|  |  |  |
| --- | --- | --- |
|  | ***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** | | |
| **Проектирование и разработка информационной системы автоматизации процесса взаимодействия пациента и врача**  по дисциплине «Искусственный интеллект» | | |
| Допущен к защите  «\_\_» \_\_\_\_\_\_\_\_2016г.  Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Оценка, полученная на защите «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_» | | Проект выполнен  студентом группы ДИНРМ-11 Смирнов. К.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Руководитель работы доцент Седова.Я.А. |
| 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-11*** | | |
| Дисциплина | | ***Искусственный интеллект*** | |
| Тема курсового проекта | | | ***Проектирование и разработка информационной системы*** |
| ***автоматизации процесса взаимодействия пациента и врача*** | | | |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Руководитель | | ***Доцент*** |  | ***Седова.Я.А.*** |
|  | | 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  «Искусственный интеллект» |
| 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-полнени |
| Дата | Объем в % | Дата | Объем в % |
| 1 | Выбор темы | 20.02.2016 | 1 |  |  |  |
| 2 | Техническое задание | 28.02.2016 | 5 |  |  |  |
| 3 | Разработка модели,  проектирование системы   * *введение,* * *технический проект,* * *программа и методика испытаний,* * *литература* | 22.03.2016 | 30 |  |  |  |
| 4 | Программная реализация системы   * *работающая программа,* * *рабочий проект* * *скорректированное техническое задание (при необходимости)* | 30.04.2016 | 60 |  |  |  |
| 5 | Тестирование и отладка системы, эксперименты   * *работающая программа с внесёнными изменениями,*   ***окончательные тексты***   * *введение,* * *технический проект,* * *рабочий проект* * *программа и методика испытаний* | 02.05.2016 | 80 |  |  |  |
| 6 | Компоновка текста  Подготовка презентации и доклада   * *пояснительная записка* * *презентация* | 20.05.2016 | 100 |  |  |  |
| 7 | Защита курсового проекта | 27.05.2016 | 100 |  |  |  |

Руководитель курсового проекта доцент Седова.Я.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ученая степень, звание, фамилия, подпись)

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

(фамилия, инициалы, подпись)

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

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc467887337)

[1 ТЕХНИЧЕСКИЙ ПРОЕКТ 6](#_Toc467887338)

[1.1 Анализ предметной области 6](#_Toc467887339)

[1.1.1 Процесс постановки диагноза 6](#_Toc467887340)

[1.1.2 Искусственные нейронные сети 10](#_Toc467887341)

[1.1.3 Параллели из биологии 12](#_Toc467887342)

[1.1.4 Описание моделей нейронных сетей 13](#_Toc467887343)

[1.1.5 Классификация нейронных сетей 15](#_Toc467887344)

[1.1.6 Описание алгоритма работы нейронной сети 17](#_Toc467887345)

[1.2 Обучение нейронной сети 24](#_Toc467887346)

[1.3 Технология обработки информации 26](#_Toc467887347)

[1.4 Входные данные 34](#_Toc467887348)

[1.5 Выходные документы 36](#_Toc467887349)

[1.6 Системные требования 36](#_Toc467887350)

[1.7 Требования к программному обеспечению 37](#_Toc467887351)

[1.8 Требования к техническому обеспечению 37](#_Toc467887352)

[1.9 Аналоги системы 37](#_Toc467887353)

[2 РАБОЧИЙ ПРОЕКТ 39](#_Toc467887354)

[2.1 Общие сведения о работе системы 39](#_Toc467887356)

[2.2 Функциональное назначение программного продукта 39](#_Toc467887357)

[2.3 Описание физической архитектуры системы 39](#_Toc467887358)

[2.4 Разработка структуры базы данных 39](#_Toc467887359)

[2.5 Инсталляция и выполнение программного продукта 47](#_Toc467887360)

[2.6 Общий алгоритм работы программного продукта 47](#_Toc467887361)

[2.7 Разработанные меню и интерфейсы 48](#_Toc467887362)

[2.7.1 Интерфейс "Домашняя страница" 48](#_Toc467887363)

[2.7.2 Интерфейс "Постановка диагноза для мужчины" 48](#_Toc467887364)

[2.7.3 Интерфейс "Постановка диагноза для мужчины" 49](#_Toc467887365)

[2.7.4 Интерфейс "Просмотр диагноза" 50](#_Toc467887366)

[2.7.5 Интерфейс "Административная панель" 50](#_Toc467887367)

[2.7.6 Интерфейс "Создание изменения систем" 51](#_Toc467887368)

[2.7.7 Интерфейс "Создание изменения частей тела" 51](#_Toc467887369)

[2.7.8 Интерфейс "Создание изменение симптомов" 52](#_Toc467887370)

[2.7.9 Интерфейс "Создание изменение диагнозов" 52](#_Toc467887371)

[2.7.10 Интерфейс "Обучение диагноза" 53](#_Toc467887372)

[2.7.11 Интерфейс "Апи" 53](#_Toc467887373)

[2.7.12 Сообщения системы 54](#_Toc467887374)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 55](#_Toc467887375)

[ПРИЛОЖЕНИЕ 1 56](#_Toc467887376)

[ПРИЛОЖЕНИЕ 2 57](#_Toc467887377)

[ПРИЛОЖЕНИЕ 3 58](#_Toc467887378)

# 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.

* + 1. **Экспертные системы**

Основным назначением ЭС является разработка программных средств, которые при решении задач, трудных для человека, получают результаты, не уступающие по качеству и эффективности решения, решениям получаемым человеком-экспертом. ЭС используются для решения так называемых неформализованных задач, общим для которых является то, что:

* Задачи не могут быть заданы в числовой форме;
* Цели нельзя выразить в терминах точно определенной целевой функции;
* Не существует алгоритмического решения задачи;
* Если алгоритмическое решение есть, то его нельзя использовать из-за
* Ограниченности ресурсов (время, память).

Кроме того, неформализованные задачи обладают ошибочностью, неполнотой, неоднозначностью и противоречивостью как исходных данных, так и знаний о решаемой задаче.

Экспертная система - это программное средство, использующее экспертные знания для обеспечения высокоэффективного решения неформализованных задач в узкой предметной области. Основу ЭС составляет база знаний (БЗ) о предметной области, которая накапливается в процессе построения и эксплуатации ЭС. Накопление и организация знаний - важнейшее свойство всех ЭС.

Особенности ЭС, отличающие их от обычных программ, заключаются в том, что они должны обладать:

* Компетентностью, то есть:
  + Достигать экспертного уровня решений (т.е. в конкретной предметной области иметь тот же уровень профессионализма, что и эксперты-люди).
  + Быть умелой (т.е. применять знания эффективно и быстро, избегая, как и люди, ненужных вычислений).
  + Иметь адекватную робастность (т.е. способность лишь постепенно снижать качество работы по мере приближения к границам диапазона компетентности или допустимой надежности данных).
* Возможностью к символьным рассуждениям, то есть:
  + Представлять знания в символьном виде
  + Переформулировать символьные знания. На жаргоне искусственного интеллекта символ - это строка знаков, соответствующая содержанию некоторого понятия. Символы объединяют, чтобы выразить отношения между ними. Когда отношения представлены в ЭС они называются символьными структурами.
* Глубиной, то есть:
  + Работать в предметной области, содержащей трудные задачи
  + Использовать сложные правила (т.е. использовать либо сложные конструкции правил, либо большое их количество)
* Самосознанием, то есть:
  + Исследовать свои рассуждения (т.е. проверять их правильность)
  + Объяснять свои действия

Существует еще одно важное отличие ЭС. Если обычные программы разрабатываются так, чтобы каждый раз порождать правильный результат, то ЭС разработаны с тем, чтобы вести себя как эксперты. Они, как правило, дают правильные ответы, но иногда, как и люди, способны ошибаться.

Традиционные программы для решения сложных задач, тоже могут делать ошибки. Но их очень трудно исправить, поскольку алгоритмы, лежащие в их основе, явно в них не сформулированы. Следовательно, ошибки нелегко найти и исправить. ЭС, подобно людям, имеют потенциальную возможность учиться на своих ошибках.

Основные свойства экспертных систем:

* Применение для решения проблем высококачественного опыта, который представляет уровень мышления наиболее квалифицированных экспертов в данной области, что ведёт к решениям творческим, точным и эффективным;
* Наличие прогностических возможностей, при которых ЭС выдаeт ответы не только для конкретной ситуации, но и показывает, как изменяются эти ответы в новых ситуациях, с возможностью подробного объяснения каким образом новая ситуация привела к изменениям;
* Обеспечение такого нового качества, как институциональная память, за счeт входящей в состав ЭС базы знаний, которая разработана в ходе взаимодействий со специалистами организации, и представляет собой текущую политику этой группы людей. Этот набор знаний становится сводом квалифицированных мнений и постоянно обновляемым справочником наилучших стратегий и методов, используемых персоналом. Ведущие специалисты уходят, но их опыт остаeтся;
* Возможность использования ЭС для обучения и тренировки руководящих работников, обеспечивая новых служащих обширным багажом опыта и стратегий, по которым можно изучать рекомендуемую политику и методы.
  + 1. **Искусственные нейронные сети**

Искусственная нейронная сеть (ИНС) — [математическая модель](https://ru.wikipedia.org/wiki/%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_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C), а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования [биологических нейронных сетей](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C)  — сетей [нервных клеток](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD) живого организма.

Идея нейронных сетей была позаимствована у природы, а точнее в качестве аналога использовалась нервная система животных и человека. ИНС представляют собой [систему](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) соединённых и взаимодействующих между собой простых процессоров ([искусственных нейронов](https://ru.wikipedia.org/wiki/%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%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD)). Такие процессоры обычно довольно просты (особенно в сравнении с процессорами, используемыми в персональных компьютерах). Каждый процессор подобной сети имеет дело только с [сигналами](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB), которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И, тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие локально простые процессоры вместе способны выполнять довольно сложные задачи.

Нейросетевые технологии призваны решать трудноформализуемые задачи, к которым, в частности, сводятся многие проблемы медицины. В первую очередь это связано с тем, что исследователю часто предоставлено большое количество разнородного фактического материала, для которого еще не создана математическая модель. Кроме того, необходимо представлять результаты анализа в виде, понятном специалисту прикладной области.

Одним из наиболее удобных инструментов для решения подобных задач являются искусственные нейронные сети – мощный и одновременно гибкий метод имитации процессов и явлений. Нейронные сети бывают разные по структуре и по форме, однако у них есть несколько общих черт.

Основу каждой нейронной сети составляют простые элементы, называемые искусственными нейронами, которые имитируют работу нейронов головного мозга. Отличительное свойство нейронных сетей состоит в их способности обучаться на основе экспериментальных данных предметной области.

Применительно к медицинской тематике экспериментальные данные представляются в виде множества исходных признаков или параметров объекта и поставленного на основе них диагноза.

Обучение нейронной сети представляет собой интерактивный процесс, в ходе которого нейронная сеть находит скрытые нелинейные зависимости между исходными параметрами и конечным диагнозом, а также оптимальную комбинацию весовых коэффициентов нейронов, при которой погрешность определения класса образа стремится к минимуму. В процессе обучения («с учителем») на вход нейронной сети подается последовательность исходных параметров наряду с диагнозами, которые эти параметры характеризуют.

