HCAR - Database/Webapp

Software Project Management Plan

Team Members: Carson Gustafson, Justin Crittenden, Michael Goodwyn, Orlando Trujillo-Ortiz

Last Modified: 2/7/2025

Project Website:

https://cs459-xplorers.atlassian.net/jira/software/projects/SCRUM/summary

GitHub Repository: https://github.com/Trujillo707/HCAR-Client-Database

Table of Contents

1. Introduction

1.1. Project Overview

HCAR (Humboldt Community Access and Resource Center) has proposed the creation of a new database for their clients. This project will require HIPAA certification, and will consist of a database for HCAR's client information, and a web application to securely access that database, while meeting their requirements.

This SPMP (Software Project Management Plan) contains project deliverables, how the project will be organized for each phase/sprint, the processes the project will contain, and how the work is split amongst the team.

1.2. Project Deliverables

Sprint (week)	Deliverable
1	Kickoff meeting (Intro to Project / Teams / Sponsor) Draft Initial/Skeleton Documentations (Management Plan, Requirement, Architecture, Detail Design, Test Plan, etc.) Setup Project Environment, Vision Control, etc.
2-3	Sprint Planning #1 Refine documentation draft(s) TDD on initial selected user stories (based on priorities) Sprint 1 implementation Regular Stand-ups Sprint Report, Review, and Retro #1 (End of Week 3)
4-5	Sprint Planning #2 Update/Refine previous documentations TDD expansions on selected/additional user stories (based on priorities) Sprint 2 implementation (and/or refactoring) Regular Stand-ups Sprint Report, Review, and Retro #2 (End of Week 5)

6-7	Sprint Planning #3 Update/Refine previous documentations TDD expansions on selected/additional user stories (based on priorities) Sprint 3 implementation (and/or refactoring) Regular Stand-ups Sprint Report, Review, and Retro #2 (End of Week 7)
8-10	Sprint Planning #4 Update/Refine previous documentations TDD expansions on selected/additional user stories (based on priorities) Sprint 4 implementation (and/or refactoring) Regular Stand-ups Sprint Report, Review, and Retro #2 (End of Week 10) SPRING BREAK on Week 9
11-12	Sprint Planning #5 Update/Refine previous documentations TDD expansions on selected/additional user stories (based on priorities) Sprint 5 implementation (and/or refactoring) Regular Stand-ups Sprint Report, Review, and Retro #2 (End of Week 12)
13-14	Sprint Planning #6 Update/Refine previous documentations TDD expansions on selected/additional user stories (based on priorities) Sprint 6 implementation (and/or refactoring) Regular Stand-ups Sprint Report, Review, and Retro #2 (End of Week 14)
15-16	Sprint Planning #7 Update/Refine previous documentations TDD expansions on selected/additional user stories (based on priorities) Sprint 7 implementation (and/or refactoring) Regular Stand-ups Sprint Report, Review, and Retro #2 (End of Week 16)
17	Final Demo and Presentations (5/15/2025, 5:00PM - 6:50PM) Final submission (the Canvas submission and submission to the sponsor) of the online repository including all final versions of documentations, source code, etc.

1.3. Evolution of Document

The table below will be used to track revisions of this document as our team moves forward with project development.

Revision Number	Date of Revision	Revised By	Summary of update
1.0	2/3/2025	All Group Members	Initial Draft outlining the project description, goals for each sprint, and project structure

1.4. Definitions, Acronyms, and Abbreviations

UML - Unified Modeling Language

SPMP - Software Project Management Plan Template

WRS - World Requirement Specification

2. Project Organization

2.1. Process / Lifecycle Model

The development of this database / web application will be through this group, with weekly consultation from the HCAR representatives. The team will be working using Scrum, an agile sprints methodology, making incremental progress to promote productivity and success for the project.

The development of the database and database manager / webapp will be through this group. During the development phases, the team will get together every week, and allocate the work needed to be done to complete the phase using an agile working methodology. By doing so will ensure a larger safety net when considering possible errors or flaws that come with the finished product, and provide proper validation for each user story in the project.

2.2. Organizational Structure

This team is organized as a single group of four, with the role of Team Lead revolving to each member every two sprints as outlined in section 1.2. of this document. The team lead will be responsible for primary communication with the client, and will lead standup meetings with the rest of the team. Every member of the team is expected to present progress reports, and any issues that may come up during project development.

The beginning of the first stage will start with three developers and a team lead. The group will decide every week the workload that is needed to complete the sprint, adding them as issues to the Jira project board. The members will be responsible for splitting up and working on the required deliverables before the next sprint.

2.3. Organizational Boundaries and Interfaces

The project progress will be tracked through an associated Jira project board in Scrum format (To-Do → In-Progress → Done). The link to the Jira board and the associated GitHub repository is on the title page of this document, any technical documentation for this project can also be found on the Jira board.

2.4. Work Elements, Schedule, and Budget

Phase (weeks)	Sub-Phase	Developers	Start date	End date	Software resources
1					

3. Managerial Process

3.1. Management Objectives and Priorities

The focus and priorities of the members of this project are to meet the proper requirements to provide a product the client can satisfice. This group will be meeting for a minimum of 1 hour a week in-person to discuss the goals of our project and the system functionality needed to achieve them, as well as collaborating on the implementation of the code for this project. Standup meetings will be held between the team members every other day to discuss progress on various tasks, as well as potential blockers preventing progress.

