

## Team Squirrel FDM Mobile Application Testing

There were two main methods used to test the android application; an automated high level GUI testing framework, and low level unit testing. The GUI testing was created by using the Espresso framework to generate code that automates GUI functions. The low level testing was handled by custom test classes that extend the tested class page.

### GUI Testing with the Espresso Framework

Several tests were created with the Espresso test recorder that records *assertions* - individual interactions with the GUI by manipulating the GUI through the Espresso test recording interface.

The created tests:

*loginTest.java* – responsible for testing the different login eventualities on the app.

*searchCourseTest.java* – responsible for testing the search courses feature on the app.

*bookTest.java* – responsible for testing booking courses from the app.

*unbookCourseTest.java* – responsible for testing the un-book course function on the app.

As well as their specific aims, the GUI tests also have many implications for other aspects of the app's functionality: they test all possible GUI interactions the user could perform, such as viewing courses as selected from the search courses or my courses pages. Also tested is the ability of the app to generate a map widget with the location of the classroom. The pin on the map is generated at runtime from the address given by the server.

### loginTest.java

#### Basic flow:

User lands on welcome page.

User clicks login button.

User enters details incorrectly with wrong username and password.

User enters details incorrectly with correct username and wrong password.

User enters correct details and logs in.

User logs out.

### Flow Diagram



bookTest.java

Basic flow:

User lands on welcome page.

User clicks login button.

User enters details and logs in.

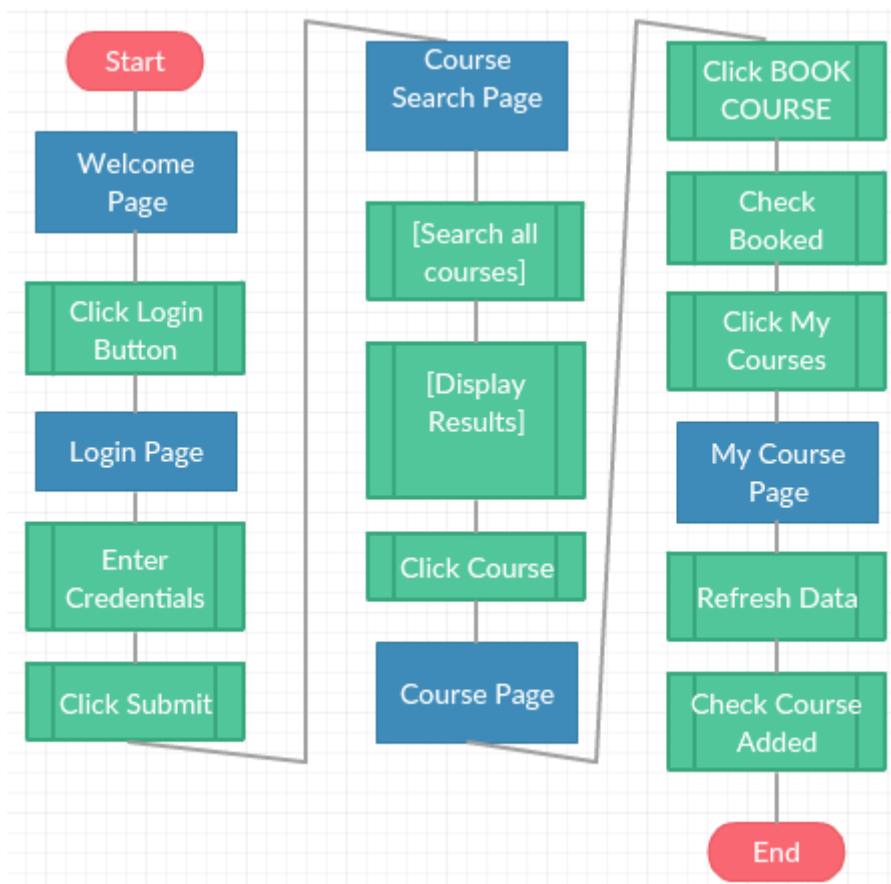
User selects course.

User books course.

User checks course is added to My Courses.

