



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА – Российский технологический университет»  
**РТУ МИРЭА**

Институт кибербезопасности и цифровых технологий

Кафедра КБ-14 «Цифровые технологии обработки данных»

**РАБОТА ДОПУЩЕНА К ЗАЩИТЕ**

Заведующий кафедрой

И.А. Иванова

«13 июня 2024 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

по направлению подготовки бакалавров **09.03.02**

**«Информационные системы и технологии»**

на тему: Разработка приложения для поиска и возврата утерянных вещей

Обучающийся

*подпись*

Верхотуров Валерий Сергеевич

*фамилия, имя, отчество*

шифр

20Б0629

Группа

БСБО-05-20

Руководитель  
работы

*подпись*

к.т.н., зав. каф.  
*ученая степень, должность*

Иванова И.А.  
*фамилия, имя, отчество*

Консультант  
(при наличии)

*подпись*

к.э.н., доцент  
*ученая степень, должность*

Шеверева Е.А.  
*фамилия, имя, отчество*

**Москва 2024 г.**



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА – Российский технологический университет»  
**РТУ МИРЭА**

**Институт кибербезопасности и цифровых технологий**

Кафедра КБ-14 «Цифровые технологии обработки данных»

СОГЛАСОВАНО

Заведующий кафедрой

(подпись)

**И.А. Иванова**

УТВЕРЖДАЮ

Директор института

(подпись)

**В.В. Никонов**

«19»

апреля

2024 г.

«19»

апреля

2024 г.

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы**

Обучающийся

**Верхотуров Валерий Сергеевич**

Шифр

**20Б0629**

Направление

подготовки

**09.03.02**

**«Информационные системы и технологии»**

Группа

**БСБО-05-20**

**1. Тема выпускной квалификационной работы**

Разработка приложения для поиска и возврата утерянных вещей

**2. Цель и задачи выпускной квалификационной работы**

Цель работы: Разработать приложение для поиска и возврата утерянных вещей

Задачи работы:

Провести анализ существующих систем поиска утерянных вещей и выделить их преимущества и недостатки, определить требования к разрабатываемой системе, рассмотреть методы и технологии для реализации системы, определить процесс разработки системы и рассчитать смету затрат.

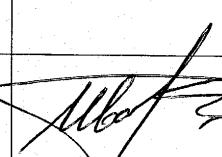
### 3. Этапы выпускной квалификационной работы

№ этапа	Содержание этапа выпускной квалификационной работы	Результат выполнения этапа ВКР	Срок выполнения
1.	<b>Аналитический раздел</b>	<i>в процессе</i>	
1.1.	Статистика потерянных и найденных вещей		
1.2.	Типы существующих решений для поиска и возврата утерянных вещей		
2.	<b>Специальный раздел</b>	<i>в процессе</i>	
2.1.	Требования к разрабатываемой системе		
2.2.	Функциональные требования		
3.	<b>Технологический раздел</b>	<i>в процессе</i>	
3.1.	Выбор технологий для реализации системы		
3.2.	Реализация модулей автоматизации процессов		
4.	<b>Экономический раздел</b>		

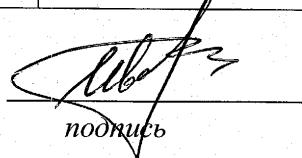
### 4. Перечень разрабатываемых документов и графических материалов

Цель и задачи ВКР, характеристика предметной области, метод и алгоритм решения задачи, апробация (моделирование или программное обеспечение), выводы.

### 5. Руководитель выпускной квалификационной работы:

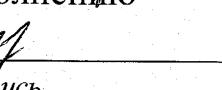
Функциональные обязанности	Должность в Университете	Фамилия, Имя, Отчество	Подпись
Руководитель ВКР	Зав. каф.	Иванова Ирина Алексеевна	

Задание выдал

Руководитель ВКР 

«19» апреля 2024 г.

Задание принял к исполнению

Обучающийся: 

*подпись*

«19» апреля 2024 г.

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	5
<b>1. АНАЛИТИЧЕСКИЙ РАЗДЕЛ</b>	6
1.1. Статистика потерянных и найденных вещей	6
1.2. Типы существующих решений для поиска и возврата утерянных вещей	6
1.3. Анализ существующих систем для поиска и возврата утерянных вещей	8
<b>2. СПЕЦИАЛЬНЫЙ РАЗДЕЛ</b>	14
2.1. Требования к разрабатываемой системе	14
2.1.1. Функциональные требования	14
2.1.2. Нефункциональные требования	16
2.2. Проектирование модулей автоматизации процессов	16
2.2.1. Модуль регистрации и авторизации пользователей	17
2.2.2. Модуль бесконечных лент объявлений потерянных, найденных вещей	17
2.2.3. Модуль добавления и поиска вещей	19
2.2.4. Модуль генерации описания объявлений	19
2.2.5. Непрерывная интеграция	20
<b>3. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ</b>	22
3.1. Выбор технологий для реализации системы	22
3.2. Реализация модулей автоматизации процессов	23
3.2.1. Модуль регистрации и авторизации пользователей	23
3.2.2. Модуль бесконечных лент объявлений потерянных, найденных вещей	25
3.2.3. Модуль добавления и поиска утерянных вещей	26
<b>4. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ</b>	29
4.1. Оценка готовности к коммерциализации программной разработки «Бюро находок»	29
4.2. Оценка фактора масштаба	31
4.3. Оценка фактора трудоемкости	32

4.4. Расчет полной стоимости программного продукта (в части трудозатрат) . . . . .	33
4.5. Оценка стоимости разработки и внедрения программного продукта	33
4.6. Расчет общего количества капитальных затрат ТСО (CAPEX) . . .	35
4.7. Калькуляция затрат на закупку специального оборудования или каналов связи . . . . .	36
4.8. Расчет общего количества операционных затрат ТСО (OPEX) . . .	36
4.9. Расчет совокупной стоимости владения (TCO) . . . . .	38
4.10. Оценка конкурентоспособности программного продукта . . . . .	39
<b>ЗАКЛЮЧЕНИЕ</b> . . . . .	<b>45</b>

## **ВВЕДЕНИЕ**

Поиск утерянных вещей является актуальной проблемой, которая возникает при различных обстоятельствах. Эта проблема может возникнуть в результате потери ключей, документов, мобильных телефонов, кошельков или других ценных или важных вещей [1, 2]. В связи с этим существует необходимость разработки системы, которая поможет людям вернуть утерянные вещи.

Целью данной работы является разработка системы для поиска утерянных вещей на основе анализа существующих систем и технологий, а также определение требований к системе и ее функциональности. Для достижения этой цели будут рассмотрены различные методы и технологии, которые могут быть использованы для создания такой системы.

В аналитическом разделе будет проведен анализ существующих систем поиска утерянных вещей и выделены их преимущества и недостатки. В специальном разделе будут определены требования к разрабатываемой системе, рассмотрены методы и технологии, которые можно использовать для реализации системы. Технологический раздел будет посвящен описанию процесса разработки и тестирования системы. В экономическом разделе будет приведен план разработки и расчет сметы затрат.

Таким образом, разработка системы для поиска утерянных вещей позволит создать удобный инструмент для поиска потерянных вещей, что приведет к уменьшению количества потерянных вещей и улучшению качества жизни людей.

# 1. АНАЛИТИЧЕСКИЙ РАЗДЕЛ

## 1.1. Статистика потерянных и найденных вещей

Для подтверждения актуальности и важности разрабатываемой системы, необходимо провести исследование рынка и определить основные проблемы и потребности пользователей. Одним из способов сбора информации является проведение опроса среди пользователей.

Одним из основных факторов, определяющих актуальность разрабатываемой системы является статистика потерянных и найденных вещей. Необходимо определить количество потерянных вещей в месяц, год и за весь период работы системы. Это поможет оценить нагрузку на систему и определить ее производительность.

Статистика, взятая с сайта «Столнаходок.рф» [5], утверждает, что только 20 % пользователей их сайта смогли установить и вернуть вещи. Также на рисунках 1 и 2 представлена гистограмма количества созданных объявлений за 2022 и 2023 года.

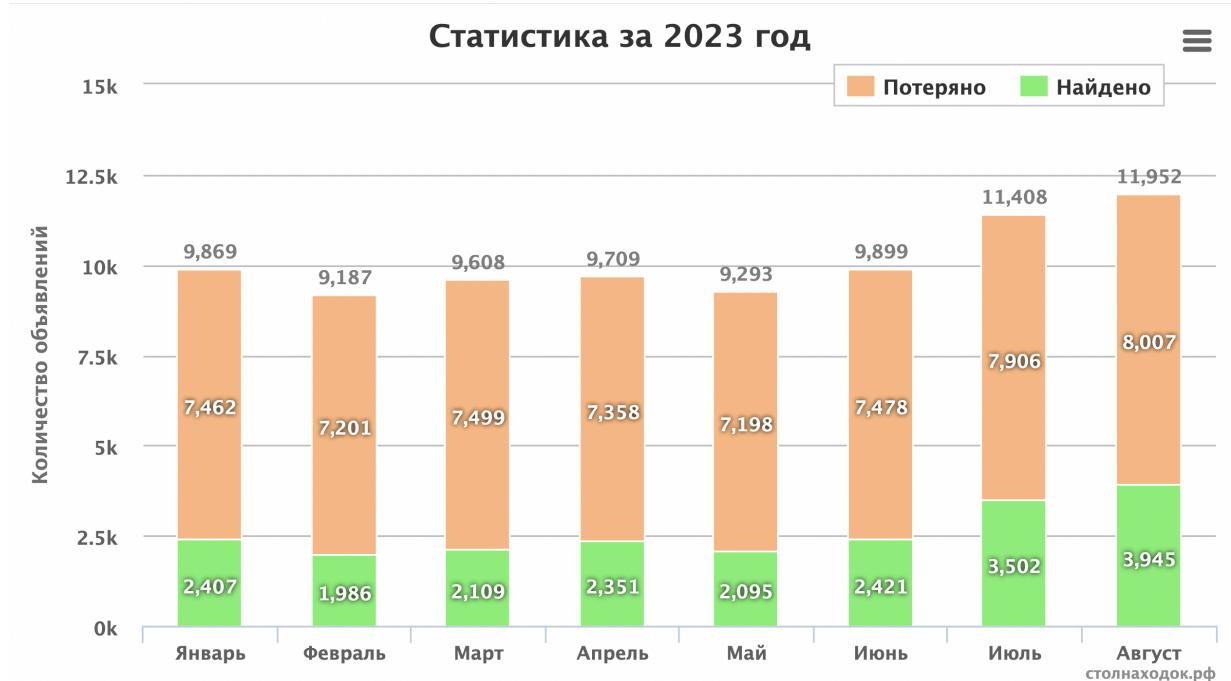


Рисунок 1 – Востребованность системы столнаходок.рф в 2023 году

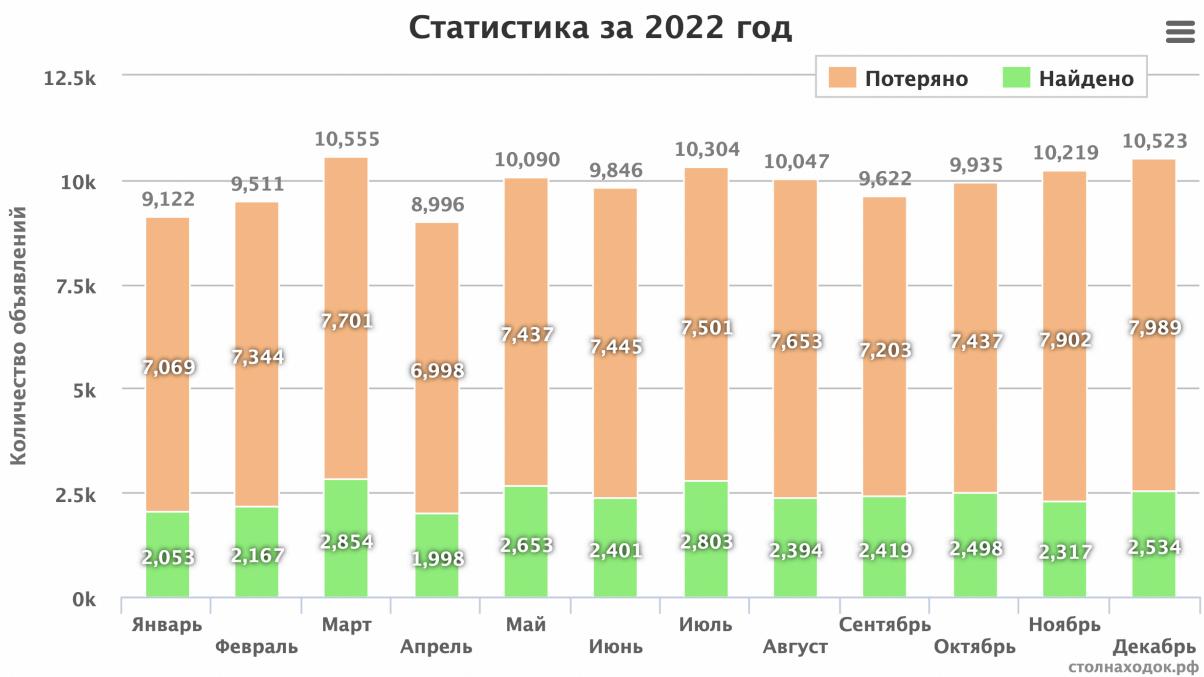


Рисунок 2 – Востребованность системы столнаходок.рф в 2022 году

## 1.2. Типы существующих решений для поиска и возврата утерянных вещей

Существует несколько типов существующих решений для поиска и возврата утерянных вещей. Ниже приведены некоторые из них:

- веб-сайты и мобильные приложения: «Бюро находок» [5, 6]. Эти сервисы предоставляют платформу, где люди могут регистрировать утерянные вещи и искать их владельцев. Пользователям предлагается создать объявления о найденных или потерянных вещах и связаться друг с другом, чтобы вернуть вещи. Некоторые сервисы предлагают добавить фотографии или описание вещей, чтобы облегчить поиск;
- технология RFID (Radio Frequency Identification) позволяет прикреплять RFID-метки к ценным объектам и определить владельца с помощью специальных считывателей [8, 9]. Это возможно благодаря использованию радиоволн, которые позволяют быстро определять местоположение потерянных вещей с помощью дополнительного программного обеспечения. Одним из наиболее распространенных применений технологии RFID является микрочипирование домашних животных или чипов для домашних животных. Эти микрочипы имплантируются ветеринарами и содержат информацию, касающуюся домашних животных, включая их имя, медицинские записи и

контактную информацию их владельцев. Если домашнее животное пропадает и его отправляют в спасательную службу или в приют, работник приюта сканирует животное на наличие микрочипа. Если у домашнего животного есть микрочип, работнику приюта достаточно одного телефонного звонка или поиска в Интернете, чтобы связаться с владельцами домашнего животного. Считается, что чипы для домашних животных более надежны, чем ошейники, которые можно упасть или снять;

— GPS-трекеры — это устройства с встроенным GPS-модулем. Они могут быть прикреплены практически к любому объекту, после чего его местоположение определяется через смартфон или компьютер по сети Интернет. При использовании приложения на смартфоне пользователь может получать уведомления о передвижении объекта и быстро определять его текущее местоположение;

— автоматизированные системы поиска утерянных предметов: некоторые организации, например, аэропорты и железнодорожные станции, используют системы обнаружения утерянных предметов. В этих системах используются технологии, такие как видеонаблюдение, детекторы движения и распознавание образов для отслеживания и возвращения потерянных предметов их владельцам.

Каждый из этих типов решений имеет свои преимущества и недостатки. Некоторые из них могут быть более подходящими для конкретных ситуаций, например, GPS-трекеры могут быть полезными при поиске утерянных вещей на открытой местности, в то время как RFID-метки могут быть более подходящими для использования внутри помещений. Веб-сайты и приложения «Бюро находок» предоставляют более универсальное решение, которое может быть использовано в различных ситуациях.

### **1.3. Анализ существующих систем для поиска и возврата утерянных вещей**

В настоящем разделе будет проведен обзор существующих сервисов и приложений, которые предлагают функциональность поиска и возврата утерянных вещей. Данный обзор позволит выявить основные преимущества и недостатки этих сервисов, а также определить потенциальные возможности для

улучшения их функциональности.

«столнаходок.рф» [5] — это один из наиболее популярных веб-сервисов, предоставляющих возможность объявлять о потерянных и найденных предметах. Сервис имеет простой и интуитивно понятный интерфейс, позволяющий пользователям быстро разместить информацию о потерянных вещах и связаться с владельцами найденных предметов, примеры пользовательского интерфейса представлены на рис. 3, 4. Однако, отсутствие системы уведомлений и неудобное сопоставление объявлений ограничивают его функциональность.

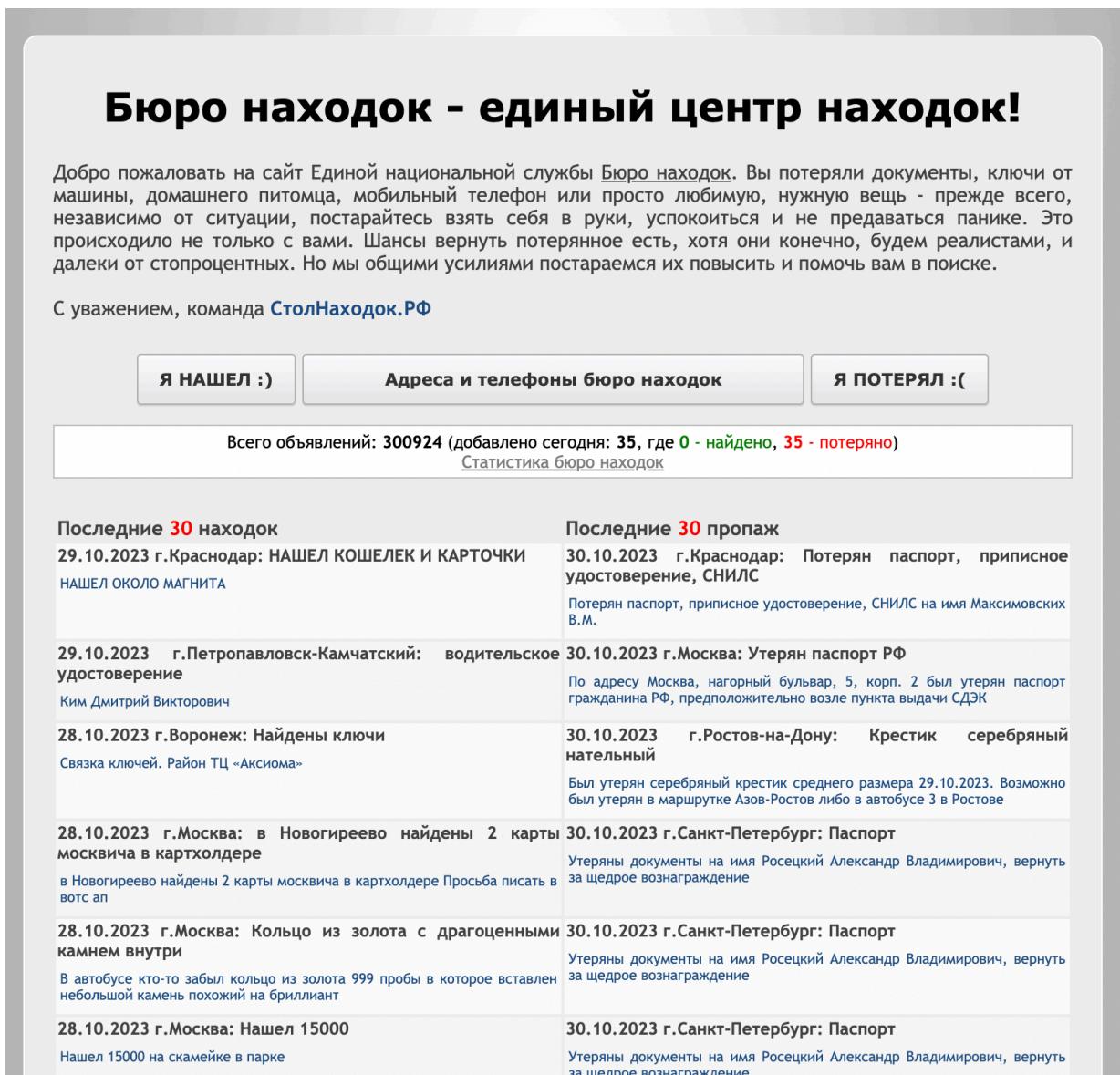


Рисунок 3 – Скриншот системы «столнаходок.рф»

«Стол находок» [5] — это мобильное приложение, разработанное для операционных систем iOS и Android. Оно предлагает функцию отслеживания

БЮРО НАХОДОК
НАХОДОК  
 Информационно Поисковый Портал РФ

Выберите регион и введите ключевое слово для поиска:  
 Москва  находки    
 Пример: Москва, находки, паспорт Иванов Иван Иванович (расширенный поиск)

Вы сейчас здесь: [Бюро находок](#) > [Новое объявление](#)
Гость (вход/регистрация)

**Внимание**

При добавлении объявления запрещается в тексте указывать контактные данные и ссылки (в том числе номера телефонов, адреса электронной почты и сайтов).

**Убедительная просьба, при размещении объявления о находке или пропаже документов, не указывайте серийные номера, так как этими данными могут воспользоваться третьи лица в корыстных или преступных целях.**

**Случаи мошенничества! Предупрежден - значит вооружен!**

Пожалуйста, не дублируйте свои объявления.

Размещаемые объявления проверяются на соответствие правилам [buro.nahodok.ru - Условия и правила](#).

**Статистика бюро находок**

Группы	Найдено	Потеряно
1. Документы	<a href="#">17752</a>	<a href="#">185029</a>
2. Драгоценности	<a href="#">1957</a>	<a href="#">6045</a>
3. Животные	<a href="#">3285</a>	<a href="#">2898</a>
4. Ключи	<a href="#">5326</a>	<a href="#">15576</a>
5. Прочее	<a href="#">4423</a>	<a href="#">16803</a>
6. Сумки	<a href="#">780</a>	<a href="#">16339</a>
7. Телефоны	<a href="#">1636</a>	<a href="#">14684</a>
8. Электроника	<a href="#">1391</a>	<a href="#">6983</a>

Добавлено  
сегодня: **35**  
  
 Всего объявлений: **300924**  
 Нашли  
владельцев: **69106**

[Подать объявление](#)

**Случаи мошенничества**

**Уважаемые пользователи!**  
 На сайте участились случаи мошенничества, о которых нам ежедневно сообщают наши пользователи.

**Новое объявление**

Регион \*  **Москва**

Тип Объявления \*  найдено  потеряно

Группа \*  **Документы**

Заголовок объявления \*

Не сообщайте номер паспорта, банковской карты и другие персональные данные!

Контактное лицо

Телефон

Адрес эл.почты (E-mail) \*  **на сайте отображается!**

Фотография  **Файл не выбран**

Защитный код \*  **30-2=?** чему равен результат?

[Добавить объявление](#)

**Нажимая кнопку «Добавить объявление», Вы принимаете условия [Пользовательского соглашения](#).**

**для публикации объявления необходимо:**

Рисунок 4 – Скриншот системы «столнаходок.рф»

утерянных предметов через GPS-модуль смартфона, представлено на рис. 5, 6. Пользователи могут отмечать свои вещи на карте и получать уведомления, когда они находятся рядом с утерянным предметом. Однако, ограничение использования только наличием смартфона с GPS-модулем и низкая точность определения местоположения представляют существенные ограничения данного приложения.

«Lost Property Office» [10] — это веб-сервис, предоставляемый государственными организациями и органами правопорядка, см. рис. 7. Сервис позволяет пользователям сообщать о потерянных и найденных предметах, а также предоставляет информацию о процедуре возврата утерянных вещей.

