CHAPTER THREE

SYSTEM DESIGN METHODOLOGY

3.0 Introduction

This chapter covers the detailed explanation of the methodology that is used to develop this project.

This project is aimed at developing a progressive web application, which manages the activity of “Student Project Management”.

The term "application model" in Web technology usually refers to the structure of the system software layer. Monoliths architecture, Two-tier architecture, N-tier architecture, Micro services architecture, are some of the commonly used application model.

3.1 Application Architecture

The architecture used for this system is clean architecture**.**

The goal of software architecture is to minimize the human resources required to build and maintain the required system. ― Robert C. Martin.

Clean Architecture is the system architecture guideline proposed by Robert C. Martin also known as Uncle Bob.

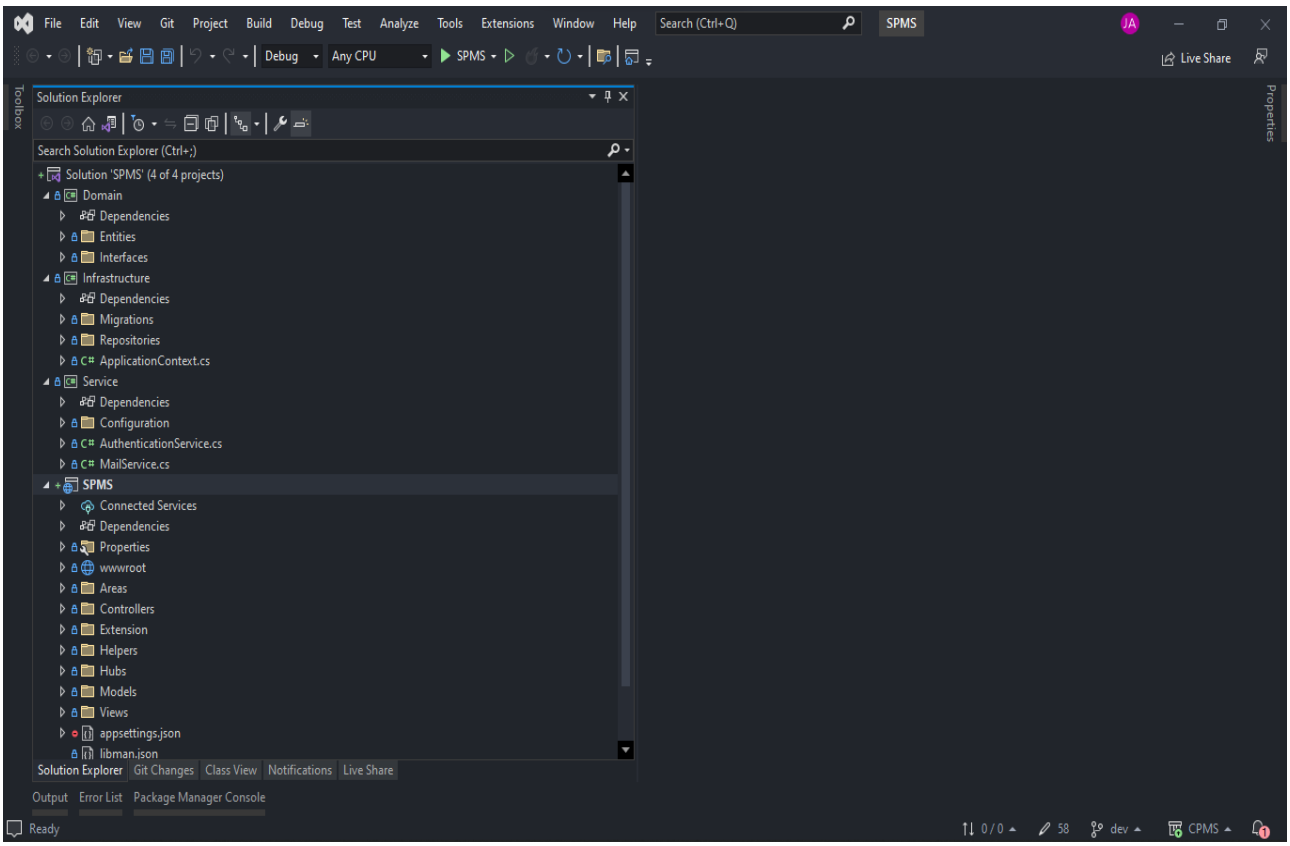
In his book "Clean Architecture: A Craftsman's Guide to Software Structure and Design," Robert C. Martin (MARTIN, 2017) defined the term "Clean Architecture." Systems in this architecture can be broken down into two basic categories: policies and details. The policies are the corporate rules and processes, while the details are the items necessary to carry out the policies. 2017 (MARTIN) this phase is when Clean Architecture starts to set itself apart from other architectural styles. The system must be able to identify the policies as the primary components of the architecture and the details as unimportant to the policies.

It is not necessary to choose the database or framework at the start of the development process in a clean architecture because these are details that do not interfere with the policies and, as a result, can be changed over time.

Layer Separation

There is a clear division of layers in Clean Architecture. The architecture is framework-independent, i.e., the internal layers that contain the business rules do not depend on any third-party library, which allows the developer to use a framework as a tool and not adapt the system to meet the specifications of a particular technology. Clean Architecture also has the following advantages: testability, UI independence, database independence, and independence from any external agents (business rules should not know anything about the interfaces of the external world).

