Challenges and Solutions in Developing an Event Planning Application

Mark Hahn

Colorado State University Global Campus

CSC475

Professor Thakkar

9/7/24

# Challenges and Solutions in Developing an Event Planning Application

The development of mobile applications involves numerous challenges, ranging from conceptual design to the actual coding and debugging process. During the creation of the event planning application in Kotlin, several issues arose that needed to be addressed to ensure the app's functionality. This paper provides an overview of the challenges encountered during the development process, focusing on the implementation of the Room database for data storage, the use of DAO (Data Access Objects), and other features like RecyclerView. Various strategies were used to resolve these issues, and the lessons learned from this process are discussed.

## Challenges and Solutions

Developing an event planning application in Kotlin presented several challenges, particularly concerning data persistence, implementing complex features, and resolving build configuration issues. Each challenge required careful consideration, and the strategies employed to overcome them provided valuable lessons in mobile development, especially within the Android environment. One of the primary challenges involved working with the Room database for storing and retrieving event data. Room was chosen due to its simplicity and its compatibility with Kotlin, which allows for streamlined data persistence in Android apps Android Developers (n.d.). However, an early issue arose with defining the primary key in the Event entity class. The initial design incorrectly used a String type for the primary key while attempting to auto-generate it. Room, however, only supports auto-generation for numeric types. This mismatch resulted in compilation errors, and resolving this challenge required adjusting the primary key to an appropriate numeric data type. This experience highlighted the importance of understanding the specific requirements and constraints imposed by libraries like Room, especially when dealing with critical features such as auto-generation of IDs.

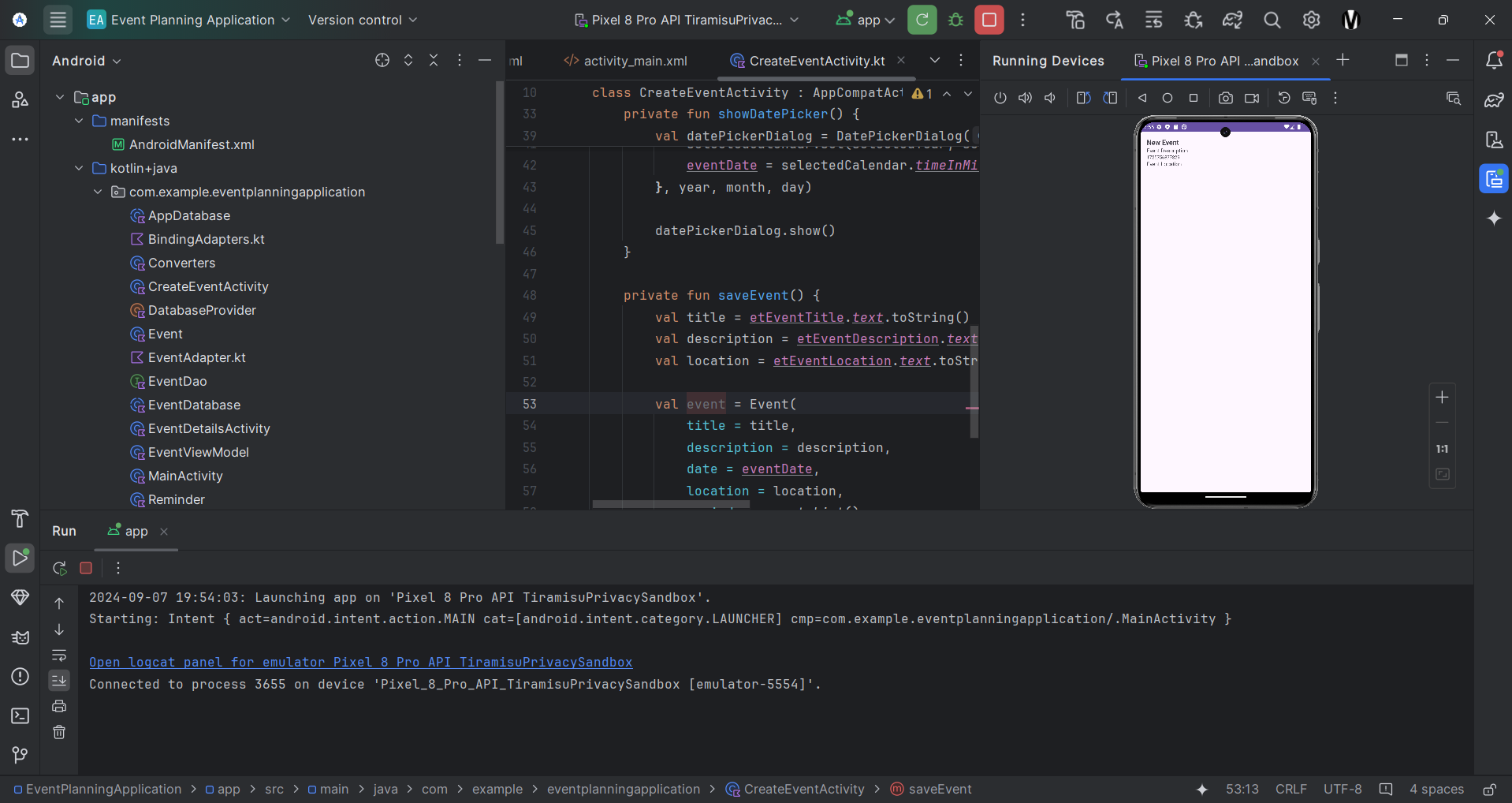
In addition to primary key issues, further complications were encountered with the DAO (Data Access Object) interface, which facilitates communication between the app and the Room database. Specifically, the return types for certain methods were incorrectly defined, leading to compilation errors. For instance, the method responsible for deleting an event by its ID required a proper return type to reflect the number of affected rows, but this was initially overlooked. Similarly, retrieving a list of all events posed challenges as the Room database could not map the query result to the expected data structure. These issues were ultimately resolved by revising the method signatures to align with Room's expected return types, which underscored the importance of careful attention to the details of database interaction methods GeeksforGeeks (2022).

