WalleTx

Software Research Specification

Prestige Worldwide

Group Members:
Arron Solano
Brian Howell
Daniel Carroll
Michael Danko

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to define the requirements for Bitcoin WalleTx. The intended audience of this document includes the CEN4020 instructional staff at FSU and the end users of the WalleTx. It is also intended that this document serve as the principle point of reference for the Prestige Worldwide team throughout the development of WalleTx.

Bitcoin WalleTx (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.

1.2 Scope

1.2.1 Product

Bitcoin WalleTx (referred to as WalleTx)

1.2.2 Scope of Work

WalleTx is an android application that enables users to import their bitcoin wallets by public key and tag their bitcoin transactions in order to obtain an aggregated overview of their bitcoin finances. WalleTx shall allow users to manage their public keys; manage wallet groups; manage transaction tags; tag individual transactions; and receive feedback regarding trends via charts and graphs. A full listing of functional requirements for WalleTx is located in the Functional Requirements section of this document.

1.2.3 Out of Scope

WalleTx shall not provide any functionality allowing users to spend or receive bitcoins since only public keys shall be stored on the device of the user.

1.2.4 Application

Bitcoin is a new technology and the ecosystem is currently undergoing major infrastructure development. There currently do not exist any tools that allow users to label their bitcoin transactions in order to identify trends in their bitcoin spending behavior. WalleTx aims to bring this functionality to the Bitcoin space.

1.3 Definitions

- **WalleTx** - Shorthand for Bitcoin WalleTx
- **Bitcoin** - a tradeable virtual asset existing on the ledgers of a distributed and synchronized network of ledgers
- **bitcoin** - a tradeable virtual asset unit, divisible into 100 million units (also known as a satoshi)
- **Blockchain** - A block chain is made of a blocks, linked to the block before and block after, that contain information pertinent to the unerlying system. In the case of Bitcoin these blocks contain transaction information.
- **Public key** - A hash of a wallet's public key that allows other users on the Bitcoin network to send bitcoins to the Public Key.
- **Tx** - A transaction that moves bitcoins from one address to another, usually in exchange for goods or services.
- tag?
- Master Public Key

1.4 Acronyms

BTC Common unit of Bitcoin currency

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

Description of the remainder of the SRS. The remainder of the document outlines the functionality and operation of the WalleTx software. It outlines a general description of who, what, why, when, and how individuals will use our software. It provides the functional and non-functional constraints our software will incur. The document will also serve to illustrate software functionality from a high level architectural view to minute details about every user scenario the software may encounter. The appendix provides information and graphs pertinent to many of these details.

2 General Description

WalleTx is an internet connected, bitcoin connected software that is used as a financial managemnt tool. With this software you are able to review a history of spending with the option to categorize individual transactions.

2.1 Product Perspective

WalleTx is dependent on the bitcoin blockchain to source information regarding tranactions amounts, times, and dates. It is possible that manual data entry can be implemented for those transactions however it is assumed the blockchain access will be available at all times. If there is not internet, you cannot spend your bitcoin. Accessing this software will be completed through mobile smartphones with Android operating systems.

2.2 Product Functions

Product functions include accessing the bitcoin blockchain to download transactions, allowing users to tag and categoryize transactions into which categories they wish. And then to analyze data by association to come to their own conclusions with regard to spending habits. Users will be able to add multiple wallets to allow for spending and saving 'wallets'.

2.3 User Characteristics

Users are from a broad spectrum of every day life. It is WalleTx's intention to be available to people of all ages from every corner of the globe. Bitcoin is a global network, with many users from young to old and WalleTx will be usuable by the same demographic. Anyone who is capable of utilizing bitcoin technology to transact will be able to use WalleTx to manage their financial records.

2.4 General Constraints

The system will be designed by a group of 4 junior developers and this may impose constraints on development. The software will not be constrained by regulatory policies.

To finish
(1) Regulatory policies
(2) Hardware limitations; for example, signal timing requirements
(3) Interface to other applications
(4) Parallel operation
(5) Audit functions
(6) Control functions
(7) Higher-order language requirements
(8) Signal handshake protocols; for example, XON-XOFF, ACK-NACK.
(9) Criticality of the application
(10) Safety and security considerations

2.5 Assumptions and Dependencies

The assumption is that the application will be used on a mobile device. The next assumption is that the mobile device is utilizing the Android operating system. At this time the application will only be able to run on the Android operating system.

