

THE TECHNICAL UNIVERSITY OF KENYA

SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SYSTEMS AND TECHNOLOGY

**SCIT STUDENTS PROJECT PROGRESS MANAGEMENT SYSTEM (SCIT SPPMS)**

A CASE STUDY OF THE SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY AT THE TECHNICAL UNIVERSITY OF KENYA

PRESENTED BY:

**CHARLES KAIRU**

**SCII/00329/2016**

SUPERVISED BY:

MR. PETER KARIUKI

MS. NJERI NGARUIYA

A PROJECT PROPOSAL SUBMITTED TO THE SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY IN PARTIAL FULFILMENT FOR THE DEGREE OF TECHNOLOGY IN INFORMATION TECHNOLOGY PROJECT OF THE TECHNICAL UNIVERSITY OF KENYA

## TABLE OF FIGURES

[Figure 1: Level 0 DFD 6](#_Toc34068195)

[Figure 2: Level 1 DFD 7](#_Toc34068196)

[Figure 3: Level 2 DFD 8](#_Toc34068197)

[Figure 4: Use case diagram 9](#_Toc34068198)

[Figure 5: Class Diagram 10](#_Toc34068199)

[Figure 6: Sign Up & Sign In flowchart 11](#_Toc34068200)

[Figure 7: Students’ flowchart 12](#_Toc34068201)

[Figure 8: Supervisors’ flowchart 13](#_Toc34068202)

[Figure 9: Project coordinators’ flowchart 14](#_Toc34068203)

[Figure 10: Sign Up wireframe 15](#_Toc34068204)

[Figure 11: Login wireframe 16](#_Toc34068205)

[Figure 12: Students’ dashboard wireframe (a) 16](#_Toc34068206)

[Figure 13: Students’ dashboard wireframe (b) 17](#_Toc34068207)

[Figure 14: Add project wireframe 17](#_Toc34068208)

[Figure 15: Add milestone wireframe 18](#_Toc34068209)

[Figure 16: Supervisors’ dashboard wireframe 18](#_Toc34068210)

[Figure 17: Approving milestones wireframe 19](#_Toc34068211)

[Figure 18: Project coordinator’s dashboard wireframe (a) 19](#_Toc34068212)

[Figure 19: Project coordinator’s dashboard (b) 20](#_Toc34068213)

[Figure 20: Conceptual database design 21](#_Toc34068214)

[Figure 21: Logical database design 22](#_Toc34068215)

[Figure 22: Allocations’ schema 23](#_Toc34068216)

[Figure 23: Projects’ schema 23](#_Toc34068217)

[Figure 24: Milestones’ schema 24](#_Toc34068218)

[Figure 25: Students’ schema 24](#_Toc34068219)

[Figure 26: Project coordinator’s schema 25](#_Toc34068220)

[Figure 27: Supervisors’ schema 25](#_Toc34068221)

## CHAPTER FOUR: SYSTEM ANALYSIS AND REQUIREMENT MODELING

### Introduction

System analysis is the second phase of SDLC and is a problem-solving technique of studying a system in order to determine its goals and create a software or system that will meet them efficiently. It divides a system into its components for the purpose of determining its objectives. This section describes the requirements which will need to be met by the proposed system for it to be considered efficient, effective and complete. Different UML diagrams will also be used later in this section to better visualize and understand the system requirements.

### How the facts and data of the current system were gathered

**Participation and Observation**: Being the system analyst of this system and a member of SCIT, I was able to observe and identify the exact project management processes, system functionality, users’ inputs, and system outputs.

**Advantages of using Participation and Observation**

1. A very direct method for collecting data
2. Data collected is very accurate in nature and also very reliable.
3. Improves the precision of the research results.
4. The problem of depending on respondents is decreased.
5. Observation is less demanding in nature, which makes it less bias in working abilities.
6. By observation, one can identify a problem by making an in-depth analysis of the problems.

To further understand user requirements and how to improve the existing system, a **Questionnaire** was used to collect data from students. The appendix section contains a sample of questions used.

**Advantages of using a questionnaire**

1. It saves time as the student could fill it quickly on their own time
2. The cost of conducting the study with the help of the questionnaire method is very low
3. It helped to capture a larger audience than using an interview
4. It puts less pressure on the respondents for immediate response. He can answer it at his own leisure.

### Requirement definitions of the current system

The current system is a manual system where projects are managed manually by use of papers (project progress forms) and important notices communicated via email. The following are the current user requirements of the manual system.

**User Requirements**

1. **Students**
   1. Concept paper – students write a brief description of their project idea and submit to the project co-ordinator manually. The students then wait for the evaluation of the concept paper. If accepted, the students proceed to the next step.
   2. Submit personal details and project title – Students submit their details and project titles to the project co-ordinator and wait for the coordinator to schedule and allocate supervisors to them.
   3. Progress form – Students print a progress form received via email form the project coordinator. They are supposed to carry the project form every time they meet with their supervisor in order to feel in remarks.
2. **Supervisor**
3. Monitoring students’ progress – the supervisor monitors students’ progress and evaluates their milestones by giving them remarks which are filled on the progress form.
4. Approving the project – Supervisors approve students’ projects by signing in the progress form and project report for students to do the official presentation of the project.
5. **Project coordinator**
   1. Schedule and allocate supervisors to students.
   2. Give important project notices and guidelines such as requirements and deadlines.

### Requirement definitions and specifications of the proposed system

#### System requirements

System requirements involves defining the programs and hardware required in order to run or use the software product. SPPMS will be a web-based application and the following are the minimum system requirements.

1. A computer or a mobile device with the capability to connect to the internet.
2. An operating system e.g. Mac O.S or Windows for computers and Android or iOS for mobile devices.
3. A browser – The computer should be preinstalled with a browser e.g. chrome.
4. Internet access – This can be Wi-Fi, Ethernet connection or data bundles from an ISP of choice.
5. Power – The device should be powered through a direct cable power connection or battery.

#### Functional Requirements

Functional requirement describes the services that the system provides. It gives detailed information on the inputs, behavior and outputs of the system. It can be how the users interact with the system, data manipulation or calculations performed by the system. The following are SPPMS function requirements.

1. **Registration and Login** – New users should be able to register to the system using the first and last name, email address and create a password. A password should have a minimum length of 6. Registered users can then log in using an email and password. The system comes with a default super admin account which will be used by the project coordinator only. After registering as a user, by default you get the students' account. In order to gain supervisors' privileges, a new user should submit a request to the project coordinator by filling in form designed for that purpose only. The project coordinator reviews the request and decides when to approve it or deny it.
2. **Adding a new project** – This functionality is done by students. A form is filled with some students’ details such as registration number and project details such as title and a short description of the project. Projects are stored in the project's collection in the database. Students can view the project status and can remove their projects from the system.
3. **Adding a new milestone** – One of the key functionality of the system is to enable students to divide the system into milestones. Students create milestones for their projects by filling in the milestones form. They can view milestone remarks and status, and can also delete a milestone they created. Milestones are stored in the milestones sub-collection in the database.
4. **Approving a project/milestone –** This functionality is performed by the supervisors. Approved projects or milestones are marked as complete with a green budge which unapproved projects and milestones are marked as incomplete with a blue budge.
5. **Adding milestone remarks –** After evaluation of each milestone, the supervisors can enter in the remarks. These remarks can then be viewed by the student and project coordinator.
6. **Allocation of user –** The system allows the project coordinator to schedule and allocate supervisors to students. Allocated users can view their allocations.

