CA400 Functional Specification

"Parklet"

Shaun Kinsella – Student ID -: 14740175
Mike Purcell – Student ID -: 15446908
Supervisor: Dr Stephen Blott
Date Completed: 22/11/19

Table of Contents

1. Introduction

1.1 Overview

"Airbnb for parking"

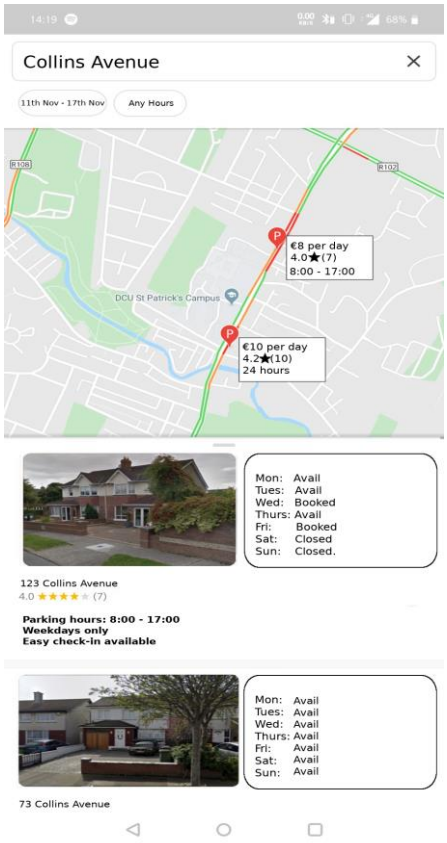
The aim of our project will be to develop an android app that will allow homeowners to easily rent out their driveways as parking spaces to commuters with minimal effort. And likewise enable commuters to easily view and book parking spots with competitive rates.

The app will allow homeowners to list their driveway property as available to rent. They will be able to set their own rates on a daily, weekly or even monthly basis and can choose to give discounts for booking multiple days. The homeowner can choose to allow overnight parking or a time frame such as parking between 8am to 6pm regardless of the amount of days booked.

From the renter's perspective the app will allow them to choose an area they wish to park nearby and show them all available spaces to rent that fit their requirements. They will be able to view the availability of the driveways to rent for their preferred dates and compare the rates for each property.

We will be utilising NFC to provide a hassle-free way to check-in for both renters and homeowners. The homeowner can optionally register a property NFC tag that the renter can use in conjunction with the app to tap and alert the owner that they have arrived. Likewise, when the renter is leaving, they will tap and the homeowner will be notified through the app's push notification that their driveway is now free.

Mock-up of the park space search UI



1.2 Glossary

- Android: Mobile operating system
- Android Studio: IDE built upon IntelliJ targeting android app development
- NFC: Near field communication, a protocol allowing two electronic devices to pass data between each other when in close proximity.
- QR code: Quick response code, machine readable label that can be used to store information or links.
- Firebase: Realtime cloud database allowing us to query and sync information in real time.
- Google maps API: Leveraging google maps API will allow us to implement the geographical aspect of our app.

2. General Description

2.1 Product / System Functions

The goal of the product is to allow homeowners to be able to lease their driveway and commuters to book those properties with ease. This will be accomplished through the android app by using the google maps API to display all properties available in a given area of interest to the commuter.

The homeowner will be able to set the hours of availability and pricing with minimal input through the android app.

The app will handle management of bookings, storing them on a firebase database instance, ensuring that both homeowner and commuters have the latest snapshot of bookings for a given property.

The check-in functionality will be accomplished using NFC tags at the homeowner's property to give notice of the commuter's arrival and then to ensure they have left by the agreed time. The app will handle the writing of the unique property identifier to the card as well as the reading by the commuter's phone to confirm the arrival and leaving of the property. For users who do not wish to utilise NFC tags, we will give the option of generating a QR code that may be exported and printed to use for the check-in system instead.

Analysis on historical bookings for a given area will be utilised to provide price listing advisement to homeowners to ensure they're staying competitive in their area. Likewise, this will be provided to commuters to so they are aware they're getting a reasonable deal on the rental of the driveway.

The app will allow all users of the app to give a review in the form of a star-based system following the booking and use of the parking space, which will then be displayed on property listings and commuter profiles.

2.2 User Characteristics and Objectives

Assumed User Knowledge:

- Basic knowledge of android OS phones and installing apps from Playstore.
- Familiarity with NFC or QR code usage.

