Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Факультет Программной инженерии и компьютерной техники

**Лабораторная работа №3**

по дисциплине

«Тестирование программного обеспечения»

**Выполнил**

Ореховский А.,

группа P3317

**Преподаватель**

Исаев И. В.

Санкт-Петербург

2020

Оглавление

[Цель 3](#_Toc54003515)

[Задачи 3](#_Toc54003516)

[Архитектура Appium 3](#_Toc54003517)

[Требования 3](#_Toc54003518)

[Тестовые сценарии 3](#_Toc54003519)

[Листинг тестов 4](#_Toc54003520)

[Найденные баги 7](#_Toc54003521)

[Листинг кода 7](#_Toc54003522)

[Выводы 14](#_Toc54003523)

# Цель

Разработка мобильного приложения. Приложение, заботящиеся о здоровье. Представляет собой приложения с возможностью консультации врача, заказа лекарств, постановки диагноза и другое.

# Задачи

* Сформировать требования к приложению (не менее 20)
* Реализовать данные требования
* Покрыть UI тестами с помощью Appium

# Архитектура Appium

* Appium это HTTP сервер написанный на node.js
* Клиент общается с сервером с использованием сессии, где ключевые элементы передаются при помощи JSON. Общение осуществляется при помощи JSON Wire Protocol
* Сервер различает различные платформы (например iOS или Android) при помощи так называемых аргументов desiredCapabilities
* Далее Appium сервер делегирует процесс UI Automator-ам
* Затем результат сессии отправляется серверу и далее клиенту

# Требования

1. Приложение должно поддерживать вход при помощи почты и пароля.
2. Приложение должно поддерживать регистрацию в системы при помощи почты и пароля.
3. Приложение должно поддерживать возможность восстановления пароля.
4. При регистрации необходимо указывать ФИО.
5. В случае, если пользователь оставит поле с ФИО пустым, необходимо сообщить ему об этом.
6. В случае, если пароль является не надежным (состоит из менее 6 символов или не содержит и символы, и цифры), необходимо уведомить пользователя.
7. В случае, если вводимая почта пользователя не является корректной, необходимо уведомить об этом пользователя.
8. В случае успешной авторизации необходимо уведомлять об этом пользователя
9. Приложение должно предоставлять возможность добавлять товары в корзину.
10. При добавлении товара в корзину необходимо уведомлять пользователя об успешности данной операции.
11. Если количество зарезервированных товаров достигнет количества доступных, необходимо заблокировать возможность добавления этого же товара в корзину.
12. Необходимо предоставить возможность просмотра корзины.
13. Необходимо предоставить возможность подтверждения резервации товаров.
14. При успешной резервации корзина должна очищаться.
15. В случае, если пользователь не подтвердил резервацию, необходимо сохранять содержимое корзины.

# Тестовые сценарии

1. Приложение должно авторизовывать пользователя
2. Приложение не должно авторизовывать пользователя с неверным паролем
3. Добавление предмета в корзину
4. Открытие корзины
5. Резервирование пустой корзины
6. Резервирование не пустой корзины
7. Проверка того, что корзина пуста после резервирования
8. Проверка того, что возможно зарезервировать количество предметов, не превышающее доступное количество

# Листинг тестов

**public class** tests {  
 **private** AndroidDriver **driver**;  
 **private static final int *TIME\_LIMIT*** = 10;  
 **private static** WebDriverWait *wait*;  
  
 @BeforeEach  
 **public void** setUp() {  
 DesiredCapabilities dc = **new** DesiredCapabilities();  
 dc.setCapability(MobileCapabilityType.***DEVICE\_NAME***, **"emulator-5554"**);  
 dc.setCapability(**"platformName"**, **"android"**);  
 dc.setCapability(**"appPackage"**, **"ru.ifmo.lab3"**);  
 dc.setCapability(**"appActivity"**, **".MainActivity"**);  
  
 **try** {  
 **driver** = **new** AndroidDriver<AndroidElement>(**new** URL(**"http://127.0.0.1:4723/wd/hub"**), dc);  
 } **catch** (MalformedURLException e) {  
 e.printStackTrace();  
 }  
  