3 Functional Requirements

3.1 The system shall allow users to manage their imported wallets or public keys

3.1.1. Users shall be able to add new single address wallets into the system

3.1.2. Users shall be able to edit previously existing wallets

3.1.3. Users shall be able to delete existing wallets from the system

3.1.4. Users shall be able to name individual wallets and assign them to a group

3.1.5. The system should be designed to allow for future integration with other wallet types such as deterministic seed wallets or service based wallets such as Coinbase and Circle

3.2 The system shall allow users to customize wallet groups

3.2.1. System shall designate a default wallet group

3.2.2. Users shall be able to add new wallet groups by name

3.2.3. Users shall be able to edit previously existing group names

3.2.4. Users shall be able to delete existing wallet groups, an action that should reassign all wallets associated with the group to the default group

3.3 The system shall allow users to manage transaction tag categories

3.3.1. User shall be able to define new tag categories by name

3.3.2. User shall be able to edit existing tag categories

3.3.3. User shall be able to delete existing tag categories, an action that should remove the tag from any transactions it has been assigned to

3.4 The system shall allow users to manage tags for individual transactions

3.4.1. User shall be able to add a tag to an individual transaction

3.4.2. User shall be able to remove a tag from an individual transaction

3.5 System shall provide an aggregate view of all imported wallets that contains the total balance and number of transactions

3.6 System shall provide a summary view for each wallet group that provides an overview of the group balance, transactions, and charts

3.7 System shall provide a summary view for individual wallets that provides an overall summary of its balance, transactions, and charts

3.8 System shall provide a transactions detail view for each wallet group that lists the transactions associated with the group

3.9 System shall provide a transactions detail view for each individual wallet that lists its transactions

3.10 System shall provide graphical charts for identifying transaction trends

3.10.1. System shall provide pie charts views breaking down transaction percentage by tag for all wallet groups and for individual wallets

3.10.2. System should provide additional trend identifying charts for wallet groups and individual wallets

3.11 The system should provide a view containing individual transaction information

3.12 The system should allow users to backup their data to an external file and import data from a previously created backup file

3.13 The system should allow users to passcode protect the application

3.14 The system will be able to sync with the blockchain every 10 minutes to verify new blocks (possibly use web socket)

4 Non-Functional Requirements

4.1 Query and reporting times - this will be dependent on the end users RAM and processor speed. Also database placement and architecture. Must account for moderate processor speeds and run on little as RAM as possible. Android Framework provides tools to test and profile CPU and RAM usage. Querying the blockchain will be done via the Blockchain API and speed to return a query will be dependent on outside services and network connection speed.

4.2 Storage - this will be dependent on the end users storage memory on the phone. Must have small footprint on phone's storage. Need to compare application size for similar available applications to scope size. Records will be stored locally to prevent repeated connections to the blockchain network for duplication information. Records and their added tagging information will probably be measured in hundreds of bytes meaning that the storage footprint will be very reasonable.

4.3 Response time (between activities) - this will be dependent on the end users RAM and design of UI. Must account for low RAM and optimize UI for quick activity. This evaluation will fall under query and reporting time.

4.4 Screen Resolution - dependent on end users phone. Must find resolution to work on most phones. (font, layout tweaks, image changes)

4.5 Versions - Need to specify and target revision to reach optimal user targets. Evaluated by most popular revision available or with best compared resources.

4.6 No system downtime - All updates to the app will be done through Android updates pushed through Google Play Store. Standard update process for life of application on Android platform.

4.7 Battery Usage - this will be dependent on the end users battery size, installed applications, and screen size. Must try to minimize battery drain. Need to evaluate against controls to test battery usage.