User Characteristics:

- **Homeowners:** Homeowners with available car parking space on their driveway. They might be homeowners who do not drive, commute elsewhere during the day or just have the extra space to facilitate more cars. They might be already rent their driveway out in an ad hoc manner and simply wish to streamline the process.

- **Daily Commuters:** commuters in congested areas with limited parking available, offering them an alternative parking solution. Whether it's a cheaper alternative or to allow closer access than normal car parking services offer.

- **Event attendees:** Events like Slane or the Skerries 100 put a massive strain on local car parking services. Festival goers could book a car parking space outside of the festival grounds to avoid the panic of trying to secure a spot and the congestion that follows trying to leave the venue car park.

- **Holiday makers:** looking for cheaper alternatives to airport parking in the case of weekly or monthly parking rental.

User Objectives:

- For homeowners, the ability to easily list and rent their driveways.
- The ability to view whether a commuter has vacated the property on time.
- For commuters, the ability to easily view and book driveways.
- For both users, easy management of their bookings.
- (**Desirable)** The ability to list multiple properties per account for homeowners with more than one driveway.
- (**Desirable)** One to one messaging between homeowner and commuter for each booking.

2.3 Operational Scenarios

1. **User Create account**
   Objective: A new user wishes to sign up to Parklet
   Actions: User opens the app with no current login. They are asked to provide an email address or sign in using their google account.
   Success End Condition: The user account is successfully created and they are presented with the option of listing a property or adding a vehicle to allow for bookings.
   Failure End Condition: User provides invalid email address or attempts to add an already existing email. User is informed to try a valid email or to login with the already existing one.

2. **Homeowner adds property**
   Objective: A homeowner wishes to list a property to rent out to commuters.
   Actions: From the home screen the user selects manage properties, and then add property, and is then prompted to fill in the necessary information, the address of the property, their days and times of rental availability and pricing information.
   Success End Condition: User provides all necessary information, is asked if the property is correctly listed on map view and if they wish to include either an NFC tag or QR code to print out.
   Failure End Condition: User submits incomplete or invalid address. User is asked to recheck highlighted input fields to correct them.

3. **Homeowner writes check-in tag for property**
   Objective: The homeowner wishes to add or rewrite a tag to be used for property check-in
   Actions: For a new property after entering the details the user is prompted to select either NFC or QR code. For an existing property the user can select to rewrite the tag from manage properties and selecting the relevant property. For NFC the user will be prompted to make sure NFC is enabled on the phone and to press an NFC against. Selecting the QR option will generate a QR code.
   Success End Condition: The user has successfully written to the NFC tag and will be notified; they can then attach it in a visible area to allow check-ins. For QR code they can export it to print out.
   Failure End Condition: The NFC tag write was not successful; the user is notified to try again.

4. **Homeowner edits property**
   Objective: The homeowner wishes to update details of the property such as pricing or times. They may also temporarily suspend bookings on the property for given dates if they wish, or remove the property fully.
   Actions: From the home screen the user selects manage properties, and then clicks the edit icon on the relevant property. They then can select the edit action they wish.
   Success End Condition: The user saves all relevant changes to the property and receives feedback that the update was successful. If the user deletes or suspends a property with outstanding bookings, they are asked to acknowledge this. All commuters with bookings are notified that their booking is affected.
   Failure End Condition: The user fills the fields with invalid pricing or attempts to change availability to a date already passed. They are informed of this and asked to correct it.

5. **Commuter adds Vehicle**
   Objective: The commuter wishes to add a vehicle under which they can book driveways with.
   Actions: From the home screen the user selects manage vehicles and selects add. They are prompted to enter the model, colour and registration of the vehicle.
   Success End Condition: The user adds all valid and relevant information and is informed they can now search for driveways to rent. They are brought back to the manage vehicles page.
   Failure End Condition: The user fails to provide the necessary information and is prompted to fill in the highlighted fields correctly.

6. **Commuter edits Vehicle**
   Objective: The user wishes to update the vehicle, such as colour, or removing the vehicle from the account.
   Actions: The user selects manage vehicles from the home screen and selects the relevant vehicle to edit or remove.
   Success End Condition: The user is asked to confirm all changes and in the case of deletion is warned if there are outstanding bookings for that particular vehicle.
   Failure End Condition: User enters invalid information or leaves a necessary field such as colour blank. They are notified and asked to correct the highlighted fields.

