College of Natural and Computational Sciences (CNCS)

School of Information Science (SIS)

Student Project Tracking System for SIS

Name	ID
1. Andualem Sebsbe	UGR/7326/12
2. Gelila Tesfaye Gebru	UGR/0903/12
3. Hanan Hussien	UGR/1778/12
4. Huluager Alehegn	UGR/1112/12
5. Kasanesh Ayalew	UGR/8580/12

Information Systems Industrial Project II[INSY4122]

Adviser: Ms. Dagmawit Mohammed

Adviser name	**Date**	**Signature**
Examiner name	**Date**	**Signature**
Examiner name	**Date**	**Signature**

Table Content

List of Figures

List of Tables

Chapter 5
Object oriented design

5.0 Review of the first phase of the project

In the first chapter of this project, we have conducted an analysis of the organization's background and history, specific problem that the organization is facing in relation to the methods and ways of managing different students project in the school , and developed a clear statement of the problem to guide the project. To address the problem, we have defined specific objectives that our system aims to achieve, including measurable outcomes and time lines. We have also established the scope of the project and a feasibility study to assess the viability of the proposed system, considering technical, economic,operational and economic factors. Generally we have addressed the foundation for the new system and provided a clear roadmap for moving forward.

In the second chapter, we have covered the operations of the business areas and its current issues, and the functional and non-functional requirements of the proposed system.

In the third chapter we have developed the user interface (UI) of the new system, use case modeling to identify the various actors, use cases, and scenarios that will be involved in the system, and UML diagrams such as analysis class diagrams to model the system entities and relationships, sequence diagrams to model the system's behavior and interactions between different objects and classes This has helped us to understand the system requirements from the user's perspective and identify the key functionalities that the system needs to provide.

Functional Requirements for Student Project Tracking System:

1. User Authentication & Registration: The system should support authentication by means of passwords allowing the users to have access to personalized areas once identified successfully those users can also register with information like an username name, email address, course number etc..

2. Submit Title: Students should be able to submit their project titles through the system. The system should validate the input and ensure that the title is not already taken.
3. Project Tracking & Management: Ability to track existing projects assigned to students and faculties at any given point of time; ability to create a project plan outline; ability to create task plans including deliverables & deadlines associated with each task; ability to set milestones & associated goals; ability provide timely notifications on deadlines etc… .
4. The system should allow users with appropriate access permission mark a project has been completed by marking it with a check or other symbol.
5. It should provide feature for past project data search.
6. It should enable lecturers/supervisors assign tasks to students/researchers by due dates.
7. The system should generate project records as reports such as a list of projects running late or according to certain criteria like discipline wise etc.
8. It should provide feature for past project data search.
9.. The system should be able to store the project title, the deadline for completion, tasks assigned to determinate people, researcher's name and in-charge staff.
10. Ability to report on task completion status in real-time.
11. Automated notifications for students about upcoming deadlines.

5.1. Overview

This section provides the design of the new system. Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. The system requirement document is coming from the previous phase and develops the specific technical details required for the system such as the architecture of the system, system decomposition, deployment and component decomposition.

5.2. Systems architecture design

The proposed system is expected to replace the existing manual system by an automated system in all facets. The architecture of a system explains how distributed

systems are actually organized by considering where the software components are placed. It is made by keeping business logic and needs in mind.

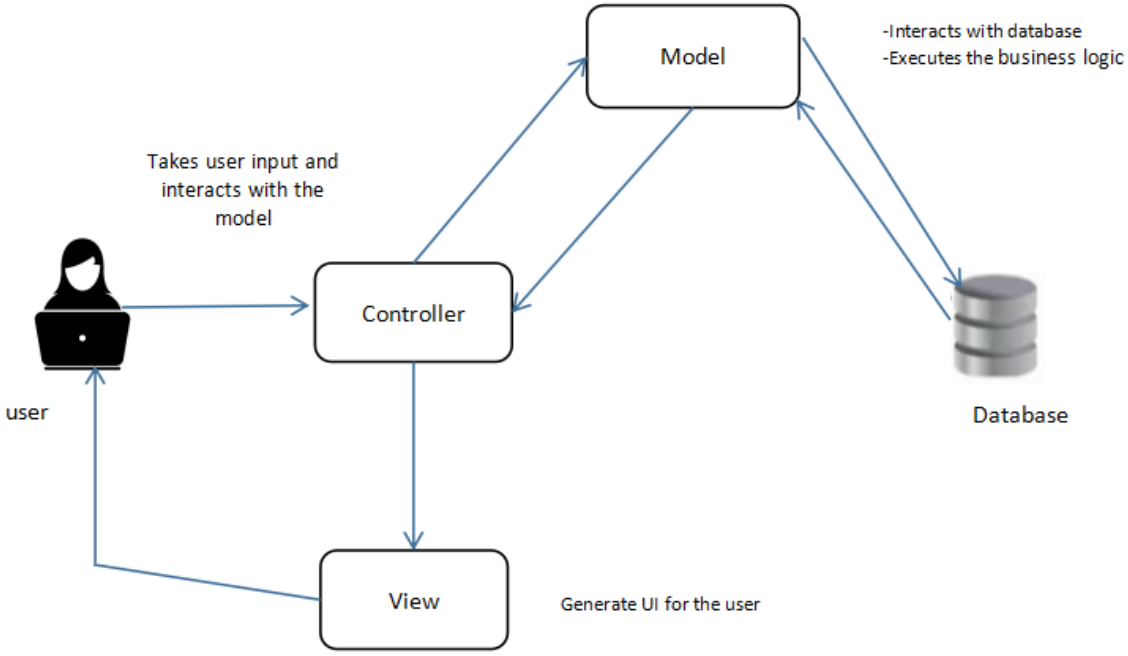


Figure 1 System Architecture

Our system is primarily focused on rendering dynamic views and handling user input and the technology and the framework that we used to develop the system is Next.js and Node.js as which allows for a separation of concerns between the various components of the system. With this regard MVC (Model-View-Controller) architecture is well-suited to our system, (i.e the Student Project Tracking System) because it separates the application logic into three distinct components: model, view, and controller. This separation of concerns makes it easier to maintain, extend, and test the system.