5 System Architecture

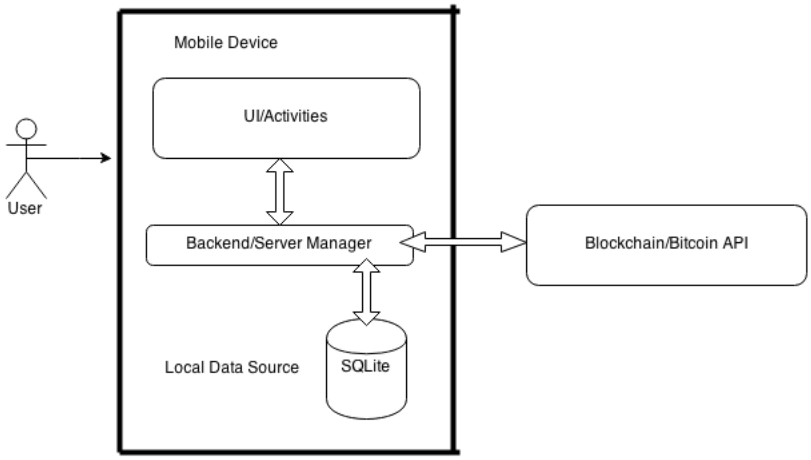


Figure 1: System Architecture UML

Architecture Overview:
The application will be accessed and stored on an Android phone. The user will interact with application using touch screen.

- Environment is mobile, can be used anywhere
- Users will have an existing bitcoin account(s)
- Main functions consist of storing and organizing bitcoin wallet(s) information and displaying data
- Input is users bitcoin account(s), output is data organization and customization

Logical Overview:

- Utilizing Android OS on mobile phones (rev to be determined), developing in Android SDK framework using JAVA
- Data storage will be local, utilizing SQLite
- Data gathered form account(s) using blockchain and bitcoin API
- IDE will Google's Android Studio

UI/Activities:

- Will be designed using Android SDK standards kn XML mark-up
- Purpose provides interaction with end user
- Interactions include end user input and data binding with back-end/server manager

Back-end/Server manager:

- Will be designed using Android SDK standards in JAVA
- Purpose provides interaction between database, blockchain, bitcoin API, and UI/Activities
- Data binding will occur in this area
- Interactions include data management, activities controller, and external data management

SQLite Database:

- Will be designed using an undecided SQLite tool utilizing sql queries and statements

- Purpose provides data storage and management for the application
- Interactions include Back-end/Server communication for queries and storage

Blockchain/Bitcoin:

- Bitcoin interaction in application will be limited to API interaction and blockchain interaction
- Will not need to be designed. Will be managed by Back-end/Server
- Purpose is to provide data and information from blockchain and user's existing bitcoin account(s)
- Interactions will occur with Back-end/Server

