Anusha Nelluri – anhx2@umsystem.edu

GitHub Link - https://github.com/NelluriAnusha/Demo_Remote/tree/main/Mobilepart/

Achyuth Kumar Valeti – avgh3@umsystem.edu

GitHub Link - https://github.com/AchyuthValeti/Demo_Remote/tree/main/Mobilepart/

Video Link: https://youtu.be/S-b_AYsc5G0

Presentation GitHub Link:
https://github.com/NelluriAnusha/Demo_Remote/blob/main/Mobilepart/ICP_Presentation_Two/ICP%20Presntation%20Two%20.pptx

ICP Presentation Two

A. Introduction:

Android is most widely used and popular mobile operating system in recent years. From the first quarter of 2011, Android has beaten the Apple iOS as the most popular mobile operating system worldwide. We have installed JDK & Android Studio and Java & XML technologies were used for development. For testing purposes, Android emulators or real Android devices were recommended. If it is a real Android device, we need to make sure the USB drivers are installed. Every Android app must have an AndroidManifest.xml file that describes its features. Once the new Android app project creation is done, each app module contains the following folders:

- manifests: "AndroidManifest.xml" file is located here.
- Java: All "Java source code files" are included in this folder.
- res: It has non-code resources, such as XML layouts, UI strings, and bitmap images.

B. Features:

1. Created User Login page in ICP8:

a. We created a new project in Android studio application and selected "empty activity". In 'activity_main.xml' file, implemented code for username, password textboxes and Login button which will be displayed on the mobile application. Here, we have taken password field as "textPassword" to hide the password.
b. For User login page, TextView, EditText and Button options were used to create an user interface.

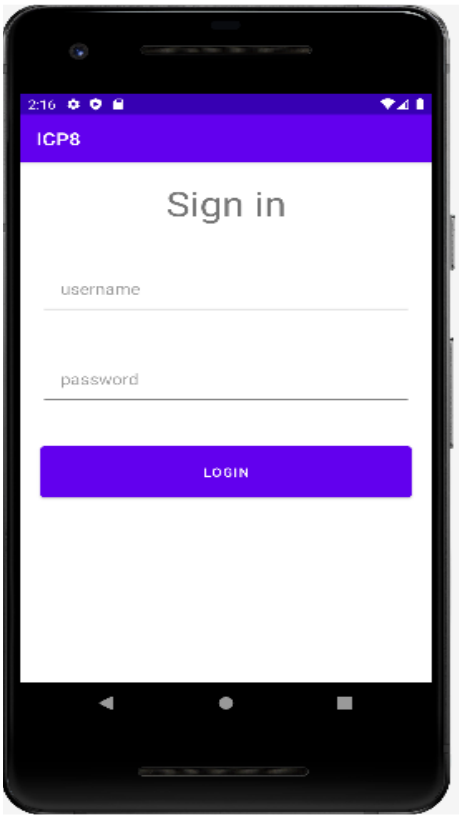
TextView: This control is used to show the text to user.It has different attributes, we used id, hint, padding, layout_width, layout_height, layout_margin etc…

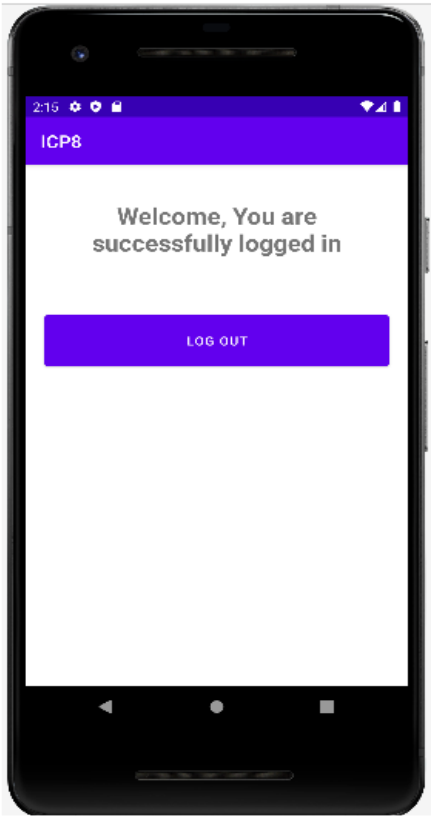
EditText: EditText is a TextView subclass that contains advanced editing features.We used id, inputType, hint, padding, layout_width, layout_height, layout_margin etc…

Button: Whenever user wants to perform an action, this button need to be clicked.

c. We have defined username and password variables with static values and implemented validations as well.
d. When valid credentials are entered and clicked on login button, it will be redirected to the next screen (welcome screen). And, we have implemented the same functionality in “MainActivity.java” file using “Intent” functionality.

