ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»
(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

РАЗРАБОТКА ПРОГРАММЫ ПРОГНОЗИРОВАНИЯ РИСКА РАЗВИТИЯ РЕСТЕНОЗА СОСУДОВ СЕРДЦА НА ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ АРМ-КАРДИОЛОГА

Выпускная квалификационная работа
обучающегося по направлению подготовки 38.03.05 «Бизнес-информатика»
очной формы обучения, группы 07001422
Пензева Константина Ильича

Научный руководитель:
Старший преподаватель Сиваков С.И.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В последнее время в различных медицинских учреждениях образовывается огромное количество медицинских данных, которые активно используются специалистами для постановки диагноза.

Для оптимизации лечебно-диагностических процессов внедряются информационные системы и базы данных – системы, предназначенные для хранения, поиска и обработки огромного количества информации.

Внедрение подобных информационных систем работников помогает медицинским работникам повысить производительность труда, поддерживать данные в достоверном и актуальном состоянии, а также экономить время.

Кроме того, одной из самых распространенных причин ишемической болезни сердца является стенозирующее атеросклеротическое поражение коронарных артерий. И одним из способов лечения данного заболевания является стентирование (имплантация коронарного стента). Однако есть риск избыточного нарастания внутренней оболочки сосуда на участке сосуда, вследствие чего возникает повторное сужение сосудов – рестеноз.

Актуальность данной работы заключается в том, что информатизация лечебно-диагностических процессов с применением математических моделей позволит специалистам повысить точность и скорость диагностирования заболеваний и скорректировать план лечения на основе полученных прогнозов.

Объектом исследования является процесс развития рестеноза сосудов сердца.

Предметом исследования являются факторы развития рестеноза сосудов сердца.

Целью данной работы является повышение эффективности ранней диагностики и прогнозирования риска развития рестеноза сосудов сердца посредством разработки программного обеспечения для АРМ-Кардиолога.

Для достижения данной цели необходимо решить следующие задачи:

— исследовать теоретические аспекты диагностирования и лечения рестеноза сосудов сердца как осложнения ишемической болезни сердца и средства разработки и проектирования информационных систем на основе математических моделей;

— проанализировать существующие способы прогнозирования риска развития рестеноза сосудов сердца и разработать математические модели прогнозирования риска развития рестеноза сосудов сердца;

— разработать программное обеспечение, предназначенное для прогнозирования риска развития рестеноза сосудов сердца на основе математических моделей.

Используемые методы для разработки и написания работы: сравнение, анализ, синтез, измерение и моделирование.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка использованных источников и приложений. В работе используются рисунки. Объем работы – 82 страницы.

1 Исследование вопросов разработки программы для информационной системы прогнозирования риска развития рестеноза сосудов сердца

1.1 Исследование способов диагностирования и лечения рестеноза сосудов сердца как осложнения ишемической болезни сердца

В настоящее время проблема диагностики и лечения больных ишемической болезнью сердца (ИБС) остается одной из самых актуальных задач отечественного здравоохранения. Несмотря на достигнутые в последние годы успехи в профилактике и лечении ИБС, она по-прежнему занимает ведущие позиции в структуре заболеваемости и смертности населения России [1].

В основе развития ИБС лежит дисбаланс между потребностью сердечной мышцы в кровоснабжении и фактическим коронарным кровотоком. Этот дисбаланс может развиваться в связи с резко возросшей потребностью миокарда в кровоснабжении, но недостаточном его осуществлении, либо при обычной потребности, но резком снижении коронарного кровообращения. Дефицит кровоснабжения миокарда особенно выражен в случаях, когда коронарный кровоток снижен, а потребность сердечной мышцы в притоке крови резко возрастает. Недостаточное кровоснабжение тканей сердца, их кислородное голодание проявляется различными формами ишемической болезни сердца. В группу ИБС входят остро развивающиеся и хронически протекающие состояния ишемии миокарда, сопровождающиеся последующими его изменениями: дистрофией, некрозом, склерозом. Эти состояния в кардиологии рассматриваются, в том числе, и в качестве самостоятельных заболеваний [2].

Диагностику ИБС осуществляют кардиологи в условиях кардиологического стационара или диспансера с использованием специфических инструментальных методик. При опросе пациента выясняются жалобы и наличие характерных для ишемической болезни сердца симптомов. При осмотре определяются наличие отеков, цианоза кожных покровов, шумов в сердце, нарушений ритма.

Среди важнейших методов диагностики ИБС выделяют:

– Электрокардиография (ЭКГ) – электрофизиологический тест, включающий регистрацию биоэлектрических потенциалов сердца с помощью накожных электродов и их графическое воспроизведение на бумаге или дисплее.

– Эхокардиография (ЭхоКГ) - метод УЗИ сердца, позволяющий визуализировать размеры сердца, состояние полостей и клапанов, оценить сократимость миокарда, акустические шумы. В некоторых случаях при ИБС проводят стресс эхокардиографию – ультразвуковую диагностику с применением дозированной физической нагрузки, регистрирующую ишемию миокарда.

– В диагностике ИБС широко используются функциональные пробы с нагрузкой. Они применяются для выявления ранних стадий ИБС, когда нарушения еще невозможно определить в состоянии покоя. В качестве нагрузочных тестов используются ходьба, подъем по лестнице, нагрузки на тренажерах (велотренажере, беговой дорожке), сопровождающиеся ЭКГ-фиксацией показателей работы сердца.

– Холтеровское суточное мониторирование ЭКГ предполагает регистрацию ЭКГ, выполняемую в течение суток и выявляющую периодически возникающие нарушения в работе сердца.

– Чрезпищеводная электрокардиография (ЧПЭКГ) позволяет детально оценить электрическую возбудимость и проводимость миокарда. Суть метода состоит во введении датчика в пищевод и регистрации

показателей работы сердца, минуя помехи, создаваемые кожными покровами, подкожно-жировой клетчаткой, грудной клеткой.

– Проведение коронарографии в диагностике ишемической болезни сердца позволяет контрастировать сосуды миокарда и определять нарушения их проходимости, степень стеноза или окклюзии. Коронарография используется для решения вопроса об операции на сосудах сердца [2].

При лечении ИБС могут использоваться различные немедикаментозные, лекарственные и хирургические методы.

Так, к немедикаментозной терапии относятся мероприятия по коррекции образа жизни и питания. При различных проявлениях ИБС показано ограничение режима активности, т. к. при физической нагрузке происходит увеличение потребности миокарда в кровоснабжении и кислороде. Неудовлетворенность этой потребности сердечной мышцы фактически и вызывает проявления ИБС. Поэтому при любых формах ишемической болезни сердца ограничивается режим активности пациента с последующим постепенным расширением его во время реабилитации [2].

Лекарственная терапия при ИБС назначается по формуле «А-В-С»: антиагреганты, β-адреноблокаторы и гипохолестеринемические препараты. Отсутствие эффекта от проводимой лекарственной терапии ишемической болезни сердца и угроза развития инфаркта миокарда являются показанием к консультации кардиохирурга для решения вопроса об оперативном лечении.

Реваскуляризация миокарда – медицинское вмешательство, нацеленное на восстановление естественного кровотока в сердечных сосудах (ликвидацию или минимизацию стеноза) [2].

На нынешнем этапе развития хирургической науки выделяют такие варианты реваскуляризации: полная анатомическая, полная функциональная и неполная функциональная [2].

Главный проблемой, которую должна решить хирургия при коронарном стенозе, является нарушение кровообращения в

ишемизированом участке. Для этого используют аортокоронарное шунтирование (АКШ) и такие радикальные способы как лазерная ангиопластика, установка стентов и разные варианты атерэктомии.

Главным преимуществом АКШ по сравнению с другими способами является продолжительность благоприятного исхода и улучшение качества жизни пациентов. Через пять лет после проведения вмешательства результат остается таким же, как и после операции в 80% пациентов с венозными и 95% пациентов с артериальными шунтами. Другим преимуществом этого способа является возможность проведения операции независимо от локализации стеноза [3].

Принципиальное различие между шунтированием и эндоваскулярными манипуляциями состоит в том, что в первом случае процедура может проводиться несколько раз и имеет минимальные осложнения, отсутствует необходимость общей анестезии.

Даже с таким послужным списком, хирургия неспособна полностью устранить проблему стеноза коронарных артерий. Это дало толчок для поиска менее вредных и экономных способов для устранения проблем ишемической болезни сердца [3].

Именно так и появилось стентирование. Оно имеет самый лучший среди всех способов и кратко-, и долгосрочный прогнозы. Устранение множественных поражений путем стентирования еще не очень развито из-за возможности тромбоза сосудов и процессов рестеноза в области вмешательства [3].

После стентирования или ангиопластики пациенты, в подавляющем большинстве случаев, испытывают значительное облегчение: исчезают боли в грудной клетке. Если же возникает рестеноз, то все эти симптомы возобновляются, иногда даже с большей силой.

В большинстве случаев возникновения рестеноза проводят повторное стентирование, устанавливают новый стент в старый – «стентирование стент в стент» или устанавливают стент в том месте, где раньше проводилась

баллонная ангиопластика. Таким образом, лечение рестеноза исключительно хирургическое [21].

Таким образом, были изучены основные методы диагностики и лечения ишемической болезни сердца и рестеноза как её осложнения.

1.2 Исследование средств разработки и проектирования программы прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога

В ходе ранее проведенных исследований была разработана схема электронной модели сердца, которая описывает различные аспекты функционирования сердечно-сосудистой системы (ССС). Центральное место в данной модели занимает автоматизированное рабочее место кардиолога (АРМ-Кардиолог), целью которого является генерация диагноза [22-23]. В структуру АРМ-Кардиолог, помимо описываемого в данной работе программного обеспечения прогнозирования риска развития рестеноза сосудов сердца, входят 3 системы:

– «Компьютерная система анализа» - компьютерная система ЭКГ с расширенными функциями автоматического поиска и идентификации областей с диагностически значимыми изменениями, может быть классифицирована как инструмент для повышения качества и эффективности взаимодействия человека и компьютера при диагностике болезни [24];

– «Автоматизированная экспертная система анализа вариабельности сердечных сокращений» - автоматизированная диагностическая экспертная система количественной оценки риска ишемической болезни сердца на ранних стадиях на основе результатов анализа вариабельности сердечного ритма [25-26];

– «Автоматизированная система формирования и распознавания многомерного образа состояния сердечно-сосудистой системы» - система на основе алгоритма построения многомерного виртуального образа состояния сердечно-сосудистой системы (ССС) пациента, используемого для решения задачи распознавания патологий на основе исследования топологии взаимного расположения и коллизий пересечения двумерных образов состояния ССС [27-28].

На рисунке 1.1 изображена модульная структура взаимосвязей как разрабатываемой программы, так и вышеописанных информационных систем.

Рисунок 1.1 – Модульная структура взаимосвязей информационных систем кардиологического отделения

В данной работе рассматривается разработка программы прогнозирования риска развития рестеноза сосудов сердца автономно от АРМ-Кардиолог.

Для решения поставленных задач необходимо разработать такой инструмент как математическая модель.

Математическая модель - это совокупность математических объектов (чисел, переменных, векторов, множеств и т.п.) и отношений между ними, которая адекватно отображает некоторые свойства проектируемого технического объекта. В процессе проектирования применяют те

математические модели, которые отображают существенные с позиций инженера-проектировщика свойства объекта.

Под математическим моделированием понимают способ исследования различных процессов путем изучения явлений, имеющих различное физическое содержание, но описываемых одинаковыми математическими соотношениями [4].

Часто термин «математическое моделирование» относят не только к оперированию математической моделью, но и к ее построению [4].

Классифицировать модели можно по разным критериям. Например, по характеру решаемых проблем модели могут быть разделены на функциональные и структурные. В первом случае все величины, характеризующие явление или объект, выражаются количественно. При этом одни из них рассматриваются как независимые переменные, а другие — как функции от этих величин. Математическая модель обычно представляет собой систему уравнений разного типа (дифференциальных, алгебраических и т.д.), устанавливающих количественные зависимости между рассматриваемыми величинами. Во втором случае модель характеризует структуру сложного объекта, состоящего из отдельных частей, между которыми существуют определенные связи. Как правило, эти связи не поддаются количественному измерению. Для построения таких моделей удобно использовать теорию графов [5].

По характеру исходных данных и результатов предсказания модели могут быть разделены на детерминистические и вероятностно-статистические. Модели первого типа дают определенные, однозначные предсказания. Модели второго типа основаны на статистической информации, а предсказания, полученные с их помощью, имеют вероятностный характер [5].

Одними из видов математических моделей являются искусственные нейронные сети и регрессионные модели.

Искусственная нейронная сеть (ИНС) — это математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма [6].

Регрессия (или регрессионный анализ) – это статистический метод исследования влияния одной или нескольких независимых переменных X1, X2,…..Xp на зависимую переменную Y [7, с.87].

Одной из регрессионных моделей является множественная степенная регрессия – модель зависимости одной (объясняемой, зависимой) переменной y от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) x со степенной функцией зависимости [8, с.106].

Но для успешной реализации математической модели необходимо разработать базу данных как один из структурных компонентов программы.

Основой программы является база данных – набор данных, который хранится определенным упорядоченным способом, и ее проектирование является одной из важнейших составляющих при разработке информационных систем.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро производить выборку с произвольным сочетанием признаков. При этом очень важно выбрать правильную модель данных. Модель данных – это формализованное представление основных категорий восприятия реального мира, представленных его объектами, связями, свойствами, а также их взаимодействиями [9].

Одним из этапов разработки базы данных является проектирование её структуры, её будущего образа. И одним из способов её отображения является составление ER-модели.

ER-модель (или модель «сущность-связь») – модель данных, позволяющая описывать концептуальные схемы предметной области.

ER-модель удобна при проектировании информационных систем, баз данных, архитектур компьютерных приложений и других систем (моделей). С помощью такой модели выделяют существенные элементы (узлы, блоки) модели и устанавливают связи между ними.

Основными элементами ER-модели являются:

– Сущность – класс однотипных объектов, информация о которых должна быть учтена в модели;

– Экземпляр сущности – конкретный представитель данной сущности;

– Атрибут сущности – именованная характеристика, являющаяся некоторым свойством сущности;

– Ключ сущности – неизбыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности;

– Связь – некоторая ассоциация между двумя сущностями [10].

Начальным и определяющим весь процесс разработки базы данных является выбор инструментальных средств разработки.

И при осуществлении данного выбора необходимо руководствоваться некоторыми факторами, важнейшими из которых являются:

– тип модели данных, которая поддерживает данная СУБД;

– характеристики производительности СУБД;

– набор функциональных возможностей, необходимых для разработки базы данных;

– удобство и надежность СУБД в эксплуатации [11].

СУБД (Система управления базами данных) – это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями. Система управления базами данных (СУБД) является посредником между базой данных и ее пользователями.

На сегодняшний день существует множество СУБД, которые успешно применяются людьми в различных сферах деятельности. И издание под названием DB-Engines опубликовало рейтинг популярности СУБД на ноябрь 2016 г. Рейтинг представлен на рисунке 1.1.

Rank			DBMS
Nov 2016	Oct 2016	Nov 2015	
1.	1.	1.	Oracle
2.	2.	2.	MySQL
3.	3.	3.	Microsoft SQL Server
4.	↑ 5.	↑ 5.	PostgreSQL
5.	↓ 4.	↓ 4.	MongoDB
6.	6.	6.	DB2
7.	7.	↑ 8.	Cassandra
8.	8.	↓ 7.	Microsoft Access
9.	9.	↑ 10.	Redis
10.	10.	↓ 9.	SQLite

Рисунок 1.2 – Рейтинг популярности СУБД на ноябрь 2016 г.

Рейтинг показал, что наиболее популярными СУБД являются Oracle, MySQL, Microsoft SQL Server. Ниже в рейтинге расположились не менее известные СУБД PostgreSQL, Microsoft Access и SQLite.

Oracle – это объектно-ориентированная СУБД, разработанная компанией Oracle. Данная СУБД хороша тем, что она поддерживает серверные технологии, такие как Java Server Pages, Java-сервлеты, а также интерфейсы прикладного программирования CORBA. Кроме того, данная СУБД может работать на нескольких операционных системах, в том числе Windows, Linux, Sun Solaris. Но при работе с Oracle стоит учитывать то, что она предъявляет высокие требования к аппаратному обеспечению, а также отсутствие готового программного обеспечения для работы с базами данных конечных пользователей [12].

Microsoft SQL Server – система управления реляционными базами данных, которая разработана корпорацией Microsoft. В данной СУБД используется такой язык запросов как Transact-SQL. Среди достоинств

данной СУБД можно подметить защиту данных от случайных потерь и несанкционированного доступа, а также хорошее быстродействие. Из недостатков можно отметить то, что данная СУБД работает только для одной платформы (Win32) [13].

PostgreSQL – свободная объектно-реляционная СУБД. Главным достоинством данной СУБД является то, что она обладает открытым ПО, которое удовлетворяет стандарту SQL. Кроме того, при разработке базы данных именно в PostgreSQL имеется возможность расширения функционала за счет сохранения своих процедур. Но PostgreSQL явно не славится своей производительностью. При простых операциях чтения данная СУБД может значительно замедлить сервер [12].

