

**ISTANBUL TECHNICAL UNIVERSITY**  
**Faculty of Computer Science and Informatics**

**ANDROID AI COMBINATION RECOMMENDATION APPLICATION**

# **INTERNSHIP PROGRAM REPORT**

**UMUTCAN DOĞAN**  
**150200712**

**FALL / 2023**

**Istanbul Technical University**

**Faculty of Computer Science and Informatics**

**INTERNSHIP REPORT**

Academic Year: 2022/2023  
Internship Term: ☐ Summer ☐ Spring ☒ Fall

**Student Information**

Name Surname: UMUTCAN DOĞAN  
Student ID: 150200712  
Department: Computer Engineering  
Program: 100% English  
E-Mail: doganumu20@itu.edu.tr  
Mobile Phone: +90 (506) 132 4373  
Pursuing a Double Major? ☐ Yes  
☒ No  
  
In the Graduation Term? ☐ Yes  
☒ No  
  
Taking a class at Summer  
School? ☐ Yes  
☒ No

**Institution Information**

Company Name: TEB ARF Teknoloji A.Ş.  
Department: Software  
Web Address: <https://www.tebarf.com.tr/>  
PostalAddress: Dudullu OSB Mah. Des-2 Cad. Teknopark Binası  
No:8/90 34776 Ümraniye/İstanbul

### **Authorized Person Information**

Department: Software  
Title: Department Leader  
Name Surname: İrfan Çınar  
Corporate E-Mail: irfan.cinar@tebarf.com.tr  
Corporate Phone: 90-216-2254200

### **Internship Work Information**

Internship Location: ☒ Turkey  
☐ Abroad  
Internship Start Date: 16.01.2023  
Internship End Date: 10.02.2023  
Number of Days Worked: 20  
During your internship, did you have insurance? ☒ Yes, I was insured by İTÜ.  
☐ Yes, I was insured by institution.  
☐ No, I did my internship abroad.  
☐ No.

# Table of Contents

<b>1</b>	<b>INFORMATION ABOUT THE INSTITUTION . . . . .</b>	<b>1</b>
<b>2</b>	<b>INTRODUCTION . . . . .</b>	<b>2</b>
<b>3</b>	<b>DESCRIPTION AND ANALYSIS OF THE INTERNSHIP PROJECT . . . . .</b>	<b>2</b>
<b>4</b>	<b>CONCLUSIONS . . . . .</b>	<b>13</b>
<b>5</b>	<b>REFERENCES . . . . .</b>	<b>14</b>
<b>6</b>	<b>APPENDIX . . . . .</b>	<b>15</b>

# 1 INFORMATION ABOUT THE INSTITUTION

TEB Arf Teknoloji A.Ş. aims to implement the RD projects developed within its organization jointly with the technopark infrastructures where technology interaction is at maximum level and to offer high value-added software services. It also intends to become one of the leading companies in the field of information solutions in Turkey.

Since 2020, it has been conducting its innovative activities in the IT sector from its Izmir and Istanbul offices.

## Artificial Intelligence and Data Science:

We develop data science projects creating value from data and thus, enabling data-based decision making in business processes by using structured or unstructured data types and/or internal/external data sources with advanced analytics methods. We are working on the projects which are developed through the cooperation of BÜDOTEK and TÜBİTAK and which have received incentive approval from the Ministry of Industry and Technology.

## Risk Analytics

Application and behaviour scoring, customer income estimation, risk – limit optimization and early warning and collection optimization models are developed, allowing for the end-to-end and holistic management of the loan lifecycle. Prediction models are developed by processing the data obtained from internal/external resources using advanced machine learning algorithms. Thus, credit risk management processes are rendered more effectively and efficiently.

## Fraud / Abuse Analytics

Early detection of fraud and/or abuse, especially in loan applications and card transactions, is ensured with the models developed using advanced machine learning and deep learning algorithms. Thus, in addition to scenario-based fraud alarms, predictions and relationship detections can be made quickly and easily.

## Pricing Analytics

Advanced analytical projects that will support business processes are developed in the fields of calculation and measurement of the impact of price changes for all products; implementation of cost strategies based on customer demands, behaviours, and market trends and automation of the pricing process. With customer segmentation, customer price sensitivity determination and price optimization models developed in this context, the price that will provide optimum benefits at the product, customer or corporate level is determined.

## Customer Analytics

Internal/external data obtained from customers and customer behaviours are systematically processed with advanced machine learning and deep learning algorithms and thus, predictive models that will guide critical business decisions are developed. Thanks to predictive advanced analytical models such as segmentation and tracking of such spheres as customer churn rates, product and channel trends and customer footprints; personalized processes and offers adding value to customers are developed.

## 2 INTRODUCTION

The project is constructing a weather app that fetches information about the weather of the current location from a server, recognizes the uploaded photos using Artificial Intelligence and suggests a combination of clothes according to the temperature of the current weather.

## 3 DESCRIPTION AND ANALYSIS OF THE INTERNSHIP PROJECT

Conceptually components of Android are learned. These components include:

- Activities - represent the UI of the app and handle user interactions.
- Services - perform background tasks without a user interface.
- Broadcast Receivers - respond to system-wide broadcast announcements.
- Content Providers - manage a structured set of data available to all apps.
- Fragments - represent part of an activity's UI and can be combined and reused in different activities.

What are activities?

Activities in Android are used to represent the User Interface (UI) of an app. They handle user interactions and can display information to the user. Each screen in an app is typically represented by an individual activity. An app can have multiple activities, and the user can navigate between them using the app's navigation flow. Activities are responsible for creating a window for UI, managing the lifecycle of the UI and handling the user interactions. When an activity is launched, it's added to the app's task and placed in the foreground of the device's screen. When the user navigates away from the activity, it's either paused or stopped, depending on the system's needs.

