# SOFTWARE PROJECT MANAGEMENT PLAN

## 1. Introduction

### 1.1 Purpose

This document establishes roles, responsibilities, processes and schedules for managing the software development process, in addition to outlining the tools, methods, and procedures to be used.

### 1.2 Scope

This document contains a plan for the successful execution of the goals outlined in the Project Charter, which goals are to be further defined in the Software Requirements Specification (SRS) document.

### 1.3 Definitions, Acronyms, and Abbreviations

The following are explanation of uncommon or domain-specific terms.

- **Application Programming Interface (API):**Implemented declarations of how a software component interacts with other software components. A common example of an API is a web service that provides data via a collection of resource addresses. This project will use APIs for preexisting tools to increase efficiency and avoid "reinventing the wheel".
- **Architectural Design**: High-level design of structures within a software system. Architectural design includes the set of structures composing the system, how they interact, and their properties.
- **Baseline**: A minimal or starting point estimate, used in comparison.
- **Cascading Style Sheets (CSS):** A styling language used to change the look and feel of a document written in a markup language.
- **Change request**: A formal request to alter a pre-established aspect of the project (especially project requirements).
- **Component Design**: A high-level design technique involving individual components of a system. A component is a set of related functions or actions that encapsulate one element of the system.

- **Database Design**: The design of the data store to be used in the system. Involves mapping the various entities, their attributes, and how they are associated with other entities.

- **Decision Traceability Matrix (DTM**): A document describing key decisions made during the life of a project, along with references to relevant documents, allowing decisions to be traced to their source.

- **Deliverable**: Any documentation or software produced that will be given to the client for review and use.

- **Graphic User Interface (GUI):** A category of interfaces that allows a user to visually interact with software (usually via mouse or touch screen), as opposed to traditional text-based interfaces.

- **Hypertext Markup Language (HTML):** The standard markup language for documents on the web. HTML forms the building blocks for the content of a webpage.

- **Interface Design**: The process of designing the methods for two (or more)modules or components to interact with each other.

- **Iteration**: A cycle of phases. Iteration is used in this project to refer to the process of repeating design, implementation and testing phases in order to continually refine the project.

- **JavaScript:** A programming language typically tied into a client's web browser. JavaScript is typically used to make rendered pages more interactive.

- **PHP**: Hypertext Preprocessor (PHP): A scripting language commonly used to dynamically produce HTML to be served to a web browser. In this project, it will also be used to retrieve, interpret, validate and process user submitted data.

- **Quality Assurance**: A set of methods for monitoring the software development process to ensure quality deliverables.

- **Research and Development (R&D)**: A combination of research, which is the "systematic study directed toward fuller knowledge or understanding ofthe fundamental aspects of phenomena and of observable facts", anddevelopment, which is the application of research to find solutions forrelevant problems.

- **Serial Life Cycle**: A process for developing software where each phase in the life cycle is given a set amount of time and follows linearly into the next phase. Requirements gathering and elicitation will be completed using the Serial Life Cycle.

- **Slack**: A project collaboration tool that helps team to keep track of: important documents, tasks that need to be completed, notes that were taken during Sponsor/Team meetings, etc.

- **Slippage**: The result of unforeseen delays in a project, causing baseline start or end dates to be postponed.

- **Software Design Specification (SDS):** Specifics regarding the implementation of the project. Outlines various design decisions, including: Architectural, Interface, Database, and Component Design.

- **Software Requirements Specification (SRS):** Complete description of behavior and requirements of system. May include use cases, wireframes, and a listing of functional and non-functional requirements.

- **System Test Specification (STS)**: Used to describe the team's plan for testing the software, and for specifying test cases and test procedures necessary to demonstrate that the software satisfies the requirements specified in the SRS.

- **Universal Modeling Language (UML):** A collection of tools for abstractly modeling software systems.

- **Use case**: An algorithmic description of a user's interactions with a system.

- **User manual (UM):** A deliverable written for the client, this document includes the recommended and basic use of system, as well as figures representative of use cases.

- **Testing**: The process of finding, avoiding, and detecting defects in the project.

- **Work Breakdown Structure (WBS):** A method of decomposing work to be done into measurable and manageable units.

## 1.4 References

Tutorialspoint.com

IEEE Template for System Requirement Specification Documents:

https://goo.gl/nsUFwy

## 1.5 Overview of Contents of Document

The following is a brief description of the sections remaining in this document:

**2 Project Overview**: The project overview describes the end result of the project as currently defined, as well as the process for evaluating and improving the SPMP during the course of the project.

