GitHub Username: arbuleac

Loan calculator

Description

Did you ever want to buy a house or a car? Did not know how much would it actually cost when you opt in for a bank loan? Loan calculator will help you calculate the monthly payments, the interest paid and other metrics for your needs.
Loan calculator will help you schedule your payments and try to minimize any interests you pay.

Intended User

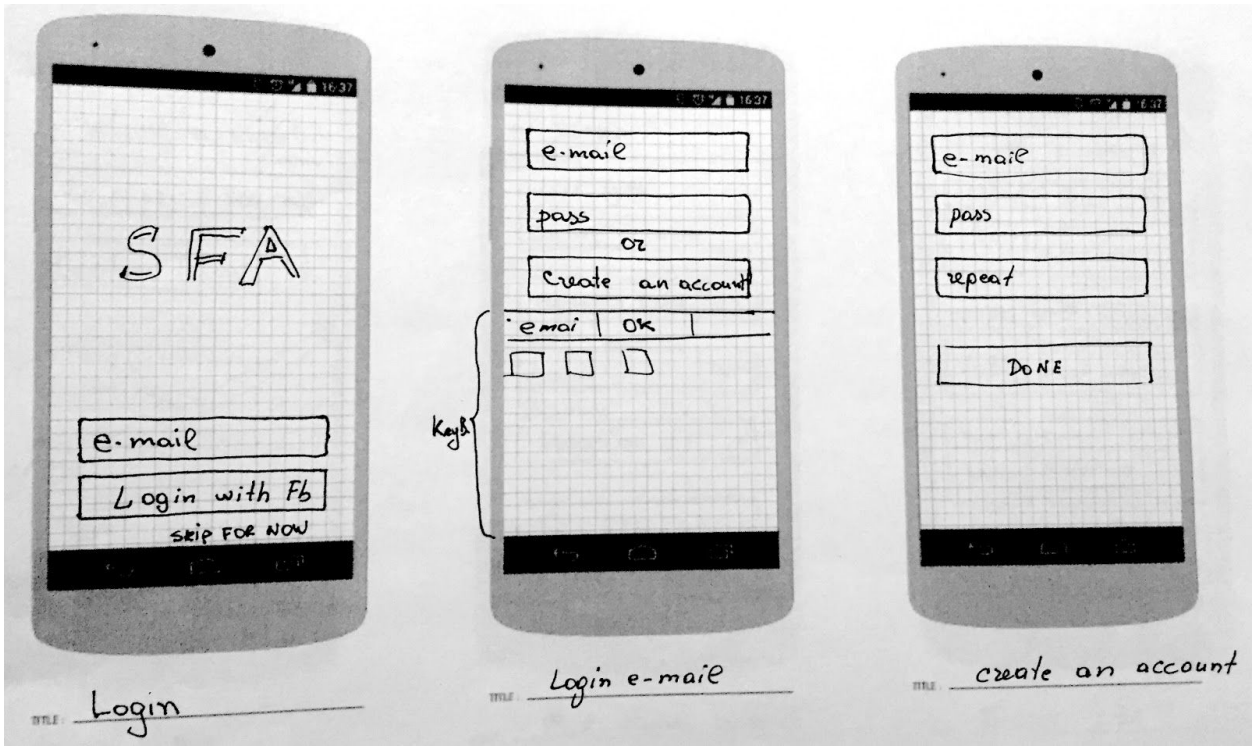
Young couples or established families, businessmen and anyone willing to get a loan in a local Romanian bank.