Output:



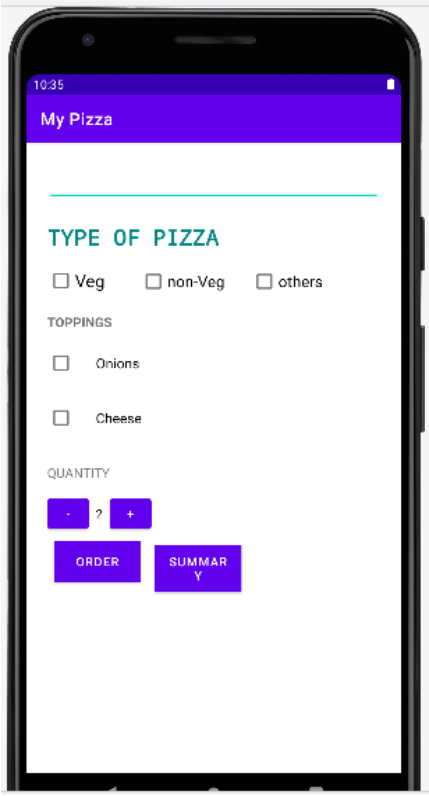


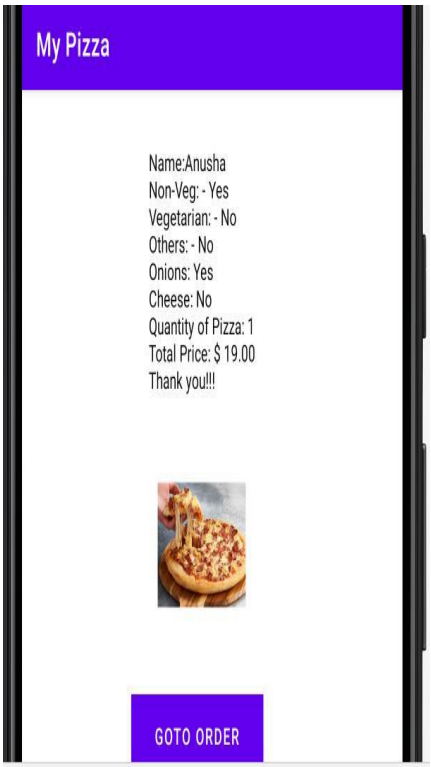
2. Developing Pizza Ordering mobile application in ICP9:

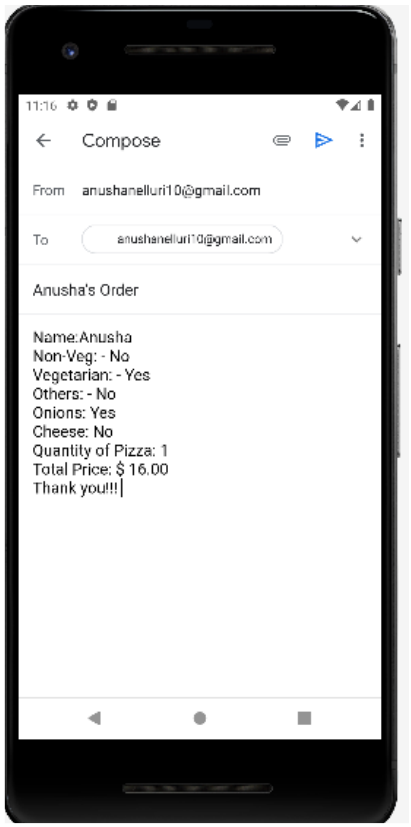
a. For developing this application, we used LinearLayout. It is a view group that aligns all its children vertically or horizontally.
b. Used “android: orientation” attribute, it specifies "horizontal" for a row and "vertical" for a column to specify the layout orientation. The default is “horizontal”.

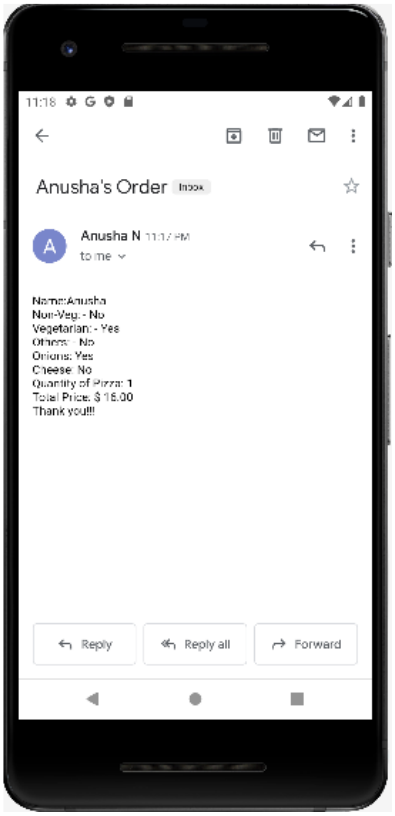
c. We used EditText, TextView, checkbox and buttons as well for developing all the options in the application.
d. Intent functionality is used to navigate between the activities. Intents are generally used for switching between different activities inside the same application, but they are not restricted to the same application; they may also be used to go from one application to another. Intent works as both implicit and explicit.
e. We have written the functionality in sendEmail() function to get the subject and order confirmation details and then sent the mail to recipient.
f. We have implemented Goto order button. When we click on this button it will redirect to the order page and the same is implemented in Java file.
g. All the entered details by user can be seen on Summary page. So, we can see the toppings added, type of page, customer name, quantity of pizza and total price of the pizza as well.

