



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра прикладной информатики
Институт информационных систем и геотехнологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)
по направлению подготовки 09.03.03 Прикладная информатика
(квалификация – бакалавр)

На тему «Проектирование мессенджера для рассылки информационных сообщений»

Исполнитель Яковлев Владислав Сергеевич

Руководитель Д.т.н., профессор кафедры МИС Завгородний Владимир Николаевич

«К защите допускаю»

заведующий кафедрой _____

доктор технических наук, профессор

Истомин Евгений Петрович

« ____ » _____ 2020 г.

Санкт-Петербург
2020

ОГЛАВЛЕНИЕ

Введение.....	4
1 Анализ объекта и предмета исследования.....	7
1.1 Описание предмета и объекта исследования, постановка задачи на проектирование	7
1.2 Анализ существующих разработок и технологий для рассылки информационных сообщений.....	9
1.3 Обоснование выбора средств разработки.....	15
2 Методология проектирования инфор мационной системы.....	22
2.1 Функциональное проектирование исследуемого процесса.....	22
2.2 Реализация базы данных	31
2.3 Проектирование программного приложения.....	34
3 Комплексная оценка реализованной информационной системы	44
3.1 Тестирование информационной системы.....	44
3.2 Экономические параметры реализации предлагаемой информационной системы	45
Заключение	52
Список использованной литературы.....	55

Введение

В настоящее время совершенно немислимо представить человечество без средств коммуникации: почтовые службы и службы доставки, телефонная и видео связь и, безусловно, самая многофункциональная и распространенная — связь посредством интернета. Широкие возможности цифровых коммуникаций: мессенджеров, электронной почты, видео- и аудиоконференции, различных чатов и форумов, а в последнее время и широко распространенных служб мгновенных сообщений (мессенджеров) делают общение нынешних людей разноплановыми, разнообразным по содержанию и широте охвата контактов. Связь посредством рассылки мгновенных сообщений дает возможность обеспечить моментальную коммуникацию с огромным числом людей в режиме реального времени. Большое значение имеет возможность обмена моментальными текстовыми сообщениями, прикрепления к ним различного рода материалов, документов, имеет и для современного бизнеса — это повышает коммуникационные связи между сотрудниками, отделами, рабочими группами в разы, что дает возможность скорейшим образом реагировать на проявления внешней среды и изменения экономической ситуации.

Таким образом выбранная тема выпускной квалификационной работы, как и разрабатываемая в ней информационная система являются актуальными.

Степень научной разработанности проблемы. Коммуникация носит характер массовой коммуникации, теоретические основы которой изложены в трудах зарубежных и отечественных ученых: Беквит Г., Будовская Ю.В., Голдсмит Р., Збровская М.И., Котлер Ф., Левинсон Д. К., Марсден П., Рашкофф Д., Dennett, D. C., Gillett J., Jurvetson, S. и другие.

В ходе выпускной квалификационной работы (ВКР) разрабатывается проект приложения мессенджера, с помощью которого будет осуществляться рассылка информационных сообщений.

Объектом проекта является коммуникационная информационная среда Акционерного общества Туапсинское автотранспортное предприятие.

Предметом является автоматизированная система рассылки информационных сообщений.

Целью выпускной квалификационной работы является разработка корпоративного мессенджера для рассылки информационных сообщений.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ объекта и предмета исследования;
- определить методологию проектирования информационной системы;
- реализовать программное приложение мессенджера по рассылке информационных сообщений;
- произвести комплексную оценку реализованной информационной системы.

Теоретической базой данной выпускной квалификационной работы служат научные труды и публикации зарубежных и российских авторов в сфере информационных технологий, проектирования баз данных и информационных систем, а также другими, напрямую связанными с тематикой настоящей работы.

Информационную базу данной составляют публикации в научных изданиях и периодической печати, специальная и учебная литература.

Методами и средствами дипломной работы являются: системный анализ, математическое моделирование СУБД, информационное моделирование данных. В коде применялись стандарты объектно-ориентированного подхода к проектированию и программированию, сетевые технологии, WEB- приложение полностью реализовано на языке Java с выполнением операторов SQL-запросов и технологии java server pages (JSP).

Основными обозначенными критериями для разработки информационной системы являлись следующие: универсальность, адаптивность и простота автономного использования пользователем разрабатываемой системы, возможность интеграции с существующими, уже эксплуатируемыми

инструментальными средствами.

Практическая значимость исследования определяется тем, что материал данной работы и его результаты могут частично или полностью найти применение в коммерческих организациях осуществляющих свою деятельность в сфере транспортного обслуживания, так же может быть использовано автотранспортными предприятиями для диагностики результативности работы и совершенствования своей деятельности.

Основными структурными элементами проекта являются: самостоятельный программный продукт, предоставляемый на цифровом носителе; пояснительная записка; графический материал.

Структура и объем работы. Работа состоит из введения, трех разделов, заключения, списка использованной литературы. В первой главе проанализированы объект и предмет исследования, в рамках чего проведено описание предмета и объекта исследования, постановка задачи на проектирование, проведен также анализ существующих разработок и технологий для рассылки информационных сообщений, после чего обоснован выбор средств разработки. Вторая глава описывает методологию проектирования информационной системы, для чего приводятся схемы функционального проектирования исследуемого процесса, реализуется база данных, происходит проектирование программного приложения. Третья глава содержит комплексную оценку реализованной информационной системы: тестирование информационной системы и экономические параметры реализации предлагаемой информационной системы. В заключении работы приводятся выводы по каждой поставленной задаче.

1 Анализ объекта и предмета исследования

1.1 Описание предмета и объекта исследования, постановка задачи на проектирование

Объектом разработки информационной системы является акционерное общество Туапсинское Автотранспортное предприятие.

В рамках выполнения дипломного проекта поставлена задача разработать систему оперативного обмена служебными данными в виде текстовых файлов, файлов изображений, файлов других форматов и сообщений внутри определенной группы пользователей - корпоративного мессенджера.

Корпоративный мессенджер разрабатывается для рабочих групп, одновременно выполняющих несколько проектов, в каждом из которых предполагается назначение ролей между одними и теми же исполнителями (пользователями).

Организация работы с мессенджером предполагает, что в рабочей группе имеются специалисты, выполняющие конкретные задачи в конкретном проекте. При этом в другом проекте этот же сотрудник может выполнять задачу, соответствующую другой роли. Данный продукт ориентирован на организацию удаленной работы закрытых групп пользователей.

До того как начать проектирование необходимо определить цель проекта. Изменение ситуации после успешного осуществления проекта – это определение цели. Уменьшение затрачиваемого времени на рассылку информационных сообщений и оповещений является непосредственной целью данного проекта, это поможет увеличить эффективность работы отделов и организации, в общем.

Далее определим задачи проекта. Под задачей нужно понимать, определенные результаты, которые необходимы для решения проблем, что, конечно, нужно достигнуть в результате проекта. Отметим задачи описанные далее:

- простота дизайна и настраиваемый интерфейс;

- возможность отключение невостребованных функций;
- оптимизация.

В настоящее время сервисы для обмена мгновенными сообщениями становятся все более популярными, это связано со многими факторами, в числе которых:

1. Бесплатность. Все наиболее распространенные приложения бесплатны, что делает их использование выгоднее простой отправки смс сообщения с телефона, так как пользователем оплачивается только интернет трафик, стоимость которого снижается из года в год.

2. Кастомизируемость. Возможность изменить настройки приложений для конкретного пользователя, такие как: изображение контакта, имя, статус, настройка времени оповещений, черные списки, использование различных тем для приложения и пр.

3. Более широкие возможности. Помимо отправки текста, приложения поддерживают отправку медиа-контента (изображения, звуковые файлы, бинарные файлы), использование эмодзи и стикеров в тексте сообщений, поддержку аудио/видео звонков и пр.

На момент написания данной дипломной работы существовали следующие наиболее популярные сервисы для обмена сообщениями:

1. «Viber» - позволяет отправлять сообщения и звонить другим пользователям «Viber» в любой сети и стране мира. Существует для мобильных платформ и компьютеров под управлением Windows, Linux и OSX.

2. «WhatsApp Messenger» - кросс-платформенное приложение, позволяющее обмениваться сообщениями. Приложение доступно только для мобильных платформ. Кроме стандартного обмена сообщениями пользователи WhatsApp могут создавать группы, отправлять друг другу неограниченное число сообщений, фотографий, аудио- и видеофайлов.

3. « WeChat» поддерживает текстовые сообщения, передачу голосовых сообщений, рассылку сообщений множеству адресатов, возможность делиться фотографиями и видео. Существует для мобильных

платформ, имеется клиент для OSX и web-приложение.

4. «Telegram» - бесплатный кроссплатформенный мессенджер для смартфонов и других устройств, позволяющий обмениваться текстовыми сообщениями и медиа-файлами различных форматов.

Основная задача данной дипломной работы это создание отказоустойчивого сервиса для обмена сообщениями. Данную задачу можно разбить на следующие подзадачи:

1. Подготовка и настройка необходимого программного обеспечения для выделенной серверной части
2. Разработка программного обеспечения серверной части:
3. Разработка клиентских приложений для взаимодействия с сервером:
4. Запуск сервиса в эксплуатацию.

Основные возможности сервиса:

- регистрация, хранение и выдача информации о пользователях в системе;
- ведение переписки с другими зарегистрированными пользователями с использованием шифрования;
- возможность использования эмодзи в тексте сообщений;
- возможности по кастомизации: изменение стандартного изображения пользователя на собственное, изменение отображаемого имени;
- интеграция с телефонной книгой (для пользователей мобильного приложения);
- динамическое обновление данных для обеспечения быстрого получения и быстрой доставки сообщений.

1.2 Анализ существующих разработок и технологий для рассылки информационных сообщений

В первую очередь, проведём обзор существующих приложений.

Мессенджеры являются самым популярным средством общения в сети. В Топ-10 самых используемых приложений на мобильных платформах (по данным компании Kleiner Perkins Caufield Byers) шесть — это мессенджеры - WhatsApp, Facebook Messenger, LINE, Viber, Какао и WeChat соответственно. И все они имеют некоторые недостатки, такие как закрытый исходный код, реализация только под определенные платформы, слабая степень защиты.

Практически все компании наравне с электронной почтой и сотовой связью используют в альтернативные каналы связи для решения мгновенных рабочих вопросов. Это могут быть приложения для видеосвязи, мессенджеры, социальные сети, например: Skype, WhatsApp, Viber.

Однако вышеперечисленные приложения имеют ряд недостатков:

- сотрудники часто пользуются разными приложениями для обмена информацией;
- в контакт-листе мессенджеров присутствуют контакты, которые не имеют отношения к работе, например, родственники и друзья.

Для удобства коммуникаций и работы внутри компании, а также для компенсации вышеперечисленных недостатков используются корпоративные мессенджеры.

Достоинства корпоративных мессенджеров:

- повышение скорости коммуникации в решении рабочих вопросов и разделение потоков общения. Специализированные мессенджеры могут акцентировать их внимание на решении исключительно рабочих вопросов, позволяя не отвлекаться и не засорять корпоративную почту лишней информацией;
- командная работа и возможность сведения воедино нескольких информационных потоков;
- экономия внутренних технических ресурсов компании. Готовые мессенджеры, как правило, работают по принципу «облачных хранилищ» данных, и позволяют организации сэкономить место на серверах и снизить затраты на создание и обслуживание собственного сервиса коммуникации.

Был проведен поиск и анализ существующих чат-сервисов для корпоративных пользователей. На рынке существует множество программных продуктов для коммуникаций внутри рабочей команды. Ниже рассмотрены некоторые из них:

1. Staply. Мессенджер с проектным функционалом. Позволяет: составить иерархию групп; создавать задачи, задавать сроки, важность и ответственность в рабочих группах; осуществлять контроль над группами.

2. Slack. Корпоративный мессенджер с поддержкой интеграции с десятками сторонних сервисов. Объединяет в одном окне обсуждения в общих темах (каналах), приватных группах и личных сообщениях. Позволяет осуществлять видеозвонки; хранить сообщения без доступа к сети.

3. HipChat. Представляет собой групповой чат, видеочат для команд и компаний с возможностями обмена файлами и демонстрацией экрана.

4. Битрикс24. Система управления внутренним информационным ресурсом компании для коллективной работы над задачами, проектами и документами, для эффективных внутренних коммуникаций. Позволяет контролировать, делегировать, оценивать задачи; создавать резервные копии в нескольких местах; работать в нескольких комнатах одновременно.