The model component is used to represent the data and business logic of the system. This include information such as

- ✓ Project
- ✓ User account
- ✓ instructor
- ✓ Student
- ✓ coordinator

The view component represents the user interface, which allows students, instructors and coordinators to interact with the system. This would include forms for submitting project titles, status updates, view progress, and other relevant information.

The controller component handles the flow of data between the model and view components, as well as processing user input and updating the model accordingly. Compared to other system architectures, such as a monolithic architecture, in which all the application logic is contained in a single codebase, making it difficult to modify or scale the system, MVC architecture, allows for a more modular approach, making it easier to add new features or functionality to the system as needed. Additionally, the separation of concerns makes it easier to test each component independently, reducing the likelihood of bugs or errors.

5.3 class diagram modelling

5.3.1 Class diagram

Student project tracking system describes the structure of a student project tracking system classes, their attributes, operations (or methods), and the relationships among objects. The main classes of the system are:-

- ✓ Student
- ✓ Course project
- ✓ Final year project
- ✓ Evaluation
- ✓ Calendar
- ✓ Instructor
- ✓ Coordinator
- ✓ User account
- ✓ Adviser and examiner

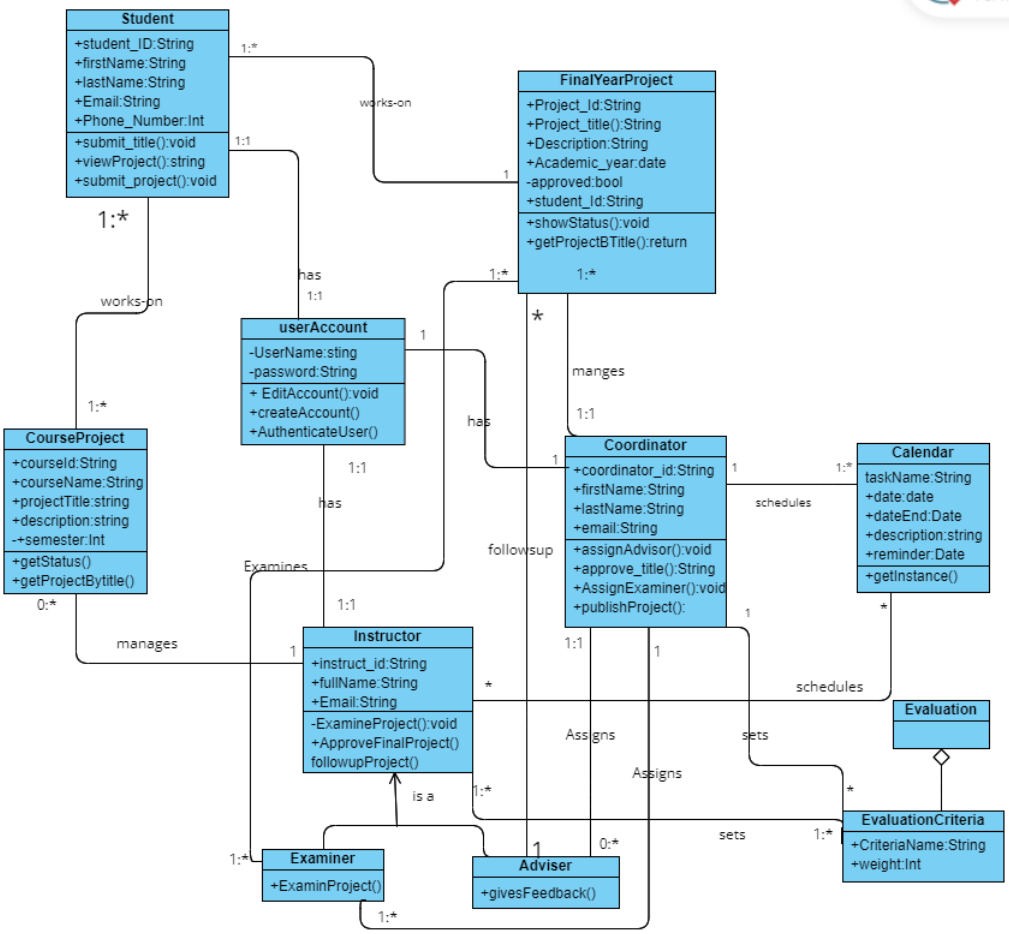
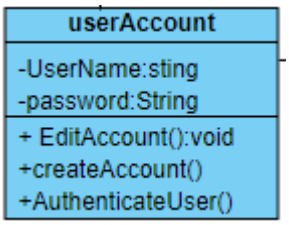


Figure 2. Class Diagram

5.3.2 Class Descriptions

1. **User Account:** This class is used to get the user information from the database and is also used for authenticating the users.



Attributes

- **Username**

- ✓ Description: The username is a unique identifier that is assigned to each user account. It is used to log in to the system and to identify the user in the system.
    - ✓ Data type: string(30)
    - ✓ Visibility: public

- **Password**
    - ✓ Description: it is a secret code that is used to authenticate the user and to protect the user's account from unauthorized access.
    - ✓ Data type: string
    - ✓ Visibility: private
- ✓ **User role**
- ✓ The user role determines the level of access and permissions that the user has within the system, i.e. the coordinator have its own access to different features and functions of the system that cannot be accessed by the student. Data type is string .
    - ✓ Data type: string
    - ✓ Visibility: public

Methods of the user account class

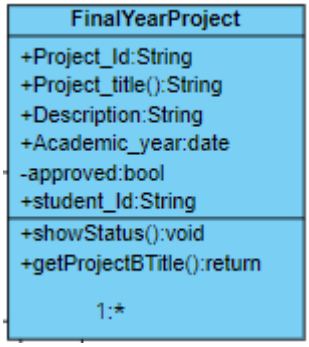
- **AuthenticateUser()**
- ✓ Description: this method is used to authenticate a particular user who has provided the login credentials and wishes to login in the system. It checks the credentials and roles from the database.
    - ✓ Visibility: public
    - ✓ Return type: void

Edit account()

- ✓ Description**:** this method is used to edit account such us password.
    - ✓ Return type: void
    - ✓ Visibility: public
- **Create accoun**t()

- ✓ Description: this method will save the details provided in the database. The user login will be functional immediately after the execution of this method.
- ✓ Visibility: public
- ✓ Return type: void

2. Final year /industrial Project class: This class is used to process all information regarding the industrial projects.



