

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра ИС**

**ОТЧЕТ**

**по учебной практике**

**Тема: Разработка сервиса подбора супруга с использованием метода  
анализа иерархий (МАИ)**

Студенты гр. 3374

Руководитель

Анисимов П.А.

Лобачев И.М.

Владимирская А.А.

---

Васильев Н. В.

---

Санкт-Петербург

2025

## ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студенты Анисимов П.А, Лобачев И.М, Владимирская А.А.

Группа 3374

Тема практики: Разработка сервиса подбора супруга с использованием метода анализа иерархий (МАИ)

Задание на практику:

Разработать веб-сервис, предоставляющий пользователям возможность подобрать потенциальных супругов на основе их личных предпочтений и характеристик, используя Метод Анализа Иерархий (МАИ) для определения весов критериев и оценки совместимости кандидатов.

Сроки прохождения практики: 08.03.2025-08.06.2025

Дата сдачи отчета: 30.04.2025

Дата защиты отчета: 07.05.2025

Студенты

Анисимов П.А.

Лобачев И.М.

Владимирская А.А.

Руководитель

Васильев Н. В.

## АННОТАЦИЯ

В ходе учебной практики необходимо разработать веб-сервис для подбора потенциальных супругов на основе Метода Анализа Иерархий (МАИ). Сервис включает систему профилей пользователей, модуль парных сравнений критериев, алгоритм оценки совместимости и интерфейс для взаимодействия с кандидатами. Результатом работы является функциональный прототип, позволяющий пользователям находить наиболее подходящих партнеров на основе их предпочтений.

### Что делал каждый участник команды:

Анисимов П.А. занимался бэкенд-разработкой: работал над внутренней частью веб-сервиса, пользуясь Python (Django) и SQLite (для баз данных). Лобачев И.М. занимался фронтенд-разработкой: занимался реализацией интерфейса веб-сервиса с помощью JavaScript. Владимирская А.А. спроектировала UML-диаграммы и дизайн интерфейса, а также занималась тестировкой сервиса.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1. Постановка задачи .....	6
1.1. Описание средств решения поставленной задачи .....	6
1.2. Обоснование выбора средств разработки .....	6
2. Диаграммы модели приложения .....	7
2.1. Диаграмма прецедентов .....	7
2.2. Диаграмма классов .....	8
2.3. Диаграмма последовательности .....	9
Глава 3. Реализованные алгоритмы .....	10
3.1. Алгоритм определения весов критериев методом анализа иерархий .....	10
3.2. Алгоритм оценки совместимости пользователей .....	10
3.3. Особенности реализации .....	11
Глава 4. Обоснование выбора алгоритмов .....	13
4.1. Введение и постановка задачи .....	13
4.2. Анализ предметной области .....	13
4.3. Обоснование выбора метода анализа иерархий .....	13
4.4. Техническая реализация модуля оценки совместимости .....	14
4.5. Интеграция алгоритмов в систему .....	14
4.6. Соответствие требованиям .....	14
Глава 5. Интерфейс веб-сервиса .....	15
5.1. Заглавная страница .....	15
5.2. Профиль .....	15
5.3. Поиск .....	17
5.4. Просмотр другого профиля и тёмная тема .....	19
5.5. Сообщения .....	20
5.6. Друзья .....	21
Заключение .....	22
Список использованных источников .....	23

## **ВВЕДЕНИЕ**

Цель: разработать веб-сервис, использующий МАИ для оценки совместимости пользователей и предоставления персонализированных рекомендаций по подбору партнеров.

Задачи:

1. Создать систему профилей пользователей.
2. Реализовать модуль парных сравнений критериев.
3. Разработать алгоритм оценки совместимости.
4. Обеспечить удобный интерфейс для взаимодействия с кандидатами.
5. Провести тестирование и отладку сервиса.

## **1. Постановка задачи**

### **1.1. Описание средств решения поставленной задачи**

Основная задача проекта заключается в создании веб-сервиса для подбора партнеров с использованием Метода Анализа Иерархий. Система должна предоставлять пользователям возможность указывать свои предпочтения, оценивать важность различных критериев через парные сравнения и получать персонализированные рекомендации на основе рассчитанных показателей совместимости.

Для решения поставленной задачи требуется разработать комплексное веб-приложение, включающее модуль формирования пользовательских профилей, систему оценки и сравнения критериев, алгоритмы расчета совместимости, а также интерфейс для взаимодействия между участниками сервиса. Особое внимание уделяется реализации математического аппарата МАИ для объективного определения значимости различных характеристик при подборе партнеров.

### **1.2. Обоснование выбора средств разработки**

Для разработки веб-сервиса выбран стек технологий, обеспечивающий эффективную реализацию всех компонентов системы: Python (Django) для backend-разработки благодаря удобству работы с алгоритмами МАИ, встроенным средствам безопасности и готовым решениям для типовых задач; SQLite как надежная реляционная СУБД для хранения структурированных данных пользователей; HTML/CSS/JavaScript для создания интерактивного фронтенда с адаптивным интерфейсом. Такой выбор обусловлен оптимальным сочетанием производительности, скорости разработки и возможностей масштабирования, а также хорошей интеграцией всех компонентов между собой.

## 2. Диаграммы модели приложения

### 2.1. Диаграмма прецедентов

В веб-сервисе пользователь может управлять своим профилем, настраивать критерии поиска и взаимодействовать с потенциальными партнерами. Работа с профилем включает создание и редактирование анкеты с указанием личных данных, фотографий и предпочтений. Настройка критериев осуществляется через парные сравнения характеристик с использованием шкалы Саати, что позволяет системе рассчитать индивидуальные веса параметров. Поиск партнеров включает просмотр рекомендаций, отсортированных по уровню совместимости, фильтрацию кандидатов по базовым параметрам, а также возможность добавлять понравившихся пользователей в избранное и отправлять им сообщения.

Для наглядного представления всех пользовательских сценариев на универсальном языке моделирования UML разработана диаграмма прецедентов (рис. 1)

