Gamification of Data

***Project Management Plan***

**Senior Design Project - CIS 4951 - Fall 2019**

**Version No. 1.0**

Team Members:

Ryan Ross (Team Lead)

Cameron Kozan

Zaid Alsafi

Alex Navarre

**Project Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Date | Revising Author | Description of Revision |
| 1.0 | 11/2/2019 | Cameron Kozan | Document Created |

|  |
| --- |
| **Table of Contents** |

[**1.0 Project Overview/ Introduction**](#_v81ftvfmib83) **5**

[1.1 Project Scope](#_orgkxpxhw7vb) 5

[1.2 Major Software Functions](#_rdgqjr5hrkau) 5

[1.2.1 Inputs](#_izrabkq7joz3) 5

[1.2.2 Processes](#_lbqphzrp0ugm) 5

[1.2.3 Outputs](#_lffbupei1els) 5

[1.3 Performance/Behavior Issues](#_he46pyfvih) 5

[1.4 Management and Technical Constraints](#_rp0ncblyo47c) 5

[1.5 Product and Process Models](#_s0cqi0aiaeyr) 6

[1.6 Methods, Tools, and Techniques](#_s35afn4plc5) 6

[**2.0 Project Budget and Resource Estimation**](#_6r77i7146eai) **6**

[2.1 Historical Data Used for Estimates](#_603luj9271tf) 6

[**2.2 Estimation Techniques Applied and Results**](#_o2vi9lrt3imc) **6**

[2.3.1 Estimation Technique 1 – Lines of Code](#_dy8jxdjj76u5) 6

[2.3.2 Estimate for Technique 1 – Lines of Code](#_xijntmyxr3w9) 6

[2.3.3 Estimation Technique 2 – Function Points](#_6g6u8jqvqy4e) 6

[2.3.4 Estimate for Technique 2 – Function Points](#_bd1vk3hutres) 7

[2.3.5 Estimation Technique 3 – Process/Task](#_nkvc7esd0761) 7

[2.3.6 Estimate for Technique 3 – Process/Task](#_3xjekg16e5ux) 7

[2.3 Reconciled Estimate](#_fgr9jnoaqbk6) 7

[2.4 Hardware/ Software Resource Needs](#_c6potpcbwwe) 7

[**3.0 Risk Analysis/ Management**](#_mftvunn53dp6) **8**

[3.1 Project Risks](#_9zce3jy0f4eb) 8

[3.2 Risk Table](#_k9jfy0r2jnjz) 8

[3.3 Overview of Risk Mitigation, Monitoring, and Management](#_q4sz3drvq0gy) 9

[**4.0 Project Schedule**](#_a4gveaecrrq) **9**

[4.1 Project Task Set](#_2f47wrac8j3t) 9

[4.2 Functional Decomposition](#_osel0aiy7lw5) 9

[4.3 Task Network](#_imm0qe5bz7wk) 10

[4.3 Project Schedule](#_j3t2mblmkmca) 10

[**5.0 Staff Organization**](#_lt00q2d7ql4r) **11**

[5.1 Team Organizational Structure](#_b0yyrgwcqmrf) 11

[5.2 Management Reporting and Communication](#_a9uhzwgdcdqu) 11

[5.3 Team Member Responsibilities](#_4x7vr6l0nsi) 11

[**6.0 Tracking and Control Mechanisms**](#_g1tisua7yo8o) **12**

[6.1 Quality Assurance and Control](#_1ht1n2ew2p5r) 12

[6.1.1 Objective](#_rp8149tsmg9l) 12

[6.1.2 Test Strategy](#_l8647wylwtwh) 12

[6.1.3 Checkpoint](#_5ffmarpsge2y) 12

[6.2 Change Management and Control](#_ywkgzv1y0z3o) 13

[**7.0 Appendix**](#_dhgezg977xeb) **13**

[7.1 Appendix A - Staffing plan](#_ng1xknesd7s0) 13

# 1.0 Project Overview/ Introduction

## 1.1 Project Scope

To create a webpage for companies to present easily digestible data for their employees to view and track metrics. The purpose of this is to create a friendly and competitive environment where workers will be motivated to change and develop better behaviors and practices determined by the company.

