

# Project Proposal - Malcolm, Minh, Zul

---

## Statement

Our final project is a budgeting application that helps users manage their savings and transactions. The goal of the application is to help users visualize their financial transactions, set saving goals, and make suggestions on where to reduce spending or allocate extra money. The application will showcase our proficiency in JavaFX, graphical user interfaces, and software design principles. The program was created to address a problem we see in the financial applications that we use. Often, the applications that give you up-to-date info on how much money you spend and receive, such as bank statements, do not include tools that help you manage your finances. These tools should do things such as break down the nature of your transactions and give relevant analytics, set budgeting goals, or list things that the user needs or wants to pay for. This application integrates these features into a single package, thereby addressing the aforementioned problem.

## Technical Outline

The project will utilize the following data structures and techniques:

- Classes and objects for managing transactions, accounts, and categories
- Hashmaps for storing Account data specifically.
- JavaFX components such as TableView, TabPane, PieChart, and Canvas for the user interface
- Trello for Project Management so we can create story maps and break the project down into small tasks.
- Loading and saving data, such as to a .dat file
- Integration with GitHub for version control and collaboration
- Group communication with a discord server and resource/reference storage
- Drag-and-Drop functionality:
  - Budget categories: Allow users to drag-and-drop budget categories to reorder them or to move a category between different groups (such as needs, wants, or savings). This would enable users to easily organize and prioritize their budget categories based on their personal preferences.

## Backend

- **Transaction** class representing a single transaction, with fields such as name, amount, date, and payment method
- **Account** class representing a user account, with fields such as account balance and transaction history
- **Category** class representing a spending category, with fields such as name and transactions

## Frontend

- Main window with tabs for switching between ATM and spending tracker features
- Budget tab with deposit, withdrawal, and transfer functionality
- Spending tracker tab with transaction history, spending visualization, and budgeting tools

## Bibliography

1. JavaFX Documentation: Official JavaFX documentation to learn about various UI components and their usage.
2. JavaFX Tutorials: Online tutorials to understand how to create and manage JavaFX applications.
3. Design Inspiration: Websites like Dribbble and Behance for UI design inspiration and ideas.

## Objectives

Our objectives for this project are as follows:

1. Intro screen: Create an attractive and visually appealing intro screen and overall user interface using JavaFX components, incorporating design inspiration from various sources; intro screen should incorporate the application's logo and an animation.
2. Login screen: screen prompting user for their login information, which will access the persistent user data if they are not loggin in for the first time.
3. NWS (Needs/Wants/Savings Goals) page: create a page that prompts the user for their spending needs, wants, and savings goals that help users with budgeting, such as setting saving goals, amount of budget remaining, amount of budget saved/exceeded/...
4. Transactions page: page for inputting and storing individual transactions.
5. Transaction trends/metrics visualization: Implement a visually informative way to display helpful information to the user about their spending behavior and trends, such as using graphs or charts.
6. Spending category breakdown: Create a feature that categorizes users' spending and provides an overview of their spending habits in each category.
7. Animation for transactions: Implement smooth animations for transaction-related actions, such as adding, editing, or deleting transactions.
8. Drag-and-drop functionality: Implement drag-and-drop capabilities for easy manipulation and organization of transactions and budget elements.
9. Data persistence and file handling: Implement a robust back-end system for loading and saving user data, ensuring the application can reliably store and retrieve user information.
10. Resizability: make windows/components/widgets resizable, make them properly adjust to the user's screen resolution. Properly display app on mobile.
11. Back-end data integration between modules: choose data structures and ensure changes update other modules: transactions <-> NWS <-> metrics (visualization) <-> suggestions/allowances
12. Suggestions/Allowances: effectively convey to the user how much they can spend on needs/wants and how close they are to a limit in order to reach their savings goals.

## Dream Objectives

1. Real-time suggestions and feedback: Provide real-time suggestions and feedback to users based on their spending habits, helping them make better financial decisions.
2. Deploy our application either through an executable or through a web app.
3. Automated testing harness

## Individuality and Collaboration

The project will exhibit individuality by offering unique features and a customized design inspired by various sources. The team will collaborate effectively by dividing tasks and responsibilities, ensuring a cohesive final product.

## Teamwork and Reflection

Upon completion of the project, our team will provide a reflection on our experience, detailing the following:

- A description of our group's work process and division of tasks
- A list of resources used to implement each objective
- A discussion of the objectives that were particularly challenging
- A description of any aspects that don't work as desired
- A description of what we would have improved if we had more time

Additionally, each team member will individually reflect on:

- Their personal contributions and self-assessment of their work
- Their partners' contributions and assessment of their partners' work

