

# eduNEP

## UCS 503 Software Engineering

Submitted by:

101803186	–	Kulpreet Singh
101803187	–	Sajjal Tiwari
101803189	–	Arshnoor Batra
101853038	–	Vishrut Agrawal

B.E Third Year, Computer Engineering

Submitted to:

Ms. Deepali Bhagat



**THAPAR INSTITUTE**  
OF ENGINEERING & TECHNOLOGY  
(Deemed to be University)

Thapar Institute of Engineering and Technology

November 2020

UCS 503

Software Engineering, BE 3<sup>rd</sup> Year

# TABLE OF CONTENTS

---

## Contents

1. Project overview
2. Software Requirement Specification (SRS)
3. Structured Analysis
  - 3.1 Data Flow Diagrams
    - 3.1.1 DFD Level 0
    - 3.1.2 DFD Level 1
    - 3.1.3 DFD Level 2
4. Object Oriented Analysis
  - 4.1 Use Case Diagram
  - 4.2 Activity Diagram
  - 4.3 Class diagram showcasing Logical structure
  - 4.4 Collaboration Diagram
  - 4.5 Sequence Diagram
  - 4.6 State Chart Diagram
  - 4.7 Component Diagram
  - 4.8 Deployment Diagram
5. Testing
  - 5.1 Test Cases for various test strategies (Scenario, Boundary value analysis and Equivalence class testing with ScreenShots)

# 1. Project Overview

We aim to build a mobile application named **eduNEP** (a flutter and django based app) which will facilitate the smooth implementation of the new **National Education Policy (NEP)** in all schools across the country. This app will be a unified platform for school management, students and **The Central Board Of Secondary Education (CBSE)** providing its users with all the necessary details like records, mark sheets, student progress report, etc. The idea is feasible in terms of implementation, and flutter framework helps in materialization enabling its use in iOS as well as Android.

eduNEP will integrate the benefits of a physical classroom and provide the students with an online platform to store their progress as well as other important information. It will also provide all the student, parent, mentor data to be collected by the school admin which will make it a completely secure network. It will also traverse the data to the **The Central Board Of Secondary Education (CBSE)** which will help in generation of mark sheets.

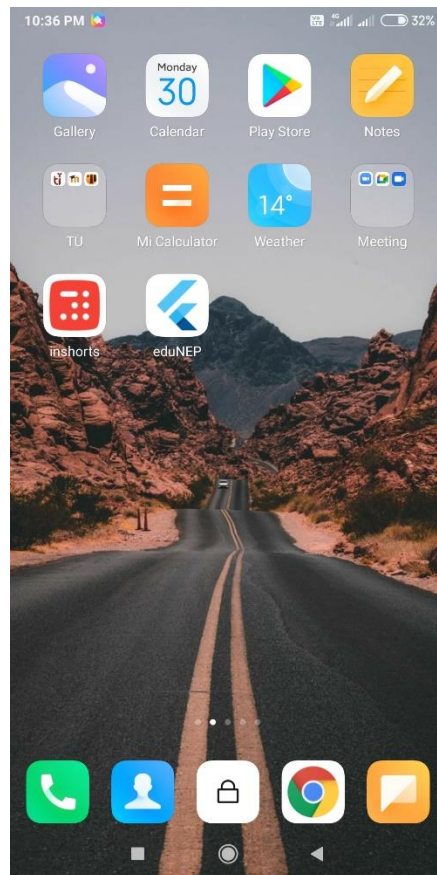


Figure 1 App logo

---

# Software Requirements Specification for eduNEP

Version 1.0

Prepared by

Kulpreet Singh  
Sajjal Tiwari  
Arshnoor Batra  
Vishrut Agrawal

**Group Name: #2** Advanced Idea Mechanics (AIM)  
101803186      ksingh\_be18@thapar.edu  
101803187      ssajjal\_be18@thapar.edu  
101803189      abatra\_be18@thapar.edu  
101853038      vagrawal\_be18@thapar.edu

**Instructor:** Dr. Seema Bawa

**Course:** UCS503 Software Engineering

**Lab Section:** 3-COE-9

**Teaching Assistant:** Ms. Deepali

**Date:** 22-09-2020

# Contents

<b>CONTENTS</b>	<b>4REVISIONS</b>
<b>61</b>	<b>INTRODUCTION</b>
71.1	DOCUMENT PURPOSE
71.2	PRODUCT SCOPE
71.3	INTENDED AUDIENCE AND DOCUMENT OVERVIEW
71.4	DOCUMENT CONVENTIONS
2	
1.5 REFERENCES AND ACKNOWLEDGMENTS	<b>82</b>
<b>OVERALL DESCRIPTION</b>	<b>3</b>
2.1 PRODUCT OVERVIEW	3
2.2 PRODUCT FUNCTIONALITY	4
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS	4
2.4 ASSUMPTIONS AND DEPENDENCIES	4
<b>3 SPECIFIC REQUIREMENTS</b>	<b>5</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS	5
3.2 FUNCTIONAL REQUIREMENTS	6
3.3 USE CASE MODEL	7
<b>4 OTHER NON-FUNCTIONAL REQUIREMENTS</b>	<b>13</b>
4.1 PERFORMANCE REQUIREMENTS	13
4.2 SAFETY AND SECURITY REQUIREMENTS	13
4.3 SOFTWARE QUALITY ATTRIBUTES	13
<b>5 OTHER REQUIREMENTS</b>	<b>14</b>
<b>APPENDIX A – DATA DICTIONARY</b>	<b>15</b>
<b>APPENDIX B - GROUP LOG</b>	<b>16</b>

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Team AIM	Initial Version - Development Phase	22/09/20

# 1 Introduction

We aim to build a mobile application named **eduNEP** (a flutter and django based app) which will facilitate the smooth implementation of the new **National Education Policy (NEP)** in all schools across the country. This app will be a unified platform for school management, students and **The Central Board Of Secondary Education (CBSE)** providing its users with all the necessary details like records, mark sheets, student progress report, etc. The idea is feasible in terms of implementation, and flutter framework helps in materialization enabling its use in iOS as well as Android.

## 1.1 Document Purpose

With the former education system being replaced with a **5+3+3+4 model**, there would be a huge requirement of paperwork to cater to the needs of every student. As the new policy offers a large number of subjects, projects and grouping criteria for all the choices, it is going to be difficult to keep in mind the need of every student without a proper framework.

Our application serves the purpose here. It provides a single destination for the students to fill in their choices and to get the subject of their choice. eduNEP will also store the data regarding the ongoing projects of the student with the allocation of a mentor. It will also serve as a platform for the upgradation of marks by the teachers and will also provide a space to get remarks from the teachers. Furthermore, Letters Of Recommendation (LORs) can be a huge motivation for students to perform well in their projects and be consistent with them, which will be guaranteed in eduNEP.

## 1.2 Product Scope

eduNEP will integrate the benefits of a physical classroom and provide the students with an online platform to store their progress as well as other important information. It will also provide all the student, parent, mentor data to be collected by the school admin which will make it a completely secure network. It will also traverse the data to the **The Central Board Of Secondary Education (CBSE)** which will help in generation of mark sheets.

## 1.3 Intended Audience and Document Overview

eduNEP will cater to the needs of the new National Education Policy (NEP) in an efficient manner. With a user friendly interface, it is projected to deliver service in an ordered fashion. This document contains the detailed description of the data flow and use case diagrams, explaining every feature eduNEP has to offer. It will explain in detail the basic functionality, implementation constraints, functional requirements and hardware and the software interfaces.

## 1.4 Document Conventions

The document has used bold lettered words to highlight key words. The document has tried to maintain a priority of requirements. The priority has been determined by the judgement of the author and may be subject to change. The document has used short forms for some commonly abbreviated terms.

## 1.5 References and Acknowledgments

### Courses :

- Udemy Course to learn Flutter & Dart to Build iOS and Android Apps
- Youtube Tutorials for Django

### Websites :

- [https://www.mhrd.gov.in/sites/upload\\_files/mhrd/files/NEP\\_Final\\_English\\_0.pdf](https://www.mhrd.gov.in/sites/upload_files/mhrd/files/NEP_Final_English_0.pdf)
- <https://www.jurist.org/commentary/2020/08/naina-mathew-india-new-education-policy-2020/>
- <https://www.mhrd.gov.in/nep-new>

### Acknowledgements :

We would like to express our sincere gratitude to our Software Engineering Mentors **Dr. Seema Bawa** and **Ms. Deepali**, for their vital support, guidance and encouragement; which has helped us patch up this project and make it successful. We would also like to express our gratitude to the staff of the CSED at TIET, Patiala for their support during the making of this project.

We take this opportunity to thank our parents, who gave us valuable advice and support, without which this project would not have come forth.