#### Non-functional Requirement

Non-functional requirements refer to the quality attributes of a system. The plan for implementing non-functional requirements is usually detailed in the system design. Failing to meet the non-functional requirements can result in a software product that fails to meet the users' needs.

1. **Reliability** – The system should be reliable in that; it should not fail throughout the project management period. Failure of a single component in the system should not bring down the whole system.
2. **Performance** – The system should be fast and operate accurately and efficiently.
3. **Availability** – The system should run 24 hours a day and 7days a week. Any registered user should be able to access the system at any time for monitoring and evaluation purposes.
4. **Data integrity** – Proper database rules should be written to ensure that the data is secure. Secure data is not altered which makes it to remain accurate, complete and reliable.
5. **Confidentiality** – Students should only see their own project and milestones progress and not others.
6. **Scalability** – The system should be able to expand with an increase in the number of users and amount of data.
7. **Security** – Unauthorised users should not be able to enter the system or manipulate the data stored in the database.
8. **Capacity** – The system should be able to handle at least ten thousand users at a go without affecting its performance.
9. Students should not be able to approve their own projects or milestones.

### Data Flow Diagrams

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles, and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one.

#### Level 0 DFD: Context Level Diagram

A context diagram gives an overview and it is the highest level in a data flow diagram, containing only one process representing the entire system.

1. All external entities are shown on the context diagram as well as a major data flow to and from them.
2. The diagram does not contain any data storage.
3. The single process in the context-level diagram, representing the entire system, can be expanded to include the major processes of the system in the next level diagram.

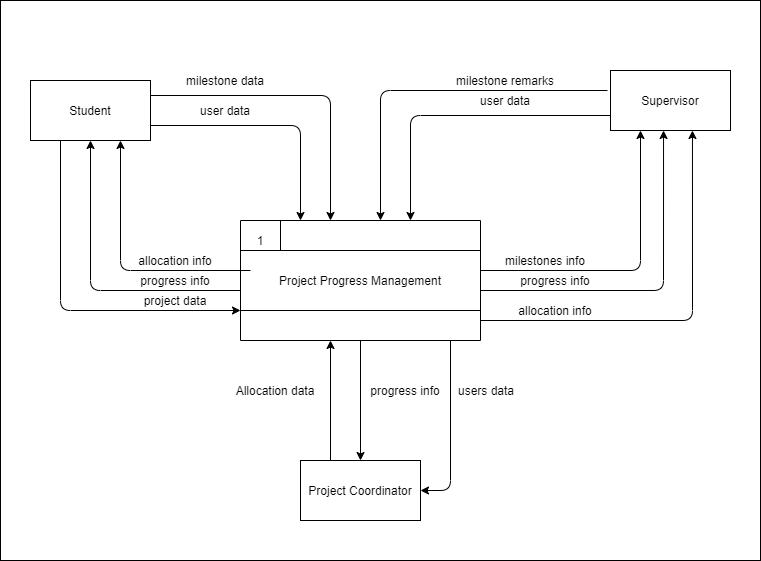


Figure 1: Level 0 DFD

#### Level 1 DFD

The process in the context diagram can be expanded further to represent details of the processing activities as shown in the level 1 DFD below.

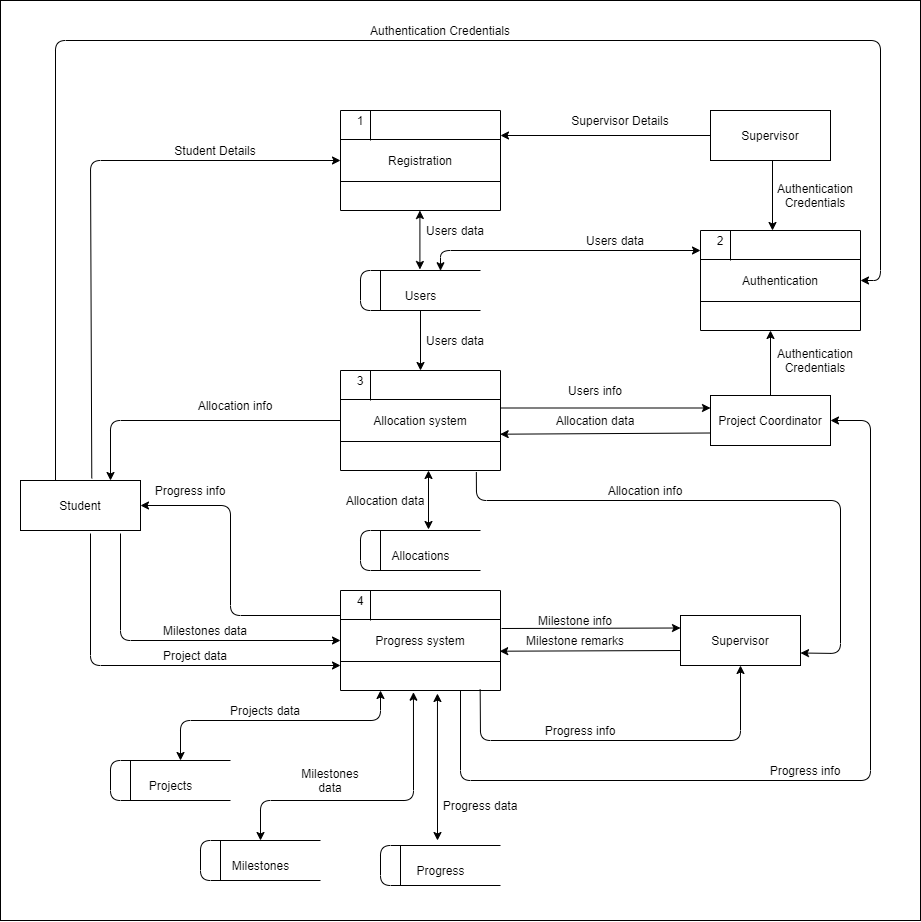


Figure 2: Level 1 DFD

#### Level 2 DFD

This level involves extracting a particular process that has a lot of data flow linking between a few external entities. The diagram below shows a level 2 DFD which extracts the progress system above into a separate diagram.

