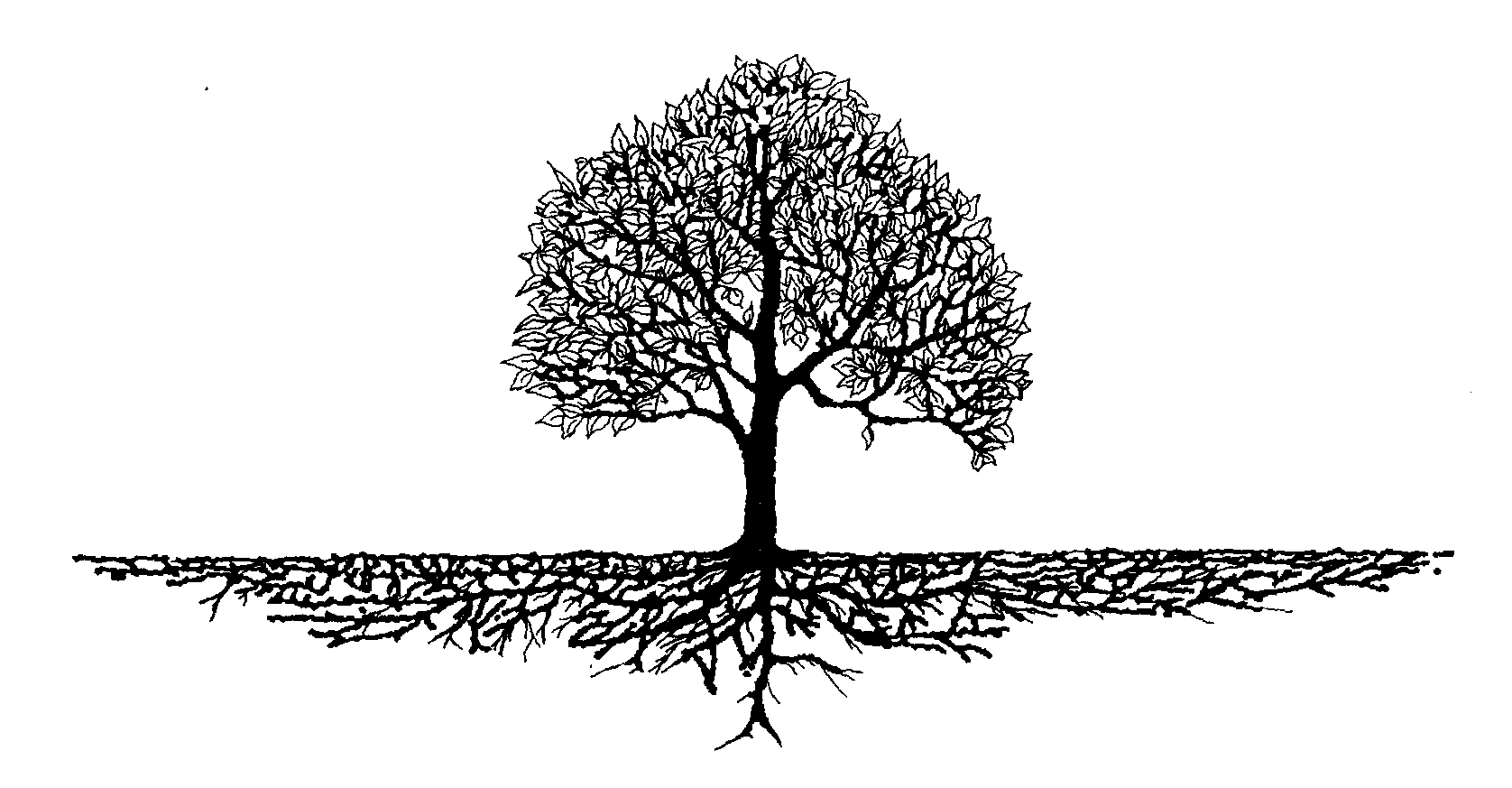
Software Development Plan



Jonathan MacKenzie

Bradley Baker

Christopher Fevrier

Mark Podrouzek

Table of Contents

[Documentation Scope 3](#_Toc494271832)

[Pillars of the Development Process 3](#_Toc494271833)

