

Курск, 2024 г.

СОДЕРЖАНИЕ

1	Анализ предметной области	3
1.1	Понятие и принципы приложения	3
1.1.1	Основная задача	5
1.1.2	Моделирование	5
1.1.3	Социальная коммерция	7
1.1.4	UML диаграммы	7
1.2	Классы UML диаграмм	8
1.3	Диаграмма вариантов использования	9
1.4	Диаграмма классов	10
1.5	Цель инфологического моделирования	12
1.6	UML – диаграммы для визуализации	13
1.7	Проектирование реляционного хранилища данных	14
2	Технический проект	16
2.2	Обоснование выбора технологий проектирования	17
2.2.1	Язык программирования C#	18
2.2.2	Представления появления окон приложения	18
2.3	Создание хранимой процедуры	21
2.4	Создание триггеров	22
3	Архитектура desktop-приложения	23
3.1	Формы приложения	24
3.2	Тестирование программного приложения	33
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	36

1 Анализ предметной области

1.1 Понятие и принципы приложения

В настоящее время большое внимание уделяется расчётам с поставщиками и подрядчиками. Это обусловлено тем, что постоянно совершающийся кругооборот хозяйственных средств вызывает непрерывное возобновление многообразных расчётов. Одним из наиболее распространённых видов расчётов как раз и являются расчёты с поставщиками и подрядчиками за сырьё, материалы, товары и прочие материальные ценности. Актуальность темы заключается в рассмотрении основных аспектов формирования системы взаимоотношений с поставщиками и подрядчиками для разработки рекомендаций по формированию системы взаимодействия с поставщиками и подрядчиками.

Так же актуальность выбранной темы дипломной работы связана с тем, что в современном обществе очень большую роль играют социальные сети и разнообразные мессенджеры как средства виртуального общения. Растет число людей, которые могут часами проводить в виртуальном общении, а многим оно заменяет реальную коммуникацию.

Целью дипломной работы является разработка проекта выбора и внедрения информационной системы взаимодействия с клиентами и партнерами организации. Согласно представленного задания, с учетом формируемых общих и профессиональных компетенций, необходимо:

- спроектировать программный продукт на основе анализа предметной области и исходных данных;
- разработать UML-диаграммы и ER-диаграмму базы данных.

Система должна состоять из нескольких модулей, доступ к которым будет определяться типом учетной записи.

Общее назначение разрабатываемых модулей:

- просмотр списка объектов;

- добавление/удаление/редактирование данных об объектах;
- управление списками возможных объектов.

Часть бизнес-логики приложения реализуется с помощью библиотек, тестирование которых производится с помощью unit-тестов.

Разработанная информационная система так же должна быть протестирована методом черного ящика, по итогам тестирования заполняется тестовая документация.

По созданной системе разрабатывается документация: руководство пользователя, user story и др.

Система должна состоять из нескольких модулей, доступ к которым будет определяться типом учетной записи.

Общее назначение разрабатываемых модулей:

- просмотр списка объектов;
- добавление/удаление/редактирование данных об объектах;
- управление списками возможных объектов.

Часть бизнес-логики приложения реализуется с помощью библиотек, тестирование которых производится с помощью unit-тестов.

Разработанная информационная система так же должна быть протестирована методом черного ящика, по итогам тестирования заполняется тестовая документация.

По созданной системе разрабатывается документация: руководство пользователя, user story и др.

К задачам разрабатываемой ИС относятся:

- автоматизация процесса мониторинга всей информации;
- хранение всей информации в базе данных;
- разграничение прав доступа пользователей к ИС.

Объект исследования – корпоративная почта.

Предмет – информационная система взаимодействия с клиентами.

1.1.1 Основная задача

В рамках выполнения практической работы 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).

Под мессенджером в современном интернет-пространстве понимают «систему мгновенного обмена сообщениями», т. е. программы онлайн-консультанты и программы-клиенты для обмена сообщениями в реальном времени через Интернет.

Основная задача данной дипломной работы — это создание сервиса для обмена сообщениями основные возможности которого являются:

- регистрация, хранение и выдача информации о пользователях в системе;
- ведение переписки с другими зарегистрированными пользователями с использованием шифрования;
- динамическое обновление данных для обеспечения быстрого получения и быстрой доставки сообщений.

1.1.2 Моделирование

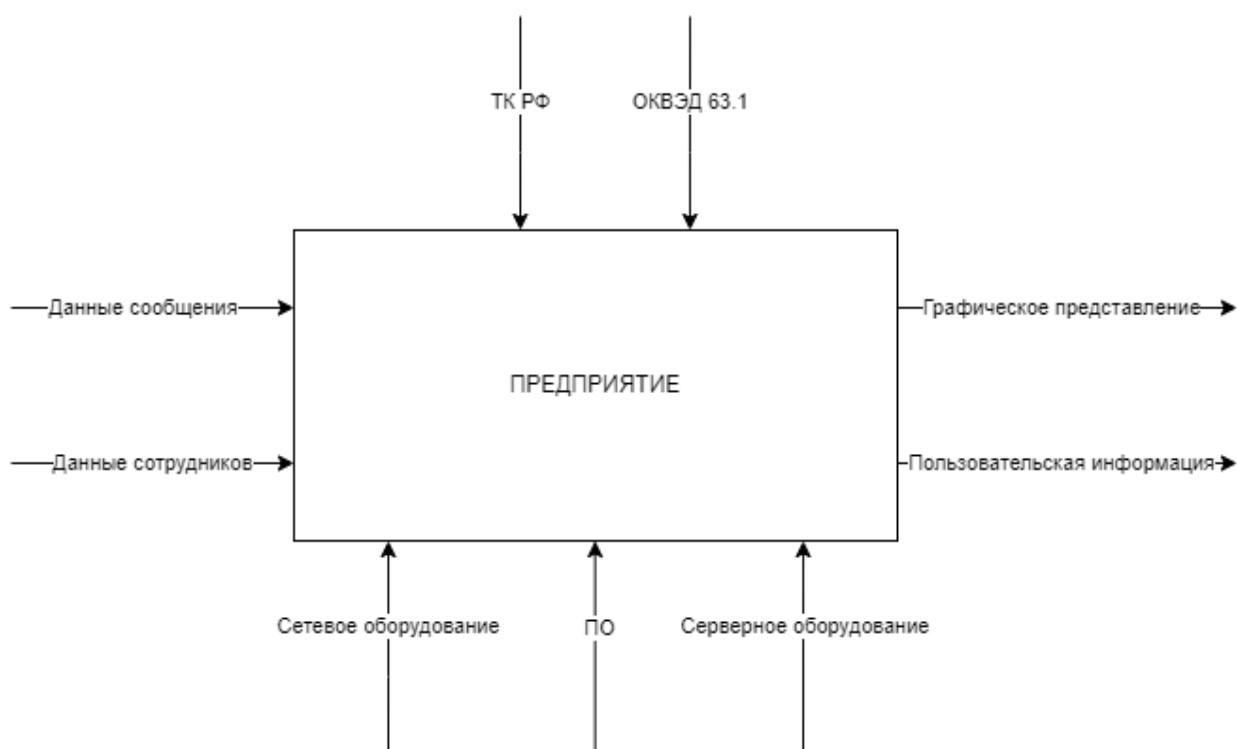
Моделирование — исследование объектов познания на их моделях, построение и изучение моделей реально существующих объектов, процессов или явлений с целью получения объяснений этих явлений, а также для предсказания явлений, интересующих исследователей.

Моделирование процессов описывает логическую взаимосвязь всех элементов процесса от его начала до завершения в рамках организации.

1.1.3 Социальная коммерция

IDEF0 — методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ).

Диаграмма процессов входных и выходных данных IDEF0 представлена на рисунке 1.



1.1.4 UML диаграммы

Для дипломной работы были построены UML-диаграммы для визуализации, конструирования и документирования программных систем.

Построение UML диаграмм можно выполнять вручную на листе бумаги или на доске, а также с помощью специализированного программного обеспечения. Среди достаточно большого количества средств построения диаграмм на UML можно выделить два класса.

1.2 Классы UML диаграмм

Первый класс – простые и дешевые (иногда бесплатные) программы, позволяющие автоматизировать построения диаграммы без генерации программного кода. К таким программам относятся графический редактор MS Visio, программный инструмент моделирования StarUML, бесплатный онлайн сервис Lucidchart и некоторые другие.

Второй класс это, так называемые CASE-средства, представляющие собой набор инструментов, предназначенный для автоматизации визуального моделирования, проектирования, документирования и генерации кода реализации на выбранном алгоритмическом языке. К таким средствам относятся CASE-средства визуального моделирования и проектирования компании IBM Rational Software Corp — Rational Rose и Rational Software Architect, продукт проектирования и интеграции компании Borland – Together и другие.

1.3 Диаграмма вариантов использования

Диаграмма вариантов использования – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

Диаграмма вариантов использования представлена на рисунке 2.



Рисунок 2 – Диаграмма вариантов использования

Для удобства проектирования системы была разработана диаграмма классов. Диаграмма классов на унифицированном языке моделирования (UML) — это диаграмма статической структуры, которая описывает структуру системы, показывая ее классы, их атрибуты, операции (или методы) и отношения между объектами.

1.4 Диаграмма классов

Диаграмма классов — это план системы или подсистемы. Диаграммы классов можно использовать для моделирования объектов, составляющих систему, для демонстрации отношений между объектами и для описания ролей этих объектов и предоставляемых ими услуг.

Данная диаграмма описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами.

Диаграмма классов представлена на рисунке 3.