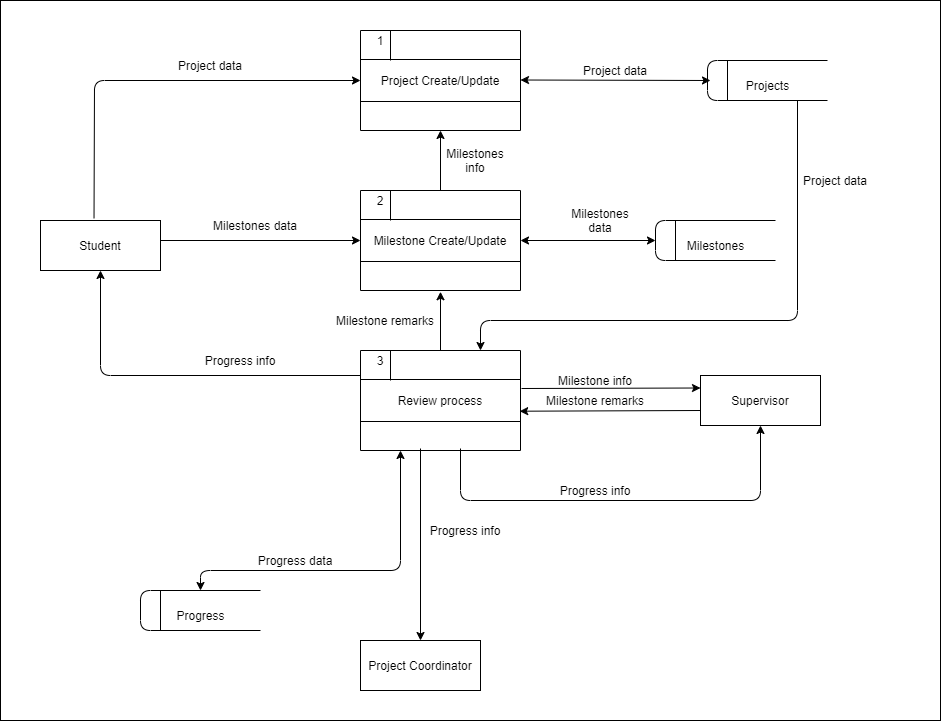


Figure 3: Level 2 DFD

### Use Case Diagram

A use case diagram models the functionality of a system using actors and use cases. Use cases are a set of functions that the system needs to perform. A use case diagram is valuable for visualizing the functional requirements of a software that will translate into design and development priorities. They help to identify any internal or external factors that may influence the system. The following is a use case diagram of SPPMS.

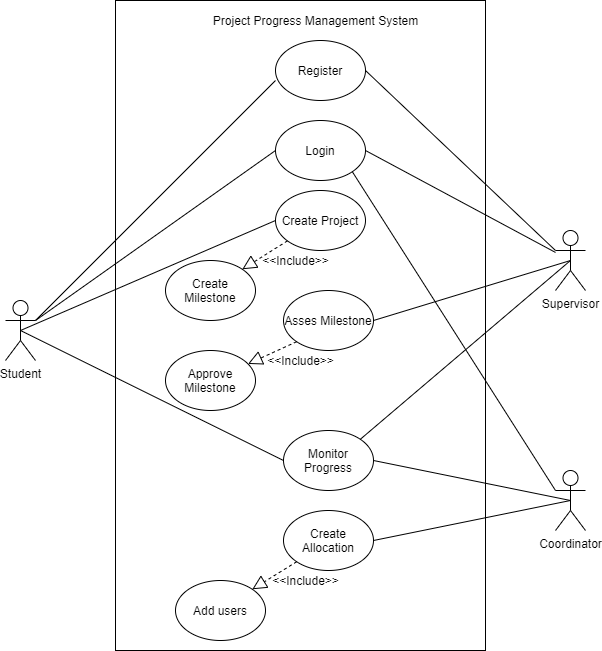


Figure 4: Use case diagram

### Class Diagram

A class diagram describes the types of objects in the software system and the different types of relationships that exist among them. It shows classes, their attributes, operations (or methods), and the relationships among objects.

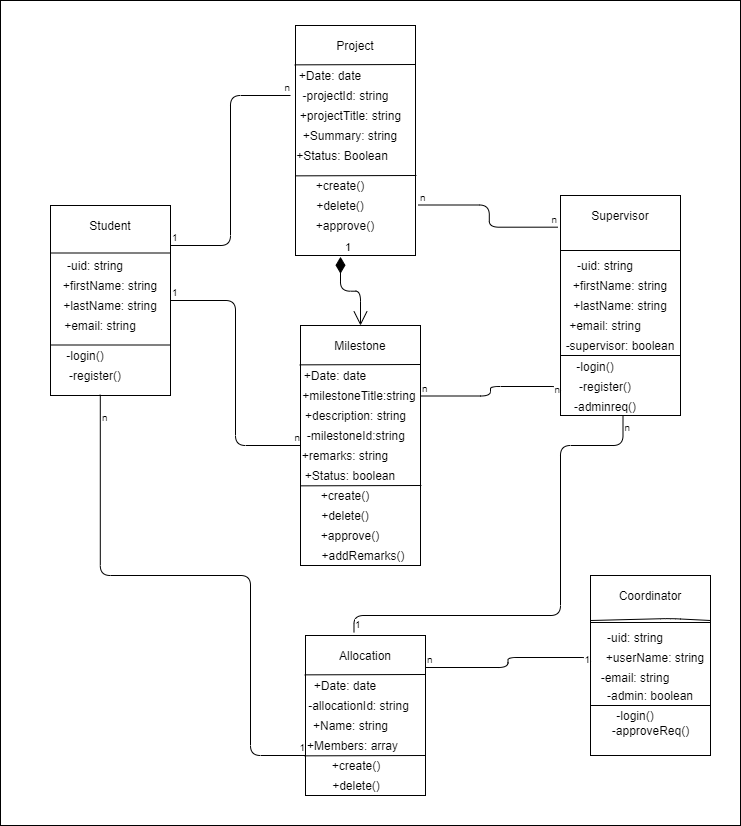


Figure 5: Class Diagram

### Flowcharts

A flowchart is a graphical representation of steps. It is used for representing algorithms and programming logic. It can play an extremely important role in displaying information and assisting reasoning. It helps us visualize the complex processes of a system.

