федеральное государственное автономное образовательное учреждение высшего образования

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

**Факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №3**

**ПО ДИСЦИПЛИНЕ «ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»**

**Вариант 19**

Выполнил: Чудаков М.И.

Группа: P3319

Преподаватель: Исаев И.В.

Санкт-Петербург

2019

**Цель**

Изучить методику функционального тестирования Android приложения, используя Appium.

**Задачи**

1. Необходимо взять приложение согласно варианту, установить эмулятор Genymotion (использовать андроид 6.0 и выше);
2. Настроить Appium (http://appium.io) и написать 15 тестовых сценариевЖ
3. Сценарий установки виджета на экран обязателен;
4. Проверка регистрации обязательна.

**Appium**

Appium – это инструмент автоматизации с открытым исходным кодом для запуска скриптов и тестирования нативных приложений, мобильных веб-приложений и гибридных приложений на Android и iOS с помощью WebDriver.

**Genymotion**

Genymotion представляет собой полнофункциональный эмулятор Android, который сочетает простоту использования с передовой производительностью 3D графики, предоставляя пользователю практически неограниченный возможности работы с мобильной ОС.

**Примеры тестовые сценариев**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 1 | | Create account test with phone number | | Проверка функционирование страниц регистрации приложения. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “sign up” | |  | | Приложение отображает страницу с регистрацией через номер телефона | Приложение отображает страницу с регистрацией через номер телефона |
| 2 | Пользователь вводит имя | | имя = 1 | | Приложение отображает введенное имя в текстовом поле | Приложение отображает введенное имя в текстовом поле |
| 3 | Пользователь вводит номер мобильного телефона | | номер = 1 | | Приложение выводит сообщение с предупреждением о некорректном номере | Приложение выводит сообщение с предупреждением о некорректном номере |
| 4 | Пользователь вводит существующий номер мобильного телефона | |  | | Приложение отображает введенный номер в текстовом поле | Приложение отображает введенный номер в текстовом поле |
| 5 | Пользователь нажимает кнопку “регистрация” | |  | | Приложение отображает сообщением с информацией о том, что данный номер уже зарегистрирован | Приложение отображает сообщением с информацией о том, что данный номер уже зарегистрирован |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 2 | | Create account test with email address | | Проверка функционирование страниц регистрации приложения. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “sign up” | |  | | Приложение отображает страницу с регистрацией через номер телефона | Приложение отображает страницу с регистрацией через номер телефона |
| 2 | Пользователь вводит имя | | имя = 1 | | Приложение отображает введенное имя в текстовом поле | Приложение отображает введенное имя в текстовом поле |
| 3 | Пользователь нажимает кнопку “регистрация через email” | |  | | Приложение отображает страницу с регистрацией через почту | Приложение отображает страницу с регистрацией через почту |
| 4 | Пользователь вводит адрес электронной почты | | адрес = 1 | | Приложение выводит сообщение с предупреждением о некорректном адресе электронной почты | Приложение выводит сообщение с предупреждением о некорректном адресе электронной почты |
| 5 | Пользователь вводит адрес электронной почты | |  | | Приложение отображает введенный адрес электронной почты в текстовом поле |  |
| 5 | Пользователь нажимает кнопку “регистрация” | |  | | Приложение отображает сообщением с информацией о том, что данный адрес уже зарегистрирован | Приложение отображает сообщением с информацией о том, что данный адрес уже зарегистрирован |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 3 | | Login with exist account test | | Проверка функционирование страницы авторизации. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “reset password” | |  | | Приложение отображает страницу со сбросом пароля | Приложение отображает страницу со сбросом пароля |
| 2 | Пользователь нажимает кнопку “navigate up” | |  | | Приложение отображает страницу с авторизацией | Приложение отображает страницу с авторизацией |
| 3 | Пользователь вводит логин | |  | | Приложение отображает введенный логин | Приложение отображает введенный логин |
| 4 | Пользователь вводит пароль | |  | | Приложение отображает введенный пароль | Приложение отображает введенный пароль |
| 5 | Пользователь нажимает кнопку “login” | |  | | Приложение авторизует пользователя и отображает главную страницу | Приложение авторизует пользователя и отображает главную страницу |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 4 | | Login with un exist account test | | Проверка функционирование страницы авторизации. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “reset password” | |  | | Приложение отображает страницу со сбросом пароля | Приложение отображает страницу со сбросом пароля |
| 2 | Пользователь нажимает кнопку “navigate up” | |  | | Приложение отображает страницу с авторизацией | Приложение отображает страницу с авторизацией |
