Abstract

Application Description:

Over 7.7 billion people living their lives and are so busy in completing the entire list of daily tasks displayed in their to-do list, you almost have no -time to finish everything in your to-do list. Not to mention but would not be wonderful that your most of personal to-do list things can done through one app? For this reason, we have developed On-Demand Delivery app for people who have important things to do than wasting their time in shopping and at standing checkout lane for long time. With HeyAlizha, now you can order anything you want; from any store you like and we will get it deliver it to your door faster than it takes you to decided what to order next.

Hey Alizah is the best On-Demand Delivery app that you will ever need. We can deliver anything from food, groceries, alcohol, flowers for your valentine day and pet food supplies. Our Assistant can shop local from any store in your city and have your order delivered right at your door step. Now you can get power to enjoy your free time in finishing your important to-do list than worrying about personal shopping things,

SKIP THE LINE, LET US DELIVER IT.

Design:

HeyAlizah is a on-demand delivery app for fast, reliable local shopping in minutes does not matter if its day or night. With HeyAlizah, now you can just tap to request an Assistant and they will be at your doorstep in few minutes. You can let them know what you want to shop locally and your assigned Assistant can shop local from any store and have your order delivered right at your door step. Your assigned Assistant can deliver anything from: food, alcohol, groceries or anything you name it. Now you can get power of local delivery at the tap of a button.

Technical Design:

Entire development of HeyAlizah app was carried out using Android studio. HeyAlizah app development included consist of basic toolset such as SDK, IDEs, Kotlin programming language, Google and GeoFire libraries and Firebase plugins for backend setup.

**Backend:** In order to deploy HeyAlizah app, we were required to use real-time database to store Customer and Assistant Login information, customer and Assistant current real-time location and payment options. After lot of research, I have team decided to use Firebase as cloud backend for this application

Firebase is a BaaS (Backend as a service) which provides real-time database services for mobile applications. Firebase provides an API for mobile application developers and let you synchronize data and store it in the Firebase cloud and other main reason I choose firebase because it provides the simplified login process through different networks which is backbone of my application.

**API:** During the Development of HeyAlizah app I have used following API

1. Google API: In order to get real-time location for both Customers and Assistant, I have used Google API to connect real-time Google Map with location of available Assistant and customers.
2. Firebase API: Firebase API was used in order to store customer and Assistant information such as their name, address, current location and email/password in real-time cloud database

**Library:** In development of HeyAlizah application I have used following libraries in order to perform goal tasked

1. GeoFire Library: Geofire is an open-source JavaScript library that allows developers to store and query a set of items based on their geographic location. GeoFire supports Firebase for data storage, allowing query result to be updated in real-time database. Geofire helped HeyAlizah application to measure the distance between Customer and Assistant and also, I used Geofire for getting nearby Assistant to customer for fastest and reliable service
2. Google Map Library: In order to load Google Map in the application, I have used Google Map JavaScript library. Google Map library helped me to draw a graphical interface marker for customer and Assistant on the map for their real-time updated location. It also helped me to calculate distance between customer and Assistant, So customer knows the current location of their assigned Assistant who are on their way to help customers with their chores.
3. FirebaseUI Library: I have used FirebaseUI library for my application for Sign-in/Sign-up methods such as email address and password

**Kotlin Programming Language:**

Kotlin Programming language was used to develop this application since Kotlin code is inherently safer than Java code because it prevents common programming mistake in design which result in fewer system failures and application crashes. Google has officially supported and recommended Kotlin language for android apps development. Kotlin also helps to reduce errors and bugs in the application.

This app is consist of 4 main Kotlin files along with 4 XML files for Design

1. MainActivity.kt

MainActivity.kt is first file which gets executed when users run the application along with activity\_main.xml UI design. This activity has two buttons asking user if he is customer or Assistant, based on selection it moves next intent.

1. CustomerLoginActivity.kt

If user selected Customer Button in MainActivity.kt. it moves to CustomerloginActivity.kt intent which shows activity\_customer\_login.xml UI design. This file takes the email address and password of the user as textview field and provide two buttons to user to ask if they want to Login or Register their self as user in an app.

* + 1. Login Button: This button calls login() function which runs firebase query to check if user exists. If user does not exist, it through a Toast saying “Invalid User or password”. If user exist in real-time database, it moves to CustomerMapActivity.kt intent and through Toast “Login Successfully”
    2. Registration Button: This button calls register() function which runs firebase query to check if user already exist or not. If user does not exist, it uses firebase query to signup the user in real-time database. If user already exist in database, it through Toast saying “This account is already registered”

1. AssistantLoginActivity.kt

If user selected Assistant Button in MainActivity.kt. it moves to AssistantloginActivity.kt intent which shows activity\_Assistant\_login.xml UI design. This file takes the email address and password of the user as textview field and provide two buttons to user to ask if they want to Login or Register their self as user in an app.

