

# CMP309 – SOFTWARE DEVELOPMENT FOR MOBILE DEVICES

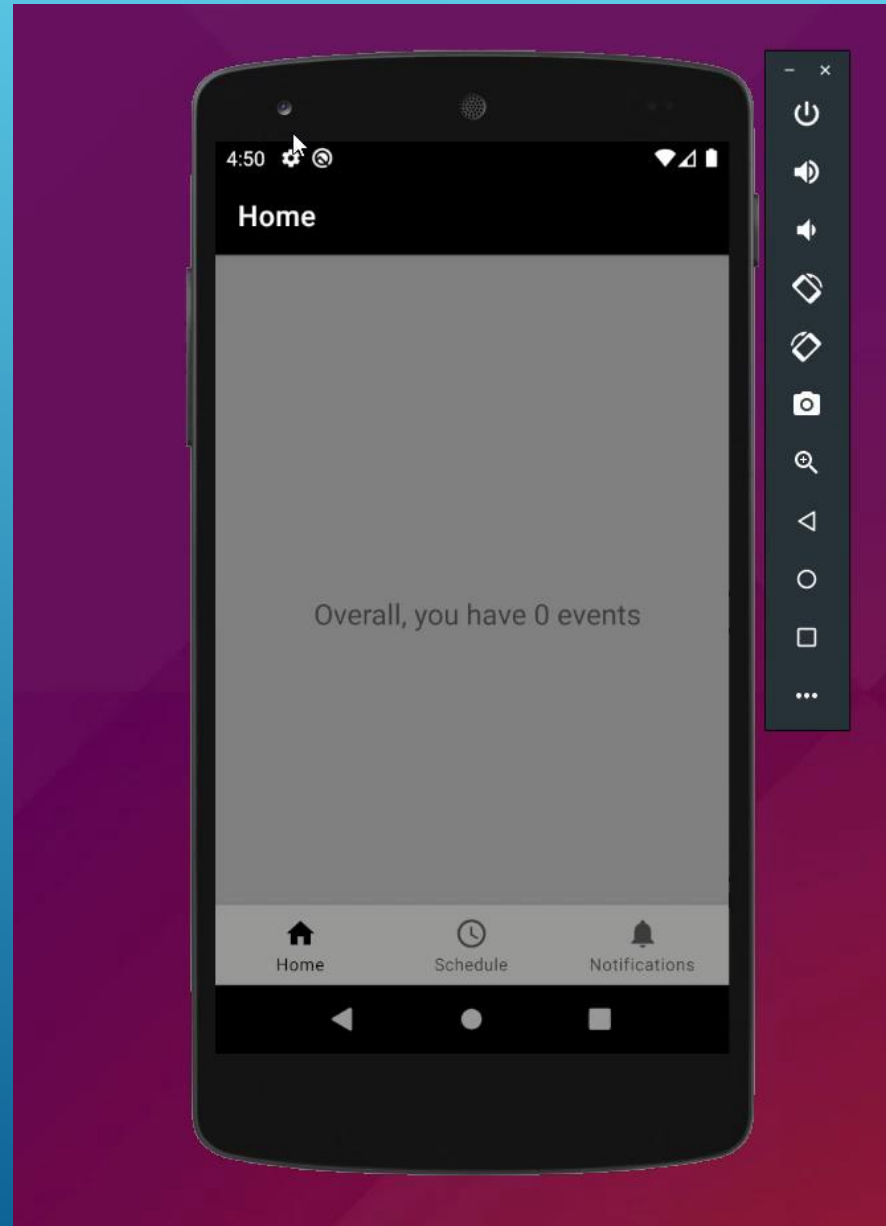
Ethan Hastie

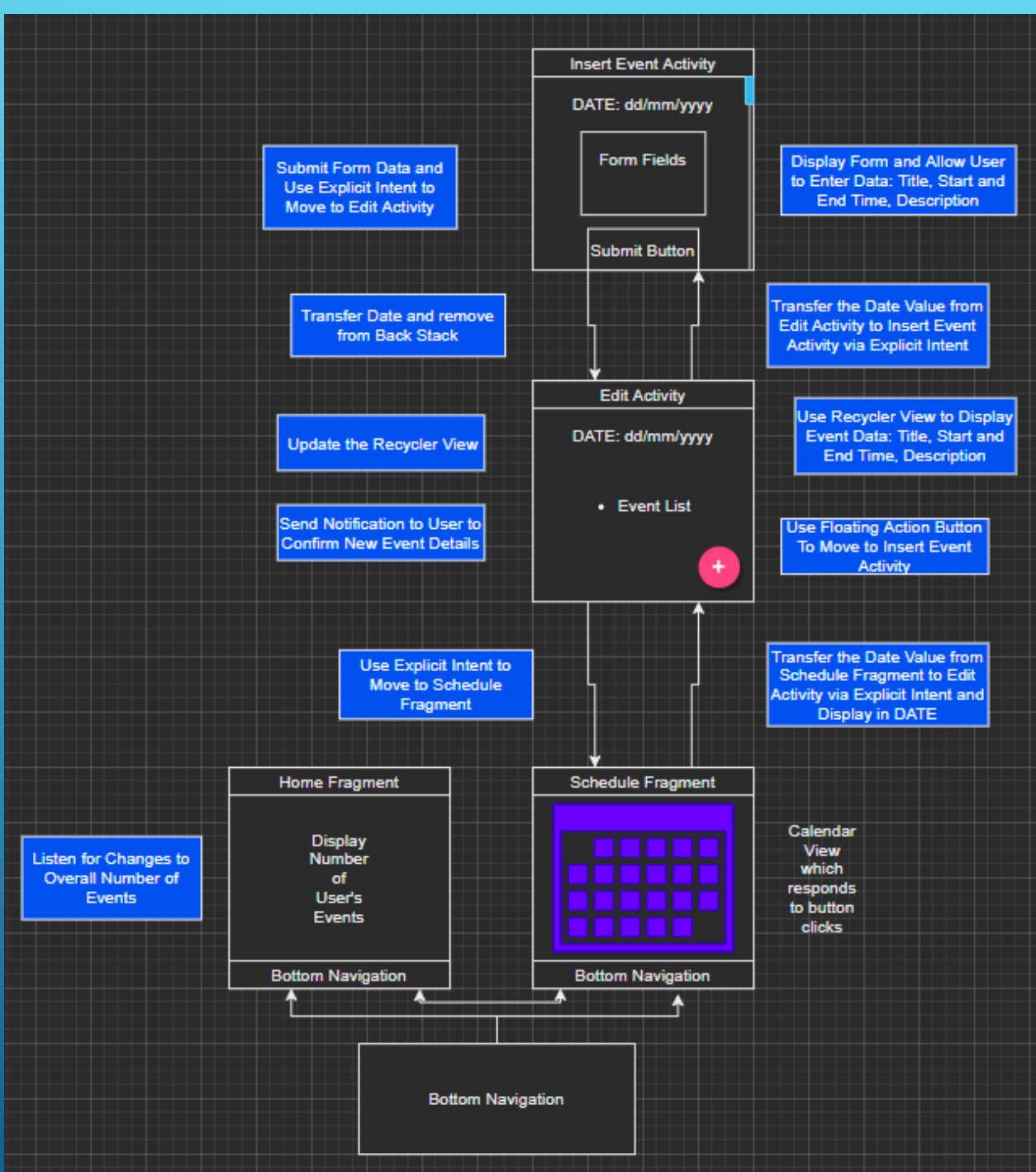
Student ID:1801853

Project Title: To-Do App



# APP DEMONSTRATION

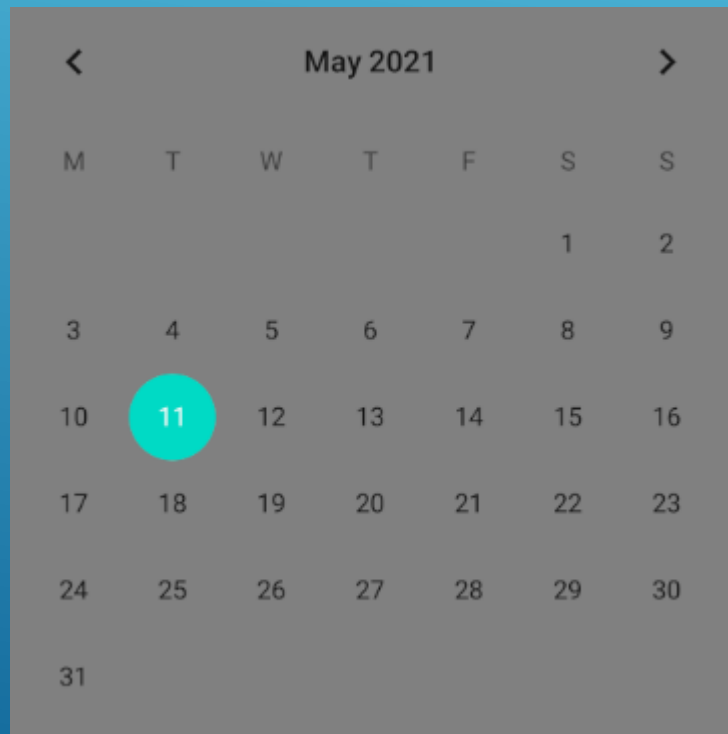




# CODE - OVERVIEW



```
// on click listener for calendar view
calendar.setOnDateChangeListener { _, year: Int, month: Int, day: Int ->
    changeValues(day, month, year)
    moveActivity(viewModel.dateValue)
}
```



Date: 11/05/2021

Title

---

00:00



00:00

Description

---

```
if (viewModel.formValidate()) {  
    val item = EventModel( ID: 0,  
                            viewModel.title,  
                            viewModel.date,  
                            viewModel.startTime,  
                            viewModel.endTime,  
                            viewModel.desc  
    )  
  
    viewModel.insertEvent(item)
```



```

}/*
 * create functions for db
 */
@Dao
interface EventDao {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insert(event: EventModel)

