НИТУ «МИСиС»

Кафедра **инженерной кибернетики**

Квалификация (степень): **магистр**

Направление подготовки: 09.04.03 «Прикладная информатика»

Профиль (программа) «**Инновационные ИТ проекты**»

Дисциплина: Современные инструментальные средства разработки

КУРСОВАЯ РАБОТА

на тему

«Проектирование и разработка мобильного приложения под Android для тестирования уровня знаний пользователей в определенных областях информационных технологий»

Выполнили:

Системный аналитик: Денисова Е.А.
Системный аналитик: Небуко А.В.
Аналитик-дизайнер: Алексеева Е.А.
Разработчик клиентской части: Гахраманова С.З.
Разработчик серверной части: Кирсанов Г.В.
Группа: МПИ-20-4-2

Проверил: Тарханов И.А.

Оценка: _____

Дата: _____

Москва 2021

Оглавление

1. Общие сведения

1.1. Полное наименование системы и ее краткое описание

Полное наименование: Duckest

Краткое описание: Мобильное приложение под Android для тестирования уровня знаний пользователей в определенных областях информационных технологий. Предлагается пройти тестирование по таким направлениям, как «Java», «JavaScript», «SQL», «Python». Тесты внутри категории ранжированы по уровням. По завершении каждого уровня выдается Сертификат Duckest, подтверждающий знания в соответствующей области информационных технологий.

1.2. Исполнители проекта

Бизнес-аналитик: Денисова Екатерина

Системный аналитик: Небуко Анна

Аналитик-дизайнер: Алексеева Екатерина

Разработчик клиентского приложения: Гахраманова Салима

Разработчик серверной части: Кирсанов Глеб

1.3. Цель создания системы

Тестирование уровня профессиональных знаний пользователей системы в определенных областях информационных технологий.

1.4. Задачи проекта

1. Предоставление возможности пользователю зарегистрироваться и пройти тесты по разным направлениям и уровням;
2. Оценивание результата пройденного пользователем теста;
3. Выдача пользователю сертификата в случае успешного прохождения теста;
4. Способность создавать и удалять тесты.

2. Функциональные требования

1. Регистрация

Доступ в систему должен быть доступен только после процедуры регистрации. Пользователю должна быть предоставлена специальная форма, в которую ему нужно будет ввести свои ФИО, e-mail, пароль и подтверждение пароля. Далее осуществляется отправка на указанную пользователем почту письма с инструкцией для подтверждения адреса электронной почты. После перехода по ссылке открывается окно, сообщающее пользователю, что регистрация аккаунта произведена успешно.

2. Авторизация

Пользователю при входе в приложение предоставлена форма, куда ему нужно ввести логин (e-mail) и пароль. Также есть возможность восстановления пароля. Если пароль и логин являются верными, пользователь входит в систему. Если нет, то появляется соответствующее уведомление об ошибке.

3. Восстановление пароля

На экране авторизации пользователя находиться кнопка «Забыли пароль?». При нажатии на нее пользователь попадает на форму восстановления пароля, при этом ему доступно поле для ввода email. После ввода email и нажатия кнопки «Восстановить пароль» на почту пользователя отправляется инструкция по восстановлению пароля.

Пользователю необходимо ввести поле «Новый пароль» и нажать “Сохранить”. После отправки указанных данных произойдет смена пароля.

4. Настройки

При переходе в раздел «Настройки» пользователь попадает на страницу с возможностями изменения ФИО и пароля. Пользователь может изменить имя, отчество и фамилию, введя новую информацию в соответствующее поле и нажав сохранить.

Для изменения пароля необходимо ввести старый пароль, новый пароль и подтвердить новый пароль. Пароль сохраняется после нажатия на кнопку «Сохранить».

5. Выбор теста

При входе в приложение пользователю открывается раздел «Тесты». В этом разделе размещены доступные к прохождению тесты. При нажатии на тест пользователь переходит на экран выбора уровня теста (уровни «Junior», «Middle» и «Senior»). После прохождения трех уровней на карточке с тестом серая галочка меняет цвет на зеленый. Такой функционал действует на каждом уровне теста.

6. Прохождение теста

Пользователю предлагается ознакомиться с инструкцией к тесту, а также с описанием теста и условиями его успешного прохождения. Все вопросы имеют только один правильный ответ. У пользователя нет возможности сделать скриншот вопроса. После прохождения теста пользователю показаны результат тестирования в процентах. При условии успешного прохождения теста (80% правильных ответов), пользователю выдается сертификат.

7. Поделиться сертификатом

После успешного прохождения уровня пользователь имеет возможность скачать сертификат и поделиться им в иные приложения. Сертификат выдается после прохождения теста на 80%.

8. Обратная связь

Возможность обращения в техническую поддержку по проблемам, связанным с работой системы. Пользователь, в случае возникновения технических неполадок, имеет возможность заполнить форму с сообщением о своей проблеме. Пользователю необходимо ввести тему сообщения, текст сообщения и нажать на кнопку «Отправить». После нажатия кнопки «Отправить» система перебрасывает пользователя на почтового клиента.

3. Нефункциональные требования

1. Требования к персоналу системы

Для работы с системой от пользователей не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы со смартфонами.

2. Требования к надежности

Приложение должно обеспечивать целостность и непротиворечивость хранимых данных при любых действиях конечных пользователей.

Приложение должно обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях приложение должно выдавать пользователю соответствующие сообщения об ошибках.

При условии нормального режима функционирования подсистема доступна пользователям круглосуточно.

При сбоях в системе электроснабжения аппаратной части, при ошибках в работе аппаратных средств, при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), при ошибках, связанных с прикладным программным обеспечением программный комплекс должен автоматически восстанавливать свою работоспособность после устранения сбоев.

