May 30, 2019

CS 364

Section 1

Brigham Young University Idaho

Wage Study

A BYUI AND RDBC project

Table of Contents

[1 Project Approved By: 1](#_Toc10156362)

[2 Introduction 3](#_Toc10156363)

[2.2. Purpose 3](#_Toc10156364)

[2.2. Scope 3](#_Toc10156365)

[2.3. Definitions 3](#_Toc10156366)

[3 Product Overview 5](#_Toc10156367)

[3.1. Product Perspective 5](#_Toc10156368)

[3.2. Product Functions 5](#_Toc10156369)

[3.3. User Characteristics 5](#_Toc10156370)

[3.4. Limitations 6](#_Toc10156371)

[4 References 7](#_Toc10156372)

[5 REVISION HISTORY  8](#_Toc10156373)

[6 Specific Requirements 9](#_Toc10156374)

[6.1. Database Design 9](#_Toc10156375)

[6.2. User Interface 10](#_Toc10156376)

[6.4. Server-Side 13](#_Toc10156377)

[6.4. Server Communication 13](#_Toc10156378)

[6.5. API Integration 14](#_Toc10156379)

[2.2. User Inputs 14](#_Toc10156380)

[6.7. Reports 15](#_Toc10156381)

[6.8. Security 20](#_Toc10156382)

[7 Verification 22](#_Toc10156383)

[7.1. User Profiles 22](#_Toc10156384)

[7.2. Data Types 23](#_Toc10156385)

[7.3. Ensures the aforementioned Items Exhibits Proper Functionality 23](#_Toc10156386)

[7.4. User Input 25](#_Toc10156387)

[7.5. Reports 27](#_Toc10156388)

[8 NON-ESSENTIAL REQUIREMENTS 32](#_Toc10156389)

[8.1. This section is dedicated to requirements beyond the scope of the minimum viable product. 32](#_Toc10156390)

[9 Appendices 32](#_Toc10156391)

[9.1. Credits 32](#_Toc10156392)

[9.2. Assumptions and Dependencies 32](#_Toc10156393)

Project Approved By:

|  |  |  |  |
| --- | --- | --- | --- |
| Title: | Name | Signature | Date |
| Owner | Will Jensen |  |  |
| Project Manager | Cody Lundberg |  |  |
| Team Lead | Shawn Ashton |  |  |
| Team Lead | Brian Bates |  |  |
| Team Lead | Mario da Silva Filho |  |  |
| Team Lead | Spencer Eccles |  |  |
| Team Lead | Christian Martinez |  |  |
| Team Lead | Robert Neilson |  |  |
| Team Lead | Braden Rice |  |  |
| Sponsor | William Clements |  |  |

* + - 1. Introduction

The Wage Study WebApp Project was created to provide government entities with a wage study tool. This tool’s requirements specify the purpose and scope of the system, and provide an outline for the required user, hardware, software, and communication interfaces. Additionally, the specifications for the necessary functionality, operations, and the limitations of the system are provided here.

* 1. Purpose

The purpose of the Wage Study WebApp Project is to provide an outline to build a web application. That application will be linked to a database that gives wage estimations for government agencies.

* 1. Scope

This system shall provide wage estimations for government agencies. It shall be used by a government agency to help track government entity and market wage ranges for each job under that agency.

* 1. Definitions
  2. ASCII: ASCII, abbreviated from American Standard Code for Information Interchange, is a character encoding standard for electronic communication. ASCII codes represent text in computers, telecommunications equipment, and other devices.
     1. AWS: Amazon Web Services (AWS) is a comprehensive, evolving cloud computing platform provided by Amazon. It provides a mix of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings.
     2. CFO: Chief Financial Officer
     3. Cost of Living Adjustment (COLA): The Cost of Living Adjustment (COLA) adjusts the reported incomes of different cities to take into consideration the cost of living in a specific area. This way, cities with different costs of living can compare salaries more accurately.
     4. EJB: Enterprise Java Beans (EJB) is a development architecture for building highly scalable and robust enterprise level applications to be deployed on J2EE compliant Application Server such as JBOSS, Web Logic etc.
     5. End User: a person who will use the system to perform a task.
     6. Entity: short form of government entity.
     7. Government Entity: a city, county, municipality, or state department.
     8. Hashing: hashing performs a one-way transformation on a password, turning the password into another String, called the hashed password.
     9. HR: Human Resources
     10. Pay Grade: A grade or a level on a pay scale which specifies the pay range for each grade.
     11. UI: User Interface, which, in the industrial design field of human–computer interaction, is the space where interactions between humans and machines occur.

Product Overview

* 1. Product Perspective
* This system could be categorized as a human resources product, since it can be used to provide government entities a with nationwide comparison for wages for different positions required by that entity.
* The system comprises data from different agencies throughout the country who can upload their current database information on salaries and positions.
* The system allows administrators to estimate costs for personnel wages.
* The system allows supervisors to view the positions they oversee, pay grade for those positions, and market ranges, compared with those of other government entities.
* As a web application, the system stores information in a database and all inputs are saved to the database.
* The system is accessible through an internet browser and users can input their city’s job information.
  1. Product Functions
* The system’s main function is to present reports on a government entity’s wages.
* The system will provide an interface to enter wage data.
* The system will provide an interface to evaluate the comparability of positions other government entities have provided.
* The system will show the average wage range of a user-selected list of comparable positions within a specified period.
* The system will be capable of fitting wages to a grade scale for the entity’s positions.
* The system will provide a suggested budget projection to bring a government entities wages in line with market rates.
  1. User Characteristics

The intended end users for the application are primarily government entity human resource managers. Secondary users may include an entity’s chief financial officer, Mayor, City Administrator, an individual department head, a city council member or other user given access by an administrator. These users will be assigned roles within the system. These roles include: Owner, Administrator, User, View-Only User, and Custom-defined roles.

* An Owner role will typically be assigned to a CFO, Mayor or City administrator. An owner’s primary interactions with the system will be assigning an administrator and arranging for payment.
* An Administrator role will be held by an HR director and/or HR manager. An administrator’s primary interactions with the system will be entering wage data, evaluating position comparability, and viewing reports across all the departments they oversee.
* A User’s role will typically be held by a government entity’s department head. A user’s primary interactions with the system will be editing wage data for their department and viewing reports associated with their department.
* A View-Only User’s role will typically be held by a city council member. A View Only User’s primary interaction with the system will be viewing reports.
* A Custom-Defined user’s role will be assigned based on decisions made by the Owner and Administrator roles. These decisions will include what permissions will be given to a custom-defined user, and those permissions will determine a custom-defined user’s interaction with the system.
  1. Limitations

1. The application shall not be an employee payroll software.
2. The application shall not be used to track the number of employees.
3. The application shall not store personally identifying information, except for employee names.