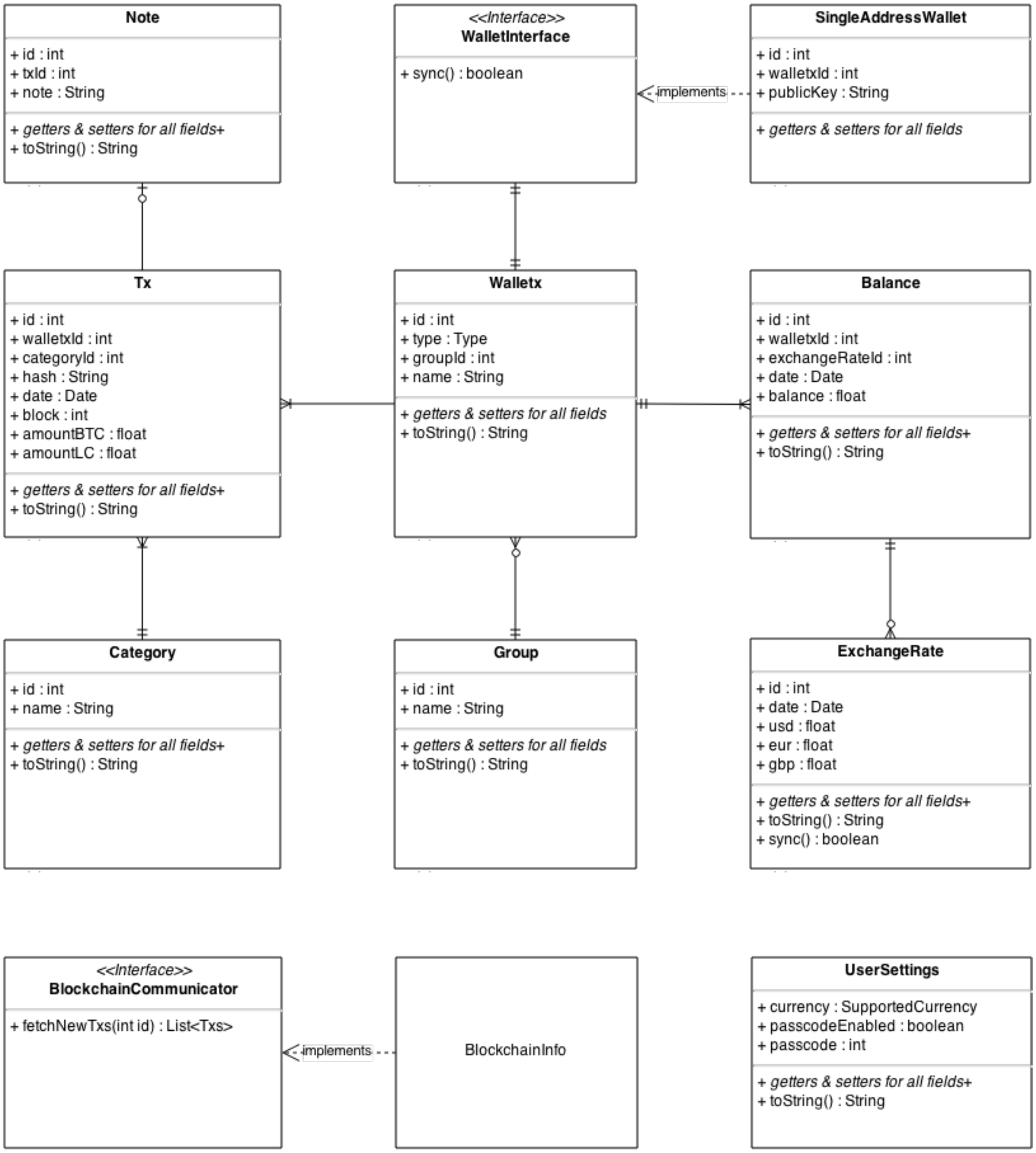


Figure 2: Class Models

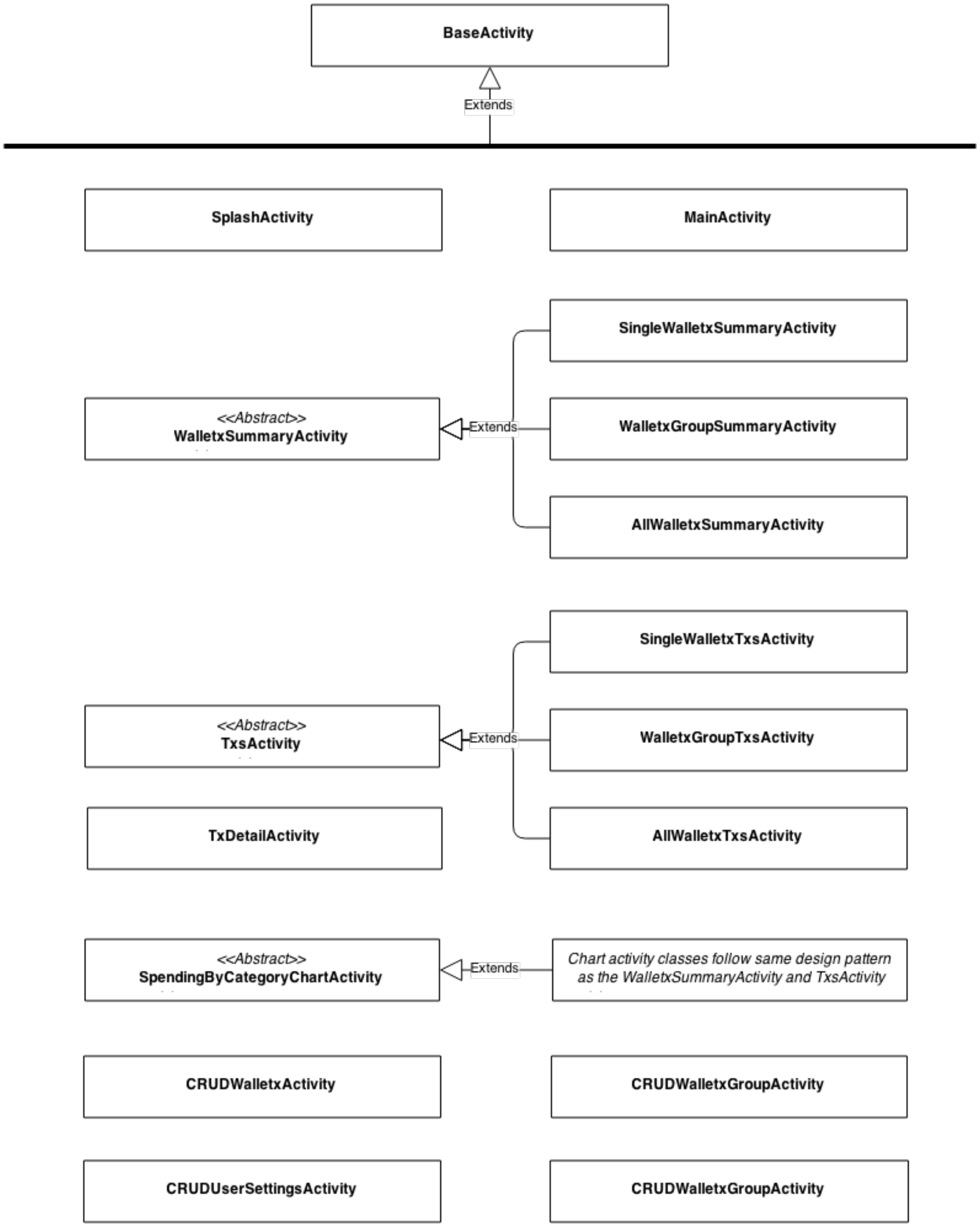


