Hotel Reservation
Analysis and Design Document
Student: Roca Eric
Group: 30238

Revision History

Date	Version	Description	Author
22/03/19	1.0	Entered initial details	Roca Eric

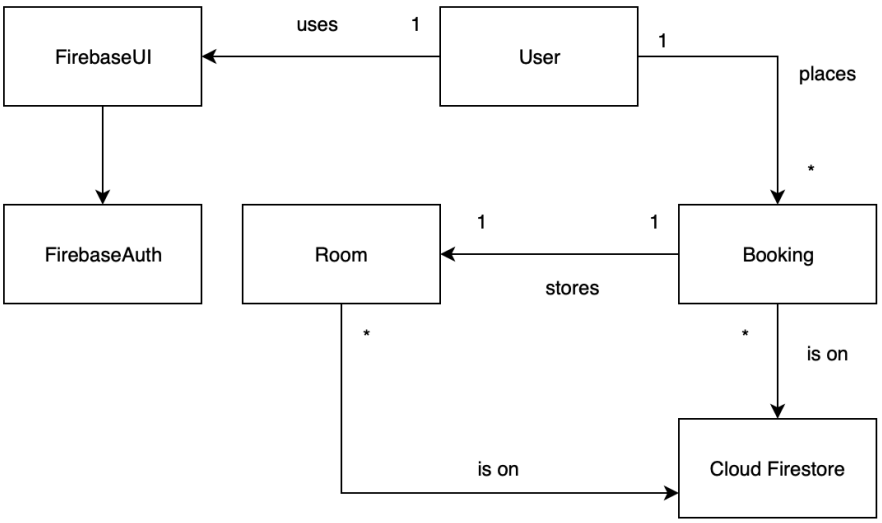
Table of Contents

I. Project Specification

This application allows users to book hotel rooms from their iOS phones. The user can view and check the various rooms available and book them using online payment. The application also allows users to select additional facilities like jacuzzi, swimming, and meals. The app then calculates the total cost of the booking. Once the user makes the payment, the app provides an online receipt to the user in the form of an email. When selecting rooms, only the rooms that are available will be shown to the user. Once the user completes the payment of their booking, and admin is notified with the details of the booking. When the user visits the hotel, they must show their receipt for the booking. This application requires a smartphone connected to the Internet.