#### Sign Up and log in

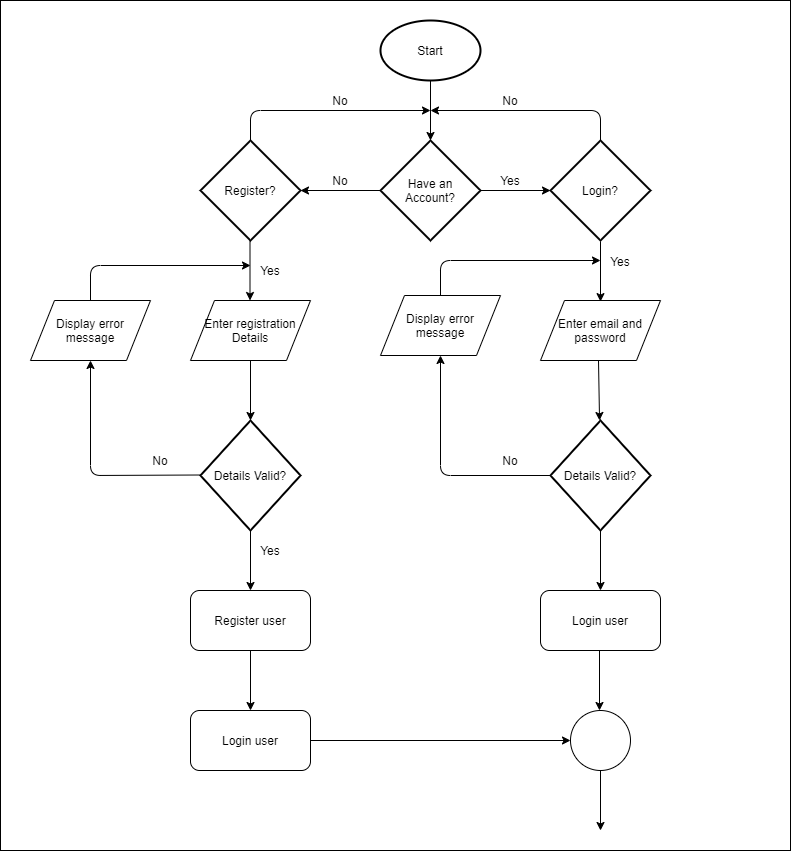


Figure 6: Sign Up & Sign In flowchart

#### Student’s Flowchart

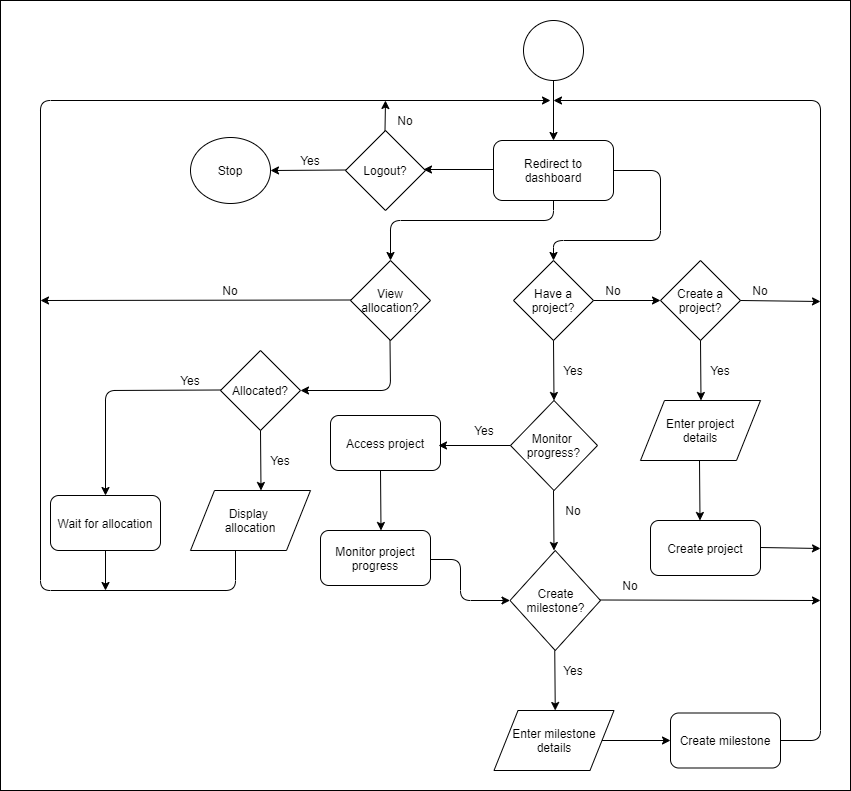


Figure 7: Students’ flowchart

#### *Supervisor’s Flowchart*

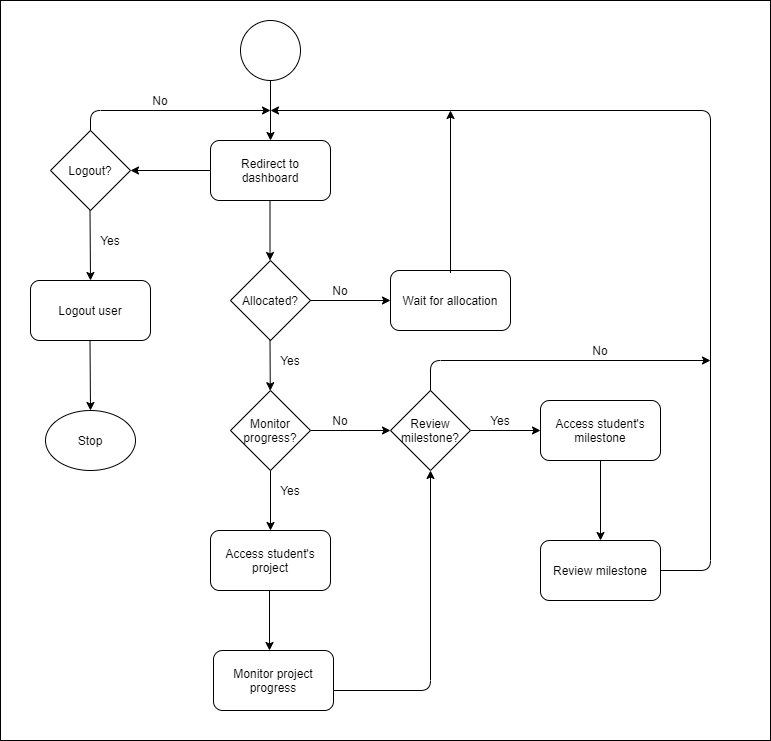


Figure 8: Supervisors’ flowchart

#### *Coordinator’s dashboard*

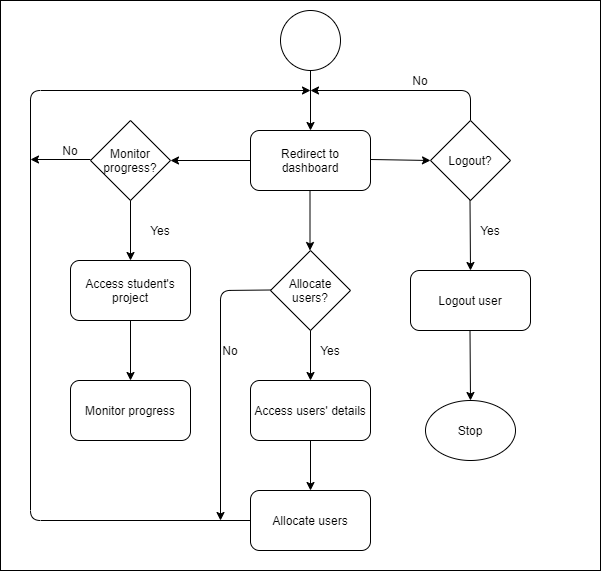


Figure 9: Project coordinators’ flowchart

## CHAPTER FIVE: SYSTEM DESIGN

### Introduction

System design is the process of defining the elements of a system such as the user interfaces and modules in order to meet the requirements of a system. This section focuses on the solution domain. It tries to answer, “How will the system be implemented?”

