МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

Проектирование модуля ELMA для чтения писем

**Курсовой проект**

по дисциплине «Проектирование информационных систем»

Выполнили студенты группы ПИ-43:

Ревякина Д.А., Рычкова А.А.

Проверил: Стариков А.И.

Дата защиты \_\_\_\_\_\_\_\_\_\_

Оценка\_\_\_\_\_\_\_\_\_\_\_

Киров 2018

**Оглавление**

[Введение 3](#_Toc532541617)

[I. Техническое задание на создание автоматизированной системы «Модуль ELMA для чтения писем» (по ГОСТ 34.602-89) 6](#_Toc532541618)

[II. Проектирование информационной системы 24](#_Toc532541619)

[2.1. Выбор инструмента проектирования 24](#_Toc532541620)

[2.2. Проектирование информационной системы 28](#_Toc532541621)

[2.2.1. UseCase Diagram 28](#_Toc532541622)

[2.2.2. Package Diagram 34](#_Toc532541623)

[2.2.3. Class Diagram 35](#_Toc532541624)

[2.2.4. Activity Diagrams 37](#_Toc532541625)

[Заключение 47](#_Toc532541626)

[Список использованной литературы 49](#_Toc532541627)

# **Введение**

В современном мире невозможно представить ни одно предприятие, не использующее программное обеспечение для хранения, дублирования важной документации и/или для автоматизации его процессов. В некоторых случаях требования к ПО настолько нестандартны, что готовые продукты не подходят. В таких случаях создается специализированная информационная система, удовлетворяющая требованиям.

Информационная система (ИС) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию. В зависимости от требований, предъявляемых к ИС, выбирается ее архитектура (настольные или распределенные, файл-серверные или клиент-серверные), степень автоматизации (автоматизированные или автоматические), характер обработки данных ИС (информационно-справочные или ИС обработки данных), и другие параметры. [2]

Стоит отметить, что анализ современного состояния рынка ИС показывает устойчивую тенденцию роста спроса на информационные системы организационного управления. Причем спрос продолжает расти именно на интегрированные системы управления. Автоматизация отдельной функции, например, бухгалтерского учета или сбыта готовой продукции, считается уже пройденным этапом для многих предприятий.

В то же время, многие проекты систем оказываются «замороженными»: требования некоторых оказывались практически невыполнимыми, а стоимость других оказывалась слишком велика. [9]

Актуальность данного исследования определяется тем, что современные предприятия вынуждены постоянно заниматься улучшением своей деятельности. Это требует разработки новых технологий и приемов ведения бизнеса и, конечно, внедрения новых, более эффективных методов управления и организации деятельности предприятий.

Избежать таких проблем еще на ранней стадии разработки позволяет проектирование ИС. Основными задачами, решению которых должна способствовать методология проектирования корпоративных ИС, являются следующие:

* обеспечивать создание корпоративных ИС, отвечающих целям и задачам организации, а также предъявляемым требованиям по автоматизации деловых процессов заказчика;
* гарантировать создание системы с заданным качеством в заданные сроки и в рамках установленного бюджета проекта;
* поддерживать удобную дисциплину сопровождения, модификации и наращивания системы;
* обеспечивать преемственность разработки, т.е. использование в разрабатываемой ИС существующей информационной инфраструктуры организации (задела в области информационных технологий).

Цель курсовой работы: анализ процессов компании, нуждающихся в автоматизации.

Результат выполнения работы: создание документации к будущей ИС.

Задачи курсовой работы:

* получение представлений о методах и средствах проектирования современных ИС;
* приобретение навыков работы со средствами UML их использования и роли в проектировании информационных систем.

Работа выполнена по заказу ООО «Брендмашина», которая указывает услуги в сфере бренд-менеджмента:

* [Платформа бренда](http://brand-machine.ru/services/brend-menedzhment/platforma-brenda/) – [концепция](http://brand-machine.ru/services/brend-menedzhment/platforma-brenda/kontseptsiya-brenda/) и [позиционирование](http://brand-machine.ru/services/brend-menedzhment/platforma-brenda/pozitsionirovanie-brenda/);
* [Стратегия бренда](http://brand-machine.ru/services/brend-menedzhment/strategiya-brenda/) – от [создания бренда](http://brand-machine.ru/services/brend-menedzhment/strategiya-brenda/sozdanie-brenda/) до [регистрации торговой марки](http://brand-machine.ru/services/brend-menedzhment/strategiya-brenda/registratsiya-torgovoy-marki/);
* [Вербальная идентификация](http://brand-machine.ru/services/brend-menedzhment/verbalnaya-identifikatsiya/) – [нейминг](http://brand-machine.ru/services/brend-menedzhment/verbalnaya-identifikatsiya/razrabotka-neyminga/), [слоганы](http://brand-machine.ru/services/brend-menedzhment/verbalnaya-identifikatsiya/reklamnyy-slogan/), [копирайтинг](http://brand-machine.ru/services/brend-menedzhment/verbalnaya-identifikatsiya/uslugi-kopirayta/);
* Визуальная [идентификация](http://brand-machine.ru/services/brend-menedzhment/dizayn-i-vizualnaya-identifikatsiya/)– [логотип](http://brand-machine.ru/services/brend-menedzhment/dizayn-i-vizualnaya-identifikatsiya/razrabotka-logotipa/), [фирменный стиль](http://brand-machine.ru/services/brend-menedzhment/dizayn-i-vizualnaya-identifikatsiya/razrabotka-firmennogo-stilya/), [упаковка](http://brand-machine.ru/services/brend-menedzhment/dizayn-i-vizualnaya-identifikatsiya/dizayn-upakovki/)и другие элементы;
* [Разработка знаковой системы](http://brand-machine.ru/services/brend-menedzhment/znakovaya-sistema/) – символы и сигналы, доносящие ключевые отличия бренда и продукта до сознания потребителя;
* [Брендинг компании](http://brand-machine.ru/services/brend-menedzhment/brending-kompanii/) – [подготовка](http://brand-machine.ru/services/brend-menedzhment/brending-kompanii/razrabotka-reklamnoy-kampanii/)и реализация рекламных активностей.

Ввиду большого объема системы, работа была поделена на 2 части и выполнена двумя студентами. Первый студент, Рычкова Алена, выполнил: UseCase, диаграмму активности «Генерация БД», диаграмму активности «Добавление письма в спам», диаграмму пакетов. Второй студент, Ревякина Диана, выполнил: диаграмму классов, диаграмму активности «Добавление нового пользователя», диаграмму активности «Выгрузка сообщений», диаграмму активности «Авторизация». Теоретическая часть была распределена поровну.

# **Техническое задание на создание автоматизированной системы «Модуль ELMA для чтения писем» (по ГОСТ 34.602-89)**

1. Общие сведения

1.1. Наименование системы

1.1.1. Полное наименование системы «Модуль ELMA для чтения писем»

1.1.2. Краткое наименование системы «Модуль для чтения писем».

1.2. Основания для проведения работ

Учебное задание на разработку проекта в рамках курсовой работы.

1.3. Наименование организаций – Заказчика и Разработчика

1.3.1. Заказчик: ООО «Брендмашина»

1.3.2. Разработчик: студенты группы ПИб-3301-01-00 Ревякина Д.А., Рычкова А.А.

1.4. Плановые сроки начала и окончания работы: с 01.10.2018 по 25.12.2018 г.

1.5. Источники и порядок финансирования: работа выполняется на безвозмездной основе.

1.6. Порядок оформления и предъявления заказчику результатов работ.

Работы по созданию проекта выполняются поэтапно в соответствии с календарным планом Проекта. По окончании всех работ Разработчик сдает Заказчику определенные договором отчетные документы.

2. Назначение и цели создания системы

2.1. Назначение системы

«Модуль чтения писем» предназначен для автоматизации сбора деловой переписки из различных источников, целью которого является повышение продуктивности сотрудников данного предприятия.