II. Elaboration – Iteration 1.1

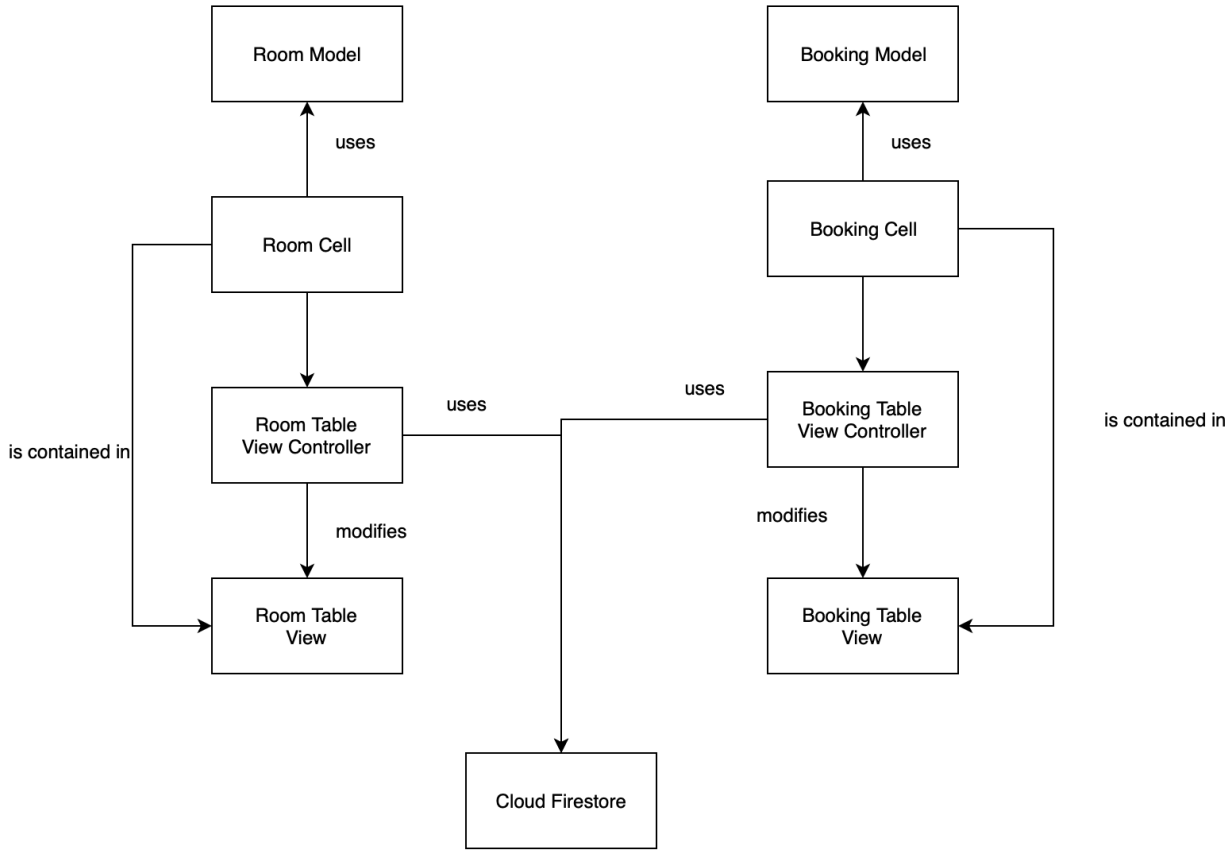
1.Domain Model



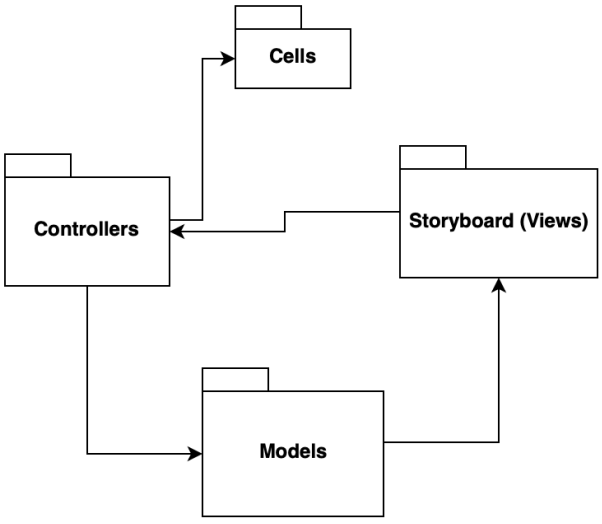
2.Architectural Design

2.1.Conceptual Architecture

The conceptual architecture of the project is Model View Controller (MVC). This architecture design was used because it is the fundamental iOS pattern. Many **iOS** frameworks, like UIKit, use the **MVC** pattern to structure data flow and messaging. The project also uses the client-server architecture, with the iOS app being the client and the Firebase services being the server.