| 3 | Пользователь вводит логин | | логин = 1 | | Приложение отображает введенный логин | Приложение отображает введенный логин |
| 4 | Пользователь вводит пароль | | пароль = 1 | | Приложение отображает введенный пароль | Приложение отображает введенный пароль |
| 5 | Пользователь нажимает кнопку “login” | |  | | Приложение выводит сообщение о неверном логине и пароле | Приложение выводит сообщение о неверном логине и пароле |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 5 | | Home activity test | | Проверка функционирование стартовой страницы. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Авторизированный пользователь запускает приложение | |  | | Приложение отображает диалоговое окно | Приложение отображает диалоговое окно |
| 2 | Пользователь нажимает кнопку “ок” | |  | | Приложение отображает диалоговое окно с подтверждением геолокации | Приложение отображает диалоговое окно с подтверждением геолокации |
| 3 | Пользователь нажимает кнопку “ок” | |  | | Приложение отображает стартовую страницу | Приложение отображает стартовую страницу |
| 4 | Пользователь нажимает кнопку “curation action” | |  | | Приложение отображает выпадающий список | Приложение отображает выпадающий список |
| 5 | Пользователь нажимает кнопку кликает по экрану, по заданным координатам | | x = 400  y = 150 | | Приложение закрывает выпадающий список | Приложение закрывает выпадающий список |
| 6 | Пользователь нажимает кнопку “timeline switch” | |  | | Приложение отображает выпадающий список | Приложение отображает выпадающий список |
| 7 | Пользователь нажимает кнопку кликает по экрану, по заданным координатам | | x = 400  y = 150 | | Приложение закрывает выпадающий список | Приложение закрывает выпадающий список |
| 8 | Пользователь нажимает кнопку “image ” | |  | | Приложение отображает выпадающий список c информацией об аккаунте | Приложение отображает выпадающий список c информацией об аккаунте |
| 9 | Пользователь нажимает кнопку кликает по экрану, по заданным координатам | | x = 700  y = 400 | | Приложение закрывает выпадающий список c информацией об аккаунте | Приложение закрывает выпадающий список c информацией об аккаунте |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 6 | | Create twit activity test | | Проверка функционирование стартовой страницы. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “create twit” | |  | | Приложение отображает страницу с созданием нового твита | Приложение отображает страницу с созданием нового твита |
| 2 | Пользователь нажимает кнопку “imag view” | |  | | Приложение отображает текстовое поле | Приложение отображает текстовое поле |
| 3 | Пользователь вводит текст | | qwerty | | Приложение отображает введенный текст | Приложение отображает введенный текст |
| 4 | Пользователь нажимает кнопку “gif” | |  | | Приложение отображает страницу с разделами gif-изображений | Приложение отображает страницу с разделами gif-изображений |
| 5 | Пользователь выбирает раздел | |  | | Приложение отображает страницу с выбранным разделом | Приложение отображает страницу с выбранным разделом |
| 6 | Пользователь выбирает gif | |  | | Приложение отображает выбранное gif-изображение в блоке с созданием нового твита | Приложение отображает выбранное gif-изображение в блоке с созданием нового твита |
| 7 | Пользователь нажимает кнопку создать твит | |  | | Приложение отображает страницу твитов с новым твитом | Приложение отображает страницу твитов с новым твитом |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 7 | | Search info test | | Проверка функционирование страницы поиска. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “найти” | |  | | Приложение отображает страницу с поиском | Приложение отображает страницу с поиском |
| 2 | Пользователь нажимает кнопку “search” | |  | | Приложение отображает текстовое поле | Приложение отображает текстовое поле |
| 3 | Пользователь вводит текст | | a | | Приложение отображает результат поиска | Приложение отображает результат поиска |
| 4 | Пользователь нажимает выбирает первый полученный результат | |  | | Приложение отображает страницу с выбранным названием | Приложение отображает страницу с выбранным названием |
| 5 | Пользователь нажимает кнопку “настройки” | |  | | Приложение отображает выпадающий список с настройками | Приложение отображает выпадающий список с настройками |
| 6 | Пользователь нажимает кнопку “search settings” | |  | | Приложение отображает страницу с настройками поиска | Приложение отображает страницу с настройками поиска |
| 7 | Пользователь нажимает кнопку создать “назад” | |  | | Приложение отображает страницу с результатом поиска | Приложение отображает страницу с результатом поиска |
| 8 | Пользователь нажимает кнопку создать “назад” | |  | | Приложение отображает главную страницу | Приложение отображает главную страницу |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 8 | | Edit notifications test | | Проверка функционирование страницы уведомлений. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “notifications” | |  | | Приложение отображает страницу c уведомлениями | Приложение отображает страницу c уведомлениями |