 *wait* = **new** WebDriverWait(**driver**, ***TIME\_LIMIT***);  
 }  
  
 @Test  
 **public void** shouldAuthorize() **throws** InterruptedException {  
  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/email"**)));  
 MobileElement el1 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/email"**);  
 el1.sendKeys(**"anton.o.0619@gmail.com"**);  
 MobileElement el2 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/button\_next"**);  
 el2.click();  
  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/password"**)));  
 Thread.*sleep*(1000);  
 MobileElement el3 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/password"**);  
 el3.sendKeys(**"anton123"**);  
 MobileElement el4 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/button\_done"**);  
 el4.click();  
 }  
  
 @Test  
 **public void** shouldNotLetAuthorizeWithWrongPassword() {  
  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/email"**)));  
 MobileElement el1 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/email"**);  
 el1.sendKeys(**"anton.o.0619@gmail.com"**);  
 MobileElement el2 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/button\_next"**);  
 el2.click();  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/password"**)));  
 MobileElement el3 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/password"**);  
 el3.sendKeys(**"blahblah123"**);  
 MobileElement el4 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/button\_done"**);  
 el4.click();  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/textinput\_error"**)));  
 Assert.*assertEquals*(**driver**.findElementById(**"ru.ifmo.lab3:id/textinput\_error"**).getText(), **"Incorrect password."**);  
 }  
  
 @Test  
 **public void** shouldAddItemToCart() **throws** InterruptedException {  
 shouldAuthorize();  
  
 Thread.*sleep*(7000);  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*(**"/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.view.ViewGroup/android.widget.FrameLayout[2]/android.widget.RelativeLayout/android.widget.ListView/android.widget.RelativeLayout[1]/android.widget.ImageButton"**)));  
 MobileElement el6 = (MobileElement) **driver**.findElementByXPath(**"/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.view.ViewGroup/android.widget.FrameLayout[2]/android.widget.RelativeLayout/android.widget.ListView/android.widget.RelativeLayout[1]/android.widget.ImageButton"**);  
 el6.click();  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/snackbar\_text"**)));  
 String snackbarText = **driver**.findElementById(**"ru.ifmo.lab3:id/snackbar\_text"**).getText();  
 Assert.*assertEquals*(snackbarText, **"1 piece of Aspirin has been added to the cart"**);  
 }  
  
 @Test  
 **public void** shouldOpenAnEmptyCart() **throws** InterruptedException {  
 shouldAuthorize();  
  
 Thread.*sleep*(7000);  
 *//wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("ru.ifmo.lab3:id/btnCart")));* MobileElement el1 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/btnCart"**);  
 el1.click();  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/title"**)));  
 String title = **driver**.findElementById(**"ru.ifmo.lab3:id/title"**).getText();  
 String body = **driver**.findElementById(**"ru.ifmo.lab3:id/medicine\_name"**).getText();  
  
 Assert.*assertEquals*(title, **"Confirm reservation"**);  
 Assert.*assertEquals*(body, **"Nothing to confirm"**);  
 }  
  
 @Test  
 **public void** shouldReserveNothing() **throws** InterruptedException {  
 shouldOpenAnEmptyCart();  
 MobileElement el2 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/btnReserve"**);  
 el2.click();  
 Thread.*sleep*(7000);  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/btnCart"**)));  
 }  
  
 @Test  
 **public void** shouldReserveNotEmptyCart() **throws** InterruptedException {  
 shouldAddItemToCart();  
  
 Thread.*sleep*(7000);  
 MobileElement el1 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/btnCart"**);  
 el1.click();  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/title"**)));  
 String title = **driver**.findElementById(**"ru.ifmo.lab3:id/title"**).getText();  
 String body = **driver**.findElementById(**"ru.ifmo.lab3:id/medicine\_name"**).getText();  
  
 Assert.*assertEquals*(title, **"Confirm reservation"**);  
 Assert.*assertEquals*(body, **"Aspirin: 1"**);  
  
 MobileElement el2 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/btnReserve"**);  
 el2.click();  
 Thread.*sleep*(7000);  
 *//wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("ru.ifmo.lab3:id/btnCart")));* Assert.*assertNotNull*(**driver**.findElementById(**"ru.ifmo.lab3:id/btnCart"**));  
 }  
  
 @Test  
 **public void** shouldHaveAnEmptyCartAfterReservation() **throws** InterruptedException {  
 shouldReserveNotEmptyCart();  
  
