

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ	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
2 СПЕЦИАЛЬНЫЙ РАЗДЕЛ	15
2.1 Требования к разрабатываемой системе	15
2.1.1 Функциональные требования	16
2.1.2 Нефункциональные требования	16
2.2 Проектирование модулей автоматизации процессов	18
2.2.1 Модуль регистрации и авторизации пользователей	18
2.2.2 Модуль бесконечных лент объявлений потерянных, найденных вещей	19
2.2.3 Модуль добавления и поиска вещей	19
2.2.4 Модуль генерации описания объявлений	19
3 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ	22
3.1 Выбор технологий для реализации системы	22
3.2 Реализация модулей автоматизации процессов	24
3.2.1 Модуль регистрации и авторизации пользователей	24
3.2.2 Модуль бесконечных лент объявлений потерянных, найденных вещей	25
3.2.3 Модуль добавления и поиска утерянных вещей	27
3.2.4 Модуль генерации описания объявлений	30
4 ЭКОНОМИЧЕСКИЙ РАЗДЕЛ	33
4.1 Планирование разработки системы	33
4.1.1 Определение трудоемкости и продолжительности работ по созданию УСПД	33
4.1.2 Построение ленточного графика проведения исследования	36

4.2	Расчет сметы затрат на разработку представленной работы	38
4.2.1	Основная заработка плата исполнителей	38
4.2.2	Дополнительная заработка плата исполнителей	39
4.2.3	Расчет отчислений на социальное страхование	39
4.2.4	Накладные расходы	39
4.2.5	Расчет стоимости машинного времени	40
	ЗАКЛЮЧЕНИЕ	43
	ПРИЛОЖЕНИЕ А	46

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

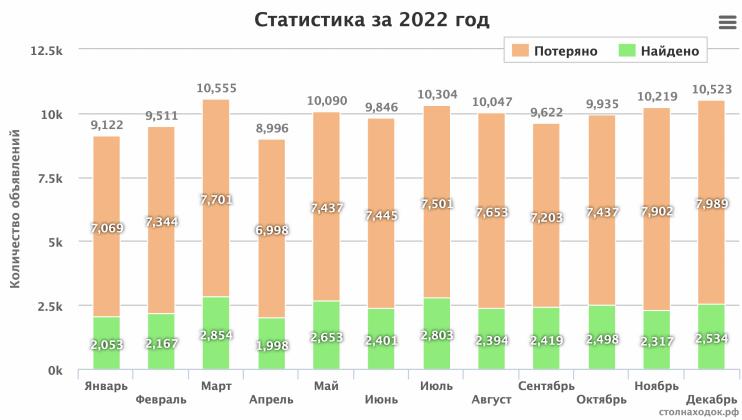


Рисунок 2 – Востребованность системы столнаходок.рф в 2022 году

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:

- 1) 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.
- 2) 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.

- 3) 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.
- 4) 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.

The screenshot shows the homepage of the 'Buro Naходок' website. At the top, there is a banner with the text 'Бюро находок - единый центр находок!' (Buro Naходок - unified center for finds!). Below the banner, there is a message in Russian welcoming visitors to the site. Further down, there are three buttons: 'Я НАШЕЛ :)' (I found), 'Адреса и телефоны бюро находок' (Addresses and phone numbers of the Bureau of Finds), and 'Я ПОТЕРЯЛ :((I lost). A statistics box displays 'Всего объявлений: 300924 (добавлено сегодня: 35, где 0 - найдено, 35 - потеряно)' (Total announcements: 300924 (added today: 35, where 0 - found, 35 - lost)) and a link to 'Статистика бюро находок'. The main content area is divided into two sections: 'Последние 30 находок' (Last 30 finds) and 'Последние 30 пропаж' (Last 30 losses). Each section contains a grid of announcement cards. For example, under 'Последние 30 находок', one card from 29.10.2023 in Krasnodar says 'НАШЕЛ КОШЕЛЕК И КАРТОЧКИ НАШЕЛ ОКОЛО МАГНИТА'. Under 'Последние 30 пропаж', one card from 30.10.2023 in Moscow says 'Утерян паспорт, приписанное удостоверение, СНИЛС'.

Рисунок 3 – Скриншот системы «столнаходок.рф»

«Стол находок» [5] — это мобильное приложение, разработанное для операционных систем iOS и Android. Оно предлагает функцию отслеживания утерянных предметов через GPS-модуль смартфона, представлено на рис. 5, 6. Пользователи могут отмечать свои вещи на карте и получать уведомления,

БЮРО НАХОДОК
Информационно Поисковый Портал РФ

Выберите регион и введите ключевое слово для поиска:
 Москва Искать Подать объявление
 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](#).

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

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

Подать объявление

Новое объявление

Регион *

Тип Объявления * найдено потеряно

Группа *

Заголовок объявления *

Текст объявления *

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

Контактное лицо

Телефон

Адрес эл.почты (E-mail) *

Фотография Файл не выбран

Заданный 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»

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

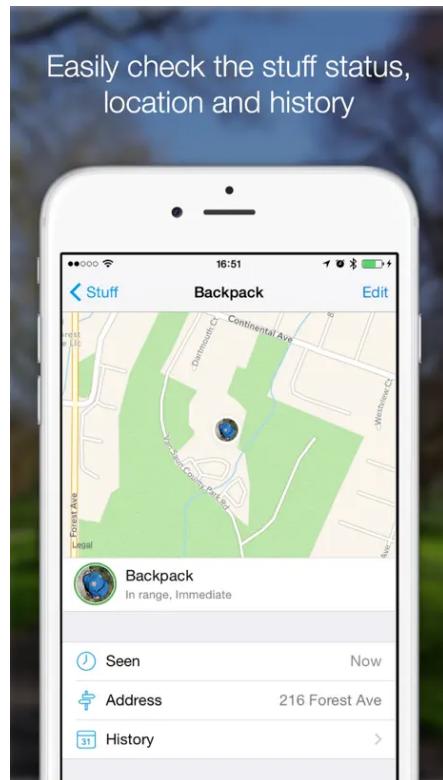


Рисунок 5 – Скриншот системы «Find My Stuff»

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:

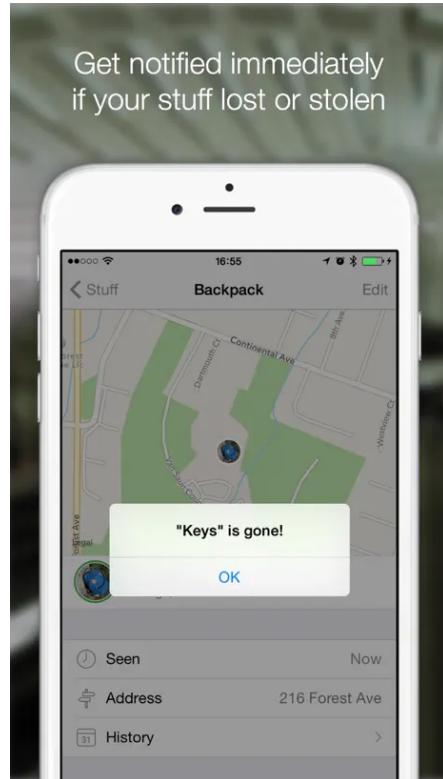


Рисунок 6 – Скриншот системы «Find My Stuff»

The screenshot shows the official website of the UK Parliament. The top navigation bar includes links for Business, MPs, Lords & offices, About, Get involved, Visit, More, and a search icon. The main content area is titled 'UK Parliament' and features a breadcrumb navigation: UK Parliament > Visit > Plan your visit > Shops, cafe and other visitor facilities > Lost Property Office. The main section is titled 'Lost Property Office' and contains text about handing in lost property at the Hallkeeper's Lodge. It includes a large photograph of a grand hall with ornate decorations and a statue. Below the photo is a paragraph of text providing contact information for lost property inquiries.

