QUICK PLAN
GROUP 6

Caroline Roberson, Ian Riggins, Riley Powell, Zachary Powell, Montgomery Russell, and Sotheary Pong

PROFESSOR POURNAGHSHBAND
SWE 4663, SECTION W01
SEPTEMBER 29, 2024

Table of Contents

Team Information

Caroline Roberson

Project Manager, Lead Technical Writer, and Editor

crobe215@students.kennesaw.edu

Contribution: A

Ian Riggins

Lead Programmer and Repo Manager

iriggins@students.kennesaw.edu

Contribution: A

Riley Powell

Technical Writer

rpowel57@students.kennesaw.edu

Contribution: A

Monte Russell

Programmer and UI Designer

mrusse80@students.kennesaw.edu

Contribution: A

Zachary Powell

Programmer

Zcoile@students.kennesaw.edu

Contribution: A

Sotheary Pong

Programmer

spong@students.kennesaw.edu

Contribution: A

All team members had satisfactory performances during this deliverable.

Introduction

Project Description

Author: Caroline

Project management is a ubiquitous field. Every industry will have complex projects involving a large number of people and moving parts. Careful management of time, money, and manpower is essential for a successful project. Project management systems, like the application described in this document, are a major asset for any organization.

Our project management system will allow the user to track projects over the course of the project's lifespan, including ongoing and completed projects. The system will store information such as project requirements, time spent on sections of the project, the people associated with the project, risks, and more. The system will allow the user to create new projects, view existing projects, and update the details at any time.

Target Market

Our application is targeted towards smaller organizations, such as those with fewer than 100 employees. Our application will offer functionality comparable to popular alternatives like Trello or Jira, which are largely web-based and inaccessible if a user cannot connect to the Internet. Industries that require work in locations with slow or non-existent Internet connection, such as remote locales or overseas, will especially benefit from a product specifically designed for use in those situations. Although our application can be hosted remotely over the Internet, it can also be hosted locally to a single machine, allowing isolated users to continue their work while away from home or office.

User Types

The primary user base for this application includes users from professional and non-professional fields. The application is not targeted towards any specific industry and can be used for any situation involving a project with requirements and other tracking details. Additionally, the application is a viable tool for any organization size, including larger enterprise organizations with more than 1,000 employees, smaller organizations with 100 employees or fewer, and users who are working independently. The two primary user types are described below:

- **Project Managers**: use this application to create new projects, track the progress of ongoing projects, and manage hours for a large number of team members
- **Team Members**: use it to view project requirements and log activity

Vision Statement

For project managers who are looking to improve their productivity, the application is a project management system that will provide a central database to store and display information pertaining to projects, team members, and requirements. This application will facilitate tracking project details such as requirements, progress updates, and team member activity by providing a simple-to-use interface between user and database. Unlike many cloud-based alternatives, our product will function as a localized tool to be used in any setting, regardless of connectivity.

Operating Environment

OPE-1 The system shall have its web server and database hosted via the latest stable version of Blazor Server.

OPE-2 The website system shall be able to run at optimal performance on computer systems with a minimum of 64MB of RAM and 70Mb of disk space that is running Windows 7 or later.

OPE-3 The website system shall run on displays that support 720x1280 up to 1920x1080.

OPE-4 The website system shall operate as intended on the following web browsers:

- Google Chrome (versions: 1.0.154, 2.0.172, 3.0.195, and 4.0.249)
- Apple's Safari Browser (versions: 11.1.2, 12.1.2, 13.1.2, and 14.1.2)
- Microsoft Edge (Versions: 107.0.1418.42, 107.1418.36, and 107.0.1418.28)
- Firefox (Versions: 106.0.5 and 106.1.0, and 106.0.2)

OPE-5 The system shall contain the following APIs: NUnit and iTextSharp.

Quality Attributes

External Quality Attributes

Authors: Caroline, Sotheary

Usability – High Priority

Ease of use is the most critical aspect of the system as it will be utilized by a wide variety of users with varying backgrounds, personal history with project management systems, and ranging ability. Users must be able to quickly and easily set up and begin using the system with little to no training or prior experience with similar applications and encounter minimal difficulty due to disability.

USA-1 The system shall display all textual UI elements at a minimum font size of 12.

USA-2 The system shall order context actions (such as cancel or save buttons) in a consistent order across the entire application.

