*Министерство образования и науки Российской Федерации*

*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

*«Владимирский государственный университет*

*имени Александра Григорьевича и Николая Григорьевича Столетовых»*

*Кафедра информационных систем и программной инженерии*

***КУРСОВОЙ ПРОЕКТ***

*Мобильное приложение для оценки профессиональных интересов по методике И. Л. Соломина*

*Выполнили:*

*студенты гр. ПРИ-115*

*Обрубов М. О.*

*Обрубова В. Д.*

*Принял: доц. Х. М. Салех*

*Владимир, 2018*

**Аннотация**

Суть курсовой работы заключается в разработке модильного приложения для прохождения оценки профессиональных интересов по методике Соломина.

Пояснительная записка содержит:

* 39 страниц;
* 22 рисунков.

The scope of the course work is to develop applications for the assessment of professional interests by the method of Solomin.

The explanatory memorandum contains:

* 39 pages;
* 22 figures

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc516109891)

[1 ПОСТАНОВКА ЗАДАЧИ 4](#_Toc516109892)

[1.1 Описание предметной области 4](#_Toc516109893)

[1.2 Основные понятия предметной области 4](#_Toc516109894)

[1.4 Функциональные требования к системе 7](#_Toc516109895)

[1.4 Нефункциональные требования к системе 9](#_Toc516109896)

[1.5 Сравнительный анализ аналогов 10](#_Toc516109897)

[2 АНАЛИЗ ЗАДАЧИ И ФУНКЦИОНАЛЬНАЯ ДЕКОМПОЗИЦИЯ СИСТЕМЫ 13](#_Toc516109898)

[2.1 Анализ бизнес-процесса "Достижение цели" 13](#_Toc516109899)

[2.2 Сценарий взаимодействия пользователя с системой 18](#_Toc516109900)

[2.3 Авторизация пользователя в системе 19](#_Toc516109901)

[2.4 Регистрация пользователя в системе 20](#_Toc516109902)

[2.5 Изменение данных пользователя 20](#_Toc516109903)

[2.6 Добавление дефолтной цели 21](#_Toc516109904)

[2.7 Добавление своей цели 21](#_Toc516109905)

[2.8. Просмотр своих целей 22](#_Toc516109906)

[2.9.Отметка о прохождении цели 22](#_Toc516109907)

[2.10. Просмотр рейтинга пользователей 23](#_Toc516109908)

[2.11. Создание темы на форуме 23](#_Toc516109909)

[2.12. Создание сообщения на форуме 23](#_Toc516109910)

[3 СТРУКТУРНАЯ ОРГАНИЗАЦИЯ СИСТЕМЫ 25](#_Toc516109911)

[3.1 Общие принципы организации системы 25](#_Toc516109912)

[3.2 Взаимодействие с внешними сервисами/удалённым модулем для оплаты 26](#_Toc516109913)

[3.3 Организация доступа к данным 27](#_Toc516109914)

[3.4 Организация бизнес-логики 29](#_Toc516109915)

[3.5 Организация веб-интерфейса 31](#_Toc516109916)

[3.6 Взаимодействие компонентов системы для цели 36](#_Toc516109917)

[3.7 Организация обработки ошибок 38](#_Toc516109918)

[3.8 Организация управления доступом 39](#_Toc516109919)

[4 РЕАЛИЗАЦИЯ КОМПОНЕНТОВ СИСТЕМЫ 40](#_Toc516109920)

[4.1 EJB-компоненты/компоненты бизнес-логики 40](#_Toc516109921)

[4.2 Веб-компоненты 42](#_Toc516109922)

[4.3 Физическая структура баз данных 47](#_Toc516109923)

[4.4 Обеспечение целостности данных 49](#_Toc516109924)

[4.5 Реализация взаимодействия с внешним сервисом 50](#_Toc516109925)

[5 ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРИЛОЖЕНИЯ 52](#_Toc516109926)

[**5.1 Установка приложения** 52](#_Toc516109927)

[**5.2 Установка Web-сервиса** 59](#_Toc516109928)

[6 ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ ПРОЕКТОМ 64](#_Toc516109929)

[6.1 Общие принципы взаимодействия в команде 64](#_Toc516109930)

[6.2 Распределение ролей в команде/зоны ответственности 65](#_Toc516109931)

[6.3 Календарный план работ 65](#_Toc516109932)

[Рисунок 41. Диаграмма Ганта. 66](#_Toc516109933)

[7 НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ 67](#_Toc516109934)

[ЗАКЛЮЧЕНИЕ 73](#_Toc516109935)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 74](#_Toc516109936)

[ПРИЛОЖЕНИЕ А Листинг компонентов системы 75](#_Toc516109937)

[ПРИЛОЖЕНИЕ Б Скрипт создания базы данных 106](#_Toc516109938)

[ПРИЛОЖЕНИЕ B Скрипт paying-service 110](#_Toc516109939)

ВВЕДЕНИЕ

Целью данного курсового проекта является разработка мобильного приложения для определения профессиональной направленности. Человек ведь вечно озадачен вопросом в каком направлении ему двигаться, и определить свой путь ему поможет наш опросник.

Сама методика опросника профориентации была разработана петербургским психологом Игорем Леонидовичем Соломиным. Тест Соломина применяется для самооценки профессиональных интересов и способностей молодых и взрослых людей. Методика проста в обращении, не требует специальных психологических знаний. Время тестирования занимает 10-15 мин [1].

Второй из областей применения разрабатываемого мобильного приложения является сбор статистики. Актуальной проблемой любого вида человеческой деятельности является проблема обработки увеличивающегося потока информации. Мы поможем в обработке статистики о результатах прохожения тестирования, представив информацию в графическов виде и добавив фильтрацию.

Для реализации поставленной задачи будут выполнены следующие пункты:

* Изучение предметной области;
* Формирование требований к системе;
* Проектирование и разработка;
* Проверка работоспособности системы.

1. ПОСТАНОВКА ЗАДАЧИ
   1. Цель и краткое описание системы

С помощью нашего теста пользователь сможет узнать какая профессия ему больше всего подходит, он может протестировать себя и своих близких по научной методике Соломина. Он не потеряет результаты предыдущих тестирований и сможет их посмотреть в любой момент в своем личном кабинете. Данное приложение не только является развлекательным контентом, но и помогает в сборе статистических данных необходимых для экспертов, что как раз и является самоцелью разработки данного мобильного приложения.

Предполагаемой аудиторией приложения будут являться:

* Дети, школьники, ученики старших классов, которые хотят определиться с профориентацией, выбрать в какой ВУЗ поступать;
* Молодежь, студенты, которые хотят выбрать специализацию, дополнительное образование, компанию, в которую устроиться на работу;
* Взрослые любого возраста, те, кто хочет попробовать в жизни что-то новое, сменить работу, профессию, сферу деятельности, ищут свое призвание, предназначение или просто запутались в себе
  1. Основные понятия предметной области

В проекте будет присутствовать несколько характерных для нашего приложения сущностей. Введем понятия, которые будут использоваться дальше по тексту:

Сервис – REST-сервис, который хранит, обрабатывает и предоставляет данные для мобильного приложения.