Для создания базы данных была использована СУБД MySQL по ряду причин:

— Возможности данной СУБД позволяют спроектировать удобную и функциональную базу данных. Кроме того, можно установить дополнительные приложения, которые увеличивают функционал СУБД;

— MySQL поддерживает большое количество функций, обеспечивающих безопасность.

— MySQL с легкостью работают с большими объемами данных и масштабируется;

— Данная СУБД обладает высокой производительностью за счет соответствия конкретным стандартам современных СУБД [12].

Но, как и любая СУБД, MySQL имеет свои недостатки. Несмотря на широкий функционал, и он имеет свои ограничения, которые необходимо учитывать при проектировании базы данных. Например, MySQL не поддерживает некоторые возможности реляционных СУБД [12]. Среди них:

В бесплатной версии недоступны транзакции – объединение нескольких SQL-запросов в единую логическую единицу работы в данными [14];

Недоступны триггеры – набор команд, которые хранятся в базе данных и выполняются тогда, когда происходит определенное событие;

Но в некоторых задачах можно обойтись без вышеперечисленных конструкций.

Для успешного проектирования базы данных необходимо программное обеспечение для администрирования сервера MySQL, запуска команд SQL и просмотра содержимого таблиц и баз данных.

Для работы с базой данных MySQL используется phpMyAdmin – это веб-приложение, представляющее собой веб-интерфейс для администрирования СУБД. PhpMyAdmin пользуется большой популярностью за счет удобства и функциональности интерфейса [15].

Также необходимы программы для удобного ввода и вывода информации из базы данных.

Для этих целей при разработке программы и базы данных используется Flask – микрофреймворк для Python по разработке веб-сервисов.

Преимущества Flask в том, что он позволяет создать веб-сайты довольно просто и быстро. Хоть и преимущества Flask в простоте разработки, одной из отрицательных черт является то, что он имеет довольно узкий функционал [16, с.25].

Flask использует механизм шаблонов Jinja2. Также он имеет полную поддержку Unicode. Считается, что создание шаблонов на Jinja2 имеет широкие возможности

Особенности Jinja2:

- мощная автоматическая система HTML;

- наследование шаблонов;

- компилирует до оптимального кода Python [16, с.25].

Таким образом, были определены наиболее оптимальные средства разработки информационной системы и базы данных [16].

2 Анализ и разработка моделей прогнозирования риска развития рестеноза сосудов сердца

2.1 Анализ существующих способов прогнозирования риска развития рестеноза сосудов сердца

На сегодняшний день учеными было разработано несколько разновидностей математических моделей прогнозирования риска развития рестеноза сосудов сердца.

Разработан способ прогнозирования риска развития рестеноза сосудов сердца после стентирования коронарных артерий без лекарственного покрытия (RU №2395091), основанный на определении отдельных особенностей генетической структуры пациента: Glu298Asp-полиформизмов гена эндотелиальной синтазы оксида азота (eNOS) и Prol98Leu-полиформизмов гена глутатионпероксидазы-1 (GPx-1). При выявлении у пациента только аллелей Glu298 и Pro198 полиморфизмов Glu298Asp гена eNOS и Prol98Leu гена GPx-1 прогнозируют низкий риск развития рестеноза. При выявлении в генотипе либо аллеля 298Asp полиморфизма Glu298Asp гена eNOS либо аллеля 198Leu полиморфизма Prol98Leu гена GPx-1 прогнозируют средний риск развития рестеноза. При выявлении присутствия и аллеля 298Asp полиморфизма Glu298Asp гена eNOS и аллеля 198Leu полиморфизма Prol98Leu гена GPx-1 прогнозируют высокий риск развития рестеноза. Данный способ обеспечивает высокую степень достоверности прогнозирования развития рестеноза после стентирования коронарных артерий с использованием стентов без лекарственного покрытия [17, с.1]. Но, имеется и такой существенный недостаток, как отсутствие количественной оценки прогнозирования.

В основу другого способа прогнозирования риска развития рестеноза стентов в коронарных артериях в ранние сроки после эндоваскулярной реваскуляризации миокарда лежит оценка различных веществ, выделяемых клетками организма (патент №2349919, опубл. 20.03.2009). Для осуществления способа прогнозирования рестеноза стентов в коронарных артериях в ранние сроки после эндоваскулярной реваскуляризации миокарда оценивают иммунный воспалительный ответ путем определения динамики экспрессии цитокина ИЛ-1. Для этого определяют его концентрацию до и через сутки после стентирования. И при повышении в крови уровня ИЛ-1 через сутки после эндоваскулярной реваскуляризации на 25% прогнозируют рестеноз стентов в коронарных артериях. Предлагаемый способ позволяет прогнозировать развитие рестеноза внутри стента на ранних стадиях прогрессирования заболевания [18, с.1]. Однако также недостатком данного способа прогнозирования является отсутствие количественной оценки прогнозирования.

Также известен способ прогнозирования развития рестеноза в стенте у мужчин трудоспособного возраста с первичным неосложненным инфарктом миокарда (патент № 2410019, опубл. 27.01.2011), основанный на математической модели. Данный метод прогнозирования риска развития рестеноза может быть использовано для раннего прогнозирования повторных сердечно-сосудистых катастроф, в частности рецидива стенокардии после первичного неосложненного инфаркта миокарда у мужчин трудоспособного возраста. Учитывая расположение стента, показатели нагрузочной пробы, ангинозный приступ, депрессию ST во время пробы, повторность госпитализации, рассчитывают прогностический коэффициент вероятности развития рестеноза в стенте по оригинальной математической формуле. Прогноз основан на учете факторов, которые доступны врачу на амбулаторном этапе, позволяет прогнозировать вероятность рестеноза в стенте в динамике [19, с.1] Данный способ прогнозирования отличается тем, что используются математические методы. Так, для мужчин

трудоспособного возраста с первичным неосложненным инфарктом миокарда рассчитывают прогностический коэффициент вероятности развития рестеноза в стенте по формуле:

$$P=-0,0301*K_1-0,0758*K_2+1,4024*K_3+0,0456*K_4+0,3417*K_5+0,0289, \quad (2.1)$$

где $К_1$ - локализация стента: передняя нисходящая артерия - 1, правая коронарная артерия - 2, огибающая артерия - 3, диагональная ветвь - 4;

$К_2$ - Возобновление стенокардии: 0 - нет, 1 - есть;

$К_3$ - Повторная госпитализация: 0 - не было, 1 - была;

$К_4$ - Развитие типичного ангинозного приступа во время ВЭМ: 0 - нет, 1 - да;

$К_5$ - Депрессия сегмента ST (косонисходящая или горизонтальная), мм;

Свободный член = 0,0289 [19, с.5]

Среди недостатков данного метода можно отметить ограниченность его использования, т.к. его можно применять только для мужчин. Кроме того, в расчетах вероятности развития рестеноза не учитываются результаты биохимических анализов, величина стеноза и т.д.

Еще одним способом, основанном на математическом моделировании, является способ прогнозирования вероятности развития рестеноза после стентирования коронарных артерий (патент № 2532340, опубл. 10.11.2014). Сущность способа прогнозирования вероятности развития рестеноза в том, что на момент стентирования осуществляют забор крови пациента и регистрируют в физических величинах значения протромбинового индекса, коэффициента атерогенности, липопротеидов очень низкой плотности, липопротеидов высокой плотности, вычисляют величину стеноза S. После чего рассчитывают коэффициент вероятности развития рестеноза R, соответствующий прогнозируемой величине рестеноза через 6 месяцев, по формуле. При этом адекватность прогнозируемой величины рестеноза через

6 месяцев после стентирования обеспечивается для следующих интервалов: при локализации стеноза в огибающей артерии 0<R<40; при локализации стеноза в ветвях тупого края 0<R<50, 90<R<100; при локализации стеноза в правой коронарной артерии 0<R<50; при локализации стеноза в передней межжелудочковой артерии 0<R<100. Использование заявленного способа позволяет осуществить раннее прогнозирование рецидивов сердечнососудистых осложнений, в частности развития рестеноза, повторного сужения сосудов сердца после стентирования в правой коронарной артерии, огибающей артерии, в передней межжелудочковой артерии и ветвях тупого края [20, с.1].

Рассчитывают прогностический коэффициент вероятности развития рестеноза через 6 месяцев с учетом локализации стента в правой коронарной артерии, огибающей артерии, передней межжелудочковой артерии и ветвях тупого края по формуле:

$$R = K_0+K_1*S+K_2*BC_1+K_3*BC_2+K_4*BC_3+K_5*BC_4+K_6*S^2+ K_7*BC_1^2+K_8*BC_2^2+K_9*BC_4^2+K_{10}*BC_2*S+K_{11}*BC_3*S+K_{12}* BC_4*S+K_{13}*BC_1*BC_2+K_{14}*BC_1*BC_3+K_{15}*BC_3*BC_4 \quad (2.2)$$

где BC_1 - протромбиновый индекс ПТИ;

BC_2 - коэффициент атерогенности;

BC_3 - липопротеиды очень низкой плотности (ЛПОНП);

BC_4 - липопротеиды высокой плотности (ЛПВП);

Ki - коэффициент, значение которого зависит от локализации стеноза, где индекс i принимает значения от 1 до 15 [20, с.5].

Данный способ прогнозирования также имеет свои недостатки. Так, при прогнозировании учитываются не все виды локализаций стеноза.

Таким образом, в данном разделе были рассмотрены имеющиеся способы прогнозирования риска развития рестеноза сосудов сердца.

2.2 Разработка математических моделей прогнозирования риска развития рестеноза сосудов сердца для информационной системы АРМ-Кардиолог

Для решения задачи прогнозирования риска развития рестеноза сосудов сердца используются такие математические модели как искусственные нейронные сети (ИНС) и регрессионные математические модели.

Для сравнения результатов, которые были получены при помощи данных моделей используется коэффициент детерминации. Данный коэффициент обычно применяется для расчета тесноты нелинейной связи между переменными. Чем ближе коэффициент детерминации к единице, тем сильнее зависимость переменной x от переменной y [8, с.106].

Предыдущие исследования в данном направлении показали взаимосвязь величины рестеноза от следующих генетических маркеров:

- Гаптоглобина(Hp)
- Группоспецифического компонента(Gc)
- Трансферрина(Tf)
- Цитратсинтаза(C‘s).

По системе Hp и Gc имеются 3 вида фенотипов: 1-1, 1-2, 2-2. По системе Tf имеется 2 вида фенотипов CC и CB. По системе C‘s имеется 3 типа фенотипов: SS, FS, FF.

Выходными данными является прогнозируемая величина рестеноза [8, с.106].

Так как математические модели ИНС и множественной степенной регрессии не воспринимают текстовые виды входных и выходных данных, то все входные данные были закодированы [8, с.106]. Расшифровка входных данных представлена в таблице 2.1.

Таблица 2.1 – Кодирование входных данных

Фенотип	Код фенотипа в модели
1	2
Hp 1-1	1
Hp 1-2	2
Hp 2-2	3
Gc 1-1	4
Gc 1-2	5
Gc 2-2	6
Tf CB	7
Tf CC	8
C’s SS	9
C’s FS	10
C’s FF	11

Математическая модель ИНС проектируется при помощи Neural Network Toolbox – пакет расширения Matlab, содержащий средства для проектирования, моделирования, разработки и визуализации нейронных сетей.

После кодирования входных данных при помощи инструментального средства nntool (графический интерфейс, позволяющий, не обращаясь к командному окну системы Matlab, выполнить создание, обучение, моделирование, а также импорт и экспорт нейронных сетей и данных) были введены входные данные (множества кодированных фенотипов генных маркеров) и целевые данные (наличие или отсутствие рестеноза).

После чего формируется и обучается нейронная сеть, выводятся выходные данные (прогнозируемая величина рестеноза).

Как видно на рисунке 2.1, Input- это множества кодированных фенотипов генных маркеров (входные данные), Target – наличие или отсутствие рестеноза (целевые данные), Outputs – это прогнозируемая величина рестеноза (выходные данные), Errors – массив ошибок сети [8, с.107].

После формирования и обучения нейронной сети представляется возможным прогнозировать величину рестеноза у новых пациентов двумя путями.

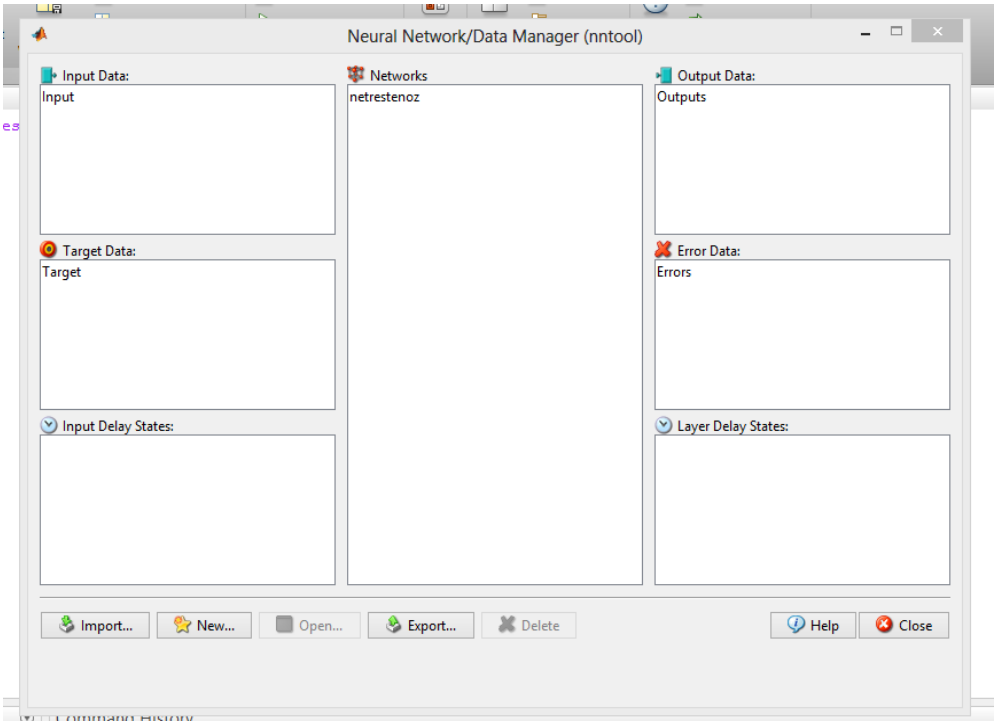

Рисунок 2.1 – Моделирование нейронной сети в nntool

Первый способ предполагает формирование идентичного входного множества (Input1), добавляя лишь новые входные данные. Затем во вкладке Simulate формируется новое выходное множество forecast, где и формируется новая формирующая величина рестеноза.

Второй способ представлен на рисунке 2.2. В данном случае в командном окне вводится команда sim с названием нейронной сети и входными данными [8, с.107].

```
>> sim (netrestenoz, [2;4;7;9])

ans =

    0.9537
```

Рисунок 2.2 – Прогнозирование величины рестеноза с помощью команды sim

При этом выведется прогнозируемая вероятность возникновения рестеноза сосудов сердца.

Проблемой при формировании нейронной сети является то, что у некоторых пациентов при одинаковом наборе фенотипов разные исходы наличия или отсутствия рестеноза. В данной ситуации при обучении нейронной сети выходные данные размываются.

В таблице 2.2. представлены сведения о пациентах с одинаковым набором фенотипов, разными исходами наличия или отсутствия рестеноза, а также с рассчитанной величиной рестеноза.

Таблица 2.2. – Неопределенные данные в нейронной сети

Совокупность входных данных	Количество пациентов с отсутствием рестеноза	Количество пациентов с наличием рестеноза	Прогнозируемая величина рестеноза
1	2	3	4
[Hp 2-2;Gc 1-2;Tf CC;C's SS]	9	3	0,21
[Hp 2-2;Gc 1-1;Tf CC;C's FS]	3	2	0,49
[Hp 1-2;Gc 1-1;Tf CC;C's SS]	12	3	0,19
[Hp 1-2;Gc 1-2;Tf CC;C's SS]	16	2	0,03
[Hp 2-2;Gc 1-1;Tf CC;C's SS]	20	1	0,08
[Hp 1-2;Gc 1-2;Tf CC;C's FS]	6	1	0,02
[Hp 2-2;Gc 2-2;Tf CC;C's SS]	4	1	0,1

Для проектирования математической модели с помощью метода множественной степенной регрессии был применен табличный редактор Microsoft Excel. С его помощью выводится уравнение модели множественной степенной регрессии типа:

$$y=a_0*x_1^{a_1}*x_2^{a_2}*x_3^{a_3}*x_4^{a_4}, \qquad (2.3)$$

где y- это поставленный или не поставленный диагноз «рестеноз» пациенту;

a_0, a_1, a_2, a_3, a_4 – свободные коэффициенты;

x_1 - это вид фенотипа пациента по генетическому маркеру Гаптоглобин(Hp);

x_2- по маркеру Группоспецифического компонента(Gc);

x_3-по маркеру Трансферрин(Tf);

x_4 – по маркеру Цитратсинтаза(C’s). [8, с.108]

Так как значения у равны 0 и 1, то проведение моделирования с помощью множественной степенной регрессии невозможно из-за того, что необходимо просчитывать натуральные логарифмы. В связи с этим было проведено нормирование выборки х и у по формуле:

$$x_{норм} = \frac{|x_i - x_{ср}|}{\varphi}, \qquad (2.4)$$

где xi – i-й член выборки;

xср – среднее значение выборки;

φ – стандартное отклонение выборки.