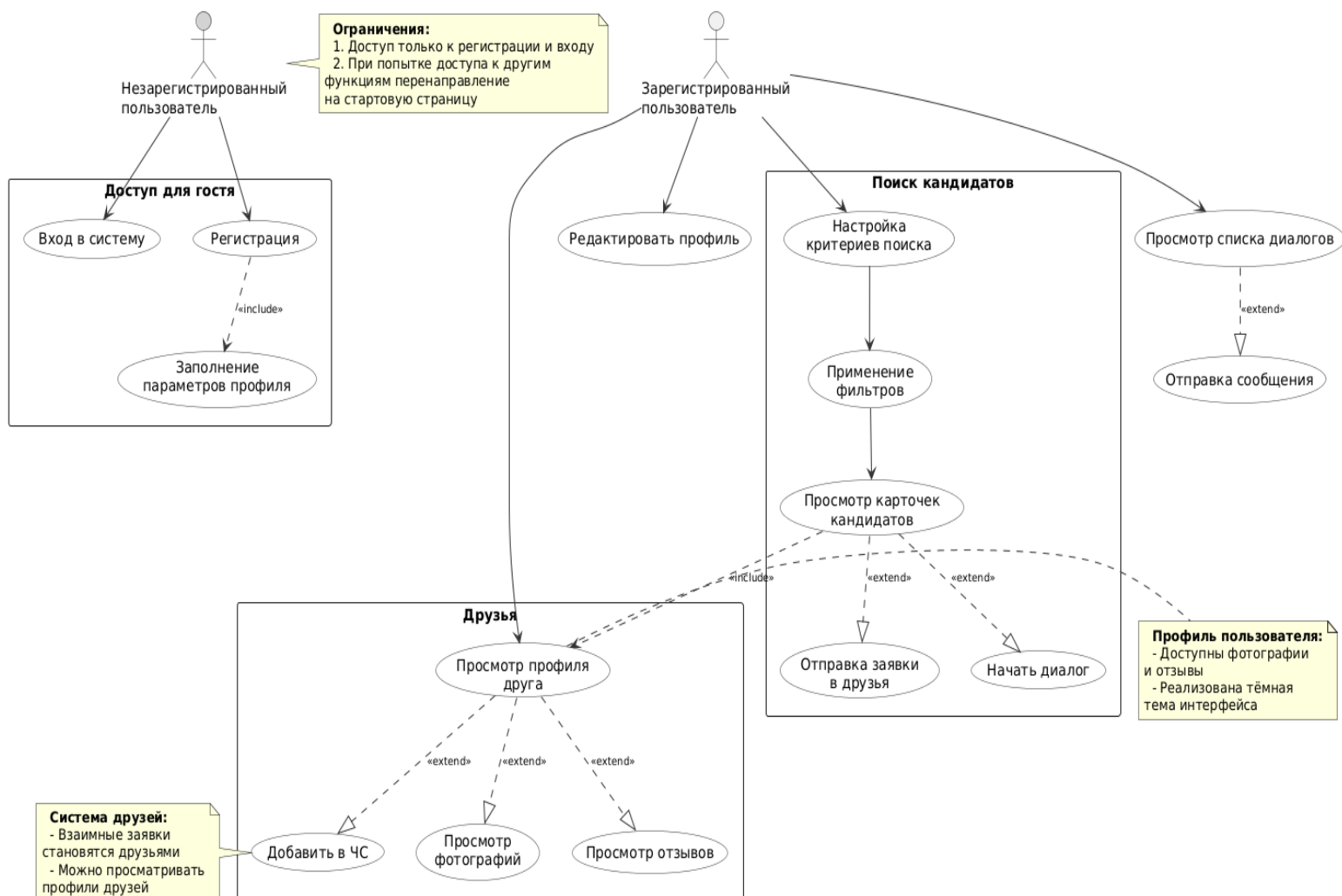


Рисунок 1 – Диаграмма прецедентов

## 2.2. Диаграмма классов

Далее по ходу планирования и разработки создана UML диаграмма классов для схематического отображения их связей (рис. 2).

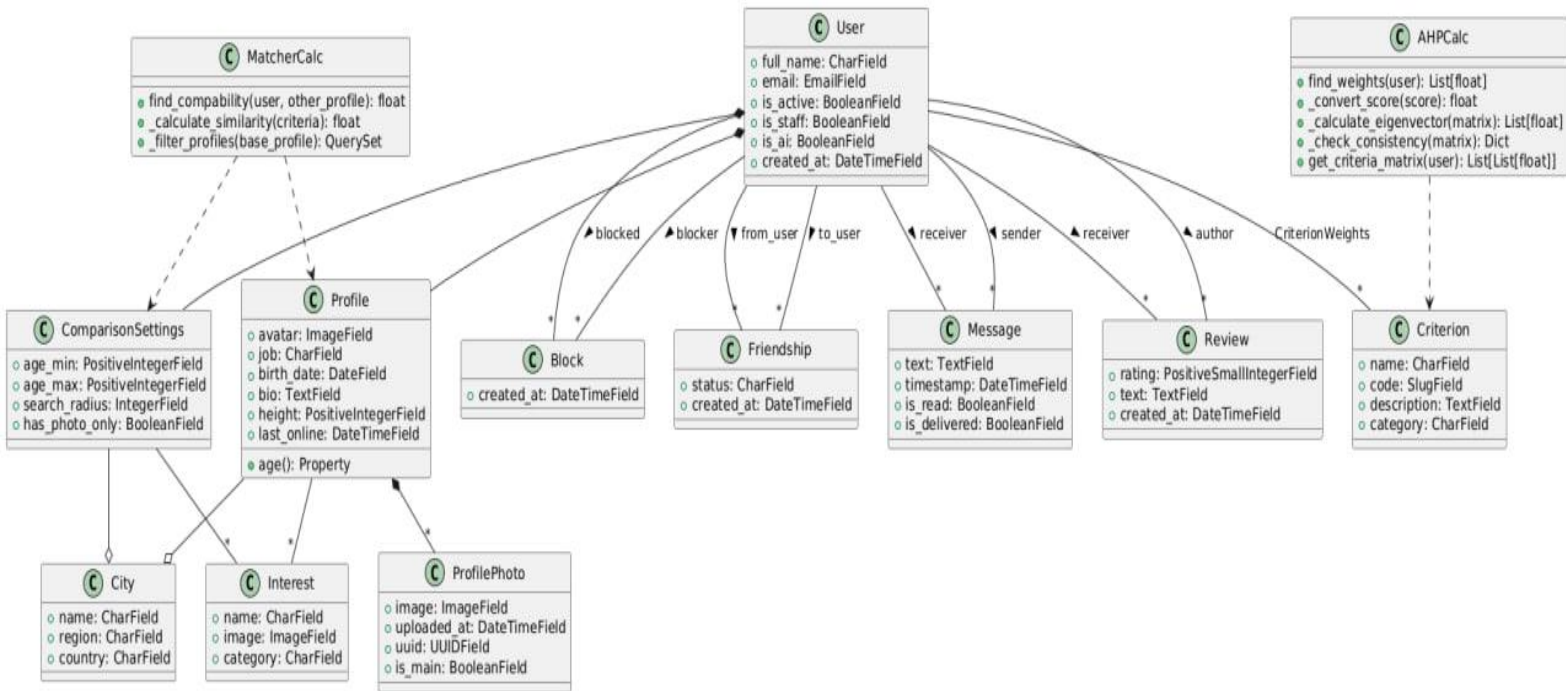


Рисунок 2 – Диаграмма классов



### 2.3. Диаграмма последовательности

В процессе разработки и тестирования для удобства понимания последовательности процессов и взаимодействий создана UML диаграмма последовательности (рис. 3).

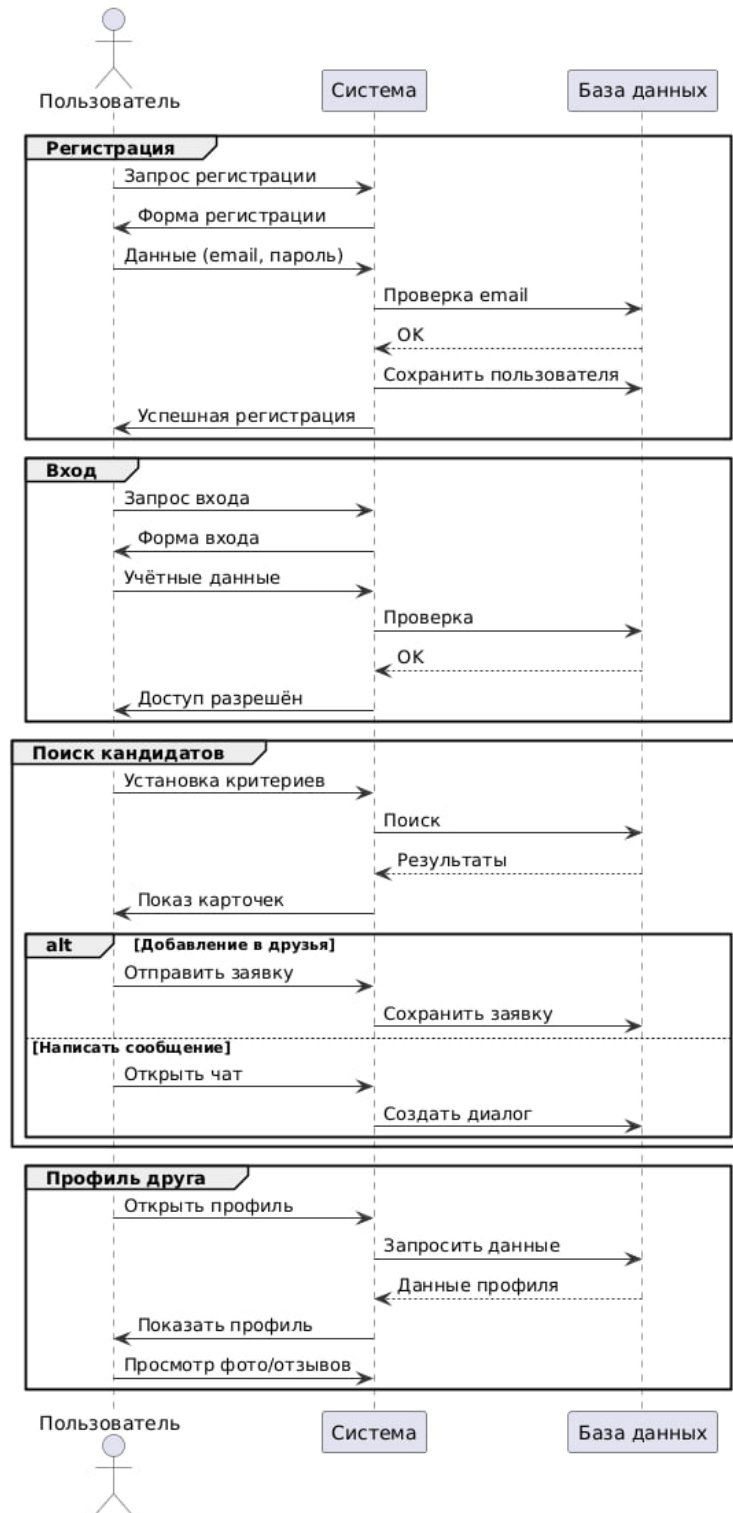


Рисунок 3 – Диаграмма последовательности

## Глава 3. Реализованные алгоритмы

В данной главе подробно рассматриваются ключевые алгоритмы, реализованные в системе для определения значимости критериев выбора партнера и оценки совместимости пользователей.