I need to add a service for fetching the weather degree from a weather API.

What is a service?

Services in Android are used to perform background tasks without a user interface. They are used to perform operations that don't need to be directly interacted with by the user and can continue to run even when the app is not in the foreground.

Services can be used to play music, download files, perform network operations, and more. They can also be used to update a UI element, even when the app is not visible. Services run on the main thread of the app, so it's important to make sure that they do not block the main thread or perform long-running operations, to avoid ANR (Application Not Responding) errors.

A service can be started or bound by another component, such as an activity, and it can continue to run even if the component that started or bound it is destroyed.

Firstly, I used okhttp3 and JsonObject to parse the data that gets returned by the request from okhttp3. Also, to not get an ANR (Activity not responding) error, I used AsyncTask library.

What is ANR?

ANR stands for "Application Not Responding" and it is an error message displayed by the Android operating system when it detects that an app has stopped responding to user input. This can occur if the app is doing a long-running operation that blocks the main thread, such as network or file I/O, or if it is stuck in an infinite loop. To avoid ANR errors, developers should ensure that their app's main thread is not blocked for too long and that they use background threads or services for long-running tasks, such as AsyncTask.

I also created a github repository for the changes that I made on the code. I committed the first version where the application works. In this version, when the get data button is pushed, weather information is fetched and according to temperature, the app says what you should wear.

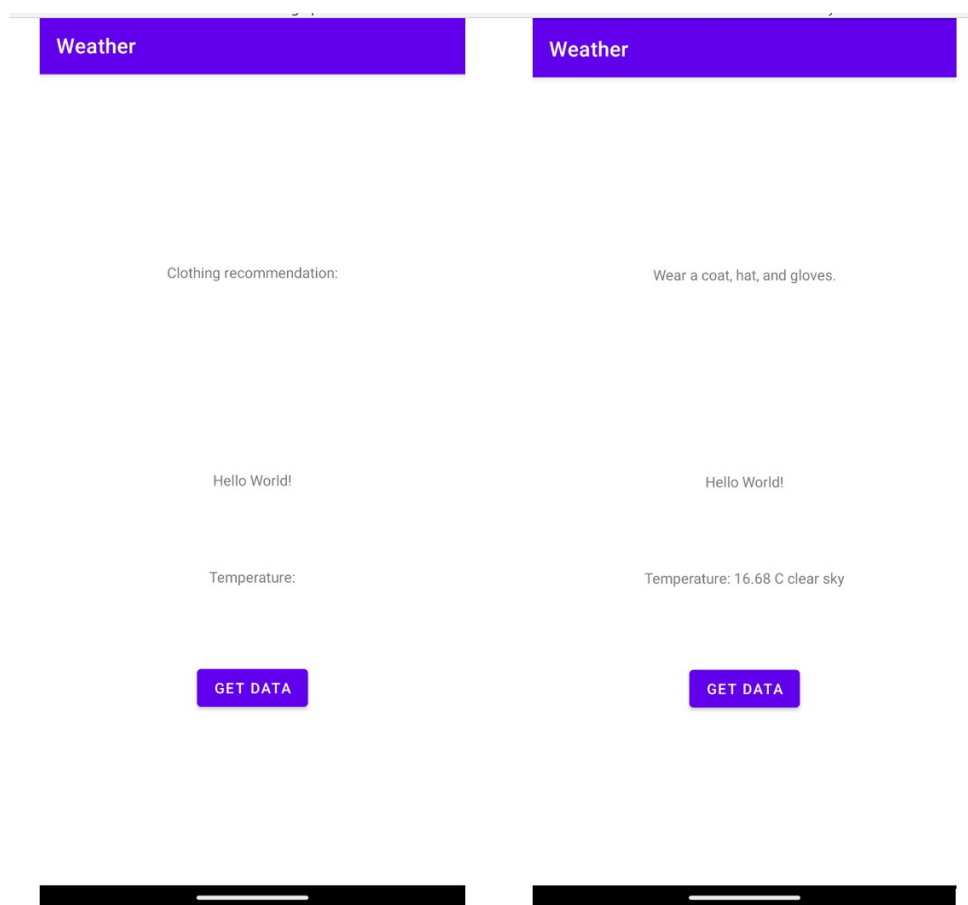


Figure 1: When the button is pressed, it displays the data

For the fetching, my instructor said I should use the Retrofit library since it also ensures resolving the ANR error.

What is Retrofit?

Retrofit is a type-safe REST client for Android and Java, developed by Square. It makes it easy to consume RESTful web services in Android or Java application by abstracting away the details of low-level HTTP requests, allowing developers to focus on the business logic of the application.

Retrofit allows you to define a simple interface that describes the endpoints of a web service, and then automatically generates the code to make the appropriate HTTP requests and parse the responses. This makes it easy to build a client for a web service, without having to manually construct and parse the URLs and request bodies.

Retrofit supports several converters for deserializing the API responses, such as Gson, Jackson, Moshi, and Protobuf. This means the user can parse the response in desired format.

Retrofit also supports additional features such as request interceptors, connection pooling, and caching. It's very easy to use, and it's widely used in the android development community.

I added the necessary code for using retrofit to fetch the weather data. I also added Gson since it was needed at parsing fase. I created a class for the data that gets fetched from the internet.

This class included the variables that I need, for example temperature and description of the weather.

I also implemented an interface for calling the data from the server. I committed the version 2.

What is an interface?

In Android, an interface is a way for a class to specify a set of methods that another class must implement. It is similar to an abstract class, but it can only contain method signatures and fields, and cannot contain any implementation details.

My task is to get the location from the mobile to automatically show the weather. I will also add the city name to the screen, and try to make the User Interface look better.

