Description
Intended User
Features
User Interface Mocks
Screen 1
Screen 2
Screen 3
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.
Describe how you will implement Google Play Services.
Next Steps: Required Tasks
Task 1: Project Setup
Task 2: Implement UI for Each Activity and Fragment
Task 3: Implement and test Firebase libraries

GitHub Username: DanzigD

RageStats

Making the app for a friend who work with emotionally unstable persons and need an app to document when an emotional outburst occurs.

An app for those who work with emotionally unstable persons and need to keep notes and stats.

Intended User

A collaborative tool for those who assist emotionally unstable persons and need to keep track of outbursts and their possible causes.

Features

- Keeps statistics daily and over time
- Keeps notes sorted by tags and time

- Quickly capture students currents emotional state
- Allow several teachers to view and edit the same group
- Track causes over time by tagging occurrences.
- Widget to quickly take dokument observations

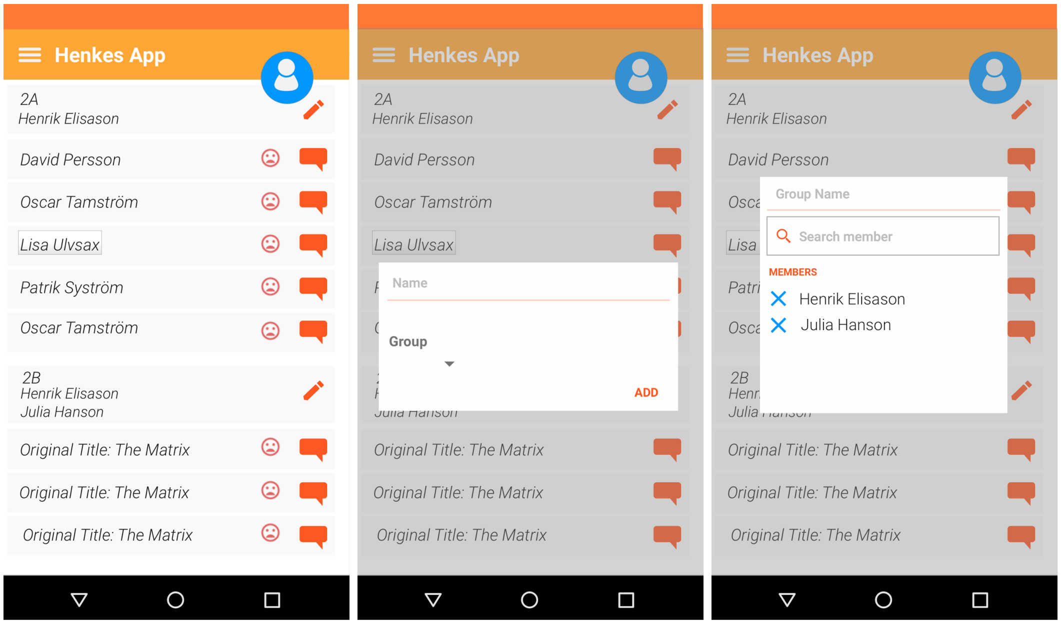
MainActivity mock

The main screen consist of a list of students, categorised in different groups. All list items has different quick-action buttons where user can take notes, and tap the smiley face to change the smiley into a mood that reflects the student's current emotions.

The dialog in pic 2 appears when user tap the blue fab on top and is used for creating a new student to the group. The Dialog in pic 3, appears when the user tap the spinner, in the previous dialog, and choses the “create new group” item.

Tapping on one of the students list items will navigate the user to the StatisticsActivity.