| 2 | Пользователь нажимает кнопку “mentiosn” | |  | | Приложение отображает страницу c упоминаниями | Приложение отображает страницу c упоминаниями |
| 3 | Пользователь нажимает кнопку “настройки” | |  | | Приложение отображает страницу c настройками | Приложение отображает страницу c настройками |
| 4 | Пользователь нажимает кнопку “назад” | | пароль = 1 | | Приложение отображает страницу c упоминаниями | Приложение отображает страницу c упоминаниями |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 9 | | Send message test | | Проверка функционирование страницы с отправкой сообщений. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “сообщения” | |  | | Приложение отображает страницу c сообщениями | Приложение отображает страницу c сообщениями |
| 2 | Пользователь нажимает на первый диалог | |  | | Приложение отображает диалог | Приложение отображает диалог |
| 3 | Пользователь вводит сообщений | | qwerty | | Приложение отображает введенное сообщение | Приложение отображает введенное сообщение |
| 4 | Пользователь нажимает кнопку “отправить” | |  | | Приложение отображает отправленное сообщение | Приложение отображает отправленное сообщение |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 10 | | Create widget test | | Добавление виджета | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “назад” | |  | | Пользователь сворачивает приложение | Пользователь сворачивает приложение |
| 2 | Пользователь открывает список приложений | |  | | Отображение списка приложений | Отображение списка приложений |
| 3 | Пользователь переносит иконку приложения на главный экран | |  | | Отображение иконки приложения на главном экране | Отображение иконки приложения на главном экране |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 11 | | View profile test | | Проверка функционирование страницы с информацией о профиле. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “Профиль” | |  | | Приложение отображает страницу c информацией о профиле | Приложение отображает страницу c информацией о профиле |
| 2 | Пользователь нажимает кнопку “tweets” | |  | | Приложение отображает страницу c твитами | Приложение отображает страницу c твитами |
| 3 | Пользователь нажимает кнопку “tweets and replies” | |  | | Приложение отображает страницу c твитами и ответами | Приложение отображает страницу c твитами и ответами |
| 4 | Пользователь нажимает кнопку “media” | |  | | Приложение отображает страницу c медиа | Приложение отображает страницу c медиа |
| 5 | Пользователь нажимает кнопку “likes” | |  | | Приложение отображает страницу c лайками | Приложение отображает страницу c лайками |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 12 | | Create lists test | | Проверка функционирование страницы со списками. | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “Списки” | |  | | Приложение отображает страницу cо списками | Приложение отображает страницу cо списками |
| 2 | Пользователь нажимает кнопку “создать” | |  | | Приложение отображает страницу c созданием списка | Приложение отображает страницу c созданием списка |
| 3 | Пользователь вводит имя | | qwerty | | Приложение отображает веденное имя | Приложение отображает веденное имя |
| 4 | Пользователь вводит описание | | qwerty | | Приложение отображает введенное описание | Приложение отображает введенное описание |
| 5 | Пользователь нажимает кнопку “сохранить” | |  | | Приложение отображает сохраненный список | Приложение отображает сохраненный список |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 13 | | Delete all Bookmarks | | Удаление всех закладок | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “Закладки” | |  | | Приложение отображает страницу с закладками | Приложение отображает страницу cо списками |
| 2 | Пользователь нажимает кнопку “больше опцией” | |  | | Приложение отображает выпадающий список с опциями | Приложение отображает выпадающий список с опциями |
| 3 | Пользователь нажимает кнопку “удалить все” | |  | | Приложение отображает окно с подтверждением | Приложение отображает окно с подтверждением |
| 4 | Пользователь нажимает кнопку “подтвердить” | |  | | Приложение удаляет все закладки | Приложение удаляет все закладки |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 14 | | Open moments | | Открытие моментов | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “Burger” | |  | | Приложение выпадающий список | Приложение выпадающий список |
| 2 | Пользователь нажимает кнопку “Моменты” | |  | | Приложение отображает страницу с моментами | Приложение отображает страницу с моментами |
| 3 | Пользователь нажимает кнопку “Назад” | |  | | Главную страницу | Главную страницу |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Идентификатор** | | **Название** | | **Описание** | | |
| 15 | | Open moments | | Открытие моментов | | |
| **Шаг №** | **Описание** | | **Тестовые данные** | | **Ожидаемый результат** | **Фактический результат** |
| 1 | Пользователь нажимает кнопку “Burger” | |  | | Приложение выпадающий список | Приложение выпадающий список |
| 2 | Пользователь нажимает кнопку “Изменить тему” | |  | | Приложение меняет тему | Приложение меняет тему |

