Part 1: Setting up Your Environment

Part 2: It's all about intent

Part 2.1: What's the difference?

1. **What are the two types of Intents?**

Explicit intent specifies which application will satisfy the intent, by supplying either the target app's package name or a fully qualified component class name. Generally speaking, use of an explicit intent to start a component in one’s app, because you know the class name of the activity or service to start.
Example, start a new activity within an app in response to a user action or start a service to download a file in the background.

Implicit intent outside of the system that does not name a specific component, but instead declares a general action to perform, which allows a component from another app to handle it.
Example, to show the user a location on a map, - use an implicit intent to request that another capable app show a specified location on a map.

2. **Which of the two types of Intents are more secure?**

Explicit intent is more secure which have specified a component (via setComponent(ComponentName) or setClass(Context, Class)), which provides the exact class to be run.

3. **What type of Intent is shown on lines 69 to 73 of SecondFragment.kt?**

Implicit intent – does not specify a component; instead, they must include enough information for the system to determine which of the available components is best to run for that intent. (e.g. intent-filter)

```
var intent = Intent(Intent.ACTION_VIEW)
intent.type = "text/giftcards_browse"
intent.data = Uri.parse("https://appsecclass.report/api/index")
intent.putExtra("User", loggedInUser);
startActivity(intent)
```

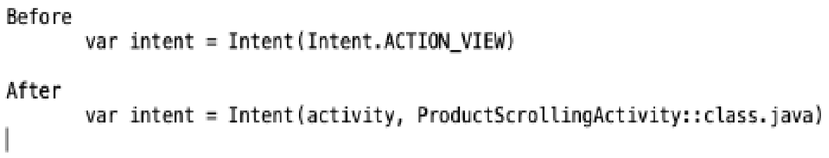
4. **What type of Intent is shown on lines 68 to 70 of ThirdFragment.kt?**

Explicit which have specific a component and called the activity in Java and pass the values.

```
var intent = Intent(activity,
ProductScrollingActivity::class.java)
intent.putExtra("User",
loggedInUser);
startActivity(intent)
```

5. **Which of these two Intents is the proper way to do an Intent?**

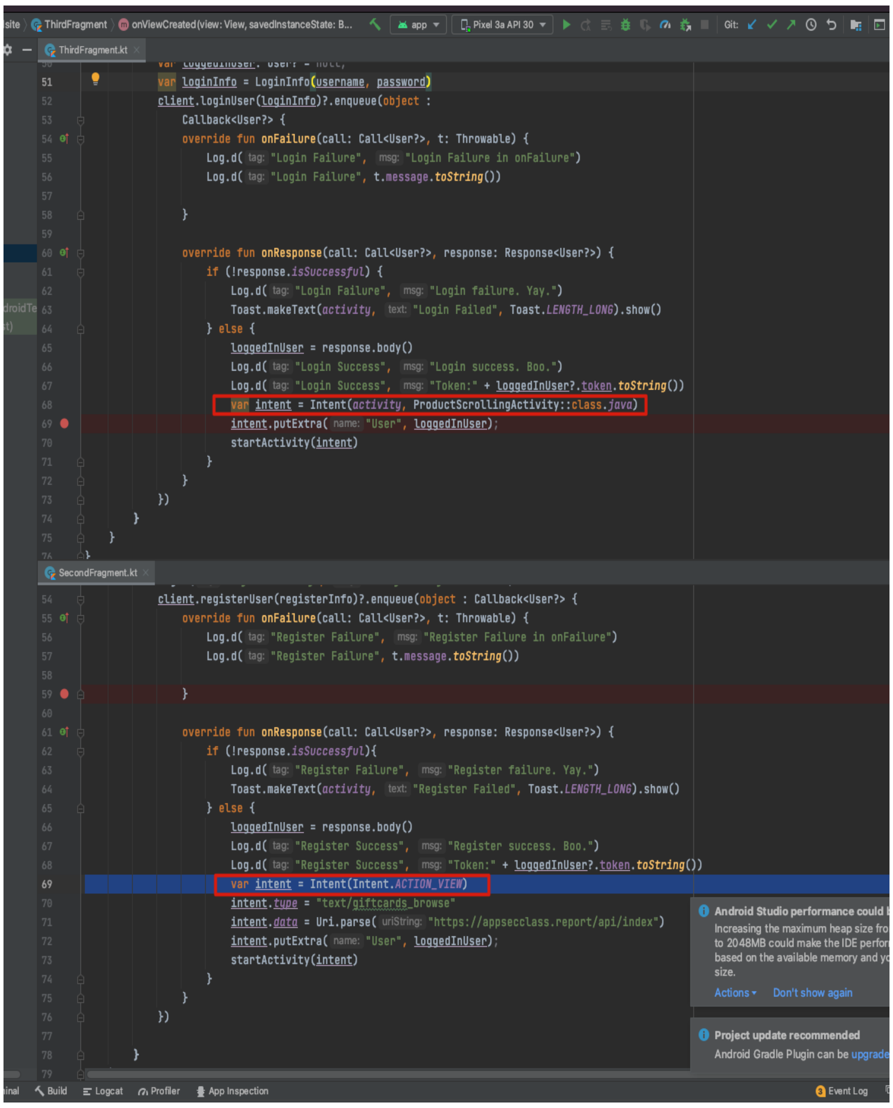
Explicit intent is more secure so proper way to do an Intent.



```
Before
      var intent = Intent(Intent.ACTION_VIEW)

After
      var intent = Intent(activity, ProductScrollingActivity::class.java)
```

In SecondFragment line 69

```
Before
// var intent = Intent(Intent.ACTION_VIEW)
After
var intent = Intent(activity,
        ProductScrollingActivity::class.java)
```


site 〉 ThirdFragment 〉 onViewCreated(view: View, savedInstanceState: B...
app
Pixel 3a API 30
Git:
ThirdFragment.kt

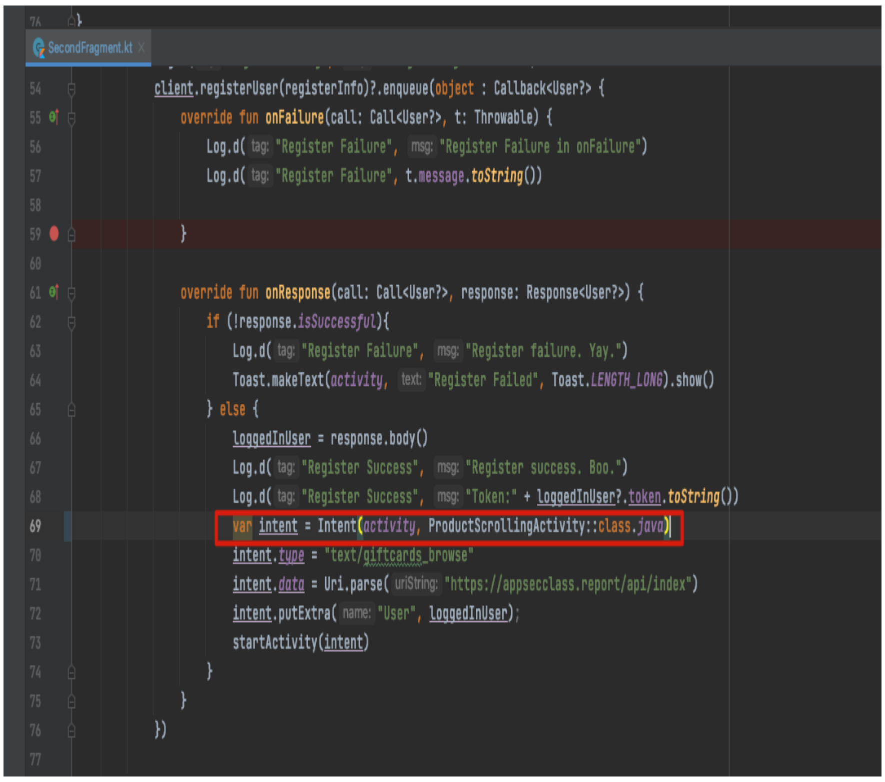
```
        var loggedInUser: User? = null,
        var loginInfo = LoginInfo(username, password)
        client.loginUser(loginInfo)?.enqueue(object :
            Callback<User?> {
            override fun onFailure(call: Call<User?>, t: Throwable) {
                Log.d( tag: "Login Failure",  msg: "Login Failure in onFailure")
                Log.d( tag: "Login Failure", t.message.toString())

            }

            override fun onResponse(call: Call<User?>, response: Response<User?>) {
                if (!response.isSuccessful) {
                    Log.d( tag: "Login Failure",  msg: "Login failure. Yay.")
                    Toast.makeText(activity,  text: "Login Failed", Toast.LENGTH_LONG).show()
                } else {
                    loggedInUser = response.body()
                    Log.d( tag: "Login Success",  msg: "Login success. Boo.")
                    Log.d( tag: "Login Success",  msg: "Token:" + loggedInUser?.token.toString())
                    var intent = Intent(activity, ProductScrollingActivity::class.java)
                    intent.putExtra( name: "User", loggedInUser);
                    startActivity(intent)
                }
            }
        })
    }
}
}
```

```
        client.registerUser(registerInfo)?.enqueue(object : Callback<User?> {
            override fun onFailure(call: Call<User?>, t: Throwable) {
                Log.d( tag: "Register Failure",  msg: "Register Failure in onFailure")
                Log.d( tag: "Register Failure", t.message.toString())

            }

            override fun onResponse(call: Call<User?>, response: Response<User?>) {
                if (!response.isSuccessful){
                    Log.d( tag: "Register Failure",  msg: "Register failure. Yay.")
                    Toast.makeText(activity,  text: "Register Failed", Toast.LENGTH_LONG).show()
                } else {
                    loggedInUser = response.body()
                    Log.d( tag: "Register Success",  msg: "Register success. Boo.")
                    Log.d( tag: "Register Success",  msg: "Token:" + loggedInUser?.token.toString())
                    var intent = Intent(Intent.ACTION_VIEW)
                    intent.type = "text/giftcards_browse"
                    intent.data = Uri.parse( uriString: "https://appsecclass.report/api/index")
                    intent.putExtra( name: "User", loggedInUser);
                    startActivity(intent)
                }
            }
        })

    }
```

Android Studio performance could
Increasing the maximum heap size fro
to 2048MB could make the IDE perfor
based on the available memory and yo
size.

