

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**  
**ITMO University**

**Лабораторные работы 4 - 7**

**По дисциплине** Проектирование инфокоммуникационных систем

**Тема работы** Анализ потребностей и возможностей создания программных систем

**Обучающийся** Морозова Яна Александровна

**Факультет** факультет инфокоммуникационных технологий

**Группа** К3320

**Направление подготовки** 11.03.02 Инфокоммуникационные технологии и системы связи

**Образовательная программа** Программирование в инфокоммуникационных системах

<b>Обучающийся</b>	<u>22.11.2024</u> (дата)	<u>                    </u> (подпись)	<u>Морозова Я.А.</u> (Ф.И.О.)
--------------------	-----------------------------	--	----------------------------------

<b>Руководитель</b>	<u>                    </u> (дата)	<u>                    </u> (подпись)	<u>Осипов Н.А.</u> (Ф.И.О.)
---------------------	---------------------------------------	--	--------------------------------

## СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ .....	3
2. ХОД РАБОТЫ .....	3
2.4 Создание документа-концепции .....	3
2.5 Разработка спецификации требований к программному обеспечению .....	7
2.6 Разработка технического задания .....	10
2.7 Реализация диаграммы взаимодействия .....	12
ВЫВОД .....	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	16

## **1. ЦЕЛЬ РАБОТЫ**

Для выбранного варианта инфокоммуникационной системы определить набор требований и спецификаций на создание системы, на основании которых будет организована реализация проекта.

## **2. ХОД РАБОТЫ**

### **2.4 Создание документа-концепции**

#### **Дополнительная спецификация:**

##### **1. Границы системы**

Входные данные: названия ингредиентов, вводимые пользователем; названия рецептов; оценки и отзывы пользователей; предпочтения пользователя.

Выходные данные: список подходящих рецептов; уведомления о новых рецептах; рекомендации на основе пользовательских предпочтений.

Основные ограничения: система не предоставляет пошагового видео приготовления; система работает только с текстовыми данными о рецептах и фотографиями.

##### **2. Безопасность**

Управление доступом: разделение пользователей по ролям (администратор, пользователь, гость); авторизация через логин и пароль с поддержкой двухфакторной аутентификации.

Защита данных: использование шифрования для хранения паролей; шифрование данных пользователей в базе данных; обеспечение защиты от SQL-инъекций и XSS-атак.

Конфиденциальность: соблюдение стандартов обработки пользовательских данных; ограничение доступа к данным только для авторизованных пользователей.

Логирование: Ведение журнала событий для отслеживания подозрительной активности.

### 3. Надёжность

Обеспечение доступности системы: система должна обеспечивать 99,9% времени безотказной работы; использование серверов с резервным копированием данных.

Обработка сбоев: встроенные механизмы для автоматического восстановления системы после сбоев; регулярное резервное копирование данных с возможностью восстановления.

Работа под нагрузкой: оптимизация запросов и кэширование данных для быстрого действия; обеспечение масштабируемости для обработки увеличенного числа пользователей.

Тестирование на отказ: регулярное тестирование системы на предмет ошибок и отказов.

### 4. Ограничения

Технические ограничения: система требует постоянного интернет-соединения для работы; максимальный объём базы данных ингредиентов ограничен производительностью серверов.

Функциональные ограничения: система не может предоставлять информацию о специфических диетах без их предварительного добавления в базу данных; поиск рецептов ограничен только заданной базой данных, загрузка новых рецептов возможна только администраторами.

Пользовательские ограничения: ограниченное количество сохранённых рецептов в профиле для бесплатной версии; уведомления и рекомендации доступны только для зарегистрированных пользователей.

Юридические ограничения: использование рецептов, защищённых авторскими правами, допускается только с разрешения автора; данные о предпочтениях пользователя могут храниться только с согласия пользователя.

## **Словарь терминов**

Рецепт – структурированное описание процесса приготовления блюда с указанием ингредиентов и шагов.

Ингредиенты – продукты или вещества, используемые для приготовления блюда.

Пользователь – физическое лицо, взаимодействующее с системой.

Рекомендация – персонализированное предложение рецептов на основе предпочтений и истории пользователя.

