

Description

Intended User

Features

User Interface Mocks

Tasks screen

Notes screen

Create task screen

Create project screen

Create note screen

Normal category screen

Finished tasks category screen

Task details screen

Note details screen

Project screen

Login screen

Widget

Key Considerations

How will your app handle data persistence?

Describe any edge or 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 or other external services.

Next Steps: Required Tasks

Task 1: [Project Setup](#)

Task 2: [Implement UI for Each Activity and Fragment](#)

Task 3: [Implement Firebase Realtime Database](#)

Task 4: Implement Firebase Auth

Task 5: Create widget that displays tasks for today

Task 6: Implement UI for tablets

Tidy

Description

Tidy allows you to be more productive and organized in your everyday life. Quickly create tasks, plan new projects or just write something down in your notes. Thanks to the cloud sync, you can be productive on all of your android devices.

Intended User

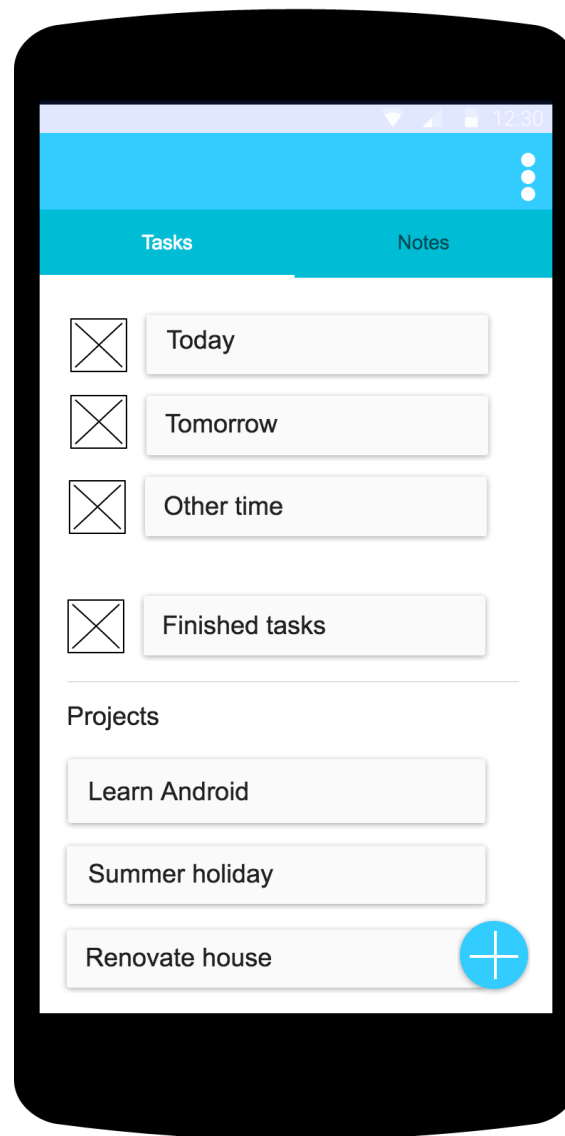
People around age 18-40 who are focused on doing more every day and organizing their life.

Features

- Create tasks
- Add details to tasks
- Group tasks into projects
- Quickly filter your tasks for today or tomorrow
- Look through finished tasks in archive
- Write your thoughts in notes
- Sync your data across multiple android devices
- See your tasks for today on home screen, thanks to the awesome widget

