2025

Aaliyah Allie: ST10212542

Aadam Naidoo: ST10280755

Amaan Williams: ST10396397

Kenneth Michael Remmitz: ST10403075

Moghammad Saadiq Jattiem: ST10303758

IIE ROSEBANK COLLEGE

OPSC6311 Research, Planning & Design



Part 1

Table of Contents

[Research Section 3](#_Toc194432110)

[Introduction 3](#_Toc194432111)

[Overview of Personal Budgeting Applications 4](#_Toc194432112)

[Spendee 4](#_Toc194432113)

[Screenshots of Application 4](#_Toc194432114)

[Description of all main screens 4](#_Toc194432115)

[Strengths and Weaknesses 6](#_Toc194432116)

[Innovative Features of Spendee: 6](#_Toc194432117)

[Monefy 7](#_Toc194432118)

[Screenshots of Application 7](#_Toc194432119)

[Description of main screens 7](#_Toc194432120)

[Strengths and Weaknesses 8](#_Toc194432121)

[Innovative Features of Monefy: 8](#_Toc194432122)

[Wallet by BudgetBakers 9](#_Toc194432123)

[Screenshots of Application 9](#_Toc194432124)

[Description of main screens 9](#_Toc194432125)

[Strengths and Weaknesses 11](#_Toc194432126)

[Innovative Features of Wallet by BudgetBakers: 12](#_Toc194432127)

[Best features we’d like to include in our application 13](#_Toc194432128)

[Conclusion 15](#_Toc194432129)

[References 16](#_Toc194432130)

[Planning and Design Section 17](#_Toc194432131)

[Introduction 17](#_Toc194432132)

[Overview of app 18](#_Toc194432133)

[App Overview: Budget Madness App 18](#_Toc194432134)

[Purpose of application 18](#_Toc194432135)

[Logo of application 19](#_Toc194432136)

[Scope and Features 19](#_Toc194432137)

[Target Audience 19](#_Toc194432138)

[Background 20](#_Toc194432139)

[Stakeholders 20](#_Toc194432140)

[Project Timeline and Milestones 20](#_Toc194432141)

[Milestones and deliverables 21](#_Toc194432142)

[Resources 22](#_Toc194432143)

[Risks and Challenges 24](#_Toc194432144)

[Detailed list of requirements 26](#_Toc194432145)

[Minimum Requirements 26](#_Toc194432146)

[Added Requirements 31](#_Toc194432147)

[Functional Requirements 32](#_Toc194432148)

[Non-functional requirements 32](#_Toc194432149)

[User Interface 33](#_Toc194432150)

[Design Mock-ups 33](#_Toc194432151)

[Description of screens 34](#_Toc194432152)

[App Flow Diagram 36](#_Toc194432153)

[Project plan 37](#_Toc194432154)

[Conclusion 46](#_Toc194432155)

[References 47](#_Toc194432156)

# Research Section

## Introduction

This report represents the research conducted by the group "Codes" on three of the most popular personal budgeting applications. The applications chosen for analysis were **Spendee, Monefy, and Wallet by BudgetBakers**. These apps were selected based on their prominence in searches for the top free personal budgeting applications available in South Africa. The group aimed to explore and understand these apps in more detail.

The report covered an overview of the applications, their functionalities, strengths and weaknesses, innovative features, physical appearance, and key differences. Additionally, it included a list of their best features and how those features served as inspiration.

The purpose of the report was to analyze the functionalities of these applications, assess their strengths and weaknesses, and examine the unique features that set them apart. Furthermore, the group intended to use the findings from this research to develop their own application, designed to assist their target audience with personal budgeting.

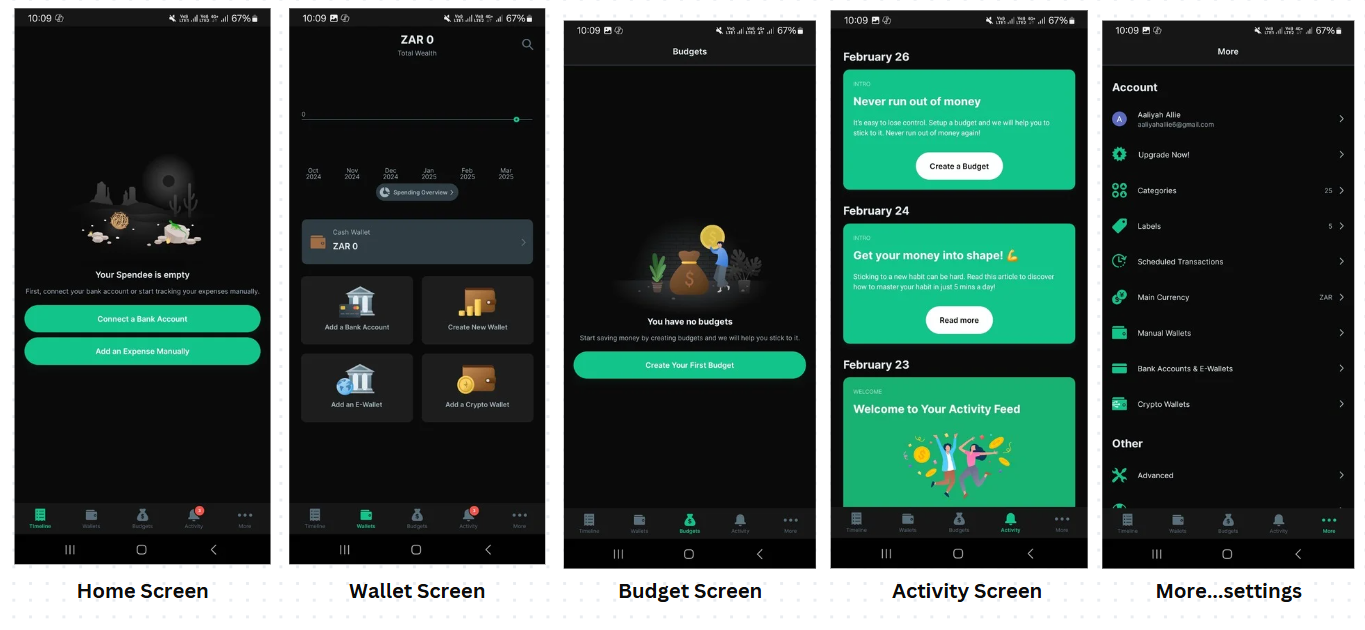
## Overview of Personal Budgeting Applications

### Spendee

Spendee was described as a personal finance application that helped users track their expenses, manage a budget, and analyze their spending habits. The app allowed users to input transactions manually or link their bank accounts for automatic tracking. It categorized expenses into groups such as transport and groceries and provided budgeting tools that issued alerts when users approached their spending limits. (Money, 2025)

Additionally, Spendee supported a shared expense feature, enabling users to track finances within family or group accounts. The app also generated charts and reports to help users assess their financial health. Moreover, it accommodated multiple currencies, making it useful for travelers managing expenses across different countries.

#### Screenshots of Application

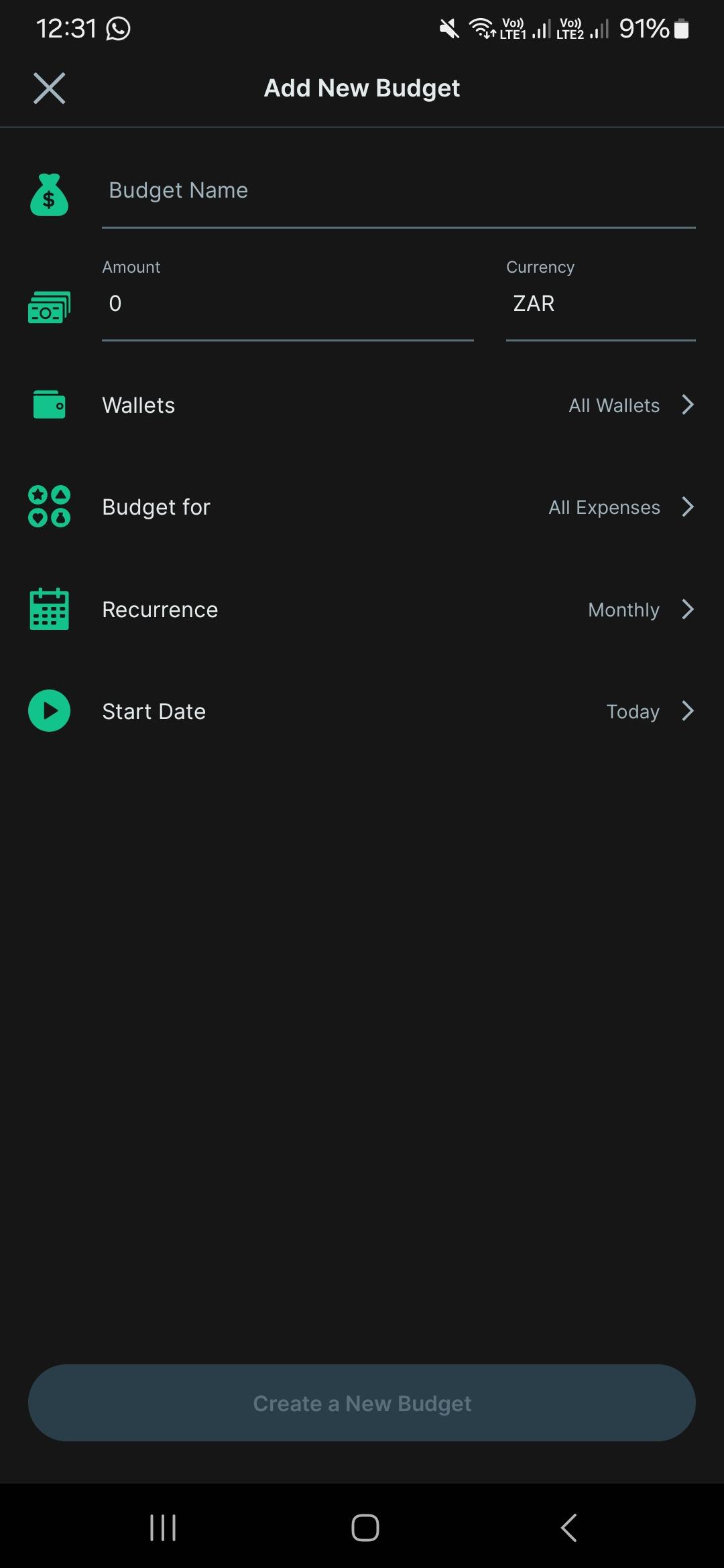


#### Description of all main screens

Home Screen: Home screen also seen as “Timeline” prompts a user to connect their bank account to the application, alternatively if a user prompts out of connecting their bank account, they can connect expenses manually to the “Timeline” screen.