На основе функций аналогов и поставленных задач можно выделить следующие критерии для сравнения:

- интеграция со сторонними сервисами;
- создание иерархии групп;
- распределение управляющим участников по ролям;
- возможность контроля управляющим над группами;
- резервное копирование в нескольких местах;
- работа в нескольких комнатах одновременно;
- настраиваемое оформление;
- инструменты;
- оффлайн-сообщения;
- сохранение истории сообщений.

В таблице 1.1 представлена сравнительная характеристика вышеперечисленных продуктов.

Таблица 1.1 - Сравнительная характеристика рассмотренных аналогов

Критерий	Аналоги				Примечание
	Staply	Slack	HipChat	Битрикс24	
Интеграция со сторонними сервисами		Dropbox, Google Drive, GitHub, Google Docs, Google Hangouts, Twitter и другие.	GitHub, MailChimp Heroku и другие.	MS SharePoint, MS Exchange Server, MS Outlook, Продукты Apple и Google.	Позволяет пользователям отслеживать прогресс в различных проектах с помощью одной платформы
Создание иерархии групп	+				Позволяет равномерно распределить задачи
Распределение управляющим участников по ролям					Позволяет эффективно распределить задачи
Возможность контроля управляющим над проектом	+	+	+	+	Обеспечивает контролирование выполнения задач
Резервное копирование в нескольких местах	+/-			+	Обеспечивает сохранность данных
Работа в нескольких комнатах одновременно	+			+	Позволяет решать несколько задач одновременно
Настраиваемое оформление					Позволяет индивидуализировать интерфейс
Инструменты	Блокнот, Задачи			Ежедневник, планировщик событий, календарь, график отсутствий, собрания и планерки	Позволяют оперативно выполнять задачи
Оффлайн сообщения	+	+	+		Позволяет работать с информацией без сети

Далее рассмотрим технологии создания мессенджеров для рассылки информационных сообщений.

Веб-приложение - клиент-серверное приложение, оно позволяет при помощи браузера клиенту взаимодействовать с сервером, несомненно веб-сервер отвечает за сервер.

Логика веб-приложения распределена между клиентом и сервером, данные хранят в основном на сервере, соответственно обмен информацией осуществляется по сети. Данный подход позволяет клиентам не быть зависимыми, непосредственно от определенной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами, что, несомненно, можно назвать преимуществом.

«Клиент-сервер» - сетевая или вычислительная архитектура, в ней сетевая нагрузка или задания распределяются между заказчиками услуг, соответственно клиентами и поставщиками услуг, которые называют серверами.

Технология «клиент-сервер» осуществляется непосредственно серверной и клиентской частями, из которых состоит веб-приложение.

Формирование запросов к серверу, кроме этого обработка ответов от него и реализацию пользовательского интерфейса осуществляется клиентской частью. Код клиентской части написан с использованием HTML, CSS и JavaScript.

Серверная часть получает запрос от клиента, выполняет вычисления, после этого формирует веб-страницу и отправляет её клиенту по сети с использованием протокола HTTP. Код серверной части может быть написан на любом количестве языков программирования - примеры популярных языков серверной части включают в себя PHP, Python, Ruby и C#.

Проект разработан на PHP с использованием фреймворка Yii 2. Данный фреймворк основан на использовании концепции MVC (Model-View-Controller).

Важнейшая цель употребления данной концепции заключается в

разделении бизнес-логики (модели) от её визуализации (представления, вида). Благодаря данному разделению увеличивается вероятность вторичного применения.

Данная концепция больше всего полезна, когда пользователю необходимо увидеть одни данные одновременно в разнообразных контекстах и/или с различных точек зрения. Далее рассмотрим выполняемые задачи:

- Не задевая основу модели, к ней можно прибавить несколько разных видов. К примеру, часть данных можно в одно время изобразить в виде гистограммы, электронной таблицы и круговой диаграммы;

- можно изменить реакции на действия пользователя (нажатие мышью на кнопке, ввод данных), не задевая реализацию видов, для этой цели можно воспользоваться другим контроллером;

- некоторые разработчики специализируются исключительно в одной области: это может быть разработка графического интерфейса или разработка бизнес логики. Таким образом существует возможность, что программисты, которые разрабатывают непосредственно бизнес-логику (модели), совсем не будут знать о том, какое представление будет использоваться.

Концепция MVC дает возможность произвести разделение данных, представлений и обработку действий пользователя на три отдельных компонента:

- модель (англ. Model). Данная модель даёт знания: данные и методы работы с этими данными, изменяет свое состояние, реагируя на запросы. Нет информации о визуализации знаний;

- представление, вид (англ. View). Эта модель несет ответственность за отображение информации (визуализацию). Чаще в роле представления используется форма (окно) с графическими элементами;

- контроллер (англ. Controller). Обеспечивает связь между системой и пользователем: отвечает за контроль ввода данных пользователем, в свою очередь использует модель и представление для осуществления нужной реакции.

Следует обратить пристальное внимание на то, что представление и контроллер зависят непосредственно от модели. В свою очередь, модель от них не зависит. Несомненно, это дает возможность строить модель независимо от визуального представления, кроме этого создавать несколько разнообразных представлений для одной модели.

1.3 Обоснование выбора средств разработки

HTML – является стандартным языком разметки документов во Всемирной паутине. Соответственно большую часть веб-страниц создают при помощи языка HTML (или XHTML). Язык HTML распознается браузерами, и, конечно, отображают его в виде документа, с которым удобно работать человеку. HTML представляет собой приложение («частным случаем») SGML (типового обобщённого языка разметки) и, конечно, отвечает международному стандарту ISO 8879. XHTML в свою очередь является приложением XML.

Ученый из Великобритании Тим Бернерс-Ли разработал язык HTML приблизительно в 1986—1991 годах в стенах Европейского совета по ядерным исследованиям в Женеве (Швейцария). HTML был создан для возможности обмена научной и технической документацией, которую смогут использовать люди не специализирующиеся непосредственно в области вёрстки. Несомненно, HTML хорошо справился с проблемой сложности SGML путём определения незначительного набора семантических и структурных элементов — дескрипторов, которые чаще всего называют «тегами». Несомненно, пользуясь HTML можно легко создать простой, красиво оформленный документ. Не считая упрощения структуры документа, в HTML добавлена поддержка гипертекста. Стоит отметить, что внедрение мультимедийных возможностей произошло позже.

Обработка и отображение текстовых документов, содержащие разметку на языке HTML (такие документы обычно обладают расширением .html или .htm), осуществляют специальные приложения. Браузеры дают пользователю

возможность запрашивать и просматривать веб-страницы в удобном интерфейсе (и вывода на иные внешние устройства) и, при необходимости, конечно, отправлять введенные пользователем данные на сервер.

HTML — теговый язык разметки документов. Тегами обозначают начало и конец всех элементов, из которых состоит любой документ на языке HTML. В свою очередь элементы могут быть пустыми, другими словами, они не будут содержать никакого текста и других данных (например, тег перевода строки `
`). В таком случае наиболее часто не указывают закрывающий тег. К тому же, у элементов могут быть атрибуты, устанавливающие те или иные свойства (например, размер шрифта для элемента `font`). Атрибуты указывают в открывающем теге.

Web-messaging. Средство общения между сущностями, свободное от междоменных ограничений. Данная технология обеспечивает возможности обмена сообщениями между документами из разных местоположений - подобным образом данные процессы организовываются более защищенно, чем раньше, когда не существовало прямого способа и применялось большое количество так называемых «костылей». Однако, все еще есть несколько требования и рекомендаций, которые следует иметь ввиду:

- отправляя сообщение, необходимо явно специфицировать ожидаемый адрес отправки в качестве второго аргумента для `postMessage`, с целью предотвращения посылки в неизвестность после перенаправления или какого-либо другого события

- принимающая страница всегда должна выполнять следующие действия:

1. Проверять атрибут `origin`, чтобы удостовериться, что сообщение пришло из ожидаемого источника;

2. Проверять формат атрибута `data`, чтобы удостовериться, что он был корректно сформирован;

- полный контроль за атрибутом `data` отсутствует - простая CSS уязвимость в посылающей странице позволит злоумышленнику отправить

сообщения любого формата и содержимого;

- обе страницы должны работать с сообщениями только как с data. Не следует работать с сообщениями при помощи eval() или InnerHTML, так как это создаст XSS уязвимость;

- чтобы назначить значение элементу, вместо использования небезопасных методов вроде element.innerHTML = data следует использовать более безопасный вариант element.textContent = data;

- следует проверять адрес источника в соответствии с ожидаемыми FQDN.

- если необходимо добавить дополнительный контент и разрешить пользовательские скрипты, следует рассмотреть возможность использования JavaScript-фреймворка вроде Google C'aja или технологии sandboxed frames.

Cross Origin Resource Sharing. Совместное использование ресурсов между разными источниками. При его использовании следует соблюдать следующие требования:

- необходимо проверять URL адреса, переданные XMLHttpRequest.open. Современные браузеры позволяют данным URL быть кросс-доменными, это может привести к атаке типа code injection;

- необходимо проверять, что URL с выставленными Access-Control-Allow-Origin: не содержат никакой конфиденциальной информации, или любой другой информации, которая могла бы помочь атакующему. Следует использовать заголовок Access-Control-Allow-Origin только на отдельных URL, которые должны будут использоваться при междоменных взаимодействиях;

- для заголовка Access-Control-Allow-Origin должны использоваться только доверенные домены. Предпочтительно использован не «белых списков» (не следует использовать * или пренебрегать проверками содержимого заголовка);

- использование стандартных мер противодействия CSRF атакам также необходимо;

- в то время, как RFC рекомендует использование предварительного

запроса с параметром OPTIONS, текущие имплементации могут не выполнять данный запрос, соответственно для GET- и POST-запросов проверка доступа обязательна к использованию.

WebSockets. Не следует обеспечивать обратную совместимость, и поддерживать протоколы версий hybi-00 и ниже. Рекомендуемой к использованию версией сокетов является RFC 6455. Не смотря на то, что достаточно легко можно организовать туннельное соединение для TCP через веб-сокеты, подобная практика позволяет атакующему, получившему доступ к браузеру, получить доступ к такому туннельному соединению при CSS атаке — не следует прибегать к подобной практике;

По причине гою, что данный протокол не поддерживает авторизацию или аутентификацию, протоколы уровня приложений должны обеспечивать выполнение данных функций. Аналогично предыдущему разделу, не следует работать с телом сообщений как-либо иначе, как с двоичными данными. Для защиты от MitM атак, следует использовать веб-сокеты через SSL/TLS. Сервер веб-сокетов должен верно обрабатывать некорректные или потенциально опасные входные данные, проверка входных данных должна использоваться для всех соединений;

Следует проверять заголовок Origin в процедуре «рукопожатия» веб-сокетов. По причине того, что к клиенту веб-сокетов в браузере может быть получен доступ с помощью JavaScript-вызовов, защита от CSS должна использоваться для каждого соединения;

CSS - формальный язык. То какой будет внешний вид документа, написанного с использованием языка разметки, описывают при помощи CSS.

Рассматриваемый язык чаще всего используют как средство описания, того как будет оформлен внешний вид веб-страниц, безусловно, написанных с помощью языков разметки HTML и XHTML, кроме того можно применять к любым XML- документам, например, к SVG или XUL.

CSS создатели веб-страниц использует для задания шрифтов, цветов, расположения отдельных блоков и других аспектов внешнего вида веб-страниц.

CSS язык разработали, чтобы разделить описание логической структуры веб-страницы (осуществляемое с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (осуществляемое непосредственно с помощью формального языка CSS). Подобное разделение может повысить доступность документа, предоставить большую гибкость и возможность управления его представлением, вдобавок снизить сложность и повторы в структурном содержимом. Кроме этого CSS даёт возможность представить один документ в разнообразных стилях или методах вывода, например экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

В качестве языка разработки подключения к базе данных был выбран язык PHP.

PHP - это язык программирования для динамической генерации HTML кода со стороны сервера. В него встроена поддержка базы данных MySQL, это дает возможность полагать, что выбранная связка MySQL-PHP наиболее оптимальная. PHP-скрипты истолковываются, и выполняются на сервере.

