

**МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Кафедра информационных технологий управления

Web-приложение «Система тестирования по Java»

09.03.02 Информационные системы и технологии

Курсовой проект по дисциплине «Технологии программирования»

Допущен к защите

Преподаватель _____ В.С. Тарасов

Обучающийся _____ Н.Р. Данкин, 3 курс, д/о

Обучающийся _____ Ю.Д. Кривоносова, 3 курс, д/о

Воронеж 2020

Содержание

Введение	4
Глоссарий	5
Постановка задачи	6
1. Анализ предметной области	7
1.2. Границы функциональности	8
1.3. Сравнение с аналогами	10
1.4. Анализ задач	12
1.5. Графическое описание работы системы	18
1.5.1. Диаграмма прецедентов	18
1.5.2. Диаграмма классов	20
1.5.3. Диаграмма последовательностей	21
1.5.4. Диаграмма коммуникаций	22
1.5.5. Диаграмма состояний	23
1.5.6. Диаграмма активности	24
1.5.7. Диаграмма объектов	25
1.5.8. Диаграмма взаимодействия	26
1.5.9. Диаграмма развертывания	27
2. Продуктовые воронки	28
3. Реализация приложения	33
3.1 Анализ средств реализации	33
3.2 Проектирование системы тестирования	33
3.2.1 Диаграмма классов	34
3.2.2 Диаграмма объектов	37
3.2.3 Диаграмма развертывания	37
3.3 Разработка приложения	38
3.3.1 Frontend разработка	38
3.3.2 Backend разработка	41
3.4 Тестирование	42
3.4.1 Дымовое тестирование	42
3.4.2 Юзабилити тесты	43

Заключение.....	45
-----------------	----

Введение

В современном мире мы наблюдаем участие информационных технологий практически во всех сферах человеческой деятельности, и сфера подбора персонала не является исключением. Ежедневно в фирмах, связанных с IT, рекрутеры и технические специалисты затрачивают несколько часов на проведение собеседований с потенциальными кандидатами. Иногда претендент может представить резюме, которое не имеет ничего общего с настоящим положением дел и характеризует превосходного специалиста. В итоге, при проведении собеседования с данным кандидатом, сотрудники только потеряют время, а ведь этого можно было избежать.

Для решения данной проблемы предлагается создать систему тестирования по Java. Система позволит кандидатам проходить первый этап собеседования - тестирование удаленно, что не будет отнимать время у работников организации. Претенденты, набравшие больше 80% правильных ответов. Получат приглашение на второй этап, который будет проходить уже с сотрудниками.

При наличии такого приложения у технического специалиста исчезнет необходимость в проведении первого этапа собеседования очно, что позволит освободить время для работы над проектами компании.

Глоссарий

Пользователь - человек, который прошел авторизацию, регистрацию в системе.

Посетитель - человек, который не вошел в систему, т.е не прошел регистрацию или авторизацию.

Система – веб-приложение «Система тестирования по Java».

Постановка задачи

Необходимо разработать веб-приложение, которое позволит пользователям проходить тестирование - первый этап отбора на такие вакансии как: Junior Developer, Middle Developer, Senior Developer. Результат теста определяет проходит пользователь на второй этап отбора или нет.

Приложение должно удовлетворять таким требованиям как:

1. Наличие русскоязычного интерфейса в Системе.
2. Стабильная работа в браузере Google Chrome версии 81.0.4044.122.
3. Выполнение основных функциональных задач Системы.
 - 3.1 Возможность регистрации / авторизации в Системе.
 - 3.2 Возможность прохождения тестирования, получения результата, анализа результата, а также выявления администратором кандидатов, получивших результат 80% и более правильных ответов, и приглашение их на второй этап собеседования.

Архитектура разрабатываемого приложения должна иметь front-end и back-end части. Выбор данной архитектуры обусловлен требованием отделить логику приложения и интерфейс. Должны быть реализованы следующие компоненты Системы:

1. База данных.
2. Клиентская часть приложения.
3. Серверная часть приложения.

1. Анализ предметной области

1.1. Пользователи системы

Система предназначена для работы:

- Администратора
- Пользователя
- Посетителя

В зависимости от роли пользователя будет доступна различная функциональность приложения.

В результате анализа можно выявить несколько типов пользователей, которые могут работать с системой.

Администратор

Данный тип пользователя обладает такими параметрами как:

- ФИО;
- Пароль;
- Информация о том, что этот пользователь является администратором.

Использование системы производится с целью поиска новых кадров на вакансии, посредством приглашения на второй этап собеседования кандидатов, успешно прошедших тестирование.

Пользователь

Данный тип пользователя обладает такими параметрами как:

- ФИО;
- Пароль;

Использование системы производится с целью получения приглашения на второй этап собеседования по интересующей пользователя вакансии в случае успешного прохождения тестирования.

Посетитель

Данный тип пользователя не обладает параметрами, информация о нем не хранится в системе.

1.2. Границы функциональности

Данная система реализовывает строго определенные возможности пользователей.

Работа с администратором

- Создание тестирования
- Редактирование тестирования
- Просмотр тестирования
- Отправление сообщения пользователю
- Просмотра личного кабинета пользователя
- Изменение пароля

- Изменение ФИО
- Изменение пароля пользователя
- Изменение ФИО пользователя
- Выход из системы

Работа с пользователем

- Выбор тестирования
- Прохождение тестирования
- Просмотр результата тестирования
- Просмотр анализа тестирования
- Просмотр сообщения администратора
- Изменение пароля
- Изменение ФИО
- Выход из системы

Работа с посетителем

- Регистрация / Авторизация в системе

1.3. Сравнение с аналогами

Среди аналогов Web-приложений, направленных на проведение тестирований по языку программирования Java, можно выделить следующие приложения:

1. Quizful (<http://www.quizful.net/test>)

Сайт разработан для пользователей, которые хотят проверить свои знания в интересующей их области программирования, позволяет проходить тестирования по таким языкам как Java, C++, C#, PHP и многим другим.

Минусами этого приложения можно считать:

- Отсутствие поддержки сайта с 2016 года
- Возможность регистрации только через электронную почту

Плюсами этого приложения можно считать:

- + Возможность просмотра правильных ответов с объяснениями после прохождения теста
- + Доступ к вопросам из реальных сертификационных экзаменов
- + Возможность получения сертификата
- + Наличие личного кабинета
- + Возможность прохождения теста на вакансию Junior Java Developer

2. Proghub (<https://proghub.ru/tests>)

Сайт позволяет проверить знания пользователя с помощью тестов и задач, а также подготовиться к интервью.

Минусами этого приложения можно считать:

- Отсутствие тестов на соответствие квалификации. Junior Java Developer, Middle Java Developer или Senior Java Developer, к примеру.
- Не работают ссылки на [linkedin.com](https://www.linkedin.com) и Telegram, представленные на сайте.

Плюсами этого приложения можно считать:

- + Возможность регистрации через социальные сети
- + Возможность анализа результата тестирования с объяснениями после прохождения теста
- + Постоянно пополняющаяся база тестов и вопросов
- + Возможность просмотра актуальных материалов о программировании
- + Наличие личного кабинета
- + Наличие рейтинга среди пользователей ресурса
- + Наличие задач для программистов
- + Наличие группы в VK, Facebook, профиля в Instagram.

Таким образом, выделим основные пункты, на которые надо равняться при реализации приложения:

- + Наличие личного кабинета
- + Возможность анализа результата тестирования
- + Возможность просмотра результатов в личном кабинете

1.4. Анализ задач

Задачи, связанные с работой администратора

Задача создания тестирования

Рассмотрим задачу создания тестирования. Данная задача включает в себя следующие этапы:

- Нажать на кнопку «Добавить тест» во вкладке «Тесты» панели администратора.
- Выбор вакансии, для которой будет создан тест.
- Написать вопросы, а также ответы к ним, среди которых будет один правильный вариант.
- Нажать на кнопку «Сохранить».

Задача редактирования тестирования

Рассмотрим задачу редактирования тестирования. Данная задача включает в себя следующие этапы:

- Нажать на кнопку «Редактировать» напротив тестирования, нуждающегося в изменении во вкладке «Тесты» панели администратора.
- Изменить вопросы, а также ответы к ним.
- Нажать на кнопку «Сохранить».

Задача удаления тестирования

Рассмотрим задачу удаления тестирования. Данная задача включает в себя следующие этапы:

- Нажать на кнопку «Удалить» напротив тестирования, нуждающегося в удалении во вкладке «Тесты» панели администратора.

Задача просмотра тестирования

Рассмотрим задачу просмотра тестирования. Данная задача включает в себя следующие этапы:

- Нажать на название тестирования, во вкладке «Тесты» панели администратора.

Задача отправки сообщения пользователю

Рассмотрим задачу отправки сообщения пользователю. Данная задача включает в себя следующие этапы:

- Нажать на кнопку «Написать сообщение» напротив ФИО пользователя во вкладке «Пользователи» панели администратора.
- Написать текст сообщения.
- Нажать на кнопку «Отправить».

Задача просмотра личного кабинета пользователя

Рассмотрим задачу просмотра личного кабинета пользователя. Данная задача включает в себя следующие этапы:

- Нажать на ФИО интересующего вас пользователя во вкладке «Пользователи» панели администратора.

Задача изменения пароля пользователя

Рассмотрим задачу изменения пароля пользователя. Данная задача включает в себя следующие этапы:

- Нажать на ФИО интересующего вас пользователя в панели администратора.
- Нажать на кнопку «Редактировать профиль»
- Изменить пароль
- Нажать на кнопку «Сохранить»

Задача изменения ФИО пользователя

Рассмотрим задачу изменения ФИО пользователя. Данная задача включает в себя следующие этапы:

- Нажать на ФИО интересующего вас пользователя в панели администратора.
- Нажать на кнопку «Редактировать профиль»
- Изменить ФИО
- Нажать на кнопку «Сохранить»

Задача изменения ФИО

Рассмотрим задачу изменения ФИО. Данная задача включает в себя следующие этапы:

- Нажать на круг справа в шапке сайта.
- Нажать на кнопку «Редактировать профиль»
- Изменить ФИО
- Нажать на кнопку «Сохранить»

Задача изменения пароля

Рассмотрим задачу изменения пароля. Данная задача включает в себя следующие этапы:

- Нажать на круг справа в шапке сайта.
- Нажать на кнопку «Редактировать профиль»
- Изменить пароль
- Нажать на кнопку «Сохранить»

Задача выхода из системы

Рассмотрим задачу выхода из системы. Данная задача включает в себя следующие этапы:

- Нажать на кнопку «Выйти» в шапке сайта.

Задачи, связанные с работой пользователя

Задача выбора тестирования

Рассмотрим задачу выбора тестирования. Данная задача включает в себя следующие этапы:

- Выбор вакансии, для которой будет открыт тест.
- Нажать на кнопку «Пройти».

