**1. Understand the M-Pesa API**

* **Documentation:** Obtain the official documentation for the M-Pesa API.
* **Integration:** Understand how to integrate with the API, including authentication, sending money, and managing transactions.

**2. Define the App Features**

* **User Accounts:**
  + **Parent Account:** Create and manage budget for the child.(basically the parent puts in the money and sets a daily budget)
  + **Child Account:** Receive daily disbursements.
* **Budget Management:**
  + Set monthly budget.
  + Calculate and set daily disbursements.
* **Transaction History:**
  + Track all transactions.
  + Notify parents and children of transactions.
* **Notifications:**
  + Daily disbursement notifications.
  + Alerts for low balances or budget limits.

**3. Technical Requirements**

* **Backend (PHP):**
  + Server to handle API requests, user authentication, and database management.
  + Secure storage for user data and transaction history.
* **Frontend:**
  + Simple web interface for parents and children.
  + Forms for setting budgets and viewing transactions.
* **Database:**
  + MySQL (included with XAMPP) to store user information, budgets, and transaction records.
* **Security:**
  + Ensure secure communication with the M-Pesa API.
  + Protect user data and financial information.

**4. Tech Stack**

* **Frontend:** HTML, CSS, JavaScript (with a library like jQuery or Vue.js for dynamic interactions).
* **Backend:** PHP (possibly with Laravel), XAMPP.
* **Database:** MySQL.
* **API Integration:** Use cURL or GuzzleHTTP for making HTTP requests to the M-Pesa API.

**5. Development Plan**

* **Week 1:**
  + Set up the development environment with XAMPP.
  + Integrate the M-Pesa API and test basic transactions using cURL or GuzzleHTTP.
  + Develop the user authentication system.
  + Start the frontend design with HTML, CSS, and JavaScript.
* **Week 2:**
  + Implement the budget management features.
  + Complete the frontend design and link it with the backend.
  + Test the entire flow from setting a budget to daily disbursements.
  + Prepare the presentation for the hackathon.

**6. Development Steps**

1. **Set Up Environment:**
   * Install XAMPP and configure it for PHP development.
   * Set up a MySQL database for your app.
2. **M-Pesa API Integration:**
   * Obtain API credentials and set up a sandbox environment for testing.
   * Write PHP functions to authenticate with the API and perform transactions.
3. **User Authentication:**
   * Develop a registration and login system using PHP and MySQL.
   * Secure the user data and ensure proper session management.
4. **Budget Management:**
   * Create a dashboard for parents to set and manage budgets.
   * Develop the logic to calculate daily disbursements and store these in the database.
5. **Transaction Management:**
   * Implement PHP scripts to handle daily disbursements.
   * Use CRON jobs to automate the daily disbursement process.
6. **User Interface:**
   * Design forms and dashboards for both parents and children.
   * Use HTML, CSS, and JavaScript to create a responsive and user-friendly interface.
7. **Notifications:**
   * Integrate a notification system using email (PHPMailer) or SMS (using an API like Twilio).
8. **Testing and Deployment:**
   * Thoroughly test all features to ensure reliability and security.
   * Prepare a demo and documentation for the hackathon presentation.

**Questions to Clarify**

1. **User Authentication:**
   * Will you use email/password login, or do you need to integrate with M-Pesa for authentication?
2. **Transaction Limits:**
   * Are there any daily transaction limits or fees from M-Pesa that you need to consider?
3. **User Interface:**
   * Do you have any specific design ideas or wireframes for the app?
4. **Notification System:**
   * How will you notify users? Via SMS, in-app notifications, or email?

**Next Steps**

1. **Get the M-Pesa API credentials and set up a development account.**
2. **Design the database schema and API endpoints for your backend.**
3. **Start with basic user authentication and M-Pesa integration.**
4. **Iterate on the app's features and test thoroughly.**