

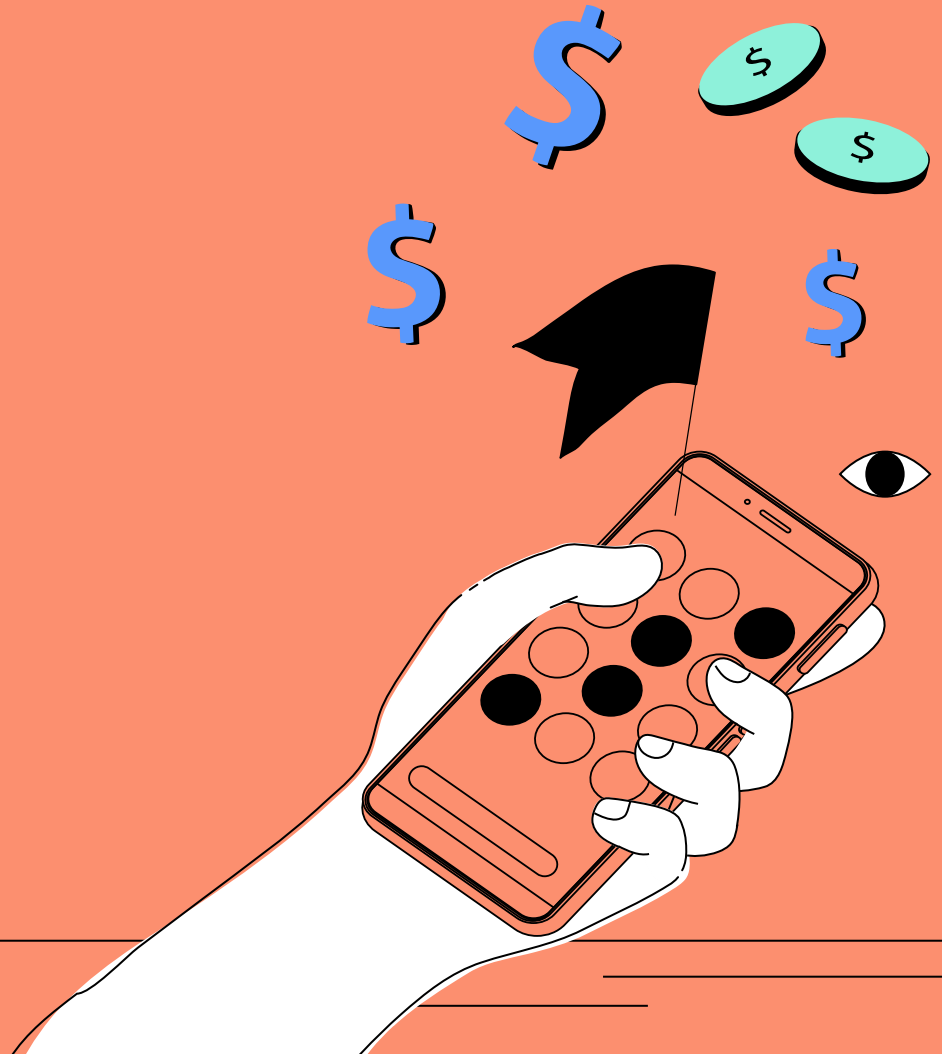
Budget Buddy!

CS Enjoys:
Dheer Prajapati, Max Stroh, Akash Dubey,
Khoi Lam, Mustafa Hawa



01

Problem Statement





The Challenge

Problem: Many individuals view budgeting as a chore or something that doesn't fit into their busy schedules. Existing budgeting apps often lack user-friendliness and fail to address this fundamental issue.

Solution: Budget Buddy is here to change the game. We've created an intuitive and effortless budgeting app that seamlessly integrates into your lifestyle.





Identifying the Challenge



Spending too much

As college students we are prone to overspending due to being the first time away from parental control.



Money Distribution

Banking Applications only provide a distribution that is categorized by the month.



Missing payments

Based on our research we found that individuals tend to miss important payments just because they didn't get a heads up a couple of days/hours before.

02

Our Solution





Budget Buddy!

