**Requirements Document for Smart Personal Budget Planner**

**1. General Observations**

Many users, including students, professionals, and families, face challenges in managing their finances effectively. They struggle with:

* Tracking daily expenses.
* Sticking to a budget.
* Saving for future goals.

**Specific Example**

Students, for instance, often spend money spontaneously without keeping track. This can result in running out of funds unexpectedly, leading to challenges like skipping meals or borrowing money.

**2. Feedback from Requirement Gathering**

**2.1 Interviews and Focus Groups**

**User Feedback:**

* **"I need an easy way to organize my spending."**
  + **Solution**: Provide predefined categories like "Food," "Rent," "Transport," with the option to add custom categories.
* **"I want to see how much money I have left after expenses compared to my budget."**
  + **Solution**: Add a feature that calculates remaining funds by comparing expenses against the budget in real-time.
* **"It would be great to split expenses with friends easily."**
  + **Solution**: Introduce a group expense-sharing feature to simplify tracking shared costs and balances.
* **"I need daily expense recommendations to meet my monthly savings goals."**
  + **Solution**: Provide daily expense limits calculated dynamically based on the user’s budget and savings goal.
* **"Automating transactions with my bank would be amazing, but I know it might be too advanced."**
  + **Solution**: Simulate a "Banking Feature" by allowing manual entry of income (e.g., allowance or salary) and automatic deduction of expenses. Future upgrades can include bank integration.
* **"I want the app to warn me when I overspend or have a low balance."**
  + **Solution**: Provide real-time alerts for budget thresholds and low balances.
* **"I’d like personalized advice on how to save more based on my spending patterns."**
  + **Solution**: Include basic advisory messages, such as "You’re spending a lot on dining out. Consider cooking at home more often."

**2.2 Surveys**

**Key Findings:**

* **Finding 1**: 60% of respondents overspend early in the month.
  + **Solution**: Implement a "Weekly Spending Plan" to divide the budget into smaller, manageable amounts.
* **Finding 2**: 70% prefer visual tools like charts to understand their spending habits.
  + **Solution**: Offer visual reports such as pie charts and bar graphs to summarize expenses.
* **Finding 3**: 50% of users struggle to set realistic savings goals.
  + **Solution**: Include a savings goal tracker with periodic progress notifications.

**2.3 Observation**

**Observed Issues:**

* Many users rely on memory or random notes to track finances, leading to inconsistent records and missed savings opportunities.
* Users often forget their spending limits, causing unplanned overspending.

**Solutions:**

* Provide a dashboard for quick logging and viewing of expenses.
* Include a daily spending limit feature that adjusts based on the remaining budget.

**3. Core Features**

**3.1 User Management**

* Users can register and log in using an email and password.
* Password reset options are available for forgotten credentials.

**3.2 Budget Management**

* Set monthly budgets based on income.
* Track remaining funds in real-time by comparing expenses against the budget.
* Receive alerts for low balances or overspending.

**3.3 Expense Tracking**

* Add expenses manually with predefined categories like "Food," "Rent," and "Other."
* View daily spending limits based on the remaining budget and savings goals.

**3.4 Group Expense Sharing**

* Record shared expenses and track contributions.
* Generate a simple summary of who owes whom.

**3.5 Visual Reports**

* Generate graphs (e.g., pie charts, bar charts) to display spending patterns.

**3.6 Recommendations and Advice**

* Display personalized advisory messages based on spending habits, such as "Reduce your dining out expenses to save more this month."

**4. Addressing Complex User Needs**

**Complex Requirement 1: Banking Integration**

* **User Request**: Automate income and expense tracking through bank integration.
* **Current Limitation**: Due to limited technical expertise, direct banking integration will not be implemented at this stage.
* **Proposed Solution**: Allow users to manually log their income and expenses to simulate banking functionality.
* **Future Scope**: Explore API integrations with local banks for real-time transaction syncing.

**Complex Requirement 2: AI-Based Insights**

* **User Request**: Provide personalized financial advice based on spending habits.
* **Current Limitation**: AI-driven insights are out of scope for now.
* **Proposed Solution**: Offer basic recommendations, such as "You’re spending too much on X category," based on predefined thresholds.

**5. Formats and Visual Representation**

* **Dashboard**: Clean and user-friendly interface displaying:
  + Current budget status.
  + Daily spending limits.
  + Savings goal progress.
* **Reports**: Generate monthly summaries in both graphical and textual formats.

**6. Conclusion**

The Smart Personal Budget Planner aims to empower users to manage their finances efficiently. By addressing common pain points and including user feedback, the system will serve as a practical and intuitive tool. Future enhancements, such as bank integration and AI-driven insights, will further elevate its utility.

**Scenario 1: Successful Registration**

1. **User:** Navigates to the "Create Account" page.
2. **User:** Enters a valid email address (e.g., "example@email.com").
3. **User:** Enters a strong password (e.g., "Password123!").
4. **User:** Confirms the password.
5. **User:** Clicks the "Create Account" button.
6. **System:** Verifies the email format and password strength.
7. **System:** Creates a new user account in the database.
8. **System:** Sends a confirmation email to the provided address.
9. **System:** Displays a success message ("Account created successfully!") and redirects the user to the login page.

**Scenario 2: Invalid Email Format**

1. **User:** Navigates to the "Create Account" page.
2. **User:** Enters an invalid email address (e.g., "example").
3. **User:** Enters a valid password.
4. **User:** Confirms the password.
5. **User:** Clicks the "Create Account" button.
6. **System:** Displays an error message ("Invalid email address format.").
7. **User:** Corrects the email address (e.g., "example@email.com").
8. **User:** Continues with the registration process.

**Scenario 3: Weak Password**

1. **User:** Navigates to the "Create Account" page.
2. **User:** Enters a valid email address.
3. **User:** Enters a weak password (e.g., "password").
4. **User:** Confirms the password.
5. **User:** Clicks the "Create Account" button.
6. **System:** Displays an error message ("Password is too weak. Please use a combination of uppercase and lowercase letters, numbers, and symbols.").
7. **User:** Enters a stronger password.
8. **User:** Continues with the registration process.

**Scenario 4: Username Already Exists**

1. **User:** Navigates to the "Create Account" page.
2. **User:** Enters an email address that is already registered.
3. **User:** Enters a valid password.
4. **User:** Confirms the password.
5. **User:** Clicks the "Create Account" button.
6. **System:** Displays an error message ("Email address is already registered.").
7. **User:** Enters a different email address and continues with the registration process.