I used geocoder to find the city with the given latitude and longitude which I got from android location service. Geocoder is a class that is used to map an address or location to a set of latitude and longitude coordinates, or vice versa. It can be used to convert a street address to a geographic location, or to determine the address at a given latitude and longitude. It can also be used to return a list of addresses that are known to describe the area at the given latitude and longitude. The Geocoder class requires the android.location.Geocoder permission in the manifest file.

After writing the required permissions to manifest file, I used LocationManager class to get the location from gps of the mobile device.

The LocationManager class provides access to the system location services. These services allow applications to obtain periodic updates of the device's geographical location, or to fire an application-specified Intent when the device enters the proximity of a given geographical location.

Now that I have the device's last known location, I used the Geocoder class to get the address of the location.

The rest of the code was pretty much the same, I gave the city to call function and it got the weather from the API.

When button is pressed the app requests for permission and if the permission was not given then the rest of the code would not work. Thus I implemented onRequestPermissionsResult class



to run the code when the permission is given. This is a callback method that is triggered when the user grants or denies a permission request. This method is called by the Android system when the user interacts with the permission request dialog. The method takes three parameters: the request code (an integer that the user provided when the user made the request), the permissions array, and the grant results array. The grant results array is an integer array that contains the result of each permission request.

Also, getting the last location was not working and the location was returning null (empty). It was also possible that the locationManager. getLastKnownLocation(bestProvider) method is returning null because there is no last known location. In this case, I tried using the request LocationUpdates() method to get the current location. This function is updating the location every given period of time. This time the code worked.

I committed the working version 3 of the code to Github.

I added a loading dialog when fetching data, since it took some time and app looked frozen. To add the loading dialog I used ProgressBar class to create a loading dialog while the getWeatherData function is trying to fetch information from the API. I added the code for xml file. I controlled its visibility in onCreate method of my MainActivity class. The app now shows a loading animation when the data is being fetched.

Now, I will focus on User Interface and try to add a bottom navigation bar with 3 fragments, first is home , which will open when the app starts, second is getting the weather, and third is getting recommendation. I did the following steps:

- Added the dependency for the bottom navigation view in the build.gradle file.
- Created 3 fragments, one for each menu item (home, get weather data, and get recommendation).
- Created a layout file for the bottom navigation view which included a bottom navigation view widget and a FrameLayout to hold the fragments.
- Inflated the layout file and set it as the content view in the activity that would host the bottom navigation view.
- In the onCreate() method of the activity, set up the bottom navigation view by setting the listener for the item selected events and attaching the home fragment to the FrameLayout.
- In the listener for the item selected events, switched between the fragments based on the item id and replaced the fragment in the FrameLayout.
- Set the activity that would host the bottom navigation view as the launcher activity in the AndroidManifest.xml file.
- Set the theme for the application to NoActionBar in the styles.xml file.
- Created menu resources and inflated them with the bottom navigation view.
- Used the following libraries for the bottom navigation view: com.google.android.material:material androidx.navigation:navigation-fragment-ktx androidx.navigation:navigation-ui-ktx

Now the application has three pages. The app, first wants permission of location at the opening. When the permission of location is not given it exits from the program. I want to add a feature which by swapping, it changes the pages.

For swapping pages, I needed to use the ViewPager library.

What is ViewPager?

A ViewPager is a layout manager that allows the user to flip left and right through pages of data. It is commonly used in conjunction with a TabLayout to implement a tabbed interface for displaying different pages of data. The ViewPager class is part of the Android Support Library and is widely used for creating swipeable views in Android apps. It can be used to display a collection of fragments, images, or other views. The user can swipe left or right to navigate through the pages, and the ViewPager will automatically handle the scrolling and animation between pages.

Since when changing the pages, the navigation bar's current item should change as well, I used the `addOnPageChangeListener()` method of the ViewPager to update the navigation bar accordingly.

Unfortunately ViewPager was deprecated and I used ViewPager2 instead. Now, the user can swap left or right to change the pages.