С точки зрения машинного обучения, нейронная сеть представляет собой частный случай методов распознавания образов, дискриминантного анализа, методов кластеризации и т. п. С математической точки зрения, обучение нейронных сетей — это многопараметрическая задача нелинейной оптимизации. С точки зрения кибернетики, нейронная сеть используется в задачах адаптивного управления и как алгоритмы для робототехники. С точки зрения развития вычислительной техники и программирования, нейронная сеть — способ решения проблемы эффективного параллелизма. А с точки зрения искусственного интеллекта, ИНС является основой философского течения коннективизма и основным направлением в структурном подходе по изучению возможности построения (моделирования) естественного интеллекта с помощью компьютерных алгоритмов.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения — одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что, в случае успешного обучения, сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искаженных данных.

* + 1. **Параллели ИНС из биологии**

Нейронные сети возникли из исследований в области искусственного интеллекта, а именно, из попыток воспроизвести способность биологических нервных систем обучаться и исправлять ошибки, моделируя низкоуровневую структуру мозга (Patterson, 1996). Основной областью исследований по искусственному интеллекту в 60-е - 80-е годы были экспертные системы. Такие системы основывались на высокоуровневом моделировании процесса мышления (в частности, на представлении, что процесс нашего мышления построен на манипуляциях с символами). Скоро стало ясно, что подобные системы, хотя и могут принести пользу в некоторых областях, не ухватывают некоторые ключевые аспекты человеческого интеллекта. Согласно одной из точек зрения, причина этого состоит в том, что они не в состоянии воспроизвести структуру мозга. Чтобы создать искусственных интеллект, необходимо построить систему с похожей архитектурой.

Мозг состоит из очень большого числа (приблизительно 10,000,000,000) нейронов, соединенных многочисленными связями (в среднем несколько тысяч связей на один нейрон, однако это число может сильно колебаться). Нейроны - это специальная клетки, способные распространять электрохимические сигналы. Нейрон имеет разветвленную структуру ввода информации (дендриты), ядро и разветвляющийся выход (аксон). Аксоны клетки соединяются с дендритами других клеток с помощью синапсов. При активации нейрон посылает электрохимический сигнал по своему аксону. Через синапсы этот сигнал достигает других нейронов, которые могут в свою очередь активироваться. Нейрон активируется тогда, когда суммарный уровень сигналов, пришедших в его ядро из дендритов, превысит определенный уровень (порог активации).

Интенсивность сигнала, получаемого нейроном (а, следовательно, и возможность его активации), сильно зависит от активности синапсов. Каждый синапс имеет протяженность, и специальные химические вещества передают сигнал вдоль него. Один из самых авторитетных исследователей нейросистем, Дональд Хебб, высказал постулат, что обучение заключается в первую очередь в изменениях "силы" синаптических связей. Например, в классическом опыте Павлова, каждый раз непосредственно перед кормлением собаки звонил колокольчик, и собака быстро научилась связывать звонок колокольчика с пищей. Синаптические связи между участками коры головного мозга, ответственными за слух, и слюнными железами усилились, и при возбуждении коры звуком колокольчика у собаки начиналось слюноотделение.

Таким образом, будучи построен из очень большого числа совсем простых элементов (каждый из которых берет взвешенную сумму входных сигналов и в случае, если суммарный вход превышает определенный уровень, передает дальше двоичный сигнал), мозг способен решать чрезвычайно сложные задачи. Разумеется, мы не затронули здесь многих сложных аспектов устройства мозга, однако интересно то, что искусственные нейронные сети способны достичь замечательных результатов, используя модель, которая ненамного сложнее, чем описанная выше.

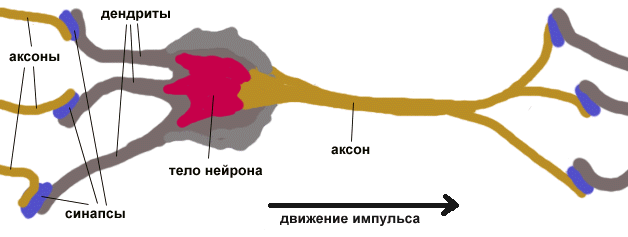


Рис. 1.1. Нейрон

* + 1. **Выбор методов применяемых для постановки диагнозов**

В процессе ознакомления с существующими методами машинного обучения было выявлено одно из основных отличий систем распознавания (нейронных сетей) от экспертных систем и состоит оно в том, что в нейронных сетях, распознаваемый образ представляется набором параметров, в котором перед началом анализа входных данных системой все значения известны, а в экспертных системах процесс распознавания совмещен с процессом получения информации об объекте, причем последовательность получения значений признаков меняется в зависимости от полученных ранее данных, т.е. процесс распознания объекта является пошаговым.

После общения с врачами-диагностами и совместного анализа существующих вышеописанных решений, которые позволяют решить заданную проблему о составления алгоритма для постановки диагноза пациенту информационной системой, стало понятно что анализ не привел к однозначному ответу на вопрос – «Что применять лучше и какой метод эффективнее?», было решено остановиться на нейронных сетях, в пользу того что сформировать базу данных для нейронных сетей показалось проще и как результат данные в ней более прозрачны для пользователя и постараться прийти к ответу на поставленный вопрос в процессе работы системы, а также в процессе анализа ее результатов по истечении некоторого времени эксплуатации.

Для распределения симптоматики по группам в конечном итоге понятным пользователям информационной системы, было решено использовать многоуровневую структуру, которая будет состоять из систем (нервная, дыхательная и т.д.), вложенных частей тела (Голова, ноги, руки и т.д.) и более низким уровнем являются сами симптомы, которые могут иметь дочерние симптомы, тем самым решается вопрос об уровнях вложенности, их может быть бесчисленное количество. Составлением базы знаний о симптоматике и распределением симптоматики по группам, а также обучением нейронной сети будут заниматься сами врачи, а база диагнозов будет составлена автоматизированным способом (парсингом официального информационного ресурса) на основе МКБ – 10.

* + 1. **Описание моделей нейронных сетей**

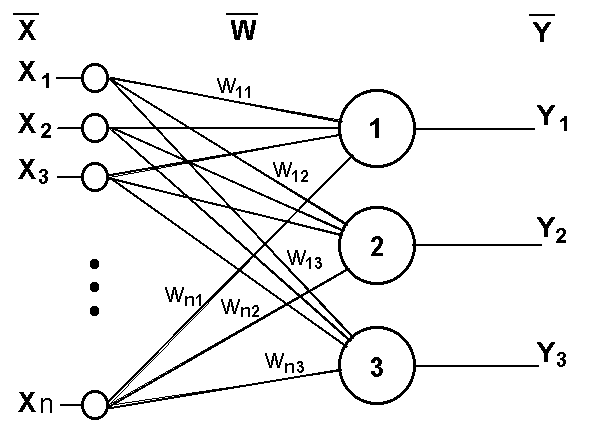
Рассмотрим наиболее простые модели нейронных сетей: однослойный и многослойный персептрон.

Большое количество моделей персептрона рассмотрено в основополагающей работе Розенблатта. Простейшая модель нейронной сети - однослойный персептрон.

Однослойный персептрон (персептрон Розенблатта) - однослойная нейронная сеть, все нейроны которой имеют жесткую пороговую функцию активации.

Однослойный персептрон имеет простой алгоритм обучения и способен решать лишь самые простые задачи. Эта модель вызвала к себе большой интерес в начале 1960-х годов и стала толчком к развитию искусственных нейронных сетей.

Классический пример такой нейронной сети - однослойный трехнейронный персептрон - представлен на рис. 1.2.

  
Рис. 1.2. Однослойный персептрон

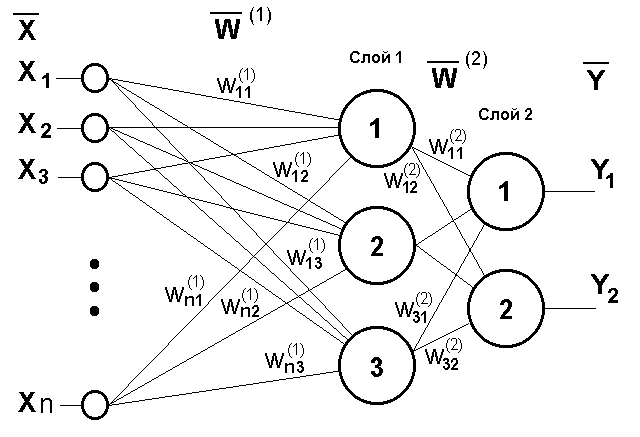
Сеть, изображенная на рисунке, имеет n входов, на которые поступают сигналы, идущие по синапсам на 3 нейрона. Эти три нейрона образуют единственный слой данной сети и выдают три выходных сигнала.

Многослойный персептрон (MLP) - нейронная сеть прямого распространения сигнала (без обратных связей), в которой входной сигнал преобразуется в выходной, проходя последовательно через несколько слоев.

Первый из таких слоев называют входным, последний - выходным. Эти слои содержат так называемые вырожденные нейроны и иногда в количестве слоев не учитываются. Кроме входного и выходного слоев, в многослойном персептроне есть один или несколько промежуточных слоев, которые называют скрытыми.

В этой модели персептрона должен быть хотя бы один скрытый слой. Присутствие нескольких таких слоев оправдано лишь в случае использования нелинейных функций активации.

Пример двухслойного персептрона представлен на рис. 1.3.

  
Рис. 1.3. Двухслойный персептрон

Сеть, изображенная на рисунке, имеет n входов. На них поступают сигналы, идущие далее по синапсам на 3 нейрона, которые образуют первый слой. Выходные сигналы первого слоя передаются двум нейронам второго слоя. Последние, в свою очередь, выдают два выходных сигнала.

* + 1. **Классификация нейронных сетей**

Классификация нейронных сетей по характеру обучения делит их на:

* нейронные сети, использующие обучение с учителем;
* нейронные сети, использующие обучение без учителя.

Нейронные сети, использующие обучение с учителем: обучение с учителем, предполагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они называются обучающей парой. Обычно сеть обучается на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором. Далее веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемого уровня.

Нейронные сети, использующие обучение без учителя: обучение без учителя, является намного более правдоподобной моделью обучения с точки зрения биологических корней искусственных нейронных сетей. Развитая Кохоненом и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с предопределенными идеальными ответами. Обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы.

Классификация нейронных сетей по настройке весов делит их на:

* сети с фиксированными связями – весовые коэффициенты нейронной сети выбираются сразу, исходя из условий задачи;
* сети с динамическими связями – для них в процессе обучения происходит настройка синаптических весов.

Классификация нейронных сетей типу входной информации делит их на:

* аналоговая – входная информация представлена в форме действительных чисел;
* двоичная – вся входная информация в таких сетях представляется в виде нулей и единиц.

