UMGC Capstone Project Proposal Management System (CaPPMS)

Technical Design Document

Version 1.0

Prepared By:

Bereket Tamrat

Ephrem Kefle

Kathryn Stewart

Marc Bueno

Tarun Lava

Yonas Mekete

Table of Contents

List of Figures

List of Tables

Technical Design Document Approvals

Name	Signature	Date
Professor Dr. Mir Assadullah		
Project Manager Kathryn Stewart		

Revision History

Date	Version	Description
09/29/2020	1.0	Initial Technical Design Document

1. Introduction

1.1. Purpose

The main purpose of the Technical Design Document (TDD) is to define and explain the functionality, view, and the architectural design of the UMGC Capstone Project Proposal Management System (CaPPMS). This technical design document is used by programmers as a blueprint of details and requirements that need to be implemented to create the system. This document is designed to be used as a baseline reference for restricting changes in the scope of the project after the TDD is approved by all the stakeholders.

1.2. Scope

This document illustrates a high-level design of the UMGC Capstone Project Proposal Management System (CaPPMS). The main goal of the system is to create an easy way to keep track of all the UMGC Capstone project proposals submitted by students and clients. The system allows the professor to search, filter, select, delete, and export proposals. It also allows external users to submit their project proposals. The project tracking system creates one central location where all users come to interact with each other.

1.3. Overview

The TDD documents all the aspects of the architecture as shown by the following subsections.

Section 2: Defines the system overview of the project tracking system.

Section 3: Defines the architectural design of the project tracking system.

Section 3.1: Defines the design of the system architecture.

Section 3.2: Defines the decomposition of the subsystem

Section 3.3: Defines the rationale behind selecting the architecture

Section 4: Defines how the system manages data

Section 4.1: Defines the models of data structures and the database.

Section 4.2: Defines the system entities

Section 5: Defines the different parts of the project tracking system

Section 6: Defines Graphical User Interface (GUI)

Section 7: Shows the relationship between system components and functional requirements.

Section 8: Reference Material

1.4. Definition and Acronyms

Below are the terms and abbreviations used in this document.

Table 1 Acronyms

Acronym	Definition
CaPPMS	Capstone Project Proposal Management System
CCB	Change Control Board
CSS	Cascading Style Sheet
FAQ	Frequently Asked Questions
GUI	Graphical User Interface
HTML	Hypertext Markup Language
MVC	Model View Controller
ORM	Object Relational Mapping
POC	Point-of-Contact
POJOs	Plain Old Java Object
SRS	Software Requirement Specification
SWEN	Software Engineering
STP	Software Test Plan
SWEN	Software Engineering
TDD	Technical Design Document
UI	User Interface
UMGC	University of Maryland Global Campus

2. System Overview

A platform is needed where clients and customers can submit capstone project proposals for future University of Maryland Global Campus (UMGC) Software Engineering (SWEN) 670 students. Those projects would then go through an approval process involving UMGC professors and stakeholders who would review and determine which project could be completed by the students.

The UMGC Capstone Project Proposal Management System (CaPPMS) is a web-based application which allows customer, clients, and former students to submit detailed proposals of projects to be designed and implemented by UMGC SWEN 670 students. The system will allow the professors to review, update, change status, and export approved projects to share with future classes.

3. System Architecture

3.1. Architectural Design

The architectural design below represents Spring Boot exporting a REST API using Spring (Web) MVC. Spring Boot will interact with a PostgreSQL Database using Spring Data and Hibernate as JPA Provider. Clients, in the architecture below, interact with the REST API using HTTP request / response roundtrips, displaying data on the components (COMP.). The routing mechanism is used to browse through pages with Angular connecting via Hibernate as JPA Provider.

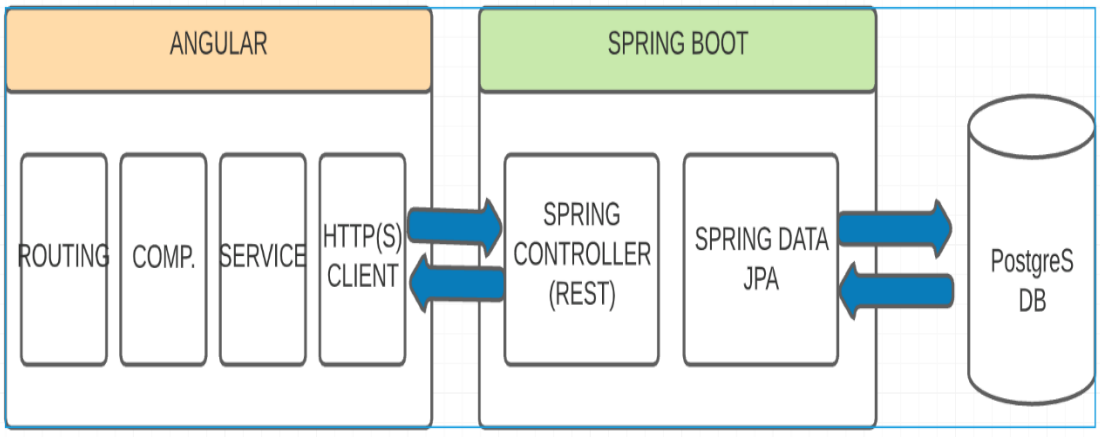

Figure 1 Architectural design

3.2. Decomposition Description

A structural decomposition diagram shows a high-level function, process, organization, data subject area, or another type of object. The diagram illustrates a more detailed view that is broken down into lower levels of the components. The following diagram shows the decomposition of the CaPPMS:

Structural Decomposition Diagram

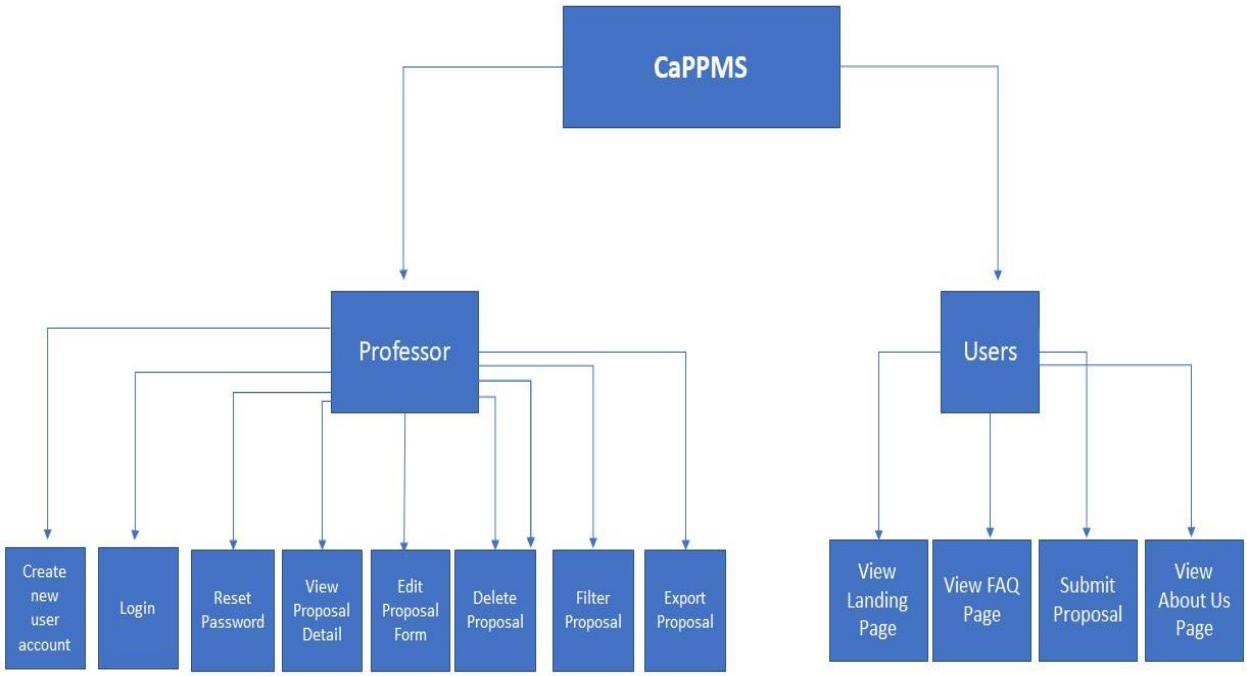

Figure 2 CaPPMS Decomposition Diagram

3.3. Design Rationale

The design rationale shows the reasoning and argument behind making decisions when designing a system. The design intends to achieve the goal that the designer has in mind to fulfill the required function. The Spring framework is selected to build the CaPPMS web application since it addresses most of the infrastructure functionalities of enterprise applications. We are using PostgreSQL as a database. Using a spring boot framework with PostgreS gives us the following advantages:

- Spring boot allows developers to develop enterprise applications using POJOs (Plain Old Java Object) which eliminates the need for an enterprise container such as an application server and Enterprise Java Beans while giving the option of using a robust

servlet container. In short, no need to use complex EJB (Enterprise Java Beans) which drastically simplifies development and maintenance and the overall architecture

- Spring framework provides an abstraction layer on existing technologies like servlets, jsps, jdbc, etc. to simplify the design, development, and maintenance processes.
- Spring has existing technologies like the ORM framework which facilitates an easy link to a database without any special data access code. Hibernate can be used as a JPA provider in this case.
- Spring Web MVC framework offers a well-designed web MVC framework compared to the legacy web framework. The MVC design pattern promotes a separation of concerns among User Interface, Business Logic and Data layer.
- Spring application also can be used for the development of different types of applications, like standalone applications, standalone GUI applications, and Web applications.
- Using a RESTful API provides a great deal of flexibility. Data is not tied to resources or methods, so REST can handle multiple types of calls, return different data formats.
- PostgreSQL database offers robustness to full SQL compliance to the architecture while keeping the solution cost-effective.
- Spring Boot and Angular form a powerful tandem that works great for developing web applications with a minimal footprint. In this project, we will use Angular for creating a JavaScript-based frontend.

4. Data Design

4.1. Data Description

A database is a collection of relational data, figures, and facts. By using the figures and facts stored in the database a piece of useful information can be processed. Different attributes of the project proposal tracking system like users, project title, contacts, login credentials, etc. can be are recorded as data. By using the stored information about each project proposal, a useful information can be produced as needed. Using a database management system makes it easy to store, retrieve, and manipulate data.

4.2. Data Dictionary

Table 2 Project Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	P_ID	Integer	NNNNNN	6	Unique number ID for projects
	Project_Title	Text		50	Proposed project's title
	Project_Description	Text		128	Description of a proposed project
	Project Website	Text		50	Website of a project
FK	Project_Status_ID	Integer	NNNNNN	6	Unique number ID for each phase of a project
	Comments	Text		128	Project related comments

Table 3 User Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	U_ID	Integer	NNNNNN	6	Unique number ID for Users
	First_Name	Text		50	First name of a user
	Last_Name	Text		50	Last name of a user
	Email	Text		50	User's email address
	Phone	Text		50	User's phone number
FK	User Type	Text		255	**User Types:** 1. User: Person entering the suggestion or idea for consideration 2. Sponsor: Person responsible for requirements definitions and ensuring the project meets the stated need. The sponsor is either the Subject Matter Expert (SME) for describing functional requirements or a Point-of-Contact (POC) who provides functional requirements. 3. Liaison(s): Other stakeholder(s) or POCs whose inputs, support,

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
					facilitation, or approval is necessary for the success of the project such as 1) SMEs designated by the Project Sponsor, 2) POCs providing access to data repositories beyond the authority of the project Sponsor, 3) POCs providing regulatory requirements or approval, 4) POCs with data governance authority over data used in a project, or 5) other critical affiliations or associations.

Table 4 Professor Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	Professor_ID	Integer	NNNNNN	6	Unique number ID for all Professors
	Professor_F_Name	Text		20	First Name for Professor
	Professor_L_Name	Text		20	Last Name for Professor
U	Email	Text		30	Professor's email

Table 5 Sponsor Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	Sponsor_ID	Integer	NNNNNN	6	Unique number ID for a sponsor
	Title	Text		20	Title of a project sponsor
	Website	Text		50	Sponsor Website

Table 6 Liaison Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	Liaison_ID	Integer	NNNNNN	6	Unique number ID for a liaison

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
	Title	Text		20	Title of a project liaison

Table 7 Phone User Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	User_ID	Integer	NNNNNN	6	Unique number ID of a user
	Phone_Number	Integer	NNNNNNNNN	10	User's phone number
FK	Phone_Type	Text		20	User's phone type

Table 8 Phone Type Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	Type_ID	Integer	NNNNNN	6	Unique number ID for phone types
	Type_Descr	Text		20	Phone type description

Table 9 Account Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	Account_ID	Integer	NNNN	4	Unique number ID for a user account
	Password	Text	NNNNNNNN	8	Professor's password
U	Username	Text	NNNNNNNN	8	Professor's username

Table 10 Attachment Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	P_ID	Integer	NNNNNN	6	Unique number ID for projects
PK	File_Name	Text		20	Name of the file where the project data is saved

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
	File_Description	Text		255	Detail about the file
	File_Path	Text		128	The file directory of a saved file

Table 11 Status Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	Status_ID	Integer	NNNNNN	6	Unique number ID for each project phase
	Status_Descr	Text		255	A detailed description of each project phase

Table 12 FAQ Table

P/F Key	Field Name	Data Type	Data Format	Field Size	Description
PK	FAQ_ID	Integer	NNNNNN	6	Unique number ID for each FAQ
	FAQ_Question	Text		255	Frequently Asked Question
	FAQ_Answer	Text		255	Answer provided for the FAQs

4.3. Entity Relationship Diagram

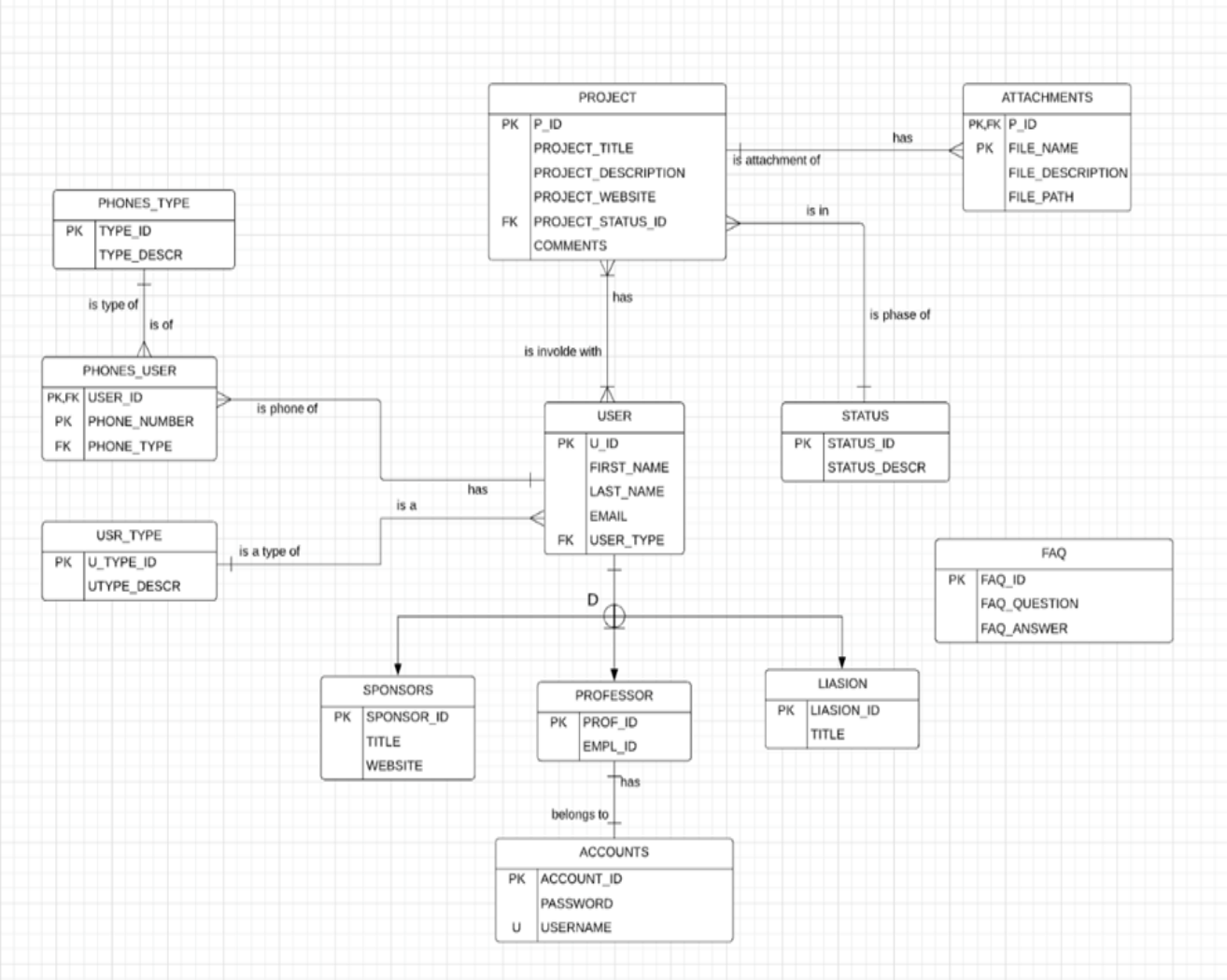

Figure 3 Entity Relationship Diagram

5. Component Design

The following section talks about the component-level design of the software. The architecture is roughly MVC (Model-View-Controller) and therefore the rest of the section will talk about the design of the three layers. The technologies that will be used are based on the Spring framework.

5.1. Presentation Layer

The presentation/ view layer in the MVC architecture deals with the views that the end-user will see. In this application, this layer will be implemented with the combination of HTML,

JavaScript, and CSS. One of the key JavaScript libraries that will be used for client-side development is Angular JS and Bootstrap for CSS formatting. The figure below displays the flow of components that the user could encounter.

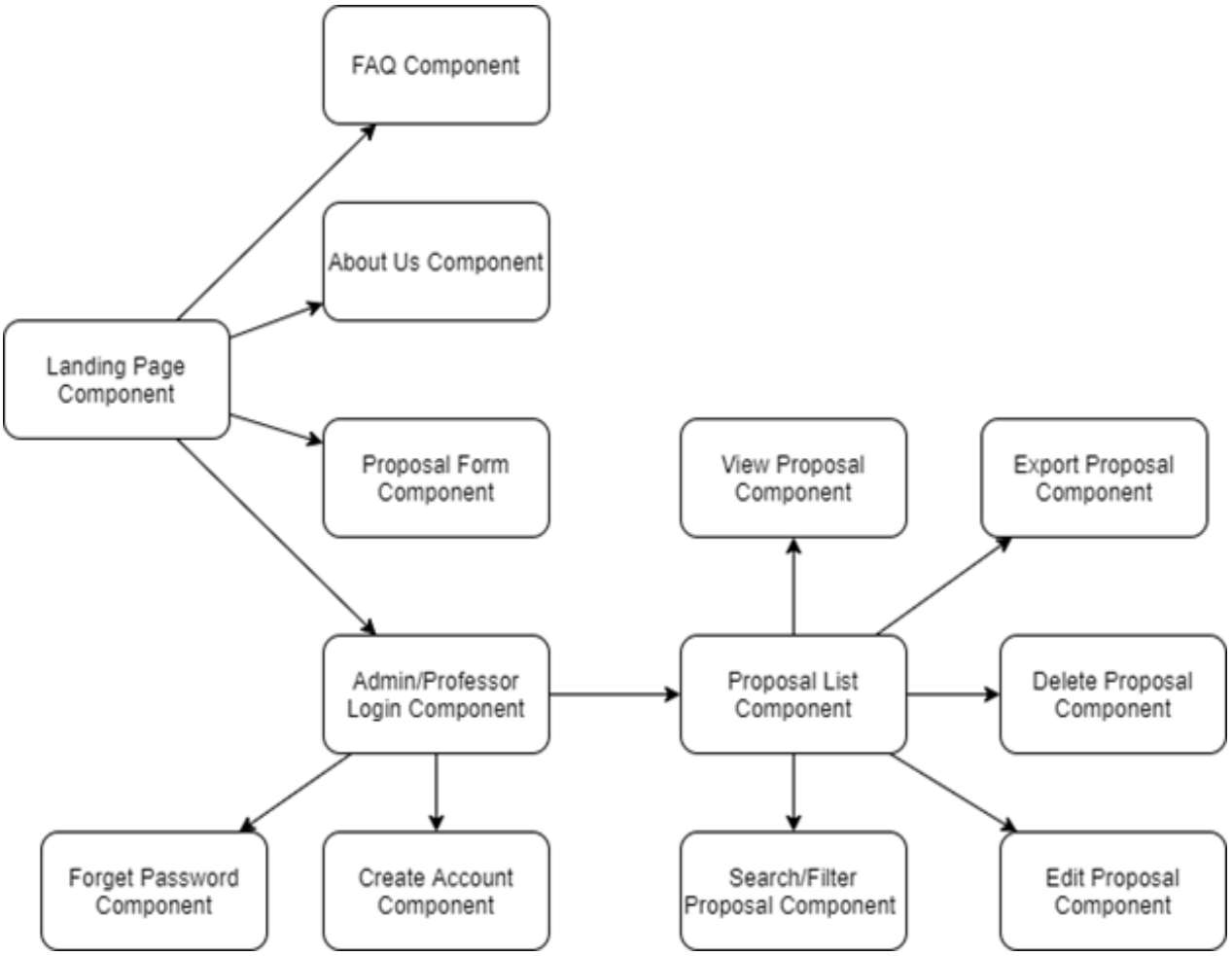

Figure 4 Component Process Flow

5.2. Service Layer

The service layer of the application will deal with the application's logic and acts as a middle layer between the views and the database model. The service layer of the CaPPMS application is designed with the Spring framework. The service layer connects to the presentation layer using Spring MVC and to the database using Spring JPA with Hibernate.

