**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**УНИВЕРСИТЕТ ИТМО**

**Факультет информационных технологий и программирования**

**КУРСОВАЯ РАБОТА**

**Тема: Анализ и моделирование программной архитектуры и архитектуры данных программного средства CRM-системы для строительной компании**

**Работу выполнил студент:** *Солоненко Николай Сергеевич* **группы** М3310

(фамилия, имя, отчество) (номер группы)

**Руководитель**  *Маятин Александр Владимирович* \_\_\_

(фамилия, имя, отчество)

**Работа защищена** " " 201 г. с **оценкой** \_\_\_

**Подписи членов комиссии:**  \_

\_

**САНКТ-ПЕТЕРБУРГ**

**2019**

**Содержание**

Теоретический этап 3

1. Анализ и моделирование процессов 4

2. Анализ средств автоматизации процессов 11

3. Синтез определённых уровней архитектуры ИС 18

Заключение 21

Список источников 22

Теоретический этап

**CRM-система** **(Система управления взаимоотношениями с клиентами)** — прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов.

**Live чат** —это гибкий инструмент коммуникации, который используется на своем сайте для онлайн техподдержки, консультации ,получения нужной информации.

**Воронка продаж** — маркетинговая модель, описывающая предполагаемое «путешествие» будущего покупателя от первого знакомства с предложением или товаром до реальной покупки.

**Бекап(Резервное копирование)** — процесс создания копии данных на носителе, предназначенном для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения.

**База данных** — это упорядоченный набор структурированной информации, или данных, которые обычно хранятся в электронном виде в компьютерной системе.

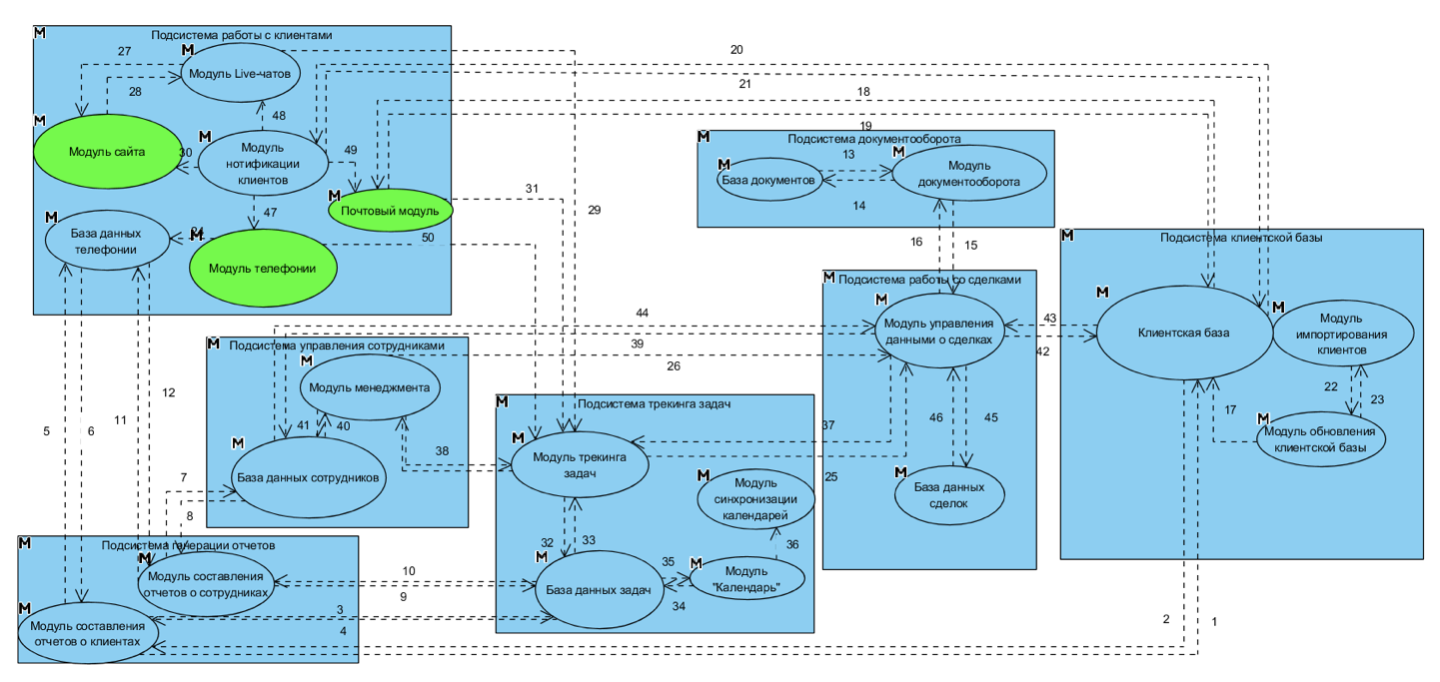
**Диаграмма вариантов использования(use case diagram)** — диаграмма, отражающая отношения между [актерами](https://ru.wikipedia.org/wiki/%25D0%2590%25D0%25BA%25D1%2582%25D0%25BE%25D1%2580_(UML)) и [прецедентами](https://ru.wikipedia.org/wiki/%25D0%259F%25D1%2580%25D0%25B5%25D1%2586%25D0%25B5%25D0%25B4%25D0%25B5%25D0%25BD%25D1%2582_(UML)) и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

**Прецедент** — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних [требований](https://ru.wikipedia.org/wiki/%25D0%25A2%25D1%2580%25D0%25B5%25D0%25B1%25D0%25BE%25D0%25B2%25D0%25B0%25D0%25BD%25D0%25B8%25D1%258F_%25D0%25BA_%25D0%25BF%25D1%2580%25D0%25BE%25D0%25B3%25D1%2580%25D0%25B0%25D0%25BC%25D0%25BC%25D0%25BD%25D0%25BE%25D0%25BC%25D1%2583_%25D0%25BE%25D0%25B1%25D0%25B5%25D1%2581%25D0%25BF%25D0%25B5%25D1%2587%25D0%25B5%25D0%25BD%25D0%25B8%25D1%258E) к системе.

