Testing Documentation

“Parklet”

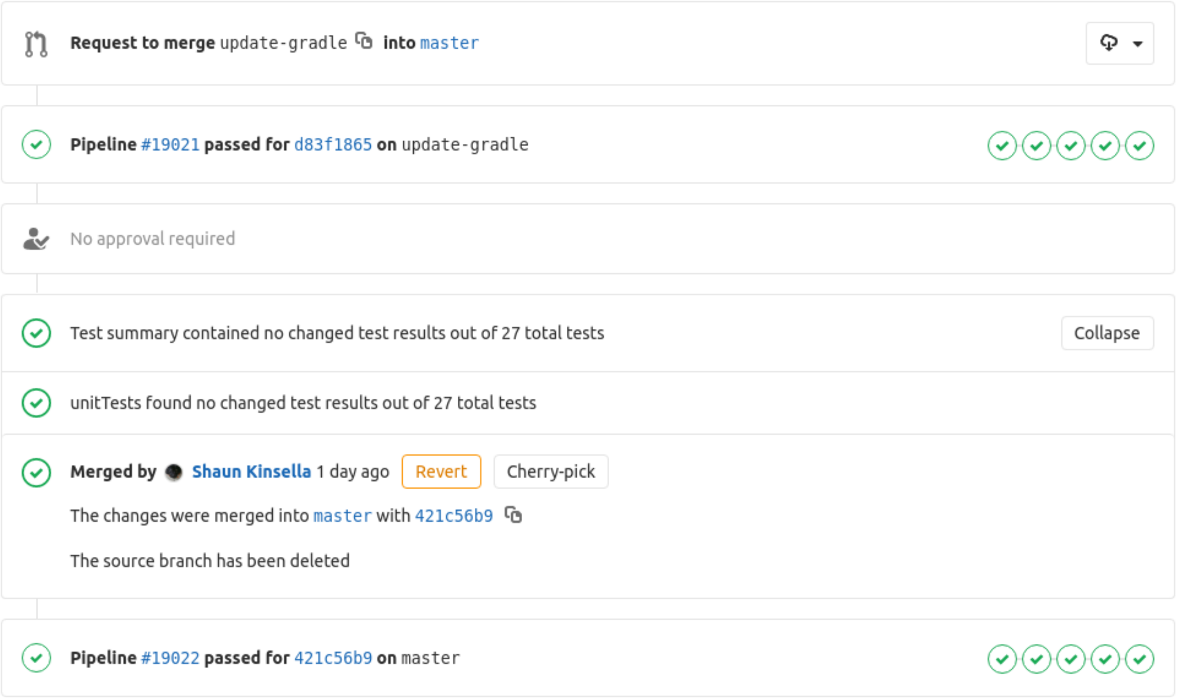
Shaun Kinsella – Student ID -: 14740175
Supervisor: Dr Stephen Blott
Date Completed: XX/XX/XX

Contents

Testing strategy

In order to thoroughly test the ParkLet app, a combination of ad hoc, unit, manual use case scenario, regression and instrumented tests were utilised. Using the Model View ViewModel (MVVM) architecture facilitated unit testing as each layer of the architecture is loosely coupled.

From the beginning of the app's development a Gitlab CI runner test script was created. This test runner would be triggered on every merge request and commit. The project repository was set to only accept merge requests on successful completion of each test stage. Below shows a merge request having been automatically merged after all stages of the pipeline passed. This setup enabled regression testing to be performed, ensuring any new changes did not break functionality.



Request to merge update-gradle **into** master

Pipeline #19021 **passed for** d83f1865 **on** update-gradle

No approval required

Test summary contained no changed test results out of 27 total tests

unitTests found no changed test results out of 27 total tests

Merged by Shaun Kinsella 1 day ago

The changes were merged into master with 421c56b9

The source branch has been deleted

Pipeline #19022 **passed for** 421c56b9 **on** master

In the case of a test failure, the code causing the failure would be amended and tests would then be re-run.

Unfortunately, it was found that performing unit tests on the apps Activity and Fragment classes would not be possible. This was found to be an issue with the migration to the AndroidX framework. Parklet itself uses Fragments from the AndroidX framework, which are incompatible with the older method of launching fragments in isolation.

The newer method utilises Fragment and Activity scenarios in order to launch the test class in a container. However, it was also found that older packages which relied on the Android.V4.Support framework were also not compatible with the newer method. When attempting to run the newer tests, a build failure with the error message of:

```
“The given artifact contains a string literal with a package reference
'android.support.v4.content' that cannot be safely rewritten. Libraries
using reflection such as annotation processors need to be updated manually
to add support for android x.“
```

No further detail was given other than the above. Investigating online pointed towards one library in particular but was not used by ParkLet. So, I manually checked the imports/dependencies of each of ParkLet’s classes and found nothing to be using the support framework. This leads me to believe that one of the libraries Parklet depends on has outdated dependencies itself. Meaning that I could try to remove each library one by one to identify the offending one, or to refactor ParkLet to use the older framework, neither of which would be feasible within the given timeframe.

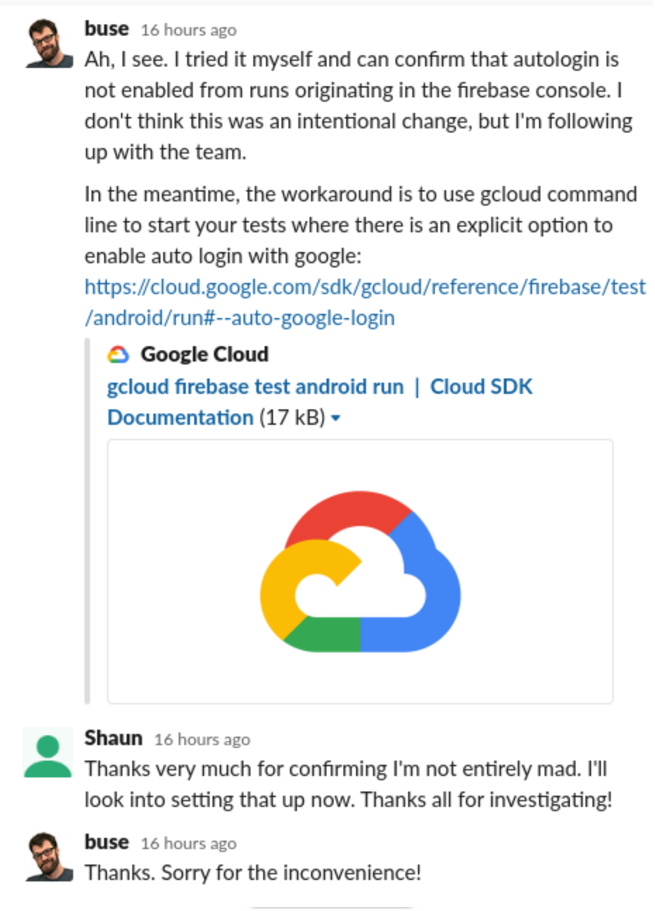
In order to mitigate against the lack of unit tests for the UI aspect of the project, a use case scenario test method was used. For every major feature of the app, both common and uncommon test cases were elicited and their expected outcomes documented. This formed the basis for end to end tests as well. In conjunction with this, Firebase test lab was set up to perform automated instrumentation tests on multiple devices.

Due to the current pandemic, actual user testing would be un-achievable. Testing online with users would require use of their own devices and validation of tasks being successfully completed would be difficult. Especially in the case of NFC related tests. So, the use case scenarios and firebase test lab tests were used to also fill this gap.

Firebase Test lab

Use of Firebases Test lab enabled me to run instrumentation style tests on matrices of real and emulated devices in parallel. These tests would provide a very detailed summary of test results, e.g. performance metrics, a video of each of the tests, screenshots for every action taken and a graph of the emulated users navigation through the app.

I was required to set up the tests from the GCloud CLI interface as tests began to fail randomly. After speaking to 3 Firebase developers on their official Slack channel, it was found that an unintended change had been made, disabling their test runners automatic sign in using google authentication and the changes were not reflected in the firebase portal GUI. Another issue encountered with tests failures was found to be due to virtual machine creation failing.



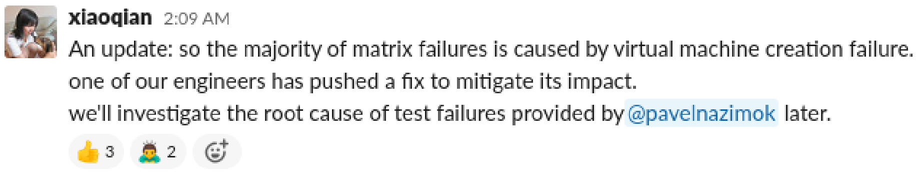
buse 16 hours ago
Ah, I see. I tried it myself and can confirm that autologin is not enabled from runs originating in the firebase console. I don't think this was an intentional change, but I'm following up with the team.

In the meantime, the workaround is to use gcloud command line to start your tests where there is an explicit option to enable auto login with google:
https://cloud.google.com/sdk/gcloud/reference/firebase/test/android/run#--auto-google-login

Google Cloud
gcloud firebase test android run | Cloud SDK Documentation (17 kB) ▾

Shaun 16 hours ago
Thanks very much for confirming I'm not entirely mad. I'll look into setting that up now. Thanks all for investigating!

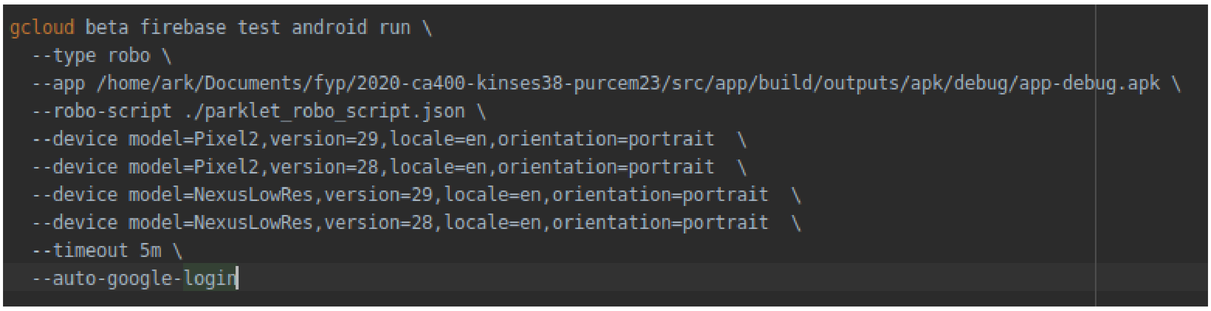
buse 16 hours ago
Thanks. Sorry for the inconvenience!



xiaoqian 2:09 AM
An update: so the majority of matrix failures is caused by virtual machine creation failure.
one of our engineers has pushed a fix to mitigate its impact.
we'll investigate the root cause of test failures provided by @pavelnazimok later.
👍 3 🙇 2

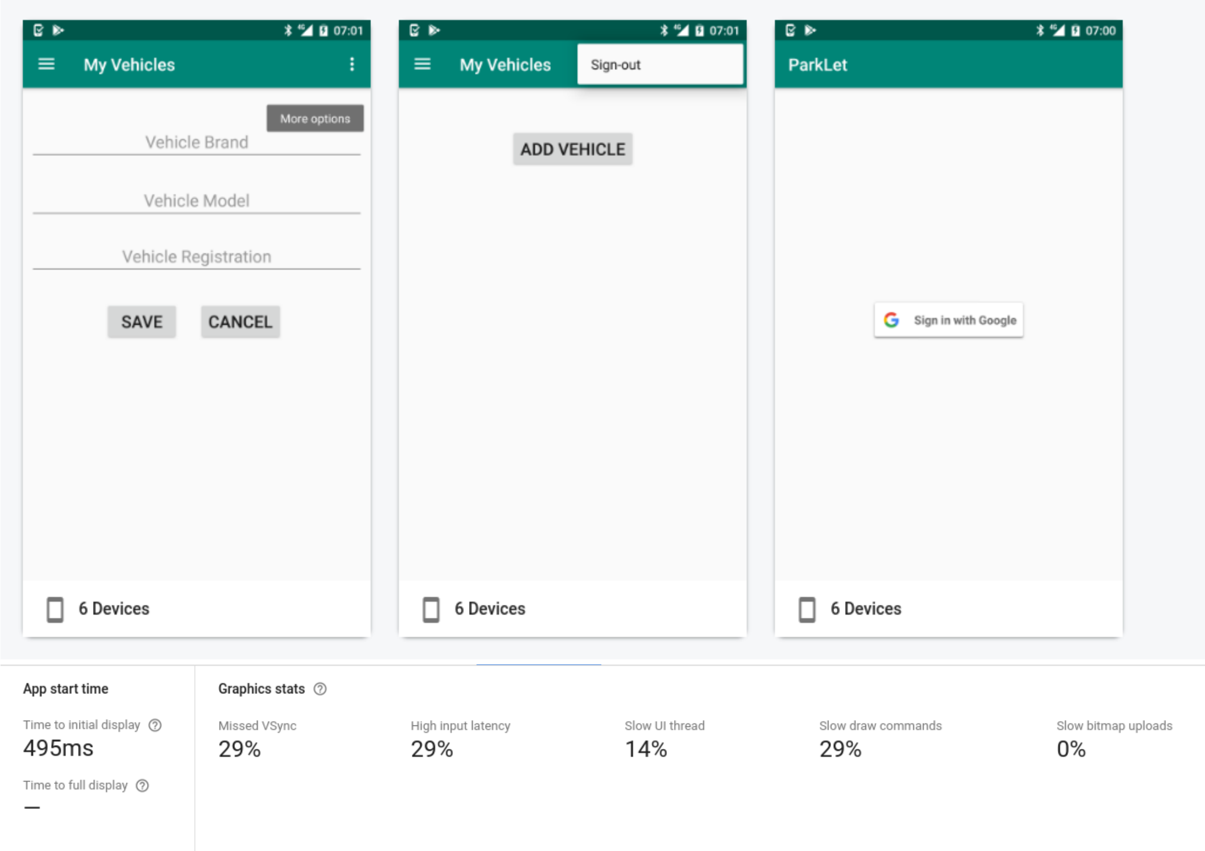
Below is an example of the Test Lab test runner script used during development. It uploads the latest version of the apps APK. The script details which screens I want transversed, the input for each text field given by its android resource id and which buttons must be clicked. The test runner will also attempt “Monkey” testing, in which it will rapidly click buttons and opposing actions in order to stress the app and instigate a crash. This also includes inserting random values into text fields or none at all.

We can then specify what devices we wish the app to run on, and what Android version.



```
gcloud beta firebase test android run \
  --type robo \
  --app /home/ark/Documents/fyp/2020-ca400-kinses38-purcem23/src/app/build/outputs/apk/debug/app-debug.apk \
  --robo-script ./parklet_robo_script.json \
  --device model=Pixel2,version=29,locale=en,orientation=portrait  \
  --device model=Pixel2,version=28,locale=en,orientation=portrait  \
  --device model=NexusLowRes,version=29,locale=en,orientation=portrait  \
  --device model=NexusLowRes,version=28,locale=en,orientation=portrait  \
  --timeout 5m \
  --auto-google-login
```

Upon completion of each test matrix, results are made available both in the terminal and the Firebase console portal. Any failed actions that I specified are instantly flagged for review, or even if a test was “flaky” but passed. Screenshot clusters are made available to see the difference between devices to identify where a specific device had difficulty.



