1. **Введение**
   1. **Назначение проекта**

Приложение для формирования меню ресторана служит для удобства посетителей и поваров: регистрация, бронирование места и заказ блюд и напитков (для посетителей), добавление блюд и напитков в меню (для поваров).

* 1. **Соглашения, приятые в документах**

Документация должна быть оформлена по ГОСТу 19.201-78 или 19.202-78. Шрифт в документации – Times New Roman, размер – 16: для заголовков пунктов, 14: для заголовков подпунктов и основного текста. Жирным шрифтом выделяются заголовки пунктов и подпунктов.

* 1. **Границы проекта**

Программный продукт создается с целью предоставления гостям ресторана удобного приложения, с помощью которого они смогут бронировать места заранее, делать заказы, регистрироваться в нем и получать бонусы, как постоянные гости. Помимо этого, данный программный продукт позволяет зарегистрироваться и поварам данного ресторана для добавления их данных в базу, также они будут иметь возможность добавления новых блюд и напитков в меню.

* 1. **Ссылки**

ГОСТы для документации: <http://www.rugost.com/index.php?option=com_content&view=category&id=19&Itemid=50>

Git хостинг1, на котором будет размещен проект: <https://github.com/pluxyryprince>

Шаблон для составления технического задания: <https://habr.com/ru/post/328822/>

Шаблон для составления спецификации программного продукта: <https://analytics.infozone.pro/requirements-analysis/template-specification-requirements/>

1. **Общее описание**
   1. **Общий взгляд на продукт**

Продукт является совершенно новым решением как касаемо конкретного ресторана, так и ресторанного бизнеса в целом. Предлагаемый функционал расширяет понимание ресторанных приложений, которые ранее были полезны лишь для гостей заведения. Данный программный продукт доступен для пользования не только посетителям, но и поварам данного ресторана, что упростит процесс обновления меню.

Для визуализации вышесказанного была составлена контекстная диаграмма, отражающая связи внутри системы:

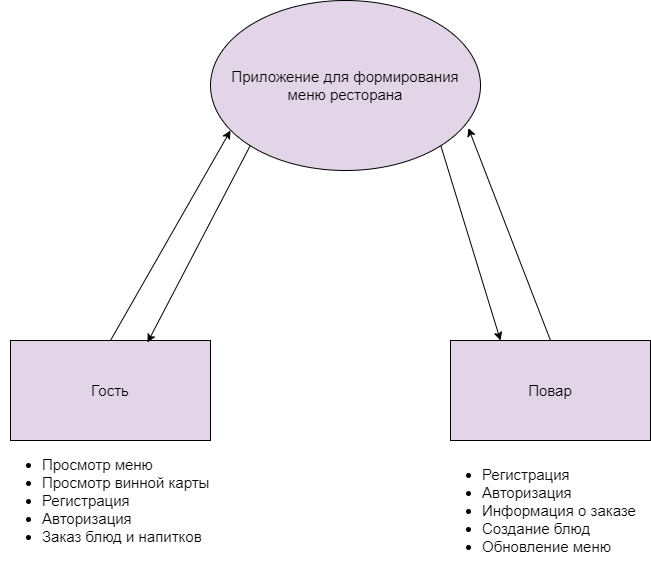


Рисунок 1 – Контекстная диаграмма

* 1. **Классы и характеристики пользователей**
     1. Гости или посетители ресторана. Бывают постоянные или не постоянные. Для не постоянных клиентов максимум нужно просмотреть меню, тогда как для постоянных клиентов нужно предусмотреть бонусы, специальные предложения и т.д. Регистрация в приложении доступна всем. Более того, для зарегистрированных посетителей будет реализована система накопительных бонусов.
     2. Повара, работающие в ресторане. Характеризуются более обширным доступом в приложении: помимо просмотра меню, у них есть возможность его редактировать, т.е. добавлять новые блюда и напитки в него. Регистрация и авторизация поваров ресторана отдельна от регистрации и авторизации посетителей.

