



CSD 326 Software Engineering

# SMARTABLE

## FINAL PROJECT REPORT

Karan Singh Samyal  
2010110331

Shashwat Prakash  
2010111108

Vedanta Patil  
2010110710

# INDEX

TOPIC	PAGE NUMBER
➤ Acknowledgement	3
➤ Problem Statement	4
➤ Software Requirement Specifications (SRS)	5
➤ Data Flow Diagram	9
➤ Use Case Diagram	11
➤ Sequence Diagram	12
➤ State Diagram	13
➤ Division of labor	14
➤ Software UI	15
➤ Design Challenges	18
➤ Testing	18
➤ Conclusion	19

# Acknowledgment

We'd like to take this opportunity to extend our heartfelt gratitude to the university and Professor Pooja Malik for the wonderful opportunity to design and deploy our own software from scratch.

We truly enjoyed the course and what it taught us, and we hope to see this project in action by early next year.

We would like to thank ma'am for her ceaseless support and guidance throughout the semester and for helping us with this project.

# Problem Statement

## 1. Project Background and Description

Timetabling is a difficult problem, which is an element belonging to the field of Scheduling. Scheduling is a NP hard problem as it is defined as an allocation of resources over time. The classes and lecture timetable problems are usually solved by hand and takes several days or weeks to repair based on feedback, clashes, credit requirements and personal interests etc. The main problem is the variety of classes to choose from that don't clash between subjects and lecture timings. In most cases students aren't aware of the interesting course they can balance in their schedule while completing the credit requirements. Due to the lack of information and analytical tools students don't tend to apply for courses that would be a better fit for them and drop the courses they don't like. This creates a problem of blocked seats and unhappy response from the student's side.

## 2. Our solution

Our website addresses the above problem by developing a flexible timetable maker that is personally accustomed to each student according to their year, core, batch, personal preferences like minor etc. We developed an open-ended program for all where the courses could be arranged around set time slots and satisfy constraints that are individual to each student. The timetable maker helps save time and energy of students and faculty by suggesting them with a schedule.

## 3. High-level requirements

- Create user accounts and log in.
- Update personal and contact information.
- View the dashboard that has been set visible for the day.
- Update Timetable on the website.
- View available courses.

# Software Requirement Specifications (SRS)

## 1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definition, Acronyms and abbreviations
- 1.4 References
- 1.5 Overview

## 2. The Overall Description

- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Characteristics
- 2.4 Constraints
- 2.5 Operations
- 2.6 System Requirements

## 3. External Interface Requirements

- 3.1 Frontend Description
- 3.2 Backend Description

## 4. System Features

- 4.1 Precise Live updates
- 4.2 Admin and user validation

## 1. Introduction

### 1.1 Purpose

1.1.1. The purpose of this SRS is to describe the requirements involved in developing a Timetable developing system.

1.1.2. The intended audience is any student, who wants to create their timetable with ease.

### 1.2 Scope

1.2.1. The product is titled as SmarTable.

1.2.2. The product performs the following tasks:

1.2.2.1. Generates a timetable for students.

1.2.2.2. Gives suggestions on which courses can be added.

1.2.2.3. Edit timetable with ease.

1.2.2.4. Create draft timetables.

### 1.3 Definition, Acronyms and abbreviations

1.3.1. DBMS – Database Management System

1.3.2. CCCs – Core Curriculum Courses

1.3.3. UWEs – University Wide Electives

1.3.4. SNU - Shiv Nadar University

1.3.5. Web Dev - Website development

### 1.4 References

1.4.1. IEEE standard 830-1998 recommended practice for Software Requirements Specifications-Description

### 1.5 Overview

1.5.1. The SRS contains an analysis of the requirements necessary to help ease design.

1.5.2. The overall description provides an interface requirement for the Library Management System, product perspective, software interfaces, communication interfaces, memory constraints, product functions, user characteristics and other constraints.

1.5.3. Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect Library Management System in any manner.

## 2. The Overall Description

### 2.1 Product Perspective

#### 2.1.1. Software interfaces

2.1.1.1. Backend: MySQL, Flask

2.1.1.2. Frontend: Javascript, Python, HTML, CSS

#### 2.1.2. Memory Constraints

2.1.2.1. No specific constraints on memory

2.1.2.2. The database can change with respect to the courses of the university.

### 2.2 Product Functions

2.2.1. Enquire about the availability and status of the courses.

2.2.2. Search the availability of the seat for each course before adding it for the user.

2.2.3. Search the compatibility of courses according to the timing before showing available ones.

2.2.4. The website validates the timetable by checking the timing overlaps etc.

2.2.5. Using user input (preferences) it shows a combination of core, uwes, major elective and CCC subjects in the timetable.