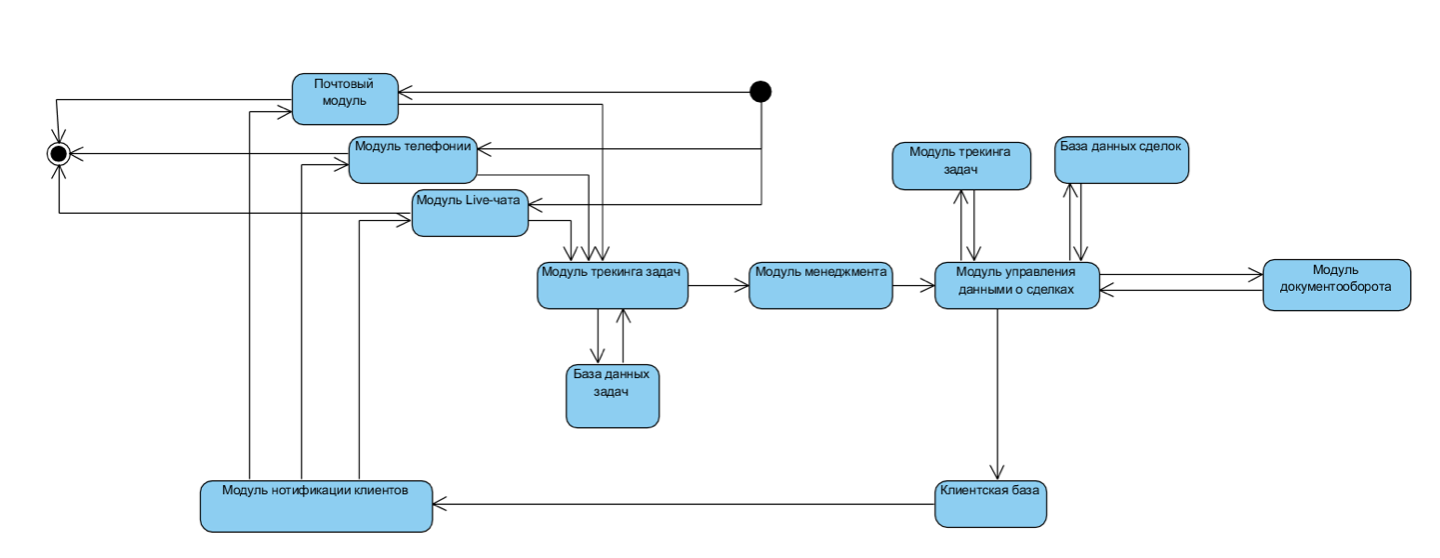
**Класс (class)** – абстрактное описание множества однородных объектов, имеющих одинаковые атрибуты, операции и отношения с объектами других классов. Графически класс в нотации языка UML изображается в виде прямоугольника, который дополнительно может быть разделен горизонтальными линиями на разделы или секции. В этих секциях могут указываться имя класса, атрибуты и операции класса

**Диаграмма классов (class diagram)** –диаграмма языка UML, на которой представлена совокупность декларативных или статических элементов модели, таких как классы с атрибутами и операциями, а также связывающие их отношения

1. Анализ и моделирование процессов

**СRM-система  для строительной компании** - включает в себя все основные принципы работы любой CRM-системы, но с изменениями основанными на особенностях и проблемах строительного бизнеса, таких как : длинный цикл сделки, низкая скорость реакции менеджера из-за рутинных задач, отсутствия современных каналов связи.  
*Рисунок 1.1 Диаграмма модели CRM-системы*

