|  |  |  |
| --- | --- | --- |
|  | ***Федеральное агентство по рыболовству***  ***Федеральное государственное бюджетное образовательное учреждение***  ***высшего профессионального образования***  ***«Астраханский государственный технический университет»***  Разработка и предоставление образовательных услуг в области среднего профессионального, высшего, дополнительного, дополнительного профессионального образования, международного бизнес-образования; воспитательная работа, научно-исследовательская и инновационная деятельность сертифицированы DQS и ГОСТ Р по ISO 9001:2008 | |
| Институт Информационных технологий и коммуникаций  Направление подготовки09.04.01 Информатика и вычислительная техника  Магистерская программа  Информационное и программное обеспечение автоматизированных систем  Кафедра Автоматизированные системы обработки информации и управления  **КУРСОВОЙ ПРОЕКТ** | | |
| **Анализ данных информационной системы автоматизации процесса взаимодействия пациента и врача**  по дисциплине «Анализ данных и процессов» | | |
| Допущен к защите  «\_\_» \_\_\_\_\_\_\_\_2016г.  Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Оценка, полученная на защите «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_» | | Проект выполнен  студентом группы ДИНРМ-21 Смирнов. К.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Руководитель работы доцент Морозов А.В. |
| Члены комиссии:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Белов С.В. | |  |

Астрахань – 2016

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ПО РЫБОЛОВСТВУ**

**АСТРАХАНСКИЙ ГОСУДАРСТВЕННЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

|  |  |
| --- | --- |
| **УТВЕРЖДАЮ** | Кафедра «Автоматизированные системы  обработки информации и управления» |
| Заведующий кафедрой  к.т.н., доцент  С.В. Белов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г. |

**ЗАДАНИЕ**

**на выполнение курсового проекта**

|  |  |  |  |
| --- | --- | --- | --- |
| Студентка | ***Смирнов Кирилл Владимирович*** | | |
| Группа | ***ДИНРМ-21*** | | |
| Дисциплина | | ***Анализ данных и процессов*** | |
| Тема курсового проекта | | | ***Анализ данных информационной системы*** |
| ***автоматизации процесса взаимодействия пациента и врача*** | | | |

Дата получения задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016г.  
Срок представления студентом КП на кафедру «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016г.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Руководитель | | ***Доцент*** |  | ***Морозов А.В.*** |
|  | | должность, степень, звание | подпись | ФИО |
|  | |  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016г. | |
| Студент | |  |  | ***Смирнов К.В*** |
|  | |  | подпись | ФИО |
|  | |  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016г. | |
| **Задачи** | | | | |
| 1 | Изучить предметную область. | | | |
| 2 | Смоделировать пользовательскую систему выбора симптомов. | | | |
| 3 | Разработать систему выбора симптомов. | | | |
| 4 | Описать математическую модель нейронной сети | | | |
| 5 | Реализовать алгоритм нейронной сети | | | |
| 6 | Изучить предметную область. | | | |
| **Список рекомендуемой литературы** | | | | |
| 1 | Хайкин.С. Нейронные сети: полный курс, 2-е издание. 2011 | | | |
| 2 | Лешек А., Анализ и проектирование информационных систем с помощью UML, 2016 | | | |

|  |  |
| --- | --- |
| **УТВЕРЖДАЮ** | К заданию  на курсовой проектпо дисциплине  «Анализ данных и процессов» |
| Заведующий кафедрой  к.т.н., доцент  С.В. Белов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г. |

**КАЛЕНДАРНЫЙ ГРАФИК**

курсового проектирования

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  п/п | Разделы, темы  и их содержание,  графический материал | По плану | | Фактически | | Отметка руководи-теля о вы-полнении |
| Дата | Объем в % | Дата | Объем в % |
| 1 | Выбор темы | 05.09.2016 | 1 |  |  |  |
| 2 | Техническое задание | 10.09.2016 | 5 |  |  |  |
| 3 | Разработка модели,  проектирование системы   * *введение,* * *технический проект,* * *программа и методика испытаний,* * *литература* | 15.10.2016 | 30 |  |  |  |
| 4 | Программная реализация системы   * *работающая программа,* * *рабочий проект* * *скорректированное техническое задание (при необходимости)* | 05.11.2016 | 60 |  |  |  |
| 5 | Тестирование и отладка системы, эксперименты   * *работающая программа с внесёнными изменениями,*   ***окончательные тексты***   * *введение,* * *технический проект,* * *рабочий проект* * *программа и методика испытаний* | 15.11.2016 | 80 |  |  |  |
| 6 | Компоновка текста  Подготовка презентации и доклада   * *пояснительная записка* * *презентация* | 21.11.2016 | 100 |  |  |  |
| 7 | Защита курсового проекта | 21.11.2016 | 100 |  |  |  |

Руководитель курсового проекта доцент Морозов А.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ученая степень, звание, фамилия, подпись)

Студент Смирнов К.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы, подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc468235534)

[1 ТЕХНИЧЕСКИЙ ПРОЕКТ 6](#_Toc468235535)

[1.1 Анализ предметной области 6](#_Toc468235536)

[1.1.1 Процесс постановки диагноза 6](#_Toc468235537)

[1.2 Анализ данных при помощи OLAP 12](#_Toc468235538)

[1.3 Технология обработки информации 15](#_Toc468235539)

[1.4 Входные данные облачного решения 16](#_Toc468235540)

[1.5 Выходные документы облачного решения 18](#_Toc468235541)

[1.6 Входные данные аналитического клиентского приложения 18](#_Toc468235542)

[1.7 Выходные данные аналитического клиентского приложения 18](#_Toc468235543)

[1.8 Системные требования 18](#_Toc468235544)

[1.9 Требования к программному обеспечению 18](#_Toc468235545)

[1.10 Требования к техническому обеспечению 19](#_Toc468235546)

[1.11 Аналоги системы 19](#_Toc468235547)

[2 РАБОЧИЙ ПРОЕКТ 20](#_Toc468235548)

[2.1 Общие сведения о работе системы 20](#_Toc468235550)

[2.2 Функциональное назначение программного продукта 20](#_Toc468235551)

[2.3 Описание физической архитектуры системы 20](#_Toc468235552)

[2.4 Разработка структуры базы данных 20](#_Toc468235553)

[2.5 Инсталляция и выполнение программного продукта 28](#_Toc468235554)

[2.6 Общий алгоритм работы программного продукта 28](#_Toc468235555)

[2.7 Разработанные меню и интерфейсы 29](#_Toc468235556)

[2.7.1 Интерфейс "Постановка диагноза для мужчины" 29](#_Toc468235557)

[2.7.2 Интерфейс "Постановка диагноза для мужчины" 29](#_Toc468235558)

[2.7.3 Интерфейс "Просмотр диагноза" 30](#_Toc468235559)

[2.7.4 Интерфейс "Административная панель" 30](#_Toc468235560)

[2.7.5 Интерфейс "Создание изменения систем" 31](#_Toc468235561)

[2.7.6 Интерфейс "Создание изменения частей тела" 31](#_Toc468235562)

[2.7.7 Интерфейс "Создание изменение симптомов" 32](#_Toc468235563)

[2.7.8 Интерфейс "Создание изменение диагнозов" 32](#_Toc468235564)

[2.7.9 Интерфейс "Обучение диагноза" 33](#_Toc468235565)

[2.7.10 Сообщения системы 33](#_Toc468235566)

[2.7.11 Построение куба и анализ данных на клиенте 34](#_Toc468235567)

[2.7.12 Подготовка базы данных AnalysisServices на сервере 36](#_Toc468235568)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 43](#_Toc468235569)

[ПРИЛОЖЕНИЕ 1 44](#_Toc468235570)

[ПРИЛОЖЕНИЕ 2 45](#_Toc468235571)

[ПРИЛОЖЕНИЕ 3 46](#_Toc468235572)

[ПРИЛОЖЕНИЕ 4 47](#_Toc468235573)

# ВВЕДЕНИЕ

Основным аспектом работы практически каждого врача, является диагностика заболеваний у пациентов.

Процесс диагностики заболевания у пациента крайне трудоемкий процесс, требующий большого багажа знаний и опыта врача и скорее всего не одного, т.к. в практике обычно постановкой диагноза занимается несколько врачей. Сначала пациента осматривает врач общей практики для выявления основных симптомов и направления пациента более узкоспециализированному специалисту.

Т.к. процесс диагностики довольно трудоемок, возникает вопрос, о его автоматизации в целях упрощения работы врача, что позволяло бы освободить некоторое время врача, что решало бы большое количество проблем в частности очереди в больницах и т.д. Процесс диагностики заболевания у пациента, совсем не тривиален и нет четкого алгоритма действий для постановки четкого диагноза, автоматизированной системой.

Т.к. нет четко сформулированного описания для работы системы постановки диагноза, по котором можно было бы сформулировать алгоритм постановки диагноза врачом, встает вопрос о поиске других решений, которые смогли бы работать без описания четкой логики. Под заданную задачу подходят алгоритмы машинного обучения.

Целью курсовой работы является описание, моделирование и разработка пользовательской системы для набора симптомов и для постановки диагноза на основе алгоритмов машинного обучения, а также анализ предметной области, анализ существующих методов и алгоритмов машинного обучения и применения их в постановке диагноза.

Назначение программы: Программное обеспечение предназначено для автоматизации и облегчения процесса деятельности врачей и простой коммуникации врачей и пациентов.

# 1 ТЕХНИЧЕСКИЙ ПРОЕКТ

* 1. **Анализ предметной области**
     1. **Процесс постановки диагноза**

Диагностика болезней  - это процесс определения возможного заболевания (диагностическая процедура) и записи мнения возникшего в результате этого процесса (диагностическое мнение). Основным компонентом диагностики является процедура посещения врача. Диагностическая процедура включает в себя разные тесты с помощью которых врач и определяет наличие или отсутствие у болезни у пациента.