 MobileElement el1 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/btnCart"**);  
 el1.click();  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/title"**)));  
 String title = **driver**.findElementById(**"ru.ifmo.lab3:id/title"**).getText();  
 String body = **driver**.findElementById(**"ru.ifmo.lab3:id/medicine\_name"**).getText();  
  
 Assert.*assertEquals*(title, **"Confirm reservation"**);  
 Assert.*assertEquals*(body, **"Nothing to confirm"**);  
 }  
  
 @Test  
 **public void** shouldNotAllowToReserveMoreThanAvailable() **throws** InterruptedException {  
 shouldAuthorize();  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*xpath*(**"/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.view.ViewGroup/android.widget.FrameLayout[2]/android.widget.RelativeLayout/android.widget.ListView/android.widget.RelativeLayout[6]/android.widget.ImageButton"**)));  
 MobileElement el6 = (MobileElement) **driver**.findElementByXPath(**"/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.view.ViewGroup/android.widget.FrameLayout[2]/android.widget.RelativeLayout/android.widget.ListView/android.widget.RelativeLayout[6]/android.widget.ImageButton"**);  
 el6.click();  
 el6.click();  
 el6.click();  
 el6.click();  
 el6.click();  
 Thread.*sleep*(7000);  
 MobileElement el1 = (MobileElement) **driver**.findElementById(**"ru.ifmo.lab3:id/btnCart"**);  
 el1.click();  
 *wait*.until(ExpectedConditions.*visibilityOfElementLocated*(By.*id*(**"ru.ifmo.lab3:id/title"**)));  
 String title = **driver**.findElementById(**"ru.ifmo.lab3:id/title"**).getText();  
 String body = **driver**.findElementById(**"ru.ifmo.lab3:id/medicine\_name"**).getText();  
  
 Assert.*assertEquals*(title, **"Confirm reservation"**);  
 Assert.*assertEquals*(body, **"Acetaminophen: 2"**);  
 }  
  
 @AfterEach  
 **public void** teardown() {  
 **driver**.quit();  
 }  
}

# Найденные баги

В ходе написания тестов мною был найден один лишь баг. Он заключался в том, что я не обновлял корзину после резервирования предметов.

# Листинг кода

*MainActivity.java*

**public class** MainActivity **extends** AppCompatActivity {  
  
 **private static int** *SIGN\_IN\_CODE* = 1;  
 **private** RelativeLayout **activity\_main**;  
 **private** Map<String, Integer> **cart**;  
 **private final** String **RESERVATION\_RESULT** = **"ReservationResult"**;  
 **private final int RESERVED** = 1;  
  
  
 @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, @Nullable Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
  
 **if**(requestCode == *SIGN\_IN\_CODE*) {  
 **if**(resultCode == ***RESULT\_OK***) {  
 Snackbar.*make*(**activity\_main**, **"You are now authorized"**, Snackbar.***LENGTH\_LONG***).show();  
 displayAllMedicines();  
 } **else** {  
 Snackbar.*make*(**activity\_main**, **"You are now authorized"**, Snackbar.***LENGTH\_LONG***).show();  
 finish();  
 }  
 }  
 }  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 Intent intent = getIntent();  
 **int** extra = intent.getIntExtra(**RESERVATION\_RESULT**, 0);  
  
 **activity\_main** = findViewById(R.id.***activity\_main***);  
  
 *// User is authorized* **if** (FirebaseAuth.*getInstance*().getCurrentUser() == **null**) {  
 startActivityForResult(AuthUI.*getInstance*().createSignInIntentBuilder().build(), *SIGN\_IN\_CODE*);  
 } **else** {  
 **if** (extra != **RESERVED**) {  
 Snackbar.*make*(**activity\_main**, **"You are now authorized"**, Snackbar.***LENGTH\_SHORT***).show();  
 }  
 displayAllMedicines();  
 }  
 **cart**= **new** HashMap<>();  
 }  
  
 **public void** showCart(View v) {  
 Intent intent = **new** Intent(**this**, CartActivity.**class**);  
 **if** (!**cart**.isEmpty()) {  
 JSONObject json = **new** JSONObject(**cart**);  
 System.***out***.println(json.toString());  
 intent.putExtra(**"cart"**, json.toString());  
 }  
 startActivity(intent);  
 }  
  