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.

Рисунок 7 – Скриншот системы «Lost Property Office»

Таблица 1 – Сравнительная таблица аналогов

Сервис / Приложение	Интерфейс и удобство использования	Оповещения	Точность определения местоположения	Регистрация 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	Ограниченный доступ

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 уянных 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

- 1) Приложение должно предоставлять возможность регистрации и авторизации пользователей.
- 2) Приложение должно предоставлять возможность поиска утерянных вещей по различным критериям (например, по типу вещи, по месту утери и т.д.).
- 3) Пользователи должны иметь возможность добавлять информацию о утерянных вещах, включая описание, фотографии и место утери.
- 4) Приложение должно предоставлять функционал для связи между пользователем, который нашел вещь, и пользователем, который ее потерял.
- 5) 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.

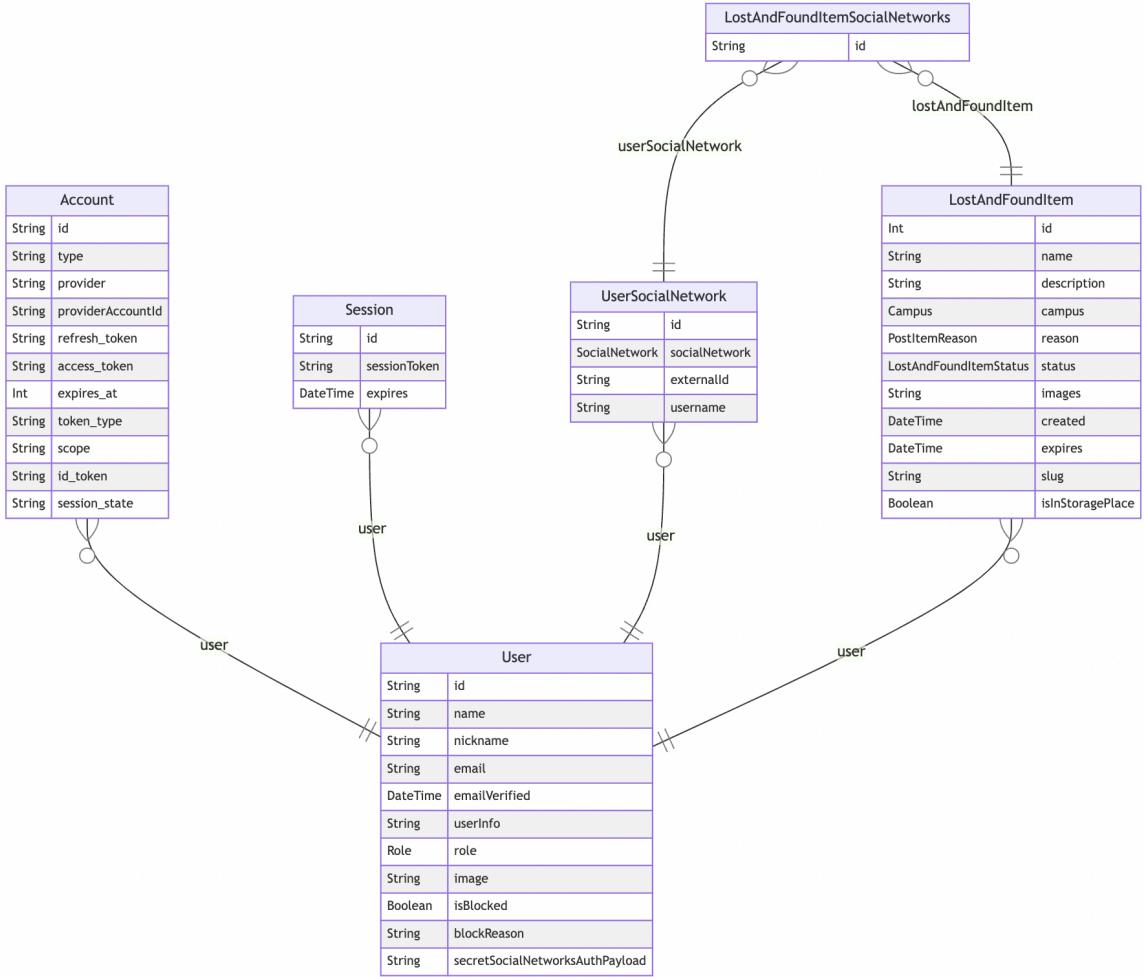


Рисунок 8 – ER-диаграмма системы

2.1.2 Нефункциональные требования

- 1) Приложение должно обеспечивать быстрый поиск и отображение результатов, а также быстрое добавление информации о утерянных вещах.
- 2) Все данные пользователей должны быть защищены.
- 3) Приложение должно быть доступно для использования 24/7.
- 4) Интерфейс приложения должен быть интуитивно понятным и удобным для пользователей разного уровня компьютерной грамотности.
- 5) Приложение должно быть совместимо с основными операционными системами (iOS, Android) и браузерами (Chrome, Firefox, Safari, Edge).
- 6) Приложение должно быть способно обслуживать большое количество пользователей одновременно без снижения производительности.
- 7) 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 процессов включает в себя разработку структуры и функционала модулей, которые будут автоматизировать ключевые процессы приложения для поиска и возврата утерянных вещей. В данном случае, ключевыми процессами являются: регистрация и авторизация пользователей, добавление и поиск утерянных вещей, связь между пользователями и система уведомлений.

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).

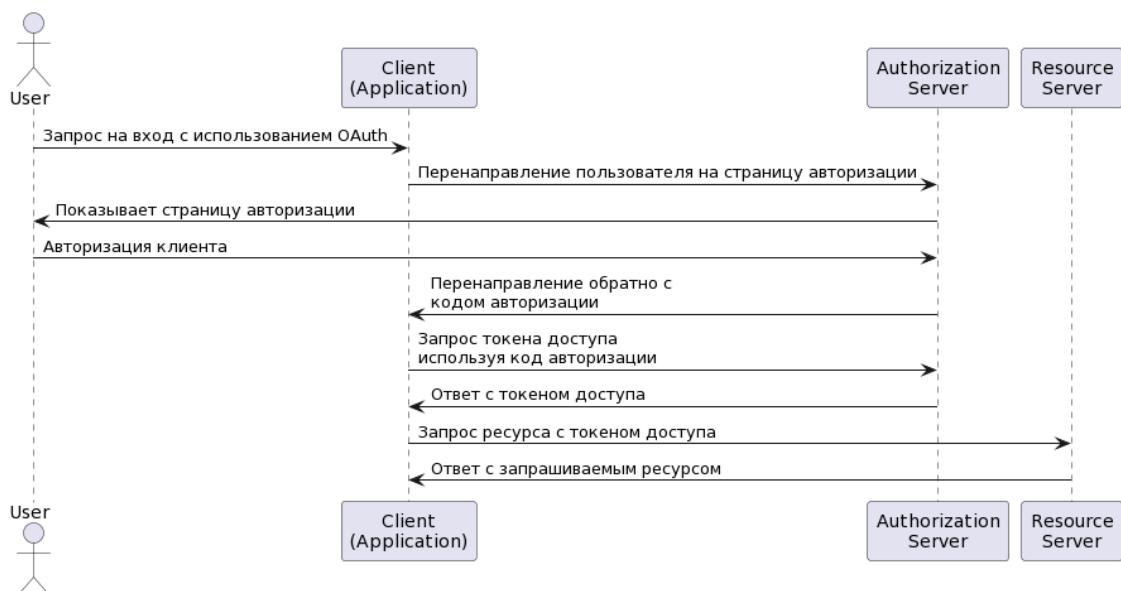


Рисунок 9 – Диаграмма последовательностей авторизации

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.