**Исходный код**

**package** ru.chudakov.pages  
  
**import** io.appium.java\_client.MobileElement  
**import** io.appium.java\_client.android.AndroidDriver  
**import** io.appium.java\_client.pagefactory.AppiumFieldDecorator  
**import** org.openqa.selenium.support.PageFactory  
  
**abstract class** AbstractPage(**private val driver**: AndroidDriver<MobileElement>) {  
  
 **init** {  
 initElements()  
 }  
  
 **private fun** initElements() {  
 PageFactory.initElements(AppiumFieldDecorator(**driver**), **this**)  
 }  
}

**package** ru.chudakov.pages  
  
**import** io.appium.java\_client.MobileElement  
**import** io.appium.java\_client.android.AndroidDriver  
**import** io.appium.java\_client.android.AndroidElement  
**import** io.appium.java\_client.pagefactory.AndroidFindBy  
  
**class** AuthorizationPage(driver: AndroidDriver<MobileElement>) : AbstractPage(driver) {  
  
 @AndroidFindBy(id = **"com.twitter.android:id/password\_reset"**)  
 **lateinit var resetPasswordButton**: AndroidElement  
  
 @AndroidFindBy(accessibility = **"Navigate up"**)  
 **lateinit var navigateUp**: AndroidElement  
  
 @AndroidFindBy(id = **"com.twitter.android:id/login\_identifier"**)  
 **lateinit var loginInput**: AndroidElement  
  
 @AndroidFindBy(id = **"com.twitter.android:id/login\_password"**)  
 **lateinit var passwordInput**: AndroidElement  
  
 @AndroidFindBy(id = **"com.twitter.android:id/login\_login"**)  
 **lateinit var loginButton**: AndroidElement  
}

**package** ru.chudakov.pages  
  
**import** io.appium.java\_client.MobileElement  
**import** io.appium.java\_client.android.AndroidDriver  
**import** io.appium.java\_client.android.AndroidElement  
**import** io.appium.java\_client.pagefactory.AndroidFindBy  
  
**class** HomePage(driver: AndroidDriver<MobileElement>) : AbstractPage(driver) {  
  
 @AndroidFindBy(xpath = **"//android.view.View[@content-desc=\"Home Tab\"]"**)  
 **lateinit var view**: AndroidElement  
  
 @AndroidFindBy(id = **"com.twitter.android:id/toolbar"**)  
 **lateinit var toolbar**: AndroidElement  
  
 @AndroidFindBy(id = **"com.twitter.android:id/tweet\_curation\_action"**)  
 **lateinit var tweetCurationAction**: AndroidElement  
  
 @AndroidFindBy(id = **"com.twitter.android:id/gripper"**)  
 **lateinit var gripper**: AndroidElement  
  
 @AndroidFindBy(id = **"com.twitter.android:id/toolbar\_timeline\_switch"**)  
 **lateinit var timelineSwitch**: AndroidElement  
  
 @AndroidFindBy(className = **"android.widget.ImageButton"**)  
 **lateinit var imageButton**: AndroidElement  
  
 @AndroidFindBy(id = **"com.twitter.android:id/drawer"**)  
 **lateinit var drawer**: AndroidElement  
}

**package** ru.chudakov  
  
**import** io.appium.java\_client.MobileElement  
**import** io.appium.java\_client.android.Activity  
**import** io.appium.java\_client.android.AndroidDriver  
**import** io.appium.java\_client.android.AndroidTouchAction  
**import** io.appium.java\_client.remote.MobileCapabilityType  
**import** io.appium.java\_client.touch.WaitOptions  
**import** io.appium.java\_client.touch.offset.PointOption  
**import** org.junit.jupiter.api.AfterAll  
**import** org.junit.jupiter.api.Test  
**import** org.junit.jupiter.api.TestInstance  
**import** org.openqa.selenium.By  
**import** org.openqa.selenium.remote.DesiredCapabilities  
**import** org.openqa.selenium.support.ui.WebDriverWait  
**import** ru.chudakov.pages.\*  
**import** java.net.URL  
**import** java.time.Duration.ofMillis  
**import** kotlin.test.assertFalse  
**import** kotlin.test.assertNotEquals  
  
@TestInstance(TestInstance.Lifecycle.**PER\_CLASS**)  
**class** TwitterTest {  
 **private val driver**: AndroidDriver<MobileElement>  
 **private val wait**: WebDriverWait  
 **private val action**: AndroidTouchAction  
  
