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 дневник диабетика»

Исполнитель: студент гр. ИП-42 Суховенко Э.С.

Руководитель от предприятия:

Семченко А.С.

Руководитель: преподаватель

Прокопенко Д.В.

Дата проверки:

Дата допуска к защите:

Дата защиты:

Оценка работы:

Подписи членов комиссии

Гомель 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc132836587)

[1 АНАЛИТИЧЕСКИЙ ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ И СРЕДСТВ АВТОМАТИЗАЦИИ ДНЕВНИКА ДИАБЕТИКА 4](#_Toc132836588)

[1.1 Обзор существующих систем автоматизации 4](#_Toc132836589)

[1.2 Анализ разработки и проектирования веб приложений 5](#_Toc132836590)

[1.3 Анализ используемых технологий для реализации поставленной задачи 6](#_Toc132836591)

[1.4 Анализ инструментальных средств автоматизации разработки и тестирования 11](#_Toc132836592)

[1.5 Техническое задание для клиент-серверного программного продукта «Программный комплекс дневника диабетика» 16](#_Toc132836593)

[2 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И АЛГОРИТМЫ 17](#_Toc132836594)

[2.1 Анализ предметной области 17](#_Toc132836595)

[2.2 Функциональная модель программного комплекса 17](#_Toc132836596)

[2.3 Информационная модель программного комплекса 20](#_Toc132836597)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 26](#_Toc132836598)

# **ВВЕДЕНИЕ**

Автоматизация процессов позволяет увеличить эффективность работы, сократить время и затраты на выполнение задач, снизить вероятность ошибок и улучшить качество услуг. Также автоматизация помогает оптимизировать использование ресурсов, повысить скорость реакции на изменения внешней среды и упростить процессы для пользователя.

Целью дипломной работы является создание программного комплекса автоматизации дневника диабетика. Данный программный комплекс предназначен для упрощения процесса ведения дневника диабетика и улучшения контроля уровня глюкозы в крови у людей, страдающих диабетом. Он позволяет пользователям записывать и отслеживать данные о своем здоровье, такие как уровень глюкозы в крови, давление, вес, количество углеводов и прочие показатели, а также вносить комментарии и заметки о своем самочувствии и настроении. Благодаря этому, пользователи могут получать детальную информацию о своем здоровье и своих привычках, а также легко выявлять тенденции и изменения, что помогает им принимать более осознанные решения в отношении своего лечения и образа жизни. В целом, предназначение программного комплекса дневник диабетика заключается в том, чтобы облегчить процесс управления диабетом и помочь людям, страдающим этим заболеванием, более эффективно контролировать свое здоровье и улучшать свое качество жизни.

Задачами дипломной работы являются:

* изучение методик разработки клиент-серверных приложений;
* классификация ролей пользователей и их ролевые политики;
* изучение методов реализации серверной части для приложения;
* проектирование структуры приложения, базы данных для хранения информации, формирование пользовательских правил для доступа к ресурсам и функциям приложения;
* разработка программных модулей, обеспечивающих авторизацию и аутентификацию пользователей; работу с данными с помощью графического интерфейса;
* верификация и опытная эксплуатация разработанного программного обеспечения.

1. АНАЛИТИЧЕСКИЙ ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ И СРЕДСТВ АВТОМАТИЗАЦИИ ДНЕВНИКА ДИАБЕТИКА
   1. Обзор существующих систем автоматизации

Существует множество методов и средств автоматизации дневника диабетика, которые помогают облегчить процесс ведения дневника и улучшить контроль уровня глюкозы в крови у людей, страдающих диабетом.

Одним из наиболее распространенных методов является использование бумажного дневника диабетика, в котором пользователь вручную записывает данные о своем здоровье. Также существуют электронные дневники, которые могут быть загружены на персональный компьютер или мобильное устройство. Эти дневники могут предоставлять возможности для ввода данных, просмотра графиков и отчетов, а также синхронизации с другими устройствами и приложениями.

Некоторые современные медицинские устройства, такие как глюкометры и инсулиновые насосы, также могут автоматически собирать и передавать данные о здоровье пользователя в электронный дневник диабетика. Это помогает пользователям сохранять более точные и надежные записи о своем здоровье и легче выявлять тенденции и изменения в уровне глюкозы в крови.

