

# **Project Edward: Final Project Documentation for UTM CSCI 352**

Hayden Nanney & Matt Mosley

## **Abstract**

This project is an application for those who are planning for their retirement at any point in their career. The goal of the project is to allow the user's to set retirement saving's goals and track their progress along the way. It can also be used to forecast what your savings will be if you adhere to a certain investment strategy.

## **1. Introduction**

This project will allow a user to enter in financial information such as income, filing status, year, as well as whether they are contributing to a pretax retirement savings account to accurately calculated taxes and tax deductions for each tax year. This will allow the app to accurately reflect its user's financial history as well as help them to plan for their retirement while early in their career, which is most beneficial due to compounding interest.

### **1.1. Background**

This application may include large amounts of financial jargon such as marginal tax rates, adjusted gross income, etc. We aim to make definitions and explanations of these terms available in app, so we can make this information more accessible.

### **1.2. Impacts**

This application should impact the financial literacy of its users, since financial education is sorely lacking in most education programs.

### **1.3. Challenges**

We expect some headaches when we start to implement the tax laws and regulations since it will be time consuming to get everything correct and easily changeable.

Current challenges are how we should document all of the tax brackets and we are debating on using a database with each table storing one year's tax bracket data.

## **2. Scope**

At the completion of this project the end user will be able to enter information that will affect their tax status such as salary, filing status, and location and then the app will project how this will impact their retirement savings accounts and provide the user a resource to show how much they need to save and invest per year to meet their goals.

### **2.1. Requirements**

The following requirements were generated based on the information needed to make financial advice.

#### **2.1.1. Functional.**

- The application will store the user's financial history data.
- The user will have a profile that will be able to be accessed on multiple visits
- The application will give the user their projecting retirement savings account worth based on a savings plan.

#### **2.1.2. Non-Functional.**

- The data will be stored securely within the app/database.
- The application will be responsive to all user input and be intuitive to use.
- The application will reliably provide the user with accurate data projections.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	View Profile	User	Low	1
2	Salary	User	Low	1

TABLE 1. USE CASE TABLE

## 2.2. Use Cases

Use Case Number: 1

Use Case Name: View Profile

Description: The user on our application wishes to view their account and the information contained in there. They will click on the "View Profile" button. Which will then refer them to their profile screen with their information contained in it.

- 1) User wishes to see their account profile, first they would need to log into their account.
- 2) The user then clicks on the View Profile button.
- 3) The user is then directed to the profile page where they can view their information

Termination Outcome: The user now is able to view their profile and view their information.

Use Case Number: 2

Use Case Name: Salary

Description: The user in the application's home screen can tap on this button to view their current salary and how much they are earning, along with their paychecks and incoming and outgoing payments.

- 1) User wishes to see their salary.
- 2) User clicks on the View Salary button.
- 3) The user is directed to the Salary page where they can view their information

Termination Outcome: The user now can view their salary information.

Use Case Number: 3

Use Case Name: Home

Description: The user will be able to go to the home screen to see updates about their information, including salary and transactions.

- 1) The user wishes to view the home page.
- 2) When the user loads up the application, this page is default, but if they are in another tab, they will click on the Home button.
- 3) The user is directed to the Home page of the application.

Termination Outcome: The user can now view the home screen and see their information.

## 2.3. Interface Mockups



Figure 1. Login Page



Figure 2. Home Page

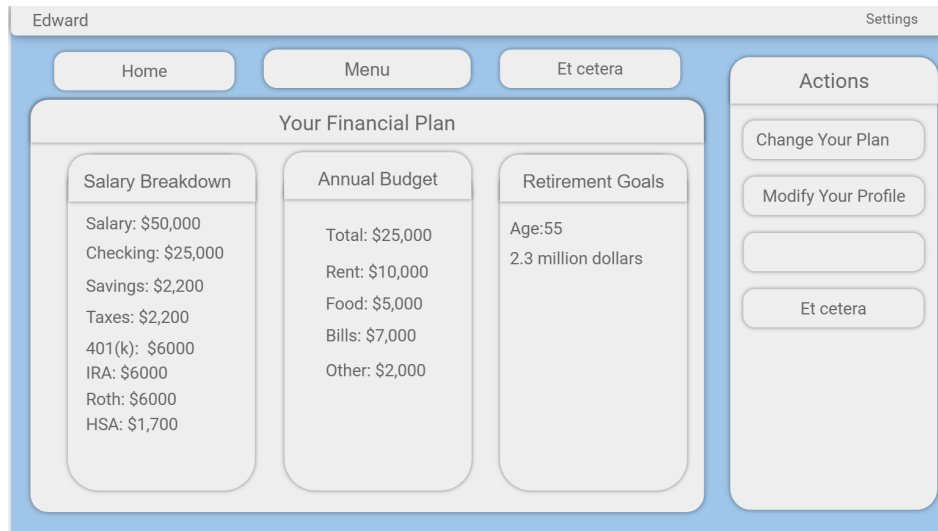


Figure 3. Financial Plan Page

### 3. Project Timeline

Timeline of events:

September 9 - Project Proposal Draft  
 September 29 - Project Proposal Update  
 October 6 - UI Design  
 October 8 - UI Presentation  
 October 29 - Tracer Code  
 October 30 - Proposal Update  
 November 3 - Tax Database Connection  
 November 4 - Test Backend Functionality  
 November 8 - Begin Integrating Backend & Front-End  
 November 15 - Writeup  
 November 23 - Present Project

### 4. Project Structure

The UI is structured so that each page uses the same basic format with special buttons or features that have unique abilities.

The Backend will likely contain a database with tax bracket data. It also will do all of the calculations to find what taxes a user will pay and can project what the user's net retirement savings will be in the future. The user's profile contains a list of all the historical tax data for the user in a list. This allows the functions to easily iterate over the list to look up tax bracket data from the database to calculate the taxes and other retirement savings predictions.

## 4.1. UML Outline

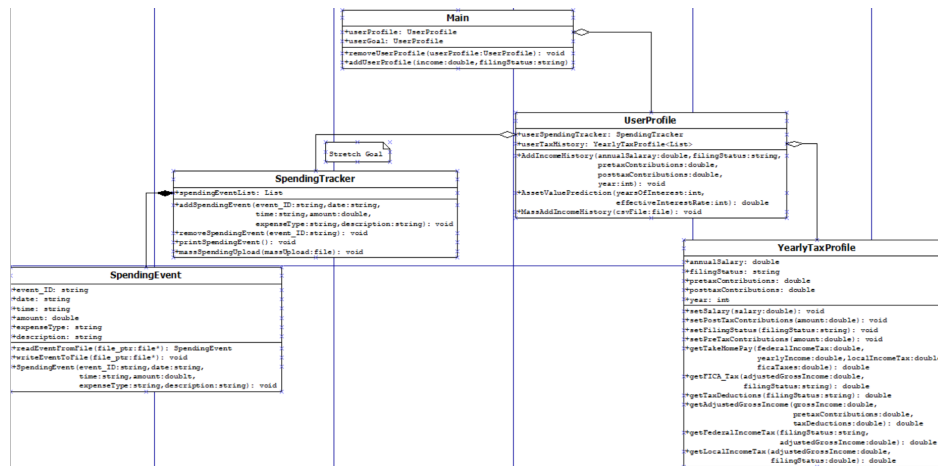


Figure 4. Backend UML

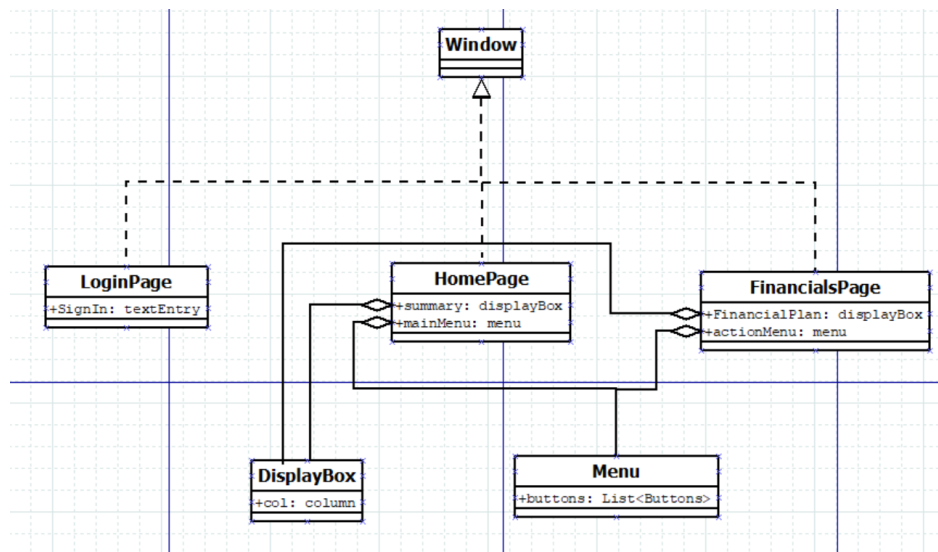


Figure 5. UI UML

## **4.2. Design Patterns Used**

We will likely use the decorator factory pattern to design each page of the UI, since each one needs to be implemented slightly differently.

## **5. Results**

We have a general idea of where we want our project to head towards.

### **5.1. Future Work**

The next steps are to get an example UI working and have the backend tracer code connected to the database, so we can test if it is functionally accurate.

Fix Use Cases in documentation.