CPS731 - Requirements Engineering Document - Group 11

Brady Cunningham - brady.cunningham@ryerson.ca
Jacky Tran (jacky.tran@ryerson.ca)
Jimmy Tran (jvtran@ryerson.ca)
Lahiru Atapattu (latapattu@ryerson.ca)
Rodrigo Goncales (rgoncales@ryerson.ca)

User Stories (Tracked on GitHub as separate issues):

As a user, I want to be able to login with a user account.

As a user, I want to be able to customize my user profile with a name, photo and contact email.

As a user, I want to be able to view other users profile page, so that I can view detailed info.

As a user, I want to be able to contact friends, so that I can notify them about money owed.

As a user, I want to be able to view balances/history between me and a friend.

As a user, when viewing another users profile page, I want to view activity history between my user account and theirs.

As a user, when viewing another users profile page, I want an indicator for friendship, so that I can see if the user is friends with me or if I have to add them.

As a user, I want to be able to view total balances for who owes me and who I owe.

As a user, I want to be able to view a list of balances and users, per who owes me and who I owe, so that I can easily see who owes how much.

As a user, I want to be able to view a history of transactions.

As a user, I want to be able to view detailed information about individual transactions, so that I can keep track of what money was spent on.

As a user, I want to be able to modify transactions, so that I can fix entry errors.

As a user, I want to be able to create a new transaction.

As a user, I want to be able to settle transactions, so that when I pay/get paid I can clear pending transactions.

As a user, I want to be able to write multiple new transactions at once, so that when I enter in a group split it is easier.

As a user, I want to be able to attach/edit notes to each transaction for context.

As a user, I would like to be able to attach a receipt to my transactions, so that I can review details of the transaction.

As a user I would like to be able to view all my friends/contacts in a list.

As a user, For whatever reason, I want to be able to delete my account from the app

As a user, I want to be able to recover my account if I accidentally delete my profile or forget my username/password.

As a user, I want to be able to receive and view notifications when I receive payment or a message.

As a user, I want to only be able to send create new transactions between friends, so that I don't get splits from unknown people.