    @Update
    suspend fun delete(event: EventModel)

    @Query( value: "SELECT COUNT(date) FROM Events")
    fun getNumberOfEvents() : LiveData<Integer>

    @Query( value: "SELECT * FROM Events WHERE date==:date")
    fun getSpecificEvents(date: String): LiveData<List<EventModel>>
}

```

```

}/*
 * NOTE: not part of the MVC architecture but makes it easier and
 * is recommended
 */
class EventRepository(private val eventDao: EventDao) {

    suspend fun insert(item: EventModel) {
        eventDao.insert(item)
    }

    suspend fun delete(item: EventModel) =
        eventDao.delete(item)

    fun getSpecificEvents(date: String) =
        eventDao.getSpecificEvents(date)

    fun getNumberOfEvents(): Int {
        return eventDao.getNumberOfEvents()
    }
}

```



```

class NotificationHelper {

    fun sendNotification(context: Context, title: String, body: String) {
        val manager = NotificationManagerCompat.from(context)

        // check for build version and build a channel if greater than API 26
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O)
            manager.createNotificationChannel(createNotificationChannel())

        // assign values to the notification from the form submission
        val notificationBuilder = NotificationCompat.Builder(context, CHANNEL_ID)
            .setSmallIcon(R.drawable.ic_launcher_foreground)
            .setContentTitle(title)
            .setContentText(body)

        // show the notification
        manager.notify(NOTIFICATION_ID, notificationBuilder.build())
    }

    @RequiresApi(Build.VERSION_CODES.O)
    private fun createNotificationChannel(): NotificationChannel {
        return NotificationChannel(
            CHANNEL_ID, CHANNEL_NAME,
            NotificationManager.IMPORTANCE_DEFAULT
        )
    }
}

```

```

private fun showNotification() {
    val notificationHelper = NotificationHelper()
    notificationHelper.sendNotification( context: this,
        viewModel.title,
        body: "Your event will begin at ${viewModel.startTime}")
}