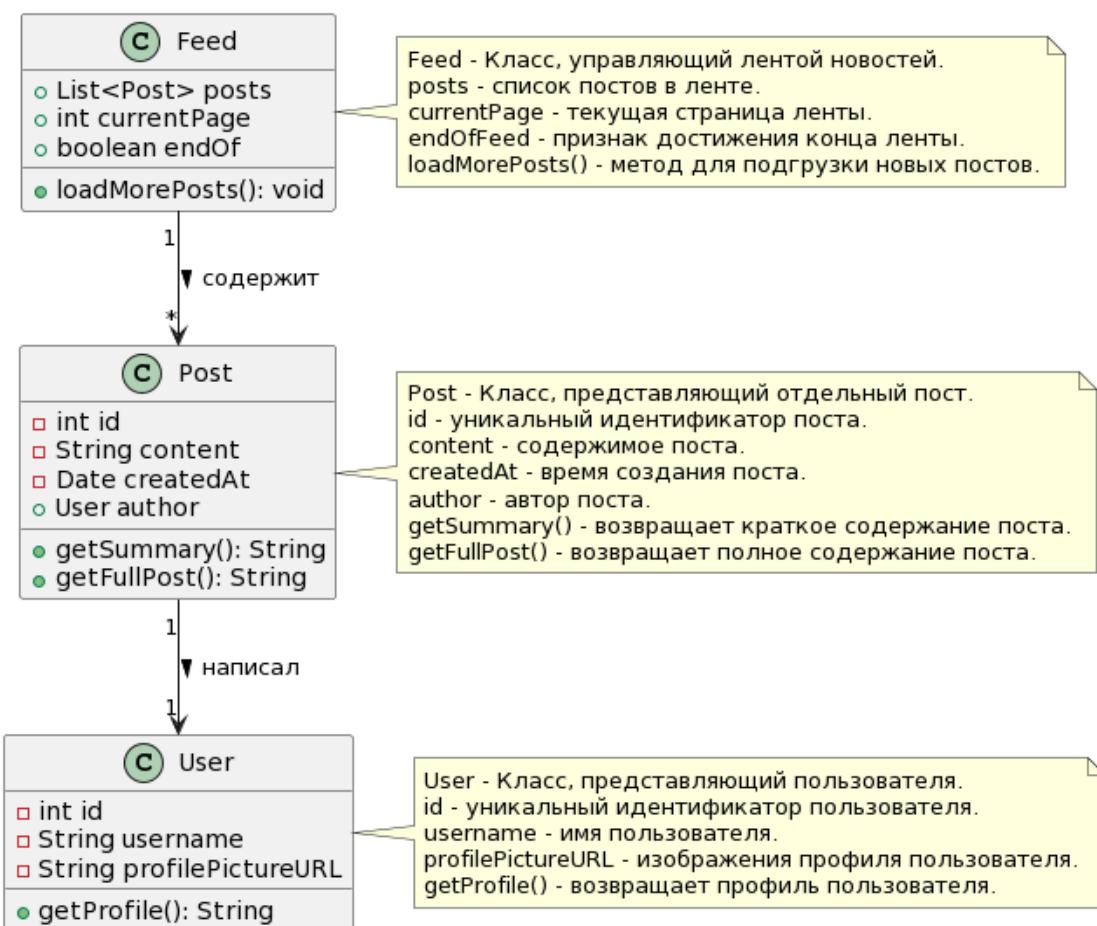


Рисунок 10 – Диаграмма классов бесконечной ленты

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.

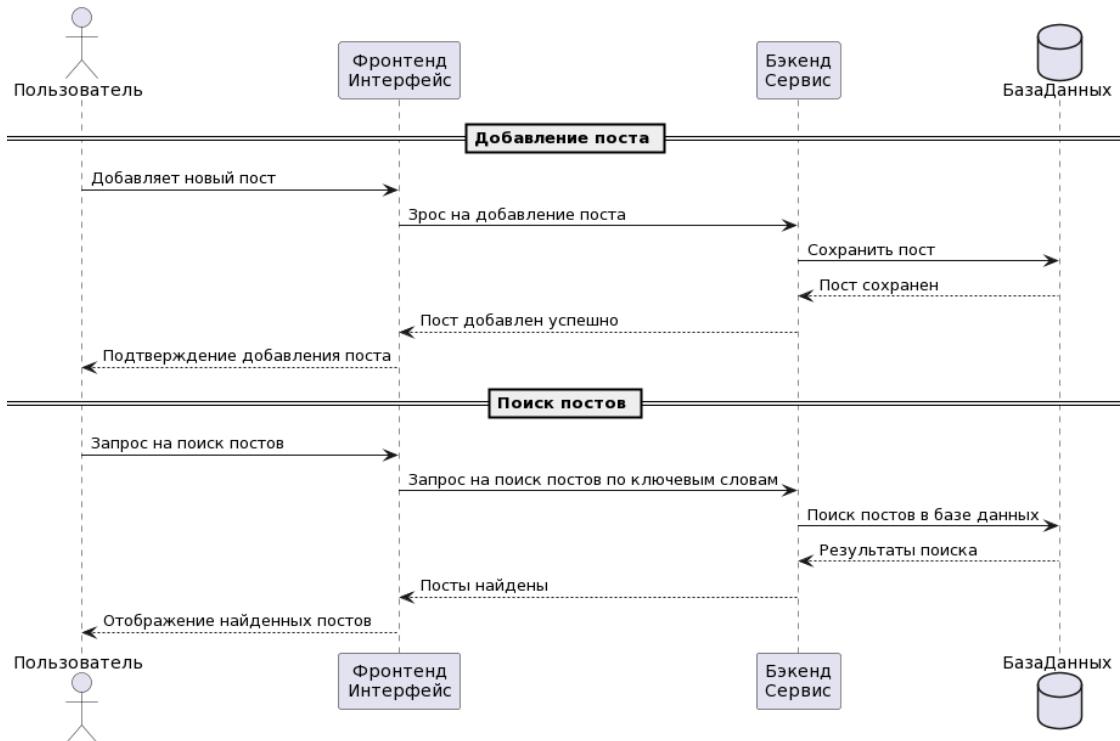


Рисунок 11 – Диаграмма последовательностей добавления и поиска вещей

2.2.4 Модуль генерации описания объявлений

Модуль генерации описания объявлений 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.

