Project Plan

Date: 25, August, 2013

*Hive Management System*

Perry Hartman

Wesley Flowers

William Pegg

*Nathan Lawson*



Professor Pourshsbound

Software Engineering Department

Southern Polytechnic State University

Marietta, GA 30060

Table of Contents

[1. Introduction 3](#_Toc372671921)

[2. Project Overview 3](#_Toc372671922)

[2.1 Scope 3](#_Toc372671923)

[2.1.1 Identification 3](#_Toc372671924)

[2.1.2 System Overview 3](#_Toc372671925)

[Document Overview 4](#_Toc372671926)

[2.2 Background, Objectives, and Scope 4](#_Toc372671927)

[2.3 Operational Policies and Constraints 6](#_Toc372671928)

[2.4 Description of Current System or Situation 6](#_Toc372671929)

[2.5 Users or Involved Member 6](#_Toc372671930)

[2.6 Support Concept 7](#_Toc372671931)

[3. Features, Primary Deliverables, and External Commitments 7](#_Toc372671932)

[3.1 Feature List 7](#_Toc372671933)

[3.2 Customer Deliverables 8](#_Toc372671934)

[4. Project Schedule 8](#_Toc372671935)

[4.1 Major Project Milestones 8](#_Toc372671936)

[4.2 Detailed Project Schedule 9](#_Toc372671937)

[5. Project Work and Product Estimates 12](#_Toc372671938)

[5.1 Estimate Summary 12](#_Toc372671939)

[6. Project Resource Requirements 17](#_Toc372671940)

[6.1 Staffing/ Skill Requirements 17](#_Toc372671941)

[7. Risk Management 18](#_Toc372671942)

[7.1 Risk Management Strategy 18](#_Toc372671943)

[7.2 Initial Risk List 18](#_Toc372671944)

[7.2.1 Risk Details: 18](#_Toc372671945)

[We have used the list of potential risks to calculate the risk value of each risk. The risk values below represent the probability that the potential risk may or may not occur and the cost that the risk would present to the project if it did occur. We have used these potential risks when determining the buffer for our project which is represented in the estimation portion of this document. 18](#_Toc372671946)

[8. Project Quality and Measurement Plans 19](#_Toc372671947)

[9. Project Process 19](#_Toc372671948)

[9.1 Software Life Cycle Model 19](#_Toc372671949)

[10. Referenced Documents 20](#_Toc372671950)

# Introduction

Client statement- “The management system must be able to handle the registration of all users, groups, boards, uploads of files and down loads of files as well as version control for files. This includes the contact information, skill set, basic personal information, and the job/role information for the user and groups. The system will also be used for all kinds of planning and resource management for the groups (everything from version control to the assignment of roles for group members). During the actual project, the system must be able to recognize the same file and alterations done within it, display the current standings for each “file” (version, timestamp, and user that altered the file). The system will be completed by Thanksgiving weekend in November.”

The management system will consist of a communication network involving boards for each related group. The boards will have threads that will expand and collapse, which will send and/or receive data to a dashboard and register.

# Project Overview [[1]](#footnote-1)

## Scope

### Identification

The software project will consist of a web site for use on both Firefox and Chrome. The website will have to major actor: admin and the user. The admin will be able to change the website as needed as well as maintenance. The basic user will be able to create a profile for his or her self. He or she will also be able to create a group or groups. By default the creator of the group will be made lead/manager. The leader/manager role can be changed at anytime to any other group member. Once a group has been formed the website will generate the threads board along with a thread. Both of which can be altered be the manager/lead. The manager/lead can add threads as needed by the group. The group can add as many roles/positions by the leader to fit the needs of the group. That allows each group to be made of 'n' number of members and tailored to the needs of the group. The school is largely windows based, so due to resource availability and familiarity an application developed using an Ubuntu sever is the best support for the end product. Because the website management system is largely based on information gathering the best architectural design to use in implementation will be shared data store for data management and project driven for the GUI engine. The data storage will be persistent and server based using SQL Server utilities.

### System Overview

The project manager specification of the product design will be Linux based. The GUI design must be as simple as possible, utilizing standard web representation is a great GUI for achieving an astatically efficient interface. The unit specific to the project manager will essentially track all workings of the system including uploaded files, downloaded files, project schedules, threads, groups, and version control. As well as track the unit will also allow the project manager to adjust, schedule, add and delete these workings in the system. The unit will have capability of alerting project manager by user based on projects. The unit will also communicate with other members in the group by sending and receiving the threads updates.

The group managers will need access to the group options and settings. While using a PC, the manager reports all statuses of the members and all statuses of the files. The manager will be able to authorize a group member for a particular role, remove a member from a particular group, and track each member in a particular group, view group schedule, and view member roles in the group. The group manager will utilize the thread page to communicate with group members, project manager specifically along with all other members. The group manager will also be able to communicate any member’s status by direct. The update dash board will also allow the group manager to view members in each group, view group schedule, and project schedule.

