Dogtor Web Application: Veterinary Scheduling and GateKeeping Application

Project Plan

ENGG4000 FR03X

Tash Zaman, Andre Aucoin and Jordan Mitchell

Table of Contents

1 Project Plan Overview

1.1 Purpose of the Plan

- To develop a gatekeeping and scheduling web application for veterinary staff and their clients.
- Developing a functional plan so that all of our milestones are met by the deadline.
- To establish an efficient development routine.
- To ensure that we are developing a quality product that meets the expectations of the veterinary staff while providing high quality user experience for clients.
- Build a robust system that can be optimized through system testing.

1.2 Management

- Organization
    - Jordan: Team Lead, Backend Developer, DevOps
    - Andre: Backend Developer, Database Designer/Implementer
    - Tash: Frontend Developer, QA Lead

- Backend Developer
    - Develops the business logic which will interact with the frontend.
    - Develops the established connection between the databases and frontend.
    - Develops a method to store information obtained from the user to the database.
    - Develops a method to manipulate information held within the database.

- Database Designer/Implementer
    - Manages elements within the database table (The who, what, where and how of data).
    - Manages the relationships between database tables.
    - Establishes the quantity of required tables.

- DevOps
    - Implements CI/CD to aid with the development process.
    - Creates a build pipeline that will be maintained and updated.
    - Develops integration tests to ensure modules are compatible.

- Team Lead
    - Manages team progress and deadlines.
    - Main source of communication between the client and development team.

    - Establishes the scope for each sprint and ensures that all team members have the same vision for the product.

- Frontend Developer
    - Develops a good quality user interface and experience (UI/UX).
    - Communicates with the backend developer about the information and structure being sent to the backend.
    - Ensures consistency between all of the web pages to improve user experience and usability.

- QA Lead
    - Ensures that modules are implemented with unit-testing.
    - Communicates with DevOps to ensure proper integration testing.
    - Leads team in development of verification and validation of the system (full system testing).

2 Risk Monitoring, Mitigation, and Management

2.1 Underestimating Time	Probability : Medium	Impact : Medium
Mitigation	Precise and continuous communication with team members and clients	
Monitoring	Ensuring that milestones are being met and updated on time	
Management	Adjust tasks undertaken per sprint based on team velocity and KPIs.	

2.2 Difficulty with Implementations	Probability : Medium	Impact : High
Mitigation	Pair programming when blocked during development	
Monitoring	Discuss encountered issues with team members as early as possible so development can resume in a timely manner	
Management	Seek expertise from experienced individuals or other resources.	

2.3 Staff Turnover	Probability : Low	Impact : High
Mitigation	Reduce the scope of the project and salvage work leftover	
Monitoring	Clear and honest communication with team members	
Management	Discuss next steps with advisors	

2.4 Staff Unavailable	Probability : Medium	Impact : Medium
Mitigation	Ensure that all team members are equally knowledgeable about the system and proper data transfer takes place when members are available to continue their work	
Monitoring	When a team member is unavailable to work the project, notify the rest of the team as early as possible	
Management	Redistribute the workload when possible to ensure the product quality is maintained	

2.5 Major Requirements Change during Development	Probability : High	Impact : Low
Mitigation	Demo working features frequently with the client and obtain client feedback which can be implemented and integrating into the product	
Monitoring	Establish business requirements with clients and continue work based on feedback given by the clients	
Management	Update established requirements based on client feedback	

2.6 Difficulty Understanding Concepts	Probability : High	Impact : Low
Mitigation	Research information from experienced individuals and the internet	
Monitoring	Discuss knowledge on concepts with team members to ensure team has proper understanding	
Management	Reiterate what was learned during the research phase to make sure topic is properly understood	

3 Quality Assurance Plan

- ***3.1 Formal Technical Reviews***
    - **3.1.1 Purpose:**
        - A way for the team to establish a written agreement between clients and developers to ensure the quality of the software product.
    - **3.1.2 Format:**
        - The development team presents information to be reviewed and a discussion will ensue with the client. Topics discussed would be documented so that they can be approached by the development team at a later date. All issues brought up will be later addressed with the client for approval.
    - **3.1.3 Review requirements:**
        - Project Plan Review
            - Roles and plans must be established for the development team.
            - A tentative document schedule must be developed.
            - A cost estimation must be completed and appropriate resources must be allocated according to that estimation.
            - A risk management plan must be completed and all project risks must be considered according to that plan.
            - A quality assurance plan has been decided upon and approved by the development team.
        - Software Requirement Review
            - Upon discussion, all basic requirements by the client are reflected in the design.
            - The development will consider all possible use cases.
            - The development team will consider internal and external interfaces during the design phase of the system.
            - All requirements reflect the project schedule and available resources.
        - Design Review
            - Requirements established by the development team must be considered.
            - Factors such as modularization and object-orientation will be considered.
            - Well structured data structures and interfaces will be considered.
        - Test Plan Review
            - The test plan will encompass unit tests, integration tests and system tests which will provide full coverage of the functionality.

- ***3.2 Test Plan:***
    - **3.2.1 Unit Testing:**
        - Each individual module shall be tested to ensure proper functionality.
    - **3.2.2 Integration Testing:**
        - When modules are pieced together to complete a feature, testing will be done to ensure there is no loss of functionality.
    - **3.2.3 Validation and Verification Testing:**
        - As joined modules are integrated to the product, testing will be done to ensure the system function works as expected and client expectations are met.