3. Безопасность персональных данных

Паролем для создания новой учетной записи или обновления пароля существующей учетной записи является цифровой код, состоящий, как минимум, из 6 цифр. Запрос разрешений (permissions) на определенные виды активности приложения обязателен и должен выполняться явно для пользователя с разъяснением, для чего именно будет использовано разрешение. При сохранении сертификата у пользователя должно запрашиваться разрешение на доступ к хранилищу. В релизной сборке приложения должны быть отключены все отладочные функции,

например опция debuggable, во избежание возможности подключения к программе внешним отладочным приложением.

4. Производительность

Время ответа системы на действия пользователя не должно превышать 3 секунд.

Допустимое количество пользователей, использующих приложение одновременно, не должно превышать 300 человек.

5. Требования к пользовательскому интерфейсу

Наличие выдвигающегося бокового меню. Предусмотрены разделы выдвигающегося бокового меню: Тесты, Обратная связь, Настройки (сменить пароль/логин), Выход.

Интерфейс системы должен поддерживать русский язык.

Интерфейс системы должен обеспечивать наглядное, интуитивно понятное представление структуры размещенной информации, быстрый и логичный переход к соответствующим разделам системы.

В целом дизайн и удобство интерфейса должны быть на уровне ожиданий современного пользователя, имеющего опыт работы с образцами, подобных сервисов.

Дизайн приложения должен быть адаптивный и работать корректно на всех устройствах с разрешением экрана 720x1280, 1080x1920, 1440×2560, 1440×3040 пикселей.

6. Требования к девайсу пользователя

Android 7.0 и выше; minimum SDK: API 24: Android 7.0

Для установки приложения необходимо 31 Мб свободного пространства внутренней памяти телефона.

7. Требования к структурам запроса на создание и удаление тестов

Чтобы создать тест, редактор теста отправляет POST-запрос на адрес *{адрес веб-приложения}/test* следующей структуры:

{

```
"test_type": "javascript",
"test_level": "junior",
"description": "dummy description",
"questions": [
  {
    "question": "dummy question?",
    "right_answer_index": 1,
    "answers": ["First dummy answer", "Second dummy answer", "Third dummy answer"]
  }
]
}
```


Это необходимый минимум полей для успешного создания теста для языка программирования javascript уровня junior с одним вопросом и тремя вариантами ответа.

Удаление теста происходит по тому же адресу *{адрес веб-приложения}/test*, но http метод должен быть DELETE. Структура запроса:


```
{
  "test_type": "javascript",
  "test_level": "junior"
}
```


Ожидаемым результатом является удаление из базы данных всего, что связано с тестом, включая вопросы и порог прохождения, а также прогресс пользователей по этому тесту.

4. Технические требования

1. Требования к среде разработки

Среды разработки:

- Back-end: Intellij Idea
- Front-end: Android Studio

2. Требования к используемому стеку технологий и языкам программирования

Стек технологий для front-end: kotlin coroutines, retrofit, dagger hilt. Язык программирования - kotlin.

Стек технологий для back-end: spring, Hibernate, Heroku, PostgreSQL и Liquibase. Язык программирования - java.

3. Требования к отправке теста на серверную часть, удалению теста с серверной части

Для создания, редактирования и удаления теста необходимо воспользоваться http клиентом, например, Postman.

Для создания текста редактор должен отправить POST-запрос на адрес *{адрес веб-приложения}/test*. Удаление теста происходит по тому же адресу, но вместо метода POST используется DELETE.

4. Требования к размещению серверной части

Для размещения веб-приложения в сети Интернет будет использоваться облачная PaaS-платформа для хостинга сервисов Heroku. На этом же сервисе будет расположена база данных приложения, которая будет обновляться скриптами миграции.

5. Сценарии использования

1. Регистрация

Название прецедента: регистрация нового пользователя в приложении

Действующее лицо: новый пользователь

Цель: добавить пользователя в приложение

Предусловия: пользователь заходит в приложение

Успешный сценарий:

1. Пользователь открывает приложение и появляется форма входа в приложение, где нужно ввести логин и пароль и нажать кнопку «Войти». Также на форме есть кнопка «Еще нет аккаунта? Зарегистрируйтесь»;
1. Пользователь выбирает возможность зарегистрироваться;
2. Система показывает форму для регистрации, содержащую поля для заполнения: ФИО, «E-mail», «Пароль», «Подтвердите пароля», а также кнопку «Сохранить»;
3. После ввода всех данных и нажатия на кнопку «Сохранить» система отправляет на указанную почту письмо со ссылкой для подтверждения регистрации;
4. Пользователь переходит по ссылке из письма;
5. Система сообщает об успешной регистрации.

Альтернативный сценарий:

1. Если пользователь указывает почту, которая уже занята:
   1. Система сообщает об ошибке.
2. Если произошел сбой в работе приложения:
   1. Пользователю отображается уведомление о том, что произошел сбой, при этом данные, введенные в форму для регистрации, не сохраняются.

2. Авторизация

Название прецедента: авторизация пользователя в приложении

Действующее лицо: зарегистрированный пользователь

Цель: вход в приложение

Предусловия: пользователь заходит в приложение

Успешный сценарий:

1. Пользователь открывает приложение и видит форму с полями: «Логин», «Пароль» и кнопками «Войти», «Забыли пароль», «Зарегистрируйтесь».
2. Пользователь вводит логин и пароль и нажимает кнопку «Войти»;
3. Открывается меню «Тесты».

Альтернативный сценарий:

1. Если пользователь ввел неверный логин или пароль:
    1. Система выводит сообщение об ошибочном логине/пароле.