*Рисунок 1.2. Диаграмма вариантов использования*

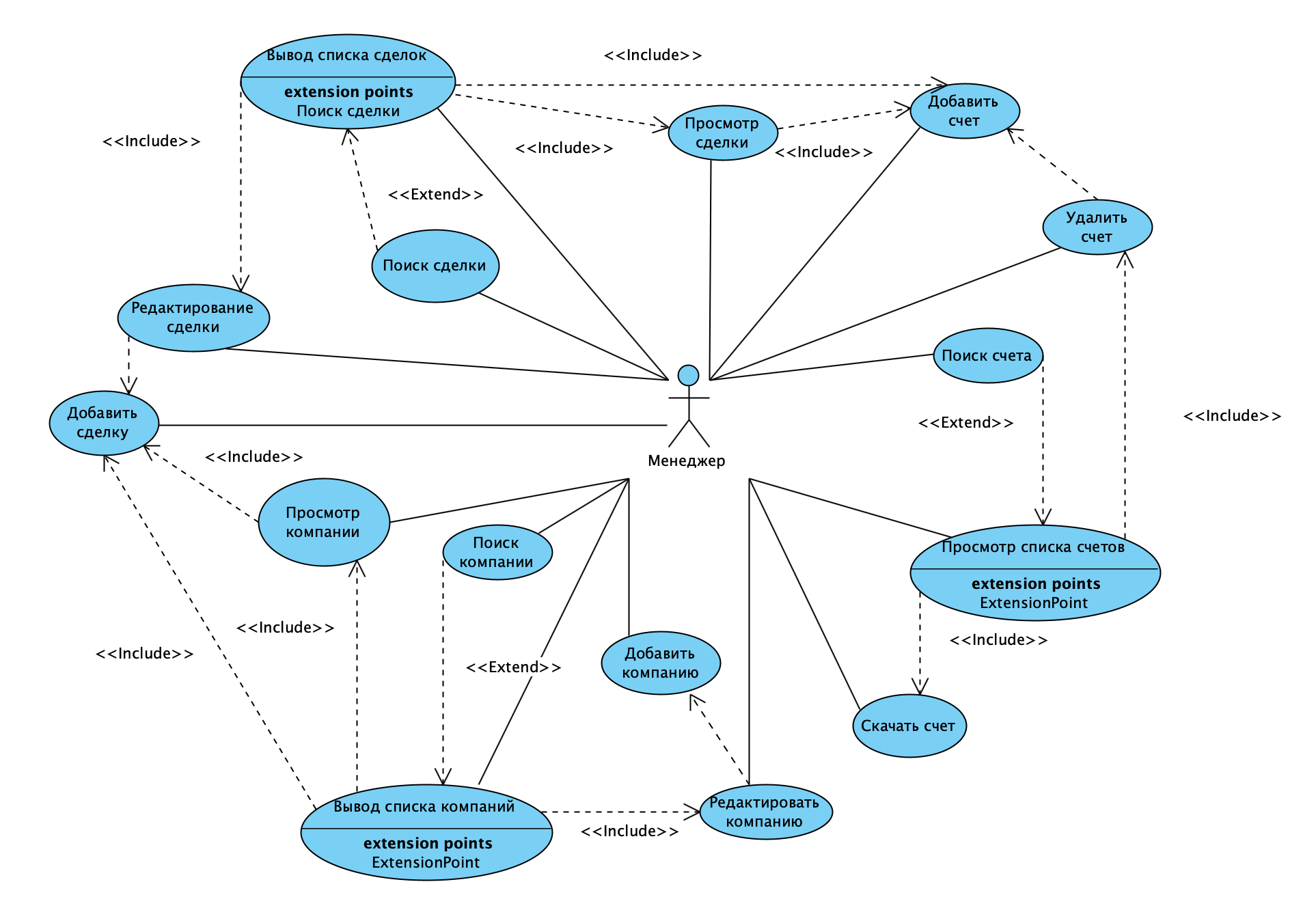


*Рисунок 1.3. Диаграмма «Сценарий обращения клиента»*

|  |  |  |  |
| --- | --- | --- | --- |
| Название элемента | Назначение | Тип информационного процесса | Принадлежность подсистеме |
| Модуль составления отчетов о клиентах | Формирование отчетов об активности клиентов | Обработка | Анализ и отчеты |
| Модуль составления отчетов о сотрудниках | Формирование отчетов о работе сотрудников | Обработка | Анализ и отчеты |
| База документов | Хранение документов | Хранение | Подсистема документооборота |
| Модуль документооборота | Управление документооборотом | Обработка | Подсистема документооборота |
| Клиентская база | Хранение данных о клиентах | Хранение | Подсистема клиентской базы |
| Модуль импортирования клиентов | Загрузка клиентской базы со сторонних ресурсов | Передача | Подсистема клиентской базы |
| Модуль обновление клинтской базы | Обновление информации о клиентах | Обработка | Подсистема клиентской базы |
| База данных телефонии | Хранение данных о звонках | Хранение | Подсистема работы с клиентами |
| Модуль Live-чатов | Система связывания между клиентами и сотрудниками | Передача | Подсистема работы с клиентами |
| Модуль нотификации клиентов | Оповещение клиентов по средствам push-уведомлений | Передача | Подсистема работы с клиентами |
| Модуль сайта | Веб-страница, которая создается внутри CRM, для информирования клиентов | Ввод/ Вывод | Подсистема работы с клиентами |
| Модуль телефонии | Загрузка информации о звонках с различных источников | Ввод/ Вывод | Подсистема работы с клиентами |
| Почтовый модуль | Работа с письмами | Ввод/ Вывод | Подсистема работы с клиентами |
| База данных задач | Хранение информации о событиях | Хранение | Подсистема трекинга задач |
| Модуль синхронизации календарей | Синхронизация задач в Office365, Google Календарь | Передача | Подсистема трекинга задач |
| Модуль трекинга задач | Управление данными о задачах | Обработка | Подсистема трекинга задач |
| Модуль управления данными о сделках | Управление данными о сделках | Обработка | Подсистема работы со сделками |
| Модуль "Календарь" | Представление информации о событиях | Передача | Подсистема трекинга задач |
| База данных сотрудников | Хранение данных о сотрудниках | Хранение | Подсистема управления сотрудниками |
| Модуль менеджмента сотрудниками | Отслеживание активности и управление сотрудниками | Обработка | Подсистема управления сотрудниками |
| База данных сделок | Хранение данных о сделках | Хранение | Подсистема работы со сделками |

*Таблица 1.1 - Описание элементов CRM-системы*

Таблица 1.1 позволяет нам увидеть какую роль выполняют элементы в CRM системе, а так же тип информационного процесса и принадлежность к определённой подсистеме.

Так как выбранная CRM-система ориентирована на строительную компанию, то взаимодействие системы с клиентом необязательно, а значит можно главным действующим лицом обозначить менеджера работающего с заказами. Так же всвязи с особенностями функционала возлагающегося на менеджера строительной компании можно выделить прецеденты данной CRM-системы. В список прецедентов включатся действия касающиеся компаний, оформляющих заказ, счетов для оплаты, и самих сделок.

*Рисунок 1.4 Диаграмма прецедентов системы*

**Функциональные**  **требования**

Главное функциональное требование любой CRM-системы в том числе и для строительной компании это автоматизация взаимодействия с клиентами. Так же выделится и другие функциональные требования:

* управление клиенской базы
* управление задачами, постановка задач, делегирование
* персональное и коллективное планирование
* интеграция с почтовым клиентом
* интеграция с модулем сайта (реализация Live-чата)
* интеграция с телефонией
* управление учётными записями сотрудников
* документооборот
* разделение прав доступа сотрудников
* отчётность (включая воронку продаж)
* база данных
* минимальная аналитика

**Нефункциональные требования**

* гарантия наибольшего времени безотказной работы
* наличия бекапа(резервного копирования)
* гарантия наименьшего времени аварийного восстановления
* импорт данных
* возможность увеличения количества записей в БД
* интерфейс должен быть интуитивно понятен работникам

2. Анализ средств автоматизации процессов

Для формирования базы данных были определены следующие сущности, с которыми предстоит работать: «Компания», «Сделка», «Статус сделки», «Объект», «Шаблон счет –оферты» и «Счет –оферта».

Для каждой сущности создана таблица с соответствующими атрибутами. Связи между таблицами базы данных определяются внешними ключевыми атрибутами. Ниже представлены таблицы сущностей.

