
SOFTWARE ENGINEERING I

Spring 2024

-Assignment #2-

Team 5:

Andrew McNeill

Krishna Sruthi Velaga

Akash Reddy Karri

Ahmed Hamza

Tobechukwu Ejike

March 18th, 2024

TABLE OF CONTENTS

S.No	Title	Page Number
1.	Section 1: Project Information.....	03
2.	Section 2: Planning and Scheduling.....	04
3.	Section 3: Problem Statement.....	06
4.	Section 4: System Requirements	
	4.1 User Requirements.....	08
	4.2 Requirements.....	12
5.	Section 5: System Modeling.....	16
6.	Appendix	18

SECTION 1

PROJECT INFORMATION

Name of the project: Budget Buddy

Semester: Spring 2024

Group Number: 5

Team members:

1. Andrew McNeill
2. Krishna Sruthi Velaga
3. Akash Reddy Karri
4. Ahmed Hamza
5. Tobechukwu Ejike

Date of Submission: March 18th, 2024

SECTION 2

PLANNING AND SCHEDULING

Assignee Name/ Email (@students.towson.edu)	Task	Duration (Hours)	Dependency	Due Date	Evaluation
Andrew McNeill / amcnei8	Allocate tasks for the report and outline use cases	1	Use Cases	03/07	100%
	Write use cases	2		03/13	
	Write use case requirements	1		03/16	
	Make final changes	1		03/17	
Krishna Sruthi Velaga/ kvelaga1	Allocate tasks for the report and outline use cases	1	Use Cases	03/07	100%
	Write use cases	2		03/13	
	Draw use case diagram	1		03/16	
	Make final changes	1		03/17	
Akash Reddy Karri/ akarri2 <i>[Coordinator]</i>	Allocate tasks for the report and outline use cases	1		03/07	100%
	Format Report	2		03/18	
	Github Collaboration	0.5		03/17	
	Make final changes	1		03/17	
Ahmed Hamza/ ahamza	Allocate tasks for the report and outline use cases	1		03/07	100%
	Github Collaboration	0.5		03/17	
	Make final changes	1		03/17	
Tobechukwu Ejike/ tejike1	Allocate tasks for the report and outline use cases	1		03/07	100%

	System Modeling Analysis	4	None	03/16	
	Make final changes	1		03/17	

SECTION 3

PROBLEM STATEMENT

The product on a high level – Budget Buddy is a web-based personal finance tool which can help users to manage their spending and budgets. It features alerts for bills and budget limits, visual spending reports, transaction tracking, and secure user authentication. It simplifies financial management and promotes informed decision-making for better financial health

Whom is it for? - It is for individuals looking to manage their personal finances, track their spending and stay within their budget. It is suitable for anyone, from students to families, who wish to understand and manage their finances better.

What problem does it solve?

Budget Buddy solves following problems:

- *Overspending*: It helps users stay within their budget limits by tracking spending and alerting them when they are close to exceeding their budget.
- *Untracked Expenses*: The application allows users to log all their transactions, providing a clear overview of where their money is going, which can help in identifying unnecessary expenses.
- *Missed Payments*: It alerts users to upcoming bills, reducing the risk of late payments and associated fees.
- *Complex Financial Tracking*: Budget Buddy simplifies the process of monitoring finances by providing visual reports, making it easier for users to understand their spending habits and financial trends.
- *Insecure Financial Data*: The application ensures that user data is secure through proper authentication measures, protecting sensitive financial information.

What alternatives are available?

Mint, YNAB, PocketGuard

- Why is this project compelling and worth developing?

The Budget Buddy project is compelling due to its innovative approach to budget planning, addressing a crucial but often neglected need for financial literacy. It introduces a feature that discerns spending patterns on a daily, monthly, quarterly, and yearly basis, enabling users to gain deeper financial insights. This analysis and personalized feedback empower individuals to effectively comprehend and control their finances. Financials being an important thing that school doesn't necessarily teach citizens, this application can help society.

- Describe the top-level objectives, differentiators, target customers, and scope of your product.