Figure 1 below shows the layer separation for this application



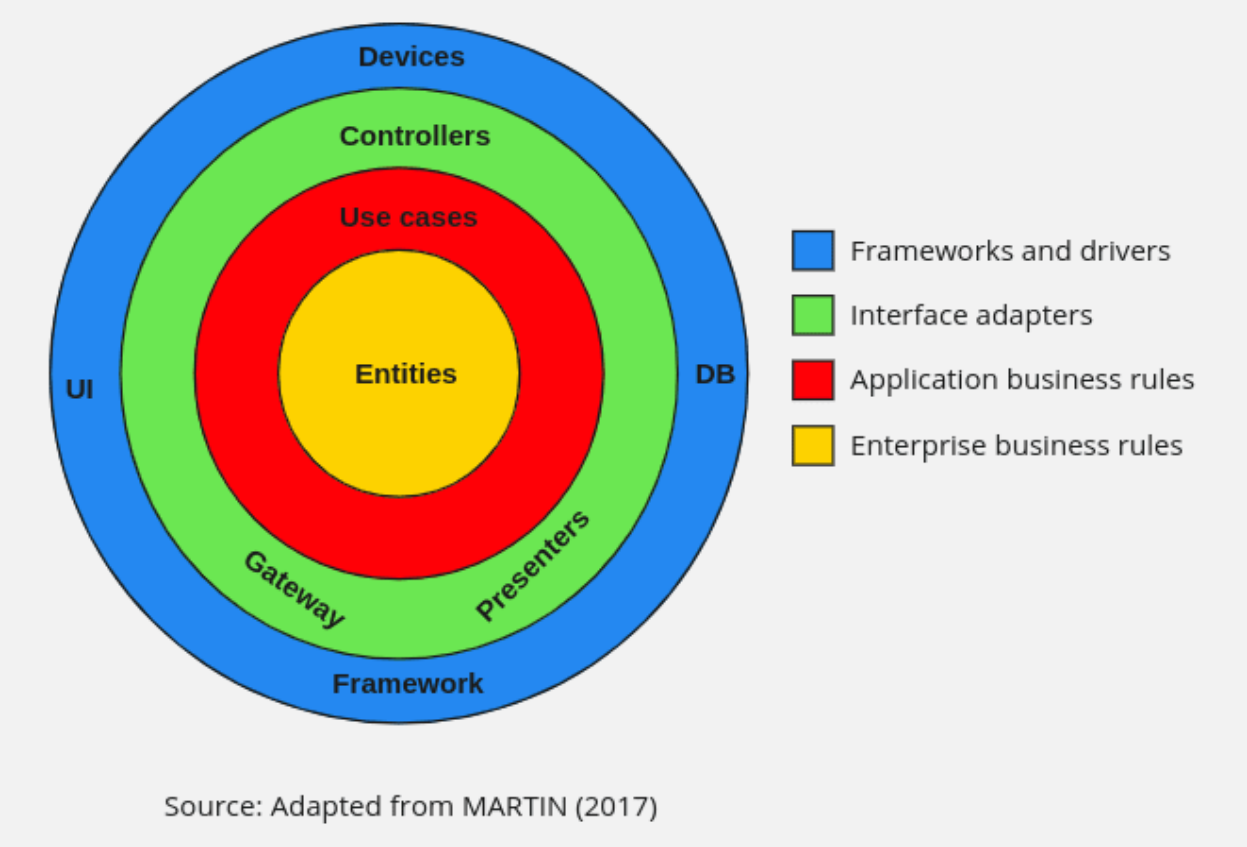
- The main idea behind clean architecture is that the application's core logic is rarely changed, allowing it to be independent and considered core.
- The Dependency Rule is the overarching rule that allows this architecture to function. According to this rule, source code dependencies can only point inwards, and nothing in an inner circle knows anything about something in an outer circle.
- By layering the software and following The Dependency Rule, you will create an intrinsically testable system with all of the benefits that entails. When any of the system's external components, such as the database or the web framework, become obsolete, you can easily replace them.
- In clean architecture, the domain and application layers remain in the center of the design which is known as the core of the application.
    - The domain layer contains enterprise logic, and the application layer contains business logic.
    - Enterprise logic can be shared across many related systems, but business logic is not sharable as it is designed for specific business needs.
    - If you do not have an enterprise and are just writing a single application, then these entities are the business objects of the application.

Clean architecture has the following advantages:

- Frameworks Independent – The architecture does not depend on the existence of some library of feature-laden software. This allows you to use such frameworks as tools.
- UI Independent – It is loosely coupled with the UI layer. So, you can change UI without changing the core business.
- Independent of Database – You can swap out SQL Server or Oracle, for Mongo DB, Bigtable, Couch DB, or something else. Your business rules are not bound to the database.
- Highly maintainable – It is following the separation of concern.

- Highly Testable – Apps built using this approach, especially the core domain model and its business rules, are extremely testable.

To illustrate all these concepts, the diagram shown in figure 2 below was created.



Source: Adapted from MARTIN (2017)

3.2 SYSTEM DESIGN

Designing a project management system entails translating the requirements specification into a physical form, which requires the use of various patterns to achieve the desired system.

3.3.1 LOGICAL DESIGN

The logical design transforms the system requirements specification into a system model by implementing the system's major features.

The system accepts input from a user then verifies if the user is part of the institution using HttpClient, then proceeds to get some required data using Html Agility Pack. The required data are stored in the database which now gives users restrictive access (based on their roles) to the project management platform

This design allows

An Admin:

- Perform administrative duties on the student project management system

A student:

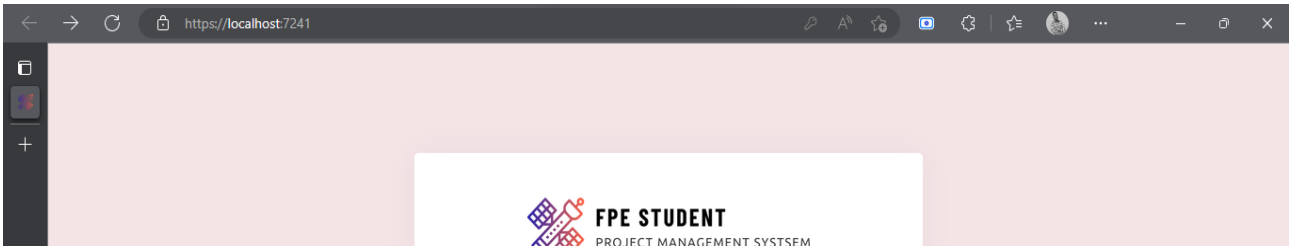
- Submit proposal for a project (either as an individual or group of students)
- Submit each chapters of the project
- Await approval, also feedback from project supervisor
- Get notified when there is an update on the submitted material
- Chat with other students in group

A Supervisor:

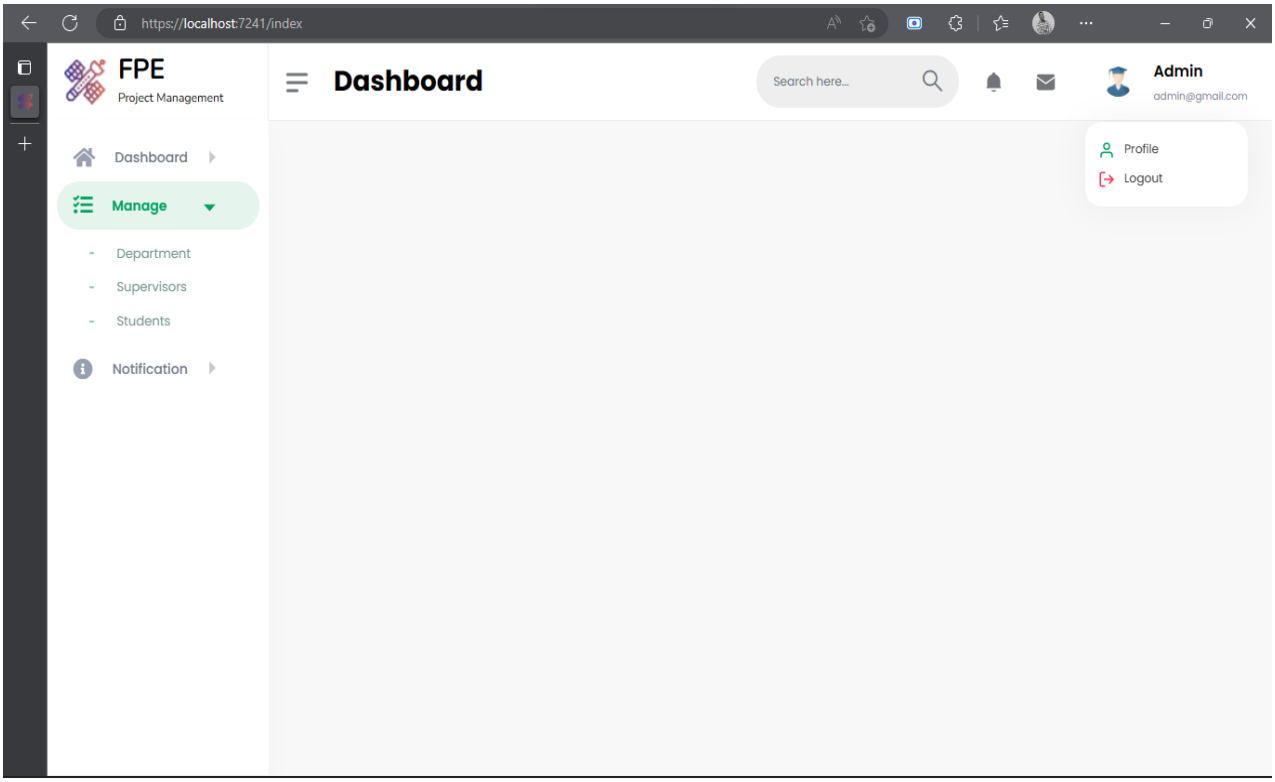
- See all project students
- Approve/Reject proposals and research materials
- Give feedback on every material submitted
- Broadcast Information to project students
- View all completed projects by students in the department (Present/Past)

