horizontal line

Software Development Plan

**30th January 2017**

# PROJECT OVERVIEW

The *Trucking Company Management System* is an application that maintains and manages data for a medium-to-large sized Trucking Company. It will be available to every employee at all times.

The objective of this project is to develop a system, which will maintain detailed information current and historical for a major trucking company. Records must be maintained for a ten-year period. To protect the privacy and security of company information users of the system will have access to information only on a “need to know” basis.

# PROJECT SCHEDULE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Requirements | Analysis | Design | Implementation | Testing |
| Sprint 1 | Read Them Thoroughly | Discuss with team meetings |  |  |  |
| Sprint 2 | Learn about backlogs | Break requirements into Epics & User stories | Begin design of model classes | Write base classes and interfaces | Instantiate classes and call get/sets |
| Sprint 3 |  |  | Complete design of all objects | Write Web Service and Implement Database | Call query functions (serialize, etc) |
| Sprint 4 | Produce a GUI conforming to requirements | Consider customer needs | Design all user controls for views | Hook up GUI to functions | Play with it |
| Sprint 5 | Reiterate through requirements |  | Design additional functionality as needed | Finish beta, then final release | Extensive user acceptance testing |

## 

# LIST OF DELIVERABLES

|  |  |
| --- | --- |
| Date | Deliverable |
| 30-Jan-2017 | Software Development Plan/Presentation |
| 15-Feb-2017 | First Backlog |
| 6-Mar-2017 | Architectural Design/2nd Backlog Version |
| 29-Mar-2017 | Preliminary GUI Design |
| 19-Apr-2017 | Application Delivery Date |

# MILESTONE

## SDP submission

* Emailed to Dr. Coleman
* Presentation completed

## Model Classes Established and Written

* Define Model Classes

**Database Finished**

* Implement Model Classes into Data Storage

## Web Services Written

* Implement connectivity between central server and active clients
* Enforce authentication and permissions of personnel

**Data Access Methods Established and Written**

* Search Queries
* Permissions

**GUI**

* Preliminary GUI Design

**Project Submission**

* Completed Backlog
* Full source submission
* Compiled Application Submitted
* All in a Zip File

# PERSONNEL

|  |  |
| --- | --- |
| **Role** | **Team Member(s)** |
| Software Lead | Donal Cavanaugh |
| Requirements Analyst | James Williamson |
| Customer Liaison | Zach Johnson |
| Designer | James Williamson, Donal Cavanaugh, Zach Johnson, John Mullen |
| Programmer | James Williamson, Donal Cavanaugh, Zach Johnson, John Mullen |
| Test Designer | James Williamson, Donal Cavanaugh, Zach Johnson, John Mullen |
| Tester | James Williamson, Donal Cavanaugh, Zach Johnson, John Mullen |

# 

# 

# RISK MANAGEMENT PLAN

|  |  |  |
| --- | --- | --- |
| **Event** | **Risk Level** | **Management** |
| Loss of a team member (illness, fired, etc.) | Critical to Catastrophic | Redistribute workload among remaining members |
| Loss of data | Negligible | Keep multiple backup copies of project files and documents |
| Underestimating Workload | Marginal | Scheduling make-up sessions and Reducing scope of project |
| Team Misunderstands requirements | Marginal | Requirements review scheduled for multiple sessions |
| Learning curves cause delays | Critical | Start reading and learning \*early\*.  Consult other team members often. |
| Loss of Network Connectivity | Marginal | Have an alternative localhost or script to produce one dynamically |
| Hosted system lacks needed system components | Marginal | Test on multiple machines and bring multiple to the presentations.  Verify when necessary. |
| Customer rejects finished application | Critical | Fix problems that customer has with application |