

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий
Кафедра программной инженерии
Специальность 1-40 01 01 Программное обеспечение информационных технологий
Направление специальности 1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет приложений)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»
Тема Web-приложение “Психологический центр”

Исполнитель
студент (ка) 2 курса группы 4 Карebo Никита Сергеевич
(Ф.И.О.)

Руководитель работы ассистент Чистякова Ю.А.
(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой _____
Председатель Пацей Н.В.
(подпись)

Минск 2023

Содержание

| | |
|---|----|
| ВВЕДЕНИЕ | 3 |
| 1 Аналитический обзор литературы | 4 |
| 1.1 INSIDE | 4 |
| 1.2 Grow Up | 5 |
| 1.3 Nora | 6 |
| 2 Анализ требований к программному средству и разработка функциональных требований..... | 8 |
| 2.1 Теоретический анализ и моделирование предметной области..... | 8 |
| 2.2 Описание функциональности ПС..... | 8 |
| 2.3 Спецификация функциональных требований | 9 |
| 3 Проектирование программного средства | 11 |
| 3.1 Проектирование базы данных | 11 |
| 3.2 Проектирование архитектуры проекта | 16 |
| 4 Создание (реализация) программного средства..... | 19 |
| 4.1 Обобщённая структура приложения | 19 |
| 4.2 Реализация архитектуры MVVM | 21 |
| 4.3 Реализация паттерна Repository и UnitOfWork | 22 |
| 4.4 Реализация паттерна Command | 23 |
| 5 Тестирование работоспособности и анализ полученных результатов | 24 |
| 6 Руководство по использованию | 30 |
| 6.1 Руководство по использованию администраторам | 31 |
| 6.2 Руководство по использованию пользователем | 35 |
| 6.3 Руководство по использованию гостем | 40 |
| ЗАКЛЮЧЕНИЕ..... | 42 |
| СПИСОК ЛИТЕРАТУРЫ | 43 |
| ПРИЛОЖЕНИЕ А..... | 44 |
| ПРИЛОЖЕНИЕ Б | 44 |
| ПРИЛОЖЕНИЕ В..... | 45 |
| ПРИЛОЖЕНИЕ Г | 46 |
| ПРИЛОЖЕНИЕ Д..... | 46 |

ВВЕДЕНИЕ

В настоящее время ни для кого не секрет, что самой важным для любого человека является – эмоции. Чувства и эмоции тесно связаны с нашими внутренними качествами, они просто являются отражением того, что происходит внутри каждого человека. Людям всегда хочется рассказать о своих переживаниях или проблемах. По этой причине люди проводят огромное количество времени в поисках нужного психолога, который им подойдёт.

Данное программное устройство облегчит поиск специалиста в данной области, предоставляя возможность бронировать сеансы с подробным описанием, не выходя из дома.

Любой желающий человек может самостоятельно создать или войти в учётную запись. Это дает возможность подробно ознакомиться со всеми психологами учреждения и процедурами, которые они предоставляют, а также совершить заказ на определенное число, выбрав удобную для себя дату и время.

Для каждого авторизованного пользователя предоставляется возможность просмотреть свою персональную страницу, управляя её составляющим. В любой момент времени пользователь может управлять и изменять забронированные сеансы, а также прочитать краткую информации о личном психологе.

Одним из важных достоинств приложения является предоставления удобного функционала для администратора, который в свою очередь имеет возможность вносить изменения в базу данных, в частности созданием страниц для специалистов и добавлении процедур.

Если посетитель не хочет иметь собственный аккаунт, или зашёл в ознакомительных целях, приложение предоставляет возможность войти в режиме гостя, который в свою очередь предоставляет ограниченный функционал.

1 Аналитический обзор литературы

Одним из важных критериев и выбора стратегии разработки приложения является анализ конкурентов. Если создать приложение типичное с другими, это не поспособствует расширению клиентской базы. При помощи анализа конкурентов и реализации востребованных пользователями требований, позволит приложению стать удобнее и в тоже время эффективнее.

1.1 INSIDE

На сегодняшний день очень популярным альтернативным решением является интернет-ресурс “INSIDE”, представлено на рисунке 1.1.

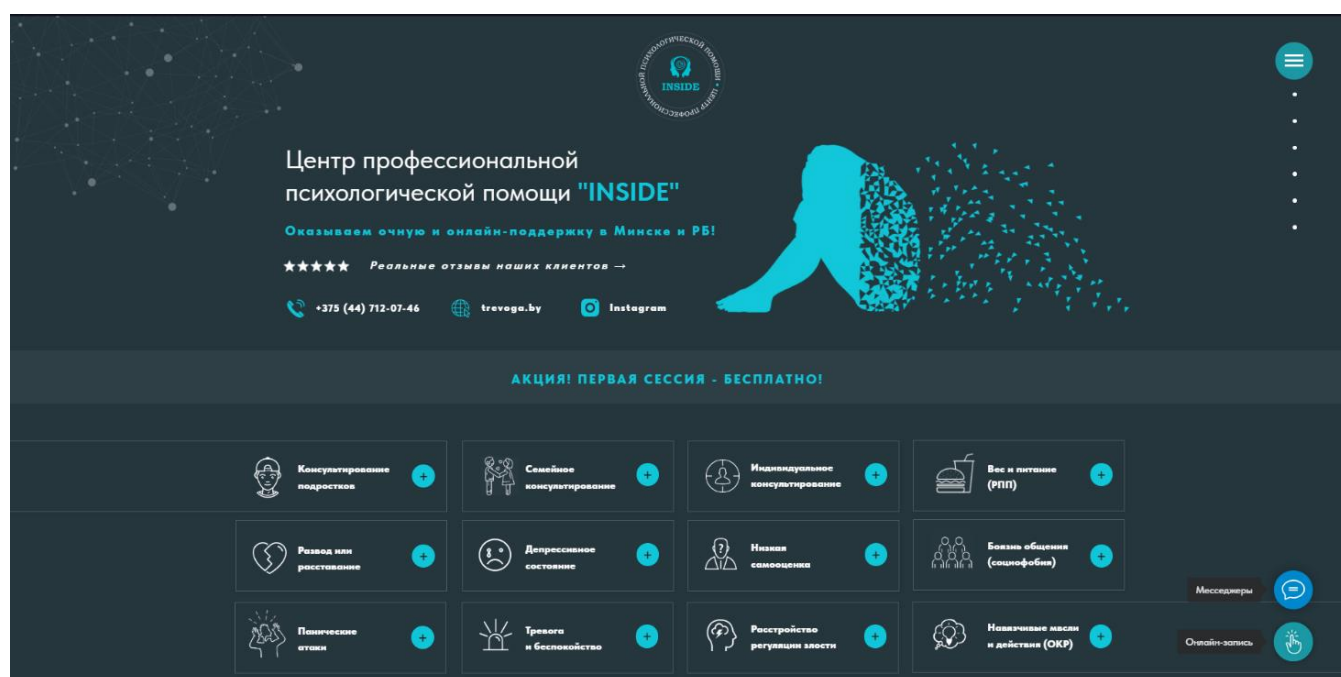


Рисунок 1.1 - Сайт “INSIDE”

Данный сайт имеет достаточно грамотный и красивый интерфейс, предоставляет удобную навигационную панель. Также есть возможность регистрации аккаунта через такие социальные сети, такие как Facebook, ВКонтакте, Одноклассники. Присутствуют отзывы от пользователей данного сайта, что положительно влияет на выборе именно этого интернет-ресурса.

При выборе направления сайт предоставляет информацию о возможной причине возникновения данной проблемы, а также возможные решения в виде предоставлении формы для регистрации заказа и краткой информации о будущих этапах консультирования.

Так же имеется вкладка “Новости”, где можно просмотреть о последних достижениях и событиях в области психологии. Как было замечено, его обновляют не часто, и данная вкладка не несет для пациента никакой нужной информации, лишь создает визуальный шум, что зачастую только может сбивать с толку.

На сайте имеется пункт “О нас” которая предоставляет краткую информацию

об организации и возможность связаться при возникновении каких-либо проблем.

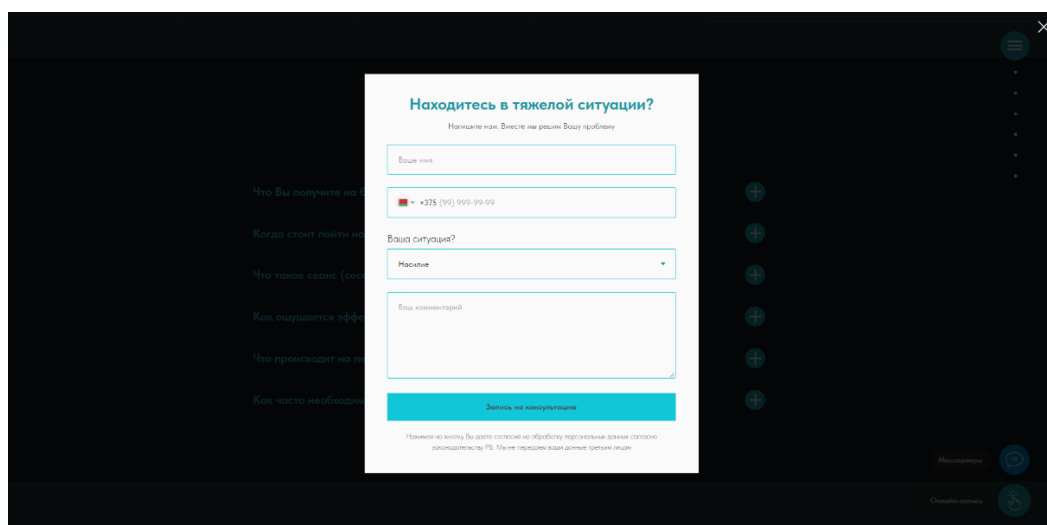


Рисунок 1.2 - Бронирование сеанса на сайте “INSIDE”

Говоря о недостатках сайта, при бронировании сеанса мы встречаемся с формой, где нас просят ввести свои данные. Нет возможности выбрать определенного специалиста, а лишь только его направление. Можно лишь указать “желательную” дату и время посещения, которая может и не совпасть с реальным графиком нужного вам специалиста.

1.2 Grow Up

Следующим сайтом является “Grow Up”, который также является популярным среди пользователей. На рисунке 1.2 предоставлен интерфейс данного сайта.

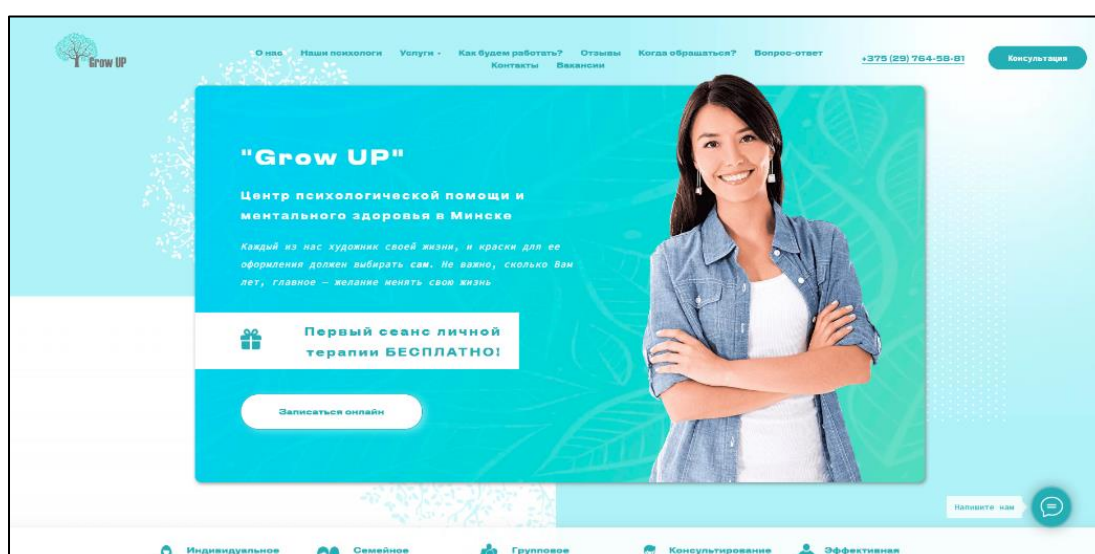


Рисунок 1.3 - Сайт “Grow Up”

Сайт предлагает пользователям простой в понимании и освоении интерфейс,

что делает его привлекательнее при выборе среди пользователей. Представляет довольно краткое и достаточное описание специалистов с указанием их времени работы. Предлагаются еженедельные скидочные акции при бронировании сеанса в определенном направлении.

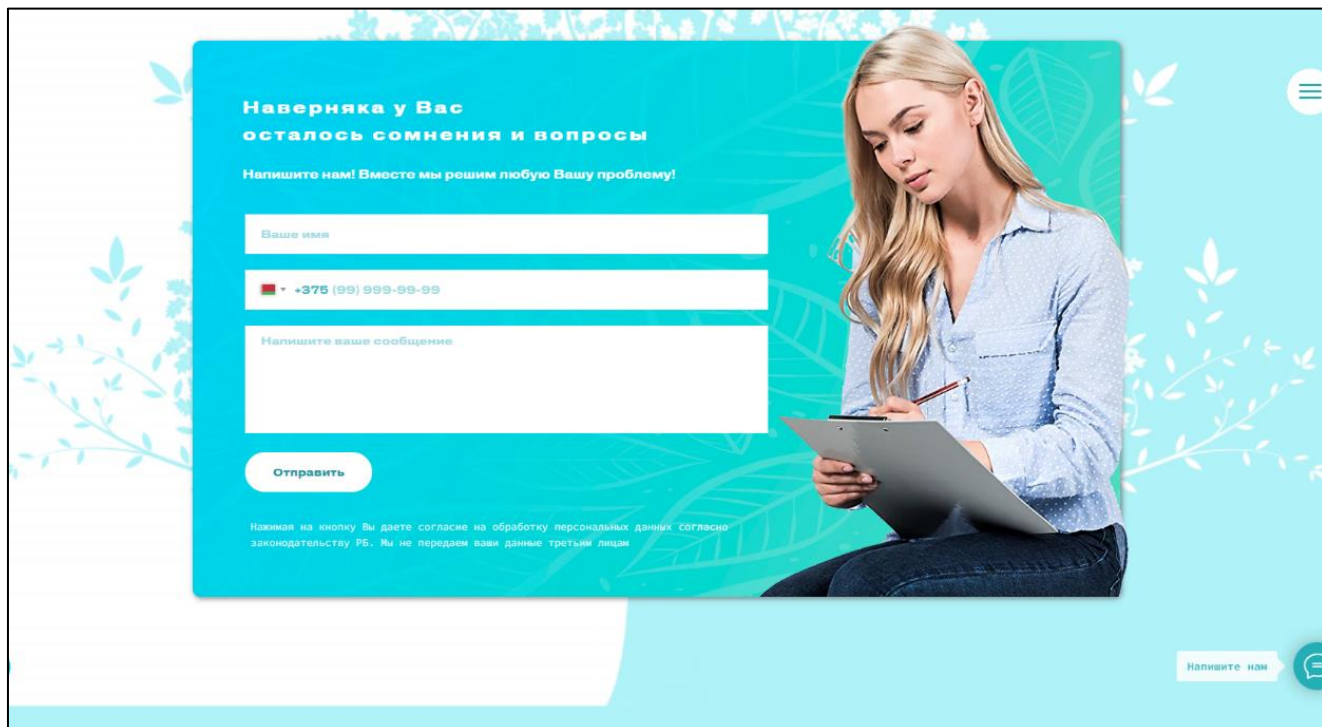


Рисунок 1.4 - Бронирование сеанса на сайте “Grow Up”

При бронировании сеанса мы встречаемся с формой для заполнения, в котором нужно указать телефон и имя клиента. Данная форма не предоставляет возможности выбрать конкретную дату, специалиста, и даже желаемое направление. Уточнить данные критерии возможно дожидаясь звонка от консультанта. Нет возможности просмотреть текущие сеансы, если только вы снова не свяжетесь с администрацией данной организации. Все вышеперечисленные проблемы говорят о лишней трате свободного времени клиента и неэффективном обслуживании.

Из недостатков можно выделить слишком простую структуру. Сайт состоит только из одной страницы, которая разделена на информативные блоки, и довольно простым навигационным меню. Не присутствуют отзывы от клиентов, что внушает недоверие и нежелание пользоваться данным ресурсом.

1.3 Nora

Ещё один и не менее популярный сайт о оказании помощи в области психологии является интернет-ресурс “Nora”.

Анализируемый сайт встречает нас довольно простым дизайном, что в достаточной мере отталкивает клиентов в приобретении услуг от данной организации, имеет краткое и интуитивно понятное навигационное меню, с возможностью перейти к интересующей вас теме или узнать подробнее об

предоставляем возможностях организации.

Дизайн главной страницы сайта представлен на рисунке 1.5.