5.3. REST Controller

The CaPPMS has several controller classes that will perform a different function based on the URL and parameters provided. The base URL would /cappms.

Table 13 HomeController.java

Method	**URL path**	**Description**
GET	/home	The default landing page
GET	/faq	Displays the FAQ
GET	/aboutus	Displays the about us page

Table 14 LoginController.java

Method	**URL path**	**Description**
GET	/login	Displays the login page
POST	/login	Submits the user credentials to log in the application
GET	/forgetpassword	Displays the forget password page
POST	/forgetpassword	Submits the email to recover the account
GET	/createaccount	Displays the create new account page
POST	/createaccount	Submits the details to create an account

Table 15 ProposalController.java

Method	**URL path**	**Description**
GET	/proposals	Displays the proposals list
GET	/proposals/{id}/view	Displays the proposal details of the selected proposals to view
GET	/proposals/{id}/edit	Displays the proposal details of the selected proposals to edit
PUT	/proposals/{id}/edit	Submits the edited proposal for validation and updating in the database
DELETE	/proposals/{id}/edit	Deletes the selected proposal
POST	/exportproposal	Generates a Word document of the current proposal

6. Human Interface Design

This section provides the graphical user interface structure of the CaPPMS application and how users navigate to perform different tasks. The sub-sections contain screen images to mock view a selection of the most important aspects of the GUI, along with textual descriptions of their purpose and contents. This human interface design intends to provide system developers the technical details of the application design to be followed during the development of the CaPPMS application. The document may need to be updated later to incorporate possible changes during development.

6.1. Overview of User Interface

The user interface of the CaPPMS application is designed based on Jacob Nielsen's general principles for interaction design and UMGC's logo and visual guidelines. The functionalities of the web application are grouped visually and logically into tabs. The graphic elements and terminology are used consistently across all pages of the web application. The UI minimizes user's memory load by making the project proposal form visible on the home page. The UI accepts input from the users and validate the format to prevent errors and present users with a confirmation option before they commit to the action. The UI design foresees the development of the system to function within a web browser and on various mobile devices.

The following sub-sections show the mock views of a selection of the most important aspects of the GUI for the CaPPMS application.

6.1.1. Home Page

Requirement ID: REQ-1.1, REQ-1.3, REQ-1.4, REQ-1.5

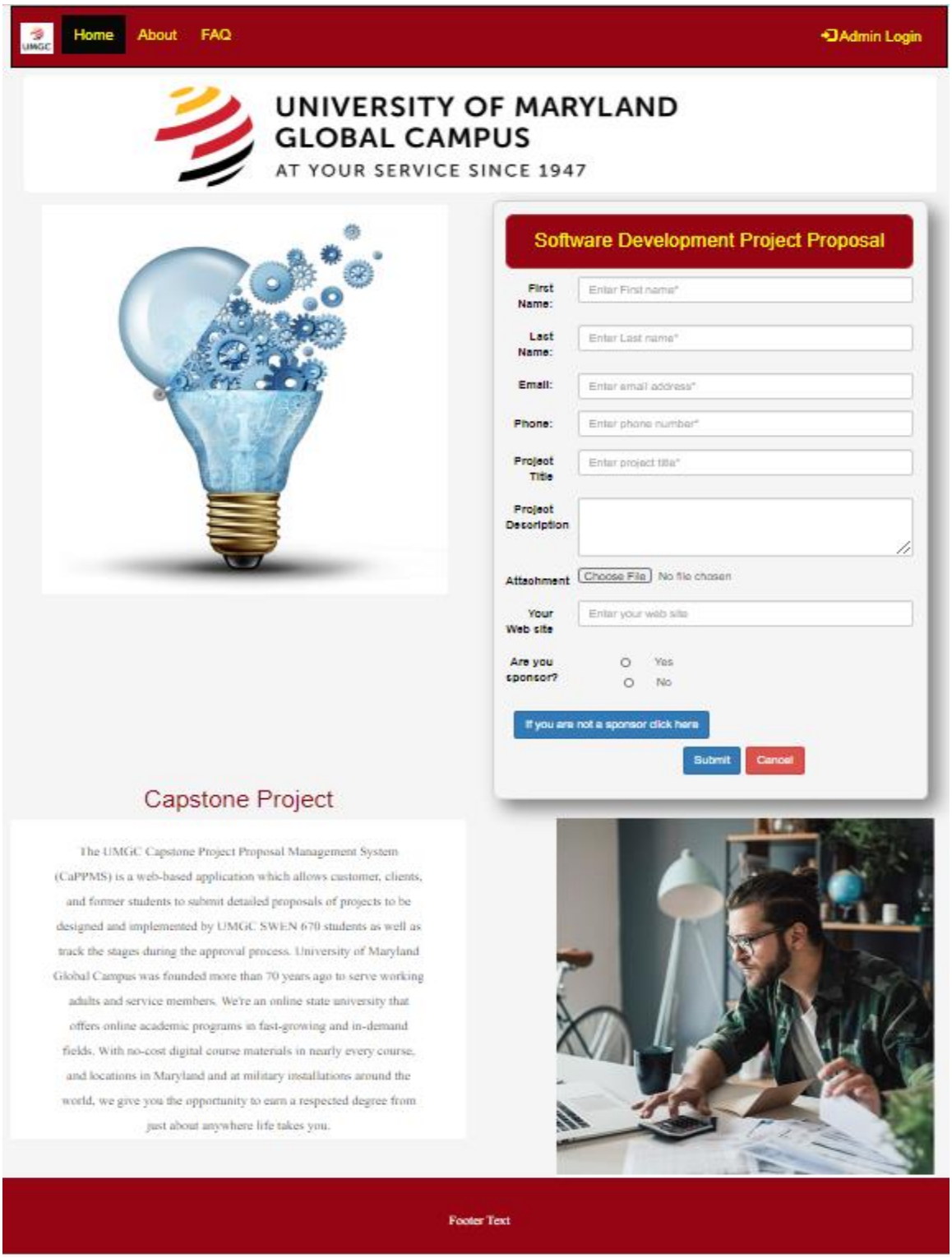

Figure 5 Home Page

Purpose:	The landing page of the project proposal website is the Home Page which contains Home, About, and FAQ buttons, and the project proposal form.
Navigation & User Interaction:	The user can click on the buttons to advance to the corresponding pages. The user can submit a project proposal from the Home Page by entering the required information.

