***Project Description Document***

1. **Project Statement**

The basic characteristic to develop or design an app is the need of advancement. In this project, we design an application that categorizes tasks and notes while assigning a priority value to both the categories.

**Explanation**

An app that manages a daily to-do list with different categories of activities and notes.   
Create various list categories –

Organized → Add to-do items to each list category →

Have time stamp/deadline for each item added → Set priorities → Add tasks, subtasks, notes for events, projects, vacation planning, Appointments. → Filter categories and notes (Search)

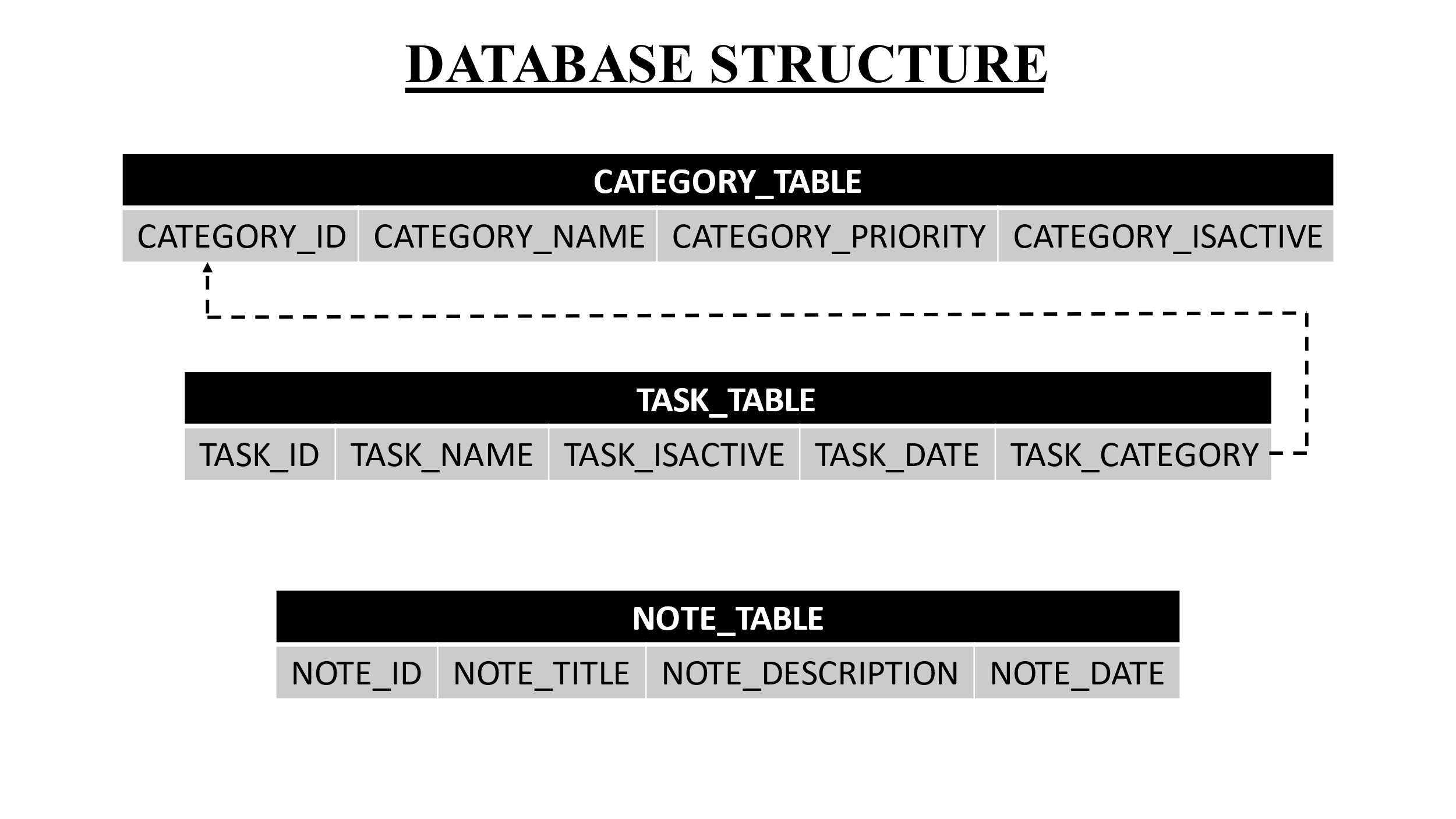
With the advancement of technology and busy life many daily things are forgotten or skipped due to lack of time and improper planning. By designing this application which allows the user to filter all his tasks to-do in a day and on a routine basis, the user can calculate the time and can schedule his appointments and other tasks in an organized and efficient way such that all the tasks can be given proper time which is needed.

