**Student Information Management System**

June 5, 2017

**Team Members**

Scott Murray

Scott Blick

Javier Rodriguez

Ahmed Shogar

Thanh Bui

Document Control

**Change History**

|  |  |  |
| --- | --- | --- |
| **Revision** | **Change Date** | **Description of changes** |
| V1.0 | 6/20/17 | Initial release |
| V1.1 | 7/3/17 | Updated information |

**Document Storage**

This document is stored on Github at:

https://github.com/UHD-SoftwareEngineering-Summer2017/Final-Project

**Document Owner**

Thanh Bui is responsible for developing and maintaining this document.

# Table of Contents

# 1 OVERVIEW 4

# 1.1 Purpose and Scope 4

# 1.2 Goals and Objectives 4

# 1.3 Project Deliverables 5

# 1.4 success criteria 6

# 2 STARTUP PLAN 6

# 2.1 Team Organization 6

# 2.2 Project Communications 7

# 2.3 Technical Process 7

# 2.4 Tools 7

# 3 WORK PLAN 8

# 3.1 Resource Estimate 8

# 3.2 Release Plan 8

# 3.3 Iteration plans 9

# 4 CONTROL PLAN... 9

# 4.1 Monitoring and Control 9

# 4.2 Configuration and Management Plans 10

# 5 SUPPORTING PROCESS PLANS 11

# 5.1 Risk Management Plans 11

# 5.2 Test Plans 11

# 5.3 Project Acceptance Plan ………………………………………………………………..... 11

# Overview

## Purpose and Scope

The purpose of this assignment is to design an interface that allows user to sign in as either a student or administrator with different functions for their respective role

The UI allows the user as a student to login view their grades, classes, and calculate their GPA, while the admin is allowed to view the students grade, classes, adjust the grades, add or remove student from the roster.

The user interface will be intuitive and simple to navigate, with a simple login and password request. Following that depends on either the user is a student or admin that it allows certain functions to be used

## Goals and Objectives

The overall objective is to create a database that allows the user to login and view the information of the user and adjust information base on who logged in

Project Goals:

1. Create an UI that functions as expected, looks great, and earns an A.
2. Give Professor Chang an app to show off as an example.
3. Learn about software engineering and creating an app.

Project Objectives:

1. Create a UI that give User access account information
2. Create an app that functions in a simple and intuitive manner.
3. Provide students to view and calculate their GPA.
4. Provide admin to view and edit students information

## Project Deliverables

|  |  |
| --- | --- |
| **Date** | **Deliverable** |
| 6/12/17 | Requirements Specification |
| 6/12/17 | Project Plan |
| 6/14/17 | Iteration #1 Plan |
| 6/23/17 | Technical Prototype |
| 6/25/17 | Architecture Document |
| 6/25/17 | Iteration #1 Complete |
| 7/3/17 | Test Report |
| 7/3/17 | Iteration #2 Complete |
| 7/4/17 | User Guide and System Administration Manual |
| 7/4/17 | Product Released |

## Success Criteria

A working prototype, which is easy to use, that allows users to sign in either as a student to view grades, view classes , and calculate GPA or administrator to add students view student information and edit student information.

# Startup Plan

## Team Organization

|  |  |  |
| --- | --- | --- |
| **Role** | **Actor(s)** | **Responsibility** |
| Project Manager | Ahem, Scott B. | Call team meetings, coordinate communications within group, coordinate communications outside group, break out tasks, assign them to teammates |
| Developer | Scott B, Scott M. | Develop software based on requirement and architect specifications |
| Programmer | Scott B, Scott M. | Program to requirement and architect specifications |
| Tester | Scott B, Scott M. | Write test cases, perform unit testing of test cases against incremental release of code, perform integrated testing of test cases against incremental release of code, report issues |
| Architect | Scott B, Scott M. | Specify overall internal workings of application |
| Requirement Engineer | Thanh, Javier | Outline and document project dependencies and requirements. This includes internal and external dependencies. |

## Project Communications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event** | **Information** | **Audience** | **Format** | **Frequency** |
| Team Meeting | Task status: completed since last meeting & planned for next;  obstacles encountered; change requests in process | All team members | Informal meetings following class; Formal meetings as needed; E-mail status updates & problems as they occur | As needed |
| Project Status Report | Review finished items, status of prototype; review any problems, schedule slippage, programming issues | All team members | E-mail with information or In-person | Iteration Closeout |

## Technical Process

An iterative and incremental development process is planned. Feedback will be used from each iteration to improve the next. The first iteration will focus on basic functionality of the application. Subsequent iterations will build upon that and incorporate more features as time allows.