### Wireframes

A wireframe is a layout of a user interface that demonstrates what interface elements will exist on key pages. It is a critical part of the interaction design process. The aim of a wireframe is to provide a visual understanding of a page early in a project to get project team approval before the creative phase gets underway.

#### Sign Up and log in

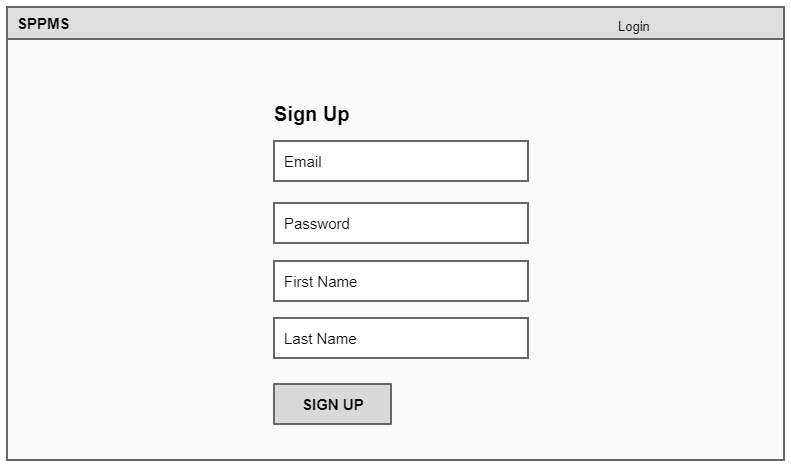


Figure 10: Sign Up wireframe

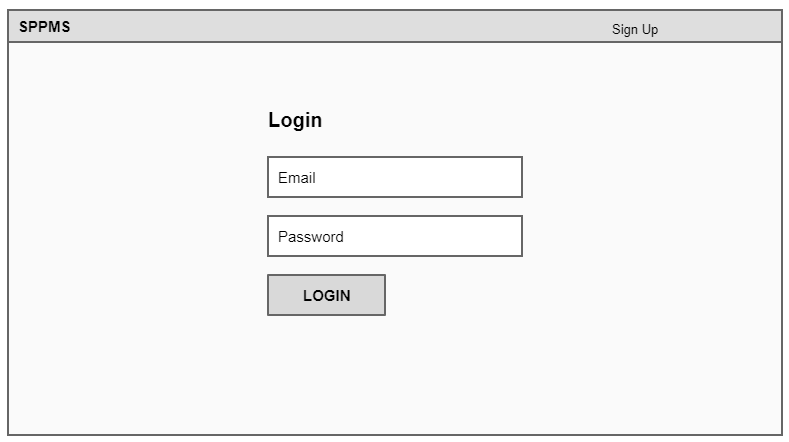


Figure 11: Login wireframe

#### Student dashboard

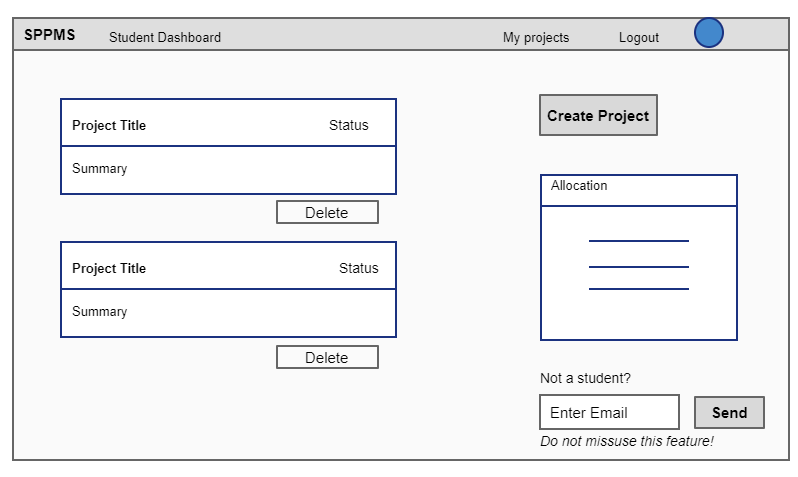


Figure 12: Students’ dashboard wireframe (a)

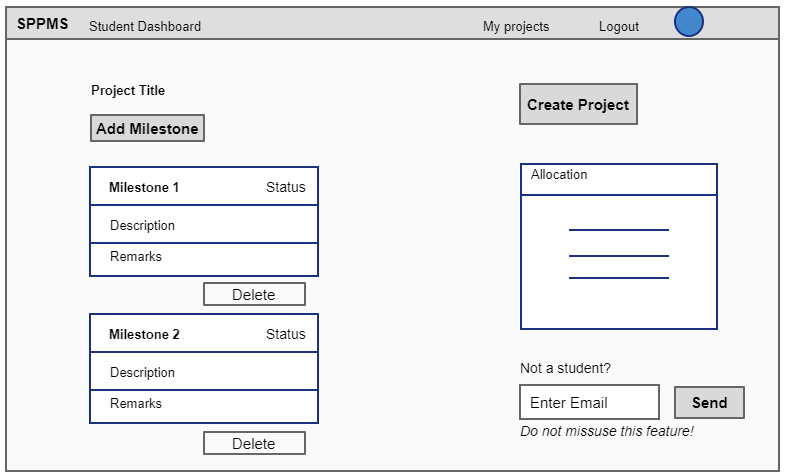


Figure 13: Students’ dashboard wireframe (b)

