ВВЕДЕНИЕ

Преддипломная практика является частью образовательного процесса подготовки специалиста, продолжением учебного процесса в производственных условиях.

Целью преддипломной практики является расширение и углубление профессиональных практических знании и умении.

Задачами преддипломной практики являются:

* анализ организационной структуры;
* анализ информационной системы предприятия;
* анализ предметной области;
* анализ требовании для разработки программного модуля;
* проектирование программного модуля;
* разработка программного модуля.

В ходе преддипломной практики необходимо собрать и проанализировать информацию об организационной структуре и информационной системе предприятия, определить программно-аппаратные средства компании ООО «ИМЦ», разработать модель процесса обработки обращения клиентов, проанализировать требования к разрабатываемому программному продукту, определить сущности ПМ, задачи выполняющие ПМ, разработать диаграммы UML ПМ, определить ПО необходимое для разработки ПМ, разработать ПМ.

1 АНАЛИЗ ОРГАНИЗАЦИОННОЙ СТРУКТУРЫ И

ИНФОРМАЦИОННОЙ СИСТЕМЫ ПРЕДПРИЯТИЯ

1.1 Анализ организационной структуры ООО «ИМЦ»

Специализацией компании «Информационно-медицинский центр» (ИМЦ) является информатизация здравоохранения, включая комплексные решения для медицинских организаций, органов управления здравоохранением и территориальных фондов ОМС.

Создана компания «ИМЦ» со специализацией в области разработки, внедрения и сопровождения медицинских информационных систем в 2005 году. До этого специалисты ИМЦ начинали свою трудовую деятельность в отделе информационных систем компании «ПАРУС» с основной специализацией - автоматизация здравоохранения и ОМС с 1994 года.

Адрес предприятия ООО «ИМЦ»: 443099, г. Самара, ул. князя Григория Засекина (Карбюраторная), д. 1, литера «З»

Руководитель предприятия ООО «ИМЦ»: директор Новиков Олег Васильевич.

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.

Основная информация об ООО «ИМЦ» представлена в таблице 1.

Таблица 1 – Общие сведения о ООО «ИМЦ»

|  |  |
| --- | --- |
| Наименование: | ООО «ИМЦ» |
| Основной ОКВЭД: | Разработка компьютерного программного обеспечения |
| Страна: | РОССИЯ |
| Регион: | Самарская область |
| ИНН: | 6317059075 |
| ОКПО или др.: | 78250264 |
| Данные госрегистрации: | Межрайонная инспекция ФНС России № 18 по  Самарской области Дата постановки на учет: 20 июня 2005 г. |
| Юридический адрес: | 443099, г. Самара, ул. князя Григория Засекина (Карбюраторная), д. 1, литера «З» |

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.

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

На рисунке 1 представлена организационная структура ООО «ИМЦ».



Рисунок 1 - Организационная структура ООО «ИМЦ»

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 - распределяет обязанности между сотрудниками своего отдела.

Системный администратор – обеспечивает штатную работу компьютерной техники, сети и программного обеспечения.

Техник – занимается обслуживанием, профилактикой и ремонтом различного типа оборудования.

Начальник разработчиков - занимается реализацией одного либо нескольких проектов.

Разработчик front-end - разрабатывает визуальная часть веб-сайта.

Разработчик back-end - разрабатывает бизнес-логики продукта.

Разработчик sql - разрабатывает базу данных.

Техническая поддержка - отдел, обрабатывающий обращений клиентов.

HR-менеджер - организует управление персоналом в компании.

1.2 Описание информационной системы предприятия ООО «ИМЦ»

В компании ООО «ИМЦ» используются следующее аппаратное обеспечение:

- ПК с базовым составом в количестве 10шт.;

- МФУ от компании Canon 4шт.;

На всех компьютерах ООО «ИМЦ» установлено следующее программное обеспечение:

- Microsoft Windows 10 – операционная система от компании Microsoft, установлена на всех компьютерах предприятия, обеспечивает работу комплектующих компьютера и облегчает взаимодействие компьютера с пользователем;

- 1С: Бухгалтерия 10» — это профессиональный инструмент бухгалтера, с помощью которого можно вести учет, готовить и сдавать обязательную отчетность;

- «1С: Предприятие» предназначено для автоматизации бухгалтерского и управленческого учётов (включая начисление зарплаты и управление кадрами), экономической и организационной деятельности предприятия.

- Microsoft Office — офисный пакет приложений, специально созданных корпорацией Microsoft для ОС Microsoft Windows, Windows Phone, Android. В состав этого пакета входит программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных и др;

- NX (система автоматизированного проектирования) – NX предлагает широкий набор инструментов, решающий специализированные задачи автоматизированного проектирования.