## Tools

*Java coding - Intelli-J IDE, Netbeam*

Documentation -Excel, power point, Microsoft word

Argo UML

Web Base Database-Github.com

Communications -SMS Messaging

# Work Plan

## Resource Estimate

Detailed resource estimates are available in this document; tasks, roles, owners, and effort are listed.

## Release Plan

### Plan By Feature

**Iteration #1 6/14 – 6/245**

**Summary:** Demonstrate fundamental architecture

|  |  |  |
| --- | --- | --- |
| Features / Deliverables | **Estimated Effort** | **Actual Effort** |
| Working UI | 50 |  |
|  |  |  |

**Iteration #2 6/25 – 7/3**

**Summary:** Demonstrate functional code and program

|  |  |  |
| --- | --- | --- |
| Features / Deliverables | **Estimated Effort** | **Actual Effort** |
| Working Gui | 70 |  |

**Iteration #3 6/25 – 7/4**

**Summary:**

|  |  |  |
| --- | --- | --- |
| Features / Deliverables | **Estimated Effort** | **Actual Effort** |
| Successful test cases | 30 |  |

**Features not scheduled, but under consideration**

|  |  |
| --- | --- |
| Features | **Estimated Effort** |
| Prototypes |  |

### 

## Iteration Plans

### First Iteration

Demonstrate fundamental architecture that complies with the customer needs .

### Second Iteration

Utilize the first iteration and create a GUI that demonstrates the architecture with test cases the customer needs.

### Final Product

Tested program that fulfills the requirements, working architecture that is easily understood and bug free program

# Control Plan

## Monitoring and Control

The following list of dates includes formal reviews outside of the Communication Plan. Milestones are included to reference where the project is scheduled to stand as these reviews occur:

|  |  |
| --- | --- |
| **Date** | **Review / Milestone** |
| *6/23/2017* | *Milestone: Technical Prototype Complete* |
| 6/24/2017 | 5-Minute Status Report |
| 6/25/2017 | Manager's Briefing |
| *6/25/2017* | *Milestone: Iteration #1 Complete* |
| *6/25/2017* | *Milestone: Test Report Complete* |
| 6/30/2017 | Inspection |
| *7/1/2017* | *Milestone: Iteration #2 Complete* |
| *7/3/2017* | *Milestone: Product Released* |
| 7/3/2017 | Final Presentations |

## Configuration Management Plan

The following procedure is to be used when making changes to all baselined work products:

1. All project work products will be stored on Github
2. All baselined documents will have a Document Control section with a change history to track initialization and subsequent changes.
3. All project work products will be stored on Github but not all will be under change control .Only the system requirements, project plan and source code will be baselined and under configuration control.
4. Items that are subject to change control will be considered baselined after a group review at the end of the initial document creation.
5. The change control procedure once a product is baselined is:

(1) anyone wanting to make a change to a baselined item sends an email to the rest of the team describing the change, reason for the change, expected schedule impact, and time line for integrating the change.

(2) if no one responds to the group within 2 days with a reason for why the change request shouldn't be permitted, it will be considered accepted and the person proposing the change may proceed with the change.

1. if anyone does object to the change, the reason for objecting will be discussed at a meeting where everyone is invited to attend and voice their opinion. At the end of the meeting a democratic vote will be held to decide whether or not the change should be allowed.
2. if a change takes place, the initiator must collaborate with the project manager to update the schedule

# Supporting Process Plans

## Risk Management Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rank** | **Risk** | **Probability of Loss** | **Size of Loss** | **Risk Exposure** | **Response** |
| 1 | Schedule / time line delivery | Likely | Major | High | Mitigate: Stick to the schedule. |
| 2 | Learning curve for new tools and technologies longer than expected | Likely | Moderate | Moderate | Avoid: Breaking the parsing engine out into a separately hosted server allows for fixes to occur in one place, instantly for all users, and with minimal intrusion. |
| 3 | Creating a UI using java to allow a user to login as either an admin or student | Unlikely | Moderate | Moderate | Begin working on a basic prototype early to test out fundamental programming concepts & knowledge |
| 4 | User not able to login and use proper functions | Unlikely | Minor | Low | Avoid: apply proper functions that allow user to login correctly as either admin or student with proper tools |

## 

## Test Plan

The test plan defines the items that will be tested, methods for testing, and a schedule detailing the tasks, owners, and time line.

## Product Acceptance Plan

At the conclusion of each iteration, the prototype created will test to ensure it meets the requirements of that iteration.

For the final iteration, product acceptance testing will ensure that the prototype functions as expected with a user's data.