

# **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 calculate 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**

I 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 projected retirement savings account worth.

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

- The data will be stored securely within the 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	Sign Up	User	High	1
2	Login	User	High	1
3	View Records	User	High	1
4	Add Record	User	High	1
5	Logout	User	High	1

TABLE 1. USE CASE TABLE

## 2.2. Use Cases

Use Case Number: 1

Use Case Name: Sign Up

Description: The user wishes to sign up for an account with this application.

Precondition: The user has this application installed.

- 1) The user clicks the Sign Up button.
- 2) The user chooses a username and password.
- 3) The user clicks the submit button and their username and password are added to the database.

Termination Outcome: The user now has an account with this application.

Use Case Number: 2

Use Case Name: Login

Description: The user wishes to login to their account.

Precondition: They have an account with the application.

- 1) The user enters their username and password into the textboxes.
- 2) The user clicks the Login button and their information is checked from the database.

Termination Outcome: The user has logged into their account and is at the homepage.

Use Case Number: 3

Use Case Name: View Records

Description: The user will be able to view all of their records held in the application in tabulated form.

Precondition: The user is logged into their account.

- 1) The user clicks on the 'View Records' button and a new window opens displaying their records.

Termination Outcome: The user can now view their records as a spreadsheet.

Use Case Number: 4

Use Case Name: Add Record

Description: The user will be able to add a record to their profile.

Precondition: The user is logged into their account.

- 1) The user clicks on the 'Add Record' button and a new window opens.
- 2) The user selects their Income, Filing Status, Pretax contributions, posttax contributions, and the year from dropdown menus.
- 3) The user clicks the 'Submit' button and the record is added to their profile.

Termination Outcome: The user has added a record to their profile and it is reflected in the database.

Use Case Number: 5

Use Case Name: Logout

Description: The user will be logged out and the application closed.

Precondition: The user is logged into their account.

- 1) The user clicks on the 'Logout' button and the app closes.

Termination Outcome: The user is logged out of their account.

