**Experiment No: 8**

**Theory:**

ChatAway is a chatting app which allows user to share images and text messages with each other. Also, users of the app are notified through a central notification service which uses cloud messaging.

**Code:**

**PeopleFragment.kt**

package com.example.divyaank.chataway.fragment

import android.os.Bundle

import android.support.v4.app.Fragment

import android.support.v7.widget.LinearLayoutManager

import android.view.LayoutInflater

import android.view.OrientationEventListener

import android.view.View

import android.view.ViewGroup

import com.example.divyaank.chataway.AppConstants

import com.example.divyaank.chataway.ChatActivity

import com.example.divyaank.chataway.R

import com.example.divyaank.chataway.recyclerview.item.PersonItem

import com.example.divyaank.chataway.util.FirestoreUtil

import com.google.firebase.firestore.ListenerRegistration

import com.xwray.groupie.\*

import kotlinx.android.synthetic.main.fragment\_people.\*

import org.jetbrains.anko.support.v4.startActivity

// TODO: Rename parameter arguments, choose names that match

// the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER

private const val ARG\_PARAM1 = "param1"

private const val ARG\_PARAM2 = "param2"

/\*\*

\* A simple [Fragment] subclass.

\*

\*/

class PeopleFragment : Fragment() {

private lateinit var userListenerRegistration: ListenerRegistration

private var shouldinitRecyclerView = true

private lateinit var peopleSection: Section

override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,

savedInstanceState: Bundle?): View? {

userListenerRegistration = FirestoreUtil.addUsersListener(this.activity!!, this::updateRecyclerView)

return inflater.inflate(R.layout.fragment\_people, container, false)

}

override fun onDestroyView() {

super.onDestroyView()

FirestoreUtil.removeListener(userListenerRegistration)

shouldinitRecyclerView = true

}

private fun updateRecyclerView(items: List<com.xwray.groupie.kotlinandroidextensions.Item>){

fun init(){

recycler\_view\_people.apply {

layoutManager = LinearLayoutManager(this@PeopleFragment.context)

adapter = GroupAdapter<ViewHolder>().apply {

peopleSection = Section(items)

add(peopleSection)

setOnItemClickListener(onItemClick)

}

}

shouldinitRecyclerView = false

}

fun updateItems() = peopleSection.update(items)

if(shouldinitRecyclerView)

init()

else

updateItems()

}

private val onItemClick = OnItemClickListener{item, view ->

if (item is PersonItem){

startActivity<ChatActivity>(

AppConstants.USER\_NAME to item.person.name,

AppConstants.USER\_ID to item.userId

)

}

}

}

**SignInActivity.kt**

package com.example.divyaank.chataway

import android.app.Activity

import android.content.Intent

import android.support.v7.app.AppCompatActivity

import android.os.Bundle

import android.support.constraint.ConstraintLayout

import android.util.Log

import com.example.divyaank.chataway.service.MyFirebaseInstanceIDService

import com.example.divyaank.chataway.util.FirestoreUtil

import com.firebase.ui.auth.AuthUI

import com.firebase.ui.auth.ErrorCodes

import com.firebase.ui.auth.IdpResponse

import com.google.firebase.iid.FirebaseInstanceId

import kotlinx.android.synthetic.main.activity\_main.\*

import kotlinx.android.synthetic.main.activity\_sign\_in.\*

import org.jetbrains.anko.\*

import org.jetbrains.anko.design.longSnackbar

import com.google.firebase.iid.InstanceIdResult

import com.google.android.gms.tasks.OnSuccessListener

class SignInActivity : AppCompatActivity() {

private val RC\_SIGN\_IN = 1

private val signInProviders = listOf(AuthUI.IdpConfig.EmailBuilder().setAllowNewAccounts(true).setRequireName(true).build())

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_sign\_in)

account\_sign\_in.setOnClickListener{

val intent = AuthUI.getInstance().createSignInIntentBuilder()

.setAvailableProviders(signInProviders).setLogo(R.drawable.ic\_fire\_emoji).build()

startActivityForResult(intent, RC\_SIGN\_IN)

}

}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {

super.onActivityResult(requestCode, resultCode, data)

if(requestCode == RC\_SIGN\_IN){

val response = IdpResponse.fromResultIntent(data)

if(resultCode == Activity.RESULT\_OK){

val progressDialog = indeterminateProgressDialog("SETTING UP YOUR ACCOUNT")

FirestoreUtil.initCurrentUserIfFirstTime {

startActivity(intentFor<MainActivity>().newTask().clearTask()) //ENABLES USER TO REVISIT THE ACCOUNT AFTER SIGNING IN

FirebaseInstanceId.getInstance().instanceId.addOnSuccessListener(this@SignInActivity) { instanceIdResult ->

val registrationToken = instanceIdResult.token

MyFirebaseInstanceIDService.addTokenToFirestore(registrationToken)