Figure 3: Class Activities

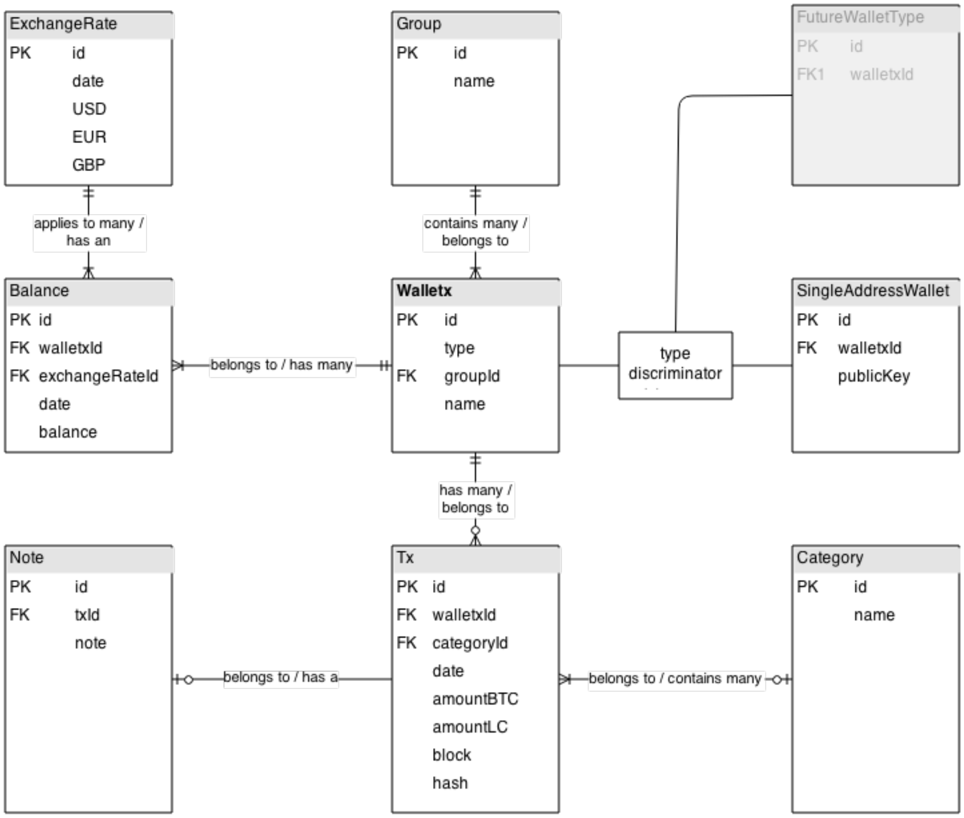


Figure 4: Database Schema

