МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

Факультет информационных технологий и программирования

КУРСОВАЯ РАБОТА

Тема: Синтез функциональной и информационной архитектуры информационной системы компании по разработке программного обеспечения «АИС город»

Работу выполнил студент: *Наскальнюк Никита Андреевич* группы *M3304*

(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

[1. Термины и определения 3](#_Toc6409621)

[2. Анализ 5](#_Toc6409622)

[2.1. Описание задачи 5](#_Toc6409623)

[2.2. Информационные объекты 5](#_Toc6409624)

[*Рисунок 1 – Диаграмма объектов* 6](#_Toc6409625)

[2.3. Функциональные требования 6](#_Toc6409626)

[2.4. Нефункциональные требования 7](#_Toc6409627)

[3. Моделирование 8](#_Toc6409628)

[3.1. Интерфейс создания ассоциаций 8](#_Toc6409629)

[*Рисунок 2 – Интерфейс создания ассоциации* 8](#_Toc6409630)

[3.2. Интерфейс просмотра и поиска ассоциаций 9](#_Toc6409631)

[*Рисунок 3 – Интерфейс для просмотра и поиска ассоциации* 9](#_Toc6409632)

[3.3. База Данных 9](#_Toc6409633)

[*Рисунок 4 – База Данных* 10](#_Toc6409634)

[3.4. Веб-сервис 11](#_Toc6409635)

[3.5. Обобщенные этапы работы клиента с приложением 11](#_Toc6409636)

[3.1. Visual Studio SVN 12](#_Toc6409637)

[3.2. 1С: Бухгалтерия 8 13](#_Toc6409638)

[3.3. Redmine 13](#_Toc6409639)

[4. Синтез определенных уровней архитектуры ИС 15](#_Toc6409640)

[5. Список использованных источников 17](#_Toc6409641)

# Термины и определения

*Информационная система (ИС)* – это среда, обеспечивающая целенаправленную деятельность организации и представляющая собой совокупность таких компонентов как информация, процедуры, персонал, аппаратное и программное обеспечение. Все это объединено регулируемыми взаимоотношениями для формирования организации как единого целого и обеспечения её целенаправленной деятельности.

*Корпоративная информационная система (КИС)* - это совокупность информационных систем отдельных подразделений организации, объединенных общим документооборотом, таких, что каждая из систем выполняет часть задач по управлению принятием решений, а все системы вместе обеспечивают функционирование предприятия в соответствии со стандартами качества ИСО 9000.

*Структурная модель* – модель, включающая в себя упорядоченный   
по определенному принципу набор процессов (групп процессов) с указанием основных связей между ними.

*Функциональная архитектура* – набор функций и их подфункций, определяющих преобразования, осуществляемые системой при выполнении своего назначения.

*Информационная архитектура* — сочетание схем организации, предметизации   
и навигации, реализованных в информационной системе.

***Автоматизация –*** одно из направлений научно-технического прогресса, использующее саморегулирующие технические средства и математические методы с целью освобождения человека от участия в процессах получения, преобразования, передачи и использования энергии, материалов, изделий или информации, либо существенного уменьшения степени этого участия или трудоёмкости выполняемых операций.

***Претензия –* обращения клиента к разработчику платежной системы с целью устранить проблемы, возникшие в процессе работы.**

*Норм-час –* единица рабочего времени, за которую сотрудник должен выполнить определенную работу.

*Программное обеспечение* — программа или их множество, используемых для управления компьютером.

*Дедлайн* — крайний или предельный срок, дата или время, к которому должна быть выполнена задача.

*Отдел сопровождения* — совокупность сотрудников, цель которых заключается в поддержке клиентов, а также в незамедлительном устранении неполадок внедренной платежной системы, возникших в процессе работы.

*БД* – база данных.