### 2.3 User Characteristics

2.3.1. The users need not have specific knowledge as to what is the internal operation of the system. Thus, the consumer is at a high level of abstraction that allows easier, faster operations and reduces the knowledge requirement of the end user.

2.3.2. The product is absolutely user friendly, so the intended users can be the naïve users.

2.3.3. The product does not expect the user to possess any technical background.

### 2.4 Constraints

2.4.1. The user has a unique username and password, there are no options to retrieve a password or username in case it is forgotten or lost hence the user is required to remember or store the username and password.

### 2.5 Operations

2.5.1. The software allows three modes of operations, enquiring about the availability and status of the courses.

2.5.2. By extracting the username and password the software allows the user to borrow a maximum of three books.

2.5.3. By extracting the username, the password, the software allows the user to add, edit timetable to their preferences.

### 2.6 System Requirements

#### 2.6.1. Logical Database requirements

2.6.1.1. The system contains databases that include all necessary information for the student to be able to make an accurate timetable according to the inputted requirements. These include relations such as user details and course details.

2.6.1.2. The user details refer to the information such as name, department, year, the title and the already done courses by the student.

2.6.1.3. The course details refer to the information such as the title of the course, availability status, minor criterion, timing, days and the number of available seats.

### 3. External Interface Requirements

#### 3.1 Frontend Description

3.1.1. SmarTable is a flexible timetable maker that is personally accustomed to each student according to their year, core subject, batch, personal preferences like minor, and more. We developed an open-ended program for all where the courses can be arranged around set time slots and satisfy constraints that are individual to each user. The timetable maker will help save time and energy for students and faculty by suggesting a schedule perfect for their requirements and interests. The website uses predictive analysis based on already existing and inputted data of courses to help suggest appropriate courses like CCCs, UWEs, electives, and others. The easy-to-access portal will showcase different subjects they can apply for based on department preference and access.

#### 3.2 Backend Description

3.2.1. The web application consists of two databases. One contains the student details such as name, hashed passwords, roll number etc. The other would contain the data of all the courses with their respective class timings and other information such as course code, course timings, course name etc.

### 4. System Features

#### 4.1 Precise Live updates

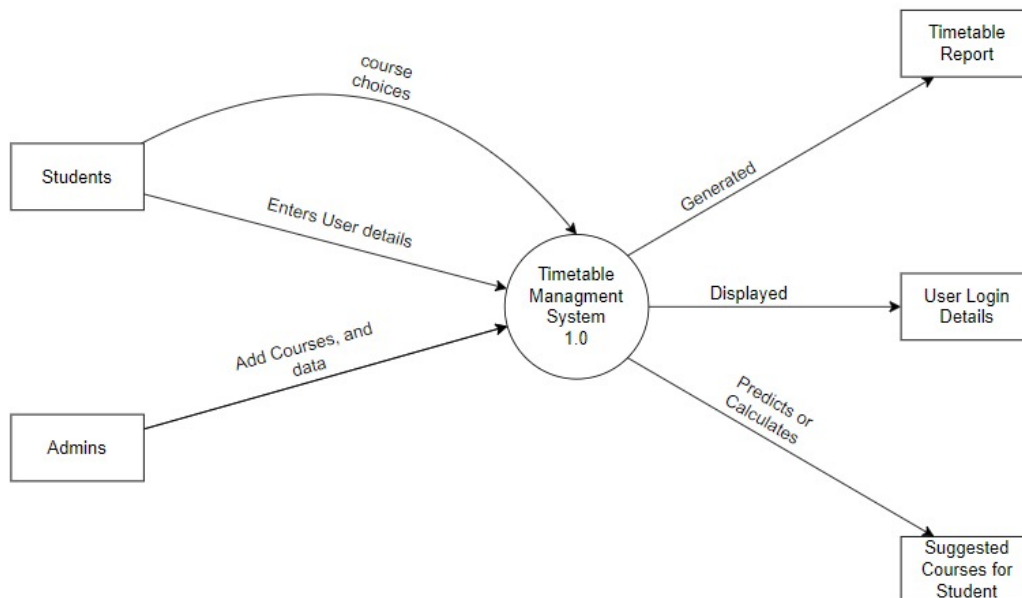
4.1.1. Live updates of the timetable from user input preferences.