6.1.2 About

Requirement ID: REQ-1.2

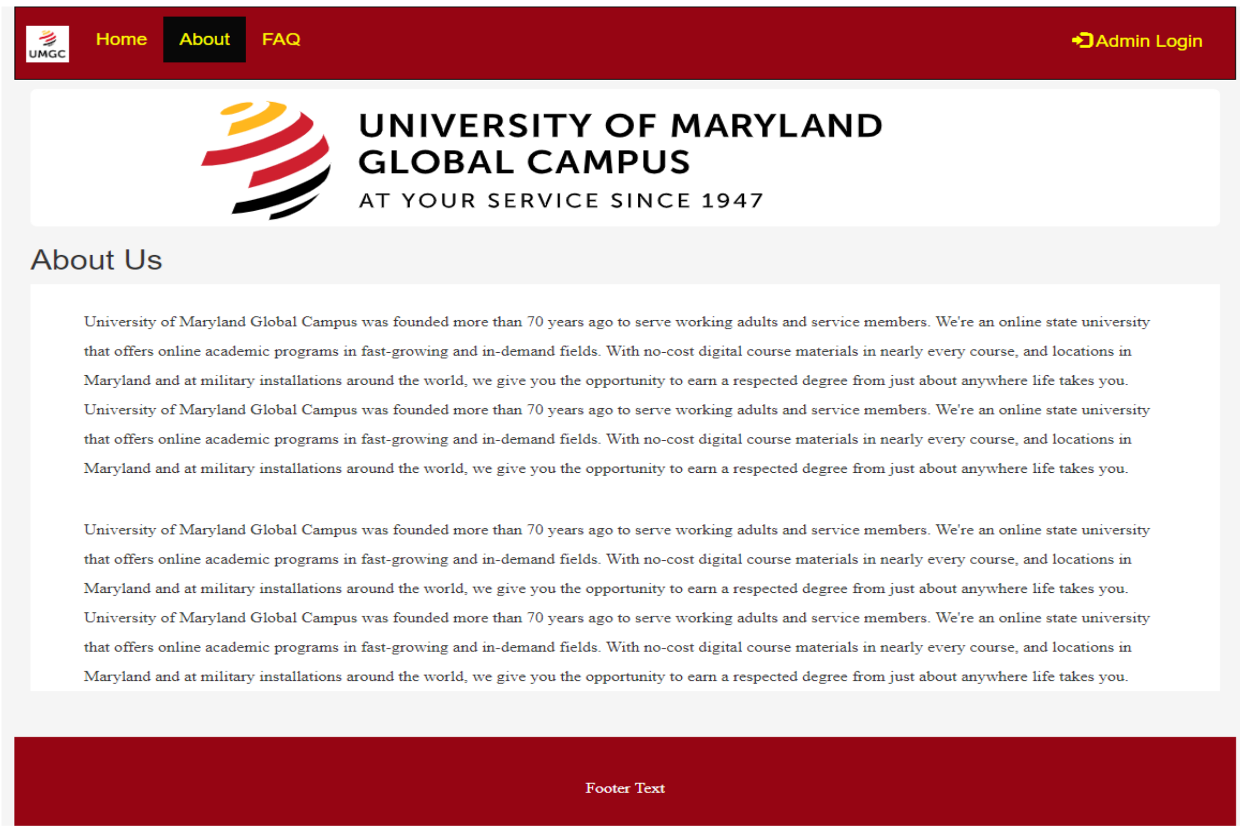

Figure 6 About page

Purpose:	The About Page provides users with the necessary information about the project proposal system.
Navigation & User Interaction:	From the top menu bar, the user clicks on the "About" button to learn about the project proposal website.

6.1.3 FAQ

Requirement ID: REQ-1.6

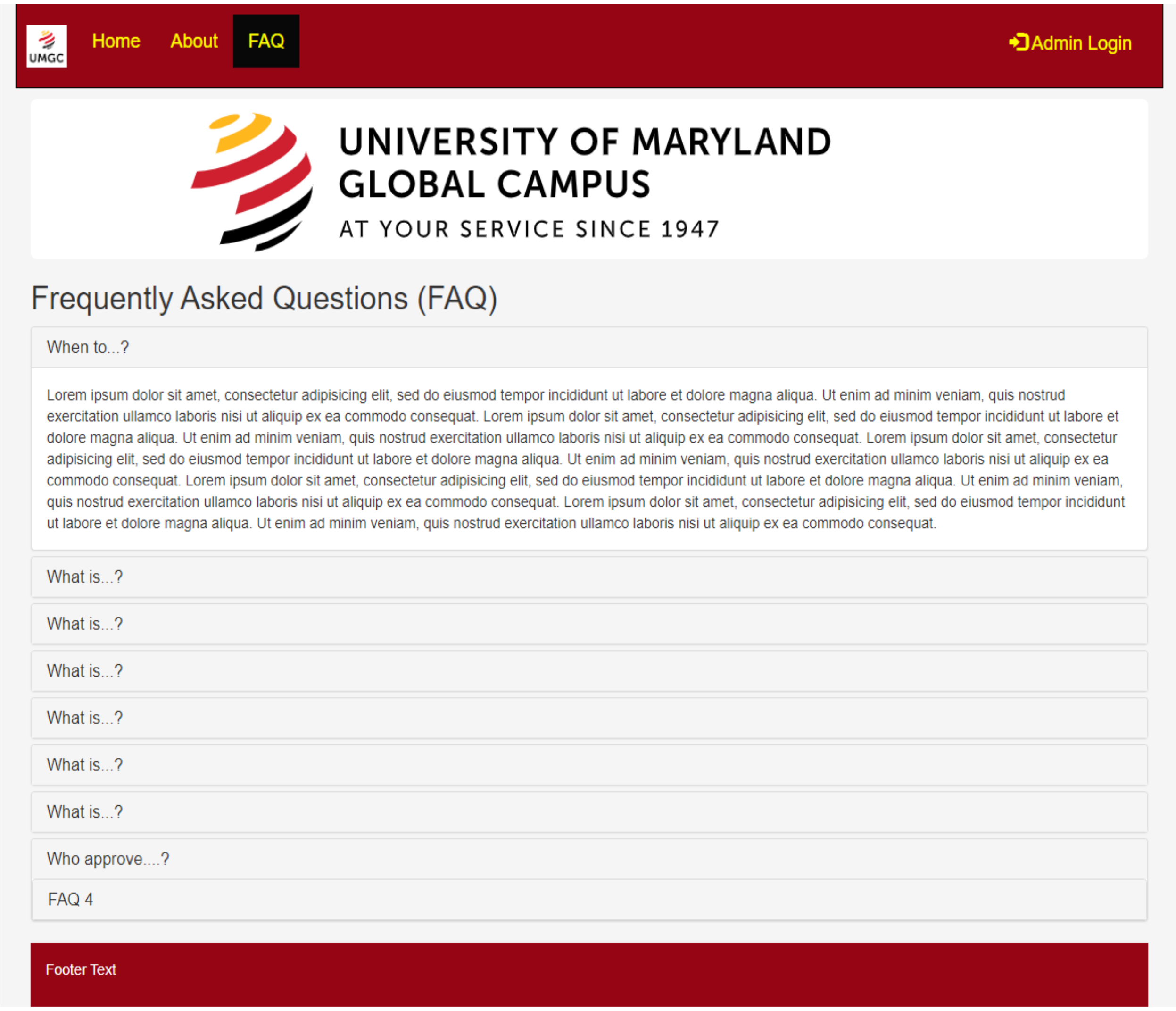

Figure 7 FAQ Page

Purpose:	The FAQ page is designed to organize and provide information on frequent questions and their answers.
Navigation & User Interaction:	From the top menu bar, the user clicks on the "FAQ" button to view the list of frequently asked questions and answers. All FAQs are initially collapsed. The user can expand to view the question or collapse as needed.