# Анализ

2.1. Описание задачи

Доктор Грин - лечащий врач мистера Брауна. Мистер Браун страдает от пневмонии, для лечения которой доктор Грин использует традиционную тактику. Доктор Грин находит в статье ссылку на тот факт, что эта тактика в некоторых случаях недостаточно эффективна, и предполагает, что мистер Браун может быть именно таким случаем (из-за других заболеваний в его анамнезе), но это станет ясно только во время лечения. Доктор Грин хочет зафиксировать найденную информацию, чтобы впоследствии иметь возможность, в зависимости от курса лечения пациента, вернуться к ней, изучить более подробно и принять соответствующее персональное решение. Очевидно, что здесь важно зафиксировать не только сами информационные объекты, которые связаны (например, ЭМК Брауна и найденную статью), но и содержание ассоциации в удобной и понятной для врача форме - тегов, которые он может выбрать или даже придумать лично. Обратите внимание, что они могут быть как семантическими («анамнез», «пневмония»), так и несемантическими («посмотрим через неделю», «страница 3 статьи», «хорошая схема» и т. д.).

* 1. Информационные объекты

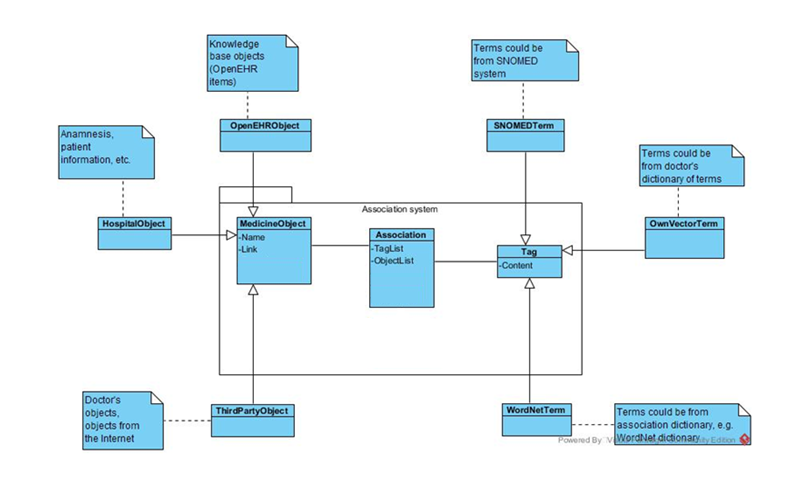
В данном проекте в качестве проблемы рассматривается система по хранению ассоциаций. Основная задача заключается в разработке решения, которое позволит не только взаимодействовать с большим объемом данных, но также предоставит возможность их хранения. В решении присутствуют такие системы автоматизации рабочего процесса, как: пользовательский интерфейс для занесения, поиска, а также просмотра ассоциаций, реляционная база данных для хранения информации и веб-сервис для взаимодействия (данный процесс происходит в качестве запросов) пользователя с данными.

Данный проект предоставляет клиенту:

* Создание и хранение ассоциаций;
* Возможность создания как семантических («анамнез»), так и несемантических («посмотрим через неделю», «страница 3 статьи», «хорошая схема» и т. д.) тегов;
* Дальнейшее взаимодействие с ассоциациями, что предполагает:
  + Их доработку или изменение;
  + Просмотр;
  + Удаление своих ассоциаций из БД.

Для предоставления таких возможностей необходимы следующие информационные объекты:

* Объект «Ассоциаций»;
* Объект «Тегов», которым может быть:
* Термин из «SNOMED»;
* Термин из словаря клиента;
* Термин из Интернета.
* «Медицинский объект», которым может быть:
* Объект «OpenEHR»;
* Анамнез, информация о пациенте и т.п.;
* Любой другой объект (врач, пациент и т.п.).

****

## Рисунок 1 – Диаграмма объектов

## Функциональные требования

Исходя из вышеперечисленного анализа, можно выдвинуть следующие функциональные требования:

1. Наличие графического интерфейса;
2. Клиент должен иметь возможность:
   * Доступа к просмотру всех ассоциаций;
   * Редактирование, а также удаление собственных ассоциаций;
   * Возможность создания ассоциации.
3. Отдел поддержки должен иметь возможность:
   * Доступа к просмотру всех ассоциаций;
   * Создание нового аккаунта;
   * Удаление аккаунта.