В рамках проекта автоматизируется информационно-аналитическая деятельность в следующих бизнес-процессах.

2.1.1. Коммуникации с клиентом

2.1.1.1. Предварительные переговоры

2 .1.1.2. Согласование объемов услуг и стоимости

2.1.1.3. Информирование заказчика

2.1.2. Взаимодействие с поставщиками

2.1.2.1. Получение информации по расходным материалам

2.1.2.2. Осуществление заказа

2.2. Цели создания системы

ИС «Модуль для чтения писем» создается с целью:

- повышения удобства работы с деловой перепиской;

- повышения качества обслуживания клиентов;

- повышения эффективности работы персонала;

-добавление подозрительных в спам (с указанием причин: наличие в письме ссылок, ведущих на подозрительные сайты; вложения в виде исполняемых файлов, например, \*.exe, \*.bat).

В результате создания хранилища данных должны быть улучшены значения следующих показателей:

- время, затрачиваемое на информационно-аналитическую деятельность;

-риски, связанные с вредоносными программами, полученными по ссылкам и исполняемыми файлами;

- время обслуживания клиентов;

- время на взаимодействие с поставщиком.

**3.** Характеристика объекта автоматизации.

Оргструктура компании представлена на рисунках 1-4.

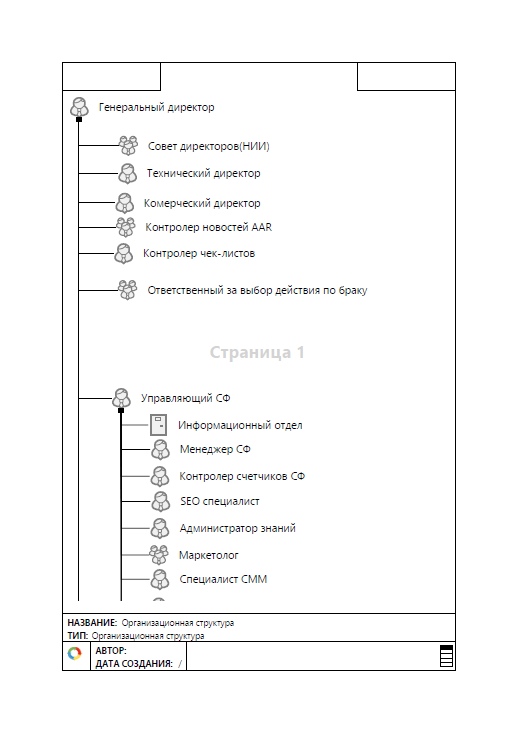


Рисунок 1 – Оргструктура компании

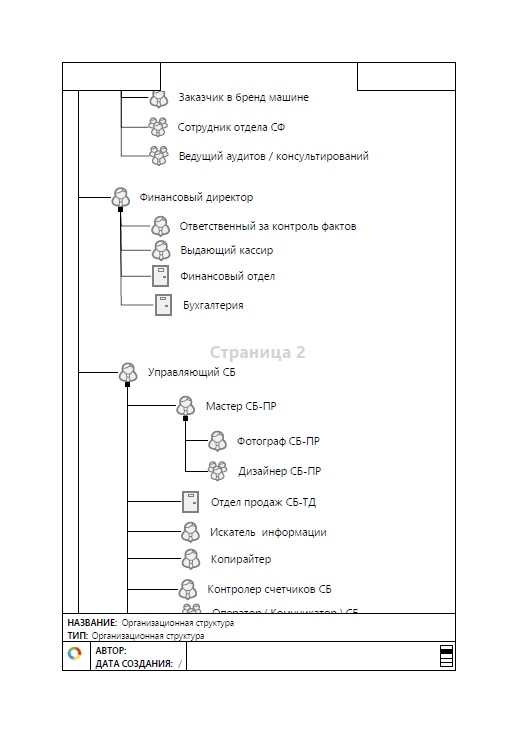


Рисунок 2 – Оргструктура компании

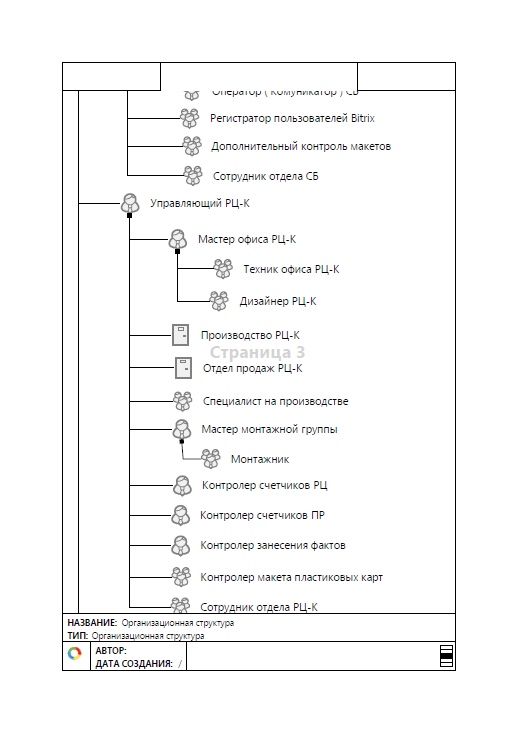


Рисунок 3 – Оргструктура компании

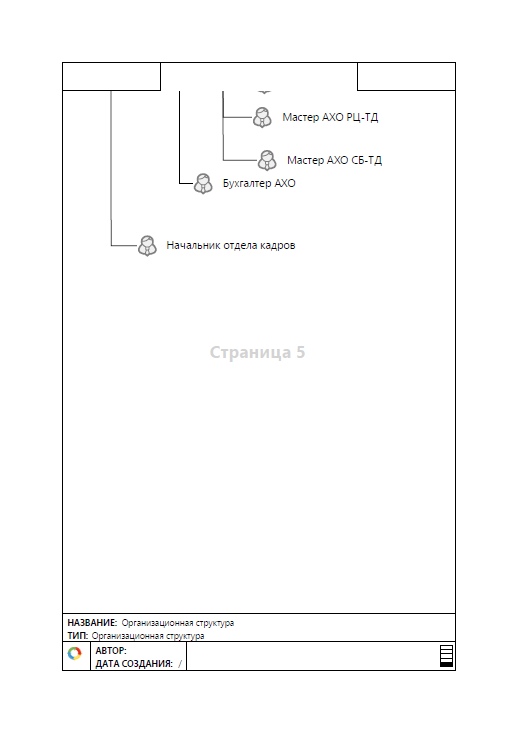
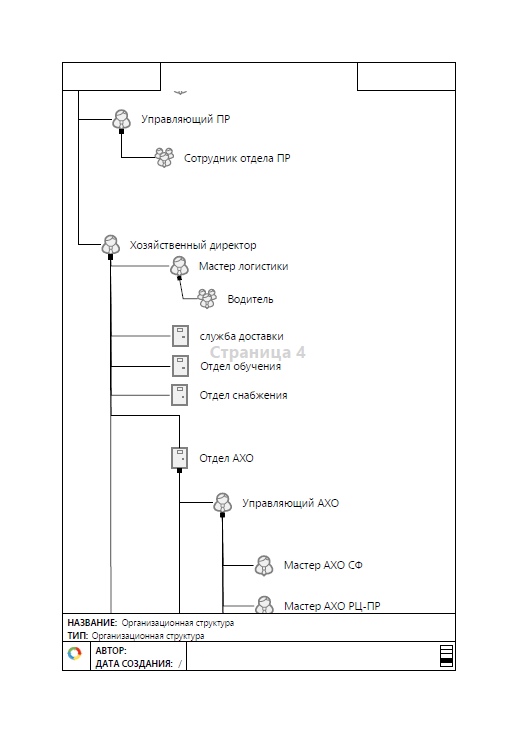


Рисунок 4 – Оргструктура компании

**4. Требования к системе**

4.1. Требования к системе в целом

4.1.1. Требования к структуре и функционированию системы

ИС «Модуль для чтения писем» должна быть реализована в архитектуре «Клиент-сервер» для настольных компьютеров на web-платформе.