Таблица 2.1 –Таблица базы данных сущности «Компания»

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибуты | Тип данных |
| Компания(company) | id(Ключевой атрибут) | Integer |
| Клиент | Text |
| ИНН | Text |
| ОГРН | Text |
| ОКВЭД | Text |
| Банк | Text |
| Номер расчетного счета | Text |
| Номер корреспондентского счета | Text |
| БИК | Text |
| Почтовый адрес | Text |
| Фактический адре | Text |
| E-mail | Text |
| Номер телефона | Text |
| ФИО сокращенное для подписи | Text |
| Серия | Text |
| Номер паспорта | Text |
| Кем выдан | Text |
| Код подразделения | Text |
| Дата выдачи | DataTime |

В таблице 2.1 в качестве атрибутов сущности «Компания» выступают реквизиты и прочие данные о ней.

Таблица 2.2 - Таблица базы данных сущности «Статус сделки».

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибут | Тип данных |
| Статус сделки(status) | ID(Ключевой атрибут) | Integer |
| Название | Text |

В таблице 2.2 сущности «Статус сделки»содержится всего два атрибута–ID и Название статуса.

Таблица 2.3–Таблица базы данных сущности «Объект»

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибут | Тип данных |
| Объект( object) | ID(Ключевой атрибут) | Integer |
| Название | Text |
| Стоймость объекта | Integer |

В таблице 2.3 сущности «Объект» указывается название объекта и его стоимость.

Таблица 2.4 Таблица базы данных сущности «Шаблон счета-оферты»

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибут | Тип данных |
| Шаблон счет-оферты(invoice\_template) | ID(Ключевой атрибут) | Integer |
| Название | Text |
| Файл | Text |

В таблице 2.4 сущности «Счет –оферта» содержатся файлы шаблонов, на основе которых будут формироваться конечные документы.

Таблица 2.5–Таблица базы данных сущности «Счет –оферта»

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибут | Тип данных |
| Счет-оферта(invoice) | ID(Ключевой атрибут) | Integer |
| Номер счета | Text |
| Файл | Text |
| Дата создания | DataTime |
| Шаблон счета(внешний ключевой атрибут) | Integer |
| Привязка к сделке(внешний ключевой атрибут) | Integer |

В таблице 2.5 сущности «Счет –оферта»содержаться внешние ключевые атрибуты, используемые для связи с шаблоном для формирования документа, а также с «Сделкой», для которой формируется счет –оферта.

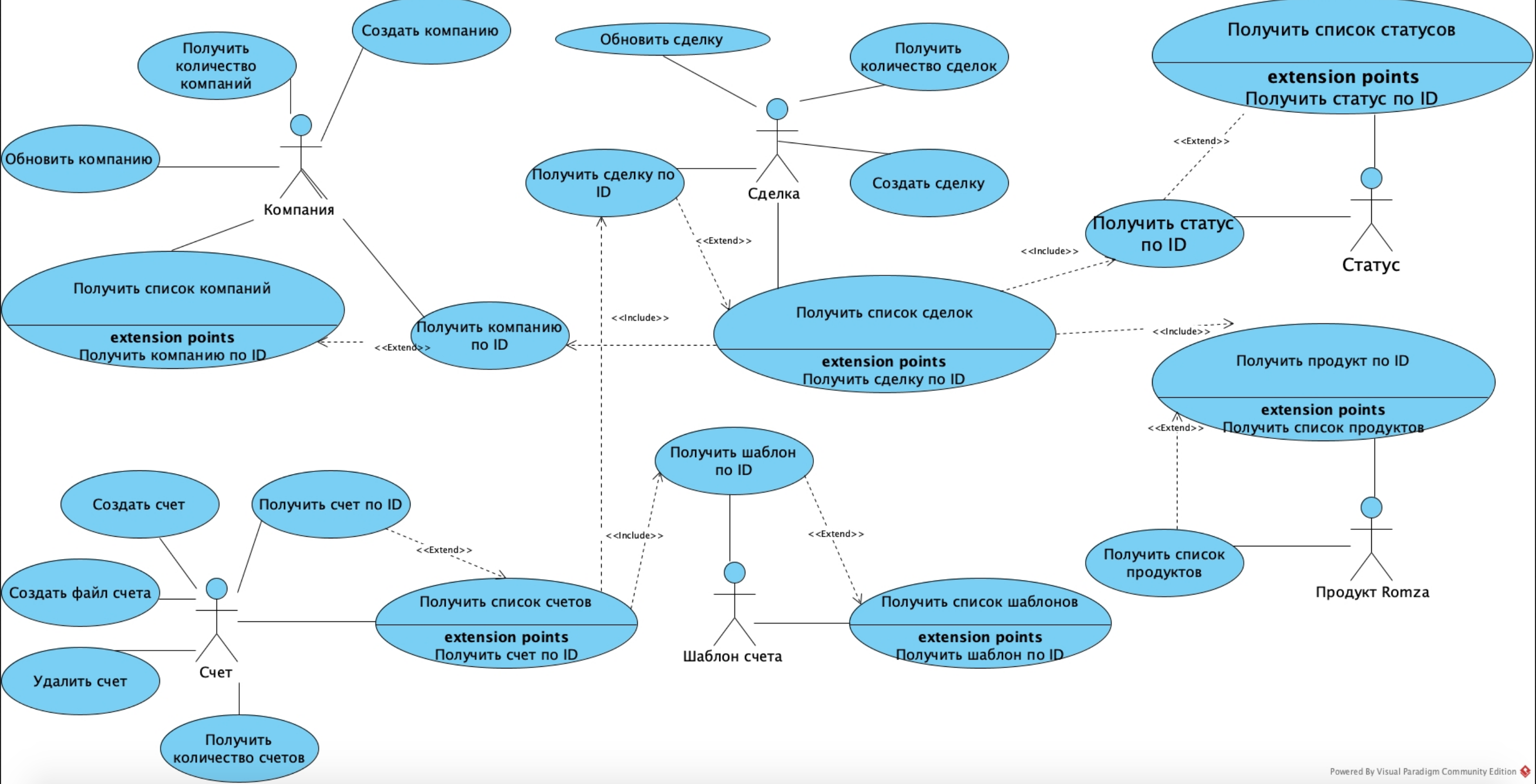
Таблица 2.6 –Таблица базы данных сущности «Сделка»