Кроме бумажных носителей для ведения дневника диабетика существуют и программные продукты, например, DiaMeter - это онлайн-сервис для учета уровня глюкозы в крови и автоматического составления дневника диабетика. Этот сервис является эффективным инструментом для улучшения контроля уровня глюкозы в крови и помогает диабетикам принимать более осознанные решения в отношении своего лечения и образа жизни. DiaMeter позволяет пользователям вводить данные о своем уровне глюкозы в крови, а также другие важные медицинские данные, такие как давление, вес и уровень холестерина. Пользователи могут использовать этот сервис для создания дневника диабетика и отслеживания изменений в своем здоровье. Одной из ключевых особенностей DiaMeter является возможность получения персонализированных рекомендаций и советов по уходу за здоровьем от медицинских экспертов. Это помогает пользователям более осознанно подходить к своему лечению и принимать более эффективные решения в отношении своего здоровья. Кроме того, DiaMeter имеет функцию анализа данных и построения графиков, которые помогают пользователям выявлять тенденции и изменения в своем уровне глюкозы в крови. Это может быть особенно полезным для людей, которые страдают от диабета и нуждаются в более тщательном контроле своего здоровья. Сервис DiaMeter также имеет мобильное приложение, которое обеспечивает доступ к данным пользователя в любом месте и в любое время. Это делает использование сервиса более удобным и доступным для пользователей, которые часто находятся в движении. Однако, следует отметить, что DiaMeter платный сервис, и цены могут варьироваться в зависимости от уровня подписки и доступных функций. Кроме того, для использования некоторых функций сервиса, таких как интеграция с другими медицинскими устройствами, может потребоваться дополнительное оборудование.

Существует большое количество различных программных продуктов и мобильных приложений, предназначенных для учета данных о глюкозе в крови, приемах пищи и физических нагрузках. Однако, несмотря на широкий выбор существующих систем автоматизации дневника диабетика, многие из них имеют ограниченный функционал и не удовлетворяют потребностям пользователей в полной мере. Также многие системы имеют ограничения в совместимости с различными устройствами и операционными системами, что может быть неудобно для пользователей, использующих различные устройства.

Таким образом, разработка собственного программного комплекса дневника диабетика может быть эффективным решением для удовлетворения потребностей пользователей в полной и точной автоматизации учета данных о глюкозе в крови, приемах пищи и физических нагрузках, а также для обеспечения более высокого уровня безопасности и защиты данных пользователя.

* 1. Анализ разработки и проектирования веб приложений

Анализ разработки и проектирования веб-приложений является важным этапом в создании качественного и функционального программного продукта. Для успешной реализации проекта необходимо провести анализ требований к приложению, выбрать наиболее подходящий веб-фреймворк для его разработки, а также продумать архитектуру и функциональность приложения.

Веб-фреймворк (web framework) – это программное обеспечение, которое предоставляет разработчикам инструменты для создания веб-приложений. Он представляет собой набор библиотек, инструментов и стандартов, которые позволяют ускорить и упростить процесс разработки веб-приложений [1]. Веб-фреймворки позволяют разработчикам сконцентрироваться на бизнес-логике и функциональности приложения, в то время как большинство вспомогательных задач, таких как маршрутизация запросов, обработка ошибок, авторизация и аутентификация, обработка форм и другие, автоматизируются и становятся проще в использовании благодаря функциональности фреймворка. Веб-фреймворки могут использоваться с различными языками программирования, такими как Python, Ruby, JavaScript и другие. Они могут быть как "тяжелыми" фреймворками, которые предоставляют множество функций и инструментов, так и "легкими" фреймворками, которые предлагают минимальный набор инструментов и позволяют более гибко настраивать приложение.

В процессе проектирования следует учитывать следующие основные аспекты:

1. Требования к приложению. Необходимо определить основные функциональные и нефункциональные требования к приложению, включая возможность работы на различных устройствах и в различных операционных системах.
2. Выбор фреймворка. В зависимости от требований к приложению, следует выбрать наиболее подходящий фреймворк для его разработки. Например, для создания масштабируемых приложений может использоваться фреймворк Django, а для создания SPA-приложений - React.
3. Архитектура приложения. Необходимо продумать архитектуру приложения, включая выбор паттернов проектирования и организацию компонентов.
4. Функциональность приложения. Необходимо определить основные функции и возможности приложения, такие как ввод и хранение данных, взаимодействие с базой данных, а также взаимодействие с пользователем.
5. Тестирование приложения. После разработки приложения необходимо провести тестирование на соответствие требованиям, проверить его на уязвимости и обеспечить безопасность хранения данных пользователей.

Важно отметить, что разработка и проектирование веб-приложений является сложным процессом, который требует умения работать с различными технологиями и языками программирования, такими как HTML, CSS, JavaScript, Python и другие. Поэтому важно следить за последними тенденциями в области разработки и проектирования веб-приложений и постоянно улучшать свои знания и навыки.

Таким образом, анализ разработки и проектирования веб-приложений является важным шагом в создании качественного программного продукта. Он позволяет определить требования к приложению, выбрать наиболее подходящие технологии для его разработки, продумать архитектуру и функциональность приложения, а также провести тестирование на соответствие требованиям.

* 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 FERN. FERN – это аббревиатура, состоящая из четырех популярных инструментов для веб-разработки: Firebase, Express, React и Node.js. На рисунке 1.1 представлены компоненты основные компоненты стека FERN и архитектура их взаимодействия.



