EXPERIMENT 5

Aim

To draft the Software Requirements Specification Document for the Hotel Management System

Description

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

Draft a Software Requirement Specification Document for the University Registration System

1. Introduction
    - 1.1 Purpose
    - 1.2 Scope
    - 1.3 Definitions, Acronyms, and Abbreviations
    - 1.4 References
    - 1.5 Overview

2. Overall Description
    - 2.1 Product Perspective
        - 2.1.1 System Interfaces
        - 2.1.2 User Interfaces
        - 2.1.3 Hardware Interfaces
        - 2.1.4 Software Interfaces
        - 2.1.5 Communication Interfaces
        - 2.1.6 Memory Constraints
        - 2.1.7 Operations
        - 2.1.8 Site Adaptation Requirements
    - 2.2 Product Functions
    - 2.3 User Characteristics
    - 2.4 Constraints
    - 2.5 Assumptions and Dependencies
    - 2.6 Apportioning of Requirements

3. Specific Requirements
    - 3.1 External Interface
    - 3.2 Functions
    - 3.3 Performance Requirements
    - 3.4 Logical Database Requirements
    - 3.5 Design Constraints
        - 3.5.1 Standards Compliance

1. Introduction

This document aims at defining the overall software requirements for 'University Registration System'. Efforts have been made to define the requirements exhaustively and accurately. The final product will be having only features/ functionalities mentioned in this document and assumptions for any additional functionality/ feature should not be made by any of the parties involved in developing/ testing/ implementing/ using this product. In case it is required to have some additional features, a formal change request will need to be raised and subsequently a new release of this document and/ or product will be produced.

1.1 Purpose

This specification document describes the capabilities that will be provided by the software application 'University Registration System (URS) '. The URS maintains the information about students, faculty, various departments and corresponding courses. This information is maintained in order to provide registration to students and faculty.

1.2 Scope

The name of the software is University Registration System (URS). The system will be referred to as URS in the rest of the SRS. The proposed URS must be able to perform the following functions :

1. Issue of login ID and password to system operators.
2. Maintain details of students to register them into the university.
3. Maintain details of faculty in the university.
4. Maintain details of department in the university.
5. Maintain details of the courses allotted to the department and its corresponding scheme.
6. Search Department Details from the available list and view its details.
7. View Timetable allows the user to view timetables of different departments and students.
8. Select the courses offered by the user's department.
9. Payment of Registration Fees.
10. Print Registration Form.

Benefits:

1. The system involves minimal manual intervention in processing of data and therefore be free of errors.
2. It would allow efficient and much easier manipulation of details.
3. It would allow easier selection of courses.

1.3 Definitions, Acronyms, and Abbreviations

i. **SRS:** Software Requirements Specifications.
ii. **URS:** University Registration System.
iii. **System Operator:** System Administrator, Data Entry Operator.
iv. **RAM:** Random Access Memory.
v. **Student:** Any candidate admitted in a program offered by university.
vi. **System Administrator/Administrator:** Users having all the privileges to operate the UMS.
vii. **Data entry operator(DEO):** User having all the privileges to maintain student and faculty employee details.
viii. **Faculty:** Teaching staff of the university- Professor, Associate Professor, Assistant professor.
ix. **University:** Academic Unit that offers various programmes.

1.4 References

IEEE Recommended Practice for Software Requirement Specifications- IEEE Std 830-1998
Object Oriented Software Engineering- Yogesh Singh , Ruchika Malhotra

1.5 Overview

The rest of this SRS document describes the various system requirements, interfaces, features and functionalities in detail

2. Overall Description

This software will help in maintaining list of students, faculty, departments, courses and their schemes.

The URS maintains records of the students, students, faculty, departments, courses, schemes and the courses selected by the student.

The URS helps to select courses offered by the department and view their timetables.

The Administrator/DEO will have to maintain the following information:

- Student Details
- Faculty Details
- Department Details
- Course Details
- Scheme Details

The Faculty will have the following functions:

- Search Department
- View TimeTable

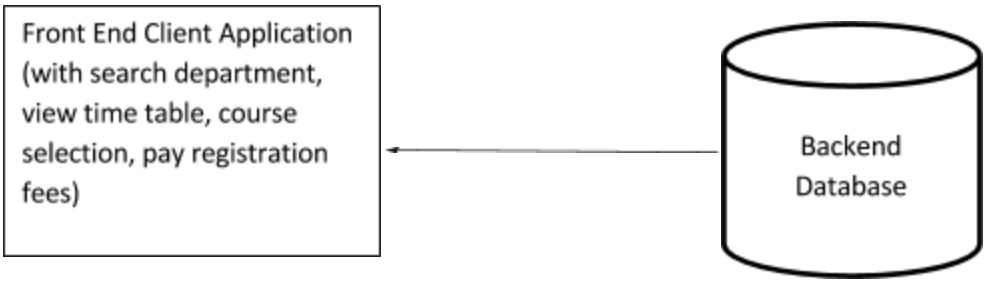
The Students will have the following functions:

- Course Selection
- Search Department

- View TimeTable
- Pay Registration Fees
- Print Registration Form

2.1 Product Perspective

The URS shall be built using client/server architecture and will be compatible with Microsoft Windows Operating System. The front-end of the system will be developed using Visual Basic 6.0 and backend will be developed using MYSQL Server 2005.



2.1.1 System Interfaces: None

2.1.2 User Interfaces

The URS will have the following user friendly and menu based interfaces:

i. **Login**: To allow the entry of only authorized users through valid username and password.
ii. **Student details:** To maintain details of all the students in the university.
iii. **Department details:** To maintain department details in the university.
iv. **Course details:** To maintain course details in each department.
v. **Scheme Details:** To maintain scheme details for each course.
vi. **Course Selection:** To allow user to select the courses.
vii. **Search Department:** To allow user to view details of the selected department.
viii. **Pay Registration Fees:** To allow user to pay the registration fees.

The following documents will be generated

- Registration Form
- Payment Receipt

2.1.3 Hardware Interfaces

1. The System must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN – LAN, Ethernet Cross-Cable.
2. Screen Resolution of at least 800X600- required for complete viewing of screens. Higher Resolution will not be a problem
3. Support for printer(dot matrix/deskJet/inkjet)- that is , appropriate drivers are installed and computer connected printer will be required to print the reports by the Administrator/Student.
4. Computer systems will be in a networked environment as it is a multi-user system.

2.1.4 Software Interfaces

i. MS Windows Operating System(NT/XP/Vista)
ii. Microsoft Visual Basic 6.0 for designing front end
iii. MS SQL Server 2005 for back end

2.1.5 Communication Interfaces: Communication will be via Internet.

2.1.6 Memory Constraints

At least 64 MB RAM and 500MB space of hard disk must be available at all times to ensure smooth functioning of the software

2.1.7 Operations: None.

2.1.8 Site Adaptation Requirements

The terminals at the client site will have to support the hardware and software interfaces specified in sections 2.1.3 and 2.1.4 respectively.

2.2 Product Functions

The URS will allow access only to the authorized users with specific roles(Administrator, DEO, Students and Faculty). Depending upon the user's role he/she will be able to access only specific modules of the system.

A summary of the major functions that the URS shall perform includes:

1. A login facility for enabling only authorized access to the system.
2. The system Administrator/DEO will be able to add, update, delete or view room student, faculty, department, course and scheme information.
3. The student will be able to pay their registration fees.
4. The student and faculty will be able to search department or view timetable.