Top-level Objectives:

1. Present users with totals showing where their money is going such as food, entertainment, gas, bills, etc.
2. Provide users with an interface that's intuitive such that spending habits can be presented in graphs and diagrams.
3. Provide graphs that show past spending habits and prediction trends for future spending habits.

Differentiators:

1. This application is different than other financial planning applications because we identify and analyze spending trends over different timeframes

-
2. We provide authentication to ensure user information is secure

Target Customers:

For individuals looking to manage their personal finances. It is suitable for anyone, from students to families, who wish to understand and manage their finances better.

Scope:

The budget planner project aims to develop a user-friendly software application for individuals and organizations to manage their finances efficiently. Key features include customizable budgeting tools for expenses and income, tracking financial goals, generating reports, and ensuring data security. Users can set budgets, track spending, and monitor progress towards savings goals. The application will offer a seamless user interface across various devices and platforms, integrating with existing financial tools if necessary. Rigorous testing and ongoing maintenance will ensure reliability and performance, with user training and support provided for optimal use. Ultimately, the budget planner project seeks to empower users with the tools and insights they need to make informed financial decisions and achieve their financial objectives.

- What are the competitors and what is novel in your approach?

Mint, YNAB(You need a Budget), PocketGuard, Personal Capital are alternatives available in the market.

Buddy Budget differentiates itself through its trend analysis capabilities, where it offers insights about the spending habits of the user on a monthly, annually basis. It also provides user-friendly design.

- Make it clear that the system can be built, making good use of the available resources and technology.

Our current plan involves using Python and Java for the backend, along with HTML/CSS for the frontend

- What is interesting about this project from a technical point of view?

From a technical point of view our project is interesting because it includes machine learning to create metrics of users spending habits for past and future dates. The application will work on general devices making it openly available to a large population of users.

SECTION 4

SYSTEM REQUIREMENTS

4.1 User Requirements

Use cases:

01	Entering Expense Data
Actors:	Registered User, Budget Buddy Database
Description:	The registered user will enter the details of an expense made, which includes the amount, category (e.g., food, bills, entertainment, grocery), and the date on which the expense is made. This data will be recorded in the user data database. The system will then update the user spending data and therefore will get reflected in relevant reports and graphs.
Alternate Path:	If the registered user enters invalid date (e.g., a non-numeric value for the amount), the system prompts an alert asking the user to correct the given information
Pre-Condition:	The registered user is logged in to the system and navigated to the home screen of the application.

02	User Registration
Actors:	New User, Budget Buddy Database
Description:	The new user will be able to register into the application. The user will be asked to enter information like username, email, mobile number, password and a few others. After completion, upon successful registration the user will be navigated into the home screen of the application.
Alternate Path:	If the new user enters an email which is already in use. The application will ask to use a different email address.
Pre-Condition:	The new user doesn't have an account with budget buddy application

03	User Sign In
Actors:	Registered User, Budget Buddy Database
Description:	The registered user will enter username and password to log into the application. The system will check if the credentials are stored in the user data database. If the given credentials are valid, the user will be able to gain access to their account. If not, the user will not be given access and may be asked to re-enter the credentials.
Alternate Path:	If the user has forgotten their password, and clicks on the “forgot password” option to reset their password after confirming user identity.
Pre-Condition:	The registered user has registered account and has valid credentials

04	Resetting forgotten password
Actors:	Registered User, Budget Buddy Database
Description:	When a registered user enters invalid password, they will be asked to enter the linked email address (or) mobile number to verify the user identity and then they will be able to change the password.
Alternate Path:	If the registered user remembers the password when they are trying to reset it, they can cancel the process and return to the sign-in page to access their account with the existing password
Pre-Condition:	The registered user has registered account and has valid credentials

05	View Financial Reports
Actors:	Registered User, Budget Buddy Database
Description:	The registered user will select the date range i.e., this month, last quarter and click the toggle button to view financial reports covering the selected period. The application displays the reports based on the expenses made by the user.
Alternate Path:	If there is insufficient data for generating a report, the system prompts an alert to the registered user to continue entering the expenses made and set budget.
Pre-Condition:	The registered user has logged into the system and has sufficient transaction history for the system to generate reports