## 1.2 Major Software Functions

### 1.2.1 Inputs

* Login Info
* Employee Data
* Weight values

### 1.2.2 Processes

* Retrieve data from database
* Verify login
* Choose user view
* Apply score weight

### 1.2.3 Outputs

* Display user view
* Display employee leaderboard
* Display employee specific information

## 1.3 Performance/Behavior Issues

The webpage should be able to retrieve the correct data from the from the database server using the Invisi-Tag API. It should retrieve, and update, twice a day to present current and accurate data for the users. If performance falls below specification or if the software behaves incorrectly, execution of the following three guidelines should occur:

* Data retrieval and update timers will reset after timeout period is reached
* Current page will refresh after an extended number of exceptions encountered
* An error message screen will be presented to the user when the program acts out of specification

## 1.4 Management and Technical Constraints

The major managerial constraint that our team faces is the overall timeline of the project. Each team member has a full class load as well as work outside of school. We were given eight months to create a functioning webpage that incorporates an existing backend set of software. We are required to learn new languages and foreign development environments in a short amount of time.

## 1.5 Product and Process Models

We will be using an agile development approach to allow for greater ability to handle changes in the project. There will be five main phases to keep on track, Client selection and requirement gathering, Design, Prototype, Implementation, and Testing and Client acceptance.

## 1.6 Methods, Tools, and Techniques

The main languages we are planning on using are C#, .NET, AngluarJS, SQL, and ReactJS. Some of the tools we are using are Microsoft SQL Server, Microsoft Azure, Eclipse, Visual Studio & Visual Studio Code. We are using Agile development to allow for quick action to changes in the requirements given by the client. We are going to be following general programming guidelines such as comments, clean code, easy readability, modularity.

# 2.0 Project Budget and Resource Estimation

## 2.1 Historical Data Used for Estimates

* School & Work Projects

# 2.2 Estimation Techniques Applied and Results

### 2.3.1 Estimation Technique 1 – Lines of Code

Account Creation = 800 LoC

Dashboard = 2000 LoC

Employee/Team Page = 1200 LoC

Inventory = 600 LoC

Login Page = 300 LoC

Password Recovery Page = 900 LoC

Job Details = 1300 LoC

Total = 7100 LoC

### 2.3.2 Estimate for Technique 1 – Lines of Code

Total = 8000 LoC

### 2.3.3 Estimation Technique 2 – Function Points

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Complexity** | | |
| **Component** | **Low** | **Average** | **High** |
| **EI** | **3** | **4** | **6** |
| **EO** | **4** | **5** | **7** |
| **WQ** | **3** | **5** | **6** |
| **EIF** | **5** | **7** | **10** |
| **ILF** | **7** | **10** | **15** |

EI = 4\*6

EO = 5\*5

EIF = 7\*0

ILF = 10\*3

79\*9= 711 Hours

Total= $53,325

### 2.3.4 Estimate for Technique 2 – Function Points

Total = 780 Hours

Total = $58,500

### 2.3.5 Estimation Technique 3 – Process/Task

Account Creation = 90 Hours

Dashboard = 200 Hours

Employee/Team Page = 120 Hours

Inventory = 100 Hours

Login Page = 70 Hours

Password Recovery Page = 70 Hours

Job Details = 80 Hours

Total = 730 Hours

Total = $54,750

### 2.3.6 Estimate for Technique 3 – Process/Task

Total = 790 Hours

Total = $59,250

## 2.3 Reconciled Estimate

Our final decision ended up being 760 hours total which would cost $57,00. We made this our final estimate because it was the closest to our Process/Task Estimates and out of all the estimates we believe that this will inform us that we have a lot of work to do.

## 2.4 Hardware/ Software Resource Needs

GitHub

AWS

Discord

Visual Studio Code

Eclipse

Visual Studio

Draw.io

Azure Microsoft

SQL

C#

Java

.Net

AngularJS

# 3.0 Risk Analysis/ Management

## 3.1 Project Risks

There are a number of risks going into this project. Some of them include: scheduling, communication along with new functionalities.

## 3.2 Risk Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Probability | Impact | Mitigation | Contingency Plan |
| Scheduling | high | high | Good communication and multiple meeting times possible to include everyone | If a person/people can't make it, keep them in the loop. Make certain everyone knows what they are responsible for. |
| Working on a project created by another team | high | medium | Figuring out how their code works along with integrating new code with theirs. | Members will have to learn a new language to understand what the current functionality does. |
| New functionality arises | high | high | Create an outline of what is in the scope of our work and what is not. | Refer to the created document to see if the functionality is within scope. If it is not in scope, they can remove functionality that is within scope as a trade off. |
| Bad communication | high | medium | Send a follow-up email 24 hours after the initial email was sent. | Take the assumption route until they respond. |
| Working with new technologies | high | medium | Use Mob programming to figure out and tackle the new technology. | Try to find someone whom may be able to help with the said technology. If we cannot find anyone, we may send an email to the clients. |

## 

## 3.3 Overview of Risk Mitigation, Monitoring, and Management

The team will review risks at meetings and over discord. We shall decide if any actions need to be implemented in order to minimize and or handle the risk. Risk and the mitigation plan will be updated as needed. If a risk needs urgent attention, the team will communicate through discord to determine the best plan of action at that time. If the mitigation plan cannot be agreed upon by all members, the team lead will make the executive decision.