3. Прохождение теста

Название прецедента: прохождение пользователем теста

Действующее лицо: зарегистрированный пользователь

Цель: пройти тест

Предусловия: пользователь зашел в приложение и выбрал тест

Успешный сценарий:

1. Пользователь выбирает тест из меню «Тесты»;
2. Пользователь выбирает уровень теста;
3. Система предлагает пользователю ознакомиться с инструкцией к тесту;
4. Пользователь нажимает на кнопку «Начать тестирование»;
5. Система показывает пользователю страницу, где написаны номер вопроса, сам вопрос теста и даны варианты ответа, а также есть кнопка «Ответить»;

6. Пользователь выбирает вариант ответа и нажимает «Ответить» для перехода к следующему вопросу;
7. Пользователь проходит тестирование на 80%;
8. Система показывает пользователю страницу, где показан результат тестирования в процентах;
9. Система генерирует сертификат о прохождении теста, который можно нажать для сохранения в галерее.
10. Появляется иконка «Поделиться» в правом верхнем углу.

Альтернативный сценарий:

1. Если пользователь прошел тест менее чем на 80%:
    1. Система выводит сообщение о неуспешном прохождении теста;
    2. Система предлагает пройти тест еще раз.

4. Отправка формы обратной связи

Название прецедента: отправка формы обратной связи

Действующее лицо: зарегистрированный пользователь

Цель: отправить форму обратной связи

Предусловия: пользователь заходит в свой профиль

Успешный сценарий:

1. Пользователь нажимает на раздел бокового меню «Обратная связь»;
2. Система предлагает пользователю заполнить форму;
3. Пользователь вводит тему сообщения;
4. Пользователь вводит сообщение;
5. Пользователь нажимает на кнопку «Отправить»;
6. Система предлагает пользователю отправить сообщение через почтовые клиенты;
7. Пользователь выбирает почтовый клиент;
8. Система перебрасывает пользователя на почтового клиента;
9. Пользователь нажимает на кнопку «Отправить»;
10. Почтовый клиент отправляет сообщение;

Альтернативный сценарий:

1. Если пользователь не заполнил все поля формы и нажал на кнопку «Отправить»:

   1.Система показывает сообщение «Это поле обязательно для заполнения»

2. Если пользователь не стал выбирать почтового клиента и нажал «Отмена»:

   1. Система закрывает форму.

5. Отправка сертификата в иные приложения и сохранение его на телефон

Название прецедента: Отправка результата в социальные сети и сохранение его на телефон

Действующее лицо: зарегистрированный пользователь

Цель: отправить результат в социальные сети или сохранить его на телефон

Предусловия: пользователь успешно прошел тестирование

Успешный сценарий:

1. Пользователь нажимает на иконку «Поделиться»;
2. Система предлагает пользователю выбрать приложение-адресата;
3. Пользователь выбирает приложение;
4. Пользователь нажал кнопку опубликовать/отправить;
5. Система отправляет сертификат.

Расширения:

1. Если пользователь нажал на изображение сертификата:

   1.Система загружает сертификат на телефон

2. Если пользователь нажимает на кнопку «Отмена» на любом этапе отправки сертификата в социальную сеть:

   1.Система возвращает пользователя на этап получения сертификата.

6. Изменение настроек

Название прецедента: изменение настроек

Действующее лицо: зарегистрированный пользователь

Цель: изменить настройки

Предусловия: пользователь заходит в свой профиль

Успешный сценарий:

1. Пользователь нажимает на раздел бокового меню «Настройки»;
2. Система предоставляет пользователю возможность изменения ФИО и пароля;
3. Пользователь вводит текущий пароль и дважды новый пароль в соответствующих полях;
4. Пользователь нажимает на кнопку «Сохранить»;
5. Система изменяет пароль;
6. Система показывает уведомление об изменении пароля;

Расширения:

1. Если пользователь изменяет ФИО:
    1. Пользователь вводит новое ФИО;
    2. Система сохраняет изменения.
2. Если пользователь ввел текущий пароль неправильно:
    1.Система сообщает «Вы ввели неверный пароль»;
    2. Система возвращается на 3-ый этап.
3. Если дважды введенный пользователем новый пароль не идентичен:
    1.Система сообщает «Не совпадают пароли»;
    2. Система возвращается к 3-ому этапу.

6. Сценарии тестирования

Тест-кейс №1

Номер	1
Название	Регистрация
Предусловие	Приложение открыто
Шаг	**Ожидаемый результат**
Заполнить поле «Имя», «Фамилия», «Отчество» буквами русского алфавита	В соответствующих полях отображаются введенное значение
Заполнить поле «Имя», «Фамилия», «Отчество» цифрами	Ошибка – «Это поле может содержать только буквы русского алфавита»
Заполнить поле «Имя», «Фамилия», «Отчество» латинскими буквами	Ошибка – «Это поле может содержать только буквы русского алфавита»
Ввести незарегистрированный и корректный e-mail в поле «E-mail»	Система сохраняет e-mail после нажатия «Сохранить»
Ввести e-mail, уже зарегистрированный в системе	Ошибка сохранения e-mail после нажатия «Сохранить»
Ввести пароль, состоящий из 6 символов и более	В поле «Пароль» отображается пароль, зашифрованный звездочками
Ввести пароль, состоящий из 5 символов и менее	Ошибка – «Пароль должен содержать 6 и более знаков»
Подтвердить пароль	В поле «Подтвердить пароль» отображается пароль, зашифрованный звездочками
Ввести в поле подтверждения пароля пароль, не идентичный с	Ошибка – «Не совпадают пароли»

ранее введенным паролем	
Оставить какое-либо поле пустым (кроме поля «Отчество»)	Ошибка – «Это поле обязательно для заполнения».
Нажать на кнопку «Зарегистрироваться», находящуюся под полями	Появляется сообщение «На ваш адрес отправлено письмо с уникальной ссылкой для подтверждения почты».
Проверить почту	На почту пришло сообщение, содержащее ссылку, ведущую в приложение
Перейти по ссылке из почты	Сообщение об успешной регистрации