A member will be able to view each group, view the group schedule, view each member in each group, and report any status of a project. The member will also be able to communicate by the threads sending and receiving with all other members.

The project logger will only be tied to the group that made it. The schedule will allow the members to track each group, track each member, view group schedules, view project schedule, view group stats, report group stats, report status of each group, report status of each member, send a group schedule change request, communicate with all other member by use of the threads sending and receiving, and display applicable information on the project display board.

### Document Overview

This document is meant to be a tool and blue print used to assist the project manager and the development team in the planning and organizational tasks of the Hive Management System. Specifically, meant for the project manager and to facilitate him or her in the management process, this document is not meant for distribution to end user, client, or the development team. It is not a deliverable.

## Background, Objectives, and Scope

1. Project Manager -

Background

The Project Manager is by default the same member the created the group. This role can be based to any other member of the team at any time. The Project Manager is responsible for the following objectives.

Objectives

• Track all members

• Track all Groups

• Schedule Groups for Project Milestones

• Update Project Milestones

• Update Groups for Project Milestones

• Sequence any tasks during Project Milestones

• Assign member to tasks

• Alert member of any tasks

• Alert member of any updates

• Schedule Project Milestones

• Track all Groups

Scope

The scope for the project manager’s role is variable and depends of the size of the project and team as well as the number of groups he or she will be responsible for.

2. Group Manager

Background

The group manager is responsible for his or her group only. The responsibility objectives are as follows.

Objectives

• Authorize qualification of members to particular group

• Track each member in particular group

• Report status of each group

• Report status of each member

• Add a member to a group

• Delete a member from a group

Scope

The scope for the Group Manager’s role is variable and depends of the size of the project and team.

3. Update Dashboard

Background

The updater is part of the system that tracks the user’s events and reports them to the other group members.

Objectives

• Report member status

• Track each group

• Track each member

Scope

The scope for the updater is variable and depends of the size of the project and team.

4. Website logger (PC)

Background

The Logger is a function of the website the logs all events generated by the user. The purpose of the logger is to sent updates to the dashboard.

Objectives

• Track each group

• Track each member

• Operate display board

• Report status of each group

• Report status of each member

Scope

The scope for the logger is variable and depends of the size of the project and team.

## Operational Policies and Constraints

The system will not operate outside of the project. It is not meant to be used for any purpose other than to assist the project members. It is specifically designed to track project member, files, members, groups, and scheduled projects during the scheduled project.

## Description of Current System or Situation

1. The operational environment consists of several stationary members. Each user is in charge of management of project participants who could range from members to groups. All these characteristics of the environment will all occur on the Hive sever.
2. The major components of the system consist of a sever, PC lap tops, dash board, and a shared database. Each component will interact via the shared data base and network. The dash board will primarily interact with the data store through the logger. Each other components will interact directly with the database and website. The project manager, members; both have read and write capabilities. All other components have read only capabilities.

## Users or Involved Member

SPSU Staff

• Good with hands, hands on

• Technically proficient

• May not be typing proficient

• High range in age

• English may not be their first language

• High range in ethnicity

• Set working hours

• Set job description

• Will be familiar with Campus

• Will be familiar with project

• Not gender dependent

SPSU Teachers

• High IQ

• Age range from 29 to 55

• Most likely detail oriented

• Proficient typing skills

• Will be familiar with Campus

• May be familiar with project

• May be disabled in some way

• High level of leadership

• Most likely very organized

• High range in ethnicity

• Not gender dependent

SPSU Students

• High IQ

• Age range from 17 to 38

• Proficient typing skills

• Will be familiar with Campus

• May be familiar with The Hive Project

• Not gender dependent

• High range in ethnicity

• May be technically proficient

• May be disabled in some way

• English may not be their first language

All users will be feeling the roles of project manager, Website logger, group manager, Website marshal, concession stand operator, project nurse, and project security.

## Support Concept

Product support will be provided by the development company, trouble shooters guides, and user guides.

# Features, Primary Deliverables, and External Commitments

## Feature List

The project manager web application will be in charge of tracking all project member and groups. The project manager specification of the product design will be web based. As well as track the project will also allow the project manager to adjust, schedule, add and delete these aspects in the system. The project manager will be able to also review real time reports of the member capacity, member capacity, project schedules, group schedules, product distribution, the project threads log, member schedule, and working schedule. The website will have a capability of alerting project manager by updating the privileges. The website will also alert the project manager of any negative issues or members or member that might occur at real time. The website will also communicate with other groups such as the logger, group manager, and all websites by sending and receiving the threads.

The group manager website will be capable of authorizing members for particular groups, and tracking each member in a particular group. The website will also be able to add or remove a member to or from a particular group. The website will also be capable of viewing the project schedule and group schedule as well as sending and receiving the threads.

