ParkHere Testing

Team Black Lightning

Oscar Leung -- 009854091

Jason Liu -- 009977695

Stephen Reyes -- 009862892

Phuong Tran -- 009321208

Ziwei Wu -- 009872499

Preface

This testing document describes the testing that we did for our ParkHere app using both black-box and white-box testing methods. This document includes the following: instructions for executing the test cases, the details of the black-box tests, and the details of the white-box tests.

Instruction

Instructions to run the tests:

White-box tests

1. You should have at least 15 executable white-box test cases.*
2. You should clearly identify the coverage criteria used in your white-box testing: What coverage objective(s) did you opt for and why? How did you ensure you are meeting your coverage criterion?
3. Each test case must have the following information
    a. The location of the test case (e.g. in which folder, which file, what is the name of the test case).
    b. Description of what does the test case do and how to execute the test case.
    c. Description of the rationale behind the test case (e.g., Why do you choose the specific inputs? How do you generate the test case?).
    d. Result of the test case (e.g., screenshot with description).
    e. If the test case helps you uncover a bug, document the details (e.g., What is the bug? How did you fix it?). Remember to do regression testing after fixing the bug, to make sure you did not break anything that was working previously.

Our coverage criteria we opted for were condition and branch coverage.

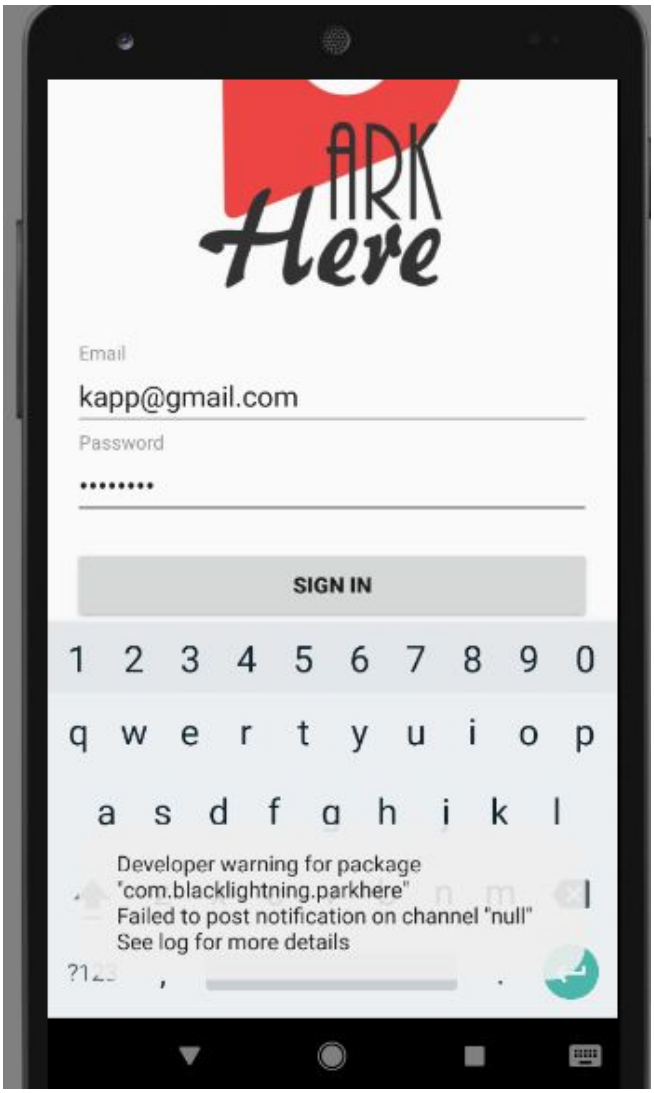
We chose these ones because we wanted to make sure that the if-else statements in the program were working. Things such as not being able to log in or logging in with an incorrect email and/or password should not happen in a functioning app.

We ensured that we were meeting our coverage criterion by testing the

Test case #1:

- Location
    - ParkHere-BlackLightning\app\src\androidTest\java\com\blacklightning\parkhere\LoginUnitTest.java

    - The name of the test case is “logInWithWrongEmail()”
- Description
    - This test is to test if the app will log someone in with an incorrect email or not. If there is no user logged in, then the getCurrentUser() method in the LoginActivity class will return null.
- Rationale
    - This test is to make sure that emails and password combinations have to be stored into Firebase before being able to log in.
- Result:



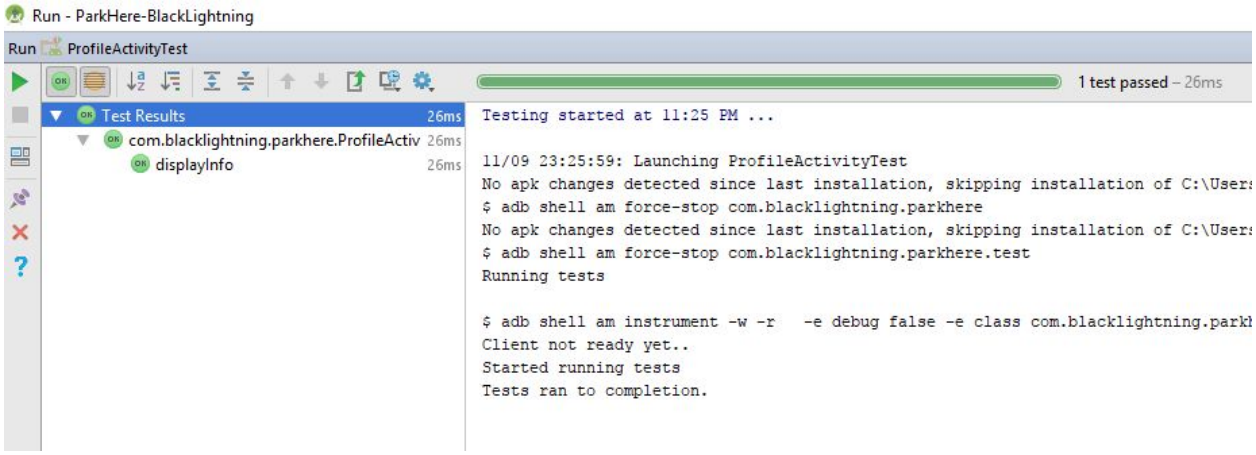
Picture of result: the getCurrentUser() method returned null, meaning no login occurred

- Bugs and details: No bugs were detected

Test case #2:

- Location

    - ParkHere-BlackLightning\app\src\androidTest\java\com\blacklightning\parkhere\ProfileActivityTest.java
- Description
    - This tests checks if the login is successful with a set email and password that has already been set up in Firebase. This also test that we can get the user information from the database and display it correctly/
- Rationale: We need to make sure that logging in works with a valid email and password.
- Result:



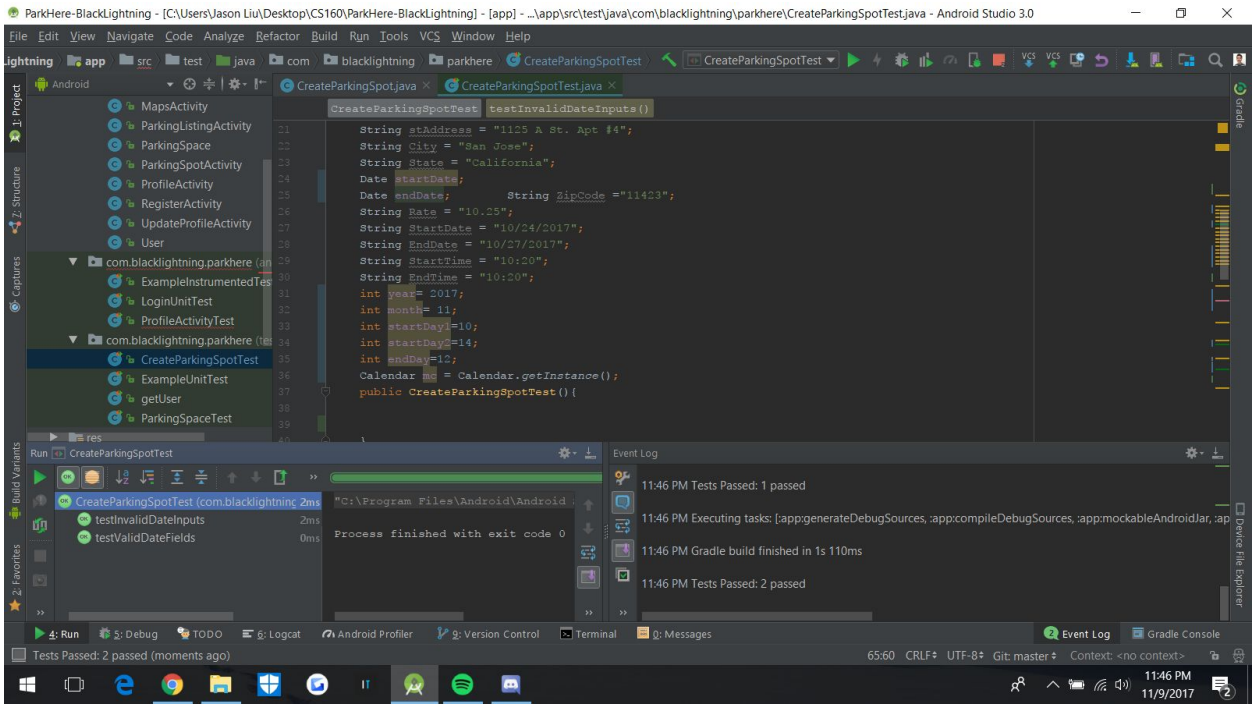
Picture of the result. The test was passed.

