

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Курсовой проект

Система оплаты услуг ЖКХ «УслугиВдом»

09.03.04 Программная инженерия

Информационные системы и сетевые технологии

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
Постановка задачи	4
Глоссарий	5
Анализ предметной области	7
Анализ существующих решений	7
Анализ целевой аудитории	10
Пользователи системы	11
Анализ задачи	12
Варианты использования системы	12
Общая структура системы	14
Взаимодействие компонентов системы	15
Взаимодействие в системе	16
Варианты состояния системы	17
Развертывание системы	18
IDEF0	19
Диаграмма классов	21
ER-диаграмма	22
Анализ средств реализации	23
Воронки конверсии	23
Реализация	23
Тестирование	23
Юзабилити тестирование	23

## **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 чеков об оплате и информации о задолженностях 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 Front-end части сервиса;
3. Разработка Back-end части сервиса;
4. Создание связи между Front-end и Back-end частями приложения;
5. Разработка базы данных.

## 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.

Представление данных пользователю, 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.

Front-end - клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

Back-end - 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.

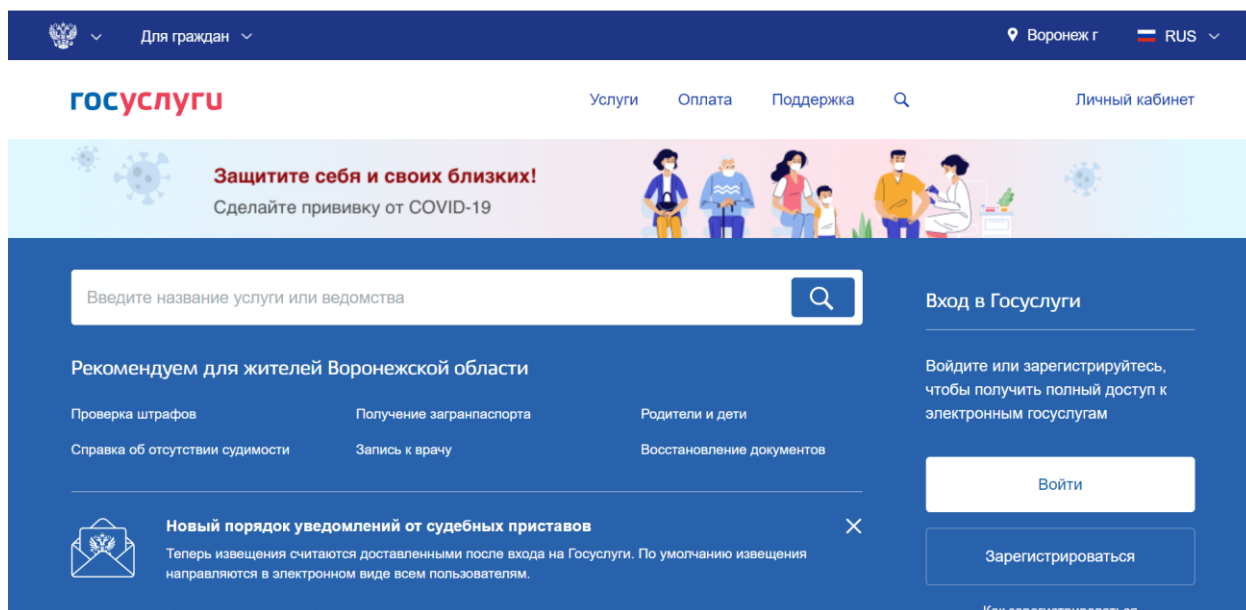


Рисунок 1 – Домашняя страница сервиса Госуслуги

##### 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.

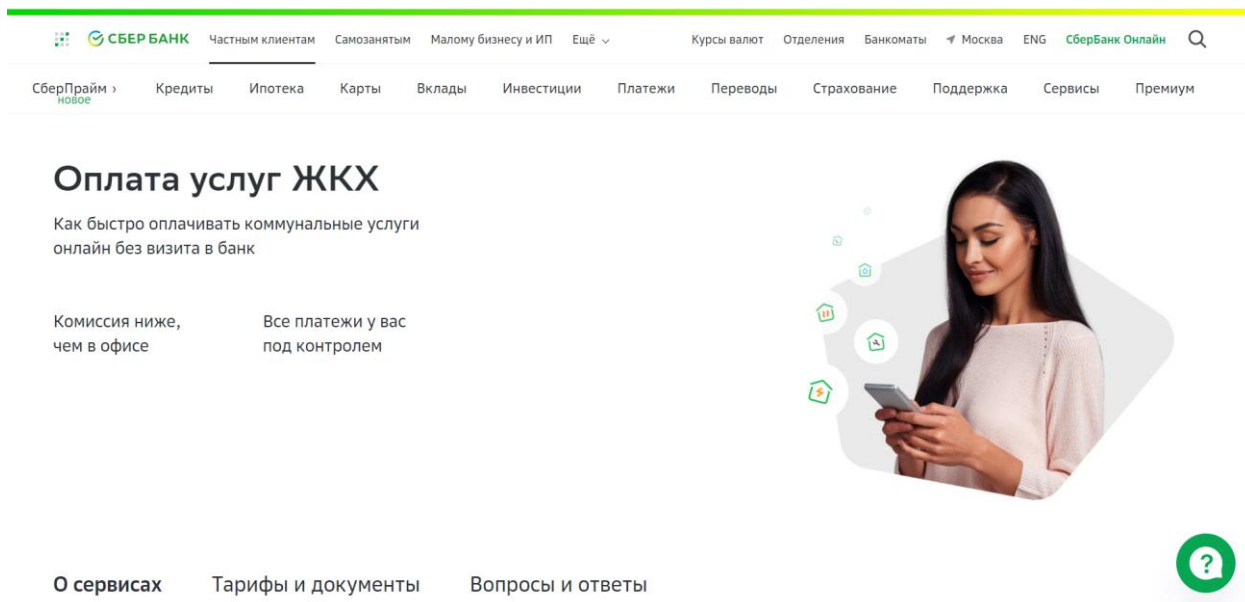


Рисунок 2 – Оплата услуг ЖКХ Сбербанка

### 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.

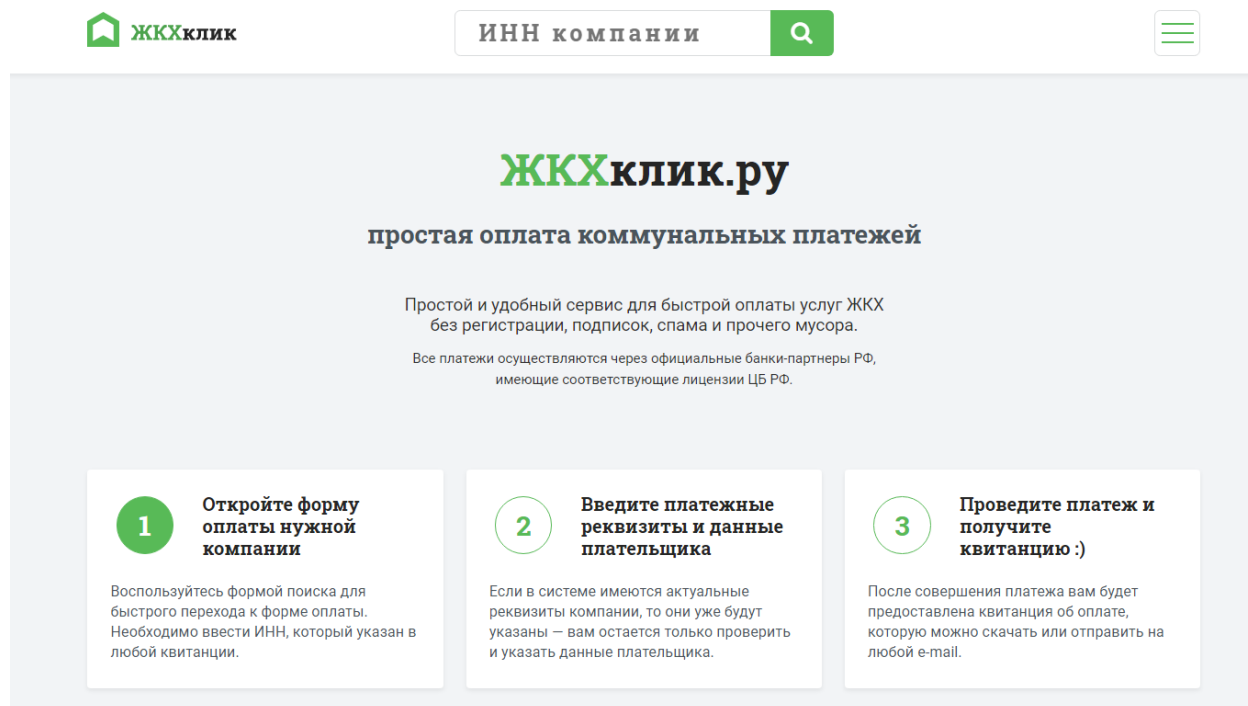


Рисунок 4 – Оплата услуг в ЖКХклик.ру

### 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.

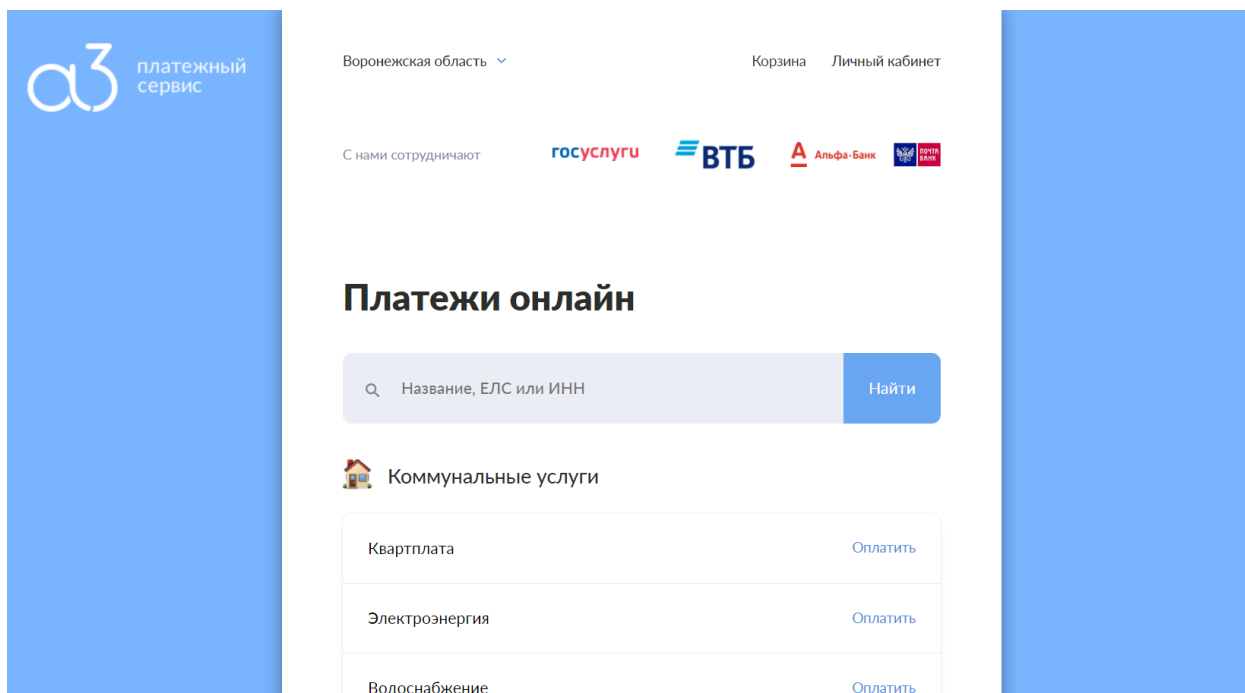