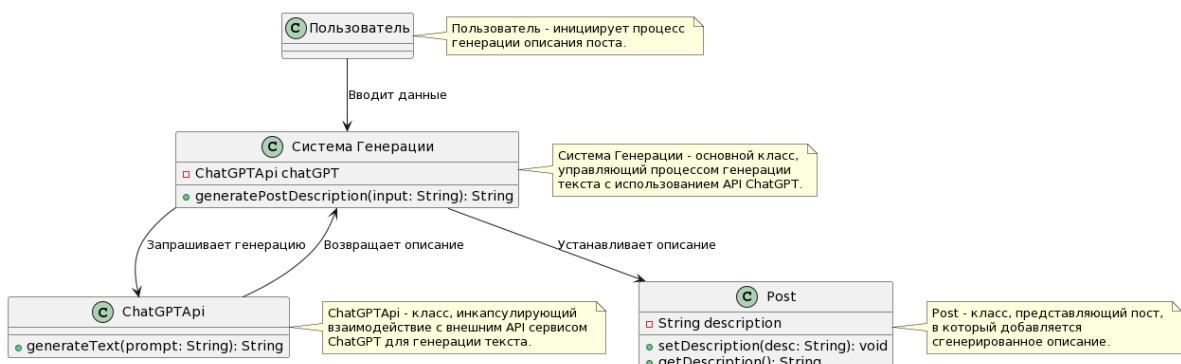


Рисунок 12 – Диаграмма классов генерации описания вещей

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 и эффективности приложения.

Каждый из этих модулей важен для обеспечения эффективности и 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

Ниже представлены некоторые из выбранных технологий:

Листинг 1 – Выбранные технологии

```
1  "dependencies": {  
2      "@headlessui/react": "^1.7.14",  
3      "@heroicons/react": "^2.0.17",  
4      "@hookform/resolvers": "^3.1.0",  
5      "@mdx-js/loader": "^2.3.0",  
6      "@mdx-js/react": "^2.3.0",  
7      "@next-auth/prisma-adapter": "^1.0.5",  
8      "@next/bundle-analyzer": "^13.2.4",  
9      "@next/mdx": "^13.4.3",  
10     "@prisma/client": "^4.15.0",  
11     "@react-three/drei": "^9.77.0",  
12     "@react-three/fiber": "^8.13.3",  
13     "@t3-oss/env-nextjs": "^0.2.2",  
14     "@tanstack/react-query": "^4.28.0",  
15     "@tanstack/react-table": "^8.9.3",  
16     "@trpc/client": "^10.18.0",  
17     "@trpc/next": "^10.18.0",  
18     "@trpc/react-query": "^10.18.0",  
19     "@trpc/server": "^10.18.0",  
20     "aws-sdk": "^2.1408.0",  
21     "axios": "^1.4.0",  
22     "classnames": "^2.3.2",  
23     "date-fns": "^2.29.3",  
24     "feed": "^4.2.2",  
25     "flowbite": "^1.6.5",  
26     "flowbite-react": "^0.4.4",  
27     "immer": "^10.0.2",  
28     "next": "^13.4.1",  
29     "next-auth": "4.20.1",  
30     "next-pwa": "^5.6.0",  
31     "next-s3-upload": "^0.3.0",  
32     "next-seo": "^6.0.0",  
33     "nextjs-progressbar": "^0.0.16",  
34     "nodemailer": "^6.9.1",  
35     "openai": "^3.2.1",  
36     "react": "^18.2.0",  
37     "react-copy-to-clipboard": "^5.1.0",  
38     "react-dom": "^18.2.0",  
39     "react-dropzone": "^14.2.3",
```

```

40   "react-highlight-words": "^0.20.0",
41   "react-hook-form": "^7.44.2",
42   "react-hot-toast": "^2.4.0",
43   "react-hotkeys-hook": "^4.4.1",
44   "react-infinite-scroll-component": "^6.1.0",
45   "react-medium-image-zoom": "^5.1.6",
46   "redis": "^4.6.7",
47   "sharp": "^0.32.0",
48   "spinners-react": "^1.0.7",
49   "superjson": "1.12.2",
50   "three": "^0.153.0",
51   "transliteration": "^2.3.5",
52   "trpc-panel": "^1.3.4",
53   "unique-names-generator": "^4.7.1",
54   "zod": "^3.21.4",
55   "zustand": "^4.3.8"
56 },
57 "devDependencies": {
58   "@tailwindcss/forms": "^0.5.3",
59   "@tailwindcss/typography": "^0.5.9",
60   "@types/eslint": "^8.21.3",
61   "@types/node": "^18.16.9",
62   "@types/prettier": "^2.7.2",
63   "@types/react": "^18.0.28",
64   "@types/react-copy-to-clipboard": "^5.0.4",
65   "@types/react-dom": "^18.0.11",
66   "@types/react-highlight-words": "^0.16.4",
67   "@types/three": "^0.152.1",
68   "@types/uuid": "^9.0.1",
69   "@typescript-eslint/eslint-plugin": "^5.56.0",
70   "@typescript-eslint/parser": "^5.56.0",
71   "autoprefixer": "^10.4.14",
72   "eslint": "^8.37.0",
73   "eslint-config-next": "^13.4.1",
74   "eslint-config-prettier": "^8.8.0",
75   "eslint-plugin-prettier": "^4.2.1",
76   "eslint-plugin-react": "^7.32.2",
77   "husky": "^8.0.3",
78   "lint-staged": "^13.2.0",
79   "postcss": "^8.4.21",
80   "prettier": "2.8.7",
81   "prettier-plugin-tailwindcss": "^0.2.6",
82   "prisma": "^4.15.0",
83   "tailwindcss": "^3.3.0",
84   "typescript": "^5.0.2"
85 }

```

3.2 Реализация модулей автоматизации процессов

3.2.1 Модуль регистрации и авторизации пользователей

Модуль представляет реализацию OAuth через login.mirea.ru с единовременной передачей данных пользователя.

Листинг 2 – Реализация OAuth

```
1 export default function MireaNinjaLksProvider(options: MireaNinjaLKSP providerConfig): Provider {
2     return {
3         id: 'lks',
4         name: 'LKS',
5         type: 'oauth',
6         version: '2.0',
7         accessTokenUrl: 'https://auth-app.mirea.ru/oauth/token',
8         requestTokenUrl: 'https://auth-app.mirea.ru/oauth/token',
9         authorization: {
10             url: 'https://auth-app.mirea.ru/oauth/authorize',
11             params: { scope: 'profile' },
12         },
13         token: {
14             url: 'https://auth-app.mirea.ru/oauth/token',
15         },
16         userinfo: {
17             url: 'https://auth-app.mirea.ru/api/?action=getData&url=https://lk.mirea.ru/profile/',
18         },
19         checks: ['state'],
20         async profile(profile: MireaProfile) {
21             const name = [profile.arUser.NAME, profile.arUser.LAST_NAME].join(' ')
22             return {
23                 id: profile.arUser.ID,
24                 name,
25                 nickname: await nicknameValidation(name),
26                 email: profile.arUser.LOGIN,
27                 emailVerified: new Date(),
28                 userInfo: null,
29                 role: Role.USER,
30                 image: 'https://lk.mirea.ru' + profile.arUser.PHOTO,
31                 isBlocked: false,
32                 blockReason: null,
33             }
34         },
35     },
36 }
```

```

35         clientId: options.clientId,
36         clientSecret: options.clientSecret,
37     }
38 }

