Государственное бюджетное профессиональное образовательное учреждение

«Нижегородский Губернский колледж»

**UI-тесты**

**Клиентская часть приложения CodeBlitz**

Группа: 41П

Студент: Фадеев А. В.

Нижний Новгород

2024

# ОСНОВНАЯ ЧАСТЬ

# Предварительные действия

Был добавлен альтернативный класс для запуска тестов, необходимый для работы hilt в тестах.

Листинг 1 – CustomTestRunner

|  |
| --- |
| package com.example.codeblitz  import android.app.Application import android.content.Context import androidx.test.runner.AndroidJUnitRunner import dagger.hilt.android.testing.HiltTestApplication  class CustomTestRunner : AndroidJUnitRunner() {   override fun newApplication(cl: ClassLoader?, name: String?, context: Context?): Application {  return super.newApplication(cl, HiltTestApplication::class.*java*.*name*, context)  } } |

Листинг 2 – build.gradle.kts

|  |
| --- |
| *android* **{** namespace = "com.example.codeblitz"  compileSdk = 35   defaultConfig **{** applicationId = "com.example.codeblitz"  minSdk = 29  targetSdk = 34  versionCode = 1  versionName = "1.0"   testInstrumentationRunner = "com.example.codeblitz.CustomTestRunner"  vectorDrawables **{** useSupportLibrary = true  **}  }**  …… |

## Описание UI-тестов

Все UI-тесты были написаны на языке Kotlin с использованием @HiltAndroidTest.

Таблица 1 – UI-тесты

| Номер | Название | Описание | Ожидаемый результат |
| --- | --- | --- | --- |
| 1 | Проверка перехода на страницу регистрации | Проверяет, что при нажатии кнопки «Зарегистрироваться» выполняется переход на страницу регистрации и страница отображается | Переход выполнен, страница отобразилась |
| 2 | Проверка отображения сообщений на странице регистрации и доступности кнопки | Проверяет, что при вводе значений в поля на странице регистрации предупреждения о некорректной почте и слабом пароле исчезают и кнопка «Регистрация» становится доступна | После ввода значений предупреждения перестали отображаться, кнопка стала доступна |
| 3 | Проверка ввода в поля логина и пароля и последующей авторизации | Проверяет, что возможно ввести значения в поле логина и пароля на странице авторизации и что авторизация выполняется | Значения вводятся в поля логина и пароля. Авторизация успешно выполняется |
| 4 | Проверка отображения плиток заданий у администратора | Проверяет, что у администратора все отображаются 2 плитки с заданиями | Число плиток равно 2 |
| 5 | Проверка перехода на страницу добавления задания при нажатии на элемент LazyColumn | Проверяет, что при нажатии на элемент LazyColumn, являющимся плиткой добавления задания, выполняется переход на страницу добавления задания | Переход выполнен |

Ниже представлен листинг всех UI-тестов:

Листинг 3 – UI-тесты

|  |
| --- |
| package com.example.codeblitz  import android.os.SystemClock.sleep import androidx.compose.ui.test.assertCountEquals import androidx.compose.ui.test.assertIsDisplayed import androidx.compose.ui.test.assertIsEnabled import androidx.compose.ui.test.assertIsNotDisplayed import androidx.compose.ui.test.assertIsNotEnabled import androidx.compose.ui.test.junit4.createAndroidComposeRule import androidx.compose.ui.test.onAllNodesWithTag import androidx.compose.ui.test.onNodeWithTag import androidx.compose.ui.test.performClick import androidx.compose.ui.test.performTextInput import androidx.test.ext.junit.runners.AndroidJUnit4 import com.example.codeblitz.view.MainActivity.MainActivity import dagger.hilt.android.testing.HiltAndroidRule import dagger.hilt.android.testing.HiltAndroidTest import org.junit.Before import org.junit.Rule import org.junit.Test import org.junit.runner.RunWith  */\*\*  \* UI-тесты  \*/* @HiltAndroidTest @RunWith(AndroidJUnit4::class) class CodeBlitzInstrumentalTests {   @get:Rule  val composeTestRule = *createAndroidComposeRule*<MainActivity>()   @get:Rule  var hiltRule = HiltAndroidRule(this)   @Before  fun setUp() {  hiltRule.inject()  }   */\*\*  \* Проверка перехода на страницу регистрации  \*/* @Test  fun testNavigationToRegister() {  composeTestRule.*onNodeWithTag*("Login").*assertIsDisplayed*()   composeTestRule.*onNodeWithTag*("To Register Button").*performClick*()   composeTestRule.*onNodeWithTag*("Register").*assertIsDisplayed*()  }   */\*\*  \* Проверка отображения сообщений на странице регистрации и доступности кнопки  \*/* @Test  fun testRegisterWarning() {  composeTestRule.*onNodeWithTag*("To Register Button").*performClick*()   composeTestRule.*onNodeWithTag*("Wrong Email").*assertIsDisplayed*()  composeTestRule.*onNodeWithTag*("Weak Password").*assertIsDisplayed*()   composeTestRule.*onNodeWithTag*("Login Register").*performClick*()  composeTestRule.*onNodeWithTag*("Login Register").*performTextInput*("errorfadeev@yandex.ru")  composeTestRule.*onNodeWithTag*("Wrong Email").*assertIsNotDisplayed*()   composeTestRule.*onNodeWithTag*("Password Register").*performClick*()  composeTestRule.*onNodeWithTag*("Password Register").*performTextInput*("1234567890")  composeTestRule.*onNodeWithTag*("Weak Password").*assertIsNotDisplayed*()   composeTestRule.*onNodeWithTag*("Register Button").*assertIsNotEnabled*()   composeTestRule.*onNodeWithTag*("Repeat Password").*performClick*()  composeTestRule.*onNodeWithTag*("Repeat Password").*performTextInput*("1234567890")   composeTestRule.*onNodeWithTag*("Register Button").*assertIsEnabled*()  }   */\*\*  \* Проверка ввода в поля логина и пароля и последующей авторизации  \*/* @Test  fun testLoginTextFieldsAndLoginIn() {  composeTestRule.*onNodeWithTag*("Login Login").*performClick*()  composeTestRule.*onNodeWithTag*("Login Login").*performTextInput*("errorfadeev@yandex.ru")   composeTestRule.*onNodeWithTag*("Password Login").*performClick*()  composeTestRule.*onNodeWithTag*("Password Login").*performTextInput*("123")   composeTestRule.*onNodeWithTag*("Login Button").*performClick*()  sleep(10000)  composeTestRule.*onNodeWithTag*("Main").*assertIsDisplayed*()  }   */\*\*  \* Проверка отображения плиток заданий у администратора  \*/* @Test  fun testAdminMainScreenTasksDisplay() {  composeTestRule.*onNodeWithTag*("Login Login").*performClick*()  composeTestRule.*onNodeWithTag*("Login Login").*performTextInput*("errorfadeev@yandex.ru")  composeTestRule.*onNodeWithTag*("Password Login").*performClick*()  composeTestRule.*onNodeWithTag*("Password Login").*performTextInput*("123")  composeTestRule.*onNodeWithTag*("Login Button").*performClick*()  sleep(10000)  sleep(10000)  composeTestRule.*onAllNodesWithTag*("TaskElement").*assertCountEquals*(2)  }   */\*\*  \* Проверка перехода на страницу добавления задания при нажатии на элемент LazyColumn  \*/* @Test  fun testLazyColumnOnItemClickNavigation() {  composeTestRule.*onNodeWithTag*("Login Login").*performClick*()  composeTestRule.*onNodeWithTag*("Login Login").*performTextInput*("errorfadeev@yandex.ru")  composeTestRule.*onNodeWithTag*("Password Login").*performClick*()  composeTestRule.*onNodeWithTag*("Password Login").*performTextInput*("123")  composeTestRule.*onNodeWithTag*("Login Button").*performClick*()  sleep(10000)  sleep(10000)  composeTestRule.*onNodeWithTag*("TaskElementID - 1").*performClick*()  composeTestRule.*onNodeWithTag*("AddTask").*assertIsDisplayed*()  } } |

Все UI-тесты были успешно пройдены:

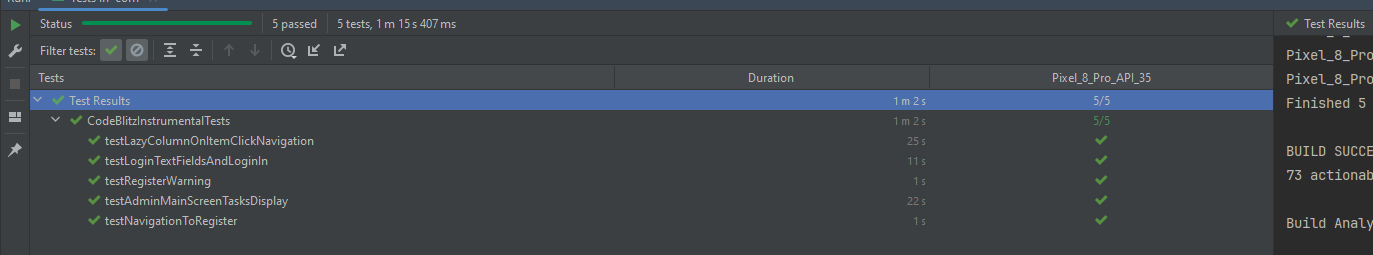


Рисунок 1 – Выполнение юнит-тестов