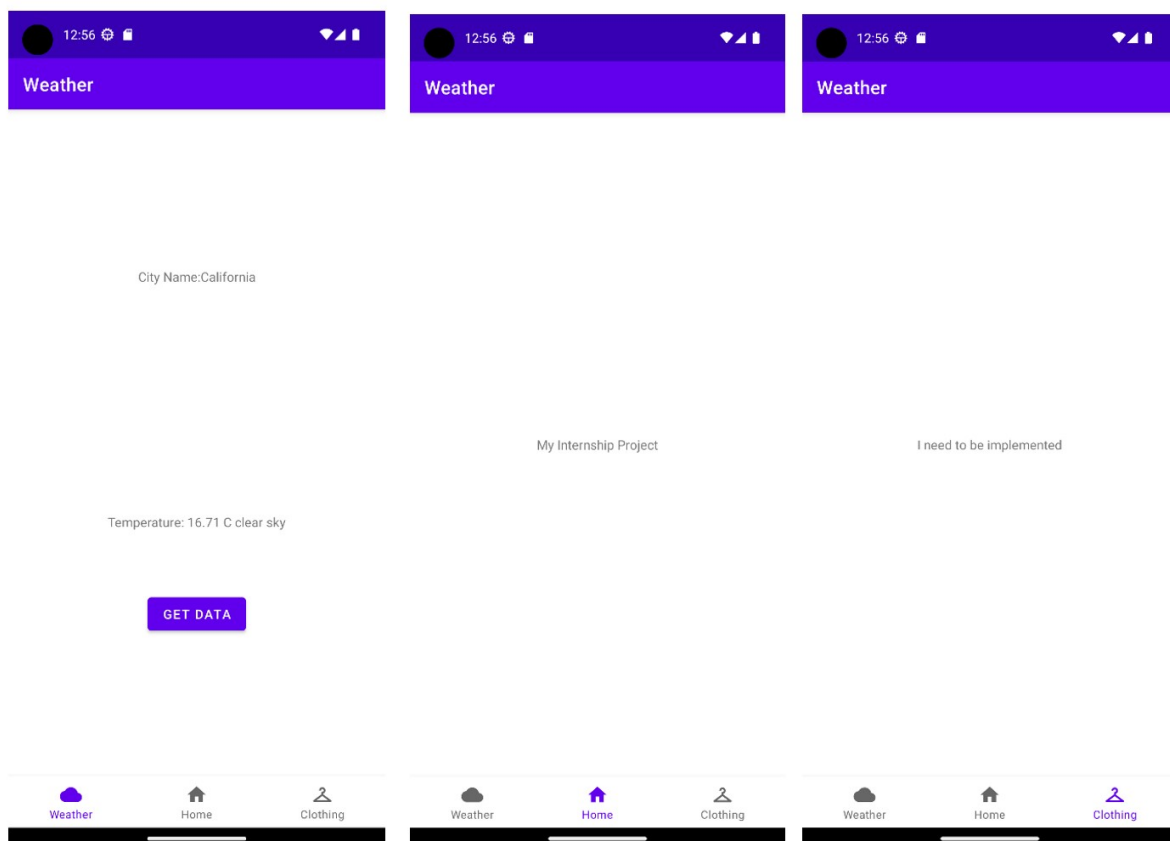


Figure 2: There are now three screens that can be changed by swapping or pressing the navigation button

Now my app have 3 pages. The home page is just saying this is my project. The weather page has a button and fetches the weather when pressed. The third page is the clothing recommendation. I did not implemented this page yet. In this page the user will upload their cloths by selecting

from gallery or simply taking a photo. The app will use an ai to recognize the cloths and make a recommendation based on weather. I will deal with the ai part later. Now I will focus on implementing the clothing fragment.

I also need to fetch the weather temperature from the weather fragment to use it in clothing fragment.

I implemented a life cycle observer for opening files to get the image. It has a onCreate method which basically opens the files application. I also implemented a method showimage selectiondialog, which shows a dialog for selection of image or taking a photo from the camera app. If the user selects "From gallery" option, the observer that created will call its method selectimage which will open the activity to select an image. After that in onactivityresult function of activityresultcallback, the returned image will be used for ai purposes.

To handle the result, I used the registerForActivityResult method, which is the recommended way to handle results from an activity or fragment starting another activity, instead of using the deprecated startActivityForResult method. This allows for a more robust and efficient way of handling results, as it ensures proper lifecycle management and eliminates the need for manual callbacks. Overall, the code in the ClothFragment allows for user to upload image either from camera or gallery and process it for further usage.

In the Clothfragment, it uses the ActivityResultRegistry and DefaultLifecycleObserver classes to handle the selection of images by the user. The MyLifecycleObserver class is created in the onCreate() method of the fragment and it is responsible for creating an ActivityResultLauncher object that is used to launch the activity for selecting an image. The ActivityResultCallback interface is used to handle the returned image Uri.

In the onCreateView() method of the fragment, a button is created and set up to show an alert dialog when clicked. The alert dialog provides the user with the option to select an image from the gallery or take a photo using the camera. When the user selects the "From Gallery" option, the selectImage() method of the MyLifecycleObserver class is called, which launches the activity for selecting an image. The returned image Uri is then handled in the onActivityResult() method of the ActivityResultCallback interface.

For the ClothFragment,I used the ActivityResultRegistry and an ActivityResultLauncher to register and launch the intent for selecting an image from the gallery. This way, the lifecycle of the fragment is considered in the process of launching the intent and handling the result, and it eliminates the need for using the deprecated startActivityForResult method. In the openCamera() function, I would also use a similar approach, using the ActivityReultRegistry and an Activity ResultLauncher to register and launch the intent for opening the camera, and handle the result in the same way as I did for the image selection from the gallery.

However, since I an using the ActivityResultRegistry to handle the result of the intent, it is unnecessary to create a separate LifecycleObserver for the camera intent.

Therefore, I updated my code, and deleted the class that I implemented, and used on request permission result for handling.

What does on requestpermissionresult function do?

onRequestPermissionsResult() is a method that is called when the user responds to a runtime permission request. The method is passed three arguments: the request code, an array of the permissions that were requested, and an array of the grant results for each of those

permissions. This method allows the app to handle the user's response and take appropriate action, such as requesting the permission again, disabling a feature, or shutting down the app if the permission is essential.

I also handled the permission requests. Now the requests shows after we click the button, not in the beginning of the app as it was before.

In order to do that, I handled the `onrequestpermissionresult` function in fragments, separately. Also when the get data button is pressed, after it shows the weather, it disappears. I also need to implement a method for getting the weather temperature from weather fragment to cloth fragment. I should use bundle for that. After that I can start to learn image recognition and AI.

I realised that I made a mistake in my code, which was when the viewpager checks for the page change, the code in there was doing nothing, it was creating new fragment everytime, which might result in memory leak.

After that I realised that the viewpager automatically deletes the pages, so that it does not result in memory leak. I changed the `FragmentManager` to `SupportFragmentManager`, since I am using only three pages, and for memory leak I added a code but it does not work properly.

My initial code sets an adapter for the viewPager and provides a `createFragment` method that returns a new fragment based on the position passed in. However, it doesn't add the fragments to any layout. The fragments will be added to the viewPager, which will handle displaying the fragments within its layout. The viewPager will automatically add the fragments to its layout, which in this case is the layout with id fragment container.

Unfortunately, the `ViewPager2` does not have a way to specify the container layout for its fragments. Its container is the parent layout of the `ViewPager2`. In other words, if I want to change the container of the fragments, I will need to change the parent layout of the `ViewPager2` to the desired layout.

The id of the container where the fragments will be added is specified in the layout file associated with your `MainActivity` class. When I call the `onCreate` method of my `MainActivity`, the layout file is inflated and all the views and their corresponding ids specified in the layout file are created and linked to the corresponding variables in my `MainActivity`.