В результате данной операции математическая модель множественной степенной регрессии представлена уравнением:

$$y = 1{,}162014 * x_1^{-0{,}17386} * x_2^{-0{,}00748} * x_3^{0{,}364522} * x_4^{0{,}112241}. \qquad (2.5)$$

Был проведен сравнительный анализ результатов построения двух математических моделей: ИНС и множественной степенной регрессии. Были рассчитаны коэффициенты детерминации (коэффициент рассчитывающий тесноту связи между переменными) обоих методов, и в итоге получилось, что коэффициент детерминации ИНС равен 0,21, а множественной степенной регрессии – 0,12.

Так как у вышеописанных моделей довольно большое количество размытых данных и коэффициент детерминации меньше 0,5, то все размытые данные были исключены из рассмотрения. Таким образом, осталось 36 пациента, на основе чьих данных и были сформированы модели на базе ИНС и множественной степенной регрессии.

Алгоритм формирования данных моделей практически идентичен с алгоритмом формирования предыдущих моделей.

Исключением в алгоритме формирования модели ИНС является разделение исходной выборки на 2 части: обучающую и проверяющую. В обучающей выборке имеются данные 30 пациентов, а в проверяющей выборке данные 6 пациентов, не входящих в обучающую выборку.

Проверочная выборка отображена в таблице 2.3.

Таблица 2.3 – Проверочная выборка искусственной нейронной сети

Совокупность входных данных	$Y_{экспер.}$	$Y_{расч.}$	Отклонение
1	2	3	4
[Hp 2-2;Gc 1-1;Tf CC;C's SS]	1	0,0000258	0,999974
[Hp 1-1;Gc 1-1;Tf CC;C's FF]	0	0,0004151	0,0004151
[Hp 2-2;Gc 1-2;Tf CC;C's FS]	0	0,0001353	0,0001353
[Hp 1-1;Gc 1-1;Tf CC;C's SS]	0	00001144	0,0001144
[Hp 2-2;Gc 2-2;Tf CC;C's FS]	0	0,00009133	0,00009133
[Hp 2-2;Gc 1-1;Tf CC;C's SS]	0	0,0000258	0,0000258

При этом, среднее отклонение расчетных данных от уже имеющихся составляет 0,17, что является приемлемым результатом.

Так, снова в результате построения математической модели с помощью множественной степенной регрессии было выведено новое уравнение математической модели множественной степенной регрессии:

$$y=1{,}180702*x_1^{-0{,}11405}*x_2^{-0{,}24044}*x_3^{0{,}752164}*x_4^{0{,}35807}. \qquad (2.6)$$

Также был проведен сравнительный анализ новых моделей. Были рассчитаны коэффициенты детерминации обоих методов, и в итоге получилось, что коэффициент детерминации ИНС равен 0,72, а множественной степенной регрессии – 0,53.

Это доказывает, что у модели ИНС теснее связь результативного признака Y с переменными X [8, с.110].

Для подтверждения повышения эффективности ранней диагностики и прогнозирования риска развития рестеноза сосудов сердца при помощи разработанной математической модели на базе искусственной нейронной сети была проведена процедура расчета средних отклонений расчетных данных от фактических у способов прогнозирования риска развития рестеноза сосудов сердца, перечисленных в подразделе 2.1 и сравнения со средним отклонением разработанной математической модели на базе искусственной нейронной сети. Так, были получены данные, представленные в таблице 2.4.

Таблица 2.4 – Средние отклонения способов прогнозирования риска развития рестеноза сосудов сердца и разработанной математической модели

№ п/п	Способ прогнозирования	Среднее отклонение
1	2	3
1	Способ прогнозирования развития рестеноза после стентирования коронарных артерий стентами без лекарственного покрытия	0,21
2	Способ прогнозирования рестеноза стентов в коронарных артериях в ранние сроки после эндоваскулярной реваскуляризации миокарда	0,3
3	Способ прогнозирования развития рестеноза в стенте у мужчин трудоспособного возраста с первичным неосложненным инфарктом миокарда	0,45
4	Способ прогнозирования вероятности развития рестеноза после стентирования коронарных артерий	0,03
5	Разработанная математическая модель на базе искусственной нейронной сети	0,17

Как было отмечено в подразделе 2.1, преимущество разработанной математической модели на базе искусственной нейронной сети перед способом прогнозирования №4 является то, что учитываются все локализации стеноза.

Также при помощи способа прогнозирования №4 оценивается величина рестеноза, а в разработанной модели оценивается вероятность возникновения рестеноза и позволит судить о генетической предрасположенности к заболеванию.

Таким образом, были разработаны математические модели прогнозирования риска развития сосудов сердца на основе анализа генетических маркеров при помощи искусственных нейронных сетей и множественной степенной регрессии. После разработки данных моделей было выявлено, что коэффициент детерминации ИНС равен 0,72, а множественной степенной регрессии – 0,53. Это доказывает, что использование ИНС делает систему более линейной, адаптивной и результативной при решении плохоформализованных задач и при работе с неполной информацией. И именно поэтому ИНС, по сравнению с реляционными моделями являются одним из самых популярных методов исследования [8, с.110].

3 Разработка программы для информационной системы прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога

3.1 Проектирование базы данных программы прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога

ER-модель проектируемой базы данных представлена на рисунке 3.1.

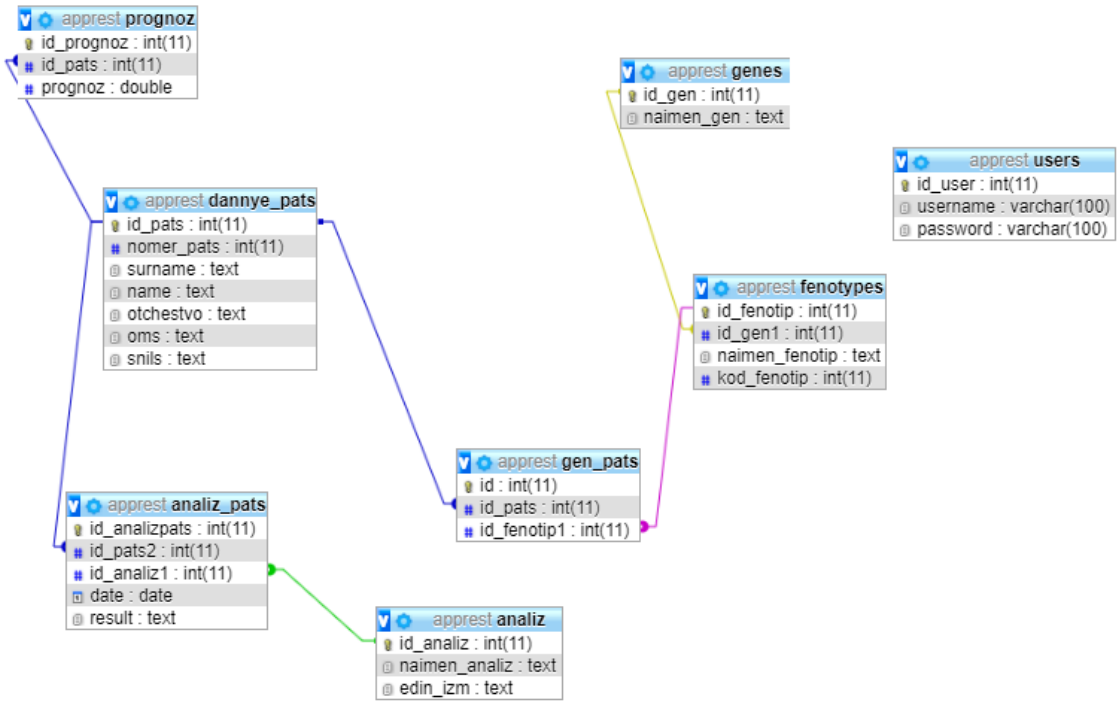

Рисунок 3.1 – ER-модель базы данных программы прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога

В данной ER-модели можно выделить 8 сущностей: dannye_pats, genes, fenotypes, gen_pats, prognoz, analiz, analiz_pats, users.

Сущность «dannye_pats» содержит персональную информацию о пациентах (ФИО, номер полиса ОМС, СНИЛС). Она содержит 7 атрибутов:

– id_pats – идентификационный номер пациента в базе данных, integer (11), первичный ключ;

– nomer_pats – номер пациента, integer (11);

– surname – фамилия пациента, text;

– name – имя пациента, text;

– otchestvo – отчество пациента, text;

– oms – номер полиса ОМС пациента, text;

– snils – номер СНИЛС пациента, text.

Сущность «genes» содержит список генов, по которым ведется прогнозирование. Она содержит 2 атрибута:

– id_gen – идентификационный номер гена в базе данных, integer (11), первичный ключ;

– naimen_gen – наименование гена, text.

Сущность «fenotypes» содержит список фенотипов, по которым ведется прогнозирование. Она содержит 4 атрибута:

– id_fenotip – идентификационный номер фенотипа в базе данных, integer (11), первичный ключ;

– id_gen1 – идентификационный номер гена в базе данных, integer (11), внешний ключ к атрибуту id_gen в сущности «genes»;

– naimen_fenotip – наименование фенотипа, text.

– kod_fenotip – закодированные данные для фенотипов, по которым ведется прогнозирование, integer (11)

Сущность «gen_pats» содержит генетические данные пациентов. Она содержит 3 атрибута:

– id – идентификационный номер значения фенотипа пациента в базе данных, integer(11), первичный ключ;

– id_pats – идентификационный номер пациента в базе данных, integer (11), внешний ключ к атрибуту id_pats в сущности «dannye_pats»;

– id_fenotip1 – идентификационный номер фенотипа в базе данных, integer (11), внешний ключ к атрибуту id_fenotip в сущности «fenotypes».

Сущность «prognoz» содержит прогнозируемые данные пациентов. Она содержит 3 атрибута:

– id_prognoz – идентификационный номер прогнозируемого значения в базе данных, integer (11), первичный ключ;

– id_pats – идентификационный номер пациента в базе данных, integer (11), внешний ключ к атрибуту id_pats в сущности «dannye_pats»;

– prognoz – прогнозируемое значение пациента, double.

Сущность «analiz» содержит список биохимических анализов. Она содержит 3 атрибута:

– id_analiz – идентификационный номер анализа в базе данных, integer (11), первичный ключ;

– naimen_analiz – наименование анализа, text;

– edin_izm – единица измерения анализа, text.

Сущность «analiz_pats» содержит информацию об анализах пациентах и результатах их проведения. Она содержит 5 атрибутов:

– id_analizpats – идентификационный номер анализа пациента в базе данных, integer (11), первичный ключ;

– id_pats2 – идентификационный номер пациента в базе данных, integer (11), внешний ключ к атрибуту id_pats в сущности «dannye_pats»;

– id_analiz1 – идентификационный номер анализа в базе данных, integer (11), внешний ключ к атрибуту id_analiz в сущности «analiz»;

– date – дата проведения анализа у пациента, date;

– result – результат проведения анализа, text.

Сущность «users» содержит аутентификационные данные для входа в АРМ. Она содержит 3 атрибута:

– id_user – идентификационный номер пользователя АРМ, integer (11), первичный ключ;

– username – имя пользователя, используемое для входа в АРМ, varchar (100);

– password – пароль, используемый для входа в АРМ, varchar (100).

Все связи в базе данных типа «один-ко-многим». Структуру связей в проектируемой базе данных также наблюдать на рисунке 3.1.

Все сущности базы данных спроектированы с применением нормальной формы, так все атрибуты являются простыми, а также неприводимо зависят от первичного ключа.

Вышеописанная схема базы данных была зарегистрирована, и было получено свидетельство о государственной регистрации базы данных (Приложение А)

Таким образом, была спроектирована структура базы данных программы прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога, а также были описаны все сущности и атрибуты. Также можно сказать, что все сущности были спроектированы с использованием нормальной формы.

3.2 Разработка программы прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога

В данном разделе выпускной квалификационной работы содержатся дерево функций, логическая и физическая структура, а также описание всех разработанных функций на языке Python, необходимых для функционирования программы.

При разработке программы и информатизации деятельности медицинских специалистов необходимо привести иерархию функций, которые он должен выполнять. Данная иерархия должна полностью

отображать всю совокупность задач, выполняемых программой и может быть представлена в виде дерева функций.

Реализация программы прогнозирования риска развития рестеноза сосудов сердца представляет собой набор основных и служебных функций.

Из служебных функций, которые выполняет программа, можно выделить проверку авторизации пользователя и своевременное обновление информации.

К основным функциям можно отнести ввод, редактирование и удаление данных, обработку введенной информации, вывод результатов прогнозирования и отчетов.

Дерево функций представлено на рисунке 3.2.

При работе с программой пользователь, в первую очередь, видит набор взаимосвязанных при помощи гиперссылок web-страниц, разделенных по какому-либо признаку.

Данный набор web-страниц, разделенных по каким-либо признакам с различными гиперсвязями между ними и называется логической структурой программного обеспечения.

Описываемое в данной выпускной квалификационной работе программное обеспечение можно представить в виде логической структуры, представленной на рисунке 3.3.

Все файлы разрабатываемой программы находятся в директории apprest.

В директории apprest расположены следующие файлы и директории:

– apprest.py – файл, содержащий параметры соединения с базой данных, а также основные функции Python разрабатываемой программы, которые вызываются в ответ на запрос какого-то адреса (url) и возвращает контекст;

– net.xml – файл, содержащий основные параметры нейронной сети (веса и смещения) и необходимый для подключения к нейронной сети;

– templates – директория, содержащая html-шаблоны, содержащие средства языка разметки HTML и Jinja2, необходимые для представления данных.

Рисунок 3.2 – Дерево функций разрабатываемой программы

Рисунок 3.3 – Логическая структура программного обеспечения

Физическая структура разрабатываемой программы представлена на рисунке 3.4.

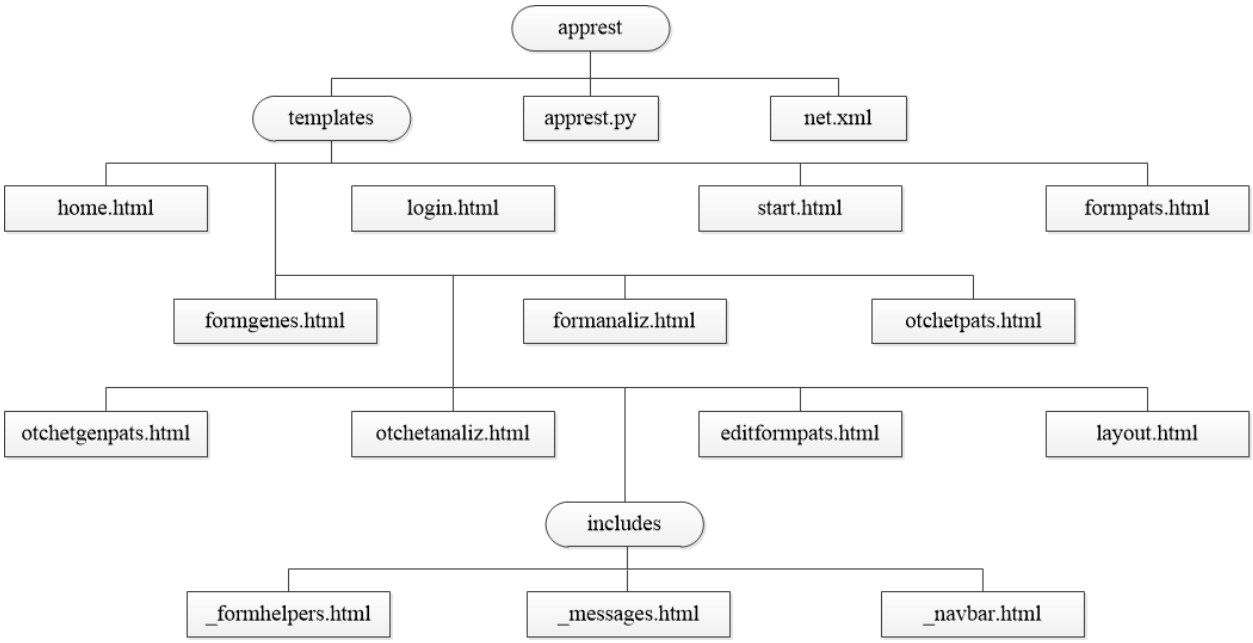

Рисунок 3.4 – Физическая структура программы

Директория templates содержит следующие файлы и директории:

– home.html – шаблон, предназначенный для отображения стартовой страницы;

– login.html – шаблон, предназначенный для отображения страницы авторизации пользователей;

– start.html – шаблон, предназначенный для отображения главного меню;

– formpats.html – шаблон, предназначенный для отображения формы ввода данных о пациенте;

– formgenes.html – шаблон, предназначенный для отображения формы ввода и редактирования данных о пациенте;

– formanaliz.html – шаблон, предназначенный для отображения формы ввода биохимических данных пациента;

– otchetpats.html – шаблон, предназначенный для отображения отчета о пациентах;

– otchetgenpats.html – шаблон, предназначенный для отображения отчета о генетических данных пациента;

– otchetanaliz.html – шаблон, предназначенный для отображения отчета о биохимических данных пациента;

– editformpats.html – шаблон, предназначенный для отображения формы редактирования информации о пациентах;

– layout.html – шаблон, содержащий элементы, имеющиеся на всех страницах веб-приложения;

– includes – директория, содержащая некоторые системные шаблоны.