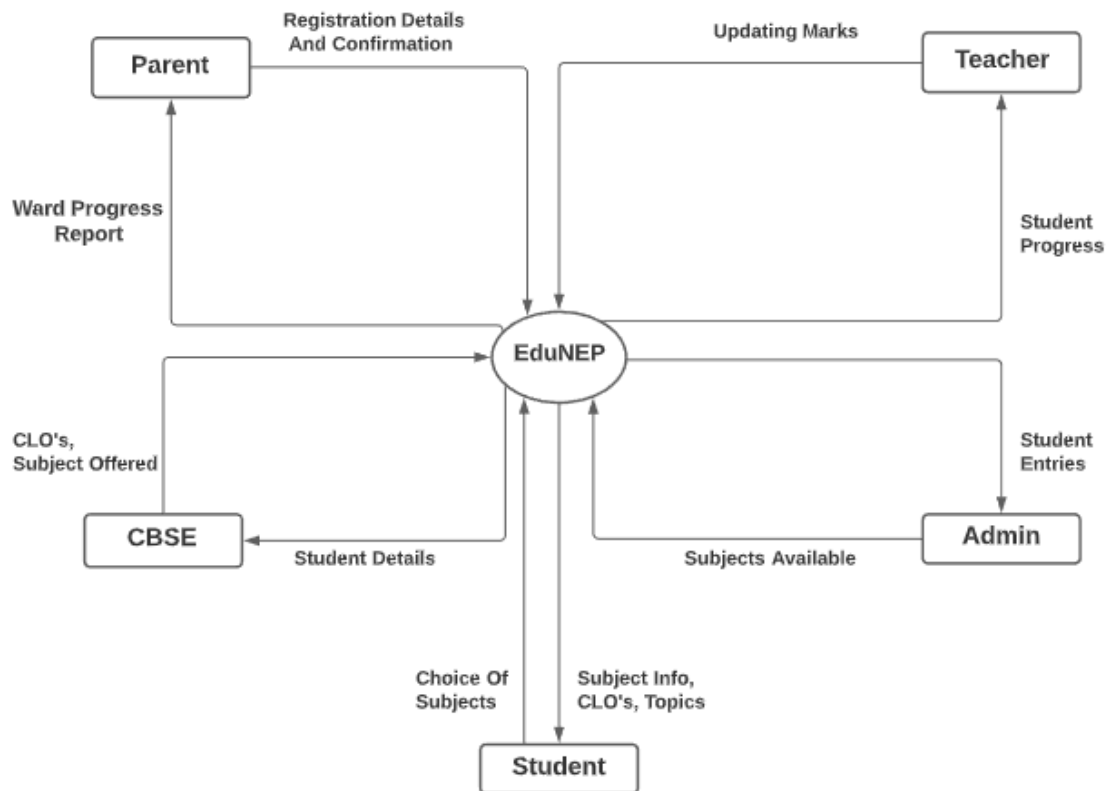
Lastly, we would also like to thank every person who has been directly or indirectly involved with this project, and whose suggestions have been important for its completion.



## 2 Overall Description

### 2.1 Product Overview

**eduNEP** is a Flutter and Django based app to facilitate the smooth implementation of the **new National Education Policy (NEP)** in all schools across the country. This app will be a unified platform for school management and marksheet/resume building. Involved parties include Govt authority of NEP, School admin, Teachers, Parents, and Students. The idea is feasible in terms of implementation, and flutter framework helps in materialization enabling its use for iOS as well as Android.



## 2.2 Product Functionality

1. Account registration for Students, Parents and Teachers
2. Subject Choice as per New National Education Plan
3. Progress tracking based on marks and syllabus covered
4. Individual Project and Mentor selection
5. Letter of Recommendation and Remarks from the teacher based on project performance.
6. Yearly Marksheet from CBSE implementing the NEP credits storage policy
7. Auto-Build Resume based on the progress of the student, including marks % and subjects and projects of the students.

## 2.3 Design and Implementation Constraints

Since we have to submit this project within a deadline of two months or so, i.e. before the end of this semester, so we are going to use **Django** (a python based backend framework) which is really helpful for object oriented backend development and has **SQLite** as a default database management system and handles the security related part of the app really well. Django is highly dependent on the changes in the environment i.e. any update in the version of python or related parts like SQLite can affect the working of Django, hence we need to work in a **virtual environment** for our backend. Also, we are using **Flutter** for the frontend part, which is a framework for app development which works on both **Android** and **IOS**. So, we have to keep in mind that **our versions of different components** are fully compatible with each other. As of now we are implementing only the App. So, the language requirements for our App will be **Python** and **Dart**.

## 2.4 Assumptions and Dependencies

- Django (as a backend) and Flutter (as a frontend) will not cause any issues while integration of the two.
- We would have a similar environment while hosting the app like we have locally in our virtual environment.
- We would not need to change any version of any framework during the entire course of project development.
- Django will handle the security part of the app implicitly and we do not need to take any explicit measures for handling security of our app.
- The default database of Django i.e. SQLite will work fine even with the large amount of data our project is going to handle.
- This project is built totally from the scratch and is not reusing any components from any other project.
- Teacher will be registered by the admin.
- Every user knows English to communicate with the App.

## 3 Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

##### App Screens :

- Student / Admin Login
- Student / Admin Registration
- Student Profile Details
- Admin Subjects Offered / Availability
- Student Subject Choice Filling
- Student Dashboard
- Student Subject Description
- Admin Upload Progress
- Admin Upload Marks
- Student View Marks / Progress
- Admin Compile Marks / Build Marksheet
- Student View Marksheet
- Student Projects
- Admin Project Mentors / Remarks
- Student / Admin Build (View) Resume

##### User Device Requirements :

- Android v8.0 and higher
- iOS v11.0 and higher
- RAM 3GB and higher

#### 3.1.2 Hardware Interfaces

- Processor : Intel i7 9th gen 9750H CPU @ 2.60 GHz (6 cores, 12 threads, L3 cache 12 MB)
- RAM : 8 GB DDR4 @ 2667 MHz
- Storage : 256 GB SSD NVME + 1 TB HDD SATA
- Graphics : NVIDIA GeForce GTX 1650 - 4 GB GDDR5
- OS : Microsoft Windows 10 Home Version 10.0.18363 Build 18363

#### 3.1.3 Software Interfaces

- Front End : Flutter Framework with Dart programming language
- Back End : Django

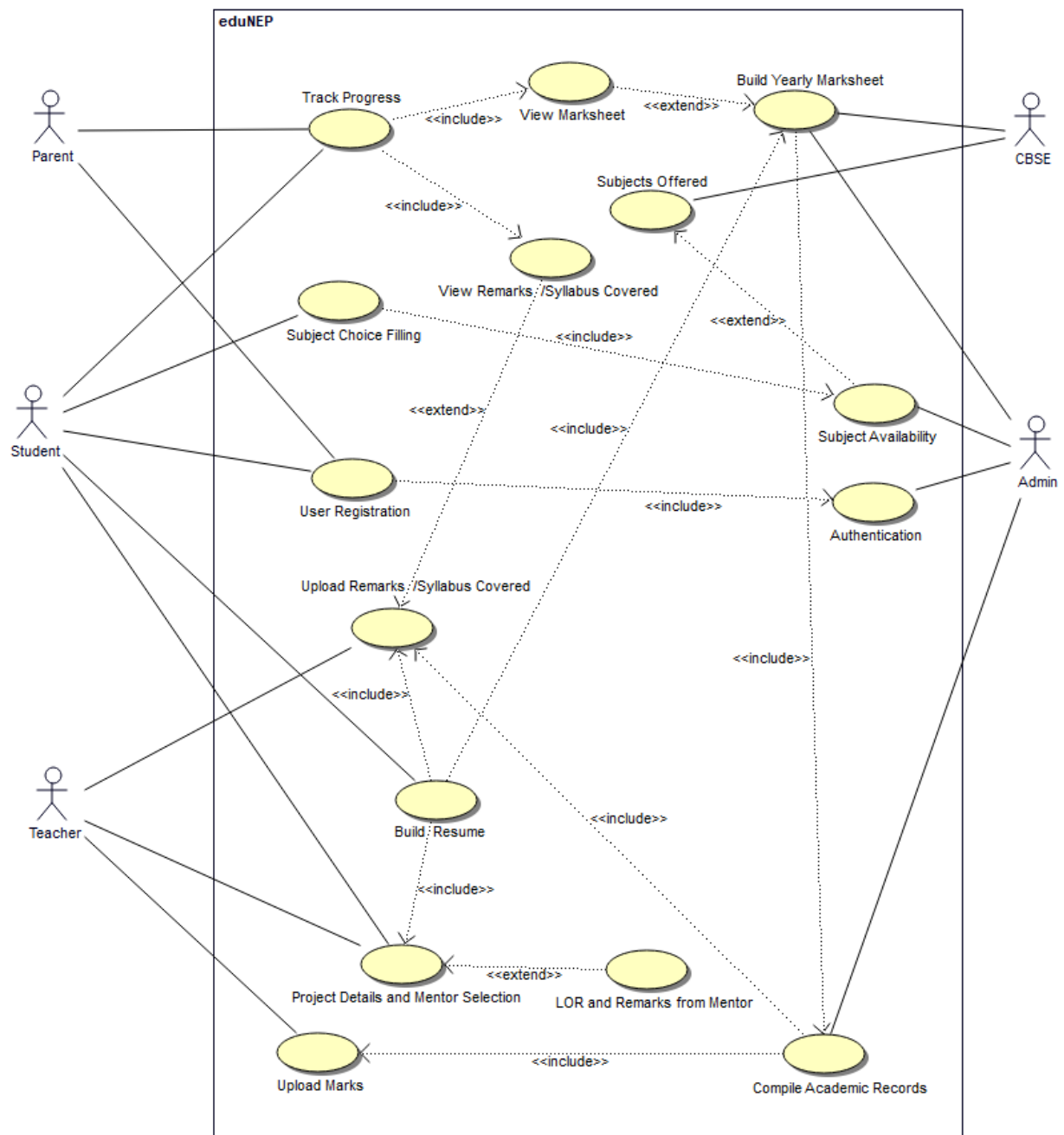
### 3.2 Functional Requirements

The functional requirements of this project are listed below:

- 3.2.1 F1: The user will provide their userID and password to login. Also, giving an option to register a new user as well.
- 3.2.2 F2: The CBSE will create a list of subjects offered in the database which will be passed on to Admin's subject choice availability.

