МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ Профиль «Информационные технологии» Командный кейс №5 «Управление спортивным инвентарем»

Участники: Анисимов Ярослав, Балан Анастасия, Давыдова Валерия, Развенков Кирилл, Молоканова Кира

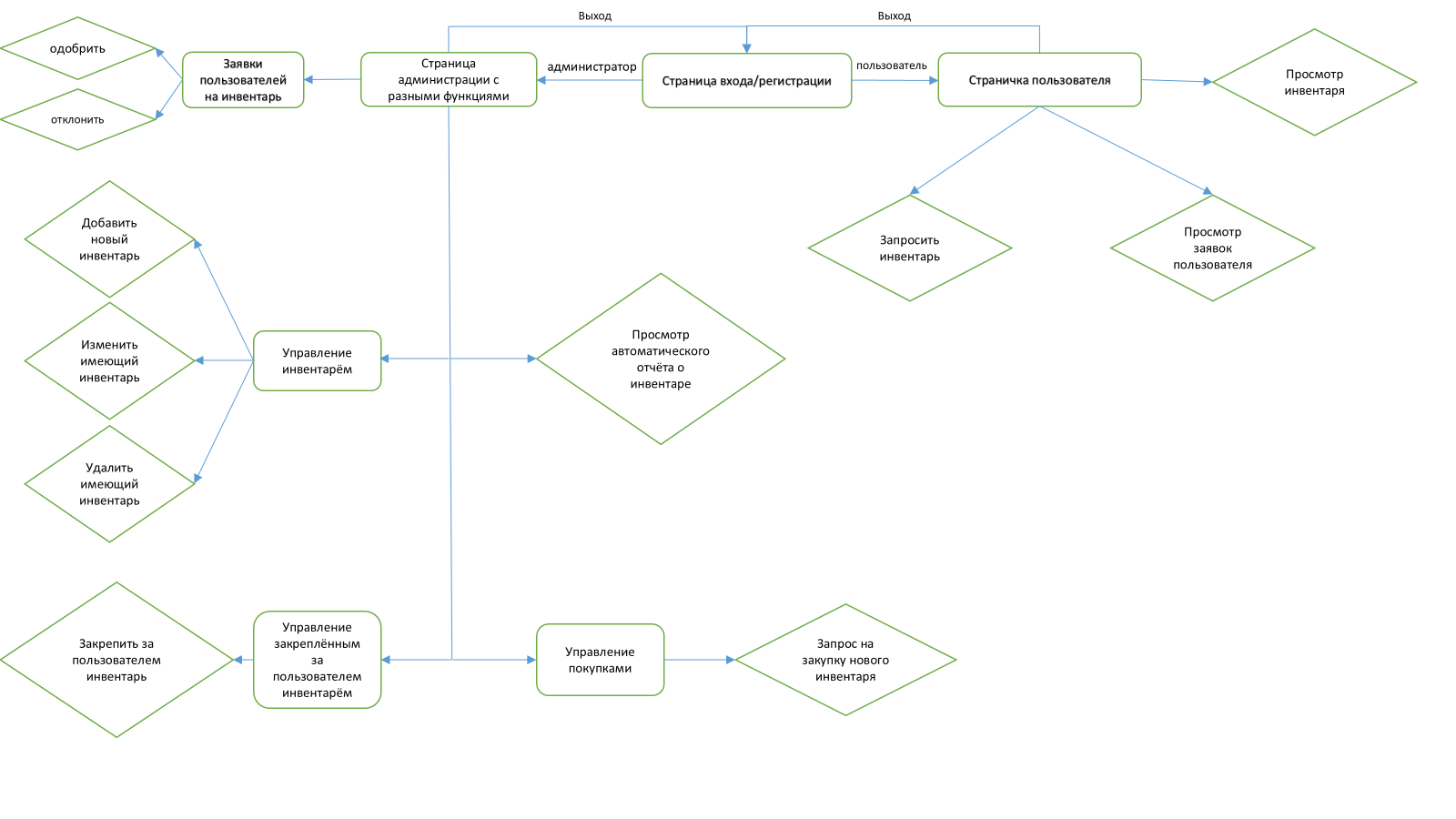
1. Обоснование выбора языка программирования и используемых программных средств
2. Структурная и функциональная схемы программного продукта.
3. Блок-схема работы основного алгоритма.
4. Описание особенностей и аргументация выбранного типа СУБД.
5. Схема базы данных.
6. Ссылка на репозиторий