Директория includes содержит следующие файлы:

– _formhelpers.html – шаблон, содержащий средства визуализации форм ввода данных;

– _messages.html – шаблон, предназначенный для отображения флеш-сообщений об ошибках или вводе данных;

– _navbar.html – шаблон, предназначенный для отображения верхней секции навигации.

Содержание данных файлов представлено в Приложении Б.

Как было отмечено ранее, в файле apprest.py содержатся функции, разработанные на языке Python программы. Данный файл содержит функции, представленные в таблице 3.1.

Таблица 3.1 – разработанные для программы функции на языке Python

Функция	Предназначение функции
1	2
index()	Вывод стартовой страницы
login()	Проверка авторизации пользователей
is_logged_in(f)	Условие выполнения других функций только в случае авторизации пользователя
logout()	Выход из учетной записи пользователя
formpats()	Вывод страницы с формой ввода пациентов
add_pats()	Ввод информации о пациентах в базу данных
editformpats(id_pats)	Редактирование данных о конкретном пациенте
otchetpats()	Вывод отчета о пациентах
delete_entry(id_pats)	Удаление данных о конкретном пациенте
otchetgenpats(id_pats)	Вывод отчета о генетических данных конкретного пациента
delete_gen(id_pats)	Удаление генетических данных конкретного пациента
otchetanaliz(id_pats)	Вывод отчета о биохимических данных конкретного пациента
delete_analiz(id_analizpats)	Удаление биохимических данных конкретного пациента
start()	Вывод главного меню
formgenes()	Вывод страницы ввода или редактирования генетических данных пациента
add_fenotype()	Ввод или редактирование генетической информации пациента в базе данных
formanaliz()	Вывод страницы с формой ввода информации биохимических данных пациента
add_analiz()	Ввод биохимической информации пациента в базу данных

Таким образом, в данном разделе было разработано дерево функций, а также логическая и физическая структура программы. Кроме того, было описано предназначение всех файлов и функций, разработанных на языке Python программного обеспечения.

3.3 Проектирование интерфейса программы прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога

Для разработки интерфейса программы для прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога используются такие элементы интерфейса как формы, таблицы.

При запуске веб-приложения открывается его стартовая страница, где предлагается пройти процедуру аутентификации для доступа к формам и отчетам веб-приложения. Стартовая страница приложения изображена на рисунке 3.5.

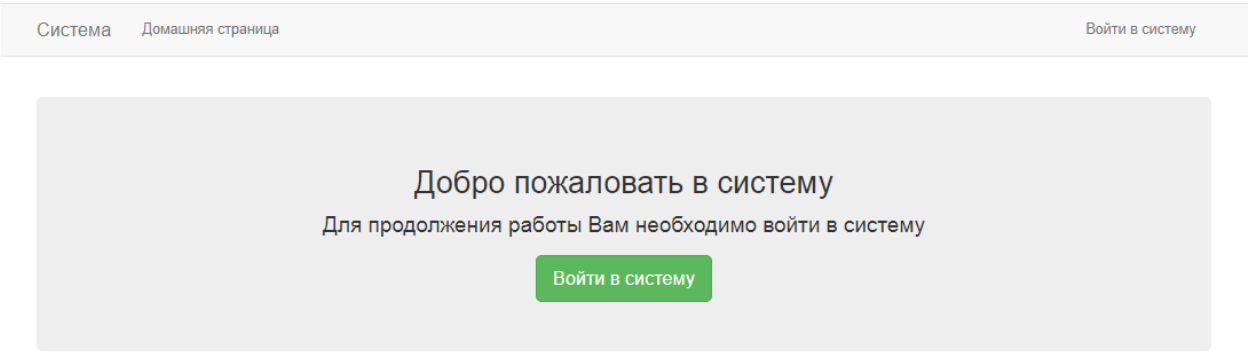

Рисунок 3.5 – Стартовая страница приложения

Для того, чтобы пройти процедуру аутентификации, необходимо нажать кнопку «Войти в систему» в центре страницы или в горизонтальном меню.

После того, как нажата кнопка «Войти в систему», пользователь попадает на страницу аутентификации, где ему нужно ввести логин и пароль для входа в систему. Страница аутентификации пользователей представлена на рисунке 3.6.

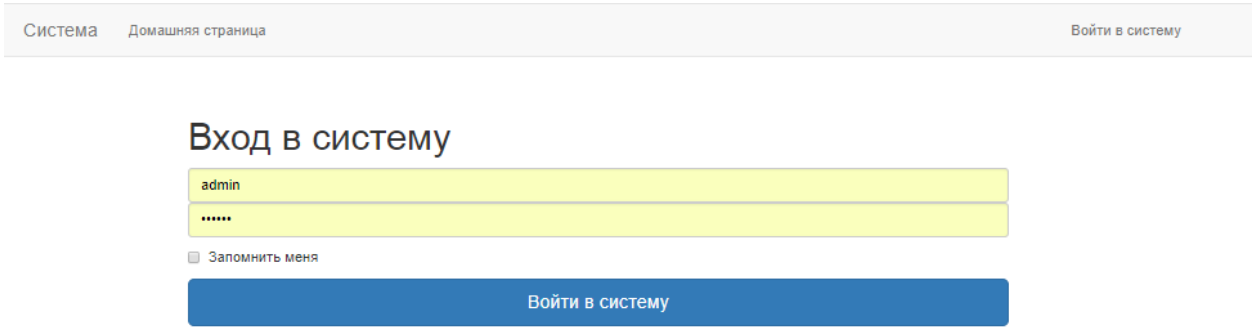

Рисунок 3.6 – Страница аутентификации пользователей

В случае неправильного ввода логина и пароля, пользователю выводится флеш-сообщение красного цвета, оповещающее об ошибке входа.

В случае правильного ввода логина и пароля, пользователь переходит в главное меню веб-приложения, а также выводится флеш-сообщение зеленого цвета, оповещающее о том, что пользователь успешно прошел процедуру аутентификации.

В главном меню имеется 4 пункта:

– «Ввести данные о пациенте»
– «Ввести генетические данные пациента»
– «Ввести данные об анализах пациента»
– «Список пациентов»

Главное меню веб-приложения изображено на рисунке 3.7.

Рисунок 3.7 – Главное меню веб-приложения

При выборе пункта «Ввести данные о пациенте» пользователь переходит к форме ввода информации о пациенте. В данной форме имеется 6 текстовых полей: Номер пациента, Фамилия, Имя, Отчество, Номер полиса ОМС и СНИЛС. При этом, при нажатии на поле «Номер полиса ОМС» и «СНИЛС», возникает маска ввода – это строка символов, указывающая формат допустимых значений входных данных.

Форма ввода данных о пациенте изображена на рисунке 3.8.

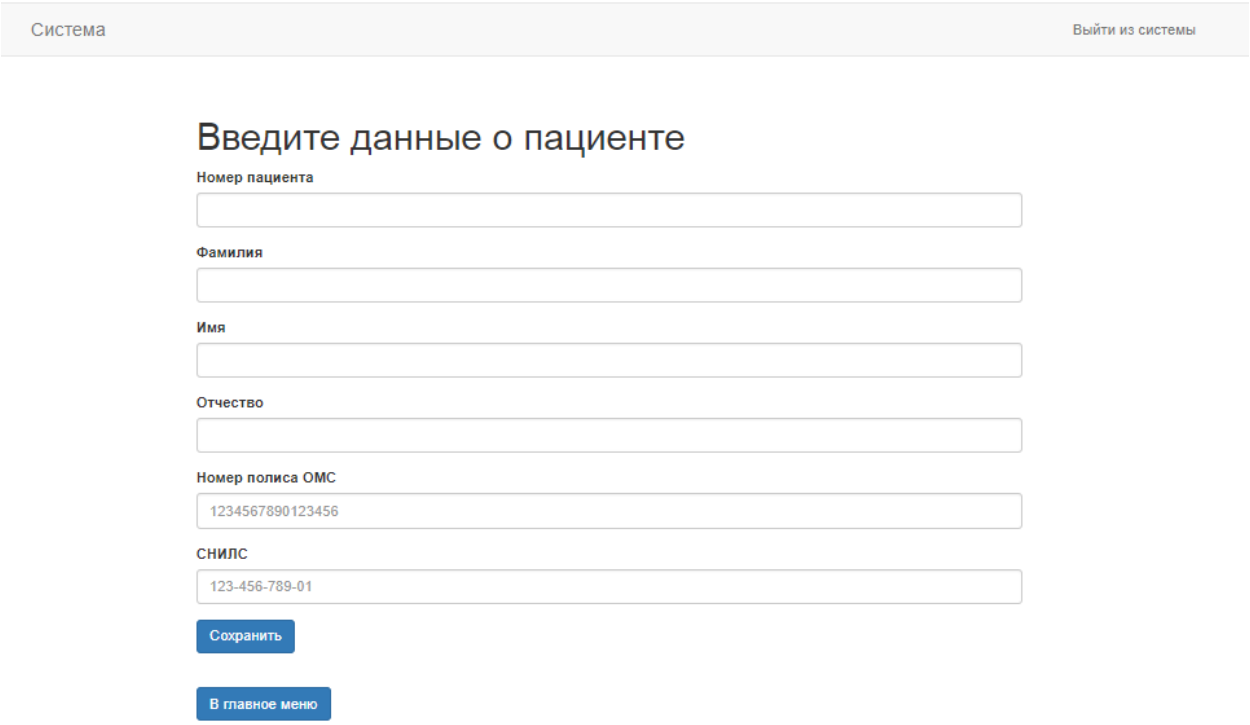

Рисунок 3.8 – Форма ввода данных о пациенте

При выборе в главном меню пункта «Ввести генетические данные пациента» пользователь переходит к форме ввода генетических данных пациента. В данной форме имеется 5 выпадающих списков: ФИО пациента, Hp, Gc, Tf, C’s.

Кроме того, после ввода генетических данных о пациенте веб-приложение с помощью нейронной сети просчитывает вероятность возникновения риска развития рестеноза сосудов сердца и выводит ее на экран с помощью флеш-сообщения, а также вводит её в базу данных в таблицу prognoz.

Форма ввода генетических данных пациента изображена на рисунке 3.9.

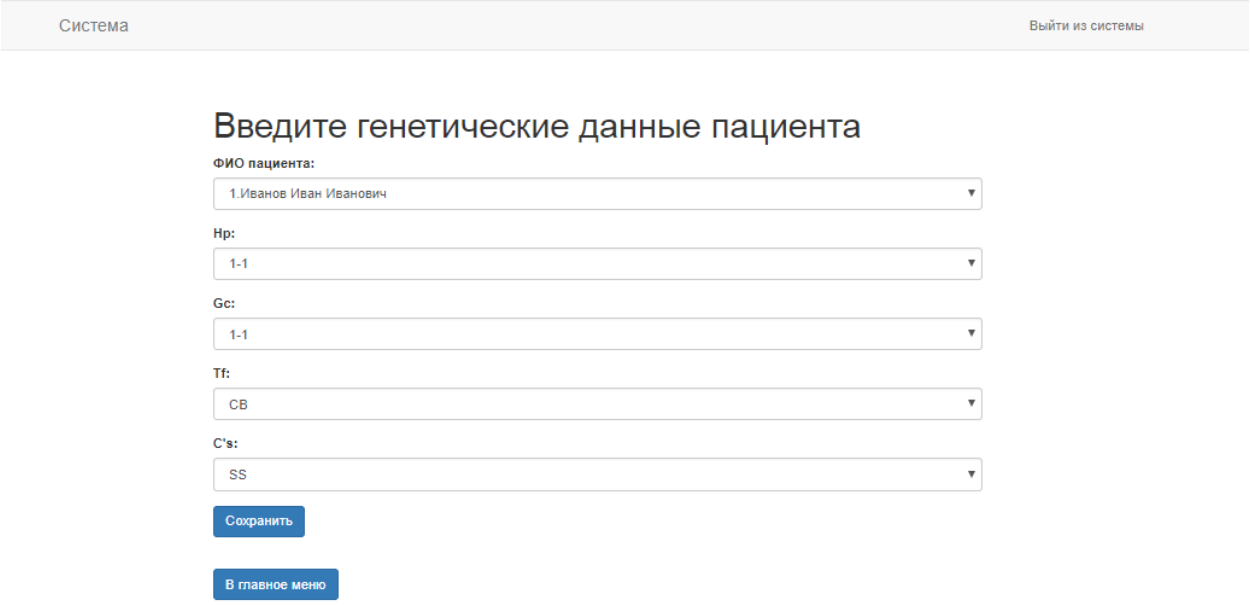

Рисунок 3.9 – Форма ввода генетических данных пациента

При выборе в главном меню пункта «Ввести данные об анализах пациентов» пользователь переходит к форме ввода биохимических данных пациента. В данной форме имеется 4 поля: Фамилия пациента, Вид анализа, Дата сдачи анализа, Результат анализа.

Форма ввода биохимических данных пациента изображена на рисунке 3.10.

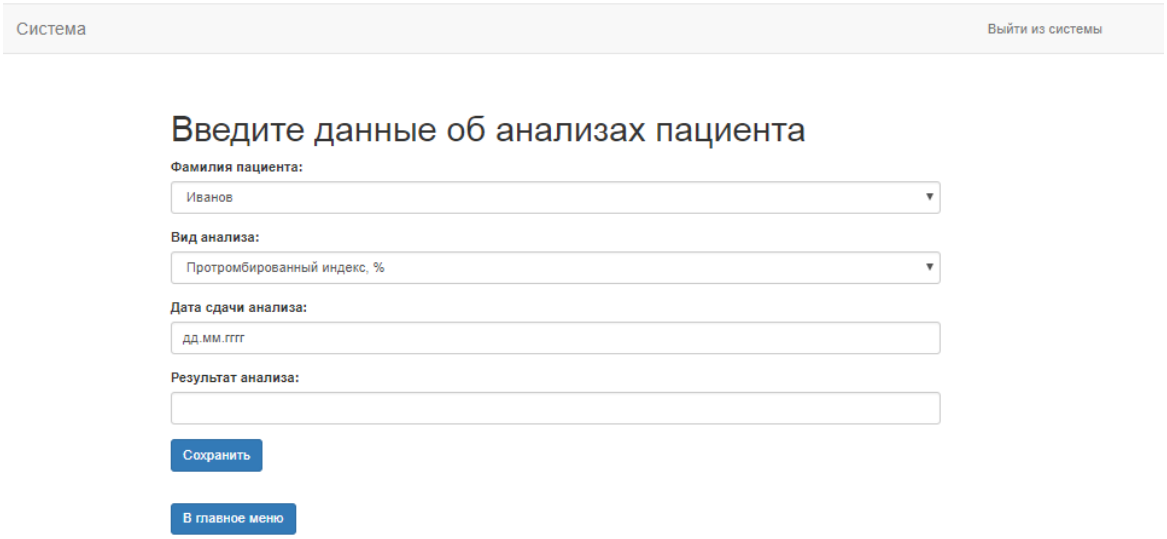

Рисунок 3.10 – Форма ввода биохимических данных пациента

Кроме того, после ввода информации во всех вышеперечисленных формах необходимо нажать кнопку «Сохранить» для занесения данных в базу и обновления страницы формы.

Для перехода из страницы формы в главное меню веб-приложения необходимо нажать на кнопку «Главное меню».

При выборе в главном меню пункта «Список пациентов» пользователь переходит к списку пациентов в табличной форме.

На данной странице имеется кнопка «Добавить данные о пациенте», которая переводит пользователя к форме ввода данных о пациенте.

Также около каждой записи о пациенте имеется 3 кнопки: «Ген. данные», «Анализы» и «Удалить».

При нажатии кнопки «Ген. данные» пользователь переходит на страницу с генетическими данными конкретного пациента.

При нажатии кнопки «Анализы» пользователь переходит на страницу со списком биохимических данных конкретного пациента.

При нажатии на кнопку «Удалить» пользователь удаляет всю информацию о пользователе из базы данных.

Таблица со списком пациентов изображена на рисунке 3.11.

