

Государственное бюджетное профессиональное образовательное учреждение
«Нижегородский Губернский колледж»

ТЕСТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Руководитель _____ Мамшева Ю.С.

30.11.2024

Студент _____ Астапчик Д.А.

30.11.2024

Специальность, группа: 09.02.07, 41П

Нижний Новгород
2024 г

СОДЕРЖАНИЕ

1. ТЕСТ ПЛАН	3
1.1 Функционал мобильного приложения	4
1.2 Критерии качества	5
1.3 Перечень тестовой документации	6
1.4 Стратегия тестирования	7
1.5 Ресурсы	8
2. ТЕСТИРОВАНИЕ	9
2.1 Unit-testing	10
2.2 UI-testing	13

1. ТЕСТ ПЛАН

На данном этапе было необходимо произвести процесс, целью которого является реализация процесса, что заключается в разработке специализированного тест-плана для разработанного продукта “FlyV”.

Рассмотрим перечень пунктов, что содержит тест-план:

1. Функционал мобильного программного продукта, что будет подвержен тестированию;
2. Критерии качества;
3. Перечень документации, что будет использоваться для тестирования серверной части мобильного программного продукта;
4. Стратегия тестирования серверной части приложения;
5. Ресурсы.

1.1 Функционал мобильного приложения

Рассмотрим функционал мобильного приложения, что будет подвержен тестированию, по причине необходимости, выявить несоответствия работоспособности функционала мобильного программного продукта “FlyV”:

1. Метод, предназначенный для преобразования типа String в LocalDate (convertStringToLocalDate);
2. Метод, предназначенный для преобразования типа LocalDate в Calendar (localDateToCalendar);
3. Метод, предназначенный для определения, находится ли дата в заданном диапазоне дат (dateBetweenBeginDateAndEndDate);
4. Метод, предназначенный для обновления пользовательских данных, при переходе на окно профиля (refreshData);
5. Класс, где присутствует возможность отображать карточку баланса отдыха пользователя (isShowCardBalanceHoliday);
6. Отображение элементов на экране профиля пользователя системы (ProfileScreen);
7. Отображение элементов на экране редактирования профиля (EditProfileScreen);
8. Отображение карточки баланса отдыха, после действий пользователя направленных на активацию карточки (BalanceHolidayCardScreen);
9. Изменение содержимого полей ввода на окне авторизации, после действий пользователя, направленных на ввод данных (LoginScreen);
10. Валидация вводимых данных пользователем, на окне авторизации (LoginScreen).

1.2 Критерии качества

На данном этапе было необходимо составить список критериев, что будут эксплуатироваться для реализации процесса, что заключается в оценке качества мобильного программного продукта “FlyV”:

1. Функциональность:

– Весь ранее указанный функционал является работоспособным;

2. Надёжность:

– Функционал мобильного приложения должен производить задуманное в период осуществления работы, исключая сценарии возникновения различных дефектов;

– Должны присутствовать механизмы защиты, предоставляющие возможность обрабатывать различные исключительные ситуации.

3. Удобство сопровождения:

– Должна присутствовать возможность расширения и сопровождения функционала мобильного программного продукта.

4. Эффективность:

– Время выполнения поставленных задач, должно находиться в пределах нормы.

1.3 Перечень тестовой документации

Рассмотрим перечень тестовой документации, что будет эксплуатироваться в целях проведения эффективного тестирования мобильного программного продукта “FlyV” (табл 1).

Таблица 1. Перечень тестовой документации

Вид документации	Краткая сводка информация о документации
Тест-план	Документ, описывающий весь объем работ по тестированию
Критерии качества	Набор информации, определяющий требования к качеству
Тест-кейс	Создание конкретных сценариев тестирования, описывающих шаги и ожидаемый результат
Стратегия тестирования	Набор информации, предназначенный для определения основных направлений и методов тестирования

1.4 Стратегия тестирования

На данном этапе выполнения поставленной задачи, было необходимо произвести создание специализированной стратегии тестирования мобильного программного продукта “FlyV” (табл 2).

