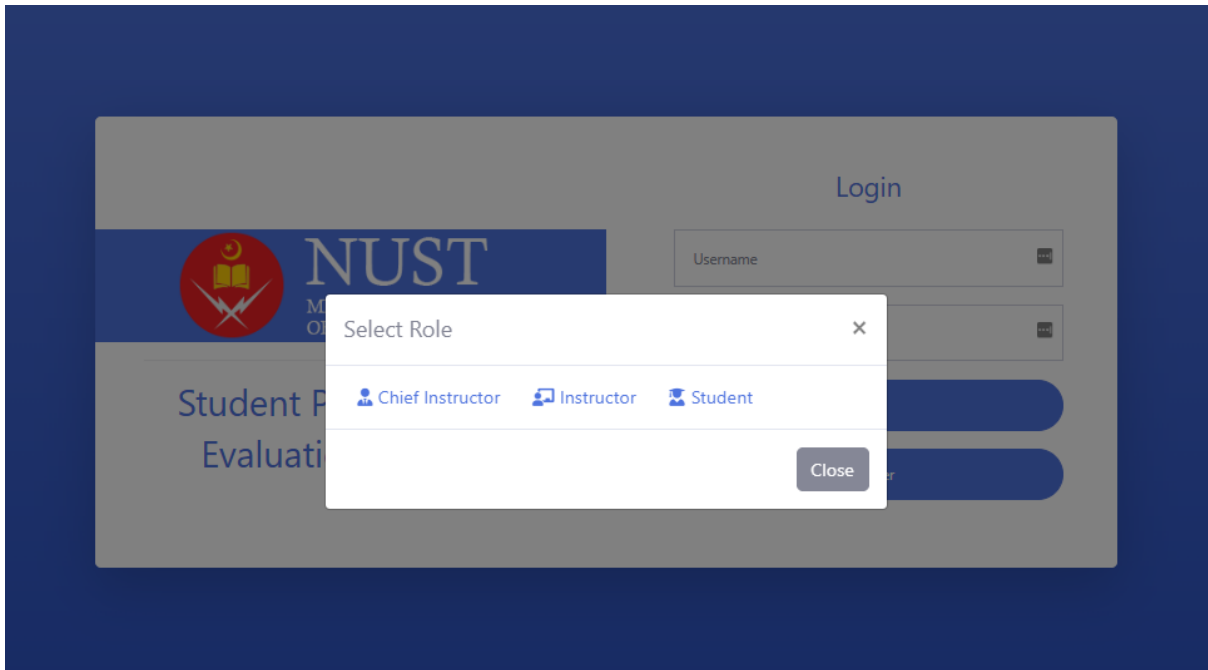


Figure 21 Select Register Role 1

6.2.3 Chief Instructor Registration Form

This form registers a Chief Instructor user role in the system. The user can upload its profile picture along with the required fields on the registration form.

The registration form is displayed on a blue background. At the top, there is a white header bar with a back arrow on the left and the word "Register" in the center. Below the header is a large circular placeholder for a profile picture. The form is organized into several sections, each with a light gray header:

- Bio:** Contains a dropdown menu for "Rank" (set to "None"), a text input for "First Name", and a text input for "Last Name".
- Army:** Contains a text input for "PA Number" and a text input for "Unit".
- Personal:** Contains a text input for "Email".
- Account:** Contains a text input for "User Name" and a text input for "Password".

At the bottom of the form is a prominent blue rounded button labeled "Submit".

Figure 22 Chief Instructor Reg

6.2.4 Instructor Registration

This is the registration form for Instructor user. It contains the course code field which assigns the specified course to this user.

The registration form is titled "Register" and is set against a blue background. At the top, there is a white header bar with a back arrow on the left and the word "Register" in the center. Below the header is a large blue area containing a white silhouette of a person's head and shoulders. The form is organized into several sections, each with a light blue header:

- Bio:** Contains a dropdown menu for "Rank" (currently set to "None"), a text input field for "First Name", and a text input field for "Last Name".
- Army:** Contains a text input field for "PA Number" and a text input field for "Unit".
- Personal:** Contains a text input field for "Email".
- Account:** Contains a text input field for "User Name" and a text input field for "Password".
- Course Code:** A text input field for "Course Code" and a blue "Submit" button.