 **init** {  
 **val** capabilities = DesiredCapabilities()  
  
 capabilities.setCapability(MobileCapabilityType.*PLATFORM\_NAME*, **"Android"**)  
 capabilities.setCapability(MobileCapabilityType.*DEVICE\_NAME*, **"Custom Phone"**)  
 capabilities.setCapability(MobileCapabilityType.*UDID*, **"192.168.147.105:5555"**)  
 capabilities.setCapability(MobileCapabilityType.*AUTOMATION\_NAME*, **"uiautomator2"**)  
 capabilities.setCapability(**"appPackage"**, **"com.twitter.android"**)  
 capabilities.setCapability(**"appActivity"**, **"com.twitter.app.main.MainActivity"**)  
 capabilities.setCapability(**"appWaitActivity"**, **"com.twitter.app.main.MainActivity"**)  
  
 **driver** = AndroidDriver(URL(**"http://127.0.0.1:4723/wd/hub"**), capabilities)  
 **wait** = WebDriverWait(**driver**, 15)  
 **action** = AndroidTouchAction(**driver**)  
  
 **wait**.until **{ driver**.findElementById(**"android:id/button1"**).*isDisplayed* **}  
 val** firstButton = **driver**.findElementById(**"android:id/button1"**)  
 firstButton.click()  
  
 **wait**.until **{ driver**.findElementById(**"com.android.packageinstaller:id/permission\_allow\_button"**).*isDisplayed* **}  
 val** allowButton = **driver**.findElementById(**"com.android.packageinstaller:id/permission\_allow\_button"**)  
 allowButton.click()  
 **wait**.until **{ driver**.findElementById(**"com.twitter.android:id/toolbar"**).*isDisplayed* **}** }  
  
 @Test  
 **fun** createAccountWithPhone() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **val** accManager = AccountsManagePage(**driver**)  
  
 accManager.*run* **{  
 wait**.until **{ imgBtn**.*isDisplayed* **}  
 imgBtn**.click()  
  
 **wait**.until **{ accountBtn**.*isDisplayed* **}  
 accountBtn**.click()  
  
 **wait**.until **{ addExistAccountBtn**.*isDisplayed* **}  
 addExistAccountBtn**.click()  
 **}  
  
 val** registrationPage = RegistrationPage(**driver**)  
  
 registrationPage.*run* **{  
 wait**.until **{ signUpButton**.*isDisplayed* **}  
 signUpButton**.click()  
  
 **wait**.until **{ nameInput**.*isDisplayed* **}  
 nameInput**.sendKeys(**"Matvey"**)  
  
 **phoneNumberInput**.sendKeys(**"1"**)  
 **nextButton**.click()  
 *assertFalse* **{ nextButton**.*isEnabled* **}  
  
 phoneNumberInput**.clear()  
 **phoneNumberInput**.sendKeys(System.getenv(**"PHONE\_NUMBER"**))  
 **nextButton**.click()  
 *assertFalse* **{ nextButton**.*isEnabled* **}  
  
 backBtn**.click()  
 **}** }  
  
 @Test  
 **fun** createAccountWithEmail() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **val** accManager = AccountsManagePage(**driver**)  
  
 accManager.*run* **{  
 wait**.until **{ imgBtn**.*isDisplayed* **}  
 imgBtn**.click()  
  
 **wait**.until **{ accountBtn**.*isDisplayed* **}  
 accountBtn**.click()  
  
 **wait**.until **{ addExistAccountBtn**.*isDisplayed* **}  
 addExistAccountBtn**.click()  
 **}  
  
 val** registrationPage = RegistrationPage(**driver**)  
  
 registrationPage.*run* **{  
 wait**.until **{ signUpButton**.*isDisplayed* **}  
 signUpButton**.click()  
  
 **wait**.until **{ nameInput**.*isDisplayed* **}  
 nameInput**.sendKeys(**"Matvey"**)  
  
 **phoneNumberInput**.click()  
  
 **wait**.until **{ emailInsteadButton**.*isDisplayed* **}  
 emailInsteadButton**.click()  
 **wait**.until **{ emailInput**.*isDisplayed* **}  
  
 emailInput**.sendKeys(**"1"**)  
 *assertFalse* **{ nextButton**.*isEnabled* **}  
  
 emailInput**.clear()  
 **emailInput**.sendKeys(System.getenv(**"EMAIL"**))  
 *assertFalse* **{ nextButton**.*isEnabled* **}  
  
 backBtn**.click()  
 **}** }  
  
 @Test  
 **fun** loginWithExistAccount() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **val** authorizationPage = AuthorizationPage(**driver**)  
  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"LoginActivity"**))  
  
 authorizationPage.*run* **{  
 wait**.until **{ resetPasswordButton**.*isDisplayed* **}  
 resetPasswordButton**.click()  
  
