МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

Специальность 1-40 04 01 «Информатика и технологии программирования»

Отчет по преддипломной практике

на тему: «Программный комплекс автоматизации обслуживания жилого фонда студенческого общежития»

Исполнитель: студент гр. ИП-42 Пархоменко П.Л.

Руководитель от предприятия:

Макаревич В.Л.

Руководитель: преподаватель

Шибеко В.Н.

Дата проверки:

Дата допуска к защите:

Дата защиты:

Оценка работы:

Подписи членов комиссии

Гомель 2023

Оглавление

[**ВВЕДЕНИЕ** 3](#_Toc132230024)

[1 АНАЛИТИЧЕСКИЙ ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ И СРЕДСТВ АВТОМАТИЗАЦИИ ЖИЛОГО ФОНДА СТУДЕНЧЕСКОГО ОБЩЕЖИТИЯ 3](#_Toc132230025)

[1.1 Обзор существующих систем автоматизации 3](#_Toc132230026)

[1.2 Анализ разработки и проектирования веб приложений 4](#_Toc132230027)

[1.3 Анализ используемых технологий для реализации поставленной задачи 5](#_Toc132230028)

# **ВВЕДЕНИЕ**

…

1. АНАЛИТИЧЕСКИЙ ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ И СРЕДСТВ АВТОМАТИЗАЦИИ ЖИЛОГО ФОНДА СТУДЕНЧЕСКОГО ОБЩЕЖИТИЯ
   1. Обзор существующих систем автоматизации

Многие университеты имеют свои личные жилые фонды (общежития), не не многие пытаются их автоматизировать, ведь в них протекают много различных и сложных процессов. Например, управление жилплощадью, распределение мест, контроль доступа, управление уборкой и ремонтом, а также учет платежей за проживание и услуги - все эти процессы могут быть оптимизированы и автоматизированы с помощью специальных программных решений. Такая автоматизация помогает университетам повысить эффективность управления общежитиями, снизить затраты и улучшить качество обслуживания студентов.

В большинстве случаем университеты предоставляют общую информацию об общежитии, сформированную в виде одной онлайн страницы. Например, на сайте университета БГПУ, приведены следующие данные:

* адреса общежитий;
* фамилия, имя и отчество заведующего;
* фамилии, имена и отчества воспитателей и педагогов-организаторов,
* общее описание здания (количество мест, год постройки);
* наличие в общежитии общественных пространств (теннисные и тренажерные залы, актовый зал, кафе, прачечная, комната самоподготовки).

На сайте «Белорусского государственного университета» имеется информация, выраженная в виде рекомендаций и фотографий, например:

* какие бытовые приборы лучше всего взять с собой;
* как распланировать бюджет родителям.

Также существую другие средства автоматизация жилого фонда студенческого общежития. Эти системы включают в себя различные программные и аппаратные компоненты, такие как системы безопасности, системы управления доступом, системы учета и бронирования проживающих, системы мониторинга и управления энергопотреблением и т.д. Такие системы могут быть полезны для упрощения управления общежитием и улучшения комфорта проживания студентов. Например, системы управления доступом позволяют управлять доступом в здание и в отдельные комнаты, что обеспечивает безопасность и предотвращает несанкционированный доступ. Системы мониторинга и управления энергопотреблением позволяют управлять энергоресурсами и экономить деньги на коммунальных услугах.

Кроме того, системы учета и бронирования проживающих позволяют упростить процесс бронирования мест в общежитии и учета проживающих, что может значительно сократить время, затрачиваемое на административные процессы.

В целом, автоматизация студенческих общежитий может быть очень полезна для улучшения условий проживания студентов и оптимизации управления общежитием.

* 1. Анализ разработки и проектирования веб приложений

Разработка и проектирование веб-приложений – это процесс создания программного обеспечения, которое работает на сервере и обеспечивает взаимодействие с пользователем через веб-браузер. Веб-приложения могут быть использованы для различных целей, включая электронную коммерцию, социальные сети, онлайн-банкинг, различные средства автоматизации и т.д.

Для создания веб-приложения нужно выполнить несколько шагов. Сначала определяются требования к приложению, которые могут включать в себя функциональные и нефункциональные требования, дизайн интерфейса, спецификации базы данных и т.д. Затем происходит проектирование архитектуры приложения, разработка базы данных и программного кода, а также тестирование и развертывание приложения на сервер.

Разработка веб-приложений может включать в себя использование различных языков программирования, фреймворков, библиотек, систем управления базами данных и других инструментов. Например, для создания веб-приложения может использоваться язык программирования JavaScript, фреймворк React, база данных MySQL и сервер Apache.

