Group 4

Professor Sungchul Jung

Software Project Management – SWE 4663 Section W01

Kennesaw State University – Spring 2025

ManageMe.io – Comprehensive Plan

Maan Bhagat, Matt Crowley, Kade Fleming, Constant Nortey, Caroline Varner

Kennesaw State University

1100 South Marietta Parkway SE, Marietta, GA, USA, 30060

Table of Contents

Group Member Contributions

Maan Bhagat – Major External Deliverables – A

Matt Crowley – Group Leader – Risks and Assumptions, High Level Schedule, Summary – A

Kade Fleming – Non-Functional Requirements, Detailed Product Description – A

Constant Nortey – Functional Requirements, Detailed Project Description – A

Caroline Varner – Introduction – Rough Cost Estimate – A

Introduction

Our product, ManageMe.io, is an innovative task management application built using ASP.NET that will aid users in enhancing their productivity, workflow, and team collaboration. Our application seeks to provide a user-friendly interface coupled with essential project management tools to increase overall productivity management and efficiency. Some examples of these tools would be a task assigner, a progress tracker, and real-time updates for task completions/additions. The task assigner would allow project managers and team members to assign tasks more efficiently. A progress tracker would users efficiently visualize the status of ongoing tasks and the big picture project milestones. Lastly, the real-time updates would help ensure that all team members on a project stay updated about new tasks, completed work, and project changes.

We guarantee that our team is committed to developing an outstanding solution with a structured and flexible workflow. Two weekly sessions were established, which are a virtual session on Tuesday for progress reporting and an on-site session on Thursday for hands-on collaboration in the Marietta campus. There is also assigned individual work on Saturdays and Sundays to ensure every deliverable smoothly completed.

This report outlines our development schedule, cost analysis, and technical deployment plan. Specifically, our initial estimate of cost provides a split of estimated costs, such as human resource costs (e.g., planning time, development time, testing time) and software/tools costs (e.g,. Cloud hosting). Using open-source tools and efficient task delegation, we aim to develop a robust, scalable, and low-cost project management tool responsive to user needs without going over budget.

Detailed Technical Description

To take a deeper dive into this project, we will go over the development process to be used as well as the technologies.

For our project management framework, we will be following the Agile methodology, specifically, **Kanban**, where we will outline tasks that will be picked up by an engineer at a time, in this case, Software Engineering students. Over the next month, we will have everyone moving back and forth between tasks relating to *Frontend* and *Backend.*

As mentioned in our introduction, we will be utilizing the **ASP.NET** framework which implement the **Model-View-Controller (MVC)** pattern, where we will be using **HTML/CSS (Frontend), C#** and **JavaScript (Backend)** to develop this web application. **Figma** will be utilized in designing and perfect our User Interface and testing User Experience. We will host the website using **Amazon Web Services (AWS)** and store information using **MongoDB**. For testing, we will utilize **NUnit** for logical operations and **Playwrigh**t for more visual tests for making sure the right page is displayed.

In summary:

Frontend: HTML/CSS

Backend: ASP.NET, C#, JavaScript NUnit, Playwright

Database: MongoDB

Deployment: Hosted on AWS

Risks and Assumptions

With a project of this large of a scale, the abundance of unmonitored risks will likely cause the project to fail. As a group and team, we must outline these risks and assumptions and reduce them as much as possible. This section regards the most prominent risks the team believes there to be, along with the assumptions of how this project be completed.

With this project, the team is assuming many things. We assume that by the end of the project's development, the system will perform exactly how we plan it will. We can ensure this by checking off the requirements as we develop the system. Another assumption that we have about the project is that our cloud-based platform will remain operational during the development and testing of this system. The team also assumes that each deliverable will be completed by the due date of each deliverable/sprint. Along with assumptions of how the system will behave, the team also assumes the risks involved with completing this project. The next paragraphs will concern the risks that the team assumes there to be.

The first risk that we as a team need to account for is the risk of individual team performance. We are a team of five people in total. We have outlined our schedule to account for five individuals providing work towards the completion of the project. Although we as a group trust each other and are working diligently, there is a possibility that one or more group members does not meet performance expectations. In the event of this, the rest of the team members will need to provide extra work in the form of time and effort in order to still complete the deliverable in time. Another situation where this risk will become reality is if one or more members of the group drops the course, causing the rest of the group members to meet again and restructure our work allocation.

