МИНОБРНАУКИ РФ

ФГБОУ ВО Тверской государственный технический университет

Кафедра “Программное обеспечение”

Тема курсовой работы:

«Разработка системы автоматизации с использованием СУБД PostgreSQL: online-библиотека.»

Выполнил:

Соколов Никита Дмитриевич, 3 курс, факультет

информационных технологий, группа: Б ПИН РИС-20.05

Проверил: Артемов И

2. Введение:

Задача проекта по созданию системы автоматизации для онлайн-библиотеки заключается в разработке комплексной программной системы, предназначенной для управления библиотечной коллекцией книг и обеспечения удобного доступа пользователей к этим ресурсам через интернет. Основная цель данного проекта - повысить эффективность и доступность библиотечных услуг, как для администраторов библиотеки, так и для пользователей.

Основные задачи проекта включают в себя:

1. Управление коллекцией книг: Создание базы данных, которая будет содержать информацию о книгах в библиотеке, включая название, автора, жанр, издательство, год издания и другие характеристики. Эта база данных будет обеспечивать эффективное хранение и организацию книг.

2. Регистрация и аутентификация пользователей: Разработка системы регистрации пользователей и обеспечение безопасности хранения их личных данных и учетных записей. Пользователи смогут создавать свои аккаунты и входить в систему для доступа к библиотечным ресурсам.

3. Выдача и возврат книг: Реализация механизма для выдачи книг пользователю при их запросе и фиксация возврата книги по окончании использования. Это позволит пользователям брать книги в аренду и возвращать их в установленные сроки.

4. Поиск и фильтрация книг: Создание возможности для пользователей осуществлять поиск книг по различным параметрам, таким как автор, жанр, ключевые слова и другие. Также предоставление фильтров для уточнения результатов поиска.

5. Управление аккаунтами пользователей: Позволение пользователям редактировать свои профили, сменять пароли и управлять своими личными данными.

6. Управление доступом и ролями: Организация доступа к системе с разными ролями, например, администраторы библиотеки и обычные пользователи. Администраторы могут управлять коллекцией книг и пользователями.

7. Обработка ошибок и безопасность: Обеспечение безопасности данных и обработки ошибок для предотвращения нежелательных ситуаций, таких как несанкционированный доступ или потеря данных.

8. Интерфейс пользователя: Разработка удобного и интуитивно понятного веб-интерфейса для взаимодействия пользователей с библиотечной системой.

9. Документация и обучение: Подготовка документации для пользователей и администраторов системы, а также обеспечение необходимого обучения для эффективного использования системы.

Этот проект направлен на автоматизацию рутинных операций в библиотеке, упрощение доступа к информации и увеличение удовлетворенности пользователей.

3. Теоретический раздел:

- Предметная область онлайн-библиотеки охватывает широкий спектр задач и операций, связанных с управлением информацией о книгах, обслуживанием пользователей и обеспечением им доступа к библиотечным ресурсам. Вот более подробное описание задач и функциональности, которые должна решать система автоматизации для онлайн-библиотеки:

1. Управление коллекцией книг:

- Добавление новых книг в библиотеку с указанием информации о названии, авторе, издательстве, годе издания и других характеристиках.

- Редактирование и обновление данных о существующих книгах, включая внесение изменений в информацию о книге.

2. Регистрация и аутентификация пользователей:

- Разработка системы регистрации новых пользователей, которые могут создавать свои учетные записи в системе.

- Аутентификация пользователей при входе в систему для обеспечения безопасности и управления доступом.

3. Выдача и возврат книг:

- Возможность пользователей арендовать книги из библиотечной коллекции.

- Отслеживание сроков аренды и оповещение пользователей о необходимости возврата книг.

- Регистрация возврата книг и освобождение их для последующей аренды.

4. Поиск и фильтрация книг:

- Разработка механизма поиска книг по различным параметрам, таким как название, автор, жанр, ключевые слова и другие характеристики.

- Предоставление пользователю возможности фильтровать результаты поиска для уточнения запроса.

5. Управление аккаунтами пользователей:

- Позволение пользователям редактировать свои профили, включая изменение личных данных и паролей.

- Возможность для администраторов библиотеки управлять аккаунтами пользователей и решать проблемы с учетными записями.

6. Управление доступом и ролями:

- Организация разных ролей и уровней доступа в системе, включая администраторов, библиотекарей и обычных пользователей.

- Установление прав доступа к определенным функциям системы в зависимости от роли.

7. Обработка ошибок и безопасность:

- Обеспечение безопасности данных пользователей и книг, включая шифрование личных данных и механизмы защиты от несанкционированного доступа.

- Разработка обработки ошибок для предотвращения ситуаций, таких как дублирование книг или неверные запросы.