База данных должна быть разработана в ПО MS SQL Server и должна взаимодействовать с БД типа Firebird.

В Системе предлагается выделить следующие функциональные подсистемы:  
- подсистема «[**Сущности**](https://www.prj-exp.ru/dwh/structure_of_etl_process.php)**»** предназначена для хранения данных о сущностях предметной области;

- подсистема **«Файлы»**, предназначенная для хранения файлов, отражающих электронные файлов различных видов, прикрепленных к письмам (например, аудиофайлы, картинки, документы, видеофайлы, а также ссылки на облачные хранилища, в которых хранятся данные ресурсы);  
- подсистема **«Статистика»**, предназначенная для формирования отчетности в виде экранных форм и вывода ее на печать.

- подсистема **администрирования** приложения.

Система должна поддерживать следующие режимы функционирования:

– основной режим работы пользователей;

– режим администрирования системы.

В *основном режиме функционирования* ИС должна обеспечивать:

- работу пользователей в режиме – 12 часов в день, 7 дней в неделю (12х7);  
- выполнение своих функций – сбор, обработка и загрузка данных; хранение данных, предоставление отчетности.

В *профилактическом режиме* Система должна обеспечивать возможность проведения следующих работ:

- администрирование пользователей системы и их обслуживание, включая добавление новых аккаунтов, управление настройками БД и указанных аккаунтов, установление прав доступа;

- администрирование системы и данных (резервное копирование, восстановление);

- ведение журнала работы пользователей системы, устранение критических ситуаций;

- диагностику и профилактику системы;

- доработку системы и внесение изменений в приложение.

Диагностирование Системы должно осуществляться штатными средствами, входящими в комплект поставки программного обеспечения. Для всех технических компонентов необходимо обеспечить регулярный и постоянный контроль состояния и техническое обслуживание.

4.1.2. Требования к численности и квалификации персонала системы и режиму его работы

4.1.2.1. Требования к численности персонала

В состав персонала, необходимого для обеспечения эксплуатации системы в рамках соответствующих подразделений Заказчика, необходимо выделение следующих ответственных лиц (ролей):

- Руководитель - 1 роль.

- Администратор информационной системы - 1 роль.

- Менеджер по продажам – 1 роль.

- Менеджер по работе с клиентами - 3 роли.

-Менеджер по закупкам – 1 роль.

Данные лица должны выполнять следующие функциональные обязанности.  
- Руководитель компании (конечный пользователь) обеспечивает общее руководство учреждением, анализирует результаты работы компании на основании получаемой отчетности.

- Администратор информационной системы обеспечивает функции администрирования системы, контроль и обслуживание технических и системных программных средств, имеет доступ ко всем функциям системы.

- Менеджер по продажам (конечный пользователь) обеспечивает работу по обновлению данных об услугах, оформляет заказы и договора, формирует необходимую отчетность. Регистрирует клиентов в базе.

- Менеджер по закупкам (конечный пользователь) отвечает за коммуникацию с поставщиками.

-Менеджер по работе с клиентами (конечный пользователь) ведет переговоры с клиентами: предоставляет информацию о товарах и услугах.

4.1.2.2. Требования к квалификации персонала

К квалификации персонала, эксплуатирующего Систему, предъявляются следующие требования.

- Конечный пользователь (руководитель, менеджеры) - знание соответствующей предметной области; знание основ работы с приложением, умение вести переговоры

- Администратор системы - знание аппаратных средств и системных программных средств на уровне обслуживания и администрирования; знание платформы разработки, используемой СУБД, основ администрирования системы.

4.1.3. Требования к эргономике и технической эстетике

Подсистема формирования и визуализации отчетности должна обеспечивать удобный для конечного пользователя интерфейс, отвечающий следующим требованиям.

В части внешнего оформления:

- интерфейсы подсистем должен быть типизированы;

- интерфейс системы должен быть русскоязычным;

- интерфейс должен быть дружественным, удобным и экономичным (требование юзабилити);

- формы статистики должны соответствовать утвержденным формам Заказчика;

- дизайн должен быть выдержан в фирменной стилистике программы «ELMA™».

В части диалога с пользователем:

- при возникновении ошибок в работе подсистемы на экран монитора должно выводиться сообщение с наименованием ошибки и с рекомендациями по её устранению на русском языке.

В части процедур ввода-вывода данных:

- должен быть предусмотрен контроль правильности ввода данных;

- для исключения дублирования ввода данных и быстрого заполнения ранее введенных данных должны использоваться справочники и словари.

4.1.4. Требования к защите информации от несанкционированного доступа

В качестве защиты от несанкционированного доступа предусмотрена система аутентификации типа «логин - пароль». Пароли формируются и задаются администратором, к паролям предъявляются требования надежности. Осуществляется посредством веб-портала программы «ELMA™»

4.1.4.1. Требования к информационной безопасности

Обеспечение информационное безопасности Системы должно удовлетворять следующим требованиям:

- Система должна иметь разграничение прав доступа пользователей по ролям;

- Система должна обеспечивать идентификацию пользователя по логину и паролю;

- Пароли хранятся в зашифрованном виде.

4.2. Требования к функциям, выполняемым системой.

4.2.1. Подсистема «Сущности»

В подсистеме реализуются функции ведения следующих сущностей:

Email:

|  |  |  |
| --- | --- | --- |
| № | Имя атрибута | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | EmailString | Varchar, 512 |

Phone:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | PhoneString | Varchar, 256 |

UC\_AccessEmailAccount:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | EmailAccount | Bigint |
| 3 | CurrentUser | Bigint |
| 4 | IsOpen | Bit |
| 5 | IsDeleted | Bit |

UC\_Account:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | Login | Varchar, 1024 |
| 3 | Password | Varchar, 1024 |
| 4 | ChannelCommunication | Bigint |
| 5 | IsDeleted | Bit |

UC\_AsteriskAccount:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | Server | Varchar, 20 |

UC\_AsteriskCall:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | EndDate | DateTime |
| 3 | StartDate | DateTime |
| 4 | LineNumber | Varchar, 1024 |
| 5 | Respondent | Bigint |
| 6 | Record | Bigint |
| 7 | Comment | Varchar, MAX |
| 8 | JsonContent | Varchar, 2048 |
| 9 | JsonFileName | Varchar, 2048 |

UC\_Attachment:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | Uid | Uniqueidentifier |
| 3 | Name | Varchar,1024 |
| 4 | CreateDate | DateTime |
| 5 | ContentFilePath | Varchar,1024 |
| 6 | ContentUrl | Varchar,3072 |
| 7 | Message | Bigint |
| 8 | IsDeleted | Bit |

UC\_Email:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | Uid | Bigint |
| 3 | EmailAccount | Bigint |
| 4 | ChanedDate | DateTime |
| 5 | ChangedAuthor | Bigint |
| 6 | DeletedAuthor | Bigint |
| 7 | Body | Varchar, MAX |
| 8 | Subject | Varchar, 1204 |
| 9 | IsHtml | Bit |
| 10 | IsRead | Bit |
| 11 | NotAnswer | Bigint |
| 12 | ReasonSuspicious | Bigint |

UC\_EmailAccount:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | Server | Varchar, 1024 |
| 3 | Port | Bigint |
| 4 | SmtpPort | Bigint |
| 5 | IsSsl | Bit |
| 6 | Login | Varchar, 1024 |
| 7 | Password | Varchar, 1024 |
| 8 | IsDeleted | Bit |