Project update recommended
Android Gradle Plugin can be upgrade

A fixed fragment:

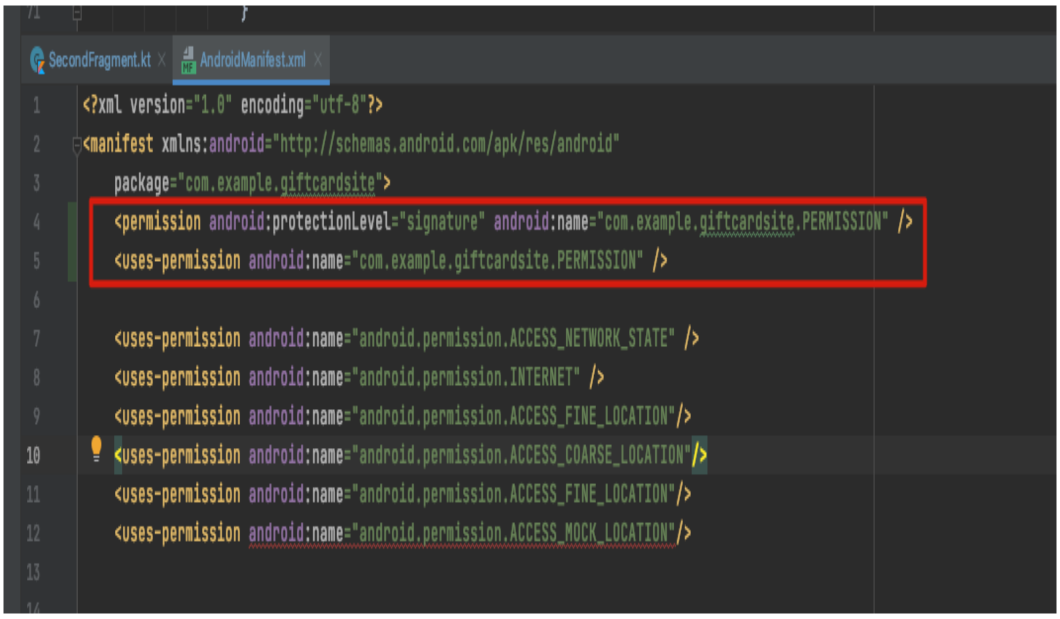


```
        client.registerUser(registerInfo)?.enqueue(object : Callback<User?> {
            override fun onFailure(call: Call<User?>, t: Throwable) {
                Log.d( tag: "Register Failure",  msg: "Register Failure in onFailure")
                Log.d( tag: "Register Failure", t.message.toString())

            }

            override fun onResponse(call: Call<User?>, response: Response<User?>) {
                if (!response.isSuccessful){
                    Log.d( tag: "Register Failure",  msg: "Register failure. Yay.")
                    Toast.makeText(activity,  text: "Register Failed", Toast.LENGTH_LONG).show()
                } else {
                    loggedInUser = response.body()
                    Log.d( tag: "Register Success",  msg: "Register success. Boo.")
                    Log.d( tag: "Register Success",  msg: "Token:" + loggedInUser?.token.toString())
                    var intent = Intent(activity, ProductScrollingActivity::class.java)
                    intent.type = "text/giftcards_browse"
                    intent.data = Uri.parse( uriString: "https://appsecclass.report/api/index")
                    intent.putExtra( name: "User", loggedInUser);
                    startActivity(intent)
                }
            }
        })
```

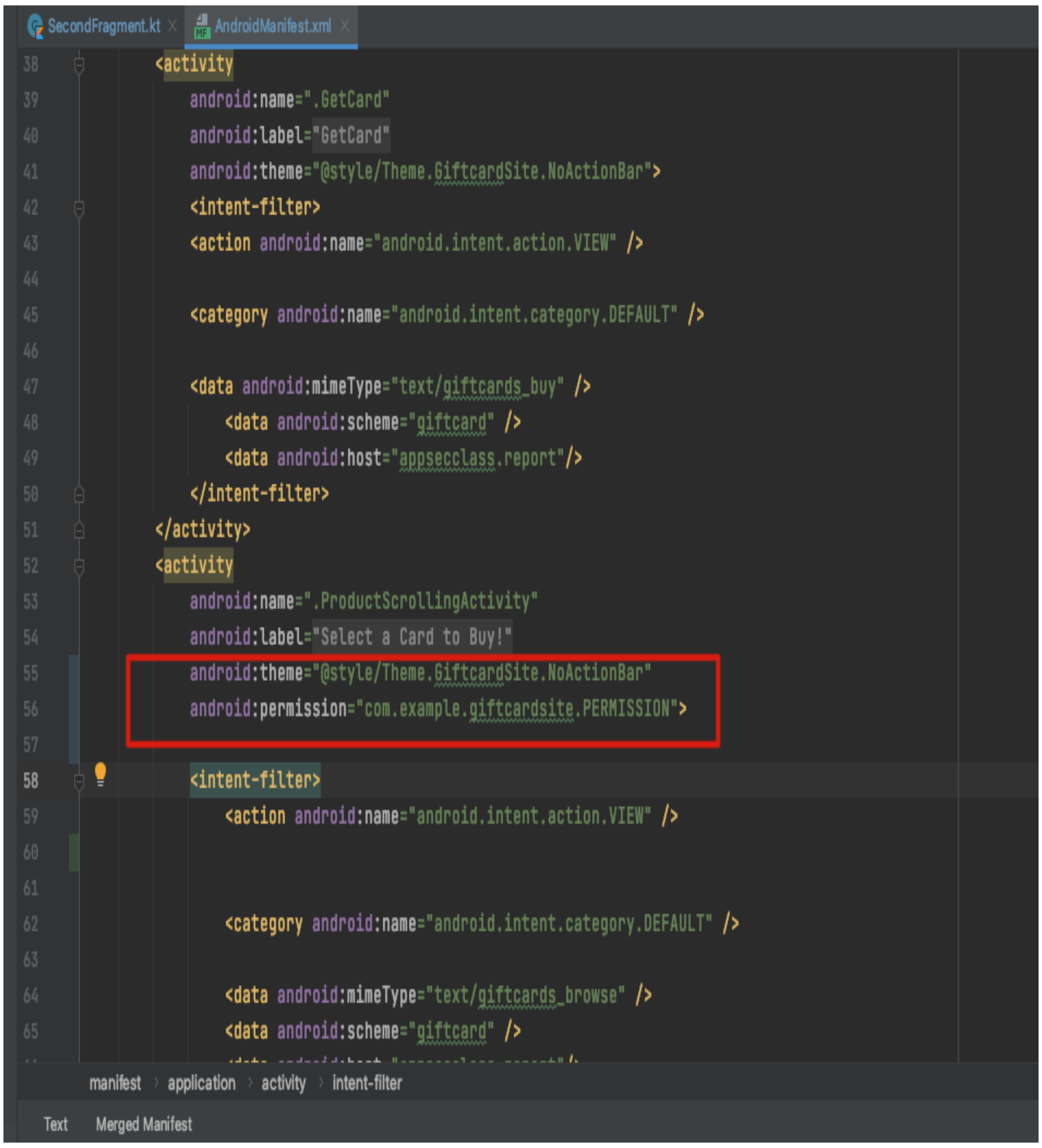
Part 2.2: Shutting out the world

The following are changes to a ‘manifest.xml’ file:

android:name="com.example.giftcardsite.CUSTOM_PERMISSION" />
<uses-permission
android:name="com.example.giftcardsite.CUSTOM_PERMISSION" />



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.giftcardsite">
    <permission android:protectionLevel="signature" android:name="com.example.giftcardsite.PERMISSION" />
    <uses-permission android:name="com.example.giftcardsite.PERMISSION" />