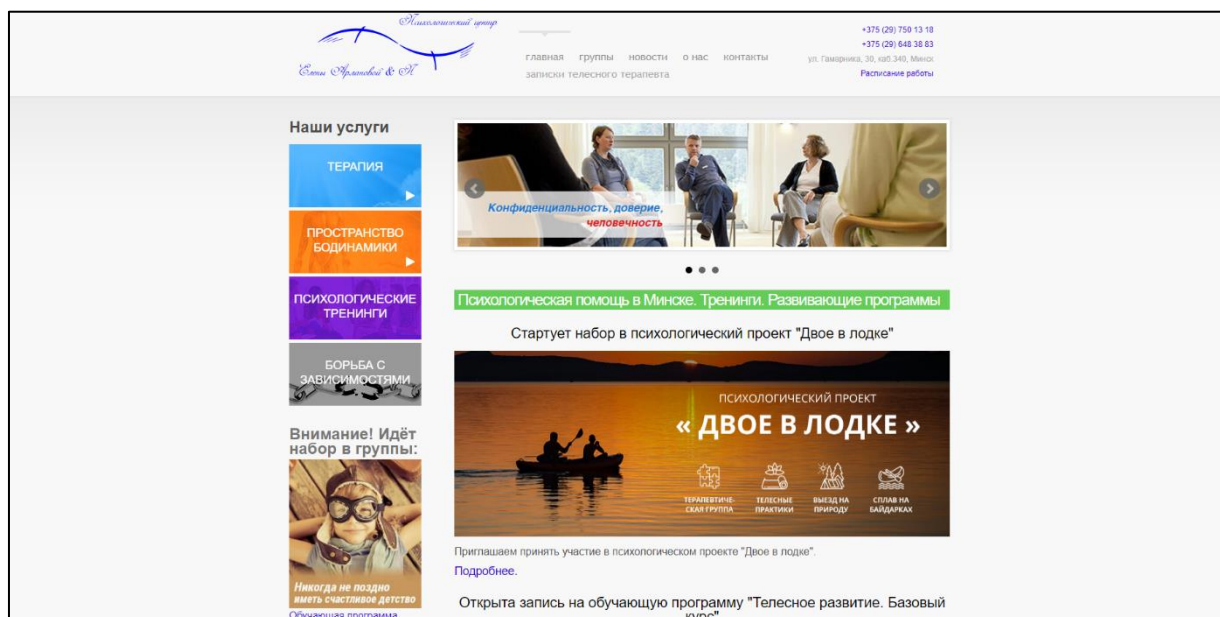


Рисунок 1.5 - Сайт “Nora”

Так же на главной странице сайта присутствуют новости о недавно открытых проектах, предлагая клиенту принять участие в их развитии. Присутствует возможность выбора направления, в котором указывается ряд специалистов в данной области с краткой информацией. Есть возможность позвонить лечащему врачу или написать ему в любой доступной соцсети.

Сайт не даёт возможность создать аккаунт или заполнить форму для бронирования сеанса, всё это происходит только при личном звонке специалисту.

2 Анализ требований к программному средству и разработка функциональных требований

2.1 Теоретический анализ и моделирование предметной области

Теоретический анализ предметной области играет важную роль в разработке приложения, поскольку он помогает полностью понять и учесть основные аспекты и требования предметной области.

Приложение “Clean Brain” реализует три основные категории функциональности. Первая категория включает в себя функции, которые доступны только авторизованным пользователям. Вторая категория включает в себя функции, которые доступны только администраторам. Третья категория предоставляет возможности, которые частично перекликаются с функциональностью, доступной авторизованным клиентам, но в рамках ограниченного и гостевого доступа. Все три категории функциональности, а именно функциональность для администратора, авторизованного клиента и гостя, будут подробно описаны в следующих подразделах.

Для реализации приложения “Clean Brain” была разработана следующая модель предметной области:

1. Авторизованный пользователь: это пользователь, который успешно прошел процесс аутентификации и имеет действительную учетную запись.

2. Гость: это незарегистрированный пользователь, который использует приложение “Clean Brain” в ознакомительных целях.

3. Администратор: это пользователь с особыми правами, который имеет доступ к различным функциям приложения, связанными с управлением системой.

4. Психолог: это высококвалифицированный специалист, обладающий знаниями и опытом в области психологии, который предоставляет индивидуальные услуги для клиентов.

5. Услуга: это профессиональная помощь, предоставляемая квалифицированным психологом, с целью помочь людям разрешить психологические проблемы.

2.2 Описание функциональности ПС

Функционал приложения играет важную роль в обеспечении качественного пользовательского опыта и успешной реализации его основной цели. В случае приложения “Clean Brain” функционал имеет особую значимость, поскольку оно предназначено для психологического центра и оказания поддержки клиентам в области психического благополучия.

Как говорилось ранее весь предоставляемый функционал приложением “Clean Brain” можно разделить на три категории. Первой категорией является функционал, который предоставляет ряд полезных возможностей и инструментов, направленных на поддержку и улучшение их психологического благополучия:

1. Регистрация и профиль: клиенты могут создать свой учетную запись, заполнив необходимую информацию о себе.

2. Просмотр и выбор специалистов: клиенты имеют возможность просматривать профили психологов, ознакомиться с их квалификацией, специализацией. Они могут выбрать психолога, соответствующего их потребностям и предпочтениям.

3. Выбор услуги: клиенты могут выбрать желаемую услугу, а также просмотреть подробную информацию об желаемой услуге.

4. Просмотр текущих назначенных сеансов: если у клиента есть назначенные сеансы с психологом, он может всегда просмотреть либо же отредактировать текущие сеансы.

5. Просмотр и добавление отзывов: клиент всегда может просмотреть добавленные другими пользователями приложения отзывы, а также оставить свой.

6. Просмотр информации о психологическом центре: пользователь всегда может подробно изучить информацию о психологическом центре.

Второй категорией является функционал, реализованный для администраторов. Данный функционал включает в себя:

1. Управление психологами: администратор имеет возможность добавлять новых психологов в систему, редактировать их профили и управлять их доступностью для клиентов. Он также может просматривать и контролировать информацию о квалификации, расписании психологов.

2. Управление услугами: администратор может добавлять новые услуги, а также их редактировать. Эта функция позволяет администратору гибко управлять списком доступных услуг, обновлять их описание, параметры и стоимость в соответствии с текущими требованиями и потребностями центра.

3. Управление отзывами: администратору предоставляется возможность просматривать все отзывы, а также если отзывы нарушают некие правила, удалять их.

Последней категорией является функционал, предоставляемый для гостей, который поможет ему получить представление о психологическом центре и принять решение о дальнейшем взаимодействии. Данный функционал включает в себя:

1. Просмотр информации о психологах: гость может получить информацию о психологах, работающих в психологическом центре. Он может прочесть их профили, узнать о квалификации, специализации и опыте каждого психолога.

2. Просмотр услуг и описаний: гость может ознакомиться с различными услугами, предлагаемыми психологическим центром, и получить детальное описание каждой из них.

3. Просмотр информации о психологическом центре: гость может получить доступ к общей информации о психологическом центре, такой как его миссия, ценности, основные услуги и преимущества.

Для более подробного ознакомления с функциональностью приложения, в Приложении А [1] представлена диаграмма вариантов использования.

2.3 Спецификация функциональных требований

В программном средстве “Clean Brain” при запуске необходимо реализовать регистрацию или авторизацию для дальнейшего полного использования

приложения, но можно воспользоваться режимом Гостя, что ограничивает функционал. Для авторизации параметрами являются уникальный логин и пароль пользователя, которые содержатся в базе данных. Для регистрации необходимо ввести почту, куда будут приходить письма-уведомления, логин и пароль. Введенные данные проходят валидацию: логин должен быть уникальным, почта должна иметь общепринятый формат. Если данные прошли валидацию успешно, то они заносятся в базу данных, а пользователь получает полный доступ к приложению.

При входе, пользователя встречает окно с приветствием, через которое возможно перейти к окну просмотра психологов, услуг, отзывов, информации о центре и своей учетной записи.

Перейдя в свою учетную запись, пользователь может изменить информацию о себе.

Администратор также может войти через окно логирования, после чего ему предоставляется особый функционал. На странице с информацией о психологах он может добавить нового психолога нажав по кнопке “добавить”, которая вызывает дополнительное окно для ввода данных о психологе: имя, фамилия, отчество, специальность, учебная степень, опыт, изображение и расписание. Здесь же администратор сможет добавить талоны. Также администратор может изменять или удалять психологов.

На странице о услугах администратор может добавить процедуры, кликнув по кнопке “добавить”, что вызывает новое окно-форму для регистрации процедуры. Также может изменить или же вовсе удалить услугу, кликнув по окну “удалить/изменить”

Кнопка для выхода из аккаунта возвращает пользователя к окну авторизации.

3 Проектирование программного средства

3.1 Проектирование базы данных

Данное приложение использует базу данных с названием "Psychological_Center" и работает на платформе Microsoft SQL Server 2020. Microsoft SQL Server 2020 является реляционной системой управления базами данных (СУБД), разработанной компанией Microsoft. Это мощное и надежное решение для хранения, управления и обработки данных, которое обеспечивает эффективное взаимодействие с базой данных для данного психологического центра. Логическая схема базы данных представлена на рисунке 3.1.

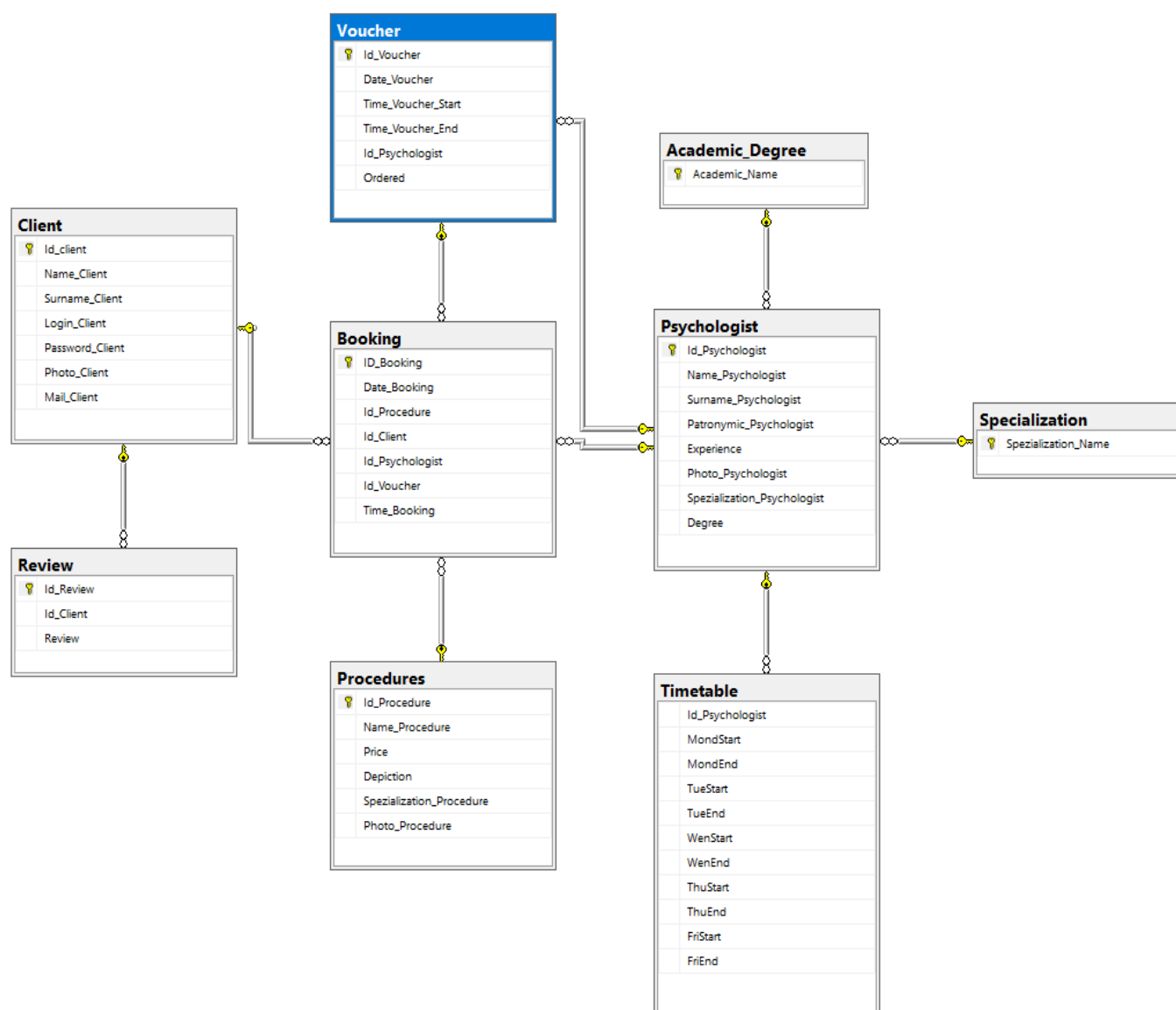


Рисунок 1.3 - Логическая схема данных

Данная база данных содержит в себе 9 таблиц: Client, Voucher, Academic_Degree, Specialization, Psychologist, Booking, Review, Procedures, Timetable.

Рисунок 3.2 иллюстрирует структуру таблицы “Client”, которая содержит информацию о клиенте.

| | Имя столбца | Тип данных | Разрешить ... |
|---|-----------------|----------------|-------------------------------------|
| 🔑 | Id_client | int | <input type="checkbox"/> |
| | Name_Client | nvarchar(MAX) | <input type="checkbox"/> |
| | Surname_Client | nvarchar(MAX) | <input type="checkbox"/> |
| | Login_Client | nvarchar(MAX) | <input type="checkbox"/> |
| | Password_Client | nvarchar(MAX) | <input type="checkbox"/> |
| | Photo_Client | varbinary(MAX) | <input checked="" type="checkbox"/> |
| | Mail_Client | nvarchar(MAX) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.2 - Структура таблицы “Client”

Таблица Client состоит из 7 столбцов, где поле “Id_Client” является первичным ключом. “Id_client” – id клиента, поле Name_Client хранит имя пользователя, Surname_Client – фамилию, Login_Client – логин клиента, Password_Client-пароль, Photo_Client – изображение, а поле Mail_client хранит в себе почту клиента, на которую будут отправляться уведомления.

| | Имя столбца | Тип данных | Разрешить ... |
|---|-----------------------------|----------------|-------------------------------------|
| 🔑 | Id_Psychologist | int | <input type="checkbox"/> |
| | Name_Psychologist | nvarchar(MAX) | <input type="checkbox"/> |
| | Surname_Psychologist | nvarchar(MAX) | <input type="checkbox"/> |
| | Patronymic_Psychologist | nvarchar(MAX) | <input type="checkbox"/> |
| | Experience | smallint | <input type="checkbox"/> |
| | Photo_Psychologist | varbinary(MAX) | <input checked="" type="checkbox"/> |
| | Description | nvarchar(MAX) | <input checked="" type="checkbox"/> |
| | Spezialization_Psycholog... | nvarchar(60) | <input type="checkbox"/> |
| | Degree | nvarchar(60) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.3 - Структура таблицы “Psychologist”

Диаграмма на рисунке 3.3 иллюстрирует структуру таблицы “Psychologist”, которая содержит информацию о психологах. Она связана с таблицей “Academic_Degree” через поле “Degree”, с таблицей “Specialization” – через поле “Spezialization_Psycholog...”. В данной таблице поле “Id_Psychologist” является первичным ключом, а также хранит в себе id психолога. “Name_Psychologist” содержит имя психолога, “Surname_Psychologist” – фамилия психолога,

“Patronymic_Psychologist” – отчество, “Specialization_Psychologist” – содержит специализацию психолога, “Degree” – ученная степень психолога, “Experience” – год начала карьеры, “Photo_Psychologist” – фотография психолога, “Description” – описание психолога.

| | Имя столбца | Тип данных | Разрешить ... |
|----|--------------------------|----------------|-------------------------------------|
| ►🔑 | Id_Procedure | int | <input type="checkbox"/> |
| | Name_Procedure | nvarchar(MAX) | <input type="checkbox"/> |
| | Price | float | <input type="checkbox"/> |
| | Depiction | nvarchar(MAX) | <input type="checkbox"/> |
| | Spezialization_Procedure | nvarchar(60) | <input checked="" type="checkbox"/> |
| | Photo_Procedure | varbinary(MAX) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.4 - Структура таблицы “Procedure”

Рисунок 3.4 иллюстрирует структуру таблицы “Procedures”, которая содержит информацию о процедурах. Она связана с таблицей “Specialization” через поле “ Specialization _Procedure”. В данной таблице поле “Id_Procedure” является первичным ключом, а также оно содержит id процедуры, “Name_procedure” – название процедуры, “Depiction” – описание, “Price” – цена процедуры, “Photo_Procedure” –.фотография процедуры, “Spezialization_Procedure” – специализация.

| | Имя столбца | Тип данных | Разрешить ... |
|----|-----------------|------------|-------------------------------------|
| ►🔑 | Id_Psychologist | int | <input type="checkbox"/> |
| | MondStart | time(7) | <input checked="" type="checkbox"/> |
| | MondEnd | time(7) | <input checked="" type="checkbox"/> |
| | TueStart | time(7) | <input checked="" type="checkbox"/> |
| | TueEnd | time(7) | <input checked="" type="checkbox"/> |
| | WenStart | time(7) | <input checked="" type="checkbox"/> |
| | WenEnd | time(7) | <input checked="" type="checkbox"/> |
| | ThuStart | time(7) | <input checked="" type="checkbox"/> |
| | ThuEnd | time(7) | <input checked="" type="checkbox"/> |
| | FriStart | time(7) | <input checked="" type="checkbox"/> |
| | FriEnd | time(7) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.5 - Структура таблицы “Timetable”

Диаграмма на рисунке 3.5 иллюстрирует структуру таблицы “Timetable”, которая содержит информацию о расписании врача. Она связана с таблицей “Psychologist” через поле “Id_Psychologist”. В данной таблице поле “Id_Psychologist” является первичным ключом, а также содержит id психолога, “MondStart” – с какого времени психолог начинает работать в Понедельник, “MondEnd” – во сколько психолог заканчивает работать в Понедельник, “TueStart” – с какого времени психолог начинает работать во Вторник, “TueEnd” – во сколько психолог

заканчивает работать во Вторник, “WenStart” – с какого времени психолог начинает работать в Среду, “WenEnd” – во сколько психолог заканчивает работать в Среду, “ThuStart” – с какого времени психолог начинает работать в Четверг, “ThuEnd” – во сколько психолог заканчивает работать в Четверг, “FriStart” – с какого времени психолог начинает работать в Пятницу, “FriEnd” – во сколько психолог заканчивает работать в Пятницу.