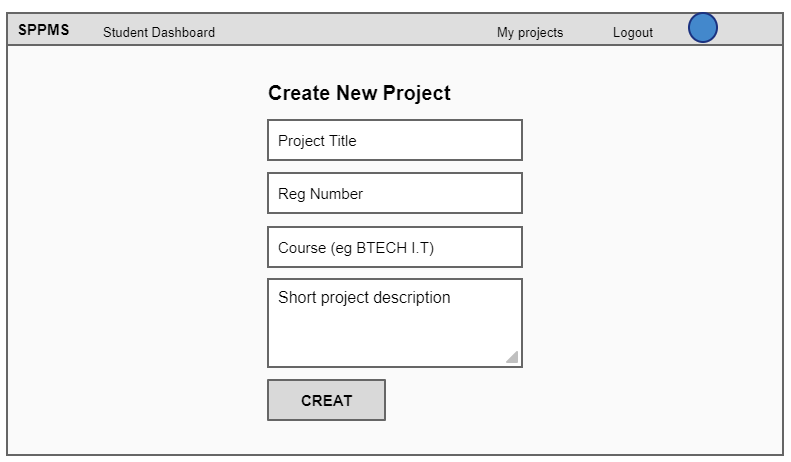


Figure 14: Add project wireframe

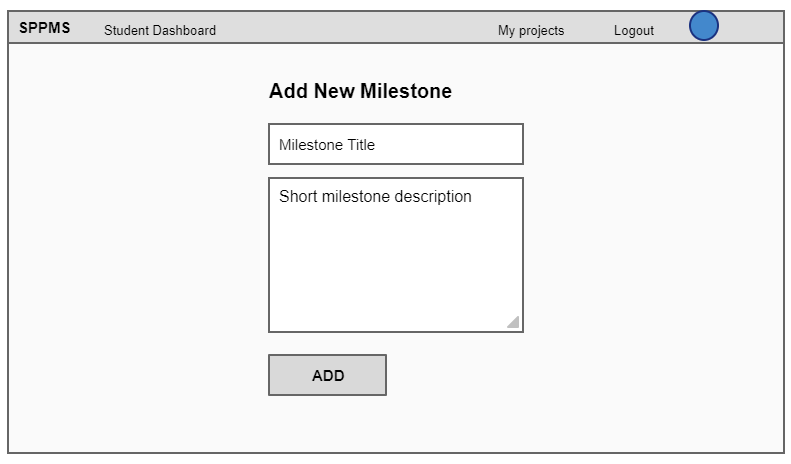


Figure 15: Add milestone wireframe

#### Supervisor dashboard

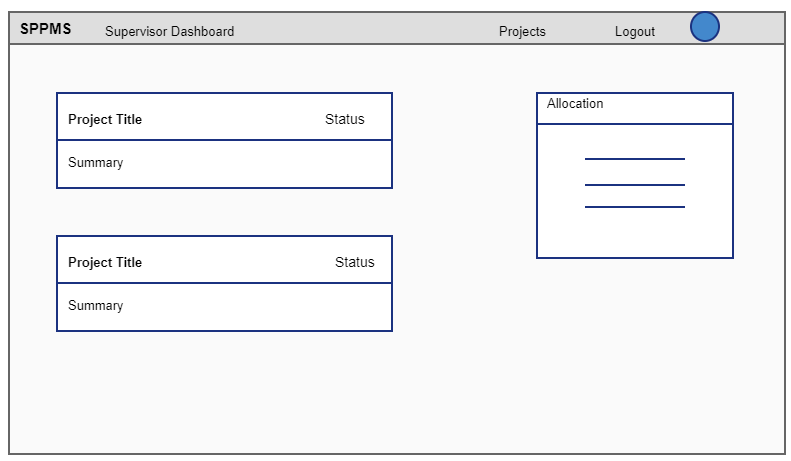


Figure 16: Supervisors’ dashboard wireframe

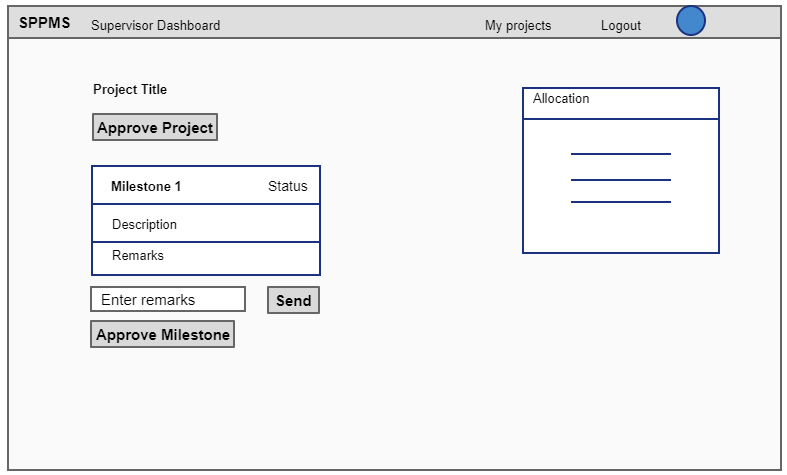


Figure 17: Approving milestones wireframe

#### Project co-ordinator dashboard

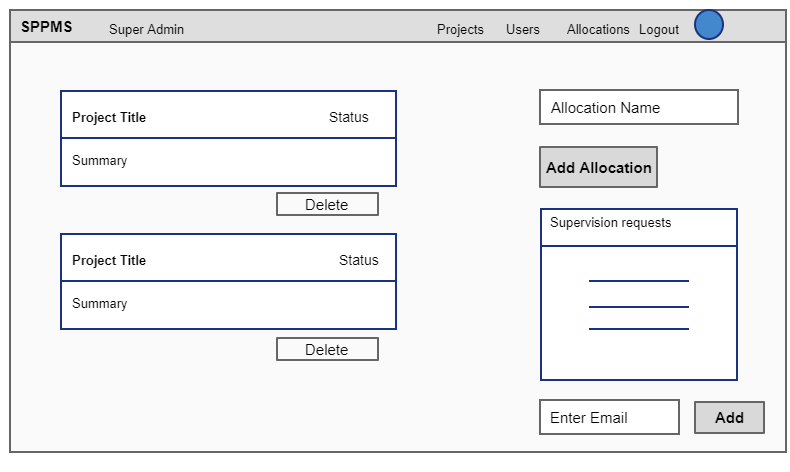


Figure 18: Project coordinator’s dashboard wireframe (a)

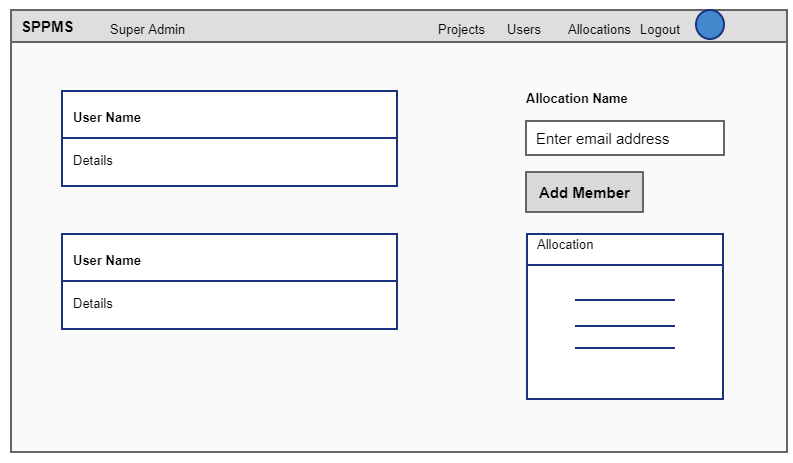


Figure 19: Project coordinator’s dashboard (b)

### Database schema

A database schema describes the structure of a database and is an organization of data that acts as a blueprint of how the database is constructed. It explains in detail the attributes of each entity in the database and the logical relationship that exist between these entities.

