**Приложение «Конфигуратор ПК»**

**Руководство программиста**

**Содержание**

[1 Введение 2](#_Toc181120925)

[1.1 Наименование системы 2](#_Toc181120926)

[1.2 Область применения 2](#_Toc181120927)

[1.3 Условные обозначения системы 2](#_Toc181120928)

[2 Установка системы 3](#_Toc181120929)

[2.1 Требование для установки 3](#_Toc181120930)

[2.2 Установка зависимостей 3](#_Toc181120931)

[2.3 Настройка базы данных 3](#_Toc181120932)

[2.4 Запуск сервера 3](#_Toc181120933)

[3 Внутреняя структура системы 4](#_Toc181120934)

[3.1 Используемый технологический стек 4](#_Toc181120935)

[3.2 Функции и их описание 4](#_Toc181120936)

[3.3 Структура данных 4](#_Toc181120937)

[3.4 Описание API 6](#_Toc181120938)

# Введение

## Наименование системы

Приложения «Конфигуратор ПК».

## Область применения

«Конфигуратор ПК» представляет из себя приложение с веб-интерфейсом, предназначенное для конфигурирования сборок. Основная цель «Конфигуратора ПК» - дать независимую платформу для сборки своего ПК в которой будут сравниваться цены с разных магазинов.

## Условные обозначения системы

Условное обозначение приложения «Конфигуратор ПК» - Приложение.

# Установка системы

## Требование для установки

Сервер должен поддерживать Node.js и Python (в зависимости от выбранного фреймворка: Flask или FastAPI).

База данных: PostgreSQL должна быть установлена на сервере и настроена для работы с приложением.

Зависимости Python: Установите нужные библиотеки Python для работы бэкенда.

Поддержка браузеров: web-UI должен корректно работать в современных браузерах, таких как Chrome, Firefox, Safari, Edge.

## Установка зависимостей

Фронтенд:

Перейдите в папку с фронтенд-проектом и установите зависимости:

npm install

Бэкенд: Установите зависимости из requirements.txt:

pip install -r requirements.txt

## Настройка базы данных

Чтоб установить базу данных на сервер нужно создать новую базу данных в PostgreSQL CREATE DATABASE mydatabase; и запустить TableBuilder.py.

## Запуск сервера

Чтоб запустить сервер, Перейдите в папку с бэкендом и запустите сервер командой:

flask run

# Внутреняя структура системы

## Используемый технологический стек

Сайт использует для фронтенда Vue.js, для бекэнда используется Python с библиотекой Flask. В качестве базы данных используется PostgreSQL

## Функции и их описание

У Сайта имеются следующие функции:

* создать конфигурацию – создает уникальный id для новой конфигурации и добавляет запись в бд с id выбранных комплектующих.
* назвать конфигурацию – добавляет в БД в столбец named название конфигурации;
* изменить конфигурацию – выбирает существующий id конфигурации, подставляет в интерфейс создания конфигурации, ранее выбранные комплектующие, после нажатия на кнопку сохранить в БД подается запрос на обновление;
* удалить конфигурацию – удаляет запись в БД выбранной конфигурации с соответствующим id;
* выбрать конфигурацию из предложенных – выбор из готовых пресетов прописанных в самой программе;
* создать комплектующую – выбирается нужная комплектующая, записывается нужная информация, проверяется на наличие обязательных записей (цена, название, основная характеристика) и добавляется в БД с уникальным id;
* изменить комплектующую – выбирает существующий id комплектующей, подставляет в интерфейс создания комплектующей, ранее выбранные характеристики, после нажатия на кнопку сохранить в БД подается запрос на обновление;
* удалить комплектующую – удаляет запись в БД выбранной конфигурации с соответствующим id;
* Спарсить данные с магазина – это функция работает с помощью библиотеки Selenium;

## Структура данных

Приложение имеет база данных, структура этой базы данных изображена в виде логической ER на рисунке 1 и физической ER которая изображена на рисунке 2 соответственно.

