GitHub Username: yuro8

Deprecure

Description

The Deprecure app is created for giving a helping hand for sad and depressed people. The resources and tools are great way to start when you are in a bad mood. When you are anywhere, the Deprecure is always with you.

Intended User

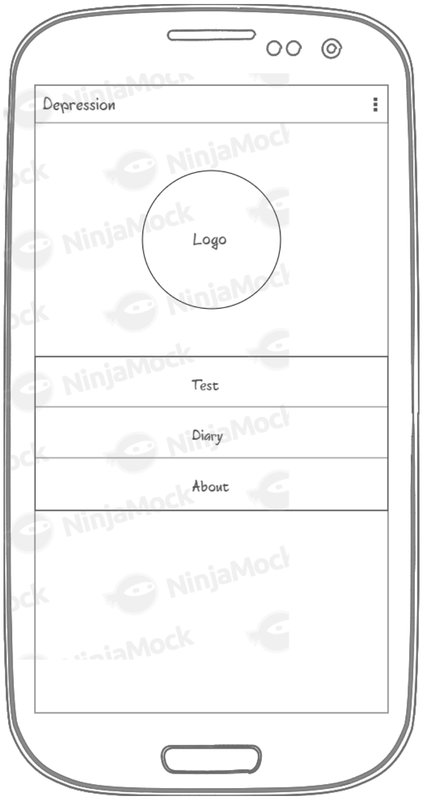
The app is aimed for depressed people and specialists looking for tools and resources how to overcome depression.

Features

- Written in Java
- Gradle version 4.6
- Android Studio version 3.0.1

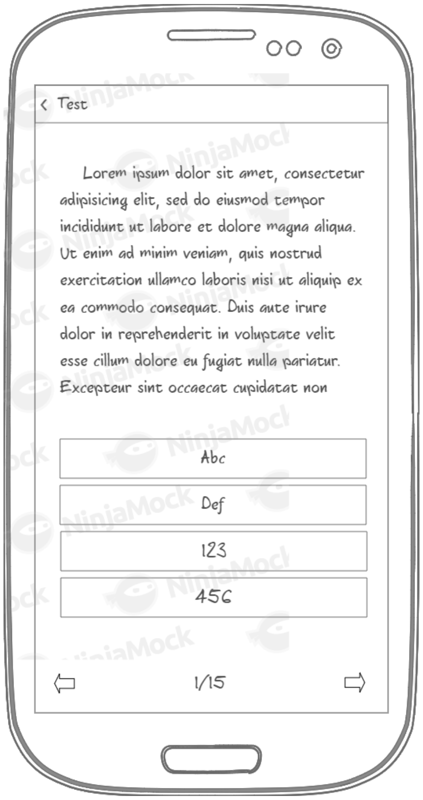
- Tracks user’s mood
- Diary
- Depression test

User Interface Mocks



Screen 1

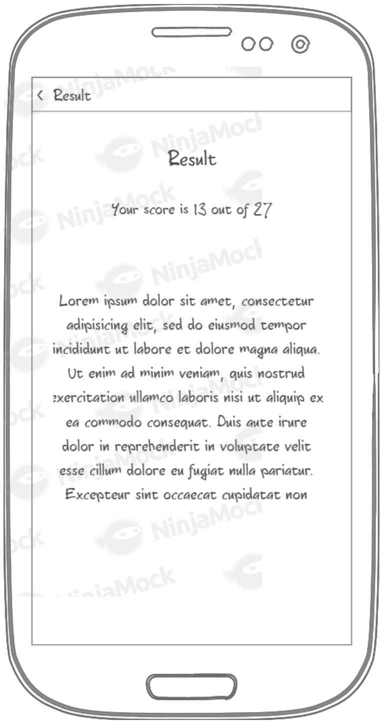
The main screen which shows menu with three items: Test, Diary and About. In an app bar is a menu button for logout.



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non

Screen 2

The test activity with PagerAdapter. Each page show a one question.



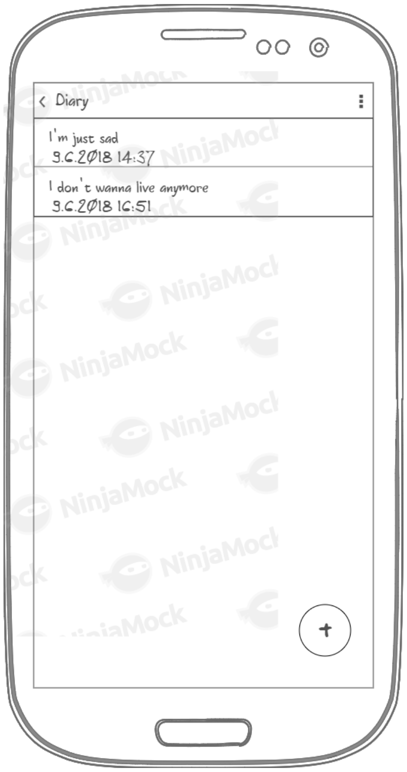
Result

Your score is 13 out of 27

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud xercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non

Screen 3

The fragment shows result from a test activity.