1. – Компоненты стека FERN

В основе этих фреймворков и библиотек находятся языки программирования JavaScript и TypeSctipt.

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) [2].

TypeScript - это язык программирования, который является расширением JavaScript и добавляет в него статическую типизацию. Он был разработан компанией Microsoft и позволяет улучшить производительность и надежность кода. TypeScript также поддерживает объектно-ориентированное программирование и может быть использован для создания сложных веб-приложений. Главное отличие между JavaScript и TypeScript заключается в том, что TypeScript предоставляет возможность определения типов переменных, функций и объектов на этапе компиляции, что упрощает обнаружение ошибок в коде и облегчает его поддержку. Также TypeScript позволяет использовать новые возможности языка JavaScript, которые еще не поддерживаются стандартом ECMAScript. Оба языка широко используются в веб-разработке и могут быть использованы для создания динамических веб-приложений. Однако, при использовании TypeScript возможно повышение качества кода и его поддержки, благодаря статической типизации и дополнительным возможностям языка [3].

Firebase - это платформа для разработки мобильных и веб-приложений, которая предоставляет инструменты для создания приложений без необходимости написания серверного кода и настройки инфраструктуры. Firebase предоставляет широкий спектр сервисов, включая базы данных в реальном времени, хранение файлов, аутентификацию, аналитику, отправку уведомлений и другие. Firebase использует облачную инфраструктуру Google и обеспечивает высокую масштабируемость и производительность для приложений любого размера. Она предоставляет возможность создавать многоуровневые приложения с поддержкой авторизации пользователей и безопасной передачи данных. Firebase может использоваться как в качестве backend для веб-приложений, так и для мобильных приложений на платформах Android и iOS. Она имеет открытое API, что позволяет легко интегрировать ее в различные приложения и сервисы.

Firebase предоставляет базу данных в реальном времени, которая является хранилищем данных на сервере, доступным для приложений. Эта база данных работает в режиме реального времени, что означает, что изменения, сделанные в базе данных, мгновенно отображаются в приложении, не требуя дополнительного обновления страницы. База данных имеет древовидную структуру данных и работает с JSON-объектами. Каждый элемент в базе данных представляет собой JSON-объект, который имеет уникальный ключ и может содержать несколько полей.

Firebase также обеспечивает синхронизацию данных между различными клиентами, что делает ее идеальным выбором для создания многопользовательских приложений, таких как чаты, онлайн-игры и другие. Также платформа предоставляет бесплатный тарифный план, который включает большинство основных функций, что делает его доступным для малых и средних проектов. Для более крупных проектов доступны платные тарифы с более широкими возможностями и функциональностью.

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 [4].

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 [5].

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.

* 1. Анализ инструментальных средств автоматизации разработки и тестирования

Для создания программного комплекса дневника диабетика будет использована среда разработки Webstorm от компании JetBrains. Для визуализации базы данных Firebase лучше всего подходит приложение FireAdmin. Хранить исходный код только на локальном компьютере плохая практика, поэтому будет создан удаленный репозиторий на GitHub. Для постройки различных диаграмм, которые позволят упростить разработку, а также предоставят полное понимание работы приложения, будет использоваться StarUml.

Рассмотрим подробнее каждый из этих инструментов.

Webstorm – это интегрированная среда разработки (IDE), которая предоставляет обширный функционал для разработки веб-приложений. Среда разработки позволяет работать с различными языками программирования, включая JavaScript, TypeScript, HTML, CSS, Node.js, Angular, React и другие. Webstorm включает в себя встроенные инструменты отладки, систему автодополнения кода, автоматическую проверку ошибок, систему контроля версий, анализаторы кода и многие другие полезные функции.

Webstorm обладает многоплатформенностью, что позволяет разрабатывать на разных операционных системах, включая Windows, macOS и Linux. Среда разработки также имеет мощную систему плагинов, которая позволяет расширять функционал IDE, добавляя поддержку новых языков программирования и инструментов. Она также имеет множество инструментов для работы с базами данных, включая поддержку MongoDB, MySQL, PostgreSQL, Oracle и других, обеспечивает интеграцию с браузерами для отладки веб-приложений в режиме реального времени.

Одним из основных преимуществ Webstorm является его эффективность и производительность. Среда разработки использует многопоточную архитектуру и оптимизированный механизм работы с памятью, что обеспечивает быстродействие и позволяет работать с большими проектами. Кроме того, Webstorm имеет обширную документацию и активное сообщество пользователей, которые создают полезные плагины, советы и обучающие ресурсы, что делает процесс разработки еще более комфортным и эффективным.

Webstorm – это мощная среда разработки, которая облегчает и ускоряет процесс создания высококачественных веб-приложений, идеально подходящая для опытных и начинающих разработчиков [6].