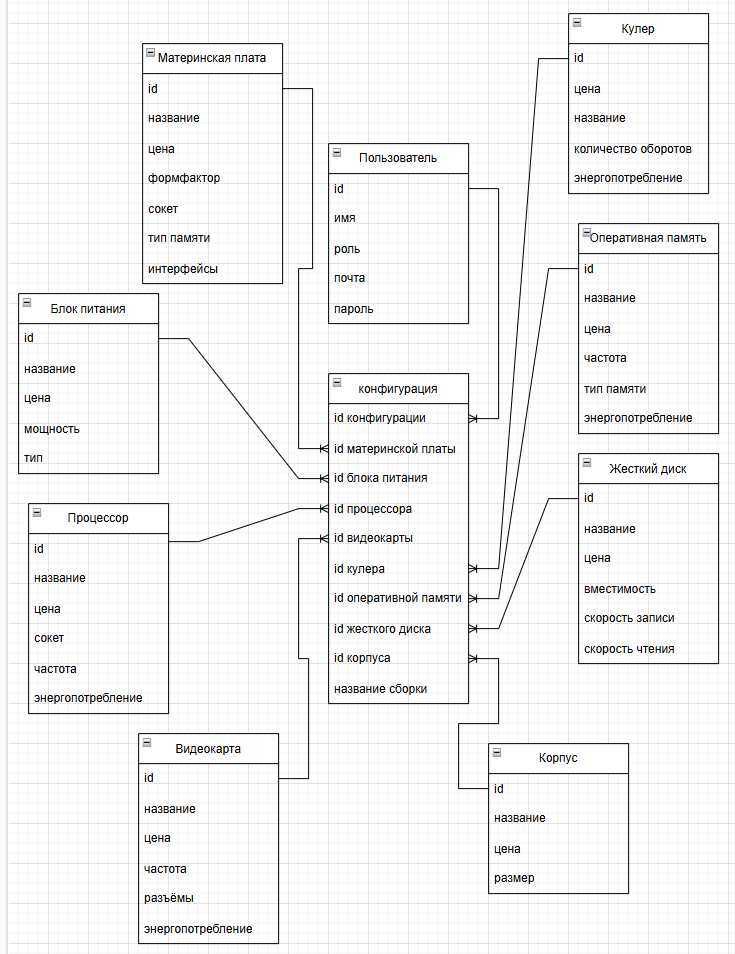


Рисунок 1 – Логическая ER диаграмма

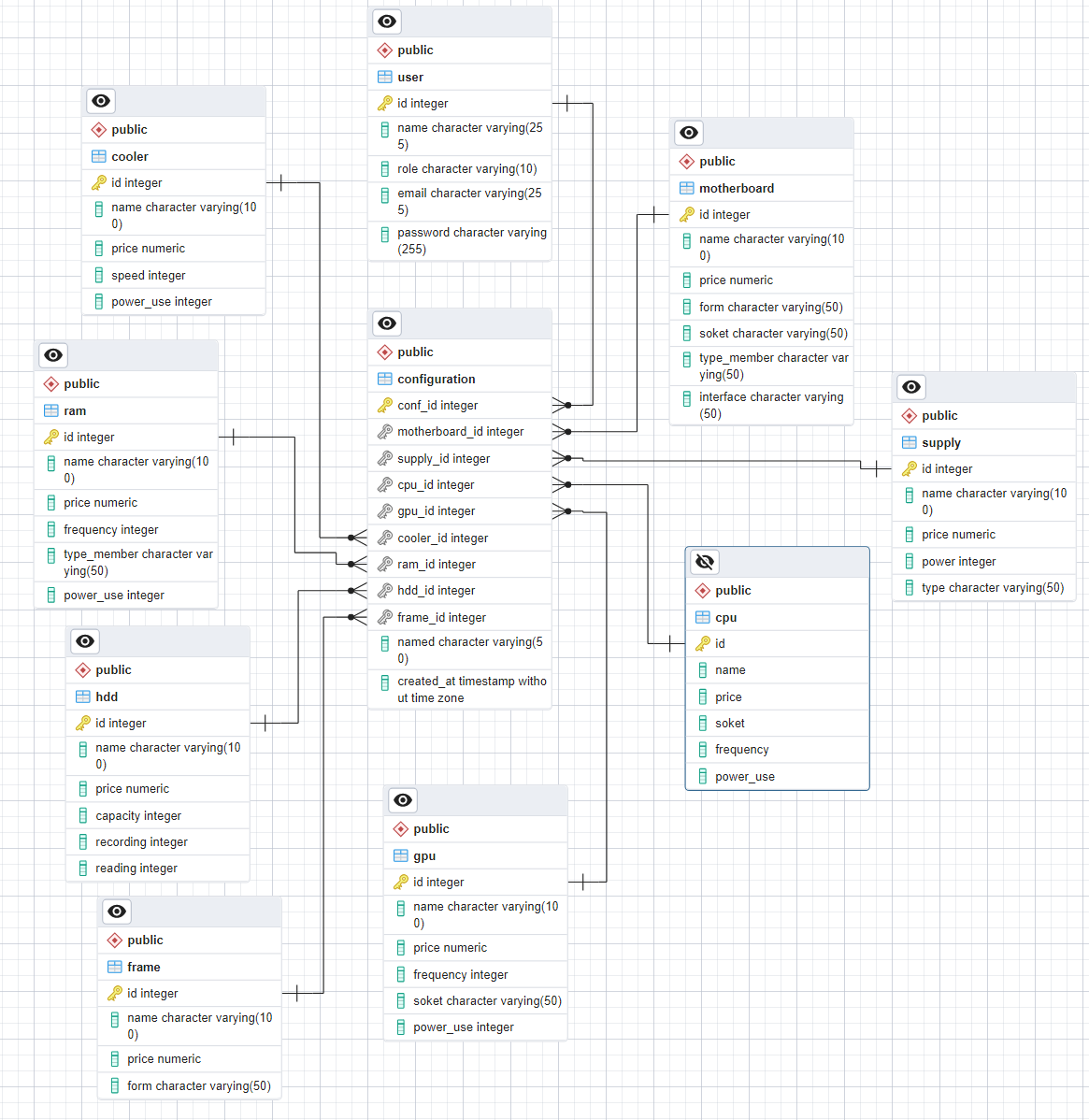


Рисунок 2 – Физическая ER диаграмма

На ER диаграмме показано 10 таблиц: «Материнская плата», «Блок питания», «Процессор», «Видеокарта», «Кулер», «Оперативная память», «Жесткий диск», «Корпус», «Конфигурация» и «Пользователь».

* Таблица «Материнская плата»
* id: Уникальный идентификатор материнской платы.
* name: Название материнской платы.
* price: Стоимость материнской платы.
* form: Форм-фактор материнской платы (например, ATX, Micro-ATX).
* soket: Тип сокета, поддерживаемого материнской платой.
* Type\_member: Тип оперативной памяти, поддерживаемый материнской платой.
* interface: Список интерфейсов, доступных на материнской плате.
* Таблица «Блок питания»
* id: Уникальный идентификатор блока питания.
* name: Название блока питания.
* price: Стоимость блока питания.
* power: Мощность блока питания в ваттах.
* type: Тип блока питания.
* Таблица «Процессор»
* id: Уникальный идентификатор процессора.
* name: Название процессора.
* price: Стоимость процессора.
* soket: Тип сокета, поддерживаемый процессором.
* frequancy: Частота процессора.
* Power\_use: Потребляемая мощность процессора.
* Таблица «Видеокарта»
* id: Уникальный идентификатор видеокарты.
* name: Название видеокарты.
* price: Стоимость видеокарты.
* frequancy: Частота видеокарты.
* soket: Список доступных разъемов на видеокарте.
* Power\_use: Потребляемая мощность видеокарты.
* Таблица «Кулер»
* id: Уникальный идентификатор кулера.
* price: Стоимость кулера.
* name: Название кулера.
* speed: Количество оборотов кулера.
* Power\_use: Потребляемая мощность кулера.
* Таблица «Оперативная память»
* id: Уникальный идентификатор оперативной памяти.
