1. [**Разработка технического задания**](#_30j0zll)2
   1. [Исходная постановка задачи проектирования](#_1fob9te) 2
   2. [Анализ условий предприятия](#_3znysh7) 2
      1. [Цели функционирования предприятия](#_2et92p0) 3
      2. [Организационная структура предприятия](#_tyjcwt) 4
      3. [Состав бизнес-процессов](#_y0pdkaa1a84s) 5
      4. [Содержание бизнес-процессов](#_1t3h5sf) 6
      5. [Классы-объекты предметной области](#_4d34og8) 7
      6. [Автоматизируемые элементы](#_2s8eyo1) 9
   3. [Формирование требований к системе](#_17dp8vu) 9
      1. [Состав требований](#_3rdcrjn) 9
      2. [Состав подсистем](#_26in1rg) 11
      3. [Варианты использования](#_lnxbz9) 14
      4. [Содержание сценариев](#_35nkun2) 15
2. [**Разработка рабочего проекта**](#_1ksv4uv)22
   1. [Определение классов анализа](#_44sinio) 22
   2. [Определение методов объекта](#_2jxsxqh) 25
   3. [Выбор технологии реализации](#_z337ya) 32
   4. [Проектирование хранилища данных](#_3j2qqm3) 33
3. [**Разработка программного кода**](#_ls90hda2m6rg)37
4. [**Развертывание**](#_2xcytpi)47
   1. [Интерфейс системы](#_1ci93xb) 47
5. [**Разработка сценариев тестирования**](#_1y810tw)54
6. [**Список литературы**](#_3whwml4)57

# 

# **Разработка технического задания**

## **Исходная постановка задачи проектирования**

Задачей проектирования является разработка системы мониторинга и исправления неисправностей оборудования Кемеровского областного медицинского колледжа (далее КОМК).

## **Анализ условий предприятия**

Полное официальное наименование: Кемеровский областной медицинский колледж.

Сокращенное наименование: КОМК.

Статус организации, форма юридического лица: государственное бюджетное профессиональное учреждение «Кемеровский областной медицинский колледж».

Общие принципы функционирования организации:

Деятельность КОМК проводится в соответствии со следующими общими принципами функционирования организаций:

* Скалярный принцип (scalaris - ступенчатый) - иерархичность построения организаций.
* Принцип функциональности - должностные обязанности на каждой ступени управления определяются настолько детально, насколько возможно.
* Принцип диапазона контроля – руководитель умственной деятельности имеет не более 8 человек.
* Принцип личной ответственности - ответственность руководителя за действия подчинённых ему людей в полном объёме.

Вид и профиль деятельности организации:

Основная деятельность организации включает:

1. Обучение абитуриентов по специальностям:
   * «Лечебное дело», профиль фельдшер;
   * «Акушерское дело», профиль акушерка;
   * «Сестринское дело», профиль мед. сестра/брат;
   * «Лабораторная диагностика», профиль медицинский лабораторный техник;
   * «Фармация», профиль фармацевт;
   * «Стоматология», профиль зубной техник.
2. Предоставление услуг по повышению квалификации младшего мед. персонала и людей смежных профессий (МЧС, Пожарные службы).

Физический адрес: 650064, Кемеровская область, г. Кемерово, пр. Советский, 58.

### **Цели функционирования предприятия**

Цели функционирования КОМК приведены на рисунке 1.

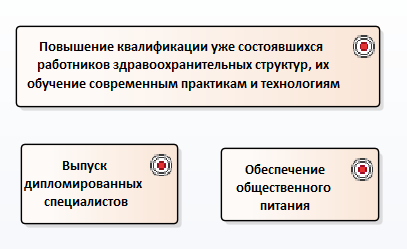


Рисунок 1. Цели функционирования КОМК

Описание целей функционирования колледжа:

* Повышение квалификации уже состоявшихся работников здравоохранительных структур, их обучение современным практикам и технологиям – обучение людей пришедших на повышение квалификации;
* Выпуск дипломированных специалистов – обучение студентов.

Для поддержки основных целей функционирования, выделим отдельную цель:

* Обеспечение бесперебойной работы колледжа - поддержание колледжа в работоспособном режиме.

### **Организационная структура предприятия**

Модель организационной структуры приведена на рисунке 2.

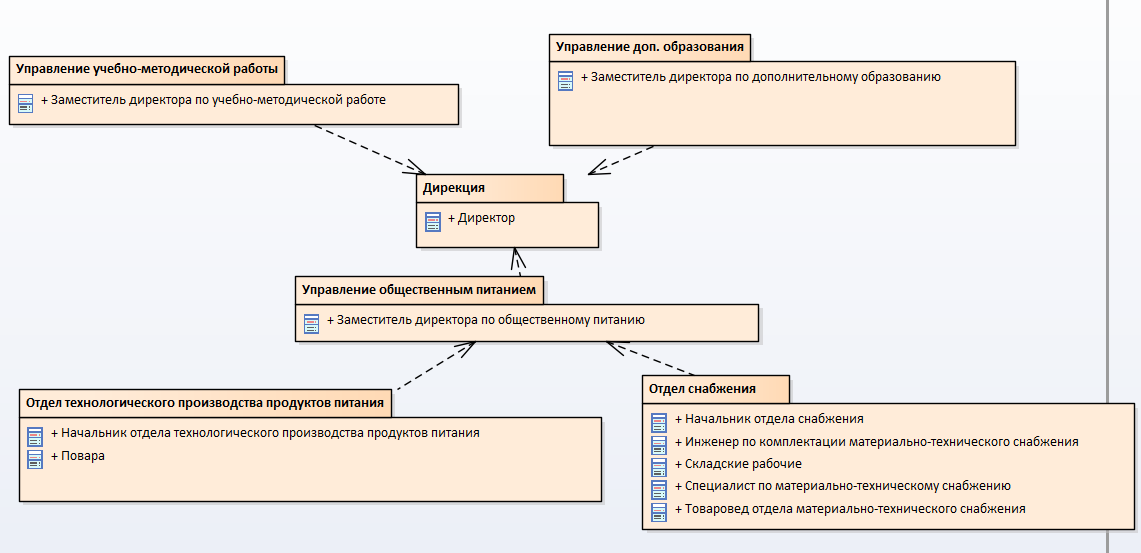


Рисунок 2. Организационная структура предприятия

Руководство деятельностью КОМК осуществляет: Дирекция. Блоку дирекция подчиняются:

* Управление учебно-методической работой – осуществление контроля учебного процесса.
* Управление доп. образованием – осуществляет контроль за дополнительным образованием;
* Управление общественным питанием - включает в себя: заместитель директора по общественному питанию; осуществляет координацию отделов:
  + Отдел технологического производства продуктов питания - состоит из: начальник отдела технологического производства продуктов питания; повара; осуществляет организацию питания колледжа и проверку его качества.
  + Отдел снабжения - состоит из: начальник отдела снабжения, складские рабочие, специалист по материально-техническому снабжению, инженер по комплектации материально-технического снабжения, товаровед отдела материально-технического снабжения; осуществляет контроль за оборотом товаров.

### **Состав бизнес-процессов**

Состав бизнес-процессов приведен на рисунке 3.

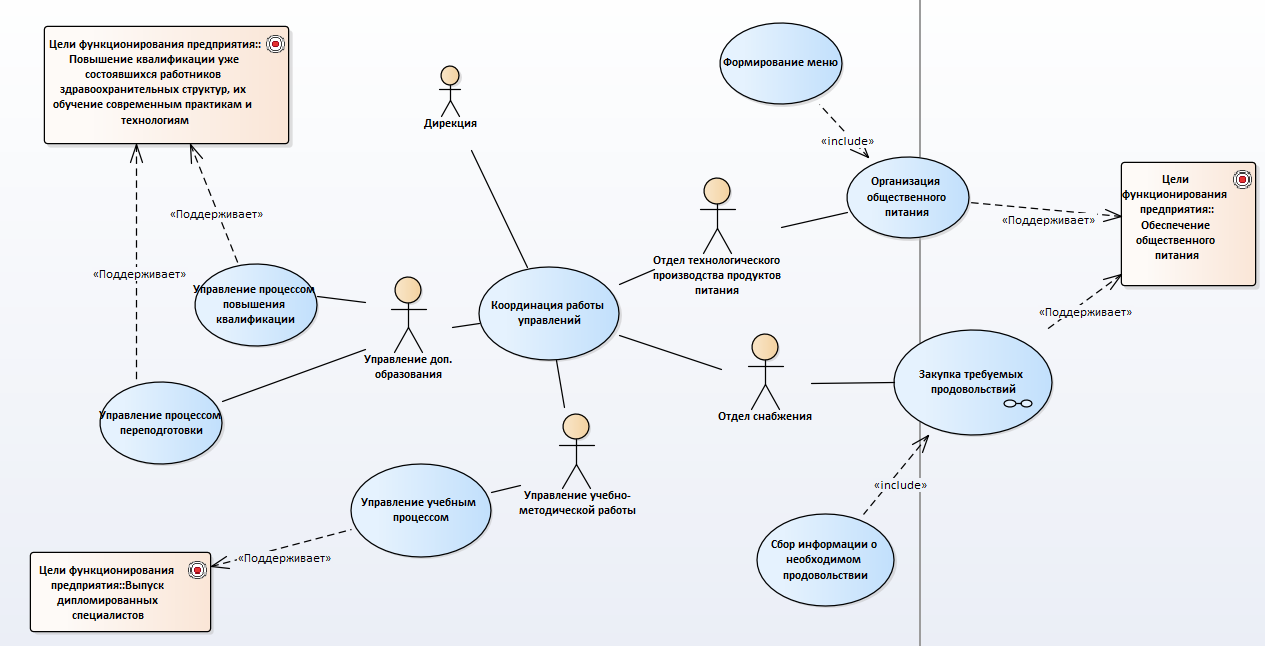


Рисунок 3. Состав бизнес-процессов

* Дирекция - осуществляет координацию работы управлений;
* Отдел технологического производства продуктов питания - осуществляет организацию общественного питания;
* Формирование меню;
* Отдел снабжения – осуществляет закупку требуемых продовольствий;
* Сбор информации о необходимом продовольствии;
* Отдел управление доп образованием осуществляет управление процессом переподготовки и процессом повышения квалификации;
* Отдел управление учебно-методической работой осуществляет управление учебным процессом.

### **Содержание бизнес-процессов**

Для подробного анализа выбран следующий процесс: “Формирование меню”.

Содержание бизнес-процесса “ Формирование меню” приведено на рисунке 4.

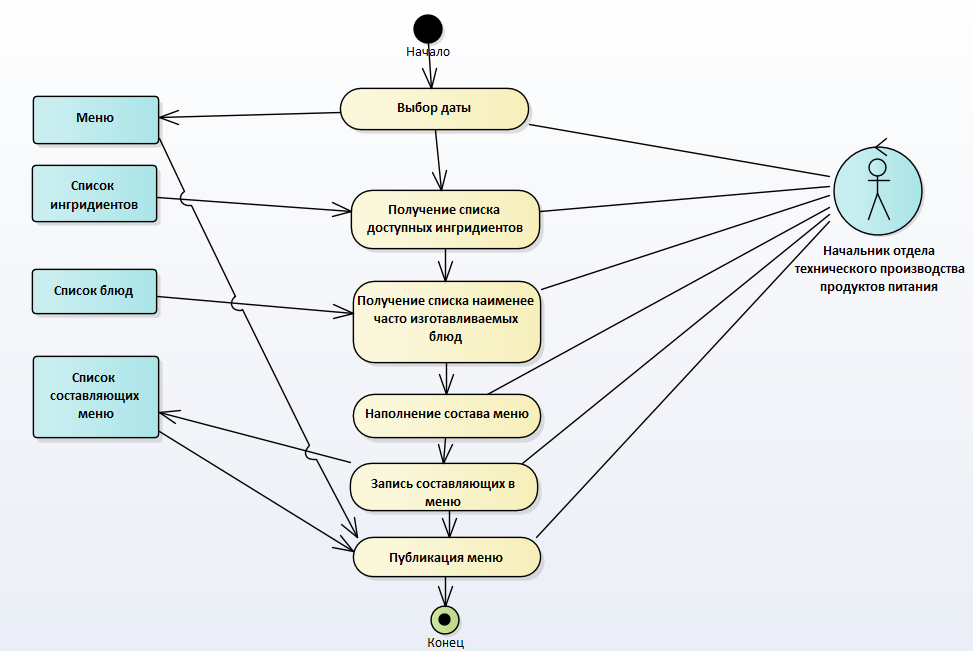


Рисунок 4. Формирование меню

Работы выполняемые начальником отдела технического производства продуктов питания:

* Выбор даты – сотрудник выбирает дату, на которую будет создано новое меню;
* Получение списка доступных ингредиентов – сотрудник получает информацию о возможности использования ингредиентов;
* Получение списка наименее часто изготавливаемых блюд – сотрудник выбирает редко используемые блюда для повышения разнообразия меню;
* Наполнение состава меню – сотрудник наполняет меню блюдами;
* Запись составляющих в меню – сотрудник вносит выбранные блюда в меню;
* Публикация меню – начальник подписывает меню;

Описание объектов:

* Меню – является выходным объектом выбора даты; входным объектом для публикации меню;
* Список ингредиентов – является входным объектом для получения списка доступных ингредиентов;
* Список блюд– является входным объектом для получения списка наименее часто изготавливаемых блюд;
* Список составляющих меню – является входным объектом для записи составляющих меню; выходным объектом для публикации меню.

