

**Программный комплекс «TENEMENT»**

**ОПИСАНИЕ АРХИТЕКТУРЫ И  
ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

## ОГЛАВЛЕНИЕ

ОПИСАНИЕ АРХИТЕКТУРЫ «TENEMENT»	2
Основные решения программно-технической архитектуры «TENEMENT»	2
Описание используемых программных компонентов, платформы и технологий	4
Описание клиентской и серверной части	5
Безопасность	7
ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК	8
Назначение комплекса	8
Задачи и функции	8
Выходные данные	11
Затрачиваемые ресурсы	11

## **ОПИСАНИЕ АРХИТЕКТУРЫ «TENEMENT»**

### **Основные решения программно-технической архитектуры «TENEMENT»**

#### **1) Трехуровневая архитектура «Клиент-сервер»**

Согласно функциональным требованиям к программному продукту требуется обеспечить масштабируемость, централизованность данных и высокую оперативность получения необходимой информации, и ее достоверность. Трехуровневая архитектура отвечает данным требованиям, помимо этого данная архитектура обеспечивает безопасность, хранимой в базе данных информации.

#### **2) Серверная часть**

Программный комплекс реализован на модульной платформе для создания веб-приложений Spring Framework. Spring является одним из самых популярных инструментов при разработке промышленных приложений на языке программирования Java. Также используется популярный и легковесный веб-сервер – Nginx, сервер приложения – Apache Tomcat, Hibernate - фреймворк для работы с базами данных (объектно-реляционного отображение (ORM)).

#### **3) Клиентская часть (Web-интерфейс)**

Пользовательский интерфейс реализован в форме Web-интерфейса, за основу которого взята платформа Angular в связке с фреймворком Clarity. Выбранный подход позволяет, не привязывается к операционной система устройства, обеспечивая кросс-платформенность программного продукта. В качестве минимальных требований для работы с приложением выступают требования используемого браузера пользователя.

#### **4) База данных**

Выбор СУБД основывается на поставленных задачах и простоте настройки и сопровождения БД, при возможности выполнения больших и сложных операций. Проанализировав текущий рынок СУБД, было выбрано одно из самых популярных и современных решений – PostgreSQL. Данная технология обладает необходимыми для проекта характеристиками: масштабируемость благодаря своим характеристикам с открытым исходным кодом; кластерные решения для хранения данных обеспечивают свободное расширение; поддерживает собственный SSL, который помогает шифровать обмен данными с сервером; является open-source проектом.

#### **5) Интеграция решений с открытым исходным кодом**

Использование в проекте open-source решений позволяет избежать сторонних отчислений, что уменьшает затраты на конечный продукт, но при этом упрощает и ускоряет процесс реализации, модернизации и сопровождение проекта.

Выше перечисленные решения были выбраны на основе требований к реализации проекта.

На рисунке 1 продемонстрирована схема программно-технической архитектуры, с указанием взаимодействия различных компонентов.