PHP предпочтителен по следующим характеристикам:

- в модулях PHP все запускается в области памяти, выделенной программе операционной системой;
- интеграция PHP с выбранной СУБД MySQL. Существует множество утилит на PHP для работы с базами данных MySQL, где реализуется набор свойств наиболее полный в сравнении с другими базами данных. Имеются весьма полезные встроенные функции, которые отсутствуют в других базах данных. К существенным преимуществам PHP, безусловно, относится поддержка широкого круга баз данных: Oracle, Microsoft SQL server, MySQL и другие;
- возможность оперативно реагировать и исправлять недоработки за счет отсутствия временных проблем с исправлением внутренних ошибок.

Фреймворк Yii2. Проект написан с использованием фреймворка Yii2. Yii2 -

это высокопроизводительный компонентный PHP фреймворк, предназначенный для быстрой разработки современных веб-приложений. Слово Yii в китайском языке означает «простой и эволюционирующий». Также Yii может расшифровываться как акроним «Yes It Is».

Yii2 является универсальным фреймворком и задействуется во всех типах веб-приложений. Благодаря тому, что он имеет компонентную структуру и возможность поддержки операций кэширования, данный фреймворк в особенности хорошо подходит для разработки таких крупных проектов, как, к примеру, форумы, порталы, CMS, Интернет-магазины или мессенджеры.

Особенности данного фреймворка заключаются в следующем:

- Yii2 использует архитектурный паттерн MVC для организации кода;
- Yii2 не усложняет дизайн кода, а придерживается философии простого кода;
- Yii2 является по сути full-stack фреймворком и включает непосредственно в себя только проверенные и хорошо зарекомендовавшие инструменты, такие как ActiveRecord для NoSQL и реляционных баз данных, поддержку процедуры REST API, многоуровневое кэширование и др.;
- Yii2 является расширяемым фреймворком, что означает, что его можно настроить или, при необходимости, заменить в нем практически любую часть основного кода. Применяя архитектуру расширений, можно очень просто делиться фрагментами кода либо использовать код сообщества;
- фреймворк Yii2 обладает высокой производительностью.

Далее рассмотрим выбранную систему управления базами данных (СУБД). В данной работе таковой является СУБД MySQL. Выбранная система будет являться идеальной совокупность программных и лингвистических средств общего или специального назначения, и даст возможность полномасштабно обеспечивать управление созданием и использованием таблиц данных, справочников и отчетов.

Система управления базами данных MySQL является бесплатной,

непосредственно свободно распространяемой СУБД. Данная система имеет значительно функциональную клиент-серверную архитектуру: к серверу MySQL имеют возможность обращаться разнообразные клиентские приложения, в том числе и с возможностью доступа с удаленных компьютеров. Существует визуальный интерфейс для облегчения работы с базами данных MySQL — PhpMyAdmin. СУБД MySQL версий 5.0 и выше соответствуют стандарту структурированного языка запросов SQL, поэтому она совместима с другими базами данных. Основными языками разработки являются C/C++.

Рассмотрим функциональные особенности системы управления базами данных MySQL:

- реляционная СУБД;
- имеет открытый код;
- система имеет кроссплатформенное строение;
- обладает большим числом программных интерфейсов (API), с помощью которых непосредственно к базе данных MySQL имеют возможность подключения разнообразные приложения, созданные на следующих языках программирования: C/C++, Eiffel, Java, Perl, PHP, Python, Tcl, ODBC, NET и Visual Studio.

- обладает значительными техническими характеристиками, такими как многопользовательский доступ, многопоточность, масштабируемость, быстрое действие;

- располагает развитой системой обеспечения безопасности функционирования и разграничения доступа к информационным ресурсам на основе системы привилегий.

Таким образом, в ходе проведения исследования были рассмотрены теоретические аспекты организации управления в предприятиях с иерархичной организацией, а также разработана платформа на принципе иерархии, с помощью которой нет необходимости заикливаться на отдельных видах организационных структур.

2 Методология проектирования информационной системы

2.1 Функциональное проектирование исследуемого процесса

Разрабатываемое приложение будет позволять осуществлять обмен данными между зарегистрированными пользователями, обмениваться заданиями и файлами, объединять пользователей в рабочие группы и администрировать работу системы.

Разрабатываемая информационная система рассылки информационных сообщений должна, по возможности, наискорейшем образом, восстанавливать состояние полного функционала при корректном перезапуске аппаратных средств. В случае аварии необходимо обеспечить сохранность всех данных Системы. В рамках этого предусматривается вероятность организации автоматического (планового) и (или) ручного копирования данных в резервное хранилище базы данных средствами системного и базового программного обеспечения (операционной системы, системы управления базами данных), непосредственно входящего в состав программно-технического комплекса разрабатываемой информационной системы.

Также, в обязательном порядке предусматривается возможность корректного восстановления данных за период до при помощи их импорта из внешних источников или повторного ввода.

Для обеспечения защиты от несанкционированного доступа необходимо предусмотреть авторизацию пользователя при входе в систему.

Администратор при резервном копировании и архивировании информации базы данных должен защитить ее служебным паролем.

В системе требуется предусмотреть возможность разграничения полномочий пользователей. Необходимо, чтобы администратор мог распределять уровни доступа к информации каждого из пользователей и наделять им права доступа к системе.

В современных разнообразных предприятиях, существует, множество различных способов информирования сотрудников, тем не менее, по большей

части они не практичны. Среди них: конечно, оповещение по телефону, рассылка емс, социальные сети, кроме этого электронная почта. Перечисленные способы, несомненно, имеют как положительные, так и отрицательные стороны, но нам необходимо создать универсальное средство, которое будет максимально удобным и, несомненно, эффективным решением. Потому как этих способов довольно немало была создана общая SADT-модель (IDEF0) «как-есть» (рисунок 2.1).

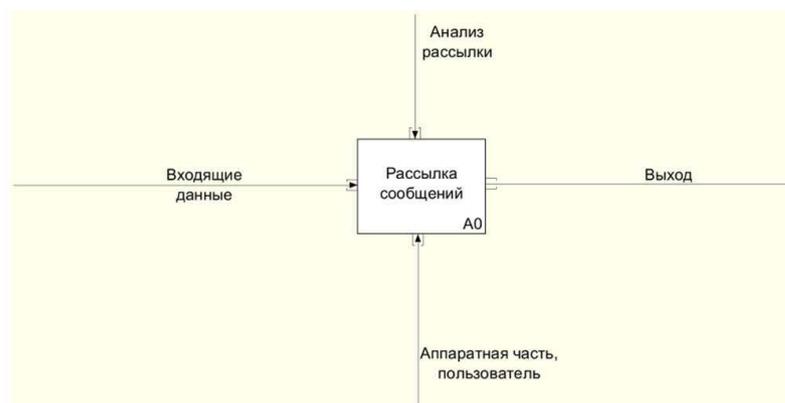


Рисунок 2.1 - Родительская диаграмма «как-есть»

На дочерней диаграмме (рисунок 2.2) видно, что модель достаточно проста, но из этого следуют и минусы. Например, неверный выбор адресата или в случае использования электронной почты нет полной гарантии, что письмо дойдет или не попадет в папку со спамом.

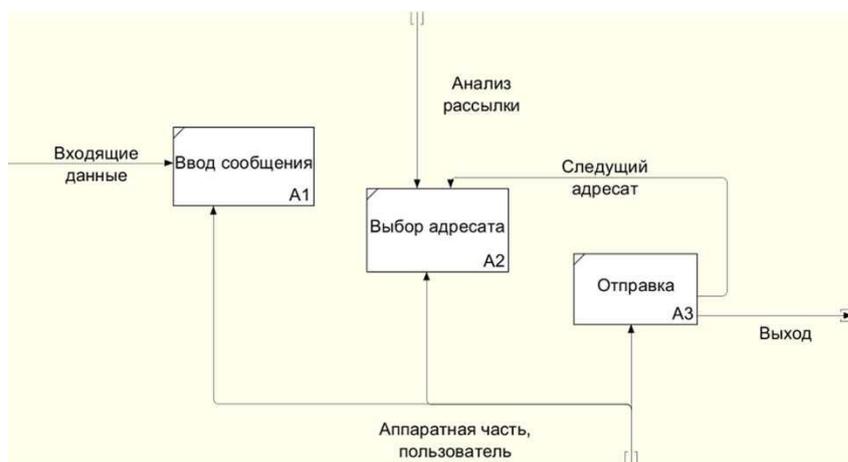


Рисунок 2.2 - Дочерняя диаграмма «как-есть»

На предоставленной функциональной модели (рисунки 2.3, 2.4) можно увидеть, как реализуется главная функция отправка сообщения. Работа пользователя с приложением: осуществление непосредственного входа под своей учетной записью или регистрация, если это необходимо, в свою очередь, безусловно, есть возможность создания группы, в которые можно пригласить нужных пользователей и непосредственная отправка сообщений.

На родительской диаграмме (рисунок 2.3) бизнес-функцией является отправление сообщения, управление осуществляется с помощью анализа доступных возможностей, а механизмом в данном случае является пользователь и аппаратная часть. На входе и выходе осуществляется запуск приложения и завершение работы, соответственно.

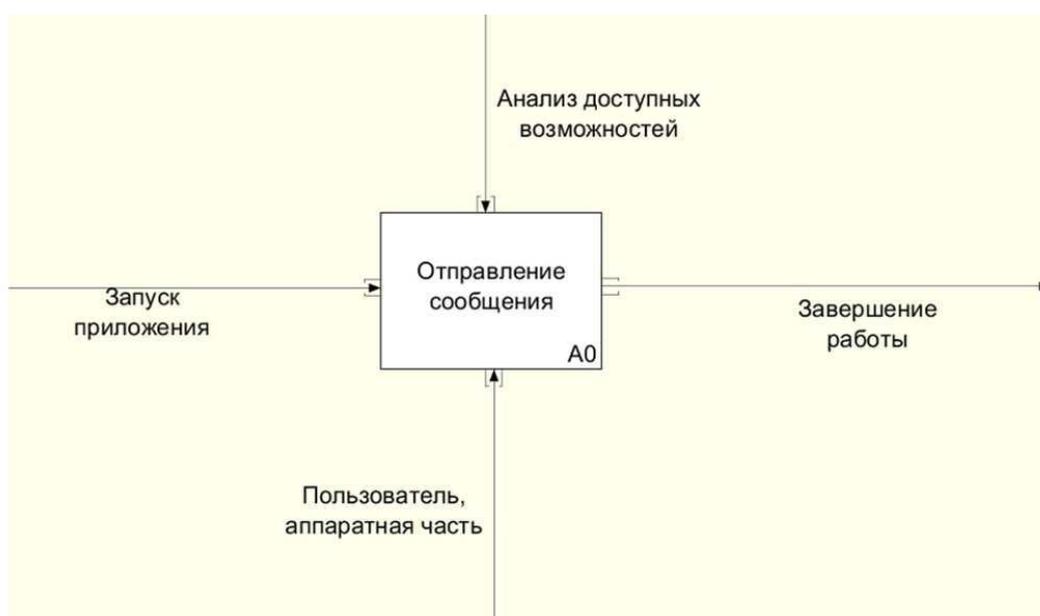


Рисунок 2.3 - Родительская диаграмма «как- должно быть»

Диаграмма прецедентов (Use Case Diagram) - диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Данная диаграмма предназначена для определения функциональных требований к программному продукту.

На дочерней диаграмме (рисунок 2.4) указаны все функции приложения,

а так же возможные действия в проектируемом приложении.