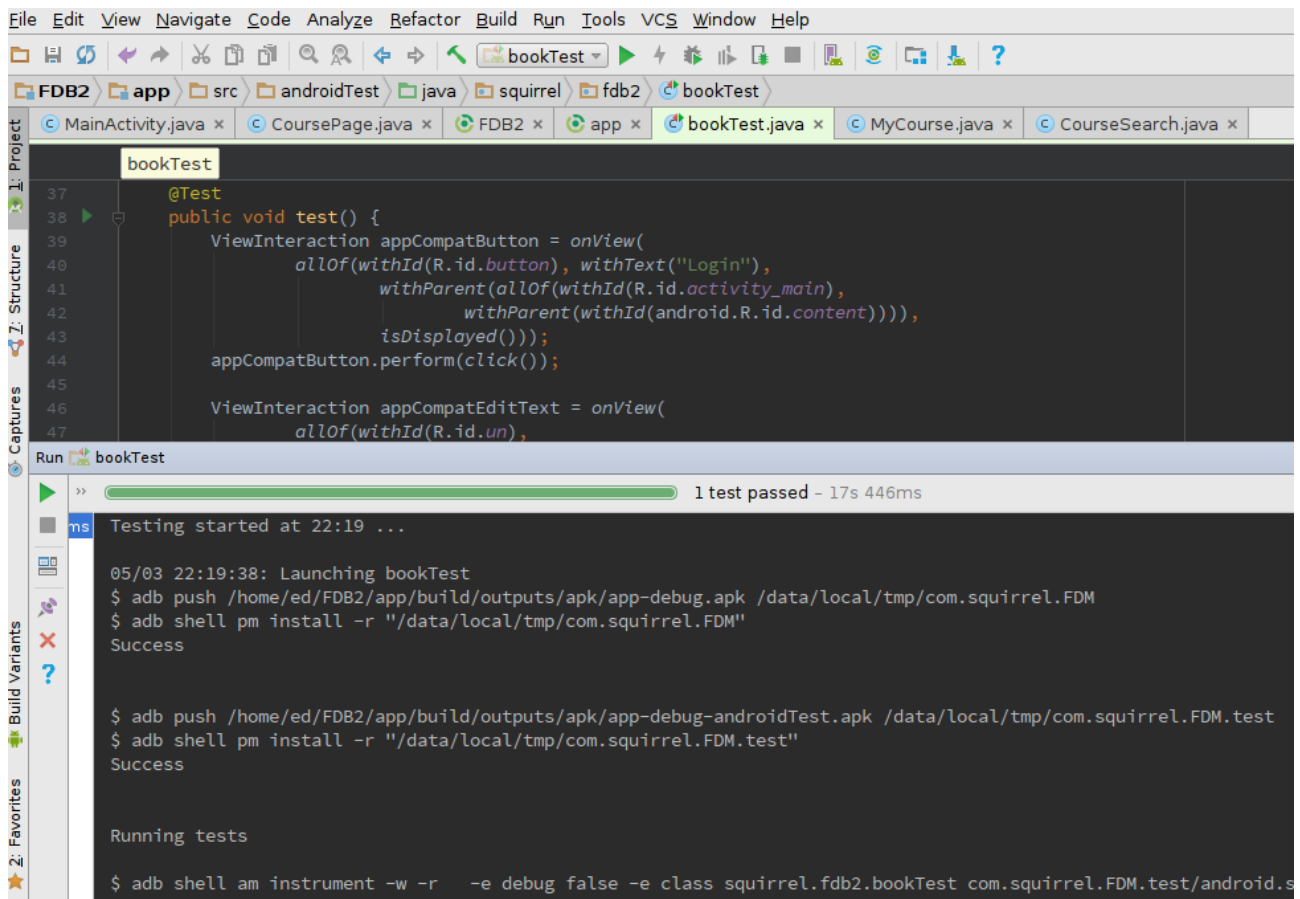
User Logs out.

Flow Diagram:



*Flow diagram of the bookCourse.java test.*

Results:



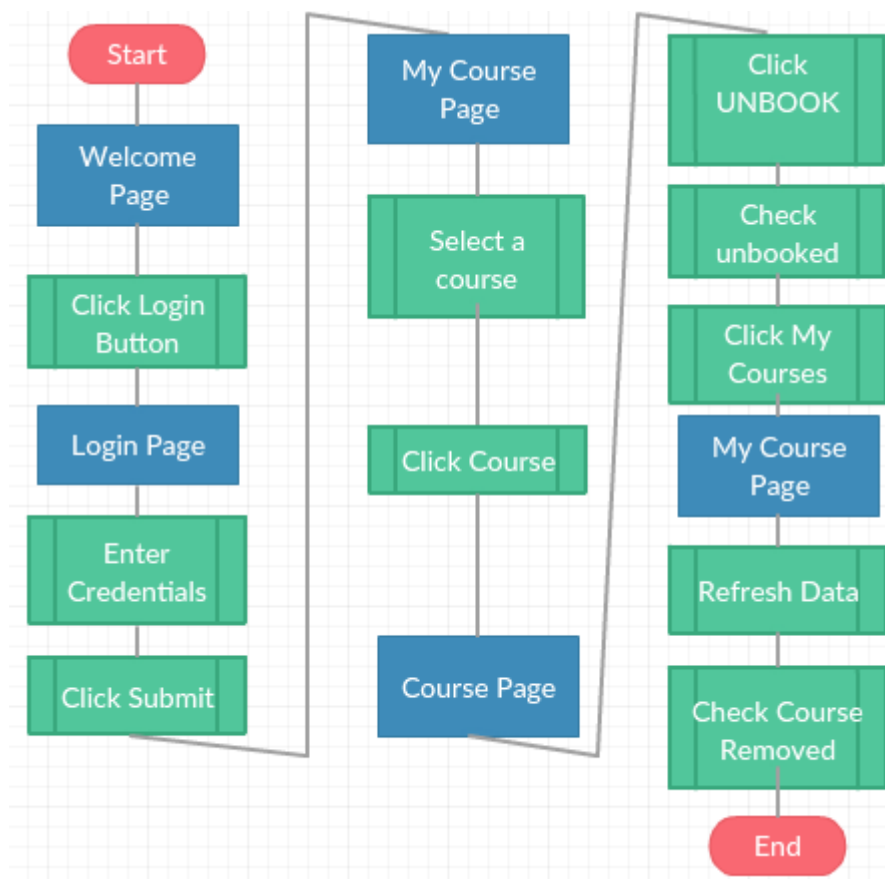
*Result after running the bookCourse.java test file with the database and server connected.  
Test Success.*