Another significant hurdle was the implementation of a RecyclerView to display event data dynamically within the application. The adapter used to bind data to the RecyclerView encountered multiple unresolved reference errors, particularly in relation to binding the view to the data. These issues primarily stemmed from misconfigurations in the adapter class and improper use of generics. Addressing these errors required a deeper understanding of how RecyclerView works, specifically with regard to correctly handling view holders and binding classes. This challenge was overcome through careful debugging and revision of the adapter’s structure, leading to a functioning dynamic event display.

Beyond the implementation of specific features, the project encountered broader issues related to the build configuration. Errors stemming from the build.gradle.kts file halted progress during the compilation process. These errors were caused by misconfigured repositories and unresolved dependencies. Initially, the configuration did not correctly reference the required libraries and failed to properly set up the project’s repository settings. This challenge was resolved by revisiting the build file and ensuring the appropriate repositories were included. Although this was more of a project setup issue, it reinforced the need for a solid foundation when dealing with complex build configurations in Android development.

Overall, the development of the event planning application in Kotlin provided significant insight into navigating common issues encountered in mobile development. The challenges ranged from technical errors in the Room database and DAO interface to structural issues within the adapter class and build configuration. Each of these obstacles required a methodical approach to problem-solving, emphasizing the importance of a deep understanding of Kotlin and Android's development framework. Despite the difficulties, resolving these issues helped ensure the application functioned as intended and provided a valuable learning experience for future projects.

**Screenshot**



*Event Planning App Running on Pixel 8 Emulator*

# Conclusion

Developing the event planning application involved navigating several challenges, particularly in database implementation, adapter creation, and project configuration. The issues with Room’s database schema, DAO queries, and adapter classes were resolved through careful debugging and appropriate modifications. This process emphasized the importance of understanding Kotlin's type system and the workings of Room for data persistence in Android development. The lessons learned will be invaluable for future projects, ensuring more efficient handling of similar challenges.

# References

Android Developers. (n.d.). Save data in a local database using Room. *developer.android.com.* Retrieved from [Save data in a local database using Room | Android Developers](https://developer.android.com/training/data-storage/room)

GeeksforGeeks. (2022). Android - Data Access Object in Room Database. *geeksforgeeks.com.* Retrieved from [Android - Data Access Object in Room Database - GeeksforGeeks](https://www.geeksforgeeks.org/android-data-access-object-in-room-database/)