FireAdmin - это инструмент для визуализации и управления базой данных Firebase. Он предоставляет более продвинутые функции и интерфейс для управления данными в базе данных Firebase.

С помощью FireAdmin вы можете:

1. Просматривать и управлять всеми данными в вашей базе данных Firebase, включая коллекции, документы и поля;
2. Определять права доступа на уровне коллекций и документов, а также назначать роли и пользователей, которым разрешен доступ к определенным данным;
3. Управлять индексами, которые используются для быстрого доступа к данным в базе данных;
4. Использовать инструменты для экспорта и импорта данных в вашу базу данных;
5. Создавать пользовательские запросы, чтобы быстро найти и отфильтровать нужные данные.

FireAdmin предоставляет удобный интерфейс пользователя, который позволяет легко просматривать и управлять данными в вашей базе данных Firebase. Он также предоставляет инструменты для безопасности и экспорта/импорта данных, что делает его полезным инструментом для управления большими и сложными базами данных Firebase.

FireAdmin работает с базой данных Firebase в реальном времени, что позволяет быстро и легко обновлять данные в базе данных. Он также интегрируется с Firebase Auth, что позволяет управлять правами доступа к базе данных на основе аутентификации пользователей. FireAdmin доступен как веб-приложение, а также может быть установлен локально для работы с базой данных Firebase на локальной машине разработчика [7].

GitHub – это веб-сервис для хранения и совместной работы над Git-репозиториями. Это платформа, которая позволяет разработчикам хранить и совместно работать над кодом, отслеживать ошибки, создавать ветки, а также просматривать и редактировать код, управлять версиями, изменениями и запросами на слияние.

GitHub позволяет работать как индивидуальным разработчикам, так и командам, предоставляя возможность контролировать доступ к репозиторию и назначать роли для разных пользователей. Кроме того, GitHub интегрируется с другими сервисами, такими как Travis CI, Slack, JIRA и многие другие.

Репозиторий в GitHub – это хранилище для кода и других файлов, которые могут быть загружены и управляемы в Git. Репозиторий в GitHub позволяет хранить и управлять кодом в облаке, а также предоставляет множество функций для работы с кодом, включая возможность комментирования кода, управления задачами и многое другое. Если хранить код только на локальном компьютере, то это может привести к потере данных в случае сбоя жесткого диска или других проблем с компьютером. Кроме того, хранение кода на локальном компьютере не обеспечивает возможность совместной работы и синхронизации изменений между различными разработчиками.

Основные функции GitHub:

1. хранение и управление кодом. GitHub позволяет создавать удаленные репозитории для хранения кода, а также выполнять действия, такие как коммиты, создание веток и слияние веток;
2. отслеживание изменений. GitHub отслеживает и сохраняет историю изменений в коде, позволяя легко переключаться между различными версиями и возвращаться к предыдущим версиям, если это необходимо;
3. ведение проектов. GitHub предоставляет инструменты для ведения проектов, включая задачи и проблемы, обсуждение кода и код-ревью;
4. совместная работа. GitHub позволяет нескольким разработчикам работать с одним и тем же кодом одновременно, сливая свои изменения вместе;
5. интеграция с другими сервисами. GitHub интегрируется с другими сервисами, такими как Travis CI, Slack и JIRA, упрощая совместную работу и автоматизируя процессы.

Работа с открытым исходным кодом. GitHub является популярной платформой для работы с открытым исходным кодом, позволяя разработчикам сотрудничать над проектами, созданными сообществами, и вносить свой вклад в развитие открытых проектов.

Таким образом, GitHub – это мощный и удобный инструмент для хранения и совместной работы над кодом. Он предоставляет широкий спектр возможностей для управления кодом, удобный интерфейс для просмотра и редактирования кода, а также возможность совместной работы и синхронизации изменений между различными разработчиками.

StarUML – это приложение для создания UML-диаграмм, которое позволяет разработчикам создавать модели проектов, планировать архитектуру и дизайн приложений. StarUML имеет графический интерфейс пользователя, который предоставляет множество инструментов для создания и редактирования диаграмм, включая диаграммы классов, диаграммы последовательностей, диаграммы состояний, диаграммы активностей и многое другое.



Рисунок 1.2 – Интерфейс главного окна StarUml

В StarUML можно создавать различные типы диаграмм, включая:

1. Диаграммы классов - используются для описания структуры классов и их отношений;
2. Диаграммы последовательностей – используются для описания взаимодействия между объектами и процессов, происходящих во времени;
3. Диаграммы состояний – используются для описания жизненного цикла объекта и его состояний;
4. Диаграммы компонентов – используются для описания структуры и отношений между компонентами системы;
5. Диаграммы развертывания – используются для описания физического размещения компонентов и системы в целом;
6. Диаграммы активностей – используются для описания последовательности действий и процессов в системе.