Attributes

- project Id
  - ✓ Description: this attribute uniquely identifies the project.
  - ✓ Data type: string
  - ✓ Visibility: public
- project title
  - ✓ Description: this attribute is used to search and retrieve projects.
  - ✓ Data type: string
  - ✓ Visibility: public
- Project_Description
  - ✓ Description: this attribute is used to show the overview of the project
  - ✓ Data type: string
  - ✓ Visibility: public
- academic year
  - ✓ Description: is used to identify the year in which the project is done.
  - ✓ Data type: string
  - ✓ Visibility: public
- Student_id

- ✓ Description: this attribute references the student id in the student collection.
- ✓ Data type: reference array
- ✓ Visibility: public

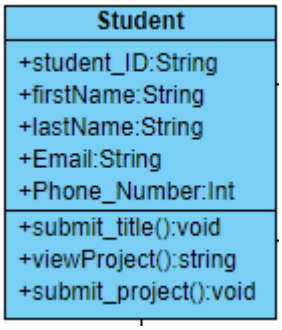
Methods

GetProjectByTitle()

- ✓ Description: This role populates the details of a selected project.
- ✓ Return type: list<Projects>
- ✓ Visibility: public(+)

- **GetStatus()**
  - ✓ Description: this method used to see the active projects.
  - ✓ Return type: void
  - ✓ Visibility: public

3. **Student: -** This class used to access information about a student who is enrolled in the program. They have attributes such as student ID, name, and email address phone number.



Attributes

- student_id
  - ✓ Uniquely identify student.
  - ✓ Visibility: public
  - ✓ Type: string(12)
- first name and last name
  - ✓ Type: string(30)
  - ✓ Visibility: public
- Email
  - ✓ Type: string(30)

- ✓ Visibility: public

Methods

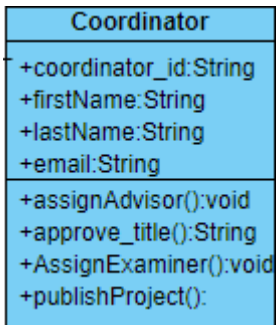
- Submit_title()

Description: this method allows the students to submit project title to the system.

- ✓ Return type: void
- ✓ Visibility: public

- ViewProject()
  - ✓ Description: this method used to allow the user to view project details
  - ✓ Return type: List<Project>
  - ✓ Visibility: public
- Submitprojects()
  - ✓ Description: this method is used to submit either midterm or final project to the system.
  - ✓ Visibility: public
  - ✓ Return type: void

4. **Coordinator**: this is mainly to access the final year project coordinators information.



Attributes: -

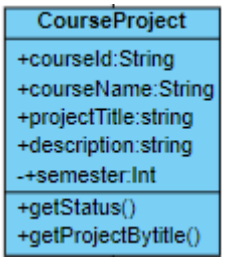
- coordinator id
  - ✓ Description: this attribute used to uniquely identify a record.
  - ✓ Data type: string
  - ✓ Visibility: public
- first name
  - ✓ Data type: string(20)
  - ✓ Visibility: public

- last name
  - ✓ Data type: string(20)
  - ✓ Visibility: public
- Email
  - ✓ Data type: string(30)
  - ✓ Visibility: public

Methods

- Assign adviser ():- this method is used to assign adviser for a final year (industrial - project) from the instructors.
- Approve title (): this method allows the coordinator to approve a project once it has been submitted by the student.
- Assign examiner: helps the coordinator to select instructors and assign as an adviser for a project.
- Publish project: this method is used to publish the final project.

5. **Course Project class:** this class is used to track different course projects.



Attributes:

- course id
  - ✓ Type: string
  - ✓ Visibility: public
- course name
  - ✓ Type:string
  - ✓ Visibility: public
- Semester
  - ✓ Type: integer
  - ✓ Visibility: public

Methods

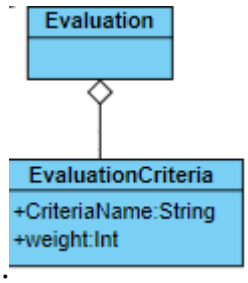
GetProjectByTitle()

- ✓ Description: This role populates the details of a selected project.
- ✓ Return type: list<Course projects>

- ✓ Visibility: public(+)

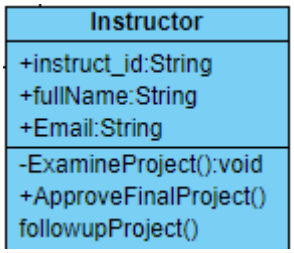
- **GetStatus()**
  - ✓ Description: this method used to see the active projects.
  - ✓ Return type: void
  - ✓ Visibility: public

6. **Evaluation** this class is a simple class



The evaluation class is an aggregate of evaluation criteria because it represents a collection of individual evaluation criteria that are combined to form a comprehensive evaluation of the software system.

7. **Instructor class**



Attributes

- **Instructor_id**
  - ✓ Description: uniquely identify each Instructor records.
  - ✓ Type: string
  - ✓ Visibility: public
- **Full_name**
  - ✓ Description: this attribute is used to store the name of the instructor.
  - ✓ Type: string
  - ✓ Visibility: public
- Email
  - ✓ Type: string
  - ✓ Visibility: public

Methods

Examine project ()

- ✓ Description: this method is used to evaluate the course project.
- ✓ Return type: void
- ✓ Visibility: public

Approve project ()

- ✓ Description: this method allows the instructor to approve course projects.
- ✓ Return type: string
- ✓ Visibility: public

Follow up projects ()

- ✓ Description: this method allows the instructor to manage and follow the progress of the project activities.
- ✓ Return type: void
- ✓ Visibility: public

8. **Adviser class**: this class is a child class inherited from the instructor class.

9. **Examiner class**: this class also a child class inherited from the instructor class.

