1.INTRODUCTION

1.1 OVERVIEW:

A headlines is the text at the top of a newspaper article, indicating the nature of the article below it. Headlines are usually written in bold and in a much larger size than the article text . Front page headlines are often in upper case so that they can be easily read by the passing potential customer.

A decade ago, we would pick up a newspaper every morning to read the headlines. Now, many of us are accustomed to rolling over in bed and grabbing our phones to checking for the most up to date happenings in the world.

This dependency on apps and perhaps most prominently, social media, offers a fresh perspective on events as they happen.

2.2 PURPOSE:

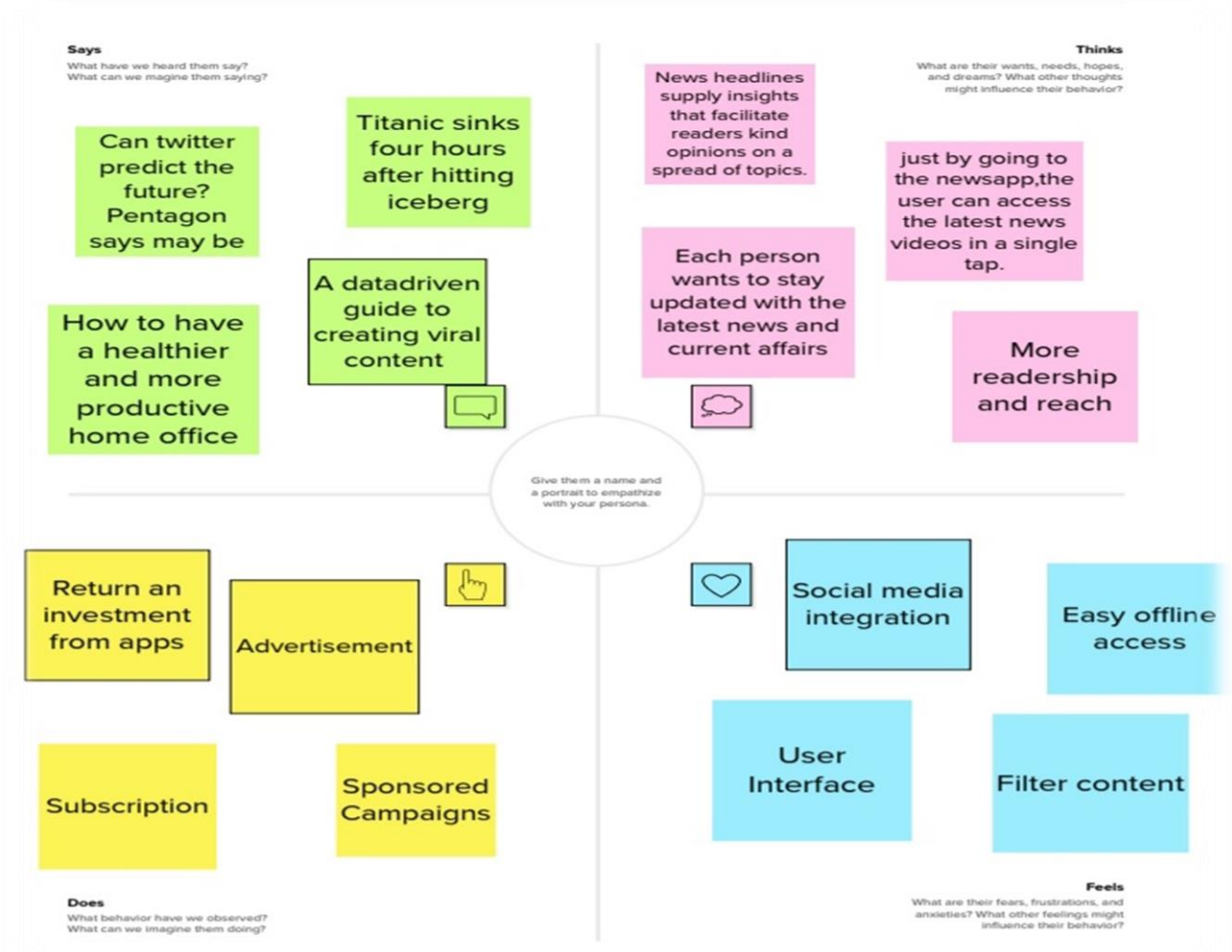
The purpose of a headline is to 'tell and sell' a story. A headlines purpose is to quickly and briefly draw attention to the story. It is generally written by a copy editor, but may also be writer, the page layout designer, or other editors.

Great headline writers know that there are several ways to do this. For instance, when iconic news stories break, articles can become representations of these moments in history.

This is done when a newspaper captures the general feeling of the public after such as event.

2. PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy map:



2.2 Brainstrom:

VIDYA

User interface	Filter content	Easy offline access
Premium content	Strengthens reading and writing skills	Provides entertainment and sports news
Improves vocabulary skills	Makes a good speaker	Get up to date with politics

ANJANA

Best source of general knowledge	Useful ideas for researches and projects	News apps are popular
People prefer mobile apps to websites	News app offers offer a better user experience	News app offers more personalization
News app nurtures engagement	News apps are easy to browse	News app are faster

VARSHA

News app is easy to monetize	stories, images, videos, ads, pop-ups	"digestible" feeds and chunks of headlines and stories.
Pick and save their favorites	Submit their own content	Access exclusive stories
Earn loyalty points and rewards	a news app allows you to collect data about the user's interests and behavior	people have a habit of using news apps to get their daily dose of information.

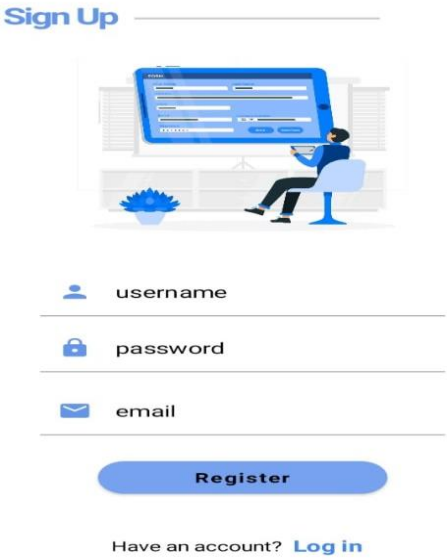
VINI

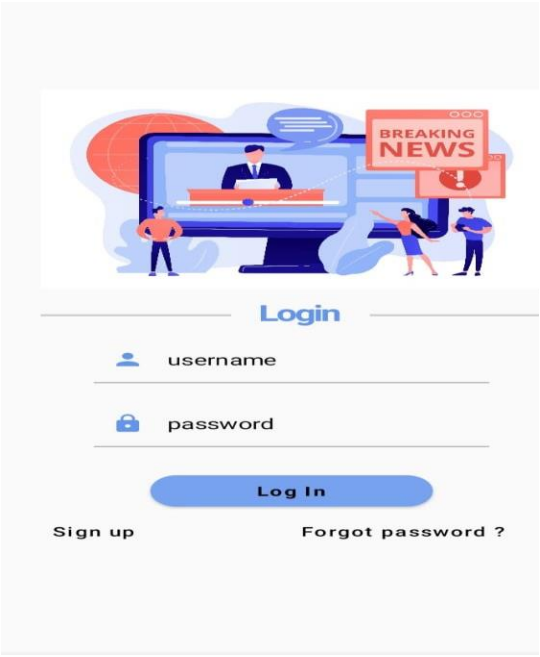
41% of mobile users' digital time is dedicated to news apps.	news app is a significant investment	Mobile apps are more convenient
Good user interface	Push notifications	Simple navigation
Access podcast	Change is needed and publishers must embrace mobile.	user can access to the content offline

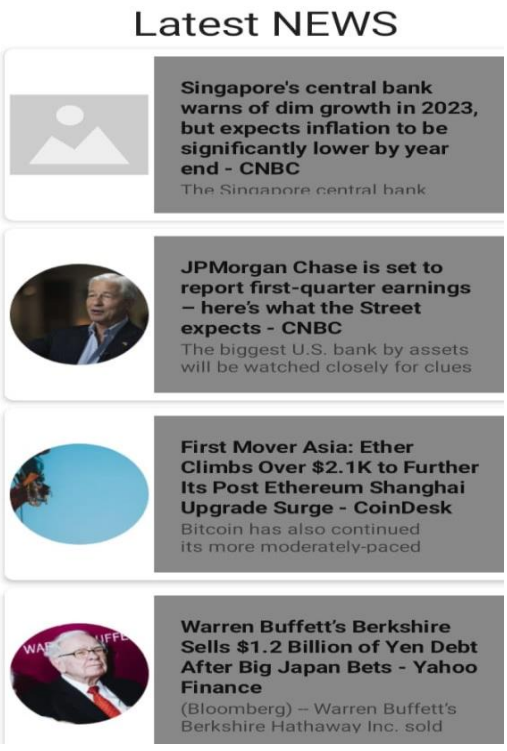
3.RESULT

OUTPUT:

REGISTER ACTIVITY & LOGIN ACTIVITY:







4.ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

In this changing and transforming world,each person wants to stay updated with the latest news and current affairs.

Here are some of the advantages of mobile news apps:

Automatic updates:

The user gets automatic updates of the latest news on the app and so does not have to refresh the app window again and again.

Video coverage:

The mobile news app provide the user with the latest news videos and coverages on the mobile phone.

Push notifications:

The mobile news apps provide the users with push notifications as well.

DISADVANTAGES:

With the ease of access and instant updates come some negative effects of Digital News Media.

Fabricated content/hoax:

A sudden wave of fake news hits a wider part of global population when the circulated through digital platforms like face book, twitter and whatsapp.

Negative content:

Accessing the internet is more convenient to everyone. It's easier for unscrupulous elements to spread negativity about a particular individual or group.

5. APPLICATIONS

News headlines have a wide range of applications, including:

1. Keeping people informed: The primary purpose of news headlines is to keep people informed about current events, both locally and globally.
2. Decision-making: News headlines can be used by individuals, businesses, and governments to make informed decisions about investments, policy changes, and other important matters.

3. Education: News headlines can be used as educational tools to teach people about history, culture, and other important topics.

4. Entertainment: News headlines can be entertaining, particularly when they involve celebrity gossip, sports, or other popular topics.

5. Social commentary: News headlines can be used as a means of social commentary, to highlight issues of social injustice, political corruption, or other important topics.

Overall, news headlines serve a critical role in informing, educating, and entertaining people, as well as shaping public opinion and influencing decision-making.

6. CONCLUSION

As world's technology is rapidly growing we have fast internet connection and network to instantly connect to other people.As day to day use of mobile, tablets and laptop is increasing, most of the people already have these facilities. In this fast and Information oriented world we need to stay updated with every incidents and news too. This News app is android mobile application where users can access to latest news from multiple newspapers from many countries. The main focus of this application is to collect news articles from all around the world and deliver it to user as fast as possible in best visualized way.

This is a fast evolving world with modernization taking a toll on the world, people nowadays prefer to spend time more on their mobile rather than read a newspaper. Newsapp is an android application which makes newspaper reading much interesting and easier. The easily usable UI makes this application very user friendly and most of all saves time.

7. FUTURE SCOPE

This version of the app is in the seeding stage and there is a lot of room for future enhancements. Features that can be added for later versions include:

1. Include video

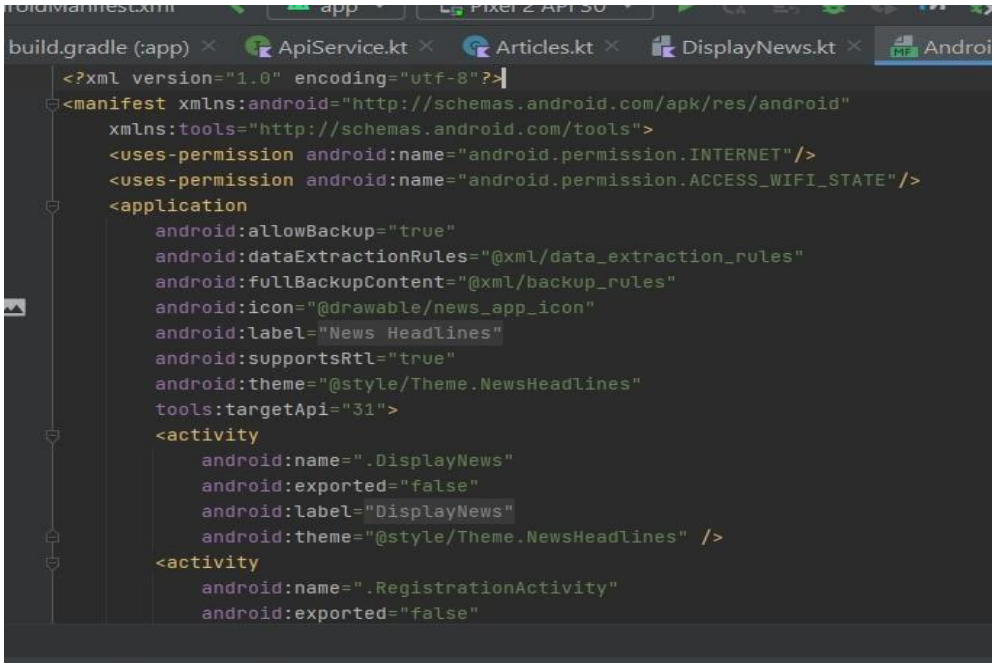
2. Read this later

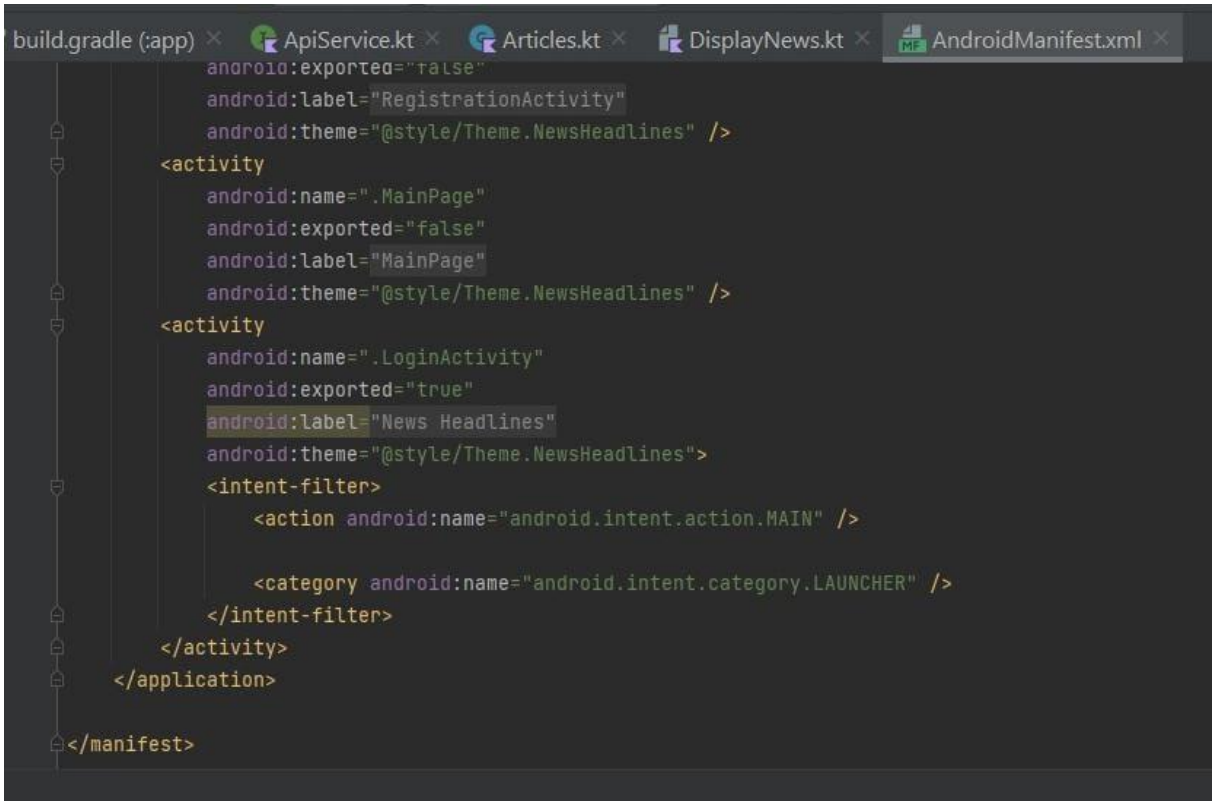
3. Notification for updates on a particular article.