**3 Project Organization**: Project Organization details the architecture of the project, including: the process model, organizational structure, interfaces, and project responsibilities.

**4 Project Management and Control**: With any project, plans must be managed, maintained, and assessed. Outlined in this section are the various methods, objectives, assumptions, and risks that this project is based upon.

**5 Technical Process**: The technical process model describes the methods that the team will use in representing the technical details of the project. In addition, the technical process outlines how the team will record and publish technical details during project development, as well as the version control process that will be used.

**6 Activities and Schedule**: The Activities and Schedule section detailsall phases and their associated tasks. The section also details the costs associated with each phase and the resources that team will require to complete the project.

**7 Approvals**: Signatures here indicate that team, and the project advisor understand, and agree to, the terms, conditions, and commitments contained in this SPMP.

# 2. PROJECT OVERVIEW

The project overview describes the end result of the project as currently defined, as well as the process for evaluating and improving the SPMP during the course of the project.

## 2.1 Project Summary:

The goal of this project is to produce a android application that allows the public to access information about available products needy for their pets. In pursuit of that goal, the team will work in coordination with the sponsor organization to continually verify the project's progress and direction. To that end, the team will prepare and deliver several documents for sponsor approval (Table 5) that describe the functionality and design of the software. The sponsor will also receive all code included in the finished project.

## 2.2 Evolution of the SPMP:

As the sponsor and team investigate the problem domain, new information may come to light that requires changing the processes or timelines established in this document. To ensure that such changes can be quickly and easily incorporated, the sponsor and team will set aside a time at each weekly meeting to review the project management process and discuss necessary changes. If changes are made, this document (and any other relevant documents) will be revised and a new version will be distributed to the sponsor and the team.

# 3. PROJECT ORGANIZATION

Project Organization details the architecture of the project, including: process model, organizational structure, interfaces, and project responsibilities.

## 3.1 Process Model

The project will begin with serial processes to focus on the gathering and elicitation of requirements. After requirements are gathered and well defined, the team will enter an iterative cycle during design and implementation. This process is to ensure all functionality will be correctly implemented and thoroughly tested.
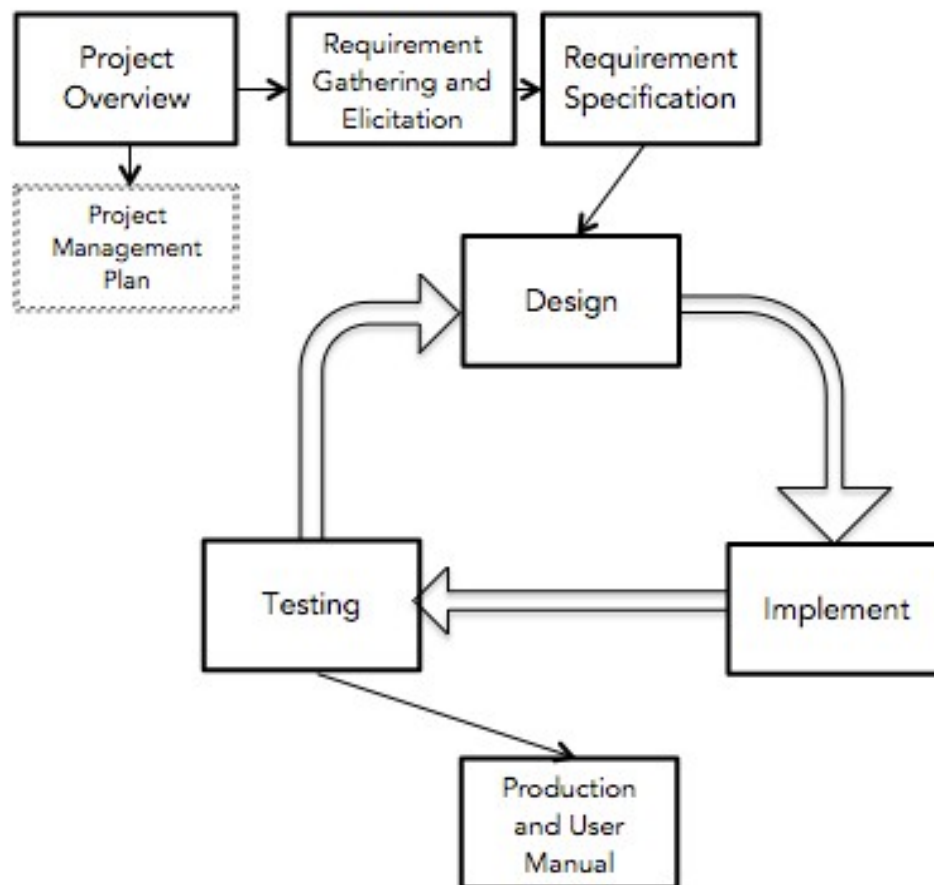