10. **Calendar class**

Description: It provides a wide range of functionality for performing common date and time operations, and can be used in a variety of applications, such as scheduling systems, reminders, and time-tracking applications

Attributes:

- Task_name: this attribute used to show the task assigned in the given deadline.
  - ✓ Type: string
  - ✓ Visibility: public
- Start_date and end_date
  - ✓ Type: Date
  - ✓ Visibility: public
- Description: shows the description of the task.
- Remainder: to notify the deadline.
  - ✓ Data type: ate,
  - ✓ visibility: public

5.4. Relational persistent model

A relational persistent model is a way of organizing data in a software application using a relational database management system that provides persistent storage capabilities, allowing data to be stored and accessed efficiently and reliably over time. Hence we are using MongoDB which is a non-relational database management system that stores data in a document-oriented format, rather than table-based structure of relational databases. In MongoDB, data is stored as documents, which can contain nested arrays and key-value pairs, making it easy to represent complex data structures. Since MongoDB does not rely on the tabular structure of relational databases, we don't need to draw a relational persistent model, Instead, data can be stored in a more flexible and dynamic format..

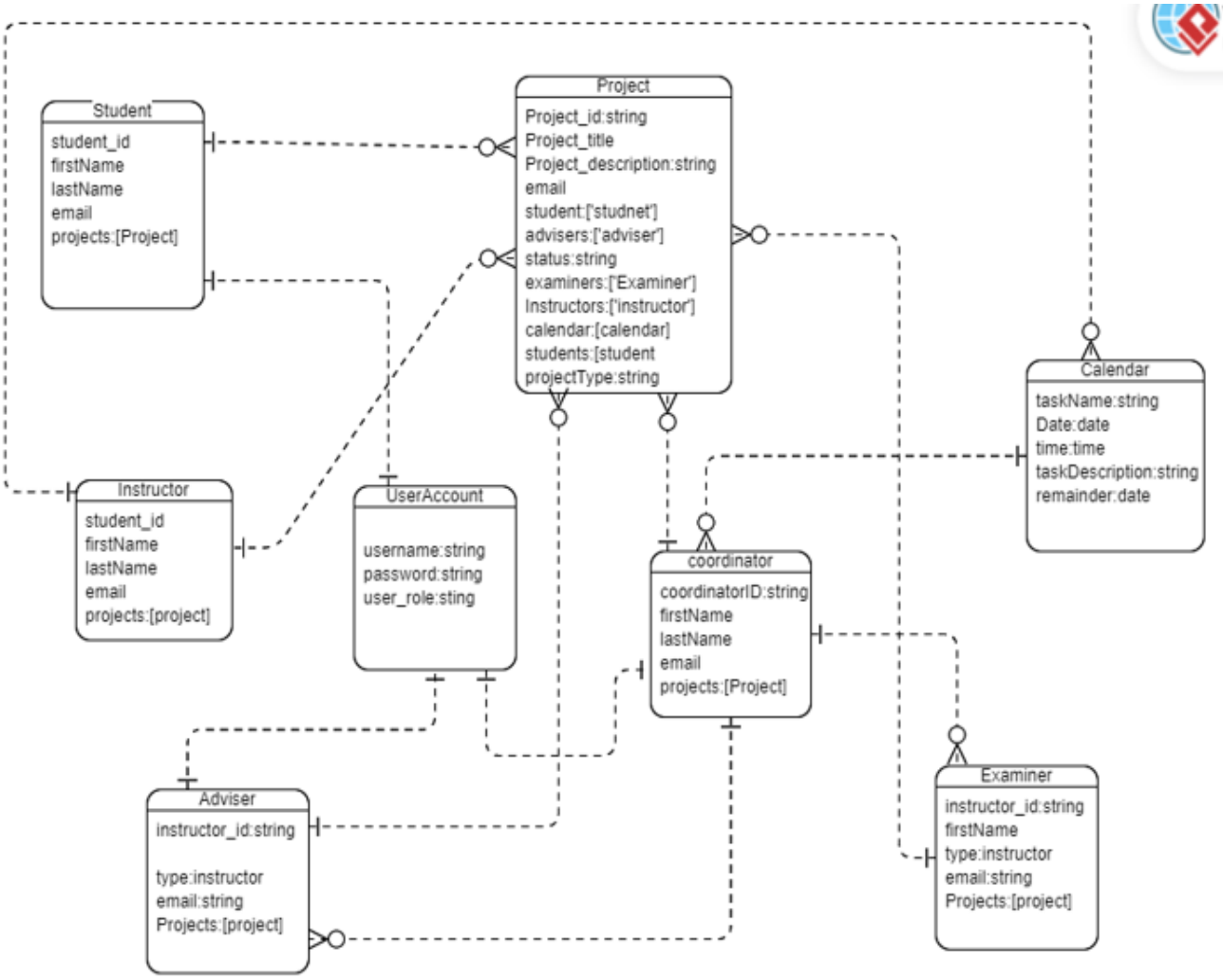


Figure 3. Non-relational persistent model

In this model, each entity (Coordinator, Student, Instructor, Adviser, Examiner, Project, and Calendar) is represented as a separate document or collection in our MongoDB database. The "projects" field in each entity is an array that contains references to the projects that the entity is associated with. The "calendar" field in the Project entity is also an array that contains references to the Calendar events associated with the project.

5.5 Component Diagram

A UML component diagram is a type of diagram in Unified Modeling language (UML) that shows the structure of a software system by representing the components and their dependencies. Components in a UML component diagram are software modules, libraries, executables, or other parts of a system that can be independently deployed and replaced.

A UML component diagram typically includes components, ports, interfaces, connectors, and relationships between them. The components are shown as rectangles with the name inside, and the ports are depicted as small squares at the edges of the components, which serve as communication endpoints for the components.

The relationships between components are depicted as connectors, which can be either simple lines or more complex shapes such as arrows or diamonds. UML component diagrams are helpful in understanding the architecture of complex systems and in designing new software systems.