Location feature with automation can also be implemented which means as user move from one city to other local news can be made to change as per the location. Offline Reading can be improve will more efficient way on full articles. Data quality check may be needed. If API can’t reach to certain article source it gives null value which can cause problem in JSON parsing.

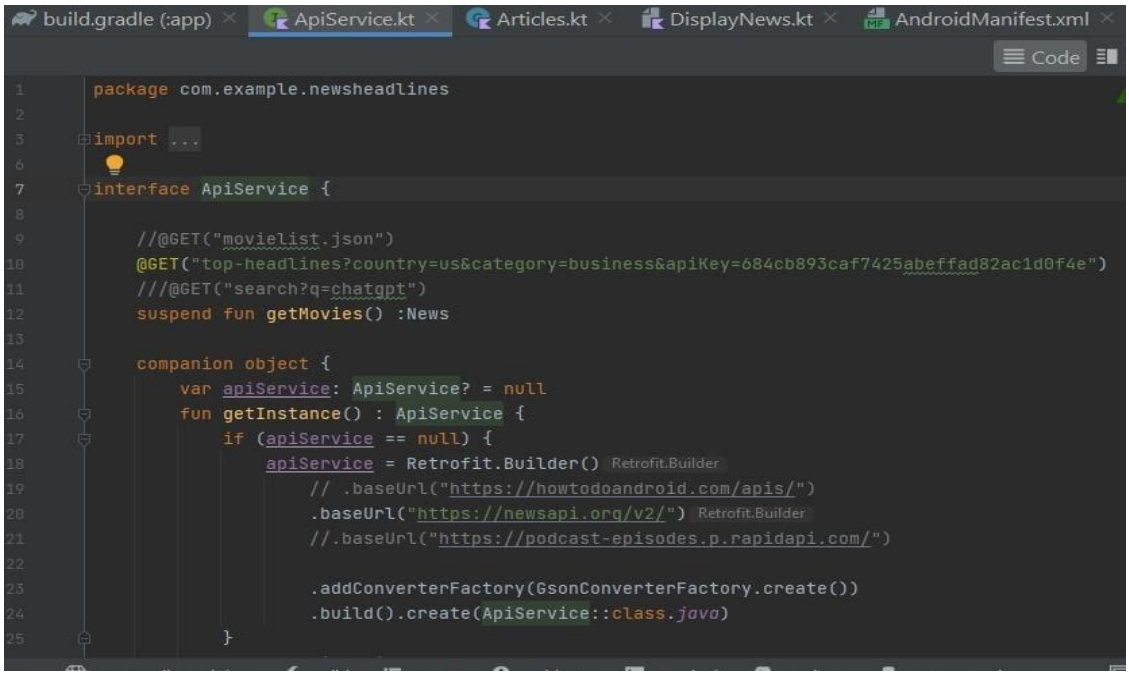
8. APPENDIX

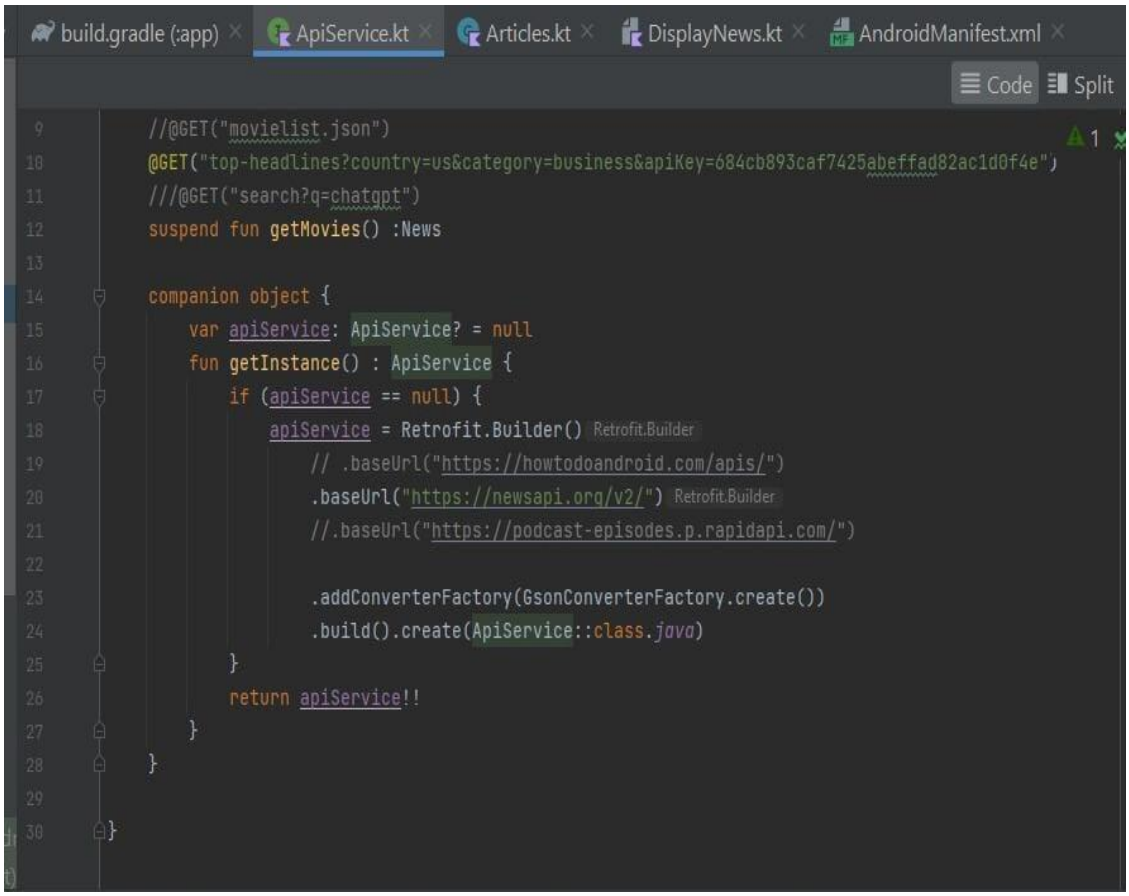
Manifest.xml



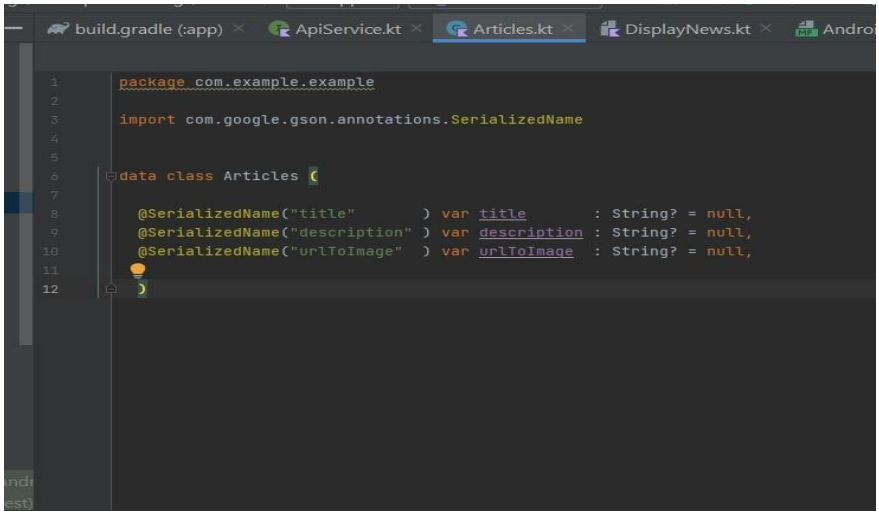


ApiService.kt

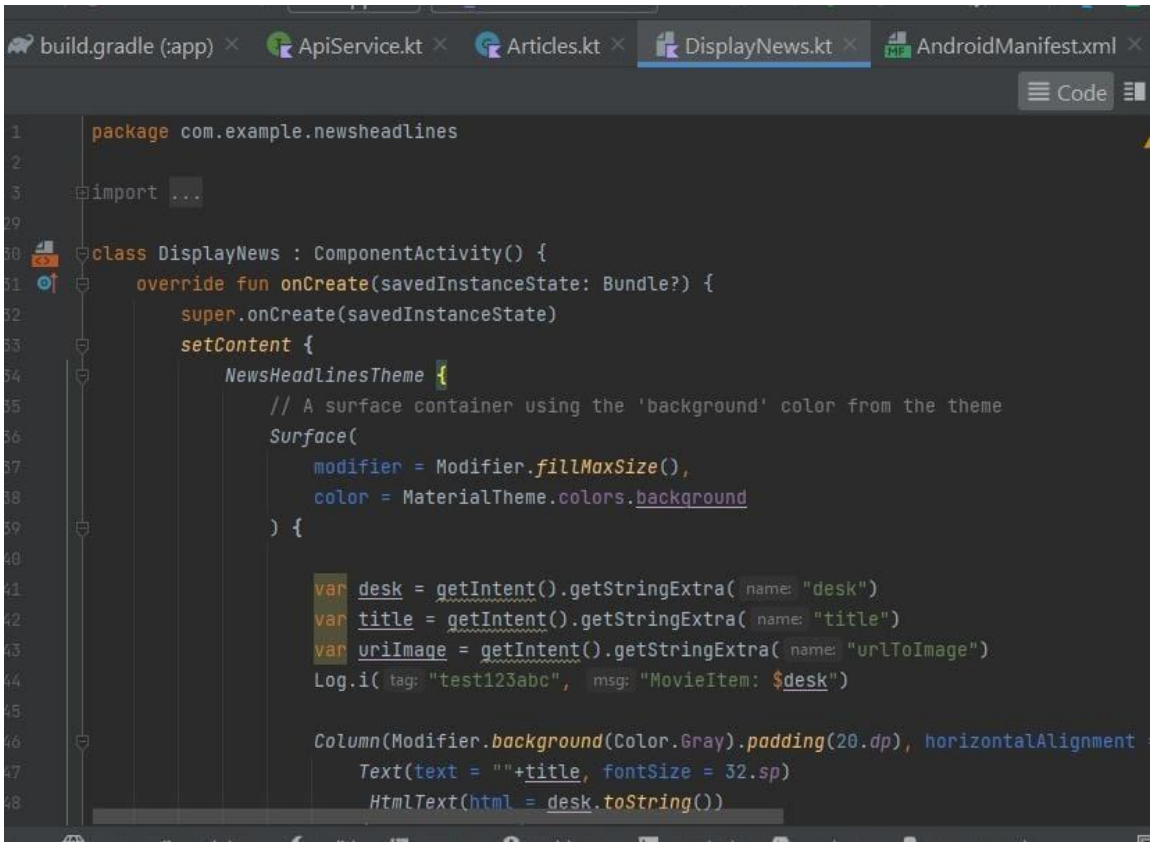

build.gradle (:app)
ApiService.kt
Articles.kt
DisplayNews.kt
AndroidManifest.xml
Code
package com.example.newsheadlines
import ...
interface ApiService {
//@GET("movielist.json")
@GET("top-headlines?country=us&category=business&apiKey=684cb893caf7425abeffad82ac1d0f4e")
///@GET("search?q=chatgpt")
suspend fun getMovies() :News
companion object {
var apiService: ApiService? = null
fun getInstance() : ApiService {
if (apiService == null) {
apiService = Retrofit.Builder() Retrofit.Builder
// .baseUrl("https://howtodoandroid.com/apis/")
.baseUrl("https://newsapi.org/v2/") Retrofit.Builder
//.baseUrl("https://podcast-episodes.p.rapidapi.com/")
.addConverterFactory(GsonConverterFactory.create())
.build().create(ApiService::class.java)
}

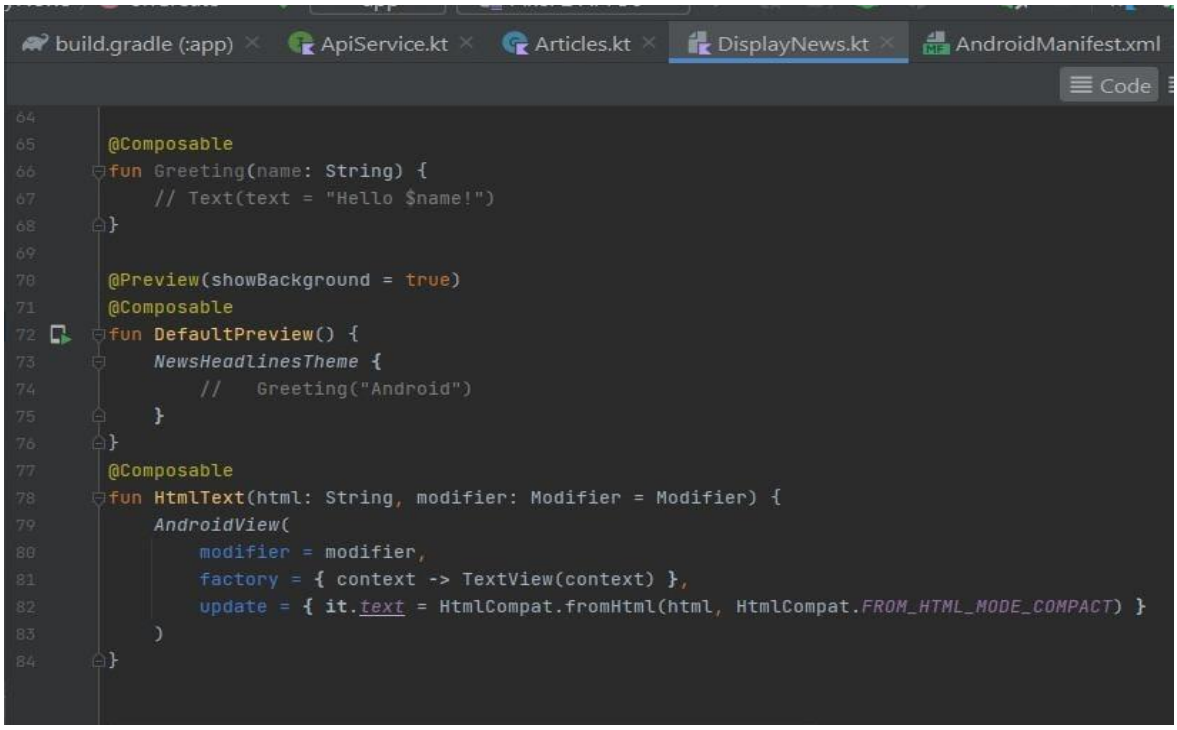

build.gradle (:app)
ApiService.kt
Articles.kt
DisplayNews.kt
AndroidManifest.xml
Code
Split
1
//@GET("movielist.json")
@GET("top-headlines?country=us&category=business&apiKey=684cb893caf7425abeffad82ac1d0f4e")
///@GET("search?q=chatgpt")
suspend fun getMovies() :News
companion object {
var apiService: ApiService? = null
fun getInstance() : ApiService {
if (apiService == null) {
apiService = Retrofit.Builder() Retrofit.Builder
// .baseUrl("https://howtodoandroid.com/apis/")
.baseUrl("https://newsapi.org/v2/") Retrofit.Builder
//.baseUrl("https://podcast-episodes.p.rapidapi.com/")
.addConverterFactory(GsonConverterFactory.create())
.build().create(ApiService::class.java)
}
return apiService!!
}
}
}