Работа веб-приложения обычно происходит следующим образом: пользователь запрашивает определенную страницу через веб-браузер, сервер обрабатывает запрос и отправляет ответ обратно в браузер в виде HTML-страницы. Эта страница может содержать данные из базы данных, такие как пользовательский профиль или список продуктов, а также интерактивные элементы, такие как кнопки и формы, которые позволяют пользователю взаимодействовать с приложением.

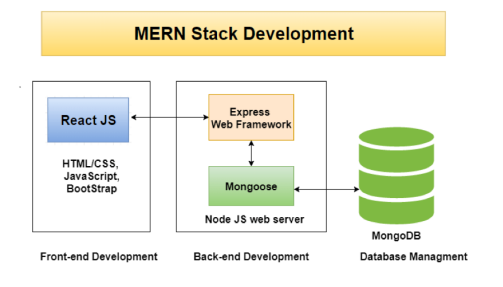
Кроме того, веб-приложения могут использовать клиент-серверную архитектуру, где клиентское приложение работает на стороне пользователя, а серверное приложение обрабатывает запросы и предоставляет данные. Например, веб-приложение может использовать AJAX для асинхронной передачи данных между клиентом и сервером, что улучшает пользовательский интерфейс и ускоряет процесс загрузки страниц.

В целом, веб-приложения могут быть очень полезны для обеспечения взаимодействия пользователей с программным обеспечением через веб-браузер.

Разработка и проектирование веб-приложений требует определенных навыков и знаний в области программирования, веб-технологий, дизайна пользовательского интерфейса и баз данных. Для успешной разработки веб-приложений необходимо понимание требований к приложению и его пользователей, умение разрабатывать эффективную архитектуру, использовать соответствующие инструменты и библиотеки, а также тестировать и развертывать приложение на сервер. Важным аспектом является также поддержка и обновление приложения после его запуска, а также обеспечение безопасности при обработке пользовательских данных и защите от взломов.

* 1. Анализ используемых технологий для реализации поставленной задачи

Веб-приложения могут быть написаны на разных языках программирования и фреймворках, но одним из наиболее популярных стеков технологий для разработки веб-приложений является стек MERN. MERN – это аббревиатура, состоящая из четырех популярных инструментов для веб-разработки: MongoDB, Express, React и Node.js.



В основе этих фреймворков и библиотек находится язык программирования JavaScript.

JavaScript – мультипарадигменный (с одновременным использованием множества парадигм) язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript (стандарт ECMA-262). Его обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты:

* динамическая типизация;
* слабая типизация;
* автоматическое управление памятью;
* прототипное программирование;
* функции как объекты первого класса.

JavaScript включает в себя объектную модель браузера – браузер-специфичная часть языка, являющаяся прослойкой между ядром и объектной моделью документа. Основное предназначение объектной модели браузера — управление окнами браузера и обеспечение их взаимодействия. Каждое из окон браузера представляется объектом window, центральным объектом DOM. Объектная модель браузера на данный момент не стандартизирована, однако спецификация находится в разработке WHATWG и W3C.

В JavaScript используется в AJAX, популярном подходе к построению интерактивных пользовательских интерфейсов веб-приложений, заключающемся в «фоновом» асинхронном обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью и интерфейс веб-приложения становится быстрее, чем это происходит при традиционном подходе (без применения AJAX).

MongoDB – это документо-ориентированная NoSQL база данных, которая используется для хранения, управления и обработки больших объемов данных. Она позволяет хранить данные в формате документов BSON (Binary JSON), которые имеют структуру, аналогичную формату JSON, но могут содержать бинарные данные, даты и другие типы данных.

MongoDB разрабатывалась с учетом масштабируемости и производительности, что делает ее особенно привлекательной для проектов, которые требуют быстрой и эффективной обработки больших объемов данных. Она имеет множество функций, которые делают ее полезной для многих типов приложений, включая:

1. Гибкий документо-ориентированный подход: MongoDB предоставляет гибкую модель документо-ориентированной базы данных, что позволяет хранить данные любого формата, включая структурированные, полуструктурированные и неструктурированные данные;
2. Высокая доступность: MongoDB предоставляет высокую доступность благодаря встроенному механизму репликации, который позволяет хранить несколько копий данных на разных серверах. Если один сервер не работает, MongoDB может быстро переключиться на другой сервер и продолжить работу;
3. масштабируемость: MongoDB легко масштабируется, что позволяет ей обрабатывать большие объемы данных. MongoDB поддерживает масштабирование горизонтальным и вертикальным способами;
4. мощный язык запросов: MongoDB предоставляет мощный язык запросов, который позволяет быстро и эффективно извлекать данные из базы данных;
5. индексы: MongoDB поддерживает множество типов индексов, которые позволяют ускорить поиск и фильтрацию данных;

MongoDB широко используется в различных областях, включая социальные сети, онлайн-магазины, приложения для анализа данных и многие другие. Ее популярность объясняется ее гибкостью, производительностью и масштабируемостью.