Модели искусственных нейронных сетей бывают следующих типов:

Сети прямого распространения – все связи направлены строго от входных нейронов к выходным. К таким сетям относятся, например: простейший персептрон (разработанный Розенблаттом) и многослойный персептрон описанные в предыдущем разделе.

Реккурентные нейронные сети – сигнал с выходных нейронов или нейронов скрытого слоя частично передается обратно на входы нейронов входного слоя.

Радиально базисные функции – вид нейронной сети, имеющий скрытый слой из радиальных элементов и выходной слой из линейных элементов. Сети этого типа довольно компактны и быстро обучаются. Предложены в работах Broomhead and Lowe (1988) и Moody and Darkin (1989). Радиально базисная сеть обладает следующими особенностями: один скрытый слой, только нейроны скрытого слоя имеют нелинейную активационную функцию и синаптические веса входного и скрытого слоев равны единицы.

Самоорганизующиеся карты или Сети Кохонена – такой класс сетей, как правило, обучается без учителя и успешно применяется в задачах распознавания. Сети такого класса способны выявлять новизну во входных данных: если после обучения сеть встретится с набором данных, непохожим ни на один из известных образцов, то она не сможет классифицировать такой набор и тем самым выявит его новизну. Сеть Кохонена имеет всего два слоя: входной и выходной, составленный из радиальных элементов.

Более схематичное представление нейронной сети можно увидеть на рис. 1.2. Диагноз n – представляет собой диагноз, а круг внутри, вес симптома в диагнозе, чем больше диаметр круг, тем выше значение весового коэффициента данного симптома для диагноза.



Рис. 1.4. Cхема нейронной сети

* + 1. **Описание алгоритма работы нейронной сети**

Все ИНС представляют собой [систему](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) соединённых и взаимодействующих между собой простых процессоров ([искусственных нейронов](https://ru.wikipedia.org/wiki/%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%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD)). Такие процессоры обычно довольно просты (особенно в сравнении с процессорами, используемыми в персональных компьютерах). Каждый процессор подобной сети имеет дело только с [сигналами](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB), которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И, тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие локально простые процессоры вместе способны выполнять довольно сложные задачи.

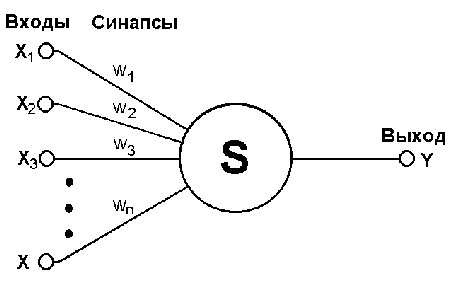


Рис. 1.2. Искусственный нейрон

* Нейрон обладает группой синапсов - однонаправленных входных связей, соединенных с выходами других нейронов. Каждый синапс характеризуется величиной синоптической связи или ее весом wi.
* Каждый нейрон имеет текущее состояние, которое обычно определяется, как взвешенная сумма его входов:

Нейрон имеет аксон - выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Выход нейрона есть функция его состояния.

Множество всех нейронов искусственной нейронной сети можно разделить на подмножества - т.е. слои. Взаимодействие нейронов происходит послойно.

Слой искусственной нейронной сети - это множество нейронов на которые в каждый такт времени параллельно поступают сигналы от других нейронов данной сети

Выбор архитектуры искусственной нейронной сети определяется задачей. Для некоторых классов задач уже существуют оптимальные конфигурации. Если же задача не может быть сведена ни к одному из известных классов, разработчику приходится решать задачу синтеза новой конфигурации. Проблема синтеза искусственной нейронной сети сильно зависит от задачи, дать общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант искусственной нейронной сети получается опытным путем.

Алгоритм, представленный далее, применим к нейронной сети с одним скрытым слоем, что является отличным решением для задачи, поставленной в работе, т.к. в разрабатываемом решении предполагается вынести всего один скрытый слой нейронов, который будет представлять диагнозы, входной сигнал будет представлять собой набор симптомов, а выходной ответ сети, будет представлять собой правильный диагноз.

Как уже было сказано ранее, обучение сети включает в себя три стадии: подача на входы сети обучающих данных, обратное распространение ошибки и корректировка весов. В ходе первого этапа каждый входной нейрон  получает сигнал и широковещательно транслирует его каждому из скрытых нейронов . Каждый скрытый нейрон затем вычисляет результат его активационной функции (сетевой функции) и рассылает свой сигнал  всем выходным нейронам. Каждый выходной нейрон , в свою очередь, вычисляет результат своей активационной функции , который представляет собой ничто иное, как выходной сигнал данного нейрона для соответствующих входных данных. В процессе обучения, каждый нейрон на выходе сети сравнивает вычисленное значение  с предоставленным учителем  ( целевым значением ), определяя соответствующее значение ошибки для данного входного шаблона. На основании этой ошибки вычисляется .  используется при распространении ошибки от  до всех элементов сети предыдущего слоя ( скрытых нейронов, связанных с  ), а также позже при изменении весов связей между выходными нейронами и скрытыми. Аналогичным образом вычисляется  для каждого скрытого нейрона . Несмотря на то, что распространять ошибку до входного слоя необходимости нет,  используется для изменения весов связей между нейронами скрытого слоя и входными нейронами. После того как все   были определены, происходит одновременная корректировка весов всех связей.

**В алгоритме обучения сети используются следующие обозначения:**

x Входной вектор обучающих данных .

t Вектор целевых выходных значений, предоставляемых учителем .

 Составляющая корректировки весов связей  , соответствующая ошибке выходного нейрона ;

Информация об ошибке нейрона , которая распространяется тем нейронам скрытого слоя, которые связаны с .

 Составляющая корректировки весов связей24a23e82102f3b8b5145fe44d657ce1d, соответствующая распространяемой от выходного слоя к скрытому нейрону  информации об ошибке.

 Нейрон на входе с индексом i. Для входных нейронов входной и выходной сигналы одинаковы — .

 Скрытый нейрон j;

Суммарное значение подаваемое на вход скрытого элемента  обозначается   и

Сигнал на выходе (результат применения к  активационной функции) обозначается :

 Нейрон на выходе под индексом k; Суммарное значение подаваемое на вход выходного элемента  обозначается  :  .

Сигнал на выходе  (результат применения к  активационной функции) обозначается :

Функция активации в алгоритме обратного распространения ошибки должна обладать несколькими важными характеристиками: непрерывностью, дифференцируемостью и являться монотонно неубывающей. Более того, ради эффективности вычислений, желательно, чтобы ее производная легко находилась. Зачастую, активационная функция также является функцией с насыщением. Одной из наиболее часто используемых активационных функций является бинарная сигмоидальная функция с областью значений в (0, 1) и определенная как:

Другой широко распространенной активационной функцией является биполярный сигмоид с областью значений (-1, 1) и определенный как:

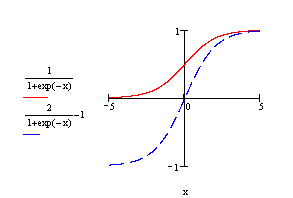


Рис. 1.4. Функция активации

**Алгоритм обучения выглядит следующим образом:**

Инициализация весов (веса всех связей инициализируются случайными небольшими значениями).

До тех пор, пока условие прекращения работы алгоритма неверно, выполняются нижеследующие шаги.

Для каждой пары {данные, целевое значение} выполняются шаги ниже следующие шаги.

**Распространение данных от входов к выходам:**

Каждый входной нейрон   отправляет полученный сигнал  всем нейронам в следующем слое ( скрытом ).

Каждый скрытый нейрон    суммирует взвешенные входящие сигналы:  и применяет активационную функцию:   После чего посылает результат всем элементам следующего слоя ( выходного ).

Каждый выходной нейрон    суммирует взвешенные входящие сигналы:   и применяет активационную функцию, вычисляя выходной сигнал:

**Обратное распространение ошибки:**

Каждый выходной нейрон  получает целевое значение — то выходное значение, которое является правильным для данного входного сигнала, и вычисляет ошибку:, так же вычисляет величину, на которую изменится вес связи : . Помимо этого, вычисляет величину корректировки смещения:  и посылает  нейронам в предыдущем слое.

Каждый скрытый нейрон   суммирует входящие ошибки (от нейронов в последующем слое)     и вычисляет величину ошибки, умножая полученное значение на производную активационной функции: , , так же вычисляет величину, на которую изменится вес связи : .

**Изменение весов:**

Каждый выходной нейрон  изменяет веса своих связей с элементом смещения и скрытыми нейронами:  Каждый скрытый нейрон  изменяет веса своих связей с элементом смещения и выходными нейронами: :

**Проверка условия прекращения работы алгоритма:**

Условием прекращения работы алгоритма может быть, как достижение суммарной квадратичной ошибкой результата на выходе сети предустановленного заранее минимума в ходе процесса обучения, так и выполнения определенного количества итераций алгоритма. В основе алгоритма лежит метод под названием градиентный спуск. В зависимости от знака, градиент функции (в данном случае значение функции — это ошибка, а параметры — это веса связей в сети) дает направление, в котором значения функции возрастают (или убывают) наиболее стремительно.

В разрабатываемой системе принято разработать нейронную систему на основе однослойного прецептрона, активационной функцией принята бинарная сигмоидальная функция и методом обучения на основе анализа сбора входных данных для начального обучения нейронной сети и обучения сети в будущем было принято выбрать метод обучения «с учителем». Входными нейронами которой являются симптомы, которые с определенными весами, связаны с каждым нейроном из скрытого слоя, которые являются диагнозами. Нейронная сеть спроектирована так, что имеет два возможных выхода. Первым является одиночный нейрон, который является ответом для сети, он представляет собой диагноз, который по расчетам сети оказался самым сильным, на входных параметрах. И второй выход представляет собой набор нейронов у каждого из которых определен свой вес или так называемая сила(power) нейрона, которая формируется на основе входного импульса сети.

* 1. **Обучение нейронной сети**

Обучить нейронную сеть - значит, сообщить ей, чего мы от нее добиваемся. Этот процесс очень похож на обучение ребенка алфавиту. Показав ребенку изображение буквы "А", мы спрашиваем его: "Какая это буква?" Если ответ неверен, мы сообщаем ребенку тот ответ, который мы хотели бы от него получить: "Это буква А". Ребенок запоминает этот пример вместе с верным ответом, то есть в его памяти происходят некоторые изменения в нужном направлении. Мы будем повторять процесс предъявления букв снова и снова до тех пор, когда все 33 буквы будут твердо запомнены. Такой процесс называют "обучение с учителем".

При обучении нейронной сети мы действуем совершенно аналогично. У нас имеется некоторая база данных, содержащая примеры (набор рукописных изображений букв). Предъявляя изображение буквы "А" на вход нейронной сети, мы получаем от нее некоторый ответ, не обязательно верный. Нам известен и верный (желаемый) ответ - в данном случае нам хотелось бы, чтобы на выходе нейронной сети с меткой "А" уровень сигнала был максимален. Обычно в качестве желаемого выхода в задаче классификации берут набор (1, 0, 0, ...), где 1 стоит на выходе с меткой "А", а 0 - на всех остальных выходах. Вычисляя разность между желаемым ответом и реальным ответом сети, мы получаем 33 числа - *вектор ошибки*. Алгоритм обратного распространения ошибки - это набор формул, который позволяет по вектору ошибки вычислить требуемые поправки для весов нейронной сети. Одну и ту же букву (а также различные изображения одной и той же буквы) мы можем предъявлять нейронной сети много раз. В этом смысле обучение скорее напоминает повторение упражнений в спорте - тренировку.