### **Классы-объекты предметной области**

Описание классов-объектов предметной области приведено на рисунке 5.

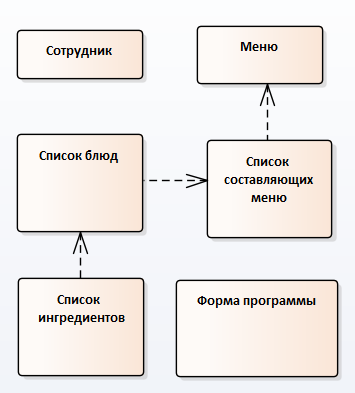


Рисунок 5. Описание классов-объектов предметной области

Выявленные классы:

* Меню – информационный объект, представляет собой созданное меню на определенную дату;
* Список составляющих меню – информационный объект, представляет собой список блюд, включаемых в меню;
* Список блюд – информационный объект, представляет собой список всех используемых блюд;
* Список ингредиентов – информационный объект, представляет собой список всех используемых ингредиентов;
* Сотрудник – информационный объект, представляет собой данные о сотруднике;
* Форма программы – информационный объект, представляет собой интерфейс программы.

### **Автоматизируемые элементы**

Предполагаемые автоматизируемые элементы приведены на рисунке 6.

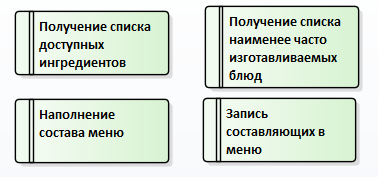


Рисунок 6. Автоматизируемые элементы бизнес-процессов

Получение списка доступных ингредиентов соответствует функции «Получение списка доступных ингредиентов». Автоматизация данного процесса позволит сократить время формирования меню и уменьшить количество ошибок.

Получение списка наименее часто изготавливаемых блюд соответствует функции «Получение списка наименее часто изготавливаемых блюд». Автоматизация данного процесса позволит сократить время формирования меню и уменьшить количество ошибок.

Наполнение состава меню соответствует функции «Наполнение состава меню». Автоматизация данного процесса позволит быстро выбрать необходимые блюда.

Запись составляющих в меню соответствует функции «Запись составляющих в меню». Автоматизация данного процесса позволит моментально включить выбранные блюда в меню.

## **Формирование требований к системе**

### **Состав требований**

Соответствие функциональных требований автоматизируемым бизнес-решениям отобразим в виде матрицы трассировки (таблица 1).

Таблица 1. Требования

|  |  |  |
| --- | --- | --- |
| **№** | **Бизнес-процесс** | **Требование** |
| 1 | Получение списка доступных ингредиентов | Обеспечение возможности загрузить информацию о ингредиентах в фиксированной форме. |
| 2 | Получение списка наименее часто изготавливаемых блюд | Обеспечение возможности автоматического выбора редко используемых блюд. |
| 3 | Наполнение состава меню | Обеспечение возможности выбора блюд, удовлетворяющих необходимым условиям. |
| 4 | Запись составляющих в меню | Обеспечение возможности сохранить состав меню на конкретную дату. |

Диаграмма соответствия функциональных требований автоматизируемым бизнес-решениям приведена на рисунке 7.

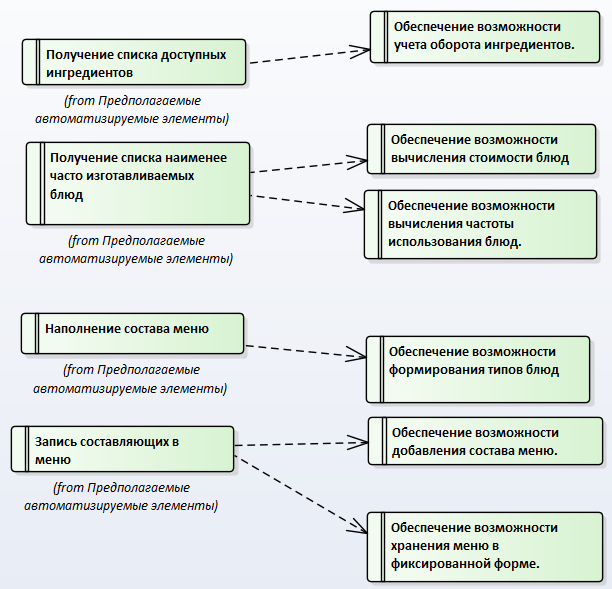


Рисунок 7. Диаграмма соответствия функциональных требований автоматизируемым бизнес-решениям

### **Состав подсистем**

Для отображения выявленных подсистем построим матрицу трассировки «подсистема – функциональные требования» (таблица 2).

Таблица 2. Подсистемы

|  |  |
| --- | --- |
| **Подсистема** | **Требование** |
| Подсистема работы с ингредиентами | Обеспечение возможности учета прихода ингредиентов. |
| Подсистема работы с блюдами | Обеспечение возможности вычисления стоимости блюд. |
| Обеспечение возможности вычисления частоты использования блюд. |
| Обеспечение возможности формирования типов блюд. |
| Подсистема работы с меню | Обеспечение возможности добавления состава меню. |
| Обеспечение возможности добавления состава меню. |
| Обеспечение возможности хранения меню в фиксированной форме. |

Диаграмма выявленных подсистем приведена на рисунке 8.

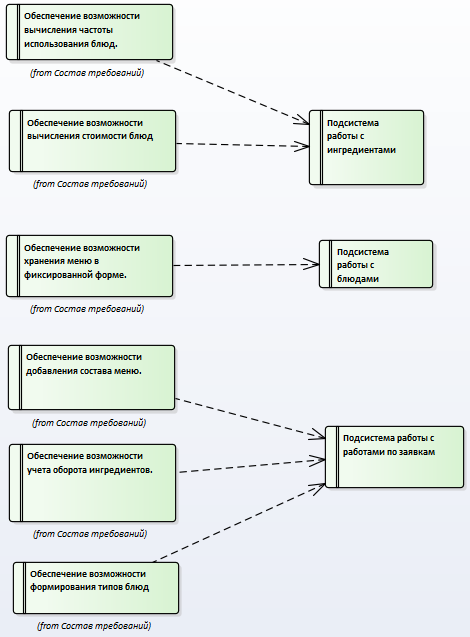


Рисунок 8. Диаграмма выявленных подсистем

### **Варианты использования**

Диаграмма вариантов использования, отражающая функциональные требования к системе и сценарии, их реализующие, приведены на рисунке 9.

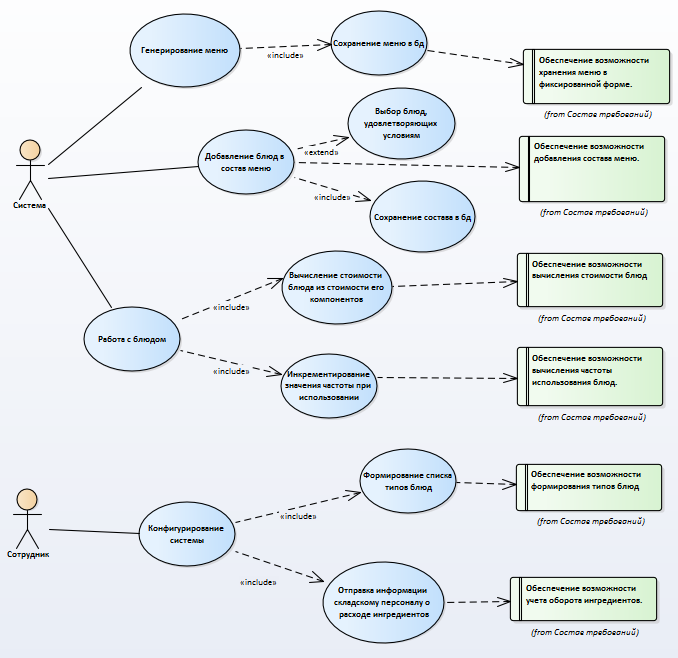


Рисунок 9. Диаграмма вариантов использования

Функции выполняемые сотрудником:

* Конфигурирование системы, которая включает в себя: формирование списка типов блюд, отправка информации складскому персоналу о расходе ингредиентов;

Функции выполняемые системой:

* Генерирование меню, включает в себя сохранение меню в БД.
* Добавление блюд в состав меню, расширяется выбором блюд, удовлетворяющих условием и сохранением состава меню в БД.
* Работа с блюдом, включает в себя вычисление стоимости блюда из стоимости составляющих и инкрементирование значения частоты при использовании.

### **Содержание сценариев**

Проведем анализ следующих сценариев: «добавление блюда», «изменение состава блюда», «ввод требований к меню» и «автоматическая генерация меню».

Сценарии выполнения остальных ВИ в данном отчете рассмотрены не будут, т.к. они имеют косвенное отношение к работе с меню.

Диаграмма деятельности сценария «добавление блюда» приведена на рисунке 10.

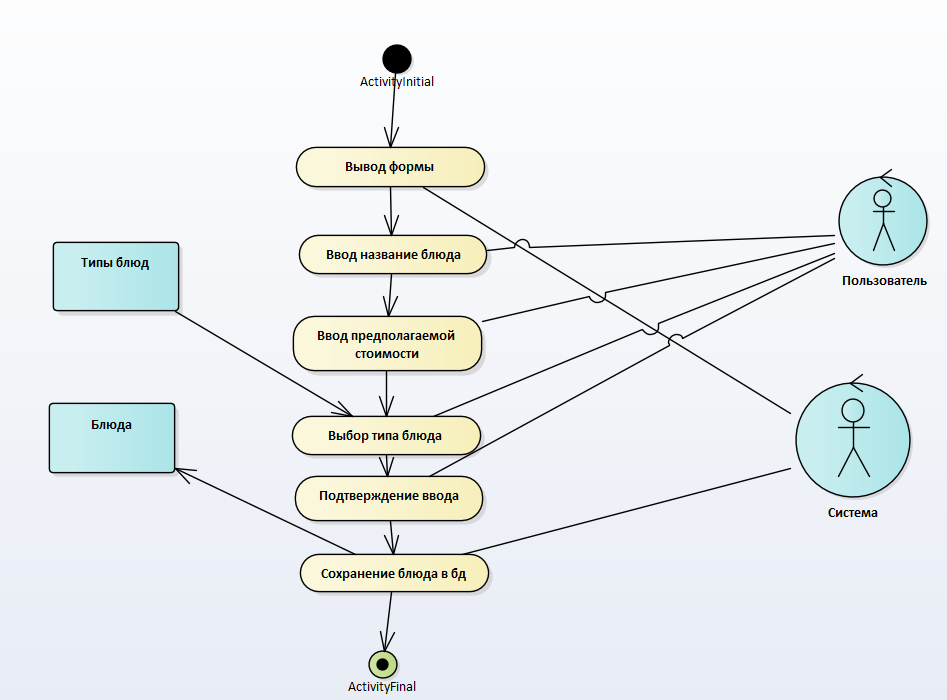


Рисунок 10. Создание блюда

В результате работы процесса выявлены следующие объекты:

* Граничные объекты:
* Форма – содержит формы «Форма создания блюда». Позволяет создавать, изменять, удалять, искать и выгружать документ в Word.
* Сущности:
* Типы блюд – содержит список используемых типов.
* Блюда – содержит список сохраненных в бд блюд

Сценарии выполняются пользователем и системой:

* Вывод блюда, открывается форма создания блюда с полем названия блюда, возможного типа и возможной стоимости (выполняется системой);
* Заполнение полей блюда (выполняется пользователем);
* Подтверждение ввода (выполняется пользователем);
* Сохранение блюда в БД (выполняется системой).

Диаграмма деятельности сценария «изменение состава блюда» приведена на рисунке 11.