На рисунке 1.2 в виде выпадающего меню представлен весь список возможных диаграмм.

StarUML поддерживает различные языки моделирования, включая UML 2.x, SysML, ERD, BPMN, SoaML и другие. Он также позволяет пользователям импортировать и экспортировать диаграммы в различных форматах файлов, включая XMI, SVG, PDF, PNG, JPEG и другие. Данное приложение имеет множество функций, которые облегчают создание и редактирование диаграмм, такие как автоматическое выравнивание элементов, возможность группировки элементов, подсветка синтаксиса, подсказки и многое другое. Кроме того, StarUML позволяет пользователю создавать свои собственные элементы, шаблоны и плагины, чтобы улучшить функциональность и адаптировать инструмент под свои потребности.

StarUML – это мощный и удобный инструмент для создания UML-диаграмм, который предоставляет широкий спектр возможностей для создания и редактирования диаграмм, а также импорта и экспорта диаграмм в различных форматах файлов. Он также имеет множество функций, которые облегчают создание и редактирование диаграмм, и может быть расширен за счет пользовательских элементов, шаблонов и плагинов [8].

Postman – это инструмент для тестирования API, который позволяет разработчикам быстро и удобно отправлять запросы к API, тестировать их и анализировать ответы.

С помощью Postman можно отправлять запросы на сервер и получать ответы в разных форматах, таких как JSON, XML, HTML и другие. Также в Postman есть множество функциональных возможностей, таких как автоматическое генерирование документации, управление авторизацией, создание и выполнение тестовых сценариев и другие. Он поддерживает различные типы запросов, включая GET, POST, PUT, DELETE и многие другие. Вы можете отправлять запросы с параметрами, заголовками и телом запроса в формате JSON или форм-данных.

Один из самых удобных и полезных аспектов Postman – это коллекции запросов, которые можно создавать и организовывать для более эффективного управления тестированием API. Коллекции могут быть экспортированы и импортированы в различных форматах, таких как JSON и XML.

Postman имеет множество преимуществ, которые делают его популярным среди разработчиков, вот некоторые из них:

1. Удобный интерфейс. Интерфейс Postman интуитивно понятен и удобен в использовании. Для отправки запроса не нужно писать код, все нужные функции находятся в графическом интерфейсе.
2. Простота в использовании. Postman не требует от разработчика особых навыков, чтобы начать использовать его. Для создания запросов нужно просто заполнить соответствующие поля.
3. Расширяемость. Postman можно легко расширять, используя плагины. Например, с помощью плагина можно добавить поддержку авторизации на основе OAuth или создать пользовательский набор инструментов для тестирования API.
4. Автоматизация. Postman позволяет автоматизировать тестирование API. Это особенно полезно при наличии большого количества запросов, которые нужно тестировать.
5. Множество функций. Postman имеет множество функций, таких как создание коллекций, генерация документации и выполнение тестовых сценариев, что делает его полезным инструментом для разработки и тестирования приложений.
6. Бесплатность. Postman имеет бесплатную версию, которая покрывает большинство потребностей разработчиков, а также платную версию с дополнительными функциями.

Postman является полезным и удобным инструментом для тестирования API, который может существенно ускорить процесс разработки приложений [9].

* 1. Техническое задание для клиент-серверного программного продукта «Программный комплекс дневника диабетика»

Цель разработки: разработать приложение, предназначенное для автоматизации дневника диабетика.

Данный программного комплекс предназначается для людей, больных диабетом.

Для пациентов приложение предусматривает следующие функции:

* ведение дневника уровня сахара в крови и принимаемых лекарств;
* получение рекомендаций и уведомлений по питанию и уровню сахара;
* просмотр истории показателей уровня сахара;
* создание целей;
* просмотр тематических материалов;
* взаимодействие с врачом и другими специалистами.

Для родственников пациентов приложение предусматривает следующие функции:

* просмотр информации о состоянии здоровья пациента;
* получение уведомлений о состоянии здоровья и уровне сахара;
* взаимодействие с пациентом и врачом при необходимости.

Для докторов приложение предусматривает следующие функции:

* просмотр истории показателей уровня сахара и анализ изменений в здоровье пациента;
* общение с пациентом;
* редактирование тематических материалов.

Приложение должно иметь следующую структуру и функциональность:

* работать как веб-приложение основанное на REST архитектуре;
* являться кроссплатформенным;
* являться кроссбраузерным;
* иметь принцип работы, аналогичный веб-приложениям (реализовывать REST архитектуру, быть доступным посредством веб-браузера, иметь пользовательский интерфейс).