### unbookCourseTest.java

#### Basic Flow

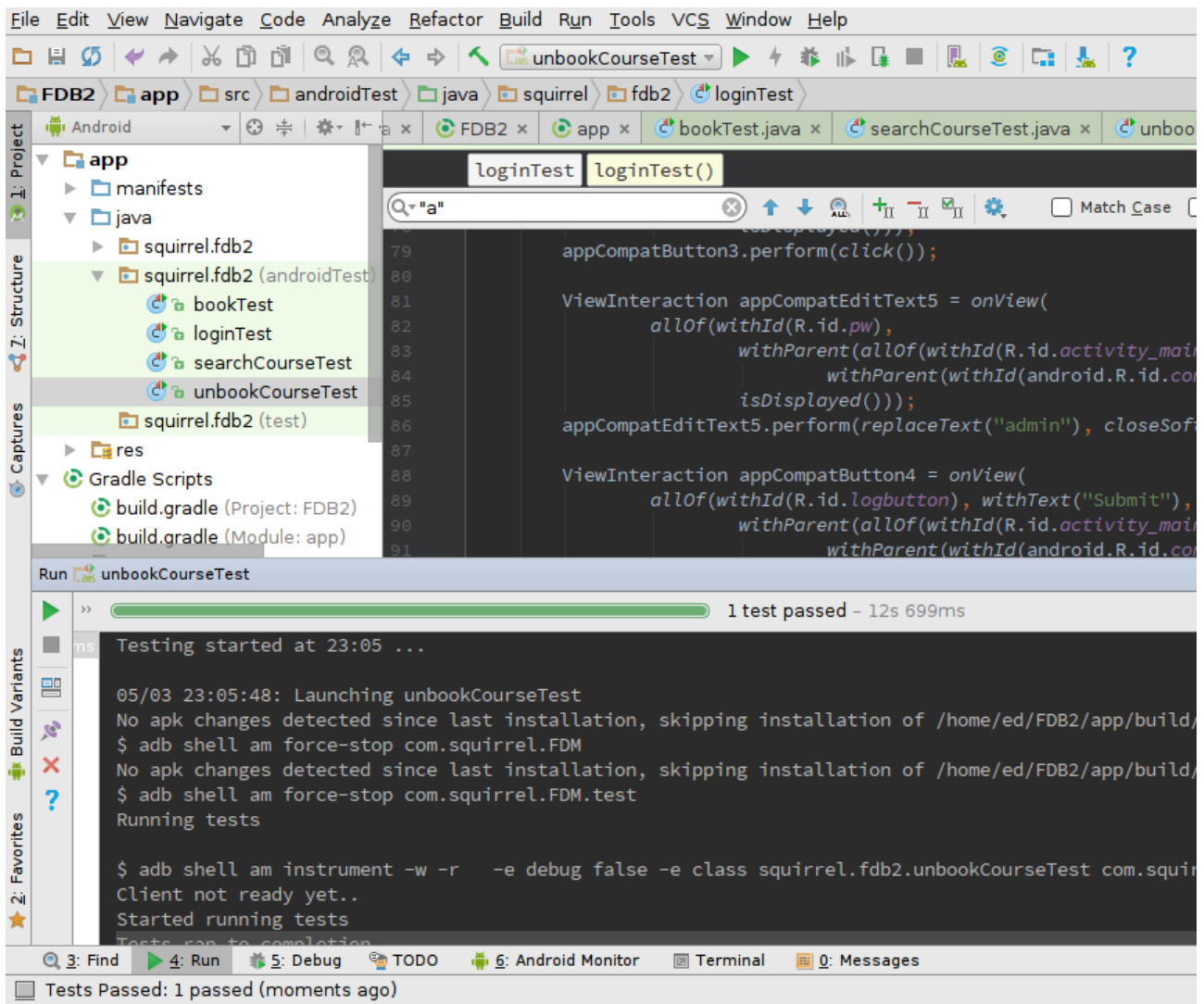
User lands on welcome page.  
User clicks login button.  
User enters details and logs in.  
User goes to My Course Page.  
User selects a course.  
User unbooks course.  
User checks course is removed from My Courses.  
User Logs out.

#### Flow Diagram



Flow diagram of `unbookCourseTest.java`.

## Results



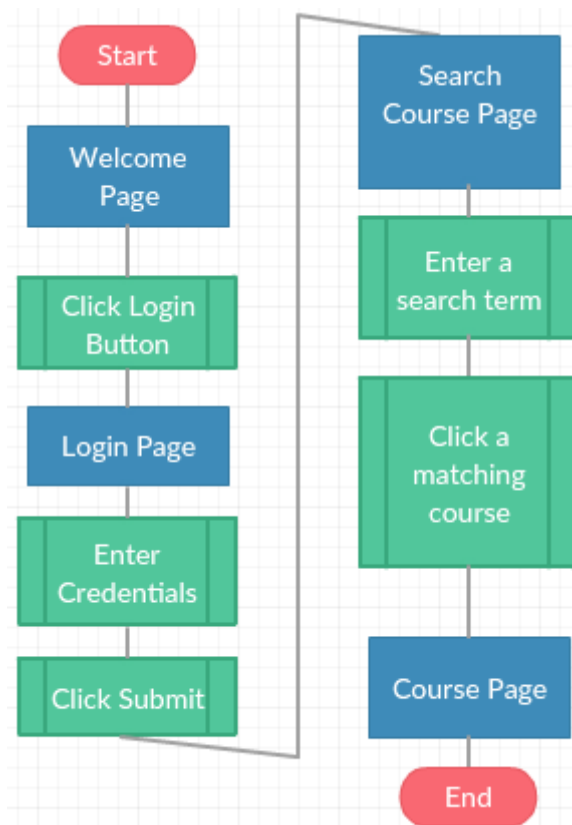
Results after running `unbookCourseTest.java` with the database and server connected.  
Test Success.