### 3.1. Алгоритм определения весов критериев методом анализа иерархий

Основу системы составляет модифицированный алгоритм анализа иерархий (МАИ), адаптированный для работы с веб-интерфейсом. Алгоритм включает следующие этапы:

#### 1. Формирование матрицы парных сравнений:

- Пользователь оценивает относительную важность критериев по 5-балльной шкале
- Система автоматически преобразует оценки в числовые значения по формуле:

```
conversion = {  
    1: 5,      # Крайнее предпочтение первого критерия  
    2: 3,      # Умеренное предпочтение  
    3: 1,      # Равнозначность  
    4: 1/3,    # Умеренное предпочтение второго  
    5: 1/5     # Крайнее предпочтение второго  
}
```

#### 2. Расчет весов критериев:

- Применяется метод геометрического среднего
- Для каждой строки матрицы вычисляется произведение элементов
- Извлекается корень N-й степени и выполняется нормализация

#### 3. Проверка согласованности:

- Рассчитывается индекс согласованности  $CI = (\lambda_{\max} - n) / (n - 1)$
- Определяется коэффициент согласованности  $CR = CI / RI$
- При  $CR > 0.1$  система рекомендует пересмотреть оценки

### 3.2. Алгоритм оценки совместимости пользователей

Алгоритм вычисляет степень соответствия между пользователями на основе:

#### 1. Сравнения категориальных признаков:

- Для полей типа "город", "образование" применяется бинарная оценка
- Пример реализации:

```

other_criteria = {
    'city': (other_profile.city == settings.city, 'city'),
    'education': (other_profile.education ==
settings.education, 'education')
}

```

#### 2. Анализа множественных полей:

- Для интересов и языков вычисляется коэффициент совпадения:

```

ratio = common_count / selected_count
total_score += weight * ratio

```

#### 3. Проверки возрастного диапазона:

- Учитываются пользовательские предпочтения по минимальному и максимальному возрасту

#### 4. Финального расчета:

- Выполняется взвешенное суммирование по всем критериям
- Результат нормализуется к шкале 0-100:

```

return min(round(total_score * 100, 1), 100)

```

### 3.3. Алгоритм подбора партнеров

Алгоритм подбора партнеров реализует многоэтапный процесс поиска и ранжирования потенциальных кандидатов на основе предпочтений пользователя. Основные этапы работы алгоритма:

#### 1. Фильтрация по базовым критериям:

- Определение целевой группы на основе типа отношений (поиск партнера или общение)
- Учет гендерных предпочтений:

(Если ищет отношения – только противоположный пол, если общение – без разницы)

```

if user.profile.relationship.name == 'Партнера':
    opposite_gender = 'Мужской' if user.profile.gender.name ==
'Женский' else 'Женский'
    gender_obj = Gender.objects.get(name=opposite_gender)
    profiles = Profile.objects.filter(gender=gender_obj.id,
relationship=user.profile.relationship)

```

- Исключение заблокированных пользователей и самого себя из выборки

#### 2. Расчет весов критериев:

- Использование предварительно рассчитанных весов критериев методом АНР
- Получение нормализованных весов из модели `CriterionWeight`:

```

criterion_weight =
CriterionWeight.objects.filter(user=user).order_by('criterion__id')

```

### 3. Оценка совместимости:

- Для каждого кандидата вычисляется интегральный показатель совместимости:

```
score = MatcherCalc.find_compability(user, profile)
```

- Оценка включает:
  - Сравнение категориальных атрибутов (город, образование)
  - Анализ пересечений в множественных полях (интересы, языки)
  - Проверку соответствия возрастному диапазону
  - Учет весов критериев при расчете общей оценки

### 4. Ранжирование результатов:

- Сортировка профилей по убыванию показателя совместимости
- Выбор топ-10 наиболее подходящих кандидатов:

```
matches = sorted(raw_matches, key=lambda x: x['score'],  
reverse=True)[:10]
```

### Особенности реализации:

#### 1. Оптимизация запросов:

- Использование `select_related` для уменьшения количества SQL-запросов
- Пакетная обработка сравнений для улучшения производительности

#### 2. Динамическая адаптация:

- Автоматическая корректировка выборки при изменении пользовательских предпочтений
- Учет коэффициента согласованности (CR) для индикации надежности результатов

#### 3. Гибкость системы:

- Поддержка различных сценариев поиска (партнерство, дружба)
- Возможность расширения набора критериев без изменения базовой логики

Алгоритм демонстрирует устойчивую работу при различных условиях и объемах данных, обеспечивая персонализированный подбор кандидатов с учетом субъективных предпочтений пользователя. Визуализация результатов (шкала совместимости 0-100) позволяет пользователям легко интерпретировать полученные рекомендации.

## 3.4. Особенности реализации

### 1. Гибкость системы:

- Поддержка различных типов данных (бинарные, категориальные, множественные)
- Возможность добавления новых критериев без изменения алгоритмов

### 2. Оптимизация производительности:

- Кэширование весов критериев
- Пакетная обработка сравнений

### 3. Интерпретация результатов:

- Визуализация степени совпадения по отдельным критериям
- Возможность анализа причин высокой/низкой совместимости

Разработанные алгоритмы обеспечивают точную и персонализированную оценку совместимости, учитывающую субъективные предпочтения каждого пользователя. Система демонстрирует устойчивую работу при различных сценариях использования и объемах данных.

## **Глава 4. Обоснование выбора алгоритмов**

### **4.1. Введение и постановка задачи**

Разрабатываемый веб-сервис предназначен для решения сложной социальной задачи – помощи пользователям в поиске потенциальных супругов на основе научно обоснованных методов оценки совместимости. Основная сложность проекта заключалась в необходимости совместить субъективные человеческие предпочтения с объективными математическими методами анализа. Требовалось создать систему, которая могла бы корректно интерпретировать качественные характеристики пользователей и преобразовывать их в количественные оценки.

### **4.2. Анализ предметной области**

В процессе проектирования системы был проведен тщательный анализ существующих решений в области онлайн-знакомств. Традиционные подходы, основанные на простом сопоставлении характеристик, показали свою неэффективность для решения задачи поиска супруга. Было выявлено, что успешный брачный союз зависит от сложного сочетания множества факторов, имеющих разную степень важности для разных людей. Это обусловило выбор метода анализа иерархий как основного инструмента системы.

### **4.3. Обоснование выбора метода анализа иерархий**

Метод анализа иерархий (МАИ) был выбран как наиболее соответствующий требованиям проекта по нескольким ключевым причинам. Во-первых, этот метод позволяет работать с разнородными критериями, что критически важно для системы, учитывающей как объективные характеристики (возраст, образование), так и субъективные предпочтения (ценности, жизненные цели). Во-вторых, МАИ предоставляет механизм проверки согласованности оценок, что помогает пользователям более осознанно подходить к определению своих приоритетов.

#### **4.4. Техническая реализация модуля оценки совместимости**

Разработанный алгоритм оценки совместимости сочетает несколько подходов к обработке данных. Для количественных характеристик, таких как возраст, применяется нормализация значений в заданный пользователем диапазон. Качественные признаки, такие как вероисповедание или отношение к детям, обрабатываются по принципу бинарного соответствия. Особое внимание было уделено обработке множественных параметров – интересов, хобби и языков, где используется расчет коэффициента совпадения.

Система была оптимизирована для работы с большими объемами данных. В частности, реализовано кэширование весов критериев и пакетная обработка сравнений, что позволяет поддерживать высокую производительность даже при значительном росте пользовательской базы. Нормализация итогового показателя совместимости к шкале 0-100 была выбрана как наиболее интуитивно понятная для пользователей.