Задача прохождения тестирования

Рассмотрим задачу прохождения тестирования. Данная задача включает в себя следующие этапы:

- Ответить на вопросы теста.
- Нажать на кнопку «Завершить тестирование».

Задача просмотра результата тестирования

Рассмотрим задачу просмотра результата тестирования. Данная задача включает в себя следующие этапы:

- После нажатия на кнопку «Завершить тестирование», результат появится на экране.

Задача просмотра анализа тестирования

Рассмотрим задачу просмотра анализа тестирования. Данная задача включает в себя следующие этапы:

- Нажать на кнопку «Анализ тестирования».
- Посмотреть на какие вопросы вы ответили верно, а на какие нет.
- Нажать на кнопку «Завершить обзор».

Задача просмотра сообщения администратора

Рассмотрим задачу отправки сообщения пользователю. Данная задача включает в себя следующие этапы:

- Нажать на круг справа в верхней панели.
- Нажать на кнопку «Сообщения».
- Посмотреть текст сообщения администратора.

Задача изменения пароля

Рассмотрим задачу изменения пароля. Данная задача включает в себя следующие этапы:

- Нажать на круг справа в шапке сайта.
- Нажать на кнопку «Редактировать профиль»
- Изменить пароль
- Нажать на кнопку «Сохранить»

Задача изменения ФИО

Рассмотрим задачу изменения ФИО. Данная задача включает в себя следующие этапы:

- Нажать на круг справа в шапке сайта.
- Нажать на кнопку «Редактировать профиль»
- Изменить ФИО
- Нажать на кнопку «Сохранить»

Задача выхода из системы

Рассмотрим задачу выхода из системы. Данная задача включает в себя следующие этапы:

- Нажать на кнопку «Выйти» в шапке сайта.

1.5. Графическое описание работы системы

Для удобства описания работы системы была использована графическая нотация UML.

1.5.1. Диаграмма прецедентов

Для описания системы на концептуальном уровне была спроектирована диаграмма прецедентов, представленная на Рисунке 1. Участниками являются зарегистрированный и незарегистрированный пользователи и администратор. Диаграмма показывает возможности участников выполнять те или иные действия в системе.

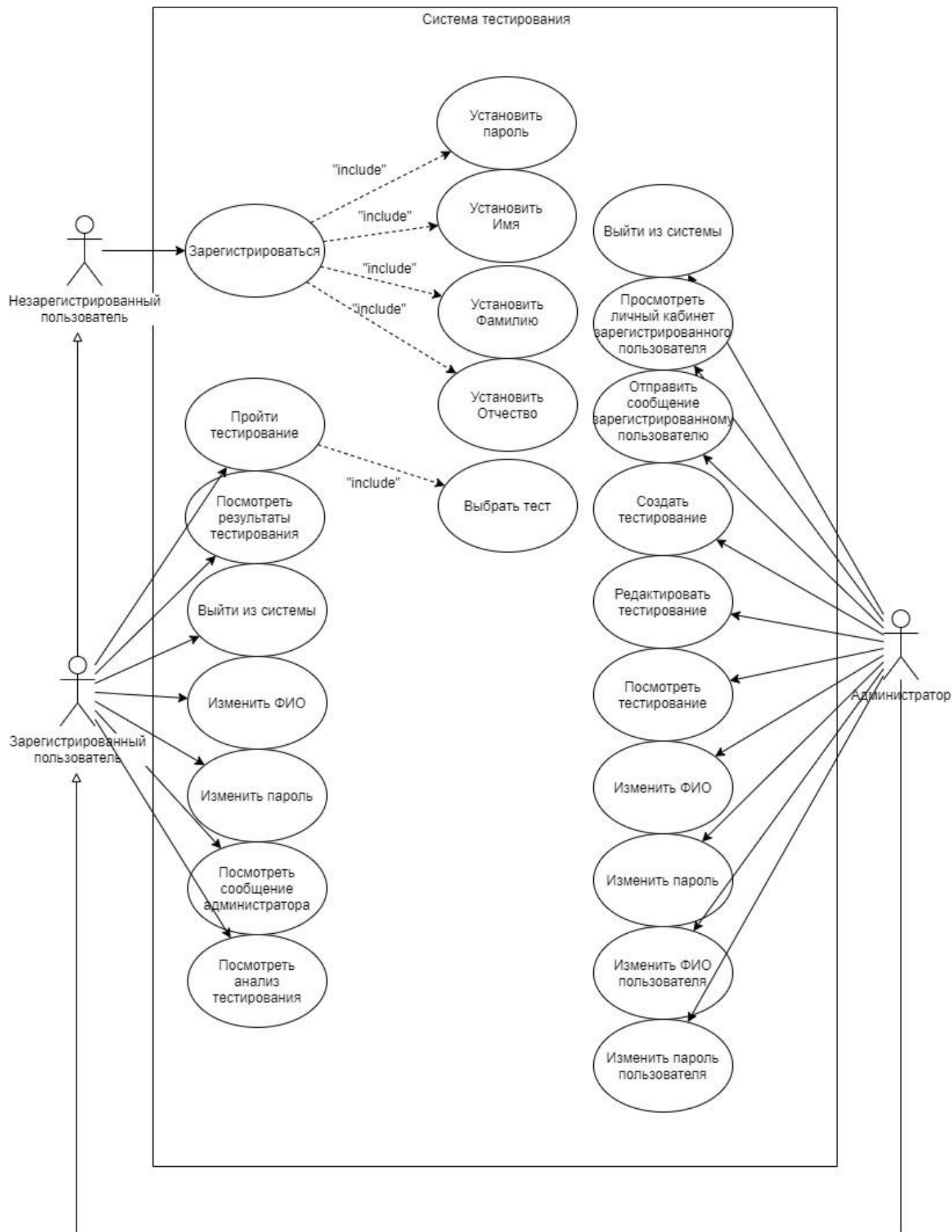


Рисунок 1 – Диаграмма Прецедентов

1.5.2. Диаграмма классов

На диаграмме представлены классы, используемые в проектируемой системе: Рисунок 2.

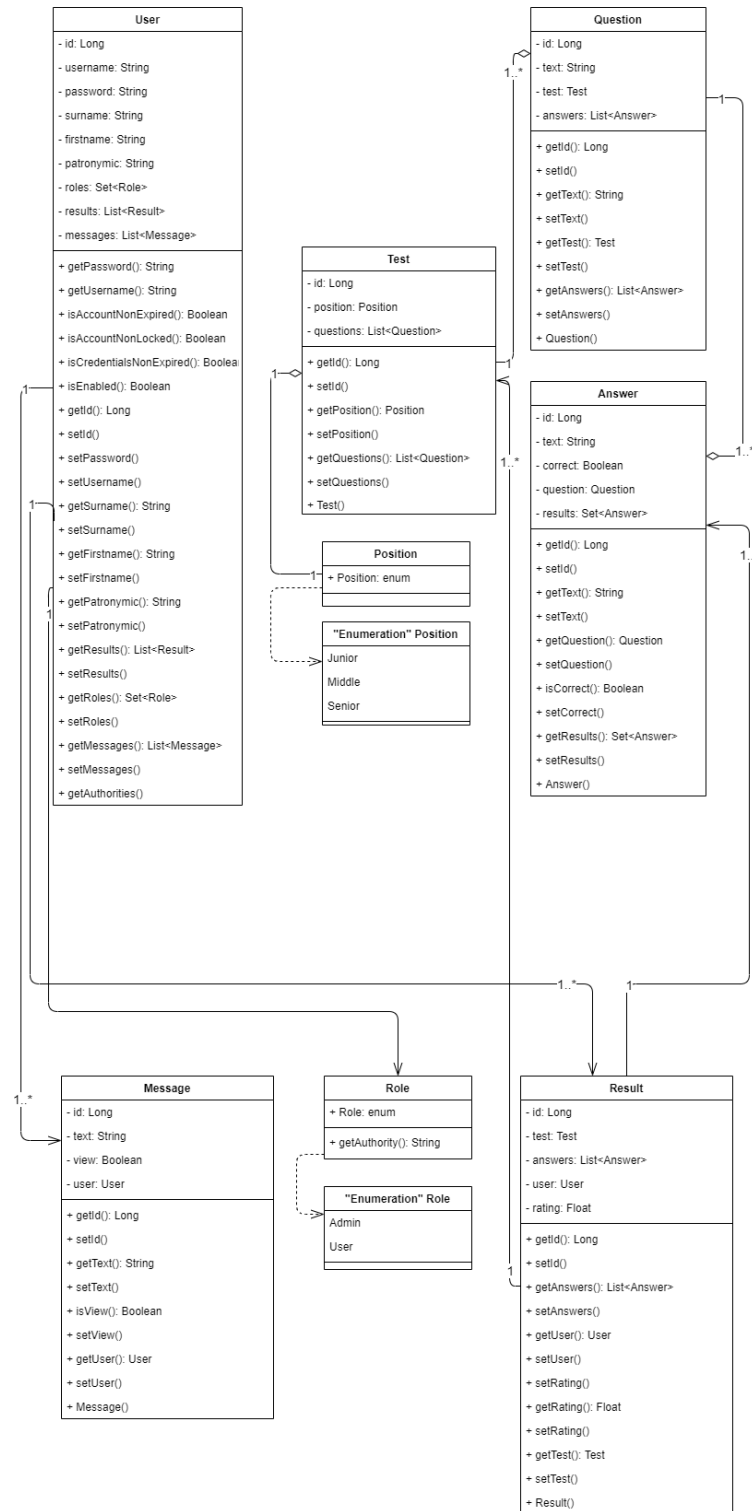


Рисунок 2 – Диаграмма Классов

1.5.3. Диаграмма последовательностей

Для демонстрации взаимодействия актеров системы была создана диаграмма последовательностей, представленная на Рисунке 3.