```





```
/**
 * bind item views with adapter
 */
@SuppressLint( ...value: "SetTextI18n")
override fun onBindViewHolder(holder: EventViewHolder, position: Int) {
    var currentPosition = list[position]
    holder.itemView.titleText.text = "Title: " + currentPosition.title
    holder.itemView.startTimeText.text = "Start: " + currentPosition.timeStart
    holder.itemView.endTimeText.text = "End: " + currentPosition.timeEnd
    holder.itemView.descText.text = "Description: " + currentPosition.description
    holder.itemView.buttonDelete.setOnClickListener { it: View!
        viewModel.delete(currentPosition)
    }
}
```



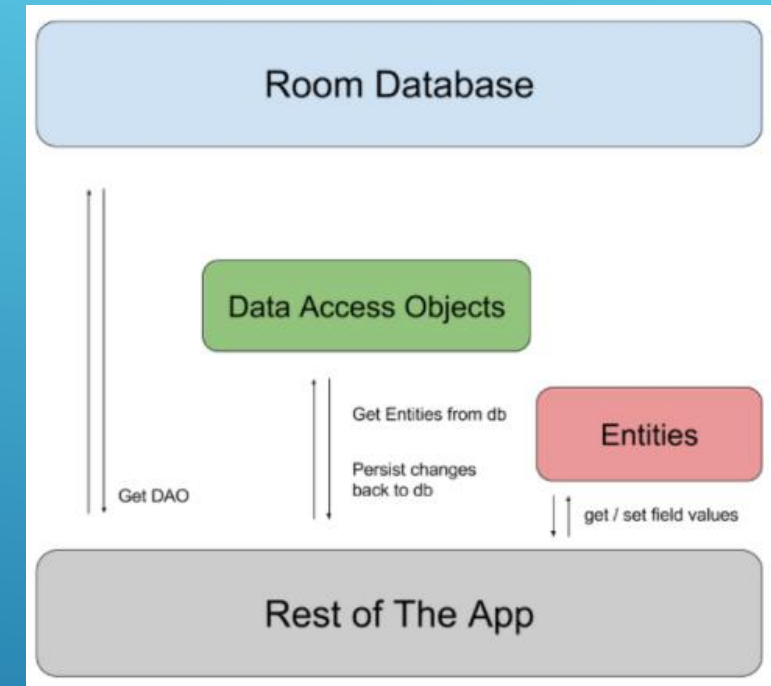
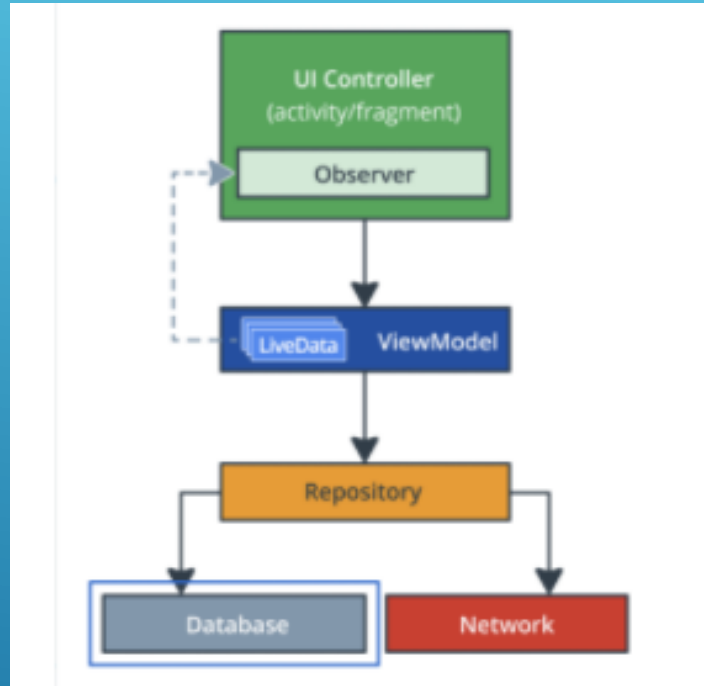
# OVERVIEW - CRITICAL ANALYSIS