#### 4.2 Admin and user validation

4.2.1. Secure and strict admin and user validation. Passwords are encrypted.

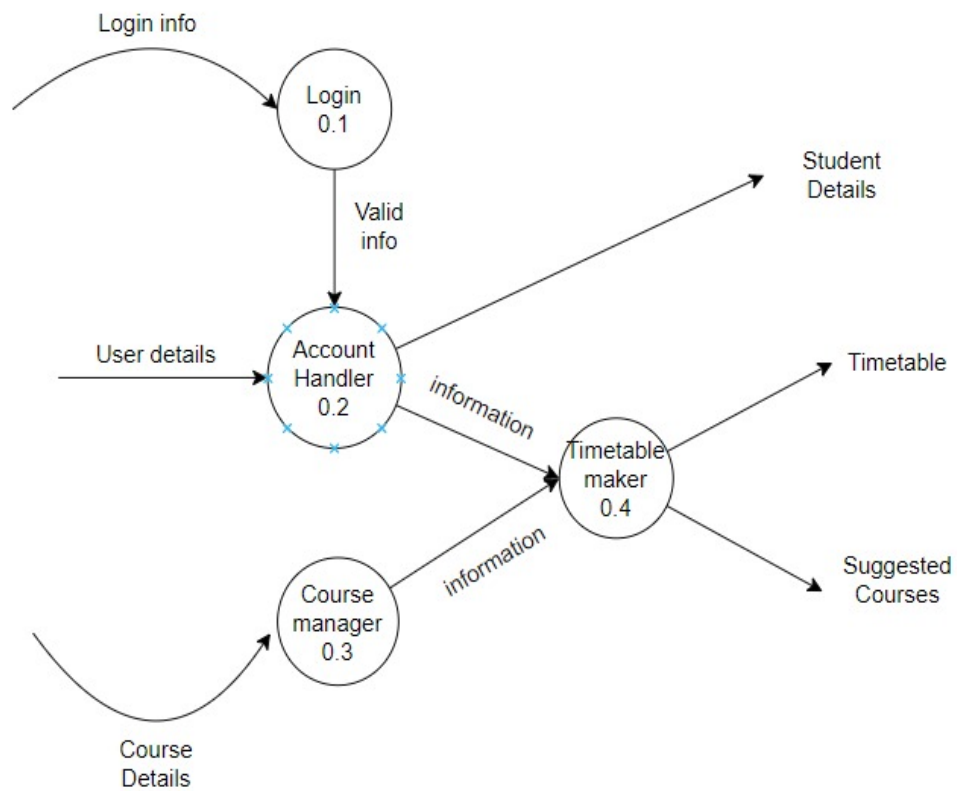


# Data Flow Diagram



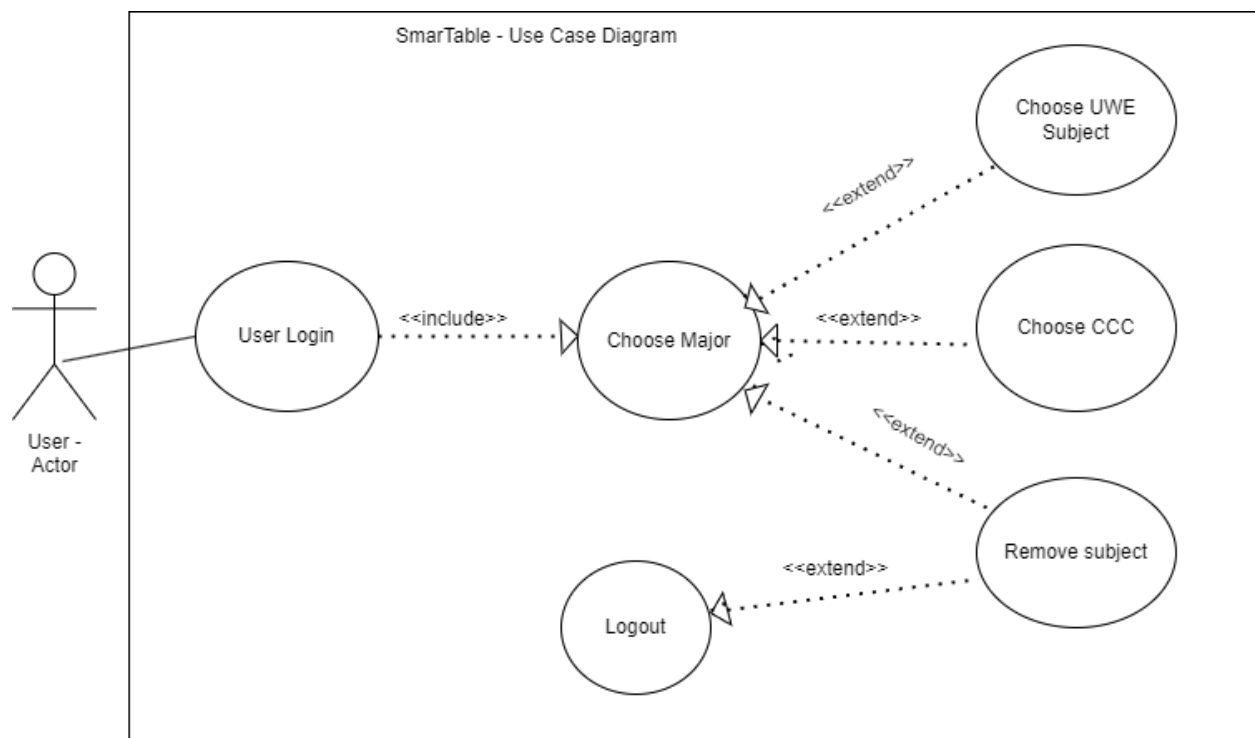
