**Money Management Application**

**Daniyal and Saket**

# PROJECT PROBLEM STATEMENT

Discuss the problem or opportunity that is scheduled for resolution.

**A company is asking us to create a personal finance tracker that can allow users to log and categorize their expenses. The application will give monthly reports, provide graphically friendly charts and category filters to make it easier to manage**.

# CONCISE DESIGN OVERVIEW

After researching potential technologies, you will identify the components of your solution. Develop a prototype or model of what your final solution would look like, providing abstract information about the individual components.

**Using JavaFX, we can code the application. The components of our solution will consist of certain classes Expenses, Category and Budget. The prototype of the model will consist of a simple but working application with basic GUI, a login page and basic money management.**

# SCOPE STATEMENT

SMART goals or objectives would include deliverables. Clarify as necessary what the scope includes and **does not include.**

**The application will rely a lot on OOP programing. Using this we will divide certain deliverables for the project. The scope includes the model layer, containing the expense, category and asset classes. The persistence layer is for saving data, containing the ExpenseRepository and FileExpenseRespository. The Business Logic will have the ExpsenseManger class.**

# TIMELINE

Draw a timeline, including milestones to serve as the basis for a work breakdown structure (WBS) and appropriate Gantt or PERT charts. For this class, you can refer to the Work Breakdown Excel Sheet. This can be a table with anticipated tasks listed for each school day of the project. You will submit completed chart(or Record of Tasks) along with your final deliverables.

**We will create a timeline using excel with the dates and when certain objectives are started/completed.**

# TEST PLAN

Describe how you will test your solution. The plan should include testing the whole solution and testing individual components of the solution. Include details such as if you will automate testing or you will manually test with positive and negative test cases.

**We will test our solution step by step when we create each OOP class. We will finish test our login page, then our main page where the user enters details, then the user profit, and lastly the data calculator.**

# RISKS AND CONTINGENCY PLANS

List known and potential risks by estimated probability, with mitigation plans.

Version control issues with GitHub, poor user input (user caused errors), possible program crashes, and collaboration issues within the group. To prevent GitHub control issues, each member in the group must state when they are going to do an action with the code relating to GitHub. Various methods of testing the program will help us make it “dummy” proof, so the application knows what to do if a user inputs wrong data. Possible program crashes may occur, which is why code should be backed up on git to prevent data loss. For collaboration issues within the group, the instructor can be notified allowing for a solution or extra commendation.