Screen 1



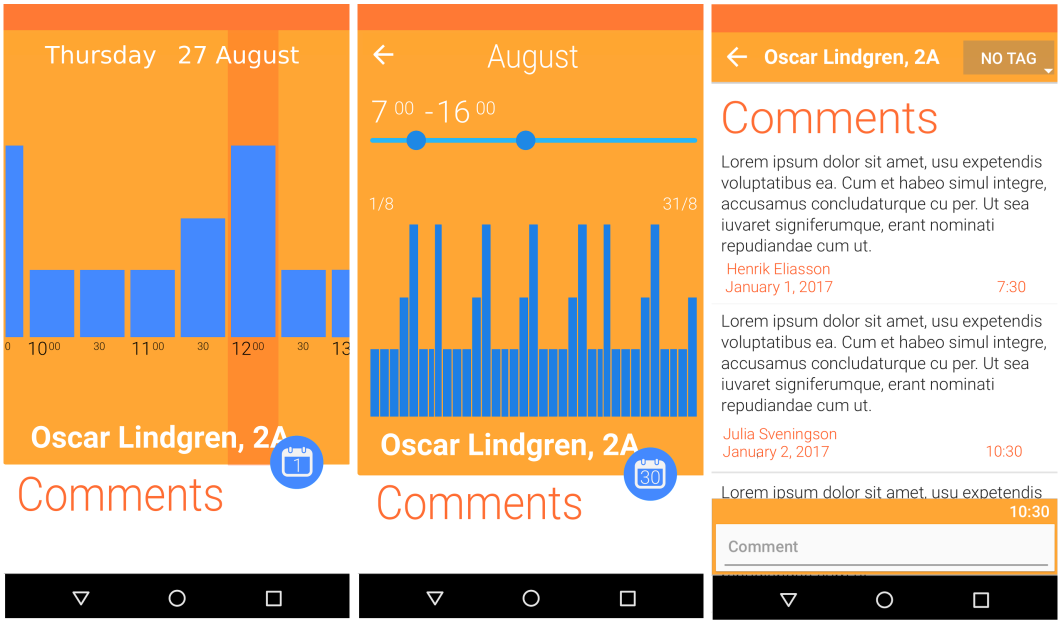
StatisticActivity mock

The second screen is where the user add and view statistics, and also adding and viewing comments.

The top of the layout is a fragment that, when user navigate down, move away in a parallax motion and reveal a list of comments. The fragment (DayStatistic Fragment) turn into another fragment (LongStatistic Fragment) when tap the FAB.

Day Statistic Fragment:
this fragment lets the user see and change statistics for that day and also let the user add notes for every hour.
Long Statistic Fragment:
This fragment shows statistics over a longer period of time.
Below the AppbarLayout appears a list of comments. These comments is ordered by time and by tags. The DayStatisticFragment has a comment button that let the user add comments for that hour and day and let the user change the tag for that comment.

Screen 2



Lorem ipsum dolor sit amet, usu expetendis voluptatibus ea. Cum et habeo simul integre, accusamus concludaturque cu per. Ut sea iuvaret signiferumque, erant nominati repudiandae cum ut.

Henrik Eliasson
January 1, 2017 7:30

Lorem ipsum dolor sit amet, usu expetendis voluptatibus ea. Cum et habeo simul integre, accusamus concludaturque cu per. Ut sea iuvaret signiferumque, erant nominati repudiandae cum ut.

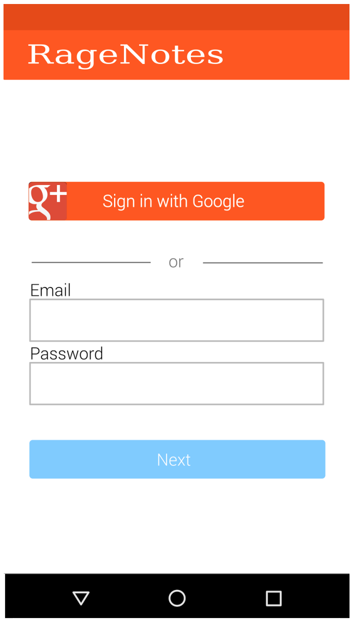
Julia Sveningson
January 2, 2017 10:30

Lorem ipsum dolor sit amet, usu expetendis

LogInActivity mock

RageStats is a collaborative tool and therefore allows users to sign in and add other users to their groups. This activity lets the user log with a single tap, or by adding email and password.