Output:





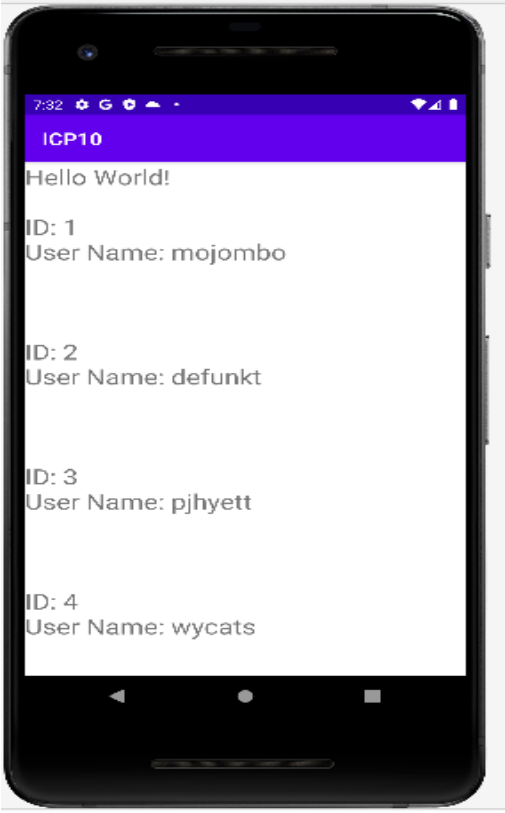




3. Fetching information from a remote server using RESTful API's in ICP10:

a. An API allows users to get information from a remote service quickly and easily. RESTful API uses HTTP requests to GET, PUT, POST and DELETE data. Through HTTP, a browser sends a request to a server and receives a response.
b. Data can be retrieved using JSON and exact names should be used when retrieving the data from server. By using @SerializedName annotation, any alternative name can be used instead of using the original name.
c. To get the access from internet, permissions need to be added in "AndroidManifest.xml" file.
d. Used Retrofit functionality to create a type-safe HTTP client for Android and Java. Retrofit is the class that converts API interfaces into callable objects(Java Interface).
e. To get the retrofit configurations in the created project, required dependencies need to be added in 'build.gradle' file.
f. Used Recycling and List functionalities as we were calling data from an array.
g. Used "androidx.core.widget.NestedScrollView" to scroll the page Up/Down.

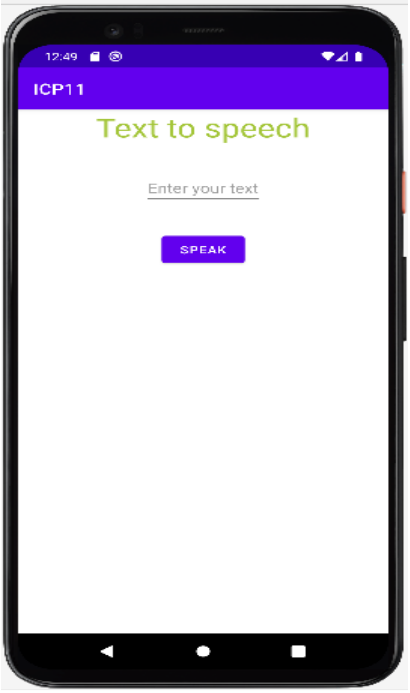
Output:

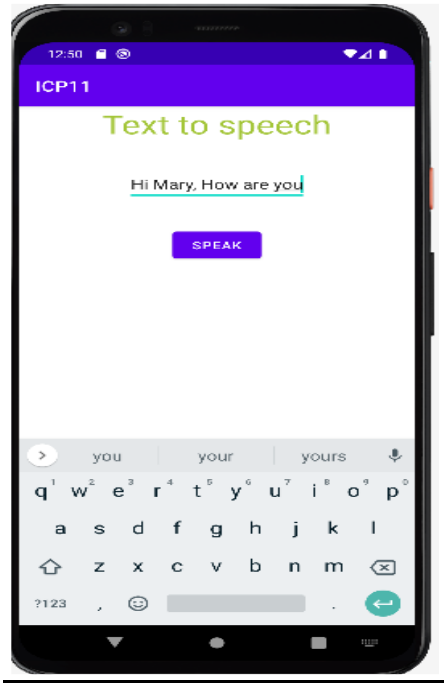


4. Text-to-Speech:

a. Text-to-speech is a popular accessibility function that assists those who have difficulty reading on-screen text, but it's also useful for those who wish to be read to. This function has proven to be a popular and beneficial feature among users.
b. It converts text on the screen into speech.
c. Any language can be chosen, and it will be converted according to that choice.
d. Used stop() function, it helped us to interrupt the current sentence conversion. It will also discard the rest of the queue's sentences.
e. Used shutdown)() function, it helped us to release the Text-to-Speech native resources.

Output:





E. Contribution:

We have contributed equally.

F. Conclusion:

In this Mobile ICP's, we have learned basics of Android and all its features and developed a mobile android application using the same.

G. Challenges:

We have not faced any major challenges while doing the implementation of Mobile applications.