Избранное – список рецептов, сохранённый пользователем для быстрого доступа.

Оценка рецепта – числовое значение (1-5), выражающее мнение пользователя о рецепте.

Уведомления – сообщения, информирующие пользователя о новых функциях или подходящих рецептах.

Список покупок – генерируемый системой список ингредиентов, необходимых для приготовления блюд.

## **Документ-концепция («Видение»)**

### **1. Формулировка проблемы**

Современные пользователи часто сталкиваются с проблемами поиска рецептов, соответствующих их наличию ингредиентов или диетическим ограничениям. Имеющиеся решения часто предлагают ограниченный функционал или неудобный интерфейс, что затрудняет повседневное использование. Основные проблемы:

- трудности поиска рецептов по имеющимся ингредиентам;
- недостаток рекомендаций для персонализации;
- отсутствие возможности учитывать диетические предпочтения или аллергии;

- ограниченная интеграция с социальными платформами и возможностями сохранения рецептов.

## 2. Заинтересованные лица

Пользователи системы: домашние кулинары, готовящие по имеющимся ингредиентам; люди с диетическими ограничениями (веганы, аллергики и т.д.); любители кулинарии, желающие расширить свои знания.

Администраторы системы: ответственные за управление базой данных рецептов и ингредиентов.

Разработчики: внедряющие новые функции и поддерживающие систему.

Партнёры: поставщики рецептов, производители продуктов и рекламодатели.

## 3. Задачи уровня пользователя

- Поиск рецептов по имеющимся ингредиентам;
- сохранение рецептов для последующего использования;
- установка фильтров для поиска (например, вегетарианские блюда, блюда без глютена);
- просмотр пошаговых инструкций и рекомендаций;
- получение автоматических предложений рецептов на основе пользовательских предпочтений;
- создание списка покупок на основе выбранного рецепта.

## 4. Перспективы продукта

Интеграция с умными устройствами (умными холодильниками, голосовыми помощниками); развитие системы рекомендаций на базе искусственного интеллекта (AI) для создания персонализированного опыта; внедрение социального компонента (обмен рецептами, рейтинги, комментарии); многоязыковая поддержка для охвата глобальной аудитории.

## 5. Преимущества системы

Удобство: пользователи получают рецепты, основываясь на уже имеющихся ингредиентах.

Персонализация: учитываются предпочтения и диетические ограничения.

Интеграция: возможность синхронизации с социальными сетями и другими приложениями.

Экономия времени: быстрый поиск рецептов и создание списков покупок.

Расширение знаний: широкая база данных рецептов для любого уровня навыков.

## **2.5 Разработка спецификации требований к программному обеспечению**

### **1. Введение**

#### **1.1 Назначение**

Этот документ описывает требования к разработке системы поиска рецептов по ингредиентам. Основная цель системы — предоставить пользователям инструмент для быстрого и удобного поиска блюд на основе доступных продуктов, учёта их предпочтений и ограничения по диетам.

#### **1.2 Область действия**

Система предназначена для индивидуальных пользователей (домашних поваров, людей с диетическими потребностями) и администраторов, управляющих контентом. Разработка направлена на веб-платформу с адаптацией для мобильных устройств.

#### **1.3 Определения, акронимы и сокращения**

Рецепт – инструкция по приготовлению блюда.

Ингредиенты – продукты, необходимые для приготовления блюда.

Администратор – лицо, управляющее содержимым системы.

Пользователь - лицо, использующее систему для поиска и сохранения рецептов.

#### 1.4 Краткий обзор

Система поиска рецептов позволяет вводить список ингредиентов, выбирать рецепты, сохранять их, создавать списки покупок и делиться рецептами.

## 2. Полное описание

### 2.1 Перспектива изделия

Система заменяет устаревший ручной поиск рецептов, предоставляя более эффективный и персонализированный интерфейс. Используются автоматические фильтры по диетам, аллергиям и другим параметрам.

### 2.2 Функции изделия

- Поиск рецептов по ингредиентам.
- Фильтрация по предпочтениям.
- Сохранение рецептов и составление списков покупок.
- Персонализация рекомендаций.
- Уведомления о новых рецептах и обновлениях.