Wallet Screen: The second screen is called the “ Wallet”, this screen starts by displaying a Timeline of the months starting in October 2024, leading into March 2025. Then it allows a user to view their spending overview (summary of their expenses over the months). Below that users can enter their “Cash Wallet” balance ( this balance is increased and decreased based on income and expenses). There are then displays for different features that allow users to carry out different tasks, such as adding a bank account, if they didn’t do that when the application first ran, they can create a new wallet, add an e-wallet (like paypal), or add a crypto-wallet.

Budget Screen: The next screen is called the “Budget” Screen, this screen allows users to add their budget. When a user clicks the button displayed on this screen they are taken to a new screen, “Add a new budget”, this allows users to enter budget details. In this screen it displays input areas for “budget name”, “Amount”, “Currency” (in this screen currency is set to ZAR), “Wallets” (Allowing user to add the budget to a certain wallet”, “Budget for” (allows user to link budget to a certain expense category), “Recurrence” (Allows user to set budget to reoccur, daily, weekly, biweekly, monthly or yearly), “Start Date” (Allows user to set a timeline for the budget”.



Activity Screen: The next screen, “Activity Screen”, allows a user to see their activities if they have created ones already, if not it will display prompts to allow users to add budgets and information to their timeline.

More…settings: More or settings, allows users to view their account details, upgrade their applications subscription plan (to access more features), view categories, view labels, schedule transactions ( if connected to their bank account), change the currency they are defaulted to, view their wallets and banked accounts, and there are other settings such as advanced, where they can edit the applications displays.

#### Strengths and Weaknesses

|  |  |
| --- | --- |
| Strengths | Weaknesses |
| Simple Interface: Spendee has been praised for its easy and user-friendly interface, which enables one to easily budget and track one's spending. | Subscription Cost: The added benefits like bank synchronisation and analytic reporting require paid subscription, and it may not be liked by all. |
| Bank Integration: Spendee enables users to connect their bank accounts, which allows for automatic imports of transactions as well as real-time summaries of finances. | Syncing Problems: There have been rare complaints regarding bank syncing problems, leading to differences in tracking finances. |
| Custom Categories: Users can also create custom categories based on personalized expense tracking on individual spending tendencies. | Free Version Limitation: The free version has fewer features, which might need upgrading for full financial management |
| Visual Insights: There are visual graphs and charts supported by Spendee, allowing it to be intuitive for users in understanding their spendings and income. | Unreliable Customer Support: There are complaints of inconsistent responses and efficacies from Spendee's customer support. |

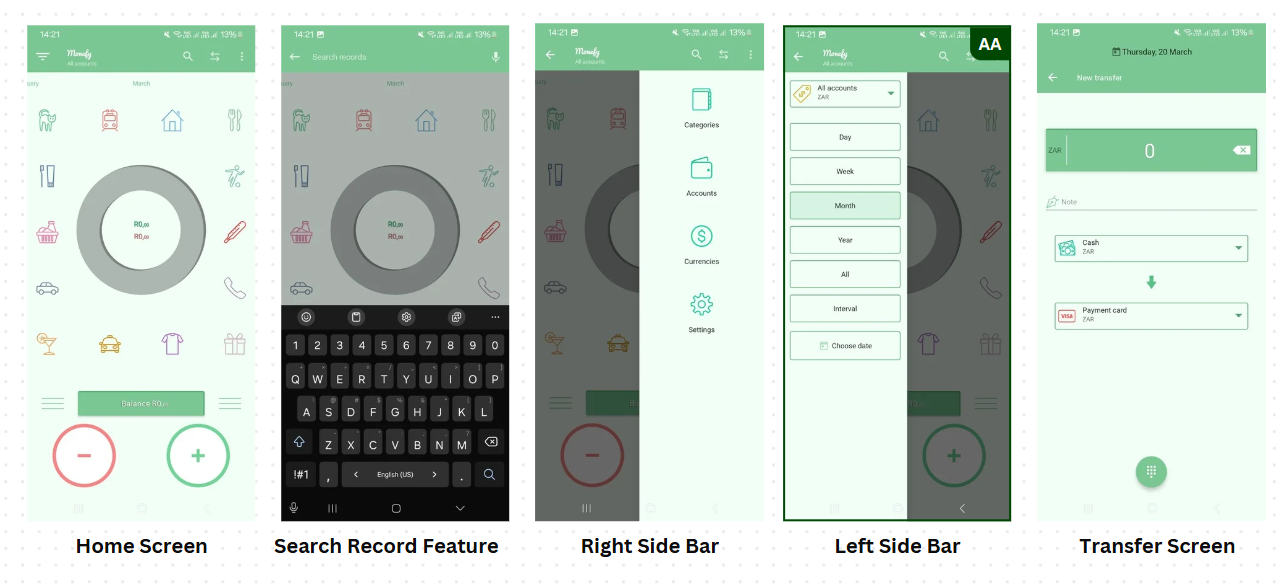
#### Innovative Features of Spendee:

* Spendee automatically synchronized user data across its mobile application and website.
* Its shared wallet allowed multiple users to track who paid for what and the total amount spent.
* The budgeting tool notified users when they had spent 75% and 90% of their total budget.
* The "All Wallet" feature provided an overview of income and expenses across all user accounts.

### Monefy

Monefy was described as a simple and intuitive expense-tracking application that enabled users to enter expenses, organize their spending, and view budget insights through clear graphs and charts. The app prioritized simplicity and a clean user interface, making it easy to navigate. (Monefy, 2025)

#### Screenshots of Application

****

#### Description of main screens

Home Screen : In Monefy, the home screen is very creative, using icons that depict their name of the expenses organized in a circular display, the income and expenses can be added at the bottom by adding them into either the “ - “ or “ +” then it will display the balance overall.

Search Record Feature : The next screen allows users to search their entered expenses by searching for it in the search icon.

Right Sidebar: This feature displays the categories icon (allows users to see all categories by name and enter an expense into it), Accounts icon ( allows users to add an amount into their cash wallet or card), Currencies (allows users to change currencies based on their location (advanced feature, needs to be paid for)), Settings (most settings are locked for an advanced subscription)

Left sidebar: This feature allows users to change the expense timeline to daily, weekly, monthly, yearly or all, interval, this allows them to see their expenses for that specific category. Or users can choose a date of an expense.

Transfer screen: The transfer screen allows users to convert their balance from card to cash or cash to card by transferring the amount manually.

#### Strengths and Weaknesses

|  |  |
| --- | --- |
| **Strengths** | **Weaknesses** |
| Ease of Use: Monefy has also been appreciated for ease of use and easy interface, making it handy for users who prefer straightforward expense tracking. | Synchronization Issues: There have been issues of data synchronization among devices by some users, leading to potential inconsistencies. |
| Customizable Categories: The app offers users customizable expense categories, enhancing individualized budgeting experiences. | Limited Features in Free Version: Only basic functionality is available in free versions, while premium versions are necessary for advanced features. |
| Customizable Categories: The app offers users customizable expense categories, enhancing individualized budgeting experiences. | Ads and Paywalls: Display of ads and partial access to certain features in the free version can be a hindrance to the user experience. |
|  | Data Reset Issues: There have been reported cases of data being randomly deleted or reset, which has hindered continuity tracking by users. |

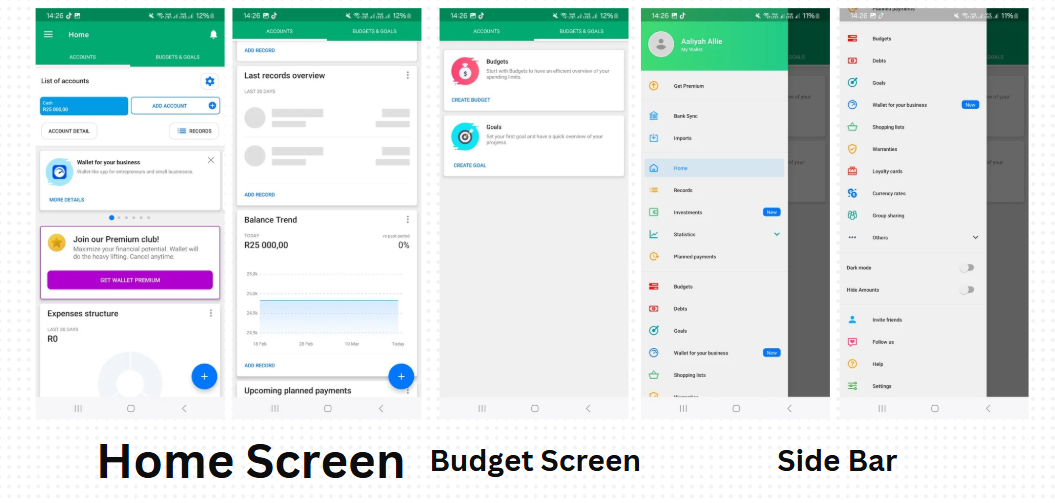
#### Innovative Features of Monefy:

* Monefy categorized expenses into understandable sections and identified spending patterns.
* Users could quickly add transactions by tapping either the "+" or "-" icons or selecting a specific category to modify.
* The app allowed customization of categories, making the experience more personalized.

### Wallet by BudgetBakers

Wallet by BudgetBakers was described as a more comprehensive personal finance management tool that included bank synchronization, bill tracking, and financial report generation. It enabled users to set budgets, forecast future expenses, and manage multiple financial accounts from a single platform. (Finder, n.d.)

