

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
ИТМО»**



ITMO UNIVERSITY

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №1
по дисциплине
«Основы программной инженерии»:

Вариант 323216
Выполнил:
Ястребов-Амирханов А.
ГРУППА: Р3232
ПРЕПОДАВАТЕЛЬ: Егошин А. В.

Санкт-Петербург,
2025

Оглавление

Текст задания.....	3
1. System features	4
2. Nonfunctional requirements	7
3. Use Case Diagram	11
4. Precedents	12
Вывод.....	20

Текст задания

Лабораторная работа #1

Вариант

Вариант №323216: 🍎 Кулинарные рецепты 🍔 1000.menu 🍽 - <https://1000.menu/>

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.

1. System features

1.1 Functional requirements

Требования пользователя сайта:

1.1.1 Система должна предоставлять возможность просмотра рецептов с подробными инструкциями, включающими список ингредиентов с точным количеством (граммы, миллилитры, штуки), пошаговое описание приготовления с таймерами, фотографии/видео каждого этапа (минимум 1 изображение на шаг, формат JPEG/PNG, до 5 МБ), Общее время приготовления в минутах.

1.1.2 Система должна предоставлять возможность просмотра рецептов по категориям: Завтраки, Обеды, Ужины, Десерты, Напитки, Закуски, Салаты, Первые блюда, Вторые блюда, Выпечка.

1.1.3 Система должна предоставлять возможность фильтрации рецептов по ингредиентам, сложности приготовления, времени приготовления, калорийности и категориям:

1. Стиль и качество:
 - *Домашний мишлен*
 - *От шеф-повара*
 - *Классические блюда*
 - *Старинные рецепты*
2. Техники и особенности приготовления:
 - *Способы приготовления*
 - *Быстро*
 - *Просто*
 - *Вкусно*
 - *Веселые и смешные рецепты*
3. Прием пищи и ситуации:
 - *Праздничный стол*
 - *На завтрак*
 - *На ужин*
4. Целевая аудитория и условия:
 - *Рецепты для детей*
 - *Рецепты на природу*
 - *Ограничения в питании*
 - *Используем кухонную утварь*
5. Тип блюда:
 - *Закуски*
 - *Напитки*
 - *Салаты*
 - *Тесто домашнее*
 - *Бутерброды*
 - *Молочные продукты домашние*
 - *Супы*
 - *Вторые блюда*
 - *Гарнир*
 - *Соусы*
 - *Десерты*
 - *Выпечка*
 - *Торты домашние*
 - *Для торта и выпечки*
 - *Заготовки на зиму*

Логика фильтрации:

Внутри каждой группы фильтров применяется логический оператор «ИЛИ». Это означает, что если пользователь выбирает несколько значений в рамках одной группы, рецепт будет отображен, если удовлетворяет хотя бы одному из выбранных значений.

Между различными группами фильтров применяется логический оператор «И».

1.1.4 Система должна предоставлять возможность просмотра рейтинга рецептов, рассчитанного на основе голосов пользователей (лайков и дизлайков). Рейтинг определяется следующим образом: если количество лайков превышает количество дизлайков, рецепт считается положительным; если количество дизлайков больше – отрицательным; при равном количестве голосов рейтинг считается нейтральным.

1.1.5 Система должна предоставлять возможность оставлять комментарии и отзывы к рецептам (только для авторизованных пользователей и администраторов), максимальная длина комментария — 500 символов, автоматическая фильтрация запрещенной лексики, модерация комментариев перед публикацией (см. п. 1.1.22).

1.1.6 Система должна предоставлять возможность авторизованным пользователям добавлять рецепты в избранное.

1.1.7 Система должна предоставлять возможность авторизованным просматривать и управлять (добавление, просмотр, удаление) списком избранных рецептов.

1.1.8 Система должна предоставлять возможность авторизованным пользователям делиться рецептами в социальных сетях (ВКонтакте, Meta, X, Одноклассники, Telegram). При нажатии на кнопку «Поделиться» система автоматически формирует стандартное сообщение, которое включает:

- Заголовок рецепта: название блюда.
- Краткое описание: "Попробуйте этот рецепт на 1000.menu – быстро, вкусно, просто!"
- Ссылка на страницу рецепта: для перехода непосредственно к рецепту.
- Рекомендуемые хештеги: #рецепты, #1000menu, #кухня.