## Context Diagram

# Data Flow Diagram

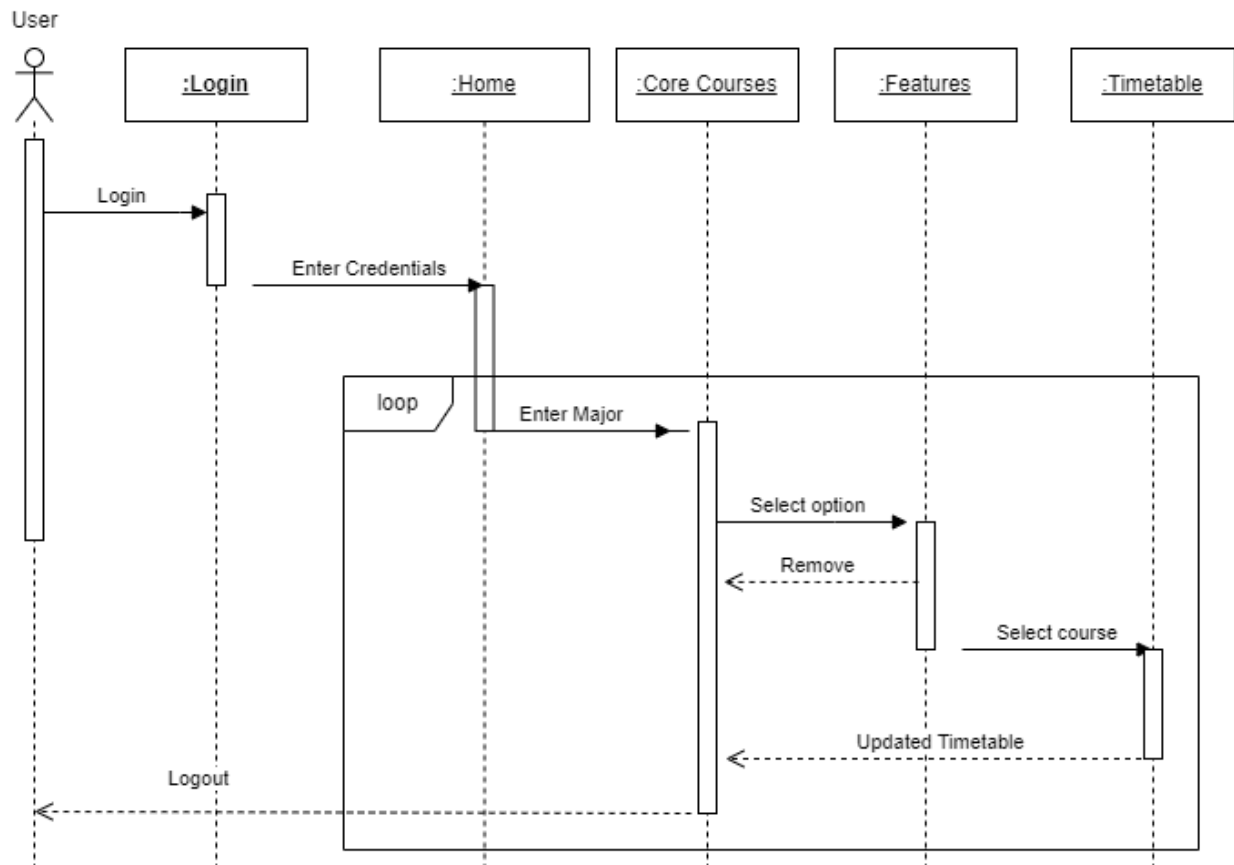


## DFD Level 1

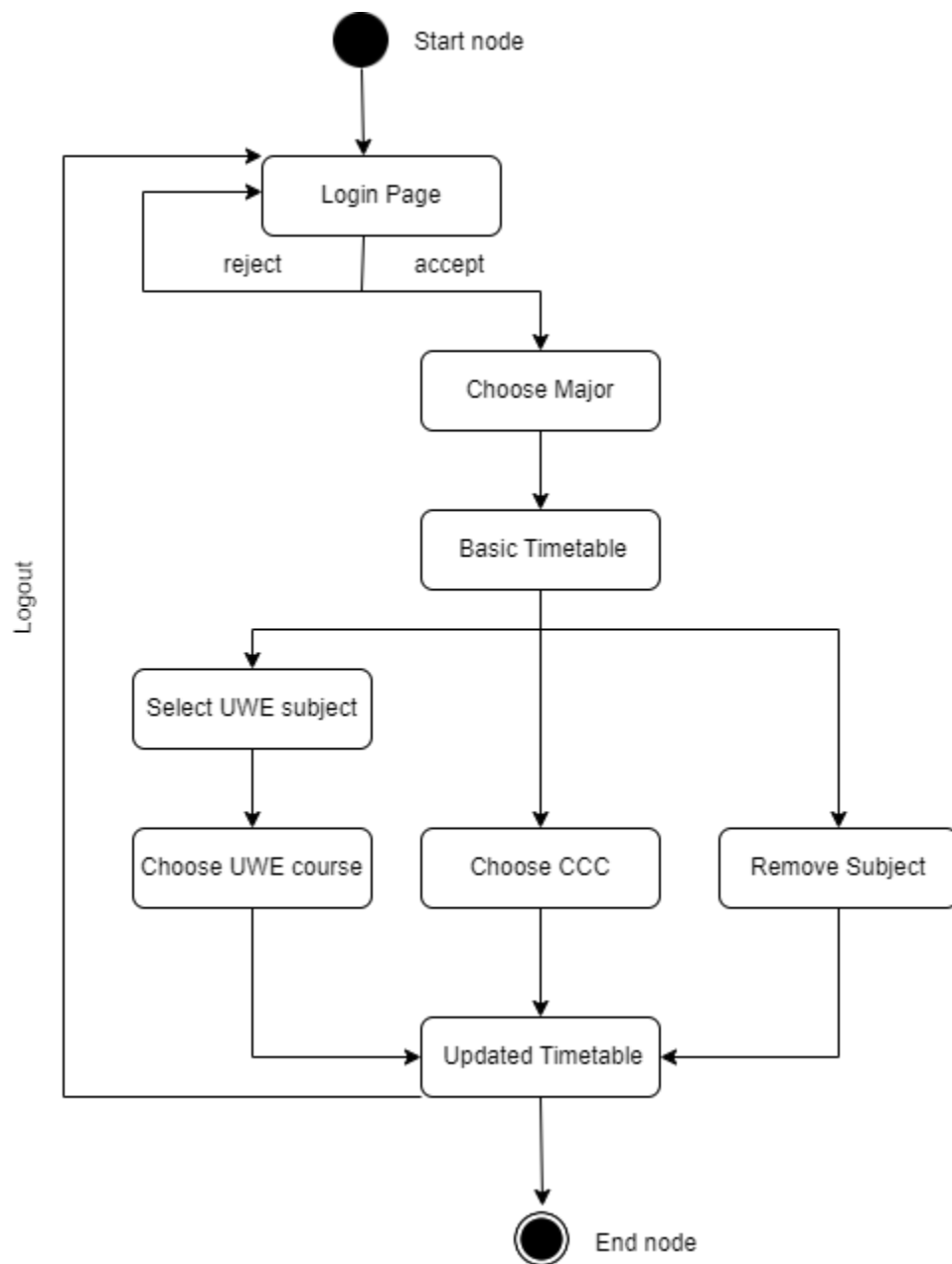
# Use-Case Diagram



# Sequence Diagram



# State Diagram

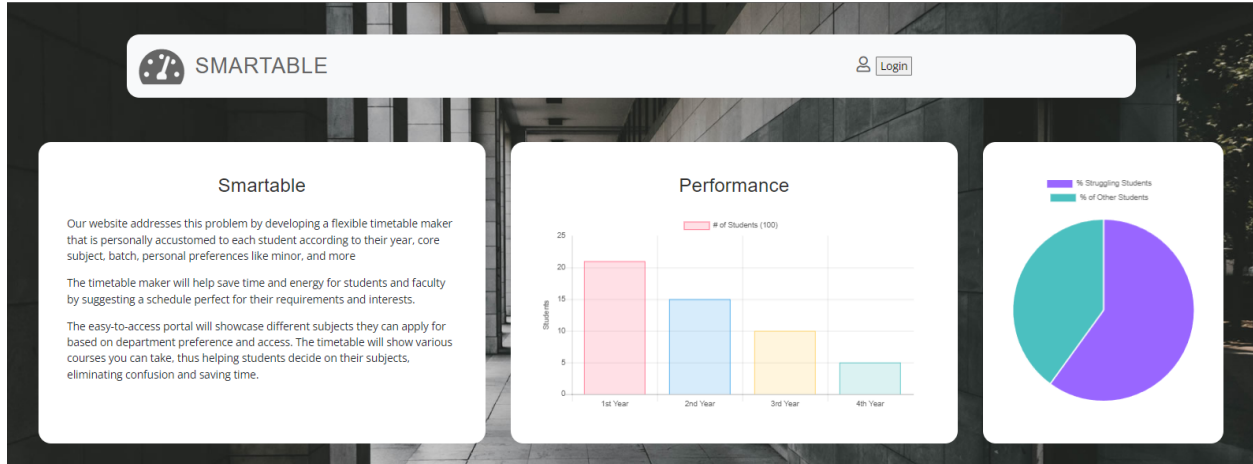