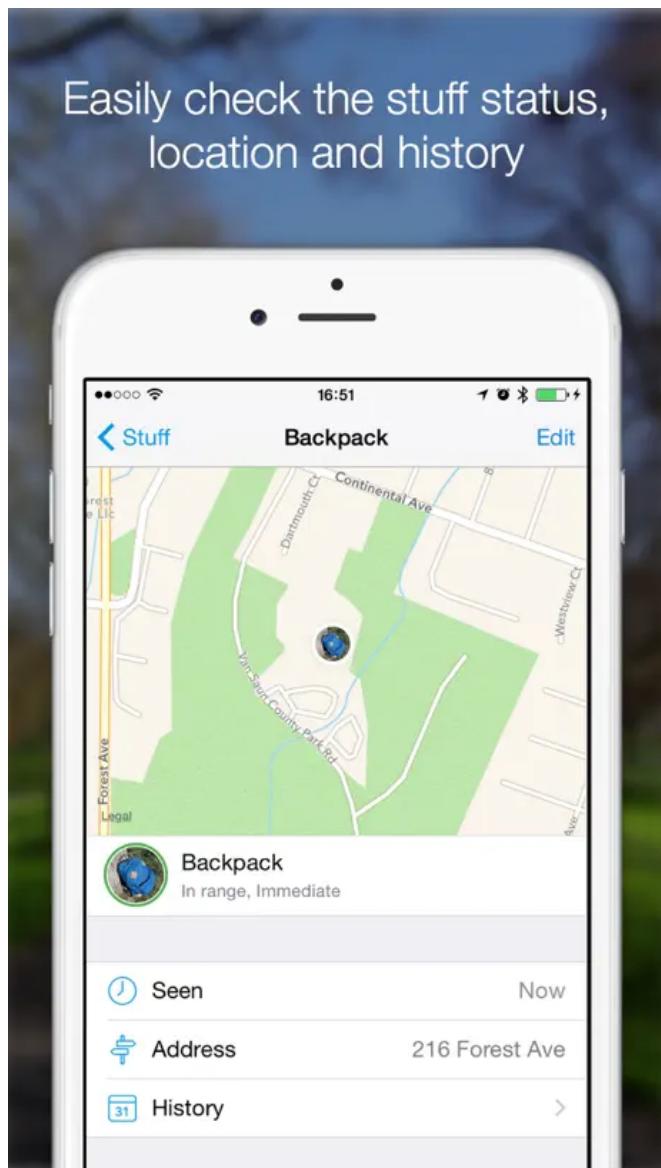


Рисунок 5 – Скриншот системы «Find My Stuff»

Однако, ограниченный доступ к сервису и неудобный процесс регистрации и подачи заявки являются значительными недостатками данного сервиса.

На основании проведенного обзора можно сделать вывод, что существующие веб-сервисы и приложения для поиска и возврата утерянных вещей имеют некоторые преимущества, но также недостатки, которые ограничивают их функциональность и удобство использования. Веб-сервис Бюро находок будет разработан с учетом этих недостатков и предлагать более удобное взаимодействие между пользователями и сервисом.

Ниже приведена сравнительная таблица 1 основных характеристик и функций приведенных выше аналогов:

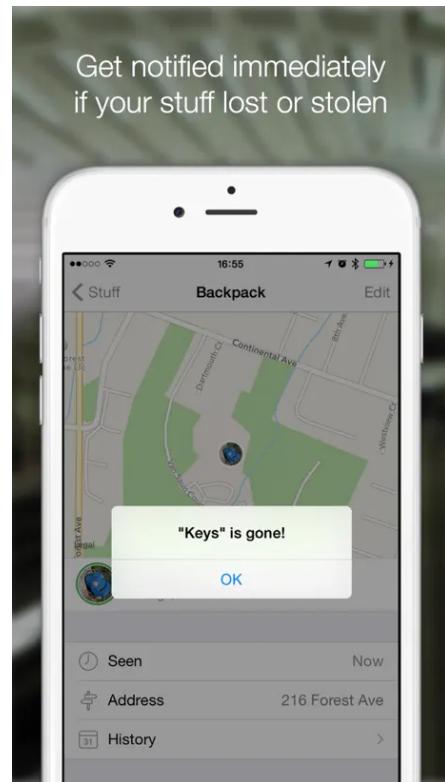


Рисунок 6 – Скриншот системы «Find My Stuff»

Таблица 1 – Сравнительная таблица аналогов

Сервис / Приложение	Интерфейс и удобство использования	Оповещения	Точность определения местоположения	Удобство регистрации и подача заявки	Доступность
столна-ходок.рф	Простой и интуитивно понятный интерфейс	Отсутствуют	Неопределено	Простой процесс регистрации	Широкий доступ
Find My Stuff	Простой и интуитивно понятный интерфейс	Оповещения через уведомления	Низкая точность	Простой процесс регистрации	Доступен только на смартфонах с GPS
Lost Property Office	Неудобный процесс регистрации и подачи заявки	Отсутствуют	Неопределено	Неудобный процесс регистрации и подачи заявки	Ограниченнный доступ

## UK Parliament

[UK Parliament](#) > [Visit](#) > [Plan your visit](#) > [Shops, cafe and other visitor facilities](#) > Lost Property Office

### Lost Property Office

Lost property is managed by staff in the Hallkeeper's Lodge, which is situated off St Stephen's Hall. This is where lost property should be handed in and where enquiries can be made.



If you think you have lost property during your visit to Parliament you can call into the Hallkeeper's Lodge. After your visit, please call 020 7219 4626 to see if your lost items have been handed in. You will be asked to provide details of the time and date of your visit.

[Hallkeeper's Lodge opening times](#)

Рисунок 7 – Скриншот системы «Lost Property Office»

### Вывод по разделу

В аналитической части работы проведен детальный анализ существующих веб-ресурсов и приложений, предназначенных для поиска и возвращения утерянных вещей. Были изучены и проанализированы их функциональность, характеристики, преимущества и ограничения.

Одним из наиболее популярных и востребованных решений в данной сфере являются веб-сервисы и приложения «Бюро находок». Они предоставляют пользователям платформу для регистрации утерянных вещей и связи с их владельцами, что упрощает процесс поиска и возвращения потерянных предметов.

## **2. СПЕЦИАЛЬНЫЙ РАЗДЕЛ**

### **2.1. Требования к разрабатываемой системе**

Требования к разрабатываемой системе представляют собой совокупность параметров и характеристик, которыми должно обладать разрабатываемое приложение для достижения поставленных целей и решения задач. Они определяют функциональность системы, ее поведение, а также условия, необходимые для ее корректной работы. Требования подразделяются на функциональные и нефункциональные.

Функциональные требования описывают специфические функции или действия, которые должна выполнять система. В контексте разрабатываемого приложения для поиска и возврата утерянных вещей, это могут быть функции регистрации и авторизации пользователей, поиска утерянных вещей, добавления информации о утерянных вещах, связи между пользователями и системы уведомлений.

Нефункциональные требования определяют качественные характеристики системы, такие как производительность, безопасность, доступность, удобство использования, совместимость, масштабируемость, тестирование и документация.

Требования к разрабатываемой системе играют ключевую роль в процессе разработки приложения. Они служат основой для проектирования, реализации и тестирования системы. Без четко определенных требований невозможно разработать эффективное и надежное приложение, которое будет отвечать потребностям пользователей и бизнес-задачам.

В контексте курсовой работы на тему “Разработка приложения для поиска и возврата утерянных вещей”, требования к разрабатываемой системе позволяют сформулировать и структурировать задачи, которые должно решать приложение, а также определить параметры, необходимые для его успешной работы. Они служат основой для дальнейшего проектирования и разработки приложения, а также для оценки его эффективности и качества после внедрения.

#### *2.1.1. Функциональные требования*

Функциональные требования включают в себя пункты:

- приложение должно предоставлять возможность регистрации и авторизации пользователей;
- приложение должно предоставлять возможность поиска утерянных вещей по различным критериям (например, по типу вещи, по месту утери и т.д.);
- пользователи должны иметь возможность добавлять информацию о утерянных вещах, включая описание, фотографии и место утери;
- приложение должно предоставлять функционал для связи между пользователем, который нашел вещь, и пользователем, который ее потерял;
- приложение должно иметь систему уведомлений, которая будет информировать пользователей о новых найденных вещах, соответствующих их критериям поиска.

В соответствии с требованиями была составлена ER-диаграмма, которая представлена на рис. 8. Пользователь регистрируется посредством OAuth, при этом заполняются таблицы Account и User. Пользователь заполняет свои социальные сети UserSocialNetwork. Пользователь заполняет форму с потерянной или найденной вещью в LostAndFoundItem, и привязывает к карточки вещи соц. сети, по которой с ним можно связаться.

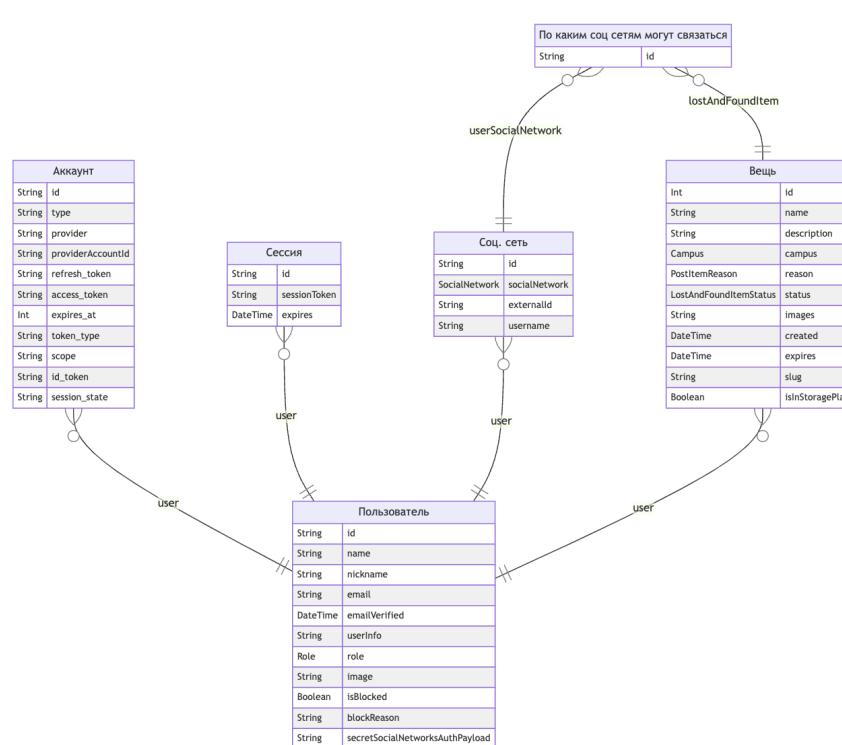


Рисунок 8 – ER-диаграмма системы

### *2.1.2. Нефункциональные требования*

Нефункциональные требования включают в себя пункты:

- приложение должно обеспечивать быстрый поиск и отображение результатов, а также быстрое добавление информации о утерянных вещах;
- все данные пользователей должны быть защищены;
- приложение должно быть доступно для использования 24/7;
- интерфейс приложения должен быть интуитивно понятным и удобным для пользователей разного уровня компьютерной грамотности;
- приложение должно быть совместимо с основными операционными системами (iOS, Android) и браузерами (Chrome, Firefox, Safari, Edge);
- приложение должно быть способно обслуживать большое количество пользователей одновременно без снижения производительности;
- приложение должно быть тщательно протестировано на наличие ошибок и уязвимостей перед запуском.

Клиентское приложение работает в вебе, использует кросс-платформенные технологии (JS, HTML, CSS). Защита пользователя возложено на независимый сервер авторизации.

## **2.2. Проектирование модулей автоматизации процессов**

Проектирование модулей автоматизации для приложения по поиску и возврату утерянных вещей требует разработки и планирования (см. рис. 9). Вот как можно подойти к проектированию каждого из ключевых модулей:

1. модуль регистрации и авторизации пользователей отвечает за создание учетных записей пользователей и их аутентификацию. Основные аспекты:
  - форма регистрации — сбор данных о пользователе, таких как имя, email, и пароль;
  - валидация данных — проверка введенных данных на корректность и уникальность;
  - безопасность — шифрование паролей и защита данных пользователя;
2. модуль добавления и поиска утерянных вещей позволяет пользователям добавлять информацию о потерянных вещах и искать их в базе данных. Основные аспекты:

- форма добавления — пользователи могут вводить описание вещи, место и время утери.
- база данных — хранение информации о всех зарегистрированных вещах;
- поиск — возможность поиска вещей по различным параметрам, таким как категория, местоположение, и время утери;
- фильтрация и сортировка — улучшение пользовательского опыта за счет фильтров и сортировки результатов поиска;

### 3. Система уведомлений. Основные аспекты:

- email-уведомления: отправка регулярных обновлений и важных уведомлений на email.

