ВВЕДЕНИЕ

1 АНАЛИЗ ПРЕДПРИЯТИЯ ООО «ИМЦ»

Специализацией компании «Информационно-медицинский центр» (ИМЦ) является информатизация здравоохранения, включая комплексные решения для медицинских организаций, органов управления здравоохранением и территориальных фондов ОМС. Автоматизируется также финансово-хозяйственная работа бюджетных учреждений, деятельность ВУЗов и библиотек.

ООО «ИМЦ» осуществляет разработку компьютерного программного обеспечения (62.01).

С момента организации предприятие стремительно развивается, специализируясь на разработке программных продуктов являющимся комплексными решениями для медицинских организаций, органов управления здравоохранением и территориальных фондов ОМС.

Постоянно улучшающий практический опыт разработки, внедрения и сопровождения информационных систем, хорошее знание особенностей действующих нормативных документов, положений, стандартов и технологий.

ООО «ИМЦ» имеет возможность адаптировать систему под нужды организации, их эволюционное развитие в зависимости от совершенствования технической инфраструктуры.

1.1 Задачи и функции предприятия ООО «ИМЦ»

Целью предприятия ООО «ИМЦ» является Разработка компьютерного программного обеспечения для выполнения требований потребителя.

Задачами предприятия ООО «ИМЦ» являются:

- получение дохода владельцам ООО «ИМЦ»;

- обеспечение потребителей программным обеспечением в соответствии с договорами;

- обеспечение персонала предприятия заработной платой, нормальными условиями труда и возможностью профессионального роста;

- создание рабочих мест для населения, в пределах муниципального округа.

К основному виду деятельности ООО «ИМЦ» относится: Разработка компьютерного программного обеспечения.

К дополнительным видам деятельности ООО «ИМЦ» относятся:

- торговля оптовая компьютерами, периферийными устройствами к компьютерам и программным обеспечением;

- торговля оптовая неспециализированная;

- торговля розничная компьютерами, периферийными устройствами к ним и программным обеспечением в специализированных магазинах;

- деятельность консультативная и работы в области компьютерных технологий;

- деятельность, связанная с использованием вычислительной техники и информационных технологий, прочая;

- деятельность по обработке данных, предоставление услуг по размещению информации и связанная с этим деятельность;

- деятельность по созданию и использованию баз данных и информационных ресурсов;

- деятельность по оказанию консультационных и информационных услуг;

- научные исследования и разработки в области естественных и технических наук прочие;

- деятельность по предоставлению прочих вспомогательных услуг для бизнеса, не включенная в другие группировки;

- ремонт компьютеров и периферийного компьютерного оборудования.

Важным процессом организации ООО «ИМЦ» является «обработка обращений клиентов». Благодаря этому процессу клиенты могут оставлять свои требования, пожелания и информировать о найденных ошибках программного продукта.

Для более наглядного представления процесса «обработка обращении клиентов» составлена функциональная блок-схема, представленная на рисунке 1.



Рисунок 1 - Функциональная блок-схема

Основными документами, регламентирующими предпринимательскую деятельность ООО «ИМЦ» являются:

* Конституция РФ;
* Федеральный закон от 08.02.1998 N 14-ФЗ (ред. от 23.04.2018) «Об обществах с ограниченной ответственностью»;
* Лицензия ФСТЭК на деятельность по технической защите конфиденциальной информации;
* Лицензия ФСБ на осуществление деятельности по разработке, производству, распространению, техническому обслуживанию шифровальных (криптографических) средств.

1.3 Моделирование бизнес-процессов предприятия ООО «ИМЦ»

Для разработки моделей процессов информационно-технической деятельности отдела техническая поддержка ООО «ИМЦ» будет использоваться задача – обработка обращении от клиентов.

Для разработки модели процесса «обработка обращении от клиентов» будет использоваться методология IDEF0.

IDEF0 - это методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов.

Контекстная диаграмма процесса «обработка обращении от клиентов» представлена на рисунке 3.



Рисунок 3 - Контекстная диаграмма процесса «обработка обращении от клиентов»

Функциональный блок контекстной диаграммы подвергается декомпозиции. Получившаяся диаграмма содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы, и называется дочерней по отношению к нему.

Для более детального изучения процесса, проведём декомпозицию контекстной диаграммы процесса «обработка обращении от клиентов», которая представлена на рисунке 4.



