**МИНОБРНАУКИ РОССИИ**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ОТЧЕТ**

**по учебной практике УП.02.01 (по профилю специальности)**

по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студент(ка) 4 курса 42919/7 группы

Гневанова Анна Алексеевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «28» октября 2024 г. по «09» ноября 2024 г.

Руководитель практики Хисамутдинова А.С.

(подпись) (расшифровка подписи)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2024

**ЗАДАНИЕ**

**на учебную практику (по профилю специальности)**

по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студент(ка) 4 курса 42919/7 группы

Гневанова Анна Алексеевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «28» октября 2024 г. по «09» ноября 2024 г.

**Виды работ, обязательные для выполнения** *(переносится из программы, соответствующего ПМ):*

* Участие в выработке требований к программному обеспечению;
* Стадии проектирования программного обеспечения. Проектирование UML-диаграмм;
* Конструирование пользовательского интерфейса. Разработка модулей программного обеспечения;
* Разработка рабочего проекта и технологической документации.

**Индивидуальное задание: ВАРИАНТ 2**

Задание выдал «28» октября 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_ Хисамутдинова А.С.

(подпись) (Ф.И.О.)

Задание получил «28» октября 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_ Гневанова А. А. (подпись) (Ф.И.О.)

**МИНОБРНАУКИ РОССИИ**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ДНЕВНИК**

**прохождения учебной практики УП.02.01**

**(по профилю специальности)**

по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей»

(код и наименование)

Специальность09.02.07 Информационные системы и программирование

(код и наименование специальности)

Студент(ка) 4 курса 42919/7 группы

Гневанова Анна Алексеевна

(Фамилия, имя, отчество)

Место прохождения практики: УВЦ, пр. Энгельса, 23

(наименование и адрес организации)

Период прохождения практики

с «28» октября 2024 г. по «09» ноября 2024 г.

Руководитель практики Хисамутдинова А.С.

(подпись) (расшифровка подписи)

Итоговая оценка по практике

Санкт-Петербург

2024

**Содержание дневника**

|  |  |  |
| --- | --- | --- |
| **Дата** | **Виды выполненных работ и заданий по программе практики** | **Подпись руководителя практики** |
| **1** | **2** | **3** |
| 28.10.2024 | Анализ предметной области. Анализ методов решения. Анализ и выбор средств. |  |
| 29.10.2024 | Анализ и выбор средств. Исследовательские работы. Разработка технического задания. |  |
| 30.10.2024 | UML. Проектирование диаграммы вариантов использования и последовательности. UML. Проектирование диаграммы вариантов активности и состояния. |  |
| 31.10.2024 | Моделирование структуры ПО. Проектирование инфологической и даталогической модели данных. |  |
| 01.11.2024 | Проектирование интерфейса пользователя. |  |
| 02.11.2024 | Разработка схемы алгоритма программного продукта. |  |
| 05.11.2024 | Разработка модели данных. Разработка программного продукта. Создание функциональности ПО. |  |
| 06.11.2024 | Модульное тестирование. Создание тестовых случаев. |  |
| 07.11.2024 | Создание тестовых случаев. Отладка программных модулей. |  |
| 08.11.2024 | Разработка документа «Руководство пользователя» в соответствии с ЕСПД. Разработка документа «Программа и методика испытаний» в соответствии с ЕСПД. |  |
| 09.11.2024 | Подготовка отчета. |  |

# СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

Целью данной работы является создание настольного приложения для базы отдыха с использованием Windows Forms. Приложение позволяет пользователям регистрироваться, бронировать номера и заказывать дополнительные услуги, а сотрудникам — управлять данными о номерах, статусах бронирований и услугах. Особое внимание уделено разработке простого и интуитивно понятного интерфейса, а также интеграции с базой данных для хранения информации о пользователях и бронированиях.

По итогам работы было создано рабочее приложение, которое предоставляет базовые функции для пользователей и сотрудников базы отдыха. Внедренная база данных позволяет эффективно управлять данными и служит основой для дальнейшего расширения функционала системы.

# Задание 1. Техническое задание.

Учитывая предметную область, было написано техническое задание, для разработки программного модуля.

Техническое задание

1. Общие сведения

1.1. Наименование проекта: Разработка программного модуля управления бронированиями и услугами базы отдыха.

1.2. Заказчик: база отдыха «sunset dreams»

1.3. Исполнитель: «Aurora Apps»

2. Функциональные требования

2.1 Добавление бронирований в базу данных с указанием следующих параметров:

- Номер бронирования;

- Номер клиента;

- Бронируемый номер;

- Даты выезда и въезда;

- Статус бронирования (предварительная бронь, подтверждённая, завершённая, аннулированная)

2.2 Редактирование бронирований:

- Изменение бронируемого номера

- Изменение даты въезда и выезда

- Изменение статуса бронирования

2.3 Авторизация пользователей:

- Авторизация предполагает несколько уровней доступа

- Администратор базы отдыха: имеет полный доступ к системе, может управлять всеми сущностями и просматривать отчеты;

- Сотрудник: ограниченные права, возможность управления номерами, услугами и бронированиями;

- Клиент: регистрируется в системе, может искать жилье, бронировать номера, просматривать доступные услуги и мероприятия, оформлять заказы на питание;

2.4 Добавление в базу данных информации о новых видах оказываемых услуг и мероприятиях:

- При добавлении новых услуг должна быть возможность добавить их в базу данных для дальнейшей работы с ними

2.5 Составление отчетов

- Администратор может составлять финансовые отчеты, содержащие информацию о выручке, использованных услугах и бронированиях

3. Нефункциональные требования

3.1. Кроссплатформенность:

- Поддержка работы на ОС семейства Windows.

3.2. Безопасность:

- Логин и пароль для доступа к приложению;

- Доступ к данным должен быть ограничен в зависимости от роли пользователя.

3.3. Удобство использования:

- Простой и интуитивный интерфейс;

- Информативные уведомления и подсказки.

3.4. Производительность:

- Приложение должно иметь быстрый доступ к данным;

- Минимальное время отклика на запросы пользователя.

4. Требования к реализации

4.1. Язык программирования: C#

4.2. СУБД: MS SQL Server

5. Требования к документации

5.1. Техническое задание на разработку программного модуля.

6. Руководство по стилю

6.1. Шрифт: Roboto, размер шрифта для основного текста - 14px, для заголовков - 18px.

6.2. Цветовая схема: для текста используется цвет HEX#606C38, для заголовков – HEX#283618. Для фона используется HEX#FEFAE0, а для обозначения контрастных элементов (так же могут использоваться и для фона) цвета HEX#DDA15E и HEX#BC6C25

# Задание 2. Разработка алгоритмов и диаграмм

Задание включает в себя разработку алгоритмов и диаграмм, необходимых для проектирования системы на основе составленного технического задания. В рамках задания требуется создать следующие диаграммы и схемы: диаграммы вариантов использования, диаграммы последовательности, диаграммы активности (Рисунки 1-3).

