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

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.

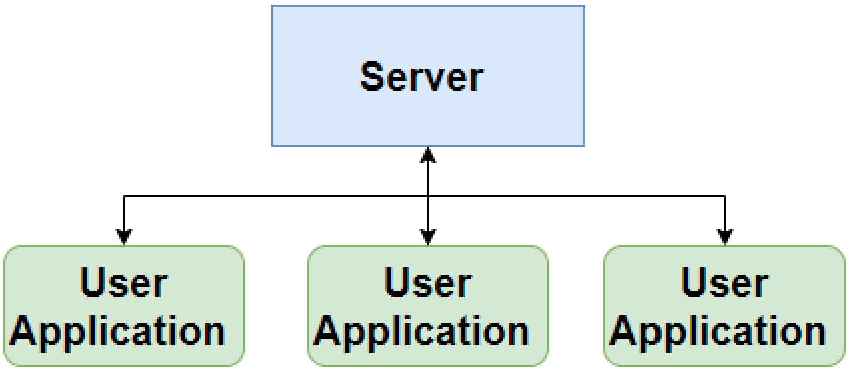


Figure 1: Client-Server Architecture

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.

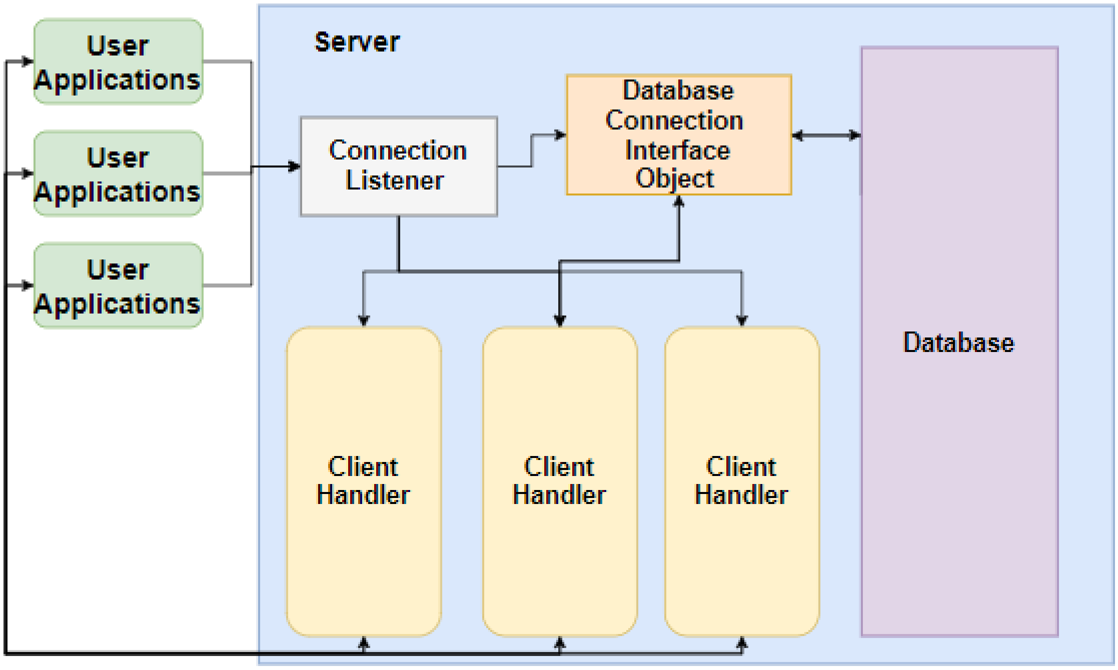


Figure 2: Server Architecture

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.

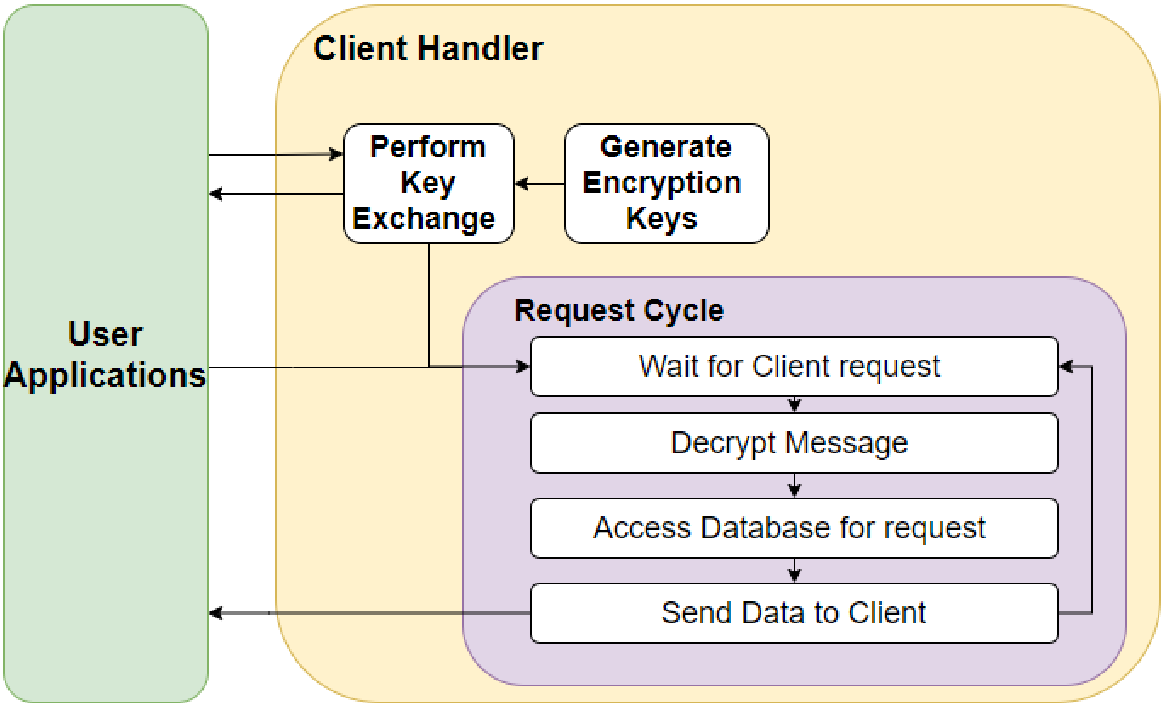


Figure 3: Client Handler Architecture

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.

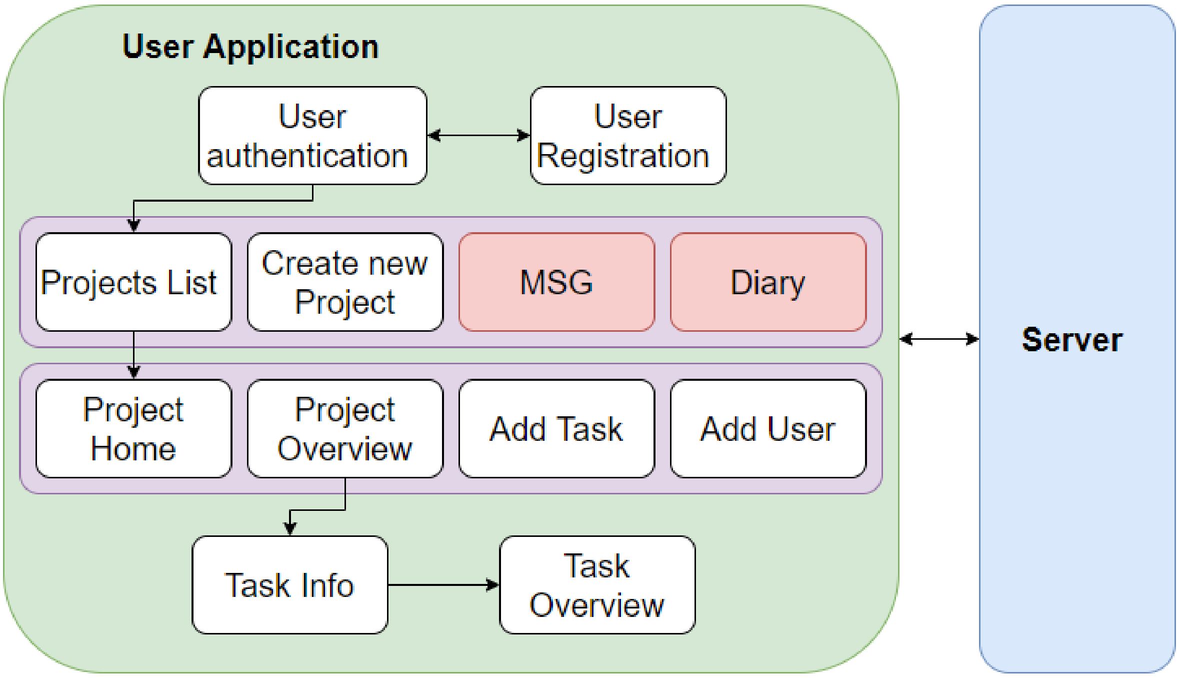


Figure 4: User application diagram

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.

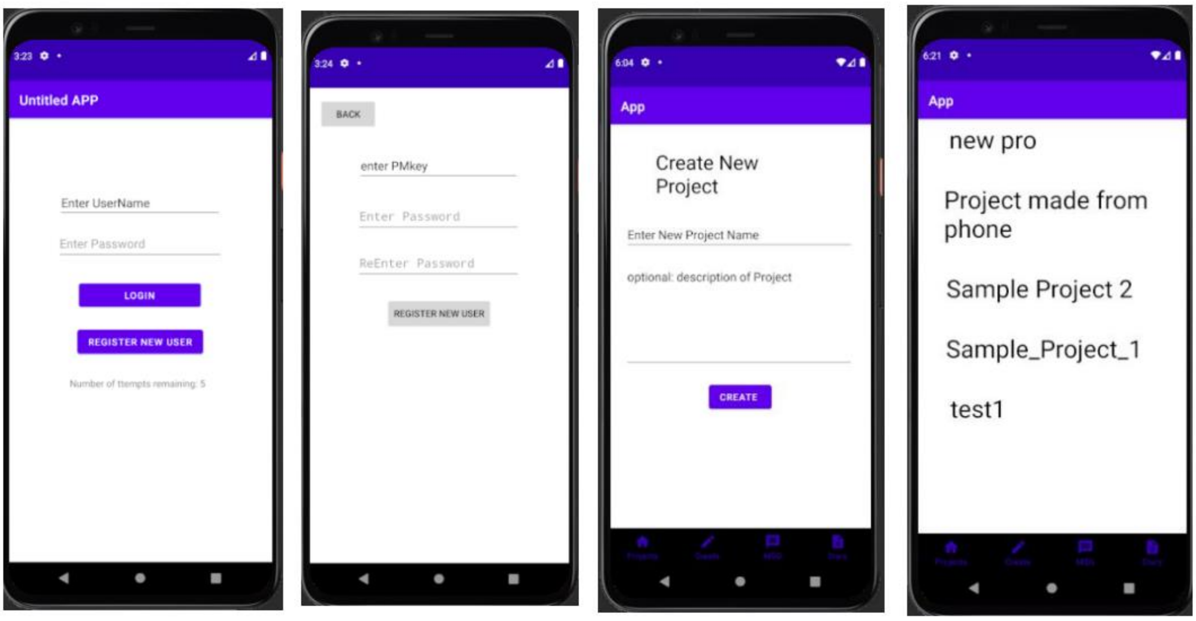


Figure 5: From left to right - Login Page Figure, Registration Page, Create New Project Page, Projects Page

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.

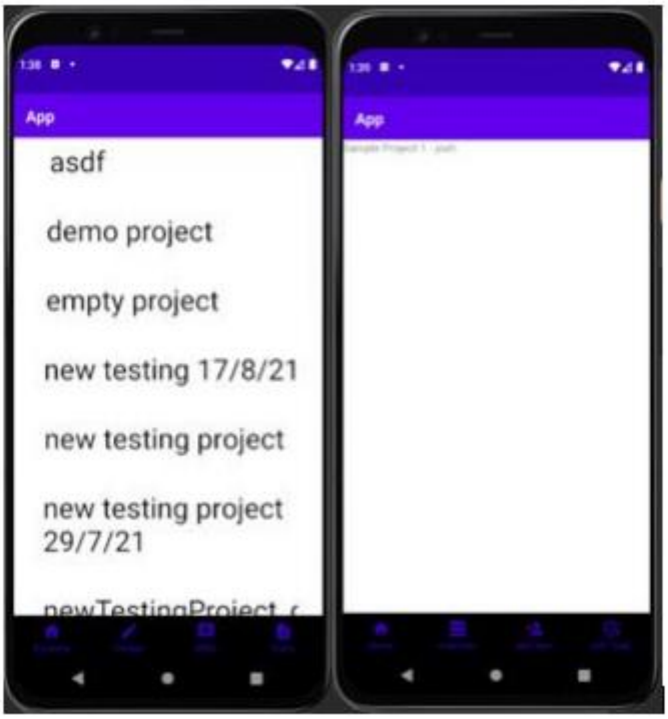
Once the database was redesigned a way to allow users to store information was needed. Under the create new project tab from the bottom navigation option a create new projects page was created seen in figure 5. This page functioned like the registration page as it would send information to the server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



asdf

demo project

empty project

new testing 17/8/21

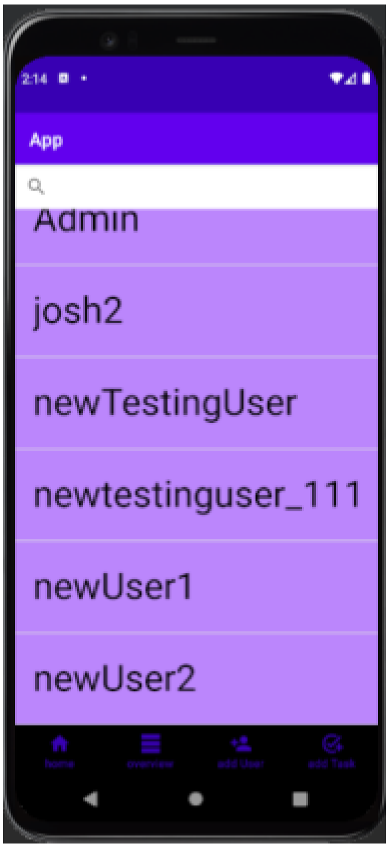
new testing project

new testing project 29/7/21

newTestingProject_c

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.



Admin

josh2

newTestingUser

newtestinguser_111

newUser1

newUser2

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.

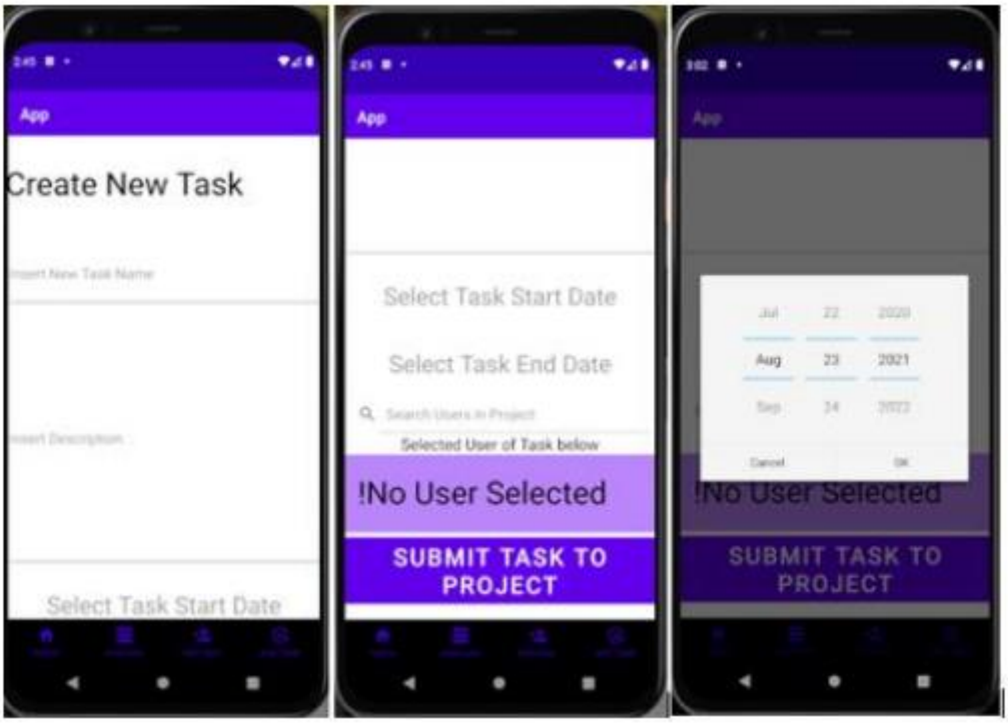
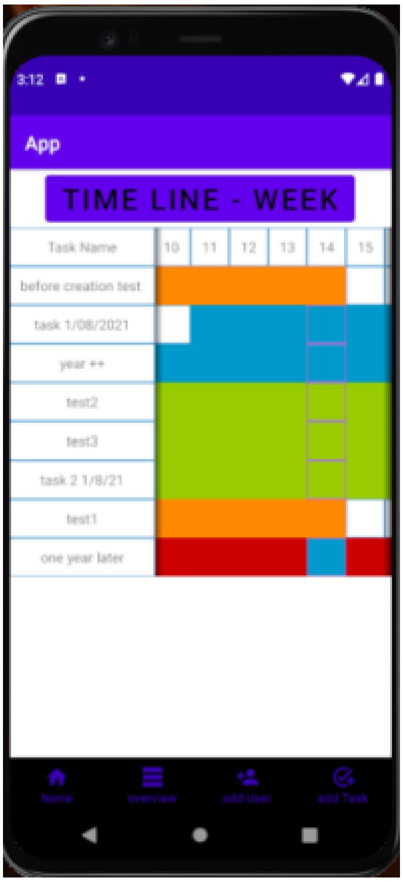


Figure 8: Add Task Tab

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.

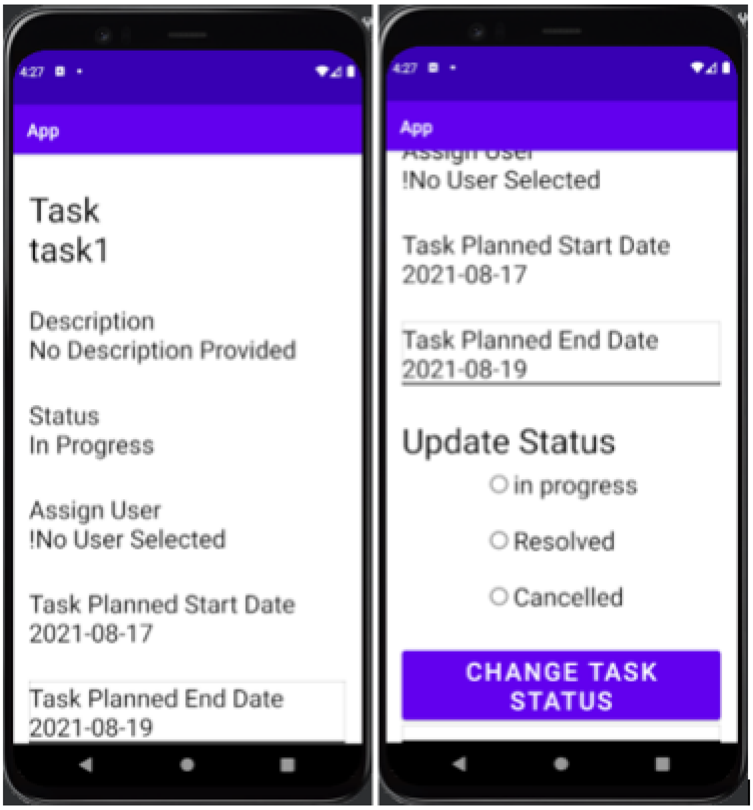


TIME LINE - WEEK

Task Name	10	11	12	13	14	15
before creation test	🟧	🟧	🟧	🟧	🟧	
task 1/08/2021		🟦	🟦	🟦	🟦	🟦
year ++	🟦	🟦	🟦	🟦	🟦	🟦
test2	🟩	🟩	🟩	🟩	🟩	🟩
test3	🟩	🟩	🟩	🟩	🟩	🟩
task 2 1/8/21	🟩	🟩	🟩	🟩	🟩	🟩
test1	🟧	🟧	🟧	🟧	🟧	
one year later	🟥	🟥	🟥	🟥	🟦	🟥

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.



Task
task1

Description
No Description Provided

Status
In Progress

Assign User
!No User Selected

Task Planned Start Date
2021-08-17

Task Planned End Date
2021-08-19

Assign User
!No User Selected

Task Planned Start Date
2021-08-17

Task Planned End Date
2021-08-19

Update Status

○ in progress
○ Resolved
○ Cancelled

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.