Log.e("Token", registrationToken)

}

progressDialog.dismiss()

}

}

else if(resultCode == Activity.RESULT\_CANCELED){

if(response == null) return

when(response.error?.errorCode){

ErrorCodes.NO\_NETWORK ->

container.longSnackbar("NO NEWTWORK :(")

ErrorCodes.UNKNOWN\_ERROR ->

container.longSnackbar("UNKNOWN ERROR")

}

}

}

}

}

**MyAccountFragment.kt**

package com.example.divyaank.chataway.fragment

import android.app.Activity

import android.content.Intent

import android.graphics.Bitmap

import android.os.Bundle

import android.provider.MediaStore

import android.support.v4.app.Fragment

import android.view.LayoutInflater

import android.view.View

import android.view.ViewGroup

import com.example.divyaank.chataway.R

import com.example.divyaank.chataway.SignInActivity

import com.example.divyaank.chataway.glide.GlideApp

import com.example.divyaank.chataway.util.FirestoreUtil

import com.example.divyaank.chataway.util.StorageUtil

import com.firebase.ui.auth.AuthUI

import kotlinx.android.synthetic.main.fragment\_my\_account.\*

import kotlinx.android.synthetic.main.fragment\_my\_account.view.\*

import org.jetbrains.anko.clearTask

import org.jetbrains.anko.newTask

import org.jetbrains.anko.support.v4.intentFor

import org.jetbrains.anko.support.v4.startActivity

import org.jetbrains.anko.support.v4.startActivityForResult

import org.jetbrains.anko.support.v4.toast

import java.io.ByteArrayOutputStream

import java.io.OutputStream

// TODO: Rename parameter arguments, choose names that match

// the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER

private const val ARG\_PARAM1 = "param1"

private const val ARG\_PARAM2 = "param2"

/\*\*

\* A simple [Fragment] subclass.

\*

\*/

class MyAccountFragment : Fragment() {

private val RC\_SELECT\_IMAGE = 2

private lateinit var selectedImageBytes: ByteArray

private var pictureJustChanged = false

override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?,

savedInstanceState: Bundle?): View? {

// Inflate the layout for this fragment

val view = inflater.inflate(R.layout.fragment\_my\_account, container, false)

view.apply {

imageView\_profile\_picture.setOnClickListener {

val intent = Intent().apply {

type = "image/\*"

action = Intent.ACTION\_GET\_CONTENT

putExtra(Intent.EXTRA\_MIME\_TYPES, arrayOf("image/jpeg","image/png"))

}

startActivityForResult(Intent.createChooser(intent, "SELECT IMAGE"),RC\_SELECT\_IMAGE)

}

btn\_save.setOnClickListener {

if(::selectedImageBytes.isInitialized)

StorageUtil.uploadProfilePhoto(selectedImageBytes){

imagePath -> FirestoreUtil.updateCurrentUser(editText\_name.text.toString(),

editText\_bio.text.toString(),imagePath)

}

else

FirestoreUtil.updateCurrentUser(editText\_name.text.toString(),

editText\_bio.text.toString(),null)

toast("SAVING")

}

btn\_sign\_out.setOnClickListener {

AuthUI.getInstance().signOut(this@MyAccountFragment.context!!)

.addOnCompleteListener {

startActivity(intentFor<SignInActivity>().newTask().clearTask())

}

}

}

return view

}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {

if(requestCode == RC\_SELECT\_IMAGE && resultCode == Activity.RESULT\_OK && data != null && data.data != null){

val selectedImagePath = data.data

val selectedImageBmp = MediaStore.Images.Media.getBitmap(activity?.contentResolver, selectedImagePath)

val outputStream = ByteArrayOutputStream()

selectedImageBmp.compress(Bitmap.CompressFormat.JPEG, 90, outputStream)

selectedImageBytes = outputStream.toByteArray()

GlideApp.with(this)

.load(selectedImageBytes)

.into(imageView\_profile\_picture)

pictureJustChanged = true

}

}

override fun onStart(){

super.onStart()

FirestoreUtil.getCurrentUser { user ->

if(this@MyAccountFragment.isVisible){

editText\_name.setText(user.name)

editText\_bio.setText(user.bio)

if(!pictureJustChanged && user.profilePicturePath!=null)

GlideApp.with(this)

.load(StorageUtil.pathToReference(user.profilePicturePath))

.placeholder(R.drawable.ic\_account\_circle\_black\_24dp)

.into(imageView\_profile\_picture)

