

**Федеральное государственное автономное образовательное учреждение
высшего образования
Национальный исследовательский
университет ИТМО**

Факультет программной инженерии и компьютерной техники

Направление подготовки:

Системное и Прикладное Программное Обеспечение (09.03.04

Программная инженерия)

Дисциплина «Основы программной инженерии»

Отчет

По лабораторной работе №1

Вариант №140

Студент:

**Волков Алексей Иванович,
группа Р3213**

Практик:

Сорокин Роман Борисович

г. Санкт-Петербург, 2024 г.

Оглавление

Задание Выполнение

Introduction введение

Purpose цель

Document conventions описание технического сленга

Intended Audience and Reading Suggestions целевая аудитория Project scope масштаб проекта

References справки / ссылки

Overall Description общее описание Product

features особенности продукта Operating

environment рабочая среда

Functional requirements функциональные требования

Non functional requirements нефункциональные требования Usability

requirements требование к удобству использования Performance

requirements требование к производительности

Security & safety requirements требование безопасности & сохранности Software

quality attributes атрибуты качества программного обеспечения

External interface requirements требования к внешнему интерфейсу

UseCase-диаграмма

Прецедент 1: поиск нужного новости

Прецедент 2: загрузка своей новости

Вывод

Задание

Вариант №140: Собака.Ру: Люди. Мода. Город - <http://www.sobaka.ru/>

Составить список требований, предъявляемых к разрабатываемому веб-сайту (в соответствии с вариантом). Требования должны делиться на следующие категории:

- Функциональные.
 - Требования пользователей сайта.
 - Требования владельцев сайта.
- Нефункциональные.

Требования необходимо оформить в соответствии с шаблонами RUP (документ SRS - Software Requirements Specification). Для каждого из требований нужно указать его атрибуты (в соответствии с методологией RUP), а также оценить и аргументировать приблизительное количество часов, требующихся на реализацию этого требования.

Для функциональных требований нужно составить UML UseCase-диаграммы, описывающие реализующие их прецеденты использования.

Отчёт по лабораторной работе должен содержать:

1. Документ Software Requirements Specification, содержащий список требований к сайту.
2. UseCase-диаграммы прецедентов использования, реализующих функциональные требования.
3. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Методологии разработки ПО. Унифицированный процесс.
2. Требования и их категоризация. Атрибуты требований.
3. Язык UML.
4. Прецеденты использования. UseCase-диаграммы - состав, виды связей.

Выполнение

Introduction введение

Purpose цель

Sobaka.ru — это российский новостной портал, посвященный культуре, искусству, политике и обществу. Сайт предлагает читателям широкий спектр статей, рецензий, интервью и аналитических материалов, охватывающих различные аспекты современной жизни.

Document conventions описание технического сленга

Бекенд - это часть программного обеспечения, работающая на серверной стороне и обрабатывающая логику приложения, взаимодействуя с базой данных и обеспечивая данные для фронтенда.

Библиотека - коллекция функционала, добавляемая к готовому проекту для решения новых задач.

Фреймворк - набор инструментов, предоставляющих готовую архитектуру для построения приложений.

Фронтенд - это часть программного обеспечения, отвечающая за пользовательский интерфейс и взаимодействие с пользователем.

CSS - каскадные таблицы стилей, используемые для оформления и внешнего вида веб-страниц, созданных с помощью HTML.

Data Flow Diagram - графическое представление потоков данных

HTML - язык разметки, используемый для создания структуры веб-страниц.

JS - язык программирования, используемый для добавления интерактивности на веб-сайты.

Java — это мощный и универсальный язык программирования, который широко применяется в различных сферах разработки программного обеспечения. Он обладает богатым набором инструментов и возможностей для создания разнообразных приложений, начиная от веб-сервисов и мобильных приложений и заканчивая корпоративными системами и игровыми приложениями.