Рисунок 4 - Декомпозиция контекстной диаграммы процесса «обработка обращении от клиентов»

Клиент, обращается к технической поддержки из различных источников передавая свое обращения и персональные данные необходимые для исправления программного модуля после чего сотрудник технической поддержки на основе договора проверяет есть ли у клиента сопровождение технической поддержкой и изучает другую информацию необходимую для принятия обращения.

Далее сотрудник технической поддержки на основе личного устава организации требуется обработать и распределить обращение клиента.

Далее, происходит выполнения обращения либо разработчиком, либо сотрудником технической поддержки.

После выполнение обращения, сотрудник технической поддержки связаться с клиентом что бы оповестить его о проделанных работах.

В этом процессе можно выделить такие проблемы, как:

- техническая поддержка принимает обращения от клиентов через множество разных источник: социальные сети, Email, по телефону и СМС;

- техническая поддержка обрабатывают задачу в систему;

- после выполнения обращения клиента, техническая поддержка оповещает клиентов о выполненной работе.

Для решения этих проблем нужно провести оптимизацию данного процесса с помощью создание информационной системы. Создание информационной системы для ООО «ИМЦ» позволит: сократить нагрузку сотрудников технической поддержки сократив трудозатраты этого отдела и увеличить эффективность работы технической поддержки.

Контекстная диаграмма модели TO-BE, с использованием возможностей процесса «обработка обращении от клиентов» представлена на рисунке 5.



Рисунок 5 - Контекстная диаграмма модели TO-BE процесса «обработка обращении от клиентов»

Декомпозиция контекстной диаграммы для модели TO-BE, процесса «обработка обращении от клиентов» представлена на рисунке 6.



Рисунок 6 – Декомпозиция контекстная диаграмма модели TO-BE процесса «обработка обращении от клиентов»

Клиент, заходит в ИС и заполняет обращение, система на основе категории определяет на какого сотрудника поставить эту задачу и выполняет оформление обращения так же ограничивает доступ, что только клиент с возможностью обслуживание технической поддержки имеет доступ к ИС. После чего техническая поддержка или разработчик в зависимости от обращения получает уведомление об оповещение и приступает к его выполнению. После выполнения задачи используя информационную систему выполняющий задачу меняют статус задаче на проверку, и система совершает уведомление клиент о выполнении его обращении.

Данное изменение процесса приводит к созданию одного источника получения данных от клиента и уменьшает нагрузку отдела технической поддержки на счет автоматизации процесса посредствам ИС.

DFD — общепринятое сокращение от англ. data flow diagrams — диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

На рисунке 7 представлено диаграмма потоков данных в нотации DFD.



Рисунок 7 - Контекстная диаграмма потоков данных.

На диаграмме, представленной на рисунке 7 выделены 3 сущности: клиент, разработчик, техническая поддержка. В процесс «обработка обращения клиента» клиент передает свое обращение и данные на изменение программного продукта из разных источников, данное обращение передается в техническую поддержку на оформление задачи и распределение обращения. Далее разработчику приходит техническое задание на основе которого он разрабатывает программные модуль, который передается клиенту, и сотрудник технической поддержки передает уведомлением о проделанных работ клиенту.

Декомпозиция процесса «Обработка обращения клиента» приведена на рисунке 8.