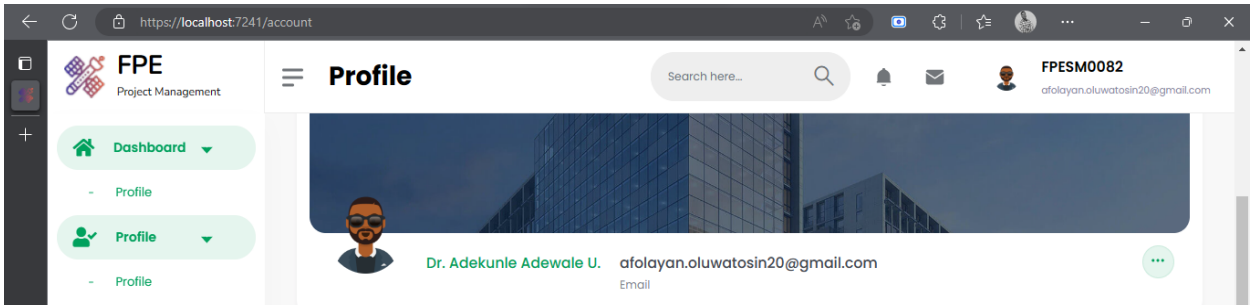
STUDENT PROGET MANAGEMENT SYSTEM

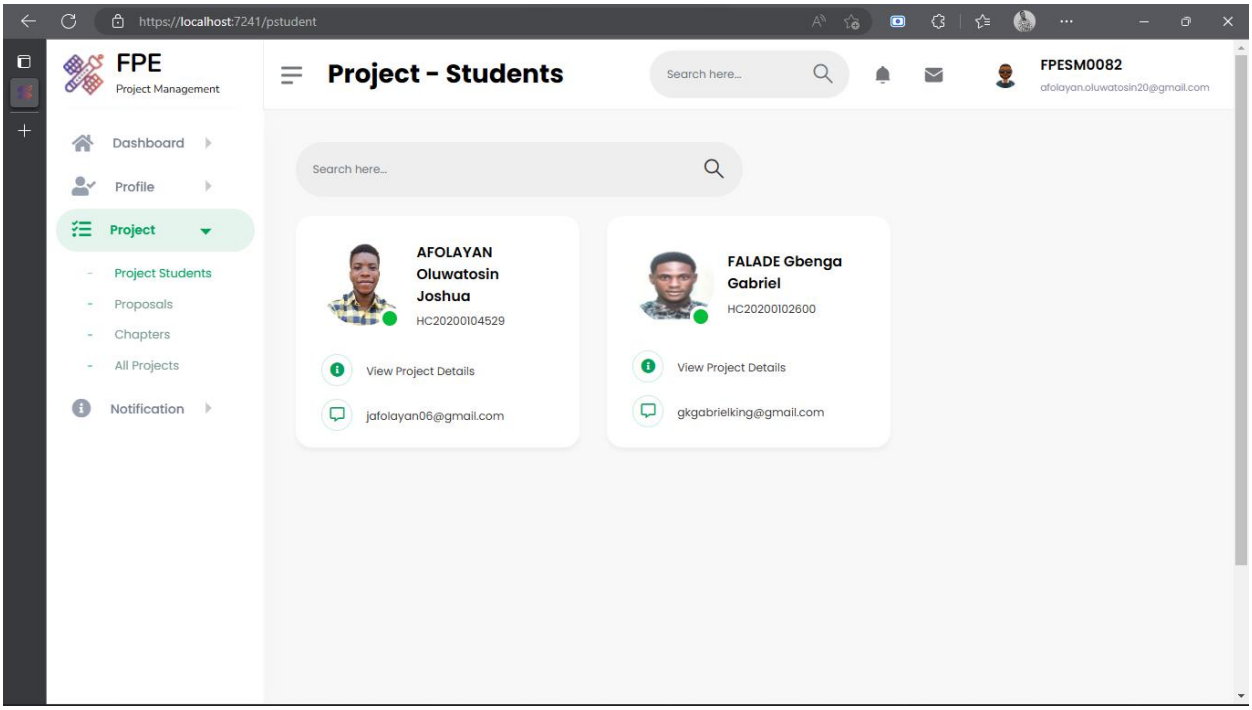
Login Screen

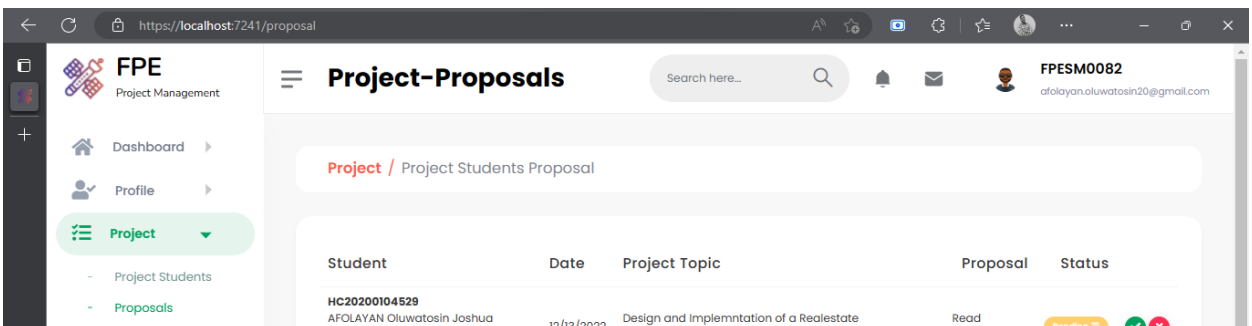


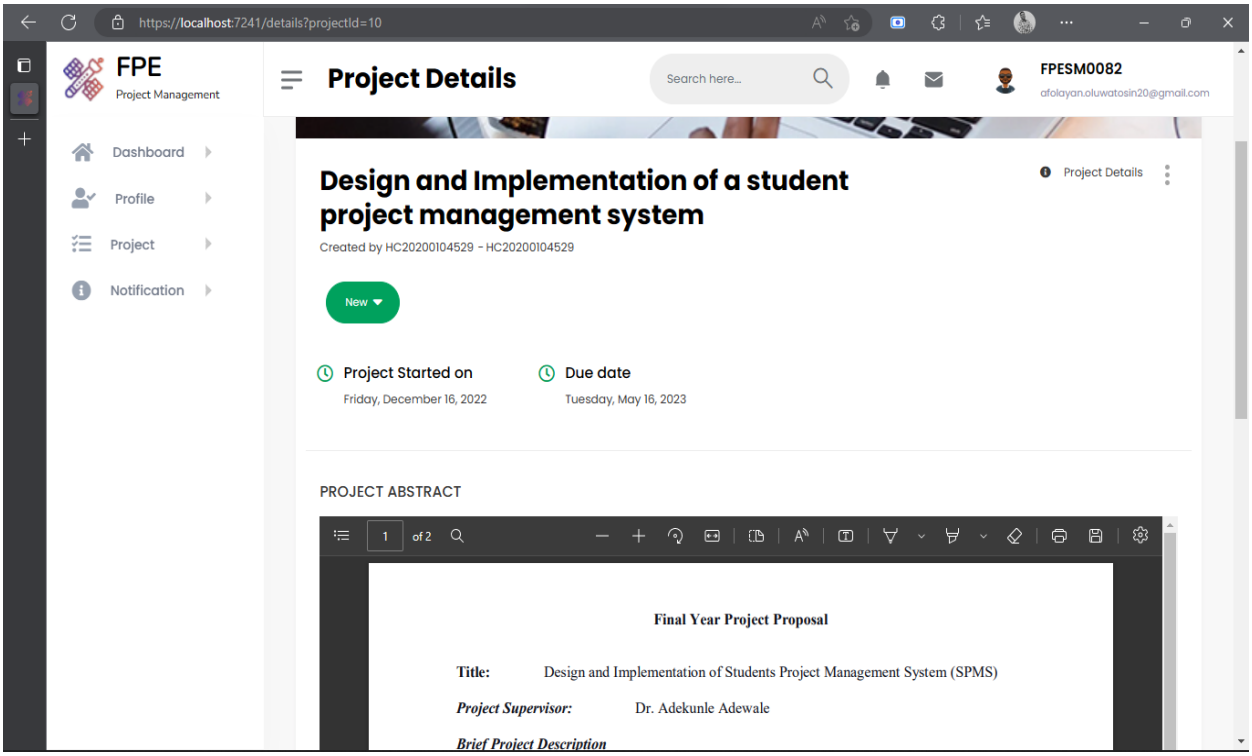
Admin View Screen

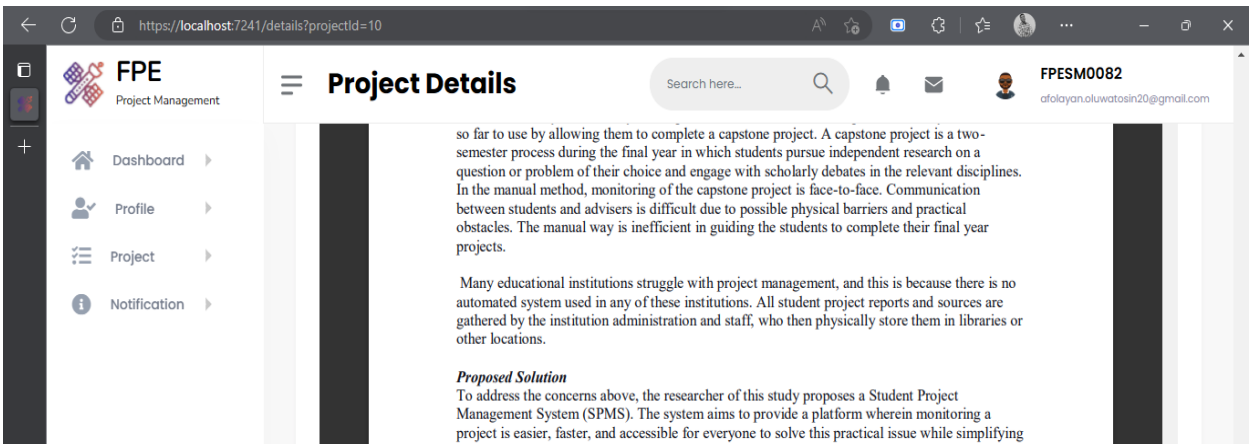




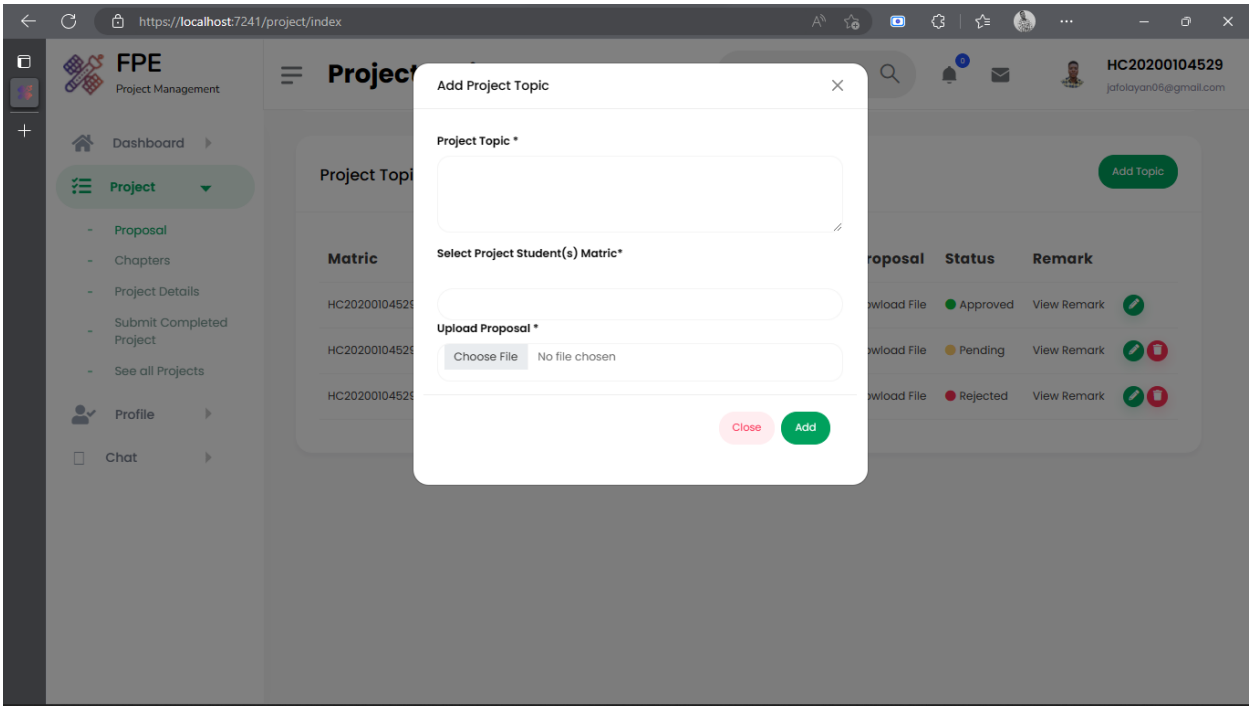

https://localhost:7241/pstudent
FPE
Project Management
Project - Students
Search here...
FPESM0082
afolayan.oluwatosin20@gmail.com
Dashboard
Profile
Project
Project Students
Proposals
Chapters
All Projects
Notification
Search here...
AFOLAYAN Oluwatosin Joshua
HC20200104529
View Project Details
jafolayan06@gmail.com
FALADE Gbenga Gabriel
HC20200102600
View Project Details
gkgabrielking@gmail.com