7. **Commuter searches for driveway to rent**
   Objective: The user has a current valid vehicle and wishes to book a driveway.
   Actions: From the home screen the user selects "Search Driveways" and is prompted to enter a location and the date range they wish to search for.
   Success End Condition: The user is presented with a map of the relevant area decorated with pins of driveways available to rent with a small summary attached to each one.
   Failure End Condition: If the user enters an invalid area, date or if the given area has no available driveways for the given area they are informed of the reason and asked to correct the location or try a larger search respectfully.

8. **Commuter Places booking**
   Objective: The user has selected a property they wish to book.
   Actions: Selecting the property pin on the map display will bring up a window with more details for a given property. The user can input the days they wish to book and is given a summary price before they confirm the booking application.
   Success End Condition: The user is given a confirmation that the booking has been placed. Their profile's listing of bookings is updated. The property owner is then informed through push notification.
   Failure End Condition: The user might have selected a date range such as a week that clash with a single day booking. They are informed to correct the booking dates.
   If the user takes too long in making the booking it may become unavailable and will be informed that the booking is no longer available.

9. **Commuter Cancels booking**
   Objective: A commuter wishes to cancel a pre-existing booking.
   Actions: From the home screen the user selects "My parking bookings". The user then selects cancel booking for the relevant booking. The user is then warned of any penalties listed for cancelling the booking too close to the date and is asked to confirm.
   Success End Condition: The user is notified of the successful cancellation and returned to the "My parking bookings" page.
   Failure End Condition: User fails to confirm cancellation, booking remains in place and is still listed as an active booking under "My parking bookings".

10. **Homeowner cancels booking**

    Objective: The homeowner wishes to cancel a current booking for their property.

    Actions: From the home screen the user selects “My property bookings” and selects the relevant booking under that property. They then select cancel and are informed that cancelling a booking may incur a negative review and will give a full refund back to the commuter before confirming the cancellation.

    Success End Condition: The homeowner is given confirmation that the booking has been cancelled and the commuter is notified of the cancellation and refund. The application returns to the “My property bookings” screen.

    Failure End Condition: The homeowner fails to confirm they wish to cancel the booking and it remains valid. If the cancellation fails to write to the database they informed and can try again.

11. **Commuter Checks into Property**

    Objective: The commuter has arrived to the property and wishes to check-in as is required.

    Actions: With NFC enabled on their phone the commuter taps their phone against the NFC tag on display. In the case of a QR code they can scan it with a QR reader.

    Success End Condition: The commuter receives a confirmation on screen that they are now successfully checked in and a reminder to remember to check out. The home screen is then displayed. The homeowner receives a notification that the commuter has arrived at their property.

    Failure End Condition: If the tag fails to read, the commuter is prompted to try again. If they somehow arrive at the wrong property, they are informed that they have no existing booking for the property.

12. **Commuter Checks out of Property**

    Objective: The commuter is retrieving their vehicle from the property and wishes to check-out of the property.

    Actions: With NFC enabled on their phone the commuter taps their phone against the NFC tag on display. In the case of a QR code they can scan it with a QR reader.

    Success End Condition: The user is notified on screen that they are now successfully checked out and a reminder to leave a review for the homeowner. The homeowner receives a push notification that the commuter has now left and their driveway is now free.

    Failure End Condition: If the tag fails to read, the commuter is prompted to try again. If they have already checked out, they are notified of this.

13. **User submits Review**

    Objective: Following a completed booking either a homeowner or commuter may give a rating for the property or commuter respectively.

    Actions: From the home screen selecting either “My property bookings” or “My parking bookings”, the user can select the relevant booking and click the star icon to leave a star-based review for the relevant entity. They are then asked to confirm their review.

    Success End Condition: The user receives confirmation that the review has been placed.

    Failure End Condition: If the user fails to confirm the review or cancels, it will not be placed and they will return to the booking screen.

2.4 Constraints

1. **Time**: The project is on a deadline of 11th of May 2020 and as such the scope of the project must be achievable within this time frame. In light of this we will aim to deliver the projects features in stages, ensuring that we have a working prototype at the end of each sprint. Additional features will be built upon base functionality.

2. **Scope**: Due to the aforementioned time frame, we must be careful in deciding the exact scope of the project, what our expected outcome of the project will be. This will require constant assessment subject to the progress of the development of the app's base functionality and subsequent features, each will be ranked in order of importance.

3. **Performance**: The app must be able to process all user requests in a quick and efficient manner, whether that is adding a new property, searching for one, booking a park space or placing a review. This is a separate constraint from UX (User experience) or usability of the app, rather the throughput of the app.