6.1.4 Login

Requirement ID: REQ-1.8, REQ-1.11

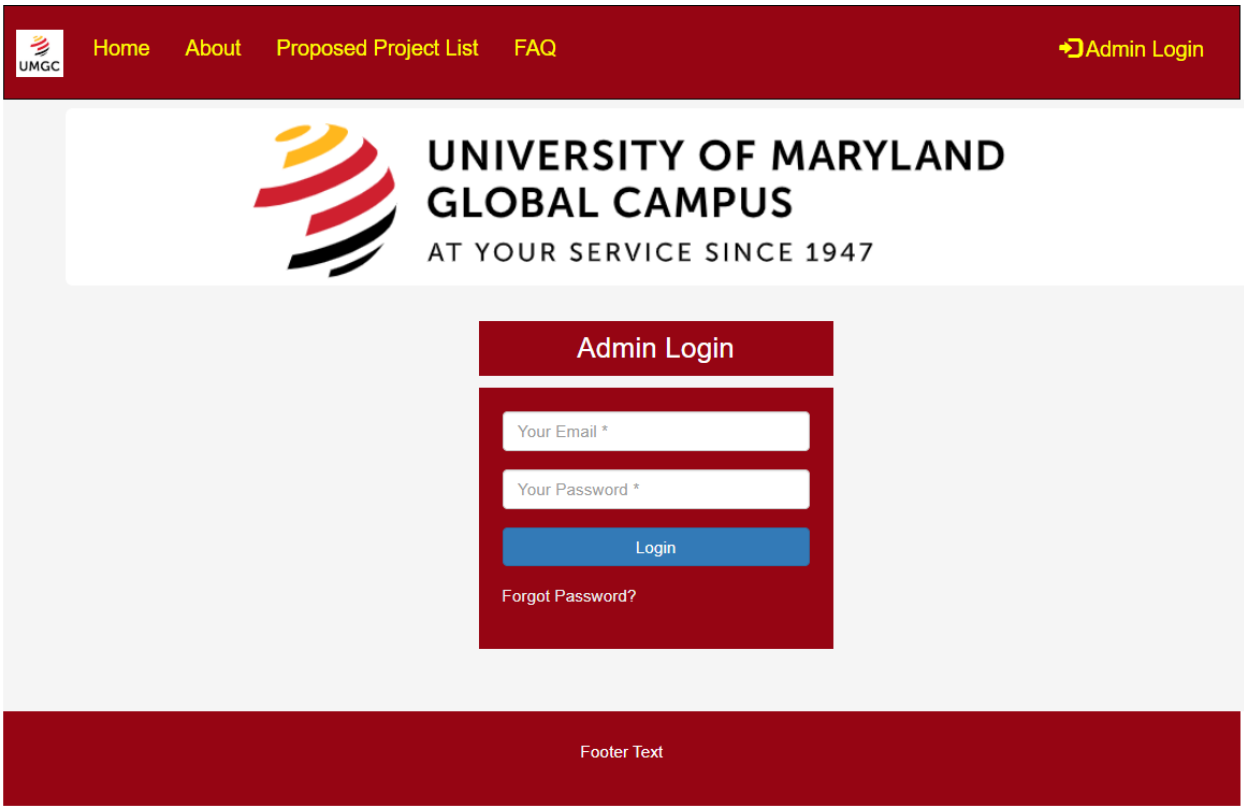

Figure 8 Login Page

Purpose:	The Login Page provides system access after verifying user credentials and allows users to reset a forgotten password.
Navigation & User Interaction:	From the top menu bar, the user clicks on the “Admin Login” button. Users will be prompted to enter valid credentials to access the system. The user clicks on the “Forgot password” to reset the password and recover their account.

6.1.5 Proposal list

Requirement ID: REQ-1.7, REQ-1.13, REQ-14, REQ-1.20

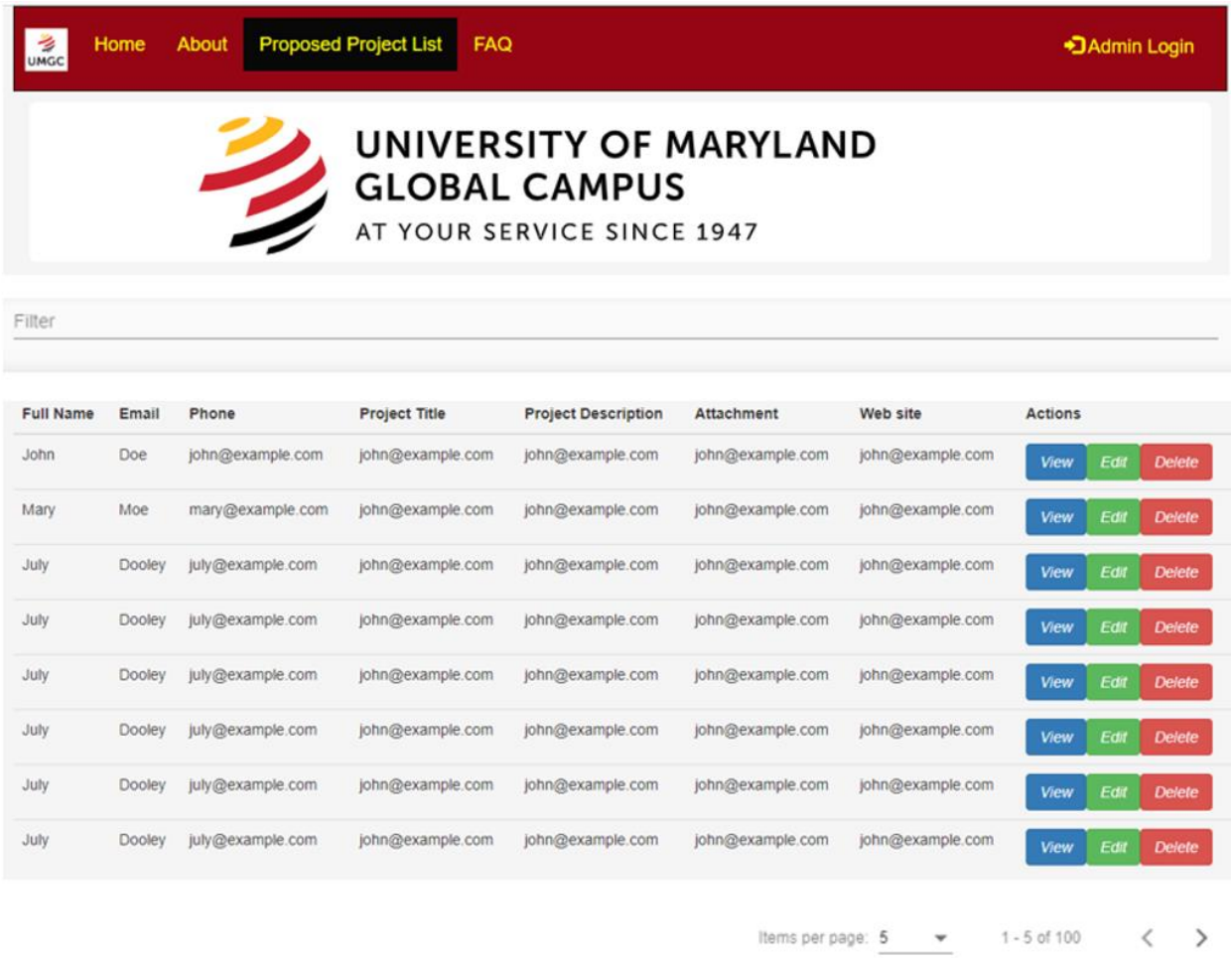

Figure 9 Proposal List Page

Purpose:	This page lists submitted project proposals and allows the user to search, filter, export, view, edit, and delete project proposals.
Navigation & User Interaction:	A logged-in user clicks on the “Proposed Project List” button to view the list of submitted proposals.