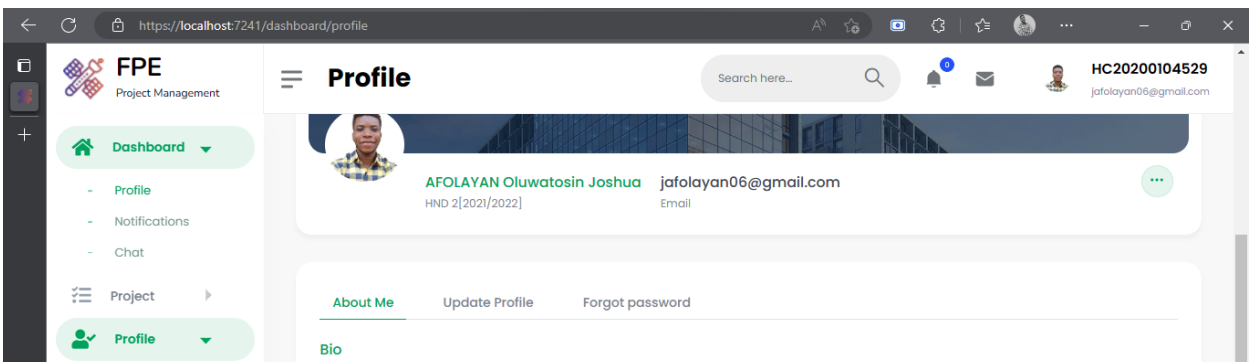

https://localhost:7241/proposal
FPE
Project Management
Project-Proposals
Search here...
FPESM0082
afolayan.oluwatosin20@gmail.com
Dashboard
Profile
Project
Project Students
Proposals
Project / Project Students Proposal
Student
Date
Project Topic
Proposal
Status
HC20200104529
AFOLAYAN Oluwatosin Joshua
Design and Implemntation of a Realestate
Read

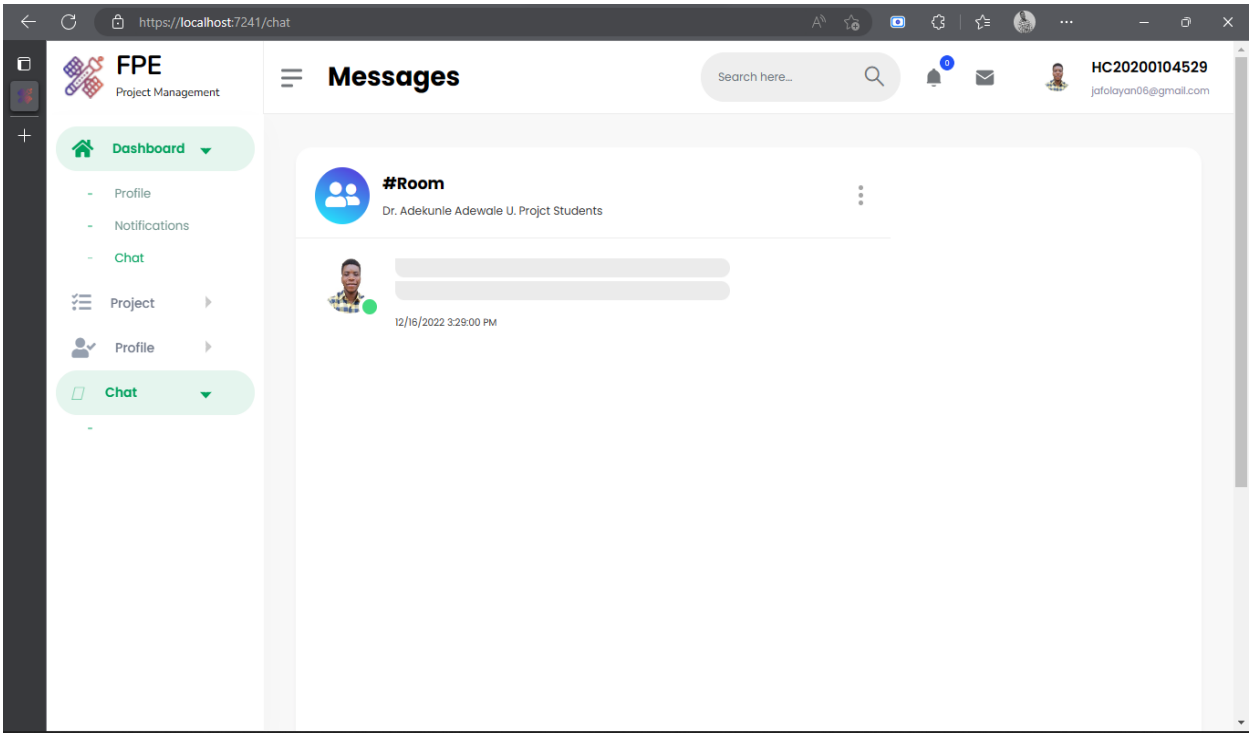

https://localhost:7241/details?projectId=10
FPE
Project Management
Project Details
Search here...
FPESM0082
afolayan.oluwatosin20@gmail.com
Dashboard
Profile
Project
Notification
Design and Implementation of a student project management system
Project Details
Created by HC20200104529 - HC20200104529
New
Project Started on
Friday, December 16, 2022
Due date
Tuesday, May 16, 2023
PROJECT ABSTRACT
1 of 2
Final Year Project Proposal
Title: Design and Implementation of Students Project Management System (SPMS)
Project Supervisor: Dr. Adekunle Adewale
Brief Project Description