Процесс диагностики заболевания у пациента крайне трудоемкий процесс, требующий большого багажа знаний и опыта врача и скорее всего не одного, т.к. в практике обычно постановкой диагноза занимается несколько врачей. Сначала пациента осматривает врач общей практики для выявления основных симптомов и направления пациента более узкоспециализированному специалисту.

Благодаря анализу опроса врачей о процессе диагностики заболеваний, были выделены основные аспекты диагностирования:

* оценка жалоб и симптомов;
* постановка предварительного диагноза;
* дифференциальная диагностика;
* постановка клинического диагноза.

Первое, что врач должен обнаружить при лечении пациента, это является ли его признаки физиологическими или патологическими. То есть врач должен определить, являются ли предполагаемые признаки болезни нормальными или связанными с болезнью. Врач определяет болезнь на основании нескольких входных параметров:

* История болезни пациента;
* Пол;
* Вес;
* Возраст;
* Раса;
* Основные жалобы;
* История настоящего заболевания (ИНЗ);
* Обзор систем;
* Осмотр;
* Дополнительные тесты.

В некотором смысле, история пациента является наиболее важной частью процесса установления диагноза. История пациента включает в себя информацию о пациенте исходя из того, что пациент говорит врачу. Одно недавнее исследование показало, что среднее количество времени, которое врач слушает пациента без прерывания составляет 17 секунд. Это, очевидно, очень мало времени. Врачи должны провести время, слушая своих пациентов, так как многие тонкости и другие детали имеют решающее значение для подбора адекватной базы информации, чтобы начать установление дифференциального диагноза.

Возраст, пол, раса, вес имеют очень важное значение при постановке диагноза. Зачастую молодые специалисты не обращают внимание на данные параметры. Возраст и пол пациента важны, потому что определенные заболевания имеют склонность к диапазону от возраста и пола. Болезнь рассеянный склероз, например, как правило, поражает женщин чаще, чем мужчины, и, как правило в возрасте от 20 до 40 лет. Раса может быть также связана с распространенностью некоторых заболеваний. Афроамериканцы более склонны к развитию серповидно-клеточной анемии и непереносимости лактозы, чем кавказцы.

Основная жалоба - это объяснение проблем пациента в его или ее собственных словах. Основной жалобой может быть, например: "Я просыпаюсь каждое утро с сильной болью в животе" или "Каждое утро, я испытываю чувство тошноты при чистке зубов" и т.д. Подробное описание проблемы очень полезно для того чтобы узнать, что случилось с пациентом. За годы практики, врач услышал широкий спектр жалоб и может использовать этот опыт, чтобы разработать дифференциальный диагноз. В некоторых случаях врач может иметь хорошее представление о том, какой диагноз у пациента по его жалобам и может установить то, что называют "предполагаемый диагноз".

ИНЗ в основном подробное описание симптомов пациента. Врач должен знать, например, какая часть тела болит, при болях в других частях тела, когда начали проявляться симптомы, как долго продолжается дискомфорт, что делает его более или менее интенсивным и многие, другие детали и симптомы. Симптомы должны быть получены в хронологическом порядке.

Обзор систем - это процесс, в котором врач оценивает каждую часть тела, чтобы определить, откуда происходит заболевание. Такой обзор систем в себя включает общее состояние кожи, головы, глаз, ушей, носа, горла, органов дыхания, сердечно-сосудистой системы, желудочно-кишечной системы, гинекологической системы, мочеполовой системы, эндокринной системы, опорно-двигательного аппарата, периферических сосудов, гематологию, и нервно-психическую систему. Некоторые примеры того, что врач ищет в этих системах, являются следующие (по опросу врачей):

* Общее состояние: потеря или набор веса, слабость, лихорадка, озноб и т.д.;
* Уши: изменения слуха, выделения, боль и т.д.;
* Желудочно-кишечного тракта: изжога, тошнота, боли в животе, понос и другие симптомы.

Осмотр – это визуальный осмотр пациента. Есть много вещей, которые вы можете увидеть, глядя на пациента и это может помочь в определении диагноза. Вы можете оценить настроение человека, как он ходит (походка), а также потенциальные патологии на открытых участках кожи (сыпь, шрамы, родинки и т.д.), повреждения глаз (расширенные сосуды, кровоизлияния, движения глаз и т.д.) и многие другие элементы. Большинство полученных результатов осмотра не являются диагностическими сами по себе, но есть и исключения.

Дополнительные тесты, это те, которые назначает врач, чтобы помочь подтвердить или опровергнуть диагноз, давая дополнительную информацию. Некоторые из наиболее распространенных тестов являются: Биопсия (разреза, иссечения, тонкоигольная аспирационная и т.д.)   
Полезна в выявлении злокачественной или доброкачественной опухоли, инфекционных заболеваний и других патологических процессов обработки изображений исследования (рентген, МРТ, компьютерная томография, УЗИ, ангиография и т.д.) Помогает визуально определить заболевание или повреждения в твердой и / или мягких тканей, таких как опухоли, переломы костей, кисты, воспаления, накопление жидкости, травмы мягких тканей, врожденных уродств, воздушной эмфиземы и т.д. Исследования крови (полный анализ крови, тромбоциты, дифференциальные лейкоциты в крови и т.д.) Помогает определить некоторые виды рака (например, лейкемию), аномальное количество красных или белых кровяных клеток и тромбоцитов, слишком высокий или слишком низкий гемоглобин и т.д. Дополнительные тесты, как правило, назначают, чтобы подтвердить предполагаемый диагноз, устранить не правильный диагноз, сузить дифференциальный диагноз или дать дополнительные сведения, чтобы помочь прийти к трудному окончательному диагнозу.

В процессе анализа предметной области диагностирования заболеваний у пациентов выявлено достаточно много проблем с которыми сталкивается лечащий врач с самого начала диагностики заболевания у пациента.

Прежде всего, постановка диагноза процесс творческий, как правило, не укладывающийся в стандартные схемы. Как нет двух одинаковых врачей, так и нет двух одинаковых пациентов, поэтому и спектр диагностических приемов для распознавания «образа больного», страдающего определенной формой заболевания, весьма широк у опытного, квалифицированного, профессионально подготовленного специалиста. У молодого врача этот спектр беднее, он более схематичен.

Несмотря на то, что диагностика – это не столько ремесло, сколько искусство, действия врача можно уложить в логическую схему «диагностической технологии» и рассматривать процесс диагностики как работу некоего конвейера по производству диагностических услуг, в начале которого будет некий образ «человека-невидимки», а в конце - этот образ наполнится красками, объемом и содержанием. Технологический подход позволит представить лечащему врачу подробную информацию о пораженных органах и системах организма пациента, причинно-следственные связи возникновения заболевания, нарушенные функции сложной биологической системы, которой является пациент.

Т.к. процесс диагностики нельзя описать в четко определенный алгоритм действий, принято решение использовать для решения проблем диагностики методы машинного обучения.

Машинное обучение (Machine Learning)  — обширный подраздел [искусственного интеллекта](http://www.machinelearning.ru/wiki/index.php?title=%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82), изучающий методы построения [алгоритмов](http://www.machinelearning.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC), способных обучаться.

Различают два типа обучения. Обучение по прецедентам, или индуктивное обучение, основано на выявлении общих закономерностей по частным [эмпирическим данным](http://www.machinelearning.ru/wiki/index.php?title=%D0%92%D1%8B%D0%B1%D0%BE%D1%80%D0%BA%D0%B0). Дедуктивное обучение предполагает формализацию знаний экспертов и их перенос в компьютер в виде [базы знаний](http://www.machinelearning.ru/wiki/index.php?title=%D0%91%D0%B0%D0%B7%D0%B0_%D0%B7%D0%BD%D0%B0%D0%BD%D0%B8%D0%B9&action=edit). Дедуктивное обучение принято относить к области [экспертных систем](http://www.machinelearning.ru/wiki/index.php?title=%D0%AD%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), поэтому термины машинное обучение и обучение по прецедентам можно считать синонимами.

Машинное обучение находится на стыке [математической статистики](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0), [методов оптимизации](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D1%8B_%D0%BE%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8&action=edit) и классических математических дисциплин, но имеет также и собственную специфику, связанную с проблемами [вычислительной эффективности](http://www.machinelearning.ru/wiki/index.php?title=%D0%92%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%8D%D1%84%D1%84%D0%B5%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C&action=edit) и [переобучения](http://www.machinelearning.ru/wiki/index.php?title=%D0%9F%D0%B5%D1%80%D0%B5%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Многие методы индуктивного обучения разрабатывались как альтернатива классическим статистическим подходам. Многие методы тесно связаны с [извлечением информации](http://www.machinelearning.ru/wiki/index.php?title=%D0%98%D0%B7%D0%B2%D0%BB%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%B8&action=edit) и [интеллектуальным анализом данных](http://www.machinelearning.ru/wiki/index.php?title=%D0%98%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) (Data Mining).

Наиболее теоретические разделы машинного обучения объединены в отдельное направление, [теорию вычислительного обучения](http://www.machinelearning.ru/wiki/index.php?title=%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) (Computational Learning Theory, COLT).

Машинное обучение — не только математическая, но и практическая, инженерная дисциплина. Чистая теория, как правило, не приводит сразу к методам и [алгоритмам](http://www.machinelearning.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC), применимым на практике. Чтобы заставить их хорошо работать, приходится изобретать дополнительные [эвристики](http://www.machinelearning.ru/wiki/index.php?title=%D0%AD%D0%B2%D1%80%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0&action=edit), компенсирующие несоответствие сделанных в теории предположений условиям реальных задач. Практически ни одно исследование в машинном обучении не обходится без эксперимента на [модельных](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5&action=edit) или [реальных](http://www.machinelearning.ru/wiki/index.php?title=%D0%A0%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5&action=edit) данных, подтверждающего практическую работоспособность метода.