#### Conceptual database design

The main aim of conceptual design is to show the entities of the system and the relationship between them. It does not show all the details of the actual database structure. It is the simplest model among all.

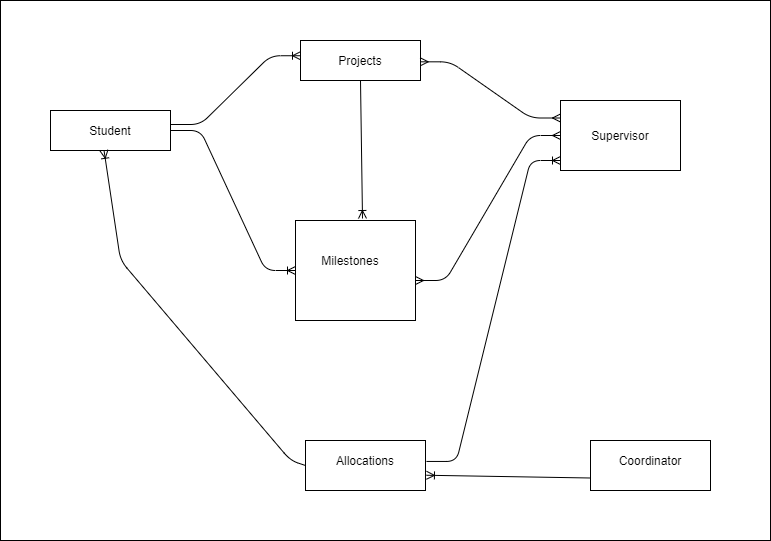


Figure 20: Conceptual database design

#### Logical database design

The logical design explains the database structure in a detailed way without regard to how the data will be physically designed in a database. All entities and relationships among them is show. All attributes of each entity are listed. In relation to the database, the primary key and foreign key are determined, and normalization happens in this level of database design. The following logical design shows the entities and their attributes only since SPPMS will be using a NoSQL database. There is no use of primary and foreign keys in NoSQL databases.

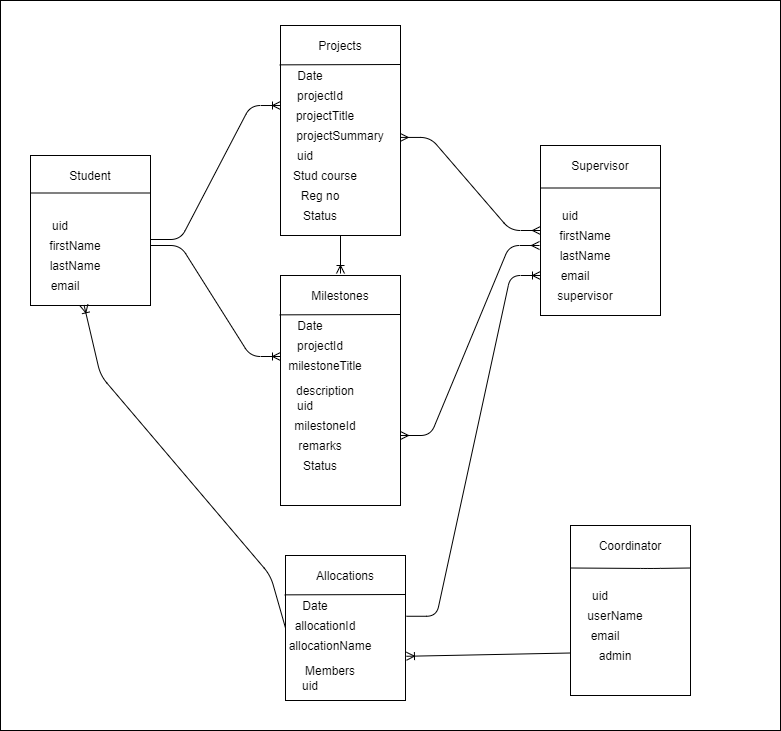


Figure 21: Logical database design

#### Physical database design

This level of database design shows how the database will be implemented. In a relational database, all tables, column names, column data types, primary key and foreign keys are included. In a NoSQL database, there are no tables, columns, primary keys and foreign keys. Collections, sub-collections, documents and subdocuments are used instead as shown in the following diagrams.

