**Документация для тестирования мобильного приложения**

# **1. Тест план.**

# **Перечень работ.**

Перечень функциональных областей приложения, которые будут подвергаться тестированию:

1. Пользовательский интерфейс: проверка отображения ошибок и самих экранов.
2. Конвертеры: проверка функций, отвечающих за преобразование одного вида даты в другой.
3. Функции валидации данных: проверка регулярных выражений для почты и никнейма.
4. Авторизация в системе: возможность создать аккаунт и авторизоваться
5. Навигация: можно ли переключать страницы.
6. Совместимость: проверка совместимости с разными версиями Андроид.

# **Критерии качества.**

1. Надежность: способность программного продукта выполнять свои функции без сбоев в течение определенного времени или количества операций.
2. Безопасность: обеспечение защиты информации и ресурсов от несанкционированного доступа, изменений.
3. Производительность: способность программного продукта эффективно выполнять свои функции в разумные сроки и с минимальным потреблением ресурсов.
4. Масштабируемость: способность программного продукта адаптироваться к увеличению объема данных или количества пользователей без значительного снижения производительности.
5. Удобство использования: уровень удобства и простоты использования программного продукта для пользователя, включая понятный интерфейс, интуитивную навигацию и доступность функций.
6. Сопровождаемость: способность программного продукта быть легко модифицируемым, исправляемым и обновляемым без необходимости переписывания большого количества кода.
7. Поддерживаемость: возможность программного продукта быть поддержанным и обслуживаемым в течение всего его жизненного цикла, включая обновления, исправления ошибок и обучение пользователей.
8. Эффективность: способность программного продукта выполнять свои функции с минимальным использованием ресурсов, таких как память, процессорное время и сетевой трафик.

# **Оценка рисков.**

Таблица 2. Оценка рисков

|  |  |  |  |
| --- | --- | --- | --- |
| **Риск** | **Оценка риска** | **Описание риска** | **Выход из ситуации** |
| Некорректный ввод данных | Высокий | Неправильный ввод данных может привести к ошибочным результатам и недовольству пользователей. | Провести тщательную проверку ввода данных, проверяя возможность ввода пустой строки, null, отрицательных значений (для числовых полей), специальных символов, проверить регулярные выражения для ввода числовых значений |
| Нестабильная работа приложения | Средняя | Нестабильная работа приложения может привести к потере данных и недовольству пользователей. | Провести тщательное тестирование на различных устройствах и в различных условиях использования - использовать механизмы отладки и мониторинга для выявления и исправления ошибок |
| Низкая производительность | Средняя | Низкая производительность приложения может привести к недовольству пользователей. | Оптимизировать код, использовав эффективные алгоритмы - произвести тестирование производительности и мониторинг нагрузки на приложение. |

# **Стратегия тестирования.**

Приложение планируется протестировать, применив следующие стратегии:

1. UI – тестирование.
2. Негативное тестирование - тестирование интерфейса пользователя и его удобства использования. Оно включает проверку навигации, доступности элементов управления и понятности сообщений об ошибках.
3. Позитивное тестирование - тестирование, при котором приложение проверяется на корректную работу при вводе правильных данных и выполнении ожидаемых функций.
4. Тестирование белого ящика - тестирование, при котором проверяются внутренние компоненты и функциональность приложения. Оно включает проверку алгоритмов, обработку ошибок и безопасность.
5. Тестирование черного ящика - тестирование, при котором проверяется работа приложения без знания его внутренних компонентов и функциональности. Оно включает проверку интерфейса пользователя, совместимости с различными операционными системами и браузерами, а также проверку производительности и надежности.

Стратегия тестирования – это план действий, который определяет, как будет проводиться тестирование приложения. Она включает в себя выбор методов и инструментов тестирования, а также определение приоритетов и целей тестирования.

# **Ресурсы.**

Для разработки мобильного приложения StudyBuddy могут быть задействованы следующие ресурсы:

Человеческие ресурсы: человек, который будут заниматься разработкой приложения.

Механические ресурсы: это оборудование, которое будет использоваться для разработки приложения. Это рабочее место студента (АРМ) в колледже, рабочее место студента дома.