* + 1. Login Button: This button calls login() function which runs firebase query to check if user exists. If user does not exist, it through a Toast saying “Invalid User or password”. If user exist in real-time database, it moves to AssistantMapActivity.kt intent and through Toast “Login Successfully”
    2. Registration Button: This button calls register() function which runs firebase query to check if user already exist or not. If user does not exist, it uses firebase query to signup the user in real-time database. If user already exist in database, it through Toast saying “This account is already registered”

1. AssistantMapActivity.kt

This file uses Google map activity along with activity\_Assistant\_map.xml UI design interface for user. This file executes the following functions along with google default activity functions to retrieve the user’s current location and get real-time location update.

* + 1. onCreate()
       1. This function is called by OS when activity is first created and does initialize the activity\_Assistant\_map UI elements. This function also calls the getLastLocation(), onMapReady(), getAssignedCustomer() functions
    2. getLastLocation()
       1. This function calls checkPermission() and isLocationEnable() functions to check if user have granted the permission to access the current location. Once access has provided, it takes user latitude and longitude and show marker at user current location. This function also saves the current Assistant User ID into a table called “AvailableAssistant” using FirebaseReference object into database to show all available assistant working. Assistant User ID gets remove from “AvailableAssistant” table once user logout or close the application
    3. checkPermission()
       1. This function takes Boolean variable to check if user has granted the permission to access the location or not. If user did not granted, the functions calls requestPermission() function
    4. requestPermission()
       1. This function prompt user to give appropriate permission to app, if they have been not already granted
    5. onRequestPermissionResult()
       1. This function checks if user have granted the permission to app to use their current location or not
    6. onMapReady()
       1. This function is called by OS and create Google Map interface in current activity
    7. IsLocationEnable()
       1. This function check if the user has turned on location from settings, because in rare case user may grant the app to use user location but if the location setting is off then it will not show correct location of user
    8. getAssignedCustomer()
       1. This function will call when customer requested the Assistant. This function uses FirebaseReference object to pull the ID of customer from Firebase real-time database and store it at user->Assistant->uid->customerRideId in real-time database and put this Assistant ID in “Working Assistant” table. So that this Assistant should not be available to any other customer until he is done with current assigned customer. This function also executes getAssignedCustomerPickupLocation() function
    9. getAssignedCustomerPickupLocation()
       1. This function provides the latitude and longitude and place marker in customer location to show where Assistant needs to go to provide service.

1. CustomerMapActivity.kt

This file uses Google map activity along with activity\_customer\_map.xml UI design interface for user. This file executes the following functions along with google default activity functions to retrieve the user’s current location and get real-time location update.

* + 1. onCreate()
       1. This function is called by OS when activity is first created and does initialize the activity\_customer\_map UI elements. This function also calls the getLastLocation() function
    2. getLastLocation()
       1. This function call checkPermission() and isLocationEnable() functions to check if user have granted the permission to access the current location. Once access has provided, it takes user latitude and longitude and drop marker at user current location
    3. checkPermission()
       1. This function takes Boolean variable to check if user has granted the permission to access the location or not. If user did not granted, the functions calls requestPermission() function
    4. requestPermission()
       1. This function prompt user to give appropriate permission to app, if they have are not already granted
    5. onRequestPermissionResult()
       1. This function checks if user have granted the permission to app to use their current location or not
    6. onMapReady()
       1. This function is called by OS and create Google Map interface in current activity
    7. IsLocationEnable()
       1. This function check if the user has turned on location from settings, because in rare case user may grant the app to use user location but if the location setting is off then it will not show correct location of user
    8. getCloserAssistant()
       1. This function is recursive function that calls itself. This function is use to find close Assistant for customer to provide fast service. This function creates Geo-fence around the customer location and look for Assistant within that Geo-fence, if it does not find within same Geo-fence. The function will keep increasing the radius of Feo-fence and call same function again. This procedure will be repeated until closest Assistant is found. Once closest Assistant is found, it will change Request Assistant button text to “Assistant Found” and will calls getAssistantLocation() function.
    9. getAssistantLocation
       1. This function provides the latitude and longitude and place marker in assistant location to show customer how far their assigned Assistant is. So customer have real-time location of their assigned Assistant

**Application Using Instructions:**

This instruction can be followed to use HeyAlizah On-Demand Delivery app

**Select User Type:**

In main app screen, User need to select appropriate user type either they are customer or assistant

**Create an account**

All you need is an email address and password. It asks user to type their email address password. After typing required field, the user must click on Registration button. It will create user profile in HeyAlizah database and will prompt user that they are been successfully registered.

**Login into account:**

User need to type their registered email address and password in order to login in the application. Once the user is successfully logged in, the application will ask user to give access to their current location. Once permission Is granted, the application will show google map and place blue marker at user current location

**Customer account:**

Once customer is successfully logged in the application. Customer can see their location and can click on Request Assistant button for requesting the assistant. The application will then match the nearby assistant to you and notify assistant to arrive at your location. Customer can track their arrival and location of assistant on the map

**Assistant account:**

Once assistant is successfully logged in the application. Assistant will automatically begin to receive request in their area. Your phone will notify you, if customer has requested assistant. Assistant can log out if they are not ready to provide service at moment. Once you have request, the app makes it easy to find your customer by putting marker on their location and you can navigate to their location.

**Limitations:**

following features of application were not implemented as mentioned in proposal due to lack of time