

МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)**

Факультет *Компьютерных наук*

Кафедра *программирования и информационных технологий*

Система оплаты услуг ЖКХ «УслугиВдом»

Курсовой проект

09.03.04 Программная инженерия

Преподаватель _____ В.С. Тарасов __.__.20__

Обучающийся _____ Ю.В. Шишко, 3 курс, д/о

Обучающийся _____ Д.А. Михалев, 3 курс, д/о

Обучающийся _____ М.В. Евлаков, 3 курс, д/о

Воронеж 2021

Содержание

| | |
|--|----|
| Введение..... | 3 |
| 1 Постановка задачи | 4 |
| 2 Глоссарий..... | 5 |
| 3 Анализ предметной области | 7 |
| 3.1 Анализ существующих решений | 7 |
| 3.2 Анализ целевой аудитории..... | 10 |
| 3.3 Пользователи системы | 10 |
| 4 Анализ задачи..... | 12 |
| 4.1 Варианты использования системы | 12 |
| 4.2 Общая структура системы | 13 |
| 4.3 Взаимодействие компонентов системы | 14 |
| 4.4 Взаимодействие в системе..... | 15 |
| 4.5 Варианты состояния системы | 16 |
| 4.6 Развертывание системы | 17 |
| 4.7 IDEF0 | 17 |
| 4.8 Диаграмма классов | 19 |
| 4.9 ER-диаграмма..... | 20 |
| 5 Анализ средств реализации | 22 |
| 6 Сценарии воронок конверсии | 24 |
| 7 Тестирование | 25 |
| 7.1 Дымовое тестирование | 25 |
| 7.2 UI тесты | 26 |
| 7.3 Юзабилити тесты | 33 |
| 8. Используемые технологии | 35 |
| Заключение | 36 |
| Список использованных источников | 37 |

Введение

В настоящее время непросто помнить о платежах, которые необходимо производить раз в месяц и которые уже прижились в нашей жизни. Держать в голове все будущие необходимые платежи за коммунальные услуги довольно затруднительно, учитывая количество иной важной информации, которую необходимо помнить. Можно вести записи на бумаге, но это не всегда удобно, так как блокнот или тетрадь не всегда доступны. Доступность информации в различные моменты времени с различных устройств упрощает процесс слежения за платежами и становится преимуществом перед контролем на бумаге или в голове.

Сайт должен облегчать оплату и ведение статистики коммунальных платежей. Основная функциональность сайта должна включать в себя:

1. Работу с информационными ресурсами (новостями и документами).
2. Возможность производить онлайн оплату предоставляемых услуг.
3. Возможность производить мониторинг собственных чеков.
4. Возможность получать информации о задолженностях.
5. Возможность привязать Telegram-аккаунт для получения актуальных новостей и чеков об оплате через мессенджер.

Целью нашей командной работы является создание проекта, удовлетворяющего всем вышеперечисленным условиям и облегчающим оплату услуг ЖКХ.

1 Постановка задачи

Данный проект является информационным интернет-сервисом об оплате коммунальных услуг.

Система автоматизирует процесс оплаты коммунальных платежей, позволяет следить за статистикой расходов на платежи за прошлые месяцы, позволяет узнать важные новости (к примеру, ремонт на электростанции), которые напрямую влияют на качество жизни клиента сервиса.

Система предназначена для:

- упрощения процесса слежения за задолженностями;
- оплаты коммунальных услуг онлайн;
- получения информации о важных новостях;
- получение статистики расходов на коммунальные платежи прошлых месяцев.

Для достижения данной цели были выделены следующие подзадачи:

1. Разработка макета сайта.
2. Разработка Frontend части сервиса.
3. Разработка Backend части сервиса.
4. Создание связи между Frontend и Backend частями приложения.
5. Разработка базы данных.
6. Разработка Телеграм бота.

2 Глоссарий

Личный кабинет — это раздел сервиса, в котором Пользователь может получить доступ к своим данным.

MVC (Model-View-Controller) — схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

Model (база данных) — этот компонент отвечает за данные в проекте, а также реагирует на команды контроллера, изменяя свое состояние.

Графический интерфейс (View) — это компонент, отвечающий за взаимодействие с пользователем. То есть данный компонент определяет внешний вид приложения и способы его использования.

Контроллер (Controller) — это компонент, который управляет запросами пользователя. Его основная функция — вызывать и координировать действие необходимых ресурсов и объектов, нужных для выполнения действий, задаваемых пользователем. Обычно контроллер вызывает соответствующую модель для задачи и выбирает подходящий вид.

Frontend — клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

Backend — программно-аппаратная часть сервиса.

REST API — это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов.

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

Веб-сервис, интернет-сервис, система, веб-приложение, проект — идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя.

Гость — неавторизованный на веб-сервисе человек, пользующийся ограниченным функционалом веб-сервиса.

Пользователь — авторизованный на портале человек, пользующийся функционалом веб-сервиса.

Администратор — человек, имеющий доступ к расширенному функционалу веб-сервиса.

3 Анализ предметной области

3.1 Анализ существующих решений

Госуслуги

Является самым популярным интернет-сервисом для оплаты разного рода услуг, включая ЖКХ. Но отсутствует возможность использовать оплачивать коммунальные услуги без регистрации, т.к. наличие аккаунта является обязательным для использования предоставляемых услуг.

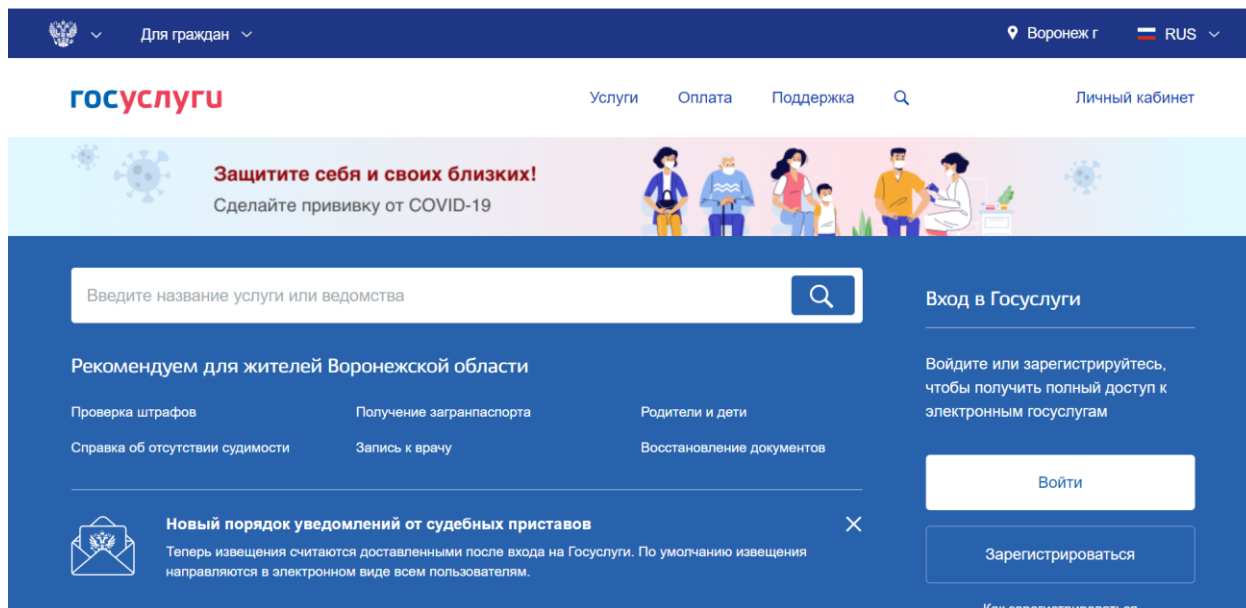


Рисунок 1 – Домашняя страница сервиса Госуслуги

Сбербанк

Крупнейший банк на территории Российской Федерации, предоставляющий большой спектр услуг по оплате через свой собственный сервис.

На сайте оплата услуг ЖКХ невозможна, для этого необходимо скачать приложение «СберБанк Онлайн». Также для оплаты услуг ЖКХ можно использовать только карты данного банка.