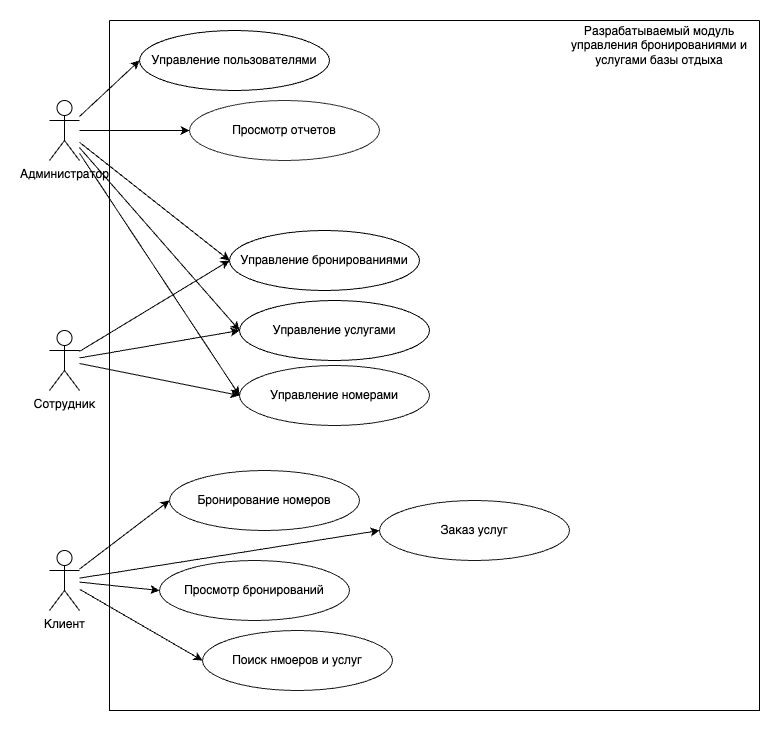


Рисунок 1 - Диаграмма прецедентов модуля

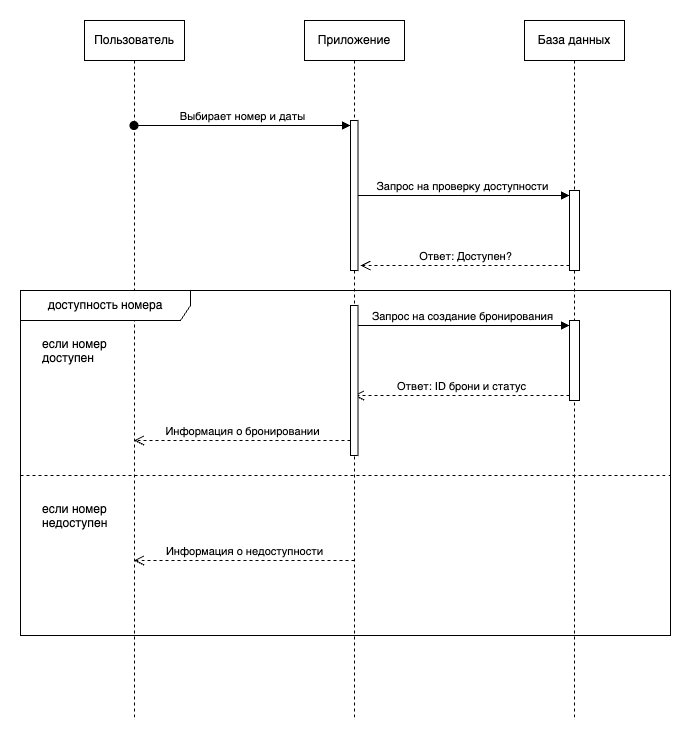


Рисунок 2 – Диаграмма последовательности бронирования номера клиентом

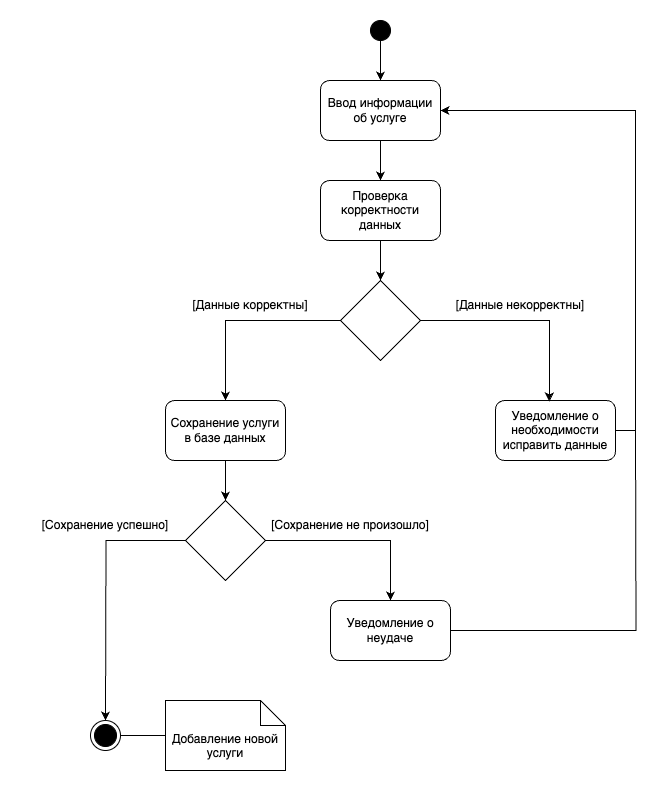


Рисунок 3 – Диаграмма активностей добавления новой услуги

Необходимо разработать ER-диаграмму (диаграмму сущностей и их взаимосвязей) для базы данных (Рисунок 4). Важным моментом является приведение базы данных к третей нормальной форме (3НФ), что позволяет минимизировать избыточность данных и обеспечить ссылочную целостность между сущностями.

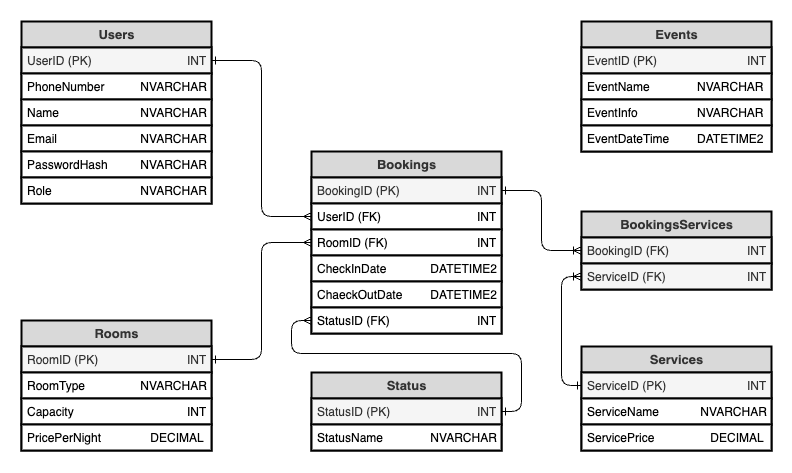


Рисунок 4 - ER-диаграмма базы данных

Также важно создать словарь данных, который будет содержать описание всех сущностей и атрибутов, их типов и связей, что позволит унифицировать и стандартизировать структуру базы данных (Таблица А1). После составления схем и словаря для БД, составляется скрипт на создание всех таблиц (Приложение А2), база заполняется начальными данными для работы (Рисунки 5-11).