UC\_GeneralSettings:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | SignatureEmail | Varchar,MAX |
| 3 | IsDeleted | Bit |

UC\_HrefDownloadFileInfo:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | Message\_Id | Bigint |
| 3 | Lenght | Bigint |
| 4 | FileName | Varchar, 1024 |
| 5 | Href | Varchar, 2048 |
| 6 | IsDeleted | Bit |

UC\_MarkMessage:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | MarkMessage | Bigint |
| 3 | UC\_Message\_Id | Bigint |
| 4 | CurrentUser\_Id | Bigint |
| 5 | IsDeleted | Bit |

UC\_Message:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | Account\_Id | Bigint |
| 3 | [From] | Varchar, 512 |
| 4 | [To] | Varchar, 512 |
| 5 | ChannelCommunication | Bigint |
| 6 | DirectionMessage | Bigint |
| 7 | CreateDate | DateTime |
| 8 | IsDeleted | Bit |

UC\_ShowAccount:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | CurrentUser\_Id | Bigint |
| 3 | Account\_Id | Bigint |
| 4 | IsDeleted | Bit |

UC\_SpamAddress:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | Address | Varchar, 768 |
| 3 | Crc | Bigint |
| 4 | ChangedAuthor | Bigint |
| 5 | IsDeleted | Bit |

UC\_TemplatesMessage:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | Subject | Varchar, 1024 |
| 3 | Body | Varchar, MAX |
| 4 | ChannelCommunication | Bigint |
| 5 | IsDeleted | Bit |

UC\_UserSettings:

|  |  |  |
| --- | --- | --- |
| № | Имя реквизита | Тип, длина, свойства |
| 1 | Id | Integer |
| 2 | CurrentUser | Bigint |
| 3 | FilterList | Varchar, MAX |
| 4 | EmailAccount | Bigint |
| 5 | IsDeleted | Bit |

4.3. Требования к видам обеспечения

4.3.1 Требования к математическому обеспечению

Не предъявляются.

4.3.2. Требования к информационному обеспечению

Состав и структура данных приведена в описании подсистемы «Сущности». Обязанность ведения возлагается на роль администратора.

4.3.2.1. Требования к защите данных от разрушений при авариях и сбоях в электропитании системы.

Информация в базе данных системы должна сохраняться при возникновении аварийных ситуаций, связанных со сбоями электропитания. Система должна иметь бесперебойное электропитание, обеспечивающее её нормальное функционирование в течение 15 минут в случае отсутствия внешнего энергоснабжения, и 5 минут дополнительно для корректного завершения всех процессов.

Резервное копирование данных должно осуществляться на регулярной основе, в объёмах, достаточных для восстановления информации в подсистеме хранения данных.

4.3.2.2. Требования к контролю, хранению, обновлению и восстановлению данных:

- система должна протоколировать события, связанные с изменением своего информационного наполнения данных.

- хранение исторических данных в системе должно производиться не более чем за 5 (пять) предыдущих лет. По истечению данного срока данные должны переходить в архив;

- исторические данные, превышающие пятилетний порог, должны храниться в архиве на внешних носителях с возможностью их восстановления.

4.3.3. Требования к лингвистическому обеспечению

При реализации системы должны использоваться отечественные программные продукты. Для разработки документации и в разработке системы используются следующие языки:

язык стандартов UML;

язык запросов SQL;

язык программирования C#.

4.3.4. Требования к программному обеспечению

Система должна функционировать в OC Windows версии 7 и выше.

Система должна быть реализована на платформе Microsoft Visual Studio, веб-приложение ASP.NET. Данная система должна быть реализована в MS SQL Server, Microsoft SQL Server Management Studio.

4.3.5. Требования к техническому обеспечению

Так как система реализуется в архитектуре «Клиент-сервер», то предъявляется 2 типа требований: для машины, на которой будет запущена клиентская часть и для машины, которая будет использоваться в качестве сервера (минимальные и рекомендуемые требования соответственно).

Минимальные требования

Оперативная память - 2ГБ.

Процессор - Pentium 4 с частотой 1,5 GHz либо Athlon XP 1500+ и выше.

Свободное место на диске - 300 МБ (без установки программы ELMA) и 1 ГБ (с установкой программы ELMA).

Устройства взаимодействия с пользователем – клавиатура и мышь.

Сетевая карта.

Рекомендуемые требования

Оперативная память - 4ГБ и более.

Процессор – Intel Core I5 с частотой 2 GHz и выше.

Свободное место на диске - 1 ГБ и больше.

Устройства взаимодействия с пользователем – клавиатура и мышь.

Сетевая карта. [6]

4.3.6. Требования к организационному обеспечению

Основными пользователями Системы являются сотрудники учреждения Заказчика. Состав сотрудников определяется штатным расписанием Заказчика, которое, в случае необходимости, может изменяться. Каждому сотруднику устанавливается роль в соответствии с выполняемыми функциями. Для снижения ошибочных действий пользователей должно быть разработано полное и доступное руководство пользователя и проведено обучение пользованию системой.

**5. Состав и содержание работ по созданию системы**

Работы по созданию системы выполняются в три этапа.

**1 этап**. Предпроектное обследование, включающее в себя постановку задачи автоматизации, выявление и анализ требований. Результатом выполнения данного этапа работ является согласованное и утвержденное заказчиком техническое задание. Продолжительность этапа работ 2 месяца.

**2 этап**. Реализация проекта, включающая в себя проектирование и разработку программного продукта, интеграционное тестирование, разработку рабочей документации. Результатом этапа является готовый программный продукт, в полном объеме реализующий указанную в данном техническом задании функциональность. Продолжительность – 2 месяца).

**3 этап**. Передача системы заказчику, включающая в себя выполнение приемо-сдаточных испытаний. Продолжительность 2 недели.

Конкретные сроки выполнения стадий и этапов разработки и создания Системы определяются Планом выполнения работ, являющимся неотъемлемой частью Договора на выполнение работ по настоящему техническому заданию.

**6. Порядок контроля и приёмки системы.**

Порядок контроля и приемки Системы представлены в документе «Программа и методика испытаний программного продукта»

**7. Требования к документированию.**

Проект предполагает разработку и оформление следующих документов:

|  |  |
| --- | --- |
| **Этап** | **Документ** |
| Предпроектное обследование | UML диаграммы деятельности |
| Техническое задание |
| Программа и методика испытаний программного продукта |
| Проектирование | Ведомость эксплуатационных документов |
| Ведомость машинных носителей информации |
| Паспорт |
| Общее описание системы |
| Технологическая инструкция |
| Руководство пользователя |
| Программный продукт |
| Документация на программный продукт |
| Структура базы данных |
| Тексты программных модулей |
| Приемо-сдаточные испытания | Протокол испытаний |
| Акт завершения работ |

Вся документация должна быть подготовлена и передана как в печатном, так и в электронном виде (в формате Microsoft Word).

Настоящее Техническое Задание разработано на основе следующих документов и информационных материалов:

- ГОСТ 34.601-90 «Автоматизированные системы. Стадии создания».

- ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы.»

1. Проектирование информационной системы
   1. Выбор инструмента проектирования

Для написания данной курсовой работы нами был выбран язык моделирования UML. UML (англ. Unified Modeling Language) — язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур. [7]

Диаграммы создаются с помощью специальных CASE средств, например Rational Rose и Enterprise Architect. На основе технологии UML строится единая информационная модель. Приведенные выше CASE средства способны генерировать код на различных объектно-ориентированных языках, а также обладают очень полезной функцией реверсивного инжиниринга. (Реверсивный инжиниринг позволяет создать графическую модель из имеющегося программного кода и комментариев к нему.)

На рисунке 5 представлена схема всех видов UML диаграмм.