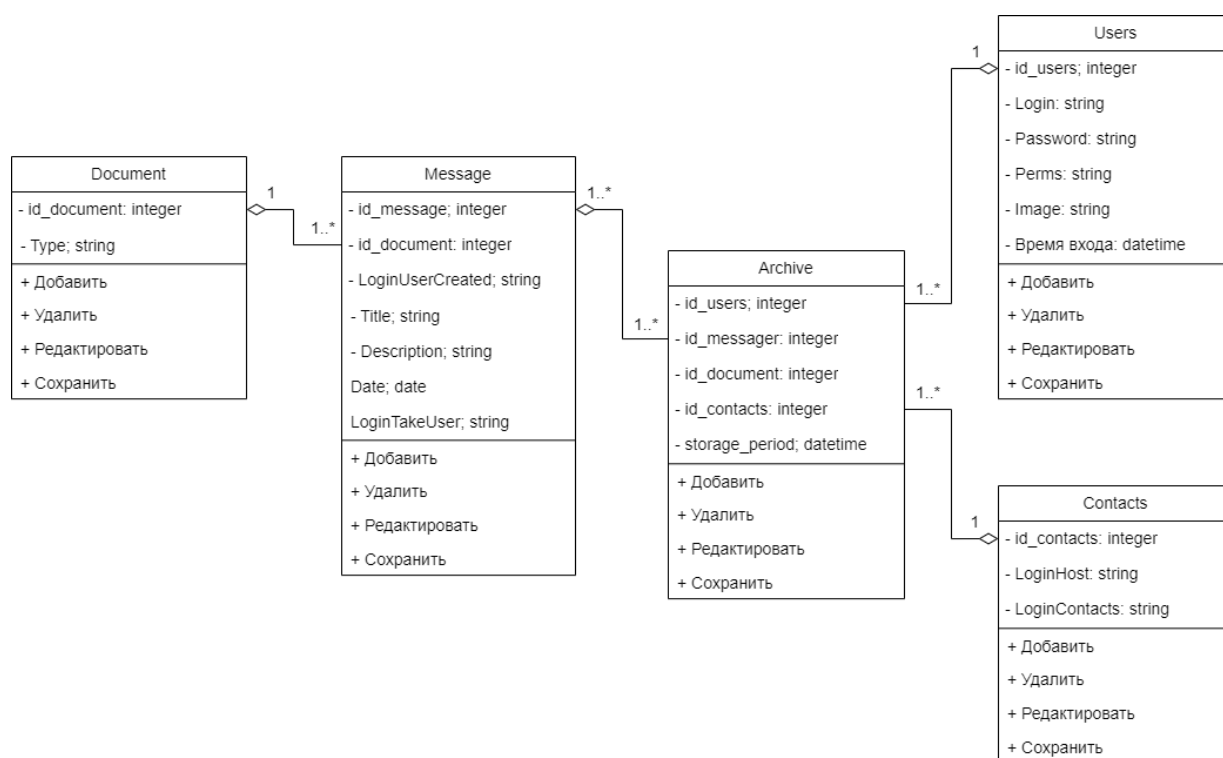


Рисунок 3 – Диаграмма классов

Диаграмма последовательности — диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления.

Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники), вертикальные линии, отображающие течение времени при деятельности объекта, и стрелки, показывающие выполнение действий объектами.

Диаграмма последовательности представлена на рисунке 4.

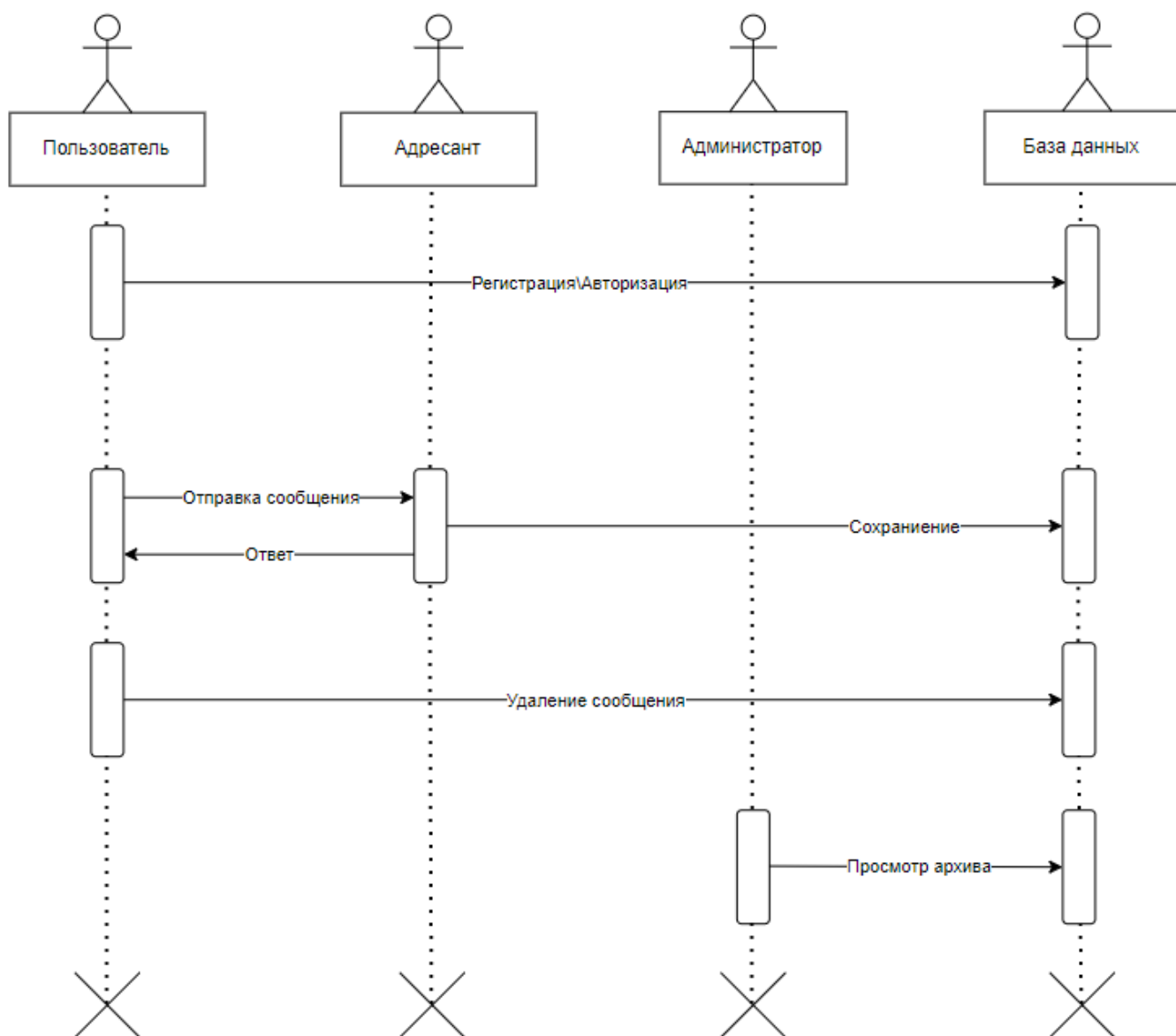


Рисунок 4 – Диаграмма последовательности

Диаграмма деятельности – это своеобразная блок-схема, которая описывает последовательность выполнения операций во времени. Их можно использовать для моделирования динамических аспектов поведения системы. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние срабатывает только при завершении этой операции в предыдущем состоянии.

Диаграмма деятельности представлена на рисунке 5.

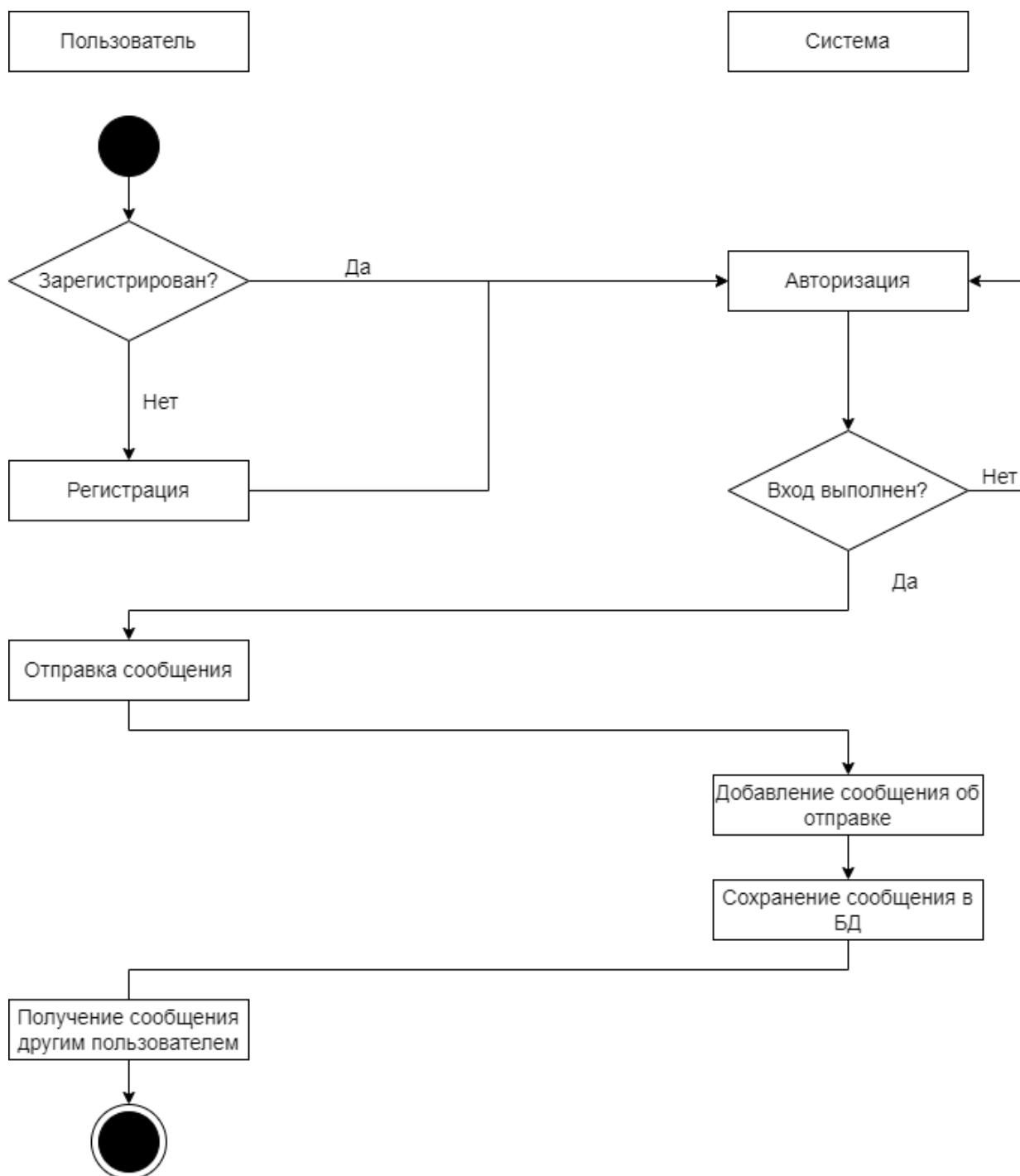


Рисунок 5 – Диаграмма деятельности

1.5 Цель инфологического моделирования

Цель инфологического моделирования — обеспечить оптимальные способы сбора и представления информации, хранимой в базе данных. Целью данной модели является обеспечение наиболее естественных для пользователя способов представления и сбора информации. При работе над данным курсовым проектом были созданы таблицы.

Моделирование бизнес-процессов (BPM) в управлении бизнес-процессами и системной инженерии — это деятельность по представлению процессов предприятия, позволяющая анализировать, улучшать и автоматизировать текущие бизнес-процессы [4].

BPM обычно выполняется бизнес-аналитиками, которые предоставляют экспертные знания в области моделирования; экспертами по предмету, которые обладают специальными знаниями о моделируемых процессах; или, чаще всего, командой, состоящей из обоих.

1.6 UML – диаграммы для визуализации

Для дипломной работы были построены UML-диаграммы для визуализации, конструирования и документирования программных систем.