Figure 1. Process Model

## 3.2 Organizational Structure and Interfaces

Introduces the members of team and their roles and responsibilities.

| Name | Role & Responsibilities |
|---|---|
| Dr. K.S. Mann | Project Guide – Oversee the work the team produces and deal with any inquires they may have. Assign duties to the team, and ensure assignments are finished on time. |
| Manpreet Singh | Manage the look-and feel of the project, and deals with the front-end of the project. |
| Navjot Singh | Manage the front-end, back-end of the project, and ensure that the project's database is reliable. |
| Nikhil Sarna | Manage the documentation of the project, and deals with the back-end of the project . |

Table: Organizational Structure

## 3.3 Project Responsibilities

Describes each phase of the project and the member of the team who is in charge.

| Project Phase | Description | Phase Lead |
|---|---|---|
| System Concept Development Phase | Establish the vision and scope of the project. | Dr. KS Mann - As project guide, she will have a greater understanding of concept behind the project. |
| Planning Phase | Develop a plan to manage the project, gather the | Manpreet Singh,Nikhil Sarna, Navjot Singh – As they manage front-end and back- |

| | requirements and document the plan. | end, will give the team a greater idea of the time, and resources needed for the project. |
|---|---|---|
| Requirements Analysis Phase | Analyze user needs and developer requirement specifications. | Nikhil Sarna – As Document Lead, Nikhil Sarna will have a better understanding of how to break down the requirements into smaller parts. |
| Design Phase | Transform requirements intosoftware design document. | Navjot Singh, Manpreet Singh – She will create a design that addresses all requirements while still having a quality look-and-feel. |
| Development Phase | Develop and define the system test plan and preparing use case procedures. | Manpreet Singh, Nikhil Sarna, Navjot Singh – They will be in charge of documenting the procedures the team will take in making sure the system is usable. |
| Integration and Testing | Demonstrate that developed system conforms to Requirements specified in System Requirements Specification. | Navjot Singh, – He will work with Manpreet Singh in making sure that all parts of the System come together and are thoroughly tested. |
| | | |

| Documentation | Develop User's Manual. | Nikhil Sarna –He will lead in development of the User Manual, and ensure that the manual covers all use cases. |
|---|---|---|

Table: Project Responsibilities

# 4. PROJECT MANAGEMENT AND CONTROL

With any project, plans must be managed, maintained, and assessed. Outlined in this section are the various objectives, assumptions, and risks related to Project.

## 4.1 Project Management Objectives and Priorities

Team's vision for this project is to replicate the ease-of use associated with domain-specific private sector search applications in a public services search application. The key to this ease-of use is masking and abstracting complex functionality with a simple interface. This apparent simplicity can be a trap for development teams; one that Team will avoid through regular meetings with stakeholders, and careful documentation processes.

**Team Meetings**: The team will meet twice weekly, once on Tuesdays (10am – 11am) and once on Thursdays (10am – 11am). The Tuesday meeting will focus on planning and strategy, while the Thursday meeting will provide an opportunity for the team to work collaboratively on the project. The team lead will prepare an agenda for each Tuesday meeting, and a member of the team will take meeting minutes. Any key decisions made in these meetings (or any other meeting) will be recorded in the project's DTM. The team will conduct and record technical reviews of deliverable documents at this meeting.

At each Tuesday meeting, the team will create a 2-week WBS after assessing the previous WBS. The team will assess any slippage that has occurred and, in case of slippage, will update the project's baseline schedule. All assignments will be recorded in the minutes as "New

Assignments", and will appear as "Last Week's Assignments" on the next Tuesday's meeting agenda. The work to be done at Thursday meetings will be determined during the Tuesday agenda or minutes will be prepared for Thursday meetings, as they should result in tangible artifacts evidencing work completed.

**Supervisor meeting:** Every Tuesday, the team will meet briefly with their faculty advisor (10am – 11am). The team lead will prepare an agenda for this meeting. The team will give a brief oral report of their progress over the last week, and solicit advisement on upcoming work. The team will submit any documents requiring the faculty advisor's approval at these meetings.