| | Имя столбца | Тип данных | Разрешить ... |
|-----|---------------|--------------|--------------------------|
| ▶ 🔑 | Academic_Name | nvarchar(60) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.6 - Структура таблицы “Academic_Degree”

Диаграмма на рисунке 3.6 иллюстрирует структуру таблицы “Academic_Degree”, которая содержит информацию о учённых степенях. В данной таблице поле “Academic_Name” является первичным ключом и содержит название учённой степени.

| | Имя столбца | Тип данных | Разрешить ... |
|-----|---------------------|--------------|--------------------------|
| ▶ 🔑 | Spezialization_Name | nvarchar(60) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.7 - Структура таблицы “Specialization”

Диаграмма на рисунке 3.7 иллюстрирует структуру таблицы “Specialization”, которая содержит информацию о специализациях. В данной таблице поле “Spezialization_Name” является первичным ключом и содержит специализацию.

| | Имя столбца | Тип данных | Разрешить ... |
|-----|--------------|----------------|-------------------------------------|
| ▶ 🔑 | Id_Review | int | <input type="checkbox"/> |
| | Photo_Review | varbinary(MAX) | <input checked="" type="checkbox"/> |
| | Name_Client | nvarchar(MAX) | <input type="checkbox"/> |
| | Id_Client | int | <input type="checkbox"/> |
| | Review | nvarchar(MAX) | <input type="checkbox"/> |

Рисунок 3.8 - Структура таблицы “Review”

Диаграмма на рисунке 3.8 иллюстрирует структуру таблицы “Review”, которая содержит информацию о отзывах. Она связана с таблицей “Client” через поле “Id_Client”. В данной таблице поле “Id_Review” является первичным ключом, а также содержит id отзыва, поле “Id_Client” содержит id пользователя,

совершившего заказ. Поле “Name_Client” – имя пользователя оставившего отзыв, “Photo_Review” – изображение пользователя, “Review” – сам отзыв.

| | Имя столбца | Тип данных | Разрешить ... |
|---|--------------------|------------|--------------------------|
| 🔑 | Id_Voucher | int | <input type="checkbox"/> |
| | Date_Voucher | date | <input type="checkbox"/> |
| | Time_Voucher_Start | time(7) | <input type="checkbox"/> |
| | Time_Voucher_End | time(7) | <input type="checkbox"/> |
| | Id_Psychologist | int | <input type="checkbox"/> |
| | Ordered | nchar(10) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.9 - Структура таблицы “Voucher”

Диаграмма на рисунке 3.9 иллюстрирует структуру таблицы “Voucher”, которая содержит информацию о талонах. Она связана с таблицей “Psychologist” через поле “Id_Psychologist”. В данной таблице поле “Id_Voucher” является первичным ключом. Поле “Id_Voucher” содержит id талона, “Date_Voucher” – дата талона, “Time_Voucher_Start” – время начала приема, “Time_Voucher_End” – время окончания приема, “Id_Psychologist” -id психолога, “Ordered” - поле, оповещающее, что талон забронирован или нет.

| | Имя столбца | Тип данных | Разрешить ... |
|---|-----------------|------------|-------------------------------------|
| 🔑 | ID_Booking | smallint | <input type="checkbox"/> |
| | Date_Booking | date | <input type="checkbox"/> |
| | Id_Procedure | int | <input type="checkbox"/> |
| | Id_Client | int | <input type="checkbox"/> |
| | Id_Psychologist | int | <input type="checkbox"/> |
| | Id_Voucher | int | <input type="checkbox"/> |
| | Time_Booking | time(7) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.10 - Структура таблицы “Booking”

Диаграмма на рисунке 3.10 иллюстрирует структуру таблицы “Booking”, которая содержит информацию о талонах. Таблица “Booking” содержит информацию о заказах. Она связана с таблицей “Client” через поле “Id_Client”, с таблицей “Voucher” – через поле “Id_Voucher”, с таблицей “Psychologist” – через поле “Id_Psychologist”, с таблицей “Procedure” – через поле “Id_Procedure”. В

данной таблице поле “ID_Booking” является первичным ключом, а также содержит id заказа, “Date_Booking” – дата заказа, “Id_Procedure” – id процедуры, “Id_Client” – id клиента, “Id_Psychologist” – id психолога, “Id_Voucher” – id талона, “Time_Booking” – время заказа.

3.2 Проектирование архитектуры проекта

На рисунке 3.11 изображена диаграмма вариантов использования данного приложения.

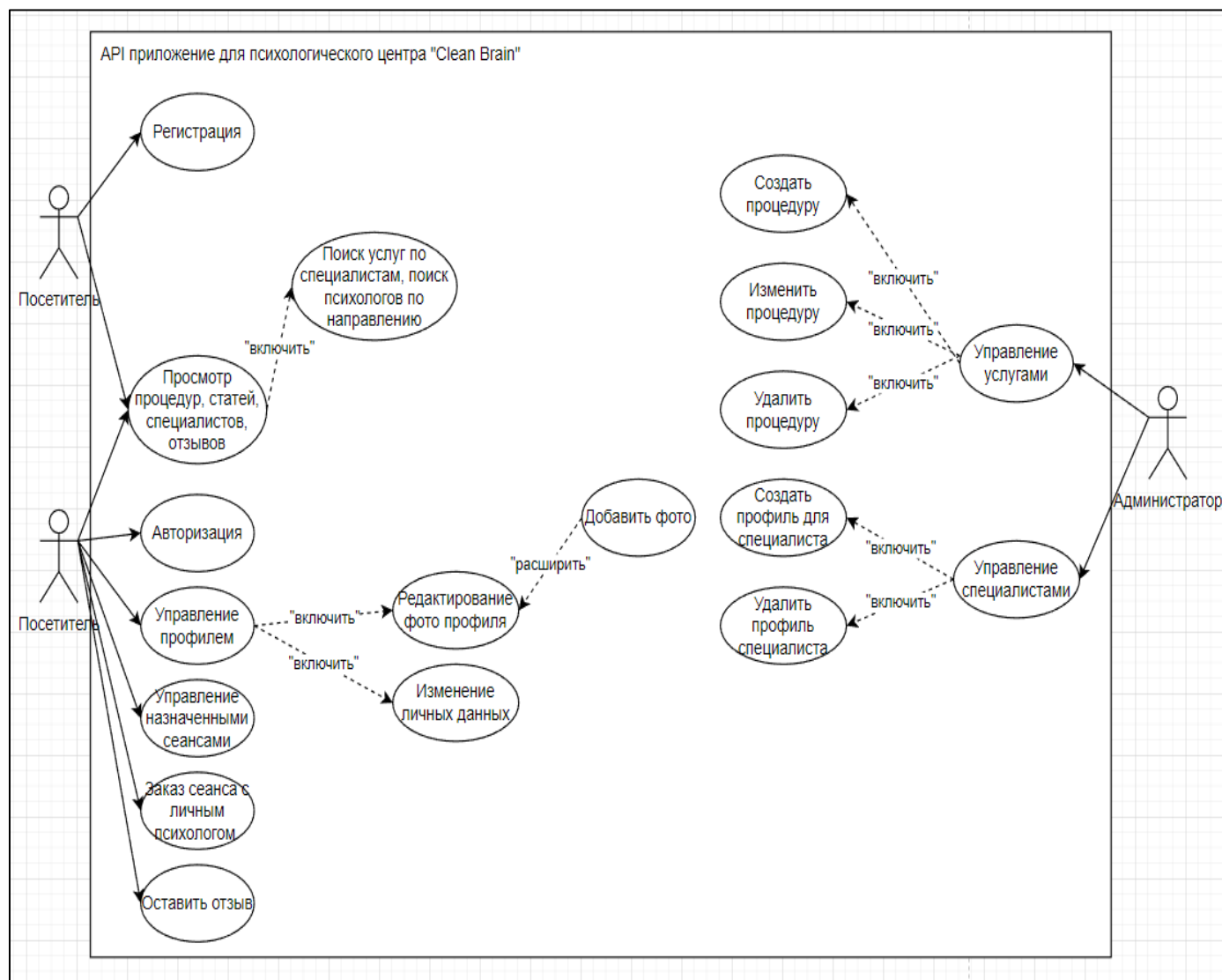


Рисунок 3.11 - Диаграмма вариантов использования

Данная диаграмма вариантов использования наглядно демонстрирует функциональность приложения "Clean Brain" для различных ролей пользователей, таких как администратор, гость и авторизованный пользователь. Она позволяет легко понять, какие функции доступны каждому из этих пользователей и как они могут взаимодействовать с приложением.

На рисунке 3.12 изображена диаграмма последовательности для авторизации пользователя.

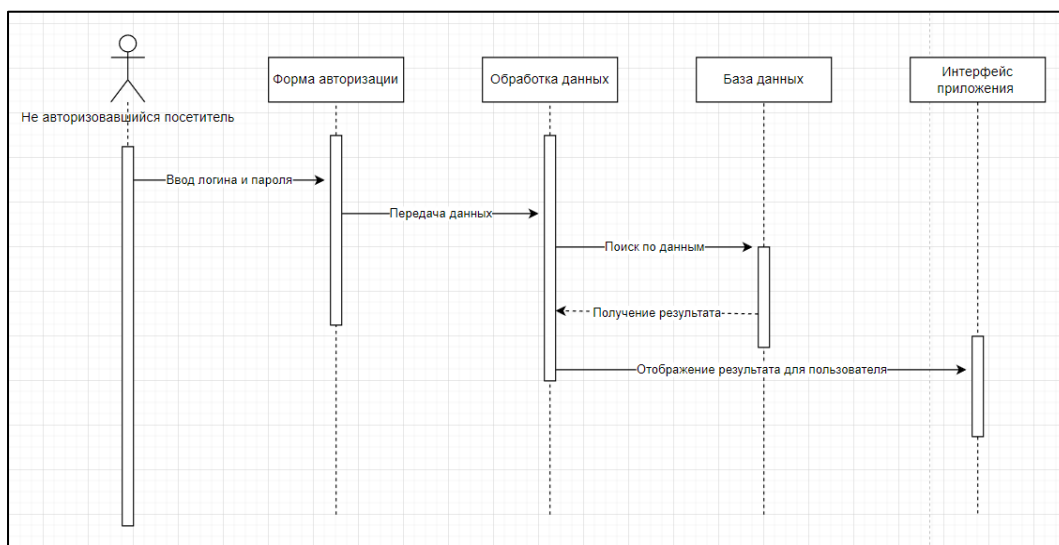


Рисунок 3.12 - Диаграмма последовательности для авторизации пользователя

Данная диаграмма последовательности наглядно иллюстрирует процесс авторизации пользователя и вывода результата. Она демонстрирует последовательность шагов, которые происходят при попытке пользователя войти в систему.

На рисунке 3.13 изображена диаграмма компонентов приложения.



Рисунок 3.13 - Диаграмма компонентов приложения

Данная диаграмма компонентов наглядно демонстрирует компоненты, которые будут использованы для дальнейшей работы и функционировании приложения. Она предоставляет обзор основных компонентов, которые взаимодействуют между собой для обеспечения работоспособности приложения.

4 Создание (реализация) программного средства

Следующим этапом разработки приложения является непосредственная реализация программного решения в соответствии с уже сформированными требованиями и шаблонами.

4.1 Обобщённая структура приложения

Для создания высококачественного приложения необходимо построить диаграмму классов и определить обобщенную структуру. Этот этап разработки играет важную роль, так как позволяет получить ясное представление о компонентах приложения и их взаимосвязи.

Диаграммой классов представлена на рисунке 4.1.

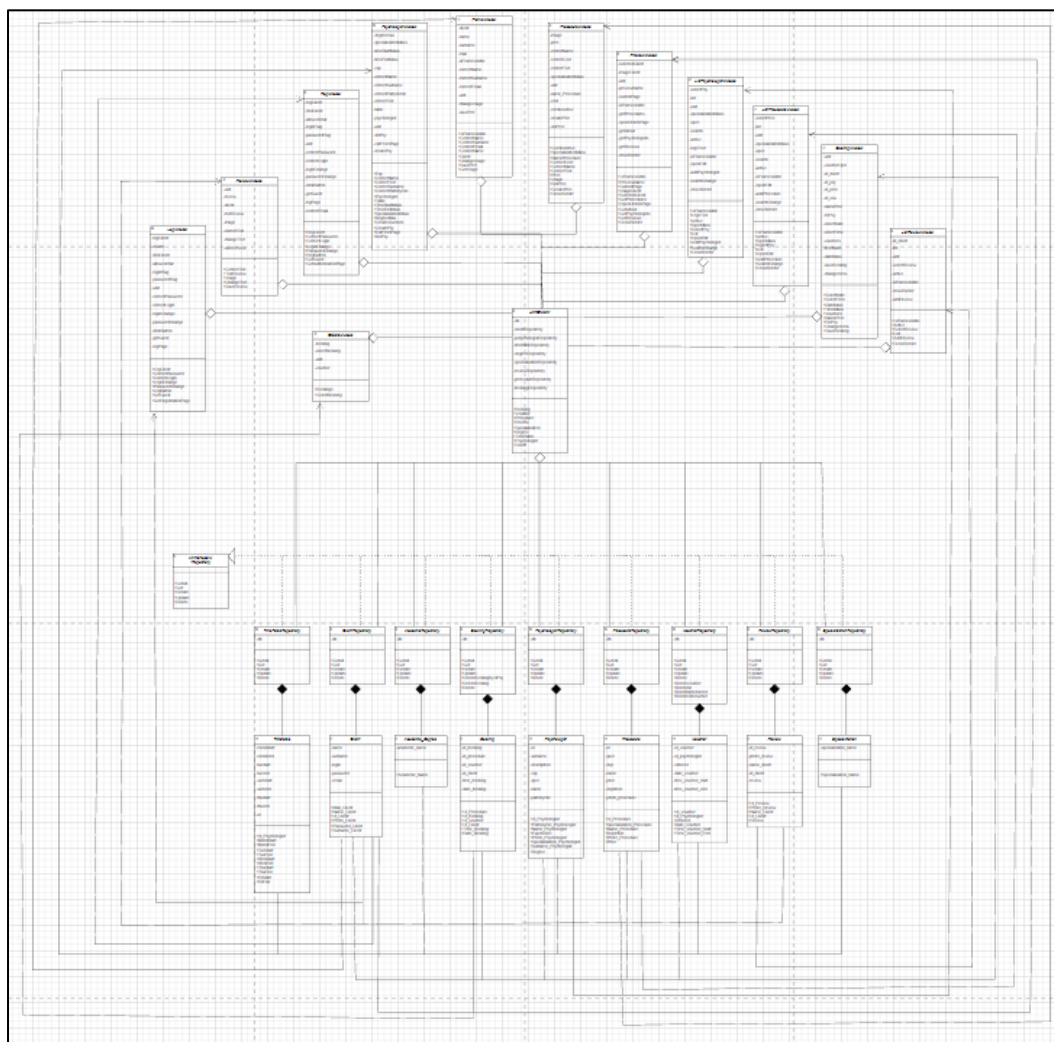


Рисунок 4.1 - Диаграмма классов

Данная диаграмма классов наглядно демонстрирует обобщенную структуру приложения. Она представляет собой визуальное отображение классов, их свойств и взаимосвязей, которые существуют внутри приложения.

Для более чёткого представления и детализации структуры пользовательского интерфейса в приложении необходимо составить таблицу с названиями XAML-файлами и их кратким содержанием. Данная файлы представлены в таблице 4.1.

Таблица 4.1 – Описание XAML-файлов

| Имя файла | Содержание |
|------------------------|---|
| Eng.xaml | Словарь ресурсов для смены языка в приложении на английский. |
| Rus.xaml | Словарь ресурсов для смены языка в приложении на русский. |
| Registration.xaml | Страница с формой регистрации нового аккаунта. |
| Authorization.xaml | Страница с формой авторизации. |
| OrdersPage.xaml | Страница с текущими сеансами клиента. |
| Profile.xaml | Страница профиля текущего клиента с возможностью изменения личных данных. |
| ReviewClient.xaml | Форма для создания отзыва. |
| AboutPage.xaml | Страница с ознакомительной информацией о психологическом центре. |
| GreetingPage.xaml | Приветствующая страница при входе в приложение. |
| ProcedurePage.xaml | Страница выбранной процедуры. |
| ProceduresPage.xaml | Страница со список всех существующих процедур в приложении. |
| PsychologistPage.xaml | Страница выбранного психолога. |
| PsychologistsPage.xaml | Страница со список всех существующих психологов в приложении. |
| ReviewsPage.xaml | Страница со всеми отзывами оставленных от клиентов. |
| Settings.xaml | Окно с настройками приложения. |
| WarningWindow.xaml | Окно с ошибкой. |
| WindowApp.xaml | Страница с навигационной панелью. |
| Style.xaml | Словарь ресурсов с набором стилей для элементов приложения. |
| OrderItem.xaml | Элемент списка заказов. |
| ProcItem.xaml | Элемент списка процедур. |
| PsyItem.xaml | Элемент списка психологов. |

Каждый из этих файлов является частью приложения и содержит разметку для соответствующей страницы или окна. Они играют важную роль в определении структуры и внешнего вида пользовательского интерфейса приложения.

4.2 Реализация архитектуры MVVM

Архитектура MVVM (Model-View-ViewModel) является важным и широко применяемым подходом при разработке приложений на платформе WPF.

Для реализации паттерна MVVM файлы программы были распределены по соответствующим директориям и реализовали соответствующие функции.

