

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

Проектирование информационной системы

«Электронная библиотека»

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 441 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Нарожней Анастасии Дмитриевны

Научный руководитель
доцент, к.ф.-м.н.

Е.Ю. Крылова

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2020

Введение. В современном мире управление организацией является сложной задачей и не обходится без использования информационных технологий. Организация сталкивается с огромным объёмом данных, для обработки которых используются компьютерные технологии. Разработка и использование информационной системы является одним из способов улучшения функционирования организации.

Информационная система представляет собой хранилище информации, снабженное процедурами ввода, вывода, размещения и поиска информации. Её основной задачей является своевременное обеспечение клиентов необходимой информацией, поэтому основная работа информационной системы заключается в обслуживании двух встречных потоков информации: ввод новой информации и выдача текущей информации по запросам.

Данная бакалаврская работа посвящена проектированию информационной системы, обеспечивающей работу электронной библиотеки. Актуальность работы связана с важностью использования информационной системы для повышения эффективности работы организации и стабильности её функционирования, своевременного снабжения персонала и клиентов необходимой информацией.

Одним из главных этапов разработки информационной системы является определение требований, которым она должна удовлетворять. Для решения этой задачи в данной работе используются средства UML.

Для обеспечения хранения информации осуществляется разработка базы данных, предоставляющей собой поименованную совокупность данных, организованных по определённым принципам описания, хранения и манипулирования данными. Последним этапом является создание пользовательского интерфейса, необходимого для взаимодействия пользователя с базой данных.

Цель данной работы заключается в проектировании информационной системы, обеспечивающей работу электронной библиотеки. Для достижения данной цели необходимо осуществить проектирование информационной системы средствами UML и ER-диаграммы, разработать базу данных и пользовательский интерфейс.

Структура работы. Основная часть работы состоит из 4 разделов:

1. Описание предметной области;
2. Проектирование информационной системы средствами UML;
3. Проектирование базы данных для информационной системы;
4. Разработка графического интерфейса для информационной системы.

В первом разделе собрана информация об особенностях разрабатываемой информационной системы для выбранной предметной области (библиотека).

Городская библиотека занимается хранением книг и оказанием таких услуг, как выдача книг для чтения в читальном зале, выдача книг для чтения вне библиотеки. За нарушение правил эксплуатации библиотечного имущества применяются штрафные санкции в виде денежных сумм, устанавливаемых библиотекой. Поиск необходимой книги можно осуществлять по названию книги, произведению из книги (в случае, если книга — сборник произведений), автору, году создания, году публикации, жанру и т.д. Кроме того, можно осуществлять поиск клиентов по их персональным данным и/или читательскому билету. Также, можно осуществлять поиск работников, оказывавших услуги, по их персональным данным.

База данных предназначена для сотрудников городской библиотеки, занимающихся административными вопросами и вопросами клиентского обслуживания. Цель ее создания — упрощение процедуры поиска необходимой книги либо клиента (в случае, например, если требуется взыскать штраф), а также хранение всех проведенных в библиотеке операций, включая историю применения штрафных санкций по отношению к клиентам библиотеки (читателям).

В базе данных содержится информация о книгах, произведениях, их авторах, издательствах, жанрах, сотрудниках библиотеки (включая должность), клиентах библиотеки, читательских билетах, оказанных услугах и штрафных санкциях (включая тип возможных услуг и статус оказания услуги: в процессе либо завершена).

Второй раздел относится к проектированию, которое осуществляется с применением средств UML. Он разделяется на теоретическую часть, в которой описываются особенности языка UML и предоставляется необходимая

информация для осуществления работы. Остальные части раздела посвящены написанию основного и альтернативных потоков событий для варианта использования «Take book requirements» и построению следующих диаграмм: диаграмма вариантов использования (use case diagram), диаграмма последовательности (sequence diagram) и диаграмма деятельности (activity diagram). Код реализации всех диаграмм содержится в приложении А, изображения можно увидеть в самой работе.

Унифицированный язык объектно-ориентированного моделирования Unified Modeling Language (UML) – это семейство графических нотаций, в основе которого лежит единая модель.

UML часто ассоциируется с моделированием программных систем, построенных с использованием объектно-ориентированных (ОО) технологий, однако он имеет намного более широкое применение благодаря свойственной ему расширяемости.

