

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[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: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: *viktorot*

Notefy

Description

Notefy allows you to organize your thoughts and day-to-day activities. Never forget and appointment or to buy milk from the supermarket.

Intended User

The application will be intended for everyone that is multitasking throughout the day.

Features

- Save and organize your notes
- Customize the look of your notes
- Filter by importance or category
- Automatic notifications
- Pin your notes, so you never forget a thing
- Synchronize your data between multiple devices

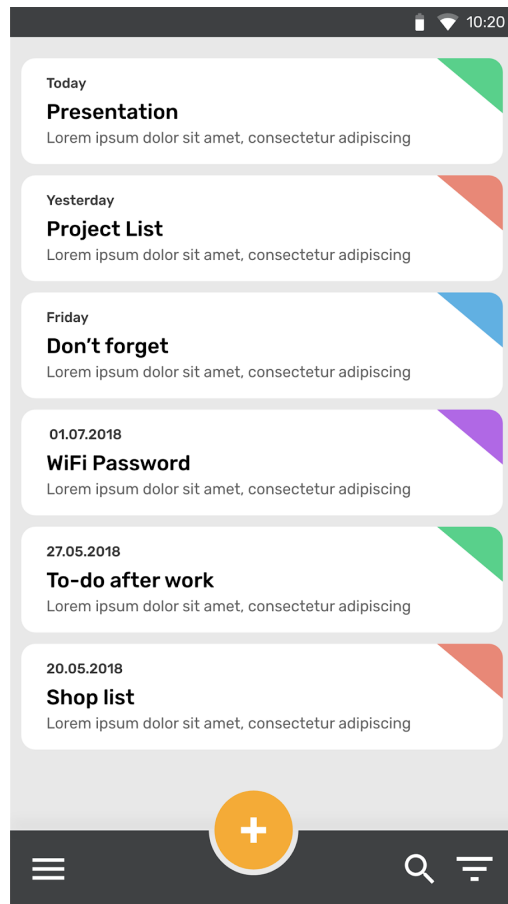
The application will be programmed on the Android platform using the Java programming language.

The Firebase push messaging will be used to notify the user of upcoming tasks/appointments.

All resources (strings, dimensions, styles) will be stored in their appropriate .xml files for reuse and easier refactoring.

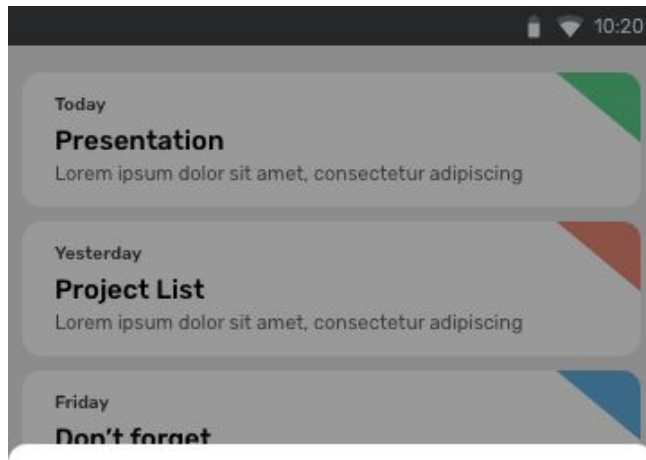
User Interface Mocks

Screen 1 - Notes list



Screen that shows a list of notes and allows the user to access, search and filter their notes.

Dialog - Filter notes



Filters

TAGS

- ☐ To-Do
- ☐ Project
- ☐ Work
- ☐ Personal
- ☐ Shopping
- ☐ Holiday

COLOURS

- ☒ Red
- ☐ Green
- ☐ Yellow
- ☐ Blue
- ☐ Orange
- ☐ Pink
- ☐ Purple

Dialog that allows the user to filter notes.