Пользователь имеет возможность отредактировать сгенерированный текст перед публикацией, чтобы добавить личные комментарии или изменения.

1.1.9 Система должна предоставлять возможность для авторизованного пользователя скачивания рецептов в формате PDF. При нажатии на кнопку «Скачать PDF» формируется файл, содержащий следующие элементы:

Основное содержимое:

- Название рецепта: заголовок, выполненный на русском языке.
- Изображение блюда: (если имеется) фотография готового блюда.
- Описание рецепта: подробное описание, включающее список ингредиентов, пошаговую инструкцию по приготовлению, время и сложность приготовления, а также калорийность.
- Информация об авторе: имя автора рецепта, может сопровождаться изображением автора и/или датой публикации.

Формирование имени файла:

- Название PDF-файла должно формироваться по следующему шаблону: «Рецепт_[название рецепта]_[имя автора].pdf»
- При этом название рецепта и имя автора указываются на русском языке.

1.1.10 Система должна предоставлять авторизованным пользователям возможность добавлять рецепты, обязательные поля: название, категория (из списка 1.1.2), ингредиенты, шаги приготовления, медиафайлы: до 5 фото (JPEG/PNG, до 5 МБ), 1 видео (MP4, до 50 МБ), предпросмотр рецепта перед публикацией.

1.1.11 Система должна предоставлять возможность просмотра видеорецептов, прикрепленных к рецептам.

1.1.12 Система должна предоставлять регистрацию через email/социальные сети (минимум:

Google, ВК), авторизацию через логин/пароль, сохранение в профиле: истории просмотров, избранных рецептов.

1.1.13 Система должна автоматически генерировать список покупок, объединяя ингредиенты из выбранных пользователем рецептов, с возможностью ручного редактирования количества, группировки по категориям:

- Овощи
- Фрукты
- Молочные продукты
- Мясо и птица
- Рыба и морепродукты
- Хлеб и выпечка
- Крупы, макаронные изделия и бобовые
- Специи и приправы
- Соусы
- Сладости и десерты
- Прочие продукты (для ингредиентов, не попадающих в вышеперечисленные категории)

1.1.14 Система должна предоставлять возможность просмотра популярных рецептов дня, недели и месяца, сортируя их по количеству просмотров и лайков.

1.1.15 Система должна позволять авторизованным пользователям просматривать профили авторов и подписываться на них. При подписке пользователь:

- Добавляет автора в свой список подписок.
- Получает уведомления о новых рецептах через email и в личном кабинете.
- Может управлять подписками, изменения настройки уведомлений или отменяя подписку.

1.1.16 Система должна формировать меню на неделю/месяц на основе диетических предпочтений пользователя, с возможностью ручной замены блюд из предложенных вариантов, автоматической генерации списка покупок для меню

1.1.17 Система должна предоставлять возможность расчета калорийности и пищевой ценности рецептов.

1.1.18 Система должна показывать блок "Похожие рецепты" на странице рецепта, используя для рекомендаций: совпадение ключевых ингредиентов, категорий и рейтинга.

1.1.19 Система должна предоставлять возможность отправки уведомлений о новых рецептах, акциях и рекомендациях.

1.1.20 Система должна включать форму обратной связи с полями: Тема сообщения (выбор из данных вариантов: "Техническая проблема", "Сотрудничество", "Другое"), текстовое поле для сообщения (макс. 500 символов), возможность прикрепления файлов до 5 МБ

1.1.21 Система должна предоставлять возможность просмотра информации о сайте и его разработчиках. ()

Требования владельца сайта:

1.1.22 Система должна предоставлять администраторам возможность ручной проверки пользовательских рецептов перед их публикацией. При проверке администратор может:

- 1) Одобрить рецепт:
 - После одобрения рецепт публикуется на сайте и становится доступен всем пользователям.
- 2) Отменить рецепт:
 - Рецепт не публикуется.
 - Пользователю отправляется уведомление с указанием причины отказа и, при необходимости, рекомендациями по доработке рецепта от администратора.

