How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: https://github.com/pthanasack

RPG Notes

Description

RPG Notes is an Android Application that allows you to take notes of your RPG (Role Playing Game) ideas. You can take notes as in any other Notes app, but you can categorize and order them as specific RPG categories; PC (Playable Characters), NPC (Non Playable Characters), Items, Monsters, Weapons, Areas, Events, Settings, Backgrounds.

App is written solely in the Java Programming Language

App utilizes stable release versions of all libraries, Gradle, and Android Studio.

Intended User

This is an app for any RPG Game Master, Game Designer to make notes

Features

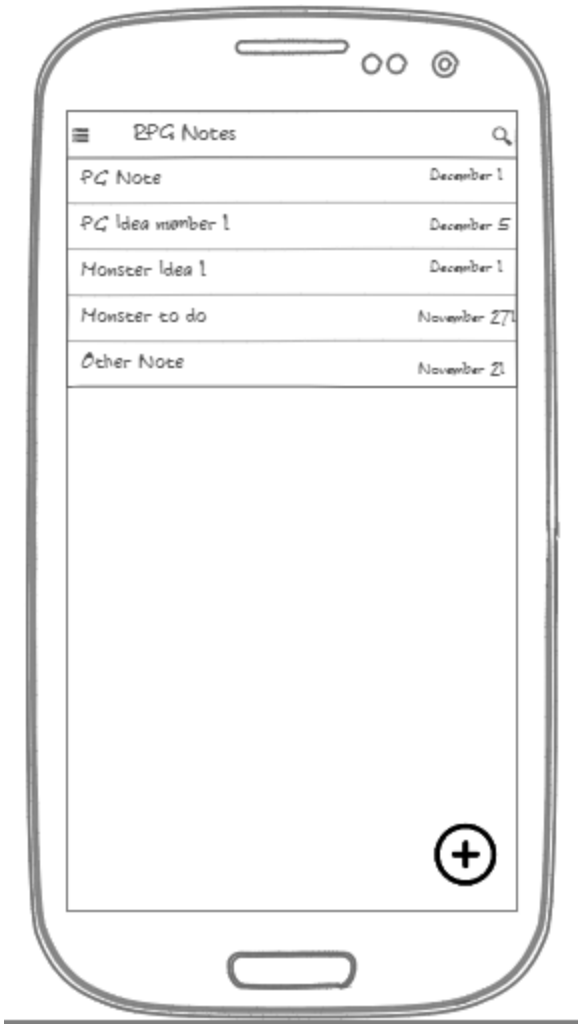
- Save Notes
- Edit Notes
- View Notes
- Delete Notes
- Widget

App includes support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.

App keeps all strings in a `strings.xml` file with attribute translatable where applicable so it can be used for internalization and proper translation