Построение UML диаграмм можно выполнять вручную на листе бумаги или на доске, а также с помощью специализированного программного обеспечения. Среди достаточно большого количества средств построения диаграмм на UML можно выделить два класса.

Первый класс – простые и дешевые (иногда бесплатные) программы, позволяющие автоматизировать построения диаграммы без генерации программного кода. К таким программам относятся графический редактор MS Visio, программный инструмент моделирования StarUML, бесплатный онлайн сервис Lucidchart и некоторые другие.

Второй класс это, так называемые CASE-средства, представляющие собой набор инструментов, предназначенный для автоматизации визуального моделирования, проектирования, документирования и генерации кода реализации на выбранном алгоритмическом языке. К таким средствам относятся CASE-средства визуального моделирования и проектирования компании IBM Rational Software Corp – Rational Rose и Rational Software Architect, продукт проектирования и интеграции компании Borland – Together и другие.

Диаграмма компонентов позволяет определить архитектуру

разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Диаграмма компонентов, в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы.

Диаграмма компонентов представлена на рисунке 6.

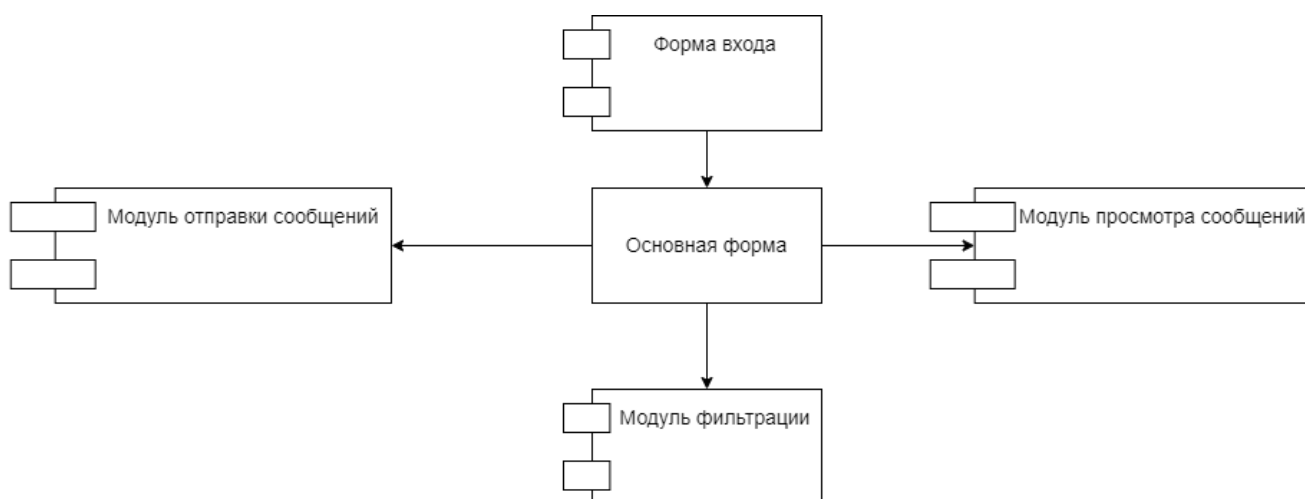


Рисунок 6 – Диаграмма деятельности

1.7 Проектирование реляционного хранилища данных

Диаграммы сущность–связь (ERD) предназначены для разработки моделей данных и обеспечивают стандартный способ определения данных и отношений между ними.

С помощью ERD осуществляется детализация хранилищ данных, а также документируются сущности системы и способы их взаимодействия, включая идентификацию объектов предметной области (сущностей), свойств этих объектов (атрибутов) и их связей (отношений).

Сущность — это любой объект, отличающийся от другого, информацию о котором необходимо сохранить.

Связь — это ассоциирование нескольких сущностей с целью отыскания одних из них по значениям других.

Атрибут — это характеристика сущности. Это может быть числовой характеристикой, классификацией, идентификацией. Его наименование

должно быть уникальным для конкретного типа сущностей и может совпадать с атрибутами других сущностей[9].

Ключ представляет собой минимальное количество атрибутов, с помощью которого можно отыскать необходимый экземпляр сущности.

Связи между сущностями:

- один-к-одному (1:1);
- один-ко-многим (1: M);
- многие-ко-многим (M: M).

На рисунке 7 изображена ER-диаграмма БД «Memail» и содержит в себе следующие сущности: Contact, Message, User, Archive, Document. Связь между сущностями – «один-ко-многим».

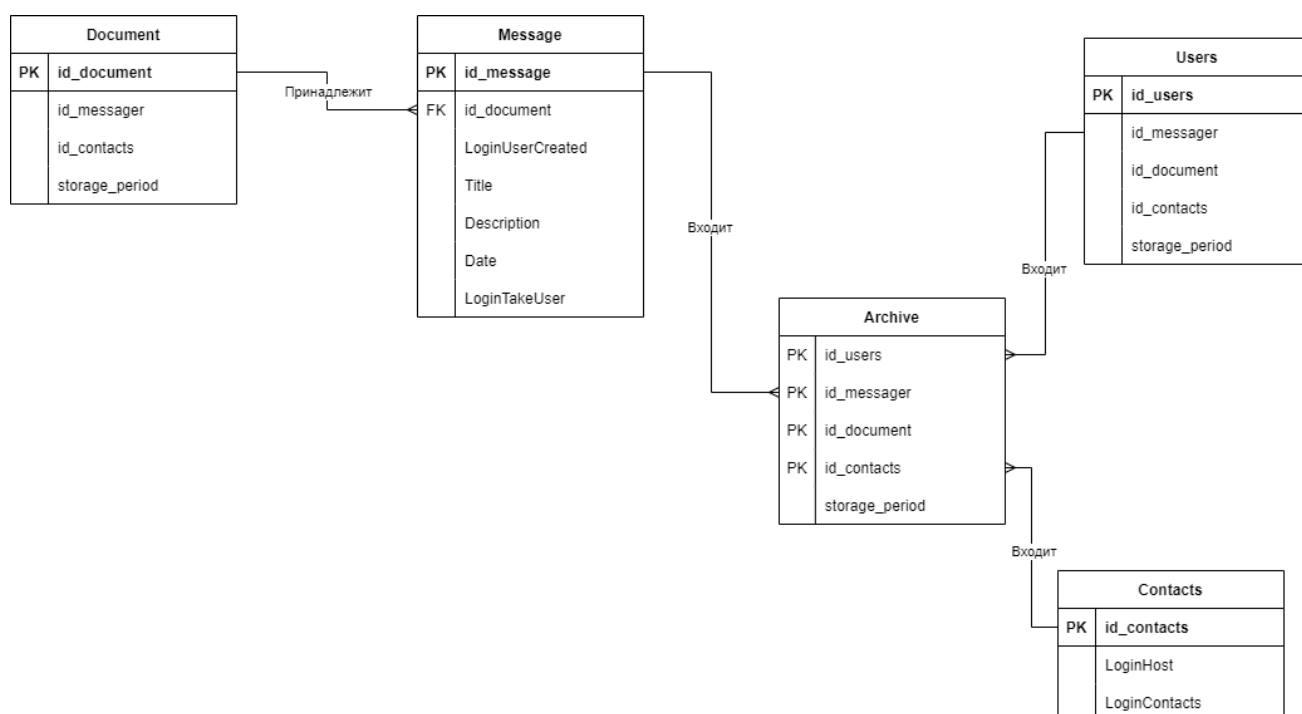


Рисунок 7 – ER-диаграмма.

2 Технический проект

2.1 Разработка объектов базы данных.

Для построения реляционной базы данных была выбрана СУБД MS SQL Server 2019Lbfu.

Реляционная база данных – это набор данных с предопределенными связями между ними. Эти данные организованы в виде набора таблиц, состоящих из столбцов и строк. В таблицах хранится информация об объектах, представленных в базе данных. В каждом столбце таблицы хранится определенный тип данных, в каждой ячейке – значение атрибута. Каждая строка таблицы представляет собой набор связанных значений, относящихся к одному объекту или сущности.

Исходя из описания атрибутов сущностей, были спроектированы таблицы БД, представленные на рисунках с 8 по 12.

 id_contact	int	<input type="checkbox"/>
LoginHost	varchar(50)	<input checked="" type="checkbox"/>
LoginContacts	varchar(50)	<input checked="" type="checkbox"/>

Рисунок 8 – Таблица «Contacts»


 id_message	int	<input type="checkbox"/>
id_document	int	<input checked="" type="checkbox"/>
LoginUserCreated	varchar(50)	<input type="checkbox"/>
Title	varchar(MAX)	<input checked="" type="checkbox"/>
Description	varchar(MAX)	<input checked="" type="checkbox"/>
Date	datetime	<input type="checkbox"/>
LoginTakeUser	varchar(50)	<input type="checkbox"/>

Рисунок 9 – Таблица «Message»


 id_users	int	<input type="checkbox"/>
Login	varchar(50)	<input type="checkbox"/>
Password	varchar(50)	<input checked="" type="checkbox"/>
Perms	varchar(50)	<input checked="" type="checkbox"/>
Image	image	<input checked="" type="checkbox"/>
[Время входа]	datetime	<input checked="" type="checkbox"/>

Рисунок 10 – Таблица «Users»

id_users	int	<input type="checkbox"/>
id_message	int	<input type="checkbox"/>
id_document	int	<input type="checkbox"/>
id_contacts	int	<input type="checkbox"/>
storage_period	datetime	<input type="checkbox"/>

Рисунок 11 – Таблица «Archive»

id_document	int	<input type="checkbox"/>
id_message	int	<input checked="" type="checkbox"/>
id_contacts	int	<input checked="" type="checkbox"/>
storage_period	datetime	<input checked="" type="checkbox"/>

Рисунок 12 – Таблица «Document»

Схема данных представляет набор схем всех таблиц созданной базы данных и представлена на рисунке 13.

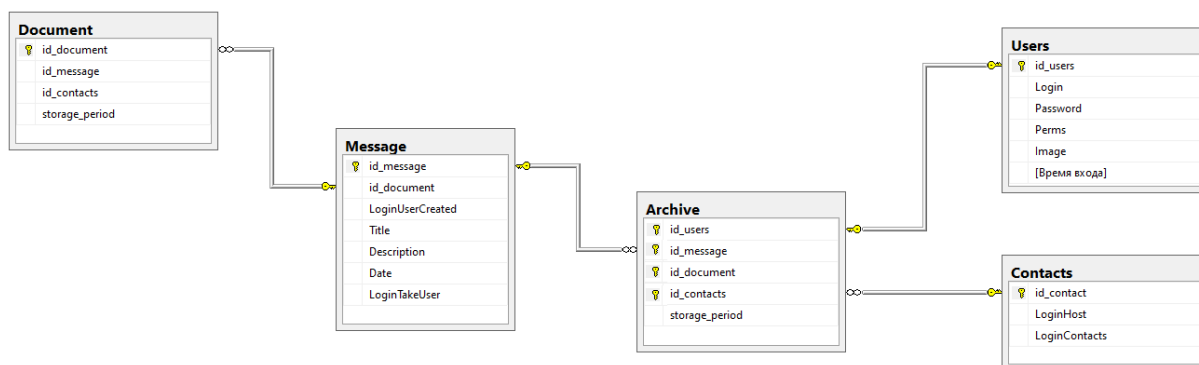


Рисунок 13 – Схема данных

2.2 Обоснование выбора технологий проектирования

Разрабатываемая база данных и приложение содержат в себе различные представления, хранимые процедуры, триггеры и скалярные функции.