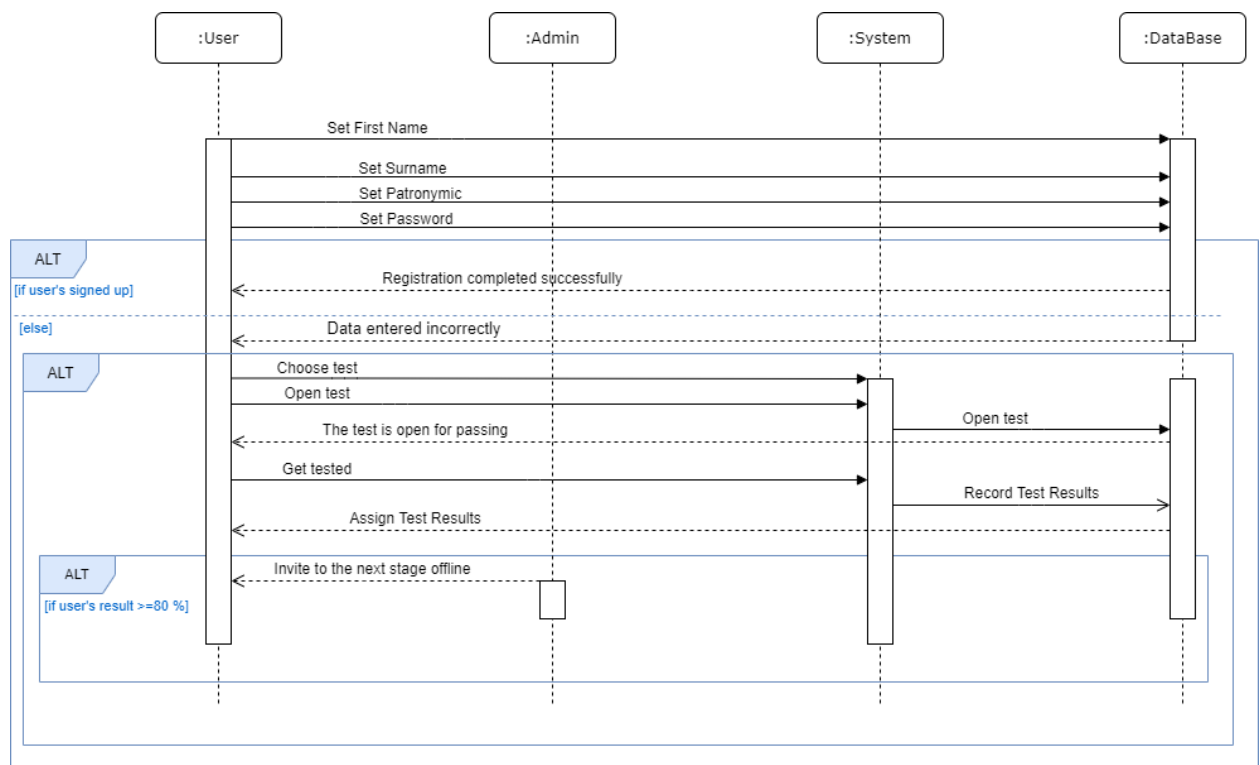


Рисунок 3 – Диаграмма последовательностей

1.5.4. Диаграмма коммуникаций

Для визуализации процесса работы пользователя в системе была спроектирована диаграмма коммуникаций, представленная на Рисунке 4.

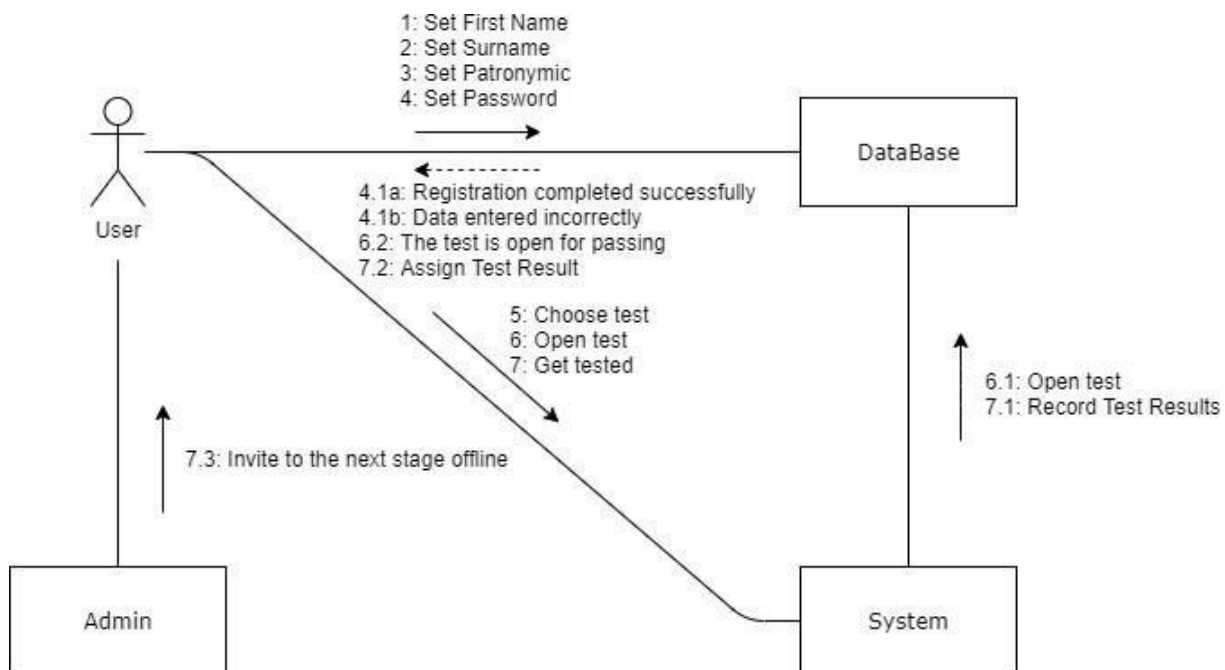


Рисунок 4 – Диаграмма коммуникаций

1.5.5. Диаграмма состояний

Диаграмма, представленная на Рисунке 5, показывает возможные состояния системы. При запуске приложения система находится в ожидании действия пользователя. В зависимости от выборов, совершаемых пользователем, меняется состояние системы.

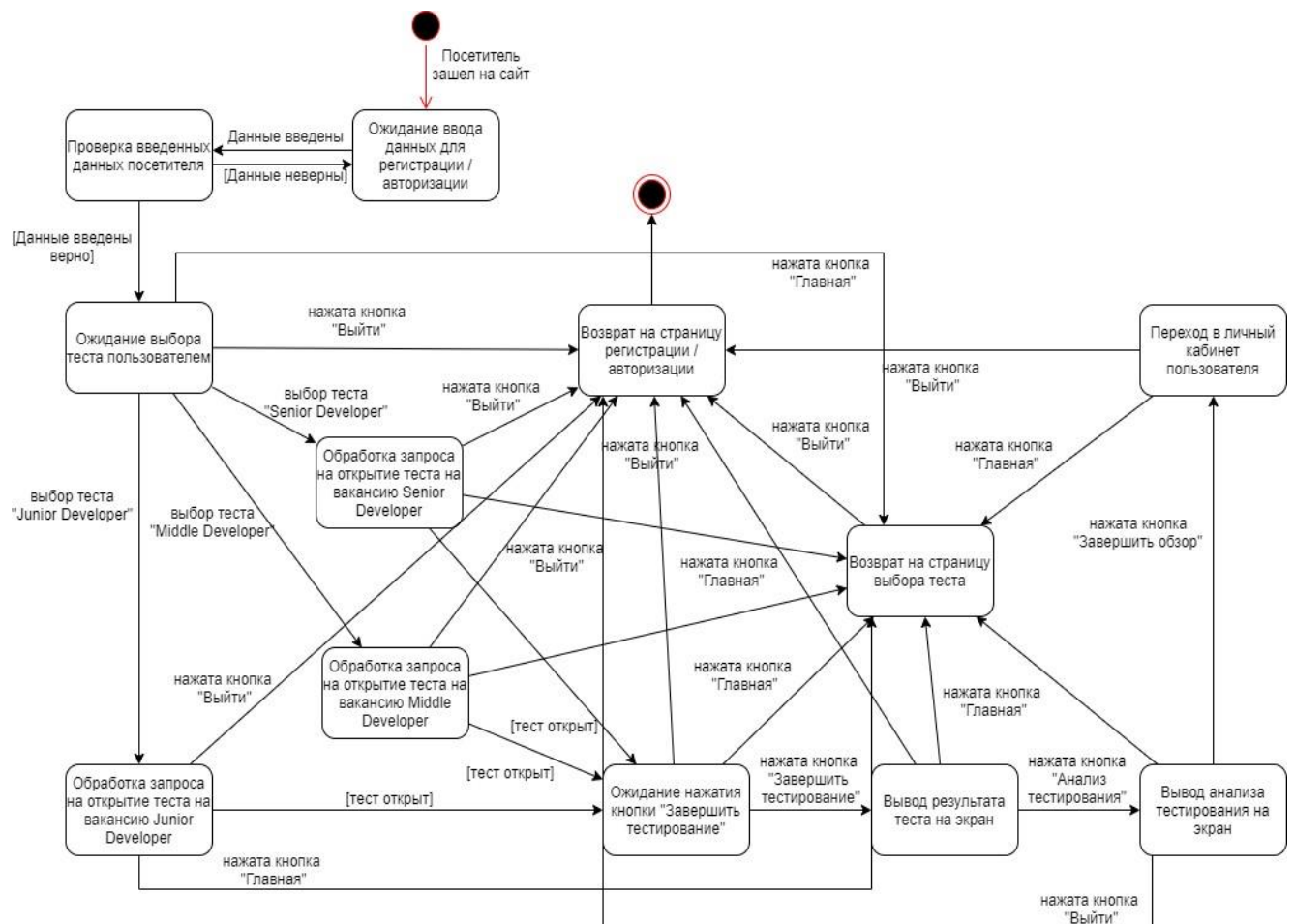


Рисунок 5 – Диаграмма состояний

1.5.6. Диаграмма активности

Диаграмма, представленная на Рисунке 6, показывает работу пользователя с системой.

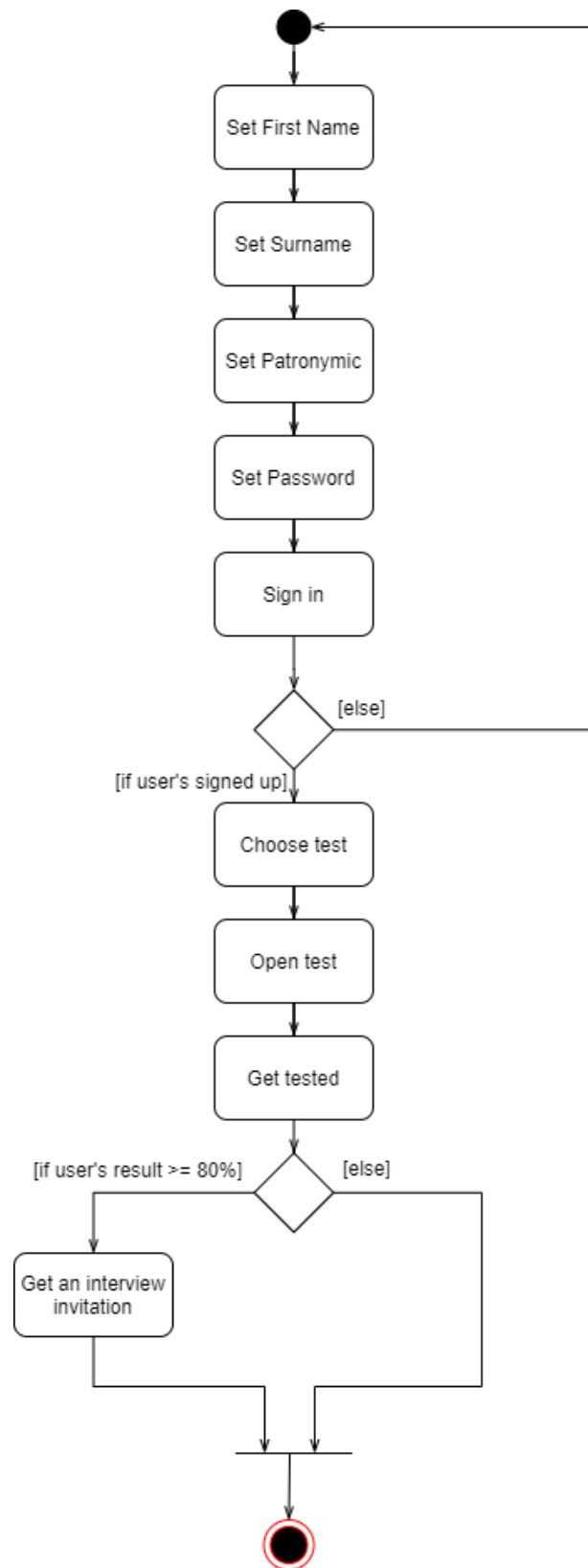


Рисунок 6 – Диаграмма активности

1.5.7. Диаграмма объектов

Диаграмма, представленная на Рисунке 7, описывает какие экземпляры классов будут взаимодействовать между собой.