### SearchCourseTest.java

#### Basic Flow

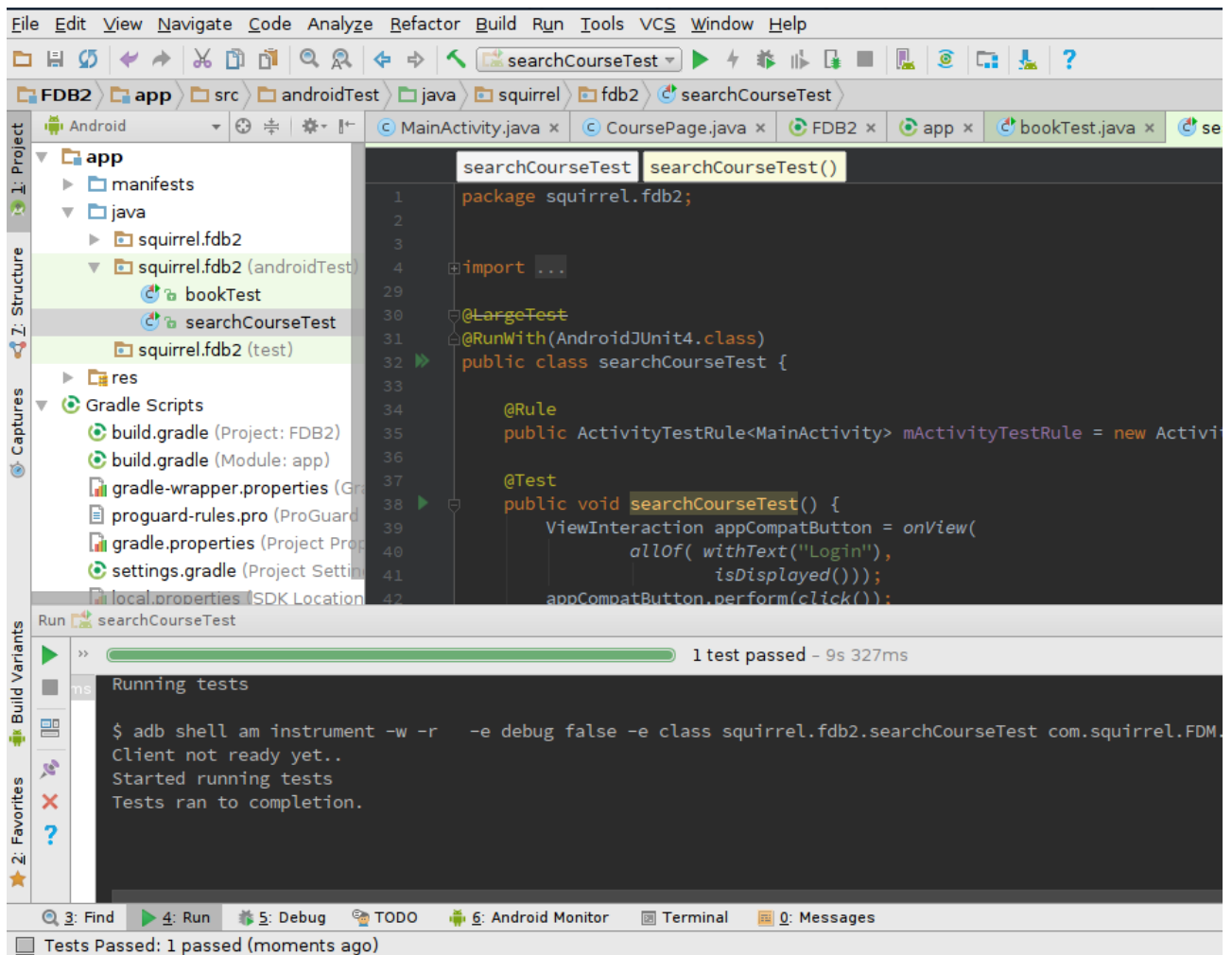
User lands on welcome page.  
User clicks login.  
User enters details and logs in.  
User enters a search term into the box.  
User clicks on matching result.  
User logs out.