 **wait**.until **{ navigateUp**.*isDisplayed* **}  
 navigateUp**.click()  
  
 **wait**.until **{ loginInput**.*isDisplayed* **}  
 loginInput**.clear()  
 **loginInput**.sendKeys(System.getenv(**"PHONE\_NUMBER"**))  
 **passwordInput**.clear()  
 **passwordInput**.sendKeys(System.getenv(**"TWITTER\_PASSWORD"**))  
 **loginButton**.click()  
  
 **wait**.until **{ driver**.findElement(By.id(**"com.twitter.android:id/toolbar"**)) != **null }** *assertNotEquals*(**"LoginActivity"**, **driver**.currentActivity())  
 **}** }  
  
 @Test  
 **fun** loginWithUnExistAccount() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **val** accManager = AccountsManagePage(**driver**)  
  
 accManager.*run* **{  
 wait**.until **{ imgBtn**.*isDisplayed* **}  
 imgBtn**.click()  
  
 **wait**.until **{ accountBtn**.*isDisplayed* **}  
 accountBtn**.click()  
  
 **wait**.until **{ addExistAccountBtn**.*isDisplayed* **}  
 addExistAccountBtn**.click()  
 **}  
  
 val** authorizationPage = AuthorizationPage(**driver**)  
  
 authorizationPage.*run* **{  
 wait**.until **{ loginInput**.*isDisplayed* **}  
 loginInput**.sendKeys(**"1"**)  
 **passwordInput**.sendKeys(**"1"**)  
 **loginButton**.click()  
 **}** }  
  
 @Test  
 **fun** viewHomePage() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **val** homePage = HomePage(**driver**)  
  
 homePage.*run* **{  
 wait**.until **{ view**.*isDisplayed* **}  
 view**.click()  
  
 **wait**.until **{ toolbar**.*isDisplayed* **}  
 tweetCurationAction**.click()  
  
 **wait**.until **{ gripper**.*isDisplayed* **}  
 action**.press(PointOption.point(400, 150))  
 **action**.release()  
 **action**.perform()  
  
 **wait**.until **{ toolbar**.*isDisplayed* **}  
 timelineSwitch**.click()  
 **wait**.until **{ gripper**.*isDisplayed* **}  
 action**.press(PointOption.point(400, 150))  
 **action**.release()  
 **action**.perform()  
  
 **wait**.until **{ toolbar**.*isDisplayed* **}  
 imageButton**.click()  
 **wait**.until **{ drawer**.*isDisplayed* **}  
 action**.press(PointOption.point(700, 400))  
 **action**.release()  
 **action**.perform()  
 **wait**.until **{ toolbar**.*isDisplayed* **}  
 }** }  
  
 @Test  
 **fun** createTwit() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **wait**.until **{ driver**.findElement(By.id(**"com.twitter.android:id/composer\_write"**)).*isDisplayed* **}  
 driver**.findElementByAccessibilityId(**"Home Tab"**).click()  
  
 **val** createTwitBtn = **driver**.findElement(By.id(**"com.twitter.android:id/composer\_write"**))  
 **wait**.until **{** createTwitBtn.*isDisplayed* **}** createTwitBtn.click()  
  
 **val** createTwitPage = CreateTwitPage(**driver**)  
  
 createTwitPage.*run* **{  
 wait**.until **{ textArea**.*isDisplayed* **}  
 imageView**.click()  
 **wait**.until **{ accountFrame**.*isDisplayed* **}  
 action**.press(PointOption.point(350, 350))  
 **action**.release()  
 **action**.perform()  
 **wait**.until **{ textArea**.*isDisplayed* **}  
  
 textInput**.sendKeys(**"qwerty"**)  
  
 **gifBtn**.click()  
 **wait**.until **{ gifView**.*isDisplayed* **}  
  
 gifView**.click()  
 **wait**.until **{ gifImageView**.*isDisplayed* **}**  
  
 **gifImageView**.click()  
 **wait**.until **{ textArea**.*isDisplayed* **}  
  
 twitButton**.click()  
 **}  
  
 wait**.until **{ driver**.findElement(By.id(**"com.twitter.android:id/composer\_write"**)).*isDisplayed* **}** }  
  
 @Test  
 **fun** searchInfo() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **val** searchPage = SearchPage(**driver**)  
  
 searchPage.*run* **{  
 wait**.until **{ searchBtn**.*isDisplayed* **}  
 searchBtn**.click()  
  