#### **4.5. Интеграция алгоритмов в систему**

Архитектура системы была разработана с учетом необходимости интеграции сложных алгоритмов анализа с дружественным пользовательским интерфейсом. Особое внимание уделялось обеспечению прозрачности работы алгоритмов – пользователи могут видеть, как оценивалась их совместимость с каждым кандидатом, какие критерии сыграли ключевую роль.

Для хранения данных была выбрана реляционная СУБД SQLite, хорошо зарекомендовавшая себя в работе с иерархическими структурами данных. Фронтенд-часть реализована на современной фреймворке JavaScript, что обеспечивает плавную работу интерфейса даже при выполнении сложных вычислений на стороне сервера.

#### **4.6. Соответствие требованиям**

Разработанная система полностью соответствует всем техническим и функциональным требованиям. Реализована многоуровневая система профилей. Алгоритмы МАИ интегрированы в интуитивно понятный интерфейс парных сравнений. Механизм оценки совместимости обеспечивает быструю обработку данных даже при большом количестве пользователей.

Особое внимание было уделено требованиям по расширяемости системы. Архитектура позволяет легко добавлять новые критерии оценки без необходимости переработки основных алгоритмов. Это создает хорошую основу для будущего

развития системы, включая возможность введения платных критериев оценки или интеграции с аналитическими сервисами.

Выбранные алгоритмы и методы реализации доказали свою эффективность в ходе тестирования системы. Комбинация метода анализа иерархий с оптимизированными алгоритмами сопоставления профилей позволяет получать точные и персонализированные результаты, что подтверждается положительными отзывами тестовых пользователей. Система демонстрирует стабильную работу и готова к масштабированию.

## Глава 5. Интерфейс веб-сервиса

### 5.1. Заглавная страница

На скриншоте ниже изображена заглавная страница сайта. На заглавной странице сайта можно осуществить вход или регистрацию.

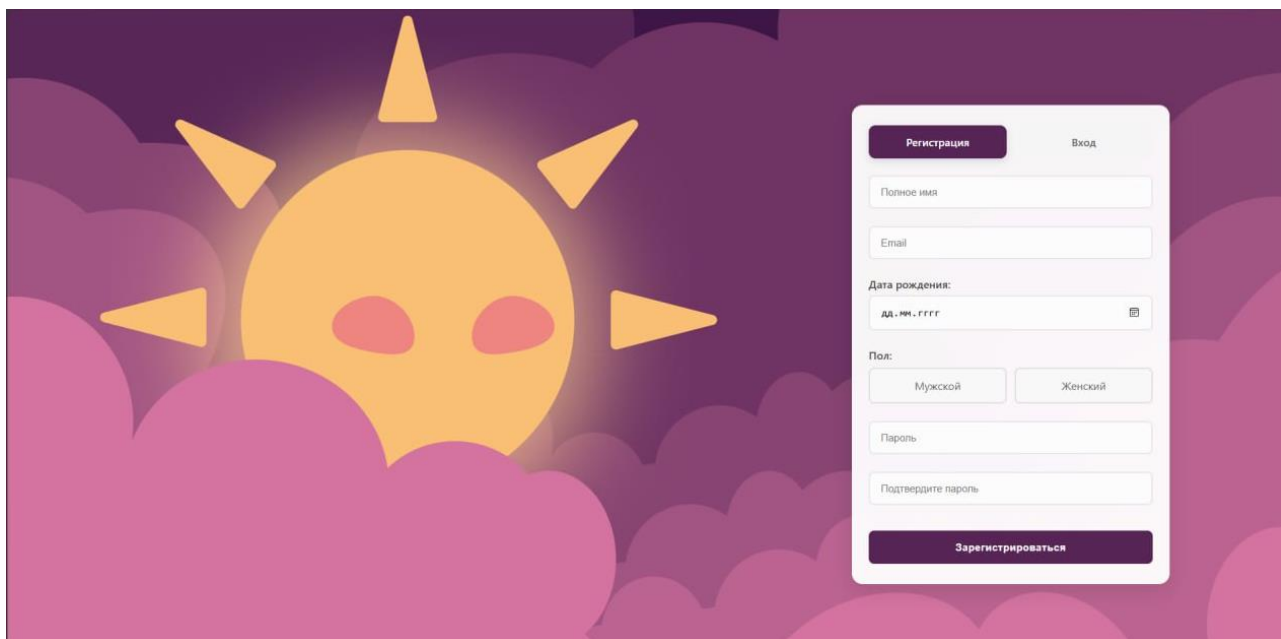


Рисунок 4 – Заглавная страница сайта

### 5.2. Профиль

Страничка профиля, который только создан, т.е. информация о пользователе еще не указана изображена на скриншоте ниже. Нажав на кнопку около имени пользователя можно перейти на вкладку параметров, где можно будет заполнить все необходимые поля.