## Нефункциональные требования

Исходя из вышеперечисленного анализа, можно выдвинуть следующие нефункциональные требования:

1. Надежность

Так как в системе будет вестись медицинская деятельность, то любая поломка системы подвергает опасности жизни пациентов, также непозволительна утечка информации.

1. Масштабируемость (вертикальная и горизонтальная)

Количество ассоциаций с каждым днем будет увеличиваться, переполненности системы произойти не должно. Может потребоваться перенос данных на сервера большего объема памяти и файлов. Также должно быть доступно повышение производительности и отказоустойчивости системы.

1. Простота

Требования к удобству использования системы (с точки зрения пользователя) и требования к удобству и простоте поддержки.

1. Расширяемость

В связи с появлением новых функциональных требований, должна быть возможность переносимости кода.

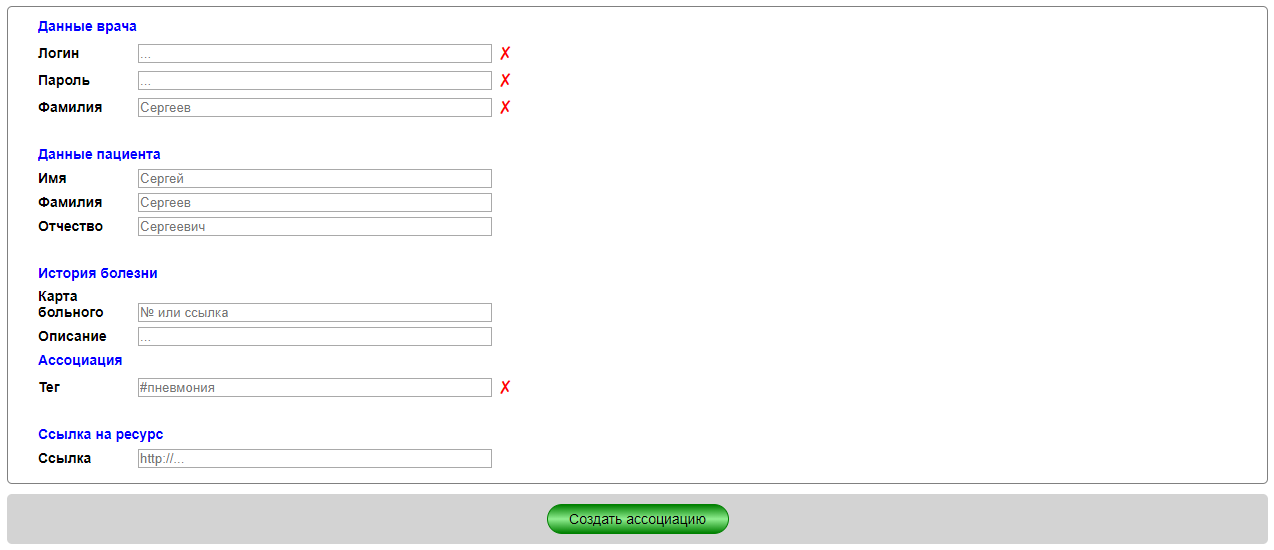
1. Тестирование

Возможность тестирования  системы.

# Моделирование

## 3.1. Интерфейс создания ассоциаций

Главной задачей интерфейса создания ассоциаций является взаимодействие пользователя с объектами, которые хранятся в Базе Данных.

**

## Рисунок 2 – Интерфейс создания ассоциации

Данные врача (логин, пароль и фамилия) обязательны к заполнению, что необходимо для поддержания надежности системы. Логины и пароли создаются автоматически при регистрации нового пользователя. Тип данных: текст.

Далее идут данные пациента, которые необязательны к заполнению, т.к. ассоциация у медицинского работника необязательно должна быть с больным. Тип данных: текст.

История болезни: карта больного и история болезни, которые так же, как и данные пациента необязательны к заполнению. Тип данных для карты больного: текст: номер карты или ссылка на «EHR» (в зависимости от того, как происходит хранение историй болезни в медицинском учреждении). Тип данных для описания: текст или ссылка на ресурс.