The second risk the team must account for is not understanding the software we are to build well enough. To combat this risk from becoming reality, we must clearly outline each and every requirement for the system that we are building. This will minimize the possibility of team members not understanding the system. If we do not properly plan for this, there will likely be a delay in the completion of each project deliverable because tasks that we believe will take a short amount of time might end up taking a large amount of time.

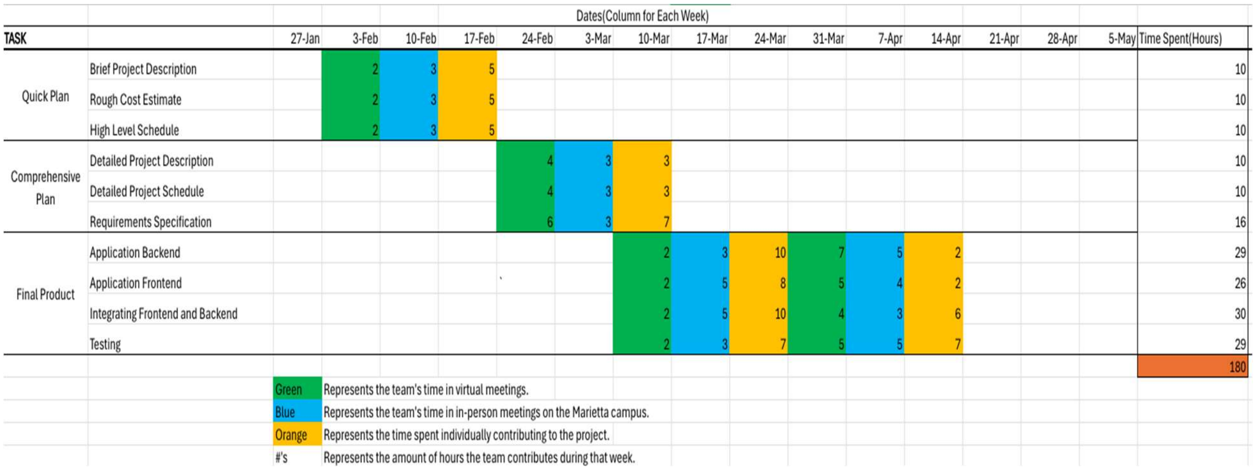
The third risk that the team must account for in the completion of this project is the changing of the project at a micro level or a macro level. Even the tiniest number of changes to the requirements need to be planned carefully. Although the group assumes that the project will not drastically change from our stakeholders, we must plan for the event of the requirements being altered. If the team decides to alter the system, we must talk about it and agree to the changes being made. Along with full team agreement, the team must alter the schedule to complete the altered tasks.

The fourth and final primary risk to this project the team will account for is the risk of not enough testing done. Throughout the development of the project, the team will need to test each function of the system to ensure that it aligns with the requirements. If the team fails to test and ends up neglecting it, the chance of the project failing will increase drastically. We have decided that it is best for us to test each function upon completion of the function, and then a full system testing phase once the entire system is completed. If we test properly and adequately, we will ensure that the system abides and completes each of the requirements.

High Level Schedule

Like any software project, there must be a schedule to follow to ensure that the project runs smoothly. A project of this scale is no exception. This section of the document will highlight the delivery of key milestones and deliverables, and the plan/schedule that our group plans to tackle each one with.

There are three primary deliverables for this project: including the quick plan, comprehensive plan, and the final project submission. Within each of these primary tasks/deliverables, there are subtasks that the team must account for and create a schedule for. Conclusively, the team decided to implement a Gannt chart to better understand the timeline and resource allocation for each task/subtask. Under this paragraph is the Gannt chart that we created for this project, and the next paragraphs will be outlying our schedule in detail for the completion of the system.



		Dates(Column for Each Week)															
TASK		27-Jan	3-Feb	10-Feb	17-Feb	24-Feb	3-Mar	10-Mar	17-Mar	24-Mar	31-Mar	7-Apr	14-Apr	21-Apr	28-Apr	5-May	Time Spent(Hours)
Quick Plan	Brief Project Description		2	3	5												10
	Rough Cost Estimate		2	3	5												10
	High Level Schedule		2	3	5												10
Comprehensive Plan	Detailed Project Description					4	3	3									10
	Detailed Project Schedule					4	3	3									10
	Requirements Specification					6	3	7									16
Final Product	Application Backend							2	3	10	7	5	2				29
	Application Frontend					`		2	5	8	5	4	2				26
	Integrating Frontend and Backend							2	5	10	4	3	6				30
	Testing							2	3	7	5	5	7				29
																	180
		Green	Represents the team's time in virtual meetings.														
		Blue	Represents the team's time in in-person meetings on the Marietta campus.														
		Orange	Represents the time spent individually contributing to the project.														
		#'s	Represents the amount of hours the team contributes during that week.														