### 2.3 Характеристики пользователя

Обычные пользователи: ищут рецепты по ингредиентам, фильтруют результаты и сохраняют их.

Администраторы: управляют базой данных рецептов, обновляют контент и контролируют пользователей.

### 2.5 Ограничения

Максимальная нагрузка: 10 000 одновременных пользователей.

Поддержка интерфейсов на русском и английском языках.

Только рецепты, проверенные администраторами, добавляются в базу.



## 2.6 Допущения и зависимости

Интернет-соединение требуется для всех функций.

Используется внешняя база данных рецептов.

Система должна быть совместима с популярными браузерами и ОС.

## 3. Специфические требования

### 3.1 Функциональные требования

Поиск рецептов: система должна предоставлять возможность поиска рецептов по введённым ингредиентам.

Сохранение рецептов: система позволяет сохранять рецепты в личном кабинете пользователя.

Создание списков покупок: автоматическая генерация списка покупок для выбранных рецептов.

Фильтрация: пользователи могут фильтровать рецепты по диетическим предпочтениям, времени приготовления и рейтингу.

### 1.2 Нефункциональные требования

Производительность: среднее время отклика системы — до 2 секунд.

Безопасность: все пользовательские данные должны быть зашифрованы (AES-256).

Надёжность: гарантированное время доступности системы — 99,9%.

### 1.3 Ограничения

Поддержка платформ: iOS, Android, веб. Локализация доступна только на двух языках.

### 1.4 Взаимодействие с внешними системами

Использование API для получения новых рецептов из партнёрских баз данных. Интеграция с почтовыми сервисами для уведомлений.

### 1.5 Дополнительные характеристики

Масштабируемость: возможность расширения базы данных рецептов.

Интерфейс: должен быть адаптирован для пользователей с ограниченными возможностями.

## **2.6 Разработка технического задания**

### **1. Назначение и цели создания системы**

#### **1.1 Назначение системы**

Система предназначена для автоматизации процесса поиска рецептов на основе доступных ингредиентов и предоставления рекомендаций, учитывающих предпочтения пользователей, диетические ограничения и аллергенные факторы.

#### **1.2 Цели создания системы**

Повышение удобства поиска и использования рецептов.

Ускорение процесса планирования блюд.

Обеспечение рекомендаций на основе пользовательских предпочтений.

Расширение доступа пользователей к проверенным и адаптированным рецептам.

### **2. Характеристика объектов автоматизации**

Объектами автоматизации являются:

Пользователи системы (обычные пользователи и администраторы).

База данных рецептов, включающая ингредиенты, инструкции и изображения.

Процессы фильтрации, сортировки и рекомендаций.

### **3. Требования к системе**

#### **3.1 Требования к системе в целом**

Система должна быть доступна через веб-интерфейс и мобильные приложения (iOS и Android).

Поддержка одновременной работы до 10 000 пользователей.

Уровень доступности системы — не менее 99,9%.

Безопасное хранение данных пользователей.

### 3.2 Требования к функциям, выполняемым системой

Поиск рецептов на основе списка введенных ингредиентов.

Фильтрация по времени приготовления, сложности и диетическим предпочтениям.

Сохранение рецептов в личном кабинете.

Создание списка покупок.

Уведомления о новых рецептах и рекомендациях.

### 3.3 Требования к видам обеспечения

Программное обеспечение: Реализация на платформе с открытым исходным кодом.

Техническое обеспечение: Серверная архитектура с базой данных.

Информационное обеспечение: Список рецептов, соответствующих требованиям по качеству и лицензиям.

## 2. Состав и содержание работ по созданию системы

1 Анализ требований заказчика.

2 Проектирование архитектуры системы.

3 Разработка пользовательского интерфейса.

4 Создание и настройка базы данных.

5 Реализация функциональных модулей системы.

6 Тестирование системы (модульное, интеграционное, нагрузочное).

7 Внедрение системы.

8 Обучение администраторов.

## 3. Порядок контроля и приёмки системы

Проведение контрольных испытаний по сценарию заказчика.

Тестирование функциональности согласно требованиям.

