**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

Борушко Дмитрий , Равинский Радион

**Исследование предметной области, пользователей и контекстов**

**Отчет по лабораторной работе №1**

«Проектирование человеко-машинных интерфейсов» студента 4 курса 13 группы

Преподаватель

Романюк Вадим Николаевич

2024

**Постановка задачи**

**Задача:** Спроектировать Приложение для социальной сети,

позволяющее пользователям создавать профиль, публиковать заметки с

указанием типа доступа (для себя, для друзей, для всех), а также добавлять

изображения, видео и гиперссылки.

**Стратегия дизайна**

**Заинтересованные стороны:** Совершеннолетние и несовершеннолетние мужчины и женщины любого достатка, активно использующие интернет в своей повседневной жизни.

**Видение продукта заинтересованными лицами (задачи продукта):** Приложение должно предоставлять пользователям возможность легко создавать и управлять своим профилем, делиться контентом и взаимодействовать с другими пользователями, используя браузер и/или мобильные устройства (Android или iOS).

**Конфликты и противоречия:** приложением не получиться воспользоваться с мобильного устройства, если пользователь использует операционную систему iOS. Тем не менее, даже в этом случае пользователь по-прежнему сможет запустить приложение в своём браузере (например, Safari).

**Задачи бизнеса (верифицируемые), задачи маркетинга и брендинга:** Приложение позволит заказчику минимизировать обработку запросов, обеспечивая высокую степень автоматизации управления контентом и взаимодействия с пользователями.

**Измеримые критерии успешности:** Количество зарегистрированных пользователей, количество опубликованных заметок, активность пользователей (лайки, комментарии), отзывы пользователей.

**Технические возможности и ограничения:**

**1. Веб-приложение:**

Язык программирования Python версии 3.9, библиотека Flask.

**2. Мобильное приложение:**

Технологии: Kotlin, Kotlin Coroutines, Android Architecture Components(clean architecture) (view model, live data), Retrofit, Okhttp, Room DB.

**3. База данных:**

PostgreSQL

**4. Технологии сборки, доставки и развертывания приложения:**

Анализ кода – SonarQube, инструменты CI/CD – Gitlab CI/Jenkins, инструменты для развертывания приложения – Docker, Google Cloud Platform.

**Представления заинтересованных лиц о пользователях (целевая аудитория):** мужчины/женщины 10-45 лет любого достатка, активно использующие интернет в своей повседневной жизни

**График работы:** суббота/воскресенье.

**Бюджет:** жетон метро и билетик на автобус до университета (опционально).  
  
**Профиль группы (профиль пользователя, среды и задач)  
  
Профиль пользователя:**

| **Группа 1** | **Группа 2** |
| --- | --- |
| 10-18 лет | 18-30 лет |
| В процессе получения образования | Законченное высшее образование |
| Незамужние/неженатые | Незамужние/неженатые |
| Нет опыта работы | Срок работы в основном от 1 лет и выше |
| Продукт знаю | С продуктом знакомы |
| Основы компьютерной грамотности | Продвинутые пользователи компьютера |

**Профиль среды:**

| **Характеристика** | **Признак** | **Влияние на интерфейс** |
| --- | --- | --- |
| Место использования | Хорошее \ плохое освещение | Регулировка яркости экрана, смена темы в зависимости от освещения |
| Разрешение экрана | ● 1920x1080  ● 1600x720  ● 2400x1080  ● Другие | ● общий размер экрана интерфейса  ● границы интерфейса |
| Устройства ввода/вывода | ● Сенсорный экран  ● Клавиатура  ● Мышь  ● Графический планшет | ● возможность использования периферийных устройств  ● установка драйверов  ● дополнительные элементы для ввода/вывода |
| Программное обеспечение (браузер, мобильное приложение) | ● Браузер (Google Chrome, Opera, Mozila Firefox)  ● Версия браузера  ● Особенности установки мобильных приложений (apk, IPA) | ● соответствующий дизайн  ● возможность установки расширений  ● элементы для получения обновлений (для мобильных приложений) |
| Устройство взаимодействия с приложением (ПК, мобильное устройство, планшет) | ● Операционная система (Windows, iOS, Android)  ● Версия операционной системы | ● совместимость  ● производительность  ● размещение элементов в сочетании с встроенными функциями ОС |