## Division of Labour

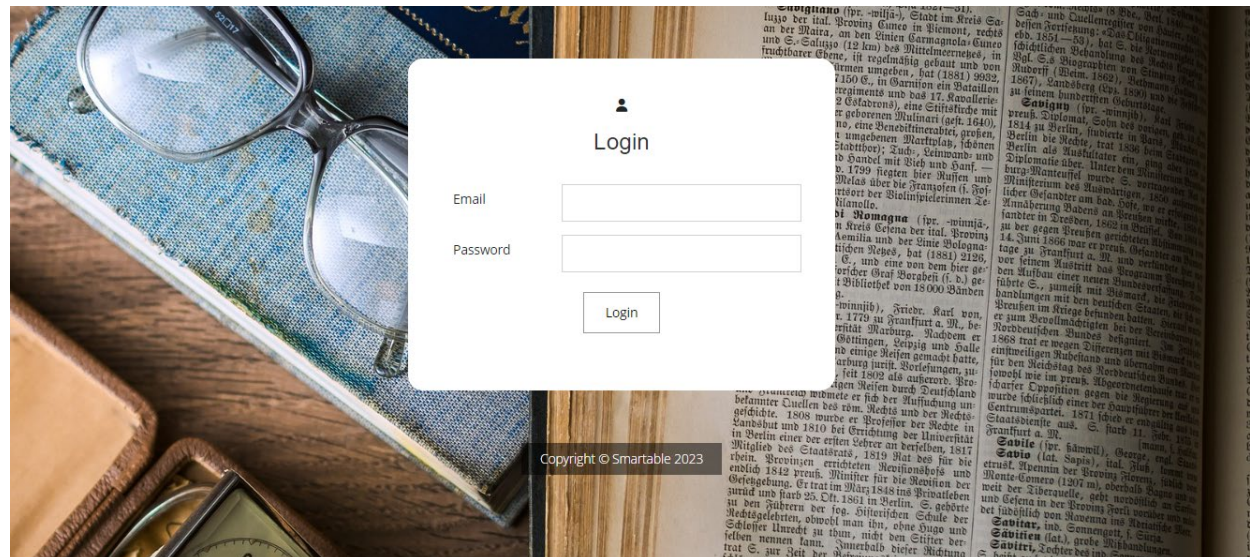
➤ Problem Statement	ALL
➤ Software Requirement Specification	Karan
➤ Data Flow Diagram	Shashwat
➤ Use case Diagram	Vedanta
➤ Sequence Diagram	Vedanta
➤ Class Diagram	Vedanta
➤ Activity Diagram	Vedanta
➤ User Side Website	Shashwat
➤ Backend Set-up	Karan
➤ Backend Integration	Vedanta
➤ Admin Side Website	Shashwat
➤ Database Entries	Karan
➤ Report	ALL