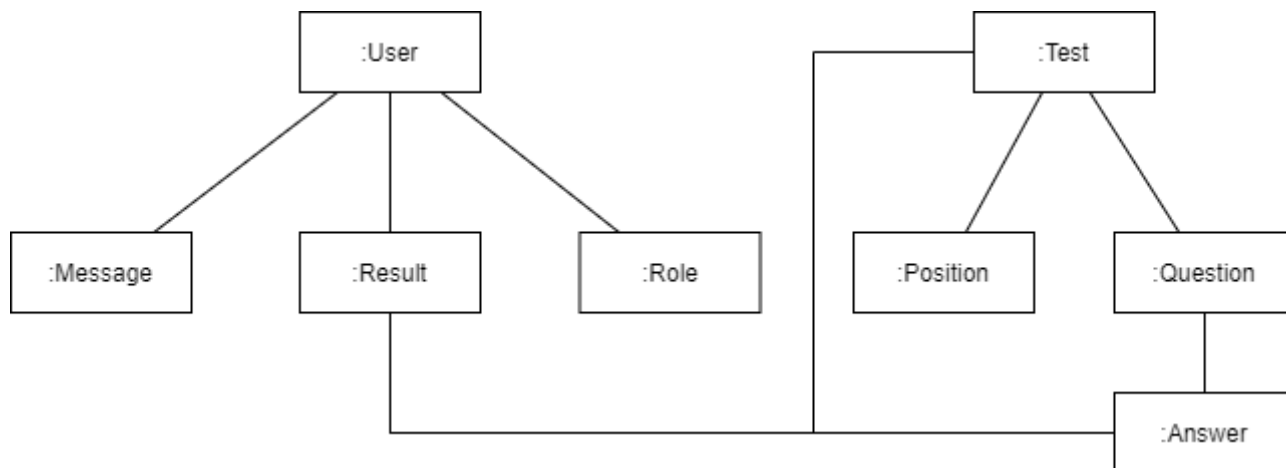


Рисунок 7 – Диаграмма объектов

1.5.8. Диаграмма взаимодействия

Диаграмма, представленная на Рисунке 8, показывает взаимодействие элементов системы.

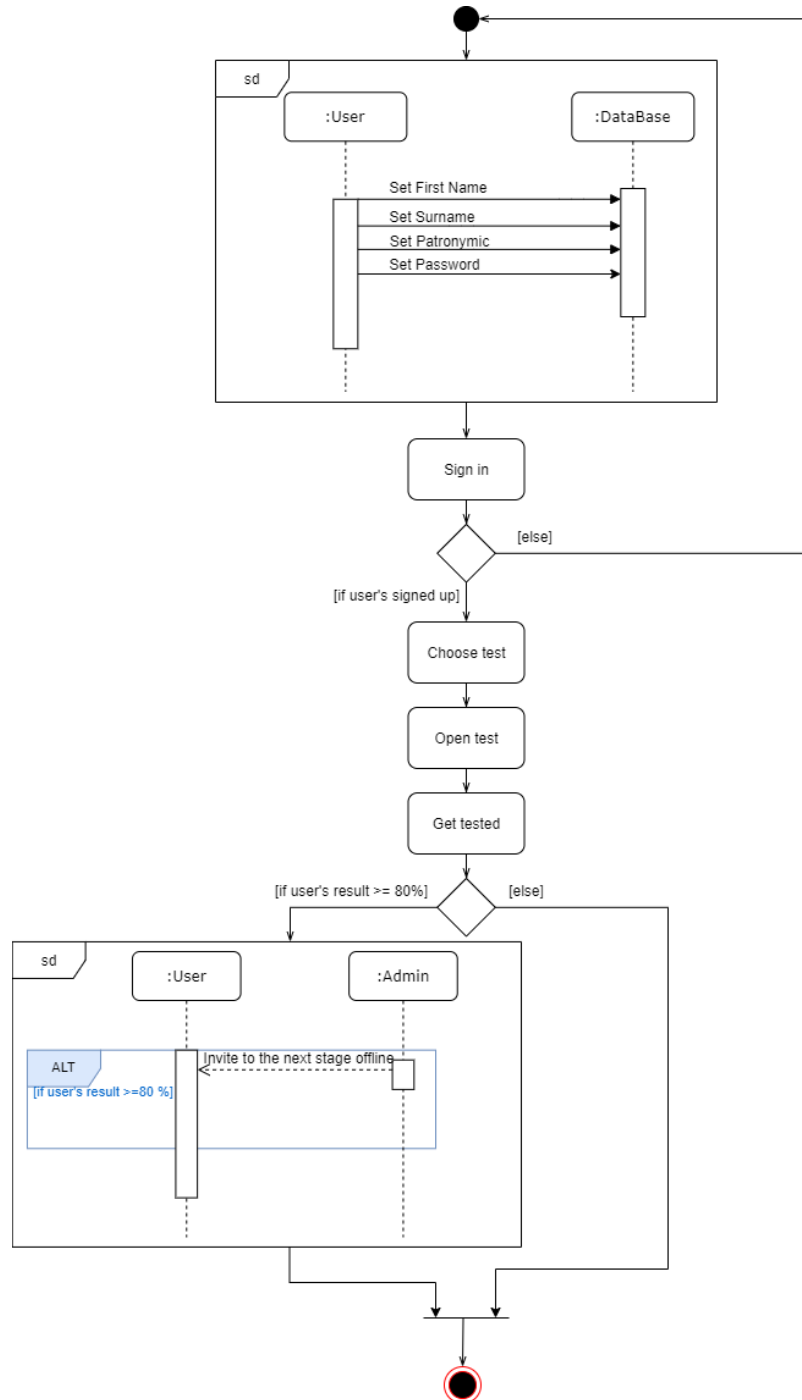


Рисунок 8 – Диаграмма взаимодействия

1.5.9. Диаграмма развертывания

Диаграмма, представленная на Рисунке 9, визуализирует элементы и компоненты программы, которые существуют на этапе ее исполнения.

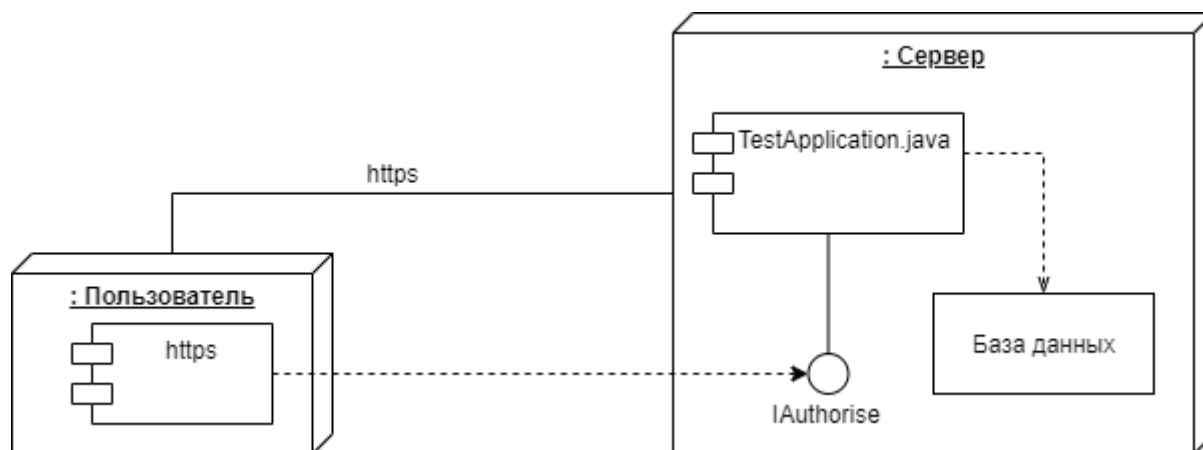


Рисунок 9 – Диаграмма развертывания

2. Продуктовые воронки

Для отслеживания поведения пользователей при взаимодействии с приложением, было принято решение о создании воронок с помощью сервиса Яндекс Метрика на основании целевых действий пользователя.

В соответствии с целями создания приложения были выделены три сценария, которые будут отслеживаться:

1. Тест пройден (для пользователей, которые зарегистрировались).
2. Тест пройден (для пользователей, которые авторизовались).
3. Получено приглашение на второй этап.

Описание:

- Тест пройден (для пользователей, которые зарегистрировались), Тест пройден (для пользователей, которые авторизовались).

Разбиение одной цели на две, отличающихся первым этапом, нужно для отслеживания новых пользователей, которые вошли в систему с помощью регистрации, а также пользователей, которые решили пройти тестирование снова и вошли с помощью авторизации. Сценарии отслеживают количество пользователей, которые выбрали какой-либо тест и завершили его прохождение.

- Получено приглашение на второй этап.

Сценарий показывает нам количество пользователей, чей результат составил 80% и выше правильных ответов.