User Interface Mocks

Tasks screen

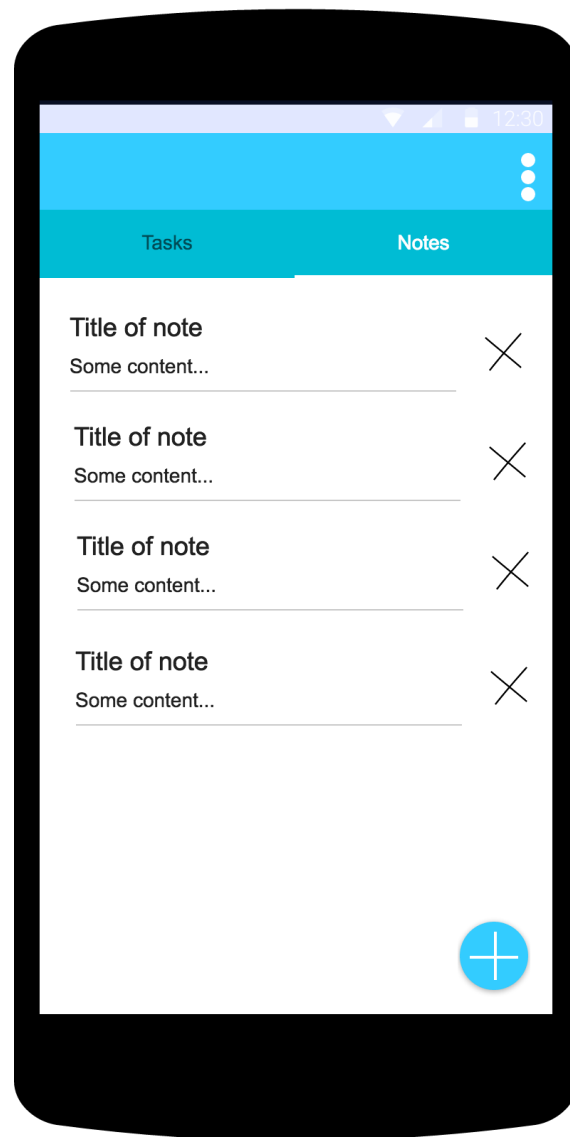


Main screen, displayed after user opens app

- Contains task categories, projects and FAB
- Task categories
 - Today: task due date = current date
 - Tomorrow: task due date = current date + 1 day
 - Other time: task due date = current date + 2 days and more OR no due date
 - Finished tasks: archive for tasks marked as “done”. User can permanently delete tasks that are here, if he wants
 - After clicking on task category, app displays a screen with contents of that category

- Projects
 - Project is a group of tasks which contribute to the same end goal
 - Once user creates a new project, he can add new or existing tasks into it
 - After clicking on project, app displays a screen with contents of that project
- FAB
 - Positioned in the bottom right corner
 - Allows user to create new tasks, projects or notes
- Settings
 - Positioned on the right side of action bar
 - Allows user to manually sync data
 - Allows user to logout

Notes screen



Displayed after user selects Notes tab.

- Contains list of notes and FAB
- Each note has title on top and short part of note content under it
- On the right side of each note is delete icon
- FAB
 - Positioned in the bottom right corner
 - Allows user to create new tasks, projects or notes

Create task screen

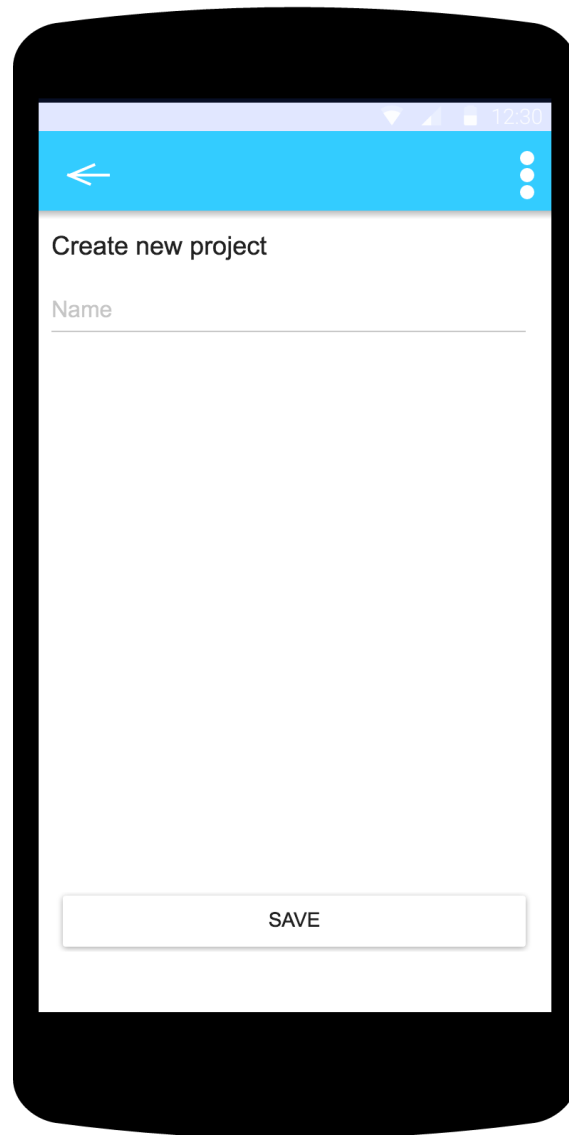
The image shows a mobile application interface for creating a new task. The screen is titled "Create new task" and features a blue header bar with a back arrow on the left and a menu icon on the right. The main content area is white and contains the following elements:

- A text input field labeled "Name".
- A text input field labeled "Description (optional)".
- A section titled "Set due date (optional)" containing a button labeled "DATE".
- A section titled "Add to project (optional)" containing a button labeled "SELECT PROJECT".
- A large "SAVE" button at the bottom of the screen.

Displayed after user selects "Create new task" in FAB.

- Allows user to set details of new task
 - Name (required)
 - Description (optional)
 - Due date (optional)
 - Project (optional)

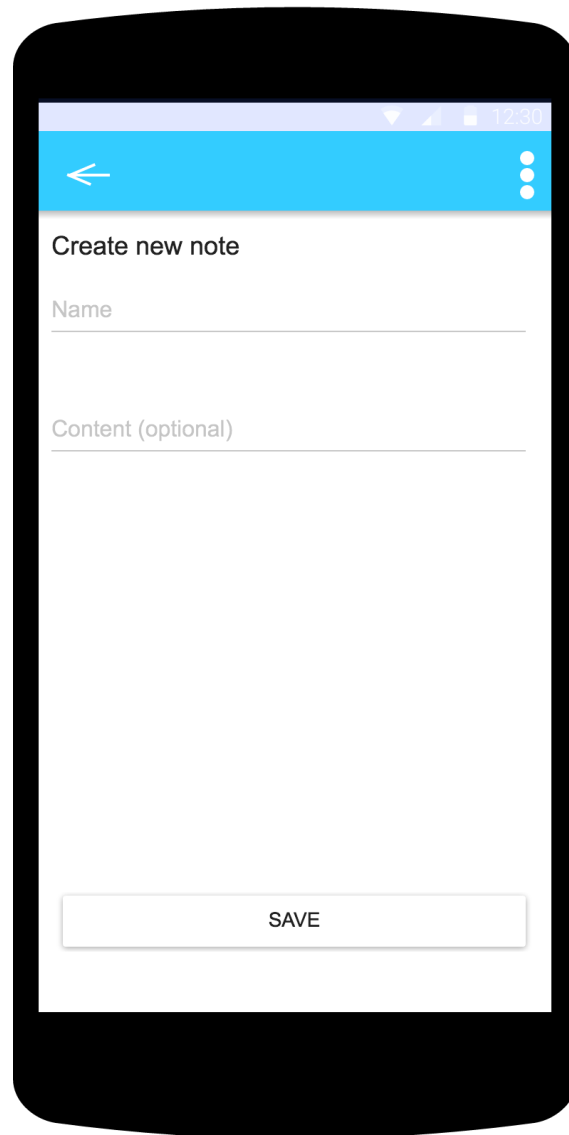
Create project screen



Displayed after user selects “Create new project” in FAB.

