Final Report

Joshua Paterson (n10193197)

# Abstract

The following report details the design and development of an application designed to assist with the Signal Regiment a unit of the Australian army work project organisation. Currently the project is incomplete and will need to be continued on by another however substantial progress has been made. The application is two applications a server and a mobile app with a simple server- client architecture written in the programming language Java. The project has two major objectives being the applications capability and security. The project in its current state has all security measures in place and the general structure and some basic functionality has been implemented however this is far from complete.

Table of Contents

[Abstract 1](#_Toc87112893)

[Introduction 3](#_Toc87112894)

[Problem statement 3](#_Toc87112895)

[Client Needs 3](#_Toc87112896)

[High level objectives 3](#_Toc87112897)

[Low Levels objectives 4](#_Toc87112898)

[Project Constraints 5](#_Toc87112899)

[Literature Review 5](#_Toc87112900)

[Methodology 6](#_Toc87112901)

[Overarching approach 6](#_Toc87112902)

[Data flow description 7](#_Toc87112903)

[Overview 7](#_Toc87112904)

[Server 7](#_Toc87112905)

[Client apps 9](#_Toc87112906)

[Software tools used 10](#_Toc87112907)

[Programming language selection 10](#_Toc87112908)

[Development Environment 10](#_Toc87112909)

[Testing 10](#_Toc87112910)

[Building 11](#_Toc87112911)

[Results and Discussion 11](#_Toc87112912)

[Conclusion 12](#_Toc87112913)

[Future works 12](#_Toc87112914)

[Timeline 13](#_Toc87112915)

[Socket connections. 13](#_Toc87112916)

[Build process setup 13](#_Toc87112917)

[Testing methodology setup 13](#_Toc87112918)

[Encryption and key exchange 13](#_Toc87112919)

[Multi-threaded server 13](#_Toc87112920)

[Login functionality 13](#_Toc87112921)

[User registration 14](#_Toc87112922)

[SHA encryption 14](#_Toc87112923)

[Home UI skeleton 14](#_Toc87112924)

[Database design 14](#_Toc87112925)

[Create project page. 14](#_Toc87112926)

[Project Selection page. 15](#_Toc87112927)

[Project Visualisation 15](#_Toc87112928)

[References 20](#_Toc87112929)

[Appendices 21](#_Toc87112930)

[Source code 21](#_Toc87112931)

[Server 21](#_Toc87112932)

[Client app 38](#_Toc87112933)

[Creation scripts 126](#_Toc87112934)

# Introduction

## Problem statement

The projects finished capability as described by the client (the Signal Regiment a unit of the Australian army) is an application accessible on PED’s (personal electronic device) designed for co-ordinating/collaboration of teams, tasks and projects that is approved for security classification of up to official. Currently teams use outlook calendars, excel spreadsheets vera teams or a “defence diary” on the DRN (Defence restricted network) to organise their projects, the issue with the current methods are information is split between many different applications or by individuals which can cause team managers to lose track of ongoing work due to difficulty finding where information is stored. Documenting task completion can be difficult for personnel as DRN access is not always easy on a job and their competed task are document long after it has been done (end of the day for example). This can cause needless delays or even outright forgetting to document task completion. With the current system used for organising projects there are many points where confusion can be caused, and information can go missing or undocumented.

## Client Needs

The intent behind this project seems to be to create one easy to access application for both team managers and there personal to track day to day projects in real time as well as standardising a communication method for projects while keeping a detailed record of both communication and task progress (likely to aid with accountability and mistake finding). This will likely be most accessible as a downloadable application on an individual’s phone or app.

The main problem with current options in the market seem to be the security risk associated with using applications where direct control of information, servers and source code are not controlled by defence Australia. These risks include uncertainty of encryption levels of information being send over networks and server hosting location (application database access not controlled by defence). These seem to be the main reasons why a dedicated app made and controlled by defence is needed. Even through not directly stated being able to time maintenance and server down time to prevent needless confusion for personal is also a reason on why defence would prefer to use their own personal servers. Also, from my own personal experience with using the DRN (current solution used DRN access) I have found that having to access program through the DRN entail going through a virtual machine which causes all application usage to come with lag or a delay causing user experience to be slow when performing any actions. To have a highly responsive product for end users the application will need to be separate from the DRN to prevent the issues associated with virtual machines.

## High level objectives

The project can be defined by 2 high level objectives the product or application must be designed to meet. These being official+ level encryption and application capability.

* official+ level security (HL-1)

The application will need to be cleared by defence to have access to information at official at minimum or higher securities levels. This will mean data send over any network will need to be encrypted, users will need to be authenticated and data stored will also need to be encrypted. Any other possible security vulnerability will also need to be addressed, eliminated, or minimised to a reasonable degree. These will include but not limited to cross website scripting, sql injection, PED’s loss (device password storage and loss of access), key loggers, remote access to device and shoulder surfing.

* Application Capability (HL-2)

The application must be capable of organising, sorting, and tracking on going work projects from wherever a user is.

## Low Levels objectives

The high-level objectives can be converted to the following low-level objectives. Note that all algorithms where specifically chosen as they were the recommended algorithm by the Australian Signals Directorate as well as could be used to implement the key exchange method detailed in the literature review.

* RSA encryption (HL-1)

RSA encryption is an asymmetric encryption method approved for information with the classification TOP SECRET. The RSA will need a 3072-bit key to be approved for this level of classification. RSA encryption stated will need to be used to exchange AES keys. (Australian Signals Directorate, 2021)

* AES encryption (HL-1)

AES will be used to send information across a network at minimum AES-256 will need to be used for information of up to TOP SECRET (Australian Signals Directorate, 2021). AES will be used for general information exchange as RSA will likely be far too slow to both encrypt and decrypt data causing end users to have an unresponsive user experience (Priyadarshini Patil, 2015).

* Password hashing (HL-1)

To prevent storing passwords in plain text in the database hashing will need to be used. SHA-2-384 will be used and is clear of TOP SECRET classified information using a salt with the SHA encryption is also necessary. (Australian Signals Directorate, 2021)

* Ability to create team/project and create/track, update tasks (HL-2)

The application will need to capable of assisting team leaders and personal in managing tasks. Therefore, the application will need features and functionality to create team/project and create/track, update tasks.

* Ability to communicate to team or individuals (HL-2)

Project management no matter the job will require a way for individual assigned to teams to communicate in a quick and easy manner. This will entail the ability to send messages to groups and individuals in teams as well as push notifications to alert users of messages.

* Easy to understand UI design (HL-2)

Projects can become quickly overwhelming with the sheer number of tasks required to be done. The user interface for clients will need to be easy to navigate to important information as well as ways to view only important information to prevent screen cluttering as most users will be using the application on small screens.

## Project Constraints

The constraints on this project’s solution revolve around security.

The most impactful security issues are with iOS kill codes installed on all apple devices using iOS 9 onwards. Mobile device management (MDM) administrator can remotely enable lost mode and from there remotely lock and wipe devices. (2021 Apple Inc, 2021) It should be noted that this can be done without any agreement with user if desired from apple (from a technical perspective). This is an issue as if apple desired, they could wipe all apple devices used by Australian defence personnel. As the app aims to be used to coordinate defence personnel having a 3rd party with the power to disrupt communication would be a major security concern. To avoid this the app will not be designed for apple devices forcing users to have PED’s with operating systems that do not have iOS installed.

There are many security issues with using webs-apps that include but are not limited to cross website scripting, sql injection and key loggers for example. It will not be feasible to develop an application that is able to handle all these security issues to complete this project a mobile app will be developed as they are fully executable programs with far less security issues to handle as they don’t not share a runtime environment with other programs on a user’s system.

# Literature Review

The Australian Signals Directorate (ASD) has approved the cryptographic algorithms of RSA with 3072 bit key, AES-256 and SHA-2-384 for information classified up to TOP SECRET. (Australian Signals Directorate, 2021). Due to defences high security standards these algorithms will need to be implemented to allow for secure communication between user apps and defence servers in this project.

RSA is a widely used encryption and decryption algorithm but also an asymmetric algorithm meaning it has different keys for encryption and decryption allowing for safe key exchanges between parties. (A. A. Hasib and A. A. M. M. Haque, 2008)

An example of using RSA to secure connections for Secure e-learning web-based application has been implemented and shown capability to prevent data theft, data modification, data fabrication of an unauthorized user and prevents files from being readable both in storage and transmission through the encryption process. The RSA algorithm also had the benefit of authentication data to a specific user (Baihaqi, 2017). This example shows that the RSA algorithm is an effective algorithm to establish secure connection between parties and is an acceptable method to establish security keys and encrypt data over a network.

As stated, the RSA is an effective algorithm however there are disadvantages to using it. These being the time it takes to encrypt/decrypt data as well as the avalanche effect. The avalanche effect is simply how much the data will change once encrypted due to a small change in the original text. Out of the most used encryption algorithms (DES, 3DES, AES, Blowfish) RSA performed the worst having the least amount of change. Note that AES performed the best (Priyadarshini Patil, 2015). The RSA algorithm also is easily the worst algorithm to choose from when comparing encryption and decryption times as it will grow with the size of the data (over 2 seconds for 3MB of data) where other algorithms will not grow due to these circumstances such as AES for example (Priyadarshini Patil, 2015). This shows that using AES algorithm will allow for a far more responsive app as well as shown far greater changes from the original plaintext when encrypting data. For the project using AES will be a better choice over RSA however AES is a symmetrical algorithm creating a problem with secret key exchange between communicating parties.

To solve the issue with key exchange an implementation of using a combination of RSA and AES was used to secure electronic health record application. This method used the RSA algorithm to send the AES secret key over a network encrypted. This allowed a secure connection between party to be form using the AES algorithm without risk of the disclosure of the secret key (Wardhani, 2016). A similar method will be used in this project to create a connection between a server and client apps.

RSA and AES will be used to establish an encrypted communication between the server and user apps however authenticating users will be done using knowledge possessed by the user in the form of passwords. To prevent users’ passwords or other identifying knowledge being possessed by the server to be disclosed a hashing algorithm will needed to be used on user’s data. Approved hashing algorithms include SHA-2 with 384-to-512-bit outputs may be used for information with the classification of TOP SECRET (Australian Signals Directorate, 2021).

SHA-2 can protect user’s data as it transforms an input message into a 256 bits message. This transformation is one way, and the original message cannot be recreated from the resulting transformation (R. V. Mankar, 2013). The SHA-2 has shown a high level for randomness in tests being able to completely remove the original input and compress it down to the specified bits (Z. Al-Odat, 2019). These tests have shown that the SHA-2 hashing algorithm is a highly effective algorithm with no noticeable issues and is therefore the algorithm that will be used to secure passwords in the sever database for the project.

In the java standard libraries or API provided by Oracle for the Java platform includes built-in many of the most used cryptographic algorithms, including the RSA and AES encryption algorithm as well as the SHA message digest algorithm and key agreement algorithms (2021 Oracle, 2021). These libraries will allow the security required by defence to be implemented using the programming language java therefore java will be used to develop the project.

# Methodology

## Overarching approach

The overarching approach when designing and implementing this project will be the programming models of object-oriented programming (OOP) and an agile development methodology. Object-oriented programming or OOP is a model of programming that centres a piece of code around data and objects rather than functions and logic. In practice this will translate to encapsulating a work project as an object, along with encryption algorithms and users when writing the program. The reason for developing the project using this model is because it has the advantages of easily partition the code to allow for easier agile development as well as provide successors to the project an easy way to visualise the data structures of the project. The second key aspect of this project development methodology is an iterative approach to development. This will mean in practice that the project will need to be broken down into smaller parts that do not directly involve long term planning. The reason for this development methodology is centred around the ability to be flexible with client needs this is crucial as the client does not have any specific need but as been very general with the specification. This methodology will make responding to feed back more integral to the development process.

## Data flow description

### Overview

The project or application follows your typical client-server architecture. This model is a defined by having a server application host, deliver and manage the resources of a service in which many clients or remote process request and receive service from a centralised host or server. Client processors or User Applications provide an interface to allow for the visualisation of processed and stored data. Servers wait for request to arrive from User applications and responds to them. In the below image or figure 1 it shows a simplified version of how this application will work with this model. Multiple User application shared a common server and in return that server will handle multiple users handling their request. Note that User application are not in direct communication but only with the server application.

Diagram

Description automatically generated

Figure 1: Client-Server Architecture

### Server

In the below image or figure 2 there is a more detail diagram of how the server operates. The main or original thread (an execution sequence) the server begins execution on is called the connection listener this process starts by creating an interface with a database on the system the server program is running on represented by the orange box in the image. This database Connection interface object will be how client handler threads (represented in the yellow boxes) will get and update information on the database. After this object is created and pass all health checks required the connection listener will then enter an infinite loop in which it will listen on the designated port for incoming Connections. Each connection that is detected by the server will then generate a client handler thread which will execute in parallel with the connection listener and other client handler threads which will handle each unique connection or User Application.

Diagram

Description automatically generated

Figure 2: Server Architecture

In the below image or figure 3 the client handler function is detailed. After the connection listener has created a client handler to respond to a connection it will directly communicate with the connecting source. In practice these connections will be User Applications however it is possible for other sources that are not user applications to communicate with a client handler however if the outside communicating body does not respond in the desired way the connection will be terminated making it very difficult for non-user applications to access the system without source code knowledge. The client handler thread will start its execution by generating random and unique encryption keys for the duration of the session a user is connected to the server. The client handler will then send unencrypted the public key for an asymmetrical encryption algorithm (RSA) to the user application which will then send back a symmetrical key for general encryption (AES). Performing the key exchange will allow the two sources to communicate without threat of an outside party listening. Once this is achieved the client handler will enter an infinite loop referred to in diagram as the request cycle. In this cycle the client handler will wait for a message from the User application then decrypt the message then respond to the message by accessing the database for the requested information then send it back to the User application then it will return to waiting for a message from the user application. This cycle will continue until the connection is terminated from the user application which will cause the client handler to terminate.

Diagram

Description automatically generated

Figure 3: Client Handler Architecture

### Client apps

Each user will have an application or app in which to access the server and databases information as previously stated. In the below image or figure 4 there is a user application diagram which details how a user will navigate their way through the application. User applications act as way to visualise data requested from the server. The first page a user will see when entering the application will be a user authentication page (login screen). This screen will provide the user with the means to login into the app or register themselves on the system. The user authentication and user registration functionality’s can be thought of the first layer which users will be restricted at. If a user is unable to identify themselves to the server and receive a valid message allowing them access the user application will prevent a user from making any request for data from the server. If a user is allowed further access the user will be presented with a list of projects the user is a part of, the ability to create new project and two more pieces of functionality which is yet to be implemented being individual messaging and personal notes. To access the next tier of information the user must be assign to a project either by creating one themselves or being assigned into one by a project admin. A list of projects a user has access to will be listed on a projects list tab. This list is the method to gain access to that project specific information which when clicked on will allow the user to access that project information. A projects page will then appear to the user which will have 4 tabs being a home page, overview page, add users page and add task page. This layer is the final layer of restriction on users. To access project information the user will navigate to the project overview tab which will visualise the project information as well as provide a method to access task information and a way to visualise sub tasks.

Diagram

Description automatically generated

Figure 4: User application diagram

## Software tools used

### Programming language selection

The programming language selected for this project is Java. Java was chosen for this project due to the security benefits that the java runtime environment has. The java runtime environment will create a more secure program since a java program will run independently and separate from the other programs on a system and cannot be access by the kernel or other runtime environment that many programming language can be accessed by. The run time environment also makes accessing it memory by directly accessing it location near impossible due to the fact java does not use pointer (memory addresses) to store data. Other than security benefits java has there are many other reasons why it was chosen for this project these include it is an object-oriented programming language, multithreading support (optimization) and as it was once androids’ official language meaning there will be years of support behind this platform.

### Development Environment

To streamline development the use of an integrated development environment needed to be selected. The program that was chosen was android studio. Android studio is an ide that is specifically focus on mobile development and as Java was once android official language there remains a high level of functionality for the language. These functionality’s include tools to make compiling code easier/faster as well as debugging tools built in.

### Testing

To test the application android studio has a range of phone emulators available for download to test the program on. Downloading a range to test the application on with different size screen would be used to test the application. When running emulators or connecting physical phones with the ide live logs of a device’s activity can be seen which will be used debug issues during development.

### Building

To assist with building and compiling the written java code for this application the build automation tool of Gradle was used.

# Results and Discussion

The results of the above planning were implemented over the duration of the last year (2021) and were able to produce an application with basic functionality although limit with its capability or the high-level objective of capability meaning that the application will need to be further developed or continued to meet the needs of the client. The project in its current state has been able to implement a functioning client-server application capable of handling multiple clients simultaneously with the security measures required to satisfy a classification level of official and the beginnings of implementing the capability required by the client have been started.

After a rudimentary server and client applications were created and communicating the encryption algorithms detailed under low level objectives were implemented in the method investigated in the literature review section. In practice this method of using RSA asymmetrical encryption to exchange AES encryption key for general encryption over socket connection was an effective method of establishing a secure connection between the client applications and the server. Messages sent between the server and client apps were seen to have been completely transformed as intended and the encryption and decryption times had neglectable processing time and processing requirements. The actual key exchange process was also effective as the negatives of using the RSA encryption algorithms including processing time and poor data transformation where minimised to a neglectable degree due to the small amount of data needing to be encrypted using it. Although not in the initial research this method also provided the benefits of having unique and randomly generated encryption keys per connection ensuring that communication seen by outsiders will have no similarity. Using SHA with salting was also implemented as a one-way encryption to store password data. In the database no noticeable negatives were observed during development and password data was visibly transformed from their original form. With the required security algorithms implemented and the security benefits of creating a java application the high-level objective of security has been achieved.

The second high level objective of capability was not fully implemented the development of adding the features required by the client are in the programs current state are limited. The applications current state is capable of authenticating and registering new users to the server as well as the ability to create projects and assign user within the system to those projects. These projects can be assigned tasks with some ability to alter, assign and change information of those tasks. This information can be visualised to users in the form of a Gantt chart. A more detail explanation of project capability can be found under the heading timeline below.

# Conclusion

In conclusion the project will need to be continued on if the client’s needs are going to be satisfied specifically the app still doesn’t have the required capability that is needed. The application in its current state has satisfied the security requirement needed and has started to implement the desired capability the client needs however is far from complete. Overall, the work done on the application has created an application with some basic functionality that will meet the needs of the clients required security concerns as well has provided a structure and functions that will assist in future development of the application.

### Future works

The project will need to be continued on to complete and will therefore require future development this will include further development on the applications capability, deployment and potentially UX testing of the final application. Currently the programs capability will need to be further developed to meet the needs of the client this will entail successors to the project to work with the client to produce a result that is acceptable. Once the application is at a state ready for deployment there are many issues associated with the process that successors to the project will need to consider. The major issues are with the scalability of the application. Currently the server does all major processing and will therefore slow down in proportion to the number of user application making request to it. This could potentially be an issue as the limit of the server’s ability is currently unknown. These issues could potentially be a major issue as the server scalability is currently depended on the server’s CPU’s number of cores and clock speed and as this is a hardware issues there is a limit on how powerful these devices are capable of. If the server is unable to handle the demand, there are three possible solutions that could be considered. Firstly, the client could set up the infrastructure required to turn the server into a cloud computing application and scales the application with the use of instances like how AWS and azure have their systems set up. A second possible solution could be to use distributed computing methods (such as MPI for example) to apply multiple processors to the servers computing power allowing for the ability to scale up the processing power of the server. The last and most probability the most feasible solution is set up multiple servers for client apps to access giving an option at the user application level to decide which server they will be connecting with. This would distribute processing to multiple servers which could be working location specific. Once the project is capable of being deployed the benefit of conducting UX (user experience) testing/research could prove to be an important aspect of the process. By performing theses test the app could be updated to provide a better service to the clients target audience as well as find unforeseen issues.

# Timeline

## Socket connections.

The projects development stated with creating two rudimentary programs being the server and user app applications. To create a socket connection, it was decided that these programs would communicate over the internet using TCP/IP (transmission control protocol/internet protocol). TCP was chosen due to it being a standardised protocol that governs communications among computers on the internet that is used on every device connected to the internet. Note that port 12345 was selected to be used for the sever, this port was selected as it is used for nothing that a dedicated server would use. An unused port was selected to prevent other programs from congesting the sever program.

## Build process setup

Once a simple program that could both send and receive messages was created a method to compile the written program was needed. The open-source software Gradle was selected to compile code for both server and user apps. Gradle was chosen due to it being the compiler recommended by android and because it is available on nearly all ide’s (integrated developer environment).

## Testing methodology setup

After a rudimentary server and mobile app could be compiled several mobile phone emulators with different sized screens were downloaded from android to use for testing the mobile app (used to check if GUI would work on all sized screens) during this time a physical phone was also sourced to use for testing. Also, during this time debugging tools where explored.

## Encryption and key exchange

At this point the key exchange and general encryption protocols where implemented. For key exchange RSA with a key size of 3072 bit was selected. General encryption functions were implemented to allow for future use in development.

## Multi-threaded server

At this point the server could only handle one user at a time, to allow multiple users the sever was redesigned to create a new tread per connection to it (allow programs to execute code in parallel). This also had the benefit of creating a unique encryption key per user app connections.

## Login functionality

The first thing after the programs were communicating fully encrypted was the ability to authenticate what the sever was communicating with, which was done in the form of a login page. To do this a rudimentary database was created to store username and passwords and sever connection to said database was tested as well as the ability to query the database (how to get and change information on the database from the sever program). A login page for the user app was also created to allow users to send inputted information over to the server then check it against the record in its database and send a response back to the user app. After this login functionality was implemented. A visual of the login page can be seen in figure 5 below.

Graphical user interface, application

Description automatically generated

Figure 5: From left to right - Login Page Figure, Registration Page, Create New Project Page, Projects Page

## User registration

A page to register new users to the database was created, this worked the same way as the login page except it will add information to the database instead of authenticating a user. If/when this app is implemented the server should probability have some way of authenticating new users such as checking pmkeys or needing a defence email to add a new user to simply prevent non-defence personnel from even accessing the network. Currently program just excepts all new entries if username is not already in database. A visual of the registration page can be seen in figure 5 above.

## SHA encryption

Storing passwords and other identifying information in plain text on severs would allow anyone with database access to looked at user’s information, to prevent this a one-way encryption is used on this sort of information. The encryption algorithm of SHA with 384-bit output with salting (a key is used during encryption) was used as recommended by defence for information up to top secret.

## Home UI skeleton

Once a user has been authenticated by the sever the user app will enter its main navigation area. It was decided to use a bottom navigation bar with four possible displays being a projects page, create project page, messages, and diary tabs to navigate to the future planed feature of the app. At this point each of the displays where empty/had placeholding images.

## Database design

The database in its current form would not be capable of storing the necessary information needed to represent a user’s work-project. At this point creating some functionality to end users was being planned out and the database needed to go through a redesign to represent the information needed for a user’s work project.

## Create project page.

Once the database was redesigned a way to allow users to store information was needed. Under the create new project tab from the bottom navigation option a create new projects page was created seen in figure 5. This page functioned like the registration page as it would send information to the server to be stored.

## Project Selection page.

The project selection page is the tab of the bottom navigation options that a user will be sent to after login. This page displays each project the user is connected to (as indicated by the server) and can be scrolled through. The plan is to make each of these projects a button which will send the user to a projects individual information page however this is not yet implemented.

## Project Visualisation

Below in figure 6 the image on the right side shows the projects tab which displays in a scrollable list all the projects the currently log in user is associated with and the image on the right shows the page a user is redirected to when one of the projects are selected. This area that users are redirected to is made up of 4 tabs used to interacted with the selected project. The tabs being home, overview, add user and add task. Note that the home page hasn’t been implemented yet. This tab is intended to be used to display project updates and team leader messages to the team.

Graphical user interface, application

Description automatically generated

Figure 6: Projects page (left), Project Home page (right)

The first tab that was implemented was the add users tab. This tab can be seen in figure 7 below. This tab will display all user that are not currently associated with the project and when a user is selected that user will be given access to the project. There is also a search tab at the top of this page which can be used to filter out users. After a recent meeting with the client this tab will need to be updated to only be accessible to team leaders and users accounts with increase access.

A screenshot of a phone

Description automatically generated with low confidence

Figure 7: Add Users Tab

The next tab that was implemented was the add task tab which can be seen below in figure 8 the left image shows the top of the page, the middle the bottom and the right shows how dates are selected. This tab is used to create a task for the project and upload the information to the server.

Graphical user interface, application

Description automatically generated

Figure 8: Add Task Tab

The next tab that was implemented was the project overview tab as seen below in figure 9. This tab is used to display task information of the selected project to users in the form of a Gantt chart. The page is split with tasks listed vertically on the left (can be scrolled through) and right of that a table that can be scrolled horizontal where each column will represent a time. At the top of the page there is a button that acts as a title and a way to change the columns between weeks and days. The columns represent either a day or a week the first being the project creation date or earliest task start date in project. The current day/week can be found by finding the columns that have a purple boarder around them (note that there are some missing graphics as seen in figure 9 as orange is yet to have a boarder around it and there isn’t a red block with a boarder either). It is planned that the top number column will be able to be click on and the date it represents will be displayed. Tasks are ordered from top to bottom first their status as tasks that are in progress will be first in the list, then followed by completed then cancelled. After task status tasks are ordered by their planned finished date. The colours that fill the chart are blue for in progress, orange for task running overtime, green for complete and red for cancelled.