Пользователь – лицо, заинтересованное в выявлении своих профессиональных качеств.

Эксперт – лицо имеющее профессиональные навыки в оценке результатов опросов, для которых данная статистика имеет определенную ценность.

Тест – тест для оценки профессиональных интересов методики Соломина.

Вопрос – высказывание, которое респондент оценивает по шкале от 0 до 3, в зависимости от степени личного согласия или несогласия с высказыванием.

Результат тестов – проанализированная совокупность ответов респондента на вопросы теста, выраженная в форме результата, а именно профессии с описанием.

* 1. Функциональные требования к системе

Необходимо было сформулировать конкретные функциональные требования к системе.

Система должна предоставлять пользователю требуемый функционал, а именно: прохождение тестирования, получение результатов тестирования, управление личными данными, а также для статистического эксперта должна быть предусмотрены функции получения результатов тестирования пользователей по фильтрам.

По вышеописанному функционалу были сформированы бизнес требования и разработаны основные сценарии работы приложения.

Бизнес требования:

* Интерфейс приложения должен соответствовать концепциям Google Material Design;
* При входе в приложение пользователю должна быть доступна информация о приложении и логотип;
* Для авторизованного пользователя доступно: управление личными данными, результаты пройденных тестирований и соответственно прохождение тестирования;
* Для авторизованного статистического эксперта должна быть доступна функция получения данных прохождения тестирований пользователями.

Сценарий работы приложения для пользователя:

* При запуске приложения пользователь видит главный экран, на котором находятся кнопки «Вход» и «Регистрация», а также описание приложения.
* Если пользователь зарегистрирован, он может войти в систему с помощью кнопки «Вход», если нет, то пользователю необходимо нажать кнопку «Регистрация».
* При нажатии кнопки «Регистрация» пользователь перенаправляется на экран регистрации, после чего заполняет необходимые поля и нажимает кнопку «Зарегистрироваться» и система регистрирует пользователя.
* После входа пользователь попадает на главный экран, на котором у него есть возможность начать прохождение теста с помощью кнопки «Пройти тест»
* На верхней части экрана после входа в систему появляется кнопка перехода на профиль пользователя, в которой представлена личная информация пользователя, а также результаты предыдущих тестов.
* При нажатии кнопки «Пройти тест» пользователь перенаправляется на экран прохождения теста
* При прохождении теста пользователь перенаправляется на экран с результатом.

Сценарий работы приложения для статистического эксперта:

* При запуске приложения пользователь видит главный экран, на котором находятся кнопки «Вход» и «Регистрация», а также описание приложения.
* После входа статистический эксперт попадает на главный экран, на котором у него есть возможность перейти на экран получения статистики.
* При нажатии кнопки «Получить статистику» эксперт перенаправляться на экран заполнения фильтров для получения статистики.
* Заполнив все необходимые поля поиска и нажав кнопу «Найти» должен выдаться список о всех пройденных тестах и графики по статистике.
  1. Нефункциональные требования к системе

В нашей системе присутствует две связанные составляющие это веб-сервис и мобильный клиент.

* + 1. Нефункциональные требования к сервисной части

1. Технологии и инструменты, среда разработки

Программа должна быть реализована на языке Java, в среде разработки IntelliJ IDEA. Используемым фреймворком является Spring Boot 2.1.1. В качестве хранилища данных должна использоваться выбранная реляционная СУБД PostgreSQL. Для автоматизации сборки проекта используется Apache Maven 3.6.0.

1. Требования к производительности

Количество пользователей, которые могут одновременно пользоваться приложением без потери производительности – не менее 100, а отклик системы должен составлять не более 3 секунд.

1. Требования к сохранности данных

Сохранность данных должна обеспечивается целостностью данных на уровне базы данных.

1. Критерии качества программного обеспечения

* Должны полностью выполняться требования к производительности.
* Описанная функциональность разработанной программной системы должна быть реализована на 100%.
  + 1. Нефункциональные требования к мобильной части

1. Технологии и инструменты, среда разработки

Программа должна быть реализована с использованием фреймворка Flutter на языке Dart, в среде разработки Android Studio или IntelliJ IDEA.

1. Требования к производительности

Отклик системы должен составлять не более 3 секунд.

1. Требования к сохранности данных

Для сохранности данных на стороне мобильного приложения необходимо стабильное подключение к сети Интернет.

1. Критерии качества программного обеспечения

* Должны полностью выполняться требования к производительности.
* Описанная функциональность разработанной программной системы должна быть реализована на 100%.
* Разработанный интерфейс должен соответствовать представленному макету.
* Для работы приложения необходима операционная система Android 6.0 и выше.
  1. Сравнительный анализ аналогов

Все аналоги были взяты из Google Play – открытого мобильного приложения, позволяющего владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения.

***Наименование аналога:*** Тест на Профессию

***Краткое описание, представленное в Google Play:*** С помощью нашего теста Вы сможете узнать какая профессия Вам больше всего подходит и какую зарплату Вы заслуживаете! Проверьте себя и своих близких и может наш тест подскажет Вам то чего Вы не знали.

***С аналогом можно ознакомиться по ссылке:*** <https://play.google.com/store/apps/details?id=com.alexappcommpany.test_salary_new>

