Hotels Booking System

*Описание проекта:*

WPF приложение, позволяющее авторизованному пользователю бронировать номер в отеле и смотреть отчет о платежах.

После регистрации и входа открывается страница бронирования, на которой нужно выбрать отель, даты заезда и выезда и тип номера (family, single, lux, presidential). Далее после подтверждения введенных данных, пользователь переходит на страницу оплаты и вводит данные карты. Все данные о пользователе, кредитной карте, оплате и информации о бронировании заполняются в базе данных с использованием Entity Framework. На странице истории платежей с помощью LINQ выводятся все данные о покупке (кредитная карта, платеж).

*Репозиторий:*

<https://github.com/Konstantinov10>

*Члены команды и их роли:*

Константинов Иван 164 - Main logic, database design

Цваева Ася 164 - UI, main logic

Данилочева Мария 164 – UI, main logic

*Классы:*

* Model:

1) Customer.cs

Класс, содержащий информацию о пользователях.

2) Hotel.cs

Класс, содержащий информацию об отелях.

3) Room.cs

Класс, содержащий информацию о комнатах в отеле.

4) CreditCard.cs

Класс, содержащий информацию о кредитных картах.

5) Payment.cs

Класс, содержащий информацию о платежах.

6) ReservationInfo.cs

Класс, содержащий информацию о данных бронирования.

* Context.cs

Класс, отвечающий за создание баз данных.

* Repository:

1) CustomerRepository.cs

Класс, связывающий Customer с UI и DataBase. Содержит методы, возвращающие таблицы в формате листа из базы данных Customers, CreditCards, Payments. Метод PaymentHistory, содержащий LINQ-запрос Inner Join двух таблиц CreditCards и Payments. Метод AddCustomer, добавляющий пользователя в базу данных в таблицу Customers. Метод CheckLogPass, проверяющий совпадение введенного пользователем логина с уже имеющимися в базе данных. Методы AccountFirstName, AccountLastName, AccountEmail, сравнивающие введенный логин с логином в базе данных и при совпадении возвращающие одно значение Customer.FirstName, Customer.LastName, Customer.Email соответственно.

2) HotelRepository.cs

Класс, связывающий Hotel с UI и DataBase.Содержит метод, возвращающий таблицу формате листа из базы данных Hotels.

3) Repository.cs

Класс, связывающий ReservationInfo с UI и DataBase.Содержит метод, возвращающий таблицу формате листа из базы данных ResrvationInfos. Метод AddResrvationInfo, добавляющий данные о бронировании в таблицу ReservationInfos.

4) CreditCardRepository.cs

Класс, связывающий CreditCard с UI и DataBase.Содержит метод, возвращающий таблицу формате листа из базы данных CreditCards. Метод AddCreditCard, добавляющий данные о бронировании в таблицу CreditCards.

5) PaymentRepository.cs

Класс, связывающий Payment с UI и DataBase.Содержит метод, возвращающий таблицу формате листа из базы данных Payments. Метод AddPayment, добавляющий данные о бронировании в таблицу Payments.

* Pages:

1) MainPage.xaml.cs

Класс авторизации, где прописано событие перехода на страницу регистрации или на страницу бронирования при совпадении введенных логина и пароля с уже существующими в базе данных. Прописаны события анимации кнопок.

2) RegistrationPage.xaml.cs

Класс регистрации нового пользователя, где происходит запись новых данных в базу данных при нажатии на кнопку и переход назад на страницу входа. Прописаны события анимации кнопок.

3) BookingPage.xaml.cs

Класс бронирования, содержащий метод очистки ListBox. Описывает события при нажатия на кнопки, а именно добавление отеля, даты и выбранного типа номера в ListBox, подсчета суммы платежа, очистку страницы и переход на страницу оплаты или на страницу истории платежей. Прописаны события анимации кнопок.

4) CreditCardPage.xaml.cs

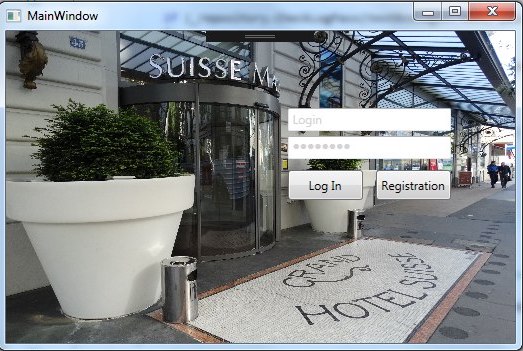
Класс оплаты, добавляющий данные о карте в базу данных при нажатии на кнопку и осуществляющий переход назад к бронированию. Прописаны события анимации кнопок.

