

Университет ИТМО  
Факультет ФПИ и КТ

Лабораторная работа №1  
По дисциплине  
“Архитектура программных систем”

Выполнил:

Кочнев Р. Д.

Группа: Р33081

Преподаватель:

Перл И.А.

Санкт-Петербург, 2023

## Текст задания

Выбрать любую реально существующую систему и описать её в терминах UML. Желательно, чтобы система была не полностью информационной, но опиралась на информационную систему как показано в примере на лекции (Point of sale). Необходимо описать границы системы на разных уровнях, а также описать сценарии использования для нескольких Акторов.

Отчёт по работе должен содержать:

1. Титульный лист с указанием автора и номера группы
2. Само задание
3. Описание рассматриваемой системы с требованиями к ней
4. Формальное описание системы с необходимым количеством UML диаграмм
5. Словесное описание сценариев использования для рассматриваемых акторов

## Рамблер

### 1 Описание продукта

#### 1.1 Product functions (Функционал продукта)

Разрабатываемая система должна осуществлять поиск информации и других веб-сайтов в интернете. Также необходимо наличие сервиса электронной почты, сервиса показа новостей с делением по различным категориям, сервиса онлайн-игр, сервиса онлайн-знакомств, погода. Каждый из сервисов должен отображаться на главной странице-поиске в виде виджета, все сервисы должны быть привязаны к единому аккаунту, объединяющему все сервисы.

#### 1.2 User characteristics (Описание пользователей)

Группа пользователей	Взаимодействие с системой
Администраторы	Управление всеми процессами системы, просмотр различных отчетов, возможность изменения алгоритмов и прочее
Менеджер	Просмотр финансовых отчетов, работа с рекламой

Тех. поддержка	Доступ к техническим конфигурациям системы, изменение и просмотр данных пользователей
Редактор	Редактирование новостей, добавление контента в разные сервисы
Пользователь	Возможность пользования всеми предоставленными сервисами (персонализированный поиск и новости, почта, игры и другое)
Гость	Поиск в интернете без персонализированных результатов, возможность просмотра новостей, а также возможность зарегистрироваться в системе

### 1.3 Assumptions and dependencies (Влияющие факторы и зависимости)

Веб-приложение должно предусматривать зависимость от сторонних API, связанных с предоставлением данных о погоде, новостях и др.

### 1.4 Constraints (Ограничения)

Веб-приложение должно отображаться корректно во всех браузерах, начиная с IE10. Пользователь любого браузера должен иметь полный доступ ко всем возможностям портала в соответствии с его ролью.

Приложение должно безопасно хранить конфиденциальные данные пользователей и запрещать несанкционированный доступ к данным.

Приложение должно регулярно совершать резервное копирование во избежание потери данных пользователей.

## **2. Specific Requirements (Спецификация требований)**

### **2.1 Functionality (Функциональные требования)**

2.1.1 Поисковой робот должен парсить очередную веб-страницу, выделяя из нее главное содержимое, основные мета-теги, переходить по всем возможным ссылкам, составлять карту сайта (sitemap) и выделять ключевые слова.

2.1.2 Система должна ежедневно индексировать 100.000.000 страниц.

2.1.3 Система индексации должна уметь сопоставлять ключевым словам в запросе индексированные веб-страницы, при этом ранжировать страницы по релевантности содержимого.

2.1.4 Электронная почта должна принимать письма на собственный домен и отправлять письма на различные домены.

2.1.5 Необходимо реализовать систему классификации писем, выделяющую наиболее важные письма для конкретного пользователя, а также фильтруя спам.

2.1.6 Необходимо реализовать систему менеджмента писем, позволяющую пользователю архивировать, удалять, помечать письма метками, добавлять различные папки и категории.

2.1.7 Предоставлять возможность просмотра ленты новостей.

2.1.8 Предоставлять возможность сортировки новостей по тегам, по географическим зонам, категориям (Политика, Технологии, Экономика, Общество, Происшествия, Спорт, Культура).

2.1.9 Предоставлять возможность поделиться статьей в популярных социальных сетях.