***Оценка приложения пользователями:*** представлена на рисунке 1.

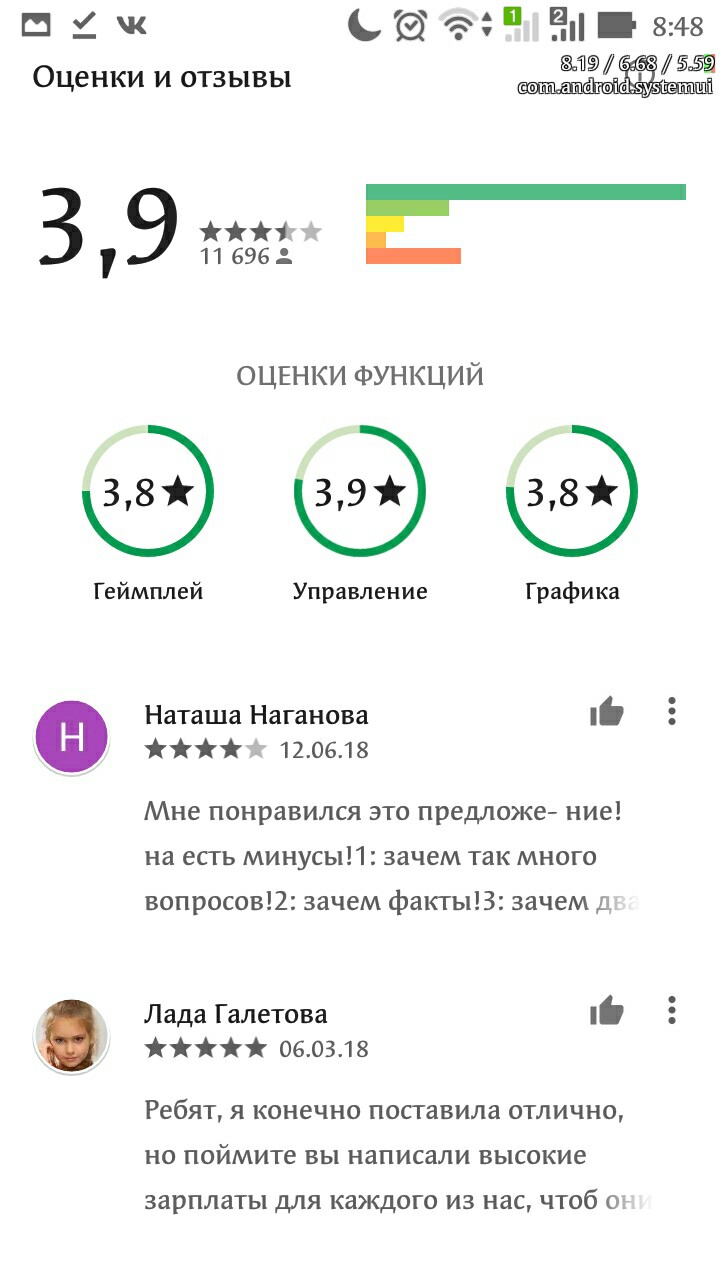


Рисунок 1. Оценка пользователей аналога «Тест на Профессию».

***Наименование аналога:*** Тесты на Профессию

***Часть описания, представленного в Google Play:*** Тесты на Профессию, потому что только в нем содержится целых 5 уникальных и независимых тестов! Пройдя все наши тесты, ты точно найдешь ответ на вопрос какую профессию выбрать.

***С аналогом можно ознакомиться по ссылке:*** <https://play.google.com/store/apps/details?id=com.infokombinat.testprofession>

Оценка приложения пользователями: представлена на рисунке 2.

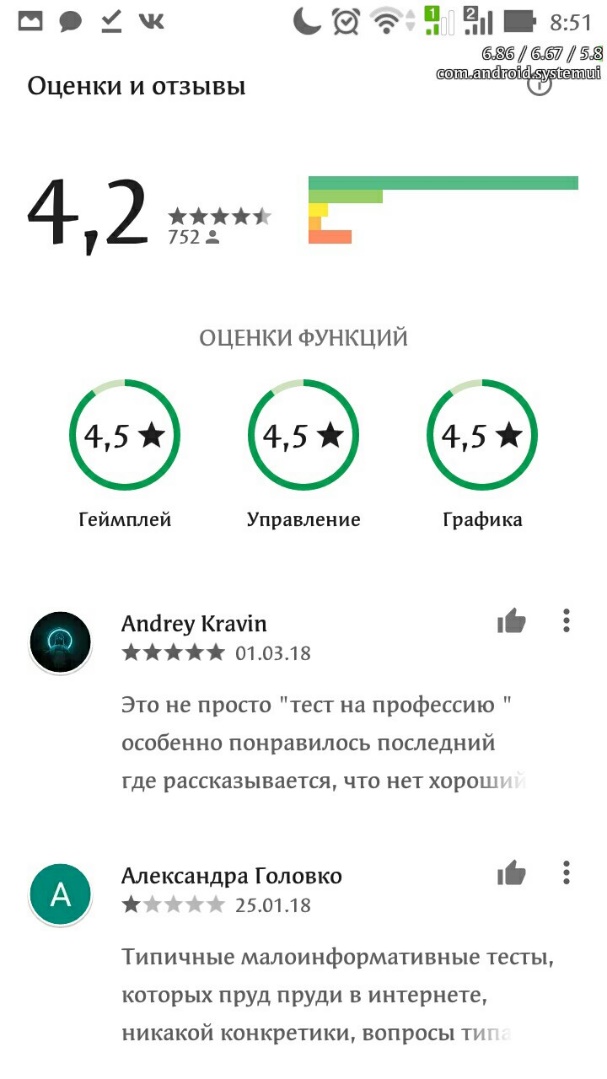


Рисунок 2. Оценка пользователей аналога «Тесты на Профессию».

***Наименование аналога:***Тест на будущую профессию

***Лаконичное описание в Google Play:*** Узнай будущую профессию и зарплату. В приложении присутствует реклама.

***С аналогом можно ознакомиться по ссылке:*** <https://play.google.com/store/apps/details?id=com.jifisher.profession>

***Оценка приложения пользователями:*** представлена на рисунок 3.

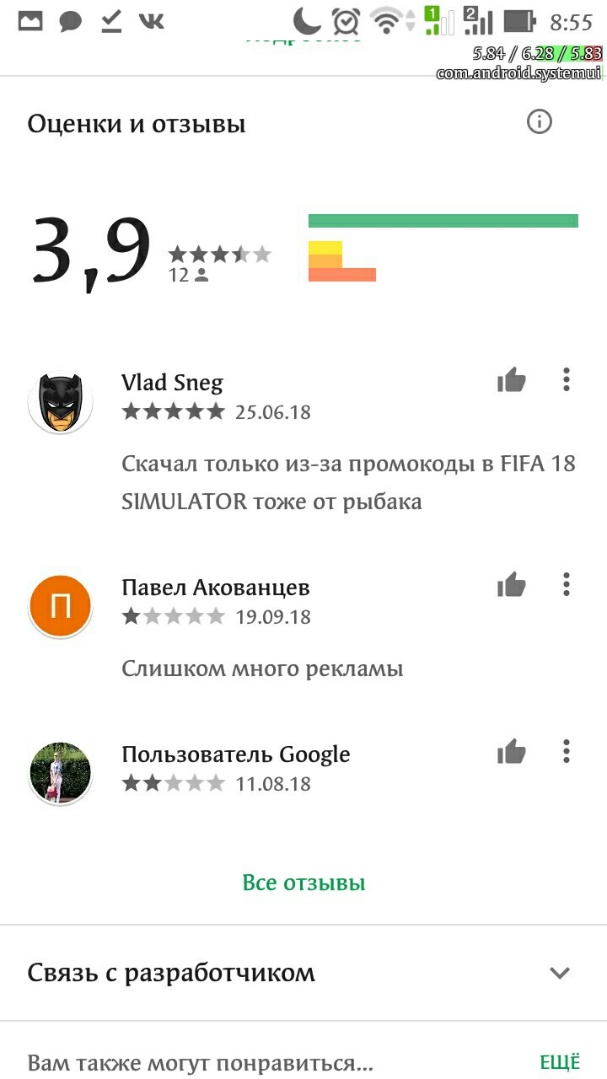


Рисунок 3. Оценка пользователей аналога «Тест на будущую профессию».

***Выводы:*** У каждого из аналогов есть минусы будь то навязчивый контент или субъективность результата. Необходимо избавиться от недостатков, тем самым выпустив более качественный продукт.

1. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

Проектирование системы включает в себя спецификацию и описание требований, определение компонентов системы.

* 1. Спецификация требований

Функциональные требования, определенные в пункте 1.3 представим на диаграмме прецедентов (рисунок 4). Также для каждого прецедента была написана полная спецификация.



Рисунок 4. Диаграмма прецедентов.