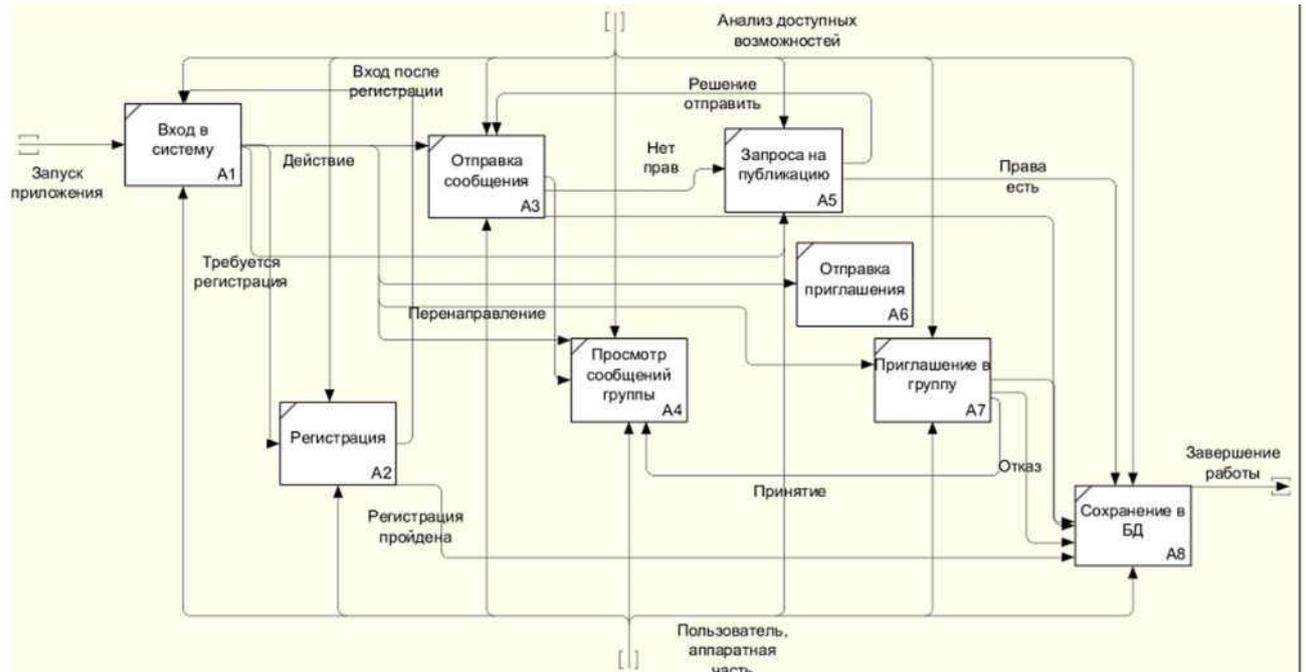


Рисунок 2.4 - Дочерняя диаграмма «как-должно быть»

Базовыми элементами диаграммы прецедентов являются акторы и прецеденты. Актор - роль, которую играют внешние сущности. Прецедент - последовательности действий, которые система или другая сущность могут выполнять в процессе взаимодействия с акторами. Были выделены акторы: администратор и пользователь. Диаграмма работы с профилем пользователя представлена на рисунке 2.5. Пользователь может просматривать и редактировать свой профиль, просматривать проекты, сообщения, организацию.



Рисунок 2.5 - Use case диаграмма работы с профилем пользователя

Диаграмма работы администратора представлена на рисунке 2.6. Администратор ведет учет существующих пользователей, может редактировать список должностей, пользователей, а также может редактировать существующие проекты, менять статус пользователей.

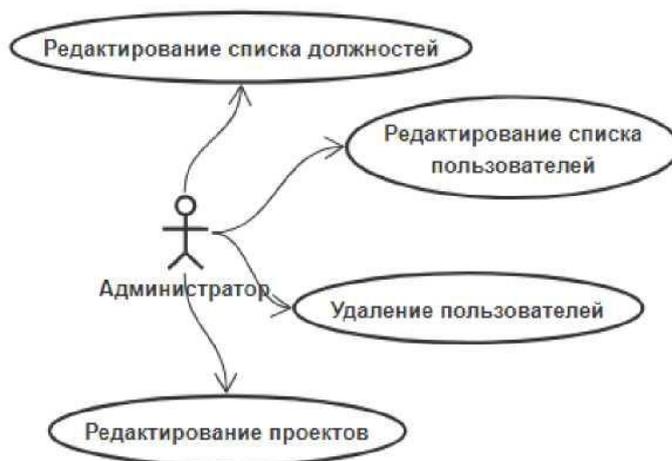


Рисунок 2.6 - Use case диаграмма работы администратора

Диаграмма работы с проектами представлена на рисунке 2.7. Пользователь может создавать, редактировать проекты, добавлять участников, документы, рабочие группы проекта, распределять задания рабочих групп, создавать, редактировать беседы рабочих групп.



Рисунок 2.7 - Use Case диаграмма работы с проектами

Диаграмма работы с сообщениями представлена на рисунке 2.8. Пользователь может создавать беседы, писать и удалять сообщения, добавлять участников. Беседы могут быть в рабочих группах и просто между пользователями.

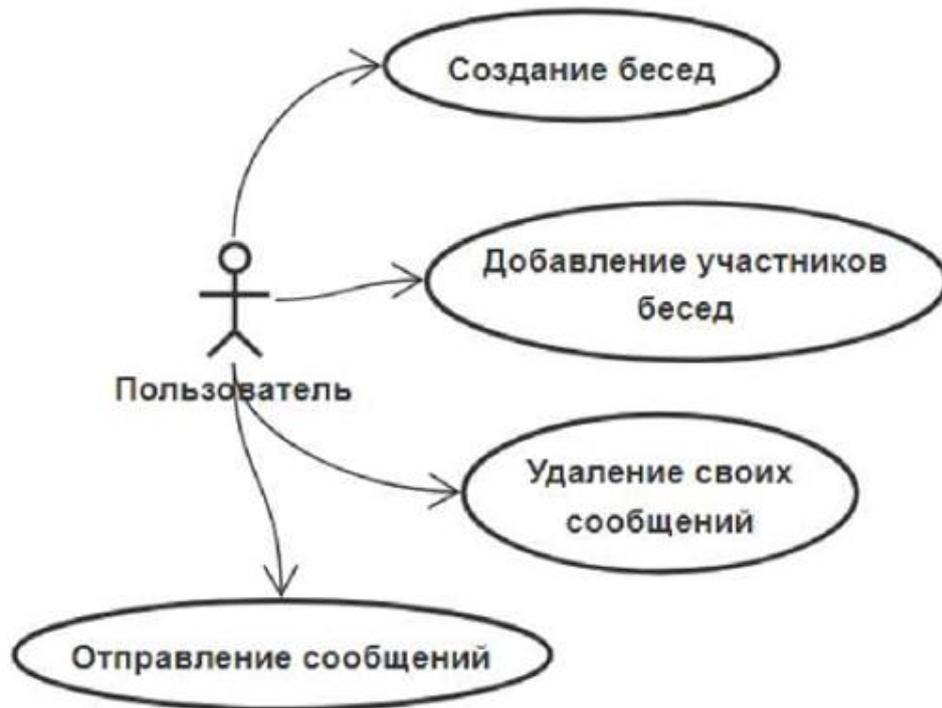


Рисунок 2.8 - Use case диаграмма работы с сообщениями

Диаграмма классов - диаграмма, на которой изображена совокупность статических или декларативных элементов модели, таких как классы с атрибутами и операциями, в свою очередь объединяющие их отношения.

На диаграмме классов (рисунок 2.9) представлены следующие классы: Процесс обработки - управляющий класс, который организывает все остальные классы, имеет операцию - взаимодействие, кроме этого атрибут - Алгоритм, другими словами способ обработки;

Диаграмма деятельности визуализирует особенности реализации операций классов, когда нужно представить алгоритмы их выполнения. Диаграмма деятельности (рисунок 2.9) реализована с помощью дорожек, чтобы ясно видеть, в каком состоянии, находится система на разных этапах и

соответственно какое подразделение за него отвечает.

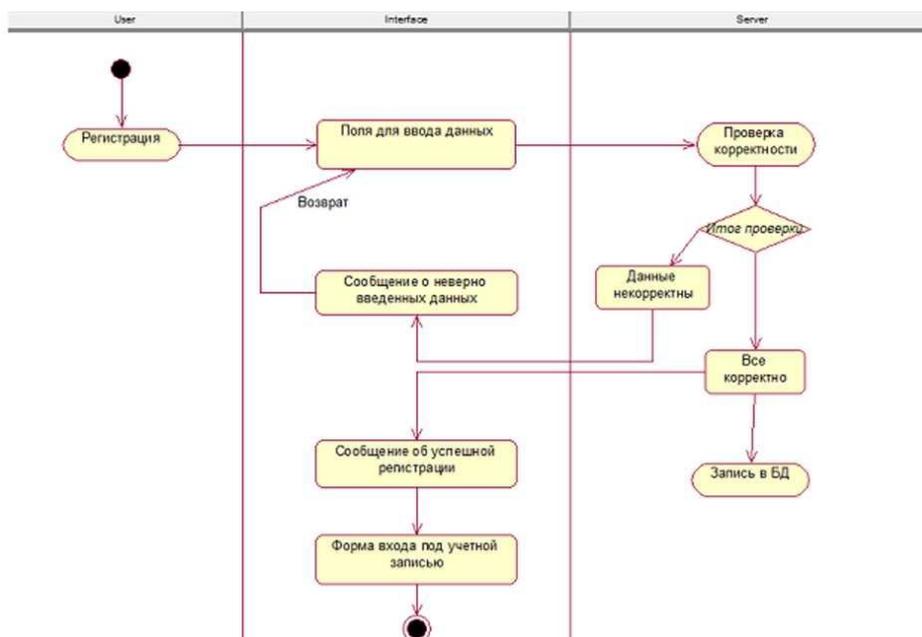


Рисунок 2.9 - Диаграмма деятельности (Регистрация)

Регистрируясь, пользователь должен произвести ввод своих данных, которые программа проверит на точность (запрет специальных символов, проверка обязательных полей) и доступность (доступность логина, регистрации на вводимую почту). В случае ошибки в данных, пользователя вернут на форму регистрации с соответствующим оповещением. Далее корректно введенные данные записываются в базу данных, и будет выведено сообщение об успешной регистрации, с последующей возможностью войти под своей учетной записью.

Внимательно разберем один из возможных вариантов использования приложения пользователем. Допустим, ему необходимо проверить в группе наличие новых сообщений. Для этого ему нужно запустить приложение и войти под своей учетной записью, если не произвел этого ранее. Далее на общей форме (дорожка интерфейс) пользователь может ознакомиться со списком доступных ему действий, которые были проверены ранее (дорожка сервер), и, наконец, выбрав просмотр сообщений группы, он переходит на форму с её сообщениями (рисунок 2.10).

Благодаря диаграмме деятельности (рисунок 2.10) можно увидеть состояния, в которых находится система и ознакомиться с возможными

На представленной диаграмме (рисунок 2.11) показано, что в состав приложения входит модуль главной программы (MainProgram), которая в свою очередь реализуется при помощи модуля подпрограммы проверки (Checking.cpp), и связываются с сервером, который непосредственно взаимодействует с Базой данных, информация соответственно будет выводиться через интерфейс.

Диаграмма развертывания (рисунок 2.12) позволяет полностью физически представить проектируемую систему, так как показывает на какой платформе и каких вычислительных средствах написана система.

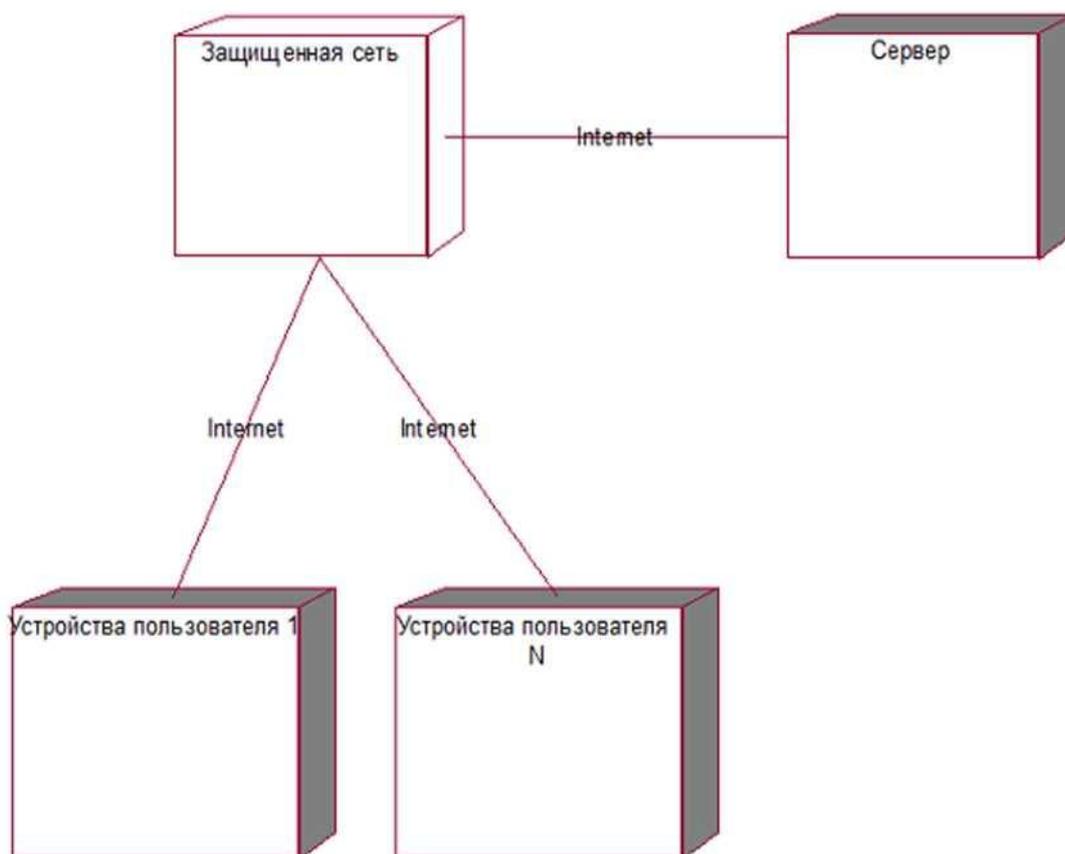


Рисунок 2.12 - Диаграмма развертывания

Данная диаграмма показывает, что устройства пользователей подключаются к защищенной сети с помощью сети интернет и производят взаимодействия с сервером. Сервер, несомненно, подобным образом

взаимодействует с устройствами пользователей.