Большинство задач, для решения которых используются системы искусственного интеллекта, можно разделить на две группы. Первая группа – поиск последовательности действий, которые приведут от начального состояния к цели. К ним относятся стратегические игры, поиск оптимального маршрута, планирование операций и т.п.

Теоретическую основу СИИ, предназначенных для решения таких задач, составляет теория графов. Вторая группа – отнесение объекта к одному из известных классов. К ним относятся экспертные системы в таких областях как медицина, техника, геология, химия, генетика, прогнозирование погоды и т.п., а также системы распознавания, идентификации и контроля различных объектов.

Теоретическую основу СИИ, предназначенных для решения таких задач, составляет теория распознавания образов. Одно из основных отличий систем распознавания от экспертных систем состоит в том, что в системе распознавания, как правило, распознаваемый образ представляется вектором признаков, в котором перед началом распознавания все значения элементов известны. В экспертных системах процесс распознавания совмещен с процессом получения информации об объекте, причем последовательность получения значений признаков меняется в зависимости от полученных ранее данных. Такой подход к распознаванию применяется при большом количестве признаков и классов, иерархической структуре классов и отсутствии в описании каждого класса всех признаков. Кроме этого, в экспертных системах часто требуются большие затраты на получение значений признаков.

В разрабатываемом решении предполагается, что при введении системы в эксплуатацию, база данных будет содержать больше количество, часто изменяемы и добавляемых данных, что само по себе предполагает вывод о том, что эти данные придется рано или поздно анализировать, составлять разнообразные отчеты.

Создание отдельных интерфейсов в системе для генерации отчетов, довольно дорогостоящий и длительный процесс, требующий разработки, тестирования и т.д., поэтому встает вопрос о том, можно ли как-то ускорить и упростить процесс аналитики данных?

В данном случае на помощь приходят OLAP-системы для анализа данных.Термин OLAP (On-Line Analytical Processing) служит для описания модели представления данных и соответственно технологии их обработки в хранилищах данных. В OLAP применяется многомерное представление агрегированных данных для обеспечения быстрого доступа к стратегически важной информации в целях углубленного анализа. OLAP имеет большое количество приемуществ по сравнении с разработкой отдельных интерфейсов и тем более над анализом данных на уровне OLTP.

Причина использования OLAP для обработки запросов — это скорость. [Реляционные БД](http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%91%D0%94) хранят сущности в отдельных таблицах, которые обычно хорошо нормализованы. Эта структура удобна для операционных БД (системы [OLTP](http://ru.wikipedia.org/wiki/OLTP)), но сложные многотабличные запросы в ней выполняются относительно медленно.

OLAP-структура, созданная из рабочих данных, называется [OLAP-куб](http://ru.wikipedia.org/wiki/OLAP-%D0%BA%D1%83%D0%B1). Куб создаётся из соединения таблиц с применением [схемы звезды](http://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B7%D0%B2%D0%B5%D0%B7%D0%B4%D1%8B) или [схемы снежинки](http://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D1%81%D0%BD%D0%B5%D0%B6%D0%B8%D0%BD%D0%BA%D0%B8). В центре схемы звезды находится[таблица фактов](http://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0_%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D0%B2), которая содержит ключевые факты, по которым делаются запросы. Множественные таблицы с измерениями присоединены к таблице фактов. Эти таблицы показывают, как могут анализироваться [агрегированные](http://ru.wikipedia.org/wiki/%D0%90%D0%B3%D1%80%D0%B5%D0%B3%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_(%D1%8D%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D0%BA%D0%B0)) реляционные данные. Количество возможных агрегирований определяется количеством способов, которыми первоначальные данные могут быть иерархически отображены.

Трудно найти в компьютерном мире человека, который хотя бы на интуитивном уровне не понимал, что такое базы данных и зачем они нужны. В отличие от традиционных реляционных СУБД, концепция OLAP не так широко известна, хотя загадочный термин "кубы OLAP" слышали, наверное, почти все. Что же такое OnLine Analytical Processing?

OLAP - это не отдельно взятый программный продукт, не язык программирования и даже не конкретная технология. Если постараться охватить OLAP во всех его проявлениях, то это совокупность концепций, принципов и требований, лежащих в основе программных продуктов, облегчающих аналитикам доступ к данным. Несмотря на то, что с таким определением вряд ли кто-нибудь не согласится, сомнительно, чтобы оно хоть на йоту приблизило неспециалистов к пониманию предмета. Поэтому в своем стремлении к познанию OLAP лучше идти другим путем. Для начала надо выяснить, зачем аналитикам надо как-то специально облегчать доступ к данным.

Дело в том, что аналитики - это особые потребители корпоративной информации. Задача аналитика - находить закономерности в больших массивах данных. Поэтому аналитик не будет обращать внимания на отдельно взятый факт, ему нужна информация о сотнях и тысячах событий. Кстати, один из существенных моментов, который привел к появлению OLAP - производительность и эффективность. Представим себе, что происходит, когда аналитику необходимо получить информацию, а средства OLAP на предприятии отсутствуют. Аналитик самостоятельно (что маловероятно) или с помощью программиста делает соответствующий SQL-запрос и получает интересующие данные в виде отчета или экспортирует их в электронную таблицу. Проблем при этом возникает великое множество. Во-первых, аналитик вынужден заниматься не своей работой (SQL-программированием) либо ждать, когда за него задачу выполнят программисты - все это отрицательно сказывается на производительности труда. Во-вторых, анализ, это не едино-разовая процедура, даже для анализа одних данных может потребоваться выгрузка данных в разных формах и проекциях. В-третьих, как правило анализируются именно большие объемы данных, а это означает (хотя техника и идет вперед семимильными шагами), что сервер корпоративной реляционной СУБД, к которому обращается аналитик, может задуматься глубоко и надолго, заблокировав остальные транзакции.

* 1. **Анализ данных при помощи OLAP**

Анализ – исследование данных с целью принятия решений. Аналитические системы так и называют — системы поддержки принятия решений (СППР).

Здесь стоит указать на отличие работы с СППР от простого набора регламентированных и нерегламентированных отчетов. Анализ в СППР практически всегда интерактивен и итеративен. Т.е. аналитик копается в данных, составляя и корректируя аналитические запросы, и получает отчеты, структура которых заранее может быть неизвестна.

Большие информационные системы, как правило, содержат приложения, предназначенные для комплексного анализа данных, их динамики, тенденций и т.п. Соответственно, основными потребителями результатов анализа становится топ-менеджмент. Такой анализ, в конечном итоге, призван содействовать принятию решений. А чтобы принять любое управленческое решение необходимо обладать необходимой для этого информацией, обычно количественной. Для этого необходимо эти данные собрать из всех информационных систем предприятия, привести к общему формату и уже потом анализировать. Для этого создают хранилища данных (Data Warehouses) с помощью технологии OLAP.

Системы поддержки принятия решений обычно обладают средствами предоставления пользователю агрегатных данных для различных выборок из исходного набора в удобном для восприятия и анализа виде (таблицы, диаграммы и т.п.). Традиционный подход сегментирования исходных данных использует выделение из исходных данных одного или нескольких многомерных наборов данных (нередко называемый гиперкубом или метакубом), оси которых содержат атрибуты, а ячейки – агрегируемые количественные данные. (Причем храниться такие данные могут и в реляционных таблицах, но в данном случае мы говорим о логической организации данных, а не о физической реализации их хранения.) Вдоль каждой оси атрибуты могут быть организованы в виде иерархий, представляющих различные уровни их детализации. Благодаря такой модели данных пользователи могут формулировать сложные запросы, генерировать отчеты, получать подмножества данных.

Технология комплексного многомерного анализа данных получила название [OLAP](http://ru.wikipedia.org/wiki/OLAP) (On-Line Analytical Processing). OLAP — это ключевой компонент организации традиционных хранилищ данных. Концепция OLAP была описана в 1993 году [Эдгаром Коддом](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D0%B4,_%D0%AD%D0%B4%D0%B3%D0%B0%D1%80), известным исследователем баз данных и автором реляционной модели данных. В 1995 году на основе требований, изложенных Коддом, был сформулирован так называемый тест FASMI (Fast Analysis of Shared Multidimensional Information — быстрый анализ разделяемой многомерной информации), включающий следующие требования к приложениям для многомерного анализа:

* Предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;
* Возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;
* Многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;
* Многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это — ключевое требование OLAP);
* Возможность обращаться к любой нужной информации независимо от ее объема и места хранения.

Следует отметить, что OLAP-функциональность может быть реализована различными способами, начиная с простейших средств анализа данных в офисных приложениях и заканчивая распределенными аналитическими системами, основанными на серверных продуктах. Т.е. OLAP — это не технология, а идеология.

В качестве системы для построения аналитических кубов на основе OLAP, было решено использовать Microsoft Analysis Services (ранее называвшиеся OLAP Services, часть SQL Server).

При построении информационной системы OLAP-функциональность может быть реализована как серверными, так и клиентскими OLAP-средствами.

При использовании OLAP-сервера необходимо построить 2 разные системы - для создания кубов на сервере, и для разработки клиентского приложения. OLAP-клиент предоставляет единый визуальный интерфейс для описания кубов и настройки к ним пользовательских интерфейсов.

В качестве клиентской системы для просмотра аналитических данных при помощи кубов на основе OLAP, было решено использовать Microsoft Excel 2016 в виду его удобности и распространенности среди пользователей, что приводит к крайне малым требованиям к порогу вхождения для работы с аналитическими данными и построению отчетов.

Принцип работы ROLAP-клиентов - предварительное описание семантического слоя, за которым скрывается физическая структура исходных данных. После этого пользователь может самостоятельно манипулировать понятными ему объектами в терминах предметной области для создания кубов и аналитических интерфейсов.

Принцип работы клиента OLAP-сервера иной. В OLAP-сервере при создании кубов пользователь манипулирует физическими описаниями БД.

При этом в самом кубе создаются пользовательские описания. Клиент OLAP-сервера настраивается только на куб.