- Budget Buddy is an innovative application designed to revolutionize the way you manage your finances.
- It offers personalized budgeting plans tailored to your individual's financial situation.
- Our user-friendly interface makes budgeting efficient and enjoyable for users of all levels.



Key Features

Security Features

Budget buddy has a series of robust security features that can block any sort of suspicious activity.

Easy to use UI

Gives user a easy to use UI which makes it easy for users to navigate through the app.

Payment Reminders

Reminds users to pay upcoming bill sby sending a notification.

Expense Tracking

Gives users the ability to track their expenses by categorizing transactions

Income Tracking

Allows users to track where there income is coming from

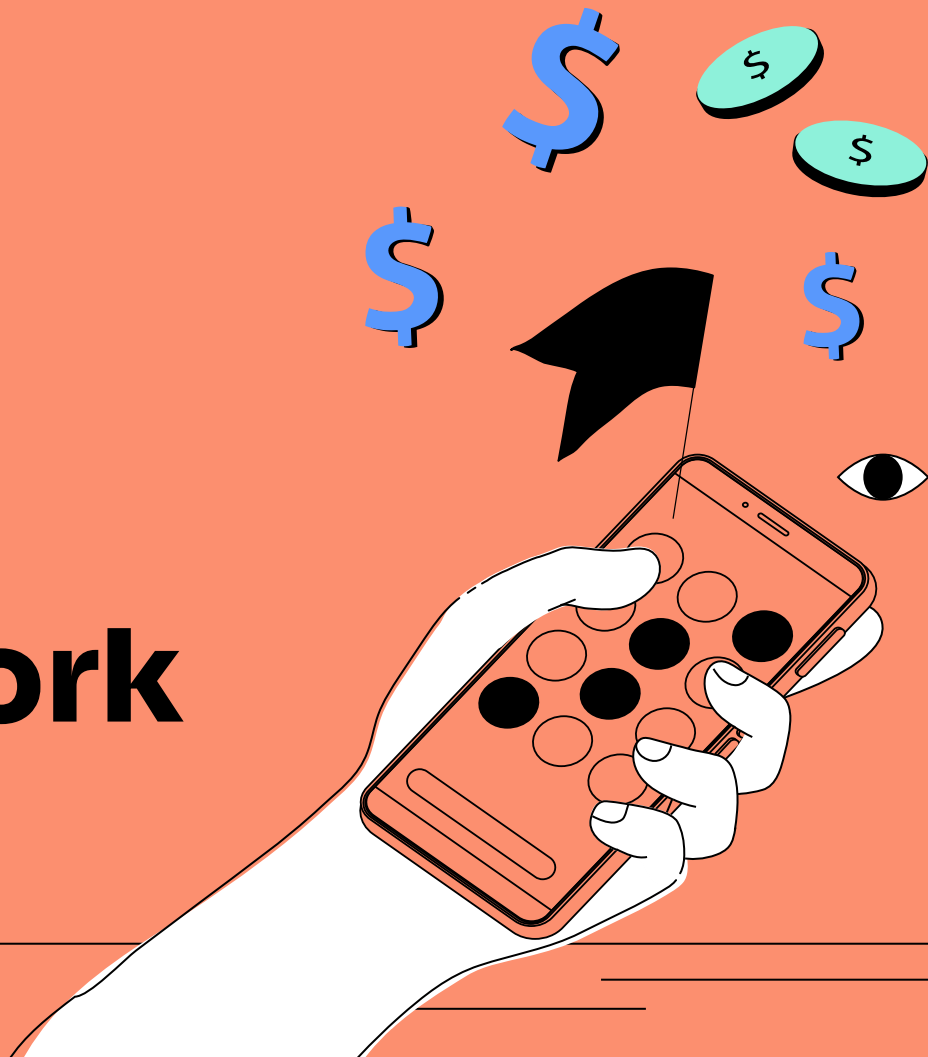
Goal Setting

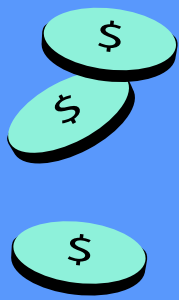
Allows users to set financial goals.