При использовании подхода DataBase First в разработке приложения, классы-модели были автоматически сгенерированы самим приложением на основе существующей базы данных. Разберём все модели:

1. Academic_Degree.cs – содержит информацию о учёной степени.
2. Booking.cs – содержит необходимую информацию для оформления сеанса.
3. Client.cs – содержит информацию о клиенте.
4. Procedure.cs – содержит информацию о процедуре.
5. Psychologist.cs – содержит информацию о психологе.
6. Review.cs – содержит информацию об отзыве.
7. Specialization.cs – содержит информацию о специализации.
8. Timetable.cs – содержит информацию о графике работы психолога.
9. Voucher.cs – содержит необходимую информацию о талоне.

Для реализации архитектуры MVVM (Model-View-ViewModel) необходимо создать классы, которые будут выступать в роли ViewModel.

ViewModel представляет прослойку между моделью (Model) и представлением (View). Его основная задача - предоставить данные и логику, необходимую для отображения и взаимодействия с моделью в пользовательском интерфейсе.

Данные классы находятся в папке MVVM, рассмотрим их поподробнее:

1. AppViewModel.cs – содержит необходимые команды для закрытия приложения.
2. BookingModel.cs – содержит необходимые команды для оформления сеанса.
3. GlobalContextModel.cs – содержит общедоступные команды для эффективного взаимодействия элементов в приложении и обработки данных.
4. GreetingModel.cs – содержит необходимые команды для перехода на другие страницы приложения.
5. ListProceduresModel.cs – содержит необходимые команды для работы со списком процедур.
6. ListPsychologistsModel.cs – содержит необходимые команды для работы со списком психологов.
7. ListReviewModel.cs – содержит необходимые команды для работы со списком отзывов.
8. LogModel.cs – содержит команды необходимые для авторизации пользователя.
9. PreviewModel.cs – содержит команды необходимые для навигационного меню.
10. ProcedureModel.cs – содержит необходимые команды для добавления, изменения и удаления процедуры.

11. ProfileModel.cs – содержит команды необходимые для изменения личной информации пользователя.

12. PsychologistModel.cs – содержит команды для добавления, изменения, удаления психолога.

13. RegModel.cs – содержит команды необходимые для добавления нового пользователя.

14. ReviewModel.cs – содержит команды для добавления нового отзыва.

15. SettingsModel.cs – содержит команды для управлением стилевого оформления приложения, а также для смены языка в приложении.

Для реализации архитектуры MVVM (Model-View-ViewModel) необходимо создать страницы, которые будут выступать в роли представления (View).

Представление (View) отображает пользовательский интерфейс и взаимодействует с пользователем. В контексте MVVM, представление является пассивным и не содержит бизнес-логики. Его основная задача - отображать данные, полученные из ViewModel, и передавать пользовательские действия обратно в ViewModel для обработки.

Данные страницы были описаны выше в таблице 4.1.

4.3 Реализация паттерна Repository и UnitOfWork

Реализация паттерна Repository и UnitOfWork является важной в разработке приложений, особенно при работе с базами данных. Эти паттерны помогают организовать доступ к данным и обеспечить эффективное управление транзакциями.

Паттерн Repository реализуется для всех классов-моделей в приложении, чтобы абстрагироваться от работы с различными видами баз данных.

Рассмотрим текущие классы-репозитории в приложении:

1. BookingRepository.cs – данный класс реализует методы для работы с моделью Booking.

2. ClientRepository.cs – данный класс реализует методы для работы с моделью Client.

3. DegreeRepository.cs – данный класс реализует методы для работы с моделью Degree.

4. ProcedureRepository.cs – данный класс реализует методы для работы с моделью Procedure.

5. PsychologistRepository.cs – данный класс реализует методы для работы с моделью Psychologist.

6. ReviewRepository - данный класс реализует методы для работы с моделью Review.

7. SpecializationRepository - данный класс реализует методы для работы с моделью Specialization.

8. TimeTableRepository - данный класс реализует методы для работы с моделью Timetable.

9. VoucherRepository - данный класс реализует методы для работы с моделью Voucher.

Далее рассмотрим реализацию паттерна UnitOfWork в приложении “Clean Brain”. Данный паттерн тесно взаимодействует с классами-репозиториями и служит общей точкой для доступа к репозиториям, а также объединяет все изменения в единую транзакцию.

Класс реализующий паттерн UnitOfWork называется “UnitOfWork” и содержит все вышеописанные классы-репозитории.

4.4 Реализация паттерна Command

Паттерн Command является важным в разработке приложений, поскольку он обеспечивает отделение логики действия от элементов пользовательского интерфейса и позволяет управлять выполнением операций в ответ на пользовательские действия.

В приложении реализован Relay Command, который достаточно популярен своим использованием в приложениях написанный на WPF.

Класс реализующий паттерн Command называется “RelayCommand” и содержит следующие методы:

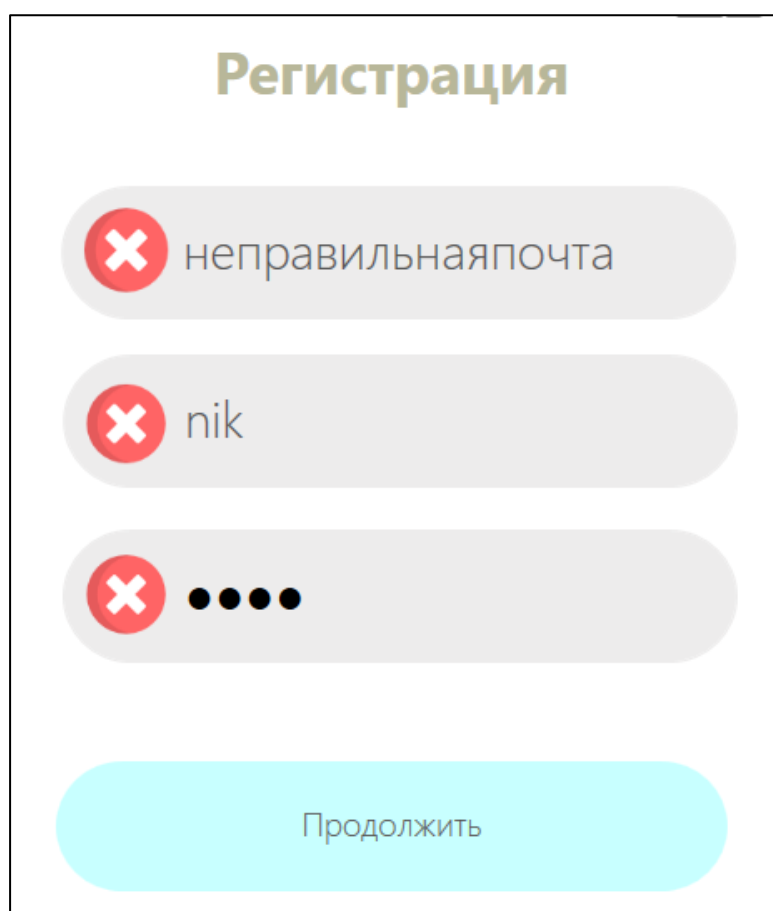
1. CanExecute – данный метод проверяет, может ли данная команда выполниться прямо сейчас.
2. Execute – данный метод определяет метод который будет вызываться при срабатывании команды.

5 Тестирование работоспособности и анализ полученных результатов

Данное программное средство тестировалось вручную, были выполнены все технические моменты, предусмотренные реализацией, а также были предприняты попытки нарушить работу приложения.

В курсовом проекте задействуется обработка исключений, таким образом, что пользователь будет уведомлен о неудачном выполнении операции. Присутствуют различные всплывающие окна, выводящие сообщение об ошибке, также присутствует, а также различные изображения.

В момент регистрации возможна ситуация, когда пользователь вводит уже существующий логин или же просто некорректно вводит электронную почту, либо пароль. На рисунке 5.1 продемонстрирована обработка данных исключения.



The image shows a registration form titled "Регистрация" (Registration). It contains three input fields, each with a red "X" icon indicating an error:

- The first field contains the text "неправильная почта" (incorrect email).
- The second field contains the text "nik".
- The third field contains four black dots, representing a masked password.

Below the input fields is a large cyan button with the text "Продолжить" (Continue).

Рисунок 5.1 - Неверно введен логин, пароль, а также почта

При регистрации нового пользователя его данные заносятся в таблицу Client в базе данных.

Когда пользователь вводит свои учетные данные и нажимает кнопку "Войти" или аналогичное действие, приложение проверяет введенные данные на соответствие с данными, хранящимися в базе данных. Если соответствующая запись не найдена, возникает ситуация ошибки авторизации.

В такой ситуации приложение открывает диалоговое окно с ясным сообщением об ошибке.

Возможная ситуация продемонстрирована на рисунке 5.2.

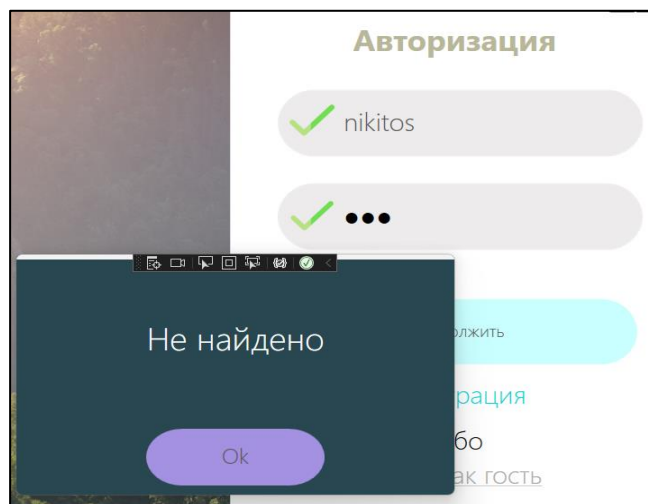


Рисунок 5.2 - Не найден пользователь с такими данными

После успешного входа в приложение и авторизации пользователь получает возможность изменить свой профиль. Это позволяет пользователям настраивать свои личные данные и предоставлять актуальную информацию о себе.

При изменении своих данных в профиле, вводимая информация проходит постоянную проверку, в случае неправильного ввода информации, текст сменит свой цвет на красный.

Данная ситуация изображена на рисунке 5.3.

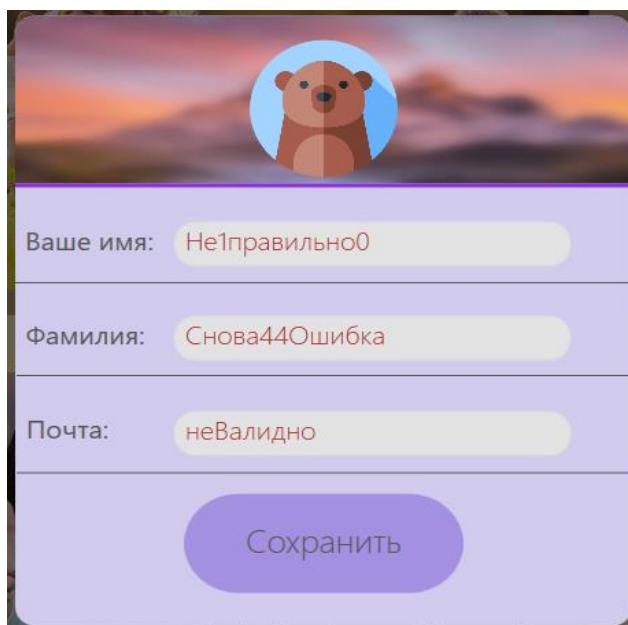


Рисунок 5.3 - Неправильно заполнены поля в профиле

Когда пользователь пытается ввести отзыв и введенный текст не соответствует требованиям, приложение может немедленно изменить цвет текста на красный. Это действие является визуальным индикатором ошибки и предупреждает пользователя о том, что введенные данные не валидны.

Данная ситуация изображена на рисунке 5.4.

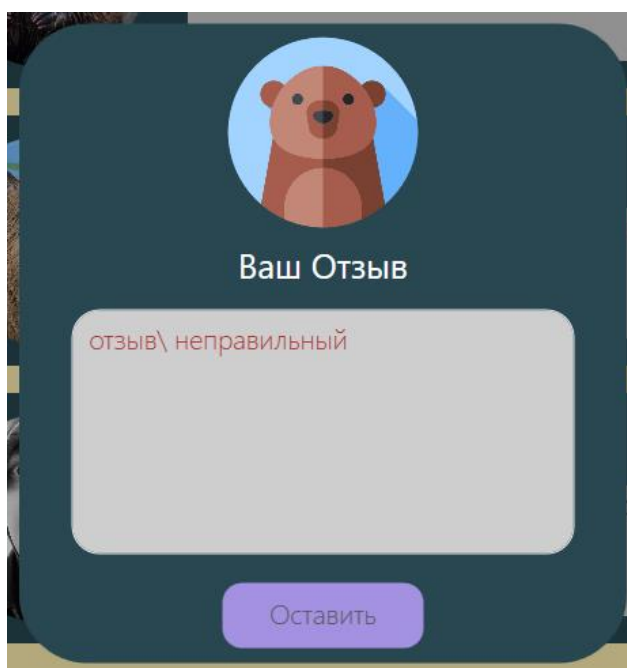


Рисунок 5.4 - Некорректный отзыв клиента

В случае, когда пользователь пытается оформить услугу и не заполнил некоторые обязательные поля, приложение обнаружит эту неполадку и выведет соответствующее сообщение.

Ситуация изображена на рисунке 5.5

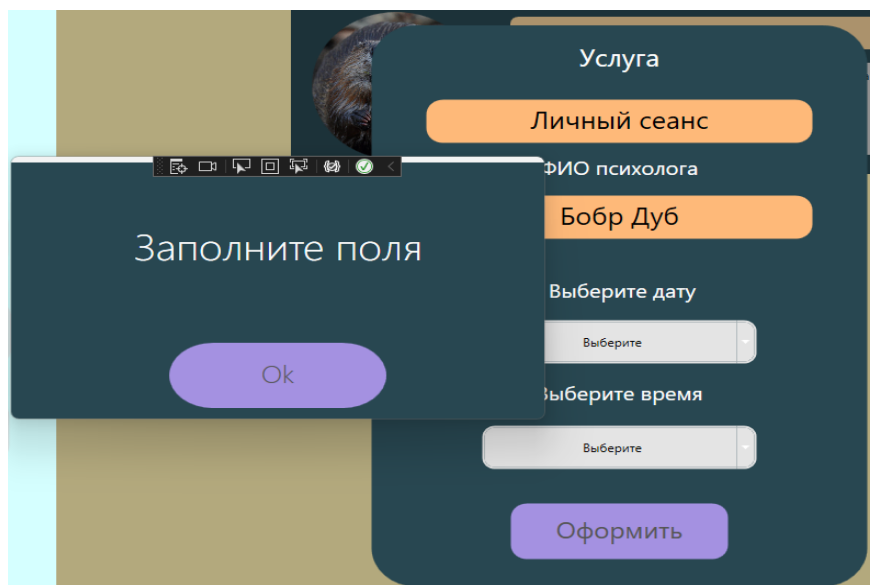


Рисунок 5.5 - При оформлении услуги, пользователь не заполнил поля

Подобные случаи могут случиться и у администратора, поэтому рассмотрим несколько из них.

В случае, если администратор решит добавить психолога, приложение предоставит форму для заполнения. Если введённая информация не валидна, текст

в соответствующих полях изменить своё отображение на красный, изображено на рисунке 5.6.

The screenshot shows a registration form for a psychologist. On the left, there is a profile picture of a bear and input fields for 'Имя' (Name), 'Фамилия' (Surname), and 'Отчество' (Patronymic). The 'Имя' field contains '234ИмяНеВерно', 'Фамилия' contains 'ФамилияТОжеНеверна', and 'Отчество' contains '1Отчество1'. Below these is a large 'Описание' (Description) field with a red error message: '*№;%Описание не валидно'. On the right, there are dropdown menus for days of the week (Понедельник, Вторник, Среда, Четверг, Пятница) with 'Choose' and 'До: Выберите' options. Below these is a 'Стаж работы' (Work Experience) field with a 'число' (number) input and two 'Выберите' (Choose) dropdowns. At the bottom right is a purple 'Создать' (Create) button.

Рисунок 5.6 - Введённая информация в поля карточки психолога не валидна

Когда администратор пытается создать нового психолога в приложении, возможна ситуация, когда введенные данные уже соответствуют существующему психологу в базе данных, поэтому получит соответствующую ошибку, которая изображена на рисунке 5.7.

This screenshot shows the same registration form as Figure 5.6, but with an error dialog box overlaid. The dialog box has a dark blue background and white text that reads 'Врач с таким ФИО существует' (A doctor with this FIO exists). Below the text is a purple 'Ok' button. The background form is partially obscured by the dialog box, but the 'Создать' (Create) button is still visible at the bottom right.

Рисунок 5.7 - Врач с такими данными существует

Подобная ситуация обстоит и с добавлением новой услуги, когда пользователь вводит данные в поля формы услуги и эти данные не проходят проверку на

валидность, приложение может визуально отобразить ошибку, изменяя отображение текста на красный, изображено на рисунке 5.8.

The screenshot shows a form for creating a service card. It is divided into two main sections: a left section for basic information and a right section for a description.

- Left Section:**
 - Image:** A placeholder image of a family sitting in a field.
 - Название (Name):** The input field contains "Что-то тут явно не верно" (Something is clearly wrong here), with the text highlighted in red.
 - Стоимость (Cost):** The input field contains "число" (number), with the text highlighted in red.
 - Выберите (Select):** A dropdown menu is currently empty.
- Right Section:**
 - Описание (Description):** A large text area containing the text "№;%Так не надо делать" (№;% This shouldn't be done), with the text highlighted in red.
 - Создать (Create):** A purple button at the bottom right.

Рисунок 5.8 - Введённая информация в поля карточки услуги не валидна

При попытке администратора создать новую процедуру в приложении, возможна ситуация, когда введенные данные уже соответствуют существующей услуге в базе данных, поэтому получит соответствующую ошибку, которая изображена на рисунке 5.9.