USA-3 The system shall clearly label all interactive elements with a symbol or textual label to indicate their use.

USA-4 The system shall not contain any UI elements that solely rely on color to indicate their use.

USA-5 The system shall allow the user to return to the main menu at any time, from any screen.

USA-6 The system shall allow the user to return to the previous page immediately unless the action would result in data loss due to uncommitted database changes.

Integrity – Medium Priority

Integrity is a significant element of the system since it will be handling frequent database modifications with no backup available on the cloud. The system must be able to reliably commit changes and retain temporary data in the event of a failure on the part of the system or the supporting hardware.

INT-1 The system shall commit changes to the database only once the user has selected to save the changes and confirmed their decision.

INT-2 The system shall use transactions to ensure that all modifications to the database either fully succeed or fail without partial updates.

INT-3 The system shall perform regular checks to detect and fix any database corruption.

INT-5 The system shall retain temporary data in memory, enabling recovery in case of an unanticipated system crash or shutdown.

Internal Quality Attributes

Authors: Caroline, Sotheary

Maintainability – Low Priority

Code maintainability is an important quality attribute given the system will be constructed, expanded upon, and updated by multiple programmers over the course of several weeks. Good code writing standards must be upheld to reduce resources wasted on untangling poorly written code.

MNT-1 The system shall follow industry-standard coding practices and conventions.

MNT-2 The system shall utilize GitHub as a version control system to track change history.

MNT-3 The system shall be subject to mandatory code review ahead of merging changes into the primary branch of the version control system.

MNT-4 The system shall contain unit tests sufficient to cover at least 90% of the code base.

Efficiency – Low Priority

Database efficiency is an important attribute because the system is expected to quickly handle reading and writing large amounts of text and to perform mathematical operations at a sufficiently high rate of speed as to not disrupt the user experience.

EFF-1 The system shall complete and return search queries in less than 3 seconds.

EFF-2 The system shall use caching to minimize retrieving and processing data.

Project Requirements

Author: Riley Powell

1. Create Project Details

1.1 Input Project Description

1.1.1 The system must provide a text area for the user to input a description of the project.

1.1.2 The system must allow the user to input a description of up to 5000 characters.

1.1.3 The system must display an error if the user attempts to input more than 5000 characters.

1.1.4 The system must commit changes to the project description to the database once prompted by the user.

1.1.5 The system shall allow the user to modify the description at any time.

1.1.6 The system must display a success message " Description saved successfully" once the save operation is completed.

1.2 Input Project Owner's Name

1.2.1 The system must provide a text area for the user to input the project owner or manager's name.

1.2.2 The system must allow input of a name of up to 50 characters.

1.2.3 The system must validate the name field to ensure it is not blank.

1.2.4 The system shall allow users to update the name of the owner or manager at any time.

1.3 Input Project Members

1.3.1 The system must provide an interface for the user to add, remove, or update the list of members assigned to a project.

1.3.2 The system shall allow the user to input a name of each member with a max of 50 characters.

1.3.3 The system must allow the user to input a description of each member's role with a max of 150 characters.

1.3.4 The system shall display an error message if the character count for names or description is above the set limit.

1.3.5 The system must store the list of members in the database including their roles.

1.3.6 The system shall allow the user to add members at any time.

1.3.7 The system shall allow the user to delete members from the list at any time.

1.3.8 The system shall allow the user to update member details on the list at any time.

1.4 Input Risks and Risk Status

1.4.1 The system must provide an interface for the user to add, delete, or modify risks.

1.4.2 The system must allow the users to input a textual description of each risk, with a max of 350 characters.

1.4.3 The system shall display an error if the user goes over max character limit.

1.4.4 The system must provide an interface for the user to select the status of the risk from predefined options (Active, Mitigated, Resolved).

1.4.5 The system must store the list of risks in the database.

1.4.6 The system shall allow the users to modify their risks and statuses.

1.4.7 The system should display a list of risks and their statuses when the user requests them.

1.5 Assign Project Start and End Dates

1.5.1 The system must provide a date picker for the user to select the start of the project.

1.5.2 The system must provide a date picker for the user to select the end of the project.

1.5.3 The system must validate that the project end date is not earlier than the start date.

1.5.4 The system must store the project start and end dates in the database once selected.

1.5.5 The system shall allow the user to modify the start and end dates to the project at any time.

1.6 Assign Project Status

1.6.1 The system must provide a menu for the user to set the status of the project from predetermined options (Not started, In Progress, On Hold, Completed).