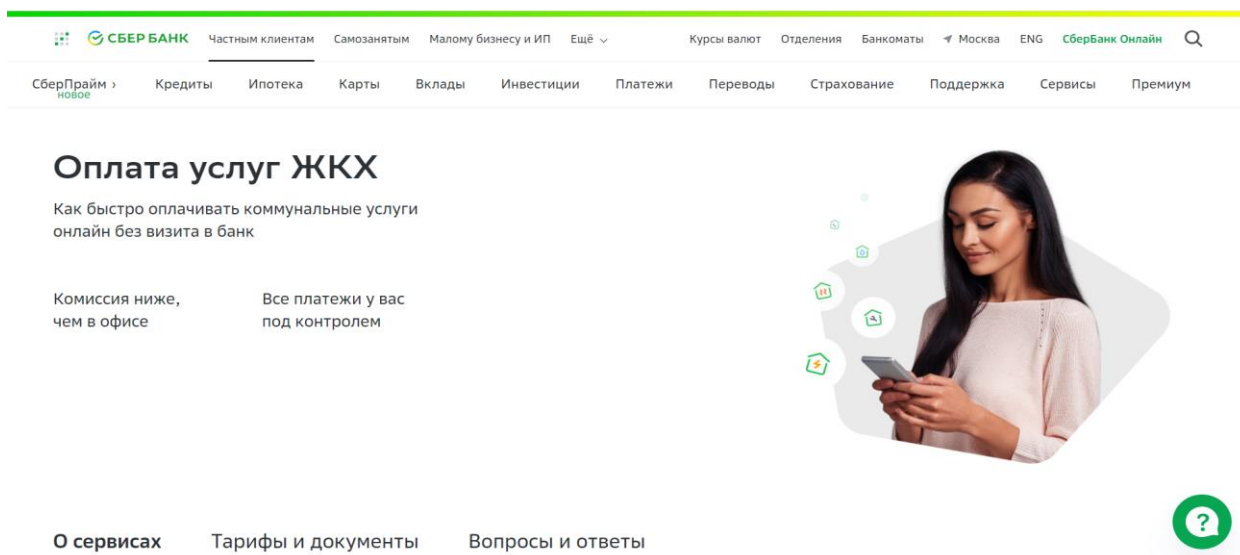


Рисунок 2 – Оплата услуг ЖКХ Сбербанка

ЖКХ-онлайн

Еще один из сервисов, выдаваемых на первых позициях при запросе в поисковике. Автоматическая система учета и реализации энергоресурсов, к сожалению, не позволяет посмотреть ничего, кроме контактов и FAQ, без логина, что является существенным минусом и сильно влияет на конверсию.



Рисунок 3 – Оплата услуг в ЖКХ-онлайн

ЖКХ-клик

Приятный на первый взгляд сервис для оплаты широкого спектра коммунальных услуг, но в нем нет личного кабинета, а, соответственно, нет и

никакой возможности посмотреть статистику и сделать какой-либо анализ собственных затрат на услуги ЖКХ.

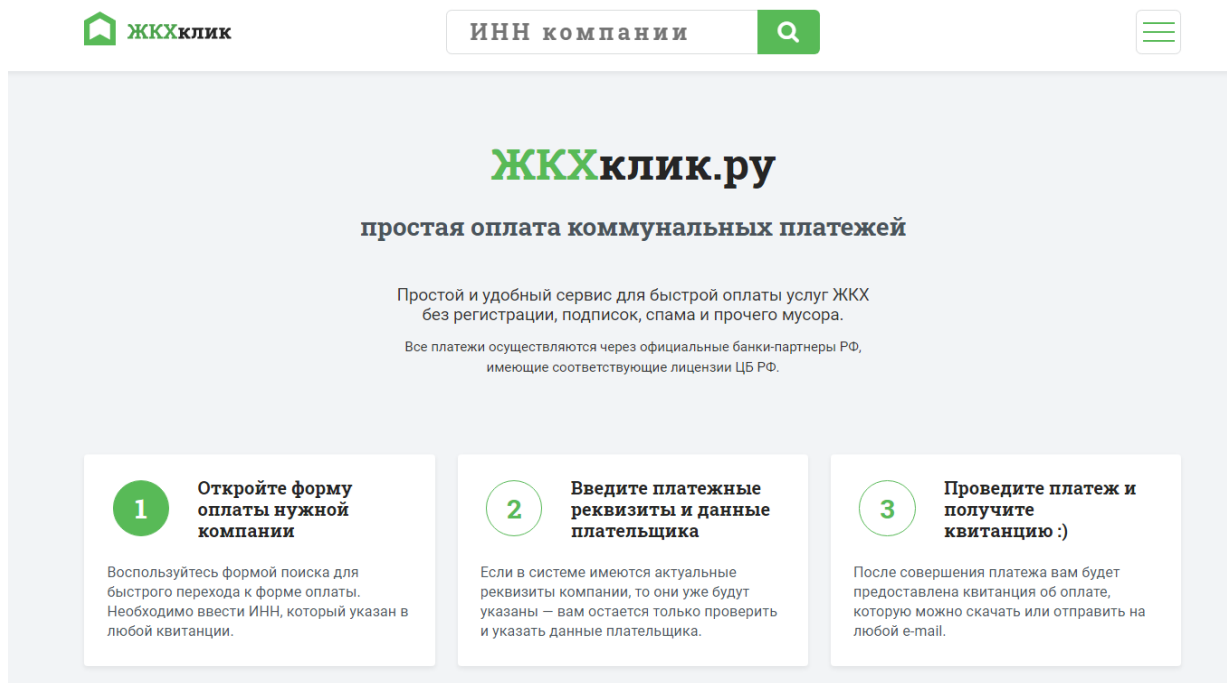


Рисунок 4 – Оплата услуг в ЖКХклик.ру

Платежный сервис а-3

Крупный сервис по предоставлению услуг по оплате ЖКХ. Из минусов можно выделить отсутствие оповещений о поступлении новых квитанций. Также в личном кабинете хотелось бы видеть статистику за предыдущие месяцы, которая отсутствует на данном сайте.

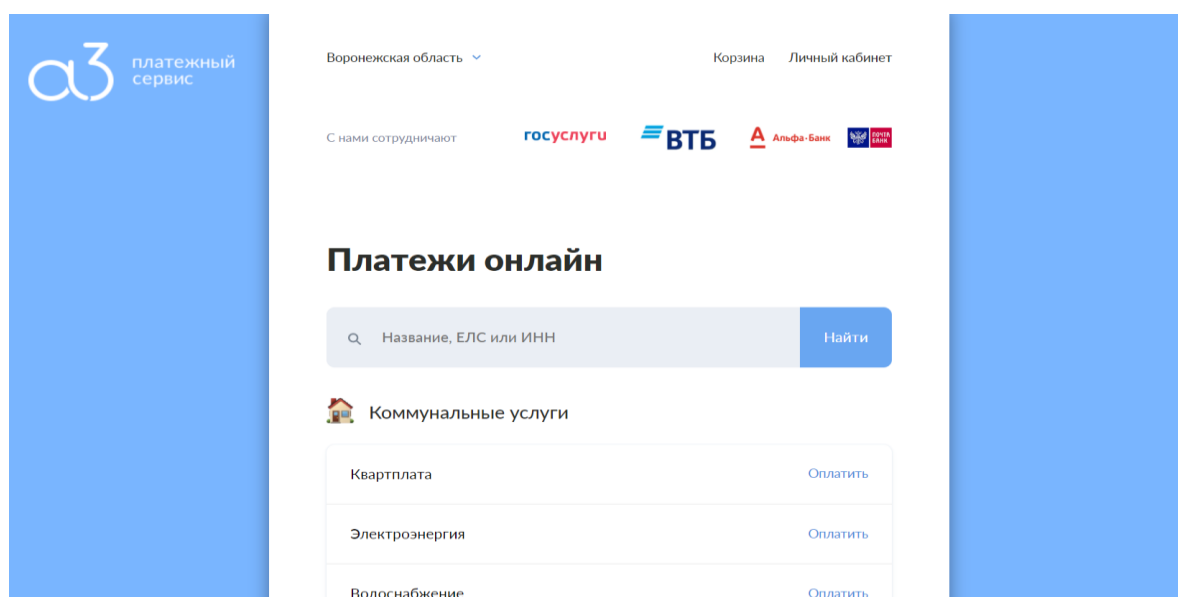


Рисунок 5 – Платежный сервис а-3

Вывод:

Среди перечисленных аналогов есть различные варианты функциональности, но особенной фишкой нашего приложения будет являться статистика, а также Телеграм-бот для оповещения и дополнительного взаимодействия с системой. Таким образом, наш веб-сервис объединит преимущества различных аналогов и избежит некоторых их недостатков.

3.2 Анализ целевой аудитории

Наше приложение предоставляет возможность оплачивать коммунальные услуги, при этом не требует регистрации и предустановки. Все необходимые функции отображаются в меню навигации, приложение не кажется перегруженным. Таким образом, оно будет удобно для использования пользователям всех возрастных групп. В данном случае, нас интересует платежеспособная совершеннолетняя часть населения, которая будет заниматься оплатой коммунальных услуг.

Итого:

- 1 Оплата коммунальных платежей через сервис «УслугиВдом», услуга рассчитана на пользователей разного возраста.
- 2 Целевая аудитория: женщины и мужчины от 18 до 80 лет.
- 3 В ЦА входят все слои населения.
- 4 Семейное положение: замужем/не замужем.
- 5 Интересы: путешествия, работа, отдых и другие.
- 6 Ядро ЦА — женщины и мужчины от 18 до 60 лет.

3.3 Пользователи системы

Для взаимодействия с разрабатываемой системой выделяют следующие типы пользователей: гость, пользователь, администратор.

Гость не предоставляет системе никаких личных данных.

Пользователь (авторизованный в системе) обладает следующими параметрами:

- фамилия, имя и отчество;
- адрес проживания;

- логин;
- пароль.

Администратор обладает следующими параметрами:

- фамилия, имя и отчество;
- логин;
- пароль.