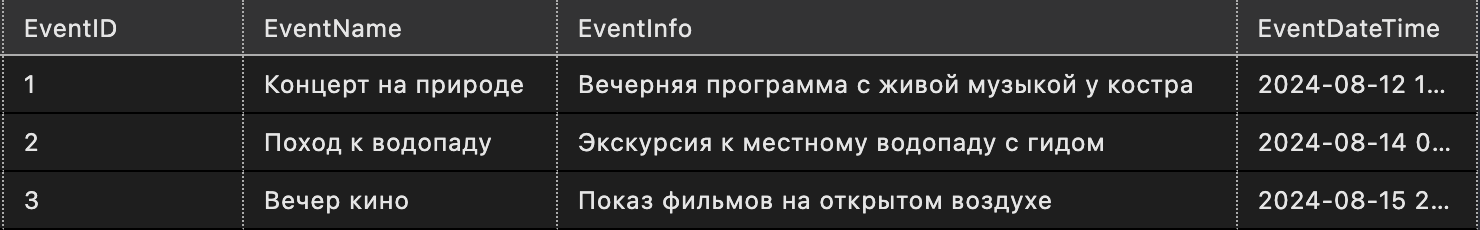


Рисунок 5 - Заполнение таблицы Events

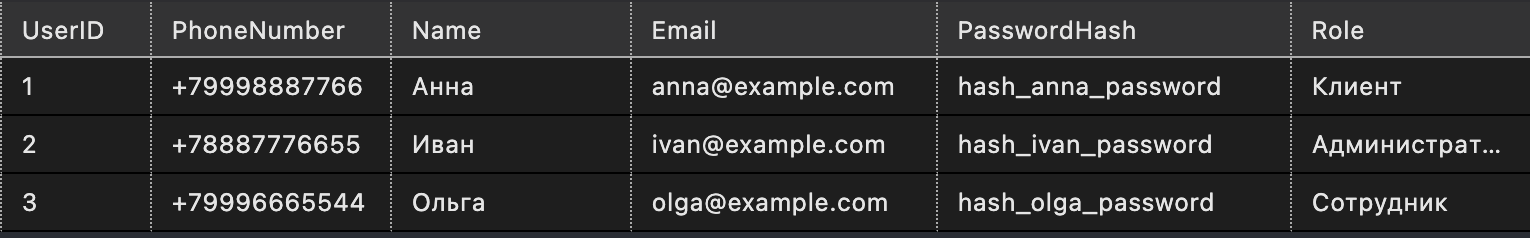


Рисунок 6 - Заполнение таблицы Users



Рисунок 7 - Заполнение таблицы Rooms

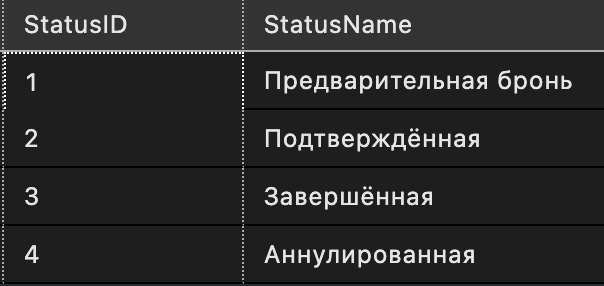


Рисунок 8 - Заполнение таблицы Status

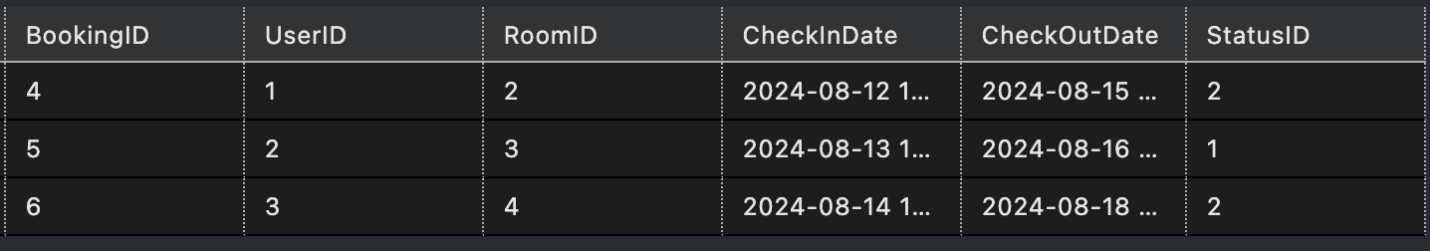


Рисунок 9 - Заполнение таблицы Bookings

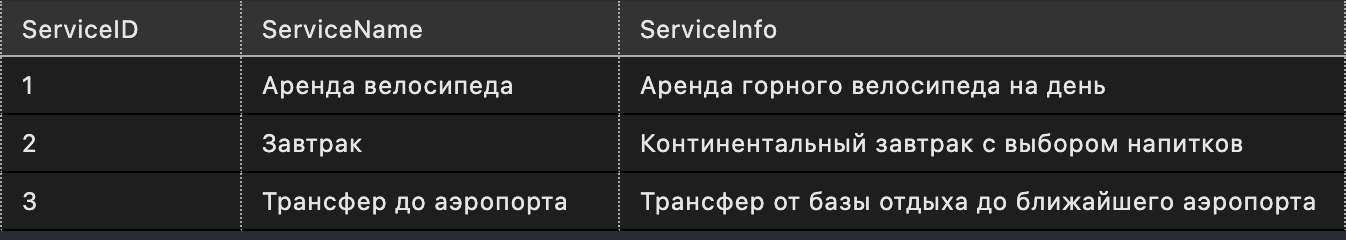


Рисунок 10 - Заполнение таблицы Services

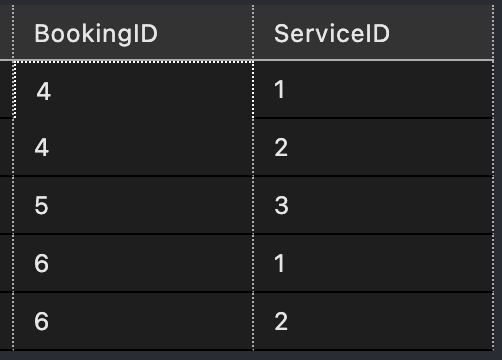


Рисунок 11 - Заполнение таблицы BookingServices

# Задание 3. Разработка макетов программы

В процессе разработки важно учесть функциональные и визуальные предпочтения пользователей, чтобы создать удобный и интуитивно понятный интерфейс. В качестве первого шага следует создать Wireframe — черно-белые эскизы, которые помогут определить структуру страниц, расположение основных элементов интерфейса (Рисунки 12-16).