A screenshot of a phone

Description automatically generated with medium confidence

Figure 9: Project Overview Tab

Below in figure 10 is an example of the task info page which is accessible by clicking on the tasks in the left most column in figure 4. This page is used to display task specific information and the ability to update task information. Currently only the ability to update the status is available however it is planned to create a task specific log and the ability to change the assigned user to the project.

Graphical user interface, text, application, Teams

Description automatically generated

Figure 10: Task info Page

The task info page seen in figure 10 was updated with the ability to add minor tasks to task and an option to view a Gantt chart like in figure 9 was added to visualise the minor tasks for a task.

# References

2021 Apple Inc. (2021, 4 11). *Deployment Reference for iPhone and iPad*. Retrieved from Lost Mode, remote wipe and remote lock: https://support.apple.com/en-gb/guide/deployment-reference-ios/apd713df1b14/web

2021 Oracle. (2021, 4 20). *Java Platform, Standard Edition Security Developer’s Guide*. Retrieved from Documentation: https://docs.oracle.com/javase/9/security/java-security-overview1.htm#JSSEC-GUID-2EF0B3B8-9F3A-41CF-A7DA-63DB52180084

A. A. Hasib and A. A. M. M. Haque. (2008). A Comparative Study of the Performance and Security Issues of AES and RSA Cryptography,. *008 Third International Conference on Convergence and Hybrid Information Technology* (pp. pp. 505-510). Busan, Korea (South): IEEE. doi:doi: 10.1109/ICCIT.2008.179.

Australian Signals Directorate. (2021, March). Australian Government Information Security Manual . Canberra, ACT, Australia .

Baihaqi, O. C. ( 2017). Implementation of RSA 2048-bit and AES 128-bit for Secure e-learning web-based application. *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)* (pp. pp. 1-5). Lombok, Indonesia: IEEE. doi:doi: 10.1109/TSSA.2017.8272903.

Priyadarshini Patil, P. N. (2015). A Comprehensive Evaluation of Cryptographic Algorithms: DES,. *International Conference on Information Security & Privacy (ICISP2015).* *78*, pp. 617 – 624. Nagpur, INDIA: Procedia Computer Science. doi:https://doi.org/10.1016/j.procs.2016.02.108.

R. V. Mankar, S. I. (2013). C Implementation of SHA-256 Algorithm. *International Journal of Emerging Technology and Advanced Engineering*, pp. 167-170. Retrieved from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.7088&rep=rep1&type=pdf

Wardhani, M. A. (2016). Implementation of RSA 2048-bit and AES 256-bit with digital signature for secure electronic health record application. *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA),* (pp. pp. 387-392). Lombok, Indonesia, : IEEE. doi:doi: 10.1109/ISITIA.2016.7828691.

Z. Al-Odat, A. A. (2019). Randomness Analyses of the Secure Hash Algorithms, SHA-1, SHA-2 and Modified SHA. *2019 International Conference on Frontiers of Information Technology (FIT)* (pp. pp. 316-3165). Islamabad, Pakistan: IEEE. doi:doi: 10.1109/FIT47737.2019.00066.

# Appendices

## Source code

### Server

#### Main

package com.company;  
  
import javax.crypto.SecretKey;  
import javax.crypto.spec.SecretKeySpec;  
import java.io.\*;  
import java.net.ServerSocket;  
import java.net.Socket;  
import java.util.Base64;  
import java.util.concurrent.Semaphore;  
  
public class Main {  
 public static RSA *rsa*;  
 public static server\_db\_conection *main\_con*;  
  
  
  
 public static void main(String[] args) {  
 int PORT = 12345;  
 //database connection  
 try{  
 //create connection  
 *main\_con* = new server\_db\_conection("jdbc:mariadb://localhost:1433","egh400\_test","root","jpate101");  
 }catch(Exception e){  
 System.*out*.println("error: unable to connect to database");  
 //e.printStackTrace();  
 }  
  
 //  
 //start server  
 ServerSocket server = null;  
 try {  
 server = new ServerSocket(PORT);  
 server.setReuseAddress(true);  
 System.*out*.println(server.getLocalSocketAddress());  
 //server.getLocalSocketAddress();  
  
 // The main thread is just accepting new connections  
 while (true) {  
 Socket client = server.accept();  
 System.*out*.println("New client connected " + client.getInetAddress().getHostAddress());  
 //begin client handling  
 ClientHandler clientSock = new ClientHandler(client);  
 // The background thread will handle each client separately  
 new Thread(clientSock).start();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 if (server != null) {  
 try {  
 server.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
  
 */\*\*  
 \* handles client requests  
 \*/* private static class ClientHandler implements Runnable {  
  
 private final Socket clientSocket;  
 SHA sha = new SHA();  
 private String Test\_xml = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<billboard background=\"#7F3FBF\">\n<message>Billboard with custom background and default-coloured message</message>\n</billboard>";  
 static Semaphore *semaphore* = new Semaphore(1);  
 public ClientHandler(Socket socket) {  
 this.clientSocket = socket;  
 }  
  
  
 @Override  
 public void run() {  
 *rsa* = new RSA();  
 try{  
 *rsa*.*getRSAKeys*();  
 }catch(Exception e){  
 e.printStackTrace();  
 }  
 byte[] key\_en = *rsa*.*privateKey*.getEncoded();  
 DataOutputStream dOut = null;  
 try {  
 String line;  
  
 PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);  
 BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));  
  
 dOut = new DataOutputStream(clientSocket.getOutputStream());  
 dOut.writeInt(key\_en.length); // write length of the message  
 DataInputStream dIn = new DataInputStream(clientSocket.getInputStream());  
 dOut.write(key\_en);  
  
 int length = dIn.readInt();// read length of incoming message  
 byte[] message = new byte[length];  
 if(length>0) {  
 dIn.readFully(message, 0, message.length); // read the message  
 }  
  
 String message\_s = *rsa*.*decryptMessage\_cipher*(message,*rsa*.*publicKey*);  
 //System.out.printf("Sent from the client: %s\n", message\_s);  
  
 //gen aes  
 AES aes = new AES();  
  
 length = dIn.readInt();// read length of incoming message  
 message = new byte[length];  
 if(length>0) {  
 dIn.readFully(message, 0, message.length); // read the message  
 }  
  
 //rsa decryption  
 String aes\_key\_en = *rsa*.*decryptMessage\_cipher*(message,*rsa*.*publicKey*);  
  
 // decode the base64 encoded string  
 byte[] decodedKey = Base64.*getDecoder*().decode(aes\_key\_en);  
 // rebuild key using SecretKeySpec  
 SecretKey originalKey = new SecretKeySpec(decodedKey, 0, decodedKey.length, "AES");  
 //System.out.println(originalKey);  
  
 aes.*secretKey* = originalKey;  
  
 String success = "success";  
 dOut.writeUTF(aes.*encrypt*(success));  
  
  
 line = "empty";  
  
 while (!"exit".equalsIgnoreCase(line)) {  
 line = in.readLine();  
 if(line == null || line.equals("empty")){  
 line = "empty";  
 }else{  
 System.*out*.printf("Sent from the client: %s\n", line);  
 }  
 line = aes.*decrypt*(line);  
  
 //out.println(line);  
  
 if(line.equals(("LOGIN\_request"))){  
 System.*out*.println("login request rev");  
 String LOGIN\_request\_user = aes.*decrypt*(in.readLine());  
 String LOGIN\_request\_pass = aes.*decrypt*(in.readLine());  
 String From\_db = *main\_con*.*get\_user\_salt*(LOGIN\_request\_user);  
 if(From\_db.equals("SQL\_ERROR")){  
 System.*out*.println("F");  
 out.println(aes.*encrypt*("F"));  
 }else{  
 byte[] user\_salt = Base64.*getDecoder*().decode(From\_db);  
 String user\_pass\_db = *main\_con*.*get\_user\_Pass*(LOGIN\_request\_user);  
  
  
 if(sha.*encrypt*(LOGIN\_request\_pass,user\_salt).equals(user\_pass\_db)){  
 System.*out*.println("T");  
 out.println(aes.*encrypt*("T"));  
 }else{  
 System.*out*.println("F");  
 out.println(aes.*encrypt*("F"));  
 }  
 }  
  
 }  
 if(line.equals(("NEW\_USER\_request"))){  
 System.*out*.println("NEW\_USER\_request");  
 String User = aes.*decrypt*(in.readLine());  
 String Pass = aes.*decrypt*(in.readLine());  
  
 System.*out*.println(User);  
 System.*out*.println(Pass);  
  
 byte[] salt = sha.*getSalt*();  
 String str = Base64.*getEncoder*().encodeToString(salt);  
 if(*main\_con*.*Insert\_New\_Current\_user*(User,str,sha.*encrypt*(Pass,salt))){  
 out.println(aes.*encrypt*("T"));  
 }else{  
 out.println(aes.*encrypt*("F"));  
 }  
  
 }  
 if(line.equals(("CREATE\_NEW\_PROJECT"))){  
 System.*out*.println("CREATE\_NEW\_PROJECT");  
 String P\_Name = aes.*decrypt*(in.readLine());  
 String Description = aes.*decrypt*(in.readLine());  
 String User = aes.*decrypt*(in.readLine());  
 //search db for same entry  
 Boolean check\_1 = server\_db\_conection.*check\_for\_project*(P\_Name);  
 //if success  
 System.*out*.println(check\_1);  
 if(check\_1 == true){  
 System.*out*.println("here true");  
 out.println(aes.*encrypt*("T"));  
 server\_db\_conection.*insert\_new\_project*(P\_Name,Description,User);  
 }else{  
 System.*out*.println("here false");  
 out.println(aes.*encrypt*("F"));  
 }  
 }  
 if(line.equals(("GET\_USER\_PROJECTS"))){  
 System.*out*.println("GET\_USER\_PROJECTS");  
 String User = aes.*decrypt*(in.readLine());  
 String[] projects = server\_db\_conection.*get\_user\_projects*(User);  
 for (int i = 0; i < projects.length; i++) {  
 //System.out.println(projects[i]);  
 out.println(aes.*encrypt*(projects[i]));  
 }  
 out.println(aes.*encrypt*("end\_of\_String\_array\_n10193197"));  
 }  
 if(line.equals(("GET\_ALL\_USERS"))){  
 System.*out*.println("GET\_ALL\_USERS");  
 String User = aes.*decrypt*(in.readLine());  
 String[] Users = server\_db\_conection.*get\_all\_users*(User);  
  
 for (int i = 0; i < Users.length; i++) {  
 //System.out.println(projects[i]);  
 out.println(aes.*encrypt*(Users[i]));  
 }  
 out.println(aes.*encrypt*("end\_of\_String\_array\_n10193197"));  
 }  
 if(line.equals(("ASSIGN\_USER\_TO\_PROJECT"))){  
 System.*out*.println("ASSIGN\_USER\_TO\_PROJECT");  
 String User = aes.*decrypt*(in.readLine());  
 String New\_User = aes.*decrypt*(in.readLine());  
 String Project = aes.*decrypt*(in.readLine());  
  
 System.*out*.println(User);  
 System.*out*.println(New\_User);  
 System.*out*.println(Project);  
  
 String check = server\_db\_conection.*insert\_user\_into\_project*(User,New\_User,Project);  
 if(check.equals("T")){  
 out.println(aes.*encrypt*("T"));  
 } else if (check.equals("error check permission")) {  
 out.println(aes.*encrypt*("error check permission"));  
 }else if (check.equals("error inserting new user")){  
 out.println(aes.*encrypt*("error inserting new user"));  
 } else if (check.equals("need permission")){  
 out.println(aes.*encrypt*("need permission"));  
 }else{  
 out.println(aes.*encrypt*("unknown error"));  
 }  
 }  
 if(line.equals(("GET\_ALL\_USERS\_IN\_PROJECT"))){  
 System.*out*.println("GET\_ALL\_USERS\_IN\_PROJECT");  
 String project = aes.*decrypt*(in.readLine());  
 String[] Users = server\_db\_conection.*get\_all\_users\_in\_project*(project);  
  
 for (int i = 0; i < Users.length; i++) {  
 //System.out.println(projects[i]);  
 out.println(aes.*encrypt*(Users[i]));  
 }  
 out.println(aes.*encrypt*("end\_of\_String\_array\_n10193197"));  
  
 }  
 if(line.equals(("INSERT\_NEW\_TASK"))){  
 System.*out*.println("INSERT\_NEW\_TASK");  
 String TaskName = aes.*decrypt*(in.readLine());  
 String TaskDes = aes.*decrypt*(in.readLine());  
 String Task\_start = aes.*decrypt*(in.readLine());  
 String Task\_end = aes.*decrypt*(in.readLine());  
 String Task\_User = aes.*decrypt*(in.readLine());  
 String Project = aes.*decrypt*(in.readLine());  
 String Created\_by = aes.*decrypt*(in.readLine());  
  
 boolean response = server\_db\_conection.*insert\_task\_into\_project*(TaskName,Project,Task\_User,Created\_by,Task\_start,Task\_end,TaskDes);  
 System.*out*.println(response);  
 if(response){  
 out.println(aes.*encrypt*("T"));  
 }else{  
 out.println(aes.*encrypt*("F"));  
 }  
 }  
 if(line.equals(("GET\_PROJECT\_TASK\_INFORMATION"))){  
 System.*out*.println("GET\_PROJECT\_TASK\_INFORMATION");  
 String Project = aes.*decrypt*(in.readLine());  
  
 String[] results = server\_db\_conection.*get\_project\_tasks*(Project);  
  
 for (int i = 0; i < results.length; i++) {  
 //System.out.println(projects[i]);  
 out.println(aes.*encrypt*(results[i]));  
 }  
 out.println(aes.*encrypt*("end\_of\_String\_array\_n10193197"));  
 }  
 if(line.equals(("GET\_TASK\_INFO"))){  
 System.*out*.println("GET\_TASK\_INFO");  
 String Project = aes.*decrypt*(in.readLine());  
 String Task = aes.*decrypt*(in.readLine());  
  
 System.*out*.println(Project);  
 System.*out*.println(Task);  
  
 String[] results = server\_db\_conection.*GET\_TASK\_INFO*(Project,Task);  
  
 out.println(aes.*encrypt*(results[0]));  
 out.println(aes.*encrypt*(results[1]));  
 out.println(aes.*encrypt*(results[2]));  
 out.println(aes.*encrypt*(results[3]));  
 out.println(aes.*encrypt*(results[4]));  
 out.println(aes.*encrypt*(results[5]));  
 }  
 if(line.equals(("UPDATE\_TASK\_STATUS"))){  
 System.*out*.println("UPDATE\_TASK\_STATUS");  
 String Project = aes.*decrypt*(in.readLine());  
 String Task = aes.*decrypt*(in.readLine());  
 String Status = aes.*decrypt*(in.readLine());  
 if(server\_db\_conection.*UPDATE\_TASK\_STATUS*(Project,Task,Status)){  
 out.println(aes.*encrypt*("T"));  
 }else{  
 out.println(aes.*encrypt*("F"));  
 }  
  
 }  
 if(line.equals(("INSERT\_NEW\_MINOR\_TASK"))){  
 System.*out*.println("INSERT\_NEW\_MINOR\_TASK");  
 String RootTask = aes.*decrypt*(in.readLine());  
 System.*out*.println(RootTask);  
 String TaskName = aes.*decrypt*(in.readLine());  
 String TaskDes = aes.*decrypt*(in.readLine());  
 String Task\_start = aes.*decrypt*(in.readLine());  
 String Task\_end = aes.*decrypt*(in.readLine());  
 String Task\_User = aes.*decrypt*(in.readLine());  
 String Project = aes.*decrypt*(in.readLine());  
 String Created\_by = aes.*decrypt*(in.readLine());  
  
  
 boolean response = server\_db\_conection.*insert\_minor\_task\_into\_project*(RootTask,TaskName,Project,Task\_User,Created\_by,Task\_start,Task\_end,TaskDes);  
 System.*out*.println(response);  
 if(response){  
 out.println(aes.*encrypt*("T"));  
 }else{  
 out.println(aes.*encrypt*("F"));  
 }  
  
  
 }  
 if(line.equals(("GET\_PROJECT\_MINOR\_TASK\_INFORMATION"))){  
 System.*out*.println("GET\_PROJECT\_MINOR\_TASK\_INFORMATION");  
 String Project = aes.*decrypt*(in.readLine());  
 String root\_task = aes.*decrypt*(in.readLine());  
  
 String[] results = server\_db\_conection.*get\_project\_tasks\_minor*(Project,root\_task);  
  
 for (int i = 0; i < results.length; i++) {  
 System.*out*.println(results[i]);  
 out.println(aes.*encrypt*(results[i]));  
 }  
 System.*out*.println("GET\_PROJECT\_MINOR\_TASK\_INFORMATION END LOOP");  
 out.println(aes.*encrypt*("end\_of\_String\_array\_n10193197"));  
 System.*out*.println("GET\_PROJECT\_MINOR\_TASK\_INFORMATION END");  
 }  
 }  
  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 } catch (Exception e) {  
 e.printStackTrace();  
 } finally {  
 try {  
 clientSocket.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 System.*out*.println("connection terminated");  
 }  
 }  
 }  
}

#### AES

package com.company;  
  
import javax.crypto.Cipher;  
import javax.crypto.KeyGenerator;  
import javax.crypto.SecretKey;  
import javax.crypto.spec.SecretKeySpec;  
import java.io.UnsupportedEncodingException;  
import java.util.Base64;  
  
public class AES {  
 public static SecretKey *secretKey*;  
 public static byte[] *secretKey\_encoded*;  
  
 public static void set\_server\_keys(String aes\_key) throws UnsupportedEncodingException {  
 *secretKey\_encoded* = aes\_key.getBytes();  
 *secretKey* = new SecretKeySpec(*secretKey\_encoded*, 0, *secretKey\_encoded*.length, "AES");  
 }  
  
 public static void set\_server\_keys\_2(Object aes\_key) throws UnsupportedEncodingException {  
 *secretKey* = (SecretKey) aes\_key;  
 }  
  
 public static void GenerateKeys() throws Exception {  
 KeyGenerator keyGen = KeyGenerator.*getInstance*("AES/ECB/PKCS5Padding");  
 keyGen.init(256); // for example  
 *secretKey* = keyGen.generateKey();  
 *secretKey\_encoded* = *secretKey*.getEncoded();  
 }  
  
 public static String encrypt(String strToEncrypt)  
 {  
 try  
 {  
 Cipher cipher = Cipher.*getInstance*("AES/ECB/PKCS5Padding");  
 cipher.init(Cipher.*ENCRYPT\_MODE*, *secretKey*);  
 return Base64.*getEncoder*().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-8")));  
 }  
 catch (Exception e)  
 {  
 System.*out*.println("Error while encrypting: " + e.toString());  
 }  
 return null;  
 }  
  
 public static String decrypt(String strToDecrypt)  
 {  
 try  
 {  
 Cipher cipher = Cipher.*getInstance*("AES/ECB/PKCS5PADDING");  
 cipher.init(Cipher.*DECRYPT\_MODE*, *secretKey*);  
 return new String(cipher.doFinal(Base64.*getDecoder*().decode(strToDecrypt)));  
 }  
 catch (Exception e)  
 {  
 System.*out*.println("Error while decrypting: " + e.toString());  
 }  
 return null;  
 }  
}

#### RSA

package com.company;  
  
import javax.crypto.Cipher;  
import java.security.\*;  
import java.util.Base64;  
import java.util.HashMap;  
import java.util.Map;  
  
public class RSA {  
 public static PrivateKey *privateKey*;  
 public static PublicKey *publicKey*;  
 /\*  
 public static byte[] encryptMessage\_cipher(String plainText, PrivateKey ret) throws Exception {  
 Cipher cipher = Cipher.getInstance("RSA");  
 cipher.init(Cipher.ENCRYPT\_MODE, ret);  
  
 return Base64.getEncoder().encodeToString(cipher.doFinal(plainText.getBytes("UTF-8"))).getBytes();  
 }  
  
 \*/  
  
 public static String decryptMessage\_cipher(byte[] encryptedText , PublicKey publicKey1) throws Exception {  
 //System.out.println(publicKey);  
  
 Cipher cipher = Cipher.*getInstance*("RSA/ECB/PKCS1Padding");  
 cipher.init(Cipher.*DECRYPT\_MODE*, *publicKey*);  
 return new String(cipher.doFinal(Base64.*getDecoder*().decode(encryptedText)));  
 //return new String(plainText);  
 }  
  
 public static void getRSAKeys() throws Exception {  
 KeyPairGenerator keyPairGenerator = KeyPairGenerator.*getInstance*("RSA");  
 keyPairGenerator.initialize(3072);  
 KeyPair keyPair = keyPairGenerator.generateKeyPair();  
 *privateKey* = keyPair.getPrivate();  
 *publicKey* = keyPair.getPublic();  
  
 Map<String, Object> keys = new HashMap<String,Object>();  
 keys.put("private", *privateKey*);  
 keys.put("public", *publicKey*);  
 }  
}

#### SHA

package com.company;  
import java.math.BigInteger;  
import java.security.MessageDigest;  
import java.security.NoSuchAlgorithmException;  
  
import java.security.SecureRandom;  
import java.util.Base64;  
  
public class SHA {  
  
 public static void main(String[] args) throws NoSuchAlgorithmException  
 {  
 String passwordToHash = "password";  
 byte[] salt = *getSalt*();  
 System.*out*.println(passwordToHash);  
 System.*out*.println(salt);  
  
 String securePassword = *encrypt*(passwordToHash, salt);  
 System.*out*.println(securePassword);  
 System.*out*.println("\_\_\_\_\_\_\_\_\_");  
 String str = Base64.*getEncoder*().encodeToString(salt);  
 byte[] byteArr = Base64.*getDecoder*().decode(str);  
 //System.out.println(str);  
 //System.out.println(byteArr);  
  
 securePassword = *encrypt*(passwordToHash, salt);  
 System.*out*.println(securePassword);  
  
 }  
  
 public static String encrypt(String passwordToHash, byte[] salt)  
 {  
 String generatedPassword = null;  
 try {  
 MessageDigest md = MessageDigest.*getInstance*("SHA-384");  
 md.update(salt);  
 byte[] bytes = md.digest(passwordToHash.getBytes());  
 StringBuilder sb = new StringBuilder();  
 for(int i=0; i< bytes.length ;i++)  
 {  
 sb.append(Integer.*toString*((bytes[i] & 0xff) + 0x100, 16).substring(1));  
 }  
 generatedPassword = sb.toString();  
 }  
 catch (NoSuchAlgorithmException e)  
 {  
 e.printStackTrace();  
 }  
 return generatedPassword;  
 }  
  
 //Add salt  
 public static byte[] getSalt() throws NoSuchAlgorithmException  
 {  
 SecureRandom sr = SecureRandom.*getInstance*("SHA1PRNG");  
 byte[] salt = new byte[16];  
 sr.nextBytes(salt);  
 return salt;  
 }  
}

#### Server\_db\_connections

package com.company;  
  
import java.io.InputStream;  
import java.io.UnsupportedEncodingException;  
import java.sql.\*;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Date;  
  
public class server\_db\_conection {  
  
 */\*\*  
 \* stores database connection  
 \*/* public static Connection *myConn*;  
 //Connection myConn;  
  
 */\*\*  
 \* create server connection to db  
 \** ***@param*** *dbURL the URL for the Database  
 \** ***@param*** *dbName the database name to check  
 \** ***@param*** *dbUser the username of the admin  
 \** ***@param*** *dbPass the password of the admin  
 \*/* public server\_db\_conection(String dbURL, String dbName,String dbUser,String dbPass){  
 try {  
 *myConn* = DriverManager.*getConnection*(dbURL + "/" + dbName, dbUser, dbPass);  
 System.*out*.println(dbURL + "/" + dbName+" "+dbUser + dbPass);  
 System.*out*.println("connection successful");  
 }catch(Exception db){  
 db.printStackTrace();  
 }  
  
 }  
  
 public static Boolean Insert\_New\_Current\_user(String user, String salt, String pass) {  
 //Connection con = DriverManager.getConnection("jdbc:mariadb://localhost:1433/mariadb", "root", "password");  
 try{  
 Statement stmt = *myConn*.createStatement();  
  
 String query = "INSERT INTO users (ID, salt, Password) VALUES (\""+user+"\",\""+salt+"\", \""+pass+"\");";  
 ResultSet rs = stmt.executeQuery(query);  
 return true;  
 }catch (SQLException e ){  
 return false;  
 }  
  
  
 }  
  
 public static String get\_user\_salt(String user) throws UnsupportedEncodingException {  
 try{  
 Statement stmt = *myConn*.createStatement();  
 String query = "select salt from users where ID=\""+user+"\"";  
 ResultSet rs = stmt.executeQuery(query);  
 //System.out.println(rs);  
 rs.next();  
 //byte[] test = rs.getBytes("salt");  
 String test = rs.getString("salt");  
 //System.out.println(test);  
 return test;  
  
 }catch(SQLException e ){  
 return "SQL\_ERROR";  
 }  
  
  
 }  
 public static String get\_user\_Pass(String user) throws SQLException, UnsupportedEncodingException {  
  