App start time

Time to initial display
495ms

Time to full display
—

Graphics stats

Missed VSync
29%

High input latency
29%

Slow UI thread
14%

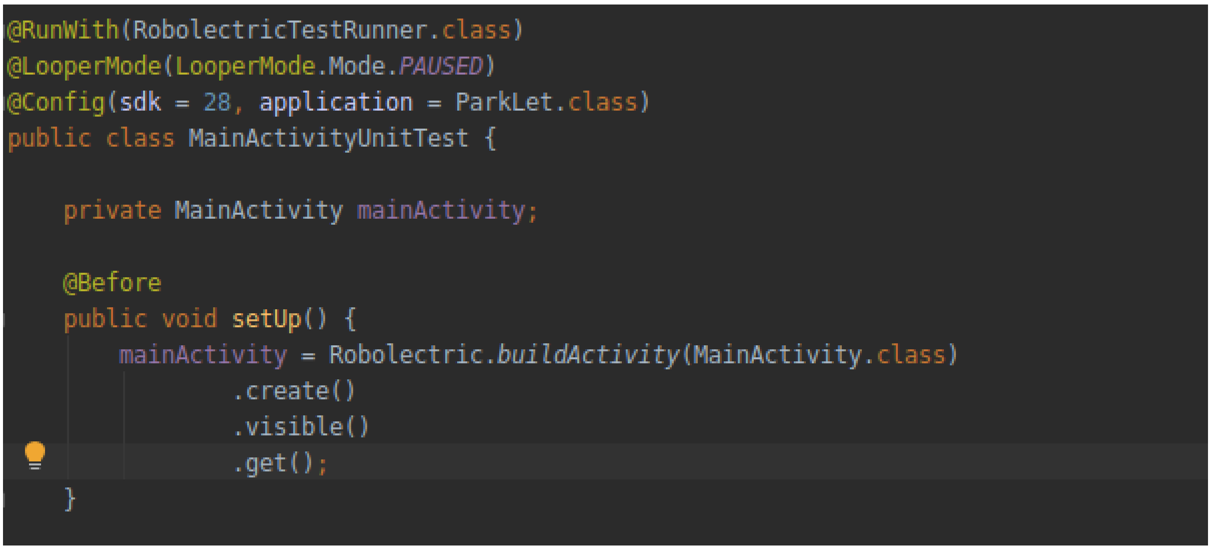
Slow draw commands
29%

Slow bitmap uploads
0%

Unit testing

Parklets unit tests were conducted using both Robolectric and Junit4 test runners, Robolectric for classes with Android framework dependencies such as the Main Activity, and junit4 for ViewModel classes.

Below is an example of an Activity being set up in isolation from other activities and fragments. The mainloop mode is set to paused to ensure that test cases do not fail due to Robolectric executing tasks earlier than they would on a real Android device. The android SDK being used to test is set to 28 as 29 requires Java 9 which is not compatible with Android Studios.

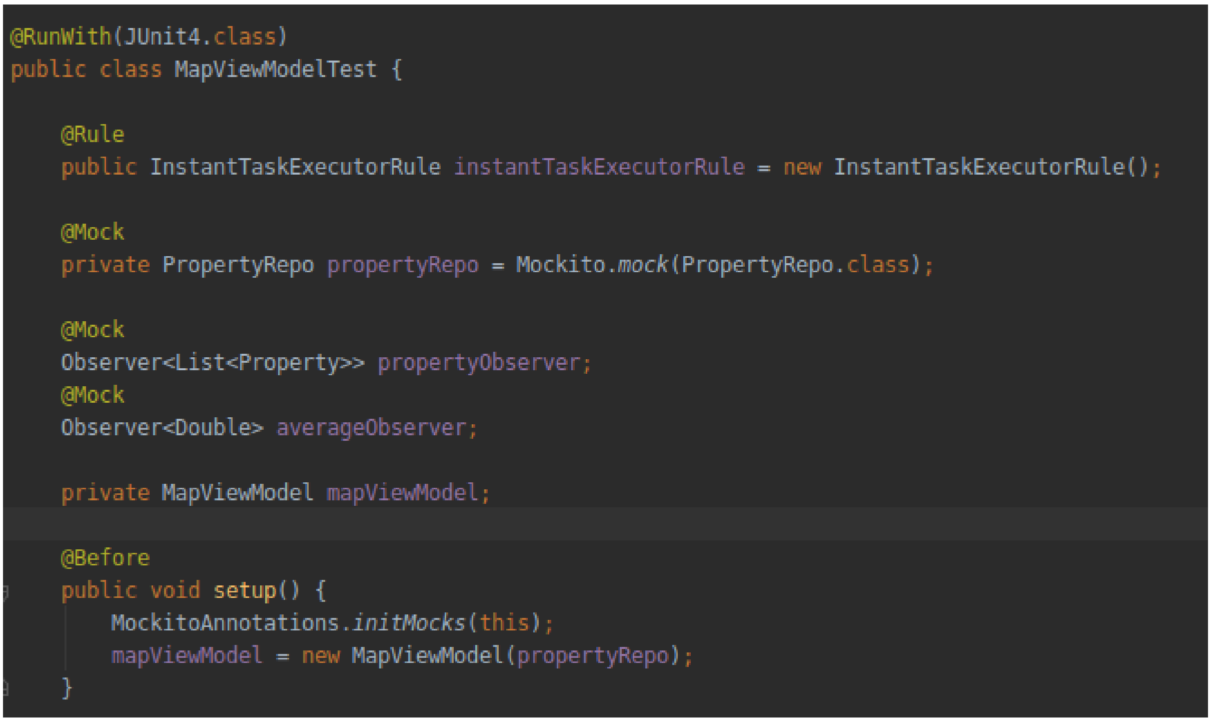


```
@RunWith(RobolectricTestRunner.class)
@LooperMode(LooperMode.Mode.PAUSED)
@Config(sdk = 28, application = ParkLet.class)
public class MainActivityUnitTest {

    private MainActivity mainActivity;

    @Before
    public void setUp() {
        mainActivity = Robolectric.buildActivity(MainActivity.class)
                .create()
                .visible()
                .get();
    }
```

For testing ViewModel classes in isolation, its required repository classes would have to be mocked. To enable this, the repositories would have to be provided as constructor arguments. However, Android's ViewModelProvider class only takes zero argument ViewModel constructors as arguments itself. To get around this, I was required to set up Dagger2 for the project to provide compile-time dependency injection.

Dagger2 requires both modules and components to be set up in order to explicitly tell it what the dependencies are, and which classes require them to be injected. Leveraging this, both the repositories and the custom ViewModel factory can be injected into classes requiring them.

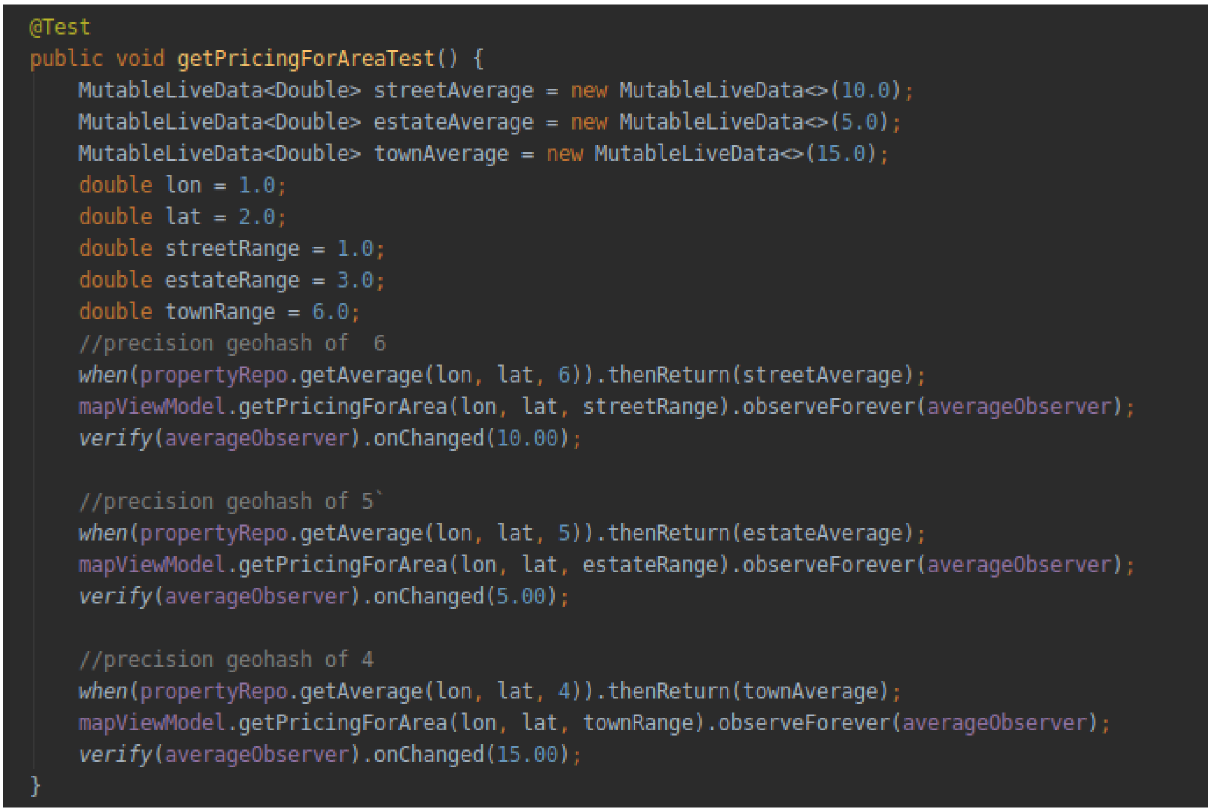
Below is an example setup for the MapViewModel class. An instant task executor rule is used to synchronously evaluate the methods being tested on the calling thread rather than a background one which would lead to test failure. The required property repository is mocked as are two observers which are required as LiveData objects will not emit any value until it is observed.



```
@RunWith(JUnit4.class)
public class MapViewModelTest {

    @Rule
    public InstantTaskExecutorRule instantTaskExecutorRule = new InstantTaskExecutorRule();

    @Mock
    private PropertyRepo propertyRepo = Mockito.mock(PropertyRepo.class);

    @Mock
    Observer<List<Property>> propertyObserver;
    @Mock
    Observer<Double> averageObserver;

    private MapViewModel mapViewModel;

    @Before
    public void setup() {
        MockitoAnnotations.initMocks(this);
        mapViewModel = new MapViewModel(propertyRepo);
    }
```

Below is an example of the MapViewModel method to map a user's range search to a particular geohash precision to fetch the average price from the correct GeoPriceBucket. The mocked repository obviously stands in for the real counterpart, while the mocked observer takes the role of the view/UI which observes changes in the ViewModel.

Mockito intercepts the repository method call “getPricingForArea” and returns a representation of how the LiveData would look in a real query. It is then verified that the correct value corresponding to the users required precision. The MVVM architecture in combination with dependency injection, enables unit tests like these to be run in complete isolation.



```
@Test
public void getPricingForAreaTest() {
    MutableLiveData<Double> streetAverage = new MutableLiveData<>(10.0);
    MutableLiveData<Double> estateAverage = new MutableLiveData<>(5.0);
    MutableLiveData<Double> townAverage = new MutableLiveData<>(15.0);
    double lon = 1.0;
    double lat = 2.0;
    double streetRange = 1.0;
    double estateRange = 3.0;
    double townRange = 6.0;
    //precision geohash of  6
    when(propertyRepo.getAverage(lon, lat, 6)).thenReturn(streetAverage);
    mapViewModel.getPricingForArea(lon, lat, streetRange).observeForever(averageObserver);
    verify(averageObserver).onChanged(10.00);

    //precision geohash of 5`
    when(propertyRepo.getAverage(lon, lat, 5)).thenReturn(estateAverage);
    mapViewModel.getPricingForArea(lon, lat, estateRange).observeForever(averageObserver);
    verify(averageObserver).onChanged(5.00);

    //precision geohash of 4
    when(propertyRepo.getAverage(lon, lat, 4)).thenReturn(townAverage);
    mapViewModel.getPricingForArea(lon, lat, townRange).observeForever(averageObserver);
    verify(averageObserver).onChanged(15.00);
}
```

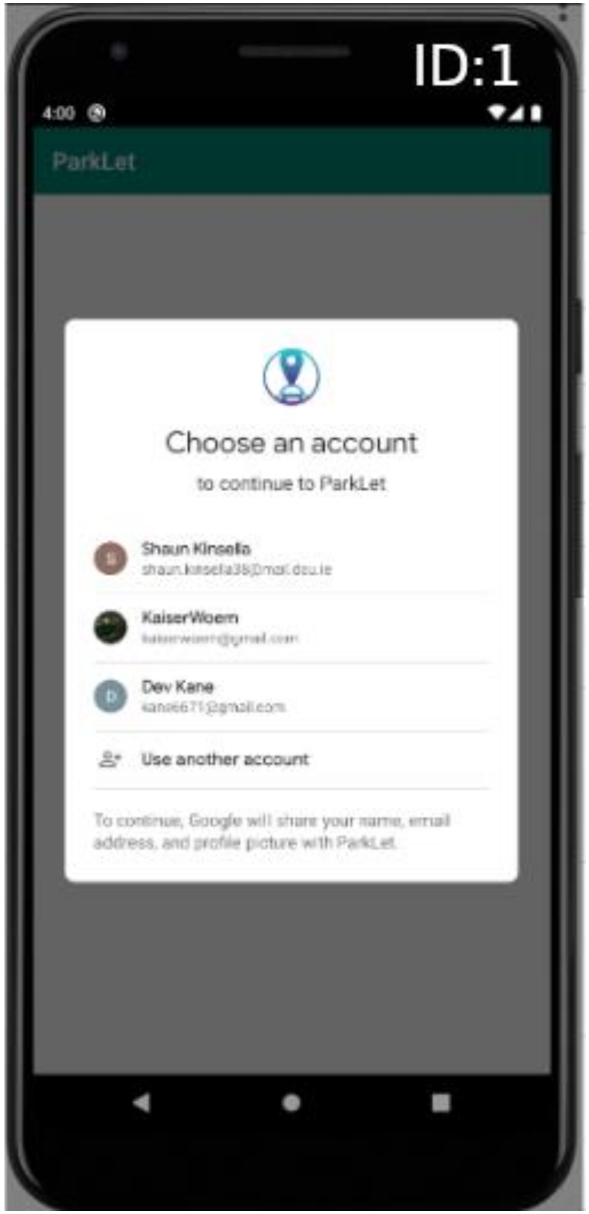
Use Case Scenario Testing

The following tests were performed on multiple emulated devices and physical One Plus 7 pro running android 9(API 28). Where appropriate, test scenarios are accompanied by screenshots of the resulting action with matching ID. All tests were performed by myself.

The purpose of these tests was to identify any potential issues from misuse of the app, common feature usage, and verifying a common user experience and UI across the app. Functionality such as notifications and NFC, where automated validation would be difficult were also tested and documented.