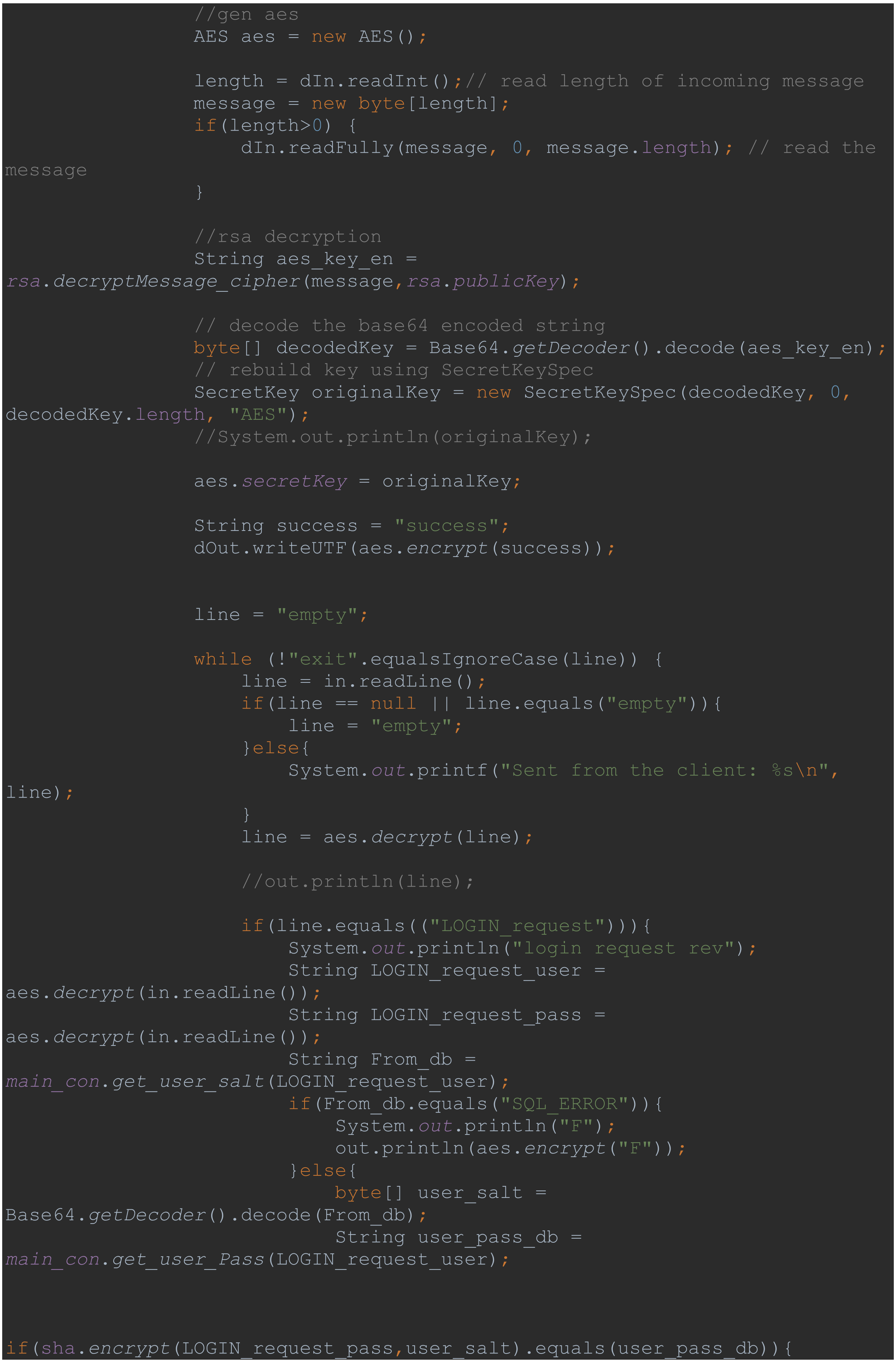
Appendices

Source code

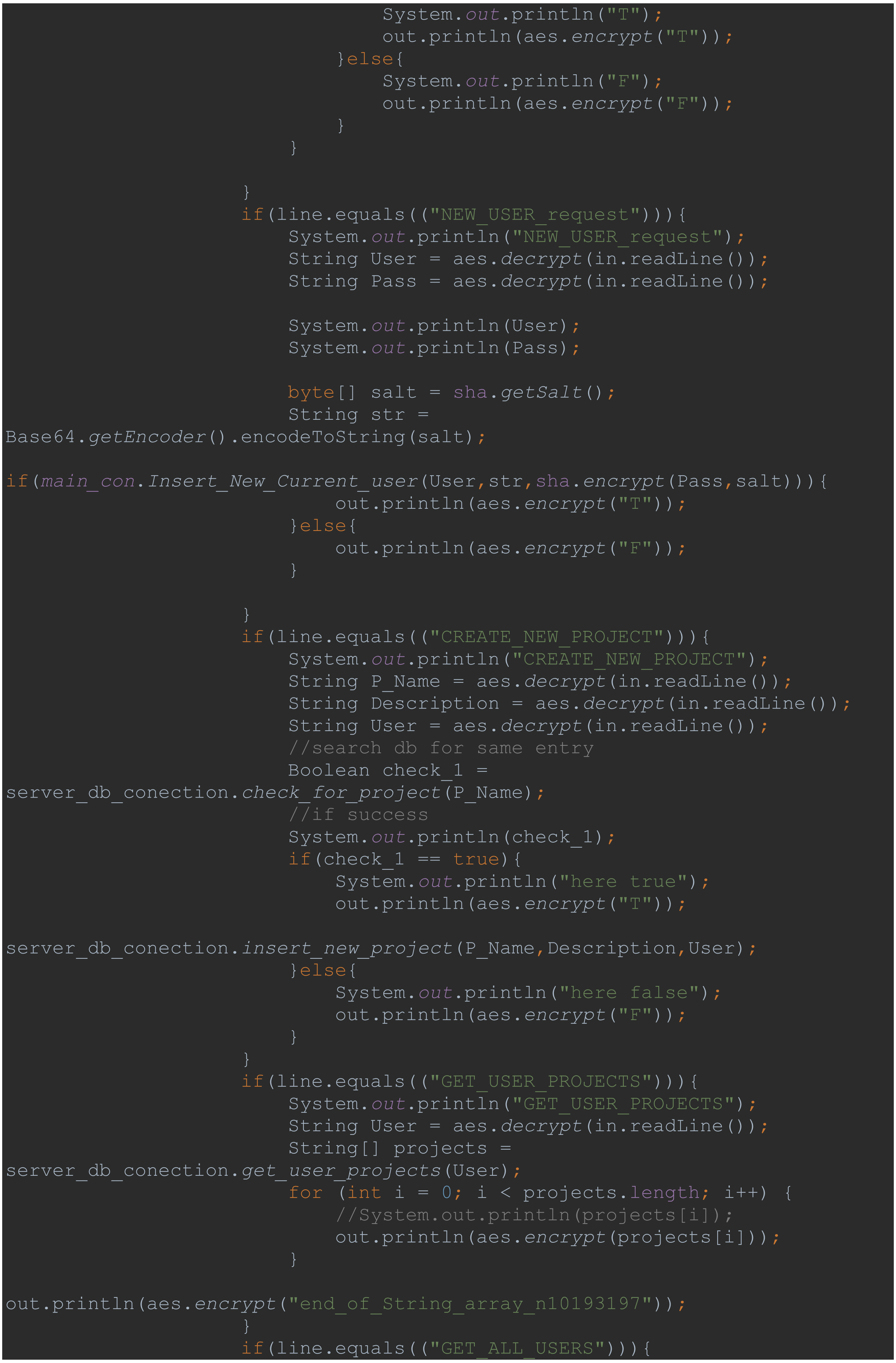
Server

Main

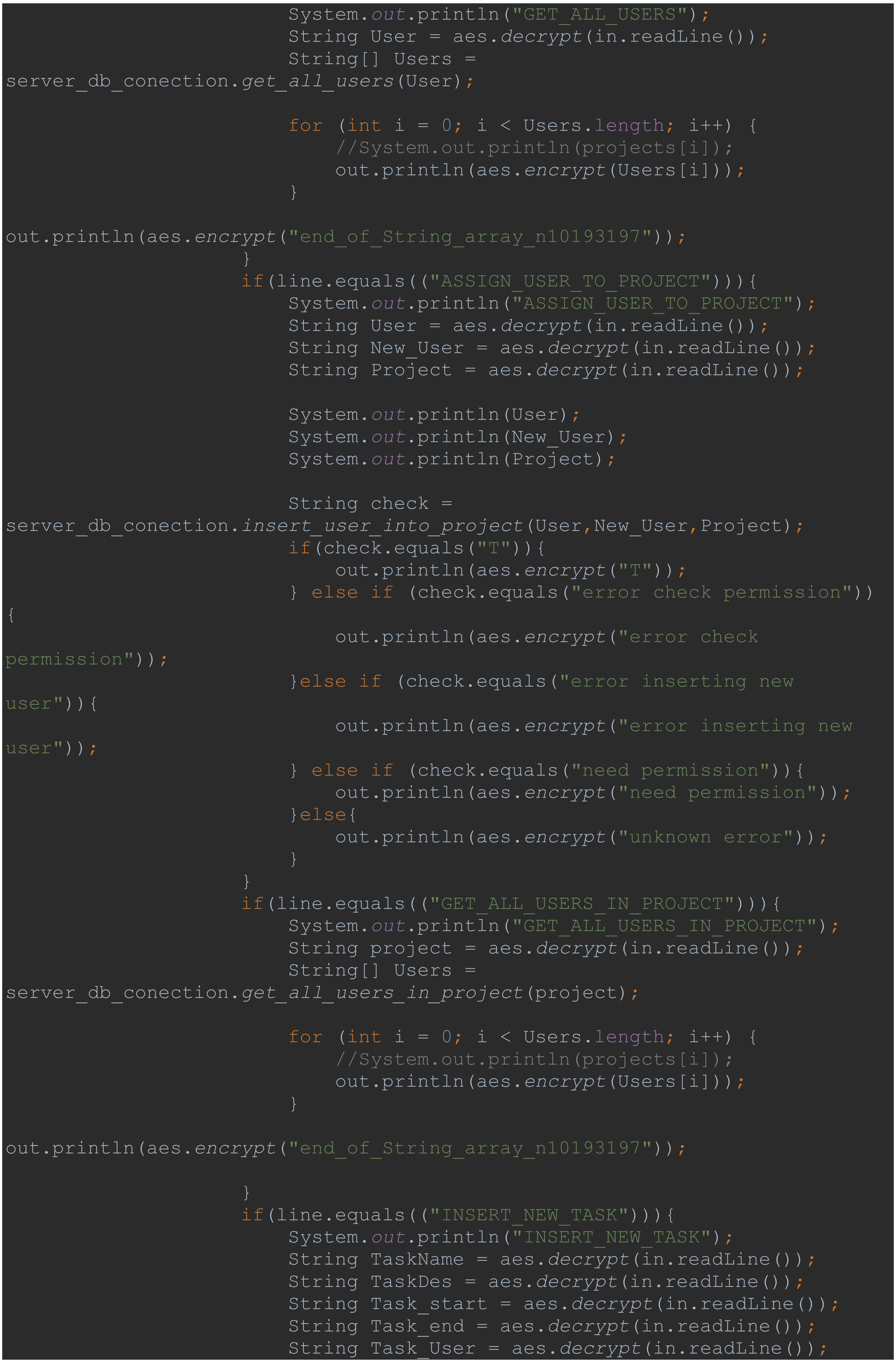
```
package com.company;

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","j
pate101");
        }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);
```



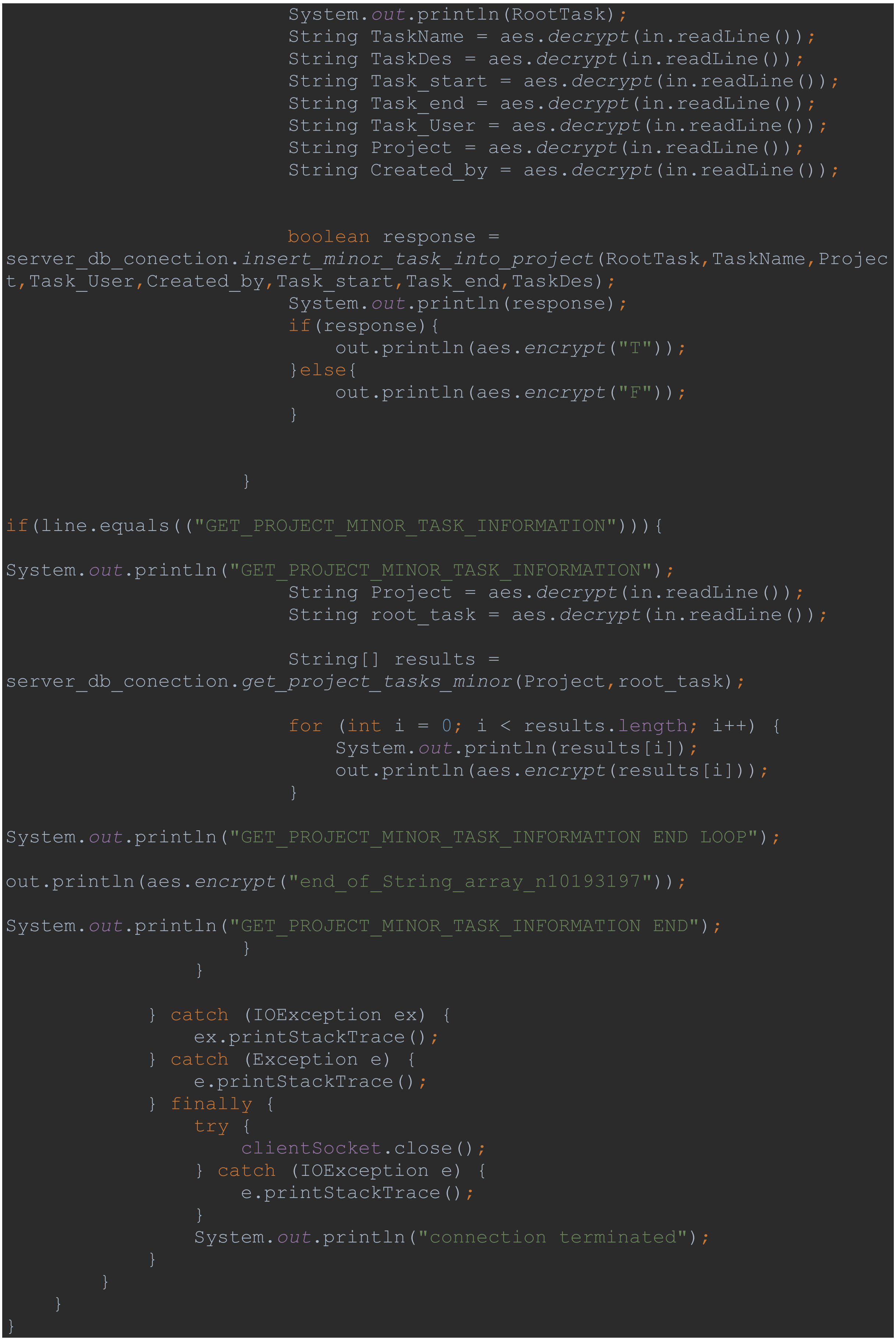
```
                //gen aes
                AES aes = new AES();

                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,Cre
ated_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,Projec
t,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("UT
F-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

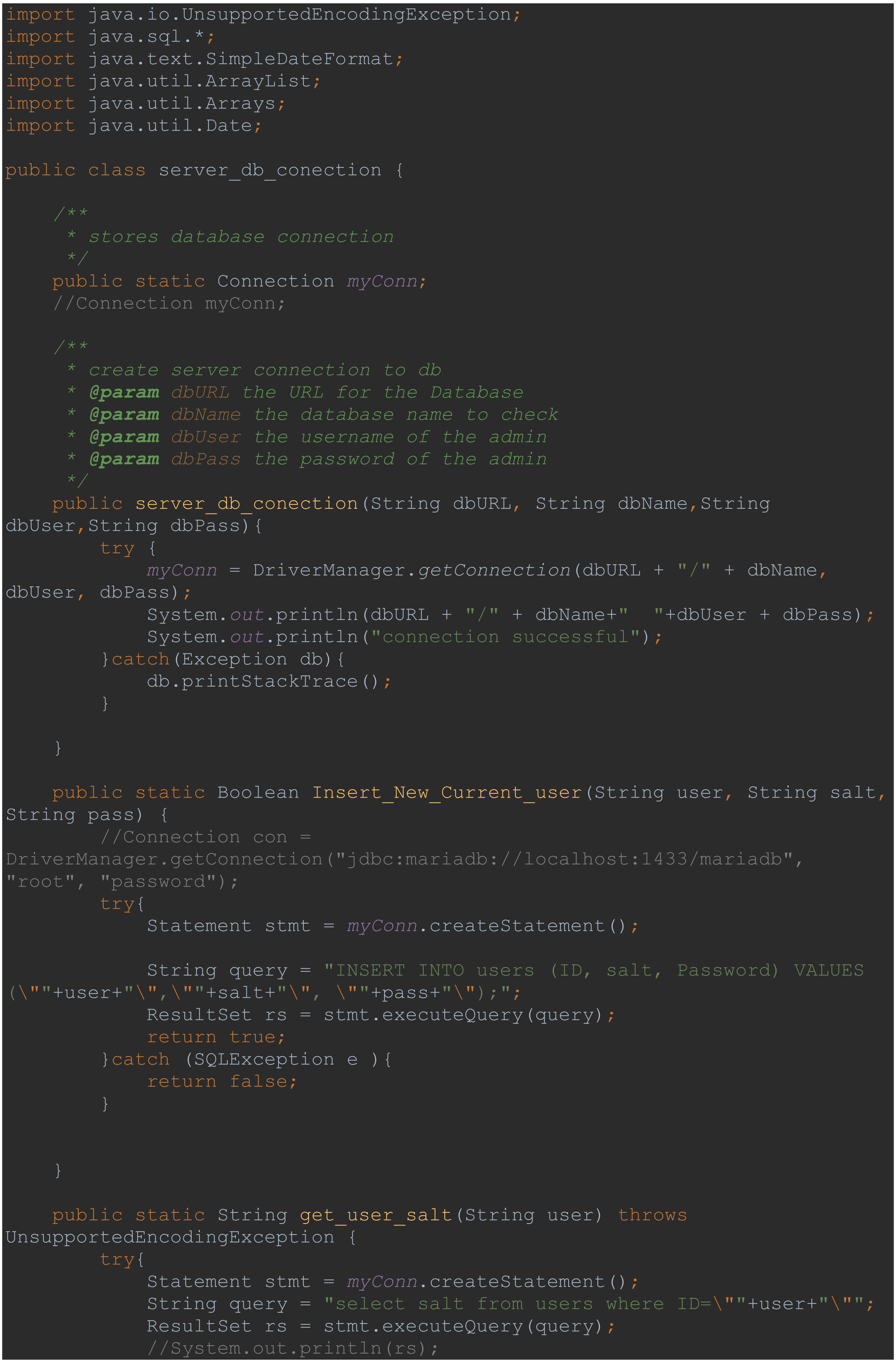
```
package com.company;

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

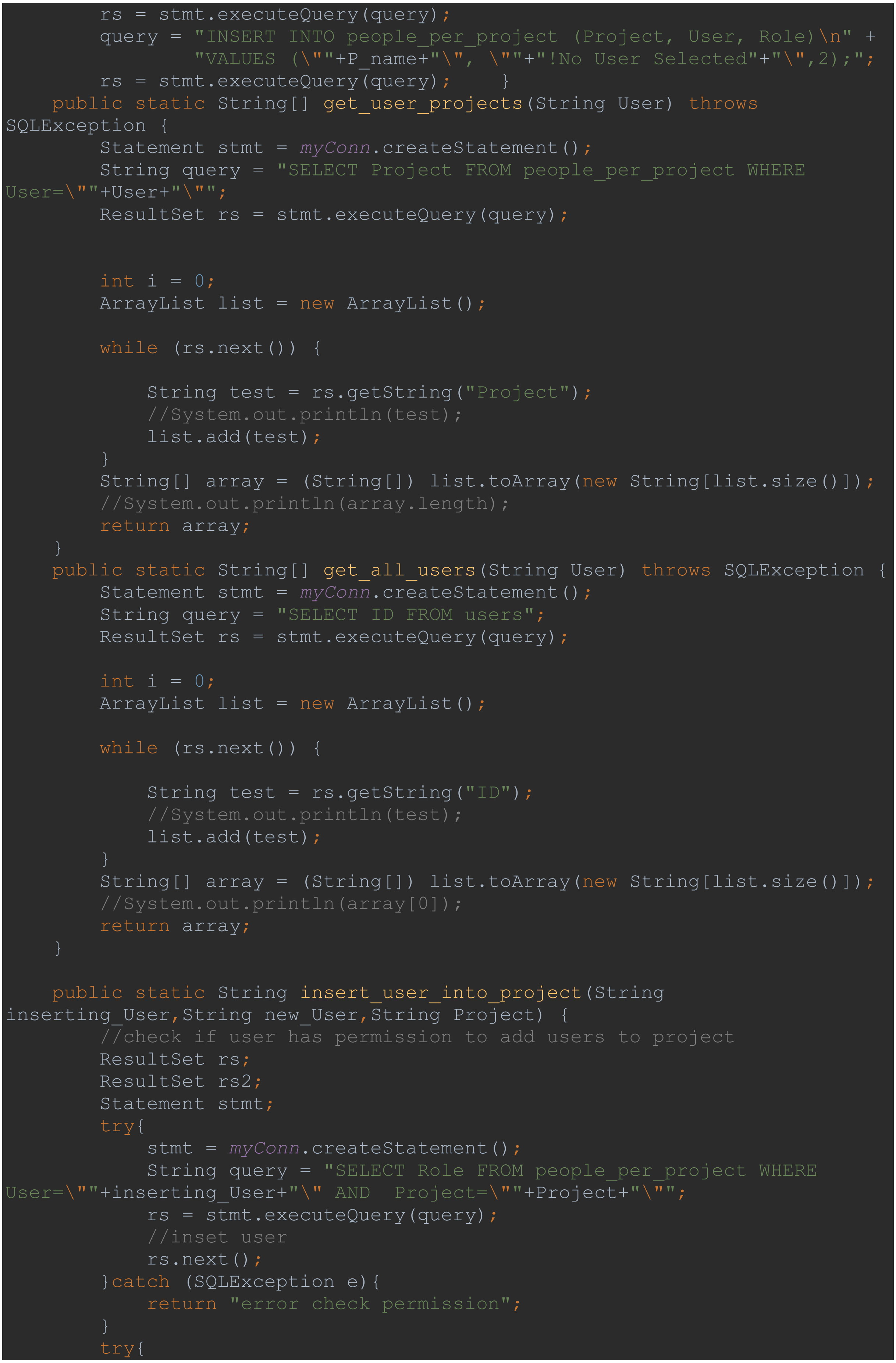
```
package com.company;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.util.Base64;

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

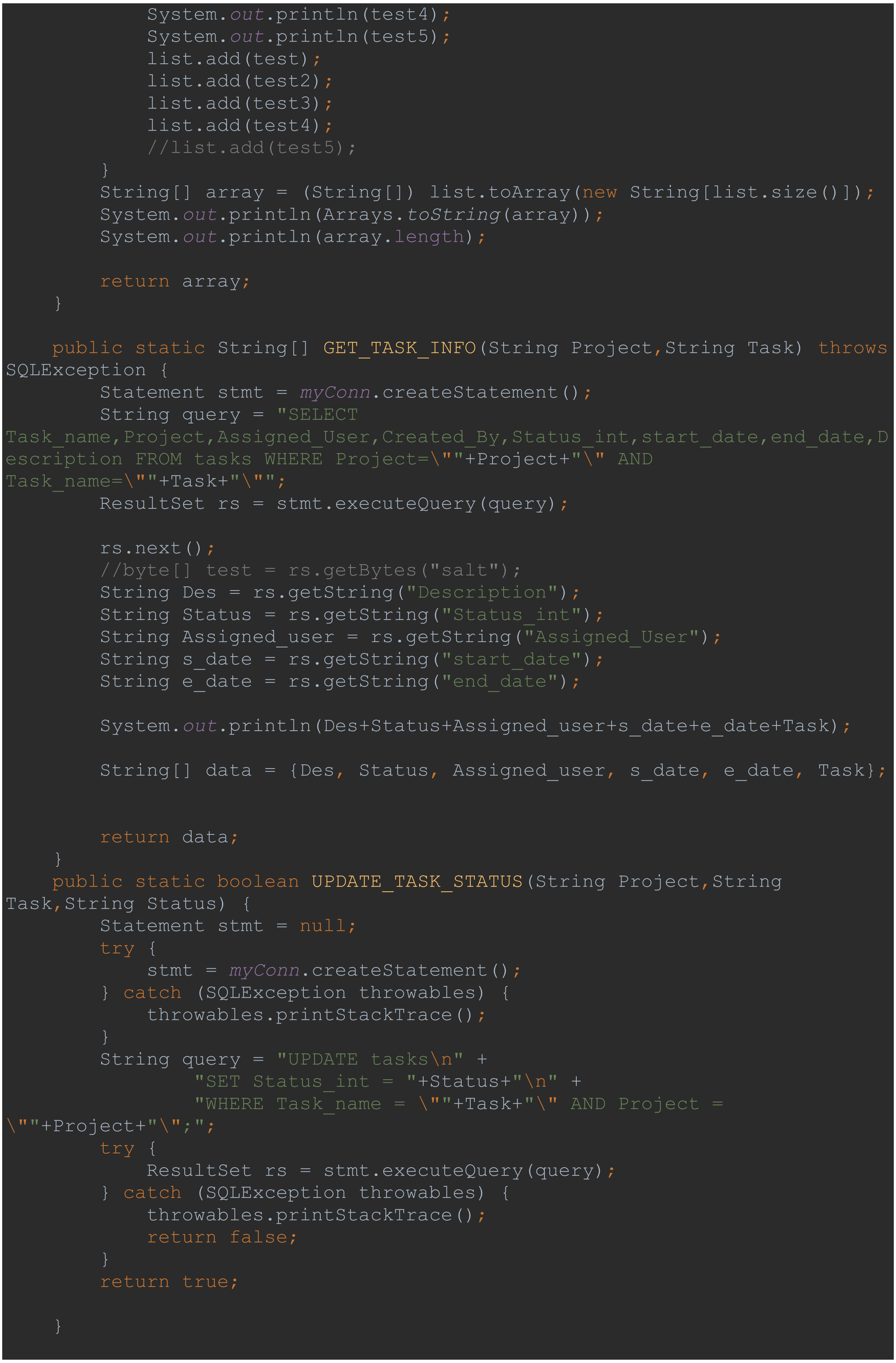
```
package com.company;