- Allows user to set name of new project

Create note screen

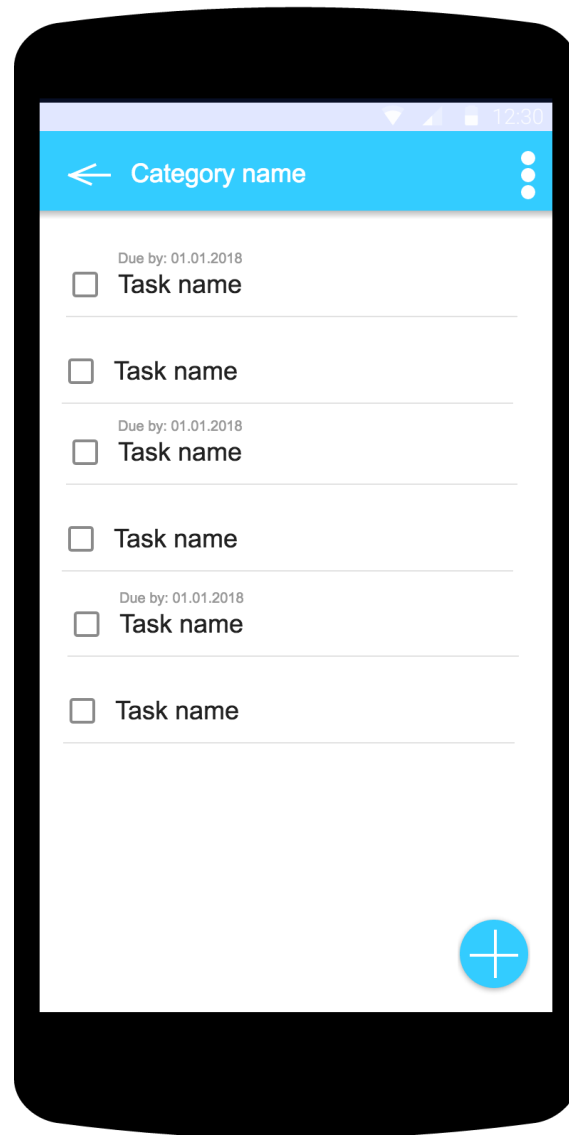


The image shows a mobile application interface for creating a new note. The screen is framed by a black border representing the phone. At the top, there is a light blue header bar with a white back arrow on the left and three white dots on the right. Below the header, the text "Create new note" is displayed. Underneath, there are two input fields: "Name" and "Content (optional)". The "Content (optional)" field is a larger text area. At the bottom of the screen, there is a white button with the text "SAVE".

Displayed after user selects "Create new note" in FAB.

- Allows user to set name and content of new note

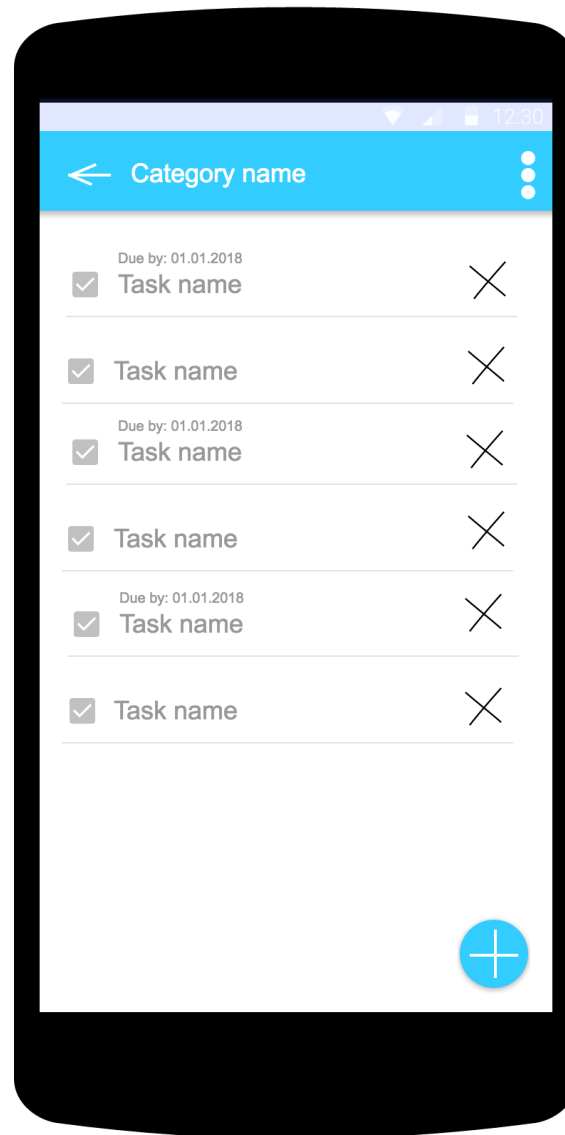
Normal category screen



Displayed after user opens any normal task category – Today, Tomorrow or Other time

- Each task has checkbox and name
- If task has due date, it will be displayed on top
- if user marks task as done, it will be hidden from this list and moved to Finished tasks category
- FAB
 - Positioned in the bottom right corner
 - Allows user to create new tasks, projects or notes