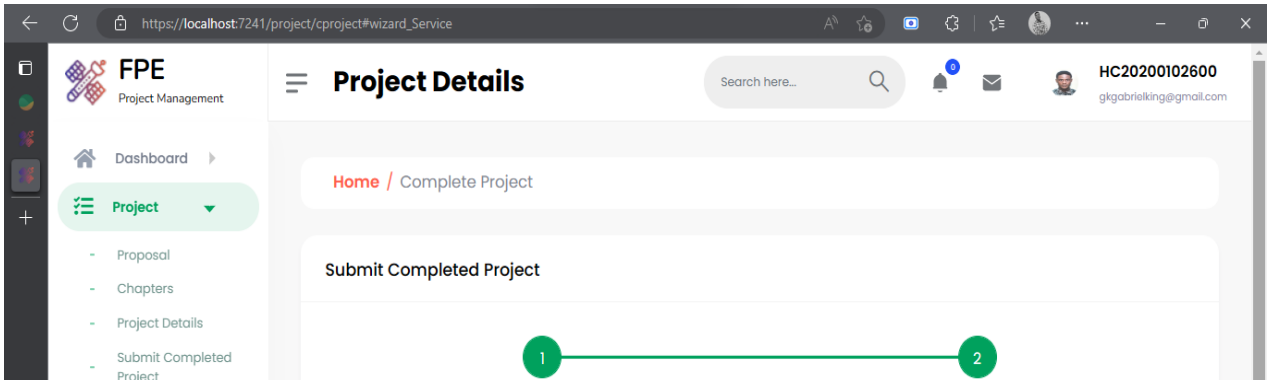

https://localhost:7241/details?projectId=10
FPE
Project Management
Project Details
Search here...
FPESM0082
afolayan.oluwatosin20@gmail.com
Dashboard
Profile
Project
Notification
so far to use by allowing them to complete a capstone project. A capstone project is a two-semester process during the final year in which students pursue independent research on a question or problem of their choice and engage with scholarly debates in the relevant disciplines. In the manual method, monitoring of the capstone project is face-to-face. Communication between students and advisers is difficult due to possible physical barriers and practical obstacles. The manual way is inefficient in guiding the students to complete their final year projects.
Many educational institutions struggle with project management, and this is because there is no automated system used in any of these institutions. All student project reports and sources are gathered by the institution administration and staff, who then physically store them in libraries or other locations.
Proposed Solution
To address the concerns above, the researcher of this study proposes a Student Project Management System (SPMS). The system aims to provide a platform wherein monitoring a project is easier, faster, and accessible for everyone to solve this practical issue while simplifying

Student View Screen

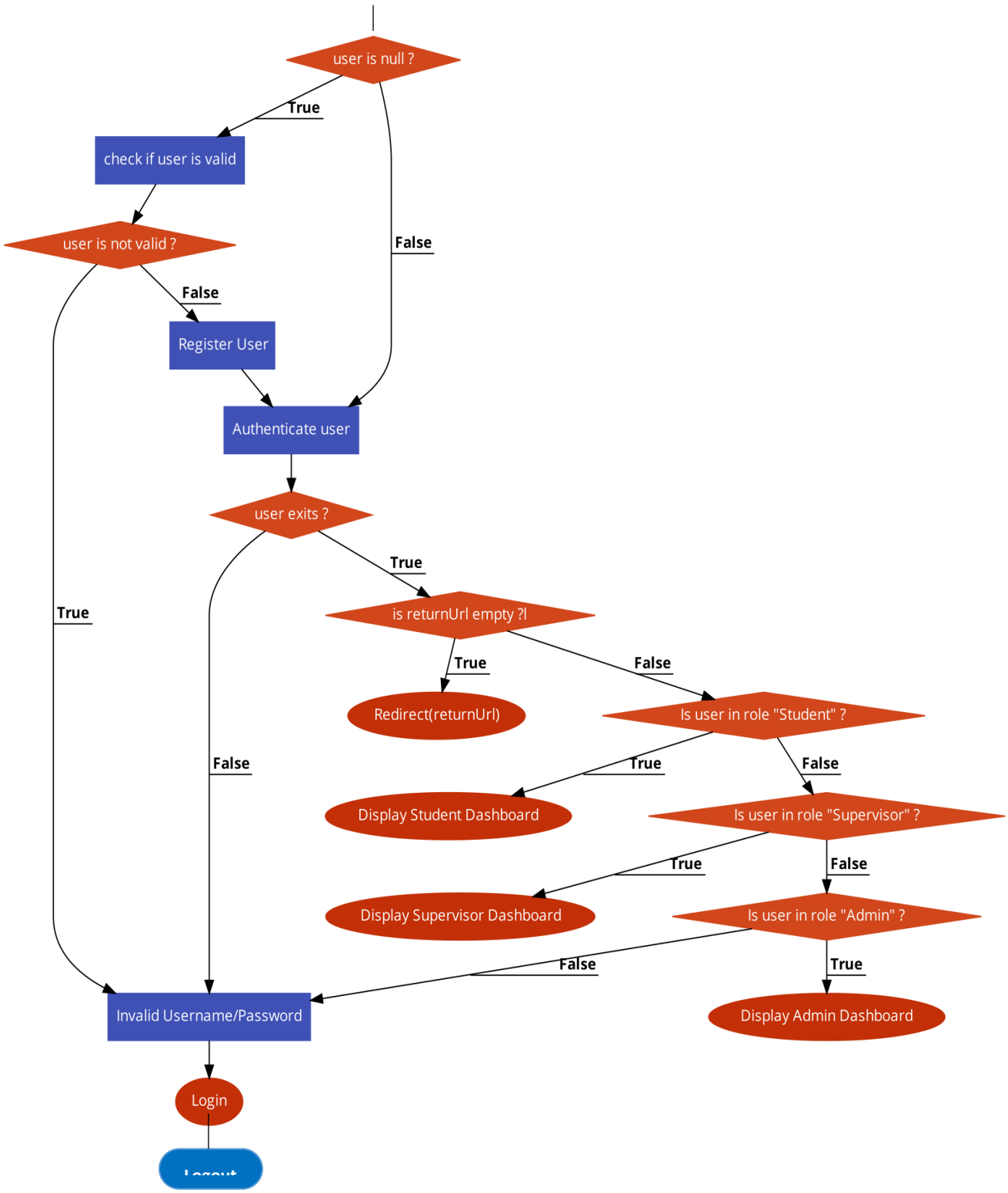





https://localhost:7241/chat
FPE
Project Management
Messages
Search here...
HC20200104529
jafolayan06@gmail.com
Dashboard
Profile
Notifications
Chat
Project
Profile
Chat
#Room
Dr. Adekunle Adewale U. Projct Students
12/16/2022 3:29:00 PM


https://localhost:7241/project/cproject#wizard_Service
FPE
Project Management
Project Details
Search here...
HC20200102600
gkgabrielking@gmail.com
Dashboard
Project
Proposal
Chapters
Project Details
Submit Completed Project
Home / Complete Project
Submit Completed Project
1
2

FLOW CHART DIAGRAM



3.3.1.2 SYSTEM CONTROLS

The input and output are controlled through the use of validations and error handling. ASP.NET Core Identity manages users, passwords, profile data, roles, claims, tokens, email confirmation, and more.

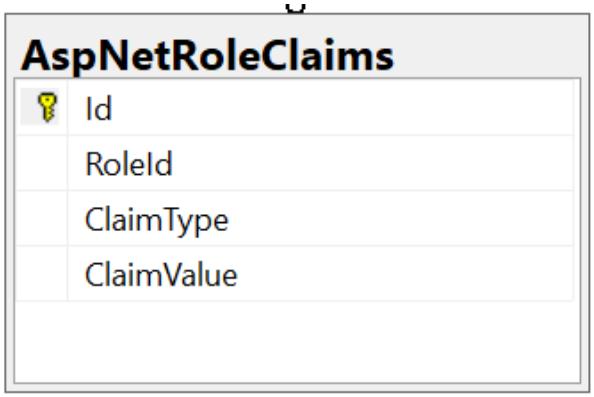
EF Core creates the database and tables using migration based on the conventions and configuration provided in the domain classes.

Session state keeps track of user data while the user browses the web application from one page to another.