#### Flow Diagram



*Flow diagram for searchCourseTest.java.*

Results



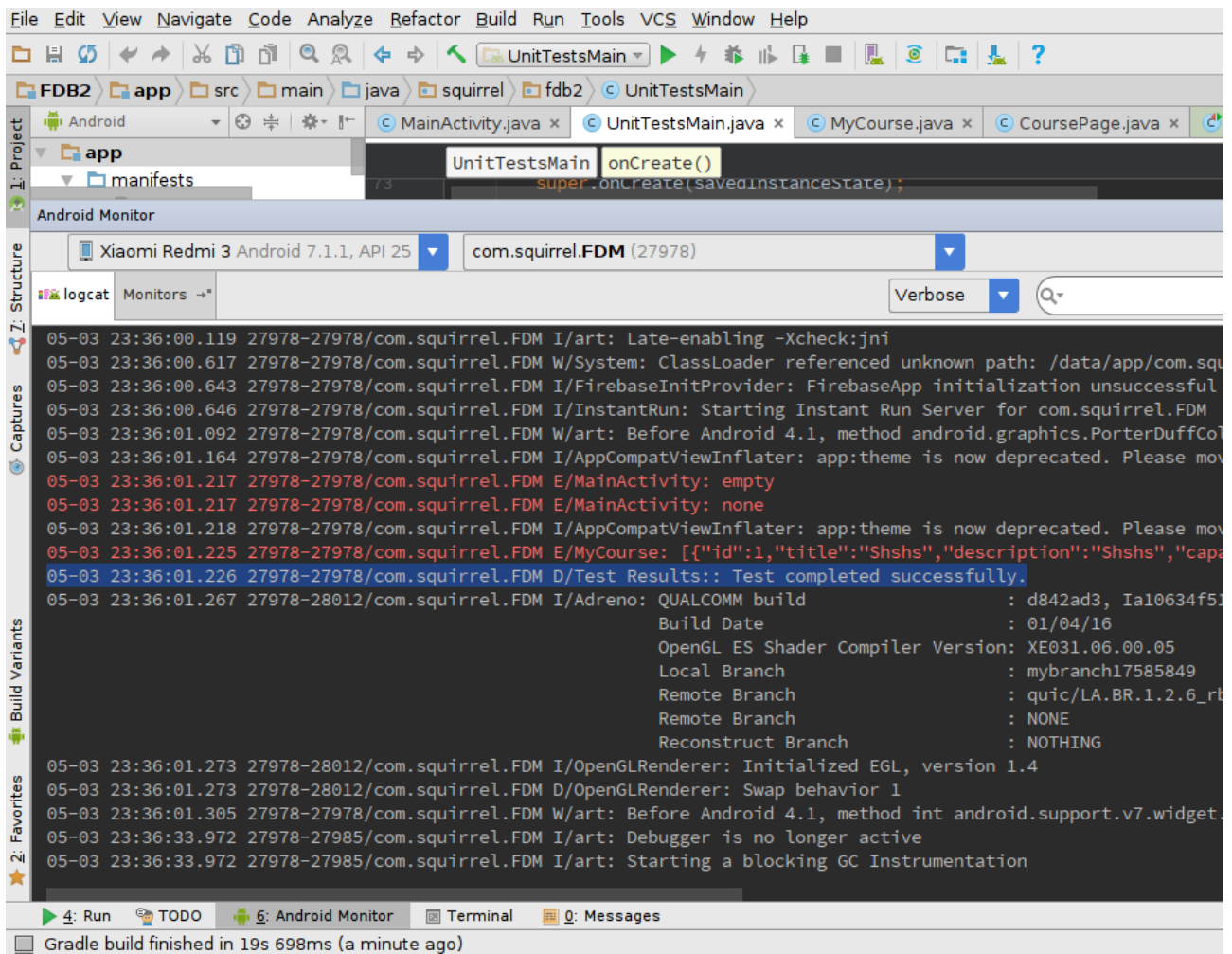
*Results after searchCourseTest.java was run with the sever and database connected.  
Test Succes.*

## Low Level Testing

Some of the vital functions were tested in the custom test class UnitTestsMain.java. This class was run isolated from the app to make sure that the core functions were working such as GUI renderring, layout application, toolbar renderring and also some get and set methods used throughout the app.

Restults:





Results after running an isolated instance of the test class `UnitTestsMain.java`.  
Test Success.

## Appendix

### `loginTest.java`

```
package squirrel.fdb2;
import android.support.test.espresso.ViewInteraction;
import android.support.test.rule.ActivityTestRule;
import android.support.test.runner.AndroidJUnit4;
import android.test.suitebuilder.annotation.LargeTest;

import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

import static android.support.test.espresso.Espresso.onView;
import static android.support.test.espresso.action.ViewActions.click;
import static
android.support.test.espresso.action.ViewActions.closeSoftKeyboard;
import static android.support.test.espresso.action.ViewActions.replaceText;
import static android.support.test.espresso.matcher.ViewMatchers.isDisplayed;
import static android.support.test.espresso.matcher.ViewMatchers.withId;
import static android.support.test.espresso.matcher.ViewMatchers.withParent;
import static android.support.test.espresso.matcher.ViewMatchers.withText;
import static org.hamcrest.Matchers.allOf;

@LargeTest
```

```

@RunWith(AndroidJUnit4.class)
public class loginTest {

    @Rule
    public ActivityTestRule<MainActivity> mActivityTestRule = new
ActivityTestRule<>(MainActivity.class);

    @Test
    public void loginTest() {
        ViewInteraction appCompatButton = onView(
            allOf(withId(R.id.button), withText("Login"),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatButton.perform(click());

        ViewInteraction appCompatEditText = onView(
            allOf(withId(R.id.un),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatEditText.perform(click());

        ViewInteraction appCompatEditText2 = onView(
            allOf(withId(R.id.un),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatEditText2.perform(replaceText("a"), closeSoftKeyboard());

        ViewInteraction appCompatButton2 = onView(
            allOf(withId(R.id.logbutton), withText("Submit"),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatButton2.perform(click());

        ViewInteraction appCompatEditText3 = onView(
            allOf(withId(R.id.un),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatEditText3.perform(click());

        ViewInteraction appCompatEditText4 = onView(
            allOf(withId(R.id.un),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatEditText4.perform(replaceText("admin"), closeSoftKeyboard());

        ViewInteraction appCompatButton3 = onView(
            allOf(withId(R.id.logbutton), withText("Submit"),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatButton3.perform(click());

        ViewInteraction appCompatEditText5 = onView(
            allOf(withId(R.id.pw),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatEditText5.perform(replaceText("admin"), closeSoftKeyboard());
    }
}