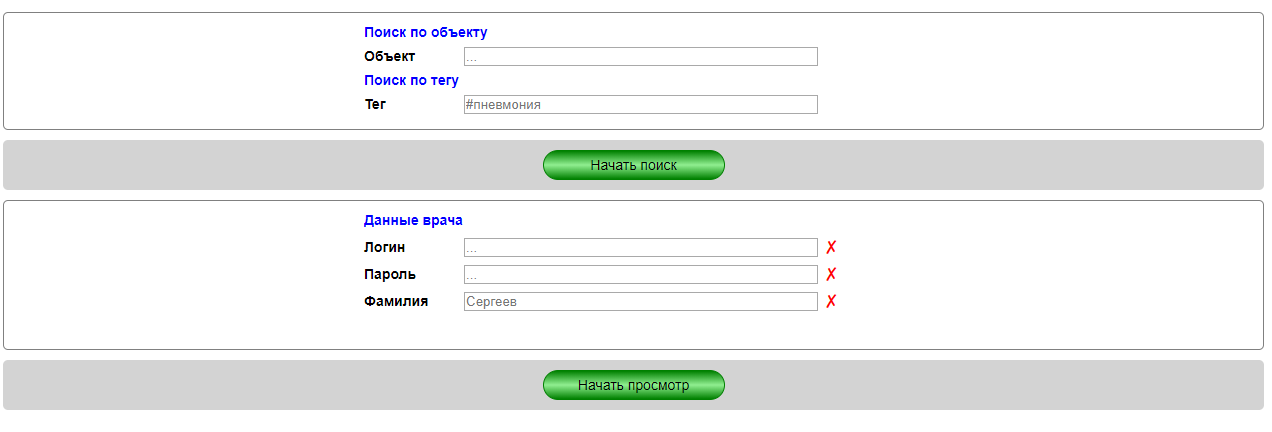
Для создания ассоциации необходимо заполнить хотя бы одно из пяти полей, находящихся в истории болезни и типе данных.

Ассоциация (тег) необходима для понимания и запоминания медицинским работником связей между объектами, которые он проассоциировал. Тип данных: текст.

Ссылка на ресурс необязательна к заполнению, но необходима, если объект ассоциации находится в интернете. Тип данных: ссылка на ресурс.

## 3.2. Интерфейс просмотра и поиска ассоциаций

Главной задачей интерфейса просмотра и поиска ассоциаций является взаимодействие пользователя с объектами: их редактирование, просмотр и удаление своих записей, которые хранятся в Базе Данных.



## Рисунок 3 – Интерфейс для просмотра и поиска ассоциации

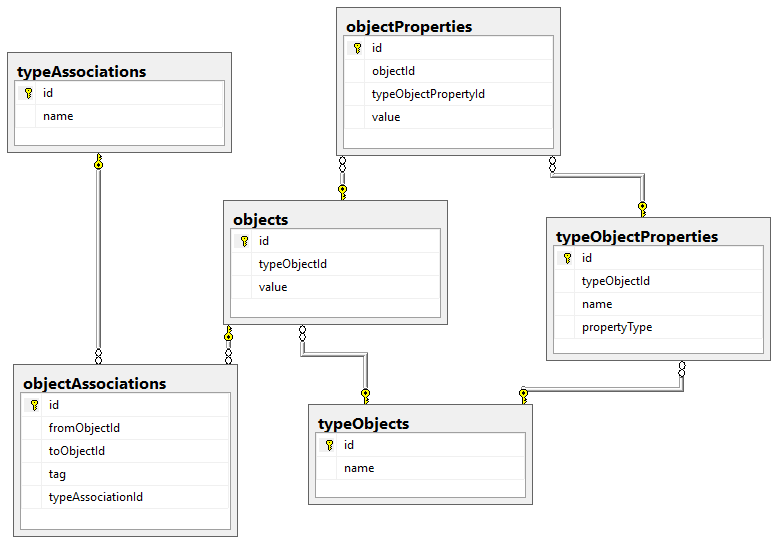
Данные врача (логин, пароль и фамилия) обязательны к заполнению, что необходимо для поддержания надежности системы, так как при просмотре ассоциаций также возможно изменение и удаление своих записей. Тип данных: текст.