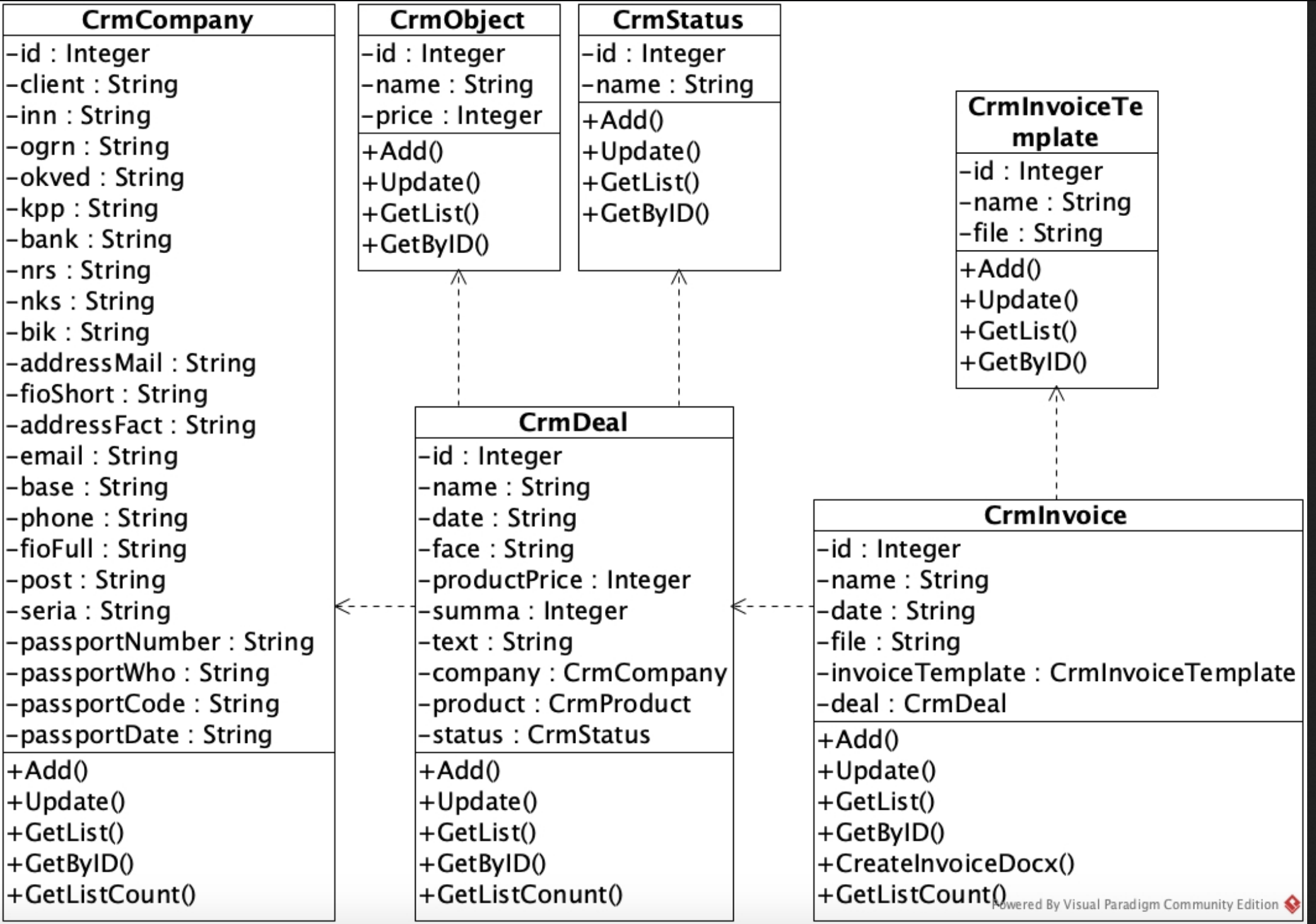
|  |  |  |
| --- | --- | --- |
| Сущность | Атрибут | Тип данных |
| Сделка( deal) | ID(Ключевой атрибут) | Integer |
| Название сделки | Text |
| Ответственный | Text |
| Дата создания | DataTime |
| Комментарий к сделке | Text |
| Цена объекта | Integer |
| Компания(внешний ключевой атрибут) | Integer |
| Объект(внешний ключевой атрибут) | Integer |
| Статус сделки(внешний ключевой атрибут) | Integer |

В таблице 2.6 сущности «Сделка» содержатся внешние ключевые атрибуты, используемые для связи с данными таблиц, представленных выше.