* name: Название оперативной памяти.
* price: Стоимость оперативной памяти.
* frequancy: Частота оперативной памяти.
* Type\_member: Тип оперативной памяти.
* Power\_use: Потребляемая мощность оперативной памяти.
* Таблица «Жесткий диск»
* id: Уникальный идентификатор жесткого диска.
* name: Название жесткого диска.
* price: Стоимость жесткого диска.
* capacity: Объем жесткого диска.
* recording: Скорость записи данных на жесткий диск.
* reading: Скорость чтения данных с жесткого диска.
* Таблица «Корпус»
* id: Уникальный идентификатор корпуса.
* name: Название корпуса.
* price: Стоимость корпуса.
* form: Размер корпуса.
* Таблица «Конфигурация»
* conf\_id: Уникальный идентификатор конфигурации.
* Motherboard\_id: Ссылка на материнскую плату, используемую в конфигурации (внешний ключ к таблице «Материнская плата»).
* Supply\_id: Ссылка на блок питания, используемый в конфигурации (внешний ключ к таблице «Блок питания»).
* Cpu\_id: Ссылка на процессор, используемый в конфигурации (внешний ключ к таблице «Процессор»).
* Gpu\_id: Ссылка на видеокарту, используемую в конфигурации (внешний ключ к таблице «Видеокарта»).
* Cooler\_id: Ссылка на кулер, используемый в конфигурации (внешний ключ к таблице «Кулер»).
* Ram\_id: Ссылка на оперативную память, используемую в конфигурации (внешний ключ к таблице «Оперативная память»).
* Hdd\_id: Ссылка на жесткий диск, используемый в конфигурации (внешний ключ к таблице «Жесткий диск»).
* Frame\_id: Ссылка на корпус, используемый в конфигурации (внешний ключ к таблице «Корпус»).
* name: Название конкретной сборки или конфигурации.
* Таблица «Пользователь»
* id: Уникальный идентификатор пользователя.
* name: имя пользователя
* role: роль (Пользователь или администратор).
* email: почта.
* Password: пароль.