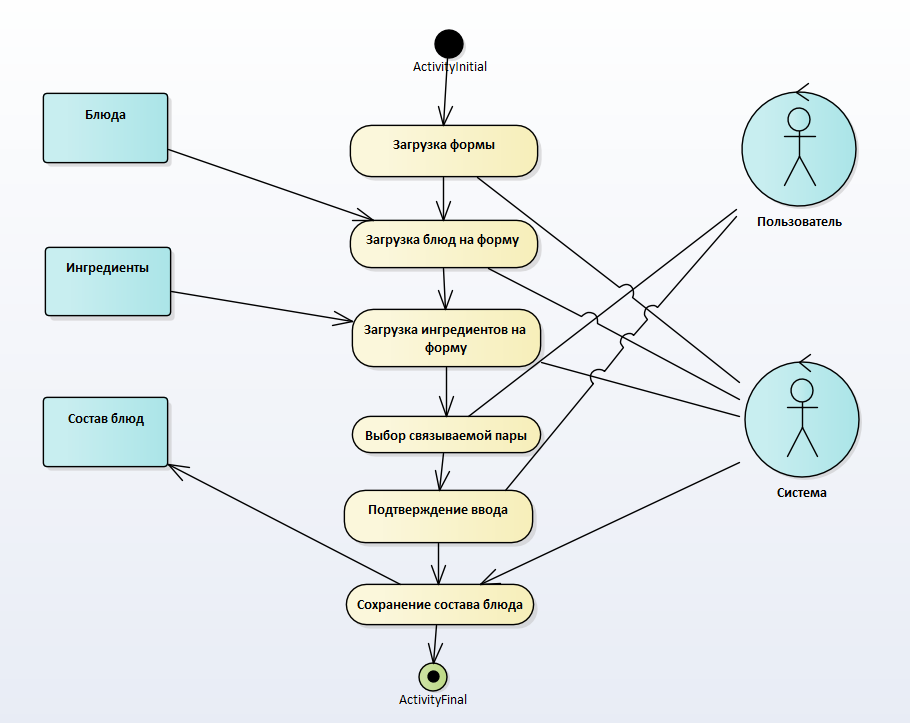


Рисунок 11. Изменение состава блюда

В результате работы процесса выявлены следующие объекты:

* Граничные объекты:
* Форма – содержит формы «Добавить состав блюд». Позволяет присваивать ингредиенты блюдам.
* Сущности:
* Блюда – содержит данные о всех блюдах;
* Ингредиенты – содержит данные о всех ингредиенты;
* Состав блюд - содержит данные о связях блюд с ингредиентами.

В выполнении сценариев участвуют система и пользователь:

* Выводится форма работы с составом блюд;
* Загружаются блюда и ингредиенты (выполняется системой);
* Сопоставляются ингредиенты блюдам (выполняется пользователем);
* Подтверждение ввода (выполняется пользователем);
* Сохранение состава блюда в БД (выполняется системой).

Диаграмма деятельности сценария «ввод требований к меню» приведена на рисунке 12.

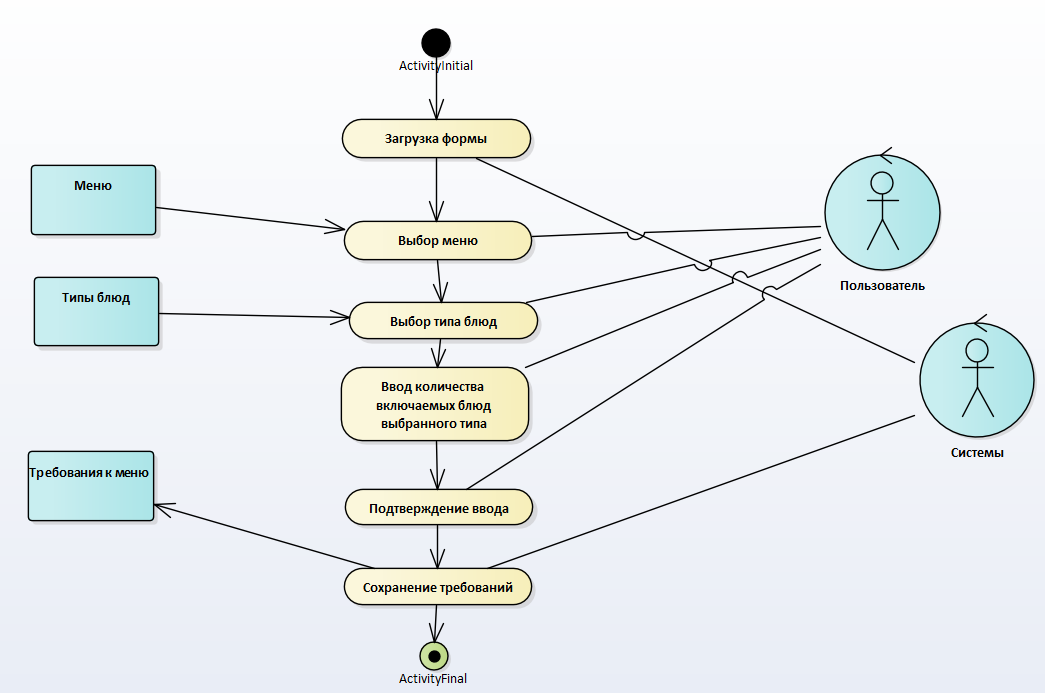


Рисунок 12. Ввод требований к меню

В результате работы процесса выявлены следующие объекты:

* Граничные объекты:
* Форма – содержит формы «Создание требований к меню». Позволяет пользователю определить каким будет генерируемое меню.
* Сущности:
* Меню - содержит информацию о всех меню.
* Типы блюд – содержит все типы блюд, содержащиеся в бд.
* Требования к меню – содержит информацию о структуре генерируемых меню.

В выполнении сценария участвуют система и пользователь:

* Выбор меню и типа блюд, пользователь выбирает в каком меню и с каким типом блюд он производит модификации.
* Ввод количества, пользователь ограничивает количество присутствующих блюд выбранного типа в меню
* Подтверждение модификации (выполняется пользователем)
* Сохранение требований, созданные требования фиксируется в БД (выполняется системой).

Диаграмма деятельности сценария «автоматическая генерация состава меню» приведена на рисунке 13

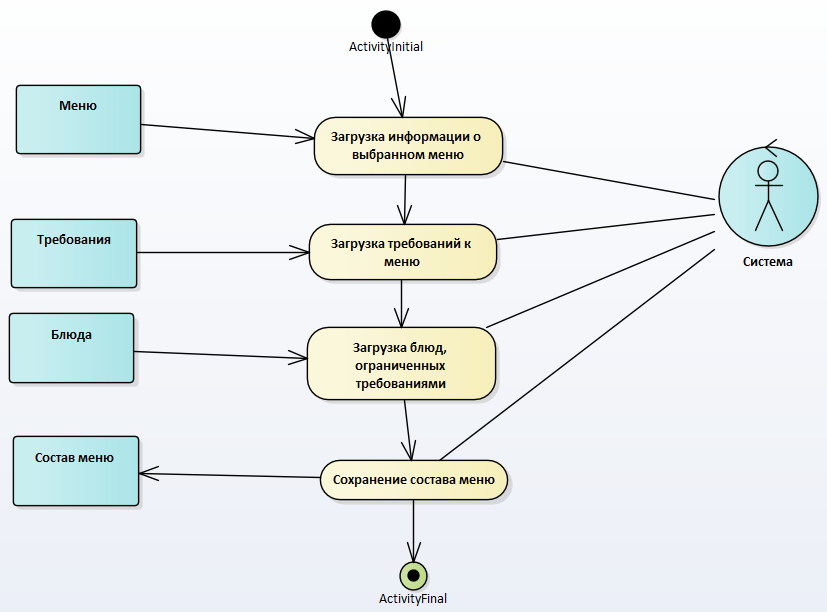


Рисунок 13. Автоматическая генерация состава меню

В результате работы процесса выявлены следующие объекты:

* Сущности:
* Меню – содержит данные о меню;
* Требования– содержит информацию о структуре генерируемых меню;
* Блюда - содержит данные о всех блюдах.
* Состав меню – содержит список блюд, включенных в меню.

В выполнении сценариев участвует система:

* Загружается выбранное меню;
* Загружаются ограничения на добавление блюд;
* Добавляется ограниченное количество блюд;
* Сохранение состава меню в БД.

Проект пользовательского интерфейса в виде диаграммы экранных форм приведены на рисунке 14.

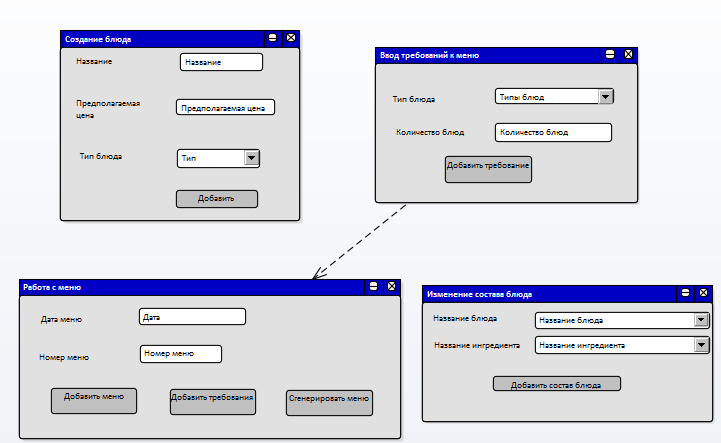


Рисунок 14. Интерфейс подсистемы «Формирование меню»

Требования к интерфейсу:

* Возможность создания блюд
* Возможность создания меню
* Возможность создания состава блюд
* Возможность изменения состава блюд
* Возможность добавления требований к меню
* Возможность автоматической генерации меню

Требования к СУБД: MSSQL Server. Должны быть реализованы сущности:

* «Блюда» - обеспечивает хранение 100 записей.
* «Ингредиенты» - обеспечивает хранение 100 записей.
* «Состав блюд» - обеспечивает хранение 250 записей.
* «Типы блюд» - обеспечивает хранение 50 записей.

Сущности «Состав меню», «Требования к меню», «Меню» являются динамическими, должна быть осуществлена возможность хранения минимум 400 записей, каждой сущности.

# **Разработка рабочего проекта**

## **Определение классов анализа**

Диаграмма классов-анализа приведена на рисунке 15:

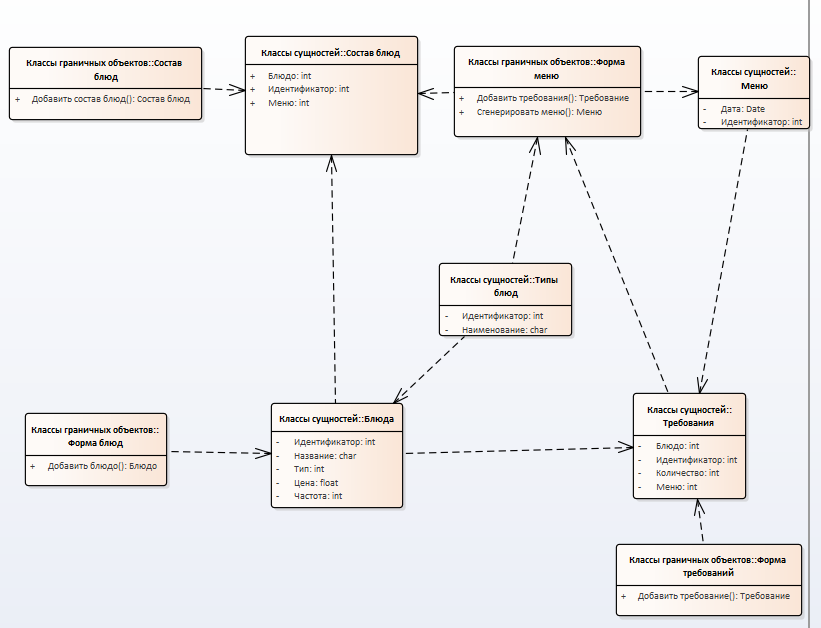


Рисунок 15. Диаграмма классов-анализа.

