Содержание

[ВВЕДЕНИЕ 2](#_Toc74028428)

[ГЛАВА 1. РАЗРАБОТКА ПРОЕКТА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ 4](#_Toc74028429)

[1. Анализ предметной области 4](#_Toc74028430)

[2. Глоссарий проекта 4](#_Toc74028431)

[3. Цель и постановка проблемы исследования 5](#_Toc74028432)

[4. Анализ существующих аналогов разработок 6](#_Toc74028433)

[5. Функциональные и нефункциональные требования к системе. 7](#_Toc74028434)

[6. Определение действующих лиц системы 8](#_Toc74028435)

[7. Описание вариантов использования 8](#_Toc74028436)

[8. ER-модель базы данных системы 21](#_Toc74028437)

[9. Выбор инструментария для выполнения работы 24](#_Toc74028438)

[10. Внешний вид разрабатываемой системы. 26](#_Toc74028439)

[ГЛАВА 2. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ 34](#_Toc74028440)

[1. Описание содержания проекта 34](#_Toc74028441)

[2. Описание программной части системы 36](#_Toc74028442)

[ЗАКЛЮЧЕНИЕ 41](#_Toc74028443)

# ВВЕДЕНИЕ

Данная работа посвящена разработке веб-приложения по подбору курса диеты, основной идеей которого является автоматизация формирования рекомендации по оптимизации питания.

Актуальность работы обусловлена реалиями настоящего времени, потому как в настоящее время ожирение стало одной из наиболее важных медико-социальных проблем в мире в связи с его высокой распространенностью.

За последние три десятилетия распространенность избыточной массы тела и ожирения в мире выросла почти на 30–50% среди взрослых и детей соответственно. Избыточная масса тела и ожирение негативно влияют на все сферы человеческой деятельности, что приводит к развитию сопутствующих заболеваний.

Однако, достаточно сложной задаче в составлении диеты является планирование своего рациона или его корректировка, особенно, если вы не диетолог. Именно поэтому существует огромное количество давно подобранных диет, но питание это достаточно индивидуальное дело, и подбирать его необходимо только из личных характеристик человека, учитывая его непереносимость каких-либо продуктов, делая расчет дневной нормы калорий и тому подобное. Именно поэтому для индивидуального подбора диеты необходимо потратить немалое количество времени.

Целью разрабатываемого веб-приложения является обеспечение:

* эффективной постановки диеты
* более комфортного соблюдения рекомендаций по питанию

В данной работе будет рассматриваться поэтапное проектирование, а затем и реализация веб-сервиса по подбору диет.

ГЛАВА 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

1. Анализ предметной области

Предметной областью данного проекта является сфера диетологии. И, так как правильно построенная система питания в состоянии оказать терапевтическое воздействие, как на пораженный орган, так и на весь организм в целом - диетотерапия требует индивидуального подхода.

Разрабатываемый веб-сервис позволит подобрать частный рацион питания, исходя из индивидуальных критериев пациента, таких как: возраст, пол и других показателей больного. При этом для удобства сервис формирует дневник приема пищи, при желании пациент, исходя из рекомендаций, поставленных системой, в праве изменять приемы пищи.

1. Глоссарий проекта

Таблица 1 – Глоссарий проекта

|  |  |
| --- | --- |
| Диетология | Область знания, изучающая вопросы питания, в том числе больного человека. |
| Диетотерапия | Лечебный метод, заключающийся в терапии различных заболеваний специальной диетой. |
| Рацион питания,  диета | Количественная и качественная характеристика питания, включающая кратность, время приёма пищи и распределение её по калорийности и химическому составу, а также поведение человека во время приёма пищи. |
| Дневник питания | Индивидуально составленное расписание приема пищи. |
| Диетологическая карта пациента | Документ, в котором диетолог ведёт запись необходимой информации о пациенте. |
| Система | Автоматизирует процесс постановки индивидуального рациона питания. |
| Администратор | Пользователь системы, который ведет информацию о блюдах и продуктах, ведет информацию о пользователях системы |
| Диетолог | Пользователь системы, который ведет карту пациента, корректирует составленный сервисом рацион питания |
| Пациент | Пользователь системы, цель которого – получить индивидуально подобранный рацион питания. |

1. Цель и постановка проблемы исследования

Целью исследования является создание системы, которая будет совершать сбор, накопление и анализ данных о пациентах, назначение питания пациентам, контроль и замена блюд на эквивалентные блюда. Данные хранятся централизованно на сервере под управлением СУБД, что позволит повысить достоверность и надежность информации.

Модернизация рассмотренной предметной области позволит сократить время, затрачиваемое на обработку данных, также позволит более эффективно вести работу по хранению информации и составлению системы питания.

Объектом исследования является сфера диетологии, цель которой нормализация обмена веществ, оздоровление организма, а также корректировка веса пациента.

Предметом исследования является подбор системы питания, которая является неотъемлемой частью процесса выздоровлении.

Основной задачей дипломного проекта является: разработка и автоматизация системы «Диета» для учета питания всех пациентов.

Основой задачей разрабатываемого приложения является подбор системы питания для каждого пациента в зависимости от его заболевания и диагноза, поставленного врачом.

1. Анализ существующих аналогов разработок

Пищевой рацион современного человека является тем важнейшим фактором, который напрямую определяет его здоровье, поэтому он должен формироваться на основе принципов рационального питания. При этом структура питания каждого человека – это его самостоятельный выбор, который отражает экономические возможности, привычки, а также элементарную грамотность и информированность в этих вопросах, но, зачастую, от переизбытка тех самых возможностей человек питается не совсем правильно.

В век информационных технологий человечество доверило всевозможным сервисам следить за их питанием.

Среди таких сервисов можно выделить Nutrilogic - это облачный сервис для оптимизации питания пациента, регистрация в котором занимает не более 2 минут.

После чего необходимо ввести все необходимые данные, вплоть до энергетического обмена. Далее сервис подберет для Вас индивидуальный рацион питания.

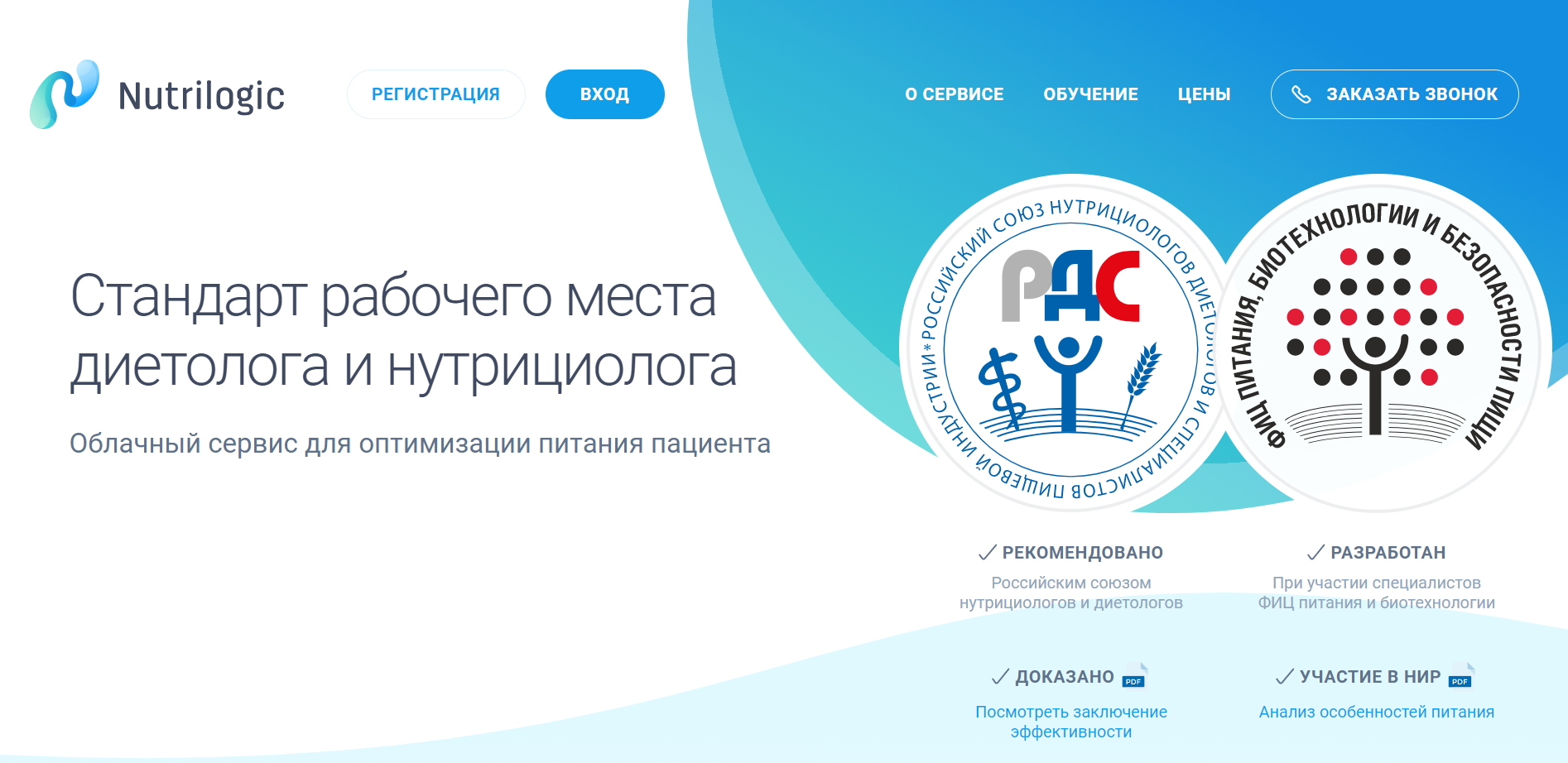
Сервис выглядит следующим образом: (Рисунок 1)

Рисунок 1 Сервис Nutrilogic

1. Функциональные и нефункциональные требования к системе.

К функциональным требованиям данной системы можно отнести следующее:

* подбор индивидуального рациона питания;
* формирование дневника приемов пищи;

К нефункциональным требованиям данной системы можно отнести следующее:

* присутствие интуитивно-понятного для пользователя интерфейса;
* хранение паролей пользователей в зашифрованном виде;
* исключение дублирования логинов пользователей;
* реализация запросов к базе данных должна непосредственно производиться в классах моделей;
* реализация системы должна быть выполнена с использованием модели MVC (Model-View-Controller)

1. Определение действующих лиц системы

Разрабатываемый сервис включает в себя следующих действующих лиц:

* Система – получает информацию о пациенте, составляет индивидуально подобранный рацион питания;
* Администратор – ведет информацию о блюдах и продуктах, ведет информацию о пользователях системы;
* Диетолог - ведет карту пациента, корректирует составленную сервисом рацион питания;
* Пациент-ведет дневник питания.

1. Описание вариантов использования

Исходя из вышеперечисленных действующих лиц системы можно составить следующий список вариантов использования:

* Войти в систему
* Вести учетные записи пациентов
* Вести учетные записи сотрудников
* Вести информацию о блюдах и продуктах
* Вести карту пациента
* Вести меню пациента
* Вести дневник питания

Подробное описание вариантов использования:

* Вариант использования " Войти в систему"

Краткое описание:

Данный вариант использования описывает вход пользователя в систему.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь хочет воспользоваться сервисом.

* Система генерирует стартовую страницу, в которой запрашивает логин и пароль.
* Пользователь вводит логин и пароль.
* Система подтверждает логин, пароль, определяет пользователя и его роль
* После чего открывается доступ к допустимым для данной роли данным.

Альтернативные потоки:

Пользователь ввел неверные логин или пароль. При неправильных логине и пароле система выведет сообщение о том, что пользователь ввел неверные логин или пароль. В этом случае пользователь вернется к пункту 2 основного потока.