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION"/>
```

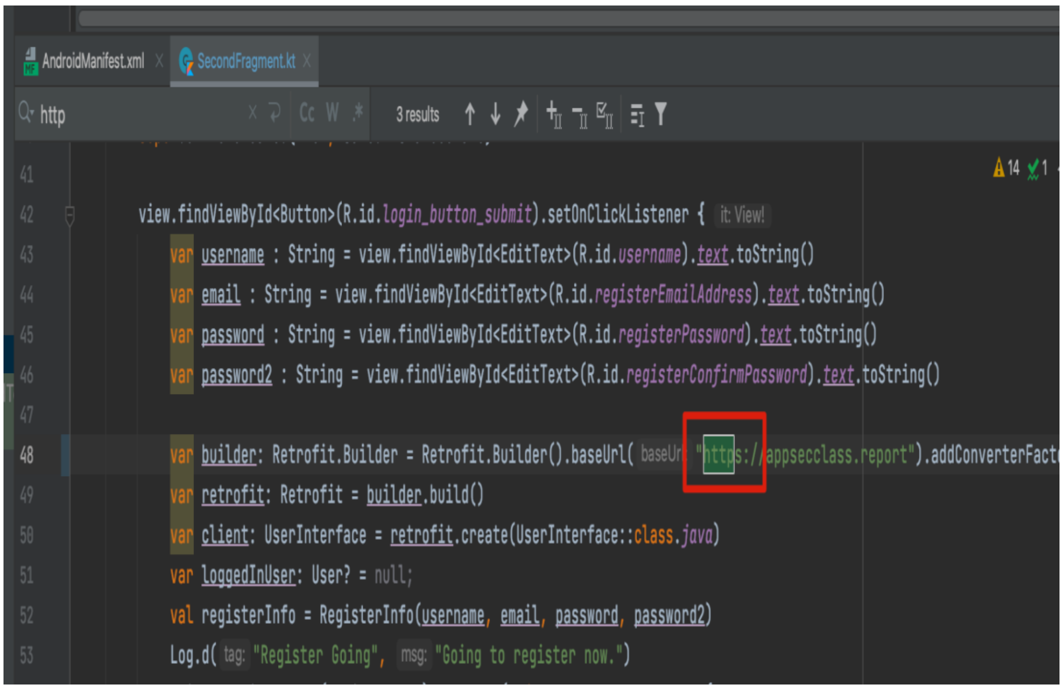


```
        <activity
            android:name=".GetCard"
            android:label="GetCard"
            android:theme="@style/Theme.GiftcardSite.NoActionBar">
            <intent-filter>
            <action android:name="android.intent.action.VIEW" />

            <category android:name="android.intent.category.DEFAULT" />

            <data android:mimeType="text/giftcards_buy" />
                <data android:scheme="giftcard" />
                <data android:host="appsecclass.report"/>
            </intent-filter>
        </activity>
        <activity
            android:name=".ProductScrollingActivity"
            android:label="Select a Card to Buy!"
            android:theme="@style/Theme.GiftcardSite.NoActionBar"
            android:permission="com.example.giftcardsite.PERMISSION">

            <intent-filter>
                <action android:name="android.intent.action.VIEW" />


                <category android:name="android.intent.category.DEFAULT" />

                <data android:mimeType="text/giftcards_browse" />
                <data android:scheme="giftcard" />
```

Part 3: Can you read me out there?

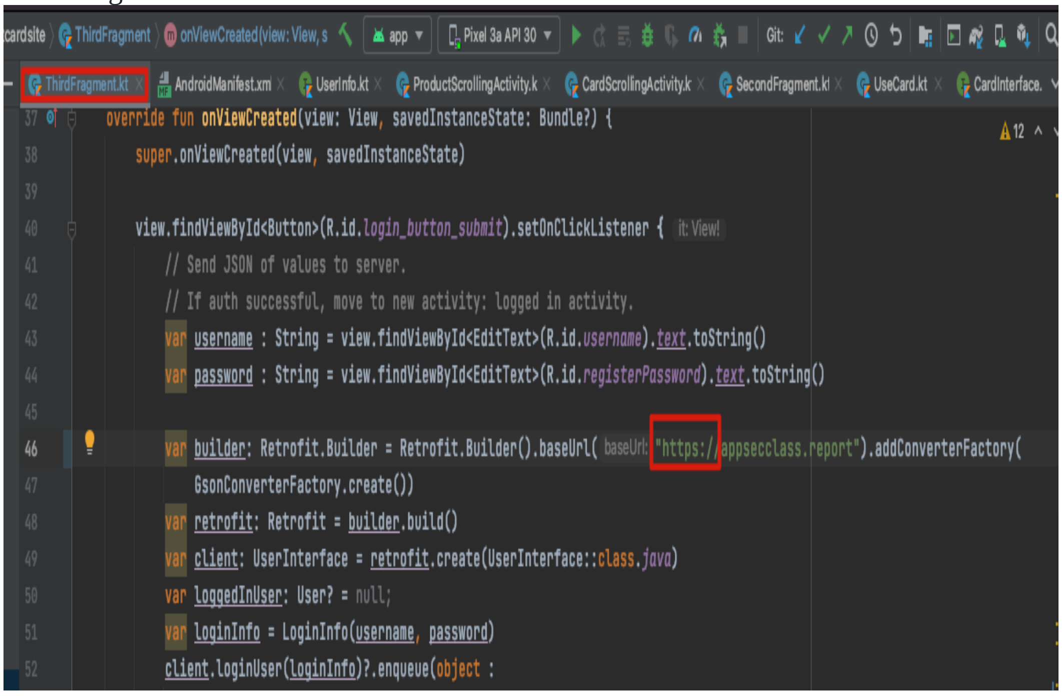
Use of ‘https’ instead of ‘http’ – in selected files:

1. SecondFragment.kt: Line #48.



```
view.findViewById<Button>(R.id.login_button_submit).setOnClickListener { it: View!
    var username : String = view.findViewById<EditText>(R.id.username).text.toString()
    var email : String = view.findViewById<EditText>(R.id.registerEmailAddress).text.toString()
    var password : String = view.findViewById<EditText>(R.id.registerPassword).text.toString()
    var password2 : String = view.findViewById<EditText>(R.id.registerConfirmPassword).text.toString()

    var builder: Retrofit.Builder = Retrofit.Builder().baseUrl( baseUrl: "https://appsecclass.report").addConverterFact
    var retrofit: Retrofit = builder.build()
    var client: UserInterface = retrofit.create(UserInterface::class.java)
    var loggedInUser: User? = null;
    val registerInfo = RegisterInfo(username, email, password, password2)
    Log.d( tag: "Register Going", msg: "Going to register now.")
```

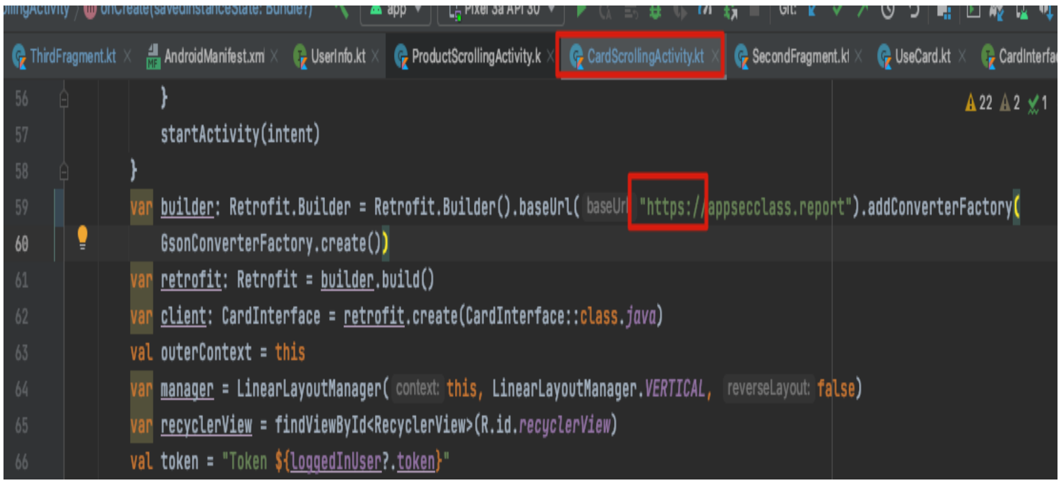
2. ThirdFragment.kt



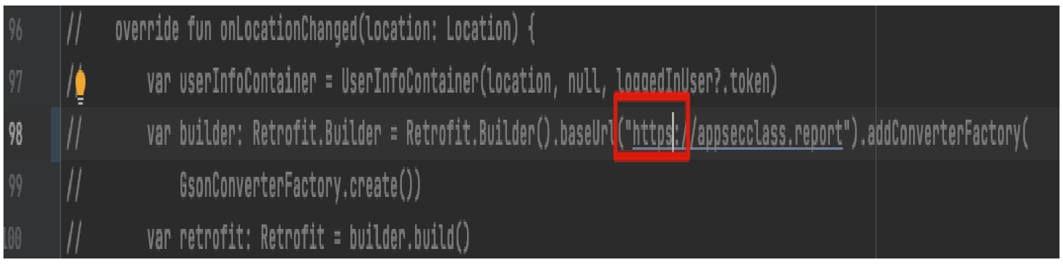
```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)

    view.findViewById<Button>(R.id.login_button_submit).setOnClickListener { it: View!
        // Send JSON of values to server.
        // If auth successful, move to new activity: logged in activity.
        var username : String = view.findViewById<EditText>(R.id.username).text.toString()
        var password : String = view.findViewById<EditText>(R.id.registerPassword).text.toString()

        var builder: Retrofit.Builder = Retrofit.Builder().baseUrl( baseUrl: "https://appsecclass.report").addConverterFactory(
            GsonConverterFactory.create())
        var retrofit: Retrofit = builder.build()
        var client: UserInterface = retrofit.create(UserInterface::class.java)
        var loggedInUser: User? = null;
        var loginInfo = LoginInfo(username, password)
        client.loginUser(loginInfo)?.enqueue(object :
```

3. CardScrollingActivity.kt

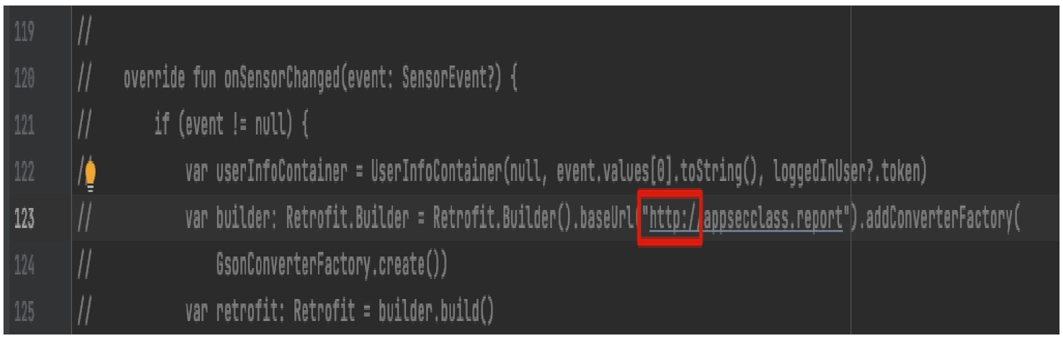
I. Line # 59
II. Line # 98
III. Line # 123