Рисунок 5 – Платежный сервис а-3

Вывод:

Среди перечисленных аналогов есть различные варианты функциональности, но особенной фишкой нашего приложения будет являться статистика, а также Телеграмм-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 покупателей 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;

Администратор от пользователя отличается содержанием параметра Роль.

## 4 Анализ задачи

### 4.1 Варианты использования системы

Диаграмма прецедентов представлена на Рис. 6:

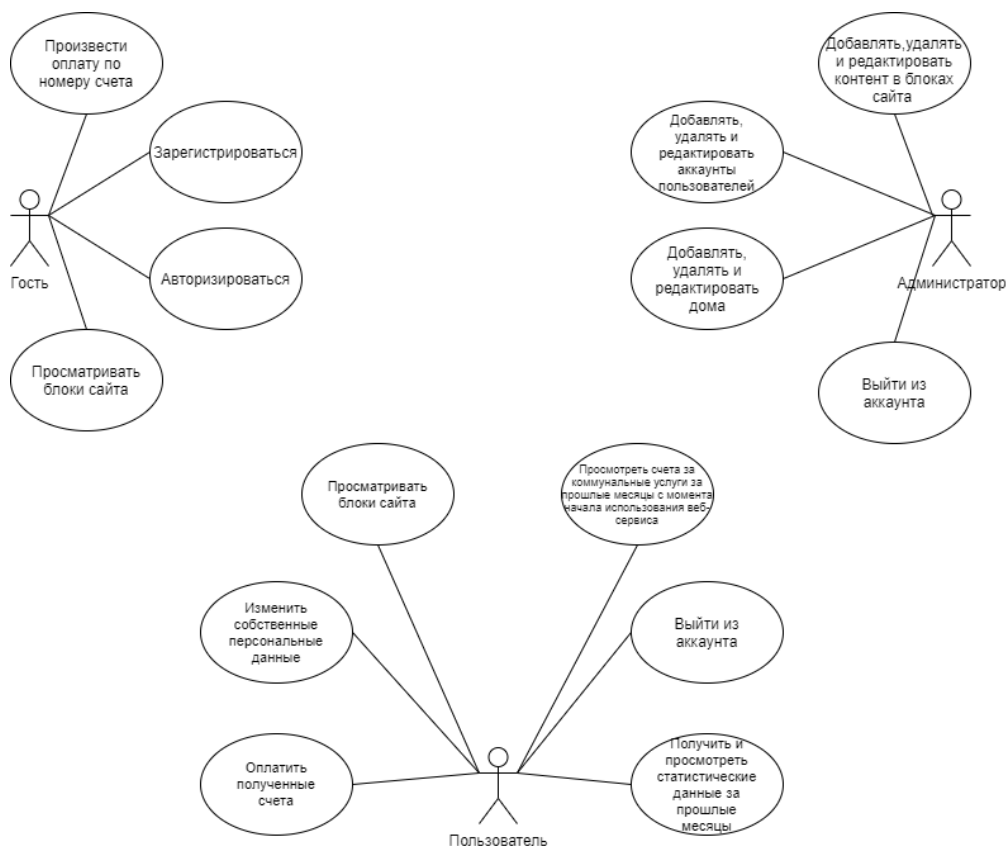


Рисунок 6 - Диаграмма прецедентов

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;

- Возможность добавления товаров в корзину.

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

- Возможность добавлять, изменять и удалять пользователей;
- Возможность добавлять дома и квартиры в них;
- Возможность изменять номера счетов.

Опишем конкретно 2 сценария использования приложения.