Предусловие:

Необходимо знать логин и пароль пользователя

* Вариант использования " Вести учетные записи о пациентах"

Краткое описание:

Данный вариант использования позволяет администратору вносить, обновлять, удалять информацию о пациентах.

Основной поток событий:

* Система генерирует новое окно с выбором раздела:
* Пациенты
* Сотрудники
* Блюда и продукты
* Администратор выбирает раздел «Пациенты»
* Система генерирует новое окно со списком Пациентов

Внести данные:

* Администратор выбирает действие «Добавить»
* Система генерирует новое окно для ввода данных.
* Администратор вводит необходимую информацию
* Администратор подтверждает свое действие.
* Система проверяет введенную информацию.
* Система сохраняет введенные данные.

Изменить данные:

* Администратор выбирает строку с пользователем, информацию о котором необходимо изменить и выбирает действие «Подробнее»
* Система генерирует окно для изменения информации
* Администратор изменяет необходимую информацию.
* Администратор подтверждает свое действие.
* Система проверяет введенные данные.
* Система сохраняет введенные данные.

Удалить данные:

* Администратор выбирает необходимую строку с пользователем и выбирает действие «Удалить».
* Система удаляет данные.

Альтернативные потоки:

Внести данные:

* Введены данные неверного формата.
* Логин занят
* В этих случаях система выдаст сообщение об ошибке. После чего система перейдет на 2 пункт данного подчиненного потока.

Изменить данные:

Введены данные неверного формата. В этом случая выдается сообщение об ошибке. После чего система перейдет на 2 пункт данного подчиненного потока.

Удалить данные:

Удаляемый пользователь в данный момент находится в системе. В этом случае система не удалит данного пользователя.

Предусловия:

Администратору необходимо войти в систему.

* Вариант использования " Вести учетные записи о сотрудниках"

Краткое описание:

Данный вариант использования позволяет администратору вносить, обновлять, удалять информацию о сотрудниках.

Основной поток событий:

* Система генерирует новое окно с выбором раздела:
* Пациенты
* Сотрудники
* Блюда и продукты
* Администратор выбирает раздел «Сотрудники
* Система генерирует новое окно со списком Сотрудников

Внести данные:

* Администратор выбирает действие «Добавить»
* Система генерирует новое окно для ввода данных.
* Администратор вводит необходимую информацию
* Администратор подтверждает свое действие.
* Система проверяет введенную информацию.
* Система сохраняет введенные данные.

Изменить данные:

* Администратор выбирает строку с пользователем, информацию о котором необходимо изменить и выбирает действие «Подробнее»
* Система генерирует окно для изменения информации
* Администратор изменяет необходимую информацию.
* Администратор подтверждает свое действие.
* Система проверяет введенные данные.
* Система сохраняет введенные данные.

Удалить данные:

* Администратор выбирает необходимую строку с пользователем и выбирает действие «Удалить».
* Система удаляет данные.

Альтернативные потоки:

Внести данные:

* Введены данные неверного формата.
* Логин занят
* В этих случаях система выдаст сообщение об ошибке. После чего система перейдет на 2 пункт данного подчиненного потока.

Изменить данные:

Введены данные неверного формата. В этом случая выдается сообщение об ошибке. После чего система перейдет на 2 пункт данного подчиненного потока.

Удалить данные:

Удаляемый пользователь в данный момент находится в системе. В этом случае система не удалит данного пользователя.

Предусловия:

Администратору необходимо войти в систему.

* Вариант использования “Вести информацию о блюдах и продуктах”

Краткое описание:

Данный вариант использования позволяет администратору вносить, обновлять, удалять информацию о блюдах и продуктах.

Основной поток событий:

* Система генерирует новое окно с выбором раздела:
* Пациенты
* Сотрудники
* Блюда и продукты
* Администратор выбирает раздел «Блюда и продукты»
* Система генерирует новое окно со списками блюд и продуктов и запрашивает действие.

Внести данные:

* Администратор выбирает действие «Добавить» (блюдо или продукт)
* Система генерирует новое окно для ввода данных.
* Администратор вводит всю необходимую информацию.
* Администратор подтверждает свое действие.
* Система проверяет введенную информацию.
* Система сохраняет введенные данные.

Изменить данные:

* Администратор выбирает строку с блюдом или продуктом, информацию о котором необходимо изменить и выбирает действие «Подробнее»
* Система генерирует окно для изменения информации
* Администратор вводит всю необходимую информацию.
* Администратор подтверждает свое действие.
* Система проверяет введенные данные.
* Система сохраняет введенные данные.

Удалить данные:

* Администратор выбирает необходимую строку с блюдом или продуктом и выбирает действие «Удалить».
* Система удаляет данные.

Альтернативные потоки:

Внести данные:

Введены данные неверного формата. В этом случае система выдаст сообщение об ошибке. После чего система перейдет на 2 пункт данного подчиненного потока.

Изменить данные:

Введены данные неверного формата. В этом случае выдается сообщение об ошибке. После чего система перейдет на 2 пункт данного подчиненного потока.

Удалить данные:

Удаляемый продукт входит в состав какого-либо блюда.

Предусловия:

Администратору необходимо войти в систему.

* Вариант использования " Вести карту пациента"

Краткое описание:

Данный вариант использования описывает ведение карты пациента.

Основной поток событий:

* Система генерирует новое окно со списком пациентов, наблюдающихся у данного диетолога

Изменять:

* Диетолог выбирает пациента из списка и выбирает раздел «Карта».
* Система генерирует окно с картой пациента.
* Диетолог редактирует всю необходимую информацию.
* Диетолог подтверждает свое действие.
* Система обновляет информацию в карте пациента.
* Система генерирует окно со списком пациентов, наблюдающихся у данного диетолога

Добавлять:

* Диетолог выбирает действие «Добавить»
* Система генерирует окно со списком всех пациентов

Добавлять нового пациента:

* Диетолог выбирает действие «Добавить»
* Система генерирует окно, где запрашивает все необходимые данные
* Диетолог вводить информацию
* Система сохраняет информацию

Добавить существующего пациента:

* Диетолог выбирает необходимого пациента и выбирает действие «Подробнее»
* Система генерирует окно с информацией о пациенте
* Диетолог проверяет информацию
* Диетолог выбирает действие «Добавить»

Расширение:

* Вести информацию об анализах пациента

Данное расширение начинает выполняться, когда диетологу необходимо ввести значения показателей пациента.

* Диетолог выбирает необходимого пациента;

Система генерирует окно с картой пациента и запрашивает действие:

* Анализы
* Диагнозы
* Сохранить
* Диетолог выбирает действие «Анализы»
* Система генерирует список анализов пациента

Добавлять:

* Диетолог выбирает действие «Добавить»
* Система генерирует окно для добавления анализа
* Диетолог вводит необходимые данные
* Система проверяет введенные данные
* Система сохраняет введенные значения.

Изменять:

* Диетолог выбирает действие «Подробнее»
* Система генерирует окно для добавления анализа
* Диетолог изменяет необходимые данные
* Система проверяет введенные данные
* Система сохраняет введенные значения.

Расширение:

* Вести информацию о диагнозах пациента

Данное расширение начинает выполняться, когда диетологу необходимо ввести значения показателей пациента.

* Диетолог выбирает необходимого пациента;

Система генерирует окно с картой пациента и запрашивает действие:

* Анализы
* Диагнозы
* Сохранить
* Диетолог выбирает действие «Диагнозы»
* Система генерирует список диагнозов
* Диетолог выбирает необходимые диагнозы
* Система сохраняет выбранные диагнозы.

Альтернативные потоки:

Изменять:

* Введены данные неверного формата. В этом случае система выдаст сообщение об ошибке. После чего система перейдет на 2 пункт подчиненного потока.

Добавлять:

* Введены данные неверного формата. В этом случае система выдаст сообщение об ошибке. После чего система перейдет на 2 пункт подчиненного потока.

Добавлять нового пациента:

* Введены данные неверного формата. В этом случае система выдаст сообщение об ошибке. После чего система перейдет на 2 пункт подчиненного потока.

Расширение:

Вести информацию об анализах:

* Введены данные неверного формата. В этом случае система выдаст сообщение об ошибке. После чего система перейдет на 2 пункт подчиненного потока.

Предусловие:

Диетологу необходимо войти в систему.

* Вариант использования «Вести меню пациента»

Данный вариант использования начинает выполняться, когда диетологу необходимо корректировать меню пациента.

* Система генерирует список всех пациентов, которые наблюдаются у данного диетолога и запрашивает действие:
* Карта
* Анализы
* Питание
* Диетолог выбирает необходимого пациента;
* Диетолог выбирает действие «Питание»
* Система генерирует список блюд, а также блюда, подобранные пациенту исходя из его данных.
* Диетолог выбирает либо удаляет блюда из подобранного меню и подтверждает свое действие.
* Система проверяет данные
* Система сохраняет введенные значения.

Альтернативные потоки:

* Не удается связаться с базой данных

Предусловие:

Диетологу необходимо войти в систему.

* Вариант использования "Вести дневник питания"

Краткое описание:

Данный вариант использования описывает ведение дневника питания

Основной поток событий:

* Система генерирует новое окно с выбором раздела:
* Дневник питания
* Моя карта пациента
* Пользователь выбирает раздел «Дневник питания»
* Система генерирует новое окно со днями и приемами пищи.

Изменять:

* Пациент выбирает необходимый прием пищи и блюдо.
* Пациент выбирает действие «Заменить»
* Система генерирует окно со списком доступных пациенту блюд
* Пациент выбирает блюдо из списка.
* Пациент подтверждает свое действие.
* Система сохраняет прием пищи.

Альтернативные потоки:

У пациента нет активного курса диеты. В данном случае система вернется к 1 пункту основного потока.

Предусловия:

Пациенту необходимо войти в систему.

* Вариант использования "Смотреть карту пациента"

Краткое описание:

Данный вариант использования описывает просмотр карты пациента

Основной поток событий:

* Система генерирует новое окно с выбором раздела:
* Дневник питания
* Моя карта пациента
* Пользователь выбирает раздел «Моя карта пациента»
* Система генерирует новое окно с картой пациента.

Альтернативные потоки:

У пациента нет активного курса диеты. В данном случае система вернется к 1 пункту основного потока.

Предусловия:

Пациенту необходимо войти в систему.