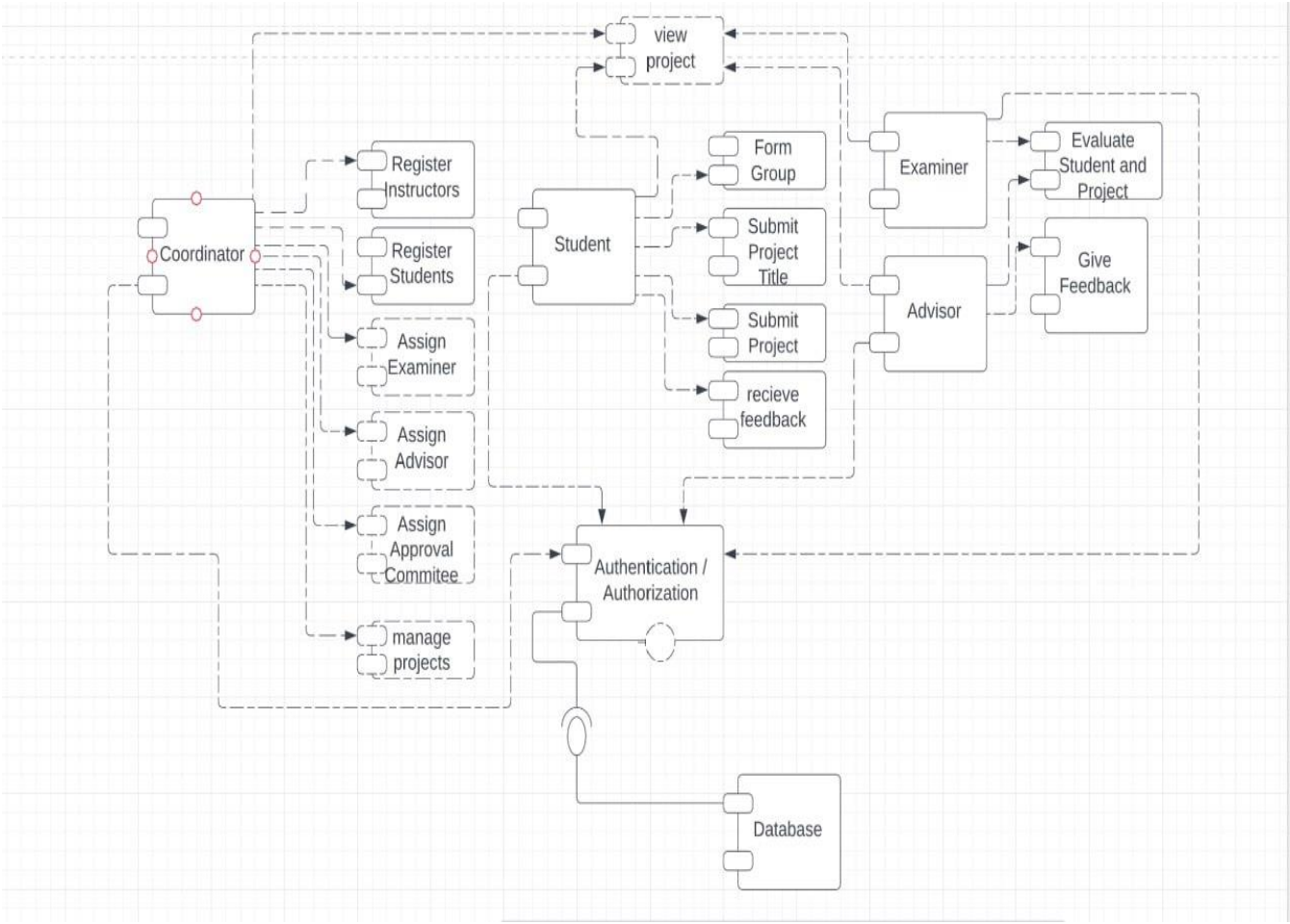


Figure 4. Component Diagram

5.6 Deployment diagram

The deployment diagram is a type of UML diagram that shows the physical deployment of software components on hardware nodes. It consists of nodes, which represent the hardware elements, and components, which represent the software elements. The nodes can be physical devices or virtual machines that are used to host the components. The components can be software applications, web services.
The deployment diagram for the student project tracking system shows the distribution of the processes using the system's physical architecture. It shows the relationships between software and hardware (physical architecture) that the project tracking process. This system includes three separate nodes as shown in the following diagram.