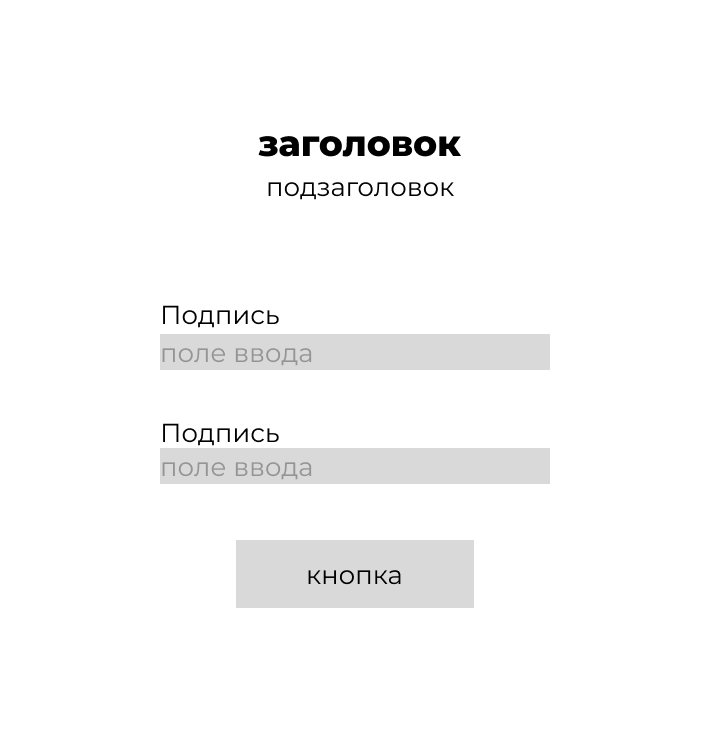


Рисунок 12 - Wireframe формы входа

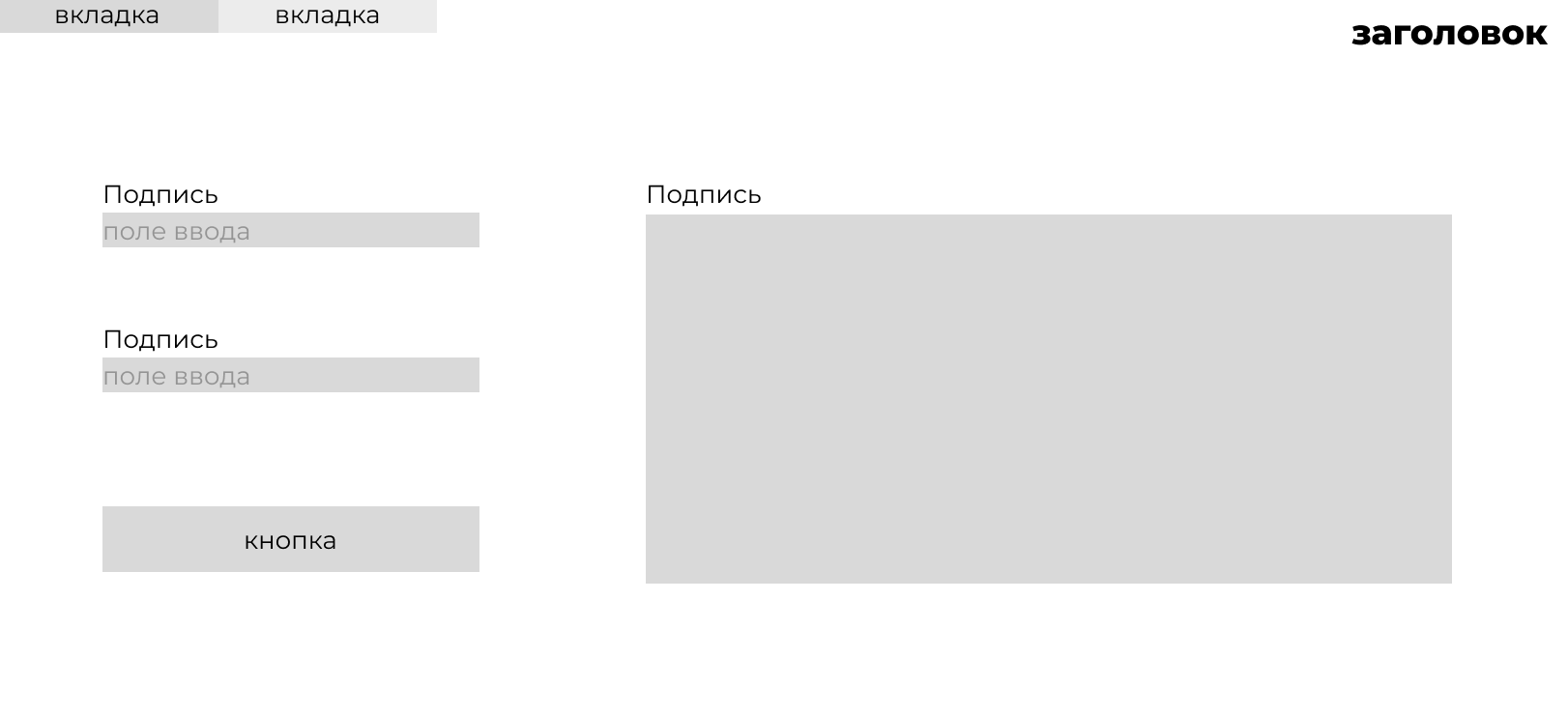


Рисунок 13 - Wireframe формы бронирования домика



Рисунок 14 – Wireframe просмотра бронирований клиента



Рисунок 15 - Wireframe формы работника



Рисунок 16 - Wireframe формы администратора

После утверждения структуры и расположения элементов интерфейса, можно переходить к созданию Mockup — цветных макетов, которые будут более детализированными и визуально привлекательными (Рисунки 17-21). Эти макеты отображают внешний вид интерфейса с учетом всех графических и визуальных элементов, таких как цветовая схема, шрифты и изображения. Для выполнения работы был использован графический редактор Figma, который предоставляет все необходимые инструменты для создания и редактирования макетов.