[Design Coordination: 3](#_Toc494271834)

[Server Development: 3](#_Toc494271835)

[Client Application Development: 3](#_Toc494271836)

[Testing & Documentation: 3](#_Toc494271837)

[Personnel & Duties 3](#_Toc494271838)

[Software Life Cycle (Dates & Timelines) 4](#_Toc494271839)

[Major Activities 4](#_Toc494271840)

[Deliverables 4](#_Toc494271841)

[Risk Management & Mitigation 5](#_Toc494271842)

[Design & Prototyping 6](#_Toc494271843)

[Technical Information 6](#_Toc494271844)

[System Boundaries 6](#_Toc494271845)

[Procedures & Techniques 6](#_Toc494271846)

[Review & Inspection Techniques 7](#_Toc494271847)

[Testing Techniques 7](#_Toc494271848)

[Documentation Information & Contributions 7](#_Toc494271849)

# Documentation Scope

The software development plan is a collective document accumulating all information required to manage the project. This document is maintained and updated throughout the development and life cycle of the project. Found within are the team’s responsibilities and deadlines for each aspect of the software.

# Pillars of the Development Process

## Design Coordination:

This pillar involves the collection and presentation of the suggestions brought upon by internal & external, while also providing information such as weighing risks and rewards. As well as being the lead discusser within negotiations between the development group & the clients, and updating requirements accordingly.

## Server Development:

This pillar involves programing, implementing, and maintaining the server infrastructure to accommodate the design requirements of the application.

## Client Application Development:

This pillar involves programing, maintaining, and exporting the client side software to accommodate the design requirements of the application. As well as debugging and fixing bugs found within the code.

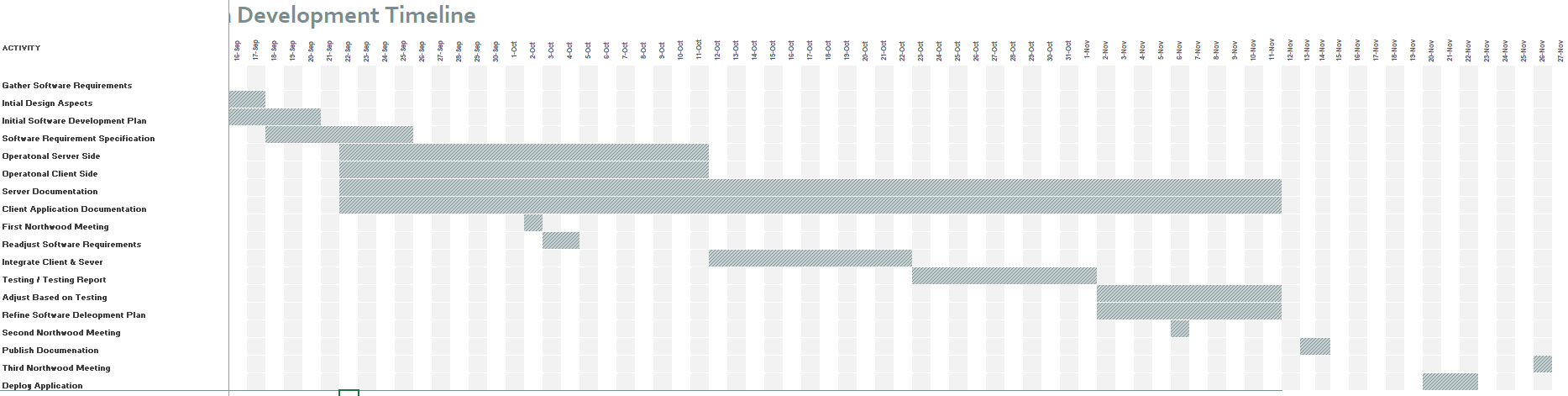
## Testing & Documentation:

This pillar involves strenuous testing of the prototypes programs, and properly reporting causes and symptoms of imperfections found. This pillar is also involved with collecting and editing the technical documentation of the application.