Spring — это мощный фреймворк для разработки Java-приложений, предоставляющий разработчикам удобные инструменты для создания масштабируемых, надежных и поддерживаемых приложений. Он предлагает множество готовых модулей и компонентов, упрощающих разработку, а также обеспечивает поддержку различных архитектурных шаблонов, таких как внедрение зависимостей и инверсия управления, что делает его популярным выбором среди Java-разработчиков.

PostgreSQL - система управления базами данных.

Software Requirements Specification - это документ, который описывает функциональные и нефункциональные требования к программному обеспечению. **UseCase** - сценарий использования.

Intended Audience and Reading Suggestions целевая аудитория

Студенты 1-4 курса бакалавриата технических ВУЗов.

Project scope масштаб проекта

Малый масштаб проекта: "...к малым проектам относятся проекты стоимостью до 10 млн. амер. долларов...".

References справки / ссылки

Use Case (сценарий использования) - <https://habr.com/ru/articles/699522/> Информация

о SRS - <https://habr.com/ru/articles/52681/>

Информация о RUP - <https://qaevolution.ru/metodologiya-menedzhment/rup/>

Классификация проектов - <https://clck.ru/38qJ3n>

Прецеденты - <https://clck.ru/38qekT>

Overall Description общее описание

Operating environment рабочая среда

- Для фронтенда используются такие технологии как: HTML, CSS, JS, VUE
- Для бекенда используется Java, Spring Framework, PostgreSQL, MinIO

Functional requirements функциональные требования

Для пользователя:

№	Требование
U1	Система должна предоставлять возможность выбрать один из четырех языков: русский, английский, немецкий, французский.
U2	Система должна предоставлять возможность находить контент с помощью поиска по различным критериям: категория, дата.
U3	Система должна предоставлять возможность просмотра новостей в зависимости от географического положения: Россия, Германия, Средняя Азия, Япония и т.д.
U4	Система должна предоставлять возможность быстрого поиска интересующего контента: Лучшее из, Хиты, Ключевые слова, Картинки, Мода, Светская Хроника, Образ жизни, Здоровье, Развлечения.
U5	Система должна предоставлять возможность просматривать ленту, конфигурируемую по предпочтениям (предпочтения сохраняются в сессии на основе тегов просмотренных новостей при помощи логического И).
U6	Система должна предоставлять возможность оставлять комментарии и ставить оценки новостям, чтобы обменяться мнениями и помочь другим пользователям в выборе контента.
U7	Система должна предоставлять возможность подписываться на определенные категории контента или на конкретных авторов и получать уведомления о новых материалах.
U8	Система должна предоставлять возможность загрузить свою новость.
U9	Система должна предоставлять возможность поделиться новостью с друзьями и коллегами.
U10	Система должна предоставлять возможность создание личного профиля.

Для администратора / владельца:

№	Требование
O1	Система должна предоставлять Администраторы должны иметь возможность добавлять, редактировать и удалять контент на сайте, а также управлять категориями и тегами для лучшей организации контента.
O2	Система должна предоставлять администраторы должны иметь возможность управлять пользователями, включая блокировку или удаление учетных записей, установку ограничений доступа и т.д.
O3	Система должна предоставлять владельцам возможность отслеживать статистику посещаемости, поведения пользователей, популярности контента и другие метрики с помощью аналитических инструментов и генерировать соответствующие отчет.

Non functional requirements нефункциональные требования

Usability requirements требование к удобству использования

N1	Система должна предоставлять отображение сайта с полностью работающим функционалом и без нарушения дизайна в следующих популярных браузерах: Chrome 121, Safari 16, Mozilla Firefox 122, Яндекс Браузер 23.
N2	Система должна предоставлять мобильную версию сайта должна корректно работать на современных разрешениях экранов, таких как: 1280x720, 1920x1080, 3840x2160.

Performance requirements требование к производительности

N3	Система должна предоставлять время загрузки главной страницы не должно превышать 5 секунд при скорости интернета не менее 20 Мбит/с.
N4	Сайт должен быть способен обрабатывать до 10000 одновременных пользовательских запросов.