* + 1. Спецификация варианта использования «Регистрация»

***Название:*** Регистрация

***Краткое описание:*** Действующее лицо вводит данные для регистрации, и, в случае успеха, создается регистрационная запись, и действующее лицо получает возможность зайти в систему

***Действующее лицо:*** Пользователь

***Предусловие:*** Нет

***Основной поток:***

1. Действующее лицо нажимает на кнопку «Регистрация» на главном экране системы;
2. Система перенаправляет действующее лицо на страницу ввода данных для регистрации;
3. Действующее лицо вводит свои данные для регистрации;
4. Действующее лицо нажимает кнопку «Зарегистрироваться»;
5. Система проверяет правильность введенных данных;
6. Система отсылает запрос на сервис;
7. Система получает и обрабатывает ответ от сервиса;
8. Система перенаправляет действующее лицо на главный экран.

***Постусловие:*** Создана регистрационная запись о действующем лице

***Альтернативный поток 1:***

1. Альтернативный поток начинается на шаге 5, если введенные данные не правильны;
2. Система сообщает действующему лицу в каких полях ошибка;
3. Поток возвращается на шаг 3.

***Альтернативный поток 2:***

1. Альтернативный поток начинается на шаге 7, если сервис не отвечает на запрос;
2. Система показывает действующему лицу всплывающее окно с ошибкой.
   * 1. Спецификация варианта использования «Вход»

***Название:*** Вход

***Краткое описание:*** Действующее лицо вводит данные для аутентификации, и, в случае успеха, система предоставляет ему доступ к закрытым функциям

***Действующее лицо:*** Пользователь, Эксперт

***Предусловие:*** Действующее лицо зарегистрировано в системе

***Основной поток:***

1. Действующее лицо нажимает на кнопку «Вход» на главном экране системы;
2. Система перенаправляет действующее лицо на страницу ввода данных для аутентификации;
3. Действующее лицо вводит свои данные для аутентификации;
4. Действующее лицо нажимает кнопку «Войти»;
5. Система проверяет правильность введенных данных;
6. Система отсылает запрос на сервис;
7. Система получает и обрабатывает ответ от сервиса;
8. Система перенаправляет действующее лицо на главный экран.

***Постусловие:*** Действующее лицо авторизовано в системе

***Альтернативный поток 1:***

1. Альтернативный поток начинается на шаге 5, если введенные данные не правильны;
2. Система сообщает действующему лицу в каких полях ошибка;
3. Поток возвращается на шаг 3.

***Альтернативный поток 2:***

1. Альтернативный поток начинается на шаге 7, если сервис не отвечает на запрос;
2. Система показывает действующему лицу всплывающее окно с ошибкой.
   * 1. Спецификация варианта использования «Просмотр личной информации»

***Название:*** Просмотр личной информации

***Краткое описание:*** Действующее лицо просматривает информацию своего профиля

***Действующее лицо:*** Пользователь

***Предусловие:***Действующее лицо авторизовано в системе

***Основной поток:***

1. Действующее лицо нажимает на иконку профиля на любом экране системы, кроме экрана с вопросами;
2. Система отсылает запрос на сервис о получении данных действующего лица;
3. Система получает и обрабатывает ответ от сервиса;
4. Система перенаправляет действующее лицо на экран профиля;

***Постусловие:*** Действующему лицу предоставлена информация о профиле

***Альтернативный поток:***

1. Альтернативный поток начинается на шаге 3, если сервис не отвечает на запрос;
2. Система показывает пользователю всплывающее окно с ошибкой.
   * 1. Спецификация варианта использования «Редактирование личной информации»

***Название:*** Редактирование личной информации

**Краткое описание:** Действующее лицо изменяет информацию своего профиля

***Действующее лицо:*** Пользователь

***Предусловие:*** Действующее лицо авторизовано в системе

***Основной поток:***

1. Действующее лицо нажимает на кнопку «Изменить» на экране профиля;
2. Система перенаправляет действующее лицо на экран изменения профиля;
3. Действующее лицо вводит измененные данные;
4. Действующее лицо нажимает кнопку «Сохранить»;
5. Система проверяет правильность введенных данных;
6. Система отсылает запрос на сервис об изменении данных действующего лица;
7. Система получает и обрабатывает ответ от сервиса;
8. Система перенаправляет действующее лицо на экран профиля;

***Постусловие:*** Изменена информация профиля действующего лица

***Альтернативный поток 1:***

1. Альтернативный поток начинается на шаге 5, если введенные данные не правильны;
2. Система сообщает действующему лицу в каких полях ошибка;
3. Поток возвращается на шаг 3.

***Альтернативный поток 2:***

1. Альтернативный поток начинается на шаге 7, если сервис не отвечает на запрос;
2. Система показывает действующему лицу всплывающее окно с ошибкой.
   * 1. Спецификация варианта использования «Прохождение теста»

***Название:*** Прохождение теста

***Краткое описание:*** Действующее лицо отвечает на вопросы теста

***Действующее лицо:*** Пользователь

***Предусловие:*** Действующее лицо авторизовано в системе

***Основной поток:***

1. Действующее лицо нажимает на кнопку «Пройти тест» на главном экране;
2. Система отсылает запрос на сервис о получении данных вопросов;
3. Система получает и обрабатывает ответ от сервиса;
4. Для всех вопросов теста:
   1. Система перенаправляет действующее лицо на экран со следующим вопросом;
   2. Действующее лицо выбирает вариант ответа;
   3. Действующее лицо нажимает кнопку «Ответить»;
   4. Система проверяет выбран ли вариант ответа;
5. Система обрабатывает ответы действующего лица;
6. Система отсылает запрос на сервис о сохранении результата теста;
7. Система получает и обрабатывает ответ от сервиса;
8. Система перенаправляет действующее лицо на экран с результатом теста;

***Постусловие:*** Действующему лицу предоставлена информация о профиле

***Альтернативный поток 1:***

1. Альтернативный поток начинается на шаге 4.4, если вариант ответа не выбран;
2. Система сообщает действующему лицу, чтобы он выбрал вариант ответа;
3. Поток возвращается на шаг 4.2.

***Альтернативный поток 2:***

1. Альтернативный поток начинается на шаге 3 и 7, если сервис не отвечает на запрос;
2. Система показывает действующему лицу всплывающее окно с ошибкой.
   * 1. Спецификация варианта использования «Получение статистики по фильтру»

***Название:*** Получение статистики по фильтру

***Краткое описание:*** Действующее лицо просматривает информацию по тестам, пройденным пользователями, соответствующим критериям примененного фильтра

***Действующее лицо:*** Эксперт

***Предусловие:*** Действующее лицо авторизовано в системе и находится на главном экране

***Основной поток:***

1. Действующее лицо нажимает кнопку «Получить статистику»;
2. Действующее лицо вводит критерий фильтрации и нажимает кнопку «Найти»;
3. Система отсылает запрос на сервис о получение результата;
4. Система перенаправляет на экран просмотра статистики и выводит отфильтрованные данные, в соответствии с выбранным фильтром и критерием фильтрации;

***Постусловие:*** Действующему лицу предоставлена информация по тестам, пройденным пользователями, соответствующим критериям примененного фильтра

***Альтернативный поток 1:***

1. Альтернативный поток начинается на шаге 2 если поля фильтров введены на корректно;
2. Система сообщает действующему лицу в каких полях ошибка;

***Альтернативный поток 2:***

1. Альтернативный поток начинается на шаге 3, если сервис не отвечает на запрос;
2. Система показывает действующему лицу всплывающее окно с ошибкой.
   * 1. Спецификация варианта использования «Просмотр результата теста»

***Название:*** Просмотр результата теста

***Краткое описание:*** Действующее лицо просматривает информацию о пройденном тесте

***Действующее лицо:*** Пользователь

***Предусловие*:** Действующее лицо авторизовано в системе и находится на экране личного профиля

***Основной поток:***

1. Система обрабатывает запрос действующего лица;
2. Система предоставляет пользователю данные о пройденном тесте.