Исходя из вышеперечисленных вариантов использования строится следующая диаграмма вариантов использования: (Рисунок 2)

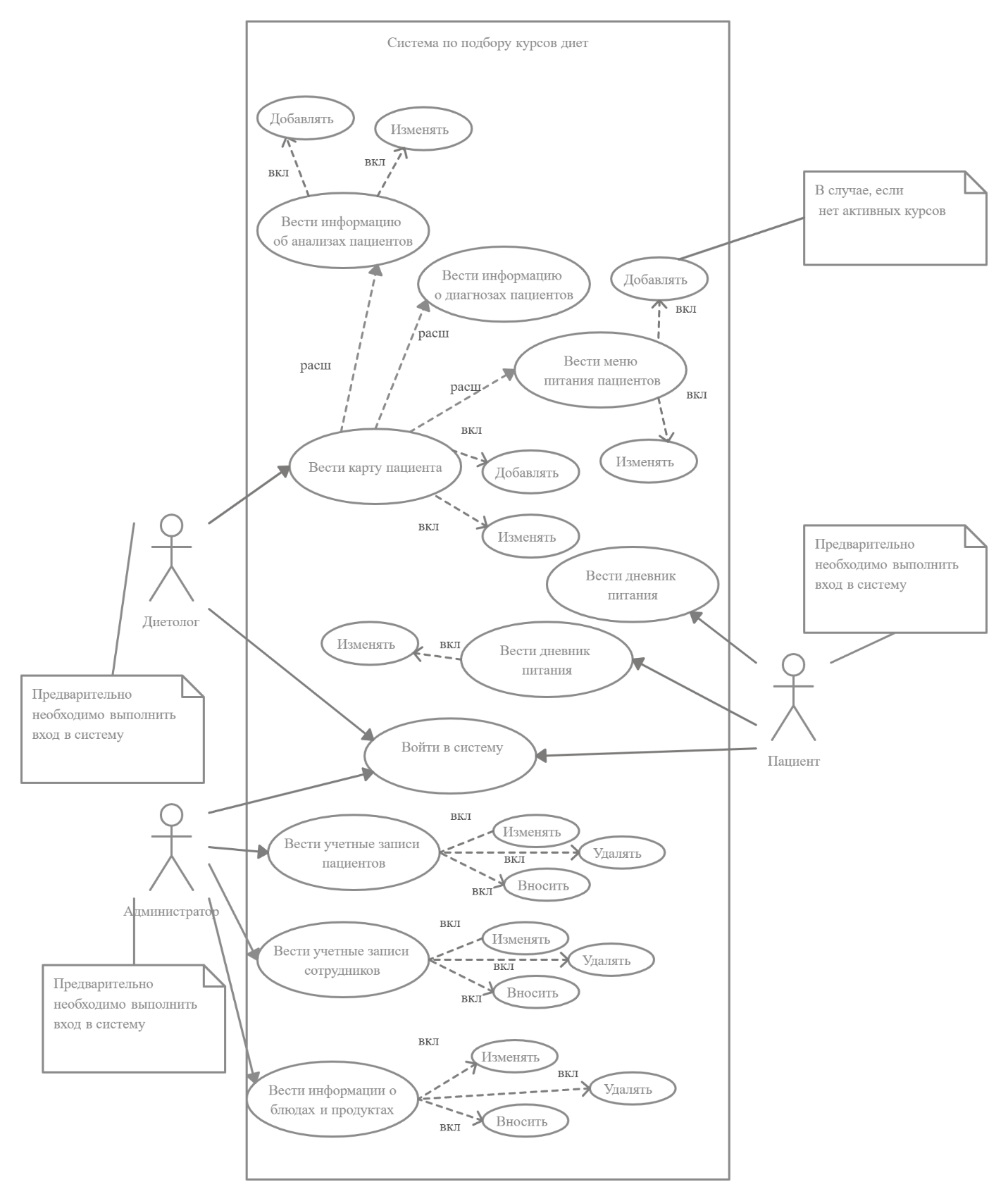


Рисунок 2 Диаграмма вариантов использования

1. ER-модель базы данных системы

Правильно составленная ER-модель способствует корректной работе разрабатываемой системы. Ниже представлена ER-модель системы «Диета» (Рисунок 3)

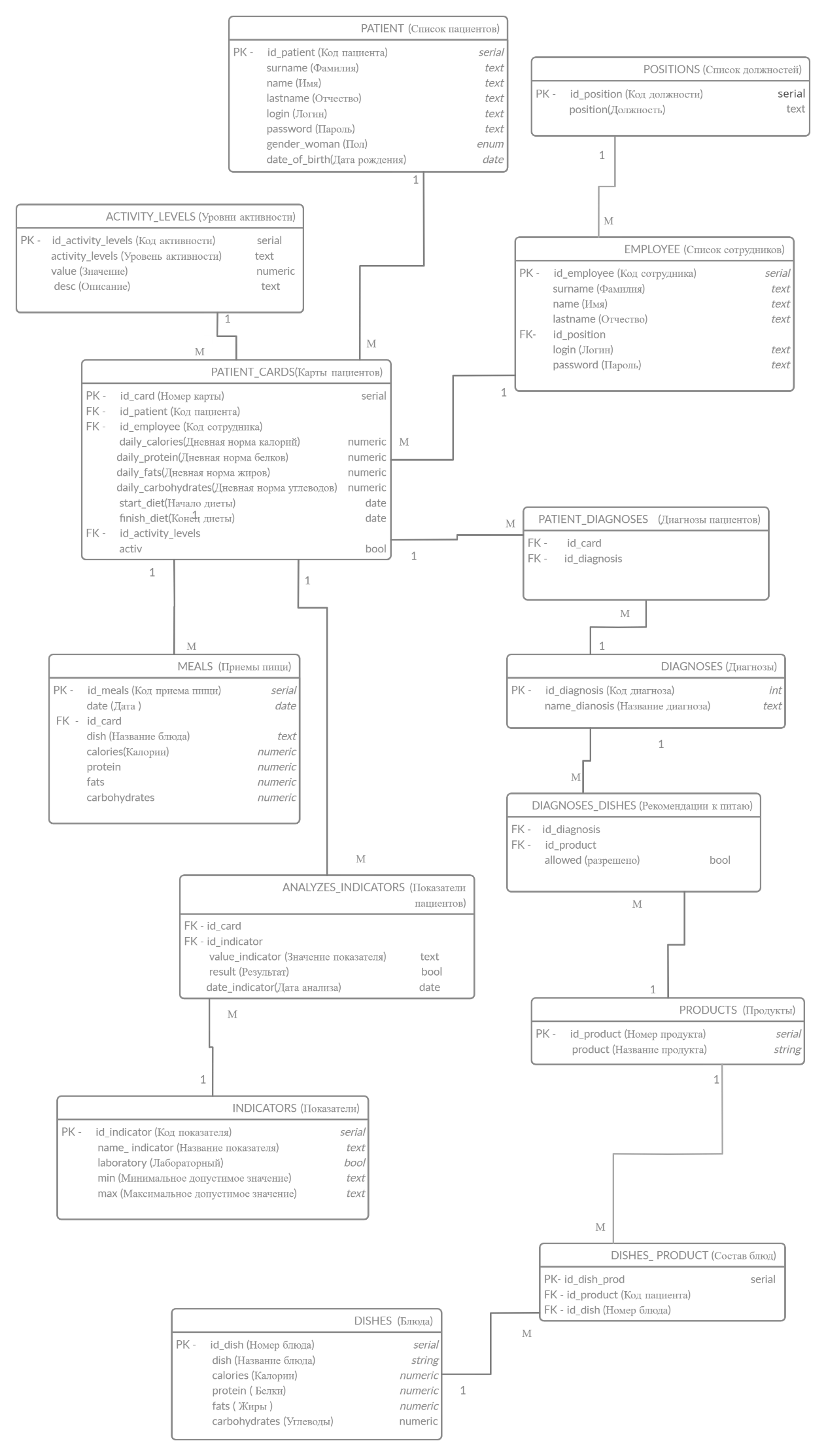


Рисунок 3 ER-модель базы данных веб-приложения "Диета"

Описание сущностей ER-модели:

* Должности - содержит информацию о должностях сотрудников;
* Сотрудники - содержит информацию о всех сотрудниках;
* Пациенты - содержит информацию о всех пациентах;
* Уровни активности - содержит информацию об уровнях активности;
* Карта пациента – содержит информацию, которая необходима для подбора рациона питания;
* Приемы пищи – содержит информацию о приемах пищи;
* Анализы пациентов – содержит информацию показателей пациентов;
* Показатели – список всех показателей
* Диагнозы пациентов – содержит диагнозы пациентов
* Диагнозы-список всех диагнозов
* Рекомендации к питанию – содержит список разрешенных и запрещенных блюд при определенном диагнозе
* Продукты – список всех продуктов
* Состав блюд – содержит состав всех блюд
* Блюда – содержит список всех блюд

Данная ER- модель является нормализованной и находится в третьей нормальной форме.

1. Выбор инструментария для выполнения работы

Для разработки веб-приложения используются технологии: HTML, CSS, ASP.Net Core MVC, ADO.NET Entity Framework, Bootstrap.

HTML – стандартный язык разметки документов. Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования.

Все документы, содержащие разметку на языке HTML, имеют разрешением .html или .htm.

Для просмотра этих документов используются специальные приложения – браузеры.

CSS – формальный язык описания внешнего вида документа, написанного с помощью языка разметки. Как правило используется в виде средства описания, оформления внешнего вида Web-страниц, написанных с помощью языков разметки HTML.

CSS используется создателями Web-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления

внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы от описания внешнего вида этой веб-страницы.

Bootstrap – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления, что облегчает написание разметки страниц.

ASP.NET CORE MVC Framework – фреймворк для создания кросс-платформенных веб-приложений, который реализует шаблон Model-view-controller.

Платформа ASP.NET CORE MVC базируется на взаимодействии трех компонентов (Рисунок 4): контроллера, модели и представления. Контроллер принимает запросы, обрабатывает пользовательский ввод, взаимодействует с моделью и представлением и возвращает пользователю результат обработки запроса.

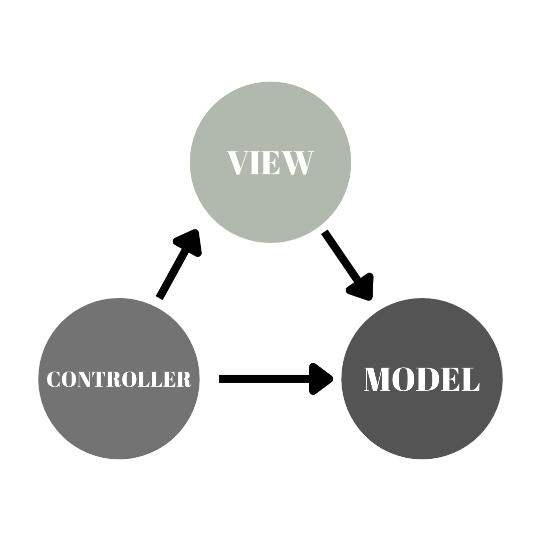


Рисунок 4 Схема модели MVC

Модель представляет слой, описывающий логику организации данных в приложении. Представление получает данные из контроллера и генерирует элементы пользовательского интерфейса для отображения информации.

ADO.NET Entity Framework(EF) – объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для .NET Framework от Microsoft. Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL.

Bootstrap — свободный набор инструментов для создания сайтов и веб-приложений.

1. Внешний вид разрабатываемой системы.