 Statement stmt = *myConn*.createStatement();  
 String query = "select Password from users where ID=\""+user+"\"";  
 ResultSet rs = stmt.executeQuery(query);  
 //System.out.println(rs);  
 rs.next();  
 String test = rs.getString("Password");  
 return test;  
  
 }  
 public static Boolean check\_for\_project(String P\_name) throws SQLException {  
 Statement stmt = *myConn*.createStatement();  
 String query = "SELECT Project\_Name\n" +  
 "FROM projects\n" +  
 "WHERE EXISTS\n" +  
 "(SELECT Project\_Name FROM projects WHERE Project\_Name=\""+P\_name+"\");";  
 ResultSet rs = stmt.executeQuery(query);  
 try{  
 rs.next();  
 String test = rs.getString("Project\_Name");  
 //System.out.println(test);  
 }catch (SQLException e){  
 return true;  
 }  
 return false;  
 }  
 public static void insert\_new\_project(String P\_name,String Des,String User) throws SQLException {  
  
 SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");  
 Date date = new Date();  
 System.*out*.println(formatter.format(date));  
  
 String start\_D = formatter.format(date);  
 start\_D = "STR\_TO\_DATE(\'"+start\_D+"\',\'%d/%m/%Y\')";  
  
  
 Statement stmt = *myConn*.createStatement();  
 String query = "INSERT INTO projects (Project\_Name, Created\_By, Description,Project\_Creation\_date)\n" +  
 "VALUES (\""+P\_name+"\", \""+User+"\",\""+Des+"\","+start\_D+");";  
 ResultSet rs = stmt.executeQuery(query);  
 query = "INSERT INTO people\_per\_project (Project, User, Role)\n" +  
 "VALUES (\""+P\_name+"\", \""+User+"\",1);";  
 rs = stmt.executeQuery(query);  
 query = "INSERT INTO people\_per\_project (Project, User, Role)\n" +  
 "VALUES (\""+P\_name+"\", \""+"!No User Selected"+"\",2);";  
 rs = stmt.executeQuery(query); }  
 public static String[] get\_user\_projects(String User) throws SQLException {  
 Statement stmt = *myConn*.createStatement();  
 String query = "SELECT Project FROM people\_per\_project WHERE User=\""+User+"\"";  
 ResultSet rs = stmt.executeQuery(query);  
  
   
 int i = 0;  
 ArrayList list = new ArrayList();  
  
 while (rs.next()) {  
   
 String test = rs.getString("Project");  
 //System.out.println(test);  
 list.add(test);  
 }  
 String[] array = (String[]) list.toArray(new String[list.size()]);  
 //System.out.println(array.length);  
 return array;  
 }  
 public static String[] get\_all\_users(String User) throws SQLException {  
 Statement stmt = *myConn*.createStatement();  
 String query = "SELECT ID FROM users";  
 ResultSet rs = stmt.executeQuery(query);  
  
 int i = 0;  
 ArrayList list = new ArrayList();  
  
 while (rs.next()) {  
  
 String test = rs.getString("ID");  
 //System.out.println(test);  
 list.add(test);  
 }  
 String[] array = (String[]) list.toArray(new String[list.size()]);  
 //System.out.println(array[0]);  
 return array;  
 }  
  
 public static String insert\_user\_into\_project(String inserting\_User,String new\_User,String Project) {  
 //check if user has permission to add users to project  
 ResultSet rs;  
 ResultSet rs2;  
 Statement stmt;  
 try{  
 stmt = *myConn*.createStatement();  
 String query = "SELECT Role FROM people\_per\_project WHERE User=\""+inserting\_User+"\" AND Project=\""+Project+"\"";  
 rs = stmt.executeQuery(query);  
 //inset user  
 rs.next();  
 }catch (SQLException e){  
 return "error check permission";  
 }  
 try{  
 int Roledb = rs.getInt("Role");  
 System.*out*.println(Roledb);  
 if(Roledb == 1){  
 stmt = *myConn*.createStatement();  
 String query2 = "INSERT INTO people\_per\_project (Project, User, Role)" +  
 "VALUES (\""+Project+"\", \""+new\_User+"\",2);";  
 rs2 = stmt.executeQuery(query2);  
 return "T";  
 }else{  
 return "need permission";  
 }  
 }catch (SQLException e){  
 return "error inserting new user";  
 }  
  
 }  
  
 public static String[] get\_all\_users\_in\_project(String project) throws SQLException {  
 System.*out*.println(project);  
  
 Statement stmt = *myConn*.createStatement();  
 String query = "SELECT Project,User,Role FROM people\_per\_project WHERE Project=\""+project+"\"";  
 ResultSet rs = stmt.executeQuery(query);  
  
  
 int i = 0;  
 ArrayList list = new ArrayList();  
  
 while (rs.next()) {  
  
 String test = rs.getString("User");  
 //System.out.println(test);  
 list.add(test);  
 }  
 String[] array = (String[]) list.toArray(new String[list.size()]);  
 //System.out.println(array.length);  
 return array;  
 //String[] test = {"Hello", "World","test","testing"};  
 //return test;  
 }  
  
 public static boolean insert\_task\_into\_project(String Task\_name,String Project,String Assign\_User,String Created\_By,String start\_D,String end\_D,String Des) {  
 start\_D = start\_D.substring(7);  
 end\_D = end\_D.substring(5);  
  
 if (start\_D.charAt(1) == '/') start\_D = "0" + start\_D;  
 if (start\_D.charAt(4) == '/') start\_D = start\_D.substring(0,3) + "0" + start\_D.substring(3);  
  
 start\_D = "STR\_TO\_DATE(\'"+start\_D+"\',\'%d/%m/%Y\')";  
 end\_D = "STR\_TO\_DATE(\'"+end\_D+"\',\'%d/%m/%Y\')";  
  
  
  
  
 try{  
 Statement stmt = *myConn*.createStatement();  
  
 // eg TO\_DATE('17/12/2015', 'DD/MM/YYYY')  
 String query = "INSERT INTO tasks (Task\_name, Project, Assigned\_User,Created\_By,Status\_int,start\_date,end\_date,Description) VALUES (\""+Task\_name+"\",\""+Project+"\",\""+Assign\_User+"\",\""+Created\_By+"\",\""+String.*valueOf*(1)+"\","+start\_D+","+end\_D+",\""+Des+"\");";  
 ResultSet rs = stmt.executeQuery(query);  
 return true;  
 }catch (SQLException e ){  
 System.*out*.println(e);  
 return false;  
 }  
 /\*  
 status  
 1 = in Progress  
 2 = Resolved  
 3 = Canceled  
 \*/  
 }  
 public static String[] get\_project\_tasks(String Project) throws SQLException {  
 //project creation date  
 Statement stmt2 = *myConn*.createStatement();  
 String query2 = "SELECT Project\_Name,Project\_Creation\_date FROM projects WHERE Project\_Name=\""+Project+"\"";  
 ResultSet rs2 = stmt2.executeQuery(query2);  
  
 int i = 0;  
 ArrayList list = new ArrayList();  
  
 while (rs2.next()) {  
  
 String test = rs2.getString("Project\_Creation\_date");  
 System.*out*.println(test);  
 list.add(test);  
 }  
 //String[] array = (String[]) list.toArray(new String[list.size()]);  
 //System.out.println(array);  
  
  
 //get project tasks  
 Statement stmt = *myConn*.createStatement();  
 String query = "SELECT Task\_Name,Status\_int,start\_date,end\_date FROM tasks WHERE Project=\""+Project+"\"ORDER BY end\_date ASC";  
 ResultSet rs = stmt.executeQuery(query);  
  
 i = 0;  
 //list = new ArrayList();  
  
 while (rs.next()) {  
  
 String test = rs.getString("Task\_Name");  
 String test2 = rs.getString("Status\_int");  
 String test3 = rs.getString("start\_date");  
 String test4 = rs.getString("end\_date");  
 String test5 = "\_\_\_\_\_\_\_\_\_\_\_";  
 System.*out*.println(test);  
 System.*out*.println(test2);  
 System.*out*.println(test3);  
 System.*out*.println(test4);  
 System.*out*.println(test5);  
 list.add(test);  
 list.add(test2);  
 list.add(test3);  
 list.add(test4);  
 //list.add(test5);  
 }  
 String[] array = (String[]) list.toArray(new String[list.size()]);  
 System.*out*.println(Arrays.*toString*(array));  
 System.*out*.println(array.length);  
  
 return array;  
 }  
  
 public static String[] GET\_TASK\_INFO(String Project,String Task) throws SQLException {  
 Statement stmt = *myConn*.createStatement();  
 String query = "SELECT Task\_name,Project,Assigned\_User,Created\_By,Status\_int,start\_date,end\_date,Description FROM tasks WHERE Project=\""+Project+"\" AND Task\_name=\""+Task+"\"";  
 ResultSet rs = stmt.executeQuery(query);  
  
 rs.next();  
 //byte[] test = rs.getBytes("salt");  
 String Des = rs.getString("Description");  
 String Status = rs.getString("Status\_int");  
 String Assigned\_user = rs.getString("Assigned\_User");  
 String s\_date = rs.getString("start\_date");  
 String e\_date = rs.getString("end\_date");  
  
 System.*out*.println(Des+Status+Assigned\_user+s\_date+e\_date+Task);  
  
 String[] data = {Des, Status, Assigned\_user, s\_date, e\_date, Task};  
  
  
 return data;  
 }  
 public static boolean UPDATE\_TASK\_STATUS(String Project,String Task,String Status) {  
 Statement stmt = null;  
 try {  
 stmt = *myConn*.createStatement();  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 String query = "UPDATE tasks\n" +  
 "SET Status\_int = "+Status+"\n" +  
 "WHERE Task\_name = \""+Task+"\" AND Project = \""+Project+"\";";  
 try {  
 ResultSet rs = stmt.executeQuery(query);  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 return false;  
 }  
 return true;  
  
 }  
  
 public static boolean insert\_minor\_task\_into\_project(String root\_task,String Task\_name,String Project,String Assign\_User,String Created\_By,String start\_D,String end\_D,String Des) {  
 start\_D = start\_D.substring(7);  
 end\_D = end\_D.substring(5);  
  
 if (start\_D.charAt(1) == '/') start\_D = "0" + start\_D;  
 if (start\_D.charAt(4) == '/') start\_D = start\_D.substring(0,3) + "0" + start\_D.substring(3);  
  
 start\_D = "STR\_TO\_DATE(\'"+start\_D+"\',\'%d/%m/%Y\')";  
 end\_D = "STR\_TO\_DATE(\'"+end\_D+"\',\'%d/%m/%Y\')";  
  
  
  
  
 try{  
 Statement stmt = *myConn*.createStatement();  
  
 // eg TO\_DATE('17/12/2015', 'DD/MM/YYYY')  
 String query = "INSERT INTO minortasks (root\_task,Task\_name, Project, Assigned\_User,Created\_By,Status\_int,start\_date,end\_date,Description) VALUES (\""+root\_task+"\",\""+Task\_name+"\",\""+Project+"\",\""+Assign\_User+"\",\""+Created\_By+"\",\""+String.*valueOf*(1)+"\","+start\_D+","+end\_D+",\""+Des+"\");";  
 ResultSet rs = stmt.executeQuery(query);  
 return true;  
 }catch (SQLException e ){  
 System.*out*.println(e);  
 return false;  
 }  
 }  
  
 public static String[] get\_project\_tasks\_minor(String Project, String root\_task) throws SQLException {  
  
 int i = 0;  
 ArrayList list = new ArrayList();  
  
  
  
 //get project tasks  
 Statement stmt = *myConn*.createStatement();  
 String query = "SELECT Task\_Name,Status\_int,start\_date,end\_date FROM minortasks WHERE Project=\""+Project+"\" AND root\_task=\""+root\_task+"\"ORDER BY end\_date ASC";  
 ResultSet rs = stmt.executeQuery(query);  
 //list = new ArrayList();  
  
 while (rs.next()) {  
  
 String test = rs.getString("Task\_Name");  
 String test2 = rs.getString("Status\_int");  
 String test3 = rs.getString("start\_date");  
 String test4 = rs.getString("end\_date");  
 String test5 = "\_\_\_\_\_\_\_\_\_\_\_";  
 System.*out*.println(test);  
 System.*out*.println(test2);  
 System.*out*.println(test3);  
 System.*out*.println(test4);  
 System.*out*.println(test5);  
 list.add(test);  
 list.add(test2);  
 list.add(test3);  
 list.add(test4);  
 //list.add(test5);  
 }  
 String[] array = (String[]) list.toArray(new String[list.size()]);  
 System.*out*.println(Arrays.*toString*(array));  
 System.*out*.println(array.length);  
  
 return array;  
 }  
  
  
}

### Client app

#### Manifest scripts

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.app">  
  
 <uses-permission android:name = "android.permission.ACCESS\_WIFI\_STATE" />  
 <uses-permission android:name = "android.permission.ACCESS\_NETWORK\_STATE" />  
 <uses-permission android:name = "android.permission.INTERNET"/>  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.App">  
 <activity android:name=".temp\_Home"></activity>  
 <activity android:name=".act\_registration\_Main"></activity>  
 <activity android:name=".MainActivity"  
 android:label="Untitled APP">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity android:name=".Project\_activity"/>  
 <activity android:name=".Task\_Page"/>  
  
 </application>  
  
</manifest>

#### Java scripts

package com.example.app;  
  
import androidx.annotation.RequiresApi;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.AsyncTask;  
import android.os.Build;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
  
import java.io.BufferedReader;  
import java.io.DataInputStream;  
import java.io.DataOutputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.ObjectInputStream;  
import java.io.ObjectOutputStream;  
import java.io.PrintWriter;  
import java.lang.ref.WeakReference;  
import java.net.Socket;  
import java.net.ServerSocket;  
import java.security.KeyFactory;  
import java.security.PrivateKey;  
import java.security.PublicKey;  
import java.security.spec.PKCS8EncodedKeySpec;  
import java.text.DateFormat;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Base64;  
import java.util.Collections;  
import java.util.Comparator;  
import java.util.Date;  
import java.util.Locale;  
import java.util.Scanner;  
  
  
public class MainActivity extends AppCompatActivity {  
 //  
 String message = "";  
 private static final int *SERVERPORT* = 12345;  
 private static final String *ip* = "192.168.0.6";//server ip address  
 //private static final String ip = "172.16.11.249";//server ip address gab  
 //  
  
 public static client\_con *con*;  
  
 private int counter = 5;  
  
 private String temp\_UserName = "Admin";  
 private String temp\_Passwowrd = "1234";  
 private boolean login\_isValid = false;  
  
 AES aes;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 *con* = new client\_con();  
 *con*.execute();  
 }  
  
 */\*\*  
 \* class used to interact with server  
 \*/* class client\_con extends AsyncTask<Void,Void,Void>{  
  
  
 private PrintWriter out;  
 private BufferedReader in;  
 private Scanner scanner;  
  
 private EditText eName;  
 private EditText ePassword;  
 private Button elogin;  
 private TextView eAttemptsInfo;  
  
 private Button eRegister\_Main;  
 private Button eBack\_act\_reg;  
  
 private String USER\_id;  
  
 public ObjectInputStream ois;  
 public ArrayList<Task\_Object> Project\_Tasks;  
  
 String Currently\_selected\_project\_view = "no project selected";  
  
  
  
 String state = "empty";  
 //reg new user var  
 String New\_user = "empty";  
 String New\_Pass = "empty";  
 String Project\_Name;  
 String Project\_Description;  
  
  
 private WeakReference<MainActivity> activityWeakReference;  
  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 @Override  
 protected Void doInBackground(Void... voids) {  
 eName = findViewById(R.id.*et\_UserName*);  
 ePassword = findViewById(R.id.*et\_Password*);  
 elogin = findViewById(R.id.*btn\_login*);  
 eAttemptsInfo = findViewById(R.id.*tv\_login\_response*);  
 eRegister\_Main = findViewById(R.id.*btn\_registration*);  
 eBack\_act\_reg = findViewById(R.id.*btn\_back\_act\_reg*);  
  
  
  
 Object server\_encryptKey;  
  
  
  
 elogin.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String inputName = eName.getText().toString();  
 String inputPass = ePassword.getText().toString();  
 if(inputName.isEmpty() || inputPass.isEmpty()) {  
 state = "empty";  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"login fail", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }else{  
 state = "LOGIN\_request";  
 }  
 }  
  
 });  
  
 eRegister\_Main.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent reg\_page = new Intent(MainActivity.this,act\_registration\_Main.class);  
 startActivity(reg\_page);  
  
 }  
  
 });  
  
  
  
 try (Socket socket = new Socket(*ip*, 12345)) {  
 ObjectInputStream inputStream;  
 DataOutputStream dOut = new DataOutputStream(socket.getOutputStream());  
 out = new PrintWriter(socket.getOutputStream(), true);  
 in = new BufferedReader(new InputStreamReader(socket.getInputStream()));  
  
 DataInputStream dIn = new DataInputStream(socket.getInputStream());  
 int length = dIn.readInt();// read length of incoming message  
 byte[] key\_en = new byte[length];  
 if(length>0) {  
 dIn.readFully(key\_en, 0, key\_en.length); // read the message  
 }  
  
 KeyFactory kf = KeyFactory.*getInstance*("RSA"); // or "EC" or whatever  
 PrivateKey ret = kf.generatePrivate(new PKCS8EncodedKeySpec(key\_en));  
  
 byte[] test;  
 test = RSA.*encryptMessage\_cipher*("this is the message i want to see",ret);  
 dOut.writeInt(test.length); // write length of the message  
 dOut.write(test);  
  
 aes = new AES();  
 aes.*GenerateKeys*();  
 // get base64 encoded version of the key  
 String encodedKey = Base64.*getEncoder*().encodeToString(aes.*secretKey\_encoded*);  
  
 //rsa encryption  
 byte[] aes\_key\_en = RSA.*encryptMessage\_cipher*(encodedKey,ret);  
  
 dOut.writeInt(aes\_key\_en.length); // write length of the message  
 dOut.write(aes\_key\_en);  
  
 Log.*e*("YOUR\_APP\_LOG\_TAG", aes.*decrypt*(dIn.readUTF()));  
  
 while(!"exit".equalsIgnoreCase(state)){  
 if(state == null || state.equals("empty")){  
 state = "empty";  
 }  
  
 if(state.equals("EXIT")){  
 break;  
 }  
 if(state.equals("LOGIN\_request")){  
 String inputName = eName.getText().toString();  
 String inputPass = ePassword.getText().toString();  
 LOGIN\_request(inputName,inputPass);  
 //state = "empty";  
 state = "GET\_USER\_PROJECTS";  
 }  
 if(state.equals("NEW\_USER\_request")){  
 NEW\_USER\_request(New\_user,New\_Pass);  
 state = "empty";  
 }  
 if(state.equals("CREATE\_NEW\_PROJECT")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Create new project sig "+ USER\_id);  
 CREATE\_NEW\_PROJECT(Project\_Name, Project\_Description);  
 state = "empty";  
 }  
 if(state.equals("GET\_USER\_PROJECTS")){  
 //HomeFragment.Projects =  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "get user projects "+ USER\_id);  
 GET\_USER\_PROJECTS();  
 state = "empty";  
 }  
 if(state.equals("GET\_ALL\_USERS")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "get all users"+ USER\_id);  
 GET\_ALL\_USERS();  
 state = "empty";  
 }  
 if(state.equals("ASSIGN\_USER\_TO\_PROJECT")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "ASSIGN\_USER\_TO\_PROJECT"+ USER\_id);  
 ASSIGN\_USER\_TO\_PROJECT();  
 state = "empty";  
 }  
 if(state.equals("GET\_ALL\_USERS\_IN\_PROJECT")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "GET\_ALL\_USERS\_IN\_PROJECT"+ Currently\_selected\_project\_view);  
 GET\_ALL\_USERS\_IN\_PROJECT();  
 state = "empty";  
 }  
 if(state.equals("GET\_ALL\_USERS\_IN\_PROJECT\_TASKPAGE")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "GET\_ALL\_USERS\_IN\_PROJECT\_TASKPAGE"+ Currently\_selected\_project\_view);  
 GET\_ALL\_USERS\_IN\_PROJECT\_TASKPAGE();  
 state = "empty";  
 }  
 if(state.equals("INSERT\_NEW\_TASK")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "INSET\_NEW\_TASK"+ Currently\_selected\_project\_view);  
 INSERT\_NEW\_TASK();  
 state = "empty";  
 }  
 if(state.equals("GET\_PROJECT\_TASK\_INFORMATION")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "GET\_PROJECT\_TASK\_INFORMATION"+ Currently\_selected\_project\_view);  
 GET\_PROJECT\_TASK\_INFORMATION();  
 state = "empty";  
 }  
 if(state.equals("GET\_TASK\_INFO")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "GET\_TASK\_INFO");  
 GET\_TASK\_INFO();  
 state = "empty";  
 }  
 if(state.equals("UPDATE\_TASK\_STATUS")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "UPDATE\_TASK\_STATUS");  
 UPDATE\_TASK\_STATUS();  
 state = "empty";  
 }  
 if(state.equals("GET\_ALL\_USERS\_2")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "get all users 2"+ USER\_id);  
 GET\_ALL\_USERS2();  
 state = "empty";  
 }  
 if(state.equals("INSERT\_NEW\_MINOR\_TASK")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "INSERT\_NEW\_MINOR\_TASK"+ USER\_id);  
 INSERT\_NEW\_MINOR\_TASK();  
 state = "empty";  
 }  
 if(state.equals("GET\_PROJECT\_MINOR\_TASK\_INFORMATION")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "GET\_PROJECT\_MINOR\_TASK\_INFORMATION"+ Currently\_selected\_project\_view);  
 GET\_PROJECT\_MINOR\_TASK\_INFORMATION();  
 state = "empty";  
 }  
  
  
  
 }  
  
  
 scanner.close();  
  
 } catch (IOException | ClassNotFoundException e) {  
 e.printStackTrace();  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "I got an error", e);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "connection terminated");  
  
 return null;  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void LOGIN\_request(String inputName, String inputPass) throws Exception {  
  
  
  
 out.println(aes.*encrypt*("LOGIN\_request"));  
 out.println(aes.*encrypt*(inputName));  
 out.println(aes.*encrypt*(inputPass));  
 if(aes.*decrypt*(in.readLine()).equals("T")){  
 Intent intent = new Intent(MainActivity.this,temp\_Home.class);  
 startActivity(intent);  
 USER\_id = inputName;  
 }else{  
  
  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"login fail", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void NEW\_USER\_request(String User,String Pass) throws Exception {  
  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "signal sent new user request");  
 out.println(aes.*encrypt*("NEW\_USER\_request"));  
 out.println(aes.*encrypt*(User));  
 out.println(aes.*encrypt*(Pass));  
  
 if(aes.*decrypt*(in.readLine()).equals("T")){  
 Intent reg\_page = new Intent(MainActivity.this,MainActivity.class);  
 startActivity(reg\_page);  
 }else{  
  
  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"login fail", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
  
  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void CREATE\_NEW\_PROJECT(String P\_Name, String Description) throws Exception {  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "signal sent new user request");  
 out.println(aes.*encrypt*("CREATE\_NEW\_PROJECT"));  
 out.println(aes.*encrypt*(P\_Name));  
 out.println(aes.*encrypt*(Description));  
 out.println(aes.*encrypt*(USER\_id));  
 if(aes.*decrypt*(in.readLine()).equals("T")){  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"Project has been created", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }else{  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"Project Name is taken please selected new name", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void GET\_USER\_PROJECTS() throws Exception {  
 out.println(aes.*encrypt*("GET\_USER\_PROJECTS"));  
 out.println(aes.*encrypt*(USER\_id));  
 boolean check1 = true;  
 String check2;  
 ArrayList list = new ArrayList();  
 while(check1){  
 check2 = aes.*decrypt*(in.readLine());  
 //System.out.println(test);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", check2);  
  
 if(check2.equals("end\_of\_String\_array\_n10193197")){  
 break;  
 }  
 list.add(check2);  
 }  
 String[] temp\_store = (String[]) list.toArray(new String[list.size()]);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of get user projects");  
 HomeFragment.*Projects* = new String[temp\_store.length];  
 HomeFragment.*Projects* = temp\_store;  
 HomeFragment.*check1* = true;  
 //HomeFragment.setDone();  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void GET\_ALL\_USERS() throws IOException {  
 out.println(aes.*encrypt*("GET\_ALL\_USERS"));  
 out.println(aes.*encrypt*(USER\_id));  
  
 boolean check1 = true;  
 String check2;  
 ArrayList list = new ArrayList();  
 ArrayList<User\_object\_add\_user> User\_List\_obj\_temp = new ArrayList<User\_object\_add\_user>();  
 while(check1){  
 check2 = aes.*decrypt*(in.readLine());  
 if(check2.equals("end\_of\_String\_array\_n10193197")){  
 break;  
 }  
 //System.out.println(test);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", check2);  
 User\_object\_add\_user test\_u = new User\_object\_add\_user(check2);  
 if(check2.equals(USER\_id)){  
  