#### Screenshots of Application

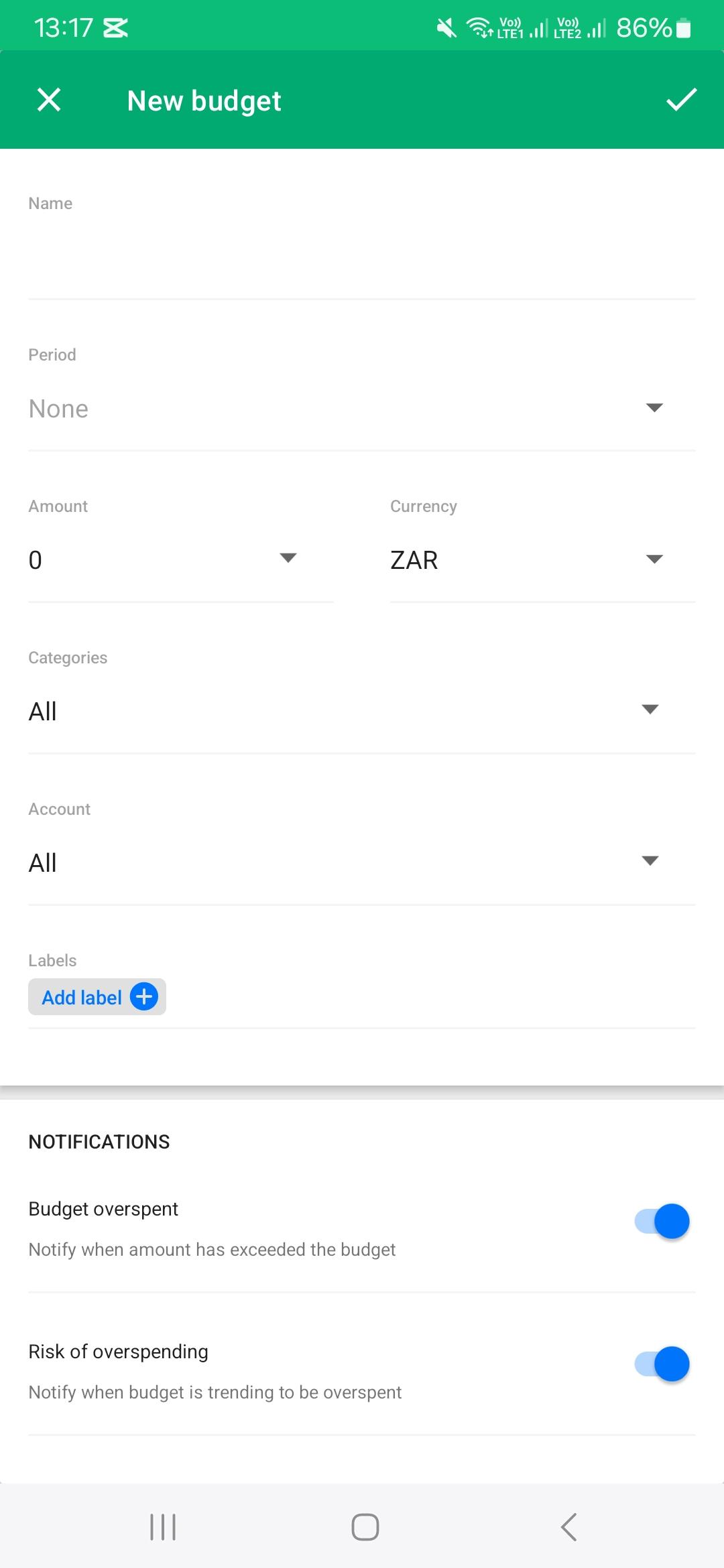
****

#### Description of main screens

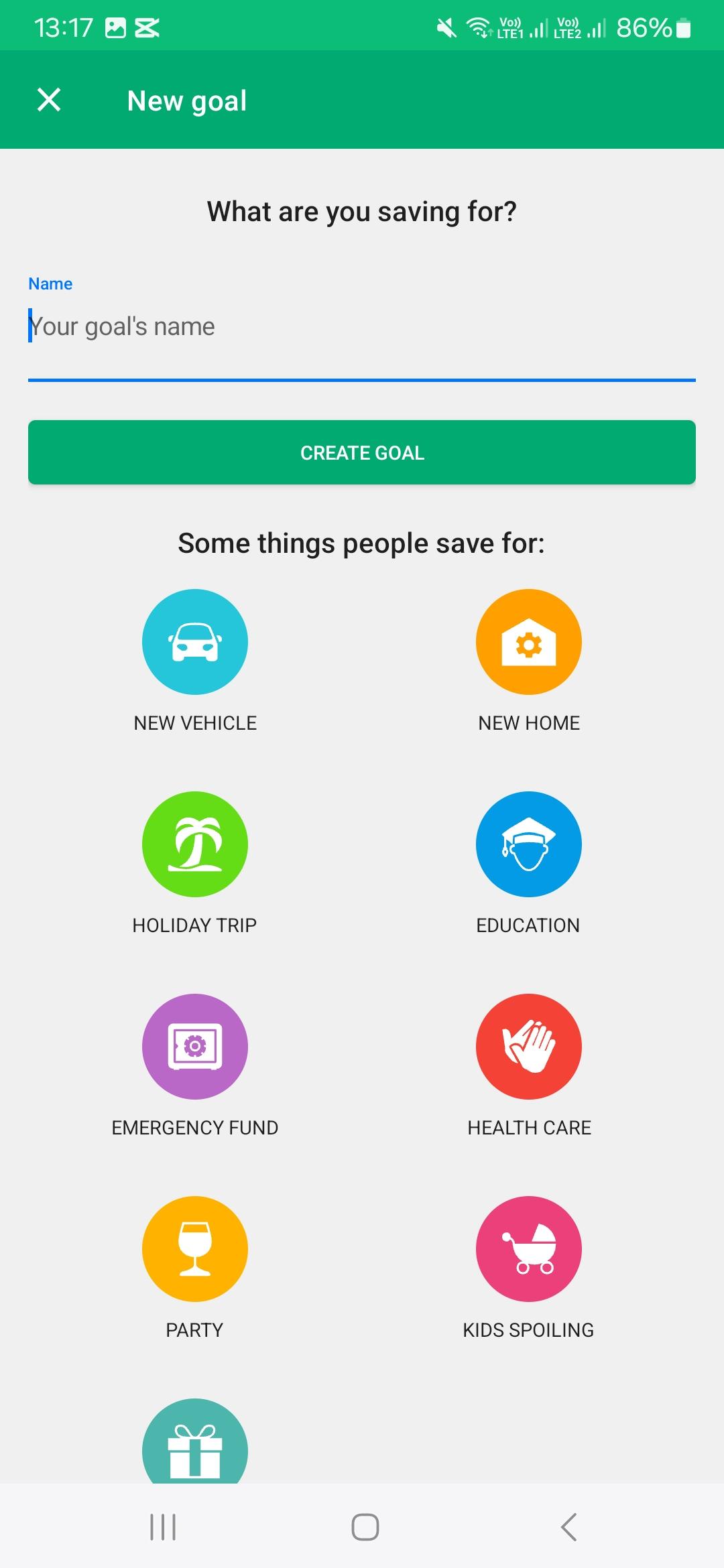
Home Screen: The home screen for Wallet, is divided into two tabs, “ACCOUNTS” and “BUDGETS & GOALS”. Under the accounts tab, the user can enter their current balance in cash and they can connect to a bank account, they can also view their account details and records of expenses/transactions. Further on there are advertisements for premium membership to the application and wallet for business. The users also get the opportunity to view a summary of their expenses displayed in a pie chart or add an expense. The user can also view their balance trend under this tab or any upcoming payments they have scheduled.

Budget Screen:

The other tab “BUDGET & GOALS” has a section to add a budget, which takes them to another screen where they can enter their details for their budget similar to spendee.



Goals always the user to enter a new goal with a name, and select a goal categories, like a vehicle or a house.



Sidebar: The sidebar allows users to change their plan (subscription) , view their account, and change their settings, as well as view their balance and expenses

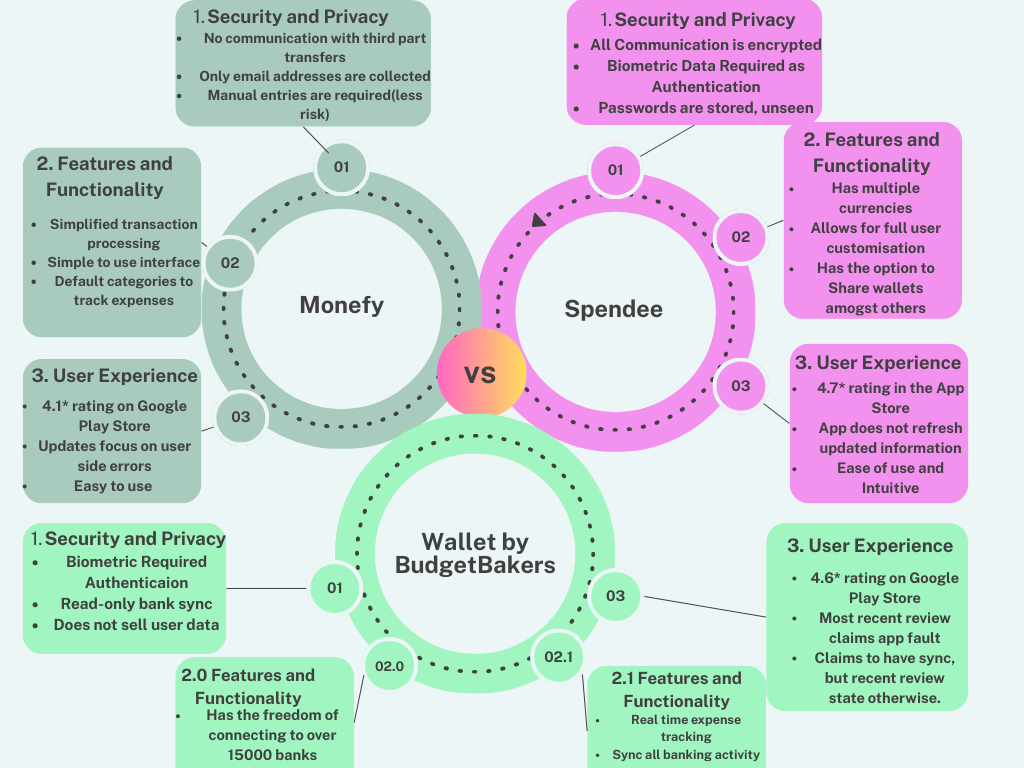
#### Strengths and Weaknesses

|  |  |
| --- | --- |
| **Strengths** | **Weaknesses** |
| Detailed Financial Management: Wallet provides an entire suite of features for tracking expenses, budgeting, and account management and is a total financial management tool. | Subscription Cost: While there is a free option, most sophisticated features are only available with a subscription, which may not suit a user in search of a free solution. |
| Synchronization with Bank: Wallet is able to connect with over 3,500 banks across the world, supporting automatic importation of transactions and current financial information. | Steep Learning Curve: The numerous features and customizable parameters can be demotivating to new users, requiring time to master and effectively use. |
| Automated Expense Categorization: The app automatically categorizes expenses, reducing manual effort and providing easily understandable spending reports. | Limited Bank Coverage: Not every bank is covered at the moment, which may limit the usage for some users whose banks are not included. |
|  | Sync Issues: There were issues with infrequent syncing issues between devices, possibly leading to tracking inconsistencies. |

#### Innovative Features of Wallet by BudgetBakers:

* The app provided features to track upcoming payments.
* Users could analyze their cash flow and identify areas for improved spending.