<p>Тест пройден. Для пользователей, прошедших регистрацию</p>	<p>составная цель</p> <p>Регистрация (ID: 99409216):</p> <ul style="list-style-type: none"> url: содержит «/registration» <p>Выбор теста (ID: 99409219):</p> <ul style="list-style-type: none"> url: совпадает «https://java-testing-system.herokuapp.com/» <p>Прохождение теста (ID: 99409222):</p> <ul style="list-style-type: none"> url: содержит «/test» <p>Результат тестирования (ID: 99409225):</p> <ul style="list-style-type: none"> url: содержит «/record» 	99405355	
<p>Тест пройден. Для пользователей, прошедших авторизацию</p>	<p>составная цель</p> <p>Авторизация (ID: 99409078):</p> <ul style="list-style-type: none"> url: содержит «/login» <p>Выбор теста (ID: 99409081):</p> <ul style="list-style-type: none"> url: совпадает «https://java-testing-system.herokuapp.com/» <p>Прохождение теста (ID: 99409084):</p> <ul style="list-style-type: none"> url: содержит «/test» <p>Результат тестирования (ID: 99409087):</p> <ul style="list-style-type: none"> url: содержит «/record» 	99409075	
<p>Получено приглашение на второй этап</p>	<p>составная цель</p> <p>Личный кабинет:</p> <ul style="list-style-type: none"> url: содержит «/profile» <p>Сообщения:</p> <ul style="list-style-type: none"> url: содержит «/message» 	99659647	

Рисунок 10 –Яндекс Метрика. Настройка воронок

Результаты работы первого сценария - Тест пройден (для пользователей, которые зарегистрировались) за 24.04.2020 – 26.04.2020 показаны на Рисунке 11:

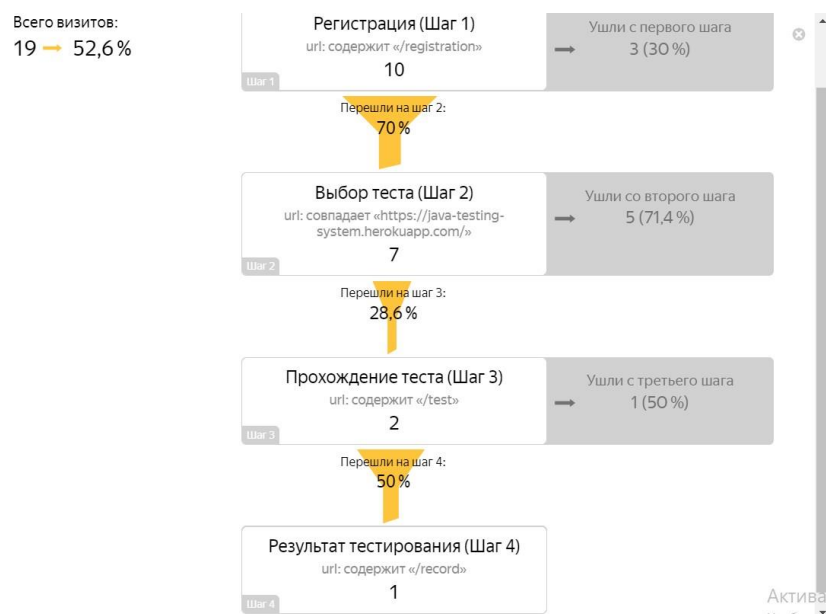


Рисунок 11 –Яндекс Метрика. Воронка 1.

Результаты работы второго сценария - Тест пройден (для пользователей, которые авторизовались) за 24.04.2020 – 26.04.2020 показаны на Рисунке 12:

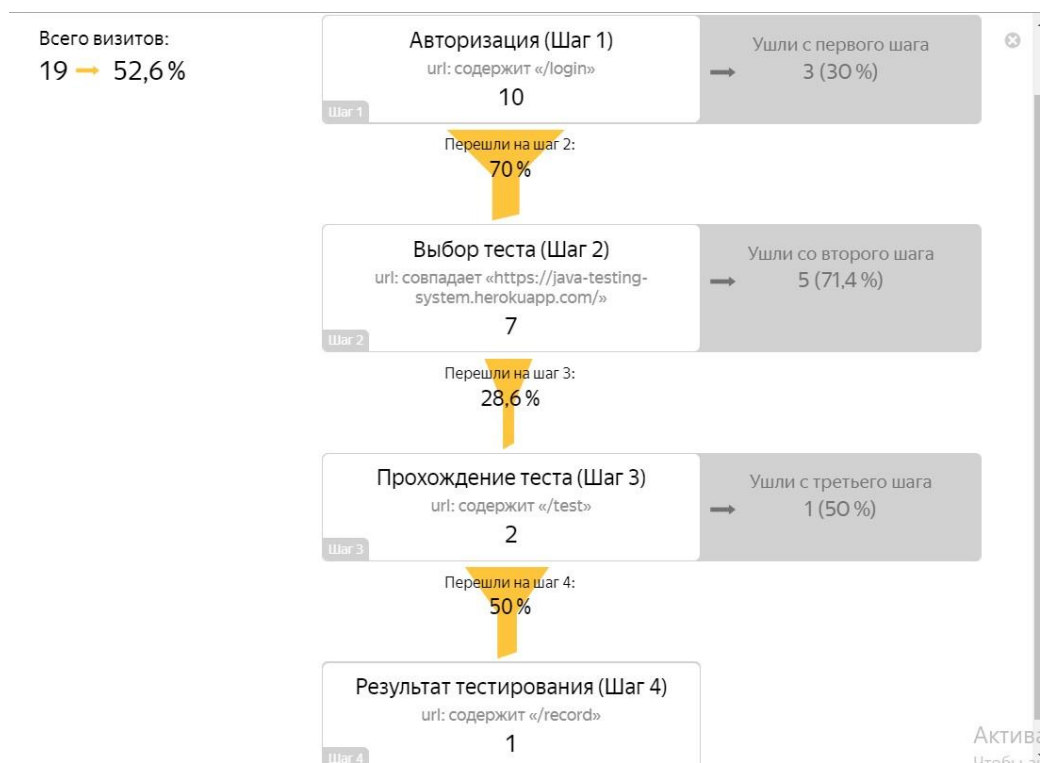


Рисунок 12 –Яндекс Метрика. Воронка 2.

Результаты работы третьего сценария - Получено приглашение на второй этап за 26.04.2020 –показаны на Рисунке 13:

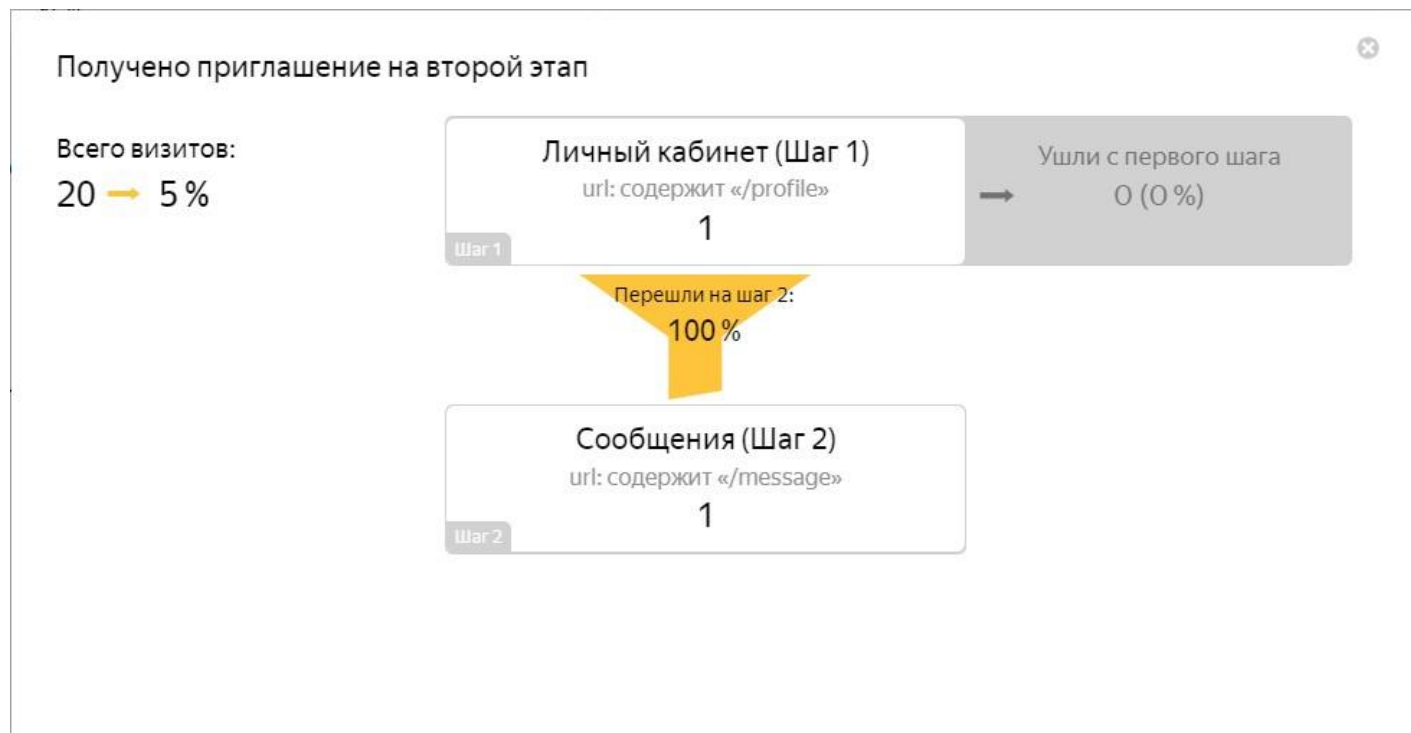


Рисунок 13 –Яндекс Метрика. Воронка 3.

Отчеты:

Цель: понимание предпочтений целевой аудитории.

1. Источники, сводка. Нужен для понимания откуда приходят посетители на сайт и в каком количестве. Рисунок 14.

Источники, сводка

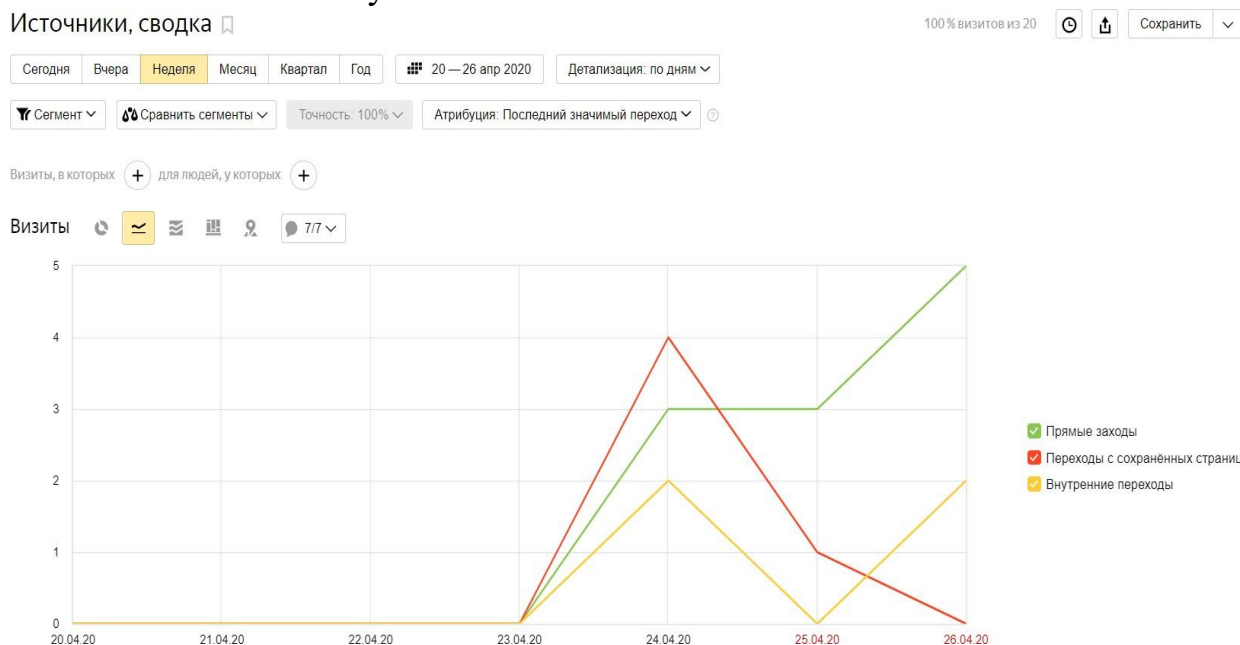


Рисунок 14 –Яндекс Метрика. Отчёт 1.

