Requirement & Analysis Document

1. Introduction

1.1 Purpose of application

The project aims to create a personal finance application that serves the purpose of handling debt between participants in a certain group. The application users should be able to create accounts in the application and add other users into groups. The application simplifies debt between the participants in each group.

1.2 General characteristics of application

- The application is an android app written in Java.
- The application simplifies debt by subtracting, adding, and distributing debts between users in a group.
- Users in a group should be able to add new debts/transactions in the group and select other users to which the debt applies.
- Users should be able to see a sorted log of all transactions.
- Users should be able to add other users (by phone number) to a group.
- Users should be able to create an account with a name and phone number.

1.3 Scope of application

- No actual financial value will be transferred through the application. It will resemble a simplified ledger for personal debt between participants.
- The application will be limited to an offline prototype only. Users will be simulated and retrieve data from the local storage of the device.

1.4 Objectives and success criteria of the project

1. Users should be able to create an account.
    a. With a name, a phone number, and a password.
2. Users should be able to create a group and add other people to the group.
    a. Group has a name.
    b. Users can be added to the group by phone number.
3. Users should be able to add debt to others.
    a. Either individually or collectively as a group.
4. It should be possible to see the most recent transactions.

2. Requirements

2.1 Backlog

2.1.X US00x, [Title]

1. Implemented?
   - No
2. Description
   - As a [type of user], I want to [] so I can [].
3. Acceptance criteria
   - Functional
     i.
   - Non-functional
     i. Accessibility
     ii. Portability
     iii. Monitorability
     iv. Personalization
     v. Performance
     vi. Maintainability
     vii. Authorization
     viii. Localization
     ix. Testability

2.1.1 US001, Show debts for specific groups.

4. Implemented?
   - No
5. Description
   - As a user, I want to see all the added debts so I can understand my debt-situation.
6. Acceptance criteria
   - Functional
     i. There exists a button to reach the View for the log of debts.
     ii. There exists a View for the log of debts.
     iii. The debts must be logged somewhere to be able to display them.
     iv. The debts are sorted by date and viewable.
     v. There exists a way to exit the View.
     vi. There exists a way to get more information about a specific debt.
   - Non-functional
     i. Accessibility
        1.

2.1.2 US002, Create account

7. Implemented?
    - No
8. Description
    - As a user, I want to be able to create an account so I can use the application.
9. Acceptance criteria
    - Functional
        i. If the user is not logged in, the user is supplied with a button that leads the user to create an account (or login).
        ii. The user is prompted to enter Phone number, Name and Password on account creation.
        iii. The user can complete the account creation and the information is saved/stored and an account is created.
    - Non-functional
        i. Accessibility
        ii. Portability
        iii. Monitorability
        iv. Personalization
        v. Performance
        vi. Maintainability
        vii. Authorization
        viii. Localization
        ix. Testability

2.1.3 US001, Log in

10. Implemented?
    - No
11. Description
    - As a registered user, I want to log in so I can access my account and use the application.
12. Acceptance criteria
    - Functional
        i. If the user is not logged in, the user is supplied with a button to access a login page. (Or create an account).
        ii. On the login page, a user is prompted to enter the Phone number and password and can subsequently log in.
        iii. There exists a way to log out of the account.
    - Non-functional
        i. Accessibility
        ii. Portability
        iii. Monitorability
        iv. Personalization
        v. Performance

        vi. Maintainability
        vii. Authorization
        viii. Localization
        ix. Testability

2.1.4 US001, Dashboard

13. Implemented?
    - No
14. Description
    - As a [type of user], I want to [] so I can [].
15. Acceptance criteria
    - Functional
        i.
    - Non-functional
        i. Accessibility
        ii. Portability
        iii. Monitorability
        iv. Personalization
        v. Performance
        vi. Maintainability
        vii. Authorization
        viii. Localization
        ix. Testability