2.1.10 Предоставлять возможность комментирования прочитанных статей

2.1.11 Предоставлять возможность быстро перейти к прочтению похожих статей

2.1.12 Предоставлять возможность просмотра текущей погоды на период до 30 дней в зависимости от геолокации пользователя.

2.1.13 Предоставлять возможность просмотра текущего курса валют.

2.1.14 Предоставлять возможность изменения языка интерфейса (английский, русский)

2.1.15 Предоставлять возможность выводить виджеты различных внутренних сервисов на главную страницу сайта.

№ требования	Приоритетность	Трудоемкость человеко-час	Стабильность
2.1.1	10	300	Средняя
2.1.2	8	100	Низкая
2.1.3	10	100	Средняя
2.1.4	6	100	Средняя
2.1.5	8	20	Высокая
2.1.6	10	15	Высокая
2.1.7	4	10	Высокая
2.1.8	8	6	Высокая
2.1.9	5	2	Высокая
2.1.10	1	3	Средняя
2.1.11	2	6	Высокая
2.1.12	6	5	Высокая
2.1.13	4	4	Средняя
2.1.14	7	10	Средняя
2.1.15	7	5	Средняя

## 2.2 Usability (Требования к удобству использования)

2.2.1 Поиск персонализированной информации, согласно предпочтениям пользователя.

2.2.2 Просмотр новостей, согласно предпочтениям пользователя.

2.2.3 Современный интерфейс почты со всеми популярными интерфейсами.

## **2.3 Reliability (Требования к надежности)**

*2.3.1 Доступность система должна составлять не менее 99.97% (150 минут простоя в год).*

*2.3.2 В случае ошибки на одном из нескольких кластеров восстановление работоспособности системы без этого кластера должно составлять не более 5 минут, при этом полное восстановление работоспособности со всеми кластерами должно составлять не более 60 минут. В случае ошибки межкластерного взаимодействия на восстановление системы должно уйти не более 60 мин.*

*2.3.2 Количество критических ошибок в которых необходима остановка и перезапуск всей системы не должно превышать 15% от общего числа ошибок.*

## **2.4 Performance (Требования к производительности)**

*2.4.1 Среднее время ответа от сервера должно быть около 0.1с, при этом максимальное время ответа не должно превышать 0.3с.*

*2.4.2 Система должна иметь возможность обрабатывать до 1000 запросов в секунду.*

*2.4.3 Пиковая нагрузка на систему (1000 запросов в секунду) не должна загружать каждый CPU более чем на 80%.*

*2.4.4 Общий объем занимаемой памяти на каждом диске должен составлять не более 80% от максимального объема диска.*

## 2.5 Design Constraints (Ограничения разработки)

2.5.1 *Frontend технологии: Typescript, React, Redux + RTK query.*

2.5.2 *Backend технологии: Spring Boot (web MVC, Spring Data, Spring Security).*

2.5.3 *Взаимодействие Frontend и Backend должно осуществляться посредством REST API в формате JSON.*

2.5.4 *Используемая база данных: MySQL.*

2.5.5 *Использовать систему контроля версий: git, удаленный репозиторий: GitLab.*

*В процессе разработки использовать главную ветку main для хранения рабочей версии ПО, использовать ветку dev для разработки, для каждой отдельной фичи создавать новую ветку, которую после окончания разработки и тестирования фичи сливать в dev.*

2.5.6 *Рабочая версия на production должна выполняться внутри K8s-кластера.*

2.5.7 *Разработка должна осуществляться в соответствии с гибкой методологией: Agile.*

## 2.6 Interfaces (Интерфейсы)

2.6.1 User Interfaces (Пользовательские интерфейсы)

Пользовательский интерфейс предоставляется в виде HTML5 страницы со стилями CSS и языком клиентских сценариев JS. Рендер полученного гипертекста осуществляется браузером.

2.6.2 Hardware Interfaces (Аппаратные интерфейсы)

Аппаратные интерфейсы системы должны включать в сетевые интерфейсы взаимодействия.