 }else{  
 P\_addUser.*User\_List\_obj*.add(test\_u);  
 }  
 list.add(check2);  
 }  
 String[] temp\_store = (String[]) list.toArray(new String[list.size()]);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of get all users");  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void ASSIGN\_USER\_TO\_PROJECT() throws IOException {  
 out.println(aes.*encrypt*("ASSIGN\_USER\_TO\_PROJECT"));  
 out.println(aes.*encrypt*(USER\_id));  
 out.println(aes.*encrypt*(P\_addUser.*new\_user\_to\_project*));  
 out.println(aes.*encrypt*(Currently\_selected\_project\_view));  
 String response = aes.*decrypt*(in.readLine());  
 if(response.equals("T")){  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"Success", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 } else if (response.equals("error check permission")) {  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"user is not permitted to take this action", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }else if (response.equals("error inserting new user")){  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"unable to add user", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 } else if (response.equals("need permission")){  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"User is not permitted to take this action", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }else{  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"Unknown Error", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void GET\_ALL\_USERS\_IN\_PROJECT() throws IOException, InterruptedException {  
  
 out.println(aes.*encrypt*("GET\_ALL\_USERS\_IN\_PROJECT"));  
 out.println(aes.*encrypt*(Currently\_selected\_project\_view));  
  
 boolean check1 = true;  
 String check2;  
 ArrayList list = new ArrayList();  
 ArrayList<User\_object\_add\_user> User\_List\_obj\_temp = new ArrayList<User\_object\_add\_user>();  
 while(check1){  
 check2 = aes.*decrypt*(in.readLine());  
 if(check2.equals("end\_of\_String\_array\_n10193197")){  
 break;  
 }  
 //System.out.println(test);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", check2);  
 User\_object\_add\_user test\_u = new User\_object\_add\_user(check2);  
 P\_addTask.*User\_List\_obj*.add(test\_u);  
 list.add(check2);  
 }  
 String[] temp\_store = (String[]) list.toArray(new String[list.size()]);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of get all users");  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void GET\_ALL\_USERS\_IN\_PROJECT\_TASKPAGE() throws IOException, InterruptedException {  
  
 out.println(aes.*encrypt*("GET\_ALL\_USERS\_IN\_PROJECT"));  
 out.println(aes.*encrypt*(Currently\_selected\_project\_view));  
  
 boolean check1 = true;  
 String check2;  
 ArrayList list = new ArrayList();  
 ArrayList<User\_object\_add\_user> User\_List\_obj\_temp = new ArrayList<User\_object\_add\_user>();  
 while(check1){  
 check2 = aes.*decrypt*(in.readLine());  
 if(check2.equals("end\_of\_String\_array\_n10193197")){  
 break;  
 }  
 //System.out.println(test);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", check2);  
 User\_object\_add\_user test\_u = new User\_object\_add\_user(check2);  
 Task\_Page.*User\_List\_obj*.add(test\_u);  
 list.add(check2);  
 }  
 String[] temp\_store = (String[]) list.toArray(new String[list.size()]);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of get all users");  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void INSERT\_NEW\_TASK() throws IOException {  
 out.println(aes.*encrypt*("INSERT\_NEW\_TASK"));  
 out.println(aes.*encrypt*(P\_addTask.*task\_name*.getText().toString()));  
 String TASK\_DES = P\_addTask.*task\_des*.getText().toString();  
 if(TASK\_DES == null){  
 TASK\_DES = "No Description Provided";  
 }else if(TASK\_DES.isEmpty()){  
 TASK\_DES = "No Description Provided";  
 }  
 out.println(aes.*encrypt*(TASK\_DES));  
 out.println(aes.*encrypt*(P\_addTask.*task\_start*.getText().toString()));  
 out.println(aes.*encrypt*(P\_addTask.*task\_end*.getText().toString()));  
 String Assigned\_User = P\_addTask.*select\_user*;  
 if(Assigned\_User == null){  
 Assigned\_User = "No Assigned User";  
 }else if(Assigned\_User.isEmpty()){  
 Assigned\_User = "No Assigned User";  
 }  
 out.println(aes.*encrypt*(Assigned\_User));  
 out.println(aes.*encrypt*(Currently\_selected\_project\_view));  
 out.println(aes.*encrypt*(USER\_id));  
  
  
  
  
  
 String response = aes.*decrypt*(in.readLine());  
 if(response.equals("T")){  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"Task Added to Project", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }else{  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"Unable to add Task", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void GET\_PROJECT\_TASK\_INFORMATION() throws IOException, ParseException {  
  
 Project\_Tasks = new ArrayList<Task\_Object>();  
 out.println(aes.*encrypt*("GET\_PROJECT\_TASK\_INFORMATION"));  
 out.println(aes.*encrypt*(Currently\_selected\_project\_view));  
  
 String Project\_Creation = new String();  
 Project\_Creation = in.readLine();  
  
 boolean check1 = true;  
 while(check1){  
 String Task\_Name = aes.*decrypt*(in.readLine());  
 if(Task\_Name.equals("end\_of\_String\_array\_n10193197")){  
 break;  
 }  
 int Status = Integer.*parseInt*(aes.*decrypt*(in.readLine()));  
 String temp\_start\_date = aes.*decrypt*(in.readLine());  
 DateFormat format = new SimpleDateFormat("yyyy-MM-dd", Locale.*ENGLISH*);  
 Date start\_date = format.parse(temp\_start\_date);  
  
 String temp\_end\_date = aes.*decrypt*(in.readLine());  
 DateFormat format2 = new SimpleDateFormat("yyyy-MM-dd", Locale.*ENGLISH*);  
 Date end\_date = format2.parse(temp\_end\_date);  
 Project\_Tasks.add(new Task\_Object(Task\_Name,Status,start\_date,end\_date));  
 //System.out.println(test);  
 }  
 //Log.e("YOUR\_APP\_LOG\_TAG", String.valueOf(Project\_Tasks.size()));  
 //Log.e("YOUR\_APP\_LOG\_TAG", Project\_Tasks.get(0).Task\_Name);  
 //Log.e("YOUR\_APP\_LOG\_TAG", String.valueOf(Project\_Tasks.get(0).Status));  
 //Log.e("YOUR\_APP\_LOG\_TAG", String.valueOf(Project\_Tasks.get(0).Start));  
 //Log.e("YOUR\_APP\_LOG\_TAG", String.valueOf(Project\_Tasks.get(0).End));  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of GET\_PROJECT\_TASK\_INFORMATION");  
 //Project\_Tasks.sort();  
 P\_overview.*check1* = true;  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void GET\_TASK\_INFO() throws IOException {  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "GET\_TASK\_INFO "+Currently\_selected\_project\_view+P\_overview.*task\_name*.getText().toString());  
 out.println(aes.*encrypt*("GET\_TASK\_INFO"));  
 out.println(aes.*encrypt*(Currently\_selected\_project\_view));  
 out.println(aes.*encrypt*(P\_overview.*task\_name*.getText().toString()));  
  
 P\_overview.*sel\_task* = P\_overview.*task\_name*.getText().toString();  
  
 Task\_Page.*Des* = aes.*decrypt*(in.readLine());  
 Task\_Page.*Status* = aes.*decrypt*(in.readLine());  
 Task\_Page.*Assigned\_user* = aes.*decrypt*(in.readLine());  
 Task\_Page.*s\_date* = aes.*decrypt*(in.readLine());  
 Task\_Page.*e\_date* = aes.*decrypt*(in.readLine());  
 Task\_Page.*e\_taskname* = aes.*decrypt*(in.readLine());  
  
 Task\_Page.*check1* = true;  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "GET\_TASK\_INFO "+Task\_Page.*e\_taskname*);  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void UPDATE\_TASK\_STATUS() throws IOException {  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "UPDATE\_TASK\_STATUS");  
 out.println(aes.*encrypt*("UPDATE\_TASK\_STATUS"));  
 out.println(aes.*encrypt*(Currently\_selected\_project\_view));  
 out.println(aes.*encrypt*(P\_overview.*task\_name*.getText().toString()));  
 out.println(aes.*encrypt*(Task\_Page.*selected\_radio\_option*));  
  
 if(aes.*decrypt*(in.readLine()).equals("T")){  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"Status Update Successful", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }else{  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"Status Update Fail", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void GET\_ALL\_USERS2() throws IOException {  
 out.println(aes.*encrypt*("GET\_ALL\_USERS"));  
 out.println(aes.*encrypt*(USER\_id));  
  
 boolean check1 = true;  
 String check2;  
 ArrayList list = new ArrayList();  
 ArrayList<User\_object\_add\_user> User\_List\_obj\_temp = new ArrayList<User\_object\_add\_user>();  
 while(check1){  
 check2 = aes.*decrypt*(in.readLine());  
 if(check2.equals("end\_of\_String\_array\_n10193197")){  
 break;  
 }  
 //System.out.println(test);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", check2);  
 User\_object\_add\_user test\_u = new User\_object\_add\_user(check2);  
 if(check2.equals(USER\_id)){  
  
 }else{  
 MessageesFragment.*User\_List\_obj*.add(test\_u);  
 }  
 list.add(check2);  
 }  
 String[] temp\_store = (String[]) list.toArray(new String[list.size()]);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of get all users");  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void INSERT\_NEW\_MINOR\_TASK() throws IOException {  
 out.println(aes.*encrypt*("INSERT\_NEW\_MINOR\_TASK"));  
 out.println(aes.*encrypt*(Task\_Page.*e\_taskname*));  
 out.println(aes.*encrypt*(Task\_Page.*task\_name*.getText().toString()));  
  
 String TASK\_DES = Task\_Page.*task\_des*.getText().toString();  
 if(TASK\_DES == null){  
 TASK\_DES = "No Description Provided";  
 }else if(TASK\_DES.isEmpty()){  
 TASK\_DES = "No Description Provided";  
 }  
 out.println(aes.*encrypt*(TASK\_DES));  
 out.println(aes.*encrypt*(Task\_Page.*task\_start*.getText().toString()));  
 out.println(aes.*encrypt*(Task\_Page.*task\_end*.getText().toString()));  
 String Assigned\_User = Task\_Page.*select\_user*;  
 if(Assigned\_User == null){  
 Assigned\_User = "No Assigned User";  
 }else if(Assigned\_User.isEmpty()){  
 Assigned\_User = "No Assigned User";  
 }  
 out.println(aes.*encrypt*(Assigned\_User));  
 out.println(aes.*encrypt*(Currently\_selected\_project\_view));  
 out.println(aes.*encrypt*(USER\_id));  
  
  
  
  
  
 String response = aes.*decrypt*(in.readLine());  
 if(response.equals("T")){  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"minor Task Added to Project", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }else{  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(MainActivity.this,"Unable to add minor Task", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void GET\_PROJECT\_MINOR\_TASK\_INFORMATION() throws IOException, ParseException {  
  
 Project\_Tasks = new ArrayList<Task\_Object>();  
 out.println(aes.*encrypt*("GET\_PROJECT\_MINOR\_TASK\_INFORMATION"));  
 out.println(aes.*encrypt*(Currently\_selected\_project\_view));  
 out.println(aes.*encrypt*(Task\_Page.*e\_taskname*));  
  
 boolean check1 = true;  
 int count = 0;  
 while(check1){  
 String Task\_Name = aes.*decrypt*(in.readLine());  
 if(Task\_Name.equals("end\_of\_String\_array\_n10193197")){  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of GET\_PROJECT\_MINOR\_TASK\_INFORMATION");  
 break;  
 }  
 int Status = Integer.*parseInt*(aes.*decrypt*(in.readLine()));  
 String temp\_start\_date = aes.*decrypt*(in.readLine());  
 DateFormat format = new SimpleDateFormat("yyyy-MM-dd", Locale.*ENGLISH*);  
 Date start\_date = format.parse(temp\_start\_date);  
  
 String temp\_end\_date = aes.*decrypt*(in.readLine());  
 DateFormat format2 = new SimpleDateFormat("yyyy-MM-dd", Locale.*ENGLISH*);  
 Date end\_date = format2.parse(temp\_end\_date);  
 Project\_Tasks.add(new Task\_Object(Task\_Name,Status,start\_date,end\_date));  
 count++;  
 Log.*e*("YOUR\_APP\_LOG\_TAG", " GET\_PROJECT\_MINOR\_TASK\_INFORMATION here \n");  
 }  
 //Project\_Tasks.sort();  
 P\_overview.*check1* = true;  
 }  
 }  
  
}

package com.example.app.Common;  
  
*/\*\*  
 \* class used to store common variables  
 \*/*public class Common {  
 public static int *HEADER\_COUNT* = 45;  
 public static int *COLUMN\_COUNT* = 60 ;//always larger header count  
 public static int *CURRENT\_WEEK* = 10;  
}

package com.example.app.GanttAdapter;  
import android.content.Context;  
  
import miguelbcr.ui.tableFixHeadesWrapper.TableFixHeaderAdapter;  
  
import com.example.app.Common.Common;  
import com.example.app.GanttItem;  
import com.example.app.P\_overview;  
import com.example.app.R;  
import com.example.app.ViewGroup.BarCellViewGroup;  
import com.example.app.ViewGroup.CellViewGroup;  
import com.example.app.ViewGroup.IOnHeaderClickListener;  
  
import java.util.ArrayList;  
import java.util.List;  
  
*/\*\*  
 \* adapter class for gantt chart  
 \*/*public class GanttTableFixHeaderAdapter extends TableFixHeaderAdapter<  
 String, CellViewGroup,// first header  
 String, CellViewGroup,//header items  
 List<String>,//data source  
 CellViewGroup,//first body  
 BarCellViewGroup,//body items  
 CellViewGroup// Section  
 > {  
  
 private Context context;  
 private List<GanttItem> ganttItems;  
 private IOnHeaderClickListener listener;  
  
 public void setListener(IOnHeaderClickListener listener){  
 this.listener = listener;  
 }  
  
 public GanttTableFixHeaderAdapter(Context context, List<GanttItem> ganttItems) {  
 super(context);  
 this.context = context;  
 this.ganttItems = ganttItems;  
 }  
  
 @Override  
 protected CellViewGroup inflateFirstHeader() {  
 return new CellViewGroup(context);  
 }  
  
 @Override  
 protected CellViewGroup inflateHeader() {  
 CellViewGroup defaultCellViewGroup = new CellViewGroup(context);  
 defaultCellViewGroup.setListener(listener);//set click  
 //return new CellViewGroup(context);  
 return defaultCellViewGroup;  
 }  
  
 @Override  
 protected CellViewGroup inflateFirstBody() {  
 CellViewGroup defaultCellViewGroup = new CellViewGroup(context,ganttItems);  
 defaultCellViewGroup.setListener(listener);//set click  
 //return new CellViewGroup(context);  
 return defaultCellViewGroup;  
 }  
  
 @Override  
 protected BarCellViewGroup inflateBody() {  
 return new BarCellViewGroup(context, ganttItems);  
 }  
  
 @Override  
 protected CellViewGroup inflateSection() {  
 return new CellViewGroup(context);  
 }  
  
 @Override  
 protected List<Integer> getHeaderWidths() {  
 List<Integer> headerWidths = new ArrayList<>();  
 //first body will have 150dp  
 headerWidths.add((int) context.getResources().getDimension(R.dimen.*\_150dp*));  
 for(int i = 0; i< Common.*COLUMN\_COUNT*; i++){  
 headerWidths.add((int) context.getResources().getDimension(R.dimen.*\_40dp*));//each item will have 40dp  
  
 }  
 return headerWidths;  
 }  
  
 @Override  
 protected int getHeaderHeight() {  
 return (int)context.getResources().getDimension(R.dimen.*\_40dp*);  
 }  
  
 @Override  
 protected int getSectionHeight() {  
 return (int)context.getResources().getDimension(R.dimen.*\_40dp*);  
 }  
  
 @Override  
 protected int getBodyHeight() {  
 return (int)context.getResources().getDimension(R.dimen.*\_40dp*);  
 }  
  
 @Override  
 protected boolean isSection(List<List<String>> list, int i) {  
 return false;  
 }  
  
}

package com.example.app.ViewGroup;  
  
import android.annotation.SuppressLint;  
import android.content.Context;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.widget.FrameLayout;  
import android.widget.RelativeLayout;  
  
import androidx.annotation.NonNull;  
import androidx.core.content.ContextCompat;  
  
import com.example.app.GanttItem;  
import com.example.app.R;  
  
import java.util.List;  
  
import miguelbcr.ui.tableFixHeadesWrapper.TableFixHeaderAdapter;  
  
*/\*\*  
 \* class for gantt chart cells  
 \*/*public class BarCellViewGroup extends FrameLayout implements TableFixHeaderAdapter.BodyBinder<List<String>> {  
 RelativeLayout card\_item;  
 Context context;  
 List<GanttItem> ganttItemList;  
  
 public BarCellViewGroup(@NonNull Context context) {  
 super(context);  
 this.context = context;  
 LayoutInflater.*from*(context).inflate(R.layout.*card\_view\_item*,this,true);  
 card\_item = (RelativeLayout) findViewById(R.id.*card\_item*);  
 }  
  
 public BarCellViewGroup(@NonNull Context context, List<GanttItem> ganttItemList) {  
 super(context);  
 this.context = context;  
 this.ganttItemList = ganttItemList;  
 LayoutInflater.*from*(context).inflate(R.layout.*card\_view\_item*,this,true);  
 card\_item = (RelativeLayout) findViewById(R.id.*card\_item*);  
 }  
 //this function to add more colours  
 @Override  
 public void bindBody(List<String> strings, int row, int col) {  
 //will be base on string value to set data  
 //Log.e("YOUR\_APP\_LOG\_TAG", "bindBody CarCell"+ strings);  
 //Log.e("YOUR\_APP\_LOG\_TAG", "bindBody CarCell"+ row);  
 //Log.e("YOUR\_APP\_LOG\_TAG", "bindBody CarCell"+ col);  
 //Log.e("YOUR\_APP\_LOG\_TAG", "bindBody CarCell"+ strings.size());  
  
  
 if (col >= strings.size()) {  
 //card\_item.setBackgroundColor(ContextCompat.getColor(context,android.R.color.holo\_orange\_dark));  
 } else if(strings.get(col).equals("done & current\_week")){  
 card\_item.setBackgroundResource(R.drawable.*border\_current\_with\_task*);  
 }else if(strings.get(col).equals("error & current\_week")){  
 card\_item.setBackgroundResource(R.drawable.*border\_current\_without\_task*);  
 }else if(strings.get(col).equals("error")) {  
 card\_item.setBackgroundColor(ContextCompat.*getColor*(context, android.R.color.*holo\_red\_dark*));  
 }else if (strings.get(col).equals("done")) {  
 card\_item.setBackgroundColor(ContextCompat.*getColor*(context, android.R.color.*holo\_blue\_dark*));  
 }else if(strings.get(col).equals("current\_week")){  
 card\_item.setBackgroundResource(R.drawable.*border\_current*);  
 }else if(strings.get(col).equals("overtime")){  
 card\_item.setBackgroundColor(ContextCompat.*getColor*(context, android.R.color.*holo\_orange\_dark*));  
 }else if(strings.get(col).equals("complete & current\_week")){  
 card\_item.setBackgroundResource(R.drawable.*border\_current\_week\_complete*);  
 }else if(strings.get(col).equals("complete")){  
 card\_item.setBackgroundColor(ContextCompat.*getColor*(context, android.R.color.*holo\_green\_light*));  
 }else{  
 card\_item.setBackgroundResource(R.drawable.*border*);  
 }  
  
  
 }  
}

package com.example.app.ViewGroup;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.widget.FrameLayout;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
  
import com.example.app.GanttItem;  
import com.example.app.R;  
import com.inqbarna.tablefixheaders.TableFixHeaders;  
  
import java.util.List;  
  
import miguelbcr.ui.tableFixHeadesWrapper.TableFixHeaderAdapter;  
  
*/\*\*  
 \* class for gantt task cells  
 \*/*public class CellViewGroup extends FrameLayout implements  
 TableFixHeaderAdapter.FirstBodyBinder<List<String>>,  
 TableFixHeaderAdapter.FirstHeaderBinder<String>,  
 TableFixHeaderAdapter.HeaderBinder<String>,  
 TableFixHeaderAdapter.BodyBinder<List<String>>,  
 TableFixHeaderAdapter.SectionBinder<List<String>>  
{  
 //global  
 TextView txt\_content;  
 List<GanttItem> ganttItemList;  
 IOnHeaderClickListener listener;  
  
 public void setListener(IOnHeaderClickListener listener) {  
 this.listener = listener;  
 }  
  
 public CellViewGroup(@NonNull Context context) {  
 super(context);  
 LayoutInflater.*from*(context).inflate(R.layout.*gantt\_view\_item*,this,true);  
 txt\_content = (TextView)findViewById(R.id.*txt\_content*);  
 }  
  
 public CellViewGroup(@NonNull Context context, List<GanttItem> ganttItemList) {  
 super(context);  
 this.ganttItemList = ganttItemList;  
 LayoutInflater.*from*(context).inflate(R.layout.*gantt\_view\_item*,this,true);  
 txt\_content = (TextView)findViewById(R.id.*txt\_content*);  
 }  
  
 @Override  
 public void bindBody(List<String> strings, int i, int i1) {  
 txt\_content.setText(strings.get(i1));  
  
 }  
  
 @Override  
 public void bindFirstHeader(String s) {  
 txt\_content.setText(s);  
 }  
  
  
 @Override  
 public void bindSection(List<String> strings, int i, int i1) {  
 txt\_content.setText(i1 == 0 ? "Section: "+(i+1) : "");  
  
 }  
  
 @Override  
 public void bindHeader(String s, int i) {  
 txt\_content.setText(s);  
 }  
  
  
 @Override  
 public void bindFirstBody(List<String> strings, int i) {  
 txt\_content.setText(ganttItemList.get(i).getTaskName());  
 //Implement Click here  
 txt\_content.setOnClickListener(new TaskClickListener(i,listener));  
  
 }  
}

package com.example.app.ViewGroup;  
  
import android.view.View;  
  
public interface IOnHeaderClickListener {  
 void onHeaderItemClick(View view, int row);  
}

package com.example.app.ViewGroup;  
  
import android.view.View;  
  
*/\*\*  
 \* modified on click listener for task title cells  
 \*/*public class TaskClickListener implements View.OnClickListener {  
 private int row;  
 private IOnHeaderClickListener listener;  
  
 public TaskClickListener(int row, IOnHeaderClickListener listener) {  
 this.row = row;  
 this.listener = listener;  
 }  
  
 @Override  
 public void onClick(View v) {  
 listener.onHeaderItemClick(v,row);  
  
  
 }  
}

package com.example.app;  
  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.navigation.NavController;  
import androidx.navigation.Navigation;  
import androidx.navigation.ui.NavigationUI;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
  
*/\*\*  
 \* inflate home activity after login  
 \*/*public class act\_Home {  
  
 public class balance extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_home*);  
  
 BottomNavigationView bottomNavigationView = findViewById(R.id.*bottomNavigationView2*);  
 NavController navController = Navigation.*findNavController*(this, R.id.*homeFragment*);  
 NavigationUI.*setupWithNavController*(bottomNavigationView, navController);  
  
  
 }  
 }  
}

package com.example.app;  
  
import android.app.Activity;  
import android.content.Intent;  
import android.os.Build;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import androidx.annotation.RequiresApi;  
  
*/\*\*  
 \* registration page java file  
 \*/*public class act\_registration\_Main extends Activity {  
  
 private Button eBack\_act\_reg;  
 private Button eReg\_new\_user;  
  
 private EditText eName\_reg;  
 private EditText ePass\_reg;  
 private EditText ePass2\_reg;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_registration*);  
  
 eBack\_act\_reg = findViewById(R.id.*btn\_back\_act\_reg*);  
 eReg\_new\_user = findViewById(R.id.*btn\_register*);  
  
 eName\_reg = findViewById(R.id.*et\_newUser*);  
 ePass\_reg = findViewById(R.id.*et\_newPass*);  
 ePass2\_reg = findViewById(R.id.*et\_newPass2*);  
  
 eBack\_act\_reg.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent reg\_page = new Intent(act\_registration\_Main.this,MainActivity.class);  
 startActivity(reg\_page);  
 }  
 });  
 eReg\_new\_user.setOnClickListener(new View.OnClickListener() {  
  
 @Override  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public void onClick(View v) {  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "button click");  
 String inputName = eName\_reg.getText().toString();  
 String inputPass = ePass\_reg.getText().toString();  
 String inputPass2 = ePass2\_reg.getText().toString();  
  
 try {  
 if(inputPass.equals(inputPass2)){  
  
 MainActivity.*con*.New\_user = inputName;  
 MainActivity.*con*.New\_Pass = inputPass;  
 //send new user request to server  
 MainActivity.*con*.state = "NEW\_USER\_request";  
  
 }else{  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 Toast.*makeText*(act\_registration\_Main.this,"reg fail", Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 });  
  
  
 }  
  
  
  
}

package com.example.app;  
  
import android.os.Build;  
  
import androidx.annotation.RequiresApi;  
  
import java.io.UnsupportedEncodingException;  
import java.nio.charset.StandardCharsets;  
import java.security.spec.KeySpec;  
import java.util.Base64;  
  
import javax.crypto.Cipher;  
import javax.crypto.KeyGenerator;  
import javax.crypto.SecretKey;  
import javax.crypto.SecretKeyFactory;  
import javax.crypto.spec.IvParameterSpec;  
import javax.crypto.spec.PBEKeySpec;  
import javax.crypto.spec.SecretKeySpec;  
  