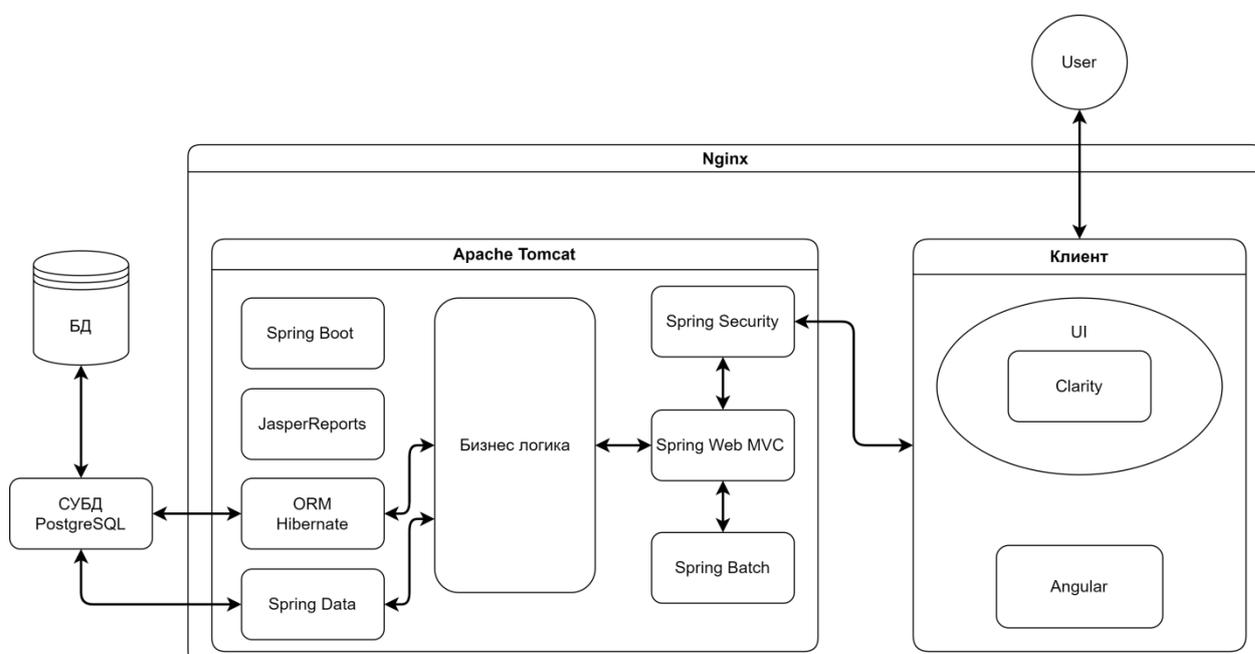


Рисунок 1 – Схема программно-технической архитектуры проекта

На представленной схеме отображены три уровня реализованной клиент-серверной архитектуры. Пользователь взаимодействует с клиентской частью приложения, которая представляет собой Web-интерфейс, в котором используются элементы фреймворка Clarity. Сам интерфейс разработан на основе рекомендаций по User experience (UX), описанным в документации фреймворка, то есть в проекте используются ключевые подходы для разработки интуитивно понятного и разборчивого интерфейса. Структура серверной части позволяет обеспечить аутентификацию для безопасного доступа к данным, а также оптимизировать процессы по логистике проекта за счет использования единого стека open-source технологий фреймворка Spring.

## Описание используемых программных компонентов, платформы и технологий

### Платформы

- Java 11

## **Хранение и обработка данных**

- PostgreSQL
- Spring Data
- Hibernate

## **Развертывание**

- Nginx
- Apache Tomcat

## **Безопасность**

- Spring Security

## **Web-интерфейс**

- Clarity
- Angular

## **Инструменты разработки (рекомендуемые)**

- Visual Studio Code

## **Описание клиентской и серверной части**

### 1) Одностраничное приложение

Данный подход позволяет осуществлять навигацию по веб-сайту, обновляя содержимое тела веб-документа, без загрузки новых документов и как следствие снижение нагрузки на сервер, повышение производительности.

## 2) Реализация CRUD-операций с данными в представлениях

Согласно требованиям к программному продукту, пользователи с определенной категорией прав, должны иметь возможность манипулировать данными. В приложении продумана и реализована логика осуществления операций для удаления, создания, редактирования и просмотра данных при помощи специальных форм заполнения.

## 3) Параллельное выполнение задач (например, запуск формирования квитанций, задачи расчета)

При помощи пакетной среды Spring Batch, реализована возможность запуска выполнения операций по формированию, обработке и манипуляции большими данными, с возможностью дальнейшей работы в комплексе, то есть без блокировки операций и интерфейса для пользователя.

## 4) Пагинация, фильтрация и поиск данных

Для обеспечения удобной работы с большими объемами данных, в приложении реализована пагинация данных, поиск и фильтрация по различным параметрам.

## 5) Валидация введенных данных в формах операций

Для обеспечения безопасности некорректных операций над данными используется трехуровневая валидация: анализ параметров запроса; валидация на уровне модели; валидация на уровне СУБД.

Для разработки клиентской части был выбран фреймворк Angular, так как он обладает мощным синтаксисом шаблонов, позволяя быстро создавать и модифицировать представления пользовательского интерфейса

В серверной части комплекса используется несколько модулей:

- Spring Boot – модуль для упрощенной конфигурации Spring-приложений.
- Spring Web MVC – обработчик HTTP запросов (RESTful API).
- Spring Batch – модуль для разработки пакетных обработок данных.
- Spring Security – модуль для построения авторизации и аутентификации.
- Spring Data – модуль для взаимодействия с любыми базами данных.

## **Безопасность**

Для доступа к программному продукту необходимо получить логин и пароль у администраторов проекта, затем пройти авторизацию в самом приложении. В проекте существует система разделения ролей, то есть, каждому зарегистрированному пользователю присваивается роль, например: оператор, директор, администратор и т.д. Для каждой роли доступны свои разделы и опции. Помимо вышеописанного разделения ролей, за каждым пользователем закрепляется определенная организация, каждой организации доступны действия, разрешенные администраторской организацией. Таким образом, пользователи конкретной организации не могут просмотреть, изменить или удалить данные, на которые им не было выдано разрешение.

Авторизация, аутентификация и защита от распространенных атак обеспечивается платформой Spring Security с интегрированным стандартным протоколом OAuth 2, который де-факто является стандартом для защиты приложений на основе фреймворка Spring.

На серверном уровне доступ к конкретным функциям и операциям обеспечен посредством аннотаций для ролей и организации пользователя

## **ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК**

### **Назначение комплекса**

Основным назначением проекта является автоматизация мероприятий по осуществлению деятельности в сфере ЖКХ (ведение документации, расчеты, начисления, аналитика, судебно-претензионная работа и т.д.).