Тест-кейс №2

Номер	2
Название	Авторизация
Предусловие	Приложение открыто, и пользователь имеет в нем аккаунт
Шаг	**Ожидаемый результат**
Ввести свой логин в поле «Логин»	В поле «Логин» отображается введенное значение
Ввести свой пароль	В поле «Пароль» отображается пароль, зашифрованный звездочками
Ввести неверный пароль	Ошибка – «Вы ввели неверный пароль»
Ввести неверный e-mail	Ошибка – «Пользователь не найден»
Нажать кнопку «Войти», находящуюся под полями	Пользователь заходит в свой профиль

Тест-кейс №3

Номер	3
Название	Выбор теста для прохождения
Предусловие	Пользователь зарегистрировался в приложении или вошел в уже существующий аккаунт
Шаг	**Ожидаемый результат**
Открыть меню приложения	Появляется меню приложения
Выбрать пункт меню «Тесты»	Открывается список тестов, доступных для прохождения
Выбрать подходящий тест	Открывается окно выбора уровня теста
Выбрать подходящий уровень	Открывается описание теста

Тест-кейс №4

Номер	4
Название	Ответ на вопрос в процессе прохождения теста
Предусловие	Открытый выбранный тест
Шаг	**Ожидаемый результат**
Перейти на вопрос теста	Появляется текст вопроса и варианты ответа
Выбрать вариант ответа	Вариант ответа выбран и отмечен
Нажать на кнопку «Ответить» и перейти к следующему вопросу	Открытый следующий вопрос
Нажать на кнопку «Ответить», не выбрав вариант ответа	Ошибка – «Выберите вариант ответа»

Тест-кейс №5

Номер	5
Название	Закрытие теста до его завершения
Предусловие	Открытый выбранный тест
Шаг	**Ожидаемый результат**
Открыть тест	Открытое окно с вопросом теста и стрелочкой в левой верхней части экрана для закрытия окна
Нажать на стрелочку в левой верхней части экрана	Диалог «Вы хотите прервать прохождение теста? Данные не будут сохранены» и кнопками «Да» и «Нет»
Нажать на кнопку «Да»	Закрытое окно теста и открытая главная страница с тестами

Тест-кейс №6

Номер	6
Название	Прохождение теста более чем на 80%
Предусловие	Пользователь ответил на 80% и более вопросов в тесте правильно
Шаг	**Ожидаемый результат**
Ответить на последний вопрос теста и нажать «Завершить тест»	Открытая страница, где показан результат тестирования в процентах и иконка сертификата.
Нажать на иконку сертификата	В правом верхнем углу появляется иконка «Поделиться»
Нажать на стрелочку для закрытия страницы	Возвращение на изначальную страницу со списком тестов

Тест-кейс №7

Номер	7
Название	Прохождение теста менее чем на 80%
Предусловие	Результаты теста показывают более 20% неверных ответов
Шаг	**Ожидаемый результат**
Ответить на последний вопрос теста и нажать «Завершить тест»	Открытая страница, где написан результат теста в процентах. Показано сообщение «Не расстраивайтесь», кнопка «Попробовать еще раз» и стрелочка для закрытия теста.
Нажать «Попробовать еще раз»	Открытая заглавная страница теста
Нажать стрелочку для закрытия теста	Открытая страница с меню тестов

Тест-кейс №8

Номер	8
Название	Отправка формы обратной связи
Предусловие	Зарегистрироваться в системе
Шаги:	**Ожидаемый результат**
Нажмите на раздел бокового меню «Обратная связь»	Появляется форма обратной связи
Введите тему сообщения и текст сообщения с описанием проблемы	Все верно, форма обратной связи отправляется в службу поддержки
<Оставить поле пустым>	Ошибка – «Это поле обязательно для заполнения»
Ввести текст с количеством	Ошибка – «Недостаточное

символов меньше 30	количество символов»
Нажмите на кнопку «Отправить»	Перенаправление пользователя на почтового клиента

Тест-кейс №9

Номер	9
Название	Отправка сертификата в иные приложения и сохранение его на телефон
Предусловие	Пользователь ответил на 80% и более вопросов в тесте правильно
Шаги:	**Ожидаемый результат**
Нажать на иконку «Поделиться»	Появляется окно с возможностью выбора приложения-адресата
Выбрать приложение	Открывается выбранное приложение
Опубликовать/отправить	Сертификат отправлен
Нажать на изображение сертификата	Скачивание сертификата в галерею

Тест-кейс №10

Номер	10
Название	Изменение в настройках фамилии, имени, отчества
Предусловие	Зарегистрироваться в системе
Шаги:	**Ожидаемый результат**
Нажать на раздел меню «Настройки»	Переход в раздел «Настройки»
Изменить имя	В поле «Имя» отображается введенное значение
Изменить отчество	В поле «Отчество» отображается введенное значение

Изменить фамилию	В поле «Фамилия» отображается введенное значение
<Оставить поле пустым>	Ошибка – «Это поля обязательно для заполнения»
Ввести 1*6*0	Ошибка – «Это поле может содержать только буквы русского алфавита»
Ввести в одно из полей ФИО латинские буквы	Ошибка – «Это поле может содержать только буквы русского алфавита»
Нажать на кнопку «Сохранить»	Изменения сохранены

Тест-кейс №11

Номер	11
Название	Изменение в настройках текущего пароля
Предусловие	Зарегистрироваться в системе
Шаги:	**Ожидаемый результат**
Нажать на раздел меню «Настройки»	Переход в раздел «Настройки»
Ввести текущий пароль правильно	В поле «Текущий пароль» отображается введенное значение, закрытое звездочками
Ввести текущий пароль неправильно	Ошибка – «Вы ввели неверный пароль»
Ввести новый пароль	В поле «Пароль» отображается введенное значение, закрытое звездочками
<Оставить поле пустым>	Ошибка – «Это поля обязательно для заполнения»