- 3.2.3 F3: The Admin will filter out subjects offered on basis of faculty and resources available at the school, to create a list of subject choices available in the database which will be passed on to Student's subject choice filling.
- 3.2.4 F4: The Student will filter out the subjects available on basis of their interest in the subjects, to create a list of subject choices filled in the database which will be passed on to other use cases.
- 3.2.5 F5: The Teacher will create an event (test, assignment, project, etc.) and set its weightage in total marks of the subject, then upload marks of all students in their class.
- 3.2.6 F6: The Teacher will maintain a check list showing the total syllabus covered of the subject at that point, and also give remarks to all students in their class.
- 3.2.7 F7: The Admin will access all uploaded marks of all subjects for all students, along with the weightage assigned to them; Then Admin will compile these and store in a database which will be passed on to CBSE's Build Marksheet.
- 3.2.8 F8: The CBSE will access all compiled marks from the databases of schools, build yearly marksheet for students, then store in a database which will be passed on to View Marksheet.
- 3.2.9 F9: The User will have access to their yearly marksheet, they can only view and download it, but cannot edit it.
- 3.2.10 F10: The User will access the dashboard of the app to see progress and remarks given by teachers in all enrolled subjects.
- 3.2.11 F11: The Student will create a new project and enter details, then access a list of available mentors and send request for it, which will be seen by Teacher and may or may not be accepted. This final data will then be stored on database.
- 3.2.12 F12: The Teacher who has accepted mentor role for a project will have option to give their remarks to Student, maybe a LOR as well based on their performance, which the Student can view and download, but not edit.
- 3.2.13 F13: The User will have access to yearly marksheets, projects and LORs to build a resume of Student following a predefined format, then view and download the final document.

### 3.3 Use Case Model



**Author –** Kulpreet Singh

**Purpose –** This use case holds all the functional requirements of the system and specifies the interaction of each actor with the system.

**Requirements Traceability –**

1. User Login
  - User must be registered on the app
  - User should be authenticated
2. Subjects Offered
  - User must be CBSE and needs to be logged in
  - User must have the option to upload offered subjects
  - Offered Subject need to be stored in the database
3. Subject choice Availability
  - User must be the School admin and needs to be logged in
  - User must have access to the subjects offered by CBSE
  - User must have the data about the faculty available for each subject
  - User needs the functionality to apply a filter on the subjects offered by CBSE based on faculty available.
  - Available choices need to be stored in the database to be shown to the student.
4. Subject choice filling
  - User must be a student and needs to be logged in
  - User must have access to the choice available in the school
  - User must have the option to fill his/her choice
  - Filled choice need to be stored in the database
  - Admin must have the access to the choices filled by all students and store them in the CBSE database
5. Upload Marks
  - User must be a teacher and needs to be logged in
  - User should have the option to upload the marks of the students in his class in his subject
  - Uploaded marks need to be stored in the database
6. Upload Remarks/ Syllabus Covered
  - User must be a teacher and needs to be logged in
  - User should have the option to upload the Remarks/Syllabus covered of the students in his class in his subject
  - Uploaded Remarks/ Syllabus Covered need to be stored in the database
7. Compile Academic Records
  - User must be the School Admin and needs to be logged in
  - User must have the access to the uploaded marks, remarks and Syllabus covered by each teacher for each student
  - User must have the option to Compile the records as per the weightage associated
  - Compiled records need to be stored in the database and sent to the CBSE database
8. Build Yearly Marksheet
  - User must be CBSE and needs to be logged in
  - User must have the access to Compiled academic records of each school
  - User must have the option to build yearly marksheet for all the students
  - Yearly marksheet need to be stored in the database.

#### 9. View Marksheet

- User must be a Parent or Student and must be logged in.
- User must have the access to the yearly marksheet generated by CBSE associated to him
- User must be able to only view and Download the marksheet and not Edit it.

#### 10. View Remarks/Syllabus Covered

- User must be a Parent or Student and must be logged in
- User must have read access to Uploaded remarks/ Syllabus covered by a teacher
- User must be able to only view this information

#### 11. Project Details and Mentor Selection

- User must be a Student who initiates this Process and need to be logged in
- User must have the option to start a project
- User should be able to enter the details
- User should be presented with a list of mentors available
- User should have the option to choose a mentor and send a request
- Mentor should have the option to accept/reject the request
- All this data needs to be stored in the database

#### 12. LOR and Remarks from the mentor

- User must be a Teacher who has accepted the role of mentor for any student
- User should be able to enter remarks for the project based on the student's performance
- User should have the option to write a Letter of Recommendation for the student he is currently assigned to, If the mentor is impressed by his performance
- Remarks and LOR need to be stored in the database and Student should be able to access it with view permission

#### 13. Build Resume

- User must be a Student and need to be logged in
- User must have the access to Yearly Marksheet from the CBSE, Project Details and Remarks, and the Letter of Recommendation from the mentor, academic remarks from the teachers
- User must have the option to compile these records in a specific format and generate the Resume for that User
- Resume need to be stored in the database and student should be able to View or Download it.

**Priority - HIGH**

**Preconditions** – Teacher needs to be registered by the School Admin.

**Post conditions** – All interactions of each actor with the system will be specified.

**Actors** – Student, Parent, Teacher, Admin, CBSE

**Flow of Events**

1. Basic Flow -

B1: User opens the App in mobile  
Login page Appears

B2: User enters username and password  
System confirms correct login, presents the main dashboard

B3: Student chooses subject Choice filling option  
System shows all the subjects (with their CLOs) available by the school after filtering on the subjects offered by CBSE.

B4: Student confirms his choice of subject  
System stores the choice filled by the student in the database.

B5: Admin accesses choice filled by all the students and sends the data to CBSE  
System accesses the data from the database, compiles it in a desired format and forwards the data to the database of CBSE.

B6: CBSE can access the choice filled by the students of all schools  
System presents the data from the centralised database of CBSE in the desired format.

B7: Teacher uploads Marks  
System stores the marks uploaded by the teacher.

B8: Teacher uploads Remarks / Syllabus Covered  
System stores the Remarks / Syllabus Covered uploaded by the Teacher.

B9: Admin Compiles Academic Records  
System accesses the marks uploaded and the Remarks/Syllabus Covered from all the Teachers and compiles them as desired according to the weightage.

B10: Admin sends the Compiled Academic Records to CBSE  
System accesses the compiled academic records from the school database and sends this data to the database of CBSE.



B11: CBSE builds the Yearly Marksheet  
System accesses the data from the CBSE database and generates a Marksheet in the desired format and stores it.

B12: Student OR Parent tracks progress  
System shows the option to View Marksheet and View Remarks/Syllabus Covered.

B13: Student OR Parent chooses View Marksheet  
System shows the Yearly Marksheet generated by CBSE.

B14: Student OR Parent chooses View Remarks/Syllabus Covered  
System accesses the Remarks/ Syllabus Covered uploaded by the Teachers and Displays them.

B15: Student chooses to work on a project and chooses a mentor  
System stores the details of the project idea and shows the student the list of available mentors, and then stores the choice of mentor by the student.

B16: Teacher confirms his role as a mentor in that project  
System shows Teacher a request by a Student to accept the role of mentor for his Project and stores the mentor's confirmation.

B17: Mentor gives Remarks to the project  
System stores the mentor's remarks in the database.

B18: Student views Remarks from the mentor  
System accesses the remarks from the mentor and displays them.

B19: Student chooses to build resume  
System accesses the Yearly Marksheet from the CBSE, Project Details and Remarks from the mentor, academic remarks from the teachers and compiles them in a specific format and generates the Resume for the student.

## 2. Alternative Flow -

A1: Student OR Parent is a first-time user

A1.1: Student OR Parent Fills the User Registration Form  
System records the details entered by the student

A1.2: Details of the filled form get authenticated by the School Admin  
System Authenticates the Student and generates the username and the password for Student and Parent and sends this to the Student and parent via email and message.

A1.3: Parent confirms the registration details are filled up to best of his knowledge.

System Store the parent's confirmation by the parent and now student can proceed as per the basic flow.

A2: Teacher rejects the request to be a mentor

A2.1: Student chooses a teacher to act as a mentor for his project

System pops up a request to the teacher to accept the request.

A2.2: Teacher rejects the request

System shows the student that the mentor has declined his request and shows the option to select some other mentor from the list of teachers and Now the basic flow continues.

A3: Mentor gives a Letter of Recommendation to the student

A3.1: Mentor chooses to Give a Letter of Recommendation to the student

System shows him the form to Write LOR for the student he was currently assigned as a mentor to.

A3.2: Mentor fills the Write LOR form and uploads it.

System records the LOR form filled by the mentor and notifies the student about it.

A3.3: Student Views the LOR

System accesses the LOR from the database and shows it to the student in the desired format.

A3.4: Student builds the resume

System accesses the Yearly Marksheet from the CBSE, Project Details and Remarks, and the Letter of Recommendation from the mentor, academic remarks from the teachers and compiles them in a specific format and generates the Resume for the student.

3. Exceptions –

E1: Teacher might not be registered by the school admin.

E2: Parent might not have filled the confirmation form which might prevent the student from logging in to the app.

**Notes/Issues** – No issues yet.

## 4 Other Non-functional Requirements

### 4.1 Performance Requirements

- All the user details are stored locally also, so user details , marksheets are always available
- Fetching and storing data from database should be done in less 2 seconds
- All the processing should be done at database(server) side to achieve peak efficiency

### 4.2 Safety and Security Requirements

- Secure access of student records
- Adding and removal of teachers is accessible by school administrator only
- All the data to be sent via HTTP protocol for better encryption
- Backup of School database at regular intervals to avoid any data loss
- User details that are stored locally are to be encrypted to avoid data leak

### 4.3 Software Quality Attributes

- While sending data to database it should also be stored locally first so in case internet service gets disrupted, data can be sent again without any delay
- User device details to be registered to avoid hacking

## 5 Other Requirements

### 5.1 Database Requirements

All the data for student records will be stored in a relational database schema. For the time being, the servers of FireBase will be used.