Выявлены классы:

Классы граничных объектов:

* Класс "Форма блюд" – определён на основе объекта «Форма блюд» сценаря «Добавление блюда»
  + Добавить блюдо, тип возврата Блюдо – сохраняет в бд сущность Блюдо;
* Класс "Форма требований" – определён на основе объекта «Форма требований» сценария «Добавление требований». Методы:
  + Добавить требование, тип возврата требование – сохраняет в бд сущность требование;
* Класс "Состав блюд" – определён на основе объекта «Форма состава блюд» сценария «Добавление составляющих блюд». Методы:
  + Добавить состав блюд, тип возврата Состав блюд– сохраняет в бд сущность состав блюд;
* Класс "Форма меню" – определён на основе объекта «Форма меню» сценария «Добавление меню». Методы:
  + Добавить меню, тип возврата Меню – сохраняет в бд сущность Меню;
  + Добавить требование, тип возврата требование – сохраняет в бд сущность требование;

Классы сущностей:

* Класс "Блюдо" – определён на основе объекта «Блюдо» сценария «Работа с блюдом». Атрибуты:
  + Идентификатор, тип int - искусственный атрибут, хранит в себе идентификатор объекта;
  + Название, тип char- содержит в себе название блюда;
  + Тип, тип int – содержит в себе указатель на сущность Тип блюда
  + Цена, тип float – содержит в себе значение стоимости блюда.
  + Частота, тип int – содержит в себе количественный показатель используемости.
* Класс "Меню" – определён на основе объекта «Меню» сценария «Добавление меню» . Атрибуты:
  + Идентификатор, тип int - искусственный атрибут, хранит в себе идентификатор объекта;
  + Дата, тип date - содержит в себе дату применения меню;
* Типы блюд - определен на основе объекта «Типы блюд» сценария «Добавление меню». Атрибуты:
  + Идентификатор, тип int - искусственный атрибут, хранит в себе идентификатор пользователя;
  + Название, тип char- содержит в себе имя типа;
* Состав блюд - определен на основе объекта «Состав блюд» сценария «Добавление блюда». Атрибуты:
  + Идентификатор, тип int - искусственный атрибут, хранит в себе идентификатор статуса;
  + Блюдо, тип int – содержит в себе указатель на сущность блюдо.
  + Меню, тип int – содержит в себе указатель на сущность меню.
* Требования - определен на основе объекта «Требования» сценария «Автоматическая генерация состава меню». Атрибуты:
  + Идентификатор, тип int - искусственный атрибут, хранит в себе идентификатор записи;
  + Блюдо, тип int – содержит в себе указатель на сущность блюдо.
  + Меню, тип int – содержит в себе указатель на сущность меню.
  + Количество, тип int – содержит в себе необходимое количество блюд в меню.

## **Определение методов объекта**

Диаграммы последовательности сценариев: «Добавление блюда», «Ввод требований к меню» и «Автоматическая генерация меню».

Диаграмма последовательности сценария «Добавление блюда» приведена на рисунке 16.

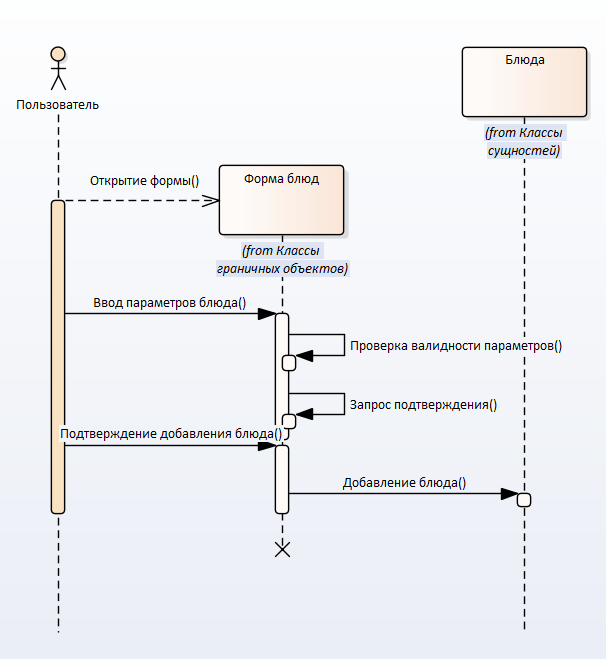


Рисунок 16. Добавление блюда

В ходе построения объекты диаграммы выявлены следующие объекты:

1. Пользователь
2. Форма блюд
3. Блюдо

Выявлены следующие сообщения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Вызывающий объект | Сообщение | Получатель сообщения | Примечание |
| 1 | Пользователь | Открыть приложение | Форма |  |
| 2 | Пользователь | Ввод параметров блюда | Форма |  |
| 3 | Форма | Проверить валидность параметров | Форма |  |
| 4 | Форма | Запросить подтверждение | Форма |  |
| 5 | Пользователь | Подтверждение ввода | Форма |  |
| 6 | Форма | Добавить блюдо | Блюдо |  |

|  |  |
| --- | --- |
| Операторы | Список сообщений |
| Форма | Проверить валидность;  Запросить подтверждение;  Добавить блюдо; |

Диаграмма последовательности сценария «Ввод требований к меню» приведена на рисунке 17.

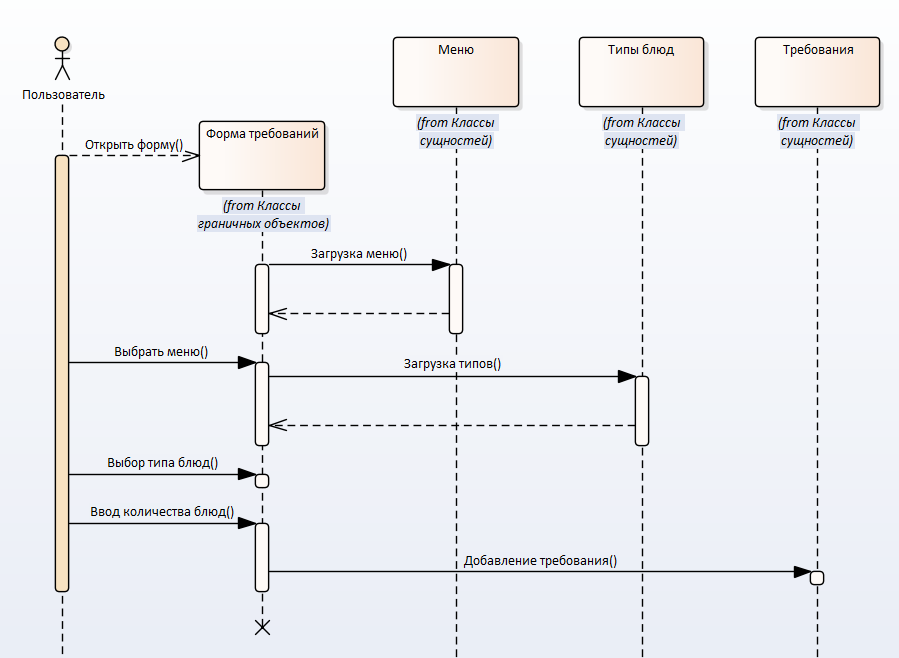


Рисунок 17. Ввод требований к меню

В ходе построения объекты диаграммы выявлены следующие объекты:

1. Пользователь
2. Форма требований
3. Меню
4. Типы блюд
5. Требования

Выявлены следующие сообщения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Вызывающий объект | Сообщение | Получатель сообщения | Примечание |
| 1 | Пользователь | Открыть форму | Форма требований |  |
| 2 | Форма требований | Загрузка меню | Меню |  |
| 3 | Меню | Вернуть меню | Форма требований |  |
| 4 | Пользователь | Выбрать меню | Форма требований |  |
| 5 | Форма требований | Загрузить типы блюд | Типы блюд |  |
| 6 | Типы блюд | Вернуть типы блюд | Форма требований |  |
| 7 | Пользователь | Выбор типа блюд | Форма требований |  |
| 8 | Пользователь | Ввод количества блюд | Форма требований |  |
| 9 | Форма требований | Добавить требование | Требования |  |

|  |  |
| --- | --- |
| Операторы | Список сообщений |
| Форма требований | Загрузить меню;  Загрузить типы блюд;  Добавить требования. |

Диаграмма последовательности сценария «Автоматическая генерация меню» приведена на рисунке 18.

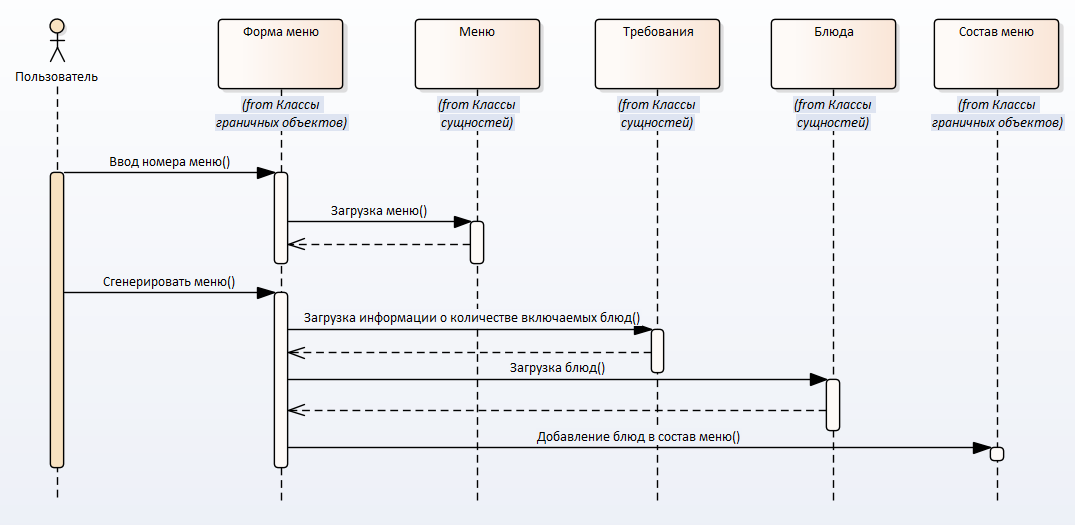


Рисунок 18. Автоматическая генерация меню

В ходе построения объекты диаграммы выявлены следующие объекты:

1. Пользователь
2. Форма меню
3. Меню
4. Требования
5. Блюда
6. Состав меню

Выявлены следующие сообщения:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Вызывающий объект | Сообщение | Получатель сообщения | Примечание |
| 1 | Пользователь | Ввод номера меню | Форма меню |  |
| 2 | Форма меню | Загрузка меню | Меню |  |
| 3 | Меню | Вернуть меню | Форма меню |  |
| 4 | Пользователь | Сгенерировать меню | Форма меню |  |
| 5 | Форма меню | Загрузка количества включаемых блюд | Требования |  |
| 6 | Требования | Вернуть количество | Форма меню |  |
| 7 | Форма меню | Загрузка блюд | Блюда |  |
| 8 | Блюда | Вернуть блюда | Форма меню |  |
| 9 | Форма меню | Добавление состава меню | Состав меню |  |

|  |  |
| --- | --- |
| Операторы | Список сообщений |
| Форма меню | Загрузка меню;  Загрузка количества включаемых блюд;  Загрузка блюд;  Добавление состава меню; |

## **Выбор технологии реализации**

Система будет реализована на языке программирования C# в среде разработке Microsoft Visual Studio.

Пользователь будет взаимодействовать с системой через интерфейс в виде веб-страницы ASP.NET MVC приложения.

Реализация требования возможна в рамках операционных систем семейства Windows, которые установлены на предприятии.

Пользовательские компоненты будут взаимодействовать с данными с помощью технологии ADO.NET, которая будет реализована через соответствующие компоненты среды проектирования Visual Studio.

Для развертывания приложения будет использоваться:

* Клиентский узел для установки клиентского приложения. Характеристики:
  + CPU: AMD Athlon-5150 1,6Ghz;
  + ОЗУ: 2 ГБ;
  + HDD: 250 ГБ.

Серверный узел для установки базы данных. Характеристики:

* + CPU: Intel core i5 3Ghz;
  + ОЗУ: 6 ГБ;
  + HDD: 1000 ГБ.

Диаграмма начальной среды развертывания приведена на рисунке 19:

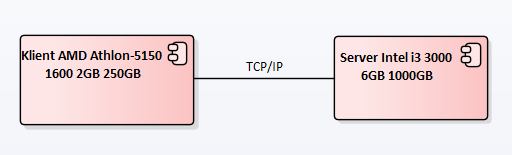


Рисунок 19. Начальная среда развертывания

## **Проектирование хранилища данных**

Предполагаемые состав сущностей, отображаемых в БД приведен на рисунке 20:

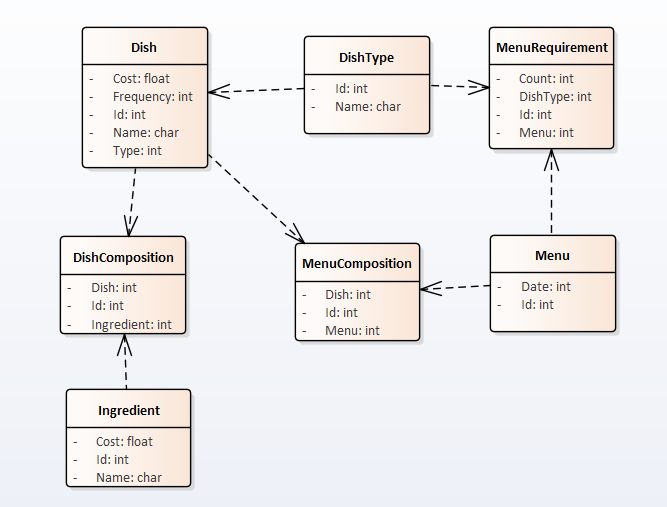


Рисунок 20. Состав сущностей

На основе состава сущностей, созданы таблицы БД:

* " MenuRequirements" – хранит в себе возможные статусы заявки. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор требования;
  + Menu, тип int – хранит в себе идентификатор меню, на которое действует требование
  + DishType, тип int – хранит в себе идентификатор типа блюда
  + Count, тип int – хранит количество требуемых блюд

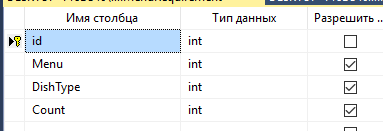


Рисунок 21. Таблица MenuRequirement

* " MenuComposition" – хранит в себе возможные статусы работы. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор состава меню;
  + Menu, тип int - хранит в себе идентификатор меню;
  + DishComposition, тип int – хранит идентификатор блюда, включаемого в меню.