4 Анализ задачи

4.1 Варианты использования системы

Диаграмма прецедентов представлена на Рисунке 6:

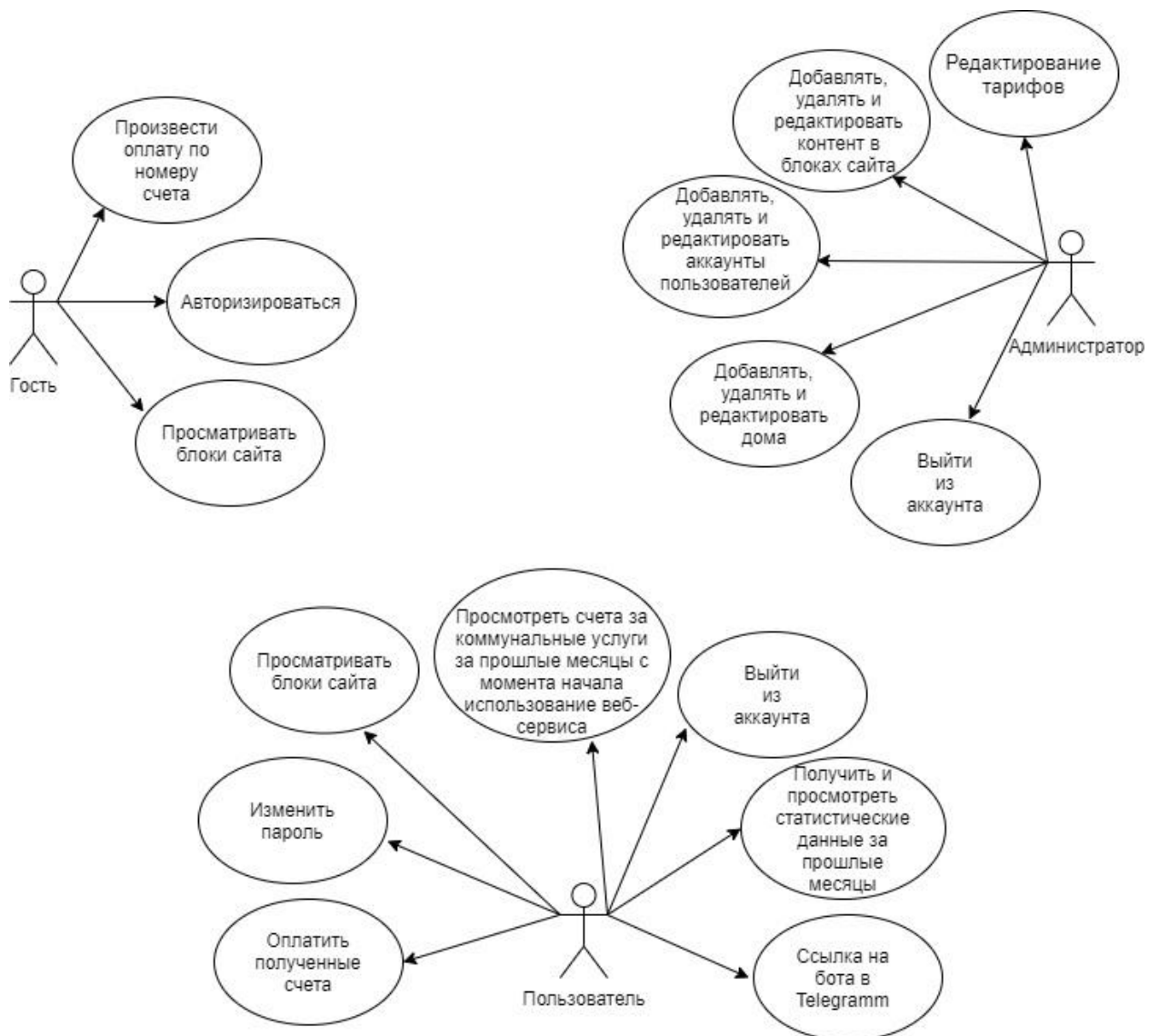


Рисунок 6 – Диаграмма прецедентов

Гость (неавторизованный в системе пользователь) имеет следующие возможности:

- возможность просмотра всех страниц сайта, кроме личного кабинета;
- возможность оплатить коммунальные услуги.

В процессе взаимодействия с системой пользователь имеет следующие возможности:

- возможность просмотра всех страниц;

- возможность редактирования личных данных.

В процессе взаимодействия с системой администратор имеет те же возможности, что и пользователь, и несколько дополнительных:

- возможность добавлять, дома и квартиры;
- возможность добавлять, удалять и редактировать пользователей;
- возможность привязывать пользователей к квартирам;
- возможность изменять номера счетов.

Опишем конкретно два сценария использования приложения.

Первый сценарий – сценарий незарегистрированного пользователя. Гость, попадая на сайт, может просматривать контент сайта и оплатить квитанцию онлайн. Для того, чтобы посмотреть неоплаченные квитанции и историю платежей, необходимо авторизоваться. Здесь часть пользователей может покинуть сайт из-за того, что их дом еще не участвует в тестировании данной системы.

Второй сценарий – сценарий зарегистрированного пользователя. Ему доступны все возможности незарегистрированного пользователя. Некоторая группа пользователей может прекратить оплату счета по причине нежелания производить оплату на сайте, так как сайт поддерживает только безналичную оплату.

4.2 Общая структура системы

Приложение представляет собой клиент-серверное приложение с панелью администрирования, что делает его поддержку нетребовательной в персонале. Состоит из трех элементов:

1. Клиентская часть веб приложения — это графический интерфейс. Графический интерфейс отображается в браузере. Пользователь взаимодействует с веб-приложением через браузер, кликая кнопкам.
2. Серверная часть веб-приложения — это программа или скрипт на сервере, обрабатывающая запросы пользователя (точнее, запросы браузера). При каждом переходе пользователя по ссылке браузер

отправляет запрос серверу. Сервер обрабатывает запрос и отправляет ответ обратно. Браузер отображает полученный результат в виде очередной веб-страницы.

3. Система управления базами данных (СУБД) - программное обеспечение на сервере, занимающееся хранением и управлением данными. Серверная часть веб-приложения обращается к базе данных, извлекая данные, которые необходимы для формирования страницы, запрошенной пользователем.

4.3 Взаимодействие компонентов системы

Рассмотрим все варианты последовательностей для пользователя (Рисунок 7) и гостя (Рисунок 8).

