Software Project Management Plan

For

Belleza Scheduling WebApp

Contents

2.1 **Introduction**

2.1.1 Project Overview

2.1.2 Project Deliverables

2.2 **Project Organisation**

2.2.1 Software Process Model

2.2.2 Roles and Responsibilities

2.2.3 Tools and Techniques

2.3 **Project Management Plan**

2.3.1 Work Breakdown Structure

2.3.2 Assumptions,Dependencies and Constraints

2.3.3 Schedule

2.4 **Risk Management Plan**

2.1. Introduction

2.1.1 Project Overview:

BSW is a multi-platform web application that focuses on aiding the saloon by providing them with a scheduling system. This system will serve as a useful interface between the customers and the saloon. BSW is one such application, which is simple to use and has an interactive user interface.

Being a stand-alone application, this software will be centrally governed. All the clients will have access to the system but shall be managed by one admin only.

Clients can cancel the appointment if he/she wishes to. Overall this project is about helping the clients to easily get their appointments without them having to wait in the saloon. Also, it helps the saloon to get a better idea of number of clients they might have the following day.

2.1.2 Project Deliverables:

The project aims to deliver a working application, which works on multi-platform (stand-alone and web). It has a verified login and guarantees security.

With stand-alone application, only admin gets to edit/delete the current status information of the saloon. However, a user can cancel their appointments as well.

2.2. Project Organization

2.2.1. Software Process Model:

The BSW project will follow an incremental and an iterative development model for its deliverables.

The development will be done in several phases and each phase will represent a complete development cycle, with certain functionality of the system delivered at the end of each phase. The phased approach to delivery provides flexibility in what the team will deliver, gives an opportunity to reassess the effort for each phase and allows both the team and the client to change any of the phase's content.

2.2.2. Roles and Responsibilities:

Projects of different sizes have different needs for how the people are organized. In a small project, little organization structure is needed. However, for large projects, there are more and more people involved, and it is important that people understand what they are expected to do, and what role people are expected to fill.

This section identifies some of the project roles that may be required for the project.

Software Analyst:

- By definition, a software analyst is a person who studies the software application domain and prepares the software requirements and specification (SRS) document. The analyst is expected to do the analysis, validate it with the client and felicitate the requirements, and convert them into an architecture and design that will become the blueprint for the solution.
- He should possess communication and relationship skills, as well as skills pertaining to word processing apps, spreadsheets.

BACK END DEVELOPER:

- A **back-end** web **developer** is responsible for server-side web application logic and integration of the work front-**end** web developers do. **Back-end** developers usually write web services and APIs used by front-**end** developers and mobile application developers.

FRONT END DEVELOPER:

- **Front end developers use HTML, CSS, and JavaScript to code the website and web app designs created by web designers. The code they write runs inside the user's browser (as opposed to a backend developer, whose code runs on the web server).**

TESTING TEAM:

- **The testing team tests the software on the basis of the test cases. The role of the testing team is to develop test cases and prioritize testing activities, execute all the test case and report defects, define severity and priority for each defect and carry out regression testing every time when changes are made to the code to fix defects.**

2.2.3. Tools and Techniques:

Tools:

- **Programming Languages: HTML, CSS, JAVASCRIPT, PYTHON, SQL.**
- **Database:** SQLite and SQL.
- **Design:** Object-oriented analysis and design (OOAD)
- **Additional: Flask** framework hosted on **HEROKU** Server

Techniques:

- **Software Documentation:** Documentation such as Software Project Management Plan (SPMP), Software Requirement Specification (SRS), detailed Software Design Document (SDD) and Software Test Document (STD).

2.3. Project Management Plan

2.3.1. Work Breakdown Structure:

Sahil: Design, Front-End

Dushyant: Back-end,testing

Madhav: Back-end,project analyst

Ujjval: Testing,Documentation

2.3.2. Assumptions, Dependencies and Constraints:

Assumptions:

- Stable internet connection.
- The project team will deliver the work as scheduled.
- The project team will have access to needed systems.
- The required resources will be available throughout the project.

Dependencies:

- The information stored in the database can be edited or deleted as and when required.
- Real-Time applicability needs the database to be synchronized.
- Documentation and Testing can happen only along or after the coding.

Constraints:

- **Schedule constraint**: To deliver the milestones within the deadline.

- **Skills:** The team members may not have the skills and capacity to deliver the milestones.
- **Resource constraint:** All the team members are not available all the time. Conflicts are bound to happen with the individual schedule and project's schedule.
- **Database design:** The database should be designed to cater to the needs of the software.

TIMELINE

Phases	Planned Start Date	Planned End Date
Project Planning	12/08/2017	9/09/2017
Design	9/09/2017	25/09/2016
Coding	27/09/2017	20/10/2017
Unit Testing and Review	20/10/2017	10/11/2017
Documentation	15/11/2017	20/11/2017

2.4. Risk Management Plan

Risk Identification:

Risk Type	Possible Risks
Team	• Conflicts between team members • Members with negative attitude towards project • Lack of cooperation from any member or lack of adequate involvement • Inexperienced team members • Lack of knowledge required for the software development
Users	• Failure to manage end user expectations • Admin not satisfied with the robustness of the software
Requirements	• Changes to requirements that require major design rework proposed • Customer may fail to understand the impact of requirement change • Misunderstandings of the requirements • Continuous change of requirements
Technology	• Insufficient procedures to ensure security, integrity and availability of the database • Inadequate security features being built into the system • System failure and data loss
Tools	• Software tools cannot work together in an integrated way

	• **Possible failure reports not generated on time or correctly.**
Estimation	• **The time required to develop the software is estimated** • **Artificial Deadlines** • **Lag on progress due to unavoidable reasons like vacation/exam/member being absent**

Risk Planning:

Risk	Strategy
Staff Illness	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
Requirements changes	Derive traceability information to access requirements change impact;
System failure	The development team should cease work on that system until the environment is made stable again, or should move to a system that is stable and continue working there.
Data Loss	Create regular back - ups of the files and data used in the project.
Customer satisfaction	Give Admin sufficient control over the software.
Technology Does Not Meet Specifications	The customer should be immediately notified; a meeting should be held between the development team and the customer to discuss at length this issue.
Lack of Development Experience	Experienced team members to help the inexperienced team members.
Ineffective communication	Hold regular and interactive project meetings.

Risk Monitoring

Risk Type	Potential Indicators
User	Multiple error messages; user/customer complaints
Requirements	Many requirements change; confusion w.r.t. requirements in the team
Technology	Many reported technology problems; no back-up of files and data created; discrepancies in database; web browser incompatibility
Tools	Reluctance by team members to use tools; no error messages generated for exceptions
Estimation	Failure to meet agreed schedule; failure to clear reported defects; no progress with the development of software