User Account Scenarios

ID	Test case	Expected:	Actual:	Outcome:
1	Create account with Gmail already added to phone	User brought to Home page with name welcoming them. User profile made on FB with FCM device token	As expected.	Pass
2	Create account with no email registered to phone	Asked to register email, then email displayed as choice for registering with Parklet, without having to reselect sign-in	As expected.	Pass
3	Create account, then sign-out and back in	Home page displayed, no duplicated data on fb	As expected.	Pass
4	Uninstall app/Switch phone	Home page displayed, no duplicated data on fb. FCM device token updated	As expected.	Pass



Choose an account

to continue to ParkLet

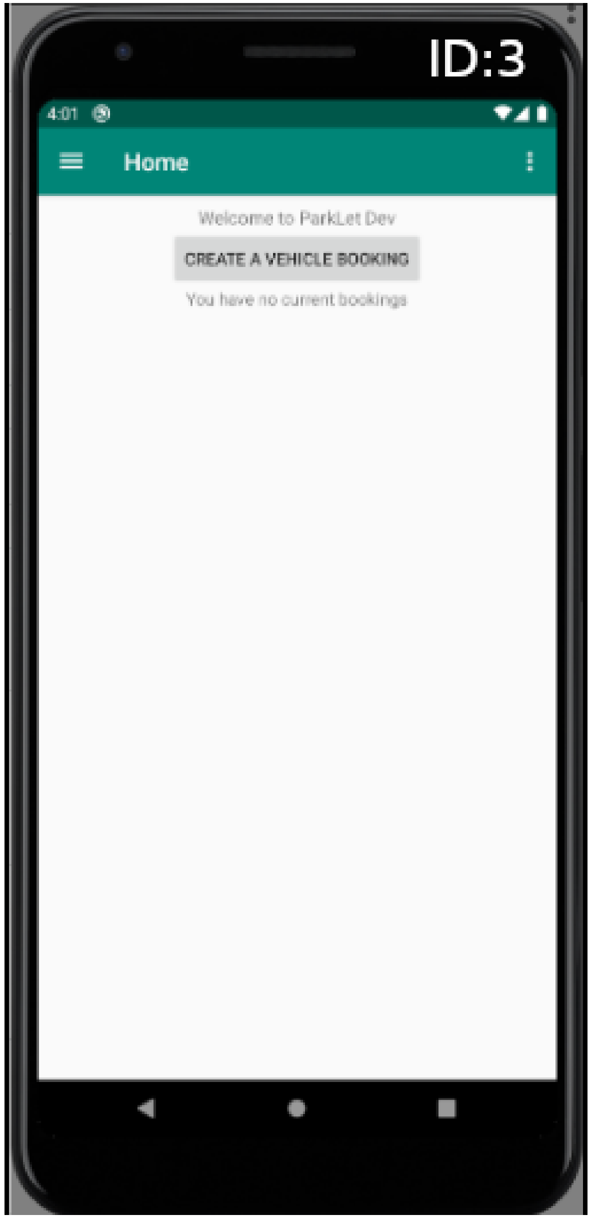
Shaun Kinsella
shaun.kinsella38@mail.dcu.ie

KaiserWoern
kaiserwoern@gmail.com

Dev Kane
kane6671@gmail.com

Use another account

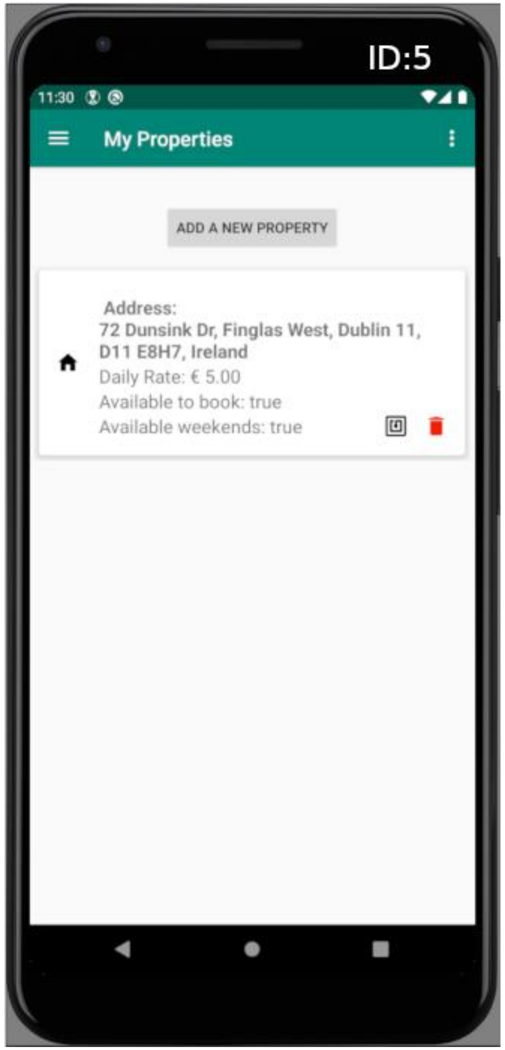
To continue, Google will share your name, email address, and profile picture with ParkLet.



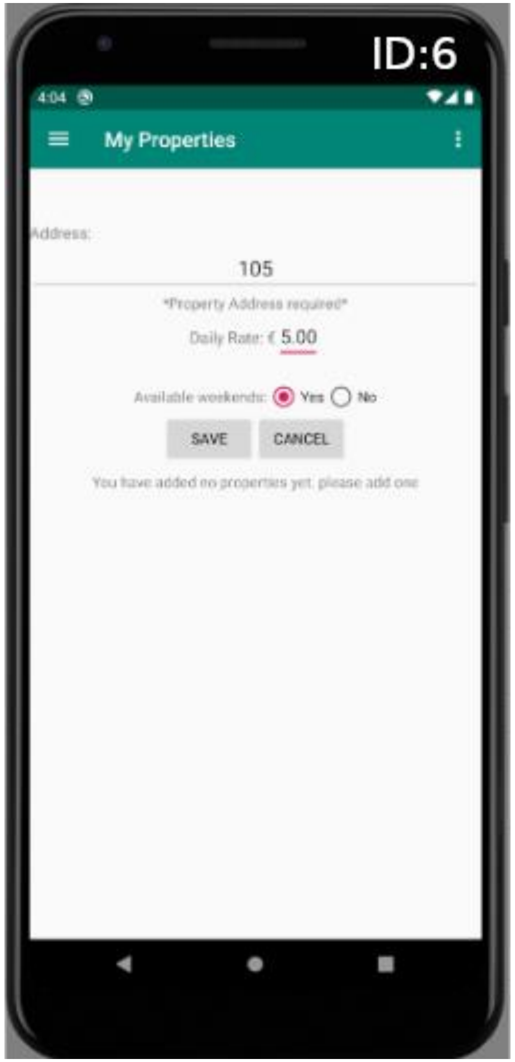
Welcome to ParkLet Dev

You have no current bookings

User Property Scenarios				
ID	**Test case**	**Expected:**	**Actual:**	**Outcome:**
5	User adds new property, inputs valid address, price and weekend availability	Form closes, toast notification showing success appears. Property list is updated with new property	As expected.	Pass
6	User adds new property, inputs invalid address. Valid price and weekend availability	Toast message "Cannot find address displayed". Clicking save will show an error text under the address line to notify address is required.	As expected.	Pass
7	User adds new property, inputs invalid address, invalid price and weekend availability	Clicking save will display two text errors, prompting user to enter valid address and price	As expected.	Pass
8	User inputs valid but incorrect address, i.e. neighbours house. Then corrects	Address will update correctly and success notification displayed. Properties will update	Address will only update when user loses focus of the address input box. Otherwise old address will be posted to firebase.	Pass
9	User enters values into property form, clicks cancel, then attempts to enter new property	Form will be reset of all values	As expected.	Pass
10	User Deletes Property	Property removed from recyclerview and toast message displayed confirming deletion of property. Firebase updated and triggers removal of property from GeoBucket. User properties shows that they currently have no properties	As expected. Should use a confirmation dialog for this action	Pass
11	User selects one property in RecyclerView then deletes second property	Only the correct property is removed. Remaining properties still displayed	As expected.	Pass
12	User attempts to add property with same Eircode as already existing house, owned by other users or themselves	Form closes, toast notification informs user that the property already exists	As expected.	Pass


ID:5

Address:
72 Dunsink Dr, Finglas West, Dublin 11, D11 E8H7, Ireland
Daily Rate: € 5.00
Available to book: true
Available weekends: true


ID:6

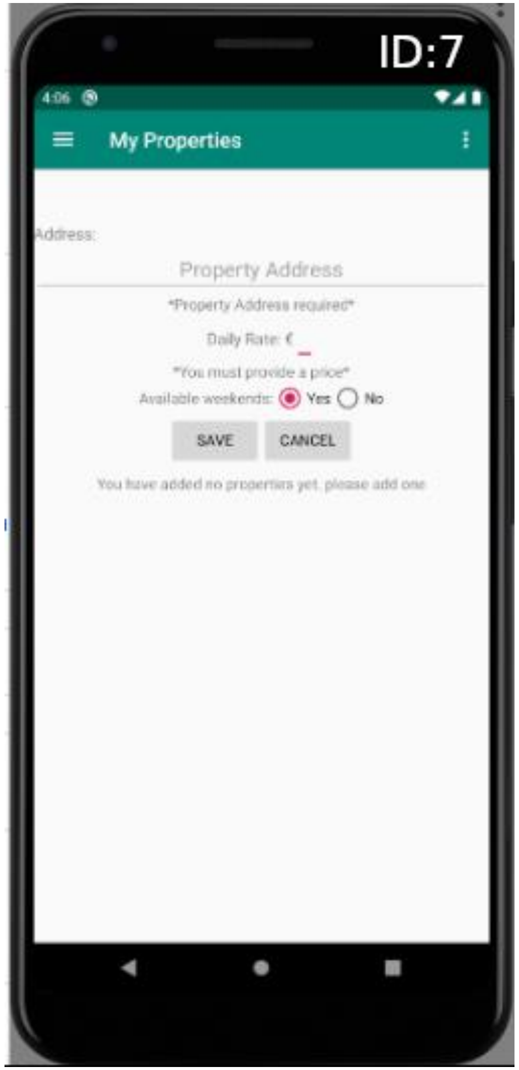
Address:
105

Property Address required

Daily Rate: € 5.00

Available weekends: ● Yes ○ No

You have added no properties yet, please add one


ID:7

Address:
Property Address

Property Address required

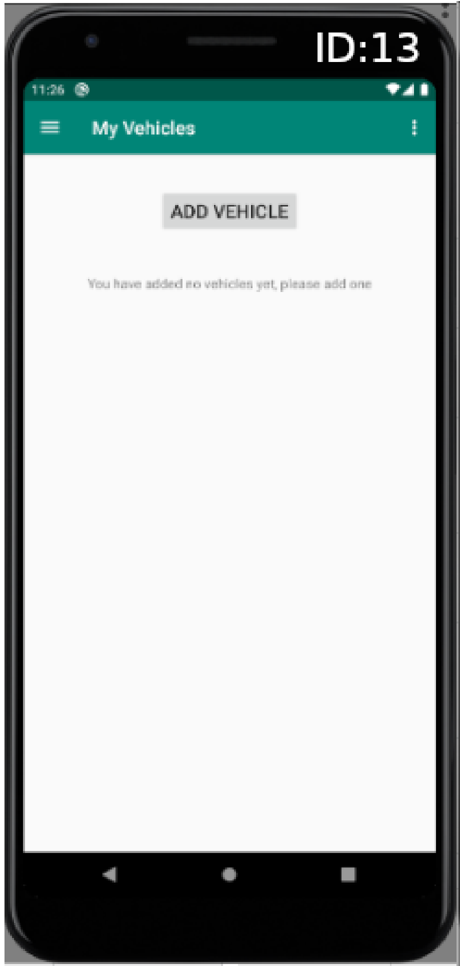
Daily Rate: €

You must provide a price

Available weekends: ● Yes ○ No

You have added no properties yet, please add one

User Vehicle Scenarios				
ID	**Test case**	**Expected:**	**Actual:**	**Outcome:**
13	User visits vehicle page for first time	Add vehicle button displayed, Textview informs user they have no vehicles and to add. RecyclerView hidden	As expected.	Pass
14	User adds new vehicle, inputs valid make, model and reg. Clicks save	Form is reset and hidden. Toast notification appears informing user of success. Add vehicle text is replaced with recyclerview displaying the vehicles info	As expected.	Pass
15	User adds new vehicle, inputs invalid make, valid model and reg. Clicks save	User is informed that the make is required through toast notification.	As expected. However form resetting and closing does not present good UX	Pass
16	User adds new vehicle, inputs invalid model, valid make and reg. Clicks save	User is informed that the model is required through toast notification.	As expected. However form resetting and closing does not present good UX	Pass
17	User adds new vehicle, inputs invalid reg, valid make and model. Clicks save	User is informed that the reg is required through toast notification.	As expected. However form resetting and closing does not present good UX	Pass
18	User attempts to add vehicle with all fields blank	User is informed that all fields are mandatory through toast notification	As expected however form resetting and closing does not present good UX	Pass
19	User attempts to add an already existing vehicle with same reg	User is informed that the vehicle with that reg already exists through toast notification. Form closes	As expected.	Pass
20	User attempts to add vehicle with unique reg but duplicate make and model	Vehicle is successfully added	As expected.	Pass
21	User deletes singular vehicle	User is informed of successful deletion. Recyclerview updates and removes said vehicle	As expected.	Pass13
22	User selects a vehicle in the recyclerview, then deletes a different vehicle	The correct vehicle is deleted and the user is informed of success	As expected.	Pass
23	User enters values into vehicle form, clicks cancel, then attempts to enter new property	Form will be reset of all values	As expected.	Pass