import java.io.InputStream;
```



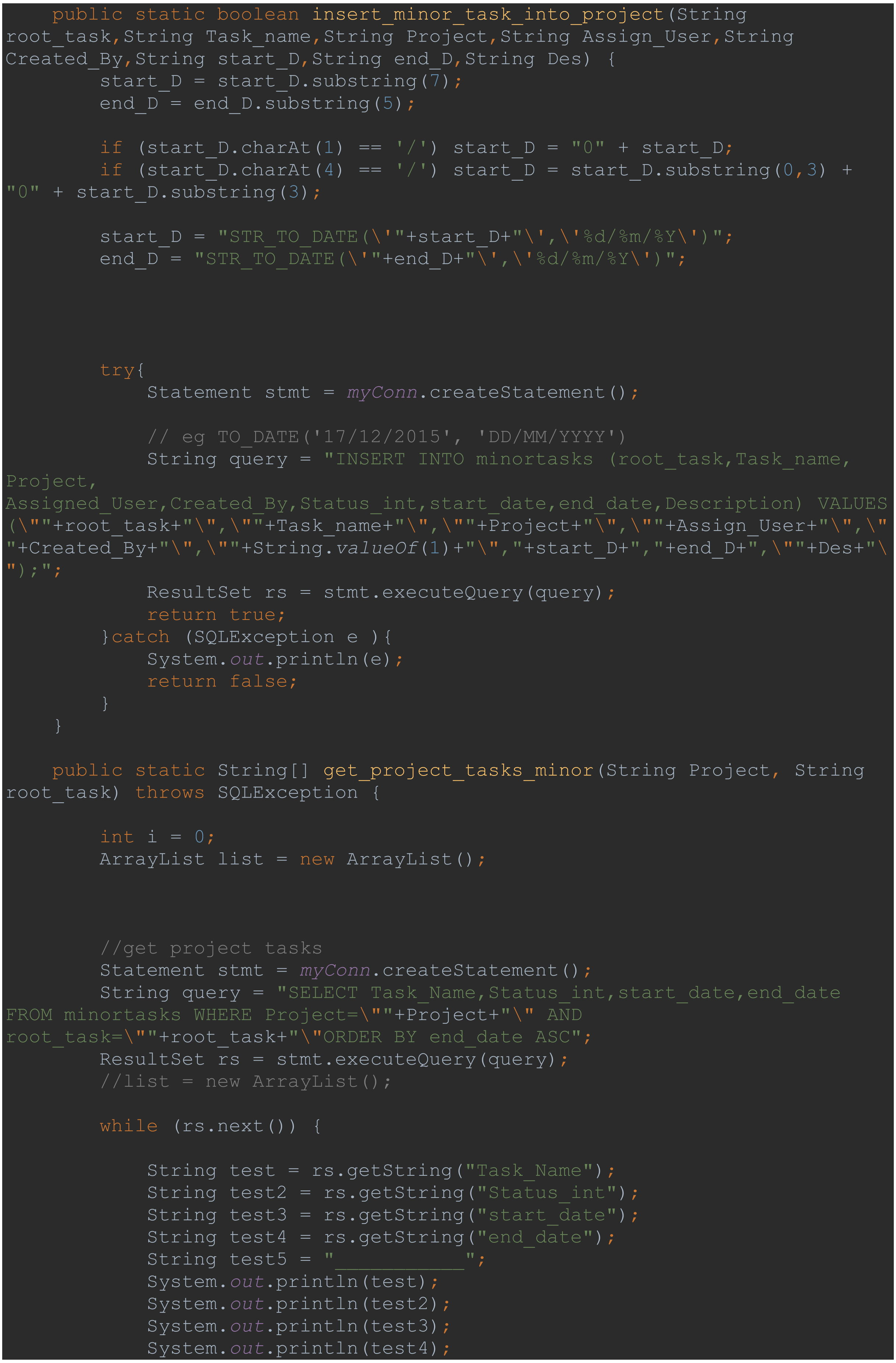
```
import java.io.UnsupportedEncodingException;
import java.sql.*;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;

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,D
escription 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;
    }


}
```

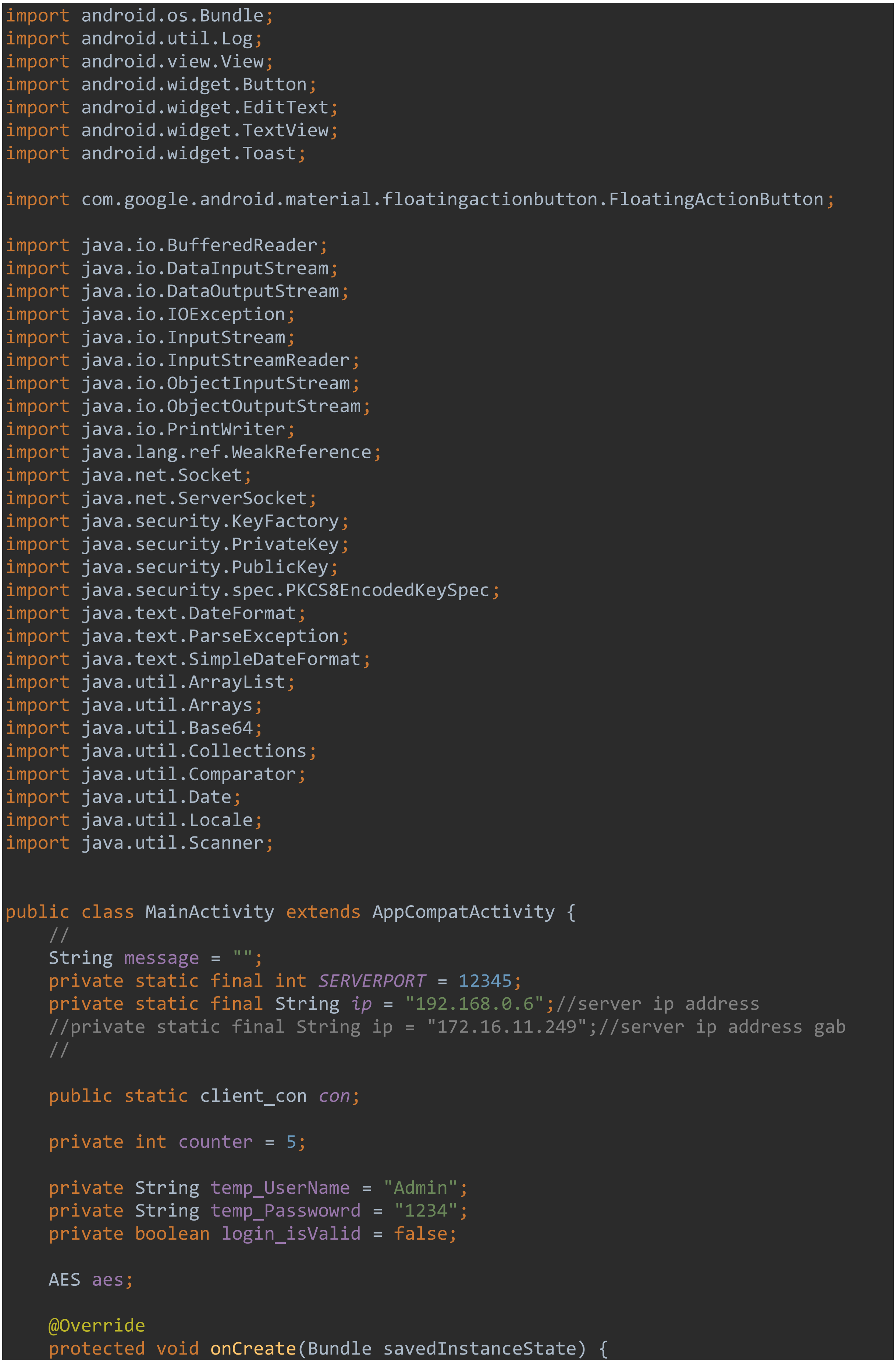
Client app

Manifest scripts

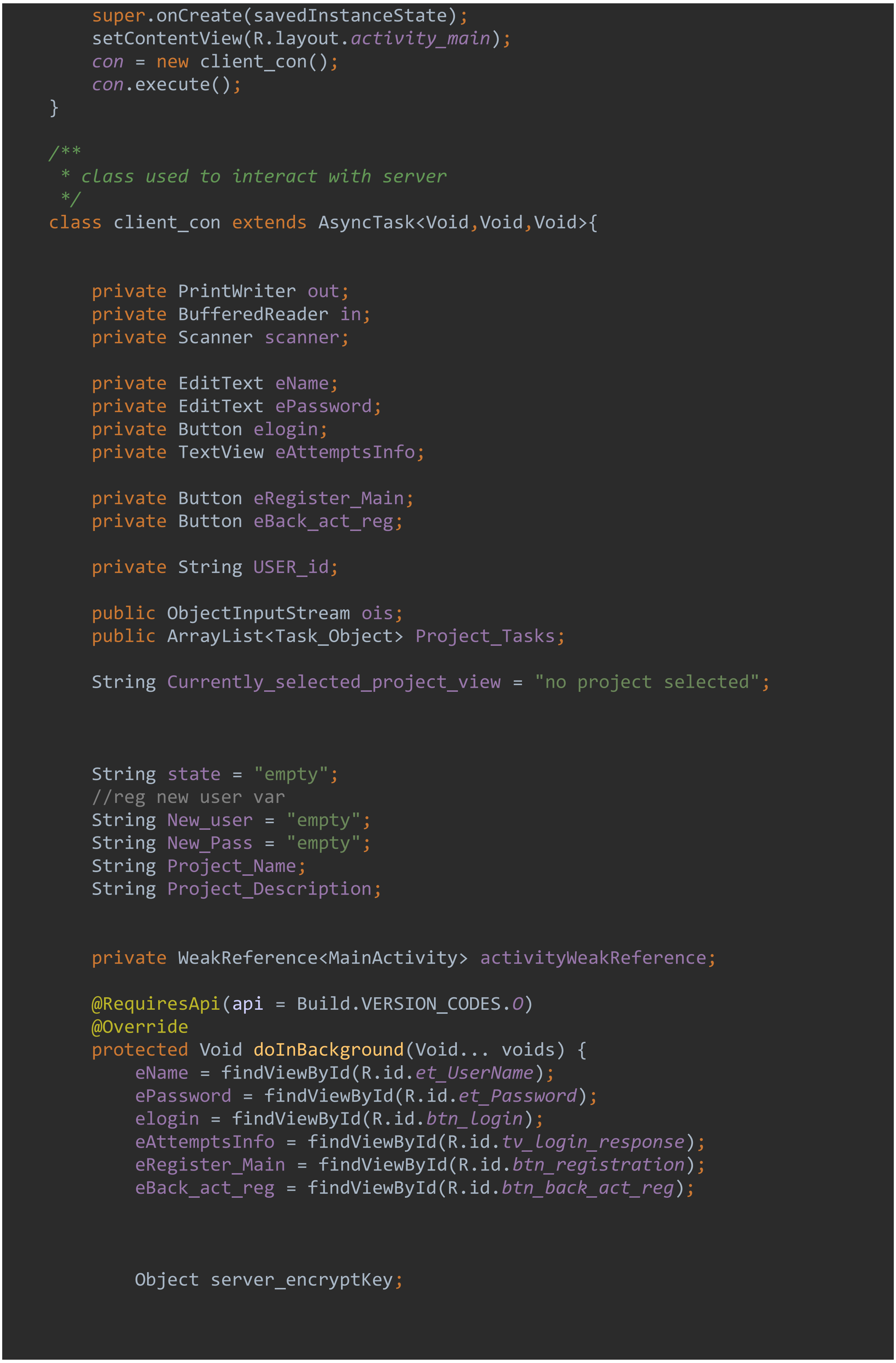
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.app">

    <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

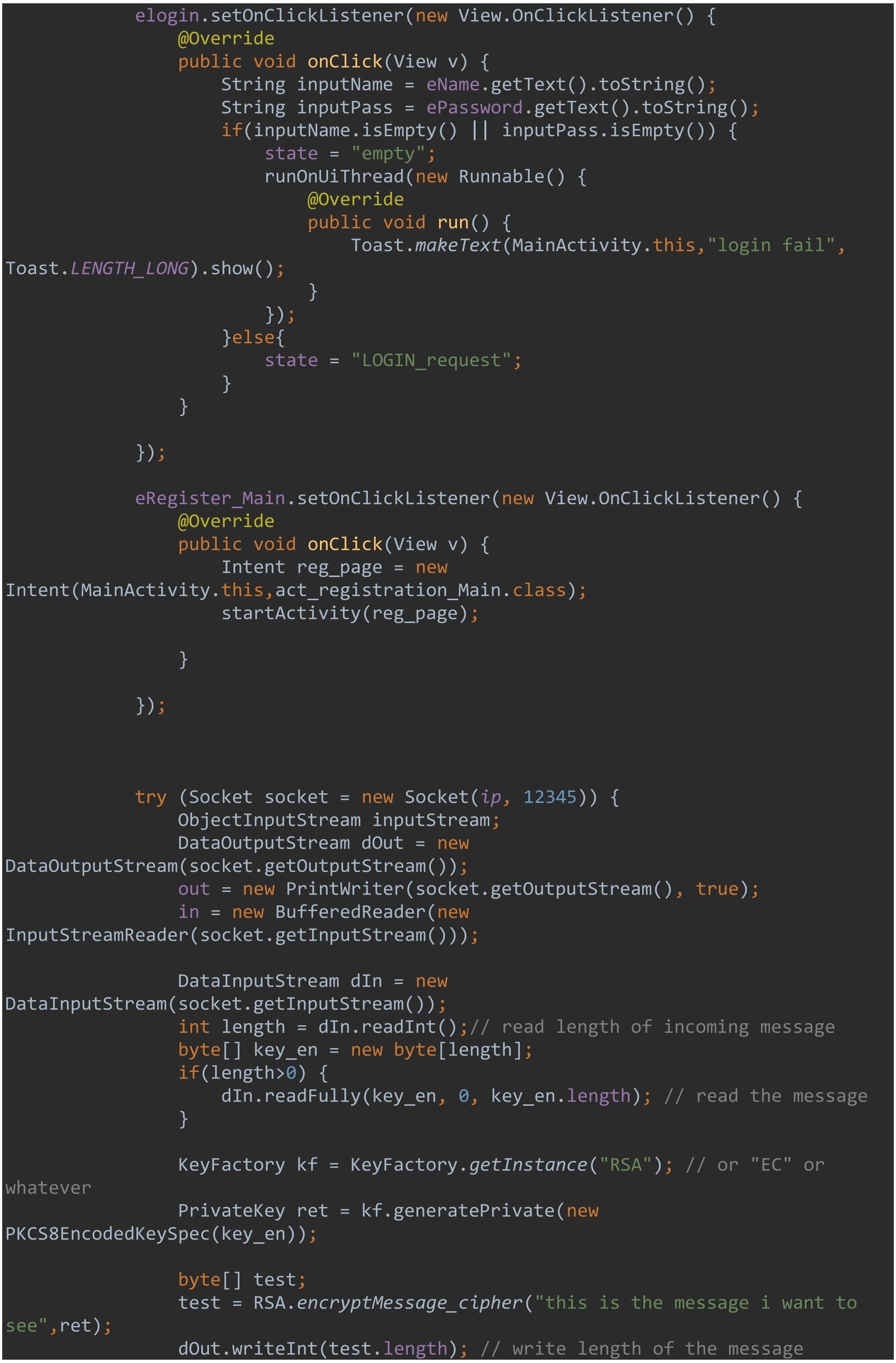
```
package com.example.app;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.AsyncTask;
import android.os.Build;
```



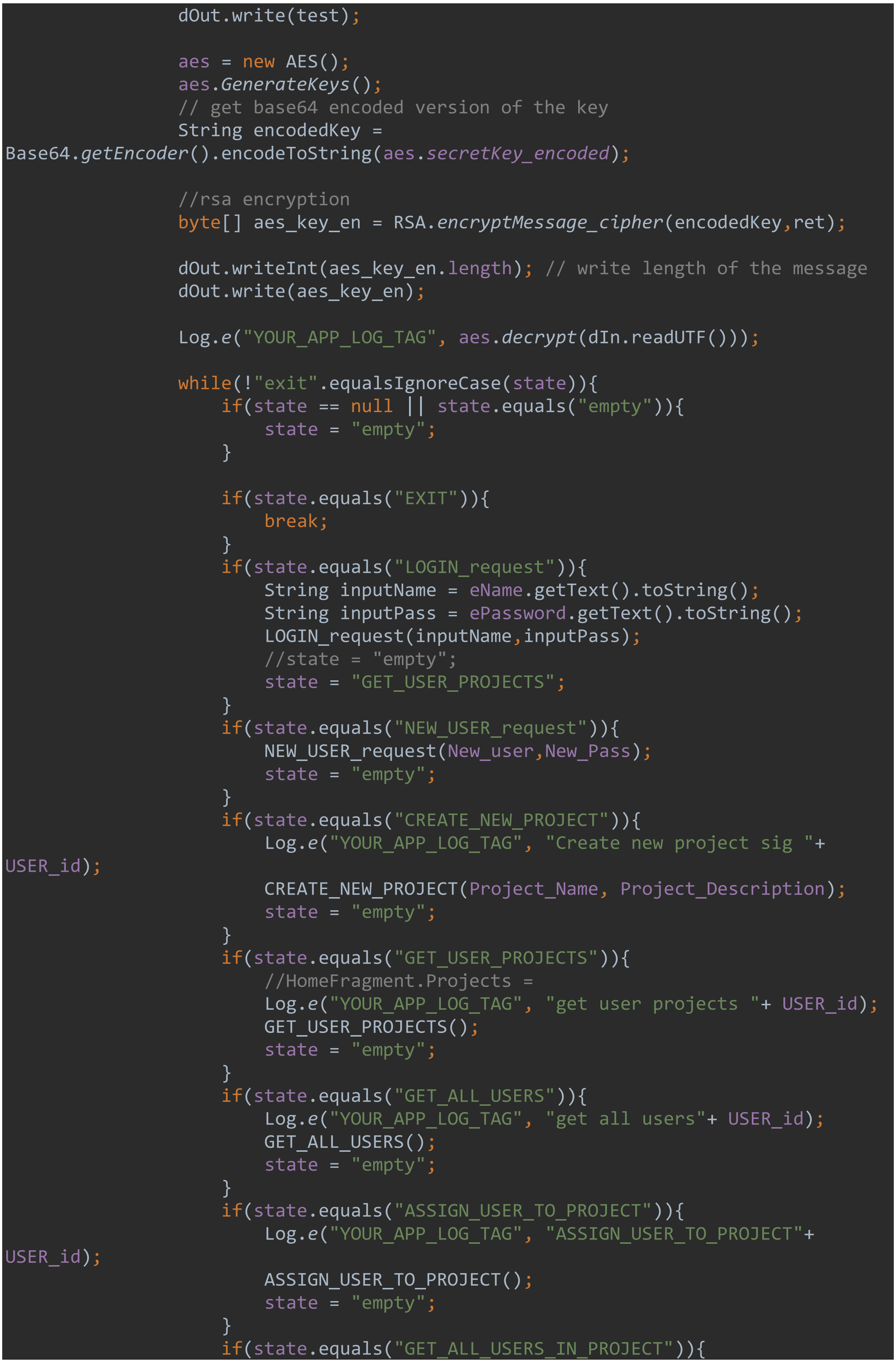
```java
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