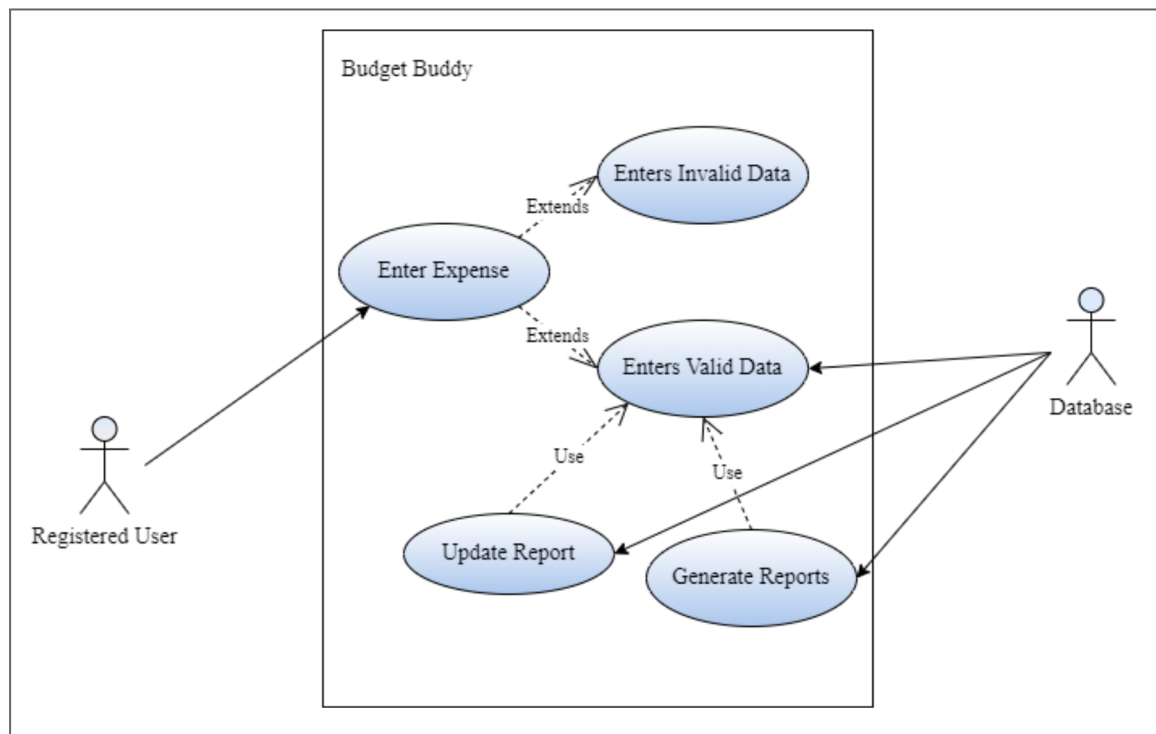
06	Edit Expense and Budget Amount
Actors:	Registered User, Budget Buddy Database
Description:	The registered user will select an existing expense or budget amount to edit. They can update the amount, category, or date before saving the changes. The application updates the information in the registered user data database and reflects this change in all reports.
Alternate Path:	If a registered user cancels the edit before saving, no changes are made to the database.
Pre-Condition:	The registered user is logged into the application and has an existing expense or budget amount.

07	Delete Expense
Actors:	Registered User, Budget Buddy Database
Description:	The user selects a transaction to remove, confirms the deletion, and the application then removes the transaction from the user data database
Alternate Path:	The user may decide to keep the transaction, so cancels the deletion process. Also, if application sometimes cannot find the transaction (it may have been already deleted or never existed), and alerts the user same
Pre-Condition:	The registered user is logged into the application and has an existing expense or budget amount.

08	Alerts and Notifications
Actors:	Registered User, Budget Buddy Database
Description:	The application monitors the user's spending against the budget limit. If the user is nearing the limit (or) exceeds the limit, the application will send an alert or notification to the user. This will help the user adjust their spending habits.
Alternate Path:	The registered user can see budget status in the home screen of the application
Pre-Condition:	The registered user is logged in, has set a budget limit and also has entered expense data.