**2.3. Операционная среда**

2.3.1 Минимальные системные требования2:

* Процессор Intel Core 2 Duo E4500 или Athlon 64 X2 5000+;
* Видеокарта Nvidia GeForce 7600 GS или ATI Radeon X800 GTO;
* Оперативная память не менее 2 гб;
* Место на накопителе не менее 100 мб;
* Операционная система Windows 7 Home Basic или выше.

2.3.2 Географическое положение баз данных

Сервер3, на котором будут расположены базы данных, должен иметь географическое положение не далее границ европейской части Евразийского материка. Допускается расположение за пределами Российской федерации.

2.3.3 Географическое положение пользователей

Основная часть пользователей приложения находится в РФ и странах СНГ. Однако не исключено, что будут пользователи из других стран или материков.

**2.4 Ограничения дизайна и реализации**

Базы данных должны быть реализованы в СУБД4, работающей на языке запросов SQL. У баз данных и содержащихся в них таблицах должны быть понятные и «говорящие» названия.

Программный продукт должен быть создан на языке программирования C# и иметь совместимость с языком запросов SQL. Дизайн приложения должен быть реализован в среде разработки MS Visual Studio 2019 с использованием стандартных элементов дизайна. Допускается использование сторонних цветовых гамм, шрифтов и рисунков для исполнения пожеланий заказчика.

1. **Функции системы**
   1. Регистрация
      1. Описание: Данная функция доступна как посетителям ресторана, так и поварам. Для данных посетителей и поваров использованы разные базы данных.
      2. Требования:

* Ввод ФИО;
* Ввод логина;
* Ввод пароля;
* Занесение данных в базу.
  1. Авторизация
     1. Описание: данная функция доступна только зарегистрированным пользователям и поварам.
     2. Требования:
* Ввод логина;
* Ввод пароля;
* Сравнение введённых данных с базой.
  1. Просмотр меню
     1. Описание: из базы данных вытягивается список блюд и выводится на соответствующий элемент в приложении, пользователи могут его просматривать.
     2. Требования:
* Вывод данных из базы;
* Обработка исключений на случай потери соединения с базой данных.
  1. Просмотр винной карты
     1. Описание: см. п. 3.3.1
     2. Требования: см. п. 3.3.2
  2. Бронирование места
     1. Описание: посетители выбирают место в ресторане, вносят предоплату, вносят личные данные и время в приложение. Таким образом место бронируется за конкретными гостями.
     2. Требования:
* Ввод данных;
* Выбор места;
* Выбор времени бронирования;
* Объявление на случай отсутствия свободных мест.
  1. Редактирование меню
     1. Описание: данная функция доступна только поварам, работающим в данном заведении. При добавлении/удалении блюда или напитка оно должно быть удалено и из базы данных, чтобы не вызывать недоразумений.
     2. Требования:
* Ограниченный доступ;
* Редактирование базы данных;
* Подтверждение удаления.
  1. Заказ блюд и напитков
     1. Описание: посетитель имеет возможность сделать заказ через приложение
     2. Требования:
* Ввод личных данных;
* Выбор блюд и напитков;
* Ввод времени;
* Отображение времени готовки.

1. **Требования к данным**
   1. **Логическая модель данных**

Логическая модель данных должна быть представлена визуально в виде диаграммы классов и диаграммы отношений сущностей. Они представлены ниже:

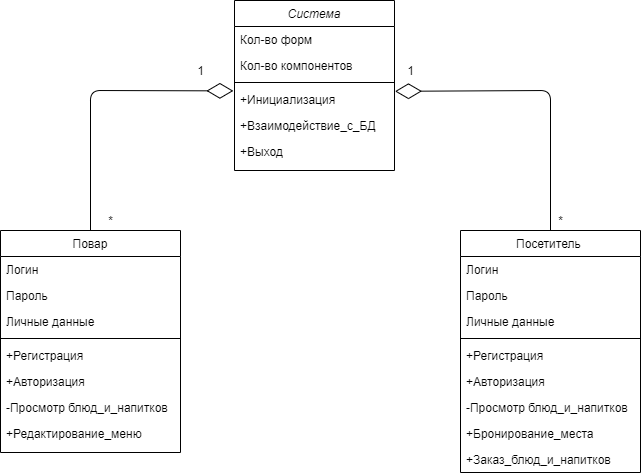


Рисунок 2 – Диаграмма классов

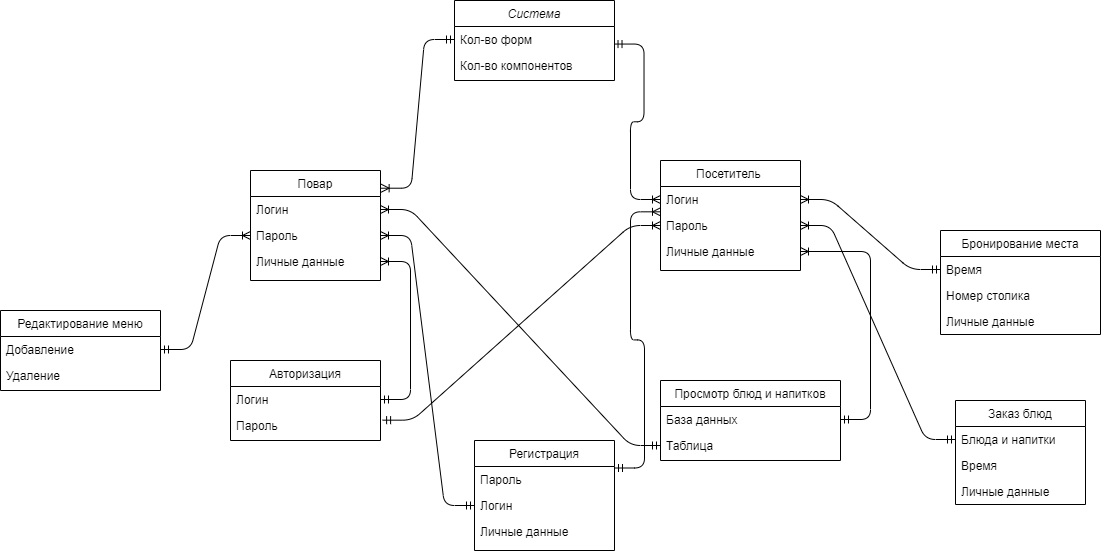


Рисунок 3 – Диаграмма сущностей

1. **Атрибуты качества** 
   1. **Удобство использования**

Требования к удобству использования следующие:

* Легкая изучаемость и понятность интерфейса;
* Эргономичность цветовых и шрифтовых схем;
* Предотвращение критических ошибок и сбоев программы;
* Обработка исключительных ситуаций.
  1. **Производительность**

Каждый модуль программного продукта должен запускаться не более 0.3 сек. Подгрузка баз данных, отправка данных в базу должна занимать не больше 0.7-1 сек.

* 1. **Безопасность**

В первую очередь личные данные пользователей должны быть защищены от несанкционированного доступа к ним. Использовать данные для действий, не описанных в разделе 3 и без ведома пользователей запрещено.

Помимо этого, функция редактирования меню должна быть защищена от доступа обычных пользователей и третьих лиц. Добавлять/изменять/удалять данные из меню могут только сотрудники кухни ресторана и системные администраторы.

* 1. **Техника безопасности**

Чтобы предотвратить утечку данных или полную их потерю, сброс настроек, заказа/брони и пр. не рекомендуется выключать компьютер или иное при работе с приложением.

Для безопасности хранения и использования данных советуется хранить базы данных на серверах с бесперебойными источниками питания.

**Приложение А – словарь терминов**

1. **Git** — система контроля версий, которая позволяет хранить и отслеживать внесённые в файлы изменения. С Git над одним проектом могут работать несколько разработчиков. **Хостинг** - услуга по предоставлению ресурсов для размещения информации на сервере, постоянно имеющем доступ к сети. В данном случае.
2. Системные требования – железо устройства, которое необходимо для запуска программного продукта
3. Сервер — выделенный или специализированный компьютер для выполнения сервисного программного обеспечения (в том числе серверов тех или иных задач).
4. СУБД – система управления базами данных