Оказывается, что после многократного предъявления примеров веса нейронной сети стабилизируются, причем нейронная сеть дает правильные ответы на все (или почти все) примеры из базы данных. В таком случае говорят, что "нейронная сеть выучила все примеры", "нейронная сеть обучена", или "нейронная сеть натренирована". В программных реализациях можно видеть, что в процессе обучения величина ошибки постепенно уменьшается. Когда величина ошибки достигает нуля или приемлемого малого уровня, тренировку останавливают, а полученную нейронную сеть считают натренированной и готовой к применению на новых данных.   
    Важно отметить, что вся информация, которую нейронная сеть имеет о задаче, содержится в наборе примеров. Поэтому качество обучения нейронной сети напрямую зависит от количества примеров в обучающей выборке, а также от того, насколько полно эти примеры описывают данную задачу. Так, например, бессмысленно использовать нейронную сеть для предсказания финансового кризиса, если в обучающей выборке кризисов не представлено. Считается, что для полноценной тренировки нейронной сети требуется хотя бы несколько десятков (а лучше сотен) примеров.

Повторим еще раз, что обучение нейронных сетей - сложный и наукоемкий процесс. Алгоритмы обучения нейронных сетей имеют различные параметры и настройки, для управления которыми требуется понимание их влияния.

После того, как нейронная сеть обучена, мы можем применять ее для решения полезных задач. Важнейшая особенность человеческого мозга состоит в том, что, однажды обучившись определенному процессу, он может верно действовать и в тех ситуациях, в которых он не бывал в процессе обучения. Например, мы можем читать почти любой почерк, даже если видим его первый раз в жизни. Так же и нейронная сеть, грамотным образом обученная, может с большой вероятностью правильно реагировать на новые, не предъявленные ей ранее данные. Например, мы можем нарисовать букву "А" другим почерком, а затем предложить нашей нейронной сети классифицировать новое изображение. Веса обученной нейронной сети хранят достаточно много информации о сходстве и различиях букв, поэтому можно рассчитывать на правильный ответ и для нового варианта изображения.

В разрабатываемом решении было принято, что обучение будет проводиться врачами, после установки верного диагноза, на основе введенных данных пользователем при начале диагностирования. Тем самым решаются проблемы не только обучения нейронной сети, но и постоянная корректировка весов, а также, сеть обучаться воспринимать ошибки ввода данных пользователем, т.к. естественные симптомы могут восприниматься пациентом, за совершенно другие. После общения с врачами был выявлен и тот факт, что большинство врачей поддерживают развитие данной системы и инновационных технологий в диагностировании заболеваний впринципе и готовы не только работать в системе, но и способствовать ее развитию. Для удобства использования системы врачами, был решено перенести диагнозы в базу данных информационной системы из международной классификации болезней (МКБ-10). Что позволит осуществлять быстрый поиск диагноза по коду болезни или по его утвержденному мировым сообществом врачей названием.

* 1. **Технология обработки информации**

Разрабатываемое решение представляет собой облачный сервис (сервис для работы в сети интернет), 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 выбора симптомов пациентом (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;
* Также, после составления симптоматики связаться с одним из предоставленных ему врачей для уточнения симптоматики и онлайн консультации;
* Пользователь может записать 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.

Разрабатываемая информационная система решает большое количество важных задач, связанных с врачебной деятельностью:

* Сбор первичных сведений о заболевании;
* Постановка первичного диагноза;
* Хранение и сбор данных о заболеваниях пациента в онлайн медицинской карте;
* Осуществление удобной коммуникация с пациентами;

Все эти действия занимают не мало времени и сил на их осуществление, поэтому нуждаются в автоматизации.

Доступ к различным интерфейсам осуществляется через интерфейс авторизации пользователей с помощью логина и пароля, для которых могут быть определены следующие роли:

Таблица 1.1

|  |  |
| --- | --- |
| **Пользователя** | **Действия** |
| Администатор | Администратору доступны все действия в системе. |
| Врач | Его привилегией является общение с пациентами, ведение амбулаторных карт. |
| Пользователь | Его привилегией является описание симптоматики заболевания, общение с врачами, просмотр информации о лечебных заведениях и отзывов о них.  Просмотр предварительного диагноза и просмотр своей амбулаторной карты. |

Одному пользователю может соответствовать одновременно несколько ролей, например:

* Администратор - Врач - Пользователь;
* Врач - Пользователь;
* Администратор - Пользователь;

При этом должны автоматически определяться все роли пользователя и формироваться список всех личных интерфейсов, соответствующих его ролям. Для каждый роли, соответствует её основной интерфейс, куда должен попасть пользователь при авторизации в системе. Если у пользователя несколько ролей переход осуществляется на основной интерфейс роли с большим приоритетом.

Интерфейсы системы.

* Главная страница врача;
* Главная страница пациента;
* Административная панель;
* Сообщения (Чаты);
* Постановка диагноза;
* Кабинет врача;
* Кабинет пациента;
* Лечебное заведение;
* Амбулаторная карта;
* Апи.

Описание операций, реализованных в системе

Таблица 1.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№п/п** | **Операция** | **Действия** | **Входные данные** | **Предшествующие действия** |
| 1 | Вход под пользователем «Администратор» | Ввод логина и пароля администратора | Логин, пароль |  |
| 2 | Вход под пользователем «Врач» | Ввод логина и пароля врача | Логин, пароль |  |
| 3 | Вход под пользователем «Пользователь» | Ввод логина и пароля пользователя | Логин, пароль |  |
| 4 | Просмотр главной страницы врача | Переход по ссылке |  | 2 |
| 5 | Просмотр главной страницы пациента | Переход по ссылке |  | 3 |
| 6 | Выбор симптомов пациентом | Выбор системы и части тела из списка и выбор конкретного симптома | Симптомы | 3 |
| 7 | Просмотр предварительного диагноза пациентом | Нажатие на кнопку «Получить диагноз» |  | 1 |
| 8 | Просмотр предварительного диагноза врачом | Нажатие на кнопку «Получить диагноз» |  | 2 |
| 9 | Просмотр амбулаторной карты врачом | Выбор конкретного пациента и нажатие на кнопку «Просмотреть амбулаторную карту» |  | 8 |
| 10 | Просмотр амбулаторной карты пациентом | Нажатие на кнопку «Просмотреть амбулаторную карту» в личном кабинете |  | 3 |
| 11 | Редактирование амбулаторной карты врачом | Ввод информации об пациенте, его заболевании и сопутствующих симптомов | Информация о пациенте | 9 |
| 12 | Просмотр информации о враче пациентом | Переход на личный кабинет врача |  | 3 |
| 13 | Просмотр информации о лечебном заведении пациентом | Переход на подробное описание лечебного заведения |  | 3 |
| 14 | Ведение чата с врачом пациентом | Выбор диалога, оправка сообщения | Сообщение, получатель | 3 |
| 15 | Ведение чата с пациентом врачом | Выбор диалога, оправка сообщения | Сообщение, получатель | 2 |
| 16 | Просмотр списка систем | Переход в административную панель |  | 1,2 |
| 17 | Просмотр списка частей тела | Переход в административную панель |  | 1,2 |
| 18 | Просмотр списка симптомов | Переход в административную панель. |  | 1,2 |
| 19 | Просмотр списка диагнозов | Переход в административную панель |  | 1,2 |
| 20 | Создание системы | Переход в административную панель и ввод информации о системе | Информация о системе | 16 |
| 21 | Редактирование системы | Переход в административную панель и ввод информации о системе | Информация о системе | 16 |
| 22 | Удаление системы | Переход в административную панель и выбор системы для удаления |  | 16 |
| 23 | Создание части тела | Переход в административную панель и ввод информации о части тела | Информация о части тела | 17 |
| 24 | Редактирование части тела | Переход в административную панель и ввод информации о части тела | Информация о части тела | 17 |
| 25 | Удаление части тела | Переход в административную панель и выбор части тела для удаления |  | 17 |
| 26 | Создание симптома | Переход в административную панель и ввод информации о симптоме | Информация о симптоме | 18 |
| 27 | Редактирование симптома | Переход в административную панель и ввод информации о симптоме | Информация о симптоме | 18 |
| 28 | Удаление симптома | Переход в административную панель и выбор симптома для удаления |  | 18 |
| 29 | Создание диагноза | Переход в административную панель и ввод информации о диагнозе | Информация о диагнозе | 19 |
| 30 | Редактирование диагноза | Переход в административную панель и ввод информации о диагнозе | Информация о диагнозе | 19 |
| 31 | Удаление диагноза | Переход в административную панель и выбор диагноза для удаления |  | 19 |
| 32 | Обучние диагноза | Переход в административную панель, выбор диагноза и симптомов для его обучения | Информация о диагнозе и симптомах | 19 |
| 33 | Просмотр апи методов системы | Просмотр апи методов системы |  | 2 |
| 34 | Логирование всех вышеперечисленных действий | Любое действие в системе | Любые данные входящие в систему | Любое действие в системе |

Действия в системе осуществляются при помощи интуитивно понятных интерфейсов. Описание интерфейсов системы:

Таблица 1.3

|  |  |  |
| --- | --- | --- |
| **Интерфейс** | **Пользователи** | **Операции доступные в интерфейсе** |
| Главная страница врача | Администратор, Врач |  |
| Главная страница пациента | Администратор, Пациент |  |
| Административная панель | Администратор | Создание/изменение/удаление системы/части тела/ симптома/диагноза и обучение диагноза |
| Сообщения (Чаты) | Администратор, Врач, Пациент | Ввод и отправка сообщений, просмотр диалогов |
| Постановка диагноза | Администратор, Врач, Пациент | Выбор симптомов, постановка диагноза |
| Кабинет врача | Администратор, Врач | Просмотр пациентов, амбулаторных карт и изменение данных о себе |
| Кабинет пациента | Администратор, Врач, Пациент | Просмотр амбулаторной карты и изменение данных о себе |
| Лечебное заведение | Администратор, Врач, Пациент | Просмотр информации о себе |
| Амбулаторная карта | Администратор, Врач, Пациент | Просмотр информации о амбулаторной карты |
| Апи | Администратор | Просмотр методов апи |

* 1. **Входные данные**

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

Таблица 1.4

|  |  |  |
| --- | --- | --- |
| Информация о пользователе | | |
| **Название атрибута** | **Тип атрибута** | **Комментарий** |
| Имя | Текст | Имя пользователя |
| Фамилия | Текст | Фамилия пользователя |
| Отчество | Текст | Отчество пользователя |
| Роль | Число | Роль пользователя |
| Почта | Текст | Email пользователя |
| Пароль | Текст | Пароль пользователя |

Таблица 1.5

|  |  |  |
| --- | --- | --- |
| 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 1.6

|  |  |  |
| --- | --- | --- |
| 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 1.7

|  |  |  |
| --- | --- | --- |
| 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 1.8

|  |  |  |
| --- | --- | --- |
| 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 1.9

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

* 1. **Выходные документы**

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

* 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:

* Любое устройство где установлен один из браузеров Chrome, FireFox, IE выше 8 версии, Safari и т.д. (любой браузер с поддержкой HTML5) или мобильное устройство iPhone с версией ОС iOS выше 8;

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:

* 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 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 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.