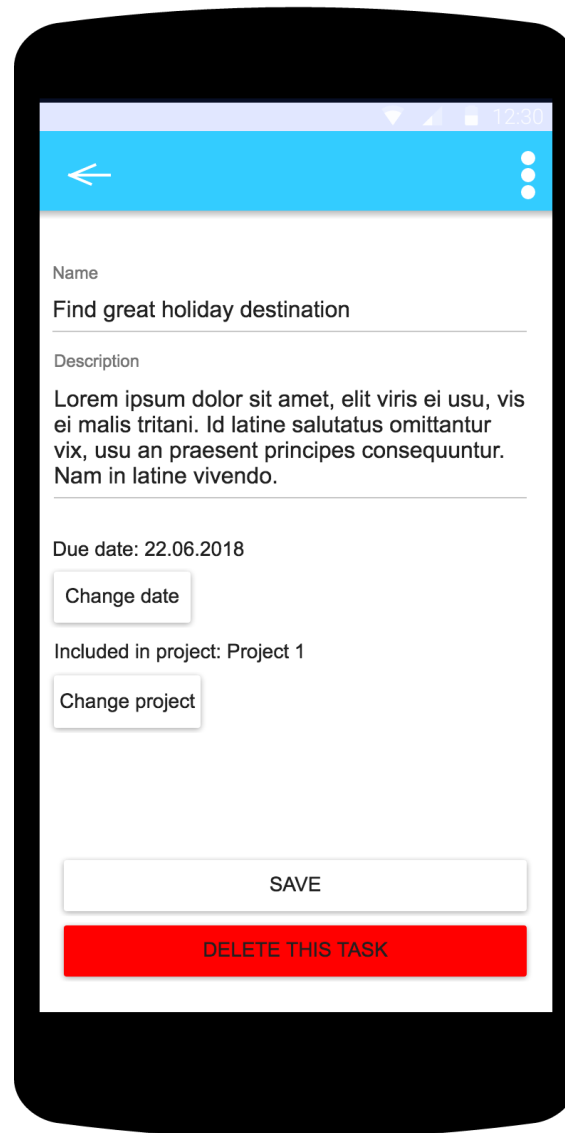
Finished tasks category screen



Displayed after user opens Finished tasks category

- Allows user to see all completed tasks
- By clicking on checkbox, user has option to mark task as not done
- Each task has delete button, which permanently deletes it from DB
- FAB
 - Positioned in the bottom right corner
 - Allows user to create new tasks, projects or notes

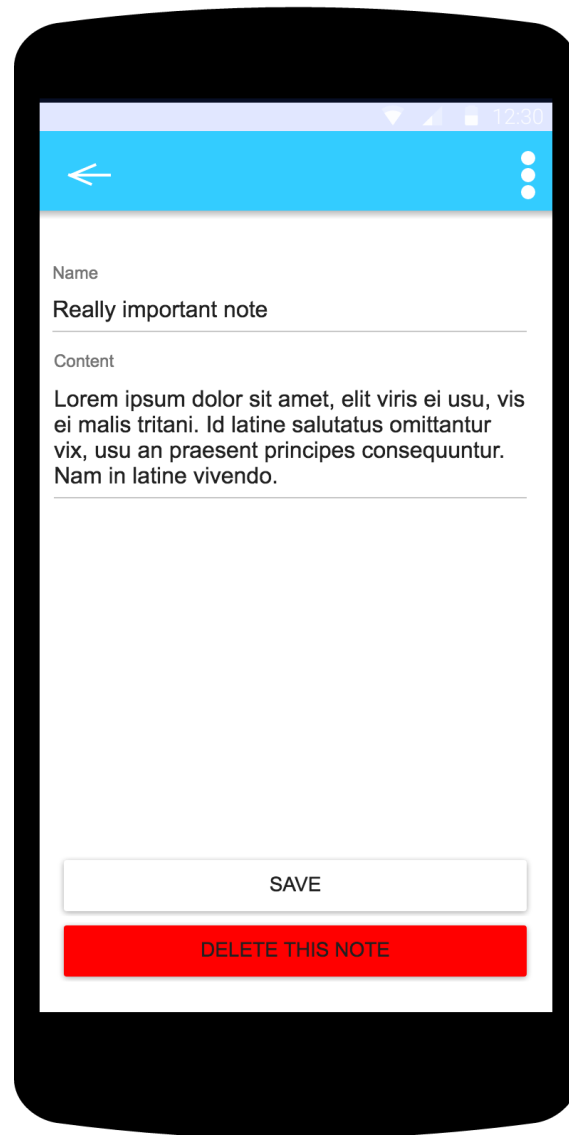
Task details screen



Displayed after user opens any task

- Allows user to edit all details of selected task
 - Name
 - Description
 - Due date
 - Project
- Delete this task button will permanently delete task from DB

