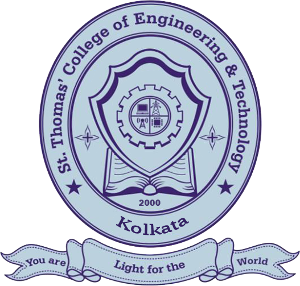
**St. Thomas’ College Of Engineering & Technology**



PROJECT ON

TO-DO APP WITH MULTIPLE USER SUPPORT

BY

GROUP-7

GROUP MEMBERS

Sujan Mitra-24

Debojit Mondal-25

Poulami Mukherjee-26

Kaustuv Nandy-27

# About The Project

To-do list apps are some of the oldest available apps on both [iOS](https://www.digitaltrends.com/mobile/best-iphone-apps/) and [Android](https://www.digitaltrends.com/mobile/best-android-apps/), that keeps a track of the tasks to be done by the user. An attempt has been made to create a To-Do Android Application with multiple user support.

This To-Do app is as simple as to-do list apps come. It’s a gorgeously minimal and well-designed app that does exactly what it’s supposed to and no more. It can be used to create tasks, make a description for the tasks, under different user name .These appear in a recycler view, and each subtask can be marked complete when the time is right. Setting a due date for the main task is also included as on of the features . Each task sits under a list, and there’s no limit to how many lists that can be created. It can have a shopping list, a to-do list, and more. In exchange for its simplicity, there is a lose some of the more in-depth tagging and organizational features that may be found in other apps.

The user needs to first login to add tasks and getting started is as simple as typing your first task in and hitting save . Setting up new tasks is just as easy as typing the task and hitting “save task”. You can also set a deadline for the work to be completed. Completing tasks involves ticking off the box to the side of the task, and there’s a certain satisfaction to be had just ticking off tasks.

The App can be used to organize the schedule of the user or can simply act a notepad for the user to keep a note of the things to be done and also the stipulated time for the particular task. Inspite of allowing multiple user support, the app doesn’t compromise the privacy of each user. The users can access his/her own to-do list after logging into the app using the correct credentials. To set the credentials for each user , the app first allows the user to “sign up” through a username and a password for each new username.

# Technologies involved

The application has been developed for Android Operating System, which is basically a [mobile operating system](https://en.wikipedia.org/wiki/Mobile_operating_system) developed by [Google](https://en.wikipedia.org/wiki/Google). It is based on a modified version of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and other [open source](https://en.wikipedia.org/wiki/Open-source_software) software, and is designed primarily for [touchscreen](https://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smartphones](https://en.wikipedia.org/wiki/Smartphone) and [tablets](https://en.wikipedia.org/wiki/Tablet_computer). In addition, Google has developed [Android TV](https://en.wikipedia.org/wiki/Android_TV) for televisions, [Android Auto](https://en.wikipedia.org/wiki/Android_Auto) for cars, and [Wear OS](https://en.wikipedia.org/wiki/Wear_OS) for wrist watches, each with a specialized user interface. Variants of Android are also used on [game consoles](https://en.wikipedia.org/wiki/Video_game_console), [digital cameras](https://en.wikipedia.org/wiki/Digital_camera), [PCs](https://en.wikipedia.org/wiki/Personal_computer) and other electronics.

Android's default user interface is mainly based on [direct manipulation](https://en.wikipedia.org/wiki/Direct_manipulation_interface), using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, along with a [virtual keyboard](https://en.wikipedia.org/wiki/Virtual_keyboard).

Android devices boot to the homescreen, the primary navigation and information "hub" on Android devices, analogous to the [desktop](https://en.wikipedia.org/wiki/Desktop_metaphor) found on personal computers. Android homescreens are typically made up of app icons and [widgets](https://en.wikipedia.org/wiki/Software_widget); app icons launch the associated app, whereas widgets display live, auto-updating content, such as a [weather forecast](https://en.wikipedia.org/wiki/Weather_forecast), the user's email inbox, or a [news ticker](https://en.wikipedia.org/wiki/News_ticker) directly on the homescreen.