Article.kt

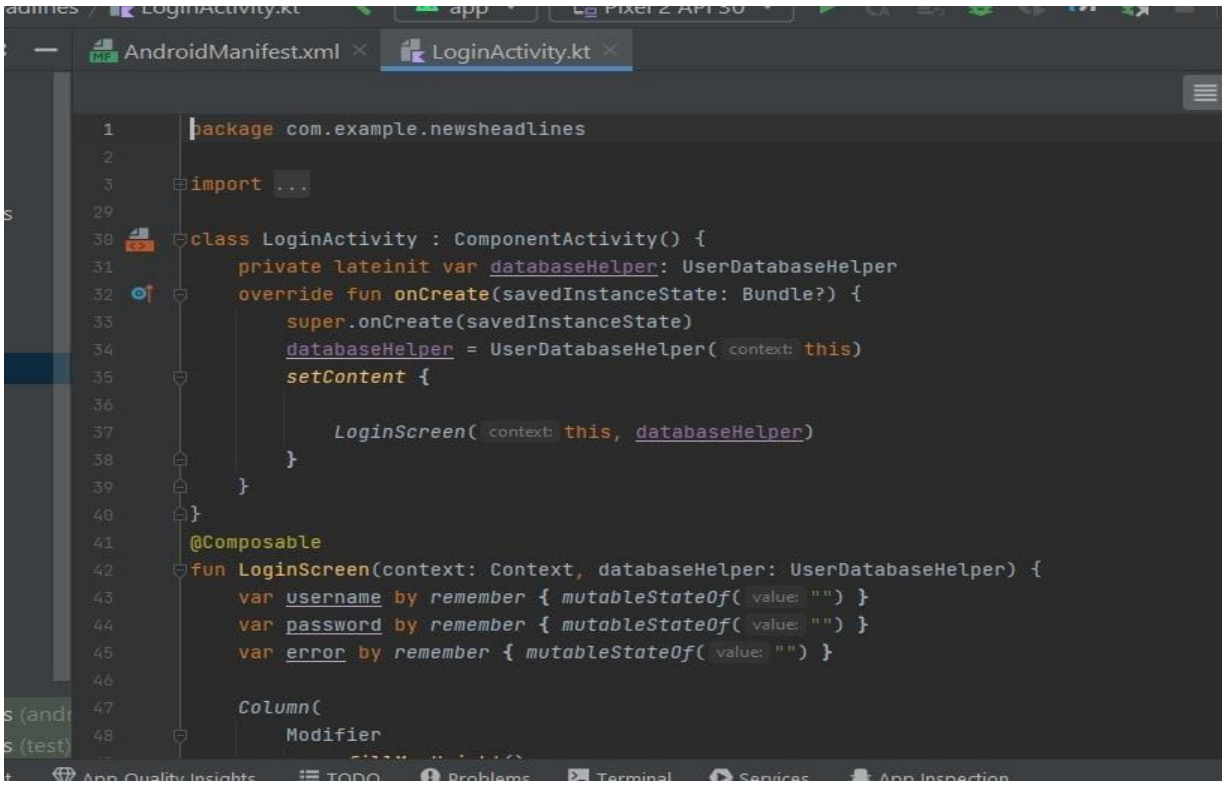


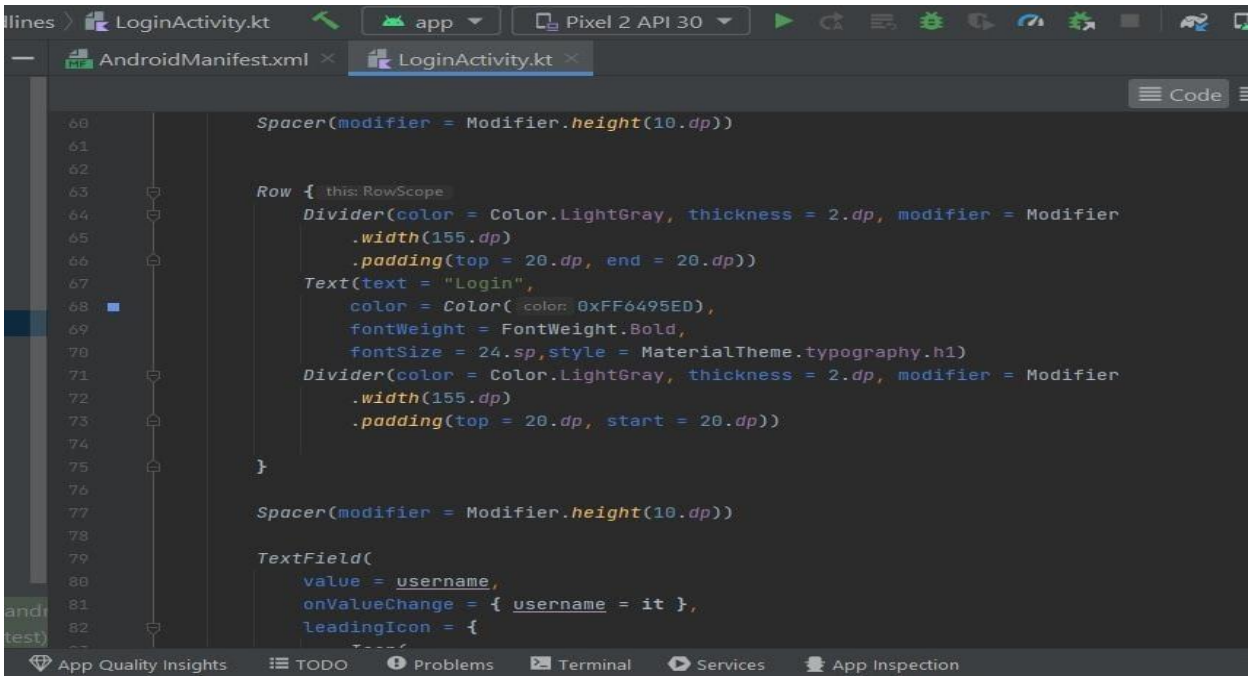
DisplayNews.kt

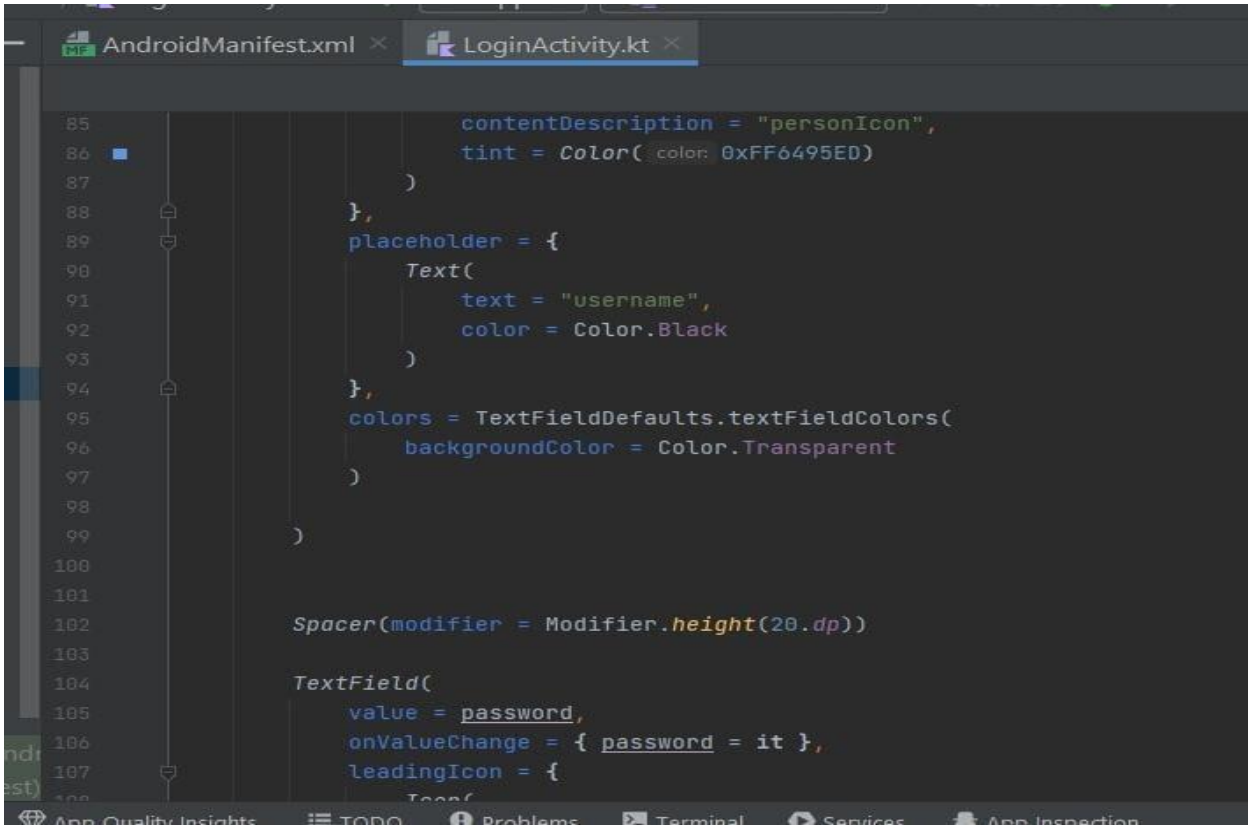


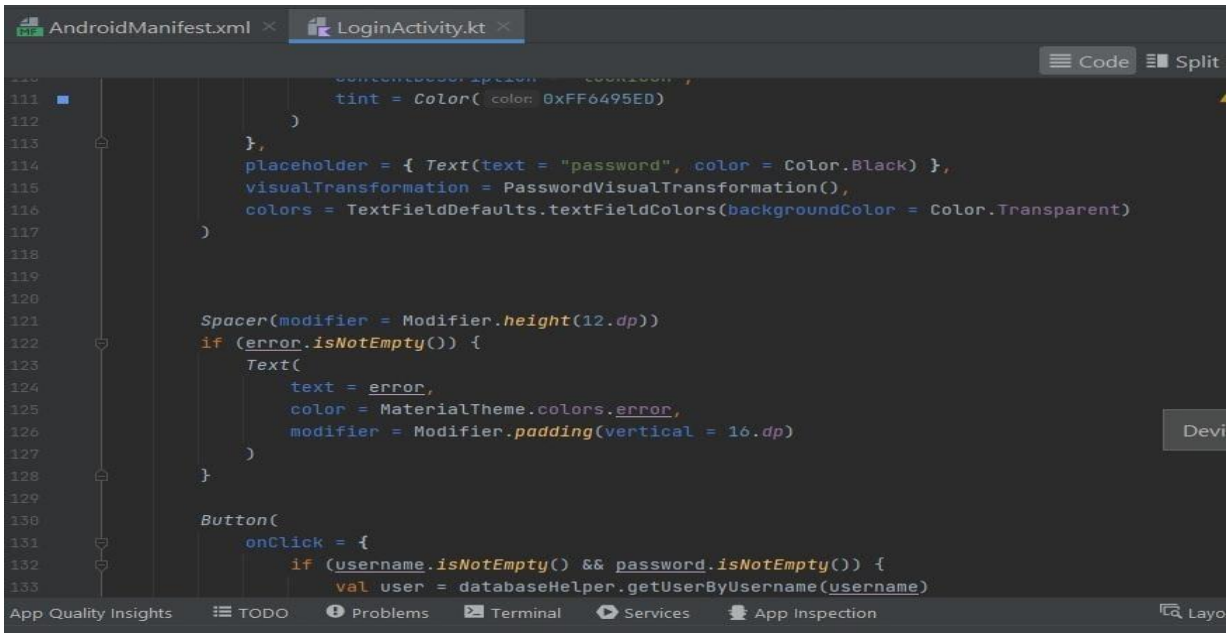


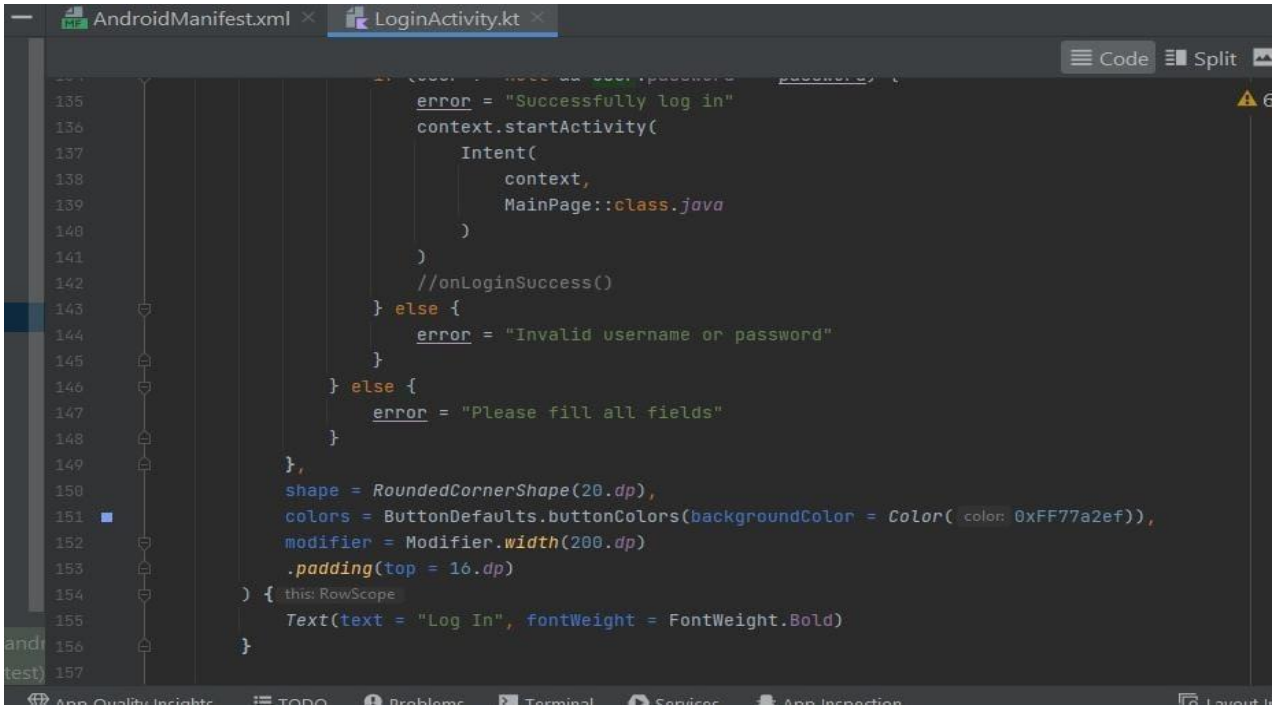
LoginActivity.kt

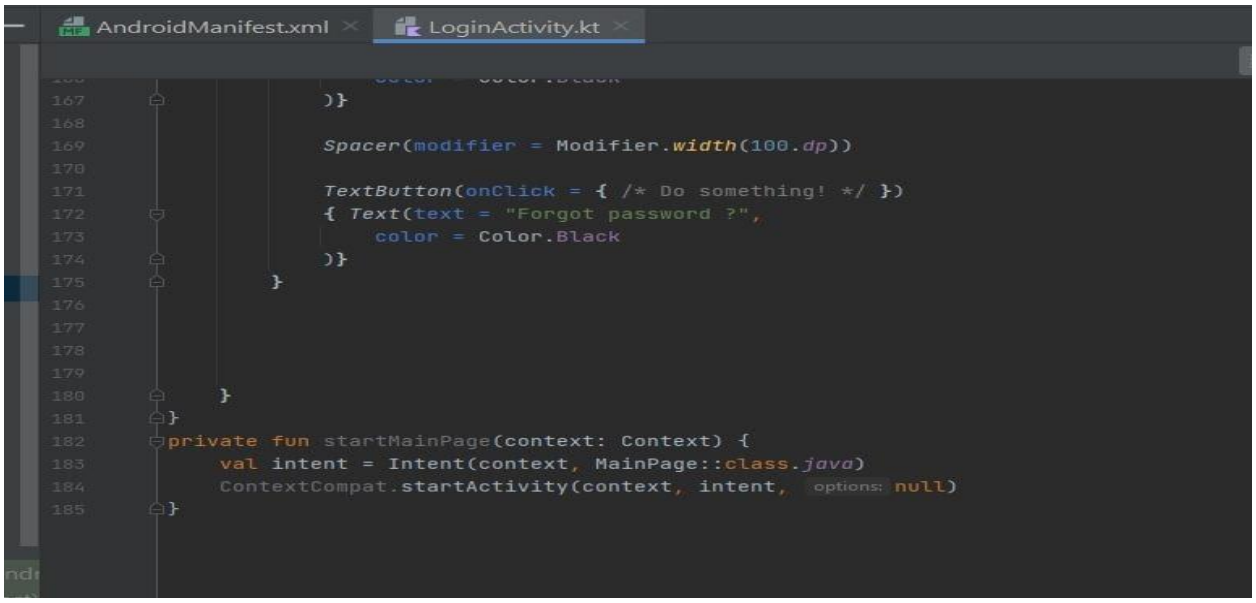



LoginActivity.kt
app
Pixel 2 API 30
AndroidManifest.xml
LoginActivity.kt
Code
Spacer(modifier = Modifier.height(10.dp))
Row { this: RowScope
Divider(color = Color.LightGray, thickness = 2.dp, modifier = Modifier
.width(155.dp)
.padding(top = 20.dp, end = 20.dp))
Text(text = "Login",
color = Color(color: 0xFF6495ED),
fontWeight = FontWeight.Bold,
fontSize = 24.sp,style = MaterialTheme.typography.h1)
Divider(color = Color.LightGray, thickness = 2.dp, modifier = Modifier
.width(155.dp)
.padding(top = 20.dp, start = 20.dp))
}
Spacer(modifier = Modifier.height(10.dp))
TextField(
value = username,
onValueChange = { username = it },
leadingIcon = {
App Quality Insights
TODO
Problems
Terminal
Services
App Inspection