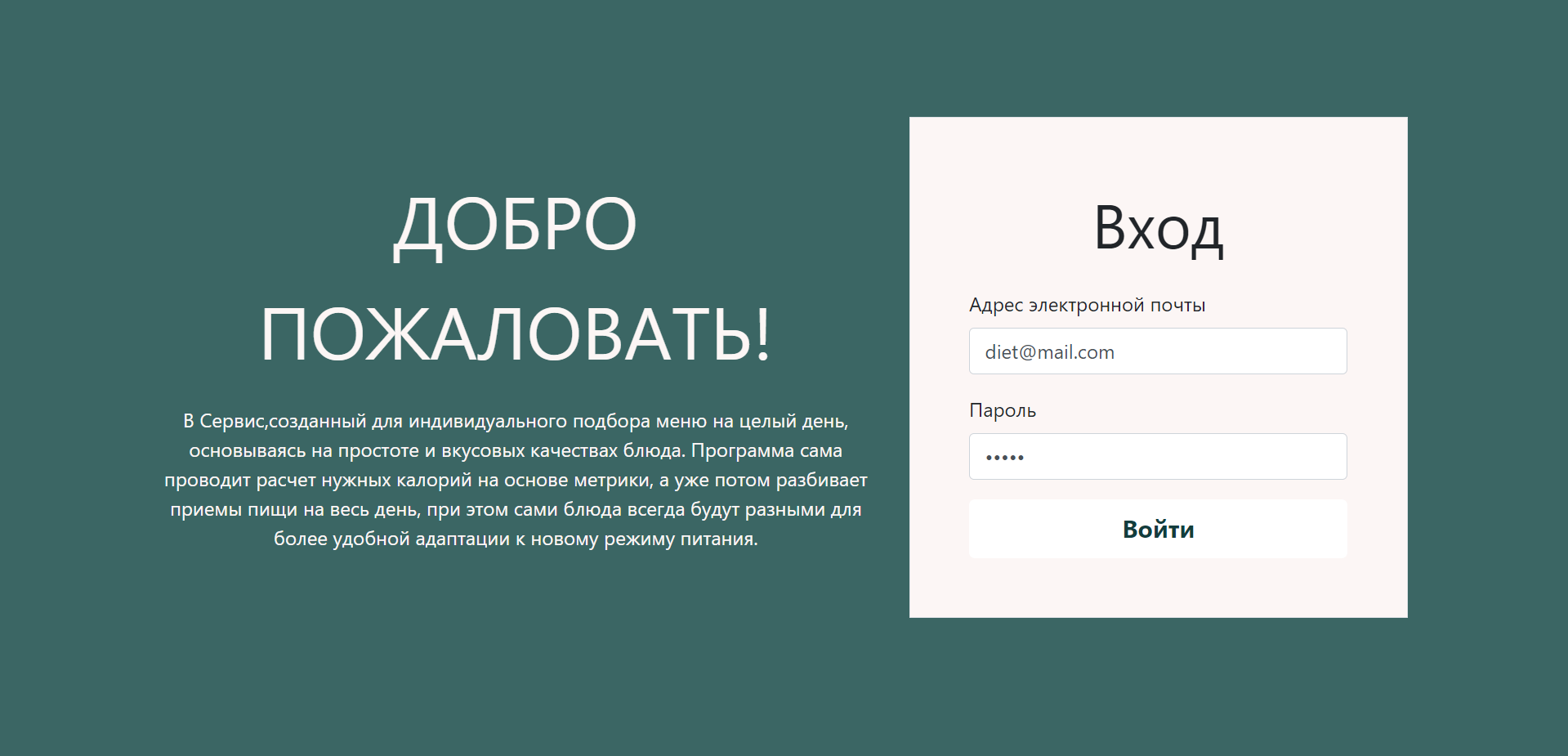
При запуске веб-приложения генерируется стартовое окно (Рисунок 5). 

Рисунок 5 Стартовое окно приложения

Благодаря языку разметки CSS интерфейс сайта является адаптивным (Рисунок 6).

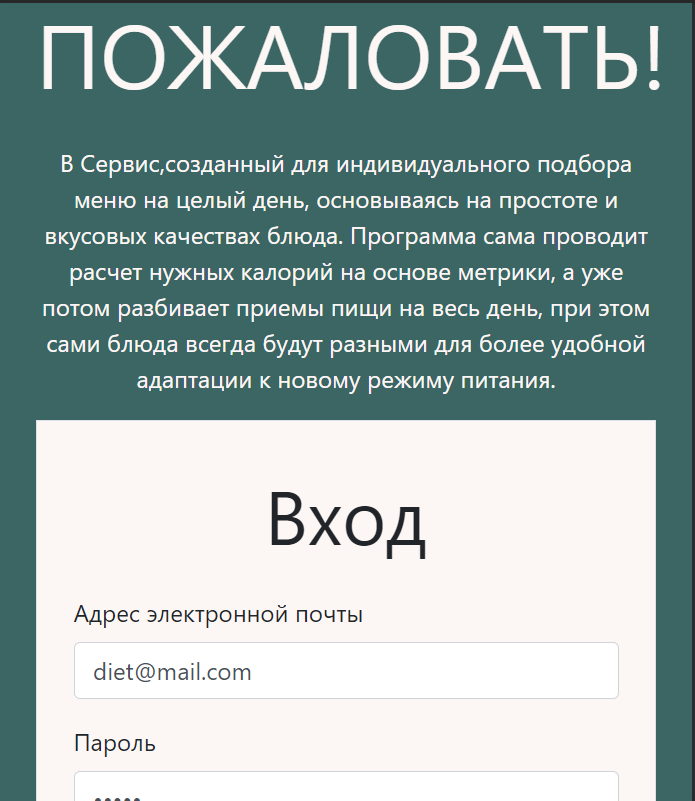


Рисунок 6 Стартовое окно приложения на мобильном устройстве

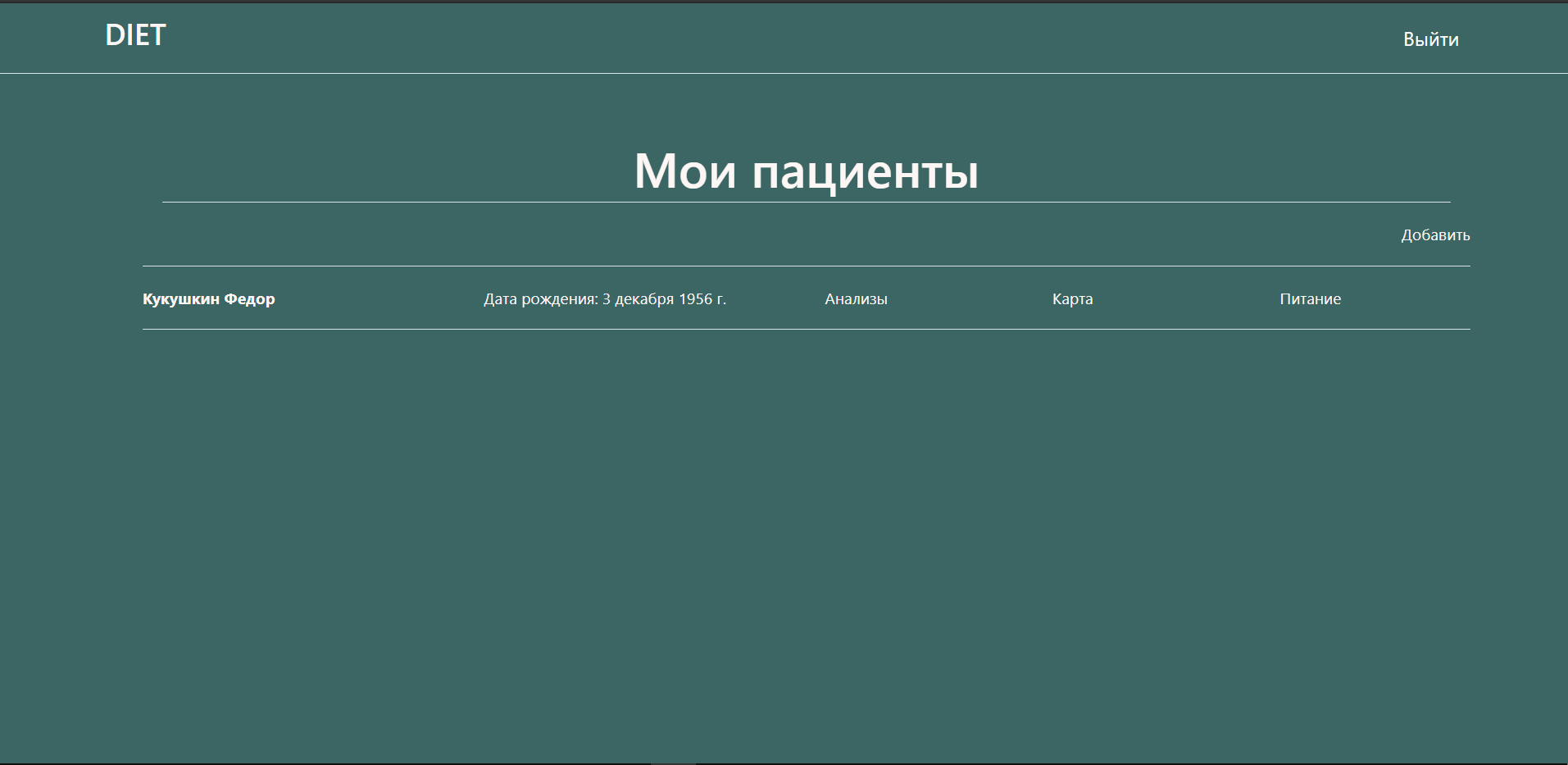
В случае входа диетолога открывается следующее окно со списком его пациентов (Рисунок 7). 

Рисунок 7 Окно со списком пациентов диетолога

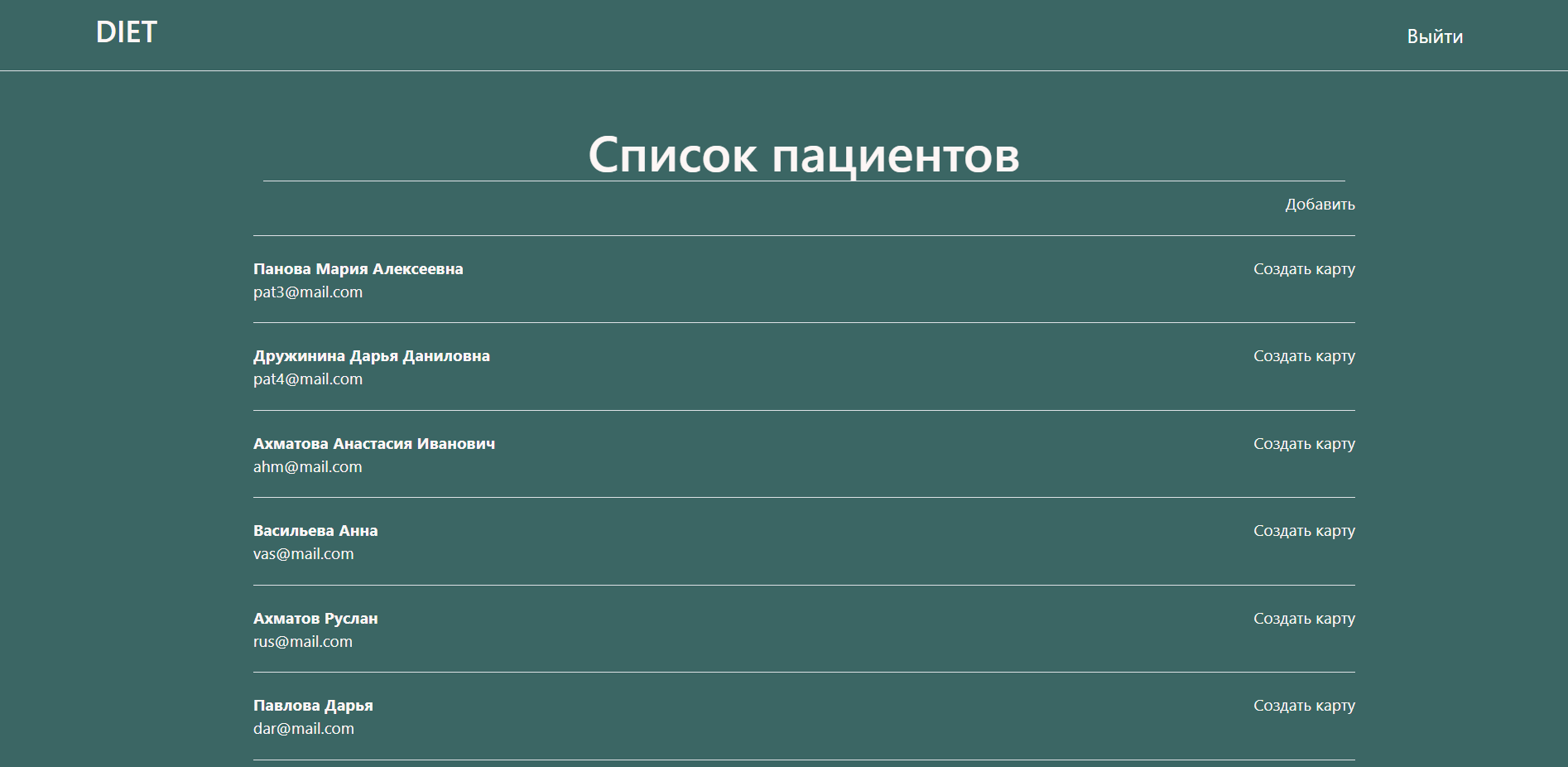
Окно со списком всех пациентов системы выглядит следующим образом (Рисунок 8). 