# Personnel & Duties

|  |  |
| --- | --- |
| Name | Main Pillar |
| Jonathan MacKenzie | Server Development |
| Bradley Baker | Design Coordination / Testing & Documentation |
| Christopher Fevrier | Client Application Development |
| Mark Podrouzek | Design Coordination / Testing & Documentation |

# Software Life Cycle (Dates & Timelines)



Tracking is found within the major activity table and in the Timeline.xlsx found within the project GitHub repository.

# Major Activities

|  |  |  |
| --- | --- | --- |
| Date | Group Member | Description of Process |
| September 11 2017 | Bradley Baker | Creating of the Development Plan |
| September 12 2017 | Jonathan Mackenzie | Creation of the GitHub Repository |
| September 13 2017 | All | Group Meeting |
| September 13 2017 | Mark Podrouzek | Completion of Concept logo and art for Application |
| September 13 2017 | Christopher Fevrier | Starting front end application |
| September 13 2017 | Jonathan MacKenzie | Starting back end for application |
| September 20 2017 | Bradley Baker | Initial Draft of Software Development Plan Complete |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Deliverables

|  |  |
| --- | --- |
| Date | Documentation Description |
| November 14 2017 | Software Development Plan |
| November 14 2017 | Software Requirements |
| November 14 2017 | Source Code |
| November 14 2017 | Installation Plan |
| November 14 2017 | Testing Documentation |

# Risk Management & Mitigation

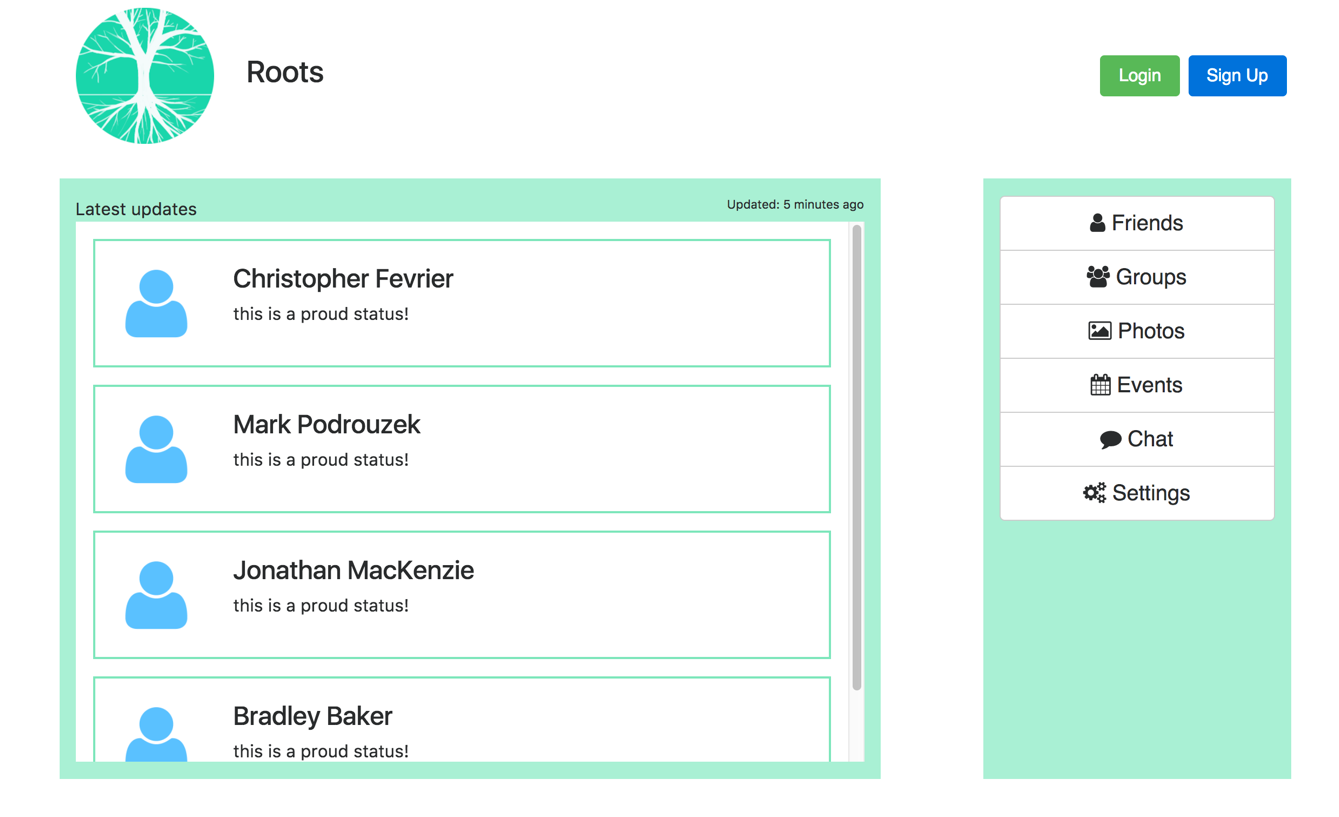
Many risks exist within the development cycle of a piece of software. Outlined here are multiple risks identified by the software development team and how the team would respond to such issues. Priority rated on a scale from 1-5. 5 being most critical and/or most likely.