# 2 РАБОЧИЙ ПРОЕКТ

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

Программный продукт 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.

* 1. **Функциональное назначение программного продукта**

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

Результатом разработки является система DocApp.

* 1. **Описание физической архитектуры системы**

Архитектура системы представлена на диаграмме развертывания (Приложение 1) и включает в себя следующие элементы:

* Бэкенд-сервер соответствуюший 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 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 база данных DocApp и реализована в среде MS SQL Server 2016.

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 DocApp входят следующие связанные таблицы (см. Приложение 3):

**AspNetRoles –** Содержит список ролей пользователей (таб. 2.1).

Таблица 2.1

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Char | Код роли (Первичный ключ) |
| [Name] | Char | Название роли |

**AspNetUsers –** Содержит список пользователей (таб. 2.2).

Таблица 2.2

| **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 |

**AspNetUserRoles -** Содержит список соответствий между пользователем и ролью (табл.2.3).

Таблица 2.3

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [UserId] | Char | Код пользователя (Первичный ключ) |
| [RoleId] | Char | Гражданство |

**BaseUsers -** Содержит информацию о пользователе системы(табл.2.4).

Таблица 2.4

| **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.2.5).

Таблица 2.5

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

**Doctors -** Содержит информацию о докторе (табл.2.6).

Таблица 2.6

| **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 |

**Dialogs -** Содержит информацию о диалоге (табл.2.7).

Таблица 2.7

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код диалога (Первичный ключ) |
| [FirstUserId] | Int | Идентификатор создателя диалога |
| [SecondUserId] | Int | Идентификатор пользователя с которым завели диалог |

**Recalls -** Содержит информацию об отзыве (табл.2.8).

Таблица 2.8

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код отзыва (Первичный ключ) |
| [CallerId] | Int | Идентификатор пользователя оставившего отзыв |
| [DoctorId] | Int | Идентификатор доктора о котором оставили отзыв |
| [HospitalId] | Int | Идентификатор лечебного заведения о котором оставили отзыв |

**Hospitals -** Содержит информацию о лечебном заведении (табл.2.9).

Таблица 2.9

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код лечебного заведения (Первичный ключ) |
| [Name] | Char | Название лечебного заведения |
| [Type] | Char | Тип лечебного заведения |

**Messages -** Содержит информацию о сообщении (табл.2.10).

Таблица 2.10

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код сообщения (Первичный ключ) |
| [Text] | Char | Текст сообщения |
| [DialogId] | Int | Идентификатор диалога которому принадлежит сообщение |

**Images -** Содержит информацию о изображении (табл.2.11).

Таблица 2.11

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код изображения (Первичный ключ) |
| [Name] | Char | Название изображения |
| [Bytes] | Byte[] | Данные изображения |
| [MessageId] | Int | Идентификатор сообщения к которому принадлежит изображение |

**Documents -** Содержит информацию о документе (табл.2.12).

Таблица 2.12

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код документа (Первичный ключ) |
| [Name] | Char | Название документа |
| [Bytes] | Byte[] | Данные документа |
| [MessageId] | Int | Идентификатор сообщения к которому принадлежит документ |

**Hrefs -** Содержит информацию о ссылке (табл.2.13).

Таблица 2.13

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код ссылки (Первичный ключ) |
| [Name] | Char | Название ссылки |
| [Link] | Char | Адрес ссылки |
| [MessageId] | Int | Идентификатор сообщения к которому принадлежит ссылка |

**OutpatientCards -** Содержит информацию о амбулаторной карте (табл.2.14).

Таблица 2.14

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

**Diseases -** Содержит информацию о болезни (табл.2.15).

Таблица 2.15

| **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.2.16).

Таблица 2.16

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

**Symptoms -** Содержит информацию о симптоме (табл.2.17).

Таблица 2.17

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код симптома (Первичный ключ) |
| [Name] | Char | Название симптома |
| [IsMale] | Bool | Принадлежность мужчине |
| [IsFemale] | Bool | Принадлежность женщине |
| [CoordX] | Int | Координата х на родительском изображении |
| [CoordY] | Int | Координата y на родительском изображении |
| [CoordR] | Int | Радиус на родительском изображении |
| [DieaseId] | Int | Идентификатор болезни которой принадлежит симптом |

**Groups -** Содержит информацию о части тела (табл.2.18).

Таблица 2.18

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код части тела (Первичный ключ) |
| [Name] | Char | Название части тела |
| [Image] | Byte[] | Информация изображения для мужчины |
| [IsMale] | Bool | Принадлежность мужчине |
| [IsFemale] | Bool | Принадлежность женщине |
| [CoordX] | Int | Координата х на родительском изображении |
| [CoordY] | Int | Координата y на родительском изображении |
| [CoordR] | Int | Радиус на родительском изображении |
| [ImageWoman] | Byte[] | Информация изображения для женщины |

**SymptomGroups -** Содержит информацию о связи между симптомом и частью тела (табл.2.19).

Таблица 2.19

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [SymptomId] | Int | Идентификатор симптома для связи многие ко многим |
| [GroupId] | Int | Идентификатор группы для связи многие ко многим |

**Departaments -** Содержит информацию о системе (табл.2.20).

Таблица 2.20

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код пользователя (Первичный ключ) |
| [Name] | Char | Название системы |
| [SmallImage] | Byte[] | Информация маленького изображения |
| [BigImage] | Byte[] | Информация изображения для мужчины |
| [IsMale] | Bool | Принадлежность мужчине |
| [IsFemale] | Bool | Принадлежность женщине |
| [BigImageWoman] | Byte[] | Информация изображения для женщины |

**GroupsDepartaments -** Содержит информацию о связи между частью тела и системой (табл.2.21).

Таблица 2.21

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [GroupId] | Int | Идентификатор симптома для связи многие ко многим |
| [DepartamentId] | Int | Идентификатор системы для связи многие ко многим |

**Weights -** Содержит информацию о весе симптома(связи) в диагнозе(нейроне) (табл.2.22).

Таблица 2.22

| **Параметр** | **Тип** | **Назначение** |
| --- | --- | --- |
| [Id] | Int | Код пользователя (Первичный ключ) |
| [Value] | Int | Вес симптома(связи) в диагнозе(нейроне) |
| [KeyId] | Int | Идентификатор симптома привязанного к диагнозу |
| [DiagnosId] | Int | Идентификатор диагноза |

* 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.

В случае если сайт размещается в Интернете, то изначально требуется зарегистрироваться на сайте хостинговой компании и получить URL. После этого необходимо скопировать все файлы системы в выделенный на сервере каталог и изменить в конфигурационных файлах системы настройки соединения с базой данных на выданные при регистрации настройки.

Пользователь получает доступ к сайту, используя браузер. Для доступа к системе пользователь должен набрать в адресной строке браузера URL web-сайта DocAppDemo. Дальнейшее взаимодействие пользователя с системой происходит с использованием гиперссылок и других интерактивных элементов.

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

Взаимодействие пользователя с сервисом может осуществляться на любом устройстве, с установленным браузером и какими-либо устройствами ввода (компьютерная мышь, клавиатура, сенсорный экран и т.д.) или с по средствам мобильного устройства на базе ОС iOS равное или выше 8 версии.

Система разработана так, что пользователь может комфортно работать с ней практически на любом устройстве с практически любым разрешением.

* 1. **Разработанные меню и интерфейсы**
     1. **Интерфейс "Домашняя страница"**

Роли: "Администратор", "Врач", "Пользователь".

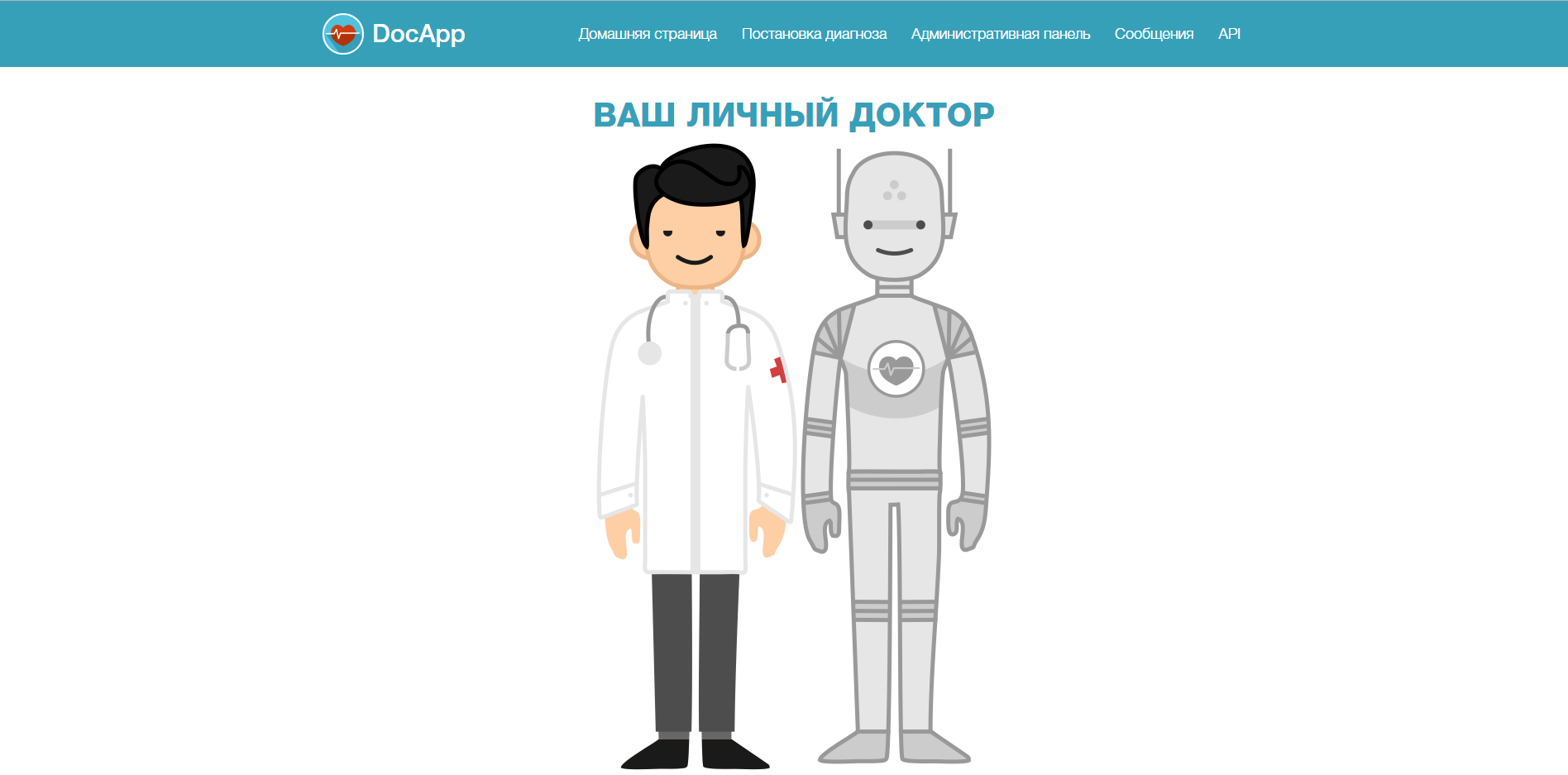


Рис. 2.2. Интерфейс " Домашняя страница "

Интерфейс предназначен для просмотра ознакомительной информации о системе, ее создателей и прохождения детального обучения по работе в системе

* + 1. **Интерфейс "Постановка диагноза для мужчины"**

Роли: "Администратор", "Врач", "Пользователь".