- FEATURES
- USABILITY
- PERFORMANCE
- SECURITY



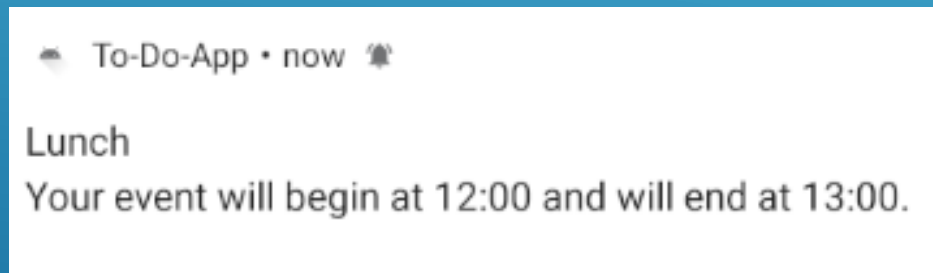
# FEATURES

- ▶ Database
  - ▶ SQLite
  - ▶ Room
  - ▶ Firebase Firestore



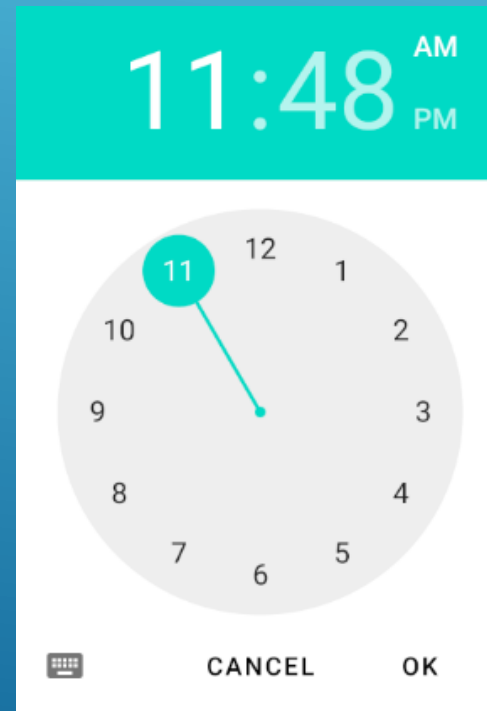
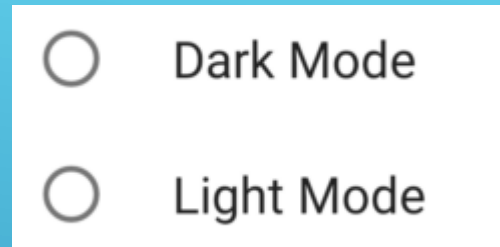
# FEATURES