Figure 23 Instructor Registration

6.2.5 Student Registration

This is the student registration form. It contains fields that are necessary for the student user. It contains the marital status, address and contact information along with the basic bio.

The screenshot shows a mobile registration form titled "Register". At the top, there is a back arrow and the title "Register". Below the title is a large blue header area containing a white silhouette of a person's head and shoulders inside a dark grey circle. The form is divided into several sections, each with a light blue header:

- Bio:** Contains a "Rank" dropdown menu with "None" selected, a "First Name" text input field with a small icon, and a "Last Name" text input field.
- Army:** Contains four text input fields: "PA Number", "Unit", "Father Name", and "Father Unit".
- Contact:** Contains two text input fields: "Phone" and "Address".
- Personal:** Contains an "Email" text input field and "Marital Status" with two radio button options: "Single" and "Married".
- Account:** Contains a "User Name" text input field and a "Password" text input field with a small icon.

At the bottom of the form, there is a "Course Code" text input field and a blue rounded rectangular "Submit" button.

Figure 24 Student Registration

6.2.6 Course Listing

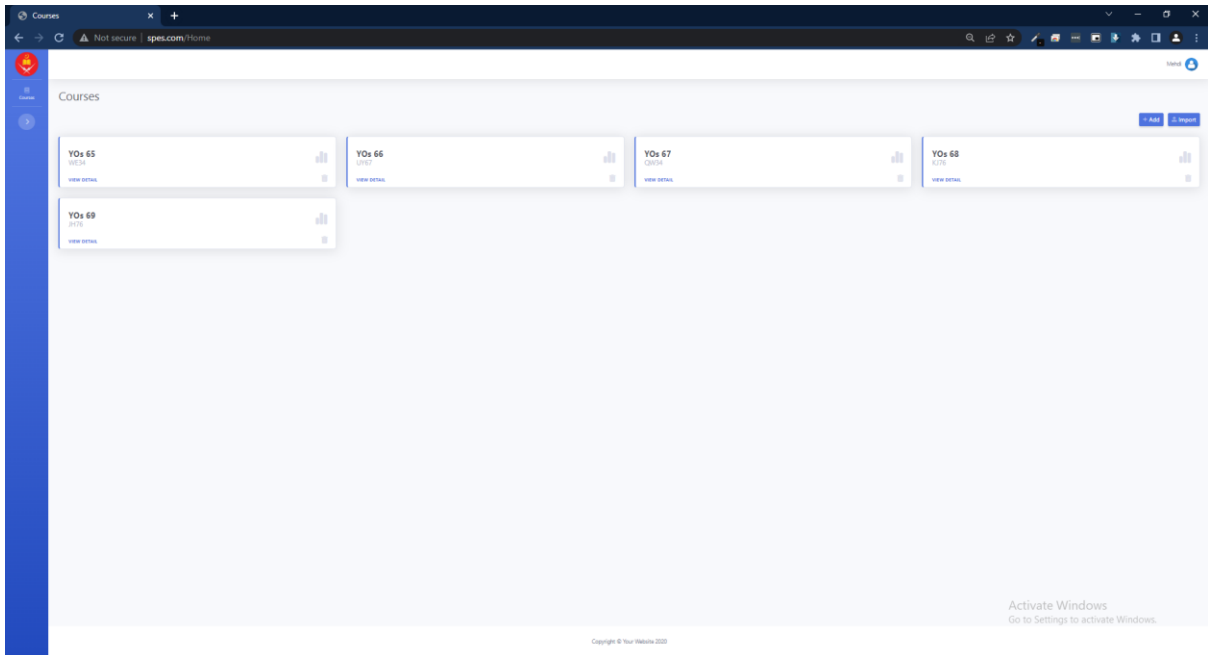


Figure 25 Courses List

6.2.7 Subjects

SubjectName	Sub Cat	Cat	Type	TotalMarks	Instructor	
Discussion Hr	CL Performance	Comm Skills	IA	15	inst1	Details Delete
Oral Presentation	CL Performance	Comm Skills	IA	15	inst1	Details Delete
AP 1	IE	Comm Skills	IA	5	inst1	Details Delete
SI AP	IE	Comm Skills	IA	10	inst1	Details Delete
CI AP	IE	Comm Skills	IA	15	inst1	Details Delete
Class Assessment by C/Adv Term 1	CL Performance	Faculty Assessment	IA	10	inst1	Details Delete
Class Assessment by C/Adv Term 2	CL Performance	Faculty Assessment	IA	10	inst1	Details Delete
IRP Presentation		IRPs	IA	100	inst1	Details Delete
Radio Comm Sys	Subject Test (PRAC)	Appl	Knowledge	40	inst1	Details Delete
LOS Comm Sys	Subject Test (PRAC)	Appl	Knowledge	50	inst1	Details Delete
GIS	Subject Test (PRAC)	Appl	Knowledge	10	inst1	Details Delete

Figure 26 Subjects List

6.2.8 Students

Sadiq Ali PA 57784 VIEW DETAIL		Muhammad PA 57772 VIEW DETAIL	
Mehroz PA 57792 VIEW DETAIL		Danya PA 57826 VIEW DETAIL	
Hamza Nasim PA 57801 VIEW DETAIL		Muhammad Ahtisham PA 57820 VIEW DETAIL	
Usama Hameed Tarar PA 57871 VIEW DETAIL		Muhammad Ibrahim Janjua PA 57887 VIEW DETAIL	