I tried to understand how the viewpager knows how to add to correct frame layout, but it just looks at the first frame that it sees. I think this is a bad design, they should have used something like finding the id of the frame. I wasted a bit of time there.

I also migrated to Viewpager from Viewpager 2, to handle the memory leak by myself, which I failed to achieve for now.

I also added bundle to pass the temperature variable from weather fragment to cloth fragment for ai recommendation.

I added a piece of code, and solved the memory problem. The problem was when I change pages, the page that I change from was getting reset, which I did not want. I added the code:

```
mPager.setOffscreenPageLimit(2);
```

This code solved the issue. It sets the number of pages that are not visible to 2, which is what I have on my app. So when the user swaps pages it does not reset the pages that are invisible.

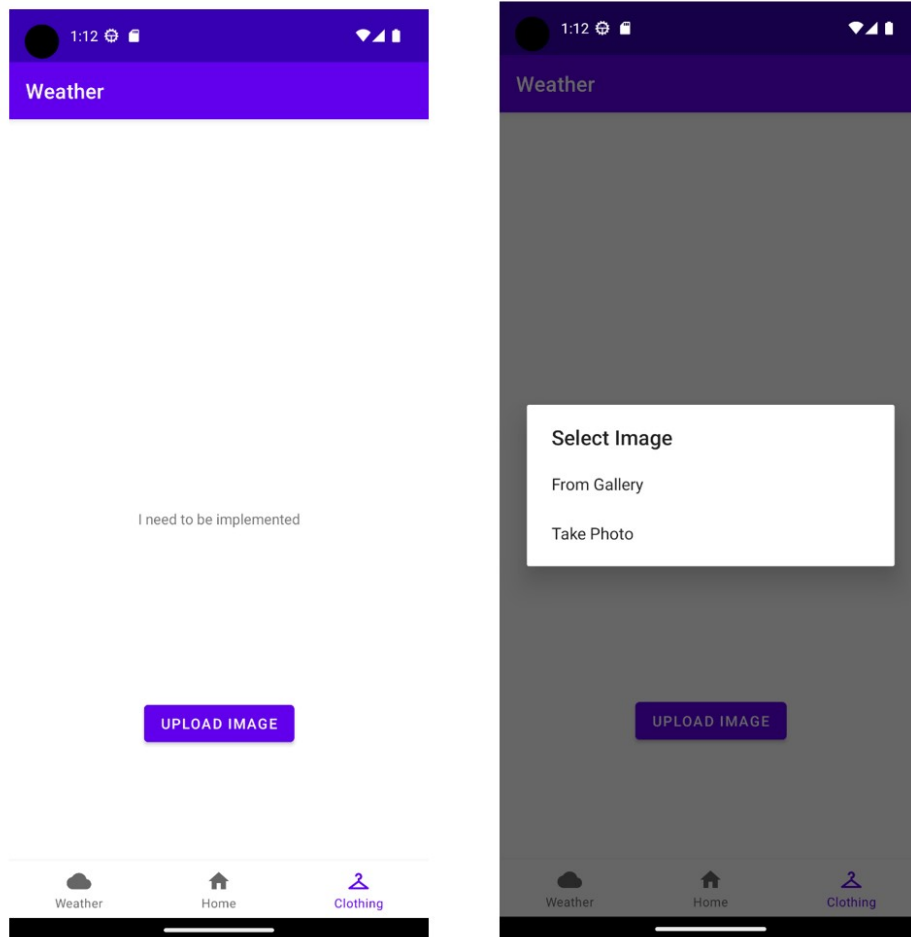


Figure 3: When the button is pressed there are two options of uploading from the gallery or taking the photo.

I am starting the AI part of my project. I am going to use Tensorflow lite, since it is specially designed for mobile devices, compact and fast.

What is tensorflow lite?

TensorFlow Lite is a lightweight version of TensorFlow that can be used for mobile and embedded devices. It allows the user to run machine learning models on-device, which can be useful for applications that require low latency or offline functionality.

Also I will use teachable machine from google, to train my ai, since training in my own computer can take more time because I do not have a powerful device.

What is Teachable machine?

Teachable Machine is a web-based tool developed by Google that allows the user to train machine learning models quickly and easily without any prior knowledge of coding or machine learning. It is designed to be accessible to a wide range of users, from beginners to experts.

One of the advantages of using Teachable Machine is that it allows the user to train models using user's own data, rather than relying on pre-trained models. This means that the user can train models that are specific to their use case, and that are able to perform well on specific dataset.

The training process of Teachable Machine is generally fast, as it uses a technique called transfer learning which allows the model to learn from a pre-trained model that has already learned to recognize a wide variety of objects. So, it will not take much time to train the model.

It also allows the user to export the models to different platforms, such as TensorFlow.js, which allows the user to run the models on the web or in mobile apps.

I will implement two ai's, one for women, one for men, since the cloths are different for each.

The app should first ask the gender therefore. I also added the code where the app first asks for the gender, and then shows the upload image dialog. If the gender is already asked before, the app does not ask it again, it only shows the upload image dialog.

I thought that it will be a good idea to display the images that the user uploads on screen.

When the user selects an item an uploads it to the application, a popup window will appear on top of the fragment, and show all the images that have been uploaded including the newly uploaded image. The user can swipe up to see the images, so it will be a vertical list. I used RecyclerView for managing this.