Алгоритм работы OLAP-системы следующий:

* Получение данных в виде плоской таблицы или результата выполнения SQL запроса.
* Кэширование данных и преобразование их к многомерному кубу.
* Отображение построенного куба при помощи кросс-таблицы или диаграммы и т.п. В общем случае, к одному кубу может быть подключено произвольное количество отображений.

Подготовка данных OLAP-системы на сервере:

Данные, хранящиеся в таблице необходимо преобразовать для их использования. Так, в целях повышения производительности при построении гиперкуба, желательно находить уникальные элементы, хранящиеся в столбцах, являющихся измерениями куба. Кроме того, можно производить предварительное агрегирование фактов для записей, имеющих одинаковые значения размерностей. Как уже было сказано выше, для нас важны уникальные значения, имеющиеся в полях измерений.

При использовании такой структуры мы значительно снижаем потребность в памяти. Что довольно актуально, т.к. для увеличения скорости работы желательно хранить данные в оперативной памяти. Кроме того, хранить можно только массив элементов, а их значения выгружать на диск, так как они будут нам требоваться только при выводе кросс-таблицы.

OLAP-технологии – это мощный инструмент обработки данных в реальном времени. OLAP-сервер позволяет организовывать и представлять данные в разрезе различных аналитических направлений и превращает данные в ценную информацию, которая помогает компаниям принимать более обоснованные решения.

И в заключении описания по использованию OLAP-систем в разрабатываемом решении стоит выделить что заданный подход аналитики данных обеспечивает стабильно высокий уровень производительности и масштабируемости, поддерживая объемы данных размером в несколько гигабайт, доступ к которым могут получить тысячи пользователей. OLAP-технологии упрощают доступ к аналитической информации, который к тому же осуществляется в реальном времени, т.е. обработка запросов теперь не замедляет процесс анализа, обеспечивая его оперативность и эффективность. Визуальные инструменты администрирования позволяют разрабатывать и внедрять даже самые сложные аналитические приложения, делая этот процесс простым и быстрым.

OLAP-система позволяет пользователям проводить сложный анализ данных, что дает возможность лучше понять принципы функционирования ИС и найти способы улучшения ее работы.

* 1. **Технология обработки информации**

Разрабатываемое решение представляет аналитические инструменты, для анализа данных облачного сервиса взаимодействия пациента и врача, который позволит производить сбор и анализ первичных симптомов пациента ни прибегая к услугам врача, постановку первичных диагнозов посредством анализа входных данных нейронной сетью, на основании которого в будущем специалист сможет начать анализ и ведение онлайн-кабинета пациента с составлением медицинской карты пациента и онлайн-кабинета врача. Система позволит осуществлять дистанционные коммуникации между пациентом и врачом. А также сбор статистики по лечебным центрам, поликлиникам, а также по конкретным специалистам, что является не маловажной информацией для пациента при выборе лечебного заведения и конкретного врача.

Принцип работы системы взаимодействия врачей и пациентов (см. Приложение 2):

* Пациент на основе интуитивно понятного пользовательского интерфейса составляет симптоматику своего заболевания, путем выбора системы органов, части тела и конкретного симптома;
* После выбора симптомов, пользователю предоставляется возможность просмотреть предварительно постановленный системой диагноз, на данном этапе данные о диагнозе сохраняются в БД;
* Также, после составления симптоматики связаться с одним из предоставленных ему врачей для уточнения симптоматики и онлайн консультации;
* Пользователь может записаться на прием к врачу;
* Врач на основе данных системы и собственном анализе может добавить данные в медицинскую карту пациента для накопления дальнейшей базы знаний о пациенте.

Многие из этих действий нуждаются в частом анализе, например, активность переписок или сбор и анализ данных о постановках диагнозов в определенный период времени, для анализа заболеваемости людей в разные временные интервалы.

Принцип работы аналитической системы:

* Разработчик создает куб для анализа определенных данных;
* Разработчик оповещает аналитиков он ново-созданном аналитическом кубе;
* Аналитик запускает аналитическое клиентское приложение, подключается в базе аналитики и собирает нужные ему данные для анализа;
  1. **Входные данные облачного решения**

Входными данными являются: информация о пользователе, информация о сообщении, информация о симптомах, системах, частях тела, диагнозах.

|  |  |  |
| --- | --- | --- |
| Таблица 1.1 - Информация о сообщении. | | |
| **Название атрибута** | **Тип атрибута** | **Комментарий** |
| Текст | Текст | Текст сообщения |
| Диалог | Число | Диалог к которому прикреплено сообщение |
| Ссылка | Текст | Ссылка которая прикреплена к сообщению |
| Изображение | Массив байт | Изображение которое прикреплено к сообщению |
| Документ | Массив байт | Документ который прикреплен к сообщению |

|  |  |  |
| --- | --- | --- |
| Таблица 1.2 - Информация о симптоме. | | |
| **Название атрибута** | **Тип атрибута** | **Комментарий** |
| Имя | Текст | Название симптома |
| Мужчина | Флаг | Принадлежность мужчине |
| Женщина | Флаг | Принадлежность женщине |
| Координаты на изображении X | Число | Координаты по оси X на изображении системы |
| Координаты на изображении Y | Число | Координаты по оси Y на изображении системы |
| Координаты на изображении R | Число | Радиус |

|  |  |  |
| --- | --- | --- |
| Таблица 1.3 - Информация о системе. | | |
| **Название атрибута** | **Тип атрибута** | **Комментарий** |
| Имя | Текст | Название системы |
| Мужчина | Флаг | Принадлежность мужчине |
| Женщина | Флаг | Принадлежность женщине |
| Маленькое изображение | Массив байт | Дата создания постановления. |
| Большое изображение для мужчины | Массив байт | Изображение для мужчины |
| Большое изображение для женщины | Массив байт | Изображение для женщины |

|  |  |  |
| --- | --- | --- |
| Таблица 1.5 - Информация о части тела. | | |
| **Название атрибута** | **Тип атрибута** | **Комментарий** |
| Имя | Текст | Дата создания реестра. |
| Изображение для мужчины | Массив байт | Изображение для мужчины |
| Мужчина | Флаг | Принадлежность мужчине |
| Женщина | Флаг | Принадлежность женщине |
| Изображение для женщины | Массив байт | Изображение для женщины |
| Координаты на изображении X | Число | Координаты по оси X на изображении системы |
| Координаты на изображении Y | Число | Координаты по оси Y на изображении системы |
| Координаты на изображении R | Число | Радиус |

|  |  |  |
| --- | --- | --- |
| Таблица 1.6 - Информация о диагнозе. | | |
| **Название атрибута** | **Тип атрибута** | **Комментарий** |
| Имя | Текст | Название диагноза |
| Изображение | Массив байт | Изображение |
| Описание | Текст | Описание диагноза |

* 1. **Выходные документы облачного решения**

Выходной информацией является генерируемый нейронной системой ответ в виде списка десяти самых высоковероятных диагнозов с их процентным соотношением и амбулаторная карта пациента.

* 1. **Входные данные аналитического клиентского приложения**

Входной информацией для клиентского приложения являются созданные на сервере для анализа инструменты т.е. OLAP-куб.

* 1. **Выходные данные аналитического клиентского приложения**

Выходной информацией является генерируемый клиентской системой отчет на основе созданных инструментов на сервере, выбранных для анализа данных на клиенте и выборе типа отчета для отображения.

* 1. **Системные требования**

Разрабатываемая система должна соответствовать следующим требованиям, обусловленным техническим заданием к проекту.

* 1. **Требования к программному обеспечению**

Рекомендуемые требования к программному обеспечению - это такие требования, при которых разрабатываемая система будет успешно функционировать.

Сервер базы данных:

* Windows Server 2016;
* Microsoft SQL Server 2016.

Рабочее место аналитика:

* Любое устройство где установлен как минимум, Microsoft Excel 2000.

Эти требования являются минимальными требованиями к программному обеспечению.

* 1. **Требования к техническому обеспечению**

Рекомендуемые требования к техническому обеспечению – это такие требования, при которых разрабатываемая система будет успешно функционировать.

Сервер базы данных:

* процессор с тактовой частотой 3 ГГц;
* ОЗУ – 2 GB;
* 1 GB на жестком диске;
* сетевая плата.

Рабочее место аналитика:

* процессор с тактовой частотой в 1,5 ГГц;
* ОЗУ – 512 MB;
* 1 GB на жестком диске;
* сетевая плата;
* принтер.

Эти требования являются минимальными требованиями к программному обеспечению.

* 1. **Аналоги системы**

Аналогами системы автоматизации взаимодействия пациента и врача как облачного сервиса, являются такие сервисы как Microsoft Office 365 это пакетированное предложение облачных сервисов для повышения бизнес-продуктивности сотрудников. Это все те же хорошо знакомые Microsoft Outlook и приложения Microsoft Office – теперь вместе с дополнительными облачными службами. Или Evernote средство для ведения мультимедийных заметок, инструментом для которых выступает любой компьютер или мобильный терминал.

Так же аналогами системы автоматизации взаимодействия пациента как бизнес-процесса, являются такие системы как «[Online диагноз](http://online-diagnos.ru/)» и «[Diagnos.ru](http://www.diagnos.ru/)». Ни один из аналогов системы как бизнес процесса не использует в своих расчетах нейронную сеть.

Т.к. в нашем решении используется сервер базы данных MS SQL Server, системой для OLAP, является SQL Server Analysis Services. Аналоги данной системы представлены в Приложении 4.

# 2 РАБОЧИЙ ПРОЕКТ

1. 1. **Общие сведения о работе системы**

Программный продукт облачного решения разработан на языке программирования C# под управлением ОС Windows и взаимодействует с любыми видами устройств, где установлен один из браузеров Chrome, FireFox, IE выше 8 версии, Safari и т.д. (любой браузер с поддержкой HTML5) или на мобильном устройстве с ОС iOS8 или выше.