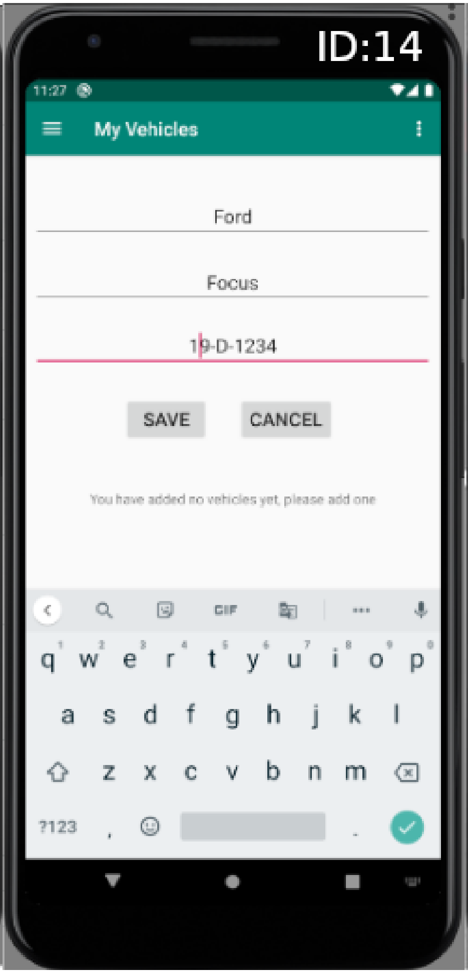

ID:13

11:26

My Vehicles

ADD VEHICLE

You have added no vehicles yet, please add one


ID:14

11:27

My Vehicles

Ford

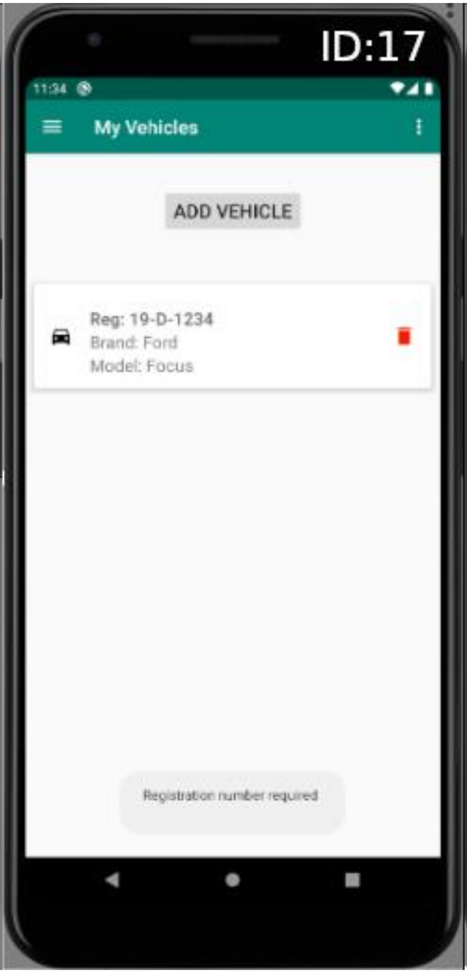
Focus

19-D-1234

SAVE

CANCEL

You have added no vehicles yet, please add one


ID:17

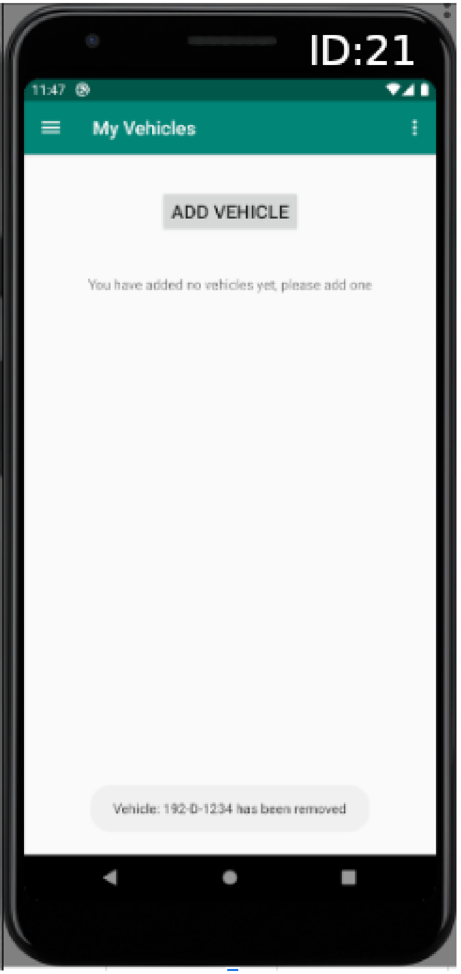
11:34

My Vehicles

ADD VEHICLE

Reg: 19-D-1234
Brand: Ford
Model: Focus

Registration number required


ID:21

11:47

My Vehicles

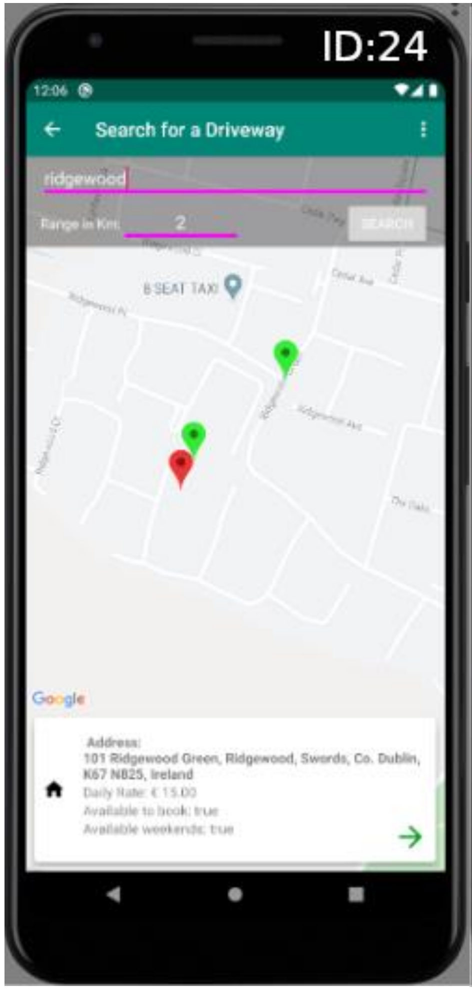
ADD VEHICLE

You have added no vehicles yet, please add one

Vehicle: 192-D-1234 has been removed

User Map scenarios				
ID	**Test case**	**Expected:**	**Actual:**	**Outcome:**
24	User searches for available driveways in area with multiple properties	Map populated with markers colour coded by the properties price in comparison to area. Map camera zooms to area encompassing markers so all are displayed. Map RecyclerView populated with same properties	As expected.	Pass
25	User searches for available driveways in area with **one** property	Map populated with markers colour coded by the properties price in comparison to area. Map camera zooms to area encompassing markers so all are displayed. Map RecyclerView populated with same properties	As expected. Possibly relax the zoom level to show more of the area	Pass
26	User searches for available driveways in area with **no** properties	Map zooms to search area, user is informed that there are no properties in area and to try expanding their search area	As expected.	Pass
27	Area with properties but small range	Some properties that fall into the range of the area of searches perimeter will show.	As expected. User has to increase the range of search to show any other properties. They can either be more specific in area of search or increase the range	Pass
28	Area with properties but range larger than perimeter of search area	Properties from other areas will "bleed" into the users search area. EG: Searching Ballymun with range of 10km will show some properties from Finglas/Swords	As expected.	Pass
29	User searches for property in area while another user adds new property to that area	Map markers updated, camera repositions to encompass all new markers, average price comparison updates to reflect new GeoPriceBucket average	As expected. User will lose focus of selected property however. Store last selected properties id?	Pass
30	User searches for property in area, while another user adds property **outside** of that area	Nothing. Should have no impact on the user's screen/experience as the original search query is the only thing being observed for updates	As expected. No map flickering	Pass
31	User searches for property in area, while another user deletes a property in that area	Map markers updated, camera repositions to encompass remaining markers, average price comparison updates to reflect new GeoPriceBucket average	As expected. Slight delay on average price comparison recalculation but out of control due to cloud trigger. No action required	Pass

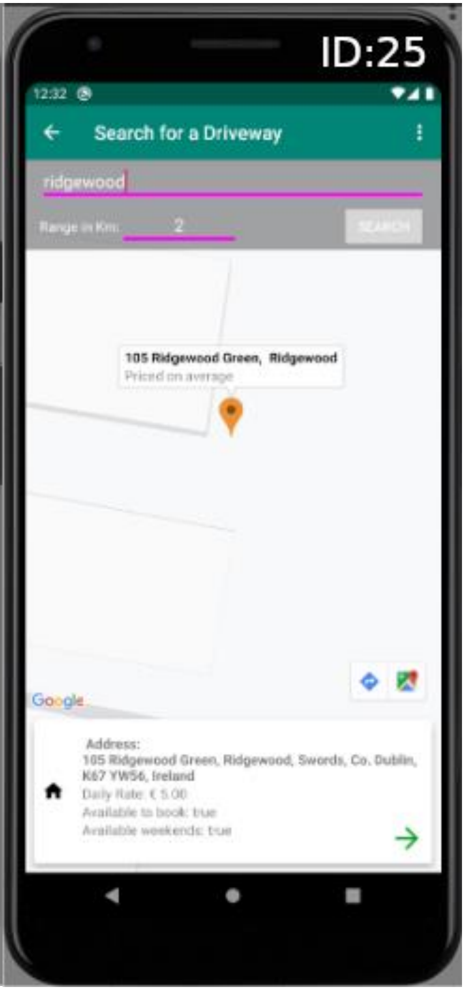
32	User browses through map markers, then selects property to book through map recyclerview	The selection of map markers causes the recyclerview to automatically scroll to show a summary of the property. When a property has the green arrow clicked the user is presented with booking calendar for appropriate property	As expected.	Pass
33	User searches specific property address	Map updates and zooms to show selected property	As expected. Other properties that fall within the users specified range will also show but the searched property will be centred	Pass
34	User searches for properties with blank address input	Map defaults camera to Ireland and shows properties within the specified range	As expected. For better UX experience should zoom to users current area	Pass
35	User searches for properties with blank range input	Inform user that a range is required	App crashes. Number format exception with empty string. Validation or default value required	Fail
36	User selects map marker then clicks directions widget button. Then backpresses back to parklet app	Google maps directions to property are displayed from users' current position. Back press results in being brought back to parklet app on map view	As expected. the property is still selected	Pass
37	User browses through map recyclerview, half scrolls a property itemview then selects a property to book	When user releases the finger press, the recyclerview snaps to the nearest horizontal property item, then when pressing the go to bookings icon the correct property is loaded	As expected.	Pass


ID:24

ridgewood

2

Address:
101 Ridgewood Green, Ridgewood, Swords, Co. Dublin, K67 N825, Ireland
Daily Rate: € 15.00
Available to book: true
Available weekends: true

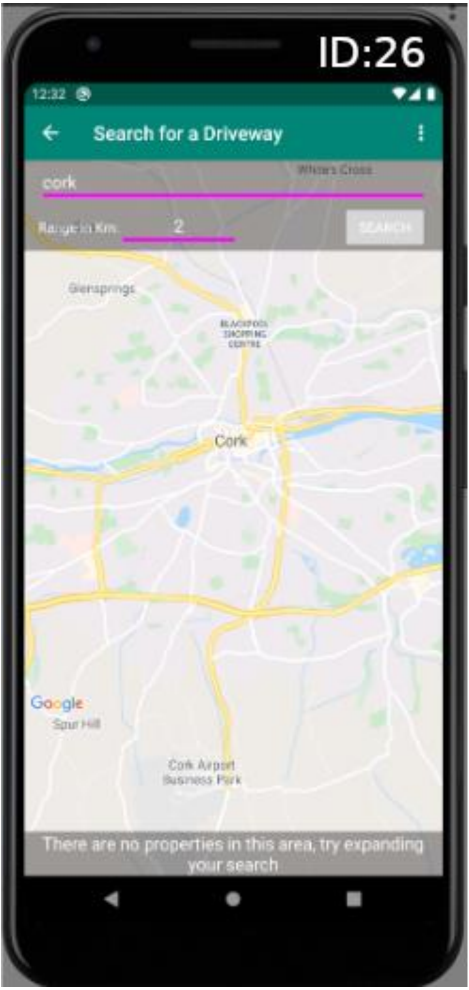

ID:25

ridgewood

2

105 Ridgewood Green, Ridgewood
Priced on average

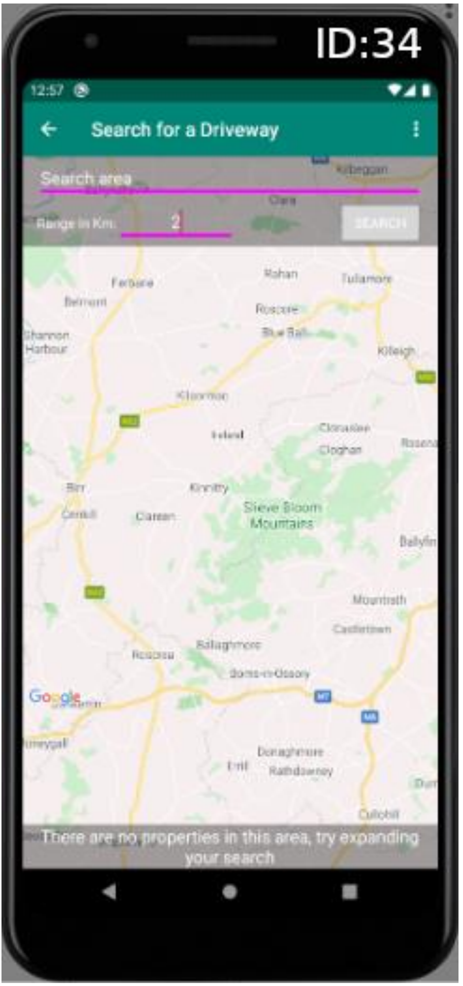
Address:
105 Ridgewood Green, Ridgewood, Swords, Co. Dublin, K67 YW56, Ireland
Daily Rate: € 5.00
Available to book: true
Available weekends: true


ID:26

cork

2

There are no properties in this area, try expanding your search

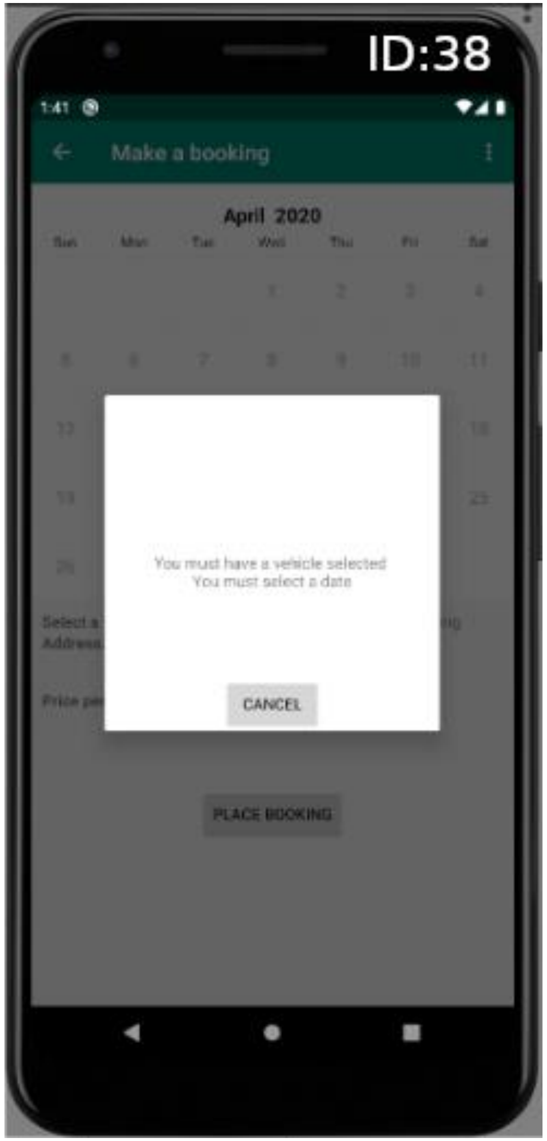

ID:34

2

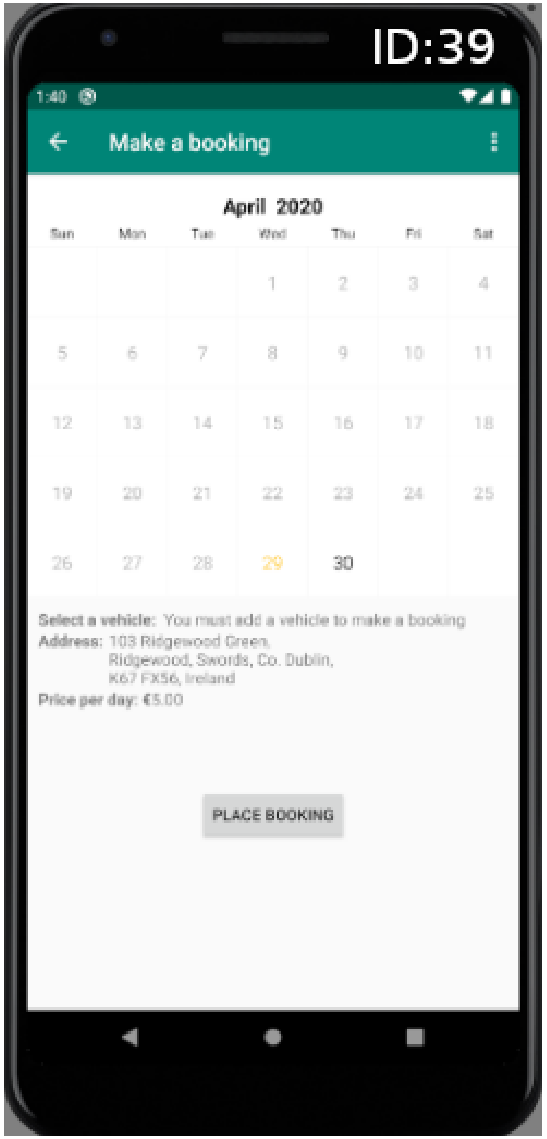
There are no properties in this area, try expanding your search

