# Project Charter for Transaction Tracker App

## Project Title

Project Name: Transaction Tracker App

Date: [Insert Date]

Project Manager: [Insert Name]

## Project Scope

Description of the Project: Develop a Django/React application that runs locally on a Raspberry Pi to manage and track transaction data across various accounts.

Objectives: The app will only be accessible within a local area network.

## Stakeholders

List of Stakeholders: Identify all internal and external parties involved.

## Roles and Responsibilities

Team Structure: Outline the project team and the specific roles of each member.

## Deliverables

Major Deliverables: Develop a robust system to handle transaction data for multiple sources, provide a unified view of account transactions and balances, and include customer data management, differentiating between owned and non-owned accounts.

## Milestones and Timeline

Gantt Chart: Include a Gantt chart to visualize the project timeline, key milestones, and dependencies.

## Project Summary

Transaction Tracker App: A local Django/React application designed to run on a Raspberry Pi for managing transaction data within a LAN.

## Technical Specifications

Backend (Django):

Models: Source, Transaction, Account, Customer.

API: RESTful API using Django REST Framework for CRUD operations.

Database: SQL-based, details to be finalized.

Security: Implement security best practices for local network usage.

Frontend (React):

UI Components: Interactive tables for displaying data, forms for data entry, and account management features.

State Management: Use tools like Redux or Context API.

Communication: Fetch/Axios for API calls to the backend.

Development Environment:

Local Setup: Development on a laptop with a simulated Raspberry Pi environment using Docker or virtual machines.

Deployment: Actual deployment on Raspberry Pi, setup to run the developed application within a local network.

## Resources

Budget: Outline the budget allocated for the project.

Tools and Technology: List the tools and technologies that will be used, including Django, React, Docker, and Raspberry Pi.

## Risk Management

Risk Identification: Identify potential risks and their impact on the project.

Mitigation Strategies: Describe the strategies to mitigate identified risks.

## Communication Plan

Communication Methods: Specify how and when the project updates will be communicated.

## Project Tracking

Kanban Board: Use a Kanban board to track ongoing tasks, their status, and progress.

## Visual Mapping

Mind Map: Create a mind map to visualize project ideas, processes, and task relationships.

## Development Phases

Research Phase:

Research Docker ARM Emulation.

Choose security measures for the Pi deployment.

Research Django REST Framework Security Practices.

Study React State Management.

Planning Phase:

Outline Project Requirements and Specifications (Apr 26-29).

Outline API endpoints needed for CRUD operations.

Design database schema for Source, Transaction, Account, and Customer Models.

Implementation Phase:

Set up Django Project Structure.

Develop Initial React components.

Draft Data Models.

Testing Phase:

Test API with Postman.

UI testing on laptop.

Review Phase:

Functionality checks on emulated environment.

Document API Endpoint Design.

Deployment Phase:

Final review and adjustments for deployment on Raspberry Pi.

Obtain stakeholder sign-off and go live within the local network.

## Approval

Signatures: Space for signatures from key stakeholders to approve the charter.

## Appendices

Additional Information: Any supplementary information relevant to the project.

# Notional Kanban Board for Transaction Tracker App

## To Do

Research Docker ARM Emulation

Choose Security Measures for Pi Deployment

Research Django REST Framework Security Practices

Study React State Management

Outline Project Requirements and Specifications (Apr 26-29)

Outline API Endpoints Needed for CRUD Operations

Design Database Schema for Source, Transaction, Account, and Customer Models

## In Progress

Set Up Django Project Structure

Develop Initial React Components

## Testing

Test API with Postman

UI Testing on Laptop

## Review

Functionality Checks on Emulated Environment

Document API Endpoint Design

## Done

Draft Data Models

Design database schema for Source, Transaction, Account, and Customer Models

# Kanban Board for Transaction Tracker App

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| To Do | In Progress | Testing | Review | Done |
| Research Docker ARM Emulation | Set Up Django Project Structure | Test API with Postman | Functionality Checks on Emulated Environment | Draft Data Models |
| Choose Security Measures for Pi Deployment | Develop Initial React Components | UI Testing on Laptop | Document API Endpoint Design | Design database schema for Source, Transaction, Account, and Customer Models |
| Research Django REST Framework Security Practices |  |  |  |  |
| Study React State Management |  |  |  |  |
| Outline Project Requirements and Specifications (Apr 26-29) |  |  |  |  |
| Outline API Endpoints Needed for CRUD Operations |  |  |  |  |
| Design Database Schema for Source, Transaction, Account, and Customer Models |  |  |  |  |