Технические ресурсы: это программное обеспечение и инструменты, которые будут использоваться для разработки приложения. Это язык программирования Kotlin, современный фреймворк для создания пользовательских интерфейсов на платформе Android Jetpack Compose, библиотеки для тестирования, среда разработки Android Studio, инструменты тестирования, наличие системы удаленных версий.

Информационные ресурсы: это данные, которые будут использоваться для тестирования приложения. Видеоуроки и статьи в интернете, гайды по тестированию пользовательского интерфейса.

# **Метрики**

Точность: Программа должна правильно регистрировать пользователя в системе и выводить только данные авторизованного пользователя. Это может быть измерено путем ручного тестирования авторизации пользователя в системе и проверке, всегда ли пользователь получает свои данные после авторизации, нет ли случайного отображения данных пользователя, который входил ранее.

Скорость: Программа должна быть быстрой и эффективной в обработке данных.

Надежность: Программа должна быть надежной и стабильной, то есть она не должна вылетать или вызывать ошибки при обработке данных. Это может быть измерено путем тестирования программы на различных наборах данных и проверки ее стабильности.

Использование ресурсов: Программа должна потреблять минимальное количество ресурсов, таких как память и процессорное время.

# **Расписание и ключевые точки.**

Таблица 3. Расписание и ключевые точки.

|  |  |
| --- | --- |
| **Этап** | **Количество пар** |
| Разработка тестового набора и тест-кейсов | 1 |
| Реализация 5 unit-test и 5 UI-test к приложению | 1 |

# **Тестовый набор**

Тестовый набор – это набор тест-кейсов, в которых результат описывается предисловием, то есть очередность проводимых тестов.

* 1. **Выбор фреймворка для тестирования**

Для тестирования будут использованы следующие инструменты:

1. Фреймворк JUnit 4 - популярный фреймворк для написания юнит-тестов в Java (и других языках, поддерживающих JVM, Kotlin в том числе).
2. Аннотации: для обозначения методов как тестов (@Test), методов настройки перед тестом (@Before), методов очистки после теста (@After), методов настройки всего тестового набора (@BeforeClass), методов очистки всего тестового набора (@AfterClass) и др. Эти аннотации упрощают организацию и управление тестами.
3. Ассерты (Assertions): Методы для проверки ожидаемых результатов. Например, assertEquals(), assertTrue(), assertFalse() и другие, позволяют сравнивать фактические результаты с ожидаемыми и сообщать об ошибках, если они не совпадают.
4. Средства тестирования производительности в Android Studio
   1. **Проверка валидации данных при помощи модульного тестирования.**

В мобильном приложении есть регистрация и авторизация, в которых присутствуют поля, которые необходимо проверять на клиенте, перед отправкой на сервер в POST запросе, такие как почта и никнейм пользователя. Почта должна соответствовать паттерну. В Таблице 4 ниже указаны тест-кейсы.

Таблица 4. Тест-кейсы для unit-тестов системы авторизации.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Описание** | **Входные данные** | **Ожидаемый результат** | **Проверка** | **Прошел** |
| 1 | Проверка соответствия паттерну почты строки, которая соответствует паттерну почты (Позитивный тест) | 1. Почта: user@mail.ru. | – Возвращается результат проверки истина - true | Assert.assertEquals(expectedResultAfterCheckEmail, actualResultAfterCheck) | + |
| 2 | Проверка несоответствия паттерну почты строки, которая не соответствует паттерну почты (Негативный тест) | 1. Почта: usermail.ru. | – Возвращается результат проверки ложь - false | Assert.assertEquals(expectedResultAfterCheckEmail, actualResultAfterCheck) | + |
| 3 | Проверка соответствия паттерну для никнейма строки, которая состоит только из символов кириллицы и латиницы (Позитивный тест) | 1. Никнейм: NiCkНеЙм. | – Возвращается результат проверки истина - true | Assert.assertEquals(expectedResultAfterCheckNickname, actualResultAfterCheck | + |
| 4 | Проверка несоответствия паттерну для никнейму, которая состоит не только из символов кириллицы и латиницы (Негативный тест) | 1. Никнейм: \_(NiCkНеЙм)1. | – Возвращается результат проверки ложь - false | Assert.assertEquals(expectedResultAfterCheckNickname, actualResultAfterCheck) | + |

* 1. **Проверка функций-конвертеров даты.**

В мобильном приложении есть разные форматы отображения даты. Из API дата приходит в том же формате, что и хранится в базе данных (например, 2024-11-26), а значит конвертеры должны преобразовать эту дату в следующие форматы: 26 ноября 2024, 26.11.2024. В Таблице 5 ниже указаны тест-кейсы.

