

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Authentication screen](#)

[Error screen](#)

[Incoming call notification](#)

[End of call notification](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 0: Create web server](#)

[Task 1: Create authentication activity](#)

[Task 2: Handle permission for Android M](#)

[Task 3: Manage calls](#)

[Task 4: Offline mode](#)

GitHub Username: thomasthiebaud

Quiet

Description

Are you fed up to receive useless or dangerous calls ? Quiet is a community black list which will warn you if an incoming call is a scam or an ad.

Intended User

This application targets everyone.

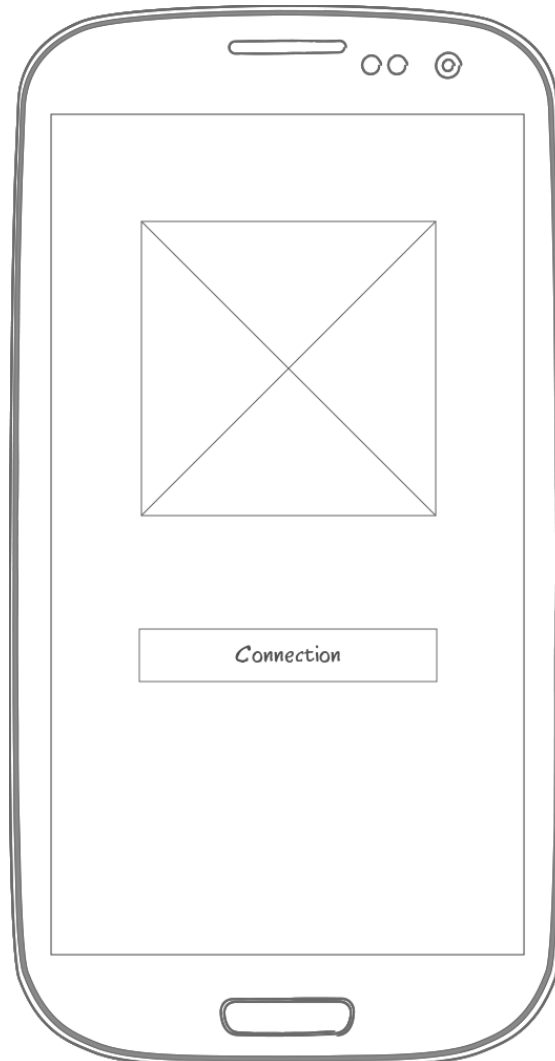
Features

The main features of the app are :