Comparison of budgeting applications



## Best features we’d like to include in our application

Based on our research on spendee, Monefy and Wallet by BudgetBakers these are the best features to add in our personal budgeting application:

1. Expense tracking and categorization (Spendee, Monefy and Wallet by BudgetBakers)

* Manual and automatic expense entry it allows users to manually enter expenses or to sync bank accounts for automatic tracking
* Customizable categories allow for users to categorize spending categories for example transport, groceries etc

(Softonic, 2025)

1. Budget management and alerts(Spendee and Wallet)

* For users to get notifications when they reach there 75% to 90% of their budget to alert them to maintain mindful spending
* Multiple budget categories it allows for users to set different budgets for different categories

1. Bill tracking and payment reminders(wallet)

* Upcoming payment alerts which will notify users about due bills to avoid late payment
* Recurring expenses allows that automatic logging of monthly bills example rent, subscriptions or loan payments

1. Cloud syncing and data backup (Spendee, wallet)

* Cross platform access which allows users to sync data between web and mobile applications
* Secure cloud backup which will prevent loss of any data

((Medium), 2025)

## Conclusion

Spendee boasts a user-friendly interface with custom categories and visual reports but can require a subscription for full features and has had syncing issues. Monefy is appreciated for its simplicity and reliability but is restricted in its free version and potential data synchronisation problems.Wallet by BudgetBakers provides built-in financial management capabilities with complete bank synchronisation but may have a learning curve and requires a subscription to access all features.All apps have their strengths and potential weaknesses. Users must consider their own financial management needs and preferences when choosing the most suitable app. The "Codes" team thoroughly examined the features, benefits, drawbacks, and unique qualities of three popular personal budgeting apps: Spendee, Monefy, and Wallet by BudgetBakers. This study pinpointed essential components for successful budgeting, including expense monitoring, budget alerts, bank integration, and flexible categories. The results provide a basis for creating their own budgeting app, incorporating the top features of these applications to design a user-friendly and effective tool that improves financial monitoring and expense control for their intended users.

The study shed important information on how various budgeting apps satisfy users' expectations for financial management, emphasizing the value of customization, automation, and user-friendliness in raising user satisfaction. The team “Codes” intends to use these findings to create an app that not only makes tracking expenses easier but also includes clever features that adjust to users' spending patterns. This strategy will guarantee that the new program is functional and user-friendly, assisting users in better budget management.

## References

Apple App Store, n.d. *Monefy - Money Tracker*. Available at: <https://apps.apple.com/us/app/monefy-money-tracker/id1212024409> [Accessed 20 March. 2025].

BudgetBakers, n.d. *5 Ways Wallet Now Makes Financial Data Useful*. Available at: <https://budgetbakers.com/5-ways-wallet-now-makes-financial-data-useful/> [Accessed 28 March. 2025].

BudgetBakers, n.d. *What is Wallet?*. Available at: <https://budgetbakers.com/what-is-wallet/> [Accessed 20 March 2025].

**Capterra**, n.d. *Spendee Reviews*. Available at: <https://www.capterra.com/p/238829/Spendee/reviews/> [Accessed 20 March 2025].

Finder, n.d. *Wallet by BudgetBakers Review*. Available at: <https://www.finder.com/uk/budgeting/wallet-budgetbakers-review> [Accessed 20 March. 2025].

Google Play Store, n.d. *Monefy Reviews on Google Play Store*. Available at: <https://play.google.com/store/apps/details?id=com.monefy.app.lite&hl=en_ZA> [Accessed 20 March. 2025].

Incubating Wallet, n.d. *The Checkbook App: Monefy Product Review*. Available at: <https://incubatingwallet.com/the-checkbook-app-monefy-product-review/> [Accessed 20 March 2025].

Millennial Money, n.d. *Spendee Review*. Available at: <https://millennialmoney.com/spendee-review/> [Accessed 20 March 2025].

Monefy, n.d. *For info*. Available at: <https://monefy.com/> [Accessed 20 March. 2025].

Softonic, n.d. *Monefy - Expense Tracker Review*. Available at: <https://moneyfy.en.softonic.com/android#:~:text=Users%20can%20track%20their%20expenses,their%20finances%20in%20one%20place> [Accessed 28 March. 2025].

Spendee, n.d. *For info*. Available at: <https://www.spendee.com/> [Accessed 20 March 2025].

SpendeeApp (Medium), n.d. *8 Advantages of Spendee Plus*. Available at: <https://spendeeapp.medium.com/8-advantages-of-spendee-plus-9ffa57ed91b2> [Accessed 28 March. 2025].

Trustpilot, n.d. *Wallet by BudgetBakers reviews*. Available at: <https://www.trustpilot.com/review/budgetbakers.com> [Accessed 20 March 2025].

# Planning and Design Section

## Introduction

This document provides a comprehensive overview of the **Budget Madness App** project, outlining its purpose, scope, features, and development phases. The purpose of the app is to offer users a simple and engaging platform for managing personal finances through tools that track expenses, set budgets, and promote better financial habits. The app incorporates gamification elements, such as rewards and progress tracking, to enhance user engagement and retention.

The report covers the project plan of the application, defining the application's purpose, scope, target audience, background, stakeholders, project timeline and milestones as well as resources and risk management.

By the end of this document an establishment of the application will be made as well as an in detailed feature list of requirements needed to create a fully functional personal budgeting application. This document will cover the creation of the applications design, or prototype as well as the database creation of the application.

## Overview of app

### App Overview: Budget Madness App

The application we are creating is going to be a financial aid application which will help users operate their finances better. Our application is designed to make tracking and managing personal finances easier, more engaging, and less stressful. With an intuitive, user-friendly interface, the app encourages consistent use through gamification elements, rewards, and interactive features, our application will make use of innovative features that were inspired by the previous research section to allow user interaction to be more friendly and easier to transition to.

#### Purpose of application

Budget Madness aims to solve personal finance challenges by being an engaging, user-friendly platform. It will focus on tracking income and expenses, set goals and budgets, and improve financial habits through gamification of application features, rewards and innovative features. The overall objective of the application is to simplify personal budgeting, promote consistent use, and enhance user’s financial wellbeing.

#### Logo of application



#### Scope and Features

The application will allow users to register, login, track expenses through different categories, set goals for budgets, view expense lists, and access receipt books. It includes gameful thinking by making use of different interactions like rewards, badges, and progress dashboards. It stores data locally and provides a visual interaction to track transactions through graphs or pie charts. Unique features will be included through customizable categories, spending graphs, and user engagement.

#### Target Audience

The intended audience for this budgeting application is young-adults, students and individuals looking for a simpler, more engaging way to track their finances. The application will cater to users with basic to moderate technical skills and expects them to want an intuitive, fun way to stay on top of their budgets.

#### Background

This application is essential for tackling the usual challenge of effectively managing finances. Numerous current financial applications are either complicated or not engaging. The Budget Madness App seeks to address this deficiency by providing an enjoyable, straightforward method to monitor and handle personal spending, emphasizing gamification to keep users engaged.

#### Stakeholders

|  |  |  |
| --- | --- | --- |
| Stakeholder | Definition of role | Roles and responsibilities |
| Project Manager | Oversees the development process and ensures the application meets requirements and deadlines. | Coordinates tasks, set deadlines, and ensures smooth project execution. |
| Developers | Responsible for coding, implementing features, and ensuring application functionality | Implement the app’s functionality and integrate gamification elements |
| UI/UX Designers | Create the interface and user experience, ensuring it’s intuitive and engaging. | Design an aesthetically pleasing and user-friendly interface. |
| End Users | The primary stakeholders, whose feedback will be crucial to the app’s success. | Provide feedback on app usability and functionality. |
| QA Team | Conducts testing and ensures the app’s reliability. | Test the app for bugs and ensure it meets all requirements. |

#### Project Timeline and Milestones

This project is structured to be created in 6 months, the first 3 months are reserved for the research and planning phase of the application and the last 3 months are reserved for the development, testing and final submission phases of the application.

##### Milestones and deliverables

Phase 1: Research and Planning Phase

Milestone 1: Research For Application Development

Deliverables:

* Research three budgeting applications
* Take note of innovative features
* Discuss what features will be implemented into Budget Madness
* Analysis of budgeting applications

Milestone 2: Project Scope & Requirements Definition

Deliverables

* Project proposal with objective, features, and limitations
* Risk management
* Functional and Non-functional requirements identified
* UI/UX wireframes and conceptual designs discussed

Phase 2: Design & Prototype Creation

Milestone 3: UI/UX Design & Gamification Strategy

Deliverables:

* Prototype creation ( Figma)
* User flow diagram and wireframes
* Gamification elements added to prototype

Milestone 4: Database & System Architecture Design

Deliverables:

* ERD (Entity Relationship Diagram) for database structure
* System architecture and API documentation
* Backend technology stack finalized

Phase 3: Development phase

Milestone 5: Core Functionality Development

Deliverables:

* User authentication (registration, login, password reset)
* Expense tracking with categories and receipt uploads
* Budget goal setting and expense listing

Milestone 6: Advanced Features & Gamification Implementation

Deliverables:

* Dashboard with budget progress and spending breakdown
* Visual analytics (spending graphs, category trends)
* Gamification elements (badges, rewards for savings goals)

Phase 4: Testing & Refinements

Milestone 7:Unit Testing & Bug Fixing

Deliverables:

* Unit test reports for core functionalities
* Security assessment and data protection measures

Milestone 8: User Testing & Feedback Implementation

Deliverables:

* Usability testing report with key improvements
* Final refinements based on user feedback
* Performance optimization and app stability checks

Phase 5: Deployment

Milestone 9: Final Submission & Deployment

Deliverables:

* Fully functional **Budget Madness App**
* Source code pushed to GitHub with documentation
* Final presentation and demo
* Deployment to Google Play Store or Web Platform

(Świtek, 2024).