```

3.2.2 Модуль бесконечных лент объявлений потерянных, найденных вещей

Листинг 3 – Реализация бесконечной ленты

```

1  function ScrollGridLoader({ visible }: { visible: boolean }) {
2      return visible ? (
3          <p className='col-span-2 flex justify-center py-5 text-center md:col-
4              span-4'>
5              <Spinner />
6              <span className='sr-only Загрузка'>...</span>
7              </p>
8          ) : null
9      }
10
10 function ScrollGridEndMessage() {
11     return (
12         <div className='col-span-2 flex flex-col items-center text-sm text-
13             gray-600 md:col-span-4'>
14             <Image
15                 src='/assets/svg-illustrations/items-not-found.svg'
16                 alt=''
17                 width={200}
18                 height={200}
19                 priority={false}
20             />
21             <p className='mt-2'> Чего-то нет </p>
22         </div>
23     )
24 }
25 interface InfiniteScrollGridWithFilterProps {
26     reason: PostItemReason
27 }
28
29 export default function InfiniteScrollGridWithFilter({
30     reason,
31 }: InfiniteScrollGridWithFilterProps) {
32     const { enabledSortOption, checkedFilters } = useScrollGridStore(((
33         state) => state[reason]))

```

```

33  const postsQuery = api.posts.infinitePosts.useInfiniteQuery(
34  {
35      limit: 12,
36      reason,
37      orderByCreationDate: enabledSortOption,
38      filters: checkedFilters,
39  },
40  { getNextPageParam: (lastPage) => lastPage.nextCursor },
41  )
42  const [posts, setPosts] = useState<RouterOutputs['posts']['
43      infinitePosts'][ 'items' ]>([])
44  const [hasMore, setHasMore] = useState(true)
45
46  useEffect(() => {
47      if (postsQuery.data) {
48          setPosts(postsQuery.data.pages.map((query) => query.
49              items).flat())
50          setHasMore(postsQuery.data.pages.at(-1)?.nextCursor
51              !== undefined)
52      }
53  }, [postsQuery.data])
54
55  const fetchMoreData = () => {
56      if (postsQuery.data && postsQuery.data.pages.length * 10 >=
57          500) {
58          setHasMore(false)
59          return
60      }
61  }
62  void postsQuery.fetchNextPage()
63
64  return (
65      <>
66      <GridFilter reason={reason} />
67      <InfiniteScroll
68          dataLength={posts.length}
69          next={fetchMoreData}
70          hasMore={hasMore}
71          loader={<ScrollIndicator visible={postsQuery.isFetchingNextPage &&
72              posts.length > 0} />}
73          className='grid grid-cols-2 gap-4 sm:grid-cols-2 md:grid-cols-4'
74      >
75      {posts.map((post) => (
76          <div key={post.id.toString()} className='relative'>
77              <PostCard post={post} displayReasonLabel={reason ===
78                  PostItemReason.ANY} />
79      
```

```
73             </div>
74         ))
75     </InfiniteScroll>
76     {(
77         postsQuery.isFetched && (!posts || posts.length === 0) && <
78             ScrollGridEndMessage />) || null}
79     </>
80     )
81 }
```

3.2.3 Модуль добавления и поиска утерянных вещей

Листинг 4 – Реализация бесконечной ленты

```
1 const inputs: { name: string; className: string; input: React.ReactNode }[] = [
2 {
3     name: 'Фотографии',
4     className: 'sm:col-span-6',
5     input: (
6     <>
7     <DynamicDropzone
8     images={post.images}
9     addImage={(img: string) =>
10         setPost((current) => ({ ...current, images: [...current.images,
11             img] }))
12     }
13     />
14     <div className='mt-6 flow-root'>
15     <ul role='list' className='my-5 divide-y divide-gray-200'>
16     {post.images.map((url, index) => (
17         <li key={index} className='py-4'>
18         <div className='flex items-center space-x-4'>
19         <div className='flex-shrink-0'>
20             <Image
21                 className='h-8 w-8 rounded-full'
22                 height={30}
23                 width={30}
24                 src={url}
25                 alt=''
26             />
27             </div>
28         <div className='min-w-0 flex-1'>
29             <p className='truncate text-sm font-medium text-gray-900'>
30                 {url.split('/').at(-1)}
31             </p>
32         </div>
33     )}
```

```

32         <div>
33             <button
34                 onClick={() =>
35                     setPost({
36                         ...post,
37                         images: post.images.filter((currUrl, currIndex
38                             ) => index !== currIndex),
39                     })
40             className='inline-flex items-center rounded-full border border
41                 -gray-300 bg-white px-2.5 py-0.5 text-sm font-medium
42                 leading-5 text-gray-700 shadow-sm hover:bg-gray-50'
43             >
44                 <XMarkIcon className='h-5 w-5' />
45             </button>
46         </div>
47     </div>
48     ))})
49 </ul>
50 </div>
51 </>
52 ) ,
53 },
54     name: 'Название',
55     className: 'sm:col-span-3',
56     input: (
57         <div className='flex flex-col sm:flex-row'>
58             <input
59                 maxLength={100}
60                 value={post.name}
61                 onChange={(e) => setPost((value) => ({ ...value, name: e.target.value
62                     }))}
63                     type='text'
64                     name='name'
65                     id='name'
66                     className='w-full rounded-md border-gray-300 shadow-sm focus:border-
67                         blue-500 focus:ring-blue-500 sm:text-sm'
68             />
69             {post.images.length > 0 && (
70                 <button
71                     onClick={(e) => {
72                         e.preventDefault()
73                         void generateImageCaption.mutateAsync({
74                             imageUrl: post.images[0] ?? '' })
```

```

72         })
73         disabled={generateImageCaption.isLoading}
74         className={'mt-2 inline-flex items-center justify-center
75             rounded-md border border-transparent bg-blue-600 px-4 py-2
76             text-sm font-medium text-white shadow-sm hover:bg-blue-700 focus:outline-none focus:ring-2 focus:ring-blue-500
77             focus:ring-offset-2 disabled:opacity-50 sm:ml-3 sm:mt-0 ${
78                 generateImageCaption.isLoading ? 'cursor-not-
79                 allowed' : ''
80             }'}
81         >
82         {generateImageCaption.isLoading ? (
83             <span>Загрузка...</span>
84         ) : (
85             <>
86             <Image
87                 src='/icons/magic-icon.svg'
88                 alt=''
89                 width={10}
90                 height={10}
91                 className='mr-1 h-5 w-5'
92                 style={{ filter: 'invert(1)' }}
93                 aria-hidden='true'
94             />Сгенерировать
95         )
96     </button>
97   )
98 },
99 {
100   name: 'Описание',
101   className: 'sm:col-span-6',
102   input: (
103     <textarea
104       maxLength={512}
105       value={post.description}
106       onChange={(e) => setPost((value) => ({ ...value, description: e.target
107         .value }))}
108       id='description'
109       name='description'
110       rows={3}
111       className='block w-full rounded-md border-gray-300 shadow-sm focus:
112         border-blue-500 focus:ring-blue-500 sm:text-sm'

```

```

111     placeholderМесто='находденияиликонтактнаяинформация' ,
112     />
113   ) ,
114 },
115 {
116   name: 'Кампус',
117   className: 'sm:col-span-3',
118   input: (
119     <select
120       id='campus'
121       name='campus'
122       onChange={({e) => setPost({ ...post, campus: e.target.value as
123         PrismaCamus })})
124       className='block w-full rounded-md border-gray-300 shadow-sm focus:
125         border-blue-500 focus:ring-blue-500 sm:text-sm'
126     >
127       {Object.values(PrismaCamus).map((value, index) => (
128         <option key={index} value={value}>
129           {Campus[value]}
130         </option>
131       )))
132     </select>
133   ),
132 },
133 ]