Features

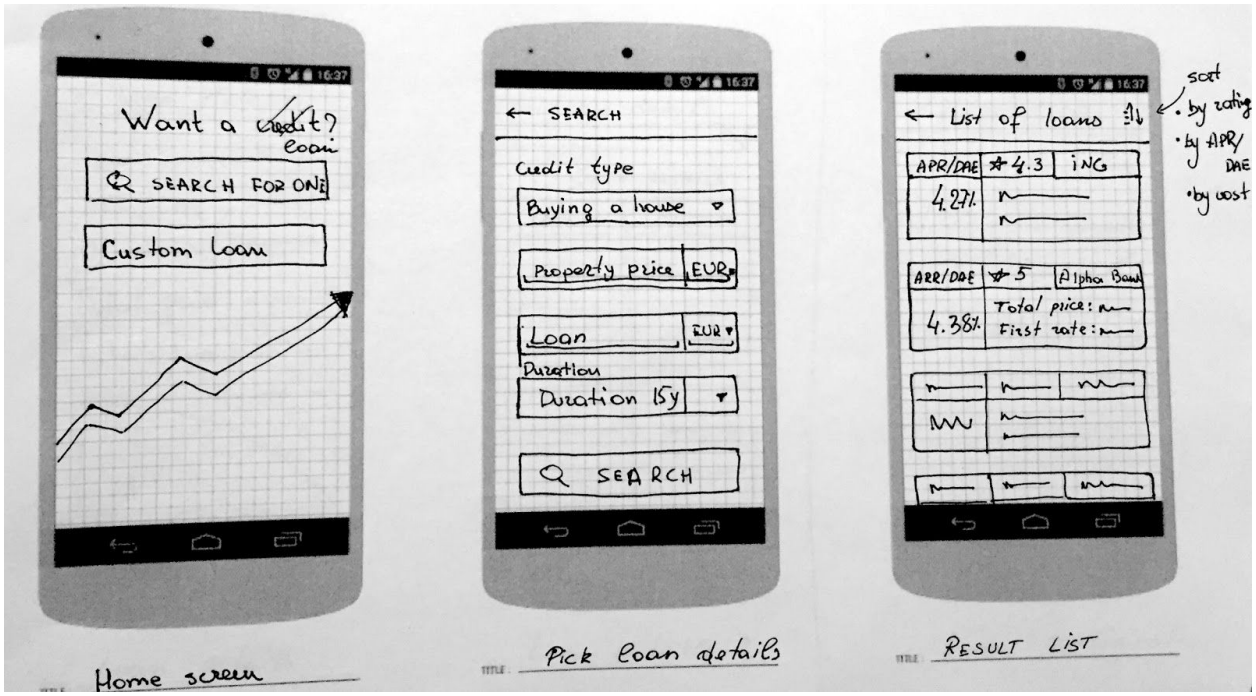
- Account manager
    - Create account using your email address
    - Login with email
    - Login with facebook
    - Anonymous account
- Loan tool
    - Mortgage loan calculator
    - Custom loan calculator
    - ~~Loan trends (v2) - (how do specific loan evoluate)~~
    - ~~Personal needs loan~~
    - ~~Auto loan calculator~~
    - ~~Auto leasing~~
    - ~~Credit card~~
    - ~~Loans for small and medium companies~~
- ~~Rent vs Buy (v2)~~
- ~~Economy tool (v2)~~
- ~~Investment tool (v2)~~
    - ~~Invest in real estate~~
    - ~~Investment fund analysis~~
    - ~~Investment funds list~~
    - ~~Investment funds rates~~

User Interface Mocks

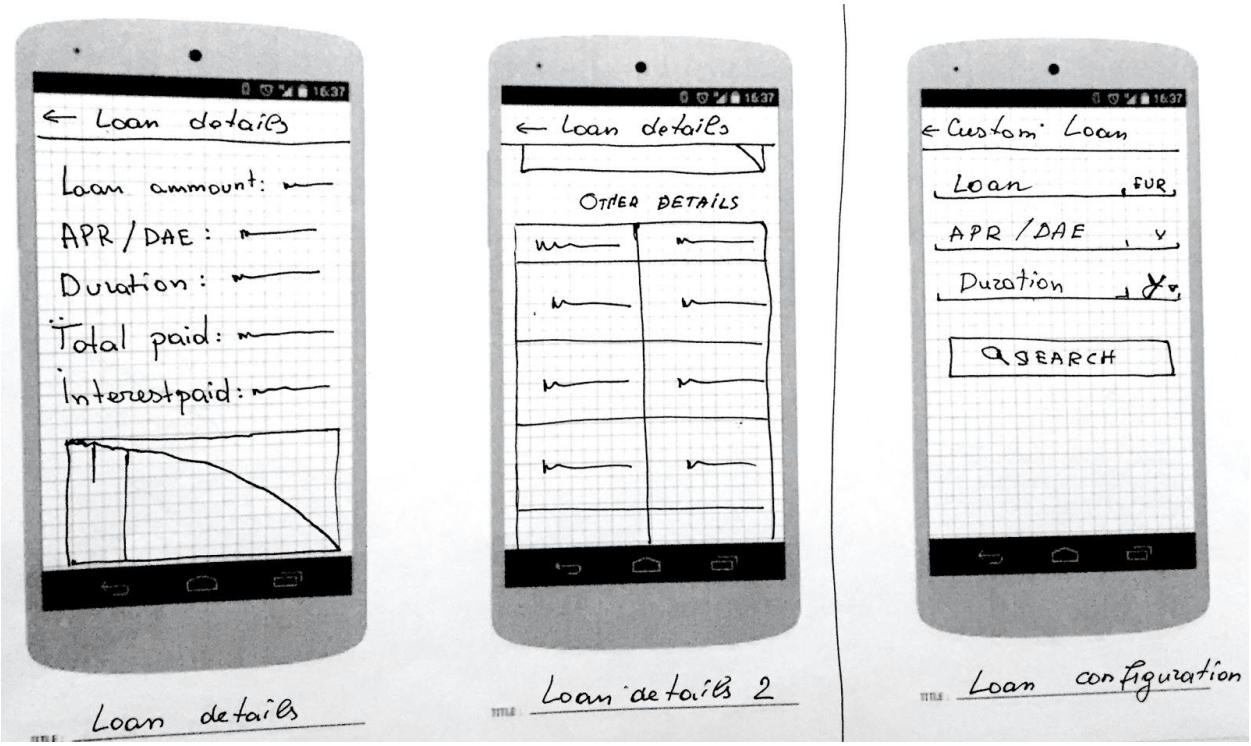
Screen Login/Login with email/Create account