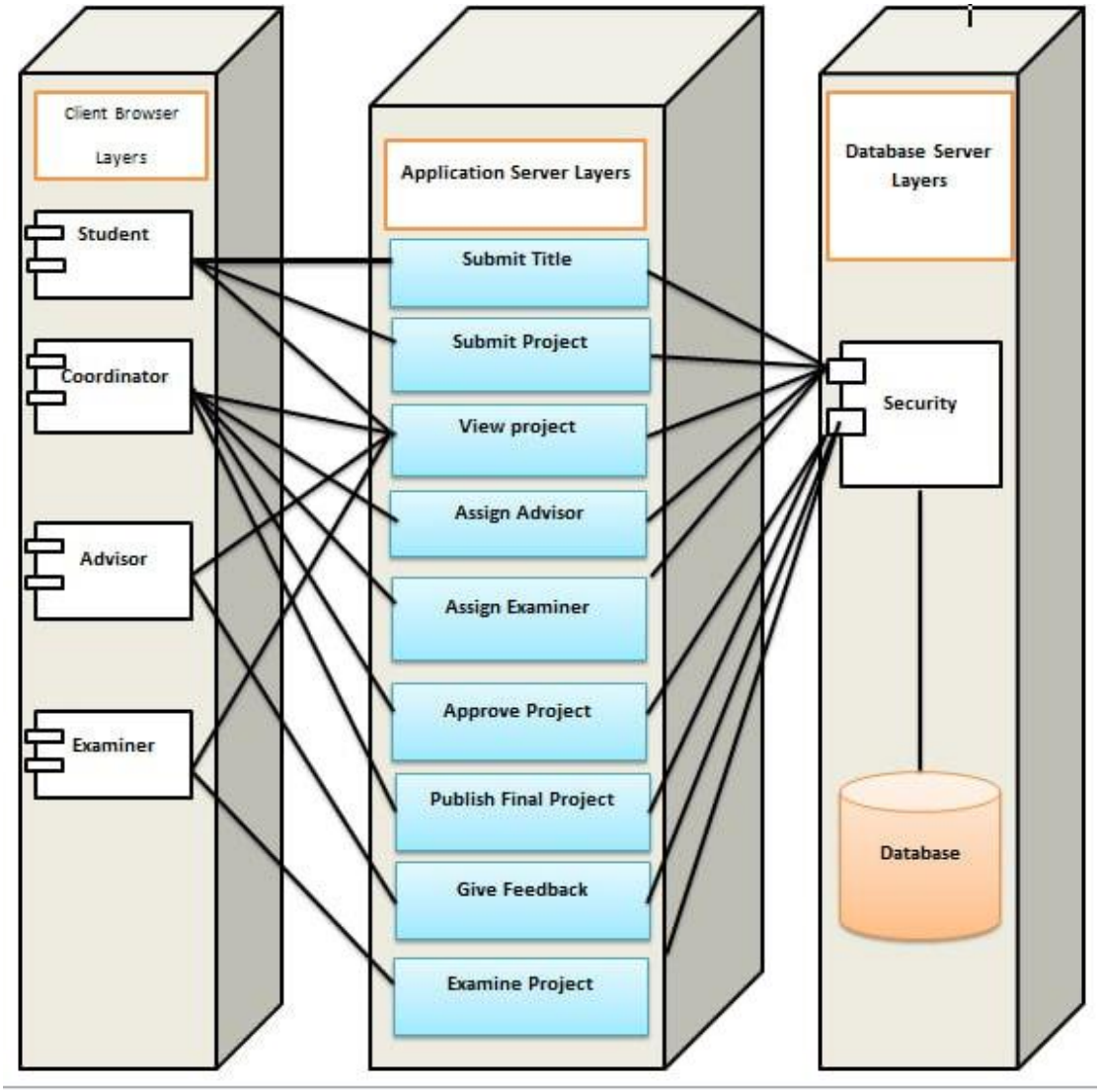


Figure 5. Deployment Diagram

5.7 User interface

5.7.1 User Interface Flow Diagram

User flow diagram is used primarily by product and UX teams to figure out the flow of a website or application after we have thought about the users experience and user needs. To best understand these needs and the experience we want our customers to have, it is important to map and visualize them. The following diagram shows user interface flow diagram.

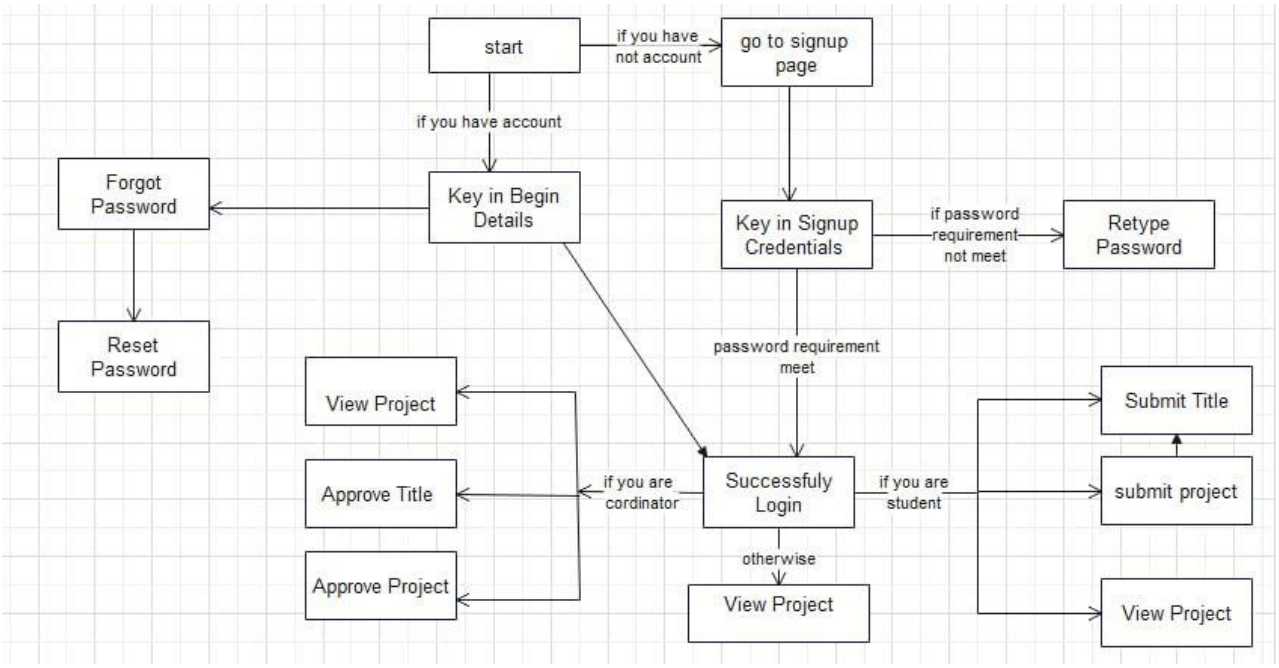


Figure 6. User Interface Flow Diagram

5.7.2 User interface design

Home Page

This page provide an overview of the system's features and navigation, and also display the highlight information of some active project for users.