## 2.3. Interface Mockups

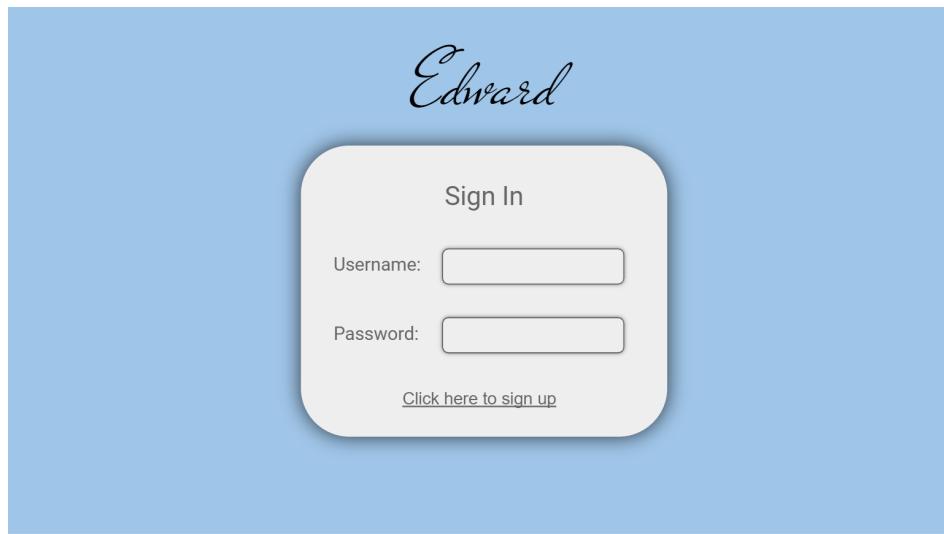


Figure 1. Login Page



Figure 2. Home Page

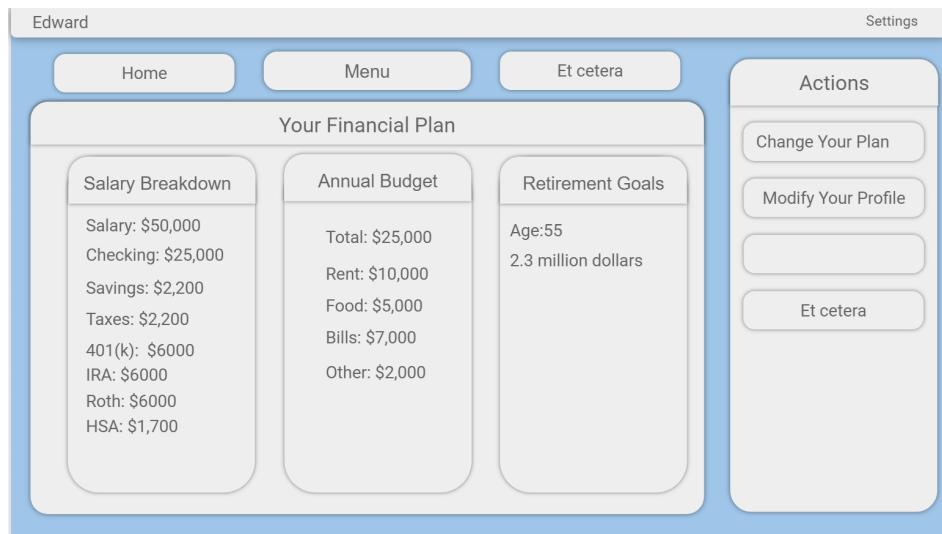


Figure 3. Financial Plan Page

### 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 contains 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

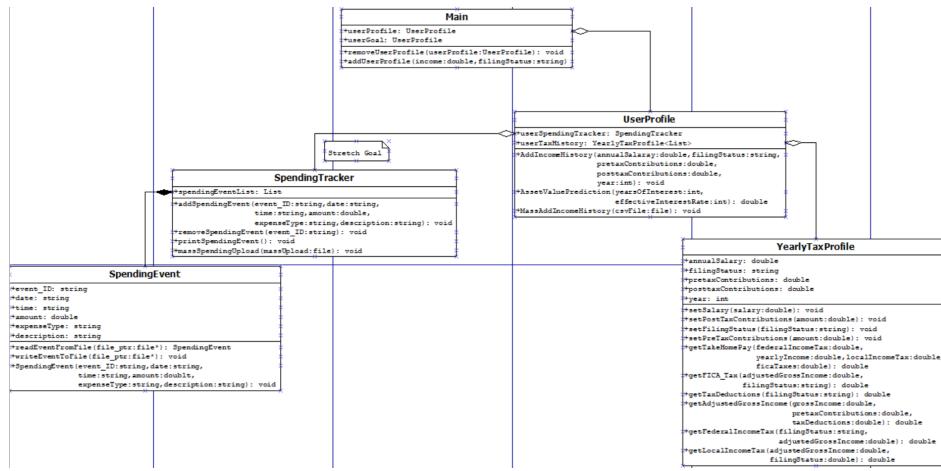


Figure 4. Backend UML

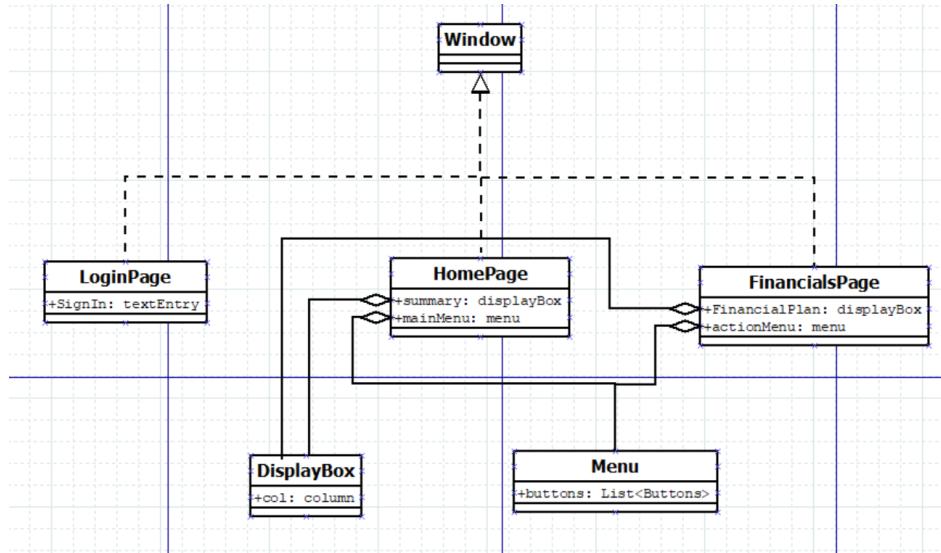


Figure 5. UI UML

## **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

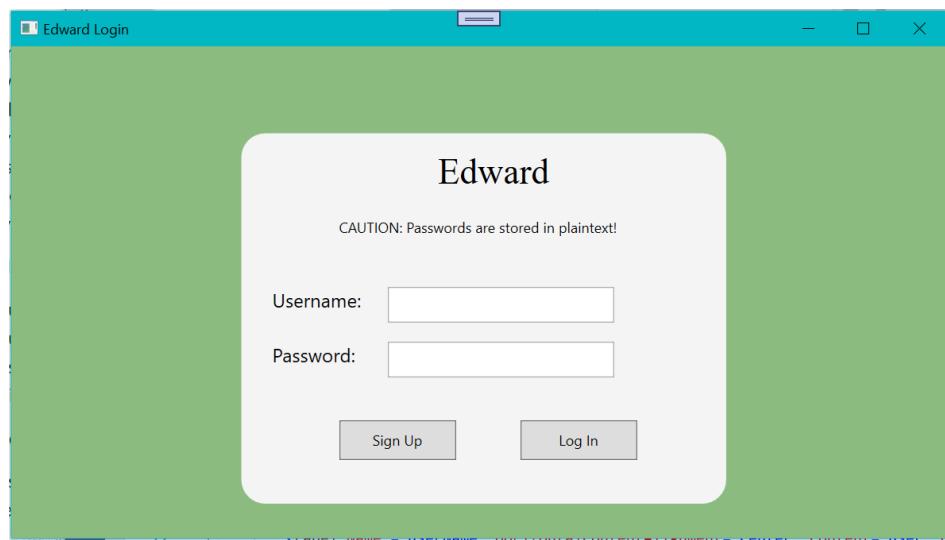


Figure 6. Login Page



Figure 7. Home Page

The screenshot shows a window titled "AddRecordWindow". Inside, there is a white rounded rectangle containing five input fields and a submit button. The fields are labeled "Yearly Income:", "Filing Status:", "Pretax Contributions:", "Posttax Contributions:", and "Year:". Each field has a dropdown arrow next to it. Below these fields is a large grey "Submit" button.

Figure 8. Add Record Page

The screenshot shows a window titled "ViewRecordsWindow". Inside, there is a table with the following data:

	UID	IDNUM	SALARY	FILINGSTATUS	PRETAX	POSTTAX	YEAR	
7	2	37000	Single		4000	3000	2014	
8	2	37000	Single		2000	5000	2015	
9	2	38000	Single		4000	6000	2016	
10	2	39000	Single		4000	4000	2017	
11	2	40000	Single		2000	3000	2018	
12	2	45000	Single		8000	3000	2019	
4	2	29000	Married Filing Jointly		3000	4000	2011	
5	2	30000	Married Filing Jointly		2000	5000	2012	
6	2	25000	Married Filing Jointly		2000	2000	2013	
2	2	35000	Married Filing Separate		2000	7000	2010	

Figure 9. View Records Page

### 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.