Our team will meet 2 times a week until we have submitted the final version of our project. We will meet for an hour every Tuesday, and then again for another hour for an in-person meeting on the Marietta campus. Along with these two weekly meetings, we are designating every Saturday and Sunday as our individual workdays to put in effort on the project.

The purpose of the virtual Tuesday meetings is to briefly discuss what the team will be talking about in the in-person meeting on Thursday. Along with this, we are also confirming our work with all of the other team members and making sure that we all agree with the quality of the work that we did over the weekend. The purpose of the in-person Thursday meetings on the Marietta campus is to heavily discuss our planned work for the weekend to complete the next deliverable. Once we all understand what each team member will be doing over the weekend days, we conclude the meeting. Our meeting hours are outlined in the Gannt chart for meeting about the system is included in the Gannt chart.

We have included four subtasks for the task of completing the final project. These primary tasks include, but are not limited to, the application backend, the application frontend, integrating the backend with the frontend, and final testing of the application. According to our Gantt chart, we will be dedicating time each week either actively building and completing a subtask, or planning for a further task, such as integrating the backend with the front end and testing. Following this strict schedule, we believe that we can finish the backend and the frontend of the application within two weeks, and then we can begin integrating the two and begin testing of the final product. During these two weeks, our meetings will primarily consist of confirming our work with others ensuring the completion of the modules are running smoothly and discussing our next steps within those modules.

With multiple meetings per week, we are accounting for the risks that we can expect in an earlier portion of this document. If difficulties arise, we can always postpone a meeting and hold one at a later date to be determined. Our two meetings a week and multiple workdays over the weekend give our group a lot of leniencies with the completion of each module subtask in a smooth fashion, especially in the case that we need to add previously unaccounted for buffer time.

The final product submission is due the week of April 14th, and the team plans to spend the final week and a half of the meetings and individual work with thorough testing. We will be testing our product well before this week and a half before the submission deadline, but we feel as if this extensive testing period is needed to ensure a quality product.

Rough Cost Estimate

ManageMe.io development includes planning budgets so that we can make best use of resources. Our cost estimation in the beginning splits expenses into two large categories: human resource costs and software/tools costs.

Human Resource Costs

The biggest portion of our budget consists of planning, development, and testing labor. Based on our estimate of how much work we need to accomplish, we have estimated each stage taking so many hours times an hourly rate typical of industry rates for each job within our budget.

Role	**Estimated Hours**	**Rate Per Hour ($)**	**Sub-total Cost ($)**
Planning	66	30	1.980
Development	85	60	5,100
Testing	29	30	870
Total Cost			7,950

Software & Tools Costs

To keep our costs low, we use open-source development software, which lowers our requirement for costly licenses. We do need cloud hosting, though, in order to deploy, so have a little extra cost added on.

Item	**Estimated Cost ($)**
Development Tools	0
Cloud Hosting	200
Software Licenses	0
Total Cost	200

Overall Project Cost

Combining both categories together yields our estimate for the project cost:

Item	**Estimated Cost ($)**
Human Resources	7.950
Software & Tools	200
Total Cost	8,150

While this is an estimate on the baseline, there can be fluctuations in the cost as the development is underway. Unexpected expenses, such as additional hosting fees, higher-level tool subscriptions, or longer development time, are possible. Our group has, however, used a cost-controlling approach to stay within our budget without compromising on our offering of a quality project management tool that meets all of the functional needs.

With careful planning of our resources and low-cost equipment, we will be trying to maximize efficiency at the lowest cost possible, with successful completion of ManageMe.io within budget.

Requirements (Functional and Non-Functional)

Functional Requirements

Here are the functional requirements of our application software. All requirements here will be implemented. As of now, each section will represent a page/view

Version 1

- Initial Release
    - User Login Page
        - The user must be able to login to their account using their email and password
        - The user must have the option of creating a new account if new to the web application
            - The user must provide their First and Last Name, Current Role, Email and Password
    - General Section
        - The user must be able to input a high-level description of the software project.
        - The user must be able to input the owner or project manager's name
        - The user must be able to input a project team member.
        - The user must be able to input a list of risks and risk status
    - Project Requirements Section

        - The user must be able to enter a list of both functional and non-functional requirements for the software project.
            - The user must be able to add multiple tasks to a project
                - The user, specifically the project manager, should be able to assign team members to a task
    - Effort Monitoring and Tracking Section
        - The user must be able to enter the effort in person-hours for:
            - Requirements Analysis
            - Designing
            - Coding
            - Testing
            - Project Management

          (by each requirement)
        - The user must have the ability to view total expended hours by:
            - Requirements Analysis
            - Designing
            - Coding
            - Testing
            - Project Management

          (by each requirement)

Non-Functional Requirements

Here are the non-functional requirements of our software. All requirements in this section describe how the system will perform.