Доступ к серверу «ИМЦ» отдела разработки представляет подключение через VPN, локальные машины сотрудников представляют собой виртуальный машины на сервере, ибо большинство работает удаленно.

Таким образом, проведен анализ деятельности компания ООО «ИМЦ», составлена организационная структура компании ООО «ИМЦ», определены программно-аппаратные средства компании ООО «ИМЦ».

2 АНАЛИЗ ТРЕБОВАНИЙ ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО

МОДУЛЯ

2.1 Анализ предметной области разработки программного модуля

Для разработки моделей процессов информационно-технической деятельности отдела технической поддержки ООО «ИМЦ» выбрана задача – обработка обращения клиентов.

Для разработки модели процесса обработки обращении от клиентов будет использоваться методология IDEF0.

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

Функциональная модель IDEF0 представляет собой набор блоков, каждый из которых представляет собой черный ящик со входами и выходами, управлением и механизмами, которые детализируются (декомпозируются) до необходимого уровня.

Функциональный блок контекстной диаграммы процесса обработки обращения клиентов представлена на рисунке 3.



Рисунок 3 - Функциональный блок контекстной диаграммы процесса обработки обращения клиентов.

Для более детального изучения процесса обработки обращения клиентов необходимо разработать декомпозицию контекстной диаграммы.

Декомпозиция контекстной диаграммы процесса обработки обращения клиентов представлена на рисунке 4.



Рисунок 4 - Декомпозиция контекстной диаграммы процесса обработки обращения клиентов

Клиент, обращается к технической поддержке из различных источников, передавая свои обращения и персональные данные, необходимые для исправления программного модуля, после чего сотрудник технической поддержки на основе договора проверяет наличие у клиента сопровождения технической поддержки и изучает другую информацию, необходимую для принятия обращения.

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

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

После выполнения обращения сотрудник технической поддержки должен связаться с клиентом, чтобы оповестить его о проделанной работе.

В этом процессе можно выделить следующие проблемы:

* техническая поддержка принимает обращения от клиентов через множество разных источников: социальные сети, Email, по телефону и СМС;
* техническая поддержка обрабатывает задачи в системе;
* после выполнения обращения клиента техническая поддержка оповещает клиента о выполненной работе.

Для решения этих проблем нужно провести оптимизацию данного процесса с помощью создания программного модуля. Создание программного модуля для предприятия ООО «ИМЦ» позволит сократить нагрузку сотрудников технической поддержки, сократить трудозатраты отдела технической поддержки и увеличить эффективность работы технической поддержки и доход ООО «ИМЦ».

Для оптимизации процесса требуется создать контекстную диаграмму модели TO-BE, которая должна решить проблемы в происходящем процессе обработки обращения клиентов.

Контекстная диаграмма процесса обработки обращения клиентов с использованием ПМ представлена на рисунке 5.



Рисунок 5 - Контекстная диаграмма процесса обработки обращения клиентов с использованием ПМ

Далее необходимо разработать декомпозицию контекстной диаграммы для модели TO-BE процесса обработки обращения клиентов.

Декомпозиция контекстной диаграммы для модели TO-BE, процесса обработки обращения клиентов представлена на рисунке 6.



Рисунок 6 – Декомпозиция контекстной диаграммы процесса обработки обращения клиентов с использованием ПМ

Клиент проходит процесс авторизации в программном модуле, в этот момент происходит проверка клиента на обслуживание технической поддержки, после чего клиенту предоставляется доступ к ПМ, далее происходит заполнение формы обращения, затем техническая поддержка осуществляет процесс определения исполнителя обращения. После чего исполнитель получает уведомление об обращении и приступает к его выполнению. После выполнения обращения исполнитель изменяет статус обращения «на проверку», далее ПМ уведомляет клиента о проделанных работах.

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

2.2 Сравнительный анализ программных модулей для взаимодействия с

клиентами

Рассмотрим какие программные продукты существуют на рынке, которые могли бы оптимизировать процесс обработки обращения клиентов.

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, которые позволяют оптимизировать работу с клиентами.

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.

YouTrack был разработан в соответствии с парадигмой языково-ориентированного программирования, использует JavaScript и Kotlin. Система использует встроенную базу данных Xodus для записи и хранения данных. Для удалённых вызовов процедур использует REST-стиль.

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.

