WalleTx

Software Research Specification

Prestige Worldwide

Group Members:
Arron Solano
Brian Howell
Daniel Carroll
Michael Danko

Contents

1 Introduction

1.1 Purpose

Bitcoin WalleTx is a bitcoin wallet tracker tool for Android that assists users in monitoring their bitcoin balance, transaction history, and spending trends across multiple wallets. It is common for bitcoin users to possess numerous wallets, yet there do not exist many services that are capable of aggregating wallet information in order to provide the user with an overall picture of their bitcoin funds. Bitcoin WalleTx solves this problem by allowing users to group or categorize their bitcoin wallets, as well as tag their transactions with real-world information related to how their bitcoins are being spent. Charts and graphs help to identify trends in both single wallet spending and across wallet groups, a useful feature that is essential for enabling users to integrate their bitcoin finances with a more traditional budget.

Bitcoin WalleTx is targeted toward all bitcoin users, as it is useful for identifying trends regardless of the number of wallets held. The average person is more concerned with budgeting than with hashes and confirmation numbers, and transaction tagging will enable users to identify trends in their spending. The concept of financial trackers has been extremely successful outside of the bitcoin space with services like Mint.com, and Bitcoin WalleTx aims to bring some of these needed features to bitcoin.

1.2 Scope

Scope

1.3 Definitions

Definitions

1.4 Acronyms

Acronyms

1.5 References

- Bitcoin: A Peer-to-Peer Electronic Cash System
  `https://bitcoin.org/bitcoin.pdf`
- Blockchain Data API
  `https://blockchain.info/api/blockchain_api`

1.6 Overview

2 General Description

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 General Constraints

2.5 Assumptions and Dependencies

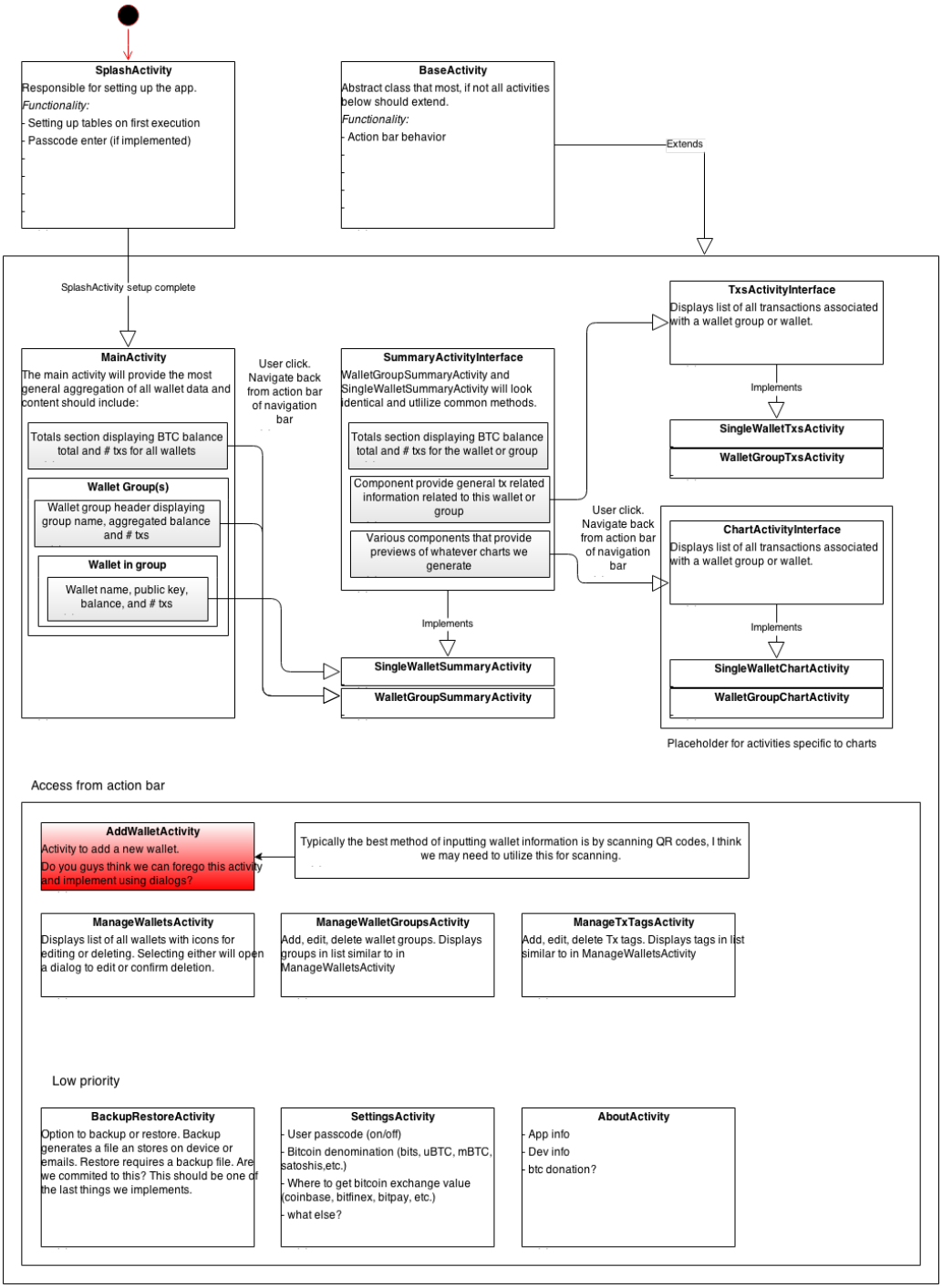
3 Functional Requirements

- Import & delete public keys for balance\transaction tracking
- Define & delete wallet groups
- Group public keys based on user preference
- Define & delete transaction tags
- Tag transactions
- View single wallet (or group) balance & transaction summaries
- View single wallet (or group) balance & transactions detail view
- View single wallet (or group) balance & transactions graphical views
- View single wallet (or group) transaction trends using tags
- View transactions details

4 Non-Functional Requirements

5 System Architecture

6 System Model



7 Appendices

7.1 Data Dictionary

7.1.1 Actor Descriptions

Identifying actors:

- Who will supply, use, or remove information from the system?
- Who will use the system?
- Who is interested in a certain feature or service provided by the system?
- Who will support and maintain the system?
- What are the system's external resources?
- What other systems will need to interact with the system under development?