```

```

        ViewInteraction appCompatButton4 = onView(
            allOf(withId(R.id.logbutton), withText("Submit"),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatButton4.perform(click());
    }
}

```

## searchCourseTest.java

```

package squirrel.fdb2;
import android.support.test.espresso.ViewInteraction;
import android.support.test.rule.ActivityTestRule;
import android.support.test.runner.AndroidJUnit4;
import android.test.suitebuilder.annotation.LargeTest;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewParent;

import org.hamcrest.Description;
import org.hamcrest.Matcher;
import org.hamcrest.TypeSafeMatcher;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

import static android.support.test.espresso.Espresso.onView;
import static android.support.test.espresso.action.ViewActions.click;
import static
    android.support.test.espresso.action.ViewActions.closeSoftKeyboard;
import static android.support.test.espresso.action.ViewActions.replaceText;
import static android.support.test.espresso.action.ViewActions.scrollTo;
import static android.support.test.espresso.matcher.ViewMatchers.isDisplayed;
import static android.support.test.espresso.matcher.ViewMatchers.withId;
import static android.support.test.espresso.matcher.ViewMatchers.withParent;
import static android.support.test.espresso.matcher.ViewMatchers.withText;
import static org.hamcrest.Matchers.allOf;

@LargeTest
@RunWith(AndroidJUnit4.class)
public class searchCourseTest {

    @Rule
    public ActivityTestRule<MainActivity> mActivityTestRule = new
ActivityTestRule<>(MainActivity.class);

    @Test
    public void searchCourseTest() {
        ViewInteraction appCompatButton = onView(
            allOf( withText("Login"),
                isDisplayed()));
        appCompatButton.perform(click());

        ViewInteraction appCompatEditText = onView(
            allOf(withId(R.id.un),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
    }
}

```

```

appCompatEditText.perform(click());

ViewInteraction appCompatEditText2 = onView(
    allOf(withId(R.id.un),
        withParent(allOf(withId(R.id.activity_main),
            withParent(withId(android.R.id.content)))),
        isDisplayed()));
appCompatEditText2.perform(replaceText("admin"), closeSoftKeyboard());

ViewInteraction appCompatEditText3 = onView(
    allOf(withId(R.id.pw),
        withParent(allOf(withId(R.id.activity_main),
            withParent(withId(android.R.id.content)))),
        isDisplayed()));
appCompatEditText3.perform(replaceText("admin"), closeSoftKeyboard());

ViewInteraction appCompatButton2 = onView(
    allOf(withId(R.id.logbutton), withText("Submit"),
        withParent(allOf(withId(R.id.activity_main),
            withParent(withId(android.R.id.content)))),
        isDisplayed()));
appCompatButton2.perform(click());

ViewInteraction appCompatButton3 = onView(
    allOf(withId(R.id.my_courses), withText("MY COURSES"),
isDisplayed()));
appCompatButton3.perform(click());

ViewInteraction tableLayout = onView(
    allOf(childAtPosition(
        withId(R.id.mc_list),
        1),
        isDisplayed()));
tableLayout.perform(click());

ViewInteraction appCompatButton4 = onView(
    allOf(withId(R.id.search_courses), withText("SEARCH"),
isDisplayed()));
appCompatButton4.perform(click());

ViewInteraction appCompatEditText4 = onView(
    withId(R.id.cs_field));
appCompatEditText4.perform(scrollTo(), replaceText("che"),
closeSoftKeyboard());

ViewInteraction tableLayout2 = onView(
    childAtPosition(
        withId(R.id.cs_list),
        0));
tableLayout2.perform(scrollTo(), click());

ViewInteraction appCompatButton5 = onView(
    allOf(withId(R.id.logout), withText("LOGOUT"), isDisplayed()));
appCompatButton5.perform(click());

}

private static Matcher<View> childAtPosition(
    final Matcher<View> parentMatcher, final int position) {

    return new TypeSafeMatcher<View>() {
        @Override
        public void describeTo(Description description) {
            description.appendText("Child at position " + position + " in

```

```

parent "));
        parentMatcher.describeTo(description);
    }

    @Override
    public boolean matchesSafely(View view) {
        ViewParent parent = view.getParent();
        return parent instanceof ViewGroup &&
parentMatcher.matches(parent)
        && view.equals(((ViewGroup)
parent).getChildAt(position));
    }
};
}
}