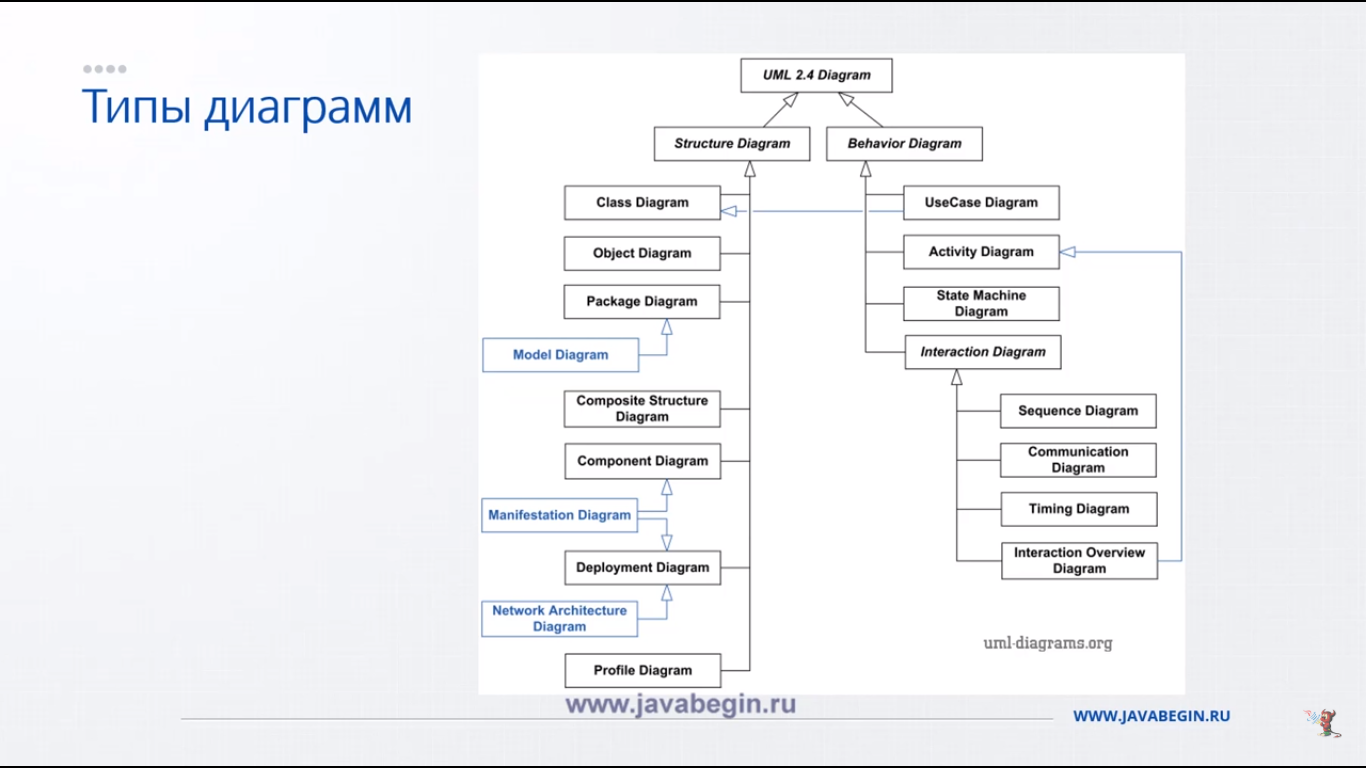


Рисунок 5 – Виды UML диаграмм

Данная схема представляет собой разделение UML диаграмм на структурные (левая часть схемы) и поведенческие (правая часть схемы). Объясним преимущества некоторых из них. [1]

**Структурные диаграммы:**

1. Диаграмма классов (Class Diagram) позволяет:

* описывать классы, интерфейсы, связи, поля методов;
* структуру в стиле ООП;
* понять работу кода без изучения самого кода;
* сначала рисовать диаграммы, из которых автоматически можно сгенерировать код на одном из языков программирования на выбор.

1. Диаграмма объектов (Object Diagram) имеет:

* состояние экземпляров классов с конкретными значениями полей в определенный момент времени;
* сходство с диаграммой классов, но в режиме runtime (во время работы программы).

1. Диаграмма пакетов (Package Diagram) имеет такие преимущества, как:

* Отображение вложенности и связей между пакетами;
* более высокий уровень, чем классы.

1. Model Diagram используется для:

* описания «слоев» проекта – из каких глобальных уровней состоит проект;
* многоуровневых приложений;
* в ТЗ для общего описания частей проекта.

**Поведенческие диаграммы:**

1. Диаграмма вариантов использования (UseCase Diagram) имеет такие преимущества, как:

* описание возможных сценариев работы с системой с точки зрения пользователей;
* возможные пути использования пользователем системы;
* описание всех участников системы (акторы).

1. Диаграмма активности (Activity Diagram) позволяет описывать:

* возможные бизнес-процессы приложения;
* взаимодействие потоков, пошаговое представление действий;
* блоки UseCase диаграмм.

1. Диаграмма последовательности (Sequense Diagram) показывает:

* последовательность взаимодействия объектов для определенного бизнес-процесса;
* как объекты друг друга вызывают и какие параметры передают;
* в отличии от Class Diagram и Object Diagram – показывает объекты «в действии»;
* время жизни объектов (создание, удаление).

1. Deployment Diagram содержит:

* описание архитектуры, топологии системы (ОС, БД, сервера);
* информация для администраторов.

Диаграммы логически связаны между собой. Связь показана на рисунке 6.

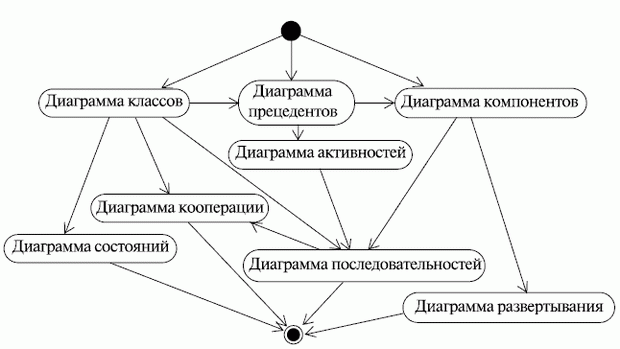


Рисунок 6 – Связь диаграмм

В ходе выполнения курсовой работы были разработаны следующие виды диаграмм: UseCase Diagram, Activity Diagram, Package Diagram, Class Diagram. Все диаграммы были выполнены в программном продукте StarUml, и представлены в части 2.2. [3]

# **Проектирование информационной системы**

# **UseCase Diagram**

На рисунке 7 представлена UseCase диаграмма для системы, рассматриваемой в курсовой работе.

**Изображение выглядит как текст, газета

Описание создано с высокой степенью достоверности**

Рисунок 7 – UseCase диаграмма

На данной диаграмме выделено 3 действующих лица (актера) – администратор системы, руководитель компании, менеджер по работе с клиентами. Также было выделено 18 вариантов использования, некоторые из которых можно описать подробнее:

1. Вариант использования (use case 1): авторизация

*Цель в контексте*: вход в систему под определенным аккаунтом.

*Сфера деятельности*: ELMA.

*Предпосылки (предварительное условие)*: наличие логина и пароля.

*Состояние успешного завершения*: вход в систему осуществлен.

*Неуспешное завершение*: вход не осуществлен.

*Первичный актер*: администратор системы.

*Поддерживающий актер*: менеджер по работе с клиентами, руководитель компании.

*Триггер:* авторизация в ИС.

**Успешный сценарий (основной поток событий).**

* + 1. Запуск системы.
    2. Ввод авторизационных данных.
    3. Вход в систему.
    4. Открытие стартовой страницы системы.

**Альтернативный сценарий (расширение, extention).**

1а. Данные введены неверно, вход не выполнен.

1.а.1. Восстановление данных.