Figure 27 Students List

6.2.9 Student Profile

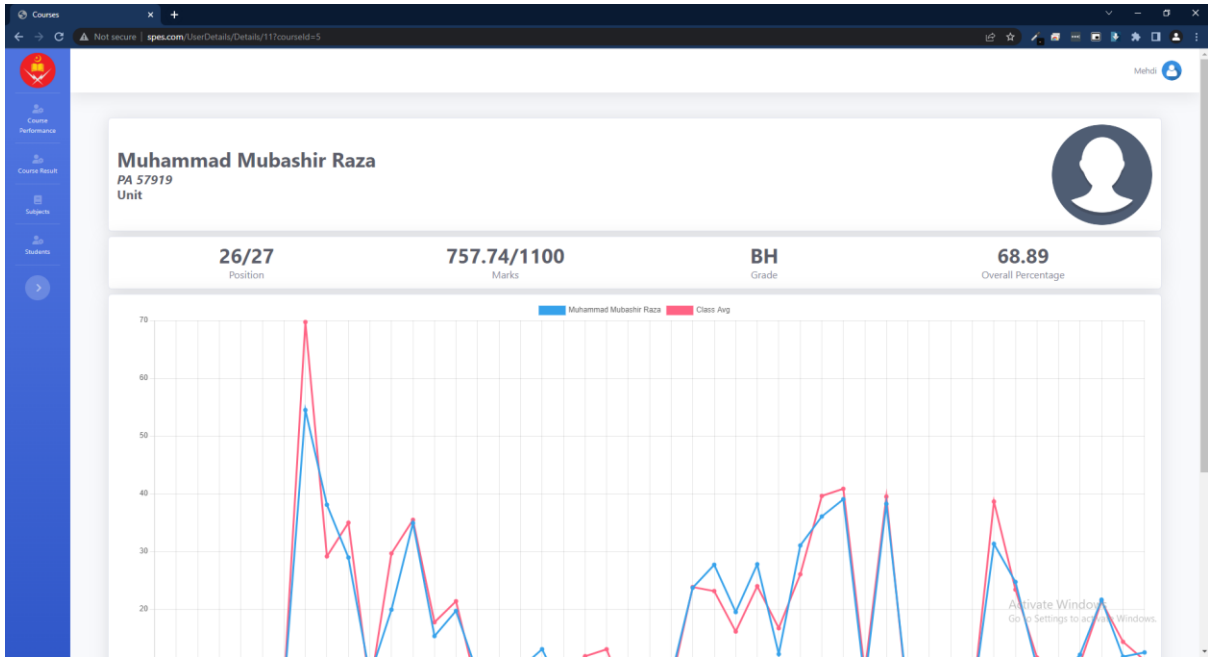


Figure 28 Student Profile

6.2.9.1 Student Result vs Class Average

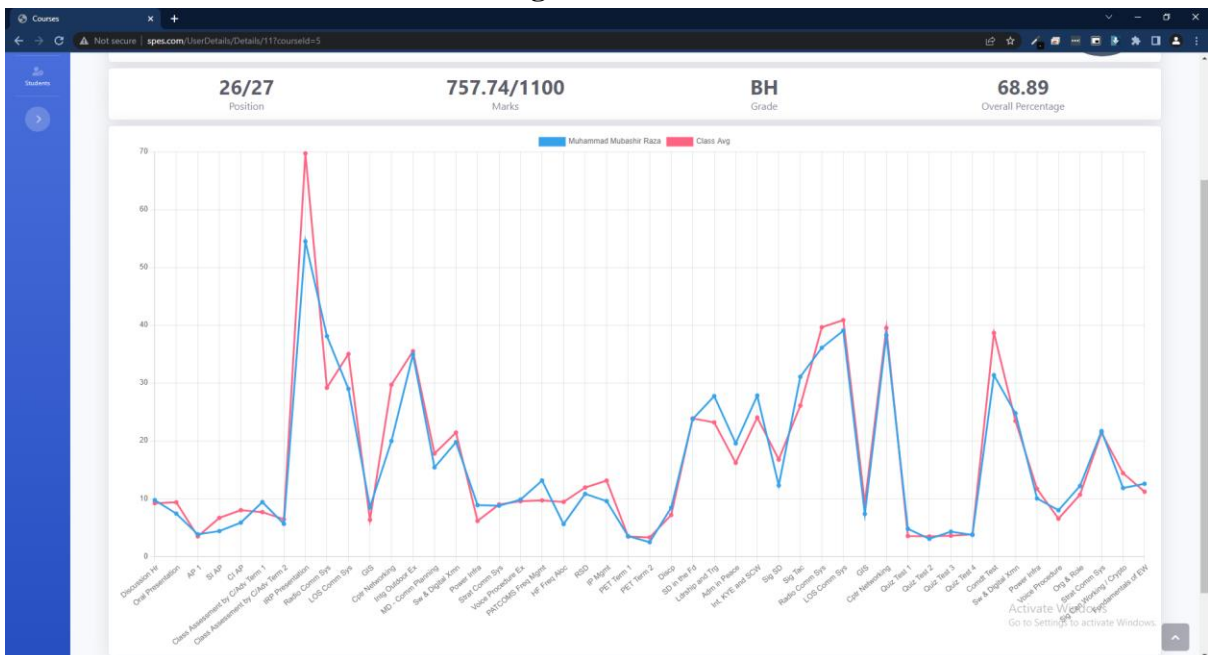


Figure 29 Student vs Class Average Graph

6.2.10 Course Result

CourseName	UserName	SubjectName	Marks	
YOs 69	student1	Discussion Hr	10.33	Edit Details Delete
YOs 69	student1	Oral Presentation	8.65	Edit Details Delete
YOs 69	student1	AP 1	3.86	Edit Details Delete
YOs 69	student1	SI AP	6.13	Edit Details Delete
YOs 69	student1	CI AP	5.32	Edit Details Delete
YOs 69	student1	Class Assessment by C/Adv Term 1	8.96	Edit Details Delete
YOs 69	student1	Class Assessment by C/Adv Term 2	6.64	Edit Details Delete
YOs 69	student1	IRP Presentation	53.76	Edit Details Delete
YOs 69	student1	Radio Comm Sys	30.4	Edit Details Delete
YOs 69	student1	LOS Comm Sys	42.97	Edit Details Delete
YOs 69	student1	GIS	7.53	Edit Details Delete
YOs 69	student1	Cptr Networking	39.18	Edit Details Delete
YOs 69	student1	Intg Outdoor Ex	37.61	Edit Details Delete
YOs 69	student1	MD - Comm Planning	12.22	Edit Details Delete