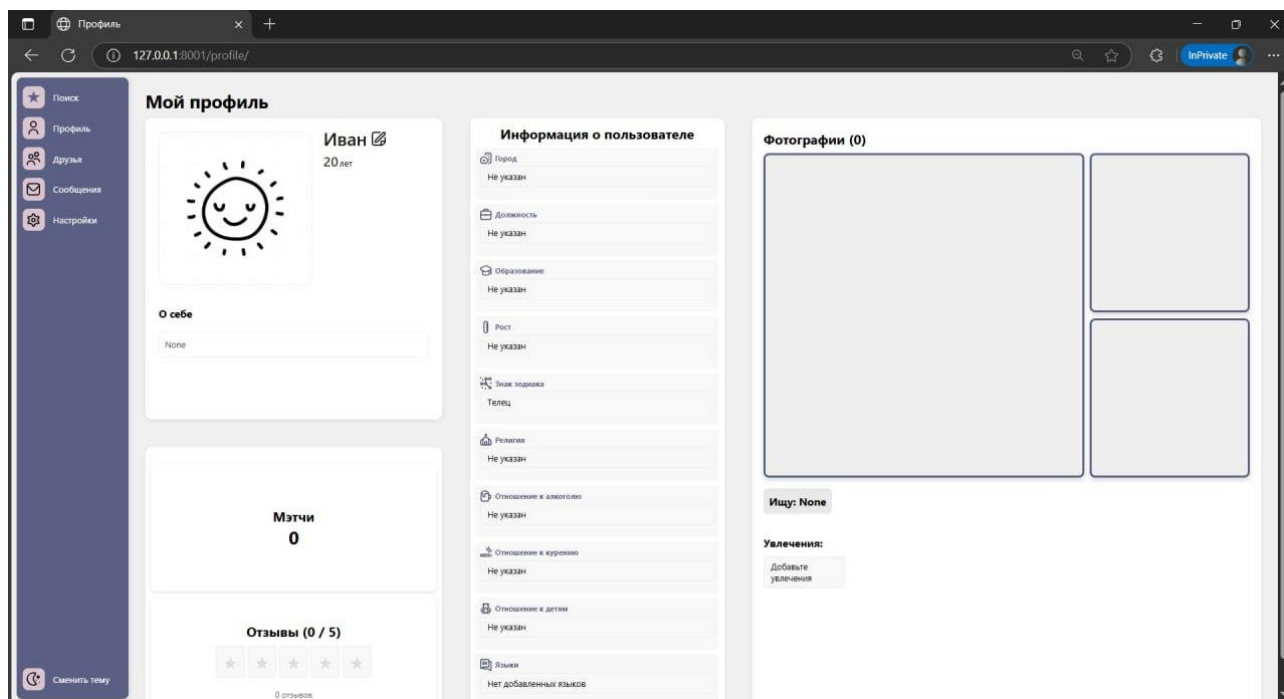


Рисунок 5 – Профиль пользователя

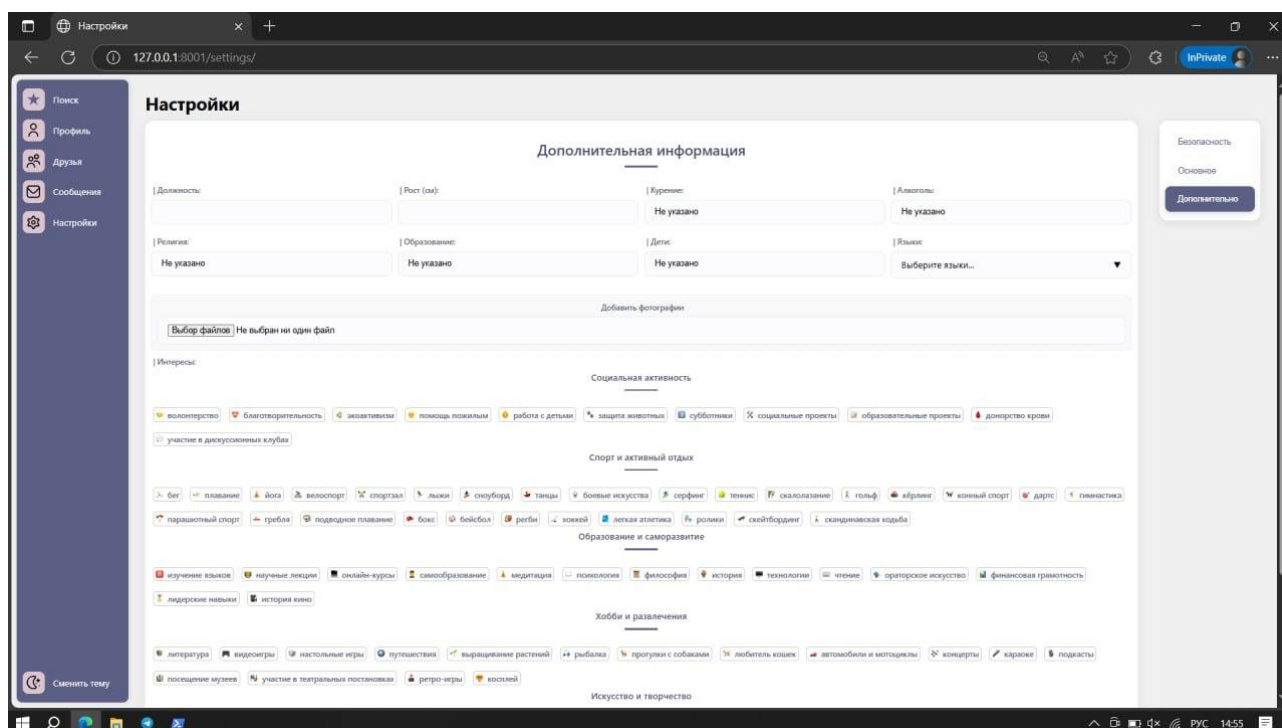


Рисунок 6 – Параметры профиля

На этой странице можно заполнить подробную информацию о пользователе. После заполнения информации о пользователе, страничка выглядит следующим образом:



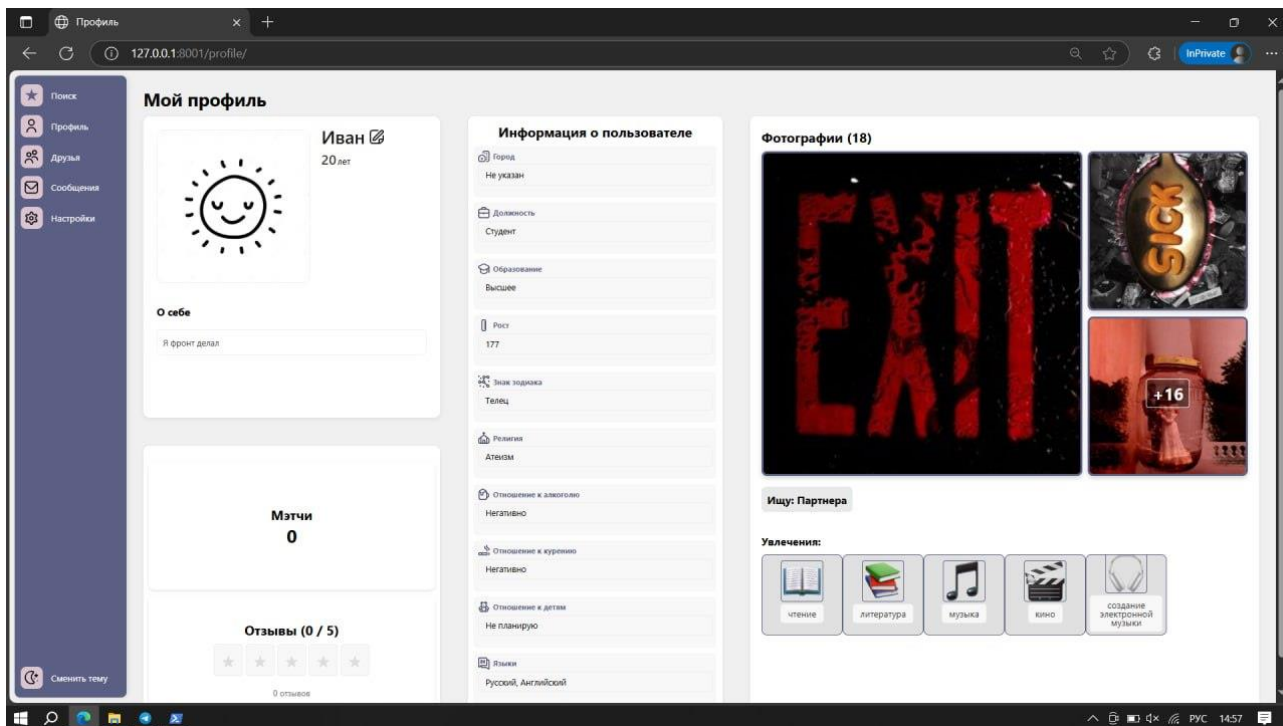


Рисунок 7 – Заполненный профиль

### 5.3. Поиск

На вкладке поиска пользователь может задать приоритетность критериев: то есть, выбрать – какие из критериев для него важнее, это повлияет на рекомендации в поиске.