Actors will be the customer/user of the application. Most interested in our application would be those interested in Bitcoin. Service and maintenance of the system will be performed by the developers(Prestige Worldwide). Systems external resources will be blockchain, SQLite, and other external wallets providing information. (Michael and Brian please confirm external resources of Bitcoin)

7.1.2 Use Case Descriptions

Identifying use cases:

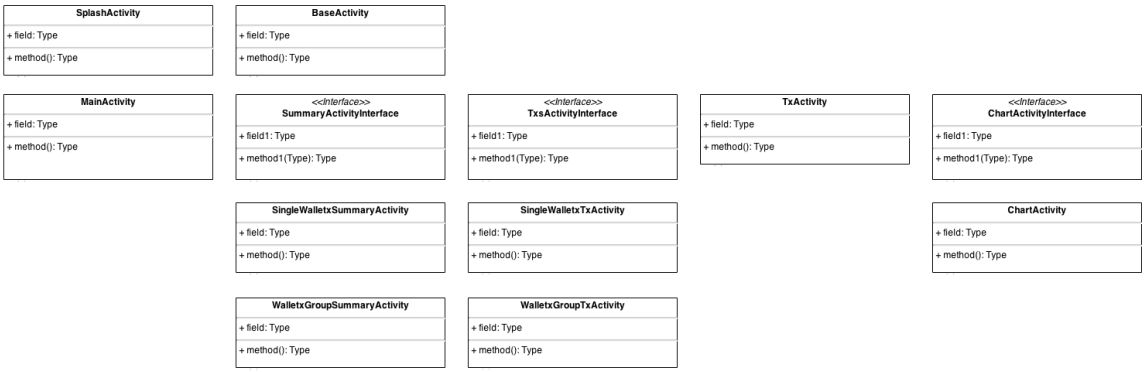
- What are the goals that the actor will attempt to accomplish with the system?
- What are the primary tasks that the actor wants the system to perform?
- Will the actor create, store, change, remove, or read data in the system?
- Will the actor need to inform the system about sudden external changes?
- Does the actor need to be informed about certain occurrences, such as unavailability of a network resource, in the system?
- Will the actor perform a system startup or shutdown?

Use cases will be outlined in diagrams provided.

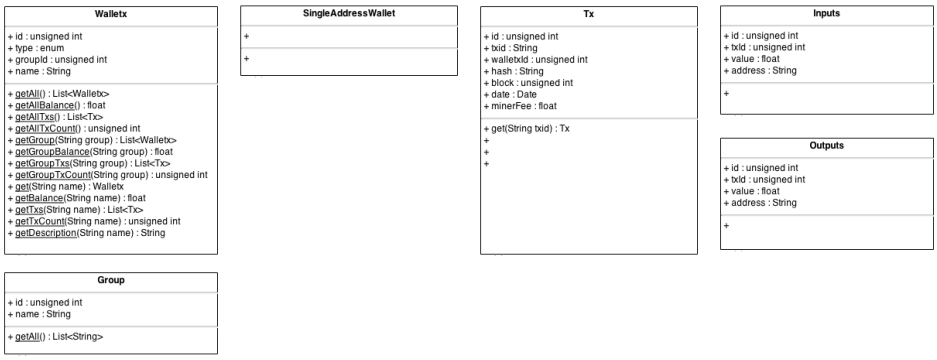
The primary task of the customer/user will be to interact with the application. Their main purpose will be to monitor their bitcoin currency from multiple wallets and tag transactions to better organize all transactions. The actor will have the ability to add and remove wallets/add and remove tags pertaining to transactions. The actor can observe aggregate information in the form of graphs, charts, etc. The actor will be alerted to certain trigger events. (updated wallet info, specific transactions, etc). The actor has the ability to close and open application in running OS and also has the ability to completely remove the application from existing OS.

7.1.3 Class Descriptions

ANDROID ACTIVITIES



MODEL



BACKEND

7.1.4 Attribute Descriptions

7.2 Raw Use Case Point Analysis

7.2.1 Actor Summary Table

Actors	Summary
Customer/End User	This actor will be the end user. They will interact with WalleTx, providing it with wallet information, tagging transactions, and viewing aggrated wallet information
Development Team/Maintenance	This actor will be interacting via the support and maintance of the application. They will maintain DB orginzation, revision handling, and overall maintaince of the application
External Bitcoin Wallets	External Bitcoin Wallets will be providing our application with vital information. All external wallets' totals and transactions will aggreated in WalleTx.
SQLite Database	This actor is a database stored on each end user's phone. This will contain data from wallets, transactions, and display information. It will interact soley with the application via code.
Bitcoin Blockchain	The bitcoin blocktrain will provide transaction specific data for all included wallets. This actor will interact with the application via database backend.

7.2.2 Use Case Summary Table

7.2.3 Screens and Reports with Navigation Matrix

7.3 Other Appendices