- ▶ Notifications
  - ▶ When the user submits an event



# USABILITY

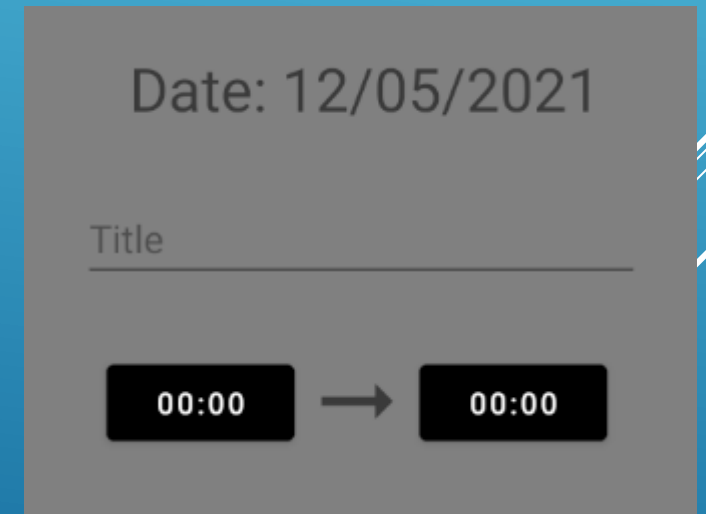
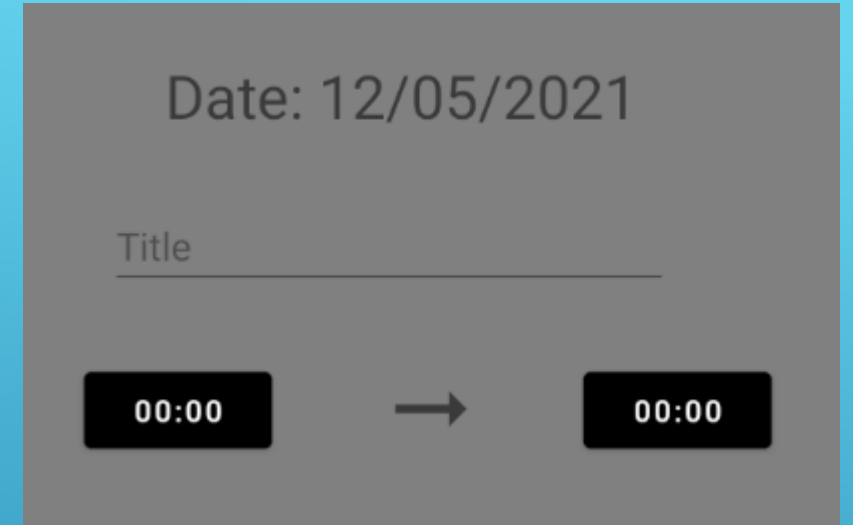
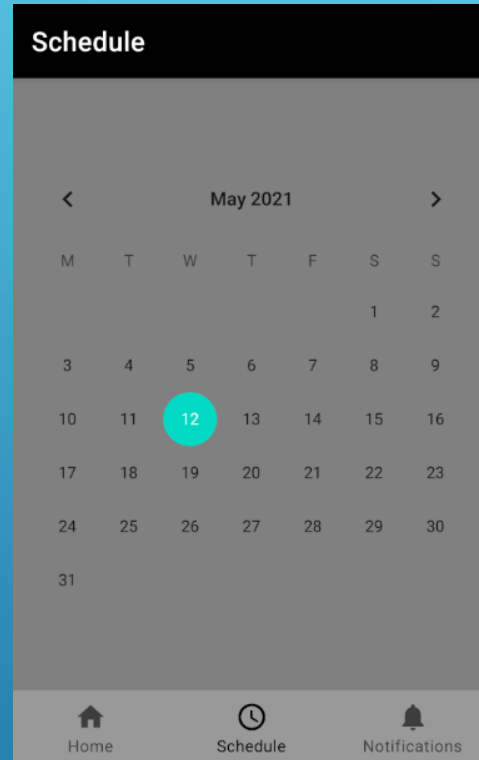
- ▶ Dark Mode/Light Mode options
- ▶ Time Picker Dialog