The website will be able to report status of all members in the current group. Track each member in that particular group, and track each group and communicate by messaging in the threads.

The concession stand operator will be able to send the threads to all project members. The website will also have the capability of reading in information from bar codes. It will track water consumption and provide alerts to user on price changes in concession stand products.

The logger will allow the logger to track each group, track each member, view group schedule, view project schedule, view group stats, report group stats, report status of each group, report status of each member, send a group schedule change request, communicate with all other member by use of the threads sending and receiving, and display applicable information on the project display board.

Product Requirements:

Please refer to the SRS in section 3 Specific Requirements.

## Customer Deliverables

• User Guide (system): overall book to assist user, based on overall functionality of system and overall system component interaction.

• Troubleshooter Guide (system): overall book to assist user and system support, based on possible system discrepancies, inconsistencies and risks that may occur and how to approach them. For both end user and system support

• User Guide (project manager, PC): overall book to assist user, based on overall functionality of project manager website.

• User Guide (website, PC): overall book to assist user, based on overall functionality of website.

• Design Document: detailing system design and architecture

• Requirements Document: detailing the overall requirements of the system

• Product Support contract: detailed description of product support criteria as well as cost, scope, and length of service

• Bill of sale: cost of product

• Phase 1 (rapid) prototype: first prototype, only includes user interface and interactions between them

• Phase 2 (rapid) prototype: second prototype, includes phase1 plus website capabilities

• Phase 3 (iterative) prototype: third prototype, includes all basic system functionality

• Product warranty: includes a detailed description of product void and return policies

• Website management system software package (executable)

• Website management system software package (source code)

• Website management system software package (data store)

# Project Schedule

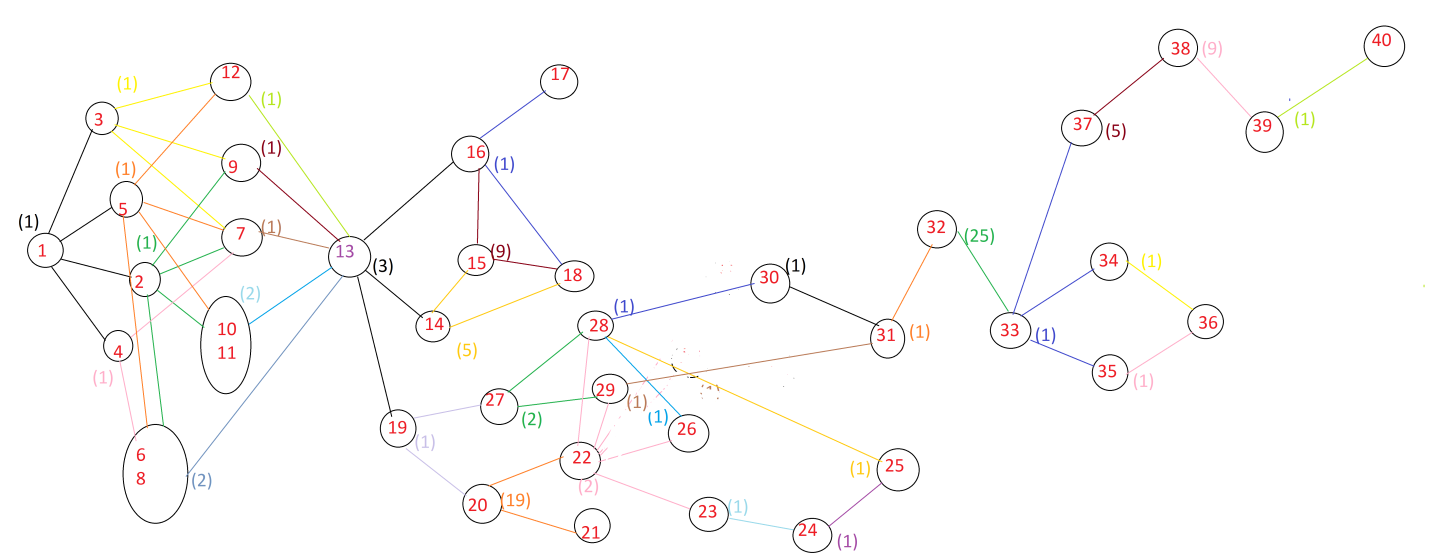
## Major Project Milestones

| **Date (**YYYY-MM-DD) | **Milestone/ Project** | **Entry Deliverable & Criteria** | **Exit / Notes** |
| --- | --- | --- | --- |
| 2013-08-22 | Establish System Requirements | System goals, constraints, attributes, metric, notes from customer, requirement related questions | Requirements well defined  Initial requirements document |
| 2013-09-10 | Risk assessment | Risk prioritization, risk assessment, risk analysis, all possible risks | Risk analysis document formed |
| 2013-09-12 | Detailed Design Document | Planned design, system constraints, attributes, requirements document | Detailed design document formed |
| 2013-10-08 | Prototype 2.0 | All functionality for websites, data store, and display board deemed testable | Prototype 2.0 tested |
| 2013-11-12 | Product Red Zone | Product design, code, test cases, requirements, plan, production artifacts | Product finished by this date or 5 milestones after |

## 4.2 Detailed Project Schedule

|  |  |  |
| --- | --- | --- |
| Idex | Date | Milestone |
|  | By August 22, 2013 | initial assessment of system requirements and deliverables are defined and established (quick plan) |
|  | By August 27, 2013 | all actions desired for each paid employee shall be well defined including job title, job requirements, work rotation, pay, designated restrictions |
|  | By August 29, 2013 | all member requirements will be well defined, including all required information, restrictions, and needed designated actions taking on their behave by system for each group |
|  | By September 3, 2013 | the requirements for Website will be well defined including job requirements, work rotation, and designated restrictions |
|  | By September 5, 2013 | the requirements for project managers will be well defined including organizational actions, all managerial actions, designated access, designated restrictions, and job qualifications and requirements |
|  | By September 5, 2013 | the requirements for concession stand operators will be well defined including actions needed, information needed, designated restrictions, work routine, and qualifications. |
|  | By September 10, 2013 | establish system requirements, scope of project, constraints, external and internal components involved, tools needed, system attributes well defined, deliverables required, and processes and methodologies needed. Form Requirements Document (1st milestone) |
|  | By September 10, 2013 | skills matrix as well as a hiring matrix will be established and defined with all required member well defined |
|  | By September 12, 2013 | all member needed for project are met in accordance to skills matrix |
|  | By September 12, 2013 | establish potential pitfalls in process define potential risks for the system and a risk assessment and risk prioritization time period will be defined and established. Establish risk mitigation techniques. |
|  | By September 17, 2013 | re access requirements and development process, examine internal and external product properties. Re access steps and tasks needed for the overall development process. Assign milestones |
|  | By September 17, 2013 | estimate effort required, resources needed. Form WBS |
|  | By 17, 2013 | form detailed design document and establish most ideal system design, form initial test cases and matrices for each |
|  | By September 30 2013 | all GUI interfaces built |
|  | By September 30, 2013 | a prototype 1.0 (rapid) user interface will be developed for each designated staff member including project managers, group managers |
|  | By October 3, 2013 | a prototype 1.1 (rapid) user interface including functionality, boards will be developed for test purposes for the designated end user operation. Form test cases for member registration and management. |
|  | By October 15, 2013 | project team meeting to form test cases from discrepancies brought up through phase one and two of rapid prototype testing. |
|  | By October 17, 2013 | Midterm presentation of the prototype includes all documentation. |
|  | By October 22, 2013 | functionality for all desktop websites is deemed testable |
|  | By October 24, 2013 | functionality for data store is deemed testable |
|  | By October 28, 2013 | all test cases concerning member registration and management are satisfactory |
|  | By October 28, 2013 | all test cases concerning member registration and management are satisfactory |
|  | By October 28, 2013 | all test cases concerning member registration and management are satisfactory |
|  | By October 28, 2013 | a prototype 2.0 (iterative) with all different end user functionality and requirements developed and tested using all end users with desktop units. |
|  | By October 28, 2013 | all test cases concerning member registration and management are satisfactory |
|  | By October 28, 2013 | registration and management are satisfactory |
|  | By October 28, 2013 | project team meeting, develop test cases established by prototype 2.0 test, all test cases should be fully defined, understandable, and measurable. |
|  | By November 5, 2013 | all test cases established by prototype 2.0 are improved and satisfied. |
|  | By November 7, 2013 | User Manual is fully developed for each end user |
|  | By November 7, 2013 | Trouble Shooter Manual is fully developed for the Website Management System |
|  | By November 11, 2013 | functional system evaluation meeting with user manual, trouble shooter manual and end users |
|  | By November 14, 2013 | system design evaluated by 2nd party, there assessment noted and evaluated |
|  | By October 14, 2013 | project meeting over 2nd party evaluation, more test cases established |
|  | November 14, 2013 | all test cases formed from 2nd party evaluation have been assessed and deemed satisfactory. System is ready for distribution |

Scheduled Goals PERT chart:



Critical paths:

* 1 + 2 + 10 + 13 + 19 + 20 + 22 + 23 + 24 + 25 + 28 + 30 + 31 + 32 + 33 + 37 + 38 + 39 + 40
* 1 + 2 + 11 + 13 + 19 + 20 + 22 + 23 + 24 + 25 + 28 + 30 + 31 + 32 + 33 + 37 + 38 + 39 + 40
* 1 + 2 + 6 + 13 + 19 + 20 + 22 + 23 + 24 + 25 + 28 + 30 + 31 + 32 + 33 + 37 + 38 + 39 + 40
* 1 + 2 + 8 + 13 + 19 + 20 + 22 + 23 + 24 + 25 + 28 + 30 + 31 + 32 + 33 + 37 + 38 + 39 + 40
* 1 + 4 + 10 + 13 + 19 + 20 + 22 + 23 + 24 + 25 + 28 + 30 + 31 + 32 + 33 + 37 + 38 + 39 + 40
* 1 + 4 + 11 + 13 + 19 + 20 + 22 + 23 + 24 + 25 + 28 + 30 + 31 + 32 + 33 + 37 + 38 + 39 + 40
* 1 + 4 + 6 + 13 + 19 + 20 + 22 + 23 + 24 + 25 + 28 + 30 + 31 + 32 + 33 + 37 + 38 + 39 + 40
* 1 + 4 + 8 + 13 + 19 + 20 + 22 + 23 + 24 + 25 + 28 + 30 + 31 + 32 + 33 + 37 + 38 + 39 + 40
* 1 + 5 + 10 + 13 + 19 + 20 + 22 + 23 + 24 + 25 + 28 + 30 + 31 + 32 + 33 + 37 + 38 + 39 + 40
* 1 + 5 + 11 + 13 + 19 + 20 + 22 + 23 + 24 + 25 + 28 + 30 + 31 + 32 + 33 + 37 + 38 + 39 + 40

Description:

The list of critical paths listed above are composted of multiple combinations of similar paths and nodes with one exception of the fourth position in which is always node 13. Each node is a basic representation of an event that occurs. The critical paths are the events that must be executed to complete the project.

# Project Work and Product Estimates

## Estimate Summary

**Project Estimation**

The Project Estimation includes the cost, effort and time estimates for the project using the Process Based Estimation Technique.

**Historical Data**

The historical data used for this estimate is average labor rate information based on data from the Bureau of Labor Statistics. This is because there is no previous development project data to base our estimates on; therefore these statistics will serve as a baseline for our estimate.

Job Title: Software Development

Experience:

Lowest 10 percent: $54,360

Median: $87,790

Highest 10 percent: $133,110

Source: US Bureau of Labor Statistics, Occupational Employment Statistics

Job Description for Software Engineers as defined by the US Bureau of Labor Statistics:

Software Developers typically perform the following functions:

* Analyze users’ needs, then design, test, and develop software to meet those needs
* Recommend software upgrades for customers' existing programs and systems
* Design each piece of the application or system and plan how the pieces will work together
* Create flowcharts and other models that instruct programmers how to write the software’s code
* Ensure that the software continues to function normally through software maintenance and testing
* Document every aspect of the application or system as a reference for future maintenance and upgrades
* Collaborate with other computer specialists to create optimum software

The US Bureau of Labor Statistics does not make the distinction among Software Engineers regarding their roles in requirement analysis, development, or quality assurance. We will therefore use the data for Software Developers as applicable to the Requirements Analyst, Developer, and Quality Assurance Analyst roles for our project. The US Bureau of Labor Statistics does, however, make a distinction between Software Developers and Systems Developers. We will therefore use the data for systems developers as the historical data for the Software Architect.

Job Title: Systems Development

Experience:

Lowest 10 percent: $61,040

Median: $94,180

Highest 10 percent: $143,330

Source: US Bureau of Labor Statistics, Occupational Employment Statistics

Job Description for Systems Engineers as defined by the US Bureau of Labor Statistics:

Systems software developers create the systems that keep computers functioning properly. These could be operating systems that are part of computers the general public buys or systems built specifically for an organization. Often, systems software developers also build the system’s interface, which is what allows users to interact with the computer. Systems software developers create the operating systems that control most of the consumer electronics in use today, including those in phones or cars.

To calculate the proper estimates for the Project Manager of this project, we will use the historical data for Computer and Information Systems Managers as defined by the US Bureau of Labor Statistics.

Job Title: Computer and Information Systems Manager

Experience:

Lowest 10 percent: $71,420

Median: $115,780

Highest 10 percent: $166,400

Source: US Bureau of Labor Statistics, Occupational Employment Statistics

Job Description for Computer and Information Systems Managers as defined by the US Bureau of Labor Statistics:

Computer and information systems managers typically do the following:

* Analyze their organization’s computer needs and recommend possible upgrades to top executives
* Plan and direct installing and upgrading computer hardware and software
* Ensure the security of an organization’s network and electronic documents
* Assess the costs and benefits of a new project to justify spending to top executives
* Learn about new technology and look for ways to upgrade their organization’s computer systems
* Determine short- and long-term member needs for their department
* Plan and direct the work of other IT professionals, including computer systems analysts, software developers, information security analysts, and computer support specialists
* Negotiate with technology vendors to get the highest level of service for their organization

Our project team consists of a Project Manager, a Requirements Analyst, a Software Architect, two Software Developers and a Quality Assurance Analyst. Since four members of our six member team for this project are categorized as Software Developers, we are able to use the median rates for Software Developers to calculate the estimated labor rate. Since our Quality Assurance Analyst is not experienced in the application development, we will apply a 15% correction to the median salary. The estimated labor rate per person-week according to the historical data will be:

Software Developers: $((87,790 / 52)(1 + 1 + 1 + 0.85)/4 = $1,624.96 per week

To calculate the estimated labor rate per person-week for the Software Architect we will use the median rate for Software Systems Developers, which is $94,180.