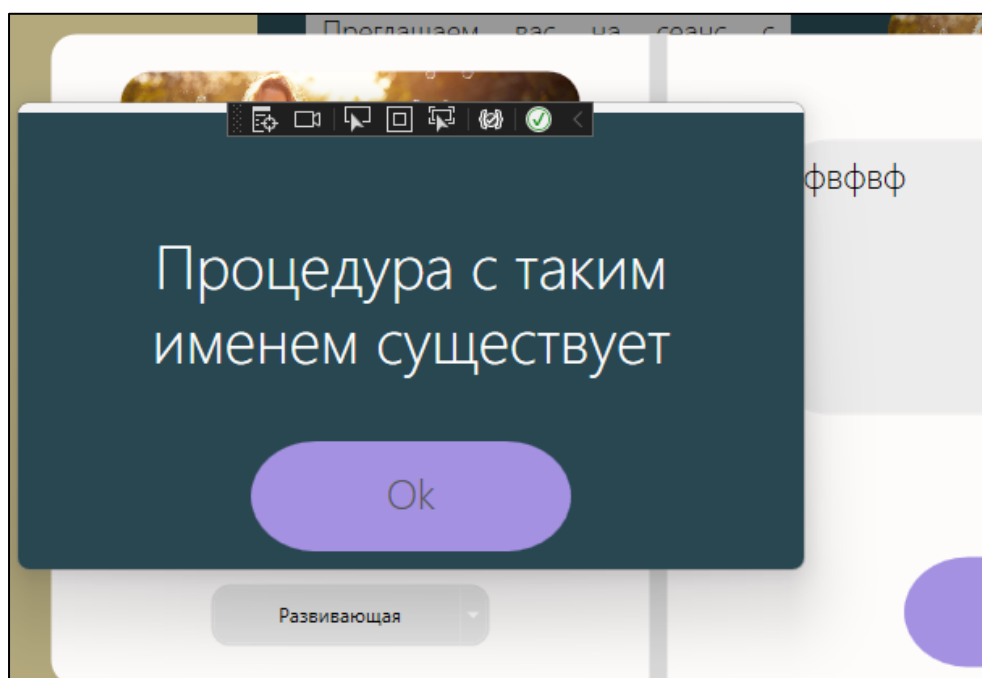


Рисунок 5.9 - Услуга с такими данными существует

Обработка ошибок так же предусмотрена и в режиме гостя. Возможна такая ситуация, что гость захочет войти в несуществующий профиль, просмотреть

текущие сеансы, оформить услугу, но данные действия будут вызывать сообщение, изображенное на рисунке 5.10.

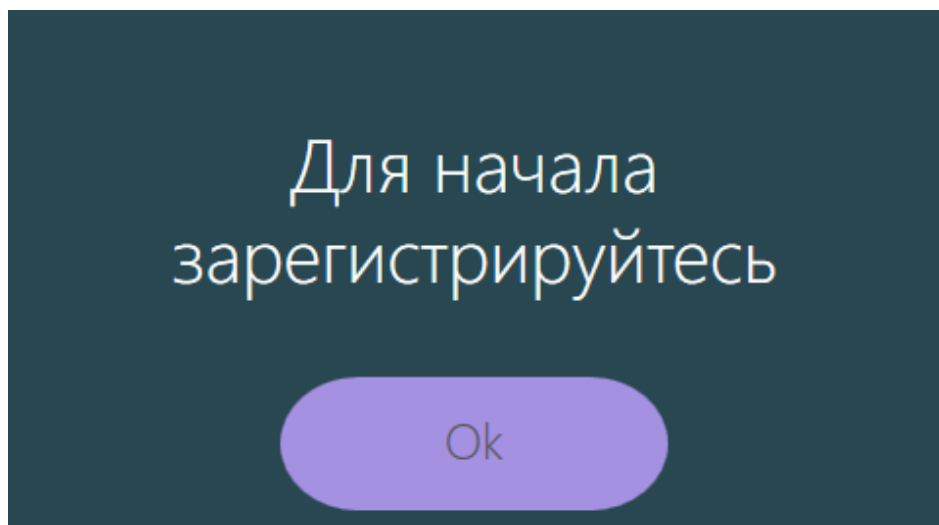


Рисунок 5.10 - Гость попытался выполнить действие доступное только авторизованному пользователю

Были продемонстрированы все возможные ситуации неверной эксплуатации программного средства пользователем, проведена проверка работоспособности и проанализированы полученные результаты.

6 Руководство по использованию

Окно регистрации содержит поля для ввода учетных данных, таких как электронная почта и пароль. Пользователь может использовать эти поля для авторизации, если у него уже есть учетная запись. После ввода данных пользователь нажимает на кнопку "Войти" или аналогичную, чтобы авторизоваться в приложении.

В случае, если пользователь не имеет учетной записи или не хочет создавать новую, он может выбрать опцию "Гость" или "Войти как гость". При выборе этой опции приложение предоставляет пользователю ограниченные возможности функционала, не требуя от него создания аккаунта.

Форма регистрации представлена на рисунке 6.1.

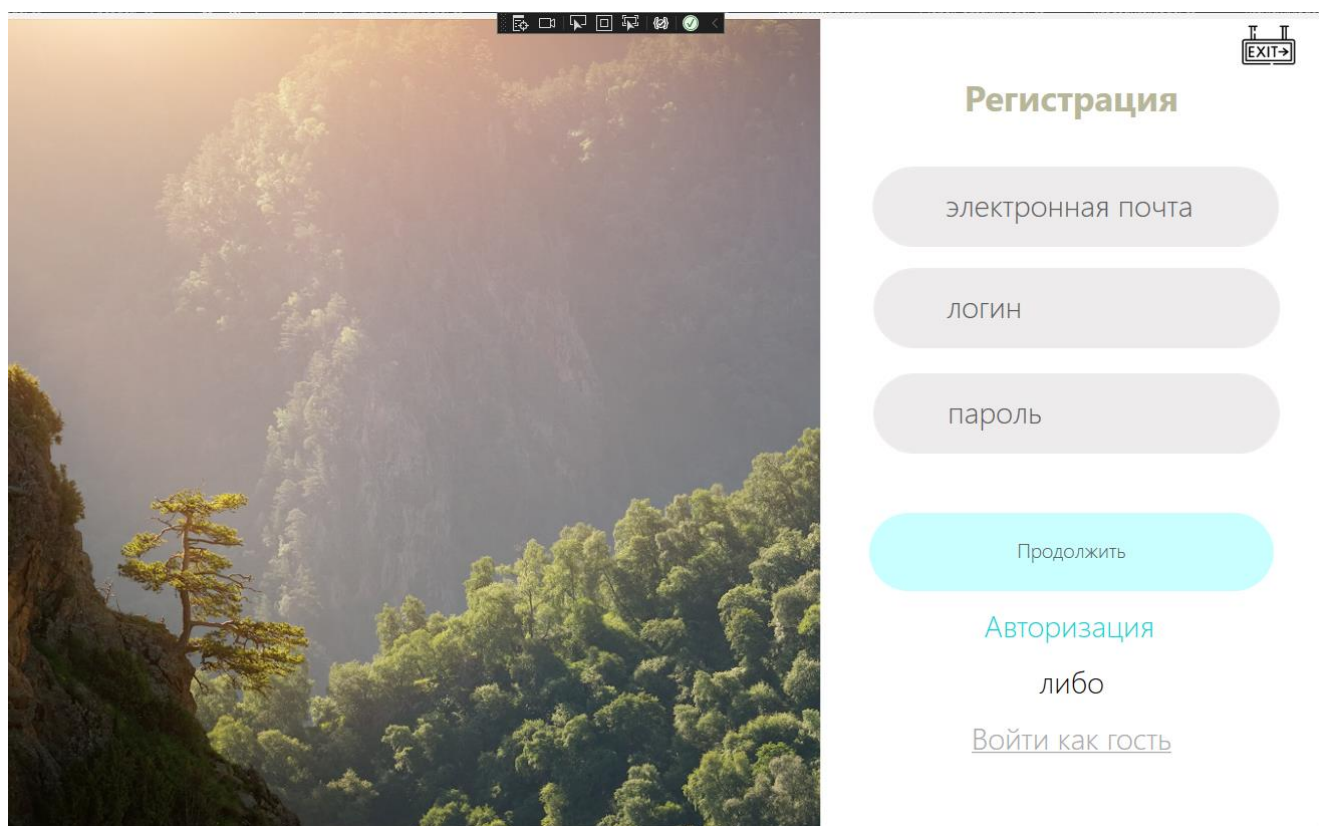
The image shows a mobile application registration screen. On the left is a vertical landscape photo of a forested mountain. On the right is a white registration form. At the top right of the form is an 'EXIT' button. The title 'Регистрация' is in green. Below it are three light gray rounded input fields labeled 'электронная почта', 'логин', and 'пароль'. A bright cyan button labeled 'Продолжить' is below these. Underneath is the word 'Авторизация' in cyan, followed by 'либо' in gray, and a link 'Войти как гость' in gray.

Рисунок 6.1 - Окно регистрации

Если у пользователя имеется личный аккаунт в приложении, ему всегда доступна авторизация.

При выборе опции авторизации, пользователю предлагается ввести свой логин и пароль в соответствующие поля. Пароль вводится в отдельное поле в зашифрованном виде для обеспечения безопасности.

При вводе данных пользователь может проверить являются ли они валидными, а именно будет отображаться индикатор в виде галочки, что говорит о валидности данных, а также крестик, что обозначает противоположный результат.

После заполнения полей логина и пароля пользователь нажимает на кнопку "Войти", чтобы произвести авторизацию в приложении.

Форма авторизации представлена на рисунке 6.2.

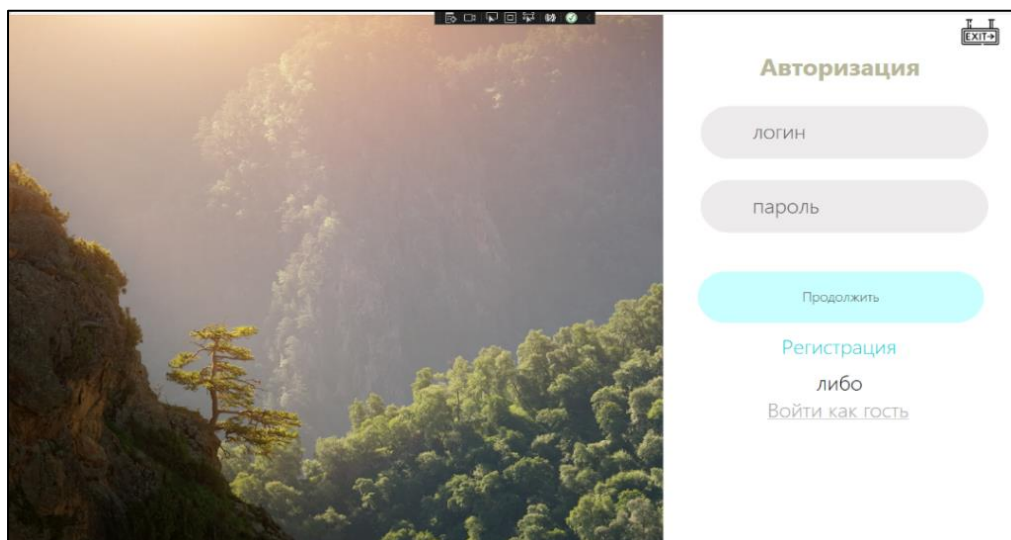


Рисунок 6.2 - Окно авторизации

Данные формы общие как для гостя, администратора и для авторизованного пользователя.

6.1 Руководство по использованию администраторам

После входа в приложения администратора встречается приветствующая страница, а также навигационное меню, расположенное вертикальном положении. Данная страница изображена на рисунке 6.3.

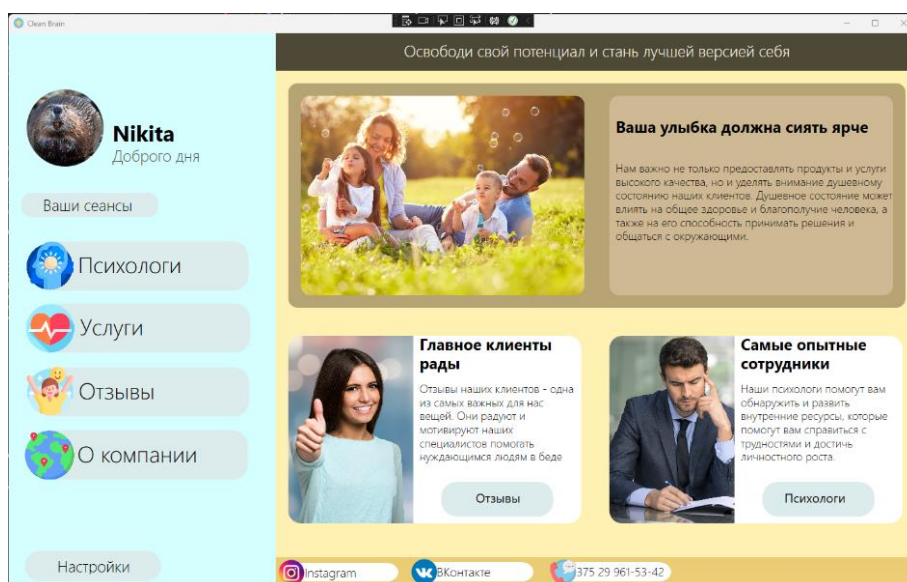


Рисунок 6.3 - Приветствующая страница

Нажав на кнопку “Психологи” мы переходим к общему перечню психологов, которых мы можем отсортировать по возможным специализациям. Так же для

администратора имеется дополнительная кнопка “добавить”, с помощью которого можно добавить нового психолога в базу данных. Представлено на рисунке 6.4.

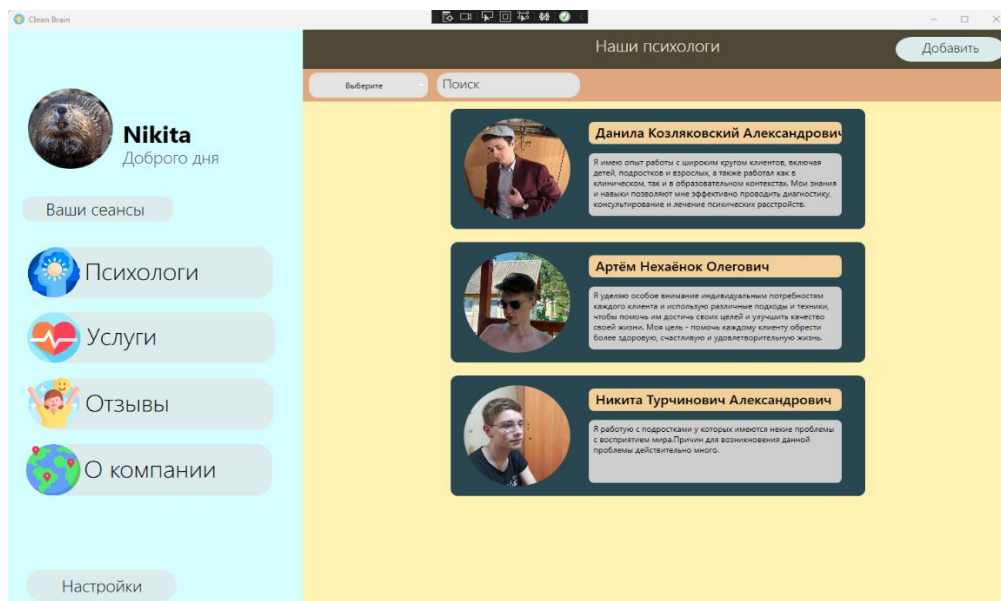


Рисунок 6.4 - Страница со списком психологов

При нажатии на кнопку “добавить”, администратор попадает в новое окно, с помощью которого может добавить психолога в базу данных. При добавлении и выхода из страницы с существующим психологом, талончики автоматически обновляются. Данная форма представлена на рисунке 6.5.

Рисунок 6.5 - Страница добавления психолога

При клике на изображение психолога из списка, администратору открывается окно с заполненными данными доступные для редактирования. Также

присутствует кнопка “удалить”, которая удалит текущего психолога из базы данных.

Окно редактирования психолога представлено на рисунке 6.6.

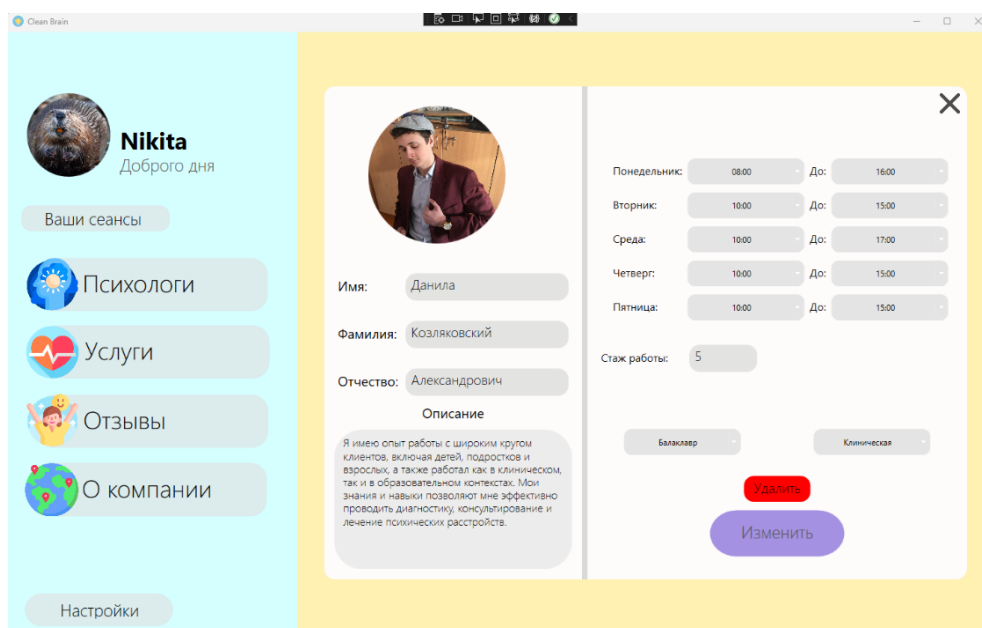


Рисунок 6.6 - Страница редактирования психолога

Далее нажав на кнопку “Услуги”, мы попадаем на страницу, где выдается перечень услуг, которые так же можно сортировать и добавлять новые в базу данных, представлено на рисунке 6.7.

