GitHub Username: **LaviniaDragunoi**

KeepIt

Description

KeepIt is your personal agenda that keeps everything together and helps you with your busy life. You can enjoy your life because **KeepIt** is keeping you organized.

Intended User

For people that want to keep everything together: daily schedule, meetings, birthdays, notes.

Features

List the main features of your app. For example:

- Widget for Home Screen with the daily schedule;
- Keeps all your daily schedule, meetings, birthdays and notes together in one Database
- In the daily schedule, the user has a summary of all her/his schedule, meetings, birthdays.
- Can check in the daily schedule if one of its items was achieved.

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1

Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Homepage for **KeepIt** app. Has a GridView with all four agenda entries.

- **About** – here is a short description of the app and is presented the possibility to upgrade to the paid version that is no advertising version
- **Images** – are clickable and it will open each specified Activity
- **FAB** - this button allows to add an event or note.

Screen 2

Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Provide descriptive text for each screen

Today is the activity that will provide:
the schedule for all day as a list of clickable items.

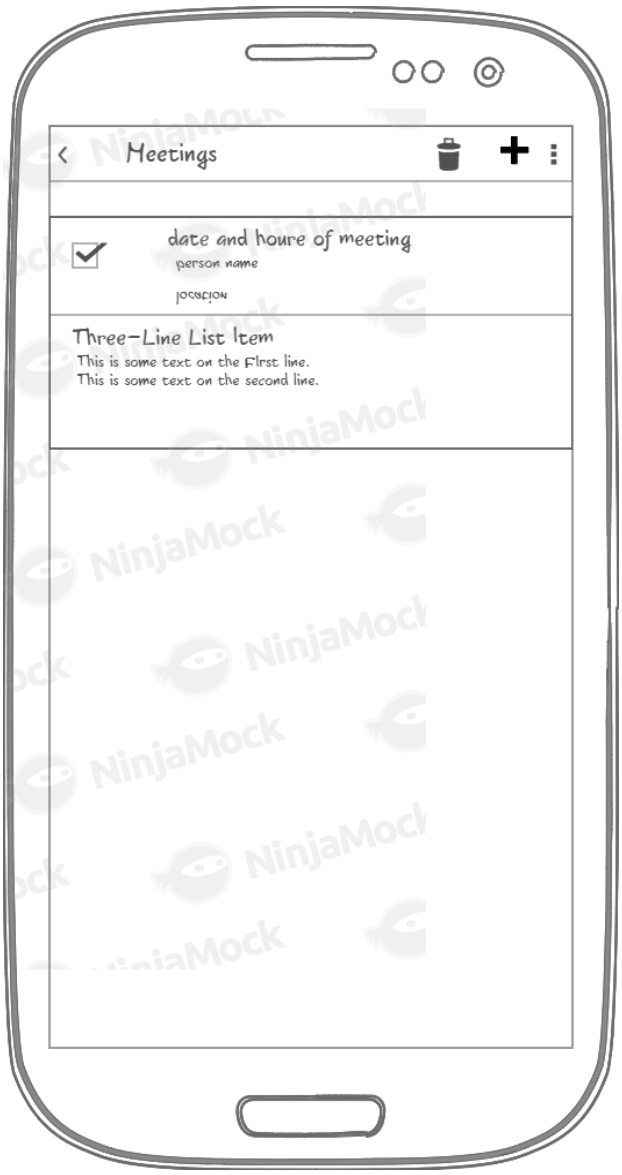
In the AppBar are two options to delete item/items of the list or to add – this will open an separated activity where the user can add or edit item for the today's list.
Menu that will have the path for **Home, Birthdays, Notes, Meetings**

Screen 3

Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Provide descriptive text for each screen

Add for Today 's is the activity that will guide the user to the adding's activities for each category of events.