4. **User Experience (UX)**: The app must be usable by any user with a base understanding of android apps. Each element of the UI that can be interacted with must give a clear affordance of what it is responsible for and how it can be used. In particular the map view and listing of driveways to rent must be conveyed in a simple but informative manner.

3. Functional Requirements

Requirement	1
Description	The system must be able to create a user's account
Criticality	Critical as all user activity is based off the assumption the user has an account
Technical Issues	Implementation of user authentication and validation of all supplied user information.
Dependencies	N/A
Use Cases	1

Requirement	2
Description	The system must be capable of storing user's data.
Criticality	Critical as all functionality involves storing or sending data from the user
Technical Issues	Ensuring that all data is stored efficiently and logically on the database.
Dependencies	1
Use Cases	2, 5, 7

Requirement	3
Description	Google maps integration
Criticality	Critical for both homeowners and customers to be able to view and book properties through the UI
Technical Issues	Correctly implementing the map view to display available properties to rent. This solution must be efficient and offer clear affordances for the map's usage to users.
Dependencies	1, 2
Use Cases	2, 7, 8

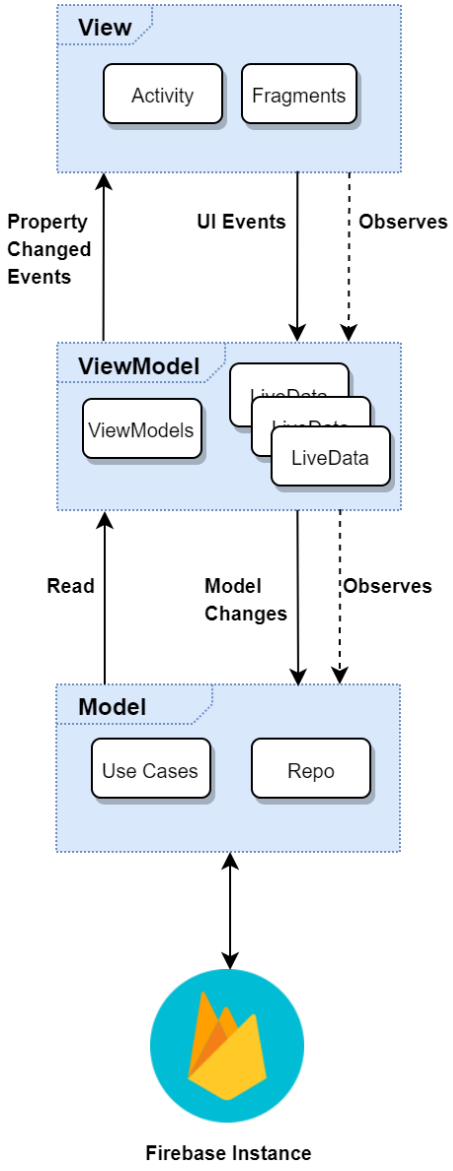
Requirement	4
Description	User profile management
Criticality	Critical as the user may want to update a detail about their driveway or to update a booking from the customers perspective
Technical Issues	Ensuring that any updates to a user's profile, vehicles or properties do not invalidate current bookings.
Dependencies	1, 2
Use Cases	2, 4, 5, 6, 8, 9, 10, 13

Requirement	5
Description	Setup of NFC/QR code functionality
Criticality	Critical as it is used for homeowners to know the status of their driveway booking.
Technical Issues	Implementing NFC writer and reader within the app. The app must be associated with the NFC tag content to automatically open and check the user in. Implementation of the QR generator which must also be associated with the app upon reading.
Dependencies	1
Use Cases	3, 11, 12

Requirement	6
Description	App must provide price advisement for homeowners when listing properties and for commuters when searching for properties.
Criticality	Medium
Technical Issues	Implementing historical analysis on previous successful booking rates for a given area in an efficient manner.
Dependencies	1, 2, 4
Use Cases	13

Requirement	7
Description	Implementation of a star based “blind” review system
Criticality	Medium
Technical Issues	Implementing the review system so neither user is able to view the associated review until both have submitted one. To ensure unbiased reviews.
Dependencies	1,2,4
Use Cases	13

4. System Architecture



The System Architecture will follow **a** Model-View-ViewModel (MVVM) pattern. This will allow separation of concerns and loose coupling between the UI and the business logic of the application, facilitating unit testing and modularity.