Первый сценарий – сценарий незарегистрированного пользователя. Гость, попадая на сайт, может просматривать контент сайта и оплатить квитанцию онлайн. Для того, посмотреть неоплаченные квитанции и историю платежей, необходимо авторизоваться. Здесь часть пользователей может покинуть сайт из-за того, что их дом ещё 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 той причине, что не желает производить оплату на сайте (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 классических трех элементов:

- Клиентская часть веб приложения - это графический интерфейс. Графический интерфейс отображается в браузере. Пользователь взаимодействует с веб-приложением именно через браузер, кликая по ссылкам и кнопкам.
- 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 к серверу. Сервер обрабатывает запрос и отправляет ответ обратно. Браузер тут же отображает полученный результат в виде очередной веб-страницы.
- База данных (БД, или система управления базами данных, СУБД) - программное обеспечение на сервере, занимающееся хранением данных и их выдачей в нужный момент. В данном случае, в базе данных хранятся пользователи, данные о домах, квартирах, тарифах и так далее. 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 (Рис. 7) и гостя (Рис.8)

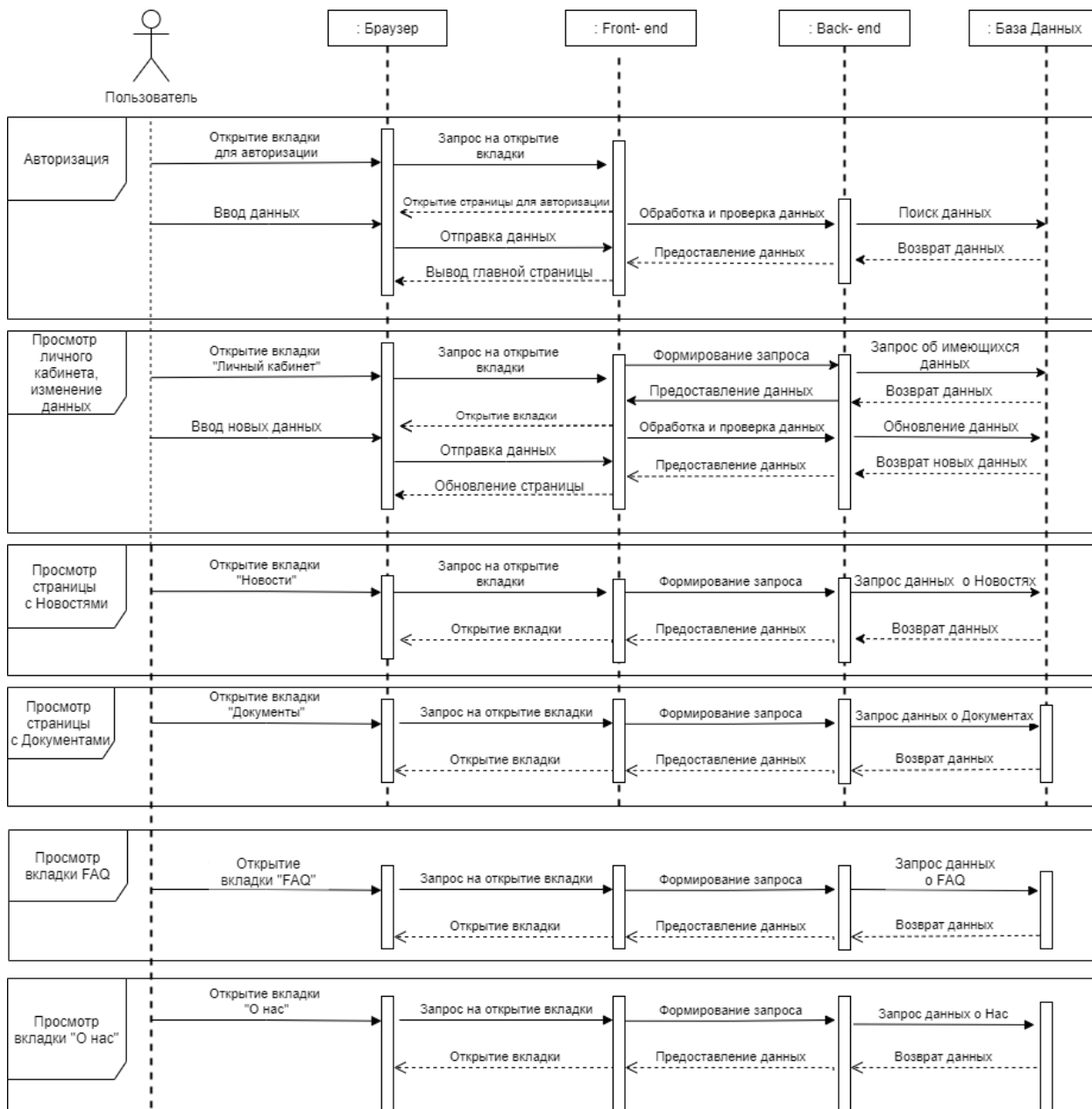


Рисунок 7 - Взаимодействие пользователя и системы

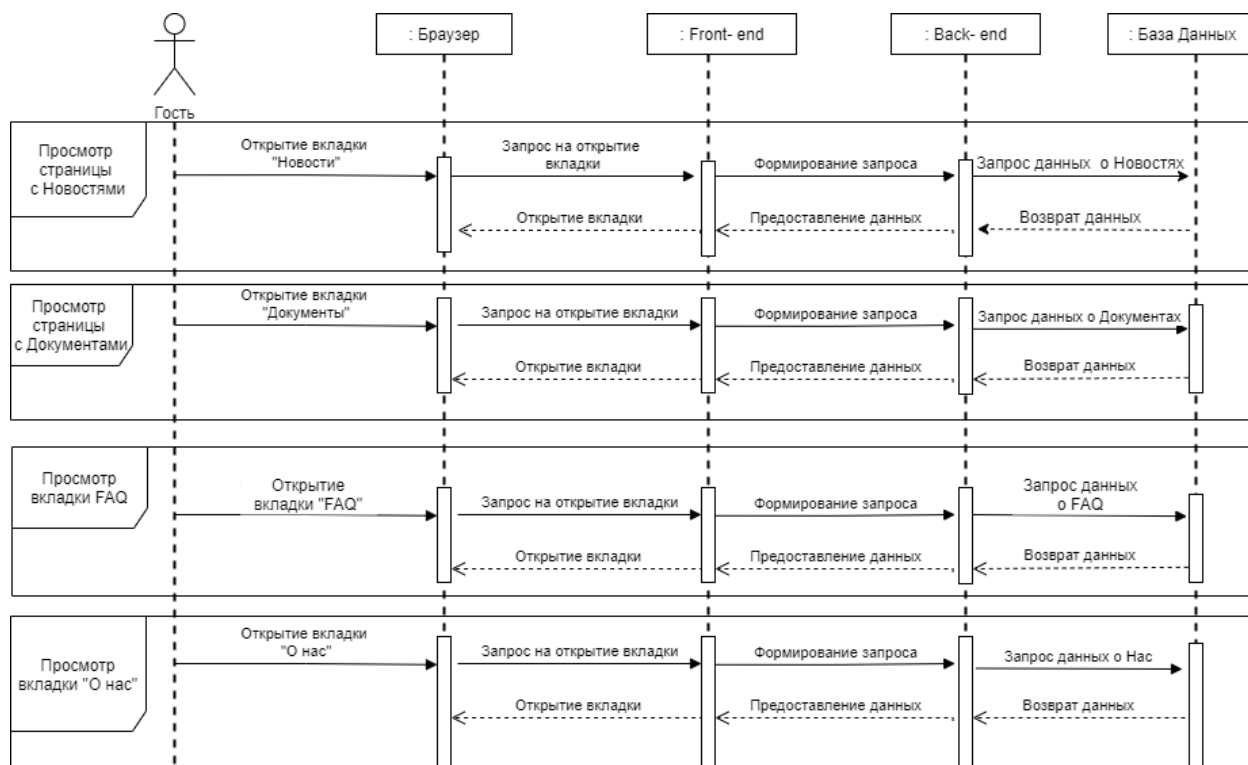


Рисунок 8 - Взаимодействие гостя и системы

#### 4.4 Взаимодействие в системе

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

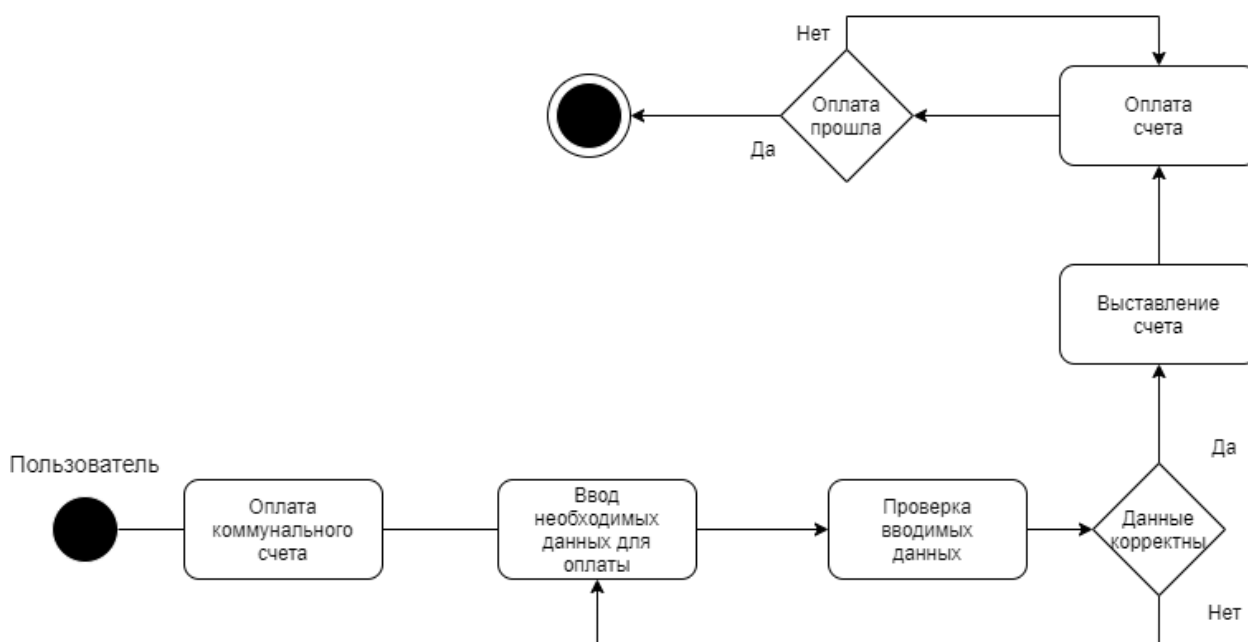


Рисунок 9 - Диаграмма активностей оформления заказа



## 4.5 Варианты состояния системы

Данная диаграмма (Рис.10) отражает все состояния, которые может пройти гость и пользователь при просмотре всех страниц приложения и оформлении заказа.

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.