2. Устройства. Рисунок 15.

Устройства

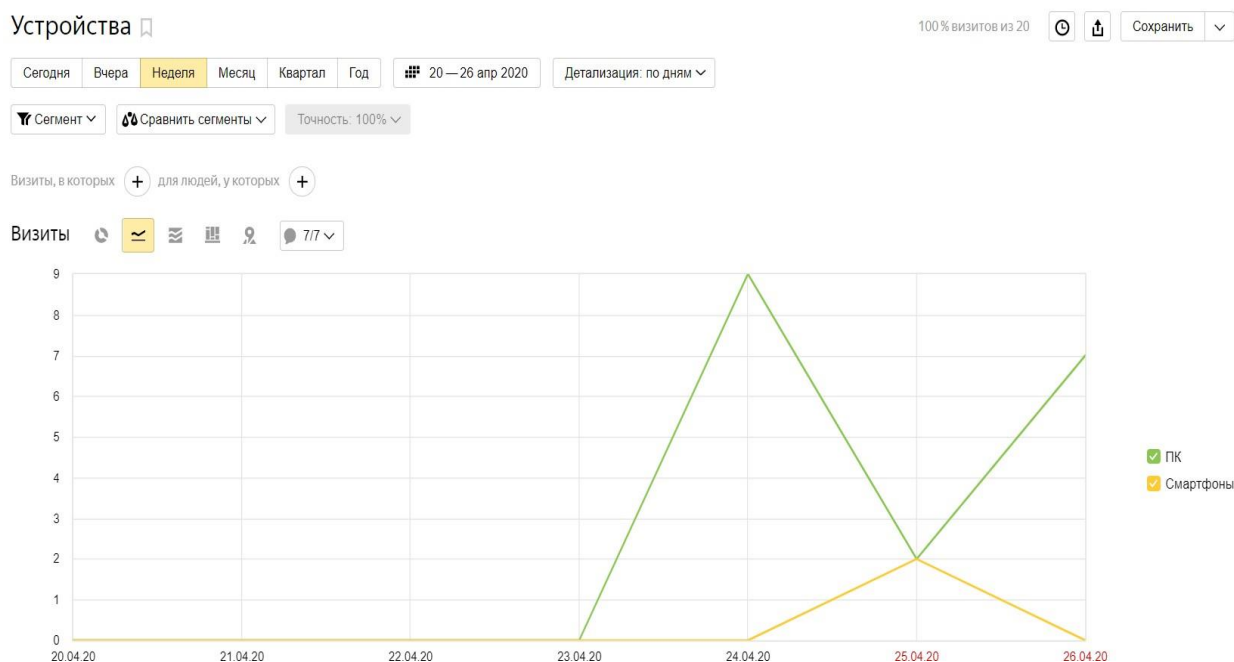


Рисунок 15 –Яндекс Метрика. Отчёт 2.

3. Возраст. Рисунок 16.

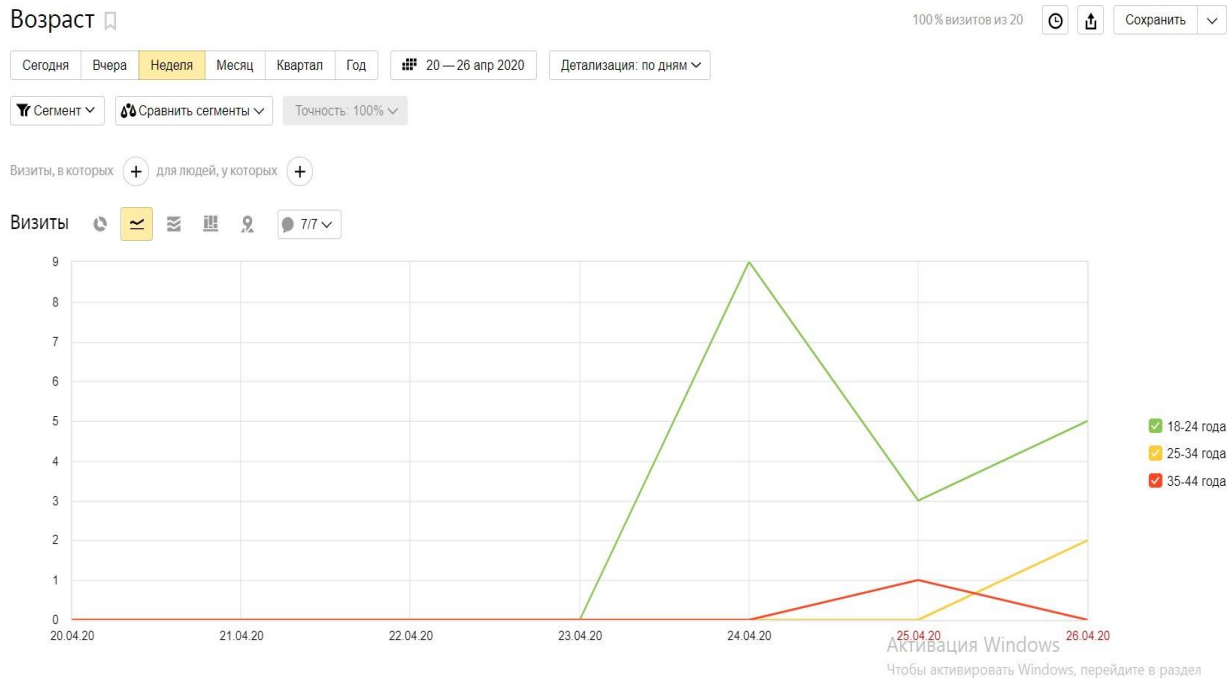


Рисунок 16 –Яндекс Метрика. Отчёт 3.

4. Сводка от Яндекс Метрики. Рисунок 17.

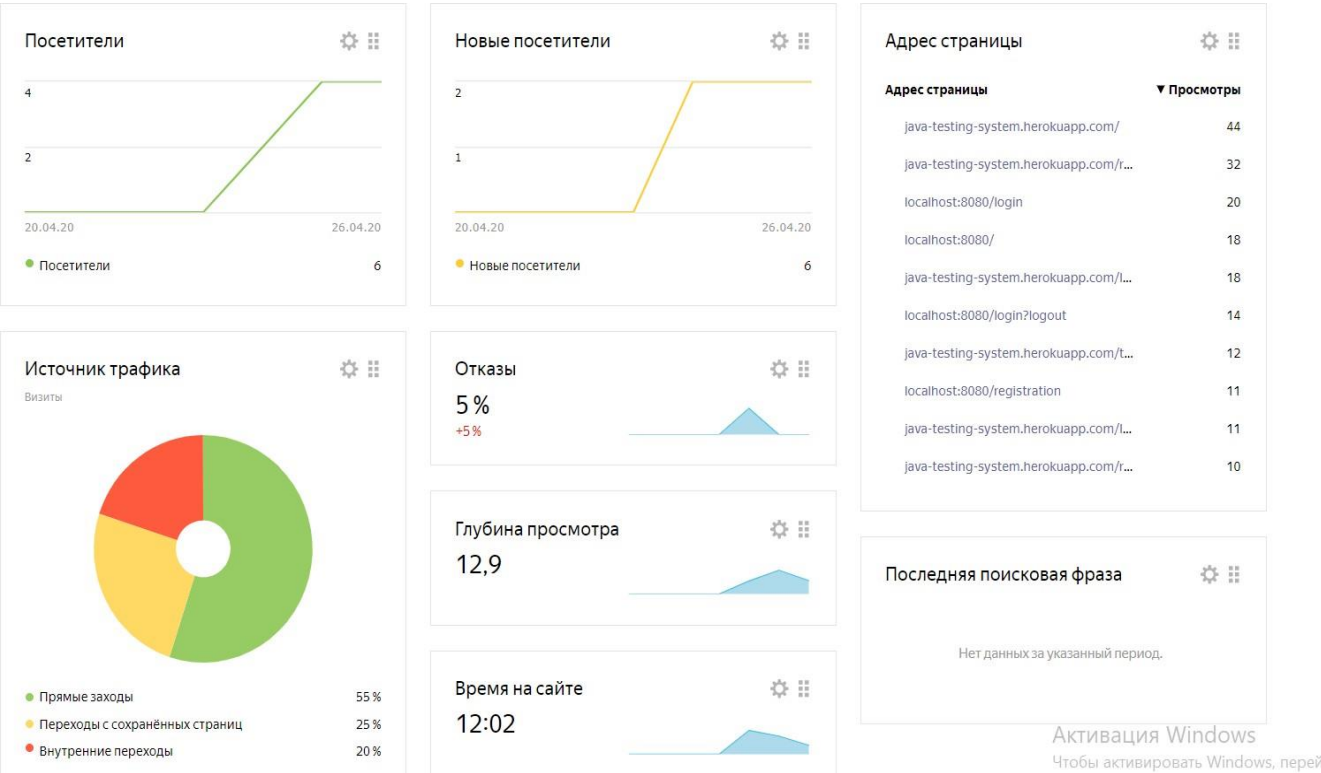


Рисунок 17 –Яндекс Метрика. Сводка.

3. Реализация приложения

3.1 Анализ средств реализации

В качестве средств реализации системы тестирования по Java были выбраны следующие технологии:

1. В качестве СУБД была выбрана PostgreSQL в силу относительной легковесности и высокой производительности, а также за счёт поддержки данной СУБД schema-less данных, такие как JSON.
2. Для реализации клиентской части приложения были выбраны технологии Html5, CSS3 – стандартный набор для вёрстки сайта, bootstrap4 из-за таких преимуществ как: адаптивность, кросс-браузерность, совместимость с Google Chrome и ускорение разработки за счёт готовой разметки.
3. Для реализации серверной части системы был выбран высокоуровневый язык программирования Java, так как он является проверенным временем выбором. Существует большое количество различных материалов и обучающих статей по написанию web-приложений на Java.
4. Фреймворк Spring был выбран, так как он является наиболее популярной средой и обладает такими преимуществами как: ускорение разработки на Java, большое количество сторонних библиотек, высокая производительность.