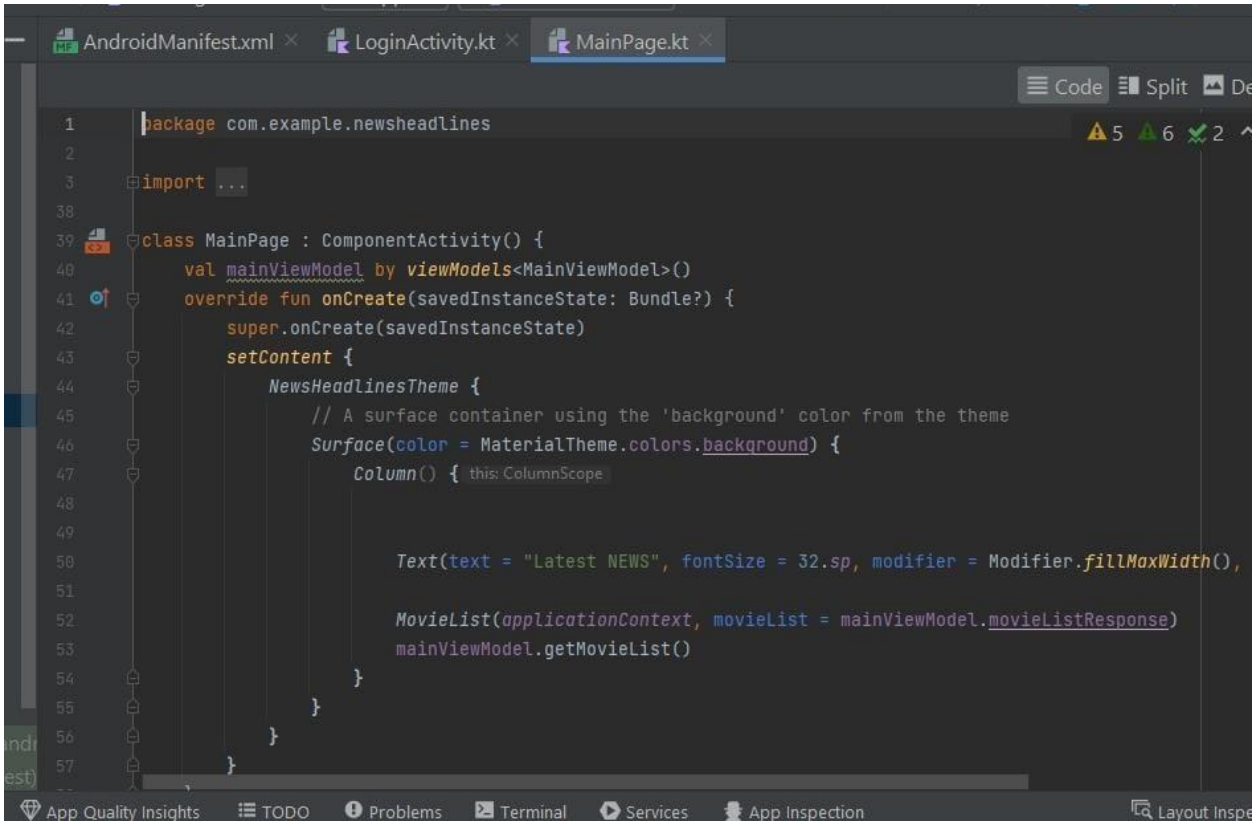

AndroidManifest.xml
LoginActivity.kt
contentDescription = "personIcon",
tint = Color(color: 0xFF6495ED)
placeholder = {
Text(
text = "username",
color = Color.Black
colors = TextFieldDefaults.textFieldColors(
backgroundColor = Color.Transparent
Spacer(modifier = Modifier.height(20.dp))
TextField(
value = password,
onValueChange = { password = it },
leadingIcon = {

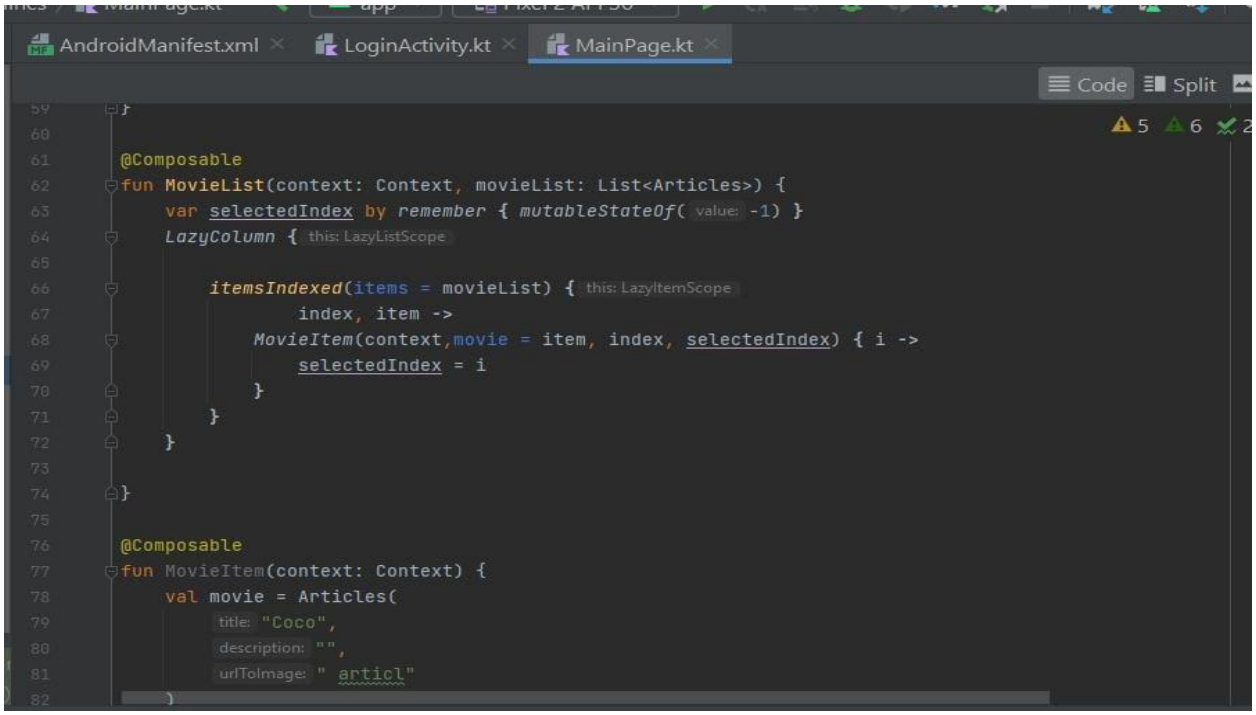

AndroidManifest.xml
LoginActivity.kt
Code
Split
tint = Color(color: 0xFF6495ED)
)
},
placeholder = { Text(text = "password", color = Color.Black) },
visualTransformation = PasswordVisualTransformation(),
colors = TextFieldDefaults.textFieldColors(backgroundColor = Color.Transparent)
)
Spacer(modifier = Modifier.height(12.dp))
if (error.isNotEmpty()) {
Text(
text = error,
color = MaterialTheme.colors.error,
modifier = Modifier.padding(vertical = 16.dp)
)
}
Button(
onClick = {
if (username.isNotEmpty() && password.isNotEmpty()) {
val user = databaseHelper.getUserByUsername(username)
App Quality Insights
TODO
Problems
Terminal
Services
App Inspection

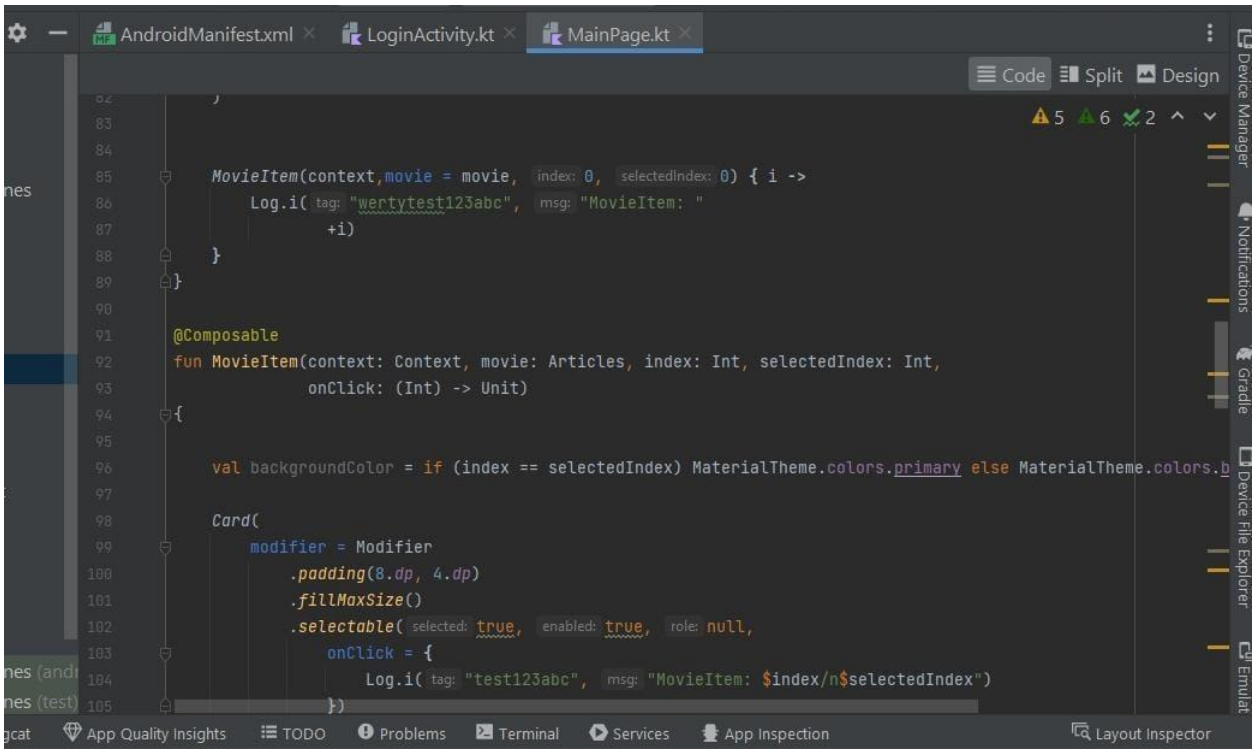

AndroidManifest.xml
LoginActivity.kt
Code
Split
error = "Successfully log in"
context.startActivity(
Intent(
context,
MainPage::class.java
)
)
//onLoginSuccess()
} else {
error = "Invalid username or password"
}
} else {
error = "Please fill all fields"
}
},
shape = RoundedCornerShape(20.dp),
colors = ButtonDefaults.buttonColors(backgroundColor = Color(color: 0xFF77a2ef)),
modifier = Modifier.width(200.dp)
.padding(top = 16.dp)
) { this: RowScope
Text(text = "Log In", fontWeight = FontWeight.Bold)
}

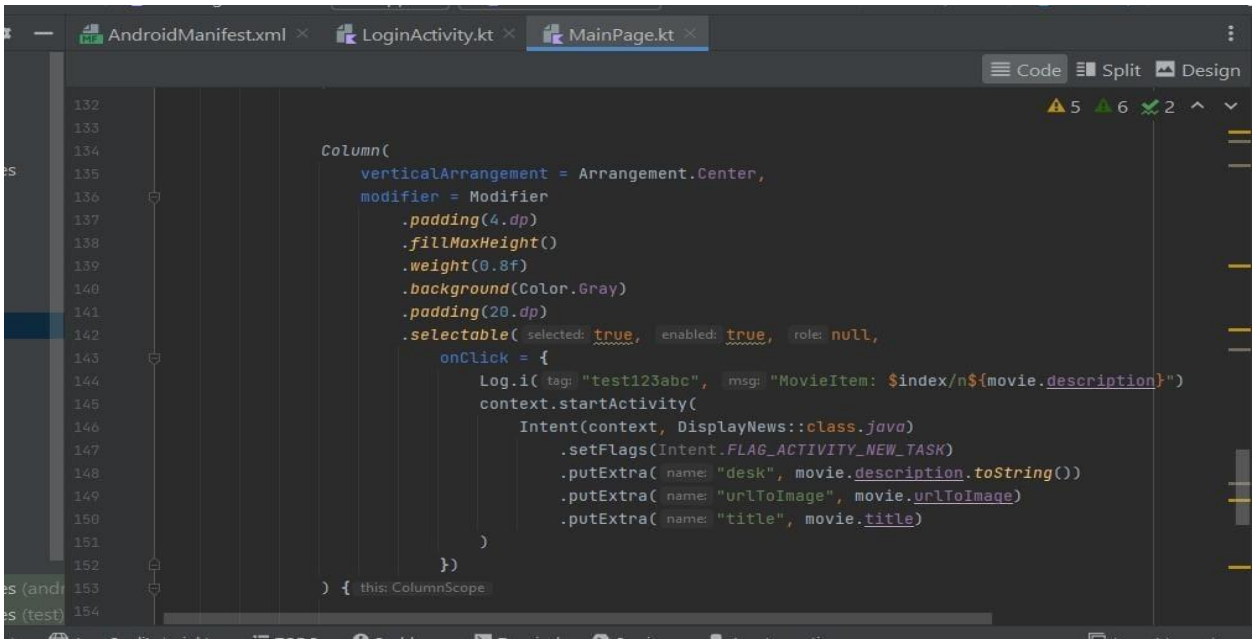

AndroidManifest.xml
LoginActivity.kt
)}
Spacer(modifier = Modifier.width(100.dp))
TextButton(onClick = { /* Do something! */ })
{ Text(text = "Forgot password ?",
color = Color.Black
)}
}
}
}
private fun startMainPage(context: Context) {
val intent = Intent(context, MainPage::class.java)
ContextCompat.startActivity(context, intent, options: null)
}

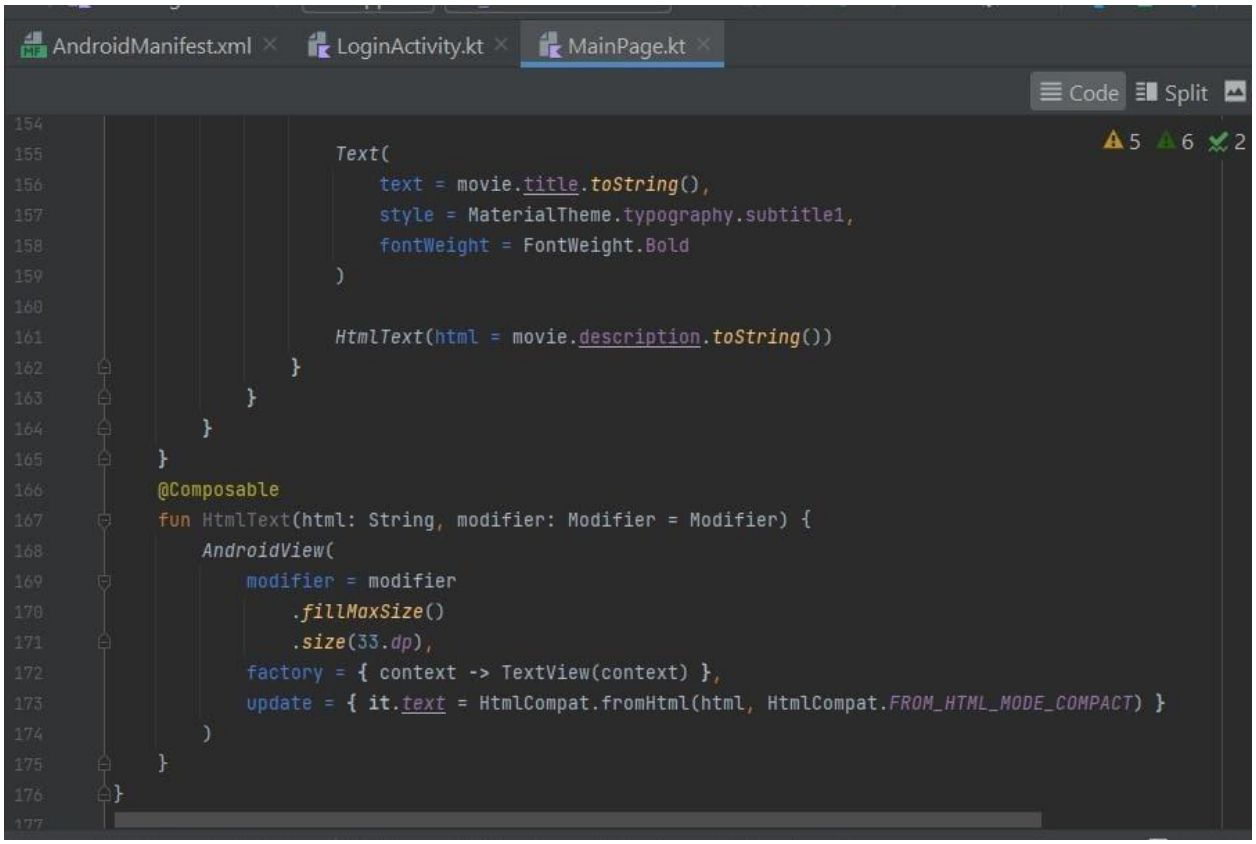
MainPage.kt


AndroidManifest.xml
LoginActivity.kt
MainPage.kt
Code
Split
5 6 2
package com.example.newsheadlines
import ...
class MainPage : ComponentActivity() {
val mainViewModel by viewModels<MainViewModel>()
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContent {
NewsHeadlinesTheme {
// A surface container using the 'background' color from the theme
Surface(color = MaterialTheme.colors.background) {
Column() { this: ColumnScope
Text(text = "Latest NEWS", fontSize = 32.sp, modifier = Modifier.fillMaxWidth(),
MovieList(applicationContext, movieList = mainViewModel.movieListResponse)
mainViewModel.getMovieList()
}
}
}
}
App Quality Insights
TODO
Problems
Terminal
Services
App Inspection