## Appendix A – Data Dictionary

Page No.	Heading	Description
3	2.1	DFD Level 0
7	3.3	Use Case Diagram

## Appendix B - Group Log

Timeline	Work Done	Members Present	Members Absent
Week 1 (2 hours)	Orientation of the software's to be used i.e. Lucid chart, Bouml	Arshnoor Kulpreet Sajjal Vishrut	-
Week 2 (4 hours)	Software Bid form and deciding the project.	Arshnoor Kulpreet Sajjal Vishrut	-
Week 3 (2 hours)	Data Flow Diagram Level 0	Arshnoor Kulpreet Sajjal Vishrut	-
Week 4 (3 hours)	Data Flow Diagram Level 1	Arshnoor Kulpreet Sajjal Vishrut	-
Week 5 (2 hours)	Data Flow Diagram Level 2	Arshnoor Kulpreet Sajjal Vishrut	-
Week 6 (2 hours)	Use Case Diagram	Arshnoor Kulpreet Sajjal Vishrut	-
Week 7 (5 hours)	SRS	Arshnoor Kulpreet Sajjal Vishrut	-
Week 8 (3 hours)	Activity Diagram	Arshnoor Kulpreet Sajjal Vishrut	-
Week 9 (2 hours)	Class Diagram	Arshnoor Kulpreet	-

		Sajjal Vishrut	
Week 10 (4 hours)	Sequence Diagram and Collaboration Diagram	Arshnoor Kulpreet Sajjal Vishrut	-
Week 11 (2 hours)	State Chart Diagram	Arshnoor Kulpreet Sajjal Vishrut	-
Week 12 (3 hours)	Component Diagram and Deployment Diagram	Arshnoor Kulpreet Sajjal Vishrut	-
Week 13 (12 hours)	Coding	Arshnoor Kulpreet Sajjal Vishrut	-
Week 14 (12 hours)	Coding and Testing	Arshnoor Kulpreet Sajjal Vishrut	-