Основная идея UML – возможность моделировать программное обеспечение и другие системы как наборы взаимодействующих объектов. Это подходит как для ОО программных систем и языков программирования, так и для бизнес-процессов и других прикладных задач.

Потоки событий – это текстовые описания пошагового выполнения прецедентов, они понятны не только разработчику, но и стороннему читателю. Потоки событий уточняют или детализируют последовательность действий, совершаемых системой при выполнении ее вариантов использования, а также описывают логику переходов через варианты использования.

Потоки событий бывают трех типов: основной, альтернативный и поток ошибок.

Основной (главный) поток описывает наилучший сценарий либо наиболее используемый путь исполнения прецедента.

Альтернативный поток специфицирует отклонения от основного потока, которые не рассматриваются как ошибочные.

Диаграмма прецедентов (use case diagram, или диаграмма вариантов использования) – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Диаграмма последовательности – диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления. На диаграмме последовательности объекты в основном представляют экземпляры класса или сущности, обладающие поведением.

Диаграммы деятельности – это технология, позволяющая описывать логику процедур, бизнес-процессы и потоки работ. Во многих случаях они напоминают блок-схемы, но принципиальная разница между диаграммами деятельности и нотацией блок-схем заключается в том, что диаграммы поддерживают параллельные процессы.

Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия, а дугами – переходы от одного состояния действия к другому. Все диаграммы построены с помощью Интернет-ресурса PlantUML.

Третий раздел относится к проектированию базы данных для информационной системы «Электронная библиотека». В теоретической части раздела описывается технология баз данных, приводятся основные термины и информация о выбранной СУБД для данной работы.

База данных – это именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Реляционная база данных – это совокупность взаимосвязанных нормализованных таблиц, отражающих некоторую предметную область.

Программное обеспечение, предназначенное для работы с БД, называется системой управления базами данных (СУБД). СУБД используются для упорядоченного хранения и обработки больших объемов информации.

В данной работе используется СУБД PostgreSQL. PostgreSQL – свободная объектно-реляционная система управления базами данных (СУБД).

Преимуществами PostgreSQL считаются:

1. высокопроизводительные и надёжные механизмы транзакций и репликации;
2. расширяемая система встроенных языков программирования: в стандартной поставке поддерживаются PL/pgSQL, PL/Perl, PL/Python и PL/Tcl; дополнительно можно использовать PL/Java, PL/PHP, PL/Py, PL/R,

PL/Ruby, PL/Scheme, PL/sh и PL/V8, а также имеется поддержка загрузки модулей расширения на языке C;

3. наследование;

4. возможность индексирования геометрических объектов и наличие базирующегося на ней расширения PostGIS;

5. встроенная поддержка слабоструктурированных данных в формате JSON с возможностью их индексации;

6. расширяемость (возможность создавать новые типы данных, типы индексов, языки программирования, модули расширения, подключать любые внешние источники данных).

Далее приводится теоретическая информация о моделировании предметной области с помощью ER-диаграмм и прилагается реализованная диаграмма для выбранной предметной области.

Одной из наиболее популярных семантических моделей данных на этапе инфологического проектирования является неформальная модель "Сущность-Связь" (Entity-Relationship – ER-модель). Моделирование предметной области базируется на использовании графических диаграмм, включающих небольшое число разнородных компонентов. В связи с наглядностью представления концептуальных схем баз данных (и не только их) ER-модели получили широкое распространение в CASE-системах (Computer Aided Software Engineering – программные средства, поддерживающие процессы автоматизированного проектирования баз данных, создания и сопровождения ПО (приложений) и баз данных, генерацию кода, тестирование, документирование и управление проектом).

Сущность (объект) – это реальный или представляемый объект предметной области, информация о котором должна сохраняться и быть доступна. Различают такие понятия, как тип сущности и экземпляр сущности. Понятие тип сущности относится к набору однородных предметов, событий, личностей, выступающих как единое целое. Экземпляр сущности относится к конкретной вещи в наборе. В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности – это имя типа, а не некоторого конкретного экземпляра этого типа.

Для идентификации конкретных экземпляров сущности используются атрибуты, выполняющие роль идентификаторов.