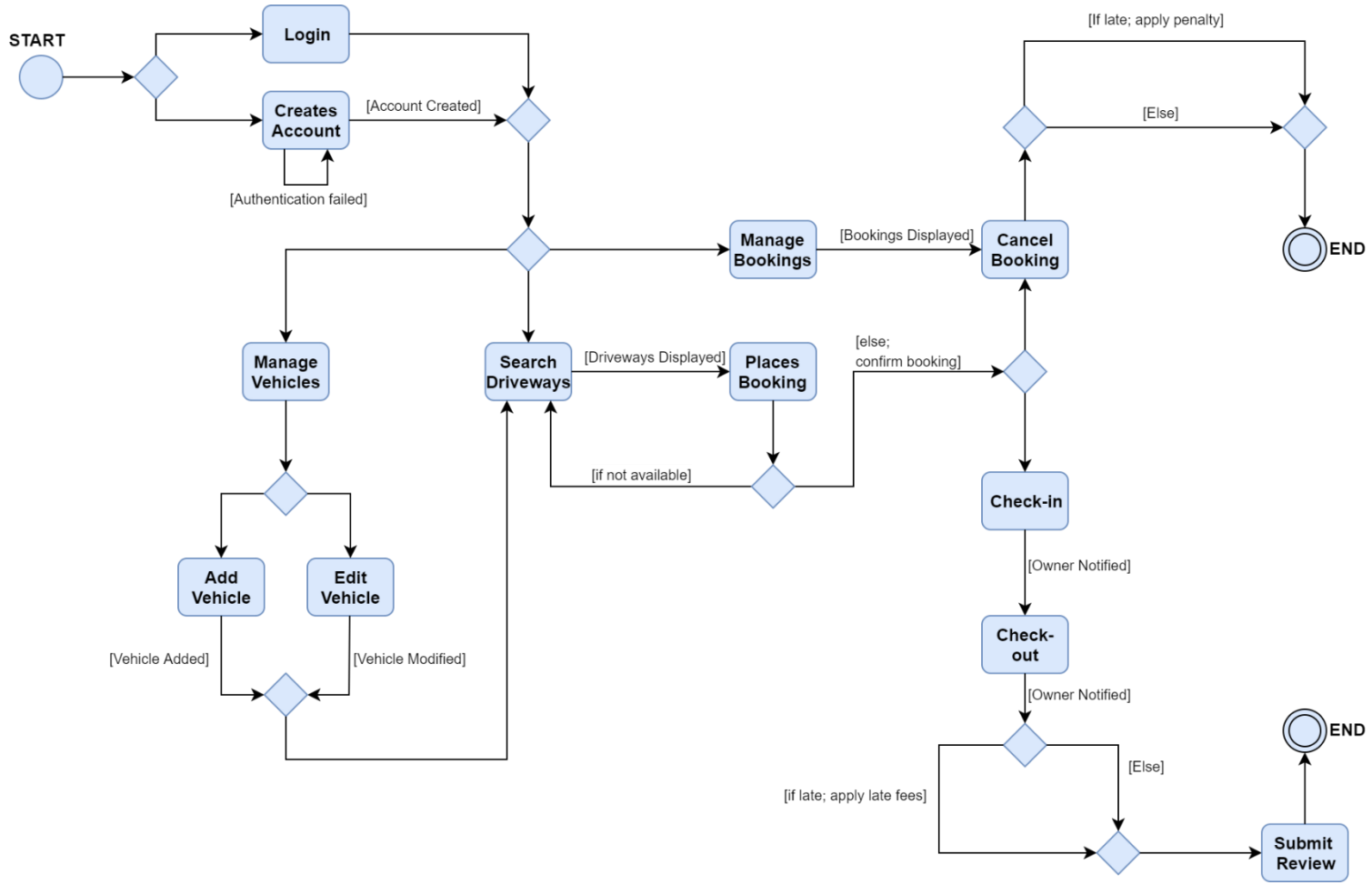
The View:
The View is responsible for the application's UI consisting of activities and fragments, it is completely devoid of any business logic. A View observes the relevant ViewModel for data to update the UI and pushes any UI Events or user interactions to the ViewModel.

The ViewModel:
The ViewModel is responsible for reading data from the Model, transforming it to be consumed by the View. The ViewModel itself does not contain a reference to the View, instead the View is subscribed to a ViewModel and observes it for changes to the data so it can update the UI. This ensures that the ViewModel is decoupled from the View, and makes the unit testing of each ViewModel easier.