```
56            }
57            startActivity(intent)
58        }
59        var builder: Retrofit.Builder = Retrofit.Builder().baseUrl( baseUrl: "https://appsecclass.report").addConverterFactory(
60            GsonConverterFactory.create())
61        var retrofit: Retrofit = builder.build()
62        var client: CardInterface = retrofit.create(CardInterface::class.java)
63        val outerContext = this
64        var manager = LinearLayoutManager( context: this, LinearLayoutManager.VERTICAL, reverseLayout: false)
65        var recyclerView = findViewById<RecyclerView>(R.id.recyclerView)
66        val token = "Token ${loggedInUser?.token}"
```



```
96  //    override fun onLocationChanged(location: Location) {
97  //        var userInfoContainer = UserInfoContainer(location, null, loggedInUser?.token)
98  //        var builder: Retrofit.Builder = Retrofit.Builder().baseUrl("https://appsecclass.report").addConverterFactory(
99  //            GsonConverterFactory.create())
100 //        var retrofit: Retrofit = builder.build()
```



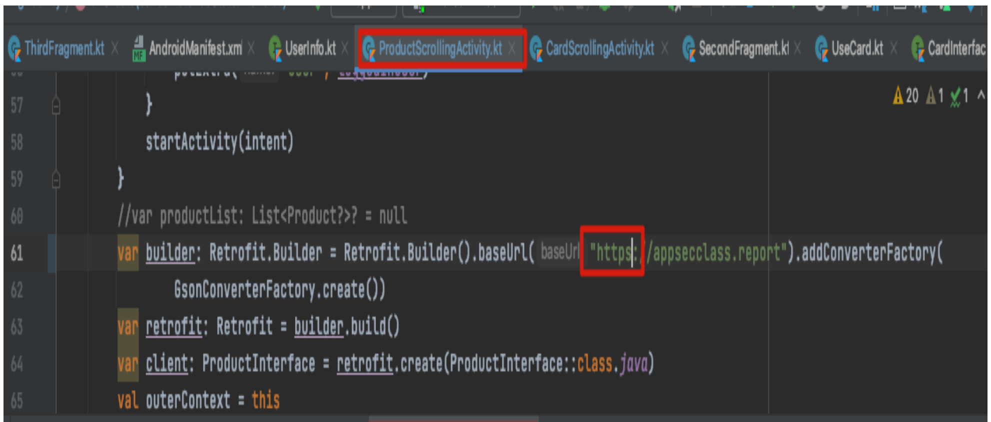
```
119 //
120 //    override fun onSensorChanged(event: SensorEvent?) {
121 //        if (event != null) {
122 //            var userInfoContainer = UserInfoContainer(null, event.values[0].toString(), loggedInUser?.token)
123 //            var builder: Retrofit.Builder = Retrofit.Builder().baseUrl("http://appsecclass.report").addConverterFactory(
124 //                GsonConverterFactory.create())
125 //            var retrofit: Retrofit = builder.build()
```

4. ProductScrollingActivity.kt

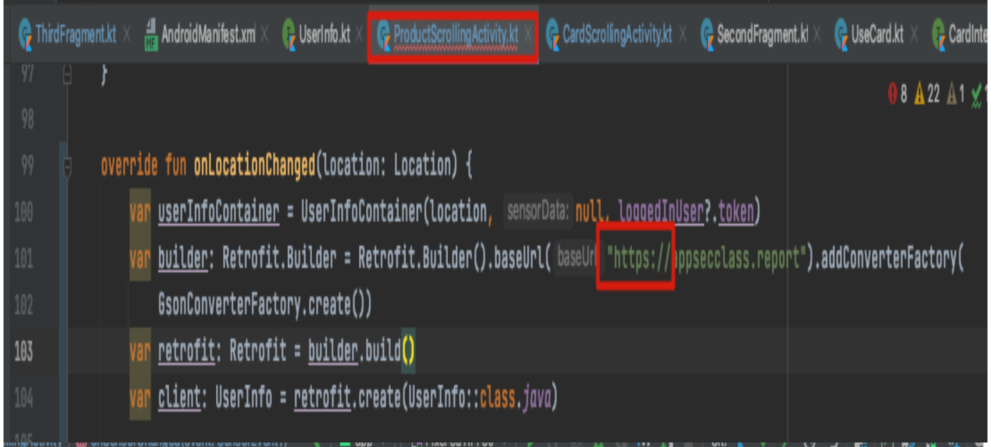
I. Line # 61

II. Line # 101

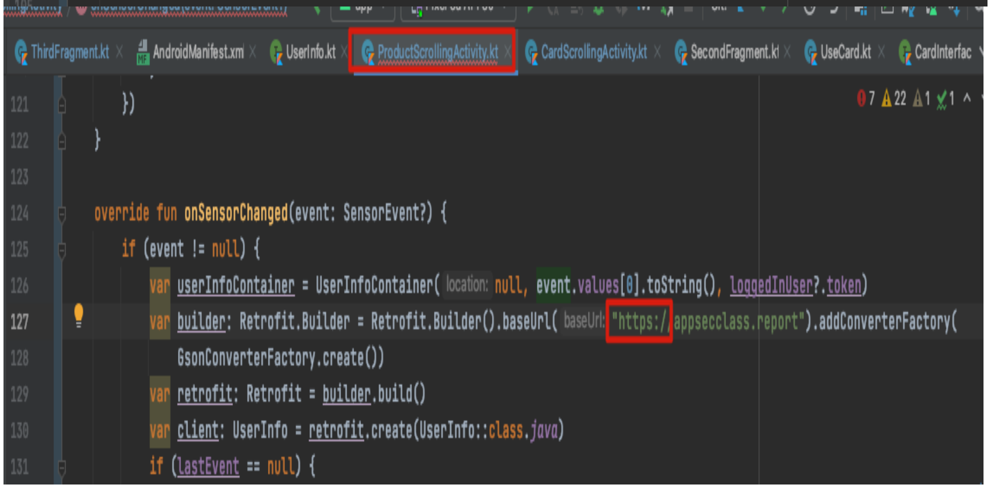
III. Line # 127



```
57              }
58              startActivity(intent)
59          }
60          //var productList: List<Product?>? = null
61          var builder: Retrofit.Builder = Retrofit.Builder().baseUrl( baseUrl: "https://appsecclass.report").addConverterFactory(
62              GsonConverterFactory.create())
63          var retrofit: Retrofit = builder.build()
64          var client: ProductInterface = retrofit.create(ProductInterface::class.java)
65          val outerContext = this
```

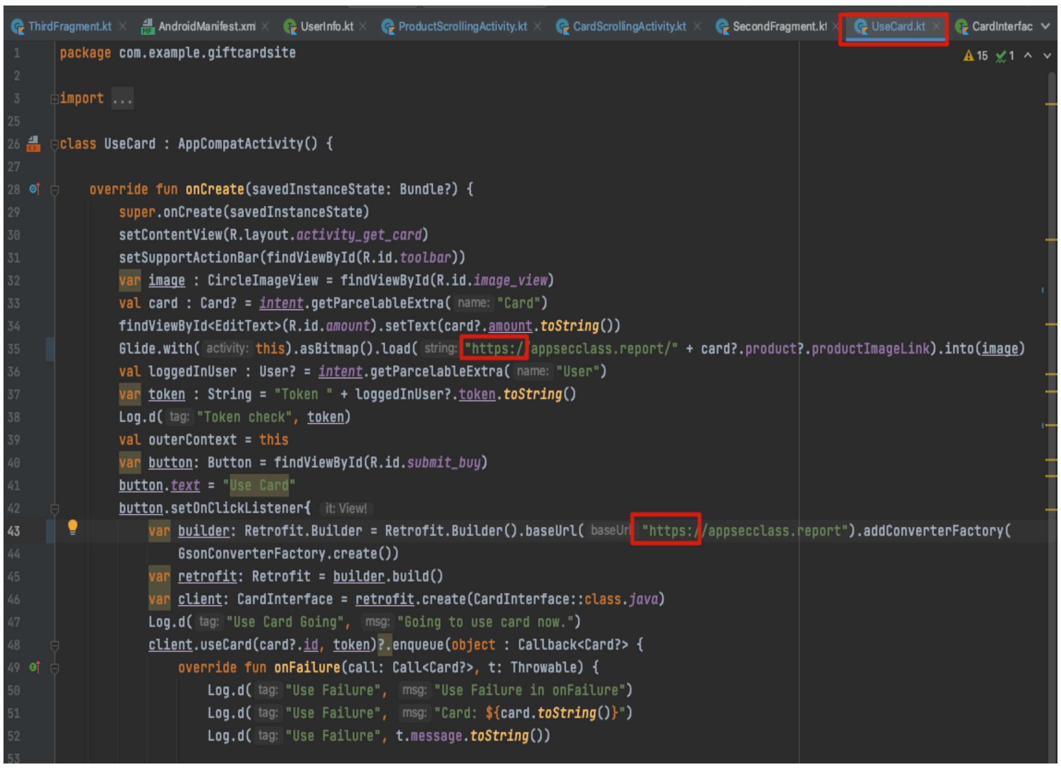