**Профиль задач:**

Пользователь соц-сети:

1. Очень важная задача, выполняется редко - Регистрация нового пользователя и создание профиля

2. Очень важная задача, выполняется часто - Просмотр ленты новостей

3. Очень важная задача, выполняется часто - Лайк новости

4. Менее важная задача, выполняется в обычном режиме - Просмотр лайкнутых новостей

5. Очень важная задача, выполняется в обычном режиме - Поиск по ключевому слову

6. Важная задача, выполняется редко - Удаление новости из списка избранных

Приоритет, по которому следует реализовать функционал онлайн-магазина:

1. Регистрация нового пользователя

2. Просмотр/редактирование новостей в базе данных

3. Добавление новости в лайкнутые

4. Просмотр лайкнутых новостей

5. Фильтрация новостей по категориям.

6. Удаление лайкнутой новости

**Проанализировать задачи и роли пользователей**

| **Составляющие процесса продажи** | **Роли** | |
| --- | --- | --- |
| **Пользователь** | **Администратор** |
| Регистрация | + |  |
| Регистрация редактора/ администратора |  | + |
| Поиск пользователя |  | + |
| Просмотр аккаунта пользователя | + | + |
| Ввод/редактирование данных пользователя | + |  |
| Просмотр истории публикаций | + |  |
| Просмотр отчетов |  | + |
| Просмотр списка новостей | + | + |
| Добавление новостей | + |  |
| Просмотр новости | + | + |
| Удаление комментариев |  | + |
| Проверка статуса публикации |  | + |
| Удаление новости |  | + |
| Проверка валидности новостей |  | + |
| Проверка валидности комментариев |  | + |