1.1.23 Система должна предоставлять администраторам возможность редактировать любой рецепт (с сохранением истории изменений: автор правки, дата, внесённые изменения), перемещение рецепта в архив с возможностью восстановления в течение 30 дней, полного удаления (без восстановления) при явных нарушениях правил.

1.1.24 Система должна позволять администраторам блокировать пользователей временно (на срок от 1 до 30 дней) или перманентно

1.1.25 Система должна собирать метрики для каждого рецепта: просмотры, лайки/дизлайки, комментарии, время на странице, добавление в избранное, и формировать еженедельные отчёты с возможностью фильтрации по категориям, дате, рейтингу.

1.1.26 Система должна предоставлять администраторам возможность редактировать страницы «О сайте», «Контакты», «Политика конфиденциальности» и сохранять историю изменений (версии страниц с датой и автором правки).

1.1.27 Система должна позволять администраторам загружать баннеры в форматах JPEG/PNG/WebP (макс. размер: 5 МБ), настраивать расписание показа (дата/время, частота), указывать целевую аудиторию (геолокация, возраст, интересы), отслеживать статистику: показы, клики, автоматически деактивировать баннеры по истечении срока.

1.1.28 Система должна включать тикет-систему для обращений (техподдержка, жалобы, вопросы по контенту), назначение тикетов конкретным администраторам, шаблоны ответов (для частых вопросов) и ручное редактирование, автоматическое уведомление пользователя о получении и решении обращения, срок обработки: не более 48 часов, при превышении — передача старшему администратору.

1.1.29 Система должна отображать встроенный YouTube-плеер на странице рецепта, расположенный под основным контентом, с видео пояснением. При этом:

- Плеер должен загружаться через YouTube API с корректной авторизацией.
- В случае недоступности видео система отображает информативное сообщение с предложением перейти на страницу YouTube.

2. Nonfunctional requirements

2.1 Usability requirements

2.1.1. Поддержка (отсутствие горизонтальной прокрутки, сохранение функциональности элементов) всех разрешений экрана пользователя (для телефонов: 360x640, 375x667, 414x896, 360x780, 360x760, 375x812, 360x720, 414x736, 412x846, 412x892, для компьютеров: 1366x768, 1920x1080, 1536x864, 1440x900, 1280x720, 1600x900, 1280x800, 1280x1024, 1024x768, 768x1024).

2.1.2 Система должна обеспечивать адаптивный дизайн, гарантирующий корректное отображение (без горизонтальной прокрутки, сохранение функциональности элементов, соответствие макетам для разрешений из 2.1.1) на устройствах: ПК, планшеты, смартфоны, в браузерах Google Chrome (90+), Mozilla Firefox (88+), Microsoft Edge (90+), Яндекс.Браузер (22+) и их новых версиях, с поддержкой портретной/ландшафтной ориентации.

2.1.3 Система должна отображать сообщения об ошибках (текст + код ошибки + рекомендации) во всплывающих окнах или рядом с полями форм, с примерами: «Неверный email-формат. Пример: user@example.com»

2.2 Reliability requirements

2.2.1 Система должна обеспечивать высокий уровень доступности (не менее 99% времени безотказной работы) для стабильного доступа пользователей к функционалу сайта.

2.2.2 Система должна обеспечивать безопасность данных: HTTPS, соответствие PCI DSS для платежей, хранение паролей в хэшированном виде.

2.2.3 Система должна вести журналирование событий (ошибки уровня ERROR+, действия пользователей) в формате JSON с полями: timestamp, error_code, message, user_id, device; хранить логи 90 дней; автоматически оповещать администраторов при критических сбоях.

2.3 Performance requirements

2.3.1 Время полной загрузки главной страницы, страницы рецепта и профиля не должно превышать 3 секунд при скорости соединения 3G (1.5 Мбит/с).

2.3.2 Система должна выдерживать 1000 одновременных активных сессий (по 3 запроса/сек на пользователя) с сохранением времени ответа сервера ≤ 2 сек.

