

|  |
| --- |
| Project Document |
| [Student Management App] |

|  |  |
| --- | --- |
| **C2009L –E Project 4 – Student Management App**  **[Group 5]** | |
| **Group Member** | **La Van Hop**  **Nguyen Lam Khoa**  **Nguyen Dinh Duc**  **Nguyen Thanh Long** |
| **Date** | **21/4/2023** |
| **Instructor** | **Dao Quang Linh** |

- Hanoi, January/<2023> -

**Table of Contents**

[Acknowledgement 4](#_30j0zll)

[Definition and Acronyms 4](#_1fob9te)

[1](#_3znysh7) Project Introduction 5

[1.1](#_2et92p0) Project Information 5

[1.2](#_tyjcwt) Project Team 5

[1.3](#_3dy6vkm) Product Background 5

[1.4](#_1t3h5sf) Existing Systems 5

[1.5](#_4d34og8) The proposed system 5

[1.6](#_2s8eyo1) Project Scope & Limitations 5

[2](#_17dp8vu) Project Management Plan 6

[2.1](#_3rdcrjn) The purpose and target of a project 6

[2.2](#_35nkun2) Management Approach 6

[2.3](#_2jxsxqh) Responsibility Assignments 7

[2.4](#_z337ya) Project Communications 7

[2.5](#_3j2qqm3) Project Work Plan 7

[2.6](#_4i7ojhp) Configuration Management 7

[3](#_2bn6wsx) Software Requirement Specification 8

[3.1](#_qsh70q) User Requirements 8

[3.2](#_3as4poj) Functional Requirements 9

[3.3](#_49x2ik5) Non-Functional Requirements 10

[3.4](#_2p2csry) Other Requirements… 10

[4](#_147n2zr) Software Design Description 11

[4.1](#_3o7alnk) 1. System Design 11

[4.2](#_32hioqz) 2. Database Design 11

[4.3](#_1hmsyys) 3. Detailed Design 11

[5](#_2grqrue) Software Testing Documentation 12

[5.1](#_vx1227) 1. Scope of Testing 12

[5.2](#_3fwokq0) 2. Test Strategy 12

[5.3](#_19c6y18) 3. Test Plan 12

[5.4](#_28h4qwu) 4. Test Cases 12

[5.5](#_37m2jsg) 5. Test Reports 12

[6](#_1mrcu09) Release Package & User Guides 13

[6.1](#_46r0co2) 1. Deliverable Package 13

[6.2](#_111kx3o) 2. Installation Guides 13

[6.3](#_4k668n3) 3. User Manual 13

# Acknowledgement

*[Fill team’s acknowledgement here…]*

During the time of studying, we have well provided and imparted for all the necessary and highly valuable of professional knowledge by our teacher in Aptech, we have our faithful, careful style and creative mind in studying in the future.

Project is a good opportunity to apply appropriate for our knowledge and subjects. While performing project, we have actual and valuable experience. After one term, we were getting all one’s thoughts together and labor at a task of project. During executing this project, we have received a lot of helping from our guider, teachers.

Especial, we wish thank to our teacher- Mr. Dao Quang Linh - for his enthusiastic, thoughtful, and careful guidance.

# Definition and Acronyms

*[Fill all the definitions, acronyms,… used within the document] in the table format as below]*

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| BA | Business Analysis |
| BR | Business Rule |
| ERD | Entity Relationship Diagram |
| GUI | Graphical User Interface |
| PM | Project Manager |
| SDD | Software Design Description |
| SPMP | Software Project Management Plan |
| SRS | Software Requirement Specification |
| UC | Use Case |
| API | Application Program Interface |

# Project Introduction

## Project Information

*Project name: Student Management*

*Project code: E-Project04-Student Management App*

*Group name: Group 5*

## Project Team

*List of the member of project team:*

**+ La Van Hop**

**+ Nguyen Lam Khoa**

**+ Nguyen Dinh Duc**

**+ Nguyen Thanh Long**

The lecturer supporting the project is Mr. Dao Quang Linh.

