**Описание курсового проекта по базам данных (система управления библиотекой)**

Первым и самым важным выбором была база данных. После рассмотрения нескольких вариантов, было решено остановиться на PostgreSQL. Почему именно она? PostgreSQL – это мощная и надежная реляционная СУБД, которая предоставляет все необходимые инструменты для хранения и управления данными, что очень важно для такого приложения как библиотека. Она имеет открытый исходный код, что очень важно для некоммерческих проектов, а так же имеет очень хороший функционал. Так как она реляционная база данных, то это позволяет эффективно организовывать данные в таблицы и устанавливать связи между ними, что очень важно для создания системы управления библиотекой.

Для создания графического интерфейса пользователя и основной логики приложения был выбран Python с библиотекой Tkinter. Python – это простой, понятный, и при этом достаточно мощный язык программирования, который хорошо подходит для создания десктопных приложений. А библиотека Tkinter предоставляет все необходимые инструменты для создания графического интерфейса, так же она очень проста в использовании и изучении.

Для работы с базой данных PostgreSQL используется библиотека psycopg2. Она предоставляет удобный и эффективный интерфейс для выполнения SQL-запросов и управления данными. Это позволяет мне напрямую общаться с базой данных, и при этом не усложнять работу с ней.

Также стоит отметить, что для создания резервных копий базы данных используется утилита командной строки pg\_dump, и psql для их востановления. Это надежный способ, позволяющий сохранять резервные копии в виде SQL-файлов и гарантирующий что данные не будут утеряны.

Для работы с протоколом FTP использована стандартная библиотека Python ftplib. Так же эта библиотека довольно проста в использовании.

Для защиты данных и хранения паролей используется библиотека bcrypt. Она обеспечивает надежное хеширование паролей, что делает систему более безопасной.

Для работы с протоколом SFTP была использована библиотека Paramiko.

**Предметная область**

Данная система управления библиотекой призвана автоматизировать основные процессы, происходящие в библиотеке. Эти процессы включают в себя: формирование библиотечного фонда, обработку и каталогизацию новых ресурсов, хранение библиотечного фонда, обслуживание читателей и регистрацию новых читателей и выдачу книг.

В системе выделяются три основные роли, каждая из которых имеет свой набор обязанностей и функционала.

1. Читатель, или обычный пользователь, имеет может просматривать свою историю выданных книг.

2. Библиотекарь ведет учет библиотечного фонда, осуществляет выдачу и прием книг, отвечает за каталогизацию.

3. Администратор — это пользователь с полными правами. В его обязанности входит управление пользователями, назначение ролей и, в том числе, управление резервным копированием базы данных, что бы обеспечить сохранность информации.

В основе предметной области лежат следующие объекты:

1. Книга: Представляет собой основной ресурс библиотеки. Книга имеет название, автор(ов), жанр, год издания, количество экземпляров и состояние (доступна или нет).

2. Автор: Создатель книги. Автор имеет имя, фамилию, а так же может иметь дату рождения.

3. Жанр: Определяет тематическую принадлежность книги (фантастика, детектив, роман и т.д.)

4. Читатель: Пользователь библиотеки, имеющий имя, фамилию, контактную информацию, ID. Читатель может иметь или не иметь связи с пользователем системы.

5. Выдача книг: Представляет собой связь между книгой и читателем. Содержит информацию о дате выдачи, сроке возврата и дате фактического возврата (если есть).

**Требования к системе**

Стояла задача создать полноценную систему, которая помогла бы в работе библиотеки. Поэтому, были выделены несколько ключевых моментов, которые должны быть в системе в плане функциональности, и то, как она должна работать в принципе.

Функциональные требования:

Управление пользователями: система должна давать возможность регистрировать новых пользователей. Все логины, пароли, имена, фамилии, а так же телефоны, если это нужно – все это должно где то хранится, ну и, конечно же, нельзя забыть про безопасность. Система также должна позволять пользователям входить, используя логин и пароль, чтобы убедиться что это именно тот пользователь.

Помимо этого, в системе должны быть роли (администратор, библиотекарь, обычный пользователь), которые определяют, что каждый из них может делать в системе. Должна быть возможность удалять ненужных пользователей.

Управление книгами: библиотекарь должен уметь добавлять в каталог информацию о новых книгах. Это значит, что нужно будет добавлять название книги, авторов, год издания, жанр, количество копий и т.д. Должна быть возможность редактировать информацию о книгах, если что-то изменилось или если нашли ошибку. Также нужно удалять записи о книгах, которых уже нет в фонде.

Управление выдачей книг: система должна давать возможность выдавать книги читателям, регистрировать дату выдачи и срок возврата. Должна быть возможность отмечать возврат книг и вестись история выдачи.

Бэкап и восстановление: должна быть возможность восстановить базу данных из бэкапа.

**Команда для создания и поддержания проекта**

Backend-разработчик – 2 чел:

• Разработка ядра системы на Python.

• Создание API для фронтенда.

• Реализация бизнес-логики (выдача книг, управление пользователями).

• Работа с PostgreSQL и оптимизация запросов.

• Интеграция с FTP/SFTP, реализация бэкапов.

Frontend-разработчик – 1 чел:

• Создание удобного и современного пользовательского интерфейса.

DevOps – 1 чел:

• Настройка серверов для БД и приложения.

• Организация процесса автоматического развертывания и мониторинга.

• Настройка отказоустойчивости и регулярного бэкапа БД

Менеджер проекта – 1 чел:

• Постановка задач команде, контроль сроков и бюджета.

• Коммуникация с заказчиком.

• Решение организационных вопросов.

Аналитик – 1 чел:

• Детальное описание предметной области и процессов библиотеки.

• Написание подробных технических требований (ТЗ).

• Проектирование пользовательских сценариев (UX).

Дизайнер – 1 чел:

• Проектирование логики взаимодействия пользователя с системой (UX).

• Создание визуального дизайна интерфейсов (UI) — макетов, кнопок, форм.

Тестировщик – 1 чел:

• Тестирование приложения на предмет ошибок и соответствия требованиям.

**Список источников и литературы**

1. PostgreSQL Documentation:

Официальная документация по PostgreSQL [Электронный ресурс] Режим доступа: https://www.postgresql.org/docs/

Руководство по утилите pg\_dump[Электронный ресурс] Режим доступа: https://www.postgresql.org/docs/current/app-pgdump.html

Руководство по утилите psql [Электронный ресурс] Режим доступа: https://www.postgresql.org/docs/current/app-psql.html

2. Python Documentation[Электронный ресурс] Режим доступа: https://docs.python.org/3/

3. Tkinter Documentation[Электронный ресурс] Режим доступа: https://docs.python.org/3/library/tkinter.html

4. psycopg2 Documentation[Электронный ресурс] Режим доступа: https://www.psycopg.org/docs/

5. bcrypt Documentation[Электронный ресурс] Режим доступа: https://pypi.org/project/bcrypt/

6. Paramiko Documentation[Электронный ресурс] Режим доступа: http://docs.paramiko.org/en/stable/

7. ftplib Documentation[Электронный ресурс] Режим доступа: https://docs.python.org/3/library/ftplib.html