2.3 User Characteristics

- *Qualification:* At least senior secondary; should be comfortable with English language
- *Experience:* Should be well versed about the basic outline of the Hotel Management
- *Technical Expertise:* Should be comfortable using general purpose applications on a computer.

2.4 Constraints

i. There will be only one administrator.
ii. The user will not be allowed to update the primary key.
iii. Since the DBMS being used is MS Access 2010 which is not a very powerful DBMS , it will not be able to store a large number of records
iv. Due to limited features of the DBMS database auditing will not be provided
v. Administrator must implement a security policy to safeguard the information of students, faculty, university details from being manipulated by unauthorized users.

2.5 Assumptions and Dependencies

I. The details related to the university will be provided by administrator only.
II. Administrator is created in the system already.
III. The login ID and password are generated and sent to the user via their email id.
IV. Section will be allotted to students automatically.
V. Faculty will be allotted to each section automatically.

2.6 Apportioning of Requirements: None

3. Specific Requirements

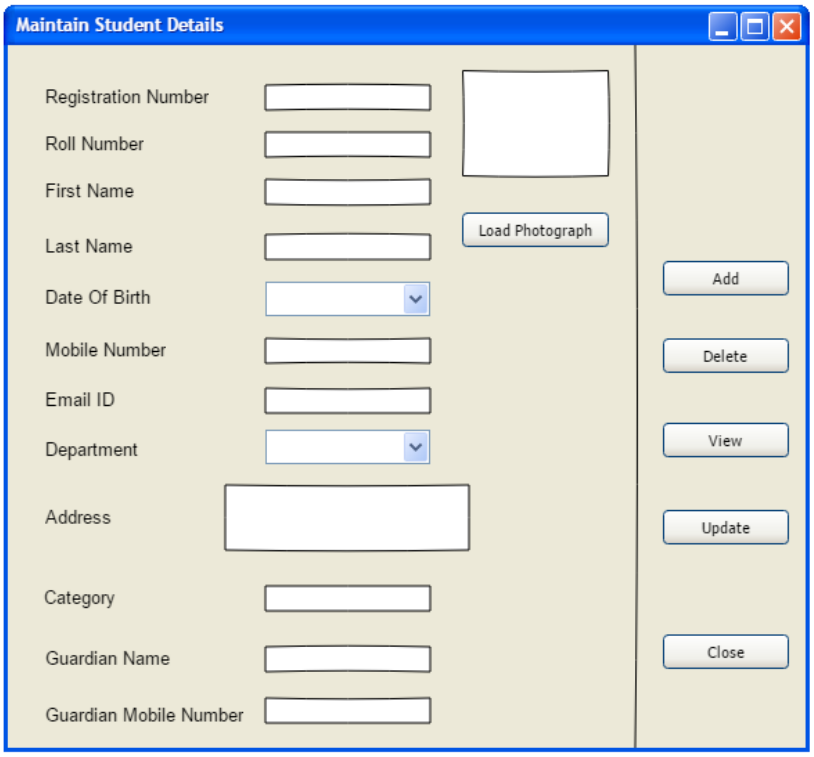
This section contains requirements in details along with various forms to be developed.

3.1 External Interface Requirements

3.1.1 User Interfaces

The following screens will be provided

1. **Maintain Student Details**



This form will be accessible system administrator.It will allow him/her to add/delete/modify student details.

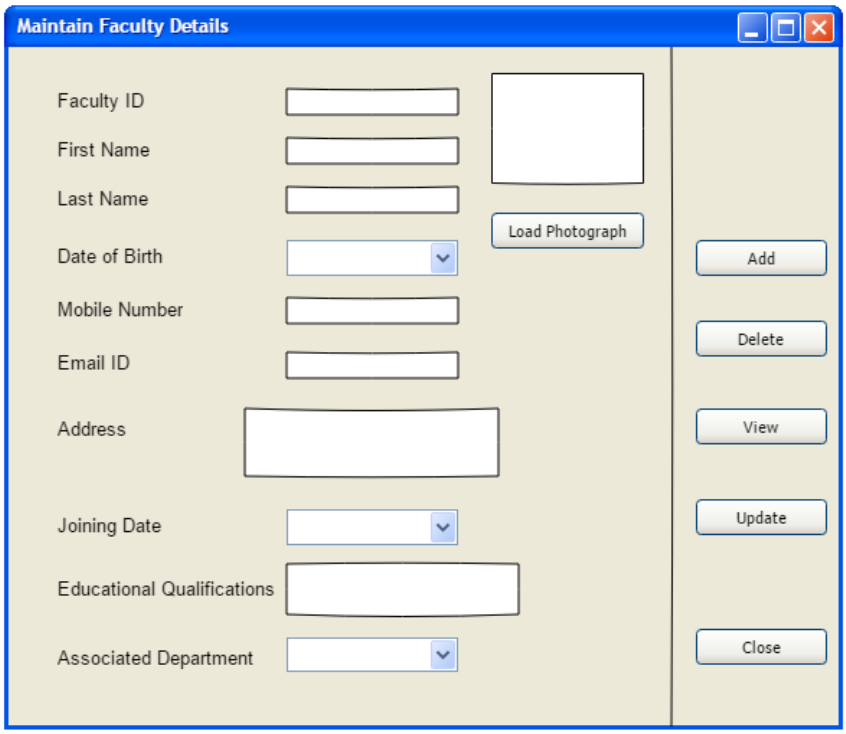
Various forms available on this form will be:

1. **Registration number:** Numeric and will have value from 100 to 19999.The system will not

allow the user to enter non-numeric characters and out-of-range numbers.

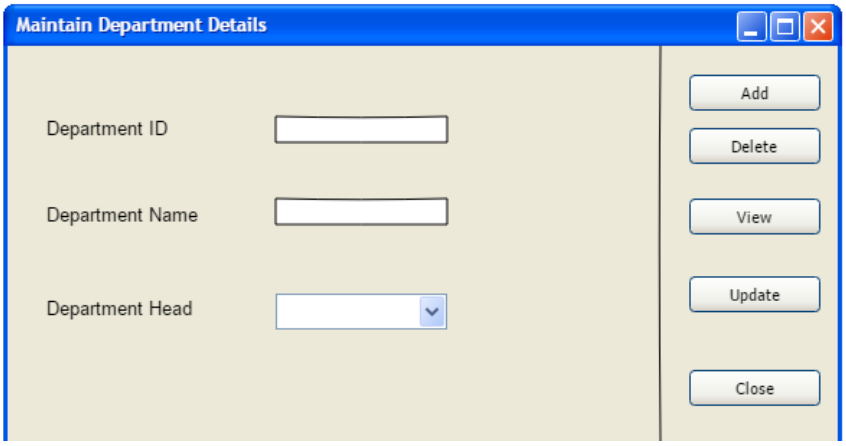
2. **Roll number:** Alphanumeric of length 11.The system will not allow the user to enter special characters except '/', blank spaces, out of range characters.
3. **First name:** Alphanumeric of length 1 to 100. special characters and spaces are not allowed.Numeric data will not be allowed.The system should not allow the user to enter special characters, spaces and out of range characters.
4. **Last name:** Alphanumeric of length 1 to 100. special characters and spaces are not allowed.Numeric data will not be allowed.The system should not allow the user to enter special characters, spaces and out of range characters.
5. **Date of Birth:** Will be of format mm/dd/yyyy. It will have 10 alphanumeric characters.
6. **Mobile number:** Numeric and of length should be exactly equals to 10 digit.system should not allow user to enter Alphabets and special characters and out of range or under range numbers.
7. **Email-id: :** Alphanumeric of length 1 to 1000. special characters are allowed and spaces are allowed.Numeric data will be allowed.The system should not allow the user to enter spaces and out of length characters.
8. **Branch:** A drop down menu for choosing branch from the available options.
9. **Address: :** Alphanumeric of length 1 to 1000. special characters and spaces are allowed.The system should not allow the user to enter out of range characters.
10. **category:** alphanumeric of length 1 to 200. special characters and spaces are not allowed.Numeric data not allowed.The system should not allow the user to enter numeric data and out of range characters.
11. **Guardian name**: alphanumeric of length 1 to 200. special characters and numeric data are not allowed spaces are allowed . system should not allow the user to enter numeric data and out of range characters.
12. **Guardian mobile number: :** Numeric and of length should be exactly equals to 10 digit.system should not allow user to enter Alphabets and special characters and out of range or under range numbers.