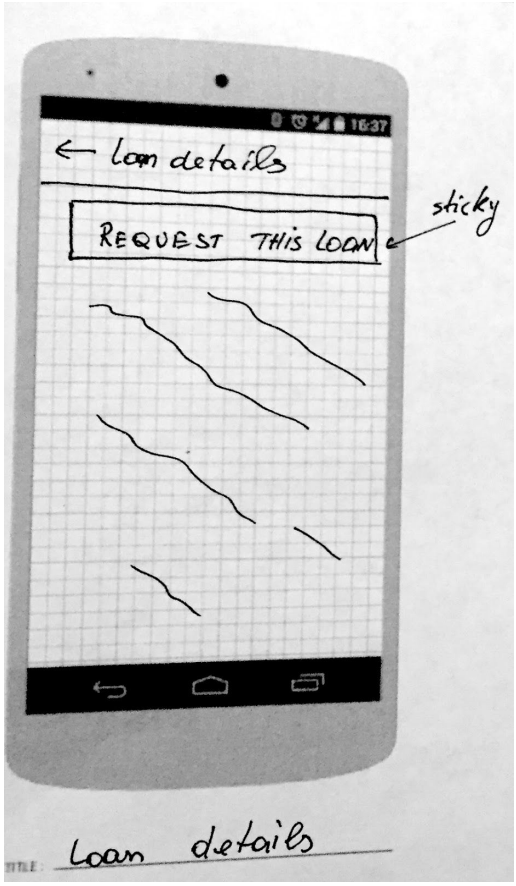
Screen Home/Search for loan/Result list



Screen Loan details/Loan details 2/Custom loan



Screen Loan detail - sticky CTA (call to action)



Key Considerations

How will your app handle data persistence?

Data persistence will be handled using Realm. The data about local banks will be retrieved from server side and stored in a local database. A user will be able to save any custom loan preferences and this will be handled within the local database as well.

Describe any corner cases in the UX.

No edge case scenarios. Linear navigation within the application is intended.

Describe any libraries you'll be using and share your reasoning for including them.

1. Glide - Image loading. Alternatively, use Picasso

2. Retrofit - Network calls, Http caching
3. Realm - Database layer implementation. Alternative to ContentProviders and SQLite
4. Android support library - backward compatibility
5. Android design support library - design backward compatibility, material widgets
6. Retrolambda (optional) - easy lambda expressions
7. RxAndroid (optional) - reactive programming
8. Kotlin (optional) - develop the application using Kotlin programming language
9. Dagger - dependency injection

Next Steps Backend: Required Tasks

Task 1: Project Setup

1. Create a remote (git) github/bitbucket repository
2. Install the latest Node.js / mongoDB versions
3. Create a new project structure

Task 2: Implement RESTful API

1. Implement [POST login with facebook]
2. Implement [POST login with email]
3. Implement [POST create an account]
4. Implement [GET loan list]
5. Implement [GET loan details]
6. Implement [GET custom loan details]
7. Implement [POST request a loan]

Next Steps Android: Required Tasks

Task 1: Project Setup

1. Create a remote (git) github/bitbucket repository
2. Make sure latest ADT are installed and latest Android Studio IDE is installed
3. Update all the ADT components if any of them were previously installed and are outdated now
4. Create a new Android Gradle project in Android studio
5. Add all gradle dependencies to libraries you will use in this project

Task 2: Implement UI for Each Activity and Fragment using Mock data and dependency injection frameworks

1. Build UI for Login screen
2. Build UI for Sign-up screen
3. Build UI for Home screen
    - Search for loan
    - Custom loan
4. Build UI for Search for loan
5. Build UI for Result list
6. Build UI for Loan details
    - sticky CTA contact button
    - common loan details
    - interest vs main graph
    - loan calculator and loan rates table
7. Build Ui for Custom loan
    - bind to result list and loan details

Task 3: Develop network access and persistence layer

1. Implement network layer using retrofit
2. Add persistence layer with realm
3. Implement login with facebook
4. Implement login with email
5. Implement create account
6. Implement search for loan
7. Implement get loan details
8. Implement get custom loan details
9. Implement request a loan