1.Обоснование выбора языка программирования и используемых программных средств

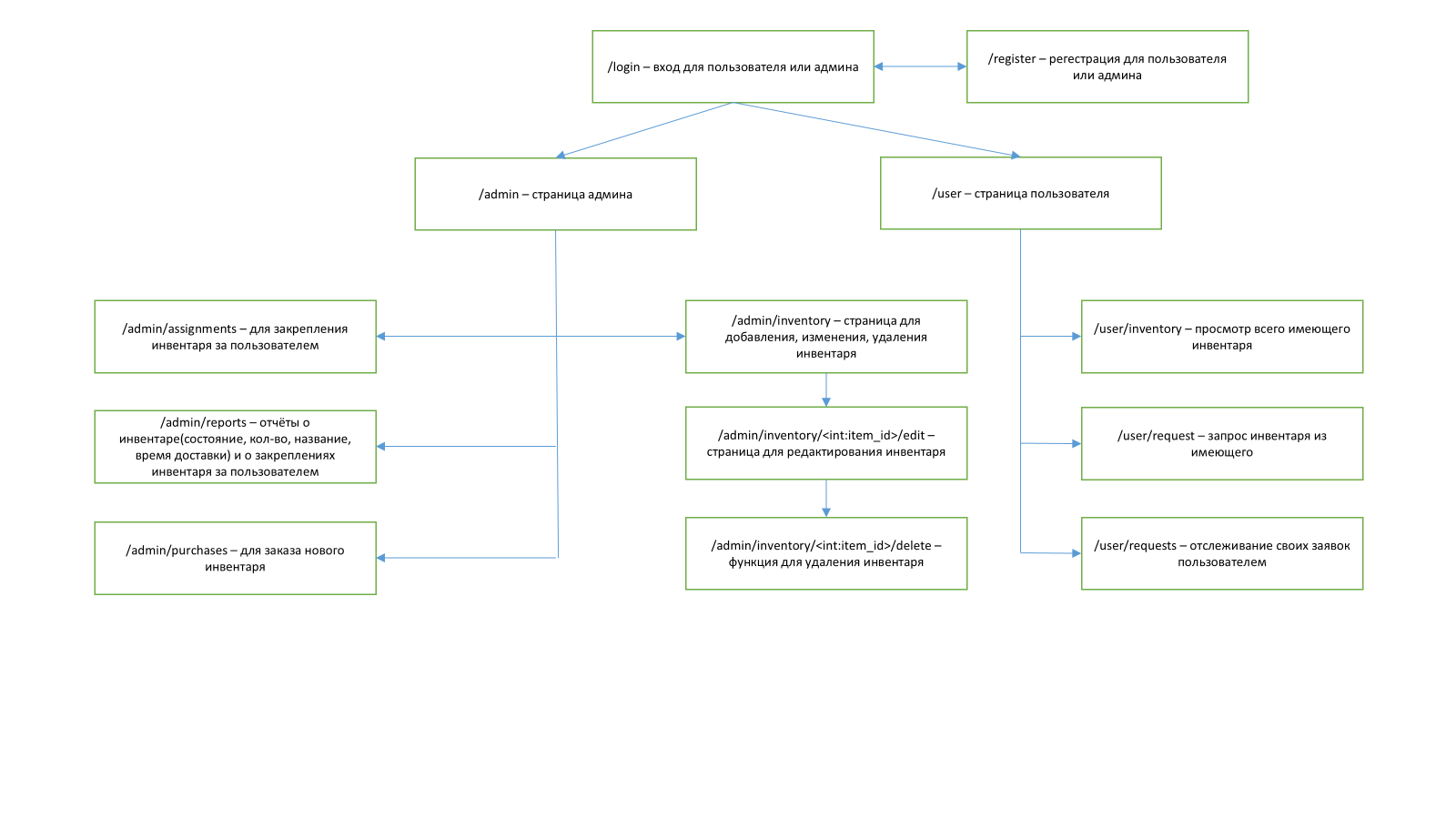
Для данного проекты мы использовали именно HTML, CSS, Pyton потому, что они максимально удобны для использования и связи друг с другом, также они удовлетворяют техническим требованиям, целям и сложности проекта.

2.Структурная и функциональная схемы программного продукта.