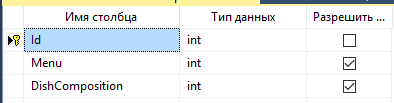


Рисунок 22. Таблица MenuComposition

* " Menu" – хранит в себе возможные статусы работы. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор меню;
  + Date, тип date – содержит дату применения меню;

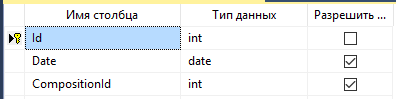


Рисунок 23. Таблица Menu

* "Ingredient" – хранит в себе возможные статусы работы. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор ингредиента;
  + Name, тип nvarchar– содержит название ингредиента;
  + Cost, тип nvarchar – содержит стоимость ингредиента.

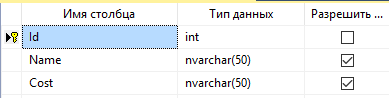


Рисунок 24. Таблица Ingredient

* "DishType" – хранит в себе возможные статусы работы. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор типа блюда;
  + Name, тип nvarchar – содержит название типа блюд;

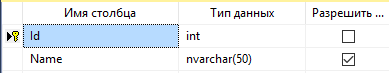


Рисунок 25. Таблица DishType

* " DishComposition " – хранит в себе возможные статусы работы. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор состава блюда;
  + Dish, тип int – хранит в себе идентификатор блюда;
  + Ingredient, тип int – хранит в себе идентификатор ингредиента, включаемого в состав блюда.

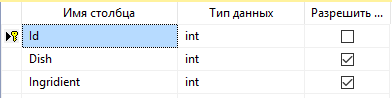


Рисунок 26. Таблица DishComposition

* " Dish " – хранит в себе возможные статусы работы. Атрибуты:
  + Id, тип int – искусственный атрибут, хранит в себе идентификатор блюда;
  + Name, тип int – содержит название блюда;
  + Cost, тип char – содержит стоимость блюда;
  + Type, тип int – хранит в себе идентификатор типа блюда;
  + Frequency, тип int – количество использований блюда.

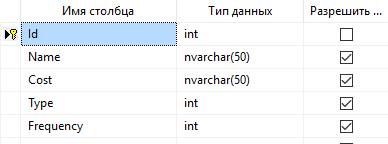
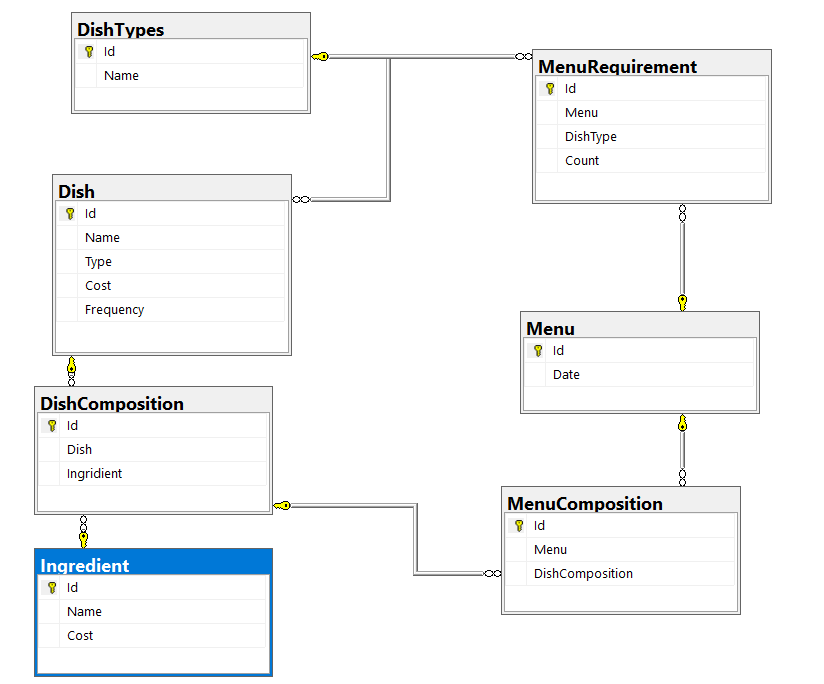


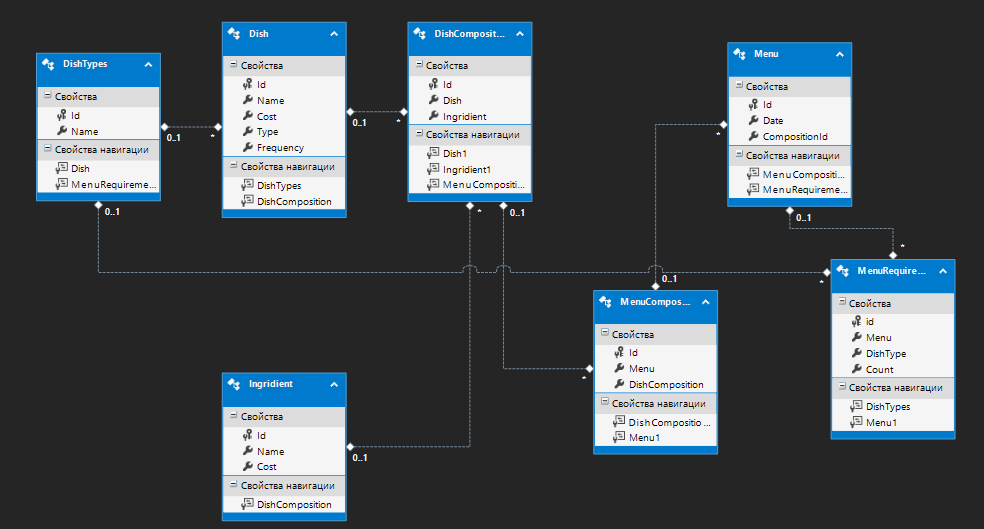
Рисунок 27. Таблица Dish

# **Разработка программного кода**

Так как к моменту создания клиента база данных уже была разработана, то логичным было использовать технологию Entity Framework, с подходом Database First.



Сгенерированные модели из бд с помощью EF:



Состав компонентов, содержащих программный код представлен на рисунке 28:

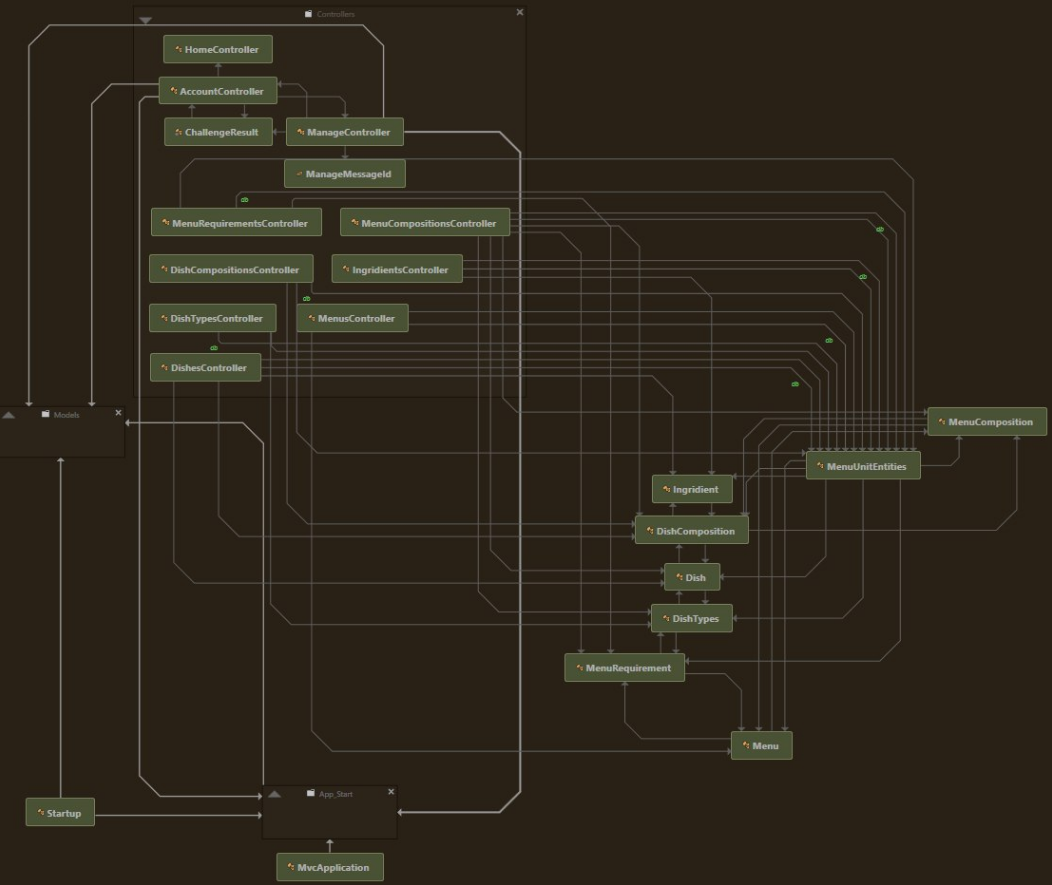
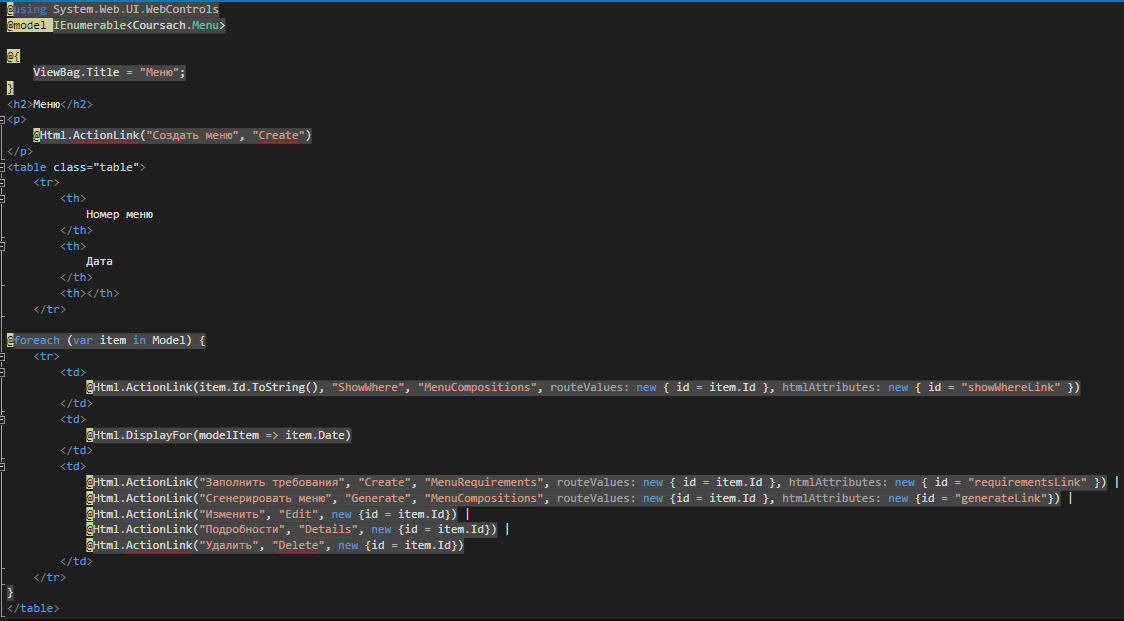