Figure 30 Course Result List

6.2.11 Performance

6.2.11.1 Class Standings

Name	Position	Grade	Overall Percentage
Ahmad Ali	1	B+	77.69%
Hamza Nasim	2	B+	77.27%
Muhammad Ibrahim Janjua	3	B+	74.04%
Mehroz	4	B+	73.52%
Shehroz Butt	5	B+	73.34%
Muhammad Osama Shahid	6	B+	73.3%
Safi Ullah Shah	7	B+	72.98%
Niranja Bhandari	8	B+	72.96%
Rafi Ullah	9	B+	72.76%
Muhammad Mehran Ul Haq	10	B+	72.74%
Muhammad	11	B+	72.54%
Sadiq Ali	12	B+	72.35%
Shahiq Jamil	13	B+	72.26%
Muhammad Abdullah Saqib	14	B+	72.2%
Akbar Hussain	15	B+	71.78%
Usama Hameed Tarar	16	B+	71.46%

Figure 31 Student Position List

6.2.11.2 Previous Courses Comparison

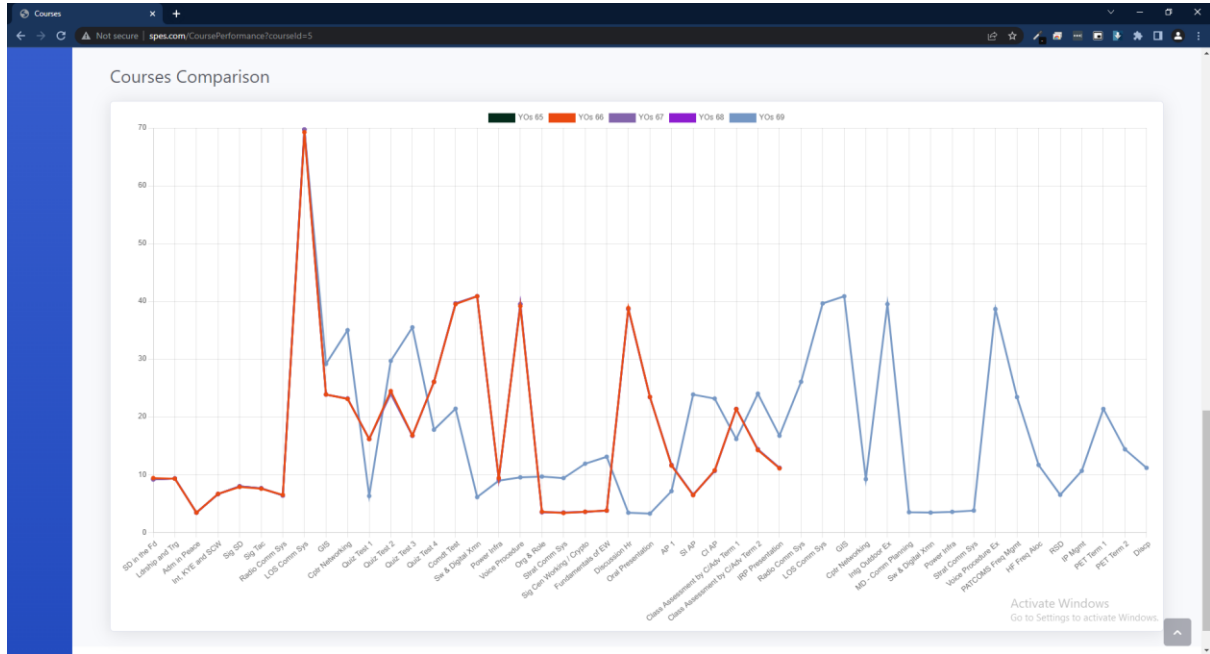


Figure 32 Courses Comparison Graph

Chapter 7

7 System Implementation

7.1 Overview

Testing of software projects include different levels of testing to ensure that the software which is being developed is error and fault free. The different levels at which testing was performed is argued here.

7.1 Unit Testing

It includes the testing of each module at completion.