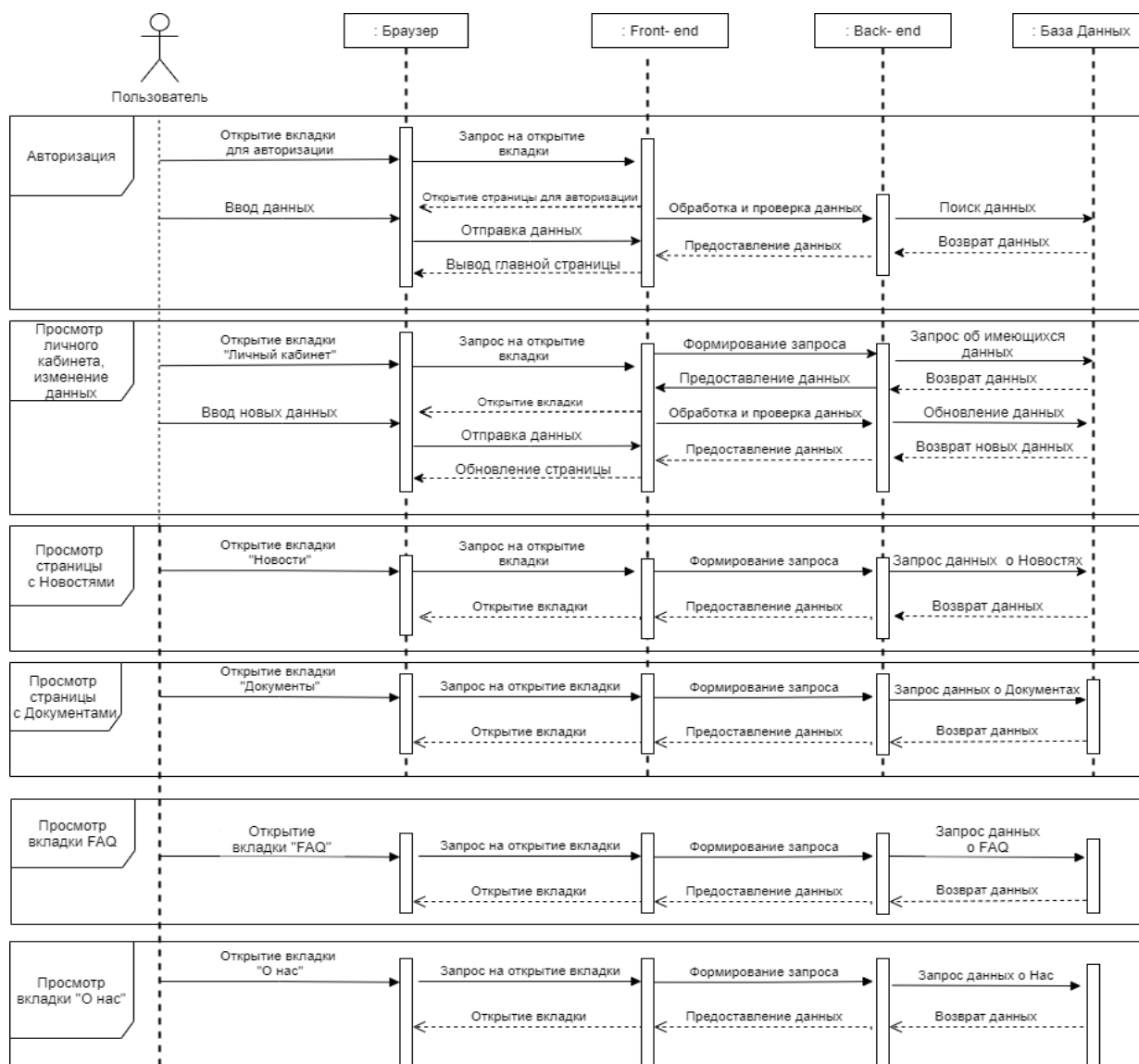


Рисунок 7 – Взаимодействие пользователя и системы

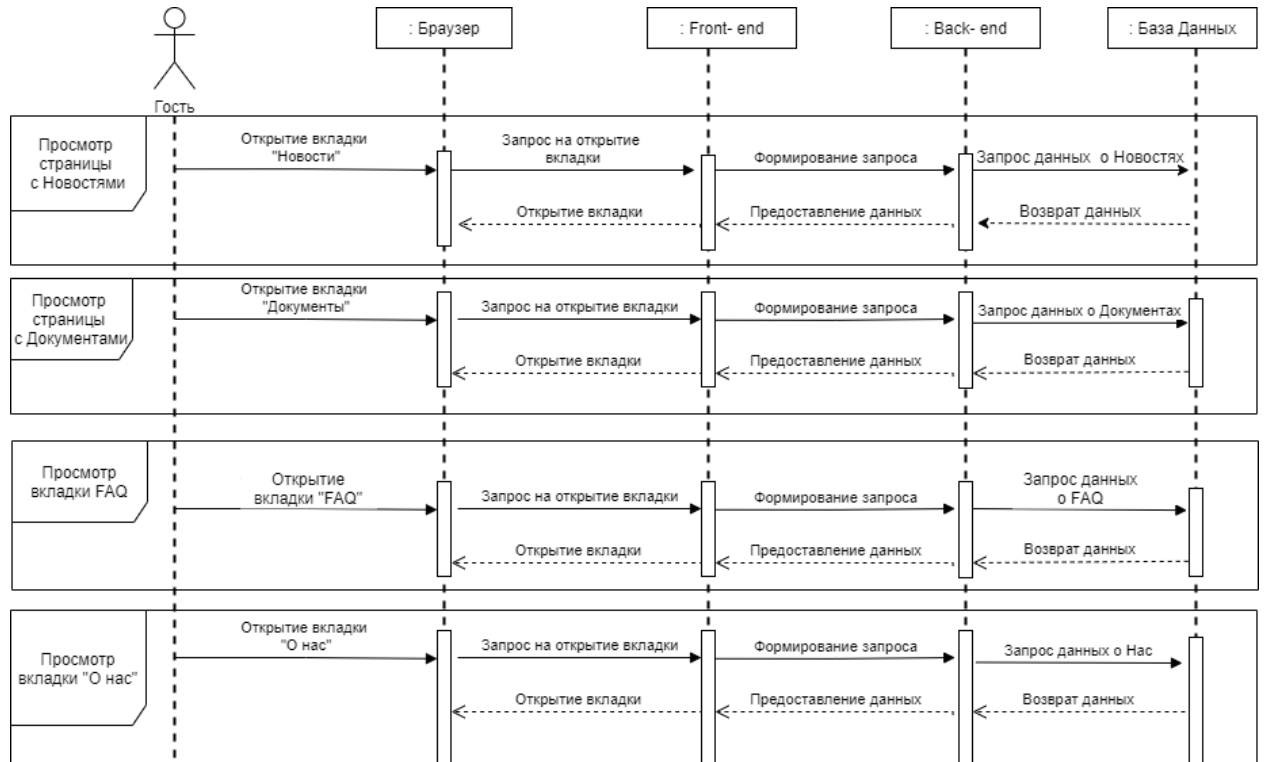


Рисунок 8 – Взаимодействие гостя и системы

4.4 Взаимодействие в системе

Ниже представленная диаграмма (Рисунок 9) дает возможность увидеть все варианты действий пользователя при оплате счёта.

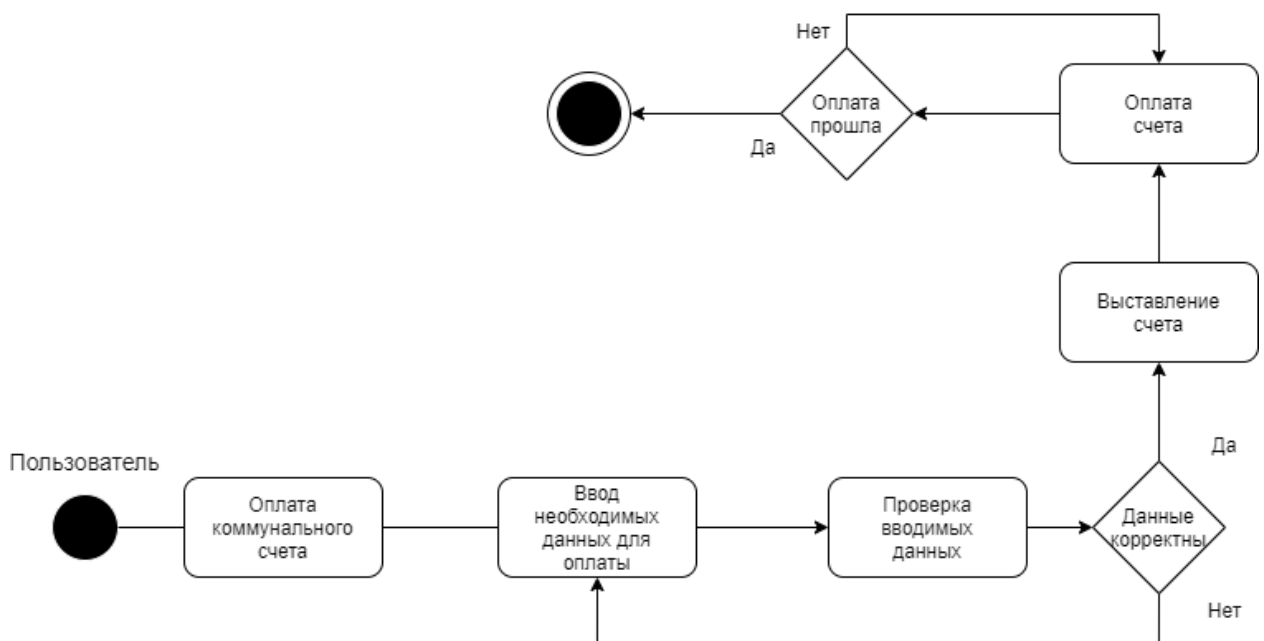


Рисунок 9 – Диаграмма активностей

4.5 Варианты состояния системы

Данная диаграмма (Рисунок10) отражает все состояния, которые может пройти гость и пользователь при просмотре всех страниц приложения. Запуск системы возможен при переходе по ссылке на сайт. Всем посетителям сайта доступны возможности гостя. Если у посетителя есть аккаунт, то необходима лишь авторизация для возможности использования всех возможностей сайта. После авторизации в системе пользователь может перейти на все страницы приложения, просматривать статистику, проверять задолженности и оплачивать их, а также оплачивать свои текущие коммунальные счета.

Авторизованный пользователь также может перейти на страницу личного кабинета, изменить личные данные и выйти из системы.