 **wait**.until **{ searchTwitBtn**.*isDisplayed* **}  
 searchTwitBtn**.click()  
  
 **wait**.until **{ searchTwitInput**.*isDisplayed* **}  
 searchTwitInput**.sendKeys(**"a"**)  
  
 **wait**.until **{ searchResult**.*isDisplayed* **}  
 searchResult**.click()  
  
 **wait**.until **{ settingBtn**.*isDisplayed* **}  
 settingBtn**.click()  
  
 **wait**.until **{ title**.*isDisplayed* **}  
 title**.click()  
  
 **wait**.until **{ settingsToolbar**.*isDisplayed* **}  
 closeBtn**.click()  
  
 **wait**.until **{ settingBtn**.*isDisplayed* **}  
 closeBtn**.click()  
  
 **wait**.until **{ searchBtn**.*isDisplayed* **}  
 }** }  
  
 @Test  
 **fun** editNotification() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **val** notificationPage = NotificationPage(**driver**)  
  
 notificationPage.*run* **{  
 wait**.until **{ notificationBtn**.*isDisplayed* **}  
 notificationBtn**.click()  
  
 **wait**.until **{ mentions**.*isDisplayed* **}  
 mentions**.click()  
  
 **wait**.until **{ settingsBtn**.*isDisplayed* **}  
 settingsBtn**.click()  
  
 **wait**.until **{ backwardBtn**.*isDisplayed* **}  
 backwardBtn**.click()  
 **wait**.until **{ settingsBtn**.*isDisplayed* **}  
 }** }  
  
 @Test  
 **fun** sendMessage() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **val** messagePage = MessagePage(**driver**)  
  
 messagePage.*run* **{  
 wait**.until **{ messageBtn**.*isDisplayed* **}  
 messageBtn**.click()  
  
 **wait**.until **{ createMessage**.*isDisplayed* **}  
 createMessage**.click()  
  
 **wait**.until **{ profile**.*isDisplayed* **}  
 profile**.click()  
  
 **wait**.until **{ nextBtn**.*isDisplayed* **}  
 nextBtn**.click()  
  
 **wait**.until **{ tweetText**.*isDisplayed* **}  
 tweetText**.click()  
  
 **wait**.until **{ tweetText**.*isDisplayed* **}  
 tweetText**.sendKeys(**"qwerty"**)  
  
 **wait**.until **{ sendBtn**.*isDisplayed* **}  
 sendBtn**.click()  
  
 **wait**.until **{ backwardBtn**.*isDisplayed* **}  
 backwardBtn**.click()  
  
 **wait**.until **{ messageBtn**.*isDisplayed* **}  
 }** }  
  
 @Test  
 **fun** createWidget() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **driver**.navigate().back()  
 **if** (**driver**.findElementsByAccessibilityId(**"Apps list"**).*isNotEmpty*()) {  
 **val** appsListBtn = **driver**.findElementByAccessibilityId(**"Apps list"**) **as** MobileElement  
 appsListBtn.click()  
 }  
 **wait**.until **{ driver**.findElementById(**"com.android.launcher3:id/search\_box\_input"**).*isDisplayed* **}  
  
 val** size = **driver**.manage().window().*size* **val** anchor = (size.**width** \* 0.1).toInt()  
 **val** startPoint = (size.**height** \* 0.9).toInt()  
 **val** endPoint = (size.**height** \* 0.1).toInt()  
  
 **action** .press(PointOption.point(anchor, startPoint))  
 .waitAction(WaitOptions.waitOptions(ofMillis(1000)))  
 .moveTo(PointOption.point(anchor, endPoint))  
 .release().perform()  
  
 **action** .longPress(PointOption.point(290, 1060))  
 .moveTo(PointOption.point(anchor, startPoint))  
 .release().perform()  
 }  
  
 @Test  
 **fun** viewProfilePage() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **wait**.until **{ driver**.findElementByAccessibilityId(**"Show navigation drawer"**).*isDisplayed* **}  
 val** navigation = **driver**.findElementByAccessibilityId(**"Show navigation drawer"**)  
 navigation.click()  
  
 **val** profilePage = ProfilePage(**driver**)  
  
 profilePage.*run* **{  
 wait**.until **{ profileBtn**.*isDisplayed* **}  
 profileBtn**.click()  
  