4 Project Schedule

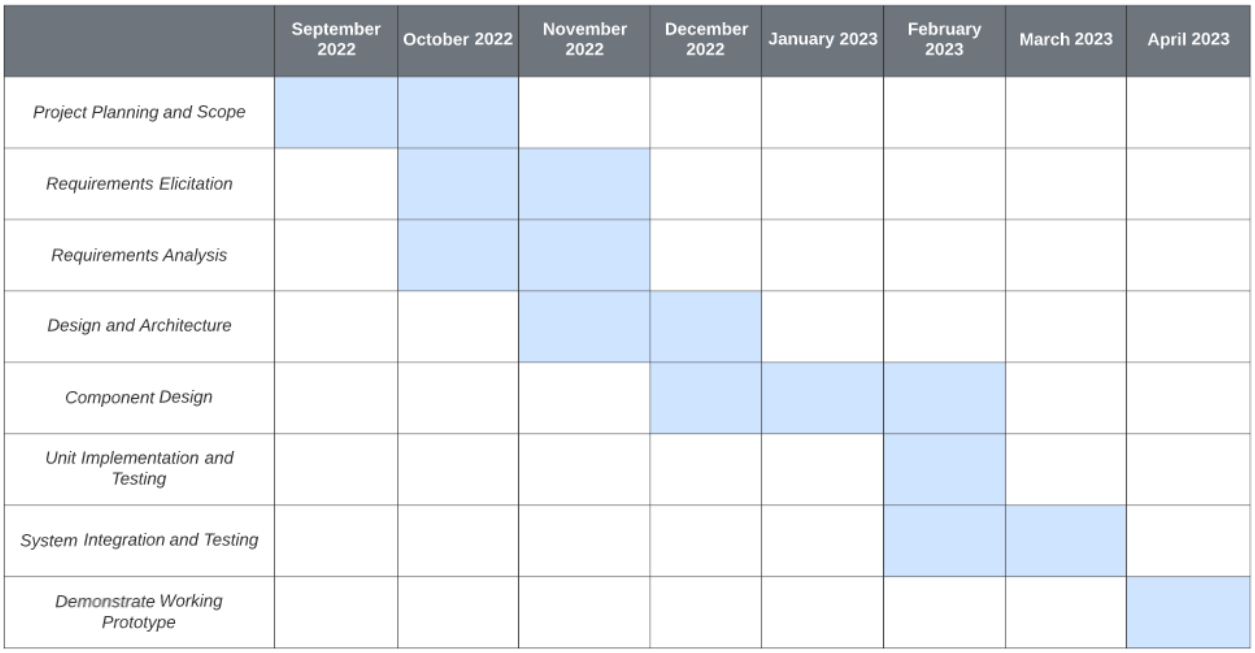

Figure 1: Gantt Chart of Project Schedule

The Gantt chart above, Figure 1, displays a schedule of where our focus will be for documentation during the completion of the project. The Dogtor team will be following a waterfall process to deliver documentation for the milestones from September 2022 until March 2023. The team will be using an agile scrum process for development: there will be 2 week-long sprints and scrum activities such as sprint retrospectives, sprint demos and sprint planning. The first sprint planning event will be October 23rd, 2022 at 2pm on Teams. Due to the team using an agile process for development, the milestones will be resubmitted at the end of the term.

5 Project Cost Estimation

Function points were used to estimate the entire cost of the project. The analysis is shown below and in section 6, Function Point Analysis.

5.1 Number of User Inputs

Function	Count	Weight
Login Screen (Information, Password)	2	3
Search (Doctor, Schedule, Surgery Type, etc…)	1	3
Fill Questionnaire	1	6
Create Profile (Customer, Technician/Doctor, Admin)	1	6
Add Repository	1	6

5.2 Number of Outputs

Function	Count	Weight
Login	1	4

Create Profile (Customer, Technician/Doctor, Admin)	1	4
Search (Doctor, Schedule, Surgery Type, etc…)	1	7
Add Repository	1	4
Fill Questionnaire	1	4

5.3 User Inquiries

Function	Count	Weight
Search (Doctor, Schedule, Surgery Type, etc…)	1	15

5.4 Number Of Files

Function	Count	Weight
Login	1	7
Profile	1	10

Output	1	15
Questionnaire	1	10
Repositories	1	7

5.5 External Interfaces

Function	Count	Weight
0	0	0

Total Weight = 111

6 Function Point Analysis

To calculate the function points, our team has answered 14 questions about our system. Each question is answered using an integer value between 0 to 5. A value of 0 means that it has no effect or is not applicable and a value of 5 means it has a major impact.

1. Does the system require reliable backup and recovery? **4**
2. Are data communications required? **5**
3. Are there distributed processing functions? **2**
4. Is performance critical? **5**
5. Will the system run in an existing, heavily utilized operational environment? **2**
6. Does the system require on-line data entry? **4**
7. Does the on-line data entry require the input transaction to be multiple screens or operations? **0**
8. Are the master files updated on-line? **0**
9. Are the inputs, outputs, files, or inquiries complex? **5**
10. Is the internal processing complex? **4**
11. Is the code designed to be reusable? **5**
12. Are conversion and installation included in the design? **3**
13. Is the system designed for multiple installations in different organizations? **5**
14. Is the application designed to facilitate change and ease of use? **5**

Function Point Total: 49
FP= Total Weight * [0.65 + 0.01 * Function Point Total]
FP= 111 * [0.65 + 0.01 * 49]
FP= 126.54

Efficiency in the span of 8 months is:
Efficiency = 8 person months / 126.54 FP = 0.063 person months/FP.