Хранимая процедура — это объект базы данных, представляющий собой набор SQL – инструкций, который компилируется один раз и хранится на сервере.

Хранимая функция отличается от хранимой процедуры, тем что хранимая функция всегда возвращает только скаляр, а процедура возвращает набор значений.

Триггер — это хранимая процедура, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено действием по

модификации данных: добавлением INSERT, удалением DELETE строки в заданной таблице, или изменением UPDATE данных в определенном столбце заданной таблицы реляционной базы данных.

2.2.1 Язык программирования C#

C# [15] является статистически типизированным языком программирования (типы переменных не подлежат изменению, должны быть явно объявлены, либо получены из контекста), компилятор которого способен генерировать байт-код, полностью совместимый с инструкциями Java Virtual Machine. Официальная документация предоставляет разработчику право выбора между объектно-ориентированным и функциональным стилем программирования. В мае 2017 года на конференции разработчиков Google I/O было объявлено о включении языка в список официальных наравне с Java и добавлении его поддержки в Android Studio – основную среду разработки под Android. На конференции Google I/O 2019 C# получил статус приоритетного языка, что ознаменовало последовательный переход библиотек и компонентов Android SDK от Google на новый язык программирования. Также C# стал языком по умолчанию при создании проектов в Android Studio, а разработчики многих популярных приложения (Periscope, Twitter, Pinterest и т.д.) успешно используют C# в актуальных версиях своих программ.

На языке C# написано как клиентское приложение, так и серверная часть, что позволяет переиспользовать часть кодовой базы между компонентами системы.

2.2.2 Представления появления окон приложения

Представления - особый объект, который содержит данные, полученные запросом SELECT из обычных таблиц. Это виртуальная таблица, к которой можно обратиться как к обычным таблицам и получить хранимые данные. Представление в SQL может содержать в себе как данные из одной единственной таблицы, так и из нескольких таблиц.

B5-304-6\SQLXP...25 - dbo.View_1*

Столбец	Псевдоним	Таблица	Выход	Тип сортиро...	Порядок сорт...	Filter	Or...	Or...	Or...
Date		Message	<input checked="" type="checkbox"/>						
id_message		Archive	<input checked="" type="checkbox"/>						
id_contacts		Archive	<input checked="" type="checkbox"/>						
[Время входа]		Users	<input checked="" type="checkbox"/>						
id_message	Expr 1	Message	<input checked="" type="checkbox"/>						

```

SELECT TOP (100) PERCENT dbo.Message.Date, dbo.Archive.id_message, dbo.Archive.id_contacts, dbo.Users.[Время входа], dbo.Users.id_users, dbo.Message.id_message AS Expr 1
FROM
  dbo.Archive INNER JOIN
    dbo.Message ON dbo.Archive.id_message = dbo.Message.id_message INNER JOIN
      dbo.Users ON dbo.Archive.id_users = dbo.Users.id_users
  
```

Рисунок 14 – Представление просмотра даты сообщения пользователя

B5-304-6\SQLXP...25 - dbo.View_1*

Столбец	Псевдоним	Таблица	Выход	Тип сортиро...	Порядок сорт...	Filter	Or...
Title		Message	<input checked="" type="checkbox"/>				
Description		Message	<input checked="" type="checkbox"/>				
id_document		Archive	<input checked="" type="checkbox"/>				
id_message		[Document]	<input checked="" type="checkbox"/>				

```

SELECT TOP (100) PERCENT dbo.Message.Title, dbo.Message.Description, dbo.Archive.id_document, dbo.[Document].id_message
FROM
  dbo.Archive INNER JOIN
    dbo.[Document] ON dbo.Archive.id_document = dbo.[Document].id_document INNER JOIN
      dbo.Message ON dbo.Archive.id_message = dbo.Message.id_message
  
```

Рисунок 15 – Представление поиска сообщения пользователя в архиве

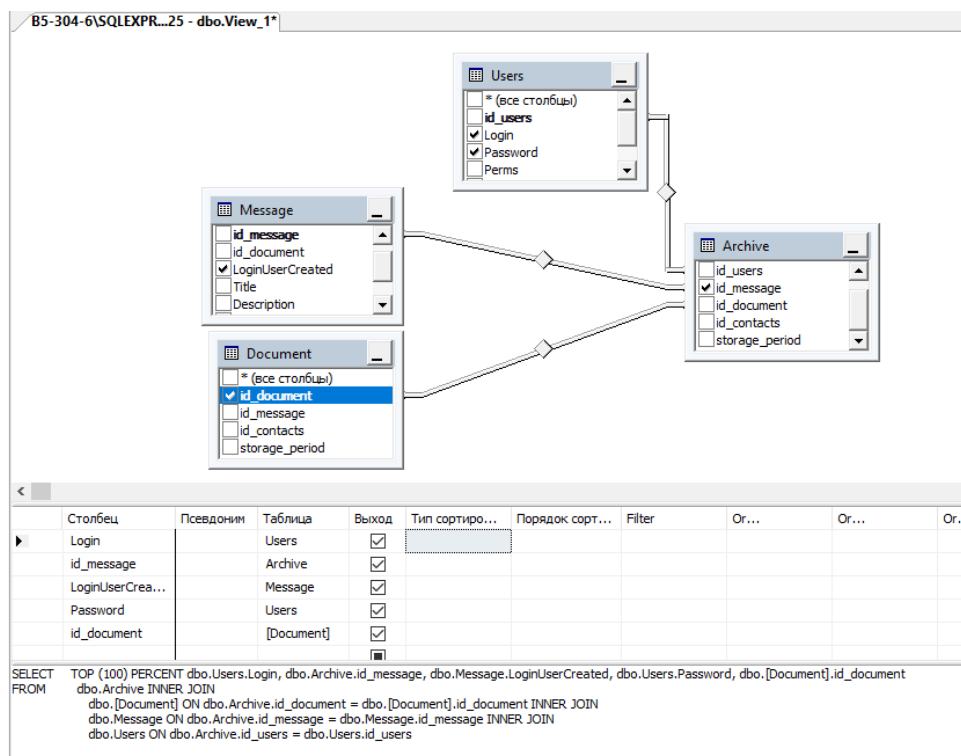


Рисунок 16 – Представление создания логина и пароля пользователем

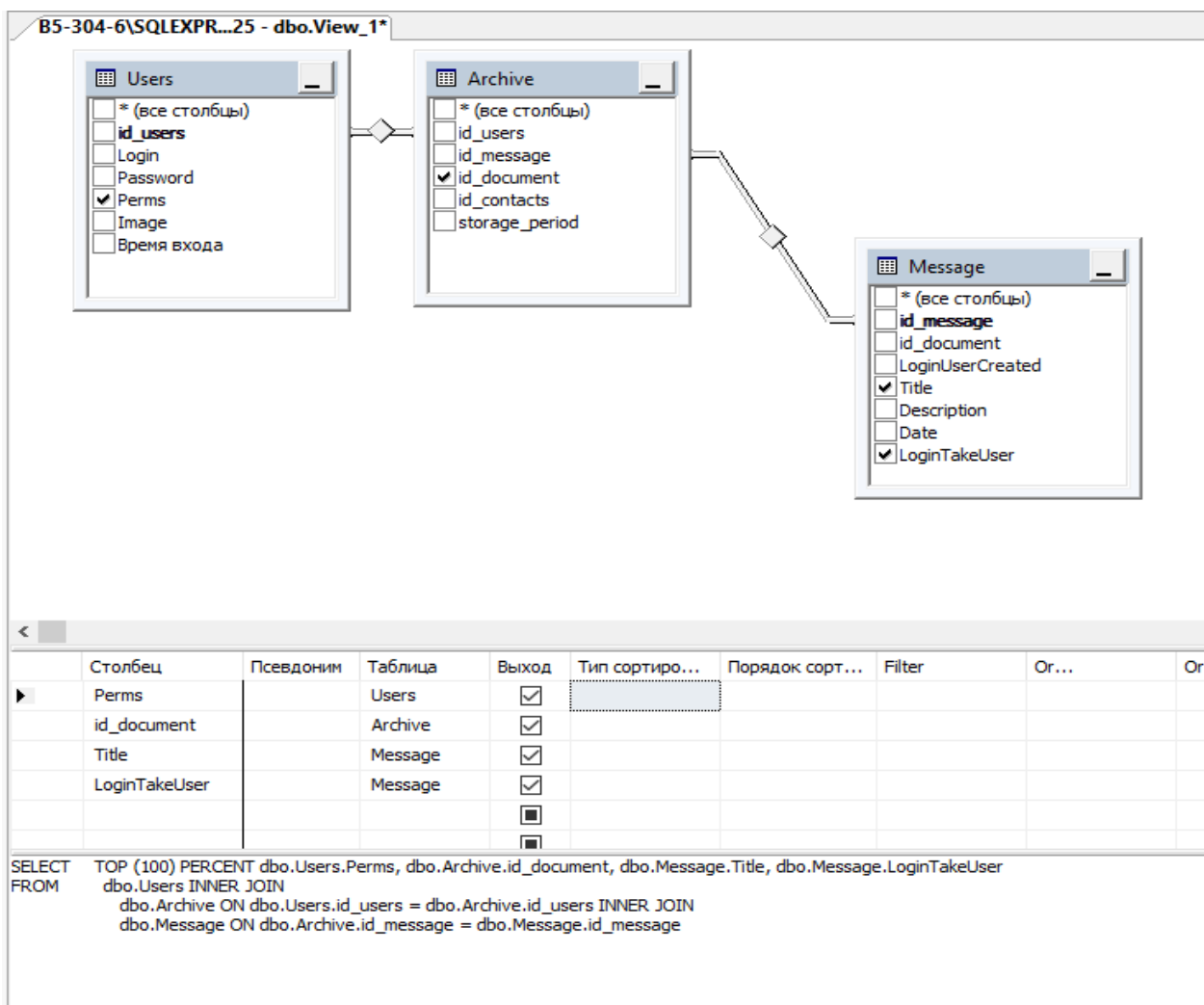


Рисунок 17 – Представление появления сообщения после взаимодействия пользователя с архивом