*/\*\*  
 \* AES-256 encryption  
 \*/*public class AES {  
 public static SecretKey *secretKey*;  
 public static byte[] *secretKey\_encoded*;  
  
 public static void set\_server\_keys(String aes\_key) throws UnsupportedEncodingException {  
 *secretKey\_encoded* = aes\_key.getBytes();  
 *secretKey* = new SecretKeySpec(*secretKey\_encoded*, 0, *secretKey\_encoded*.length, "AES/ECB/PKCS5Padding");  
 }  
  
 public static void set\_server\_keys\_2(Object aes\_key) throws UnsupportedEncodingException {  
 *secretKey* = (SecretKey) aes\_key;  
 }  
  
 */\*\*  
 \* generate random keys  
 \** ***@throws*** *Exception  
 \*/* public static void GenerateKeys() throws Exception {  
 KeyGenerator keyGen = KeyGenerator.*getInstance*("AES");  
 keyGen.init(256); // for example  
 *secretKey* = keyGen.generateKey();  
 *secretKey\_encoded* = *secretKey*.getEncoded();  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public static String encrypt(String strToEncrypt)  
 {  
 try  
 {  
 Cipher cipher = Cipher.*getInstance*("AES/ECB/PKCS5Padding");  
 cipher.init(Cipher.*ENCRYPT\_MODE*, *secretKey*);  
 return Base64.*getEncoder*().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-8")));  
 }  
 catch (Exception e)  
 {  
 System.*out*.println("Error while encrypting: " + e.toString());  
 }  
 return null;  
 }  
  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public static String decrypt(String strToDecrypt)  
 {  
 try  
 {  
 Cipher cipher = Cipher.*getInstance*("AES/ECB/PKCS5PADDING");  
 cipher.init(Cipher.*DECRYPT\_MODE*, *secretKey*);  
 return new String(cipher.doFinal(Base64.*getDecoder*().decode(strToDecrypt)));  
 }  
 catch (Exception e)  
 {  
 System.*out*.println("Error while decrypting: " + e.toString());  
 }  
 return null;  
 }  
  
  
  
}

package com.example.app;  
  
import android.os.Build;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.annotation.RequiresApi;  
import androidx.fragment.app.Fragment;  
  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \* Use the {****@link*** *CreateTeamFragment#newInstance} factory method to  
 \* create an instance of this fragment.  
 \*  
 \*  
 \* inflate fragment that contains create new team gui  
 \*/*public class CreateTeamFragment extends Fragment {  
  
  
  
 // *TODO: Rename parameter arguments, choose names that match* // the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER  
 private static final String *ARG\_PARAM1* = "param1";  
 private static final String *ARG\_PARAM2* = "param2";  
 private Button e\_create\_project;  
  
  
 // *TODO: Rename and change types of parameters* private String mParam1;  
 private String mParam2;  
  
 public CreateTeamFragment() {  
 // Required empty public constructor  
  
 }  
  
 */\*\*  
 \* Use this factory method to create a new instance of  
 \* this fragment using the provided parameters.  
 \*  
 \** ***@param*** *param1 Parameter 1.  
 \** ***@param*** *param2 Parameter 2.  
 \** ***@return*** *A new instance of fragment CreateTeamFragment.  
 \*/* // *TODO: Rename and change types and number of parameters* public static CreateTeamFragment newInstance(String param1, String param2) {  
 CreateTeamFragment fragment = new CreateTeamFragment();  
 Bundle args = new Bundle();  
 args.putString(*ARG\_PARAM1*, param1);  
 args.putString(*ARG\_PARAM2*, param2);  
 fragment.setArguments(args);  
 return fragment;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 if (getArguments() != null) {  
 mParam1 = getArguments().getString(*ARG\_PARAM1*);  
 mParam2 = getArguments().getString(*ARG\_PARAM2*);  
 }  
  
  
 }  
 private Button e\_pro\_create;  
 private TextView e\_P\_Name;  
 private TextView e\_P\_Decription;  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View view = inflater.inflate(R.layout.*fragment\_create\_team*, container, false);  
  
 e\_pro\_create = view.findViewById(R.id.*btn\_create\_project*);  
 e\_P\_Name = view.findViewById(R.id.*et\_new\_project\_name*);  
 e\_P\_Decription = view.findViewById(R.id.*et\_new\_description*);  
 e\_pro\_create.setOnClickListener(new View.OnClickListener() {  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 @Override  
 public void onClick(View v) {  
 String inputName = e\_P\_Name.getText().toString();  
 String inputDes = e\_P\_Decription.getText().toString();  
 if(inputName.equals("Enter New Project Name")){  
  
 }else{  
 Log.*e*("Here", inputName+" "+inputDes);  
 Log.*e*("Here", "button click");  
 MainActivity.*con*.Project\_Name = inputName;  
 MainActivity.*con*.Project\_Description = inputDes;  
 MainActivity.*con*.state = "CREATE\_NEW\_PROJECT";  
 }  
  
 }  
 });  
  
  
 return view;  
 }  
  
  
 public void test\_fun(View view){  
 Log.*e*("Here", "button click");  
 }  
}

package com.example.app;  
  
import android.os.Bundle;  
  
import androidx.fragment.app.Fragment;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \* Use the {****@link*** *DiaryFragment#newInstance} factory method to  
 \* create an instance of this fragment.  
 \*/*public class DiaryFragment extends Fragment {  
  
 // *TODO: Rename parameter arguments, choose names that match* // the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER  
 private static final String *ARG\_PARAM1* = "param1";  
 private static final String *ARG\_PARAM2* = "param2";  
  
 // *TODO: Rename and change types of parameters* private String mParam1;  
 private String mParam2;  
  
 public DiaryFragment() {  
 // Required empty public constructor  
 }  
  
 */\*\*  
 \* Use this factory method to create a new instance of  
 \* this fragment using the provided parameters.  
 \*  
 \** ***@param*** *param1 Parameter 1.  
 \** ***@param*** *param2 Parameter 2.  
 \** ***@return*** *A new instance of fragment DiaryFragment.  
 \*/* // *TODO: Rename and change types and number of parameters* public static DiaryFragment newInstance(String param1, String param2) {  
 DiaryFragment fragment = new DiaryFragment();  
 Bundle args = new Bundle();  
 args.putString(*ARG\_PARAM1*, param1);  
 args.putString(*ARG\_PARAM2*, param2);  
 fragment.setArguments(args);  
 return fragment;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 if (getArguments() != null) {  
 mParam1 = getArguments().getString(*ARG\_PARAM1*);  
 mParam2 = getArguments().getString(*ARG\_PARAM2*);  
 }  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 return inflater.inflate(R.layout.*fragment\_diary*, container, false);  
 }  
}

package com.example.app;  
  
import android.graphics.Point;  
  
*/\*\*  
 \* class stores individual task information needed it fill barCells  
 \*/*public class GanttItem {  
 private String taskName;  
 private boolean isError,isEmpty;  
 private Point point;  
 private int status;  
  
 public GanttItem(String taskName, boolean isError, Point point, int status) {  
 this.taskName = taskName;  
 this.isError = isError;  
 this.point = point;  
 this.status = status;  
 }  
  
 public GanttItem(String taskName, boolean isEmpty) {  
 this.taskName = taskName;  
 this.isEmpty = isEmpty;  
 }  
  
 public int getStatus() {  
 return status;  
 }  
  
 public String getTaskName() {  
 return taskName;  
 }  
  
 public void setTaskName(String taskName) {  
 this.taskName = taskName;  
 }  
  
 public boolean isError() {  
 return isError;  
 }  
  
 public void setError(boolean error) {  
 isError = error;  
 }  
  
 public boolean isEmpty() {  
 return isEmpty;  
 }  
  
 public void setEmpty(boolean empty) {  
 isEmpty = empty;  
 }  
  
 public Point getPoint() {  
 return point;  
 }  
  
 public void setPoint(Point point) {  
 this.point = point;  
 }  
}

package com.example.app;  
  
import android.content.Intent;  
import android.os.Bundle;  
  
import androidx.fragment.app.Fragment;  
import androidx.recyclerview.widget.DefaultItemAnimator;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
  
import java.util.ArrayList;  
import java.util.concurrent.Semaphore;  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \* Use the {****@link*** *HomeFragment#newInstance} factory method to  
 \* create an instance of this fragment.  
 \*  
 \* fill act\_home.java/activity\_home.xml activity home tab fragment  
 \*/*public class HomeFragment extends Fragment implements recyclerAdapter.OnProjectListener {  
 private recyclerAdapter.RecyclerViewClickListener listener;  
 private static ArrayList<Project\_Names\_list> *projectsList*;  
 private RecyclerView recyclerView;  
 public static String[] *Projects*;  
 public static Semaphore *s* = new java.util.concurrent.Semaphore(0);  
 public static boolean *check1* = false;  
  
 // *TODO: Rename parameter arguments, choose names that match* // the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER  
 private static final String *ARG\_PARAM1* = "param1";  
 private static final String *ARG\_PARAM2* = "param2";  
  
 // *TODO: Rename and change types of parameters* private String mParam1;  
 private String mParam2;  
  
 public HomeFragment() {  
 // Required empty public constructor  
 }  
  
 */\*\*  
 \* Use this factory method to create a new instance of  
 \* this fragment using the provided parameters.  
 \*  
 \** ***@param*** *param1 Parameter 1.  
 \** ***@param*** *param2 Parameter 2.  
 \** ***@return*** *A new instance of fragment HomeFragment.  
 \*/* // *TODO: Rename and change types and number of parameters* public static HomeFragment newInstance(String param1, String param2) {  
 HomeFragment fragment = new HomeFragment();  
 Bundle args = new Bundle();  
 args.putString(*ARG\_PARAM1*, param1);  
 args.putString(*ARG\_PARAM2*, param2);  
 fragment.setArguments(args);  
 return fragment;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 if (getArguments() != null) {  
 mParam1 = getArguments().getString(*ARG\_PARAM1*);  
 mParam2 = getArguments().getString(*ARG\_PARAM2*);  
 }  
 }  
 private static View *view*;  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
  
 // Inflate the layout for this fragment  
 *view* = inflater.inflate(R.layout.*fragment\_home*, container, false);  
 //get projects  
 MainActivity.*con*.state = "GET\_USER\_PROJECTS";  
 //  
 recyclerView = *view*.findViewById(R.id.*listRev\_1*);  
 *projectsList* = new ArrayList<>();  
 try {  
 setProjectinfo();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 setAdapter();  
  
 return *view*;  
 }  
  
 private void setAdapter() {  
 recyclerAdapter adapter = new recyclerAdapter(*projectsList*,listener,this);  
 RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(getActivity().getApplicationContext());  
 recyclerView.setLayoutManager(layoutManager);  
 recyclerView.setItemAnimator(new DefaultItemAnimator());  
 recyclerView.setAdapter(adapter);  
 }  
  
  
  
 private void setProjectinfo() throws InterruptedException {  
 //String[] Projects = new String[] { "Orange", "Apple", "Pear", "Strawberry" , "Strawberry" , "Strawberry" , "testing project 1 and some more words and stuff" };  
 while(*check1* == false){  
 if(*check1* == true){  
 break;  
 }  
 }  
 *check1* = false;  
 if(*projectsList* == null){  
 *projectsList*.add(new Project\_Names\_list("no projects in list"));  
 }else {  
 for (int i = 0; i < *Projects*.length; i++) {  
 *projectsList*.add(new Project\_Names\_list(*Projects*[i]));  
 }  
 }  
 }  
 public static void setDone() {  
 *s*.release();  
 }  
  
 public static void waitUntilDone() throws InterruptedException {  
 *s*.acquire();  
 }  
  
 @Override  
 public void onProjectClick(int position) {  
 //Intent intent = new Intent(MainActivity.this,temp\_Home.class);  
 //projectsList.get(position);  
 Intent intent = new Intent(getActivity(), Project\_activity.class);  
 startActivity(intent);  
 }  
}

package com.example.app;  
  
import android.os.Bundle;  
  
import androidx.appcompat.widget.SearchView;  
import androidx.fragment.app.Fragment;  
  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
import java.util.ArrayList;  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \* Use the {****@link*** *MessageesFragment#newInstance} factory method to  
 \* create an instance of this fragment.  
 \*  
 \* fill act\_home.java/activity\_home.xml activity messagesFragment tab fragment  
 \*/*public class MessageesFragment extends Fragment {  
 public static ArrayList<User\_object\_add\_user> *User\_List\_obj* = new ArrayList<User\_object\_add\_user>();  
 private ListView listView;  
  
 // *TODO: Rename parameter arguments, choose names that match* // the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER  
 private static final String *ARG\_PARAM1* = "param1";  
 private static final String *ARG\_PARAM2* = "param2";  
  
 // *TODO: Rename and change types of parameters* private String mParam1;  
 private String mParam2;  
  
 public MessageesFragment() {  
 // Required empty public constructor  
 }  
  
 */\*\*  
 \* Use this factory method to create a new instance of  
 \* this fragment using the provided parameters.  
 \*  
 \** ***@param*** *param1 Parameter 1.  
 \** ***@param*** *param2 Parameter 2.  
 \** ***@return*** *A new instance of fragment MessageesFragment.  
 \*/* // *TODO: Rename and change types and number of parameters* public static MessageesFragment newInstance(String param1, String param2) {  
 MessageesFragment fragment = new MessageesFragment();  
 Bundle args = new Bundle();  
 args.putString(*ARG\_PARAM1*, param1);  
 args.putString(*ARG\_PARAM2*, param2);  
 fragment.setArguments(args);  
 return fragment;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 if (getArguments() != null) {  
 mParam1 = getArguments().getString(*ARG\_PARAM1*);  
 mParam2 = getArguments().getString(*ARG\_PARAM2*);  
 }  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 // Inflate the layout for this fragment  
 View v = inflater.inflate(R.layout.*fragment\_messagees*, container, false);  
  
 *setUpData*();  
 setUpList(v);  
 initSearchWidgets(v);  
 setUpOnClick();  
  
  
 return v;  
 }  
 private void initSearchWidgets(View v){  
 SearchView searchView = v.findViewById(R.id.*userListSearchView*);  
  
 searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {  
 @Override  
 public boolean onQueryTextSubmit(String query) {  
 return false;  
 }  
  
 @Override  
 public boolean onQueryTextChange(String newText) {  
 ArrayList<User\_object\_add\_user> filteredUser = new ArrayList<User\_object\_add\_user>();  
 for(User\_object\_add\_user user : *User\_List\_obj*){  
 if(user.getName().toLowerCase().contains(newText.toLowerCase())){  
 filteredUser.add(user);  
 }  
 }  
 User\_object\_adapter adapter = new User\_object\_adapter(getContext(),android.R.layout.*simple\_list\_item\_1*,filteredUser);  
 listView.setAdapter(adapter);  
 return false;  
 }  
 });  
 }  
  
 public void setUpOnClick() {  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 Log.*e*("user clicked : ", String.*valueOf*(position));  
 }  
 });  
  
  
 }  
  
 public void setUpList(View v) {  
 User\_object\_adapter adapter = new User\_object\_adapter(getContext(),android.R.layout.*simple\_list\_item\_1*,*User\_List\_obj*);  
 listView = v.findViewById(R.id.*add\_user\_list\_view\_msg*);  
 listView.setAdapter(adapter);  
 }  
  
 static void setUpData() {  
 //User\_List\_obj = new ArrayList<User\_object\_add\_user>();  
 //User\_object\_add\_user test\_u = new User\_object\_add\_user("admin\_from\_client");  
 //User\_List\_obj.add(test\_u);  
 //User\_object\_add\_user test\_u2 = new User\_object\_add\_user("josh\_from\_client");  
 //User\_List\_obj.add(test\_u2);  
 if(*User\_List\_obj*.isEmpty()){  
 MainActivity.*con*.state = "GET\_ALL\_USERS\_2";  
 }  
  
 }  
}

package com.example.app;  
  
import android.app.DatePickerDialog;  
import android.content.pm.ActivityInfo;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.os.Bundle;  
  
import androidx.appcompat.widget.SearchView;  
import androidx.fragment.app.Fragment;  
  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.ExpandableListView;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.HashMap;  
  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \* Use the {****@link*** *P\_addTask#newInstance} factory method to  
 \* create an instance of this fragment.  
 \*/*public class P\_addTask extends Fragment {  
  
 private static final String *TAG* = "P\_addTask";  
 private TextView mDisplayDate;  
 private DatePickerDialog.OnDateSetListener mDateSetListener;  
  
 private TextView mDisplayDate2;  
 private DatePickerDialog.OnDateSetListener mDateSetListener2;  
  
  
 // *TODO: Rename parameter arguments, choose names that match* // the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER  
 private static final String *ARG\_PARAM1* = "param1";  
 private static final String *ARG\_PARAM2* = "param2";  
  
 // *TODO: Rename and change types of parameters* private String mParam1;  
 private String mParam2;  
  
 public P\_addTask() {  
 // Required empty public constructor  
 }  
  
 */\*\*  
 \* Use this factory method to create a new instance of  
 \* this fragment using the provided parameters.  
 \*  
 \** ***@param*** *param1 Parameter 1.  
 \** ***@param*** *param2 Parameter 2.  
 \** ***@return*** *A new instance of fragment P\_addTask.  
 \*/* // *TODO: Rename and change types and number of parameters* public static P\_addTask newInstance(String param1, String param2) {  
 P\_addTask fragment = new P\_addTask();  
 Bundle args = new Bundle();  
 args.putString(*ARG\_PARAM1*, param1);  
 args.putString(*ARG\_PARAM2*, param2);  
 fragment.setArguments(args);  
 return fragment;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 if (getArguments() != null) {  
 mParam1 = getArguments().getString(*ARG\_PARAM1*);  
 mParam2 = getArguments().getString(*ARG\_PARAM2*);  
 }  
  
 }  
 public static ArrayList<User\_object\_add\_user> *User\_List\_obj* = new ArrayList<User\_object\_add\_user>();  
 ArrayList<User\_object\_add\_user> filteredUser;  
  
 private ListView listView;  
 public static String *new\_user\_to\_project*;  
 private Button eSubmitTask;  
 public static EditText *task\_name*;  
 public static EditText *task\_des*;  
 public static TextView *task\_start*;  
 public static TextView *task\_end*;  
 public static String *select\_user*;  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 P\_overview.*get\_minor\_task* = false;  
 // Inflate the layout for this fragment  
 View v = inflater.inflate(R.layout.*fragment\_p\_add\_task*, container, false);  
 //  
 mDisplayDate = v.findViewById(R.id.*tvDateStart*);  
 mDisplayDate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Calendar cal = Calendar.*getInstance*();  
 int year = cal.get(Calendar.*YEAR*);  
 int month = cal.get(Calendar.*MONTH*);  
 int day = cal.get(Calendar.*DAY\_OF\_MONTH*);  
  
 DatePickerDialog dialog = new DatePickerDialog(  
 getContext(),  
 android.R.style.*Theme\_Holo\_Light\_Dialog\_MinWidth*,  
 mDateSetListener,  
 year,month,day);  
 dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*TRANSPARENT*));  
 dialog.show();  
 }  
 });  
  
 mDisplayDate2 = v.findViewById(R.id.*tvDateEnd*);  
 mDisplayDate2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Calendar cal = Calendar.*getInstance*();  
 int year = cal.get(Calendar.*YEAR*);  
 int month = cal.get(Calendar.*MONTH*);  
 int day = cal.get(Calendar.*DAY\_OF\_MONTH*);  
  
 DatePickerDialog dialog = new DatePickerDialog(  
 getContext(),  
 android.R.style.*Theme\_Holo\_Light\_Dialog\_MinWidth*,  
 mDateSetListener2,  
 year,month,day);  
 dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*TRANSPARENT*));  
 dialog.show();  
 }  
 });  
  
 mDateSetListener = new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
 month++;  
 Log.*d*(*TAG*,"date: "+dayOfMonth+"/"+month+"/"+year);  
 String date = "Start: "+dayOfMonth+"/"+month+"/"+year;  
 mDisplayDate.setText(date);  
 }  
 };  
 mDateSetListener2 = new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
 month++;  
 Log.*d*(*TAG*,"date2: "+dayOfMonth+"/"+month+"/"+year);  
 String date = "End: "+dayOfMonth+"/"+month+"/"+year;  
 mDisplayDate2.setText(date);  
 }  
 };  
 //select assigned users  
 try {  
 setUpData();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 setUpList(v);  
 initSearchWidgets(v);  
 //setUpOnClick();  
 //setup submit button and et's  
 *task\_name* = v.findViewById(R.id.*etTaskName*);  
 *task\_des* = v.findViewById(R.id.*etTaskDes*);  
 *task\_start* = v.findViewById(R.id.*tvDateStart*);  
 *task\_end* = v.findViewById(R.id.*tvDateEnd*);  
  
 eSubmitTask = v.findViewById(R.id.*btn\_SubmitTask*);  
 eSubmitTask.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 *select\_user* = null;  
 if(filteredUser == null){  
 *select\_user* = *User\_List\_obj*.get(0).getName();  
 }else if(filteredUser.isEmpty()){  
 *select\_user* = *User\_List\_obj*.get(0).getName();  
 }else if(!filteredUser.isEmpty()){  
 *select\_user* = filteredUser.get(0).getName();  
 }else{  
 //condition should never be used  
 *select\_user* = "No User Assigned to Task";  
 }  
 Log.*e*("task submit clicked : ", *task\_name*.getText().toString()+" "+*task\_des*.getText().toString()+" "+*task\_start*.getText().toString()+" "+*task\_end*.getText().toString()+" "+*select\_user*);  
 if(*task\_name*.getText().toString().isEmpty()){  
 Toast.*makeText*(getContext(),"Insert Task Name", Toast.*LENGTH\_LONG*).show();  
  
 }else if (*task\_start*.getText().toString().isEmpty()){  
 Toast.*makeText*(getContext(),"Insert Task Start Date", Toast.*LENGTH\_LONG*).show();  
 }else if (*task\_end*.getText().toString().isEmpty()){  
 Toast.*makeText*(getContext(),"Insert Task Start Date", Toast.*LENGTH\_LONG*).show();  
 }else{  
 MainActivity.*con*.state = "INSERT\_NEW\_TASK";  
  
 }  
 }  
 });  
 getActivity().setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_PORTRAIT*);  
 //  
 return v;  
 }  
  
 private void initSearchWidgets(View v){  
 SearchView searchView = v.findViewById(R.id.*userListSearchView2*);  
  
 searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {  
 @Override  
 public boolean onQueryTextSubmit(String query) {  
 return false;  
 }  
  
 @Override  
 public boolean onQueryTextChange(String newText) {  
 filteredUser = new ArrayList<User\_object\_add\_user>();  
 for(User\_object\_add\_user user : *User\_List\_obj*){  
 if(user.getName().toLowerCase().contains(newText.toLowerCase())){  
 filteredUser.add(user);  
 }  
 }  
 User\_object\_adapter adapter = new User\_object\_adapter(getContext(),android.R.layout.*simple\_list\_item\_1*,filteredUser);  
 listView.setAdapter(adapter);  
 return false;  
 }  
 });  
 }  
  
 public void setUpOnClick() {  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 Log.*e*("user clicked : ", String.*valueOf*(position));  
 *new\_user\_to\_project* = *User\_List\_obj*.get(position).getName();  
 }  
 });  
  
  
 }  
  
 public void setUpList(View v){  
 while(true){  
 if(*User\_List\_obj* != null){  
 break;  
 }  
 }  
 User\_object\_adapter adapter = new User\_object\_adapter(getContext(),android.R.layout.*simple\_list\_item\_1*,*User\_List\_obj*);  
 listView = v.findViewById(R.id.*add\_user\_list\_view2*);  
 listView.setAdapter(adapter);  
 }  
  
 public void setUpData() throws InterruptedException {  
 //User\_List\_obj = new ArrayList<User\_object\_add\_user>();  
 //User\_object\_add\_user test\_u = new User\_object\_add\_user("admin\_from\_client");  
 //User\_List\_obj.add(test\_u);  
 //User\_object\_add\_user test\_u2 = new User\_object\_add\_user("josh\_from\_client");  
 //User\_List\_obj.add(test\_u2);  
 //  
 if(*User\_List\_obj*.isEmpty()){  
 MainActivity.*con*.state = "GET\_ALL\_USERS\_IN\_PROJECT";  
 }  
  
  
  
  
 }  
}

package com.example.app;  
  
import android.annotation.SuppressLint;  
import android.content.pm.ActivityInfo;  
import android.os.Bundle;  
  
import androidx.annotation.NonNull;  
import androidx.fragment.app.Fragment;  
  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import androidx.appcompat.widget.SearchView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \* Use the {****@link*** *P\_addUser#newInstance} factory method to  
 \* create an instance of this fragment.  
 \*/*public class P\_addUser extends Fragment {  
 //initialize list var  
 ListView add\_users\_listView;  
 ArrayList<String> StringArrayList = new ArrayList<>();  
 ArrayAdapter add\_users\_adapter;  
 public static String *new\_user\_to\_project*;  
  
  
  
 // *TODO: Rename parameter arguments, choose names that match* // the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER  
 private static final String *ARG\_PARAM1* = "param1";  
 private static final String *ARG\_PARAM2* = "param2";  
  
