

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра Математического и компьютерного моделирования

Разработка документо-ориентированного хранилища данных

для платной поликлиники

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 441 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Мартыненко Юлии Алексеевны

Научный руководитель
доцент, к.т.н.

И.А. Панкратов

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

Ю.А. Блинков

Саратов 2019

Введение. Актуальность темы исследования: успех предприятия в условиях современного конкурирующего рынка зависит от большого количества различных факторов. Одним из таких факторов является грамотное использование прикладных информационных технологий.

В современном обществе каждое предприятие сталкивается с такими трудностями, как обработка, поиск и хранение необходимой информации. С целью оптимизации данных процессов – они автоматизируются, т.е. создаются информационные системы. Информационные системы дают возможность хранить большие объемы данных, экономить время и выдавать нужную информацию в кратчайшие сроки в удобном для пользователя виде.

Функционирование платной поликлиники включает в себя хранение и поиск информации о сотрудниках, пациентах, информацию о заболеваниях, датах приема пациентов, а также информацию о стоимости и видах платных услуг.

Цель работы: создание автоматизированной информационной системы для платной поликлиники, а также изучение UML-моделирования, изучение и сравнение noSQL и SQL систем управления базами данных.

Объект исследования: информационная система платной поликлиники

Предмет исследования: использование программных продуктов для создания информационной системы платной поликлиники.

Научная новизна исследования заключается в следующем:

- рассмотрены все методологии проектирования информационных систем и на основе их преимуществ и недостатков выбрана наиболее актуальная;
- проведено поэтапное создание информационной системы для платной поликлиники.
- Сравнение и выбор более подходящей системы управления базами данных.

Практическая значимость данной работы состоит в том, что разработана рабочая информационная система, позволяющая обрабатывать, хранить и выдавать необходимую информацию, тем самым повышая эффективность работы предприятия.

Основное содержание работы. Выпускная работа состоит из введения, четырех разделов, заключения, списка использованных источников и приложений.

Введение содержит постановку задачи и краткое описание предметной области.

В первом разделе раскрывается понятие информационных систем, а также описываются две основные парадигмы в моделировании информационных систем, преимущества и недостатки каждой из них.

Информационная система (ИС) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Существуют два наиболее популярных подхода к анализу и проектированию информационных систем: структурный и объектно-ориентированный. В работе используется объектно-ориентированный подход. Выбор подхода определяет использование соответствующих средств для реализации информационной системы.

При разработке информационной системы приходится решать две основные задачи:

- задачу разработки базы данных, предназначенной для хранения информации;
- задачу разработки графического интерфейса пользователя клиентских приложений.

Второй раздел содержит проектирование информационной системы и информацию о предметной области системы, в нем выделена информация о

наиболее важных бизнес-процессах, описывается язык моделирования программных систем UML и строится диаграмма вариантов использования.

Любой процесс проектирования информационной системы начинается с определения цели проекта. На момент внедрения системы в производство и в течение всего времени ее эксплуатации проект должен обеспечивать: требуемую функциональность системы и степень адаптации к изменяющимся условиям ее функционирования;

- Безотказную работу системы в требуемом режиме, иными словами - готовность и доступность системы для обработки запросов пользователей;
- Простоту эксплуатации и поддержки системы;
- Требуемое время реакции системы на запрос;
- Требуемую пропускную способность системы;
- Необходимую безопасность от утечки данных и несанкционированного использования;
- Высокую производительность и, следовательно, эффективность системы.

Проектирование информационных систем охватывает три основные области:

- Проектирование объектов данных, которые будут реализованы в базе данных;
- Проектирование программ, экранных форм, отчетов, которые будут обеспечивать выполнение запросов к данным;
- Учет конкретной среды или технологии: конфигурация аппаратных средств, архитектура системы (файл-сервер или клиент-сервер), распределение обработки данных и т.д.

Одной из составляющих деятельности платной поликлиники является сбор и хранение информации. Обработываемая информация бывает следующих видов:

- Информация о мед. персонале. А именно, информация о докторах, уровне их квалификации, графике работы. Она включает в себя фамилию и имя, специальность, занимаемую должность (главный врач, заведующий отделением, врач и т.д.), дни приема.
- Информация о возможности оказания той или иной услуги, стоимость услуг. Она включает перечень оказываемых услуг на территории платной поликлиники и стоимость каждой услуги.
- Информация о пациентах платной поликлиники, даты их посещения, информация о виде заболевания. Данная информация включает в себя фамилию и имя пациента, контактный номер телефона, даты посещений пациентом данной платной поликлиники и цель обращения каждого посещения, обнаруженные заболевания при наличии.