## Product Background

- In today's society, there are quite a few universities with thousands of students. Because education is always one of the top concerns in the country. But with the number of students entering and leaving the school up to hundreds of thousands, the management of information and classrooms of those students is extremely important. But with today's society, the era of advanced technology, the management of students will also be modernized. Instead of before, we store information on traditional papers and books, now, we will be able to store them on software so that student information management will become modern and easy. more than ever. Management software was born to help us manage students easier, more convenient, and save more time. We will no longer need to search, take notes, delete and edit hard on traditional papers anymore. But now we can add students, delete students, edit students directly on the software. Teachers, training rooms only need to log in to the software with an account provided or register themselves. And the next job is just a click away to find a student's profile. Or we can enter the information of a new student to the school. And here, we would like to introduce the student management software that we have created.

## Existing Systems

- Ehust

- Hust Student

## The proposed system

- Currently, at universities, the management of students and students is increasingly facing difficulties and struggles in managing information from grades to subjects, timetables, and personal information. Because of that, we have chosen the topic "Building a student management application for mobile devices", in order to meet the requirements set forth. This application is built on the Android Studio operating system platform and java programming language in installing and processing functions according to specific requirements

## Project Scope & Limitations

The system still has some limitations such as:

+ Unable to upload student’s avata.

+ Unable to connect between majors and classes.

***\**Hardware and Software Requirements:**

|  |  |  |
| --- | --- | --- |
|  | Serve | Client |
| **Hardware** | - Ram 1GB or higher  - CPU 2.0 Ghz or higher  - HDD 80 GB or higher  - Internet & LAN connection | Android phone |
| **Software** | Operating System:  - Windows/Mac/Linux  -Web Server Software:  -Database: SQLite Server |  |

# Project Management Plan

## The purpose and target of a project

### Project Objectives

Project goal:

- The project must be satisfactory and completed on time.

- fully functional project, no bugs.

- The app must have a reasonable layout and high aesthetics

### Project Risks

*[List out the details on project risks in the table below]*

## Management Approach

*-* The approach and project management puzzle is to break the project down into several phases. Each phase of the project must be completed before the start of the next, leading to possible final completion.

### Project Process

Diagram

Description automatically generated

### Quality Management

*-* Project quality is about 90%. There were still a few bugs, but they were quickly fixed.

### Training Plan

*-* Team members have enough knowledge/skills to handle

## Responsibility Assignments

*[Describe the main responsibilities in your project (to complete the outputs as defined in the above section), in the format as the sample below]*

Project Manager

Nguyễn Văn A

Analysis

Nguyến

Design

Nguyễn Văn B

Code

Nguyễn Văn A

Nguyễn Văn B

Nguyễn Văn C

Test

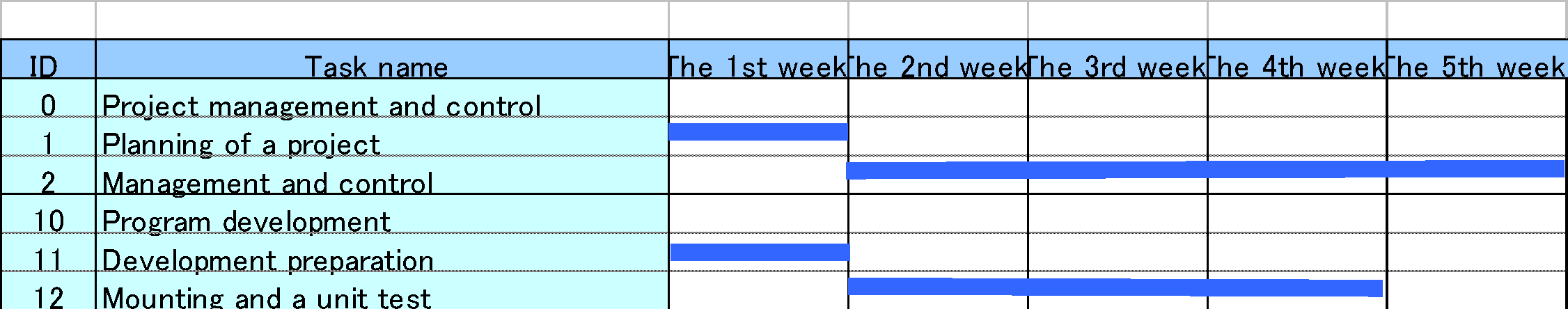
Nguyễn Văn B

Nguyễn Văn C

## Project Communications

*[Provide the details of project communication plan, the to-be-used tools, the project interface, etc.]*