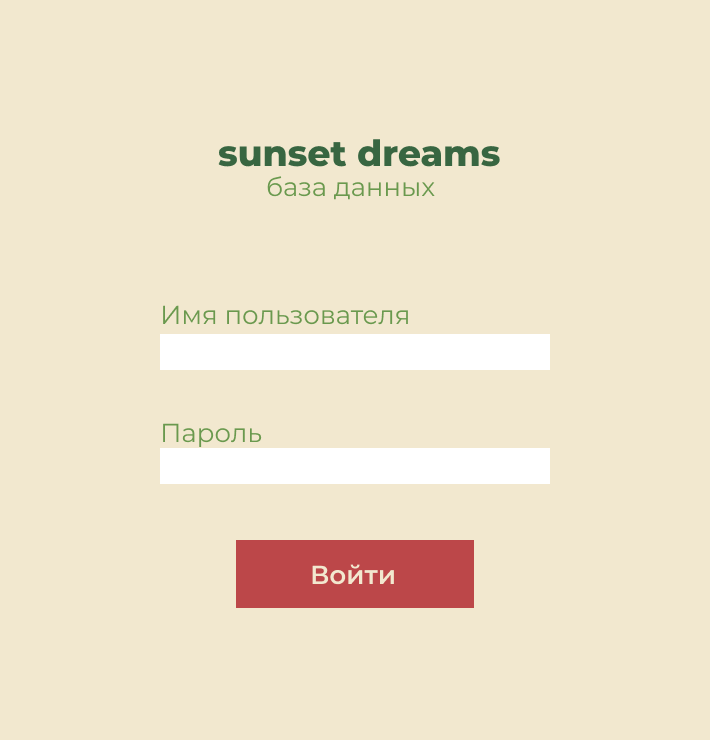


Рисунок 17 - Mockup формы входа

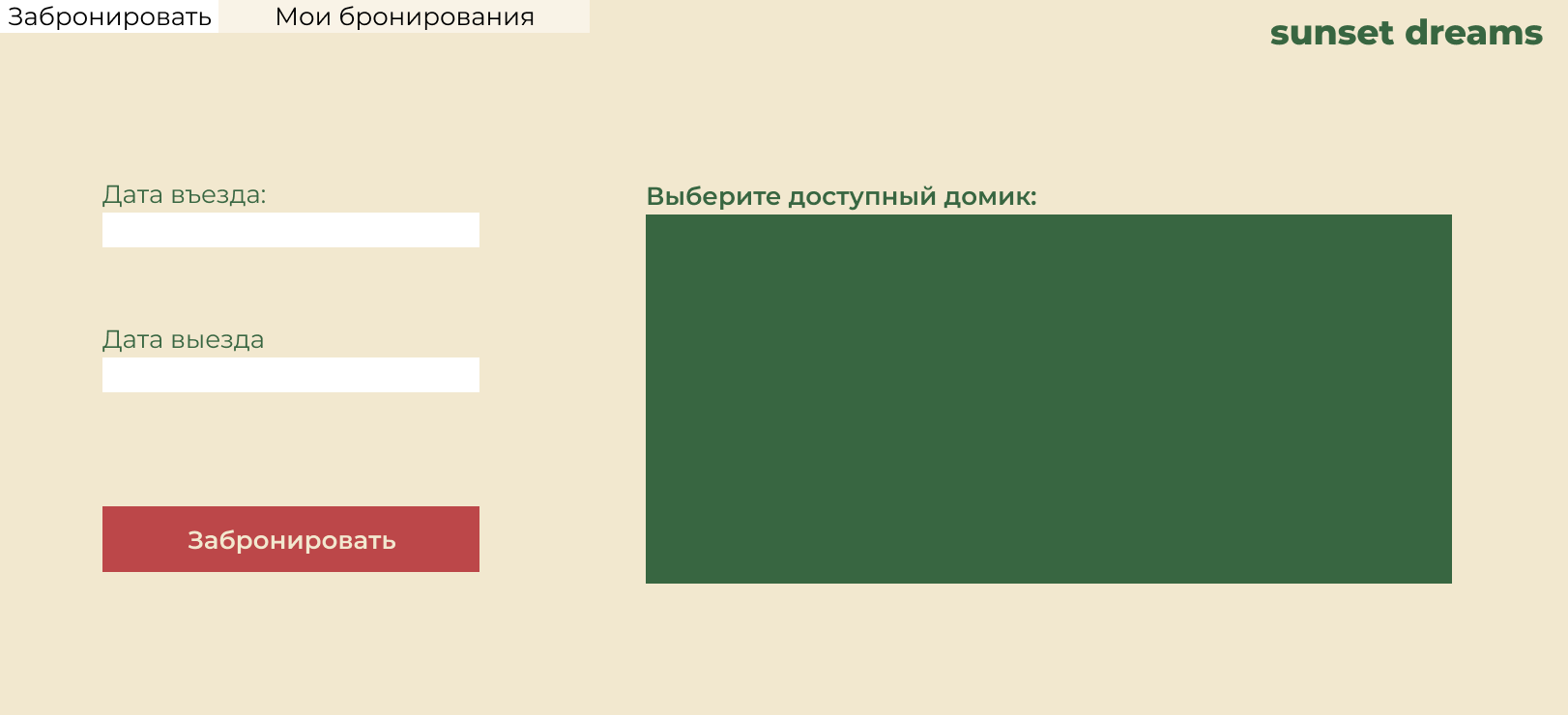


Рисунок 18 - Mockup формы бронирования домика

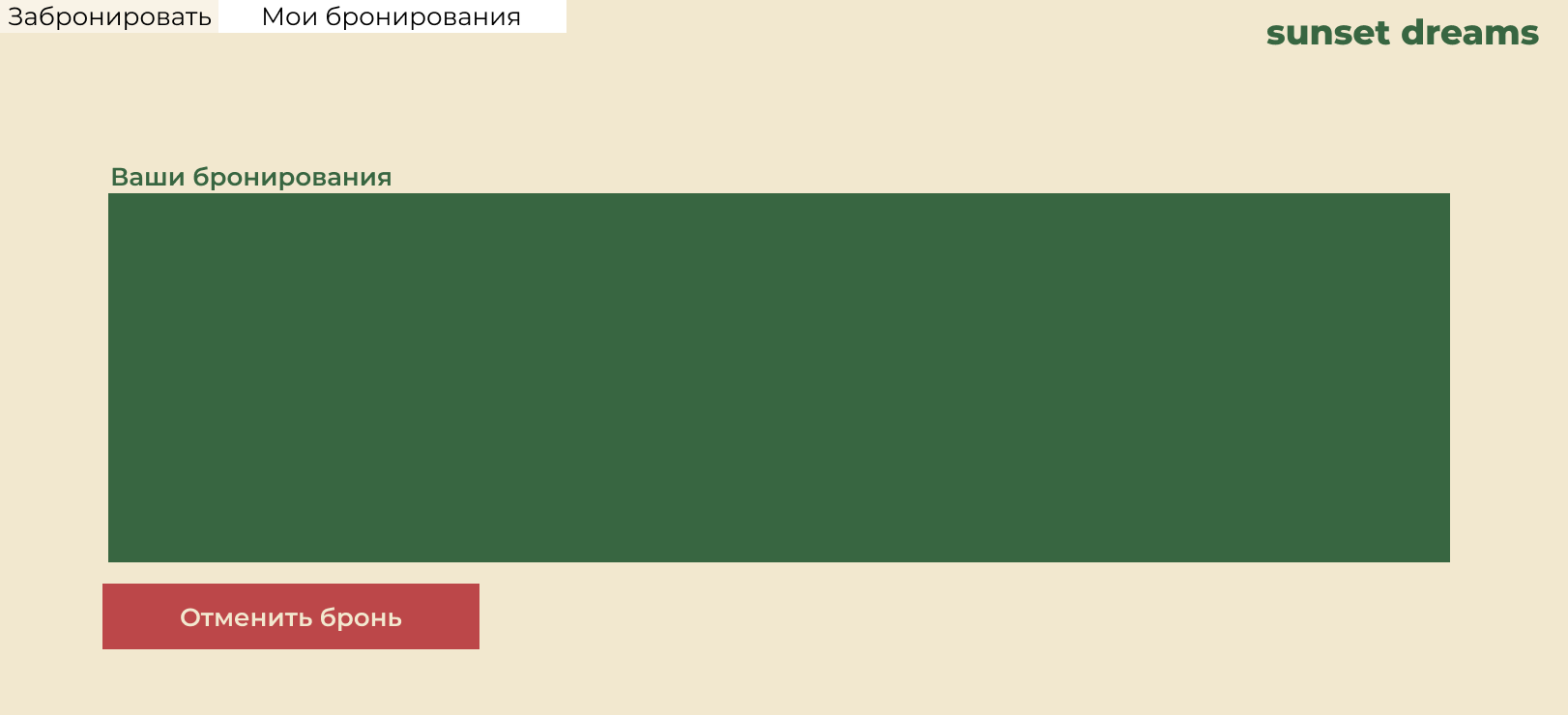


Рисунок 19 - Mockup просмотра бронирований клиента



Рисунок 20 - Mockup формы работника

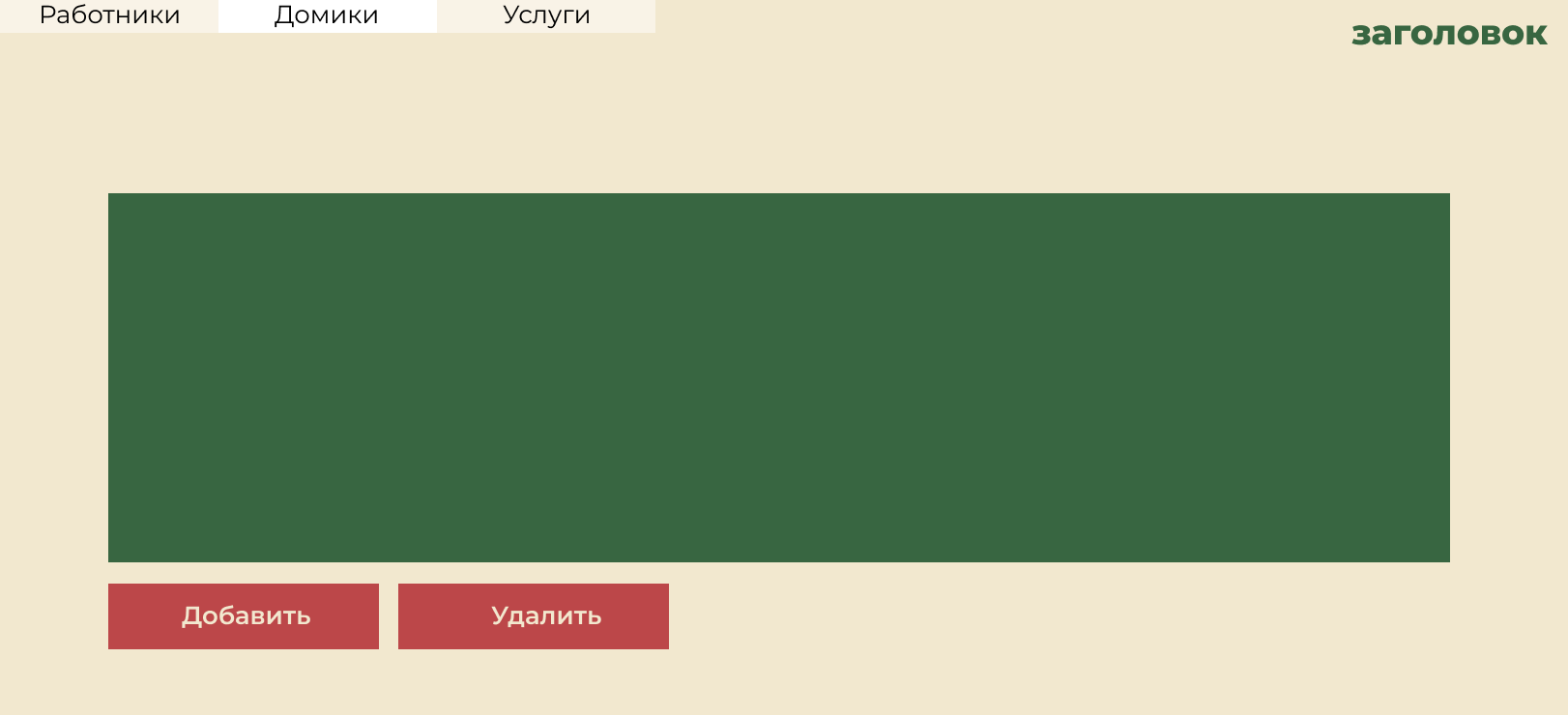


Рисунок 21 - Mockup формы администратора

Кроме того, для разрабатываемого приложения создается его карта, для понимания его работы (Рисунок 22).

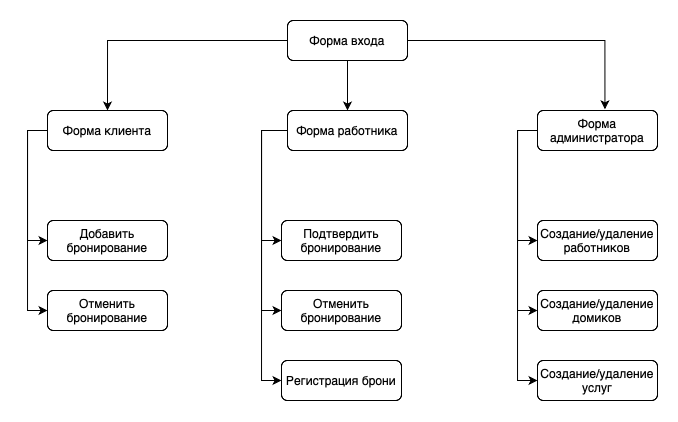


Рисунок 22 – Схема навигации приложения

# Задание 4. Разработка программы

Опираясь на проведенный ранее этап проектирования, был разработан программный модуль на языке программирования С# на технологии Windows Forms.