Express.js – это легковесный фреймворк для Node.js, который используется для разработки веб-приложений и API. Он предоставляет удобный и гибкий механизм для обработки запросов и ответов, маршрутизации и создания модульной структуры приложений.

Основные преимущества Express.js:

1. Удобство: Express.js облегчает разработку веб-приложений благодаря простому и интуитивно понятному API. Он позволяет быстро создавать маршруты, обрабатывать запросы и ответы, а также работать с различными middleware-пакетами;
2. Гибкость: Express.js позволяет разработчикам создавать приложения с различными функциональными возможностями. Он не навязывает строгую структуру приложения, что дает возможность гибко настраивать его под конкретные задачи;
3. Масштабируемость: Express.js позволяет легко масштабировать приложения, что особенно важно для больших и сложных проектов. Он поддерживает работу с кластерами и позволяет распределять нагрузку между несколькими серверами;
4. Поддержка middleware: Express.js предоставляет широкие возможности для работы с middleware, что позволяет улучшать функциональность приложения и повышать его безопасность. С помощью middleware можно добавлять авторизацию, обработку ошибок, логгирование и многое другое;
5. Большое сообщество: Express.js имеет большое сообщество разработчиков, которые создают и поддерживают множество пакетов и расширений. Это позволяет разработчикам быстро решать задачи и получать поддержку при возникновении проблем.

Express.js используется для создания различных типов приложений, включая API, веб-серверы, приложения для обработки данных и многие другие. Он позволяет быстро создавать и масштабировать приложения, обеспечивая при этом гибкость и удобство разработки.

React - это библиотека JavaScript, разработанная Facebook, которая используется для создания пользовательских интерфейсов. React использует декларативный подход для описания компонентов пользовательского интерфейса, что делает его более простым и понятным для разработчиков. Он позволяет создавать переиспользуемые компоненты, которые могут быть легко использованы для создания сложных пользовательских интерфейсов.

Основные преимущества React:

1. Декларативный подход: React использует декларативный подход для описания пользовательского интерфейса, что делает его более понятным для разработчиков. Он позволяет описывать, как должен выглядеть интерфейс, а не как его создать;
2. Переиспользуемые компоненты: React позволяет создавать переиспользуемые компоненты, которые могут быть легко использованы для создания сложных пользовательских интерфейсов. Это сокращает время разработки и улучшает качество кода;
3. Эффективный: React использует виртуальный DOM, который позволяет изменять только те элементы, которые действительно изменились. Это уменьшает количество дорогостоящих операций, связанных с обновлением интерфейса, и улучшает производительность приложения;
4. Большое сообщество: React имеет большое сообщество разработчиков, которые создают и поддерживают множество пакетов и расширений. Это позволяет разработчикам быстро решать задачи и получать поддержку при возникновении проблем;
5. Простота: React является относительно простым и понятным инструментом для создания пользовательских интерфейсов. Это делает его доступным для начинающих разработчиков и уменьшает время на обучение;

React используется для создания интерактивных пользовательских интерфейсов, включая веб-приложения, мобильные приложения, игры и многое другое. Он позволяет создавать переиспользуемые компоненты, которые могут быть легко использованы для создания сложных пользовательских интерфейсов.

Node.js - это среда выполнения JavaScript на стороне сервера, которая позволяет разрабатывать высокопроизводительные и масштабируемые веб-приложения. Она основана на движке V8, разработанном Google для браузера Chrome, и позволяет использовать JavaScript для создания приложений на серверной стороне.

Основные преимущества Node.js:

1. Высокая производительность: Node.js основан на движке V8, который обеспечивает быстрое выполнение JavaScript. Node.js также позволяет использовать асинхронное программирование, что улучшает производительность приложений;
2. Масштабируемость: Node.js позволяет создавать масштабируемые приложения с помощью механизма обработки запросов в нескольких потоках. Это позволяет распределять нагрузку на несколько серверов и обеспечивать высокую доступность приложения;
3. Широкие возможности: Node.js имеет большое количество библиотек и модулей, которые позволяют упростить разработку и расширить функциональность приложения. Это позволяет создавать приложения для различных сфер, включая веб-приложения, мобильные приложения, игры и многое другое;
4. Единый язык: Node.js использует JavaScript как единый язык для программирования на серверной и клиентской стороне, что упрощает разработку и повышает эффективность работы разработчика;
5. Активное сообщество: Node.js имеет большое сообщество разработчиков, которые создают и поддерживают множество пакетов и расширений. Это позволяет разработчикам быстро решать задачи и получать поддержку при возникновении проблем;

Node.js используется для создания различных приложений на серверной стороне, включая веб-приложения, микросервисы, API и многое другое. Он позволяет разработчикам создавать высокопроизводительные и масштабируемые приложения с использованием JavaScript на стороне сервера, что упрощает разработку и повышает эффективность работы разработчика.