Trello ограничил поддержку тегов в виде десяти цветных меток, которые можно переименовать. Карточки поддерживают комментарии, вложения, сроки выполнения и контрольные списки. Trello имеет API. В настоящее время поддерживаются мобильные платформы приложений iPhone и Android. Также был разработан веб-сайт, чтобы быть доступным в большинстве мобильных веб-браузеров.

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

Таблица 1 - сравнение программных продуктов в области технической поддержки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Удобство | Оплата | Поддержка | Безопасность | Простота |
| Trello | - | - | - | + | + |
| YouTrack | + | + | + | + | - |
| Jira | + | + | - | + | + |

У каждого варианта есть свои недостатки и следующие ограничения:

* нет возможности изменять ПМ под свои нужды;
* 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.

В рамках данной главы проведен анализ требовании к ПМ, разработаны диаграммы IDFE0 процесса обработки обращения клиентов, разработана модель TO-BE процесса обработки обращения клиентов с использованием ПМ, произведен анализ аналогов требуемого ПМ, произведен выбор разработки собственного ПО.

3 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО МОДУЛЯ

ВЗАИМОДЕЙСТВИЯ С КЛИЕНТАМИ

Программный модуль будет разработан для информационно-технической деятельности отдела технической поддержки ООО «ИМЦ» для оптимизации процесса обработки обращения клиента.

Задача программного модуля - систематизировать поступающие обращения клиентов и оптимизировать процесс их обработки.

Входными данными программного модуля являются данные клиента и его обращение.

Выходными данными программного модуля является выполненное обращение клиента.

Сущности программного модуля:

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

Программный модуль 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.

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

Ниже на рисунке 18 представлена диаграмма вариантов использования.



Рисунок 18 - Диаграмма вариантов использования

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 21 ниже представлена диаграмма компонентов.



Рисунок 21 - 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 22 представлена диаграмма развертывания.



Рисунок 22 - Диаграмма развертывания

В рамках данной главы были определены сущности ПМ, определены задачи выполняющие ПМ, разработаны диаграммы вариантов использования ПМ, классов ПМ, последовательности ПМ, компонентов ПМ, состояния ПМ.

4 РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ ВЗАИМОДЕЙСТВИЯ С

КЛИЕНТОМ

Реализация ПМ предусматривает создание программного продукта, состоящего из нескольких взаимосвязанных компонентов. Архитектура системы построена по принципу разделения на серверную часть и пользовательский интерфейс. Такой подход обеспечивает гибкость развития и поддержки приложения, а также позволяет масштабировать систему при необходимости.

Для реализации серверной части необходимо разработать базу данных.

После разработки базы данных необходимо разработать серверную часть приложения, которое будет выполнять логику и взаимодействовать с базой данных и возвращать данные клиентской частью приложения для вывода информации. Серверная часть представлена в виде REST API.

После разработки серверной части ПМ необходимо разработать интерфейс ПМ, расположение элементов в и его функциональность.

ПМ является закрытым ПО с ограниченным доступом, поэтому для получения доступа к ПМ необходимо произвести процесс авторизации пользователя путем ввода номера телефона и пароля. Для этого требуется разработать экран авторизации, представленный на рисунке 40.



Рисунок 40 - Экран авторизации

В данном экране пользователь должен ввести номера телефона и пароля от своего аккаунта и нажать на кнопку войти, после этого при успешном вводе серверная часть вернет токен авторизации для работы в ПМ, при неудачном вводе появится сообщение об ошибке.

На рисунке 41 представлен компонент навигационное меню для исполнителей.



Рисунок 41 - Компонент навигационное меню для исполнителей.

На рисунке 42 представлен компонент навигационное меню для клиентов.



Рисунок 42 - Компонент навигационное меню для клиентов.

Данный компонент позволяет обеспечивать удобные переходы между разными разделами ПМ.

Далее представлен основной экран ПМ задачи.

Данный экран представляет собой возможность просматривать задачи в таблице, фильтрация задач для более удобного поиска и возможность создавать новую задачу.

На рисунке 43 представлен экран задачи



Рисунок 43 - Экран задачи

На данном экране представлен компонент фильтр задач с кнопкой поиска который обновить таблицу.

В таблице есть кнопка открыть заявку позволяет более детально рассмотреть задачу и изменить статус задачи с помощью модального окна с задачей.

На рисунке 44 представлено модальное окно с задачей



Рисунок 44 - Модальное окно с задачей

В данном модальном окне есть возможность изменить содержимое задачи и представлены наборы кнопок с помощью которых происходит изменения статуса задач.

На рисунке 45 представлен экран исполнители для исполнителей



Рисунок 45 - Экран исполнители для исполнителей

Экран представляет возможность просмотра исполнителей в таблице, изменения активности исполнителей и создания новых исполнителей.