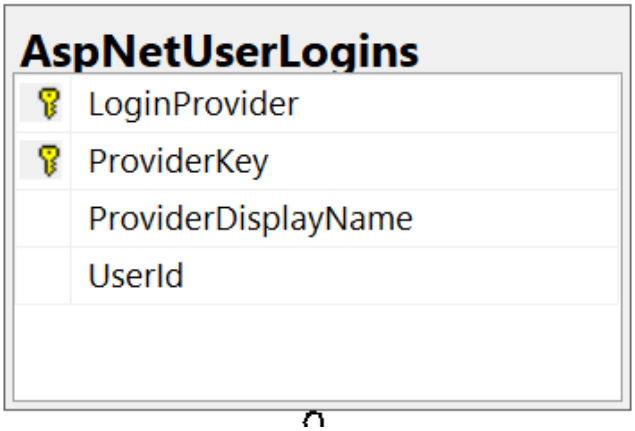
3.3.2 STRUCTURE OF DATABASE

The structure of relational database shows the different tables that make up the database and links among the fields, the database consists of fifteen tables which are:

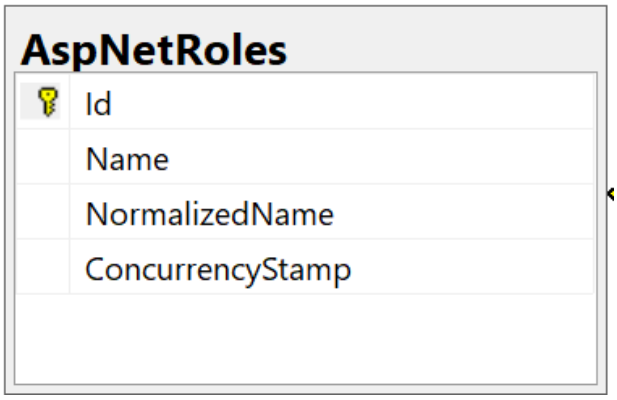
AspNetRoleClaims



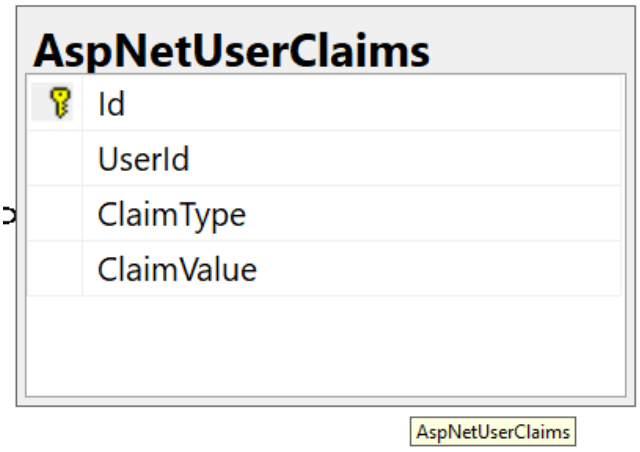
AspNetUserLogins



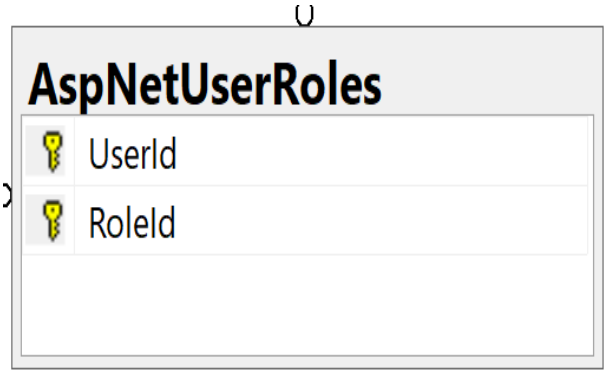
AspNetRoles



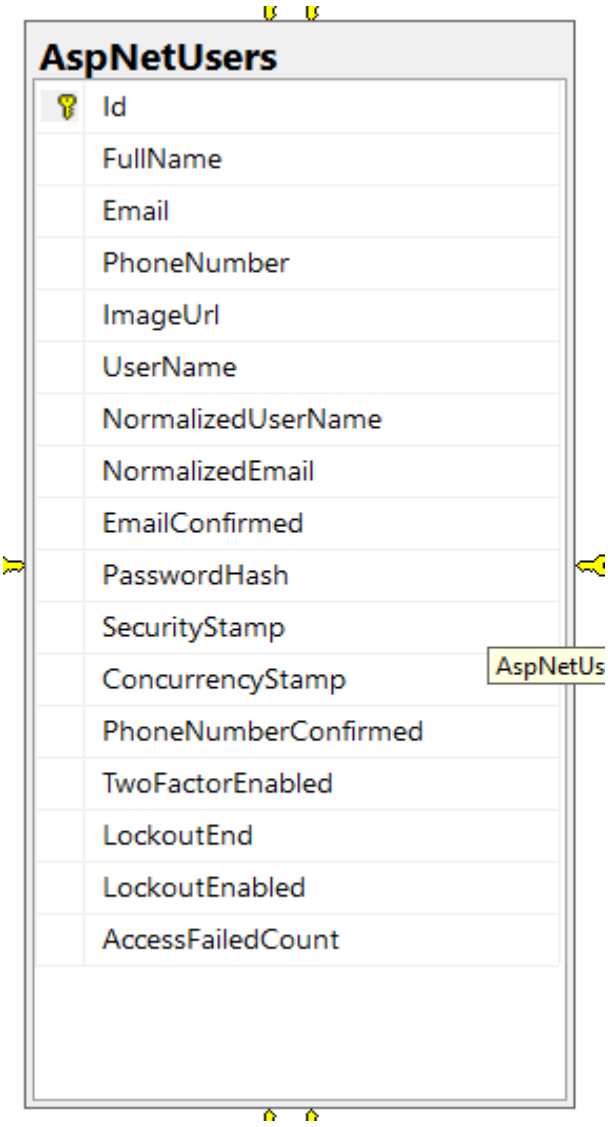
AspNetUserClaims