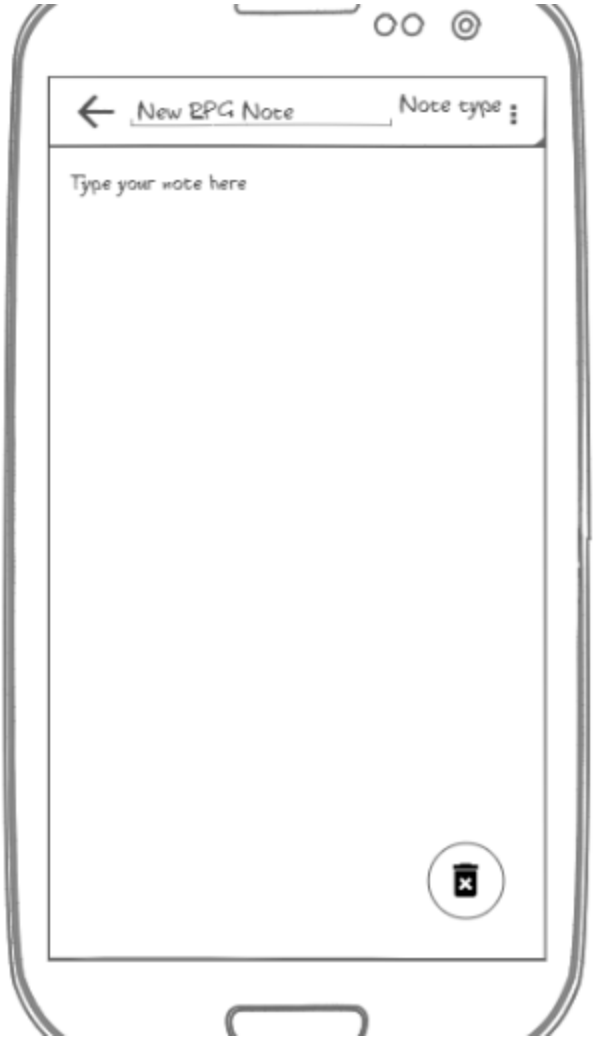
User Interface Mocks

Default Activity Screen (Main Activity)



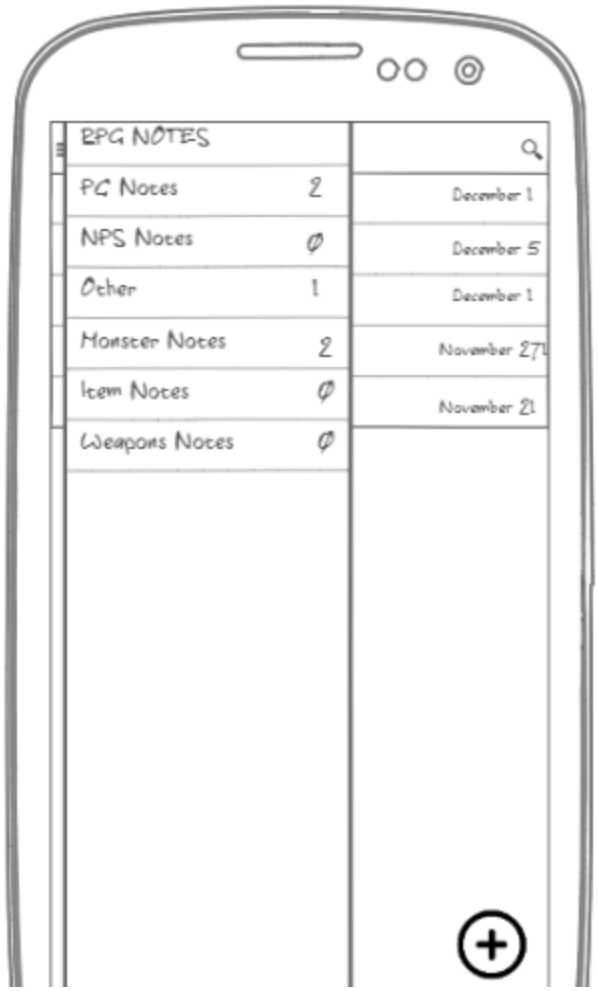
Activity - displayed by default
Lists all Notes sorted by date in a list (only title of the each Note is displayed, with timestamp).
Paging will be used if too many notes to display
FAB to add a new note (see Add Note Screen)
In the Toolbar you can search for a note (see search Screen) and get t o the Drawer Menu
Selecting a note and you will display it in a Detail Activity (see View Note screen)

Add Note Screen



Edit text area where you can type your new Note and change its type with the spinner in the toolbar
Name of the Note is also a Text Field and is editable
Back button automatically saves the note and go back to default activity
FAB will delete the current note and will lead back to Default Activity

Drawer Menu



Drawer Menu displays all Notes categories and number of notes of each type already created
Each category displays the Notes of that category, in same layout as Default Activity