2. Maintain Faculty Details:



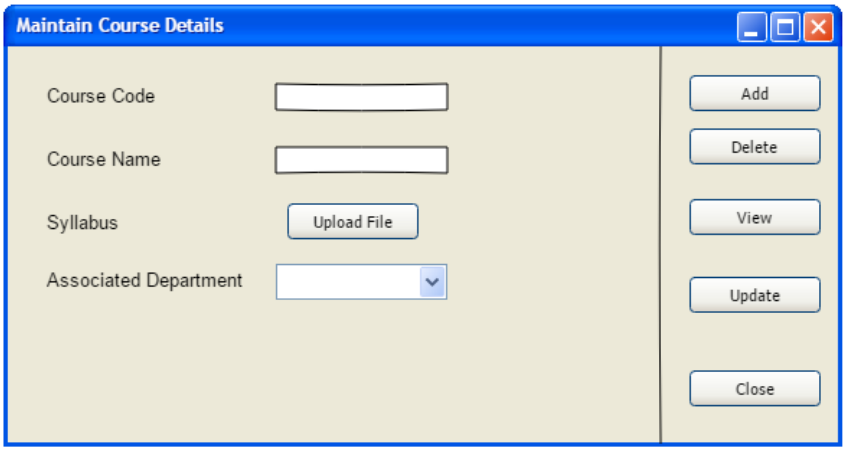
1. **Faculty Id:** Numeric and will have value from 100 to 19999.The system will not allow the user to enter non-numeric characters and out-of-range numbers.
2. **First name:** Alphanumeric of length 1 to 100. special characters and spaces are not allowed.Numeric data will not be allowed.The system should not allow the user to enter special characters, spaces and out of range characters.
3. **Last name:** Alphanumeric of length 1 to 100. special characters and spaces are not allowed.Numeric data will not be allowed.The system should not allow the user to enter special characters, spaces and out of range characters.
4. **Date of Birth:** Will be of format mm/dd/yyyy. It will have 10 alphanumeric characters.
5. **Mobile number:** Numeric and of length should be exactly equals to 10 digit.system should not allow user to enter Alphabets and special characters and out of range or under range numbers.
6. **Email-id:**Alphanumeric of length 1 to 1000. special characters are allowed and spaces are allowed.Numeric data will be allowed.The system should not allow the user to enter spaces and out of length characters.
7. **Address: :** Alphanumeric of length 1 to 1000. special characters and spaces are allowed.The system should not allow the user to enter out of range characters.
8. **Joining Date:** A drop down menu for choosing the date of joining should be present.
9. **Associated Department:** Alphanumeric of length 1 to 1000. special characters and spaces are not allowed.Numeric data will not be allowed.The system should not allow the user to numeric data,special characters , spaces and out of length characters.

3. Maintain Department Details



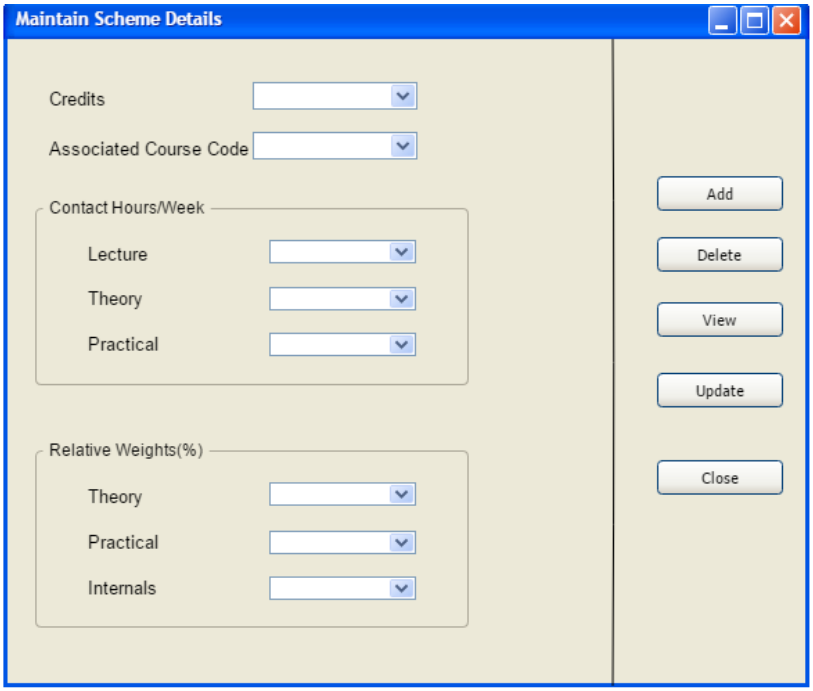
1. **Department Id:** Numeric and will have value from 100 to 19999.The system will not allow the user to enter non-numeric characters and out-of-range numbers.
2. **Department Name:** Alphanumeric of length 1 to 100. special characters and spaces are not allowed.Numeric data will not be allowed.The system should not allow the user to enter special characters, spaces and out of range characters.
3. **Department Head:** A drop down menu to choose the department head respectively for particular department.

4. Maintain Course Details



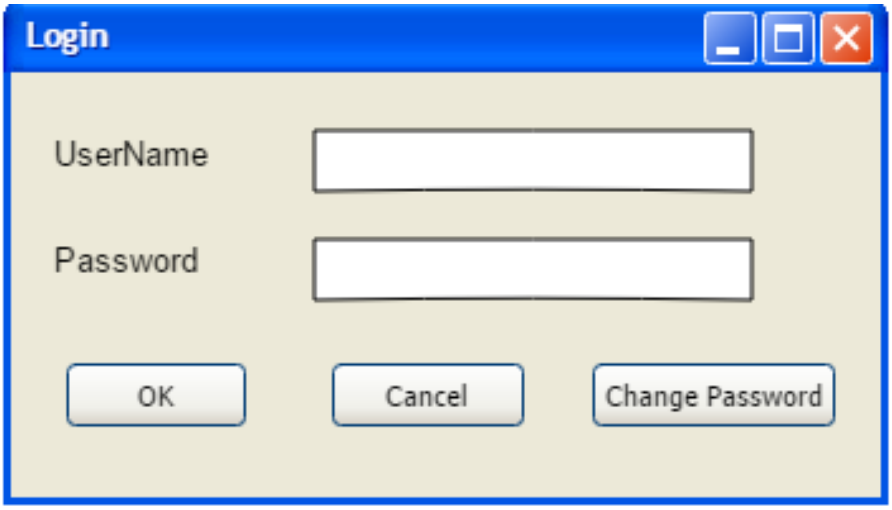
1. **Course Id:** Numeric and will have value from 100 to 500.The system will not allow the user to enter non-numeric characters,special characters and out-of-range numbers.
2. **Course Name::** Alphanumeric of length 1 to 100. special characters are not allowed and spaces are allowed.Numeric data will not be allowed.The system should not allow the user to enter special characters and out of range characters.
3. **Syllabus:** A button through which user will be able to upload the syllabus in pdf format only.
4. **Associated department:** A drop down menu for selecting the appropriate department for the particular course.