#### Resources

Hardware:

* Laptops
* Phones
* Module Manual

Software:

* Word
* Google Docs
* GanttPro
* Draw.io
* Google
* Android Studio
* Figma

### Risks and Challenges

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Name | Risk Definition | Probability | Impact | Proactive Measures |
| Technical Difficulties | Challenges related to coding, integration, or platform compatibility issues. | High | High | Early testing of platforms and frameworks.  Use proven tools for backend and frontend.  Regular code reviews and debugging. |
| User Adoption Resistance | Users may find the app difficult or unappealing to use, resulting in low engagement. | Medium | High | Conduct user interviews and gather feedback during development.  Iterative UI/UX improvements based on feedback. |
| Data Security Risks | Potential for sensitive user data (e.g., passwords, receipts) to be exposed. | Medium | Very High | Implement strong encryption for sensitive data.  Regular security audits.  Data protection best practices. |
| Scope Creep | Unplanned changes or additions to the app’s features after project initiation. | High | High | Strictly define project scope and avoid feature creep.  Use change control procedures. |
| Budget Overruns | Project costs exceeding the allocated budget due to unanticipated expenses. | Medium | Medium | Track and review costs regularly.  Contingency budget for unexpected expenses. |
| Timeline Delays | Delays in the project schedule impacting delivery or milestones. | Very High | High | Set realistic timelines with buffer periods.  Use project management tools (e.g., Gantt charts) for tracking. |
| Inadequate Testing | Insufficient testing leading to bugs or performance issues post-launch. | Medium | Very High | Implement automated testing and unit tests.  Set aside time for beta testing. |
| Integration Issues | Difficulty in integrating external APIs or services (e.g., payment systems). | Medium | High | Early integration tests with third-party services.  Have backup solutions in place for critical integrations. |
| Lack of Gamification Appeal | Gamification elements fail to engage users, reducing retention and motivation. | Medium | Medium | Conduct user testing focused on gamification features.  Adjust gamification elements based on user preferences. |
| Legal and Compliance Issues | Potential legal risks related to financial data handling or app privacy concerns. | Low | Very High | Research financial data compliance regulations (e.g., GDPR).  Implement privacy and terms of use policies. |

(indeed.com, 2025).

### Detailed list of requirements

#### Minimum Requirements

* Registration and login to the application using a username and password.
  + To create a login and registration page you will need to work in the drawable section of the res folder to design a box to house your buttons and input text boxes or you can design it in the main activity folder where a cardview can be used that will house the buttons and edit text boxes.. The way you want to design your layout is up to you. You can use the same layout for the login and registration page but each may differ because of edit text boxes and buttons.
  + This link is another guide on how to create a login page:
  + <https://androidknowledge.com/loginpage-in-android-kotlin/>

(Android Knowledge, n.d.-b)

* Category creation for expenses.
  + Depending on where you want your categories that will decide where you create them for example if i want my categories to be in our user folder i will click on the user folder and add an empty activity view. In this activity view I will add a scroll view so that the user is able to scroll when they add more categories themselves. In the scrollview you will use a linear layout so that each category is under each other. Then in the liner layout you will use a relative layout to design each category, the relative layout will contain the width, height, icon padding, vertical layout and alignment for the category. You will then need to code the buttons and icon you used in the java file of the folder where the activity was created.Each category will need to be stored on the database you decide to go with.
  + This link is a video on how to create categories in kotlin:
  + <https://www.youtube.com/watch?v=7S7646Cymn0>

(YouTube, 2023a)

* Add an expense entry in which the user must specify the amount, date, description and category.
  + In the room database a table will be created called expenses, here there will be 4 val namely id, category name and amount. In the database actions there will be interface expenseDao, suspend fun insertexpense(expense ; Expense) and @Query("SELECT SUM(amount) FROM expenses")suspend fun getTotalExpenses(): Double?

CODE EXAMPLE:

* + @Entity(tableName = "expenses")
  + data class Expense(
  + @PrimaryKey(autoGenerate = true) val id: Int = 0,
  + val categoryName: String,
  + val amount: Double
  + )
  + import androidx.room.\*
  + @Dao
  + interface ExpenseDao {
  + @Insert
  + suspend fun insertExpense(expense: Expense)
  + @Query("SELECT SUM(amount) FROM expenses")
  + suspend fun getTotalExpenses(): Double?
  + }
* Attach a photo of receipt (optional).
  + To add camera functionality to the application we need to create a file provider in the androidManifest.xml file. This file will contain a provider that houses the content name,authorities and permissions. It will also contain the meta data for the fileprovider paths.next in the res/xml/files\_path.xml folder you will add an external media path that names images and the path pictures. Next you will create a directory: /Pictures where you will create a temporary dictionary for the pictures that will link to file\_path. Now in mock mediastore.extra\_output with fileprovider.getUriforfile you input the intent mediastore.action\_image\_capture and image createtempfile. These images will be stored in the image.absolutePath.replace("/storage/emulated/0", "sdcard")<< store file name, replace "/storage/emulated/0" to "sdcard" intent.putExtra(MediaStore.EXTRA\_OUTPUT, uri)

launcher.launch(intent). In the launcher the image will be saved and a file will be created in the sdcard/android/media/pictures folder.

* + This is a link to further explain how to add camera functionality:
  + <https://lomana.medium.com/kotlin-take-a-photo-in-android-3f9d5c131688>

(Lomana, 2023)

* Be able to set a monthly total budget goal and limits for categories.
  + In the room database a table will be created called expenses, here there will be 3 val namely id, category name and amount. In the database actions there will be interface expenseDao, suspend fun insertexpense(expense ; Expense) and @Query("SELECT SUM(amount) FROM expenses")suspend fun getTotalExpenses(): Double?. Now to check expenses you have to get the total expenses from the expense table and the budget limit from the shared preferences. You will now need to create this to avoid your app from crashing
  + lifecycleScope.launch {
  + // Get total expenses from DB
  + val totalExpenses = expenseDao.getTotalExpenses() ?: 0.0
  + // Get budget limit from SharedPreferences
  + val sharedPref = getSharedPreferences("BudgetPrefs", Context.MODE\_PRIVATE)
  + val budgetLimit = sharedPref.getInt("budget\_limit", 0)
  + if (totalExpenses >= budgetLimit) {
  + // Budget limit reached or exceeded
  + runOnUiThread {
  + Toast.makeText(this@MainActivity, "⚠️ Budget Limit Reached!", Toast.LENGTH\_LONG).show()
  + }
  + } else {
  + // Optional: show how much is left
  + runOnUiThread {
  + Toast.makeText(this@MainActivity, "You have R${budgetLimit - totalExpenses} left.", Toast.LENGTH\_SHORT).show()
  + }
  + }
  + }
  + Lastly to get the notification for the user you will need to create a pop up
  + if (totalExpenses >= budgetLimit) {
  + runOnUiThread {
  + btnAddExpense.isEnabled = false
  + Toast.makeText(this@MainActivity, " No more expenses allowed. Budget limit reached.", Toast.LENGTH\_LONG).show()
  + }
  + }
* A list view must be created for all expenses during the users selected period as well as all receipts added to that category.
  + To create a list view you will need to create another empty activity, then in the app-level build gradle file you will need to add the buildfeatures viewbinding true and click sync now. Once that is synced you can go to your res folder and find colors.xml and add the colours you would like to use, next you would go to the string.xml and add the app name and the content you want to add. Now in the activity\_main.xml you will need to add ListView and give the list its width, height and scrollbars. Create a file called ListData.kt where you add all the variables. Now you will create a file called ListAdapter.kt where you will implement an arrayadapter. Now in the main file you will need to enable view binding and create arrays for variables, arraylist,listdata and setup the listadapter. Lastly open detailedActivity.kt and enable view binding and receive and populate the ui elements.
  + Extra information:
  + <https://androidknowledge.com/custom-listview-android-studio-kotlin/>

(Android Knowledge, n.d.-a)

* All data must be stored in a local database.
  + From the research I have gathered the best database to use for a budget tracker is a room database that uses SQLite. This database offers features like type- safe queries and data binding and SQLite is a lightweight embedded database engine.
  + Extra information:
  + <https://developer.android.com/kotlin/multiplatform/sqlite>
  + <https://kotlinlang.org/docs/data-analysis-connect-to-db.html>

(Kotlinlang.org, n.d.)

* Graph view for select period.
  + To create a graph we are going to need to add dependencies to the gradle one would be the implementation and the other would be repositories and then you will need to sync them. Next would be to add viewbindings and sync again. Now in the activity\_main.xml you will have to update the layout for the chart. Next would be to add the logic in the MainActivity.kt file for the array and bardataset and a barchart should form
  + Extra information
  + <https://kotlinlang.org/docs/data-analysis-visualization.html>

(Kotlinlang.org, n.d.-a)

* Budget progression dashboard.
  + To create a progression bar you need to add the progressbar to the activity\_main.xml file. Once the progressbar is added you have to give it the logic to get the total expenses and budgetlimit and give notification when certain point are reached in the bar
* Weekly and monthly badges if the budget goal is met.
  + To create a rewards badge system you will need to start with an empty activity window then in your re/layout/activity\_main.xml folder you will design the badge UI. After the UI is created you will need to add images in the res/drawable folder for the badges icons. Next you will create a badge model in Badge.kt that will be a data class and contain that badges variables. Now in the MainActivty.kt you will need to add the logic so that the badges will create and update
  + Extra information:
  + <https://forum.adalo.com/t/how-to-create-a-badge-reward-system-when-users-complete-activities/28388>

(Adalo Forum, 2023)

#### Added Requirements

* Developed on android 10
  + The reason why android 10 was selected for the development of application was because of the south african economy. If we look at South Africa's smartphone we will find that smartphones are quite expensive and people usually tend to keep their phones for 4 - 6 years instead of upgrading every 2 - 3 years. Another reason android 10 was chosen was because South Africa has a huge budget phone(R1000 - R3000) market and these phones normally ship with older versions of android so as a group android 10 was decided to make our application wider available.
* Responsive design
  + To have a responsive design you will need to use the wrap\_content, match\_parent or 0dp for the width and height of the components you choose to use
  + Extra information on how to create a responsive design:
  + <https://developer.android.com/develop/ui/views/layout/responsive-adaptive-design-with-views#:~:text=UI%20With%20ConstraintLayout.-,Responsive%20width%20and%20height,space%20within%20the%20view's%20constraints>.

(Android Developers, n.d.-a)

* Notification alert when 70% and 90% of budget is used
  + At this point you should have already created a total budget and expense now you will need to create a function to calculate percentage. Once that is complete create a function that will check the threshold to see if the budget goes over 70% and 90%. Now in MainActivty.kt you will need to add the logic for the notification for these budget alerts and to avoid getting multiple alerts for each category you need to use a shared preference.
* Quick add for transactions
  + In the activity\_main.xml you will need to add an EditText and a button. Once you added the logic for the EditText and the button you will need to open the MainActivity.kt and create a list that will hold the added data now you need to add the logic for the onCreate and add a listview to view the newly added items.