2.6.3 Software Interfaces (Программные интерфейсы)

Предложение должно предоставлять API, осуществляющий все возможности графического интерфейса для других программ. Доступ к API должен осуществляться по специальному ключу, правила пользования API портала должны быть отражены в документации. Формата выходных данных: JSON.

2.6.4 Communications Interfaces (Сетевые интерфейсы)

Каждый кластер системы должен иметь сетевой интерфейс с пропускной способностью 1Gb/s, при этом общая пропускная способность сетевых интерфейсах должна составлять не менее 10GB/s.

## 2.7 Licensing Requirements (Требования к лицензированию)

Проприетарная лицензия: *Commercial License*.

## 3. Use cases (Пример прецедентов)

Прецедент: Нахождение конкретной новости
ID: 1
<b>Краткое описание:</b> Гость нуждается в конкретной новости
<b>Главный актер:</b> Гость
<b>Второстепенные актеры:</b> нет
<b>Предусловия:</b> Гость знает заголовок
<b>Основной поток:</b> <ol style="list-style-type: none"><li>1. Гость открывает поиск</li><li>2. Гость находит конкретную новость</li><li>3. Гость открывает ее и при желании делится ссылкой на страницу</li></ol>

Прецедент: Пользователь забыл пароль
ID: 2
<b>Краткое описание:</b> Пользователь забыл пароль и хочет его восстановить
<b>Главный актер:</b> Пользователь
<b>Второстепенные актеры:</b> нет
<b>Предусловия:</b> Пользователь забыл пароль
<b>Основной поток:</b> <ol style="list-style-type: none"><li>1. Пользователь нажимает “забыл пароль”</li><li>2. Пользователь указывает почту</li><li>3. Пользователь заходит на стороннюю почту, куда приходит ссылка для установки нового пароля</li><li>4. Пользователь устанавливает новый пароль</li><li>5. Пользователь авторизуется</li></ol>



Прецедент: Редактирование новости
<b>ID:</b> 3
<b>Краткое описание:</b> Редактор новостей вносит изменения в определенную новость
<b>Главный актер:</b> Редактор
<b>Второстепенные актеры:</b> нет
<b>Предусловия:</b> Редактор новостей имеет доступ к редактированию новости
<b>Основной поток:</b> <ol style="list-style-type: none"> <li>1. Редактор новостей открывает редакторскую панель</li> <li>2. Редактор новостей выбирает конкретную новость для редактирования</li> <li>3. Редактор новостей вносит необходимые изменения в новость</li> <li>4. Редактор новостей сохраняет изменения</li> </ol>