Рисунок 8 Список всех пациентов системы

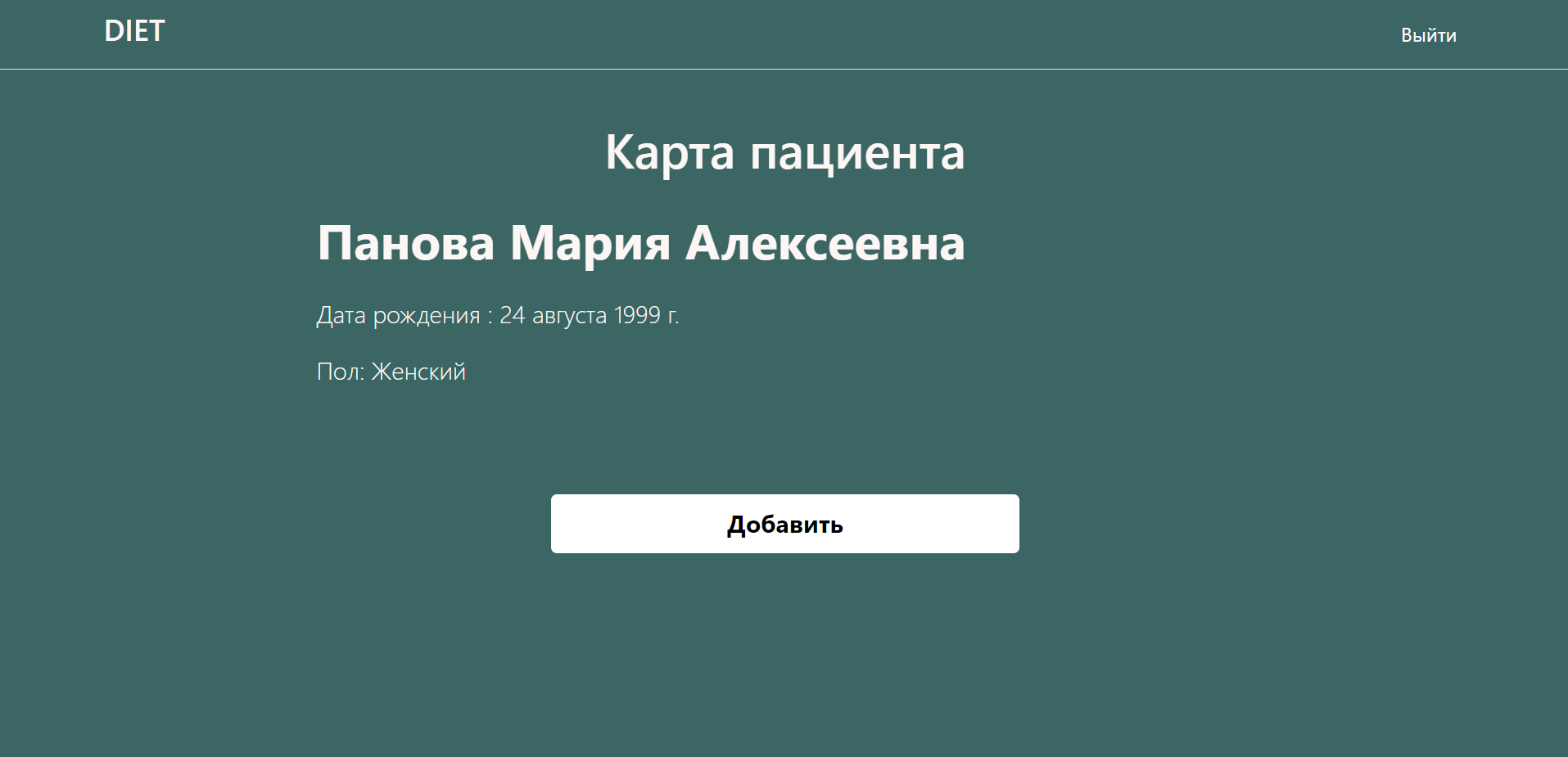
При создании новой карты пациента генерируется следующее окно (Рисунок 9). 

Рисунок 9 Окно для создания карты пациента

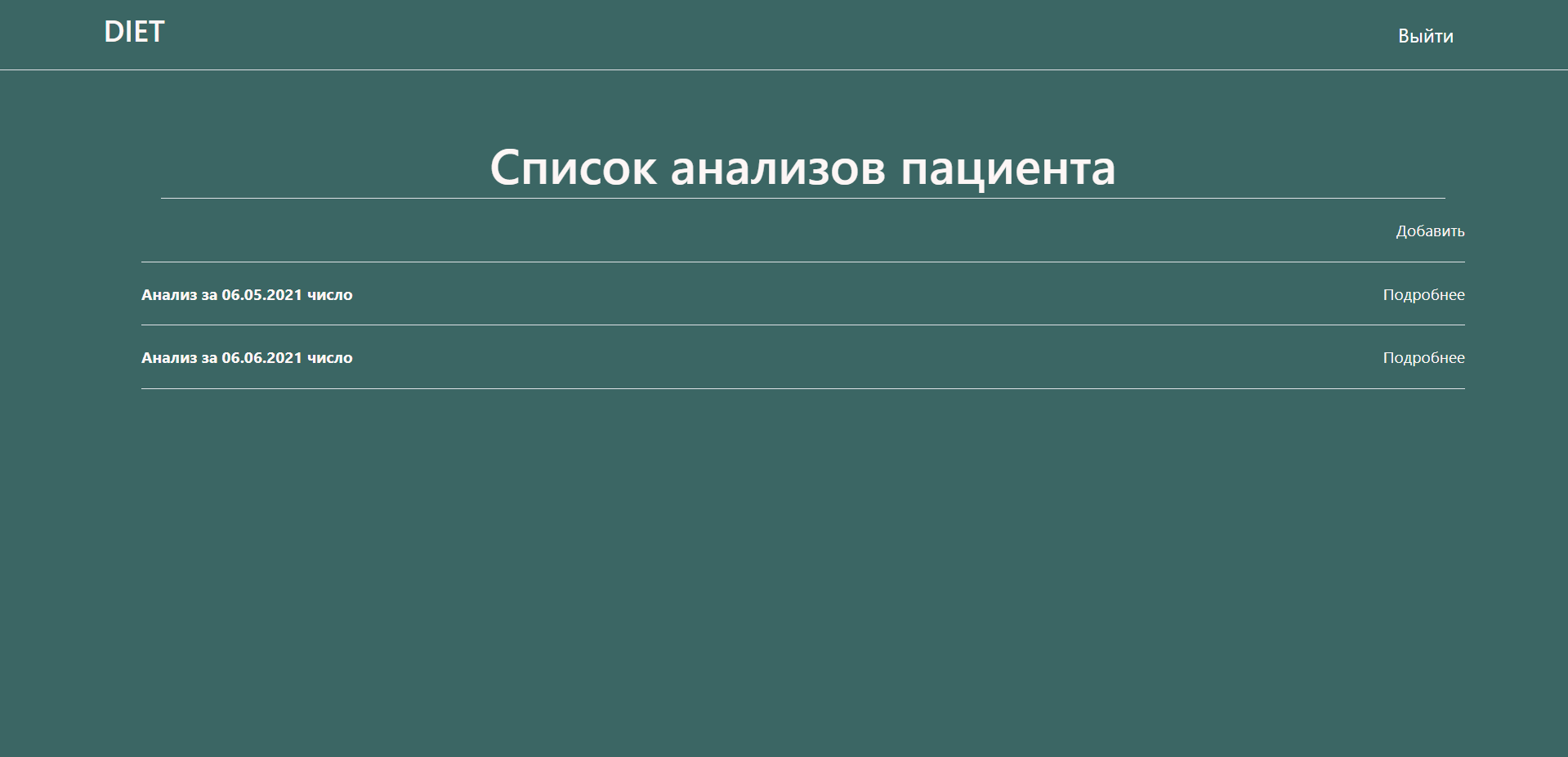
Окно со списком анализов определенного пациента выглядит следующим образом: (Рисунок 10).

Рисунок 10 Окно со списком анализов пациента

При добавлении или просмотре одного из анализов, система генерирует следующее окно (Рисунок 11).

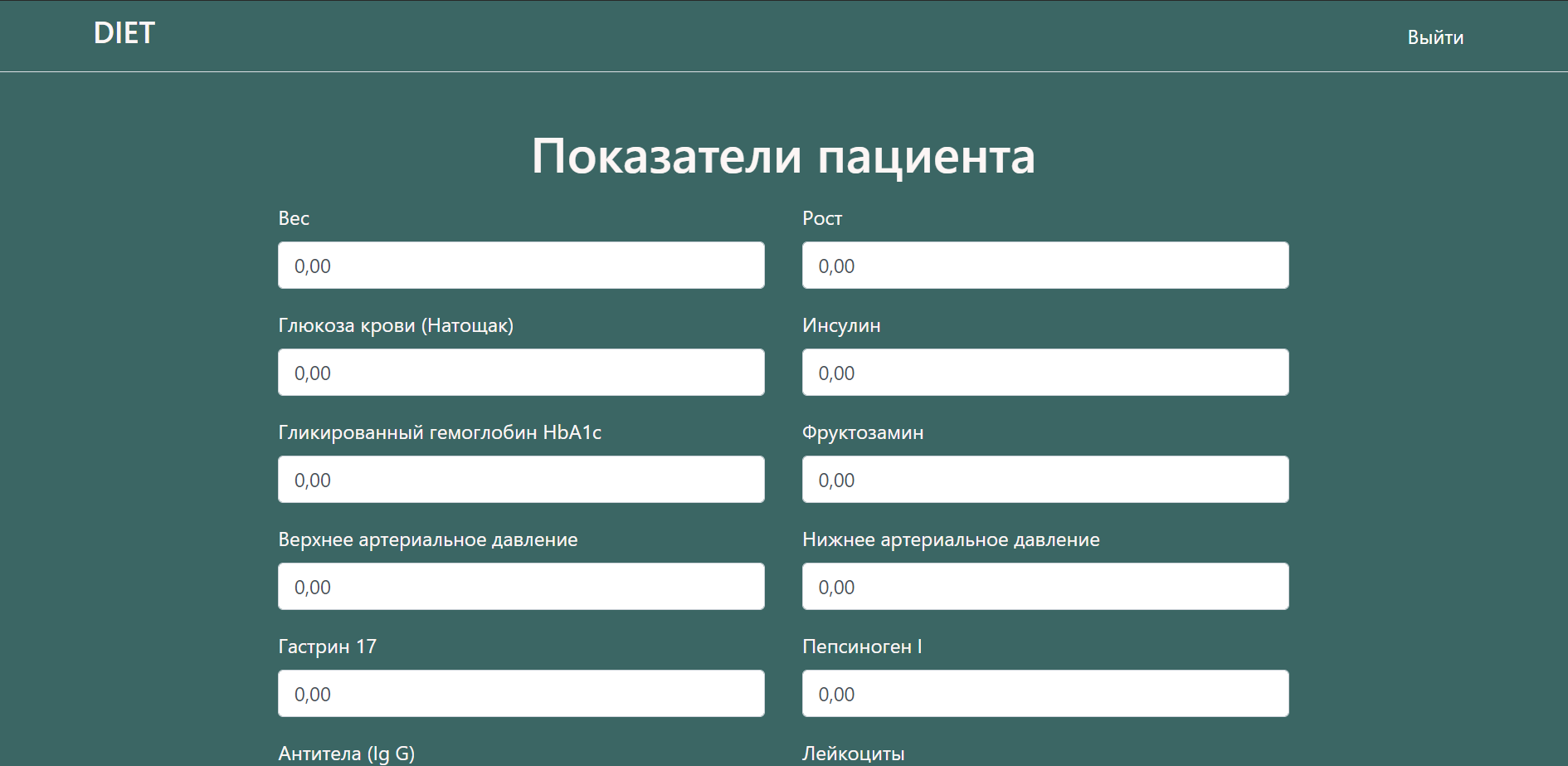


Рисунок 11 Окно с показателями пациента

При просмотре карты пациента открывается следующая страница (Рисунок 12).