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) {
```



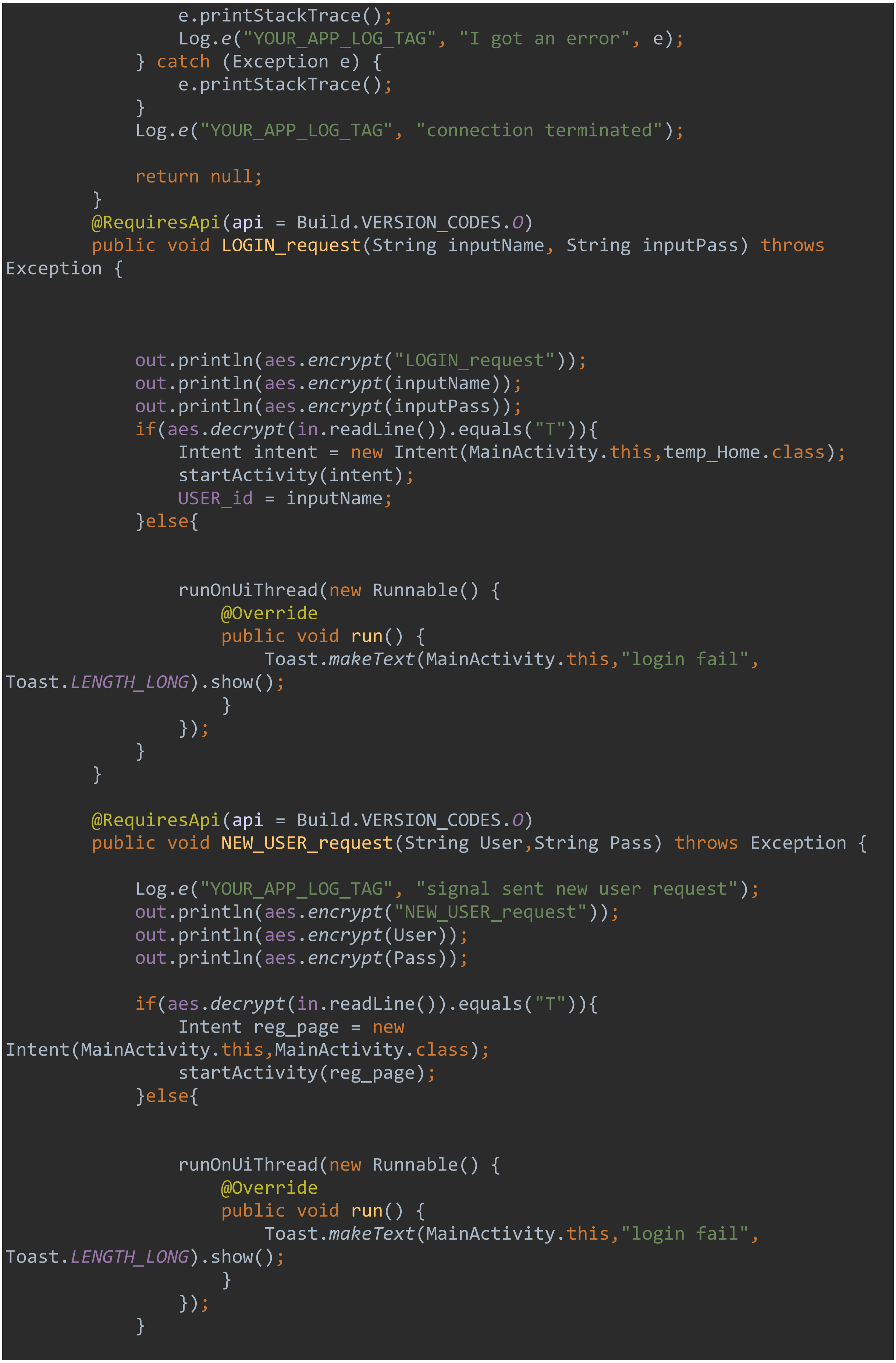
```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        con = new client_con();
        con.execute();
    }

    /**
     * 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;
```



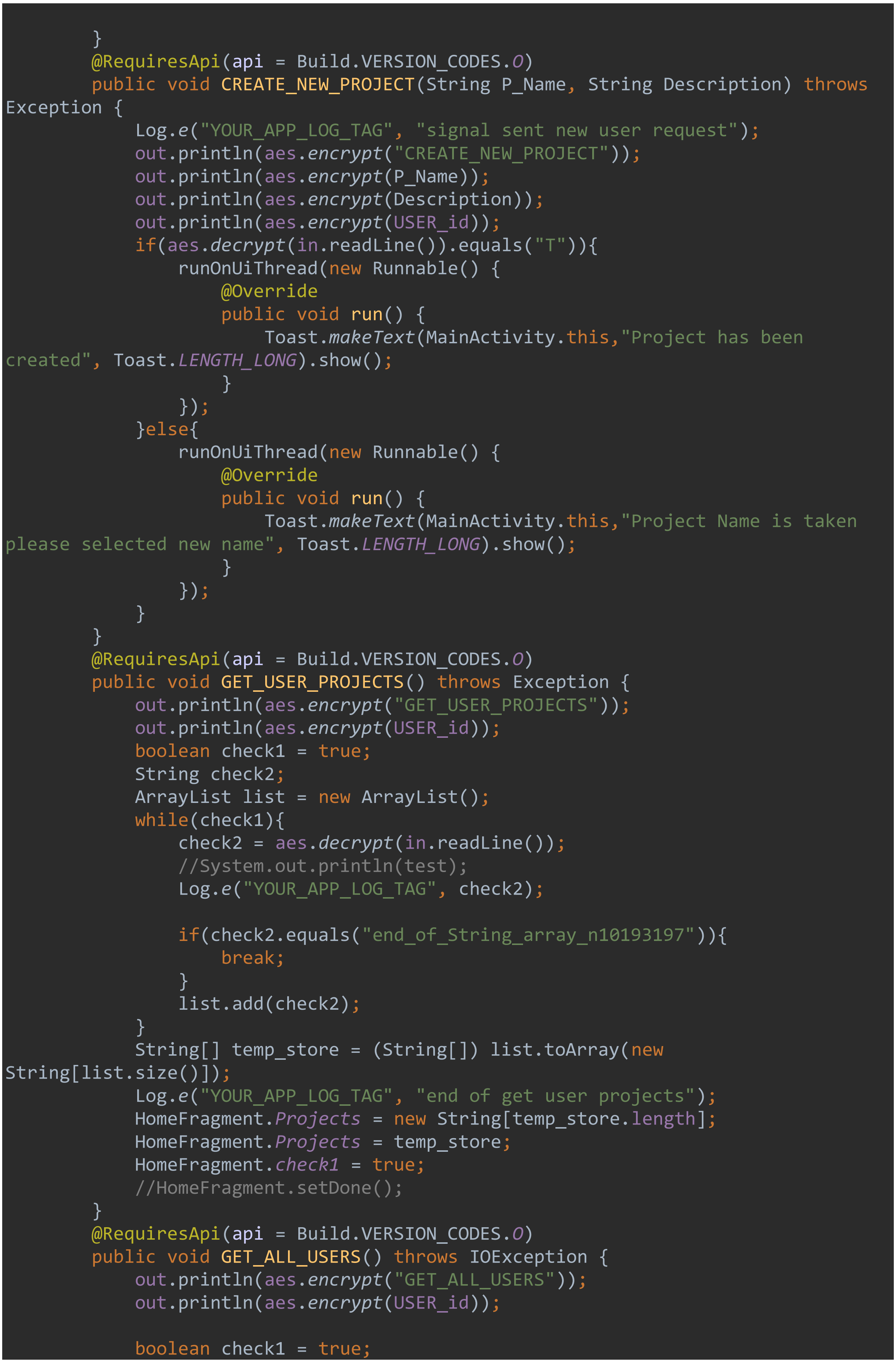
```
            elogin.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    String inputName = eName.getText().toString();
                    String inputPass = ePassword.getText().toString();
                    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
```



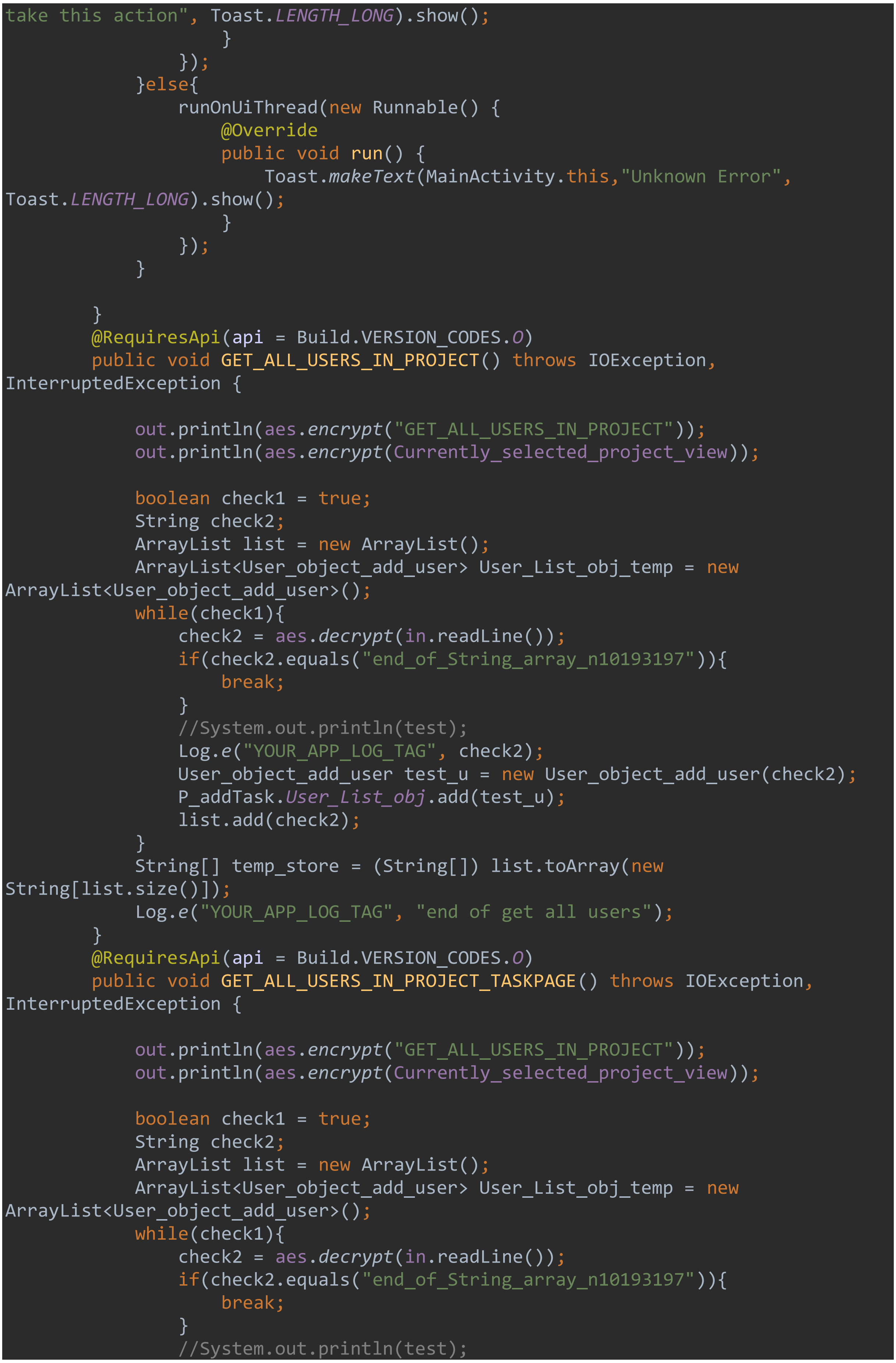
```
                dOut.write(test);

                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) {
```



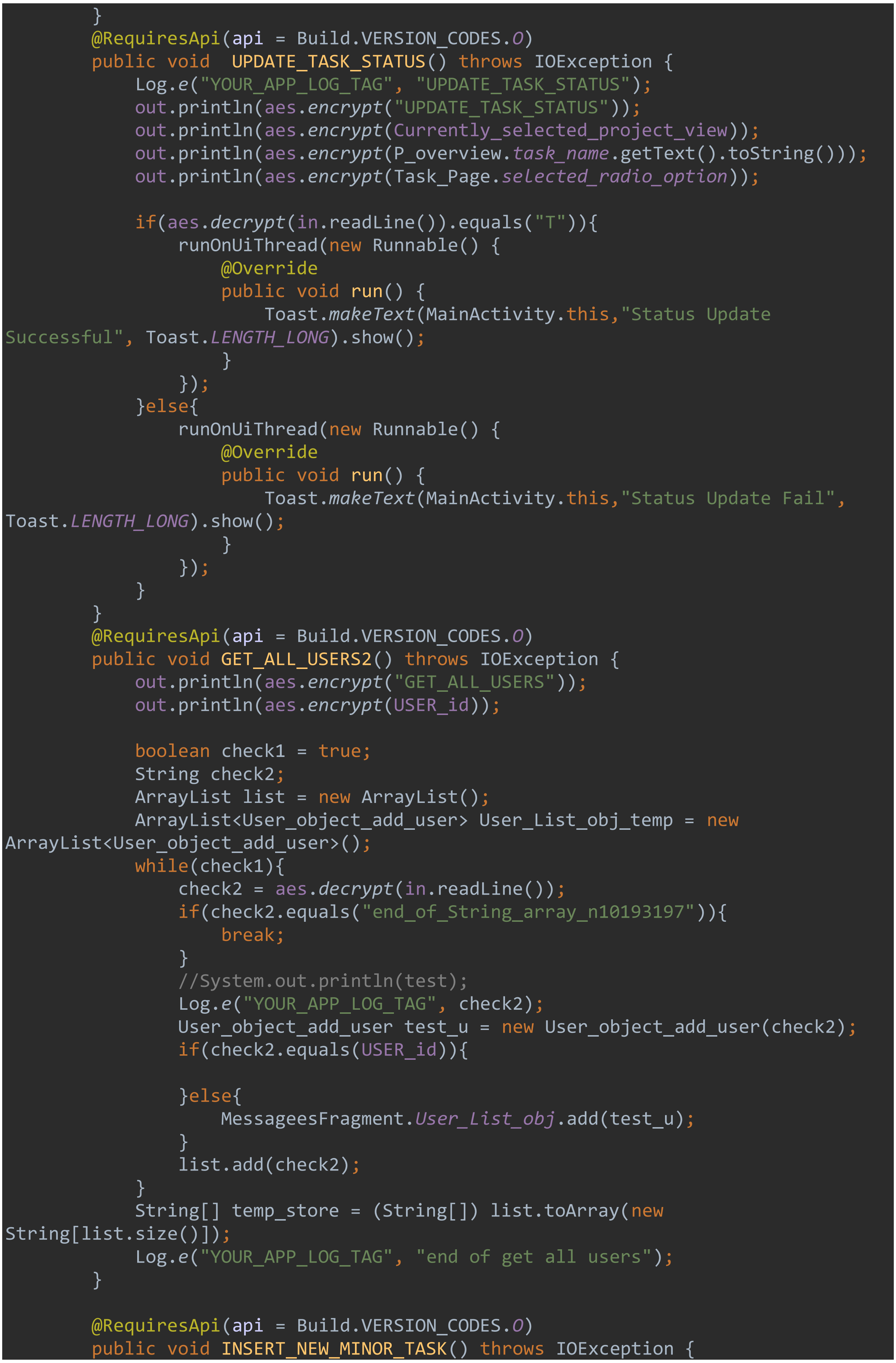
```
                e.printStackTrace();
                Log.e("YOUR_APP_LOG_TAG", "I got an error", e);
            } catch (Exception e) {
                e.printStackTrace();
            }
            Log.e("YOUR_APP_LOG_TAG", "connection terminated");

            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);
```



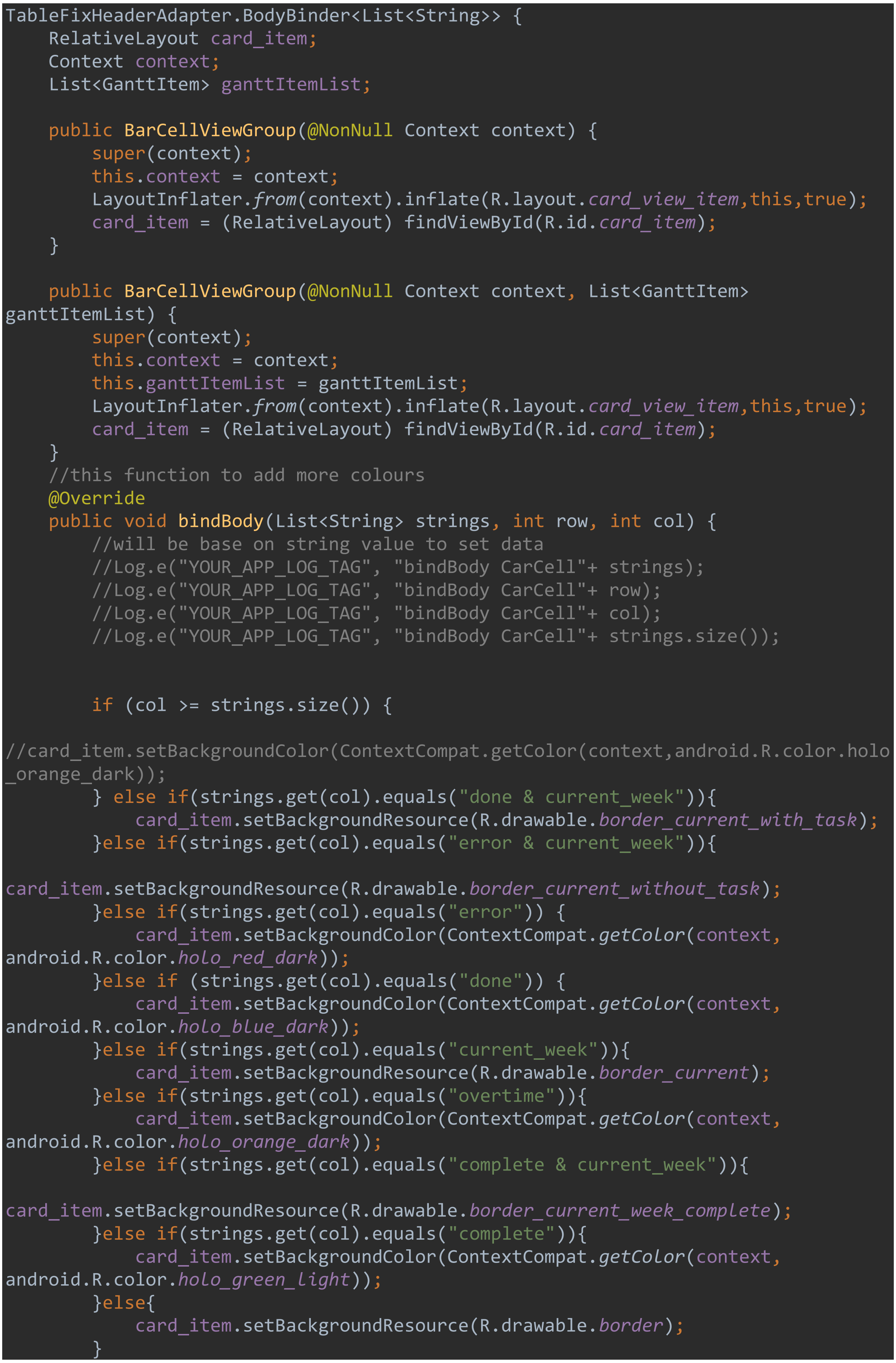
```
        }
        @RequiresApi(api = Build.VERSION_CODES.O)
        public void  UPDATE_TASK_STATUS() throws IOException {
            Log.e("YOUR_APP_LOG_TAG", "UPDATE_TASK_STATUS");
            out.println(aes.encrypt("UPDATE_TASK_STATUS"));
            out.println(aes.encrypt(Currently_selected_project_view));
            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);
        }
    }
}
```

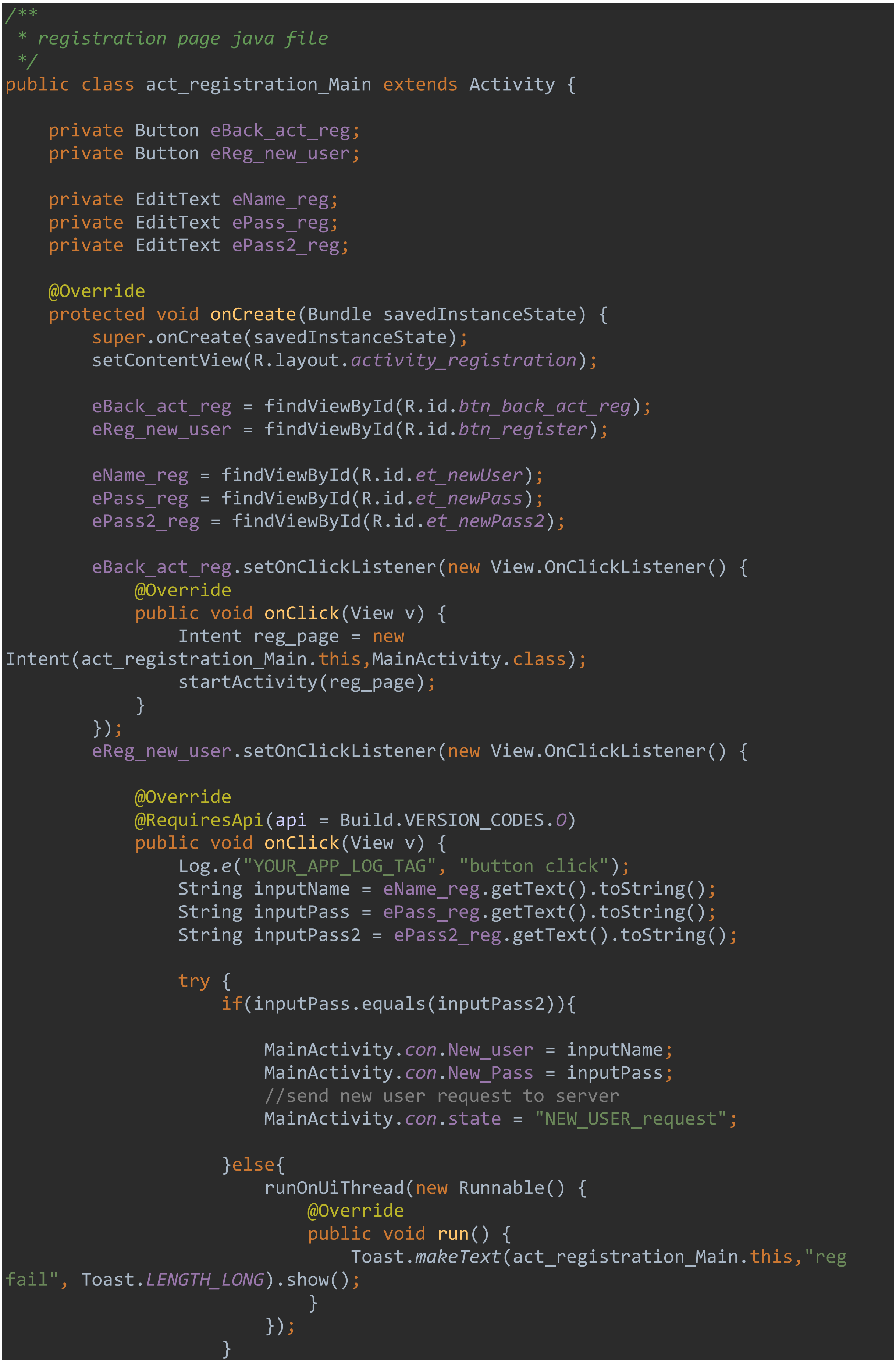
```
package com.example.app.ViewGroup;