Автоматизированная система запускается переходом по ссылке DocApp.ru.

OLAP-сервер построен на базе SQL Server Analysis Services. Для работы с аналитическими средствами на стороне предполагается использование Microsoft Excel 2000 и выше.

* 1. **Функциональное назначение программного продукта**

Программное обеспечение предназначено для автоматизации и облегчения процесса анализа данных аналитиками.

* 1. **Описание физической архитектуры системы**

Архитектура облачного сервиса представлена на диаграмме развертывания (Приложение 1.), и включает в себя следующие элементы:

* Бэкенд-сервер, который также является и OLAP-сервером, соответствующий требованиям, описанным в пункте 1.5.2 технического проекта - «Требования к техническому обеспечению»
* сервер базы данных (ОС: Windows Server 2012, СУБД: MS SQL Server 2012).

В качестве средства разработки используется Microsoft Visual Studio 2015 и Visual Studio Code и текстовый редактор Sublime, а также MSSQL Server 2016.

Взаимодействие с Web-сервером осуществляется по протоколу HTTP.

* 1. **Разработка структуры базы данных**

Для решения поставленной задачи на основании анализа объектов автоматизации была спроектирована и разработана база данных DocApp (см. Приложение 3), а также была создана и реализована в среде Visual Studio 2015 с помощь MS SQL Analysis Services, модель куба DocApp.cube.

Целостность данных обеспечивается заложенными во время создания таблиц ограничениями на тип, размер и диапазон допустимых значений данных.

Ссылочная целостность поддерживается определенными во время создания таблиц ссылками на родительские таблицы.

В базу данных DocAppFinal входят следующие связанные таблицы из основной базы DocApp (см. Приложение 5), которые полностью наследуют все свойства сущностей из основной базы данных.

**AspNetUsers –** Содержит список пользователей (таб. 2.1).

Таблица 2.1

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Char | Код пользователя (Первичный ключ) |
| [Email] | Char | Электронная почта пользователя |
| [EmailConfirmed] | Bit | Информация о подтверждении электронной почты |
| [PasswordHash] | Char | Зашифорванный пароль |
| [SecurityStamp] | Char |  |
| [PhoneNumber] | Char | Телефонный номер пользователя |
| [PhoneNumberConfirmed] | Bit | Информация подтверждении телефонного номера |
| [TwoFactorEnabled] | Bit | Информация использовании двухфакторной авторизации для данного пользователя |
| [LockoutEnabledUtc] | Datetime | Дата заблокирования пользователя |
| [LockoutEnabled] | bit | Информация о доступе пользователя в систему |
| [AccessFailedCount] | Int | Количество неудачный авторизаций пользователя |
| [UserName] | Char | Имя пользователя |

**BaseUsers -** Содержит информацию о пользователе системы(табл.2.2).

Таблица 2.2

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код пользователя (Первичный ключ) |
| [Name] | Char | Имя пользователя |
| [SecondName] | Char | Фамилия пользователя |
| [Surname] | Char | Отчество пользователя |
| [IsDeleted] | Bool | Флаг информирующий о том удален пользователь или нет |
| [UserId] | Int | Код пользователя из таблицы AspNetUsers |
| [ConnectionId] | Char | Строка являющаяся идентификатором соединения для возможности работать с протоколом web sockets |

**Patients –** Содержит информацию о пациенте (табл.2.3).

Таблица 2.3

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код пациента (Первичный ключ) |
| [UserId] | Int | Код пользователя из таблицы BaseUser |

**Doctors -** Содержит информацию о докторе (табл.2.4).

Таблица 2.4

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код доктора (Первичный ключ) |
| [Speciality] | Char | Строка информирующая о специальности врача |
| [HospitalId] | Int | Идентификатор лечебного заведения где работает врач |
| [UserId] | Int | Код пользователя из таблицы BaseUser |

**Diseases -** Содержит информацию о болезни (табл.2.5).

Таблица 2.5

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код болезни (Первичный ключ) |
| [Name] | Char | Название диагноза |
| [CardId] | Int | Идентификатор карты которой принадлежит болезнь |
| [DiagnosId] | Int | Идентификатор поставленного диагноза |
| [DoctorId] | Int | Идентификатор доктора поставившего диагноз |
| [PatientId] | Int | Идентификатор пациента которому поставили диагноз |

**Diagnos -** Содержит информацию о диагнозе (табл.2.6).

Таблица 2.6

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код диагноза (Первичный ключ) |
| [Name] | Char | Название диагноза |
| [Image] | Byte[] | Информация изображения |
| [Description] | Char | Описание диагноза |

**OutpatientCards -** Содержит информацию о амбулаторной карте (табл.2.7).

Таблица 2.7

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код амбулаторной карты (Первичный ключ) |
| [PatientId] | Int | Идентификатор пациента которому принадлежит карта |

А также таблица Time, которая была создана в OLAP-кубе и не имеет никакого отношения к основной базе данных. Используется для построения отчетов в проекции времени. Связана с атрибутом DateDisease в таблице Disease.

**Time –** Таблица для привязки данных по времени (табл.2.8).

Таблица 2.8

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Год] | String | Год |
| [Дата] | Date | Дата |
| [День года] | String | День года |

* 1. **Инсталляция и выполнение программного продукта**

Для использования информационной системы разработчику необходимо:

* разместить файлы библиотек и файлы системы на web-сервере;
* настроить на web-сервере IIS (Internet Information Services) – привязка виртуального каталога на сервере к реальному каталогу на жестком диске;
* настроить базу данных – развернуть базу данных DocApp на сервере баз данных;
* развернуть куб.

Аналитик получает доступ к данным сервера по средствам MS Excel.

* 1. **Общий алгоритм работы программного продукта**

Взаимодействие разработчика с OLAP-сервером может осуществляться либо через MS SQL Server 2016, либо через Visual Studio 2015. Все действия с OLAP-сервисами должны происходить в проекте MultidimensionalProject, решения DocAppBackend.

Взаимодействие аналитика с сервисами, предоставленными OLAP-сервером, может осуществляться на любом устройстве, на котором установлен MS Excel выше 2000.

* 1. **Разработанные меню и интерфейсы**
     1. **Подготовка базы данных Analysis Services на сервере**

Первым шагом будет создание проекта в среде Business Intelligence Development Studio (BIDS). Запустите BIDS из группы программ Microsoft SQL Server 2016, нажмите File | New | Project (Файл | Создать | Проект). В диалоговом окне New Project (Создать проект) выберите Analysis Services Project (Проект Analysis Services). В текстовом поле Name (Имя) введите имя и измените расположение проекта по своему усмотрению. Нажмите OK для создания проекта.

Теперь добавьте источник данных, чтобы задать строку соединения для киоска данных. В Обозревателе решений правой кнопкой мыши щелкните по папке Data Sources (Источники данных) и выберите New Data Source (Создать источник данных). В Мастере создания источника данных на странице приветствия, если она не отключена, нажмите Next (Далее). На странице Select How to Define the Connection (Выбор метода определения соединения) нажмите New (Создать) для создания нового соединения. В Менеджере соединения по умолчанию установлено SQL Server Native Client, который подходит для этого проекта, хотя вы так же можете использовать OLE DB или .NET для доступа к данным.

Для определения соединения, введите имя вашего сервера в текстовом поле Server Name (Имя сервера) или выберите его из раскрывающегося списка, затем выберите ваш сервер в раскрывающемся списке и нажмите OK. Затем вернитесь в Мастера создания источника данных и нажмите Next. На странице Impersonation Information (Сведения об олицетворении) выберите настройку Use the service account (Использовать учетную запись сервиса). Учетная запись сервиса используется для чтения данных из источника при загрузке данных в базу данных SSAS и для этого должна иметь разрешения на чтение. Нажмите Next, а затем Finish (Завершить) для завершения работы мастера.

Создание представления источника данных. Теперь давайте создадим представление источника данных (data source view – DSV) как обобщение таблиц (или представлений) из источника данных, которое мы будем использовать для создания измерений и кубов. Вы можете вносить изменения в DSV без какой-либо модификации источника данных, что является весьма удобным, если вы имеете права только на чтение киоска и не можете вносить изменения в источник. В Обозревателе решений щелкните правой кнопкой мыши по папке Data Source Views (Представление источника данных) и затем нажмите New Data Source View (Создать представление источника данных).

При необходимости нажмите Next на странице приветствия. На странице выбора источника данных выберите источник данных, который вы только что добавили в проект и нажмите Next. Добавьте объекты в DSV, сделав двойной щелчок по всем необходимым таблицам или представлениям. Для создания DSV, с помощью которого мы найдем ответы на поставленные в начале статьи вопросы, добавьте в DSV таблицы по которым собираетесь собирать аналитические данные. Для изучения работы SSAS не плохо начать с простого DSV. При необходимости, в последствии вы всегда сможете добавить необходимое количество таблиц для решения других вопросов в своей системе бизнес-аналитики. Нажмите Next в Мастере представления источника данных, когда добавите все таблицы, и затем нажмите Finish. Законченный вид DSV показан на рис.1.