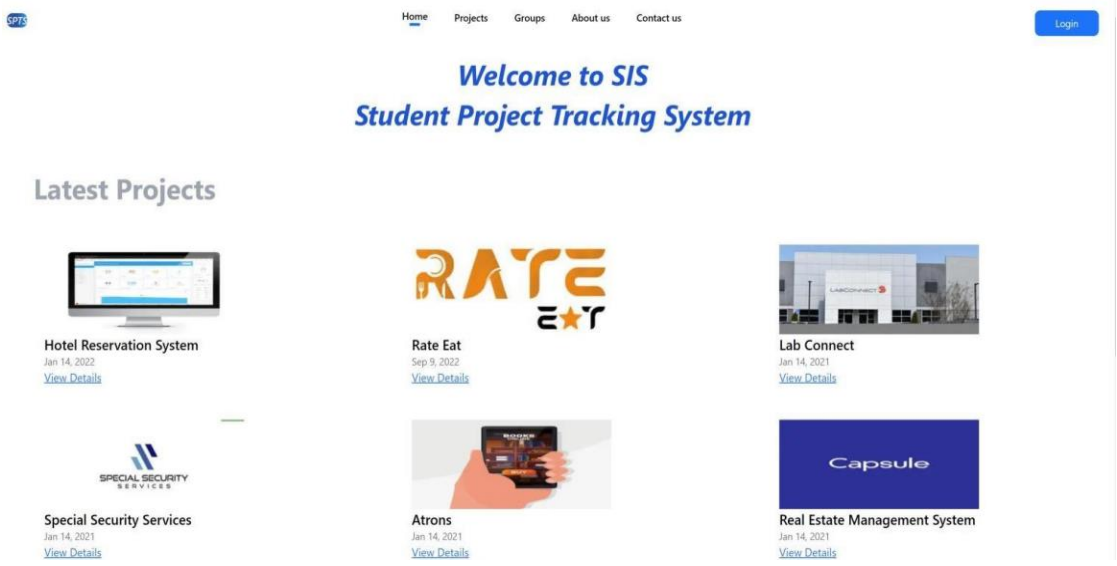


Figure 7. Home Page UI

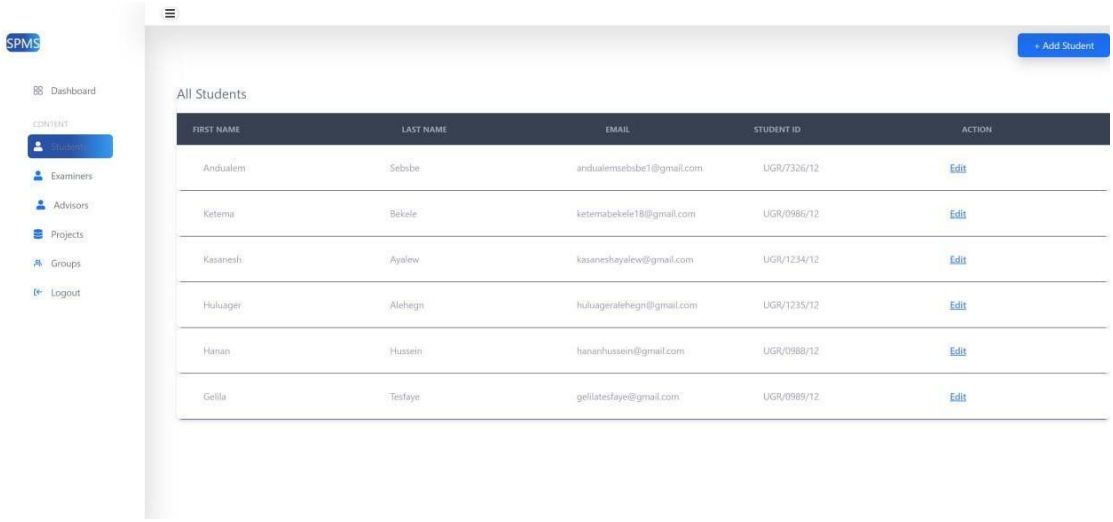


Figure 8. Coordinator Dashboard

Description: All registered student are viewed and modified with one screen.

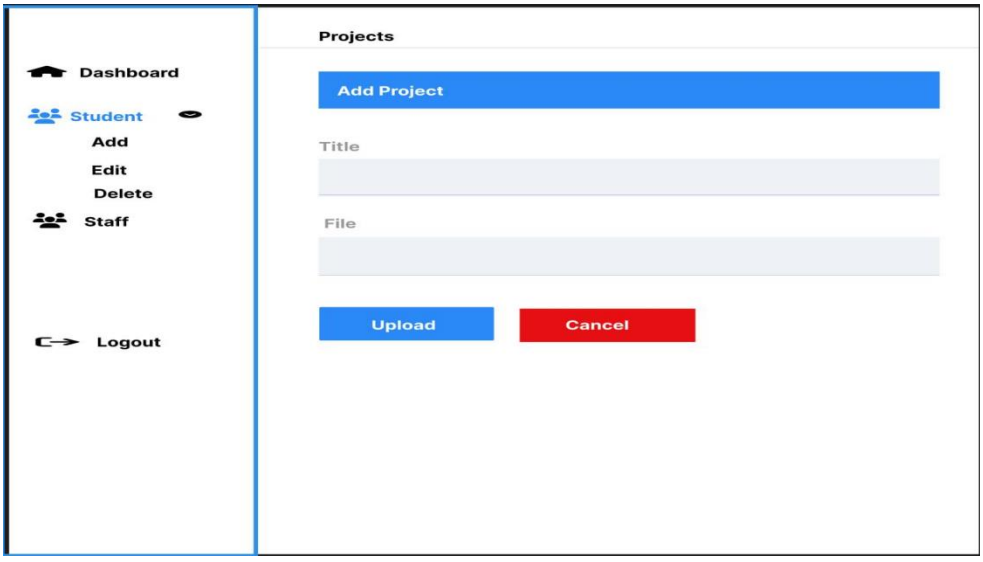


Figure 9. student dashboard

Description: this interface allows the student to upload project title with description.