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);


        }
    }
}
```

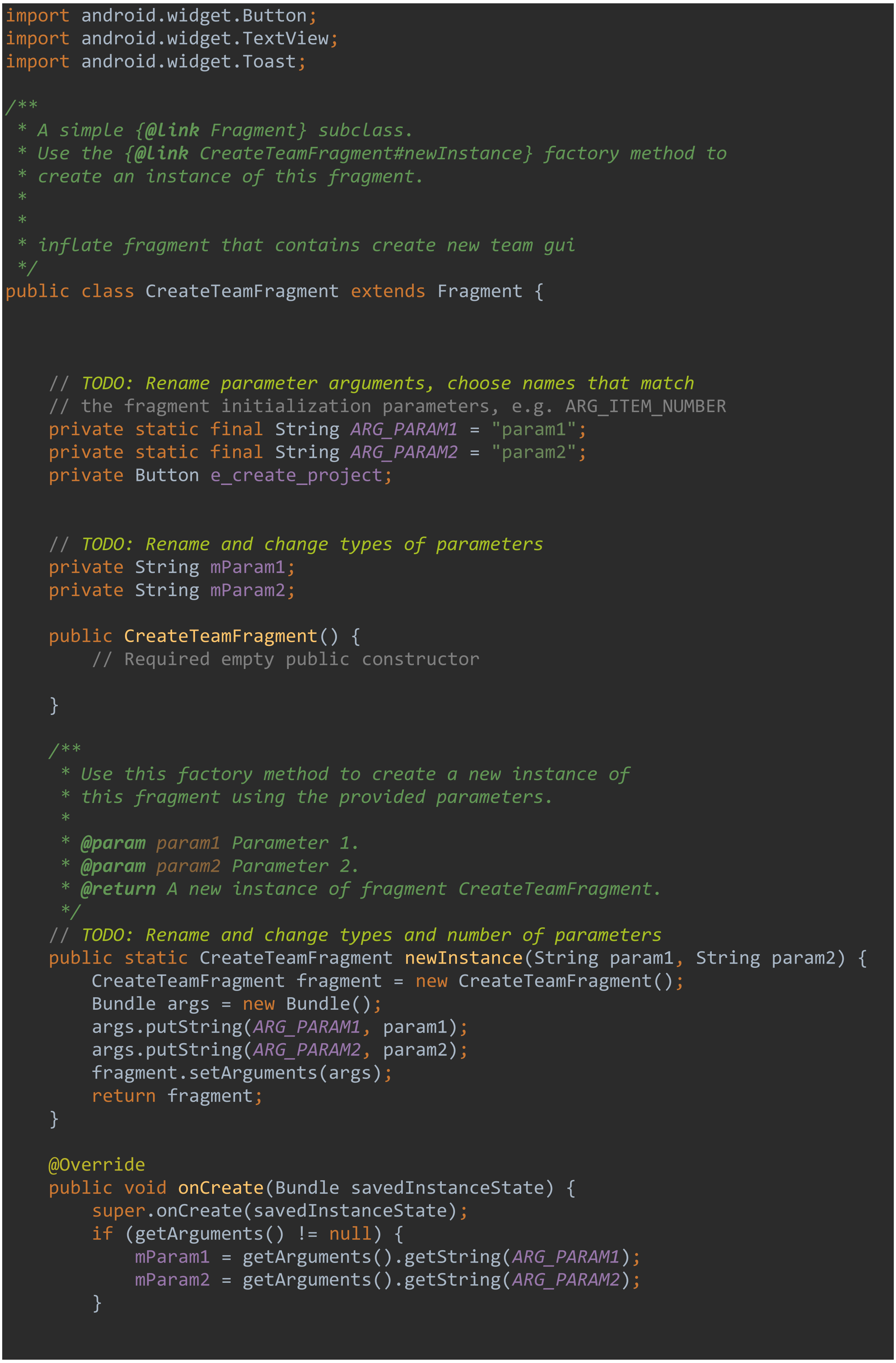
```
package com.example.app;

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;

```



```java
/**
 * 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;
```



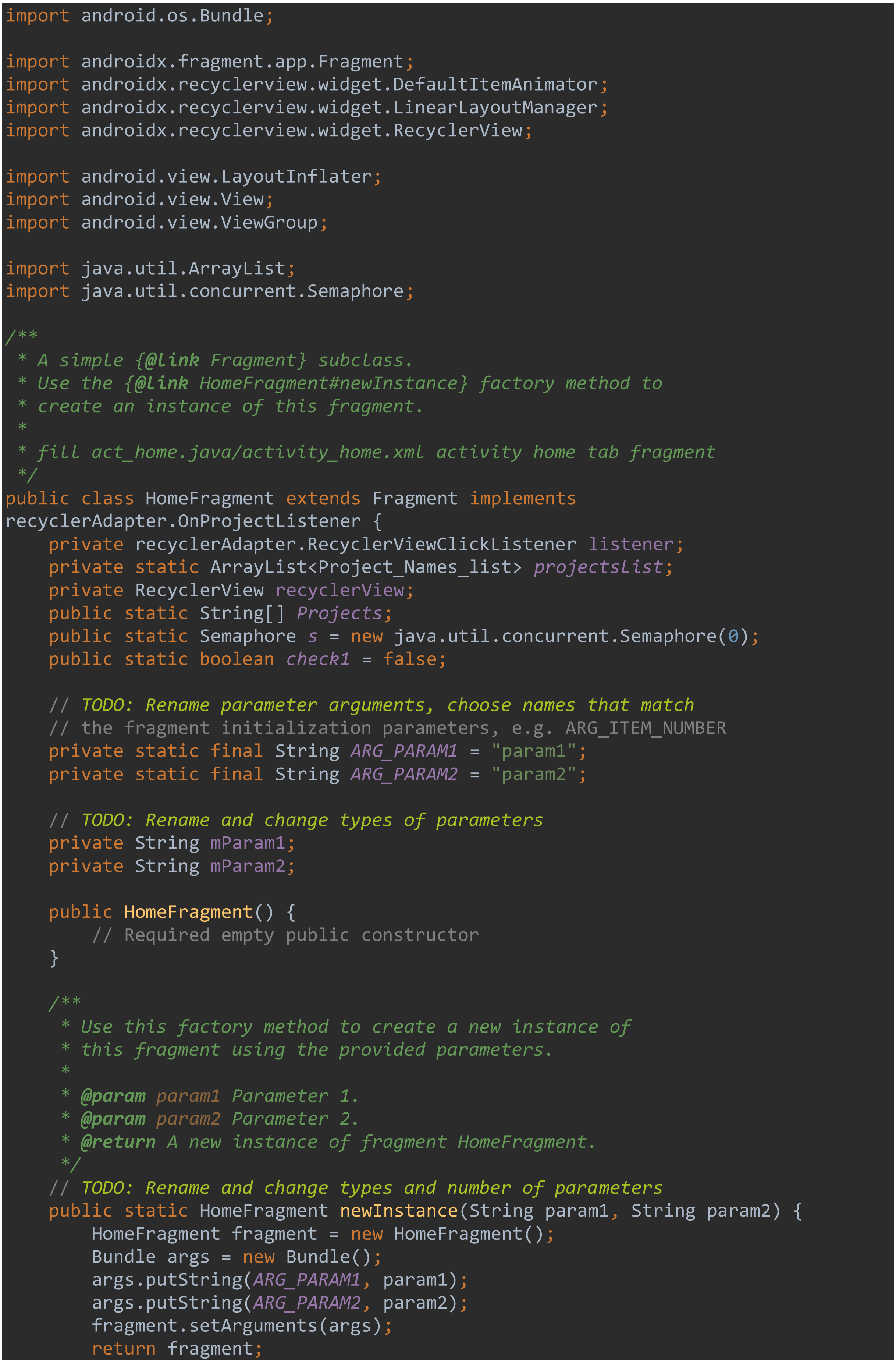
```
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

/**
 * 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);
    }
}
```

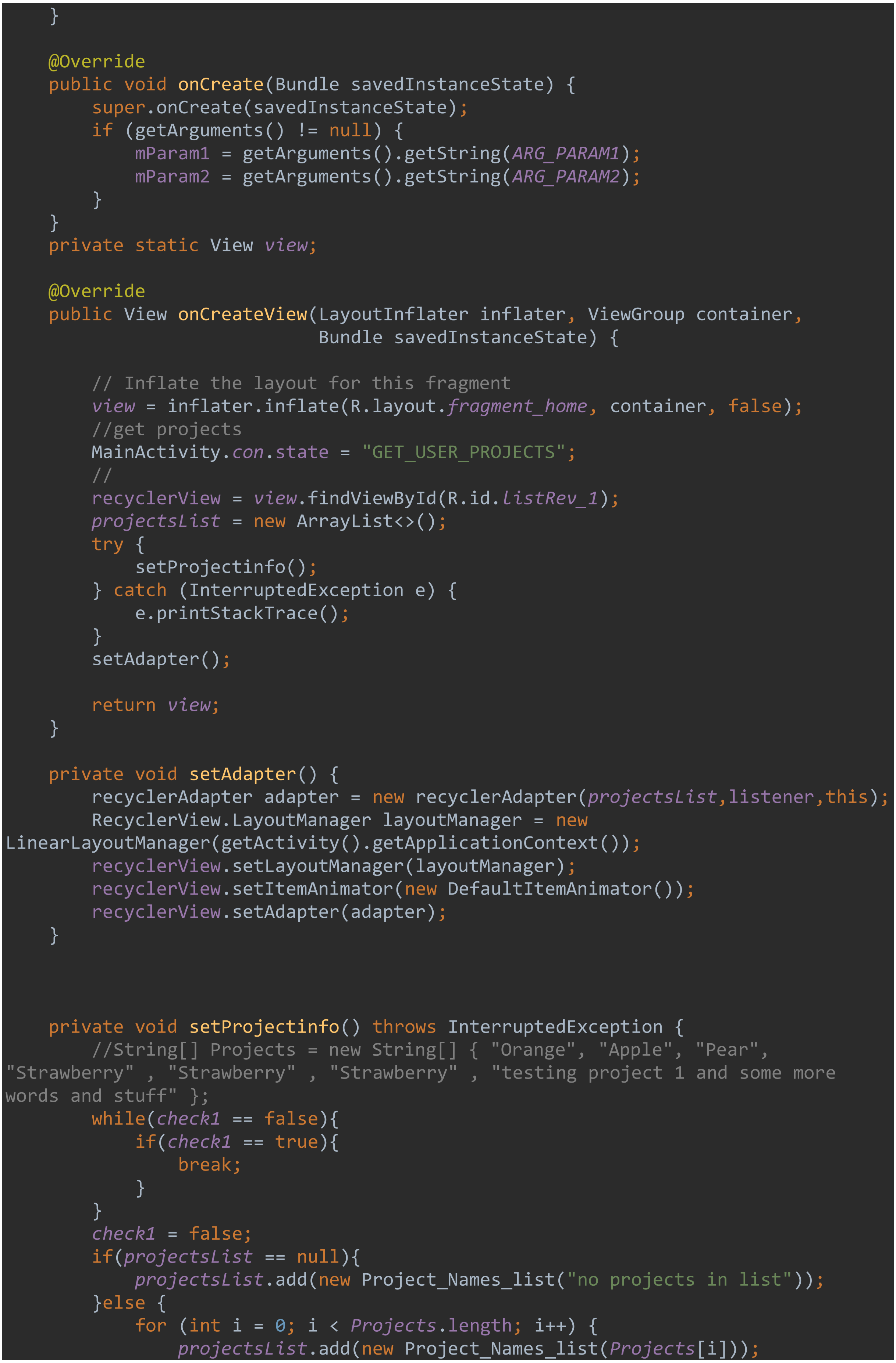
```
package com.example.app;

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;
```

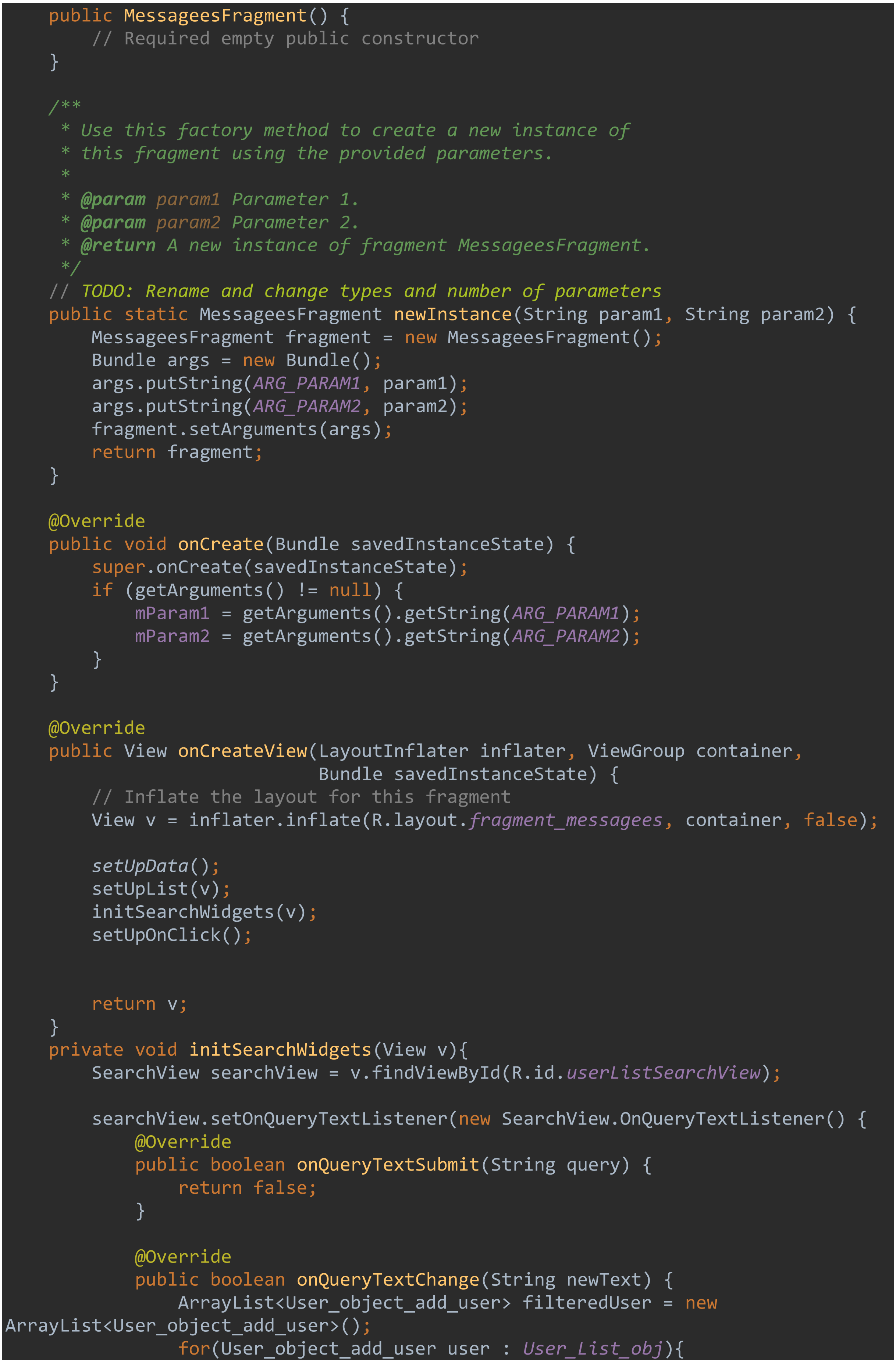


```java
import android.os.Bundle;

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);
    }
}
```

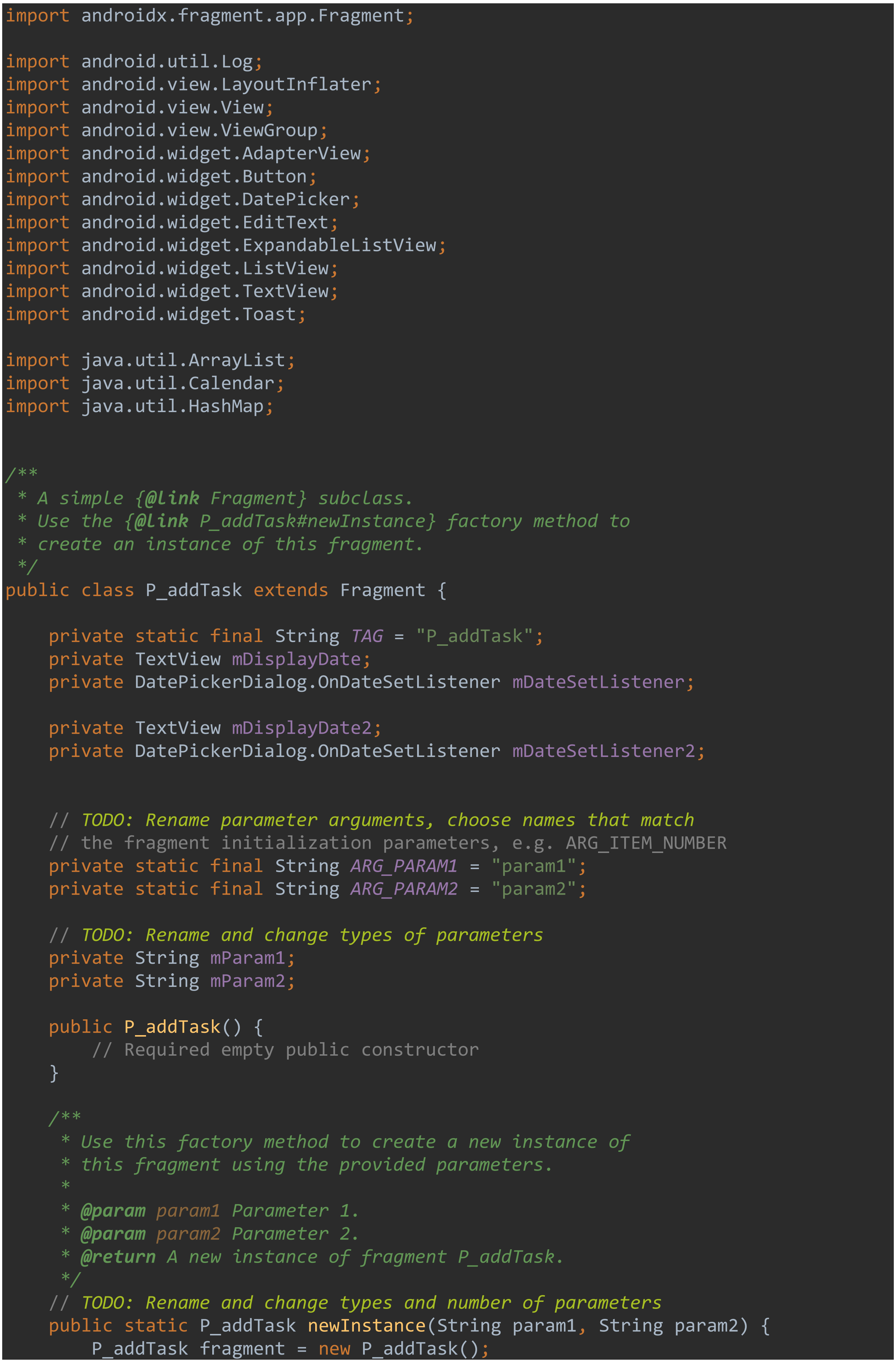
```
package com.example.app;

