Министерство образования Республики Беларусь

Учреждение образования Белорусский государственный университет информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Теория разработки программного обеспечения

Техническое задание

к лабораторным работам на тему

«Приложение для управления заказами, сервисом и персоналом отеля»

Выполнил: студент гр. 653503 Хлопцев А.А.

Руководитель: ассистент кафедры информатики Шнейдер. В.В.

Минск 2019

**Оглавление**

[Введение 3](#_Toc20071066)

[1. Постановка задачи 4](#_Toc20071067)

[1.1 Описание предметной области 4](#_Toc20071068)

[1.2 Задачи проектирования 4](#_Toc20071069)

[1.3 Требования к разрабатываемому программному обеспечению 5](#_Toc20071070)

[2. Проектирование приложения 7](#_Toc20071071)

[2.1 Архитектура приложения 7](#_Toc20071072)

[2.2 Используемые технологии 7](#_Toc20071073)

[2.3 Среды реализации задачи 7](#_Toc20071074)

[2.3.1 Microsoft Visual Studio 2017 7](#_Toc20071075)

[2.3.2 .Net Framework 8](#_Toc20071076)

[2.3.3 C# 9](#_Toc20071077)

[2.3.4 XAML 9](#_Toc20071078)

[3. Программная реализация 11](#_Toc20071079)

[3.1 Структура приложения 11](#_Toc20071080)

[4. Описания применения (USE CASE) 13](#_Toc20071081)

[4.1 USER 13](#_Toc20071082)

[4.2 Client 14](#_Toc20071083)

[4.3 Admin 16](#_Toc20071084)

# Введение

Приложение для управления отелем позволяющее упростить контроль над всеми процессами жизни отеля. Быстро и удобно осуществлять кадровые перестановки, добавление нового сервиса и функционала.

С реализация многоуровневого статуса для предоставления полного перечня прав и возможностей для должностного лица.

Пользователь - Воспользовавшись данным статусом, вам будет предоставлена вся информация об отеле: виды комнат, развлечений, услуг, меню ресторана французской кухни. Пользователю будет предоставлена возможность бронирования номера, и получения дополнительной информации об отеле. На выбор гостей представлено 5 номеров, с различными видами услуг.

Гость – статус, который получает человек, зарезервировавший номер в гостинице. Данный тип посетителей получает все привилегии статуса Пользователь + возможность просмотра информации о своём номере, а также использование дополнительных услуг в соответствии с зарезервированным номером.

Администратор – статус работников отеля. Данный вид привилегий позволяет полностью управлять всеми видами услуг отеля, собирать необходимую информацию и статистику, нанимать/увольнять работников, следить за расходами отеля. В дополнении ко всему администраторы утверждают введение-удаление новых услуг и сервиса в гостиницу. Им доступно изменение всех данных о гостях и работниках отеля, а также управление заселением-выселением посетителей в случае непредвиденной ситуации.

Данное приложение избавит от ручного труда, горы документов и бумаг. Сделает более наглядными все основные процессы менеджмента заведения, значительно их ускорит и автоматизирует.

# Постановка задачи

## 1.1 Описание предметной области

* Гости́ница, оте́ль — средство размещения, состоящее из определённого количества номеров, имеющее единое руководство, предоставляющее набор услуг (минимум — заправка постелей, уборка номера и санузла). Гостиницы подразделяются на различные классы в соответствии с предоставляемыми услугами и оборудованием номеров.
* Персонал – специальные люди, нанятые для поддержания функционирования отеля. Каждый член персонала отвечает за свою конкретную область, на которую он был назначен в соответствии с его навыками.
* Клиенты – приобретатель услуг гостиницы. (в виде бронирования номера, заказа столика в ресторане, пользование баром и т.п.)

## 1.2 Задачи проектирования

Чтобы приступить к непосредственному проектированию базы данных, необходимо выделить минимальный набор основных сущностей и указать типы связи между ними.

* Человек (People) – содержит информацию о человеке: ФИО, дату рождения, фотографию и комментарии.
* Гость (Guest) – наследуется от «Человека», дополнительно содержит информацию о комнате, в которую был заселён, дату въезда и выезда из отеля.
* Персонал (Personal) – наследуется от «Человека», дополнительно содержит информацию о зарплате, месте работы и статусе (главный или нет).
* Комната (Room) – содержит информацию о помещении: фотография и этаж расположения.
* Апартаменты (Apartament) – наследуется от «Комната», дополнительно содержит информацию: цена за ночь, номер, кол-во комнат, статус номера, используется ли в данный момент, жилец, список услуг номера.
* Комната развлечений (EntertaimentRoom) – наследуется от «Комната», дополнительно содержит: описание, время работы, рентабельность, рейтинг, стоимость содержания, название, список услуг, перечень персонала, информацию о главном, статус (только для Администратора или нет).