Правильно подтвердить новый пароль	В поле «Подтвердить пароль» отображается введенное значение, закрытое звездочками
<Оставить поле пустым>	Ошибка – «Это поля обязательно для заполнения»
Неправильно подтвердить новый пароль	Ошибка – «Не совпадают пароли»
Нажать на кнопке «Сохранить»	Данные сохранены

Тест-кейс №12

Номер	12
Название	Восстановление пароля
Предусловие	Система сообщает, что введенный пароль недействительный
Шаги:	**Ожидаемый результат**
Нажать на кнопку «Забыли пароль?»	Появляется форма для восстановления пароля
Ввести верный логин	Все верно, на почту отправляется инструкция по восстановлению пароля
<Оставить поле пустым>	Ошибка – «Это поля обязательно для заполнения»
Неверный логин	Ошибка – «Пользователь не найден»
Проверить электронную почту	Пришло письмо с инструкцией по восстановлению пароля
Нажать на кнопку «Не получили письмо?»	Появляется форма для восстановления пароля
Создать новый пароль	Пароль восстановлен

7. Прототип основных окон системы

В прототипе представлено следующие формы:

1. Иконка
2. Окно загрузки
3. Регистрация
4. Авторизация
5. Меню
6. Восстановление пароля
7. Обратная связь
8. Настройки
9. Меню тестов
10. Прохождение тестирования
11. Результаты теста
12. Сертификат