- Manage calls
- Display notifications
- Get data from remote services

User Interface Mocks

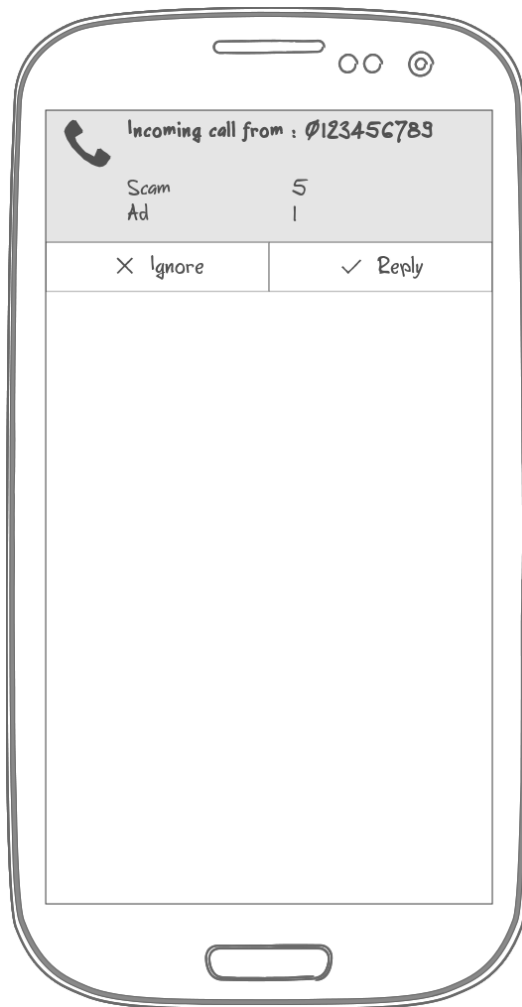
Authentication screen



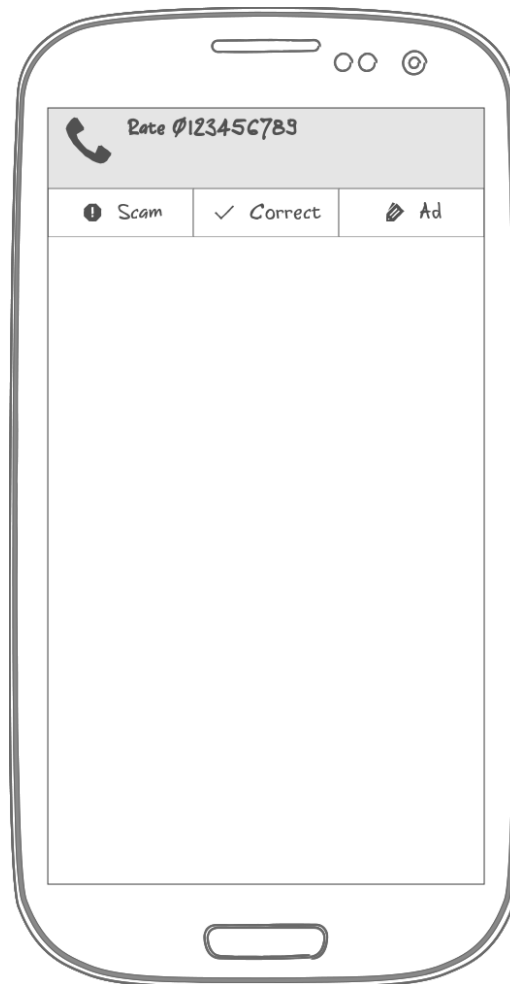
Error screen



Incoming call notification



End of call notification



Key Considerations

How will your app handle data persistence?

The application will provide its own content provider in order to enable offline mode and retrieve data from the server.

Describe any corner cases in the UX.

The application is based on notifications so there is no need for screen except the first time when the user needs to provide its account and permissions. In a later version, more settings may be added and a screen will come for it.

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

The app will use :

- Realm for the data persistency
- Retrofit for web services calls
- Google identity service for g+ authentication

Next Steps: Required Tasks

Task 0: Create web server

This task is not related to android development, but must be done before starting it.

TODO :

- Initialise NodeJS project
- Initialise MongoDB database
- Handle google authentication
- Create web services in order to report a phone number and check if a number is already reported
- Create associated tests

Task 1: Create authentication activity

When the app is launched for the first time, the user see the authentication page in order to provide its credentials.

The account will be used to identify which user reported a number.

TODO :

- Create activity
- Setup google identity toolkit.
- Handle errors. The user must be redirected to the error page if the authentication is wrong.

IMPROVEMENT IDEA :

- Allow Facebook and Twitter authentication too

Task 2: Handle permission for Android M

Permissions can be asked at runtime for android M users.

TODO:

- Ask for permissions at runtime.
- Handle errors. The user must be redirected to the error page if a needed permission is not set.

IMPROVEMENT IDEA :

- Create a custom view in order to ask for permissions
- Explain for each permissions why they are needed

Task 3: Manage calls

This is the core feature of the application. If an unknown call is incoming, the app must check if this number is already reported. If so it must display a notification to the user with the details of the reports. At the end of the call, another notification must pop in order to ask if the user wants to report the call.

TODO :

- Detect unknown incoming calls
- Retrieve data from the server (using retrofit)
- Display “incoming” notification
- Detect the end of the call
- Display “End of the call” notification

IMPROVEMENT IDEA :

- Add settings to automatically refuse a call if it is already reported

Task 4: Offline mode

The offline mode will use a local database in order to save reported numbers which already tried to contact the user. Moreover, if the user reports a number while offline, it must be synchronised with the server later.

TODO :

- Add custom content provider for local storage.
- Setup realm

IMPROVEMENT IDEA :

- Allow users to download presets for offline usage