5) AccountPage.xaml.cs

Класс историй платежа, содержащий событие загрузки DataGrid, при котором выводятся имя, фамилия, почта зарегистрированного пользователя и информация о совершенных платежах.

*Интерфейс:*

1) MainPage.xaml



Страница авторизации. Пользователь может пройти на страницу регистрации, нажав на кнопку “Registration” или, если уже зарегистрирован, ввести логин и пароль и войти, перейдя на страницу бронирования.

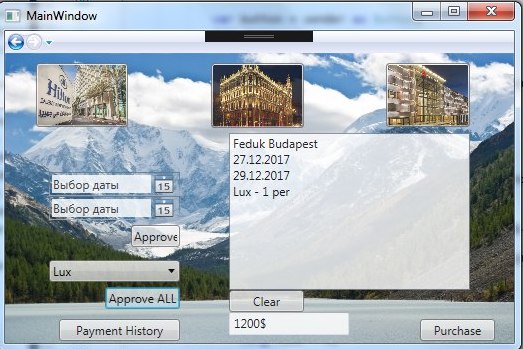
2) RegistrationPage.xaml



Страница регистрации. Пользователь должен ввести свои данные и зарегистрироваться, перейдя обратно на страницу авторизации.

3) BookingPage.xaml





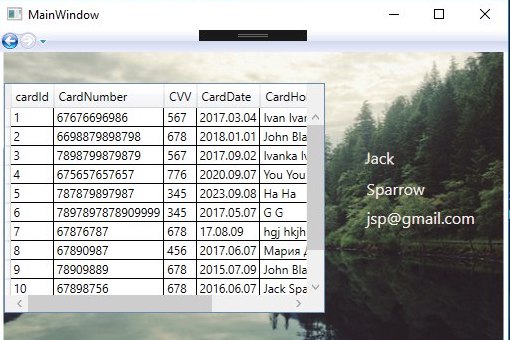
Страница бронирования. Пользователю нужно сначала выбрать отель, нажав на его иконку, далее выбрать даты заезда и выезда с помощью календаря и нажать на кнопку подтверждения, которая станет доступна только после выбора отеля, затем из ComboBox, который станет доступен после подтверждения даты, выбрать тип номера и подтвердить свой выбор, нажав на кнопку, которая будет доступна лишь после выбора всей информации. После чего ему станет доступна информация о сумме платежа и кнопка перехода к странице платежа. Пользователь может очистить все данные, нажав на кнопку “Clear”. Он также может перейти на страницу Payment History, нажав на кнопку.

4) CreditCard.xaml



Страница оплаты. Пользователь должен ввести данные и нажать на кнопку “Purchase”, после чего данные вносятся в базу данных, и пользователь возвращается на страницу бронирования.

5) AccountPage.xaml



Страница истории платежей. Страница для просмотра истории платежей, совершенных пользователями в приложении.

Test case:

1. Открыли приложение и ввели несуществующий логин и пароль, после чего появился MessageBox “Incorrect login or password”.
2. Переходим на страницу регистрации, вводим личные данные и нажимаем на кнопку “OK” и переходим обратно к авторизации, данные записываются в базу данных.
3. Вводим зарегистрированный логин и пароль и переходим к бронированию.
4. Кнопки подтверждения даты и подтверждения информации в целом - недоступны, так что мы нажимаем на понравившийся отель, его имя высвечивается в ListBox.
5. Становится активна кнопка подтверждения даты “Approve”, но если мы нажимаем на нее, не выбрав обе даты или одну из них, появляется MessageBox “Please, enter two dates!”, выбрав обе даты, мы нажимаем кнопку.
6. После этого активируется кнопка “Approve ALL”, нажав на нее, появляется TextBox с суммой, которую мы должны оплатить, также появляется кнопка “Purchase”, нажимаем на нее и переходим на страницу оплаты.
7. Чтобы совершить покупку, мы должны ввести все данные корректно. Если мы вводим не 8-ми значный номер карты, не 3-х значный CVV или дату карты в неверном формате (xxxx.xx.xx), появляется MessageBox с ошибкой и нам предлагается заменить некорректные данные. После ввода и нажатия на кнопку “Purchase” данные заполняются в базу данных, и мы переходим обратно на страницу бронирования, где вся информация уже очищена.
8. Нажимаем на кнопку “Payment History” и переходим на другую страницу, где видна наша личная информация (имя, фамилия, почта) и таблица с информацией о всех совершенных платежах (кредитная карта+платеж).