**Project Log**: The team will collect certain management documents in a project log that will serve as a paper trail for the activities of the team over the life of the project. The project log is meant to aid the team in evaluating their progress, tracking key decisions, and meeting scheduled deadlines. The Project Log will include the following categories:

1. Team Meeting Agendas and Minutes
2. Supervisor Meeting Agendas, Minutes and Reports
3. Project Baseline Schedules
4. Technical Review Reports
5. Baseline Schedules and Updates
6. Change Requests
7. Requirements Traceability Matrix (RTM)

## 4.2 Assumptions, Dependencies, and Constraints.

This section describes the assumptions, dependencies, and constraints that this project is based upon.These assumptions are derived from the Project Charter.

- This project will be internet based.
- Time constraints include the length of the semester.
- Technical constraints include, but are not limited to: minimal support for legacy browsers, and scalability of deployment.
- Client is aware of time constraints and will regularly dedicate time on a weekly basis for collaboration and requirements elicitation throughout the process.

## 4.3 Risk Management

The goal of risk management is to identify and mitigate potential sources of expense or delay. Some risks are common to every project phase, and some risks are closely associated with a particular project phase. Risks for this project have been classified accordingly:

## Common Risks

**1 Team Member Unavailability:** During the course of a project, it is almost certain that some members of the team will be unavailable for certain project activities due to illness or emergency.

**Probability**: High

**Impact**: Low

**Prevention**: Team members should alert the team lead at the first opportunity regarding potential absences, and coordinate with other members of the team to cover their responsibilities for the duration of the absence.

**Correction**: Excessive (or unannounced) unavailability will trigger a team discussion. Continued unavailability will trigger discussion with faculty advisors and potential removal from the team.

**2 Miscommunication**: The volume of communication regarding this project almost guarantees that miscommunications will occur.

**Probability**: High

**Impact**: Medium

**Prevention**: The primary method of avoiding miscommunication is to document and verify verbal communications. For this project, the documentation and verification process will consist of meeting agendas, minutes and reports that will be shared with the guide within 48 hours of any meeting. Key decisions will be documented in the DTM. To avoid internal miscommunication, weekly team meetings will include time for informal socializing to help build relationships between team members, and provide a forum for airing grievances or concerns.

**Correction**: If miscommunications are occurring regularly, the guide and team will reassess the documentation and verification process.

**3 Changes to Project Scope**: Changes to project scope are a common request, but can derail project timelines.

**Probability**: High

**Impact**: High

**Prevention**: The Project Charter and this document define processes for handling change requests in a responsible manner.

**Correction**: The defined processes for change requests specify deadlines for such changes. Those deadlines will be honored to mitigate the risk of disruptive scope change.

**4 Missed Deadlines:** As students, Team face a number of external responsibilities that could detrimentally affect their ability to complete tasks on time.

**Probability**: Medium

**Impact**: High

**Prevention**: During weekly apportioning of responsibilities, team members are expected to anticipate their non-project-related time commitments and accept only work they can reasonably expect to accomplish by the given deadline.

**Correction**: Multiple failures to meet deadlines will trigger a team discussion. Continued failures will trigger discussion with faculty advisors and potential removal from the team.

**5 Mismanagement:** As students, the team is learning what project management requires, and mistakes will be made.

**Probability**: High

**Impact**: High

**Prevention**: Project leads are expected to elicit feedback from team members regarding their performance, and attend class sessions and make appropriate inferences regarding management from the information provided.

**Correction:** Project leads may be unseated by a majority vote within the team.

**6 Forgotten Stakeholders**: Project is intended to help a wide range of people with a diverse set of needs. It can be easy to omit relevant stakeholders when eliciting requirements.

**Probability**: High

**Impact**: Medium

**Prevention**: To help the team understand the diversity of the project's intended users, users will be categorized according to need. The team will research standards for defining public service needs in order to make those groupings comprehensive.

**Correction**: Incorporating the needs of omitted stakeholder groups after the requirements elicitation and analysis phase will qualify as a change of scope, and must be addressed via the change request process.

**7 Misunderstanding of Problem Domain:** There are many questions to be answered about how and why people search for public services on the web.

**Probability**: Low

**Impact**: High

**Prevention**: The team plans to conduct extensive research on this topic, including interviewing relevant stakeholders. The team will also use guide meetings and discussions to verify conceptions of the problem domain.

**Correction**: While work is in progress, the guide is expected to correct team misunderstandings when such misunderstandings are presented. If submitted work contains such misunderstandings, the sponsor is encouraged to submit a change request.