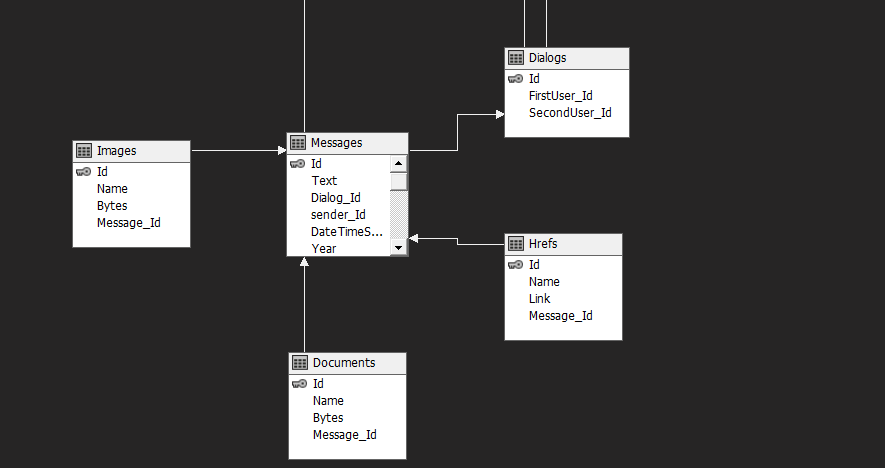


Рис. 1.1. Конструктор представления источника данных

Кроме изменения свойств таблиц в DSV, вы можете определить логические первичные ключи или логические взаимодействия между таблицами, если физический источник не имеет своих. SSAS не сможет отобразить данные корректно без этих определений. Вы также можете добавить именованные вычисления, которые подобны добавлению производного столбца к представлению, или замещению объекта таблицы именованным запросом, что подобно созданию представления.

Определение измерений в Analysis Services. После добавления таблиц измерений в DSV вы готовы к созданию измерений в базе данных SSAS. Как помните, измерения используются для хранения информации о бизнес-сущностях, таких как люди, места, предметы.

Чтобы добавить измерения в проект, запустите Мастер измерений, щелкнув правой кнопкой мыши по папке Dimensions (Измерения). на странице Select Creation Method (Выбор метода создания). В случае, когда вы будете создавать простую базу данных на основании небольшой таблицы транзакций, включающей даты, вы можете обойтись без киоска данных, а используйте одну из предлагаемых опций на этой странице для генерации таблицы дат в источнике данных или на сервере.

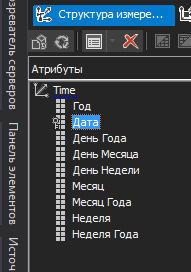


Рис. 1.2. Таблица Time с именованными вычислениями

В обоих случаях вы должны определить начальную и конечную даты для данных в таблице транзакций. SSAS может создать и заполнить физическую таблицу в источнике данных, который используется для заполнения измерений, созданных в базе данных SSAS, или может просто поддерживать измерения исключительно в базе данных SSAS.

Вы можете выбрать генерацию физической таблицы при поддержке запросов к реляционным источникам данных. Нажмите Next для продолжения.

На странице Specify Source Information (Определение исходных сведений) выберите таблицу с датами в раскрывающемся списке Main Table (Основная таблица). Ключевой столбец должен уникально идентифицировать каждую запись в таблице для того, чтобы SSAS корректно накапливал или группировал данные при возвращении результатов запроса. Затем, в раскрывающемся списке Name Column (Имя колонки) выберите колонку. Наименование колонки SSAS использует для пользовательского представления, вместо отображения ключевого столбца.

Нажмите Next.

На странице Select Dimension Attributes (Выбор атрибутов измерения) вы должны выбрать атрибуты, которые будут включены в измерение, для определения иерархии или дополнительных группировок.

Включение атрибутов из таблицы измерения зависит от задачи, которую вы будете решать с помощью своей системы бизнес-аналитики. Добавьте лишь те атрибуты, которые необходимы вам для оптимальной работы SSAS. Это поможет избежать излишней нагрузки на систему, и не будет пугать пользователей обилием опций.

Для того чтобы обеспечить уникальную идентификацию каждого атрибута, обновите свойство KeyColumn. Настроив это свойство в ключевых колонках, вы добьетесь того, что ваши измерения будут корректно выводиться при сортировке в численной последовательности, нежели в буквенной (по наименованию месяца), которая устанавливается по умолчанию.

При правильной настройке в Обозревателе решений появится новое измерение .dim и конструктор измерения (рис.3) отобразится в основном рабочем пространстве BIDS.

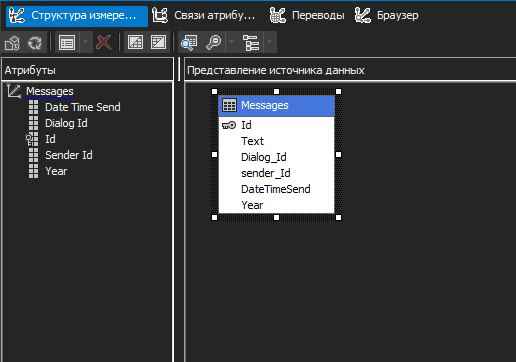


Рис. 1.3. Измерение Messages в конструкторе

Вы можете увидеть измерение со связанными атрибутами, включая ключевые атрибуты, на панели Attributes, слева. Если вы потом захотите добавить больше атрибутов, вы можете перетащить их один за другим с панели Data Source View на панель Attributes. Используйте Мастер измерений только для создания новых измерений, а все изменения вносите с помощью конструктора измерений.

Создание куба. Так же, как вы использовали Мастер измерений для создания измерения, для создания куба используйте Мастер куба. В обозревателе решений щелкните правой кнопкой мыши по папке Cubes (Кубы), нажмите New Cube (Создать Куб) и затем Next на странице приветствия, при необходимости. На странице Select Creation Method (Выбор метода создания) оставьте настройки по умолчанию. (Две другие настройки вы можете использовать при создании схемы вручную или с помощью шаблона.Тогда SSAS самостоятельно создаст таблицы в источнике данных на основе этой схемы, которые нужно будет заполнить данными с помощью службы Integration Services, прежде чем вы сможете просмотреть свой куб.) Нажмите Next.

На странице Select Measure Group Tables (Выбор таблиц групп мер) выберите интересующие вас таблицы и нажмите Next. (Таблицы групп мер являются синонимами таблиц фактов.) Мастер покажет все числовые колонки, найденные в выбранных таблицах групп мер. В соответствии с простым подходом, которого мы придерживались до сих пор, очистите все меры, убрав флажок Measure, расположенный в верхней части страницы, и затем выберите интересуюшие вас меры для каждой.

Вы также можете переименовать меры на этой странице. Щелкните правой кнопкой мыши на имени меры и введите новое имя, но убедитесь, что каждое имя будет уникальным. Имя меры должно быть коротким, но не настолько коротким, чтобы быть непонятным.

На странице Select Existing Dimensions (Выбор существующего измерения), мастер покажет уже созданные измерения. Нажмите Next. Если существуют таблицы в DSV, которые еще не являются таблицами групп мер или существующими измерениями, страница Select New Dimensions (Выбор нового измерения) позволит быстро добавить любое другое необходимое измерение.

Добавьте новые измерения и нажмите Next, переименуйте куб и нажмите Finish. Поздравляю!

Вы только что построили простейший куб! Конструктор куба покажет группы мер и измерения, добавленные в куб на панели слева и DSV справа, как показано на рис.5.

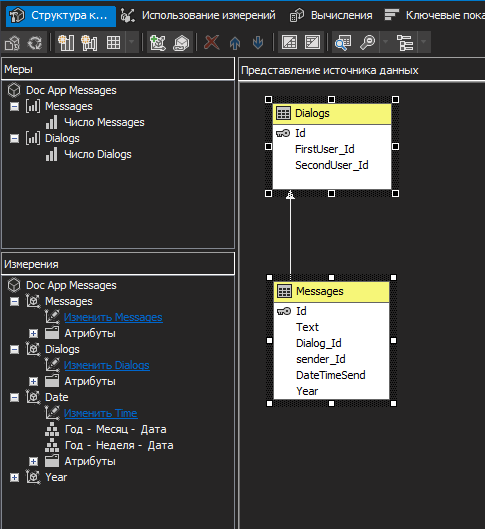


Рис. 1.4. Конструктор куба показывает группы мер и измерений

Первым шагом после создания куба является настройка свойства FormatString для каждой меры. Это делается для облегчения просмотра куба в браузере. Быстрый способ сделать это — просмотреть меры в таблице. На панели конструктора куба нажмите пятую слева кнопку и переключите вид с дерева на таблицу. В таблице вы можете использовать клавишу Ctrl для выбора нескольких мер одновременно. Для начала выберите интересующие вас меры. В окне Properties, в раскрывающемся списке FormatString, выберите Standard. Затем выберите все оставшиеся меры и установите свойство FormatString в Currency.

Развертывание базы данных Analysis Services из BIDS. Несмотря на то, что мы построили объекты системы, теперь необходимо создать базу данных SSAS на сервере, ведь сама по себе она еще не существует. Те задачи проектирования, которые выполнены в BIDS, создают XML-файлы, которые должны быть развернуты на сервере.

Как только эти файлы будут развернуты, вы можете обработать базу данных, которая выполнит команды, необходимые для построения структуры хранилища.

Структура хранилища определяется XML-файлами и заполняется с помощью выполнения запросов к указанному источнику данных. Откроется окно Deployment Progress (Выполнение развертывания), показывающее каждый шаг и успешность выполнения операции.

После того, как вы развернете проект, вы все еще можете делать изменения в конструкторе. Просто разверните проект снова, как было описано ранее, чтобы внести изменения на сервер, и запустите обработку. При появлении подсказки, что база данных будет обновлена — согласитесь, если вы абсолютно уверены в том, что вы единственный, кто сделал изменения в базе.

Иногда изменения, которые вы сделали, не будут запускать команду обработки базы данных после развертывания. В этом случае, просто щелкните правой кнопкой мыши по измерению или кубу, в который внесены изменения, нажмите Process, и затем Run. Если вы обрабатываете измерение при помощи настройки Process Full (которая потребуется, если вы сделали значительные изменения в структуре измерения), вам так же может потребоваться обработать куб.

* + 1. **Построение куба и анализ данных на клиенте**

Запускаем Microsoft Office Excel 2016 и переходим в раздел «Данные», после чего для выбора источника данных, для осуществления взаимодействия с OLAP-сервером, выбираем выгрузку из служб аналитики SQL Server.