6 System Model

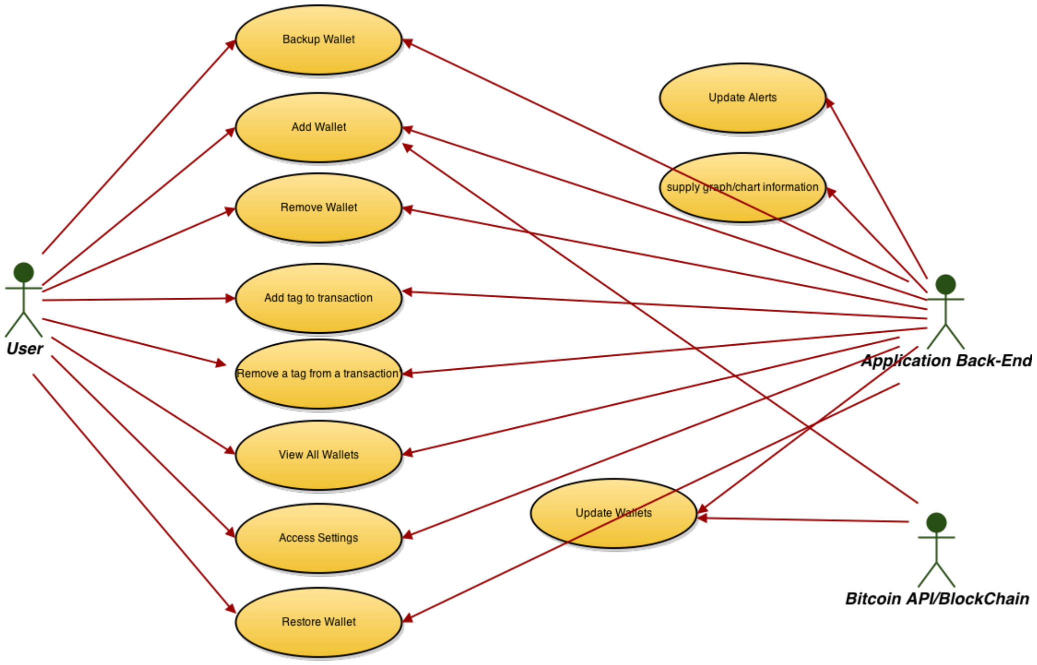
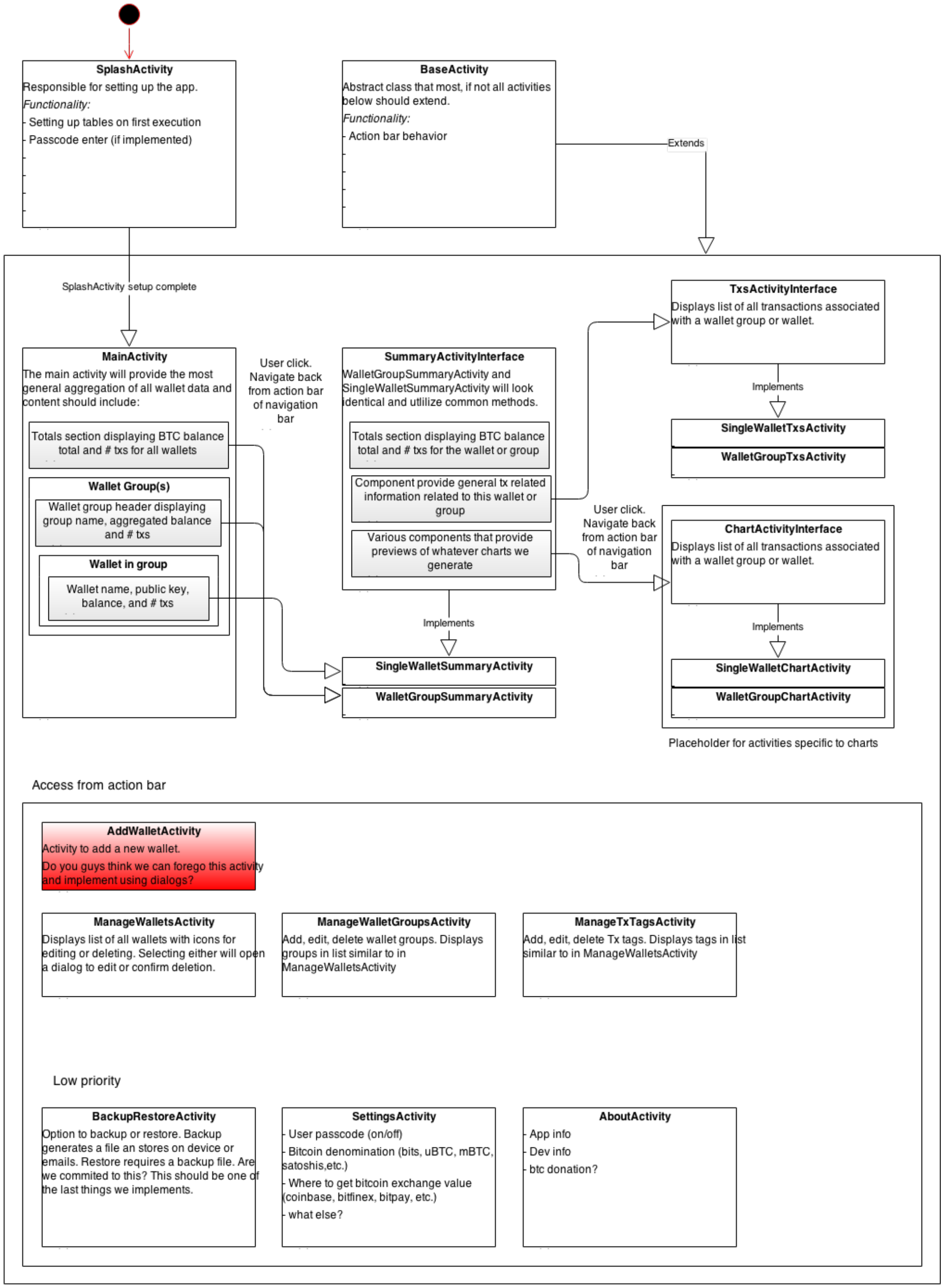