Таблица 5. Тест-кейсы для unit-тестов конвертеров.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Описание** | **Входные данные** | **Ожидаемый результат** | **Проверка** | **Прошел** |
| 1 | Проверка правильного преобразования даты из формата YYYY-MM-DD в формат даты, представленной в словесной форме | Строка: 2024-11-26. | Строка: 26 ноября 2024. | assertEquals(expectedFormattedDate, actualFormattedDate) | + |
| 2 | Проверка правильного преобразования даты из формата YYYY-MM-DD в формат DD.MM.YYYY | Строка: 2024-11-26. | Строка: 26.11.2024 | assertEquals(expectedFormattedDate, actualFormattedDate) | + |

* 1. **Тестирование пользовательского интерфейса на экранах авторизации и регистрации.**

В приложении при регистрации и авторизации есть определенное количество проверок полей, перед отправкой запроса на сервер, такие как: соответствие почты паттерну, наличие в никнейме только символов латиницы и кириллицы, пароли должны совпадать и так далее. Для того, чтобы пользователь понимал, какие проверки не были пройдены, отображается всплывающее сообщение пользователю в зависимости от ошибки. Корректность отображение этих сообщений и будет проверено. Также необходимо проверить контекст приложения и отображение экранов (хотя-бы 1). В Таблице 6 ниже указаны тест-кейсы.

Таблица 6. Тест-кейсы для ui-tests.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Описание** | **Входные данные** | **Ожидаемый результат** | **Проверка** | **Прошел** |
| 1 | Верификация корректного контекста приложения | Нет прямых входных данных (внутренний вызов) | Получен контекст приложения с правильным именем пакета | Утверждение, что имя пакета контекста приложения равно «com.example.studybuddy» (assertEquals) | + |
| 2 | Проверка отображения элементов на экране авторизации | Запуск экрана авторизации (Auth) | Отображение заголовка «Умное планирование учёбы», описания, кнопок «ВОЙТИ» и «СОЗДАТЬ ПРОФИЛЬ». | assertIsDisplayed() для каждого из элементов: заголовка, описания и двух кнопок. | + |
| 3 | Проверка отображения ошибки при пустых полях входа | Запуск экрана логина (Login) с пустыми полями логина и пароля | Отображение сообщения об ошибке «Не все поля заполнены» после нажатия кнопки «ВОЙТИ». | 1. Нажатие кнопки «ВОЙТИ» (performClick()).  2. Проверка отображения сообщения об ошибке (isDisplayed()). | + |
| 4 | Проверка обработки некорректного email при входе | Email: «erwer», пароль: «12345678» | Отображение сообщения об ошибке «Неверный формат почты» | 1. Ввод некорректного email.  2. Нажатие кнопки «ВОЙТИ». 3. Проверка отображения сообщения об ошибке (isDisplayed()). | + |
| 5 | Проверка обработки некорректного nickname при регистрации | Email: «[user@email.com](mailto:user@email.com)», пароль: «12345678», подтверждение пароля: «12345678», nickname: «NiCk0Name1» | Отображение сообщения об ошибке «В никнейме недопустимые символы» | 1. Ввод данных. 2. Нажатие кнопки «СОЗДАТЬ ПРОФИЛЬ». 3. Проверка отображения сообщения об ошибке (isDisplayed()). | + |
| 6 | Проверка обработки несовпадения паролей при регистрации | Email: «[user@email.com](mailto:user@email.com)», пароль: «12345678», подтверждение пароля: «1345678», nickname: «NiCkname» | Отображение сообщения об ошибке «Пароли не совпадают» | 1. Ввод данных. 2. Нажатие кнопки «СОЗДАТЬ ПРОФИЛЬ». 3. Проверка отображения сообщения об ошибке (isDisplayed()). | + |