Screen 4



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Provide descriptive text for each screen

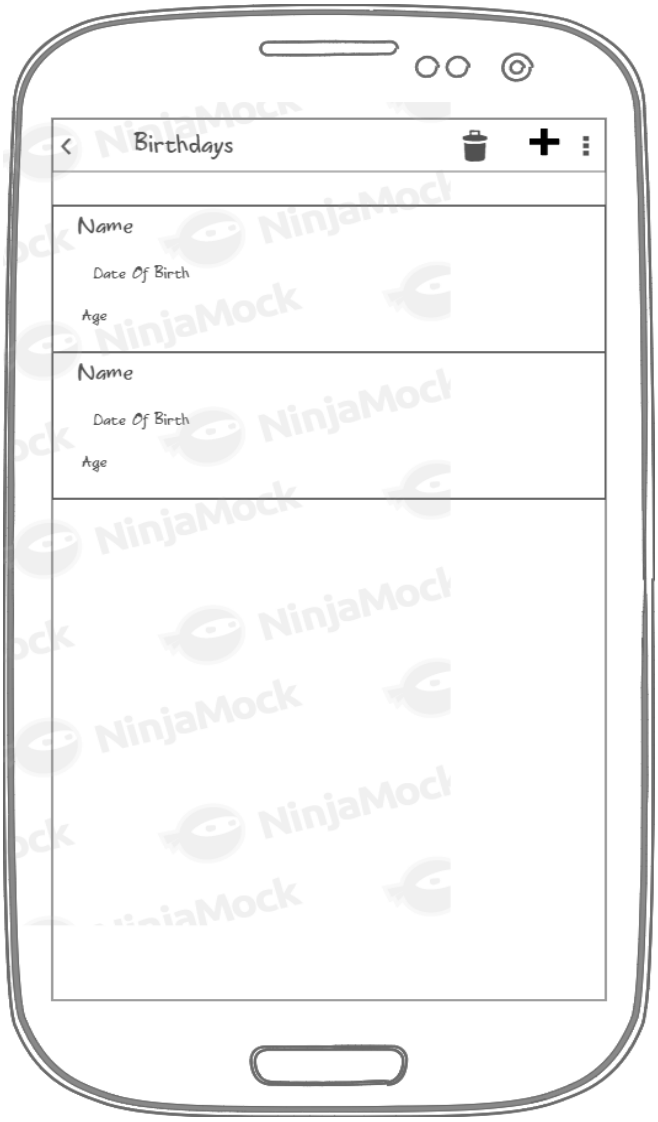
“ Meetings” is the activity that will display for the user all the meetings in a checkable list. There are in the AppBar the possibilities of deleting and adding as well, the last one is opening the Add Meeting Activity. The menu that will have the path for **Home, Today, Birthdays, Notes.**

Screen 5

Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Provide descriptive text for each screen

Add/Edit meeting is the activity that will offer the possibility to add or edit a meeting's items. In the AppBar are options for deleting(if is editing a meeting) and saving the new meeting or editing.

Screen 6



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Provide descriptive text for each screen

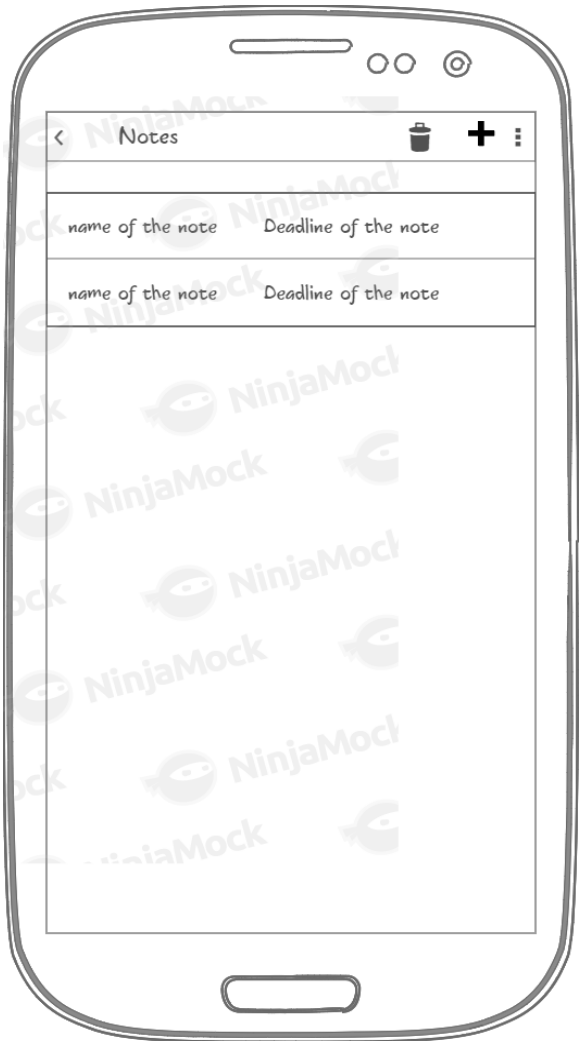
“Birthdays” is the activity that will display for the user a list with all the entered birthdays. There are in the AppBar the possibilities of deleting and adding as well, the last one is opening the Add Birthday Activity. The menu that will have the path for **Home, Today, Meetings, Notes.**