03

Related Work

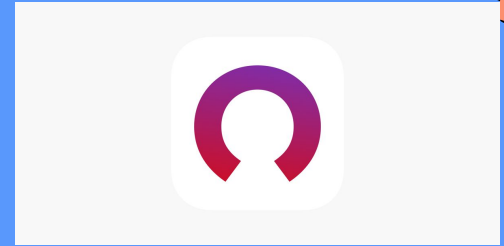




Exploring Similar Tools



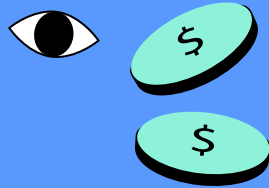
- **Rocket Money**
 - Provide subscriptions management
 - Monthly spending
- **Quicken**
 - Tracking spending
 - Manage debt



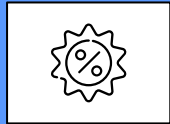
04

Concepts from Class





Applying Class Concepts



Storyboarding



Pair Programming



Black Box
Testing

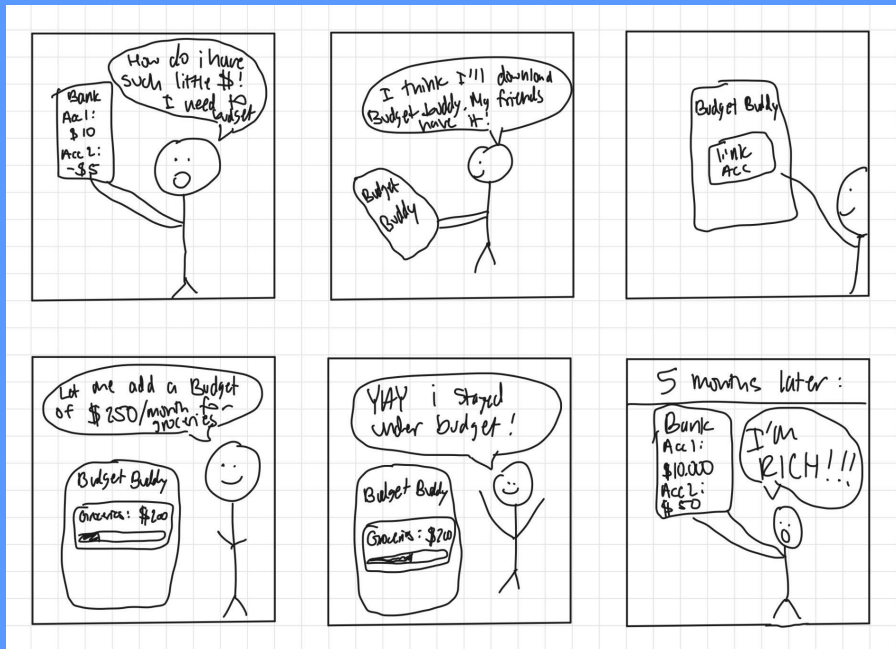


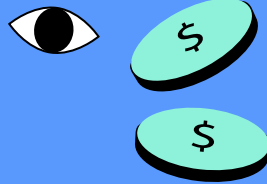
Project
Management



Storyboarding

- Used storyboarding to help in our developmental process
- Useful because:
 - Brainstorming
 - Identifying challenges
 - Breaking down project into manageable tasks





Pair Programming

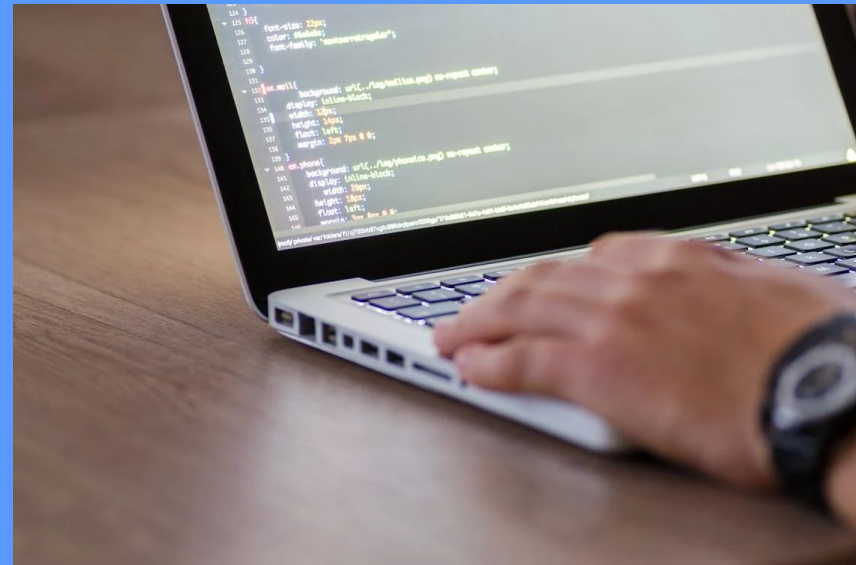
- Agile Software Technique
- Involves two programmers
 - The driver
 - The observer
- The observer strategizes
 - Overlooks the each line of code written
- The driver writes the code
- Driver has to think less
- Less errors and more efficient