2.2 Реализация базы данных

В проектируемом приложении были созданы следующие таблицы, чтобы хранить данные: пользователи, общая информация, сообщения, группы, пользователи в группе N и пользователи в группе N. Она отображает таблицы, хранящие данные проектируемого приложения. База данных приведена к третьей нормальной форме.

Для работы приложения была создана база данных, которая содержит 17 таблиц. Диаграммы базы данных изображена на рисунке 2.13. Таблица 2.1 содержит описание таблиц базы данных.

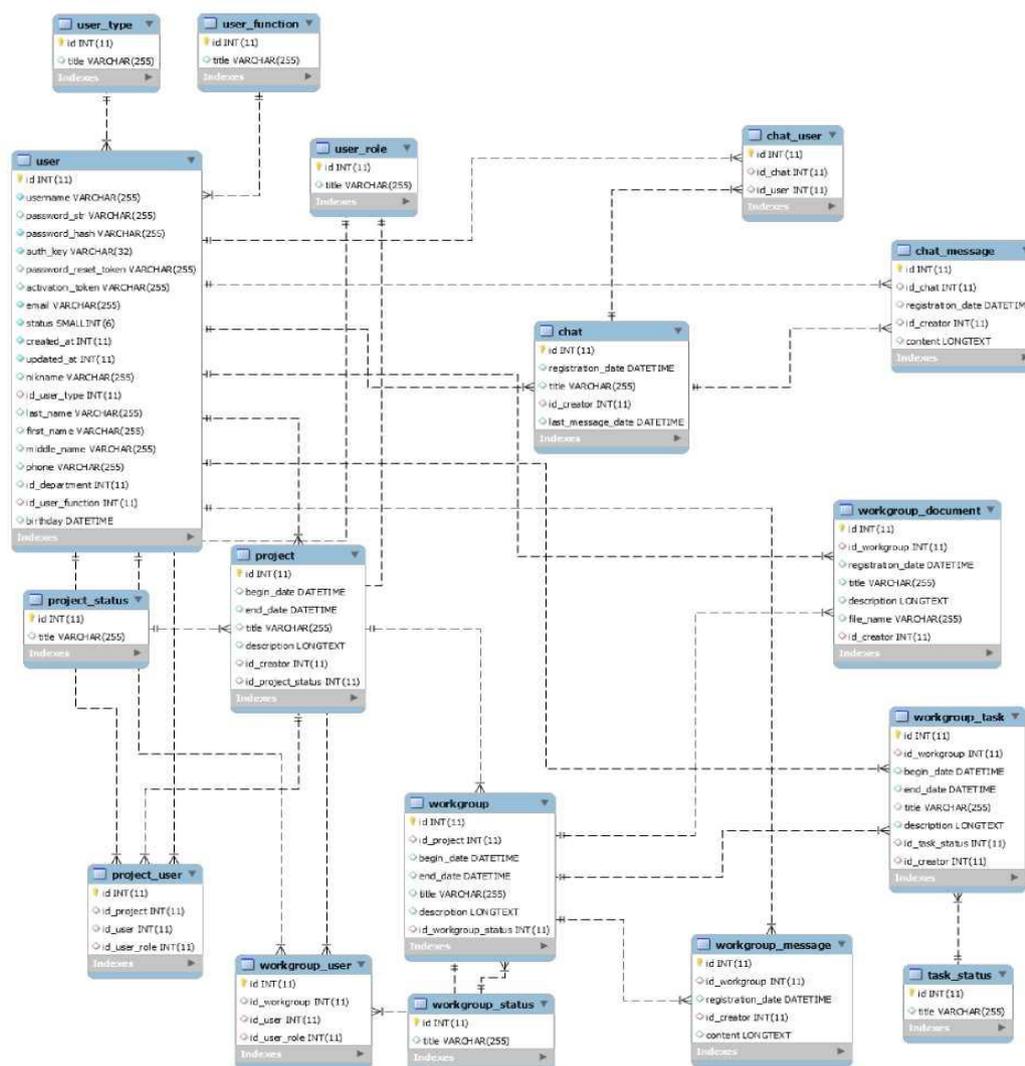


Рисунок 2.13 - Диаграмма базы данных

Таблица 2.1 - Описание таблиц базы данных

Тип модуля	Название таблицы	Описание
Справочники	project status	Статус проекта
	task status	Статус задачи
	user_role	Роль пользователя внутри группы
	workgroup status	Статус рабочей группы
	user	Пользователи
	user type	Тип пользователя
Таблицы данных	user function	Должность пользователя
	Chat	Чаты пользователя
	chat message	Сообщения чата
	chat user	Состав чата
	project	Проекты
	project user	Состав участников проекта
	workgroup	Рабочие группы проекта
	workgroup document	Документы рабочей группы
	workgroup message	Сообщения рабочей группы
	workgroup task	Задачи рабочей группы
	workgroup user	Состав рабочей группы

Проект разработан на PHP с использованием фреймворка Yii 2. Данный фреймворк основан на использовании концепции MVC (Model-View-Controller). Основная цель применения этой концепции состоит в разделении бизнес-логики от её визуализации.

Логика ввода находится в контроллере, бизнес логика в модели, а пользовательский интерфейс располагается в модели.

Для каждой таблицы базы данных создается модель (Model из Model-View-Controller). Все модели проекта располагаются в директории «models».

Большинство контроллеров проекта (кроме контроллера ClassifierController) реализует CRUD - (англ. create, read, update, delete — «создание, чтение, обновление, удаление») сокращённое именование 4-х базовых функций, используемых при работе с наборами данных.

Консоль управления Yii2 позволяет для конкретной модели выполнить генерацию CRUD. Генератор CRUD создает контроллер, набор действий контроллера («Index», «View», «Create», «Update», «Delete») и ряд видов («View») соответствующих действиям.

Таблица 2.2 содержит описание базовых действий контроллеров.

Таблица 2.2 - Описание базовых действий контроллеров

Название действия	Описание действия
Index	Метод просмотра списка всех записей
View	Метод просмотра записи
Create	Метод добавления записи
Update	Метод редактирования записи
Delete	Метод удаления записи

Контроллеры проекта находятся в папке «controllers». Представления, соответствующие действиям контроллеров, располагаются в папке «views». В таблице 2.3 описаны контроллеры системы.

Таблица 2.3 - Описание контроллеров системы

Тип модуля	Название модели	Название контроллера	Действия контроллера
Системные	Использует различные модели (модели фреймворка Yii 2.0)	SiteController	Index (главная страница системы), Login (вход), Logout (выход), Signup (регистрация), RequestPasswordReset (отправка пользователю на email ссылки для сброса пароля), ResetPassword (метод ввода нового пароля)
Справочники	ProjectStatus	ProjectStatusController	Index, View, Create, Update, Delete
	TaskStatus	TaskStatusController	Index, View, Create, Update, Delete
	UserRole	UserRoleController	Index, View, Create, Update, Delete
	WorkgroupStatus	WorkgroupStatusController	Index, View, Create, Update, Delete
	User	UserController	Index, View, Create, Update, Delete
	UserType	UserTypeController	Index, View, Create, Update, Delete
Данные	UserFunction	UserFunctionController	Index, View, Create, Update, Delete
	Chat	ChatController	Index, View, Create, Update, Delete
	ChatMessage	ChatMessageController	Index, View, Create, Update, Delete
	ChatUser	ChatUserController	Index, View, Update, Delete Create,
	Project	ProjectController	Index, View, Update, Delete Create,

Окончание таблицы 2.3

	ProjectUser	ProjectUserController	Index, View, Update, Delete	Create,
	Workgroup	WorkgroupController	Index, View, Update, Delete	Create,
Данные	WorkgroupDocument	WorkgroupDocumentController	Index, View, Update, Download	Create, Delete,
	WorkgroupMessage	WorkgroupMessageController	Index, View, Update, Delete	Create,
	WorkgroupTask	WorkgroupTaskController	Index, View, Update, Delete	Create,
	WorkgroupUser	WorkgroupUserController	Index, View, Update, Delete	Create,

Работа пользователя с системой осуществляется через обращения к действиям контроллеров. Пример обращения пользователя к действию «view» (просмотр) контроллера «project» (проект): <http://corpchat.ru/project/view/1>

Т.е. в данном случае обращение пользователя к действию системы осуществляется через GET запрос протокола HTTP. Результатом данного действия является html-код, который генерирует сервер на запрос пользователя.

2.3 Проектирование программного приложения

Работа в системе начинается с загрузки главной страницы. Если пользователь не авторизован, ему предлагается войти в систему или зарегистрироваться.

Для начала работы с приложением пользователю необходимо пройти регистрацию. Для этого пользователю необходимо заполнить его основные данные, фамилию имя и отчество, адрес электронной почты, логин и пароль. Страницы регистрации пользователя представлена на рисунке 5.

На рисунке 6 представлена форма авторизации пользователя в системе.

Для этого ему необходимо ввести логин и пароль.

Корпоративный мессенджер

Вход

Главная / Регистрация

Регистрация

Пожалуйста, заполните следующие поля, чтобы зарегистрироваться:

Имя

Фамилия

Имя отчество

Дата рождения

Телефон

Должность

Email

Логин

Пароль

Подтверждение пароля

РЕГИСТРАЦИЯ

Рисунок 2.14 - Страница регистрации пользователя

Корпоративный мессенджер

Вход

Главная / Вход

Вход

Пожалуйста, заполните следующие поля для входа:

Логин

Пароль

Запомнить меня

Если вы забыли пароль, то можно его [восстановить](#).

[Я ещё не зарегистрирован](#)

ВХОД

Рисунок 2.15 - Вход в систему

Страница, содержащая данные о пользователе, представлена на рисунке 2.16. Здесь имеются такие данные, как имя пользователя, его контактная информация, должность и тип пользователя.

Также предусмотрена возможность редактирования профиля пользователя.



Мои данные

+	Фамилия	Яковлев
	Имя	Владислав
	Отчество	Сергеевич
	Дата рождения	<i>(не задано)</i>
	Телефон	+7 988 162 49 33
	Email	jakavlad43@gmail.com
	Логин	user
	Должность	Стажёр
	Тип пользователя	Пользователь
	Статус	Активен

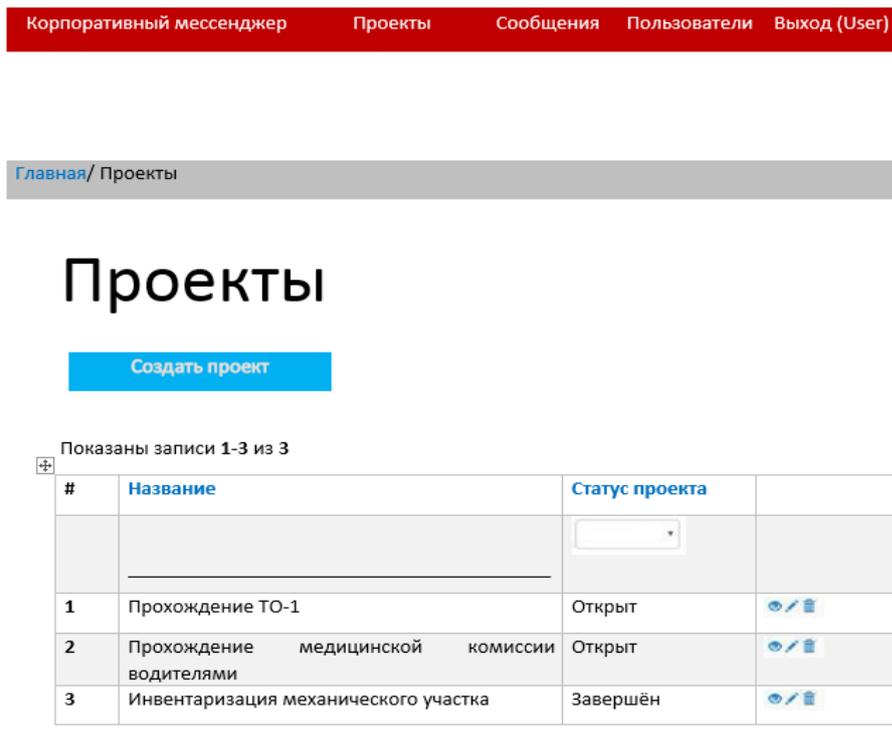
[Редактировать](#)

© Корпоративный мессенджер 2019

Рисунок 2.16 - Профиль пользователя

Переход на страницу, содержащую информацию о проектах пользователя (рисунок 2.17) возможен при нажатии пункта меню «Проекты». На данной

странице можно просмотреть проекты пользователя, их статус.