User Booking scenarios				
ID	**Test case**	**Expected:**	**Actual:**	**Outcome:**
38	User attempts to book with no date or vehicle selected	User has warning displayed under "select vehicle" field. Upon pressing the place booking button a dialog informs them they must add a vehicle and select dates to book	As expected. Better UX would allow them to add a vehicle there rather than navigating back however	Pass
39	User attempts to book with no vehicle selected	User has warning displayed under "select vehicle" field. Upon pressing the place booking button a dialog informs them they must add a vehicle.	As expected. Better UX would allow them to add a vehicle there rather than navigating back however	Pass
40	User makes valid singular day booking	User presses place booking and is shown a dialog confirming the details of their booking. The total price, and days booked. Upon confirming and the booking created, the user is presented with the success state and brought back to home page, showing the new booking	As expected.	Pass
41	User attempts to select day already booked	Day in question is non-selectable	As expected.	Pass
42	User selects day to book but another user books before	Day in question is de-selected and when the user attempts to confirm booking they are notified the day is not available	As expected.	Pass
43	User observes available days while another user cancels a booking	The cancelled day is then released to be booked	As expected. Does not affect current selection	Pass
44	User selects day and vehicle to book but cancels	App returns to CalendarView	As expected. but days are deselected. Should retain days selected if its a case the user wishes to amend	Pass
45	User attempts to select day before current days date	UI prevents selection	As expected. Could be circumvented by manually setting the date of phone. Valid booking dates enforced by firebase rules	Pass
46	User books consecutive days	Booking info correctly shown, price correctly calculated	As expected.	Pass
47	User books non-consecutive days	Booking info correctly shown, price correctly calculated	As expected.	Pass
48	User attempts to scroll to previous month	UI prevents selection	As expected.	Pass

49	User to months in advance	Calendar prevents user from scrolling more than 2 months in advance. Customisable in ParkLet CalendarView	Actually, calculates as 2 months' worth of days rather than 2 Calendar months. However, this makes more sense	Pass
50	User selects booking dates but then suspends app and brings back to foreground	App successfully resumes	As expected, dates still selected	Pass
51	User makes successful booking then cancels and attempts to rebook same date	Allowed to rebook.	As expected, in terms of business, it could be blocked to stop abuse. But considering payment up front is intended that might be self-moderating. This is set in booking ViewModel	Pass



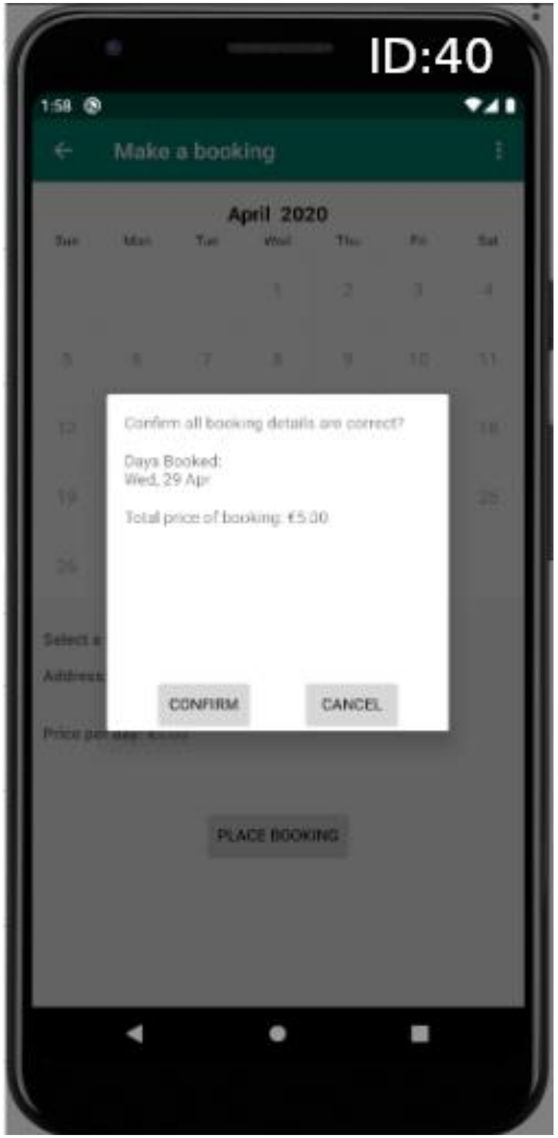
You must have a vehicle selected
You must select a date



April 2020

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

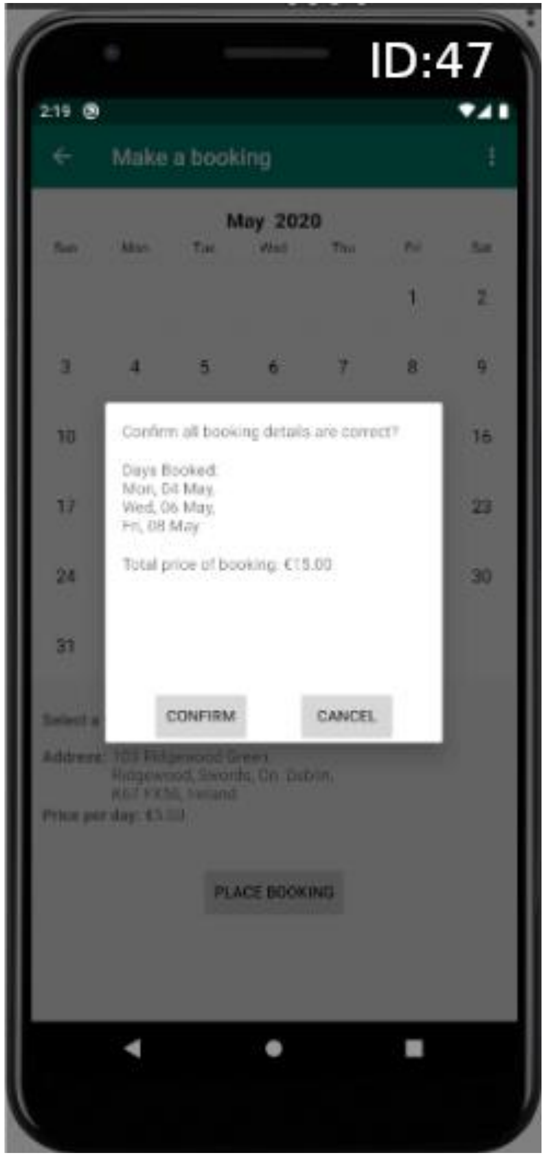
Select a vehicle: You must add a vehicle to make a booking
Address: 103 Ridgewood Green,
Ridgewood, Swords, Co. Dublin,
K67 FX56, Ireland
Price per day: €5.00



Confirm all booking details are correct?

Days Booked:
Wed, 29 Apr

Total price of booking: €5.00



ID:47

Make a booking

May 2020

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10						16
17						23
24						30
31						

Confirm all booking details are correct?

Days Booked:
Mon, 04 May,
Wed, 06 May,
Fri, 08 May

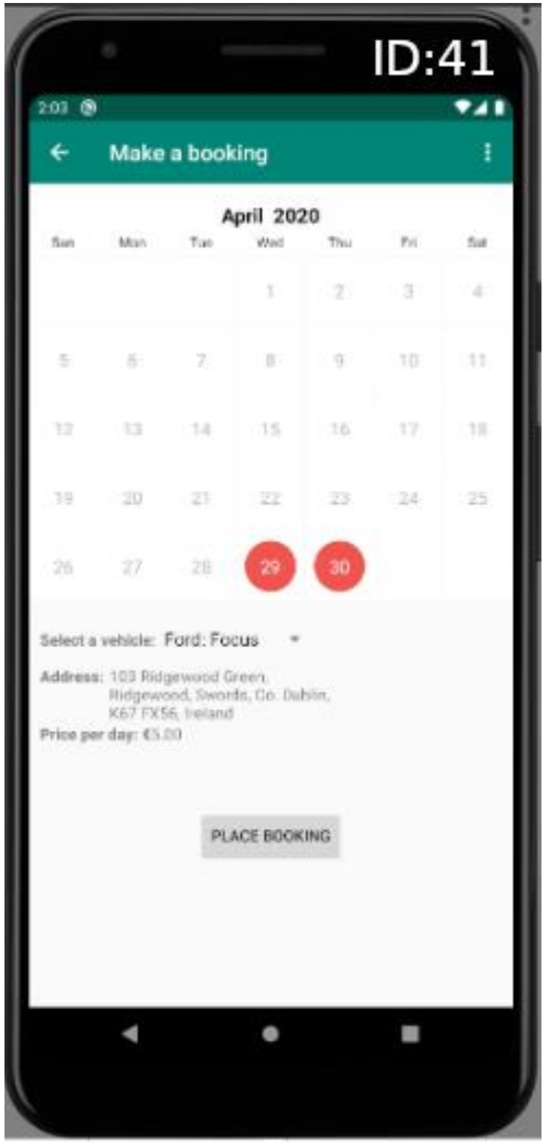
Total price of booking: €15.00

CONFIRM CANCEL

Select a

Address: 103 Ridgewood Green
Ridgewood, Swords, Co. Dublin,
K67 FX56, Ireland
Price per day: €5.00

PLACE BOOKING



ID:41

Make a booking

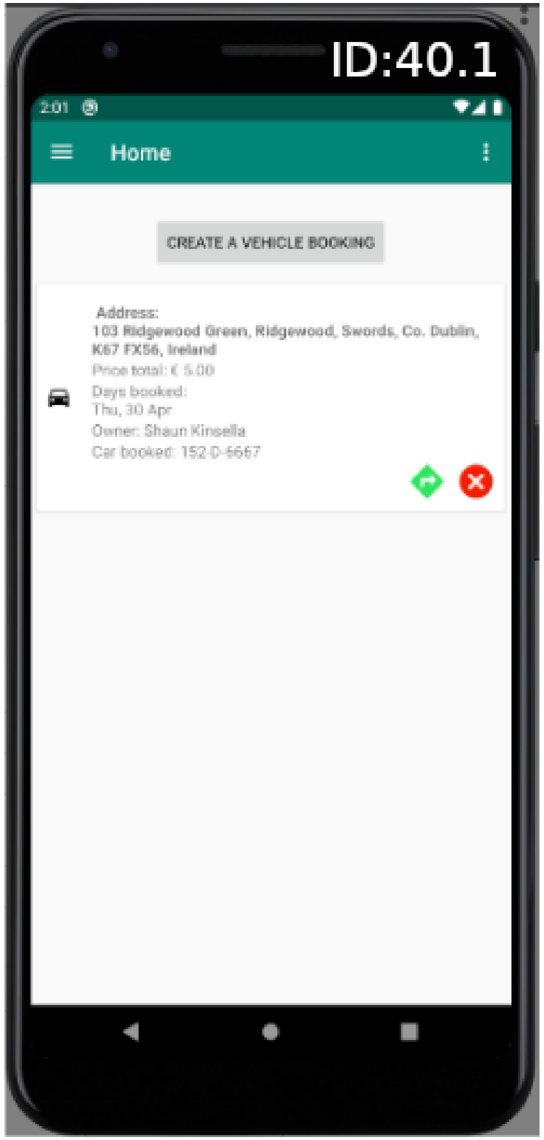
April 2020

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Select a vehicle: Ford: Focus ▾

Address: 103 Ridgewood Green,
Ridgewood, Swords, Co. Dublin,
K67 FX56, Ireland
Price per day: €5.00

PLACE BOOKING



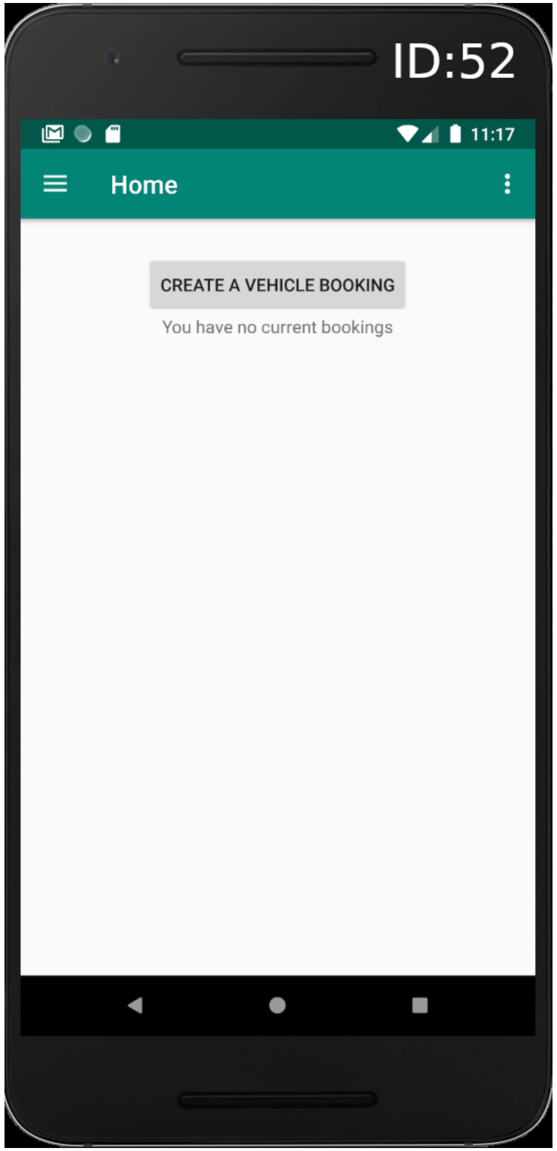
ID:40.1

Home

CREATE A VEHICLE BOOKING

Address:
103 Ridgewood Green, Ridgewood, Swords, Co. Dublin,
K67 FX56, Ireland
Price total: € 5.00
Days booked:
Thu, 30 Apr
Owner: Shaun Kinsella
Car booked: 152-D-6667