Выделим дополнительные сущности:

* Кухня – наследуется от «Комнаты развлечений»
* Спортивная комната (бассейн и тренажёрный зал) – наследуется от «Комнаты развлечений»

## 1.3 Требования к разрабатываемому программному обеспечению

* Иметь простой и понятный оконный интерфейс (WPF).
* База данных для всего персонала, гостей, видов услуг и заказов.
* Для пользователя – возможности просмотра информации о комнатах развлечений и их услугах, а также возможность заказа номера.
* Для гостя – возможности пользователя + просмотр информации о своём номере, и возможность использования услуг своих апартаментов.
* Для администратора:

1. возможность управления всеми сущностями отеля
2. добавлении/удалении новых работников
3. выселении жильцов
4. изменение данных о всех людях в отеле
5. добавление новых услуг в комнаты
6. просмотр финансовой информации отеля
7. просмотр общей информации об отеле

# Проектирование приложения

## 2.1 Архитектура приложения

На рис. 1 представлена UML-диаграмма архитектуры данного приложения.

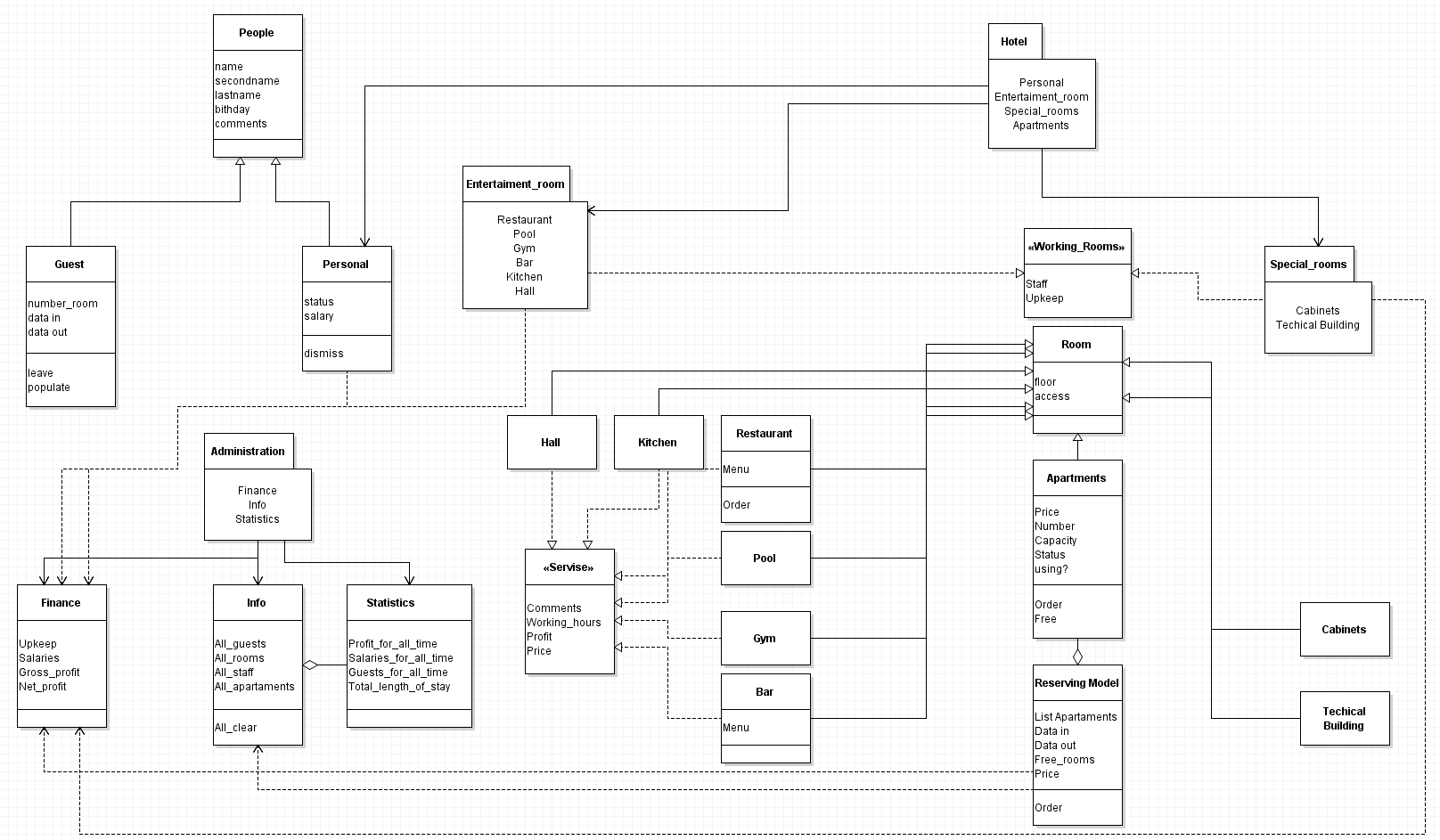


Рис. 1. Архитектура приложения

## 2.2 Используемые технологии

Приложения реализовано на языке C# 7.0 на платформе .Net с использованием языка разметки XAML. В качестве базы данных использовались файлы JSON формата. Среда разработки Microsoft Visual Studio 2017 Community.