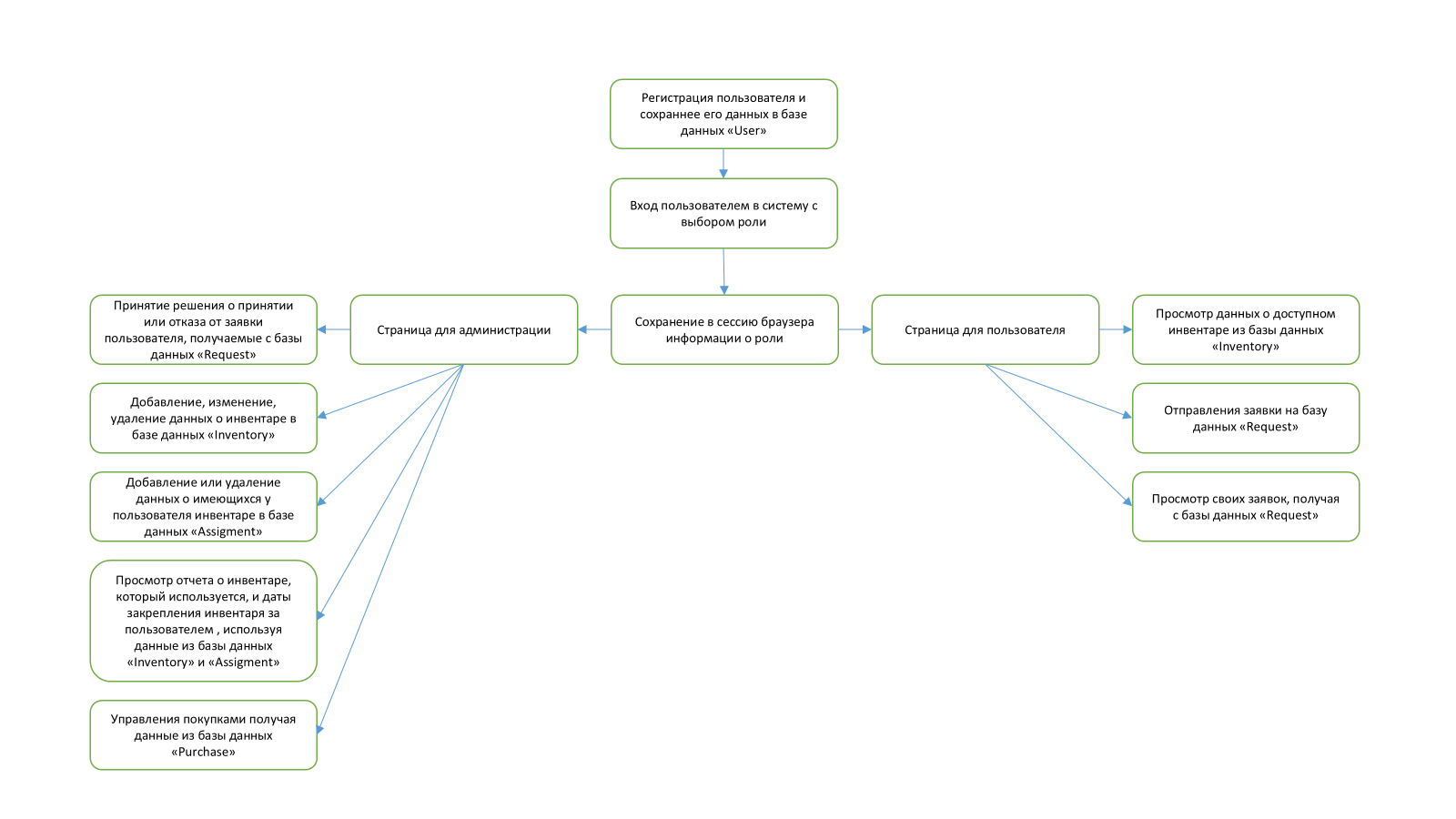
1. Структурная схема программного продукта



2)функциональная я схема программного продукта



3.Блок-схема работы основного алгоритма.



4.Описание особенностей и аргументация выбранного типа СУБД.

SQLAlchemy — это мощный и гибкий объектно-реляционный маппер (ORM) для Python. Он не является СУБД в прямом смысле, а инструментом для взаимодействия с различными СУБД. Его ключевые особенности и аргументы в пользу выбора включают:

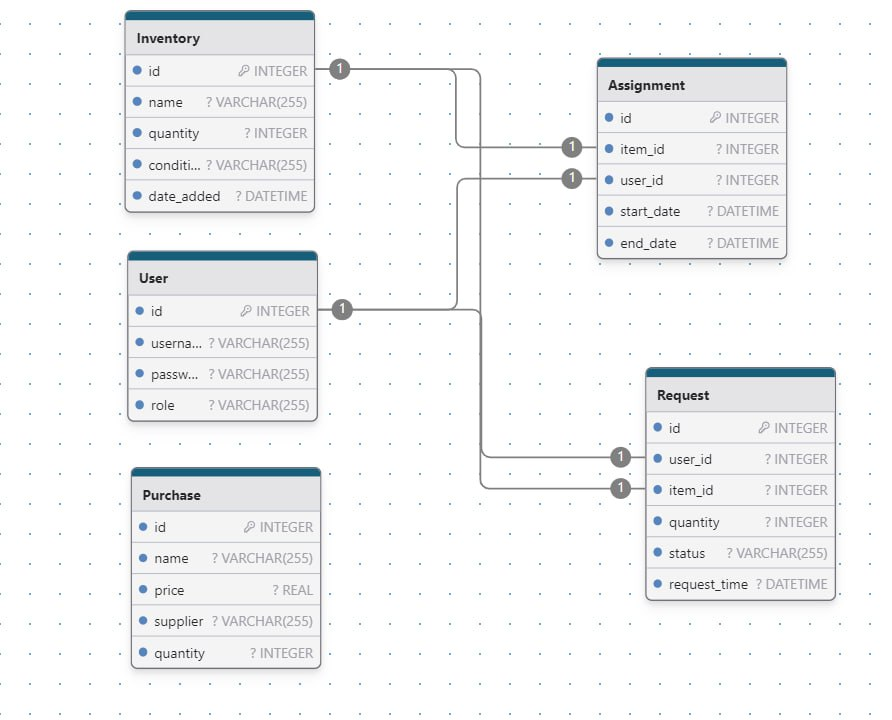
**Особенности:**

* **Объектно-реляционное отображение (ORM):** SQLAlchemy позволяет работать с базой данных на уровне объектов Python. Это значит, что вместо написания SQL-запросов, вы можете создавать, изменять и получать данные, используя привычные для вас структуры данных Python (например, классы). Это упрощает разработку, повышает читаемость кода и снижает вероятность ошибок.
* **Гибкость и масштабируемость:** SQLAlchemy может работать с различными СУБД (PostgreSQL, MySQL, SQLite, Oracle, Microsoft SQL Server и т.д.). Это позволяет вам выбрать наиболее подходящую СУБД для вашей задачи, не привязываясь к конкретному варианту. Вы можете легко переключаться между СУБД, не меняя существенно код приложения.
* **Аналитический язык запросов:** SQLAlchemy позволяет создавать запросы к базе данных не только с помощью ORM, но и в чистом SQL. Это даёт гибкость для выполнения сложных задач, которые могут быть неэффективны при использовании только ORM.
* **Высокая производительность:** Благодаря оптимизации и возможностям кэширования SQLAlchemy может работать с базой данных эффективно. Он генерирует оптимальные SQL-запросы для конкретной СУБД.
* **Поддержка различных диалектов:** SQLAlchemy предоставляет поддержку различных диалектов SQL для различных СУБД. Это гарантирует, что вы можете использовать синтаксис, привычный для конкретной СУБД, без необходимости написания сложных преобразований запросов.
* **Поддержка атрибутов, отношений и методов:** SQLAlchemy позволяет определять объекты инвентаря (например), описывая их атрибуты (такие как название, количество, состояние) и отношения между ними. Например, можно задать отношение “один ко многим” между таблицей пользователей и таблицей инвентаря. Этот подход упрощает структурирование данных и взаимодействие с ними.
* **Отладка:** SQLAlchemy предоставляет множество инструментов для отладки, включая инструменты для инспекции и модификации запросов, и для проверки связи между объектами и их представлением в БД.

**Аргументы в пользу SQLAlchemy:**

* **Упрощение разработки:** ORM-подход с использованием SQLAlchemy существенно упрощает разработку приложений, так как не требует от разработчика написания SQL-запросов в большинстве случаев. Это приводит к снижению времени разработки и упрощает поддержку кода.
* **Улучшение качества кода:** Реализуя операции с базой данных через объекты, SQLAlchemy способствует более чистому и читаемому коду. Это облегчает понимание кода и поддержку другими программистами.
* **Увеличение производительности:** SQLAlchemy может генерировать эффективные SQL-запросы, избегая неэффективных операций, которые могут возникать при ручном написании SQL. Это позволяет поддерживать оптимальную производительность приложения.
* **Гибкость:** Взаимодействие с разными СУБД реализовано через гибкие диалекты SQL.

5.Схема базы данных.



6. Ссылка на репозиторий.

https://github.com/versklin1nastya/bebobriki