### 3. Structured Analysis

#### 3.1 Data Flow Diagrams

##### 3.1.1 DFD Level 0

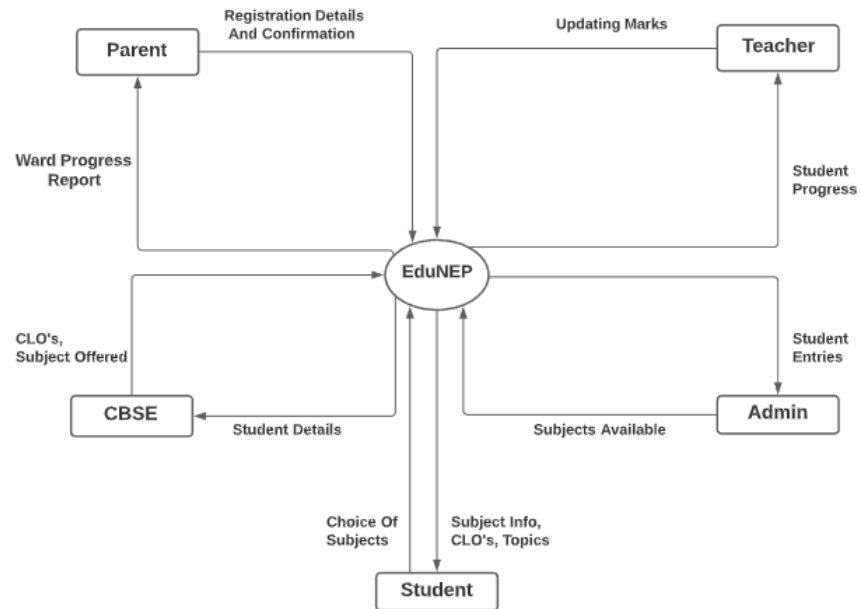


Figure 3.1.1 DFD Level 0



### 3.1.2 DFD Level 1

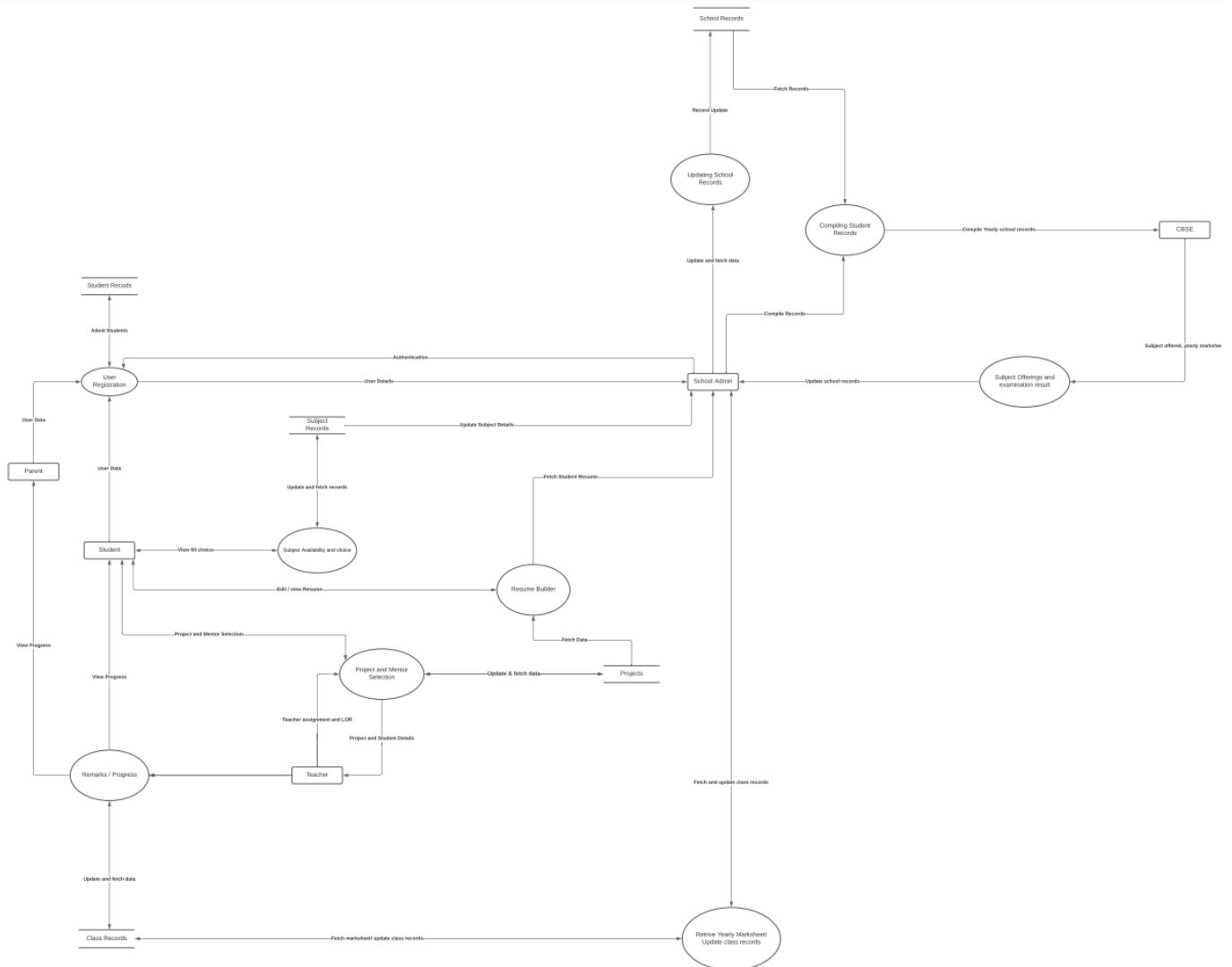


Figure 3.1.2 DFD Level 1

### 3.1.3 DFD Level 2

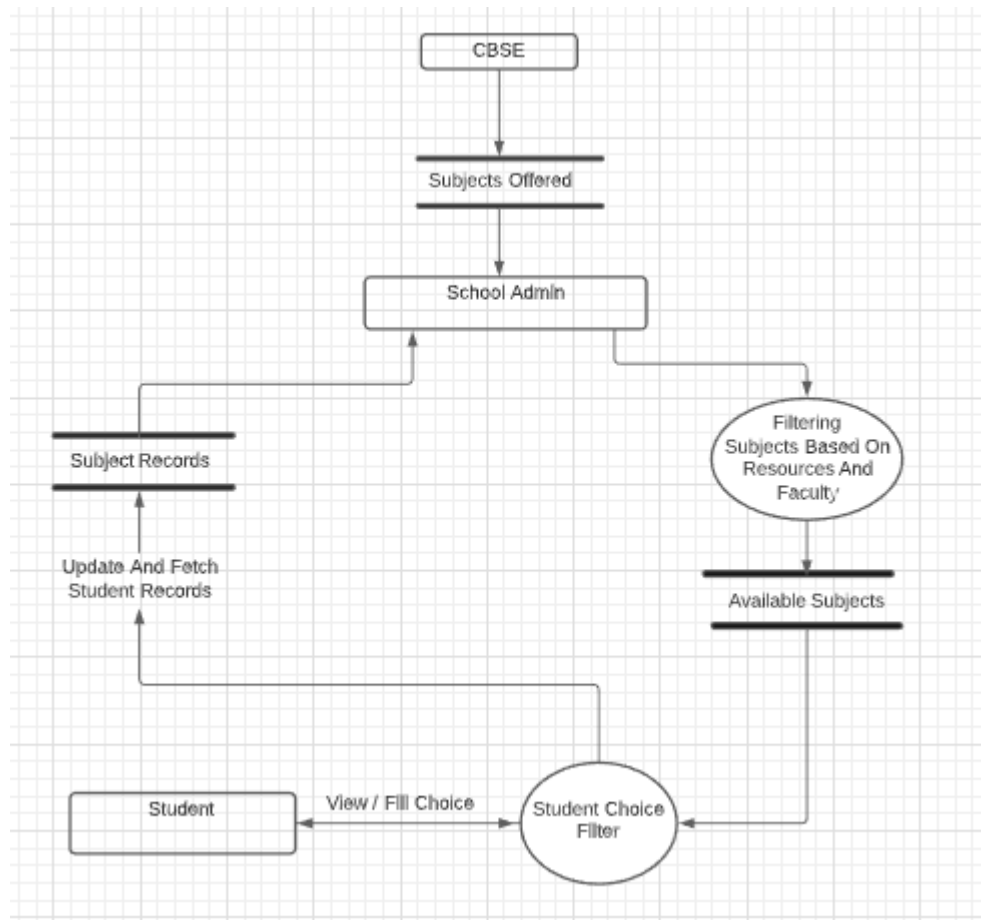


Figure 3.1.3 DFD Level 2

## 4. Object Oriented Analysis

### 4.1 Use Case Diagrams based on Use Case Scenarios

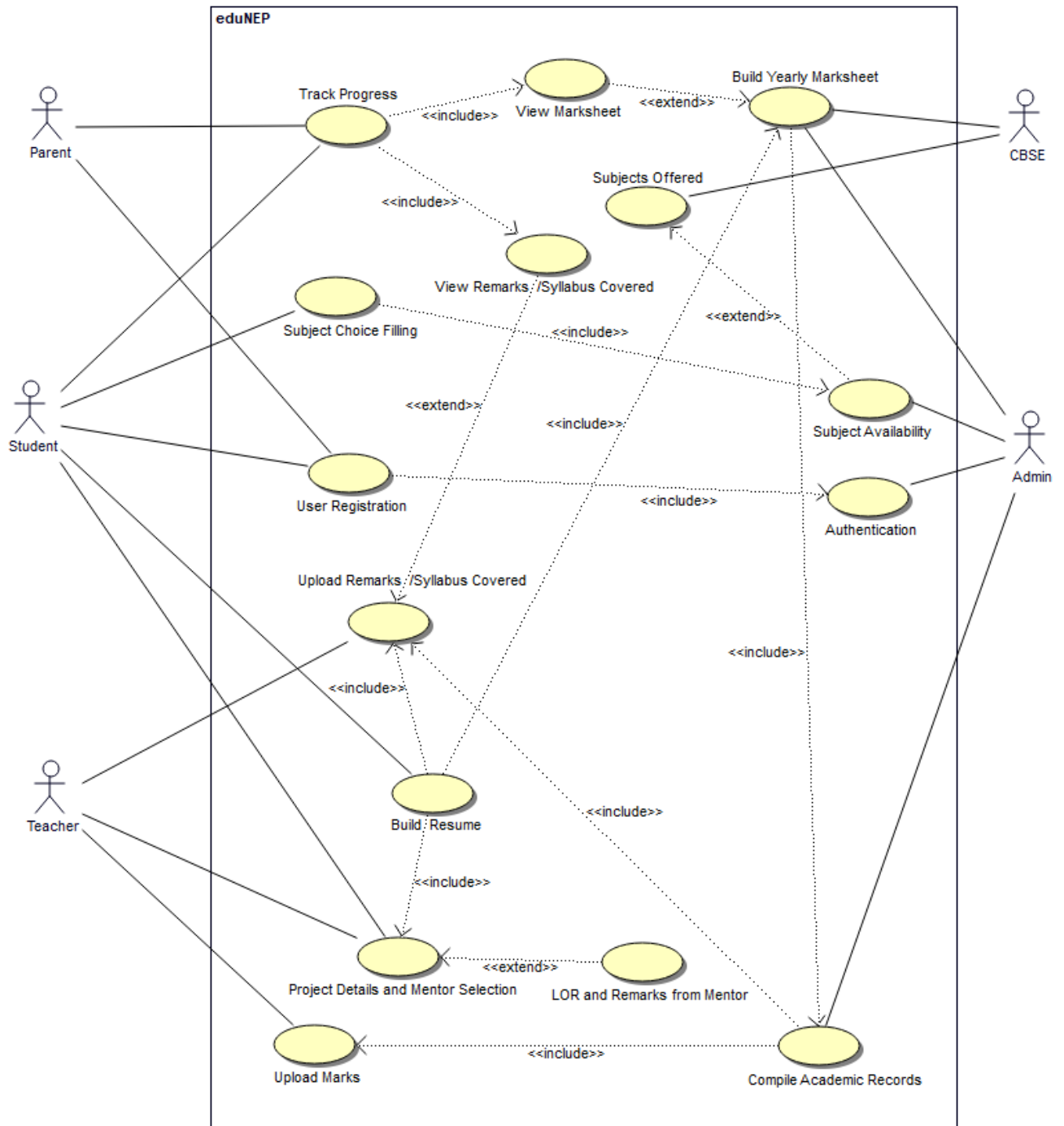


Figure 4.1 Use case diagram

#### 4.2 Activity diagram to depict various procedural and algorithmic flows

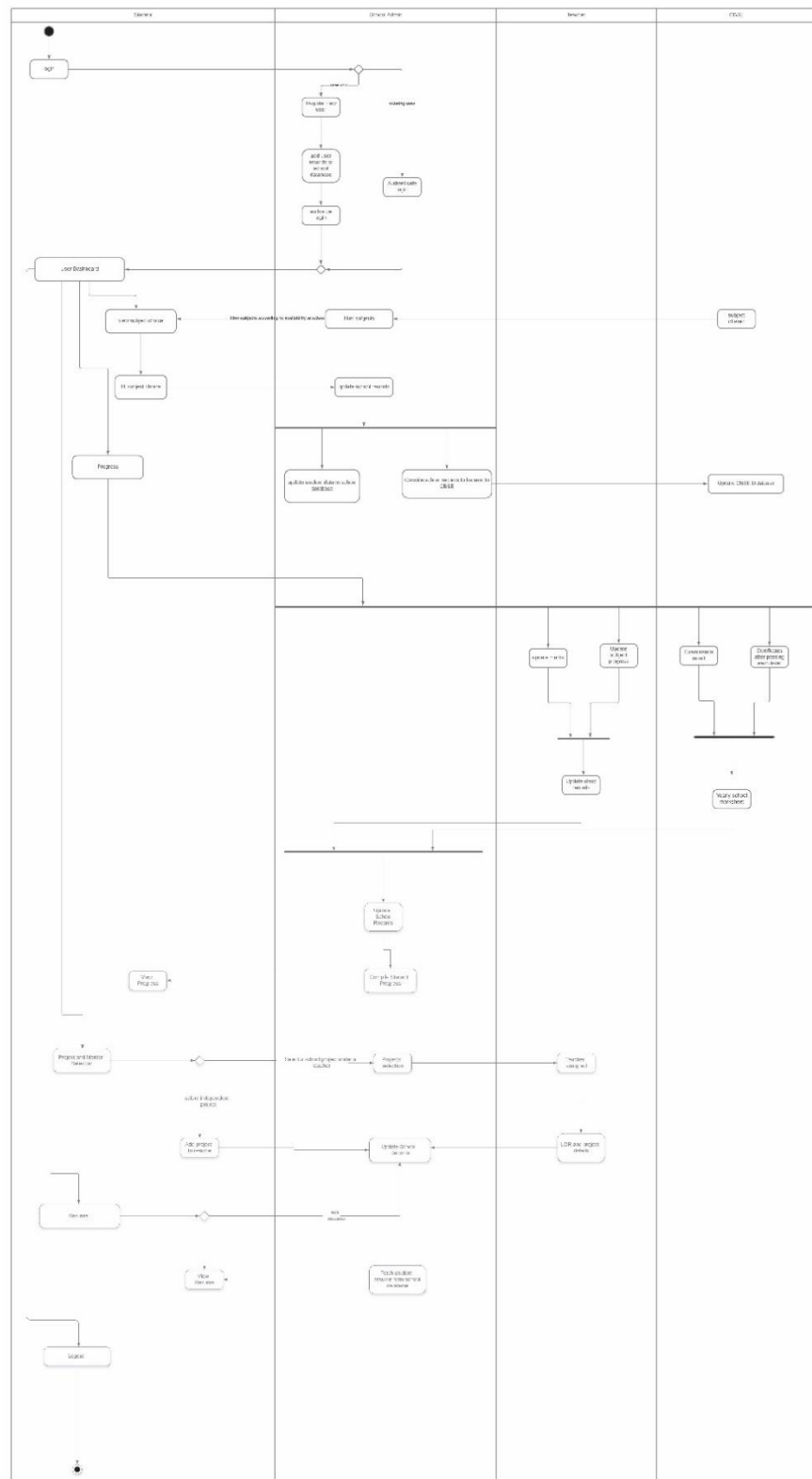


Figure 4.2 Activity diagram

### 4.3 Class diagram showcasing Logical structure

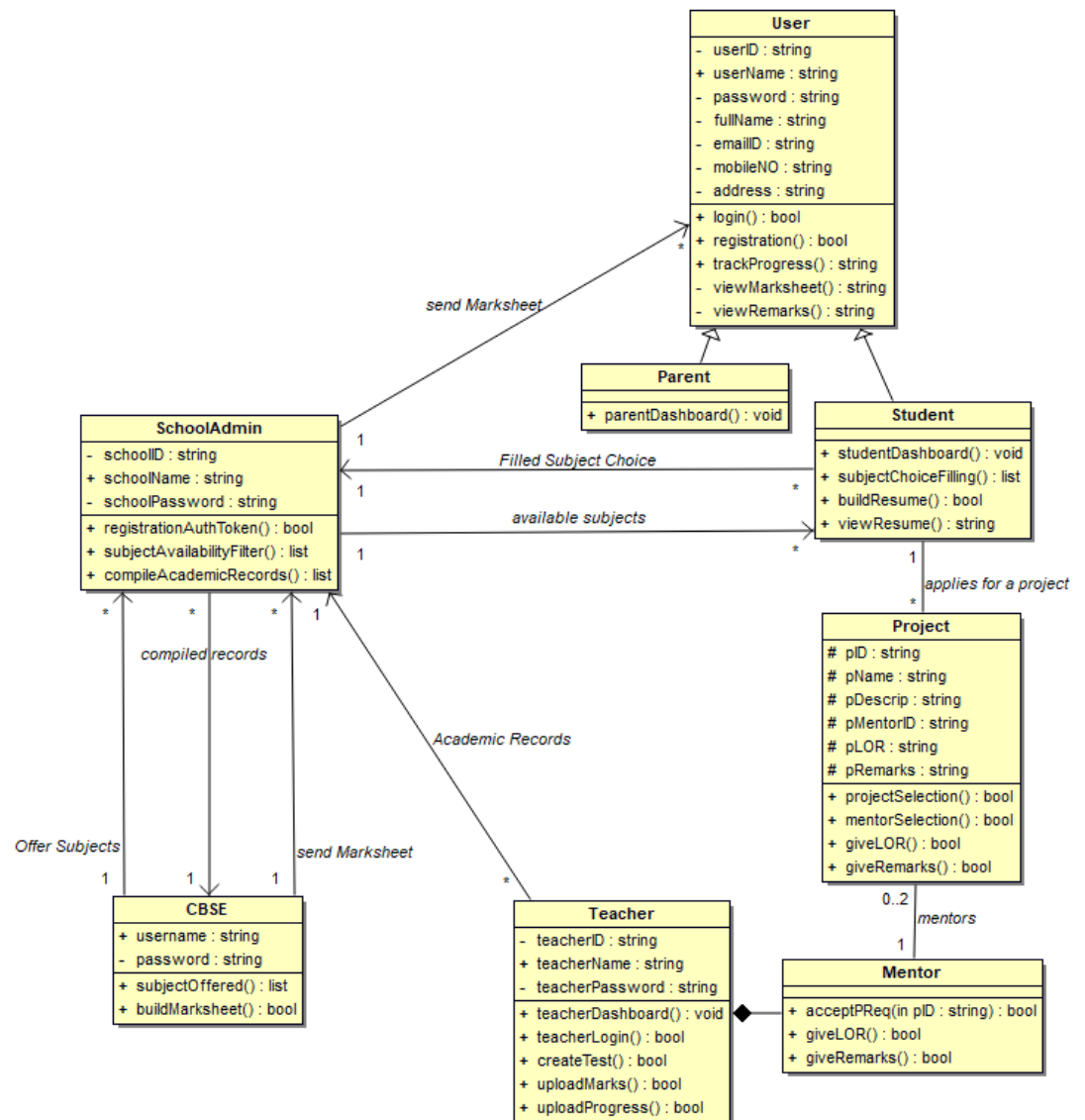


Figure 4.3 Class Diagram

## 4.4 Collaboration Diagram

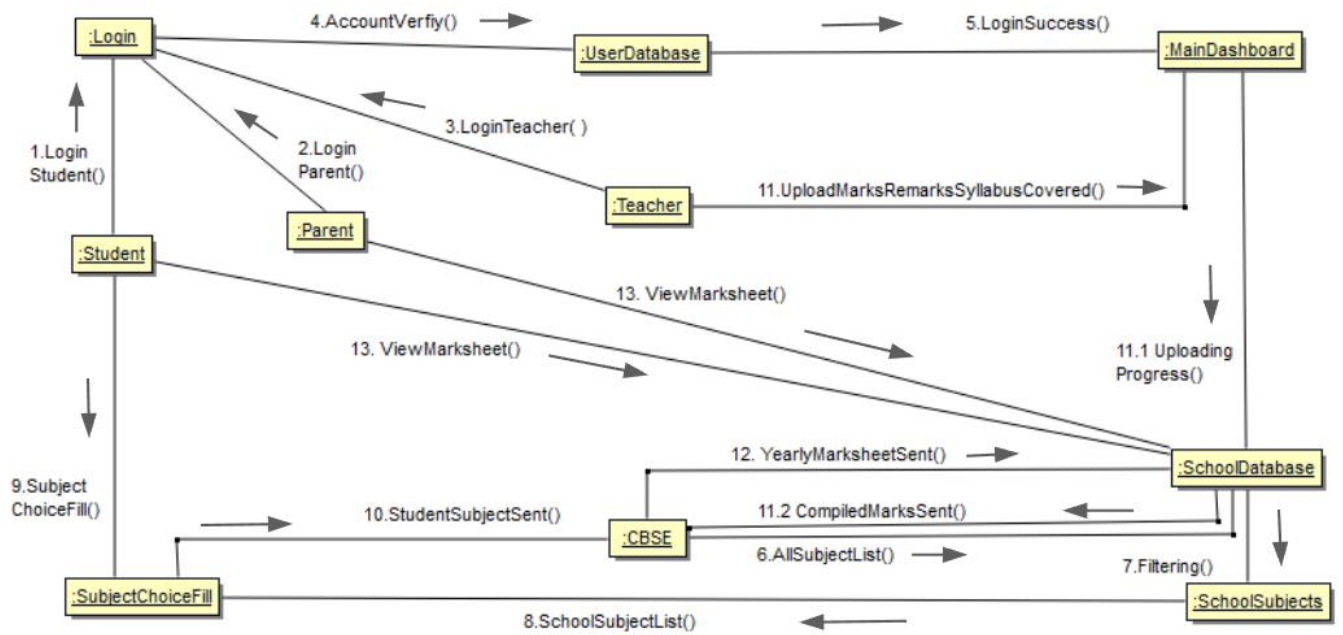


Figure 4.4 Collaboration Diagram

## 4.5 Sequence Diagrams

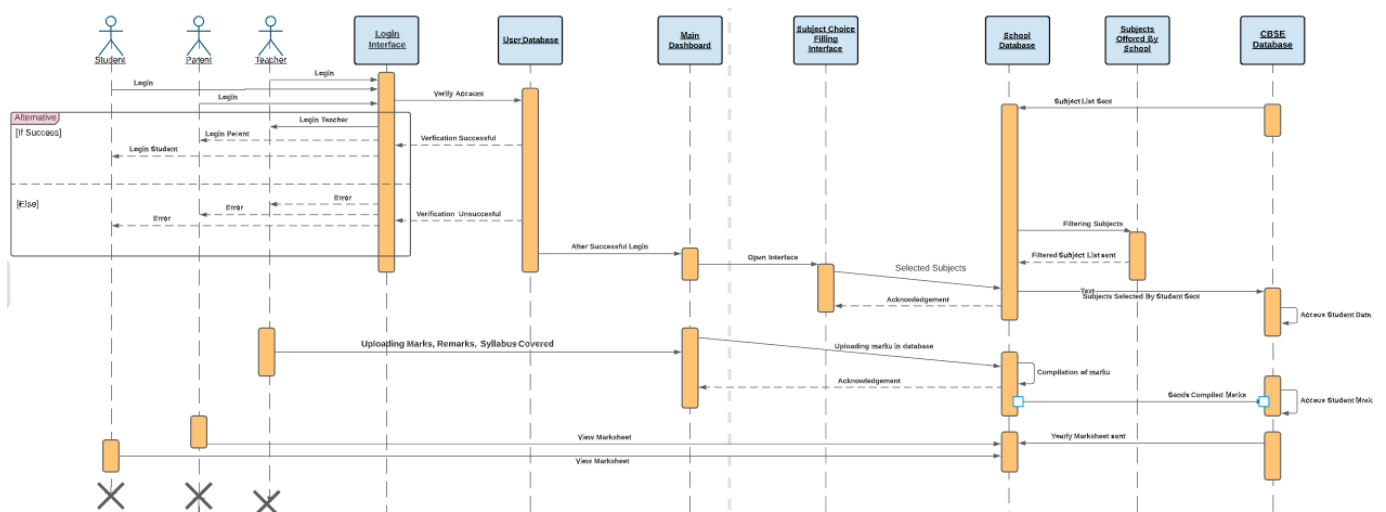


Figure 4.5 Sequence Diagram

## 4.6 State Chart Diagrams

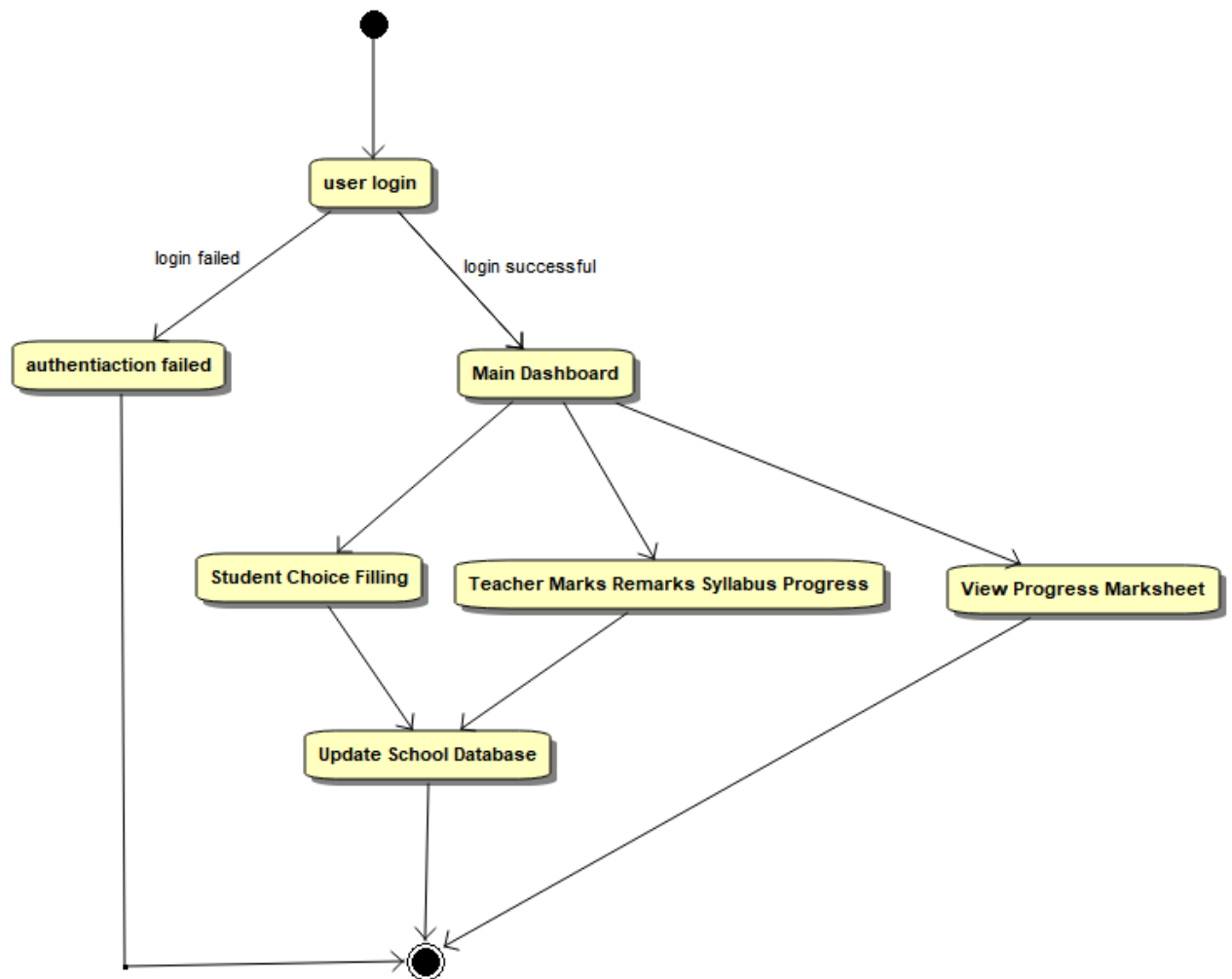


Figure 4.6 State chart Diagram

## 4.7 Component Diagram

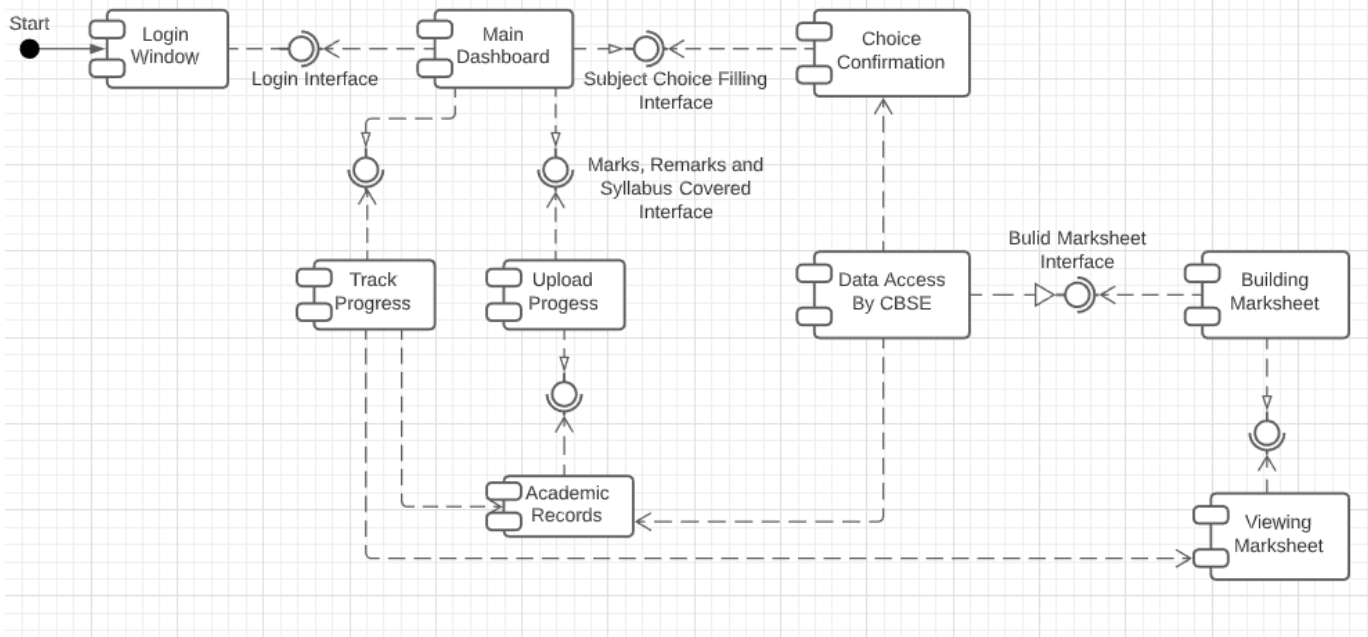


Figure 4.7 Component Diagram

## 4.8 Deployment Diagram

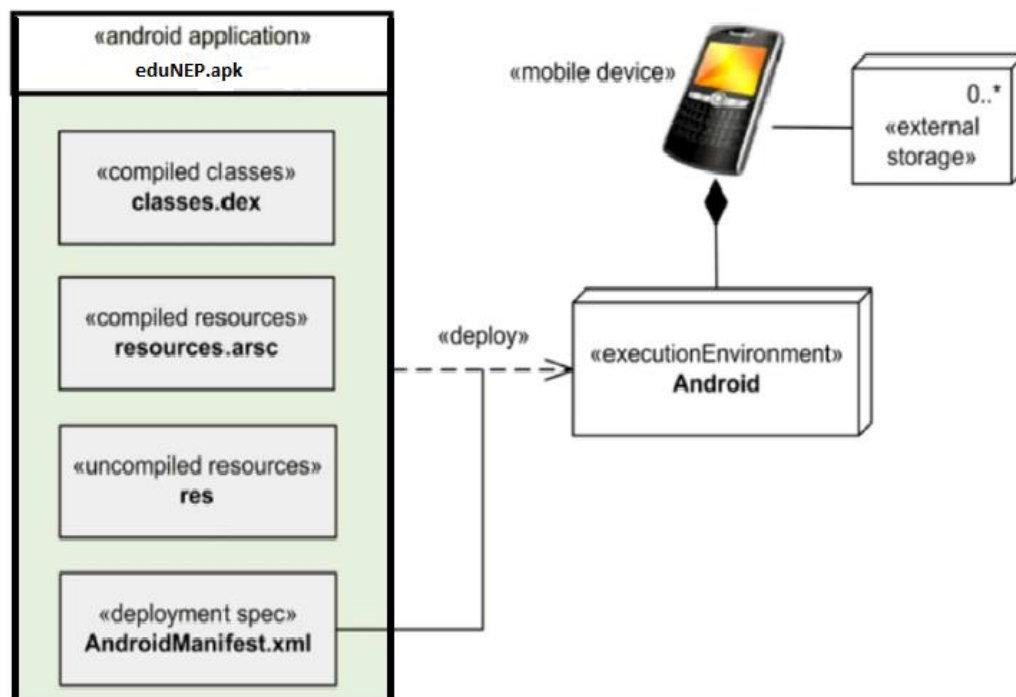


Figure 4.8 Deployment diagram



## 5. Testing


### 5.1 Test Cases for various test strategies (Scenario, Boundary value analysis and Equivalence class testing with ScreenShots)


Test Suite ID	TS001
Test Case	IDTC001
Test Case Summary	To verify that clicking the LOGIN button logs the user into the application.
Prerequisites	The user has a unique login ID.
Test Procedure	If the user is a new user, he clicks on 'Sign up' to register himself. Registered user logs in by entering login id and password.
Test Data	ID – <a href="mailto:ksingh_be18@thapar.edu">ksingh_be18@thapar.edu</a> Password – testing123
Expected Result	Correct login credentials – The user gets logged in. Incorrect login credentials – User not logged in and message 'login failed' is displayed.
Actual Result	If the login credential is valid, the result is as expected. If the login credential is invalid, the expected message is displayed.
Status	Pass
Executed By	Kulpreet and Arshnoor
Date of Execution	30/11/2020
Test Environment	PC OS – Windows 10 Mobile phone OS – Android 10.0 API 29+


17:54 0.00 KB/s 4G


<


## Register Student




 Kulpreet Singh

 Class-10 ▾

 01/09/2000

 ksingh\_be18@thapar.edu

 ....