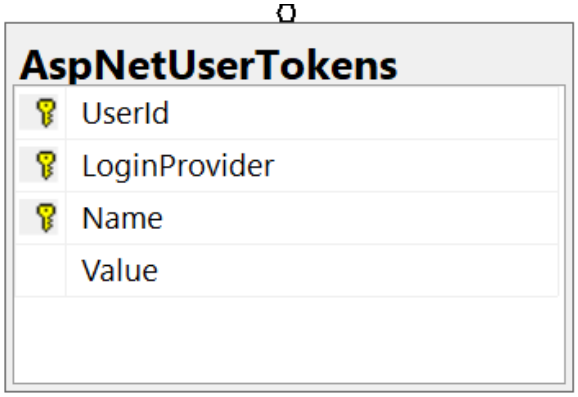
AspNetUserRoles



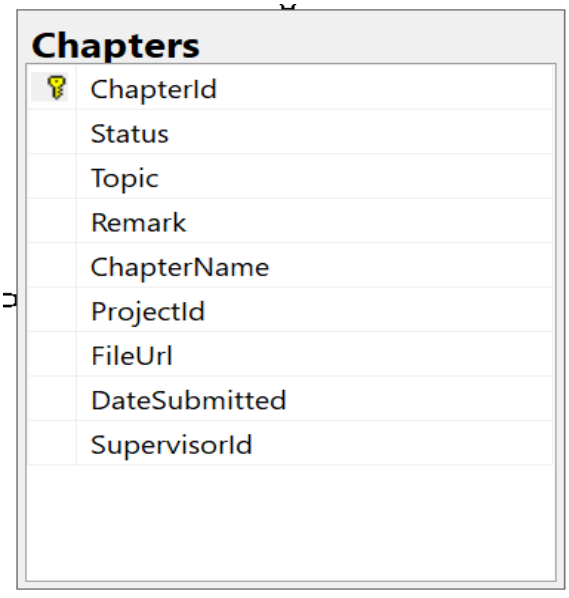
AspNetUsers



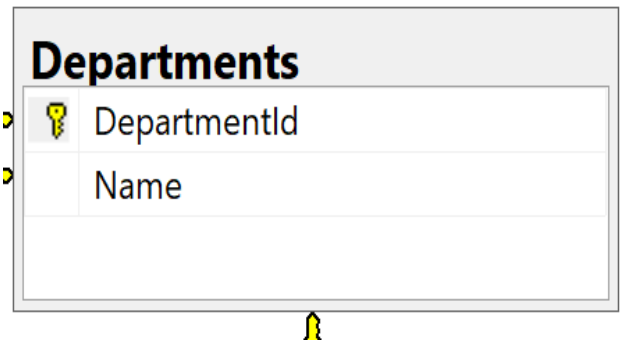
AspNetUserTokens



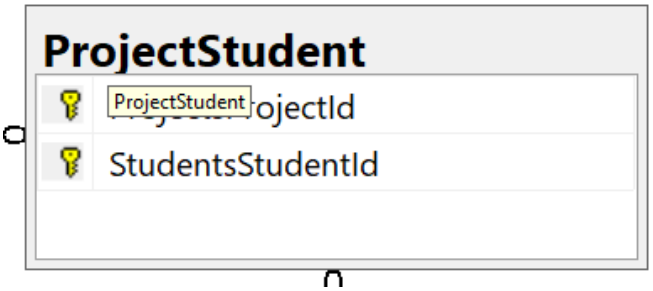
Chapters



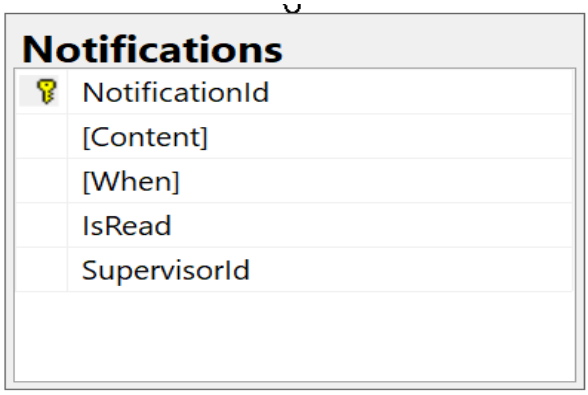
Departments



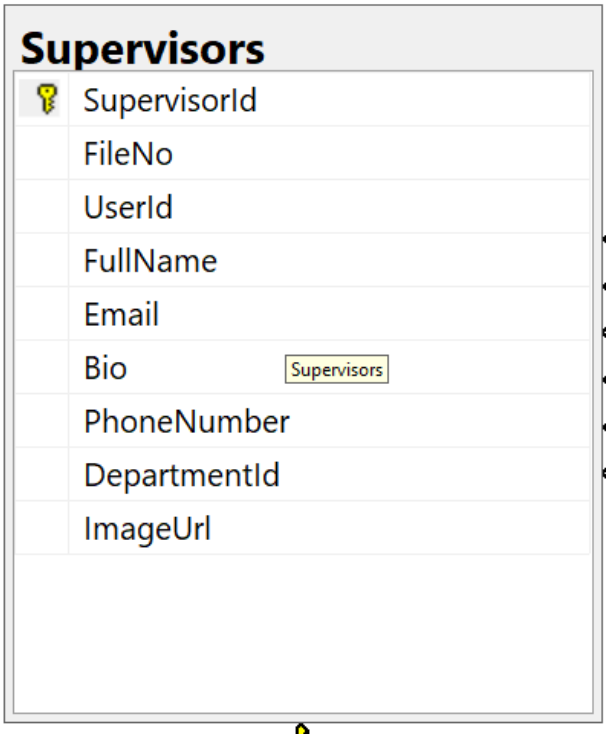
ProjectStudents



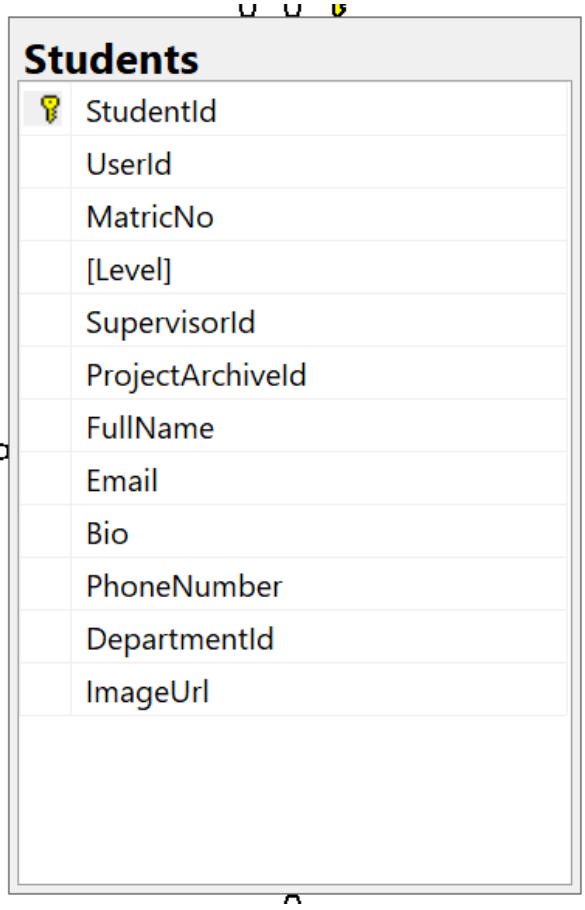
Notifications



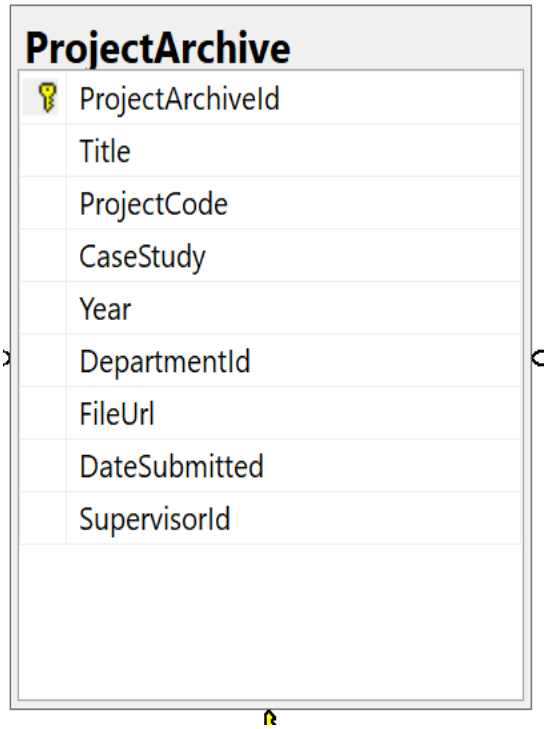
Supervisors



Students



Project Archive



Projects