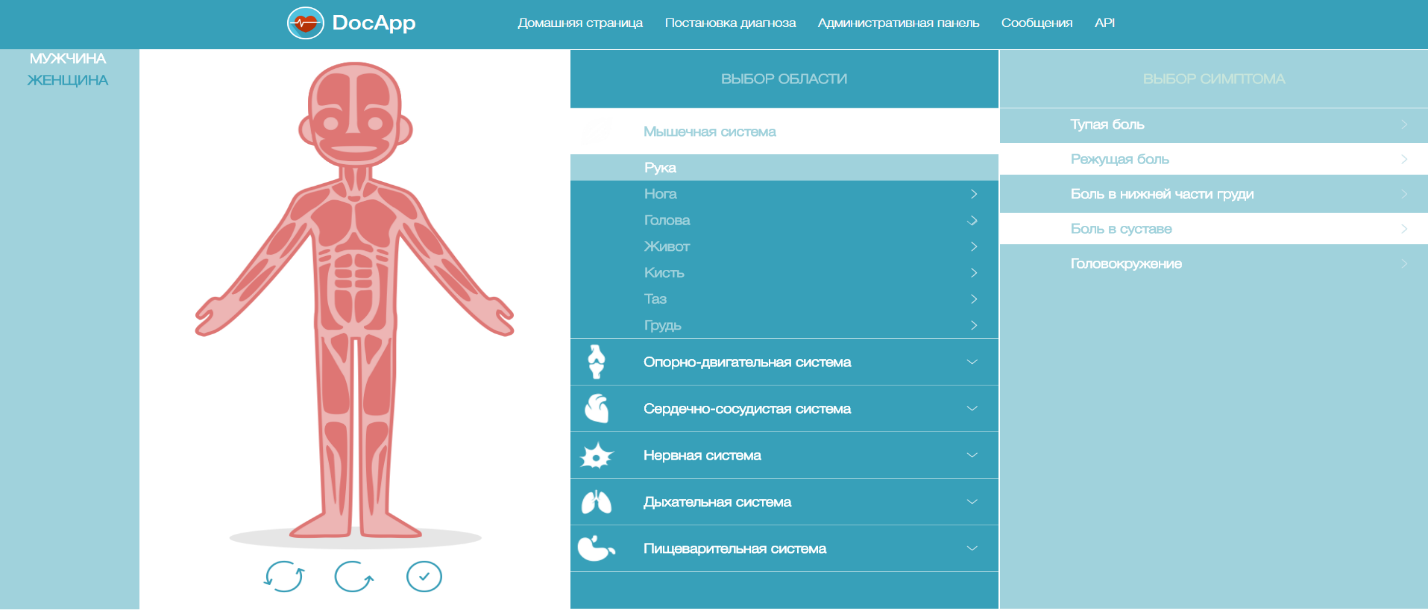


Рис. 2.3. Интерфейс " Постановка диагноза для мужчины "

Интерфейс предназначен для постановки предварительного диагноза пользователю мужчине на основе его входных данных. Для успешного результата пользователь должен выбрать систему, часть тела и выбрать симптомы которые его беспокоят. После чего пользователь должен нажать на кнопку «Поставить диагноз» (см. Приложение 2).

* + 1. **Интерфейс "Постановка диагноза для мужчины"**

Роли: "Администратор", "Врач", "Пользователь".

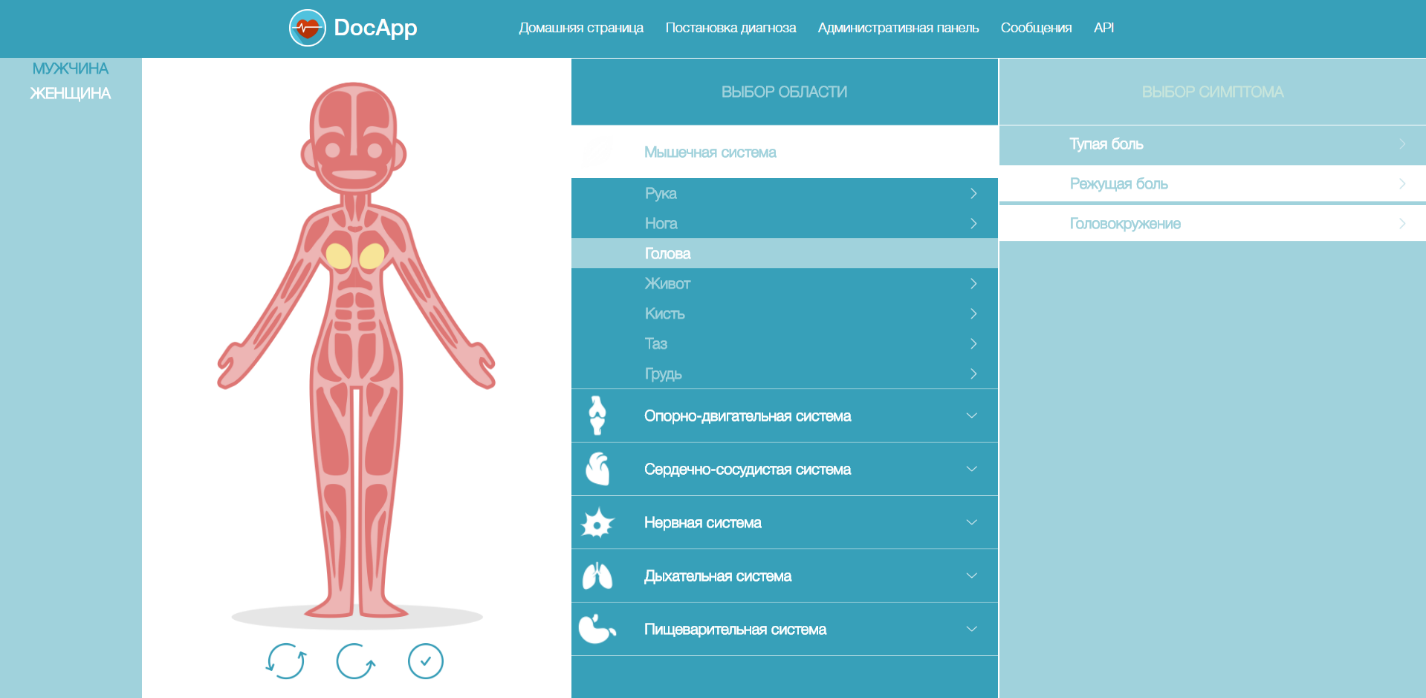


Рис. 2.4. Интерфейс " Постановка диагноза для мужчины "

Интерфейс предназначен для постановки предварительного диагноза пользователю женщине на основе его входных данных. Для успешного результата пользователь должен выбрать систему, часть тела и выбрать симптомы которые его беспокоят. После чего пользователь должен нажать на кнопку «Поставить диагноз». Симптомы и части тела для женщины могут отличаться от симптомов и частей тела мужчины (см. Приложение 2).

* + 1. **Интерфейс "Просмотр диагноза"**

Роли: "Администратор", "Врач", "Пользователь".



Рис. 2.5. Интерфейс " Просмотр диагноза"

Интерфейс предназначен для просмотра результата после успешного выполнения действий на экране рис.2.3 или рис.2.4. На экране пользователю предоставляется список из 10-ти более вероятных диагнозов с их весами и описанием (см. Приложение 2).

* + 1. **Интерфейс "Административная панель"**

Роли: "Администратор", "Врач".

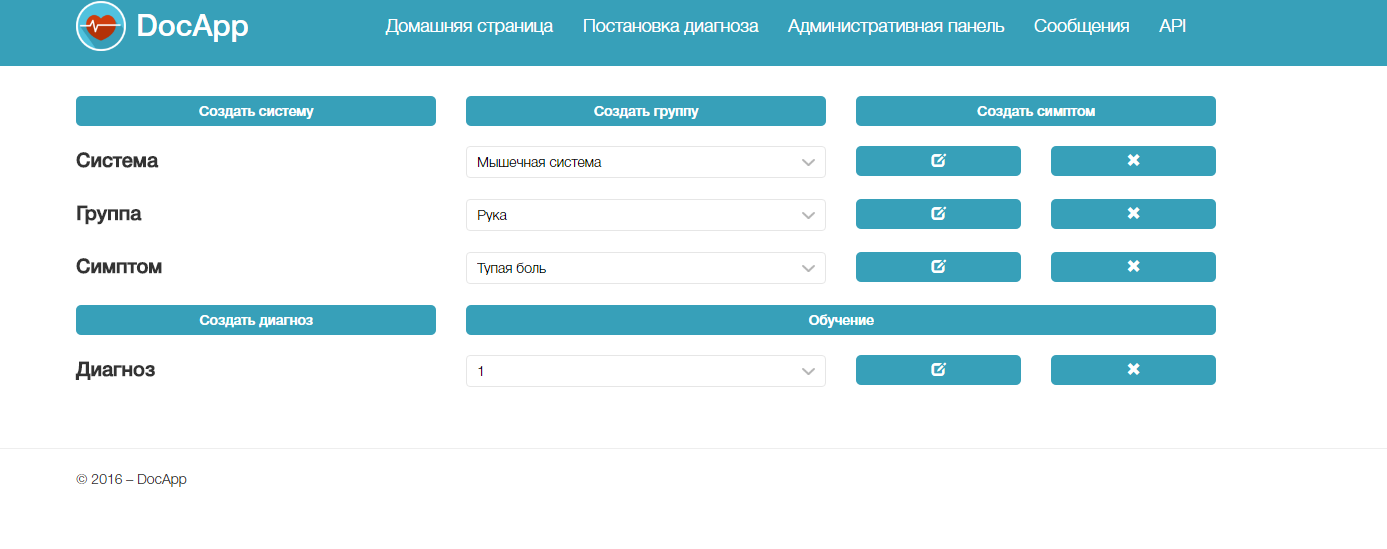


Рис. 2.6. Интерфейс " Административная панель "

Интерфейс предназначен для просмотра информации по существующим системам, частям тела, симптомам и диагнозам. С возможность выбора действия по каждому из них. Интерфейс позволяет выбрать такие действия как создание, изменение и удаление системы, части тела, симптома, диагноза, а также обучение диагноза.