#### Functional Requirements

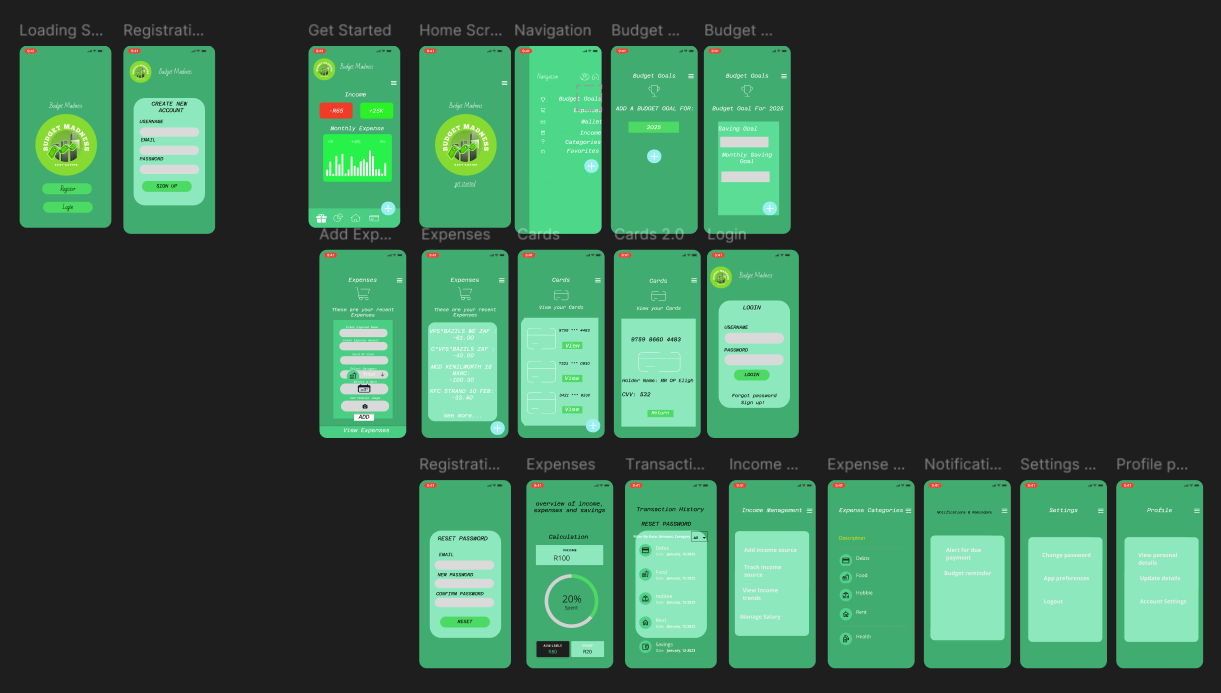
* Registration and login to the application
* Category creation for expenses.
* Notification alert when 70% and 90% of budget is used
* Quick add for transactions
* Developed on android 10
* Be able to set a monthly total budget goal

#### Non-functional requirements

* Responsive/adaptive design
* All data must be stored in a local database.
* Weekly and monthly badges if the budget goal is met.

### User Interface

#### Design Mock-ups



Screens screenshot of a phone

AI-generated content may be incorrect.

Screens screenshot of a phone

AI-generated content may be incorrect.

Screens screenshot of a phone

AI-generated content may be incorrect.

Screens screenshot of a phone

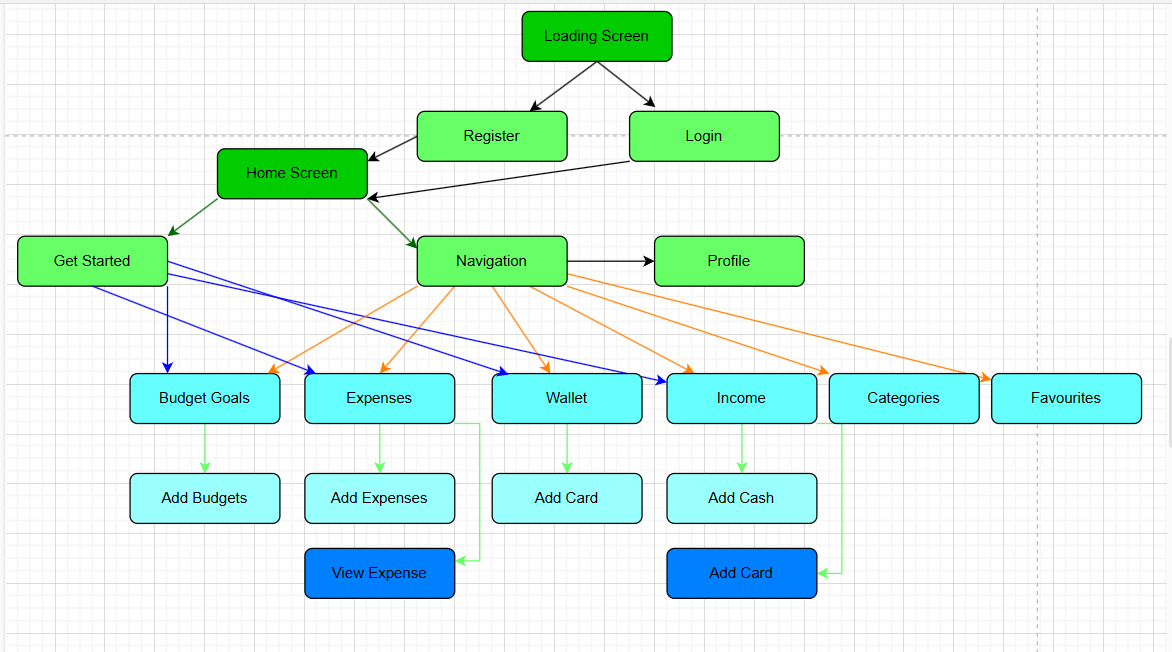
AI-generated content may be incorrect.

(Figma, n.d.-a)

#### Description of screens

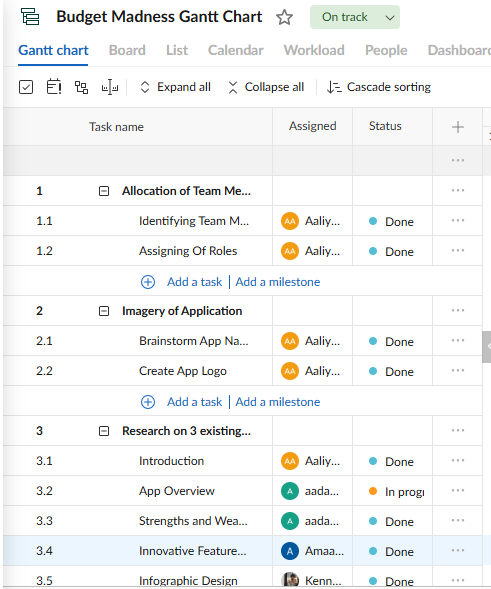
* Loading Screen: This screen is the first thing that loads when the application has just started, it has the logo, and two buttons that prompts the user to register or login.
* Registration Screen: Asks users for details, like their full name, username, password and a contact, like an email.
* Login Screen: Asks users for their username and password to login
* Home Screen: The home screen is the screen the users will view when the application opens. The screen displays the app name, logo, navigation bar, and a button called (get started).
* Get Started: Displays users balance and expenses, expense trend over the year, navigation bar, and navigation to different screens at the bottom navigation bar.
* Navigation: Allows users to navigate to their profile, settings, and other screens.
* Profile: Users can view their account details, change account details, and add a profile image.
* Budget Goals: Allows users to add a budget based on the year.
* Expenses: Allows users to view expenses for the week, month, year and so on.
* Wallet: Allows users to connect their cards.
* Income: Allows users to increase their balance based on their card balance and cash balance.
* Categories: Different categories where expenses can be sorted into.
* Favourites: Categories that are used often.
* Add Budgets: Adds a budget based on the year.
* Add Expenses: Adds an expense, based on categories, date, and receipt info.  
  View Expenses: view expenses for the week, month, year and so on
* Add Card: Allows users to add card information, and connect it to their expenses.
* Add Cash: Allows users to add cash balance to existing balance on app.