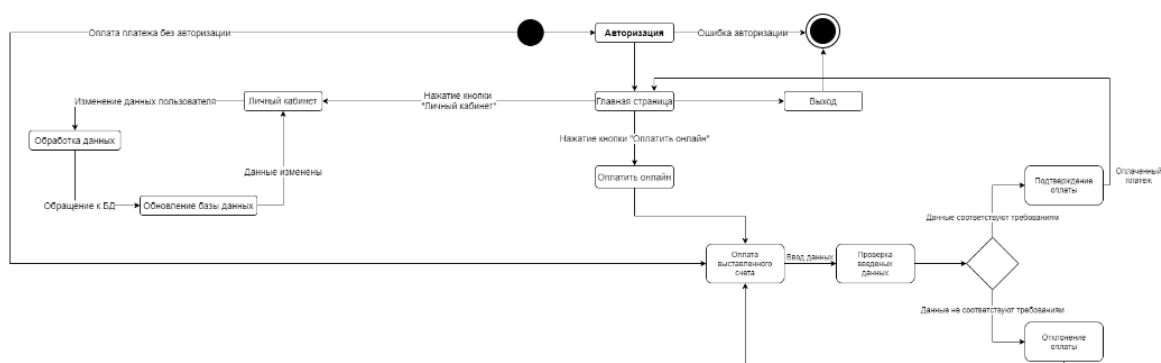


Рисунок 10 - Диаграмма состояний

## 4.6 Развертывание системы

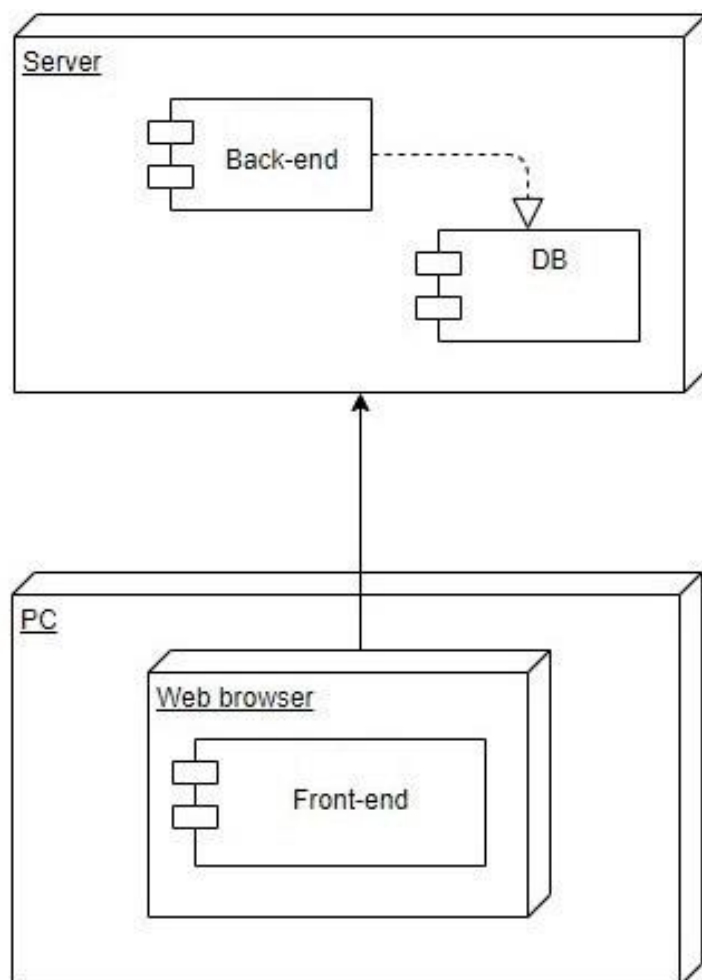


Рисунок 11 - Диаграмма развёртывания

Данная диаграмма (Рис.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 «Услуги В Дом» контекстная диаграмма представлена на Рисунке 12. Главной задачей системы, как и отображено на диаграмме, является организация оплаты жилищно-коммунальных услуг онлайн.