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;
```



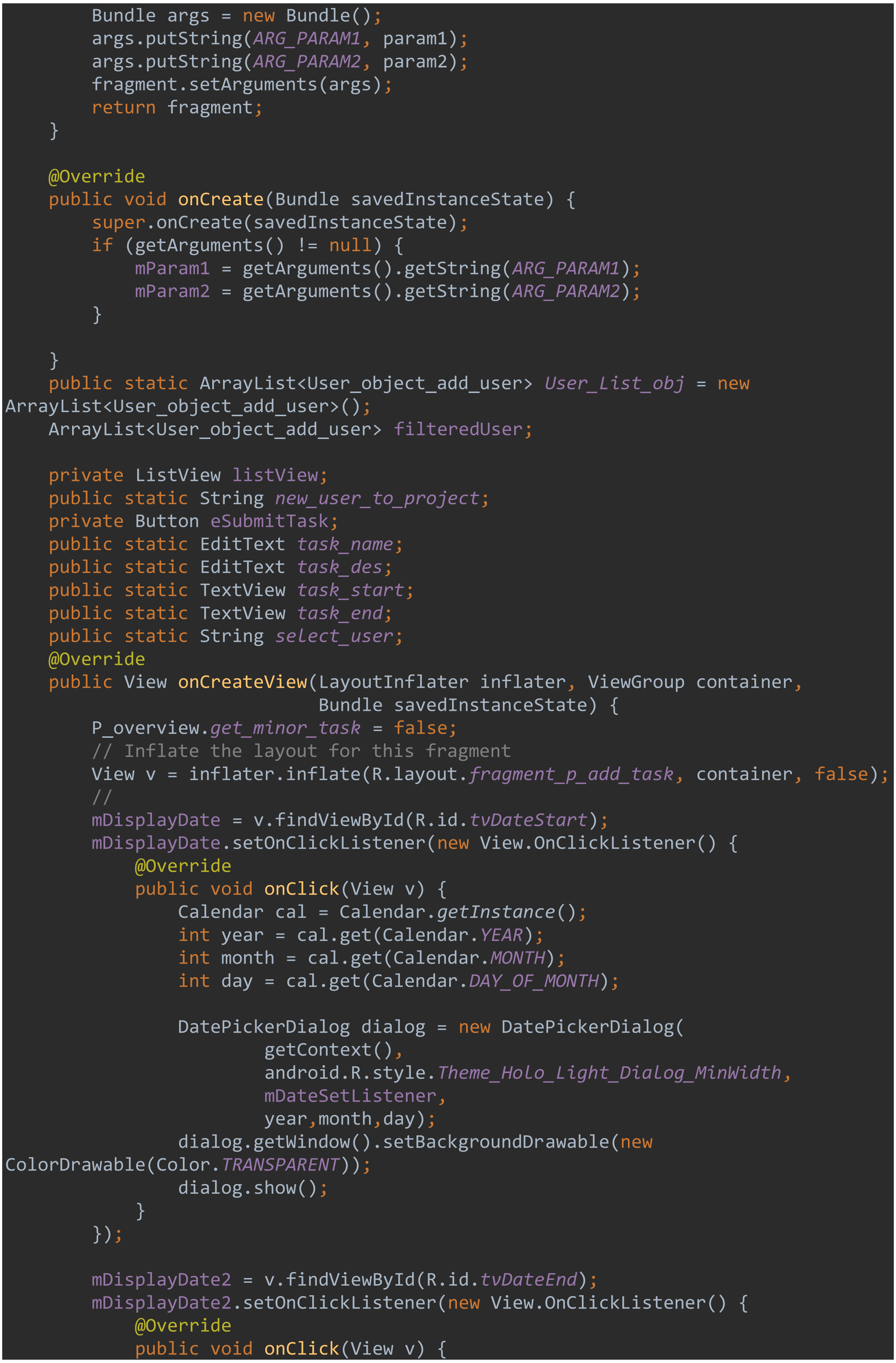
```
    public MessageesFragment() {
        // Required empty public constructor
    }

    /**
     * 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";
        }

    }
}
```

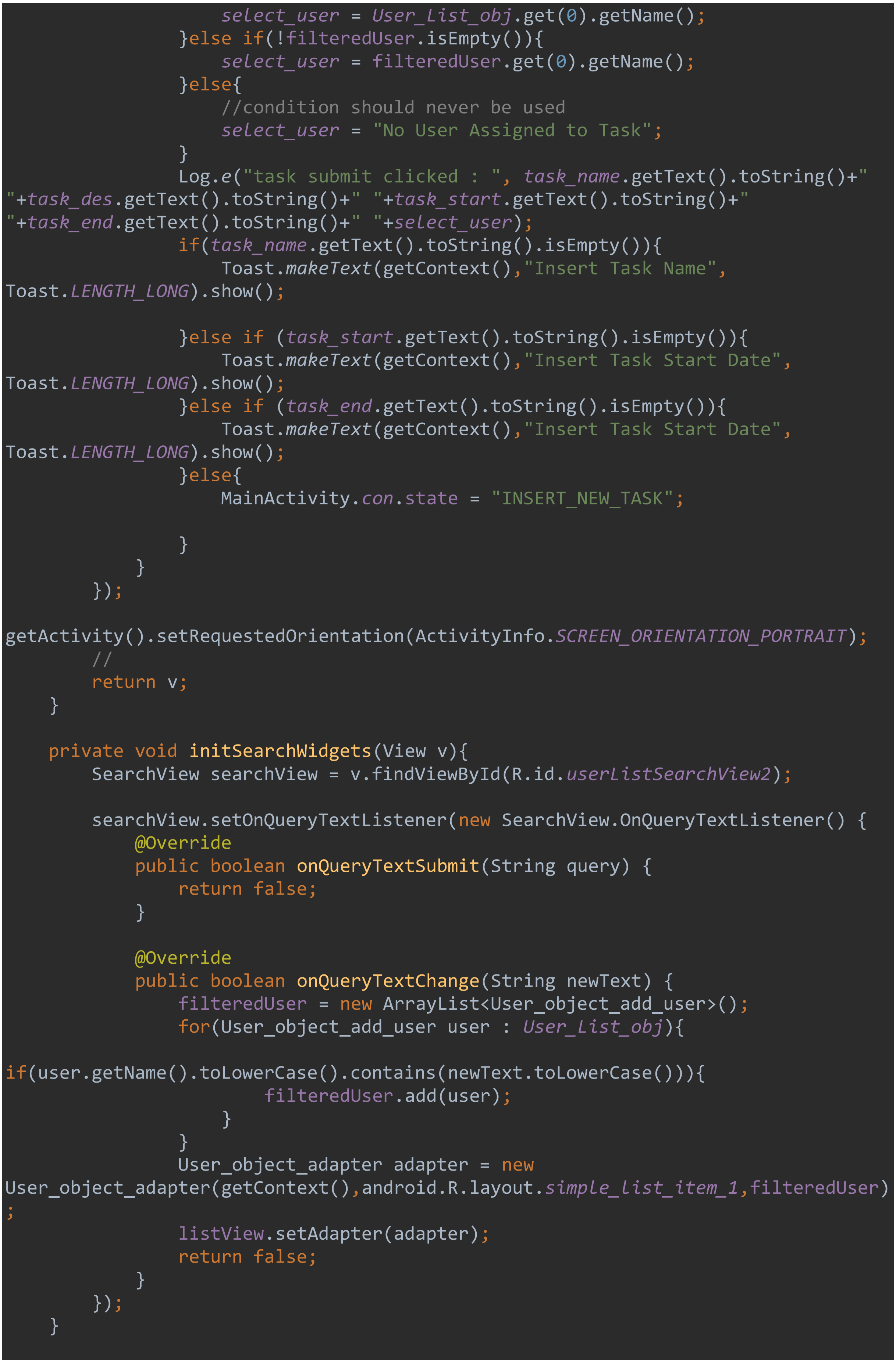
```
package com.example.app;

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;
```



```java
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();
```

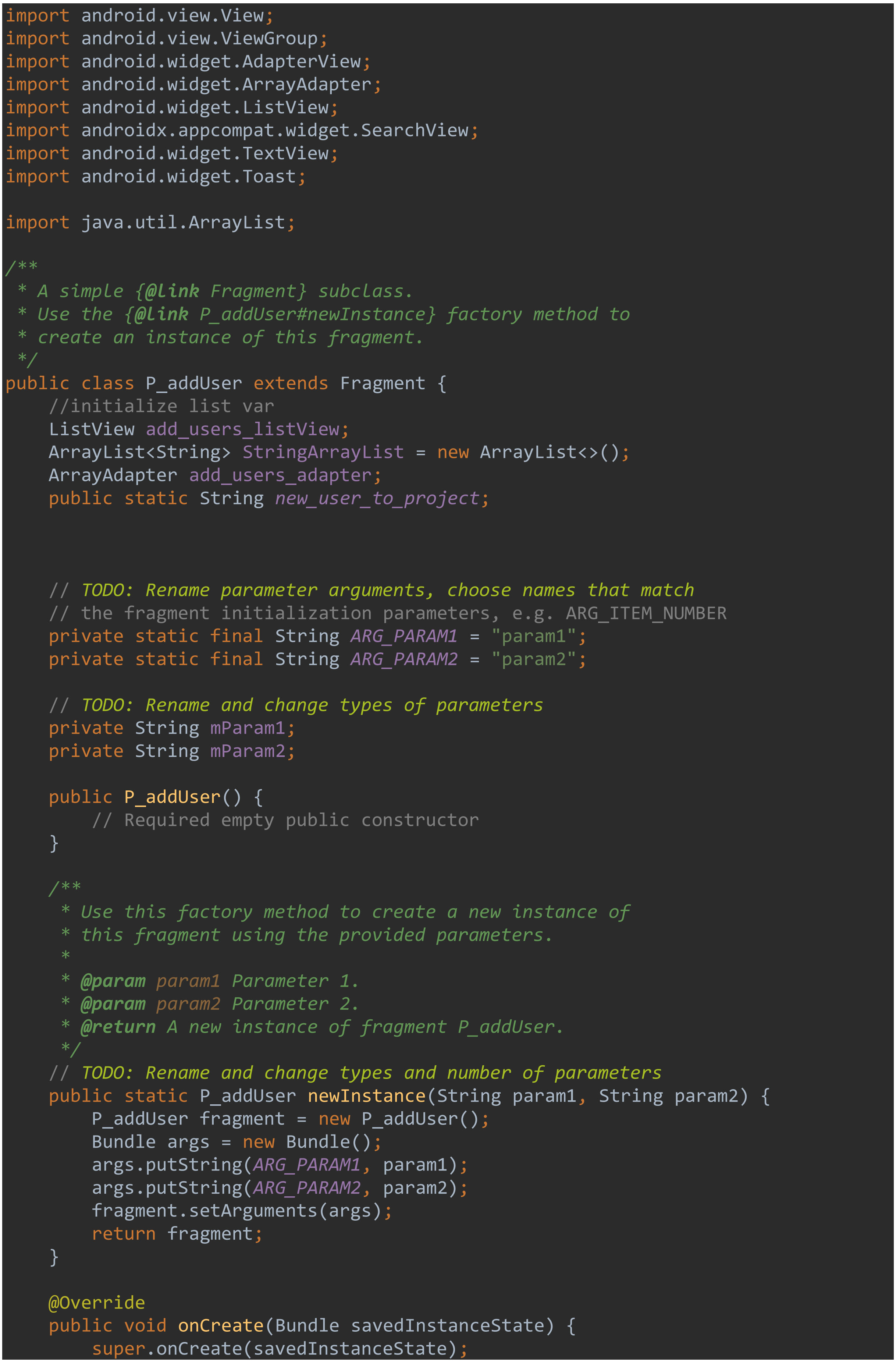


```
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @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()){
```



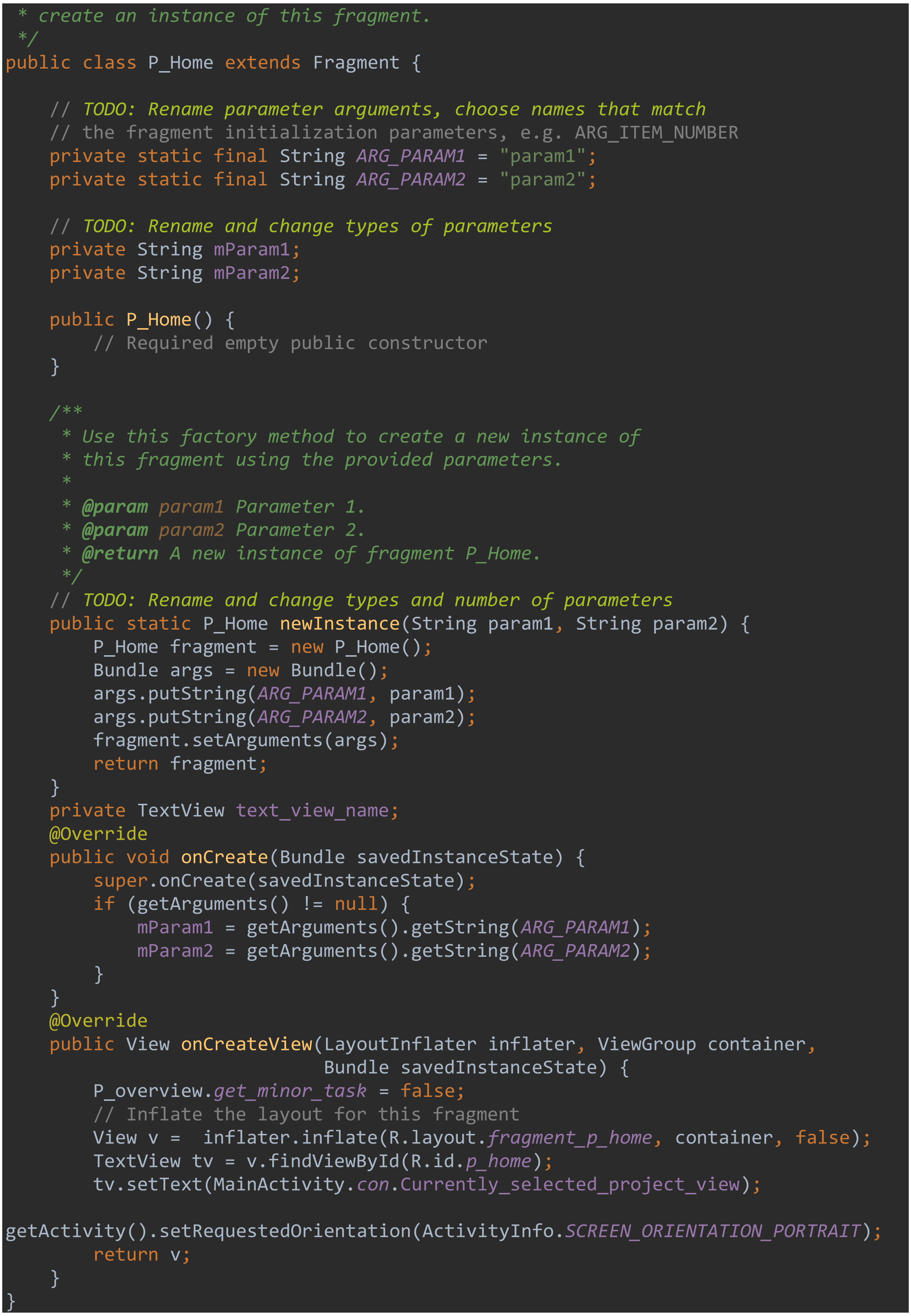
```
                    select_user = User_List_obj.get(0).getName();
                }else if(!filteredUser.isEmpty()){
                    select_user = filteredUser.get(0).getName();
                }else{
                    //condition should never be used
                    select_user = "No User 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;
```



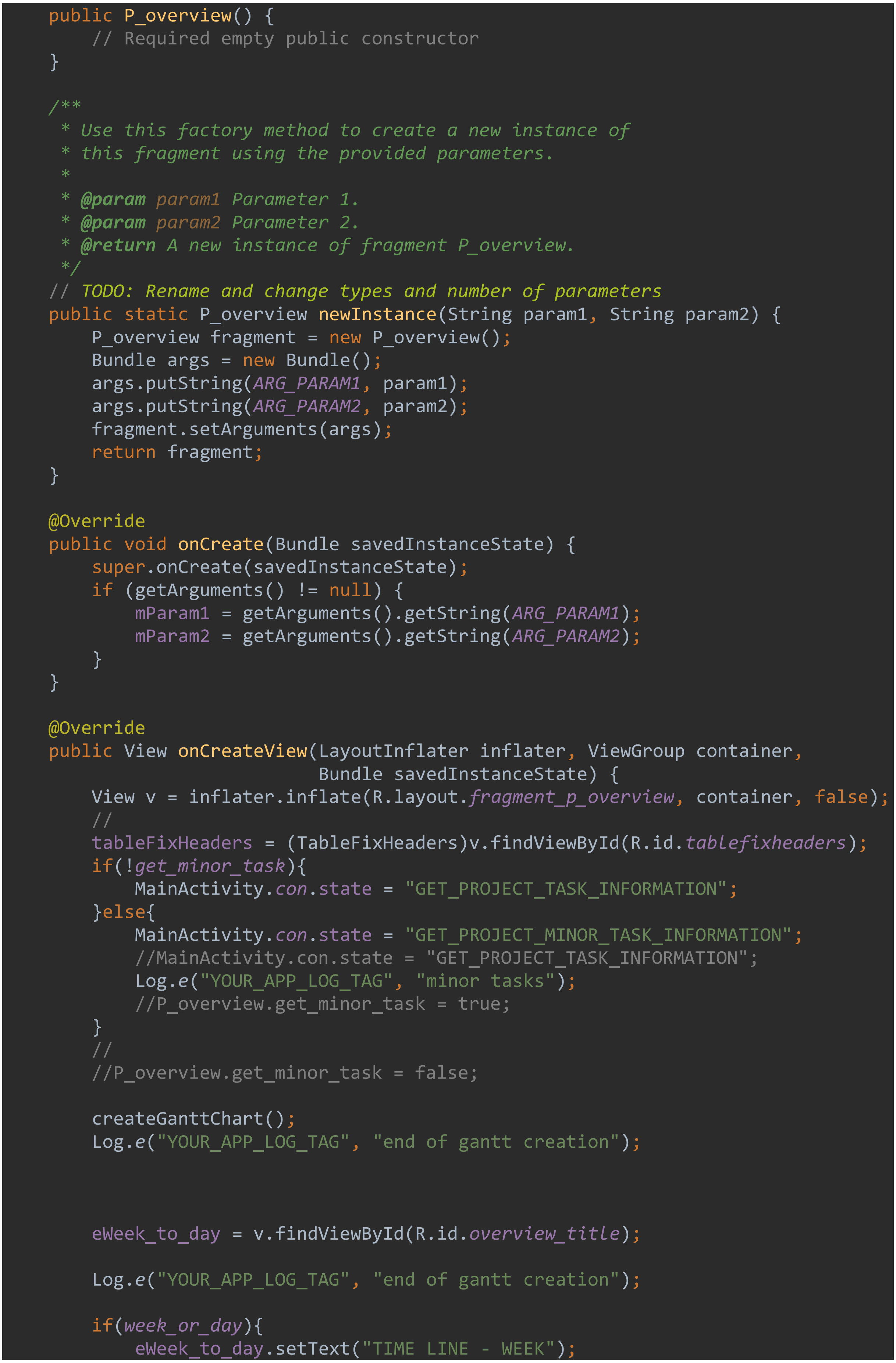
```
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import androidx.appcompat.widget.SearchView;
import android.widget.TextView;
import android.widget.Toast;

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
```

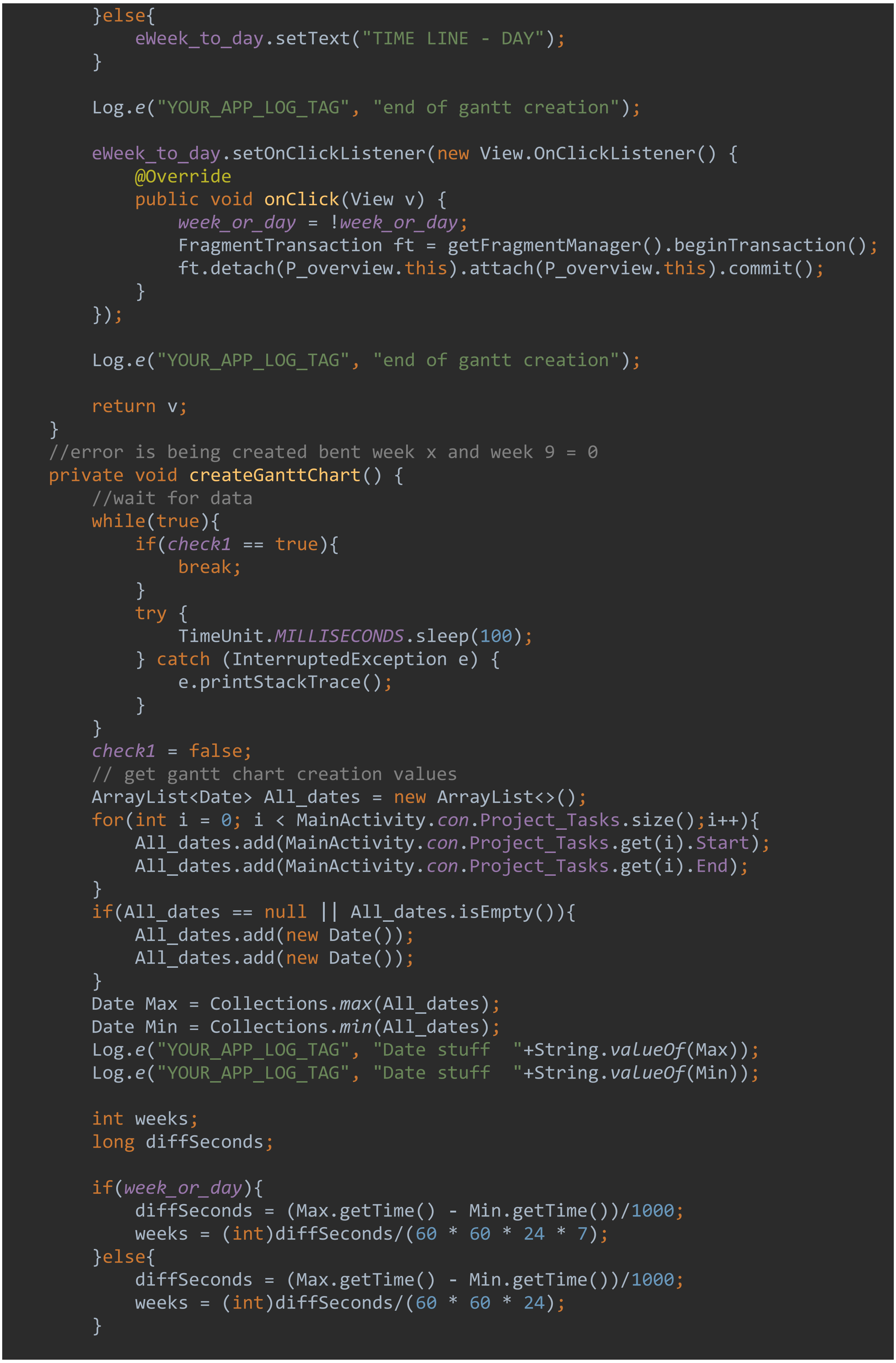