8. Интерфейс пользователя:

- Создание удобного веб-интерфейса для взаимодействия пользователей с библиотечной системой, включая поиск книг, просмотр информации о книгах и аренду.

9. Документация и обучение:

- Подготовка документации для пользователей и администраторов системы для обучения и эффективного использования всех функций.

Разработка такой системы позволит библиотеке автоматизировать и оптимизировать свои операции, обеспечивая удобный и эффективный доступ пользователей к книжным ресурсам в онлайн-режиме.

- Реляционные базы данных (РБД) представляют собой тип хранилища данных, где информация организована в виде таблиц с реляционными связями между ними. В этом контексте таблицы соответствуют сущностям предметной области (например, таблица "Книги" и таблица "Пользователи"), а реляционные связи определяют, как эти сущности взаимодействуют друг с другом (например, одна книга может быть арендована одним из пользователей).

СУБД PostgreSQL (Structured Query Language DataBase) - это одна из популярных реляционных систем управления базами данных. Вот некоторые причины, почему PostgreSQL может быть хорошим выбором для проекта системы автоматизации онлайн-библиотеки:

1. Бесплатное и открытое ПО: PostgreSQL распространяется под лицензией PostgreSQL, которая позволяет использовать, изменять и распространять СУБД бесплатно. Это снижает затраты на лицензии и поддержку, что особенно важно для небольших библиотек и стартапов.

2. Высокая производительность: PostgreSQL обладает хорошей производительностью и способен обрабатывать большие объемы данных и сложные запросы. Это важно для эффективной работы онлайн-библиотеки, особенно если она имеет большую коллекцию книг и активных пользователей.

3. Поддержка реляционных моделей данных: PostgreSQL полностью поддерживает реляционную модель данных, что делает его подходящим для проектов, где важно хорошо структурировать и организовать данные, как это необходимо для управления книгами и пользователями библиотеки.

4. Богатые возможности: PostgreSQL предоставляет богатый набор возможностей, включая поддержку сложных запросов, триггеров, процедур и многих других функций, которые могут быть полезными при разработке сложных приложений, таких как онлайн-библиотека.

5. Сообщество и поддержка: PostgreSQL имеет активное сообщество разработчиков и пользователей, что обеспечивает поддержку, обновления и решение возникающих проблем.

6. Безопасность: PostgreSQL обеспечивает высокий уровень безопасности данных, включая механизмы аутентификации и авторизации, а также шифрование данных.

7. Поддержка для многих операционных систем: PostgreSQL доступен для различных операционных систем, что делает его удобным для интеграции в разные инфраструктуры.

8. Гибкость и расширяемость: PostgreSQL позволяет разрабатывать собственные пользовательские функции и типы данных, что может быть полезно при реализации специфических требований проекта.

В целом, PostgreSQL является мощной и надежной СУБД, которая хорошо подходит для проектов, требующих хранение и управление большим объемом структурированных данных, таких как онлайн-библиотеки. Его бесплатная и открытая природа, а также богатые функциональные возможности делают его привлекательным выбором для разработчиков.

4. Практический раздел:

- Для онлайн-библиотеки, требующей хранение информации о книгах, пользователях и операциях аренды и возврата книг, можно разработать следующую схему базы данных:

1. Таблица "Книги" (Books):

- book\_id (первичный ключ)

- название книги

- автор

- жанр

- издательство

- год издания

- ISBN (международный стандартный номер книги)

- описание

- статус (доступно/арендовано)

2. Таблица "Пользователи" (Users):

- user\_id (первичный ключ)

- имя

- фамилия

- email

- пароль (хэшированный)

- дата регистрации

- роль (пользователь/администратор)

3. Таблица "Аренды" (Rentals):

- rental\_id (первичный ключ)

- user\_id (внешний ключ, связь с таблицей "Пользователи")

- book\_id (внешний ключ, связь с таблицей "Книги")

- дата аренды

- дата возврата (может быть NULL, если книга еще не возвращена)

Эта схема предоставляет следующие связи между таблицами:

- Множество пользователей может арендовать множество книг, что создает связь многие-ко-многим между таблицами "Пользователи" и "Книги" через таблицу "Аренды".

С помощью этой схемы базы данных можно эффективно управлять информацией о книгах и пользователях, а также отслеживать операции аренды и возврата книг. Это позволяет системе автоматизации библиотеки отслеживать, какие книги доступны для аренды, кто арендует книги и когда они должны быть возвращены.

Ниже приведены основные шаги и алгоритмы работы программы для онлайн-библиотеки, включая регистрацию и аутентификацию пользователей, выдачу и возврат книг, поиск и фильтрацию книг:

1. Регистрация пользователя:

- Пользователь заполняет форму регистрации, вводя свои данные, такие как имя, фамилия, email и пароль.