## Project Work Plan



## Configuration Management

### Document Management

*-* Our group use github to manage the project and update the versions of the project

### Source Code Management

*-* Our group use github to manage the project and update the versions of the project

### Tools & Infrastructures

*-* Our group use *Android Studio*, *SQLite Database* to develop and deploy the project*.*

# Software Requirement Specification

## User Requirements

*Diagram

Description automatically generated*

**Use Case**

|  |  |
| --- | --- |
| Use case ID | Uc-01 |
| Use case name | Student Management |
| Actors | User |
| Description | User create, update, delete a student |
| Pre-condition | User device is android phone |
| Post-condition | The student list will show up and be changed by user |
| Trigger | User want to display the student list |
| Main flow | |  |  |  | | --- | --- | --- | | No. | Actor | Action | | 1 | All | Access Class List | | 2 | System | Display Class List | | 3 | All | Click Edit Button to Edit Student | | 4 | All | Click Delete Button to Delete Student | | 5 | System | Select student in a database | | 6 | System | Display Student List | |
| Alternative flow | N/A |
| Exception flow | N/A |
| Business Rules | N/A |

|  |  |
| --- | --- |
| Use case ID | Uc-02 |
| Use case name | Search Student |
| Actors | User |
| Description | User search for student |
| Pre-condition | User’s device has android phone |
| Post-condition | The students will show up in the search results |
| Trigger | user wants to find specific students |
| Main flow | |  |  |  | | --- | --- | --- | | No. | Actor | Action | | 1 | All | Access Student List | | 2 | System | Display Student List | | 3 | All | Choose search type “Nhap ten can tim kiem“ | | 4 | All | Fill out search input by the first letter of student name then click on search button | | 5 | System | Select fit student in a database | | 6 | System | Display search result list | |
| Alternative flow | N/A |
| Exception flow | N/A |
| Business Rules | N/A |

*\* Subject:*

|  |  |
| --- | --- |
| Use case ID | Uc-03 |
| Use case name | Subject Management |
| Actors | User |
| Description | User create, update, delete a subject |
| Pre-condition | User device is android phone |
| Post-condition | The subject list will show up and be changed by user |
| Trigger | User want to display the subject list |
| Main flow | |  |  |  | | --- | --- | --- | | No. | Actor | Action | | 1 | All | Access Subject List | | 2 | System | Display Subject List | | 3 | All | Click Edit Button to Edit Subject | | 4 | All | Click Delete Button to Delete Subject | | 5 | System | Select subject in a database | | 6 | System | Display Subject List | |
| Alternative flow | N/A |
| Exception flow | N/A |
| Business Rules | N/A |

|  |  |
| --- | --- |
| Use case ID | Uc-04 |
| Use case name | Search Subject |
| Actors | User |
| Description | User search for subject |
| Pre-condition | User device is android phone |
| Post-condition | The subjects will show up in the search results |
| Trigger | user wants to find specific subjects |
| Main flow | |  |  |  | | --- | --- | --- | | No. | Actor | Action | | 1 | All | Access Subject List | | 2 | System | Display Subject List | | 3 | All | Choose search type “Nhap ten mon hoc can tim “ | | 4 | All | Fill out search input by the first letter of subject name then click on search button | | 5 | System | Select fit subject in a database | | 6 | System | Display search result list | |
| Alternative flow | N/A |
| Exception flow | N/A |
| Business Rules | N/A |