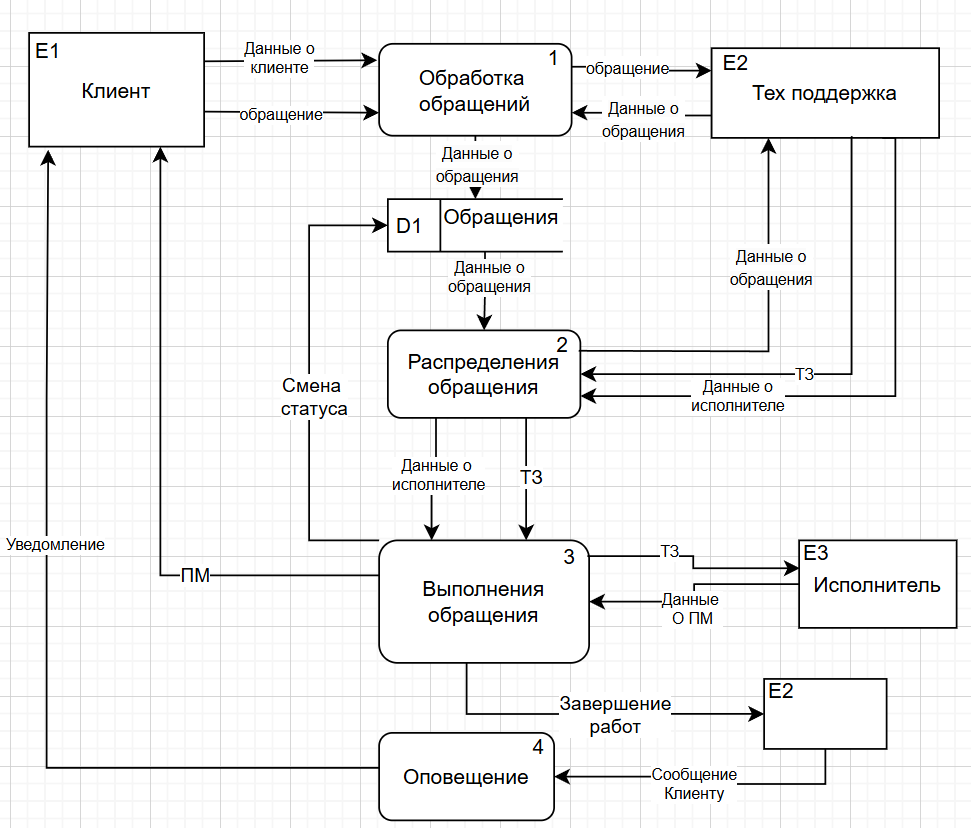


Рисунок 8 - Декомпозиция процесса «Обработка обращения клиента»

После внедрения ИС в процесс «обработка обращения клиента» клиент передает свое обращение на изменение программного продукта и данные для авторизации в ИС, которая обрабатывает эти данные и передает их в техническую поддержку, далее происходит распределение обращения. После разработчику приходит техническое задание на основе которого он разрабатывает программные модуль, который передается клиенту и ИС уведомлением о проделанных работ.

На рисунке 9 представлена диаграмма потоков данных после внедрения ИС для процесса «обработка обращения клиента».



Рисунок 9 - Диаграмма потоков данных после внедрения ИС

На рисунке 10 представлена декомпозиция диаграмма потоков данных после внедрения ИС для процесса «обработка обращения клиента».



Рисунок 10 - Диаграмма декомпозиция потоков данных после внедрения ИС

IDEF3 — методология моделирования и стандарт документирования процессов, происходящих в системе. Метод документирования технологических процессов представляет собой механизм документирования и сбора информации о процессах. IDEF3 показывает причинно-следственные связи между ситуациями и событиями в понятной эксперту форме, используя структурный метод выражения знаний о том, как функционирует система, процесс или предприятие.

На рисунке представлено 12 контекстная диаграмма процесса «обработка обращения клиента» по методологии IDEF3.



Рисунок 12 - Контекстная диаграмма процесса по методологии IDEF3

На рисунке представлено 13 декомпозиция процесса «обработка обращения клиента» с стороны клиента по методологии IDEF3.



Рисунок 13 - Декомпозиция процесса по методологии IDEF3

Клиент заходит в ИС систему вводит свой логин и пароль, после чего происходит авторизация, которая подтягивает данные о клиенте, после чего клиенту требуется заполнить форму с обращением в ИС и после выполнения обращения клиента придет уведомление клиенту о проделанных работах.

На этом этапе, смоделирован процесс, происходящий в системе, с применением методологии IDEF3 позволяет провести декомпозицию блока так, чтобы представить процесс наиболее информативно.

1.3. Классификация программных продуктов в области технической поддержки

1.4. Планирование и определение затрат на разработку ПМ

1.5 Техническое задание на разработку ПМ

1.3.1 Введение

1.3.1.1 Назначение

Разрабатываемый продукт предназначен для целевой аудитории предоставляя возможность взаимодействия с врачами. Предоставляя возможность записи пациента к врачу, просмотр истории болезни, просмотр истории записей, и так далее. в любой момент времени.

1.3.1.2 Обзор продукта

Продукт представляет собой 2 веб приложении front-end и back-end, которые будут общаться между собой по протоколу https, front-end будет передать на back-end данные что делает пользователей, после чего back-end будет обрабатывать эти данные и возвращать front-end ответ.