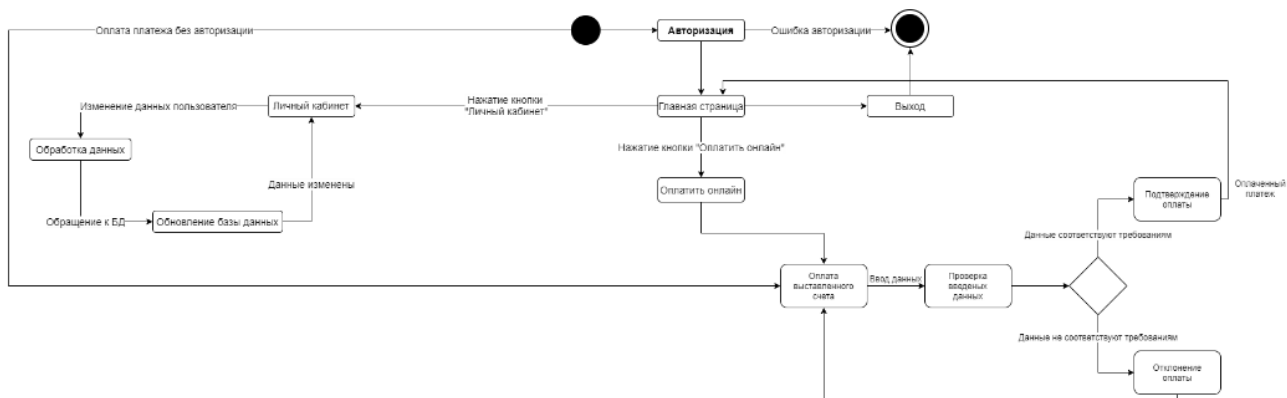


Рисунок 10 – Диаграмма состояний

4.6 Развертывание системы

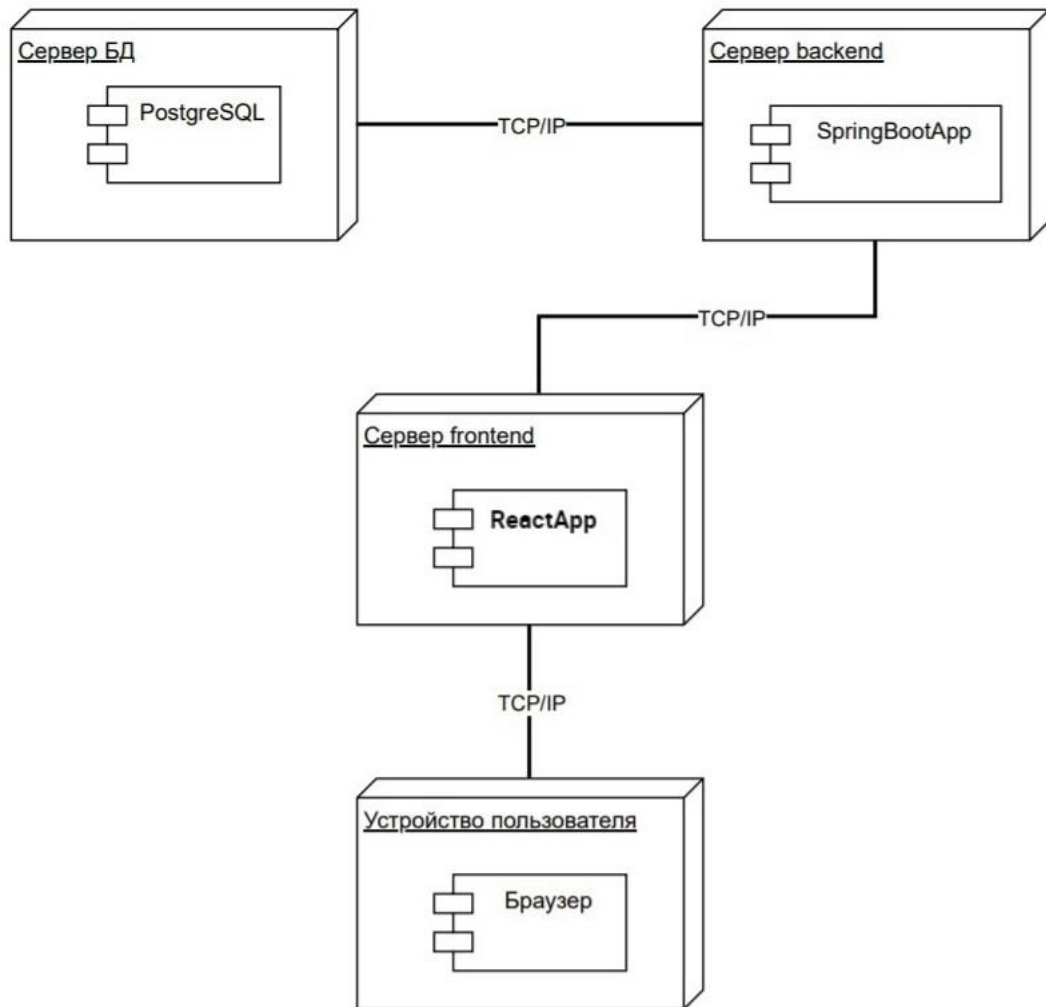


Рисунок 11 – Диаграмма развертывания

Данная диаграмма (Рисунок 11) показывает архитектуру исполнения системы, включая такие узлы, как программные среды исполнения, а также промежуточное программное обеспечение, соединяющее их.

4.7 IDEF0

Контекстная диаграмма системы функционирования сайта (или диаграмма верхнего уровня) является вершиной древовидной структуры диаграмм и показывает назначение системы (или основную функцию) и ее взаимодействие с внешней средой. Для сайта по оплате коммунальных платежей «УслугиВДом» контекстная диаграмма представлена на Рисунке 12. Главной задачей системе, как и отображено на диаграмме, является организация оплаты жилищно-коммунальных услуг онлайн.

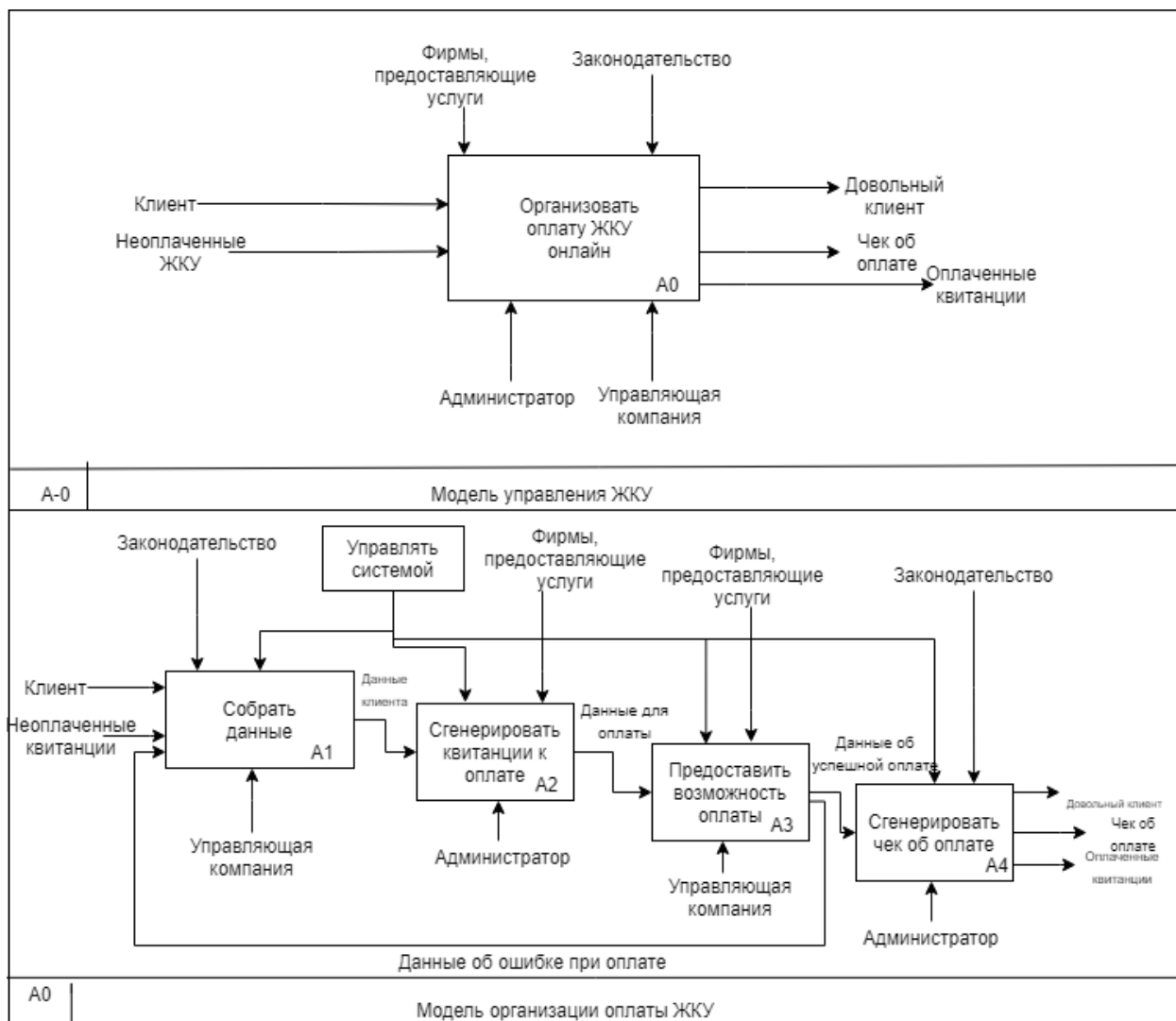


Рисунок 12 – Контекстная диаграмма

Диаграммы декомпозиции, получающиеся при разбиении контекстной диаграммы на крупные подсистемы, предназначены для детализации функций и описывают каждую подсистему и взаимодействие с другими подсистемами. Рассмотрим диаграмму декомпозиции первого уровня (Рисунок 10), демонстрирующую оплату счета клиентом.

Изначально необходимо собрать данные (сюда входят неоплаченные квитанции и ввод новых данных), за это отвечает управляющая компания и регулируется законодательством. Данные клиента поступают в блок генерации квитанций к оплате, за выставление всех необходимых тарифов на сайте и прочее ответственен администратор, данные же для них выставляют фирмы, предоставляющие услуги (водоснабжение, газоснабжение и

электричество). Далее данные для оплаты попадают в блок оплаты. За нее также ответственна управляющая компания, но регулируются фирмами услуг. Если платеж прошел успешно, то генерируется чек об оплате, если нет, то данные возвращаются обратно в блок сбора информации.

4.8 Диаграмма классов

Диаграмма классов – это структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей между ними.

Ниже приведены диаграммы классов на уровне реализации. Чтобы диаграмма не выглядела громоздко и была читаема, она была разбита на несколько частей.