**Register**


≡ □ ◀


23:20 0.00 KB/s 4G

eduNEP  
Online Learning  
is not the next,  
**big thing**  
it is now the  
**BIG THING,**

## Login



 ksingh\_be18@thapar.edu

 .....

**Login**

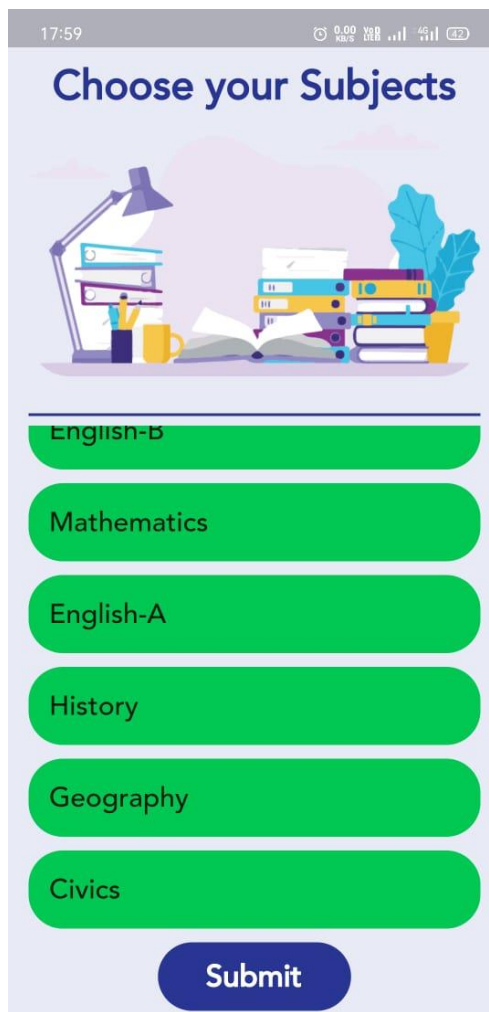
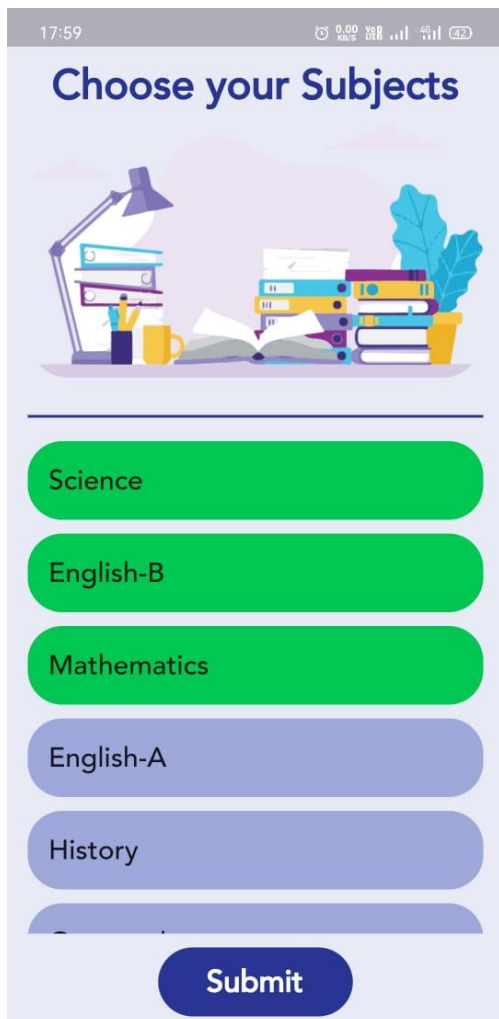
**Sign-Up**

**Syllabus**

≡ □ ◀

Figure 5.1.1 Test Case 1(Login)

Test Suite ID	TS001
Test Case	IDTC002
Test Case Summary	To verify that choosing a subject adds that subject to your subject list.
Prerequisites	The user must be logged in and newly registered.
Test Procedure	Choosing subject by a newly registered user adds it to the subject list.
Test Data	Check the subjects of student's choice from the list of offered subjects.
Expected Result	Selected subject should appear on the student's dashboard.
Actual Result	The result is as expected.
Status	Pass
Executed By	Kulpreet and Arshnoor
Date of Execution	30/11/2020
Test Environment	PC OS – Windows 10 Mobile phone OS – Android 10.0 API 29+



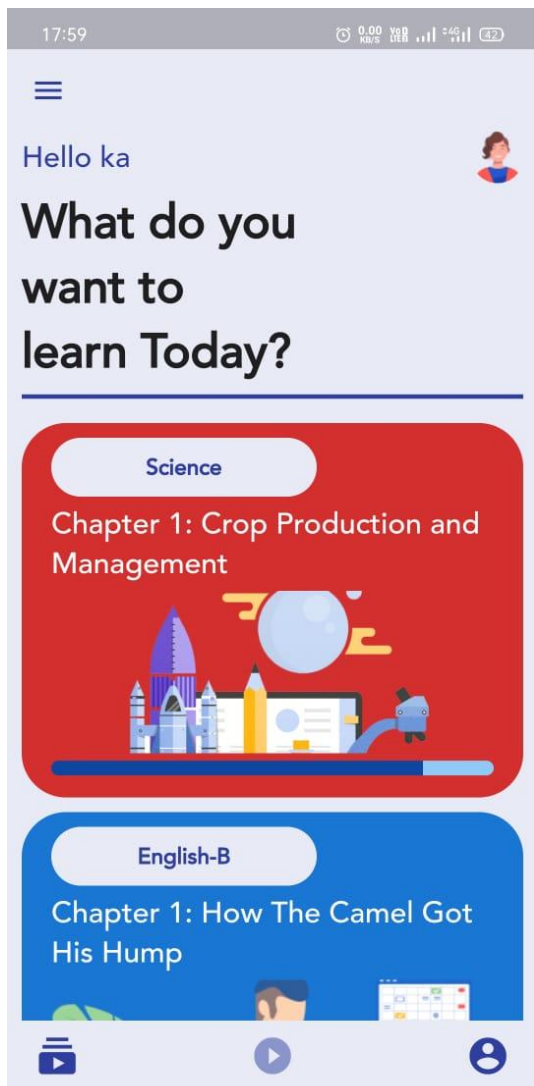


Figure 5.1.2 Test Case 2 (Subject Choice)

Test Suite ID	TS001
Test Case	IDTC003
Test Case Summary	To verify that clicking on app drawer icon on main dashboard screen opens side bar.
Prerequisites	The user must be logged in.
Test Procedure	Clicking on app drawer icon on top left corner in main dashboard screen.
Test Data	None
Expected Result	Side bar should appear over the main dashboard screen.
Actual Result	The result is as expected.
Status	Pass
Executed By	Sajjal and Vishrut
Date of Execution	30/11/2020
Test Environment	PC OS – Windows 10 Mobile phone OS – Android 10.0 API 29+

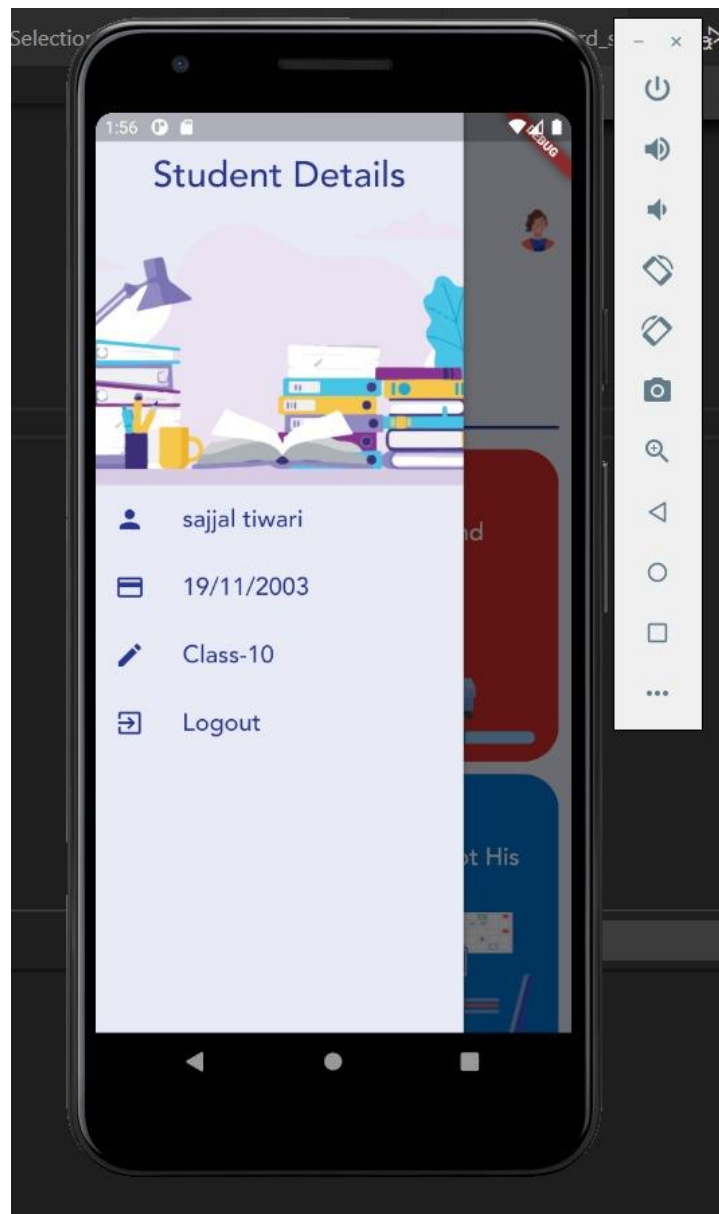


Figure 5.1.3 Test Case 3 (Side Bar)