

ОФИЦИАЛЬНЫЙ БЛАНК, ЛОГОТИП

ООО «Бэст Сервис Фасилити Менеджмент»

**Описание функциональных характеристик
программного обеспечения «Building Service Life»**

г. Москва

2024 год

Назначение и область применения

Программное обеспечение «Building Service Life» применяется в сфере эксплуатации объектов капитального строительства.

Программа предназначена для автоматизации обработки заявок, планирования, создания отчетов по показателям объектов капитального строительства, а также для детального анализа и определения проблемных зон в сфере недвижимости.

Функциональные возможности

Функционал программного обеспечения заключается в следующем:

- Создание проекта и ОКС.
- Планирование работ по техническому обслуживанию оборудования и инженерных систем.
- Назначение ответственных и отслеживание статуса выполнения плановых и внеплановых работ.
- Анализ качества выполненных работ и срока реагирования.
- Отслеживание отсележивание и фиксация изменений активов.

Архитектура ПО

Программное обеспечение «Building Service Life» состоит из следующих модулей:

- Управление проектной деятельности
 - Проекты
 - ОКС
 - Сотрудники
 - Компании
- Матрица эксплуатации
 - Помещения
 - Оборудование
 - Инженерные подсистемы
- Обслуживание
 - Технологические карты
 - Обходы
 - Заявки
 - Календарь
- МТО
 - ТМЦ
 - Инструменты
 - Склад

В программном обеспечении имеются следующие инфраструктурные элементы:

1.1. Аутентификация и авторизация

Управление учетными записями: Пользователи системы создаются и управляются исключительно администратором системы. Самостоятельная регистрация пользователей не предусмотрена. Администратор имеет возможность добавлять новых пользователей, редактировать их данные, а также удалять учетные записи.

Авторизация пользователей: Пользователи получают доступ к системе через введение предоставленных администратором учетных данных (логин и пароль). Пароли хранятся в базе данных в зашифрованном виде с использованием алгоритма bcrypt.

Ролевая модель: Система поддерживает ролевую модель управления доступом. Каждому пользователю при создании назначается определенная роль, которая определяет его права и доступ к различным частям системы. Примеры ролей могут включать:

Администратор: Полный доступ ко всем функциям системы, включая управление учетными записями пользователей, настройками системы и просмотром всех данных.

Диспетчер/Мастер/Кладовщик/Рабочий: ограниченный доступ к функциям системы, позволяющий выполнять определенные задачи и просматривать данные.

Система входа и выхода: После успешной авторизации пользователь получает токен доступа, который используется для взаимодействия с защищенными API. Система также предусматривает возможность выхода пользователя из системы, что аннулирует токен и завершает сессию.

1.2. API для взаимодействия с клиентскими приложениями

RESTful API: Серверная часть, разработанная с использованием Django и Django Rest Framework (DRF), предоставляет RESTful API, что позволяет клиентским приложениям взаимодействовать с сервером. API поддерживает стандартные методы HTTP, такие как GET, POST, PUT, DELETE, и обеспечивает удобный и гибкий доступ к данным.

API для авторизации: Специальные эндпоинты предназначены для получения и обновления токенов доступа, необходимых для авторизации запросов к защищенным API. Эти эндпоинты доступны только зарегистрированным в системе пользователям.

API для управления данными: Через API осуществляется управление основными сущностями системы (например, пользователями, проектами, объектами, задачами, оборудованием и т.д.). Эти эндпоинты позволяют создавать, обновлять, удалять и просматривать записи, а также выполнять фильтрацию, сортировку и пагинацию результатов.

API для обработки ошибок: Все ошибки, возникающие в процессе работы API, возвращаются клиенту в стандартизированном формате с указанием кода ошибки и поясняющего сообщения.

1.3. Фоновая обработка задач

Очереди задач: Система использует Celery для выполнения асинхронных задач, обработки данных, выполнения длительных вычислений и других задач, которые не требуют немедленного выполнения.

Отложенные и периодические задачи: Celery позволяет настраивать выполнение задач по расписанию или с задержкой..

1.4. Хранение данных

Реляционная база данных PostgreSQL: Основные данные приложения хранятся в реляционной базе данных PostgreSQL. Это включает данные пользователей, данные о проектах, задачах, логах, настройках системы и т.д.

Миграции базы данных: Система использует встроенные механизмы Django для миграции структуры базы данных. Это обеспечивает простое управление изменениями в базе данных и поддержку её целостности при выпуске новых версий ПО.

1.5. Кэширование и производительность

Использование Redis для кэширования: Redis используется для кэширования данных, которые часто запрашиваются, что позволяет существенно сократить время отклика системы и снизить нагрузку на базу данных.

Кэширование запросов API: Некоторые запросы к API могут кэшироваться на определенное время для увеличения производительности и уменьшения задержек при повторных запросах.

Оптимизация запросов к базе данных: Django ORM используется для оптимизации запросов к базе данных, что позволяет минимизировать количество запросов и время их выполнения.

1.6. Клиентская часть (Frontend)

Фреймворк Nuxt3: Клиентская часть приложения разработана с использованием Nuxt3, что предоставляет мощные возможности для создания современных одностраничных приложений (SPA) с серверным рендерингом (SSR) и статической генерацией страниц.

Интерактивный интерфейс пользователя: Nuxt3 обеспечивает создание высокоинтерактивного интерфейса, который быстро реагирует на действия пользователя благодаря использованию современных методов оптимизации, таких как lazy loading, code splitting и использование кэша браузера.

Поддержка SEO: За счет серверного рендеринга и динамической генерации метатегов, клиентская часть обеспечивает высокую оптимизацию для поисковых систем, что улучшает индексруемость и видимость приложения в сети.

1.7. Безопасность

Использование SSL/TLS: Вся передача данных между клиентом и сервером защищена с использованием SSL/TLS, что предотвращает возможность перехвата и компрометации данных.

Хэширование паролей: Пароли пользователей хранятся в базе данных в зашифрованном виде с использованием bcrypt, что обеспечивает защиту от атак на пароли, таких как перебор или использование радужных таблиц.

Защита от CSRF и XSS: Django предоставляет встроенные механизмы для защиты от атак типа CSRF (Cross-Site Request Forgery) и XSS (Cross-Site Scripting), что значительно повышает безопасность веб-приложения.

Эксплуатационные характеристики

Для установки и эксплуатации программного обеспечения «Building Service Life» необходимо, чтобы рабочее место соответствовало следующим требованиям:

Для SaaS:

- Подключение к интернету - обязательное

- Интернет-браузеры - Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Opera, Yandex Browser

Описание функциональной части программного обеспечения

- Язык программирования - Python 3.11
- СУБД - asgiref 3.7.2, celery 5.3.4, Django 4.2.5, pandas 2.2.2, pdfkit 1.0.0, jinja2 3.1.4, qrcode 7.4.2, requests 2.32.3, Postgresql 14, Redis 6.0.16

Информация, необходимая для установки и эксплуатации

Для установки и правильной эксплуатации программного обеспечения «Building Service Life» необходимо ознакомиться с информацией, размещенной в Руководстве пользователя.

Для активации программного обеспечения обратитесь к ООО «Бэст Сервис Фасилити Менеджмент».