Продукт должен предоставлять врачам и клиентов возможность удобнее взаимодействовать между собой, давай возможность пациентом в любое время записаться на прием или получить интересующую информацию.

1.3.1.3 Функции продукта

Продукт предоставляет возможности:

* просмотреть свою историю болезни;
* создать пользователя на основе полиса;
* просмотреть историю приемов;
* просмотреть список врачей в поликлинике;
* возможность записаться к врачу на конкретную дату и время;
* добавить врачу запись в историю болезни.

1.3.1.4 Термины и определения

Back-end –работает с компонентами на стороне сервера и манипулирует данными, обеспечивая их хранение, обработку и получение.

front-end – это презентационная часть web-приложений, информационной или программной системы, её пользовательский интерфейс и связанные с ним компоненты.

Протокол https - протокол, который обеспечивает целостность и конфиденциальность данных при их передаче между сайтом и устройством пользователя.

Веб приложение - это компьютерная программа, которая для выполнения своих функций использует веб-браузеры и веб-технологии в сети Интернет.

1.3.2 Детальные требования

1.3.2.1 Требование к юзабилити

Существует 3 ключевых критериев, характеризующих уровень юзабилити ресурса:

* простота. Насколько легко новому посетителю ориентироваться в пространстве и выполнять элементарные действия;
* эффективность. Сколько времени требуется на достижение цели;
* запоминаемость. Как быстро пользователь адаптируется к нюансам работы с сайтом, сможет ли оперативно разобраться при повторных посещениях.

1.3.2.2 Требование к производительности

Требования к производительности:

* быстродействие приложения;
* минимальное ожидание до 1 клика.

1.3.2.3 Требование к логической структуре БД

Требование к логической структуре БД:

* база данных должна включать совокупность информационных объектов различного типа;
* каждый информационный объект может иметь набор атрибутов и связей с другими информационными объектами;
* информационные объекты связаны друг с другом иерархически;
* при внесении изменений структура БД должна обеспечивать автоматическое изменение всех объектов и связей, затрагиваемых таким изменением.

1.3.2.4 Ограничения проектирования

Ограничения проектирования:

* работает только в браузере;
* требуется доступа к интернету;
* работает по http протоколу;
* срок 1-2 месяца разработки.

В данной главе были достигнуты следующие задачи произведен анализ предметной области «Поликлиника», произведен анализ целевой аудитории и составлен портрет пользователей, разработано техническое задание на разрабатываемый продукт и определены требования к разрабатываемому продукту.

2. ПРОЕКТИРОВАНИЕ ИС

2.1. Проектирование ИС с использованием средств объектного   
 моделирования.

Информационная система будет разработана для информационно-технической деятельности отдела технической поддержки ООО «ИМЦ» для оптимизации процесса – «обработка обращения клиента».

Задача информационной системы - систематизировать поступающие обращения клиентов и оптимизировать процесс их обработки.

Входными данными информационной системы является данные и обращения клиентов.

Выходные данные информационной системы является выполненное обращение клиента.

Сущности информационной системы:

* клиент;
* исполнитель;
* обращение(задача);
* организация клиента.

Информационная система выполняет следующие функции:

* оформление обращений клиентов;
* добавление клиента;
* получение списка клиентов;
* получение информации об клиенте;
* добавление исполнителя;
* редактирование исполнителя;
* получение списка исполнителей;
* добавление организации клиента;
* получение списка организаций клиентов;
* редактирование организации;
* возможность авторизации клиента и исполнителя;
* возможность аутентификации клиента и исполнителя;
* возможность просмотра всех обращений;
* создание обращения;
* изменение статуса обращения;
* фильтрация по обращениям;
* изменение обращения;
* валидация изменения статусов обращения.

Принципиальное различие между структурным и объектно-ориентированным (ОО) подходом заключается в способе декомпозиции системы. ОО подход использует объектную декомпозицию, при этом статическая структура системы описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщений между объектами.

UML предоставляет средства для создания визуальных моделей, которые единообразно понимаются всеми разработчиками, вовлеченными в проект, и являются средством коммуникации в рамках проекта. Диаграмма в UML - это графическое представление набора элементов. Диаграммы рисуют для визуализации системы с разных точек зрения. При визуальном моделировании на UML используются восемь видов диаграмм, каждая из которых может содержать элементы определенного типа.