User Home Scenarios				
ID	**Test case**	**Expected:**	**Actual:**	**Outcome:**
52	User navigates to home screen with no bookings	Presented with option to create a vehicle booking, and a TextView informing them that they have no current bookings	As expected.	Pass
53	User navigates to home screen with multiple bookings for their property and car	Presented with option to create vehicle booking and a list of all their bookings, and cancellation options for each booking. Each booking showing relevant details	As expected, for vehicle bookings, a car icon is used, directions are available and the owners name is listed. For property bookings a house icon is shown and the renters name	Pass
54	User cancels their vehicle's booking	Booking item is updated to show cancellation status and who cancelled it. Cancellation button becomes inactive for both owner and renter. Owner receives notification through FCM that it was cancelled	As expected.	Pass
55	User cancels their property's booking	Booking item is updated to show cancellation status and who cancelled it. Cancellation button becomes inactive for both owner and renter. Renter receives notification through FCM that it was cancelled	As expected.	Pass
56	User uses directions to property for booking	User is redirected to google maps directions between their current location and property. Back pressing brings them to home page app once more	As expected.	Pass
57	User observes bookings while another user books their property	Recyclerview correctly updates with new booking	As expected, new booking placed at bottom of recyclerview	Pass

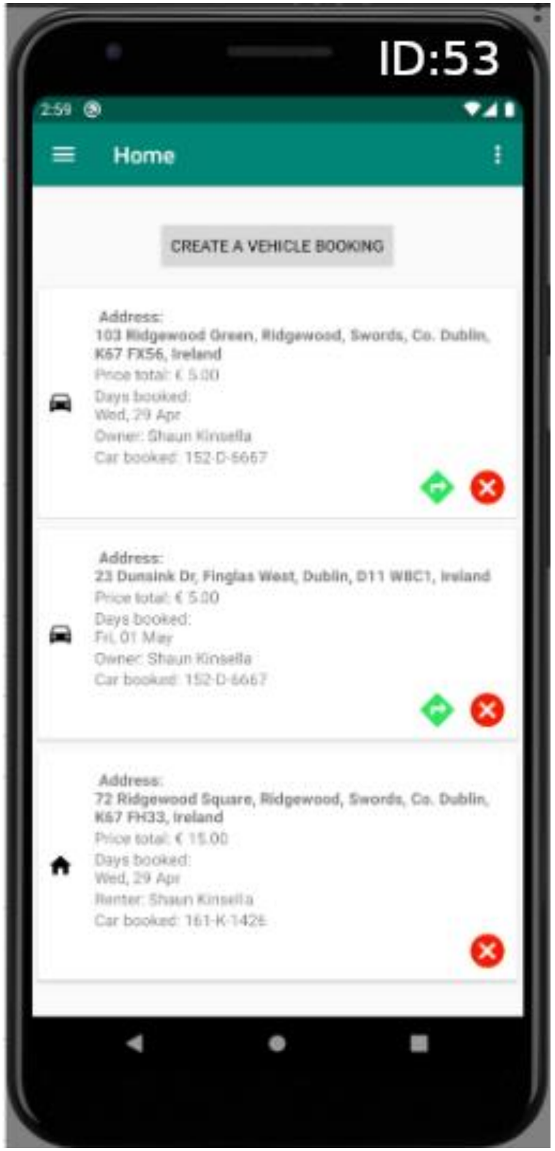

ID:52

11:17

Home

CREATE A VEHICLE BOOKING

You have no current bookings


ID:53

2:59

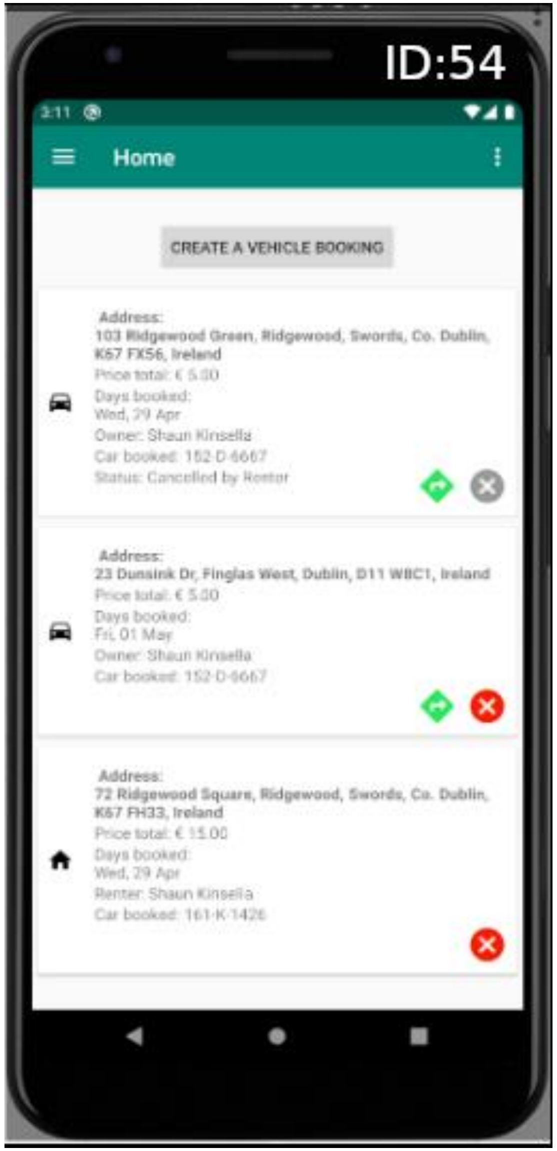
Home

CREATE A VEHICLE BOOKING

Address:
103 Ridgewood Green, Ridgewood, Swords, Co. Dublin, K67 FX56, Ireland
Price total: € 5.00
Days booked:
Wed, 29 Apr
Owner: Shaun Kinsella
Car booked: 152-D-6667

Address:
23 Dunsink Dr, Finglas West, Dublin, D11 WBC1, Ireland
Price total: € 5.00
Days booked:
Fri, 01 May
Owner: Shaun Kinsella
Car booked: 152-D-6667

Address:
72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland
Price total: € 15.00
Days booked:
Wed, 29 Apr
Renter: Shaun Kinsella
Car booked: 161-K-1426


ID:54

3:11

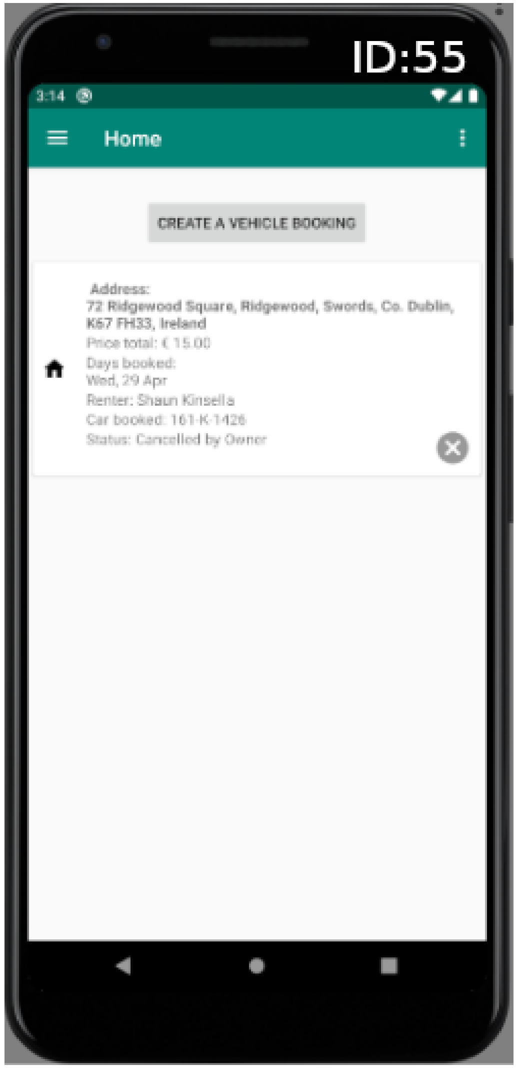
Home

CREATE A VEHICLE BOOKING

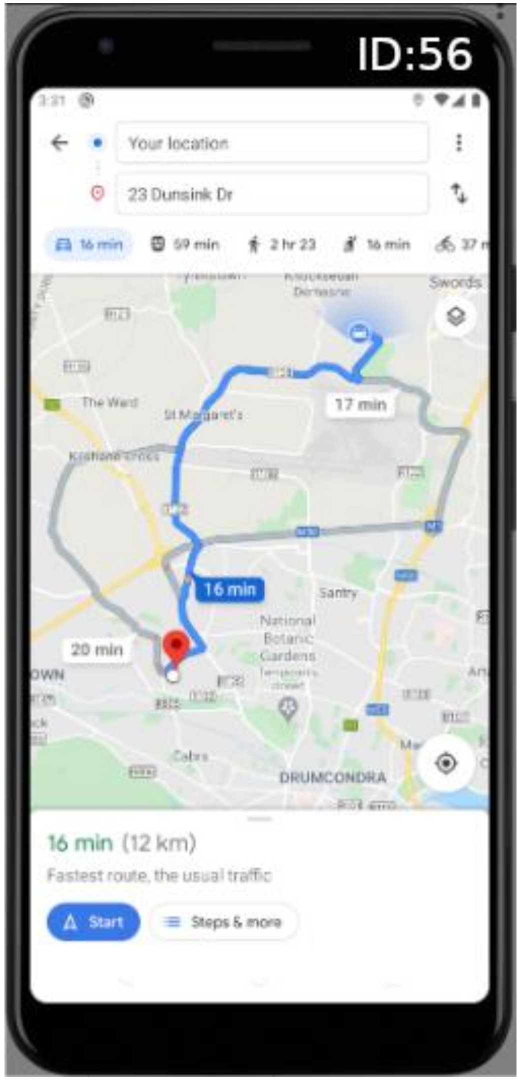
Address:
103 Ridgewood Green, Ridgewood, Swords, Co. Dublin, K67 FX56, Ireland
Price total: € 5.00
Days booked:
Wed, 29 Apr
Owner: Shaun Kinsella
Car booked: 152-D-6667
Status: Cancelled by Rentor

Address:
23 Dunsink Dr, Finglas West, Dublin, D11 WBC1, Ireland
Price total: € 5.00
Days booked:
Fri, 01 May
Owner: Shaun Kinsella
Car booked: 152-D-6667

Address:
72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland
Price total: € 15.00
Days booked:
Wed, 29 Apr
Renter: Shaun Kinsella
Car booked: 161-K-1426

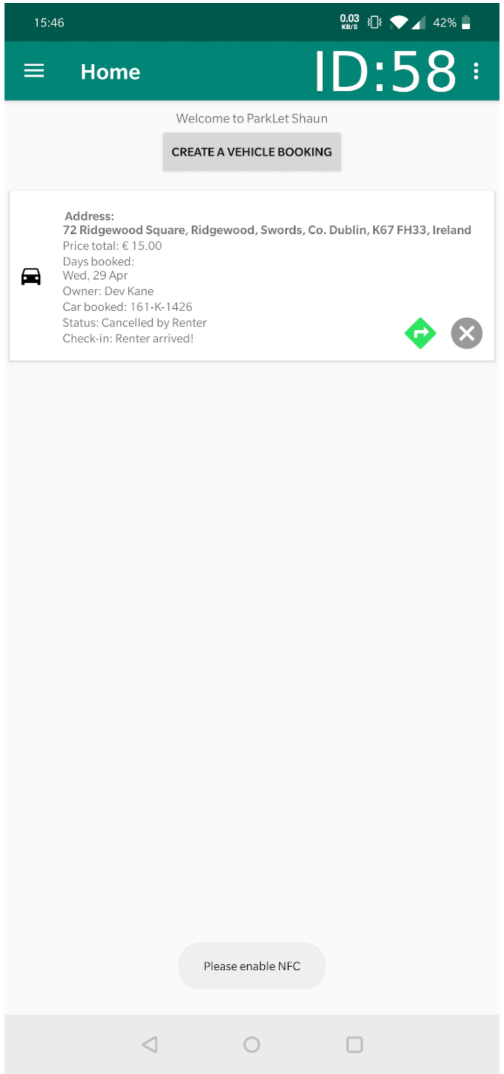

ID:55

Address:
72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland
Price total: € 15.00
Days booked:
Wed, 29 Apr
Renter: Shaun Kinsella
Car booked: 161-K-1426
Status: Cancelled by Owner


ID:56

NFC scenarios				
ID	**Test case**	**Expected:**	**Actual:**	**Outcome:**
58	User starts app with NFC disabled with phone with NFC capabilities	User informed to enable NFC	As expected.	Pass
59	User starts app with NFC disabled with phone with NFC capabilities	User informed that NFC is required. App should not crash	As expected. Alternative check-in required. QR reader	Pass
60	User taps blank NFC tag to phone with app not started	Nothing within the app	Android launches default NFC tag activity selector	Pass
61	User taps blank NFC tag with app in foreground	Booking not found displayed as dialog fragment	As expected, but realistically should be filtered if the tag is blank	Pass
62	User taps ParkLet written tag against phone with ParkLet Installed and app in background	App launched to home screen. If user is not signed in, they are brought to sign in page	As expected.	Pass
63	User taps ParkLet written tag against phone with ParkLet Installed and app in foreground	Booking not found displayed as dialog fragment if there exists no booking for that property on that date	As expected. Possibly filter it if property belongs to user but that would require a query	Pass
64	User taps ParkLet written tag against phone with ParkLet not installed	Users is directed to imaginary ParkLet app on Google Play store	As expected.	Pass
65	User attempts to write Tag for property correctly holding the tag	User is informed that the tag was written correctly.	As expected. Confirms the property's address	Pass
66	User attempts to write Tag for property, moving the tag away too quickly	User is informed the tag writing failed and is free to try again while the dialog is still present	As expected.	Pass
67	User attempts to write tag for property with a locked NFC tag	User is informed the tag writing failed and is free to try again while the dialog is still present	As expected. Should have its own warning for locked tags however	Pass
68	User attempts to scan non-parklet written tag with app in foreground	"No booking found" displayed	App crashes, tries to parse non-existent records	Fail

69	User attempts to scan non-parklet written tag with app in background	Nothing within the app	As expected.	Pass
70	Renter taps correctly written NFC tag with a valid booking for that day **and** property	User is informed that they are correctly checked in	As expected.	Pass
71	Renter taps correctly written NFC tag with a valid booking for that day but **not** the correct property	"No booking found" displayed	As expected.	Pass
72	Renter taps correctly written NFC tag with the correct property but no booking for that day	"No booking found" displayed	As expected.	Pass
73	Renter attempts to check-in	Check-in is confirmed for the property. Status is updated on the homepage for that property and the owner is notified	As expected.	Pass
74	Renter attempts to checkout	Check-out is confirmed for the property. Status is updated on the homepage for that property and the owner is notified	As expected.	Pass
75	Property owner attempts to rewrite previously written ParkLet tag in app	Tag successfully written	As expected.	Pass
76	Renter attempts to check-in at property with a cancelled booking	Informed booking is no longer valid. No firebase update or notification for owner	As expected.	Pass

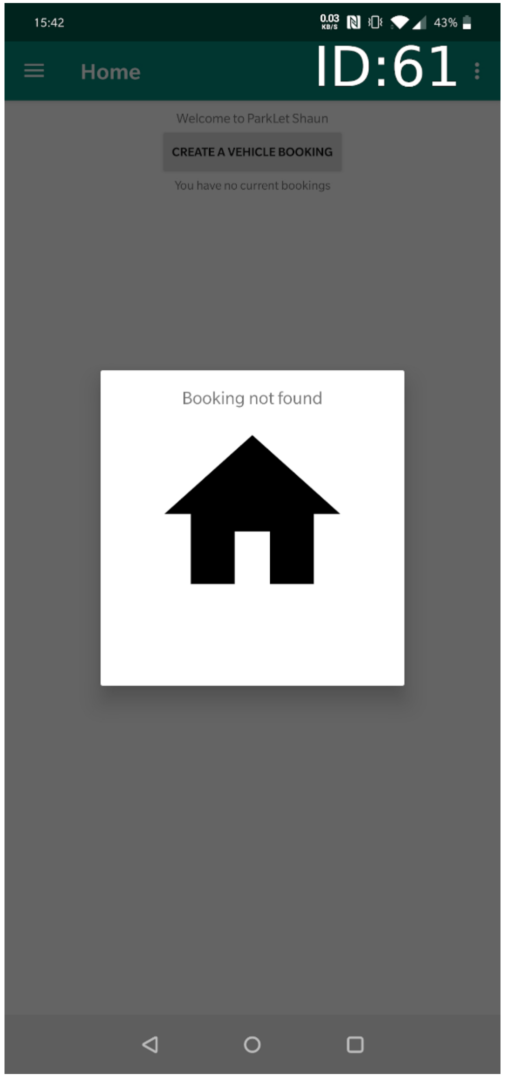


Welcome to ParkLet Shaun

CREATE A VEHICLE BOOKING

Address:
72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland
Price total: € 15.00
Days booked:
Wed, 29 Apr
Owner: Dev Kane
Car booked: 161-K-1426
Status: Cancelled by Renter
Check-in: Renter arrived!