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И АЛГОРИТМЫ
   1. Анализ предметной области

Анализ предметной области «Автоматизация обслуживания жилого фонда студенческого общежития» позволяет выделить следующие основные аспекты:

1. Учет и распределение комнат. Для автоматизации данного процесса необходимо иметь возможность вести учет свободных и занятых комнат, а также осуществлять распределение студентов в свободные комнаты.
2. Регистрация и учет студентов. Необходимо иметь возможность регистрировать студентов и вести учет их проживания в общежитии, а также осуществлять контроль за своевременной оплатой за проживание.
3. Предоставление различной информации. Необходимо иметь возможность создавать и просматривать новостные ленты для того.
4. Связь воспитателей и студентов. Необходимо иметь возможность связываться с студентом для решения вопросов, связанных с проживанием в общежитии.

Для автоматизации этих процессов может быть использована специализированная информационная система, разработанная с учетом конкретных потребностей студенческого общежития. Она будет включать в себя базу данных для хранения информации о студентах и комнатах, модули для управления ресурсами и контроля доступа, а также возможности для сотрудников общежития добавлять и изменять информацию о конкретном студенте.

Таким образом, разработка автоматизированной системы позволит упростить и ускорить процессы управления жилым фондом студенческого общежития, повысить эффективность использования ресурсов и улучшить качество обслуживания студентов. Благодаря использованию базы данных, все данные будут храниться в одном месте и сотрудники общежития смогут быстро получать необходимую информацию о студентах и комнатах. А модули управления ресурсами и контроля доступа позволят более точно контролировать использование ресурсов и обеспечить безопасность в общежитии. Разработка системы также может способствовать улучшению учебного процесса, так как облегчит жизнь студентам и позволит им сконцентрироваться на учебе, а не на решении бытовых вопросов.

* 1. Функциональная модель программного комплекса

Для того, чтобы создать качественное приложение, нужно четко понимать, какие роли у нас будут и какой функционал им будет доступен. Для этого нужно знать, какие объекты попадают в предметную область проектируемой системы и какие логические связи между ними существуют. Для формирования такого понимания используются логические модели предметной области. Целью построения логической модели является получение графического представления логической структуры исследуемой предметной области. Для стабильной работы программного обеспечения необходимо чёткое распределение на роли, на основе которых будут формироваться функции взаимодействия со внутренней средой приложения.

Актер – множество логически связанных ролей в UML, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Актером может быть человек или другая система, подсистема или класс, которые представляют нечто вне сущности.

Прецеденты представляют действия, выполняемые системой в интересах актеров. Проще говоря, прецедент – это описание последовательности действий (или нескольких последовательностей), которые выполняются системой и производят для отдельного актера видимый результат. Один актер может использовать несколько элементов прецедентов, и наоборот, один прецедент может иметь несколько актеров, использующих его. Каждый прецедент задает определенный путь использования системы. Набор всех прецедентов определяет полные функциональные возможности системы.

Приложение должно реализовывать три роли: комендант, воспитатель, студент.

Основные функции для роли «Комендант»:

* учет личного технического оборудования;
* учет количества свободных и занятых мест;
* возможность импортировать данные о студентах формата Excel;
* производить мониторинг оплаты за проживание.

Основные функции для роли «Студент»:

* просмотр новостей и событий;
* чат с воспитателем;
* создавать заявки на починку бытового оборудования.

Основные функции для роли «Воспитатель»:

* создание и ведение ленты новостей и событий;
* начисление баллов студентам;
* создание чата со студентом;
* создание общей беседы;
* возможность оставлять замечания в личном кабинете студента

В качестве среды для разработки и отображения функциональной структуры программы будет использована StarUml. На рисунке 2.2 представлена Use Case диаграмма приложения.



1. – Диаграмма прецедентов и актеров

Далее приведено описание прецедентов для роли «Комендант».

Прецендент «Учет личного техники» предназначен для просмотра и управления личным техническим оборудованием студента.

Прецендент «Загрузка данных из файла» предназначен для выгрузки личных данных студентов в базу данных.

Прецендент «Мониторинг оплаты общежития» предназначен для просмотра задолженности перед общежитием, а также для информирования студента.

Прецендент «Просмотр занятых и свободных мест» предназначен для мониторинга занятости комнат.

Далее приведено описание прецендентов для роли «Воспитатель».

Прецендент «Чат» предназначен для общения с отдельным студентом или с неким количеством.

Прецендент «Просмотр списка студентов» предназначен для просмотра информации о студентах в целом или более подробно об отдельном студенте.

Прецендент «Начисление баллов» предназначен для начисления баллов отдельному студенту.

Прецендент «Оставлять замечания». Данный прецендент необходим для того, чтобы воспитатель мог оповестить студента о каком-то совершенном нарушении в его личном кабинете.

Прецендент «Ведение ленты новостей» предназначен для просмотра и создания новых новостей.

Прецендент «Ведение ленты мероприятий» предназначен для просмотра и создания с целю оповещения новых событий.

Диаграмма прецендентов и актеров отображает функциональность системы с точки зрения пользователей. Она показывает взаимодействие между пользователем и системой в рамках конкретной задачи или сценария использования.

