МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра программных систем

**ОТЧЕТ ПО ПРАКТИКЕ**

Вид практики преддипломная

Тип практики преддипломная

(в соответствии с ОПОП ВО)

Сроки прохождения практики: с 17.05.2025 по 30.05.2025

(в соответствии с календарным учебным графиком)

по направлению подготовки 02.03.02   
«Фундаментальная информатика и информационные технологии

(уровень бакалавриата)

направленность (профиль) «Информационные технологии»

Обучающийся группы № 6401-020302D А.А. Алёнушка

Руководитель практики,

Доцент кафедры программных систем,

к.т.н, доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.А. Гордеева

Дата сдачи 30.05.2025

Дата защиты 30.05.2025

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2025

Содержание

[Задания по практике для выполнения определенных видов работ, связанных с будущей профессиональной деятельностью (сбор и анализ данных и материалов, проведение исследований) 3](#_Toc198654351)

[Введение 9](#_Toc198654352)

[1 Проект системы 10](#_Toc198654353)

[1.1 Структурная схема системы 10](#_Toc198654354)

[1.2 Диаграмма вариантов использования 12](#_Toc198654355)

[1.3 Диаграмма деятельности 12](#_Toc198654356)

[1.4 Диаграмма последовательности 14](#_Toc198654357)

[2 Руководство пользователя 16](#_Toc198654358)

[2.1 Назначение системы 16](#_Toc198654359)

[2.2 Условия работы системы 16](#_Toc198654360)

[2.3 Установка системы 16](#_Toc198654361)

[2.4 Работа с системой 17](#_Toc198654362)

[2.4.1 Начало работы: регистрация и вход 17](#_Toc198654363)

[2.4.2 Работа с коллекциями карточек 18](#_Toc198654364)

[2.4.3 Добавление и редактирование карточек 19](#_Toc198654365)

[2.4.4 Редактирование коллекций и поиск карточек 19](#_Toc198654366)

[2.4.5 Повторение карточек 20](#_Toc198654367)

[ЗАКЛЮЧЕНИЕ 22](#_Toc198654368)

[Список использованных источников 23](#_Toc198654369)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра программных систем

Задания по практике для выполнения определенных видов работ, связанных с будущей профессиональной деятельностью   
(сбор и анализ данных и материалов, проведение исследований)

Обучающемуся Алёнушка Алесандру Александровичу группы 6401-020302D

Направлен на практику приказом по университету от 24.04.2025 г. № 224-ПР

на кафедру программных систем

(наименование профильной организации или структурного подразделения университета)

Тема: Веб-приложение для обучения по системе Лейтнера с реализацией алгоритма интервального повторения

|  |  |  |
| --- | --- | --- |
| Планируемые результаты освоения образовательной программы  (компетенции) | Выполнение определенных видов работ, связанных с будущей профессиональной деятельностью  (сбор и анализ данных и материалов, проведение исследований) | Результаты  практики |
| **ОПК-1**. Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности.  **ОПК-1.1.** Знает основные положения и концепции в области математических и естественных наук, Базовые теории и истории основного, теории коммуникации; знает основную терминологию.  **ОПК-1.2.** Умеет осуществлять первичный сбор и анализ материала, интерпретировать различные математические объекты.  **ОПК-1.3.** Имеет практический опыт работы с решением стандартных математических задач и применяет его в профессиональной деятельности. | Разработать проект системы.  Ознакомиться со стандартом оформления  текстовых учебных документов  Самарского университета.  Оформить отчет по результатам прохождения  практики в строгом соответствии со  стандартом оформления  текстовых учебных документов. | Разработан проект системы.  Подготовлен отчет по практике. |
| **ОПК-2.** Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности.  **ОПК-2.1.** Использует основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, основную терминологию, знаком с содержанием Единого Реестра Российских программ.  **ОПК-2.2.** Анализирует код на типовых языках программирования, может составлять программы.  **ОПК-2.3.** Применяет опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникаций. | Проанализировать возможности Spring, Hibernate, JPA, GraphQL, PostgreSQL, Docker, React, Liquibase и Git для разработки современных веб-приложений. | Изучены и освоены следующие программные средства: Spring, Hibernate, JPA, GraphQL, PostgreSQL, Docker, React, Liquibase и Git |
| **ОПК-3.** Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям.  **ОПК-3.1.** Понимает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей.  **ОПК-3.2.** Соотносит знания в области программирования, интерпретирует прочитанное, определяет и создает информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем.  **ОПК-3.3.** Имеет практический опыт применения разработки программного обеспечения. | Разработать клиентскую и серверную часть веб-приложения, провести тестирование. | Разработаны клиентская и серверная часть веб-приложения, проведено тестирование. |
| **ОПК-4.** Способен участвовать в разработке технической документации программных продуктов и комплексов с использованием стандартов, норм и правил, а также в управлении проектами создания информационных систем на стадиях жизненного цикла.  **ОПК-4.1.** Использует принципы сбора и анализа информации, создания информационных систем на стадиях жизненного цикла.  **ОПК-4.2.** Осуществляет управление проектами информационных систем.  **ОПК-4.3.** Демонстрирует практический опыт анализа и интерпретации информационных систем. | Написать руководство пользователя. | Написано руководство пользователя. |
| **ОПК-5.** Способен инсталлировать и сопровождать программное обеспечение информационных систем и баз данных, в том числе отечественного происхождения, с учетом информационной безопасности.  **ОПК-5.1.** Понимает методику установки и администрирования информационных систем и баз данных. Знаком с содержанием Единого реестра российских программ.  **ОПК-5.2.** Реализует техническое сопровождение информационных систем и баз данных.  **ОПК-5.3.** Использует практические навыки установки и инсталляции программных комплексов, применения основ сетевых технологий. | Проанализировать технологии разработки технической спецификаций программных компонентов и их взаимодействия. | Взаимодействие клиентской и серверной части веб-приложения полностью настроено. |
| **ОПК-6.** Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.  **ОПК-6.1.** Понимает основные положения, концепции и современные методы обработки и хранения данных.  **ОПК-6.2.** Осуществляет первичный сбор и анализ данных для организации информационных процессов.  **ОПК-6.3.** Обладает практическим опытом применения современных информационных технологий для решения задач профессиональной деятельности. | Изучить и отобрать передовые методы в сфере проектирования и разработки веб-приложений | Учитывая все поставленные задачи, было спроектировано, реализовано и протестировано веб-приложение для обучения по системе Лейтнера с реализацией алгоритма интервального повторения. |