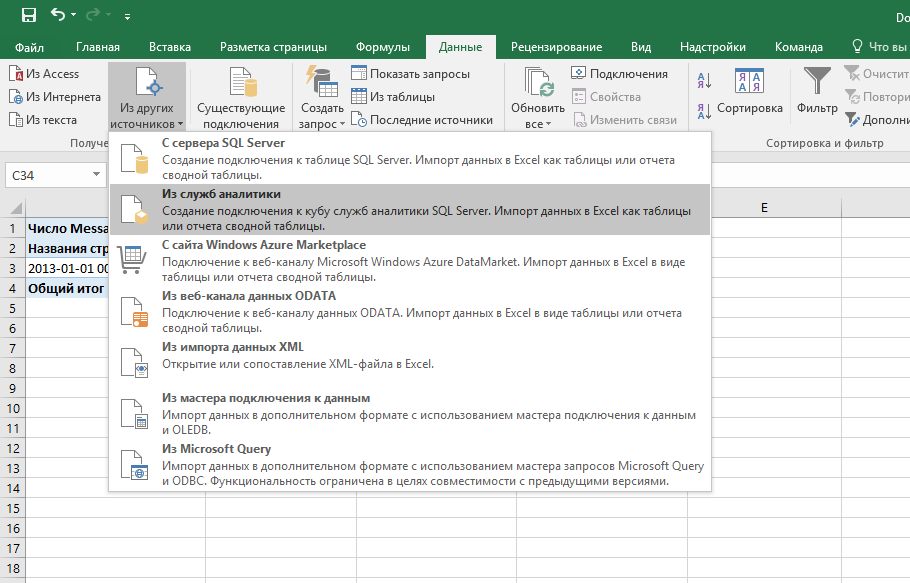
****

Рис. 1.4. Выбор источника

Выбор базы данных SSAS для синхронизации данных между OLAP-сервером и клиентом.

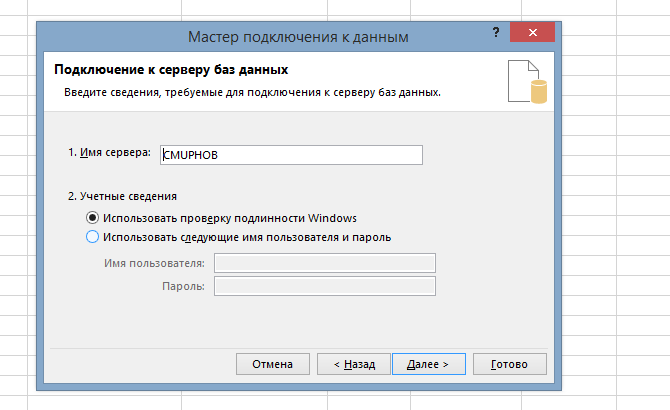
****

Рис. 1.6. Подключение к серверу БД

Выбор базы данных SSAS для синхронизации данных между OLAP-сервером и клиентом.

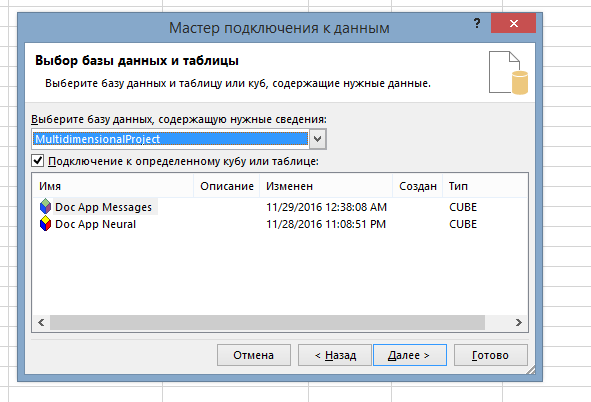
****

Рис. 1.7. Выбор SAAS базы данных

После чего, в разделе «Поля для сводной таблицы можно увидеть данные из OLAP Cube и воспользоваться имя для генерации отчетов, также можно настроить различные фильтры и даже изменить период синхронизации данных с сервером.

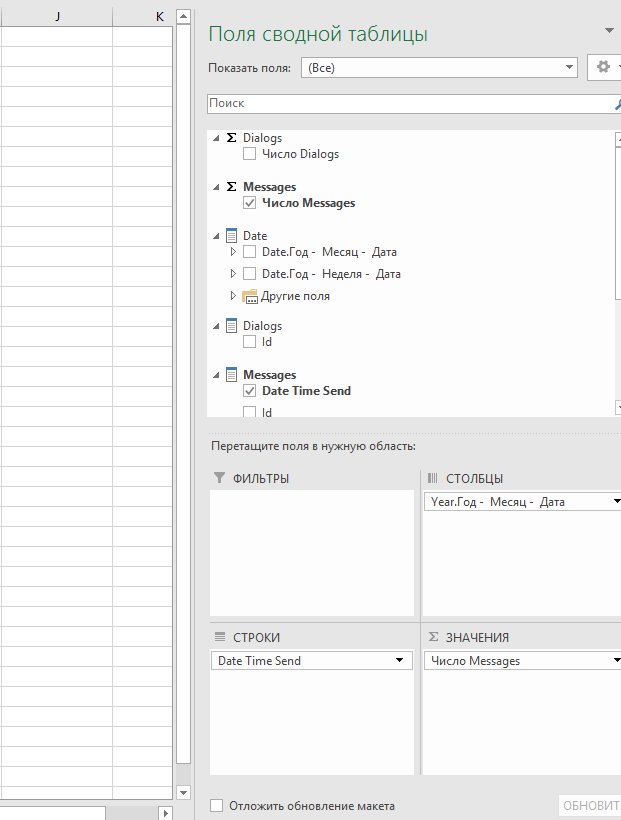
****

Рис. 1.8. Настройки отчета

* + 1. **Отчеты**
       1. **Отчеты, построенные в браузере OLAP-сервера**

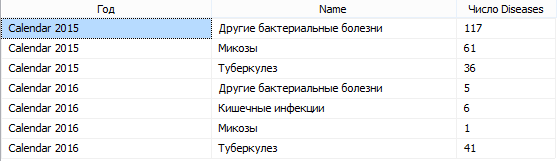


Рис. 1.10. Серверный отчет по болезням, сгруппированным по годам.

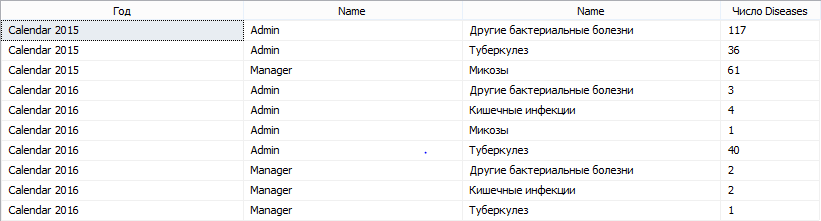


Рис. 1.11. Серверный отчет по болезням, сгруппированным по годам и по пользователям.

* + - 1. **Отчеты, построенные в MS Excel 2016**

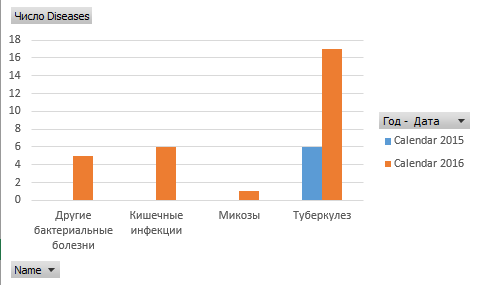


Рис. 1.11. Клиентский отчет по болезням, сгруппированным по годам.

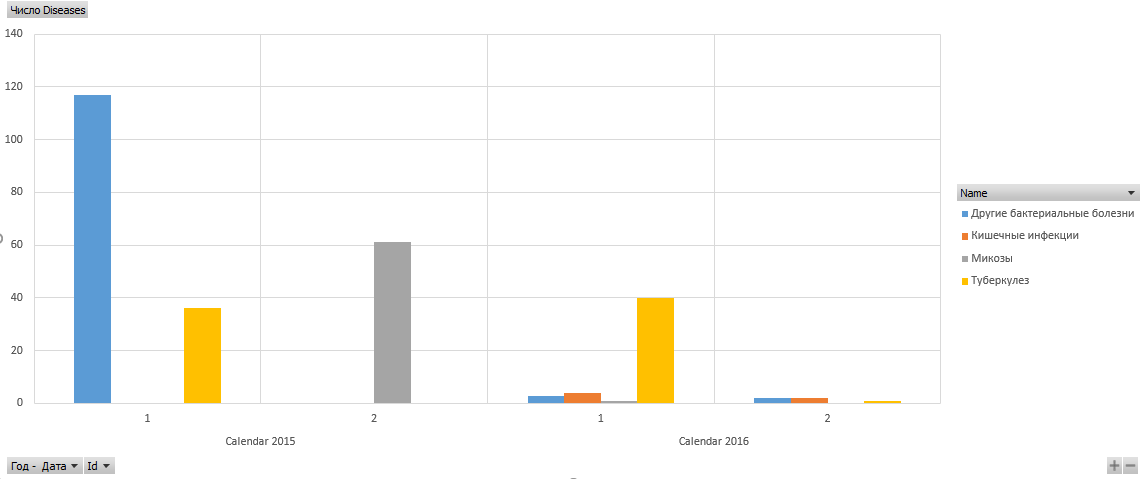


Рис. 1.12. Клиентский отчет по болезням, сгруппированным по годам и пользователям.