7.1.1 Login Feature Testing

Case Name:	System Login Testing	
Case ID:		
Description of Case	This module shall enable user to login to the system by providing his credentials. As per the sequence of authorization the credentials are checked with the database, noted that access shall be granted only to verified credentials.	
Testing technique used:	Black Box Testing	
Conditions:	System operating normally and connectivity of system and database is already established.	
Test values:	Username: String Password/PIN: String	
Test Inputs:	Registered Users/legal Users credentials Registered password/legal password	
Steps:	<ol style="list-style-type: none"> 1. Enter username/email 2. Enter PIN/password 3. Click SIGN IN 	
Expected Output	successful verification of the user credentials from the database, the user shall be directed to the dashboard.	
Actual Output	logged in successfully.	
Pass/Fail	PASSIVE/PASS	

Table 17 System Login Testing

7.1.2 Registration

Case Name:	Up Module testing	
Case ID:		
Description:	module shall enable user to register to the system by providing his credentials. As per the sequence of registration the credentials are stored within the database, noted that access shall be granted on the basis of these credentials.	
Testing technique used:	Black Box Testing	
Conditions:	System shall be operating normally and connectivity of system and database is already established.	
Test values:	Name Name Email Password	
Test Inputs:	Name; Type: Alphanumeric Name Type: Alphanumeric Email id: Format: abc@xyz.com Password: Type: Alphanumeric	
Steps:	<ol style="list-style-type: none"> 1. Enter First Name 2. Enter Last Name 3. Enter Email 4. Enter Password 5. Tap SIGN UP button 	
Expected Output	Credentials provided by user shall be stored at the database.	
Actual Output	Credentials provided by user are stored in database.	
Pass/Fail	Pass/POSITIVE	

Table 18 Sign up Feature Testing

7.1.3 Import Subjects

Case Name:	Import Subjects	
Case ID:		
Description:	This feature allows the import of bulk data through excel file.	
Testing technique used:	Black Box Testing	
Preconditions:	System is operating and database connectivity is established.	
Input values:	Select excel file from system. Submit imported file to the system. Page should be updated with the new records.	
Valid Inputs:	On-Map Location in terms of Longitudinal, Latitude coordinates Off-Map Location in terms of Longitudinal, Latitude coordinates Balance type: alphabets	
Steps	1 Open relevant Excel file 2 Add required data 3 Save and Upload to the system 4 Click Import	
Expected Output	Subject are imported and listing updated.	
Actual Output	Newly added records are displayed in the listing.	
Status	PASS	

Table 19 Subject Import

Chapter 8

8 Conclusion and Future Work

8.1 Conclusion

Our goal was to develop a system to find out innovative and creative solution for patients and emergency situations that affect us in different ways in our daily life. A system needs to be developed that will allow common people to call ambulance from a remote area using cell phones by revealing their exact location to the drivers of ambulance service.

We accomplished our objectives, successfully developing a Web Application that lets users to view the course performances. Similarly, the users can also see the course structures, current academic trends.

Due to constraints of time and team size, the scope of the project was kept small. Initially we will deploy the system in university, but its scope can be enhanced with the passage of time.

We firmly believe that our project can genuinely bring about a significant insight in the academic process.

8.2 Future work

Due to certain intrinsic limits in terms of project development time and team size, a lot of things had to be omitted from the scope of this project. However, this leaves room for a horde of enhancements, expansions and functionality add-on's.

The system can be enhanced with more features in the future. An AI based intelligent logic flow can be created which not only shows the trends in the systems data but also can predict the future directions by using the existing data.

Among minor changes, the application's User Interface could be modified to be even more user friendly and the application could be improved to run faster and enhance performance on lower end devices.