**Вариации сценария (variations):** да.

1. Вариант использования (use case 2): добавление нового пользователя

*Цель в контексте*: добавление нового пользователя в базу данных.

*Сфера деятельности*: ELMA.

*Предпосылки (предварительное условие)*: данного пользователя нет в имеющейся базе.

*Состояние успешного завершения*: новый пользователь добавлен и ему открыт доступ к системе.

*Неуспешное завершение*: данный пользователь уже имеется в системе/пользователь не добавлен в систему.

*Первичный актер*: администратор системы.

*Поддерживающий актер*: менеджер по работе с клиентами, руководитель компании.

*Триггер:* предоставление доступа к системе новому пользователю.

**Успешный сценарий (основной поток событий).**

* + 1. Ввод пользовательских данных.
    2. Данного пользователя нет в имеющейся базе.
    3. Добавление личных данных в базу.
    4. Открытие доступа.

**Альтернативный сценарий (расширение, extention).**

1а. Данный пользователь уже есть в базе.

1.а.1. Добавление в базу не происходит.

2а. Не хватает данных для добавления пользователя.

2.а.2. Добавление в базу не происходит.

**Вариации сценария (variations):** да.

1. Вариант использования (use case 3): написать сообщение

*Цель в контексте*: написание и отправка сообщения клиенту(поставщику).

*Сфера деятельности*: ELMA.

*Предпосылки (предварительное условие)*: наличие входящего сообщения от клиента/отсутствие материалов.

*Состояние успешного завершения*: сообщение отправлено.

*Неуспешное завершение*: сообщение не отправлено.

*Первичный актер*: менеджер по работе с клиентами.

*Поддерживающий актер*: руководитель компании.

**Успешный сценарий (основной поток событий).**

* + 1. Выбор получателя сообщения.
    2. Написание сообщения.
    3. Отправка сообщения.

**Вариации сценария (variations):** нет.

1. Вариант использования (use case 4): пометить сообщение как спам

*Цель в контексте*: добавление сообщения в спам.

*Сфера деятельности*: ELMA.

*Предпосылки (предварительное условие)*: наличие в письме ссылок, ведущих на подозрительные сайты.

*Состояние успешного завершения*: письмо помечено как спам.

*Первичный актер*: руководитель компании.

*Поддерживающий актер*: менеджер по работе с клиентами.

*Триггер:* предоставление доступа к системе новому пользователю.

**Успешный сценарий (основной поток событий).**

* + 1. Проверка письма на подозрительные ссылки, и др.
    2. Нахождение в письме таких ссылок.
    3. Добавление письма в спам.

**Вариации сценария (variations):** нет.

1. Вариант использования (use case 5): выгрузить сообщения.

*Цель в контексте*: загрузка новых сообщений.

*Сфера деятельности*: ELMA.

*Предпосылки (предварительное условие)*: сообщения не обновлены/вход на страницу.

*Состояние успешного завершения*: загружены новые сообщения на страницу.

*Неуспешное завершение*: новых сообщений нет.

*Первичный актер*: руководитель компании.

*Поддерживающий актер*: менеджер по работе с клиентами.

*Триггер:* выгрузка новых сообщений.

**Успешный сценарий (основной поток событий).**

* + 1. Отправка запроса сервером.
    2. Проверка новых сообщений.
    3. Выгрузка новых сообщений.
    4. Вывод сообщений на страницу.

**Альтернативный сценарий (расширение, extention).**

1а. Новых сообщений нет.

1.а.1. Выгрузка сообщений не происходит.

**Вариации сценария (variations):** да.

1. Вариант использования (use case 6): сгенерировать БД

*Цель в контексте*: генерация базы данных системы.

*Сфера деятельности*: ELMA.

*Предпосылки (предварительное условие)*: база данных ещё не существует.

*Состояние успешного завершения*: база данных создана.

*Неуспешное завершение*: база данных не создана.

*Первичный актер*: администратор системы.

**Успешный сценарий (основной поток событий).**

* + 1. Проверка наличия БД.
    2. Создание БД.
    3. Создание недостающих сущностей.
    4. Проверка наличия данных.
    5. Запись данных в БД.

**Альтернативный сценарий (расширение, extention).**

1а. База данных уже существует.

1.а.1. Проверка на наличие всех сущностей и добавление недостающих.

**Вариации сценария (variations):** да.

# **Package Diagram**

Одним из условий функционирования данной программы является стабильное взаимодействие с базой данных. При этом сама структура БД должна быть составлена правильно.

Так как оригинальная программа ELMA работает с БД через ORM, то было решено использовать пакет NHibernate. NHibernate — ORM-решение для платформы Microsoft .NET. Это бесплатная библиотека с открытым кодом, распространяется под лицензией GNU Lesser General Public License. NHibernate позволяет отображать объекты бизнес-логики на реляционную базу данных. По заданному XML-описанию сущностей и связей NHibernate автоматически создает SQL-запросы для загрузки и сохранения объектов.

NHibernate является портом на .NET популярной на платформе Java библиотеки Hibernate. Версия 1.0 покрывает набор возможностей Hibernate 2.1, а также часть возможностей Hibernate 3. NHibernate 1.2.1, выпущенная в ноябре 2007 года, предоставляет ещё больше возможностей из Hibernate 3, поддерживает .NET 2.0, хранимые процедуры, generics и nullable типы. NHibernate 2.0.1, выпущенная в конце сентября 2008 года предоставляет бо́льшую часть возможностей Hibernate 3.[10]

Для создания БД с помощью данного пакета создаются модели, которые представляют из себя описание полей каждой сущности с помощью языка С#. Затем, пишутся так называемые маппинги (от англ. Map – карта), позволяющие осуществлять трансляцию из БД в бизнес-объекты и обратно с помощью связывания через XML файл или связывания атрибутов на классы и свойства. Таким образом, после внесения изменений в класс app.config, при первом запуске создается скрипт SQL и исполняется на стороне сервера MS SQL, создавая базу данных. Если в ходе дальнейшей разработки в структуру БД необходимо внести изменения, то они вносятся непосредственно в код проекта. [4][5]

На рисунке 8 изображена диаграмма пакетов.