References

[1] A-team Elicitation data

[2] Ctrl Alt Elite Elicitation Data

[3] BronzeBeard Elicitation Data

[4] Triforce Heroes Elicitation Data

[5] Low Lifes elicitation Data

[6] Chris, Chris, & Co. Elicitation Data

[7] SegFault elicitation Data

REVISION HISTORY

|  |  |  |
| --- | --- | --- |
| Version | Changes | Date |
|  |  |  |
|  |  |  |
|  |  |  |
| 3 | Removed unused sections, revised sections, |  |
| 2 | Added introduction, product functions and overview, created table of contents |  |
| 1 | revised Specific requirements. |  |
| 0 | Added Specific Requirements section, Some verifications written |  |

Specific Requirements

* 1. Database Design
     1. User Login Data
        1. The database shall use an email address for the user's username. [2-1]
        2. The password shall be securely stored. [2-1]
        3. The system shall allow each user to have a pseudorandom salt. [2-1]
     2. User Types
        1. The system shall provide user roles such as Owner (CFO/Mayor/City Administrator), Administrator (HR Director), User (Department Head), View-Only User (City Council), Custom. [2-2]
        2. The system shall allow the “Owner” role all permissions relating to their city [2-2]
        3. The system shall allow the “Administrator” role to edit wage data, evaluate position comparability, and view reports within their respective city. [2-2]
        4. The system shall allow the “User” role to edit wage data and view reports associated with their department. [2-2]
        5. The system shall allow the “View-Only User” role to view reports. [2-2]
        6. The system shall have the permission granted to the Owner to make a “Custom” user type with specific selected permissions for the Custom user type. [2-2]
     3. Subscription Status
        1. The database shall store the date that a user’s subscription ends. [2-1]
        2. The database shall store information to determine if a subscription is: current, ending within a month, or suspended. [2-1]
        3. The database shall store the tier of each subscription. [2-1]
        4. The database shall store the annual cost of each subscription type. [2-1]
     4. Individual Employee Status Table
        1. The database shall store each employee’s name. [2-1]
        2. The database shall store each employee’s occupation start date. [2-1]
        3. The database shall store each employee’s current payment level. [2-1]
        4. The database shall store each employee’s salary history [2-1]
     5. Data Types
        1. The database shall store the area in square miles of each city. [2-1]
        2. The database shall store the population of each city. [2-1]
        3. The database shall reference information from BLS.gov/developers. [2-1]
        4. The database shall reference information from BLS.gov/oes. [2-1]
        5. The database may reference information from the MIT Living Wage Calculator on a city by city basis. [2-1]
        6. The database shall store a history of reports generated by each user. [2-1]
        7. A remote server shall host the database [2-1]
        8. The database shall store a city pay scale. [3-3]
        9. The database shall store the number of pay grades. [3-3]
        10. The database shall store the current minimum pay for each grade. [3-3]
        11. The database shall store the proposed minimum pay for each grade. [3-3]
        12. The database shall store the proposed market pay for each grade. [3-3]
        13. The database shall store the current maximum pay for each grade. [3-3]
        14. The database shall store the proposed maximum pay for each grade. [3-3]
        15. The database shall store the city pay scale. [3-3]
        16. The database shall store the current minimum pay for each grade. [3-3]
        17. The database shall store job positions. [3-3]
        18. The database shall store the position name. [3-3]
        19. The database shall store a description of the position. [3-3]
        20. The database shall store the department the position is a part of. [3-3]
        21. The database shall store the position pay grade. [3-3]
        22. The database shall store the new position to the database. [3-3]
     6. Data Calculations
        1. Avg Max Wage
           1. The database shall calculate the avg max wage.
           2. The database shall provide the avg max wage to the user interface.
        2. Avg Min Wage
           1. The database shall calculate the avg min wage.
           2. The database shall provide the avg min wage to the user interface.
        3. Max Wage Adjustment
           1. The database shall calculate the max wage adjustment.
           2. The database shall provide the max wage adjustment to the user interface.
        4. Min Wage Adjustment
           1. The database shall calculate the min wage adjustment.
           2. The database shall provide the min wage adjustment to the user interface.
        5. Market Wage
           1. The database shall calculate the market wage.
           2. The database shall provide the market wage to the user interface.
  2. User Interface
     1. User Interface
        1. The software shall provide compatibility for Internet Explorer [7-2.0]
        2. The software shall provide compatibility for Mozilla Firefox [7-2.0]
        3. The software shall provide compatibility for Google Chrome [7-2.0]
        4. The software shall provide compatibility for Safari [7-2.0]
     2. User Navigation
        1. The system shall provide a navigation menu bar [7-2.0]
        2. The menu bar shall include a link to the homepage [7-2.0]
        3. The menu bar shall include a link to the user’s account information [7-2.0]
        4. The menu bar shall include a link to the grid. [7-2.0]
     3. The Home Page
        1. The software shall navigate the user to the home page [7-2.0]
        2. The software shall provide a way to sign up for a new account [7-2.0]
        3. The software shall provide a way to sign in into an existing account [7-2.0]
        4. The software shall provide a way to view contact information [7-2.0]
     4. Signing Up
        1. Users will purchase their accounts by external means [7-3.3]
        2. After a successful purchase of this software, users shall be emailed a link which will lead to a webpage requesting for them to enter their username (email address). [7-3.3]
        3. After a successful purchase of this software, users shall be emailed a link which will lead to a webpage requesting for them to enter a password. [7-3.3]
        4. Upon successful login, users shall be able to access all the features of this software they purchased/are allowed permissions for. [7-1.1]
        5. The software shall require the user to enter an email address [7-3.3]
        6. The software shall send a validation email to the user upon creation of an account. [7-3.3]
     5. Signing In
        1. The software shall prompt the user to enter their username (email address). [7-3.3]
        2. The software shall prompt the user to enter their password. [7-3.3]
        3. The software shall allow the user to access the rest of the software upon successful login. [7-3.2]
        4. The software shall lock the user out of the sign in page for a brief amount of time to prevent brute force hacking if too many attempts are inputted. [7-1.1]
        5. The lock-out times for too many password attempts shall be standard to all users. [7-1.1]
        6. The software shall provide a password recovery page [7-1.1]
     6. Username
        1. The software shall validate username [7-1.1]
           1. The software shall ensure username uniqueness [7-1.1]
           2. The software shall ensure that usernames are long enough [7-1.1]
           3. The software shall ensure that usernames are valid emails [7-1.1]
           4. The software shall give an error message if username is invalid [7-1.1]
           5. The software shall prevent the user from proceeding until username is valid. [7-3.3]
     7. Password
        1. The software shall never display a customer’s password. [7-1.0]
        2. The password shall always be echoed with special characters representing typed characters. [7-1.0]
        3. Passwords shall run form validation to make sure passwords only contain standard ASCII values. [7-1.0]
        4. Passwords shall run form validation to make sure passwords are long enough. [7-1.0]
        5. Passwords shall run form validation to make sure passwords are complex enough to be secure. [7-1.0]
        6. The software shall not leave any cookies on the customer’s computer containing the user’s password. [7-1.0]
     8. Receiving Server Communication
        1. When displaying charts, graphs, or reports (detailed 5.6 and 5.8), the user interface shall request the following information about each city being compared: city name, city population, number of homes in city limits, number of miles in the city, and the average cost of living in the city.
        2. The client must receive information about jobs (a title, dates of significant changes in wage or worker).
           1. The dates of wage changes along with their wage value shall be received.
           2. The dates of workers holding a job along with their name shall be received by the client.
           3. The client shall request information about jobs from a default list of jobs. For example, each city requested will have the following “default” positions: City Administrator, City Attorney, Fire Chief, Chief of Police, etc…
           4. The client shall receive jobs created by a city account like it receives all default jobs listed above
           5. The server shall provide the Avg Max Wage.
           6. The server shall provide the Avg Min Wage.
           7. The server shall provide the Max Wage Adjustment.
           8. The server shall provide the Min Wage Adjustment.
           9. The server shall provide the Market Wage.
     9. Sending Server Communication
        1. The customer’s password may be reset but never shown.
        2. The system’s back-end servers shall only be accessible to authenticated administrators.
        3. Other sent information can be separated into three groups, namely requests, additions, and updates.
           1. Requests for information shall include data graphs based on:

City – name, size, population, state, country

Job – title, number

* + - * 1. Additions and updates to the information stored in the database shall include:

New City – name, size, population, state, county

New Job – title, description, wage, category

Wage – grade, min, max

Category – Admin/IT, Planning/Building, Public Works, Parks & Recreation, Public Safety

Update City – name, size, population, construction

Construction – number of houses, miles of roads

Update Job – title, description, wage category

* 1. Server-Side
     1. Accessibility
        1. Multiple users shall be able to access the server 24/7 [5-20]
     2. Users
        1. *Types of Users*
           1. There shall be Administrators

Administrators shall have the ability to create super-users.

Administrators shall have the ability to deactivate users.

* + - * 1. There shall be Super-users

Super-user shall be able to send email that creates sub-user

Super-user shall be able deactivates respective sub-users

* + - * 1. There shall be Sub-users.

Sub-Users shall have same access as respective super-user except they cannot create other users.

* + 1. Authentication
       1. *Sign-up (create new user)*
          1. Shall be able to send email invitation for account creation (see 6.3)
          2. Users information shall be receivable from the email account creation.
          3. The server shall relay user information to the database.
          4. The server shall pass back a confirmation to the UI.
    2. Log-in
       1. The server shall receive the username and hashed password from the UI.
       2. The server shall query the database for the username and the corresponding password.
       3. The server shall check to make sure that the hashed password from the UI and the hashed passwords from the database match.
       4. The server shall relay to the UI that the hashed passwords either do or do not match.
  1. Server Communication
     1. Client-Database Communication
        1. The server shall get data from the clients.
        2. The server shall send data to the database.
        3. The server shall query the database
        4. The server shall send information to the clients.
  2. API Integration
     1. The server shall send requests to API.
     2. The server shall receive raw data from API.
     3. The server shall mine the data.
     4. The server shall send data to the clients.
     5. The server shall send data to database.
  3. User Inputs
     1. Editing the Pay Scale
        1. The system shall allow the user to create a city pay scale. [7-3.3]
           1. The system shall allow the user to enter the number of pay grade levels. [7-3.3]
           2. The system shall allow the user to enter the current minimum pay for each grade. [7-3.3]
           3. The system shall allow the user to enter the proposed minimum pay for each grade. [7-3.3]
           4. The system shall allow the user to enter the proposed market pay for each grade. [7-3.3]
           5. The system shall allow the user to enter the current maximum pay for each grade [7-3.3]
           6. The system shall allow the user to enter the proposed maximum pay for each grade. [7-3.3]
        2. The system shall allow the user to enter the pay scale into the database after a city pay scale has been created. [7-3.3]
        3. The system shall allow an admin user to update the city pay scale. [7-3.3]
           1. The system shall allow the user to update the number of pay grade levels. [7-3.3]
           2. The system shall allow the user to update the current minimum pay for each grade. [7-3.3]
           3. The system shall allow the user to update the proposed minimum pay for each grade. [7-3.3]
           4. The system shall allow the user to update the proposed market pay for each grade. [7-3.3]
           5. The system shall allow the user to update the current maximum pay for each grade. [7-3.3]
           6. The system shall allow the user to update the proposed maximum pay for each grade. [7-3.3]
        4. The system shall enter the updated pay scale into the database. [7-3.3]
     2. Editing Position Information
        1. The system shall allow the user to create a new job position. [7-3.3]
           1. The system shall allow the user to enter the position name. [7-3.3]
           2. The system shall allow the user to enter a description of the position. [7-3.3]
           3. The system shall allow the user to enter the department the position is a part of. [7-3.3]
           4. The system shall allow the user to enter the position pay grade.
        2. The system shall store the new position in the database. [7-3.3]
        3. The system shall allow the user to update a job position. [7-3.3]
        4. The system shall allow the user to update the position name. [7-3.3]
        5. The system shall allow the user to update a description of the position. [7-3.3]
        6. The system shall allow the user to update the department the position is a part of. [7-3.3]
        7. The system shall allow the user to update the position pay grade. [7-3.3]
        8. The system shall store the updated position to the database. [7-3.3]
     3. Editing Employee information
        1. The system shall allow the user to create new employees. [7-3.3]
           1. The system shall allow the user to enter the employee’s name. [7-3.3]
           2. The system shall allow the user to enter the employee’s current position. [7-3.3]
           3. The system shall allow the user to enter the employee’s date of hire. [7-3.3]
           4. The system shall allow the user to enter how long the employee has been in the current position. [7-3.3]
           5. The system shall allow the user to enter the employee’s current pay. [7-3.3]
        2. The system shall store the created employee in the database. [7-3.3]
        3. The system shall allow the user to update the employee’s information. [7-3.3]
           1. The system shall allow the user to change the employee’s name. [7-3.3]
           2. The system shall allow the user to change the employee’s pay grade. [7-3.3]
           3. The system shall allow the user to update the employee’s current position. [7-3.3]
           4. The system shall allow the user to change how long the employee has been in the current position. [7-3.3]
           5. The system shall allow the user to change the employee’s current pay. [7-3.3]
        4. The system shall store the information in the database. [7-3.3]
  4. Reports
     1. A default grading system shall be suggested to the user.
        1. The grading system shall have set dimensions prescribed by the user. [3.2.6]
           1. The number of rows shall be set by the user