1.6.2 The system must store the project status in the database.

1.6.3 The system shall allow the user to modify the project status at any time.

1.6.4 The system should display a project's status on the main screen of the project.

1.7 Project Priority

1.7.1 The system must provide an option for the user to set the priority of the project from predetermined options (High, Medium, Low).

1.7.2 The system must store the project priority in the database.

1.7.3 The system shall allow the user to modify the priority of the project at any time.

1.7.4 The system should display the project priority on the main screen.

2. Project Requirements

2.1 Input Functional Requirements

2.1.1 The system must allow the user to add their functional requirements.

2.1.2 The system must allow the user to input a textual description for each functional requirement, with a max character count of 1000.

2.1.3 The system shall send an error if the description exceeds 1000 characters.

2.1.4 The system must store each functional requirement in the database.

2.1.5 The system shall allow the user to modify functional requirements at any time.

2.1.6 The system should allow the user to view the list of functional requirements.

2.1.7 The system must allow the user to assign a reference number when making a new requirement.

2.1.8 The system must allow the list of requirements to be sorted by reference number.

2.1.9 The system must allow each requirement to be assigned to an owner (Needs owners first and last name to assign).

2.2 Input Non-Functional Requirements

2.2.1 The system must provide an interface for the user to add their non-functional requirements.

2.2.2 The system must allow the user to input a textual description for each non-functional requirement, with a max of 1000 characters.

2.2.3 The system shall send an error if the description exceeds 1000 characters.

2.2.4 The system must store each non-functional requirement to the database once the user has saved it.

2.2.5 The system shall allow the user to modify nonfunctional requirements at any time.

2.2.6 The system shall allow the user to view the entire list of non-functional requirements.

2.2.7 The system must allow the user to assign a reference number when making a new requirement.

2.2.8 The system must allow the list of requirements to be sorted by reference number.

2.2.9 The system must allow each requirement to be assigned to an owner (Needs owners first and last name to assign).

2.3 Requirements Prioritization

2.3.1 The system must allow the user to assign a priority level for each requirement (High, Medium, Low).

2.3.2 The system must store the requirement’s priority in the database.

2.3.3 The system shall allow the user to modify the priority at any time.

2.3.4 The system should display the requirements based on priority level.

3. Project Effort Monitoring and Tracking

3.1 Input Effort by Activity

3.1.1 The system must provide an interface for the user to input the hours expended on each activity within the project (requirements analysis, coding, designing, testing, management) on a daily or weekly basis.

3.1.2 The system must allow the user to associate each activity with a specific functional or non-functional requirement.

3.1.3 The system must allow the user to enter a date on when the hours were spent.

3.1.4 The system must validate that each time entry is non-negative.

3.1.5 The system must validate that no time entry exceeds 24 hours in a single day per member.

3.1.6 The system must store all effort entries in the database.

3.1.7 The system shall allow the user to modify the effort record at any time.

3.2 Display Total Expended Hours

3.2.1 The system must provide a summary view of the total hours expended per activity.

3.2.2 The system must break down the total hours by specific activities, this includes (requirements analysis, coding, design, testing, and management).

3.2.3 The system must allow the user to view the total expended hours for each functional and non-functional requirement.

3.2.4 The system shall allow the user to filter the hours based on member names, requirement, or activity type.

3.3 Track Team Member Contributions

3.3.1 The system must allow the user to track and display the total hours contributed by each member across all activities (requirements analysis, coding, designing, testing, management).

3.3.2 The system must provide a summary showing all the hours contributed.

3.3.3 The system should allow the user to filter through the contributions based on member names, activities, or requirements.

3.4 Automated Effort Alerts

3.4.1 The system must allow the user to set effort thresholds (This will be a predetermined number of hours set on each activity. This number can be exceeded).

3.4.2 The system must send an automated alert to the user if the hours on a requirement or activity exceed the threshold.

3.4.3 The system must provide a notification system to display the alerts.

3.4.4 The system shall allow the user to modify the threshold at any time.

3.4.5 The system must allow the user to set tasks under a “watch” (This will notify all members associated with the task).