Please enable NFC

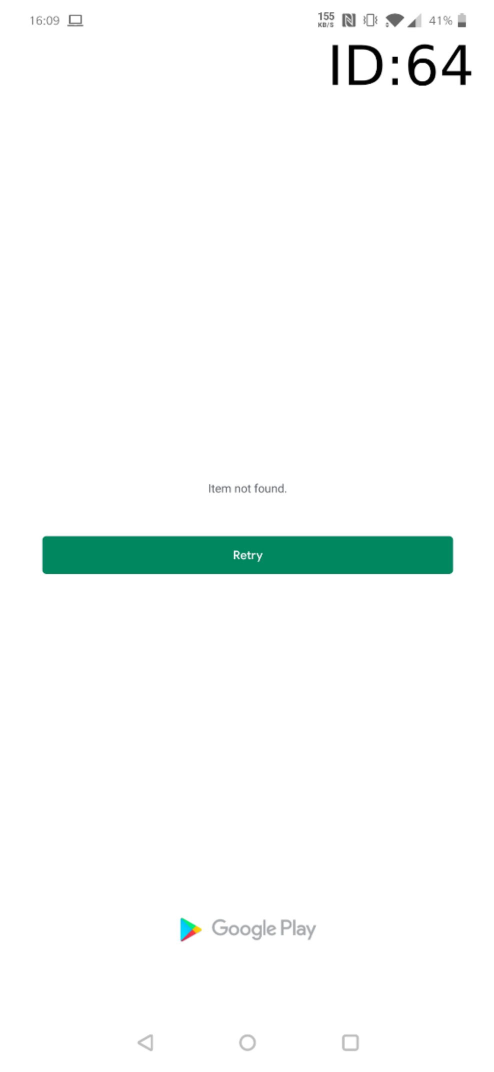


Welcome to ParkLet Shaun

CREATE A VEHICLE BOOKING

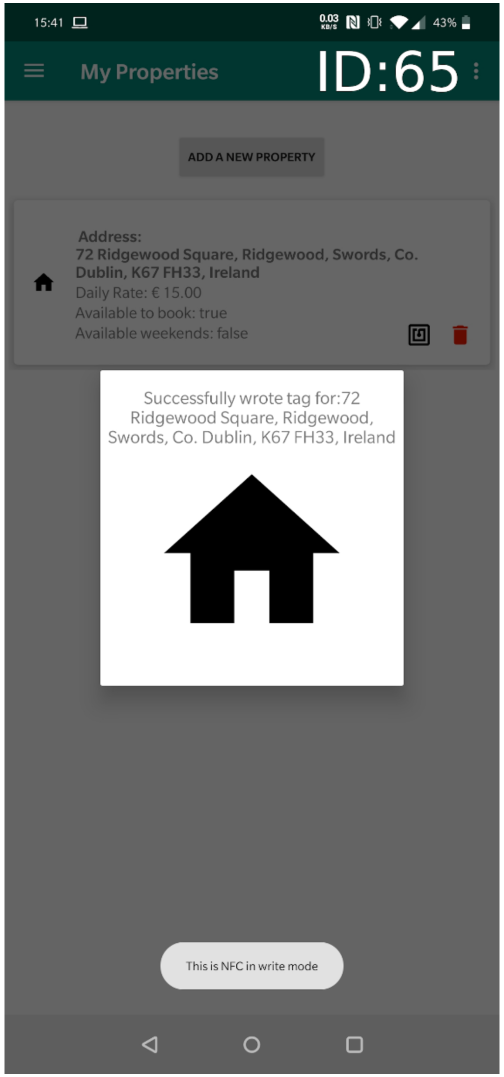
You have no current bookings

Booking not found



Item not found.

Retry

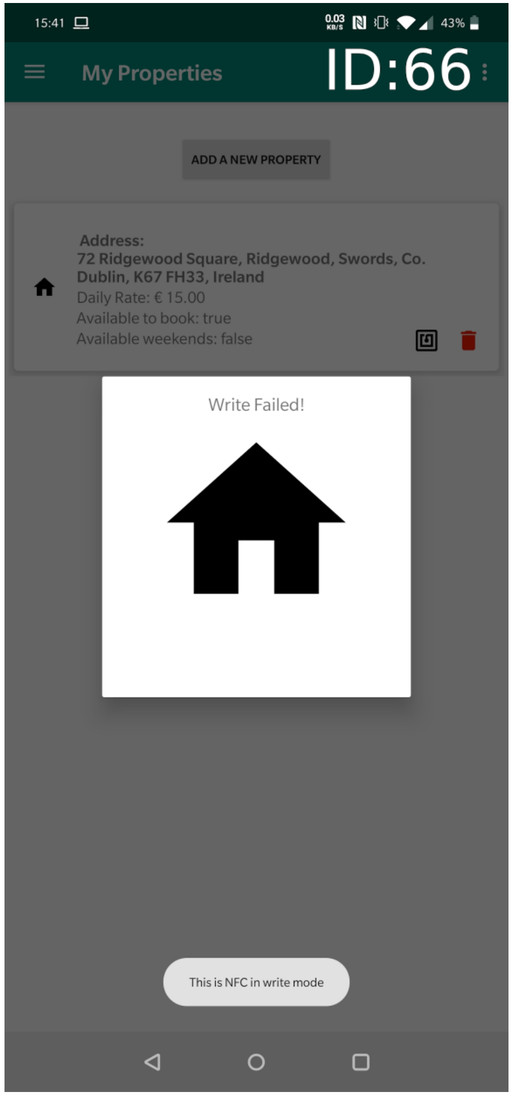


ADD A NEW PROPERTY

Address:
72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland
Daily Rate: € 15.00
Available to book: true
Available weekends: false

Successfully wrote tag for:72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland

This is NFC in write mode



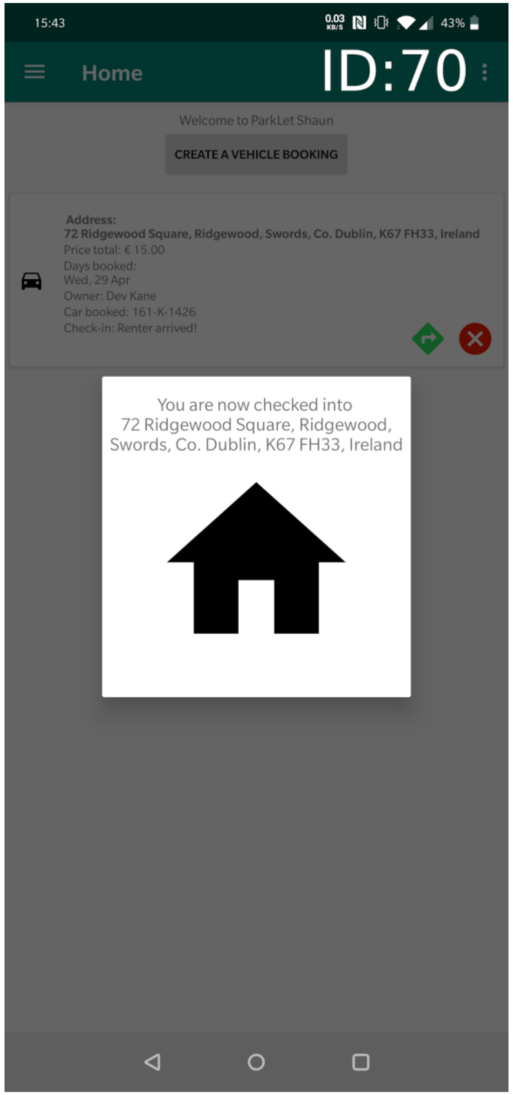
My Properties ID:66

ADD A NEW PROPERTY

Address:
72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland
Daily Rate: € 15.00
Available to book: true
Available weekends: false

Write Failed!

This is NFC in write mode



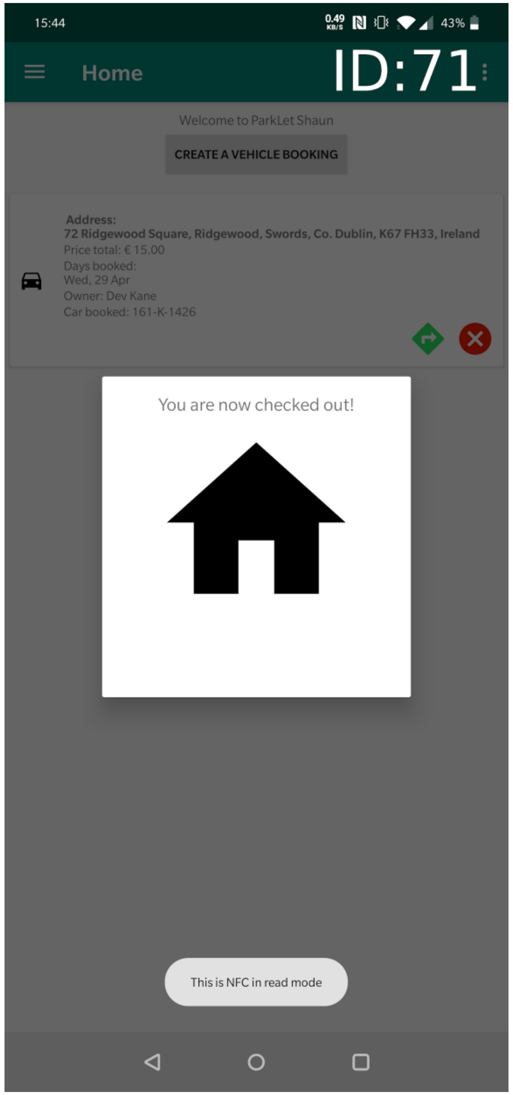
Home ID:70

Welcome to ParkLet Shaun

CREATE A VEHICLE BOOKING

Address:
72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland
Price total: € 15.00
Days booked:
Wed, 29 Apr
Owner: Dev Kane
Car booked: 161-K-1426
Check-in: Renter arrived!

You are now checked into 72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland



Home ID:71

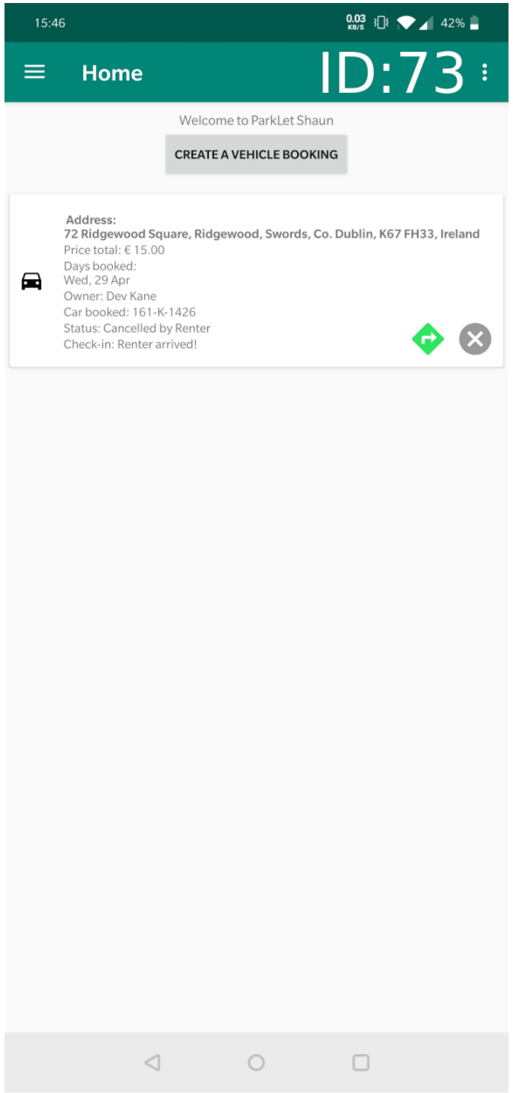
Welcome to ParkLet Shaun

CREATE A VEHICLE BOOKING

Address:
72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland
Price total: € 15.00
Days booked:
Wed, 29 Apr
Owner: Dev Kane
Car booked: 161-K-1426

You are now checked out!

This is NFC in read mode



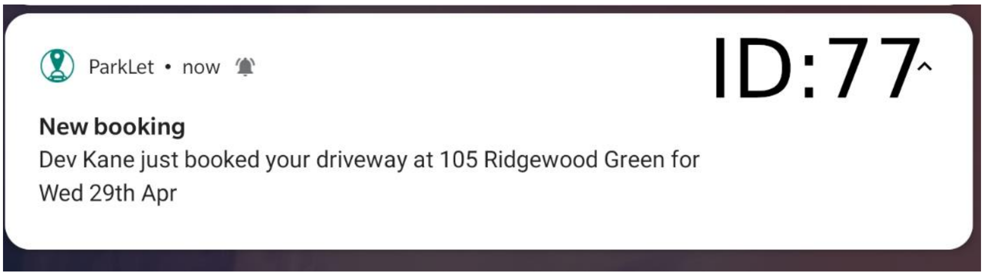
Home ID:73

Welcome to ParkLet Shaun

CREATE A VEHICLE BOOKING

Address:
72 Ridgewood Square, Ridgewood, Swords, Co. Dublin, K67 FH33, Ireland
Price total: € 15.00
Days booked:
Wed, 29 Apr
Owner: Dev Kane
Car booked: 161-K-1426
Status: Cancelled by Renter
Check-in: Renter arrived!