3.2. Assumptions, Dependencies and Constraints

Our assumptions, dependencies, and constraints going into this project are the following:

Assumptions:

- The primary stakeholders / users of the web application will be staff, with managing staff as administrators.

Dependencies:

- The application will depend on the data entered by the users in order to tailor certain functionalities
- The application functionality will require the proper devices (can be referenced in the User Manual)

Constraints:

- The due date for this project
- The budget for subscription-based tools
- Conflicting responsibilities that inhibit the amount of time our project group is able to spend focusing on the project.
- Device hardware being up-to-date and functional

4.   Technical Process

4.1.   Methods, tools and techniques

Version control will be achieved by use of an online GitHub repository (for code) and a Jira project board (for workflow). GitHub and Jira will allow for streamlined collaboration between members.

Discord and Zoom will be the main method of communication between team members for the entirety of the project. Voice, video, and screen-sharing calls will be utilized to adequately collaborate remotely. An agenda will be set for every meeting to ensure proper goals are met during the session.

Google Docs will be the main word document collaboration tool for the project. It will allow functional synchronous and asynchronous text document collaboration that prevents data loss via cloud storage.

UML modeling will be created using the online LucidChart solution. LucidChart allows for online collaboration on documents.

The application will be written in the

4.2.   Risk Analysis

	Risk	Category/Type	Probability	Effect/Impact	RMMM/Strategy
1	Constantly changing minor requirements from organization	Requirements	High	Tolerable	**Avoidance**: Meeting every week with stakeholders with a prototype of current progress, and agenda for future progress **Contingency**: Negotiate with stakeholders, and make proper refactors based on the feasibility of the changed requirement.

2	Product will degrade / not be maintained / budget will not be held for cloud services	Stakeholder	Low	Catastrophic	**Avoidance:** Provide adequate documentation to keep the product maintainable **Contingency:** Pay for the cloud hosting services and hire a developer to maintain
3	Lack of initial budget/resources for project startup	Planning and Control	Low	Tolerable	**Avoidance**: Can avoid by preemptively optimizing application performance beforehand **Minimization:** Would have to document clearly costs and meet with stakeholder **Contingency:** Increase budget or reduce the scope of the project
4	Inadequate stakeholder commitment / conflict between stakeholder's requested requirements	Stakeholder	High	Tolerable	**Minimization:** Continue to share user stories and be transparent with project ideas. **Contingency:** ask for solid requirements, and achieve it at a minimum.
5	Team experience with tools necessary for this project is low	Team	Moderate	Insignificant	**Minimization**: The team can share experience / explain tools for project progression **Contingency**: Make time to read proper documentation for the tool being used.

6	Disagreement within the stakeholders about database features	Stakeholders	Moderate	Serious	**Minimization**: Maintain communication with both stakeholders, and negotiate any problems that arise. **Contingency**: If problems do arise, make the stakeholders decide before the next time the team meets.
7	Making product too simple (losing some functionality) or too complex for the user	Planning and Control	Moderate	Tolerable	**Avoidance**: Keep it simple before delivery **Minimization:** Repeatedly check with stakeholders to see if requirements are correct. Acceptance criteria and tests.
Cutoff Line					
8	Poor performance of cloud database requests	Technology	Very Low	Serious	-
9	Scheduling between team/stakeholders	Team/Stakeholder	Very Low	Serious	-

4.3.   Software Documentation

The following is a list of documentation associated with this project:

- User manual / WRS
- Models using UML
- Entity-Relationship Diagram
- Source code documentation for contributed files in GitHub
- Project board tasks through Jira

4.4.   Monitoring, Reporting, and Controlling Mechanisms

describe the management reports that should be produced, when these should be produced, and the project monitoring and control mechanisms used (include the rationale)

5. List of Figures/Tables

Insert tables here

6. Professional Standards

- describe the expected behavior of the team members related to scholastic dishonesty, meeting schedule and quality expectations for tasks and deliverables, etc
- include the rationale
- refer to Appendix A for more details

Configuration Management

ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS
- students should work with their project sponsor(s) to identify all the standards and constraints that should be applied for preparing this document

ADDITIONAL REFERENCES
- include other related references that are not included the section above

Appendix A.

The following provides a professional standards guideline for the teams. This guideline may be tailored. The professional standards must be agreed upon by each member in the team.

Guideline:

On the first occurrence of unacceptable behavior, determine the circumstances involved, resolve the problem, and document the event in the meeting minutes.

On a second occurrence, notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem.

On a third occurrence, again notify the instructor of the problem. A meeting will be set up to evaluate the situation and resolve the problem. At this point, the team will have the *option* of removing the team member. If removed, then the team member receives a pro-rated grade based on the number of weeks they have participated in the group.

Examples of unacceptable behavior may include not delivering on time, delivering poor quality work,

missing team meetings, being unprepared for team meetings, disrespectful or rude behavior, etc. Reasons such as "too busy" or "I forgot", or "my dog ate my design model" are unacceptable.

Valid reasons that must be considered include those listed for obtaining an incomplete standing in a course (illness, death in the family, travel for business or academic reasons, etc.)