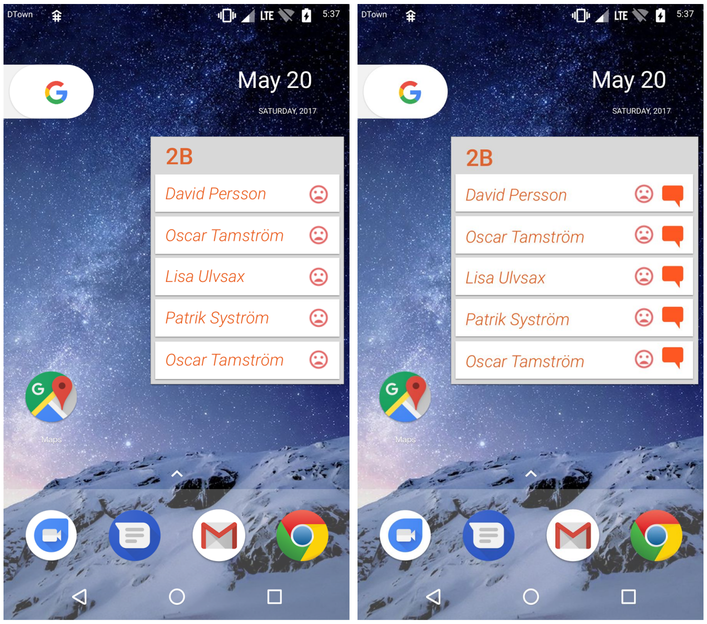
Screen 3



Widget mock

The widget works the same as the list in MainActivity and tapping on a card navigates the user to StatisticsActivity. The extra buttons appear when the widget is wide enough.

Widget



2B

David Persson ☹
Oscar Tamström ☹
Lisa Ulvsax ☹
Patrik Syström ☹
Oscar Tamström ☹

2B

David Persson ☹ 🟥
Oscar Tamström ☹ 🟥
Lisa Ulvsax ☹ 🟥
Patrik Syström ☹ 🟥
Oscar Tamström ☹ 🟥

Key Considerations

How will your app handle data persistence?

I have read up on Firebase Realtime Database and find it awesome! I will use that for both local storage and sharing data between users.
I will schedule the app to sync every hour with Firebase JobDispatcher to save battery and performance.

Describe any corner cases in the UX.

I will have four activities MainActivity, LoginActivity, SignUp Activity and Statistics Activity. User navigate to LoginActivity by logging out from the menu in MainActivity or by simply not being logged in, and back to MainActivity by signing in.
To get from MainActivity to Statistics Activity the user tap an item in the list. Since the Statistics Activity contains two fragments the user can use both a floating action button and the back button to navigate between the two fragments.

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

Will use either GraphView library or MPAndroidChart for showing stats for longer time. I will try both and keep the one that has what I need.
http://www.android-graphview.org/showcase/
https://github.com/PhilJay/MPAndroidChart

Describe how you will implement Google Play Services.

I will use Firebase Cloud Messaging to sync the data for all users. For the login functionality I will use Firebase Authentication.
I will also show ads with Firebase AdMob in my Main activity.

Next Steps: Required Tasks

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

- Organize all needed data into objects with getters and setters.
- Create mock data for simple UI testing
- Adding colors and Strings to the resource directory

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- The blocks in the DayStatisticFragment will be modified Seekbars stacked in a Recyclerview. I will use the implementation from the link below to make the seekbar vertical and then customize the appearance. https://stackoverflow.com/questions/3333658/how-to-make-a-vertical-seekbar-in-android
- The graph in the LongStatisticFragment
- Build UI for StatisticsActivity
- Build UI for LoginActivity
- Extend all activities with AppCompat

Task 3: Implement and test Firebase libraries

- Implement and register Firebase Realtime Database
  - Change write/read permissions in the Realtime Database for all until I tested the app and implemented Authentication.
- Implement Firebase Jobdispatcher to regularly sync with the database
- Implement and register Firebase Authentication
  - Change write/read permissions in the Realtime Database to only group memebers.
- Implement and register Firebase AdMob
  - Reformating code to split between free and paid flavors and make changes in build.gradle files
  - Adding a banner in MainActivity

Todo: more trello posts here

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”