3.2 Проектирование системы тестирования

Проектирование приложения включает в себя:

- Проектирование диаграммы классов
- Проектирование диаграммы объектов
- Проектирование диаграммы развертывания

3.2.1 Диаграмма классов

В ходе проектирования приложения была составлена диаграмма классов. На рисунке 18 представлен класс User, необходимый для регистрации и авторизации пользователя.

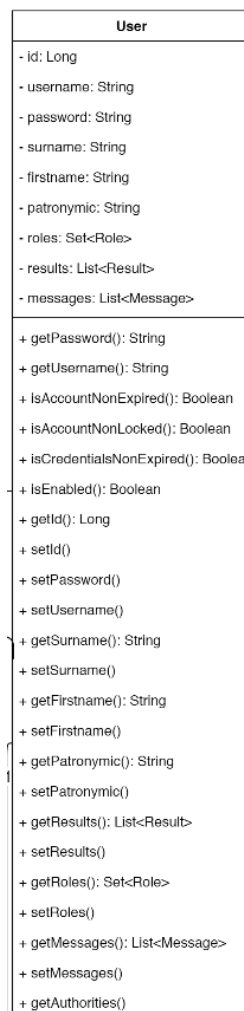


Рисунок 18 – класс User

Для реализации системы тестирования были спроектированы классы Test, Question, Answer, Result, представленные на Рисунке 19, 20, 21, 22 соответственно.

Test
<ul style="list-style-type: none"> - id: Long - position: Position - questions: List<Question>
<ul style="list-style-type: none"> + getId(): Long + setId() + getPosition(): Position + setPosition() + getQuestions(): List<Question> + setQuestions() + Test()

Рисунок 19 – класс Test

Question
<ul style="list-style-type: none"> - id: Long - text: String - test: Test - answers: List<Answer>
<ul style="list-style-type: none"> + getId(): Long + setId() + getText(): String + setText() + getTest(): Test + setTest() + getAnswers(): List<Answer> + setAnswers() + Question()

Рисунок 20 – класс Question

Answer
<ul style="list-style-type: none"> - id: Long - text: String - correct: Boolean - question: Question - results: Set<Answer>
<ul style="list-style-type: none"> + getId(): Long + setId() + getText(): String + setText() + getQuestion(): Question + setQuestion() + isCorrect(): Boolean + setCorrect() + getResults(): Set<Answer> + setResults() + Answer()

Рисунок 21 – класс Answer

Result
<ul style="list-style-type: none"> - id: Long - test: Test - answers: List<Answer> - user: User - rating: Float
<ul style="list-style-type: none"> + getId(): Long + setId() + getAnswers(): List<Answer> + setAnswers() + getUser(): User + setUser() + setRating() + getRating(): Float + setRating() + getTest(): Test + setTest() + Result()

Рисунок 22 – класс Result

Для реализации возможности отправки Администратором сообщения Пользователю был спроектирован класс Message, представленный на Рисунке 23.



Рисунок 23 – класс Message

Для реализации разграничения ролей Пользователя и Администратора было спроектировано перечисление Role, представленное на Рисунке 24.

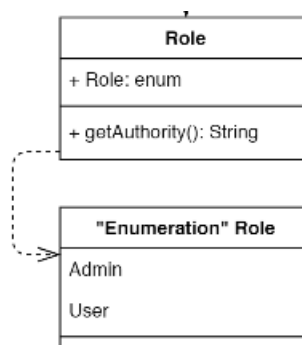


Рисунок 24 – перечисление Role

Для реализации системы тестирования для трёх различных вакансий было спроектировано перечисление Position, представленное на Рисунке 25.

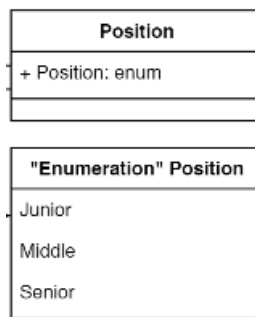


Рисунок 25 – перечисление Position

3.2.2 Диаграмма объектов

Для лучшего понимания диаграммы классов была составлена диаграмма объектов, представленная на Рисунке 26.

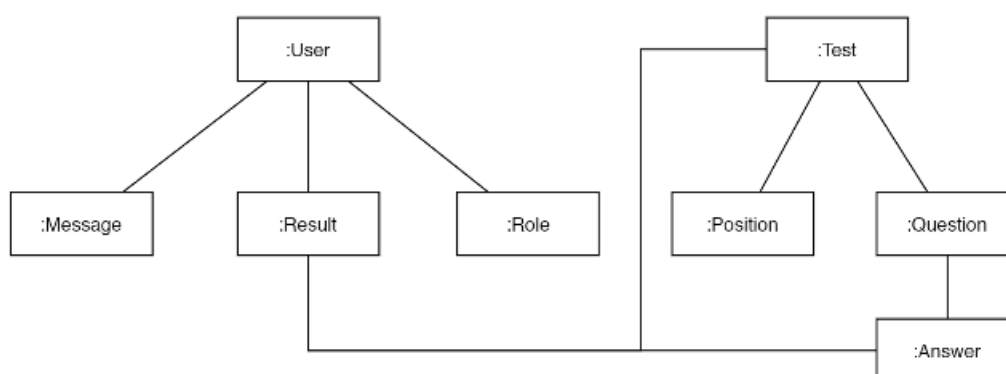


Рисунок 26 – диаграмма объектов

3.2.3 Диаграмма развертывания

Для лучшего понимания работы приложения была составлена диаграмма развертывания, представленная на Рисунке 27, которая визуализирует элементы и компоненты разрабатываемой системы на этапе её исполнения.

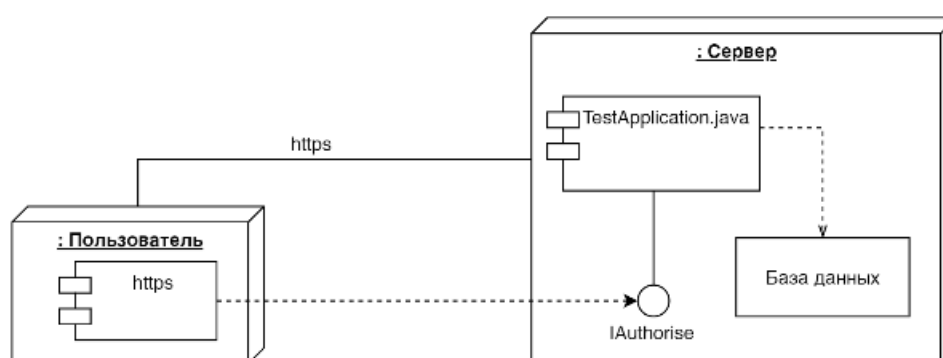


Рисунок 27 – диаграмма развертывания

3.3 Разработка приложения

3.3.1 Frontend разработка

Для регистрации пользователя была создана страница registration - скриншот представлен на Рисунке 28.

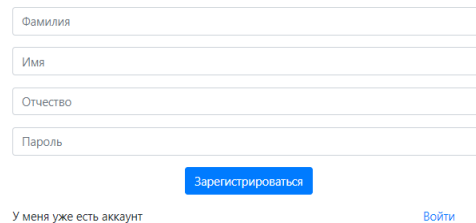


Рисунок 28 - скриншот registration

После заполнения всех полей и нажатия на кнопку «Зарегистрироваться» происходит переход на главную страницу приложения – Выбор теста, скриншот которой представлен на Рисунке 29.

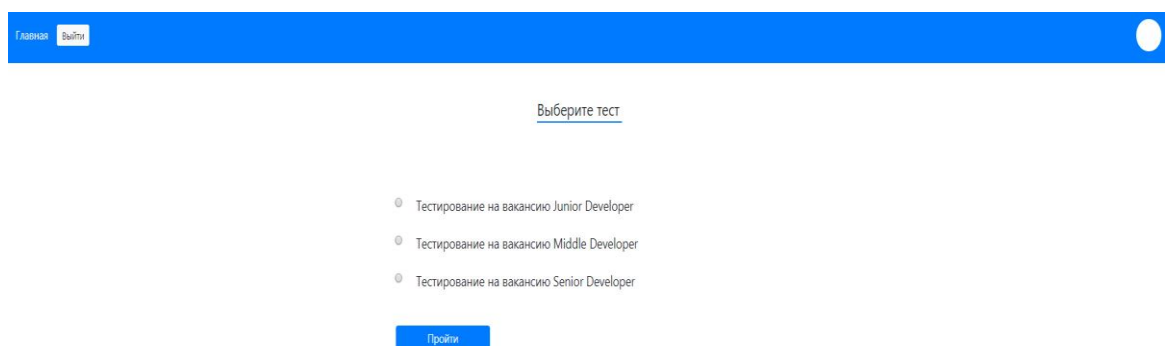


Рисунок 29 – скриншот страницы выбора теста

При выборе одного из вариантов тестирования, например, «Тестирование на вакансию Senior Developer» и нажатии на кнопку «Пройти» происходит переход на страницу test, скриншот которой представлен на Рисунке 30.

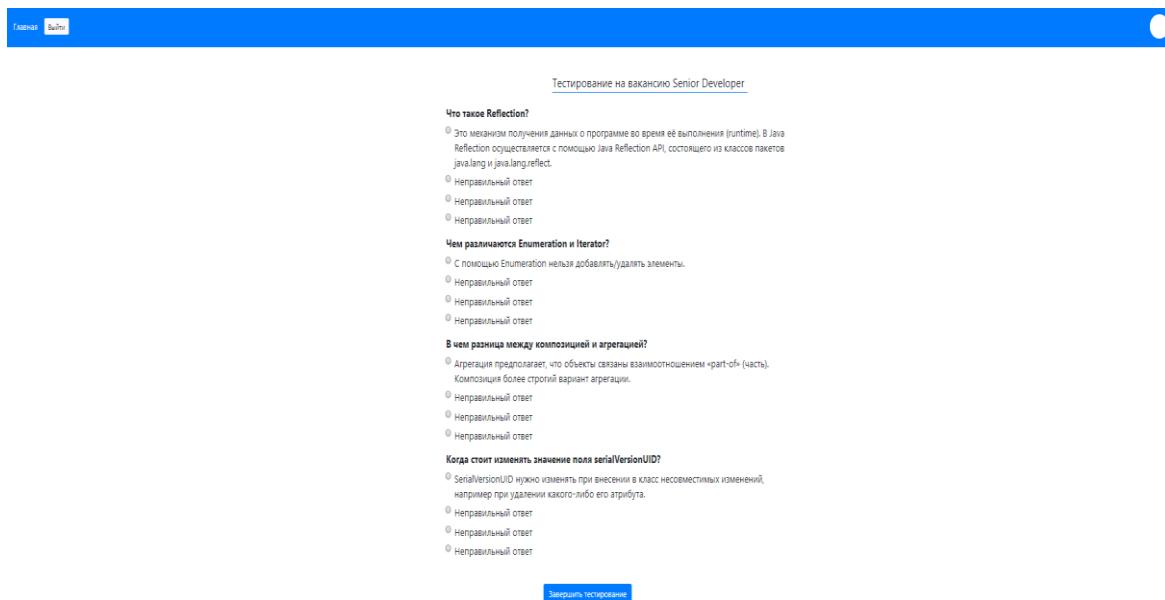


Рисунок 30 – скриншот test

После выбора ответов и нажатия на кнопку «Завершить тестирование», мы переходим на страницу record, скриншот которой представлен на Рисунке 31.