3.5 Task-Level Tracking

3.5.1 The system must allow the user to create tasks under each requirement to break down work further**.**

3.5.2 The system must allow the user to track the number of hours expended on each task in addition to overall requirement-level tracking**.**

3.5.3 The system must calculate and display the cumulative hours spent on all tasks under a specific requirement.

3.5.4 The system must store task-level effort tracking data persistently in the database.

3.5.5 The system shall allow the user to update or delete individual task records.

4. Reporting and Documentation

4.1 Generate Effort Reports

4.1.1 The system must allow the user to generate effort reports showing the total hours expended for each requirement, activity, and member.

4.1.2 The system must allow the user to export the effort reports in either a PDF or CSV format.

4.1.3 The system should allow the user to filter through effort reports by a time period (daily, weekly, monthly, yearly).

4.1.4 The system must generate effort reports in real time(24h time) based on the most recent data entered into the system.

4.2 Generate Risk Reports

4.2.1 The system must allow the user to generate a risk report which shows all current risks for the project and their statuses.

4.2.2 The system shall allow the user to export risk reports via a PDF or CSV format.

4.2.3 The system should allow the user to filter through risk reports by their statuses (Active, Mitigated, Resolved).

4.3 Risk Mitigation Strategies

4.3.1 The system must allow the user to associate mitigation strategies with each project risk.

4.3.2 The system must allow the user to input and store textual descriptions of the mitigation strategies, with a maximum of 1000 characters.

4.3.3 The system shall allow the user to update or delete mitigation strategies for any risk.

4.3.4 The system should display the mitigation strategy alongside each risk in the risk report.

5. Security and Permissions

5.1 User Roles and Permissions

5.1.1 The system must provide role-based access control, allowing different levels of access for project owners, team members, and stakeholders.

5.1.2 The system must allow the project owner to assign roles to each team member.

5.1.3 The system must restrict access to certain functionalities (e.g., risk management, effort tracking) based on the user's role.