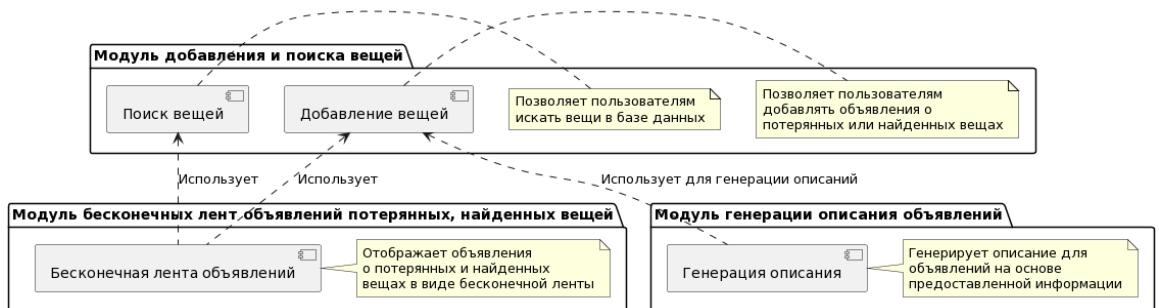


Рисунок 9 – Диаграмма компонентов системы

Схема дерева переходов по сайту будет представлена на рис. 10.

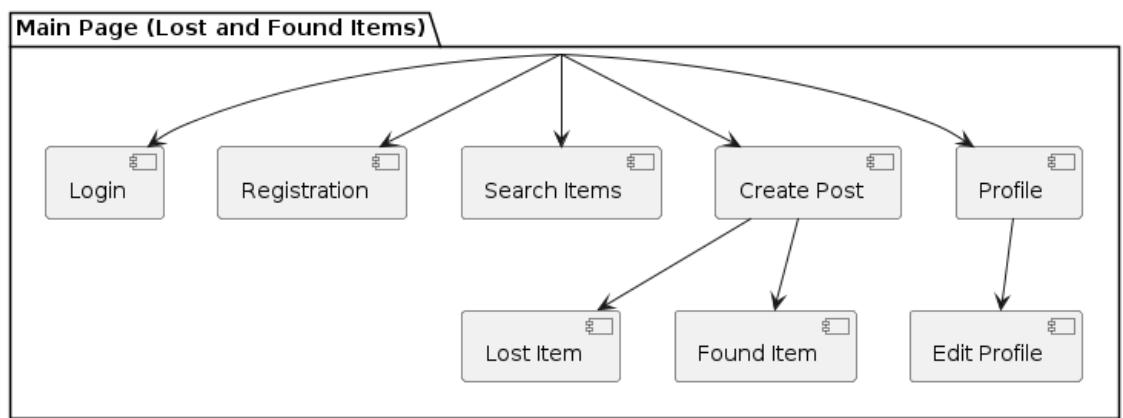


Рисунок 10 – Схема дерева переходов по сайту

#### 2.2.1. Модуль регистрации и авторизации пользователей

Этот модуль предназначен для создания и поддержки учетных записей пользователей. Он должен включать функции регистрации, авторизации через сервер посредника (сервер авторизации РТУ МИРЭА).

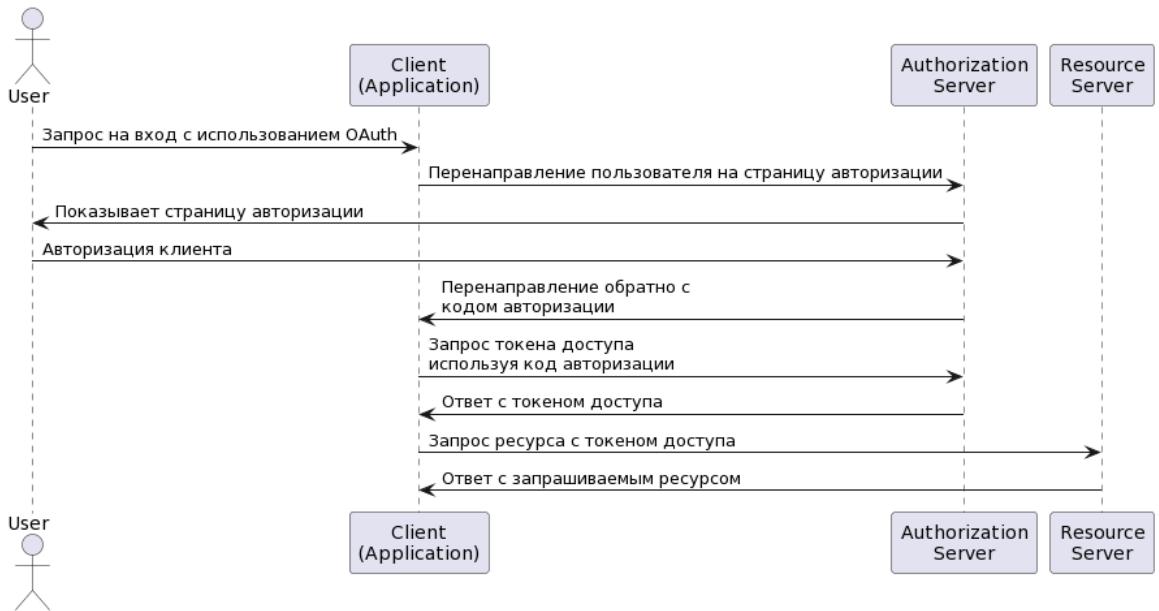


Рисунок 11 – Диаграмма последовательностей авторизации

## 2.2.2. Модуль бесконечных лент объявлений потерянных, найденных вещей

Модуль бесконечных лент объявлений представляет собой ключевой элемент приложения для поиска и возврата утерянных вещей. Он предназначен для отображения объявлений о потерянных и найденных вещах в формате бесконечной ленты, обеспечивая пользователю удобный и непрерывный доступ к информации.

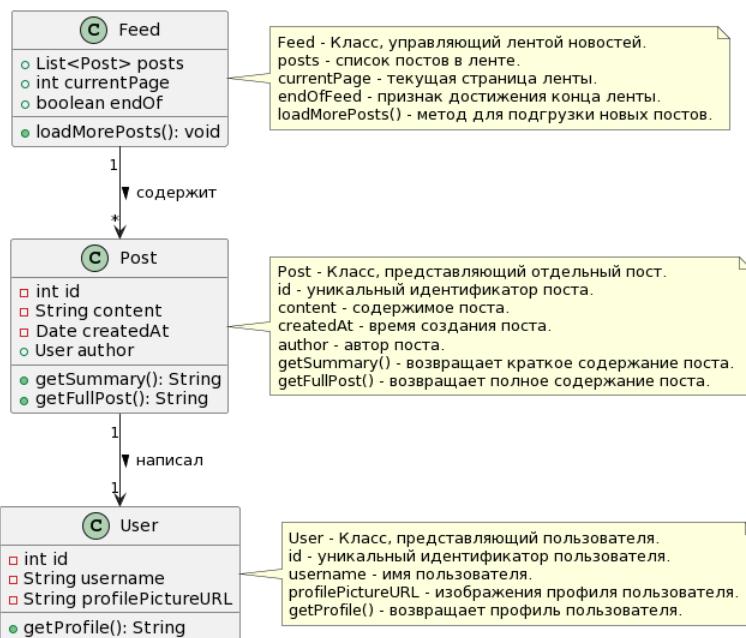


Рисунок 12 – Диаграмма классов бесконечной ленты

### 2.2.3. Модуль добавления и поиска вещей

Этот модуль отвечает за добавление информации о утерянных вещах в базу данных и поиск по этой базе. Он должен предоставлять пользователю возможность добавлять описание, фотографии и место утери вещи, а также осуществлять поиск по различным критериям 13.

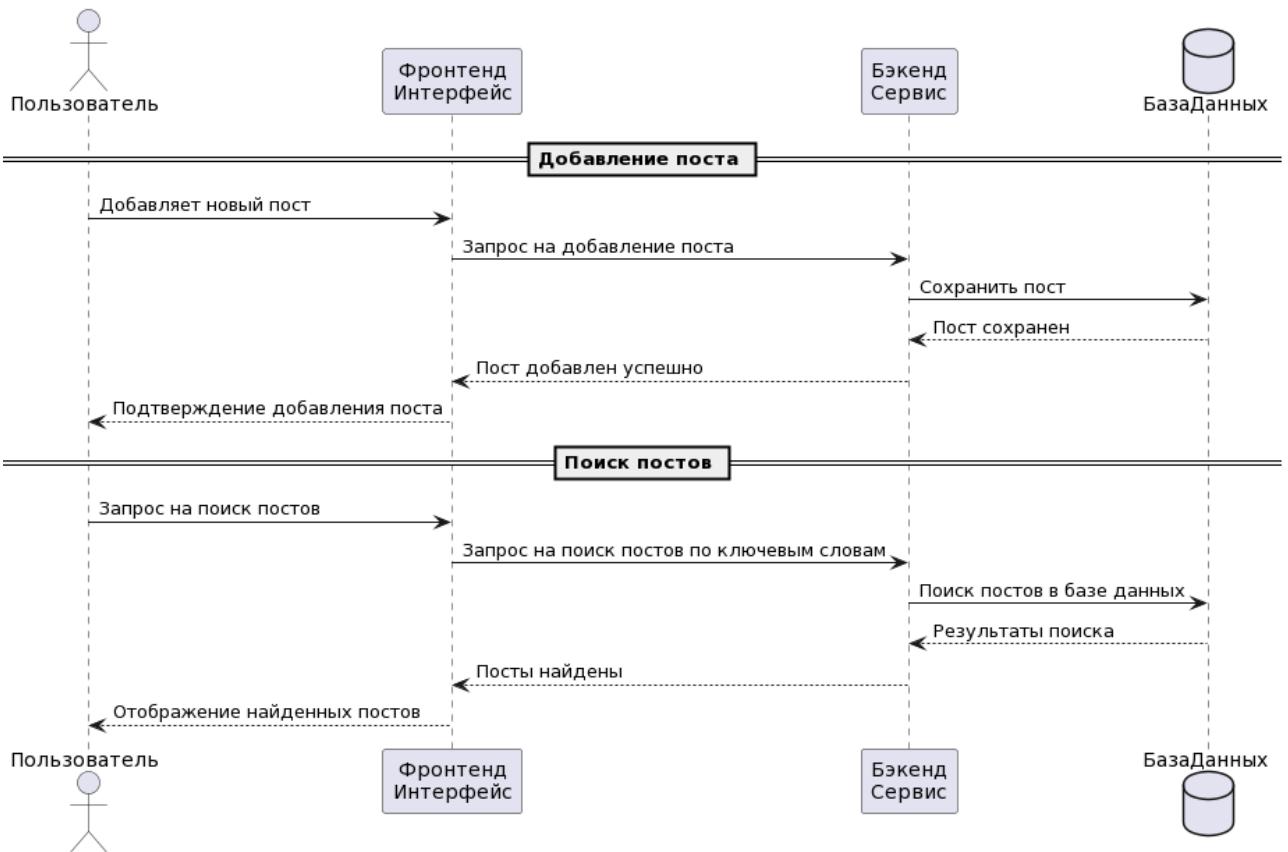


Рисунок 13 – Диаграмма последовательностей добавления и поиска вещей

Состоит из шагов:

- запрос на добавление нового поста;
- добавление нового поста в БД;
- подтверждение добавления нового поста в БД;
- запрос на поиск поста в БД;
- поиск поста в БД;
- возвращение релевантных постов.