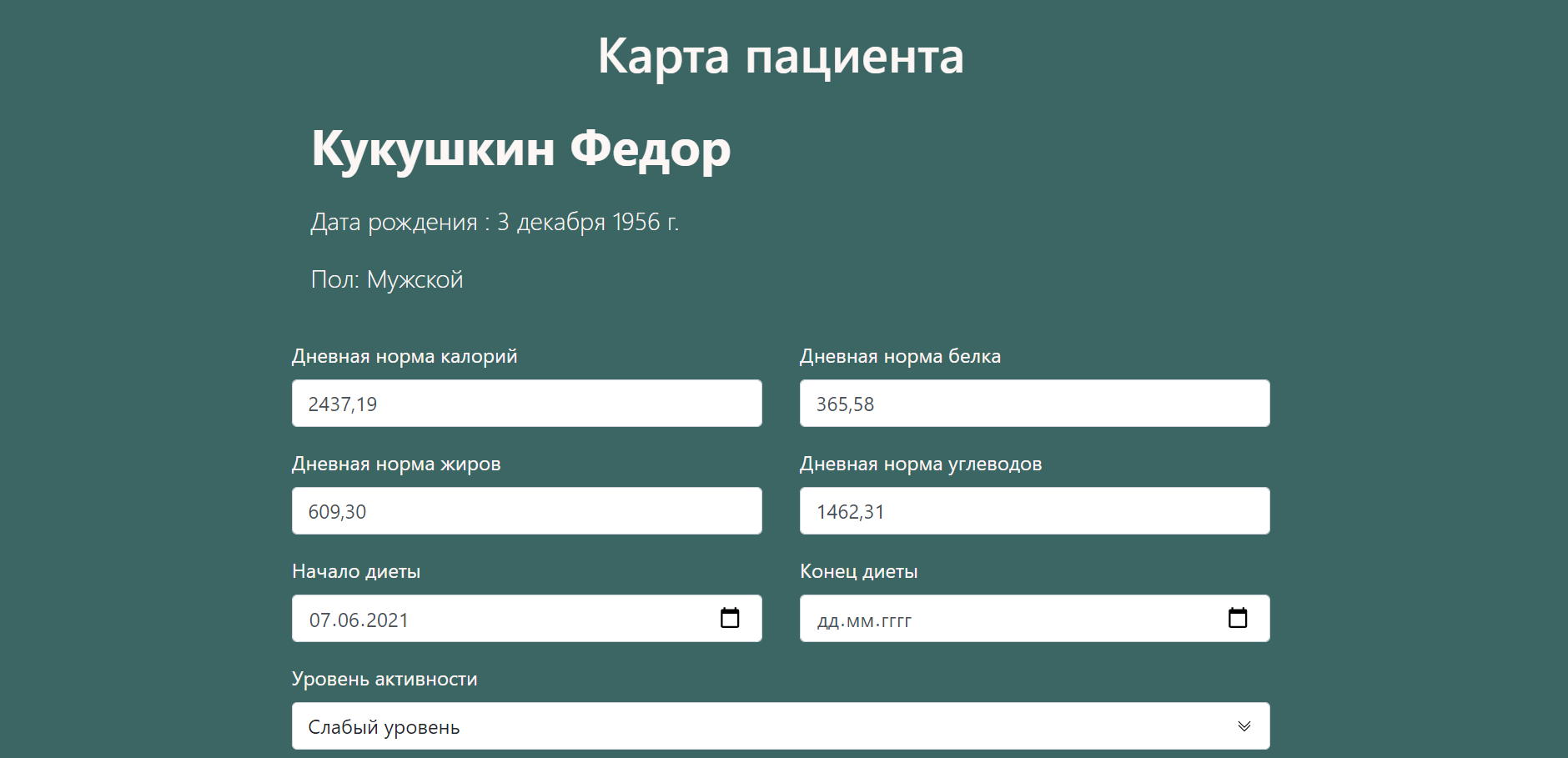


Рисунок 12 Окно с картой пациента

В том случае, если диетологу необходимо добавить диагноз пациенту открывается следующее окно: (Рисунок 13).

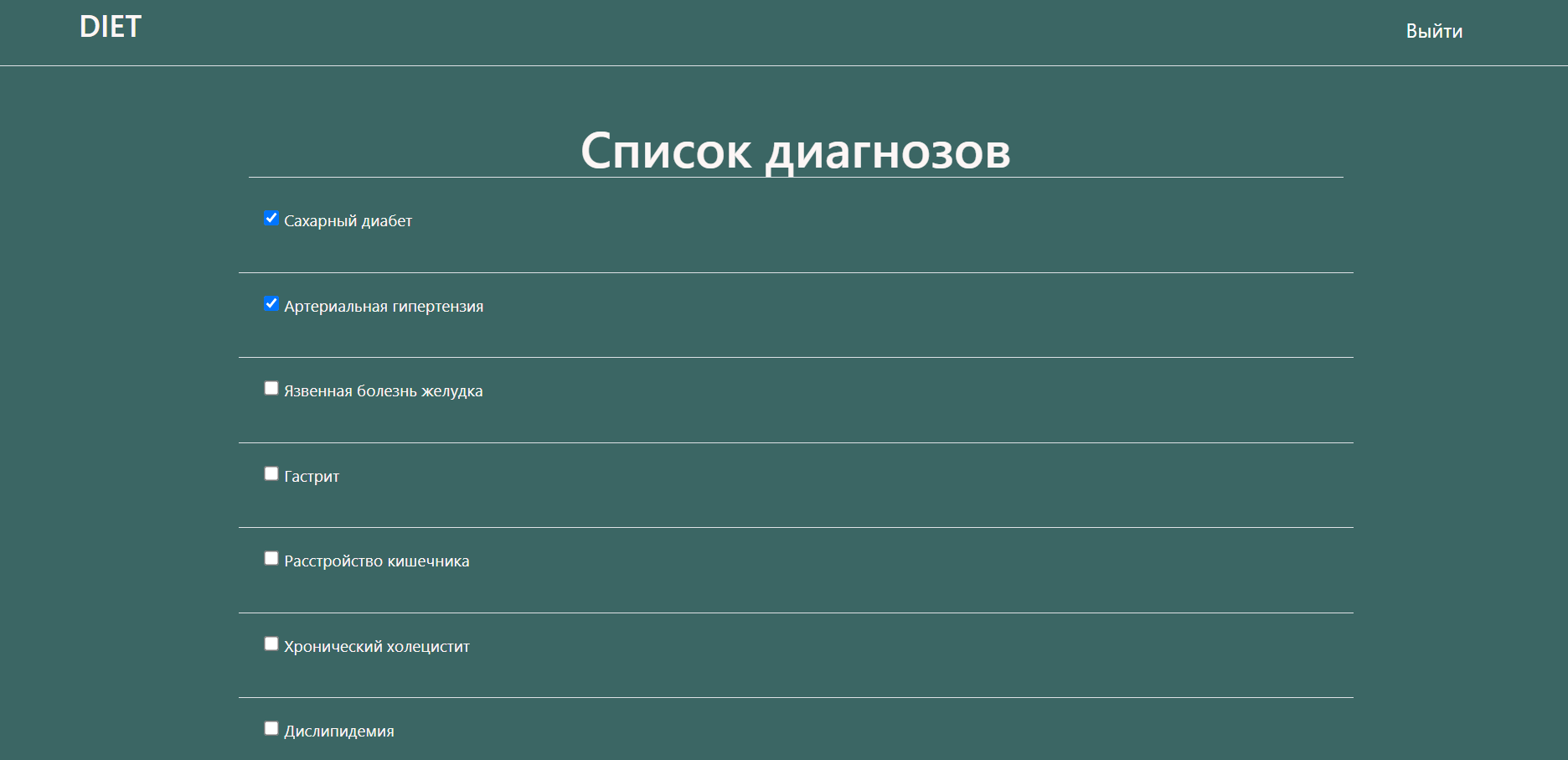


Рисунок 13 Окно со списком диагнозов пациента

При входе администратора в систему система открывает следующее окно: (Рисунок 14).

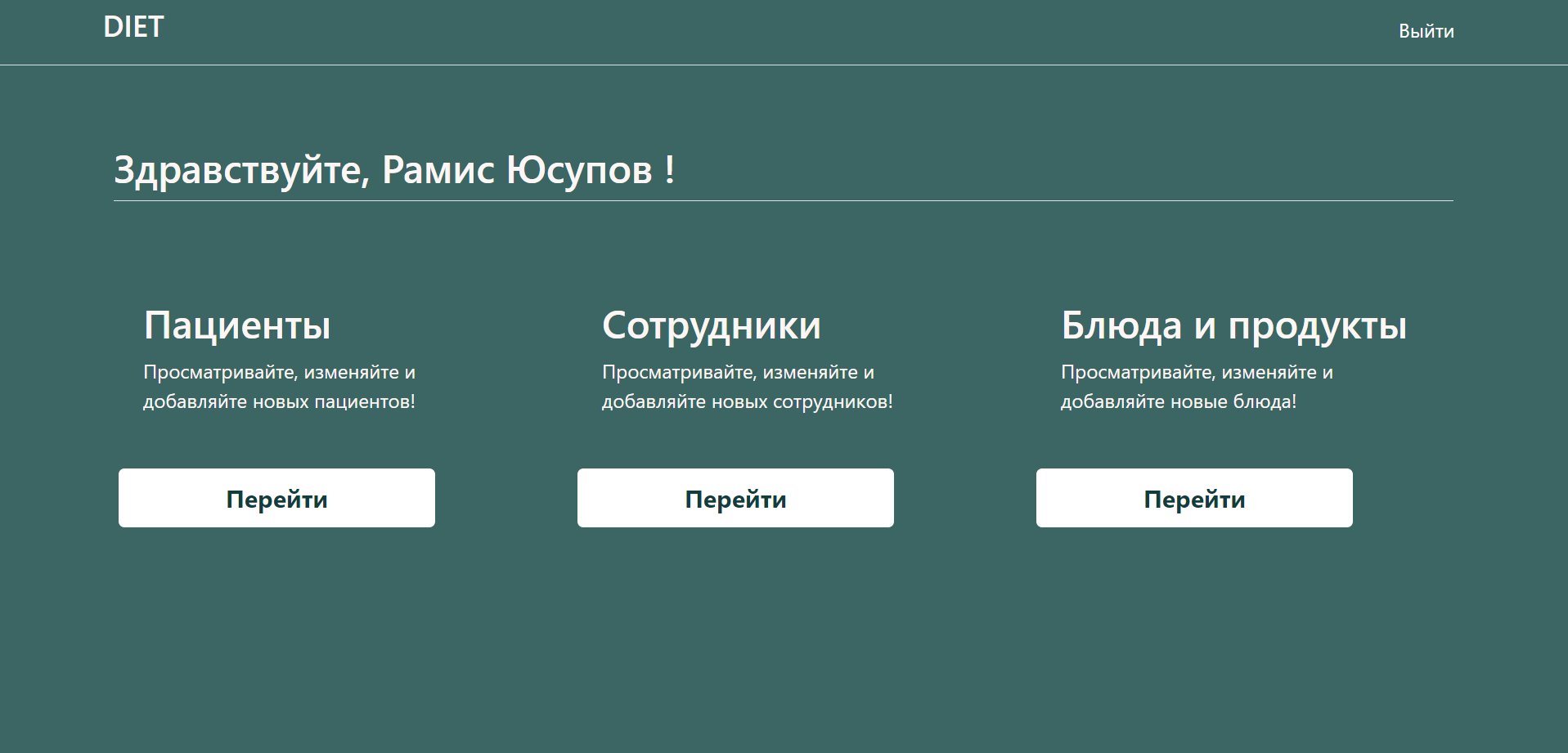


Рисунок 14 Стартовое окно администратора

Раздел «Пациенты» выглядит следующим образом: (Рисунок 15).