These rows shall represent the pay grade.

* + - 1. The number of columns shall be set by the user.

These columns shall represent the increase of pay grade over time within each row.

The start of each row shall represent starting salaries.

* + - * 1. The rows shall be labeled with numbers from top to bottom.
        2. The columns shall be labeled with letters from left to right.
      1. All salaries shall be in terms of hourly pay in USD. [3.2.6]
      2. The initial minimum salary shall be entered in by the user. [3.2.6]
         1. The minimum salary shall be placed in the top left cell.
      3. The initial maximum salary shall be entered in by the user. [3.2.6]
         1. The maximum salary shall be used as a relative threshold for calculation purposes.
      4. The default grading system shall have automatic calculations to initialize a chart. [3.2.6]
         1. These default calculations shall be in the SDD.
      5. All salaries under column 1 shall represent salaries based on the amount of work experience
    1. The grading system shall be customizable
       1. The minimum salary on the default chart shall be customizable. [3.2.6]
       2. The maximum salary on the default chart shall be customizable. [3.2.6]
       3. The dimensions of the default chart shall be customizable. [3.2.6]
       4. The algorithms generating the presumed default salaries shall be customizable. [3.2.6]
          1. All algorithms mentioned in 1.1 shall be customizable with the ability to add, remove, or replace algorithms if desired.
       5. The salary display of the chart shall be customizable. [3.2.6]
          1. The additional options shall include daily, weekly, monthly, semi-annual, annual, and single paycheck pay to replace the hourly pay default if desired.
       6. The order of the default design going down from smallest to largest starting salaries shall be reversable. [3.2.6]
    2. There shall be a page that shows what current employees are being paid. [3.3]
       1. This page shall use the data entered by the users to show each employee by job and their pay. [3.3]
       2. This page shall display the pay grade
       3. This page shall display the pay range of each grade
       4. This page shall display each job title according to department
       5. This page shall display all departments together
       6. The information displayed in this page shall be stored in the database
          1. The pay grade shall be stored in the database
          2. Each pay range associated with each pay grade shall be stored in the database
          3. Each job category shall be stored in the database

The jobs associated with each category shall be stored in the database

The database shall keep record of the number of people holding each job within each category

* + - 1. This information shall be modifiable by a user with appropriate access. [3.3]
      2. The page shall be downloadable as a spreadsheet. [3.2]
    1. The system shall determine if a job should be created based on similar cities
    2. The system shall determine if a job should be removed based on similar cities
       1. Similar cities shall be determined by factors listed in the Cost of Living Adjustment (COLA) such as city location, medium income, land mass size, population size, number of houses and number of seasons per year.
       2. There shall be information regarding how many cities of similar size have a given job.
       3. There shall be a list of jobs that currently exist in the user’s city that do not exist in other cities.
       4. There shall be a list of jobs that currently do not exist in the user’s city that do exist in other cities.
          1. The method by which jobs are sorted shall be customizable
       5. A city’s metrics and size shall be adjustable.
       6. There shall be a list of other cities and how they compare in size
       7. In places where types of jobs are specified, prices for employing or savings for removing shall be present.
    3. Compare current data with market
       1. City job comparison report
          1. The city job comparison report shall show how a specific job compares financially to similar jobs in other cities

Cities shall be listed with the title that they use for that job

A minimum hourly pay for the jobs shall be listed with each city

A maximum hourly pay for the jobs shall be listed with each city

* + - 1. The city job comparison report shall be downloadable as a spreadsheet.
    1. City compared to market averages
       1. The city compared to market report shall show how each job title compares to the market’s average values
          1. The system shall highlight extreme differences between the current country compared and the market average.
          2. Each job title in the city shall be listed

Minimum hourly pay for the city shall be listed for each job

Maximum hourly pay for the city shall be listed for each job

* + - * 1. Each job title from the market shall be listed next to the corresponding city’s listed job

Minimum hourly pay for the market shall be listed for each job

Maximum hourly pay for the market shall be listed for each job

* + - 1. The city compared to market report shall be downloadable as a spreadsheet.
    1. The system shall display budget impact reports with adjustable wages. [2.2.1]
       1. Forecasted budget impact report [2.2.1]
          1. The budget impact report shall rely on data calculated from the Cost Estimator. [3.7.1]
          2. As with all reports, the forecasted budget impact report shall be downloadable as a spreadsheet. [2.3, d, i]
          3. The forecasted budget impact report shall make a comparison between the market value employee wage budget and the government entities actual employee wage budget. [2.4]

The report shall fetch the total cost of the government entity's employees. [3.7.1.1]

The report shall fetch the total cost of the government entity’s employees at market value. [3.7.1.1]

The report shall fetch the budget impact percentage figure from 3.7.1.2 [3.7.1.1]

The budget impact for the entity shall be viewable.

* + - * 1. An employee’s wage shall be adjustable to view the effects of different wage changes. [2.2.5]

Specific employee wages shall be adjustable.

The adjust wage shall be reflected in the budget impact report.

* + - * 1. The forecasted budget impact report shall display adjusted impacts from employees. [2.2.5]

The forecasted budged impact report shall fetch the employee information listed in 3.7.1.2.

The information listed shall be viewable.