 **public void** addToCart(View v) {  
 View parentView = (View) v.getParent();  
 String medicineName = ((TextView)parentView.findViewById(R.id.***medicine\_name***)).getText().toString();  
 **int** available = Integer.*parseInt*(((TextView)parentView.findViewById(R.id.***medicine\_count***)).getText().toString());  
 **int** reservedCount = **cart**.containsKey(medicineName) ? **cart**.get(medicineName) + 1: 1;  
  
 **if** (reservedCount == available) {  
 v.setEnabled(**false**);  
 }  
  
 **cart**.put(medicineName, reservedCount);  
 Snackbar.*make*(**activity\_main**, **"1 piece of "** + medicineName + **" has been added to the cart"**, Snackbar.***LENGTH\_SHORT***).show();  
 }  
  
 *// Display all medicines on create* **private void** displayAllMedicines() {  
 ListView listOfMedicines = findViewById(R.id.***list\_of\_medicines***);  
 FirebaseListAdapter<Medicine> adapter = **new** FirebaseListAdapter<Medicine>(**this**, Medicine.**class**, R.layout.***list\_item***,  
 FirebaseDatabase.*getInstance*().getReference()) {  
 @Override  
 **protected void** populateView(View v, Medicine model, **int** position) {  
 TextView medicine\_name, medicine\_category, medicine\_count;  
 medicine\_name = (TextView)v.findViewById(R.id.***medicine\_name***);  
 medicine\_category = (TextView)v.findViewById(R.id.***medicine\_category***);  
 medicine\_count = (TextView)v.findViewById(R.id.***medicine\_count***);  
  
 medicine\_name.setText(model.getMedicineName());  
 medicine\_category.setText(model.getCategory());  
 String count = String.*valueOf*(model.getCount());  
 medicine\_count.setText(count);  
  
 }  
 };  
 listOfMedicines.setAdapter(adapter);  
 }  
}

*activity\_main.xml*

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"  
 android:id="@+id/activity\_main"**>  
  
 <**com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/btnCart"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:clickable="true"  
 android:focusable="true"  
 android:src="@drawable/ic\_cart"  
 android:tint="@android:color/black"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_margin="15sp"  
 app:fabSize="normal"  
 android:onClick="showCart"**/>  
  
 <**ListView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_above="@id/btnCart"  
  
 android:id="@+id/list\_of\_medicines"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
  
 android:divider="@android:color/transparent"  
 android:dividerHeight="16dp"  
 android:layout\_marginBottom="16dp"  
 android:layout\_margin="15sp"**/>  
</**RelativeLayout**>

*list\_item.xml*

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"  
 android:id="@+id/medicine\_name"  
 android:textStyle="normal|bold"  
 android:textSize="25sp"  
 android:textColor="@color/colorPrimary"**/>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:id="@+id/medicine\_category"  
 android:layout\_below="@id/medicine\_name"  
 android:textColor="@color/colorPrimaryDark"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_toEndOf="@id/medicine\_name"  
 android:id="@+id/medicine\_count"  
 android:layout\_marginStart="5dp"  
 android:textSize="40sp"  
 android:textColor="@color/colorCount"**/>  
  
 <**com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:clickable="true"  
 android:focusable="true"  
 android:src="@drawable/ic\_plus"  
 android:tint="@android:color/black"  
 android:layout\_centerVertical="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_margin="5sp"  
 app:fabSize="mini"  
 android:onClick="addToCart"**/>  
  
</**RelativeLayout**>

*CartActivity.java*

**public class** CartActivity **extends** AppCompatActivity {  
  
 **private** ArrayList<String> **cart** = **new** ArrayList<>();  
 **private final** String **RESERVATION\_RESULT** = **"ReservationResult"**;  
 **private final int RESERVED** = 1;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_cart***);  
  
 Intent intent = getIntent();  
 String extra = intent.getStringExtra(**"cart"**);  
 **if** (extra != **null**) {  
 **try** {  
 JSONObject json = **new** JSONObject(extra);  
 System.***out***.println(**"extra: "** + extra);  
 Iterator<String> medicineNames = json.keys();  
  
 **while** (medicineNames.hasNext()) {  
 String key = medicineNames.next();  
 **cart**.add(key + **": "** + json.getInt(key));  
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 displayCart();  
  