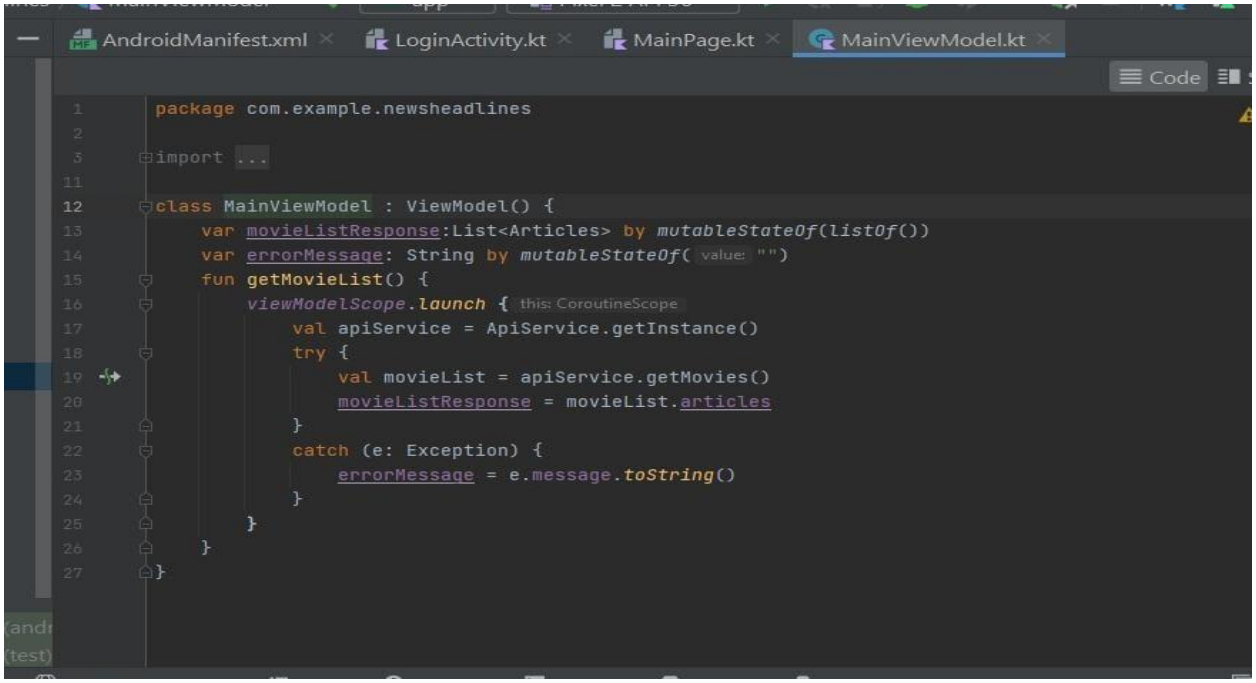

AndroidManifest.xml
LoginActivity.kt
MainPage.kt
Code
Split
5 6 2
}
@Composable
fun MovieList(context: Context, movieList: List<Articles>) {
var selectedIndex by remember { mutableStateOf(value: -1) }
LazyColumn { this: LazyListScope
itemsIndexed(items = movieList) { this: LazyItemScope
index, item ->
MovieItem(context,movie = item, index, selectedIndex) { i ->
selectedIndex = i
}
}
}
}
@Composable
fun MovieItem(context: Context) {
val movie = Articles(
title: "Coco",
description: "",
urlToImage: " articl"
)


AndroidManifest.xml
LoginActivity.kt
MainPage.kt
Code
Split
Design
5 6 2
MovieItem(context,movie = movie, index: 0, selectedIndex: 0) { i ->
Log.i(tag: "wertytest123abc", msg: "MovieItem: "
+i)
}
}
@Composable
fun MovieItem(context: Context, movie: Articles, index: Int, selectedIndex: Int,
onClick: (Int) -> Unit)
{
val backgroundColor = if (index == selectedIndex) MaterialTheme.colors.primary else MaterialTheme.colors.b
Card(
modifier = Modifier
.padding(8.dp, 4.dp)
.fillMaxSize()
.selectable(selected: true, enabled: true, role: null,
onClick = {
Log.i(tag: "test123abc", msg: "MovieItem: $index/n$selectedIndex")
})
App Quality Insights
TODO
Problems
Terminal
Services
App Inspection
Layout Inspector
Device Manager
Notifications
Gradle
Device File Explorer

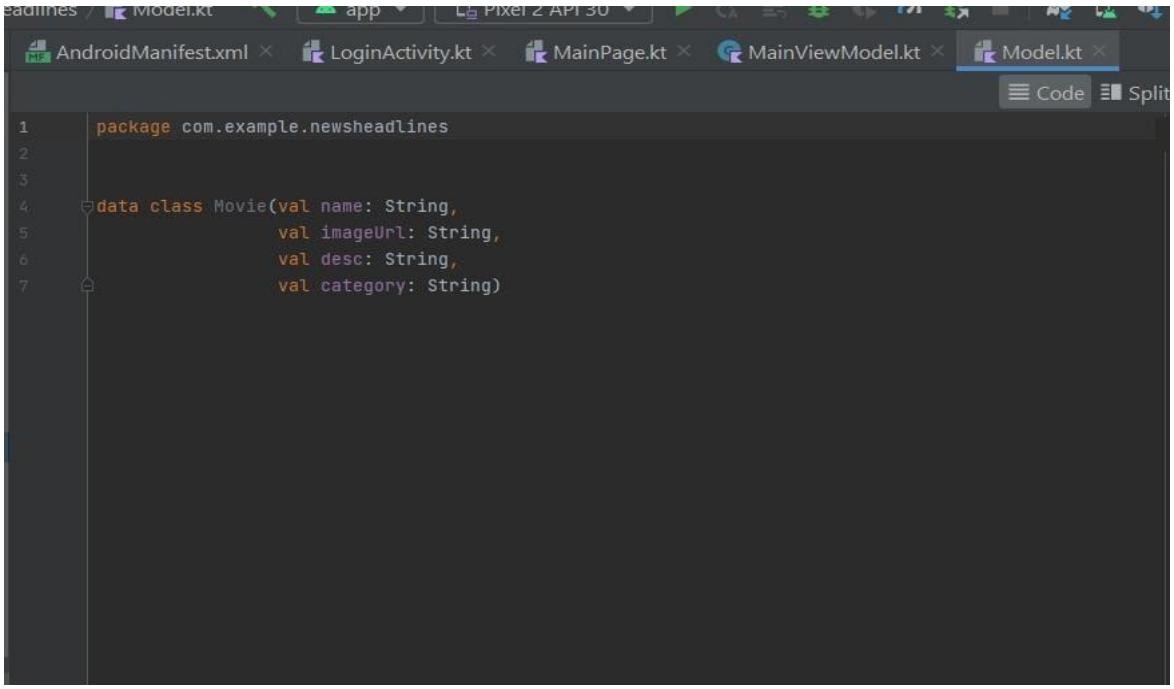

AndroidManifest.xml
LoginActivity.kt
MainPage.kt
Code
Split
Design
Column(
verticalArrangement = Arrangement.Center,
modifier = Modifier
.padding(4.dp)
.fillMaxHeight()
.weight(0.8f)
.background(Color.Gray)
.padding(20.dp)
.selectable(selected: true, enabled: true, role: null,
onClick = {
Log.i(tag: "test123abc", msg: "MovieItem: $index/n${movie.description}")
context.startActivity(
Intent(context, DisplayNews::class.java)
.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK)
.putExtra(name: "desk", movie.description.toString())
.putExtra(name: "urlToImage", movie.urlToImage)
.putExtra(name: "title", movie.title)
)
})
) { this: ColumnScope


AndroidManifest.xml
LoginActivity.kt
MainPage.kt
Code
Split
Text(
text = movie.title.toString(),
style = MaterialTheme.typography.subtitle1,
fontWeight = FontWeight.Bold
)
HtmlText(html = movie.description.toString())
@Composable
fun HtmlText(html: String, modifier: Modifier = Modifier) {
AndroidView(
modifier = modifier
.fillMaxSize()
.size(33.dp),
factory = { context -> TextView(context) },
update = { it.text = HtmlCompat.fromHtml(html, HtmlCompat.FROM_HTML_MODE_COMPACT) }
)