This application has been designed using Android Studio which is the official [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Google](https://en.wikipedia.org/wiki/Google)'s [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [operating system](https://en.wikipedia.org/wiki/Operating_system), built on [JetBrains](https://en.wikipedia.org/wiki/JetBrains)' [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA) software and designed specifically for [Android development](https://en.wikipedia.org/wiki/Android_software_development). It is available for download on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) based operating systems. It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (ADT) as the primary IDE for native Android application development. Java is programming language used for the Android App development using Android Studio.

The databases used in the application are designed using Realm which is an [open source](https://en.wikipedia.org/wiki/Open_source) [object database](https://en.wikipedia.org/wiki/Object_database) [management system](https://en.wikipedia.org/wiki/Database_management_system), initially for mobile (Android/iOS), also available for platforms such as [Xamarin](https://en.wikipedia.org/wiki/Xamarin) or React Native, and others, including desktop applications (Windows), and is licensed under the [Apache License](https://en.wikipedia.org/wiki/Apache_License).

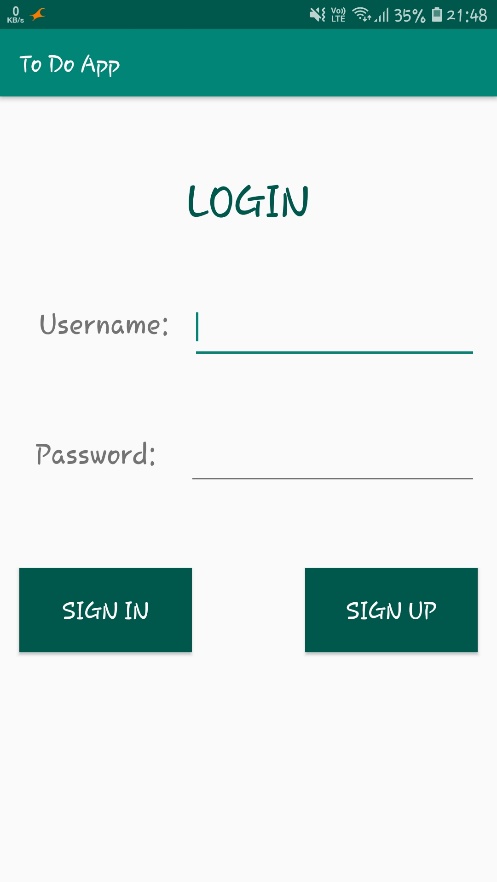
In a Realm file classes are used for schema definition and relationship between objects are allowed via ‘links’. The query results returned by Realm are thread-local views to the current "database version" , and these views "automatically update" when a transaction is committed from any thread, as long as Realm is able to update its instance version .In case of an update, Realm calls change listeners that are added to its query results.

# Methodology Followed

Design

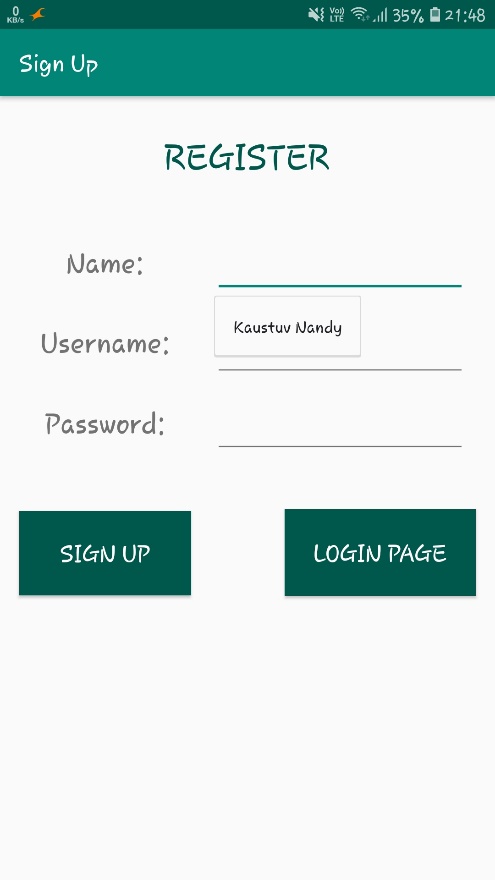
1. Login Page

The app opens up to the login page. A new user can click on the sign up button which directs him/her to the register page. A previously registered user can input the correct credentials to sing into the application. The password and username are fetched from the ‘User’ database to check if the credentials entered are valid and existing. In case of an invalid log in , a toast is added to show the error condition. If either of the fields are left empty then again a toast indicates the erroneous situation.