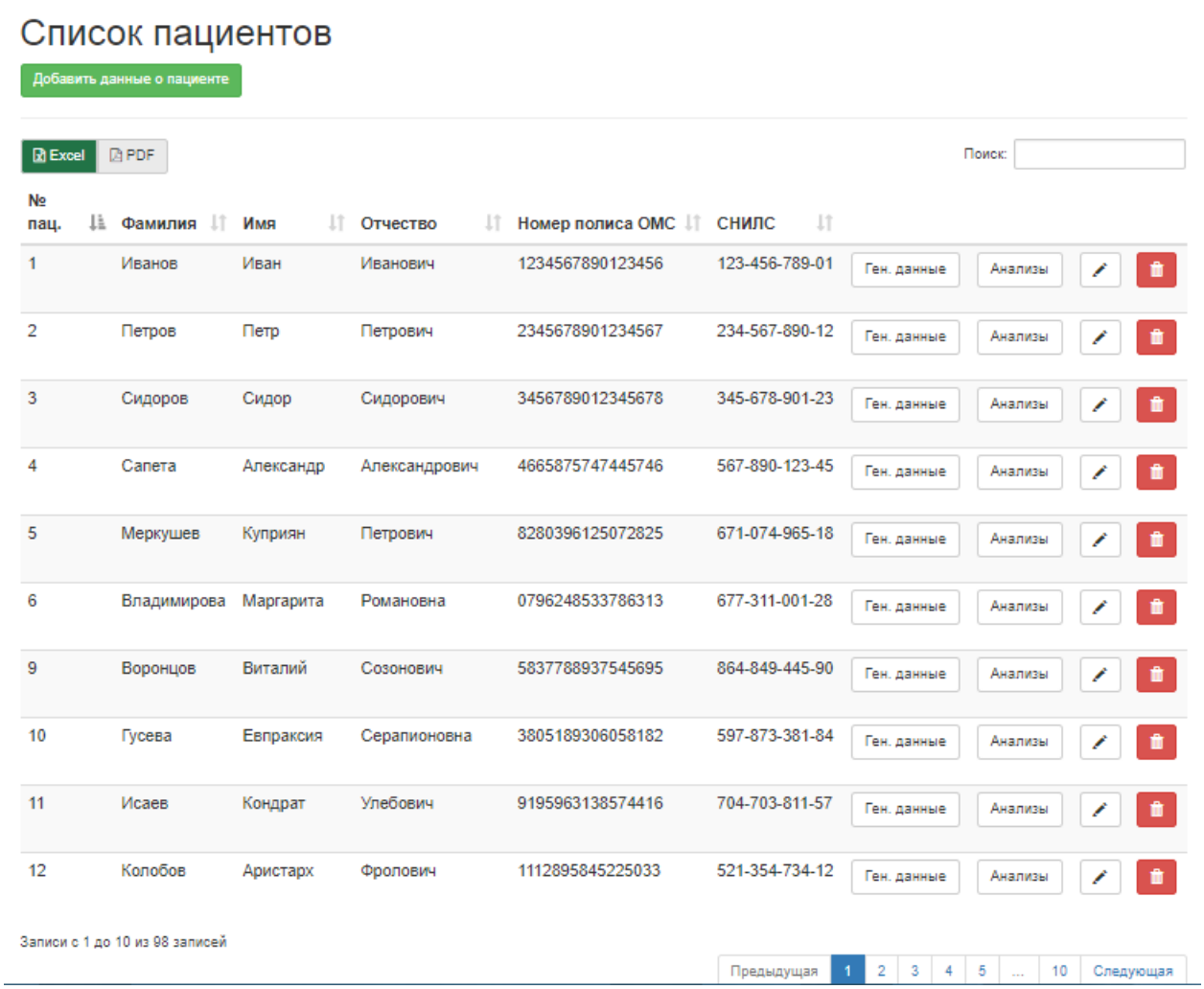

Рисунок 3.11 – Список пациентов

Страница с информацией о генетических данных пациента представлена на рисунке 3.12. На данной странице также имеется кнопка «Добавить ген. данные о пациенте», которая направляет пользователя к форме ввода генетических данных пациента, и кнопка «Удалить», которая удаляет все генетические данные пациента из базы данных.

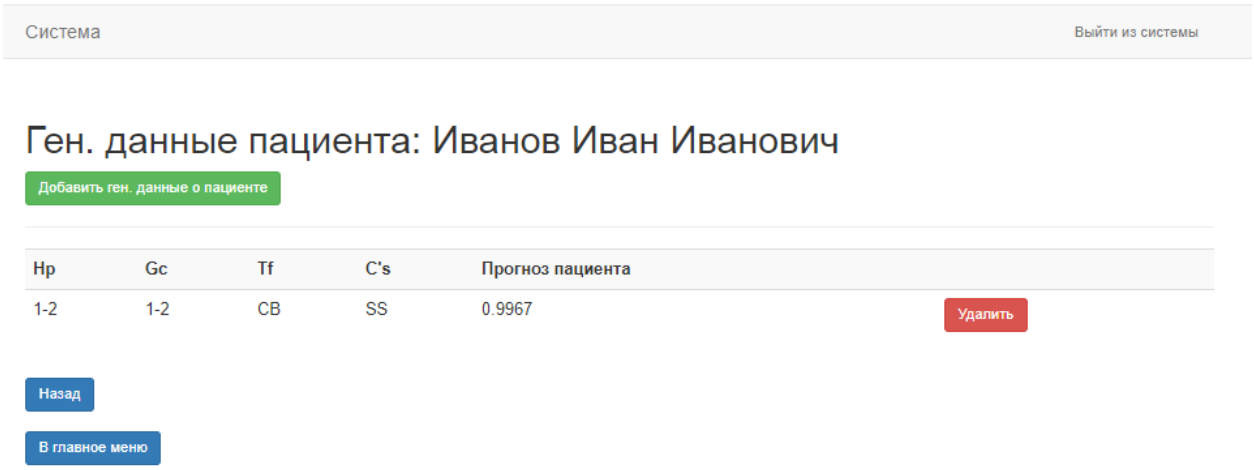

Рисунок 3.12 – Генетические данные пациента

Страница с информацией о биохимических анализах пациентов представлена на рисунке 3.13. На данной странице также имеется кнопка перехода на форму добавления биохимических данных пациента и кнопки «Удалить» для каждой записи для удаления записей о конкретном анализе пациента.

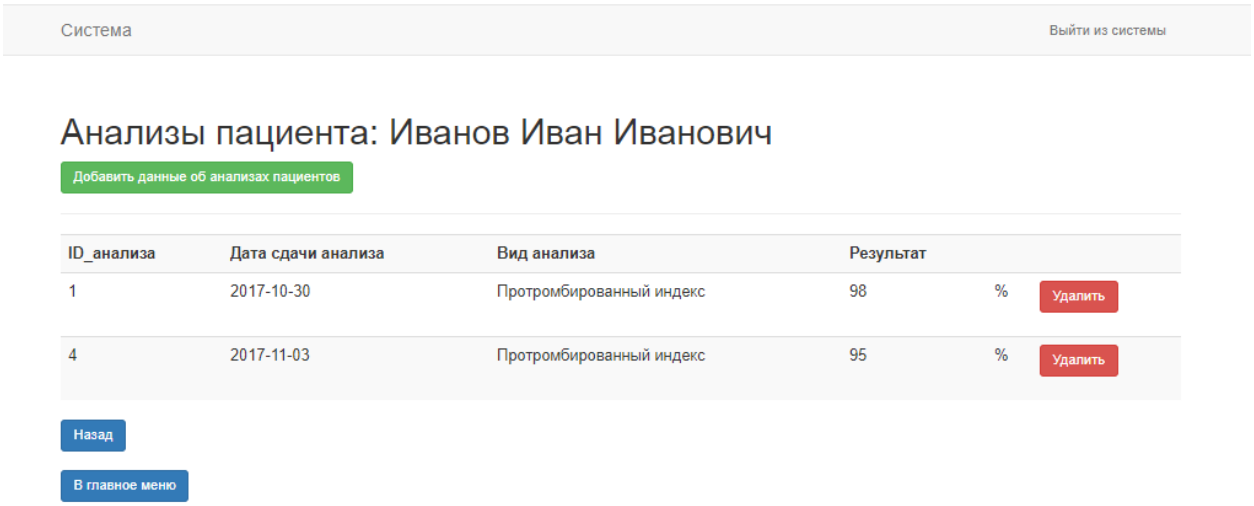

Рисунок 3.13 – Анализы пациента

Кроме того, на всех страницах, доступных аутентифицированному пользователю в горизонтальном меню имеется кнопка «Выйти из системы»

для того, чтобы выйти из системы. В этом случае система переходит на стартовую страницу веб-приложения (рисунок 3.5).

Весь интерфейс пользователя выполнен в белом цвете, наиболее подходящим для работы. Кнопки выполнены в разных цветах. Например, кнопки добавления записей и перехода в главное меню выполнены синим цветом, кнопки перехода из отчетов в формы – зеленым цветом, кнопки перехода между отчетами – белым цветом, кнопки удаления записей – красным цветом. Кроме того, имеется 2 вида флеш-сообщений: зеленое, уведомляющее об успешном входе в систему или вводе данных и красное, уведомляющее о какой-нибудь ошибке.

Таким образом, был разработан интерфейс программы для прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога, а также описаны все её страницы и элементы, необходимые для работы с ней. Каждый пункт главного меню содержит ссылку на конкретную страницу с формой или отчетом, необходимую для работы.

3.4 Расчет экономических затрат

Для оценки экономических затрат необходимо рассчитать единовременные экономические затраты на разработку программного обеспечения.

К затратам на разработку программного обеспечения относятся:

– затраты на проведение научно-исследовательской работы (НИР), которая включает в себя проведение теоретического исследования, подбор и изучение необходимой литературы, разработка математических моделей и программ, отладку и опытную эксплуатацию программного обеспечения;

– затраты на приобретение расходных материалов и оборудования для проведения НИР.

В общую смету расходов на НИР включены:

– материальные затраты;

– заработная плата разработчиков;

– отчисления на социальные нужды;

– амортизационные отчисления;

– затраты на эксплуатацию оборудования;

– затраты на программное обеспечение, используемое при эксплуатации ЭВМ;

– накладные расходы.

Оценка трудоемкости разработки проекта в рамках выпускной квалификационной работы (ВКР) представлена в таблице 3.2.

Таблица 3.2 – Оценка трудоемкости разработки проекта в рамках ВКР

Проводимые работы	Трудоемкость	
	дни	%
1	2	3
Подбор и изучение литературы, составление календарного плана работ	10	6,25
Постановка задачи	10	6,25
Разработка математических моделей	50	31,25
Разработка программ	60	37,5
Обобщения и выводы по проделанной работе	15	9,375
Подготовка отчетов о проделанной работе	10	6,25
Утверждение результатов разработки проекта	5	3,125
Итого:	160	100

К материальным затратам относят такие затраты, которые были направлены на покупку материалов, канцелярских и расходных материалов, необходимых для реализации проекта. Материальные затраты при разработке проекта представлены в таблице 3.3.

Таблица 3.3 – Смета материальных затрат при разработке проекта

Наименование покупных изделий	Марка, тип	Кол-во, шт.	Цена за ед., руб.	Стоимость, руб.
1	2	3	4	5
Бумага (упаковка)	SvetoCopy, A4, 500 л.	1	250	250
Ручка шариковая	Erich Krause	3	20	60
Тетрадь, 48 л.	ПолиГрафика	2	30	60
Итого:				370

Заработная плата Z_o включает в себя заработную плату разработчика, научного руководителя и консультанта по программированию за весь период НИР. Заработная плата рассчитывается по формуле 3.1.

$$Z_o = \sum_{i=1}^{n} (T * Z_{sr}) \quad , \qquad (3.1)$$

где T – трудоемкость одного работника, дни или ч;

Z_{sr} – среднедневная или среднечасовая заработная плата одного работника, руб.

Заработная плата разработчика:

Z_{or} = 120 руб./день * 160 дней = 19200 руб.

Заработная плата научного руководителя:

Z_{on} = 150 руб./ч * 23 ч = 3450 руб.

Заработная плата консультанта по программированию:

Z_{ok} = 150 руб./ч * 5 ч = 750 руб.

Общий фонд заработной платы за весь период НИР составляет:

Z_o = 19200 руб. + 3450 руб. + 750 руб. = 23400 руб.

Отчисления на социальные нужды по действующему законодательству составляют 30% от общего фонда заработной платы. Из них:

- пенсионные отчисления – 22%;
- отчисления в Фонд медицинского страхования – 5,1%;
- отчисления в Фонд социального страхования – 2,9%.

Отчисления на социальные нужды составляют: 23400 руб. * 0,3 = 7020 руб.

Амортизационные отчисления за период, использованного в период проведения НИР, оборудования, рассчитывается по формуле 3.2.

$$A = \frac{O*N*T}{365*100}, \quad (3.2)$$

где O – стоимость оборудования, руб.;

N – норма амортизации, %;

T – время эксплуатации оборудования, дни.

Общая смета амортизационных отчислений изображена в таблице 3.4.

Таблица 3.4 – Смета амортизационных отчислений

Вид оборудования	Стоимость, руб.	Срок службы, лет	Годовая норма амортизации, %	Сумма амортизации, руб.
1	2	3	4	5
Ноутбук Asus	31000	5	20	2717,8
Сетевой фильтр BURO	500	5	20	43,84
Мышь компьютерная Gigabyte	750	5	20	65,75
Итого:	32250			2827,39

Затраты на эксплуатацию оборудования включают стоимость затраченной электроэнергии и рассчитываются по формуле 3.3.

$$Z_{el} = S_{el} * P * T_m * T_s, \quad (3.3)$$

где S_{el} – стоимость 1 кВт/ч электроэнергии, руб.;

P – мощность оборудования, кВт/ч;

T_m – время эксплуатации оборудования за весь период проведения работ, дни;

T_s – среднее время работы оборудования в сутки, ч.

В таблице 3.5 представлены параметры эксплуатации оборудования.

Таблица 3.5 – Параметры эксплуатации оборудования

Параметр	Значение
1	2
Стоимость 1 кВт/ч электроэнергии	3,74 руб.
Мощность ноутбука	0,5 кВт/ч
Время эксплуатации ноутбука за весь период проведения работ	160 дн.
Среднее время работы ноутбука в сутки	5 ч

Таким образом, затраты на эксплуатацию оборудования составят:

$Z_{el} = 3,74 * 0,5*160*5 = 1496$ *руб.*

Затраты на программное обеспечение, используемое при эксплуатации ЭВМ содержат амортизационными отчислениями на использование программного обеспечения. Норма амортизации программного обеспечения составляет 30%.

В таблице 3.6 указана стоимость используемого программного обеспечения.

Таблица 3.6 – стоимость используемого программного обеспечения

Наименование программного обеспечения	Стоимость, руб.
1	2
Microsoft Windows 8 Профессиональная	3990
Microsoft Office 2013	3490
Matlab 2016	124000*
Итого:	131480

* - *цена программного продукта является коммерческой тайной*

Таким образом, затраты на программное обеспечение рассчитываются по формуле (3.2).

$$Z_{po} = \frac{131480*30*160}{365*100} = 17290,52 \text{ руб.}$$

Накладные расходы составляют 30% от общего фонда заработной платы и составляют: 23400 * 0,3 = 7020 руб.

В таблице 3.7 представлена общая смета затрат на выполнение проекта.

Таблица 3.7 – Общая смета затрат на выполнение проекта

Элементы затрат	Сумма, руб.
1	2
Материальные затраты	370
Затраты на заработную плату	23400
Отчисления на социальные нужды	7020
Амортизационные отчисления	2827,39
Затраты на эксплуатацию оборудования	1496
Затраты на программное обеспечение	17290,52
Накладные расходы	7020
Итого:	59423,91

Таким образом, в данном разделе были рассчитаны экономические затраты на разработку программного обеспечения. Они составили 59423,91 руб.

ЗАКЛЮЧЕНИЕ

В результате проведенного исследования была разработана программа для информационной системы кардиологического отделения, которая хранит информацию о пациентах и их фенотипических особенностях. Кроме того, данная программа предназначена для прогнозирования риска развития рестеноза сосудов сердца.

При этом были решены следующие задачи:

– исследованы теоретические аспекты диагностирования и лечения рестеноза сосудов сердца как осложнения ишемической болезни сердца. Для лечения ишемической болезни сердца используются немедикаментозные, лекарственные и хирургические методы лечения, такие как аортокоронарное шунтирование (АКШ) и также радикальные способы как лазерная ангиопластика, установка стентов и разные варианты атерэктомии;

– исследованы средства разработки и проектирования информационной системы на основе математических моделей. Для реализации данной программы были выбраны такие средства разработки веб-приложений как Python, Flask, Jinja2, а также такая СУБД MySQL.