Для описания модели информационной системы и ее компонентов в работе используется язык UML.

UML - универсальный язык моделирования, разработка которого началась с середины 90-х годов прошлого века с применением нескольких методов и нотаций описания информационных систем на базе объектно-ориентированной технологии, в настоящее время является общепринятым стандартом при документировании процесса разработки информационных систем и программного обеспечения.

Первоначально UML создавался для упрощения описания информационных систем на базе применения объектно-ориентированной технологии. Сейчас средствами языка можно описывать и программное обеспечение, оперирующее в основном потоками данных и методами их обработки (процедурное, алгоритмическое проектирование).

Для определения возможностей информационной системы была построена диаграмма прецедентов в соответствии с рисунком 1.1

Диаграмма вариантов использования для предметной области "Платная клиника"



Рисунок 1.1- Диаграмма прецедентов.

Для представления статической структуры модели системы была построена диаграмма классов в соответствии с рисунком 1.2

Диаграмма классов "Платная клиника"

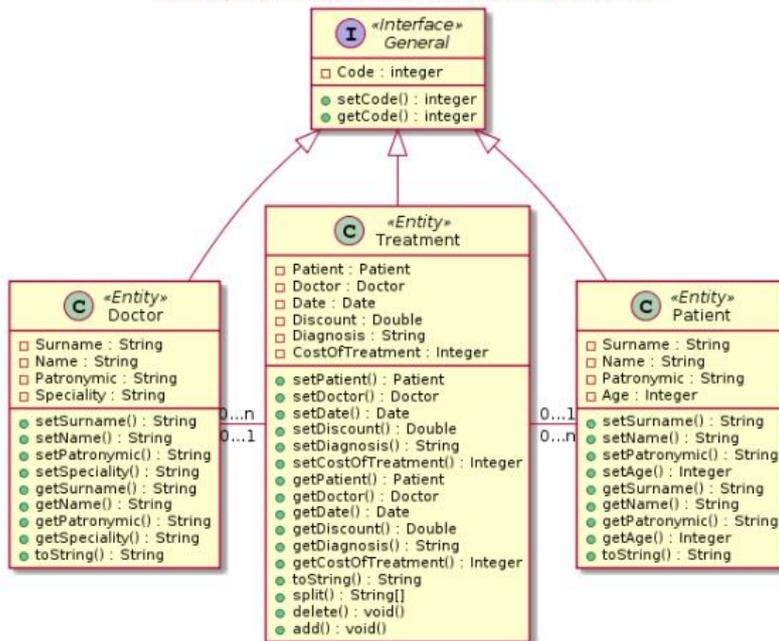


Рисунок 1.2- Диаграмма классов.

Третий раздел посвящен проектированию необходимой базы данных. В нем содержится описание реляционных и нереляционных баз данных, данный раздел содержит несколько запросов на `poSQL`.

База данных (БД) - организованная структура, предназначенная для хранения информации. Современные БД позволяют размещать в своих структурах как данные, так и методы (программный код), которые позволяют организовать взаимодействие с потребителем или программноаппаратными комплексами.

Системы управления базами данных (СУБД) - комплекс программных средств, предназначенных для создания структуры базы, наполнения ее содержанием, редактирования содержимого и визуализации информации (отображения данных в соответствии с заданным критерием, их упорядочения, оформления и последующей выдачи на устройство вывода или передачи по каналам связи).

В данной работе мы сравним реляционную модель и NoSQL.

Реляционные базы данных используют язык структурированных запросов (SQL) для того, чтобы обрабатывать данные и управлять ими. С одной стороны, это довольно удобно: SQL - один из наиболее разносторонних и общепотребимых вариантов, так что это безопасный выбор. Также этот язык подходит для сложных запросов. С другой стороны, с этим языком идут определенные ограничения. В SQL нужно использовать заданные наперед схемы и определять структуру данных перед началом работы с ней. К тому же, все данные должны иметь одну и ту же структуру.

Нереляционные базы данных, напротив, обладают гибкими схемами для неструктурированных данных. Они могут храниться по-разному: в колонках, документах, графах или в виде хранилища «ключ-значение». Эта гибкость позволяет:

- Можно создавать документы, не определяя их структуру заранее;
- Каждый документ может обладать собственной уникальной структурой;
- Синтаксис может различаться в разных базах данных;
- В процессе работы можно добавлять новые поля.

SQL БД имеют форму таблиц, а в NoSQL БД данные представляются в виде документов, пар «ключ-значение», графов или хранилищ wide-column. Из-за этого реляционные (SQL) базы лучше использовать для приложений, в которых

нужно переходить между несколькими записями, или для систем устаревшего вида, которые при создании имели реляционную структуру.