- Система проверяет, не зарегистрирован ли уже пользователь с таким email.

- Если email уникален, система создает запись о новом пользователе в таблице "Пользователи", хеширует пароль и сохраняет его хеш в базе данных.

2. Аутентификация пользователя:

- Пользователь вводит свой email и пароль в форму входа.

- Система проверяет, существует ли пользователь с таким email.

- Если пользователь найден, система сравнивает хеш введенного пароля с хешем, хранящимся в базе данных.

- Если хеши совпадают, пользователь аутентифицирован и получает доступ к своему аккаунту.

3. Выдача книги:

- Пользователь входит в систему и выбирает книгу для аренды, указывая ее ID или используя поиск и фильтрацию.

- Система проверяет, доступна ли книга (статус "доступно") и не арендована ли уже пользователем.

- Если условия выполнены, система создает запись об аренде в таблице "Аренды", устанавливая дату аренды и дату возврата.

4. Возврат книги:

- Пользователь входит в систему и выбирает книгу, которую он хочет вернуть.

- Система проверяет, арендована ли книга пользователем и не просрочена ли дата возврата.

- Если условия выполнены, система обновляет запись об аренде, устанавливая дату возврата.

5. Поиск и фильтрация книг:

- Пользователь может выполнять поиск книг по различным параметрам, таким как название, автор, жанр и др.

- Система выполняет SQL-запрос к базе данных, фильтруя книги в соответствии с запросом пользователя.

- Результаты поиска отображаются пользователю.

Эти алгоритмы позволяют пользователям регистрироваться, аутентифицироваться, арендовать и возвращать книги, а также осуществлять поиск и фильтрацию книг. Кроме того, следует добавить обработку ошибок и обеспечение безопасности данных в этих операциях, чтобы обеспечить корректную и безопасную работу системы.

Программа для онлайн-библиотеки должна предоставлять разнообразные операции для управления книгами и пользователями. Вот операции, которые программа может выполнять:

Управление книгами:

1. Добавление новых книг:

- Администратор системы может войти в учетную запись и добавить новые книги в библиотеку.

- Для этого он заполняет форму с информацией о книге, включая название, автора, жанр, издательство, год издания и другие характеристики.

- Система проверяет уникальность книги (например, по ISBN), присваивает ей уникальный ID и добавляет книгу в базу данных "Книги".

2. Редактирование информации о книгах:

- Администратор может редактировать информацию о существующих книгах.

- Система предоставляет форму для внесения изменений в данные о книге, например, исправление названия или изменение автора.

- Изменения сохраняются в базе данных.

3. Удаление книг:

- Администратор имеет возможность удалять книги из библиотеки, если они больше не доступны для аренды или неактуальны.

- Система удаляет запись о книге из базы данных "Книги".

Управление пользователями:

1. Регистрация новых пользователей:

- Новые пользователи могут зарегистрировать свои аккаунты, заполнив форму с личными данными и учетными данными.

- Система проверяет уникальность email пользователя и, при успешной регистрации, создает новую запись о пользователе в базе данных "Пользователи".

2. Редактирование профилей пользователей:

- Зарегистрированные пользователи могут редактировать свои профили, изменяя личные данные, пароли и другие сведения.

- Система обновляет данные пользователя в базе данных.

3. Удаление аккаунтов пользователей:

- Администратор имеет возможность удалять учетные записи пользователей в случае несоблюдения правил пользования библиотекой.

- Система удаляет запись о пользователе из базы данных и возврата его арендованных книг.

4. Управление ролями пользователей:

- Администратор может назначать и изменять роли пользователей, например, предоставлять определенным пользователям статус администратора для управления библиотекой.

Эти операции позволяют эффективно управлять книгами и пользователями в онлайн-библиотеке, обеспечивая администраторам полный контроль над коллекцией и пользователями, а также предоставляя пользователям удобный интерфейс для регистрации, обновления профилей и поиска книг.

5. Выводы:

Итоги работы:

В рамках разработки системы автоматизации для онлайн-библиотеки были выполнены следующие ключевые моменты и достижения:

1. Создание схемы базы данных: Была разработана схема базы данных, включая таблицы для хранения информации о книгах, пользователях и операциях аренды/возврата книг.

2. Алгоритмы работы программы: Были описаны алгоритмы работы программы, включая регистрацию и аутентификацию пользователей, выдачу и возврат книг, а также поиск и фильтрацию книг.

3. Разработка кода: Предоставлены примеры SQL-запросов для работы с PostgreSQL и кода на C/C++ для сервера. Эти примеры включают в себя операции с базой данных и обработку ошибок.

4. Безопасность и обработка ошибок: Уделено внимание обработке ошибок и безопасности данных. Примеры кода включают в себя проверку статуса соединения с базой данных и проверку параметров запросов.