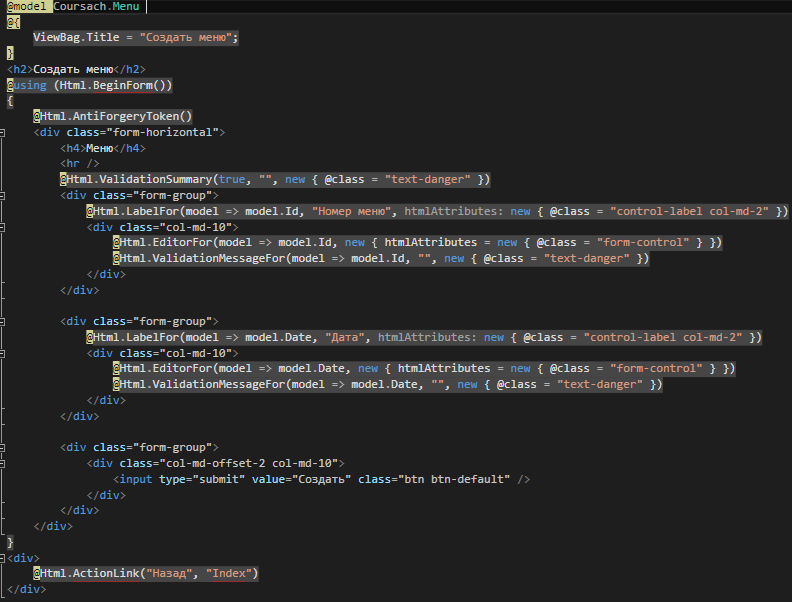
Рисунок 28. Состав компонентов программного кода

Проект реализован в виде

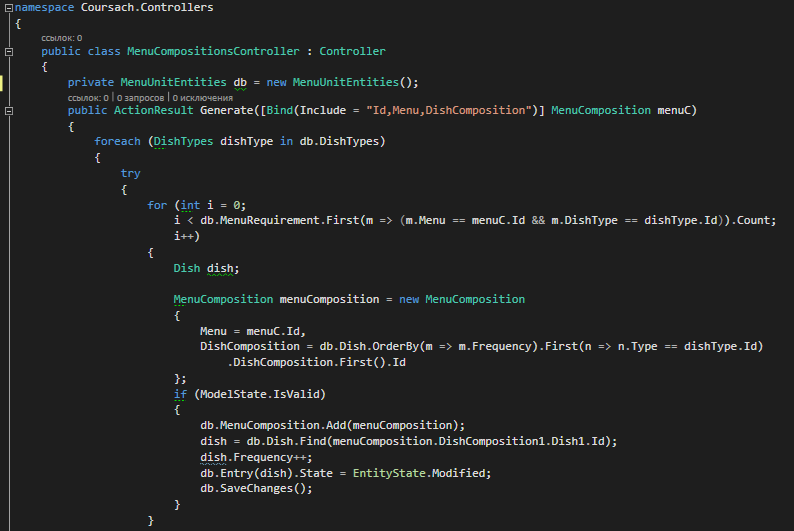
* MvcApplication, который организует проекты, элементы проектов и решений в решение;
* Startup.cs – запуска программы.
* Web.config – файл соединения и конфигурации приложения.
* Пакет Controllers – содержит методы взаимодействия страниц веб-приложения
* Пакет Models – содержит модели сущностей веб-приложения

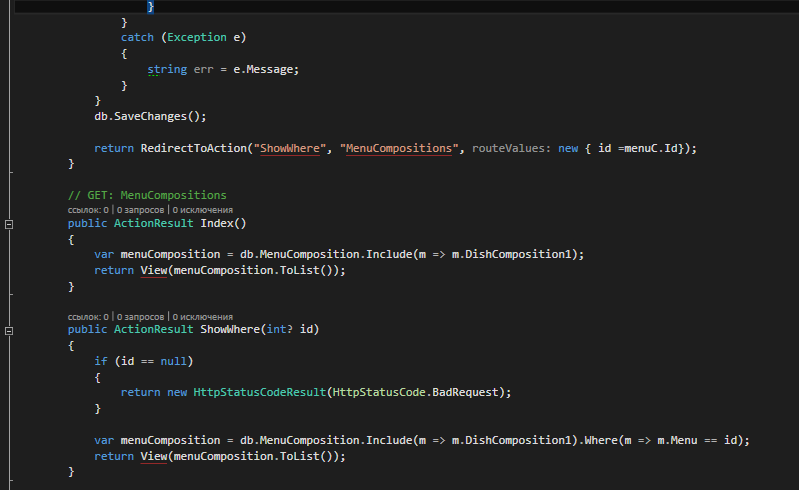
Пример кода представления на основе страницы Index.cshtml пакета Menu представлен на рисунках ниже.

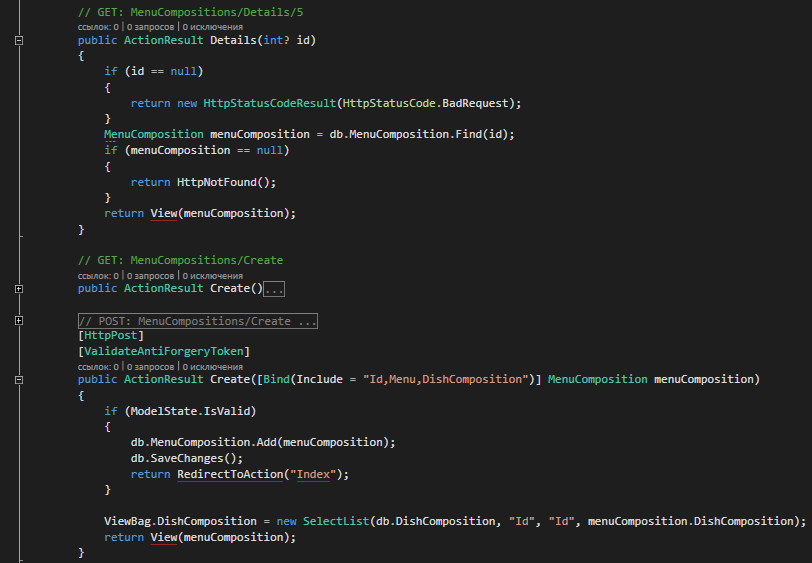




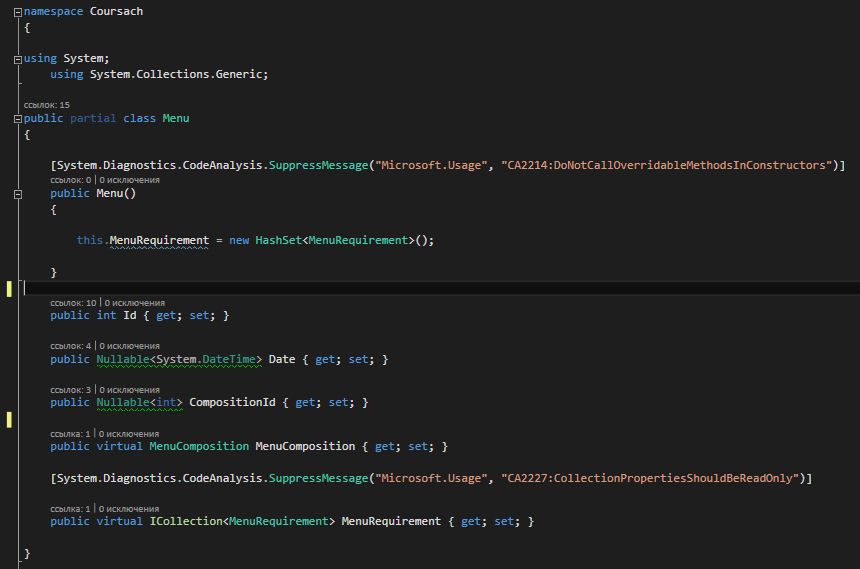
* + Dishes - пакет содержащий страницы, отвечающие работу с блюдами;
  + DishTypes - пакет содержащий страницы, отвечающие работу с типами блюд;
  + DishCompositions - пакет содержащий страницы, отвечающие работу с составом блюд;
  + Ingredients - пакет содержащий страницы, отвечающие работу с ингредиентами;
  + MenuCompositions - пакет содержащий страницы, отвечающие работу с составом меню;
  + MenuRequirements - пакет содержащий страницы, отвечающие работу с требованиями к меню;
  + Menus - пакет содержащий страницы для работы с меню;
  + \_ViewStart.cshtml - содержит ссылку на настройки стартовой страницы приложения;
  + Web.config – файл соединения и конфигурации приложения.
* Controllers - пакет, содержащий контроллеры обрабатывающие действия пользователей со страницами, контроллеры обрабатывают действия на одноименных веб-страницах;
  + MenuCompositionsController.cs - обрабатывает действия пользователя с моделью MenuCompositions, в контроллере реализованы методы: Generate(), ShowWhere(), Create(), Index(), Details(). Код описывающий контроллер приведен на рисунках ниже, код, описывающий остальные контроллеры, схож с приведенным;







* + DishCompositionsController.cs - обрабатывает действия пользователя с моделью DishCompositions, в контроллере реализованы методы:, Create(), Index();
  + DishesController.cs - обрабатывает запросы пользователя по взаимодействию с моделью Dishes, в контроллере реализованы методы: Calculate(), Create(), Edit(), Index;
  + HomeController.cs - обрабатывает действия пользователя на главной странице приложения, в контроллере реализованы методы:About(), Contact(), Index();
  + ManageController.cs - обрабатывает действия пользователя с моделью ManageViewModels, в контроллере реализованы методы:AddPhoneNumber(), ChangePassword(), Index(), ManageLogins(), SetPassword(), VeryfyPhoneNumber();
  + IngredientsController.cs - обрабатывает действия пользователя с моделью Ingredients, в контроллере реализованы методы: Create(), Index();
  + MenusController.cs - обрабатывает действия пользователя с моделью Menus, в контроллере реализованы методы: Create, Index(), Edit().
* Models - пакет, содержащий представления БД, через которые ведется работы с таблицами БД:
  + IdentityModels.cs;
  + ManageViewModels.cs;
  + AccountViewModels.cs - данные модели отвечают за описание поведения стандартной авторизации и управления авторизованными пользователями;
  + При формировании моделей использовался подход DataBase First. Была сформирована модель базы данных EntityModel.edmx. В сформированной модели присутствуют классы:
    - Menu.cs - соответствует таблице БД Menu, код автоматически сгенерированный средой разработки представлен на рисунках ниже, принцип формирования остальных классов схож с данным;



* + - Dish.cs - соответствует таблице БД Dish;
    - Ingredient.cs - соответствует таблице БД Ingredient;
    - DishComposition.cs - соответствует таблице БД DishComposition;
    - DishType.cs - соответствует таблице БД DishType;
    - MenuComposition.cs - соответствует таблице БД MenuComposition;
    - MenuRequirement.cs - соответствует таблице БД MenuRequirement;

# **Развертывание**

Диаграмма среды развертывания представлена на рисунке 29:

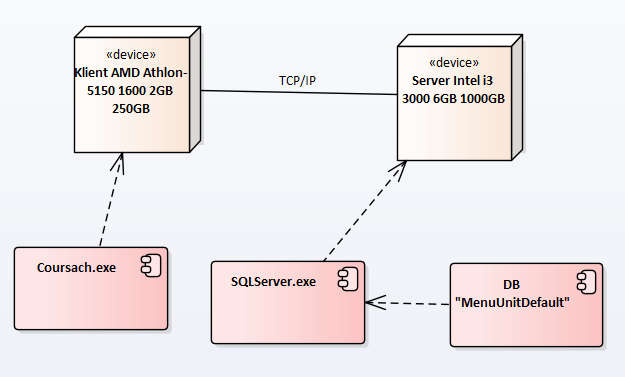


Рисунок 29. Среда развертывания

На серверной машине предприятия уже установлен сервер SqlServer. На сервер необходимо развернуть базу данных MenuUnitDefault.

На клиентских машинах необходимо установить .NET Framework, а так же скопировать файл запуска программы (exe-файл).

После загрузки всех компонентов необходимо провести тестовый запуск программы.

## **Интерфейс системы**

Главная страница приложения представлена на рисунке 30. Реализованы возможности добавления блюд, ингредиентов, состава блюд, расчета стоимости блюд и автоматической генерации меню.

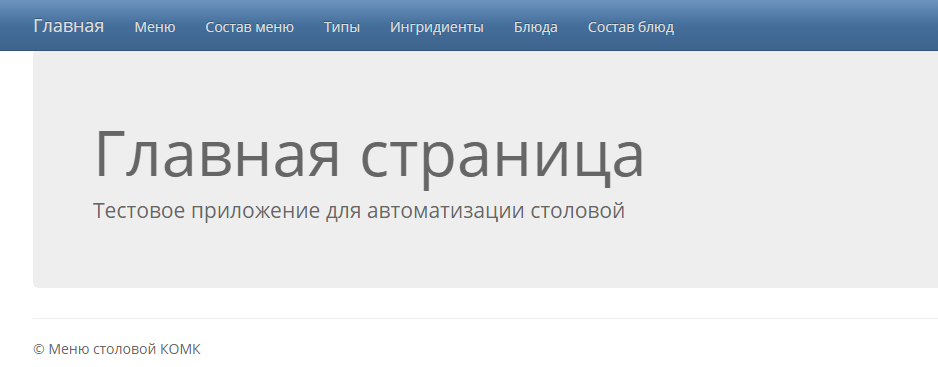


Рисунок 30. Главная страница приложения

Страница просмотра типов блюд представлена на рисунке 31. Реализована возможность добавления нового типа.