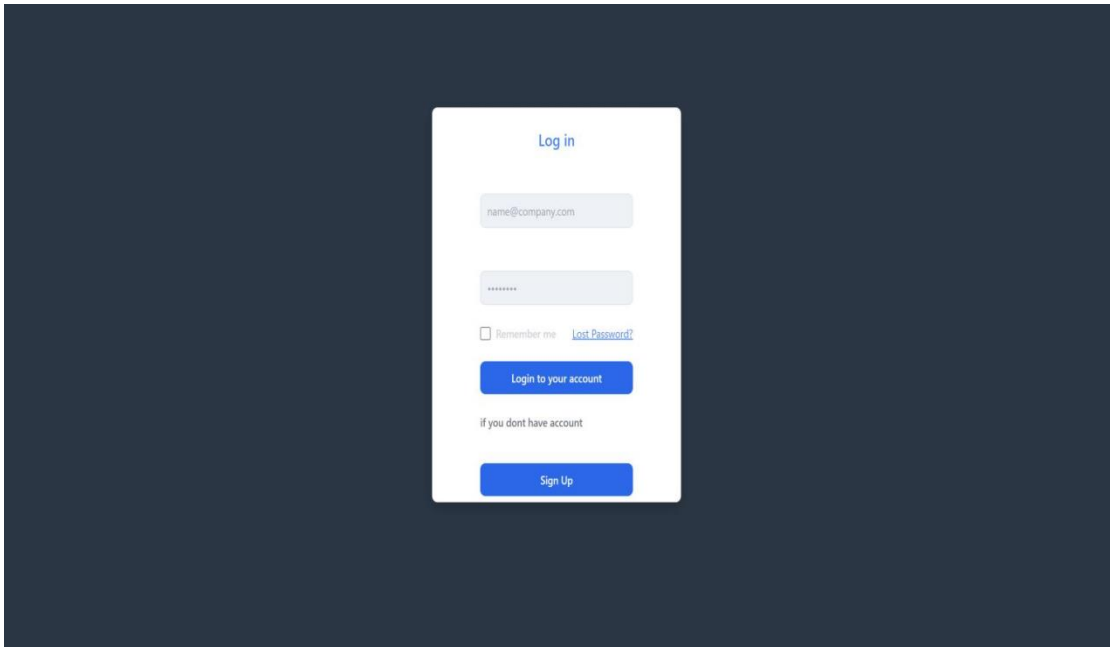


Figure 10. Login Page

Description: this screen shows the login page to authenticate the user and checks the credential.

chapter six

Object oriented implementation

6.1 Overview

System implementation is a process that converts the system requirements and design into program codes. This phase at times involve some modifications to previous design. The development environment has certain impact on the development of a system. System development consists of hardware and soft configurations. Using the suitable hardware and software is an important factor in determining the success of the project.

6.2 Testing and testing procedures

The primary purpose of testing is to ensure the resulting component of program as well as the program as a whole fulfills the requirements specification and to eliminate errors in the program, Thus, a systematic test procedure is required to ensure the system is tested thoroughly and completely.

The SPTS follows the classical strategy for testing software, beginning with unit or component testing and working towards integration and system testing as a whole.

6.2.1 unit testing

This test is conducted to check whether the internal logic is functioning properly and program inputs produce valid outputs that compare with the expected results. It is done after the completion of an individual unit. These tests are performing at component level and specific business process, application, and system configuration. In this testing module interface is tested to assure that information properly and correctly flows into and out of the module. This testing involves the testing of data truncation, the structure of the data, and whether the program correctly accepts the input data. The whole validation of the program is encountered in this testing.

Table 0-1 unit testing activities

Testing Activities	Yes	No	Remark
The first page appears on the very first call to a webPage.			
The system notifies the user if s/he tries to login with wrong username or password.			
Text fields and buttons aligned properly.			
The system forms display text fields in an order.			
Links for 'previous' and 'next' page works fine in the system.			
The system pages navigate according to a proper sequence.			
The button works for searching any content in the page			
The system fonts and background colors are smoothing to the user's eye's			

6.2.2 integration testing

Integration testing ensures that the system can integrate with other systems that it needs to interact with, such as student information systems, library management system, learning management systems, and project management tools.

6.2.3 system testing

When the unit testing and the integration testing have been completed, the whole system would be tested to ensure the software product does not fail. System testing can be broken down into two types namely security testing and performance testing. System testing is a critical part of the software development process, as it helps to ensure that the system is working correctly, meets the needs of its stakeholders, and is ready for release to users.

Our system typically involves managing the entire life cycle of a student project, from initial proposal and planning through to execution, monitoring, and evaluation. Here are the system testing activities that will be performed for our proposed system.

- **Functional testing**: Functional testing ensures that the system meets the functional requirements specified for the project. This includes testing features such as project title submission, project approval work flow, task assignment, progress tracking, and evaluation. We can designed different test cases to verify that the system can handle the full range of scenarios that students and faculty members might encounter during the project life cycle.

Table 0-2 Checklist For Functional Test

Testing Activity	Yes	No	Remark
All the linkage of webpages are working correctly and successful redirect to another page.			
All Forms are working as expected and if the user doesn't fill mandatory field in the form an error message is shown			
Login functionality is working as expected.			
Data manipulation is working like: Delete/Edit operations			
The system checks the existence of the project title in the database when students tries to submit project title.			
Is the system allows the user to search projects according to their need?			
Does the system allows the user to download the project?			

- **Security testing**

Security testing attempts to verify the protection mechanism built into the system. It attempts to protect the system from unauthorized users.

Table 0-3Checklist for security test

Testing Activities	Yes	No	remark
Does the system check access privileges and validated against authorized users.			
If functionality is not working? the system displays an error page, instead of displaying any application, server, or database information.			
Does the system Verify the important information like password, and display in encrypted format?			

- **Usability testing**: Usability testing is designed to ensure that the system is easy to use and intuitive for all users of the system. It is mainly used to analyze the potential area of improvement to the system. the This can include testing the user interface, navigation, and overall user experience.

Table 0-4 Checklist for usability test

Testing Activities	Yes	No	Remark
Is the interface of the system user friendly?			
Is the system attractive (regarding the font size and color combination) for use?			
Is the system easy to understand and to be used?			
Is the system allowing the system user to search documents effectively?			

6.3 deployment / installation process

The deployment process for a software application requires careful planning, testing, and monitoring to ensure that the application is stable, secure, and meets the requirements of end-users. The deployment process for a software application involves the steps and procedures for releasing the application into a production environment.The deployment process can vary depending on the complexity of the application, the deployment environment, and the requirements of the organization.

The deployment and installation process for a web based system can vary depending on the specific hosting environment and deployment strategy. Hence our system is developed with the nextJs there are different tools offer different features and pricing structure.in our case we will intend to use Vercel which is a cloud-based platform that specializes in hosting Next.js applications. It offers a seamless deployment process, automatic scaling, and a range of features for optimizing performance and security.

Chapter 7

Conclusion and recommendation

7.1 Conclusion

This paper shows the initial step for developing a web based Student Project Tracking System for the school of information science(SIS). It provides an automated form of submitting, approving, retrieving and storing student projects. Moving from a current manual work system to the automated system helps the SIS to reduce costs, increase project security and minimizing redundant project works. To design and format the web system different kinds of tools are used like: tailwind instead of CSS, NextJs, NodeJs and some other tools. The database was created using MongoDB database management system. Generally, the developed Student Project Tracking System will benefit the school Employee's, and the students. It reduces the time spent in managing projects and ensures that the system students will find important information about different projects timely.

7.2 Recommendations

….……..will cont……..