* + 1. The system shall have access to current information regarding employees. [3.2.7]
       1. The system shall keep track of job titles.  [3.2.7]
          1. Job titles shall have a prioritizing system that keeps track of how common a job title is. [3.2.7]
       2. The system shall keep a record of similar job titles. [3.2.7]
          1. The system shall suggest possible job title variations when inputting new job titles. [3.2.7]
       3. There shall be a pay scale that is visible for a job title in similar cities. [3.2.7]
          1. There shall be a pay scale key assigned to a job title that references the range for the job title. [3.2.7]
       4. The system shall keep a record of the amount of time spent in a job to change placement in pay scale. [3.2.7]
          1. The system shall suggest payment ranges when adding new employees. [3.3.4]
       5. When searching for employee information shall be displayed in the search results. [4-3.8]
       6. When searching for city data information shall be displayed in the search [4-3.8]
       7. Each column of data shall be labeled according to the information contained in the column [4-3.8]
       8. The User shall be able to see what the data they are looking at pertains to. [4-3.8]
          1. The labels for each column and row shall remain visible as the user scrolls through the data. [4-3.8]
          2. The user shall be able to sort the displayed data by clicking on the labels of each column. [4-3.8]
    2. The system shall have access to current information regarding employees.
       1. The system shall keep track of job titles
       2. The system shall keep a record of similar job titles.
       3. There shall be a pay scale that is visible for a job title in similar cities.
    3. The system shall provide search filters [4-4.9]
       1. Filters shall provide the ability to narrow the results [4-4.9]
          1. The search shall provide the ability to filter by keywords [4-4.9]
          2. The search shall provide the ability to filter by job title [4-4.9]
          3. The search shall provide the ability to filter by department [4-4.9]
       2. The system shall provide sort filters for returned items [4-4.9]
          1. Filters shall provide the ability to sort the results displayed [4-4.9]
          2. The sort order shall be invertible [4-4.9]
    4. Overview of Report Exporting
       1. The system shall be able to export reports [4-3.12]
          1. The system shall be able to export report CSV [4-3.12]
          2. The system shall be able to export report PDF Files [4-3.12]

The system shall be able to export report web-based page [4-3.12]

The system shall generate a sharable link [4-3.12.3]

The system shall grant access to the report through the link [4-3.12.3]

The link shall expire 30 days after its creation [4-3.12.3]

Expired links to reports shall not provide access to the report [4-3.12.3]

* + - 1. The system shall provide a way to select columns that should be included in the exported report [4-3.12]
      2. The system shall export report title by default [4-3.12.3]
      3. The system shall export headers of the selected columns by default
  1. Security
     1. User Identification
        1. Implementation on the front-end system of the application.
           1. User identification will be implemented on the client side connecting through an API.
        2. This would only be applied during registration.
        3. Account registration will verify that user is a person behind a computer and not a not making the system secure off users that want to corrupt or steal information from system.
        4. Familiarity is consistent as it is shared throughout many applications when registering.
     2. User Permissions
        1. Program shall contain multiple types of accounts in the programs.
           1. Owner Account
           2. Administrator Account
           3. User Account
           4. View-Only User Account
           5. Custom-Defined Account
        2. Different accounts shall grant various levels of access to the program
           1. Owner Account access

Create all lower level accounts

Modify permissions for all low-level accounts

Arrange for payments

Ability to maintain the system

Has access to all data

* + - * 1. Administrator Account access

Used by HR director or HR manager

Create lower level accounts

Modify permissions for lower level accounts

Add wage data

Evaluate position comparability

View reports across all departments it oversees

Has access to data in all departments it oversees

* + - * 1. User Account access

Used by a government entity’s department head

Edit wage data for the department

View reports associated with the department

* + - * 1. View-Only User Account access

Can view reports

* + - * 1. Custom-Defined Account access

Higher level accounts will define the permissions for this account

* + 1. Secure Database
       1. Users will be able to both get and post data to the database
       2. Data being referenced will be accurate each time it is accessed.
          1. Provide integrity of the data for the user
       3. Queries shall be processed in a trusted execution environment
          1. Can protect sensitive data and code, even from powerful attackers that control or have compromised the operating system and the hypervisor on a host machine
       4. Implementing Amazon SimpleDB (or AWS for short) or Oracle Database
          1. Ability to manage the database that are easy to use and can do a lot of the work for the user.
          2. System should be agile and scalable
          3. Allow backups and partitions to better store and recover data depending on the scenario.
    2. Encryption
       1. A security measure that will benefit the user’s privacy is a password set on the document that is to be downloaded.
          1. This ensures that only the people with password can view the contents within
          2. Is done through the process of encryption - the process of converting information or data into a code, especially to prevent unauthorized access.
       2. *Protecting sensitive data*
          1. Username, passwords, or credit card information
          2. Information input is made private to be disclosed to the necessary people
          3. Certificate to certify an encryption key on all passwords (SSL certificates)
       3. Access for security purposes
          1. Administrators will be granted more access than the common user
       4. *SSL certificates*
          1. Certificates will aid against preventing cyber-attacks
          2. Program shall prevent attacks such as malware, viruses, trojans, worms, etc.
       5. *Security Protocols*
          1. Will ensure the safety of user information
          2. Includes techniques such as encryption, privacy settings or passwords
          3. Including Document password encryption - Users will have the ability to download the spreadsheets for their use.

Verification

* 1. User Profiles
     1. Question to be answered in future version – no testing
     2. User Login Data
        1. Verify that emails fit proper format, verify via email registration that emails are valid.
        2. Verify by signing up, logging in.
           1. Come up with a technique to make it unique. For example, timestamp + username + 26 random characters.
     3. User Permission Verification
        1. Need to verify that the relevant information is shown only to the relevant account type using the following steps.
           1. Owner Account

Verify that the account can access all data for all cities

Verify that for each city, all job types and wages can be viewed

Verify that information about individual employees is visible

Verify that information about every job description is visible

Verify that access to market data is available

Verify that calculations for future years is possible

Verify that there is access to the market data

Verify that the account can upload and modify data

Verify that this account can create other lesser accounts

Verify that this account can modify permissions for the other accounts

* + - * 1. Administrator Account

Verify that the account has access to data for only the relevant companies it oversees

Verify that job types wages are accessible only for those companies

Verify that information about individual employees is visible only for those companies

Verify that information about every job type description is visible only for those companies

Verify that access to market data is available only for those companies

Verify that calculations for future years is possible only for those companies

Verify that there is access to the market data only for those companies

Verify that the account can upload and modify data only for those companies

Verify that this account can create individual accounts

Verify that this account can modify permissions for lesser accounts only

* + - * 1. User Account

Verify that the account has access to data for only the relevant department

Verify that job types wages are accessible only for that department

Verify that information about individual employees is visible only for that department

Verify that information about every job type description is visible only for that department

Verify that access to market data is available only for that department.