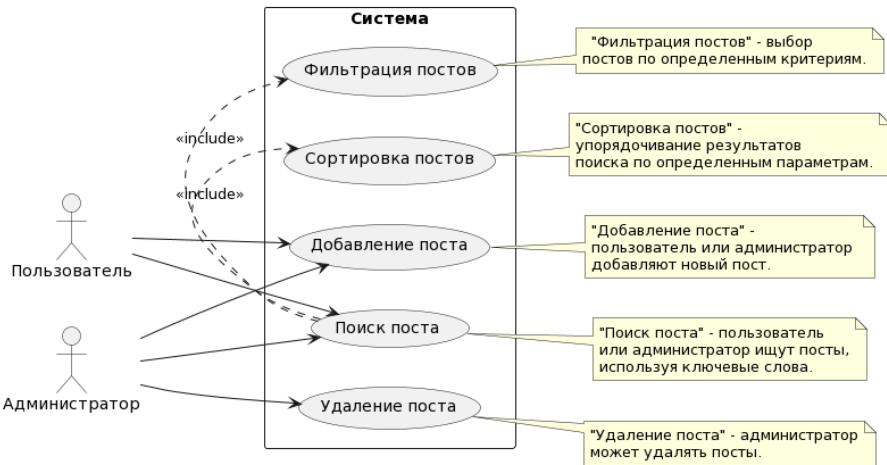


Рисунок 14 – Диаграмма вариантов использования добавления и поиска вещей

#### 2.2.4. Модуль генерации описания объявлений

Модуль генерации описания объявлений 15 является важным компонентом приложения для поиска и возврата утерянных вещей. Он предназначен для автоматического создания описаний объявлений на основе введенных пользователем данных, что облегчает процесс создания объявлений и повышает их качество.

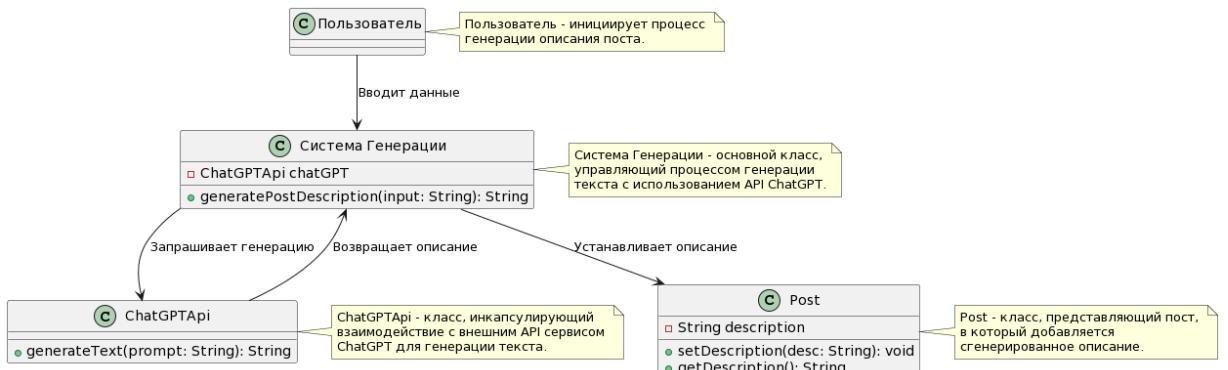


Рисунок 15 – Диаграмма классов генерации описания вещей

#### 2.2.5. Непрерывная интеграция

Для непрерывной интеграции использовалась платформа GitHub Actions. При изменении master ветки начинает работать скрипт на рулере, который собирает Docker образ и загружает его в частный Yandex Cloud Registry. Далее машина, на которой запущено приложение, скачивает образ, скачивает образ и запускает контейнер на основной машине. Диаграмма представлена на рис. 16.

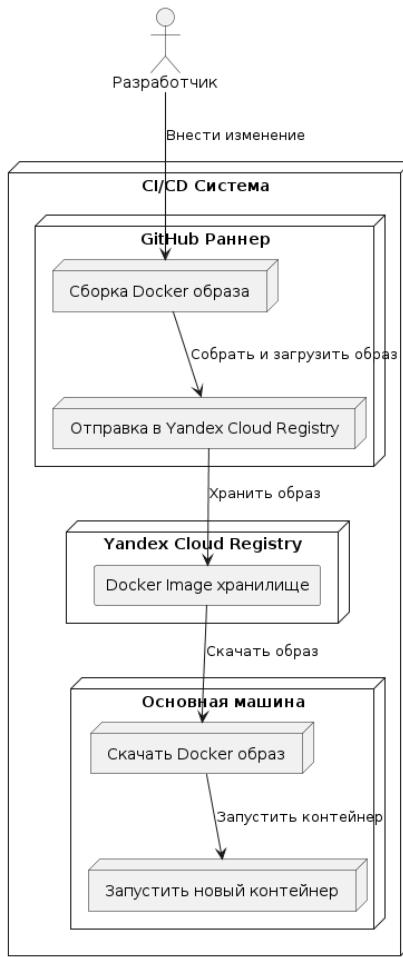


Рисунок 16 – Диаграмма выкладки

### **Вывод по разделу**

Проектирование модулей автоматизации процессов является важным этапом в разработке приложения для поиска и возврата утерянных вещей. Каждый из модулей, включая модуль регистрации и авторизации пользователей, модуль бесконечных лент объявлений, модуль добавления и поиска утерянных вещей и модуль генерации описания объявлений, играет свою уникальную роль в обеспечении функциональности приложения.

Каждый из этих модулей важен для обеспечения удобства использования приложения, и их совместная работа позволяет создать надежное и функциональное приложение для поиска и возврата утерянных вещей.

### **3. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ**

#### **3.1. Выбор технологий для реализации системы**

Были выбрали надежный и современный набор технологий для своего веб-сервиса «Бюро находок». Вот краткий обзор некоторых из наиболее важных технологий в вашем стеке и почему они полезны для проекта:

- Next.js — это мощная платформа React, которая обеспечивает рендеринг на стороне сервера и генерацию статических сайтов, что может помочь улучшить производительность и SEO веб-приложения. Он также поддерживает маршруты API, что упрощает внутреннюю разработку в рамках одного проекта [11];
- React — популярная библиотека JavaScript для создания пользовательских интерфейсов, особенно одностраничных приложений, где требуется быстрый ответ на взаимодействие с пользователем. Это помогает в создании повторно используемых компонентов пользовательского интерфейса [12];
- tRPC позволяет создавать типобезопасные API без необходимости писать схемы или генерировать типы. Он использует TypeScript, чтобы обеспечить проверку типов входных и выходных данных API, что снижает вероятность ошибок во время выполнения и повышает производительность разработчиков [13];
- Prisma — это ORM (реляционное сопоставление объектов), которое упрощает доступ к базе данных и обеспечивает безопасность типов. Адаптер Prisma для NextAuth.js упрощает создание надежных систем аутентификации, привязанных непосредственно к схеме вашей базы данных [14];
- NextAuth.js — комплексное решение для аутентификации в приложениях Next.js. Он поддерживает различные стратегии входа в систему с минимальной настройкой, повышая безопасность приложения [15];
- React Query — это библиотека для эффективного извлечения, кэширования и обновления данных в приложениях React. Это помогает управлять состоянием сервера и синхронизировать пользовательский интерфейс с данными без использования бойлерплейт кода [16];
- Zustand — это минималистичное решение для управления состоянием, которое работает «из коробки» с React. Это просто, быстро и не

- требует шаблонного кода, как это делают другие библиотеки управления состоянием [18];
- Tailwind CSS — это ориентированная на утилиты CSS-инфраструктура, наполненная классами, которые можно создавать для создания любого дизайна прямо в вашей разметке. Flowbite расширяет Tailwind дополнительными компонентами, ускоряя создание красивых и отзывчивых интерфейсов [19];
  - AWS SDK позволяет легко использовать веб-сервисы Amazon, такие как S3 для хранения, SES для отправки электронной почты и т. д., прямо из приложения. Это имеет решающее значение для масштабирования и управления инфраструктурой [20];
  - Headless UI — библиотека, которая предоставляет полностью неоформленные, полностью доступные компоненты пользовательского интерфейса, предназначенные для прекрасной интеграции с Tailwind CSS. Это помогает в создании пользовательских и доступных выпадающих списков, модальных окон и т.д [21].

## 3.2. Реализация модулей автоматизации процессов

### 3.2.1. Модуль регистрации и авторизации пользователей

Модуль представляет реализацию OAuth с единовременной передачей данных пользователя.

OAuth (Open Authorization) [22] — это открытый стандарт авторизации, который позволяет пользователям предоставлять безопасный делегированный доступ к своим учетным записям на различных сервисах, не раскрывая свои пароли. Реализация клиента OAuth включает несколько ключевых этапов и может варьироваться в зависимости от используемой версии OAuth (например, OAuth 1.0 или OAuth 2.0). Страница логина представлена на рис. 17.

Ниже описан процесс реализации клиента для OAuth 2.0.

1. Перед тем как начать, вам нужно зарегистрировать ваше приложение на платформе, которую вы хотите использовать (например, Google, Facebook, GitHub). В процессе регистрации вам нужно будет указать:

- Название приложения;
- URL, на который будет перенаправлен пользователь после авторизации (callback URL);

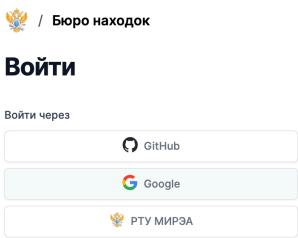


Рисунок 17 – Страница логина

После регистрации вы получите client ID и client secret, которые будут использоваться для аутентификации вашего приложения.

2. Клиент должен перенаправить пользователя на URL авторизации, предоставленный сервисом, с необходимыми параметрами запроса:

- *response\_type* (обычно *code*);
- *client\_id*;
- *redirect\_uri*;
- *scope* (необязательно, зависит от того, к каким данным приложение стремится получить доступ);
- *state* (рекомендуется для защиты от CSRF-атак);

Пользователь входит в систему (если еще не вошел) и подтверждает доступ к своим данным. После этого сервис перенаправляет пользователя обратно на *redirect\_uri* с кодом авторизации в параметрах URL;

3. После получения кода авторизации, приложение делает запрос на сервер сервиса для обмена кода на токен доступа. Этот запрос должен содержать:

- *grant\_type* (обычно *authorization\_code*);
- *code* (полученный код авторизации);
- *redirect\_uri*;
- *client\_id*;

— *client\_secret*;

Если запрос успешен, сервер ответит JSON-объектом, содержащим *access\_token* (и возможно *refresh\_token* и другие данные);

4. Токен доступа используется для доступа к защищенным ресурсам пользователя. Он добавляется в заголовок HTTP-запроса как *Authorization: Bearer <token>*;

5. Если получен *refresh\_token*, его можно использовать для получения нового *access\_token* после его истечения без необходимости повторной аутентификации пользователя.

### 3.2.2. Модуль бесконечных лент объявлений потерянных, найденных вещей

Для создания бесконечной ленты объявлений в React можно воспользоваться следующим подходом, основанным на базовых принципах React и встроенных возможностях JavaScript:

- создается состояние в компоненте, которое будет хранить массив объявлений и переменную для отслеживания, загружены ли все данные;
- реализуется функция, которая будет загружать порции данных (например, по 10 объявлений за раз). Эта функция должна обновлять ваше состояние, добавляя новые объявления к уже загруженным;
- добавляется обработчик события прокрутки к элементу, в котором отображаются объявления. Когда пользователь достигает конца списка, вызывайте функцию загрузки данных;
- необходимо убедиться, что новые данные загружаются только когда предыдущая загрузка завершена, чтобы избежать повторных запросов. Также нужно проверить, не достигнут ли конец списка данных, чтобы прекратить загрузку новых данных;
- используется метод *map* для преобразования массива объявлений в JSX-элементы, которые будут отображаться в пользовательском интерфейсе;
- используется техники оптимизации, такие как *React.memo* для компонентов объявлений, чтобы избежать ненужныхrerендеров при добавлении новых объявлений;
- также нужно добавить обработку ошибок для ситуаций, когда загрузка данных может завершиться неудачей, например, при проблемах с сетью.

Реализация бесконечной ленты объявлений представлена на рис. 18.

