# 1. ParkHere

### 1.1 Project Title
The White Walkers

### 1.2 Authors

- Romsin Khoshabian - 008570978
- Michelle Lai - 009769721
- Rohit Mathew - 012539657
- Pearl Or - 010401911


# 2. Preface
The implementation stages of the ParkHere system have begun with a focus on the core logic and services to be offered. The design and steps to implement these core functionalities will follow the detailing in the previous design and requirement specifications. This implementation specification will address changes proposed by the client and from the enactment of the implementation, this includes architectural and detailed design changes. Throughout and after the implementation, the testing of singular features will be recurring and constantly reevaluated.

# 3. Introduction
The ParkHere system's implementation starts with the setup of the proposed architecture, the client server pattern. The User Interface, Server, and Database will be handled using Android Studio and Google's Firebase Service. Initially, the implementation will have a focus on the functionalities that represent the core of the ParkHere system. These core functionalities include: account registration, user login, listing creation, listing viewing, sending requests, and replying to requests. With these functionalities successfully implemented, the user shall be able to use the ParkHere application to display their parking spot listing and respond to requests. Architectural and detailed design changes from the design specification will be reported for reference in this implementation specification. The changes will consider implementation constraints, capabilities, improvements, etc. Future implementation stages shall address the remaining functionalities, including payment transactions, ongoing listing and request cancellations, and location features.

# 4. Architectural Change

### 4.1 Firebase for Server and Database Implementation
The implementation for the Server and Database layers will be delegated to Google's Firebase service. Firebase shall be used as the primary backend for the ParkHere application. This change encourages a more consistently fluent implementation process. The layers are organized and predefined because Firebase offers API and methods to interact with each layer more seamlessly and with common security measures already considered.

## 5. Detailed Design Change

### 5.1 Assigning Role to the User

The User shall not be assigned a variable role, renter or owner, because this prevents the User from accessing both renter and owner options. A User shall be able to create Listings for their parking spots and simultaneously Request bookings for use. Any renter and owner features will be enabled for all Users. The UI has been modified to show and access the User's Listings and/or booking Requests when any are present.

### 5.2 Parking Spot Listing Tracking Rental Duration

A Listing shall have a variable to represent if the Listing is booked or available but it will not have to keep track of the duration it is rented. The responsibility will be delegated to the booking Request. This change accommodates the Listing for split bookings or multiple renters for one Listing.

### 5.3 Inbox for Viewing All Requests

The Chat will be renamed to Inbox and shall be used for Users to view booking Requests for their Listings. The Inbox UI shall display all ongoing Requests, instead of one Request at a time. When a Request has been submitted, the owner's Inbox shall be updated to receive and display snippets of the Request. The full Request information is viewable upon click. The change allows for a simplified messaging and requesting service between the Owner and Renter. The UI is enhanced for the Owner to see multiple snippets of Requests at the same time.

### 5.4 Sending and Viewing Request

The ChatMessage shall be renamed and combined with Request. Once a booking is desired, the renter shall send a Request will their message and booking parameters instead of starting a Chat with ChatMessages. The owner will view the message and contents from the Request object instead of a ChatMessage. When Requests are selected to view from the Inbox, the Request will have variables to keep track and display the subject, message, Listing information, and Renter information. Since the Chat has dissolved to the Inbox, this accommodates the simplified messaging interaction by downsizing the number of messages that a Renter can send to an Owner through the ParkHere application.

### 5.5 Homepage Shows Listings and Bookings

Since the User will not be assigned a role of Renter or Owner, the Homepage will display both active Listings and Bookings for the User. This change helps to enforce the previous change of eliminated the role variable for the User.

### 5.6 TimeDetails To Track Time and Dates

The TimeDetails object shall be used to hold a Request's starting date, starting time, ending date, ending time, and total pricing. The formats for the dates and times will be checked and

converted for usability. The price will be calculated based off of the TimeDetails' total number of hours and the hourly rate specified by the owner of the Listing. This change expands on the TimeStamp variable originally planned to be in Request so it considers time and includes a range of dates. The TimeDetails shall be held as a variable in a Request.

### 5.7 Price Calculation per Request

To handle multiple Requests for a listing, the price calculation will be handled in a Request. The price shall be determined by the number of hours specified by the renter of the Request and price per hour specified by the owner of the Listing. This change accommodates for the split booking and TimeDetails additions. Users will be allowed to make Requests and get pricing for partial durations of the Listing.

## 6. Requirement Change

### 6.1 Split Booking

Enabling the option for a renter to split a booking by their choice of days or hours will change our original design for the ParkHere application. Changes will be made in the parking spot's Listing class to keep track of the range of time and dates that the listing is available. The Booking Request shall have to keep track of the range of time and dates that the renter is requesting to book. Both the Listing and Request classes should have additional variables to keep track of the start and end date/time.

The Listing's listingStatus shall only be updated to rented or unavailable when the entire duration of the listing is booked. Listings will continue to appear as available in searches until the entire duration is fully booked. A Listing cannot be linked to a single renter because the requirement change allows a Listing to have multiple renters. Instead the Request will have variables for the Listing ID that it refers to. Multiple Requests for the same Listing will be checked so that they shall not overlap in the requested times or days.