```
 * create an instance of this fragment.
 */
public class P_Home extends Fragment {

    // 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;
    }
}
```

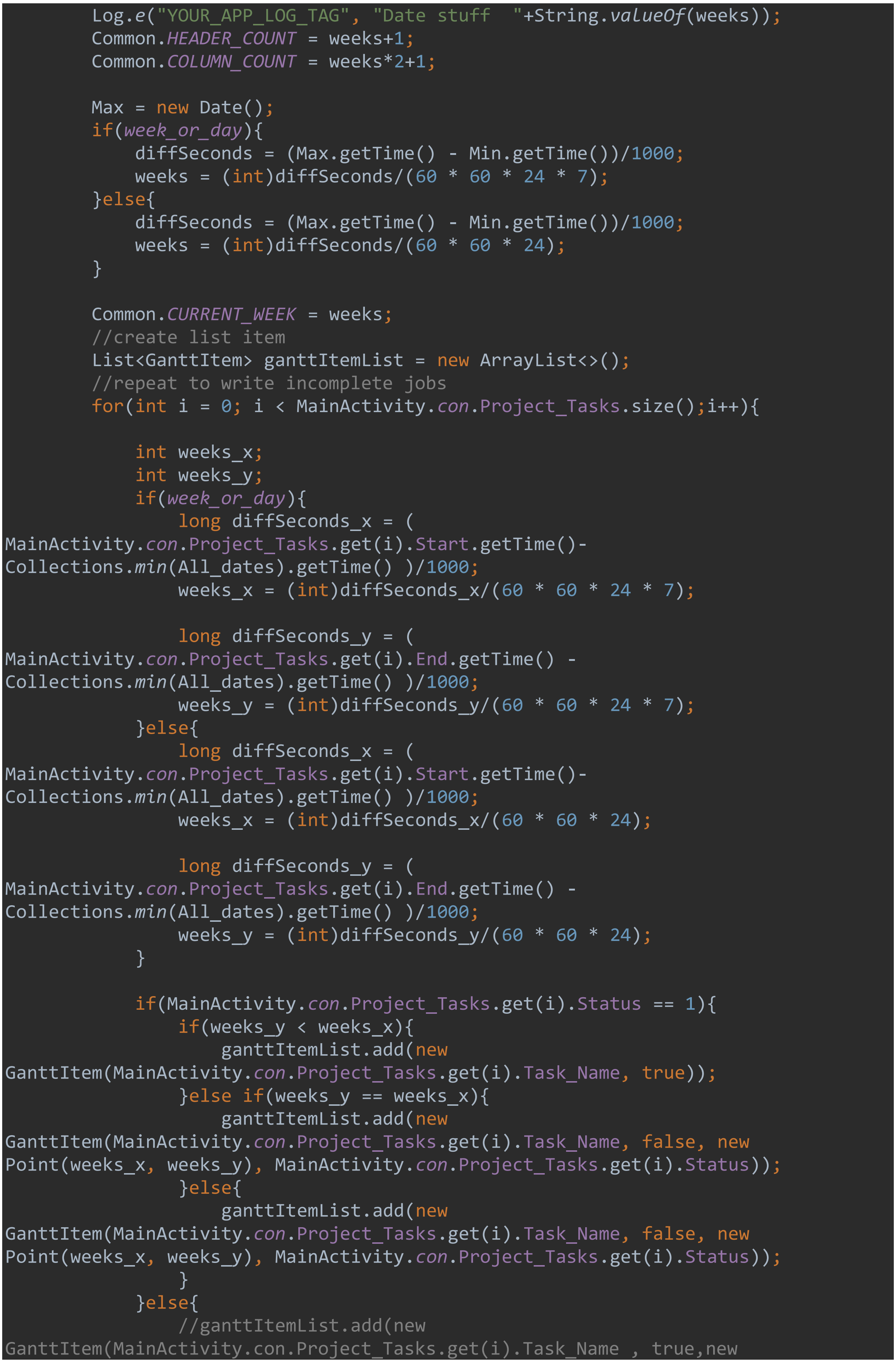
```java
package com.example.app;

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;
```

```java
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);
    }
}
```

```java
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);
    }
}
```

```java
package com.example.app;

```

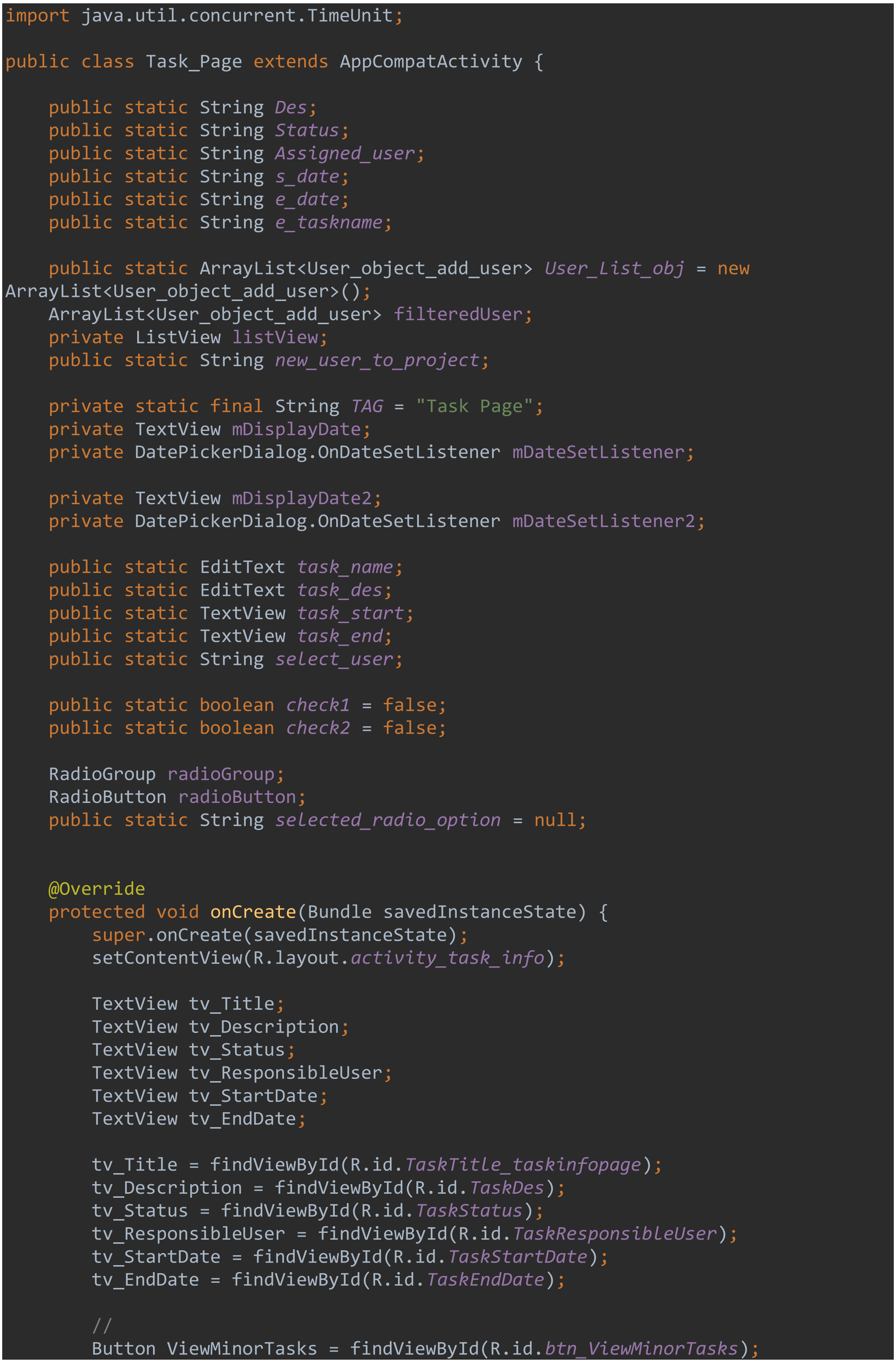
```
public class Project_Names_list {
    public String Project_names;

    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);
    }


}
```

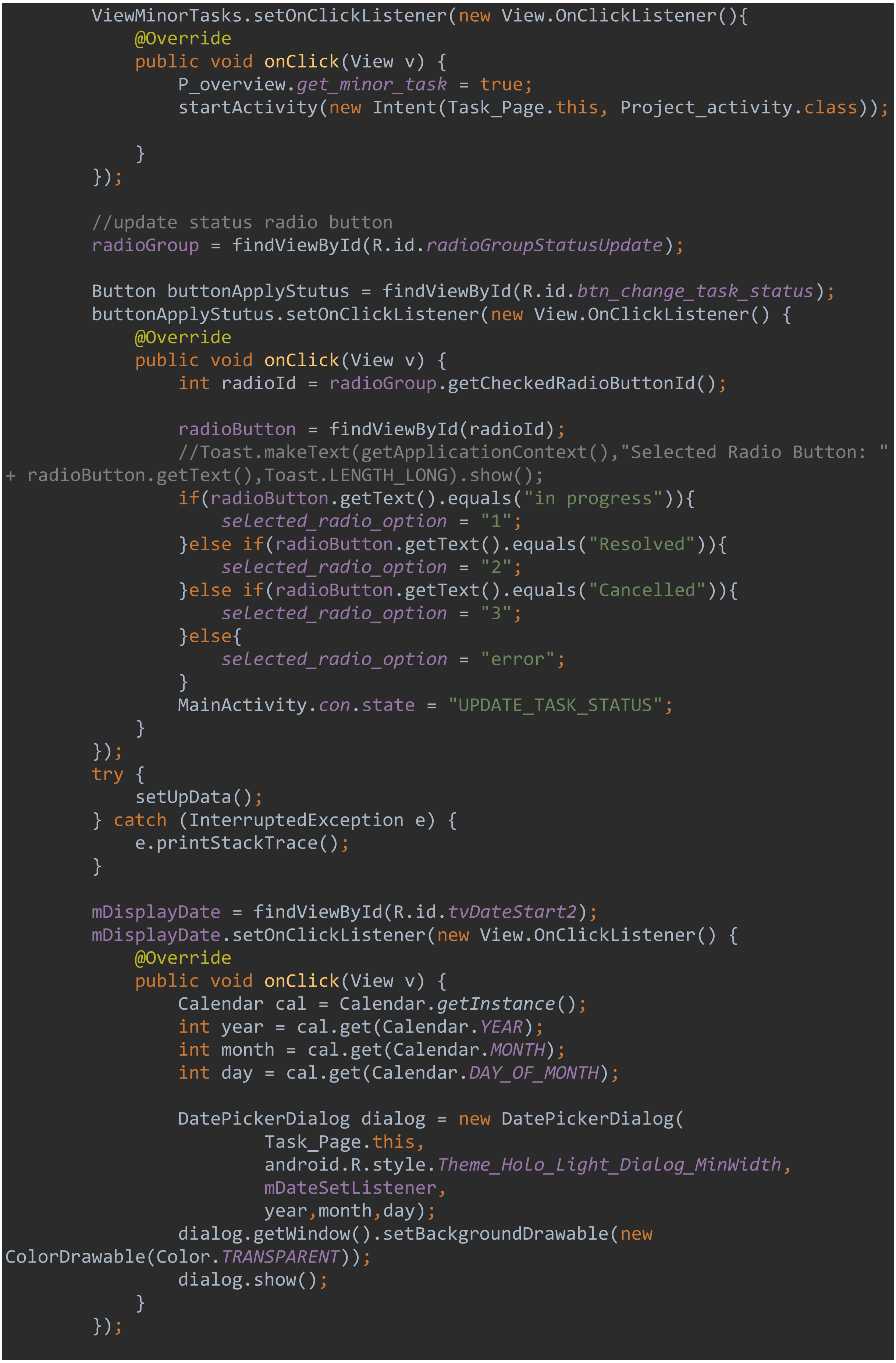
```
package com.example.app;

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;
    }
}
```

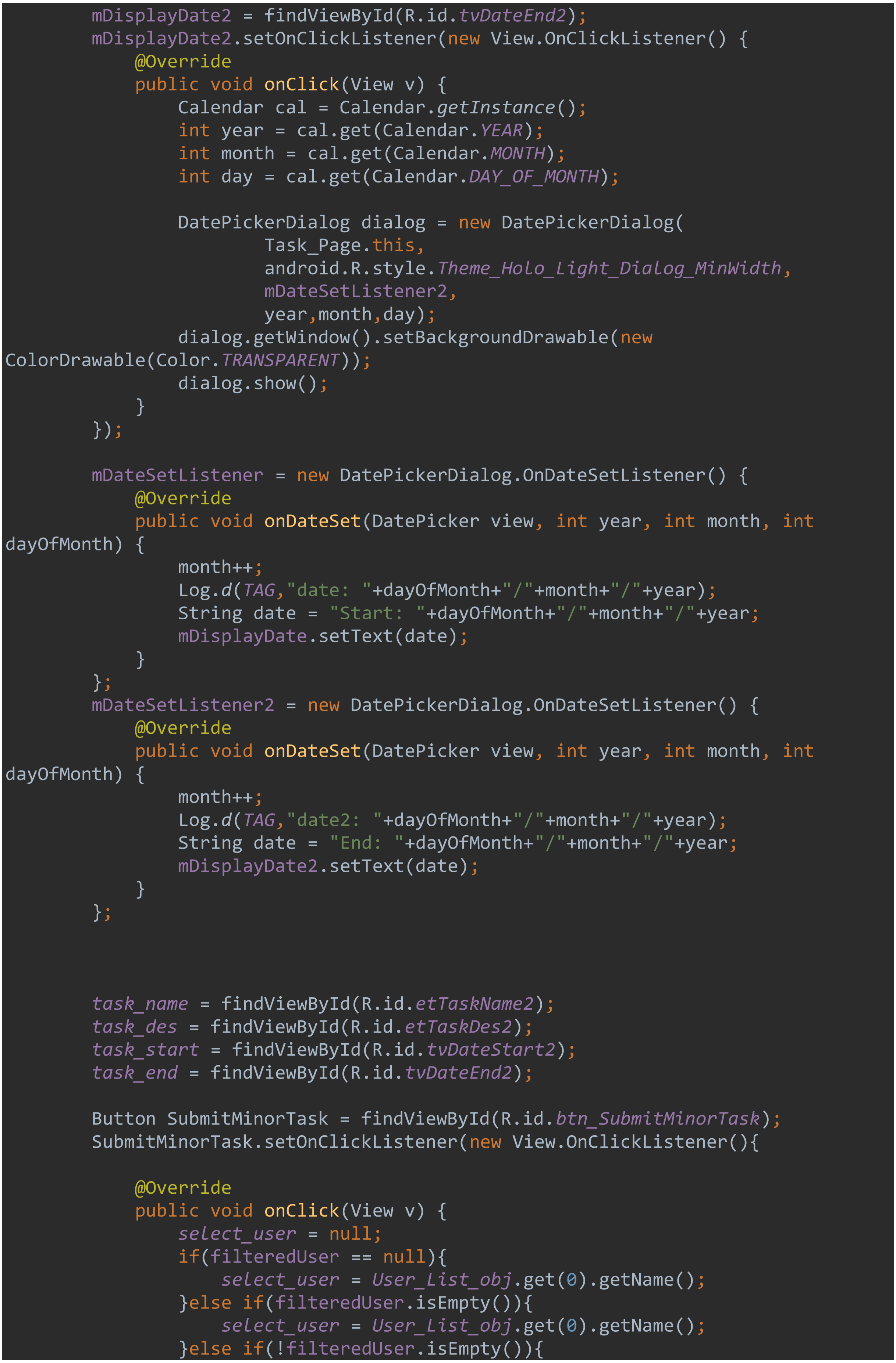
```
package com.example.app;

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;
```



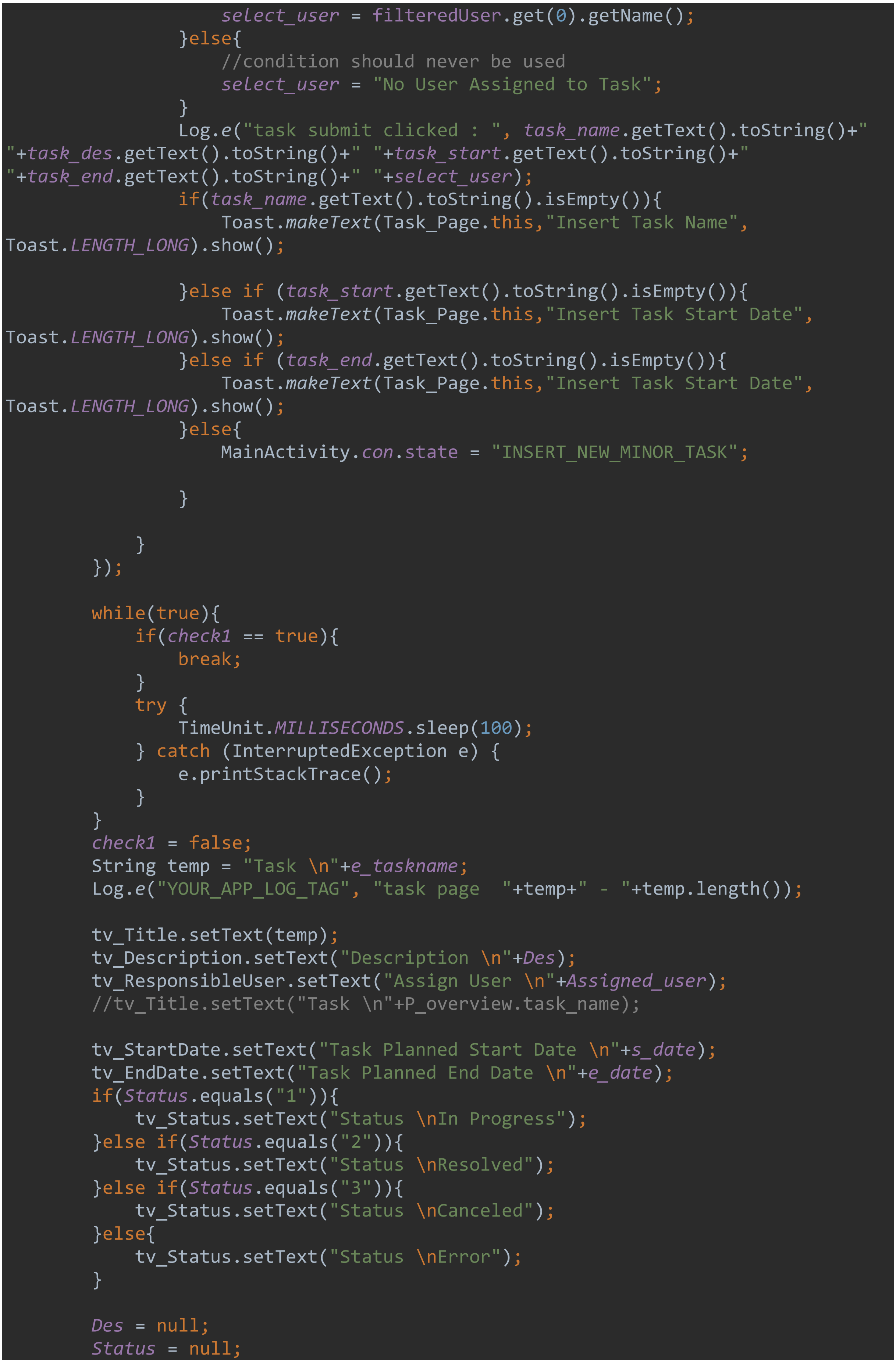
```
import java.util.concurrent.TimeUnit;

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
<?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
<?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
<?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
<?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
<?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
<?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