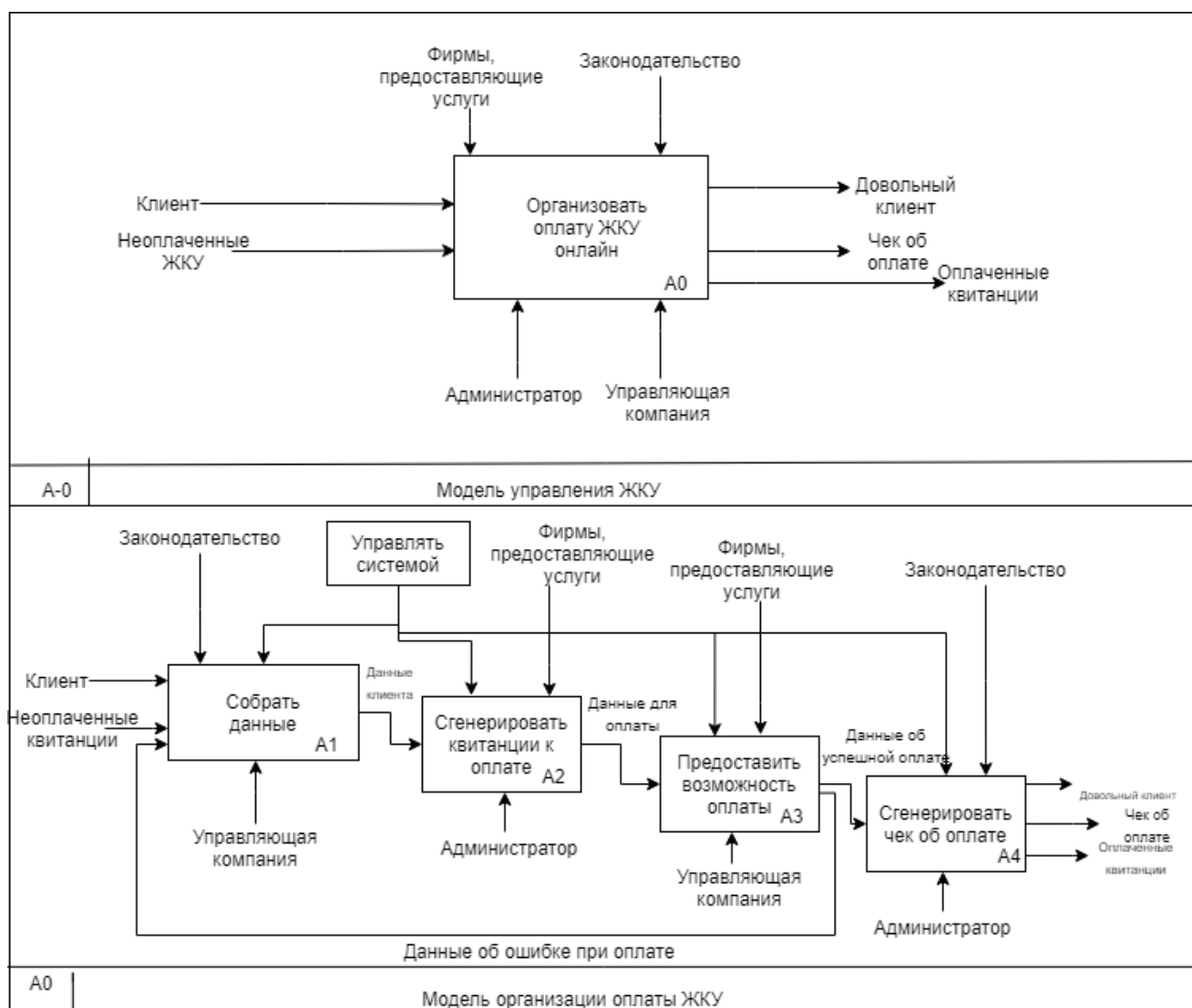


Рисунок 12 - Контекстная диаграмма

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 (Рис. 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