- Performance
  - The system must support up to 500 concurrent users with response times under 2 seconds.
  - The system must handle up to 10,000 concurrent projects with efficient retrieval and storage of data.
- Scalability
  - The system shall be deployed using AWS for easy auto-scaling options.
  - The MongoDB database must be optimized for efficient querying.
- Security
  - All user data must be encrypted.
  - The system shall use roll-based access control to restrict sensitive information.
- Usability
  - The system shall use a clean and intuitive UI for easy project management and tracking.
- Reliability
  - The system shall have a 99% uptime with automated backups.
  - Downtime must stick to under 10 minutes a month and be scheduled for low-usage hours

- Maintainability
    - The system shall follow solid design principles and modular architecture for easy updates and maintenance.
    - Unit tests must have at least 80% code coverage to prevent regression.
- Logging & Monitoring
    - All user actions and system events shall be logged with timestamps and stored securely.
    - Real-time alerting must be implemented
- Portability
    - The system shall be deployable on both Windows and Linux environments.
    - The system shall be deployed on the cloud using AWS.

Major External Deliverables

The primary product of this project will be a **Project Management System** that enables tracking of projects, requirements, team members, risks, and monitoring effort. This system will be developed as a functional software application with the following external deliverables:

Project Planning Documents

- **Quick Plan (First deliverable)** – A high-level project plan outlining scope, goals, timeline, and initial risk assessment.
- **Comprehensive Plan (Second deliverable)** – A detailed project plan with refined scope, resource estimation, scheduling, and risk management.

Final Product (Third deliverable)

- **Project Management System** – A working software system that meets the defined requirements, allowing users to manage and track project data effectively.
- **Source Code Repository** – A Git-based repository containing well-documented and version-controlled source code, ensuring integrity and collaboration throughout development.
- **System Architecture Document** – Covers the system's design, including an overview of its architecture, database schema, key components, technologies used, and design decisions.
- **Presentation** – Summarizing the project's objectives, development process, key functionalities, challenges, and results. Demonstration of the Project Management System.

Summary

Our document outlined a product description that goes into further detail than the description we provided in our initial quick plan. Along with the further detailed description, we have also outlined our primary risks and assumptions with completing this project, and how we aim to minimize these risks and reduce the impact of them in the event that they become our group's reality. We have also provided a high-level schedule that goes into more detail than the one that we provided in our quick plan, along with a Gannt chart that highlights how our time will be allocated in the completion of each task/subtask. This schedule will be considered law amongst our group, and changes to the schedule must be agreed upon amongst all group members. Our rough cost estimate highlights what the group believes to be our cost for completing this project. The main cost besides time and work allocation amongst team members will be the cost of hosting our application on a cloud-based server.

This document also includes the system requirements that we will be completing for this project. The requirements go int great detail covering function requirements along with non-functional requirements to enhance system performance and usability. We also have included a final major external deliverables module to this document to further clarify our timetables and highlight each modules subtask.

This concludes the comprehensive plan for ManageMe.io. With proper project management and planning, our group aims to create a quality project using open-source tools and efficient task delegation. We aim to develop a robust, scalable, and low-cost project management tool responsive to user needs without going over budget. Thank you very much for your time reading over our plan.

References

Aftab, Zainab. “15 Top Software Development Risks & How to Reduce Them?” Hapy, 3 Jul. 2024. https://hapy.co/journal/software-development-risks/

Burenko, Stan. “How Much Does It Cost To Develop Software For A Project?” UpTech, 29 Jan. 2025. https://www.uptech.team/blog/software-development-costs

Kytainyk, Vlad. “Software Development Time Estimation: How Long Should It Take To Develop A Product?” Forbes, 2 Dec. 2002. https://www.forbes.com/councils/forbesbusinesscouncil/2022/12/02/softwaredevelopment-time-estimation-how-long-should-it-take-to-develop-a-product/

Martins, Julia. “What is a Gannt chart? A guide to mapping your project timelines.” Asana, 23 Jan. 2025. https://asana.com/resources/gantt-chart-basics