

Names of Group Members:

Senzesihle Gama – ST10164052

Hlulani Mfomande – ST10066137

Xichavo Mukansi – ST10408068

Mikateko Baloyi – ST10364813

Course: DISD

Module: OPSC6311

Assessment: POE

A budgeting app is designed to help individuals manage their money more effectively. Its primary purpose is to track income and expenses, plan financial goals, and make smarter spending decisions.

Purpose of a Budgeting App

A budgeting app aims to:

1. **Track Expenses and Income:** Let users log financial transactions.
2. **Monitor Spending Habits:** Categorize spending to reveal trends.
3. **Set and Enforce Budgets:** Enable users to set budget limits by category (e.g., groceries, rent).
4. **Achieve Financial Goals:** Help users plan for savings, debt repayment, or investments.
5. **Provide Insights:** Use analytics to inform users about their financial health.
6. **Reminders and Alerts:** Warn users about upcoming bills, overspending, or savings goals.
7. **Improve Financial Discipline:** Encourages mindful spending **and** Helps break paycheck-to-paycheck cycles
8. **Sync Accounts (Optional):** credit cards

Design Considerations

1. User Interface (UI) / User Experience (UX)

- **Simplicity:** Finance is intimidating for many; the app must be easy to use.
- **Clear Visualizations:** Charts and graphs for budgets, income vs expenses.
- **Accessibility:** readable fonts, etc.

2. Data Privacy & Security

- Use **encryption** for sensitive financial data.
- Implement **secure authentication**

3. Scalability & Performance

- Should handle growth in user base and data size efficiently.

Utilization of GitHub

1. Version Control

- Use Git to track code changes and collaborate with team members.
- Create branches for features, bug fixes, or experiments.

2. Code Reviews

- Use Pull Requests for peer review and discussions.

3. Issue Tracking

- Manage bugs, feature requests, and development tasks using GitHub Issues or Projects.

Utilization of GitHub Actions

GitHub Actions is a Continuous Deployment tool that automates workflows.

Example Use Cases:

1. Automated Testing

- Run unit tests on every PR or commit.
- Example: Test backend logic for tracking and calculating budget totals.

2. Linting & Code Formatting

- Automatically check for code style issues .

3. Build Automation

- Build the app and generate APKs.

4. Security Checks

- Run security analysis tools to check for vulnerabilities in dependencies.

***Additional Features**

- Dark Mode** – Gives users control over appearance and can reduce eye strain.
- **Sync Accounts:** Add credit cards to the app

Presentation video:

<https://youtube.com/playlist?list=PL8aeXcf2kv0dc9wj0in6Ga94njgdbM1vT&si=6AFb1nMeJofSVVRo>

GitHub Link:

<https://github.com/xichavoselby/OPSC6311-part3>