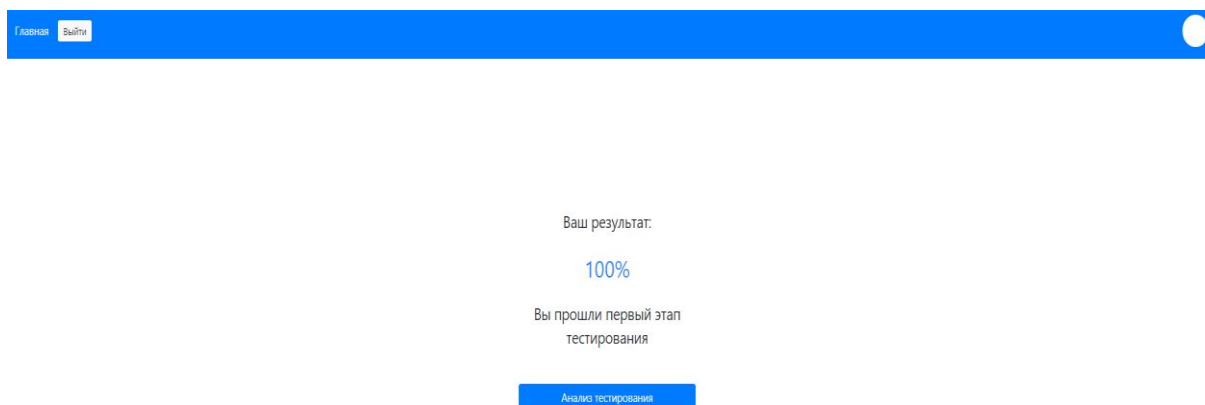


Рисунок 31 – скриншот record

При нажатии на кнопку «Анализ тестирования» совершается переход на страницу analysis/, скриншот которой представлен на Рисунке 32.

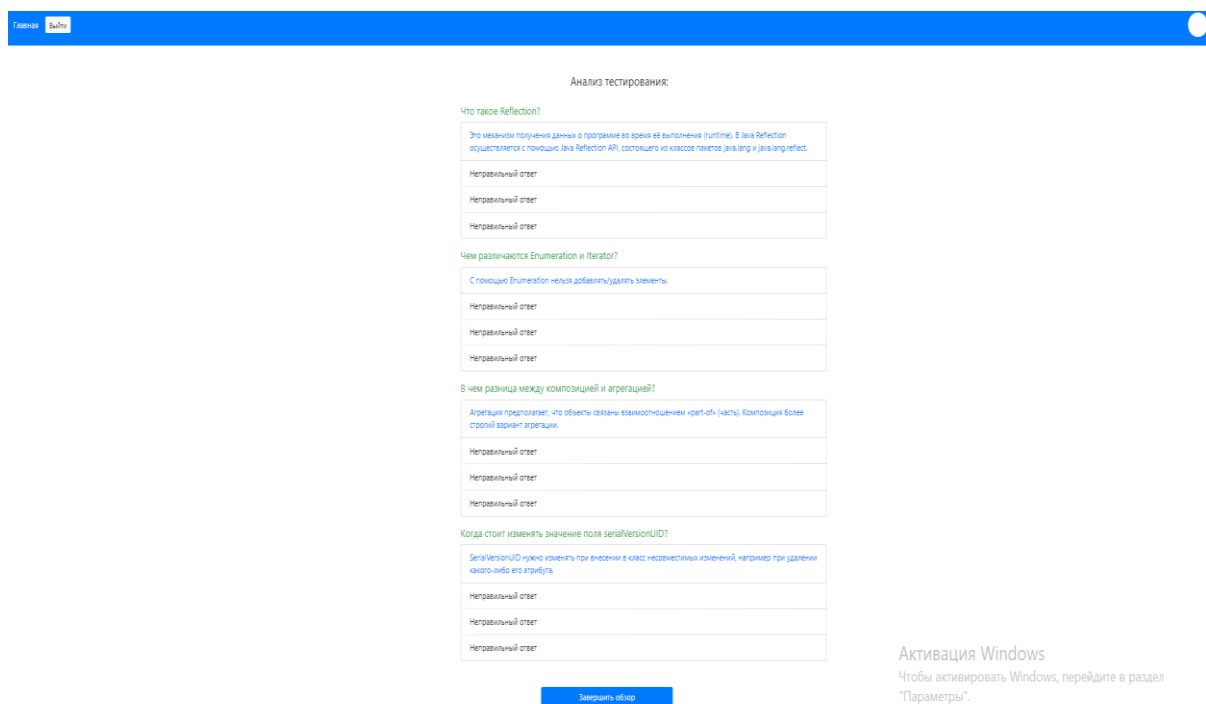


Рисунок 32 – скриншот analysis/

При нажатии на кнопку «Завершить обзор», мы переходим на страницу profile/, где можем видеть все результаты тестирований, пройденных нами. В личном кабинете можно редактировать профиль – изменить ФИО и пароль на странице update, а также посмотреть сообщение, которое прислал Администратор – страница message/. Скриншоты данных страниц представлены на Рисунках 33, 34, 35 соответственно.

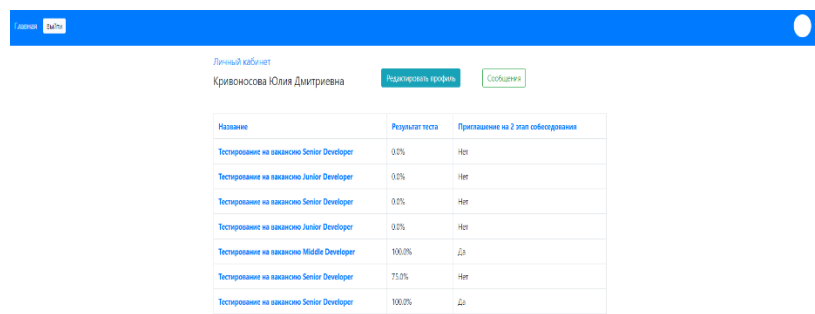
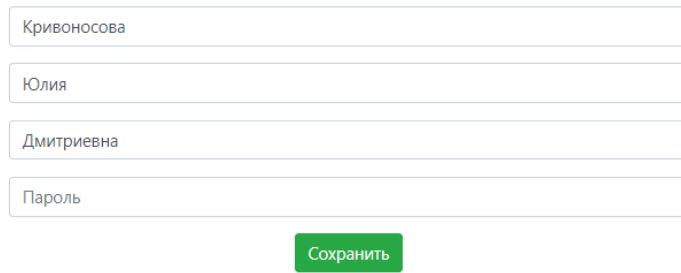


Рисунок 33 – страница profile/



Кривоносова

Юлия

Дмитриевна

Пароль

Сохранить

Рисунок 34 – скриншот update



Рисунок 35 – скриншот message

3.3.2 Backend разработка

Реализованы следующие модули, представленные в виде вкладок в веб-интерфейсе:

- Admin – панель Администратора с возможностью создания, редактирования и удаления тестов, отправки сообщения пользователю, просмотра личного кабинета пользователя, а также редактирования данных пользователя
- Analysis – анализ ответов пользователя
- CreateTest – создание тестирования
- Index – страница выбора теста

- Login – страница авторизации
- Messages – просмотр сообщений
- Profile – личный кабинет пользователя
- Registration – страница регистрации
- Result – результат тестирования
- ShowTest – просмотр тестирования администратором
- Test – страница прохождения тестирования
- UpdateProfile – редактирование профиля
- UpdateTest – редактирование теста

3.4 Тестирование

3.4.1 Дымовое тестирование

Необходимо проверить работоспособность приложения на следующих основных сценариях:

1. Для пользователя:

- Регистрация пользователя
- Выбор и прохождение тестирования
- Анализ тестирования
- Изменение данных в личном кабинете

2. Для администратора:

- Просмотр, создание, изменение, удаление тестирования
- Отправка сообщения пользователю
- Просмотр, изменение данных администратора, пользователя

Дымовое тестирование проводилось ручным способом в браузере Google Chrome версии 81.0.4044.122. Результаты, полученные в ходе тестирования, представлены в Таблице 1.

Таблица 1 – результаты дымового тестирования

Сценарий	Результат
Регистрация пользователя	Пройден
Выбор и прохождение тестирования	Пройден
Анализ тестирования	Пройден
Изменение данных в личном кабинете	Пройден
Просмотр, создание, изменение, удаление тестирования	Пройден
Отправка сообщения пользователю	Пройден
Просмотр, изменение данных администратора, пользователя	Пройден

3.4.2 Юзабилити тесты

Для проведения юзабилити тестирования были отобраны 3 человека, не пользовавшиеся ранее данным приложением. Необходимо проверить следующие основные сценарии взаимодействия пользователя с приложением:

- Регистрация пользователя
- Выбор и прохождение тестирования
- Анализ тестирования
- Редактирование профиля
- Просмотр тестирования

Результаты тестирования отображены в Таблице 2.

Таблица 2 – Результаты юзабилити тестирования

Сценарий	Пользователь 1	Пользователь 2	Пользователь 3
Регистрация пользователя	Пройден	Пройден	Пройден
Выбор и прохождение тестирования	Пройден	Пройден	Пройден
Анализ тестирования	Пройден	Пройден	Пройден
Редактирование профиля	Пройден	Пройден	Пройден
Просмотр тестирования	Пройден	Пройден	Пройден
Выход из системы	Пройден	Пройден	Пройден

Заключение

В результате работы было реализовано приложение, которое пользователям проходить тестирование на такие вакансии как: Junior Developer, Middle Developer, Senior Developer. Были выполнены следующие задачи:

1. Разработана Frontend часть приложения
2. Разработана Backend часть приложения
3. Была создана связь между Frontend и Backend частями приложения
4. Разработана базы данных, расположенная на удаленном сервере
5. Составлена документация
6. Спроектирована система
7. Проведен анализ предметной области