 // *TODO: Rename and change types of parameters* private String mParam1;  
 private String mParam2;  
  
 public P\_addUser() {  
 // Required empty public constructor  
 }  
  
 */\*\*  
 \* Use this factory method to create a new instance of  
 \* this fragment using the provided parameters.  
 \*  
 \** ***@param*** *param1 Parameter 1.  
 \** ***@param*** *param2 Parameter 2.  
 \** ***@return*** *A new instance of fragment P\_addUser.  
 \*/* // *TODO: Rename and change types and number of parameters* public static P\_addUser newInstance(String param1, String param2) {  
 P\_addUser fragment = new P\_addUser();  
 Bundle args = new Bundle();  
 args.putString(*ARG\_PARAM1*, param1);  
 args.putString(*ARG\_PARAM2*, param2);  
 fragment.setArguments(args);  
 return fragment;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 if (getArguments() != null) {  
 mParam1 = getArguments().getString(*ARG\_PARAM1*);  
 mParam2 = getArguments().getString(*ARG\_PARAM2*);  
 }  
 }  
  
 public static ArrayList<User\_object\_add\_user> *User\_List\_obj* = new ArrayList<User\_object\_add\_user>();  
 private ListView listView;  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 P\_overview.*get\_minor\_task* = false;  
 // Inflate the layout for this fragment  
 View v = inflater.inflate(R.layout.*fragment\_p\_add\_user*, container, false);  
 //  
  
 *setUpData*();  
 setUpList(v);  
 initSearchWidgets(v);  
 setUpOnClick();  
  
  
  
 //  
 getActivity().setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_PORTRAIT*);  
 return v;  
 }  
  
 private void initSearchWidgets(View v){  
 SearchView searchView = v.findViewById(R.id.*userListSearchView*);  
  
 searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {  
 @Override  
 public boolean onQueryTextSubmit(String query) {  
 return false;  
 }  
  
 @Override  
 public boolean onQueryTextChange(String newText) {  
 ArrayList<User\_object\_add\_user> filteredUser = new ArrayList<User\_object\_add\_user>();  
 for(User\_object\_add\_user user : *User\_List\_obj*){  
 if(user.getName().toLowerCase().contains(newText.toLowerCase())){  
 filteredUser.add(user);  
 }  
 }  
 User\_object\_adapter adapter = new User\_object\_adapter(getContext(),android.R.layout.*simple\_list\_item\_1*,filteredUser);  
 listView.setAdapter(adapter);  
 return false;  
 }  
 });  
 }  
  
 public void setUpOnClick() {  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 Log.*e*("user clicked : ", String.*valueOf*(position));  
 *new\_user\_to\_project* = *User\_List\_obj*.get(position).getName();  
 MainActivity.*con*.state = "ASSIGN\_USER\_TO\_PROJECT";  
  
 }  
 });  
  
  
 }  
  
 public void setUpList(View v) {  
 User\_object\_adapter adapter = new User\_object\_adapter(getContext(),android.R.layout.*simple\_list\_item\_1*,*User\_List\_obj*);  
 listView = v.findViewById(R.id.*add\_user\_list\_view*);  
 listView.setAdapter(adapter);  
 }  
  
 static void setUpData() {  
 //User\_List\_obj = new ArrayList<User\_object\_add\_user>();  
 //User\_object\_add\_user test\_u = new User\_object\_add\_user("admin\_from\_client");  
 //User\_List\_obj.add(test\_u);  
 //User\_object\_add\_user test\_u2 = new User\_object\_add\_user("josh\_from\_client");  
 //User\_List\_obj.add(test\_u2);  
 if(*User\_List\_obj*.isEmpty()){  
 MainActivity.*con*.state = "GET\_ALL\_USERS";  
 }  
  
 }  
  
  
  
  
}

package com.example.app;  
  
import android.content.pm.ActivityInfo;  
import android.os.Bundle;  
  
import androidx.fragment.app.Fragment;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \* Use the {****@link*** *P\_Home#newInstance} factory method to  
 \* create an instance of this fragment.  
 \*/*public class P\_Home extends Fragment {  
  
 // *TODO: Rename parameter arguments, choose names that match* // the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER  
 private static final String *ARG\_PARAM1* = "param1";  
 private static final String *ARG\_PARAM2* = "param2";  
  
 // *TODO: Rename and change types of parameters* private String mParam1;  
 private String mParam2;  
  
 public P\_Home() {  
 // Required empty public constructor  
 }  
  
 */\*\*  
 \* Use this factory method to create a new instance of  
 \* this fragment using the provided parameters.  
 \*  
 \** ***@param*** *param1 Parameter 1.  
 \** ***@param*** *param2 Parameter 2.  
 \** ***@return*** *A new instance of fragment P\_Home.  
 \*/* // *TODO: Rename and change types and number of parameters* public static P\_Home newInstance(String param1, String param2) {  
 P\_Home fragment = new P\_Home();  
 Bundle args = new Bundle();  
 args.putString(*ARG\_PARAM1*, param1);  
 args.putString(*ARG\_PARAM2*, param2);  
 fragment.setArguments(args);  
 return fragment;  
 }  
 private TextView text\_view\_name;  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 if (getArguments() != null) {  
 mParam1 = getArguments().getString(*ARG\_PARAM1*);  
 mParam2 = getArguments().getString(*ARG\_PARAM2*);  
 }  
 }  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 P\_overview.*get\_minor\_task* = false;  
 // Inflate the layout for this fragment  
 View v = inflater.inflate(R.layout.*fragment\_p\_home*, container, false);  
 TextView tv = v.findViewById(R.id.*p\_home*);  
 tv.setText(MainActivity.*con*.Currently\_selected\_project\_view);  
 getActivity().setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_PORTRAIT*);  
 return v;  
 }  
}

package com.example.app;  
  
import android.app.DatePickerDialog;  
import android.content.Intent;  
import android.content.pm.ActivityInfo;  
import android.graphics.Color;  
import android.graphics.Point;  
import android.graphics.drawable.ColorDrawable;  
import android.os.Bundle;  
  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentTransaction;  
  
import android.text.format.Time;  
import android.util.Log;  
import android.util.TimeUtils;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.example.app.Common.Common;  
import com.example.app.GanttAdapter.GanttTableFixHeaderAdapter;  
import com.example.app.ViewGroup.IOnHeaderClickListener;  
import com.inqbarna.tablefixheaders.TableFixHeaders;  
  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Collection;  
import java.util.Collections;  
import java.util.Date;  
import java.util.List;  
import java.util.concurrent.TimeUnit;  
  
*/\*\*  
 \* A simple {****@link*** *Fragment} subclass.  
 \* Use the {****@link*** *P\_overview#newInstance} factory method to  
 \* create an instance of this fragment.  
 \*/*public class P\_overview extends Fragment implements IOnHeaderClickListener {  
  
 TableFixHeaders tableFixHeaders;  
 public static boolean *check1* = false;  
 public static boolean *week\_or\_day* = true;  
 public static boolean *get\_minor\_task* = false;  
 public static TextView *task\_name*;  
 private Button eWeek\_to\_day;  
  
 // *TODO: Rename parameter arguments, choose names that match* // the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER  
 private static final String *ARG\_PARAM1* = "param1";  
 private static final String *ARG\_PARAM2* = "param2";  
  
 // *TODO: Rename and change types of parameters* private String mParam1;  
 private String mParam2;  
  
 public P\_overview() {  
 // Required empty public constructor  
 }  
  
 */\*\*  
 \* Use this factory method to create a new instance of  
 \* this fragment using the provided parameters.  
 \*  
 \** ***@param*** *param1 Parameter 1.  
 \** ***@param*** *param2 Parameter 2.  
 \** ***@return*** *A new instance of fragment P\_overview.  
 \*/* // *TODO: Rename and change types and number of parameters* public static P\_overview newInstance(String param1, String param2) {  
 P\_overview fragment = new P\_overview();  
 Bundle args = new Bundle();  
 args.putString(*ARG\_PARAM1*, param1);  
 args.putString(*ARG\_PARAM2*, param2);  
 fragment.setArguments(args);  
 return fragment;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 if (getArguments() != null) {  
 mParam1 = getArguments().getString(*ARG\_PARAM1*);  
 mParam2 = getArguments().getString(*ARG\_PARAM2*);  
 }  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 View v = inflater.inflate(R.layout.*fragment\_p\_overview*, container, false);  
 //  
 tableFixHeaders = (TableFixHeaders)v.findViewById(R.id.*tablefixheaders*);  
 if(!*get\_minor\_task*){  
 MainActivity.*con*.state = "GET\_PROJECT\_TASK\_INFORMATION";  
 }else{  
 MainActivity.*con*.state = "GET\_PROJECT\_MINOR\_TASK\_INFORMATION";  
 //MainActivity.con.state = "GET\_PROJECT\_TASK\_INFORMATION";  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "minor tasks");  
 //P\_overview.get\_minor\_task = true;  
 }  
 //  
 //P\_overview.get\_minor\_task = false;  
  
 createGanttChart();  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of gantt creation");  
  
  
  
 eWeek\_to\_day = v.findViewById(R.id.*overview\_title*);  
  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of gantt creation");  
  
 if(*week\_or\_day*){  
 eWeek\_to\_day.setText("TIME LINE - WEEK");  
 }else{  
 eWeek\_to\_day.setText("TIME LINE - DAY");  
 }  
  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of gantt creation");  
  
 eWeek\_to\_day.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 *week\_or\_day* = !*week\_or\_day*;  
 FragmentTransaction ft = getFragmentManager().beginTransaction();  
 ft.detach(P\_overview.this).attach(P\_overview.this).commit();  
 }  
 });  
  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "end of gantt creation");  
  
 return v;  
 }  
 //error is being created bent week x and week 9 = 0   
 private void createGanttChart() {  
 //wait for data  
 while(true){  
 if(*check1* == true){  
 break;  
 }  
 try {  
 TimeUnit.*MILLISECONDS*.sleep(100);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 *check1* = false;  
 // get gantt chart creation values  
 ArrayList<Date> All\_dates = new ArrayList<>();  
 for(int i = 0; i < MainActivity.*con*.Project\_Tasks.size();i++){  
 All\_dates.add(MainActivity.*con*.Project\_Tasks.get(i).Start);  
 All\_dates.add(MainActivity.*con*.Project\_Tasks.get(i).End);  
 }  
 if(All\_dates == null || All\_dates.isEmpty()){  
 All\_dates.add(new Date());  
 All\_dates.add(new Date());  
 }  
 Date Max = Collections.*max*(All\_dates);  
 Date Min = Collections.*min*(All\_dates);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Date stuff "+String.*valueOf*(Max));  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Date stuff "+String.*valueOf*(Min));  
  
 int weeks;  
 long diffSeconds;  
  
 if(*week\_or\_day*){  
 diffSeconds = (Max.getTime() - Min.getTime())/1000;  
 weeks = (int)diffSeconds/(60 \* 60 \* 24 \* 7);  
 }else{  
 diffSeconds = (Max.getTime() - Min.getTime())/1000;  
 weeks = (int)diffSeconds/(60 \* 60 \* 24);  
 }  
  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Date stuff "+String.*valueOf*(weeks));  
 Common.*HEADER\_COUNT* = weeks+1;  
 Common.*COLUMN\_COUNT* = weeks\*2+1;  
  
 Max = new Date();  
 if(*week\_or\_day*){  
 diffSeconds = (Max.getTime() - Min.getTime())/1000;  
 weeks = (int)diffSeconds/(60 \* 60 \* 24 \* 7);  
 }else{  
 diffSeconds = (Max.getTime() - Min.getTime())/1000;  
 weeks = (int)diffSeconds/(60 \* 60 \* 24);  
 }  
  
 Common.*CURRENT\_WEEK* = weeks;  
 //create list item  
 List<GanttItem> ganttItemList = new ArrayList<>();  
 //repeat to write incomplete jobs  
 for(int i = 0; i < MainActivity.*con*.Project\_Tasks.size();i++){  
  
 int weeks\_x;  
 int weeks\_y;  
 if(*week\_or\_day*){  
 long diffSeconds\_x = ( MainActivity.*con*.Project\_Tasks.get(i).Start.getTime()-Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_x = (int)diffSeconds\_x/(60 \* 60 \* 24 \* 7);  
  
 long diffSeconds\_y = ( MainActivity.*con*.Project\_Tasks.get(i).End.getTime() - Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_y = (int)diffSeconds\_y/(60 \* 60 \* 24 \* 7);  
 }else{  
 long diffSeconds\_x = ( MainActivity.*con*.Project\_Tasks.get(i).Start.getTime()-Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_x = (int)diffSeconds\_x/(60 \* 60 \* 24);  
  
 long diffSeconds\_y = ( MainActivity.*con*.Project\_Tasks.get(i).End.getTime() - Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_y = (int)diffSeconds\_y/(60 \* 60 \* 24);  
 }  
  
 if(MainActivity.*con*.Project\_Tasks.get(i).Status == 1){  
 if(weeks\_y < weeks\_x){  
 ganttItemList.add(new GanttItem(MainActivity.*con*.Project\_Tasks.get(i).Task\_Name, true));  
 }else if(weeks\_y == weeks\_x){  
 ganttItemList.add(new GanttItem(MainActivity.*con*.Project\_Tasks.get(i).Task\_Name, false, new Point(weeks\_x, weeks\_y), MainActivity.*con*.Project\_Tasks.get(i).Status));  
 }else{  
 ganttItemList.add(new GanttItem(MainActivity.*con*.Project\_Tasks.get(i).Task\_Name, false, new Point(weeks\_x, weeks\_y), MainActivity.*con*.Project\_Tasks.get(i).Status));  
 }  
 }else{  
 //ganttItemList.add(new GanttItem(MainActivity.con.Project\_Tasks.get(i).Task\_Name , true,new Point(weeks\_x,weeks\_y)));  
 }  
  
 //ganttItemList.add(new GanttItem(MainActivity.con.Project\_Tasks.get(i).Task\_Name ,true));  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Date stuff weeks\_x "+String.*valueOf*(weeks\_x));  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Date stuff weeks\_y "+String.*valueOf*(weeks\_y));  
  
 }  
 //write all complete  
 for(int i = 0; i < MainActivity.*con*.Project\_Tasks.size();i++){  
  
 int weeks\_x;  
 int weeks\_y;  
 if(*week\_or\_day*){  
 long diffSeconds\_x = ( MainActivity.*con*.Project\_Tasks.get(i).Start.getTime()-Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_x = (int)diffSeconds\_x/(60 \* 60 \* 24 \* 7);  
  
 long diffSeconds\_y = ( MainActivity.*con*.Project\_Tasks.get(i).End.getTime() - Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_y = (int)diffSeconds\_y/(60 \* 60 \* 24 \* 7);  
 }else{  
 long diffSeconds\_x = ( MainActivity.*con*.Project\_Tasks.get(i).Start.getTime()-Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_x = (int)diffSeconds\_x/(60 \* 60 \* 24);  
  
 long diffSeconds\_y = ( MainActivity.*con*.Project\_Tasks.get(i).End.getTime() - Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_y = (int)diffSeconds\_y/(60 \* 60 \* 24);  
 }  
  
 if(MainActivity.*con*.Project\_Tasks.get(i).Status == 2){  
 if(weeks\_y < weeks\_x){  
 ganttItemList.add(new GanttItem(MainActivity.*con*.Project\_Tasks.get(i).Task\_Name, true));  
 }else if(weeks\_y == weeks\_x){  
 ganttItemList.add(new GanttItem(MainActivity.*con*.Project\_Tasks.get(i).Task\_Name, false, new Point(weeks\_x, weeks\_y), MainActivity.*con*.Project\_Tasks.get(i).Status));  
 }else{  
 ganttItemList.add(new GanttItem(MainActivity.*con*.Project\_Tasks.get(i).Task\_Name, false, new Point(weeks\_x, weeks\_y), MainActivity.*con*.Project\_Tasks.get(i).Status));  
 }  
 }else{  
 //ganttItemList.add(new GanttItem(MainActivity.con.Project\_Tasks.get(i).Task\_Name , true,new Point(weeks\_x,weeks\_y)));  
 }  
  
 //ganttItemList.add(new GanttItem(MainActivity.con.Project\_Tasks.get(i).Task\_Name ,true));  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Date stuff weeks\_x "+String.*valueOf*(weeks\_x));  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Date stuff weeks\_y "+String.*valueOf*(weeks\_y));  
  
 }  
 //write all cancelled  
 for(int i = 0; i < MainActivity.*con*.Project\_Tasks.size();i++){  
 int weeks\_x;  
 int weeks\_y;  
 if(*week\_or\_day*){  
 long diffSeconds\_x = ( MainActivity.*con*.Project\_Tasks.get(i).Start.getTime()-Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_x = (int)diffSeconds\_x/(60 \* 60 \* 24 \* 7);  
  
 long diffSeconds\_y = ( MainActivity.*con*.Project\_Tasks.get(i).End.getTime() - Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_y = (int)diffSeconds\_y/(60 \* 60 \* 24 \* 7);  
 }else{  
 long diffSeconds\_x = ( MainActivity.*con*.Project\_Tasks.get(i).Start.getTime()-Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_x = (int)diffSeconds\_x/(60 \* 60 \* 24);  
  
 long diffSeconds\_y = ( MainActivity.*con*.Project\_Tasks.get(i).End.getTime() - Collections.*min*(All\_dates).getTime() )/1000;  
 weeks\_y = (int)diffSeconds\_y/(60 \* 60 \* 24);  
 }  
  
 if(MainActivity.*con*.Project\_Tasks.get(i).Status == 3){  
 if(weeks\_y < weeks\_x){  
 ganttItemList.add(new GanttItem(MainActivity.*con*.Project\_Tasks.get(i).Task\_Name, true));  
 }else if(weeks\_y == weeks\_x){  
 ganttItemList.add(new GanttItem(MainActivity.*con*.Project\_Tasks.get(i).Task\_Name, false, new Point(weeks\_x, weeks\_y), MainActivity.*con*.Project\_Tasks.get(i).Status));  
 }else{  
 ganttItemList.add(new GanttItem(MainActivity.*con*.Project\_Tasks.get(i).Task\_Name, false, new Point(weeks\_x, weeks\_y), MainActivity.*con*.Project\_Tasks.get(i).Status));  
 }  
 }else{  
 //ganttItemList.add(new GanttItem(MainActivity.con.Project\_Tasks.get(i).Task\_Name , true,new Point(weeks\_x,weeks\_y)));  
 }  
  
 //ganttItemList.add(new GanttItem(MainActivity.con.Project\_Tasks.get(i).Task\_Name ,true));  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Date stuff weeks\_x "+String.*valueOf*(weeks\_x));  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Date stuff weeks\_y "+String.*valueOf*(weeks\_y));  
  
 }  
 if(ganttItemList == null || ganttItemList.isEmpty() ){  
 ganttItemList.add(new GanttItem("no project information" , true));  
 Common.*HEADER\_COUNT* = 20;  
 Common.*COLUMN\_COUNT* = 40;  
 Common.*CURRENT\_WEEK* = 1;  
 }  
 //adapter  
 GanttTableFixHeaderAdapter adapter = new GanttTableFixHeaderAdapter(getContext(),ganttItemList);  
 //set listener late  
  
 //create body  
 List<List<String>> body = getBody(ganttItemList);  
  
 adapter.setFirstHeader("Task Name");  
 adapter.setHeader(getHeader());  
 adapter.setFirstBody(body);  
 adapter.setBody(body);  
 adapter.setSection(body);  
 adapter.setListener(this);  
  
 //set adapter  
 tableFixHeaders.setAdapter(adapter);  
  
 }  
  
 private List<String> getHeader() {  
 List<String> headers = new ArrayList<>();  
 for(int i =0;i<=Common.*HEADER\_COUNT*;i++)  
 headers.add(new StringBuilder().append(i).toString());  
 return headers;  
 }  
  
 private List<List<String>> getBody(List<GanttItem> ganttItemList) {  
 List<List<String>> rows = new ArrayList<>();  
 for(GanttItem ganttItem: ganttItemList){  
 //each gantt item will create body  
 List<String> cols = new ArrayList<>();  
 if(!ganttItem.isEmpty()){  
 for(int col = 0; col< Common.*COLUMN\_COUNT*;col++) {  
  
 if(ganttItem.getStatus() == 2){  
 if(col == Common.*CURRENT\_WEEK*){  
 cols.add("complete & current\_week");  
 }else{  
 cols.add("complete");  
 }  
 }else if (col >= ganttItem.getPoint().x) {  
 if (col <= ganttItem.getPoint().y) {  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "X : " + ganttItem.getPoint().x);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Y : " + ganttItem.getPoint().y);  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "Col : " + col);  
  
 if(col == Common.*CURRENT\_WEEK* && ganttItem.isError() ){  
 cols.add("done & current\_week");  
 } else if(col == Common.*CURRENT\_WEEK* && !ganttItem.isError()){  
 cols.add("done & current\_week");  
 }else if (ganttItem.isError()) {  
 cols.add("error");  
 } else {  
 cols.add("done");  
 }  
 } else if (col <= Common.*CURRENT\_WEEK*) {  
 cols.add("overtime");  
 }else{  
 if(col == Common.*CURRENT\_WEEK*){  
 cols.add("current\_week");  
 }else {  
 cols.add("default");  
 }  
 }  
 }else{  
 if(col == Common.*CURRENT\_WEEK*){  
 cols.add("current\_week");  
 }else {  
 cols.add("default");  
 }  
 }  
 }  
  
 }else{//if just empty row  
 for(int col = 0; col< Common.*COLUMN\_COUNT*;col++){  
 if(col == Common.*CURRENT\_WEEK*){  
 cols.add("current\_week");  
 }else {  
 cols.add("default");  
 }  
 }  
 }  
 rows.add(cols);  
 }  
 Log.*e*("YOUR\_APP\_LOG\_TAG\_2", "rows : " + rows);  
 return rows;  
 }  
 public static String *sel\_task*;  
  
 @Override  
 public void onHeaderItemClick(View view, int row) {  
  
 TextView textView = (TextView)view;  
 *task\_name* = textView;  
 //Toast.makeText(getContext(),"item click "+row+" Content "+textView.getText(), Toast.LENGTH\_LONG).show();  
 MainActivity.*con*.state = "GET\_TASK\_INFO";  
  
 Intent intent = new Intent(getContext(),Task\_Page.class);  
 startActivity(intent);  
 }  
}

package com.example.app;  
  
import android.os.Bundle;  
import android.widget.ListView;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.navigation.NavController;  
import androidx.navigation.Navigation;  
import androidx.navigation.ui.NavigationUI;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
  
import java.util.ArrayList;  
  
*/\*\*  
 \* inflates individual projects activity  
 \*/*public class Project\_activity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_project*);  
  
 BottomNavigationView bottomNavigationView = findViewById(R.id.*bnv\_project*);  
 NavController navController = Navigation.*findNavController*(this, R.id.*fragment\_projects\_main*);  
 NavigationUI.*setupWithNavController*(bottomNavigationView, navController);  
 }  
}

package com.example.app;  
  
import android.os.Bundle;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.navigation.NavController;  
import androidx.navigation.Navigation;  
import androidx.navigation.ui.NavigationUI;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
  
public class Project\_activity\_minor\_tasks extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_project*);  
  
 BottomNavigationView bottomNavigationView = findViewById(R.id.*bnv\_project*);  
 NavController navController = Navigation.*findNavController*(this, R.id.*fragment\_projects\_main*);  
 NavigationUI.*setupWithNavController*(bottomNavigationView, navController);  
 }  
}

package com.example.app;  
  
public class Project\_Names\_list {  
 public String Project\_names;  
  
 public Project\_Names\_list(String Project\_names) {  
 this.Project\_names = Project\_names;  
 }  
  
 public String getProject\_names(){  
 return Project\_names;  
 }  
 public void setProject\_names(String Project\_names){  
 this.Project\_names = Project\_names;  
 }  
}

package com.example.app;  
  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.ArrayList;  
  
*/\*\*  
 \* adapter class for projects list in fragment\_home.xml  
 \*/*public class recyclerAdapter extends RecyclerView.Adapter<recyclerAdapter.myViewHolder> {  
  
 private OnProjectListener mOnProjectListener;  
 private ArrayList<Project\_Names\_list> Projects;  
 public recyclerAdapter(ArrayList<Project\_Names\_list> Projects, RecyclerViewClickListener listener,OnProjectListener onProjectListener){  
 this.Projects = Projects;  
 this.mOnProjectListener = onProjectListener;  
 }  
  
 public class myViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {  
 private TextView nametxt;  
 OnProjectListener onProjectListener;  
  
 public myViewHolder(final View view, OnProjectListener onProjectListener){  
 super(view);  
 nametxt = view.findViewById(R.id.*P\_Name*);  
 this.onProjectListener = onProjectListener;  
 itemView.setOnClickListener(this);  
  
 }  
  
 @Override  
 public void onClick(View v) {  
 onProjectListener.onProjectClick(getAbsoluteAdapterPosition());  
 //find project name that was selected  
 Log.*e*("find project tab", Projects.get(getAbsoluteAdapterPosition()).Project\_names);  
 MainActivity.*con*.Currently\_selected\_project\_view = Projects.get(getAbsoluteAdapterPosition()).Project\_names;  
  