### **Задачи и функции**

Согласно ТЗ комплекса он выполняет следующие задачи и функции:

1. Ведение базы объектов (домов, квартир, лицевых счетов (ЛС)) с датозависимым хранением характеристик.
2. Снятие (перерасчет) за некачественно предоставляемые услуги по заданной группе ЛС, домов, объединенных различными критериями.
3. Выполнение автоматических перерасчетов по измененным данным за прошлые периоды.
4. Ввод платежных документов общей суммой, по конкретной услуге (или группе услуг). Возможность указания периода оплаты.
5. Печать различных документов на оплату (в том числе счетов-квитанций).
  - Закрепление макетов за поставщиками услуг для автоматического формирования необходимых квитанций по фонду за период;
  - Выпуск квитанций по выбранной части фонду (без привязки поставщику услуг);
  - Выпущенные квитанции сохраняются, с возможностью воспроизвести квитанцию за любой период в полном соответствии с выпущенной в том периоде

- Вывод квитанций осуществляется на принтер или в файлы формата pdf по выбору
6. Наличие процедур закрытия месяца и блокировки начислений на ЛС, на доме.
  7. Расчет размера платы за коммунальные услуги по обращению с твердыми коммунальными отходами и на взносы на капитальный ремонт, реализация перерасчета, согласно Постановлению Правительства РФ от 06.05.2011 №354 «О предоставлении коммунальных услуг собственникам и пользователям помещений в многоквартирных домах и жилых домов».
  8. Расчет начислений с любой даты произвольных переменных параметров (площади, льготы, тарифы, документы на собственность, люди).
  9. Автоматическая генерация номера при создании нового ЛС, по правилу поиска свободного номера среди существующих ЛС.
  10. Учет ЛС потребителей, ведение истории ЛС с точностью до дня, в том числе обеспечено датозависимое хранение следующей информации:
    - информация о зарегистрированных и не зарегистрированных лицах, собственники с информацией о документах подтверждающих право собственности;
    - общая и жилая площадь, право собственности, а также другие необходимые характеристики ЛС;
    - информация о временном отсутствии жильцов;
    - информация о платежах.
  11. Формирование объединенной информации о взаиморасчетах потребителями услуг по периодам оплаты: дата оплаты, поступления, сумма оплат, сумма начисления, сальдо.
  12. Хранение услуг, тарифов и нормативов со следующими возможностями:

- справочник услуг обладает группировкой по видам услуг;
  - для каждой услуги хранится: номер, краткое и полное наименование, единицы измерения, история тарифов и норм;
  - датозависимое хранение тарифов и нормативов, утвержденных постановлением
  - датозависимое хранение индивидуальных домовых тарифов и их калькуляционных составляющих, принятых собраниями собственников жилых помещений;
  - определение норматива по обращению с твердыми коммунальными отходами по параметрам: дом многоквартирный/частный;
13. Расчет по различным методикам по нормативам потребления с точностью до дня. Расчет может обеспечивать как отдельного лицевого, так и произвольного набора ЛС (массовый расчет).
  14. Добавление домов с возможностью импорта данных.
  15. Массовая простановка услуг пи поставщиков на дома и лицевые счета.
  16. Массовая смена услуг и поставщиков.
  17. Массовое закрытие услуг.
  18. Выборка должников по критериям: количество месяцев долга и сумма долга.
  19. Ведение подробных персональных дел по должникам.
  20. Подробная характеристика дел, находящихся на фазах судебного и исполнительного производства.
  21. Отслеживание состояний дел по отдельным фазам.
  22. Печать сводных и аналитических отчетов по выборке дел, находящихся на той или иной фазе.
  23. Печать комплекта документов, в частности выписка из лицевого в суд, иск о взыскании, судебный приказ.
  24. Отслеживание состояний реестров должников

25. Возможность ведения общей базы данных с программами модулем расчета с абонентами (физическими лицами).
26. Формирование претензий.
27. Предоставление отсрочек и контроль графика погашения долгов.
28. Расчет пеней за просрочку по двум алгоритмам: согласно ст. 155 ЖК РФ.
29. Расчет госпошлины согласно ст. 333 НК РФ.
30. Отслеживание состояний реестров должников.

## **Выходные данные**

В качестве выходных данных можно рассматривать:

- результаты расчетов по разным данным;
- платежный документ (квитанции)
- сформированные документы и отчеты в электронном и/или напечатанном виде;
- предоставление информации по запрошенным данным в виде списка с возможностью фильтрации и поиска;
- отображение статистических данных в виде графиков, диаграмм и таблиц.

## **Затрачиваемые ресурсы**

Для корректной работы программного комплекса необходимо обеспечить следующие аппаратные требования:

- RAM 32 Гб
- CPU с 4 и более ядрами, с частотой не ниже 1,5 ГГц
- Дисковое пространство 100 Гб
- Swap 4 Гб