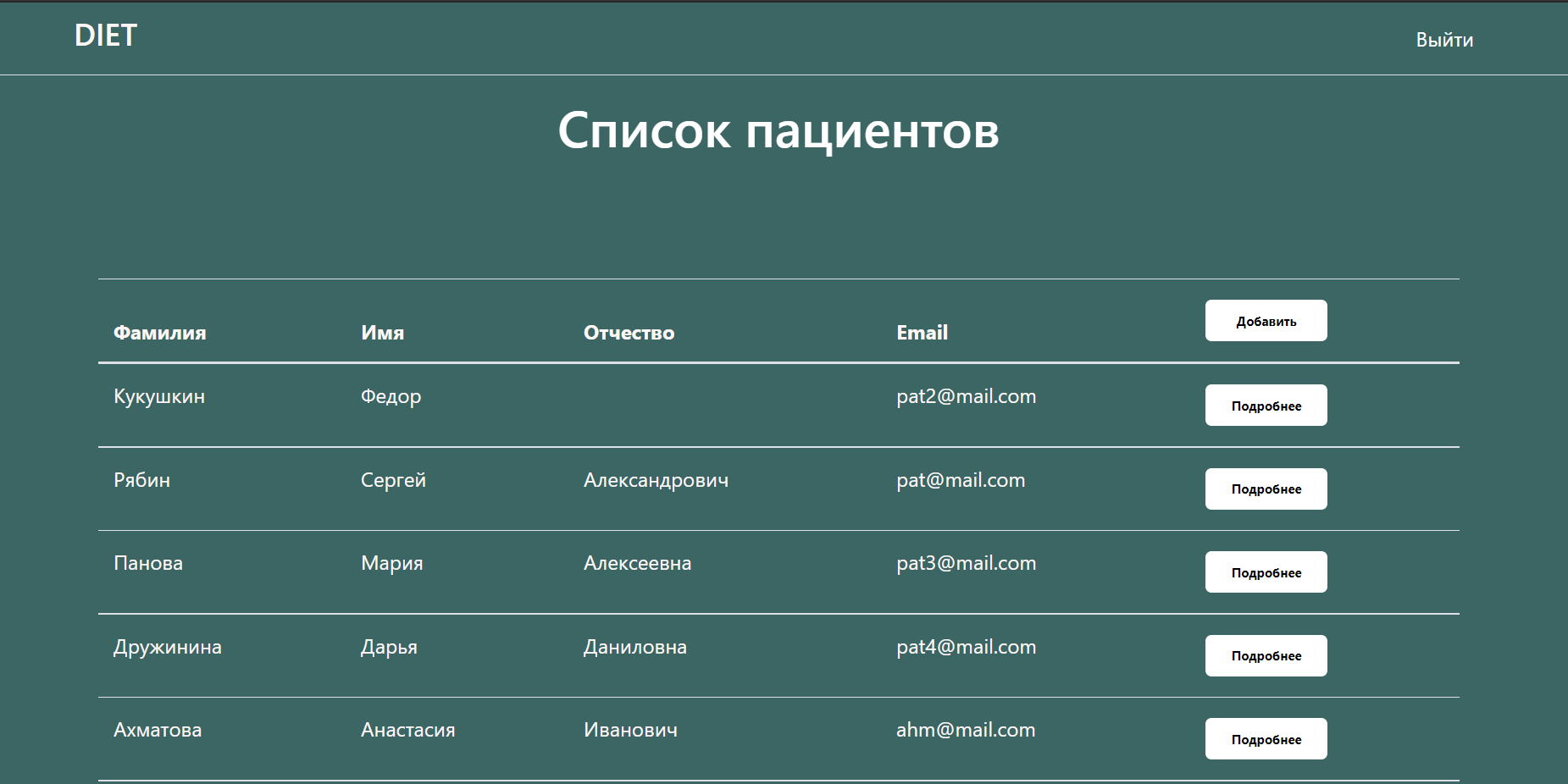


Рисунок 15 Окно со списком аккаунтов пациентов

При создании аккаунта пациента генерируется следующее окно: (Рисунок 16).

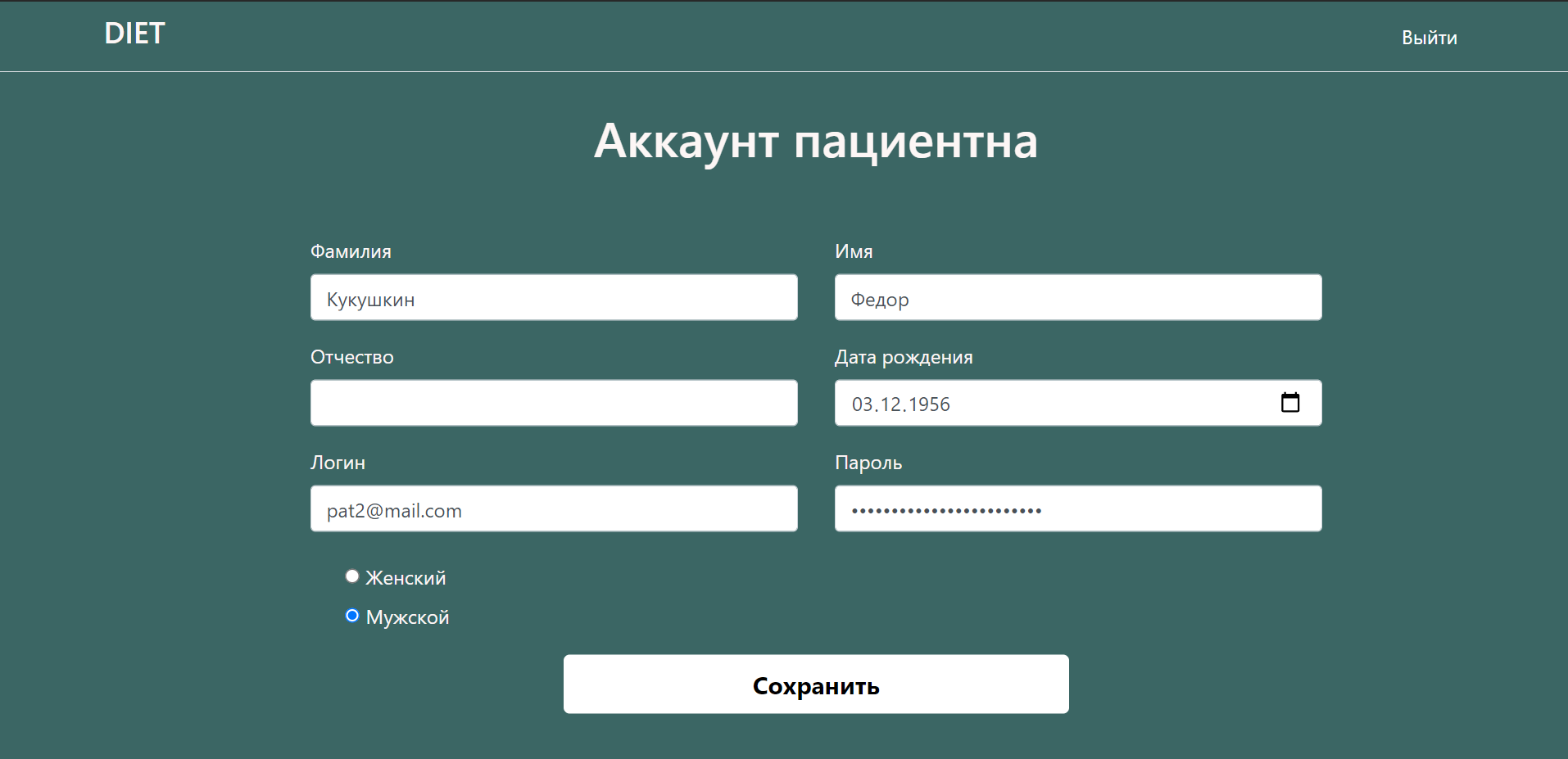


Рисунок 16 Окно со списком пользователей с ролью "Пациент"

Аналогичным образом выглядит раздел со списком сотрудников.

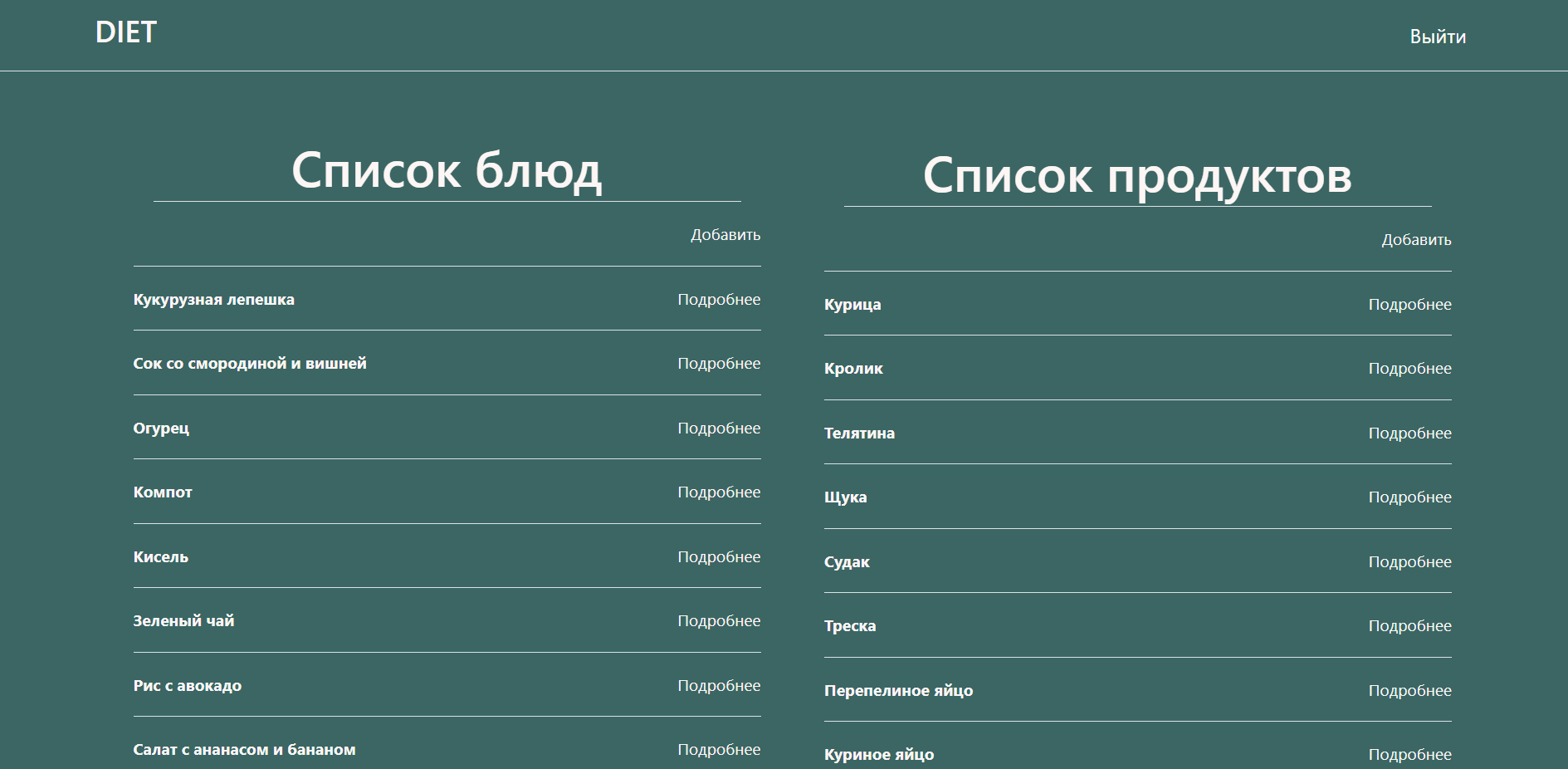
Раздел «Блюда» выглядит следующим образом: (Рисунок 17).