 }  
 }  
  
  
 @NonNull  
 @Override  
 public recyclerAdapter.myViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View itemView = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*list\_items\_home\_frag*, parent, false);  
 return new myViewHolder(itemView, mOnProjectListener);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull recyclerAdapter.myViewHolder holder, int position) {  
 String name = Projects.get(position).getProject\_names();  
 holder.nametxt.setText(name);  
  
 }  
  
 @Override  
 public int getItemCount() {  
 return Projects.size();  
 }  
  
 public interface RecyclerViewClickListener{  
 void onclick(View v, int position);  
 }  
  
 public interface OnProjectListener{  
 void onProjectClick(int position);  
 }  
  
  
}

package com.example.app;  
  
import android.os.Build;  
import android.util.Log;  
  
import androidx.annotation.RequiresApi;  
  
import java.security.PrivateKey;  
import java.security.PublicKey;  
import java.util.Base64;  
import javax.crypto.Cipher;  
import java.security.KeyPair;  
import java.security.KeyPairGenerator;  
import java.security.PrivateKey;  
import java.security.PublicKey;  
import java.util.Base64;  
import java.util.HashMap;  
import java.util.Map;  
  
import javax.crypto.Cipher;  
  
*/\*\*  
 \* RSA key exchange class  
 \*/*public class RSA {  
  
 public static PrivateKey *privateKey*;  
 public static PublicKey *publicKey*;  
  
  
 @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public static byte[] encryptMessage\_cipher(String plainText, PrivateKey ret) throws Exception {  
 Cipher cipher = Cipher.*getInstance*("RSA/ECB/PKCS1Padding");  
 cipher.init(Cipher.*ENCRYPT\_MODE*, ret);  
  
 return Base64.*getEncoder*().encode(cipher.doFinal(plainText.getBytes("UTF-8")));  
 }  
  
  
 */\*\*  
 \* unused function only use by server  
 \** ***@param*** *encryptedText  
 \** ***@param*** *publicKey  
 \** ***@return*** *\** ***@throws*** *Exception  
 \*/* @RequiresApi(api = Build.VERSION\_CODES.*O*)  
 public static String decryptMessage\_cipher(byte[] encryptedText , PublicKey publicKey) throws Exception {  
  
 Cipher cipher = Cipher.*getInstance*("RSA");  
 cipher.init(Cipher.*DECRYPT\_MODE*, publicKey);  
 byte[] de\_test = cipher.doFinal(Base64.*getDecoder*().decode(encryptedText));  
 return new String(de\_test);  
 //return new String(plainText);  
 }  
  
 */\*\*  
 \* unused function only use by server  
 \** ***@throws*** *Exception  
 \*/* public static void getRSAKeys() throws Exception {  
 KeyPairGenerator keyPairGenerator = KeyPairGenerator.*getInstance*("RSA");  
 keyPairGenerator.initialize(3072);  
 KeyPair keyPair = keyPairGenerator.generateKeyPair();  
 *privateKey* = keyPair.getPrivate();  
 *publicKey* = keyPair.getPublic();  
  
 Map<String, Object> keys = new HashMap<String,Object>();  
 keys.put("private", *privateKey*);  
 keys.put("public", *publicKey*);  
 }  
  
}

package com.example.app;  
  
import java.util.Date;  
  
*/\*\*  
 \* class stores individual task information  
 \* different to gantt item as this class will store more then minimum to display information  
 \* as well as start and end times in date form  
 \*/*public class Task\_Object {  
 public String Task\_Name;  
 public int Status;  
 public Date Start;  
 public Date End;  
  
 public Task\_Object(String task\_Name, int status, Date start, Date end) {  
 Task\_Name = task\_Name;  
 Status = status;  
 Start = start;  
 End = end;  
 }  
}

package com.example.app;  
  
import android.app.DatePickerDialog;  
import android.app.FragmentTransaction;  
import android.content.Intent;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.ListView;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.navigation.NavController;  
import androidx.navigation.Navigation;  
import androidx.navigation.ui.NavigationUI;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.concurrent.TimeUnit;  
  
public class Task\_Page extends AppCompatActivity {  
  
 public static String *Des*;  
 public static String *Status*;  
 public static String *Assigned\_user*;  
 public static String *s\_date*;  
 public static String *e\_date*;  
 public static String *e\_taskname*;  
  
 public static ArrayList<User\_object\_add\_user> *User\_List\_obj* = new ArrayList<User\_object\_add\_user>();  
 ArrayList<User\_object\_add\_user> filteredUser;  
 private ListView listView;  
 public static String *new\_user\_to\_project*;  
  
 private static final String *TAG* = "Task Page";  
 private TextView mDisplayDate;  
 private DatePickerDialog.OnDateSetListener mDateSetListener;  
  
 private TextView mDisplayDate2;  
 private DatePickerDialog.OnDateSetListener mDateSetListener2;  
  
 public static EditText *task\_name*;  
 public static EditText *task\_des*;  
 public static TextView *task\_start*;  
 public static TextView *task\_end*;  
 public static String *select\_user*;  
  
 public static boolean *check1* = false;  
 public static boolean *check2* = false;  
  
 RadioGroup radioGroup;  
 RadioButton radioButton;  
 public static String *selected\_radio\_option* = null;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_task\_info*);  
  
 TextView tv\_Title;  
 TextView tv\_Description;  
 TextView tv\_Status;  
 TextView tv\_ResponsibleUser;  
 TextView tv\_StartDate;  
 TextView tv\_EndDate;  
  
 tv\_Title = findViewById(R.id.*TaskTitle\_taskinfopage*);  
 tv\_Description = findViewById(R.id.*TaskDes*);  
 tv\_Status = findViewById(R.id.*TaskStatus*);  
 tv\_ResponsibleUser = findViewById(R.id.*TaskResponsibleUser*);  
 tv\_StartDate = findViewById(R.id.*TaskStartDate*);  
 tv\_EndDate = findViewById(R.id.*TaskEndDate*);  
  
 //  
 Button ViewMinorTasks = findViewById(R.id.*btn\_ViewMinorTasks*);  
 ViewMinorTasks.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v) {  
 P\_overview.*get\_minor\_task* = true;  
 startActivity(new Intent(Task\_Page.this, Project\_activity.class));  
  
 }  
 });  
  
 //update status radio button  
 radioGroup = findViewById(R.id.*radioGroupStatusUpdate*);  
  
 Button buttonApplyStutus = findViewById(R.id.*btn\_change\_task\_status*);  
 buttonApplyStutus.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 int radioId = radioGroup.getCheckedRadioButtonId();  
  
 radioButton = findViewById(radioId);  
 //Toast.makeText(getApplicationContext(),"Selected Radio Button: " + radioButton.getText(),Toast.LENGTH\_LONG).show();  
 if(radioButton.getText().equals("in progress")){  
 *selected\_radio\_option* = "1";  
 }else if(radioButton.getText().equals("Resolved")){  
 *selected\_radio\_option* = "2";  
 }else if(radioButton.getText().equals("Cancelled")){  
 *selected\_radio\_option* = "3";  
 }else{  
 *selected\_radio\_option* = "error";  
 }  
 MainActivity.*con*.state = "UPDATE\_TASK\_STATUS";  
 }  
 });  
 try {  
 setUpData();  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 mDisplayDate = findViewById(R.id.*tvDateStart2*);  
 mDisplayDate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Calendar cal = Calendar.*getInstance*();  
 int year = cal.get(Calendar.*YEAR*);  
 int month = cal.get(Calendar.*MONTH*);  
 int day = cal.get(Calendar.*DAY\_OF\_MONTH*);  
  
 DatePickerDialog dialog = new DatePickerDialog(  
 Task\_Page.this,  
 android.R.style.*Theme\_Holo\_Light\_Dialog\_MinWidth*,  
 mDateSetListener,  
 year,month,day);  
 dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*TRANSPARENT*));  
 dialog.show();  
 }  
 });  
  
 mDisplayDate2 = findViewById(R.id.*tvDateEnd2*);  
 mDisplayDate2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Calendar cal = Calendar.*getInstance*();  
 int year = cal.get(Calendar.*YEAR*);  
 int month = cal.get(Calendar.*MONTH*);  
 int day = cal.get(Calendar.*DAY\_OF\_MONTH*);  
  
 DatePickerDialog dialog = new DatePickerDialog(  
 Task\_Page.this,  
 android.R.style.*Theme\_Holo\_Light\_Dialog\_MinWidth*,  
 mDateSetListener2,  
 year,month,day);  
 dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*TRANSPARENT*));  
 dialog.show();  
 }  
 });  
  
 mDateSetListener = new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
 month++;  
 Log.*d*(*TAG*,"date: "+dayOfMonth+"/"+month+"/"+year);  
 String date = "Start: "+dayOfMonth+"/"+month+"/"+year;  
 mDisplayDate.setText(date);  
 }  
 };  
 mDateSetListener2 = new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
 month++;  
 Log.*d*(*TAG*,"date2: "+dayOfMonth+"/"+month+"/"+year);  
 String date = "End: "+dayOfMonth+"/"+month+"/"+year;  
 mDisplayDate2.setText(date);  
 }  
 };  
  
  
  
 *task\_name* = findViewById(R.id.*etTaskName2*);  
 *task\_des* = findViewById(R.id.*etTaskDes2*);  
 *task\_start* = findViewById(R.id.*tvDateStart2*);  
 *task\_end* = findViewById(R.id.*tvDateEnd2*);  
  
 Button SubmitMinorTask = findViewById(R.id.*btn\_SubmitMinorTask*);  
 SubmitMinorTask.setOnClickListener(new View.OnClickListener(){  
  
 @Override  
 public void onClick(View v) {  
 *select\_user* = null;  
 if(filteredUser == null){  
 *select\_user* = *User\_List\_obj*.get(0).getName();  
 }else if(filteredUser.isEmpty()){  
 *select\_user* = *User\_List\_obj*.get(0).getName();  
 }else if(!filteredUser.isEmpty()){  
 *select\_user* = filteredUser.get(0).getName();  
 }else{  
 //condition should never be used  
 *select\_user* = "No User Assigned to Task";  
 }  
 Log.*e*("task submit clicked : ", *task\_name*.getText().toString()+" "+*task\_des*.getText().toString()+" "+*task\_start*.getText().toString()+" "+*task\_end*.getText().toString()+" "+*select\_user*);  
 if(*task\_name*.getText().toString().isEmpty()){  
 Toast.*makeText*(Task\_Page.this,"Insert Task Name", Toast.*LENGTH\_LONG*).show();  
  
 }else if (*task\_start*.getText().toString().isEmpty()){  
 Toast.*makeText*(Task\_Page.this,"Insert Task Start Date", Toast.*LENGTH\_LONG*).show();  
 }else if (*task\_end*.getText().toString().isEmpty()){  
 Toast.*makeText*(Task\_Page.this,"Insert Task Start Date", Toast.*LENGTH\_LONG*).show();  
 }else{  
 MainActivity.*con*.state = "INSERT\_NEW\_MINOR\_TASK";  
  
 }  
  
 }  
 });  
  
 while(true){  
 if(*check1* == true){  
 break;  
 }  
 try {  
 TimeUnit.*MILLISECONDS*.sleep(100);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 *check1* = false;  
 String temp = "Task \n"+*e\_taskname*;  
 Log.*e*("YOUR\_APP\_LOG\_TAG", "task page "+temp+" - "+temp.length());  
  
 tv\_Title.setText(temp);  
 tv\_Description.setText("Description \n"+*Des*);  
 tv\_ResponsibleUser.setText("Assign User \n"+*Assigned\_user*);  
 //tv\_Title.setText("Task \n"+P\_overview.task\_name);  
  
 tv\_StartDate.setText("Task Planned Start Date \n"+*s\_date*);  
 tv\_EndDate.setText("Task Planned End Date \n"+*e\_date*);  
 if(*Status*.equals("1")){  
 tv\_Status.setText("Status \nIn Progress");  
 }else if(*Status*.equals("2")){  
 tv\_Status.setText("Status \nResolved");  
 }else if(*Status*.equals("3")){  
 tv\_Status.setText("Status \nCanceled");  
 }else{  
 tv\_Status.setText("Status \nError");  
 }  
  
 *Des* = null;  
 *Status* = null;  
 *Assigned\_user* = null;  
 *s\_date* = null;  
 *e\_date* = null;  
  
  
  
  
 }  
  
 public void checkButton(View v){  
 int radioId = radioGroup.getCheckedRadioButtonId();  
  
 radioButton = findViewById(radioId);  
 Toast.*makeText*(getApplicationContext(),"Selected Radio Button: " + radioButton.getText(),Toast.*LENGTH\_LONG*).show();  
 }  
  
 public void setUpData() throws InterruptedException {  
 if(*User\_List\_obj*.isEmpty()){  
 MainActivity.*con*.state = "GET\_ALL\_USERS\_IN\_PROJECT\_TASKPAGE";  
 }  
 }  
}

package com.example.app;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.navigation.NavController;  
import androidx.navigation.Navigation;  
import androidx.navigation.ui.NavigationUI;  
  
import android.os.Bundle;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
  
public class temp\_Home extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_home*);  
  
 BottomNavigationView bottomNavigationView = findViewById(R.id.*bottomNavigationView2*);  
 NavController navController = Navigation.*findNavController*(this, R.id.*fragment3*);  
 NavigationUI.*setupWithNavController*(bottomNavigationView, navController);  
  
 }  
}

package com.example.app;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
  
import java.util.List;  
  
*/\*\*  
 \* adapter for users recycler view in add users to project tab  
 \*/*public class User\_object\_adapter extends ArrayAdapter<User\_object\_add\_user> {  
 public User\_object\_adapter(@NonNull Context context, int resource, List<User\_object\_add\_user> userList) {  
 super(context, resource, userList);  
 }  
  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
  
 User\_object\_add\_user user = getItem(position);  
  
 if(convertView == null){  
 convertView = LayoutInflater.*from*(getContext()).inflate(R.layout.*user\_object\_add\_user\_cell*, parent,false);  
 }  
  
 TextView tv = convertView.findViewById(R.id.*user\_name*);  
  
 tv.setText(user.getName());  
  
 return convertView;  
 }  
}

package com.example.app;  
  
*/\*\*  
 \* class represent a user  
 \*/*public class User\_object\_add\_user {  
 private String Name;  
  
 public User\_object\_add\_user(String name) {  
 Name = name;  
 }  
  
 public String getName() {  
 return Name;  
 }  
  
 public void setName(String name) {  
 Name = name;  
 }  
}

#### Resources scripts

**border.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="@android:color/white"/>  
 <stroke android:width="1dip" android:color="#4fa5d5"/>  
  
</shape>

**border\_current.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="@android:color/white"/>  
 <stroke android:width="1dip" android:color="@android:color/holo\_purple"/>  
  
</shape>

**border\_current\_week\_complete.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="@android:color/holo\_green\_light"/>  
 <stroke android:width="1dip" android:color="@android:color/holo\_purple"/>  
  
</shape>

**border\_current\_with\_task.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="@android:color/holo\_blue\_dark"/>  
 <stroke android:width="1dip" android:color="@android:color/holo\_purple"/>  
  
</shape>

**border\_current\_without\_task.xml**

<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android">  
 <solid android:color="@android:color/holo\_red\_dark"/>  
 <stroke android:width="1dip" android:color="@android:color/holo\_purple"/>  
  
</shape>

**borderbottom.xml**

<?xml version="1.0" encoding="utf-8"?>  
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">  
 <item android:top="-2dp" android:left="-2dp" android:right="-2dp">  
 <shape android:shape="rectangle">  
 <stroke android:width="2dp" android:color="@color/black" />  
 <solid android:color="#00000000" />  
 </shape>  
 </item>  
</layer-list>

**ic\_baseline\_add\_task\_24.xml**

<vector android:height="24dp" android:tint="#556B2F"  
 android:viewportHeight="24" android:viewportWidth="24"  
 android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="@android:color/white" android:pathData="M22,5.18L10.59,16.6l-4.24,-4.24l1.41,-1.41l2.83,2.83l10,-10L22,5.18zM12,20c-4.41,0 -8,-3.59 -8,-8s3.59,-8 8,-8c1.57,0 3.04,0.46 4.28,1.25l1.45,-1.45C16.1,2.67 14.13,2 12,2C6.48,2 2,6.48 2,12s4.48,10 10,10c1.73,0 3.36,-0.44 4.78,-1.22l-1.5,-1.5C14.28,19.74 13.17,20 12,20zM19,15h-3v2h3v3h2v-3h3v-2h-3v-3h-2V15z"/>  
</vector>

**ic\_baseline\_create\_24.xml**

<vector android:height="24dp" android:tint="#556B2F"  
 android:viewportHeight="24" android:viewportWidth="24"  
 android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="@android:color/white" android:pathData="M3,17.25V21h3.75L17.81,9.94l-3.75,-3.75L3,17.25zM20.71,7.04c0.39,-0.39 0.39,-1.02 0,-1.41l-2.34,-2.34c-0.39,-0.39 -1.02,-0.39 -1.41,0l-1.83,1.83 3.75,3.75 1.83,-1.83z"/>  
</vector>

**ic\_baseline\_home\_24.xml**

<vector android:height="24dp" android:tint="#556B2F"  
 android:viewportHeight="24" android:viewportWidth="24"  
 android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="@android:color/white" android:pathData="M10,20v-6h4v6h5v-8h3L12,3 2,12h3v8z"/>  
</vector>

**ic\_baseline\_message\_24.xml**

<vector android:height="24dp" android:tint="#556B2F"  
 android:viewportHeight="24" android:viewportWidth="24"  
 android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="@android:color/white" android:pathData="M20,2L4,2c-1.1,0 -1.99,0.9 -1.99,2L2,22l4,-4h14c1.1,0 2,-0.9 2,-2L22,4c0,-1.1 -0.9,-2 -2,-2zM18,14L6,14v-2h12v2zM18,11L6,11L6,9h12v2zM18,8L6,8L6,6h12v2z"/>  
</vector>

**ic\_baseline\_note\_add\_24.xml**

<vector android:height="24dp" android:tint="#556B2F"  
 android:viewportHeight="24" android:viewportWidth="24"  
 android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="@android:color/white" android:pathData="M14,2L6,2c-1.1,0 -1.99,0.9 -1.99,2L4,20c0,1.1 0.89,2 1.99,2L18,22c1.1,0 2,-0.9 2,-2L20,8l-6,-6zM16,16h-3v3h-2v-3L8,16v-2h3v-3h2v3h3v2zM13,9L13,3.5L18.5,9L13,9z"/>  
</vector>

**ic\_baseline\_person\_add\_24.xml**

<vector android:height="24dp" android:tint="#556B2F"  
 android:viewportHeight="24" android:viewportWidth="24"  
 android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="@android:color/white" android:pathData="M15,12c2.21,0 4,-1.79 4,-4s-1.79,-4 -4,-4 -4,1.79 -4,4 1.79,4 4,4zM6,10L6,7L4,7v3L1,10v2h3v3h2v-3h3v-2L6,10zM15,14c-2.67,0 -8,1.34 -8,4v2h16v-2c0,-2.66 -5.33,-4 -8,-4z"/>  
</vector>

**ic\_baseline\_search\_24.xml**

<vector android:height="24dp" android:tint="#556B2F"  
 android:viewportHeight="24" android:viewportWidth="24"  
 android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="@android:color/white" android:pathData="M15.5,14h-0.79l-0.28,-0.27C15.41,12.59 16,11.11 16,9.5 16,5.91 13.09,3 9.5,3S3,5.91 3,9.5 5.91,16 9.5,16c1.61,0 3.09,-0.59 4.23,-1.57l0.27,0.28v0.79l5,4.99L20.49,19l-4.99,-5zM9.5,14C7.01,14 5,11.99 5,9.5S7.01,5 9.5,5 14,7.01 14,9.5 11.99,14 9.5,14z"/>  
</vector>

**ic\_baseline\_table\_rows\_24.xml**

<vector android:height="24dp" android:tint="#556B2F"  
 android:viewportHeight="24" android:viewportWidth="24"  
 android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">  
 <path android:fillColor="@android:color/white" android:pathData="M22,7H2V2h20V7zM22,9.5H2v5h20V9.5zM22,17H2v5h20V17z"/>  
</vector>

**ic\_launcher\_background.xml**

<?xml version="1.0" encoding="utf-8"?>  
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
 android:width="108dp"  
 android:height="108dp"  
 android:viewportWidth="108"  
 android:viewportHeight="108">  
 <path  
 android:fillColor="#3DDC84"  
 android:pathData="M0,0h108v108h-108z" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M9,0L9,108"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,0L19,108"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M29,0L29,108"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M39,0L39,108"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M49,0L49,108"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M59,0L59,108"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M69,0L69,108"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M79,0L79,108"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M89,0L89,108"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M99,0L99,108"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,9L108,9"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,19L108,19"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,29L108,29"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,39L108,39"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,49L108,49"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,59L108,59"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,69L108,69"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,79L108,79"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,89L108,89"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M0,99L108,99"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,29L89,29"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,39L89,39"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,49L89,49"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,59L89,59"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,69L89,69"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M19,79L89,79"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M29,19L29,89"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M39,19L39,89"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M49,19L49,89"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M59,19L59,89"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M69,19L69,89"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
 <path  
 android:fillColor="#00000000"  
 android:pathData="M79,19L79,89"  
 android:strokeWidth="0.8"  
 android:strokeColor="#33FFFFFF" />  
</vector>

**ic\_launcher\_foreground.xml**

<vector xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:aapt="http://schemas.android.com/aapt"  
 android:width="108dp"  
 android:height="108dp"  
 android:viewportWidth="108"  
 android:viewportHeight="108">  
 <path android:pathData="M31,63.928c0,0 6.4,-11 12.1,-13.1c7.2,-2.6 26,-1.4 26,-1.4l38.1,38.1L107,108.928l-32,-1L31,63.928z">  
 <aapt:attr name="android:fillColor">  
 <gradient  
 android:endX="85.84757"  
 android:endY="92.4963"  
 android:startX="42.9492"  
 android:startY="49.59793"  
 android:type="linear">  
 <item  
 android:color="#44000000"  
 android:offset="0.0" />  
 <item  
 android:color="#00000000"  
 android:offset="1.0" />  
 </gradient>  
 </aapt:attr>  
 </path>  
 <path  
 android:fillColor="#FFFFFF"  
 android:fillType="nonZero"  
 android:pathData="M65.3,45.828l3.8,-6.6c0.2,-0.4 0.1,-0.9 -0.3,-1.1c-0.4,-0.2 -0.9,-0.1 -1.1,0.3l-3.9,6.7c-6.3,-2.8 -13.4,-2.8 -19.7,0l-3.9,-6.7c-0.2,-0.4 -0.7,-0.5 -1.1,-0.3C38.8,38.328 38.7,38.828 38.9,39.228l3.8,6.6C36.2,49.428 31.7,56.028 31,63.928h46C76.3,56.028 71.8,49.428 65.3,45.828zM43.4,57.328c-0.8,0 -1.5,-0.5 -1.8,-1.2c-0.3,-0.7 -0.1,-1.5 0.4,-2.1c0.5,-0.5 1.4,-0.7 2.1,-0.4c0.7,0.3 1.2,1 1.2,1.8C45.3,56.528 44.5,57.328 43.4,57.328L43.4,57.328zM64.6,57.328c-0.8,0 -1.5,-0.5 -1.8,-1.2s-0.1,-1.5 0.4,-2.1c0.5,-0.5 1.4,-0.7 2.1,-0.4c0.7,0.3 1.2,1 1.2,1.8C66.5,56.528 65.6,57.328 64.6,57.328L64.6,57.328z"  
 android:strokeWidth="1"  
 android:strokeColor="#00000000" />  
</vector>

**activity\_home.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/white">  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:id="@+id/bottomNavigationView2"  
 app:labelVisibilityMode="labeled"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/black"  
 app:itemTextColor="@color/purple\_700"  
 app:itemIconTint="@color/purple\_700"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:menu="@menu/bottom\_menu"/>  
  
 <fragment  
 android:id="@+id/fragment3"  
 android:name="androidx.navigation.fragment.NavHostFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/white"  
 android:layout\_marginBottom="25sp"  
 android:layout\_marginEnd="30sp"  
 app:defaultNavHost="true"  
 app:layout\_constraintBottom\_toTopOf="@+id/bottomNavigationView2"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="1.0"  
 app:navGraph="@navigation/nav\_home" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <EditText  
 android:id="@+id/et\_UserName"  
 android:layout\_width="243dp"  
 android:layout\_height="44dp"  
 android:ems="10"  
 android:hint="Enter UserName"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.17" />  
  
 <EditText  
 android:id="@+id/et\_Password"  
 android:layout\_width="248dp"  
 android:layout\_height="49dp"  
 android:ems="10"  
 android:hint="Enter Password"  
 android:inputType="textPassword"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/et\_UserName"  
 app:layout\_constraintVertical\_bias="0.03" />  
  
 <Button  
 android:id="@+id/btn\_login"  
 android:layout\_width="182dp"  
 android:layout\_height="47dp"  
 android:text="Login"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/et\_Password"  
 app:layout\_constraintVertical\_bias="0.071" />  
  
 <Button  
 android:id="@+id/btn\_registration"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Register New User"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btn\_login"  
 app:layout\_constraintVertical\_bias="0.068" />  
  