Software Architect: $((94,180 / 52)(1) = $1,811.15 per week

To calculate the estimated labor rate per person-week for the Project Manager we will use the median rate for the Computer and Information Systems Manager, which is $115,780.

Project Manager: $((115,780 / 52)(1) = $2,226.54 per week

We will now further break down each position to its hourly rate, so that these rates can be used to calculate the total person-hours can be calculated in our estimate.

Software Developers: $(1,624.96 / 40) = $40.62

Software Architect: $(1,811.15 / 40) = $45.28

Project Manager $(2,226.54 / 40) = $55.66

**Process Based Estimation**

Desktop Application Interface DAI

Desktop Application Logic DAL

Website Logic WL

System Architecture SA

Data Store Engineering DSE

Bug Fixing BF

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **Customer Communication** | **Planning** | **Risk Analysis** | **Engineering** | | **Construction** | | **Customer Support** | **Totals** |
|  |  |  |  |  |  |  |  |  |  |
| **Task** |  |  |  | **Analysis** | **Design** | **Code** | **Test** |  |  |
| *Function* |  |  |  |  |  |  |  |  |  |
| DAI | 0.1 | 0.04 | 0.02 | 0.05 | 0.15 | 0.2 | 0.4 | 0.25 | 1.21 |
| DAL | 0.05 | 0.12 | 0.07 | 0.11 | 0.19 | 0.3 | 0.34 | 0.15 | 1.33 |
| SA | 0.03 | 0.2 | 0.14 | 0.19 | 0.48 | 0.02 | 0.09 | 0.08 | 1.23 |
| DS | 0.03 | 0.17 | 0.11 | 0.15 | 0.45 | 0.09 | 0.35 | 0.2 | 1.55 |
| BF | 0.04 | 0.04 | 0.03 | 0.09 | 0.05 | 0.1 | 0.21 | 0.38 | 0.94 |
|  |  |  |  |  |  |  |  |  |  |
| **Total** | **0.25** | **0.57** | **0.37** | **0.59** | **1.32** | **.71** | **1.39** | **1.06** | **6.26** |
|  |  |  |  |  |  |  |  |  |  |
| **% Effort** | **6.17** | **8.46** | **5.49** | **9.26** | **19.43** | **11.77** | **20.46** | **18.97** | **100.00** |

Based on the calculated hourly rates for the Project Team we have broken the tasks down into the required effort for each task which is measure in person-hours. The breakdown of the costs can be seen in the table below, and will be used in allocating the project budget.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WBS: Budget Allocation Table** | |  |  |  |  |
| ID | Task Name | Assigned | Work (Milestones) | Effort (Hours) | Cost |
| 1 | Assess System Requirements & Define Deliverables | Requirements Analyst | 7 | 56 | $2,274.72 |
| 2 | Assess System Requirements & Define Deliverables | Project Manager | 7 | 56 | $3,116.96 |
| 3 | Define Member Roles | Requirements Analyst | 0.25 | 2 | $81.24 |
| 4 | Define Concession Stand Requirements | Requirements Analyst | 0.25 | 2 | $81.24 |
| 6 | Define Member Requirements | Requirements Analyst | 0.25 | 2 | $81.24 |
| 7 | Define Website requirements | Requirements Analyst | 0.25 | 2 | $81.24 |
| 8 | Define Project Manager requirements | Requirements Analyst | 0.25 | 2 | $81.24 |
| 9 | Define Group Manager requirements | Requirements Analyst | 0.25 | 2 | $81.24 |
| 11 | Define Project Nurse Requirements | Requirements Analyst | 0.25 | 2 | $81.24 |
| 13 | Define Display Board Requirements | Requirements Analyst | 0.25 | 2 | $81.24 |
| 14 | Define System Requirements | Requirements Analyst | 1 | 8 | $324.96 |
| 15 | Create Skills Matrix | Project Manager | 1 | 8 | $445.28 |
| 16 | Create Hiring Matrix | Project Manager | 1 | 8 | $445.28 |
| 18 | Define potential pitfalls and risks | Project Manager | 3 | 24 | $1,335.84 |
| 19 | Assign Project Milestones | Project Manager | 1 | 8 | $445.28 |
| 20 | Form WBS | Project Manager | 2 | 16 | $890.56 |
| 21 | Finalize Detailed Design Document | Software Architect | 5 | 40 | $1,811.20 |
| 22 | GUI Design for Website | Software Architect | 3 | 24 | $1,086.72 |
| 23 | Form Initial Test Cases and Matrices | QA Analyst | 3 | 24 | $974.88 |
| 24 | Develop Prototype 1.0 (Rapid) | Developer (2) | 7 | 112 | $4,549.44 |
| 25 | Logger User Desktop Functionality | Developer (2) | 4 | 64 | $2,599.68 |
| 25 | Develop Prototype 1.1 (Rapid) | Developer (2) | 5 | 80 | $3,249.60 |
| 27 | Form Phase 1 & 2 Test Cases | QA Analyst | 3 | 24 | $974.88 |
| 29 | Form Test Cases for Desktop Website | QA Analyst | 1 | 8 | $324.96 |
| 32 | Test Desktop Functionality | QA Analyst | 2 | 16 | $649.92 |
| 33 | Test Data Store Functionality | QA Analyst | 2 | 16 | $649.92 |
| 34 | Develop Prototype 2.0 (Rapid) | Developer (2) | 7 | 112 | $4,549.44 |
| 35 | Establish new test cases for Prototype 2.0 | QA Analyst | 1 | 8 | $324.96 |
| 36 | Test Prototype 2.0 | QA Analyst | 3 | 24 | $974.88 |
| 37 | Develop User Manual for each end user | Developer (2) | 1 | 16 | $649.92 |
| 38 | Develop Trouble Shooter Manual | Developer (2) | 1 | 16 | $649.92 |
| 39 | Establish 2nd party eval Test Cases | QA Analyst | 1 | 8 | $324.96 |
| 40 | Perform 2nd part eval Test Cases | QA Analyst | 3 | 24 | $974.88 |
|  | Buffer |  |  |  | $8,000.00 |
| **Total Labor Cost:** | | | | | $44,909.48 |