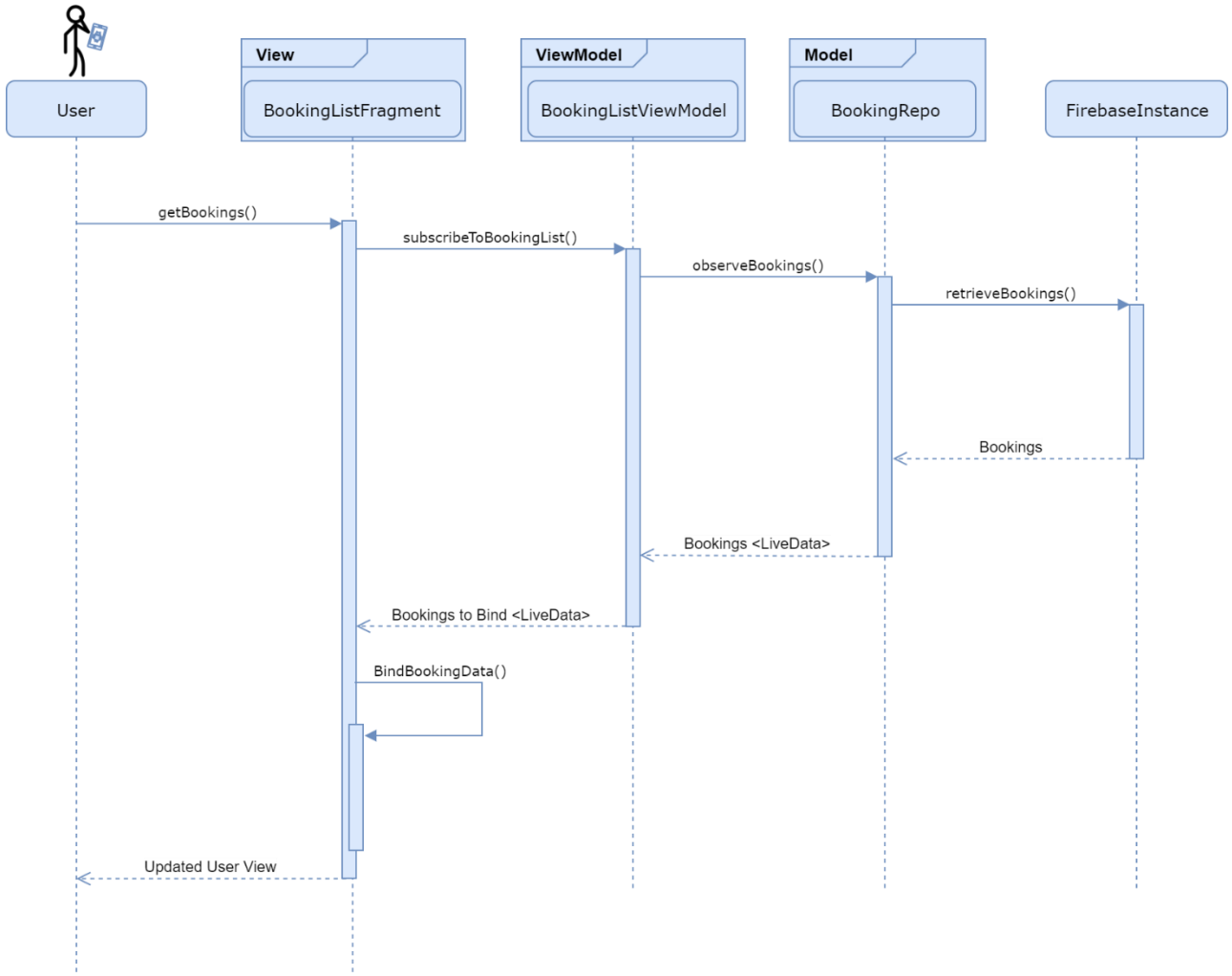
The Model:
The Model encapsulates the business logic of the application and will also host repository pattern classes that are responsible for composing data from multiple sources, in this case a firebase instance. This again decouples business logic classes from the data source, allowing us to easily test business logic while allowing for easy integration of multiple data sources or migration.

5. High-Level Design

5.1 Activity Model: Commuter Booking



5.2 Sequence Diagram: User retrieves bookings



6. Preliminary Schedule

This section provides an initial version of the project plan, including the major tasks to be accomplished, their interdependencies, and their tentative start/stop dates. The plan also includes information on hardware, software, and wetware resource requirements. The project plan should be accompanied by one or more PERT or GANTT charts.

7. Appendices

Specifies other useful information for understanding the requirements.