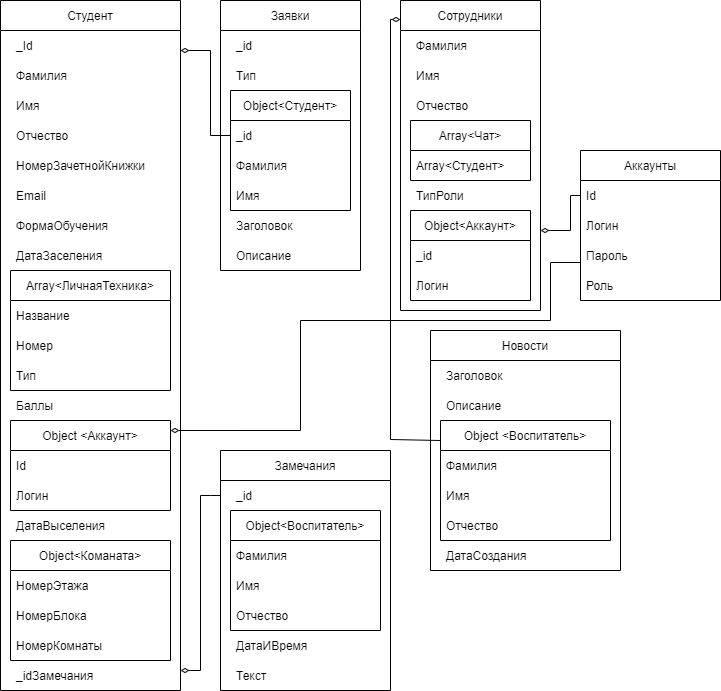
* 1. Информационная модель программного комплекса

На основе анализа предметной области определим набор коллекций и их свойства. В MongoDB коллекция представляет собой группу документов, которые хранятся в базе данных. Коллекции в MongoDB являются аналогом таблиц в реляционных базах данных. Они содержат документы в формате JSON, которые могут иметь различную структуру и не обязательно должны иметь одинаковые поля.

Для отображения информационной модели разрабатываемого продукта были выделены следующие коллекции:

* коллекция «Студент». Хранит в себе информацию о студентах, а также о комнате, где он живет;
* коллекция «Аккаунт». Хранит в себе данные для входа в личный аккаунт пользователя;
* коллекция «Сотрудники». Хранит в себе личные данные сотрудников общежития;
* коллекция «Новость». Хранит в себе весь список новостей;
* коллекция «Чат». Хранит список всех чатов.
* коллекция «Заявки». Хранит информацию о заявке студента на починку бытового оборудования.

На картинке 2.2 приведена схема базы данных.



1. – Схема базы данных

В таблицах 2.1 – 2.6 представлено подробное описание полей в коллекциях.

Таблица 2.1 – Коллекция «Студент»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Обязательное (required) | Уникальное (unique) |
| \_id | ObjectId | Да | Да |
| firstName | string | Да | Нет |

Продолжение таблицы 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Обязательное (required) | Уникальное (unique) |
| secondName | string | Да | Нет |
| middleName | string | Да | Нет |
| numberTest | string | Да | Да |
| email | string | Нет | Да |
| formEducation | union ("платное" | "бесплатное") | Да | Нет |
| dateEntry | string | Да | Нет |
| balls | number | Да | Нет |
| privateTechs | Array<Object> | Нет | Нет |
| privateTech.model | string | Нет | Нет |
| privateTech.number | string | Нет | Нет |
| privateTech.type | string | Нет | Нет |
| room | Object | Да | Нет |
| room.floor | number | Да | Нет |
| room.block | number | Да | Нет |
| room.apartament | number | Да | Нет |
| remarks | Array<Object> | Нет | Нет |
| remark.dateAndTime | string | Да | Нет |
| remark.header | string | Да | Нет |
| remark.status | string | Да | Нет |
| remark.text | string | Да | Нет |
| remark.mentor | Object | Да | Нет |
| remark.mentor.firstName | string | Да | Нет |
| remark.mentor.secondName | string | Да | Нет |
| remark.mentor.middleName | string | Да | Нет |
| account | Object | Да | Да |
| account.login | string | Да | Да |

В коллекции, описанной в таблице 2.1 поле formEducation имет специальный тип union, предоставляемый TypeScript. Данный тип представляет объединение нескольких строк, которые разрешено присваивать данной переменной.

Таблица 2.2 – Коллекция «Новости»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Обязательное (required) | Уникальное (unique) |
| \_id | ObjectId | Да | Да |
| header | string | Да | Нет |
| description | string | Да | Нет |
| mentor | Object | Да | Нет |
| mentor.firstName | string | Да | Нет |
| mentor.secondName | string | Да | Нет |
| mentor.middelName | string | Да | Нет |
| dateCreate | date | Да | Нет |

Как видно из таблицы 2.2, коллекция «Новости» хранит в себе данные воспитателя. Это необходимо для того, чтобы не осуществлять поиск по коллекции «Сотрудники» для установления данных создателя документа, что позволит нам сэкономить время загрузки данных.