– изучены существующие способы прогнозирования риска развития рестеноза сосудов сердца. В данном разделе были отмечены такие способы прогнозирования как определение Glu298Asp-полиформизмов гена эндотелиальной синтазы оксида азота (eNOS) и Prol98Leu-полиформизмов гена глутатионпероксидазы-1 (GPx-1), оценка иммунного воспалительного ответа путем определения динамики экспрессии цитокина ИЛ-1, разработка математических моделей на основе биохимических анализов пациента и т.д. Каждая из моделей имеет как преимущества, так и недостатки;

– разработаны математические модели прогнозирования риска развития сосудов сердца на основе анализа генетических маркеров при помощи искусственных нейронных сетей и множественной степенной

регрессии. После разработки данных моделей было выявлено, что коэффициент детерминации ИНС равен 0,72, а множественной степенной регрессии – 0,53;

– спроектирована структура базы данных программы прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога, а также были описаны все сущности и атрибуты. Структура была спроектирована с использованием нормальной формы;

– разработаны дерево функций, логическая и физическая структура программы прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога, а также описано предназначение его файлов и разработанных функций на языке Python;

– разработан интерфейс программы для прогнозирования риска развития рестеноза сосудов сердца для АРМ-Кардиолога, а также описаны все её страницы и элементы, необходимые для работы с ней;

– доказано повышение эффективности ранней диагностики и прогнозирования риска развития рестеноза сосудов сердца при помощи разработанной математической модели и программы для АРМ-Кардиолога, по сравнению с ранее разработанными способами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Helpiks.org [Электронный ресурс]/Электрон. дан. – Режим доступа: http://helpiks.org/1-69088.html, свободный

2. Красота и медицина [Электронный ресурс]/Электрон. дан. – Режим доступа: http://www.krasotaimedicina.ru/diseases/zabolevanija_cardiology/ischemic_heart, свободный

3. MEDISTORIYA [Электронный ресурс]/Электрон. дан. – Режим доступа: http://medistoriya.ru/kardiologiya/stenoz-koronarnyh-sosudov-serdca.html, свободный

4. Helpiks.org [Электронный ресурс]/Электрон. дан. – Режим доступа: http://helpiks.org/6-19820.html, свободный

5. Педсовет.инфо [Электронный ресурс]/Электрон. дан. – Режим доступа: http://www.pedsovet.info/info/pages/referats/info_00002.htm, свободный

6. Научный портал ДонНТУ [Электронный ресурс]/Электрон. дан. – Режим доступа: http://science.donntu.edu.ua/sii/shapovalova/library/article9.htm, свободный

7. Фомина, Е.Е. Математические методы анализа данных в социологии с использованием пакетов MS Excel и STATISTICA: учебное пособие [Текст]/ Е.Е. Фомина, Н.К. Жиганов. - Тверь: Тверской Государственный технический университет, 2017. – 168 с.

8. Пензев, К. И. Разработка математических моделей прогнозирования риска развития рестеноза сосудов сердца на основе анализа генетических маркеров [Текст] / К.И. Пензев // Естественнонаучные, инженерные и экономические исследования в технике, промышленности, медицине и сельском хозяйстве материалы I Молодѐжной научно-

практической конференции с международным участием; под общ. ред. С.Н. Девицыной. – Белгород: ИД «Белгород» НИУ «БелГУ», 2017. − 693 с.. – С. 106-110.

9. Беляев, М.А. Основы информатики [Текст]: учебник для вузов /М.А. Беляев, В.В. Лысенко, Л.А. Малинина. – Ростов н/Д: Феникс, 2006. – 352 с.

10. CITFORUM [Электронный ресурс]/Электрон. дан. – Режим доступа: http://citforum.ru/database/dblearn/dblearn08.shtml, свободный

11. PCWEEK. Инфраструктурные решения/ ЦОДы [Электронный ресурс]/Электрон. дан. – Режим доступа: http://www.pcweek.ru/infrastructure/article/detail.php?id=67933 , свободный

12. DEVACADEMY [Электронный ресурс]/Электрон. дан. – Режим доступа: http://devacademy.ru/posts/sqlite-vs-mysql-vs-postgresql/, свободный

13. Большой мир [Электронный ресурс]/Электрон. дан. – Режим доступа: http://bolshoy-mir.com/sqlserver, свободный

14. Lektsii.org [Электронный ресурс]/Электрон. дан. – Режим доступа: https://lektsii.org/10-11536.html, свободный

15. QNAP [Электронный ресурс]/Электрон. дан. – Режим доступа: https://help.qnap.ru/hc/ru/articles/214540525-MySQL, свободный

16. Наумов, Р. В. Программирование Питон. Выбор веб-фреймворка [Текст] / Р.В. Наумов // Достижения науки и образования: сб. статей. – Иваново, 2016. – Вып. 12. – С. 25-26.

17. Пат. 2395091 Российская Федерация, МПК G01N33/53. Способ прогнозирования развития рестеноза после стентирования коронарных артерий стентами без лекарственного покрытия [Текст] / Каминный А.И., Мешков А.Н., Шувалова Ю.А.; заявитель и патентообладатель Федеральное Государственное Учреждение "ФГУ "РКНПК Росмедтехнологий". - № 2009121295/14; заявл. 04.06.09; опубл. 20.07.10, Бюл. № 20– 8 с.

18. Пат. 2349919 Российская Федерация, МПК G01N33/68. Способ прогнозирования рестеноза стентов в коронарных артериях в ранние сроки

после эндоваскулярной реваскуляризации миокарда [Текст] / Тепляков А.Т., Рыбальченко Е.В., Суслова Т.Е., Груздева О.В.; заявитель и патентообладатель ГУ НИИ кардиологии ТНЦ СО РАМН. - № 2007133257/15; заявл. 04.09.07; опубл. 20.03.09, Бюл. № 8– 6 с.

19. Пат. 2410019 Российская Федерация, МПК A61B5/02. Способ прогнозирования развития рестеноза в стенте у мужчин трудоспособного возраста с первичным неосложненным инфарктом миокарда [Текст] / Кузнецова Н.В., Габинский Я.Л., Оранский И.Е., Яковлева С.В., Гофман Е.А.; заявитель и патентообладатель ГОУ ВПО УГМА Росздрава. - № 2009133539/14; заявл. 07.09.09; опубл. 27.01.11, Бюл. № 3– 7 с.

20. Пат. 2532340 Российская Федерация, МПК G01N33/48G01N33/68 . Способ прогнозирования вероятности развития рестеноза после стентирования коронарных артерий [Текст] / Сиваков С.И., Григорова С.Ю., Афанасьев Ю.И., Никитин В.М., Ломакин В.В., Трухачев С.С., Покровский М.В., Новиков О.О.; заявитель и патентообладатель Федеральное государственное автономное образовательное учреждение высшего профессионального образования "НИУ БелГУ". - № 2012155167/15; заявл. 18.12.12; опубл. 10.11.14, Бюл. № 18– 9 с.

21. LIBEMED [Электронный ресурс]/Электрон. дан. – Режим доступа: http://libemed.ru/restenoz/, свободный

22. Efremova, O.A., Nikitin, V.M., Churnosov, M.I., Kamyshnikova, L.A., Lipunova, E.A., Muromtsev V.V., 2016. Risk virtual method assessment of coronary heart disease in carriers of polymorphic cardiogenic. Belgorod State University Scientific Bulletin. Series: Medicine. Pharmacy, 26 (247): 64-69.

23. Efremova, O.A., Nikitin, V.M., Lipunova, E.A., Anohin, D.A., Kamyshnikova, L.A., 2013. Estimate or the effectiveness of intelligent information system of early diagnosis and prognosis of cardiovascular disease. World Applied Sciences Journal, 26 (9): 1204-1208

24. Efremova O.A., Nikitin V.M., Muromtsev V.V., Lipunova E.A., Kamyshnikova L.A., 2016. ECG computer analysis system with the advanced

features of automated search and identification of it’s diagnostically significant changes. International Journal of Pharmacy and Technology. 8 (2): 14174-14181.

25. Efremova O.A., Nikitin V.M., Mitin M.S., Lipunova E.A., Kamyshnikova L.A., 2015. Early diagnosis of coronary heart disease risk by the expert automated system based on the results of heart rate variability analysis. Research Journal of Medical Sciences. 9 (4): 240-244.

26. Efremova, O.A., Nikitin, V.M., Mitin, M.S., Lipunova, E.A., Kamyshnikova, L.A., 2015. Diagnosis of asthenic vegetative syndrome in patients with chronic viral hepatitis based on correlation analysis of heart rate variability. Research Journal of Pharmaceutical, Biological and Chemical Sciences. 6 (4): 161-166

27. Kochetkova, I.A., Dovgal V.M., Nikitin, V.M., Lipunova, E.A., 2011. Method of visualization and recognition of the cardiovascular system state by its multidimensional image. Belgorod State University Scientific Bulletin. Series: History. Political science. Economics. Information technologies, 20 (114): 181-185.

28. Efremova, O.A., Nikitin, V.M., Lipunova, E.A., Kochetkova, I.A., Kamyshnikova, L.A., 2014. Visualization and virtual diagnosis of the cardiovascular system current state by the results of its non-invasive monitoring. Research Journal of Pharmaceutical, Biological and Chemical Sciences. 5 (6): 1512-1518

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Свидетельство о государственной регистрации базы данных

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации базы данных

№ 2018620281

Регистр фенотипических особенностей для АРМ кардиолога

Правообладатель: *федеральное государственное автономное образовательное учреждение высшего образования «Белгородский государственный национальный исследовательский университет» (RU)*

Авторы: *Сиваков Станислав Иванович (RU), Никитин Валерий Михайлович (RU), Афанасьев Юрий Иванович (RU), Пензев Константин Ильич (RU)*

Заявка № 2017621567
Дата поступления 26 декабря 2017 г.
Дата государственной регистрации в Реестре баз данных 15 февраля 2018 г.

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Рисунок А.1. – Свидетельство о государственной регистрации базы данных