2.3 Создание хранимой процедуры

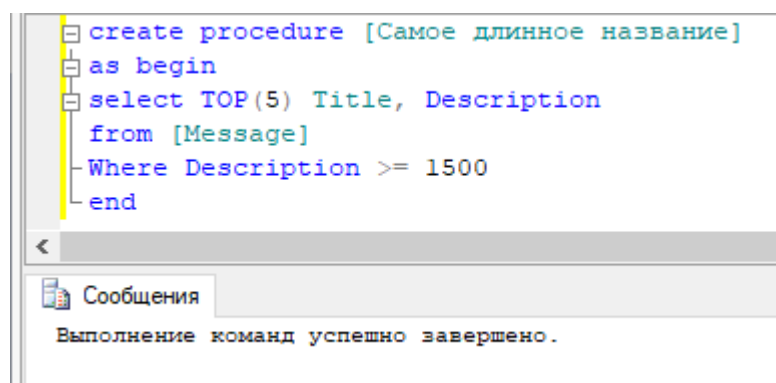


Рисунок 18 – Создание хранимой процедуры «Самое длинное название письма»

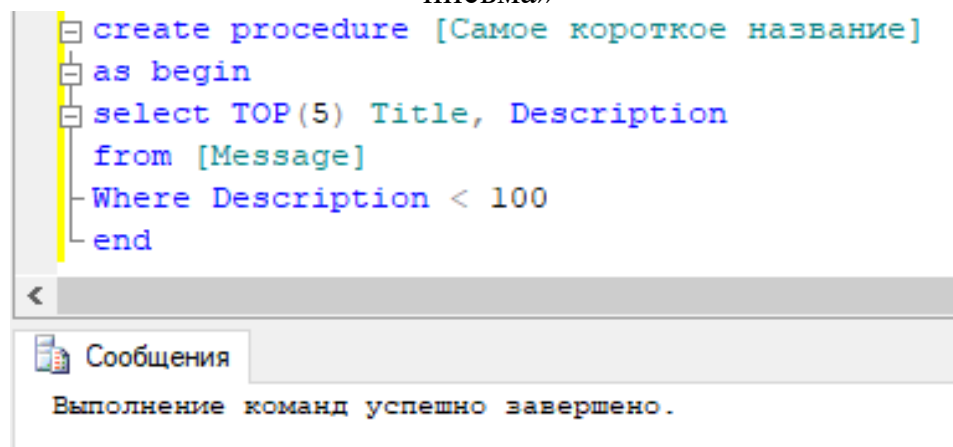


Рисунок 19 – Создание хранимой процедуры «Самое короткое название письма»

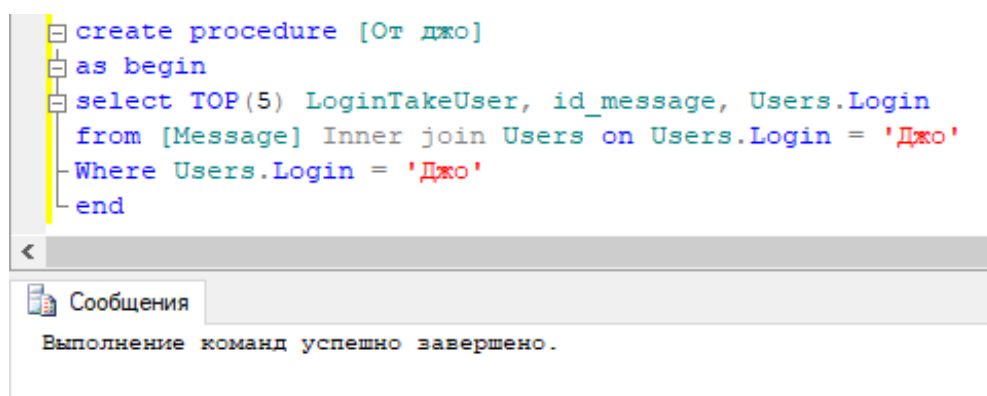


Рисунок 20 – Создание хранимой «Процедура поиска сообщения отправленным конкретным Логин»

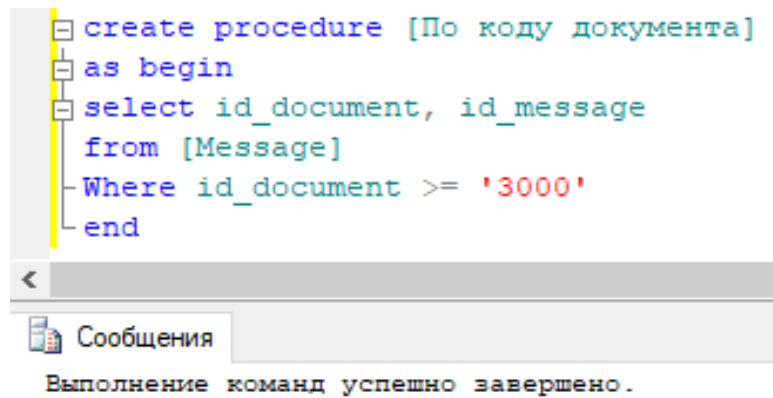


Рисунок 21 – Создание хранимой процедуры «Поиск по фильтрации писем со значением ниже конкретного числа»

2.4 Создание триггеров

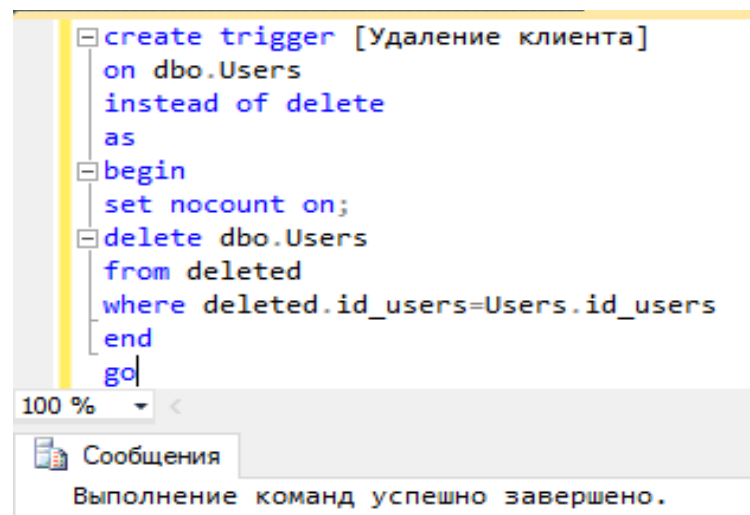


Рисунок 22 – Создание триггера «Удаление клиента»

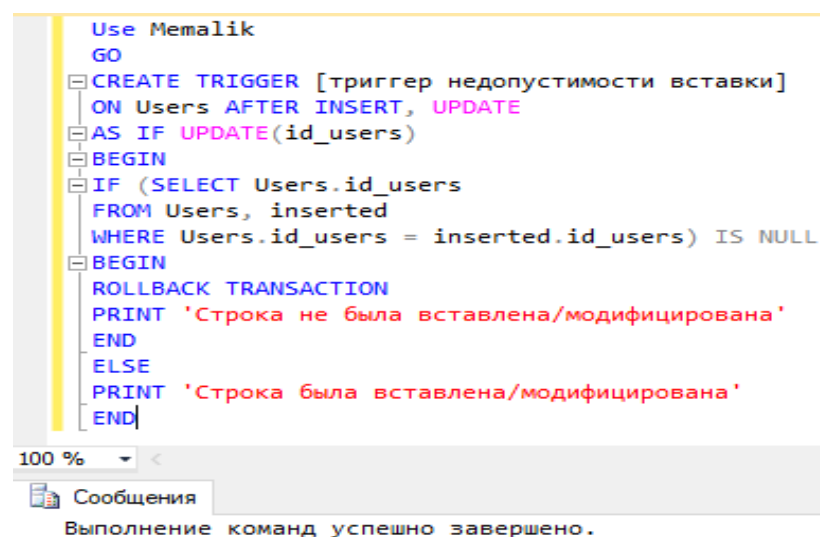


Рисунок 23 – Создание триггера «Триггер недопустимости вставки»

3. Архитектура desktop-приложения

Исходя из требований технического задания, программно-информационная система должна взаимодействовать с двумя базами данных.

На стороне сервера требуется использовать СУБД Couchbase [17]. Couchbase – это система управления базами данных класса NoSQL, которая предоставляет средства для создания документно-ориентированных баз данных, хранения и обработки JSON-документов.

Мобильное приложение должно использовать SQLite – легковесную реляционную базу данных, которая хранит данные в локальном файле и не требует отдельного сервера [18].

SQL и NoSQL представляют собой два разных подхода к хранению и обработке данных. SQL-базы данных являются реляционными базами данных, которые хранят данные в структурированных таблицах и используют язык SQL для запросов и манипуляций с данными. NoSQL-базы данных не требуют определения схемы таблиц, могут хранить данные в различных форматах: документы, графы или пары «ключ-значение». Такой подход делает их более гибкими и масштабируемыми.

Использование официальной библиотеки Room (слой абстракции над SQLite) является рекомендованным способом организации хранения данных в ОС Android. Ядро SQLite встроено в операционную систему, что позволяет уменьшить размер итогового apk-файла.

Выбор Couchbase обусловлен необходимостью поддерживать синхронизацию внутренней структуры данных с параметрами товаров на сайтах интернет-магазинов. Хранение информации в формате JSON-документов позволяет модифицировать структуру данных без проведения миграции БД. Кроме того, являясь NoSQL-базой данных, СУБД Couchbase имеет возможность масштабирования с использованием распределенных кластеров аппаратного обеспечения.

3.1 Формы приложения

Ис На основе выявленных сущностей были спроектированы формы, представленные в режиме конструктора на рисунках с 24 по 31.

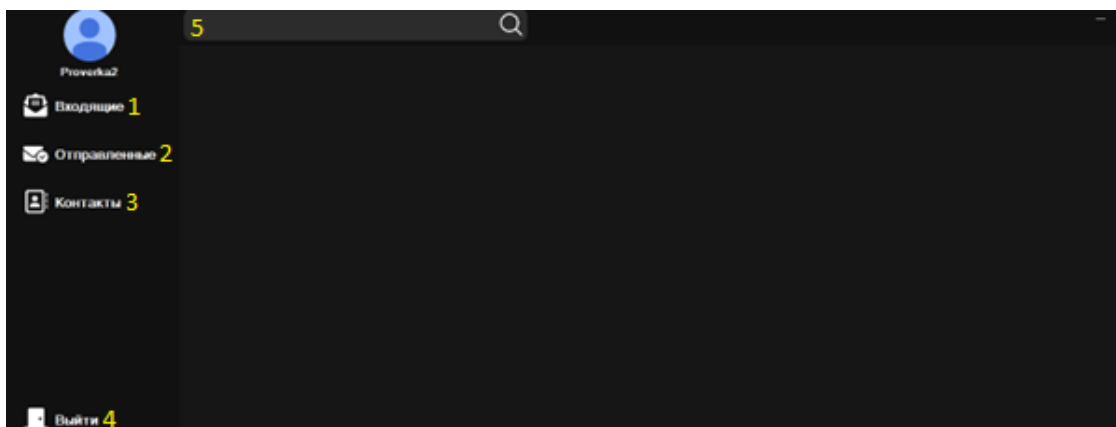


Рисунок 24 – Форма главная

- 1 – iconButton3;
- 2 – iconButton2;
- 3 – iconButton4;
- 4 – iconButton1;
- 5 – customTextBox1.