## 2.3 Среды реализации задачи

### 2.3.1 Microsoft Visual Studio 2017

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

### 2.3.2 .Net Framework

.NET Framework — программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общеязыковая среда исполнения Common Language Runtime (CLR), которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду.

Считается, что платформа .NET Framework явилась ответом компании Microsoft на набравшую к тому времени большую популярность платформу Java компании Sun Microsystems (ныне принадлежит Oracle).

Хотя .NET является патентованной технологией корпорации Microsoft и официально рассчитана на работу под операционными системами семейства Microsoft Windows, существуют независимые проекты (прежде всего это Mono и Portable.NET), позволяющие запускать программы .NET на некоторых других операционных системах. В настоящее время .NET Framework получает развитие в виде .NET Core, изначально предполагающей кроссплатформенную разработку и эксплуатацию.

### 7.3 C#

C# (произносится си шарп) — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота[4] как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

### XAML

XAML (англ. eXtensible Application Markup Language) — расширяемый язык разметки для приложений (произносится [замл] или [зэмл]) — основанный на XML язык разметки для декларативного программирования приложений, разработанный Microsoft.

Модель приложений Vista включает объект Application. Его набор свойств, методов и событий позволяет объединить веб-документы в связанное приложение. Объект Application контролирует выполнение программы и генерирует события для пользовательского кода. Документы приложения пишутся на XAML. Впрочем, с помощью XAML описывается, прежде всего, пользовательский интерфейс. Логика приложения по-прежнему управляется процедурным кодом (С#, VB, JavaScript и т. д.). XAML может использоваться как для браузер-базированных приложений, так и для настольных приложений.

# Программная реализация

В данном разделе будет рассмотрена архитектура приложения, а также будут приведены примеры некоторых пользовательских функций и хранимых процедур.

## 3.1 Структура приложения

В основе приложения лежит паттерн MVVM(рис. 2)

Паттерн MVVM (Model-View-ViewModel) позволяет отделить логику приложения от визуальной части (представления). Данный паттерн является архитектурным, то есть он задает общую архитектуру приложения.

MVVM состоит из трех компонентов: модели (Model), модели представления (ViewModel) и представления (View).

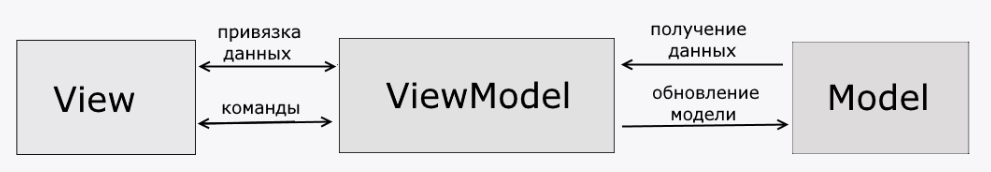


Рис. 2 Паттерн MVVM

**Model**

Модель описывает используемые в приложении данные. Модели могут содержать логику, непосредственно связанную этими данными, например, логику валидации свойств модели. В то же время модель не должна содержать никакой логики, связанной с отображением данных и взаимодействием с визуальными элементами управления.

**View**

View или представление определяет визуальный интерфейс, через который пользователь взаимодействует с приложением. Применительно к WPF представление - это код в xaml, который определяет интерфейс в виде кнопок, текстовых полей и прочих визуальных элементов.

**ViewModel**

ViewModel или модель представления связывает модель и представление через механизм привязки данных. Если в модели изменяются значения свойств, при реализации моделью интерфейса INotifyPropertyChanged автоматически идет изменение отображаемых данных в представлении, хотя напрямую модель и представление не связаны.

ViewModel также содержит логику по получению данных из модели, которые потом передаются в представление. И также VewModel определяет логику по обновлению данных в модели.

Поскольку элементы представления, то есть визуальные компоненты типа кнопок, не используют события, то представление взаимодействует с ViewModel посредством команд.

Итогом применения паттерна MVVM является функциональное разделение приложения на три компонента, которые проще разрабатывать и тестировать, а также в дальнейшем модифицировать и поддерживать.

В качестве примера проиллюстрируем функцию добавления нового клиента.