Раз уж мы разобрались, в чем состоит разница SQL и NoSQL, рассмотрим ключевые различия между ними на примере MySQL и MongoDB.

MySQL: SQL (реляционная) база данных.

- Сформированность: MySQL - хорошо известная база данных, то есть она обладает крупным комьюнити, широкими возможностями тестирования и стабильностью;
- Совместимость: MySQL доступна на всех основных платформах, включая Linux, Windows, Mac, BSD и Solaris. Также у нее есть адаптеры для таких языков, как Node.js, Ruby, C#, C++, Java, Perl, Python и PHP, то есть эта система не ограничена языком запросов SQL;
- Экономичность: Система является открытой и бесплатной;
- Воспроизводимость: Базу данных MySQL можно использовать на разных узлах, что позволяет снизить нагрузку и повысить масштабируемость и доступность приложения;
- Разделение данных: Несмотря на то что эту процедуру можно проводить на не всех SQL БД, серверы MySQL позволяют это сделать. Это не только экономично, но и может быть полезно для приложения.

MongoDB: NoSQL (нереляционная) база данных:

- Динамичность: Как говорилось ранее, динамическая схема гарантирует гибкость, позволяющую менять структуру без редактирования существующих данных;
- Масштабируемость: MongoDB можно масштабировать горизонтально, благодаря чему уменьшается нагрузка для бизнеса;
- Легкость в управлении: Для этой базы данных не требуется администратор. Так как она достаточно дружелюбна в отношении юзеров, воспользоваться ей могут как разработчики, так и администраторы;
- Скорость: Эта БД показывает отличные результаты в работе с короткими запросами;

- Гибкость: В MongoDB можно добавлять новые столбцы и поля, не влияя на уже существующие записи и производительность приложения.

В качестве СУБД для рассматриваемой базы данных была выбрана СУБД noSQL на платформе документоориентированной системы управления базами данных- MongoDB.

База данных MongoDB подходит для следующих применений:

- хранение и регистрация событий;
- системы управления документами и контентом;
- электронная коммерция;
- игры;
- данные мониторинга, датчиков;
- мобильные приложения;
- хранилище операционных данных веб-страниц (например, хранение комментариев, рейтингов, профилей пользователей, сеансы пользователей)

Для управления и администрирования системы базы данных могут быть установлены следующие команды:

- `mongo` — интерактивная оболочка, которая позволяет разработчикам и администраторам просматривать, вставлять, удалять и обновлять данные в своей базе данных, также позволяет настроить репликацию, сегментирование, отключить узлы, выполнить JavaScript или любые другие запросы к базе данных;
- `mongostat` — инструмент командной строки, который суммирует список статистических данных для исполняемого экземпляра MongoDB, это позволяет визуализировать количество вставок, обновлений, удалений, запросов и команд, а также ресурсопотребление экземпляра;
- `mongotop` — инструмент, который предоставляет метод для отслеживания времени, которое считывает или записывает данные в

экземпляре. Он также обеспечивает статистику на уровне каждой коллекции;

- `mongosniff` — инструмент, обеспечивающий перехват, сохранение и последующее воспроизведение команд, посылаемых в MongoDB;
- `mongoimport` и `mongoexport` — средства импорта и экспорта из JSON, CSV или TSV, поддерживается ряд других форматов;
- `mongodump` и `mongorestore` — инструменты создания резервной копии и восстановления базы данных из неё.

Четвертый раздел содержит в себе механизм соединения с сервером базы данных, проектирование модуля подключения к БД и непосредственно взаимодействие с самой базой данных.

В качестве интерфейса пользователя для разработки кода использовался пакет программ Haskell Platform. Haskell Platform это компилятор Haskell (GHC) и стандартная библиотека в полной комплектации. В комплекте с GHC идет программа `ghci`, запускающая компилятор в интерактивном режиме. Она часто используется в различных мануалах и облегчает отладку.

Заключение. Целью работы было изучение UML-моделирования, изучение в сравнении SQL и noSQL. Знакомство с программными продуктами для реализации информационной системы в рамках конкретной предметной области – платной поликлиники.

Были раскрыты понятия, охватывающие все аспекты проектирования ИС, и в результате получен актуальный продукт, предполагающий возможности добавления новых функций или изменения уже имеющихся при неизменных остальных функциональных частях.

Была разработана база данных, позволяющая удовлетворить информационные потребности выявленных пользователей. Был разработан механизм соединения с сервером базы данных и спроектирован модуль подключения к БД.