Security & safety requirements требование безопасности & сохранности

N5	Система должна предоставлять контроль доступа к административным функциям сайта при помощи JWT и OAuth.
N6	Система должна пароли пользователей хранить в зашифрованном виде с использованием современных хеш-функций, таких как SHA-256.
N7	Система должна иметь механизм защиты от SQL-инъекций и DDoS-атак.
N8	Система должна предоставлять резервное копирование данных должно выполняться ежедневно с возможностью быстрого восстановления.

Software quality attributes атрибуты качества программного обеспечения

N9	Система должна вести логи ошибок и событий.
N10	Должна быть написана документация на английском языке ко всему программному обеспечению.
N11	Версии и зависимости используемых библиотек и фреймворков должны регулярно обновляться.

External interface requirements требования к внешнему интерфейсу

N12	Для взаимодействия фронтенда и бэкенда используется REST API.
N13	Ответы сервера предоставляются в формате JSON.

Номер требования	Статус	Кол-во часов	Стабильность
U1	Одобрено	6	Средняя
U2	Предложено	8	Средняя
U3	Одобрено	12	Низкая
U4	Предложено	6	Низкая
U5	Предложено	8	Высокая
U6	Отклонено	4	Высокая
U7	Предложено	2	Высокая
U8	Одобрено	16	Средняя
U9	Одобрено	16	Высокая
U10	Предложено	16	Высокая
O1	Одобрено	12	Высокая
O2	Одобрено	12	Средняя
O3	Одобрено	21	Высокая
N1	Одобрено	8	Высокая
N2	Одобрено	8	Высокая
N3	Отклонено	16	Средняя
N4	Предложено	16	Низкая
N5	Предложено	6	Высокая
N6	Одобрено	4	Высокая

N7	Предложено	8	Низкая
N8	Отклонено	4	Низкая
N9	Предложено	2	Низкая
N10	Отклонено	12	Средняя
N11	Отклонено	1	Низкая
N12	Одобрено	10	Высокая
N13	Одобрено	10	Высокая

Вычисление стоимости проекта:

Присвоим веса.

Статус:

Одобрено: 2 (высокий вес)

Предложено: 1.5 (средний вес)

Отклонено: 1 (низкий вес)

Стабильность:

Высокая: 1 (низкий вес)

Средняя: 1.5 (средний вес)

Низкая: 2 (высокий вес)