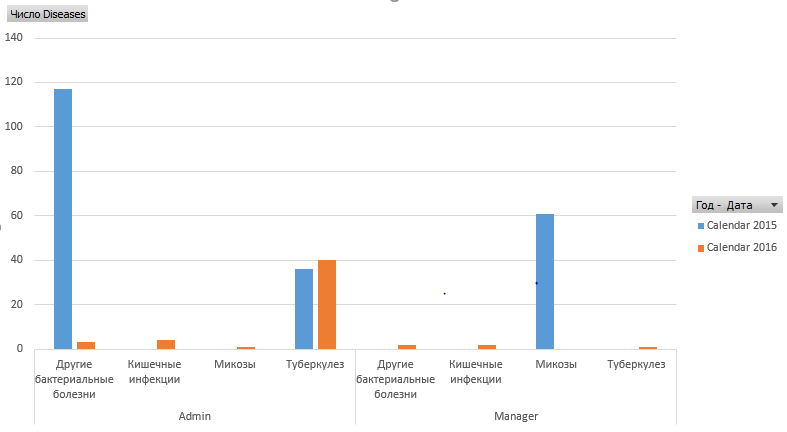


Рис. 1.13. Клиентский отчет гистограмма по болезням, сгруппированным по годам и пользователям.

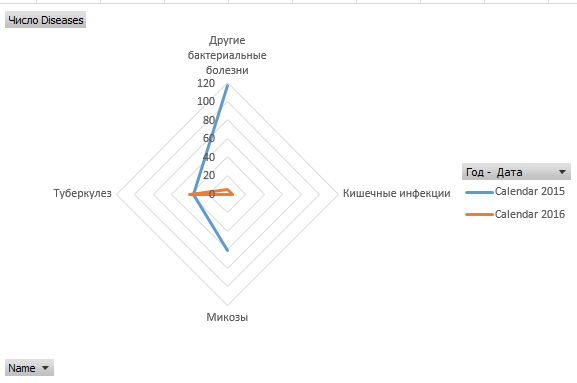


Рис. 1.14. Клиентский лепестковый отчет по болезням, сгруппированным по годам и пользователям.

Рис. 1.15. Клиентский круговой отчет по всем болезням в процентном соотношении.

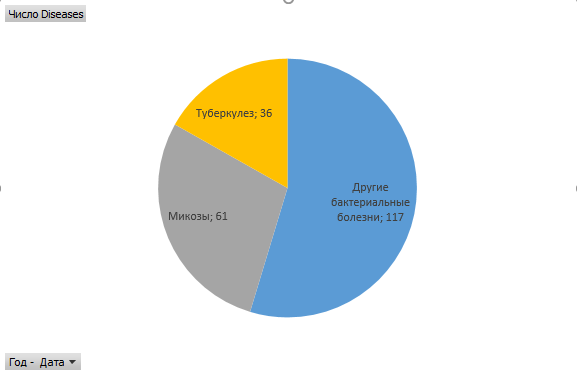
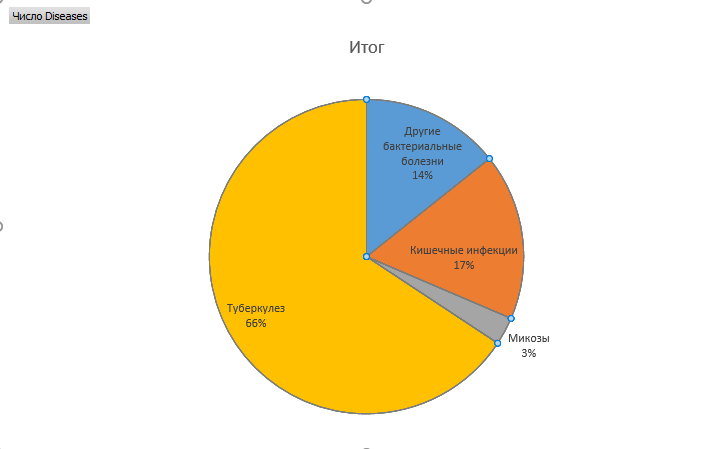


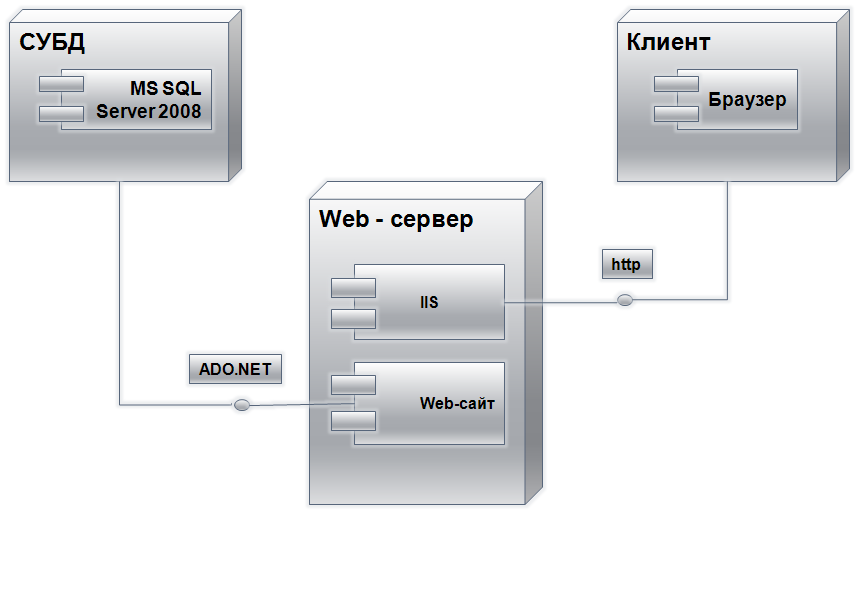
Рис. 1.16. Клиентский круговой отчет по всем болезням в количественном соотношении.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ASP.NET MVC 4 с примерами на C# 5.0 для профессионалов.;
2. Язык программирования C# 5.0 и платформа .NET 4.5.;
3. <http://knockoutjs.com>.;
4. JavaScript: Подробное руководство (Definitive Guide).;
5. Бен Хеник — HTML и CSS Путь к совершенству.;
6. Хайкин.С. Нейронные сети: полный курс, 2-е издание. 2011.;
7. <http://www.habarov.spb.ru/>;
8. <http://www.machinelearning.ru/>;
9. <https://ru.wikipedia.org/>;
10. <https://habrahabr.ru/>;
11. <http://www.olap.ru/>;
12. https://technet.microsoft.com.

**ПРИЛОЖЕНИЕ 1**

**ДИАГРАММА РАЗВЕРТЫВАНИЯ**



MS SQL Server 2012

Браузер

IIS

Web-сайт

OLAP-cервер

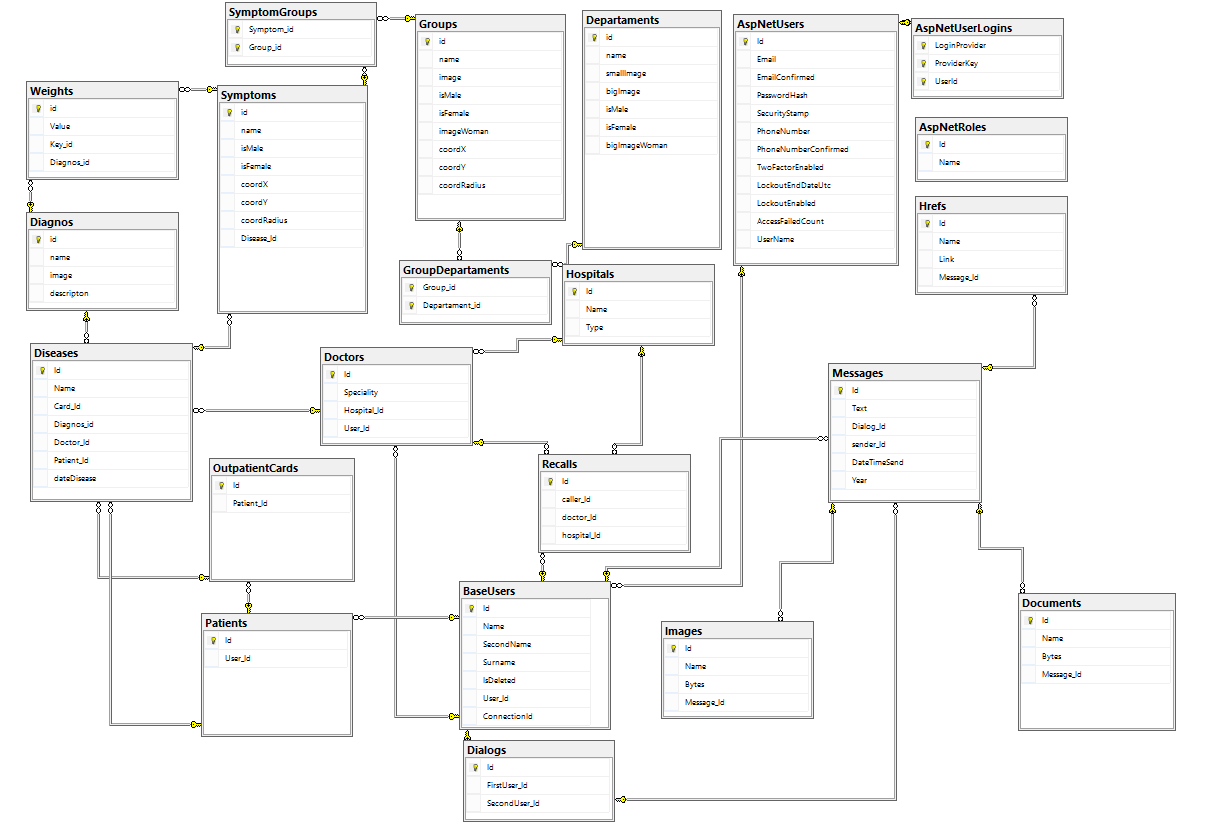
**ПРИЛОЖЕНИЕ 2**

**БЛОК-СХЕМА**

ИЛИ

**ПРИЛОЖЕНИЕ 3**

**ФИЗИЧЕСКАЯ ДИАГРАММА БАЗЫ ДАННЫХ**



**ПРИЛОЖЕНИЕ 4**

**ОБЗОР АНАЛОГОВ**



**ПРИЛОЖЕНИЕ 5**

**ДИАГРАММА БАЗЫ ДАННЫХ OLAP** 