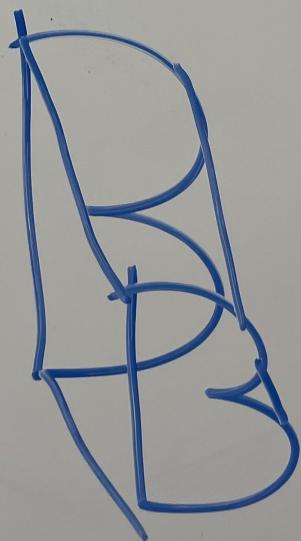
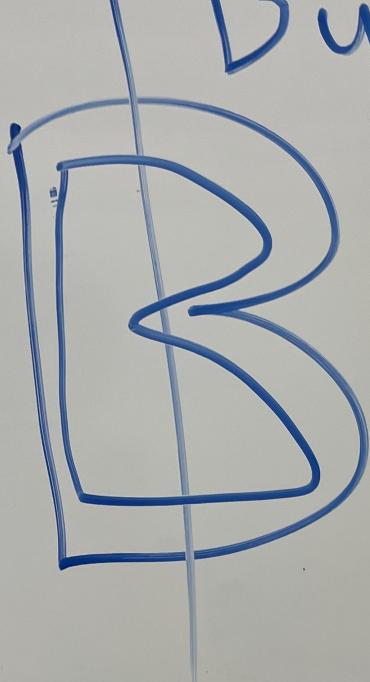
## Project members

- [Malcolm Bell](#) - mbell1@colgate.edu
- [Minh Tao](#) - mtao@colgate.edu
- [Zulnorain Ahmed](#) - zahmed@colgate.edu

## Mockup Photos

### Logo Design

Better Budget



•





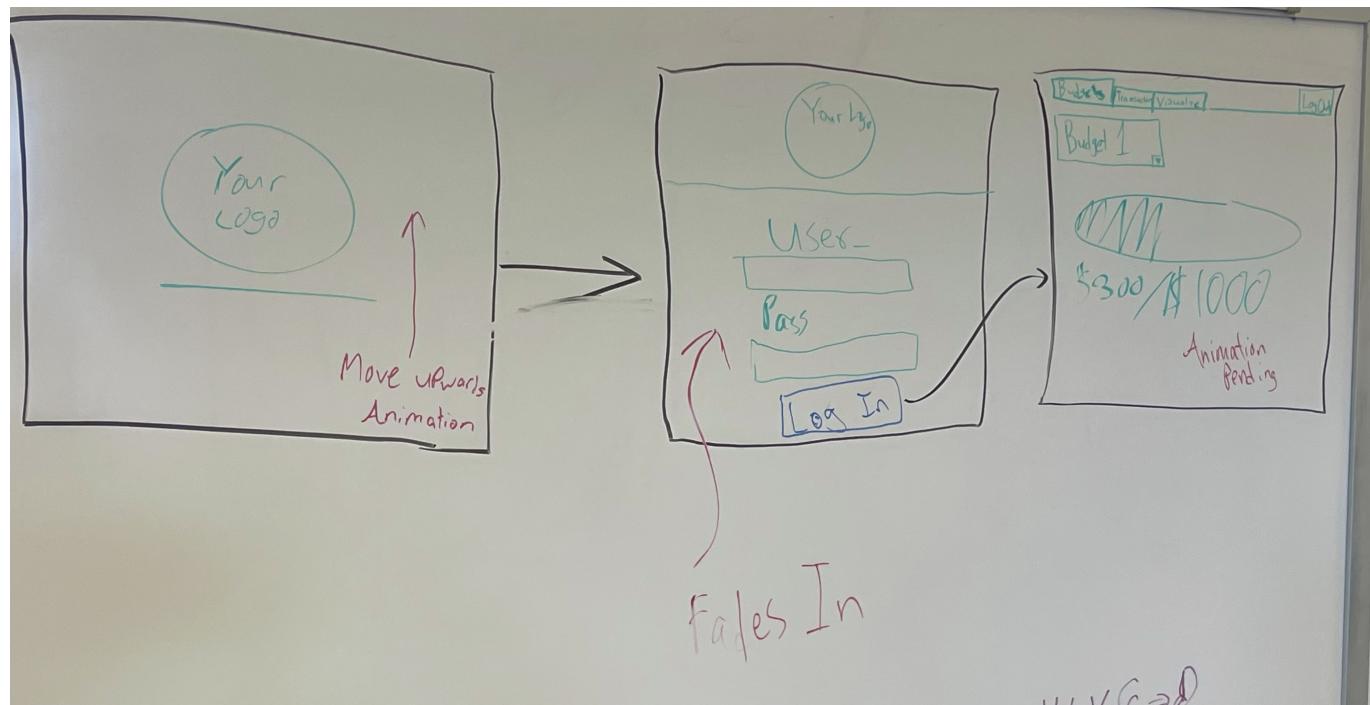
Login/Title Screen

Account Number:

PIN:

[Click here if you don't have an account yet.](#)

**Log in**



## Transactions List

The screenshot shows a software interface for managing transactions. At the top, there are two tabs: "Transactions" (selected) and "Budgeting".

**Transaction List:** A table with columns "Name", "Amount", "Category", and "Date". A message "No content in table" is displayed below the table.

**Add New Transactions:** A form with fields for "Name" (Minecraft), "Amount" (\$29.99), "Category" (Gaming), and "Transaction Date" (a date input field). An "Add Transaction" button is at the bottom.

**Filter Transactions:** A section with a dropdown menu "Filter transactions by: Date". It includes a dropdown "Find transactions before the date:" and a date input field. A "Filter" button is at the bottom.

## Program Animations/Functionality