5. Maintain Scheme Details



1. **Credits:** A drop down menu for selecting the appropriate number of credits for a particular course.
2. **Associate course code:** A drop down menu for selecting the appropriate course code.
3. **Lecture:** A drop down menu for selecting the appropriate number of lecture hours per week for the chosen course.
4. **Theory:** A drop down menu for selecting the appropriate number of theory hours per week for the chosen course.
5. **practical:** A drop down menu for selecting the appropriate number of practical hours per week for the chosen course.
6. **Theory(Relative Weights%):**A drop down menu for selecting the appropriate percentage of theory marks for the chosen course.
7. **Practical(Relative Weights%):**A drop down menu for selecting the appropriate percentage of practical marks for the chosen course.
8. **Internals(Relative Weights%):**A drop down menu for selecting the appropriate percentage of internal marks for the chosen course.

6. Login

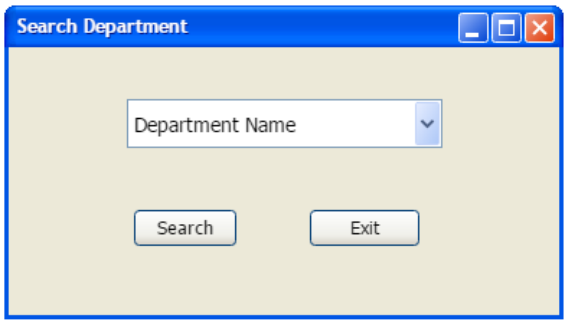


1. **Username:** Alphanumeric of length 1 to 100. special characters and spaces are not allowed.Numeric data will not be allowed.The system should not allow the user to enter special characters, spaces and out of range characters.
2. **Password:** Alphanumeric of length 1 to 50.The system should not allow the user to enter out of range characters.

7. Change Password

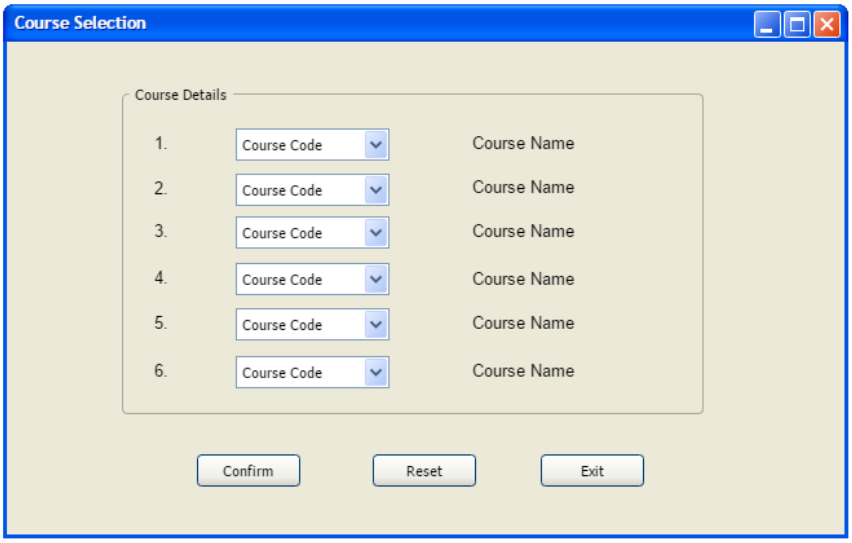
1. **Username:** Alphanumeric of length 1 to 100. special characters and spaces are not allowed.Numeric data will not be allowed.The system should not allow the user to enter special characters, spaces and out of range characters.
2. **Old Password:** Alphanumeric of length 1 to 50.The system should not allow the user to enter out of range characters.
3. **New Password:** Alphanumeric of length 1 to 50.The system should not allow the user to enter out of range characters.
4. **Confirm Password:** Alphanumeric of length 1 to 50.The system should not allow the user to enter out of range characters.The contents of this field must match with the contents of new password field.

8. Search Department



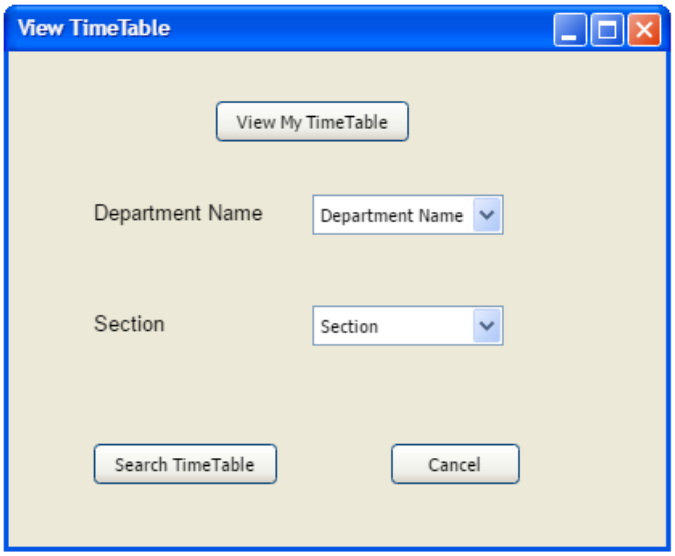
1.**Department Name:** A drop down menu to choose the name of a department from the available list.

9. Course Selection



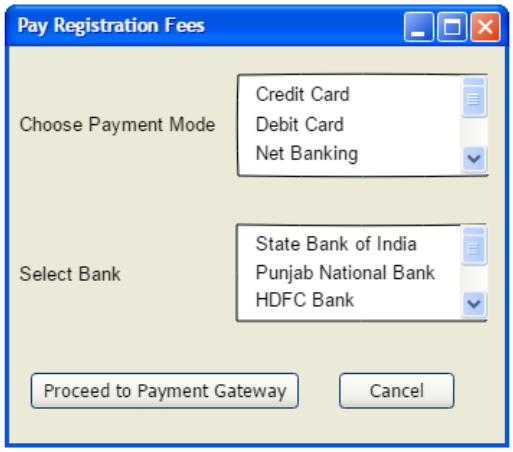
1. **Course Code:** Six drop down menus for choosing the appropriate course as per the students choices.