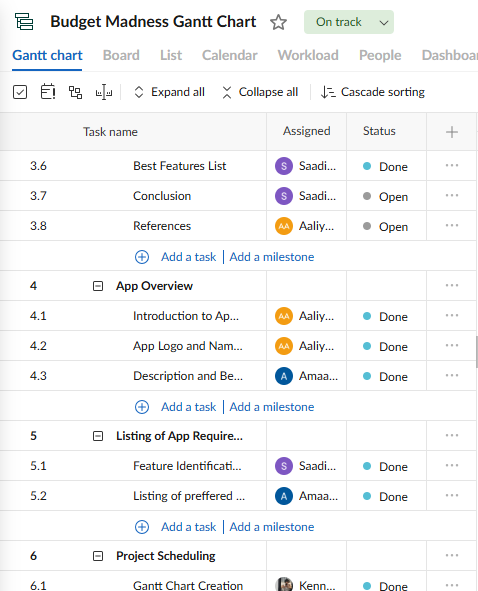
#### App Flow Diagram

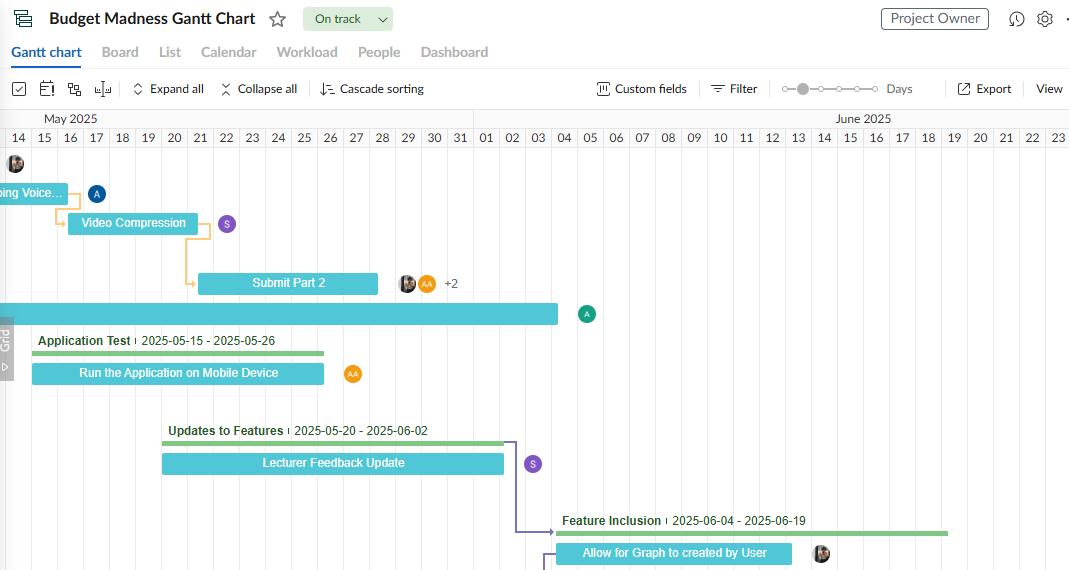
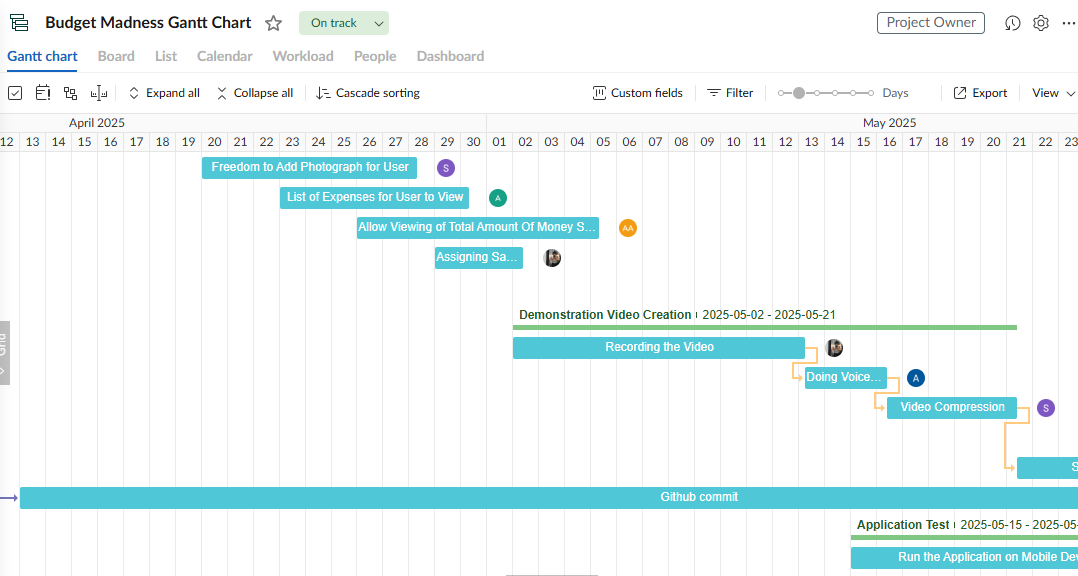
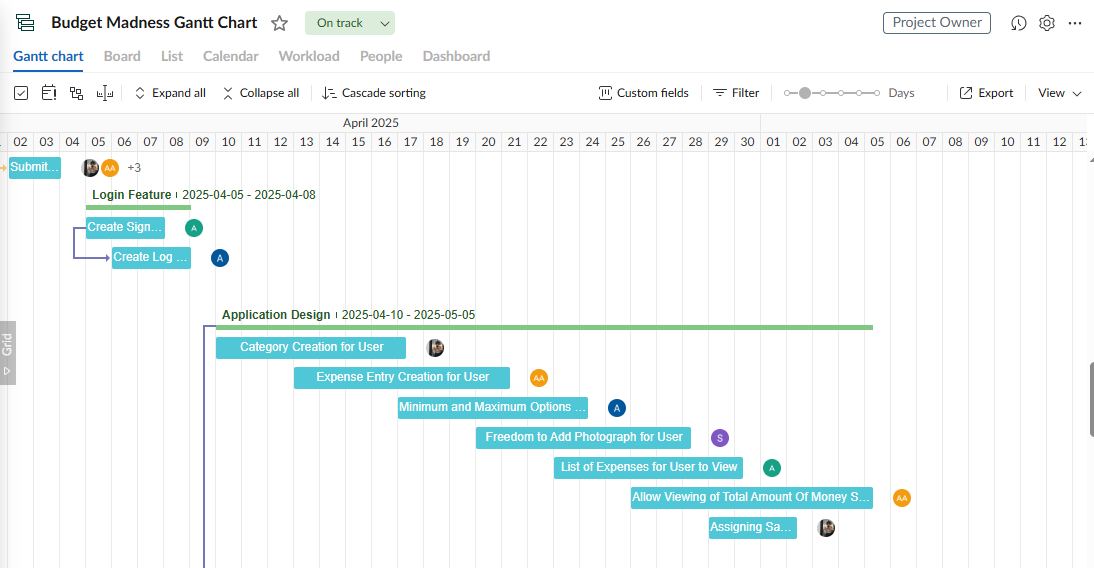
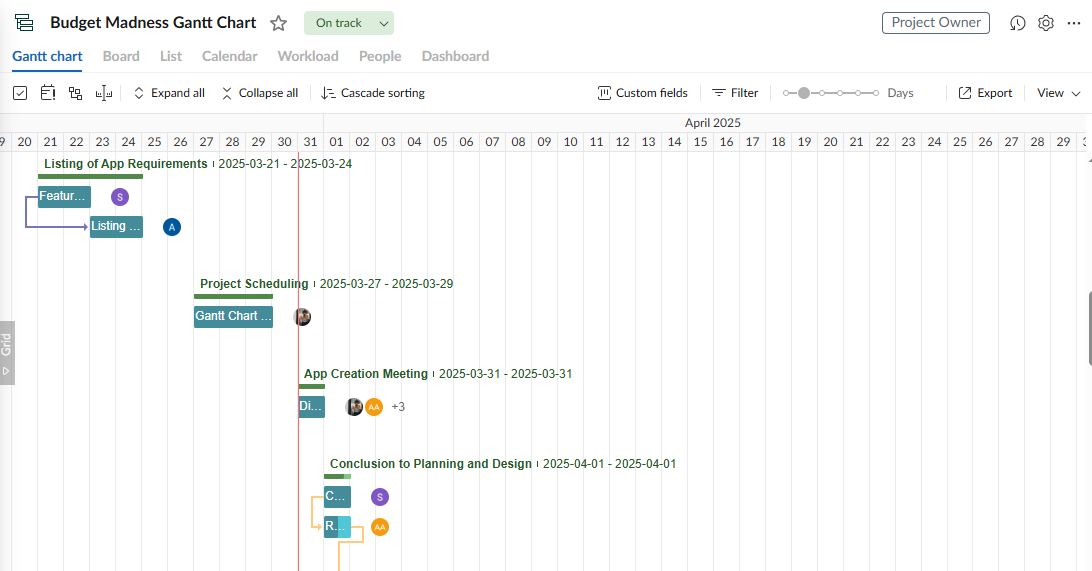
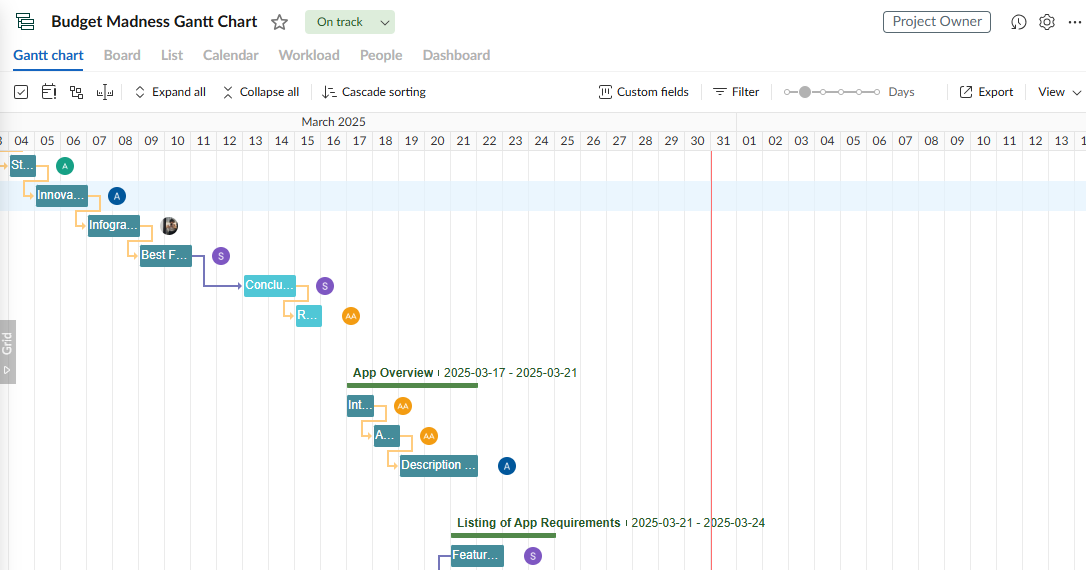
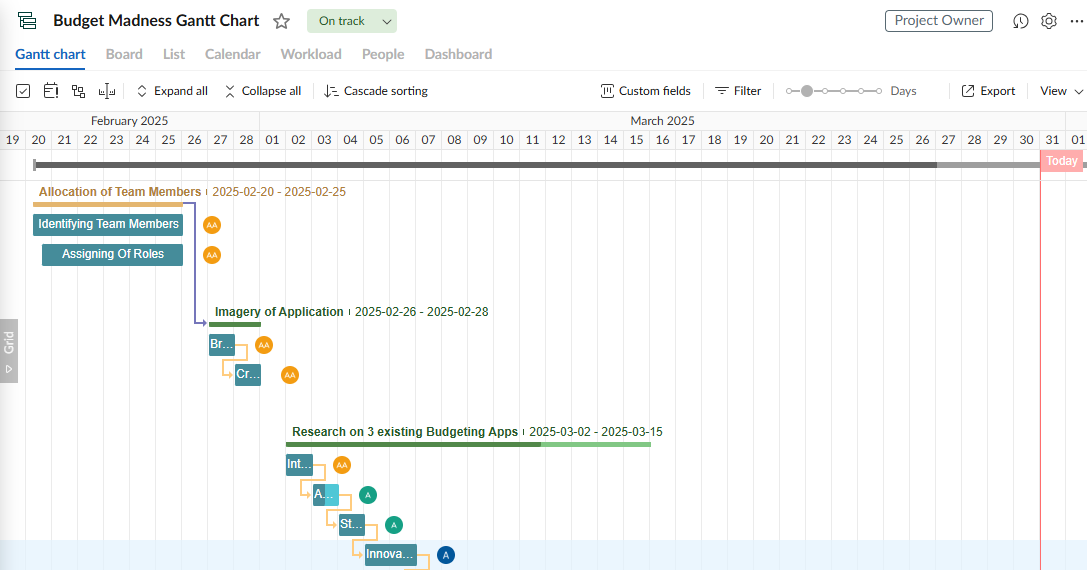
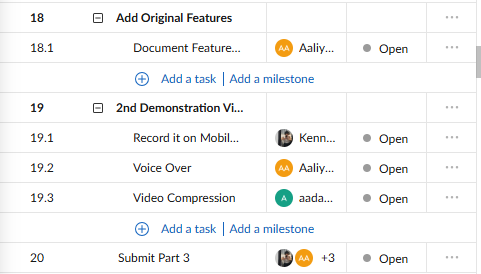
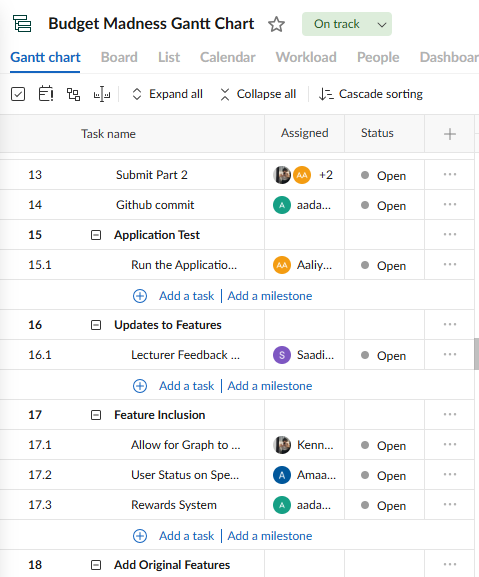
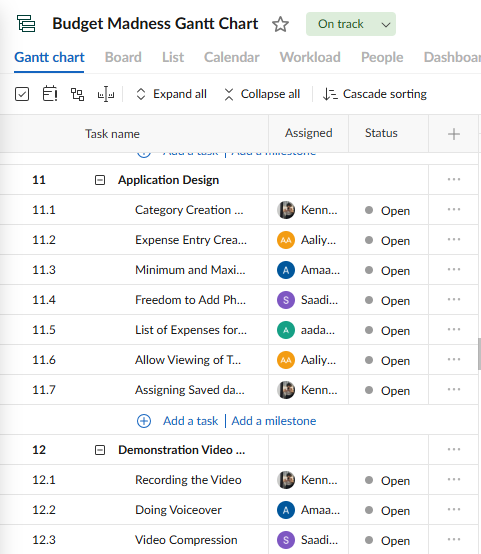
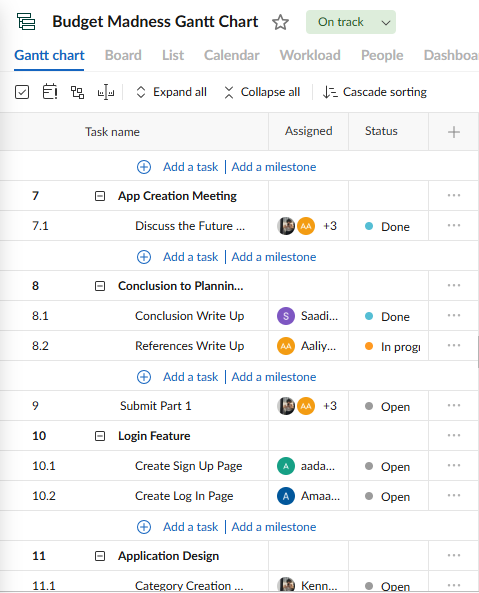
****

