**San Francisco State University**

**SW Engineering CSC 648/848**

**Milestone 3**

**Section 04 Team 05**

# Project title

Priority Hub (Code: PH)

# Team members

* Luis Aguilar (Team Lead)
* Tyler Tam (Scrum Master)
* Trevor Eichler (GitHub Master)
* Andy Li (Front Lead)
* Lam Tran (Back Lead)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 1.0 | Monday, October 30th, 2023 | Team 05 | Version to submit |
|  |  |  |  |

**October 30th, 2023**

# Table of contents

[Project title 1](#_Toc149603819)

[Team members 1](#_Toc149603820)

[Revision History 1](#_Toc149603821)

[Table of contents 2](#_Toc149603822)

[Content and structure for Milestone 3 document for review by institutors 3](#_Toc149603823)

[1. UI and functionality feedback (P1 functions only) 3](#_Toc149603824)

[1.1. Instructor’s comments on UI/functionality for your demo (should be during the class of M3 demo) 3](#_Toc149603825)

[1.2. Your Plan for the comments 3](#_Toc149603826)

[2. List of P1 features committed for delivery 3](#_Toc149603827)

[3. Project status 4](#_Toc149603828)

# Content and structure for Milestone 3 document for review by institutors

## UI and functionality feedback (P1 functions only)

During the meeting, students will demonstrate to run your SW from deployment server:

* Test 5~6 P1 features
* Show UI and usability: adherence to the feedback on UI mockup at M2, layout, flow, clarity, functionality etc.

Instructor will

* Check functionality and record issues/observe bugs.
* Share comments on key UI and functional implementation.
* Verify enough web pages are implemented and connected.
* Verify Performance of web page

**Students must** record meeting summary (use a scribe and Appendix I as template. Keep tracks of institutor’s comments). Then the team should meet to analyze feedback, prioritize, and revise and plan to implement changes accordingly. **Also, immediately after the review the team must finalize P1 set of features and focus only on those from then on.**

### Instructor’s comments on UI/functionality for your demo (should be during the class of M3 demo)

* Change the color of the buttons showed in the M3 demo. Colors should match the overall look & feel of the application (black theme).

### Your Plan for the comments

* The team updated the color to match the application's look & feel. Instead of using bright colors like green, red, and yellow in the buttons, the team changed them to a black color and lighter tones.

## List of P1 features committed for delivery

**Write down the items before the demo** and verbally explain it during the meeting if time is allowed.

**Once you commit at M3, you cannot change during the rest of the semester. You should implement by M5.**

**Authentication**

* User should be able to Sign In
* User should be able to Sign Out
* User should be able to Sign Up
* User should be able to Reset Password

**Task management**

* User should be able to Create Task
* User should be able to Edit Task
* User should be able to Delete Task
* User should be able to Complete Task
* User should be able to Search Tasks
* User should be able to List all Tasks
* User should be able to History of all Tasks
* User should be able to track the time of their Tasks

**“End of the day” summary**

* User should be able to view an “End of the day” summary for all their tasks.
* User should be able to filter the task’s charts using two date fields (data between two dates set by the user)

## Project status

**Write down the items before the demo** and verbally explain it during the meeting if time is allowed.

Risks: all actual (not hypothetical) risks (schedule, teamwork, technical, skills etc.) should be identified and either resolved or plans should be made to resolve them.

* Skills
  + A team member may become frustrated, anxious, and put off the next work if they can't fix a serious error in the code for days.
    - Mitigation plan
      * Ensure team members comprehend the existing code well before moving on to the next stage or building upon the current code.
      * Engage in having a 1-on-1 meeting with the team lead or whoever knows more about the specific issue.
      * If you don't grasp a certain section of the code, ask in team Discord, communicate, this can also help others learn.
* Teamwork
  + Other tasks are delayed if two team members end up working on the same project functionality because of poor communication.
    - Mitigation plan
      * Ensure that everyone is aware of who is working on what task at each stage, use the JIRA board to check your assigned feature, map all development effort, add comments or any required detail.
      * Focus on having good communication with all other group members. Let the team know you’re pushing code by sending a message through Discord, etc.
  + A team member who is responsible for building functionality on top of another team member's code will have a difficult time if the former team member doesn't carefully manage the code structures and files.
    - Mitigation plan
      * Engage in having a weekly code review. Maybe during team meetings.
      * When a file starts to become complex, start debugging and modularizing the code before it becomes too late.
      * Learn and adhere to the team's programming language's style documentations. Keeping things consistent and neat.