Black Box Testing

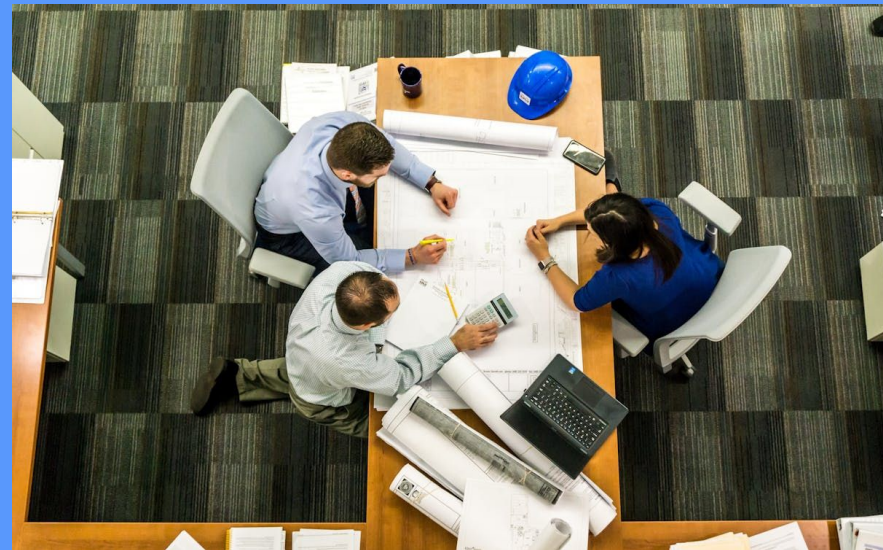
- Tester is unaware of internal workings
 - Doesn't know about the source code
- Can provide better data from testing
 - Can understand how a system will actually respond in a real scenario
- Can test unexpected actions
- Can cover a lot of tests and scenarios