As a user, I want to be able to adjust my notifications preferences, so that I can choose when my friends are reminded about what is owed.
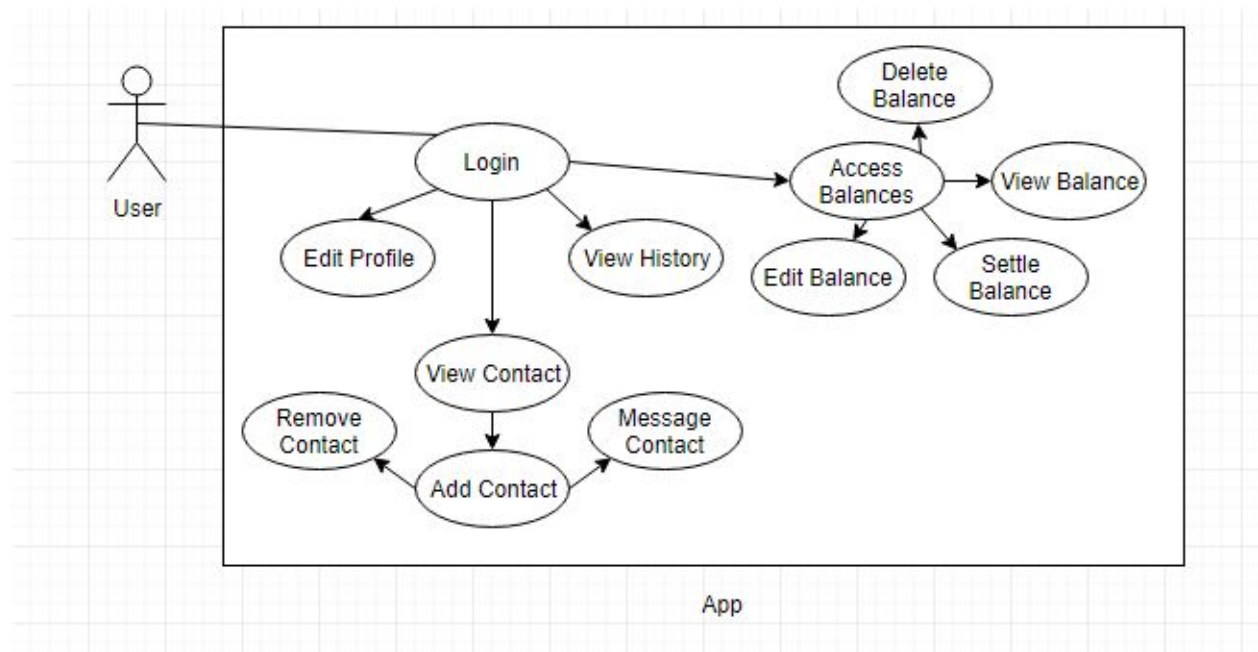
Functional Requirements (Prioritized/Ranked most important to least):
Users should be able to create and login with an account.
There should be account recovery options.
User accounts should have a profile page.
User should be able to select transactions to view in more detail.
User should be able to add/remove other users and view friendship status.
User must have a $0 balance before they can remove a friend.
User should be able to add pending payments/loans to contacts in their friends list.
User should be able to add multiple pending payments/loans at the same time.
User should be able to modify an existing pending transaction.
Profile pages should state all elements of user account.
Profile pages should show transaction history between themselves and the user viewing their page.
Display contact email on user profile page.
Display balances on profile page between two users.
Display transaction history on profile page between two users.
Users can only send splits between themselves and their friends.
Transactions should have a title/descriptor, notes, photo receipt, amount owed, and two users (Borrower/Lender).
Camera/photo gallery access to upload bills and attach them with correlating transactions.
Users can adjust notification preferences for reminding their friends.
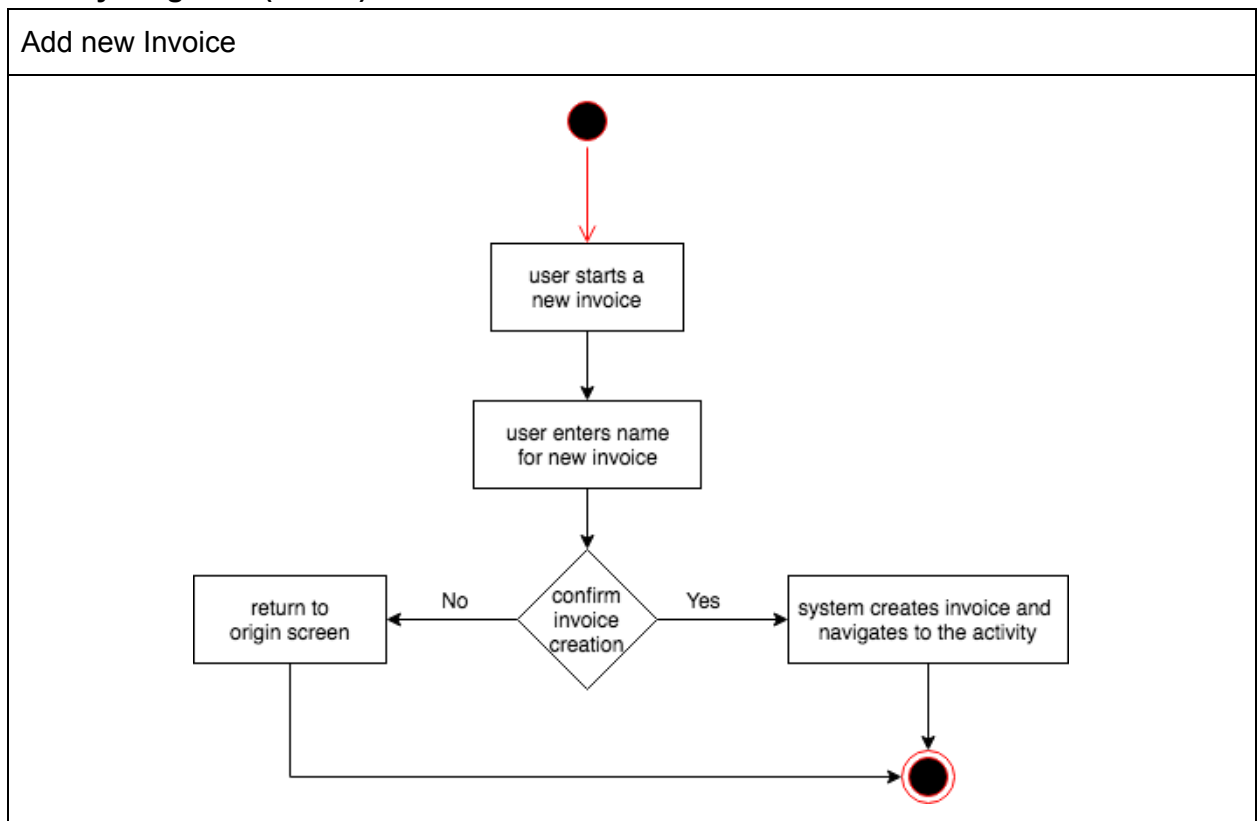Users should be able to adjust notification reminder preferences.

Non-Functional Requirements (Prioritized/Ranked most important to least):
Internet connection required.
System shall run on android devices.
System will be programmed mainly in Java.
System shall be available 24/7.
Users shall authenticate themselves with their user accounts.
The application should communicate with a DAO (data access object) to communicate with the database.
Password should be stored securely hashed and salted on database.
Data from the users and transactions will be stored on tables on SQL database.
Permissions for camera/photo gallery to save receipts.
UI shall be simple and clean.
Minimal delay between sending and receiving messages to contacts.
Minimal update delay when a balance has been modified in any way between the user who made the modification and all of their contacts that would be affected by said change.
Application has few features for less endpoints to be created and overall increases scalability, reliability and maintainability.
Users should be able to switch between tabs located at the bottom of the screen.

Use Case Model (Visual):

Activity Diagrams (Visual):

Add new Invoice

Add new Charge to Invoice

add new charge

↓

enter charge value

↓

add receipt image

→ Yes → choose file source

choose file source → local storage → select file to upload

take a picture → take a picture

take a picture → confirm selection

select file to upload → confirm selection

confirm selection → system inserts the image, and adds it to the charge

system inserts the image, and adds it to the charge → add friends to charge

add receipt image → No → add friends to charge

↓

system increments charge on invoice, selected friend's pending payments, and current user's loans

↓

Return to invoice view

↓