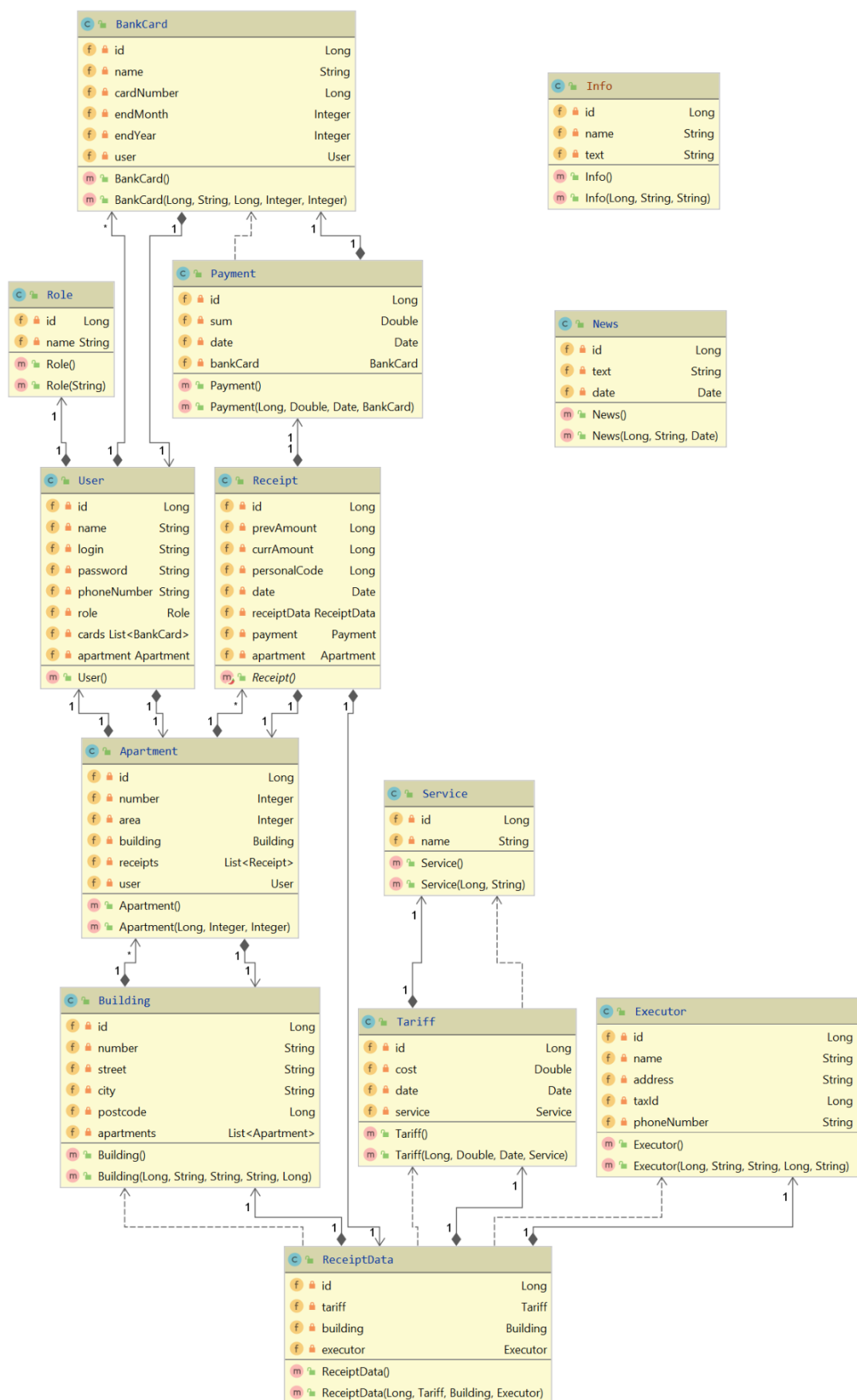


Рисунок 13 - Диаграмма классов

## 4.9 ER-диаграмма

(Рис.999) показывает, как сущности связаны между собой внутри системы.

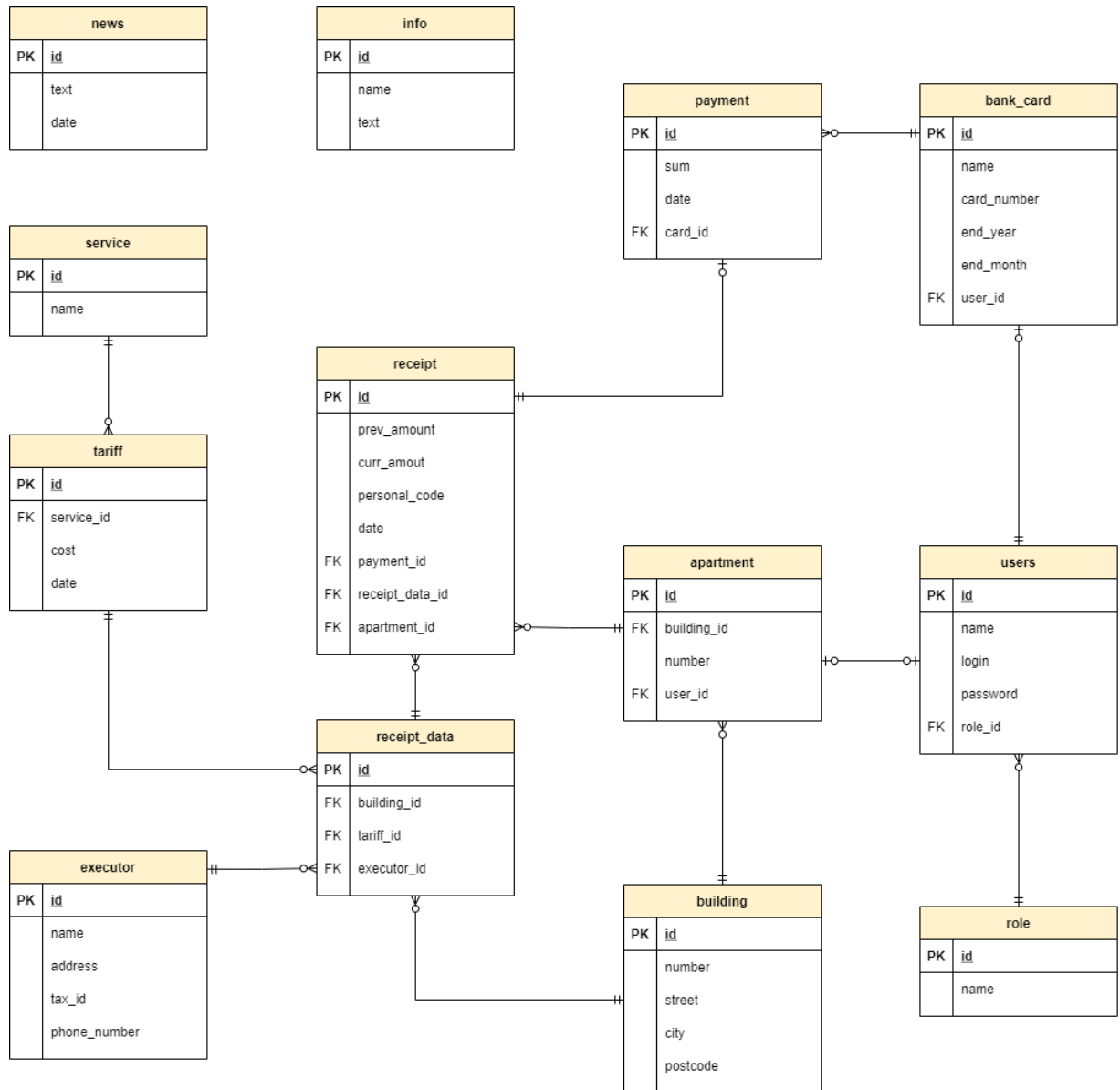


Рисунок 14 - ER-диаграмма

## **5 Анализ средств реализации**

## **6 Воронки конверсии**

## **7 Реализация**

## **8 Тестирование**

### **8.1 Юзабилити тестирование**

## **Заключение**