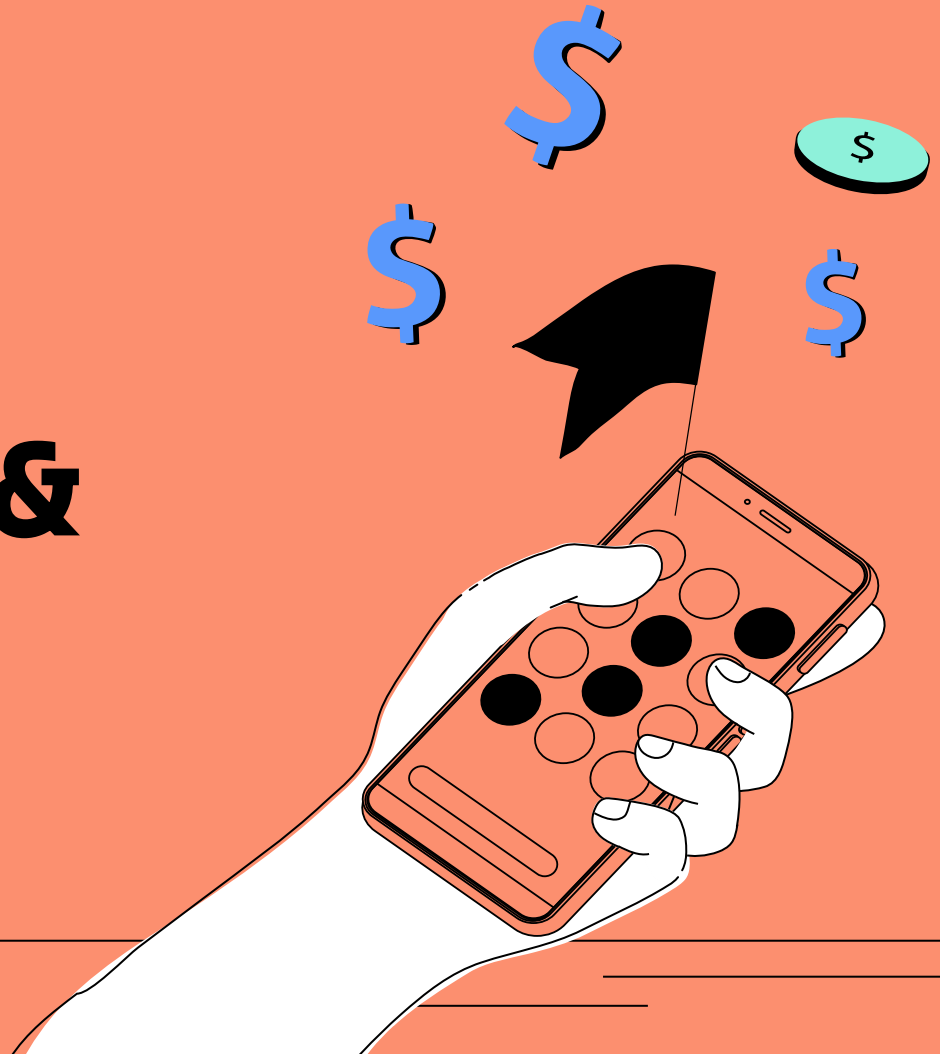
Project Management

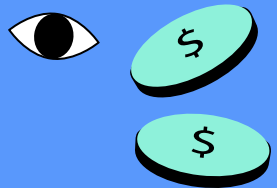
- Made goals and deadlines throughout the entire project
- Established constraints before starting project
- Divided our work into teams and sections
 - Back-end and front-end
- Held weekly meetings to understand current progress
- Established tests along the way and found most efficient way to collect data
- Actively looked at the code we wrote and adapted based on newly found problems



05

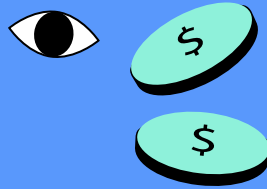
Use Cases & DEMO





Case 1: Setting Up a New Budget

- o **Goal:** User wants to set a new budget for a specific duration and amount
- o **Preconditions:** User must have an account and be logged in. User must also have at least one bank account or card connected to their account
- o **Main Flow:**
 - User selects the option to create a new budget
 - User enters the details of the budget, including category, amount, and duration/time frame
 - The system validates the information and saves the new budget to the account
 - The user receives confirmation from the system
- o **Alternative Flows:**
 - If the user enters an invalid amount (ex. Negative number, number bigger than monthly limit, etc.), the system prompts for the correct input
 - If no bank account or card is connected, the system prompts user to connect an account before budgeting
- o **Post Conditions:** The new budget is active, and tracking begins immediately for the user



Case 2: Viewing Budget Status

- o **Goal:** User wants to view the current status of their budgets
- o **Preconditions:** User has an account and is logged in
- o **Main Flow:**
 - User navigates to the budget overview section
 - The system displays all active budgets with their current status, including spent amount, remaining balance, etc.
 - User selects a budget to view more information about the budget and detailed transactions
- o **Sub Flow:**
 - Users can filter the view by budget category, duration, limit, etc.
- o **Alternative Flows:**
 - If the user has no active budgets, the system will prompt them to start budgeting so that they can view progress
- o **Post Conditions:** User gains detailed insight to their budget statuses

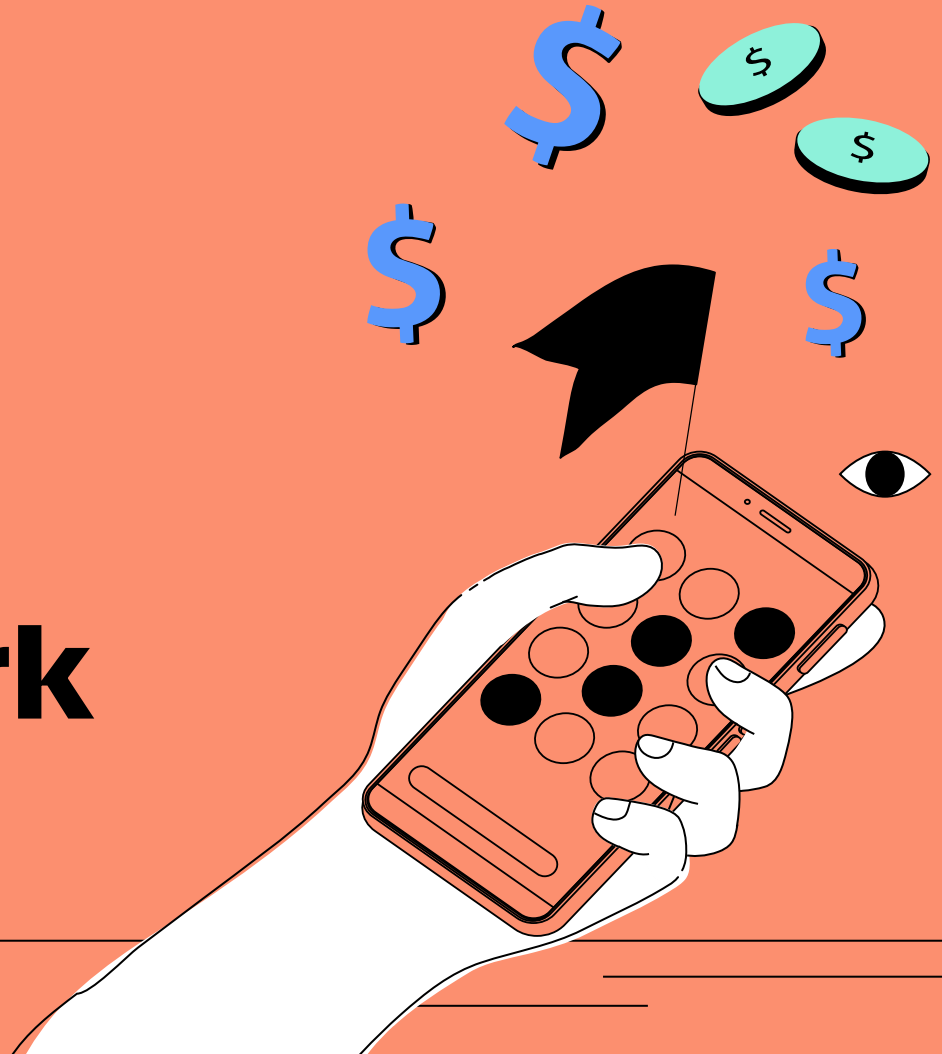


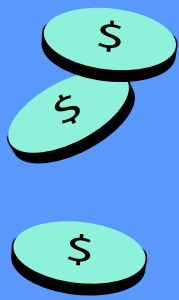
Case 3: Updating a Budget

- o **Goal:** User wants to modify the details of an existing budget
- o **Preconditions:** User must have an account, be logged in, and have an existing budget
- o **Main Flow:**
 - User selects an existing budget to update
 - User edits the budget's details such as amount, category, duration.
 - The system validates and saves the updated budget
 - The user receives a confirmation notification and their budget continue
- o **Alternative Flows:**
 - If user tries to set an invalid budget amount, the system prompts them for a valid input
- o **Post Conditions:** The budget is updated with new details and the user can continue spending within their set budget

06

Future Work

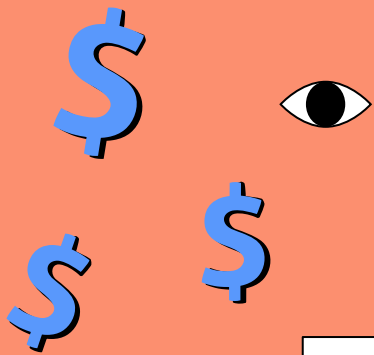




Looking Forward



1. Implement the application
2. Testing:
 - Ex: Alpha/Beta testing
 - Gain feedback on usability of the app
3. Make improvements



Thank You!

Questions?