```

## bookTest.java

```

package squirrel.fdb2;
import android.support.test.espresso.ViewInteraction;
import android.support.test.rule.ActivityTestRule;
import android.support.test.runner.AndroidJUnit4;
import android.test.suitebuilder.annotation.LargeTest;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewParent;

import org.hamcrest.Description;
import org.hamcrest.Matcher;
import org.hamcrest.TypeSafeMatcher;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

import static android.support.test.espresso.Espresso.onView;
import static android.support.test.espresso.action.ViewActions.click;
import static
android.support.test.espresso.action.ViewActions.closeSoftKeyboard;
import static android.support.test.espresso.action.ViewActions.replaceText;
import static android.support.test.espresso.action.ViewActions.scrollTo;
import static android.support.test.espresso.matcher.ViewMatchers.isDisplayed;
import static android.support.test.espresso.matcher.ViewMatchers.withId;
import static android.support.test.espresso.matcher.ViewMatchers.withParent;
import static android.support.test.espresso.matcher.ViewMatchers.withText;
import static org.hamcrest.Matchers.allOf;

@LargeTest
@RunWith(AndroidJUnit4.class)
public class bookTest{

    @Rule
    public ActivityTestRule<MainActivity> mActivityTestRule = new
ActivityTestRule<>(MainActivity.class);

    @Test
    public void test() {
        ViewInteraction appCompatButton = onView(
            allOf(withId(R.id.button), withText("Login"),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatButton.perform(click());
    }
}

```

```

ViewInteraction appCompatEditText = onView(
    allOf(withId(R.id.un),
        withParent(allOf(withId(R.id.activity_main),
            withParent(withId(android.R.id.content)))),
        isDisplayed()));
appCompatEditText.perform(click());

ViewInteraction appCompatEditText2 = onView(
    allOf(withId(R.id.un),
        withParent(allOf(withId(R.id.activity_main),
            withParent(withId(android.R.id.content)))),
        isDisplayed()));
appCompatEditText2.perform(replaceText("admin"), closeSoftKeyboard());

ViewInteraction appCompatEditText3 = onView(
    allOf(withId(R.id.pw),
        withParent(allOf(withId(R.id.activity_main),
            withParent(withId(android.R.id.content)))),
        isDisplayed()));
appCompatEditText3.perform(replaceText("admin"), closeSoftKeyboard());

ViewInteraction appCompatButton2 = onView(
    allOf(withId(R.id.logbutton), withText("Submit"),
        withParent(allOf(withId(R.id.activity_main),
            withParent(withId(android.R.id.content)))),
        isDisplayed()));
appCompatButton2.perform(click());

ViewInteraction appCompatButton3 = onView(
    allOf(withId(R.id.my_courses), withText("MY COURSES"),
isDisplayed()));
appCompatButton3.perform(click());

ViewInteraction appCompatButton4 = onView(
    allOf(withId(R.id.search_courses), withText("SEARCH"),
isDisplayed()));
appCompatButton4.perform(click());

ViewInteraction tableLayout = onView(
    childAtPosition(
        withId(R.id.cs_list),
        0));
tableLayout.perform(scrollTo(), click());

ViewInteraction appCompatButton5 = onView(
    allOf(withId(R.id.book_course)));
appCompatButton5.perform(scrollTo(), click());

// Added a sleep statement to match the app's execution delay.
// The recommended way to handle such scenarios is to use Espresso
idling resources:
// https://google.github.io/android-testing-support-
library/docs/espresso/idling-resource/index.html
try {
    Thread.sleep(5000);
} catch (InterruptedException e) {
    e.printStackTrace();
}

ViewInteraction appCompatButton6 = onView(
    allOf(withId(R.id.my_courses), withText("MY COURSES"),
isDisplayed()));
appCompatButton6.perform(click());

```

```

        // Added a sleep statement to match the app's execution delay.
        // The recommended way to handle such scenarios is to use Espresso
        idling resources:
        // https://google.github.io/android-testing-support-
        library/docs/espresso/idling-resource/index.html
        try {
            Thread.sleep(3867);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        ViewInteraction appCompatButton7 = onView(
            allOf(withId(R.id.logout), withText("LOGOUT"), isDisplayed()));
        appCompatButton7.perform(click());

    }

    private static Matcher<View> childAtPosition(
        final Matcher<View> parentMatcher, final int position) {

        return new TypeSafeMatcher<View>() {
            @Override
            public void describeTo(Description description) {
                description.appendText("Child at position " + position + " in
parent ");
                parentMatcher.describeTo(description);
            }

            @Override
            public boolean matchesSafely(View view) {
                ViewParent parent = view.getParent();
                return parent instanceof ViewGroup &&
parentMatcher.matches(parent)
                    && view.equals(((ViewGroup)
parent).getChildAt(position));
            }
        };
    }
}

```

### unbookCourseTest.java

```

package squirrel.fdb2;
import android.support.test.espresso.ViewInteraction;
import android.support.test.rule.ActivityTestRule;
import android.support.test.runner.AndroidJUnit4;
import android.test.suitebuilder.annotation.LargeTest;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewParent;

import org.hamcrest.Description;
import org.hamcrest.Matcher;
import org.hamcrest.TypeSafeMatcher;
import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

import static android.support.test.espresso.Espresso.onView;
import static android.support.test.espresso.action.ViewActions.click;
import static
android.support.test.espresso.action.ViewActions.closeSoftKeyboard;
import static android.support.test.espresso.action.ViewActions.replaceText;