Поиск по объекту необходим в тех случаях, когда медицинский работник хочет посмотреть хранящиеся ассоциации с пациентом, историей болезни или же в том случае, если он забыл тег. Тип данных: текст.

Поиск по тегу позволяет медицинскому работнику посмотреть связи между объектами, которые он проассоциировал. Тип данных: текст.

Для поиска ассоциации необходимо заполнить одно из двух полей.

## 3.3. База Данных

****

## Рисунок 4 – База Данных

РБД состоит из 6 таблиц:

* typeObjects, в которой хранится тип объекта, например:
  + врач;
  + пациент;
  + история болезни;
  + ссылка;
  + тег.

Каждый объект БД имеет свой тип, что упрощает поиск среди объектов и позволяет медицинскому работнику выполнять поиск по тегу. Каждый тип имеет свой уникальный id, что упрощает взаимодействие и уменьшает объем хранимой информации. Связь: один ко многим.

* Таблица typeObjectProperty, в которой хранится свойства типа объекта, например:
  + фамилия;
  + имя;
  + отчество;
  + специализация;
  + дата рождения и т.д.

Каждый объект БД имеет свой тип, а также его свойство. Благодаря этому, при общем поиске становится возможным найти все объекты, связанные с данным свойством. Пример: по фамилии врача медицинский работник может просмотреть всех пациентов, которых он лечил. Каждое свойство имеет свой уникальный id, что упрощает взаимодействие и уменьшает объем хранимой информации. Связь: многие ко многим.

* Objects, основная таблица в которой хранятся объекты, которые медицинский работник ассоциирует:
  + user 1;
  + пациент 1;
  + пневмония - интересный случай;
  + [http://patient.info/health и т.д](http://patient.info/health%20и%20т.д).

Каждое объект имеет свой уникальный id, что упрощает взаимодействие и уменьшает объем хранимой информации. Связь: многие ко многим.

* Таблица objectProperties, в которой хранятся свойства объекта:
  + Грин;
  + Иванов;
  + Диплом доктора общей практики;
  + Иван и т.д.

Каждое свойство относится к одному или нескольким объектам, а также имеет свой уникальный id, что упрощает взаимодействие и уменьшает объем хранимой информации. Связь: многие ко многим.

* Таблица ObjectAssociations, также является одной из главных таблиц, в которой хранятся объекты, между которыми происходит ассоциация. Связь: многие ко многим.
* Таблица typeAssociations, в которой хранятся тип ассоциаций:
  + Лечение;
  + Карта больного;
  + Пациент;
  + Интернет ресурс;
  + Интересный случай и т.д.

Каждый тип относится к одному или нескольким объектам, а также имеет свой уникальный id, что упрощает взаимодействие и уменьшает объем хранимой информации. Связь: многие ко многим.

## 3.4. Веб-сервис

Входящим запросом будет набор параметров

## 3.5. Обобщенные этапы работы клиента с приложением

## 

Создание ассоциации:

1. У медицинского работника появляется;
2. Отдел бухгалтерии готовит договор на выполнение работ в двух экземплярах;
3. Отдел бухгалтерии передает договор директору для подписания   
   их клиентом;
4. При успешном окончании переговоров директор подписывает договор, оставляя один экземпляр заказчику, и передает всю информацию руководителю отдела разработки (контактные данные клиента, техническое задание);
5. Директор отправляет клиенту письмо о начале работы, фиксирует данные по встрече и телефонному разговору, сообщает данные руководителя, который будет вести проект;
6. Руководитель распределяет между работниками необходимые задания, для каждого определяет часы на его выполнение и устанавливает дедлайн;
7. По мере выполнения проекта руководитель отдела связывается с клиентом   
   для уточнения возникших вопросов, утверждения проделанной работы или письменного согласования (если в это есть необходимость);
8. Если разработчик не справляется с поставленной задачей, то руководитель предпринимает действия, чтобы решить поставленную задачу (информация о не справившемся сотруднике передается директору);
9. После завершения проекта руководитель отдела передает все данные клиенту и сообщает главному бухгалтеру о завершении проекта;
10. Отдел бухгалтерии подготавливает акты;
11. Главный бухгалтер получает оплату от клиента и подписывает у него акты;
12. В дальнейшем внедренной платежной системой занимается отдел сопровождения.