При входе в приложение с данными пользователя с ролью «Клиент» можно увидеть форму бронирований и забронировать домик, а также увидеть бронирования пользователя, там же пользователь может отменить свою бронь (Рисунки 24-25).

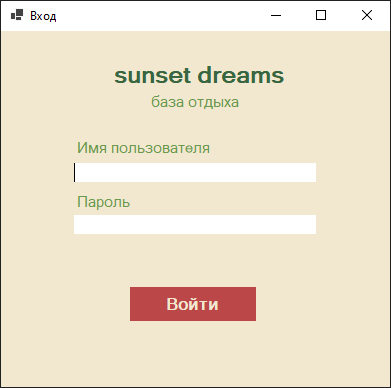


Рисунок 23 — Форма входа в приложение

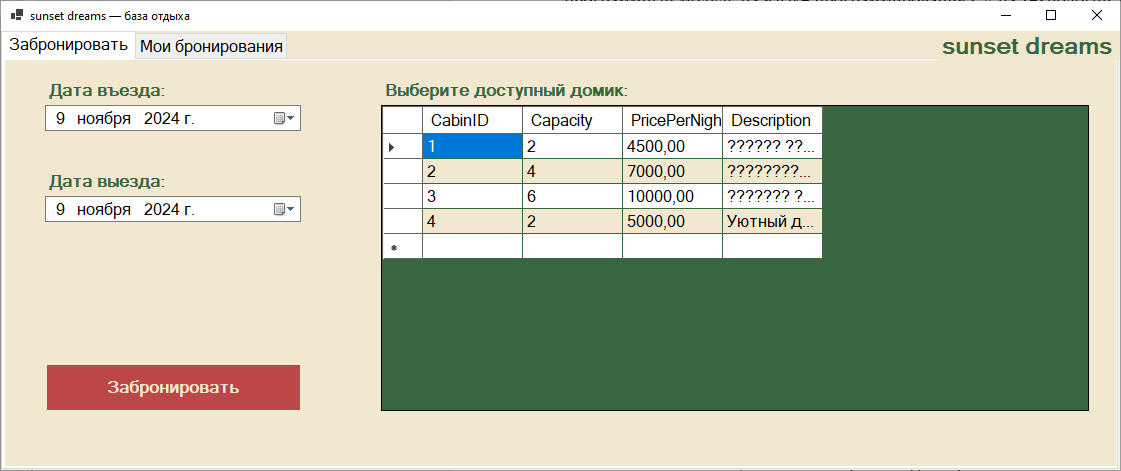


Рисунок 24 — Форма бронирований клиента

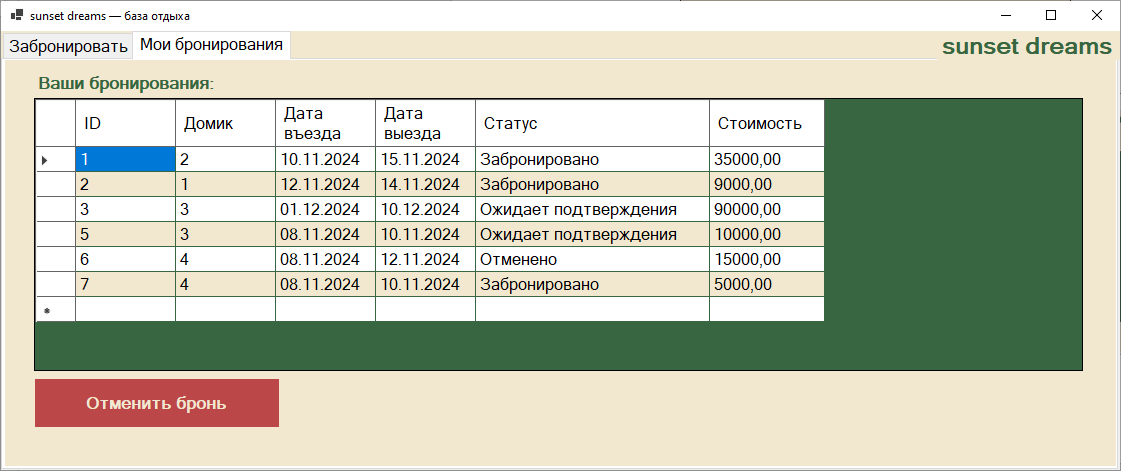


Рисунок 25 — Форма просмотра и отмены бронирований клиента

Форма пользователя с ролью «Работник» позволяет просматривать бронирования всех клиентов, подтверждать их, отменять и регистрировать (Рисунки 26-27).