Verify that calculations for future years is possible for that department

Verify that there is access to the market data for that department

Verify that the account can upload and modify data only for that department

Verify that this account cannot create any other accounts

Verify that this account cannot modify any other accounts

* + - * 1. View-Only Account

Verify that the account has access to data for only the relevant department

Verify that the account cannot create any new entries in the data

Verify that the account cannot modify entries in the data

Verify that the account can generate reports from the data

* + - * 1. Custom-Defined Account

Verify that the account has access to the relevant data the specific permissions allow

Verify that the account can perform the actions the specific permissions allow

* + 1. **Secure Database**
    2. User Types
       1. Verify by visual check.
       2. Verify by checking permissions stored in the database.
    3. Parent Accounts and Subsidiary Accounts
       1. Subscription Status
          1. Verify by creating a fake subscription, then check the details.
       2. Individual Employee status table
          1. Verify by using queries.
  1. Data Types
     1. Verify by using queries.
     2. Create a query for these other databases, using the reference from our database.
     3. Verify that the Living Wage Calculator functions properly and that the necessary pertinent information is stored in the database.
     4. Create multiple reports, ensure that they can be accessed later rather than re-generated.
     5. Verify that the data is, in fact, stored on the yet-to-be-determined cloud service.   Data Calculations
     6. Avg Max Wage
        1. Verify 3.1.6.1.1 by selecting two cities. The Avg Max field should be the avg of the max wages of the two cities.
        2. Verify 3.1.6.1.2 by selecting two cities. The Avg Max field should be filled in with the max avg.
     7. Avg Min Wage
        1. Verify 3.1.6.2.1 by selecting two cities. The Avg Min field should be the avg of the min wages of the two cities.
        2. Verify 3.1.6.2.2 by selecting two cities. The Avg Min field should be filled in with the min avg.
     8. Max Wage Adjustment
        1. Verify 3.1.6.3.1 by selecting two counties. The Max Wage Adjustment field should be the adjusted max wage.
        2. Verify 3.1.6.3.2 by selecting two counties. The Max Wage Adjustment field should populate with the adjusted max wage.
     9. Min Wage Adjustment
        1. Verify 3.1.6.4.1 by selecting two counties. The Min Wage Adjustment field should be the adjusted min wage.
        2. Verify 3.1.6.4.1 by selecting two counties. The Min Wage Adjustment field should populate with the adjusted min wage.
     10. Market Wage
         1. Verify 3.1.6.5.1 by navigating to the market wage page. The correct calculation should be populating the field.
         2. Verify 3.1.6.5.2 by navigating to the market wage page. The Market Wage field should be populated.
  2. Ensures the aforementioned Items Exhibits Proper Functionality
     1. User Authentication
        1. For the first test, we will make some random test user account. The invitation to that random user’s email will be sent by project manager. If the random user gets an email of invitation, then the invitation function is working. If the user can sign-up and get all the information that was provided by project manager, then we can confirm that the range of information one can access is working as it should. In other words, an individual must get data that are validated by his/her boss.
        2. For login test, we will sign out from the same ‘random account’ and try to login. If we can do all the activities and everything is working just like when we signed up first time, then we shall confirm that the sign-in is working.
     2. Interface with client/ UI:
        1. This test case will involve inputting information into the application. After the information is inputted into the system, we will check our database to confirm that the information was stored correctly.
        2. The second test case will involve checking the information displayed on the application and the information in the database to make sure that they match exactly.
     3. User Interface
        1. Verify that the software is compatible with Internet Explorer, Mozilla, Safari, and Chrome.
        2. Verify which tool or software package the project is implemented with (Java Applet, MS Front Page, EJB etc.)
     4. User Navigation
        1. Verify user access to all pages within the user’s privileges through menu bars, in-page motivation buttons, and hyperlinks.
        2. Verify there are sidebars that enable the user to select which graph or evaluator they want to use.
     5. The Home Page
        1. Verify that accessing the website initially displays the home page.
        2. Verify that the user has the option to sign up for a new account, sign in, or contact the company.
     6. Sign Up
        1. Verify the software allows the user to sign up for an account on the website.
           1. Verify the sign-up button leads the user to a form where they may enter their personal information and purchase the software as defined in the section below.
     7. Buying an Account
        1. Can the user enter their name?
        2. Can the user enter their email address?
        3. Can the user enter their username?
        4. Can the user enter their password?
        5. Can the user enter a security question?
        6. Can the user enter their credit card information?
     8. Email
        1. Verify it is required that the user enters an email address.
           1. Is there a validation email to the user?
     9. Credit Card Information
        1. Verify the software only displays the last 4 digits of a customer’s credit card number.
        2. Verify there are no cookies on the customer’s computer that contain any of the user’s confidential information.
     10. Signing In
         1. Can the user sign in from the home page?
         2. Does the software prompt the user to enter their username and password?
         3. Does the software allow the user to access the rest of the software upon successful login?
         4. If too many attempts are inputted, verify the software locks the user out of the sign in page for a brief amount of time to prevent brute force hacking.
         5. Verify the software provides a way for users to recover forgotten passwords.
     11. Username
         1. Verify the username offers form validation to make sure entered usernames are unique to the system, are long enough, and only contain standard ASCII characters.
         2. Verify the software gives an error message if the username doesn’t fit form validation standards.
         3. Verify the software prevents the user from proceeding until all errors are resolved.
     12. Password
         1. Verify the software never displays a customer’s password.
         2. Verify the password is always echoed with special characters representing typed characters.
         3. Verify that passwords run form validation to make sure passwords.
         4. Verify the software doesn’t leave any cookies on the customer’s computer containing the user’s password.
     13. Receiving Server Communication
         1. verify that a graph contains a City Name, City Population, Number of homes in city limits, number of miles in the city, and the average cost of living in the city.
            1. the names of the current and past workers can be displayed with date of hire.
            2. the history of all wages for a given job can be displayed.
            3. the two above verifications can be completed with each job that a specific city has.
            4. verification 7.1.2.1 and 7.1.2.2 can be completed for a custom job previously entered by a user.
         2. a console log or alert can verify whether a username or password is correct.
     14. Verify that the cost evaluator exists
         1. Individual evaluation
         2. Verify that the selected Job Position, Pay Grade, Yearly wage, Market Range for this position, and the percentage of variation between the market range and current pay grade are visible
         3. Verify that the user is able to adjust the wages of an employee. Verify that the projected personnel expenses change according to the adjustment.
         4. Verify that the system is able to save and load the projected wage adjustment. Verify that the user can change another employee’s projected wage without overwriting the previous adjustment.