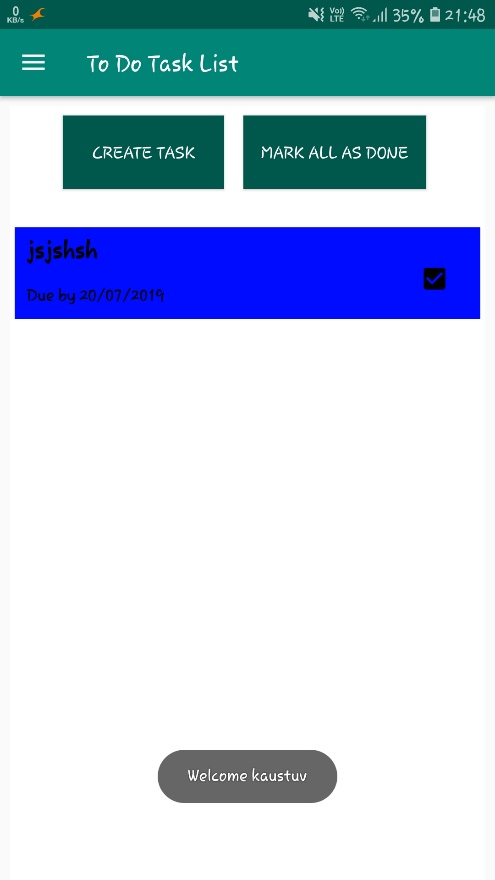
1. Sign up Page

A new user is directed to this page from the login page and asked to fill in the necessary details like the Name, Username and Password. If any of the fields is left empty a toast indicates the absence of the same and also if a username has already been used before, that is, an entry into the User database for that username already exits then again a toast shows this as an error. The sign up page redirects to the login page after successful sign up.



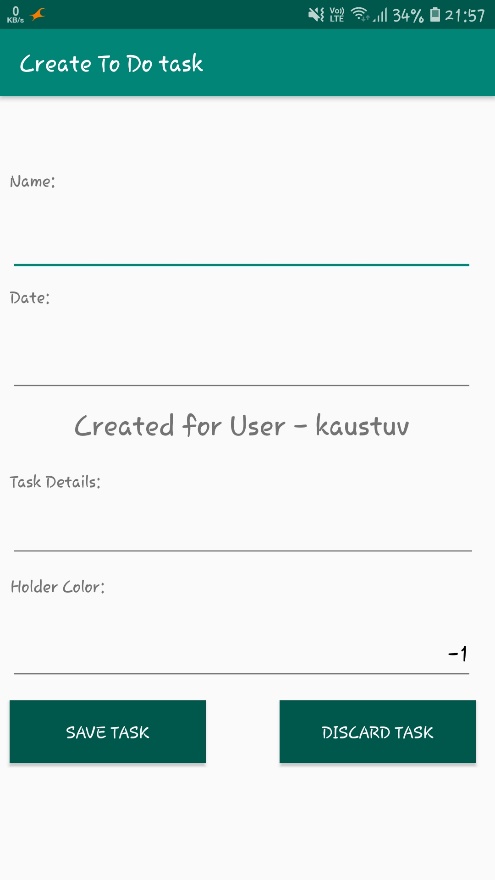
1. To-Do task list

Logging in or signing into the app leads to the to-do task list for the logged in user. Clicking the create task button will take the user to the next page to create tasks under the same username. The ‘mark all as done’ button will add the check-task field of every task as true.The refresh button will take the user to the show the tasks under one user.