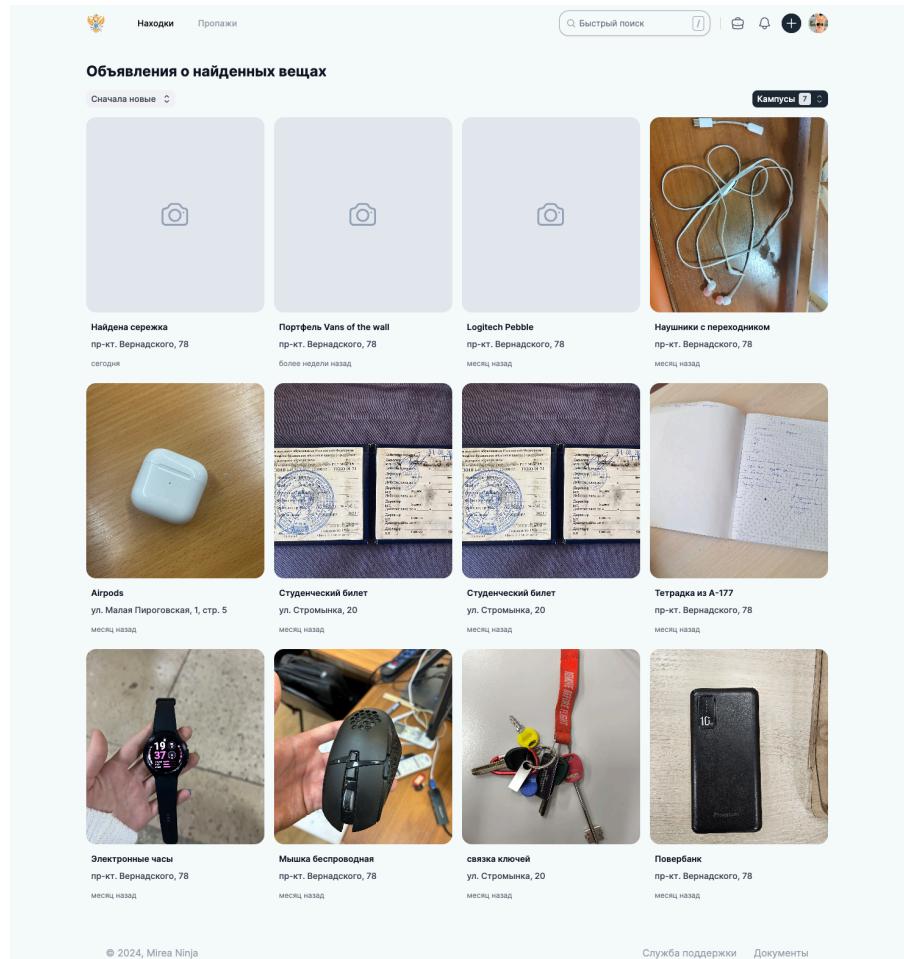


Рисунок 18 – Страница бесконечной ленты объявлений

### 3.2.3. Модуль добавления и поиска утерянных вещей

Для добавления утерянных вещей реализована форма с вводом названия, описания, возможностью загружать картинки в объектное хранилище. Форма представлена на рисунке 19.

Полнотекстовый поиск по названиям и описаниям происходит по названиям и описанию объявлений. Реализация клиентской части поиска представлена на рисунке 20.

**Сообщить о пропаже**

**Пропажа**  
Опишите пропажу, чтобы можно было определить, что она Ваша

**Фотографии:**

[Image upload area]   
Загрузите фотографии или перетащите  
PNG, JPG, JPEG, GIF, WEBP до 10MB

**Название:** [Text input field]

**Описание:** [Text input field]  
Место нахождения или контактная информация

**Кампус:** [Text input field]  
пр-кт. Вернадского, 78

Я отнес(ла) находку в А-131  
А-131 - это аудитория, в которую вы можете отнести находку. Сотрудники студенческого союза заберут её и позаботятся о том, чтобы с предметом ничего не случилось!  
Мы рекомендуем отнести найденный предмет туда  
[Аудитория на карте >](#)

[Отменить](#) [Сохранить](#)

© 2024, Mirea Ninja [Служба поддержки](#) [Документы](#)

Рисунок 19 – Форма ввода утерянных вещей

**Все объявления**

Найдена сережка  
В коридоре корпуса А найдена золотая сережка. Отнесли в А-430. За фотографией пишите в соц сети

**Бумажник**  
Чёрный бумажник. Внутри пропуск на имя Новикова Артемия Сергеевича, найден на пульке в чилл-аут зоне на 1 эт. корпуса А Вернадского 78. Лежит на охране.

AirPods 1 поколения  
Валились в чилл зоне на диване

**Золотое кольцо с красными камнями**  
Найдено в лектории А-4 после 3 пары

**Пропажи**

Найдена сережка Чёрная капка с ушками Портфель Vans of the wall Блок питания от ноутбука MSI

Рисунок 20 – Форма ввода утерянных вещей

## **Вывод по разделу**

Для обеспечения надежности приложения были внедрены механизмы логирования и мониторинга. Это позволяет оперативно выявлять и устранять возможные проблемы, а также анализировать поведение пользователей для дальнейшего улучшения приложения.

В итоге, все эти модули вместе обеспечивают безопасное и удобное функционирование приложения, соответствующее современным стандартам и требованиям.

Поиск утерянных вещей является актуальной проблемой, которая возникает при различных обстоятельствах. Эта проблема может возникнуть в результате потери ключей, документов, мобильных телефонов, кошельков или других ценных или важных вещей. В связи с этим существует необходимость разработки системы, которая поможет людям вернуть утерянные вещи.

Разработка системы для поиска утерянных вещей позволит создать удобный инструмент для поиска потерянных вещей, что приведет к уменьшению количества потерянных вещей и улучшению качества жизни людей. В ходе данной работы были проанализированы существующие системы и технологии, определены требования к разрабатываемой системе и ее функциональности. На основе этого анализа были разработаны и внедрены модули, обеспечивающие высокую безопасность, удобство использования и надежность системы.

Таким образом, итогом работы стало создание комплексного решения, которое решает проблему поиска утерянных вещей, улучшая повседневную жизнь пользователей.

## 4. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

### 4.1. Оценка готовности к коммерциализации программной разработки «Бюро находок»

Модель COCOMO II была применена для оценки затрат на разработку и внедрение программного обеспечения. Эта модель является алгоритмической и предназначена для оценки стоимости ПО, обладая рядом преимуществ и содержа проработанные алгоритмы настройки и оценки. COCOMO II использует упрощенную регрессионную формулу, основанную на данных из различных проектов, и поддерживает разнообразные модели жизненного цикла и языки программирования. Оценка трудозатрат и времени, необходимых для разработки, помогает планировать бюджет, определять оптимальные сроки и стоимость внедрения программного продукта.

В модели COCOMO II процесс оценки разделяется на два этапа: предварительная оценка на начальном этапе и более детальная оценка после разработки архитектуры. Для проведения экономического анализа выпускной квалификационной работы применяется предварительная оценка по методике COCOMO II.

Формула (1) для оценивания трудозатрат (PM) имеет вид:

$$PM = EAF \times A \times SIZE^E, \quad (1)$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j, \quad (2)$$

где  $B$  — корректирующий коэффициент. На этапе предварительной оценки его принимают равным 0,91;

— корректирующий коэффициент. На этапе предварительной оценки его принимают равным 2,94;

$SF_j$  (Scale Factors) — фактор масштаба;

$SIZE$  — объем программного продукта в тысячах строк исходного текста ( $KSLOC$  — Kilo of Source Line of Code);

$EAF$  (Effort Adjustment Factor) — произведение выбранных множителей

трудоемкости (см. форм. 5):

$$EAF = \prod_{i=1}^n EM_i, \quad (3)$$

где  $EM_i$  (Effort Multiplier) — множители трудоемкости. Для предварительной оценки количество факторов в модели равно 8 ( $n=8$ ).

Таблица 2 – Оценка готовности к коммерциализации

Наименование критерия	Оценка
Соответствие ожиданиям целевой группы	4
Качество интерфейса	5
Работоспособность и стабильность	4
Полнота документации	2
Комплексная безопасность	4
Ожидаемый эффект от внедрения	4
Известность бренда производителя	1
Качество сервиса и обслуживания	5
Коэффициент степени готовности к коммерциализации	0,725

Аргументация оценки готовности к коммерциализации (см. табл. 2):

- соответствие ожиданиям целевой группы — 4, поскольку в сфере информационной безопасности много направлений, которые невозможно осветить в рамках ознакомительной базы;
- качество интерфейса — 5, поскольку интерфакс не перегружен и содержит в себе только необходимые элементы управления;
- работоспособность и стабильность — 4, так как при большой нагрузки на хостинг есть низкая вероятность его отказа;
- полнота документации — 2, так как документация почти отсутствует (только readme файл и некоторые комментарии к функциям);
- комплексная безопасность — 4, персональные данные пользователей не хранятся в явном виде на сервере, но абсолютной безопасности не существует;
- ожидаемый эффект от внедрения — 4, потому что целевая аудитория была заинтересована проектом во время проведения опроса на ранней стадии разработки;

- известность бренда производителя — 1, поскольку это дебютный проект такого масштаба;
- качество сервиса и обслуживания — 5, поскольку обеспечивается ежедневная проверка электронной почты.

Таким образом, коэффициент степени готовности к коммерциализации составляет 0,725 (см форм. 4), что соответствует готовности к выходу на рынок.

$$EAF = \frac{\frac{4 + 5 + 4 + 2 + 4 + 4 + 1 + 5}{8} \times 2}{10} = 0,725, \quad (4)$$

#### **4.2. Оценка фактора масштаба**

Модель оценки фактора масштаба состоит из следующих факторов:

- PREC — прецедентность;
- FLEX — гибкость процесса разработки;
- RESL — архитектура и разрешение рисков;
- TEAM — сработанность команды;
- PMAT — зрелость процессов.

Таблица 3 – Оценка трудозатрат по разработке ПО согласно модели СОСМО II

<b>Оценка фактора масштаба (j)</b>	
PREC	1.24
FLEX	2.03
RESL	1.41
TEAM	1.10
PMAT	4.68
SF <sub>j</sub>	10.46
E	1.0146

Уровни значимости факторов масштаба ( $SF_j$ ) экспертным путем были оценены на следующие значения (см. табл. 3):

- PREC — средний (есть некоторый опыт в продукте и платформе);
- FLEX — очень высокий (незначительная жесткость процесса);
- RESL — очень низкий (риски известны на 90
- TEAM — очень высокий (высокая степень взаимодействия);
- PMAT — высокий.

По формуле 2 имеем:

$$E = 0,91 + 0,01 \times (1,23 + 2,03 + 1,41 + 1,10 + 4,68) = 1,0146. \quad (5)$$

### 4.3. Оценка фактора трудоемкости

Модель оценки фактора трудоёмкости ( $EM_j$ ) состоит из следующих факторов:

- PERS — квалификация персонала;
- PREX — опыт персонала;
- RCPX — сложность и надежность продукта;
- RUSe — разработка для повторного использования;
- PDIF — сложность платформы разработки;
- FCIL — оборудование (инструменты простейшие/интегрированные средства поддержки жизненного цикла);
- CSED — требуемое выполнение графика работ.

Значения множителей приведены в таблице 4. Основным отличием метода является наличие данных о размере кода (SIZE). Размер продукта может быть определен экспертами с использованием метода PERT или анализа продукта с помощью функциональных точек. Во втором случае размер может быть рассчитан с помощью таблицы 4.

Таблица 4 – Оценка фактора трудоемкости

<b>Оценка фактора трудоемкости (i)</b>	
PERS	0.83
PREX	1.00
RCPX	1.00
RUSe	1.10
PDIF	1.00
FCIL	1.00
CSED	1.00
EAF	0.913
SIZE	5.00
PM	13,74

$$EAF = 0,83 \times 1 \times 1 \times 1,1 \times 1 \times 1 = 0,913 \quad (6)$$

$$E = 2,94 \times 0,913 \times 5^{1,0146} = 13,74 \quad (7)$$

Таким образом, полученное значение из расчетов 6 и 7 говорит о том, что данный проект может быть выполнен за 13,74 месяца одним человеком.