# 4.0 Project Schedule

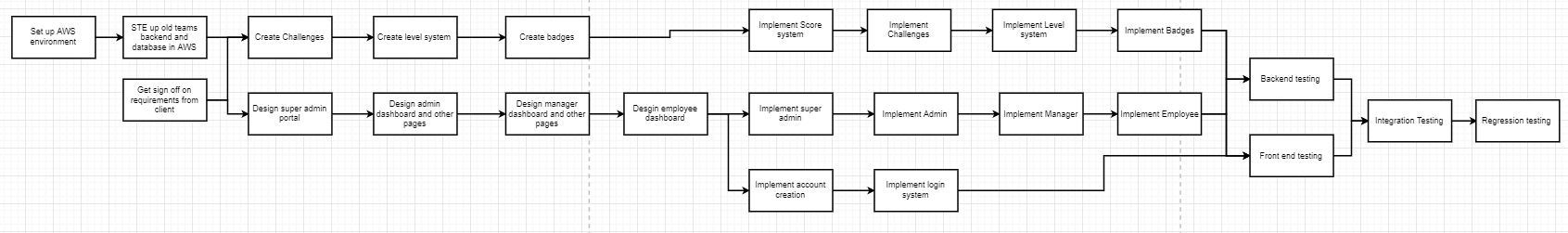
## 4.1 Project Task Set

* Set up AWS environment
* Set up old teams backend and database in AWS
* Get sign off on requirements from client
* Create challenges
* Create badges
* Create level system
* Design Super Admin portal
* Design Admin dashboard and other pages
* Design Manager dashboard and other pages
* Design Employee dashboard
* Implement account creation
* Implement login system
* Implement Super Admin
* Implement Admin
* Implement Manager
* Implement Employee
* Implement Score system
* Implement Challenges
* Implement Badges
* Implement Level system
* Backend Testing
* Frontend Testing
* Integration Testing
* Regression Testing
* Client acceptance