Рисунок 17 Окно со списком всех блюд и продуктов

При нажатии «Подробнее» система генерирует окно с информацией о выбранном блюде или продукте: (Рисунок 18).

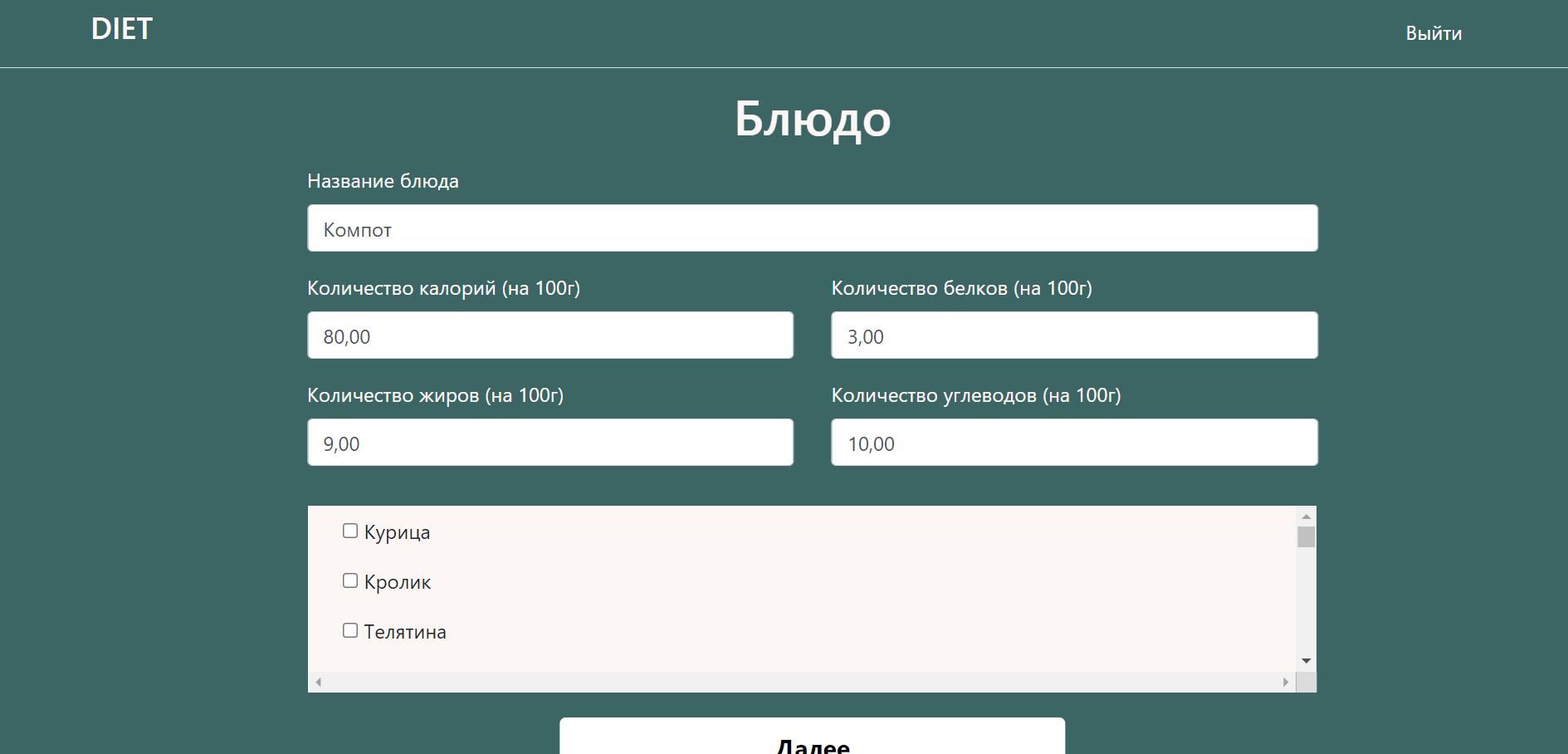


Рисунок 18 Окно с информацией о блюде

# ГЛАВА 2. ПРОЕКТНАЯ ЧАСТЬ

1. Описание содержания проекта

Программа разработана с использованием кросс-платформенного фреймворка ASP .NET CORE. Схемой разделения данных приложения, пользовательского интерфейса и управляющей логики выступает модель MVC. Взаимодействие с базой данных осуществляется с помощью технологии ADO.NET Entity Framework .

База данных хранится в СУБД Postgres. Также в проект были добавлены дополнительные пакеты: (Рисунок 19).

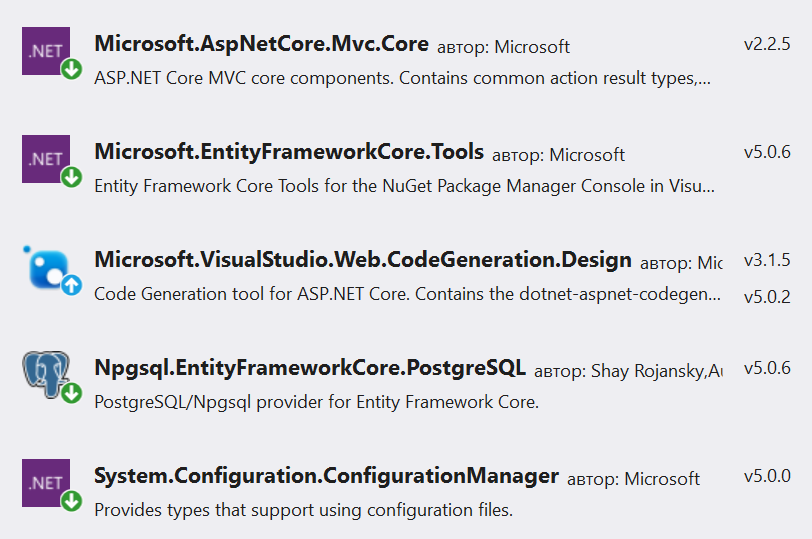


Рисунок 19 Дополнительные пакеты проекта

Далее, для удобной работы со стилями в HTML-страница, был подключен фреймворк Bootstrap. Подключался он путем добавления в проект следующих файлов: (Рисунок 20).

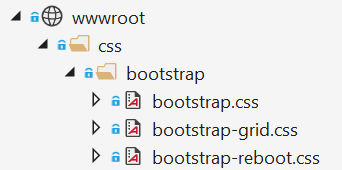


Рисунок 20 Файлы фреймворка Bootstrap

После подключения к базе данных были сформированы следующие модели: (Рисунок 21).

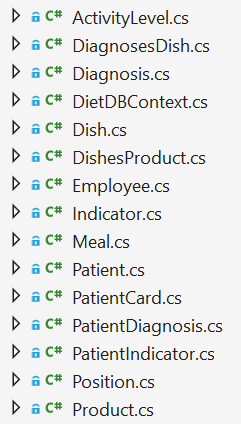


Рисунок 21 Модели проекта

Структура проекта выглядит следующим образом: (Рисунок 22).

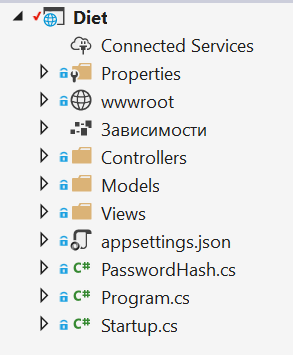


Рисунок 22 Содержание проекта

В папке «Controllers» хранятся контроллеры каждой из выделенных выше действующих лиц системы.

В папке «Models» хранятся модели, в которых осуществляются все необходимые запросы к базе данных и так далее.

В папке «Views» хранятся представления, которые содержат внешний вид страниц.

1. Описание программной части системы

Ниже представлено описание программной части системы:

При запуске веб-сервиса вызывается метод Login версии HttpGet контроллера Account (Рисунок 23), отвечающий за вход в систему. 

Рисунок 23 Конроллер для входа в систему

Метод в свою очередь вызывает одноименное представление (Рисунок 24), в котором описан внешний вид страницы



Рисунок 24 Представление входа в систему

Данные с вводимых полей передаются с помощью asp Tag-хелперов (Рисунок 25). Tag-хелперы используются в представлениях и выглядят как обычные html-элементы или атрибуты, однако при работе приложения они обрабатываются движком Razor на стороне сервера и в конечном счете преобразуются в стандартные html-элементы. Подключаются tag-хелперы следующим образом:

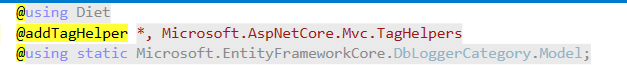


Рисунок 25 Подключение ASP Tag-хелперов

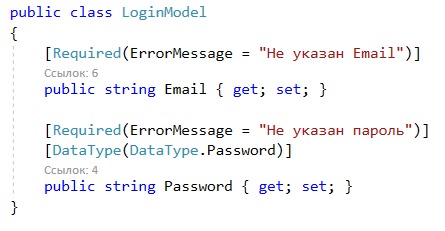
После того, как пользователь нажмет кнопку «Войти» представление передаст модель данных (Рисунок 26) в одноименный метод контроллера Account, только в HttpPost версию, где происходит аутентификация и авторизация пользователя. В противном случае система оповестит пользователя о неправильных пароле или логине (Рисунок 26). 

Рисунок 26 Модель входа в систему

Далее, исходя из того, к какой роли относится вошедший пользователь, идет вызов контроллера, отвечающего за функционирование сайта для данной роли.

В том случае, если авторизовался администратор, система вызовет контроллер AdminHome метод MenuAdmin (Рисунок 27), который выглядит следующим образом:

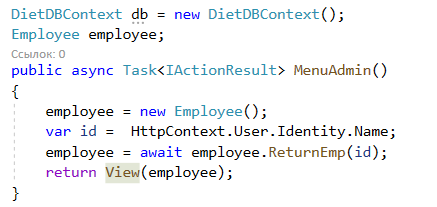


Рисунок 27 Контроллер Администратора

Посредством модели данный (Рисунок

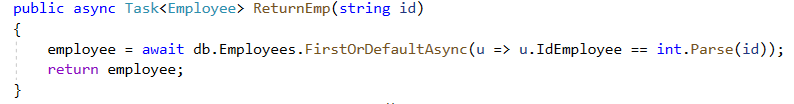
28) метод собирает всю необходимую информацию и отправляет ее в представление. 

Рисунок 28 Метод поиска сотрудника

Валидация введенных данных происходит посредством регулярных выражения, прописанных над каждым полем модели (Рисунок 29). 

Рисунок 29 Валидация полей модели "Сотрудник"

ЗАКЛЮЧЕНИЕ

В результате выполнения работы, было разработано веб-приложение по подбору индивидуального курса диеты.

Приложение помогает пользователям автоматизировать организацию рациона питания.

Разработанное приложение написано на языке C# с помощью фреймворка ASP .NET CORE MVC, что делает его кросс-платформенным.

В первой главе были рассмотрены: аналог разрабатываемого сервиса, поставлена цель ВКР, выделены функциональные и нефункциональные требования, составлена ER-модель базы данных, выявлены действующие лица системы и их варианты использования, созданы макеты приложения.

Во второй главе были реализованы все поставленные в первой главе цели, задачи и установлены все необходимые расширения.

Все поставленные цели проекты достигнуты.