Тестирование	Процедуры тестирования
Unit-testing	<ol style="list-style-type: none">1. Проверить результат работы метода <code>convertStringToLocalDate</code>;2. Проверить результат работы метода <code>localDateToCalendar</code>;3. Проверить результат работы метода <code>dateBetweenBeginDateAndEndDate</code>;4. Проверить результат работы метода <code>refreshData</code>;5. Проверить, что карточка баланса отдыха изначально скрыта.
UI-testing	<ol style="list-style-type: none">1. Проверить, отображение элементов на экране профиля пользователя системы;2. Проверить отображение элементов на экране редактирования профиля пользователя системы;3. Проверить, отображение карточки баланса отдыха после действий пользователя, направленных на активацию карточки;4. Проверить, изменение содержимого полей ввода на окне авторизации, после действий пользователя направленных на ввод данных;5. Проверить, валидацию данных на окне авторизации.

1.5 Ресурсы

Рассмотрим перечень ресурсов, что необходимы для успешной реализации процесса тестирования серверной части мобильного ПП.

Программные ресурсы:

1. Среда тестирования: Android Studio;
2. Разработанный функционал.

Человеческие ресурсы:

1. Тестировщик;
2. Разработчик.

Временные ресурсы:

1. Время на реализацию всех ранее описанных тестов программному продукту.

2. ТЕСТИРОВАНИЕ

На данном этапе разработки документации по тестированию мобильного приложения был произведён процесс тестирования, ранее описанного функционала в тест-плане. Рассмотрим составленные тестовые наборы:

1. Тестовый набор – “Unit-testing”;
2. Тестовый набор – “UI-testing”.

2.1 Unit-testing

Предварительные условия: для того, чтобы осуществить данный вид тест-кейсов, необходимо открыть Android Studio, с разработанным программным продуктом. В последствии необходимо открыть класс ExampleUnitTest.

1. Тест-кейс, преобразование типа данных String в LocalDate (табл 3).

Таблица 3. Тест-кейс 1

Шаги	Ожидаемый результат
Оставить доступным только тест convertStringToLocalDateIsSuccess	Все тесты, не относящиеся к указанному, будут закомментированы
Нажать ПКМ на класс ExampleUnitTest	Появится специализированное меню
В открывшемся меню нажать на пункт: Run 'ExampleUnitTest'	Пользователь увидит успешный результат преобразования типа String в LocalDate

2. Тест-кейс, преобразование типа данных LocalDate в Calendar (табл 4).

Таблица 4. Тест-кейс 2

Шаги	Ожидаемый результат
Оставить доступным только тест convertLocalDateToCalendarIsSuccess	Все тесты, не относящиеся к указанному, будут закомментированы
Нажать ПКМ на класс ExampleUnitTest	Появится специализированное меню
В открывшемся меню нажать на пункт: Run 'ExampleUnitTest'	Пользователь увидит успешный результат преобразования типа LocalDate в Calendar

3. Тест-кейс, определить, что дата, находится в диапазоне дат (табл 5).

Таблица 5. Тест-кейс 3

Шаги	Ожидаемый результат
Оставить доступным только тест dateBetweenBeginDateAndEndDateIsSuccess	Все тесты, не относящиеся к указанному, будут закомментированы
Нажать ПКМ на класс ExampleUnitTest	Появится специализированное меню
В открывшемся меню нажать на пункт: Run 'ExampleUnitTest'	Пользователь увидит, успешный результат того, что функция определит нахождение даты в диапазоне дат

4. Тест-кейс, обновить пользовательские данные (табл 6).

Таблица 6. Тест-кейс 4

Шаги	Ожидаемый результат
Оставить доступным только тест refreshProfileDataIsSuccess	Все тесты, не относящиеся к указанному, будут закомментированы
Нажать ПКМ на класс ExampleUnitTest	Появится специализированное меню
В открывшемся меню нажать на пункт: Run 'ExampleUnitTest'	Пользователь увидит, успешный результат того, что функция refreshData обновляет пользовательские данные

5. Тест-кейс, проверить, что изначально при открытии окна профиля пользователя карточка баланса отдыха скрыта (табл 6).