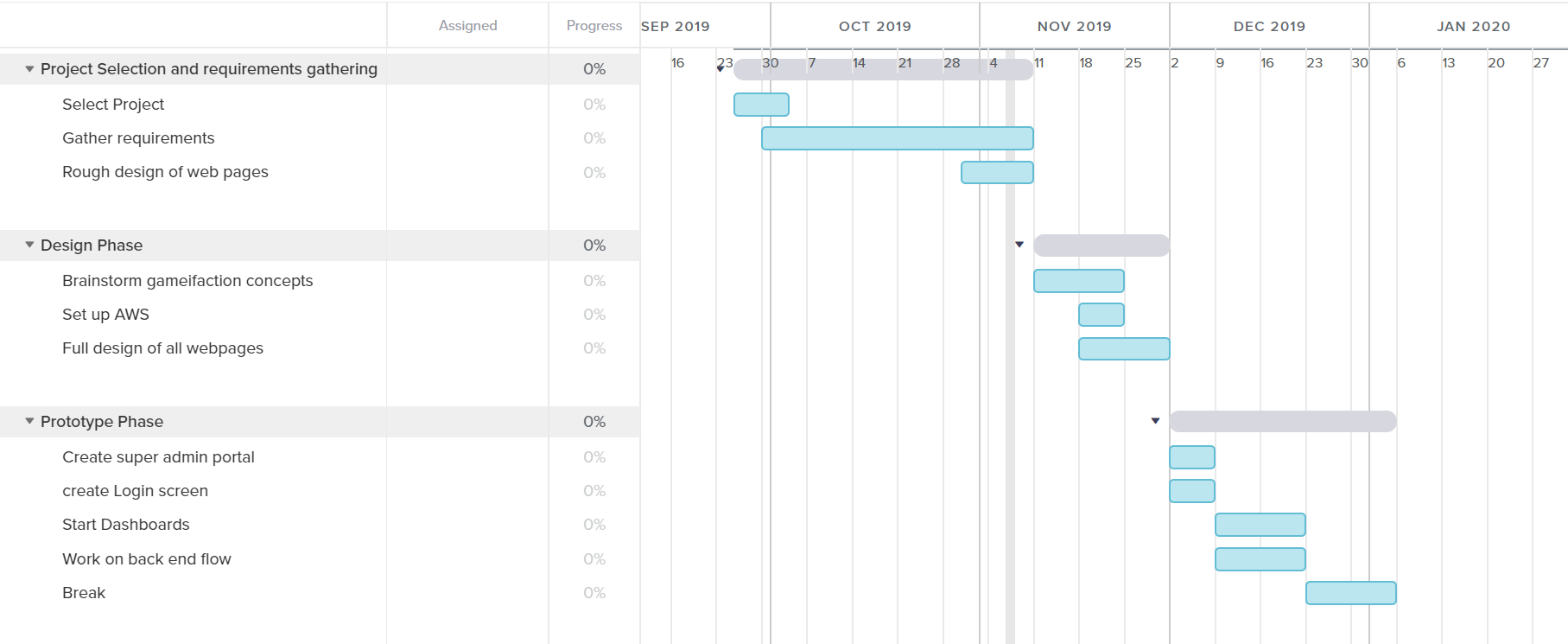
## 4.2 Functional Decomposition

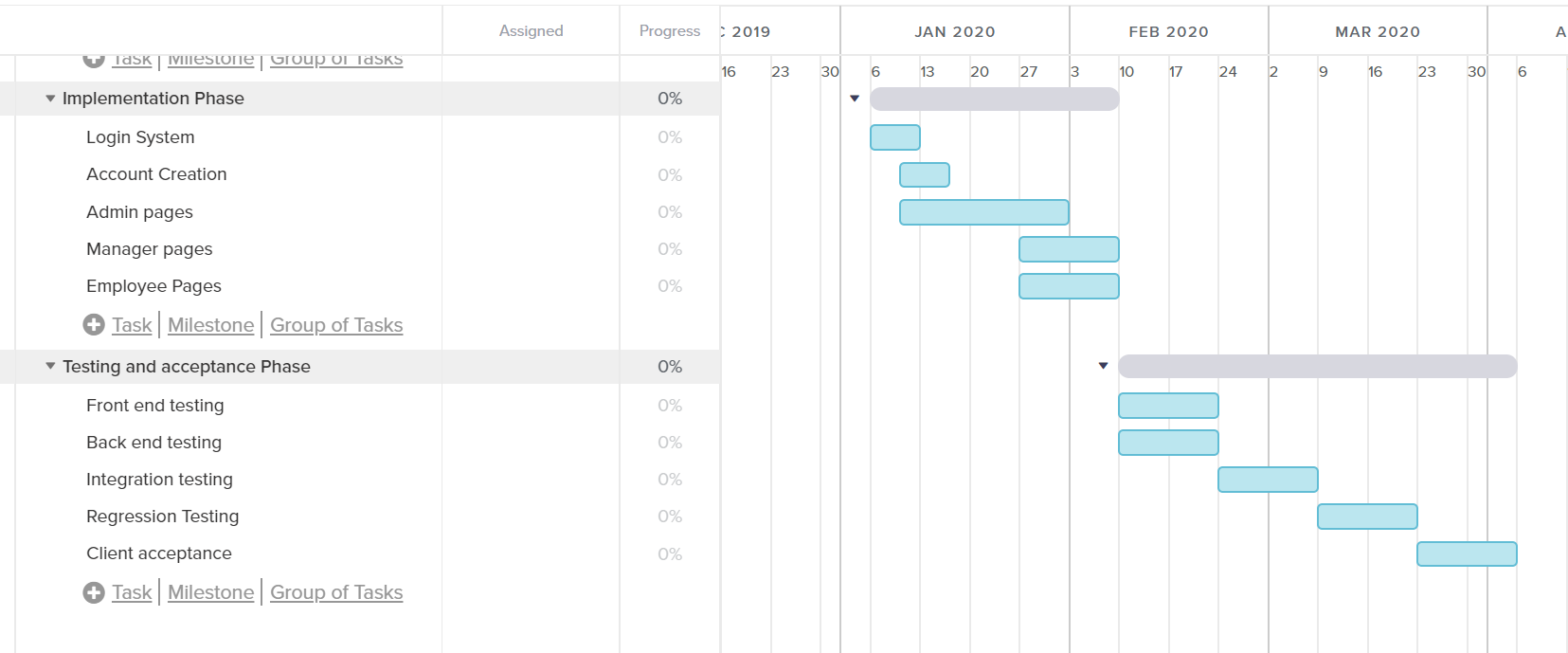
* Login
  + Input
    - Username and Password
  + Process
    - Confirm correct login
  + Output
    - Redirect to dashboard
* Display Dashboard
  + Input
    - none
  + Process
    - Get data from database
  + Output
    - Display charts and graphs
    - Display news, level, badges, and challenges
    - Display leaderboard