Figure 5: Master Use Case Diagram



SplashActivity
Responsible for setting up the app.
Functionality:
- Setting up tables on first execution
- Passcode enter (if implemented)

BaseActivity
Abstract class that most, if not all activities below should extend.
Functionality:
- Action bar behavior

MainActivity
The main activity will provide the most general aggregation of all wallet data and content should include:

Totals section displaying BTC balance total and # txs for all wallets

Wallet Group(s)
Wallet group header displaying group name, aggregated balance and # txs

Wallet in group
Wallet name, public key, balance, and # txs

User click. Navigate back from action bar of navigation bar

SummaryActivityInterface
WalletGroupSummaryActivity and SingleWalletSummaryActivity will look identical and utilize common methods.

Totals section displaying BTC balance total and # txs for the wallet or group

Component provide general tx related information related to this wallet or group

Various components that provide previews of whatever charts we generate

SingleWalletSummaryActivity

WalletGroupSummaryActivity

TxsActivityInterface
Displays list of all transactions associated with a wallet group or wallet.

SingleWalletTxsActivity

WalletGroupTxsActivity

User click. Navigate back from action bar of navigation bar

ChartActivityInterface
Displays list of all transactions associated with a wallet group or wallet.

SingleWalletChartActivity

WalletGroupChartActivity

Placeholder for activities specific to charts

Access from action bar

AddWalletActivity
Activity to add a new wallet.
Do you guys think we can forego this activity and implement using dialogs?

ManageWalletsActivity
Displays list of all wallets with icons for editing or deleting. Selecting either will open a dialog to edit or confirm deletion.

ManageWalletGroupsActivity
Add, edit, delete wallet groups. Displays groups in list similar to in ManageWalletsActivity

ManageTxTagsActivity
Add, edit, delete Tx tags. Displays tags in list similar to in ManageWalletsActivity

Low priority

BackupRestoreActivity
Option to backup or restore. Backup generates a file an stores on device or emails. Restore requires a backup file. Are we commited to this? This should be one of the last things we implements.

SettingsActivity
- User passcode (on/off)
- Bitcoin denomination (bits, uBTC, mBTC, satoshis,etc.)
- Where to get bitcoin exchange value (coinbase, bitfinex, bitpay, etc.)
- what else?

AboutActivity
- App info
- Dev info
- btc donation?

Figure 6: Android Activies Summary