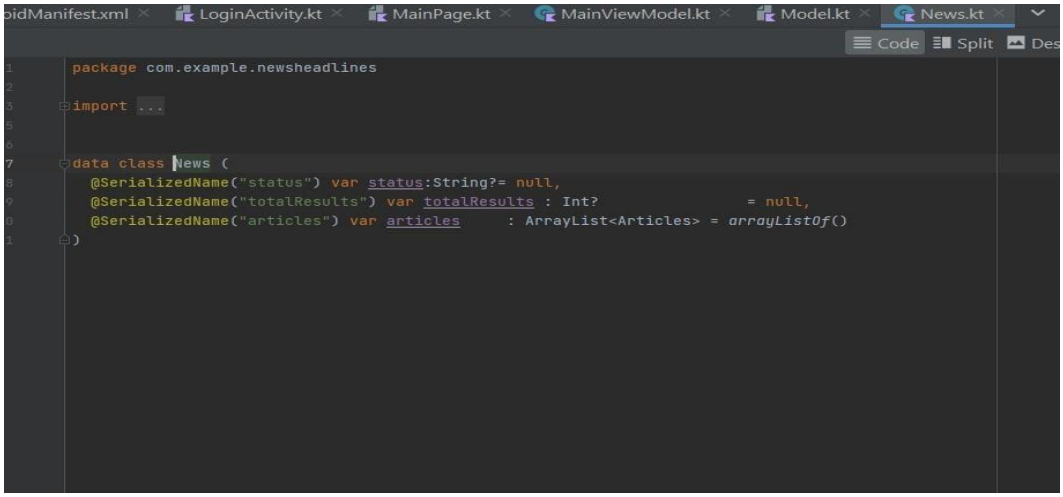
MainViewModel.kt



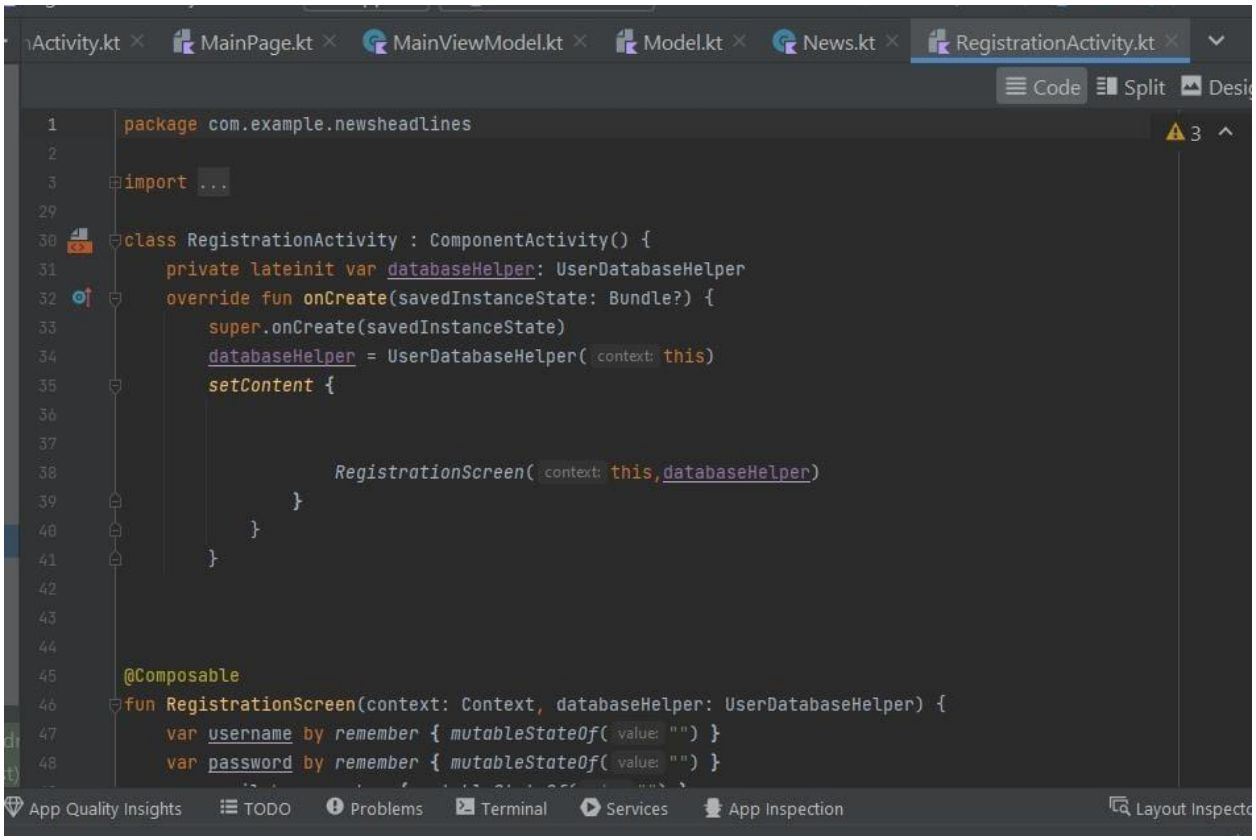
Model.kt

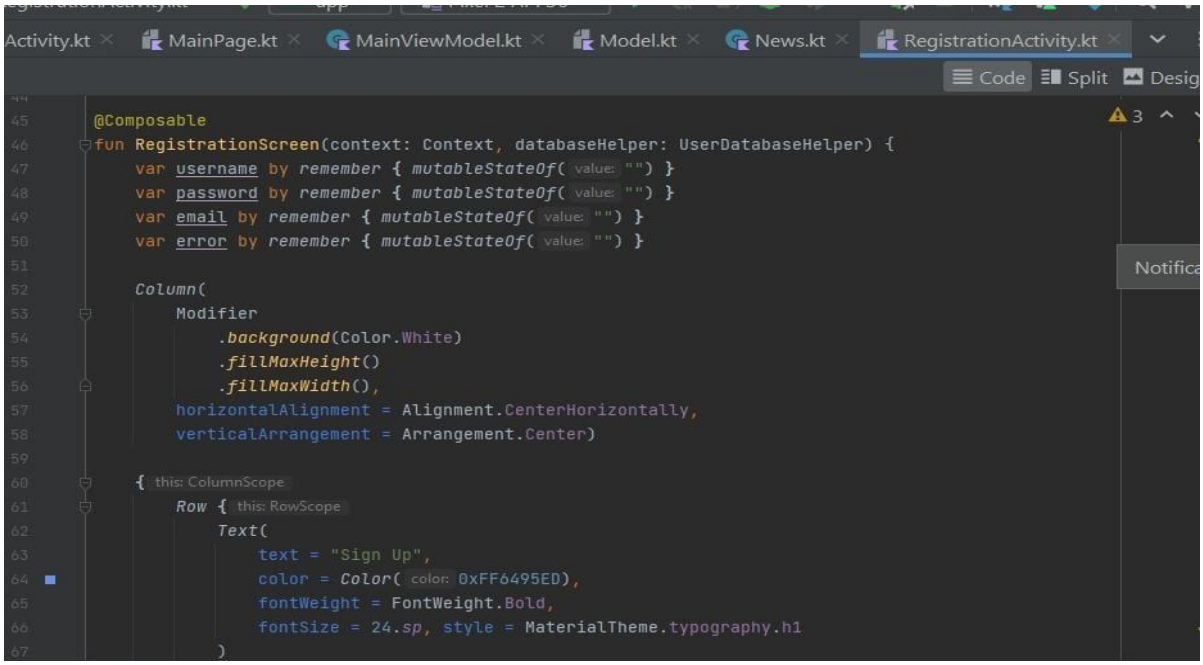


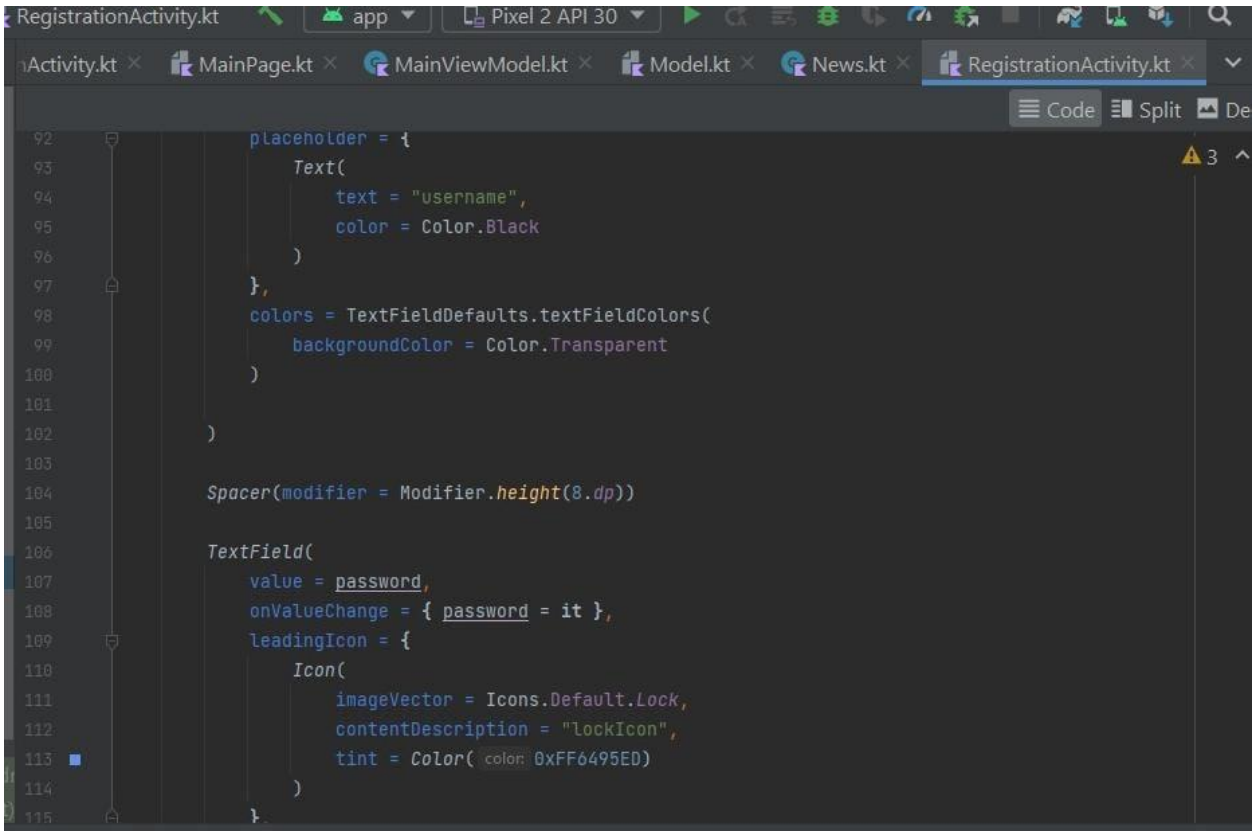
News.kt

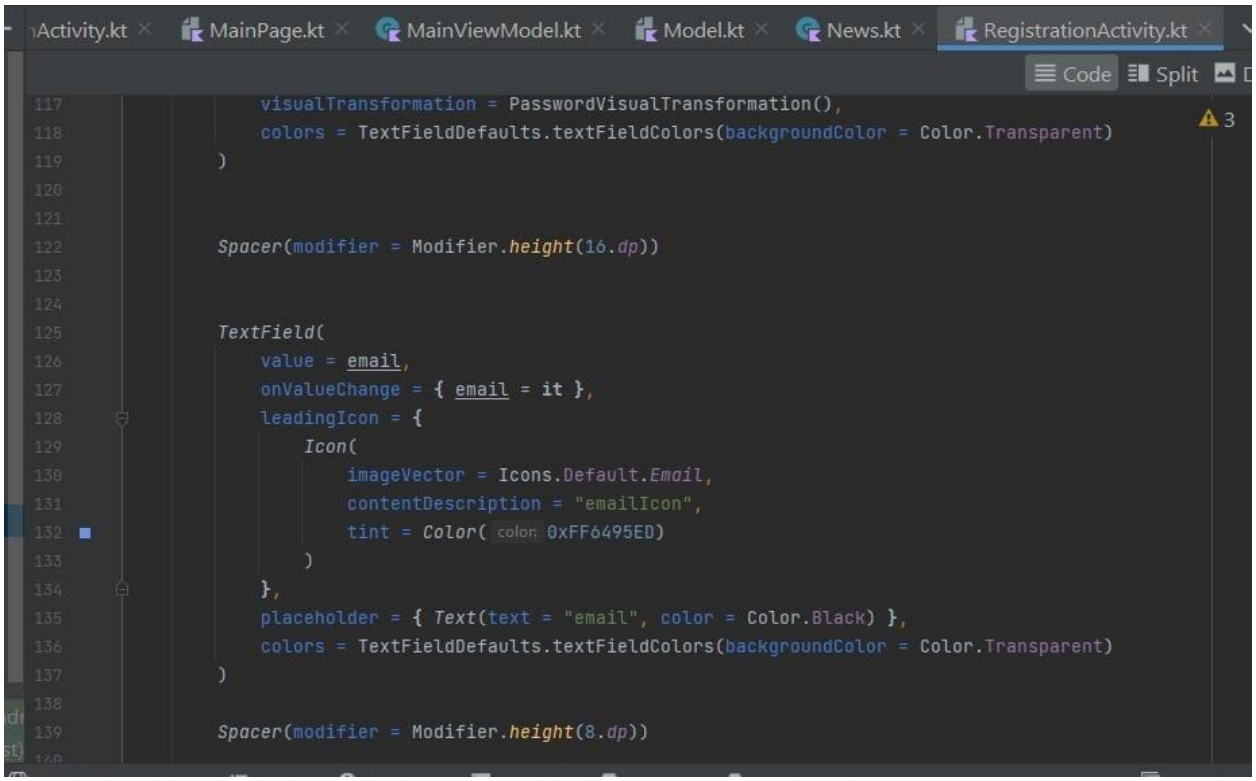


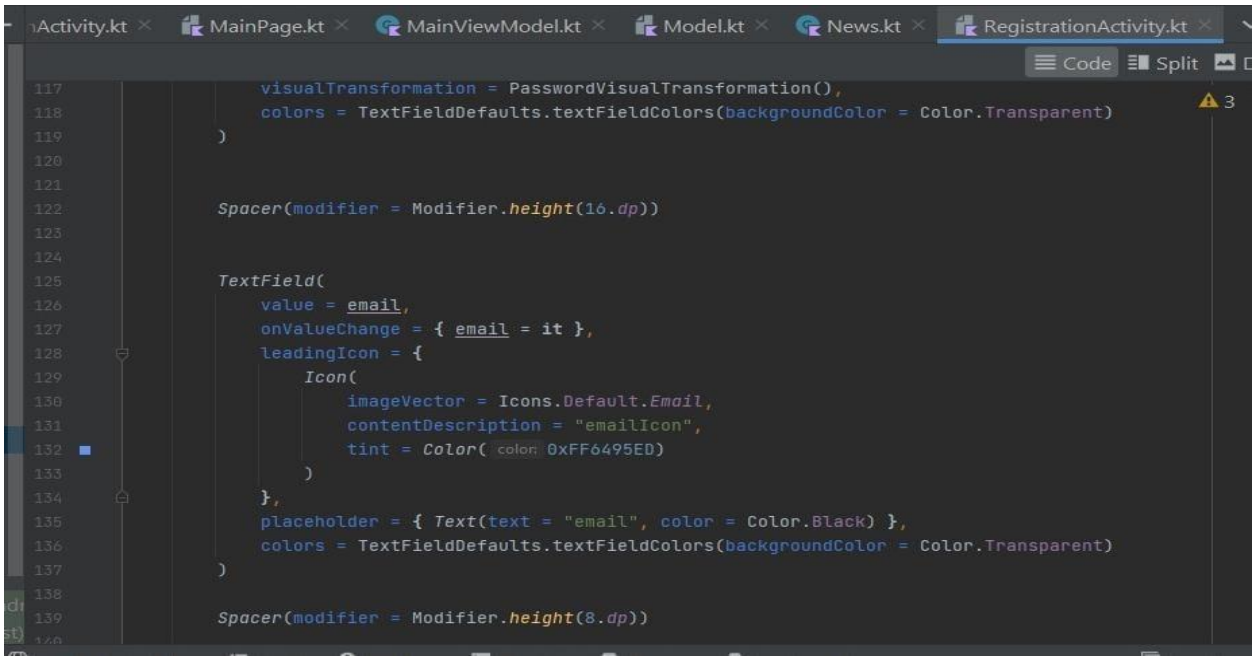
RegistrationActivity.kt

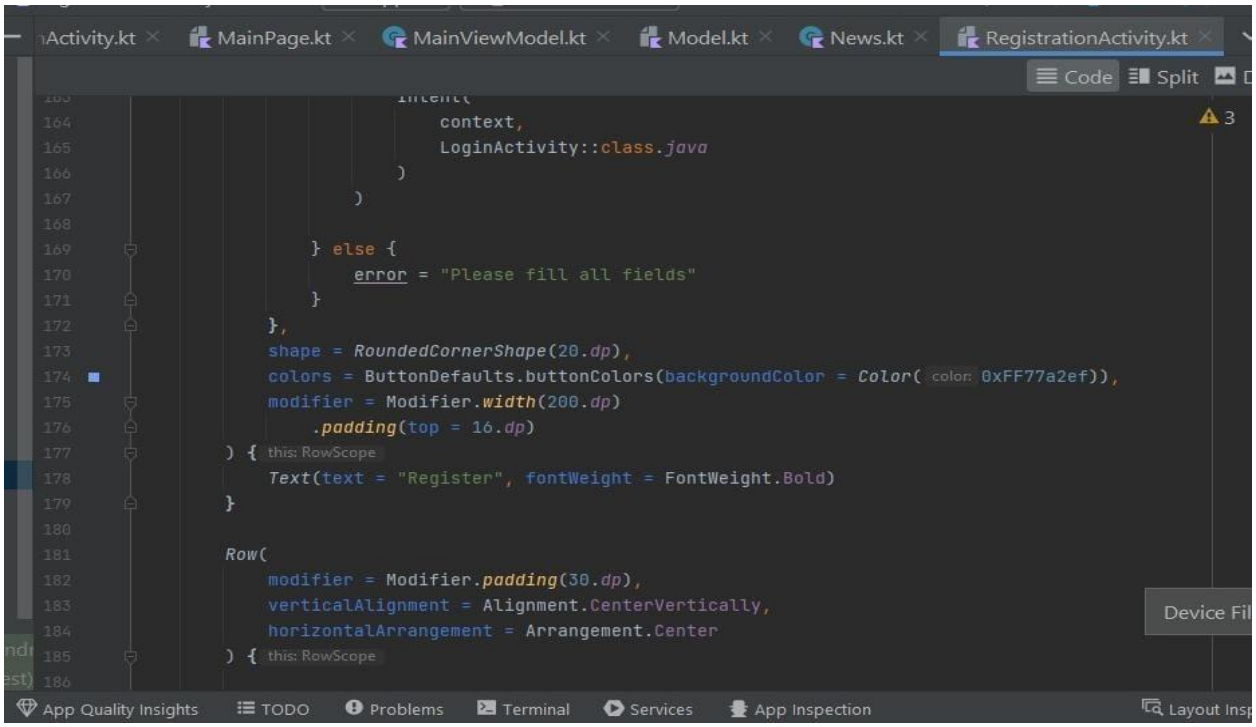


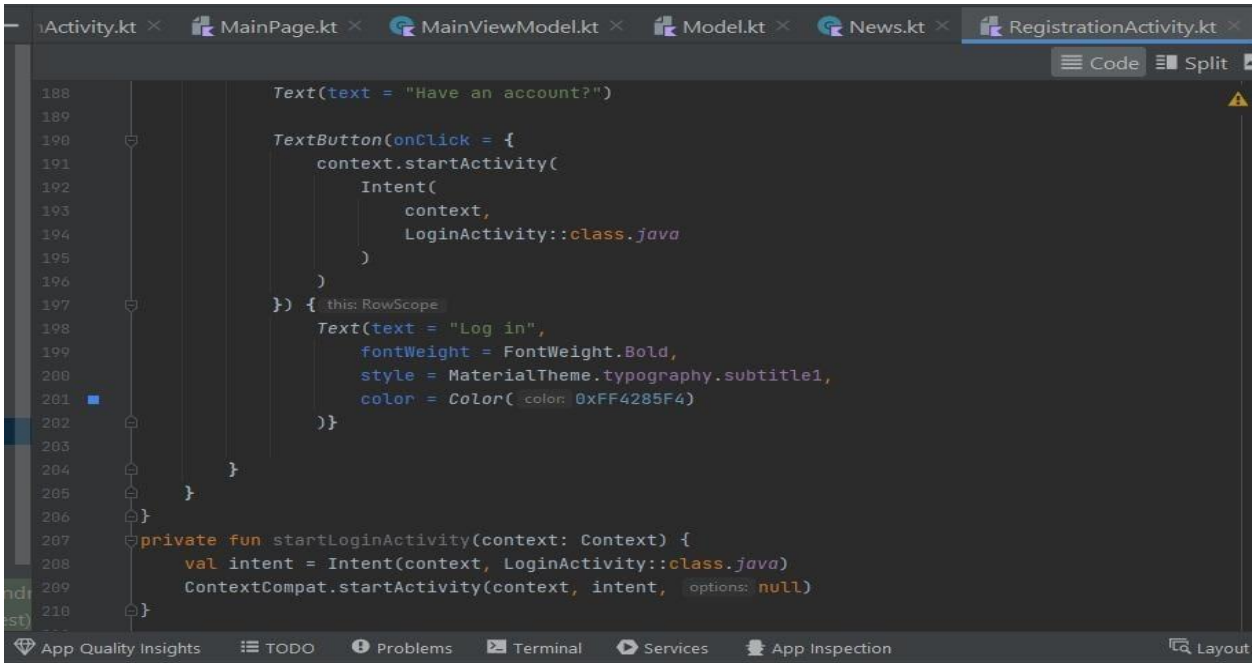

Activity.kt
MainPage.kt
MainViewModel.kt
Model.kt
News.kt
RegistrationActivity.kt
Code
Split
Desig
3
Notifica
@Composable
fun RegistrationScreen(context: Context, databaseHelper: UserDatabaseHelper) {
var username by remember { mutableStateOf(value: "") }
var password by remember { mutableStateOf(value: "") }
var email by remember { mutableStateOf(value: "") }
var error by remember { mutableStateOf(value: "") }
Column(
Modifier
.background(Color.White)
.fillMaxHeight()
.fillMaxWidth(),
horizontalAlignment = Alignment.CenterHorizontally,
verticalArrangement = Arrangement.Center)
{ this: ColumnScope
Row { this: RowScope
Text(
text = "Sign Up",
color = Color(color: 0xFF6495ED),
fontWeight = FontWeight.Bold,
fontSize = 24.sp, style = MaterialTheme.typography.h1
)


RegistrationActivity.kt
app
Pixel 2 API 30
Activity.kt
MainPage.kt
MainViewModel.kt
Model.kt
News.kt
RegistrationActivity.kt
Code
Split
3
placeholder = {
Text(
text = "username",
color = Color.Black
)
},
colors = TextFieldDefaults.textFieldColors(
backgroundColor = Color.Transparent
)
)
Spacer(modifier = Modifier.height(8.dp))
TextField(
value = password,
onValueChange = { password = it },
leadingIcon = {
Icon(
imageVector = Icons.Default.Lock,
contentDescription = "lockIcon",
tint = Color(color: 0xFF6495ED)
)
}