2.4 Licensing requirements

2.4.1 Все сторонние компоненты должны иметь лицензии категории MIT, Apache-2.0, BSD-3-Clause. Система должна интегрироваться с SCA-сканером (например, Snyk) для еженедельной проверки зависимостей. При обнаружении лицензий GPL, AGPL, BSL — блокировать сборку и уведомлять администратора.

Функциональные требования

№ требования	Приоритетность	Трудоемкость человеко-час	Стабильность
1.1.1	Must have	15	Высокая
1.1.2	Must have	10	Высокая
1.1.3	Must have	15	Высокая
1.1.4	Must have	20	Высокая
1.1.5	Must have	10	Средняя
1.1.6	Should have	15	Средняя
1.1.7	Could have	5	Средняя
1.1.8	Should have	10	Средняя
1.1.9	Should have	5	Средняя
1.1.10	Must have	10	Высокая
1.1.11	Should have	5	Средняя

1.1.12	Must have	20	Высокая
1.1.13	Should have	10	Средняя
1.1.14	Must have	10	Средняя
1.1.15	Should have	5	Средняя
1.1.16	Should have	25	Средняя
1.1.17	Must have	10	Средняя
1.1.18	Could have	10	Средняя
1.1.19	Could have	10	Низкая
1.1.20	Should have	7	Средняя
1.1.21	Could have	5	Низкая
1.1.22	Must have	10	Высокая
1.1.23	Must have	15	Высокая
1.1.24	Should have	5	Средняя
1.1.25	Should have	10	Средняя
1.1.26	Could have	10	Средняя
1.1.27	Could have	15	Низкая
1.1.28	Must have	15	Средняя
1.1.29	Must have	7	Высокая

Нефункциональные требования

Требование	Приоритетность	Стабильность
2.1.1	Should have	Средняя

2.1.2	Must have	Высокая
2.1.3	Must have	Средняя
2.2.1	Must have	Высокая
2.2.2	Must have	Высокая
2.2.3	Should have	Средняя
2.3.1	Must have	Высокая
2.3.2	Must have	Средняя
2.4.1	Should have	Высокая