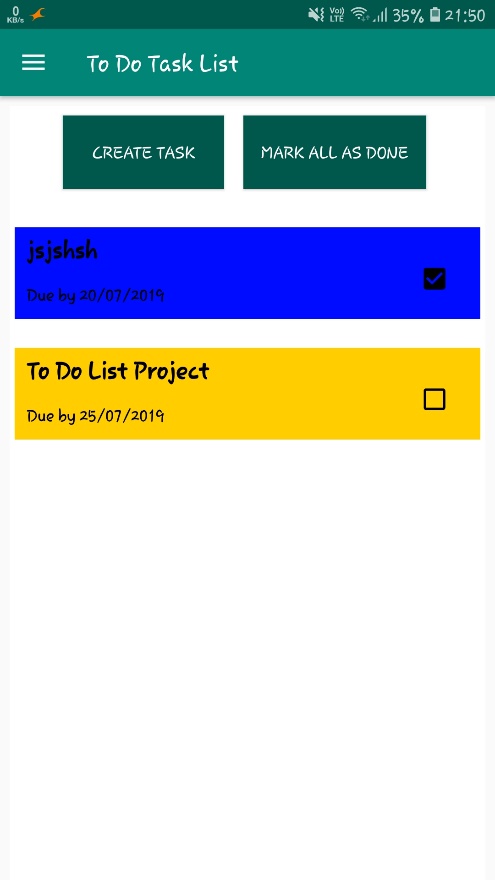
1. Create To-Do task

Clicking on the create task button in the to do list will bring the user to this page where the user may add the task name, date, task details and also the holder colour of the view where the task will be shown. Clicking the ‘Save Task’ button will add the task to the ‘Task’ database. If the user doesn’t want to add the task to the database he/she needs to click on the discard task button that will kill the intent and take the user back to the To-Do list page



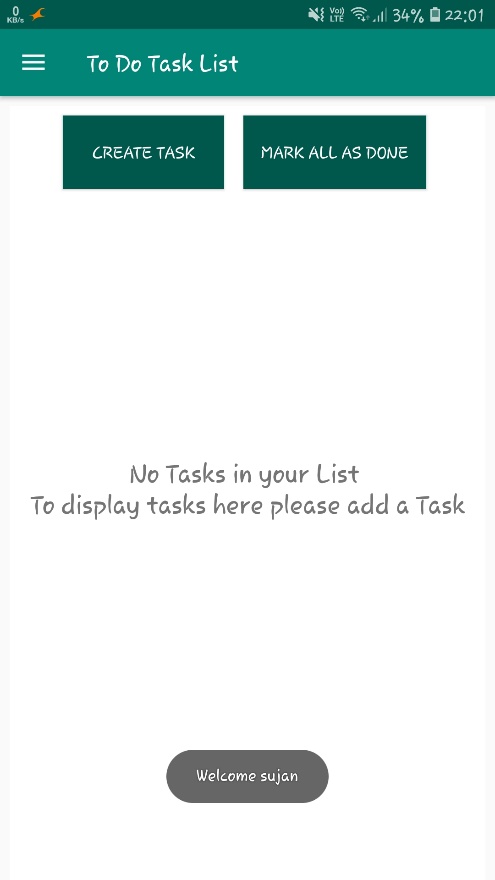
1. Recycler view

The recycler view will contain all the tasks under a particular user in different holder colours as mentioned in the create task page.



1. No Task Fragment

If there is no task under a particular user then refreshing will lead to this page



Database Schemas

1. User Database

This database stores the Name, Username and Password of each user where the username is the Primary Key

|  |  |  |
| --- | --- | --- |
| **Name** | **Username** | **Password** |
| Poulami Mukherjee | Pom | 12345 |
| Kaustuv Nandy | Nandy | 54321 |
| Sujan Mitra | Sujan | 67890 |

1. Task Database

This database stores the username, task name, task details, date and a system generated id acts as the Primary Key.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Username** | **Task\_name** | **Task\_details** | **Date** | **Check\_task** |
| 1233 | Pom | Android Project | To-Do App | 24/07/19 | False |
| 1423 | Nandy | Android Project | To-Do App | 24/07/19 | False |