09	Set Budget Limits
Actors:	Registered User, Budget Buddy Database
Description:	The registered user can set a budget limit for different spending categories (e.g., groceries, entertainment, bills) within the application. The application will track them and send alerts as they near (or) exceed them.
Alternate Path:	The registered user enters an unrealistic budget limit (e.g. zero dollars). The application prompt an alert asking the user to give realistic budget limit
Pre-Condition:	The registered user must be logged in.

Use Case Diagram:



The use case “*Entering Expense Data*” is fundamental for the application. It is required that the user needs to give accurate and up-to-date information for proper management of expenses which enables the application to provide insightful reports and also send alerts to users.

4.2 Requirements

Use Case 01	Entering Expense Data
Introduction:	The user enters details of an expense, including the amount, category (e.g., food, bills, entertainment), and the date of the expense. Budget Buddy records this data in the Stored User Data Database. The system then updates the user's current spending totals and reflects this in the relevant reports and graphs.
Inputs:	Decimals Values, allowing to show dollar and cent values;
Requirements Description:	The user needs to enter only decimal values so Budget Buddy can gather data accurately to process the data to become information through charts and spending predictions.
Outputs:	The entered expense value is stored in the Budget Buddy database.

Use Case 02	User Registration
Introduction:	The user enters their username and password to log into Budget Buddy. The system verifies the credentials against the Stored User Data Database. If the credentials are correct, the user gains access to their account. Otherwise, the system denies access and may offer the user the chance to reset their password or try again.
Inputs:	Multiple textboxes where the user enters <ol style="list-style-type: none">1. First Name2. Last Name3. Username4. Email5. Phone Number
Requirements Description:	The textboxes can only have alphanumeric characters and symbols entered for the Budget Buddy database system can successfully store the information.
Outputs:	The user's computer screen outputs valid login credentials to enter the website to authenticate to user the Budget Buddy services and features.

Use Case 03	User Sign In
Introduction:	The user enters details of an expense, including the amount, category (e.g., food, bills, entertainment), and the date of the expense. Budget Buddy records this data in the Stored User Data Database. The system then updates the user's current spending totals and reflects this in the relevant reports and graphs.
Inputs:	Decimals Values, allowing to show dollar and cent values;
Requirements Description:	The user needs to enter only decimal values so Budget Buddy can gather data accurately to process the data to become information through charts and spending predictions.
Outputs:	The entered expense value is stored in the Budget Buddy database.

Use Case 04	Resetting forgotten password
Introduction:	When a registered user enters invalid password, they will be asked to enter the linked email address (or) mobile number to verify the user identity and then they will be able to change the password.
Inputs:	Selects the text label “Forgot Password”
Requirements Description:	The user needs to enter a valid email address.
Outputs:	The valid entered email address is sent an email with directions for the process of password changing.

Use Case 05	View Financial Reports
Introduction:	The user selects a date range (e.g., this month, last quarter) and clicks the toggle button to view financial reports covering that period. Budget Buddy generates and displays the report, which includes graphs and summaries of expenses, budget adherence, and other relevant financial metrics.
Inputs:	Users select from a dropdown menu between daily, weekly, monthly, quarterly, and annually.
Requirements Description:	The user must select from the dropdown menu with the time according to the time frame.
Outputs:	Financial reports, graphs, and a summary of expenses are generated for display.

Use Case 06 Edit Expense and Budget Amount	
Introduction:	The user selects an existing expense or budget amount to edit. They can then update the amount, category, or date before saving the changes. Budget Buddy updates the information in the database and reflects these changes in all affected reports and analyses.
Inputs:	Inputs: <ol style="list-style-type: none"> 1. Enter a selected date. 2. Find the budget amount of the selected existing expense or budget amount to edit. 3. Select the budget expense and edit the information that needs to be edited. 4. Click the "update" button to save the changes made.
Requirements Description:	The user must select from the dropdown menu with the time according to the time frame. Once found, the input fields of the expense that's to be edited need textboxes to store the new data.
Outputs:	The entered expense value is stored in the Budget Buddy database for the selected date range.