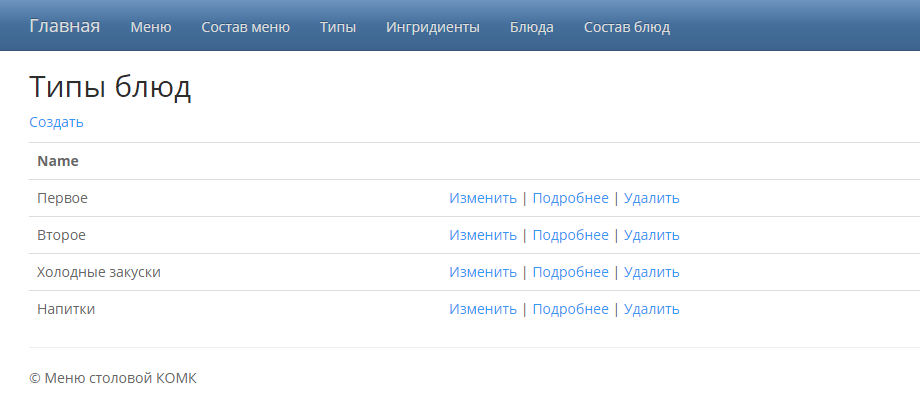


Рисунок 31. Страница просмотра типов блюд

Страница добавления типа блюда представлена на рисунке 32. Реализована возможность добавления нового типа блюд.

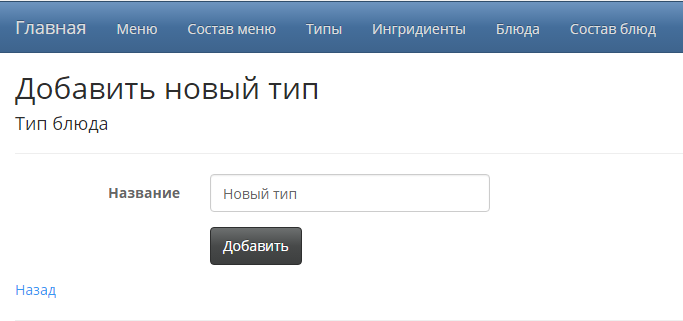


Рисунок 32. Страница добавления типа

Страница просмотра ингредиентов представлена на рисунке 33. Реализована возможность добавления нового ингредиента.

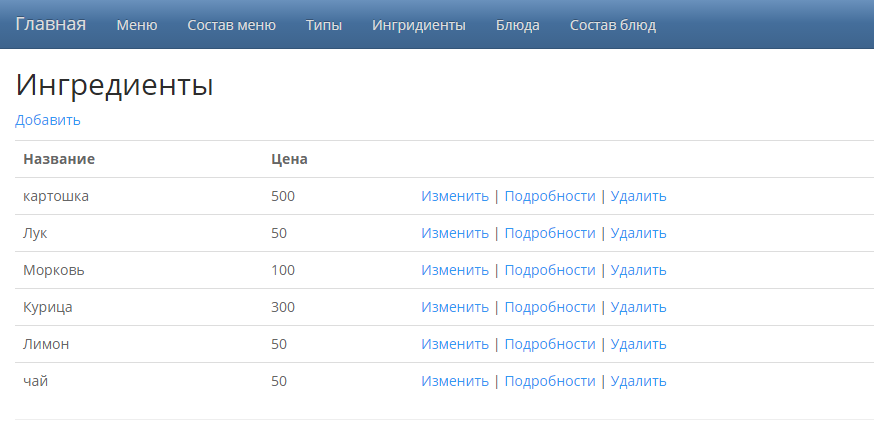


Рисунок 33. Страница добавления ингредиента

Страница добавления ингредиента представлена на рисунке 34. Реализована возможность добавления нового ингредиента.

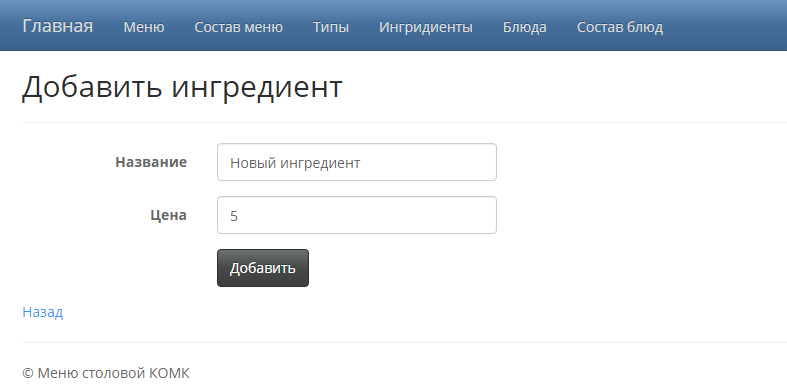


Рисунок 34. Страница добавления ингредиента

Страница просмотра блюд представлена на рисунке 35. Реализована возможность добавления нового блюда, расчета стоимости.

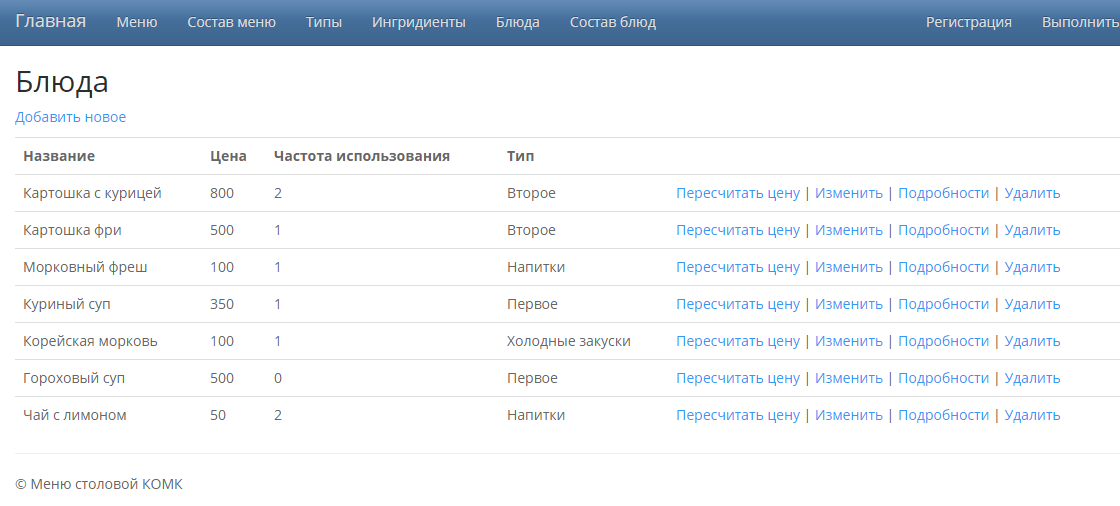


Рисунок 35. Страница добавления блюда

Страница добавления блюда представлена на рисунке 36. Реализована возможность добавления нового блюда.

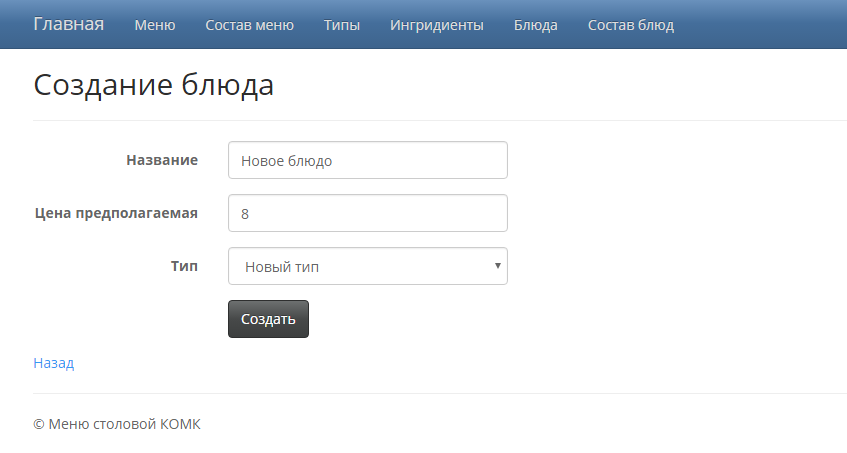


Рисунок 36. Страница добавления блюда

Страница добавления компонентов блюда представлена на рисунке 37. Реализована возможность добавления составляющих блюда.

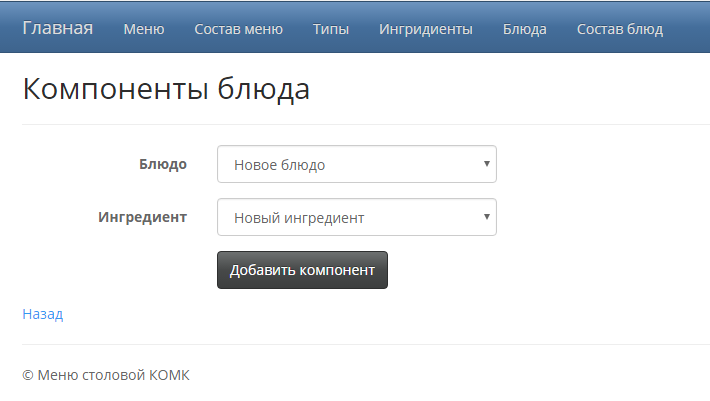


Рисунок 37. Страница добавления состава блюда

Страница просмотра меню представлена на рисунке 38. Реализована возможность добавления меню, добавления требований, генерирования состава меню.

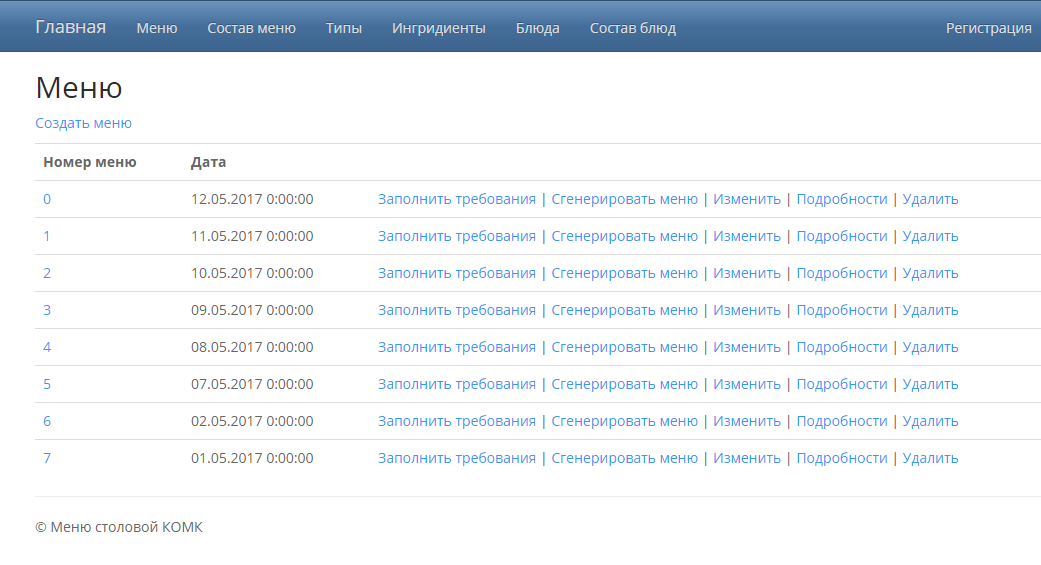


Рисунок 38. Страница просмотра списка меню

Страница добавления меню представлена на рисунке 39. Реализована возможность добавления меню.

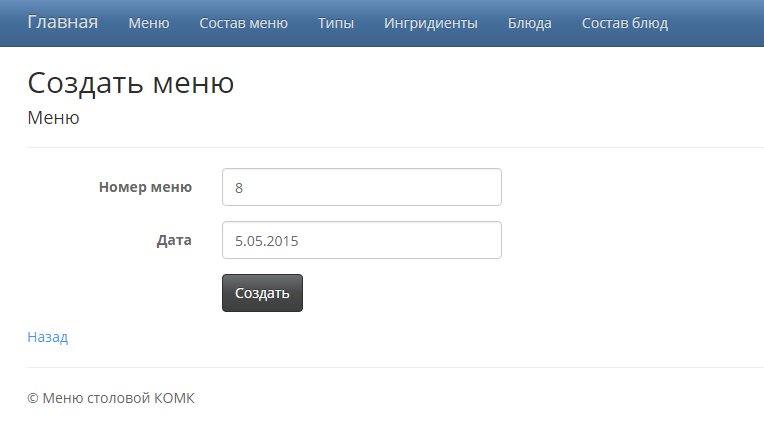


Рисунок 39. Страница создания меню

Страница добавления требований представлена на рисунке 40. Реализована возможность добавления требований.