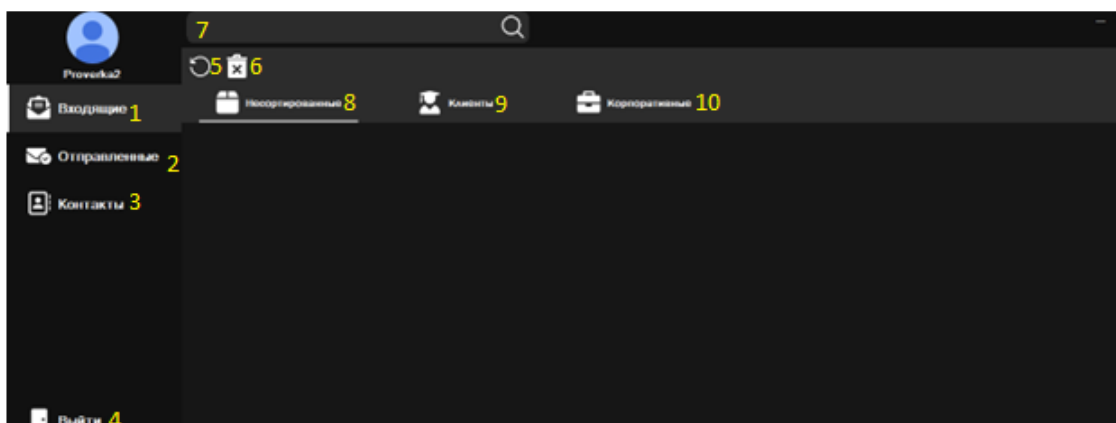


Рисунок 25 – Форма входящие

- 1 – iconButton3;
- 2 – iconButton2;
- 3 – iconButton4;
- 4 – iconButton1;
- 5 – pictureBox3;
- 6 – pictureBox1;

- 7 – customTextBox1;
- 8 – iconButton5;
- 9 – iconButton6;
- 10 – iconButton7.

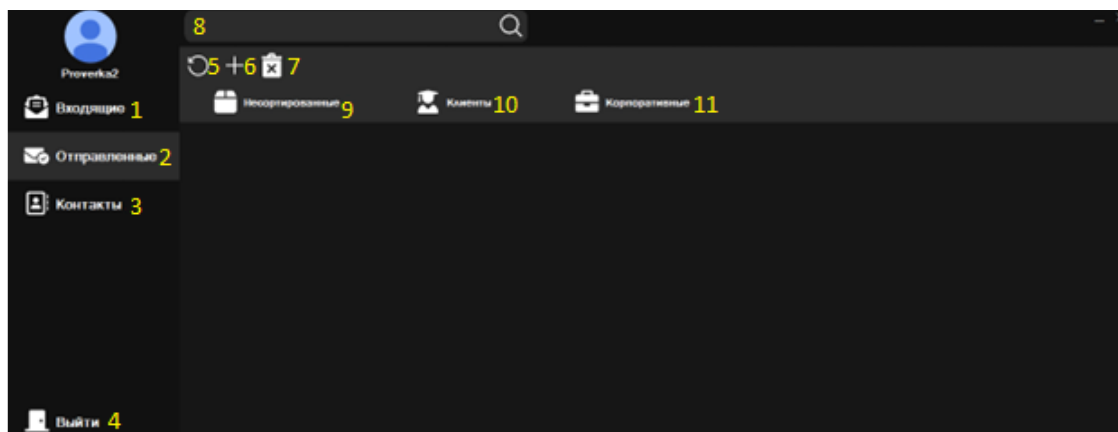


Рисунок 26 – Форма отправленные

- 1 – iconButton3;
- 2 – iconButton2;
- 3 – iconButton4;
- 4 – iconButton1;
- 5 – pictureBox3;
- 6 – pictureBox1;
- 7 – pictureBox2;
- 8 – customTextBox1;
- 9 – iconButton5;
- 10 – iconButton6;
- 11 – iconButton7.

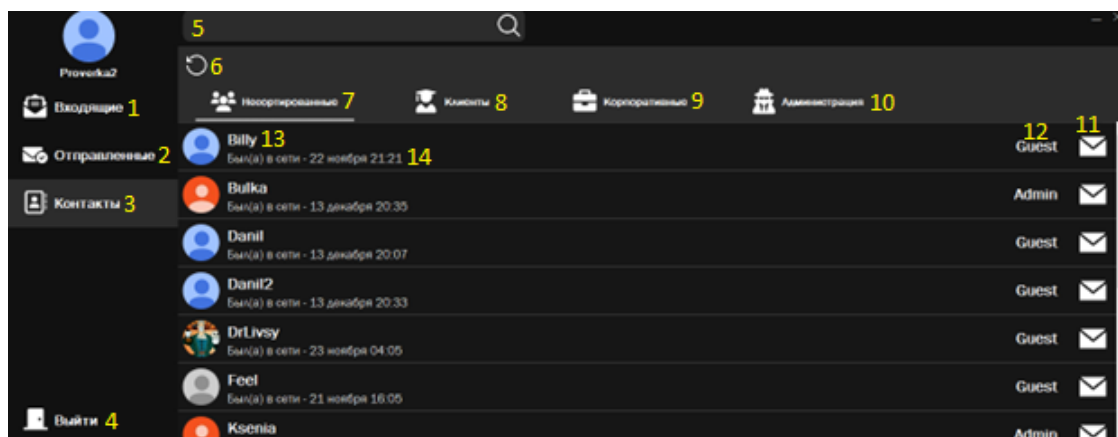


Рисунок 27 – Форма контакты

1 – iconButton3;

2 – iconButton2;

3 – iconButton4;

4 – iconButton1;

5 – pictureBox1;

6 – pictureBox2;

7 – iconButton5;

8 – iconButton6;

9 – iconButton7;

10 – iconButton8;

11 – pictureBox3;

12 – label2;

13 – label1;

14 – label3.

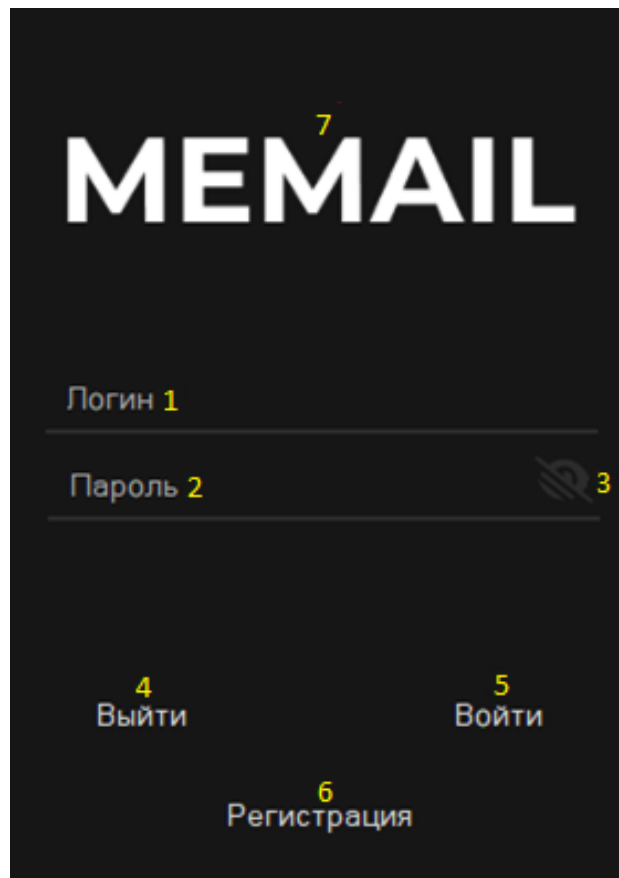


Рисунок 28 – Форма авторизация

- 1 – customTextBox1;
- 2 – customTextBox2;
- 3 – iconPictureBox3;
- 4 – iconButton1;
- 5 – iconButton2;
- 6 – iconButton3;
- 7 – pictureBox1.

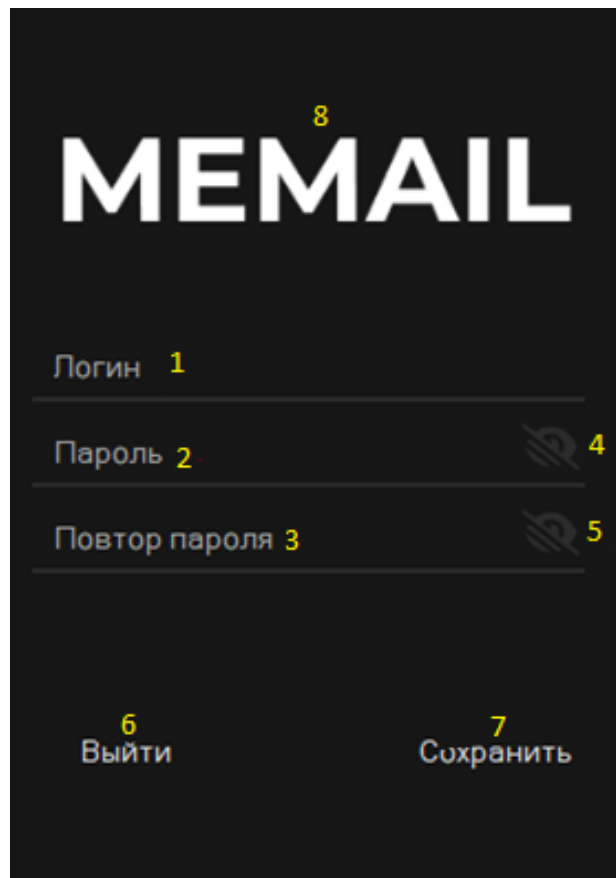


Рисунок 29 – Форма регистрация

- 1 – customTextBox1;
- 2 – customTextBox2;
- 3 – customTextBox3;
- 4 – iconPictureBox3;
- 5 – iconPictureBox1;
- 6 – iconButton3;
- 7 – iconButton4;
- 8 – pictureBox1.

Рисунок 30 – Форма добавления сообщения

1 – customComboBox1;

2 – customTextBox1;

3 – customTextBox2;

4 – iconButton1;

5 – iconButton2.

Рисунок 31 – Форма входящего сообщения я

1 – label3;

2 – iconButton1;

3 – label2;

4 – label1.

Руководство пользователя

Для того, чтобы произвести регистрацию пользователя в системе необходимо ввести логин, пароль, повторить пароль, затем нажать кнопку сохранить.