```

3.2.4 Модуль генерации описания объявлений

Листинг 5 – Реализация бесконечной ленты

```

1 import os
2 import re
3 import uuid
4 from langchain.document_loaders import ImageCaptionLoader
5 from langchain.indexes import VectorstoreIndexCreator
6
7 from langchain.vectorstores import Chroma
8 from fastapi import FastAPI, File, UploadFile, Response
9
10 app = FastAPI(title="Image Captioning API")
11
12 os.environ["TOKENIZERS_PARALLELISM"] = "true"
13
14
15 @app.post("/image-caption")
16 def image_caption(file: UploadFile = File(...)):

```

```

17     if file.content_type not in ["image/jpeg", "image/png"]:
18         return Response(status_code=415, content="Unsupported Media
19             Type")
20
21
22     try:
23         with open(random_filename, "wb") as buffer:
24             buffer.write(file.file.read())
25
26         loader = ImageCaptionLoader(path_images=random_filename)
27         list_docs = loader.load()
28
29         index = VectorstoreIndexCreator(
30             vectorstore_kwargs={"collection_name": random_filename}
31             ).from_documents(list_docs)
32
33         res = index.query("Что
34             " изображено на фотографии ? Только описание предмета ,
35             безводных фраз потипуна      ' фотографии изображено '.
36             Ответ на русском языке для заголовка фотографии : "
37             )
38
39         res = res.strip()
40         res = re.sub(r"\.$", "", res)
41
42         return res
43
44     finally:
45         os.remove(random_filename)
46
47
48
49     @app.on_event("startup")
50     async def startup_event():
51         print("Starting up...")
52
53         from transformers import BlipForConditionalGeneration, BlipProcessor
54
55         # Download model and configuration from huggingface.co and cache .
56         processor = BlipProcessor.from_pretrained(blip_processor)
57         model = BlipForConditionalGeneration.from_pretrained(blip_model)
58
59         print("Model loaded")

```

Вывод по разделу

Были разработаны модули программы и интегрированы в приложение.

4 ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

4.1 Планирование разработки системы

Основой данной выпускной квалификационной работы является «Разработка системы для поиска и возврата утерянных вещей». В данном разделе определяются уровень сложности и затраты на создание программного и аппаратного обеспечения, а также оценивается экономическая выгода, которую можно получить от использования разрабатываемого программного обеспечения.

4.1.1 Определение трудоемкости и продолжительности работ по созданию УСПД

В этом разделе определяются уровень сложности и затраты на создание программного и аппаратного обеспечения, а также проводится оценка экономической выгоды от использования разрабатываемого ПО. Процесс разработки включает анализ предметной области, имитационный анализ, создание, настройку и тестирование системы.

- Техническое задание. Регламентированы следующие этапы исследования: составление технического задания, включающего формулировку задач, подбор литературы, сбор исходных данных, определение системных требований, а также определение этапов, фаз и сроков разработки программного обеспечения.
- Эскизный проект. Этот этап включает в себя использование программных средств для анализа схожих тем, разработки общих программных структур и структур по подсистемам, создания прототипов и документации.
- Технический проект. Этот этап включает в себя определение требований к программному обеспечению и выбор инструментов и использование программных средств.
- Рабочий проект. Этап включает компоновку и дизайн, программирование, тестирование и отладку ПО, проектирование плат, а также координацию и утверждение работоспособности всей системы.

- Внедрение. Подразумевает под собой использование на реальной инфраструктуре; анализ полученных данных в результате исследований, благодаря которым можно будет скорректировать техническую документацию.

Трудоемкость работ по созданию программного обеспечения носит вероятностный характер, поскольку определяется суммой сложности этапов и видов работ, оцениваемых экспертами в ручные дни, и зависит от многих факторов, которые трудно учесть.

Трудоемкость каждого вида работ определяется по формуле (1).

$$t_i = \frac{3 \cdot t_{min} + 2 \cdot t_{max}}{5}, \quad (1)$$

где t_{min} — минимально возможная трудоемкость выполнения отдельного вида работ;

t_{max} — максимально возможная трудоемкость выполнения отдельного вида работ.

Различные виды работ имеют свою продолжительность в календарных днях (T_i), определяясь по формуле (2), в днях:

$$T_i = \frac{t_i}{\Psi_i} \cdot K_{вых}, \quad (2)$$

где t_i — трудоемкость работ, человеко-дней;

Ψ_i — численность исполнителей, человек;

$K_{вых}$ — коэффициент, учитывающий выходные и праздничные дни, находится по формуле (3):

$$K_{вых} = \frac{K_{кал}}{K_{раб}}, \quad (3)$$

где $K_{кал}$ — календарные дни;

$K_{раб}$ — рабочие дни;

$K_{вых}$ — 1,48.

Далее предоставляется перечень разновидностей и стадий рабочей деятельности по разработке ПО, экспертные оценки, а также рассчитываемые переменные их трудоемкости, а также продолжительность каждого вида работ, рассчитанные по формулам (1) и (2), представлены в таблице 2.

Таблица 2 – Расчет трудоемкости и продолжительности работ по созданию ПО и аппаратных средств календарных дней

№ работы	Стадии разработки	Трудоемкость, чел. дни			Количество работников, чел.	Продолжительность работ, календарные дни
		t_{min}	t_{max}	t_i		T_i
Техническое задание						
1	- постановка задачи	3	6	4.2	1	6.2
2	- подбор литературы	2	3	2.4	1	3.6
3	- сбор исходных данных	4	5	4.4	1	6.5
4	- определение требований к системе	3	4	3.4	1	5.0
5	- определение стадий, этапов и сроков разработки ПО	2	3	2.4	1	3.6
Эскизный проект						
6	- анализ программных средств схожей тематики	7	8	7.4	1	11.0
7	- разработка общей структуры ПО	4	8	5.6	1	8.3
8	- разработка структуры программы и подсистем	5	8	6.2	1	9.2
9	- создание прототипа	4	6	4.8	1	7.1
10	- документирование	2	3	2.4	1	3.6
11	- определение требований к ПО	3	4	3.4	1	5.0
12	- выбор инструментальных средств	3	4	3.4	1	5.0

Продолжение таблицы 2.

№ работы	Стадии разработки	Трудоемкость, чел. дни			Количество работников, чел.	Продолжительность работ, календарные дни
Рабочий проект						
13	- программирование	20	25	22	1	32.6
14	- тестирование и отладка ПО	7	8	7.4	1	11.0
15	- разработка программной документации	4	6	4.8	1	7.1
16	- согласование и утверждение работоспособности системы	2	3	2.4	1	3.6
Внедрение						
17	- опытная эксплуатация	6	8	6.8	1	10.1
18	- анализ данных, полученных в результате эксплуатации	2	4	2.8	1	4.1
19	- корректировка технической документации по результатам испытаний	1	2	1.4	1	2.1
	Общая трудоемкость разработки			98		144

Следовательно, общая продолжительность проведения работ $T_i = 144$.

4.1.2 Построение ленточного графика проведения исследования

В качестве инструмента планирования используется ленточный график (13).