* 1. Need for technological advancement
  2. Features that are being offered by the existing applications are faded
  3. Need a broader approach towards the valued application.

This application is targeted to serve a large population as it categorizes every individual function and sets a priority value to each one of them. Users can create, arrange and manage their tasks with respect to the functions we provide in the application.

As the application itself is a dynamic one, we suggest that any component used be a recursive component functioning through bugs and updates. The basic software requirement would be a Smartphone that enables the user interface in a way that enrolls systematic updates whilst keeping the principal code alive. This application requires an android device which is working with an operating system Android 5.0 or above.

1. **Application Design**

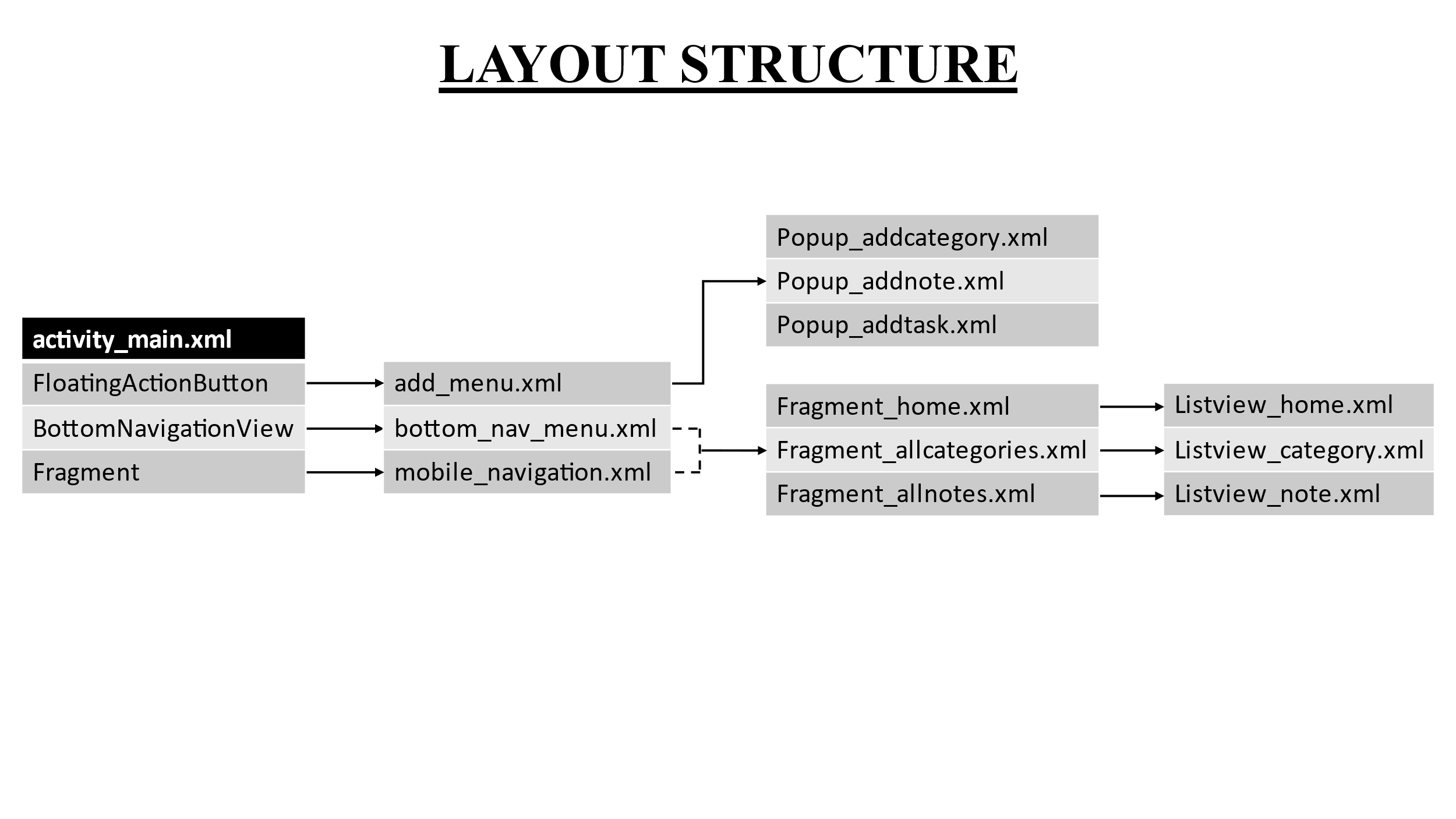


As mentioned in the above diagram, we first create our database using SQLite which we can store, retrieve and update at any given time into our application.

We will create a database within which we create three tables namely, CATEGORY\_TABLE, TASK\_TABLE and NOTE\_TABLE.

The IDs of Category, Note and Task tables are the primary keys that identify every field we assign into the table.

The TASK\_TABLE has a foreign key pointing to the CATEGORY\_ID of the CATEGORY\_TABLE.



After creating our database, we come to our application development, where we create a activity main page, a navigation menu, an add button (+) and three fragments that is categorized to perform all the tasks we can do on this application using navigation from the bottom navigation view.

The add button facilitates creation of Categories, Tasks and Notes by showing a different popup for each option selected on the add\_menu.xml. The popup has an “Add” button that inserts the new records into the database. The popups have multiple layout elements like Spinners that access the database to fetch the existing data from the database. We generate toast messages to verify our operation such as if the categories, tasks and notes are added into our database or not.

The home page displays the tasks categorizing them to “Today”, “Tomorrow” and “Other Tasks” that will display tasks accordingly. There is also a Search Tasks option to search the tasks using the Task Names.

The notes fragment displays the list of notes created sorted by the date and search using the notes title. The notes listview will display notes with the description limited to 3 lines and shows and ellipses at the end of 3 lines.

The Categories fragment displays the list of all Categories sorted by the order of priority and search using the Category name.

1. **Application Implementation and Evaluation**