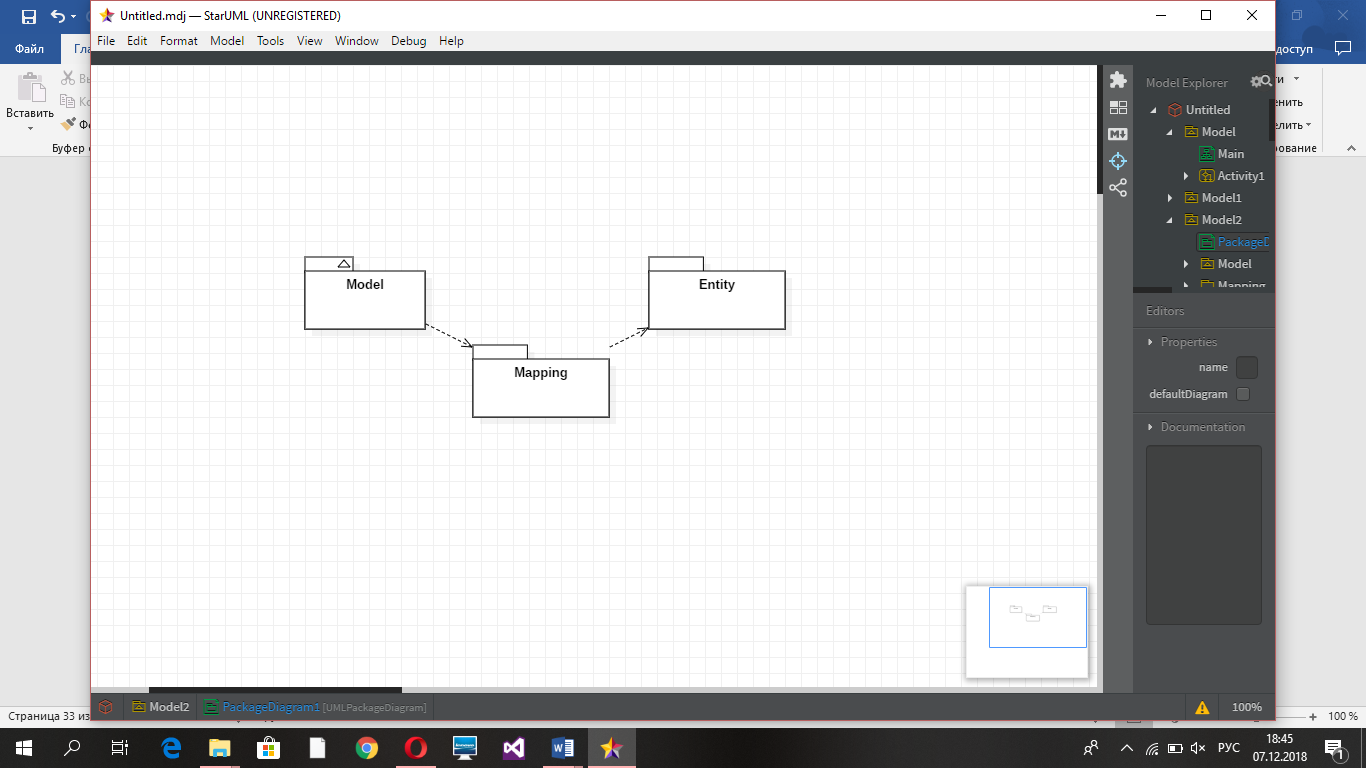


Рисунок 8 – диаграмма пакетов.

# **Class Diagram**

На рисунке 9 изображена диаграмма классов для будущей ИС.

Изображение выглядит как текст, карта

Описание создано с очень высокой степенью достоверности

Рисунок 9 – Диаграмма классов для будущей ИС

Данная диаграмма классов содержит классы, которые будут использованы для создания базы данных при помощи пакета NHibernate.

Каждый из классов содержит набор атрибутов и методов, которые отображают их динамическое поведение. Атрибуты класса определяют структуру и состав данных, хранимых в объектах этого класса. Для данной диаграммы выбран тип связи обобщение (англ. Aggregation), так как диаграмма отображает только классы-потомки и классы-родителей.

Класс Entity является базовым и его функции унаследуют все классы. В некоторых случаях методы переопределяются (например, Save()).

Для системы основными классами являются UC\_Account, UC\_Message и UC\_Attachment.

Класс UC\_Account используется для хранения данных о связке логин-пароль, а также о типе аккаунта, с которого ведется этот диалог (email, vk, телефон, и т.д.). Является базовым классом для UC\_EmailAccount, который содержит настройки соединения.

Класс UC\_Message нужен для хранения данных о сообщении, таких как отправитель, получатель, дата, пользователь, ответственный со стороны компании за данную переписку. Является базовым для класса UC\_Email, который хранит параметры электронного письма.

Класс UC\_Attachment содержит информацию о файле, прикрепленном к письму (название, путь к файлу, содержимое, и т.д.).

# **Activity Diagrams**

Диаграмма активностей (видов деятельности) - один из доступных видов диаграмм, поддерживаемых Flexberry Designer. Она, как и диаграмма состояний, отражает динамические аспекты поведения системы. По существу, эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой.

На рисунке 10 представлены основные элементы нотации UML для данного типа диаграмм.

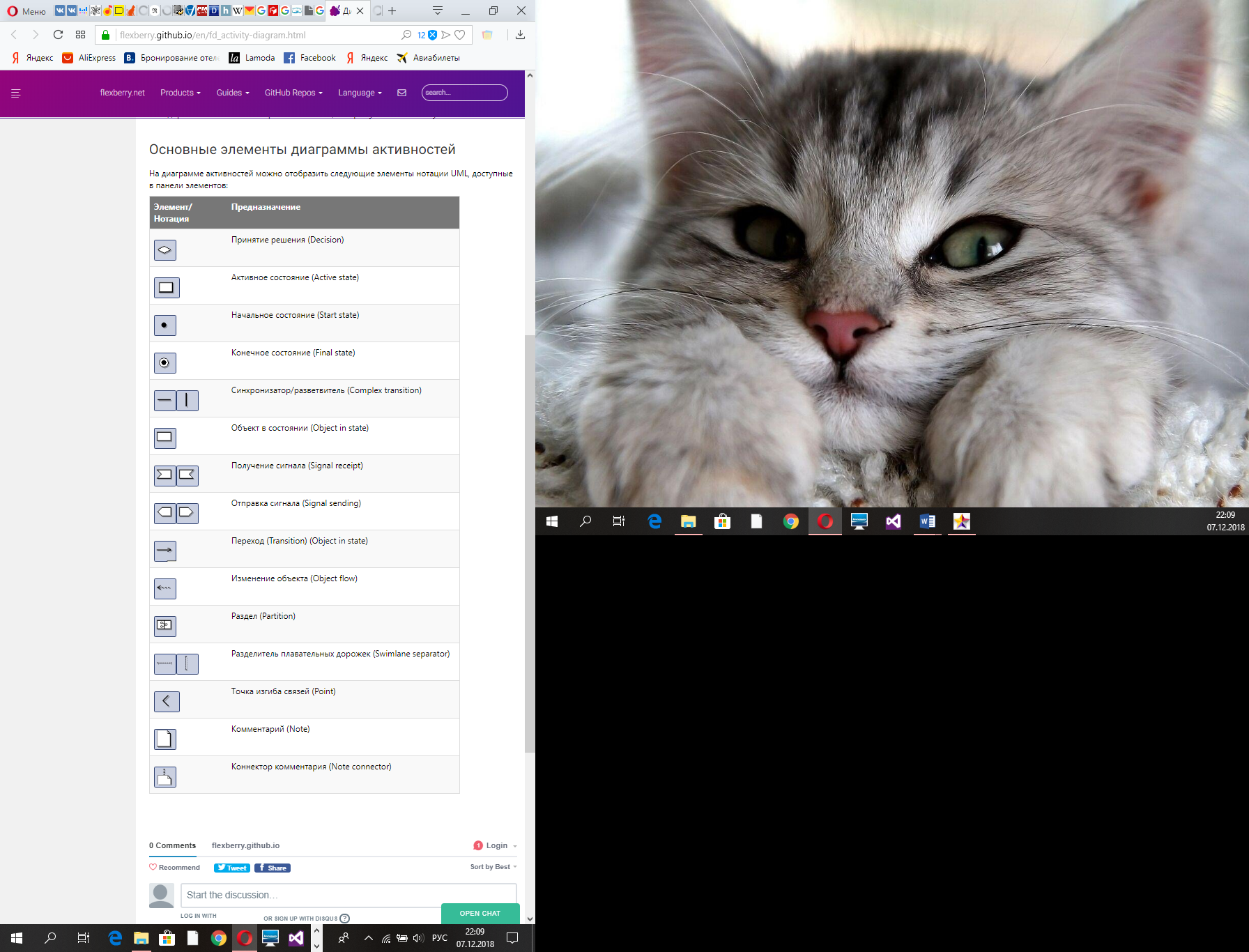


Рисунок 10 – Основные элементы диаграмм активности

На рисунке 11 представлена диаграмма активности для прецедента «Авторизация».