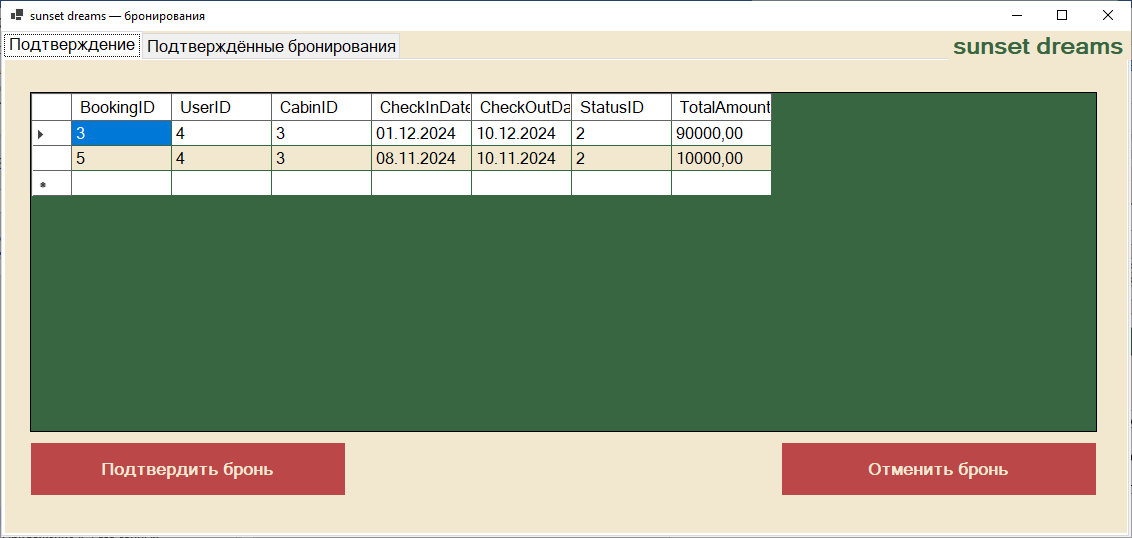


Рисунок 26 — Форма подтверждения и отмены бронирований

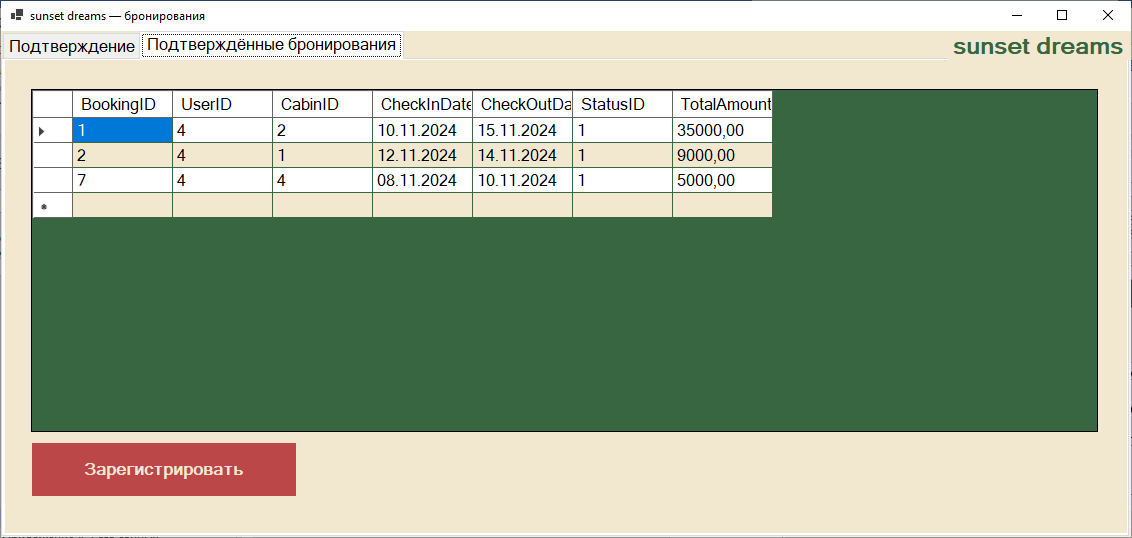


Рисунок 27 — Форма регистрации бронирований

Форма пользователя с ролью «Администратор» позволяет добавлять записи и удалять из таблиц работников, домиков и услуг (Рисунок 28).

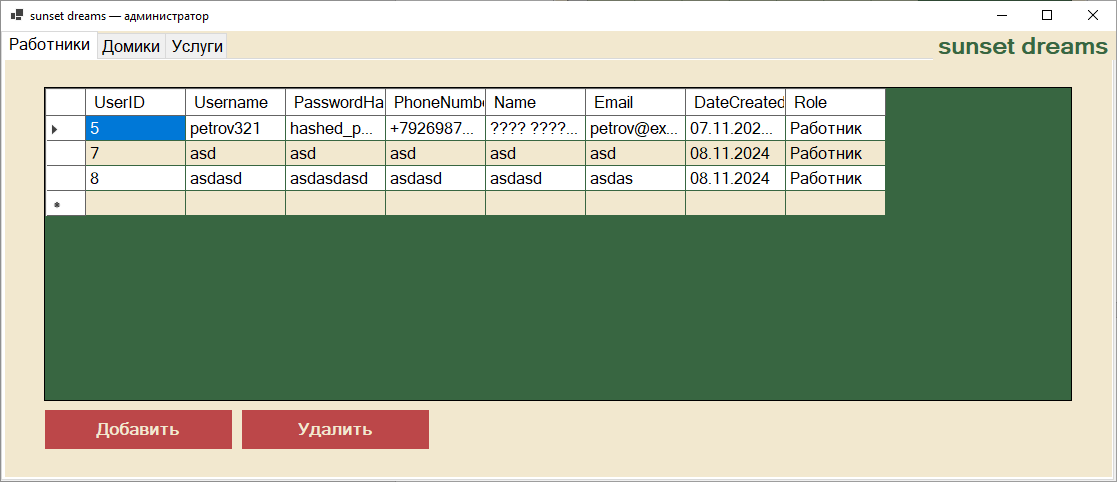


Рисунок 28 — Форма администратора

# Задание 5. Тестирование

Для программы были подготовлены Test-Case, проверяющие процессы разработанного модуля (Таблицы 1-6).

Таблица 1 — Аннотация теста

|  |  |
| --- | --- |
| **Название проекта** | Программный модуль базы отдыха |
| **Рабочая версия** | 1.0.0 |
| **Имя тестирующего** | Анна Гневанова |
| **Дата(ы) теста** | 06.11.2024 |

Таблица 2 — Тестовый пример 1. Авторизация

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_Autorization\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Авторизация приложения с верными данными |
| **Краткое изложение теста** | Проверка правильности авторизации в приложении с верными данными |
| **Этапы теста** | 1. Ввести правильные данные, соответствующие роли «Клиент» 2. Нажать на кнопку «Войти» |
| **Тестовые данные** | Имя пользователя: ivanov123  Пароль: hashed\_pasword\_1 |
| **Ожидаемый результат** | Открывается форма пользователя с ролью «Клиент» |
| **Фактический результат** | Открывается форма пользователя с ролью «Клиент» |
| **Статус** | Зачет |
| **Предварительное условие** | Приложение должно быть открыто |
| **Постусловие** | Открыта форма пользователя с ролью «Клиент» |
| **Примечания/комментарии** | — |

Таблица 3 — Тестовый пример 2. Авторизация

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_Autorization\_2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Авторизация приложения с неверными данными |
| **Краткое изложение теста** | Проверка правильности авторизации в приложении с неверными данными |
| **Этапы теста** | 1. Ввести правильные данные, соответствующие роли «Клиент» 2. Нажать на кнопку «Войти» |
| **Тестовые данные** | Имя пользователя: wrongname  Пароль: wrongpassword |
| **Ожидаемый результат** | Авторизация не происходит, предупреждение о том, что пользователя не существует. |
| **Фактический результат** | Авторизация не происходит, предупреждение о том, что пользователя не существует. |
| **Статус** | Зачет |
| **Предварительное условие** | Приложение должно быть открыто |
| **Постусловие** | Вход не произошел, форма входа остается открытой. |
| **Примечания/комментарии** | — |