#### **4.4. Расчет полной стоимости программного продукта (в части трудозатрат)**

При оценке общей стоимости создания программного продукта важно учитывать не только труд разработчиков, но и различные этапы, включая анализ требований, проектирование функций, архитектуры и интерфейсов, составление документации и плана интеграции, написание кода, тестирование и прочие задачи. В проекте по разработке программного обеспечения задействованы не только программисты, но и тестировщики, дизайнеры, специалисты по внедрению и другие профессионалы. Обычно на каждый час программирования приходится минимум один час работы других специалистов для подготовки и поддержки проекта.

Соотношение зарплат специалистов [??]:

- тестировщики получают 50% от зарплаты разработчиков;
- отдел внедрения получает так же, как и разработчики;
- проектировщики – 75% от вознаграждения разработчиков.

Согласно таблице 5 получается, что общая сумма затрат на разработку программного продукта, с полным циклом и количеством специалистов равным 6 ( $i=6$ ), составило 2 млн. 699 тыс. 949 рублей.

Таблица 5 – Расчет полной стоимости программного продукта (в части трудозатрат)

<b>Сотрудники (i)</b>	<b>Процент затрат времени на работу i-го специалиста, %</b>	<b>Трудозатраты, количество мес. РМ</b>	<b>Средний оклад, руб.</b>	<b>Приведенные затраты специалистов, руб.</b>
Проектирование (Проектировщик)	35	4,81	90000	432816,32
Разработка (Разработчик)	100	13,74	120000	1648824,08
Тестирование (Тестировщики)	50	6,87	60000	412206,02
Нагрузочное тестирование (Внедрение)	5	0,69	120000	82441,20
Внедрение (Внедрение)	5	0,69	120000	82441,20
Техническая документация (Производство)	5	0,69	60000	41220,60
<b>Итого полная стоимость ПО в части трудозатрат</b>				<b>2699949,42</b>

#### 4.5. Оценка стоимости разработки и внедрения программного продукта

Согласно принятой методике, общая стоимость разработки программного обеспечения включает в себя трудозатраты, к которым добавляется 13% налога на доходы. Кроме того, следует учесть дополнительные издержки, некоторые из которых являются обязательными, а другие — выборочными.

В деталях:

- Возможно начисление бонуса до 15% от зарплаты за своевременное выполнение задач и удовлетворение требований клиента;
- Необходимо вычтать 7,6% из зарплаты на социальное страхование, включая медицинское обеспечение, пенсионные отчисления и социальное страхование;
- Может быть предусмотрено возмещение расходов на питание в размере 2% от зарплаты.

Кроме того, специалистам необходимо обеспечить рабочее место с современным офисным оборудованием и надежными средствами коммуникации. Работодатель должен заниматься подбором соответствующего персонала, вести финансовый учет через бухгалтерию, обеспечивать порядок

в офисе и назначать ответственных за это. Социальный пакет играет ключевую роль, особенно для высококвалифицированных сотрудников.

Таблица 6 – Оценка стоимости разработки и внедрения программного продукта

Статьи калькуляции	Процент расходов	Сумма
Полная стоимость ПО в части трудозатрат	-	2699949,425
Расходы на хозяйственные и административные нужды	20%	539989,8849
Премия к заработной плате	15%	404992,4137
Компенсация питания сотрудников	2%	53998,98849
Взносы на социальное страхование	7.60%	281118,7341
<b>Полная стоимость внедрения ПО, в т.ч. НДС 0%</b>		<b>3980049,446</b>

Эти факторы влияют на включение в методику расходов на коммерческую и управлеченческую деятельность, что составляет около 20% от общих трудозатрат на разработку программного обеспечения. При определении итоговой стоимости внедрения также следует учесть НДС в размере 20% от всей суммы. Однако, если программное обеспечение включено в реестр отечественных программ, НДС не взимается. Полная стоимость внедрения данного примера подробно отражена в таблице 6.

#### 4.6. Расчет общего количества капитальных затрат ТСО (CAPEX)

Согласно методике Gartner по расчету общей стоимости владения (TCO), при определении капитальных затрат учитываются не только затраты на внедрение программного продукта, но и технические, а также консультационные расходы, которые напрямую связаны с установкой и адаптацией системы под нужды компании. Важно также принимать во внимание затраты на обучение сотрудников и приобретение специализированного оборудования и коммуникационных средств для

надлежащей работы информационной системы предприятия.

Многие организации не включают эти расходы в проектные затраты, а следовательно, и в общую стоимость продукта, что требует их дополнительного учета при подсчете окончательной суммы капитальных затрат.

Расчет затрат на обучение и переподготовку персонала может быть выполнен следующим образом.

Этот метод считается удобным и доступным для использования, однако его основной недостаток заключается в приблизительности расчетов. Он базируется на анализе множества проектов по разработке и внедрению программного обеспечения в различных, включая иностранные и отечественные, компаниях. Исследования показывают, что затраты на обучение обычно не превышают 20% от общей стоимости внедрения программного обеспечения.

Таким образом, зная первоначальные капитальные затраты (полная стоимость внедрения программного обеспечения) и максимальный установленный уровень затрат на обучение (20%), можно определить максимально возможную сумму расходов, которую компания может включить в общую стоимость владения как затраты на обучение и подготовку персонала.

$$K_{\text{п.к.}} = 3980049,446 \times 20\% = 796009,889 \quad (8)$$

Таким образом, согласно пропорциональному методу, сумма затрат, выделяемых на обучение персонала, составляет 796 тыс. 9 рублей.

#### **4.7. Калькуляция затрат на закупку специального оборудования или каналов связи**

При аудите оборудования было выявлено, что для полноценного функционирования ПО необходимо следующее оборудование (см. табл. 7).

Таблица 7 – Калькуляция затрат на закупку специального оборудования или каналов связи

<b>Статьи калькуляции</b>	<b>Стоимость оборудования и каналов связи, руб.</b>	<b>Количество, шт</b>	<b>Сумма, руб.</b>
Серверное и сетевое оборудование, СХД	2000	1	2000
Оборудование для информационной безопасности	0	0	0
Платформа виртуализации	0	0	0
Система резервного копирования	1000	1	1000
Интернет с подключением от нескольких провайдеров	2000	1	2000
<b>ИТОГО на закупку специального оборудования или каналов связи, (Кр), руб</b>			<b>5000</b>

#### 4.8. Расчет общего количества операционных затрат ТСО (OPEX)

Операционные затраты (форм. 9) — все последующие расходы компании.

$$OPEX = K_{af} \times T \quad (9)$$

где  $K_{af}$  — последующие расходы;;

$T$  — количество лет.

В перечень затрат входят платежи за продление лицензии, обновление программ, внешнее техническое сопровождение, сервисную поддержку, обеспечение безопасности, резервное копирование данных, устранение неполадок, рефакторинг, программирование, поддержание актуальности версий и другие связанные аспекты.

Операционные издержки охватывают расходы на обновление ПО без изменения его функций, управление системами и их поддержку.

Часто компании не принимают во внимание операционные издержки при планировании проектов, что приводит к их отсутствию в общей стоимости продукта. Затраты на обновления, модернизацию и техническую поддержку программного обеспечения оплачиваются заказчиками отдельно, что гарантирует его стабильную работу. Практика показывает, что заказчики

чаще всего предпочитают оплачивать техническую поддержку как часть операционных расходов, вместо того чтобы самостоятельно заниматься поддержкой кода.

Для анализа операционных расходов можно провести подробное разделение на разные категории в зависимости от требуемого уровня функциональности и необходимого сопровождения.

Примерный расчет операционных расходов можно выполнить, определив их долю от общей стоимости внедрения программного продукта. Обычно предполагается, что на поддержку и обновление ПО приходится около 30% от общей суммы расходов в год (см. табл. 8).

Таблица 8 – Расчет совокупной стоимости владения (TCO)

Статьи затрат	Сумма, руб.
Капитальные затраты (CAPEX) - первоначальные затраты на приобретение	4781059,335
Затраты на внедрение	3980049,446
Затраты на обучение персонала	796009,8892
Закупка специального оборудования или каналов связи	5000
Операционные затраты (OPEX) - эксплуатационные затраты	3582044,501
Затраты на обновление и модернизацию (например, продление лицензии, программное обновление)	398004,9446
Расходы на управление системой (сопровождение внешнее, сервис поддержки, безопасность, резервное копирование, исправление багов, рефакторинг, исправление кода, поддержка версий)	796009,8892
<b>Совокупная стоимость владения (TCO)</b>	<b>8363103,837</b>

При запланированном полезном сроке использования ПО в течении 3-х лет, общий объем операционных затрат равен 8363103,837 руб.

#### 4.9. Расчет совокупной стоимости владения (TCO)

Согласно модели расчета TCO Gartner Group, затраты компании рассчитываются по формуле 10:

$$TCO = CAPEX \times OPEX \quad (10)$$

#### **4.10. Оценка конкурентоспособности программного продукта**

Конкурентоспособность программного продукта определяется его соответствием текущим требованиям рынка и его актуальностью на момент запуска. Для оценки этой конкурентоспособности проводится сравнительный анализ с другими схожими продуктами.

Этот анализ включает оценку того, насколько хорошо продукт соответствует требованиям рынка не только с технической, математической, информационной и организационной точек зрения, но и с коммерческой стороны, включая ценообразование, уровень обслуживания, сеть продаж, рекламу и прочее.

Конкурентоспособность измеряется путем сравнения с аналогичными товарами и рассматривается через призму потребительских интересов, которые можно разделить на объективные и субъективные категории.

К “жестким”, критериям относятся:

- Технические характеристики, включая классификацию, производительность и конструктивные особенности продукта;
- Эргономические характеристики, связанные с удобством использования и функциональностью;
- Качество взаимодействия пользователя с продуктом, скорость утомления, удобство работы и другие аспекты;
- Соответствие международным и национальным стандартам, нормативным актам и законодательству.

В оценке программного обеспечения по “жестким” критериям ключевыми являются функциональные возможности, как определено в стандарте ГОСТ Р ИСО/МЭК 9126-93, включающем пригодность, корректность, совместимость и защищенность.

“Мягкие” характеристики включают эстетические и визуальные аспекты, такие как дизайн, цвет и упаковка, которые могут влиять на восприятие потребителями. В случае программного обеспечения это может включать настройку интерфейса, интеграцию с удаленными системами и обработку документов.

Также в оценке конкурентоспособности учитываются экономические характеристики, включая начальные и эксплуатационные затраты, которые

формируют общую стоимость использования продукта. Эти затраты включают расходы на обновление, апгрейд, управление системой, внешнее обслуживание, техническую поддержку, безопасность, резервное копирование, исправление ошибок и поддержку версий.

Ещё один ключевой элемент, влияющий на конкурентоспособность программного обеспечения, — это организационно-экономические аспекты сделки, включая систему скидок, условия оплаты и доставки, а также гарантийные обязательства. Вначале необходимо определить важность каждого фактора в рамках конкурентоспособности ПО, распределяя веса по аналитическим блокам (согласно таблице 9), при этом суммарный вес факторов в каждом блоке должен составлять 100 процентов.

Таблица 9 – Результаты бальной оценки ПО по субъективным пользовательским предпочтениям

Субъективные пользовательские характеристики	Исследуемые Программные продукты			Вес критерия
	finds.mirea.ninja	buro-nahodok.ru	buro.nahodok.ru	
возможность настройки ПП	5	4	1	50
возможность работы с территориально-распределенными офисами	3	5	1	25
возможность обработки документов	5	3	3	25
<b>Сумма баллов</b>	<b>200</b>	<b>200</b>	<b>100</b>	
<b>Индекс конкурентоспособности по субъективным пользовательским предпочтениям</b>	<b>1,00</b>	<b>1,00</b>	<b>0,50</b>	

На следующем этапе требуется собрать данные и провести анализ программного обеспечения и ПО двух основных конкурентов согласно установленным критериям.

В процессе сравнения ПО конкурентов важно оценить соответствие по ряду параметров: сфера применения, объем программных ресурсов, сложность разработки, применение стандартных модулей и типовых решений, затраты труда на разработку, качество результатов, которые предоставляет продукт,

время выполнения задач и ценовой уровень. Если прямых аналогов для сравнения нет, следует провести анализ технико-экономических характеристик различных вариантов программного обеспечения или системных компонентов.