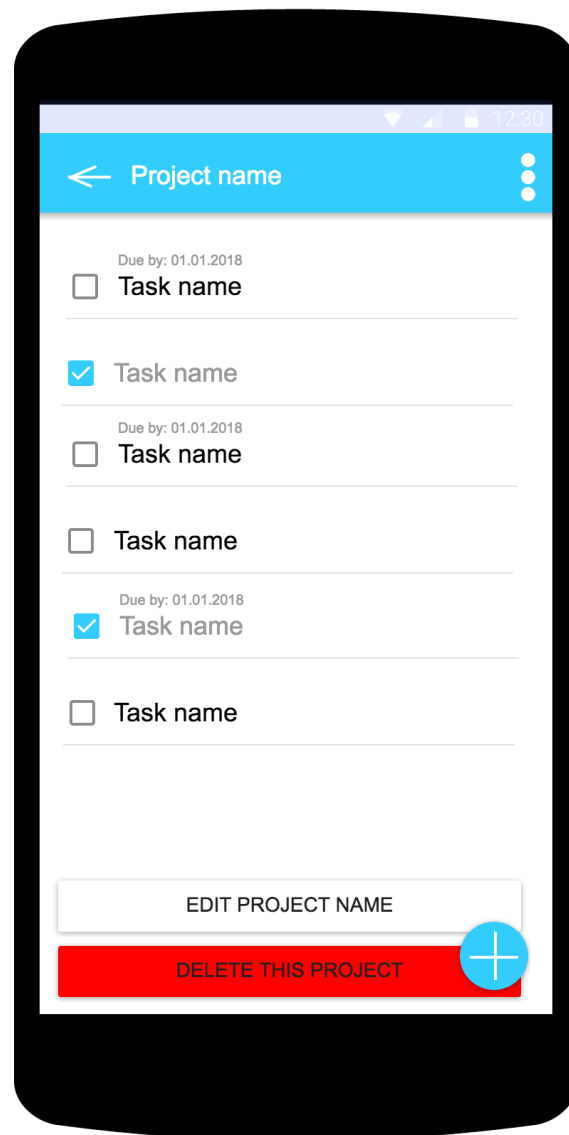
Note details screen



Displayed after user opens any note

- Allows user to edit all details of selected note
 - Name
 - Content
- Delete this note button will permanently delete note from DB