Приёмочные испытания с привлечением конечных пользователей.

Утверждение результатов заказчиком на основании согласованных критериев качества.

#### **4. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**

Установка серверного оборудования и программного обеспечения.

Настройка доступа пользователей к системе.

Импорт базы данных рецептов.

#### **5. Требования к документированию**

Руководство пользователя.

Руководство администратора.

Отчёты о тестировании.

Полная документация по программному интерфейсу и архитектуре системы.

#### **6. Источники разработки**

Техническое задание, утверждённое заказчиком.

Нормативные документы по безопасности данных.

Стандарты разработки ПО.

Библиотеки и фреймворки, используемые для разработки.

### **2.7 Реализация диаграммы взаимодействия**

Разработаны диаграммы последовательности и взаимодействия<sup>1</sup> для варианта использования системы «Поиск рецепта по ингредиентам» (Рисунок 2)

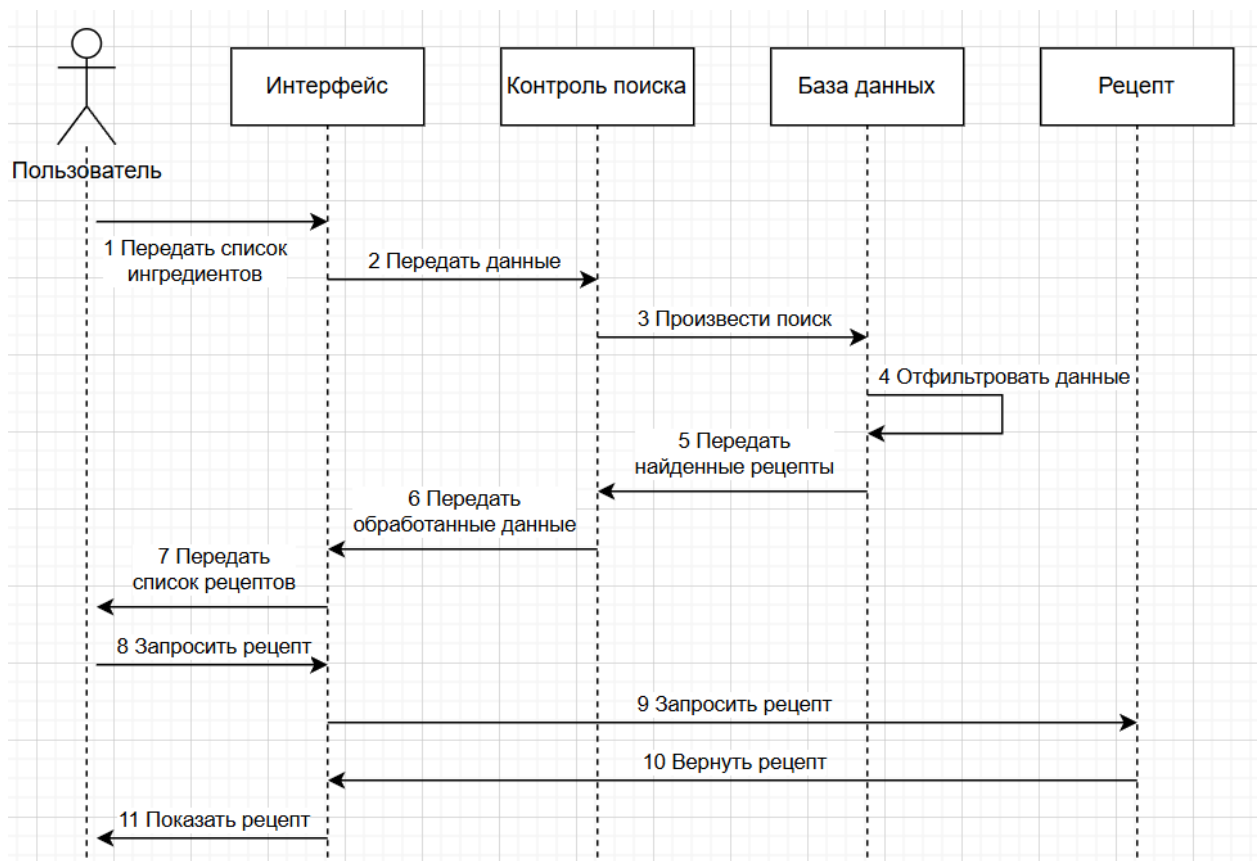


Рисунок 2 – Диаграмма последовательности

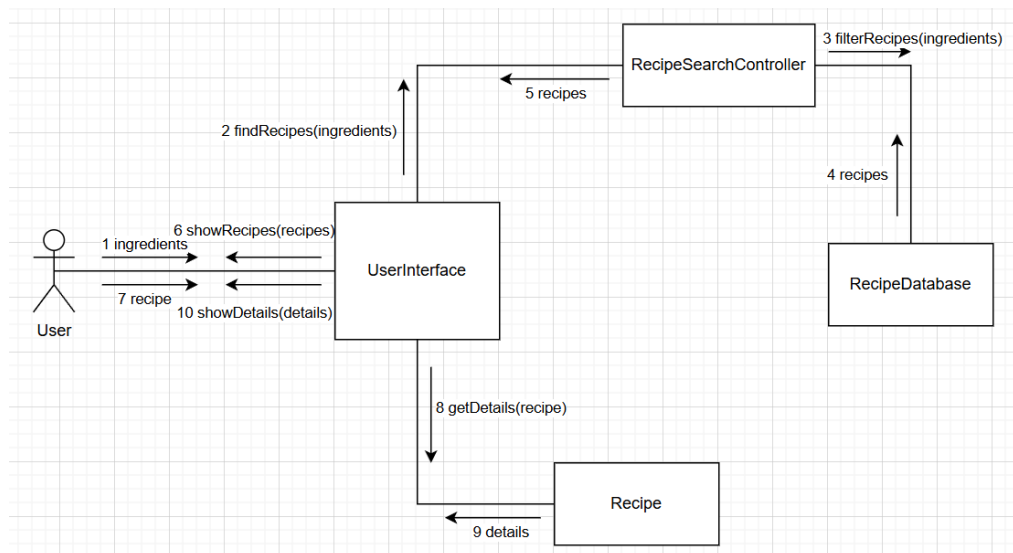


Рисунок 3 – Диаграмма взаимодействий

Для проектирования классов использованы шаблоны GRASP, чтобы обеспечить структурированное распределение обязанностей. Основные примененные шаблоны:

### 1 Information Expert:

Обязанности по обработке данных возложены на те классы, которые содержат или владеют необходимой информацией. Например, RecipeDatabase отвечает за фильтрацию рецептов, так как содержит информацию о них.