```



```

import static android.support.test.espresso.action.ViewActions.scrollTo;
import static android.support.test.espresso.matcher.ViewMatchers.isDisplayed;
import static android.support.test.espresso.matcher.ViewMatchers.withId;
import static android.support.test.espresso.matcher.ViewMatchers.withParent;
import static android.support.test.espresso.matcher.ViewMatchers.withText;
import static org.hamcrest.Matchers.allOf;

@LargeTest
@RunWith(AndroidJUnit4.class)
public class searchCourseTest {

    @Rule
    public ActivityTestRule<MainActivity> mActivityTestRule = new
ActivityTestRule<>(MainActivity.class);

    @Test
    public void searchCourseTest() {
        ViewInteraction appCompatButton = onView(
            allOf( withText("Login"),
                isDisplayed()));
        appCompatButton.perform(click());

        ViewInteraction appCompatEditText = onView(
            allOf(withId(R.id.un),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatEditText.perform(click());

        ViewInteraction appCompatEditText2 = onView(
            allOf(withId(R.id.un),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatEditText2.perform(replaceText("admin"), closeSoftKeyboard());

        ViewInteraction appCompatEditText3 = onView(
            allOf(withId(R.id.pw),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatEditText3.perform(replaceText("admin"), closeSoftKeyboard());

        ViewInteraction appCompatButton2 = onView(
            allOf(withId(R.id.logbutton), withText("Submit"),
                withParent(allOf(withId(R.id.activity_main),
                    withParent(withId(android.R.id.content)))),
                isDisplayed()));
        appCompatButton2.perform(click());

        ViewInteraction appCompatButton3 = onView(
            allOf(withId(R.id.my_courses), withText("MY COURSES"),
isDisplayed()));
        appCompatButton3.perform(click());

        ViewInteraction tableLayout = onView(
            allOf(childAtPosition(
                withId(R.id.mc_list),
                1),
                isDisplayed()));
        tableLayout.perform(click());

        ViewInteraction appCompatButton4 = onView(
            allOf(withId(R.id.search_courses), withText("SEARCH"),

```



```

isDisplayed()));
    appCompatButton4.perform(click());

    ViewInteraction appCompatEditText4 = onView(
        withId(R.id.cs_field));
    appCompatEditText4.perform(scrollTo(), replaceText("che"),
closeSoftKeyboard());

    ViewInteraction tableLayout2 = onView(
        childAtPosition(
            withId(R.id.cs_list),
            0));
    tableLayout2.perform(scrollTo(), click());

    ViewInteraction appCompatButton5 = onView(
        allOf(withId(R.id.logout), withText("LOGOUT"), isDisplayed()));
    appCompatButton5.perform(click());

}

private static Matcher<View> childAtPosition(
    final Matcher<View> parentMatcher, final int position) {

    return new TypeSafeMatcher<View>() {
        @Override
        public void describeTo(Description description) {
            description.appendText("Child at position " + position + " in
parent ");
            parentMatcher.describeTo(description);
        }

        @Override
        public boolean matchesSafely(View view) {
            ViewParent parent = view.getParent();
            return parent instanceof ViewGroup &&
parentMatcher.matches(parent)
                && view.equals(((ViewGroup)
parent).getChildAt(position));
        }
    };
}
}

```

## UnitTestsMain.java

```

package squirrel.fdb2;
import android.content.Intent;
import android.graphics.PorterDuff;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.text.TextUtils;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

```

```

public class UnitTestsMain extends MainActivity{

    private boolean testFailed = false;
    private List<String> errorList = new ArrayList<String>();

    public void getSetUsnTest(){
        String usnExample = "Example Jones";
        setUsername(usnExample);
        if (!getUserName().equals(usnExample)){
            testFailed = true;
            errorList.add("Set/Get Username Failed");
        }
    }

    public void getSetMyCoursesTest(){
        //JSONArray jsonArray = new

        String jData =
        "[{\"id\":1,\"title\":\"Shshs\",\"description\":\"Shshs\",\"capacity\":112,\"duration\":33,\"prerequisiteID\":0},
        {\"id\":2,\"title\":\"hfhf\",\"description\":\"rkrkr\",\"capacity\":14,\"duration\":33,\"prerequisiteID\":0}]";
        Log.e("MyCourse", jData);
        JSONObject jsonObject = null;
        JSONArray newJArray = null;
        try {
            newJArray = new JSONArray(jData);
        } catch (JSONException e) {
            e.printStackTrace();
        }

        MainActivity.setMyCourses(newJArray);
        if (!
MainActivity.getMyCourses().toString().equals(newJArray.toString())){
            testFailed = true;
            errorList.add("Set/Get MyCourses Failed.");
        }
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.act_bar, menu);
        for(int i = 0; i < menu.size(); i++){
            Drawable drawable = menu.getItem(i).getIcon();
            if(drawable != null) {
                drawable.mutate();

drawable.setColorFilter(getResources().getColor(R.color.colorPrimary),
PorterDuff.Mode.SRC_ATOP);
            }
        }
        return true;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Toolbar bar = (Toolbar) findViewById(R.id.act_bar);
        setSupportActionBar(bar);
        //bar.setBackgroundResource(R.drawable.fdm);
        getSupportActionBar().setDisplayShowTitleEnabled(false);
    }
}

```

```
getSetUsnTest();
getSetMyCoursesTest();

    if (testFailed) {
        Log.e("Test Results: ", "Test was failed");
        Log.e("Test Results", errorList.toString());
    }

    else{
        Log.d("Test Results: ", "Test completed successfully.");
    }

}
}
```