ПРИЛОЖЕНИЕ Б

Файл apprest.py

```
#!/flask/bin/python
# -*- coding: utf-8 -*-
import sys
from flask import Flask, render_template, flash, redirect, url_for,
session, request, logging, jsonify
from flask_mysqldb import MySQL
from wtforms import Form, StringField, TextAreaField, PasswordField,
validators, BooleanField
from functools import wraps
from pybrain.tools.shortcuts import buildNetwork
from pybrain.tools.xml.networkwriter import NetworkWriter
from pybrain.tools.xml.networkreader import NetworkReader
reload (sys)
sys.setdefaultencoding('utf-8')
app = Flask(__name__)
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'apprest'
app.config['MYSQL_CURSORCLASS'] = 'DictCursor'
mysql = MySQL(app)

@app.route('/')
def index():
    return render_template ('home.html')

class LoginForm(Form):
    username = StringField('username', [validators.Length(min=1,
max=50)])
    password = PasswordField('password', [validators.Length(min=4,
max=25)])

class Patients(Form):
    nomer_pats = StringField('Номер пациента')
    surname = StringField('Фамилия')
    name = StringField('Имя')
    otchestvo = StringField('Отчество')
    oms = StringField('Номер полиса ОМС')
    snils = StringField('СНИЛС')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password_candidate = request.form['password']
        cur = mysql.connection.cursor()
        result = cur.execute("SELECT * FROM users WHERE username =
%s", [username])
        if result > 0:
            data = cur.fetchone()
            password = data['password']
            if password == password_candidate:
                session['logged_in'] = True
                session['username'] = username
                flash('Вы вошли в систему', 'success')
                return redirect(url_for('start'))
            else:
                error = 'Ошибка входа'
```

```
                    return render_template('login.html', error=error)
                cur.close()
            else:
                error = 'Такой пользователь не найден'
                return render_template('login.html', error=error)
        return render_template('login.html')

    def is_logged_in(f):
        @wraps(f)
        def wrap(*args, **kwargs):
            if 'logged_in' in session:
                return f(*args, **kwargs)
            else:
                flash('Вы не авторизованы. Авторизуйтесь, пожалуйста',
'danger')
                return redirect(url_for('login'))
        return wrap

    @app.route('/logout')
    @is_logged_in
    def logout():
        session.clear()
        flash('Вы вышли из системы', 'success')
        return redirect(url_for('index'))

    @app.route('/formpats.html')
    def formpats():
        cursor = mysql.connection.cursor()
        return render_template('formpats.html')

    @app.route('/add_pats', methods=['POST'])
    def add_pats():
        cursor = mysql.connection.cursor()
        cursor.execute('insert into dannye_pats (nomer_pats, surname,
name, otchestvo, oms, snils) values (%s,%s,%s,%s,%s,%s)',
[request.form['nomer_pats'], request.form['surname'], request.form['name'],
request.form['otchestvo'], request.form['oms'], request.form['snils']])
        mysql.connection.commit()
        cursor.close()
        flash('Запись о пациенте успешно добавлена', 'success')
        return redirect(url_for('formpats'))

    @app.route('/editformpats/<id_pats>', methods=['GET', 'POST'])
    @is_logged_in
    def editformpats(id_pats):
        cur = mysql.connection.cursor()
        result = cur.execute("SELECT * FROM dannye_pats WHERE id_pats=%s",
[id_pats])
        entries = cur.fetchall()
        if request.method == 'POST':
            nomer_pats = request.form['nomer_pats']
            surname = request.form['surname']
            name = request.form['name']
            otchestvo = request.form['otchestvo']
            oms = request.form['oms']
            snils = request.form['snils']
            cur = mysql.connection.cursor()
            cur.execute('update dannye_pats set nomer_pats=%s, surname=%s,
name=%s, otchestvo=%s, oms=%s, snils=%s  where id_pats=%s', (nomer_pats,
surname, name,otchestvo, oms, snils, id_pats))
            entries = cur.fetchone()
            mysql.connection.commit()
            flash("Запись о пациенте успешно обновлена", "success")
```

```
                return redirect(url_for('otchetpats'))
            return render_template('editformpats.html', entries=entries)

        @app.route('/otchetpats')
        @is_logged_in
        def otchetpats():
            cur = mysql.connection.cursor()
            result = cur.execute("SELECT * FROM dannye_pats")
            entries = cur.fetchall()
            if result > 0:
                return render_template('otchetpats.html', entries=entries)
            else:
                msg = 'Нет записей сейчас'
                return render_template('otchetpats.html', msg=msg)
            cur.close()

        @app.route('/delete_entry/<string:id_pats>', methods=['POST'])
        @is_logged_in
        def delete_entry(id_pats):
            cur = mysql.connection.cursor()
            cur.execute("DELETE FROM gen_pats WHERE id_pats = %s", [id_pats])
            cur.execute("DELETE FROM prognoz WHERE id_pats = %s", [id_pats])
            cur.execute("DELETE FROM dannye_pats WHERE id_pats = %s",
[id_pats])
        mysql.connection.commit()
            cur.close()
            flash('Запись о пациенте успешно удалена', 'success')
            return redirect(url_for('otchetpats'))

        @app.route('/otchetgenpats/<string:id_pats>')
        @is_logged_in
        def otchetgenpats(id_pats):
            cur = mysql.connection.cursor()
            name2 = cur.execute("SELECT * FROM dannye_pats WHERE id_pats =
%s", [id_pats])
            names1 = cur.fetchall()
            result = cur.execute("SELECT * FROM otchet_pats WHERE id_pats =
%s", [id_pats])
            entries = cur.fetchall()
            if name2 and result > 0:
                return render_template('otchetgenpats.html', names1=names1,
entries=entries)
            else:
                 msg = 'Нет записей сейчас'
                 return render_template('otchetgenpats.html', names1=names1,
msg=msg)
            cur.close()

        @app.route('/delete_gen/<string:id_pats>', methods=['POST'])
        @is_logged_in
        def delete_gen(id_pats):
            cur = mysql.connection.cursor()
            cur.execute("DELETE FROM gen_pats WHERE id_pats = %s", [id_pats])
        mysql.connection.commit()
            cur.close()
            flash('Запись о генетических данных пациента успешно удалена',
'success')
            return redirect(url_for('otchetpats'))

        @app.route('/otchetanaliz/<string:id_pats>')
        @is_logged_in
        def otchetanaliz(id_pats):
            cur = mysql.connection.cursor()
```

```
            name1 = cur.execute("SELECT * FROM dannye_pats WHERE id_pats =
%s", [id_pats])
            names = cur.fetchall()
            result = cur.execute("SELECT * FROM otchet_analiz WHERE id_pats =
%s", [id_pats])
            entries = cur.fetchall()
            if name1 and result > 0:
                return render_template('otchetanaliz.html', names=names,
entries=entries)
            else:
                msg = 'Нет записей сейчас'
                return render_template('otchetanaliz.html', names=names,
msg=msg)
            cur.close()

        @app.route('/delete_analiz/<string:id_analizpats>', methods=['POST'])
        @is_logged_in
        def delete_analiz(id_analizpats):
            cur = mysql.connection.cursor()
            cur.execute("DELETE FROM analiz_pats WHERE id_analizpats = %s",
[id_analizpats])
        mysql.connection.commit()
            cur.close()
            flash('Запись об анализах пациента успешно удалена', 'success')
            return redirect(url_for('otchetpats'))

        @app.route('/start')
        @is_logged_in
        def start():
            return render_template('start.html')

        @app.route('/formgenes.html')
        def formgenes():
            cursor = mysql.connection.cursor()
            cur = cursor.execute("SELECT ID_pats, nomer_pats, surname, name,
otchestvo FROM dannye_pats")
            patients0=cursor.fetchall()
            cur1 = cursor.execute("SELECT ID_fenotip, naimen_fenotip FROM
fenotypes WHERE ID_gen1=1")
            fenotypeshp = cursor.fetchall()
            cur2 = cursor.execute("SELECT ID_fenotip, naimen_fenotip FROM
fenotypes WHERE ID_gen1=2")
            fenotypesgc = cursor.fetchall()
            cur3 = cursor.execute("SELECT ID_fenotip, naimen_fenotip FROM
fenotypes WHERE ID_gen1=3")
            fenotypestf = cursor.fetchall()
            cur4 = cursor.execute("SELECT ID_fenotip, naimen_fenotip FROM
fenotypes WHERE ID_gen1=4")
            fenotypescs = cursor.fetchall()
            return render_template('formgenes.html', patients0=patients0,
fenotypeshp=fenotypeshp, fenotypesgc=fenotypesgc, fenotypestf=fenotypestf,
fenotypescs=fenotypescs)
        @app.route('/add_fenotype', methods=['POST'])
        def add_fenotype():
            patient0 = request.form['patient0']
            fenotypehp = request.form['fenotypehp']
            fenotypegc = request.form['fenotypegc']
            fenotypetf = request.form['fenotypetf']
            fenotypecs = request.form['fenotypecs']
            cursor = mysql.connection.cursor()
            result = cursor.execute("SELECT * FROM gen_pats WHERE id_pats =
%s", [patient0])
            if result > 0:
```

```
                cursor.execute('UPDATE gen_pats,fenotypes SET
gen_pats.id_fenotip1=%s WHERE gen_pats.id_fenotip1=fenotypes.id_fenotip AND
gen_pats.id_pats=%s AND fenotypes.id_gen1=1', (fenotypehp, patient0))
                cursor.execute('UPDATE gen_pats,fenotypes SET
gen_pats.id_fenotip1=%s WHERE gen_pats.id_fenotip1=fenotypes.id_fenotip AND
gen_pats.id_pats=%s AND fenotypes.id_gen1=2', (fenotypegc, patient0))
                cursor.execute('UPDATE gen_pats,fenotypes SET
gen_pats.id_fenotip1=%s WHERE gen_pats.id_fenotip1=fenotypes.id_fenotip AND
gen_pats.id_pats=%s AND fenotypes.id_gen1=3', (fenotypetf, patient0))
                cursor.execute('UPDATE gen_pats,fenotypes SET
gen_pats.id_fenotip1=%s WHERE gen_pats.id_fenotip1=fenotypes.id_fenotip AND
gen_pats.id_pats=%s AND fenotypes.id_gen1=4', (fenotypecs, patient0))
                cursor.execute('SELECT kod_fenotip FROM fenotypes, gen_pats
WHERE fenotypes.id_fenotip=gen_pats.id_fenotip1 AND gen_pats.id_pats=%s AND
gen_pats.id_fenotip1=%s',[patient0, fenotypehp])
                queryhp = cursor.fetchall()
                masshp = list()
                for t in queryhp:
                    masshp.append(int(t['kod_fenotip']))
                hp = masshp[0]
                cursor.execute('SELECT kod_fenotip FROM fenotypes, gen_pats
WHERE fenotypes.id_fenotip=gen_pats.id_fenotip1 AND gen_pats.id_pats=%s AND
gen_pats.id_fenotip1=%s',[patient0, fenotypegc])
                querygc = cursor.fetchall()
                massgc = list()
                for t in querygc:
                    massgc.append(int(t['kod_fenotip']))
                gc = massgc[0]
                cursor.execute('SELECT kod_fenotip FROM fenotypes, gen_pats
WHERE fenotypes.id_fenotip=gen_pats.id_fenotip1 AND gen_pats.id_pats=%s AND
gen_pats.id_fenotip1=%s',[patient0, fenotypetf])
                querytf = cursor.fetchall()
                masstf = list()
                for t in querytf:
                    masstf.append(int(t['kod_fenotip']))
                tf = masstf[0]
                cursor.execute('SELECT kod_fenotip FROM fenotypes, gen_pats
WHERE fenotypes.id_fenotip=gen_pats.id_fenotip1 AND gen_pats.id_pats=%s AND
gen_pats.id_fenotip1=%s',[patient0, fenotypecs])
                querycs = cursor.fetchall()
                masscs = list()
                for t in querycs:
                    masscs.append(int(t['kod_fenotip']))
                cs = masscs[0]
                net = buildNetwork(4, 15, 5, 1)
                net = NetworkReader.readFrom('net.xml')
                mass = abs(net.activate([hp,gc,tf,cs]))
                a = mass[0]
                cursor.execute('update prognoz set prognoz=%s where
id_pats=%s', (a, patient0))
            else:
                cursor.execute('insert into gen_pats (id_pats, id_fenotip1)
values (%s,%s)', [patient0, fenotypehp])
                cursor.execute('insert into gen_pats (id_pats, id_fenotip1)
values (%s,%s)', [patient0, fenotypegc])
                cursor.execute('insert into gen_pats (id_pats, id_fenotip1)
values (%s,%s)', [patient0, fenotypetf])
                cursor.execute('insert into gen_pats (id_pats, id_fenotip1)
values (%s,%s)', [patient0, fenotypecs])
                cursor.execute('SELECT kod_fenotip FROM fenotypes, gen_pats
WHERE fenotypes.id_fenotip=gen_pats.id_fenotip1 AND gen_pats.id_pats=%s AND
gen_pats.id_fenotip1=%s',[patient0, fenotypehp])
                queryhp = cursor.fetchall()
```

```
                masshp = list()
                for t in queryhp:
                    masshp.append(int(t['kod_fenotip']))
                hp = masshp[0]
                cursor.execute('SELECT kod_fenotip FROM fenotypes, gen_pats
WHERE fenotypes.id_fenotip=gen_pats.id_fenotip1 AND gen_pats.id_pats=%s AND
gen_pats.id_fenotip1=%s',[patient0, fenotypegc])
                querygc = cursor.fetchall()
                massgc = list()
                for t in querygc:
                    massgc.append(int(t['kod_fenotip']))
                gc = massgc[0]
                cursor.execute('SELECT kod_fenotip FROM fenotypes, gen_pats
WHERE fenotypes.id_fenotip=gen_pats.id_fenotip1 AND gen_pats.id_pats=%s AND
gen_pats.id_fenotip1=%s',[patient0, fenotypetf])
                querytf = cursor.fetchall()
                masstf = list()
                for t in querytf:
                    masstf.append(int(t['kod_fenotip']))
                tf = masstf[0]
                cursor.execute('SELECT kod_fenotip FROM fenotypes, gen_pats
WHERE fenotypes.id_fenotip=gen_pats.id_fenotip1 AND gen_pats.id_pats=%s AND
gen_pats.id_fenotip1=%s',[patient0, fenotypecs])
                querycs = cursor.fetchall()
                masscs = list()
                for t in querycs:
                    masscs.append(int(t['kod_fenotip']))
                cs = masscs[0]
                net = buildNetwork(4, 15, 5, 1)
                net = NetworkReader.readFrom('net.xml')
                mass =
abs(net.activate([fenotypehp,fenotypegc,fenotypetf,fenotypecs]))
                a = mass[0]
                cursor.execute('insert into prognoz (id_pats, prognoz) values
(%s,%s)', [patient0, a])
            mysql.connection.commit()
            cursor.close()
            flash('Запись о генетических данных успешно добавлена', 'success')
            return redirect(url_for('formgenes'))
        @app.route('/formanaliz.html')
        def formanaliz():
            cursor = mysql.connection.cursor()
            cur = cursor.execute("SELECT ID_pats, surname FROM dannye_pats")
            patients1=cursor.fetchall()
            cur1 = cursor.execute("SELECT id_analiz, naimen_analiz, edin_izm
FROM analiz")
            analizy = cursor.fetchall()
            return render_template('formanaliz.html', patients1=patients1,
analizy=analizy)
        @app.route('/add_analiz', methods=['POST'])
        def add_analiz():
            cursor = mysql.connection.cursor()
            cursor.execute('insert into analiz_pats (id_pats2, id_analiz1,
date, result) values (%s,%s,%s,%s)', [request.form['patient1'],
request.form['analiz'], request.form['date'], request.form['result']])
           mysql.connection.commit()
            cursor.close()
            flash('Запись об анализах пациента успешно добавлена', 'success')
            return redirect(url_for('formanaliz'))
        if __name__ == '__main__':
           app.secret_key='secret123'
            app.run(debug = True)
```

Файл net.xml

```
<?xml version="1.0" ?>
<PyBrain>
    <Network
class="pybrain.structure.networks.feedforward.FeedForwardNetwork"
name="FeedForwardNetwork-11">
        <name val="'FeedForwardNetwork-11'"/>
        <Modules>
            <LinearLayer
class="pybrain.structure.modules.linearlayer.LinearLayer" inmodule="True"
name="in">
                <dim val="4"/>
                <name val="'in'"/>
            </LinearLayer>
            <LinearLayer
class="pybrain.structure.modules.linearlayer.LinearLayer" name="out"
outmodule="True">
                <dim val="1"/>
                <name val="'out'"/>
            </LinearLayer>
            <BiasUnit
class="pybrain.structure.modules.biasunit.BiasUnit" name="bias">
                <name val="'bias'"/>
            </BiasUnit>
            <SigmoidLayer
class="pybrain.structure.modules.sigmoidlayer.SigmoidLayer" name="hidden0">
                <dim val="15"/>
                <name val="'hidden0'"/>
            </SigmoidLayer>
            <SigmoidLayer
class="pybrain.structure.modules.sigmoidlayer.SigmoidLayer" name="hidden1">
                <dim val="5"/>
                <name val="'hidden1'"/>
            </SigmoidLayer>
        </Modules>
        <Connections>
            <FullConnection
class="pybrain.structure.connections.full.FullConnection"
name="FullConnection-5">
                <inmod val="bias"/>
                <outmod val="out"/>
                <Parameters>[0.075255606900566194]</Parameters>
            </FullConnection>
            <FullConnection
class="pybrain.structure.connections.full.FullConnection"
name="FullConnection-6">
                <inmod val="bias"/>
                <outmod val="hidden0"/>
                <Parameters>[0.23595714690367689, -
0.011755194078374775, -0.37094394987444945, -1.1496731616278391, -
1.0810458967635925, -7.2945365917653344, 0.6437485301950554,
0.61339588855994664, -0.51325611884204847, 0.8776061144363303,
0.40728501351280694, -0.34374237070992375, -1.9832751877165236,
1.4134699321641444, 0.96931638678996679]</Parameters>
            </FullConnection>
            <FullConnection
class="pybrain.structure.connections.full.FullConnection"
name="FullConnection-7">
                <inmod val="bias"/>
                <outmod val="hidden1"/>
```

```
                            <Parameters>[-1.4908768570081745,
0.29524853573874904, 0.51522207465781689, -1.2745630595988078, -
0.13329929036826281</Parameters>
                        </FullConnection>
                        <FullConnection
class="pybrain.structure.connections.full.FullConnection"
name="FullConnection-10">
                            <inmod val="in"/>
                            <outmod val="hidden0"/>
                            <Parameters>[1.1946171006012716, -
1.2511692027280934, -1.6685141774898398, -0.93337676724111052, -
0.54457096980706576, 0.47102235440585732, -0.87467684320916439,
3.8781965899624624, 1.1149162584441759, -1.10331477677189,
1.2471387043119786, 0.48332073154316607, 0.72914268679790339,
0.11057886351856713, -1.2195730891942522, 1.1182810050617289, -
0.72365803453633115, 0.10828687229937133, 2.4526585692823093, -
0.53690886695844109, 0.16074007178834318, -0.077967340414361208,
9.1638624889781184, -6.5112350712158484, -0.63528414187818738, -
0.89929240758227769, -0.80915050067740335, -0.051117948963902475,
2.2355469006910025, 0.54086476008001749, 1.54657106048725, -
0.89841931621166748, 1.8479188226679681, -0.23379668640859827,
0.80348235566604753, -0.007392838563661343, 0.87502551071867907, -
0.57371850921714341, 1.78358988166001, -1.6532585468115681, -
0.92108106559141223, -0.74311644338040661, 0.5176226070999288,
2.0101683393906375, -0.25804507543903965, 1.0475159582140854, -
2.4365291307881916, 0.32193459327302698, -2.4653964010084217, -
1.8702183225130216, -0.74153609380737162, 1.9256563051346081, -
1.1530133564584453, -1.3534087109407724, 0.85822885760023349, -
1.1748795209271834, 0.18458978721372044, -0.40395080813213818, -
1.4830485699445886, -0.72460342653594989]</Parameters>
                        </FullConnection>
                        <FullConnection
class="pybrain.structure.connections.full.FullConnection"
name="FullConnection-9">
                            <inmod val="hidden0"/>
                            <outmod val="hidden1"/>
                            <Parameters>[-1.9830275476096226, -
1.7756134797135052, -0.82289457398369736, -0.17739012900343387,
0.71455542091701418, 1.3653368284161256, 0.30155691166727133, -
0.044449949649000016, 0.22168421503714653, -0.40494247967861879, -
1.0085335533744559, -1.0098342720347406, -0.54174493517352273, -
0.63642366859425248, -0.89336364629742193, 0.17221133878121617, -
0.59170653182764466, -2.2114082694437593, -0.72345938182478098, -
0.53373312191025124, -1.1707123528139529, -0.55798278452014238,
0.32226677828614164, -0.37570718450439189, 0.77837353348986937,
0.53903217708302809, -0.19114787187970395, 0.94548104387747445,
0.67724986983396251, -0.20712795243153376, -1.4040417463855299, -
0.14013474553683788, 0.98353826654983201, -0.56863979500650563, -
2.9032769285711875, -1.9645397691247861, 0.092417250032365339, -
1.139745154042753, -2.4032407211477187, 0.35267038217278995,
0.90925368799578388, -1.1360679040963677, -0.09835436147938564, -
1.3880096271648092, -0.852025672281985, -0.004353916662686366,
0.56672859512523543, 0.49254202842537426, 1.4433964346467507, -
1.6001522189360586, 0.50553260864103233, 0.59381228785081697, -
1.3721278670650425, 0.7652737554427258, -1.4731074637714046, -
0.65848582032422009, -0.62208813019289555, -0.78775963695422013,
0.24948165756186266, -1.1270809317136083, 0.47371394695320301,
0.23477454496700229, 1.6185872967531083, -2.0472059463828893,
0.96104183119368936, -5.8250729749989434, -1.7705874633365679,
0.79587383577803728, -0.077728759511264622, 1.9473966146322814, -
0.61823094163872871, 0.72189267596704043, -3.9362151315817218, -
1.120423040738693, 0.64156066165128012]</Parameters>
                        </FullConnection>
```

```
                        <FullConnection
class="pybrain.structure.connections.full.FullConnection"
name="FullConnection-8">
                                <inmod val="hidden1"/>
                                <outmod val="out"/>
                                <Parameters>[0.2600927430661823, -
1.5045529013096823, 1.0720391456062137, -0.78469497262888643,
1.9459227035844981]</Parameters>
                        </FullConnection>
                </Connections>
            </Network>
        </PyBrain>>
```

Файл home.html

```
{% extends 'layout.html' %}

{% block body %}
<div class="jumbotron text-center">
     <h2>Добро пожаловать в систему</h2>
      <p class="lead">Для продолжения работы Вам необходимо войти в
систему</p>
      {% if session.logged_in == NULL %}
      <a href="/login" class="btn btn-success btn-lg">Войти в
систему</a>
    {% endif %}
</div>
{% endblock %}
```

Файл login.html

```
{% extends 'layout.html' %}

{% block body %}

  <div class="login">
  <h1>Вход в систему</h1>
  <form action="" class="form-signin" method="POST">
      <label for="username" class="sr-only">Имя пользователя</label>
      <input type="text" name="username" id="username" class="form-
control" placeholder="Имя пользователя" value={{request.form.username}}>
      <label for="password" class="sr-only">Password</label>
      <input type="password" name="password" id="password"
class="form-control" placeholder="Пароль" value={{request.form.password}}>

        <button class="btn btn-lg btn-primary btn-block"
type="submit">Войти в систему</button>
           </div>
      </form>
{% endblock %}
```

Файл start.html

```
{% extends 'layout.html' %}

{% block body %}
    <h1>Главное меню</h1>
     <ul class="list-group">
         <li class="list-group-item"><a href="/formpats.html">Ввести
данные о пациенте</a></li>
          <li class="list-group-item"><a href="/formgenes.html">Ввести
генетические данные пациента</a></li>
          <li class="list-group-item"><a
href="/formanaliz.html">Ввести данные об анализах пациента</a></li>
          <li class="list-group-item"><a href="/otchetpats">Список
пациентов</a></li>
      </ul>
{% endblock %}
```