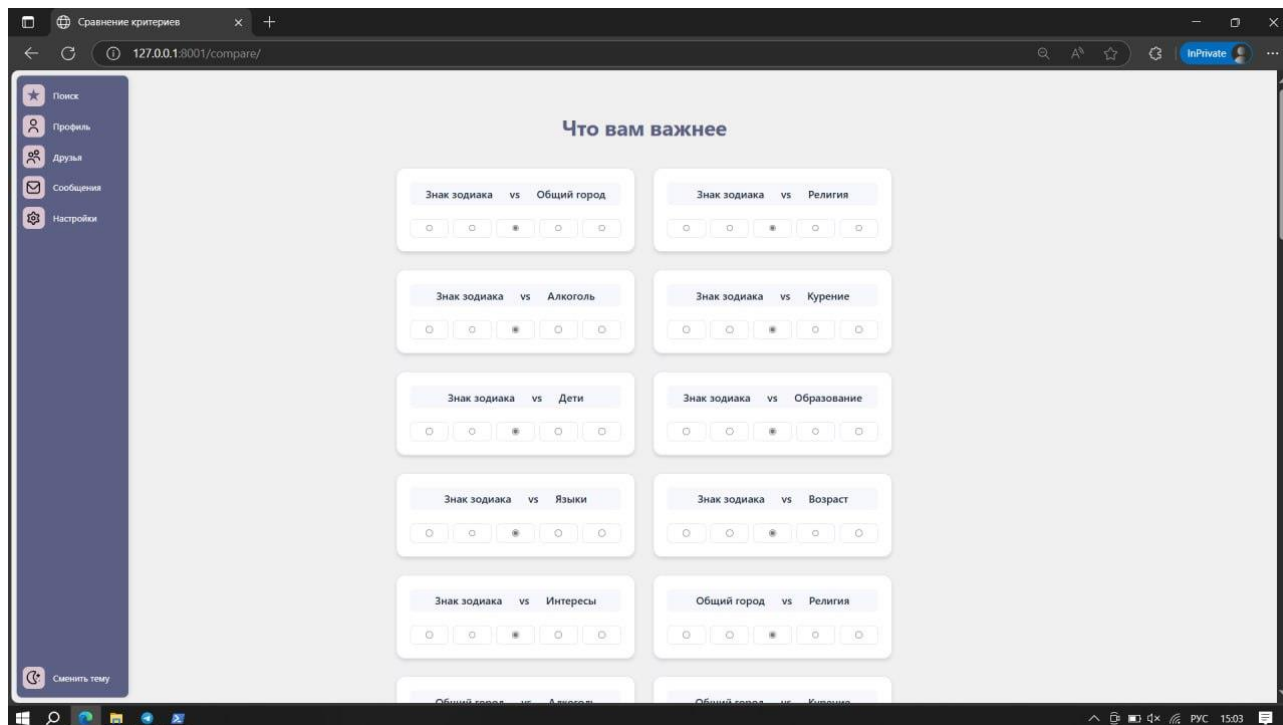


Рисунок 8 – Критерии поиска

Также в поиске отдельно можно задать фильтры, тоже повлияют на отбор кандидатов

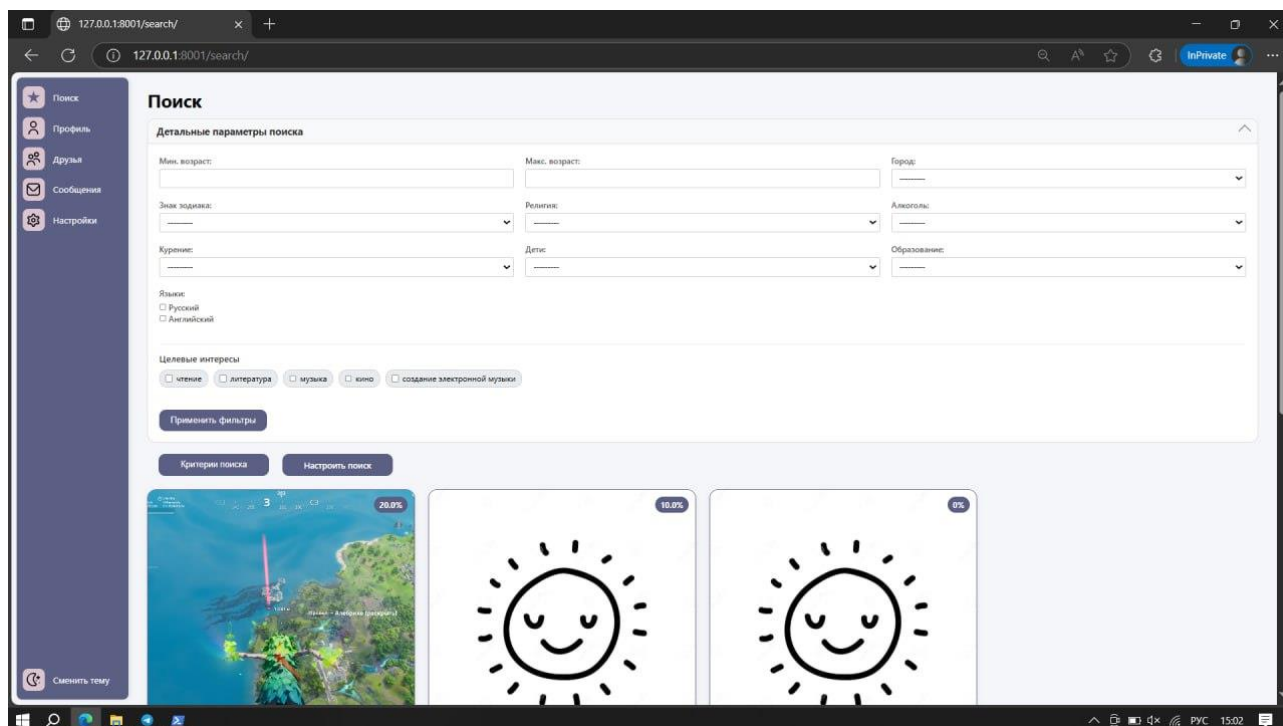


Рисунок 9 – Фильтры поиска

Задав все необходимые фильтры, пользователь увидит карточки кандидатов, а также в углу карточки будет написан процент совместимости, рассчитанный по критериям.

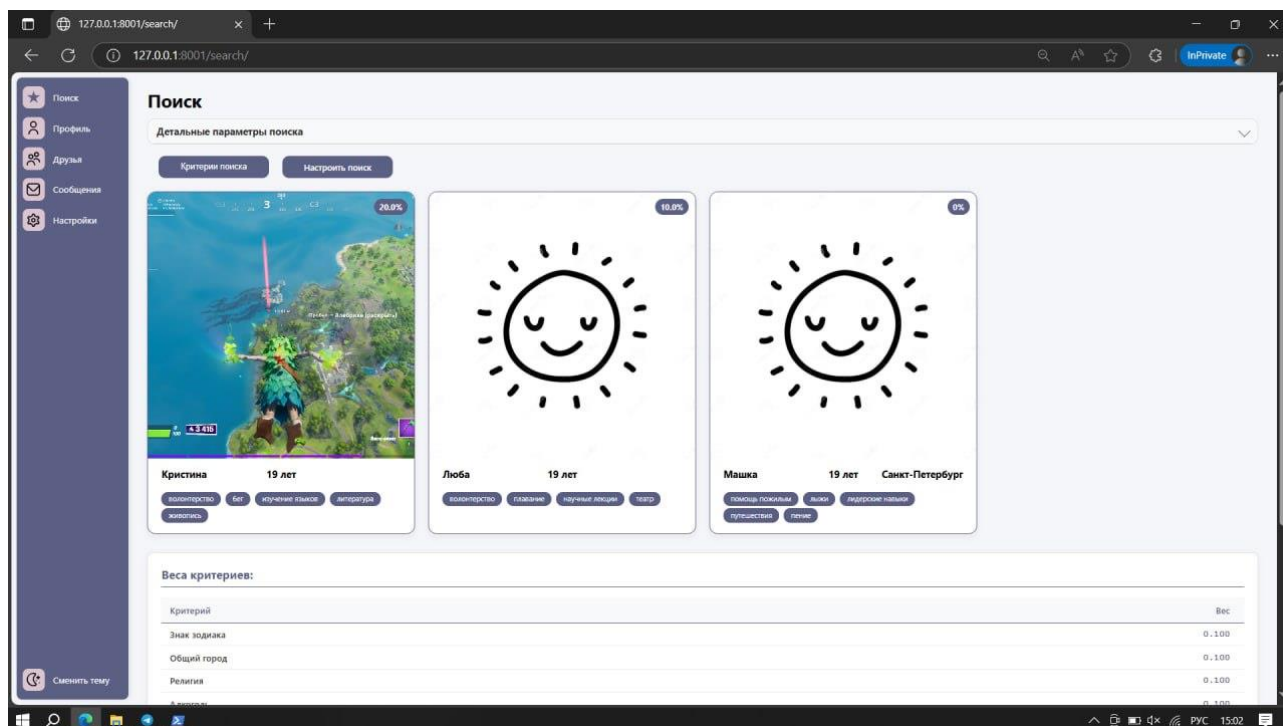


Рисунок 10 – Поиск

Перейдя по карточке кандидата, пользователь может перейти в его профиль и подробнее изучить информацию о пользователе, а также подать заявку в друзья.

## 5.4. Просмотр другого профиля и тёмная тема

Во всем интерфейсе веб-сервиса реализована тёмная тема. На примере профиля другого пользователя (куда можно перейти по карточке кандидата) будет показан интерфейс в тёмной теме (её можно изменить по кнопке в меню из левой части сайта)