Приведена диаграмма сущностей для entity (Рисунок 13).

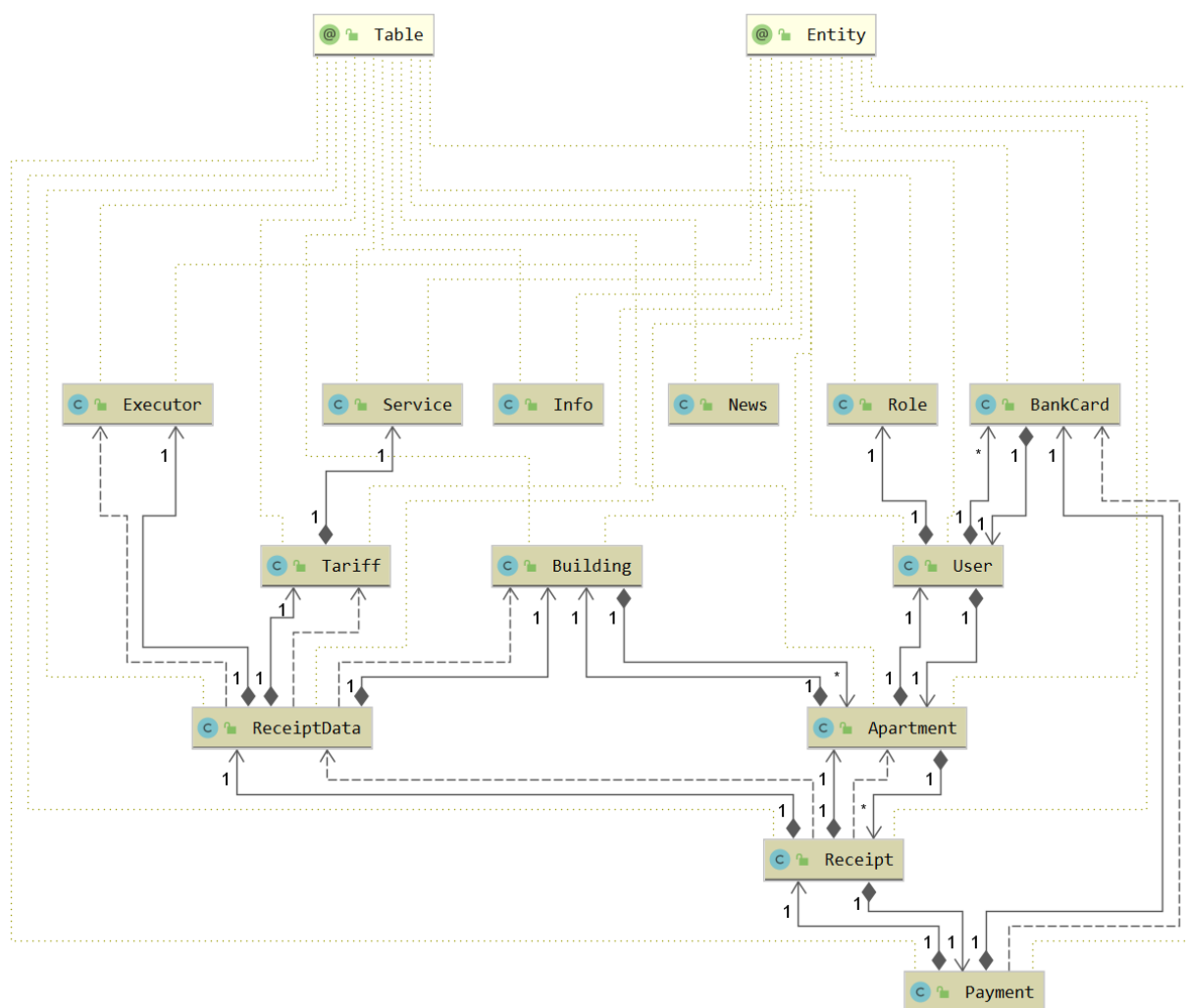


Рисунок 13 – Диаграмма классов для entity

Представлена диаграмма классов для слоя repository (Рисунок 14).

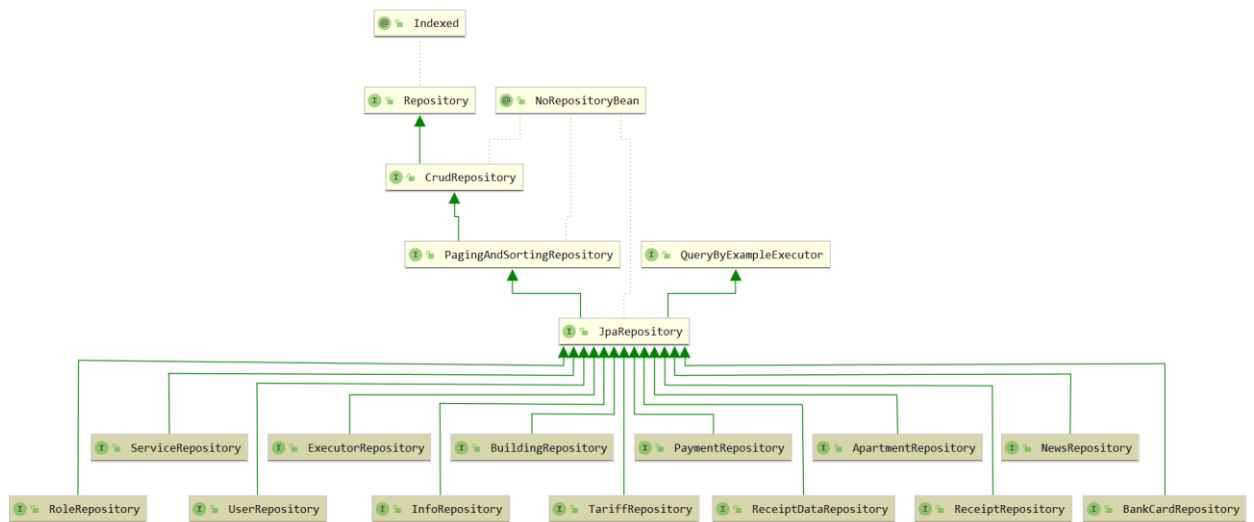


Рисунок 14 – Диаграмма классов для репозиторий

Заключительная диаграмма классов содержит сервисы (Рисунок 15).

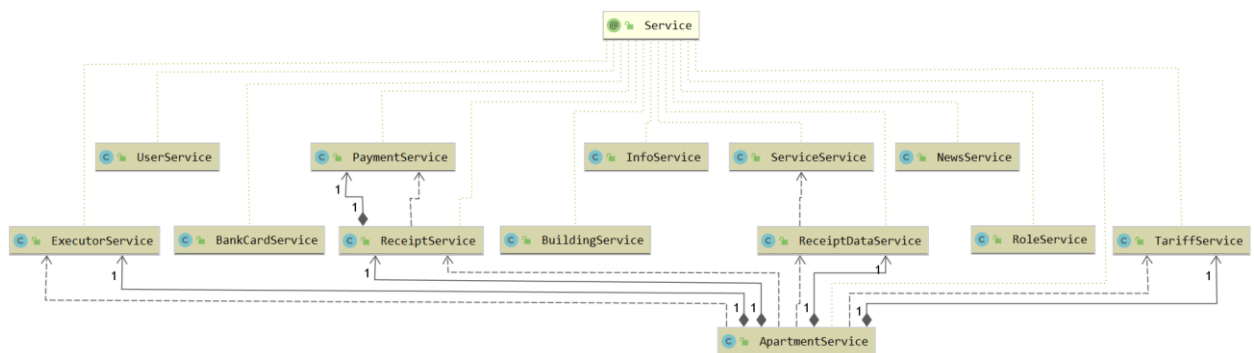


Рисунок 15 – Диаграмма классов для сервисов

4.9 ER-диаграмма

Схема ER-диаграммы базы данных (Рисунок 16) позволяет продемонстрировать, как сущности в пределах базы данных связаны между собой и какие атрибуты характеризуют каждую из них.