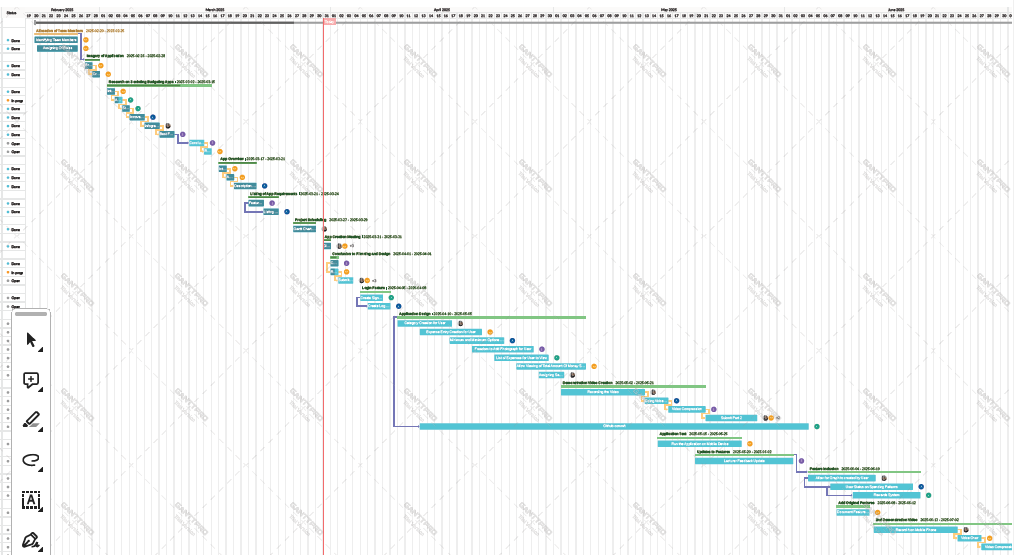
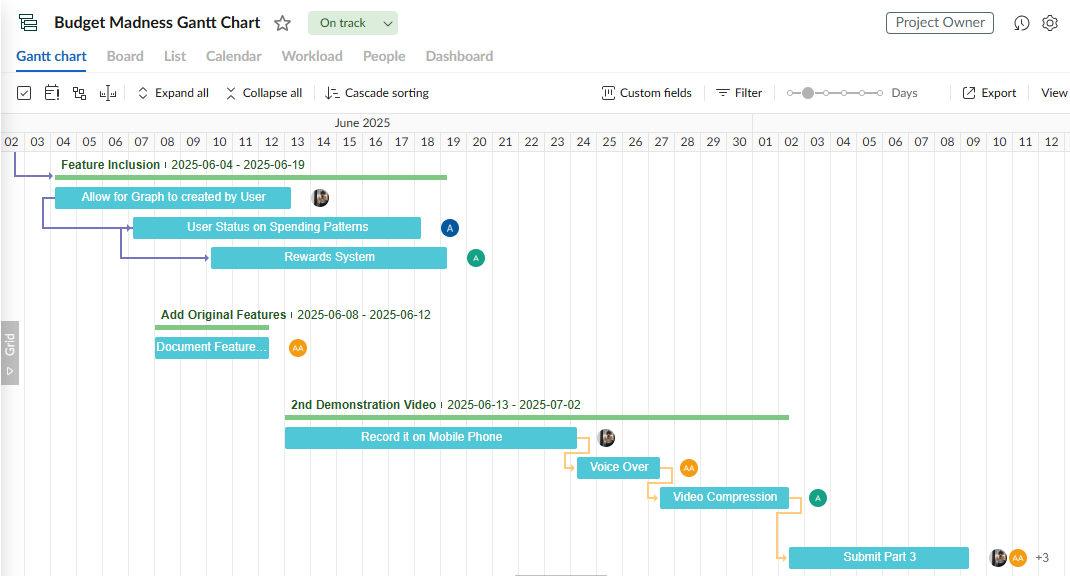
**(draw.io, n.d.)**

### Project plan









(GanttPRO, n.d.)

## Conclusion

In conclusion, the goal of the Budget Madness App project is to provide an engaging and easy-to-use personal money management tool. The planning and design components of the app, including its goals, features, limitations, and target audience, have been thoroughly described in this document. It ensures an organized approach to development by outlining the roles of key stakeholders, important milestones, and deliverables.

Outlining functional and non-functional needs, controlling risks, and creating the overall project schedule were the main goals of the planning stage. In order to lay the foundation for a seamless user experience, the design stage concentrated on UI/UX elements, gamification features, and database structure. Development milestones include adding gamification components, implementing necessary features, and conducting extensive testing to ensure security and dependability.

## References

Adalo Forum, 2023. Reward Badges - How to Create a Badge Reward System When Users Complete Activities. Available at: <https://forum.adalo.com/t/how-to-create-a-badge-reward-system-when-users-complete-activities/28388> [Accessed 29 March. 2025].

Android Developers, n.d. Adaptive Design - Responsive Adaptive Design with Views. Available at: <https://developer.android.com/develop/ui/views/layout/responsive-adaptive-design-with-views#:~:text=UI%20With%20ConstraintLayout.-,Responsive%20width%20and%20height,space%20within%20the%20view's%20constraints>. [Accessed 28 April. 2025].

Android Developers, n.d. Building Notification Alerts. Available at: <https://developer.android.com/develop/ui/views/notifications/build-notification> [Accessed 28 March. 2025].

Android Developers, n.d. Local Database - SQLite in Kotlin Multiplatform. Available at: <https://developer.android.com/kotlin/multiplatform/sqlite> [Accessed 26 March. 2025].

Android Knowledge, n.d. List View. Available at: <https://androidknowledge.com/custom-listview-android-studio-kotlin/> [Accessed 28 March. 2025].

Android Knowledge, n.d. Registration and Login Page. Available at: <https://androidknowledge.com/loginpage-in-android-kotlin/> [Accessed 1 Apr. 2025].

Canva, n.d. Infographic Design. Available at: <https://www.canva.com/> [Accessed 20 March. 2025].

draw.io, n.d**.** Diagramming Tool. Available at: <https://www.draw.io> [Accessed 28 March. 2025].

Figma, n.d. Budget Genius - Financial Budgeting App. Available at: <https://www.figma.com/community/file/1117371877111836342/budget-genius-financial-budgeting-app> [Accessed 20 March. 2025].

Figma, n.d. Personal Budget Tracking. Available at: <https://www.figma.com/community/file/1205442742432724503/personal-budget-tracking> [Accessed 20 March. 2025].

GanttPRO, n.d. Gantt Chart Design. Available at: <https://app.ganttpro.com/> [Accessed 28 March. 2025].

Indeed, 2025. How to write a project scope in 5 steps (with example). Available at: <https://www.indeed.com/career-advice/career-development/project-scope-example> [Accessed 18 March 2025].

Kotlinlang.org, n.d. GraphView - Data Analysis & Visualization in Kotlin. Available at: <https://kotlinlang.org/docs/data-analysis-visualization.html> [Accessed 28 March 2025].

Kotlinlang.org, n.d. Local Database - Connecting to a Database in Kotlin. Available at: <https://kotlinlang.org/docs/data-analysis-connect-to-db.html> [Accessed 28 March. 2025].

Lomana, 2023. Photo Receipt: Kotlin - Take a Photo in Android. Available at: <https://lomana.medium.com/kotlin-take-a-photo-in-android-3f9d5c131688> [Accessed 28 March. 2025].

Świtek, M., 2024. Deciphering the differences: A comprehensive guide to deliverables vs milestones. Teamdeck. Available at: <https://teamdeck.io/resources/deliverable-vs-milestone/> [Accessed 18 March 2025].

YouTube, 2023. Category Creation. Available at: <https://www.youtube.com/watch?v=7S7646Cymn0> [Accessed 28 March. 2025].

YouTube, 2023. Quick Add Feature in Monefy. Available at: <https://www.youtube.com/watch?v=r1P7slDbtd4> [Accessed 28 March. 2025].