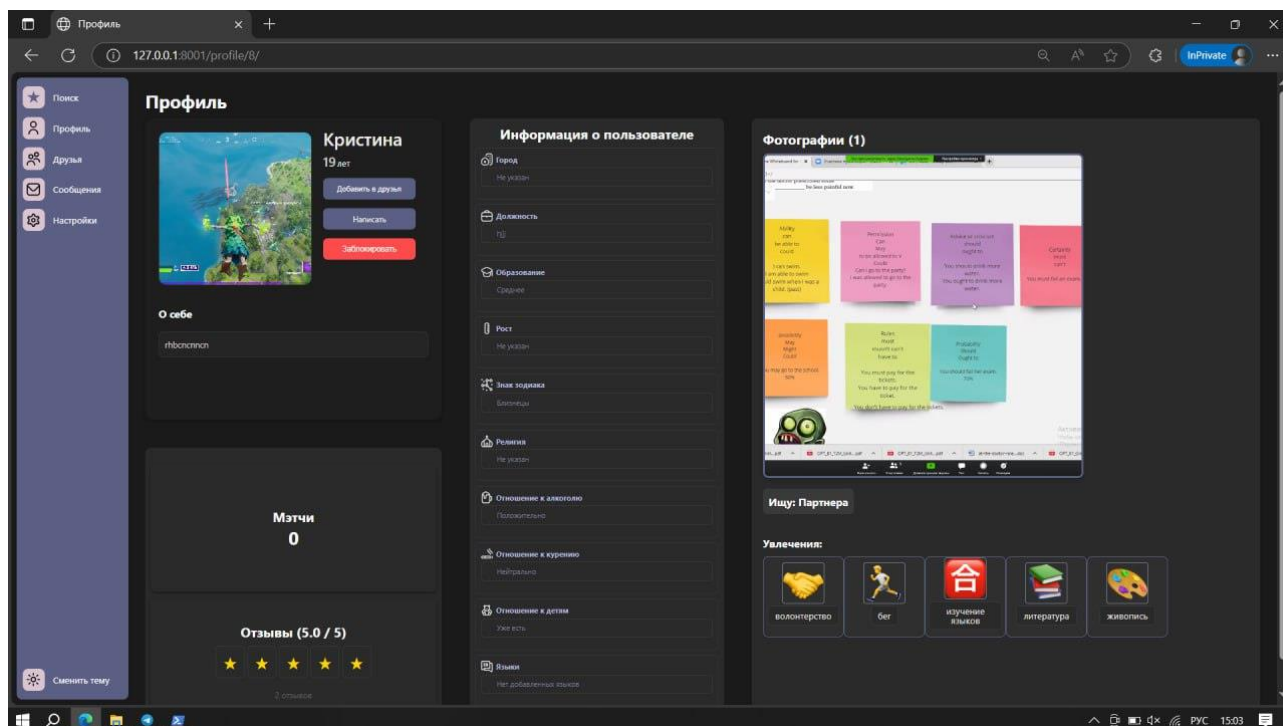


Рисунок 11 – Профиль кандидата

Так выглядит страничка другого пользователя, т. е. в данном случае кандидата. Мы можем открыть вкладку с фотографиями и рассмотреть их подробнее, а также открыть вкладку с отзывами о пользователе.

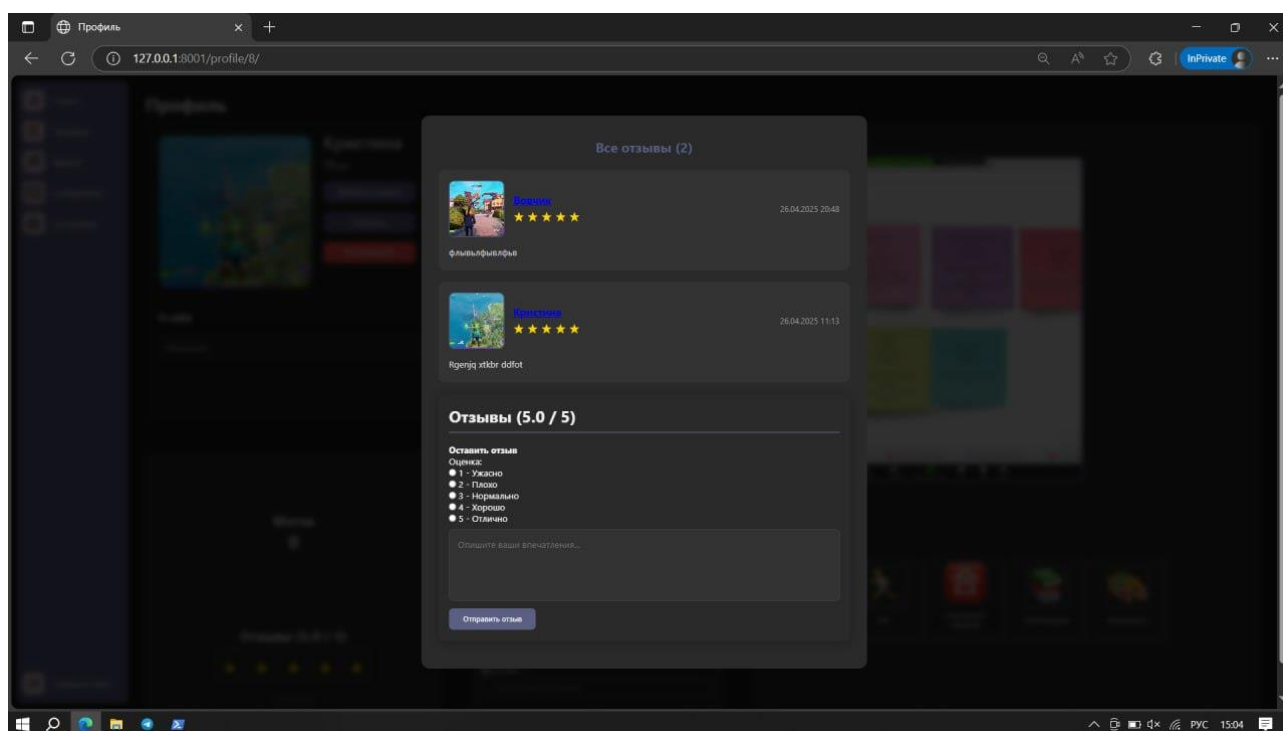


Рисунок 12 – Отзывы о пользователе

## 5.5. Сообщения

В веб-сервисе реализована система сообщений: пользователи могут вступать друг с другом в переписку. В эту вкладку можно перейти по соответствующей кнопке из левой панели. Список диалогов выглядит так:

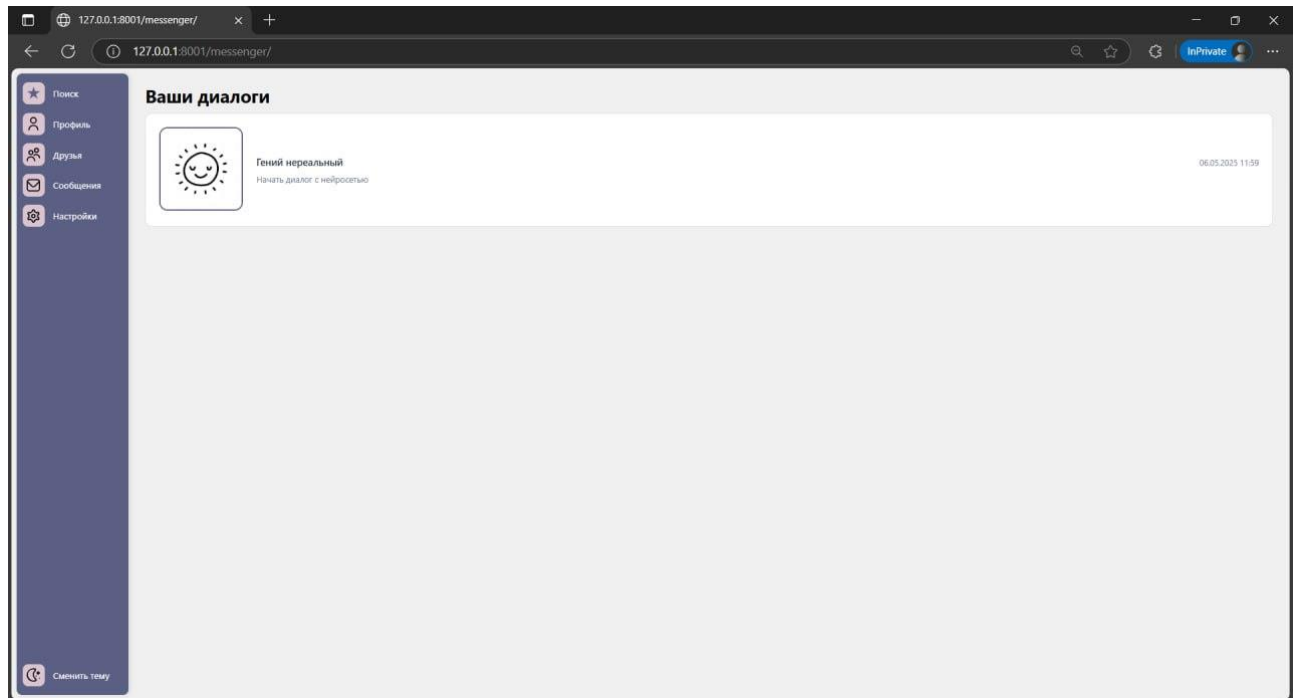


Рисунок 13 – Список диалогов

Диалог с пользователем выглядит так:

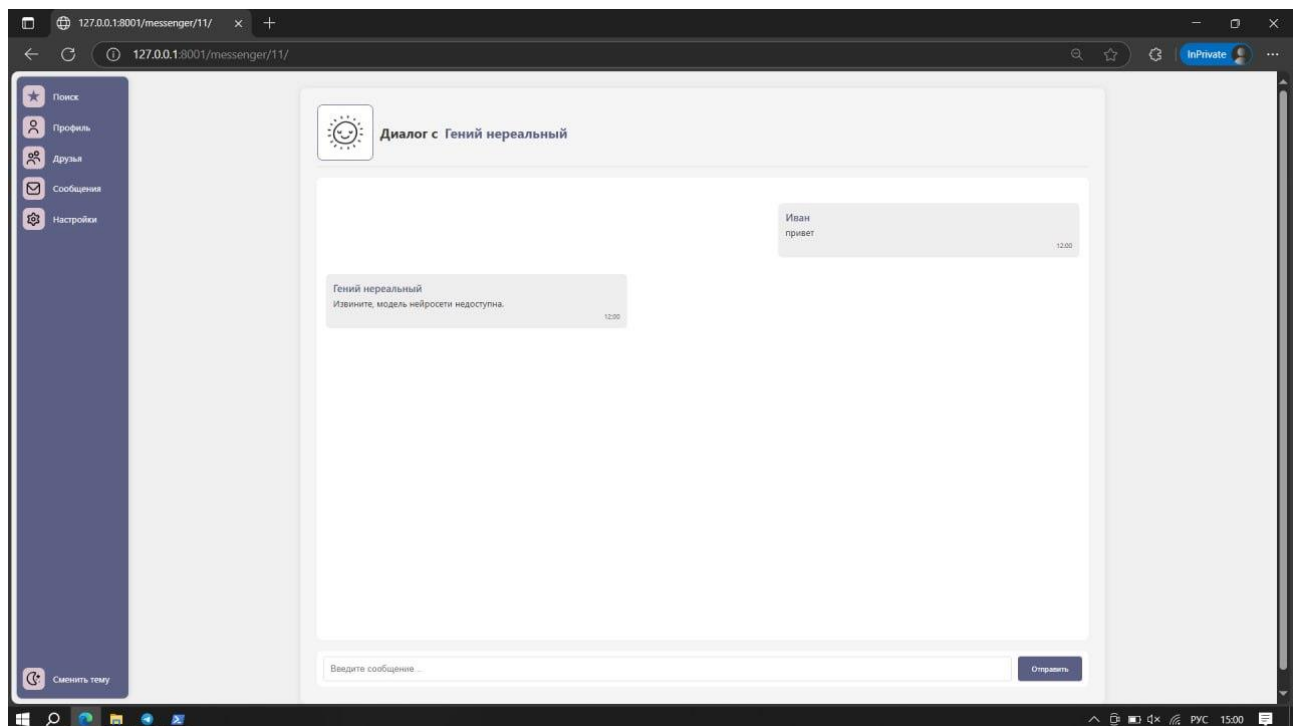


Рисунок 14 – Диалог

## 5.6. Друзья

Находя кандидатов в поиске, пользователь может с ними подружиться, отправив заявку в друзья. Список друзей, а также список заявок входящих и исходящих находится во вкладке Друзья, перейти в которую можно по соответствующей кнопке из левой панели сайта

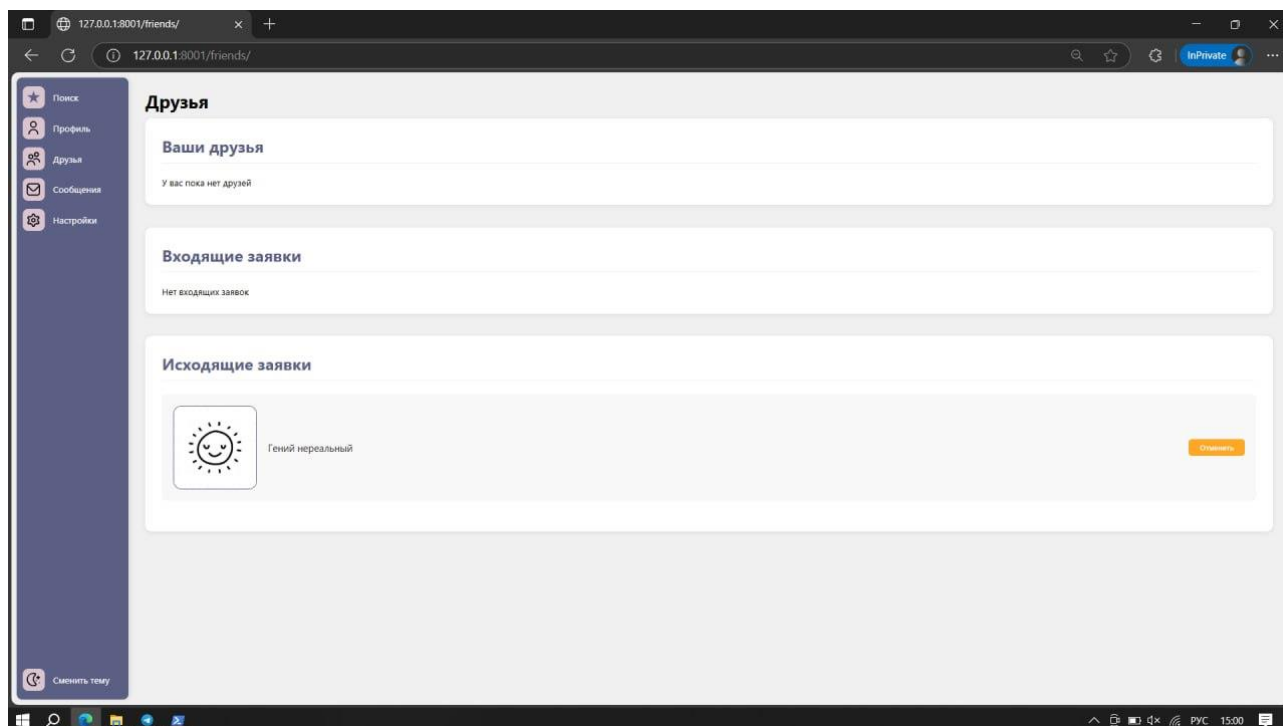


Рисунок 15 - Друзья

## **Заключение**

В ходе выполнения проекта была успешно разработана система подбора партнеров на основе метода анализа иерархий (МАИ), полностью соответствующая поставленным задачам. Реализованный алгоритм позволяет точно определять веса критериев и оценивать совместимость пользователей с учетом их индивидуальных предпочтений.

Для проектирования системы были разработаны UML-диаграммы, включая диаграмму прецедентов, диаграмму классов и диаграмму последовательностей, что позволило четко структурировать архитектуру приложения. Проведена комплексная отладка всех компонентов системы, включая backend-алгоритмы и frontend-интерфейс, обеспечив стабильную работу приложения.

Разработанное решение демонстрирует эффективное применение математических методов в социально-ориентированных проектах. Система готова к дальнейшему развитию и масштабированию функциональных возможностей.

## Список использованных источников

1. Саати Т. Принятие решений. Метод анализа иерархий. — М.: Радио и связь, 1993. — 278 с.
2. Кристофидес Н. Теория графов. Алгоритмический подход. — М.: Мир, 1978. — 432 с.
3. Дейтел П., Дейтел Х. Как программировать на Python. — СПб.: БХВ-Петербург, 2021. — 992 с.
4. Фаулер М. UML. Основы. — СПб.: Символ-Плюс, 2018. — 192 с.
5. Документация Django [Электронный ресурс]. — URL: <https://docs.djangoproject.com/> (дата обращения: 10.05.2024)
6. Официальная документация React [Электронный ресурс]. — URL: <https://react.dev/> (дата обращения: 10.05.2024)
7. Кнут Д. Искусство программирования. Том 1. Основные алгоритмы. — М.: Вильямс, 2020. — 720 с.
8. Таненбаум Э., Вудхалл А. Операционные системы. Разработка и реализация. — СПб.: Питер, 2015. — 1120 с.
9. Гамма Э. и др. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб.: Питер, 2022. — 496 с.
10. IEEE Standard for Software Requirements Specification (IEEE Std 830-1998). — 1998. — 40 p.