A RecyclerView is a flexible and efficient version of a ListView in Android. It is used to display a collection of items in a scrolling list, and allows for reusing views to improve performance. It provides built-in support for animations, item decorations, and item dividers. It also supports custom layouts for the items, and allows for easy integration with other parts of the Android framework, such as the ViewModel and LiveData classes. RecyclerView is part of the Android support library and can be added to an app by including the following dependency in the app's build.gradle file: "implementation 'com.android.support:recyclerview-v7:version'"

- First, I will need to create a layout file for the item view of the images. This layout should contain an ImageView to display the image.
- Next, I will need to create a custom adapter that extends RecyclerView.Adapter and implements the necessary methods to bind the image data to the item view. In the onBindViewHolder method, I can use Glide or Picasso library to load the images into the ImageView.
- In the ClothFragment class, I will need to initialize the RecyclerView and set the custom adapter. I will also need to provide the image data to the adapter, which can be done by calling the adapter.setData(imageList) method.
- In the onCreateView method of the ClothFragment class, I will need to inflate the layout file containing the RecyclerView and find the RecyclerView by ID.
- Finally, I will need to handle the user's swipe gestures to show or hide the images. I can use a SwipeRefreshLayout to handle the swipe gestures and refresh the images.

I Created an PhotoAdapter class that extends RecyclerView.Adapter<ImageViewHolder>. In this class, I overrided the onCreateViewHolder method to inflate the layout of each item in the list and the onBindViewHolder method to bind the data to the item views.

Now, when the user uploads a photo It shows on screen.

Next, I will implement or change the method for getting images so that it can return multiple images not only one.

I also added it. Now the user can select multiple photos from the gallery and upload them to the application.

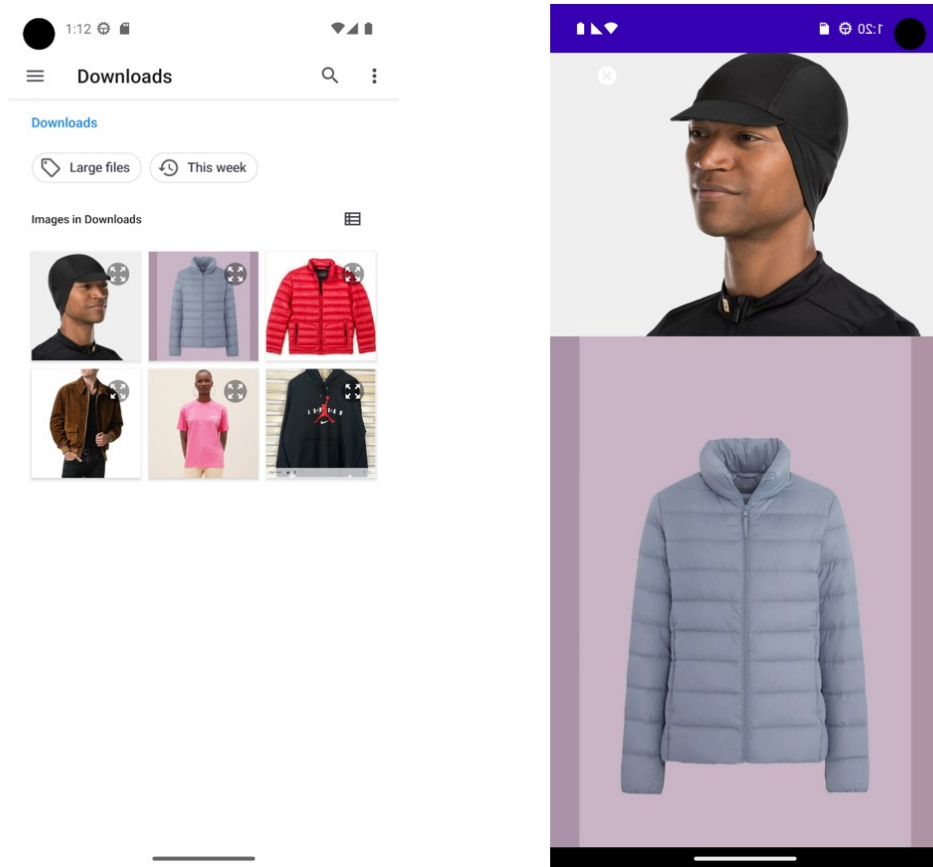


Figure 4: Selecting the photos from the gallery

I trained the teachable machine with the images that I found on internet, I created classes for all the images that I wanted the ai to recognize, there is 14 categories:

"tshirt", "sweater", "jacket", "coat", "jeans", "shorts", "shoes", "boots", "tanktop", "cap", "gloves", "scarf", "beanie", "skirt"

The ai code is a function that takes a Bitmap image as an input and uses a pre-trained machine learning model Tensorflow Lite to classify it into one of 14 different clothing categories.

- First, the function creates a new instance of the model using the "Model.newInstance()" method, passing in the application context of the current activity.
- Next, it creates a new TensorBuffer object named "inputFeature0" with a fixed size of 1x224x224x3, and with a data type of float32. This TensorBuffer will be used as the input for the machine learning model.
- It then creates a new ByteBuffer object named "byteBuffer" with a capacity of 4ximageSizeimageSize3, where imageSize is a integer value of 224. It sets the byte order of the buffer to match the native order of the device.

- It creates an array of integers named "intValues" with a size of imageSize\*imageSize, and uses the "getPixels()" method of the Bitmap image to fill it with the pixel values of the image.
- It then iterates through the "intValues" array, and for each pixel, it extracts the red, green, and blue channel values, scales them to the range of 0-1, and stores them in the "byteBuffer" object, in the order of red, green, blue.
- The inputFeature0 is then loaded with the byteBuffer.
- It runs the model inference using the "process()" method of the Model class, passing in the inputFeature0 ByteBuffer as the input. This returns an Outputs object containing the results of the inference.
- It extracts the output of the model, which is a ByteBuffer object, from the Outputs object, and assigns it to a variable named "outputFeature0".
- It then gets a float array of the confidences of each of the 14 different classes from the "outputFeature0" ByteBuffer using the "getFloatArray()" method.
- It iterates through the confidences array, and finds the index of the class with the highest confidence score.
- It creates an array of 14 class names, and logs the class with the highest confidence score to the console.
- Finally, it releases the model using the "close()" method, to free up resources.