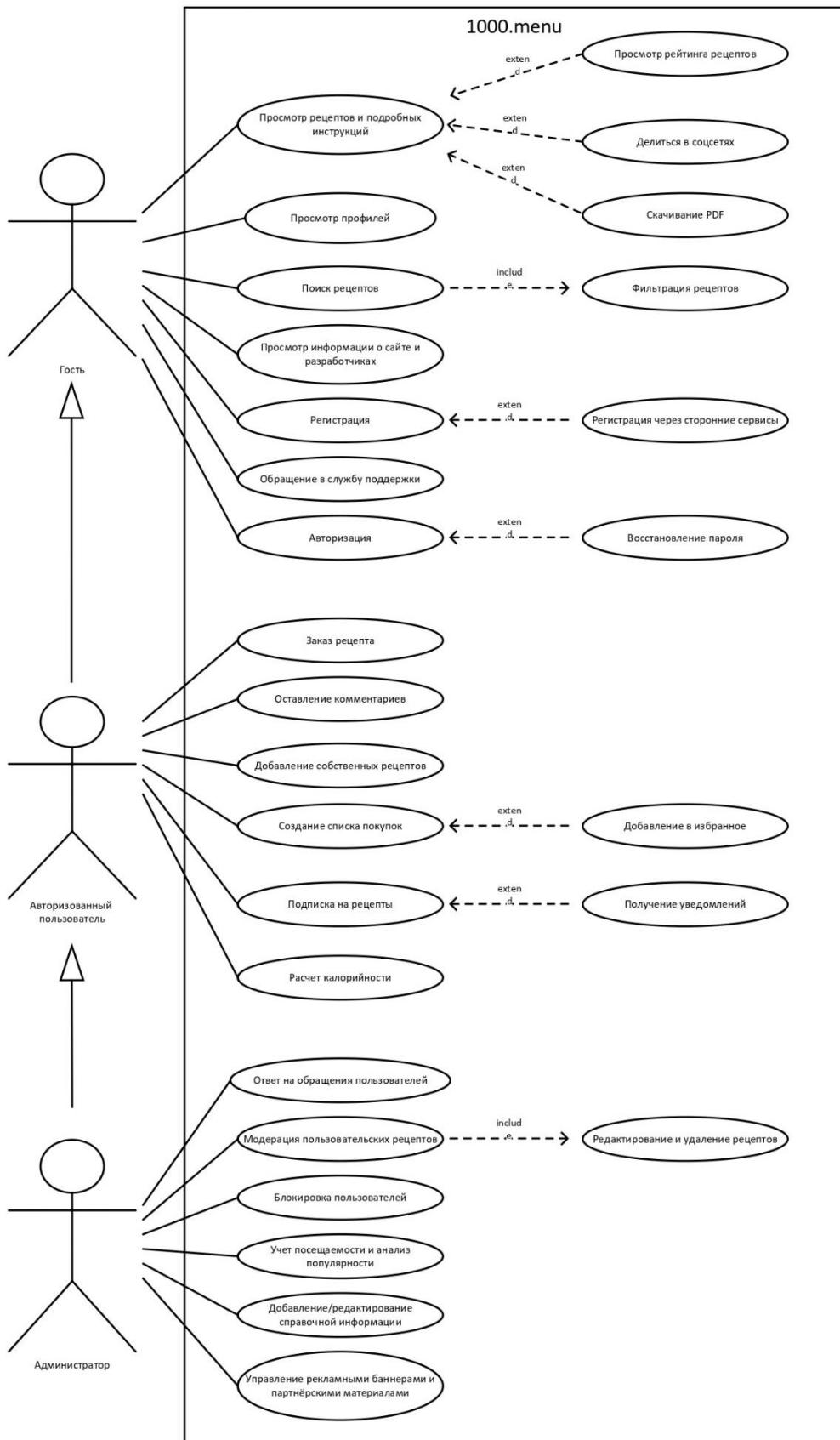
Риски

№	Риск	Тип	Вероятность	Масштаб потерь
1	Сложности интеграции с YouTube	Технический	Низкая	Средние*
2	Изменение руководства, с последующим изменением требований	Политический	Низкая	Средние**

* Из-за недостаточно подробной или устаревшей документации команда может столкнуться с трудностями в понимании работы YouTube IFrame API, что приведёт к значительным задержкам в реализации видеоплеера. + В процессе поиска обходных решений и экспериментов с интеграцией потребуется привлечение дополнительных ресурсов, что увеличит общий объём работ и может повлечь за собой перерасход бюджета.

** Новые требования могут привести к необходимости частичного или полного пересмотра уже разработанных решений, что увеличит время и ресурсы на доработку.

3. Use Case Diagram



4. Precedents

Прецедент: Просмотр рецептов
ID: 1
Краткое описание: Позволяет пользователю увидеть список рецептов и ознакомиться с подробными инструкциями.
Главный актер: Гость (также доступно Авторизованному пользователю)
Предусловие: В базе данных системы содержатся рецепты, отсортированные по категориям, и доступны для отображения.
<p>Основной поток:</p> <ol style="list-style-type: none">Пользователь выбирает категорию рецептов (например, "Завтраки" или "Десерты").Система отображает список рецептов в выбранной категории.Пользователь нажимает на конкретный рецепт для просмотра деталей.Система отображает страницу рецепта с описанием, ингредиентами, шагами приготовления и медиафайлами (фото/видео).Пользователь может просмотреть рейтинг рецепта и, если авторизован, добавить его в избранное.
<p>Альтернативный поток:</p> <p>После шага 1: Если пользователь не выбирает категорию, он может воспользоваться поиском.</p> <ol style="list-style-type: none">Пользователь вводит ключевое слово в строку поиска (например, "пирог").Система отображает рецепты, соответствующие запросу.Поток продолжается с шага 3 основного потока.
Постусловия: Счетчик просмотров выбранного рецепта в базе данных системы увеличен на 1.