}

}

}

}

**ChatActivity.kt**

package com.example.divyaank.chataway

import android.app.Activity

import android.content.Intent

import android.graphics.Bitmap

import android.support.v7.app.AppCompatActivity

import android.os.Bundle

import android.os.Message

import android.provider.MediaStore

import com.example.divyaank.chataway.R

import android.support.v7.widget.LinearLayoutManager

import com.example.divyaank.chataway.model.ImageMessage

import com.example.divyaank.chataway.model.MessageType

import com.example.divyaank.chataway.model.TextMessage

import com.example.divyaank.chataway.model.User

import com.example.divyaank.chataway.util.FirestoreUtil

import com.example.divyaank.chataway.util.StorageUtil

import com.google.firebase.auth.FirebaseAuth

import com.google.firebase.firestore.ListenerRegistration

import com.xwray.groupie.GroupAdapter

import com.xwray.groupie.Section

import com.xwray.groupie.ViewHolder

import com.xwray.groupie.kotlinandroidextensions.Item

import kotlinx.android.synthetic.main.activity\_chat.\*

import org.jetbrains.anko.startActivityForResult

import org.jetbrains.anko.toast

import java.io.ByteArrayOutputStream

import java.util.\*

private const val RC\_SELECT\_IMAGE = 2

class ChatActivity : AppCompatActivity() {

private lateinit var currentChannelId: String

private lateinit var currentUser: User

private lateinit var otherUserId: String

private lateinit var messagesListenerRegistration : ListenerRegistration

private var shouldInitRecyclerView = true

private lateinit var messagesSection : Section

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_chat)

supportActionBar?.setDisplayHomeAsUpEnabled(true)

supportActionBar?.title = intent.getStringExtra(AppConstants.USER\_NAME)

FirestoreUtil.getCurrentUser {

currentUser = it

}

otherUserId = intent.getStringExtra(AppConstants.USER\_ID)

FirestoreUtil.getOrCreateChatChannel(otherUserId){

channelId ->

currentChannelId = channelId

messagesListenerRegistration = FirestoreUtil

.addChatMessageListener(channelId, this, this::updateRecyclerView)

imageView\_send.setOnClickListener {

val messageToSend =

TextMessage(editText\_message.text.toString(), Calendar.getInstance().time,

FirebaseAuth.getInstance().currentUser!!.uid, otherUserId,

currentUser.name)

editText\_message.setText("")

FirestoreUtil.sendMessage(messageToSend, channelId)

}

fab\_send\_image.setOnClickListener {

val intent = Intent().apply {

type = "image/\*"

action = Intent.ACTION\_GET\_CONTENT

putExtra(Intent.EXTRA\_MIME\_TYPES, arrayOf("image/jpeg", "image/png"))

}

startActivityForResult(Intent.createChooser(intent,"Select Image"), RC\_SELECT\_IMAGE)

}

}

}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {

if (requestCode == RC\_SELECT\_IMAGE && resultCode == Activity.RESULT\_OK &&

data!=null && data.data != null){

val selectedImagePath = data.data

val selectedImageBmp = MediaStore.Images.Media.getBitmap(contentResolver, selectedImagePath)

val outputStream = ByteArrayOutputStream()

selectedImageBmp.compress(Bitmap.CompressFormat.JPEG, 90, outputStream)

val selectedImageBytes = outputStream.toByteArray()

StorageUtil.uploadMessageImage(selectedImageBytes){

imagePath -> val messageToSend = ImageMessage(imagePath, Calendar.getInstance().time,

FirebaseAuth.getInstance().currentUser!!.uid, otherUserId,

currentUser.name)

FirestoreUtil.sendMessage(messageToSend, currentChannelId)

}

}

}

private fun updateRecyclerView(message: List<Item>){

fun init(){

recycler\_view\_messages.apply {

layoutManager = LinearLayoutManager(this@ChatActivity)

adapter = GroupAdapter<ViewHolder>().apply {

messagesSection = Section(message)

this.add(messagesSection)

}

}

shouldInitRecyclerView = false

}

fun updateItems() = messagesSection.update(message)

if(shouldInitRecyclerView)

init()

else

updateItems()

recycler\_view\_messages.scrollToPosition(recycler\_view\_messages.adapter!!.itemCount - 1)

}

}

**TextMessageItem.kt**

package com.example.divyaank.chataway.recyclerview.item

import android.content.Context

import com.example.divyaank.chataway.R

import com.example.divyaank.chataway.model.TextMessage

import com.xwray.groupie.kotlinandroidextensions.ViewHolder

import kotlinx.android.synthetic.main.item\_text\_message.\*

class TextMessageItem(val message: TextMessage,

val context: Context): MessageItem(message)

{

override fun bind(viewHolder: ViewHolder, position: Int) {

viewHolder.textView\_message\_text.text = message.text

super.bind(viewHolder, position)