*\* Major:*

|  |  |
| --- | --- |
| Use case ID | Uc-05 |
| Use case name | Major Management |
| Actors | User |
| Description | User create, update, delete a major |
| Pre-condition | User device is android phone |
| Post-condition | The major list will show up and be changed by user |
| Trigger | User want to display the major list |
| Main flow | |  |  |  | | --- | --- | --- | | No. | Actor | Action | | 1 | All | Access Major List | | 2 | System | Display Major List | | 3 | All | Click Edit Button to Edit Major | | 4 | All | Click Delete Button to Delete Major | | 5 | System | Select major in a database | | 6 | System | Display Major List | |
| Alternative flow | N/A |
| Exception flow | N/A |
| Business Rules | N/A |

|  |  |
| --- | --- |
| Use case ID | Uc-06 |
| Use case name | Search Major |
| Actors | User |
| Description | User search for major |
| Pre-condition | User’s device is android phone |
| Post-condition | The majors will show up in the search results |
| Trigger | user wants to find specific majors |
| Main flow | |  |  |  | | --- | --- | --- | | No. | Actor | Action | | 1 | All | Access Major List | | 2 | System | Display Major List | | 3 | All | Choose search type “Nhap ten nganh can tim“ | | 4 | All | Fill out search input by the first letter of major name then click on search button | | 5 | System | Select fit major in a database | | 6 | System | Display search result list | |
| Alternative flow | N/A |
| Exception flow | N/A |
| Business Rules | N/A |

## Functional Requirements

### System Functional Overview

*[Provide functionality overview of software system: screen flow, screen descriptions, system user roles, screen authorization, non-screen functions, ERD]*

*Diagram

Description automatically generated*

**\* User:**

* View Home page:

|  |  |
| --- | --- |
| Input | User accesses homepage |
| Process | View the main screen of the App |
| Output | The home page will display all function to access Student List, Major List, Subject List, Class List, Information App. |
| Data save |  |

* View Class List, delete and edit class:

|  |  |
| --- | --- |
| Input | User click Class List |
| Process | Search the database and get all the information of the Class List : maLop, tenLop |
| Output | The class page will display all existing classes. Next is the Edit button which is used to change the information of the class, and the delete button to removal class when it is done. |
| Data save | Class Information: maLop, tenLop |

* Find information of Class:

|  |  |
| --- | --- |
| Input | Search by keywords that users enter: the first letter of the Class name |
| Process | Search the database and retrieve the information of Classes which has the first letter as the user searched for |
| Output | Display a list of Class that match the information entered by the user or a message that no results were found if the database has no matching records. |
| Data Save | Search keywords: the first letter of the Class name |

* Add New Class

|  |  |
| --- | --- |
| Input | Enter the Class information. The information includes maLop, tenLop. |
| Process | Verify the validity of the information and execute the process of adding new Class in the database. |
| Output | Show add success or failure status |
| Data Save | The information includes maLop, tenLop. |

* View Subject List, delete and edit subject:

|  |  |
| --- | --- |
| Input | User click Subject List |
| Process | Search the database and get all the information of the Subject List: maMH, tenmonhoc |
| Output | The class page will display all existing Subject. Next is the Edit button which is used to change the information of the subject, and the delete button to removal subject. |
| Data save | Class Information: maMH, tenmonhoc |

* Find information of Subject:

|  |  |
| --- | --- |
| Input | Search by keywords that users enter: the first letter of the Subject name |
| Process | Search the database and retrieve the information of Subjects which has the first letter as the user searched for |
| Output | Display a list of Subject that match the information entered by the user or a message that no results were found if the database has no matching records. |
| Data Save | Search keywords: the first letter of the Subject name |

* Add New Subject

|  |  |
| --- | --- |
| Input | Enter the Class information. The information includes maMH, tenmonhoc. |
| Process | Verify the validity of the information and execute the process of adding new Subject in the database. |
| Output | Show add success or failure status |
| Data Save | The information includes maMH, tenmonhoc. |

* View Major List, delete and edit major:

|  |  |
| --- | --- |
| Input | User click Major List |
| Process | Search the database and get all the information of the Major List: maChuyenNganh, tenChuyenNganh |
| Output | The class page will display all existing Major. Next is the Edit button which is used to change the information of the Major, and the delete button to removal Major. |
| Data save | Class Information: maChuyenNganh, tenChuyenNganh |

* Find information of Major:

|  |  |
| --- | --- |
| Input | Search by keywords that users enter: the first letter of the Major name |
| Process | Search the database and retrieve the information of Majors which has the first letter as the user searched for |
| Output | Display a list of Major that match the information entered by the user or a message that no results were found if the database has no matching records. |
| Data Save | Search keywords: the first letter of the Major name |

* Add New Major

|  |  |
| --- | --- |
| Input | Enter the Class information. The information includes maChuyenNganh, tenChuyenNganh. |
| Process | Verify the validity of the information and execute the process of adding new Major in the database. |
| Output | Show add success or failure status |
| Data Save | The information includes maChuyenNganh, tenChuyenNganh. |

* View Student List, delete and edit student:

|  |  |
| --- | --- |
| Input | User click Student List |
| Process | Search the database and get all the information of the Student List: maSV,tenSV,email,maLop,maChuyenNganh |
| Output | The class page will display all existing Student. Next is the Edit button which is used to change the information of the Student, and the delete button to removal Student. |
| Data save | Class Information: maSV,tenSV,email,maLop,maChuyenNganh |

* Find information of Student:

|  |  |
| --- | --- |
| Input | Search by keywords that users enter: the first letter of the Student name |
| Process | Search the database and retrieve the information of Students which has the first letter as the user searched for |
| Output | Display a list of Student that match the information entered by the user or a message that no results were found if the database has no matching records. |
| Data Save | Search keywords: the first letter of the Student name |

* Add New Student

|  |  |
| --- | --- |
| Input | Enter the Class information. The information includes maSV, tenSV, email, maLop, maChuyenNganh. |
| Process | Verify the validity of the information and execute the process of adding new Student in the database. |
| Output | Show add success or failure status |
| Data Save | The information includes maSV, tenSV ,email, maLop, maChuyenNganh. |

* View Information Page

|  |  |
| --- | --- |
| Input | User click Information Page |
| Process | Appear Information App |
| Output | The class page will display information of Hanoi University of Science and Techlonogy School (HUST) |
| Data save |  |

## Non-Functional Requirements

### Reliability

-This is a communication website, so the reliability is very important. The information from this website must be real and exactly (phone number, store address)

### Availability

Systems created using the platform should be generally available to user.

### 3.2.3.3 Security

The security is necessary for almost website. Protect user information is important

### Maintainability

The platform should contain interfaces that are extensible to allow new -communication protocols to be added with ease.

### Portability

We can connect website by every device have internet

## Other Requirements…

# Software Design Description

*[Provide final software design information follow the template as part II in the Report #4]*

## 1. System Design

### 1.1 System Architecture

*[The content of this section includes the overall diagram which includes the sub-systems, the external systems, and the relationship/connection among them. You need also provide the explanation for each of the diagram components (modules, sub-systems, external systems, etc.)].*

### 1.2 Package Diagram

*[Provide the package diagram for each sub-system. The content of this section includes overall package diagram(s) and the explanation for each package (or namespace)]*

## 2. Database Design

*1.User:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Name of attributes** | **Type** | **Size** | **Description** |
| **Primary key** | tenTaiKhoan | Text | 10 | Username to login |
|  | matKhau | Text | 10 | Password to login |

*2.Class:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Name of attributes** | **Type** | **Size** | **Description** |
| **Primary key** | maLop | Text | 10 | Class code |
|  | tenLop | Text | 10 | Class name |

3.Subject:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Name of attributes** | **Type** | **Size** | **Description** |
| **Primary key** | maMH | Text | 10 | Subject code |
|  | tenmonhoc | Text | 10 | Subject name |