6.1.6 View

Requirement ID: REQ-1.12, REQ-1.15

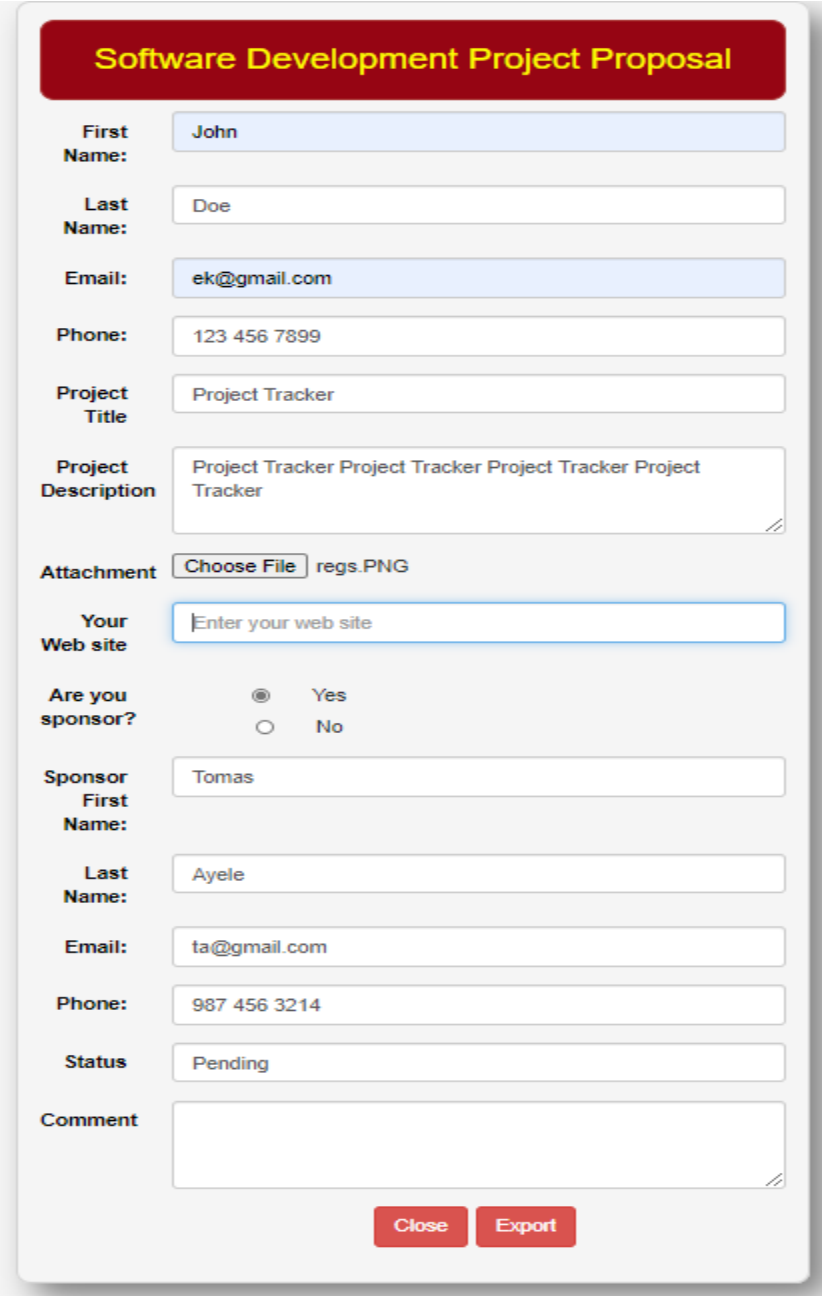

Figure 10 View Page

Purpose:	The “View” page displays submitted project proposal details.
Navigation & User Interaction:	From the grid table, the user clicks on the “View” button under the “Action” column. The user can view the submitted proposal in detail.

6.1.7 Edit

Requirement ID: REQ-1.16, REQ-1.18, REQ-1.19

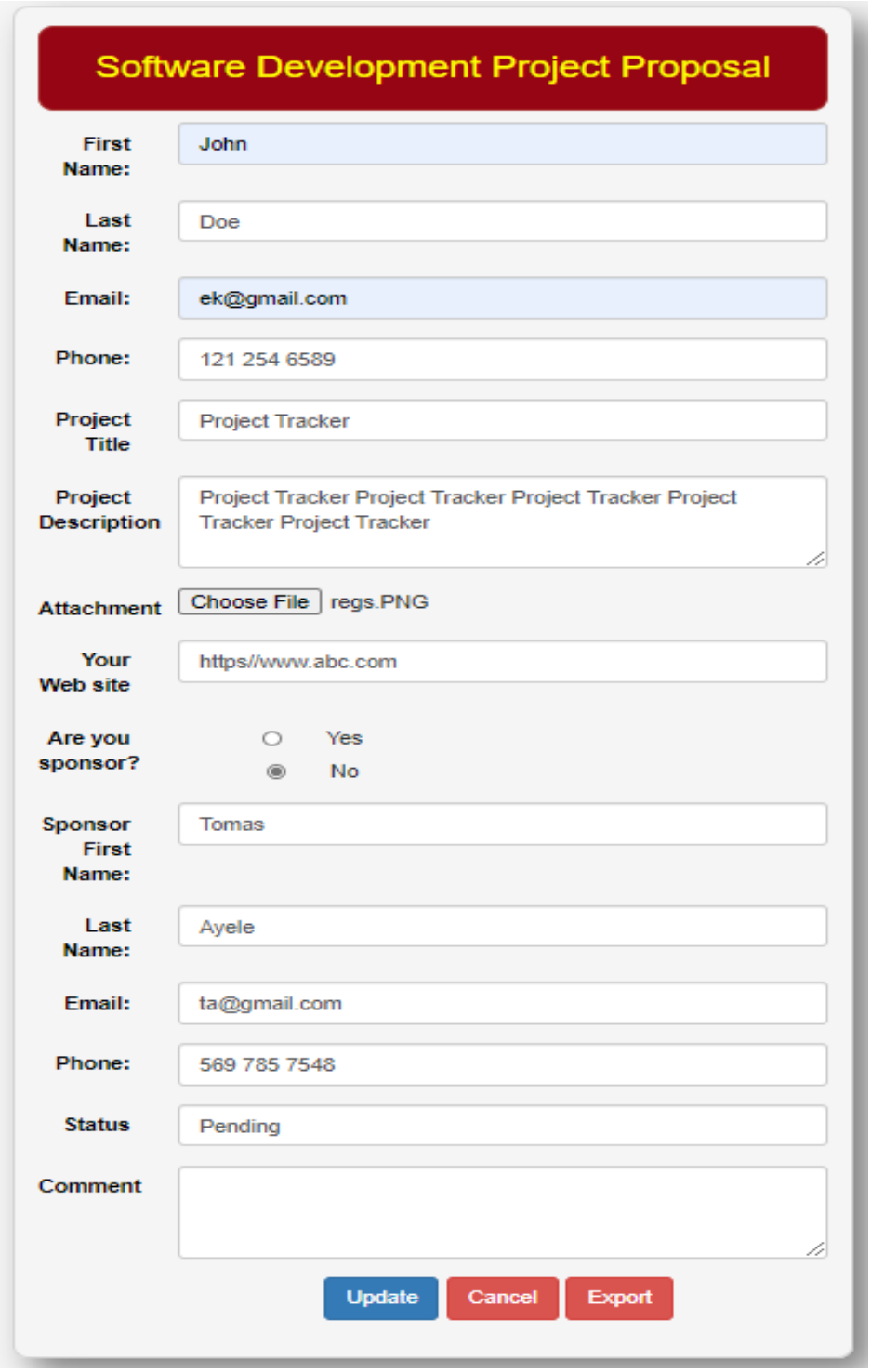

Figure 11 Edit Page

Purpose:	The "Edit" page allows the user to make changes to the submitted project proposals.
Navigation & User Interaction:	From the grid table, the user clicks on the "Edit" button under the "Action" column. The user can update (approve, reject, and add a comment) and save the changes from this page.