Screen 7

Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Provide descriptive text for each screen

Add/Edit birthday is the activity that will offer the possibility to add or edit a birthday's items. In the AppBar are options for deleting(if is editing a birthday) and saving the new birthday or editing.

Screen 8



Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Provide descriptive text for each screen

"**Notes"** is the activity that will display for the user a list with all the entered notes. There are in the AppBar the possibilities of deleting and adding as well, the last one is opening the Add Note Activity. The menu that will have the path for **Home, Today, Meetings, Birthdays.**

Screen 9

Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]
Provide descriptive text for each screen

Add/Edit note is the activity that will offer the possibility to add or edit a note's items. In the AppBar are options for deleting (if is editing a note) and saving the new note or editing.

Key Considerations

How will your app handle data persistence?

KeepIt will be handling data persistence using Room, LiveData and ViewModel

Describe any edge or corner cases in the UX.

- If the user has empty lists a message will be displayed “Take a note and **keep It** together”

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

- RecyclerView – v7:27.1.1 to display lists.
- Btterknife v8.8.1 to bind views
- Design 27.1.1 for visual impact
- Room 1.1.1 for storing in the DB
- LiveData and ViewModel 1.1.1 that sets between DB and UI

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

KeepIt uses GoogleAds with ads that will be displayed as bottom banners and GoogleAnalytics to observe user behavior.

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 project with empty activity.
- Write in build.gradle the libraries dependencies

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for main screen(Home)
- Build UI for AppBar menu
- Build UI for Today’s activity
- Build UI for Add/Edit Today’s activity
- Build UI for Meeting’s activity
- Build UI for Birthday’s activity
- Build UI for Note’s activity
- Build UI for Add/Edit Meeting’s activity
- Build UI for Add/Edit Birthday’s activity
- Build UI for Add/Edit Note’s activity

Task 3: Handle Error Cases

In each layout will be set a TextView and the app’s logo for each case of empty list.

Task 4: Handle database

- Implement Room elements(RoomDb, Entity, DAO)
- Create Repository.class that will handle the binding part between AAC and UI Components.
- Create ViewModel ‘s for each UI components that will require

Task 5: Testing

- Manual testing of the RoomDb implementation
- It will be used an external library for test and see database’s entries ('com.amitshekhar.android:debug-db:1.0.3')
- Manual testing of all UI components and handling errors(empty lists)

Task 6: Handle design

Making all design adjustments needed in order to make KeepIt app be aware of the principles of Material Design

Task 7: Implementing widget

Creating "Today’s" widget that will post on the device’s home screen the today’s schedule.

Common project requirements

- App is written solely in Java Programming Language
- App utilizes stable release versions of all libraries, Gradle (v3.1.3) Android Studio (v3.1.3)
- App keeps all strings in a string.xml file and enables RTL layout switching on all layouts.
- App validates all input from servers and users. If data does not exist or is in the wrong format, the app logs this fact and does not crash
- App includes support for accessibility including content descriptions.
- It use an AsyncTask when the user makes a search on the equivalences screen.

Submission Instructions

- After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
  - Make sure the PDF is named “**Capstone_Stage1.pdf**”
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”