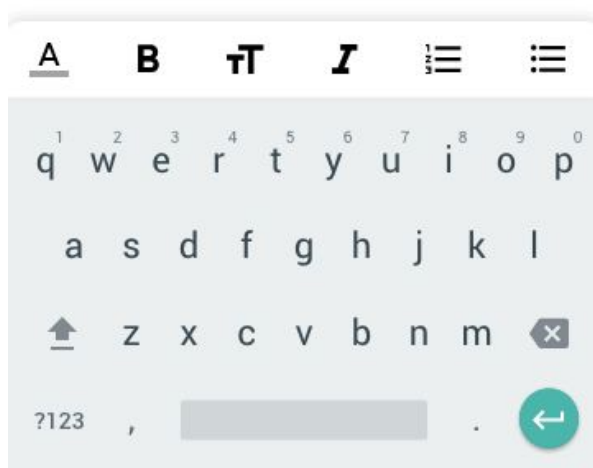
Screen - Create/edit note

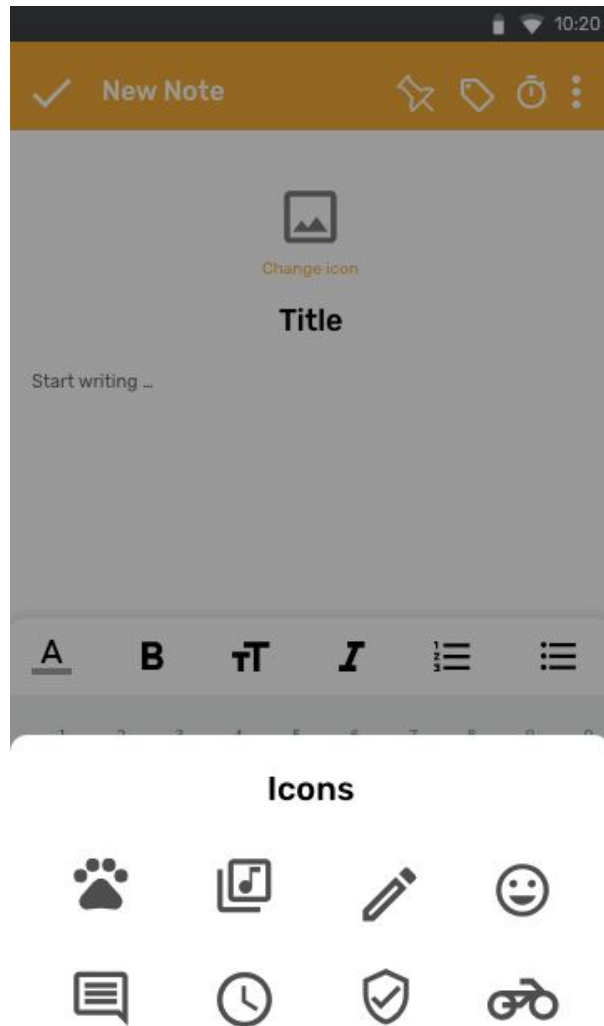


Change icon

Title

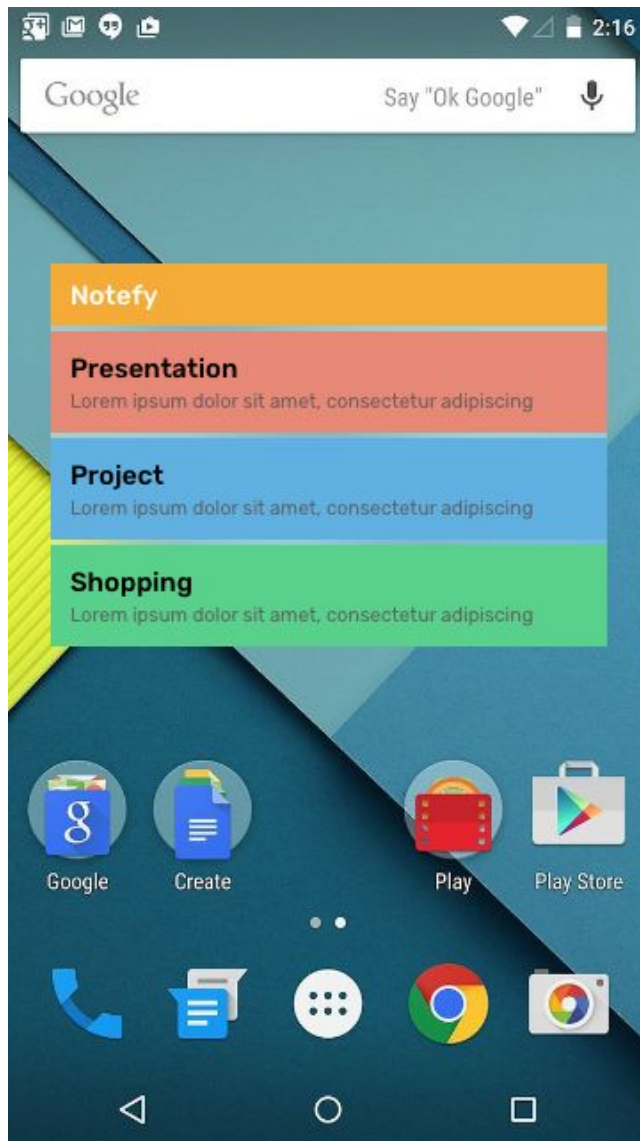
Start writing ...





Screen that allows rich-text editing, tagging, pinning, selecting and icon and setting reminders for the selected note.

Widget



Widget that displays a list of the users notes on the phone's home screen.

Key Considerations

How will your app handle data persistence?

The application will use Firebase as a backend storage.

Describe any edge or corner cases in the UX.

In the case where a user tries to open a note by clicking on a notification, while another notification is being created, the application will ask the user whether he wants to save or discard the note being created before opening the new one.

Data will be persisted through screen orientation within an Android's ViewModel class. In case of process death, data will be persisted in the onSaveInstanceState callback and restored when the application is restarted.

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

Libraries:

- Google Support libraries + new Material components (28.0.0-alpha3)
- Android Lifecycle architecture components (1.0.1)
- Firebase Realtime Database (16.0.1)
- Firebase Auth (16.0.2)
- Firebase Analytics (16.0.1)
- Firebase Messaging (17.0.1)
- RxJava (2.1.16)

Gradle

- Gradle (4.4)
- Android gradle plugin (3.1.3)

Android Studio (3.2 beta)

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

I will be using the Firebase service to authenticate the user, store their data and push notifications the all of their devices.

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

- Configure project and add the required libraries
- Create and integrate the Firebase project
- Setup the required application layers/modules (data, ui, persistence, network...)

Task 2: Implement the MainActivity and setup the basic navigation

- Build MainActivity and the UI
- Setup a navigator and navigation logic for the application

Task 3: Firebase

- Integrate Firebase messaging and handle push notifications
- Authenticate the user using Firebase Authentication
- Handle user access based on session state (with/without session)

Task 4: Add/Edit note

- Create a fragment and UI for adding a note
- Integrate a rich text editor
- Selecting a custom icon
- Add tagging functionality
- Add color selection functionality
- Basic input validation
- Store note

Task 5: Pinning

- Implement pinning note as a persistent notification
- Implement opening the application through a notification intent
- Implement notification actions for un-pinning, editing or deleting a note

Task 6: Notes list

- Create a fragment and UI that will display the complete note list
- Add filtering functionality
- Implement search functionality

Task 7: Handling error states

- Handle no network use-case
- Handle losing session use-case
- Handle failing request use-case

Task 8: Widget

- Implement Widget UI
- Implement broadcast messages that will update the widgets data & state
- Implement Intent handling that will open a note selected from the widget

Task 9: QA

- Implementing automated UI tests
- Manual testing