**Vaccine Clinic Information and Scheduling System (VCISS)**

**Initial Project Plan**

Ian Parish (ip), Jiacheng Wei (jw),

Thomas Renn (tr), Xing Qian (xq) - 4-12-2021 - v1.01

**Table of Contents**

1. PP Revision History 1

2. Management Plan 2

2.1. Team Organization 2

2.2. Work Distribution 2

2.3. Decisions and Communication 3

3. Schedule 3

3.1. Schedule Milestones 3

3.2. Monitoring and Reporting 4

4. Build Plan 4

4.1. Schedule 4

5. Rationale 5

5.1. Build Plan 5

5.2. Risk 5

5.3. Build Plan 5

# 1. PP Revision History

**Date Author Description**

**————— ——— ————————————————————————————**

4-12-2021 ip, jw, tr, xq Created the initial document.

**2. Management Plan**

* 1. **Team Organization**
* **Project Manager: Ian Parish**
* **Assistant Project Manager: Jiacheng Wei**
  + The PM will be responsible for setting deadlines and ensuring requirements are met in a timely manner, as well as maintaining relevant documentation and coordinating work.
* **Lead Engineer: Xing Qian**
  + The lead engineer will be responsible for coordinating and delegating the development of all required components of the VCISS. A co-lead should not be necessary, as all members of the team will be involved in development of the system.
* **Lead Test Analyst: Thomas Renn**
* **Assistant Test Analyst: Ian Parish**
  + The lead test analyst will be responsible for testing module functions and developing test cases.
* **Integration Specialist: Jiacheng Wei**
  + The integration specialist will be responsible for integrating each module into the greater VCISS and coordinating with the Lead Test Analyst and Lead Engineer to ensure proper functionality.
  1. **Work Distribution**
* **Ian Parish**:
  + Module: Appointment Scheduling Module
  + Documentation: Initial Project Plan/SRS/SDS
* **Jiacheng Wei:** 
  + Module: Account/Login Module
  + Documentation: Initial Project Plan/SRS/SDS
* **Thomas Renn:** 
  + Module: Search/Results Module
  + Documentation: Initial Project Plan/SRS/SDS
* **Xing Qian:**
  + Module: Clinic Finder and Information Module
  + Documentation: Initial Project Plan/SRS/SDS

**2.3 Decisions and Communication**

Our team makes decisions through communication in both writing and verbally. We will meet over Zoom if there are any big problems that come up that cannot be resolved over Discord. However, the primary extent of our communication and collaboration will happen over Discord and Google docs.

**3. Schedule**

* 1. **Schedule** **Milestones**

1. Create version control system using Git:
   1. Entire team
   2. Complete by April 16.
2. Define the pairings between the UI and the back-end database:
   1. Entire team.
   2. Complete by April 16.
3. Complete the Appointment Scheduling Module:
   1. Ian Parish
   2. Complete by April 19.
4. Complete Account/Login Module:
   1. Jiacheng Wei
   2. Complete by April 19.
5. Complete Search/Results Module:
   1. Thomas Renn
   2. Complete by April 19.
6. Complete Clinic Finder and Information Module:
   1. Xing Qian
   2. Complete by April 19.
7. Complete UI:
   1. Entire team
   2. Complete by April 19.
8. Complete backend:
   1. Entire team
   2. Complete by April 19.
9. Begin integration for UI components with backend:
   1. Entire team
   2. Begin April 19.
10. Complete Integrate UI components with backend logic:
    1. Jiacheng Wei
    2. Complete by April 23.
11. Complete testing:
    1. Thomas Renn
    2. Complete by April 26.
    3. **Monitoring and Reporting**

Project progress will be monitored via version tracking systems such as GitHub or Bitbucket. These platforms will display the editor’s information of each submission such as submission time and content. Project progress will be monitored via Discord chat where individual team members will share progression and any necessary updates.

**4. Build Plan**

* 1. **Schedule**

1. Create a version control system using Git and model the UI aspect being paired with a backend. Clearly define these pairings and what their declarations will look like. Also design MySQL table schema.
   1. Completed by April 16.
2. Begin implementing the UI as defined in the previous build, and the database API implementation. Begin to connect the UI and the MySQL database tables if time permits.
   1. Begin implementation as soon as definitions are defined. Complete by April 19.
3. Build a naked Django backend with only several routers for testing, built a naked frontend web page with only home web page for testing, create tables based on schema.
   1. Complete by April 19.
4. Implement web page based on the UI and consistently adjust the backend restful api, also modify in document for reference and cooperation, at the same time implement backend logic, consistently adjust table schema for better query efficiency and comply with normal forms
   1. Complete by April 19.
5. Connect the frontend to the backend, make sure the api works as expected, the data is normally stored and fetched, the web page interact perfectly, do this step as early as possible.
   1. Complete build step April 23.
6. Test the system with manual operation and automatic test framework if possible.
   1. Full integration should be complete, and testing begins April 23 and completed by April 26.

**5. Rationale**

**5.1 Build Plan Reasoning:**

We use Git for code management and code synchronization. Everyone develops in their own branch, creating pull requests when synchronization is need. Convention should be built as early as possible so that cooperation efficiency can be increased a lot.

Other things like data modeling and UI design should also be done early. Once UI design finished, team members can discuss and promote what each module exactly could and should do. And then what kind of service our backend restful API should provide. In this way, backend logic can be construed by the need of the frontend and divided into several modules.

Designing data schema is also important, we should pay more attention to what one kind of data should contain. Like whether the user table should include appointment information or put it into another individual appointment table. After the schema is designed, how the backend does CURD is in the obvious way.

Frontend and backend are relatively Interrelated and independent. They can be developed independently since we have many ways to mock data or http requests to debug. But we should connect them as early as possible since it can release many problems. Each module involves frontend as well as backend, connecting both and mock user behavior can test the functionality and stability of the system.

**5.2 Risks**

Since all team members are newbies to the system development cycle, the main risk is the lack of knowledge to build the system. This risk leads to inconsistent team steps, so the team needs to plan and allocate work.

**5.3 Risk Reduction**

The strategy adopted by the team is to build all the databases and UI/modules in one week so that the team gets another week to integrate all the parts, test the functionalities of each section, and improve the code and documentation. Therefore, the team can spend a reasonable amount of time on the team project to optimize the system.