\*We have included an $8,000 buffer to cover potential risks, whether they are anticipated or not.

# Project Resource Requirements

## Staffing/ Skill Requirements

**Human Resources**

The team of individuals assigned to work on this project will consist of six individuals that will occupy the roles listed below:

**Skills Description of Member Needed**

|  |  |  |
| --- | --- | --- |
| Job Title | Number Needed | Experience |
| Project Manager | 1 | At least four years of experience in leading software project teams of at least 5 people and successfully completing six month projects. |
| Requirements Analyst | 1 | At least five years of experience in software development and one year of requirements analysis experience.  Also, they must have experience working with IBM Rational Software. |
| Designer | 1 | At least three years of experience in application domain, two years of experience working with system architecture and five years of development experience. |
| Programmer | 3 | At least three years of development experience |
| Test Analyst | 2 | At least five years of experience in software development and at least two years of experience in Quality Assurance. |

The table below displays how the Human resources will be distributed throughout the duration of the project.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Initial Skills Matrix** | |  |  |  |  |  |  |  |  |  |
| Skilled Member | Weeks | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Requirements Analyst | 1 | 1 | 1 |  |  |  |  |  |  |  |
| Software Architect |  |  | 1 | 1 |  |  |  |  |  |  |
| Developers |  |  |  | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Testers |  |  |  |  |  | 1 | 1 | 1 | 1 | 1 |
| Project Manager | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Total: | 2 | 2 | 3 | 4 | 3 | 4 | 4 | 4 | 4 | 4 |

# Risk Management

## Risk Management Strategy

The team has created a list of potential risks that could possibly occur during the project and will use these risks to develop a plan to mitigate those risks.

## Initial Risk List

This is the initial risk list; the risks are listed in priority order from top to bottom. Descriptions are provided below the table. Risks with a pre-mitigation magnitude of 2.0 or below are not listed.

**Potential Risks:**

|  |  |  |
| --- | --- | --- |
| Risk Item | Problem Recovery Cost | Risk Priority |
| Executable Code is not completed by the deadline | High | High |
| A developer leaves the project causing a delay in the coding | High | Medium |
| Learning curve for new test management software causes delays | Medium | Low |
| Some equipment is dysfunctional | Medium | Medium |
| Several High Severity bugs have not been fixed by due date | Medium | High |

### Risk Details:

### We have used the list of potential risks to calculate the risk value of each risk. The risk values below represent the probability that the potential risk may or may not occur and the cost that the risk would present to the project if it did occur. We have used these potential risks when determining the buffer for our project which is represented in the estimation portion of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Risk Item | Probability of Occurrence | Recovery Cost | Risk Value (RV) |
| Executable code is not completed by the deadline | 0.6 | $2,500 | 1500 |
| A developer leaves the project causing a delay in the coding | 0.3 | $1,800 | 540 |
| Learning curve for new test management software causes delays | 0.5 | $1,500 | 750 |
| Some equipment is dysfunctional | 0.2 | $2,000 | 400 |
| Several High Severity bugs have not been fixed by the due date | 0.2 | $1,100 | 220 |

# Project Quality and Measurement Plans

The architectural design the team will use to implement 'The Hive' Website management system will be a blend of a shared-data architectural style and a project driven architectural style.