5.1.4 The system must store user roles and permissions persistently in the database.

6. Usability Features

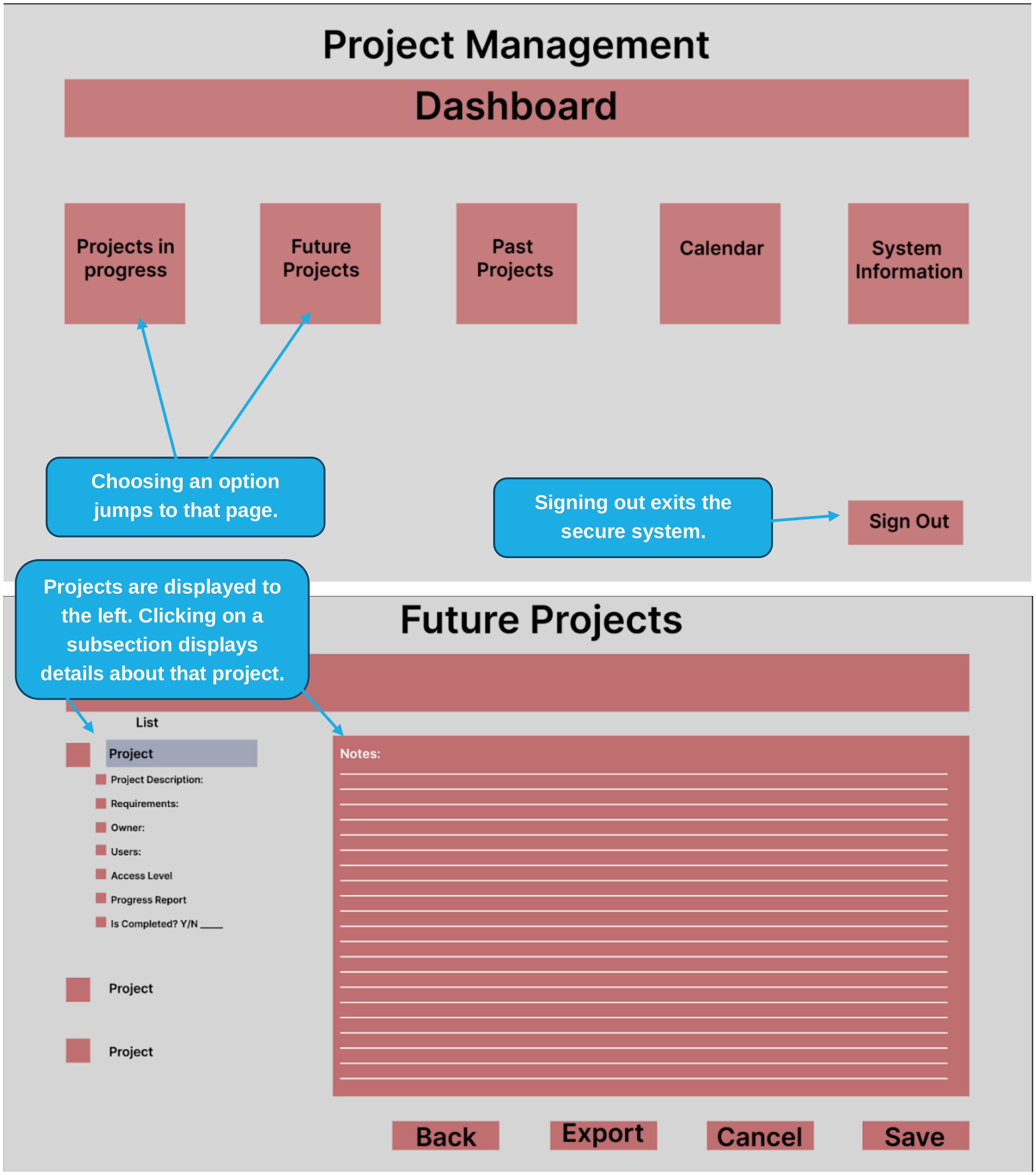
6.1 Search and Filter

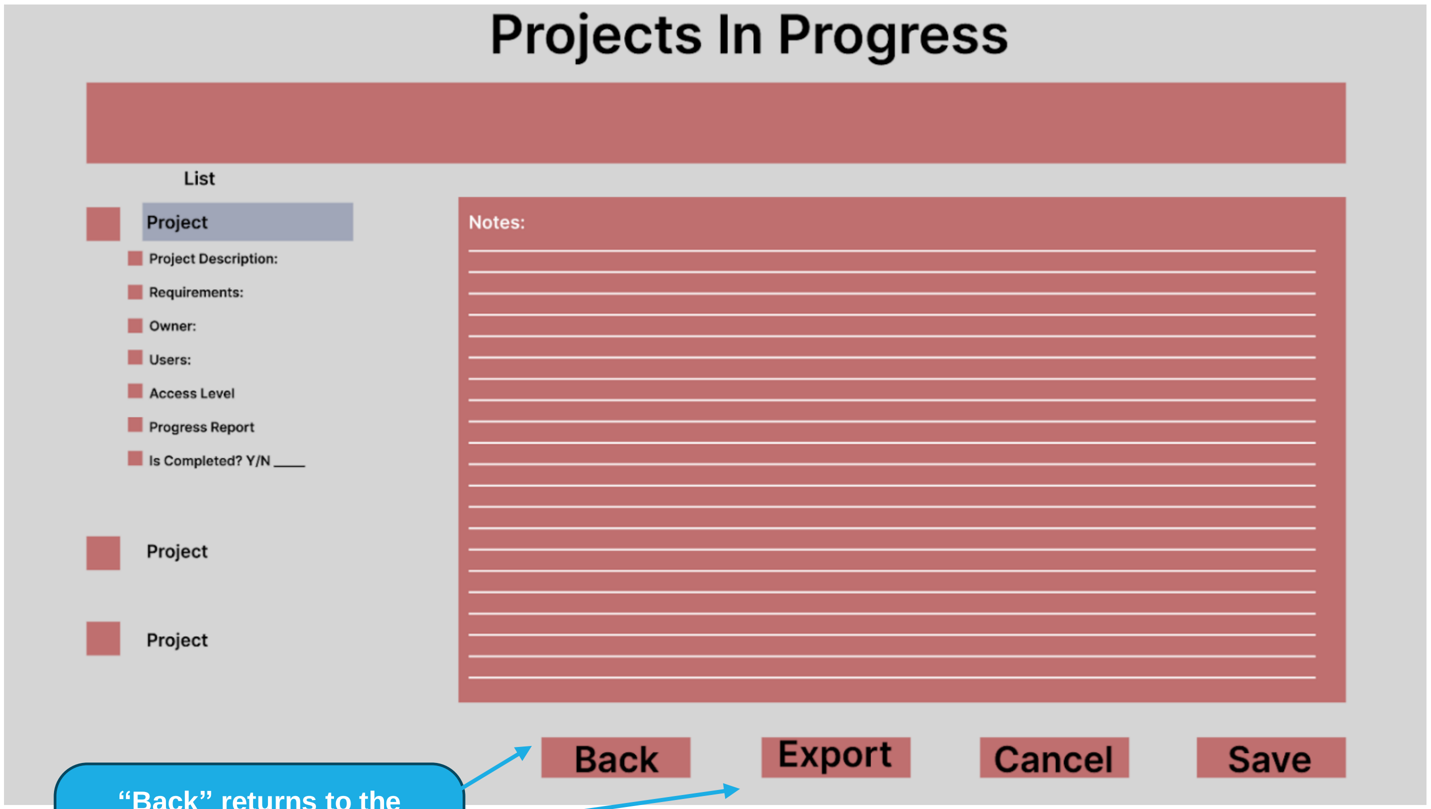
6.1.1 The system must provide a search bar allowing users to search for specific projects, team members, or requirements.