Рис. 1 Иконка

Рис. 2 Окно загрузки

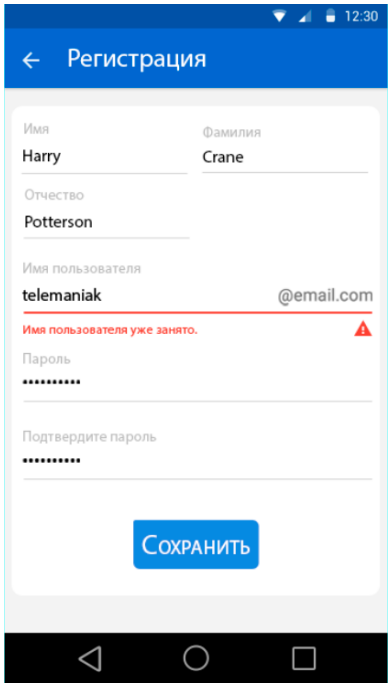

Рис. 3 Регистрация

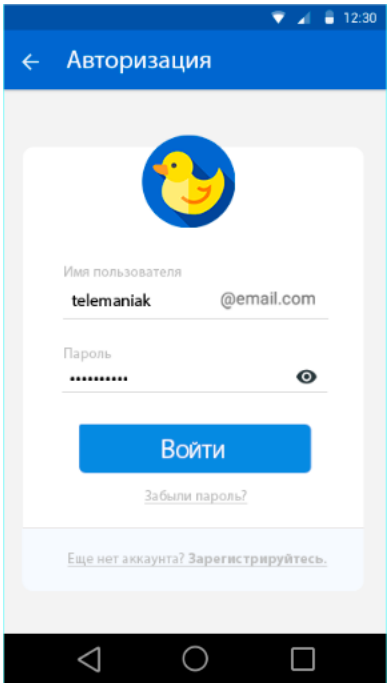

Рис. 4 Авторизация

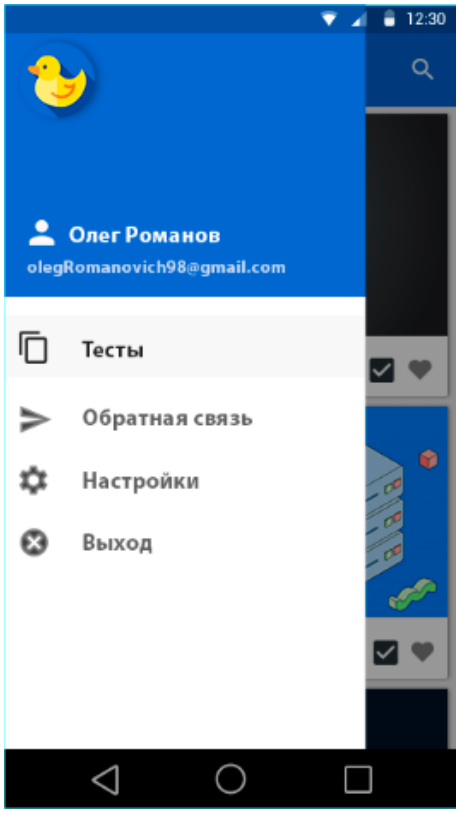

Рис.5 Меню

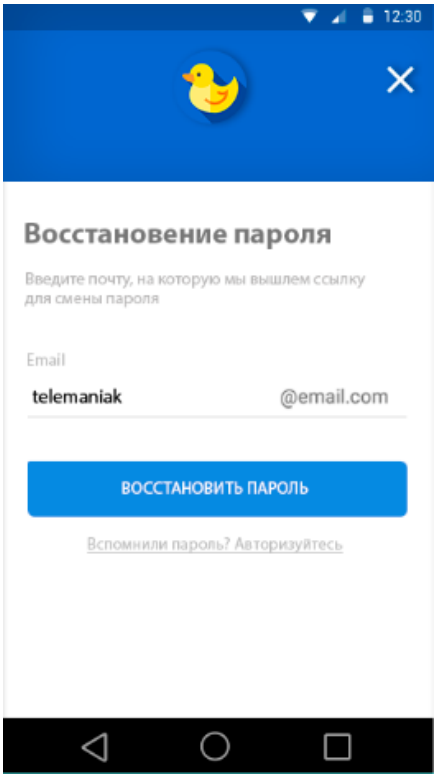

Рис. 6.1. Восстановление пароля

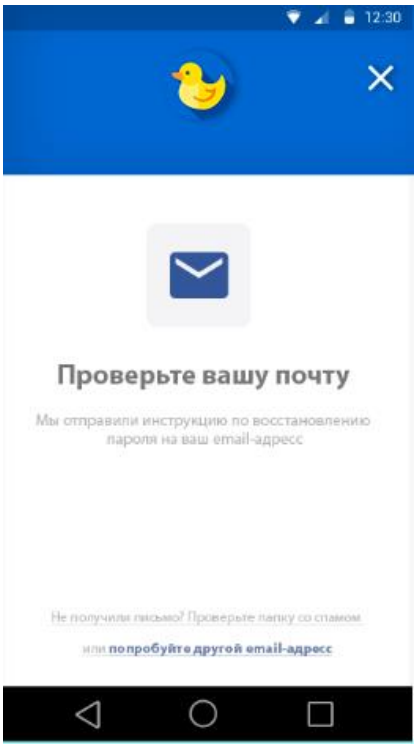

Рис. 6.2. Восстановление пароля

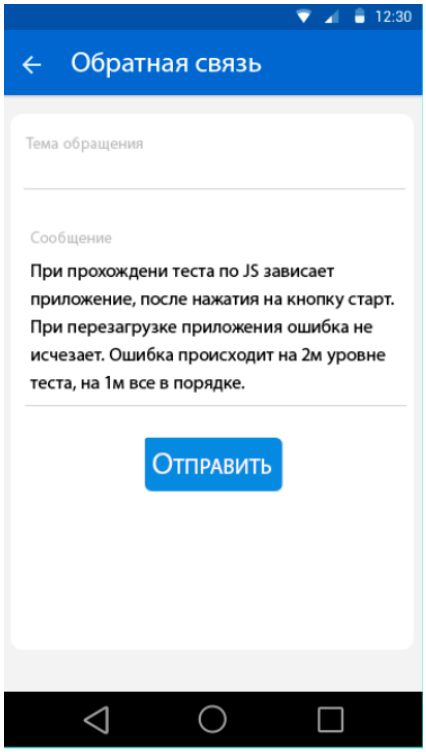

Рис.7. Обратная связь

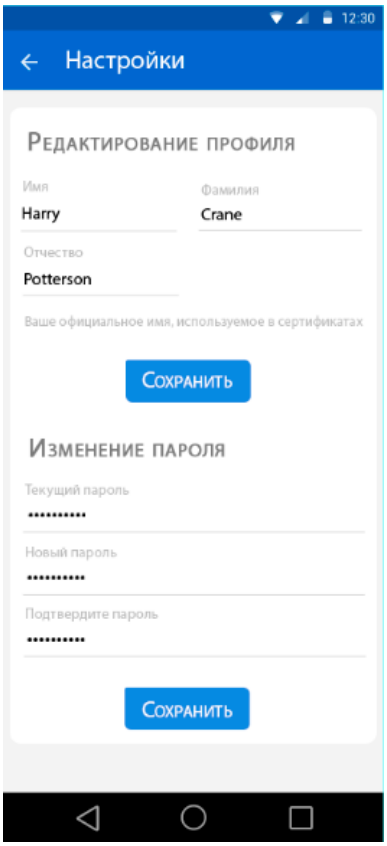

Рис.8 Настройки

Рис.9.1 Меню тестов

Рис.9.2 Меню тестов

Рис. 10.1 Прохождение тестирования

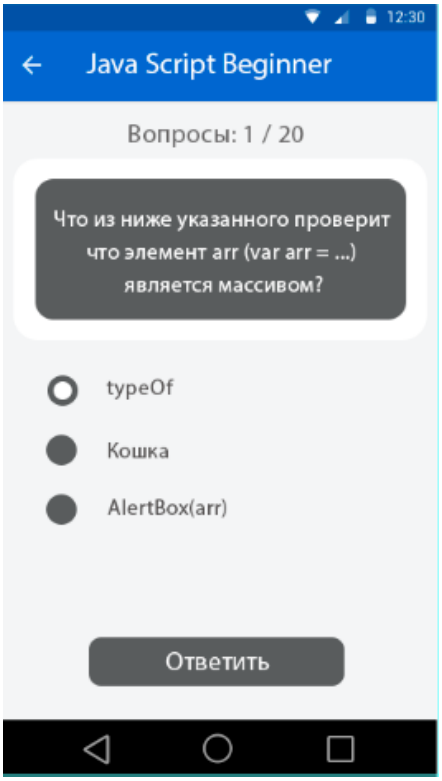

Рис. 10.2 Прохождение тестирования

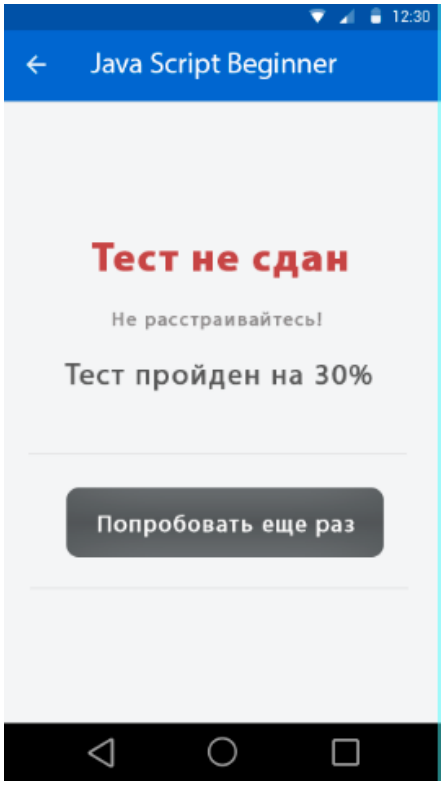

Рис. 11.1 Результаты теста

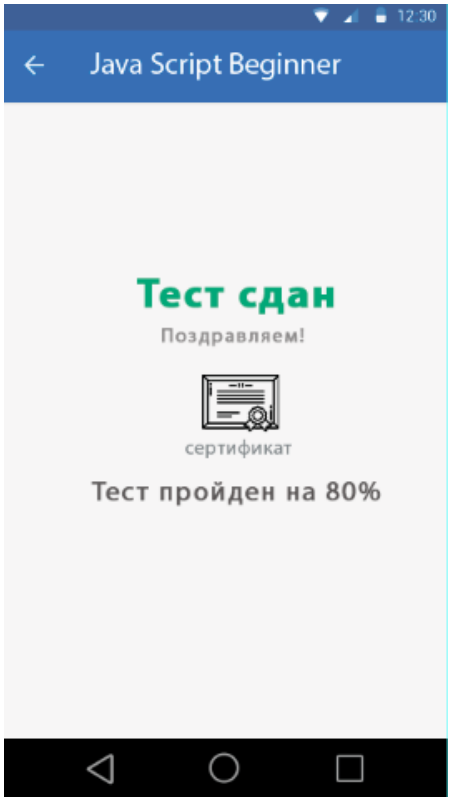

Рис. 11.2 Результаты теста

Рис.12 Сертификат

8. Листинг кода клиентской части

```
package com.duckest.duckest.ui.home.test

import android.app.AlertDialog
import android.os.Bundle
import android.view.LayoutInflater
import android.view.MenuItem
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.activity.addCallback
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment
import androidx.fragment.app.viewModels
import androidx.navigation.fragment.findNavController
import com.duckest.duckest.R
import com.duckest.duckest.data.NetworkResult
import com.duckest.duckest.data.domain.TypeLevelPair
import com.duckest.duckest.databinding.FragmentTestBinding
import dagger.hilt.android.AndroidEntryPoint

@AndroidEntryPoint
class TestFragment : Fragment() {
    private lateinit var binding: FragmentTestBinding
    private lateinit var args: TestFragmentArgs
    private val vm: TestViewModel by viewModels()

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        args = TestFragmentArgs.fromBundle(requireArguments())
        binding = FragmentTestBinding.inflate(inflater, container, false)
        (activity as AppCompatActivity).supportActionBar?.title = args.testType
        setHasOptionsMenu(true)
        return binding.root
    }
```

```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    binding.testViewModel = vm
    binding.lifecycleOwner = viewLifecycleOwner
    hideViews()
    vm.getTest(
        TypeLevelPair(
            args.testType,
            args.testLevel
        )
    )

    vm.eventTestFinish.observe(viewLifecycleOwner) {
        when(it) {
            is NetworkResult.Error -> {
                binding.progressBar.visibility = View.GONE
                binding.nextQuestion.isEnabled = true
                Toast.makeText(
                    requireContext(),
                    it.message,
                    Toast.LENGTH_LONG
                ).show()
            }
            is NetworkResult.Loading -> {
                disableViews()
                binding.progressBar.visibility = View.VISIBLE
            }
            is NetworkResult.Success -> {
                binding.progressBar.visibility = View.GONE
                if (it.data!!.first) {
                    findNavController().navigate(

TestFragmentDirections.actionTestFragmentToTestPassedFragment(
                        it.data.second,
                        args.testType,
                        args.testLevel
                    )
                )
```

```
                } else {
                    findNavController().navigate(

TestFragmentDirections.actionTestFragmentToTestFailFragment(
                            it.data.second,