Трудозатраты = Количество часов * Вес статуса * Вес стабильности

U1: 18 часов

U2: 24 часа

U3: 24 часа

U4: 9 часов

U5: 24 часа

U6: 8 часов

U7: 6 часов

U8: 48 часов

U9: 32 часа

U10: 24 часа

O1: 48 часов

O2: 27 часов

O3: 84 часа

N1: 16 часов

N2: 16 часов

N3: 24 часа

N4: 48 часов

N5: 18 часов

N6: 8 часов

N7: 12 часов

N8: 4 часа

N9: 3 часа

N10: 18 часов

N11: 2 часа

N12: 20 часов

N13: 20 часов

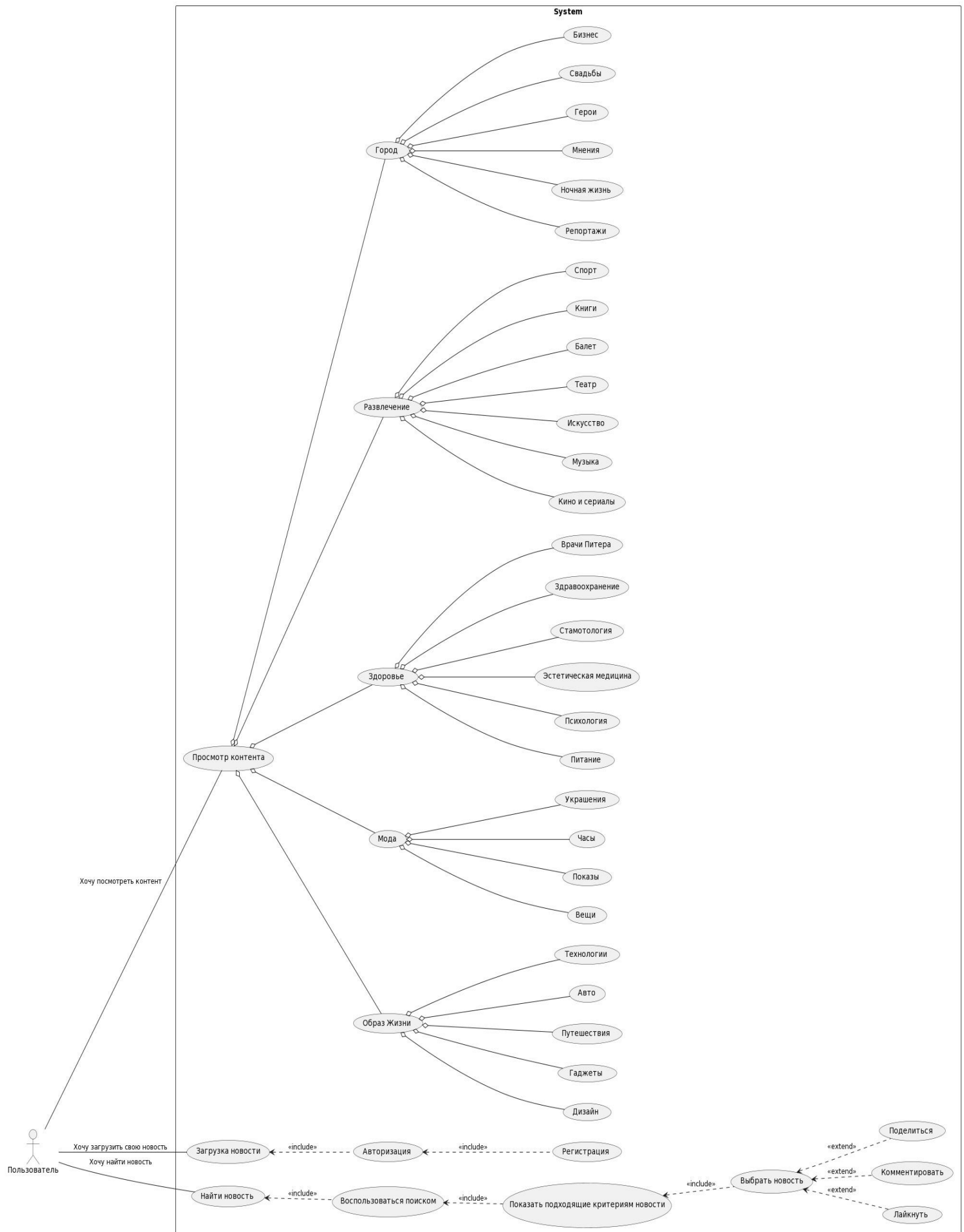
Общее кол-во: 513 часов

Теперь считаем общую стоимость умножая кол-во часов на среднюю зарплату за час

Допустим 50 долларов в час тратим

$513 * 50 = 25650$

UseCase-диаграмма



Прецедент 1: поиск нужной новости

Система	Новостной портал https://www.sobaka.ru
Основное действующее лицо	Пользователь
Цель	Поиск нужной новости
Триггер	Пользователь решает найти новость и вводит запрос в строку поиска
Результат	Результат поиска в виде выборки вместе с названиями и количеством просмотров