10. View TimeTable



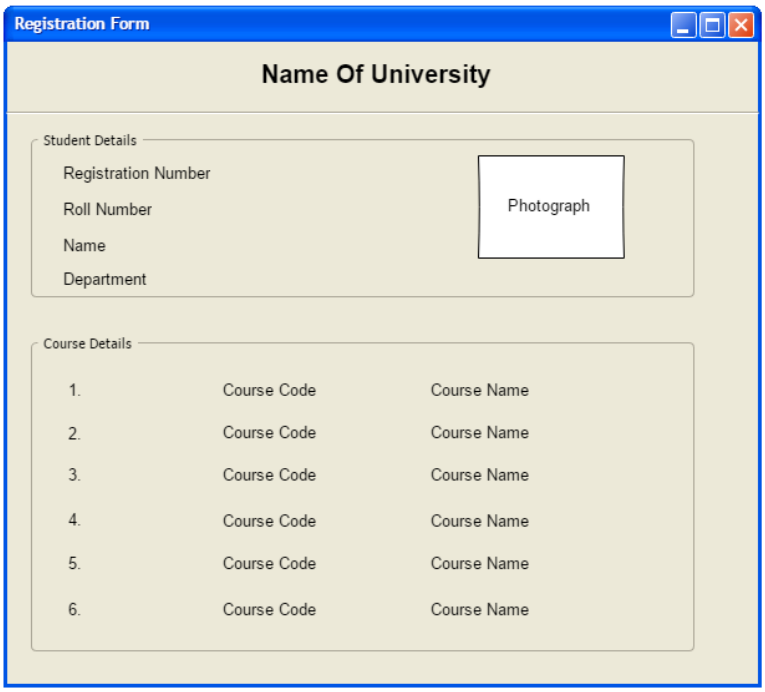
1. **Department Name:**A drop down menu for choosing the appropriate department as per choice from available options.
2. **Section:** A drop down menu for choosing the appropriate section as per choice from the available options.

11. Pay Registration fees

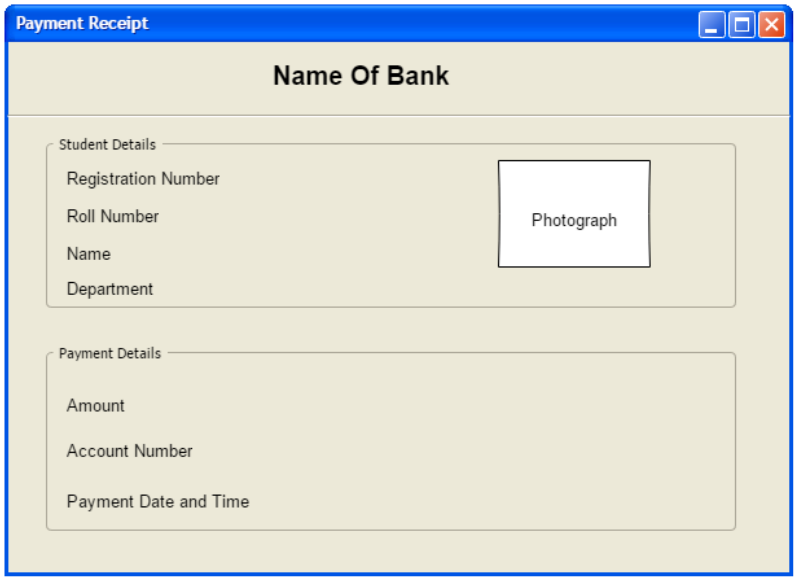


1. **Choose payment method**: Select one of the available payment options from the drop down list.
2. **Select bank**: Select one of the available banks from the drop down list.

12. Print Registration Form



13. Payment Receipt Generation



3.2 Functional Requirements

USE CASE DESCRIPTION : University Registration System

1. LOGIN

Introduction: The use case documents the steps that must be followed in order to login into the system.
Actors: Administrator, Student, Faculty, DEO
Pre-Condition: The administrator/ student/ faculty must be registered onto the system before the use case begins.
Post Conditions: If the use case is successful, the administrator/ student/ faculty is logged into the system else the system state remains unchanged.
Event Flow **Basic Flow** 1. The system requests that the actor enter his username, password. The role of the user is determined based on the username. 2. The Administrator/Student/Faculty/DEO enters his/her username, password. 3. The system validates the entered credentials and logs the actor into the system.
Alternative Flow **Alternative Flow 1: Invalid Login Details** If in the basic flow, the actor enters an invalid username, password the system displays an error message. The user can choose to either return to the beginning of the basic flow or cancel the login at which point the Use Case ends. **Alternative Flow 2: User Exits** This allows the user to exit at any time during the use case. The use case ends
Special Requirement: None
Associated Use Case: None

B. Validity Checks

1. Every user will have a unique username.
2. UserName can't be blank
3. UserName can have only 4-15 characters
4. UserName will not accept blank spaces and special characters
5. Password can't be blank.
6. Length of password can only be from 4-15 characters
7. Alphanumeric characters, hyphen and underscore are allowed in the password field
8. Password will not accept blank spaces

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

2. MAINTAIN STUDENT DETAILS

Introduction: The use case documents the steps that the administrator must follow in order to maintain student details and add, update, view and delete student information.

Actors: Administrator, DEO

Pre-Condition: The Administrator/DEO must be logged into the system before this use case begins.

Post Conditions: If the use case is successful, then the student information is added, updated, deleted or viewed else the system state remains unchanged.

Event Flow

Basic Flow:

This use case starts when the Administrator/DEO wishes to add/update/delete/view student information.

1. The system requests that the Administrator/DEO specify the function he/she would like to perform (add a student, update a student, view a student or delete a student).
2. Once the Administrator/DEO provides the requested information, one of the sub flows is executed:
    a. If the Administrator/DEO selects "Add a Student", the **Add a Student** subflow is executed.
    b. If the Administrator/DEO selects "Update a Student", the **Update a Student** subflow is executed.
    c. If the Administrator/DEO selects "Delete a Student", the **Delete a Student** subflow is executed.
    d. If the Administrator/DEO selects "View a Student", the **View a Student** subflow is executed.

Basic Flow 1: Add a Student

The system request that the Administrator/DEO enter the student information. This includes:

- Registration Number
- Roll Number
- Student name (First Name, Last Name)
- Photograph
- Date of Birth
- Mobile Number
- Email ID
- Branch
- Address
- Category
- Guardian Name
- Guardian Mobile Number

Once the Administrator/DEO provides the requested information, the student is added to the system.

Basic Flow 2: Update a Student

1. The system requests that the Administrator/DEO enter Registration Number.
2. The Administrator/DEO enters the Registration Number.
3. The system retrieves and displays the student information.
4. The Administrator/DEO makes the desired changes to the student information.
5. Once the Administrator/DEO updates the necessary information, the system updates the student information with the updated information.

Basic Flow 3: Delete a Student

1. The system requests that the Administrator/DEO enter the Registration Number.
2. The Administrator/DEO enters the Registration Number.

3. The system retrieves and displays the student information.
4. The system prompts the Administrator/DEO to confirm the deletion of the student record.
5. The Administrator/DEO verifies the deletion.
6. The system deletes the record.

Basic Flow 4: View a Student

1. The system requests that the Administrator/DEO to enter the Registration Number.
2. The system retrieves and displays the faculty information.

Alternative Flow

Alternate Flow 1: Invalid Entry
If in the **Add a Student** or **Update a Student** flow, the Administrator/DEO enters invalid Registration Number/ Roll Number/ Student Name/ Date of Birth/ Mobile Number/ Email ID/ Address/ Branch/ Category/ Guardian Details or leaves them empty, the system displays an appropriate error message. The Administrator/DEO returns to the basic flow and may re-enter the invalid entry.

Alternate Flow 2: Student Already Exists
If in the **Add a Student** flow, a student with a specified student ID already exists, the system displays an error message. The Administrator/DEO returns to the basic flow and may re-enter the Registration Number.