# Описания применения (USE CASE)

Данное приложение содержит 3 основных окна, для каждого типа пользователей (рис. 3)

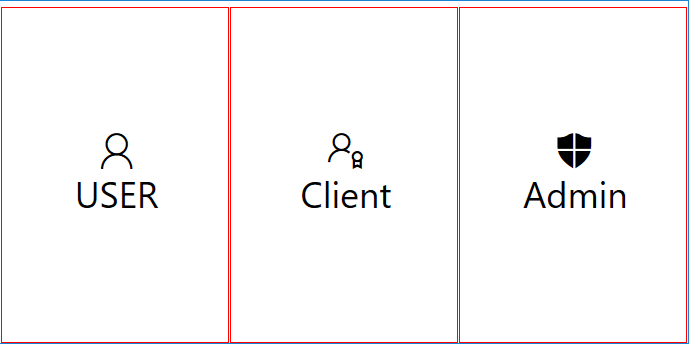


Рис. 3. Окно перехода между пользователями

USER, Clint, Admin – покажем пример работы каждого из них

## 4.1 USER

|  |  |  |
| --- | --- | --- |
| Действие | Шаги | Описание |
| Просмотр информации об отеле | 1. Выбрать тип USER 2. Выбрать интересующую категорию для просмотра 3. Получение информации | Данное действие позволяет простому пользователю просматривать общую информацию об отеле |
| Бронирование | 1. Выбрать тип USER 2. Брать пункт «Бронирование» 3. Заполнить интересующую вас дату 4. Выбрать один из свободных номеров 5. Ввести личные данные 6. Послать запрос на бронирование | Позволяет бронировать выбранную комнату на необходимый промежуток времени |

## 4.2 Client

При нажатии на вкладку клиент мы попадаем в окно выбора действий (рис.6)

В данном окне можно либо перейти к окну заказа (рис. 7), либо посмотреть информацию о своём номере (рис. 8)

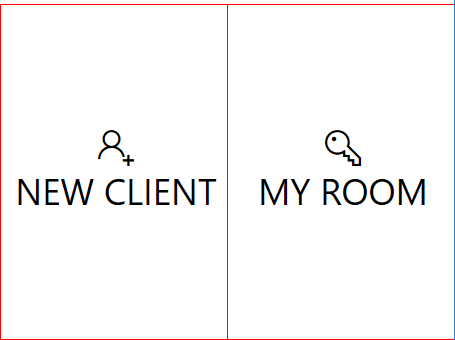


Рис. 6. Окно выбора клиента

Клиенту доступны все действия, что и пользователю + некоторые новые

|  |  |  |
| --- | --- | --- |
| Действие | Шаги | Описание |
| Просмотр информации об свое комнате | 1. Выбрать тип Client 2. Выбрать вкладку MyRoom 3. Ввести свою личные данные 4. Получить информацию о своей комнате и доступных в ней услугах | Данное действие позволяет постояльцу просматривать информацию о своём номере, видах предоставляемых ему услуг и списку заказов |
| Заказ | 1. Выбрать тип Client 2. Выбрать вкладку MyRoom 3. Ввести свою личные данные 4. Нажать на вкладку услуги 5. Выбрать одну из предоставляемых услуг 6. Выбрать удобное время 7. Нажать на кнопку «Заказать» | Позволяет заказывать услуги, предоставляемые отелем |

## Admin

При нажатии на вкладку Admin и успешного выполнения аутентификации администратора (рис. 9), мы попадаем в окно администрирования.

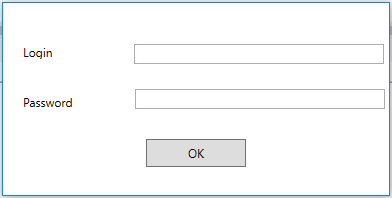


Рис. 9. Окно аутентификации администратора

|  |  |  |
| --- | --- | --- |
| Действие | Шаги | Описание |
| Просмотр общей информации об отеле | 1. Выбрать тип Admin 2. Введение логина и пароля 3. Переход на вкладу Info 4. Получение общей информации об отеле |  |
| Просмотр финансовой информации об отеле | 1. Выбрать тип Admin 2. Введение логина и пароля 3. Переход на вкладу Finance 4. Получение общей информации об отеле |  |
| Добавление новых услуг | 1. Выбрать тип Admin 2. Введение логина и пароля 3. Переход на вкладу Info 4. Выбор интересующей категории услуги 5. Выбрать пункт «Добавить услугу» 6. Введение Всей необходимой информации о новой услуге 7. Нажать клавишу «Сохранить» |  |
| Просмотр и изменение информации о персонале | 1. Выбрать тип Admin 2. Введение логина и пароля 3. Переход на вкладу Personal 4. Выбор необходимого человека 5. Выбор пункта «Изменить» 6. Введение/верификация новых данных 7. Нажать клавишу «Сохранить» |  |