© Корпоративный мессенджер 2019

Рисунок 2.17 - Список проектов пользователя

Для создания нового проекта необходимо нажать на кнопку «Создать проект» на странице списка проектов. На странице создания проекта, изображённой на рисунке 2.18, необходимо заполнить такие поля, как название проекта, его описание и выбрать его статус.

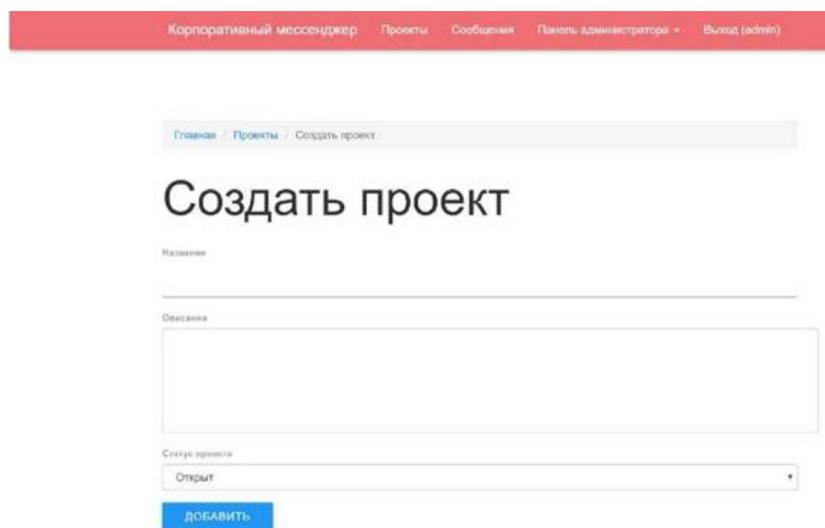


Рисунок 2.18 - Создание проекта

Переход на страницу просмотра информации о проекте, которая изображена на рисунке 2.19, возможна при нажатии кнопки просмотра проекта на странице со списком проектов пользователя или после создания нового проекта. Здесь содержится название проекта, его описание, статус, даты создания и закрытия проекта и его создатель. Проект можно редактировать или удалить при помощи нажатия на соответствующие кнопки.

The screenshot shows a navigation bar with links: 'Корпоративный мессенджер', 'Проекты', 'Сообщения', 'Пользователи', and 'Выход (User)'. Below it is a breadcrumb trail: 'Главная/ Проекты/ Прохождение ТО-1'. The main heading is 'Прохождение ТО-1'. A toolbar contains icons for menu, edit, delete, users, and refresh. A table displays project details:

Название	Прохождение ТО-1
Описание	Прохождение технического обслуживание подвижного состава АО Туапсинское АТП
Статус проекта	Открыт
Дата регистрации	2019-12-12 16:20:13
Дата закрытия	(не задано)
Создавал	Главный механик

© Корпоративный мессенджер 2019

Рисунок 2.19 - Данные проекта

На странице, которая содержит список участников проекта (рисунок 2.20), имеются такие функции, как добавление участников проекта, удаление участников из проекта, а также возможен поиск пользователей по имени и типу

ПОЛЬЗОВАТЕЛЯ.

Участники проекта

Добавить участника проекта

Показаны записи 1-3 из 3

#	Пользователь	Тип участника	
	<input type="text"/>	<input type="text"/>	
1	Юрьев Константин Евгеньевич	Администратор	 
2	Майборода Павел Юрьевич	Пользователь	 
3	Симонян Екатерина Аликовна	Пользователь	 

© Корпоративный мессенджер 2019

Рисунок 2.20 - Список участников проекта

Также имеется такая функция, как рабочие группы проекта. Рабочие группы проекта можно просмотреть на соответствующей странице (рисунок 2.21).

Рабочие группы

Создать рабочую группу

Показаны записи 1-1 из 1

#	Название	Статус группы	
	<input type="text"/>	<input type="text"/>	
1	Рабочая группа 1	Действующая	 

© Корпоративный мессенджер 2019

Рисунок 2.21 - Рабочие группы проекта

Создание новой рабочей группы возможен при нажатии на соответствующую кнопку. После необходимо заполнить соответствующую форму создания рабочей группы, которая изображена на рисунке 2.22.

Корпоративный мессенджер Проекты Сообщения Пользователи Выход (User)

Главная/ Проекты/ Прохождение ТО-1/Рабочие Группы/Создать рабочую группу

Создать рабочую группу

Название:

Описание:

Статус группы:

Действующая

Добавить

© Корпоративный мессенджер 2019

Рисунок 2.22 - Создание рабочей группы

Страница рабочей группы содержит название группы, информацию о проекте, который она выполняет, даты создания и закрытия группы и статус группы. Рабочая группа может иметь свои задачи, документы, в рабочей группа возможна переписка между участниками. Также группу можно редактировать или удалить при нажатии на соответствующие кнопки.

Документы группы можно просмотреть при нажатии на соответствующую кнопку на странице рабочей группы. Добавить документ можно при нажатии на кнопку «Добавить документ». Для рабочих групп можно создавать задачи. Страница задачи содержит информацию о задаче, дату

создания, закрытия и статус. Общение в приложении возможно внутри рабочей группы и беседах, создаваемых пользователями. Список бесед пользователя изображен на рисунке 2.23.

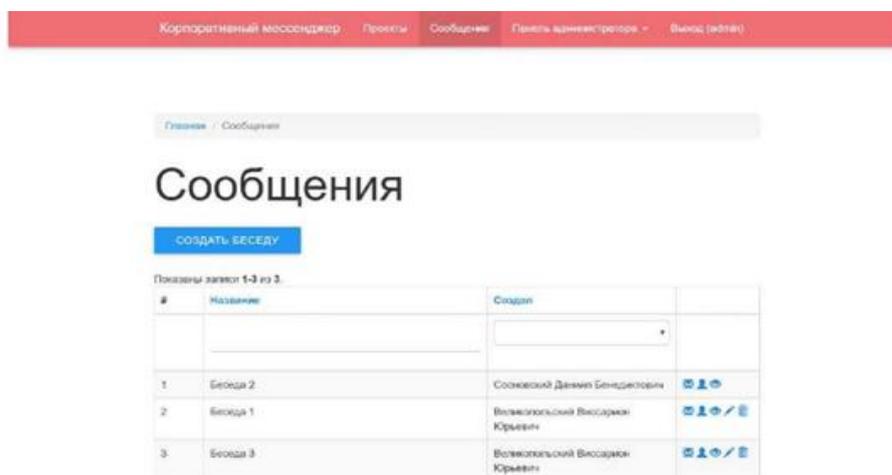


Рисунок 2.23 - Список бесед пользователя

Страница с сообщениями в беседе изображена на рисунке 2.24. В приложении есть возможность общения между несколькими пользователями. Имеется информация об отправителе сообщения, дата его отправления, также у пользователей есть возможность удалить своё сообщение, а администратор имеет возможность удалять сообщения всех пользователей.

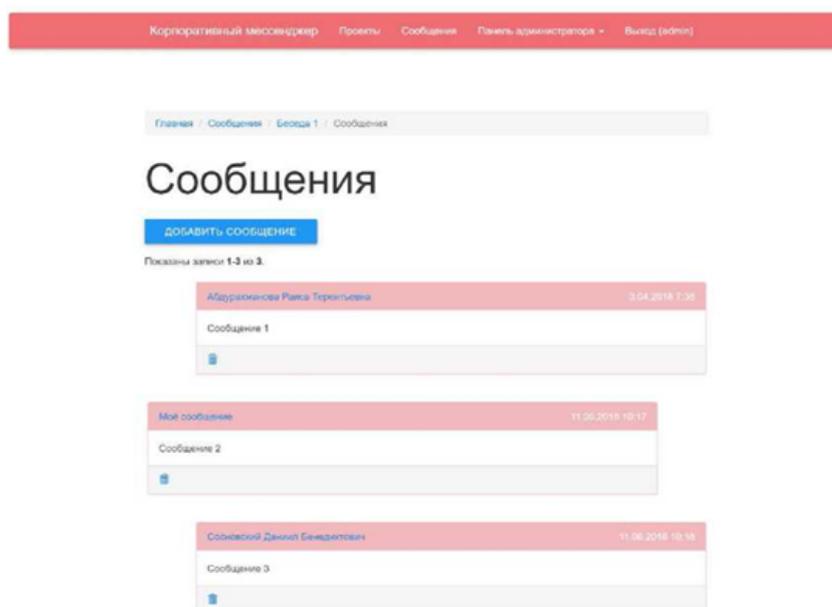


Рисунок 2.24 - Окно сообщений

Чтобы отправить сообщение участникам рабочей группы, необходимо нажать на кнопку «Создать сообщение» и заполнить форму создания сообщения (рисунок 2.25).

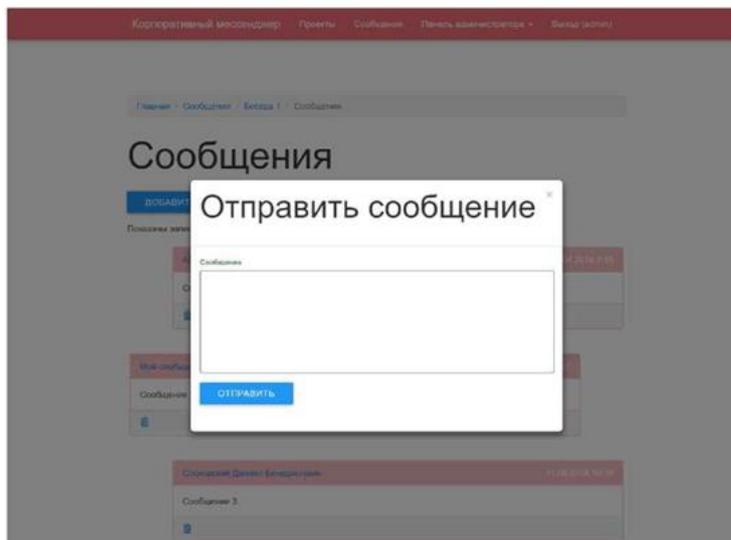


Рисунок 2.25 - Создание сообщения

В системе имеется два вида пользователей: Администратор и Пользователей. У Администраторов в панели меню имеется пункт «Панель администратора». Панель администратора содержит такие возможности, как управление списком пользователей (рисунок 2.26).

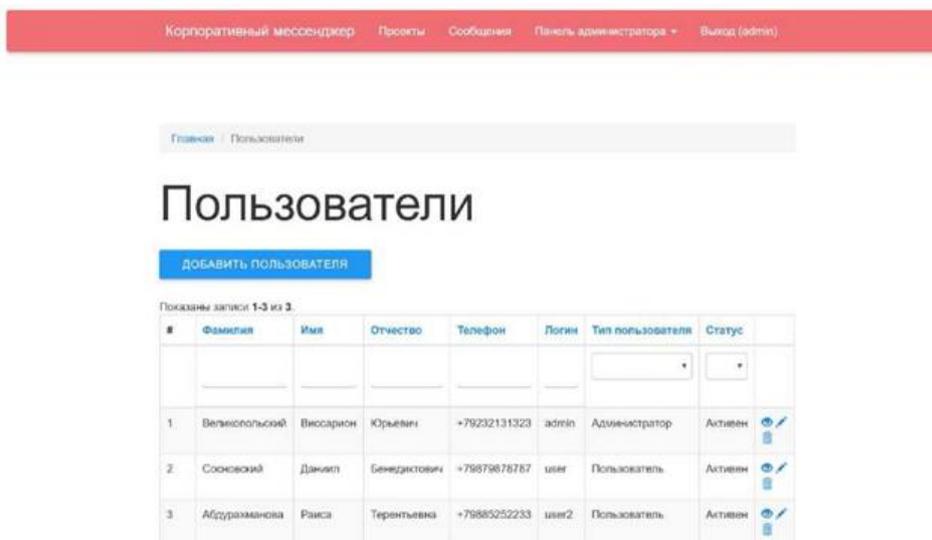


Рисунок 2.26 - Управление списком пользователей

А также управление списком должностей (рисунок 2.27).

