EECS 2311 PROJECT

Iteration 1: Starting the Implementation

Stefan Smolovic – 218149567

Fares Trad – 217281379

Kasemi Osaroedey – 215978703

Joshua Keppo – 210971752

John Donato Prabahar - 219087279

Reward Management System Vision Statement

The reward management system (RMS) will be used to incentivise and motivate members of an organization or group to perform high quality work or exhibit exemplary behavior. The goal of the app is to allow managers to reinforce a work culture in the company that is in line with the companies mission and values. Managers will be able to set up tasks that when completed will reward individuals by increasing their “reward points” total. The “reward points” will be redeemable for different prizes which will be chosen by the administrator of the system.

This application has 2 main kinds of users. The 2 groups of users are the administrators and the individuals receiving points. The administrator’s role is to set up tasks, administer points when tasks are completed, and overall have control of how the app is used within their organization. The points receivers will be able to notify the admin when a task is complete, they will be able to receive points, keep track of their total points, and spend points on different rewards, such as gift cards, or days off.

On top of being able to administer, receive, and spend points, which are the main functionalities of the app, both administrative and point-receiving users will have access to a number of different statistics which will assist the organization see trends and find ways of increasing productivity, and decreasing common issues.

A scalable database will be used to store all data related to the application. This will allow organizations and institutions of any size to use this app, whether they have a few users, or hundreds of users. The statistical data will be able to be exported as CSV files which will enable the use of spreadsheet software such as MS Excel for more in depth analysis of the data.

Initially the application has been geared towards businesses, big and small, but future iterations of the software will allow for modular customization to expand the use cases of the app far beyond business. Ideally the app will be usable by teachers, and their students, parents, and their children, and in general any sort of group where positive reinforcement is used to achieve desired outcomes.

Existing apps serving a similar purpose to this software are feature heavy and make it difficult to get started with using them. The goal of this app is to simplify the process and make it easy to use out of the box. The UI and features have been planned intentionally to create a streamlined, simple-to-use system which can still meet the needs of any business.

The project will be considered a success if the following criteria are met. First, the app must be fully functioning passing all testing methods used. Including unit tests, and user testing. Secondly the app must increase productivity of it’s users by a minimum of 25%. A definition and measure of productivity will be defined by the administrator of the application, and we will track the specified measure. And the final criteria this app must fulfill is it must satisfy all users of the application. Surveys and direct communication with the users will be used to determine if this criterion is met.

Banking / Points Management App: Big user story

Iteration 1

Account Creation/Sign up

User can create an account;
Allows the user to Log in to the system using a username and a password that they create

Priority: High Cost: 7 days

Withdraw/Deposit

User can withdraw and deposit fund into their account

Priority: High Cost: 7 Days

Basic GUI

Basic Graphical User Interface

Which provides structure to the Software

Priority: High Cost: 7 Days

Iteration 2

Data Base

A Data Base stores in User Informations

Priority: High Cost: 8 Days

Admin UI

Admin UI will contain additional features that the user UI does not

Priority: High Cost: 7 Days

Expanded User UI Features

Allows the user to have an easier use of the UI, and allows user to transfer

Priority: High Cost: 6 Days

Iteration 3

Account/Admin Statistics

Allows user to see statics about the accumulation of funds on their account in the form of graph

Priority: High Cost: 6 Days

Exportable Database File

Allows user to export their data in CSV file

Priority: High Cost: 6 Days

Iteration 3

Database hosted online

Allows us to fetch information from serves hosted online

Priority: High

Cost: 6 Days

Customizable

Allows the app to be more suited to more situations

Priority: High

Cost: 6 Days

Banking / Points Management App: Detailed story

Account Creation

User can create an account;
Allows the user to log in to the system using a username and a password that they create

Priority: High Cost: 2 days

Withdraw / Deposit

User can withdraw and deposit fund into their account

Priority: High Cost: 4 Days

Admin Curation

Allows admin to have privileges over users such as account deletion, rejection, resetting, etc.

Priority: High Cost: 2 Days

Display User Information

GUI to see their account information; name, balance, email, phone, etc.

Priority: High

Cost: 2 Days

Clear Graphical User Interface

A clear and tidy user interface; made user friendly

Priority: Medium

Cost: 2 Days

Change Password

User should be able to change their password

Priority: High

Cost: 1 Days

Log of Points

Allows the user to see logs on their pain accumilation which contains information of date and amount of points and short description of what they were for.

Priority: Medium

Cost: 2 Days

Detailed User Stories for Iteration 2

Relational database using mySQL:

Create a database to allow for permanent data storage

Cost: 3 days

Priority: High

Database Interface Class:

Implement a java class which will interface the SQL database

Cost: 3 days

Priority: High

Filter Data Method:

Implement methods in Database Interface class to filter data (i.e returning a list with only point receiver objects)

Cost: 4 days

Priority: Medium-High

Sort Data Method

Implement methods in Database Interface class to sort data (i.e sorting point receivers by most number of points to least)

Cost: 4 days

Priority: Medium-High

Refactor GUI code where necessary

Make improvements to codebase, removing unnecessary code, and reducing number of lines needed to implement methods.

Cost: 2 days

Priority: Low

Statistics functionality

Create methods to conduct statistical analysis on data from database

Cost: 3 days

Priority: High

Statistics UI

Create statistics GUI for both the admin and point receiver dash boards

Cost: 3 days

Priority: High