### 2 Controllle:

Контроллер (RecipeSearchController) управляет процессом поиска рецептов, получая команды от интерфейса пользователя и передавая запросы к базе данных.

### 3 Low Coupling:

Разделение обязанностей между UserInterface, RecipeSearchController и RecipeDatabase минимизирует зависимость между ними, что облегчает тестирование и модификацию. Например, пользовательский интерфейс не взаимодействует напрямую с базой данных.

### 4 High Cohesion:

Обязанности внутри каждого класса логически связаны. Например, Recipe отвечает за предоставление деталей одного рецепта, а не за поиск или взаимодействие с пользователем.

### 5 Creator:

Ответственность за создание объектов делегирована тому классу, который использует или агрегирует их. Например, RecipeDatabase создает объекты Recipe.

Использование шаблонов GRASP позволяет повысить гибкость системы: легко добавлять новые функции, такие как сортировка рецептов или учет новых типов данных; обеспечить модульность: логика поиска отделена от интерфейса пользователя, что улучшает читаемость и поддержку кода; облегчить тестирование: каждый класс можно тестировать отдельно

благодаря низкой связанности; ускорить разработку: распределение обязанностей между классами способствует параллельной работе разработчиков.

## **ВЫВОД**

В ходе выполнения лабораторной работы были определены наборы требований и спецификаций на создание системы, на основании которых будет организована реализация проекта

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Diagrams: официальный сайт. — Санкт-Петербург. — URL: <https://www.diagrams.net> (дата обращения: 01.11.2022)