**Постусловие:** Действующему лицу предоставлена информация о пройденном тесте

**Альтернативный поток:** нет

* 1. Диаграмма развертывания

Для общего понимания взаимодействия компонентов системы была разработана диаграмма развертывания (рисунок …)

Основными составляющими являются:

* Spring Boot сервис предоставляющий функциональное API;
* Android приложение;
* iOS приложение;
* Firebase, предоставляющий услуги push-уведомлений.
  1. Проектирование сервисной части

Проектирование в первую очередь помогает заранее определить важные моменты, принять решения и тем самым определить вектор разработки и облегчить её процесс.

В процессе проектирования была разработана объектная модель предметной области и с её использованием разработана спецификация API сервиса.

* + 1. Разработка диаграммы классов

Прежде всего была разработана диаграмма классов предметной области. Данная диаграмма позволяет зафиксировать основные объекты и с их использованием проектировать и в дальнейшем реализовывать бизнес-логику приложения [2]. Диаграмма классов в нотации UML 2.0 представлена на рис. 5.



Рисунок 5. Диаграмма классов.

* + 1. Написание спецификации (API)

Спецификация на программный интерфейс (API) приложения представляет собой контракт, который должен быть описан при проектировании сервиса и, в дальнейшем, на его основании должна быть разработана мобильная и сервисная часть.

API сервиса было спроектировано и записано в формате README. При проектировании учитывались следующие параметры:

• URL

• HTTP метод

• Роли пользователей, у которых есть доступ к этому методу

• Запрос (параметры или тело запроса)

• Ответ (тело ответа)

Текст спецификации программного интерфейса сервиса представлена в приложении A.

* 1. Проектирование клиентского приложения

Роль проектирования при разработке мобильного клиентского приложения позволяет заранее определить такие моменты как макеты экранов приложения. Проектирование макетов позволяет на ранних этапах представить как будет выглядеть приложение и принять решение начинать разработку или переработать макеты.

В системе должны быть представлены следующие экраны, соответствующие макетам:

Главная – экран на котором находится описание теста. На рисунке 6 пользователь еще не авторизован в системе, на рисунке 7 пользователь авторизован.



Рисунок 6. Макет главного экрана без входа.



Рисунок 7. Главный экран с входом.

Регистрация – экран для ввода регистрационной информации. Макет на рисунке 8.



Рисунок 8. Экран регистрации.

Вход – экран для ввода аутентификационной информации. Макет на рисуноке 9.

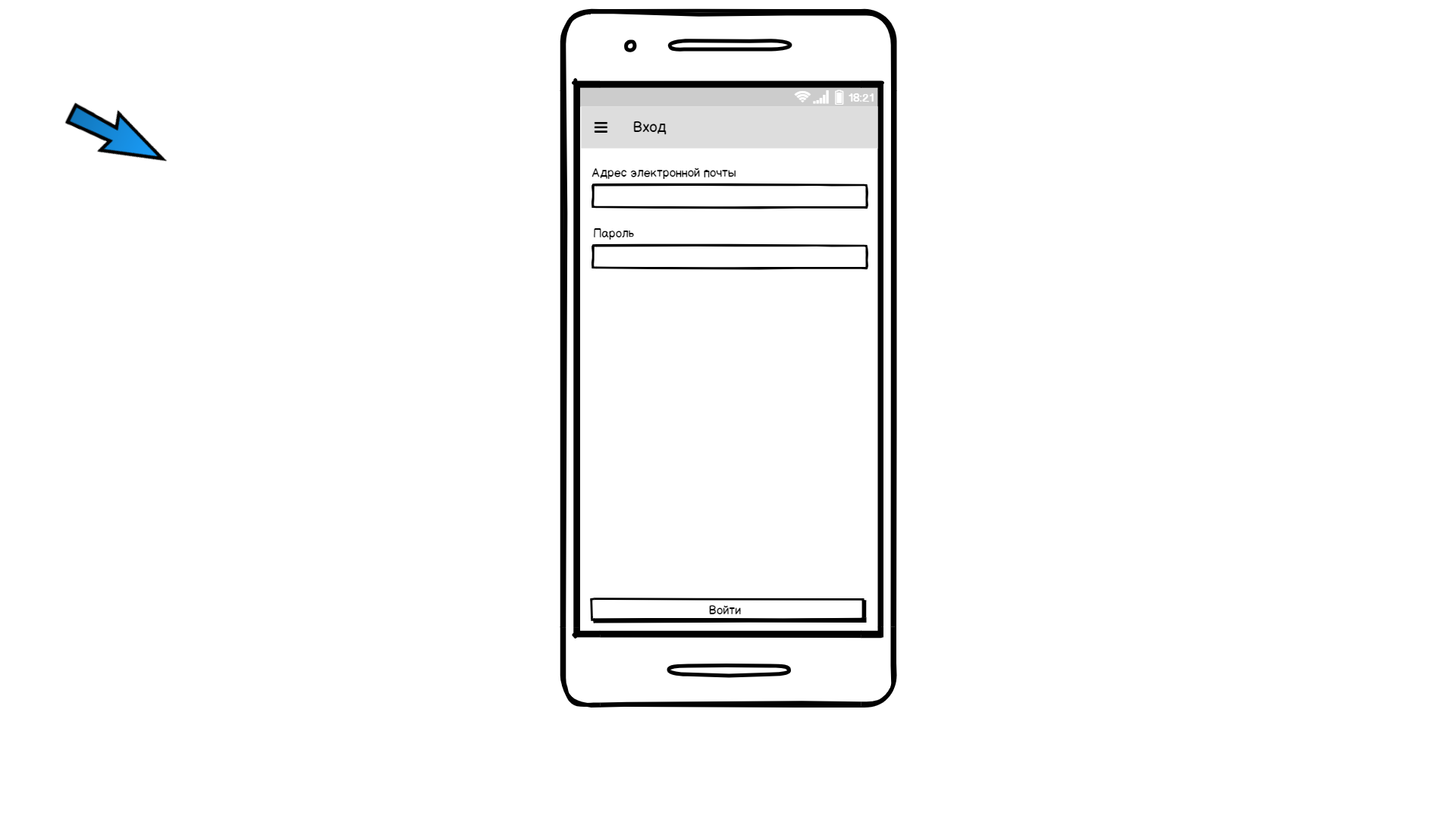


Рисунок 9. Экран входа.