```
97      }
98
99      override fun onLocationChanged(location: Location) {
100         var userInfoContainer = UserInfoContainer(location,  sensorData: null, loggedInUser?.token)
101         var builder: Retrofit.Builder = Retrofit.Builder().baseUrl( baseUrl: "https://appsecclass.report").addConverterFactory(
102             GsonConverterFactory.create())
103         var retrofit: Retrofit = builder.build()
104         var client: UserInfo = retrofit.create(UserInfo::class.java)
```



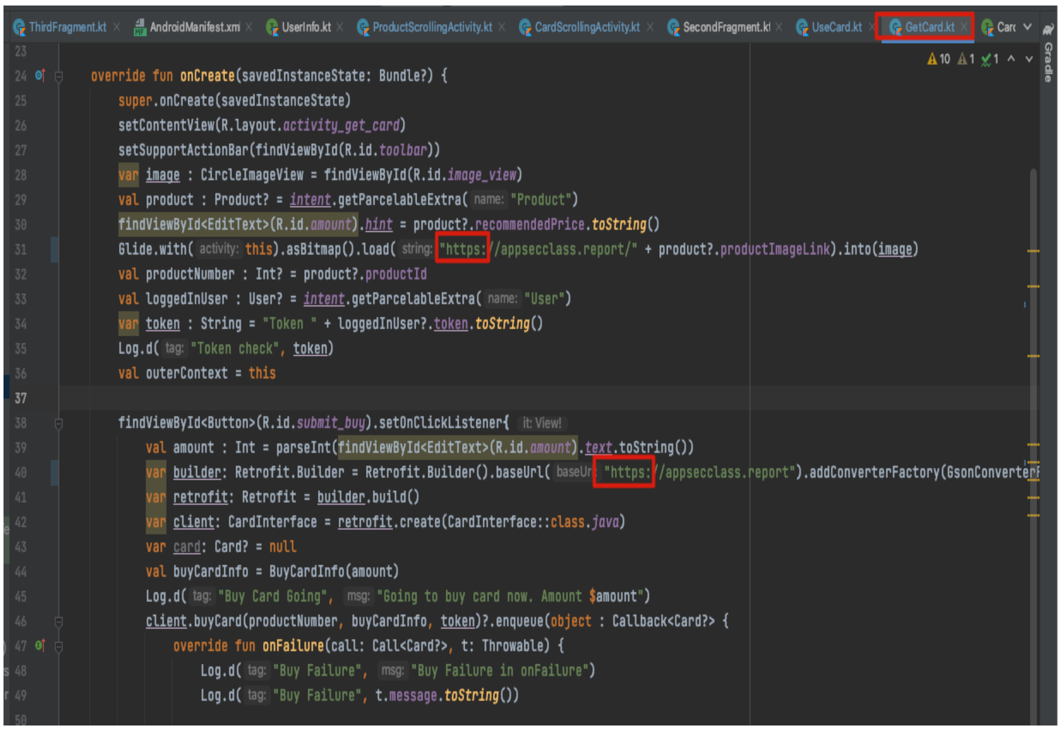
```
121         })
122     }
123
124     override fun onSensorChanged(event: SensorEvent?) {
125         if (event != null) {
126             var userInfoContainer = UserInfoContainer( location: null, event.values[0].toString(), loggedInUser?.token)
127             var builder: Retrofit.Builder = Retrofit.Builder().baseUrl( baseUrl: "https://appsecclass.report").addConverterFactory(
128                 GsonConverterFactory.create())
129             var retrofit: Retrofit = builder.build()
130             var client: UserInfo = retrofit.create(UserInfo::class.java)
131             if (lastEvent == null) {
```

5. UseCard.kt
    Line # 35
    Line # 43



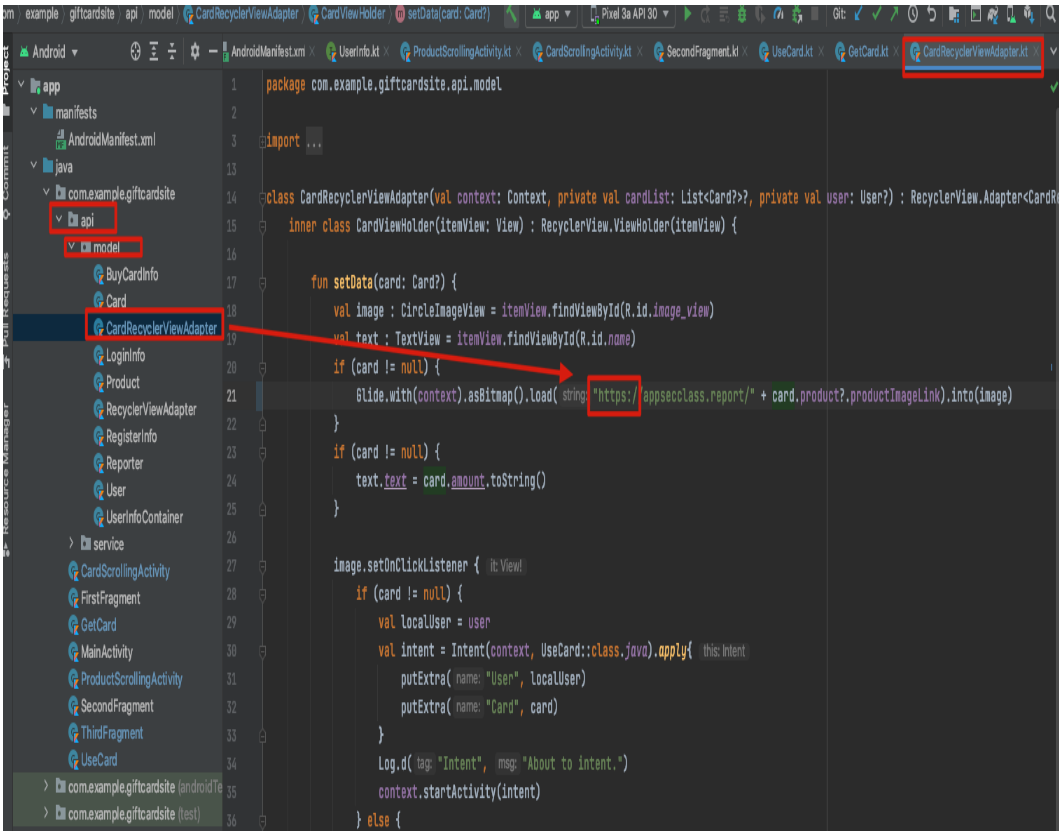
```
1   package com.example.giftcardsite
2
3   import ...
25
26  class UseCard : AppCompatActivity() {
27
28      override fun onCreate(savedInstanceState: Bundle?) {
29          super.onCreate(savedInstanceState)
30          setContentView(R.layout.activity_get_card)
31          setSupportActionBar(findViewById(R.id.toolbar))
32          var image : CircleImageView = findViewById(R.id.image_view)
33          val card : Card? = intent.getParcelableExtra( name: "Card")
34          findViewById<EditText>(R.id.amount).setText(card?.amount.toString())
35          Glide.with( activity: this).asBitmap().load( string: "https://appsecclass.report/" + card?.product?.productImageLink).into(image)
36          val loggedInUser : User? = intent.getParcelableExtra( name: "User")
37          var token : String = "Token " + loggedInUser?.token.toString()
38          Log.d( tag: "Token check", token)
39          val outerContext = this
40          var button: Button = findViewById(R.id.submit_buy)
41          button.text = "Use Card"
42          button.setOnClickListener{ it: View!
43              var builder: Retrofit.Builder = Retrofit.Builder().baseUrl( baseUrl: "https://appsecclass.report").addConverterFactory(
44                  GsonConverterFactory.create())
45              var retrofit: Retrofit = builder.build()
46              var client: CardInterface = retrofit.create(CardInterface::class.java)
47              Log.d( tag: "Use Card Going", msg: "Going to use card now.")
48              client.useCard(card?.id, token)?.enqueue(object : Callback<Card?> {
49                  override fun onFailure(call: Call<Card?>, t: Throwable) {
50                      Log.d( tag: "Use Failure", msg: "Use Failure in onFailure")
51                      Log.d( tag: "Use Failure", msg: "Card: ${card.toString()}")
52                      Log.d( tag: "Use Failure", t.message.toString())
53
```

6. GetCard.kt
    Line # 31
    Line # 40



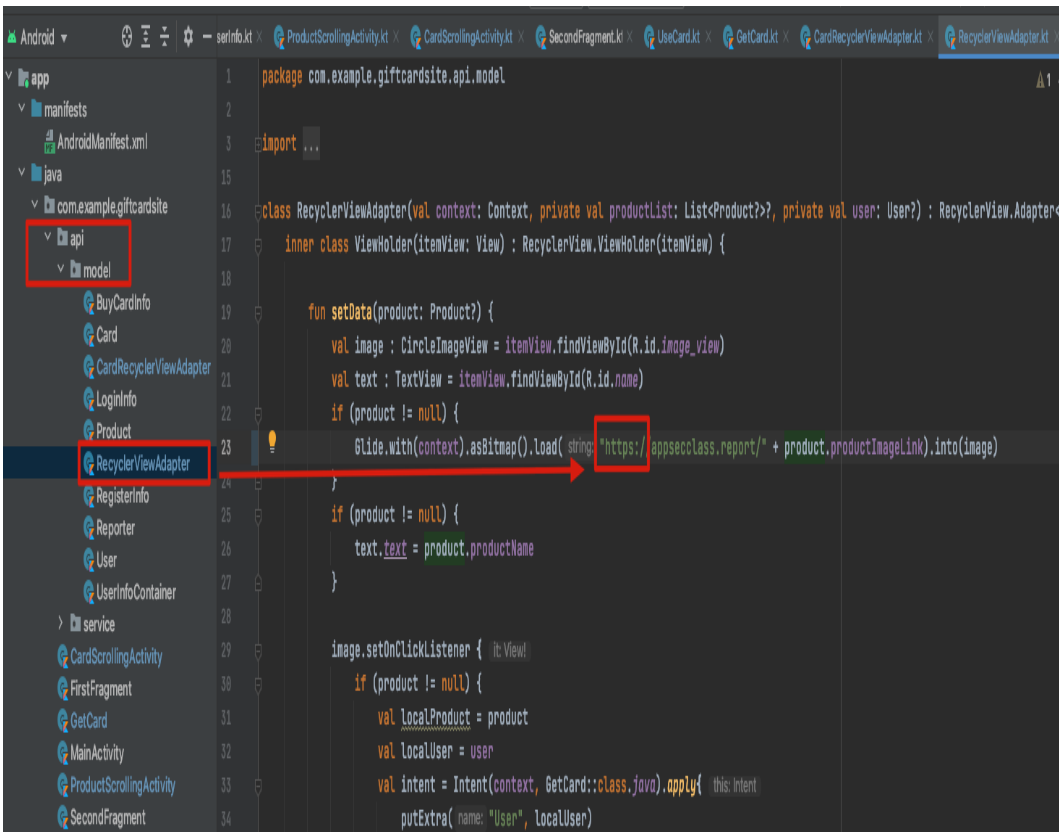
```
23
24      override fun onCreate(savedInstanceState: Bundle?) {
25          super.onCreate(savedInstanceState)
26          setContentView(R.layout.activity_get_card)
27          setSupportActionBar(findViewById(R.id.toolbar))
28          var image : CircleImageView = findViewById(R.id.image_view)
29          val product : Product? = intent.getParcelableExtra( name: "Product")
30          findViewById<EditText>(R.id.amount).hint = product?.recommendedPrice.toString()
31          Glide.with( activity: this).asBitmap().load( string: "https://appsecclass.report/" + product?.productImageLink).into(image)
32          val productNumber : Int? = product?.productId
33          val loggedInUser : User? = intent.getParcelableExtra( name: "User")
34          var token : String = "Token " + loggedInUser?.token.toString()
35          Log.d( tag: "Token check", token)
36          val outerContext = this
37
38          findViewById<Button>(R.id.submit_buy).setOnClickListener{ it: View!
39              val amount : Int = parseInt(findViewById<EditText>(R.id.amount).text.toString())
40              var builder: Retrofit.Builder = Retrofit.Builder().baseUrl( baseUrl: "https://appsecclass.report").addConverterFactory(GsonConverterF
41              var retrofit: Retrofit = builder.build()
42              var client: CardInterface = retrofit.create(CardInterface::class.java)
43              var card: Card? = null
44              val buyCardInfo = BuyCardInfo(amount)
45              Log.d( tag: "Buy Card Going", msg: "Going to buy card now. Amount $amount")
46              client.buyCard(productNumber, buyCardInfo, token)?.enqueue(object : Callback<Card?> {
47                  override fun onFailure(call: Call<Card?>, t: Throwable) {
48                      Log.d( tag: "Buy Failure", msg: "Buy Failure in onFailure")
49                      Log.d( tag: "Buy Failure", t.message.toString())
50
```

7. CardRecyclerViewAdapter.kt
    Line # 21