*Рисунок 2.1 - UML Диаграмма прецедентов взаимодействия классов системы CRM*

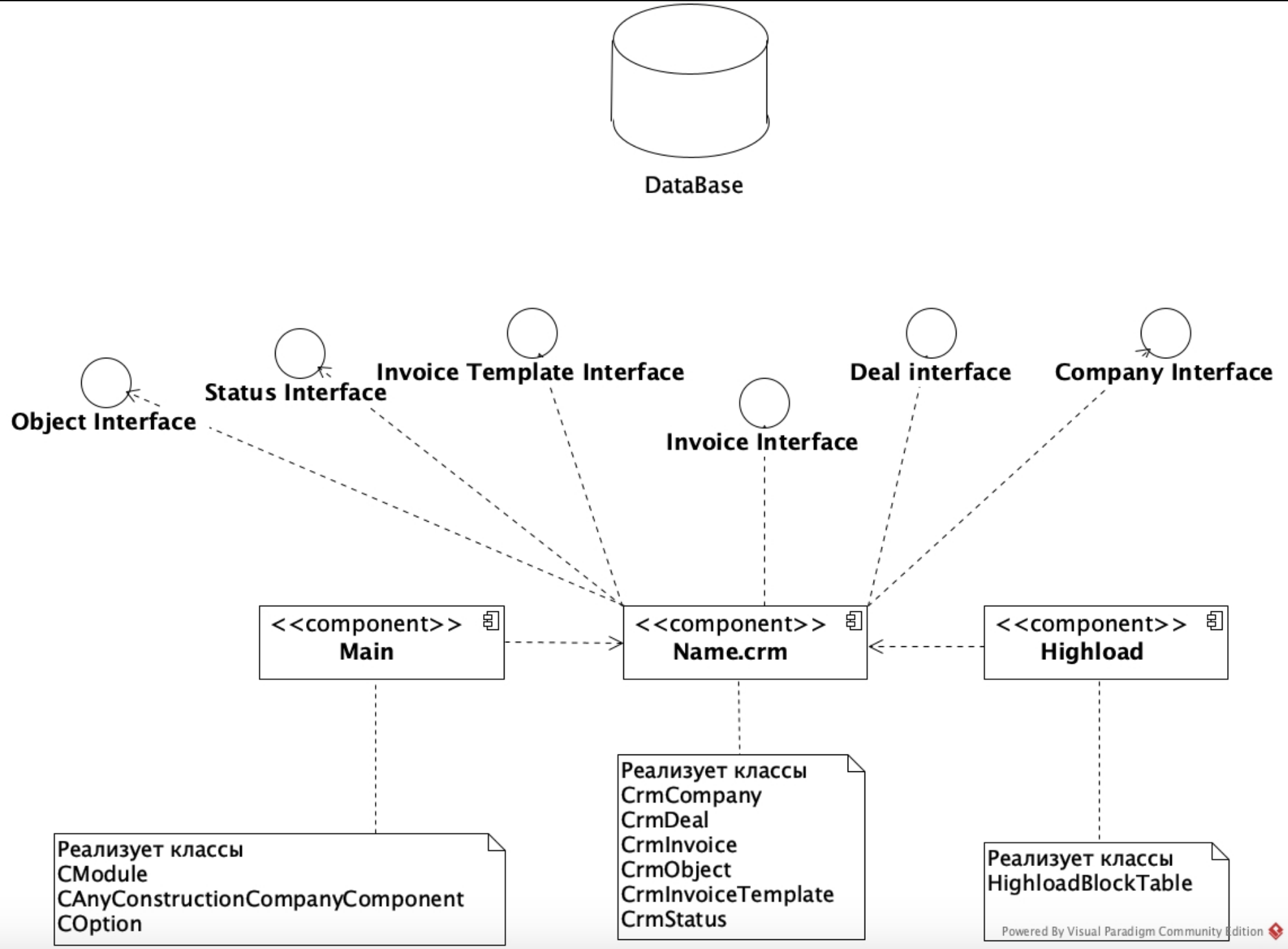
На рисунке 2.1 показана диаграмма Use Case, на которой изображено взаимодействие классов в системе. На этой диаграмме актерами изображены классы, а действия этих актеров (Use Case) изображают методы классов.

По средством методов классы взаимодействуют друг с другом. Совокупность всех методов классов образует интерфейс системы.Некоторые методы классов включают в себя методы других классов. Например, метод «Получить список сделок» включает в себя метод «Получить компанию по ID», указанной в этой сделке, «Получить объект по ID», который приобретается клиентом, «Получить статус по ID». В свою очередь, метод «Получить компанию по ID» является расширением метода «Получить список компаний»



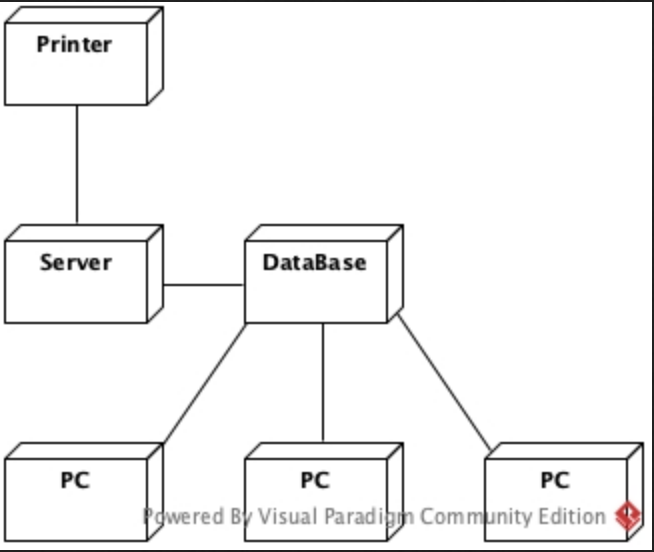
*Рисунок 2.2 - UML Диаграмма классов системы CRM*

На рисунке 2.2 изображена диаграмма классов ClassDiagramUML[18], на которой представлена структура классов из диаграммы Use Case (рисунок 2.1). Из диаграммы видно, что экземпляр класса CrmDeal включает в себя объекты классов CrmCompany, CrmObject, CrmStatus. Все классы содержат общие методы Add(), Update() для создания и редактирования сущностей,

GetList(), GetLictCount() для организации постраничного вывода сущностей и GetByID() для вывода детальной страницы сущности. Класс CrmInvoice имеет уникальный метод CreateInvoiceDocx(), с помощью которого происходит генерация готового документа счет–оферты.