     15. Verify that the table displays the current displays the current wages and the estimated personnel expenses of all employees over the course of a month, year, two years. Verify that the table display the adjusted wages and estimated adjusted personnel expenses of all employees over the course of a month, year, two years, etc.
     16. Verify that the table is able to be viewed in Microsoft Excel.
  3. User Input
     1. Editing City Pay Scale
        1. The System shall allow the user to create a city pay scale. [7-3.3]
           1. Verify that the user is able to enter the number of pay grade levels. [7-3.3]
           2. Verify that the user is able to enter the current minimum pay for each grade. [7-3.3]
           3. Verify that the user is able to enter the proposed minimum pay for each grade. [7-3.3]
           4. Verify that the user is able to enter the proposed market pay for each grade. [7-3.3]
           5. Verify that the user is able to enter the current maximum pay for each grade. [7-3.3]
           6. Verify that the user is able to enter the proposed maximum pay for each grade. [7-3.3]
        2. Verify that the pay scale has been stored in the database. [7-3.3]
        3. Verify that the user is able to update the city pay scale. [7-3.3]
           1. Verify that the user is able to update the number of pay grade levels. [7-3.3]
           2. Verify y that the user is able to update the current minimum pay for each grade. [7-3.3]
           3. Verify that the user is able to update the proposed minimum pay for each grade. [7-3.3]
           4. Verify that the user is able to update the proposed market pay for each grade. [7-3.3]
           5. Verify that the user is able to update the current maximum pay for each grade. [7-3.3]
           6. Verify that the user is able to update the proposed maximum pay for each grade. [7-3.3]
        4. Verify that the updated pay scale has been stored in the database. [7-3.3]
     2. Editing Position Information
        1. Verify that the user is able to create a new job position.
           1. Verify that the user is able to enter the position name. [7-3.3]
           2. Verify that the user is able to enter a description of the position. [7-3.3]
           3. Verify that the user is able to enter the department the position is a part of. [7-3.3]
           4. Verify that the user is able user to enter the position pay grade.
        2. Verify that the job position has been updated in the database. [7-3.3]
     3. Editing Employee Information
        1. The system shall allow the user to create new employees. [7-3.3]
           1. Verify that the user is able to enter the employee’s name. [7-3.3]
           2. Verify that the user is able to enter the employee’s current position. [7-3.3]
           3. Verify that the user entered position exists in the database. [7-3.3]
           4. Verify that the user is able to enter the employee’s date of hire. [7-3.3]
           5. Verify that the user is able to enter the how long the employee has been in the current position. [7-3.3]
           6. Verify that the user is able to enter the employee’s current pay. [7-3.3]
        2. Verify that the employee has been stored in the database. [7-3.3]
        3. Verify that the user is able to update employee’s information
           1. Verify that the user is able to change the employee’s name. [7-3.3]
           2. Verify that the user is able to change the employee’s pay grade. [7-3.3]
           3. Verify that the user is able to update the employee’s current position. [7-3.3]
           4. Verify that the user entered position exists in the database. [7-3.3]
           5. Verify that the user is able to change the how long the employee has been in the current position. [7-3.3]
           6. Verify that the user is able to change how long the employee has been in the current position [7-3.3]
           7. Verify that the user is able to change the employee’s current pay. [7-3.3]
        4. Verify that the employee has been updated in the database. [7-3.3]
  4. Reports
     1. A default grading system shall be suggested to the user.
        1. The grading system shall have set dimensions.
           1. Whenever a user opens the grading system for the first time, they should be prompted with how many rows and columns they want.
        2. All salaries shall be in terms of hourly pay USD.
           1. Each salary in each cell shall be in terms of dollars and cents separated by a dot with the dollar sign ($) directly to the left of the number.
        3. The initial minimum salary shall be entered in by the User.
           1. The user shall be prompted for what the minimum salary shall be.
        4. The initial maximum salary shall be entered in by the user.
           1. The user shall be prompted for what the maximum salary shall be.
        5. The default grading system shall have automatic calculations.
           1. When the grading chart is shown for the first time, each cell should already have wage values going from lowest to highest, top to bottom.
     2. The grading system shall be customizable.
        1. The minimum salary on the default chart shall be customizable.
           1. The user shall be able to change the minimum salary when desired.
        2. The maximum salary on the default chart shall be customizable.
           1. The user shall be able to change the maximum salary when desired.
        3. The dimensions of the default grading chart shall be customizable.
           1. The user shall be able to alter the number of columns in the chart when desired.
           2. The user shall be able to alter the number of rows in the chart when desired.
        4. The algorithms generating the presumed default salaries shall be customizable.
           1. The user shall be able to alter, add, or delete any algorithms from the default chart if desired.
        5. The salary display of the chart shall be customizable.
           1. The user shall be able to change the hourly pay in each cell to display other durations of time such as weekly and monthly pay if desired.
           2. All cells shall follow this change when the change is made.
        6. The order of the default design going down from smallest to largest starting salaries shall be reversable.
           1. The user shall be able to alter the default grading chart to have the highest pay grade on the first row with the lowest pay grade on the last row.
           2. This includes the first row being changed to the pay grade of 20 to represent the highest pay.
     3. There shall be a page that shows what the current city employees are being paid.
        1. This page shall use the data entered by the users to show each employee’s pay by job.
           1. Data regarding the employee is entered by the user.
           2. When a new report is generated, updated and new information is used.
        2. Verify that the system displays the pay grade.
        3. Verify that the system displays the pay range of each grade.
        4. Verify that the system displays job titles according to department.
        5. Verify that the system stores the information from this page into the database.
           1. Verify that the system stores the pay grades in the database.
           2. Verify that the system sores the pay range associated with each pay grade in the database.
           3. Verify that each job category is stored in the database

Verify that the system stores the jobs associated with each category in the database

Verify that the database keeps record of the number of people holding each job within each category.

* + - 1. This information shall be modifiable by a user with appropriate access.
         1. A user without appropriate access cannot modify the data
         2. A user with appropriate access can modify the data
      2. The page shall be downloadable as a spreadsheet.
         1. When activated, the button queues the report for download as a spreadsheet.
    1. Verify that the system can show if a job should be created or removed based on similar cities.
       1. Verify similar cities are determined by Cost of Living Adjustment.
       2. Verify that a user should be able to see how many cities of similar size have a given job.
          1. Go to a job description page.
          2. Verify that a job description page shows accurate information about what other cities have the same job.
       3. Verify that all the current cities jobs are in the list.
       4. Verify that all the jobs of other cities jobs are listed.
          1. Select a sorting option.