Alternate Flow 3: Student not Found
If in the **Update a Student** or **Delete a Student** or **View a Student** flow, the student information with the specified code does not exist, the system displays an error message. The Administrator/DEO returns to the basic flow and may re-enter the Registration Number.

Alternate Flow 4: Update Cancelled
If in the **Update a Student** flow, the Administrator/DEO decides not to update the student, the update is cancelled and the basic flow is re-started at the beginning.

Alternate Flow 5: Delete Cancelled
If in the **Delete a Student** flow, the Administrator/DEO decides not to delete the student, the delete is cancelled and the basic flow is re-started at the beginning.

Alternate Flow 6: User Exits
This allows the user to exit at any time during the use case. The use case ends.

Special Requirement: None

Associated Use Case: Login

B. Validity Checks

1. Only the Administrator/DEO will be authorized to access the maintain student details module
2. Every registered student will have a unique registration number and roll number.
3. Registration Number and roll number cannot be blank.
4. Name(First Name, Last Name) cannot be blank.
5. Date of Birth cannot be blank.
6. Mobile Number cannot be blank.
7. Email ID cannot be blank.

8. The field to choose Department cannot be blank
9. Category cannot be blank

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

3. MAINTAIN FACULTY DETAILS

Introduction: The use case documents the steps that the administrator must follow in order to maintain faculty details and add, update, view and delete faculty information.

Actors: Administrator, DEO

Pre-Condition: The Administrator must be logged into the system before this use case begins.

Post Conditions: If the use case is successful, then the faculty information is added, updated, deleted or viewed else the system state remains unchanged.

Event Flow

Basic Flow:

This use case starts when the Administrator/DEO wishes to add/update/delete/view faculty information.

1. The system requests that the Administrator/DEO specify the function he/she would like to perform (add a faculty, update a faculty, view a faculty or delete a faculty).
2. Once the Administrator/DEO provides the requested information, one of the sub flows is executed:
    a. If the Administrator/DEO selects "Add a Faculty", the **Add a Faculty** subflow is executed.
    b. If the Administrator/DEO selects "Update a Faculty", the **Update a Faculty** subflow is executed.
    c. If the Administrator/DEO selects "Delete a Faculty", the **Delete a Faculty** subflow is executed.
    d. If the Administrator/DEO selects "View a Faculty", the **View a Faculty** subflow is executed.

Basic Flow 1: Add a Faculty

The system request that the Administrator/DEO enter the faculty information. This includes:

- Faculty ID
- Faculty name (First Name, Last Name)
- Photograph
- Date of Birth
- Mobile Number
- Email ID
- Address
- Joining Date
- Education Qualification
- Associated Department

Once the Administrator/DEO provides the requested information, the faculty is added to the system.

Basic Flow 2: Update a Faculty

1. The system requests that the administrator enter the faculty ID.
2. The administrator enters the faculty ID.

3. The system retrieves and displays the faculty information.
4. The administrator makes the desired changes to the faculty information.
5. Once the administrator updates the necessary information, the system updates the faculty information with the updated information.

Basic Flow 3: Delete a Faculty

1. The system requests that the administrator enter the faculty ID.
2. The administrator enters the faculty ID.
3. The system retrieves and displays the faculty information.
4. The system prompts the administrator to confirm the deletion of the faculty record.
5. The administrator verifies the deletion.
6. The system deletes the record.

Basic Flow 4: View a Faculty

1. The system requests that the administrator to enter the faculty ID.
2. The system retrieves and displays the faculty information.

Alternative Flow

Alternate Flow 1: Invalid Entry
If in the **Add a Faculty** or **Update a Faculty** flow, the administrator enters invalid Faculty ID/ Name/ Department/ Education Qualification/ Date of Birth/ Mobile Number/ Email ID/ Address/ Joining Date or leaves them empty, the system displays an appropriate error message. The administrator returns to the basic flow and may re-enter the invalid entry.

Alternate Flow 2: Faculty Already Exists
If in the **Add a Faculty** flow, a faculty with a specified faculty ID already exists, the system displays an error message. The administrator returns to the basic flow and may re-enter the faculty code.

Alternate Flow 3: Faculty not Found
If in the **Update a Faculty** or **Delete a Faculty** or **View a Faculty** flow, the faculty information with the specified code does not exist, the system displays an error message. The administrator returns to the basic flow and may re-enter the faculty ID.

Alternate Flow 4: Update Cancelled
If in the **Update a Faculty** flow, the administrator decides not to update the faculty, the update is cancelled and the basic flow is re-started at the beginning.

Alternate Flow 5: Delete Cancelled
If in the **Delete a Faculty** flow, the administrator decides not to delete the faculty, the delete is cancelled and the basic flow is re-started at the beginning.

Alternate Flow 6: User Exits
This allows the user to exit at any time during the use case. The use case ends.

Special Requirement: None

Associated Use Case: Login

B. Validity Checks

1. Only the Administrator will be authorized to access the maintain faculty details module
2. Every registered faculty will have a unique faculty id.
3. Faculty ID cannot be blank.
4. Name(First Name, Last Name) cannot be blank.

5. Date of Birth cannot be blank.
6. Mobile Number cannot be blank.
7. Email ID cannot be blank.
8. Joining Date cannot be blank.
9. Educational Qualifications cannot be blank.
10. The field to choose Associated Department cannot be blank.

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

4. MAINTAIN DEPARTMENT DETAILS

Introduction: The use case documents the steps that the administrator must follow in order to maintain department details and add, update, view and delete department information.
Actors: Administrator, DEO
Pre-Condition: The Administrator must be logged into the system before this use case begins.
Post Conditions: If the use case is successful, then the department information is added, updated, deleted or viewed else the system state remains unchanged.
Event Flow **Basic Flow**: This use case starts when the Administrator/DEO wishes to add/update/delete/view department information. 1. The system requests that the Administrator/DEO specify the function he/she would like to perform (add a department , update a department , view a department or delete a department). 2. Once the administrator provides the requested information, one of the sub flows is executed: a. If the Administrator/DEO selects "Add a Department", the **Add a Department** subflow is executed. b. If the Administrator/DEO selects "Update a Department", the **Update a Department** subflow is executed. c. If the Administrator/DEO selects "Delete a Department", the **Delete a Department** subflow is executed. d. If the Administrator/DEO selects "View a Department", the **View a Department** subflow is executed. **Basic Flow 1: Add a Department** The system request that the administrator enter the department information. This includes: ● Department ID ● Department name ● Department Head Once the administrator provides the requested information, the department is added to the system. **Basic Flow 2: Update a Department** 1. The system requests that the administrator enter the Department ID. 2. The administrator enters the Department ID. 3. The system retrieves and displays the department information. 4. The administrator makes the desired changes to the department information.

5. Once the administrator updates the necessary information, the system updates the department information with the updated information.

Basic Flow 3: Delete a Department

1. The system requests that the administrator enter the Department ID.
2. The administrator enters the Department ID.
3. The system retrieves and displays the department information.
4. The system prompts the administrator to confirm the deletion of the department record.
5. The administrator verifies the deletion.
6. The system deletes the record.

Basic Flow 4: View a Department

1. The system requests that the administrator to enter the Department ID.
2. The system retrieves and displays the department information.