Прецедент: Добавление рецепта
ID: 2
Краткое описание: Авторизованный пользователь создает и отправляет свой рецепт на сайт для последующей модерации.
Главный актер: Авторизованный пользователь
Предусловие: Система настроена для приема данных нового рецепта, включая поля формы и загрузку медиафайлов.
<p>Основной поток:</p> <ol style="list-style-type: none">Пользователь заполняет форму (название, категория, ингредиенты, шаги).Пользователь загружает медиафайлы.Пользователь нажимает "Предпросмотр".Система отображает предварительный вид рецепта.Пользователь нажимает "Опубликовать".Система сохраняет данные рецепта в базе данных.
<p>Альтернативный поток:</p> <p>После шага 3: Если обязательные поля заполнены некорректно или не полностью, система уведомляет пользователя.</p> <p>3a. Система выводит сообщение об ошибке с указанием проблемных полей.</p> <p>3b. Пользователь возвращается к форме, исправляет ошибки и повторяет шаг 3.</p>
Постусловия: В базе данных системы создан новый рецепт со статусом "на модерации".

Прецедент: Модерация пользовательских рецептов

ID: 3

Краткое описание: Администратор проверяет пользовательский рецепт и принимает решение о его публикации или отклонении.

Главный актер: Администратор

Предусловие: В базе данных системы имеются рецепты со статусом "на модерации".

Основной поток:

1. Администратор открывает список рецептов на модерации.
2. Администратор выбирает рецепт.
3. Система отображает данные рецепта.
4. Администратор проверяет рецепт.
5. Если рецепт одобрен, администратор нажимает "Одобрить".
6. Система обновляет статус рецепта.

Альтернативный поток:

После шага 5: Администратор решает отклонить рецепт.

5а. Если рецепт не одобрен, администратор нажимает "Отклонить" и указывает причину.

Постусловия: Статус рецепта в базе данных системы обновлен на "опубликован" или "отклонен".

Прецедент: Создание списка покупок

ID: 4

Краткое описание: Система формирует список покупок на основе выбранных пользователем рецептов.

Главный актер: Авторизованный пользователь

Предусловие: В базе данных системы содержатся рецепты с указанием ингредиентов, доступных для выбора пользователем.

Основной поток:

1. Пользователь выбирает один или несколько рецептов из списка избранного или просмотренных.
2. Пользователь нажимает "Создать список покупок".
3. Система объединяет ингредиенты из выбранных рецептов в единый список.
4. Система отображает список пользователю с возможностью редактирования (добавление/удаление ингредиентов).
5. Пользователь сохраняет изменения.
6. Система фиксирует обновленный список покупок.

Альтернативный поток:

После шага 3: Если в рецептах есть дублирующиеся ингредиенты:

3a. Система автоматически объединяет их в одну запись с суммированием количества.

3b. Поток продолжается с шага 4 основного потока.

Постусловия: В базе данных системы сохранен новый или обновленный список покупок, привязанный к профилю пользователя.

Прецедент: Регистрация через сторонние сервисы

ID: 5

Краткое описание: Система регистрирует нового пользователя через учетную запись стороннего сервиса (например, Google или ВКонтакте).

Главный актер: Гость

Предусловие: Система настроена для интеграции с API сторонних сервисов авторизации (Google, ВКонтакте).

Основной поток:

1. Пользователь нажимает кнопку "Зарегистрироваться через [сервис]".
2. Система перенаправляет пользователя на страницу авторизации выбранного сервиса.

3. Пользователь вводит учетные данные и подтверждает доступ.
4. Сторонний сервис возвращает системе данные пользователя (например, email, имя).
5. Система создает новый профиль пользователя на основе полученных данных.
6. Система перенаправляет пользователя на главную страницу как авторизованного пользователя.

Альтернативный поток:

После шага 3: Если пользователь отменяет авторизацию на стороне сервиса:

3a. Система получает сообщение об отказе.

3b. Система возвращает пользователя на страницу регистрации с уведомлением об ошибке.

Постусловия: В системе создан новый пользовательский профиль, связанный с учетной записью стороннего сервиса.

Прецедент: Управление рекламными баннерами

ID: 6

Краткое описание: Система позволяет администратору загружать и настраивать рекламные баннеры для отображения на сайте.