Use Case 07 Delete Expense	
Introduction:	The user identifies an existing expense or budget amount to delete from the machine learning algorithm in the Budget Buddy software. Budget Buddy updates the information in the database to reflect the delete expense/budget amount.
Inputs:	Inputs: <ol style="list-style-type: none"> 1. Enter a selected date. 2. Find the budget amount of the selected existing expense or budget amount to delete 3. Select the budget expense and edit the information that needs to be delete. 4. Click the "delete" button to save the changes made.
Requirements Description:	The user must have at least one entered expense budget data.
Outputs:	The selected expense/budget value is deleted in the Budget Buddy database.

Use Case 08	Alerts and Notifications
Introduction:	Budget Buddy monitors the user's spending against their set budget. If the user is nearing, has exceeded, or is significantly under their budget, the system sends an alert or notification to the user. This helps users adjust their spending habits accordingly.
Inputs:	The user needs to enter only decimal values so Budget Buddy can gather data accurately to process the data to become information through charts and spending predictions.
Requirements Description:	The user must have at least one entered expense budget data.
Outputs:	The user is notified on their cell phone whether he is under-spending, overspending, credit limit reached.

Use Case 09	Set Budget Limits
Introduction:	The registered user can set a budget limit for different spending categories (e.g., groceries, entertainment, bills) within the application. The application will track them and send alerts as they near (or) exceed them.
Inputs:	Inputs <ol style="list-style-type: none">1. Select the spending category of interest.2. Users must enter a decimal value in the textbox.3. Click "save" to have the budget implemented and saved in the Budget Buddy database system.
Requirements Description:	The user must enter a decimal value.
Outputs:	The database system stores the budget and on the application within that category the budget will be shown to the user.

SECTION 5

SYSTEM MODELING

OBJECT MODELING

1. **User**: Represents the individuals who use the budget planner.

Attributes: UserID, Username, Password, Email.

- Budget**: Represents the budget plans created by users.

Attributes: BudgetID, UserID, BudgetName, StartDate, EndDate, TotalAmount.

3. **Category**: Represents the categories or types of expenses.

Attributes: CategoryID, CategoryName.

4. **Expense**: Represents individual expenses within a budget.

Attributes: ExpenseID, BudgetID, CategoryID, ExpenseName, Amount, Date, Description.

Defining the associations between the objects

- User has a one-to-many association with Budget (one user can have multiple budgets).
- Budget has a one-to-many association with Expense (one budget can have multiple expenses).
- Category is associated with Expense in a many-to-one relationship (multiple expenses can belong to one category).

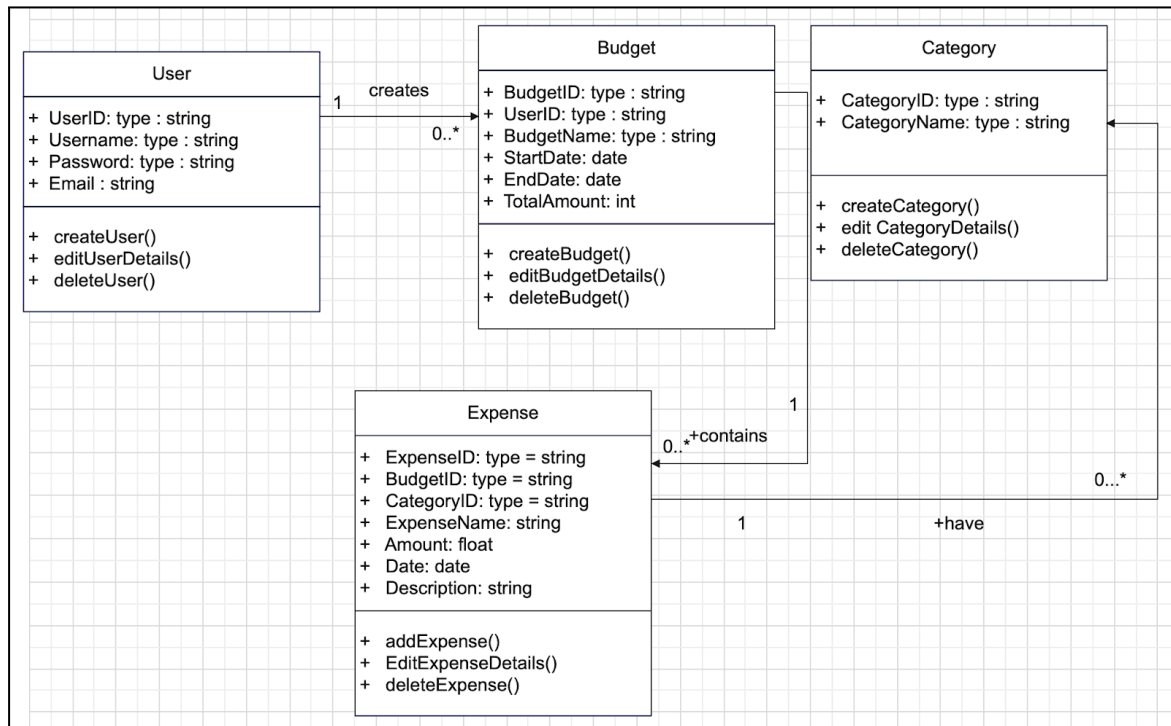
Defining the multiplicity of these associations between the objects