Корпоративный мессенджер Проекты Сообщения Пользователи Выход (User)

Главная/ Должности

Должности

Добавить должность

Показаны записи 1-3 из 3

#	Название	
1	Главный механик	  
2	Механик	  
3	Водитель	  

© Корпоративный мессенджер 2019

Рисунок 2.27 - Управление списком должностей

3 Комплексная оценка реализованной информационной системы

3.1 Тестирование информационной системы

Во время разработки проекта были допущены синтаксические ошибки, которые своевременно обнаружили и исправили.

Для выявления возможных ошибок было проведено тестирование по принципу «чёрного ящика».

В ходе испытаний проведены следующие тесты:

- редактирование, добавление и удаление записей;
- реализация поиска данных в основных справочниках;
- формирование отчетов;
- построение диаграмм.

Итоги испытаний представлены в приложении 3.

Программный продукт был целиком протестирован при помощи метода «черный ящик». В ходе тестирования ошибок не обнаружено. Что означает, информационно-справочная система «Учет приема и обработки заявок» полностью готова к работе.

В свою очередь были вовремя обнаружены и ликвидированы ошибки времени выполнения.

Ошибки во время выполнения – это ошибки, появляющиеся во время работы программы. Как правило, данное происходит, когда программа пытается выполнить недопустимую операцию.

Отметим, что на уровне сдачи и ввода в эксплуатацию база данных и информационно-справочная система «Учет приема и обработки заявок» показала себя как вполне стабильное и надежное приложение для последующей работы. Поскольку в ходе тестирования программного продукта методами белого и черного ящиков не было выявлено никаких ошибок.

Далее рассмотрим надежность и безопасность информационной системы.

Известны четыре подхода к обеспечению надежности:

1. Предупреждение ошибок;

2. Само обнаружение ошибок;
3. Само исправление ошибок;
4. Обеспечение устойчивости к ошибкам.

При разработке ПС применён подход предупреждения ошибок. Цель его применения – не допустить ошибок в готовых продуктах, в данном случае – в ПС.

Природа ошибок при разработке ПС позволяет для достижения этой цели сконцентрировать внимание на следующих вопросах:

1. Борьба со сложностью;
2. Обеспечение точности перевода;
3. Преодоление барьера между пользователем и разработчиком;
4. Обеспечение контроля принимаемых решений.

В процессе разработки применяются два общих метода борьбы со сложностью систем:

1. Обеспечения независимости компонент системы;
2. Использование в системах иерархических структур.

Обеспечение независимости компонента означает разбиение системы на такие части, между которыми должны остаться по возможности меньше связей. В ПС используются иерархические структуры, которые позволяют локализовать связи между компонентами, допуская их лишь между компонентами, принадлежащими смежным уровням иерархии.

3.2 Экономические параметры реализации предлагаемой информационной системы

Обеспечение точности перевода направляется на достижение однозначности интерпретации документов различными разработчиками, кроме того пользователями разрабатываемой информационной системы рассылки информационных сообщений. Объем и перечень функций программного модуля изображены в таблице 3.1.

Таблица 3.1 – Перечень и объем функций программного модуля

№ функции модуля	Наименование функции модуля	Объем функции модуля	
		по каталогу V_i	уточненный V_{yi}
1	2	3	4
101	Организация ввода исходной информации	100	30
102	Контроль, предварительная обработка и ввод исходной информации	290	85
104	Обработка входного заказа и формирование таблиц данных	630	16
107	Организация процессов ввода/вывода информации в интерактивном режиме	170	61
206	Манипулирование каталогизирование данных	8400	59
207	Организация поиска и поиск во внутренней базе данных	5230	24
301	Формирование последовательного файла	340	27
303	Непосредственная обработка файлов	750	16
507	Обеспечение интерфейса взаимодействия между компонентами	1120	44
706	Предварительная обработка и вывод на печать файлов	390	20
	Всего	17420	382

В приведенной выше таблице в связи с использованием более совершенных средств автоматизации объемы функций были уменьшены и уточненный объем ПО (V_y) составил 382 LOC вместо 17 420. (таблица 3.2).

Таблица 3.2 – Расчет общей трудоемкости разработки ПО и численности исполнителей с учетом стадий

Показатели трудоемкости	Стадии разработки					Итого
	ТЗ	ЭП	ТП	РП	ВН	
1. Коэффициент удельных весов трудоемкости стадии разработки ПО (d)	0,1	0,2	0,3	0,3	0,1	1,0
2. Распределение нормативной трудоемкости ПО (T_n) по стадиям, чел./дн.	2,5	5	7,5	7,5	2,5	25
3. Коэффициент сложности ПО (K_c)	1,12	1,12	1,12	1,12	1,12	
4. Коэффициент, учитывающий использование стандартных модулей				0,6		
5. Коэффициент, учитывающий новизну ПО (K_n)	0,72	0,72	0,72	0,72	0,72	
6. Общая трудоемкость ПО (T_y), чел./дн.	2,0	4,0	6,0	3,6	2,0	18
7. Численность исполнителей, чел ($Ч_i$)	1	1	1	1	1	1
8. Сроки разработки, лет	0,005	0,010	0,015	0,015	0,005	0,05

Каталог аналогов программного обеспечения предназначен для предварительной оценки объема ПО методом структурной аналогии. Разработчики ПО согласованно с руководством организации могут уточнить (скорректировать) объем на основе экспертных оценок, это зависит от организационных и технических условий разработки.

По приведенной выше таблице видно, что непосредственно на разработку ПО предполагается затратить 18 чел./дней.

Тарифная ставка первого разряда на 2019 год составляет 134 руб.

Часовая з\п программиста (Тч1) = 172,8 руб.

Расчет сметы затрат на производство и реализацию единицы ПО проведем в таблице 3.3

Таблица 3.3 – Расчет сметы затрат на производство и реализацию единицы программного обеспечения

Наименование затрат	Обозначение	Сумма (руб.)	Основание
1. Материалы используемые с целью записи и тиражирования программы	Mmi	40,32	Норматив предприятия
2. Материалы для издания руководства пользователя	Mpi	134,4	-
3. Материалы для изготовления упаковки	Myi	80,64	-
4. Зарплата основная на производство и реализацию	Zoi	3695,7632	Расчет предприятия
5. Зарплата дополнительная на производство и реализацию	Zdi	739,1552	$Z_{di} = \frac{Z_{oi} \cdot H_{д}}{100\%}$
6. Социальные отчисления с основной заработной платы	Zci	1108,72	$Z_{ci} = \frac{(Z_{oi} + Z_{di}) \cdot H_{сз}}{100\%}$
7. Социальные отчисления с дополнительной заработной платы	Zei	221,74	$Z_{ei} = \frac{(Z_{oi} + Z_{di}) \cdot H_{сз}}{100\%}$
8. Расходы на техническую поддержку ПО	Zpi	698,88	Расчет предприятия
9. Оплата регистрации программного продукта	Zki	591,36	Действующая тарифная ставка
10. Затраты на оплату услуг операторов по продвижению ПО	Zki	618,24	Действующая тарифная ставка

Окончание таблицы 3.3

11. Расходы на сопровождение демонстрационной программы	Z_{cdi}	564,48	Смета расходов разработки
12.Итого переменные затраты на производство и реализацию	C_{pi}	8493,71	$C_{pi} = M_{mi} + M_{pi} + M_{yi} + Z_{oi} + Z_{di} + Z_{ci} + Z_{ei} + Z_{ni} + Z_{ki} + Z_{ki} + Z_{cdi}$
13. Расходы на поддержку сайта	Z_{ci}	134,4	Смета расходов предприятия
14. Затраты на рекламу	Z_{pi}	134,4	Смета расходов предприятия
15. Расходы на аренду помещения	Z_{ai}	134,4	Договор аренды помещения
16. Расходы на управление производством и продажи, в том числе амортизационные отчисления	P_{yi}	1344	Смета расходов предприятия
	A_0	107,52	
17. Итого постоянные расходы	C_{pi}	1747,2	$C_{pi} = Z_{ci} + Z_{pi} + Z_{ai} + P_{yi}$
18. Всего расходов на производство и продажу ПО	C_{ppi}	10240,91	$C_{ppi} = C_{pi} + C_{pi}$

Показатели, приведенные в этой таблице, учитывают все возможные статьи расхода, которые могут присутствовать в ходе разработки ПО.

Переменные затраты на производство и реализацию составили 8493,71 руб. Постоянные расходы составили 1747,2 руб. Таким образом, общая сумма расходов на производство и продажу ПО составила 10240,91 руб.

Прогнозные данные реализации проекта ПО представлены в таблице 3.4

Таблица 3.4 – Прогнозные данные реализации проекта ПО

Показатели	Единицы измерения	Значение показателей по шагам				
		2019	2020	2021	2022	2023
		t0	t1	t2	t3	t4
1. Объем производства и реализации	экземпл.		10	15	17	20
2. Прогнозная цена	руб.		15 000	15 750	16 538	17 364
3. Выручка от продаж	руб.		150 000	236 250	281 138	347 288

Окончание таблицы 3.4

4. Переменные затраты	руб.		84 937	127 406	144 393	169 874
5. Постоянные затраты	руб.		1 747	1 922	2 114	2 326
6. Затраты на производство и реализацию	руб.		86 684	129 328	146 507	172 200
7. Валовая прибыль	руб.		63 316	106 922	134 630	175 088
8. Налог на прибыль	руб.		12 663	21 384	26 926	35 018
9. Чистая прибыль	руб.		50 653	85 538	107 704	140 070
10. Амортизация	руб.		2 533	4 277	5 385	7 004
11. Капитальные вложения	руб.	11 834				
12. Коэффициент дисконтирования		1	0,8696	0,7561	0,6575	0,5718
13. Сумма ЧДД дохода нарастающим итогом	руб.	-11 834	44 047	64 675	70 816	80 092

Из приведенной таблицы по сумме чистого дисконтированного денежного дохода отлично видно, что принятие проекта приносит прибыль в размере более 44 тыс. руб. на втором году реализации. В следующем году прибыль составляет более 64 тыс. руб., затем более 70 тыс. руб., потом более 80 тыс. руб. Приведенные в последней строке таблицы денежные суммы отвечают действительности, поскольку разрабатываемую программу вполне можно реализовать в количестве 10-20 экземпляров по цене 15-17 тыс. руб. (таблица 3.5)

Таблица 3.5 – Исходные данные для расчета экономии ресурсов в связи с применением нового ПО

Наименование показателей	Обозначения	Един. измерен.	Значение показателя		Наименование источника информации
			базовый вариант	новый вариант	
1. Капитальные вложения, включая затраты пользователя на приобретение ПО	Кпр	руб.		20343,93	Договор заказчика с разработчиком
2. Затраты на освоение ПО	Кос	руб.		1183,45	Договор заказчика с разработчиком
3. Затраты на сопровождение ПО	Кс	руб.		2366,89	Договор заказчика с разработчиком
4. Затраты на укомплектование ВТ техническими средствами в связи с внедрением ПО	Ктс	руб.		1183,45	Сметы затрат на внедрение

Окончание таблицы 3.5

5. Затраты на пополнение оборотных средств в связи с эксплуатацией нового ПО	Коб	руб.		203,43	Сметы затрат на внедрение
6. Время простоя сервиса, обусловленное ПО, в день	П1, П2	мин	21	10	Расчетные данные пользователя и паспорт ПО
7. Стоимость одного часа простоя	Сп	руб.	30,52	30,52	Расчетные данные пользователя и паспорт ПО
8. Среднемесячная ЗП одного программиста	Зсм	руб.	6630,13	6630,13	Расчетные данные пользователя
9. Коэффициент начислений на зарплату	Кнз		1,5	1,5	Рассчитывается по данным пользователя
10. Среднемесячное количество рабочих дней	Др	день		22	Принято для расчета
11. Количество типовых задач, решаемых за год	ЗТ1, ЗТ2	задача	1742	1742	План пользователя 2019-2021
12. Объем выполняемых работ	А1, А2	задача	1742	1742	План пользователя
13. Средняя трудоемкость работ в расчете на 1 задачу	ТС1, ТС2	чел./час на 1 задачу	6,00	0,87	Рассчитывается по данным пользователя
14. Количество часов работы в день	ТЧ	ч	8	8	Принято для расчета
15. Ставка налога на при.	НП	%		20	

В приведенной выше таблице находятся первичные данные для расчета экономии ресурсов в связи с применением нового ПО. Можно увидеть, что используя программу, время простоя сервиса в день понизилось в два раза, а средняя трудоемкость работ в расчете на одну задачу уменьшилась практически в семь раз. Конечно, данные цифры позволяют понять, что использование программы способствует сокращению времени на привычные операции, которые изначально без использования новой программы требовали значительно больше времени (таблица 3.6).