2.1.5 US001, Stream of Feed

16. Implemented?
    - No
17. Description
    - As a [type of user], I want to [] so I can [].
18. Acceptance criteria
    - Functional
        i.
    - Non-functional
        i. Accessibility
        ii. Portability
        iii. Monitorability
        iv. Personalization
        v. Performance
        vi. Maintainability
        vii. Authorization
        viii. Localization
        ix. Testability

2.1.6 US001, Add personal contacts

19. Implemented?
    - No
20. Description
    - As a user with an account, I want to be able to add personal contacts so that I can save their information and quickly access them in groups..
21. Acceptance criteria
    - Functional
      i. There exists a contacts view which can be accessed.
      ii. On the contacts-page all added contacts are listed.
      iii. On the contacts page there exists a way to add a new contact.
      iv.
    - Non-functional
      i. Accessibility
      ii. Portability
      iii. Monitorability
      iv. Personalization
      v. Performance
      vi. Maintainability
      vii. Authorization
      viii. Localization
      ix. Testability

2.1.7 US001, Create groups

22. Implemented?
    - No
23. Description
    - As a [type of user], I want to [] so I can [].
24. Acceptance criteria
    - Functional
      i.
    - Non-functional
      i. Accessibility
      ii. Portability
      iii. Monitorability
      iv. Personalization
      v. Performance
      vi. Maintainability
      vii. Authorization
      viii. Localization
      ix. Testability

2.1.8 US001, Add users to groups

25. Implemented?
    - No
26. Description
    - As a [type of user], I want to [] so I can [].
27. Acceptance criteria
    - Functional
        i.
    - Non-functional
        i. Accessibility
        ii. Portability
        iii. Monitorability
        iv. Personalization
        v. Performance
        vi. Maintainability
        vii. Authorization
        viii. Localization
        ix. Testability

2.1.9 US001, Add debt to user(s) in group

1. Implemented?
    - No.
2. Description
    - As a user, I want to add debt to one or more users in my group so that I can remember it in the future.
3. Acceptance criteria
    - Functional
        i. There is a button in the group view that navigates to another view where the user can add debts to other user(s).
        ii. The user can add multiple users to add debt to in the debt view.
        iii. The user can specify the total amount of debt in the debt view.
        iv. The user can choose to split the debt to all users selected.
        v. The user can send the debt specification to the group/users selected.
    - Non-functional
        i. Accessibility
        ii. Portability
        iii. Monitorability
        iv. Personalization
        v. Performance
        vi. Maintainability
        vii. Authorization
        viii. Localization
        ix. Testability

2.1.10 US001, Debt simplifier

4. Implemented?
   - No.
5. Description
   - As a user, I want to be to
6. Acceptance criteria
   - Functional
     i.
   - Non-functional
     i. Accessibility
     ii. Portability
     iii. Monitorability
     iv. Personalization
     v. Performance
     vi. Maintainability
     vii. Authorization
     viii. Localization
     ix. Testability

2.1.10 US001, My profile page

1. Implemented?
   - No
2. Description
   - As a user, I want to be able to see and edit my profile to keep it updated.
3. Acceptance criteria
   - Functional
     i. Can I see my name on my profile?
     ii. Can I see my picture on my profile?
     iii. Can I see a biography in my profile?
     iv. Can I edit the information on my profile page?
        1. Name
        2. Image
        3. Biography
   - Non-functional
     i. Monitorability
        1. Can I always see my name on my profile page?
        2. Can I always see my image on my profile page?
        3. Can I always see my biography on my profile page?
        4. Can I easily change to edit-mode on my profile page?
     ii. Security
        1. If text fields are left empty, will I not be able to save the edited information then?
        2. Are special signs allowed?

2.2 Definition of Done

- The code is peer-reviewed.
- Task has met the acceptance criteria.
- Code has passed JUnit tests.
- The application builds and compiles.
- Code is documented (JavaDoc).
- Merged into master-branch.
- Updated UML.

2.3 GUI

- Sketch

3. Domain model