Файл formpats.html

```
        {% extends 'layout.html' %}
        {% block body %}
        {% if session.logged_in %}
        <div class="form">
        <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jquery.mask/1.14.13/jquery.mask.m
in.js"></script>
        <h1>Введите данные о пациенте</h1>
          <form action="{{ url_for('add_pats') }}" method="POST">
            <div class="form-group">
              <label>Номер пациента</label>
              <input type="text" name="nomer_pats" class="form-control">
            </div>
             <div class="form-group">
              <label>Фамилия</label>
              <input type="text" name="surname" class="form-control">
            </div>
             <div class="form-group">
              <label>Имя</label>
              <input type="text" name="name" class="form-control">
            </div>
             <div class="form-group">
              <label>Отчество</label>
              <input type="text" name="otchestvo" class="form-control">
            </div>
             <div class="form-group">
              <label>Номер полиса ОМС</label>
              <input type="text" name="oms" id="oms"
placeholder="1234567890123456" class="form-control oms">
            </div>
             <div class="form-group">
              <label>СНИЛС</label>
              <input type="text" name="snils" id="snils" placeholder="123-456-
789-01" class="form-control snils">
            </div>
            <button type="submit" class="btn btn-
primary">Сохранить</button><br>
          </form>
            <div>
             <a href="/start"> <button type="submit" class="btn btn-primary">В
главное меню</button></a>
             </div>
            <script>
        $(document).ready(function(){
           $('.oms').mask("9999999999999999", {placeholder:
"1234567890123456"});
           $('.snils').mask("999-999-999-99", {placeholder: "123-456-789-
01"});
        });
            </script>
        </div>
         {% endif %}
        {% endblock %}
```

Файл formgenes.html

```
        {% extends "layout.html" %}
        {% block body %}
          {% if session.logged_in %}
          <div class="form">
            <h1>Введите генетические данные пациента</h1>
             <form action="{{ url_for('add_fenotype') }}" method="POST">
                   <div class="form-group">
                   <label for="patient0">ФИО пациента:</label>
                   <select name="patient0" class="form-control">
                   {% for patient in patients0 %}
                 <option value="{{ patient.ID_pats
}}">{{patient.nomer_pats}}.{{ patient.surname }} {{patient.name}}
{{patient.otchestvo}}</option>
                 {% endfor %}
                   </select>
                   </div>
                   <div class="form-group">
                   <label for="fenotypehp">Hp:</label>
                   <select name="fenotypehp" class="form-control">
                   {% for fenotypehp in fenotypeshp %}
                 <option value="{{ fenotypehp.ID_fenotip }}">{{
fenotypehp.naimen_fenotip }}</option>
                 {% endfor %}
                   </select>
                   </div>
                   <div class="form-group">
                   <label for="fenotypegc">Gc:</label>
                   <select name="fenotypegc" class="form-control">
                   {% for fenotypegc in fenotypesgc %}
                 <option value="{{ fenotypegc.ID_fenotip }}">{{
fenotypegc.naimen_fenotip }}</option>
                 {% endfor %}
                 </select>
                   </div>
                   <div class="form-group">
                   <label for="fenotypetf">Tf:</label>
                   <select name="fenotypetf" class="form-control">
                   {% for fenotypetf in fenotypestf %}
                 <option value="{{ fenotypetf.ID_fenotip }}">{{
fenotypetf.naimen_fenotip }}</option>
                 {% endfor %}
                 </select>
                   </div>
                   <div class="form-group">
                   <label for="fenotypecs">C's:</label>
                   <select name="fenotypecs" class="form-control">
                   {% for fenotypecs in fenotypescs %}
                 <option value="{{ fenotypecs.ID_fenotip }}">{{
fenotypecs.naimen_fenotip }}</option>
                 {% endfor %}
                 </select>
                   </div>
               <button type="submit" class="btn btn-
primary">Сохранить</button>   <br>
            </form>
            <a href="/start"> <button type="submit" class="btn btn-primary">В
главное меню</button></a>
             </div>
          {% endif %}
        {% endblock %}
```

Файл formanaliz.html

```
        {% extends "layout.html" %}
        {% block body %}
          {% if session.logged_in %}
            <div class="form">
            <h1>Введите данные об анализах пациента</h1>
             <form action="{{ url_for('add_analiz') }}" method="POST">
                   <div class="form-group">
                   <label for="patient1">Фамилия пациента:</label>
                   <select name="patient1" class="form-control">
                   {% for patient in patients1 %}
                  <option value="{{ patient.ID_pats }}">{{ patient.surname
}}</option>
                  {% endfor %}
                   </select>
                   </div>
                   <div class="form-group">
                   <label for="analiz">Вид анализа:</label>
                   <select name="analiz" class="form-control">
                   {% for analiz in analizy %}
                  <option value="{{ analiz.id_analiz }}">{{ analiz.naimen_analiz
}}, {{ analiz.edin_izm }}</option>
                  {% endfor %}
                   </select>
                   </div>
                   <div class="form-group">
                   <label for="date">Дата сдачи анализа:</label>
                   <input type="date" name="date" class="form-control">
                   </div>
                   <div class="form-group">
                   <label for="result">Результат анализа:</label>
                   <input type="text" name="result" class="form-control">
                   </div>
                  <button type="submit" class="btn btn-
primary">Сохранить</button><br>
             </form>
             <a href="/start"> <button type="submit" class="btn btn-primary">В
главное меню</button></a>
            </div>
          {% endif %}{% endblock %}
```

Файл otchetpats.html

```
        {% extends 'layout.html' %}
        {% block body %}
          <h1 class = "header">Список пациентов</h1>
          <a class="btn btn-success" href="/formpats.html">Добавить данные о
пациенте</a>
          <hr>
          <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
          <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
          <link rel="stylesheet"
href="https://cdn.datatables.net/buttons/1.5.1/css/buttons.dataTables.min.css
">
       <script type="text/javascript" language="javascript"
src="https://code.jquery.com/jquery-1.12.4.js"></script>
            <script type="text/javascript" language="javascript"
src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script
>
            <script type="text/javascript" language="javascript"
src="https://cdn.datatables.net/1.10.16/js/dataTables.bootstrap.min.js"></scr
ipt>
            <script type="text/javascript" language="javascript"
src="https://cdn.datatables.net/buttons/1.5.1/js/dataTables.buttons.min.js"><
/script>
            <script type="text/javascript" language="javascript"
src="https://cdn.datatables.net/buttons/1.5.1/js/buttons.bootstrap.min.js"></
script>
            <script type="text/javascript" language="javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></scrip
t>
            <script type="text/javascript" language="javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.32/pdfmake.min.js"></
script>
            <script type="text/javascript" language="javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.32/vfs_fonts.js"></sc
ript>
            <script type="text/javascript" language="javascript"
src="https://cdn.datatables.net/buttons/1.5.1/js/buttons.html5.min.js"></scri
pt>
            <script type="text/javascript" language="javascript"
src="https://cdn.datatables.net/buttons/1.5.1/js/buttons.print.min.js"></scri
pt>
            <script type="text/javascript" language="javascript"
src="https://cdn.datatables.net/buttons/1.5.1/js/buttons.colVis.min.js"></scr
ipt>
            <style>
            .buttons-pdf {
          background-color: #EAEAEA;
          color: #333333;
        }
        .buttons-excel {
          background-color: #217346;
          color: white;
        }
            </style>
        <script>
         $(document).ready(function() {
            var table = $('#tablepats').DataTable( {
             language: {
```

```
            "url": "//cdn.datatables.net/plug-ins/1.10.16/i18n/Russian.json"
             },
             dom: 'Bfrtip',
              columnDefs: [
          { targets: 'no-sort', orderable: false }],
               buttons: [{
                         extend: 'excel',
                                title: 'Список пациентов',
                                text: '<i class="fa fa-file-excel-o"></i>
Excel',
                                titleAttr: 'Excel',
                         exportOptions: {
                             columns: ':visible'
                                     }
                         },
                                          {
                         extend: 'pdf',
                                title: 'Список пациентов',
                                text: '<i class="fa fa-file-pdf-o"></i> PDF',
                                titleAttr: 'PDF',
                         exportOptions: {
                             columns: ':visible'
                                     }
                     }
                      ]
        })
        } );
            table.buttons().container()
                .appendTo( '#tablepats_wrapper .col-sm-6:eq(0)' );


        </script>

         <table class="table table-striped tablepats" id="tablepats">
            <thead>
            <tr>
              <th>№ пац.</th>
              <th>Фамилия</th>
              <th>Имя</th>
              <th>Отчество</th>
              <th>Номер полиса ОМС</th>
               <th>СНИЛС</th>
               <th class = "no-sort"></th>
               <th class = "no-sort"></th>
               <th class = "no-sort"></th>
               <th class = "no-sort"></th>
            </tr>
             </thead>
            {% for entry in entries %}
              <tr>
                <td>{{entry.nomer_pats}}</td>
                <td>{{entry.surname}}</td>
                <td>{{entry.name}}</td>
                   <td>{{entry.otchestvo}}</td>
                   <td>{{entry.oms}}</td>
                   <td>{{entry.snils}}</td>
                <td><a href="otchetgenpats/{{entry.id_pats}}" class="btn btn-
default pull-right">Ген. данные</a></td>
                   <td><a href="otchetanaliz/{{entry.id_pats}}" class="btn btn-
default pull-right">Анализы</a></td>
                   <td><a href="editformpats/{{entry.id_pats}}" class="btn btn-
default pull-right edit_data" style="padding-bottom: 10px; padding-top:
8px;"><span class="glyphicon glyphicon-pencil"></span></a></td>
```

```
                <td>
                <form action="{{url_for('delete_entry',
id_pats=entry.id_pats)}}" method="post">
                    <input type="hidden" name="_method" value="DELETE">
                    <button type="submit" value="" class="btn btn-danger"
style="padding-bottom: 10px; padding-top: 8px;"><span class="glyphicon
glyphicon-trash"></span></button>
                </form>
              </td>
            </tr>
          {% endfor %}
        </table>
        <a href="/start"> <button type="submit" class="btn btn-primary">В
главное меню</button></a><br><hr>
        {% endblock %}
```

Файл otchetgenpats.html

```
        {% extends 'layout.html' %}
        {% block body %}
          {% for name in names1 %}
          <h1>Ген. данные пациента: {{name.surname}} {{name.name}}
{{name.otchestvo}}</h1>
          {% endfor %}
          <a class="btn btn-success" href="/formgenes.html">Добавить ген.
данные о пациенте</a>
          <hr>
          <table class="table table-striped" method="POST">
            <tr>
              <th>Hp</th>
              <th>Gc</th>
              <th>Tf</th>
              <th>C's</th>
               <th>Прогноз пациента</th>
               <th></th>
            </tr>
            {% for entry in entries %}
              <tr>
                <td>{{entry.Hp}}</td>
                <td>{{entry.Gc}}</td>
                <td>{{entry.Tf}}</td>
                   <td>{{entry.Cs}}</td>
                   <td>{{entry.prognoz}}</td>
                   <td>
                  <form action="{{url_for('delete_gen',
id_pats=entry.id_pats)}}" method="post">
                    <input type="hidden" name="_method" value="DELETE">
                    <input type="submit" value="Удалить" class="btn btn-danger">
                     </form>
                </td>
              </tr>
            {% endfor %}
          </table>
          <a href="/otchetpats"> <button type="submit" class="btn btn-
primary">Назад</button></a><br><br>
          <a href="/start"> <button type="submit" class="btn btn-primary">В
главное меню</button></a>{% endblock %}
```

Файл otchetanaliz.html

```
        {% extends 'layout.html' %}
        {% block body %}
          {% for name in names %}
          <h1>Анализы пациента: {{name.surname}} {{name.name}}
{{name.otchestvo}}</h1>
          {% endfor %}
          <a class="btn btn-success" href="/formanaliz.html">Добавить данные
об анализах пациентов</a>
          <hr>
          <table class="table table-striped">
            <tr>
              <th>ID_анализа</th>
              <th>Дата сдачи анализа</th>
              <th>Вид анализа</th>
               <th>Результат</th>
              <th></th>
               <th></th>
            </tr>
            {% for entry in entries %}
              <tr>
                <td>{{entry.id_analizpats}}</td>
                <td>{{entry.date}}</td>
                <td>{{entry.naimen_analiz}}</td>
                   <td>{{entry.result}}</td>
                   <td>{{entry.edin_izm}}</td>
                <td>
                 <form action="{{url_for('delete_analiz',
id_analizpats=entry.id_analizpats)}}" method="post">
                  <input type="hidden" name="_method" value="DELETE">
                  <input type="submit" value="Удалить" class="btn btn-danger">
                    </form>
                </td>
              </tr>
            {% endfor %}
          </table>
          <a href="/otchetpats"> <button type="submit" class="btn btn-
primary">Назад</button></a><br><br>
          <a href="/start"> <button type="submit" class="btn btn-primary">В
главное меню</button></a>{% endblock %}
```

Файл editformpats.html

```
        {% extends 'layout.html' %}
        {% block body %}
        {% if session.logged_in %}
        <div class="form">
        {% for entry in entries %}
        <h1>Измените данные о пациенте</h1>
          <form action="{{ url_for('editformpats', id_pats=entry.id_pats) }}"
method="POST">
            <div class="form-group">
              <label>Номер пациента</label>
              <input type="text" name="nomer_pats" id="nomer_pats"
value="{{entry.nomer_pats}}" class="form-control">
            </div>
             <div class="form-group">
              <label>Фамилия</label>
              <input type="text" name="surname" value="{{entry.surname}}"
class="form-control">
            </div>
             <div class="form-group">
              <label>Имя</label>
              <input type="text" name="name" value="{{entry.name}}"
class="form-control">
            </div>
             <div class="form-group">
              <label>Отчество</label>
              <input type="text" name="otchestvo" value="{{entry.otchestvo}}"
class="form-control">
            </div>
             <div class="form-group">
              <label>Номер полиса ОМС</label>
              <input type="text" name="oms" id="oms" value="{{entry.oms}}"
class="form-control oms">
            </div>
             <div class="form-group">
              <label>СНИЛС</label>
              <input type="text" name="snils" id="snils"
value="{{entry.snils}}" class="form-control snils">
            </div>
            <button type="submit" class="btn btn-
primary">Сохранить</button><br>
          </form>
            <div>
             {% endfor %}
             <a href="/otchetpats"> <button type="submit" class="btn btn-
primary">Назад</button></a><br><br>
             <a href="/start"> <button type="submit" class="btn btn-primary">В
главное меню</button></a>
             </div>
            <script>
        $(document).ready(function(){
           $('.oms').mask("9999999999999999", {placeholder:
"1234567890123456"});
           $('.snils').mask("999-999-999-99", {placeholder: "123-456-789-
01"});
        });
            </script>
        </div>
         {% endif %}
        {% endblock %}
```

Файл layout.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
     <title>Рестеноз</title>
<script type="text/javascript" language="javascript"
src="http://code.jquery.com/jquery-2.2.4.js"></script>
<script type="text/javascript" language="javascript"
src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script
>
<script type="text/javascript" language="javascript"
src="https://cdn.datatables.net/1.10.16/js/dataTables.bootstrap.min.js"></scr
ipt>
<link rel="stylesheet" type="text/css"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.16/css/dataTables.bootstrap.min.css">
<style>
   .login {
    width: 70%; /* Ширина */
    margin:0 14% 0 14%;}
   .form {
    width: 70%; /* Ширина */
    margin:0 14% 0 14%;}
   .header {
   margin-top: 0px;}
   .col-sm-12 {
    padding-left: 10px;
    padding-right: 10px;}
</style>
</head>
<body>
    {% include 'includes/_navbar.html' %}
     <div class="container">
     {% include 'includes/_messages.html' %}
    {% block body %}{% endblock %}
     </div>
  </body>
</html>
```

Файл _formhelpers.html

```
{% macro render_field(field) %}
  {{ field.label }}
  {{ field(**kwargs)|safe }}
  {% if field.errors %}
    {% for error in field.errors %}
      <span class="help-inline">{{ error }}</span>
    {% endfor %}
  {% endif %}
{% endmacro %}
```

Файл _messages.html

```
{% with messages = get_flashed_messages(with_categories=true) %}
  {% if messages %}
    {% for category, message in messages %}
      <div class="alert alert-{{ category }}">{{ message }}</div>
    {% endfor %}
  {% endif %}
{% endwith %}
{% if error %}
  <div class="alert alert-danger">{{error}}</div>
{% endif %}
{% if msg %}
  <div class="alert alert-success">{{msg}}</div>
{% endif %}
```

Файл _navbar.html

```
        <nav class="navbar navbar-default">
          <div class="container">
            <div class="navbar-header">
               <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#navbar" aria-expanded="false" aria-
controls="navbar">
                <span class="sr-only">Toggle Navigation</span>
                 <span class="icon-bar"></span>
                 <span class="icon-bar"></span>
                 <span class="icon-bar"></span>
                </button>
              {% if session.logged_in %}
              <a class="navbar-brand" href="/start">Система</a>
              {% else %}
              <a class="navbar-brand" href="/">Система</a>
              {% endif %}
              </div>
              <div id="navbar" class="collapse navbar-collapse">
                <ul class="nav navbar-nav">
                  {% if session.logged_in %}
                    {% else %}
                  <li><a href="/">Домашняя страница</a></li>
                    {% endif %}
                </ul>
              <ul class="nav navbar-nav navbar-right">
                 {% if session.logged_in %}
                    <li><a href="/logout">Выйти из системы</a></li>
                    {% else %}
                  <li><a href="/login">Войти в систему</a></li>
                  {% endif %}
              </div>
           </div>
        </nav>
```