# Software UI


## Home Page



## Login Page



# Admin Page

 SMARTABLE

Logout

### Course Information

	Course Name	Course Code	Component	Major	R
0	Heat Transfer	CHD226	LEC1	CHD2YR	N
1	Mechanical Operations	CHD227	LEC1	CHD2YR	N
2	Industrial and Engg. Chemistry	CHD223	LEC1	CHD2YR	N
3	Chemical Reaction Engg. I	CHD225	LEC1	CHD2YR	N
4	Numerical Methods	CHD262	LEC1	CHD2YR	N

### Admin Details


Employee Name: Vedanta Patil

Account Email: vp433@snu.edu.in

Phone Number: 9876543210

Admin ID: 2010110710


### Profile Image



Upload New...

Copyright © Smartable 2023

# Student Page

 SMARTABLE

Logout

### Information

- DOM 204 is now open!
- Someone dropped CSD454!
- Seats in FAC401 filled!
- CCC343 is now open!
- New course EED321 is now being offered!

### Student Details

Student Name: Shashwat Prakash


Account Email: sp155@snu.edu.in

Phone Number: 6354350315

Roll Number: 2010111108

Timetable

### Profile Image



Upload New...

Copyright © Smartable 2023



# Timetable UI

Logout

Choose Course ▾

UWE Subject ▾

UWEs Available ▾

CCCs available ▾

Remove Subject ▾

Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
08:00:00	----	----	----	----	----	----	----
08:30:00	----	----	----	----	----	----	----
09:00:00	CHD328	ENG346	CHD328	ENG346	CHD328	----	----
09:30:00	CHD328	ENG346	CHD328	ENG346	CHD328	----	----
10:00:00	----	ENG346	----	ENG346	----	----	----
10:30:00	----	CHD327	----	CHD327	----	----	----
11:00:00	CHD413	CHD327	CHD413	CHD327	CHD413	----	----
11:30:00	CHD413	CHD327	CHD413	CHD327	CHD413	----	----
12:00:00	----	CHD415	----	CHD415	----	----	----
12:30:00	----	CHD415	----	CHD415	----	----	----
13:00:00	----	----	----	----	----	----	----
13:30:00	----	----	----	----	----	----	----

List of subjects

Mass Transfer-II

Transport Phenomenon

Chemical Technology

Process Equipment Design

Process Equipment Design

Intro. to Postcoloniality

Add New Category

## Design Challenges

This project has provided us with several valuable lessons. The challenges we encountered along the way were significant. Our team could have managed our time better by planning the project in a way that allowed us to work consistently throughout the semester, rather than cramming everything in at the end.

All of us were new to web application development, so learning the tech stack for the same proved to be quite the challenge.

Collaborating with team members proved to be challenging due to differing priorities, schedules, and commitments. Despite this, we were able to work together and build the project.

Initially, we faced issues with code collaboration due to not using Github. However, we were able to learn from this experience and gained valuable insights for future projects.

## Future Path

We plan to develop this into a full-fledged working project that can actually be deployed for SNIOE students.

There are a lot more features that could be incorporated, such as having multiple timetables that can be seen at the same time on the website itself.

Features that can be added include:

- Saving multiple timetables and being able to view each of them side by side.
- Being able to dynamically change the database.
- Better implementation of the UI.
- Make it usable for multiple devices.

## Conclusion

We feel that even though we have worked extremely hard on this project, we still need to work on it before it can actually be deployed. However, we have gained a lot of knowledge in the process and are almost certain and confident that we will be able to do so by early next year. We learnt valuable lessons about working as a team and managing time. We also learnt about various new software and technologies and we are truly grateful for the same.