6.1.8 Register New Admin User

Requirement ID: REQ-1.9, REQ-1.10

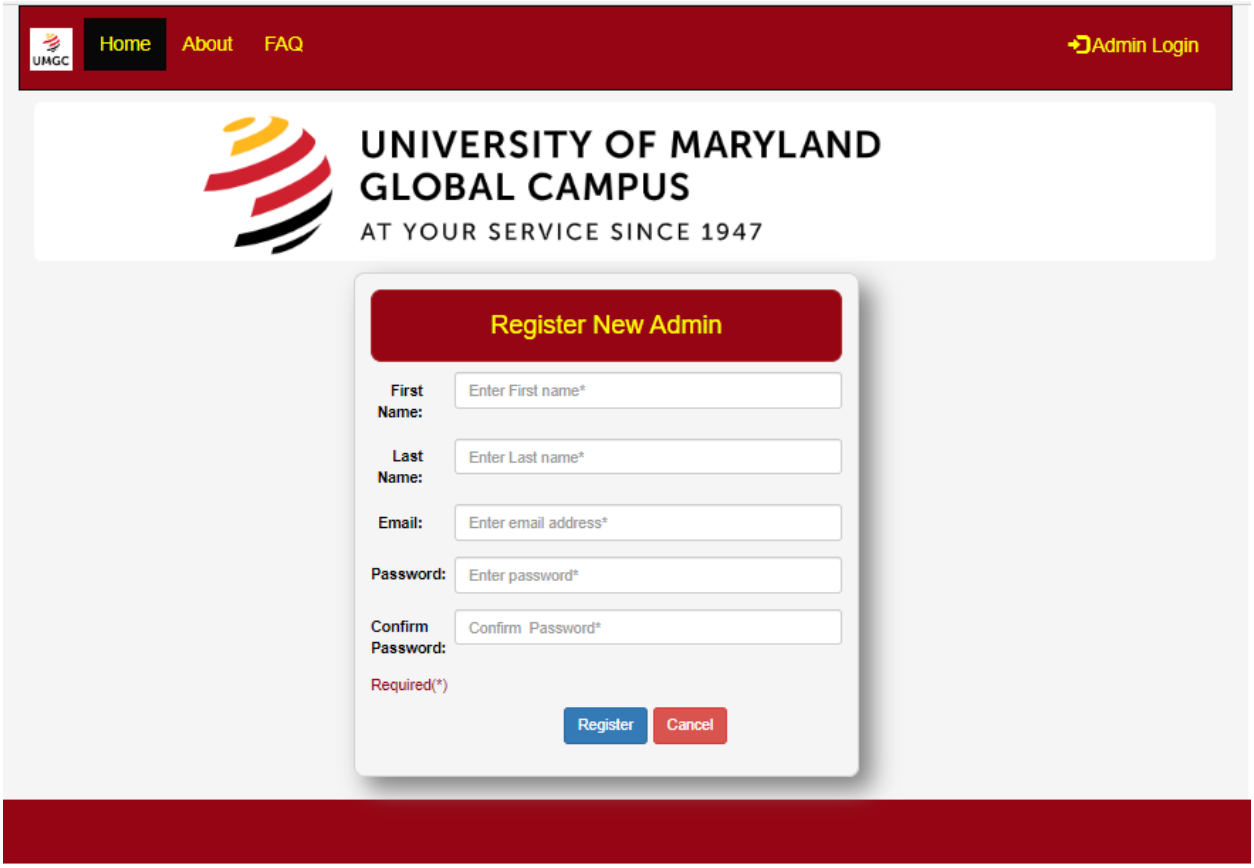

Figure 12 Register New Admin page

Purpose:	The Register new admin Page allows the admin to create a new admin account.
Navigation & User Interaction:	From the top menu bar, the user clicks on the "Register new admin" button. Users will be prompted to enter required information to add new admin user to the system.

7. Requirement Matrix

The following table is a requirements matrix for CaPPMS application. It is a cross reference table that links the technical design components to the SRS requirements.

Table 16 Requirement Matrix

Requirement _ID	Requirement Description	Section Reference
REQ-1.1	As an unauthenticated user, I want to access the SWEN 670 home page via the internet.	6.1.1
REQ-1.2	As an unauthenticated user, I want to learn about the SWEN 670 capstone project.	6.1.2
REQ-1.3	As an unauthenticated user, I want to submit a project proposal idea for consideration.	6.1.1
REQ-1.4	As an unauthenticated user, I want to cancel a project proposal idea before submission.	6.1.1
REQ-1.5	As an unauthenticated user, I want to receive notification that required data has not been entered.	6.1.1
REQ-1.6	As an unauthenticated user, I want to view frequently asked questions (FAQ) regarding the capstone project.	6.1.3
REQ-1.7	As an authenticated user, I want to view submitted project proposal ideas.	6.1.5
REQ-1.8	As the professor, I want an option to login to the system.	6.1.4
REQ-1.9	As the professor, I want to be able to create an additional authenticated account.	6.1.8
REQ-1.10	As the professor, I want to be notified if a new authenticated account already exists when creating.	6.1.8
REQ-1.11	As the professor, I want a way to reset my password.	6.1.4

REQ-1.12	As the professor, I want to login to view additional project details.	6.1.6
REQ-1.13	As the professor, I want to search based on project status, i.e. Approved.	6.1.5
REQ-1.14	As the professor, I want to filter based on project status, i.e. Approved.	6.1.5
REQ-1.15	As the professor, I want to view the details of a specific proposal.	6.1.6
REQ-1.16	As the professor, I want to add private comments to a specific proposal.	6.1.7
REQ-1.17	As the professor, I want to receive notification that required data has not been entered.	6.1.7
REQ-1.18	As the professor, I want to cancel edits made to a specific proposal.	6.1.7
REQ-1.19	As the professor, I want to edit the details of a specific proposal.	6.1.7
REQ-1.20	As the professor, I want to delete a specific proposal.	6.1.5
REQ-1.21	As the professor, I do not want the private comments to be included in the export.	6.1.7
REQ-1.22	As the professor, I want to be able to add a reason for rejection to a specific proposal.	6.1.7
REQ-1.23	As the professor, I want to export a specific proposal in Word format.	6.1.7

8. Reference

Nielsen, J. (1994, April 24). 10 Usability Heuristics for User Interface Design. Retrieved from https://www.nngroup.com/articles/ten-usability-heuristics/

University of Maryland Global Campus (n.d.). UMGC Logo and Brand Standards. Retrieved from https://www.umgc.edu/documents/upload/UMGC-Logo-and-Brand-Standards.pdf