**St. Thomas’ College of Engineering and Technology**

**Industrial Training report on**

**“PROJECT ON ANDROID- TODO APPLICATION”**

**Department of Information Technology**

**SEMESTER – 7TH YEAR- 4TH**

**Group No. – 06**

**Group Member Roll No.**

**DITIPRIYA MAJUMDER 21**

**SAKSHI MANJRI 22**

**ARGHYADEEP MANNA 23**

**AASFI MANWAR 24**

**INDEX**

* **About Project**
* **Technologies Involved**
* **Methodology**
* **System Design**
* **Database Schema**
* **Screenshots of Complete Application**
* **Future Scope**
* **Conclusion**

**About the Project :**

In this project , we have created an android based Todo application with Realm Database being its backend to store Task list or multiple users which they need to achieve before their due date. A ToDo Task app is where we can save schedules of some tasks to get reminded on the scheduled time.

User has to register for the app and once they logged in , each user can only see their own task and the rest of the tasks are hidden from their view.Tasks can be checked to denote that it has been completed and will thus be removed from the task list from the next login session.

**TECHNOLOGIES INVOLVED:**

* Android Sdk(Android 9 API 29)
* Java Programming language
* XML
* Realm Database

**METHODOLOGY**

As a user navigates through, out of, and back to your app, the activity instances in your app transition through different states in their lifecycle. The activity class provides a number of call backs that allow the activity to know that a state has changed: that the system is creating, stopping, or resuming an activity, or destroying the process in which the activity resides.

Within the lifecycle callback methods, you can declare how your activity behaves when the user leaves and re-enters the activity. For example, if you're building a streaming video player, you might pause the video and terminate the network connection when the user switches to another app. When the user returns, you can reconnect to the network and allow the user to resume the video from the same spot. In other words, each callback allows you to perform specific work that's appropriate to a given change of state. Doing the right work at the right time and handling transitions properly make your app more robust and performant. For example, good implementation of the lifecycle callbacks can help ensure that your app avoids:

* Crashing if the user receives a phone call or switches to another app while using your app.
* Consuming valuable system resources when the user is not actively using it.

**SYSTEM DESIGN:**

Our To-Do Application design is following a sequence of steps through moving between different activities. It starts with the main activity where users are provided with login and sign up options .New users have to moveto the sign up activity where they are enter details like the user name,an email-Id ,a phone number and a password.These information are stored in the user database and simultaneously a system generated user-Id is provided to the userfor login purpose.Pre-existing users can however move to the login page and enter their username and password which is being searched and matched for in the database and if authentication details are matched user is logged in or appropriate message is shown for failure.

User enters the dashboard which will be containing the list of tasks pending ranging from zero to as many as desired.The tasks are displayed as a recyclerview list.We have a navigation menu attached to the sidebar connecting to edit profile and logout options.User profile information is sent from the login page to the dashboard.Edit Profile will help tomodify users data and logout exits the user from the app.We can move to the create task activity through the click of a button.

Task details like name of the task,duedate,task details and a holder color for the recyclerview item holder can be chosen by the userand is thus saved as Task information on clicking save button and we move back to the dashboard viewing the newly created task being added to the list.We can also discard the task midway of filling the details and it will take us back to the dashborad.

Each task item also has a clickable imagebutton to record which tasks has been completed and they will be removed fromthe list on beginning our next login session.we can also mark all existing tasks as checked with clicking one button.Pressing back from dashboardwill also log us out from the system.