Таблица 2.3 – Коллекция «Аккаунты»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Обязательное (required) | Уникальное (unique) |
| \_id | ObjectId | Да | Да |
| login | string | Да | Да |
| password | string | Да | Нет |

В таблице 2.3 приведено описание полей коллекции «Аккаунты». Данная коллекция нужна для быстрого поиска пользователя в системе во время его авторизации.

Таблица 2.4 – Коллекция «Чат»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Обязательное (required) | Уникальное (unique) |
| \_id | ObjectId | Да | Да |
| name | String | Да | Нет |
| messages | Array<Object > | Да | Нет |
| messages.who | Object | Да | Нет |
| messages.who.firstName | String | Да | Нет |
| messages.who.secondName | String | Да | Нет |
| messages.who.middleName | String | Да | Нет |
| messages.when | Date | Да | Нет |
| messages.message | string | Да | Нет |

Таблица 2.5 – Коллекция «Сотрудники»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Обязательное (required) | Уникальное (unique) |
| \_id | ObjectId | Да | Да |
| firstName | String | Да | Нет |
| secondName | string | Да | Нет |
| middleName | String | Да | Нет |
| Role | Union ("mentor" | "main") | Да | Нет |
| chats | Array<ObjectId> | Нет | Нет |

В таблице 2.4 описаны поля для коллекции «Сотрудники». Все поля, кроме chats, обязательные. Поле chats хранит массив типа ObjectId. Данный массив заполняется id из коллекции «Чат», описанной в таблице 2.4.

Таблица 2.6 – Коллекция «Заявки»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Обязательное (required) | Уникальное (unique) |
| \_id | ObjectId | Да | Да |
| type | String | Да | Нет |
| student | Object | Да | Нет |
| student.firstName | string | Да | Нет |

Продолжение таблицы 2.6

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Обязательное (required) | Уникальное (unique) |
| student.secondName | String | Да | Нет |
| student.room | number | Да | Нет |
| student.room.floor | Number | Да | Нет |
| student.room.block | Number | Да | Нет |
| student.room.apartament | Number | Да | Нет |
| header | String | Да | Нет |
| description | string | Да | Нет |

Как можно заметить, каждая коллекция в MongoDB может иметь некоторые поля, которые могут быть обязательными или необязательными. Кроме того, поля могут быть уникальными или неуникальными в пределах коллекции. Поле, которое является уникальным, не позволяет вставлять документ с таким же значением, как уже имеющийся в коллекции. Поле, которое является обязательным, требует, чтобы его значение было указано при вставке документа. Если обязательное поле не указано, то операция вставки завершится неудачей. Коллекция также может иметь индексы, которые помогают ускорить выполнение запросов к базе данных. В целом, использование MongoDB позволяет эффективно хранить и управлять данными, предоставляя удобный интерфейс для доступа к ним.

# **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Проектирование и разработка web-приложений. [Электронный ресурс] – Режим доступа: <https://urait.ru/book/proektirovanie-i-razrabotka-web-prilozheniy-512113>.
2. Что такое JavaScaript? [Электронный ресурс] – Режим доступа: <https://developer.mozilla.org/ru/docs/Learn/JavaScript/First_steps/What_is_JavaScript>.
3. База данных MongoDb. [Электронный ресурс] – Режим доступа: <https://itglobal.com/ru-by/company/glossary/mongodb/>.
4. Express и NodeJs. [Электронный ресурс] – Режим доступа: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Express_Nodejs>.
5. Фреймворк React. [Электронный ресурс] – Режим доступа: <https://metanit.com/web/react/1.1.php>.
6. IDE WebStorm. [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/WebStorm#:~:text=JetBrains%20WebStorm%20%20интегрированная%20среда%20разработки,на%20основе%20платформы%20IntelliJ%20IDEA.&text=WebStorm%20обеспечивает%20автодополнение%2C%20анализ%20кода,интеграцию%20с%20системами%20управления%20версиями>.
7. Визуальная оболочка MongoDb Compass. [Электронный ресурс] – Режим доступа: <https://www.hostland.ru/articles/mongodb-compass#:~ :text=GUI%20MongoDB%20Compass&text=Это%20удобный%20Клиент%2C%20разработанный%20MongoDB,ОС%20Linux%2C%20Mac%20и%20Windows>.
8. StarUml. Описание и как он устроен. [Электронный ресурс] – Режим доступа: <https://soware.ru/products/staruml>.
9. Как работает Postman. Общие принципы [Электронный ресурс] – Режим доступа: https://blog.skillfactory.ru/glossary/postman/.