Выбор объектно-ориентированного подхода вместо структурного обоснован следующими причинами:

* объектно-ориентированные системы лучше моделируют предметную область. Они проще адаптируются к изменяющимся условиям, легче изменяются, устойчивее и позволяют создавать более крупные проекты;
* объектная декомпозиция уменьшает размер программных систем. Это достигается за счёт повторного использования общих механизмов, что приводит к существенной экономии выразительных средств;
* объектно-ориентированные системы снижают риск при создании сложной программной системы. Она развивается из меньших систем, в которых уже уверены;
* использование объектного подхода повышает уровень унификации разработки и пригодность для повторного использования.

Диаграмма вариантов использования — это диаграмма, на которой изображаются отношения между актерами и вариантами использования.

Ниже на рисунке 1 представлена диаграмма вариантов использования.



Рисунок 1 - Диаграмма вариантов использования

Диаграмма последовательности — это UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание, деятельность, уничтожение) и взаимодействие акторов (действующих лиц) информационной системы в рамках прецедента.

На рисунке 2 представлена диаграмма последовательности.



Рисунок 2 - Диаграмма последовательности

Диаграмма классов — это структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними.

Ниже на рисунке 3 представлена диаграмма классов.

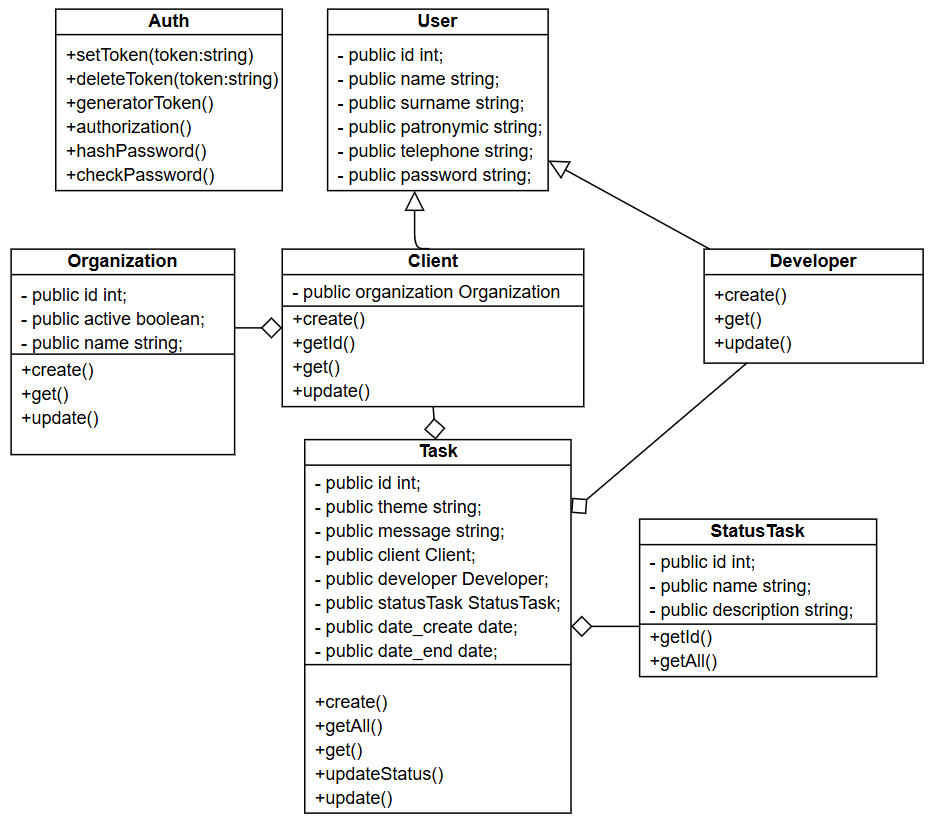


Рисунок 3 - Диаграмма классов

Диаграмма компонентов - это структурная диаграмма языка унифицированного моделирования, она описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами.

На рисунке 4 ниже представлена диаграмма компонентов.



Рисунок 4 - Диаграмма компонентов

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения. При этом представляются только компоненты-экземпляры программы, являющиеся исполнимыми файлами или динамическими библиотеками.

На рисунке 5 представлена диаграмма развертывания.



Рисунок 5 - Диаграмма развертывания

Диаграмма состояний — ориентированный граф для конечного автомата, в котором вершины обозначают состояния дуги показывают переходы между двумя состояниями. На практике вершины обычно изображаются в виде окружностей и, если нужно, двойных окружностей.

На рисунке 6 представлена диаграмма состояния.



Рисунок 6 - Диаграмма состояния

2.2 ВЫБОР ИНСТРУМЕНТАЛЬНЫХ И ПРОГРАММНЫХ СРЕДСТВ

РАЗРАБОТКИ

На начальных этапах процесса проектирования программного продукта необходимо принять принципиальные решения, во многом определяющие этот процесс, а также качество и трудоемкость разработки. К таким решениям относят:

– выбор архитектуры программного обеспечения;

– выбор типа пользовательского интерфейса;

– выбор подхода к разработке;

– выбор языка и среды программирования.

Языки веб-разработки - это языки программирования и технологии, используемые для создания веб-сайтов, веб-приложений и компонентов серверной части. Они составляют основу веб-разработки и определяют характер взаимодействия пользователей с веб-контентом. Существует два основных аспекта веб-разработки:

Front-End Development: Включает в себя проектирование и реализацию визуальных элементов, которые видят пользователи и с которыми они взаимодействуют в веб-приложении. Разработчики фронт-энда используют комбинацию HTML, CSS и JavaScript для создания визуально привлекательного и функционального пользовательского интерфейса.

Back-End Development: Работает с компонентами на стороне сервера и манипулирует данными, обеспечивая их хранение, обработку и получение. Это обеспечивает бесперебойную работу веб-приложения и его масштабирование при необходимости. К распространенным языкам back-end относятся Python, PHP, Ruby, Java и C#.

Для Front-End Development в качестве языка будет использоваться JavaScript как один из самый лидирующих языков в Front-End разработке.

Кроме языка программирования требуется выбрать используемые фреймворк.

Фреймворк (англ. framework — «каркас, структура») — это готовый набор инструментов, который помогает разработчику быстро создать продукт: сайт, приложение, интернет-магазин, CMS-систему.

В таблице 1 представлено сравнение Фреймворк для Front-End разработки.

Таблица 1 – Сравнение Фреймворк для Front-End разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Фреймворк | Производительность | Сложность | Комьюнити |
| Angular | - | Сложный | + |
| Vue3 | + | Простой | - |
| React | + | Средней | + |
| Svelte | - | Простой | - |

В качестве выбора предпочтителен Vue3.

Vue3 сочетает в себе простоту изучения и использования, за счет своего меньшего размера и улучшения производительности в 3 версии выигрывает по производительности у своих аналогов, так же имеет большое и активное сообщество разработчиков, а также широкий выбор инструментов и библиотек.

2.2.2 Выбор программных средств для создания веб-приложения

Существует множество способов писать код для веб-приложений: от текстовых редакторов до облачных сред разработки. Трудно сразу решить, какая среда лучше подходит для поставленных задач. Чтобы сэкономить вам время, выбрали наиболее популярные это IDE и редактор кода.

IDE (Integrated Development Environment) — это набор программных инструментов, которые используются для создания ПО. Второе название — интегрированная среда разработки.

Редактор кода - это программное обеспечение, которое позволяет программистам и разработчикам создавать, редактировать и отлаживать исходный код программ.

В таблице 2 представлено сравнение средств для создания веб-приложения.

Таблица 2 – Сравнение средств для создания веб-приложения

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Платность | Удобность | Быстрота |
| VS code | - | + | + |
| WebStorm | + | + | - |
| Sublime Text | - | - | + |

Из выбранных вариантов предпочтителен VS code.

Visual Studio Code сочетает в себе простоту редактора исходного кода с мощными инструментами разработчика, такими как доработка и отладка кода IntelliSense.

В основе Visual Studio Code лежит молниеносный редактор исходного кода, идеально подходящий для повседневного использования. Благодаря поддержке сотен языков VS Code позволяет мгновенно повысить производительность при работе с подсветкой синтаксиса, подбором скобок, автоматическим отступом, выделением блоков, фрагментами и многим другим.

2.2.3 Выбор системы управления базами данных

Система управления базами данных (СУБД) — это программное обеспечение, предназначенное для создания, управления, обновления и анализа баз данных. Она обеспечивает интерфейс для взаимодействия пользователя или приложения с данными, хранящимися в базе данных. СУБД позволяют структурировать данные таким образом, чтобы обеспечить их легкий доступ, безопасность и эффективное использование.

СУБД — комплекс программ, позволяющих создать базу данных и манипулировать данными (вставлять, обновлять, удалять и выбирать). Система обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства для администрирования БД.

В таблице 3 представлено сравнение СУБД.

Таблица 3 – Сравнение СУБД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| СУБД | Бесплатность | Размер базы | Размер таблицы | Число пользователей |
| HSQLDB | нет | 28 ТB | 120 GB | Не ограничено |
| Microsoft SQL Server | нет | 16 ТВ | 532 GB | Не ограничено |
| MySQL | да | 256 TB | 256 ТB | Не ограничено |
| PostgreSQL | да | Неограничен | 32 TB | Не ограничено |

Так как количество данных всегда будет увеличивать и их требуется хранить то лучшей СУБД является PostgreSQL который не имеет ограничений на размере базы данных, так же является бесплатной СУБД, так же имеет возможность быстрого чтения среди множество данных и неограниченный размер хранение индексов.

PostgreSQL заработал прочную репутацию благодаря своей проверенной архитектуре, надежности, целостности данных, надежному набору функций, расширяемости и преданности сообщества открытого исходного кода, стоящего за программным обеспечением, для последовательного предоставления производительных и инновационных решений. PostgreSQL работает во всех основных операционных системах, совместим с ACID с 2001 года и имеет мощные надстройки, такие как популярный расширитель геопространственных баз данных PostGIS. Неудивительно, что PostgreSQL стала предпочитаемой реляционной базой данных с открытым исходным кодом для многих людей и организаций.

2.2.4 Выбор средств разработки веб сервера

Backend (бэкенд) – серверная часть сайта. Отвечает за быструю загрузку страниц, обработку данных, безопасность, интеграцию с другими системами. Пользователь не видит всего этого, но благодаря backend веб-продукт или ПО нормально функционируют.

Backend-разработчик пишет код, организует хранение и передачу данных. Создает и поддерживает механизмы и алгоритмы, которые позволяют посетителям взаимодействовать со страницей. Например, делает так, чтобы платежи в интернет-магазине были безопасными.

В таблице 4 представлено сравнение фрейморков для разработки веб сервиса.

Таблица 4 – Сравнение фрейморков для разработки веб сервиса

|  |  |  |  |
| --- | --- | --- | --- |
| Фреймворк | Сложность | Документация | Производительность |
| Nest(js) | Средний | + | + |
| Django(python) | Легкий | - | - |
| Spring(java) | Сложный | - | + |

Из представленных Фреймворков выбран Nest(js) из-за своей простоты и хорошо написанной документации в сравнение с др. фреймворками, так же используется тот же язык что и в frond-end разработке что упрощает разработку.

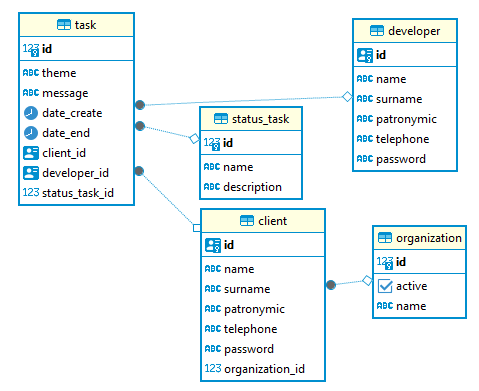
В рамках этой главы были достигнуты следующие задачи выбран язык и фреймворк разработки front-end приложения, выбран язык и фреймворк разработки back-end приложения, выбрана система управления базой данных и выбраны программные средства для разработки веб приложения.

3. Реализация ИС предприятия ООО «ИМЦ»

3.1. Разработка ИС предприятия ООО «ИМЦ»

Реализация приложения с стороны базы данных.

На рисунке 10 представлена схема базы данных.



Рисунке 10 - схема базы данных

Реализация приложения с стороны back-end части.

На рисунке 10 представлены конечные точки API группы auth.



Рисунок 10 - Конечные точки API группы auth.

На рисунке 11 представлены конечные точки API группы developer.



Рисунок 11 - Конечные точки API группы developer.

На рисунке 12 представлены конечные точки API группы client.



Рисунок 12 - Конечные точки API группы client.

На рисунке 13 представлены конечные точки API группы task.



Рисунок 13 - Конечные точки API группы client.