Далее осуществляется расчет индексов конкурентоспособности для каждого из определенных блоков.

Используя оценку функциональных возможностей был посчитан индекс конкурентоспособности по функциональным возможностям (см. табл. 10).

Таблица 10 – Результаты бальной оценки ПО по функциональным возможностям

Функциональные возможности	Исследуемые Программные продукты			Вес критерия
	finds. mirea.ninja	buro- nahodok.ru	buro.nahodok.ru	
<b>Пригодность</b>				
соответствие назначения целям применения ПС	5	2	4	9
соответствие требований к функциям назначению ПС	5	4	4	9
соответствие исходной информации требованиям к функциям ПС	3	4	2	9
соответствие состава и содержания выходной информации для потребителей ПС	5	3	3	9
соответствие структурных характеристик комплекса программ ПС	4	3	2	5
<b>Корректность (правильность)</b>				
соответствие требований к функциям ПС информационной системе	5	3	4	2
соответствие требований к функциональным компонентам ПС	5	5	4	2
соответствие текстов программ требованиям к функциональным компонентам	5	4	2	2

соответствие объектного кода исходному тексту программ	5	4	5	2
степень покрытия тестами возможных маршрутов исполнения программ	1	2	2	2
<b>Способность к взаимодействию (совместимости)</b>				
с операционной системой	1	1	1	5
с аппаратной средой	1	1	1	5
с внешней средой информационной системы и пользователями	5	3	4	5
между программными компонентами	3	1	1	5
между компонентами распределенных информационных систем	3	1	1	5
<b>Защищенность</b>				
аутентификация элементов систем обработки данных	5	3	5	8
управление доступом	3	3	3	8
протоколирование обращений	2	3	5	8
криптографическая защита	4	1	5	0
превентивное реагирование	2	3	3	0
<b>Сумма баллов</b>	<b>369</b>	<b>275</b>	<b>305</b>	

Аналогично рассчитываются частные индексы по Блокам «Субъективные пользовательские предпочтения» и «Организационные критерии».

Используя оценку функциональных возможностей был посчитан индекс конкурентоспособности по экономическим возможностям (см. табл. 11).

Таблица 11 – Результаты оценки ПО экономическим критериям (по цене потребления)

Элементы цены потребления	Исследуемые Программные продукты			Вес критерия
	finds.mirea.ninja	buro-nahodok.ru	buro.nahodok.ru	
<i>Первоначальные затраты на приобретение</i>				
Затраты на внедрение	3.980.049	15.000.000	23.000.000	10
Затраты на обучение персонала	796.010	100.000	100.000	15
Закупка специального оборудования или каналов связи	5.000	100.000	50.000	10
<i>Эксплуатационные затраты</i>				
Затраты на обновление и модернизацию	398.005	0	0	15
Расходы на управление системой	796.010	1.000.000	5.000	50
<b>Цена потребления</b>	<b>5.975.074</b>	<b>16.200.000</b>	<b>23.155.000</b>	
<b>Индекс конкурентоспособности по экономическим критериям</b>	<b>1,00</b>	<b>0,37</b>	<b>0,26</b>	

## Вывод

Основная идея данного подхода заключается в анализе и сравнении основных характеристик товара компании с аналогичными товарами конкурентов, с последующим отображением результатов анализа в виде графика в форме многоугольника. Визуализация конкурентоспособности, созданная на основе данных из таблицы 12, представлена на рисунке 21. Можно заметить, что разработанный программный продукт превосходит аналоги и готов к выходу на рынок.

Таблица 12 – Расчет итогового рейтинга конкурентоспособности

Частные индексы конкурентоспособности	Исследуемые Программные продукты			Вес группы критерiev
	finds.mirea.ninja	buro-nahodok.ru	buro.nahodok.ru	
Индекс конкурентоспособности по функциональным возможностям	1,00	0,75	0,83	40%
Индекс конкурентоспособности по субъективным пользовательским предпочтениям	1,00	1,00	0,50	15%
Индекс конкурентоспособности по организационным критериям	1,00	1,00	1,00	15%
Индекс конкурентоспособности по цене потребления	1,00	0,37	0,26	30%
<b>ИТОГОВОЕ ЗНАЧЕНИЕ ИНДЕКСА</b>	<b>1,00</b>	<b>0,71</b>	<b>0,63</b>	<b>100%</b>

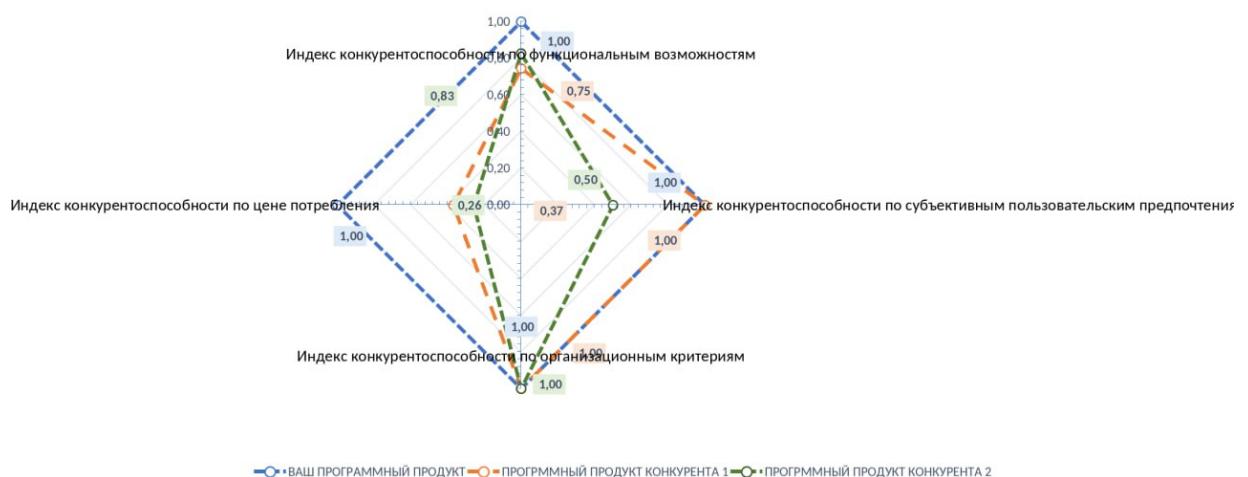


Рисунок 21 – Рейтинг конкурентоспособности

## **ЗАКЛЮЧЕНИЕ**

Проведенное исследование подтвердило актуальность и важность разработки системы бюро находок. Статистические данные о потерянных и найденных вещах продемонстрировали значительный объем утраченных предметов, что подтверждает необходимость создания эффективного инструмента для их поиска и возврата. Опрос пользователей выявил ключевые проблемы и потребности, которые могут быть удовлетворены с помощью данной системы.

Разработка системы бюро находок не только способствует возвращению потерянных вещей их владельцам, но и повышает общую безопасность и доверие среди пользователей. Оценка нагрузки на систему и ее производительность, основанная на статистических данных, позволит обеспечить ее надежность в долгосрочной перспективе.

В ходе работы было разработано приложение finds.mirea.ninja со следующими модулями:

- модуль регистрации и авторизации пользователей,
- модуль бесконечных лент объявлений потерянных, найденных вещей,
- модуль добавления и поиска утерянных вещей,
- модуль генерации описания объявлений.

Внедрение данной системы является важным шагом на пути к улучшению общественных услуг и удовлетворению потребностей пользователь.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. МОСКВА 24 Что теряют москвичи // www.m24.ru: Новости Москвы, репортажи и интервью об основных событиях города URL: <https://www.m24.ru/news/gorod/28112019/98853> (дата обращения: 01.09.2023).
2. Усинск Онлайн Какие вещи чаще всего теряют россияне // usinsk.online URL: [https://usinsk.online/news/kakie-veshhi-chashhe-vsego-teriyayut-rossiyane/#:~:text=%20%20%3A%20,1%20\)%2C%20%20.](https://usinsk.online/news/kakie-veshhi-chashhe-vsego-teriyayut-rossiyane/#:~:text=%20%20%3A%20,1%20)%2C%20%20.) (дата обращения: 01.09.2023).
3. Bataineh, Emad, Bilal Bataineh, and Shama Al Kindi. "Design, development and usability evaluation of an online web-based lost and found system." International Journal of Digital Information and Wireless Communications 5.2 (2015): 75-82.
4. Tan, Siok Yee, and Cia Rui Chong. "AN EFFECTIVE LOST AND FOUND SYSTEM IN UNIVERSITY CAMPUS." Management 8.32: 99-112.
5. Бюро находок // столнаходок.рф: информационно-поисковый портал РФ URL: <http://nahodok.ru/> (дата обращения: 01.09.2023).
6. Потерял Нашел // ponal.ru: бюро находок Пона.рф. Удобный поиск по объявлениям, большая база потерянных вещей и животных URL: <https://ponal.ru/sochi> (дата обращения: 01.09.2023).
7. Investopedia // investopedia.com: Radio Frequency Identification (RFID): What It Is, How It Works URL: [https://www.investopedia.com/terms/r/radio-frequency-identification-rfid.asp#:~:text=Radio%20Frequency%20Identification%20\(RFID\)%20is,checked%20out%20of%20a%20library.](https://www.investopedia.com/terms/r/radio-frequency-identification-rfid.asp#:~:text=Radio%20Frequency%20Identification%20(RFID)%20is,checked%20out%20of%20a%20library.) (дата обращения: 01.09.2023).
8. Investopedia // investopedia.com: Radio Frequency Identification (RFID): What It Is, How It Works URL: [https://www.investopedia.com/terms/r/radio-frequency-identification-rfid.asp#:~:text=Radio%20Frequency%20Identification%20\(RFID\)%20is,checked%20out%20of%20a%20library.](https://www.investopedia.com/terms/r/radio-frequency-identification-rfid.asp#:~:text=Radio%20Frequency%20Identification%20(RFID)%20is,checked%20out%20of%20a%20library.) (дата обращения: 01.09.2023).

9. AirTag // apple.com: магазин Apple URL: <https://www.apple.com/airtag/> (дата обращения: 01.09.2023).
10. Lost Property Office // parliament.uk: веб приложение URL: <https://www.parliament.uk/visiting/access/facilities/lost-property/> (дата обращения: 01.09.2023).
11. Next.js – The React Framework for Production. URL: <https://nextjs.org/> (дата обращения: 16.05.2024).
12. React – JavaScript библиотека для построения пользовательских интерфейсов. URL: <https://reactjs.org/> (дата обращения: 16.05.2024).
13. tRPC – End-to-end типобезопасные программные интерфейсы.. URL: <https://trpc.io/> (дата обращения: 16.05.2024).
14. Prisma – открытая Node.js и TypeScript ORM с читаемыми моделями и автоматическими миграциями. URL: <https://www.prisma.io/> (дата обращения: 16.05.2024).
15. NextAuth – JavaScript библиотека для аутентификации. URL: <https://next-auth.js.org/> (дата обращения: 16.05.2024).
16. React Query – асинхронный менеджер состояний. URL: <https://tanstack.com/query/v3> (дата обращения: 16.05.2024).
17. React Query – асинхронный менеджмент состояний для сетевых запросов. URL: <https://tanstack.com/query/v3> (дата обращения: 16.05.2024).
18. Zustand – менеджер состояний клиентского приложения. URL: <https://tanstack.com/query/v3> (дата обращения: 16.05.2024).
19. Tailwind CSS – CSS фреймворк для быстрого построения современных вебсайтов. URL: <https://tailwindcss.com/> (дата обращения: 16.05.2024).
20. AWS SDK – JavaScript библиотека для взаимодействия с интерфейсами Amazon Web Services. URL: <https://aws.amazon.com/sdk-for-javascript/> (дата обращения: 16.05.2024).

21. Headless UI – полностью нестилизованная библиотека пользовательских интерфейсов для интеграции с Tailwind CSS. URL: <https://headlessui.com/> (дата обращения: 16.05.2024).
22. OAuth 2.0 - промышленный стандарт авторизации. URL: <https://oauth.net/2/> (дата обращения: 16.05.2024).