Атрибут – поименованная характеристика сущности, определяющая его свойства и принимающая значения из некоторого множества значений. Каждый атрибут обеспечивается именем, уникальным в пределах сущности.

Диапазон допустимых значений, которые может принимать атрибут, называется доменом.

Множество из одного или нескольких атрибутов, значения которых однозначно определяют каждый экземпляр сущности, называются идентификатором (ключом). Каждая сущность должен иметь хотя бы один идентификатор. Если идентификаторов несколько, один из них выбирается как привилегированный.

Связь (Relationship) – это поименованная графически изображаемая ассоциация, устанавливаемая между сущностями и представляющая собой абстракцию набора отношений, которые систематически возникают между различными видами предметов в реальном мире. При анализе связей между сущностями могут иметь место бинарные связи (между двумя сущностями или между сущностью и ей же самой - рекурсивная связь), транзитивные связи (между тремя сущностями), в общем случае – n-арные связи. В ER-диаграммах связь обозначается направленными ребрами с соответствующими надписями.

В оставшейся части раздела приводится описание таблиц базы данных проектируемой информационной системы. Исходные коды создания таблиц находятся в приложении Б.

Четвёртый раздел посвящён разработке графического интерфейса для информационной системы «Электронная библиотека». В нём собрана информация о выбранном языке программирования и предоставлены результаты разработки пользовательского интерфейса.

Графический интерфейс является важной составляющей для любой программной среды, ориентированной на конечного пользователя. Для разработки информационных систем применяется множество языков программирования, одним из которых является язык программирования TypeScript на основе JavaScript, представляющий собой средство разработки веб-приложений.

Характерными особенностями данного языка являются:

1. Четкое разделение интерфейса и логики. Код, который отвечает за пользовательский интерфейс, и код, реализующий логику приложения, четко разделены.

2. Слабое связывание. Внедрение зависимостей (Dependency Injection, DI) позволяет установить связь между компонентами и сервисами. С помощью связывания и событий есть возможность создавать слабо связанные компоненты, которые можно использовать повторно.

3. Определение изменений. Автоматический механизм определения изменений дает возможность отказаться от обновления интерфейса вручную, также предоставляя способ выполнить тонкую настройку этого процесса.

4. Инструментальные средства поддерживаются так же хорошо, как и для платформ Java и .NET. Анализатор кода TypeScript указывает на ошибки по мере набора текста, а инструмент для создания временных платформ и развертывания (Angular CLI) позволяет избегать написания стереотипного кода и сценариев конфигурирования.

5. Отрисовка на стороне сервера. Angular Universal превращает ваше приложение в HTML на этапе сборки кода офлайн. Данный код может быть использован для отрисовки на стороне сервера.

Создание графического интерфейса пользователя осуществляется на основе построенных UML-диаграмм, ER-диаграммы и базы данных, реализованной в СУБД PostgreSQL с помощью языка программирования TypeScript и библиотеки Angular.

Пользовательский интерфейс был реализован для пользования сотрудниками библиотеки в виде веб-приложения.

Заключение. В настоящее время работа любой организации не обходится без применения информационных технологий. Создание информационной системы, обеспечивающей оптимизацию работы электронной библиотеки, помогает упростить процессы сбора, обработки и передачи информации с помощью автоматизации процессов. Использование информационной системы необходимо для повышения эффективности работы организации и стабильности её функционирования.

Проектирование информационной системы «Электронная библиотека» состоит из нескольких взаимосвязанных частей. В ходе выполнения данной работы были изучены средства UML для создания модели бизнес-процессов. Были прописаны основной и альтернативные потоки событий варианта использования «Получение требований на книгу». С помощью Интернет-ресурса PlantUML были построены следующие диаграммы:

- диаграмма вариантов использования «Электронная библиотека»;
- диаграмма последовательности для варианта использования «Получение требований на книгу»;
- диаграмма деятельности для варианта использования «Получение требований на книгу».

Для дальнейшего построения базы данных была создана ER-диаграмма «Электронная библиотека», представляющая собой способ определения данных и отношений между ними.

В процессе работы при помощи СУБД PostgreSQL была разработана база данных для проектируемой информационной системы. Также был реализован графический интерфейс пользователя при помощи языка программирования TypeScript.

В ходе выполнения работы поставленная цель была достигнута, все задачи были реализованы в полном объёме.