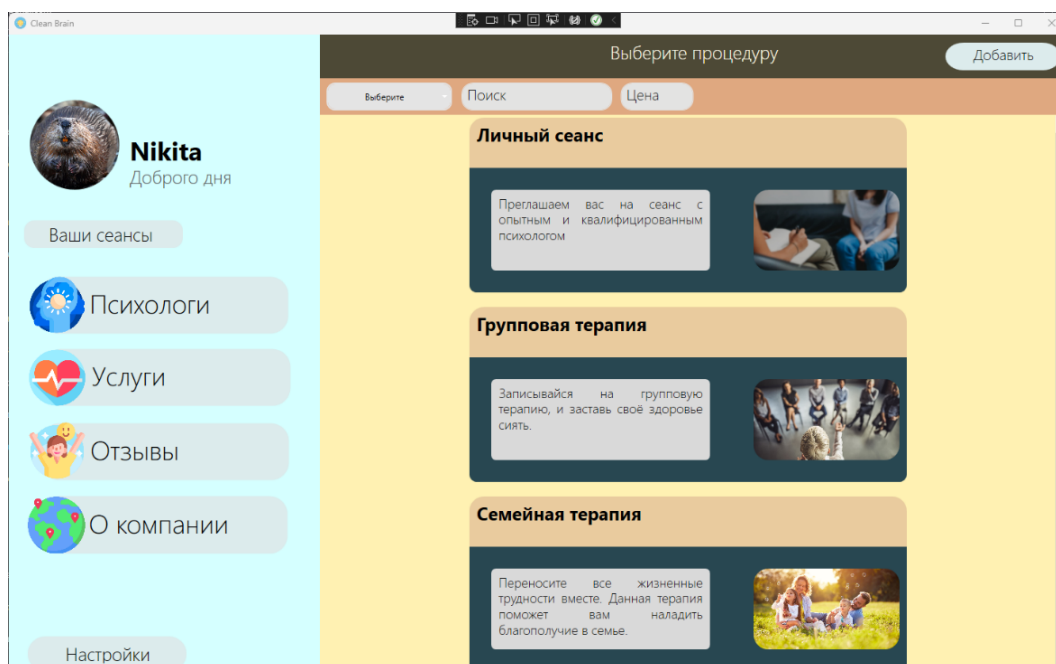


Рисунок 6.7 - Страница с перечнем услуг

Нажав на кнопку “добавить”, администратор переходит в новое окно, в которой находится форма для добавления новой услуги. В данной форме

необходимо заполнить обязательные поля, для дальнейшего добавления в базу данных, представлено на рисунке 6.8.

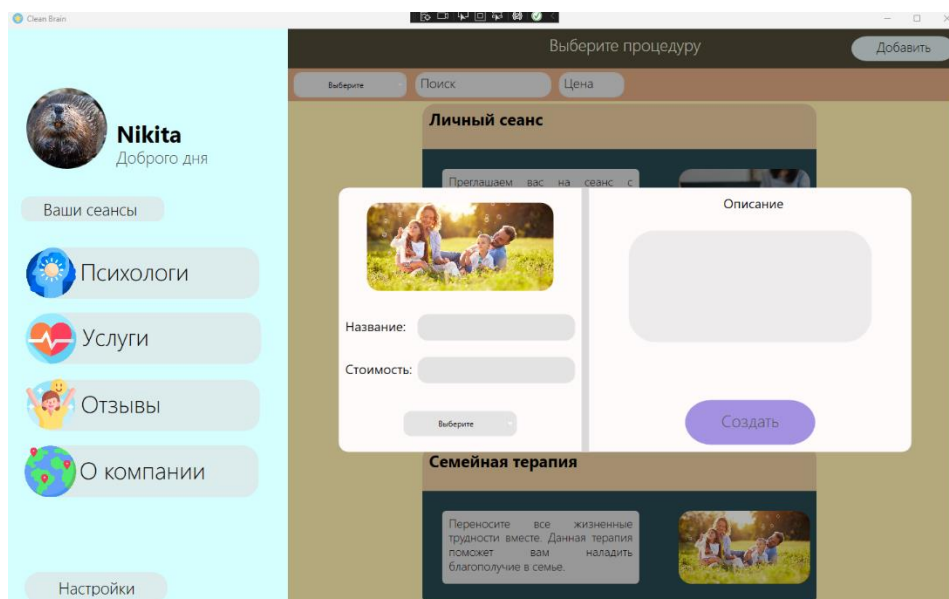


Рисунок 6.8 - Форма для добавления услуги

Для изменение текущей услуги, администратор должен нажать на изображение услуги из перечня всех услуг. Администратору вновь открывается форма с заполненными полями доступных для изменения. Так же на форме доступна кнопка “удалить”, которая удалит услугу из базы данных.

Данная форма представлена на рисунке 6.9.

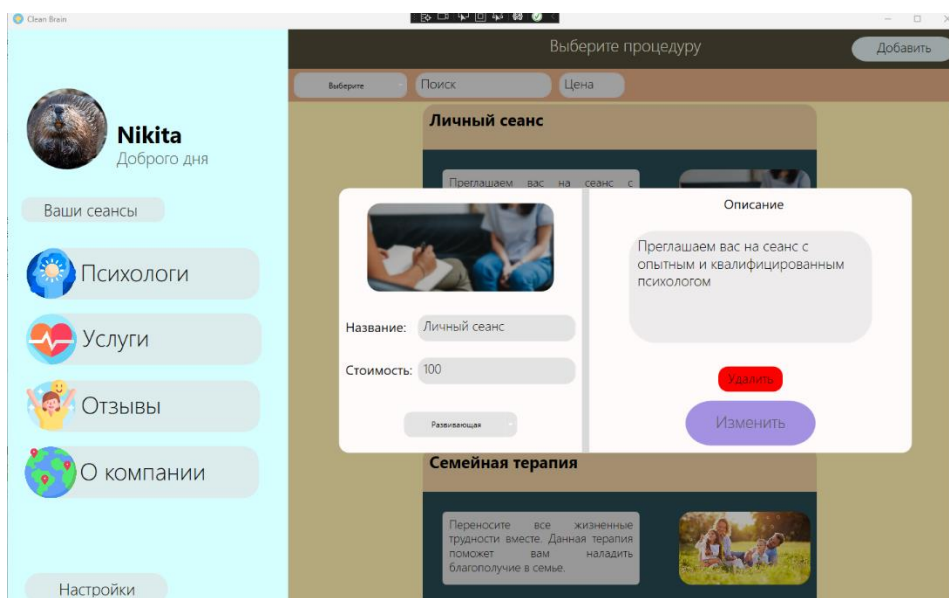


Рисунок 6.9 - Форма для изменения услуги

При выборе администратором пункта "Отзывы" в навигационном меню, открывается страница с отзывами от клиентов.

В случае, если администратор считает, что какой-либо отзыв не соответствует правилам, он может нажать на крестик около отзыва. При этом отзыв будет удален из базы данных.

Страница с отзывами представлена на рисунке 6.10.

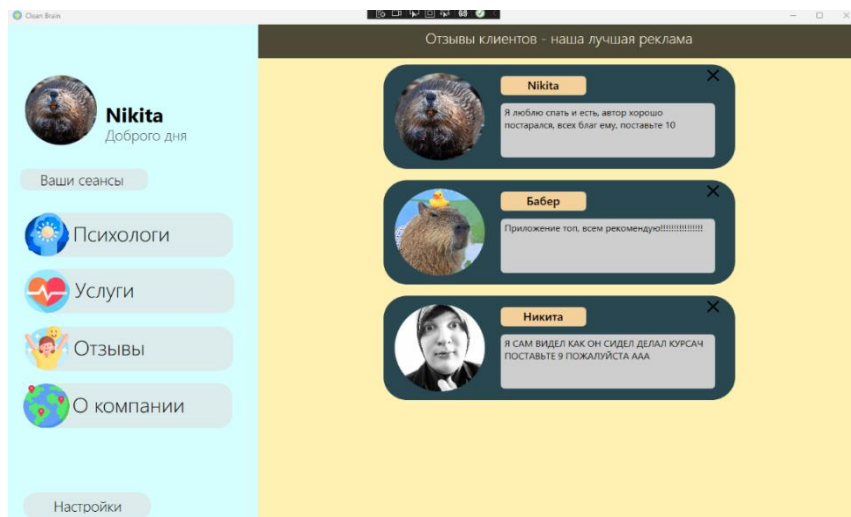


Рисунок 6.10 - Страница отзывов

Администратор не может оставлять отзывы.

6.2 Руководство по использованию пользователем

После успешной авторизации, пользователя встречает приветствующая страница с навигационным меню. Данное меню обеспечивает пользователю быстрый доступ к различным функциям приложения, позволяя ему легко перемещаться по различным страницам, что представлено на рисунке 6.11.

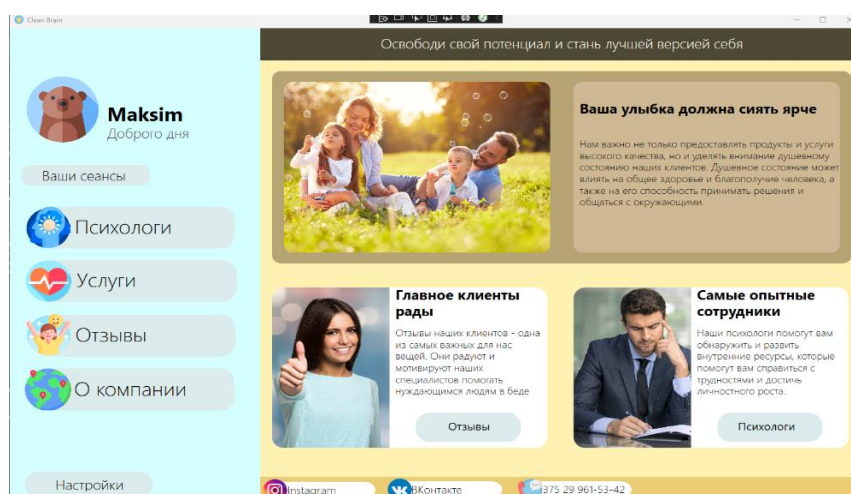


Рисунок 6.11 - Приветствующая страница

Если пользователь решит нажать на изображение аккаунта, то ему откроется новая страница с возможностью изменения личных данных, а также изображения профиля.

Если пользователь захочет выйти из данной страницы, он может кликнуть на тёмный фон.

Данная страница представлена на рисунке 6.12.

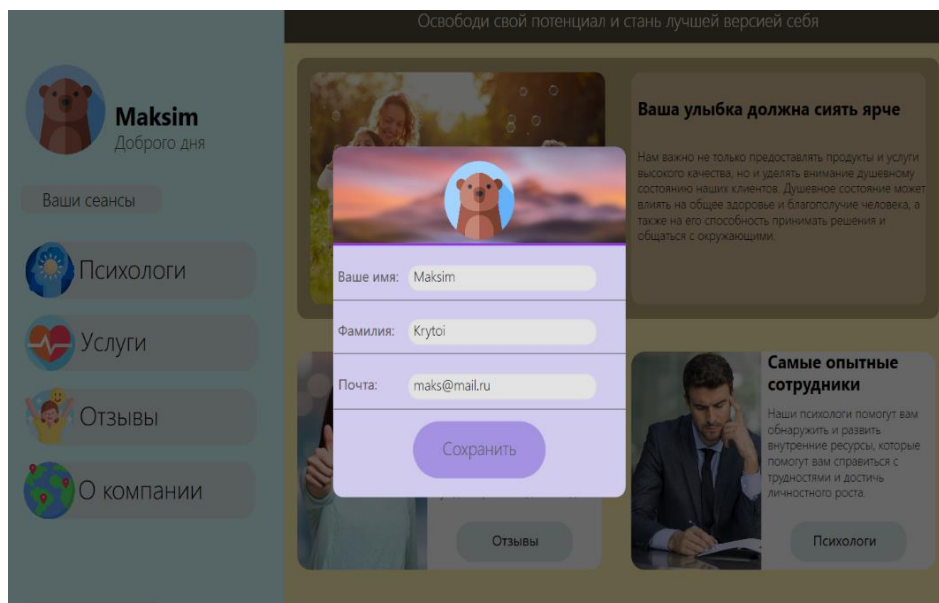


Рисунок 6.12 - Профиль пользователя

Нажав на кнопку “Психологи” мы переходим к общему перечню психологов, которых мы можем отсортировать по специализации. Как видно, кнопки для администратора скрылись. Представлено на рисунке 6.13.

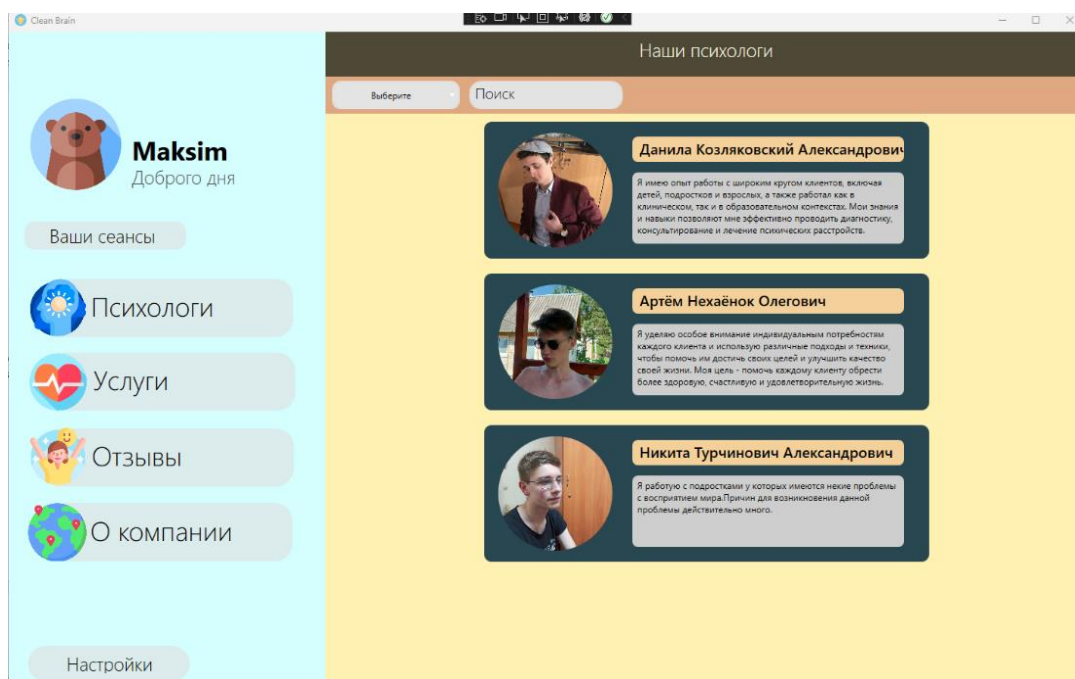


Рисунок 6.13 - Страница с перечнем психологов

При клике на изображение психолога из списка, пользователю открывается окно с заполненными данными для более детального ознакомления.

Профиль психолога представлен на рисунке 6.14.

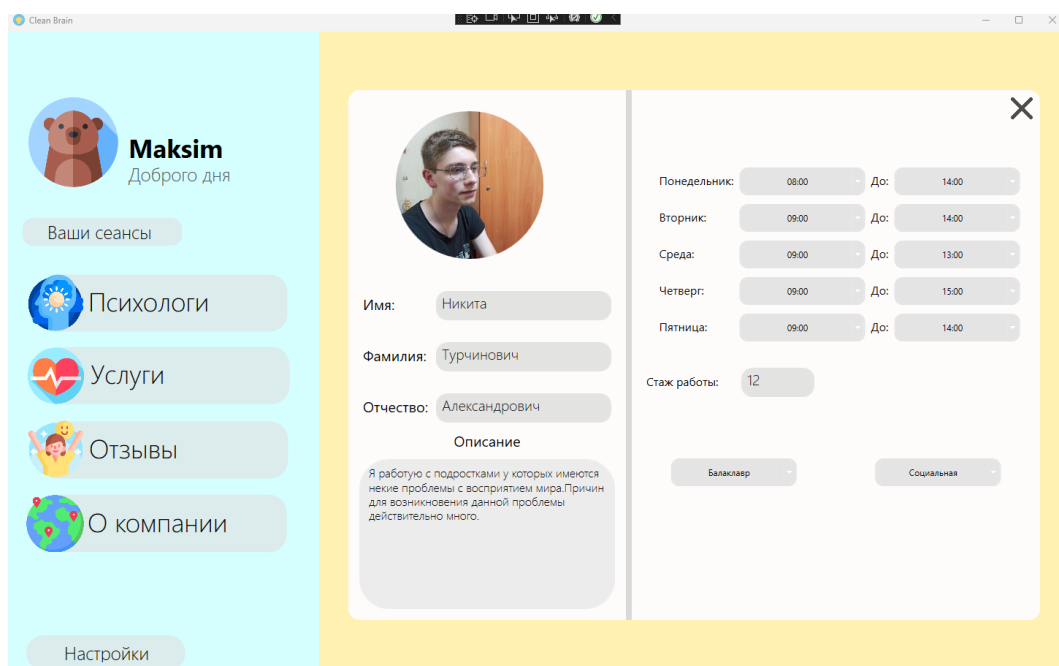


Рисунок 6.14 - Профиль психолога

Если пользователь нажмёт на карточку психолога, приложение откроет страницу с выбором услуг, которые может предоставить выбранный психолог, что представлено на рисунке 6.15.