 }  
  
 **private void** displayCart() {  
 ListView listOfMedicines = findViewById(R.id.***list\_of\_medicines***);  
 **if** (**cart**.isEmpty()) **cart**.add(**"Nothing to confirm"**);  
 ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, R.layout.***cart\_item***, **cart**);  
 listOfMedicines.setAdapter(adapter);  
 }  
  
 **public void** confirmReservation(View v) {  
 **for**(**int** i = 0; i < **cart**.size(); i++) {  
 String[] medicine = **cart**.get(i).split(**": "**);  
 String name = medicine[0];  
 **int** count = Integer.*parseInt*(medicine[1]);  
 System.***out***.println(**"confirm "** + name + **" "** + count);  
  
 */\*  
 FirebaseDatabase.getInstance().getReference().orderByChild("medicineName").equalTo(name).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 //Iterable<DataSnapshot> list = snapshot.getValue();  
 System.out.println("from listener: " + snapshot.getValue());  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });\*/* }  
  
 Intent intent = **new** Intent(**this**, MainActivity.**class**);  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP***);  
 intent.putExtra(**RESERVATION\_RESULT**, **RESERVED**);  
 startActivity(intent);  
 }  
}

*activity\_cart.xml*

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".CartActivity"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"  
 android:id="@+id/title"  
 android:textStyle="normal|bold"  
 android:textSize="40sp"  
 android:textColor="@color/colorPrimaryDark"  
 android:text="@string/confirm\_reservation"** />  
  
 <**com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/btnReserve"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:clickable="true"  
 android:focusable="true"  
 android:src="@drawable/ic\_reserve"  
 android:tint="@android:color/black"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_margin="15sp"  
 app:fabSize="normal"  
 android:onClick="confirmReservation"**/>  
  
 <**ListView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_above="@id/btnReserve"  
  
 android:id="@+id/list\_of\_medicines"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@id/title"  
  
 android:divider="@android:color/transparent"  
 android:dividerHeight="16dp"  
 android:layout\_marginBottom="16dp"  
 android:layout\_margin="15sp"**/>  
</**RelativeLayout**>

*cart\_item.xml*

*<?***xml version="1.0" encoding="utf-8"***?>*<**TextView  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"  
 android:id="@+id/medicine\_name"  
 android:textSize="25sp"  
 android:textColor="@color/colorPrimary"**/>

*Medicine.java*

**public class** Medicine {  
 **public** String **medicineName**;  
 **public** String **category**;  
 **private int count**;  
  
 **public** Medicine() {}  
 **public** Medicine(String medicineName, String category, **int** count) {  
 **this**.**medicineName** = medicineName;  
 **this**.**category** = category;  
 **this**.**count** = count;  
 }  
  
 **public** String getMedicineName() {  
 **return medicineName**;  
 }  
  
 **public void** setMedicineName(String medicineName) {  
 **this**.**medicineName** = medicineName;  
 }  
  
 **public** String getCategory() {  
 **return category**;  
 }  
  
 **public void** setCategory(String category) {  
 **this**.**category** = category;  
 }  
  
 **public int** getCount() {  
 **return count**;  
 }  
  
 **public void** setCount(**int** count) {  
 **this**.**count** = count;  
 }  
}

# Выводы

Так как данная ЛР — это мой первый опыт Android разработки, то в силу ограниченного количества времени, функционал разработанного приложения весьма скуден. Тем не менее, это не помешало мне протестировать его в полном объеме. Таким образом, я, как мне кажется, понял основы тестирования Android приложений.

Тестирование было произведено при помощи Appium сервера. Это мощный инструмент, который помогает автоматизировать ui-тестирование мобильных приложений, а также может играть роль «инспектора», как например режим разработчика в клиентских приложениях, что помогает с легкостью узнать id или xpath элемента в необходимый этап тестирования приложения. Еще одна особенность Appium состоит в возможности записи действий, что сокращает количество «не интересного кода» написанного ручками.

В целом ui-тестирование мобильных приложений схоже с тестированием клиентского интерфейса в веб приложениях. Чего только стоит использование того же подхода, что и selenium сервер. Тестирование мобильного приложения можно производить при помощи js и распространенных там callback-ов или при помощи более удобной для меня Java.