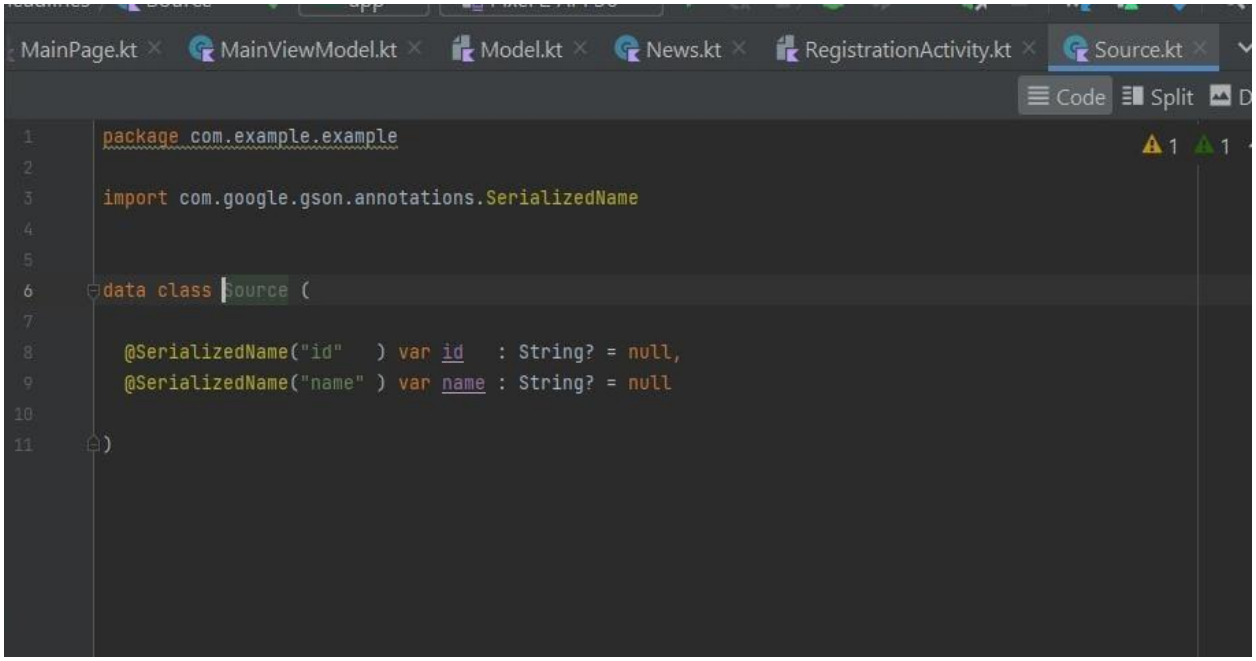

Activity.kt
MainPage.kt
MainViewModel.kt
Model.kt
News.kt
RegistrationActivity.kt
Code
Split
3
visualTransformation = PasswordVisualTransformation(),
colors = TextFieldDefaults.textFieldColors(backgroundColor = Color.Transparent)
)
Spacer(modifier = Modifier.height(16.dp))
TextField(
value = email,
onValueChange = { email = it },
leadingIcon = {
Icon(
imageVector = Icons.Default.Email,
contentDescription = "emailIcon",
tint = Color(color: 0xFF6495ED)
)
},
placeholder = { Text(text = "email", color = Color.Black) },
colors = TextFieldDefaults.textFieldColors(backgroundColor = Color.Transparent)
)
Spacer(modifier = Modifier.height(8.dp))


Activity.kt
MainPage.kt
MainViewModel.kt
Model.kt
News.kt
RegistrationActivity.kt
Code
Split
3
visualTransformation = PasswordVisualTransformation(),
colors = TextFieldDefaults.textFieldColors(backgroundColor = Color.Transparent)
)
Spacer(modifier = Modifier.height(16.dp))
TextField(
value = email,
onValueChange = { email = it },
leadingIcon = {
Icon(
imageVector = Icons.Default.Email,
contentDescription = "emailIcon",
tint = Color(color: 0xFF6495ED)
)
},
placeholder = { Text(text = "email", color = Color.Black) },
colors = TextFieldDefaults.textFieldColors(backgroundColor = Color.Transparent)
)
Spacer(modifier = Modifier.height(8.dp))

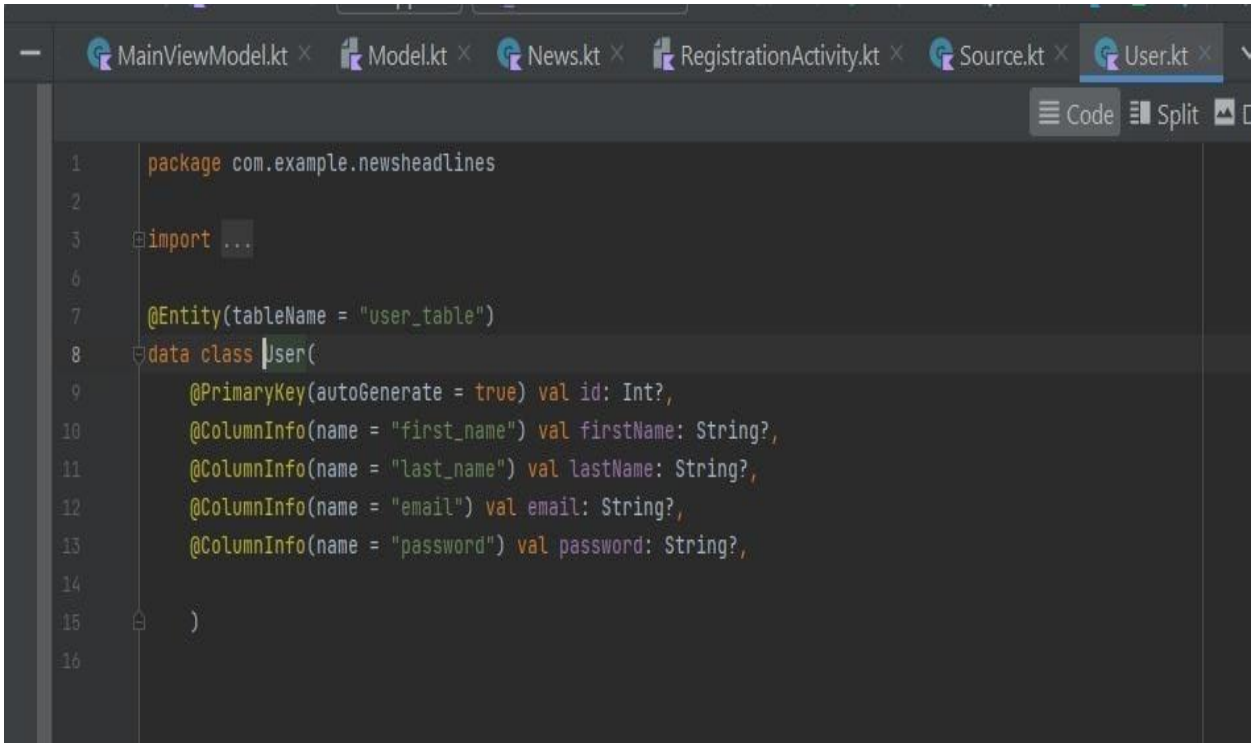

Activity.kt
MainPage.kt
MainViewModel.kt
Model.kt
News.kt
RegistrationActivity.kt
Code
Split
context,
LoginActivity::class.java
} else {
error = "Please fill all fields"
},
shape = RoundedCornerShape(20.dp),
colors = ButtonDefaults.buttonColors(backgroundColor = Color(color: 0xFF77a2ef)),
modifier = Modifier.width(200.dp)
.padding(top = 16.dp)
) { this: RowScope
Text(text = "Register", fontWeight = FontWeight.Bold)
}
Row(
modifier = Modifier.padding(30.dp),
verticalAlignment = Alignment.CenterVertically,
horizontalArrangement = Arrangement.Center
) { this: RowScope
App Quality Insights
TODO
Problems
Terminal
Services
App Inspection


Activity.kt
MainPage.kt
MainViewModel.kt
Model.kt
News.kt
RegistrationActivity.kt
Code
Split
Text(text = "Have an account?")
TextButton(onClick = {
context.startActivity(
Intent(
context,
LoginActivity::class.java
)
)
}) { this: RowScope
Text(text = "Log in",
fontWeight = FontWeight.Bold,
style = MaterialTheme.typography.subtitle1,
color = Color(color: 0xFF4285F4)
)}
}
}
}
private fun startLoginActivity(context: Context) {
val intent = Intent(context, LoginActivity::class.java)
ContextCompat.startActivity(context, intent, options: null)
}
App Quality Insights
TODO
Problems
Terminal
Services
App Inspection