На рисунке 46 представлен экран исполнители для клиентов



Рисунок 46 - Экран исполнители для клиентов

Экран исполнители для клиентов позволяет клиентам ознакомится с людьми, которыми он будет взаимодействовать через ПМ.

На рисунке 47 представлен экран клиенты



Рисунок 47 - Экран клиенты

Данный экран предоставляет возможность просматривать клиентов, изменять их активность что позволяет закрыть доступ к ПМ и создавать новых клиентов в ПМ.

На рисунке 48 представлен экран организации



Рисунок 48 - Экран организации

Данный экран позволяет создавать организации и изменять их активность, если клиент находится в неактивной организации у него пропадает доступ к ПМ.

В данной главе реализована база данных PostgreSQL в ПМ, серверная часть ПМ на node используя фрейморк Nest представляющая собой архитектуру REST, клиентская часть ПМ на node используя фрейморк Vue3 используя методологию компонентов, представлен интерфейс ПМ.

Заключение

В период прохождения практики была проанализирована компания ООО «ИМЦ», составлена организационная структура компании ООО «ИМЦ», определены программно-аппаратные средства компании ООО «ИМЦ».

Далее проведен анализ требовании к ПМ, разработаны диаграммы IDFE0 процесса обработки обращения клиентов, разработана модель TO-BE процесса обработки обращения клиентов с использованием ПМ, произведен анализ аналогов требуемого ПМ, произведен выбор разработки собственного ПО.

Также были определены сущности ПМ, определены задачи выполняющие ПМ, разработаны диаграммы вариантов использования ПМ, классов ПМ, последовательности ПМ, компонентов ПМ, состояния ПМ.

А также реализована база данных PostgreSQL в ПМ, серверная часть ПМ на node используя фрейморк Nest представляющая собой архитектуру REST, клиентская часть ПМ на node используя фрейморк Vue3 используя методологию компонентов, представлен интерфейс ПМ.

Список используемых источников

1. Информационно - медицинский центр: официальный сайт. — Самара. URL: https://imc-s.ru (дата обращения: 30.04.2025). — Текст: электронный;
2. Диаграммы потоков данных DFD. URL: https://vc.ru/u/165346-evgeniy-kazak/562998-dfdy-dlya-nachinayushchih (дата обращения: 03.05.2025). — Текст: электронный;
3. Методология IDEF0: что это такое, основные понятия, этапы построения URL: https://habr.com/ru/companies/otus/articles/471674/ (дата обращения: 03.05.2025). — Текст: электронный.
4. Хатунцев, В. А. Современные системы управления проектами: Jira, Trello, Asana, YouTrack. — Москва: ДМК Пресс, 2023. — 298 с. — ISBN 978-5-94074-564-8. — Текст: непосредственный.
5. Фаулер, М. UML. Основы. / М. Фаулер, К. Скотт. — 3-е изд. — Санкт-Петербург: Символ-Плюс, 2018. — 192 с. — ISBN 978-5-93286-060-7. — Текст: непосредственный.
6. Структурный и объектно-ориентированный подходы к проектированию ИС Studme.org. URL: https://studme.org/210387/informatika/strukturnyy\_obektno \_orientirovannyy\_podhody\_proektirovaniyu (дата обращения: 03.05.2025). Текст: электронный.
7. Немцева, Т. И. Компаративный анализ языков программирования для веб-разработки / Т. И. Немцева, К. В. Сидоров. — Санкт-Петербург: БХВ-Петербург, 2024. — 368 с. — ISBN 978-5-9775-6589-0. — Текст: непосредственный.
8. Сравнение Vue.js, React и Angular: выбор фреймворка. URL: https://habr.com/ru/articles/726490/ (дата обращения: 03.05.2025). Текст: электронный.
9. VS Code vs WebStorm: какую IDE выбрать для веб-разработки. URL: https://medium.com/better-programming/vs-code-vs-webstorm-ide-for-web-development-e251dad93fea (дата обращения: 03.05.2025). Текст: электронный.
10. Сравнение PostgreSQL, MySQL и Microsoft SQL Server: что выбрать в 2025 году. URL: https://proglib.io/p/sravnenie-postgresql-mysql-i-ms-sql-server (дата обращения: 03.05.2025). Текст: электронный.
11. NestJS vs Django vs Spring Boot: сравнительный анализ back-end фреймворков. URL: https://devby.io/news/nestjs-vs-django-vs-spring-boot (дата обращения: 03.05.2025). Текст: электронный.