5. Управление книгами и пользователями: Описаны операции по добавлению, редактированию и удалению книг, а также управлению пользователями.

Преимущества и недостатки разработанной системы:

Преимущества:

1. Автоматизация: Система позволяет библиотеке автоматизировать учет книг и пользователей, что упрощает процессы аренды и возврата книг.

2. Удобство для пользователей: Пользователи могут легко находить книги, арендовать их и управлять своими профилями онлайн.

3. Безопасность: Применение хэширования паролей и обработка ошибок способствуют повышению безопасности данных.

4. Масштабируемость: Система позволяет добавлять новые книги и пользователей по мере роста библиотеки.

Недостатки:

1. Сложность разработки: Создание полноценной системы автоматизации библиотеки требует значительного времени и усилий, включая разработку клиентской части и дополнительных функций.

2. Требования к базе данных: Необходимость настройки и поддержания базы данных PostgreSQL может потребовать дополнительных знаний и усилий.

3. Интеграция и поддержка: Интеграция с существующей инфраструктурой библиотеки и поддержка системы требуют внимания и ресурсов.

4. Обучение пользователей: Пользователям может потребоваться время и обучение для эффективного использования системы.

В целом, разработанная система предоставляет значительные преимущества в управлении онлайн-библиотекой, но ее внедрение и поддержка требуют соответствующих ресурсов и внимания к деталям.

6. Список литературы:

1. Официальная документация PostgreSQL: Официальная документация PostgreSQL предоставляет обширную информацию о создании и управлении базами данных PostgreSQL, а также примеры SQL-запросов. <https://www.postgresql.org/docs/\>
2. Stack Overflow: [Stack Overflow PostgreSQL раздел](https://stackoverflow.com/questions/tagged/postgresql)
3. GitHub [Поиск проектов PostgreSQL на GitHub](https://github.com/topics/postgresql)
4. Reddit [Subreddit /r/PostgreSQL](https://www.reddit.com/r/PostgreSQL/)
5. PostgreSQL официальные форумы [PostgreSQL Community Forums](https://www.postgresql.org/community/)
6. DBA.StackExchange [DBA.StackExchange PostgreSQL раздел](https://dba.stackexchange.com/questions/tagged/postgresql)

7. Исходный код программного обеспечения:

SQL-запрос для создания таблиц в базе данных:

CREATE TABLE IF NOT EXISTS Books (

book\_id SERIAL PRIMARY KEY,

title VARCHAR(255) NOT NULL,

author VARCHAR(255),

genre VARCHAR(255),

publisher VARCHAR(255),

publication\_year INT,

isbn VARCHAR(13) UNIQUE,

description TEXT,

status VARCHAR(20) DEFAULT 'available'

);

CREATE TABLE IF NOT EXISTS Users (

user\_id SERIAL PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

registration\_date DATE,

role VARCHAR(20) DEFAULT 'user'

);

CREATE TABLE IF NOT EXISTS Rentals (

rental\_id SERIAL PRIMARY KEY,

user\_id INT REFERENCES Users(user\_id),

book\_id INT REFERENCES Books(book\_id),

rental\_date DATE,

return\_date DATE

);

Код на C/C++ для вставки новой книги (используя библиотеку libpq для PostgreSQL):

#include <stdio.h>

#include <stdlib.h>

#include <libpq-fe.h>

int main() {

PGconn \*conn;

PGresult \*res;

const char \*conninfo = "host=povt-cluster.tstu.tver.ru dbname=mydatabase user=myuser password=mypassword";

conn = PQconnectdb(conninfo);

if (PQstatus(conn) != CONNECTION\_OK) {

fprintf(stderr, "Connection to database failed: %s", PQerrorMessage(conn));

PQfinish(conn);

exit(1);

}

const char \*insert\_query = "INSERT INTO Books (title, author, genre, publisher, publication\_year, isbn, description) VALUES ($1, $2, $3, $4, $5, $6, $7)";

const char \*params[7] = {"Название книги", "Автор книги", "Жанр", "Издательство", "2022", "ISBN1234567890", "Описание книги"};

res = PQexecParams(conn, insert\_query, 7, NULL, params, NULL, NULL, 0);

if (PQresultStatus(res) != PGRES\_COMMAND\_OK) {

fprintf(stderr, "INSERT command failed: %s", PQerrorMessage(conn));

PQclear(res);

PQfinish(conn);

exit(1);

}

printf("New book added successfully\n");

PQclear(res);

PQfinish(conn);

return 0;

}

Это простые примеры для демонстрации работы с PostgreSQL на сервере с DNS именем "povt-cluster.tstu.tver.ru". Реальная разработка потребует более сложных сценариев и дополнительной обработки ошибок.