Изображение выглядит как снимок экрана, текст

Описание создано автоматически

Рисунок 11 – Диаграмма активности для прецедента «Авторизация»

Диаграмма активности для прецедента «Авторизация» начинается с начального узла (initial node), затем выполняется операция «Запуск системы». Затем пользователь осуществляет операцию «Ввод авторизационных данных». После этого идет блок принятия решения (decision) «Проверка данных на корректность»:

1. данные введены корректно, то осуществляются операции «Вход в систему» и «Открытие стартовой страницы», после чего осуществляется переход в конечное состояние (final state);
2. данные введены некорректно, то осуществляется переход на блок принятия решения (decision) «Забыли пароль»:
3. если нет, то идет перенаправление на операцию «Ввод авторизационных данных»;
4. если данные забыты, выполняется операция «Восстановление данных» и «Вход не выполнен» , после чего осуществляется переход в конечное состояние (final state).

На рисунке 12 показана диаграмма активности для прецедента «Добавление нового пользователя»

Изображение выглядит как текст

Описание создано автоматически

Рисунок 12 – Диаграмма активности для прецедента «Добавление нового пользователя»

Диаграмма активности для прецедента «Добавление нового пользователя» начинается с начального узла (initial node), затем идет блок принятия решения (decision) «Существует ли данный пользователь в уже имеющейся базе»:

1. пользователь существует - осуществляется переход в конечное состояние (final state);
2. пользователь не существует - то осуществляется переход на блок принятия решения (decision) «Проверка введенных данных на корректность»:
3. если данные не верны, то осуществляется возврат на операцию «Ввод данных пользователя»;
4. если данные верны, выполняется операция «Добавление пользователя в базу» и «Вывод нового пользователя в списке» , после чего осуществляется переход в конечное состояние (final state).

На рисунке 13 показана диаграмма активности для прецедента «Выгрузка сообщений».

Изображение выглядит как текст

Описание создано автоматически

Рисунок 13 – Диаграмма активности для прецедента «Выгрузка сообщений»

Диаграмма активности для прецедента «Выгрузка сообщений» начинается с начального узла (initial node), затем идет блок принятия решения (decision) «Запрос к серверу»:

1. сервер не отвечает - осуществляются переходы на операцию «Вывод сообщения о недоступности сервера» и в конечное состояние (final state);
2. сервер отвечает - осуществляется переход на операцию «Получение данных от сервера» и выход на блок принятия решения (decision) «Проверка новых сообщений»:
3. если новых сообщений нет, то осуществляется переход в конечное состояние (final state);
4. если новые сообщения есть, выполняются операции «Отбор сообщений по критериям фильтра», «Вывод сообщений» и «Оповещение», после чего осуществляется переход в конечное состояние (final state).

На рисунке 14 представлена диаграмма активности для прецедента «Генерация БД».

Изображение выглядит как текст

Описание создано автоматически

Рисунок 14 – Диаграмма активности для прецедента «Генерация БД»

Диаграмма активности для прецедента «Генерация БД» начинается с начального узла (initial node), затем идет блок принятия решения (decision) «Проверка наличия БД»:

1. БД существует - осуществляется переход на блок принятия решения «Проверка наличия всех сущностей»:
2. если все сущности есть, то выполняется переход в конечное состояние (final state);
3. если не все сущности есть, то происходит переход к операции «Создание недостающих сущностей»;
4. БД не существует - осуществляются переходы на операции «Создание БД», «Создание недостающих сущностей» и выход на блок принятия решения (decision) «Проверка наличия данных»:
5. если данных нет, то осуществляется переход в конечное состояние (final state);
6. если есть, выполняется операция «Запись данных в БД», после чего осуществляется переход в конечное состояние (final state).

На рисунке 15 показана диаграмма активности для прецедента «Добавление письма в спам».

Изображение выглядит как текст

Описание создано автоматически

Рисунок 15 – Диаграмма активности для прецедента «Добавление письма в спам»

Диаграмма активности для прецедента «Добавление письма в спам» начинается с начального узла (initial node), затем выполняется операция «Получение сообщения с сервера». После этого идет блок принятия решения (decision) «Проверка сообщения системой по имеющимся критериям»:

1. соответствует критериям - осуществляется операция «Добавление письма в спам», после чего осуществляется переход в конечное состояние (final state);
2. не соответствует – осуществляется операция «Вывод сообщения», затем происходит переход на блок принятия решения (decision) «Проверка пользователем на наличие подозрительных элементов»:
3. если элементы не найдены, то идет перенаправление на операцию «Просмотр сообщения пользователем» и переход в конечное состояние (final state);
4. если элементы найдены, выполняются операции «Добавление письма в спам вручную» и «Добавление причины в комментарий» , после чего осуществляется переход в конечное состояние (final state).

# **Заключение**

В ходе выполнения курсовой работы были достигнуты поставленные цели, такие как: применение на практике знаний, полученных в процессе изучения курса «Проектирование ИС» и получение практических навыков создания документации на языке UML.

Был произведен анализ процессов компании, нуждающихся в автоматизации. Также в ходе выполнения курсовой работы были достигнуты следующие задачи:

* получены представления о методах и средствах проектирования современных ИС;
* приобретены навыков работы со средствами UML их использования и роли в проектировании информационных систем.

Использование информационных технологий для управления предприятием делает любую компанию более конкурентоспособной за счет повышения ее управляемости и адаптируемости. Подобная автоматизация позволяет:

1. Повысить эффективность управления компанией за счет обеспечения руководителей и специалистов максимально полной, оперативной и достоверной информацией на основе единого банка данных.

2. Снизить расходы на ведение дел за счет автоматизации процессов обработки информации, регламентации и упрощения доступа сотрудников компании к нужной информации.

3. Изменить характер труда сотрудников, избавляя их от выполнения рутинной работы и давая возможность сосредоточиться на профессионально важных обязанностях.

4. Обеспечить надежный учет и контроль поступлений и расходования денежных средств на всех уровнях управления.

5. Руководителям среднего и нижнего звеньев анализировать деятельность своих подразделений и оперативно готовить сводные и аналитические отчеты для руководства и смежных отделов.

6. Повысить эффективность обмена данными между отдельными подразделениями, филиалами и центральным аппаратом. Гарантировать полную безопасность и целостность данных на всех этапах обработки информации.

# **Список использованной литературы**

1. Курсовая работа: Проектирование информационных систем (4.10.2018)| Блог Энди Старикова. [Электронный ресурс]: - Режим доступа: http://caexpert.ru/kursovaya-rabota-proektirovanie-informacionnyx-sistem.html, свободный. - Загл. с экрана (дата обращения 05.12.2018).
2. Информационная система. [Электронный ресурс]: - Режим доступа: https://ru.wikipedia.org/wiki/Информационная\_система, свободный. - Загл. с экрана (дата обращения 05.12.2018).
3. Лекция 6: Диаграммы взаимодействия: крупным планом. [Электронный ресурс]: - Режим доступа: https://www.intuit.ru/studies/professional\_skill\_improvements/1364/courses/229/lecture/5960, свободный. - Загл. с экрана (дата обращения 05.12.2018).
4. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник [Текст]: -М: Финансы и статистика, 2005. -544 с.
5. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем: Учебное пособие [Текст]: -М: Финансы и статистика, 2006. -192 с.
6. В. Л. Бройдо. Вычислительные системы, сети и телекоммуникации: Учебник для вузов: Учебное пособие [Текст]: -СПб.: Питер, 2004. — 703 с.
7. UML. [Электронный ресурс]: - Режим доступа: https://ru.wikipedia.org/wiki/UML, свободный. - Загл. с экрана (дата обращения 05.12.2018).
8. ГОСТ 34.602-89.
9. Заботина Н.Н. Проектирование информационных систем: Учебное пособие / Заботина Н.Н. -Братск: Филиал ГОУВПО «БГУЭП», 2007. - Ч.1 - 146 с.
10. Основы NHibernate. Часть 1. [Электронный ресурс]: - Режим доступа: https://habr.com/post/37984/, свободный. - Загл. с экрана (дата обращения 05.12.2018).