For now, the answer is only displayed in logcat. In following days I will make it so that the answer is on the page somewhere, and also make a recommendation function, so that it will also recommend from uploaded pictures.

There was something wrong with the camera intent, so I solve the error by writing some code.

For recommendation part I will create a data structure so that it will have a string array of 14 size which are String[] classes="tshirt", "sweater", "jacket", "coat", "jeans", "shorts", "shoes", "boots", "tanktop", "cap", "gloves", "scarf", "beanie", "skirt"; it will also have 14 bitmap array list, it will have a function to add to the list, with the given class name. For example function(bitmap,"cap") will add the bitmap image to cap bitmap array which is one of the 14 bitmap arrays. It will also have a function for getting the required array for example function2(cap) will return the cap image array.

I changed my mind to use a hashmap instead of a costume data structure since writing that was not very compact. Here is a step by step explanation of the ai code that I wrote:

- The getRandomcloth method takes a string (representing the type of clothing) and a PhotoAdapter object as parameters.
- The method retrieves a list of Bitmap images that correspond to the given clothing type from the getArray method.
- If the list of images is not empty, a random image is selected and added to the images list of the PhotoAdapter object.



- The method also adds the clothing type to the predictions list of the PhotoAdapter object and increments the prediction index value.
- The aiRecommendation method takes a PhotoAdapter object as a parameter.
- It retrieves the temperature value from the arguments passed to the fragment and logs it.
- If the images list of the PhotoAdapter object is empty, a Toast message is displayed to inform the user to upload some images first.
- If the images list is not empty, a new PhotoAdapter object named recommendedAdapter is created.
- Based on the temperature value, different clothing types are added to the recommended Adapter using the getRandomcloth method.
- If the temperature is less than 10, items such as gloves, scarf, boots, coat, beanie, and jeans are added.
- If the temperature is between 10 and 20, items such as sweater, beanie, shoes, jacket, and jeans are added.
- If the temperature is between 20 and 30, items such as t-shirt, shoes, and jeans are added.
- If the temperature is greater than 30, the recommended items depend on whether the user is a man or a woman. For men, tanktop, shoes, and shorts are added. For women, t-shirt, tanktop, shoes, and skirt are added.
- Finally, the showPopup method is called with the recommendedAdapter as a parameter.

## 4 CONCLUSIONS

Appendix A is the final implementation of the application. After uploading the images, if the show images button is pressed it shows the images that are uploaded with the ai prediction at the bottom of the picture.

After pressing the combination button, it gives you a combination according to weather, which is present at Appendix B. This is the combination for the temperature 17 degrees Celsius.

During my Android internship, I gained valuable experience in developing a weather application that utilizes artificial intelligence to provide recommendations based on Weather of the current location user is in. Throughout the internship, I was responsible for designing and implementing various features, such as data retrieval from weather APIs and developing machine learning algorithms to analyze user data and provide recommendations. Additionally, I learned how to integrate different libraries and APIs, such as Retrofit and TensorFlow, into my project to streamline development and improve performance. Overall, the internship provided me with a comprehensive understanding of the Android development process, as well as experience in implementing AI-based features in real-world applications.

## 5 REFERENCES

- [1] Google Developers. (n.d.). Android Developers. Retrieved from <https://developer.android.com/>
- [2] OpenWeatherMap. (n.d.). Weather API. Retrieved from <https://www.tensorflow.org/mobile/>
- [3] Retrofit. (n.d.). A type-safe HTTP client for Android and Java. Retrieved from <https://square.github.io/retrofit/>