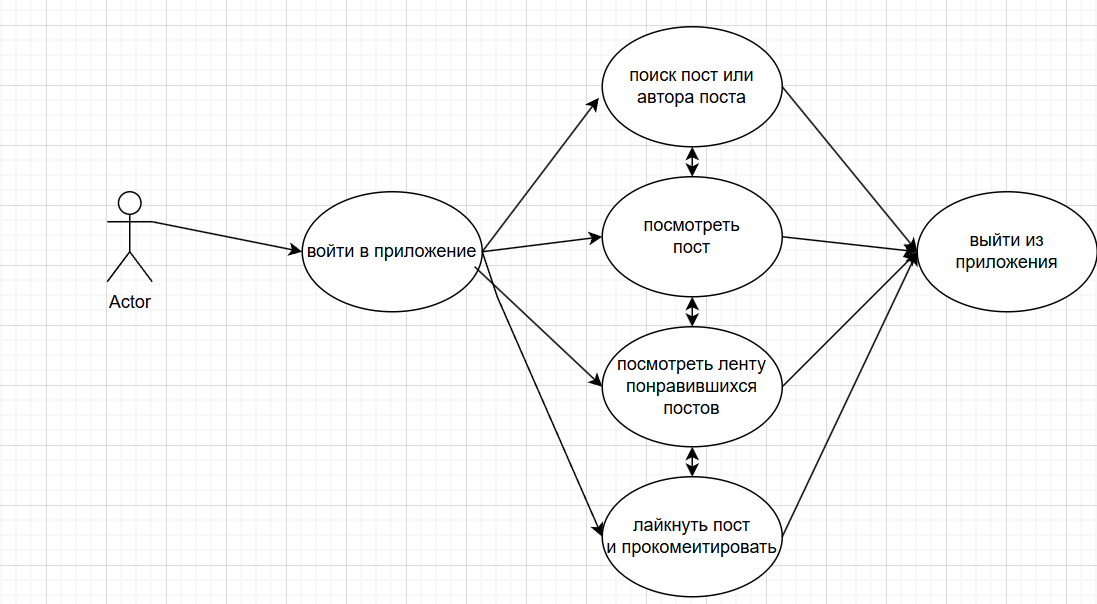
**Разработать объектную модель**

**5.1 Пример объектной модели**

| **Объект** | **Мощность** | **Представления** | **Действия** | **Атрибуты** |
| --- | --- | --- | --- | --- |
| Пользователь | Тысячи | • Детальное | •Просмотреть  • Изменить | • ФИО  • телефон  • email  • логин  • пароль |
| Новости | Тысячи | • Список  • Детальное | •Просмотреть  •Опубликовать  • Удалить  • Искать | • описание  • изображение  • автор  • категории  • название  • др. |
| Поиск | Один | • Список | • Искать  • Очистить | • текстовое поле  • по названию  • по категории  • по автору |
| Список публикаций | Один | • Список | • добавить  • удалить  • просмотреть | • публикации |
| “Избранное” | Один | • Список | • добавить  • удалить  •просмотреть | • публикации  • дата добавления |
| Категории | Сотни | • Список | •просмотреть  • искать | • название  • описание  • количество публикаций |

**5.2 Пример соответствия объектов персонажам**

| **Объекты** | **Персонажи** | |
| --- | --- | --- |
| **Объект** | **Лера** | **Макс** |
| Пользователь | ✔ | ✔ |
| Публикация | ✔ | ✔ |
| Поиск | ✔ | ✔ |
| “Избранное” |  | ✔ |
| Обратная связь | ✔ | ✔ |
| Категории | ✔ | ✔ |
| Подписка |  | ✔ |
| Комментарии | ✔ |  |

**Разработать диаграммы вариантов использования  
  
Сценарии вариантов использования**

| **Вход в сервис** | |
| --- | --- |
| 1. Пользователь нажимает кнопку “Войти” | Отображается окно входа в систему |
| 1. Пользователь вводит данные для входа   ● Исключение №1: неверные данные  ● Исключение №2: пользователь не найден | Система проверяет правильность данных |
| 1. Пользователь вошел в систему | Отображается главная страница сервиса |
| **Исключения** | |
| Исключение №1: неверные данные | Пользователю отображается сообщение, пользователь пытается войти снова |
| Исключение №2: пользователь не найден | Пользователю отображается сообщение, пользователь переходит к сценарию “Регистрация” |