# USABILITY

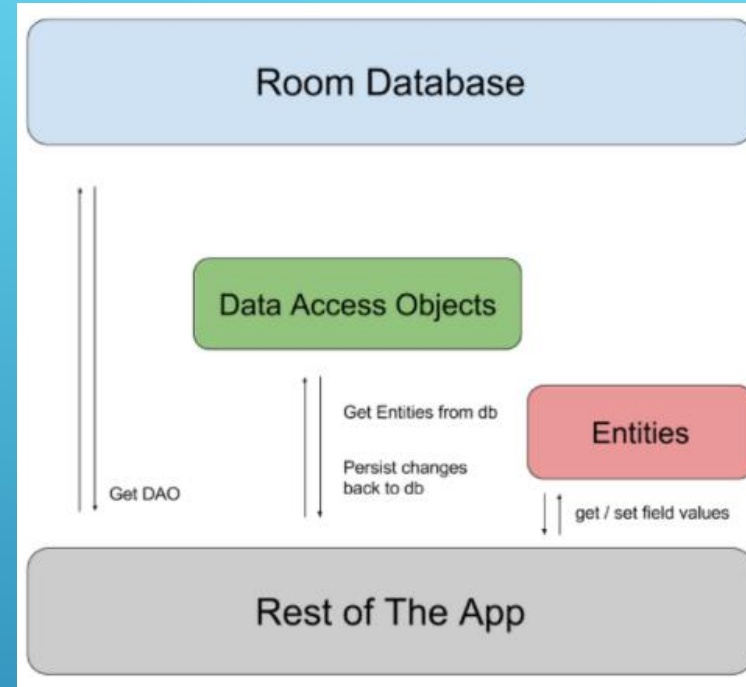
## ► Use of Colours

## ► Orientation



# PERFORMANCE

- ▶ Dark / Light Mode
- ▶ Room Database
- ▶ Use of Coroutines



```
// delete entry in rvList
fun delete(item: EventModel) {
    viewModelScope.launch(Dispatchers.IO) { this: CoroutineScope
        repository.delete(item)
    }
}
```



# SECURITY

- ▶ No user data
- ▶ No app permissions
- ▶ SQL Queries



```
@Query( value: "SELECT * FROM Events WHERE date==:date")  
fun getSpecificEvents(date: String): LiveData<List<EventModel>>
```





# FUTURE WORK

- ▶ User Interface
  - ▶ Support for Dark Mode/ Light Mode
  - ▶ Better use of colours
- ▶ Notifications
  - ▶ Motivational Quotes
  - ▶ Scheduling of Notifications



# FUTURE WORK

- ▶ Adapter
  - ▶ Issues with Binding Data to the View
- ▶ A login system
  - ▶ Firebase?



THANK YOU FOR LISTENING! ANY  
QUESTIONS?



# REFERENCES

- ▶ Notifications:
- ▶ Intensecoder.com. n.d. *Push Notifications tutorial in Android Using Kotlin – Intense Coder*. [online] Available at: <<https://intensecoder.com/push-notifications-in-android-using-kotlin/>> [Accessed 11 May 2021].
- ▶ Room Database:
- ▶ Android Developers. n.d. *Save data in a local database using Room | Android Developers*. [online] Available at: <<https://developer.android.com/training/data-storage/room#kotlin>> [Accessed 11 May 2021].
- ▶ GeeksforGeeks. 2021. *How to Build a Grocery Android App using MVVM and Room Database? - GeeksforGeeks*. [online] Available at: <<https://www.geeksforgeeks.org/how-to-build-a-grocery-android-app-using-mvvm-and-room-database/>> [Accessed 11 May 2021].
- ▶ Kotlin Codes. 2020. *Room Database with Kotlin - Kotlin Codes*. [online] Available at: <<http://www.kotlincodes.com/kotlin/room-database-with-kotlin/>> [Accessed 11 May 2021].
- ▶ Gures, C., 2020. *Basic Implementation of Room Database With Repository and ViewModel | Android Jetpack*. [online] Medium. Available at: <<https://medium.com/swlh/basic-implementation-of-room-database-with-repository-and-viewmodel-android-jetpack-8945b364d322>> [Accessed 12 May 2021].

# REFERENCES

- ▶ Time Picker:
- ▶ Tutorial. 2018. *Android Time Picker Dialog | Example in Kotlin - Eyehunt*. [online] Available at: <<https://tutorial.eyehunts.com/android/android-time-picker-dialog-example-kotlin/>> [Accessed 11 May 2021].
- ▶ Calendar View:
- ▶ Akhtar, W., 2020. *Android CalendarView Tutorial in Kotlin with Examples - Kotlin Android*. [online] Kotlin Android. Available at: <<https://kotlin-android.com/android-calendarview-tutorial-kotlin/>> [Accessed 11 May 2021].
- ▶ Images (for presentation):
- ▶ BuildFire. 2021. *What is a Push Notification and Why It Is Important*. [online] Available at: <<https://buildfire.com/what-is-a-push-notification/>> [Accessed 12 May 2021].
- ▶ Androidinformative.com. 2021. *Dark/Light Mode in Android – Sarabjit's Blog*. [online] Available at: <<http://androidinformative.com/dark-light-mode-in-android/>> [Accessed 12 May 2021].