- User to Budget: 1 to many (one user can have many budgets).
- Budget to Expense: 1 to many (one budget can have many expenses).
- Category to Expense: 1 to many (one category can have many expenses).

Defining the operations that are used for the objects

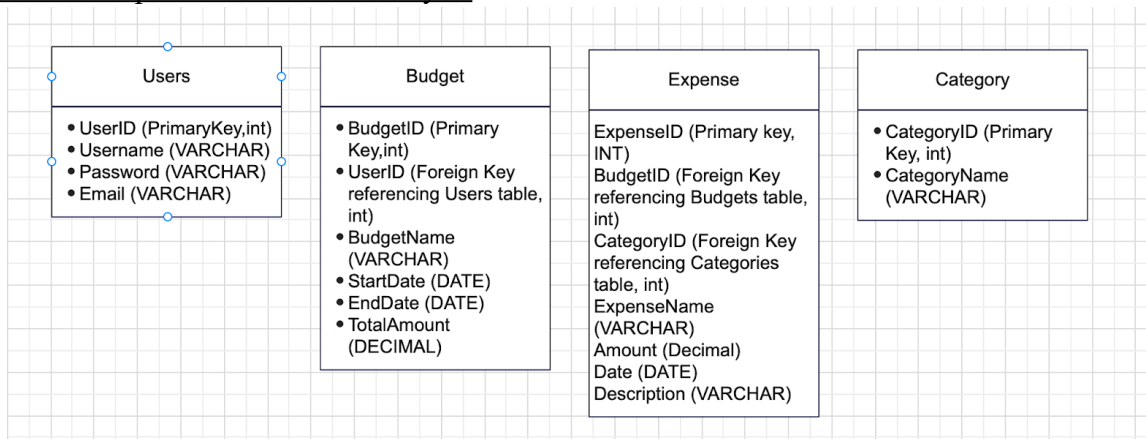
- User: createUser(), editUserDetails(), deleteUser().
- Budget: createBudget(), editBudgetDetails(), deleteBudget().
- Category: createCategory(), editCategoryDetails(), deleteCategory().
- Expense: addExpense(), editExpenseDetails(), deleteExpense().

Class Diagram representing the objects



The class diagram represents the visual aspects of the budget buddy's system structure

.Database Specifications and Analysis



This diagram shows the tables and relationships that will form the backbone of the budget buddy's system database.

MySQL is the right database management system to choose from based on the team's preferences and requirements.