- Bugs and details: No bugs were detected.

Test case #3:

- Location
    - ParkHere-BlackLightning\app\src\test\java\com\blacklightning\parkhere\CreateParkingSpotTest.java
    - The name of the test case is “public void user()”
- Description
    - This tests if the getUserId of the User class works or not
- Rationale:
    - We test this in order to make sure that the details are updated when the user changes their information

- Result:



The result was ran to completion and it successfully passed the test.

-
- Bugs and details
    - There were no bugs found in this test.

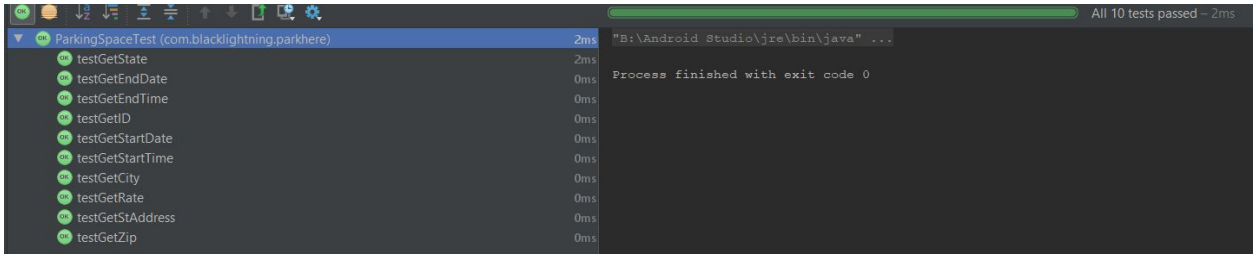
Test case #4:

- Location ParkHere-BlackLightning\app\src\test\java\com\blacklightning\parkhere\CreateParkingSpace.java
    - testValidDateFields()
- Description : checks if start and end dates are are valid entries
- Rationale : inputting invalid entries could compromise user understanding
- Result: Test was passed based on dates given
- Bugs and details : no bugs detected

Test case #5:

- Location: ParkHere-BlackLightning\app\src\test\java\com\blacklightning\parkhere\ParkingSpaceTest.java
    - testGetState()
- Description: checks if getting a state from ParkingSpace works
- Rationale: the storage class returning null items will cause reference problems

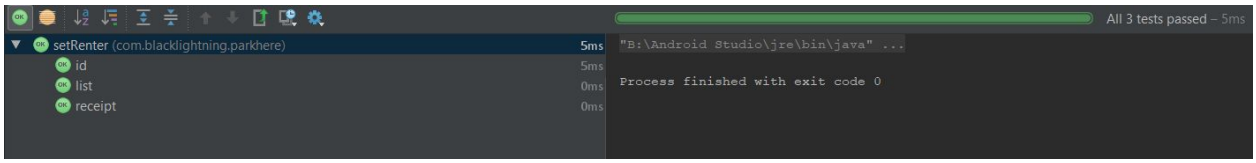
- Result:



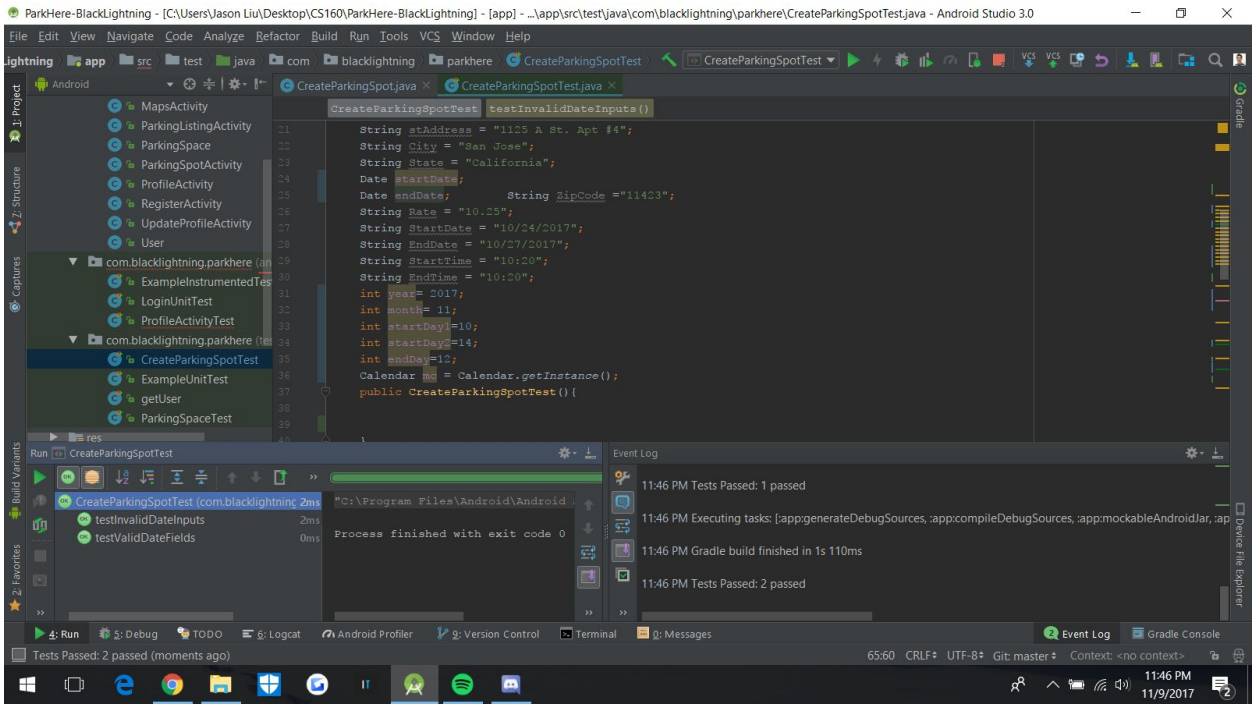
- Bugs and details: no bugs, runs smoothly

Test case #6:

- Location: B:\ParkHereABC\ParkHere-BlackLightning\app\src\test\java\com\blacklightning\parkhere\getUser.java
- Description: Test Getter for user
- Rationale
- Result: Works normally
- Bugs and details: There is no bug



Test case #7:

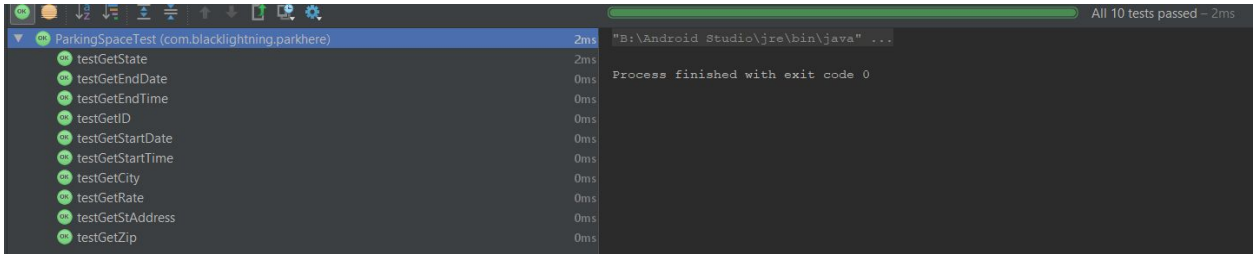


Test case #7:

- Location ParkHere-BlackLightning\app\src\test\java\com\blacklightning\parkhere\CreateParkingSpace.java
    - testInvalidDateFields()
- Description : checks for invalid starting dates input End date was before start date
- Rationale : inputting invalid entries could compromise user understanding
- Result: Test was passed based on dates given case was handled
- Bugs and details : no bugs detected

Test case #8:

- Location: ParkHere-BlackLightning\app\src\test\java\com\blacklightning\parkhere\ParkingSpaceTest.java
    - testGetID()
- Description: checks if getting a parking spot ID from ParkingSpace works
- Rationale: the storage class returning null items will cause reference problems, especially when mapping on the DB
- Result:



- Bugs and details: no bugs, can find each parking space accordingly
-

Test case #9:

- Location
- Description
- Rationale
- Result
- Bugs and details

Test case #10:

- Location
- Description
- Rationale
- Result
- Bugs and details

Test case #11:

- Location
- Description
- Rationale
- Result

- Bugs and details

Test case #12:

- Location
- Description
- Rationale
- Result
- Bugs and details

Test case #13:

- Location
- Description
- Rationale
- Result
- Bugs and details

Test case #14:

- Location
- Description
- Rationale
- Result
- Bugs and details

Test case #15:

- Location
- Description
- Rationale
- Result
- Bugs and details