Профиль – экран с личными данными пользователя и с результатами пройдённых тестов. Макет на рисунке 10.

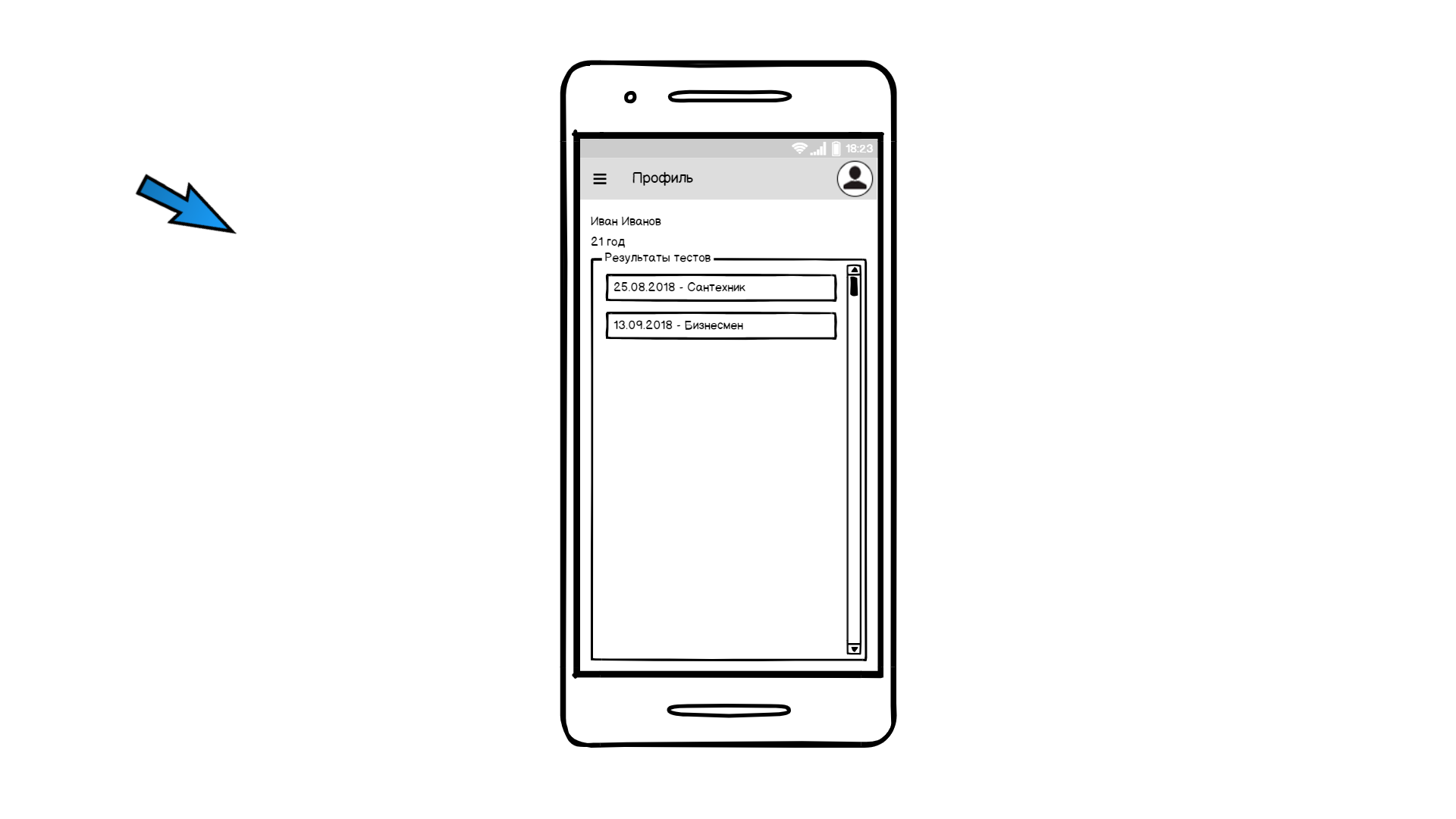


Рисунок 10. Экран профиля.

Изменение профиля – экран для ввода измененной информации пользователя. Макет на рисунке 11.



Рисунок 11. Экран изменения профиля.

Вопрос – экран для отображения вопроса и выбора варианта ответа. Макет на рисунке 12.



Рисунок 12. Экран вопроса.

Результат теста – экран для отображения результата пройдённого теста. Макет на рисунке 13.



Рисунок 13. Экран результата теста.

Статистика – экран для отображения статистики для эксперта. Макет на рисунке 14.



Рисунок 14. Экран статистики.

Для отображения ошибок и сообщений пользователям необходимо использовать всплывающее окно с сообщением об ошибке. Макет на рисунке 15.