 **wait**.until **{ tweetsBtn**.*isDisplayed* **}  
 tweetsBtn**.click()  
  
 **wait**.until **{ tweetsAndRepliesBtn**.*isDisplayed* **}  
 tweetsAndRepliesBtn**.click()  
  
 **wait**.until **{ mediaBtn**.*isDisplayed* **}  
 mediaBtn**.click()  
  
 **wait**.until **{ likesBtn**.*isDisplayed* **}  
 likesBtn**.click()  
 **}  
  
 driver**.findElementByAccessibilityId(**"Navigate up"**).click()  
 }  
  
 @Test  
 **fun** createLists() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **wait**.until **{ driver**.findElementByAccessibilityId(**"Show navigation drawer"**).*isDisplayed* **}  
 val** navigation = **driver**.findElementByAccessibilityId(**"Show navigation drawer"**)  
 navigation.click()  
  
 **val** listsPage = ListsPage(**driver**)  
  
 listsPage.*run* **{  
 wait**.until **{ listsBtn**.*isDisplayed* **}  
 listsBtn**.click()  
  
 **wait**.until **{ createBtn**.*isDisplayed* **}  
 createBtn**.click()  
  
 **wait**.until **{ nameInput**.*isDisplayed* **}  
 nameInput**.sendKeys(**"qwerty"**)  
 **descriptionInput**.sendKeys(**"qwerty"**)  
  
 **saveBtn**.click()  
 **wait**.until **{ createBtn**.*isDisplayed* **}  
 }  
  
 driver**.findElementByAccessibilityId(**"Navigate up"**).click()  
 }  
  
 @Test  
 **fun** deleteAllBookmarks() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **wait**.until **{ driver**.findElementByAccessibilityId(**"Show navigation drawer"**).*isDisplayed* **}  
 val** navigation = **driver**.findElementByAccessibilityId(**"Show navigation drawer"**)  
 navigation.click()  
  
 **val** bookmarksPage = BookmarksPage(**driver**)  
  
 bookmarksPage.*run* **{  
 wait**.until **{ bookmarksBtn**.*isDisplayed* **}  
 bookmarksBtn**.click()  
  
 **wait**.until **{ moreOptionBtn**.*isDisplayed* **}  
 moreOptionBtn**.click()  
  
 **wait**.until **{ deleteAllBtn**.*isDisplayed* **}  
 deleteAllBtn**.click()  
  
 **wait**.until **{ approvalBtn**.*isDisplayed* **}  
 approvalBtn**.click()  
  
 **wait**.until **{ moreOptionBtn**.*isDisplayed* **}  
 }  
  
 driver**.findElementByAccessibilityId(**"Navigate up"**).click()  
 }  
  
 @Test  
 **fun** openMoments(){  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **wait**.until **{ driver**.findElementByAccessibilityId(**"Show navigation drawer"**).*isDisplayed* **}  
 val** navigation = **driver**.findElementByAccessibilityId(**"Show navigation drawer"**)  
 navigation.click()  
  
 **val** momentsPage = MomentsPage(**driver**)  
  
 momentsPage.*run* **{  
 wait**.until **{ momentsBtn**.*isDisplayed* **}  
 momentsBtn**.click()  
  
 **wait**.until **{ momentsArea**.*isDisplayed* **}  
 }  
  
 driver**.findElementByAccessibilityId(**"Navigate up"**).click()  
 }  
  
 @Test  
 **fun** changeTheme() {  
 **driver**.startActivity(Activity(**"com.twitter.android"**, **"com.twitter.app.main.MainActivity"**))  
  
 **wait**.until **{ driver**.findElementByAccessibilityId(**"Show navigation drawer"**).*isDisplayed* **}  
 val** navigationBtn = **driver**.findElementByAccessibilityId(**"Show navigation drawer"**)  
 navigationBtn.click()  
  
 **wait**.until **{ driver**.findElementByXPath(**"//android.view.ViewGroup[@content-desc=\"Dark mode\"]/android.widget.ImageView"**).*isDisplayed* **}  
 val** changeThemeBtn = **driver**.findElementByXPath(**"//android.view.ViewGroup[@content-desc=\"Dark mode\"]/android.widget.ImageView"**)  
 changeThemeBtn.click()  
 }  
  
 @AfterAll  
 **fun** afterAll() {  
 **driver**.quit()  
 }  
}

**Вывод**

В ходе выполнения лабораторной работы мной был изучен стек технологий, указанный в целях и задачах. Для реализации тестовых сценариев был выбран паттерн проектирования Page Object, его применение облегчило процесс разработки, улучшило читаемость кода и уменьшило его размер. То же самое характерно и для выбора языка реализации – Kotlin.

В итоге мной был изучен и закреплён процесс взаимодействия с инструментом для автоматизации действий андроид приложения, приобретен опыт написания тестовых сценариев и их реализации, изучены основы тестирования и обеспечения качества.