## 4.3 Task Network



## 4.3 Project Schedule





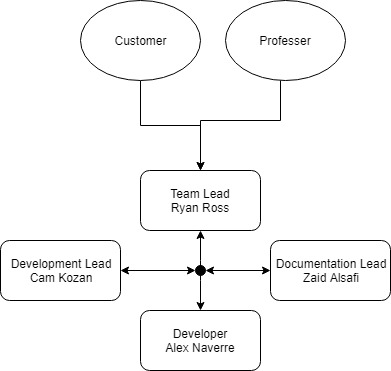
# 5.0 Staff Organization

## 5.1 Team Organizational Structure

The team is structured to involve each member during initial drafting periods, and then responsibilities are assigned as a concrete plan gets set. Members are assigned very similar jobs during the early weeks of the process to collect opinions and ideas, with the intention of obtaining input from all team members.

The roles of each team member are as follows:

* Ryan Ross- Team Lead
* Cameron Kozan- Design Lead
* Alex Navarre- Developer
* Zaid Alsafi- Documentation Lead



## 5.2 Management Reporting and Communication

The primary method of communication with the customer is via email with biweekly meetings to insure good communication. The discord app, and Google Drive is how the team will communicate and share work done. Discussions are hosted and open 24 hours a day. Every member can view all changes and proposals that are made. These communication formats are also used for questions, assignment and data sharing. Progress is tracked by all members to ensure all work gets done in a timely manner.

## 5.3 Team Member Responsibilities

Ryan Ross: Team Lead, making sure everything gets done, oversees the project, main communicator with clients. Also, creating, writing and revising documents, along with front end and back end programming.

Cameron Kozan: Design lead, create mock-ups and designs for client to approve or decline, along with general programming. Also, creating, writing and revising documents.

Zaid Alsafi: Documentation lead, In charge of overseeing documents. Final editor of documents along with creating, writing and revising documents. Also, front end and back end programming.

Alex Navarre: General programming development, database implementation and design. Being the main back end programmer. Also, creating, writing and revising documents.

# 6.0 Tracking and Control Mechanisms

## 6.1 Quality Assurance and Control

## 6.1.1 Objective

Our quality assurance and control plan was created to ensure that our webpage would function correctly with the already existing backend. The plan is a set of tests and checkpoints that allows our group to complete a cohesive piece of software in April. The checkpoints include inter team goals that help track our project and assure we are meeting our deadlines.

## 6.1.2 Test Strategy

Our test strategy is broken into four sections:

* Backend Testing
* Frontend Testing
* Integration Testing
* Regression Testing

The backend test cases will focus on assuring the already existing backend does what it is supposed to do. It should pull all item information from the Invisi-tag API and store the data in the correct location on the database server.

The frontend test cases address the aforementioned functionalities that our software will provide. The webpage should be able to access the database server that the backend created, and provide accurate company and employee data to the four individual user views.

Integration and regression test cases are used to ensure compatibility of the legacy and new software in development. Sanity tests will be performed to make sure the software logic is sound.

## 6.1.3 Checkpoint

There is a set of weekly checkpoints that our team has set to ensure a quality product will be provided in April. Group meets are held twice a week. The first meeting is to assign duties and goals for individuals for the week and the second meeting serves as a checkpoint for those goals.

The second set of checkpoints are designated by the client. Meetings are tentatively held every other week with all of the group members and Sargong Partners representatives. Goals and milestones are agreed upon and progress in the form of documentation and design mockups are assessed and signed off on.

## 6.2 Change Management and Control

Our change management and control is handled through debate. We have created a separate design and requirements document that both parties will agree upon and sign to establish a starting point. Anything outside of the scope of the document will be met with a discussion to assess the practicality of adding additional features to our software.

Our adjustment plan for when changes happen consists of a redefinition of the design and requirements document. After the new set of requirements is established, our team will schedule an immediate meeting to assess the severity of the changes made and how they will affect our timeline. After adjustments are made to the schedule, team members will apply the changes and properly document them.

# 7.0 Appendix

## 7.1 Appendix A - Staffing plan