**MainActivity.java:** This java class handles the bottom navigation and the click of the floating add button to display a popup and insert the records in the database accordingly.

**AllCategoriesFragment.java**: This java class handles the Categories fragment, fetches the category records to display in the listview by querying the CATEGORIES\_TABLE.

**AllNotesFragment.java**: This java class handles the notes fragment, fetches the notes records to display in the listview by querying the NOTES\_TABLE.

**HomeFragment.java**: This java class handles the home fragment, fetches the task records to display in multiple listviews under “Today”, “Tomorrow” and “Other Tasks” sections by querying the TASKS\_TABLE.

**DatabaseStructure.java**: This java class creates objects of Category, Task and Note to be used throughout the application.

**DatabaseHelper.java**: This java class is used to create the database and the tables on the initial installation of the application. All the database methods like insert, update and retrieve are mentioned in this class for better clarity.

**CustomCategoriesAdapter.java**: This class is used to customize the ArrayAdapter for the listview to display the list of categories and override onclick methods on edit icon click.

**CustomNotesAdapter.java:** This class is used to customize the ArrayAdapter for the listview to display the list of Notes and override onclick methods.

**CustomHomeAdapter.java:** This class is used to customize the ArrayAdapter for the listview to display the list of Tasks and override onclick methods for the checkbox and update the database accordingly.

1. **References**If you have important references for this document or for your project, list them here.

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.geeksforgeeks.com](http://www.geeksforgeeks.com)

[www.w3schools.com](http://www.w3schools.com)

**Experiences and Thoughts**There are many functionalities that can be implemented to make the app more user friendly. Some of these functions would be nice to have in this app are:

* Enhanced User Interface
* Add additional database functionalities to edit and delete tasks, categories and notes.
* Send Reminders by integrating with clock application.

1. **OUTPUT SCREENSHOTS:**

Graphical user interface

Description automatically generated

A picture containing chart

Description automatically generated

A picture containing chart

Description automatically generated

Graphical user interface, application

Description automatically generated

A picture containing graphical user interface

Description automatically generated Graphical user interface, application

Description automatically generated A picture containing chart

Description automatically generated Graphical user interface, application

Description automatically generated

A picture containing text

Description automatically generatedv Chart

Description automatically generated with medium confidence Chart

Description automatically generated with medium confidence A picture containing text

Description automatically generated