## 6 APPENDIX

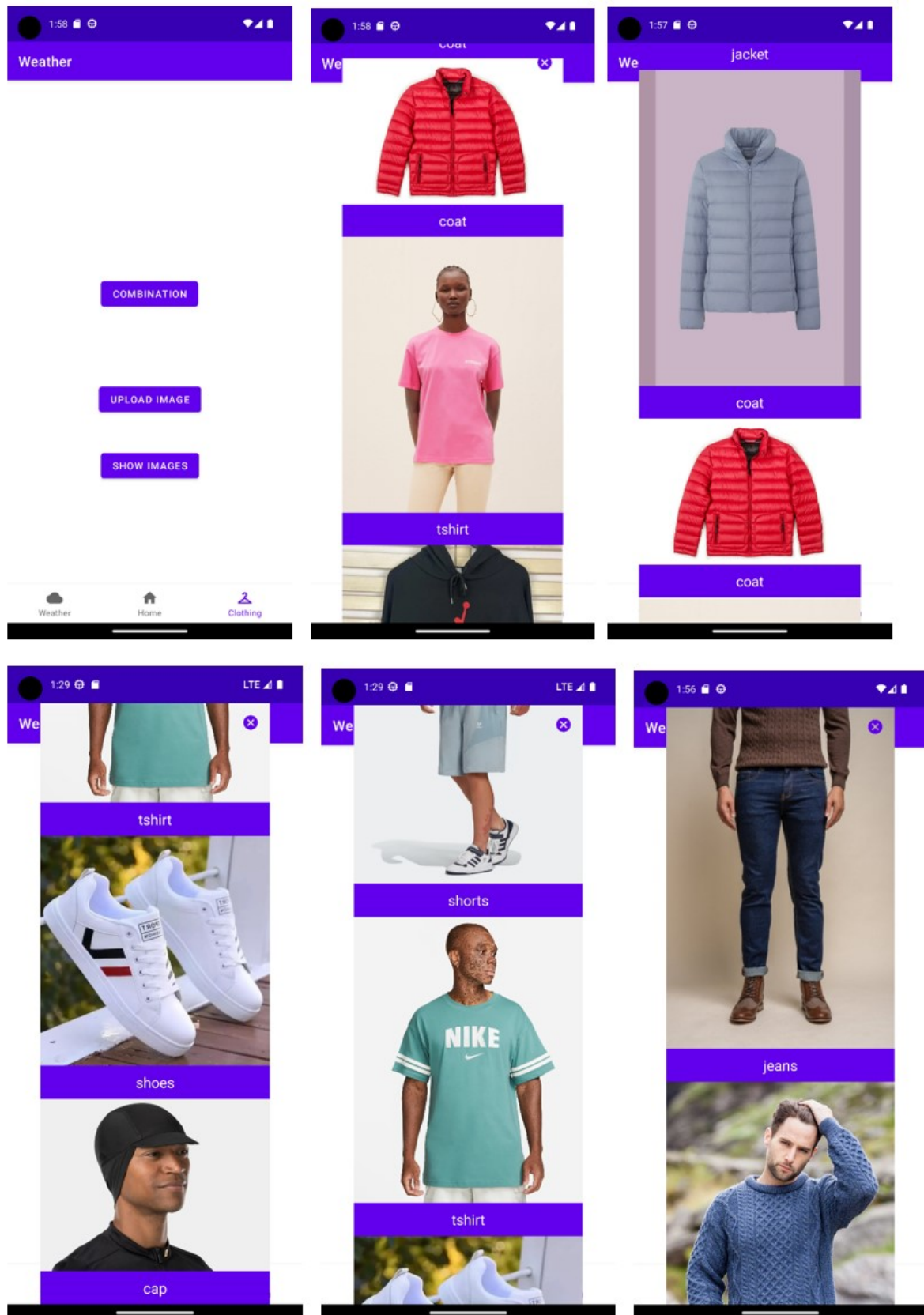


Figure 5: Appendix A: Images that AI recognized

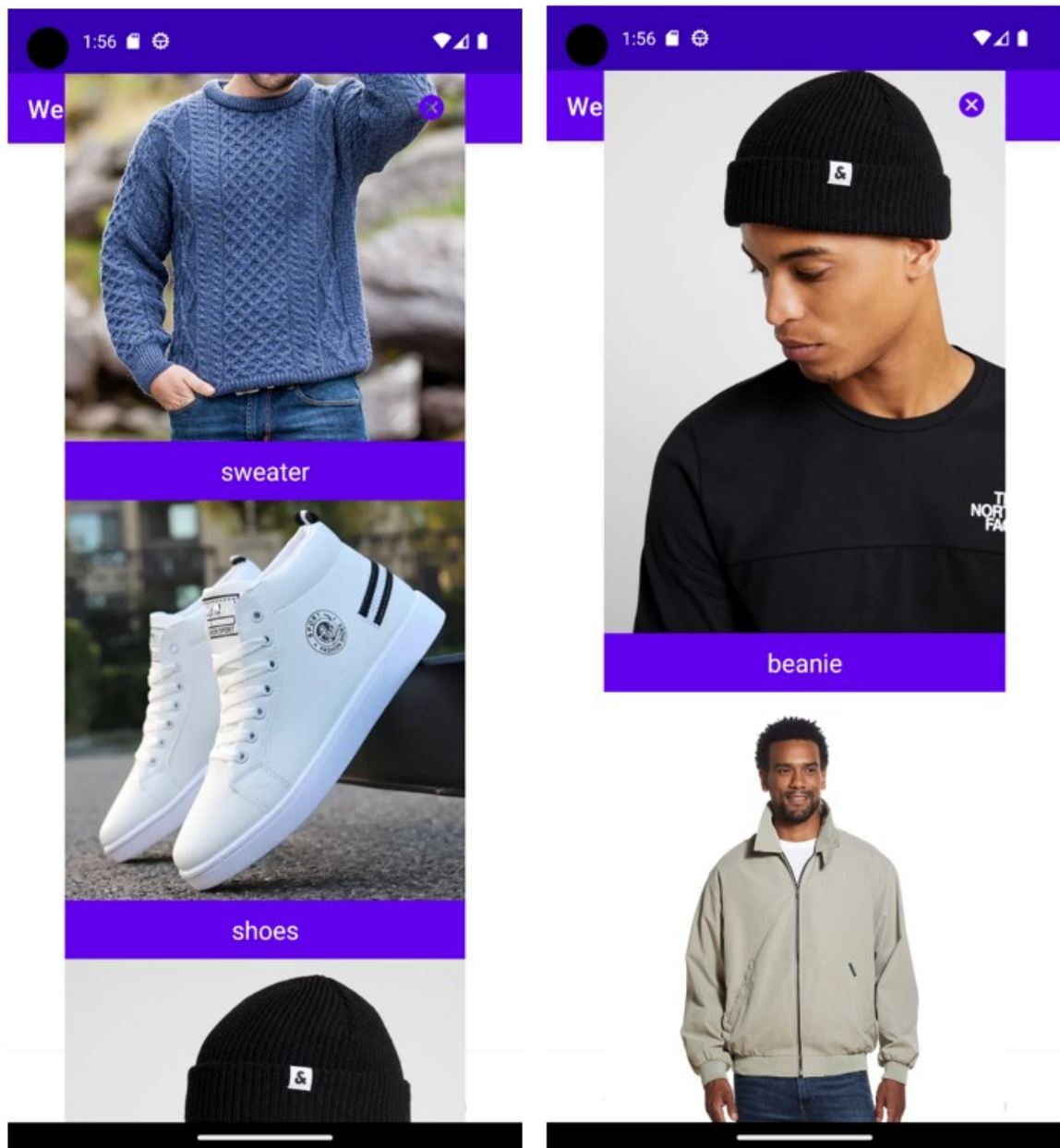


Figure 6: Appendix B: the combination AI recommended