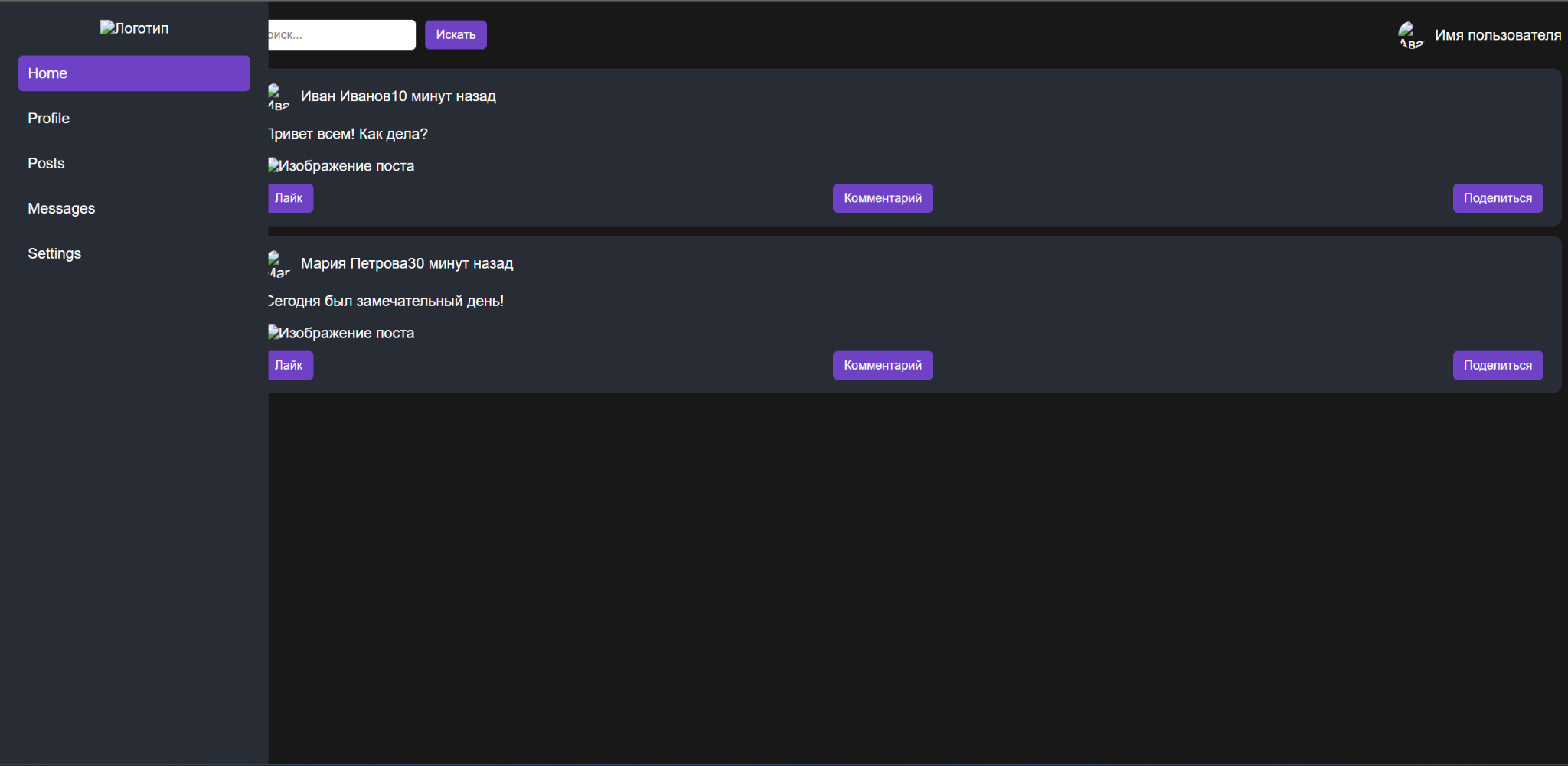
| **Регистрация** | |
| --- | --- |
| 1. Пользователь нажимает кнопку “Регистрация” | Отображается окно регистрации в системе |
| 1. Пользователь вводит личные данные   ● Исключение №1: неверные данные | Система проверяет правильность и полноту данных (карта) |
| 1. Пользователь пред заполняет способ оплаты   ● Исключение №1: неверные данные |
| 1. Пользователь нажимает кнопку “Зарегистрироваться” | Система добавляет пользователя в базу, отправляет письмо подтверждения на email |
| 1. Пользователь вводит код из письма подтверждения   ● Исключение №2: неверный код | Система проверяет правильность кода, подтверждает пользователя, переводит к сценарию “Вход в систему” |
| **Исключения** | |
| Исключение №1: неверные данные | Пользователю отображается сообщение, пользователь пытается ввести данные снова |
| Исключение №2: неверный код | Пользователю отображается сообщение, пользователь пытается ввести код снова |

| **Посмотреть новость** | |
| --- | --- |
| 1. Пользователь открывает страницу с новостями   ● Исключение №1: API недоступно | Система проверяет доступ к API |
| 1. Пользователь выбирает новость |
| 1. Пользователь просматривает новость и нажимает кнопку “нравится”   ● Исключение №2: новость уже есть в лайкнутых | Система проверяет лайкнутые новости |
| **Исключения** | |
| Исключение №1: API недоступно | Пользователю отображается сообщение, что сервис в данный момент недоступен |
| Исключение №2: новость уже лайкнута | Пользователю отображается сообщение, что новость уже лайкнута |

| **Удалить новость** | |
| --- | --- |
| 1. Удаление новости из лайкнутых | Система проверяет правильность данных |
| 1. Пользователь возвращается к окну с лайкнутыми новостями | Система выводит уведомление об удалённой новости |

| **Поиск новости по ключевому слову** | |
| --- | --- |
| 1. Пользователь переходит на страницу поиска | Отображается поисковая строка |
| 1. Пользователь вводит ключевое слово | Система ищет все возможные новости, где в заголовке или теге имеется это слово |
| 1. Пользователь выбирает интересующую его новость | Отображается детальная информация по новости |

**Разработать информационную архитектуру веб-приложения**

**  
**