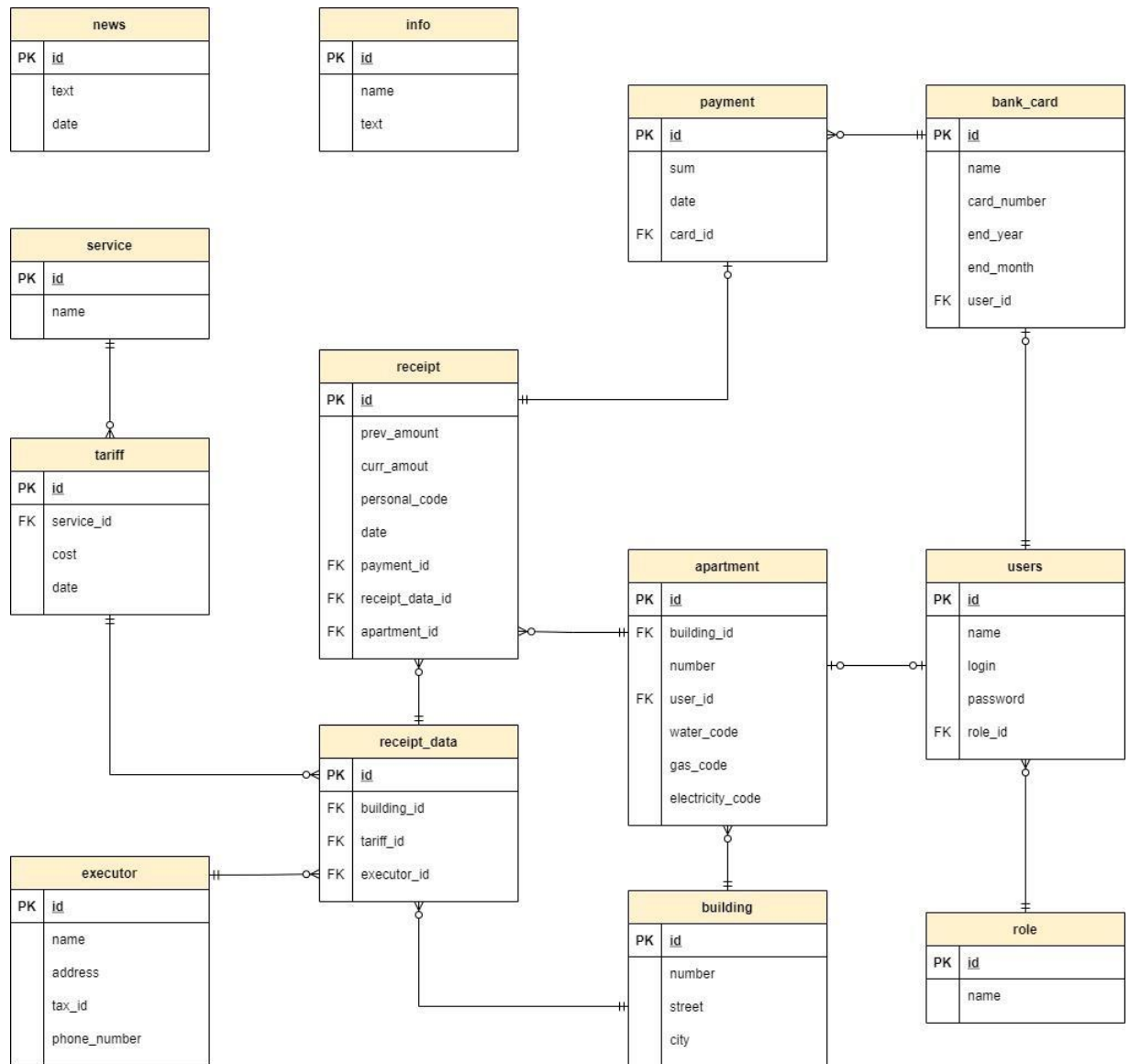


Рисунок 16 – ER-диаграмма

5 Анализ средств реализации

В качестве языка программирования был выбран Java, так как это высокоуровневый язык программирования, который обладает рядом преимуществ:

- высокая скорость разработки;
- простота масштабирования;
- структурированность приложения;
- встроенный сборщик мусора;
- наличие фреймворка Spring.

В качестве основного фреймворка был выбран Spring Boot, так как он обладает необходимыми качествами для выполнения проекта.

Spring обеспечивает слабо связанное приложение. Однако, когда есть несколько слабо связанных блоков, отслеживание их становится непростой задачей. Тут Spring Boot помогает упростить все, не предлагая никакой функции конфигурации.

- Spring Boot не требует развертывания war-файлов;
- создает автономные приложения;
- направлен на уменьшение объема исходного кода;
- имеет дополнительную функциональность «из коробки»;
- простая настройка и управление.

В качестве системы управления базой данных была выбрана PostgreSQL.

Преимущества PostgreSQL:

- поддержка БД неограниченного размера;
- мощные и надёжные механизмы транзакций;
- легкая расширяемость.

Для выполнения поставленных задач по Frontend разработке были выбраны следующие средства: ReactJS и npm, т. к. они обладают рядом преимуществ.

- высокий уровень гибкости и максимальная отзывчивость;

- виртуальная DOM (document object model), которая позволяет упорядочивать документы форматов HTML, XHTML или XML в дерево, которое лучше всего подходит веб-браузерам для анализа различных элементов веб-приложения;
- в сочетании с ES6 ReactJS может легко работать при высоких нагрузках;
- связывание данных от больших к меньшим. Это означает такой поток данных, при котором дочерние элементы не могут влиять на родительские данные;
- 100%-ая JavaScript-библиотека с открытым исходным кодом, которая получает множество ежедневных обновлений и улучшений в соответствии с отзывами разработчиков по всему миру;
- невероятно легкий вес, так как данные, которые выполняются на стороне пользователя, могут легко быть представлены на стороне сервера в то же самое время.

6 Сценарии воронок конверсии

- 1 Зашел на сайт — не авторизовался — произвел оплату.
- 2 Зашел на сайт — авторизовался — перешел в личный кабинет — произвел оплату.

7 Тестирование

После реализации всех задач был проведен запланированный набор тестов. Он включает 3 вида тестирования:

- дымовое тестирование;
- UI тесты;
- юзабилити тесты.

7.1 Дымовое тестирование

Для данного тестирования необходимо было проверить работоспособность сайта на следующих основных сценариях:

- оплата по номеру счета;
- авторизация;
- оплата выставленных коммунальных счетов;
- просмотр статистики;
- просмотр предыдущих выставленных счетов;
- просмотр вкладок «О нас», «Новости», «FAQ», «Документы»;
- изменения пароля в личном кабинете;
- добавление и удаление домов и квартир;
- добавление и удаление пользователей.

Дымовое тестирование проводилось ручным способом, в следующих браузерах: Mozilla Firefox, Google Chrome, Yandex Browser, Opera. Результаты, полученные в ходе тестирования, представлены в Таблице 1.

Таблица 1 – Результаты дымового тестирования

| Сценарий | Результат |
|---|-----------|
| Оплата по номеру счёта | Пройден |
| Авторизация | Пройден |
| Оплата выставленных коммунальных счетов | Пройден |
| Просмотр статистики | Пройден |

| | |
|---|---------|
| Просмотр предыдущих выставленных счетов | Пройден |
| Просмотр вкладок «О нас», «Новости», «FAQ», «Документы» | Пройден |
| Изменение пароля в личном кабинете | Пройден |
| Добавление и удаление домов и квартир | Пройден |
| Добавление и удаление пользователей | Пройден |

По итогу дымового тестирования было установлено, что сайт проходит все основные утвержденные сценарии.

7.2 UI тесты

В результате UI тестирования было выполнено 38 тестов, охватывающие основные возможности сайта.

Таблица 2 – Результаты UI тестирования

| Шаги теста | Ожидаемый результат | Статус |
|--|--|---------|
| 1. Нажимается кнопка «О нас» на главной странице | Открылась страница с описанием проекта | Пройден |
| 1. Нажимается кнопка «Новости» на главной странице | Открылась страница со списком новостей | Пройден |
| 1. Нажимается кнопка «FAQ» | Открылась страница с часто задаваемыми вопросами | Пройден |
| 1. Нажимается кнопка «Документы» | Открылась страница со списком документов | Пройден |

| | | |
|--|--|---------|
| 1. Нажимается кнопка «Домой» (изображение дома в левом углу) | Открылась домашняя страница, соответствующая роли пользователя | Пройден |
| 1. Нажимается кнопка «Оплатить онлайн» 2. Нажимается одна из трёх представленных кнопок 3. Вводятся корректные данные. 4. Нажимается кнопка «Далее» | Открылась страница с вводом данных платежа | Пройден |
| 1. Нажимается кнопка «Оплатить онлайн» 2. Нажимается одна из трёх представленных кнопок 3. Вводятся некорректные данные. 4. Нажимается кнопка «Далее» | Открылась страница с выбором оплачиваемой услуги | Пройден |
| 1. Вводятся корректные данные в поля «Последние показания» и «Текущие показания» 2. Нажимается кнопка «Далее» | Открылась страница с информацией об оплате | Пройден |
| 1. Вводятся некорректные данные в поля | Открылась страница с вводом данных платежа | Пройден |

| | | |
|---|---|---------|
| «Последние показания» и «Текущие показания» 2. Нажимается кнопка «Далее» | | |
| 1. Просматривается информация о сумме к оплате. 2. Нажимается кнопка «Далее» | Открылась страница с «Оплатой» | Пройден |
| 1. Нажимается кнопка «PayPal» | Открылась страница оплаты | Пройден |
| 1. Нажимается кнопка «Вход» 2. В поля вводятся корректные данные 3. Нажимается кнопка «Войти» | 1. Открылась страница с авторизацией 2. Открылась страница личного кабинета | Пройден |
| 1. Нажимается кнопка «Вход» 2. В поля вводятся некорректные данные 3. Нажимается кнопка «Войти» | 1. Открылась страница с авторизацией 2. Появилось сообщение о том, что пользователя с такими данными не существует | Пройден |
| 1. Нажимается кнопка «Вход» 2. Не во все поля вводятся данные 3. Нажимается кнопка «Войти» | 1. Открылась страница с авторизацией | Пройден |