Таблица 7. Тест-кейс 5

Шаги	Ожидаемый результат
Оставить доступным только тест <code>cardBalanceHolidayIsClosed</code>	Все тесты, не относящиеся к указанному, будут закомментированы
Нажать ПКМ на класс <code>ExampleUnitTest</code>	Появится специализированное меню
В открывшемся меню нажать на пункт: <code>Run 'ExampleUnitTest'</code>	Пользователь увидит, успешный результат того, что карточка баланса отдыха пользователя системы исходно, является закрытой

2.2 UI-testing

Предварительные условия: для того, чтобы осуществить данный вид тест-кейсов, необходимо открыть Android Studio, с разработанным программным продуктом. В последствии необходимо открыть класс ExampleInstrumentedTest.

1. Тест-кейс, отобразить все элементы окна профиля пользователя (ProfileScreen)

Таблица 8. Тест-кейс 1

Шаги	Ожидаемый результат
Оставить доступным только тест profileViewIsDisplayed	Все тесты, не относящиеся к указанному, будут закомментированы
Нажать ПКМ на класс ExampleInstrumentedTest	Появится специализированное меню
В открывшемся меню нажать на пункт: Run 'ExampleUnitTest'	Пользователь увидит, успешный результат того, все элементы окна пользователя (ProfileScreen) отображаются

2. Тест-кейс, отобразить все элементы окна редактирования пользователя (EditProfileScreen)

Таблица 9. Тест-кейс 2

Шаги	Ожидаемый результат
Оставить доступным только тест editProfileViewIsDisplayed	Все тесты, не относящиеся к указанному, будут закомментированы
Нажать ПКМ на класс ExampleInstrumentedTest	Появится специализированное меню
В открывшемся меню нажать на пункт: Run 'ExampleUnitTest'	Пользователь увидит, успешный результат того, все элементы окна редактирования пользовательских данных (EditProfileScreen) отображаются

3. Тест-кейс, проверить отображение карточки баланса отдыха пользователя в окне ProfileScreen, после действий пользователя направленных на её активацию

Таблица 10. Тест-кейс 3

Шаги	Ожидаемый результат
Оставить доступным только тест balanceHolidayCardViewIsDisplayedAfterClickUser	Все тесты, не относящиеся к указанному, будут закомментированы
Нажать ПКМ на класс ExampleInstrumentedTest	Появится специализированное меню
В открывшемся меню нажать на пункт: Run ‘ExampleUnitTest’	Пользователь увидит, успешный результат того, что карточка баланса отдыха после нажатия пользователя на кнопку отображается

4. Тест-кейс, проверить, что вводимые данные пользователем в поля “Логин” и “Пароль”, на окне авторизации, отображаются

Таблица 11. Тест-кейс 4

Шаги	Ожидаемый результат
Оставить доступным только тест <code>insertTextInOutlinedTextFieldEmailIsDisplayedInLoginView</code>	Все тесты, не относящиеся к указанному, будут закомментированы
Нажать ПКМ на класс <code>ExampleInstrumentedTest</code>	Появится специализированное меню
В открывшемся меню нажать на пункт: Run <code>ExampleUnitTest</code>	Пользователь увидит, успешный результат того, что внесённые данные пользователем отображаются

5. Тест-кейс, проверить, что осуществляется валидация данных на окне авторизации

Таблица 12. Тест-кейс 5

Шаги	Ожидаемый результат
Оставить доступным только тест emailAndPasswordValidationIsWorks	Все тесты, не относящиеся к указанному, будут закомментированы
Нажать ПКМ на класс ExampleInstrumentedTest	Появится специализированное меню
В открывшемся меню нажать на пункт: Run 'ExampleUnitTest'	Пользователь увидит, успешный результат того, что валидация данных срабатывает