6.1.2 The system should provide filters to view requirements based on priority, status, or type (functional/non-functional).

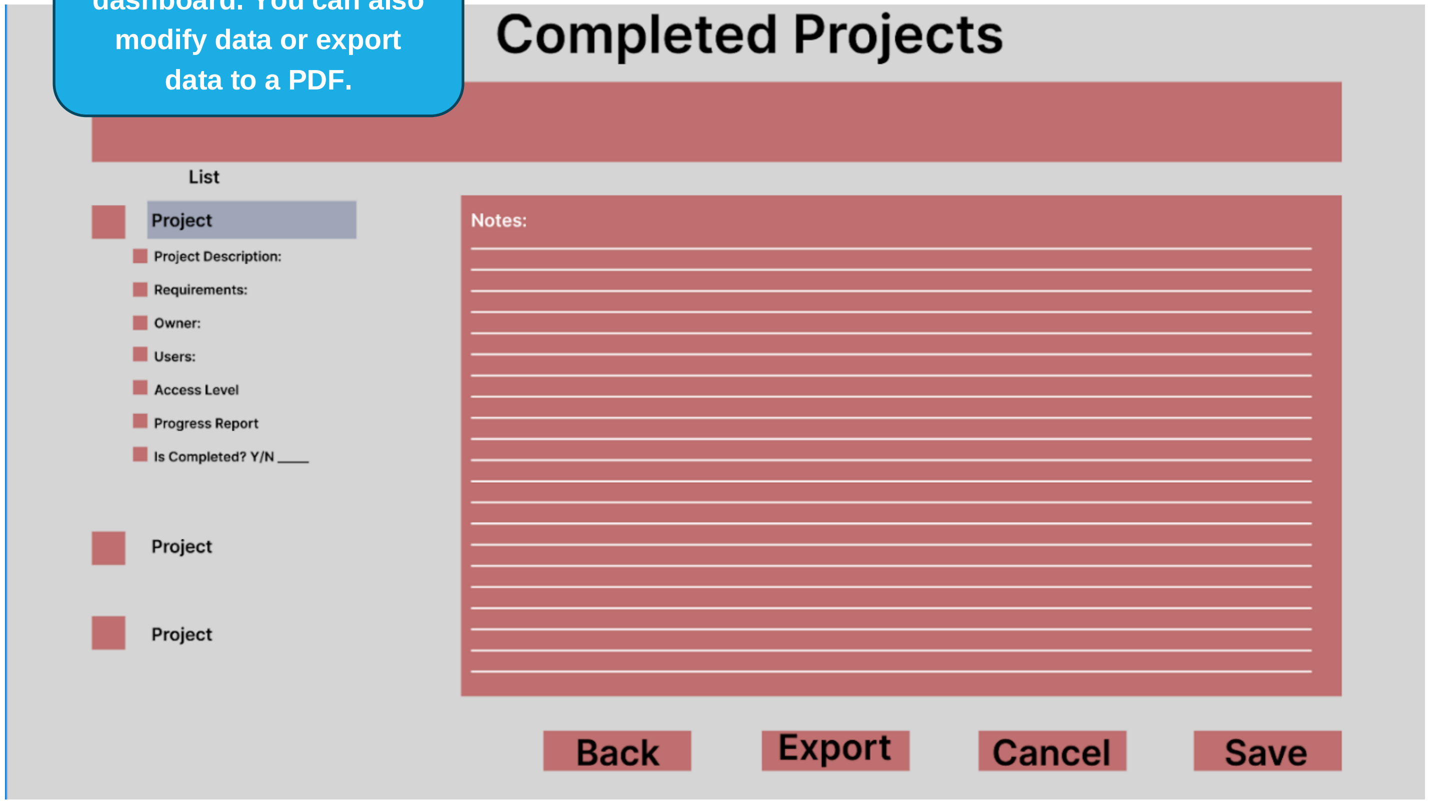
6.1.3 The system must display filtered and search results in real-time.