A shared-data store is ideal because of the little variation of system components to track and manage, and the knowledge once one component is updated, deleted, or added, the same component must be updated, deleted, or added throughout all other websites. Keeping all data in a central repository can also assure proper security and data quality. The shared data store is also independent which increases maintainability, fault tolerance, and robustness. The base reason of the advantages of the shared-data style architecture is also its greatest disadvantage. Because the shared-data store is so independent and gives the system the ability to store data in a centralized place, the system greatly depends on the data store. If the shared-data store goes down, so does the entire system, which is the main reason why most if not all shared-data architectural styles have some kind of back up website to store data once the data store is down for maintenance or malfunction. The management of the Website, is essentially tracking all members, staff, and all other aspects of the project and making sure the right project member know the workings at all times. For this reason, the shared-data architectural design was considered, because as changes in project status arise, those changes can be represented as data in the shared-data store. When a data-store accessor reads and or writes the data needed to be updated, all other websites with proper access connected to the shared-data store will receive the update in real time.

Because of the nature of the project itself, the system must also utilize a project driven architectural style as well. The shared-data architectural style is essentially for information gathering and communicating that information throughout the system. That’s not the only aspect of the system to worry about. The system primary goals are to assist the end user in a way that advances his or her capabilities. In order for the system to accomplish this, the system must be usable. The system must mimic an assistant or secondary person on the job working right with the end user. Assistants do many things like show initiative, organize data; provide a second hand in assistance. The project driven design aspect of the system is meant to make the system usable, make the system more like an assistant, not just a glorified file cabinet. The project driven design aspect of the system will be responsible for issuing warnings, processing related data, alerting the user of certain conditions, and triggering projects that need to be triggered even when the user doesn’t request them.

# Project Process

## Software Life Cycle Model

The life cycle of the project is water fall based. In any case of any process for any project, it must make use of all tools and resources acquired in the most optimal way while achieving each goal specified in a timely and efficient manner. In order for each goal to be achieved, each goal must be thoroughly understood, meaning not only does the need of the end user need to be well defined, but also all aspects and conditions of the system must be understood and considered. Both potential internal and external product structures must be thoroughly examined. All external entities to the system must be established, defined, and considered.

In order to achieve a sufficient software product, the system must be easily installed and easily usable, maintainable, execute accurately according to product design and end user requirements. To achieve this, the planning aspect of the life cycle is essential. In some cases the customer may not be fully knowledgeable about their complete needs or concerns about the system, in other cases just might fail to communicate properly with product developers. We as providers may misinterpret what the customer is asking for. The customer may forget or not take the planning phase of the process seriously. During the life cycle of the project essentially there are three iterations of the planning and requirements phase of the water fall model. To have a greater understanding of the customer needs, the project goals are gathered as a whole initially. They are then broken down into pieces, and covered with the customer to achieve maximum clarification. Basically, learn part to whole of what is expected of the system being designed.

Once, all three iterations of the planning and requirements aspect of the life cycle are completed, the life cycle then switches to the design phase. As in the planning phase it is still very important for the customer to be heavily involved in the Design phase as well. Also like the planning phase, the design phase is essentially three iterations of the design phase. Each phase produces a prototype to be tested, and bit by bit, part to whole; the prototype becomes more robust and has added functionality until essentially, the prototype mimics the desired system in detail. Coding occurs between each prototype. Basically, a design of a prototype, a test run, test cases formed, adjustments made, and then a generation of a new improved design based off of a greater understanding of the customer’s needs and desires.

The testing phase realistically is merged with the design phase, and occurs essentially within the three iterations of the design phase as well as extends further to preform further testing. After the testing of the last prototype, during this period is essentially where most of the coding occurs for the end product. Once the coding based off prototype 2.0 use cases are done, the product is then tested by a 2nd party development team. This team could be internal or external as long as they aren’t directly associated with the system being designed. Test cases are formed based of the 2nd party testers and the system is recalibrated based off the testers input.

Once the testing phase is coming to a close, the maintenance and review phase of the life cycle begins. In this phase the development team briefs the customer on the product and assesses the development team performance through self-evaluations. The customer will receive their user guides and trouble shooter manuals at a briefing with the development team. They will be fully explained, demos will be given, and system considerations, capability, and scope will be defined. There will be given a lecture on their customer support and maintenance contract, as well as a walkthrough on project costs, and deliverables. PSP, Personal Software Process, and all other self-evaluation tools will also be utilized during this phase.

# Referenced Documents

Project Plan Part I (Perry Hartman, Wesley Flowers, William Pegg, Nathan Lawson)

1. [↑](#footnote-ref-1)