Verify that the sorting is correct.

* + - * 1. Select the other sorting option.

Verify that the sorting is correct.

* + - 1. Verify that a user can adjust city metrics and size.
         1. Enter new data for cities metrics and size and click save
         2. Verify that data that is entered in stored in the data base
      2. Verify that a user can see the list of other cities and their size.
         1. Verify that the data is accurate.
      3. Verify that where types of jobs are specified, prices for employing or savings for removing shall be present.
         1. Verify that calculations are accurate.
    1. Compare current data with the market.
       1. City job comparison report
          1. Verify that the city job comparison report shows how a specific job compares financially to similar jobs in other cities.

Open the report.

Verify that a comparison between the cities job and other cities similar jobs is performed.

* + - * 1. Verify that jobs are selectable to see how they compare in other cities.

Check the box next to a city.

Verify that the corresponding city is included in the comparison.

Uncheck the box next to the city.

Verify that the corresponding city is not included in the comparison.

* + - * 1. Verify cities are listed with the title they use for that job.
        2. Verify that a minimum hourly pay for the jobs is listed for each city.
        3. Verify that a maximum hourly pay for the jobs is listed for each city.
        4. Verify that an adjusted minimum hourly pay for the jobs is listed for each city.
        5. Verify that an adjusted maximum hourly pay for the jobs is listed with each city.
        6. Verify that cities are selectable.

Check the checkbox.

Verify that the city is selected.

Uncheck the checkbox.

Verify that the city is unselected.

* + - * 1. Verify that only selected cities shall be factored in market calculations.

Uncheck the checkbox next to a city

Verify that the market calculations do not include that city.

* + - * 1. Verify that the report is downloadable as a spreadsheet.

Press download worksheet.

Verify that the spreadsheet is downloaded.

Open the document that was downloaded.

Verify that the report matches the document’s contents.

* + - 1. City Compared to Market Report
         1. Verify that the city compared to market report shows how each job title compares to the market’s average values.
         2. Verify that the system highlights extreme differences between the current county and the market average.
         3. Verify that each job title in the city is listed.
         4. Verify that the minimum hourly pay for the city is listed for each job.
         5. Verify that the maximum hourly pay for the city is listed for each job.
         6. Verify that each job title from the market is listed next to the corresponding city’s listed job.
         7. Verify that the minimum hourly pay for the market is listed for each job.
         8. Verify that the maximum hourly pay for the market is listed or each job.
         9. Verify that the city compared to market report is downloadable as a spreadsheet.

Press download spreadsheet.

Verify that a spreadsheet is downloaded.

Open the document that was downloaded

Verify that the report matches the document’s contents.

* + 1. Forecasted budget impact report
       1. Navigate to the reports page.
       2. Ensure employee wage, job title, and total employee budge is up to date.
       3. Verify that the generated forecasted budget impact report correctly calculates total employee cost, market value cost, and the impact this difference makes to the total employee budget.
       4. The budget impact report shall rely on data calculated from the Cost Estimator.
          1. The report successfully retrieves data from the Cost Estimator.
       5. As with all reports, the forecasted budget impact report shall be downloadable as a spreadsheet.
       6. The forecasted budget impact report shall make a comparison between the market value employee wage budget and the entities actual employee wage budget.
          1. The forecasted budget impact figure is available in the report
          2. The report shall calculate the total cost of the entity’s employees.

The total cost is verifiable against the entity’s records.

* + - * 1. The report shall calculate the total cost of the entity’s employees at market value

The total cost of employees at market value is a verifiable figure.

* + - * 1. The report shall calculate the difference of between employee total cost and market value, creating a budget impact figure.

The budget impact figure is mathematically verifiable.

* + 1. The system shall have access to current information regarding jobs.
       1. Verify that the system keeps track of job titles.
          1. Verify that the job titles have a prioritizing system that suggests common job titles.
       2. Verify that the system shall keep a record of similar job titles.
          1. Verify that the system suggests possible job title variations.
       3. Verify that there is a visible pay scale for job titles in similar cities.
          1. Verify that the pay scale assigned to a job title references the range for that job.
       4. Verify that the system keeps record of the amount of time spent in a job.
          1. Verify that the system suggests payment ranges when adding new employees.
    2. Verify that the system has access to current information regarding employees.
       1. Verify that the system shall keep track of job titles.
          1. Verify that the Job titles shall have a prioritizing system that suggests common job titles. The Job title that is most common should be suggested.
       2. Verify that the system shall keep a record job titles that are like each other.
          1. Verify that the system shall suggest possible job title variation when inputting new job titles. When the user goes to type I a job it will ask if they are sure the job title is correct and suggest a job title.
       3. Verify that there shall be a pay scale that is visible for a job title in similar cities.
          1. Verify that there shall be a pay scale key assigned to a job title that references the range for that particular job title.
       4. Verify that the system shall keep a record of the amount of time spent in a job to change placement in pay scale. An employee will be at the bottom of the pay scale when starting.
          1. Verify that the system shall suggest payment ranges when adding new employees. Will check the scale for that job title and suggest a starting wage.

NON-ESSENTIAL REQUIREMENTS

Database:

* + 1. The database may allow for future storage of, a yet to be developed, RBDC Living Wage Calculator. [1-1]

Appendices

* 1. Credits
     1. A-team Team Members
        1. Mario da Silva Filho
        2. Lucas Frietas
        3. Kwok Moon Ho
        4. Nimesh Subedi
        5. Heather Hanks
        6. Justen Neeley

Ctrl Alt Elite Team Members

BronzeBeard Team Members

Triforce Heroes Team Members

Low Lifes Team Members

Chris, Chris, & Co. Team Members

SegFault Team Members

Team Lead Committee

* + - 1. Mario da Silva Filho

Functional Manager committee

* + - 1. Kwok Moon Ho

Scrum Master committee

* + - 1. Lucas Frietas

Chief Editor committee

* + - 1. Justen Neeley

Tester committee

* + - 1. Heather Hanks

QA committee

* + - 1. Nimesh Subedi
      2. Daniel Slaugh
      3. Romina Palacios
      4. Chingyu Chen
      5. Trevik Peterson
      6. Zac Turner
      7. Alex Shnyrov