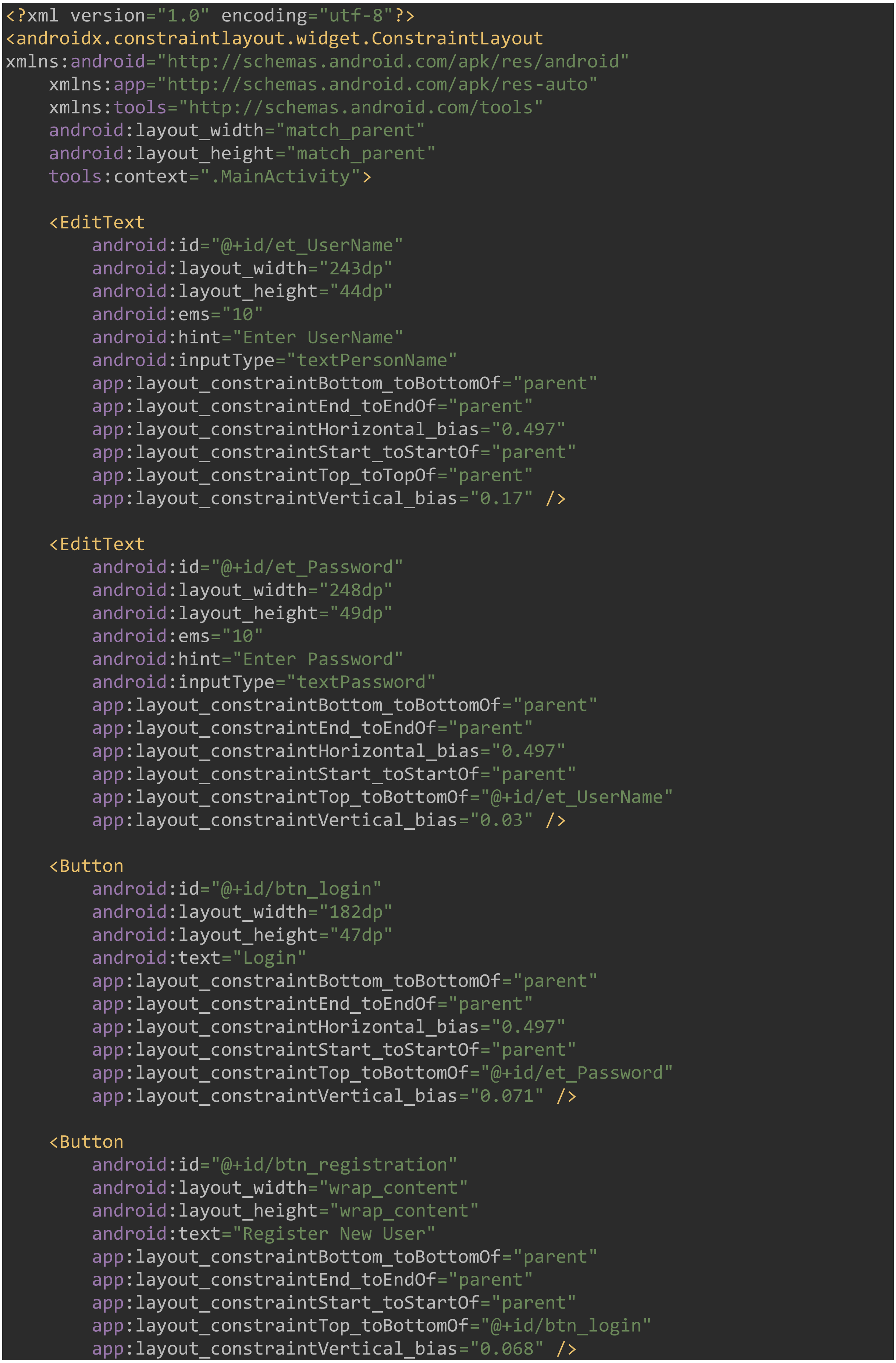
```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:aapt="http://schemas.android.com/aapt"
    android:width="108dp"
    android:height="108dp"
    android:viewportWidth="108"
    android:viewportHeight="108">
    <path android:pathData="M31,63.928c0,0 6.4,-11 12.1,-13.1c7.2,-2.6 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

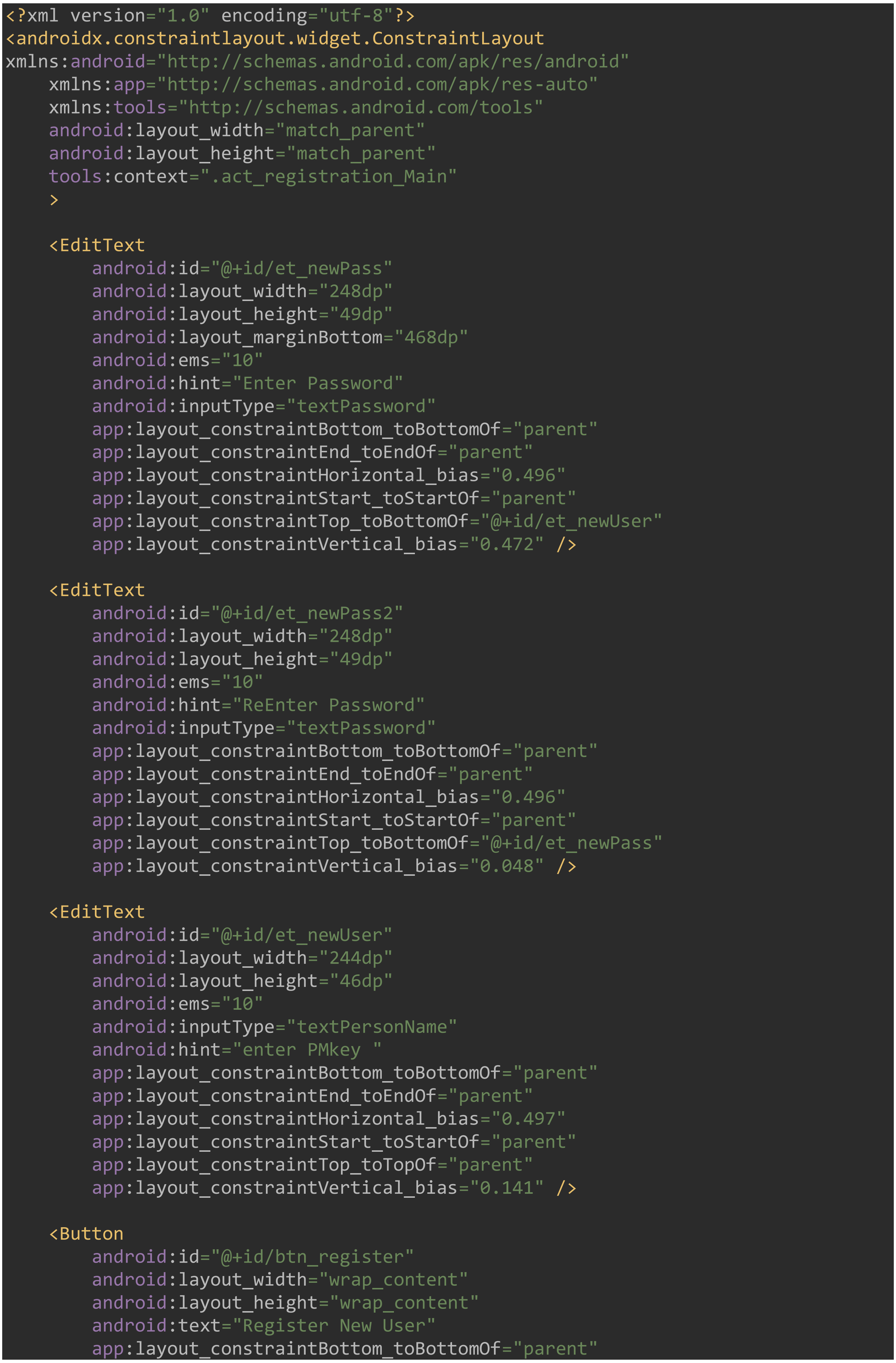


```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    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

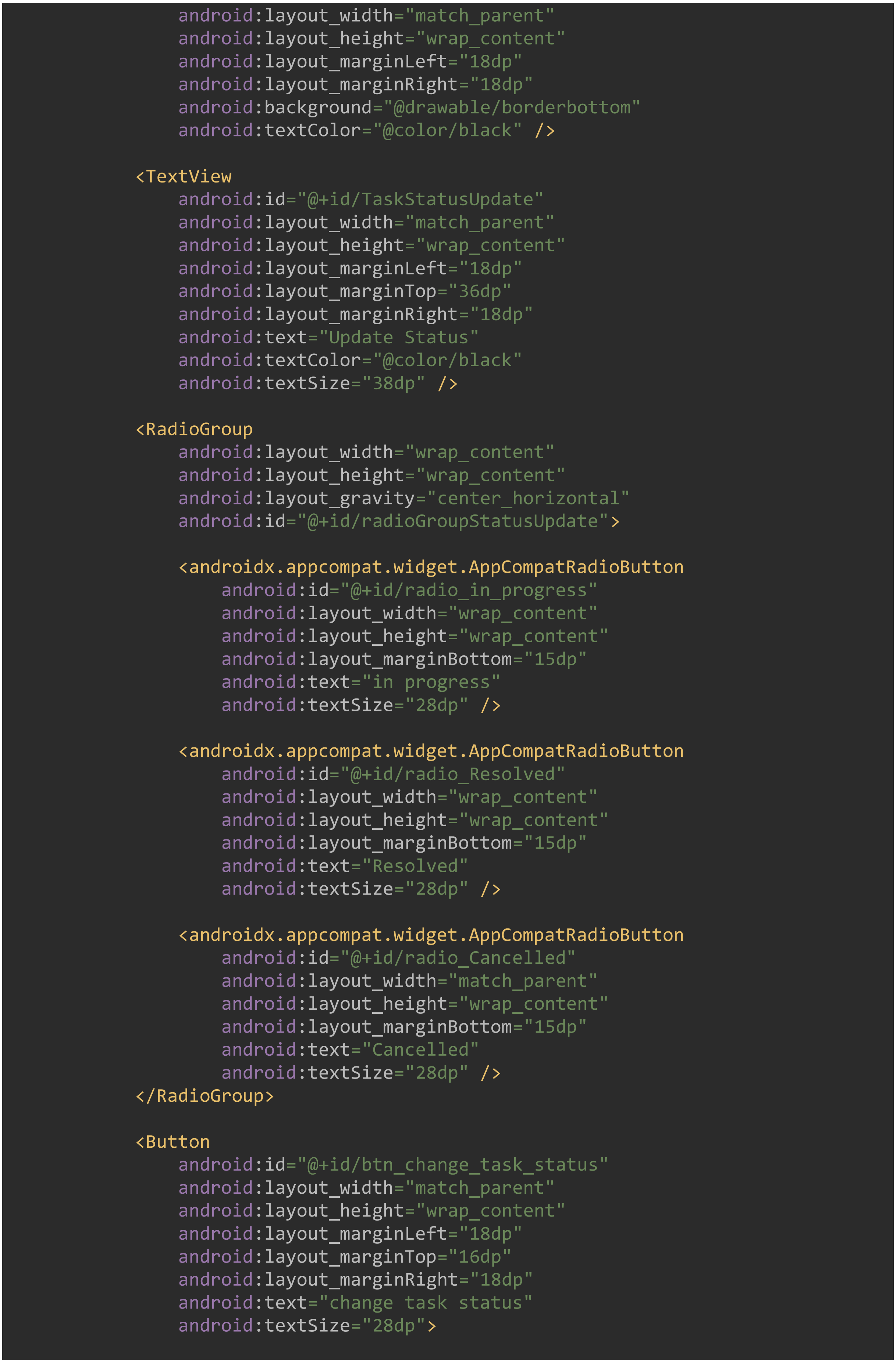


```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    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>
```

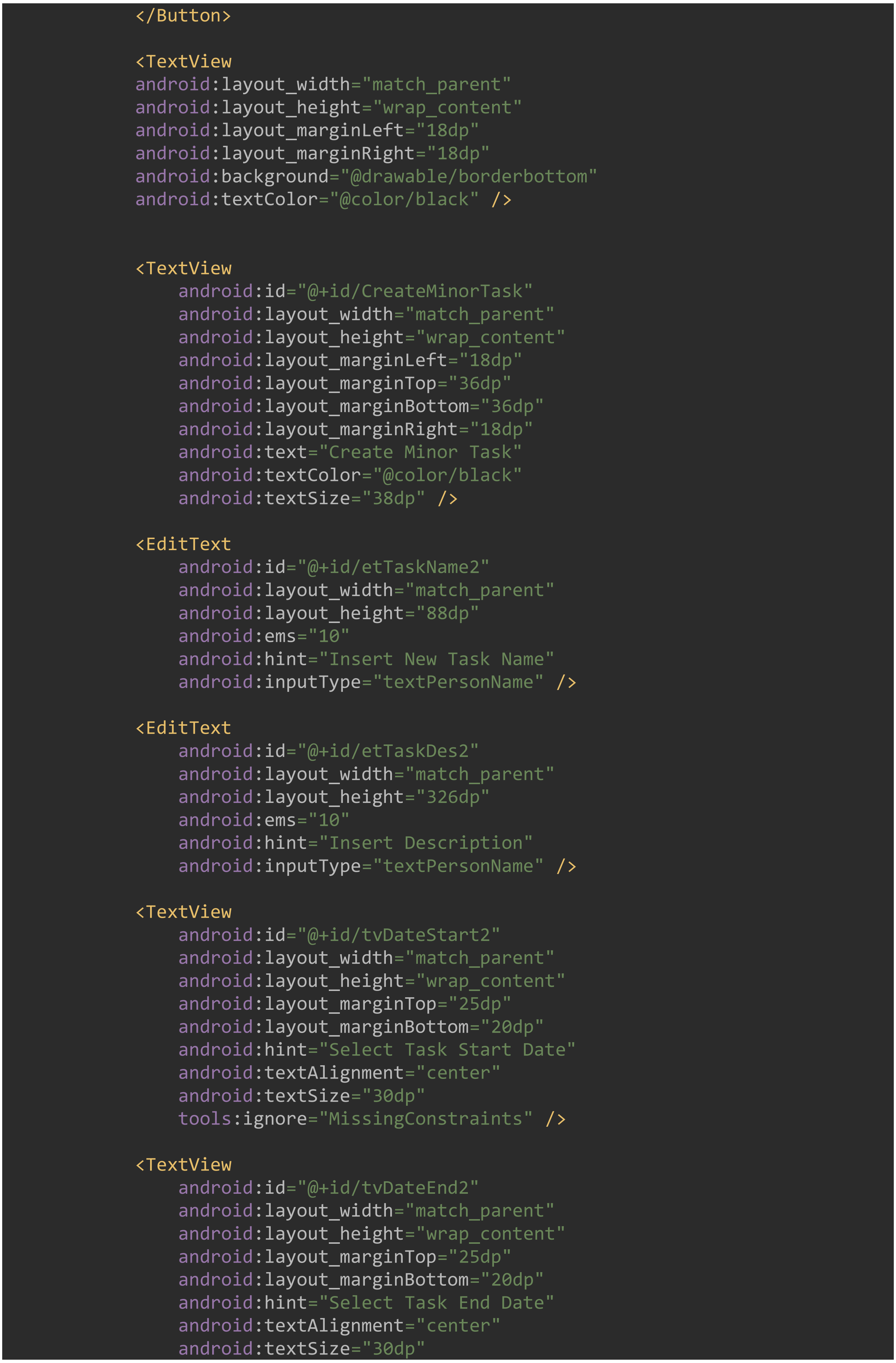
activity_task_info.xml

```
<?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"
```

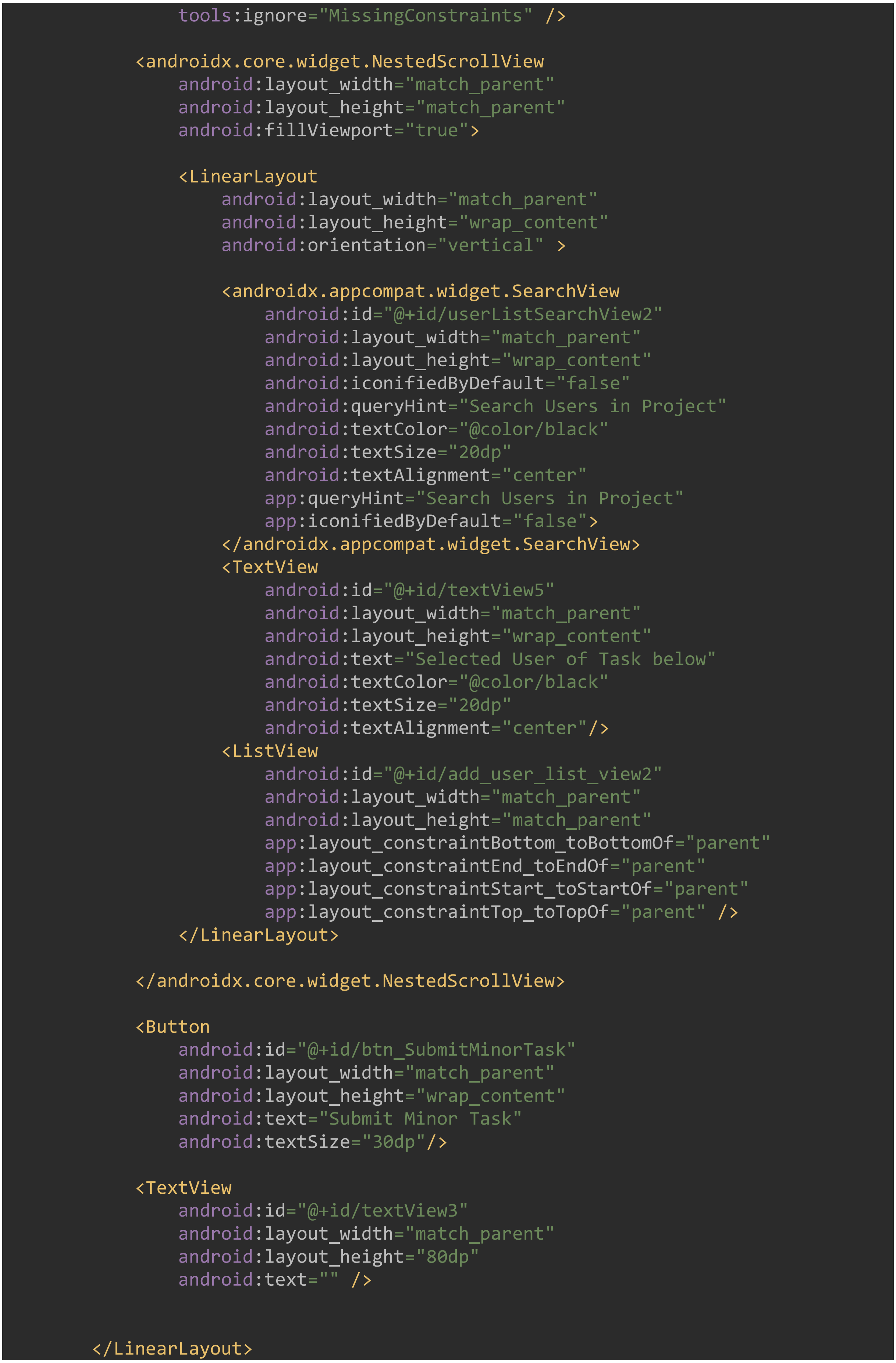
```
            android:text="Description \nA Name of a Task "
            android:textColor="@color/black"
            android:textSize="28dp" />

        <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

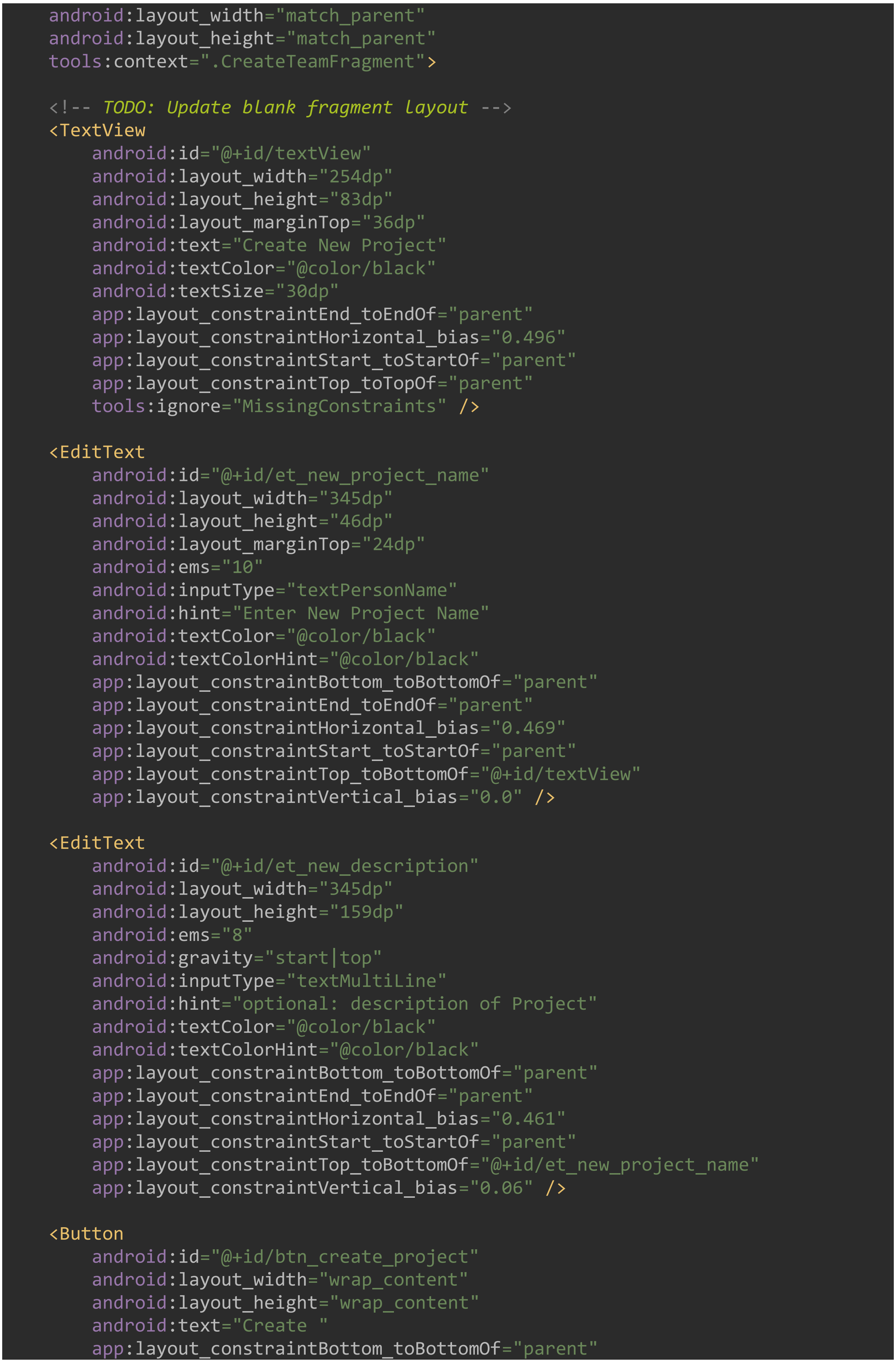
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    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"
```



```xml
    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
;
```