APPENDIX

Github link: <https://github.com/Akash8931/SWE-FlexibleDesign.git>

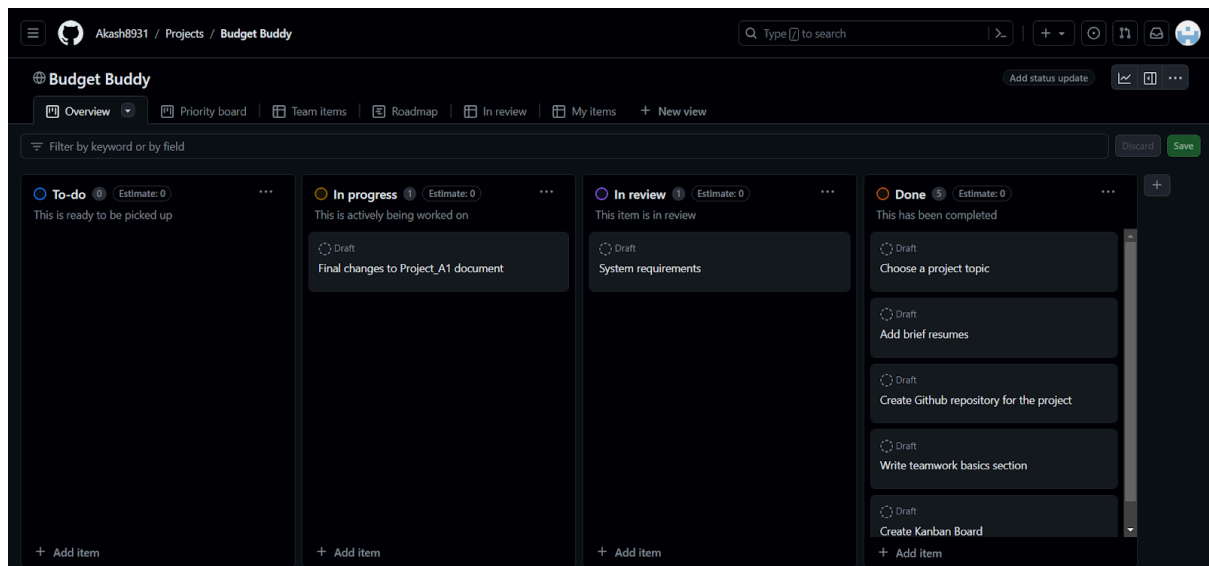


Fig: Screenshot of Kanban Board of last report

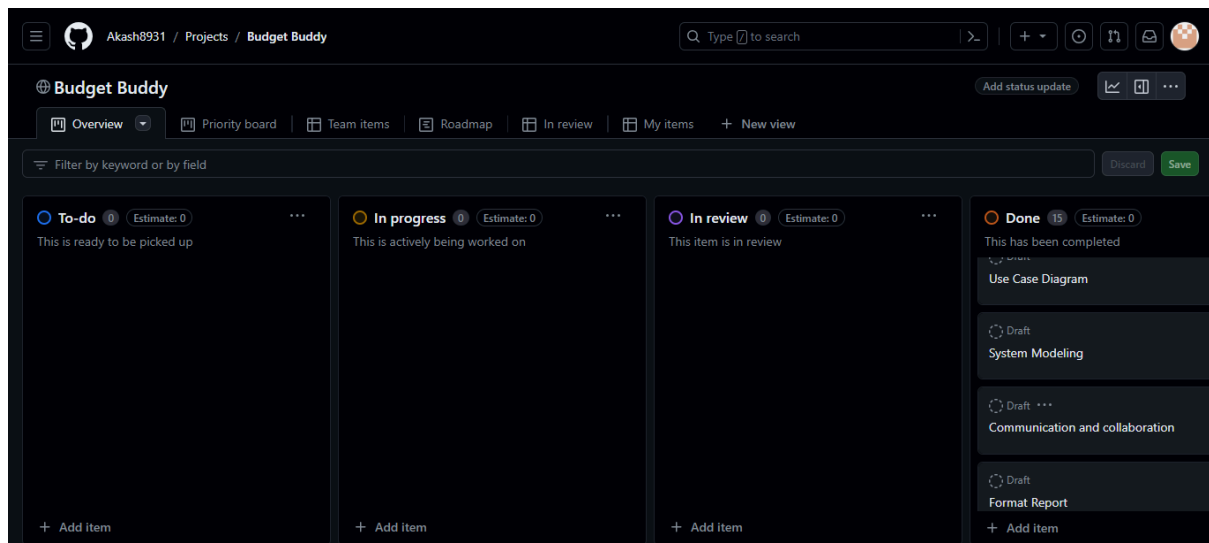


Fig: Screenshot of Kanban Board for A2