}

override fun getLayout() = R.layout.item\_text\_message

override fun isSameAs(other: com.xwray.groupie.Item<\*>?): Boolean {

if (other !is TextMessageItem)

return false

if (this.message != other.message)

return false

return true

}

override fun equals(other: Any?): Boolean {

return isSameAs(other as? TextMessageItem)

}

override fun hashCode(): Int {

var result = message.hashCode()

result = 31 \* result + context.hashCode()

return result

}

}

**ImageMessageItem.kt**

package com.example.divyaank.chataway.recyclerview.item

import android.content.Context

import com.example.divyaank.chataway.R

import com.example.divyaank.chataway.glide.GlideApp

import com.example.divyaank.chataway.model.ImageMessage

import com.example.divyaank.chataway.model.Message

import com.example.divyaank.chataway.util.StorageUtil

import com.xwray.groupie.kotlinandroidextensions.ViewHolder

import kotlinx.android.synthetic.main.item\_image\_message.\*

class ImageMessageItem(val message: ImageMessage,

val context: Context): MessageItem(message){

override fun bind(viewHolder: ViewHolder, position: Int) {

super.bind(viewHolder, position)

GlideApp.with(context)

.load(StorageUtil.pathToReference(message.imagePath))

.placeholder(R.drawable.ic\_image\_black\_24dp)

.into(viewHolder.imageView\_message\_image)

}

override fun getLayout() = R.layout.item\_image\_message

override fun isSameAs(other: com.xwray.groupie.Item<\*>?): Boolean {

if (other !is ImageMessageItem)

return false

if (this.message != other.message)

return false

return true

}

override fun equals(other: Any?): Boolean {

return isSameAs(other as? ImageMessageItem)

}

override fun hashCode(): Int {

var result = message.hashCode()

result = 31 \* result + context.hashCode()

return result

}

}

**MessageItem.kt**

package com.example.divyaank.chataway.recyclerview.item

import android.view.Gravity

import android.widget.FrameLayout

import com.example.divyaank.chataway.R

import com.google.firebase.auth.FirebaseAuth

import com.xwray.groupie.kotlinandroidextensions.ViewHolder

import kotlinx.android.synthetic.main.item\_text\_message.\*

import org.jetbrains.anko.backgroundResource

import org.jetbrains.anko.wrapContent

import java.text.SimpleDateFormat

import com.example.divyaank.chataway.model.Message

import com.xwray.groupie.kotlinandroidextensions.Item

abstract class MessageItem(private val message: Message): Item() {

override fun bind(viewHolder: ViewHolder, position: Int) {

setTimeText(viewHolder)

setMessageRootGravity(viewHolder)

}

private fun setTimeText(viewHolder: ViewHolder){//Time of sending the text

val dateFormat = SimpleDateFormat

.getDateTimeInstance(SimpleDateFormat.SHORT, SimpleDateFormat.SHORT)

viewHolder.textView\_message\_time.text = dateFormat.format(message.time)

}

private fun setMessageRootGravity(viewHolder: ViewHolder){ //Sets orientation of messages depending on whether sent by user or not

if(message.senderId == FirebaseAuth.getInstance().currentUser?.uid){

viewHolder.message\_root.apply {

backgroundResource = R.drawable.rect\_round\_white

val lParams = FrameLayout.LayoutParams(wrapContent, wrapContent, Gravity.END)

this.layoutParams = lParams

}

}

else{

viewHolder.message\_root.apply {

backgroundResource = R.drawable.rect\_round\_primary\_color

val lParams = FrameLayout.LayoutParams(wrapContent, wrapContent, Gravity.START)

this.layoutParams = lParams

}

}

}

}

**PersonItem.kt**

package com.example.divyaank.chataway.recyclerview.item

import android.annotation.SuppressLint

import android.content.Context

import com.example.divyaank.chataway.R

import com.example.divyaank.chataway.glide.GlideApp

import com.example.divyaank.chataway.util.StorageUtil

import com.firebase.ui.auth.data.model.User

import com.xwray.groupie.kotlinandroidextensions.Item

import com.xwray.groupie.kotlinandroidextensions.ViewHolder

import kotlinx.android.synthetic.main.item\_person.\*

// @SuppressLint("RestrictedApi")

class PersonItem(val person: com.example.divyaank.chataway.model.User,

val userId: String,

private val context: Context)

: Item() {

override fun bind(viewHolder: ViewHolder, position: Int) {

viewHolder.textView\_name.text = person.name

viewHolder.textView\_bio.text = person.bio

if (person.profilePicturePath != null) {

GlideApp.with(context)

.load(StorageUtil.pathToReference(person.profilePicturePath))

.placeholder(R.drawable.ic\_account\_circle\_black\_24dp)

.into(viewHolder.imageView\_profile\_picture)

}

}

override fun getLayout() = R.layout.item\_person

}

Screenshots:

