ВВЕДЕНИЕ

1 Анализ требований для разработки программного модуля

ведения учетных записей медицинского центра

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.

Управление предприятием ООО «ИМЦ» осуществляется директором, он является руководителем предприятия. В подчинении у директора находятся все начальники IT отделов и бухгалтера. Главный бухгалтер ведёт отчеты по всему предприятию.

Начальник IT отделов следят за работой своей задачи и распределяют задачи.

Главный системный администратор – ответственный системный администратор, выполняющий обязанности системного администратора и распределяющий обязанности между сотрудниками своего отдела.

Системный администратор – это работник, должностные обязанности которого включают обеспечение штатной работы компьютерной техники, сети и программного обеспечения.

Техник – это специалист, занимающийся обслуживанием, профилактикой и ремонтом различного типа оборудования.

Начальник разработчиков - технический или исполнительный директор, который непосредственно занимается реализацией одного либо нескольких проектов.

Разработчик front-end это визуальная часть веб-сайта, которую пользователь видит и с которой может взаимодействовать при помощи браузера.

Разработчик back-end это разработка бизнес-логики продукта, выполняющий функционал работой с базой данной, файлами и т.д т.е. что не должно находится в зоне видимости пользователя.

Разработчик sql это программист, который работает с базами данных, занимающийся анализом, поддержкой и проектированием приложений, которые взаимодействуют с языком программирования SQL.

Техническая поддержка - это отдельная служба (группа людей), созданная для получения и обработки обращений клиентов.

HR-менеджер - это специалист, который организует управление персоналом в компаниях и несет ответственность за то, чтобы в коллективе работали максимально подкованные сотрудники с точки зрения личных и рабочих качеств.

На рисунке 1 представлена организационная структура ООО «ИМЦ».



Рисунок 1 - Организационная структура ООО «ИМЦ»

Предприятия ООО «ИМЦ» занимается разработкой компьютерного программного обеспечения и его поддержкой, то предприятие нуждается в постоянную возможность быстро реагирования на существующие ошибки своего продукта от клиентов что бы удержать клиента на продление подписки на поддержку.

1.1 Анализ задач, функций и требований программного модуля

ведения учетных записей медицинского центра

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 2.



Рисунок 2 - 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.

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.

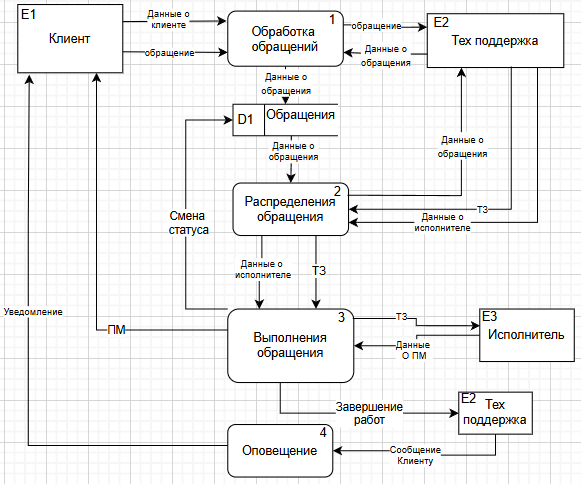


Рисунок 8 - Декомпозиция процесса «Обработка обращения клиента»

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.

Логическая модель данных — это расширение концептуальной модели данных. Она включает в себя все сущности, атрибуты, ключи и взаимосвязи, которые представляют бизнес-информацию и определяют бизнес-правила.

Использование метода IDEF1X наиболее целесообразно для построения логической структуры информационной системы после того, как все информационные ресурсы исследованы. Средства моделирования IDEF1X специально разработаны для построения реляционных информационных систем.

На рисунке 11 представлена физическая модель в нотации IDEF1X.

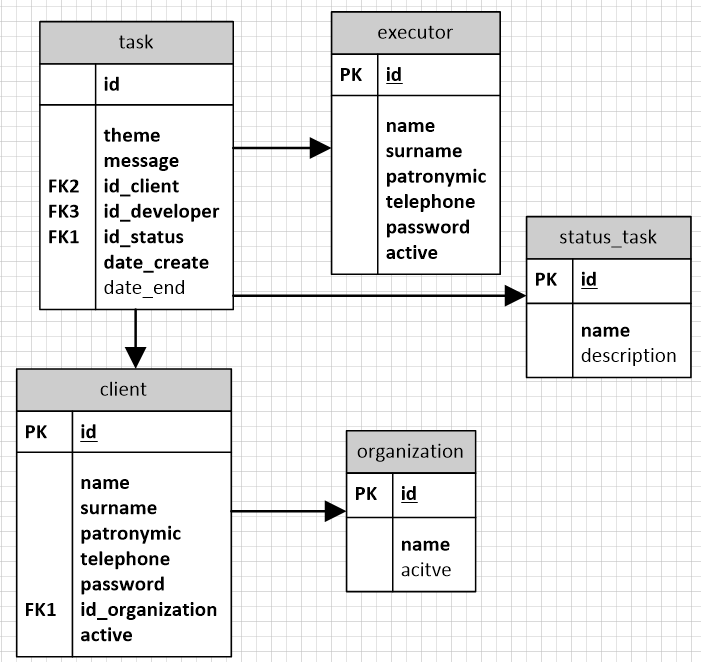


Рисунок 11 - Физическая модель в нотации IDEF1X

1.2 Обзор существующих методов и программных разработок в

создании информационных систем медицинского назначения

Рассмотрим какие программные продукты есть на рынке, которые могли подойди для оптимизации процесса.

Для начала определяются критерии, по которым будет проводиться сравнение программных продуктов. Основные критерии включают в себя следующее:

* функциональность и возможности;
* удобство использования;
* простота в использовании;
* производительность;
* безопасность;
* стоимость;
* поддержка и обновления.

Ниже рассмотрим несколько вариантов программных продуктов.

YouTrack - коммерческая система отслеживания ошибок, программное обеспечение для управления проектами, разработанное компанией JetBrains.

YouTrack поддерживает поисковые запросы, автодополнение, манипуляцию с наборами задач, настройку набора атрибутов задачи, создание пользовательских рабочих процессов и реализует подход, основанный на преимущественном использовании клавиатуры.

Jira — коммерческая система отслеживания ошибок, предназначена для организации взаимодействия с пользователями, хотя в некоторых случаях используется и для управления проектами. Разработана компанией Atlassian, является одним из двух её основных продуктов. Имеет веб-интерфейс.

Jira имеет большое количество возможностей конфигурации: для каждого приложения может быть определён отдельный тип задачи с собственным workflow, набором статусов, одним или несколькими видами представления (англ. screens). Кроме того, с помощью так называемых «схем» можно определить для каждого индивидуального Jira-проекта собственные права доступа, поведение и видимость полей и многое другое.

Trello — облачная программа для управления проектами небольших групп, разработанная Fog Creek Software.

Trello использует парадигму для управления проектами, известную как канбан, метод, который первоначально был популяризирован Toyota в 1980-х для управления цепочками поставок.

Ниже в таблице 1 представлено сравнение программных продуктов в области технической поддержки.

Таблица 1 - сравнение программных продуктов в области технической поддержки.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| название | удобство | Бесплатная | Поддержка | Безопасность | Простота |
| Trello | - | + | - | + | + |
| YouTrack | + | - | + | + | - |
| Jira | + | - | - | + | + |

Из выше представленных примерах не подходит не какой вариант.

У каждого варианта есть свои недостатки и следующие ограничения:

* нет возможности изменять программу под свои нужны;
* существует зависимость от сторонних поставщиков;
* отсутствует полный контроль над данными и безопасностью, данные могут хранится или передаваться на сторонние сервисы;
* отсутствует возможности интеграции с другими системами.

Вследствие выше описанных проблем и с перспективой на будущее лучшим решением является разработка собственного программного продукта.

1.3. Планирование и определение затрат на разработку программного

модуля ведения учетных записей медицинского центра

Для разработки ПМ требуется выполнить следующие этапы:

* анализ и проектирование ПМ;
* разработка базы данных ПМ;
* разработка серверной части ПМ;
* разработка интерфейсной части ПМ;
* тестирование разработанного ПМ.

На этапе анализ и проектирование ПМ следующие задачи:

* анализ предметной области;
* анализ задач, требуемых от ПМ;
* анализ средств разработки ПМ;
* проектирование логики работы ПМ;
* проектирования макета ПМ.

На этапе разработки базы данных ПМ следующие задачи:

* реализация сущностей базы данных;
* определения связей между сущностями;
* проставление индексов в базе данных.

На этапе разработки серверной части ПМ:

* настройка и поднятие среды;
* разработка конечный точек;
* разработка классов серверов;
* разработка классов сущностей.

На этапе разработка интерфейсной части ПМ следующие задачи:

* разработка макета ПМ;
* разработка компонентов ПМ;
* разработка стилей;
* разработка страниц.

На этапе тестирование разработанного ПМ следующие задачи:

* тестирования скорости базы данных;
* тестирование контрольных точек серверной части;
* тестирования интерфейсной части ПМ.

Для выполнения описанных задач требуются следующие сотрудники:

* аналитик - специалист, занимающийся аналитическими исследованиями и обобщением в определенной сфере деятельности, который в совершенстве владеет методами анализа, обычно способен прогнозировать процессы и разрабатывать перспективные программы развития;
* проектировщик - специалист, занимающийся разработкой планов различных конструкций;
* разработчик базы данных - программировать, разрабатывать и внедрять системы баз данных;
* разработчик серверной части - это специалист, который занимается серверной частью сайтов. Он реализует внутреннюю логику работы приложения, обеспечивает его взаимодействие с базами данных и внешними сервисами.
* разработчик интерфейсов - это специалист, который занимается разработкой пользовательского интерфейса, то есть той части сайта или приложения, которую видят посетители страницы.

Перспектива разрабатываемого продукта заключается в следующих возможностях:

* возможность отправки сообщения на почту или номер телефона о выполненных работах;
* возможность уведомлять пользователей через браузер уведомления;
* возможность создавать клиентам шаблоны текстов ошибок;
* возможность создавать исполнителям шаблоны текстов ответы клиентам;
* повышение удобства программного продукта;
* функциональность изучения обращение клиентов на поиск похожих для достижения уменьшения нагрузки на тех. поддержку;
* добавление по необходимости новый маркеров обращения;
* функциональность определения исполнителя по сообщению и теме обращения;
* создания комментариев в обращении;
* возможность добавление файлов в обращении;
* функционал отчетов по проделанных работах.

Данные возможности со временем позволят увеличить доход за счет более качественной обратной связи с клиентами и улучшения качества программного обеспечения, что влечет за собой появление новых клиентов. Так же данные изменения очень сильно облегчат процессы отдела технической поддержки, что уменьшит количество требований сотрудников на данных должностях.

ТУТ БУДЕТ ПРОДОЛЖЕНИЯ!!

1.4 Техническое задание на разработку программного модуля

ведения учетных записей медицинского центра

1 Общие сведения

1.1. Полное наименование системы и ее условное обозначение

Полное наименование: программного модуля взаимодействия с клиентами в медицинском центре.

Условное обозначение: ПМВКМЦ.

1.2. Наименование организаций - заказчика и разработчика системы

Заказчик: ООО Информационно-медицинский центр.

Юридический адрес: д. литера «З», ул. Князя Григория Засекина, 1, Самара, Самарская обл., 443099.

Номер телефона: +7(846) 222-72-32

Разработчик: Семенов Максим Александрович.

Адрес исполнителя: д 130, ул. Комарова, Балаково, Саратовская обл. 413841.

Номер телефона: 89371479157

1.3. Перечень документов, на основании которых создается программный продукт:

* ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»;
* Договор №1051 от 11.04.2025 на разработку программного модуля, заключенный между заказчиком и исполнителем;
* Календарный план работ.

1.4. Плановые сроки начала и окончания работ

Начало работ: 14.04.2025г.

Окончание работ: 26.05.2024г.

2 Назначение и цели создания системы

2.1. Назначение системы

Система предназначена для автоматизации процесса работы технической поддержки с клиентами в медицинском центре.

2.2. Цели создания системы

* повышение эффективности технической поддержки;
* улучшение коммуникации между клиентами и исполнителями;
* оптимизация процесса обработки обращений клиентов;
* обеспечение центрального хранилище обращении клиентов.

3 Характеристика объектов автоматизации

Объектами автоматизации являются: процесс обработки обращений клиентов, работа отдела технического поддержки.

4.1. Требования к системе в целом

4.1.1. Требования к структуре и функционированию системы

Система должна состоять из следующих подсистем:

* подсистема авторизации и аутентификации;
* подсистема управления обращениями;
* подсистема управления клиентами;
* подсистема управления исполнителями;
* подсистема управления организациями.

4.1.2. Требования к персоналу

Система предназначена для двух типов пользователей:

* клиент – сотрудники медицинского центра.
* исполнитель - сотрудники компании ООО «ИМЦ».

4.1.3. Требования к надежности

Требования к надежности программного обеспечения включают в себя следующие ключевые аспекты:

* безотказность – ПО должно выполнять свои функции без сбоев;
* отказоустойчивость – ПО должно сохранять возможность работоспособности при возникновении ошибок или сбоев;
* доступность - время, в течение которого система должна быть доступна для использования 24/7, не считая моментов технического обслуживания.

4.1.4. Требования к безопасности

* обеспечить конфиденциальности данных;
* предотвращать утечку данных;
* блокировка учетных записей при попытках несанкционированного доступа;
* наличия механизмов идентификации пользователей;
* наличия защиты от подбора паролей;
* наличия защиты от sql инъекций;
* наличия защиты от XSS уязвимостей.

4.1.5. Требования к эргономике и технической эстетике

ПМ должен иметь человеко-машинный интерфейс, удовлетворяющий следующим требованиям:

* взаимодействие системы и пользователя должно осуществляться на русском языке, за исключением системных сообщений, не подлежащих русификации;
* при работе с интерфейсом пользователь должен быть ориентирован на работу с клавиатурой и манипулятором графической информации «мышь»;
* должно быть реализовано отображение на экране только тех возможностей, которые доступны конкретному пользователю в соответствии с его функциональной ролью в системе;
* должна быть реализована возможность работы с системой при двух мониторной конфигурации дисплеев;
* представление управляющих элементов, экранных форм и их информационных элементов (окон, панелей и т.п.) должно быть унифицировано. Экранные формы должны полностью находиться в видимой площади экрана монитора с диагональю 17’ при разрешении экрана 1280 х 1024 и выше.

4.2. Требования к функциям, выполняемым системой

4.2.1. Подсистема авторизации и аутентификации

* авторизация клиента и исполнителя;
* аутентификация клиента и исполнителя.

4.2.2. Подсистема управления обращениями

* создание обращения клиентом;
* изменение обращения;
* изменение статуса обращения;
* просмотр всех обращений для исполнителей;
* фильтрация обращений.

4.2.3. Подсистема управления пользователями

* добавление клиента;
* изменение активности клиента;
* получение списка клиентов;
* добавление исполнителя;
* изменение активности исполнителя;
* получение списка исполнителей.

4.2.4. Подсистема управления организациями

* добавление организации клиента;
* получение списка организаций клиентов;
* изменение активности организаций.

4.3. Требования к видам обеспечения

4.3.1. Требования к информационному обеспечению

Информационное обеспечение должно обеспечивать системность, информационную полноту, избирательность, непрерывность, целостность потока информации по всей совокупности релевантной информации об объектах контроля и субъектах наблюдения.

Уровень хранения данных в ПМ должен быть построен на основе реляционных СУБД.

Для обеспечения целостности данных должны использоваться встроенные механизмы СУБД.

Технические средства, обеспечивающие хранение информации, должны использовать современные технологии, позволяющие обеспечить повышенную надежность хранения данных и оперативную замену оборудования.

4.3.2. Требования к программному обеспечению

На сервере должны быть установлены следующие ПО:

* node v 20.12.2 – для развертывания клиентской и серверной части программного модуля;
* npm v10.5.0 - менеджер пакетов;
* python – язык программирования, который используется для сборки программ в node;
* vue-cli-service – библиотека node которая используется для сборки клиентской части программного модуля;
* nest - фрейморк node который используется для сборки серверной части программного модуля;
* PostgreSQL – база данных для хранения данных.

У клиента должны быть установлен один из выше описанных браузеров последних версии: Яндекс, Chromium-Gost, Google.

4.3.3. Требования к техническому обеспечению

Сервер должен соответствовать рекомендуемым требованиям программного обеспечения.

Так же поддерживать технологию RAID1.

Персональные компьютеры клиентов должны иметь следующие минимальные характеристики:

* ОП 8Гб;
* процессор 4 ядерный с 3ГГЦ.

5. Состав и содержание работ по созданию программного модуля

* анализ и проектирование ПМ;
* разработка базы данных ПМ;
* разработка серверной части ПМ;
* разработка интерфейсной части ПМ;
* тестирование разработанного ПМ.

6 Порядок контроля и приемки системы

Тестирование ПМ будет осуществлять поочередно:

* тестирования серверной части ПМ;
* тестирования компонентов интерфейса ПМ;
* целостное тестирования ПМ.

Ответственность за организацию и проведение приемки системы должен нести заказчик. Приемка системы должна производиться по завершению приемки всех задач системы. При этом необходимо предоставить обеспечение материальной частью, проектной документацией и специально выделенным персоналом.

Завершающим этапом при приемке системы должно быть составление акта приемки.

7 Требования к документированию

Требуется разработать следующие документы:

* руководство клиентам;
* руководство исполнителей;

Документация должна быть выполнена на русском языке.

Электронная версия документации предоставляется в форматах PDF и DOC/DOCX.

Требования к содержанию руководства пользователя:

* описание интерфейса ПМ;
* описание выполняемых действии пользователями в ПМ.

8 Источники разработки

* ГОСТ 34.602-89 "Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы";
* https://nodejs.org – документация node;
* https://nestjs.com - документация фрейморка серверной части программного модуля;
* https://vuejs.org - документация фрейморка клиентской части программного модуля;
* https://www.postgresql.org - документация базы данных программного модуля.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ

АДМИНИСТРИРОВАНИЯ.

2.1 Проектирование программного модуля ведения учетных

записей медицинского центра с использованием 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 языка и среды разработки программного модуля

ведения учетных записей медицинского центра

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.

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.

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.

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 выбраны язык и фреймворк разработки front-end и back-end приложения, а также система управления базой данных и программные средства для разработки веб приложения.

(НЕ ПОЛНЫЙ ВАРИАНТ)

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

На таблице 2 описана сущность client.

Таблица 2 - Сущность client

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Примечание |
| id | int | Первичный ключ |
| name | varchar | Имя пользователя |
| surname | varchar | Фамилия пользователя |
| patronymic | varchar | Отчество пользователя |
| id\_ contract | int | Внешний ключ |
| telephone | varchar | Номер телефона |

Сущность организация представлена в таблице 3.

Таблица 3 - Сущность организация

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Примечание |
| id | int | Первичный ключ |
| name | varchar | Имя организации |
| active | boolean | Активность поддержки организации |

Сущность разработчик представлена в таблице 4.

Таблица 4 - Сущность испольнитель

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Примечание |
| id | int | Первичный ключ |
| name | varchar | Имя разработчика |
| surname | varchar | Фамилия разработчика |

Сущность статус задачи представлена в таблице 5.

Таблица 5 - Сущность статус задачи

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Примечание |
| id | int | Первичный ключ |
| name | varchar | Название статуса задачи |
| description | text | Описание статуса задачи |

Сущность задача представлена в таблице 6.

Таблица 6 - Сущность задача

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Примечание |
| id | int | Первичный ключ |
| theme | varchar | Тема задачи |
| message | text | Сообщение задачи |
| id\_client | int | Внешний ключ |
| id\_ developer | int | Внешний ключ |
| id\_status | int | Внешний ключ |
| date\_create | timestamp | Дата и время создание задачи |

Реализация приложения с стороны back-end части.

ОПИСАТЬ ВСЕ МЕТОДЫ API

На рисунке 10 представлены конечные точки API группы auth.



Authorization – это api предназначена для авторизации пользователя путем ввода логина и пароля.

getProfile - это api предназначена для получение личных данных авторизованного пользователя.

exit - это api предназначена для выхода пользователя из системы.

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.

3.2. Описание интерфейса ПМ

ОПИСАТЬ ДЛЯ ЧЕГО НУЖЕН КАЖДЫЙ ЭКРАН

На рисунке 15 представлен экран авторизации



Рисунок 15 - Экран авторизации

На рисунке 16 представлен компонент меню



Рисунок 16 - Компонент меню

На рисунке 17 представлен экран задачи



Рисунок 17 - Экран задачи

На рисунке 18 представлено модальное окно с задачей



Рисунок 18 - Модальное окно с задачей

На рисунке 19 представлен экран разработчики с правами пользователь



Рисунок 19 - Экран разработчики с правами пользователь

На рисунке 20 представлен экран разработчики с правами разработчик



Рисунок 20 - Экран разработчики с правами разработчик

На рисунке 23 представлен экран клиенты с правами разработчик



На рисунке 23 Экран клиенты с правами разработчик

На рисунке 24 представлен экран организации с правами разработчик



На рисунке 24 Экран организации с правами разработчик

3.3. Тестирование ПМ

Для проверки функционала нашего продукта будет произведено ручное тестирование.

Ручное тестирование часть процесса тестирования на этапе контроля качества в процессе разработки программного обеспечения. Оно производится тестировщиком без использования программных средств, для проверки программы или сайта путём моделирования действий пользователя. В роли тестировщиков могут выступать и обычные пользователи, сообщая разработчикам о найденных ошибках.

Проверка back-end части.

НА КАЖДЫЙ API СДЕЛАТЬ ПРОВЕРКУ ЧТО ВОЗВРАЩАЮТ ОЖИДАЕМЫЕ ДАННЫЕ

Проверка интерфейса.

Для начала проверим форму авторизации.

Попробуем ввести не правильный логин и пароль результат представлен ниже на рисунке 20.



Рисунок 20 – результат неверно введенного пароля

И ТАК ПРИДУМАТЬ ЕЩЕ ЧТО МОЖНО ПРОВЕРИТЬ И СДЕЛАТЬ СКРИНЫ