На рисунке 31 форма входа в систему.

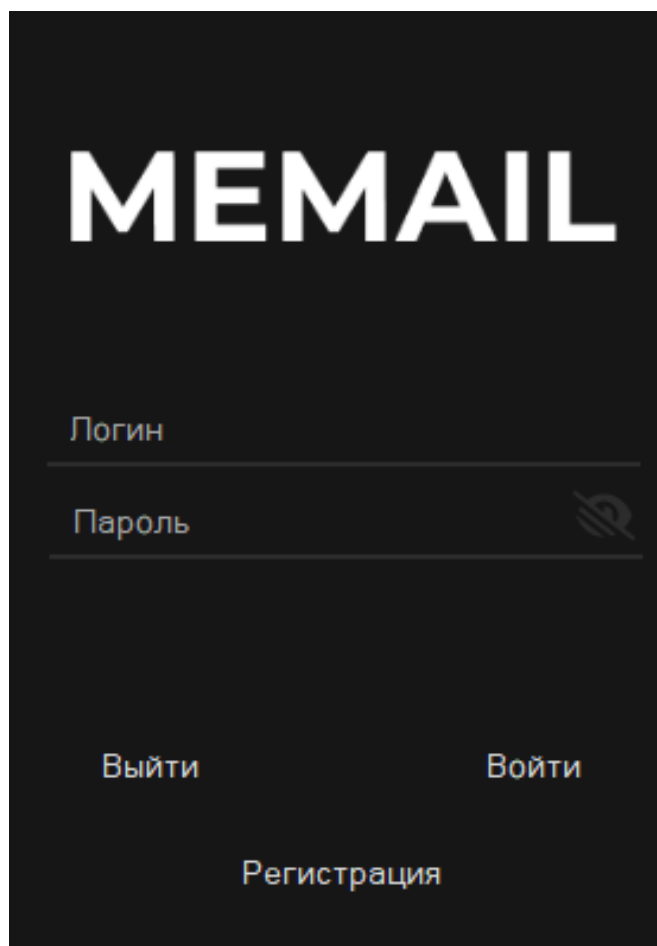
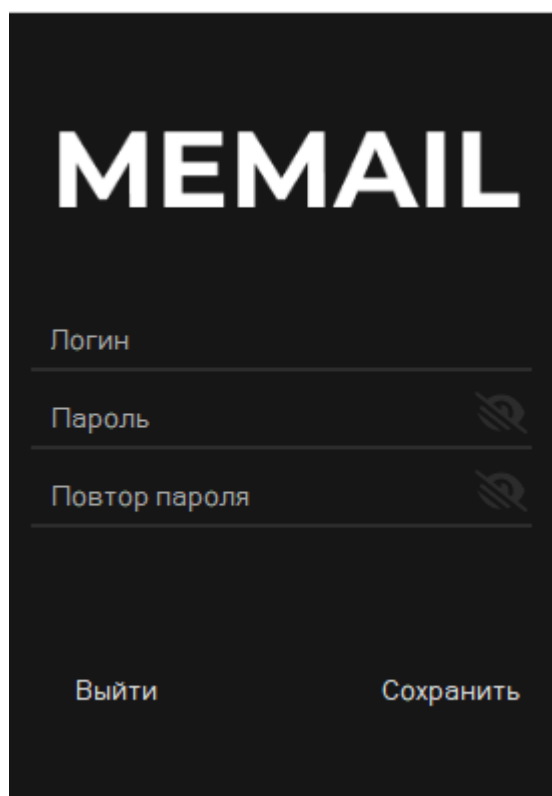
The image shows a dark-themed login interface for 'MEMAIL'. At the top, the word 'MEMAIL' is displayed in large, bold, white capital letters. Below the title, there are two input fields: the first is labeled 'Логин' (Login) and the second is labeled 'Пароль' (Password). The password field includes a small icon of an eye with a diagonal line through it, indicating a toggle for password visibility. At the bottom of the form, there are three buttons: 'Выйти' (Logout) on the left, 'Войти' (Login) on the right, and 'Регистрация' (Registration) centered below them.

Рисунок 32 – Форма «Авторизация»

На рисунке 32 представлена форма авторизации. Регистрация выполняется аналогично авторизации.



The image shows a registration form for 'MEMAIL' on a dark background. At the top, the word 'MEMAIL' is written in large, white, bold, sans-serif capital letters. Below it, there are three input fields with light gray borders. The first field is labeled 'Логин' (Login) in a light gray font. The second field is labeled 'Пароль' (Password) in a light gray font, and it has a small icon of a hand with a lightning bolt to its right. The third field is labeled 'Повтор пароля' (Repeat password) in a light gray font, and it also has the same hand icon to its right. At the bottom of the form, there are two buttons: 'Выйти' (Logout) on the left and 'Сохранить' (Save) on the right, both in a light gray font.

Рисунок 33 – Форма «Регистрация»

На рисунке 34 представлено главное меню пользователя.

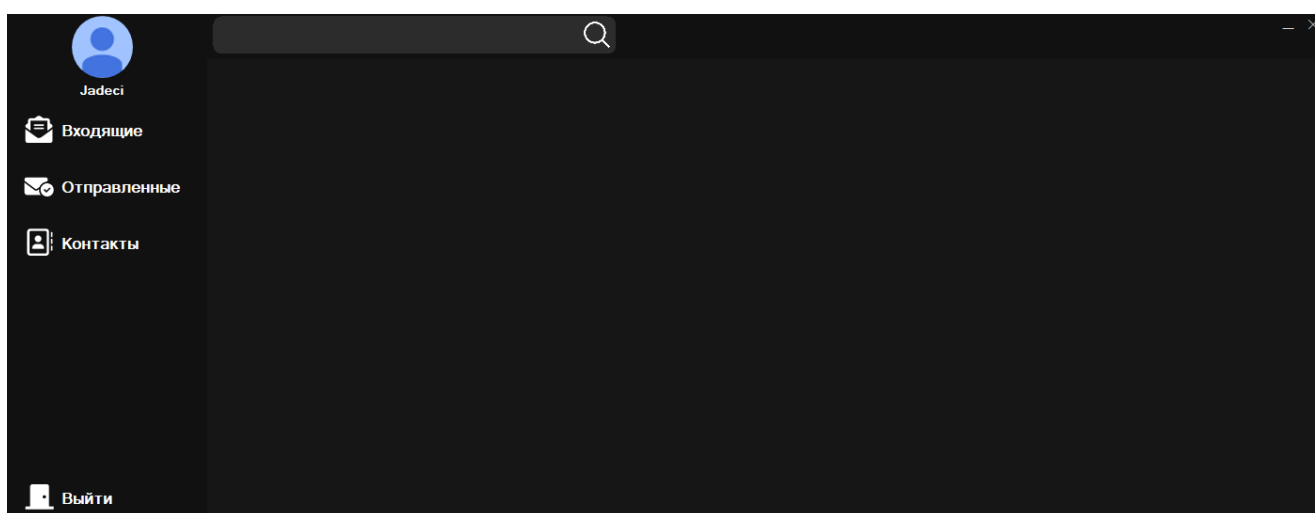


Рисунок 34 – Форма «Главная»

На рисунке 35 представлен раздел входящих сообщений, имеющий сортировку и удаление письма.

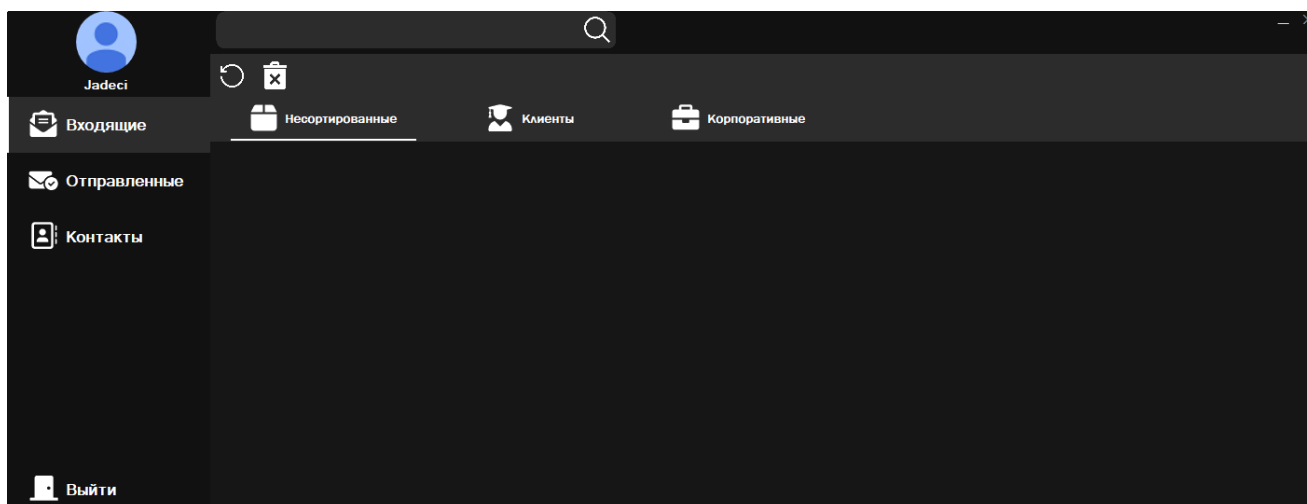


Рисунок 35 – Форма «Входящие»

На рисунке 36 показана форма отправленных сообщений имущая схожий функционал. Сортировка выполняется аналогично входящим сообщениям.

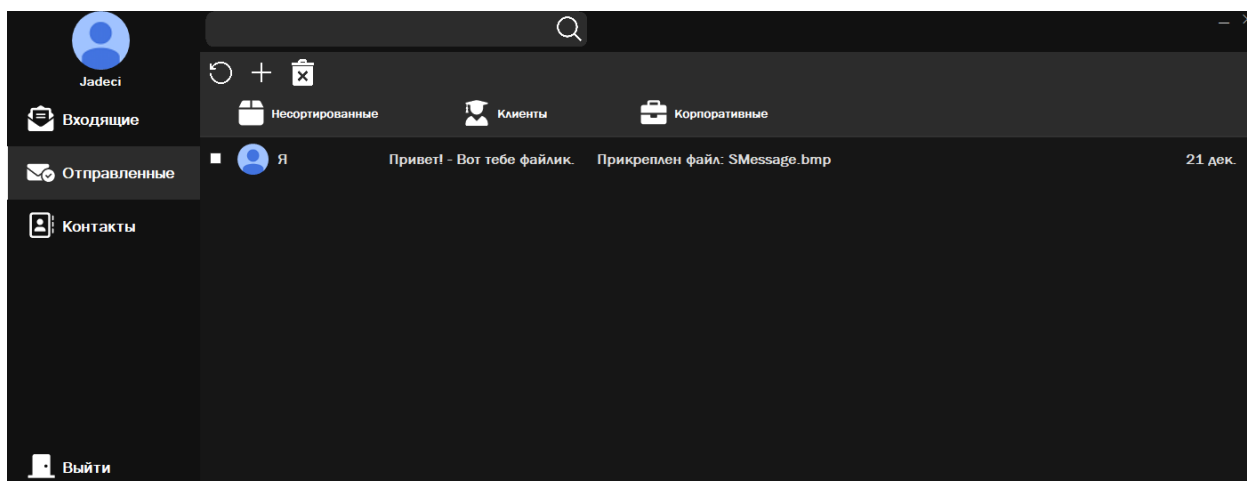


Рисунок 36 – Форма «Отправленные»

На рисунке 37 представленно форма показывает возможности просмотра и установки уже отправленного вами сообщения.