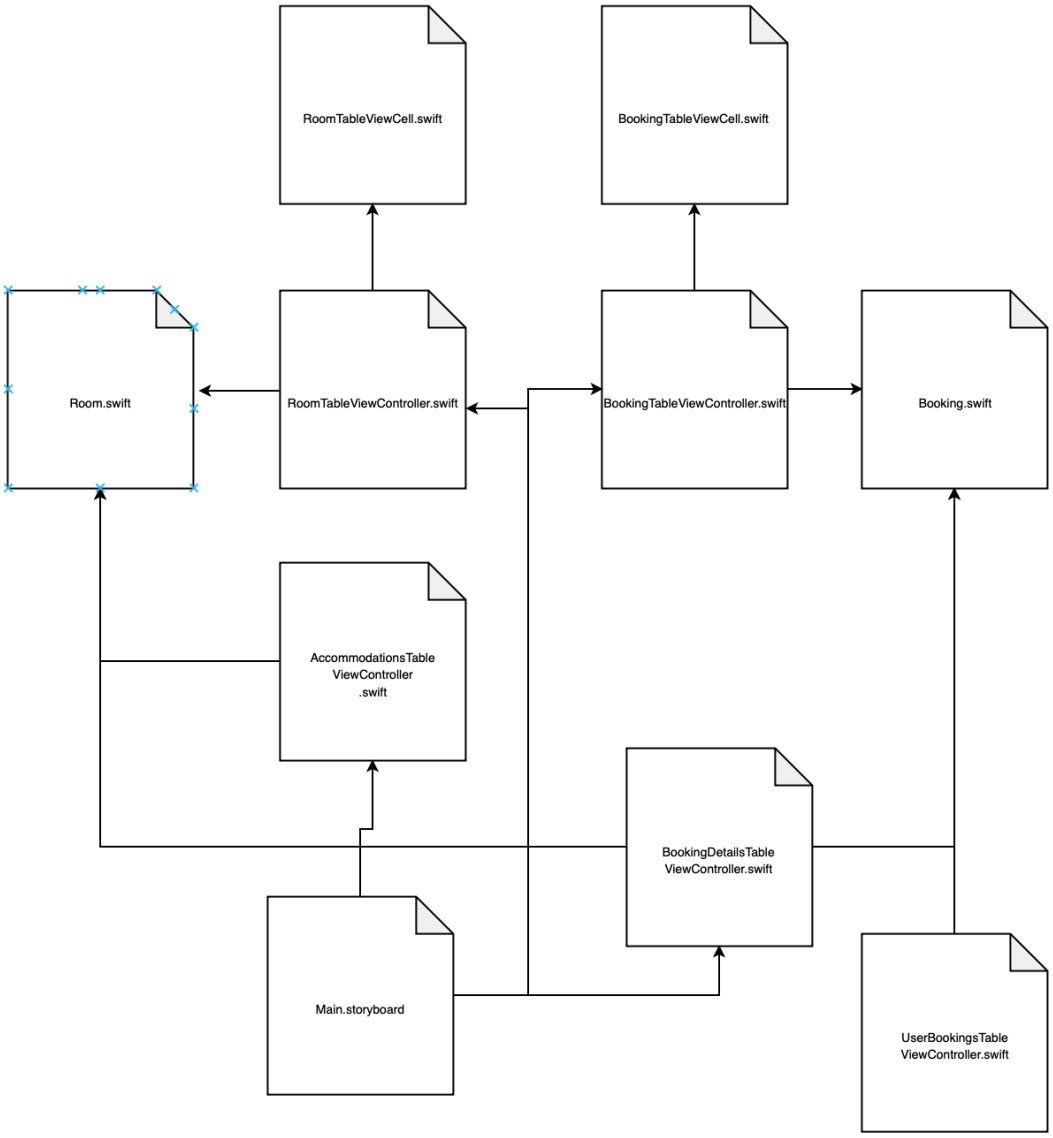


2.2.Package Design

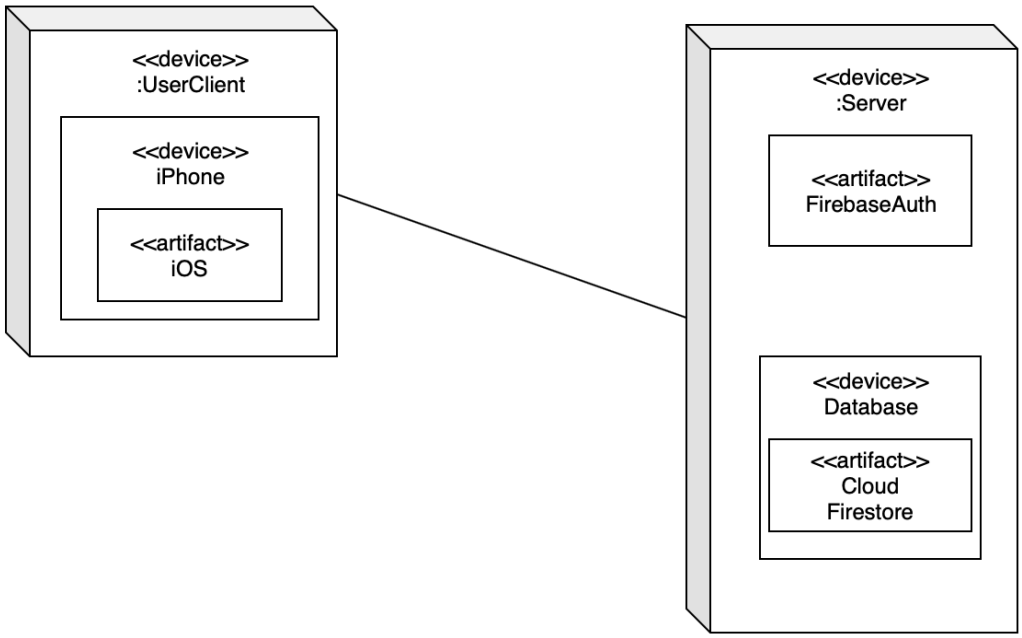


2.3.Component and Deployment Diagrams

Component Diagram



Deployment Diagram



III. Elaboration – Iteration 1.2

1. Design Model

1.1. Dynamic Behavior

View Rooms

1.2. Class Design

[Create the UML class diagram; apply GoF patterns and motivate your choice]

2. Data Model

The project has two models, Room and Booking.

Room data model:

```
isAvailable: true
number: 100
price: 100
type: "Single"
```

Booking data model:

```
hasJacuzzi: true
hasMeal: true
hasSwimming: true
room: 101
roomPrice: 150
totalPrice: 195
user: "test"
```

3. Unit Testing

Unit testing can be used on the data model classes.

IV. Elaboration – Iteration 2

1. Architectural Design Refinement

[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]

2. Design Model Refinement

[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]

V. Construction and Transition

1. System Testing

Unit testing was used to check for invalid data introduced in the model classes.

2. Future improvements

In the future, the system could have support for more hotels, or checking availability based on days.

VI. Bibliography