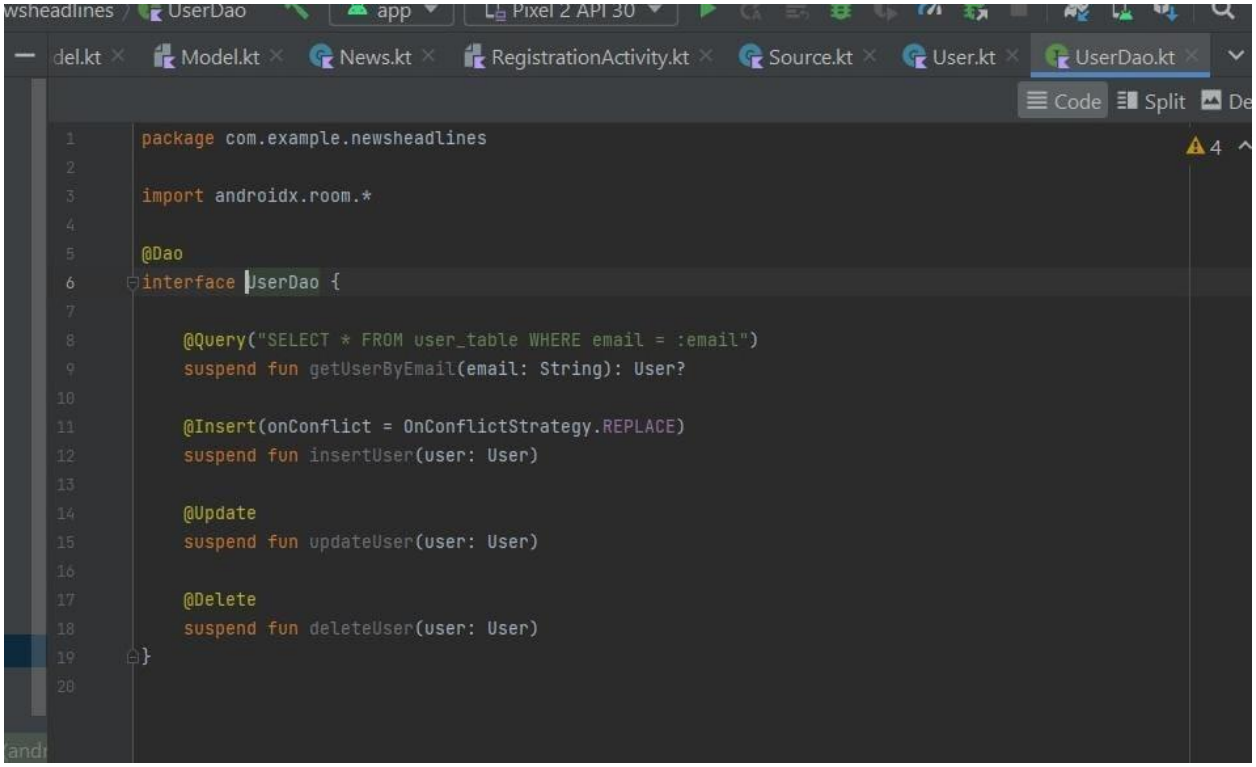
Source.kt



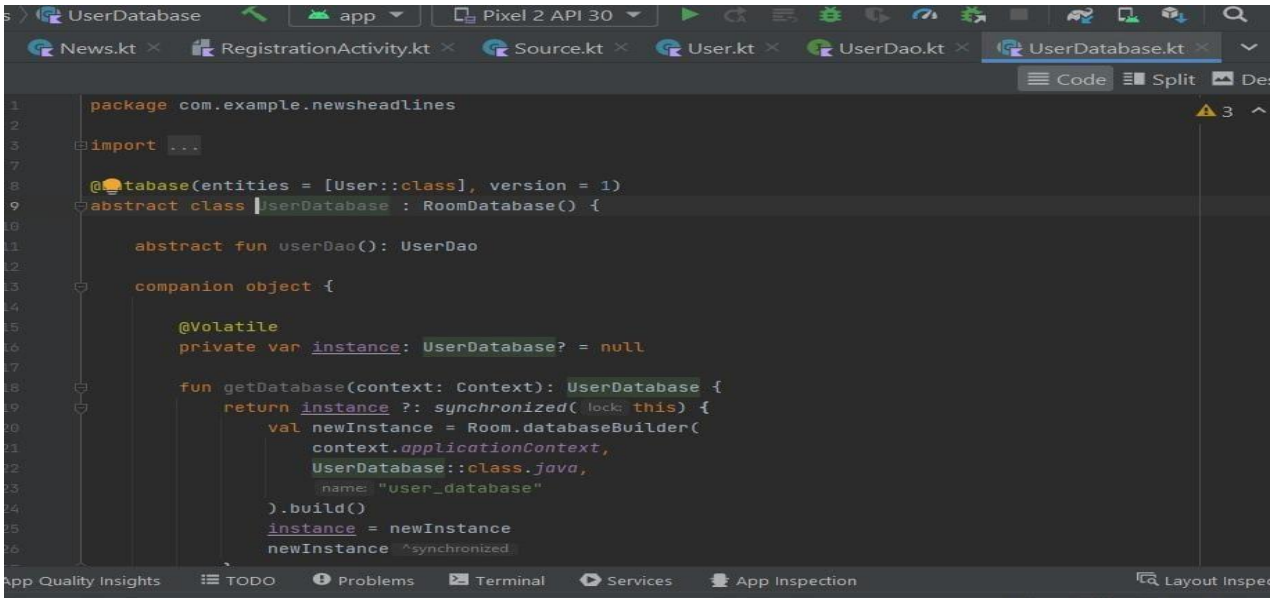
User.kt

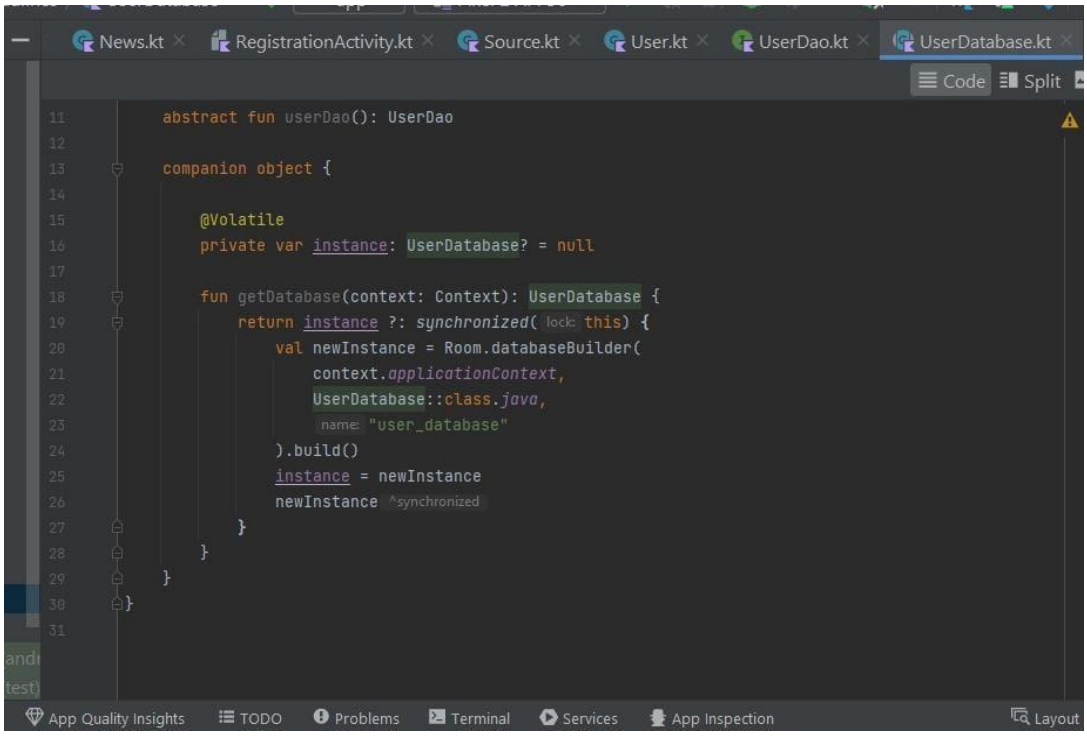


UserDao.kt

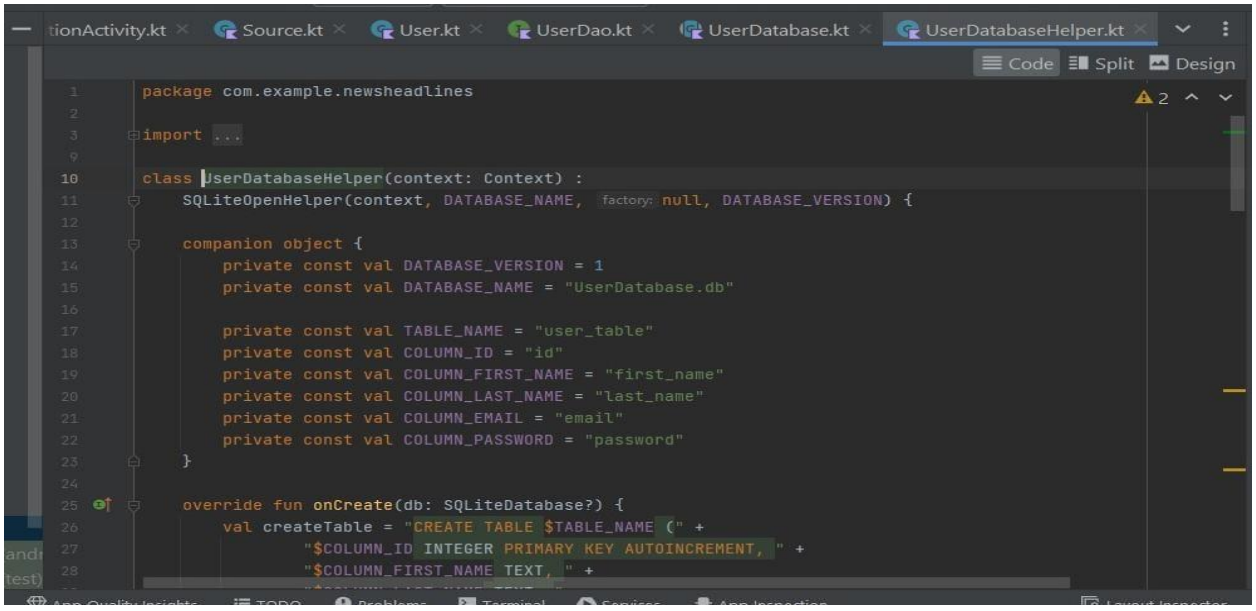


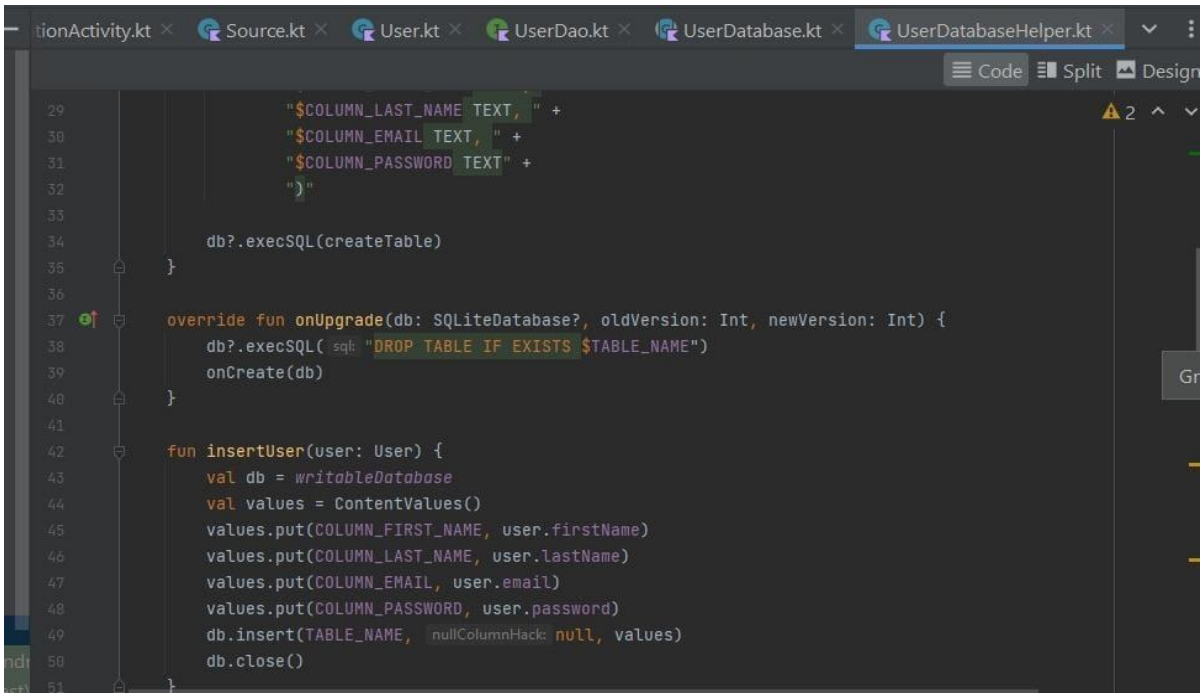
Userdatabase.kt

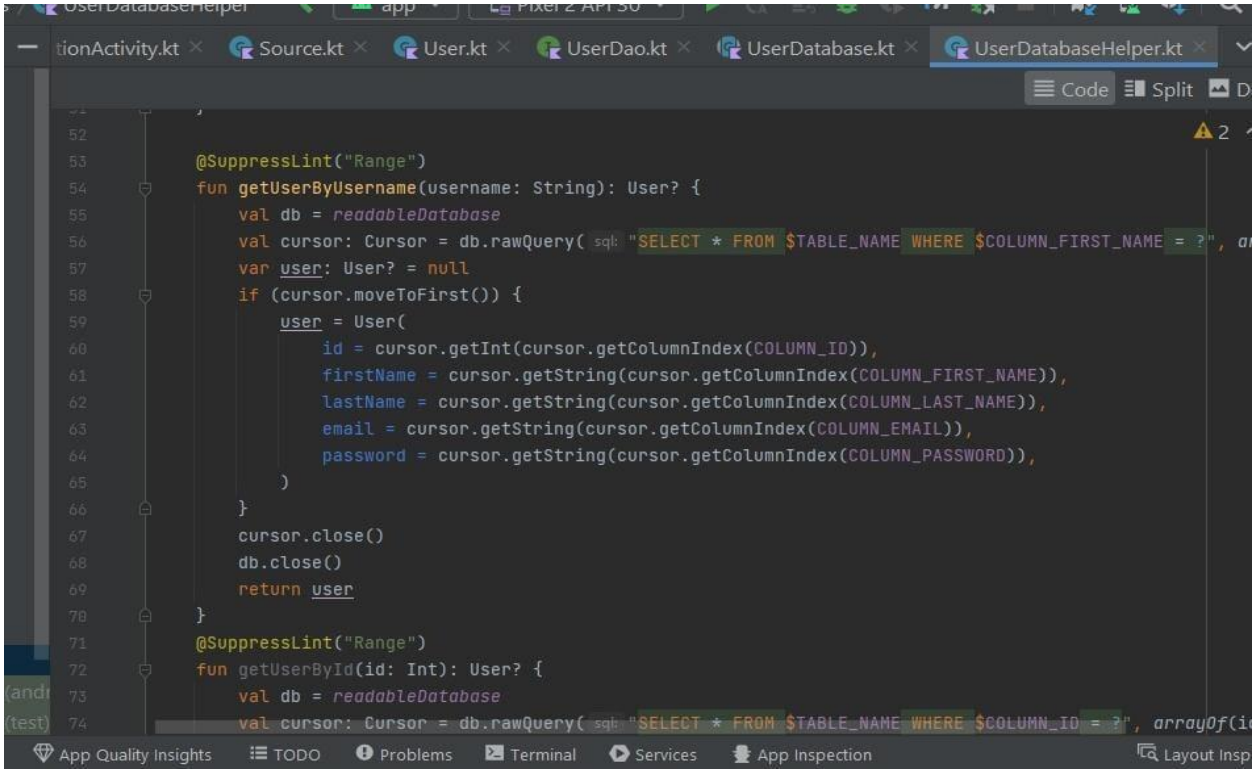


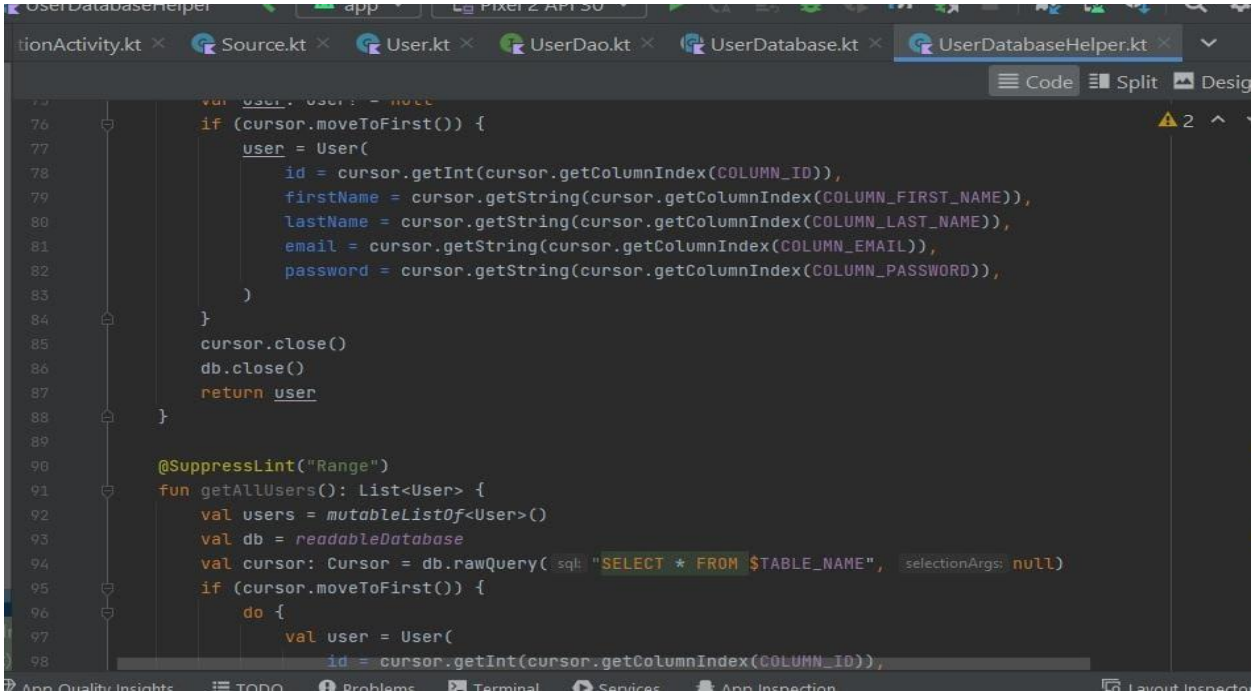


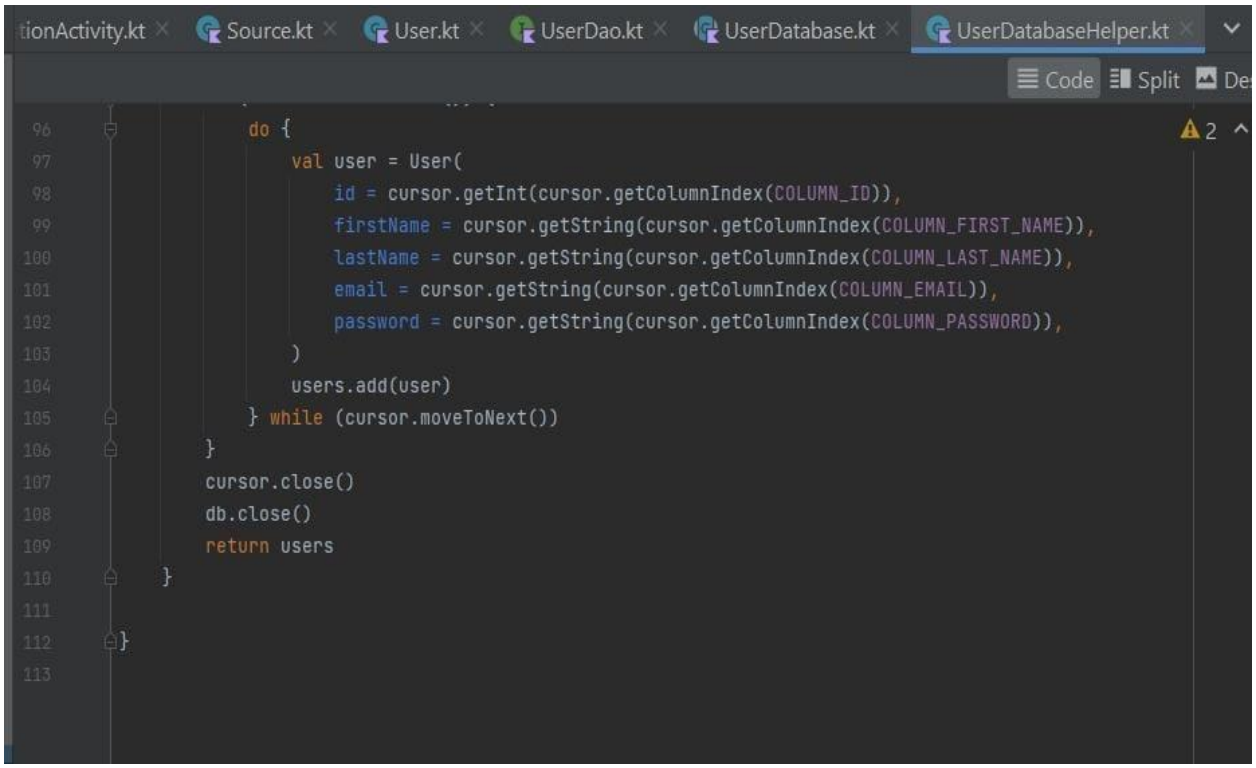
UserDatabaseHelper.kt




tionActivity.kt
Source.kt
User.kt
UserDao.kt
UserDatabase.kt
UserDatabaseHelper.kt
Code
Split
Design
"$COLUMN_LAST_NAME TEXT, " +
"$COLUMN_EMAIL TEXT, " +
"$COLUMN_PASSWORD TEXT" +
")"
db?.execSQL(createTable)
}
override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
db?.execSQL(sql: "DROP TABLE IF EXISTS $TABLE_NAME")
onCreate(db)
}
fun insertUser(user: User) {
val db = writableDatabase
val values = ContentValues()
values.put(COLUMN_FIRST_NAME, user.firstName)
values.put(COLUMN_LAST_NAME, user.lastName)
values.put(COLUMN_EMAIL, user.email)
values.put(COLUMN_PASSWORD, user.password)
db.insert(TABLE_NAME, nullColumnHack: null, values)
db.close()
}


tionActivity.kt
Source.kt
User.kt
UserDao.kt
UserDatabase.kt
UserDatabaseHelper.kt
Code
Split
@SuppressLint("Range")
fun getUserByUsername(username: String): User? {
val db = readableDatabase
val cursor: Cursor = db.rawQuery(sql: "SELECT * FROM $TABLE_NAME WHERE $COLUMN_FIRST_NAME = ?",
var user: User? = null
if (cursor.moveToFirst()) {
user = User(
id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
firstName = cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
lastName = cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
password = cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
)
}
cursor.close()
db.close()
return user
}
@SuppressLint("Range")
fun getUserById(id: Int): User? {
val db = readableDatabase
val cursor: Cursor = db.rawQuery(sql: "SELECT * FROM $TABLE_NAME WHERE $COLUMN_ID = ?", arrayOf(
App Quality Insights
TODO
Problems
Terminal
Services
App Inspection
Layout Insp


tionActivity.kt
Source.kt
User.kt
UserDao.kt
UserDatabase.kt
UserDatabaseHelper.kt
Code
Split
Desig
if (cursor.moveToFirst()) {
user = User(
id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
firstName = cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
lastName = cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
password = cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
)
}
cursor.close()
db.close()
return user
}
@SuppressLint("Range")
fun getAllUsers(): List<User> {
val users = mutableListOf<User>()
val db = readableDatabase
val cursor: Cursor = db.rawQuery(sql: "SELECT * FROM $TABLE_NAME", selectionArgs: null)
if (cursor.moveToFirst()) {
do {
val user = User(
id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),


tionActivity.kt
Source.kt
User.kt
UserDao.kt
UserDatabase.kt
UserDatabaseHelper.kt
Code
Split
do {
val user = User(
id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
firstName = cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
lastName = cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
password = cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
)
users.add(user)
} while (cursor.moveToNext())
}
cursor.close()
db.close()
return users
}
}