* + 1. **Интерфейс "Создание изменения систем"**

Роли: "Администратор", "Врач".

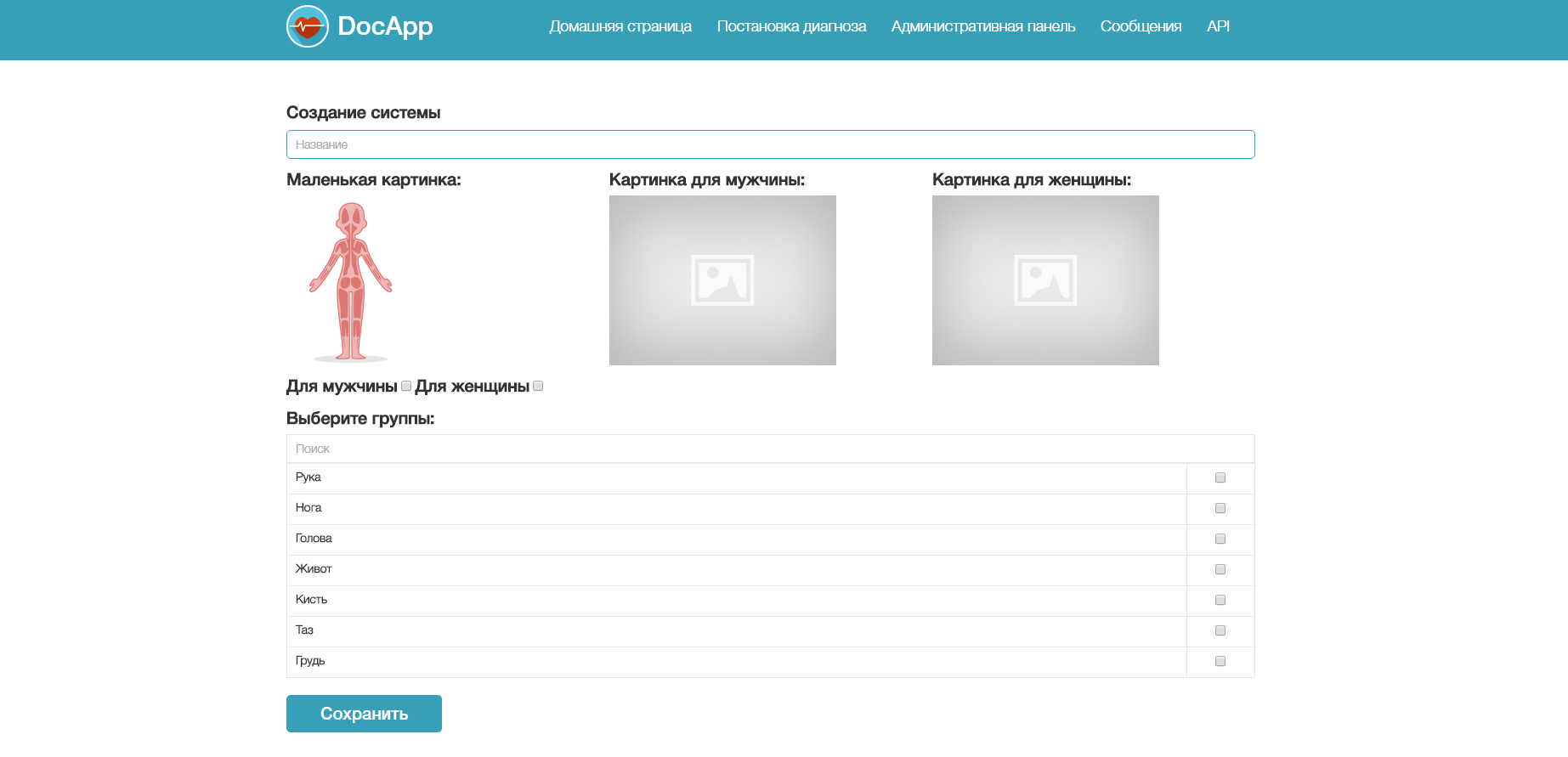


Рис. 2.7. Интерфейс " Создание изменения систем"

Интерфейс предназначен для создания или изменения информации по системам. В данном интерфейсе, можно изменить название системы, ее изображения, выбрать фильтр по полу, а также выбрать группы, которые будут входить в заданную систему. Поиск фильтрует части тела по названию. При нажатии на кнопку сохранить происходит сохранение введенной пользователем данных и переход на экран «Административная панель».

* + 1. **Интерфейс "Создание изменения частей тела"**

Роли: "Администратор", "Врач".

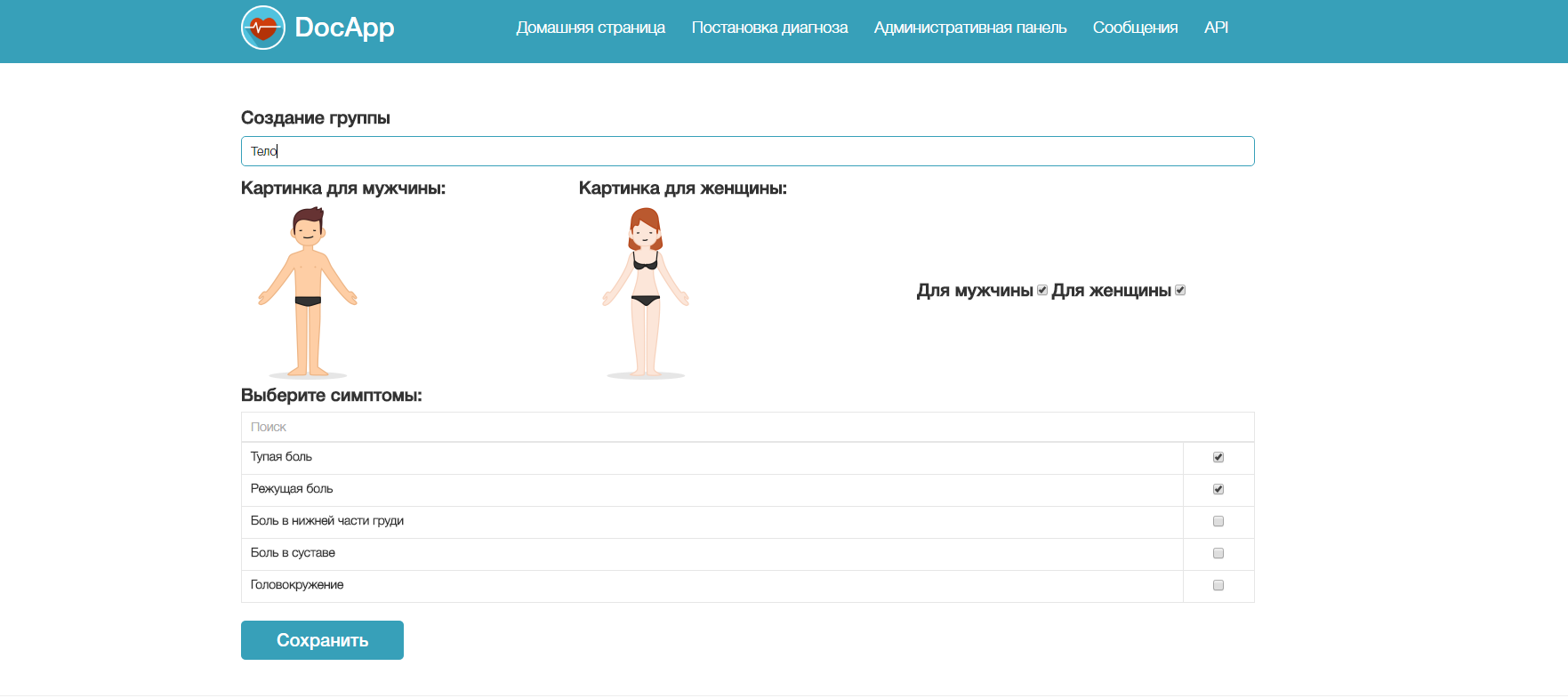


Рис. 2.8. Интерфейс " Создание изменения частей тела"

Интерфейс предназначен для создания или изменения информации по частям тела. В данном интерфейсе, можно изменить название части тела, ее изображения, выбрать фильтр по полу, а также симптомы, которые будут входить в заданную часть тела. Поиск фильтрует симптомы по названию. При нажатии на кнопку сохранить происходит сохранение введенной пользователем данных и переход на экран «Административная панель».

* + 1. **Интерфейс "Создание изменение симптомов"**

Роли: "Администратор", "Врач".

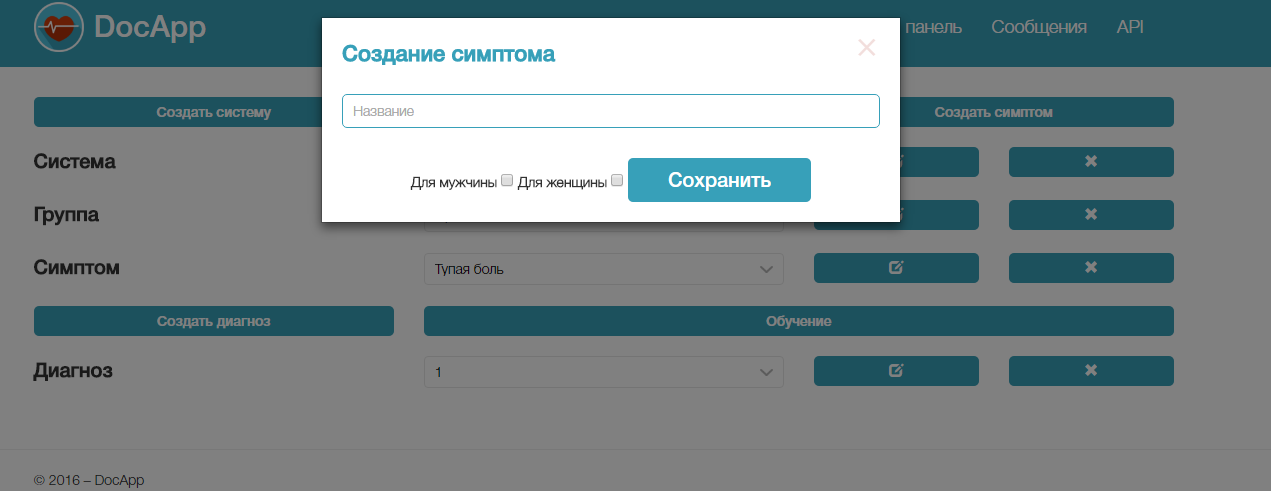


Рис. 2.9. Интерфейс " Создание изменение симптомов "

Интерфейс предназначен для создания или изменения информации по симптомам. В данном интерфейсе, можно изменить название симптома, выбрать фильтр по полу. При нажатии на кнопку сохранить происходит сохранение введенной пользователем данных и переход на экран «Административная панель».

* + 1. **Интерфейс "Создание изменение диагнозов"**

Роли: "Администратор", "Врач".