                            args.testType,
                            args.testLevel,
                            args.imageUrl
                        )
                    )
                }
            }
        }

        vm.response.observe(viewLifecycleOwner) {
            when (it) {
                is NetworkResult.Loading ->  {
                    binding.progressBar.visibility = View.VISIBLE
                }
                is NetworkResult.Success -> {
                    showViews()
                    binding.progressBar.visibility = View.GONE
                    vm.nextQuestion()
                }
                is NetworkResult.Error -> {
                    binding.progressBar.visibility = View.GONE
                    Toast.makeText(
                        requireContext(),
                        it.message,
                        Toast.LENGTH_LONG
                    ).show()
                }
            }
        }

        binding.nextQuestion.setOnClickListener {
            val checkedId = binding.questionRadioGroup.checkedRadioButtonId
            // Do nothing if nothing is checked (id == -1)
```

```
        if (-1 != checkedId) {
            var answerIndex = 0
            when (checkedId) {
                R.id.firstAnswerRadioButton -> answerIndex = 0
                R.id.secondAnswerRadioButton -> answerIndex = 1
                R.id.thirdAnswerRadioButton -> answerIndex = 2
            }
            vm.checkAnswer(answerIndex)
            binding.questionRadioGroup.clearCheck()
            vm.nextQuestion()
        } else {
            Toast.makeText(
                requireContext(),
                getString(R.string.test_message_no_answer),
                Toast.LENGTH_SHORT
            ).show()
        }
    }
    requireActivity().onBackPressedDispatcher.addCallback(
        viewLifecycleOwner,
        findNavController().previousBackStackEntry != null
    ) {
        showDialog()
    }
}

private fun hideViews() {
    binding.materialCardView.visibility = View.GONE
    binding.questionRadioGroup.visibility = View.GONE
    binding.nextQuestion.visibility = View.GONE
    binding.testTitle.visibility = View.GONE
}

private fun showViews() {
    binding.materialCardView.visibility = View.VISIBLE
    binding.questionRadioGroup.visibility = View.VISIBLE
    binding.nextQuestion.visibility = View.VISIBLE
    binding.testTitle.visibility = View.VISIBLE
}
```

```
    private fun disableViews() {
        binding.materialCardView.isEnabled = false
        binding.questionRadioGroup.isEnabled = false
        binding.nextQuestion.isEnabled = false
        binding.testTitle.isEnabled = false
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return when (item.itemId) {
            android.R.id.home -> {
                showDialog()
                true
            }
            else -> {
                super.onOptionsItemSelected(item)
            }
        }
    }

    private fun showDialog() {
        val builder = AlertDialog.Builder(requireContext())
        builder.apply {
            setTitle(getString(R.string.exit))
            setMessage(getString(R.string.exit_message))
            setPositiveButton("Да") { _, _ ->

findNavController().navigate(TestFragmentDirections.actionGlobalFeedFragment(
))
            }
            setNegativeButton("Нет") { _, _ ->
            }
            show()
        }
    }
}
```