*Рисунок 2.3 - UML Диаграмма компонентов системы CRM*

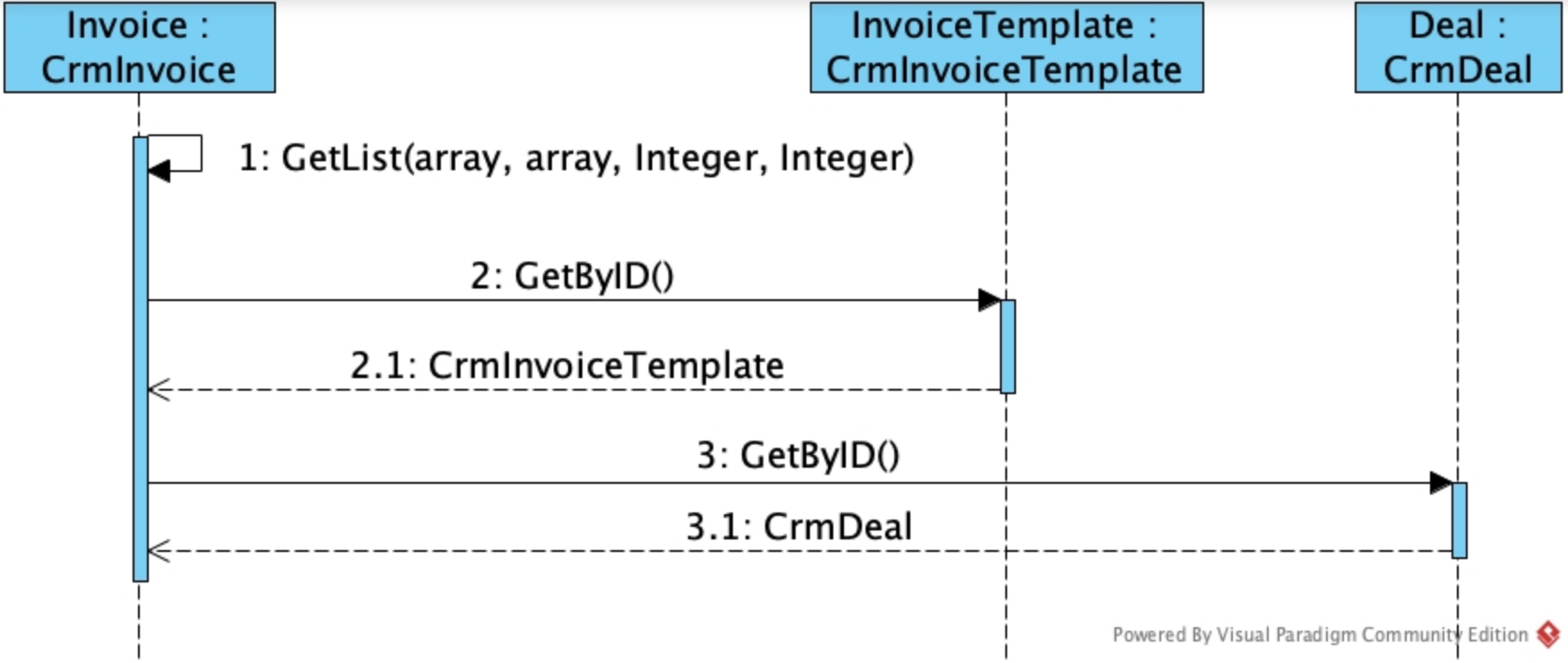
На рисунке 2.3 изображена диаграмма компонентов системы. В роли компонента выступают модули сайта любой строительной компании.Модуль Main реализует классы CModule CAnyConsructionCompanyComponent, на основе которых были созданы модуль и компоненты системы CRM. Класс COption реализует хранение настроек модуля. Модуль Highload реализует класс HighloadBlockTable, через который реализована работа с сущностями системы CRM.Все классы необходимые для работы системы CRM находятся в модуле Name.crm и используют классы модулей Main и Highload.

На диаграмме видно, что взаимодействие приложения с базой данных системы осуществляется с помощью интерфейсов классов. Так как система представляет собой сайт, интерфейсами будут являться веб-формы

*Рисунок 2.3 - UML Диаграмма развертывания системы CRM*

На рисунке 2.3 показана диаграмма развертывания системы CRM. Взаимодействие между персональным компьютером пользователя и базой данных сайта осуществляется через сервер. Для работы сайта необходимо, чтобы на персональном компьютере пользователя был установлен браузер. Отображенные страницы сайта можно распечатать, используя принтер

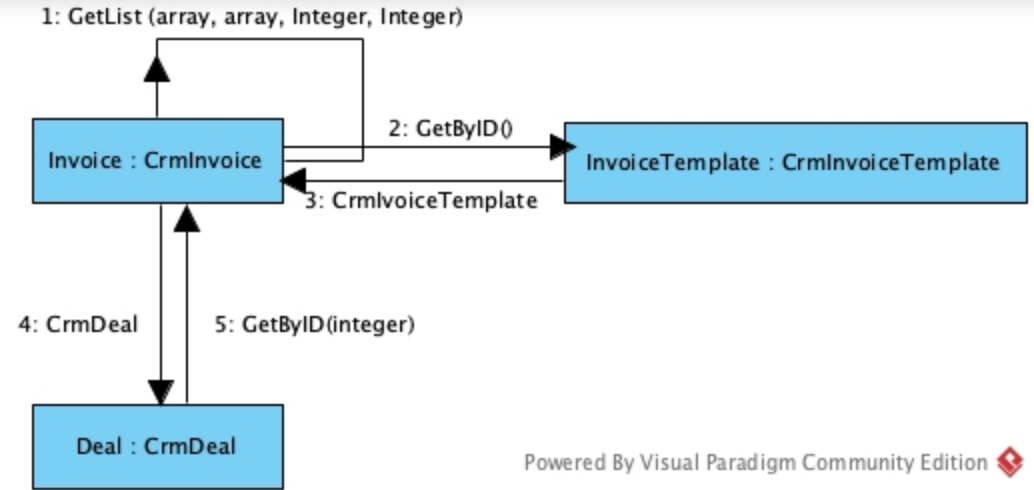
3. Синтез определённых уровней архитектуры ИС

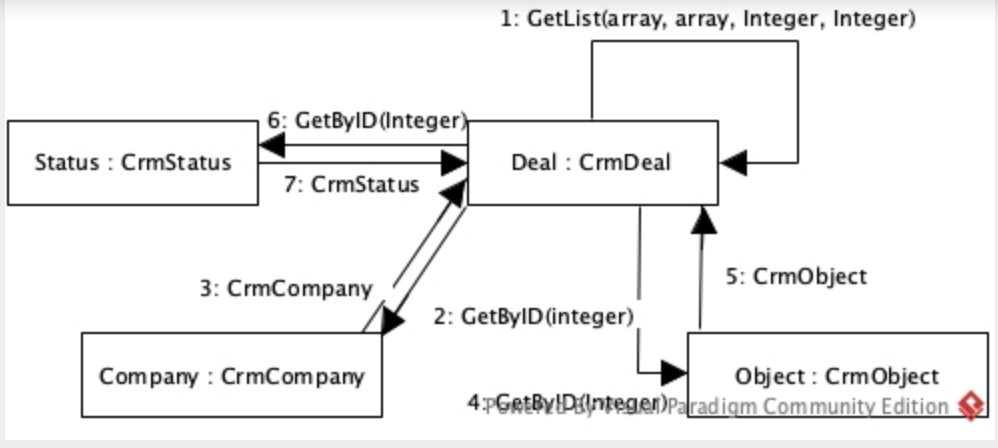
Диаграмма взаимодействия предназначена для моделирования отношений между объектами, в нашем случае данные диаграммы покажут не просто взаимодействие классов системы между собой, а взаимодействие классов во времени, посредством методов. Эти диаграммы покажут некий алгоритм работы системы.На рисунке 3.1 изображена диаграмма последовательности UML, на которой показана последовательность действий во время выполнения метода «Получить список счетов».