Таблица 3.6 – Расчет экономического эффекта от использования нового ПО

Показатели	Единица измерения	Годы			
		2020	2021	2022	2023
Результаты:					
Прирост прибыли за счет экономии затрат (Пч)	руб.		121509,86	121509,86	121509,86

Окончание таблицы 3.6

Затраты:					
Приобретение ПО (Кпр)	руб.	20343,93			
Освоение ПО (К»)»	руб.	1183,45			
Сопровождение (Кс)	руб.	2366,89			
Доукомплектование ВТ техническими средствами (Ктс)	руб.	1183,45			
Пополнение оборотных средств (Коб)	руб.	203,44			
Всего затрат:	руб.	25281,15			
То же с учетом фактора времени	руб.	25281,15			
Экономический эффект:					
Превышение результата над затратами	руб.	- 25281,15	105664,97	91873,60	79892,73
То же с нарастающим итогом	руб.	- 25281,15	80383,82	172257,43	252150,16
Коэффициент приведения	единиц	1,0	0,8696	0,7561	0,6575

Приведенная выше таблица содержит данные, которые показывают доход от применения нового программного обеспечения в организации, планирующей его использовать. На предпоследней строчке таблицы можно увидеть, что доход от использования программы к 2022 году составит около 250 тыс. рублей. Стоит обратить внимание, что эти денежные суммы не могут соответствовать действительности, потому что состав приемной комиссии обрабатывает меньшее количество заявлений, чем указано в предыдущей таблице. Все же, именно такое число получили по окончанию расчетов, основанных на данных из первой таблицы.

Вывод: по приведенным в таблицах расчетам получается, что написание программы и её использование экономически выгодно как для разработчика, так и для организации, которой будет эксплуатироваться данное программное обеспечение.

Заключение

Целью данной выпускной квалификационной работы является проектирование приложения-мессенджера для рассылки информационных сообщений в предприятии с иерархичной структурой.

В ходе проведения исследования были рассмотрены теоретические аспекты организации управления в предприятиях с иерархичной организацией, кроме этого разработана платформа на принципе иерархии, при помощи которой нет необходимости заикливаться на отдельных видах организационных структур.

В результате проведенной работы получены результаты:

- проведен анализ теоретических вопросов по теме на основе научной, учебной, справочной, производственной и др. литературы, кроме того с использованием информации, полученной с помощью глобальных сетей;

- проведен анализ состояния системы управления складским хозяйством и транспортировкой;

- разработаны модели бизнес – процессов и определены направления совершенствования системы.

Использование информационных технологий для управления предприятием делает любую компанию более конкурентоспособной за счет повышения ее управляемости и хорошей адаптации к изменениям рыночной конъюнктуры.

В работе был проведен анализ приложений-мессенджеров и текущих средств отправки сообщений. Полностью проанализировав коммуникативность в структурах с иерархичной структурой автору проекта, помимо главных задач таких рассылка и просмотр информационных сообщений, удалось спроектировать следующие задачи:

- аутентификацию пользователей,
- создание групп пользователей,
- возможность приглашения пользователей в группы,

– возможность запроса на публикацию сообщения в группе, если у пользователя недостаточно прав.

В результате выполнения работы были сформированы требования к системе, изучены информация о данной предметной области, технологии, с помощью которых спроектирована требуемая система, изучены и проанализированы аналоги разрабатываемой системы. Были спроектированы база данных и сама система.

Результатом работы является разработанное web- приложение «Корпоративный мессенджер». Приложение позволяет эффективно взаимодействовать между собой участникам рабочего процесса, что позволяет оптимизировать работу как отдельного пользователя, так и целой компании.

Была спроектирована система регистрации, эффективность которой зависит от определенных фактов, им необходимо уделить внимание: человечность, корректность, подписи, потраченное время, безопасность, пароль и соц. сети.

Основной функцией является рассылка и получение оповещений и информационных сообщений. Кроме этого была поставлена задача на проектирование следующих функций: регистрация и аутентификация пользователей, создание групп пользователей, возможность приглашения пользователей в группы, возможность запроса на публикацию сообщения в группе, если у пользователя недостаточно прав.

В данной ВКР была проделана работа по анализу и изучению предприятий с иерархичной структурой, были отмечены принципы, которые будут базой для приложения, был проделан анализ и сравнение аналогов приложений мессенджеров, в ходе работы были спроектированы SADT-модели, кроме этого UML-диаграммы, была разработана модель базы данных и был осуществлен расчет трудозатрат в человеко-днях.

Данная версия программного продукта является законченным программным продуктом. Но при этом ее можно доработать до клиент-серверного приложения, что позволит вести единую базу данных для всех

подразделений предприятия, и осуществить одновременную работу нескольких пользователей.

Проектирование приложения-мессенджера выполнено полностью в соответствии с заданием. В работе использовались методы системного анализа, структурного, объектного и информационного моделирования, математического моделирования, математической статистики, идентификации и прогнозирования.

В результате, основным выводом данной ВКР следует считать тот факт, что для повышения эффективности работы предприятия и его подразделений, для начала, нужно разработать эффективную систему для информирования персонала. В результате работы был получен готовый проект приложения, полностью готовый к разработке и реализации.

Заказчику применение данной разработки экономически выгодно, что показано в экономической части проекта.

Благодаря существенному сокращению времени ручной работы с документами, разработанный продукт увеличивает эффективность работы отдела сбыта.

Таким образом, цель данной выпускной квалификационной работы достигнута, все задачи, поставленные в проекте, полностью решены.

Список использованной литературы

1. Альтернативная коммуникация: обзор корпоративных мессенджеров [Электронный ресурс]. // ICT-Online. - Режим доступа: <http://ict-online.ru/news/n119552> (дата обращения 12.12.2019)
2. Борщёва, Н.Л. Информационные системы в экономике: учеб. пособие. / Н.Л. Борщёва. – Издательство Томского политехнического университета. –Томск: 2017. – 115 с.
3. Буч, Г., Рамбо, Дж., Джекобсон, А. Язык UML. Руководство пользователя: Пер. с англ. - М.: ДМК, 2016. – 348 с.
4. Вендров, А. М. CASE-технологии. Современные методы и средства проектирования информационных систем. - М.: Финансы и статистика, 2018. – 230с.
5. Годин, В.В. Управление информационными ресурсами [Текст] / В.В. Годин, И.К. Корнеев. - М.: Инфра-М, 2018. - 203 с.
6. Диаграммы деятельности // [Электронный ресурс]URL: <http://lektsii.com/2-100557.html> (дата обращения: 01.01.2020).
7. Диго, С. М. Базы данных: проектирование и использование: учеб. / С.М.Диго. – М.: Финансы и статистика, 2015. - 592 с.
8. Документация по MySQL [Электронный ресурс]. // SQL.ru. - Режим доступа: <http://www.sql.ru/docs/mysql/> (дата обращения 12.12.2019)
9. Дэвид, Ф. А. JavaScript: учеб. / Ф.А. Дэвид.-М.:АСВ,2017.-169 с.
10. Зандстра, М. PHP. Объекты, шаблоны и методики программирования: учеб. пособие/Зандстра М.-М.: Вильямс, 2015.- 577с.
11. Карпова, Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2016. – 387 с.
12. Кузнецов, С.Д. Проектирование и разработка корпоративных информационных систем. Центр информационных технологий.М.:МГУ,2017.– 244с.
13. Маклаков, С.В. ВРwin и ERwin. CASE-средства разработки

информационных систем. – М. : ДИАЛОГ-МИФИ, 2019. – 304 с.

14. Маклафлин, Б. PHP и MySQL. Исчерпывающее руководство: 2-е издание / Б. Маклафлин - Санкт-Петербург: Питер, 2014. - 544 с.

15. Мелихов, М. С., Тыртышный А. И. Разработка автоматизированной информационной системы приёма и обработки заявок для дизайнерской студии с целью увеличения скорости предоставления услуг // Молодой ученый. — 2016. — №7. — С. 44-46.

16. Методы оценки эффективности ИС // [Электронный ресурс] URL: <http://wiki.mvtom.ru/index.php/MeTO> оценки эффективности ИС (дата обращения 06.05.2019).

17. Моделирование данных. [Электронный ресурс]. - Режим доступа: http://citforum.ru/database/case/glava2_4_1.shtml (дата обращения 13.12.2019) .

18. Назначение, стандарты и преимущества языка SQL [Электронный ресурс] // Лекции.Орг. - Режим доступа: <http://lektsii.org/5-77336.html> (дата обращения 12.12.2019)

19. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript / Р. Никсон - Санкт-Петербург: Питер, 2017. - 496 с.

20. Нотация IDEF0 [Электронный ресурс] // businessstudio.ru - Режим доступа: <http://www.businessstudio.ru/wiki/docs/v4/doku.php/ru/csdesign/bpmodeling/idef> (дата обращения: 03.12.2019)

21. Орлов, С. Технологии разработки программного обеспечения. – СПб.: Питер, 2018. – 464 с.

22. Основные языки программирования и разметки, применяющиеся в web-разработке [Электронный ресурс] // HTML, CSS, PHP, JavaScript, SQL. - Режим доступа: <http://www.codeharmony.ru/materials/125> (дата обращения 12.12.2019)

23. Основы IDEF3 [Электронный ресурс] // citforum.ru - Режим доступа: <http://citforum.ru/cfm/idef/idef3.shtml> (дата обращения: 13.12.2019)

24. Поддубный, А. Расчет экономического эффекта от внедрения системы автоматизации. [Электронный ресурс]. - Режим доступа:

http://antegra.rU/news/experts/_det-experts/4/ (дата обращения 13.12.2019) .

25. Потапова, Е.А., Сторожева Е.В. Оценка эффективности проектов // Информационные технологии в науке, управлении, социальной сфере и медицине. - 2015. - С. 357-359.

26. Похилько, А. Ф. Моделирование процессов и данных с использованием CASE-технологий : учеб. пособие / А. Ф. Похилько, И. В. Горбачев, С. В. Рябов. – Ульяновск : УлГТУ, 2014. – 163 с.

27. Руководство Yii2.0 [Электронный ресурс]. // Электронная документация. - Режим доступа: [Электронный ресурс] <https://yiiframework.com.ua/ru/doc/guide-index.html/> (дата обращения 12.12.2019)

28. Свириденко, С.С. Современные информационные технологии. [Текст] / С.С. Свириденко - М.: Инфра-М, 2018. - 160 с.

29. Середенко, Е.С. Оценка экономической эффективности аналитических информационных систем: дисс. канд. экон. наук / Середенко Е.С.- М.: МГУ, 2018,- 166 с.

30. Современный учебник JavaScript [Электронный ресурс]. // Электронная документация. - Режим доступа: <https://learn.javascript.ru> (дата обращения 12.12.2019)

31. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности - Введ. 09.01.2014. - Красноярск: ИПК СФУ, 2014. - 60 с.

32. Сторожева, Е.В., Старков, А.Н. Методы оценки эффективности ит-проектов // Магнитогорский государственный технический университет им. Г.И. Носова (Магнитогорск). - 2016. - С. 141.

33. Хоган, Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения: 2-е издание / Б. Хоган. - Санкт-Петербург: Питер, 2014. - 320 с.

34. Якобсон, А., Буч, Г., Рамбо, Дж. Унифицированный процесс разработки программного обеспечения. Пер. с англ. - СПб: Питер, 2017. – 402с.

35. Янк,К PHP и MySQL. От новичка к профессионалу: учебное пособие / К. Янк. - Москва: Эксмо, 2013. -384с.