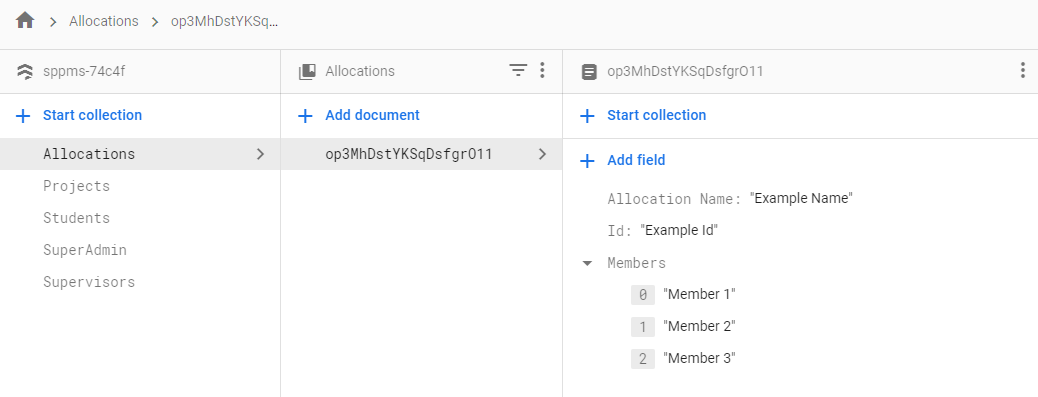


Figure 22: Allocations’ schema

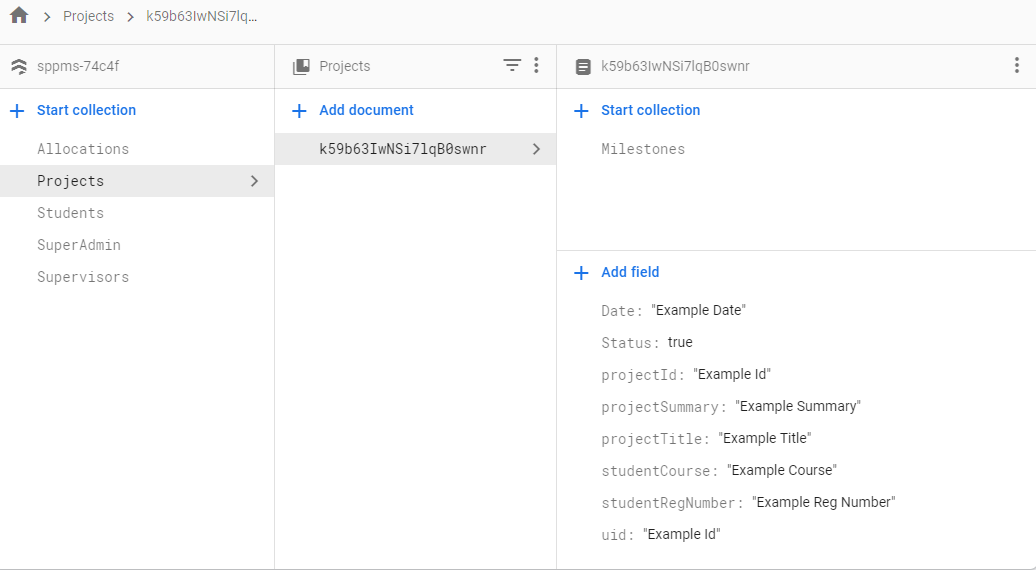


Figure 23: Projects’ schema

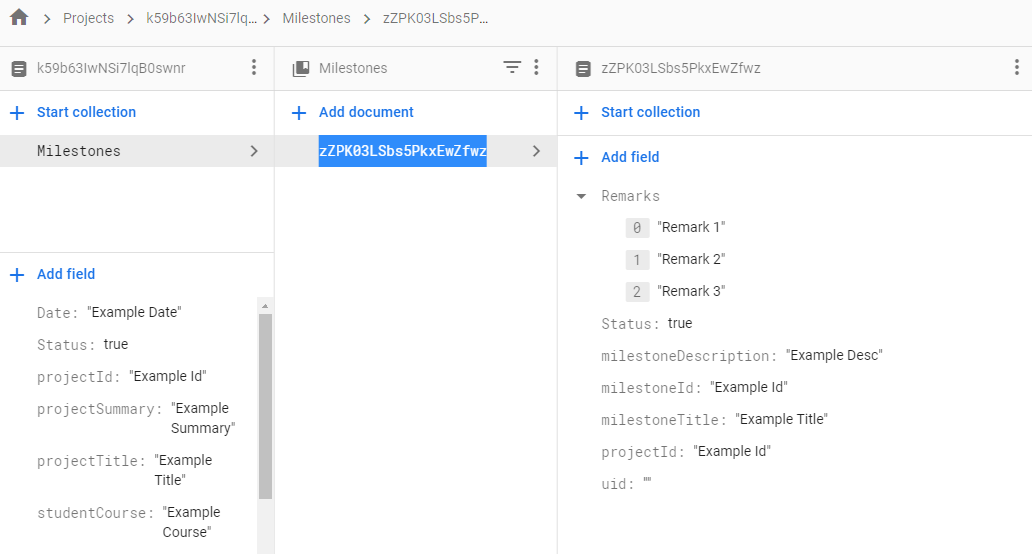


Figure 24: Milestones’ schema

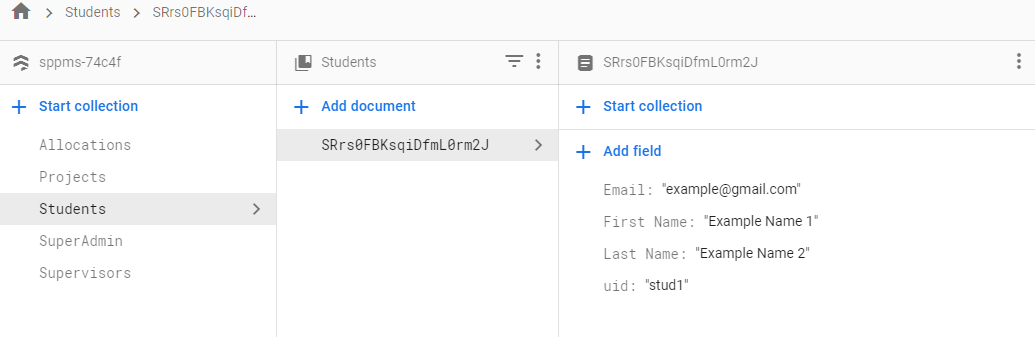


Figure 25: Students’ schema

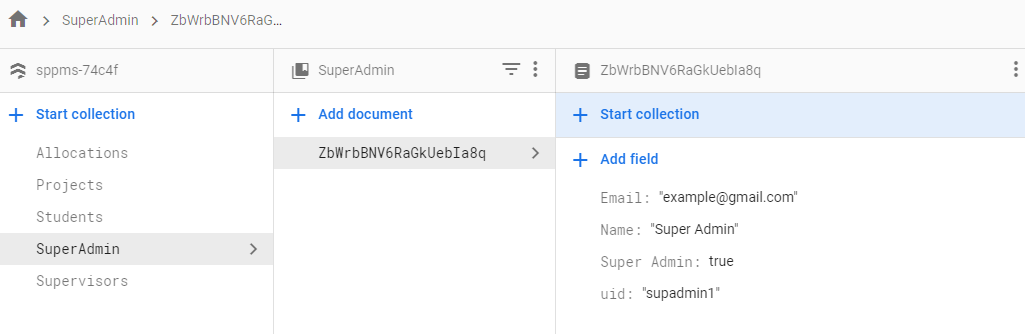


Figure 26: Project coordinator’s schema

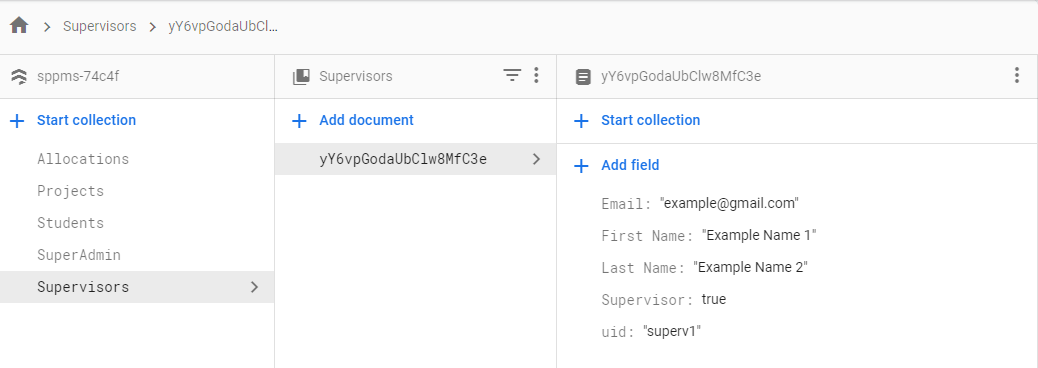


Figure 27: Supervisors’ schema