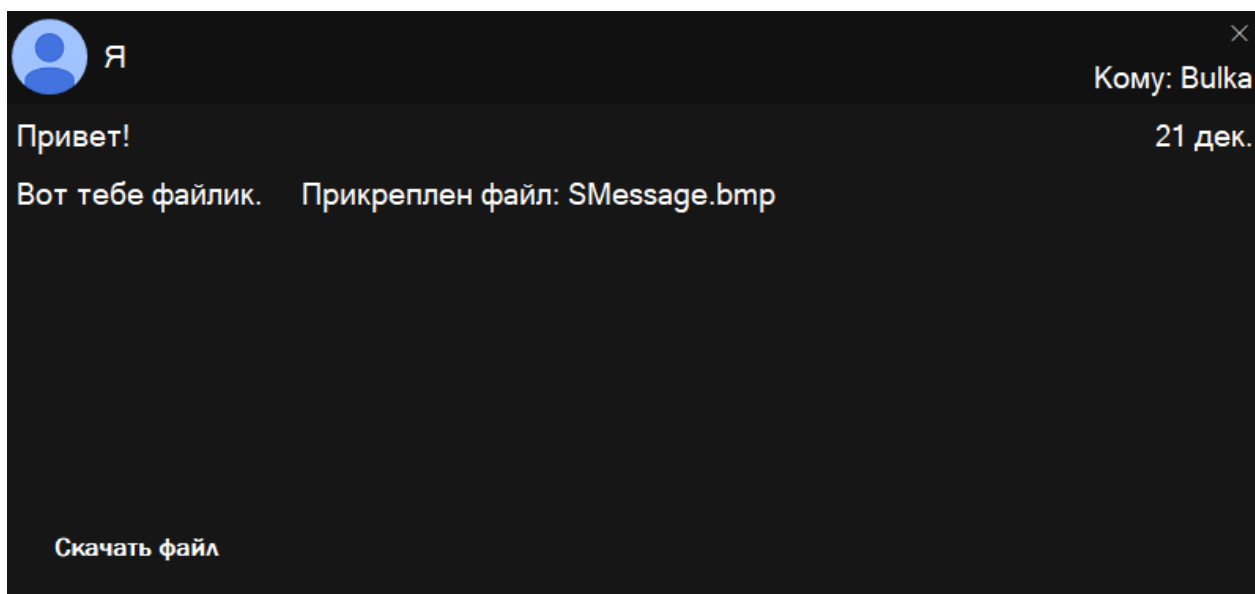


Рисунок 37 – Форма «Просмотр сообщения»

На рисунке 38 виден список сотрудников и их должности, также учет посещения, данная форма имеет возможность отправки сообщения и файла выбранному сотруднику.

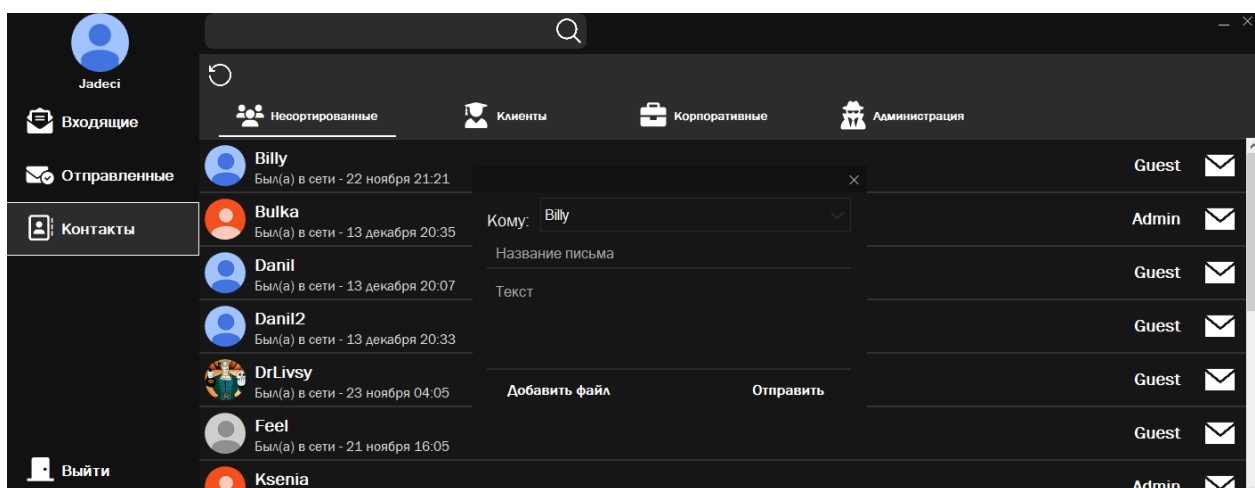


Рисунок 38 – Форма «Отправка сообщения»

3.2 Тестирование программного приложения

В рамках тестирования разработанного приложения было осуществлено модульное тестирование.

Unit-тестирование — это разновидность тестирования в программной разработке, которое заключается в проверке работоспособности отдельных функциональных модулей, процессов или частей кода приложения.

На рисунке с 39 по 41 показаны коды Unit-тестирования.

```

Ссылка 30 | 2/2 пройдены
public static DataTable sqlServer(string sqlStroke)
{
    try
    {
        SqlConnection con = new SqlConnection(sqlCon.ConString);
        SqlCommand com = new SqlCommand(sqlStroke, con);
        SqlDataAdapter adapter = new SqlDataAdapter(com);
        DataTable dat = new DataTable();
        adapter.Fill(dat);
        return dat;
    }
    catch
    {
        MessageBox.Show("Ошибка обращения к базе данных");
        return null;
    }
}

```

Рисунок 39 – Тестируемый код

Первый Unit-тест нацелен на проверку работы исключения, в случае если был получен некорректный SQL-запрос. Код Unit-теста и результат его.

```

✓ SqlCatchReturn
⌚ Длительность: 2,6 с

Запустить | Отладка | Показать в Обзорщике тестов

[TestMethod]
• | Ссылка 0
public void SqlCatchReturn()
{
    string sqlTsql = "Select";
    string result = null;
    DataTable sqlMB = sqlCon.sqlServer(sqlTsql);
    Assert.AreEqual(result, sqlMB);
}

```

Рисунок 40 – Проверка работы исключения

Второй Unit-тест необходим для проверки корректности полученного ответа. Код Unit-теста и результат его работы продемонстрирован на рисунке 41.

```

✓ SqlCheckReturn
⌚ Длительность: 62 мс

Запустить | Отладка | Показать в Обзорщике тестов

[TestMethod]
• | Ссылка 0
public void SqlCheckReturn()
{
    string sqlTsql = "Select Count(*) as 'Counts' From message.TABLE Where TABLE_TYPE = 'BASE TABLE'";
    string result = "7";
    DataTable dt = sqlCon.sqlServer(sqlTsql);
    string sqlMB = dt.Rows[0][0].ToString();

    Assert.AreEqual(result, sqlMB);
}

```

Рисунок 41 – Проверка корректности ответа

Результатом данного тестирования является успешный логин аккаунта без вызова исключений.

Тестирование черного ящика определяется как методика тестирования, при которой функциональность тестируемого приложения тестируется без учета внутренней структуры кода, деталей реализации и знания внутренних путей программного обеспечения. Этот тип тестирования полностью основан на требованиях и спецификациях программного обеспечения. В тестировании черного ящика мы просто фокусируемся на входах и выходах программной системы, не заботясь о внутренних знаниях программ.

Таблица 1 – Набор тестовых сценариев

Сортировка DataGridView	Нажатие на кнопку сортировки	DataGridView сортируется согласно порядку сортировки
Тестируемый компонент	Входные данные	Результат
Авторизация	Ввести данные пользователя, нажатие на кнопку «Вход»	Закрытие Формы «Авторизация», открытие формы «Главная»
Поиск по полю	Ввод данных в поле поиск	Список сортируется по вводимому полю
Функциональные кнопки приложения	Нажатие на кнопки «Закрыть», «Свернуть»	Приложения закрывается, Сворачивается

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Артюхина Д.Д., Коренькова Т.Н., Назарова О.И., Порядок подготовки, оформления и защиты курсовых и дипломных работ: учебно-методическое пособие – Старый Оскол: СТИ НИТУ «МИСиС», 2022. – 59 с
2. Варфоломеева, А. О. Информационные системы предприятия : учебное пособие / А. О. Варфоломеева, А. В. Коряковский, В. П. Романов. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2019. — 330 с. — (Среднее профессиональное образование). - ISBN 978-5-16-014729-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1002068>
3. Васильков, А. В. Безопасность и управление доступом в информационных системах: учеб.пособие / А.В. Васильков, И.А. Васильков. — Москва: ФОРУМ: ИНФРА-М, 2017. — 368 с. — (Среднее профессиональное образование).
4. Гвоздева, В. А. Основы построения автоматизированных информационных систем : учебник / В.А. Гвоздева, И.Ю. Лаврентьева. — Москва : ФОРУМ : ИНФРА-М, 2022. — 318 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0705-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1858934>
5. Голицына, О. Л. Основы проектирования баз данных : учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. — 2-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 416 с. — (Среднее профессиональное образование). - ISBN 978-5-91134-655-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1190668>
6. Гришина, Н. В. Основы информационной безопасности предприятия : учебное пособие / Н.В. Гришина. — 2-е изд., доп. — Москва : ИНФРА-М, 2023. — 216 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016719-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1900721>
7. Заботина, Н. Н. Методы и средства проектирования информационных систем : учебное пособие / Н.Н. Заботина. — Москва : ИНФРА-М, 2020. — 331

с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-16-015597-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1043093>

8. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие / Г.Н. Федорова. — Москва : КУРС : ИНФРА-М, 2022. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1858587>

9. Хорев, П. Б. Объектно-ориентированное программирование с примерами на C# : учебное пособие / П.Б. Хорев. — Москва : ФОРУМ : ИНФРА-М, 2023. — 200 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-713-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1895650>

10. Черников, Б. В. Управление качеством программного обеспечения : учебник / Б.В. Черников. — Москва : ФОРУМ : ИНФРА-М, 2022. — 240 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0902-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1850732>