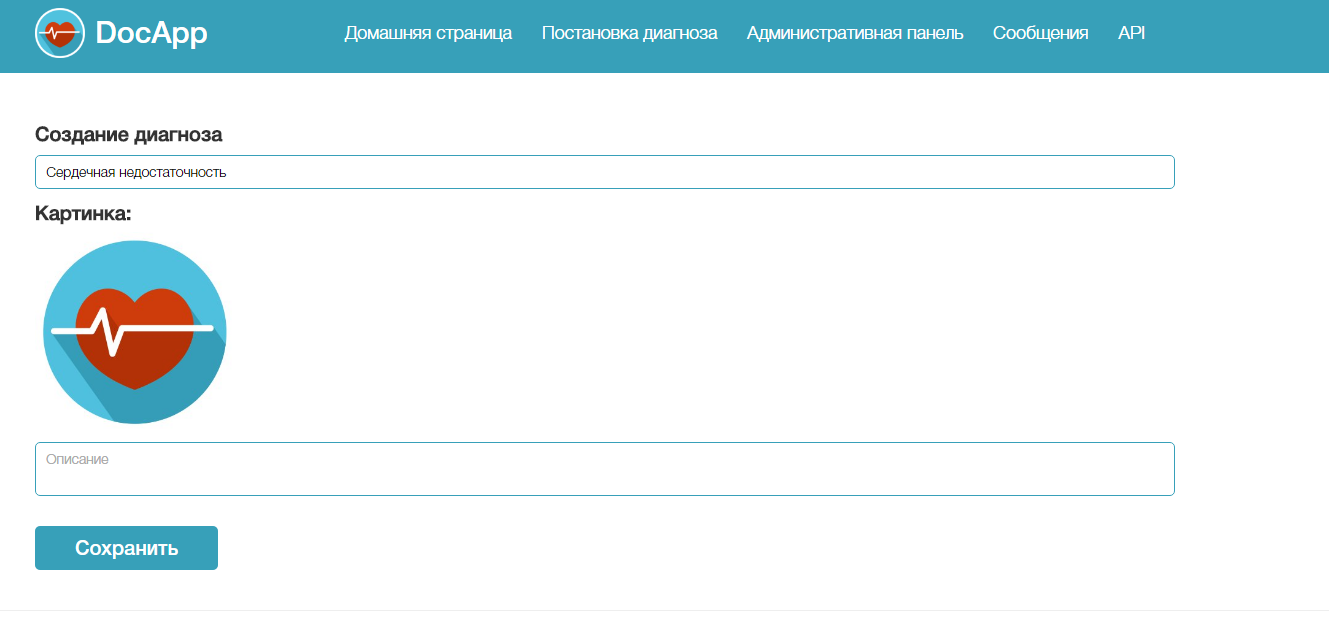


Рис. 2.10. Интерфейс " Создание изменение диагнозов "

Интерфейс предназначен для создания или изменения информации по диагнозу. В данном интерфейсе, можно изменить название диагноза, его изображение, а также можно добавить описание диагноза. При нажатии на кнопку сохранить происходит сохранение введенной пользователем данных и переход на экран «Административная панель».

* + 1. **Интерфейс "Обучение диагноза"**

Роли: "Администратор", "Врач".

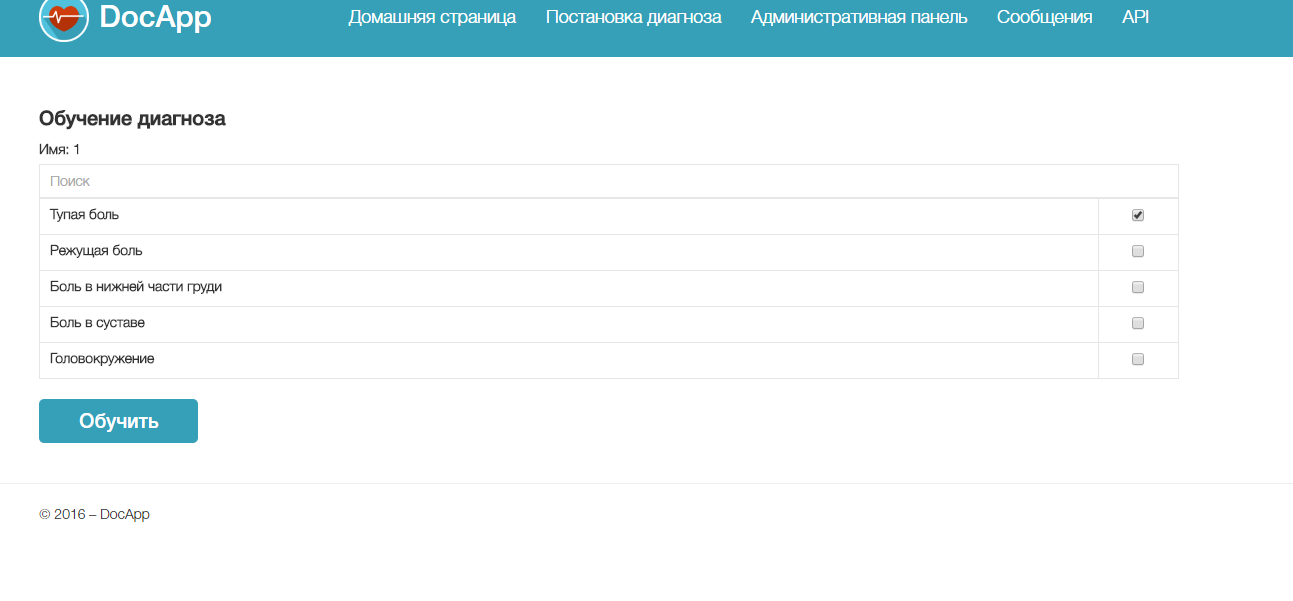


Рис. 2.11. Интерфейс " Обучение диагноза "

Интерфейс предназначен для обучения диагноза. С помощью поиска можно найти интересующие пользователя симптомы. И отметить те симптомы, которые относятся к выбранному диагнозу. При нажатии на кнопку обучить происходит сохранение введенной пользователем данных и переход на экран «Административная панель».

* + 1. **Интерфейс "Апи"**

Роли: "Администратор".

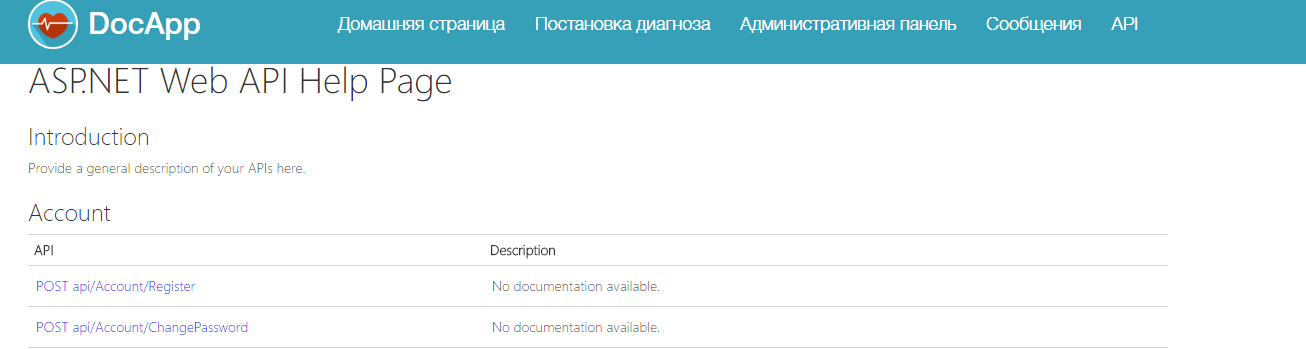


Рис. 2.12. Интерфейс "Апи"

Интерфейс предназначен для просмотра методов апи. Подробной информации по каждому из методов, включая входные и выходные параметры.

* + 1. **Сообщения системы**

Для защиты системы от некорректно введенных данных, а также для помощи пользователю при заполнении форм в системе предусмотрены сообщения валидации, которые указывают на ошибки при неверном вводе. В этих сообщениях указывается полная информация о том, почему ввод неверен.

Возможны следующие сообщения системы:

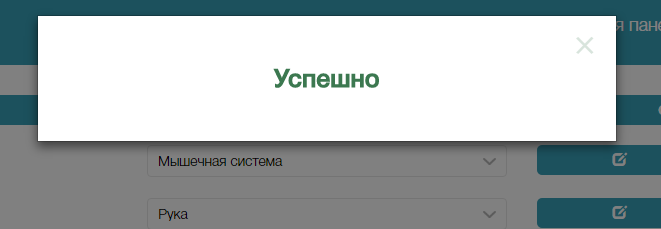


Рис. 2.13. – Сообщение о успешном вводе

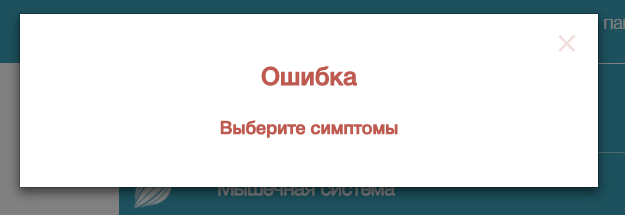
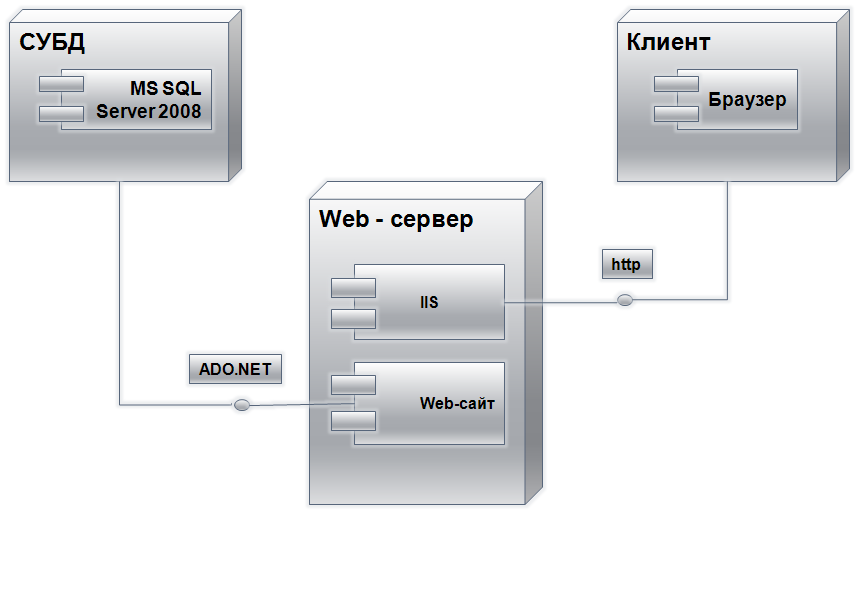


Рис. 2.14. – Сообщение о вводе некорректного выборе симптомов.

**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/.

**ПРИЛОЖЕНИЕ 1**



MS SQL Server 2012

Браузер

IIS

Web-сайт

**ПРИЛОЖЕНИЕ 2**

ИЛИ

**ПРИЛОЖЕНИЕ 3**