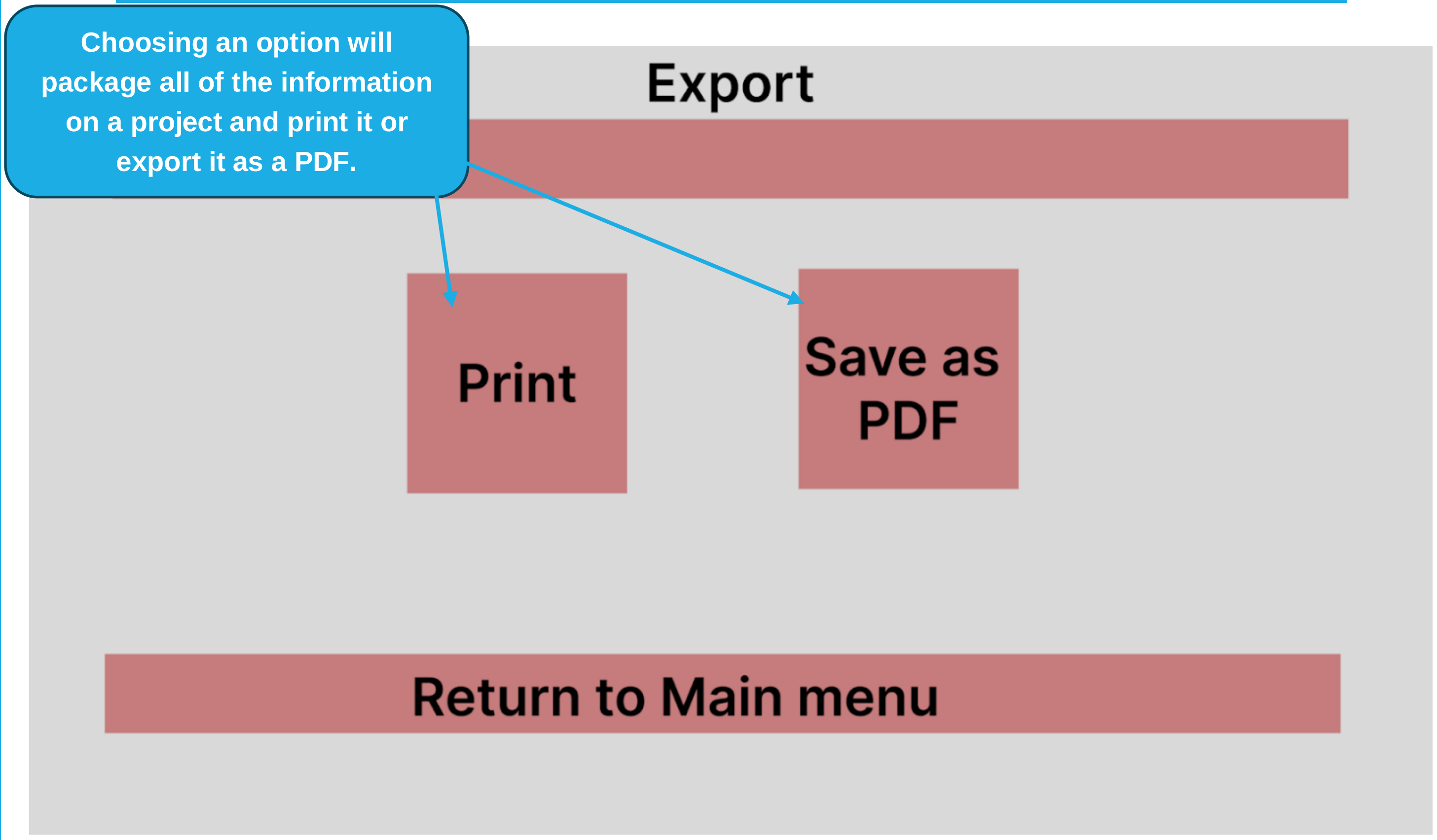
Initial UI Prototype





“Back” returns to the dashboard. You can also modify data or export data to a PDF.