FCM Notification scenarios				
ID	**Test case**	**Expected:**	**Actual:**	**Outcome:**
77	Homeowner receives new property booking	FCM message delivered with booking details, regardless of app launched or not	As expected.	Pass
78	Homeowner receives check-in status from renter	FCM showing what time the renter checked in at, the vehicle for the booking, the property booked and renters name	As expected.	Pass
79	Homeowner receives checkout status from renter	FCM showing what time the renter checked out at, the vehicle for the booking, the property booked and renters name	As expected.	Pass
80	Homeowner receives cancellation notice from renter	FCM informing homeowner that renter cancelled and their property is listed as available for the given dates	As expected.	Pass
81	Renter receives cancellation notice from homeowner	FCM informing renter that the homeowner has cancelled for given dates and they will receive refund	As expected.	Pass
82	User changes phone	FCM device token is correctly updated to newest version	As expected.	Pass
83	User changes account	FCM device token is correctly updated to newest version	As expected.	Pass
84	User uninstalls app and reinstalls	FCM device token is correctly updated to newest version	As expected.	Pass
85	Renter deletes vehicle associated with booking	Booking cancelled and owner notified	Booking not cancelled. No notification	Fail
86	Homeowner deletes property associated with bookings	Booking cancelled and renter notified	Booking not cancelled. No notification	Fail
87	User clicks notification with app in background	App is opened on home page showing new notification. If they are not signed in, sign in then redirect	As expected.	Pass
88	User clicks notification with app in foreground	App redirects to the home page showing new notification. Previous fragment destroyed	As expected.	Pass

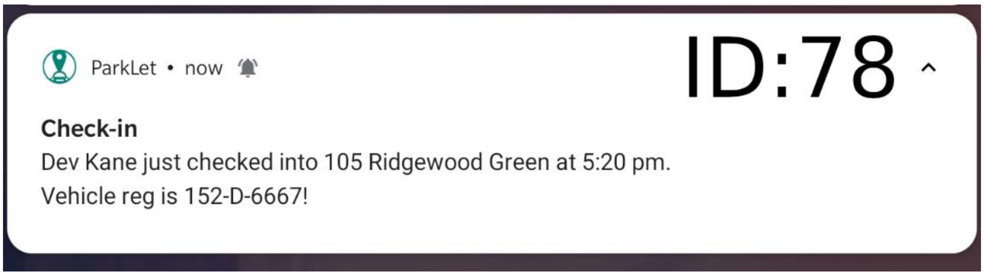


ParkLet • now

ID:77

New booking

Dev Kane just booked your driveway at 105 Ridgewood Green for Wed 29th Apr

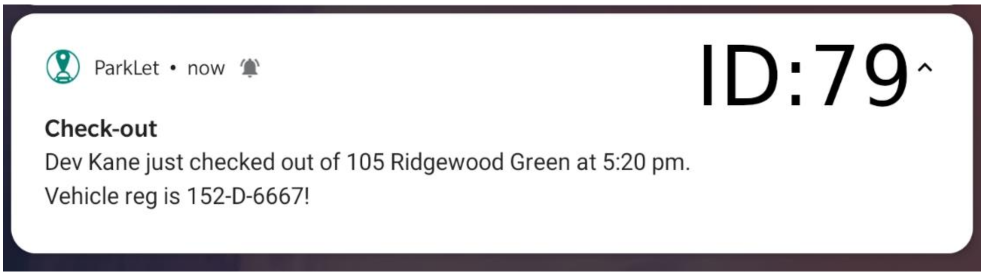


ParkLet • now

ID:78

Check-in

Dev Kane just checked into 105 Ridgewood Green at 5:20 pm.
Vehicle reg is 152-D-6667!

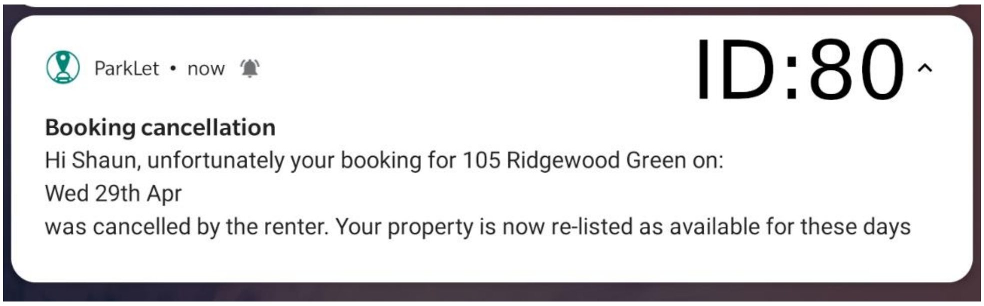


ParkLet • now

ID:79

Check-out

Dev Kane just checked out of 105 Ridgewood Green at 5:20 pm.
Vehicle reg is 152-D-6667!

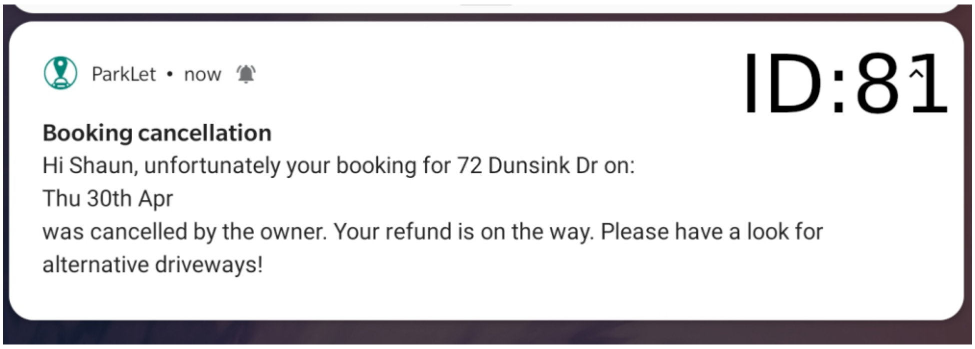


ParkLet • now

ID:80

Booking cancellation

Hi Shaun, unfortunately your booking for 105 Ridgewood Green on:
Wed 29th Apr
was cancelled by the renter. Your property is now re-listed as available for these days



ParkLet • now

ID:81

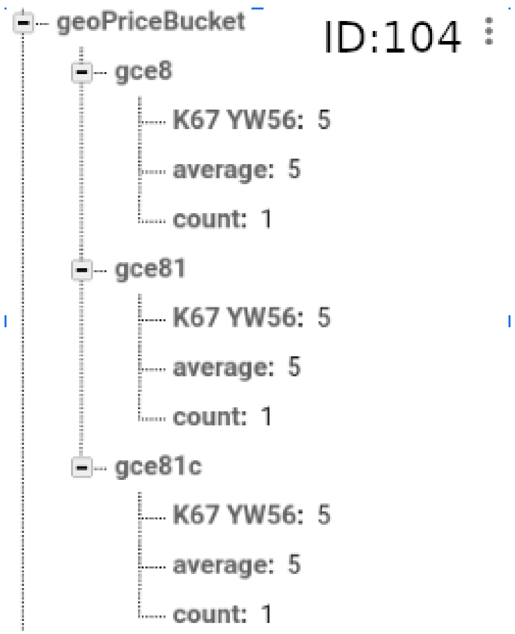
Booking cancellation

Hi Shaun, unfortunately your booking for 72 Dunsink Dr on:
Thu 30th Apr
was cancelled by the owner. Your refund is on the way. Please have a look for alternative driveways!

Navigation scenarios

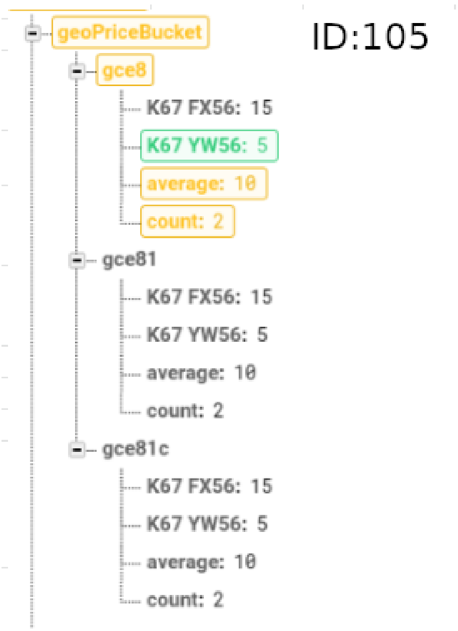
ID	Test case	Expected:	Actual:	Outcome:
89	Home to properties	Properties page loaded with properties	As expected.	Pass
90	Properties to Home	Home page loaded with bookings	As expected.	Pass
91	Properties to Vehicles	Vehicles page loaded with vehicles	As expected.	Pass
92	Home to vehicles	Vehicles page loaded with vehicles	As expected.	Pass
93	Vehicles to Home	Home page loaded with bookings	As expected.	Pass
94	Vehicles to Property	Properties page loaded with properties	As expected.	Pass
95	Home backpress	App shuts	Blank screen shown, pressing again closes app	Pass
96	Property backpress	Back to previous page, i.e. home	As expected.	Pass
97	Vehicle backpress	Back to previous page, i.e. home	As expected.	Pass
98	Home to map	Map loaded with input fields	As expected.	Pass
99	Map backpress	Home page loaded with bookings	As expected.	Pass
100	Map to booking	Booking calendar for given property	As expected.	Pass
101	Booking backpress	Back to map loaded with query	As expected. Query is entered in fields but need to submit request again	Pass
102	Booking after placing booking back to home	Home page updated with booking	As expected.	Pass
103	Home to Gmaps directions, then backpress	Homepage with bookings	As expected. Query is entered in fields but need to submit request again	Pass
104	MapView directions then backpress	Back to map loaded with query	As expected. Query is entered in fields but need to submit request again	Pass

GeoPriceBucket scenario				
ID	**Test case**	**Expected:**	**Actual:**	**Outcome:**
104	User adds first property in a given area	A GeoPriceBucket is triggered to be created for a geohash of lengths 6, 5 and 4. Sharing a common prefix. Average is calculated from each property that falls within those buckets	As expected.	Pass
105	Another property is added to the GeoBucket geohash of length 6 (Street level) length 5 (estate level) and length 4 (town level)	The previously found GeoBucket hash is found and the average is updated in a single pass. A user searching for a property will have their UI updated to reflect new price	As expected.	Pass
106	Another property is added to the GeoBucket geohash of length 5 (estate level) length 4 (Town level)	A new length 6 GeoBucket is created but length 5 and 4 are shared by all common prefixes. Average price is calculated correctly for all given GeoPriceBuckets	As expected.	Pass
107	User deletes property in area with many houses	Cloud functions is triggered to remove said property from any GeoPriceBucket it is contained in and calculates new average in single pass	As expected.	Pass
108	User deletes property which is last in a GeoPriceBucket	As it's the last property in a given GeoPriceBucket, the bucket is simply made null and removed. If the property exists in GeoBucket’s corresponding to a larger area then they are updated according to if there are remaining properties	As expected.	Pass



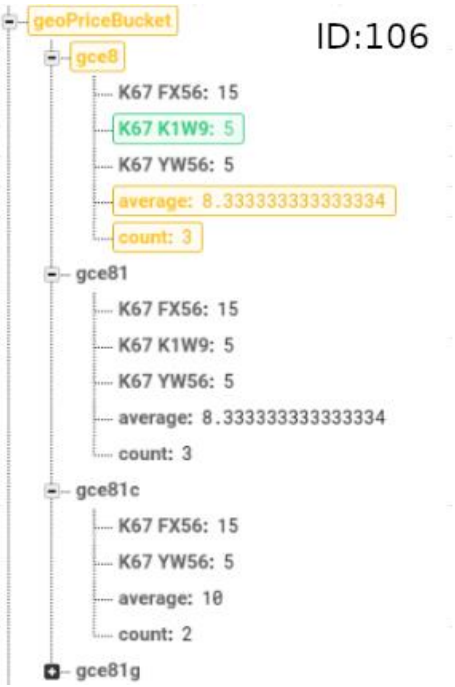
ID:104

```
geoPriceBucket
    gce8
        K67 YW56: 5
        average: 5
        count: 1
    gce81
        K67 YW56: 5
        average: 5
        count: 1
    gce81c
        K67 YW56: 5
        average: 5
        count: 1
```



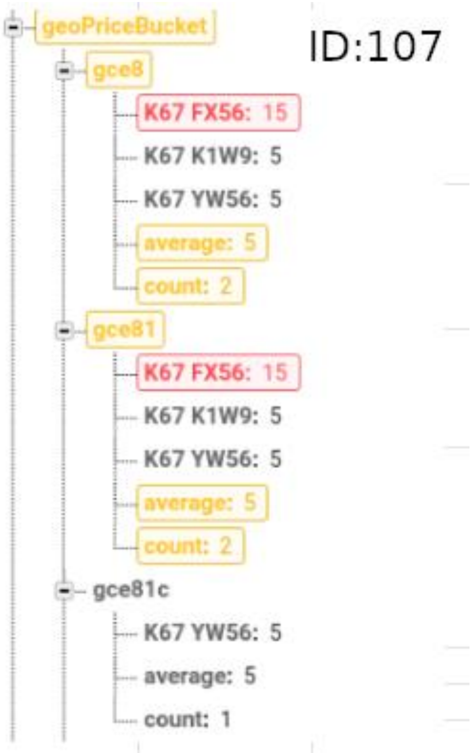
ID:105

```
geoPriceBucket
    gce8
        K67 FX56: 15
        K67 YW56: 5
        average: 10
        count: 2
    gce81
        K67 FX56: 15
        K67 YW56: 5
        average: 10
        count: 2
    gce81c
        K67 FX56: 15
        K67 YW56: 5
        average: 10
        count: 2
```



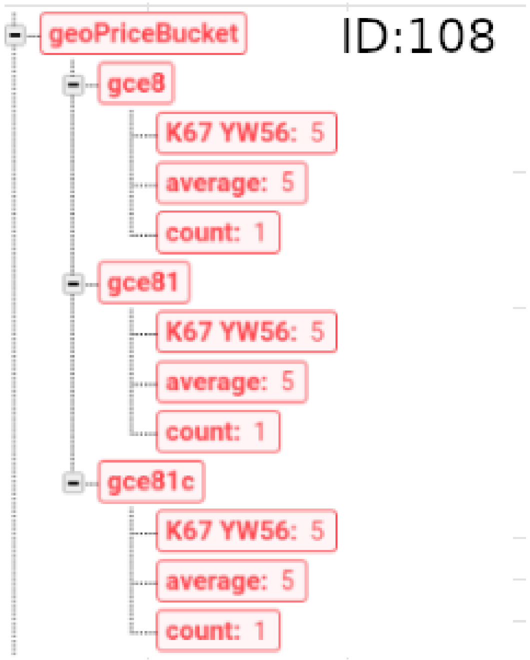
ID:106

```
geoPriceBucket
    gce8
        K67 FX56: 15
        K67 K1W9: 5
        K67 YW56: 5
        average: 8.333333333333334
        count: 3
    gce81
        K67 FX56: 15
        K67 K1W9: 5
        K67 YW56: 5
        average: 8.333333333333334
        count: 3
    gce81c
        K67 FX56: 15
        K67 YW56: 5
        average: 10
        count: 2
    gce81g
```



ID:107

```
geoPriceBucket
    gce8
        K67 FX56: 15
        K67 K1W9: 5
        K67 YW56: 5
        average: 5
        count: 2
    gce81
        K67 FX56: 15
        K67 K1W9: 5
        K67 YW56: 5
        average: 5
        count: 2
    gce81c
        K67 YW56: 5
        average: 5
        count: 1
```



ID:108

```
geoPriceBucket
    gce8
        K67 YW56: 5
        average: 5
        count: 1
    gce81
        K67 YW56: 5
        average: 5
        count: 1
    gce81c
        K67 YW56: 5
        average: 5
        count: 1
```