| | | |
|---|---|---------|
| 1. Нажимается кнопка «Оплатить квитанцию» | Открылась страница выбора оплачиваемой услуги | Пройден |
| 1. Действия по оплате пользователя, который зашел в свой аккаунт те же самые, что и гостя, которые описаны выше. Разница только в том, что в первом случае в поле «личного счета» его номер автоматически подтянется из БД. | Открылась страница ввода данных платежа | Пройден |
| 1. Нажимается кнопка «Личная информация» | Открылась страница личного кабинета | Пройден |
| 1. Нажимается кнопка «Выход» | Открылась страница главной страницы | Пройден |
| 1. Нажимается кнопка «Сменить пароль» 2. В поля вводятся корректные данные 3. Нажимается кнопка «Изменить» | 1. Открылась страница с редактированием данных 2. Открылась страница личного кабинета с сохраненными измененными данными | Пройден |
| 1. Нажимается кнопка «Сменить пароль» 2. В одно или несколько полей вводятся некорректные данные или не во все поля вводятся данные | 1. Открылась страница с редактированием данных 2. Появилось сообщение о том, что старый пароль неверный | Пройден |

| | | |
|---|--|---------|
| 3. Нажимается кнопка «Изменить» | | |
| 1. Нажимается кнопка «Телеграм бот» | Выводится окно со ссылкой | Пройден |
| 1. Нажимается кнопка «История платежей» | Открылась страница с историей платежей | Пройден |
| 1. Нажимается кнопка «Задолженности» | Открылась страница с информацией о задолженностях | Пройден |
| 1. Вводятся корректные данные входа для администратора | Открылась страница с личным кабинетом администратора | Пройден |
| 1. Нажимается кнопка «Дома, квартиры» | Открылась страница с адресом и информацией номеров счетов клиентов, а также их логином | Пройден |
| 1. Нажимается кнопка «Дома, квартиры». 2. Нажимается кнопка «Удалить» напротив записи. | Страница перегружается, предыдущих данных нет | Пройден |
| 1. Нажимается кнопка «Дома, квартиры». 2. Нажимается кнопка «Изменить» | Открылось модальное окно, где необходимо ввести данные | Пройден |
| 1. Нажимается кнопка «Дома, квартиры». | Изменения сохранены, модальное окно закрылось. | Пройден |

| | | |
|---|--|---------|
| <p>2. Нажимается кнопка «Изменить».</p> <p>3. Вводятся корректные данные</p> <p>4. Нажимается кнопка «Сохранить»</p> | | |
| <p>1. Нажимается кнопка «Дома, квартиры».</p> <p>2. Нажимается кнопка «Добавить»</p> <p>3. В поля вводятся корректные данные</p> <p>4. Нажимается кнопка «Добавить»</p> | Открылась модальное окно, где необходимо ввести данные | Пройден |
| <p>1. Нажимается кнопка «Дома, квартиры».</p> <p>2. Нажимается кнопка «Добавить» в поле «логин жителя»</p> <p>3. Вводятся корректные данные</p> | Открылось модальное окно | Пройден |
| <p>1. Нажимается кнопка «Пользователи»</p> | Открылась страница с пользователями | Пройден |
| <p>1. Нажимается кнопка «Пользователи»</p> <p>2. Нажимается кнопка «Изменить»</p> <p>3. Вводятся корректные данные.</p> | Открылось модальное окно | Пройден |

| | | |
|--|---|---------|
| 4.Нажимается кнопка «Изменить» | | |
| 1. Нажимается кнопка «Пользователи» 2. Нажимается кнопка «Добавить» 3. Вводятся корректные данные 4. Нажимается кнопка «Добавить» | Открылось модальное окно | Пройден |
| 1. Нажимается кнопка «Тарифы» | Открылась страница с тарифами | Пройден |
| 1. Нажимается кнопка «Тарифы» 2. Нажимается кнопка «Изменить» 3. Вводятся корректные данные | Открылось модальное окно | Пройден |
| 1. Нажимается кнопка «Тарифы» 2. Нажимается кнопка «Удалить» | Данные данного тарифа удалились | Пройден |
| 1. Нажимается кнопка «Информация и статистика» | Открылась страница с добавлением информации | Пройден |
| 1. Нажимается кнопка «Информация и статистика» | Текст сохраняется в выбранной вкладке на странице | Пройден |

| | | |
|--|--|--|
| 2. Нажимается одна из четырех представленных кнопок. | | |
| 3. Вводится нужный текст | | |
| 4. Нажимается кнопка «Сохранить» | | |

7.3 Юзабилити тесты

Для проведения юзабилити тестирования было отобрано три случайных человека, не пользовавшиеся заранее сайтом. Для данного тестирования необходимо проверить следующие основные сценарии взаимодействия пользователя с сайтом:

- авторизация;
- оплата счета без авторизации;
- оплата счёта будучи авторизованным;
- просмотр вкладки «О нас»;
- просмотр вкладки «Новости»;
- просмотр вкладки «FAQ»;
- просмотр вкладки «Документы»;
- просмотр задолженностей;
- просмотр истории платежей;
- редактирование информации в личном кабинете;
- возврат на главную страницу;
- переход к боту в «Telegram»;
- выход из аккаунта.

Результаты тестирования отображены в Таблице 3.

Таблица 3 – Результаты юзабилити тестирования

| Сценарий | Пользователь 1 | Пользователь 2 | Пользователь 3 |
|-------------|----------------|----------------|----------------|
| Авторизация | Пройден | Пройден | Пройден |

| | | | |
|--------------------------------------|---------|---------|---------|
| Оплата счета без авторизации | Пройден | Пройден | Пройден |
| Оплата счёта будучи авторизованным | Пройден | Пройден | Пройден |
| Просмотр вкладки «О нас» | Пройден | Пройден | Пройден |
| Просмотр вкладки «Новости» | Пройден | Пройден | Пройден |
| Просмотр вкладки «FAQ» | Пройден | Пройден | Пройден |
| Просмотр вкладки «Документы» | Пройден | Пройден | Пройден |
| Просмотр задолженностей | Пройден | Пройден | Пройден |
| Просмотр истории платежей | Пройден | Пройден | Пройден |
| Изменить пароль в личном кабинете | Пройден | Пройден | Пройден |
| Возврат на главную страницу | Пройден | Пройден | Пройден |
| Переход к Телеграм боту в «Telegram» | Пройден | Пройден | Пройден |
| Выход из аккаунта | Пройден | Пройден | Пройден |

8. Используемые технологии

Разберём технологии используемые в Backend части приложения.

Серверная часть веб-приложения написана на языке Java. Для упрощения конфигурации и взаимодействия с веб-сервером в приложении использовался фреймворк Spring Boot. Для построения системы аутентификации и авторизации, а также для обеспечения безопасности был использован фреймворк Spring Security.

Слой доступа к данным и взаимодействия с базой данной реализован с помощью фреймворка Spring Data JPA. Для управления схемой базы данных используется библиотека Liquibase. Для сокращения количества однотипного кода в Entity и DTO классах была использована библиотека Lombok. Для автоматизации сборки проекта использован Apache Maven. Для формирования PDF чеков используется библиотека iText.

В качестве реляционной СУБД использовалась PostgreSQL.

Теперь перейдём к технологиям, которые были использованы во Frontend части системы.

Первое — это JavaScript библиотека ReactJS. Этот фреймворк отлично подходит для создания веб-приложений, где данные могут меняться на регулярной основе, что в данном случае актуально.

Также используется Node Package Manager — менеджер пакетов, входящий в состав Node.js.

Для создания метрик была использована Яндекс.Метрика. Для внедрения счетчиков был использован модуль react-yandex-metrica. Яндекс.Метрика дает широкий набор отслеживаемых параметров, благодаря чему можно более точно отслеживать действия пользователей.

Для создания бота был использован язык Python и библиотека telegram-bot. Данная библиотека предоставляет широкую функциональность для создания различных видов ботов, в том числе для необходимой нам отправки документов. Получение данных от сервера происходит с помощью библиотеки requests.

Заключение

В результате работы был проведен анализ предметной области, разработано веб-приложение, которое помогает пользователю произвести оплату коммунальных услуг. Были выполнены следующие задачи:

1. Разработана Frontend часть сайта, развернутая на удаленном сервере.
2. Разработана Backend часть сайта, развернутая на удаленном сервере.
3. Была создана связь между Frontend и Backend частями приложения.
4. Разработана база данных, развернутая на удаленном сервере.
5. Разработан Telegram-бот, позволяющий получить актуальные новости и квитанции по платежам в формате PDF.
6. Подключена Яндекс.Метрика, позволяющая фиксировать активность пользователей.

Приложение отвечает всем заявленным требованиям.

Список использованных источников

1. Using Liquibase with PostgreSQL – URL: <https://docs.liquibase.com/workflows/database-setup-tutorials/postgresql.html> (дата обращения: 05.03.2021)
2. Документация ReactJS – URL: <https://ru.reactjs.org/docs> (дата обращения: 15.04.2021)
3. Craig Walls. Spring in Action (5th Edition) / Craig Walls, 2018.
4. Фаулер М. UML. Основы, 3 е издание. / Фаулер М. – Пер. с англ. – СПб: Символ Плюс, 2005. – 192 с.
5. Информационный ресурс Baeldung. – URL: <https://www.baeldung.com> (дата обращения 20.04.2021)