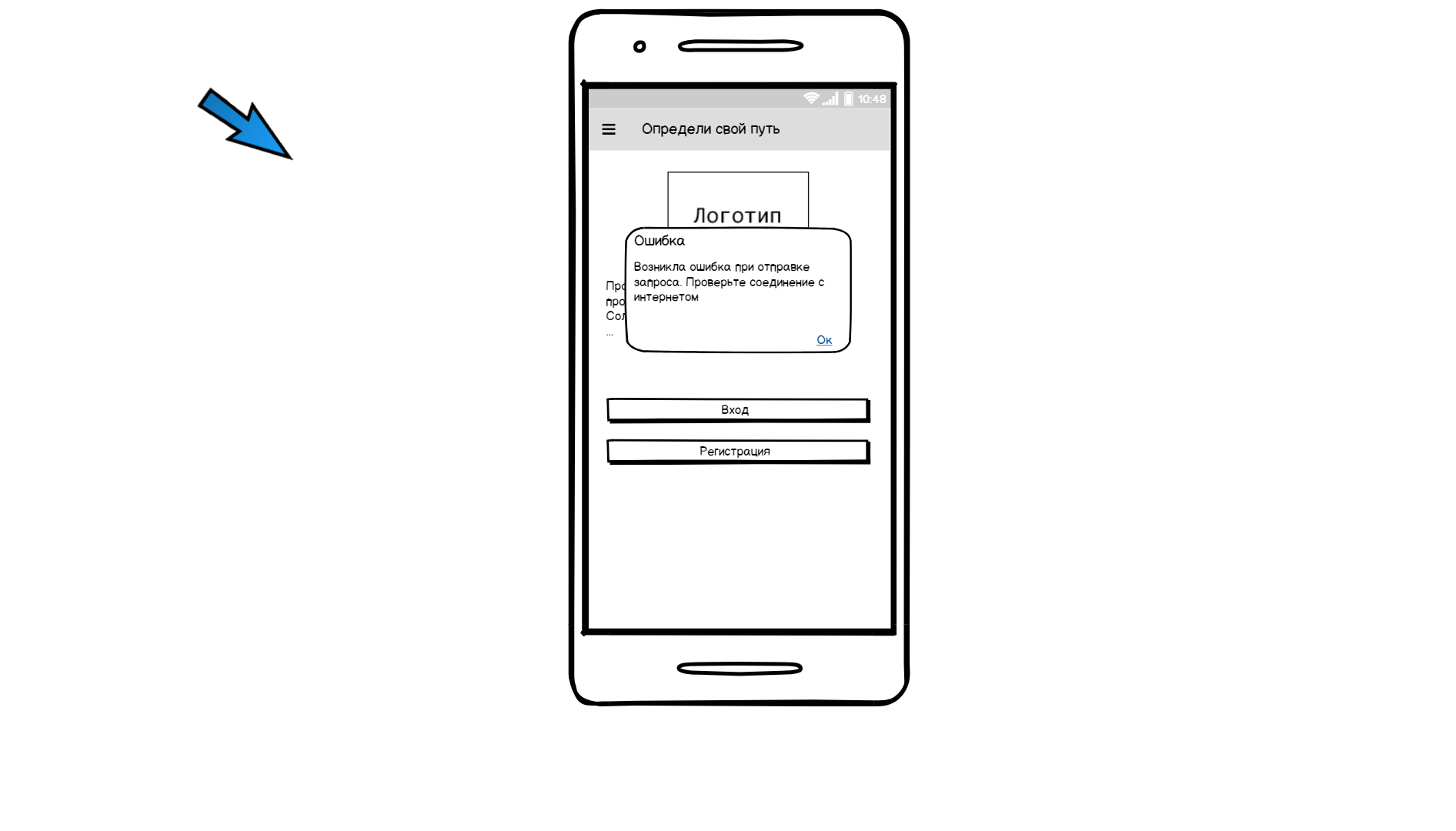


Рисунок 15. Сообщение об ошибке.

1. РЕАЛИЗАЦИЯ СИСТЕМЫ
   1. Реализация сервиса

При разработке сервиса был применен подход к построению архитектуры, при которой приложение разделено на слои, отделенные друг от друга по функциональному назначению:

* контроллеры – принимают запросы и формируют ответы;
* сервисы – выполняют определенную бизнес-логику;
* репозитории – осуществляют доступ к данным;
  + 1. Слой контроллеров

Слой, отвечающий за управление запросами и ответами он был написан в соответствии спецификации представленной в пункте …

* + 1. Слой бизнес-логики

Слой бизнес-логики представлен объектами фреймворка Spring. Сервисы являются связующим звеном между двумя другими слоями – В сервисы контроллеры отправляют данные, с которыми они производят некие действия, совершают запросы к базе данных и возвращают результат.

* + 1. Слой доступа к данным

Для связи приложения с базой данных была использована библиотека JPA, которая представляет удобный интерфейс взаимодействия с источником данных и абстрагирует разработчика от специфического языка запросов и DDL. Связь с PostgreSQL в приложении представлена Spring-репозиторием – JPARepository, который предоставляет стандартные CRUD-операции.

Для обеспечения минимальной валидации входных данных поля доменных объектов были помечены аннотациями @NotNull стандарта JSR-303, что означает, что поле объекта должно быть заполнено, в ином случае будет выброшено исключение.

Исходные тексты программного кода разработанного прототипа представлены в приложении Б.

* + 1. Аутентификация

Требования к приложению обязывают реализовать аутентификацию и авторизацию пользователей. После анализа существующих способов аутентификации пользователей была выбрана технологии JWT-токенов.

JSON Web Token (JWT) — это открытый стандарт (RFC 7519) для создания токенов доступа, основанный на формате JSON. Как правило, используется для передачи данных авторизации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения своей личности [3].

При разработке использовалась компонент экосистемы Spring – Spring Security. Он позволяет легко реализовать в приложении аутентификацию и авторизацию пользователей, а также разграничение доступа пользователей к различным функциям системы.

Работа с JWT обеспечивалась библиотекой io.jsonwebtoken:jjwt. Она позволяет без труда формировать токен, шифровать его при помощи различных протоколов (в приложении используется HMAC), а также подписывать секретным ключом.

* + 1. Проверка работоспособности

После того как сервисная часть была реализована была произведена проверка её работоспособности с помощью приложения для тестирования REST API – Postman.

Была проверена правильность функционирования сервиса, корректная работа механизмов авторизации, а также было найдено некоторое количество багов, которые были успешно устранены до выхода в продакшен.

* 1. Реализация мобильного клиента

В требованиях к реализации мобильного приложения зафиксирован язык программирования Dart и фреймворк Flutter. Flutter позволяет писать мобильные приложения, используя множество разработанных элементов интерфейса, в том числе компоненты Material Design. Приложения, написанные на Flutter, компилируются в нативный код платформ Android и iOS, что позволяет увеличить скорость выполнения приложений.

* + 1. Взаимодействие с сервисной частью

Взаимодействие с сервисной частью осуществляется с помощью асинхронных вызовов. Асинхронные вызовы позволяют отправить запрос на сервис и не дожидаясь от него ответа выполнять собственную логику. Как только ответ от сервиса приходит, вызываются callback-функции, заданные для разных исходов асинхронного вызова.

Ниже представлен пример вызова сервиса в виде GET-запроса:

static Future<http.Response> get(String path, { Map<String, String> params, bool auth = false }) async {

Uri uri = Uri.http(RestPaths.BASE\_URL, path, params);

if(auth) {

return await http.get(uri, headers: { Http.HEADER\_AUTHORIZATION : await \_getToken() });

} else {

return await http.get(uri);

}

}

Код, запроса данных от сервиса с объявлением функции успешного её завершения выглядит следующим образом:

AuthService.auth(\_login, \_password).then((res) {

if (res) {

Navigator.of(context).pop();

} else {

\_showError("Неправильный логин или пароль");

}

}).catchError((e) => \_showError("Ошибка"));

* + 1. Хранение данных

Хранение данных на стороне клиента выполняется исключительно ради оптимизации работы приложения. На клиенте хранятся данные опроса и информация для доступа к сервису в зашифрованном и подписанном виде и не хранится никаких конфиденциальных данных, что не дает злоумышленникам получить их из внутреннего хранилища мобильного телефона.

Для хранения данных на мобильных клиентах использовалась библиотека shared\_preferences, которая является абстракцией для внутренних хранилищ пар ключ-значение NSUserDefaults на iOS и SharedPreferences на Android.

* + 1. Проверка работоспособности

Работоспособность мобильного приложения проверялась путем запуска приложения на различных устройствах, с разными версиями операционных систем и разным разрешением экрана, а именно:

* Смартфон Google Pixel, Android API 23, разрешение экрана – 1920х1080 пикселей;
* Планшет Google Pixel C, Android API 25, разрешение экрана – 2560x1800 пикселей.

Как приложение выглядит на разных устройствах представлено на рисунках 16 и 17.

Рисунок 16. Главный экран приложения на Google Pixel.

Рисунок 17. Главный экран приложения на Google Pixel C.

1. ИНСТРУКЦИЯ ПО РАЗВЕРТЫВАНИЮ

В данном разделе представим пошаговую инструкцию по установке нашего приложения, начиная с этапа скачивания исходного кода и заканчивая получением полнофункционального приложения. Данная инструкция в большей степени написана для разработчиков.