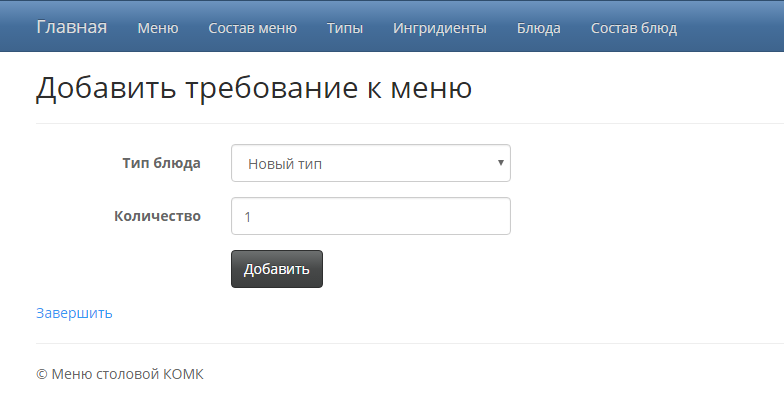


Рисунок 40. Страница добавления требований к меню

Страница просмотра состава меню представлена на рисунке 41. Реализована возможность просмотра состава меню.

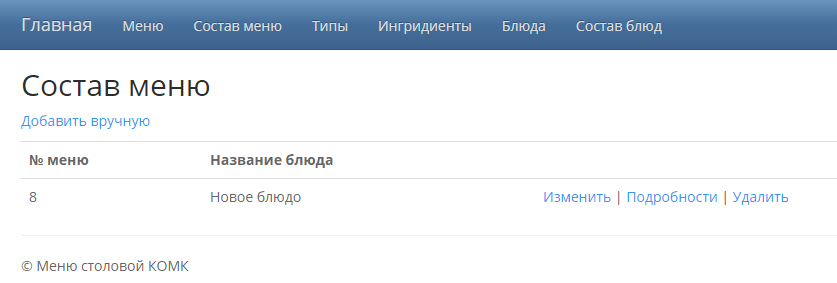


Рисунок 41. Страница просмотра состава меню

# **Разработка сценариев тестирования**

Для тестирования выбраны следующие сценарии «Добавление блюда», «Расчёт стоимости блюда», «Создание меню».

Сценарий «Добавление блюда»:

1. Пользователь открывает страницу ингредиентов, заполняет поля и нажимает «Добавить», в БД появляется новая запись в таблице Ingredient.

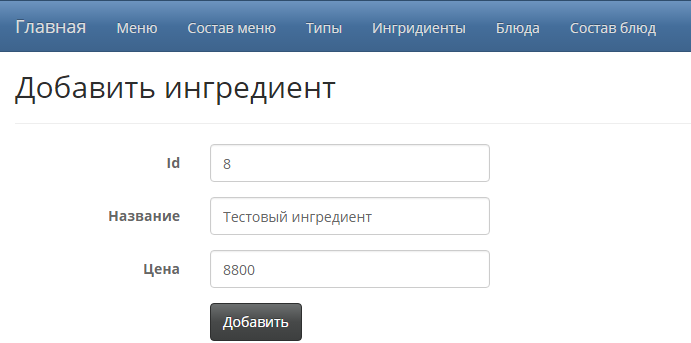


Рисунок 42. Страница «Ингредиенты»

1. Пользователь открывает страницу блюд, заполняет поля и нажимает «Добавить», в БД появляется новая запись в таблице Dish.

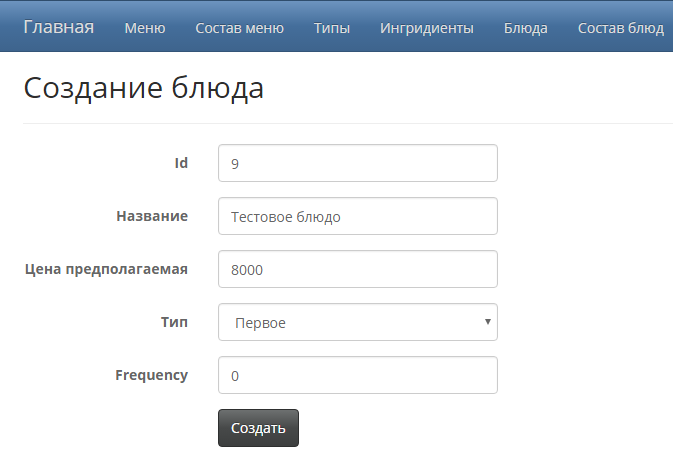


Рисунок 43. Страница «Блюда»

1. Пользователь открывает страницу Состав блюд, выбирает блюдо и ингредиент, который входит в состав блюдо. Затем нажимает «Добавить», в БД появляется новая запись в таблице DishComposition

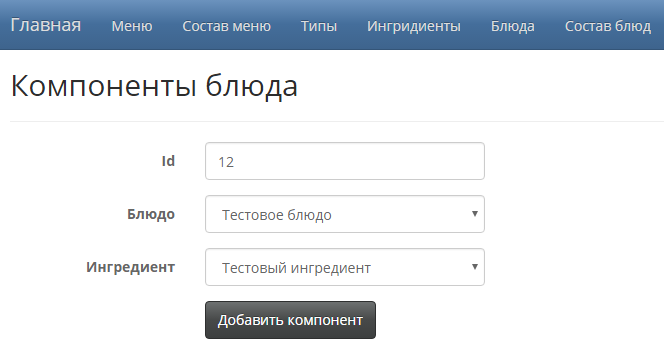


Рисунок 44. Страница «Состав блюд»

Сценарий «Расчёт стоимости блюда»:

* + - 1. Пользователь открывает страницу Блюда и нажимает «Пересчитать цену» в строке с нужным блюдом, цена блюда пересчитывается исходя из суммы цен составляющих его ингредиентов, в таблице Dish базы данных обновляется значение Cost у выбранного блюда.

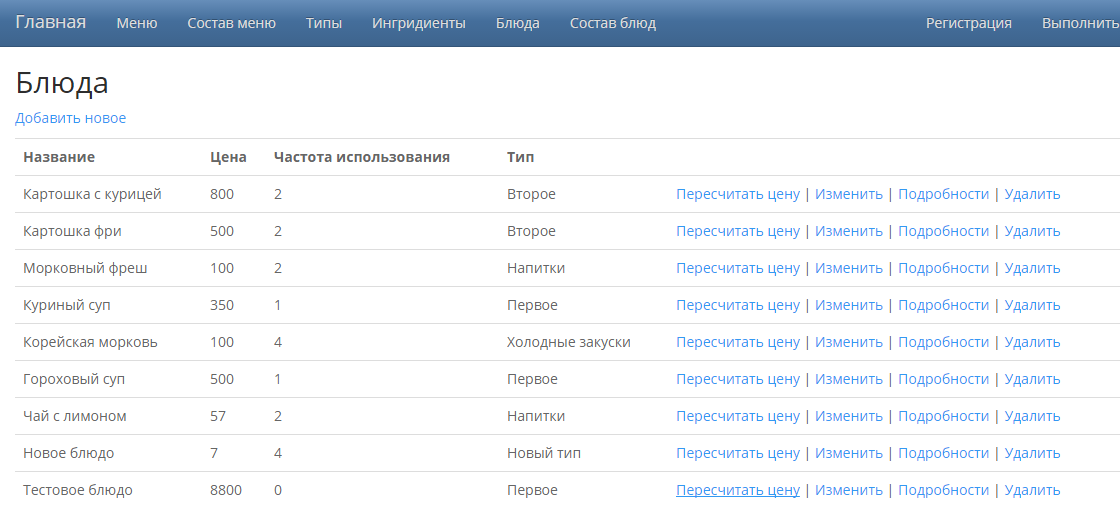


Рисунок 45. Страница «Блюда»

Сценарий «Создание меню»:

* + - 1. Пользователь открывает страницу Меню и заполняет поля, затем нажимает «Создать». В таблице Menu базы данных появляется новая запись.

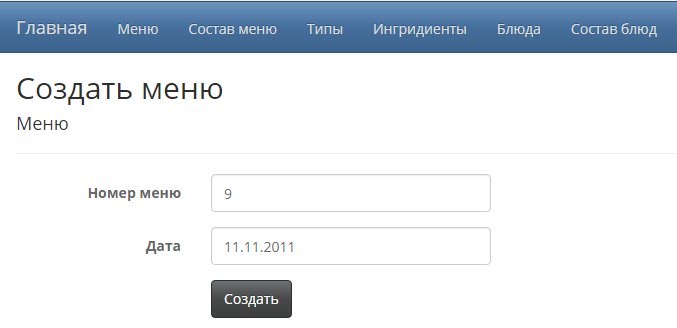


Рисунок 45. Страница «Меню»

1. Пользователь открывает страницу Требования и заполняет требования к новому меню, в базе данных появляется запись в таблице MenuRequirement.

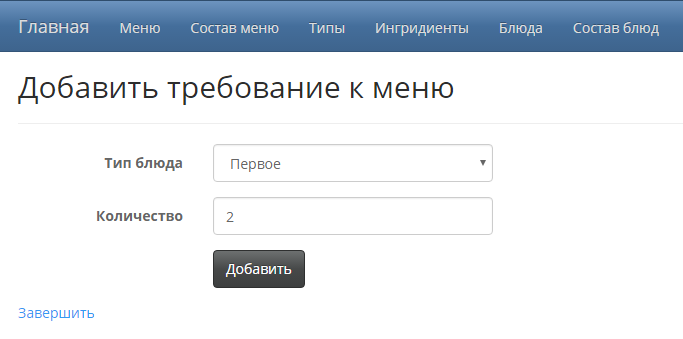


Рисунок 46. Страница «Требования»

1. Пользователь добавляет дополнительное требование для лучшей проверки функциональности. В базе данных появляется еще одна запись в таблице MenuRequirement.

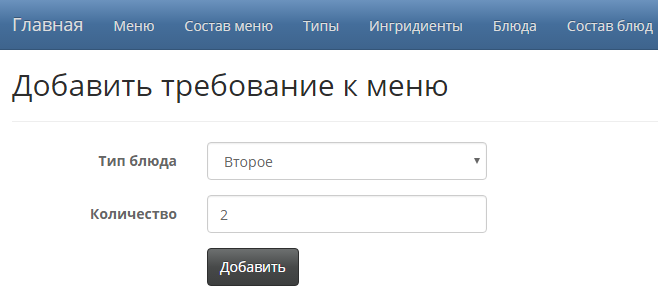


Рисунок 47. Страница «Требования»

1. Пользователь открывает страницу Меню (Рисунок 38) и нажимает «Сгенерировать меню». В базе данных появляется несколько записей в таблице MenuComposition. Пользователя перенаправляет на страницу отображающую результат генерации меню.

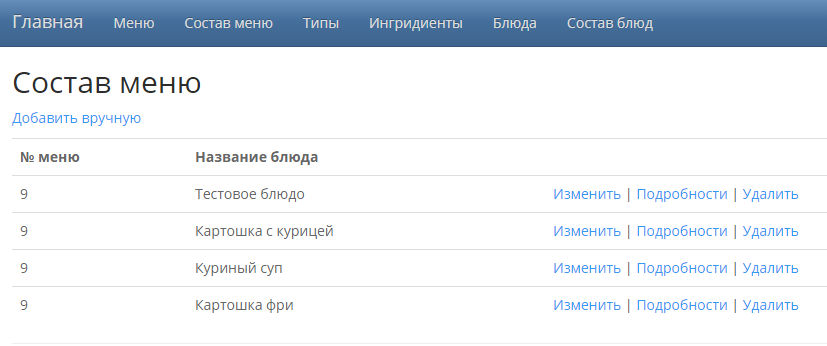


Рисунок 48. Страница «Состав меню»

# 

# **Список литературы**

1. Методические указания по выполнению курсового проекта по дисциплине «Методы и средства проектирования информационных систем и технологий» для бакалавров направление подготовки 09.03.02 «Информационные системы и технологии» профиль «Информационные системы и технологии». Составитель О.Н. Ванеев, Кемерово, 2017 г.;
2. <https://metanit.com/sharp/mvc.php> ;
3. <https://professorweb.ru/my/ASP_NET/mvc/level1/>
4. http://dotnet.today/ru/aspnet5-vnext/client-side/bootstrap.html