Главный актер: Администратор

Предусловие: Система имеет интерфейс управления баннерами и готова принимать файлы в форматах JPEG/PNG.

Основной поток:

1. Администратор переходит в раздел управления баннерами.
2. Администратор загружает файл баннера (до 5 МБ).
3. Администратор задает параметры показа: даты, время, частоту, целевую аудиторию (геолокация, возраст, интересы).
4. Система проверяет корректность загруженного файла и параметров.
5. Администратор подтверждает сохранение.

- | |
|---|
| 6. Система активирует баннер для отображения в соответствии с заданным расписанием. |
|---|

Альтернативный поток:

После шага 4: Если файл превышает допустимый размер или формат неверный:

4а. Система выводит сообщение об ошибке.

4б. Администратор загружает новый файл, и поток возвращается к шагу 2.

Постусловия: В системе сохранен новый баннер с установленными параметрами показа и статусом "активен".

Прецедент: Поиск рецептов

ID: 7

Краткое описание: Система предоставляет возможность поиска рецептов по ключевым словам с использованием фильтров.

Главный актер: Гость

Предусловие: В базе данных системы содержатся рецепты с метаданными (название, категория, ингредиенты), доступными для поиска.

Основной поток:

1. Пользователь вводит ключевое слово в строку поиска (например, "суп").
2. Пользователь выбирает дополнительные фильтры (например, категория, сложность, время приготовления).
3. Пользователь нажимает кнопку "Найти".
4. Система выполняет поиск по базе данных с учетом ключевого слова и фильтров.
5. Система отображает список соответствующих рецептов.

Альтернативный поток:

После шага 4: Если подходящих рецептов не найдено:

4а. Система возвращает сообщение "Рецепты не найдены".

4b. Пользователь изменяет запрос или фильтры и возвращается к шагу 1.

Постусловия: В базе данных системы зафиксирован запрос пользователя для статистики поисковых предпочтений.

Прецедент: Оставление комментария к рецепту

ID: 8

Краткое описание: Система позволяет авторизованному пользователю оставить комментарий к рецепту для последующей модерации.

Главный актер: Авторизованный пользователь

Второстепенный актер: Администратор

Предусловие: В базе данных системы доступен рецепт с возможностью добавления комментариев.

Основной поток:

1. Пользователь открывает страницу рецепта.
2. Пользователь вводит текст комментария (до 500 символов).
3. Пользователь нажимает "Отправить".
4. Система проверяет текст на наличие запрещенной лексики.
5. Система сохраняет комментарий и отправляет его на модерацию.

Альтернативный поток:

После шага 4: Если в тексте обнаружена запрещенная лексика:

- 4a. Система отклоняет комментарий и уведомляет пользователя о нарушении.
- 4b. Пользователь редактирует текст и возвращается к шагу 3.

Постусловия: В базе данных системы создан новый комментарий к рецепту

Прецедент: Блокировка пользователя

ID: 9

Краткое описание: Система позволяет администратору заблокировать пользователя временно или навсегда.

Главный актер: Администратор

Второстепенный актер: Авторизованный пользователь (объект блокировки)

Предусловие: В системе зарегистрированы пользователи с активными профилями.

Основной поток:

1. Администратор открывает список пользователей в панели управления.
2. Администратор выбирает пользователя для блокировки.
3. Администратор указывает тип блокировки (временная с указанием срока или перманентная).
4. Администратор подтверждает действие.
5. Система обновляет статус пользователя и ограничивает его доступ к функциям сайта.
6. Система отправляет уведомление пользователю о блокировке.

Альтернативный поток:

После шага 3: Если администратор выбирает временную блокировку:

- 3a. Система устанавливает дату окончания блокировки в базе данных.
3b. Поток продолжается с шага 4 основного потока.

Постусловия: В базе данных системы статус выбранного пользователя изменен на "заблокирован" (с указанием срока, если временная блокировка).

Вывод

В ходе выполнения работы я узнал про структуру SRS-документа, а также ознакомился с методологией RUP. Создал список требований, предъявляемых к веб-сайту, для функциональных требований составила UML Use Case - диаграммы, описывающие реализующие их прецеденты использования.