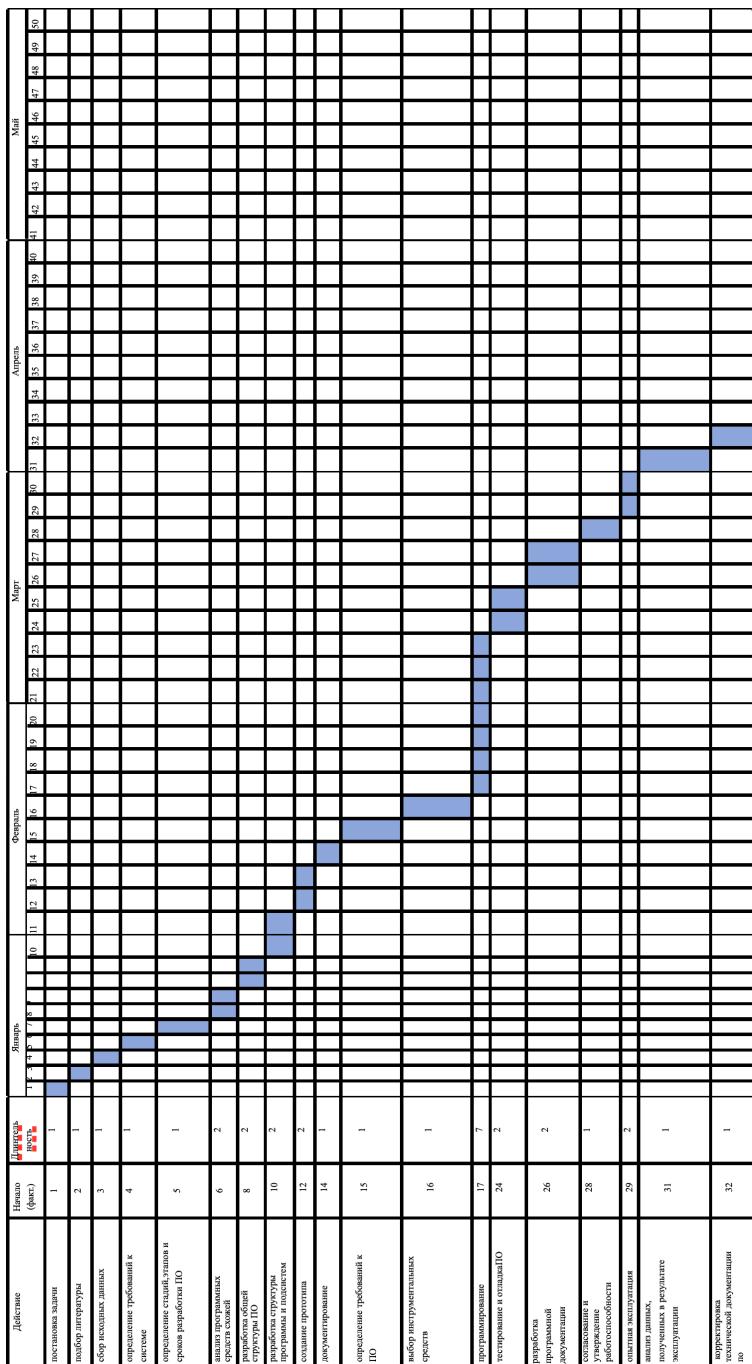


Рисунок 13 – Ленточный график

Подобный график позволяет наглядно представить логическую последовательность и взаимосвязь отдельных работ. График представляет собой таблицу с перечислением названий стадий разработки, видов работ, длительность выполнения работ. Данный график построен по данным таблицы 2. В этом графике временная единица выполнения работ оценивается в 3 дня.

4.2 Расчет сметы затрат на разработку представленной работы

Сметная стоимость проектирования и внедрения программы включает в себя затраты, определяемые по формуле (4):

$$C_{\text{пр}} = C_{\text{осн}} + C_{\text{доп}} + C_{\text{соц}} + C_{\text{м}} + C_{\text{маш.вр}} + C_{\text{н}}, \quad (4)$$

где $C_{\text{пр}}$ — стоимость разработки ПО;

$C_{\text{осн}}$ — основная заработка плата исполнителей;

$C_{\text{доп}}$ — дополнительная заработка плата исполнителей, учитывающая потери времени на отпуска и болезни (принимается в среднем 10% от основной заработной платы);

$C_{\text{соц}}$ — отчисления в фонд социального страхования – 30% от основной и дополнительной заработной платы;

$C_{\text{м}}$ — затраты на используемые материалы;

$C_{\text{маш.вр}}$ — стоимость машинного времени;

$C_{\text{н}}$ — накладные расходы включают затраты на управление, уборку, ремонт, электроэнергию, отопление и др. (принимаются в размере 60% от основной и дополнительной заработной платы).

4.2.1 Основная заработка плата исполнителей

На статью «Заработка плата» относят заработную плату научных, инженерно-технических и других работников, непосредственно участвующих в разработке ПО. Расчет ведется по формуле (5):

$$З_{\text{исп}} = З_{\text{ср}} \cdot T, \quad (5)$$

где $З_{\text{исп}}$ — заработка плата исполнителей (руб.);

$З_{\text{ср}}$ — средняя тарифная ставка работника организации разработчика ПО (руб./чел./дни);

T — трудоемкость разработки ПО (чел.дни).

$З_{\text{ср}}$ определяется по формуле (6):

$$З_{\text{ср}} = \frac{C}{\Phi_{\text{мес}}}, \quad (6)$$

где C — зарплата труда на текущий момент времени (руб./мес.);
 $\Phi_{\text{мес}}$ — месячный фонд рабочего времени исполнителя (дни).

Затраты на статью «Заработной платы» приведены в таблице 3.

Таблица 3 – Затраты на заработную плату

№	Исполнитель	Оклад, руб./мес.	Оклад, руб./мес.	Трудоемкость, чел.дни	Сумма, руб.
1	Инженер-программист	70000	3500	98	343000
Общая основная заработная плата исполнителей, $C_{\text{осн}}$				98	343000

4.2.2 Дополнительная заработка исполнителей

Дополнительная заработка на период разработки ПО рассчитывается относительно основной и составляет 10% от её величины, рассчитывается по формуле 7.

$$C_{\text{доп}} = C_{\text{осн}} \cdot 0,1 = 34300 \text{ (руб.)}. \quad (7)$$

4.2.3 Расчет отчислений на социальное страхование

Отчисления на социальное страхование рассчитываются относительно выплаченной заработной платы по формуле (8). Составляют 30

$$\begin{aligned} C_{\text{соц}} &= (C_{\text{доп}} + C_{\text{доп}}) \cdot 0,3 \\ C_{\text{соц}} &= (34300 + 343000) \cdot 0,3 = 113190 \text{ (руб.)} \end{aligned} \quad (8)$$

На эту статью относят все затраты на магнитные носители данных, бумагу, для печатных устройств, канцтовары и др. Затраты по ним определяются по экспертным оценкам. Расчет расходов на материалы приведен в таблице 4.

Таблица 4 – Расчёт расходов на материалы

№	Материалы	Количество, штуки	Стоимость, рубли
1	Бумага писчая, листов	1000	800
2	Картридж для принтера, шт.	1	1000
3	Другие канцтовары	-	1000
Общая стоимость материалов, C			2800

4.2.4 Накладные расходы

На статью «Накладные расходы» относят расходы, связанные с управлением и организацией работ. Накладные расходы рассчитываются относительно основной заработной платы. Величина накладных расходов принимается равной 60 % от основной зарплаты исполнителей. Формула расчета (9):

$$C_{\text{н}} = C_{\text{осн}} \cdot K, \quad (9)$$

где $C_{\text{н}}$ — накладные расходы;

$C_{\text{осн}}$ — основная заработка плата исполнителей;

K — коэффициент учета накладных расходов.

$$C_{\text{н}} = 343000 \cdot 0,6 = 205800 \text{ (руб.)}$$

4.2.5 Расчет стоимости машинного времени

Затраты на машинное время, необходимое для разработки ПО, расходы на приобретение и подготовку материалов научно-технической информации, расходы на использование средствами связи. Расчет затрат на машинное время осуществляется по формуле (10):

$$C_{\text{маш.вр}} = K_{\text{маш.вр}} \cdot З_{\text{маш.вр}} \quad (10)$$

где $маш.вр$ — тарифная стоимость одного часа машинного времени ($маш.вр = 60 \text{ (руб./час)}$);

$З_{\text{маш.вр}}$ — машинное время, используемое на проведение работ.