Glossary

API	Application Programming Interface
App	Application
AS	Assumption
Black box Testing	Testing emphasizes on the external behaviour of the software entity
CO	Constraints
App	Application
CEO	Chief Executive Officer
DBMS	Database Management System
DEP	Dependency
FRs	Functional Requirements
GUI	Graphical User Interface
IDE	Integrated Development Environment
MCS	Military College of Signals
NFRs	Non Functional Requirements
NUST	National University of Science and Technology
OE	Operating Environment
OS	Operating System
Parse	Cloud Server
REQ	Requirement
SQL	Structured Query Language
SR	Safety Requirements
SRS	Software Requirements Specification
UD	User Documentation
UML	Unified Modelling Language
White Box Testing	Testing emphasizes on the internal behaviour of the software entity

Table 21 Glossary

Bibliography

1. <https://docs.microsoft.com/>
2. <https://www.wikipedia.org/>
3. <https://moodle.com/>
4. <https://stackoverflow.com/>

Turnitin Plagiarism Report:

SPES

ORIGINALITY REPORT

15%

SIMILARITY INDEX

9%

INTERNET SOURCES

12%

PUBLICATIONS

10%

STUDENT PAPERS

PRIMARY SOURCES

1

**Submitted to Higher Education Commission
Pakistan**

Student Paper

2%

2

www.guru99.com

Internet Source

1%

3

www.eziline.com

Internet Source

1%

4

quod.lib.umich.edu

Internet Source

1%

5

Submitted to Curtin University of Technology

Student Paper

1%

9	Submitted to Strathmore University Student Paper	<1%
10	Submitted to Arab Open University Student Paper	<1%
11	Submitted to HELP UNIVERSITY Student Paper	<1%
12	Submitted to City University of Hong Kong Student Paper	<1%
13	Submitted to University of Ulster Student Paper	<1%
14	Submitted to Heriot-Watt University Student Paper	<1%
15	Submitted to Southampton Solent University Student Paper	<1%
16	Submitted to UT, Dallas Student Paper	<1%
17	Submitted to TechKnowledge Student Paper	<1%
18	Submitted to University of Wolverhampton Student Paper	<1%
19	Submitted to The American College of Greece Libraries Student Paper	<1%

20	Submitted to McMaster University Student Paper	<1%
21	Submitted to University of Huddersfield Student Paper	<1%
22	Submitted to University of Abertay Dundee Student Paper	<1%
23	Submitted to Queen Mary and Westfield College Student Paper	<1%
24	Submitted to University of Mauritius Student Paper	<1%
25	Submitted to University of East London Student Paper	<1%
26	Submitted to Papua New Guinea University of Technology Student Paper	<1%
27	Submitted to Swansea Metropolitan University Student Paper	<1%
28	Submitted to Mississippi State University Student Paper	<1%
29	Submitted to FPT Polytechnic Student Paper	<1%
30	Submitted to Birkbeck College Student Paper	<1%

31	Submitted to Sim University Student Paper	<1%
32	Submitted to Central Queensland University Student Paper	<1%
33	www.slideshare.net Internet Source	<1%
34	Submitted to Bournemouth University Student Paper	<1%
35	Submitted to University of Teesside Student Paper	<1%
36	spectrum.library.concordia.ca Internet Source	<1%
37	Submitted to University of Auckland Student Paper	<1%
38	Submitted to Kingston University Student Paper	<1%
39	Submitted to Waikato Institute of Technology Student Paper	<1%
40	Submitted to London School of Commerce Student Paper	<1%
41	Submitted to Letterkenny Institute of Technology Student Paper	<1%
42	Submitted to University of South Australia Student Paper	

		<1%
43	Submitted to Manchester Metropolitan University Student Paper	<1%
44	Submitted to University of Luton Student Paper	<1%
45	Submitted to University of Zululand Student Paper	<1%
46	Submitted to CITY College, Affiliated Institute of the University of Sheffield Student Paper	<1%
47	Submitted to Tshwane University of Technology Student Paper	<1%
48	Submitted to Segi University College Student Paper	<1%
49	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
50	Submitted to The Hong Kong Polytechnic University Student Paper	<1%
51	www.sixtblog.es Internet Source	<1%

52	Submitted to University of Technology, Sydney Student Paper	<1%
53	Submitted to Nottingham Trent University Student Paper	<1%
54	www.coursehero.com Internet Source	<1%
55	Submitted to President University Student Paper	<1%
56	Submitted to Al Ain University of Science and Technology Student Paper	<1%
57	Submitted to University of Western Ontario Student Paper	<1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On