## 3.1. Visual Studio SVN

Microsoft Visual Studio — это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию. Эти инструменты предназначены для максимально эффективной совместной работы; все они доступны в интегрированной среде разработки Visual Studio.

SVN – дополнение для Microsoft Visual Studio, которое позволяет использовать систему контроля версий Subversion в данной среде разработки.

Возможности SVN:

* обзор всех изменений в реальном времени;
* доступ к большинству команд в Subversion;
* простой мастер для объединения изменений;
* полная поддержка Visual Studio 2005, 2008, 2010, 2012, 2013 и 2015;
* обозреватель хранилища. Доступные действия в хранилище: копирование, переименование;
* автоматическая проверка новых версий;
* интуитивно понятный мастер разрешения конфликтов;
* автоматический импорт новых проектов.

## 1С: Бухгалтерия 8

«1С: Бухгалтерия 8» — это профессиональная бухгалтерская программа, с помощью которой можно вести учет, готовить и сдавать обязательную отчетность. Продукт сделан под законы Российской Федерации, в частности, под законы о налогах и отчётности. В состав конфигурации включен план счетов бухгалтерского учёта, соответствующий Приказу Минфина РФ «Об утверждении плана счетов бухгалтерского учёта с которого финансово-хозяйственной деятельности организаций и инструкции по его применению» от 31 октября 2000 г. № 94н (в редакции Приказа Минфина РФ от 07.05.2003 № 38н). Состав счетов, организация аналитического, валютного, количественного учёта на счетах соответствуют требованиям законодательства по ведению бухгалтерского учёта и отражению данных в отчётности. При необходимости пользователи могут самостоятельно создавать дополнительные счета и разрезы аналитического учёта.

В «1С: Бухгалтерия 8» автоматизированы многие необходимые вещи:

* расчёт НДС;
* расчёт НДФЛ;
* определение доходов и расходов упрощенной системы налогообложения;
* налоги в Пенсионный Фонд Российской Федерации;
* создание бухгалтерских отчётов;
* создание расчетных счетов и актов.

## Redmine

«Redmine» является прекрасным инструментом для обеспечения эффективной работы бизнеса.

Этот сервис включает в себя большое количество услуг, в том числе:

1. Функционал для рабочего процесса:

* ведение нескольких проектов;
* гибкая система доступа, основанная на ролях;
* система отслеживания ошибок;
* диаграммы Ганта и календарь;
* ведение новостей проекта, документов и управление файлами;
* оповещение об изменениях с помощью RSS-потоков и электронной почты;
* форумы для каждого проекта;
* учёт временных затрат;
* настраиваемые произвольные поля для инцидентов, временных затрат, проектов и пользователей;
* лёгкая интеграция с системами управления версиями
* создание записей об ошибках на основе полученных писем;
* многоязычный интерфейс (в том числе русский);
* поддержка СУБД MySQL, Microsoft SQL Server, PostgreSQL, SQLite, Oracle.

1. Функционал для работы с клиентами:

* возможность настройки робота для отправки писем и SMS клиентам;
* автоматический анализ продаж, отчеты по работе менеджеров.

Redmine — это открытое серверное веб-приложение для управления проектами и задачами (в том числе для отслеживания ошибок). Имеется возможность техподдержки сотрудников.

# Синтез определенных уровней архитектуры ИС

Ниже представлены функциональные и информационные возможности архитектуры Redmine.