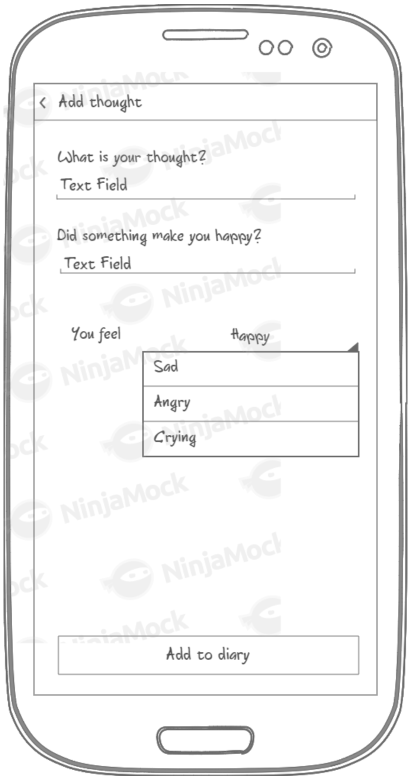


I'm just sad
9.6.2018 14:37

I don't wanna live anymore
9.6.2018 16:51

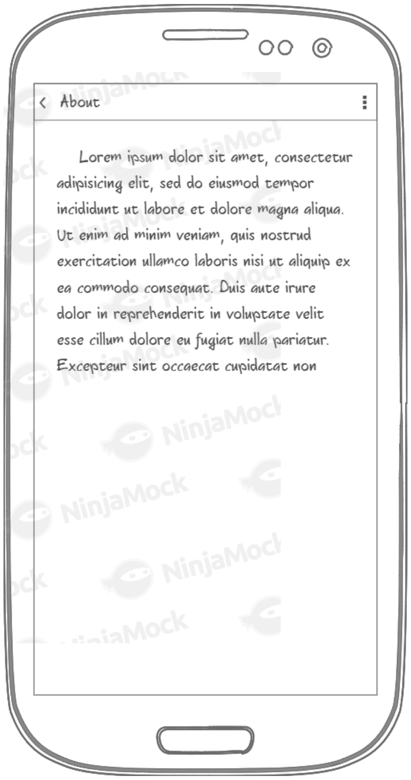
Screen 4

The diary activity with RecyclerView shows all items from Firebase.



Screen 5

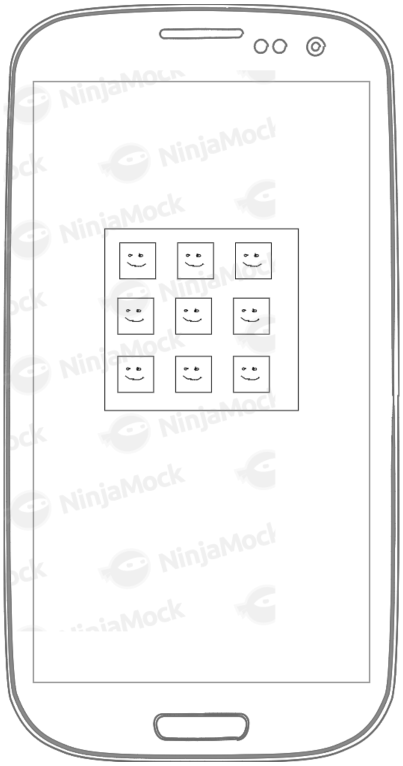
The fragment for adding items to a diary (Firebase).



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non

Screen 6

The about activity shows informations about the app.



Widget screen

The widget shows moods and user can track his mood from homescreen.

Key Considerations

How will your app handle data persistence?

I want to use Firebase because of offline data persistence support.

Describe any edge or corner cases in the UX.

- When will user complete the test, the result is showed in a result fragment. On back pressed is returned to the main.
- When there is no data communication, the data are fetched from offline Firestore DB, or is showed error
- If user won't answer to all questions in a test, user can't submit it and toast is showed

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

Butterknife(8.8.1) for binding views.
Firebase-auth(16.0.1) for authentication
Firebase-ui-auth(4.0.0) for authentication
Firebase-core(16.0.0) for firebase connection
Firebase-firestore(17.0.1) for data persistence

Describe how you will implement Google Play Services or other external services.

I will use Firebase Authentification with FirebaseUI.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Create main with menus
  - First we initialize Authentification
  - If is user logged in, we'll open main, else we'll show FirebaseUI

- Add test activity with fragment result screen
    - The data should be passed as Parcelable to fragment instance in ViewPager
- Add test result fragment
    - The estimated result is showed in a TextView
- Add diary activity
    - Firstly, we query data from Firestore
    - On FAB click is opened fragment for adding items
- Add fragment for adding item to diary
    - On adding complete are data written to Firestore and showed in RecyclerView
- Adding about activity with information about app

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
    - Layout with three buttons and logo
    - Added custom Toolbar with Log Out button
- Build UI for DiaryActivity
    - Diary activity is RecyclerView with FAB for adding items
- Build UI for AddToDiaryFragment
    - There are edittexts and spinner for user’s current mood
- Build UI for DepressionTestActivity
    - TestActivity is ViewPager with button answers
- Build UI for DepressionTestResultFragment
    - The result from the test will be showed as a text in fragment
- Build UI for AboutActivity
    - TextView with info about app
- Add resources to appropriate xml files and enable RTL layout switching
- Add content descriptions for accessibility

Task 3: Firestore setup

- Firestore:
    - “Users” is a collection of user ID documents. User ID Document points to a sub-collection “Diary” which stores documents about diary items
- Implementing writing to DB
- Implementing querying from DB
- Adding RecyclerView with data

Task 4: Authorization setup

- Adding FirebaseUI screen

  - Opened on Sign In click
- Adding authorization settings and logout button to menu

Task 5: Widget setup

- Adding GridView for smiles
- Adding onClick listeners on smiles
- Adding IntentService for updating diary