Alternative Flow

Alternate Flow 1: Invalid Entry
If in the **Add a Department** or **Update a Department** flow, the administrator enters invalid Department ID/ Department Name/ Department Head or leaves them empty, the system displays an appropriate error message. The administrator returns to the basic flow and may re-enter the invalid entry.

Alternate Flow 2: Department Already Exists
If in the **Add a Department** flow, a department with a specified Department ID already exists, the system displays an error message. The administrator returns to the basic flow and may re-enter the Department ID.

Alternate Flow 3: Department not Found
If in the **Update a Department** or **Delete a Department** or **View a Department** flow, the department information with the specified code does not exist, the system displays an error message. The administrator returns to the basic flow and may re-enter the Department ID.

Alternate Flow 4: Update Cancelled
If in the **Update a Department** flow, the administrator decides not to update the department , the update is cancelled and the basic flow is re-started at the beginning.

Alternate Flow 5: Delete Cancelled
If in the **Delete a Department** flow, the administrator decides not to delete the department , the delete is cancelled and the basic flow is re-started at the beginning.

Alternate Flow 6: User Exits
This allows the user to exit at any time during the use case. The use case ends.

Special Requirement: None

Associated Use Case: Login

B. Validity Checks

1. Only the Administrator will be authorized to access the maintain department details module.
2. Every department will have a unique Department ID.
3. Department ID cannot be blank.
4. Department Name cannot be blank.
5. Department Head cannot be blank.

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

5. MAINTAIN COURSE DETAILS

Introduction: The use case documents the steps that the administrator must follow in order to maintain course details and add, update, view and delete course information.
Actors: Administrator, DEO
Pre-Condition: The Administrator must be logged into the system before this use case begins.
Post Conditions: If the use case is successful, then the course information is added, updated, deleted or viewed else the system state remains unchanged.
Event Flow **Basic Flow**: This use case starts when the administrator wishes to add/update/delete/view course information. 1. The system requests that the administrator specify the function he/she would like to perform (add a course , update a course , view a course or delete a course). 2. Once the administrator provides the requested information, one of the sub flows is executed: a. If the administrator selects “Add a Course ”, the **Add a Course** subflow is executed. b. If the administrator selects “Update a Course ”, the **Update a Course** subflow is executed. c. If the administrator selects “Delete a Course ”, the **Delete a Course** subflow is executed. d. If the administrator selects “View a Course ”, the **View a Course** subflow is executed. **Basic Flow 1: Add a Course** The system request that the administrator enter the course information. This includes: ● Course Code ● Course Name ● Syllabus ● Associated Department Once the administrator provides the requested information, the course is added to the system. **Basic Flow 2: Update a Course** 1. The system requests that the administrator enter the Course Code. 2. The administrator enters the Course Code. 3. The system retrieves and displays the course information. 4. The administrator makes the desired changes to the course information. 5. Once the administrator updates the necessary information, the system updates the course information with the updated information. **Basic Flow 3: Delete a Course** 1. The system requests that the administrator enter the Course Code. 2. The administrator enters the Course Code. 3. The system retrieves and displays the course information. 4. The system prompts the administrator to confirm the deletion of the course record.

5. The administrator verifies the deletion.
6. The system deletes the record.

Basic Flow 4: View a Course

1. The system requests that the administrator to enter the Course Code.
2. The system retrieves and displays the course information.

Alternative Flow

Alternate Flow 1: Invalid Entry
If in the **Add a Course** or **Update a Course** flow, the administrator enters invalid Course Code/ Course Name/ Associated Department or leaves them empty, the system displays an appropriate error message. The administrator returns to the basic flow and may re-enter the invalid entry.

Alternate Flow 2: Course Already Exists
If in the **Add a Course** flow, a course with a specified Course Code already exists, the system displays an error message. The administrator returns to the basic flow and may re-enter the Course Code.

Alternate Flow 3: Course not Found
If in the **Update a Course** or **Delete a Course** or **View a Course** flow, the course information with the specified code does not exist, the system displays an error message. The administrator returns to the basic flow and may re-enter the Course Code.

Alternate Flow 4: Update Cancelled
If in the **Update a Course** flow, the administrator decides not to update the course, the update is cancelled and the basic flow is re-started at the beginning.

Alternate Flow 5: Delete Cancelled
If in the **Delete a Course** flow, the administrator decides not to delete the course, the delete is cancelled and the basic flow is re-started at the beginning.

Alternate Flow 6: User Exits
This allows the user to exit at any time during the use case. The use case ends.

Special Requirement: None

Associated Use Case: Login

B. Validity Checks

1. Only the Administrator will be authorized to access the maintain course details module.
2. Every course will have a unique Course Code.
3. Course Code cannot be blank.
4. Course Name cannot be blank.
5. Syllabus cannot be blank.
6. The field to choose Associated Department cannot be blank

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

6. MAINTAIN SCHEME DETAILS

Introduction: The use case documents the steps that the administrator must follow in order to maintain scheme details and add, update, view and delete scheme information.

Actors: Administrator, DEO

Pre-Condition: The Administrator must be logged into the system before this use case begins.

Post Conditions: If the use case is successful, then the scheme information is added, updated, deleted or viewed else the system state remains unchanged.

Event Flow

Basic Flow:

This use case starts when the administrator wishes to add/update/delete/view scheme information.

1. The system requests that the administrator specify the function he/she would like to perform (add a scheme, update a scheme, view a scheme or delete a scheme).
2. Once the administrator provides the requested information, one of the sub flows is executed:
    a. If the administrator selects “Add a Scheme”, the **Add a Scheme** subflow is executed.
    b. If the administrator selects “Update a Scheme”, the **Update a Scheme** subflow is executed.
    c. If the administrator selects “Delete a Scheme”, the **Delete a Scheme** subflow is executed.
    d. If the administrator selects “View a Scheme”, the **View a Scheme** subflow is executed.

Basic Flow 1: Add a Scheme

The system request that the administrator enter the scheme information. This includes:

- Credits
- Associated Course Code
- Contact Hours / Week Details (Lecture, Theory, Internals)
- Relative Weights (Theory, Practical, Internals)

Once the administrator provides the requested information, the scheme is added to the system.

Basic Flow 2: Update a Scheme

1. The system requests that the administrator enter Course Code
2. The administrator enters the Course Code.
3. The system retrieves and displays the scheme information.
4. The administrator makes the desired changes to the scheme information.
5. Once the administrator updates the necessary information, the system updates the scheme information with the updated information.

Basic Flow 3: Delete a Scheme

1. The system requests that the administrator enter the Course Code.
2. The administrator enters the Course Code.
3. The system retrieves and displays the scheme information.
4. The system prompts the administrator to confirm the deletion of the scheme record.
5. The administrator verifies the deletion.
6. The system deletes the record.