Таблица 4 — Тестовый пример 3. Добавление бронирования

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_Booking\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Попытка забронировать домик |
| **Краткое изложение теста** | Проверка возможности добавления бронирования пользователем |
| **Этапы теста** | 1. Зайти в приложение под пользователя с ролью «Клиент» 2. Выбрать даты бронирования 3. Нажать кнопку забронировать 4. Перейти во вкладку «Мои бронирования» |
| **Тестовые данные** | Имя пользователя: ivanov123  Пароль: hashed\_password\_1  Дата въезда: 24 ноября 2024  Дата выезда: 27 ноября 2024 |
| **Ожидаемый результат** | Бронирование появилось в таблице «Мои бронирования» |
| **Фактический результат** | Бронирование появилось в таблице «Мои бронирования» |
| **Статус** | Зачет |
| **Предварительное условие** | Приложение должно быть открыто |
| **Постусловие** | Бронирование появилось в таблице «Мои бронирования» |
| **Примечания/комментарии** | — |

Таблица 5 — Тестовый пример 4. Добавление бронирования

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_Booking\_2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Попытка забронировать домик |
| **Краткое изложение теста** | Проверка возможности добавления бронирования пользователем с неверными данными |
| **Этапы теста** | 1. Зайти в приложение под пользователя с ролью «Клиент» 2. Выбрать даты бронирования 3. Нажать кнопку забронировать 4. Перейти во вкладку «Мои бронирования» |
| **Тестовые данные** | Имя пользователя: ivanov123  Пароль: hashed\_password\_1  Дата въезда: 27 ноября 2024  Дата выезда: 24 ноября 2024 |
| **Ожидаемый результат** | Бронирование не появляется в таблице «Мои бронирования» |
| **Фактический результат** | Бронирование не появилось в таблице «Мои бронирования» |
| **Статус** | Зачет |
| **Предварительное условие** | Приложение должно быть открыто |
| **Постусловие** | Бронирование не появляется в таблице «Мои бронирования» |
| **Примечания/комментарии** | — |

Таблица 5 — Тестовый пример 4. Изменение статуса бронирования

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_BookingStatus\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Изменение статуса бронирования |
| **Краткое изложение теста** | Проверка возможности изменения статуса бронирования работником |
| **Этапы теста** | 1. Зайти в приложение под пользователя с ролью «Работник» 2. Выбрать бронирование в таблице во вкладке «Подтверждение» 3. Нажать кнопку «Подтвердить бронь» |
| **Тестовые данные** | Имя пользователя: petrov321  Пароль: hashed\_password\_2  Дата въезда: 27 ноября 2024  Дата выезда: 24 ноября 2024 |
| **Ожидаемый результат** | Бронирование появляется в таблице во вкладке «Подтверждённые бронирования» |
| **Фактический результат** | Бронирование не появилось в таблице «Подтверждённые бронирования» |
| **Статус** | Незачет |
| **Предварительное условие** | — |
| **Постусловие** | — |
| **Примечания/комментарии** | — |

Также для приложения были написаны Unit-тесты, проверяющие работу некоторых методов (Рисунок 29).

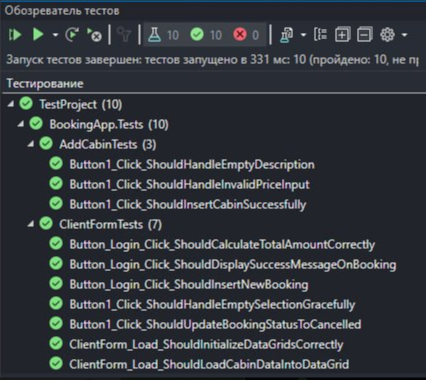


Рисунок 29 — Пройденные тесты

# Задание 6. GitHub. Оценка проекта

# Приложение А База данных

Таблица А1 — Словарь данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Events** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | EventID | INT | Да | Идентификатор события |
|  | EventName | NVARCHAR(max) | Да | Название события |
|  | EventInfo | NVARCHAR(max) | Нет | Описание события |
|  | EventDateTime | DATETIME2 | Нет | Дата и время проведения события |
|  |  |  |  |  |
| **Users** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | UserID | INT | Да | Идентификатор пользователя |
|  | PhoneNumber | NVARCHAR(15) | Да | Номер телефона пользователя, уникальный |
|  | Name | NVARCHAR(20) | Да | Имя пользователя |
|  | Email | NVARCHAR(255) | Нет | Email пользователя |
|  | PasswordHash | NVARCHAR(255) | Да | Хеш пароля пользователя |
|  | Role | NVARCHAR(30) | Нет | Роль пользователя, по умолчанию 'Клиент' |
|  |  |  |  |  |
| **Rooms** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | RoomID | INT | Да | Идентификатор комнаты |
|  | RoomType | NVARCHAR(50) | Нет | Тип комнаты |
|  | Capacity | INT | Нет | Вместимость комнаты |
|  | PricePerNight | DECIMAL(5, 2) | Нет | Цена за ночь |
|  |  |  |  |  |
| **Status** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | StatusID | INT | Да | Идентификатор статуса |
|  | StatusName | NVARCHAR(100) | Да | Название статуса |
|  |  |  |  |  |
| **Bookings** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | BookingID | INT | Да | Идентификатор бронирования |
| FK | UserID | INT | Да | Внешний ключ, указывает на таблицу Users |
| FK | RoomID | INT | Да | Внешний ключ, указывает на таблицу Rooms |
|  | CheckInDate | DATETIME2 | Да | Дата и время заезда |
|  | CheckOutDate | DATETIME2 | Нет | Дата и время выезда |
| FK | StatusID | INT | Нет | Внешний ключ, указывает на таблицу Status |
|  |  |  |  |  |
|  |  |  |  |  |
| **ServiceByTour** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | ServiceID | INT | Да | Идентификатор услуги |
|  | ServiceName | NVARCHAR(100) | Да | Название услуги |
|  | ServiceInfo | NVARCHAR(max) | Нет | Описание услуги |
|  |  |  |  |  |
| **ServiceByTour** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| FK | BookingID | INT | Да | Внешний ключ, указывает на таблицу Bookings |
| FK | ServiceID | INT | Да | Внешний ключ, указывает на таблицу Services |

А2 — Листинг SQL-скрипта на создание таблиц в БД

create table Events (

EventID INT IDENTITY NOT NULL PRIMARY KEY,

EventName NVARCHAR(max) NOT NULL,

EventInfo NVARCHAR(max),

EventDateTime DATETIME2

)

create table Users (

UserID INT IDENTITY NOT NULL PRIMARY KEY,

PhoneNumber NVARCHAR(15) NOT NULL UNIQUE,

Name NVARCHAR(20) NOT NULL,

Email NVARCHAR(255),

PasswordHash NVARCHAR(255) NOT NULL,

Role NVARCHAR(30) DEFAULT N'Клиент'

)

create table Rooms (

RoomID INT IDENTITY NOT NULL PRIMARY KEY,

RoomType NVARCHAR(50),

Capacity INT,

PricePerNight DECIMAL(5, 2)

)

create table Status (

StatusID INT IDENTITY NOT NULL PRIMARY KEY,

StatusName NVARCHAR(100) NOT NULL

)

create table Bookings (

BookingID INT IDENTITY NOT NULL PRIMARY KEY,

UserID INT FOREIGN KEY REFERENCES Users(UserID) ON DELETE CASCADE,

RoomID INT FOREIGN KEY REFERENCES Rooms(RoomID) ON DELETE CASCADE,

CheckInDate DATETIME2 NOT NULL,

CheckOutDate DATETIME2,

StatusID INT FOREIGN KEY REFERENCES Status(StatusID)

)

create table Services (

ServiceID INT IDENTITY NOT NULL PRIMARY KEY,

ServiceName NVARCHAR(100) NOT NULL,

ServiceName NVARCHAR(max)

)

create table BookingsServices (

BookingID INT FOREIGN KEY REFERENCES Bookings(BookingID) ON DELETE CASCADE,

ServiceID INT FOREIGN KEY REFERENCES Services(ServiceID) ON DELETE CASCADE

)