|  |  |  |
| --- | --- | --- |
| Risk | Priority | Response |
| Extended Absence of Member | 1 | The member will continue to work from a remote location, updating to the group using communication software. |
| Permanent Absence of Member | 5 | The remaining members of the group will re-assign priorities and required tasks per member. |
| Timelines Overdue | 3 | Extend timelines as required ad reassign tasks to ensure they are done in a timely fashion. |
| Real Time Performance Failures | 3 | Problems will be identified to the service provider and the team will develop using a separate server until the issues are resolved. |
| Development Plan Denied | 1 | The team will adjust the development plan according client request, while keeping in mind physical restrictions. |
| Additional Development Requirements | 3 | If new requirements arise not yet accounted for in the software development plan, The team will add it to the project. Or revisit the initial requirements to allow for a timely completion date. |
| Loss of Information | 1 | The team will continue to back up and create copies of documentation to eliminate the risk of data loss. |
| Developing the Wrong User Interface | 1 | The team will attempt to accommodate the clients limitations in the design of the product. |
| Client Scheduling Delays | 1 | The team will accommodate for rain dates that may be required to meet with the software clients. |
| Unforeseen Technical Limitation | 3 | To be adjusted based upon the situation in which the issue arises. |

|  |  |  |
| --- | --- | --- |
| Risk | Probability | Justification of Probability |
| Extended Absence of Member | 70% | The approaching flu season. |
| Permanent Absence of Member | 10% | Group seems unlikely to remove themselves from the class, or be forces from the class. |
| Timelines Overdue | 30% | A lot of slack has been implemented in the plan |
| Real Time Performance Failures | 50% | Server or Application errors |
| Development Plan Denied | 5% | Low chance the client rejects the application plan. |
| Additional Development Requirements | 75% | Likely the client will request more features for the application. |
| Loss of Information | 10% | Consistent Backups and a GitHub repository. |
| Developing the Wrong User Interface | 10% | Talk with client to minimize this risk |
| Client Scheduling Delays | 15% | The client is forced to reschedule meeting. |
| Unforeseen Technical Limitation | 20% | Unknown source, but possible an issue unexpected arises. |

# Design & Prototyping

User Interface:



# Technical Information

## System Boundaries

The application will be created with the intent to run using a web browser. However, there is a possibility based upon need and time provided to export the application to mobile devices.

The client side application will be written using HTML 5 and JavaScript.

The server side will be written using Node JS.

## Procedures & Techniques

The server side and client side will be completed separately, and then brought together once both have working prototypes.

## Review & Inspection Techniques

The group will review the functionalities of the application to ensure it meets all predetermined requirements. In addition a working prototype will be presented to the client for feedback.

## Testing Techniques

Research is being done into testing with Travis CI. Any issues found will be shared with the team and swiftly dealt with.

# Documentation Information & Contributions

|  |  |
| --- | --- |
| Task | Members Contributions |
| Documentation Writing & Formatting | Bradley Baker |
| Documentation Editing | Johnathan MacKenzie |
| Documentation Revising | Bradley Baker |
| Testing Research | Johnathan MacKenzie |
| Timeline Dates | All Members |
| Gantt Chart | Bradley Baker |
| Risk Identification | All Members |
| HTML Concept Interface | Christopher Fevrier |
| Concept Interface | Mark Podrouzek |
| Logo & Artwork | Mark Podrouzek |