Основной поток событий

№	Действующее лицо	Шаг
1	Система	Запрашивает ввод критериев поиска: текстовый запрос, категорий
2	Пользователь	Вводит свой запрос
3	Система	Создает выборку новостей по популярности(на основе критериев)
4	Система	Отправляет пользователь превью новостей вместе с названием, количеством просмотров
5	Пользователь	Выбирает желаемую новость
6	Система	Открывает страницу просмотра с видео
7	Система	Загружает видео по его ID и добавляет +1 к числу просмотров
8	Пользователь	Просматривает видео
9	Пользователь	Лайкает / комментирует / делится / игнорит действие

Прецедент 2: загрузка своей новости

Система	Новостной портал https://www.sobaka.ru
Основное действующее лицо	Пользователь
Цель	Загрузка своей новости
Триггер	Пользователь решает загрузить свою новость
Результат	Новость загружена, теперь наш пользователь – автор

Основной поток событий

№	Действующее лицо	Шаг
1	Система	Запрашивает авторизацию пользователя
2	Пользователь	Авторизуется / Регистрируется
3	Система	Отправляет форму для названия и загрузки новости
4	Пользователь	Придумывает название и загружает новость
6	Система	Присваивает новости ID
7	Система	Сохраняет новость с ID и добавляет информацию о ней в базу данных
8	Система	Отправляет HTTP/1.1 200 OK
9	Пользователь	*чувство выполненного долга*

Вывод

В ходе данной лабораторной работы мы ознакомились с методологией RUP и структурой SRS документа, была создана UML UseCase-диаграмма и составлен список требований к сайту, оформленный в виде документа SRS.

Для создания UML был использован сервис plantuml.com;

Исходный код UML диаграммы:

```
@startuml
left to right direction
skinparam packageStyle rectangle
actor Пользователь as User
rectangle "System" {
    usecase "Просмотр контента" as ViewContent
    usecase "Образ Жизни" as Life
    usecase "Мода" as Fashion
    usecase "Здоровье" as Health
    usecase "Развлечение" as Fun
    usecase "Город" as City
    usecase "Загрузка новости" as UploadNews
    usecase "Авторизация" as Authenticate
    usecase "Регистрация" as Register
    usecase "Найти новость" as FindNews
    User -- ViewContent : Хочу посмотреть контент
    ViewContent o-- Life
    Life o-- (Дизайн)
    Life o-- (Гаджеты)
    Life o-- (Путешествия)
    Life o-- (Авто)
    Life o-- (Технологии)
    ViewContent o-- Health
    Health o-- (Питание)
    Health o-- (Психология)
    Health o-- (Эстетическая медицина)
    Health o-- (Стаматология)
    Health o-- (Здравоохранение)
    Health o-- (Врачи Питера)
    ViewContent o-- Fashion
    Fashion o-- (Вещи)
    Fashion o-- (Показы)
    Fashion o-- (Часы)
    Fashion o-- (Украшения)
    ViewContent o-- Fun
    Fun o-- (Кино и сериалы)
    Fun o-- (Музыка)
    Fun o-- (Искусство)
    Fun o-- (Театр)
    Fun o-- (Балет)
    Fun o-- (Книги)
    Fun o-- (Спорт)
    ViewContent o-- City
    City o-- (Репортажи)
    City o-- (Ночная жизнь)
    City o-- (Мнения)
    City o-- (Герои)
    City o-- (Свадьбы)
    City o-- (Бизнес)
    User -- UploadNews : Хочу загрузить свою новость
    UploadNews <.. Authenticate : <<include>>
    Authenticate <.. Register : <<include>>
    User -- FindNews : Хочу найти новость
}
```

```
FindNews <.. (Воспользоваться поиском) : <<include>>
(Воспользоваться поиском) <.. (Показать подходящие критериям новости) : <<include>>
(Показать подходящие критериям новости) <.. (Выбрать новость) : <<include>>
(Выбрать новость) <.. (Лайкнуть) : <<extend>>
(Выбрать новость) <.. (Комментировать) : <<extend>>
(Выбрать новость) <.. (Поделиться) : <<extend>>
}
@enduml
```