## Software Design Risks

**8 Incorrect Design:** Design is the foundation of implementation, and errors in design can create a cascade of defects.

**Prevention**: Design documents must be approved by the sponsor, and by the faculty advisor. Design documents will first face internal review within Team, where documents must be approved by their creator, and at least one other team member.

**Correction**: Errors in design will be corrected via revisions to design documents. Under no circumstances will design errors be fixed in implementation without design documents being revised to reflect corrections.

## Software Implementation Risks

**9 Lack of Experience with Relevant Technologies**: Team members have varying degrees of experience with the various technologies to be used in this project.

**Probability**: High

Impact: High

**Prevention**: Work will be assigned according to experience with the relevant technology. Team members are encouraged to ask questions when faced with something they don't understand. Team members with experience are expected to assist less experienced members. Team members are expected to independently research technologies during the summer break.

**Correction**: If a team member is repeatedly unable to meet deadlines due to lack of experience, the team will reassess the division of responsibilities.

**10 Inconsistency with Design.** The time pressures of the implementation phase often encourage shortcuts that deviate from the software design.

**Probability**: Medium

**Impact**: Medium

**Prevention**: This risk is greatly mitigated by a successful design phase. The team will keep design resources in view during implementation to attempt to stay focused on the design.

**Correction**: If the team struggles to stay consistent to the design during implementation, the team will discuss updating the design to more accurately model a feasible implementation.

**11 Incomplete Data Sources**: The data source for the Project web application is under development by the sponsor organization.

**Probability**: Low

**Impact**: Low

**Prevention**: Team has already offered some assistance in the design of the data source application. The probability of this risk is low because a volunteer from the sponsor organization has already implemented the basic functionality.

**Correction**: If the data source application is not completed before Project is due for guide approval, the team will create sample data sources to test and demonstrate the application.

## Software Testing Risks

**12 Non-Representative Tests**: Writing tests to ensure that software works as expected is one of the most difficult challenges of the Quality Assurance process.

**Probability**: High

**Impact:** Low

**Prevention**: Team will review current literature on testing processes and methods. Non-automated functional testing will also be conducted.

**Correction**: During development, new tests will be written to cover any uncovered defects detected in non-automated testing.

## Software Delivery Risks

**13 Incompatible Server Architecture**: The finished web application will be dependent on a server environment chosen by the sponsor and owned and maintained by a third-party. Such

environments vary in their support for web application programming languages and language versions.

**Probability**: Low

**Impact**: Low

**Prevention**: The User Manual deliverable will contain information regarding server requirements.

**Correction**: Deployment of the software to a public server is the responsibility of the sponsor.

## 4.4 Change Management.

This section breaks down the process the team willuse to manage changes to the project.

A change request must be submitted each time the client needs a change that will impact the project schedule. A change being defined as: any new requirement or change to an existing requirement. A change request will be sent to Team through either email or the Slack project collaboration tool. Team will acknowledge the request, assess the feasibility of the change, and discuss the priority of the change with the sponsor.

If the team decides to not implement the change, the sponsor will be notified, and the team and sponsor will sign the change request, indicating both parties' agreement that no change is to be made. If the team does decide to implement the change, documents that reference updated requirements will bemodified to reflect the change, and both parties will sign the change request, indicating their agreement to the change.

Each change request will be documented internally in the project's DTM. This document will detail the team decision regarding the request, and the date the team reached this decision.

## 4.5  Schedule Control

This section explains the process for how the team will approach scheduling.

A number of factors can change when the requirements of a project are completed. Therefore the schedule is an estimate of when requirements will be finished. Overall the project involves a serial life cycle to establish project requirements and then evolves into iterations during the later design, implementation, and testing phases. All iterations will be approximately 2 weeks each, depending on the involvement of the task. Iterations will be broken down into subtasks to be divided among team members. The team as a whole will evaluate the required time for specific tasks, and the team leader will decide a due date based on the team's estimate. Estimates will become more concrete as the project moves forward.

# 5. TECHNICAL PROCESS

The technical process model describes the methods that the team will use in representing the technical details of the project. In addition, the technical process outlines how the team will record and publish technical details during project development, as well as the version control process that will be used.

## 5.1 Methods, Tools, and Techniques

The Methods, Tools, and Techniques section aims to outline specific plans, methods, or tools to be used during the course of this project.