 <TextView  
 android:id="@+id/tv\_login\_response"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Number of ttempts remaining: 5 "  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btn\_registration"  
 app:layout\_constraintVertical\_bias="0.099" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_project.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:id="@+id/bnv\_project"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/black"  
 app:itemIconTint="@color/purple\_700"  
 app:itemTextColor="@color/purple\_700"  
 app:labelVisibilityMode="labeled"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:menu="@menu/bottom\_menu\_project" />  
  
  
 <fragment  
 android:id="@+id/fragment\_projects\_main"  
 android:name="androidx.navigation.fragment.NavHostFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:itemTextColor="@color/black"  
 app:defaultNavHost="true"  
 app:layout\_constraintBottom\_toTopOf="@+id/bnv\_project"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:navGraph="@navigation/nav\_project"  
 android:layout\_marginBottom="50dp"/>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_registration.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".act\_registration\_Main"  
 >  
  
 <EditText  
 android:id="@+id/et\_newPass"  
 android:layout\_width="248dp"  
 android:layout\_height="49dp"  
 android:layout\_marginBottom="468dp"  
 android:ems="10"  
 android:hint="Enter Password"  
 android:inputType="textPassword"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.496"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/et\_newUser"  
 app:layout\_constraintVertical\_bias="0.472" />  
  
 <EditText  
 android:id="@+id/et\_newPass2"  
 android:layout\_width="248dp"  
 android:layout\_height="49dp"  
 android:ems="10"  
 android:hint="ReEnter Password"  
 android:inputType="textPassword"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.496"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/et\_newPass"  
 app:layout\_constraintVertical\_bias="0.048" />  
  
 <EditText  
 android:id="@+id/et\_newUser"  
 android:layout\_width="244dp"  
 android:layout\_height="46dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:hint="enter PMkey "  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.141" />  
  
 <Button  
 android:id="@+id/btn\_register"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Register New User"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/et\_newPass2"  
 app:layout\_constraintVertical\_bias="0.072" />  
  
 <Button  
 android:id="@+id/btn\_back\_act\_reg"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Back"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.049"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.023" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_task\_info.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:layout\_editor\_absoluteX="0dp"  
 tools:layout\_editor\_absoluteY="-82dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/TaskTitle\_taskinfopage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginTop="36dp"  
 android:layout\_marginRight="18dp"  
 android:text="Task \nA Name of a Task "  
 android:textColor="@color/black"  
 android:textSize="38dp" />  
  
 <TextView  
 android:id="@+id/TaskDes"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginTop="36dp"  
 android:layout\_marginRight="18dp"  
 android:text="Description \nA Name of a Task "  
 android:textColor="@color/black"  
 android:textSize="28dp" />  
  
 <TextView  
 android:id="@+id/TaskStatus"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginTop="36dp"  
 android:layout\_marginRight="18dp"  
 android:text="Status \nA Name of a Task "  
 android:textColor="@color/black"  
 android:textSize="28dp" />  
  
 <TextView  
 android:id="@+id/TaskResponsibleUser"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginTop="36dp"  
 android:layout\_marginRight="18dp"  
 android:text="Assign User \nA Name of a Task "  
 android:textColor="@color/black"  
 android:textSize="28dp" />  
  
 <TextView  
 android:id="@+id/TaskStartDate"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginTop="36dp"  
 android:layout\_marginRight="18dp"  
 android:text="Task Planned Start Date \nA Name of a Task "  
 android:textColor="@color/black"  
 android:textSize="28dp" />  
  
 <TextView  
 android:id="@+id/TaskEndDate"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginTop="36dp"  
 android:layout\_marginRight="18dp"  
 android:background="@drawable/borderbottom"  
 android:text="Task Planned End Date \nA Name of a Task "  
 android:textColor="@color/black"  
 android:textSize="28dp" />  
  
 <Button  
 android:id="@+id/btn\_ViewMinorTasks"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="View Minor Tasks"  
 android:textSize="30dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginBottom="16dp"/>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginRight="18dp"  
 android:background="@drawable/borderbottom"  
 android:textColor="@color/black" />  
  
 <TextView  
 android:id="@+id/TaskStatusUpdate"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginTop="36dp"  
 android:layout\_marginRight="18dp"  
 android:text="Update Status"  
 android:textColor="@color/black"  
 android:textSize="38dp" />  
  
 <RadioGroup  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:id="@+id/radioGroupStatusUpdate">  
  
 <androidx.appcompat.widget.AppCompatRadioButton  
 android:id="@+id/radio\_in\_progress"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="15dp"  
 android:text="in progress"  
 android:textSize="28dp" />  
  
 <androidx.appcompat.widget.AppCompatRadioButton  
 android:id="@+id/radio\_Resolved"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="15dp"  
 android:text="Resolved"  
 android:textSize="28dp" />  
  
 <androidx.appcompat.widget.AppCompatRadioButton  
 android:id="@+id/radio\_Cancelled"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="15dp"  
 android:text="Cancelled"  
 android:textSize="28dp" />  
 </RadioGroup>  
  
 <Button  
 android:id="@+id/btn\_change\_task\_status"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginRight="18dp"  
 android:text="change task status"  
 android:textSize="28dp">  
  
 </Button>  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginRight="18dp"  
 android:background="@drawable/borderbottom"  
 android:textColor="@color/black" />  
  
  
 <TextView  
 android:id="@+id/CreateMinorTask"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="18dp"  
 android:layout\_marginTop="36dp"  
 android:layout\_marginBottom="36dp"  
 android:layout\_marginRight="18dp"  
 android:text="Create Minor Task"  
 android:textColor="@color/black"  
 android:textSize="38dp" />  
  
 <EditText  
 android:id="@+id/etTaskName2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="88dp"  
 android:ems="10"  
 android:hint="Insert New Task Name"  
 android:inputType="textPersonName" />  
  
 <EditText  
 android:id="@+id/etTaskDes2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="326dp"  
 android:ems="10"  
 android:hint="Insert Description"  
 android:inputType="textPersonName" />  
  
 <TextView  
 android:id="@+id/tvDateStart2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="25dp"  
 android:layout\_marginBottom="20dp"  
 android:hint="Select Task Start Date"  
 android:textAlignment="center"  
 android:textSize="30dp"  
 tools:ignore="MissingConstraints" />  
  
 <TextView  
 android:id="@+id/tvDateEnd2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="25dp"  
 android:layout\_marginBottom="20dp"  
 android:hint="Select Task End Date"  
 android:textAlignment="center"  
 android:textSize="30dp"  
 tools:ignore="MissingConstraints" />  
  
 <androidx.core.widget.NestedScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fillViewport="true">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical" >  
  
 <androidx.appcompat.widget.SearchView  
 android:id="@+id/userListSearchView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:iconifiedByDefault="false"  
 android:queryHint="Search Users in Project"  
 android:textColor="@color/black"  
 android:textSize="20dp"  
 android:textAlignment="center"  
 app:queryHint="Search Users in Project"  
 app:iconifiedByDefault="false">  
 </androidx.appcompat.widget.SearchView>  
 <TextView  
 android:id="@+id/textView5"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Selected User of Task below"  
 android:textColor="@color/black"  
 android:textSize="20dp"  
 android:textAlignment="center"/>  
 <ListView  
 android:id="@+id/add\_user\_list\_view2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 </LinearLayout>  
  
 </androidx.core.widget.NestedScrollView>  
  
 <Button  
 android:id="@+id/btn\_SubmitMinorTask"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Submit Minor Task"  
 android:textSize="30dp"/>  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="80dp"  
 android:text="" />  
  
  
 </LinearLayout>  
  
  
  
  
 </ScrollView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_temp\_\_home.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".temp\_Home">  
  
 <TextView  
 android:id="@+id/tv\_tempHome"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Welcome"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**card\_view\_item.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.cardview.widget.CardView  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 app:cardCornerRadius="0dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <com.google.android.material.circularreveal.CircularRevealRelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/card\_item"  
 android:orientation="vertical"  
 />  
  
  
  
</androidx.cardview.widget.CardView>

**fragment\_create\_team.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".CreateTeamFragment">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="254dp"  
 android:layout\_height="83dp"  
 android:layout\_marginTop="36dp"  
 android:text="Create New Project"  
 android:textColor="@color/black"  
 android:textSize="30dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.496"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:ignore="MissingConstraints" />  
  
 <EditText  
 android:id="@+id/et\_new\_project\_name"  
 android:layout\_width="345dp"  
 android:layout\_height="46dp"  
 android:layout\_marginTop="24dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:hint="Enter New Project Name"  
 android:textColor="@color/black"  
 android:textColorHint="@color/black"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.469"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <EditText  
 android:id="@+id/et\_new\_description"  
 android:layout\_width="345dp"  
 android:layout\_height="159dp"  
 android:ems="8"  
 android:gravity="start|top"  
 android:inputType="textMultiLine"  
 android:hint="optional: description of Project"  
 android:textColor="@color/black"  
 android:textColorHint="@color/black"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.461"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/et\_new\_project\_name"  
 app:layout\_constraintVertical\_bias="0.06" />  
  
 <Button  
 android:id="@+id/btn\_create\_project"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Create "  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/et\_new\_description"  
 app:layout\_constraintVertical\_bias="0.089" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**fragment\_diary.xml**

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".DiaryFragment">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:text="@string/hello\_blank\_fragment" />  
  
</FrameLayout>

**fragment\_home.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".HomeFragment">  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/listRev\_1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**fragment\_messagees.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <androidx.appcompat.widget.SearchView  
 android:id="@+id/userListSearchView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:iconifiedByDefault="false"  
 android:queryHint="Search Users">  
  
 </androidx.appcompat.widget.SearchView>  
  
 <ListView  
 android:id="@+id/add\_user\_list\_view\_msg"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
  
  
  
  
</LinearLayout>

**fragment\_p\_add\_task.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".P\_addTask"  
 >  
  
 <!-- *TODO: Update blank fragment layout* -->  
  
 <ScrollView  
 android:id="@+id/scrollView2"  
 android:layout\_width="400dp"  
 android:layout\_height="622dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:fillViewport="true">  
  
 <LinearLayout  
 android:id="@+id/Linearlayout\_1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="97dp"  
 android:layout\_marginTop="36dp"  
 android:text="Create New Task"  
 android:textColor="@color/black"  
 android:textSize="40dp"  
 tools:ignore="MissingConstraints" />  
  
 <EditText  
 android:id="@+id/etTaskName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="88dp"  
 android:ems="10"  
 android:hint="Insert New Task Name"  
 android:inputType="textPersonName" />  
  
 <EditText  
 android:id="@+id/etTaskDes"  
 android:layout\_width="match\_parent"  
 android:layout\_height="326dp"  
 android:ems="10"  
 android:hint="Insert Description"  
 android:inputType="textPersonName" />  
  
 <TextView  
 android:id="@+id/tvDateStart"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="25dp"  
 android:layout\_marginBottom="20dp"  
 android:hint="Select Task Start Date"  
 android:textAlignment="center"  
 android:textSize="30dp"  
 tools:ignore="MissingConstraints" />  
  
  
 <TextView  
 android:id="@+id/tvDateEnd"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="25dp"  
 android:layout\_marginBottom="20dp"  
 android:hint="Select Task End Date"  
 android:textAlignment="center"  
 android:textSize="30dp"  
 tools:ignore="MissingConstraints" />  
  
 <androidx.core.widget.NestedScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fillViewport="true">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical" >  
  
 <androidx.appcompat.widget.SearchView  
 android:id="@+id/userListSearchView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:iconifiedByDefault="false"  
 android:queryHint="Search Users in Project"  
 android:textColor="@color/black"  
 android:textSize="20dp"  
 android:textAlignment="center"  
 app:queryHint="Search Users in Project"  
 app:iconifiedByDefault="false">  
 </androidx.appcompat.widget.SearchView>  
 <TextView  
 android:id="@+id/textView5"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Selected User of Task below"  
 android:textColor="@color/black"  
 android:textSize="20dp"  
 android:textAlignment="center"/>  
 <ListView  
 android:id="@+id/add\_user\_list\_view2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 </LinearLayout>  
  
 </androidx.core.widget.NestedScrollView>  
  
 <Button  
 android:id="@+id/btn\_SubmitTask"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Submit Task to Project"  
 android:textSize="30dp"/>  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="80dp"  
 android:text="" />  
  
 </LinearLayout>  
 </ScrollView>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**fragment\_p\_add\_user.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".P\_addUser"  
 android:orientation="vertical">  
  
 <androidx.appcompat.widget.SearchView  
 android:id="@+id/userListSearchView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:iconifiedByDefault="false"  
 android:queryHint="Search Users"  
 app:queryHint="Search Users">  
  
 </androidx.appcompat.widget.SearchView>  
  
 <ListView  
 android:id="@+id/add\_user\_list\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
  
  
  
  
</LinearLayout>

**fragment\_p\_home.xml**

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/P\_Home"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="P\_Home">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <TextView  
 android:id="@+id/p\_home"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:text="home projects page" />  
  
</FrameLayout>

**fragment\_p\_overview.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".P\_overview">  
  
 <Button  
 android:id="@+id/overview\_title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:text="Time Line - Week"  
 android:textColor="@color/black"  
 android:textSize="30dp" />  
  
 <com.inqbarna.tablefixheaders.TableFixHeaders  
 android:id="@+id/tablefixheaders"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/black"  
 app:itemTextColor="@color/black" />  
  
  
</LinearLayout>

**gantt\_view\_item.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
  
 android:background="@drawable/border"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/txt\_content"  
 android:layout\_centerInParent="true"  
 android:gravity="center"/>  
  
</RelativeLayout>

**list\_items\_home\_frag.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TextView  
 android:id="@+id/P\_Name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="40dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginEnd="40dp"  
 android:layout\_marginBottom="35dp"  
 android:gravity="center\_vertical|left|start"  
 android:text="Project "  
 android:textAlignment="gravity"  
 android:textColor="@color/black"  
 android:textSize="36sp"  
 android:background="@color/white"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.03"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.0" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**user\_object\_add\_user\_cell.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="horizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/purple\_200">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/user\_name"  
 android:gravity="center\_vertical"  
 android:text ="User Name"  
 android:textColor="@color/black"  
 android:textSize="40dp"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginBottom="25dp"/>  
  
</RelativeLayout>

**bottom\_menu.xml**

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item android:title="Projects"  
 android:icon="@drawable/ic\_baseline\_home\_24"  
 android:id="@+id/nav\_home"/>  
 <item android:title="Create"  
 android:icon="@drawable/ic\_baseline\_create\_24"  
 android:id="@+id/createTeamFragment"/>  
 <item android:title="MSG"  
 android:icon="@drawable/ic\_baseline\_message\_24"  
 android:id="@+id/messageesFragment"/>  
 <item  
 android:id="@+id/diaryFragment"  
 android:icon="@drawable/ic\_baseline\_note\_add\_24"  
 android:title="Diary" />  
</menu>

**bottom\_menu\_project.xml**

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:android="http://schemas.android.com/apk/res/android">  
  
 <item android:title="home"  
 android:icon="@drawable/ic\_baseline\_home\_24"  
 android:id="@+id/p\_Home"/>  
 <item android:title="overview" android:icon="@drawable/ic\_baseline\_table\_rows\_24"  
 android:id="@+id/p\_overview"/>  
 <item android:title="add User" android:icon="@drawable/ic\_baseline\_person\_add\_24"  
 android:id="@+id/p\_addUser"/>  
 <item android:title="add Task" android:icon="@drawable/ic\_baseline\_add\_task\_24"  
 android:id="@+id/p\_addTask"/>  
</menu>

**nav\_menu.xml**

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
 <item  
 android:title="Search"  
 android:id="@+id/search\_icon"  
 android:icon="@drawable/ic\_baseline\_search\_24"  
 app:showAsAction="ifRoom|collapseActionView"  
 app:actionViewClass="android.support.v7.widget.SearchView"  
 />  
  
</menu>

**nav\_home.xml**

<?xml version="1.0" encoding="utf-8"?>  
<navigation xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/nav\_home"  
 app:startDestination="@id/homeFragment">  
 <fragment  
 android:id="@+id/homeFragment"  
 android:name="com.example.app.HomeFragment"  
 android:label="fragment\_home"  
 tools:layout="@layout/fragment\_home" />  
 <fragment  
 android:id="@+id/createTeamFragment"  
 android:name="com.example.app.CreateTeamFragment"  
 android:label="fragment\_create\_team"  
 tools:layout="@layout/fragment\_create\_team" />  
 <fragment  
 android:id="@+id/messageesFragment"  
 android:name="com.example.app.MessageesFragment"  
 android:label="fragment\_messagees"  
 tools:layout="@layout/fragment\_messagees" />  
 <fragment  
 android:id="@+id/diaryFragment"  
 android:name="com.example.app.DiaryFragment"  
 android:label="fragment\_diary"  
 tools:layout="@layout/fragment\_diary" />  
</navigation>

**nav\_project.xml**

<?xml version="1.0" encoding="utf-8"?>  
<navigation xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/nav\_project"  
 app:startDestination="@id/p\_overview">  
  
 <fragment  
 android:id="@+id/p\_Home"  
 android:name="com.example.app.P\_Home"  
 android:label="fragment\_p\_\_home"  
 tools:layout="@layout/fragment\_p\_home" />  
 <fragment  
 android:id="@+id/p\_overview"  
 android:name="com.example.app.P\_overview"  
 android:label="fragment\_p\_overview"  
 tools:layout="@layout/fragment\_p\_overview" />  
 <fragment  
 android:id="@+id/p\_addUser"  
 android:name="com.example.app.P\_addUser"  
 android:label="fragment\_p\_add\_user"  
 tools:layout="@layout/fragment\_p\_add\_user" />  
 <fragment  
 android:id="@+id/p\_addTask"  
 android:name="com.example.app.P\_addTask"  
 android:label="fragment\_p\_add\_task"  
 tools:layout="@layout/fragment\_p\_add\_task" />  
</navigation>

**nav\_project\_minor\_task.xml**

<?xml version="1.0" encoding="utf-8"?>  
<navigation  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/nav\_project"  
 app:startDestination="@id/p\_overview">  
  
 <fragment  
 android:id="@+id/p\_Home"  
 android:name="com.example.app.P\_Home"  
 android:label="fragment\_p\_\_home"  
 tools:layout="@layout/fragment\_p\_home" />  
 <fragment  
 android:id="@+id/p\_overview"  
 android:name="com.example.app.P\_overview"  
 android:label="fragment\_p\_overview"  
 tools:layout="@layout/fragment\_p\_overview" />  
 <fragment  
 android:id="@+id/p\_addUser"  
 android:name="com.example.app.P\_addUser"  
 android:label="fragment\_p\_add\_user"  
 tools:layout="@layout/fragment\_p\_add\_user" />  
 <fragment  
 android:id="@+id/p\_addTask"  
 android:name="com.example.app.P\_addTask"  
 android:label="fragment\_p\_add\_task"  
 tools:layout="@layout/fragment\_p\_add\_task" />  
  
</navigation>

#### Gradle scripts

plugins **{** id 'com.android.application'  
**}**android **{** compileSdkVersion 30  
 buildToolsVersion "30.0.3"  
  
 defaultConfig **{** applicationId "com.example.app"  
 minSdkVersion 21  
 targetSdkVersion 30  
 versionCode 1  
 versionName "1.0"  
  
 testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
 **}** buildTypes **{** release **{** minifyEnabled false  
 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
 **}  
 }** compileOptions **{** sourceCompatibility JavaVersion.*VERSION\_1\_8* targetCompatibility JavaVersion.*VERSION\_1\_8* **}  
}**dependencies **{** implementation 'androidx.appcompat:appcompat:1.2.0'  
 implementation 'com.google.android.material:material:1.3.0'  
 implementation 'androidx.constraintlayout:constraintlayout:2.0.4'  
 implementation 'androidx.navigation:navigation-fragment:2.3.5'  
 implementation 'androidx.navigation:navigation-ui:2.3.5'  
 implementation 'androidx.legacy:legacy-support-v4:1.0.0'  
 testImplementation 'junit:junit:4.+'  
 androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
 androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
  
 implementation "androidx.recyclerview:recyclerview-selection:1.1.0"  
 implementation "androidx.recyclerview:recyclerview:1.2.0"  
  
 //for gnatt  
 implementation "com.github.miguelbcr:TableFixHeaders-Wrapper:0.2.0"  
 implementation 'androidx.cardview:cardview:1.0.0'  
  
**}**

// Top-level build file where you can add configuration options common to all sub-projects/modules.  
buildscript **{** repositories **{** google()  
 jcenter()  
 **}** dependencies **{** classpath "com.android.tools.build:gradle:4.0.2"  
  
 // NOTE: Do not place your application dependencies here; they belong  
 // in the individual module build.gradle files  
 **}  
}**allprojects **{** repositories **{** google()  
 jcenter()  
 maven**{** url "https://jitpack.io"**}  
 }  
}**task clean(type: Delete) **{** delete rootProject.buildDir  
**}**

## Creation scripts

Database creation script follows – dbgen.sql

CREATE TABLE `people\_per\_project` (

`Project` VARCHAR(50) NOT NULL COLLATE 'latin1\_swedish\_ci',

`User` CHAR(50) NOT NULL DEFAULT '' COLLATE 'latin1\_swedish\_ci',

`Role` INT(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`Project`, `User`) USING BTREE,

INDEX `FK\_\_projects` (`User`) USING BTREE,

CONSTRAINT `FK\_\_projects\_2` FOREIGN KEY (`Project`) REFERENCES `egh400\_test`.`projects` (`Project\_Name`) ON UPDATE RESTRICT ON DELETE RESTRICT

)

COLLATE='latin1\_swedish\_ci'

ENGINE=InnoDB

;

CREATE TABLE `projects` (

`Project\_Name` VARCHAR(50) NOT NULL COLLATE 'latin1\_swedish\_ci',

`Created\_by` CHAR(50) NOT NULL COLLATE 'latin1\_swedish\_ci',

`Description` TEXT(65535) NOT NULL DEFAULT '' COLLATE 'latin1\_swedish\_ci',

`Project\_Creation\_date` DATE NULL DEFAULT NULL,

PRIMARY KEY (`Project\_Name`) USING BTREE,

INDEX `users` (`Created\_by`) USING BTREE,

CONSTRAINT `users` FOREIGN KEY (`Created\_by`) REFERENCES `egh400\_test`.`users` (`ID`) ON UPDATE RESTRICT ON DELETE RESTRICT

)

COLLATE='latin1\_swedish\_ci'

ENGINE=InnoDB

;

CREATE TABLE `tasks` (

`Task\_name` VARCHAR(100) NOT NULL COLLATE 'latin1\_swedish\_ci',

`Project` VARCHAR(50) NOT NULL COLLATE 'latin1\_swedish\_ci',

`Assigned\_User` CHAR(50) NOT NULL COLLATE 'latin1\_swedish\_ci',

`Created\_By` CHAR(50) NOT NULL DEFAULT '' COLLATE 'latin1\_swedish\_ci',

`Status\_int` INT(11) NULL DEFAULT NULL,

`start\_date` DATE NULL DEFAULT NULL,

`end\_date` DATE NULL DEFAULT NULL,

`Description` TEXT(65535) NULL DEFAULT NULL COLLATE 'latin1\_swedish\_ci',

PRIMARY KEY (`Task\_name`, `Project`) USING BTREE,

INDEX `FK\_\_people\_per\_project\_2` (`Project`) USING BTREE,

CONSTRAINT `FK\_\_people\_per\_project\_2` FOREIGN KEY (`Project`) REFERENCES `egh400\_test`.`people\_per\_project` (`Project`) ON UPDATE RESTRICT ON DELETE RESTRICT

)

COLLATE='latin1\_swedish\_ci'

ENGINE=InnoDB

;

CREATE TABLE `users` (

`ID` CHAR(50) NOT NULL COLLATE 'latin1\_swedish\_ci',

`salt` VARCHAR(385) NULL DEFAULT NULL COLLATE 'latin1\_swedish\_ci',

`Password` VARCHAR(385) NULL DEFAULT NULL COLLATE 'latin1\_swedish\_ci',

PRIMARY KEY (`ID`) USING BTREE

)

COLLATE='latin1\_swedish\_ci'

ENGINE=InnoDB

;

CREATE TABLE `minortasks` (

`root\_task` VARCHAR(100) NOT NULL COLLATE 'latin1\_swedish\_ci',

`Task\_name` VARCHAR(100) NOT NULL COLLATE 'latin1\_swedish\_ci',

`Project` VARCHAR(50) NOT NULL COLLATE 'latin1\_swedish\_ci',

`Assigned\_User` CHAR(50) NOT NULL COLLATE 'latin1\_swedish\_ci',

`Created\_By` CHAR(50) NOT NULL DEFAULT '' COLLATE 'latin1\_swedish\_ci',

`Status\_int` INT(11) NULL DEFAULT NULL,

`start\_date` DATE NULL DEFAULT NULL,

`end\_date` DATE NULL DEFAULT NULL,

`Description` TEXT(65535) NULL DEFAULT NULL COLLATE 'latin1\_swedish\_ci',

PRIMARY KEY (`root\_task`, `Task\_name`, `Project`) USING BTREE

)

COLLATE='latin1\_swedish\_ci'

ENGINE=InnoDB

;