4.Major:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Name of attributes** | **Type** | **Size** | **Description** |
| **Primary key** | maChuyenNganh | Text | 10 | Major code |
|  | tenChuyenNganh | Text | 10 | Major name |

5.Student:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Name of attributes** | **Type** | **Size** | **Description** |
| **Primary key** | maSV | Text | 10 | Student code |
|  | tenSV | Text | 30 | Student name |
|  | email | Text |  | Student email |
|  | maLop | Text | 10 | Class code |
|  | maChuyenNganh | Text | 10 | Major Code |

## 3. Detailed Design

### 3.1 <Feature/Function Name1>

*[Provide the detailed design for the feature <Feature Name1>. It includes Class Diagram, Class Specifications, and Sequence Diagram(s);* ***For the features/functions with the same structure of class & sequence diagrams, you need to provide the diagrams once for one feature/function and refer to those diagrams from other features/functions****]*

*Diagram

Description automatically generated*

Note: Enter class| student | major needs to add, update, delete. Check if the input data is correct. If correct, will perform new result at the end of the action. If the data is wrong, a message will be displayed to let the user know.

Diagram

Description automatically generated

Note: **For *search function****,* search for class| student | major information by name.

**Check**: check if the user entered correctly or not and report an error to re-enter. Search in the database: find the user's information about class| student | major in the database.

**Display class| student | major information**: display the class| student | major information found in the database that matches the information the user needs to find.

# Software Testing Documentation

*[Provide final software testing information follow the template as part II in the Report #5]*

## 1. Scope of Testing

*[Describe the scopes of the test. Those include the target-of-test’s features, functions, and non-functional requirements that will or will not be tested.*

*Describe the stages/levels of testing that would be applied to your project - Unit, Integration, or System test. Each includes the in-charge, inputs/time, focuses, acceptance criteria.*

*List any constraints or assumptions made during the development of this document that may impact the design, development or implementation of testing]*

## 2. Test Strategy

*[List out and describe all testing types (you can refer the test types listed below or any other test types to selected the suitable ones for the project; for each selected test types you need to provide the following information: test objective, technique, completion criteria, etc.), test levels that those test types would be performed, & the details of test supporting tools would be used in the project]*

### 2.1 Testing Types

*[List out and describe here the testing types which you would apply in your project. You need to mention following information for each type of testing: objective, technique, completion criteria]*

### 2.2 Test Levels

*<List out and describe here the testing levels which you would execute in your project. Besides, clearly state the test types which are performed in each test level that you plan for this project>*

### 2.3 Supporting Tools

*<List of the test supporting tools which will be employed for this project>*

## 3. Test Plan

### 3.1 Human Resources

*[List and provide the details on roles and responsibilities of the project members who would involve in testing works]*

### 3.2 Test Environment

*[List and provide the details about the tools (software, hardware, infrastructure) which the project would use for testing]*

### 3.3 Test Milestones

*[Separate test milestones, which should be identified to communicate project status accomplishments]*

## 4. Test Cases

*[Prepare the details on the test cases following the provided template*

* *Unit Test Cases: Report5\_Unit Test.xls*
* *Other Test Cases (IT, ST, AT): Report5\_Test Report.xls]*

## 5. Test Reports

*[Provide the test result, statistics and the relevant test analysis for your testing in the project]*

# Release Package & User Guides

*[Provide final software testing information follow the template as part II in the Report #6]*

## 1. Deliverable Package

## 2. Installation Guides

### 2.1 System Requirements

*[Define any system requirements necessary to support the application, including the software and relevant configurations]*

### 2.2 Installation Instruction

*[Includes installation instructions and configuration guidelines]*

## 3. User Manual

### 3.1 Overview

*[Descript the overview of the application and if could, insert the features workflow to help user has the overview of all the features in this application]*

(\*) With user:

- The app has a homepage- which will display Student List Button, Major List Button, Subject List Button, Information App button, Logout Button, Class List.

- When you click on any button, you will be accest to different type of List and you can add-update-delete the element in the List.

- Users can register or change password.