Basic Flow 4: View a Scheme

1. The system requests that the administrator to enter the Course Code.

2. The system retrieves and displays the scheme information.
Alternative Flow **Alternate Flow 1: Invalid Entry** If in the **Add a Scheme** or **Update a Scheme** flow, the administrator enters invalid Credits/ Associated Course Code/ Contact Hours/ Relative Weights or leaves them empty, the system displays an appropriate error message. The administrator returns to the basic flow and may re-enter the invalid entry. **Alternate Flow 2: Scheme Already Exists** If in the **Add a Scheme** flow, a scheme with a specified Course Code already exists, the system displays an error message. The administrator returns to the basic flow and may re-enter the Course Code. **Alternate Flow 3: Scheme not Found** If in the **Update a Scheme** or **Delete a Scheme** or **View a Scheme** flow, the scheme information with the specified code does not exist, the system displays an error message. The administrator returns to the basic flow and may re-enter the Course Code. **Alternate Flow 4: Update Cancelled** If in the **Update a Scheme** flow, the administrator decides not to update the scheme, the update is cancelled and the basic flow is re-started at the beginning. **Alternate Flow 5: Delete Cancelled** If in the **Delete a Scheme** flow, the administrator decides not to delete the scheme, the delete is cancelled and the basic flow is re-started at the beginning. **Alternate Flow 6: User Exits** This allows the user to exit at any time during the use case. The use case ends.
Special Requirement: None
Associated Use Case: Login

B. Validity Checks

1. Only the Administrator will be authorized to access the maintain scheme details module.
2. Every scheme will have a unique Course Code.
3. Credits cannot be blank.
4. Course Code cannot be blank.
5. Contact Hours/Week fields cannot be blank.
6. Syllabus cannot be blank.
7. The field to choose Associated Department cannot be blank

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

7. SEARCH DEPARTMENT

Introduction: This use case documents the steps that must be followed to search department.
Actors: Administrator, Student, Faculty
Pre-Condition: The Administrator/Student/Faculty must be logged into the system before this use case begins.
Post Conditions: If the use case is successful, the department details are displayed else an error message is displayed.
Event Flow **Basic Flow:** This use case starts when the Administrator/Student/Faculty wishes to view details of a department. 1. The Administrator/Student/Faculty selects a department from the list. 2. The Department details are displayed.
Alternative Flow **Alternative Flow 1: Department Details are not available** If the Department details are not available, an error message is shown. The user can either choose to return to the basic flow of the use case or the use case ends. **Alternative Flow 2: User Exits** If at any point of time, the Administrator/Student/Faculty decides not to view the department details, the use case ends.
Special Requirement: None
Associated Use Case: Login

B. Validity Checks

1. Department Name cannot be blank.

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

8. VIEW TIMETABLE

Introduction: This use case documents the steps that must be followed to view time table.
Actors: Administrator, Student, Faculty
Pre-Condition: The Administrator/Student/Faculty must be logged into the system before this use case begins..
Post Conditions: If the use case is successful, the department details are displayed else an error message is displayed.
Event Flow

Basic Flow: This use case starts when the Administrator/Student/Faculty wishes to view the timetable. 1. The Administrator/Student/Faculty selects a department and section from the list and the time table is displayed. 2. The Faculty can directly view his/her time table.
Alternative Flow **Alternative Flow 1: Time Table is not available** If the Time Table of the selected department is not available, an error message is shown. The user can either choose to return to the basic flow of the use case or the use case ends. **Alternative Flow 2: User Exits** If at any point of time, the Administrator/Student/Faculty decides not to view the timetable, the use case ends.
Special Requirement: None
Associated Use Case: Login

9. COURSE SELECTION

Introduction: This use case documents the steps that must be followed for a student to select course.
Actors: Administrator, Student
Pre-condition: The Administrator/Student must be logged into the system.
Post-condition: If the use case is successful, the student course details are updated in the system else the system state remains unchanged.
Basic flow: 1. The student selects the courses allotted to the department of the student. 2. Submit button is clicked and the courses are updated in the system.
Alternate flow 1: No Seats Available 1. If the seats corresponding to any course selected is full, a popup message is shown containing the name of the course having no seats. The user returns to the basic flow of the use case. **Alternate flow 2: User Exits** 1. The user decides to cancel the course selection procedure and exits from the system. The use case ends.
Special Requirements: None
Associated use cases: Login

B. Validity Checks

1. Only the Administrator/Student will be authorized to access the maintain scheme details module.
2. Every course has a unique Course Code.
3. All the Course Code fields cannot be blank.
4. All the Course Code fields must have different Course Codes. No field can have the same code.

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

10. PAY REGISTRATION FEES

Brief Description: This use case documents the steps that must be followed to pay the registration fees.
Actors: Administrator, Student
Pre-Conditions : The Administrator/Student must be logged into the system
Post Conditions: If the use case is successful, the Administrator/Student is able to pay the registration fees, if any.
Flow of Events **Basic Flow** 1. The Student/Administrator selects the option to pay the registration fees. 2. The system displays the registration fees of the student inclusive of all taxes as the case may be. 3. The Student pays the bill using online modes of payment- Credit Card/Debit Card/Net Banking
Alternative Flow 1: Invalid Details If in the basic flow, the Student/Administrator enters an invalid information the system displays an error message. The user can choose to either return to the beginning of the basic flow or cancel the payment process at which point the Use Case ends. **Alternative Flow 2: User Exits** This allows the user to exit at any time during the use case. The use case ends
Special Requirements: None
Associated Use Case: Login, Receipt Generation

B. Validity Checks

1. Payment Mode cannot be blank.
2. Bank cannot be blank.

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

11. RECEIPT GENERATION

Introduction: This use case generates a receipt after successful payment by the user.
Actors: Administrator, Student
Pre-condition: The customer must be logged into the system. A successful payment should be made
Post-condition: A receipt is generated and sent to the student.
Basic flow: 1. After successful payment by the user an invoice consisting of details such as payment reference id, date, time, bank details is generated.

2. The receipt is emailed to the student.
Alternate flow 1: User Exits The user decides to cancel the post and exists from the system. The use case ends.
Special Requirements: None
Associated use cases: Login

B. Validity Checks

1. The receipt will only be downloaded when the payment will be successful.

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

12. PRINT REGISTRATION FORM

Introduction: This use case generates a registration after successful payment of registration fees and course selection.
Actors: Administrator, Student
Pre-condition: The customer must be logged into the system. A successful payment should be made. Courses should have been selected.
Post-condition: A registration form is downloaded onto the user's system.
Basic flow: 1. A registration form is generated containing student details and courses selected.
Alternate Flow 1: Courses not have been selected If the user has not selected the courses, an error message is shown to the user and the use case ends. **Alternate Flow 2: Registration Fees not paid** If the user has not yet paid the registration fees, an error message is shown to the user and the use case ends. **Alternate flow 3: User Exits** The user decides to cancel the post and exists from the system. The use case ends.
Special Requirements: None
Associated use cases: Login , Pay Registration Fees

B. Validity Checks

1. The registration form will only be downloaded when the courses have been confirmed.

C. Sequencing Information: None

D. Error Handling/ Response to abnormal situations

If any of the validation flows does not hold true, appropriate error message will be prompted to the user for doing the needful.

3.3 Performance Requirements

1. Should run on 500 Kbps internet connection and a 512 MB RAM Machine
2. Responses should be within two seconds

3.4 Design Constraints: None

3.5 Software System Attributes

a. Usability

The application will be user friendly and easy to operate and the function will be easily understandable

b. Reliability

The application will be available to the candidates through the registration period and have a high degree of fault tolerance

c. Security

The application will be password protected. Users will have to enter correct login ID and password to access the application

d. Maintainability

The application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual module

e. Portability

The application will be easily portable on any windows based system that has SQL Server installed.

3.6 Logical Database Requirements

The following information will be placed in a database:

Table name	Description
Login	Records the login details of the user
Student	Records the details of the students.
Faculty	Records the details of the faculty.
Department	Records the details of the departments..
Course	Records the details of the courses.
Scheme	Records the details of the scheme.
Courses Selected	Records the details of the courses opted by the student.

3.7 Other Requirements : None