*Рисунок 3.1 –UML Диаграмма последовательности системы CRM. Получение списка счетов.*

В процессе получения списка счетов для каждого объекта счета мы запрашиваем объекты классов «Шаблон счета» (CrmInvoiceTemplate) и «Сделка» (CrmDeal).

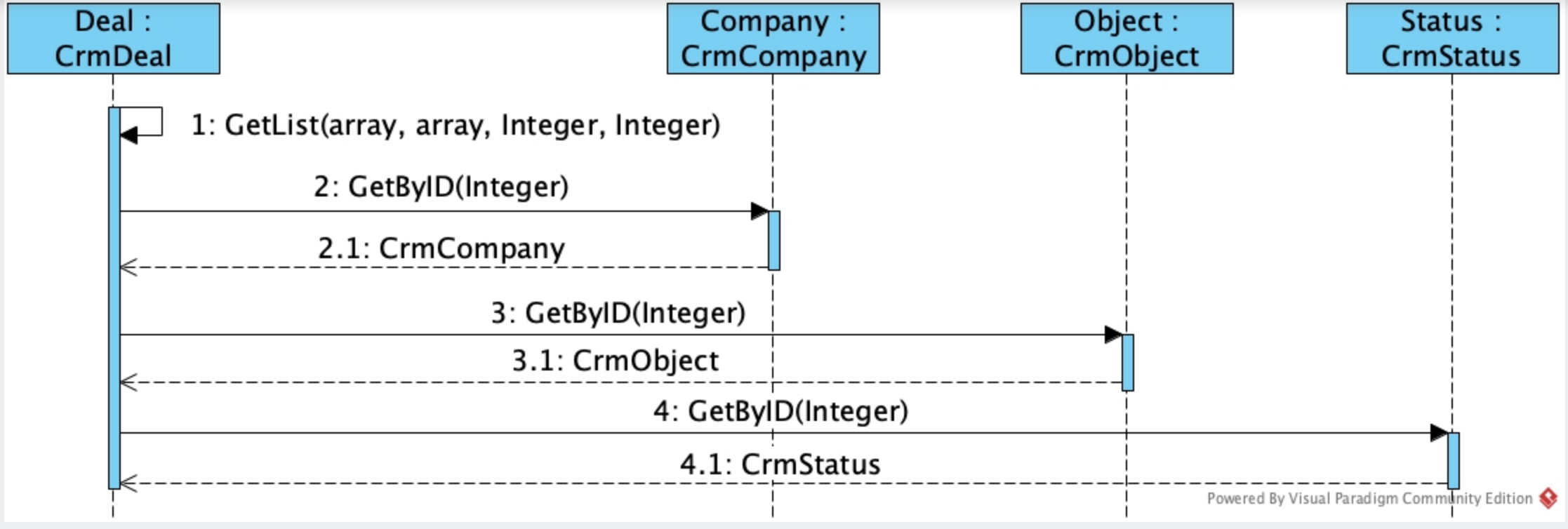
*Рисунок 3.2 –UML Диаграмма кооперации системы CRM. Получение списка счетов*

**

На рисунке 3.2 изображена диаграмма кооперации UML, на которой показано взаимодействие классов на примере метода GetList() класса CrmInvoice. Логика работы метода аналогична описанию диаграммы последовательности (рисунок 3.1).

*Рисунок 3.3 –UML Диаграмма последовательности системы CRM. Получение списка сделок.*

На рисунке 3.3 изображена диаграмма последовательности UML, на которой показана последовательность действий во время выполнения метода «Получить список сделок».



*Рисунок 3.4 –UML Диаграмма кооперации системы CRM. Получение списка сделок*

В процессе получения списка сделок для каждого объекта счета мы запрашиваем объекты классов «Компания» (CrmCompany), «Объект» (CrmObject), и «Статус» (CrmStatus).На рисунке 3.4 изображена диаграмма кооперации UML, на которой показано взаимодействие классов на примере метода GetList() класса CrmDeal. Логика работы метода аналогична описанию диаграммы последовательности (рисунок 3.3)

Заключение

В ходе работы была проанализирована программная архитектуры и архитектура данных программного средства CRM-системы для строительной компании, так же были проанализированы сущности сформированной базы данных

Был сформирован набор диаграмм для визуального представления функциональной и программной архитектуры, а так же их взаимодействия  
Было показано на нескольких примерах взаимодействие функциональной и программной архитектур.  
Так же были выведены и указаны функциональные и нефункциональные требования выбранной CRM-системы.

Список источников

1. Ньюэлл Ф. Почему не работают системы управления отношениями с клиентами (CRM) / Ф. Ньюэлл; пер. с англ. А. Стативка -М.: Издательство Добрая книга, 2004.
2. Трутнев Д. Р. Архитектуры информационных систем. Основы проектирования: Учебное пособие. — СПб.: НИУ ИТМО, 2012.
3. Шуремов Е.Л., Заложнев А.Ю., Руководство по CRM. Путь к совершенствованию менеджмента клиентов / Е.Л. Шуремов –Спб.: Питер, 2009.
4. 5. Моделирование на UML [Электронный ресурс]. — Режим доступа: https://book.uml3.ru/
5. Официальный сайт crm -системе Amocrm [Электронный ресурс]: информация о crm -системе // компания АО «амоЦРМ» –Режим доступа: <http://www.amocrm.ru/>
6. Буч Г., Рамбо Д., Джекобсон А.. Язык UML. Руководство пользователя –The Unified Modeling Language user guide. –2-е изд. / Г. Буч —М., СПб.: ДМК Пресс,Питер, 2004. –432 с