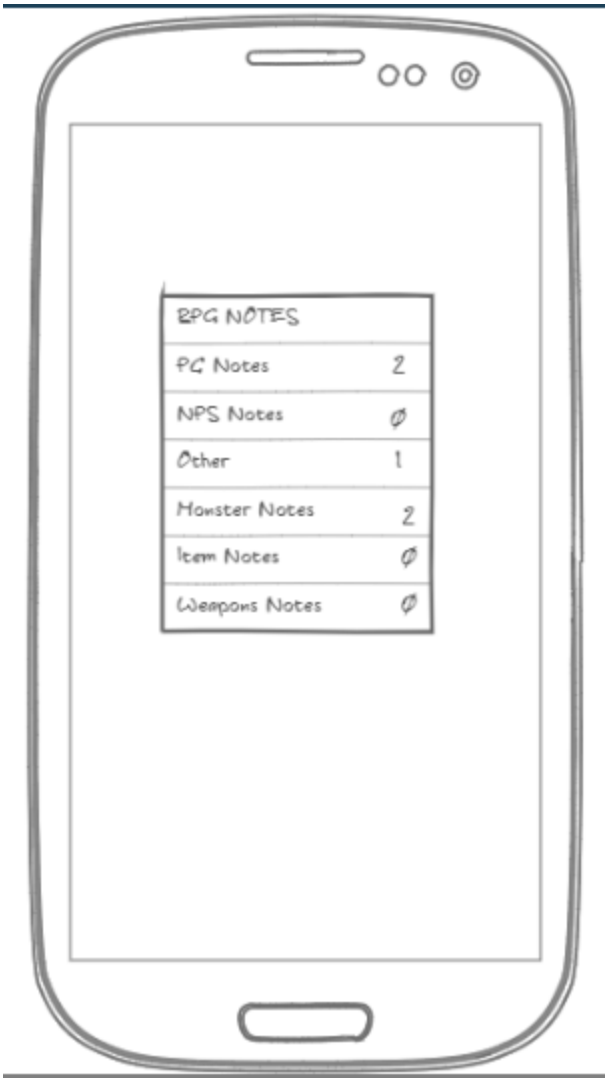
Search Screen

Search Dialog box where you can enter a search word to search in all Notes contents
Search results are displayed in same layout as Default Activity

View Note Screen

Edit text area where you can view your Note and directly edit its content in the text area, and change its type with the spinner in the toolbar
Back button automatically saves the note and go back to default activity
FAB will delete the current note and will lead back to Default Activity

Widget Screen



Widget will provide list of Notes Category and Number Notes as in the Drawer
Click on a category will open the App and display the Notes of that category, in same layout as Default Activity

All screen layout will be the same in landscape mode

Key Considerations

How will your app handle data persistence?

The app will use a Room Database to store all Notes. Notes are saved when exiting the edit note activity.

Describe any edge or corner cases in the UX.

A user types a note that is very long or too long for the app. Maybe limit the maxLength to 65536 character for future export
A user creates too many notes.

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

App will use Font Awesome Android Library to render Items icons in the Drawer menu and widget.

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

App will use Google AdMob to display adds and Google Analytics to provide app use data for Firebase.

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

Set up an Androidx project with drawer.
Add implementation for:
Recyclerview
Appcompat
Android Material
Room
Paging

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for Drawer Menu
- Build UI for Add/Edit a note Activity
- Build UI for Search Activity

Task 3: Database

- Implement the Database:
    - DAO,
    - LiveData Class,
    - Room Database,
    - Repository
    - ViewModel
- Link the Database to the UI Recyclerview (Main Activity)
- Populate the DB to test the Main Activity

Task 4: Drawer Categories

- Build the SQL Requests and Java methods to request the different Notes Categories
- Call those methods from drawer categories and displays their results in Main UI Activity

Task 5: Edit Activity

- Build the SQL Requests and Java methods to request the content of a specific Note
- Call those methods from Main RecyclerView to display the content in Add/Edit Note Activity
- Build the SQL Request and Java method to save a Note
- Call this method when exiting the Edit Activity

Task 6: New Note from fab

- Display An Edit Activity with default values when FAB is clicked

- Call the Save method when exiting the Edit Activity

Task 7: Delete Notes

- Build the SQL Requests and Java methods to delete a specific Note
- Call those methods from FABs in Add/Edit Activity

Task 8: Build Widget

- Create and declare Widget
- Use Pending Intents to open the app
- Use the corresponding SQL Requests and Java methods to request the selected Notes Category, and displays their results in Main UI Activity

---

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**"