Таблица 1 – Описание функциональной и информационной архитектуры

|  |  |  |
| --- | --- | --- |
| **Операция** | **Информационные объекты** | **Функциональный компонент** |
| Создание проекта | БД (проекты)  БД (клиенты)  Договор (для заключения сделки)  Документ (подробное описание заказа) | Создать проект |
| Составление заданий | БД (проекты)  Документ (подробное описание заказа) | Создать задачу |
| Распределение заданий | БД (проекты)  БД (сотрудники)  Документ (подробное описание задания) | Создать работу |
| Доступ к поставленным задачам | БД (проекты)  БД (сотрудники)  Документ (подробное описание задания) | Задачи и проекты (для каждого сотрудника индивидуально) |
| Редактирование задач | БД (сотрудники)  Документ (задание) | Редактирование задач до или после их распределения |
| Добавить комментарий | БД (проекты)  Документ (задание) | Добавить комментарий |
| Формирование событий | БД (сотрудники)  Документ (событие) | Создать событие на определенную дату |
| Анализ работы | БД (проекты)  БД (сотрудники)  БД (отчеты) | Анализ выполненной работы |
| Общение с клиентами | БД (клиенты) | Общение с клиентами (обычно происходит по Skype или телефонному разговору) |
| Общение между сотрудниками | БД (сотрудники) | Общение между сотрудниками возможно в Redmine, но по факту это происходит в соц. сетях |
| Обсуждение задач | БД (сотрудники)  Документ (задание) | Обсуждение задач происходит в Redmine, либо в соц. сетях |
| Просмотр всех сотрудников | БД (сотрудники) | Взаимодействия с сотрудниками |
| Доступ к БД | БД | Взаимодействие с БД |
| Создание профиля | Документ (сотрудники) | Регистрация в системе |
| Поиск необходимой информации | Открытые ресурсы | Поиск необходимой информации |

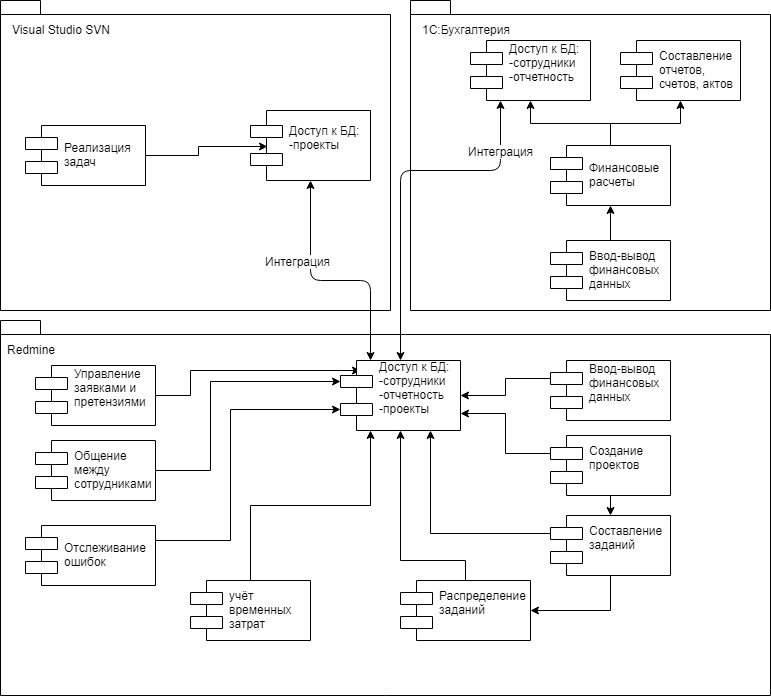


Рисунок 5 – Функциональная архитектура

# Список использованных источников

1. В. Репин. Бизнес-процессы. Моделирование, внедрение, управление. Издательство: Манн, Иванов, Фербер, год издания – 2012, с.502, ISBN 978-5-91657-521-7, 978-5-91657-907-9
2. «АИС город» [Электронный ресурс]. Режим доступа:  http://www.aisgorod.ru/. (Дата обращения 14.12.2018).
3. «Redmine» [Электронный ресурс]. Режим доступа: https://www.easyredmine.com/. (Дата обращения 14.12.2018).
4. «1С – Бухгалтерия 8» [Электронный ресурс]. Режим доступа: http://v8.1c.ru/buhv8/. (Дата обращения 14.12.2018).