```
1   package com.example.giftcardsite.api.model
2
3   import ...
13
14  class CardRecyclerViewAdapter(val context: Context, private val cardList: List<Card?>?, private val user: User?) : RecyclerView.Adapter<CardR
15      inner class CardViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
16
17          fun setData(card: Card?) {
18              val image : CircleImageView = itemView.findViewById(R.id.image_view)
19              val text : TextView = itemView.findViewById(R.id.name)
20              if (card != null) {
21                  Glide.with(context).asBitmap().load( string: "https://appsecclass.report/" + card.product?.productImageLink).into(image)
22              }
23              if (card != null) {
24                  text.text = card.amount.toString()
25              }
26
27              image.setOnClickListener {  it: View!
28                  if (card != null) {
29                      val localUser = user
30                      val intent = Intent(context, UseCard::class.java).apply{  this: Intent
31                          putExtra( name: "User", localUser)
32                          putExtra( name: "Card", card)
33                      }
34                      Log.d( tag: "Intent",  msg: "About to intent.")
35                      context.startActivity(intent)
36                  } else {
```

8. RecyclerViewAdapter.kt
   Line 23

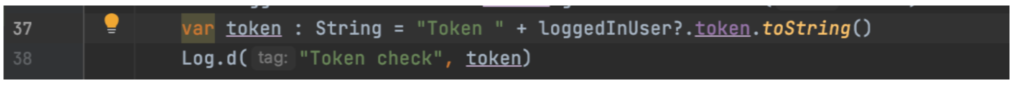


```
1   package com.example.giftcardsite.api.model
2
3   import ...
15
16  class RecyclerViewAdapter(val context: Context, private val productList: List<Product?>?, private val user: User?) : RecyclerView.Adapter<
17      inner class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
18
19          fun setData(product: Product?) {
20              val image : CircleImageView = itemView.findViewById(R.id.image_view)
21              val text : TextView = itemView.findViewById(R.id.name)
22              if (product != null) {
23                  Glide.with(context).asBitmap().load( string: "https://appsecclass.report/" + product.productImageLink).into(image)
24              }
25              if (product != null) {
26                  text.text = product.productName
27              }
28
29              image.setOnClickListener {  it: View!
30                  if (product != null) {
31                      val localProduct = product
32                      val localUser = user
33                      val intent = Intent(context, GetCard::class.java).apply{  this: Intent
34                          putExtra( name: "User", localUser)
```

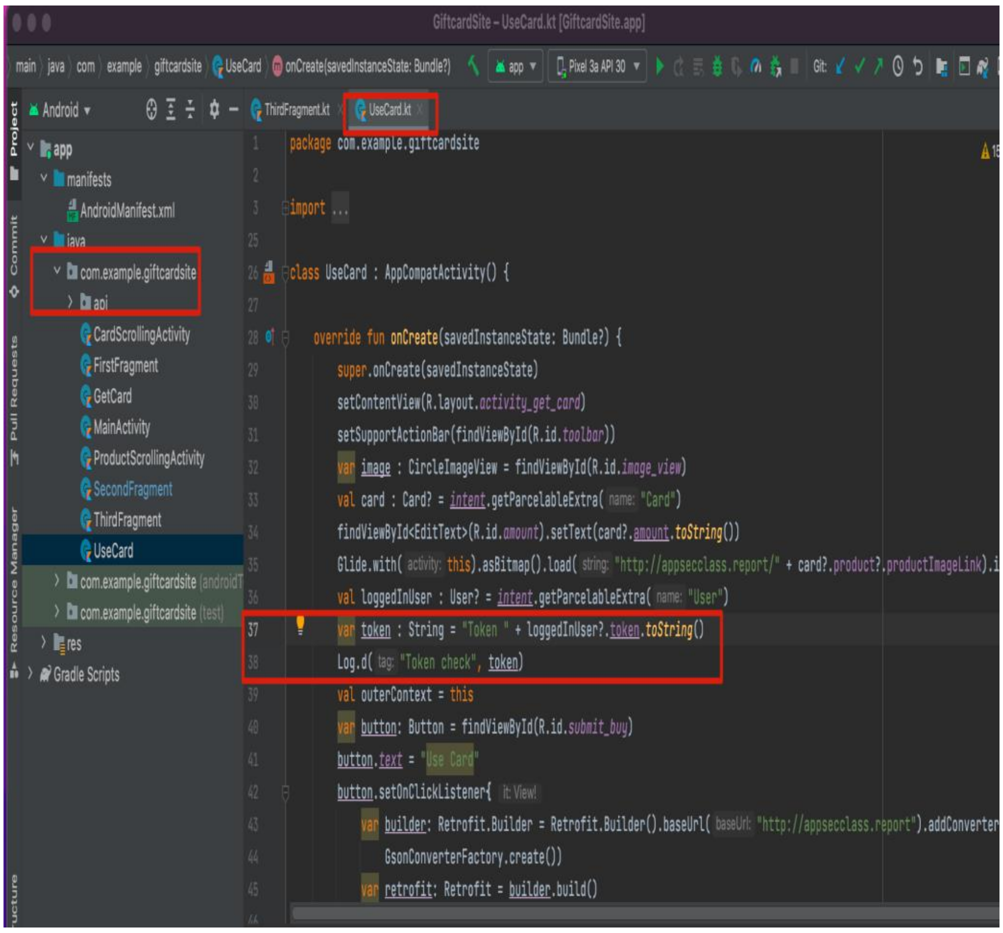
Part 4: Oops, was that card yours?

A vulnerability in the ‘UseCard.kt file is on line 37 & 38. Is a word ‘Token’ and a user name. Access to a username would permit changing a token to a desired value. In such situation use of a OAuth2 token is preferred – it is more secure.

1. UseCard.kt:



```
37      var token : String = "Token " + loggedInUser?.token.toString()
38      Log.d( tag: "Token check", token)
```

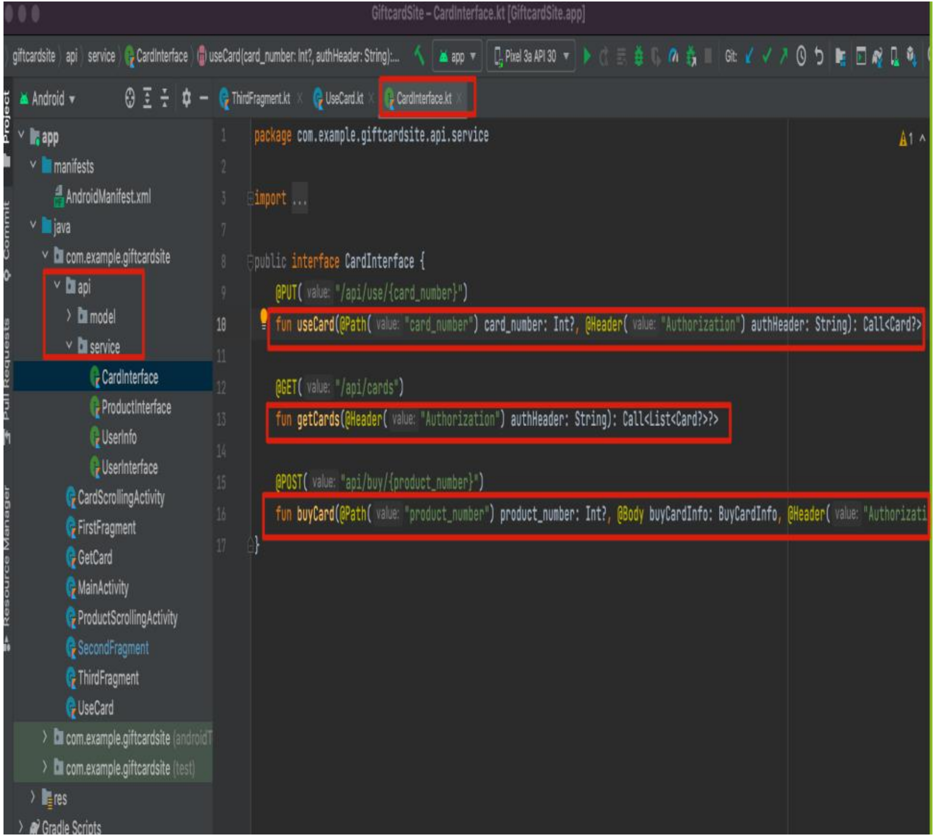


```kotlin
package com.example.giftcardsite

import ...

class UseCard : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_get_card)
        setSupportActionBar(findViewById(R.id.toolbar))
        var image : CircleImageView = findViewById(R.id.image_view)
        val card : Card? = intent.getParcelableExtra("Card")
        findViewById<EditText>(R.id.amount).setText(card?.amount.toString())
        Glide.with(this).asBitmap().load("http://appsecclass.report/" + card?.product?.productImageLink).i
        val loggedInUser : User? = intent.getParcelableExtra("User")
        var token : String = "Token " + loggedInUser?.token.toString()
        Log.d("Token check", token)
        val outerContext = this
        var button: Button = findViewById(R.id.submit_buy)
        button.text = "Use Card"
        button.setOnClickListener{
            var builder: Retrofit.Builder = Retrofit.Builder().baseUrl("http://appsecclass.report").addConverter
                GsonConverterFactory.create())
            var retrofit: Retrofit = builder.build()
```

2. **CardInterface.kt**

Tells server which card to use, since there is no authentication in the app. By getting a gift card number and username, - an attacker can make call to /api/use endpoint since no data protection takes place. The following vulnerability exists:

Line numbers are 10, 13, & 16



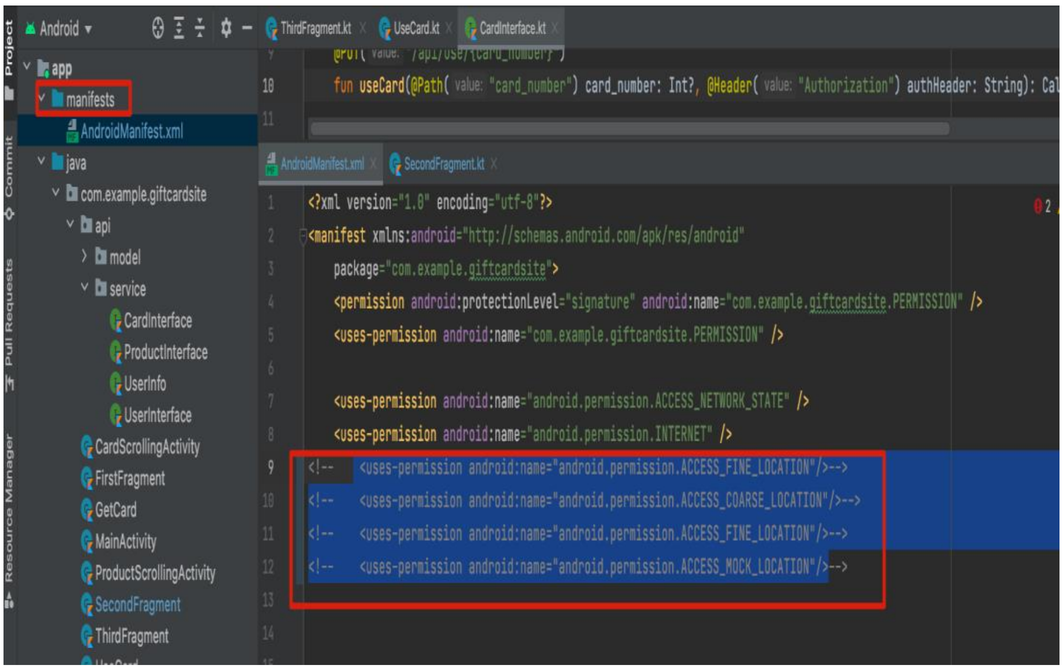
```kotlin
1   package com.example.giftcardsite.api.service
2
3   import ...
7
8   public interface CardInterface {
9       @PUT("/api/use/{card_number}")
10      fun useCard(@Path("card_number") card_number: Int?, @Header("Authorization") authHeader: String): Call<Card?>
11
12      @GET("/api/cards")
13      fun getCards(@Header("Authorization") authHeader: String): Call<List<Card?>?>
14
15      @POST("api/buy/{product_number}")
16      fun buyCard(@Path("product_number") product_number: Int?, @Body buyCardInfo: BuyCardInfo, @Header("Authorizati
17  }
```

Part 5: Privacy is Important

Privacy invasive metrics – about the user.

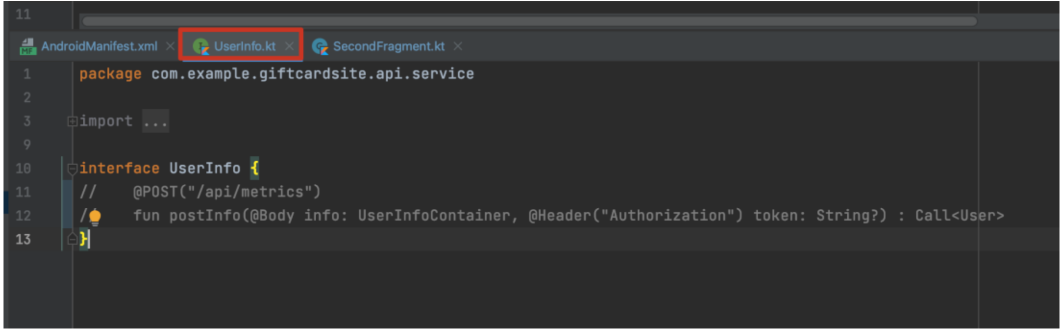
A data collection restriction of privacy and permissions is preferred.

Commenting all metrics collecting code in all areas. And all permissions that are not necessary. Unnecessary to fully carry full functionality to buy, browse, use gift cards.

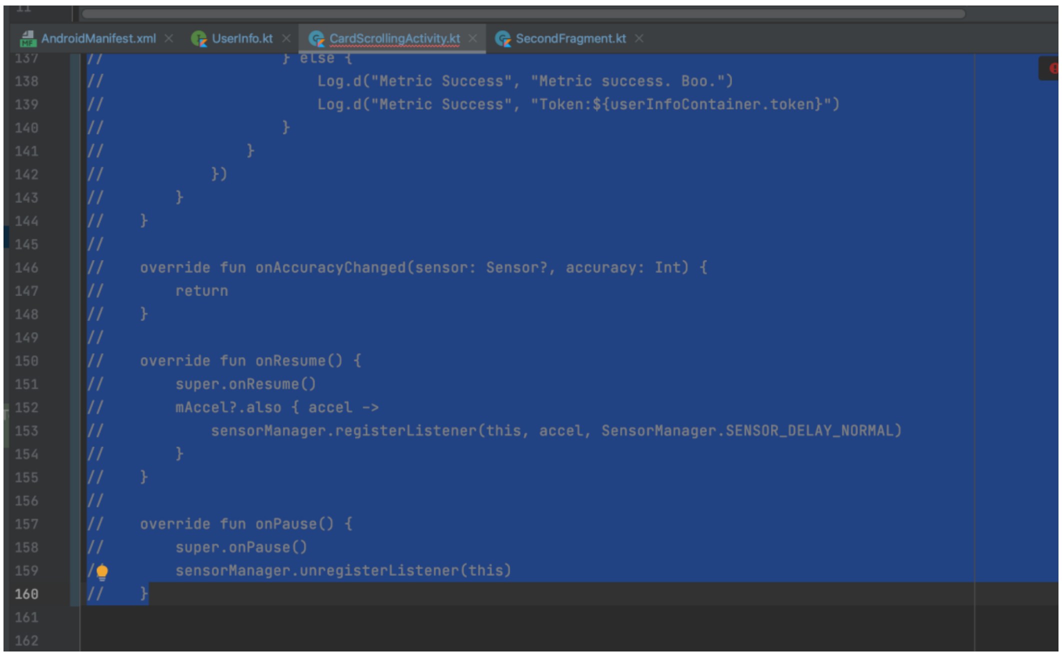