• **Diagrams**. Team will use standard UML diagrams to represent data, relationships, and requirements. Specific models and the requirements phase in which they are used can be found below.

| Phase | Description | Model/Diagram |
|---|---|---|
| Elicitation | The guide and development team will meet to identify the main 'actors' of the system and how they interact with the system. | Use Cases |
| Specification | Those developing the system will analyze the various 'entities' involved in creating the system and how they interact with each other. | Entity- Relationship Diagram |
| Specification | These diagrams will be used to outline the requirements of the system. Wireframes are to be developed by the team as a whole, and approved by the sponsor. | Wirefram |

• **Programming Languages & Tools**: Team will utilize various programming languages and tools. With the web-based nature of this project, Team will be using HTML, CSS, JavaScript, PHP and other tools, to be determined as they become necessary. Team will be using various pre-established web tools, including the Google Maps API, among others, as necessary.

## 5.2 Software Documentation

Description of Team's documentation plan.

•First Draft: during this first draft, the assigned Team Member will outline major sections and subsections. Questions raised during this phase will be addressed in team meetings. Any unanswered questions will be asked at faculty meetings.

• Second Draft: The rough draft will be expanded upon and made more specific. All raised questions will be resolved. This draft is to be shown tothe faculty advisor to receive initial feedback.

• Third Draft: The document is submitted to the faculty advisor for feedback. After the faculty advisor makes revisions, Team will address concerns and make necessary corrections.

• Final Copy: This iteration of the document will have all revisions andchanges. This document will then be sent to the various stakeholders outlined in this document, who will sign-off in agreement.

• Additional Changes: Changes after the final copy has been signed off must be approved by the development team and follow the change process outlined in Section 4.4 of this document.

## 5.3 Documents

The table below outlines the various documents associated with this project, a brief description, and the baseline delivery date.

| Document | Description | Delivery Date |
|---|---|---|
| Software Requirement Specification (SRS) | Complete description of behavior and requirements of system. May include use cases, wireframes, and a listing of functional/non- | |

| | | |
|---|---|---|
| | functional requirements. | |
| Design Documentation | Contains designs for implementation of the project. Outlines various design decisions Team has made, including: Architectural, Interface, Database, and Component Design. | |
| Software Project Management Plan | A formal document used as a guide for execution and control of project. Includes a list of team rules, hours of activity, a list of project issues and a work breakdown structure (WBS). | |
| System Test Specification | Used to describe the team's plan for testing the software, and for specifying test cases and test procedures necessary to demonstrate that the software satisfies the requirements specified in the SRS document. | |

Table: Documents

# 6 ACTIVITIES AND SCHEDULE

The Activities and Schedule section details all phases and their associated tasks.

Also details the costs associated with each phase and the resources that Team will require to complete the project.

**6.1 Activities and Tasks**. This section will be updated periodically as the development team progresses through the software development process. This section will initially include the WBS for the SRS document. Subsequent WBS will be prepared and distributed internally.

| Task | Task Designee | Baseline Duration | Baseline Start | Baseline Finish |
|---|---|---|---|---|
| Introduction | All | | | |
| Purpose &Scope | Nikhil Sarna | | | |
| Definitions | Navjot Singh | | | |
| References | All | | | |
| Overview of Documents | Nikhil Sarna, Manpreet Singh | | | |
| General Descriptions | Manpreet Singh | | | |
| Product Perspective | Navjot Singh | | | |
| Use Case Models | Manpreet Singh | | | |
| Interaction Design | Manpreet Singh | | | |

| | | | | |
|---|---|---|---|---|
| Specifications | | | | |
| User Characteristics | Nikhil Sarna | | | |
| Assumptions and Dependencies | Navjot Singh | | | |
| Specific Requirements | Nikhil Sarna | | | |
| Use Case Specifications | Manpreet Singh | | | |
| Performance Requirements | Navjot Singh | | | |
| Design Constraints | Manpreet Singh | | | |
| Quality Attributes | All | | | |
| Reliability | Nikhil Sarna | | | |
| Maintainability | All | | | |
| Program Quality | Navjot Singh | | | |
| Security | All | | | |
| Transferability | Manpreet Singh | | | |
| Operational | Navjot Singh | | | |
| Approvals | Nikhil Sarna | | | |

Table: Tasks & Activities Associated with SRS

## 6.2 Schedule

Contains a list of phases, deliverables, and the estimated date of completion.

| Phase of Work | Deliverable | Est. Date |
|---|---|---|
| Requirement analysis | Software Requirement Specification (SRS) | |
| Designing | Design Documentation | |
| Planning | Software Project Management Document | |
| Development | Deployment strategy | |
| Testing | Software Testing plan | |