**DATABASE SCHEMA:**

A single Realm database was sufficient for our application and it consisted of two Tables as follows:

1. User :

attritube datatype

userid long(primary key)

username string(primary key)

email string

phone string

password string

2.Task

attribute datatype

id long(primary key)

taskname string

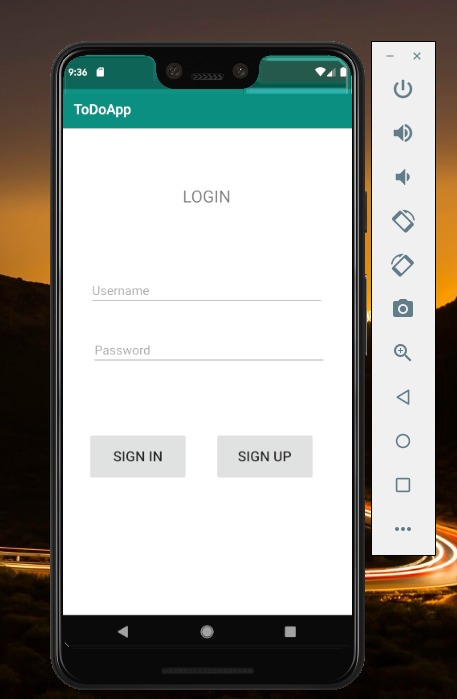
taskDetails string

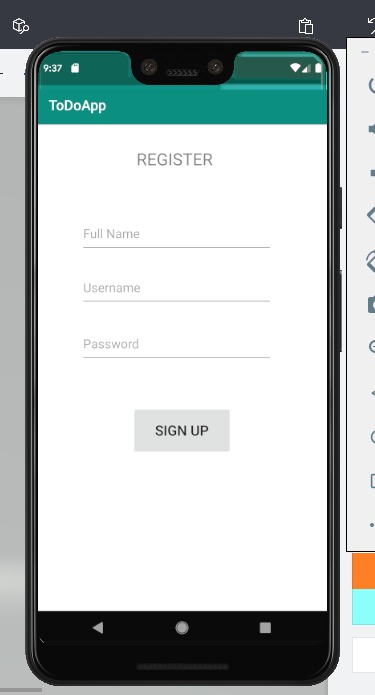
username string

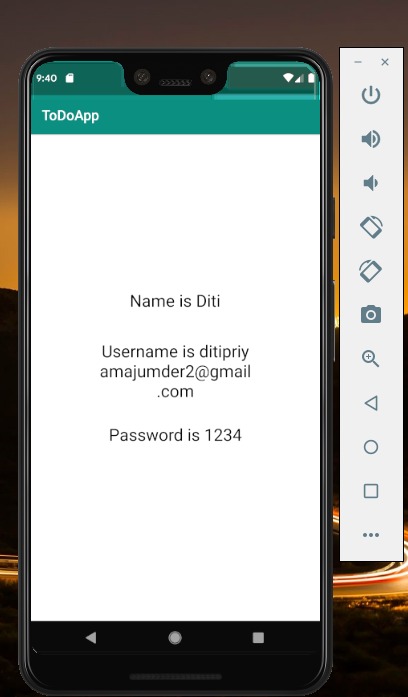
Date Date

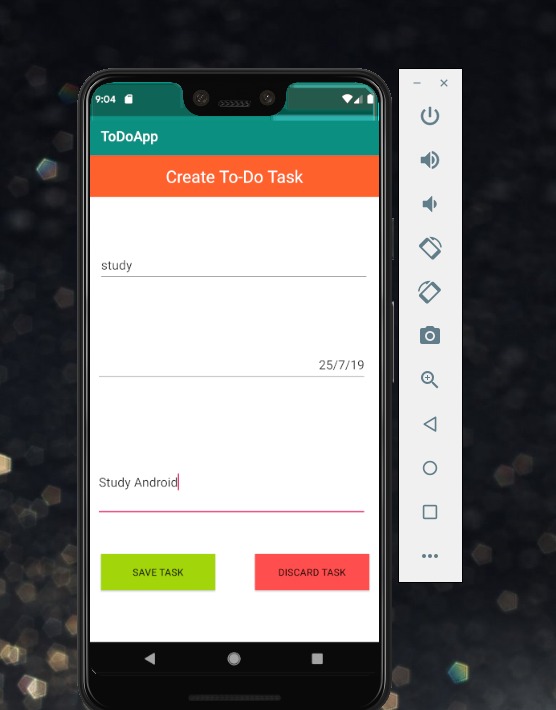
checktask boolean

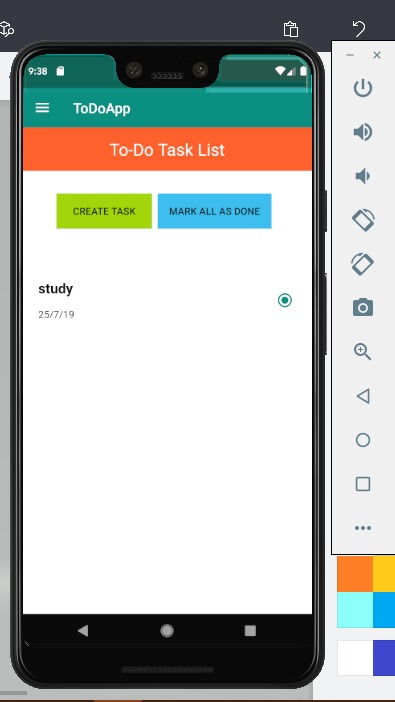
**SCREENSHOTS OF APP**

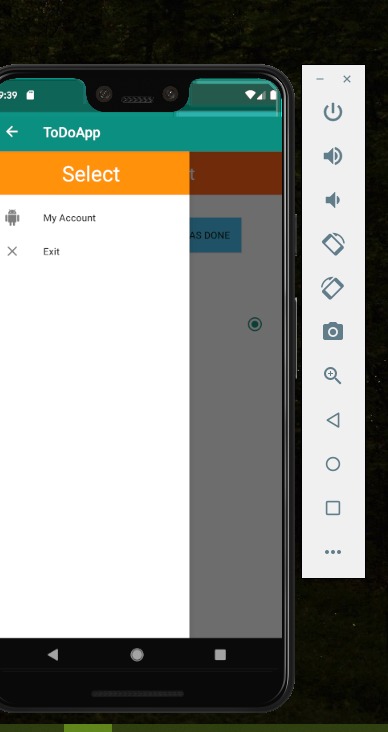
****

****

****

****

****

****

**FUTURE SCOPE**

TODO application is an elaborate task management app, n number of Task can allows user to easily jot down our to-dos and manage our tasks efficiently. As future projects, one can imagine many fields, not only directly concerning the TODO application. Of course, some minor adjustments have to be done to this app, if it should be ready to be deployed via the Android market. The UI could be polished a bit to match the design of modern mobile applications. Additionally some system parts can be improved, mainly in the area of routing performance, to guarantee a smother and faster experience for the end user.

Nevertheless one of the big strengths of this project are definitely the amount of easily, reusable Java packages. As the whole code is open sourced, all parts can be used to build other routing related applications. But also integrate with calendar data, build an improved task management application or port the Dayplan optimizing functionality to another routing engine like Google Maps for example.

Nowadays Android has become very popular which is an open source mainly designed by Google for Smartphone and tables. Android is designed in such a way that allows the developer and device manufacturers companies to alter the software design according to their needs. Todo app is a type of application which has the business opportunities so, experience in this type of field can help to survive today’s market. Need of android developer is huge and still increasing. So the future scope is quite good.

**CONCLUSION**

Several goals were accomplished while working on this teamproject. First of all an application was created which supports a user’s day planning with the following implemented functionalities:

* Generate a dayplan, which is heavily optimized on completing as many tasks as possible.
* Simple, yet powerful task management system
* Continous checking of the day plan’s consistency and compliance.
* The task management functionality of this app is also strongly encapsulated and can therefore be reused easily in any Java project. The storage mechanism of converting tasks into events and storing them at a specific date can also be adapted easily to an Realm database or any other preferred storage solution.