Необходимое количество машинного времени для реализации проекта

по разработке программы рассчитывается по формуле (11):

$$З_{\text{маш.вр}} = t_i \cdot T_{\text{см}} \cdot T_{\text{ср.маш}}, \quad (11)$$

где t_i — трудоемкость работ, чел.дней;

$T_{\text{см}}$ — продолжительность рабочей смены (При пятидневной рабочей неделе $T_{\text{см}} = 8\text{ч.}$);

$T_{\text{ср.маш}}$ — средний коэффициент использования машинного времени ($T_{\text{ср.маш}} = 0,7$).

Из этого следует:

$$З_{\text{маш.вр}} = 98 \cdot 8 \cdot 0,7 = 548,8 \text{ (ч.)}$$

Стоимость машинного времени составит:

$$C_{\text{маш.вр}} = 60 \cdot 548,8 = 32928 \text{ (руб.)}$$

Результаты расчета затрат на проектирование программного обеспечения сведены в таблице 5.

Таблица 5 – Расчёт расходов на материалы

№	Наименование статей	Обозначение	Сумма, руб.	В % к итогу
1	Основная заработка плата	$C_{\text{осн.}}$	343000	46,86
2	Дополнительная заработка плата	$C_{\text{доп.}}$	34300	4,69
3	Отчисления на социальные нужды	$C_{\text{соц.}}$	113190	15,46
4	Материалы	$C_{\text{м.}}$	2800	0,38
5	Стоимость машинного времени	$C_{\text{маш.вр.}}$	32928	4,5
6	Накладные расходы	$C_{\text{н.}}$	205800	28,11
Итого:		$C_{\text{осн.}}$	732018	100

Следовательно, себестоимость разработки составляет 732018 руб.

Данная программа может быть реализована на рынке. При расчетном количестве реализованных программ ($n = 5$) оптовая цена программы ($\Pi_{\text{опт}}$)

может быть рассчитана по формуле (12):

$$\Pi_{\text{опт}} = \frac{C_{\text{пр}}}{n} + \Pi \quad (12)$$

где $C_{\text{пр}}$ — себестоимость разработки программы;
 Π — прибыль, определяется по формуле (13):

$$\Pi_i = Y_p \cdot \frac{C_{\text{пр}}}{n} \cdot 100 \quad (13)$$

где Y_p — средний уровень рентабельность ($Y_p = 20\%$).

Таким образом, оптовая цена программы составит:

$$\Pi_{\text{опт}} = \frac{732018}{5} + \left(\frac{732018}{5} \cdot 0,2 \right) = 146403,6 + 29280,72 = 175684,32 \text{ (р.)}$$

Отпускная цена реализации программы потребителям ($\Pi_{\text{опт}}$), рассчитывается по формуле:

$$\Pi_{\text{опт}} = \Pi_{\text{опт}} + \text{НДС}$$

где НДС — налог на добавленную стоимость, рассчитывается в соответствии с действующей ставкой этого налога — 20% от оптовой цены программы.

$$\Pi_{\text{опт}} = 175684,32 + (175684,32 \cdot 0,2) = 210821,18 \text{ (руб.)}$$

Следовательно, отпускная цена программы составит 175684,32 руб., в том числе НДС — 35136,86 руб.

ЗАКЛЮЧЕНИЕ

Разработано приложение finds.mirea.ru 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.

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 // pona1.ru: бюро находок Пона.рф. Удобный поиск по объявлениям, большая база потерянных вещей и животных URL: <https://pona1.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).

ПРИЛОЖЕНИЕ А

Схема базы данных

```
1  model Account {
2      id          String    @id @default(cuid())
3      userId      String
4      type        String
5      provider     String
6      providerAccountId String
7      refresh_token  String?
8      access_token   String?
9      expires_at    Int?
10     token_type     String?
11     scope         String?
12     id_token       String?
13     session_state  String?
14     user          User     @relation(fields: [userId],
15                                         references: [id], onDelete: Cascade)
16
17 }
18
19 model Session {
20     id          String    @id @default(cuid())
21     sessionToken String    @unique
22     userId      String
23     expires      DateTime
24     user          User     @relation(fields: [userId], references:
25                                         [id], onDelete: Cascade)
26
27 }
28
29 model User {
30     id          String    @id @default(cuid())
31     name        String?
32     nickname    String    @unique
33     socialNetworks UserSocialNetwork []
34     email        String?    @unique
35     emailVerified DateTime?
36     userInfo     String?    @db.VarChar(280)
37     role         Role      @default(USER)
38     image        String?
39     isBlocked    Boolean   @default(false)
40     blockReason  String?
41     accounts     Account []

```

```

42         sessions          Session []
43         lostAndFoundItems LostAndFoundItem []
44
45         @@index([ id ], type: Hash)
46         @@index([ nickname ], type: Hash)
47     }
48
49     model VerificationToken {
50         identifier String
51         token      String    @unique
52         expires    DateTime
53
54         @@unique([ identifier , token ])
55     }
56
57     model UserSocialNetwork {
58         id           String
59                     @id @default(cuid())
60         socialNetwork SocialNetwork
61         link          String
62         userId        String
63         user          User
64                     @relation(fields: [userId],
65                     references: [id], onDelete: Cascade)
66         lostAndFoundItemSocialNetworks LostAndFoundItemSocialNetworks []
67
68         @@unique([ userId , socialNetwork ])
69         @@index([ socialNetwork , userId ])
70     }
71
72     enum Role {
73         USER
74         MODERATOR
75         ADMIN
76     }
77
78     model LostAndFoundItem {
79         id           String
80                     @id @default(
81                     cuid())
82         name          String
83                     @db.VarChar(100)
84         description   String
85                     @db.VarChar(512)
86         campus        Campus
87         reason        PostItemReason

```

```

81         status          LostAndFoundItemStatus      @default(
82             ACTIVE)
83         images          String []
84         userId          String
85         user            User                      @relation(
86             fields: [userId], references: [id], onDelete: Cascade)
87         socialNetworks LostAndFoundItemSocialNetworks []
88         created         DateTime                 @default(now())
89         expires         DateTime                @default(
90             dbgenerated("NOW() + interval '1 week'"))
91
92     }
93
94     enum LostAndFoundItemStatus {
95         ACTIVE
96         EXPIRED
97         BLOCKED
98     }
99
100    model LostAndFoundItemSocialNetworks {
101        id              String          @id @default(cuid())
102        lostAndFoundItemId String
103        lostAndFoundItem   LostAndFoundItem @relation(fields: [
104            lostAndFoundItemId], references: [id], onDelete: Cascade)
105        userSocialNetworkId String
106        userSocialNetwork UserSocialNetwork @relation(fields: [
107            userSocialNetworkId], references: [id], onDelete: Cascade)
108
109        @@unique([lostAndFoundItemId, userSocialNetworkId])
110    }
111
112    enum PostItemReason {
113        LOST
114        FOUND
115    }
116
117    enum Campus {
118        V78
119        S20
120        V86
121        MP1
122        SG22
123        SHP23
124        U7

```

```
121     }
122
123 enum SocialNetwork {
124     TELEGRAM
125     VK
126 }
```