9. Листинг кода серверной части

UserController

```
@RestController
@RequiredArgsConstructor
@RequestMapping(path = "user")
public class UserController {

   private final UserService userService;

   @PostMapping(consumes = MediaType.APPLICATION_JSON_VALUE)
   public ResponseEntity<HttpStatus> registerUser(@Valid @RequestBody
UserDto user) {
      userService.save(user);
      return new ResponseEntity<>(HttpStatus.CREATED);
   }

   @PutMapping(consumes = MediaType.APPLICATION_JSON_VALUE)
   public ResponseEntity<HttpStatus> registerUserData(@Valid
@RequestBody UserDto user) {
      userService.update(user);
      return ResponseEntity.ok(null);
   }

   @GetMapping
   public ResponseEntity<UserDto> getUserBy(@RequestParam("email")
String email) {
      UserDto user = userService.getUserBy(email);
      return ResponseEntity.ok(user);
   }
}
```

UserService (интерфейс)

```
package ru.duckest.duckest.service;
import ru.duckest.duckest.dto.UserDto;
public interface UserService {

   void save(UserDto user);
   UserDto getUserBy(String email);
   void update(UserDto user);
}
```

UserServiceImpl (реализация интерфейса)

```
@Service
@RequiredArgsConstructor
public class UserServiceImpl implements UserService {

   private final UserSaver userSaver;
   private final UserSelector userSelector;
   private final UserUpdater userUpdater;
   private final UserConverter converter;
   private final EmailValidator emailValidator = EmailValidator.getInstance();

   @Override
   public void save(UserDto user) {
      var convertedUser = converter.convert(user);
      userSaver.save(convertedUser);
   }

   @Override
   public UserDto getUserBy(String email) {
      if (!emailValidator.isValid(email)) {
         throw new IllegalArgumentException("Invalid email when getting
user");
      }
      var user = userSelector.getUserBy(email);
      return converter.convert(user);
   }

   @Override
   public void update(UserDto user) {
      var userEntity = userSelector.getUserBy(user.getEmail());
      userEntity.setEmail(user.getEmail());
      userEntity.setFirstName(user.getFirstName());
      userEntity.setLastName(user.getLastName());
      userEntity.setMiddleName(user.getMiddleName());
      userUpdater.update(userEntity);
   }
}
```

Список используемых источников

Нормативные акты

1. "ГОСТ 34.602-89. Межгосударственный стандарт. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы"// М.: ИПК Издательство стандартов, 2004;
2. "ГОСТ 19.101-77* (СТ СЭВ 1626-79). Государственный стандарт Союза ССР. Единая система программной документации. Виды программ и программных документов"// СПС КонсультантПлюс;
3. "IEEE Recommended Practice for Software Requirements Specifications," in *IEEE Std 830-1998* , vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286

Электронные ресурсы

4. Aminullah Taj Muhammad. Share Image and Text using android Intent// Сайт AndroidPub. 11.05.2020г. URL: https://medium.com/android-news/share-image-and-text-using-android-intent-2d5bc87ca09e (дата обращения: 10.04.2020).
5. Boyarsky Jeanne, Selikoff Scott. OCP: Oracle Certified Professional Java SE 8 Programmer II Study Guide//Personal website Boyarsky Jeanne 2015. URL: https://www.selikoff.net/ocp11-complete/ (дата обращения: 01.05.2020).
6. Boyarsky Jeanne, Selikoff Scott. OCA: Oracle Certified Associate Java SE 8 Programmer I Study Guide// Personal website Boyarsky Jeanne 2015. URL: https://www.selikoff.net/java-oca-8-programmer-i-study-guide/ (дата обращения: 02.05.2020).
7. Google. Save key-value data// Сайт Developers. 2021г. URL:https://developer.android.com/training/data-storage/shared-preferences (дата обращения: 01.04.2020).

8. Google. Dependency injection with Hilt// Сайт Developers. 2021г. URL: https://developer.android.com/training/dependency-injection/hilt-android (дата обращения: 07.04.2020).
10. Google. Hilt and Jetpack integrations// Сайт Developers. 2021г. URL: https://developer.android.com/training/dependency-injection/hilt-jetpack (дата обращения: 10.04.2020).
11. Google. Sending simple data to other apps // Сайт Developers. 2021г. URL: https://developer.android.com/training/sharing/send (дата обращения: 14.03.2020).
12. Google. Data and file storage overview // Сайт Developers. 2021г. URL: https://developer.android.com/training/data-storage(дата обращения: 15.04.2020).
13. Google. Android Internal Storage Example Tutorial//Сайт JoutnalDev. 27.03.2016г. URL: https://www.journaldev.com/9383/android-internal-storage-example-tutorial (дата обращения: 10.04.2020).
14. Pompeius Magnus. Пишем максимально эффективный тест-кейс//Сайт Хабр. 22.12.2014г. URL: https://habr.com/ru/post/246463/ (дата обращения: 12.04.2020)
15. Абрамова Анна. USE CASES. Что это такое и зачем они нужны?// Сайт Школа системного анализа и проектирования. 2021г. URL: https://systems.education/use-case (дата обращения: 8.04.2020)
16. Захаров Виктор. Правильно пишем тест-кейсы. Памятка начинающему специалисту по тестированию. 10.01.2020г. URL: https://victorz.ru/202001101079 (дата обращения: 10.04.2020)
17. Назина Ольга. Что такое тест-кейс и как его писать// Блог Ольги. 7.08.2014г. URL: http://okiseleva.blogspot.com/2014/08/blog-post.html (дата обращения: 13.04.2020)