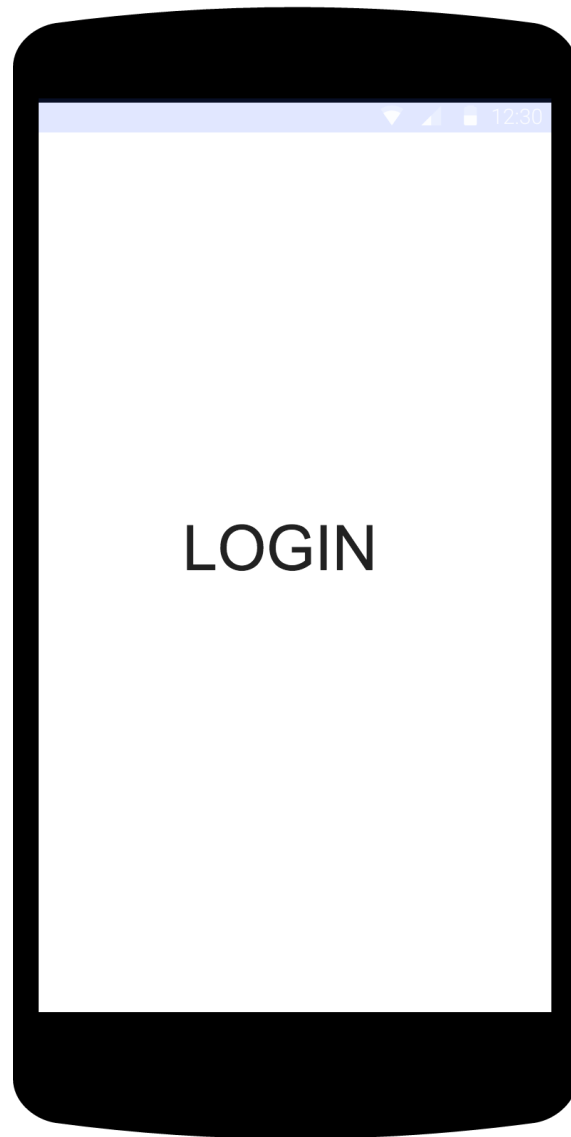
Project screen



Displayed after user opens any project

- Each task has checkbox and name
- If task has due date, it will be displayed on top
- All tasks in selected project are displayed, not just unfinished ones
- Edit project name button allows user to change name of selected project
- Delete this project button will permanently delete project from DB
- FAB
 - Positioned in the bottom right corner
 - Allows user to create new tasks, projects or notes

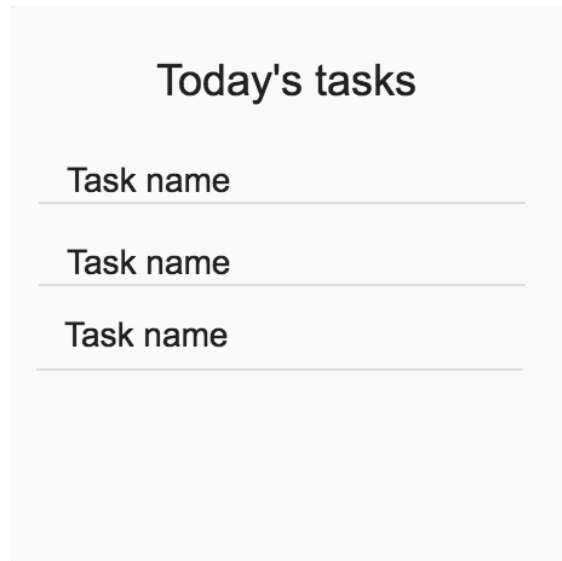
Login screen



Displayed after user open app for the first time or logs out

- Allows user to login with Google Account

Widget



Today's tasks

Task name

Task name

Task name

Widget will allow user to displays tasks for today on his home screen.

Key Considerations

- App is written solely in the Java Programming Language
- App will use IntentService to send/receive data from web on a per request basis (for example after user creates new task, project, deletes something or manually clicks sync button)
- App will keep all strings in strings.xml file and enables RTL layout switching on all layouts
- App will keep all colors in colors.xml file
- App will keep all theme adjustments in styles.xml file
- App will include support for accessibility. That includes content descriptions and navigation using d-pad

Tools and libraries to be used:

Android Studio 3.1.2
Gradle 4.4
Android Support Library 27.1.1
Google Services 4.0.1
Firebase Auth 16.0.1
Firebase Realtime Database 16.0.1
ButterKnife 8.8.1

How will your app handle data persistence?

App will use Firebase Realtime Database

Describe any edge or corner cases in the UX.

If user deletes project that contains tasks, the tasks will stay in the app and won't be deleted. Logic behind this is that some users may simply want to get rid of the project and not lose all tasks associated with it.

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

- ButterKnife for better view binding

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

- Firebase Auth for user authentication
- Firebase Realtime Database for saving data

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries

Task 2: Implement UI for Each Activity and Fragment

- Build UI for main screen fragments – Tasks and Notes
- Build UI for task categories
- Build UI for new task screen
- Build UI for new note screen
- Build UI for new project screen
- Build UI for task details activity
- Build UI for note details activity
- Build UI for project details activity

Task 3: Implement Firebase Realtime Database

- Connect app to Firebase
- Add Firebase Realtime Database to app-level build.gradle
- Configure Firebase Database Rules
- Write to database
- Read from database

Task 4: Implement Firebase Auth

- Set up sign-in through google account
- Customize the sign-in UI

Task 5: Create widget that displays tasks for today

- Build UI for widget activity

Task 6: Implement UI for tablets

- Build tablet UI for main screen fragments – Tasks and Notes
- Build tablet UI for task categories
- Build tablet UI for new task screen
- Build tablet UI for new note screen
- Build tablet UI for new project screen

- Build tablet UI for task details activity
- Build tablet UI for note details activity
- Build tablet UI for project details activity