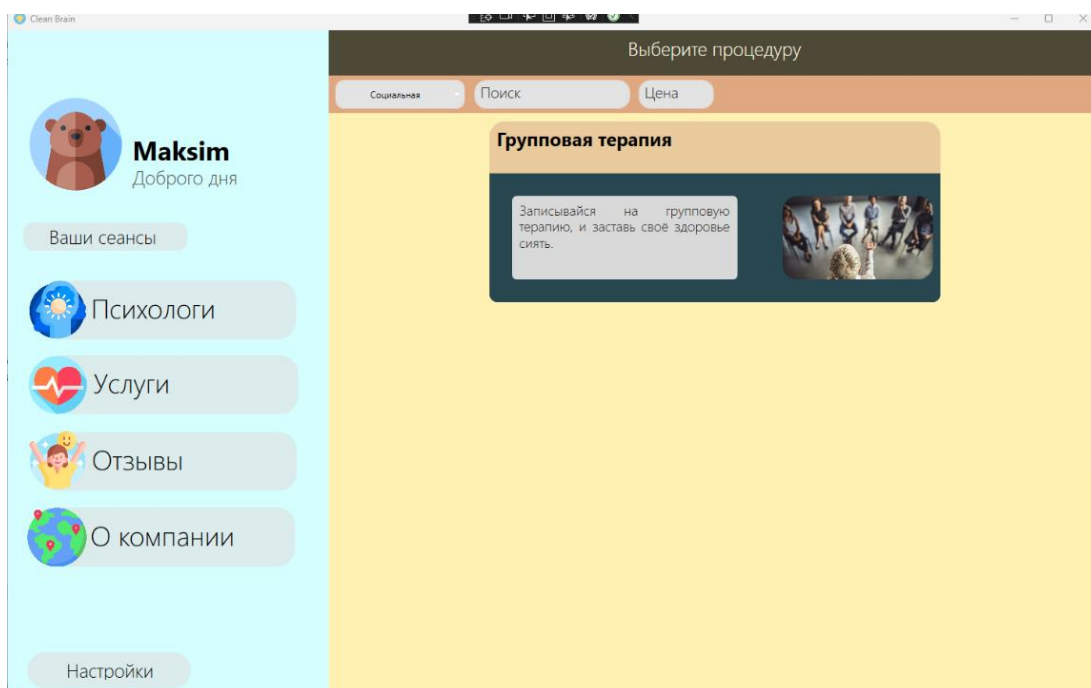


Рисунок 6.15 - Страница с услугами психолога

Далее пользователь может выбрать услугу, просто кликнув на её карточку. После данного действия приложение предоставит форму для оформления выбранной услуги.

Данная форма представлена на рисунке 6.16.

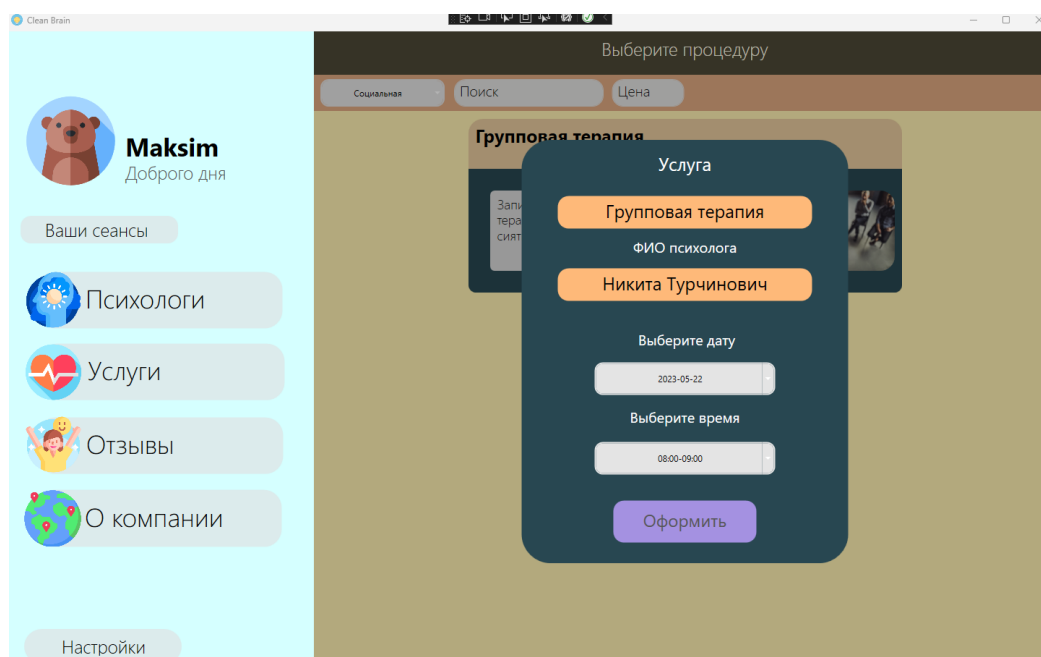


Рисунок 6.16 - Оформление выбранной услуги

После успешного оформления услуги пользователь, может просмотреть все текущие услуги просто нажав на кнопку “Ваши сеансы”. Пользователь перейдёт на список текущих заказанных услуг с возможностью их удаления, что представлено на рисунке 6.17.

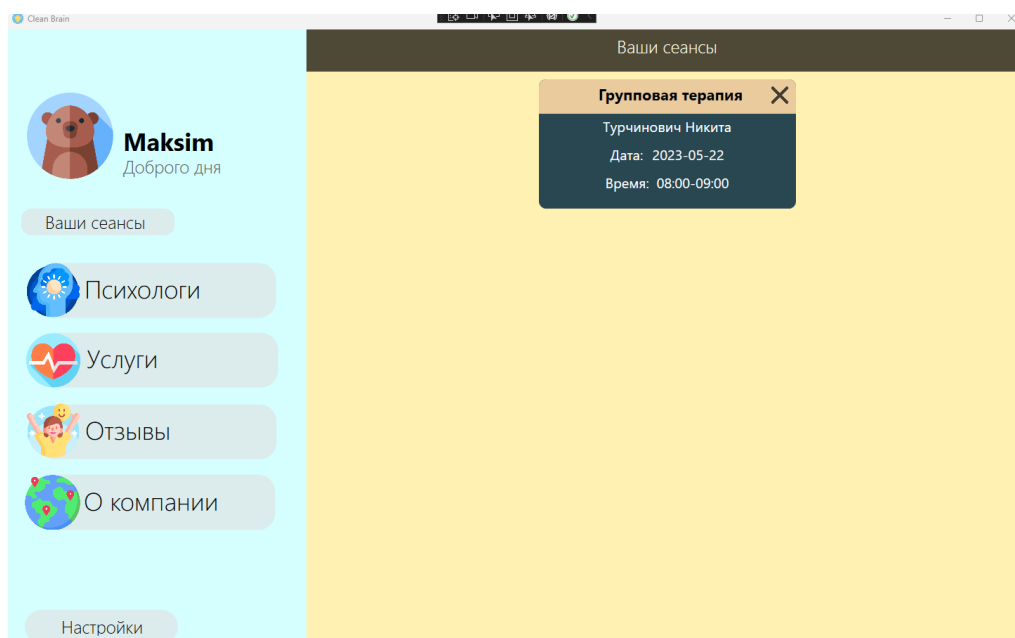


Рисунок 6.17 - Страница с текущими сеансами

Пользователь также имеет возможность нажать на кнопку "Отзывы", которая предоставит ему доступ к отзывам других пользователей. Это позволит

пользователю получить дополнительную информацию и оценить качество предоставляемых услуг.

Страница с отзывами от клиентов представлена на рисунке 6.18.

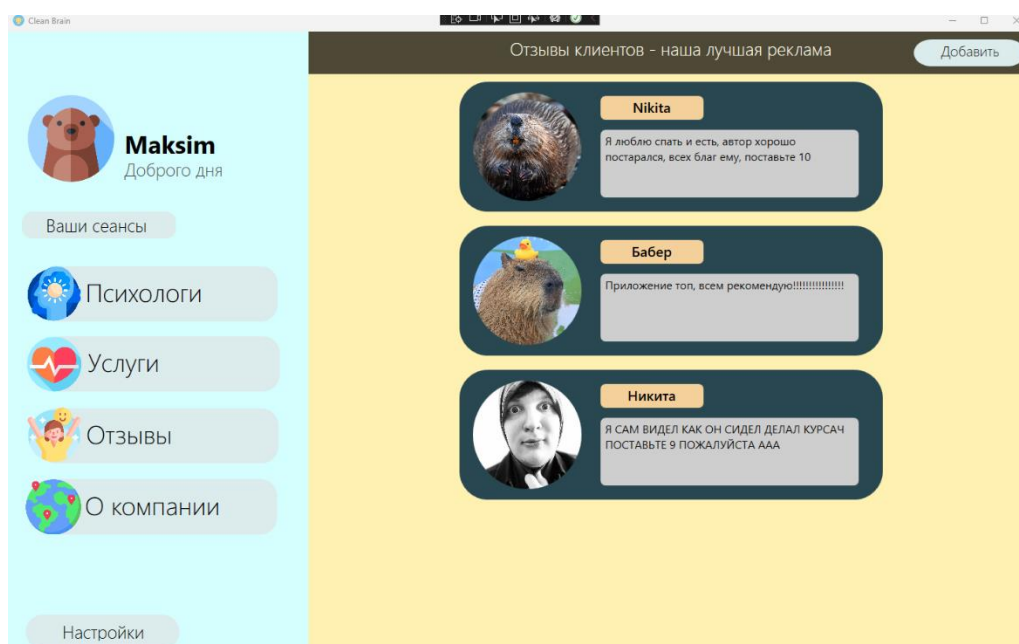


Рисунок 6.18 - Страница с отзывами

На данной странице пользователь может нажать на кнопку “добавить” и откроется форма для оформления отзыва, представленная на рисунке 6.19.

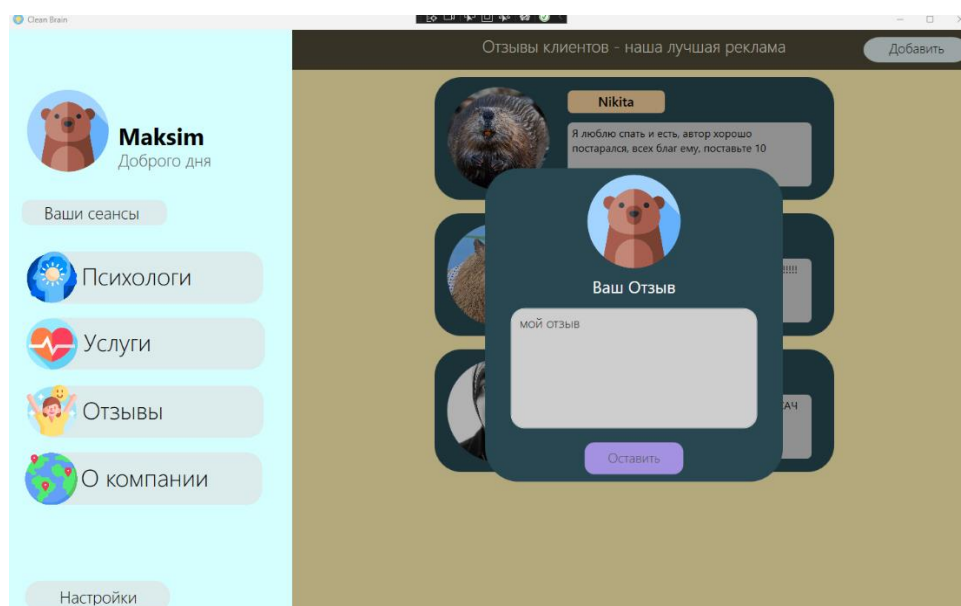


Рисунок 6.19 - Форма отзыва клиента

При заполнении формы отзыва, введенные пользователем символы проходят через процесс проверки. Это позволяет предотвратить ввод недопустимых символов.

6.3 Руководство по использованию гостем

Также как и администратор, гость попадает на приветствующее окно, после чего он может выбрать интересующие страницы приложения, путем нажатия на соответствующие кнопки в навигационном меню.

Приветствующая страница гостя представлена на рисунке 6.20.

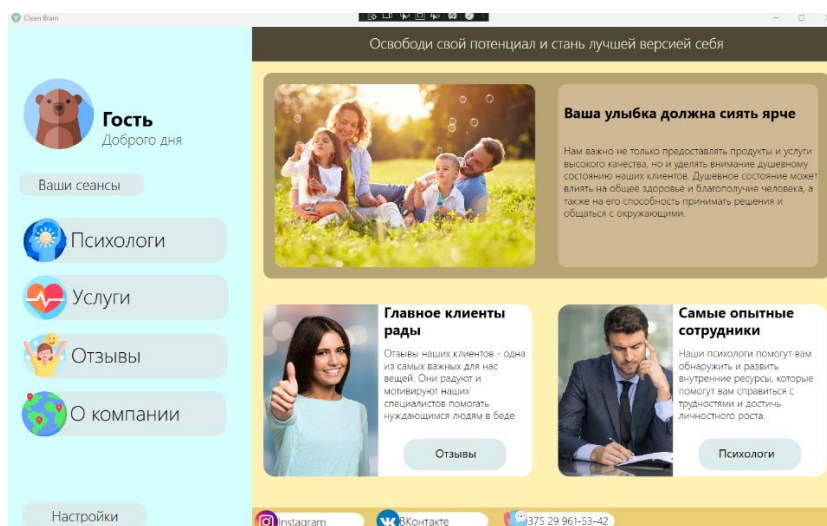


Рисунок 6.20 - Приветствующая страница

Далее гость может пройти к следующему пункту меню - "О компании". Нажав на этот пункт, гость получит доступ к информации о компании, включая ее ценности, миссию и другие важные сведения. Информация позволит гостю получить более полное представление о компании.

Данная страница представлена на рисунке 6.21.

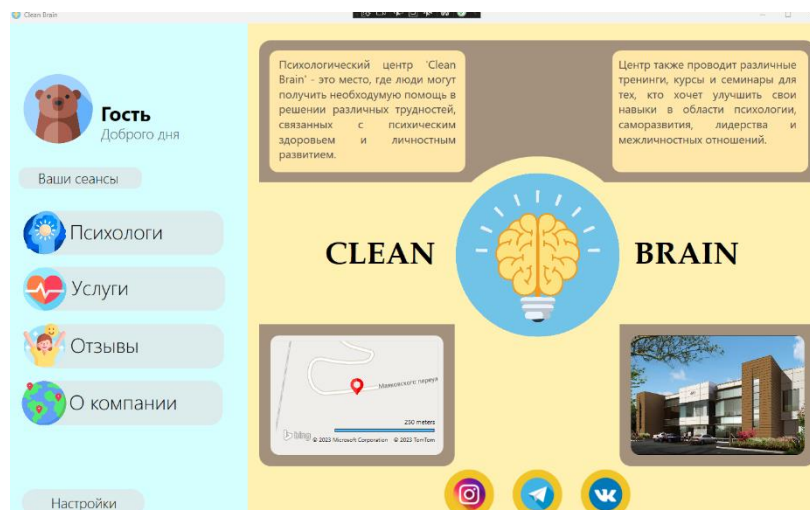


Рисунок 6.21 - Страница о компании

Гость также как и авторизованный пользователь, может просматривать список психологов, услуг, а также их страницы. Но при попытке совершить заказ,

изменить профиль и нажать на кнопку “Ваши сеансы”, гость получит сообщение призывающее его зарегистрироваться.

Данное сообщение представлено на рисунке 6.22.

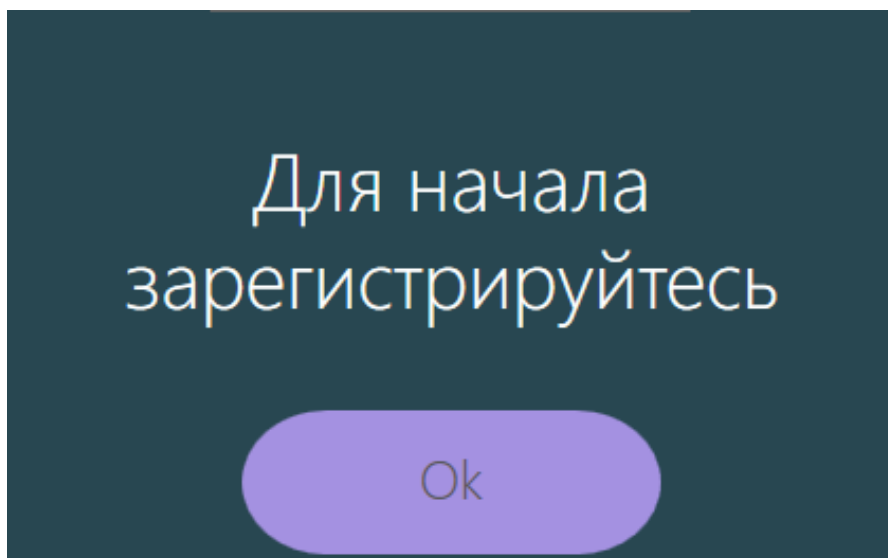


Рисунок 6.22 - Сообщение для гостя

Как можно заметить гость имеет ограниченный функционал, созданный только в ознакомительных целях.