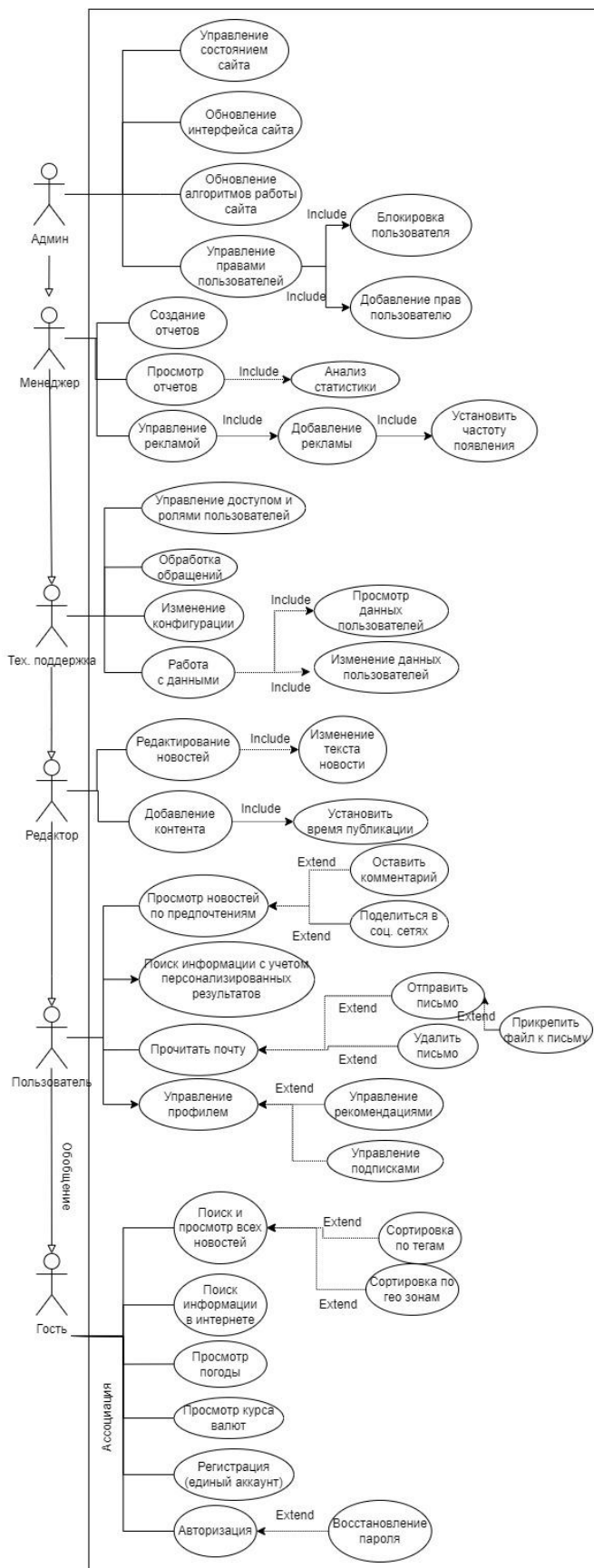
Прецедент: Обращение в техническую поддержку
<b>ID:</b> 4
<b>Краткое описание:</b> Пользователь обращается в техническую поддержку для решения проблемы
<b>Главный актер:</b> Пользователь
<b>Второстепенные актеры:</b> Тех. поддержка
<b>Предусловия:</b> Пользователь столкнулся с проблемой на сайте
<b>Основной поток:</b> <ol style="list-style-type: none"> <li>1. Пользователь находит раздел технической поддержки</li> <li>2. Пользователь описывает свою проблему</li> <li>3. Техническая поддержка анализирует проблему и предоставляет решение</li> </ol>

Прецедент: Модерация новостей
<b>ID:</b> 5
<b>Краткое описание:</b> Модератор проверяет и утверждает новости для публикации на сайте
<b>Главный актер:</b> Модератор
<b>Второстепенные актеры:</b> Нет
<b>Предусловия:</b> Новость подана на модерацию
<b>Основной поток:</b> <ol style="list-style-type: none"> <li>1. Модератор получает уведомление о новой новости, поданной на модерацию</li> <li>2. Модератор проверяет новость на соответствие правилам и стандартам сайта</li> <li>3. Модератор утверждает новость для публикации или отклоняет ее с указанием причины</li> </ol>

Прецедент: Управление пользователями
<b>ID:</b> 6
<b>Краткое описание:</b> Администратор управляет пользователями сайта
<b>Главный актер:</b> Администратор
<b>Второстепенные актеры:</b> Нет
<b>Предусловия:</b> Администратор имеет доступ к управлению пользователями
<b>Основной поток:</b> <ol style="list-style-type: none"> <li>1. Администратор открывает панель управления пользователями</li> <li>2. Администратор просматривает список пользователей и их активность</li> <li>3. Администратор имеет возможность блокировать или удалять пользователей, изменять их права доступа</li> </ol>

## 4. Use case diagramm(Диаграмма Use case)

Система



## Вывод

В процессе выполнения лабораторной работы, я приобрел навыки в составлении функциональных и нефункциональных требований, а также в определении их атрибутов. Я углубил свои знания в области анализа и документирования требований, что позволит мне лучше понимать потребности заказчика и создавать более эффективные решения.

Особое внимание было уделено построению Use-Case диаграмм на языке UML. Я изучил методику описания прецедентов, определил основных актеров в системе и выявил их взаимодействие с системой через различные сценарии использования. Этот опыт позволил мне лучше понять, как пользователи будут взаимодействовать с системой и как система будет отвечать на их запросы и действия.