* 1. Установка сервисной части

Шаг 1: Убедитесь, что у вас установлены обязательные компоненты такие как:

* IntelliJ IDEA 2017.3.4 x64 и позднее (например учебная версия) ссылка для скачивания: <https://www.jetbrains.com/idea/>;
* СУБД PostgreSQL (доступен для скачивания: <https://www.postgresql.org/download/> );
* Git (доступен по скачиванию на официальном сайте: <https://git-scm.com/downloads>).

Шаг 2: Скачайте репозиторий с помощью команды: git clone <https://github.com/m-obrubov/mobile-kp.git> (Репозиторий закрытый поэтому просьба написать на почту: [max.obrubov@gmail.com](mailto:max.obrubov@gmail.com) для получения доступа).

Шаг 3: Выполните импорт проекта questionnaire-be в среду разработки IntelliJ IDEA.

Шаг 4: Создайте базу данных и пользователя, наделенного правами на чтения и запись данной БД.

Скрипт создания:

CREATE DATABASE questionnaire;

CREATE USER questionnaire\_be\_client WITH PASSWORD 'password';

GRANT ALL ON DATABASE questionnaire TO questionnaire\_be\_client;

Шаг 5: Создайте настройки подключения, прописанные в application.yml находящегося в каталоге: \questionnaire-be\src\main\resources\.

В этом же файле в настройках укажите значение параметра:

jpa:

hibernate:

ddl-auto: create

При последующих запусках замените значение параметра на validate.

Шаг 6: При необходимости можете поправить json заполнения теста, не изменяя его структуры (например, поменять описание теста) расположенном в каталоге: \questionnaire-be\src\main\resources\json\.

Шаг 7: Выполните запуск проекта.

* 1. Установка сервисной части

Шаг 1: Убедитесь, что у вас установлены обязательные компоненты такие как:

* IntelliJ IDEA (ссылка для скачивания: <https://www.jetbrains.com/idea/>) или Android Studio (ссылка для скачивания: <https://developer.android.com/studio/>);
* Git (доступен по скачиванию на официальном сайте: <https://git-scm.com/downloads>).

Шаг 2: Выполните установку и настройку flutter (первые два шага на официальном сайте: <https://flutter.io/docs/get-started/install>).

Шаг 3: Выполните импорт проекта questionnaire\_fe в среду разработки IntelliJ IDEA или Android Studio.

Шаг 4: Выполните запуск проекта.

1. ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ ПРОЕКТОМ

6.1 Общие принципы взаимодействия в команде

В процессе разработки мы придерживались следующим принципам:

* Разграничение обязанностей;
* Разработку необходимо вести в системе контроля версий;
* Сначала думать потом делать (придерживаться заранее разработанных спецификаций);
* Необходимо вести разработку не зависимых частей на разных ветках
* Придерживаться прицепам именования веток (ветка должна кратко информировать о ее содержании);
* Если приходиться вести разработку на одной ветки, необходимо предупреждать о push-ах, и не в коем случает не делать не работающие push-ы;
* Кто поломал тот и чинит;
* Если есть общие задачи продуктивно придерживаться принципам парного программирования.

6.2 Распределение ролей в команде/зоны ответственности

По ходу работы выявлялись задачи и назначались исполнители. С распределением обязанностей по задачам можно ознакомиться по таблице 1.

Таблица. Распределение задач.

|  |  |  |
| --- | --- | --- |
| Задачи | Исполнители | Примерные сроки исполнения |
| Формирование требований | | |
| Формирование функциональных требований | Максим  Василиса  Заказчик (Преподаватель) | 2 дня |
| Формирование не функциональных требований | Максим  Василиса | 1 неделя |
| Написание ТЗ | | |
| Название проекта | Максим  Василиса | 2 недели |
| Постановка целей | Максим  Василиса |
| Описание предметной области | Максим  Василиса |
| Написание словаря предметной области | Василиса |
| Выявление предполагаемой аудитории | Василиса |
| Рассмотрение аналогов | Василиса |
| Написание сценария работы приложения | Максим  Василиса |
| Создание диаграммы прецедентов | Василиса  Максим |
| Спецификация вариантов использования | Максим |
| Определение требований к внешнему интерфейсу | Максим |
| Создание макетов страниц | Максим |
| Проектирование системы | | |
| Проектирование диаграммы классов | Максим  Василиса | 3 дня |
| Создание диаграммы классов средствами UML | Максим  Василиса | 1 день |
| Создание диаграммы развертывания | Максим  Василиса | 1 день |
| Написание API для сервиса | Максим  Василиса | 3 дня |
| Разработка сервисной части | | |
| Модель | Максим | 1 день |
| Слой доступа к данным | Максим | 2 дня |
| Контроллеры | Максим  Василиса | 1 неделю |
| Ответы от сервера | Василиса | 2 дня |
| Аутентификация | Василиса | 2 день |
| Конвертер для даты | Максим | 1 день |
| Логика на уровне сервисов | Максим  Василиса | 1 неделя |
| Обработка ошибок | Максим | 1 день |
| Разработка мобильной части | | |
| Главный экран | Максим | 1 день |
| Экран регистрации | Максим | 1 день |
| Экран авторизации | Максим | 1 день |
| Экран прохождения теста | Василиса | 2 дня |
| Экран статистики | Максим  Василиса | 2 дня |
| Экран результата | Василиса | 1 день |
| Интеграция получения теста | Максим  Василиса | 1 неделя |
| Интеграция получения и изменения личных данных | Максим | 1 неделя |
| Интеграция регистрации | Максим | 1 неделя |
| Интеграция прохождения тестирования и получения результата | Василиса | 2 недели |

ЗАКЛЮЧЕНИЕ

В данном проекте было создано веб-приложение мотивирующего характера с элементами игры. Были изучены новые технологии (такие как EJB, API), разработан собственный веб-сервис, затронута разработка Back-End и Front-End, изучены методы проектирования UX интерфейса. Но, в процессе выполнения данной курсовой работы были изучены не только технические способы разработки системы. Эта работа позволила получить существенный опыт в различных сферах, которые составляют разработку приложения: аналитика, разработка архитектуры, дизайн, который включает изучение цветовых сочетаний, создание макетов, тестирование, работа в команде. И результатом нашей совместной работы является готовое приложение, основная задача которого – помощь людям в достижении целей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Анкета Ориентация, И.Л. Соломин [электронный ресурс] <https://psycabi.net/testy/372-anketa-orientatsiya-i-l-solomin-oprosnik-proforientatsii-metodika-professionalnykh-interesov-i-sposobnostej-test-solomina-i-l> (дата обращения: 10.11.2018).
2. Буч Г., Рамбо Д., Якобсон И. – Язык UML. Руководство пользователя. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2006. – 496 с.: ил.
3. JSON Web Token [электронный ресурс] <https://ru.wikipedia.org/wiki/JSON_Web_Token> (дата обращения: 10.12.2018).
4. Spring boot, документация [электронный ресурс]. 2018. URL: <https://spring.io/guides/gs/spring-boot/> (дата обращения: 01.12.2018).

ПРИЛОЖЕНИЕ А  
NAME ….

ПРИЛОЖЕНИЕ B  
NAME …