ЗАКЛЮЧЕНИЕ

В итоге выполнения данного курсового проекта было разработано программное средство “Clean Brain”. При разработке были выполнены все пункты из указанного списка предполагаемого основного функционала приложения.

В программном средстве были реализованы следующие функции:

- регистрация пользователя в системе;
- вход пользователя в систему;
- вход гостя в систему;
- просмотр данных об психологах, услугах, отзывах и текущих сеансов;
- добавление данных о психологах и услугах;
- удаление данных о психологах и услугах;
- редактирование страниц психологов и услуг;
- бронирование талонов;
- добавление отзывов;
- регистрация новых талонов;
- изменение информации в личном кабинете;
- изменение языка приложения;
- изменение темы приложения

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

СПИСОК ЛИТЕРАТУРЫ

1. MSDN сеть разработчиков в Microsoft [Электронный ресурс] / Режим доступа: <http://msdn.microsoft.com/library/rus/> . Дата доступа: 20.04.2023.
2. METANIT.COM Сайт о программировании [Электронный ресурс] / Режим доступа: <https://metanit.com> . Дата доступа: 20.04.2023.
3. ProfessorWeb .NET & Web Programming [Электронный ресурс] / Режим доступа: <https://professorweb.ru> Дата доступа: 13.04.2023.
4. Форум для программистов или разработчиков [Электронный ресурс] – <https://stackoverflow.com/> – Дата доступа: 5.05.2023.
5. Сайт психологического центра “Nora” [Электронный ресурс] / Режим доступа: <https://www.nora.by>. Дата доступа: 20.04.2023.
6. Сайт психологического центра “INSIDE” [Электронный ресурс] / Режим доступа: <https://trevoga.by>. Дата доступа: 20.04.2023.
7. Сайт психологического центра “Grow Up” [Электронный ресурс] / Режим доступа: <https://psiholog-minsk.by>. Дата доступа: 20.04.2023.

ПРИЛОЖЕНИЕ А

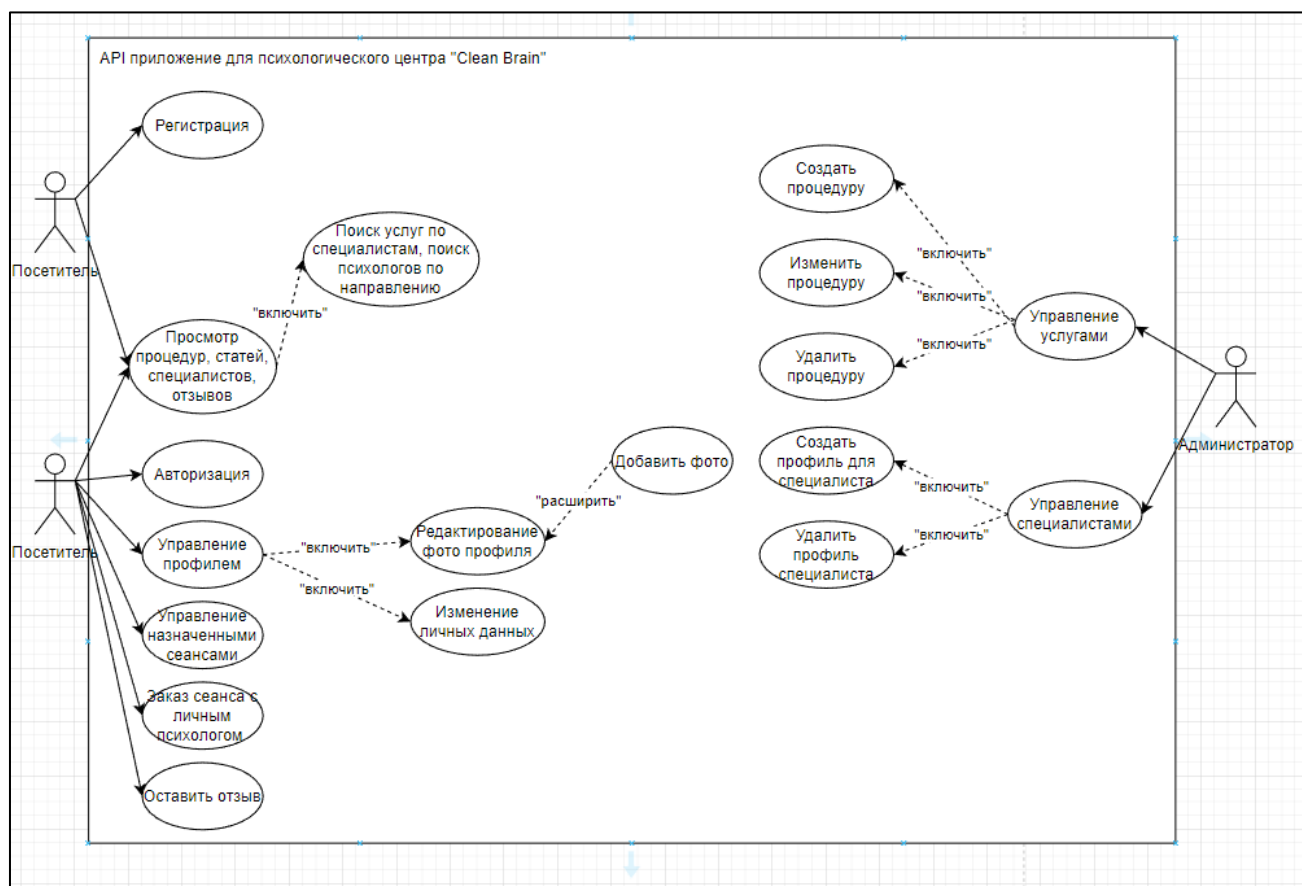


Рисунок 1 – Диаграмма вариантов использования

ПРИЛОЖЕНИЕ Б

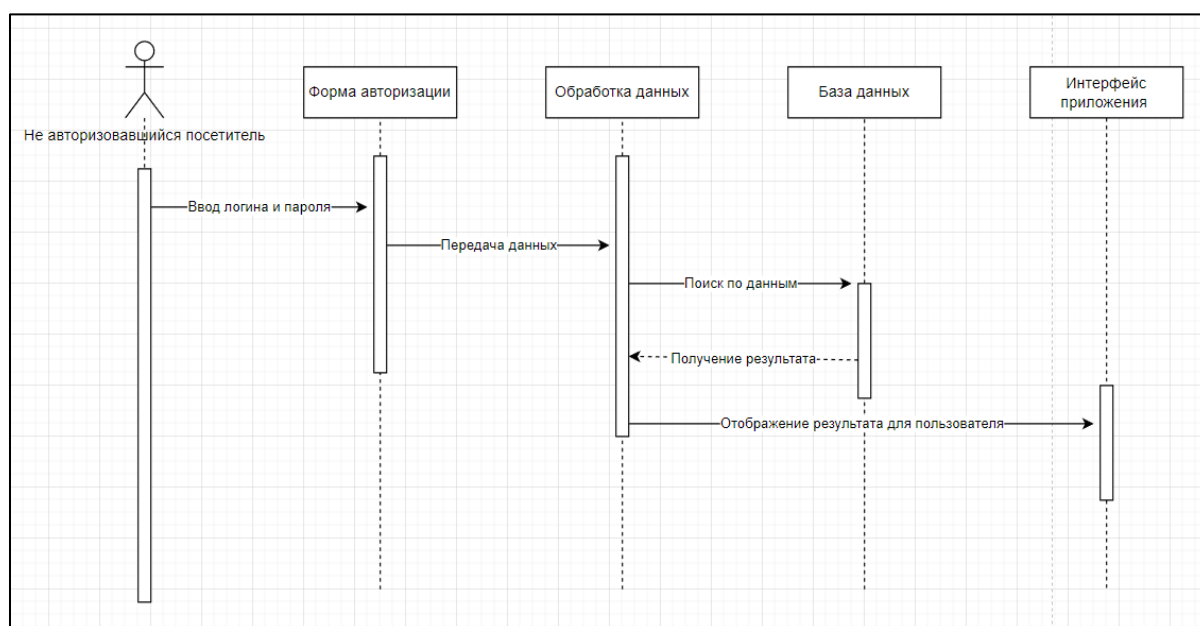


Рисунок 2 – Диаграмма последовательности для авторизации пользователя



ПРИЛОЖЕНИЕ Г

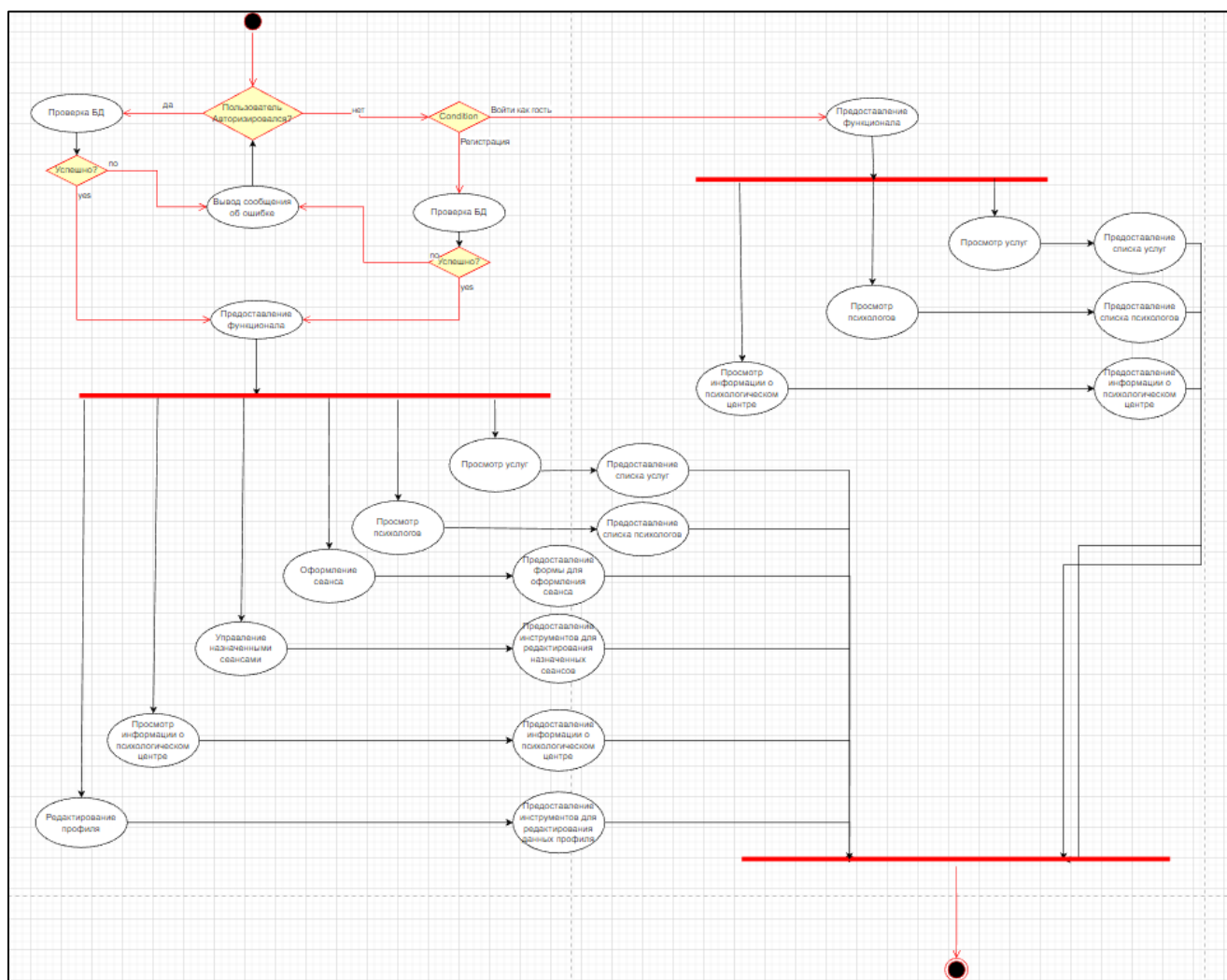


Рисунок 4 – Диаграмма деятельности для авторизованного пользователя и гостя

ПРИЛОЖЕНИЕ Д

Листинг 1: Класс UnitOfWork

```

public class UnitOfWork : IDisposable
{
    private Psychological_CenterEntities db = new
Psychological_CenterEntities();
    private ClientRepository clientRepository;
    private PsychologistRepository psychologistRepository;
    private TimeTableRepository timeTableRepository;
    private DegreeRepository degreeRepository;
    private SpecializationRepository specializationRepository;
    private ReviewRepository reviewRepository;
    private ProcedureRepository procedureRepository;
    private VoucherRepository voucherRepository;
    private BookingRepository bookingRepository;
}
  
```

```

public BookingRepository Booking
{
    get
    {
        if (bookingRepository == null)
            bookingRepository = new BookingRepository(db);
        return bookingRepository;
    }
}
public VoucherRepository Voucher
{
    get
    {
        if (voucherRepository == null)
            voucherRepository = new VoucherRepository(db);
        return voucherRepository;
    }
}

public ProcedureRepository Procedure
{
    get
    {
        if (procedureRepository == null)
            procedureRepository = new
ProcedureRepository(db);
        return procedureRepository;
    }
}
public ReviewRepository Review
{
    get
    {
        if (reviewRepository == null)
            reviewRepository = new ReviewRepository(db);
        return reviewRepository;
    }
}

public SpecializationRepository Specialization
{
    get
    {
        if (specializationRepository == null)
            specializationRepository = new
SpecializationRepository(db);
        return specializationRepository;
    }
}
public DegreeRepository Degree
{
    get

```

```

        {
            if (degreeRepository == null)
                degreeRepository = new DegreeRepository(db);
            return degreeRepository;
        }
    }
    public TimeTableRepository TimeTable
    {
        get
        {
            if (timeTableRepository == null)
                timeTableRepository = new
TimeTableRepository(db);
            return timeTableRepository;
        }
    }
    public PsychologistRepository Psychologist
    {
        get
        {
            if (psychologistRepository == null)
                psychologistRepository = new
PsychologistRepository(db);
            return psychologistRepository;
        }
    }
    public ClientRepository Client
    {
        get
        {
            if(clientRepository == null)
                clientRepository = new ClientRepository(db);
            return clientRepository;
        }
    }
    public void Save()
    {
        try
        {
            db.SaveChanges();
        }
        catch (DbEntityValidationException ex)
        {
            foreach (var entityValidationError in
ex.EntityValidationErrors)
            {
                string entityName =
entityValidationError.Entry.Entity.GetType().Name;
                foreach (var validationError in
entityValidationError.ValidationErrors)
                {
                    string propertyName =
validationError.PropertyName;

```



```

        string errorMessage =
validationError.ErrorMessage;

        MessageBox.Show($"Ошибка проверки сущности:
{entityName}, Свойство: {propertyName}, Сообщение об ошибке:
{errorMessage}");
    }
}

private bool disposed = false;

public virtual void Dispose(bool disposing)
{
    if (!this.disposed)
    {
        if (disposing)
        {
            db.Dispose();
        }
        this.disposed = true;
    }
}

public virtual void Dispose()
{
    Dispose(true);
    GC.SuppressFinalize(this);
}
}

```

Листинг 2: Скрипт для создания базы данных

```

Use master;
create database Psychological_Center

use Psychological_Center
create table Client
(
    Id_client int IDENTITY(1,1) primary key,
    Name_Client nvarchar(max) not null,
    Surname_Client nvarchar(max) not null,
    Login_Client nvarchar(max) not null,
    Password_Client nvarchar(max) not null,
    Photo_Client varbinary(max),
    Mail_Client nvarchar(max) not null
);

create table Specialization
(
    Spezialization_Name nvarchar(60) primary key
);

create table Academic_Degree

```

```

(
Academic_Name nvarchar(60) primary key
);

create table Psychologist
(
Id_Psychologist int IDENTITY(1,1) primary key,
Name_Psychologist nvarchar(max) not null,
Surname_Psychologist nvarchar(max) not null,
Patronymic_Psychologist nvarchar(max) not null,
Experience smallint not null,
Photo_Psychologist varbinary(max),
Description nvarchar(max),
Specialization_Psychologist nvarchar(60) not null foreign key
references

        Specialization(Spezialization_Name),
Degree nvarchar(60) not null foreign key references

        Academic_Degree(Academic_Name)
);
create table Voucher
(
Id_Voucher int IDENTITY(1,1) primary key,
Date_Voucher date not null,
Time_Voucher_Start time not null,
Time_Voucher_End time not null,
Id_Psychologist int not null foreign key references
        Psychologist(Id_Psychologist),
Ordered nchar(10) not null
);

create table Procedures
(
Id_Procedure int IDENTITY(1,1) primary key,
Name_Procedure nvarchar(max) not null,
Price float not null,
Depiction nvarchar(max) not null,
Spezialization_Procedure nvarchar(60),
Photo_Procedure varbinary(max)
);

create table Timetable
(
Id_Psychologist int primary key foreign key references
        Psychologist(Id_Psychologist),
MondStart time,
MondEnd time,
TueStart time,
TueEnd time,
WenStart time,
WenEnd time,
ThuStart time,

```

```

ThuEnd time,
FriStart time,
FriEnd time
);

create table Booking
(
ID_Booking smallint IDENTITY(1,1) primary key,
Date_Booking date not null,
Id_Procedure int not null foreign key references
    Procedures(Id_Procedure),
Id_Client int not null foreign key references
    Client(Id_Client),
Id_Psychologist int not null foreign key references
    Psychologist(Id_Psychologist),
Id_Voucher int not null foreign key references
    Voucher(Id_Voucher),
Time_Booking time
);

create table Review
(
Id_Review int IDENTITY(1,1) primary key,
Photo_Review varbinary(max),
Name_Client nvarchar(max) not null,
Id_Client int not null foreign key references
    Client(Id_Client),
Review nvarchar(max) not null
);

```