```
    @PUT( value: "/api/use/{card_number}")
    fun useCard(@Path( value: "card_number") card_number: Int?, @Header( value: "Authorization") authHeader: String): Cal
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.giftcardsite">
    <permission android:protectionLevel="signature" android:name="com.example.giftcardsite.PERMISSION" />
    <uses-permission android:name="com.example.giftcardsite.PERMISSION" />

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
<!--    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>-->
<!--    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>-->
<!--    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>-->
<!--    <uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION"/>-->
```

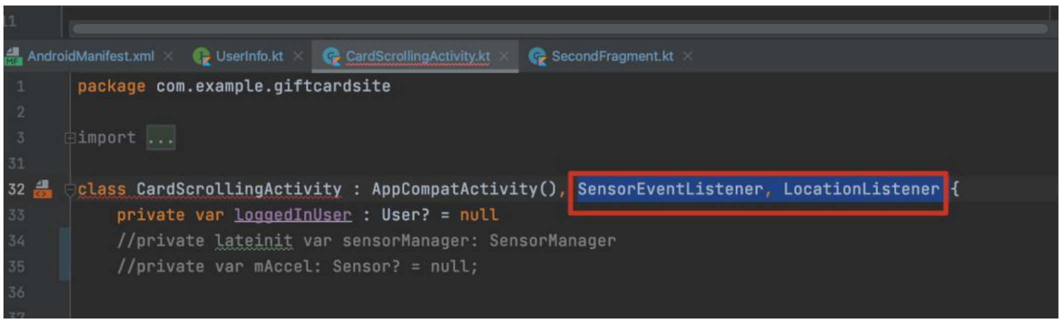


```
package com.example.giftcardsite.api.service

import ...

interface UserInfo {
//    @POST("/api/metrics")
//    fun postInfo(@Body info: UserInfoContainer, @Header("Authorization") token: String?) : Call<User>
}
```

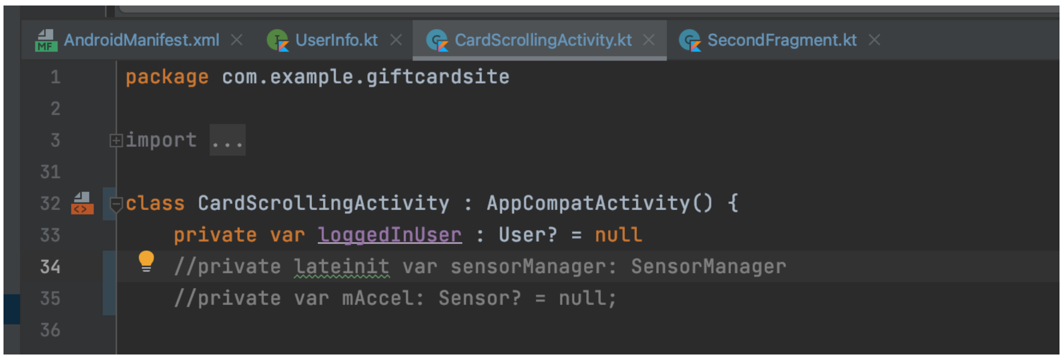


```
//                    } else {
//                        Log.d("Metric Success", "Metric success. Boo.")
//                        Log.d("Metric Success", "Token:${userInfoContainer.token}")
//                    }
//                }
//            })
//        }
//    }
//
//    override fun onAccuracyChanged(sensor: Sensor?, accuracy: Int) {
//        return
//    }
//
//    override fun onResume() {
//        super.onResume()
//        mAccel?.also { accel ->
//            sensorManager.registerListener(this, accel, SensorManager.SENSOR_DELAY_NORMAL)
//        }
//    }
//
//    override fun onPause() {
//        super.onPause()
//        sensorManager.unregisterListener(this)
//    }
```

Removed:

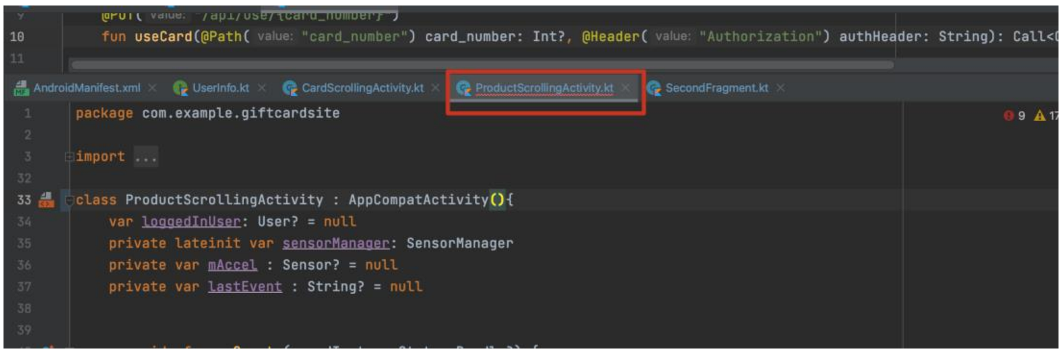


```
package com.example.giftcardsite

import ...

class CardScrollingActivity : AppCompatActivity(), SensorEventListener, LocationListener {
    private var loggedInUser : User? = null
    //private lateinit var sensorManager: SensorManager
    //private var mAccel: Sensor? = null;
```



```
package com.example.giftcardsite

import ...

class CardScrollingActivity : AppCompatActivity() {
    private var loggedInUser : User? = null
    //private lateinit var sensorManager: SensorManager
    //private var mAccel: Sensor? = null;

```

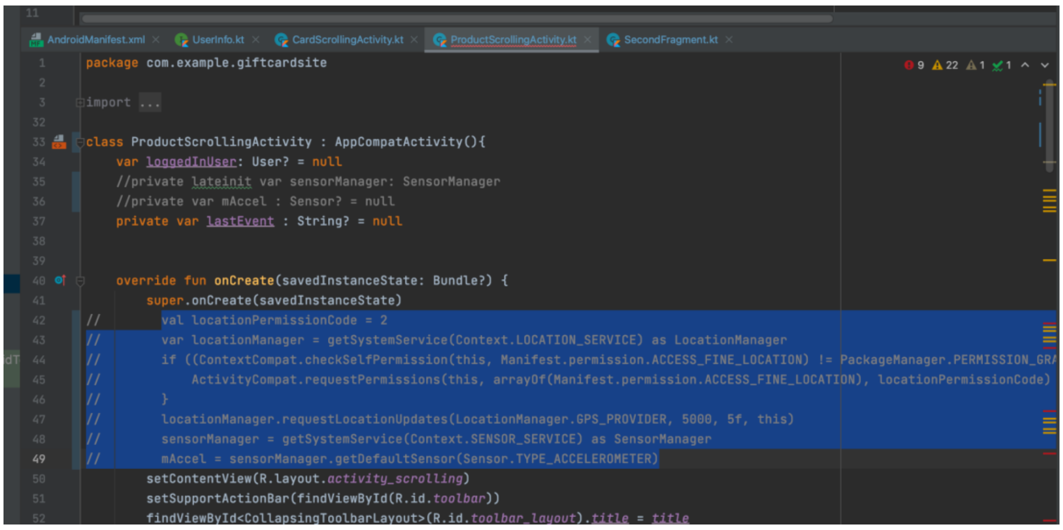
Removed:



```
    @PUT( value: "/api/use/{card_number}")
    fun useCard(@Path( value: "card_number") card_number: Int?, @Header( value: "Authorization") authHeader: String): Call<(

```

```
package com.example.giftcardsite

import ...

class ProductScrollingActivity : AppCompatActivity(){
    var loggedInUser: User? = null
    private lateinit var sensorManager: SensorManager
    private var mAccel : Sensor? = null
    private var lastEvent : String? = null

```

Comment out:



```
package com.example.giftcardsite

import ...

class ProductScrollingActivity : AppCompatActivity(){
    var loggedInUser: User? = null
    //private lateinit var sensorManager: SensorManager
    //private var mAccel : Sensor? = null
    private var lastEvent : String? = null


    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
//        val locationPermissionCode = 2
//        var locationManager = getSystemService(Context.LOCATION_SERVICE) as LocationManager
//        if ((ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRA
//            ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.ACCESS_FINE_LOCATION), locationPermissionCode)
//        }
//        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 5000, 5f, this)
//        sensorManager = getSystemService(Context.SENSOR_SERVICE) as SensorManager
//        mAccel = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER)
        setContentView(R.layout.activity_scrolling)
        setSupportActionBar(findViewById(R.id.toolbar))
        findViewById<CollapsingToolbarLayout>(R.id.toolbar_layout).title = title
```