Choosing an option will package all of the information on a project and print it or export it as a PDF.
Export
Print
Save as PDF
Return to Main menu

Resource Assessment

Authors: Caroline, Ian

Technical Writing Estimate

Out of our team composed of six individuals, we have two technical writers with a mixture of academic and limited professional experience. Based on past projects worked on by our team's writers, two writers should be more than sufficient to complete the major written deliverables of this project along with any supporting technical documents within a reasonable timeframe.

The major written components of this document **were completed within a two-week span**, so it is a reasonable estimate that the second deliverable will be completed within a similar time period, with a **margin of error of one week or less**. Should our technical writers lack any technical knowledge of the application to finish detailing part of the deliverable, they will consult with our programmers to finish the section.

Technical Writers: Caroline Roberson (Lead), Riley Powell

Implementation Estimate

We have four programmers capable of handling the implementation stage of the project, with approximately half possessing at least some professional programming experience. This project is estimated to contain at least nine separate classes that will need to be developed, each with their own unit tests.

There will be several phases of development. Implementation will take between **1-2 weeks**, overlapping with unit testing, which will also take between **1-2 weeks**. Integration testing will begin afterwards along with web development, which will take approximately **one week to ten days**. The final stage, linking the web server to the database will take an additional **one week to ten days**. Based on similar projects completed by our team in academic and professional environments, full development is estimated to take at least four weeks and at most six weeks. This estimate has a **margin of error of one week or less** in the event of encountering risks or circumstances outside of our control. In the event that four programmers are not sufficient to complete the project on time, our two writers will act as reserves.

Programmers: Ian Riggins (Lead), Zachary Powell, Monte Russell, Sotheary Pong

Risk Assessment

Authors: Caroline, Sotheary

Time Miscalculation

Description: May occur as a result of an over- or underestimation of the time required for a task, leading to the overall schedule becoming skewed. Continued time miscalculations may lead to cascading problems with the scheduling. Careful monitoring of scheduling is critical for success.

Priority: High

Mitigation Plan:

- Subdivide high-level tasks into atomic subtasks
- Review and update timeline regularly, adjusting as needed
- Generously estimate time required for important tasks
- Leave a buffer zone between periods of implementation and testing for last-minute adjustments

Team Miscommunication

Description: May occur due to inadequately explained requirements, project scheduling, or other parts of the project; may also occur as a result of poor communication between team members regarding personal schedules.

Priority: Low

Mitigation Plan:

- Maintain continuous, timely communication using team chat channels on Discord
- Schedule all-hands meetings to provide immediate, team-wide updates as needed
- Require all team members to thoroughly review project documents and communicate concerns or confusion

Skill Failure

Description: May occur as a result of a team member being unable to adequately perform or complete a task due to lacking the necessary skill; mitigation is especially difficult due to unavoidable constraints on available resources.

Priority: Medium

Mitigation Plan:

- Discuss each team member’s skill set and knowledge early on
- Carefully consider a member’s strengths and past performance before assigning a new task
- Check in often with team members to assess their progress and provide assistance
- Reassign tasks as needed if a team member is failing to perform adequately
- Encourage team members to develop their skill and knowledge whenever possible

Major Deliverables

Comprehensive Plan

Includes all of the content contained in this document with additional detail; specifically, a complete list of functional and nonfunctional software requirements, the finalized work schedule for implementing the finished product, and a thorough description of the product features.

Delivery date: October 20th, 2024

Final Product

Implements all of the functionality described in the Comprehensive Plan in full working order. The application functions as a standalone executable and can be installed and used by a non-technical user. Some minor bugs may be present but will have minimal impact on the user experience.

Delivery date: December 1st, 2024

Video Demonstration

Demonstrates the major functions of the final product, the user interface, and common use cases described in the Quick Plan and Comprehensive Plan.

Delivery date: December 1st, 2024

Schedule

The tentative project work schedule for full implementation can be found here:

https://docs.google.com/spreadsheets/d/1yz0svYtf70q_ehntwFf4FjGlw_iPV8faaxVrc4LN32g/edit?usp=sharing