Дата выдачи задания 17.05.2025.

Срок представления на кафедру отчета о практике 30.05.2025.

Руководитель практики,

доцент кафедры программных систем, к.т.н., доцент\_\_\_\_\_\_\_\_\_\_ О.А. Гордеева

*(подпись)*

Задание принял к исполнению

обучающийся группы № 6401-020302D\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Алёнушка

*(подпись)*

Введение

Этап проектирования служит связующим звеном между анализом требований и непосредственной реализацией системы. В ходе проектирования формализуются и структурируются результаты предварительного исследования: разрабатываются архитектурные решения, описываются компоненты и их взаимодействие, выбираются нотации и инструменты моделирования.

Главная цель данного этапа — получить полноту и непротиворечивость описания будущего программного продукта, что позволяет минимизировать риски при разработке и обеспечить согласованность работы всех участников проекта. В результате проектирования создаются диаграммы классов, компонентов, последовательностей и других видов, а также схемы данных, которые станут основой для последующей реализации, тестирования и поддержки системы.

Во время преддипломной практики необходимо решить следующие задачи:

* разработать проект системы;
* разработать клиентскую и серверную часть веб-приложения и провести тестирование;
* составить диаграммы, описывающие систему в нотации UML;
* написать руководство пользователя;
* подготовить и оформить письменный отчет по практике.

1. Проект системы
   1. Структурная схема системы

Структурная схема информационной системы представляет собой графическое отображение её компонентов и взаимосвязей между ними [1]. Она служит инструментом для визуализации архитектуры системы, облегчая понимание её структуры и функционирования. Согласно определению, структурная схема — это совокупность элементарных звеньев объекта и связей между ними, один из видов графической модели.

В контексте проектирования информационных систем, структурная схема помогает разработчикам и аналитикам определить основные компоненты системы, их функции и способы взаимодействия. Это особенно важно при разработке сложных систем, где необходимо обеспечить согласованную работу различных подсистем и модулей. Использование структурных схем способствует выявлению потенциальных узких мест и оптимизации процессов внутри системы.

На рисунке 1 приведена структурная схема разрабатываемой системы, в ее состав входят клиентское и серверное приложения, которые взаимодействуют между собой с помощью протокола HTTP и технологии GraphQL.

В состав клиентского приложения входят следующие подсистемы:

* подсистема клиентского взаимодействия, обеспечивающая пользователям возможность просматривать интерфейс, взаимодействовать с элементами управления и перемещаться между страницами приложения;
* подсистема передачи данных, отвечающая за коммуникацию между клиентской и серверной частями приложения;

В состав серверного приложения входят следующие подсистемы:

* подсистема аутентификации и авторизации, реализующая процессы входа, регистрации пользователей и управления доступом к функционалу системы;

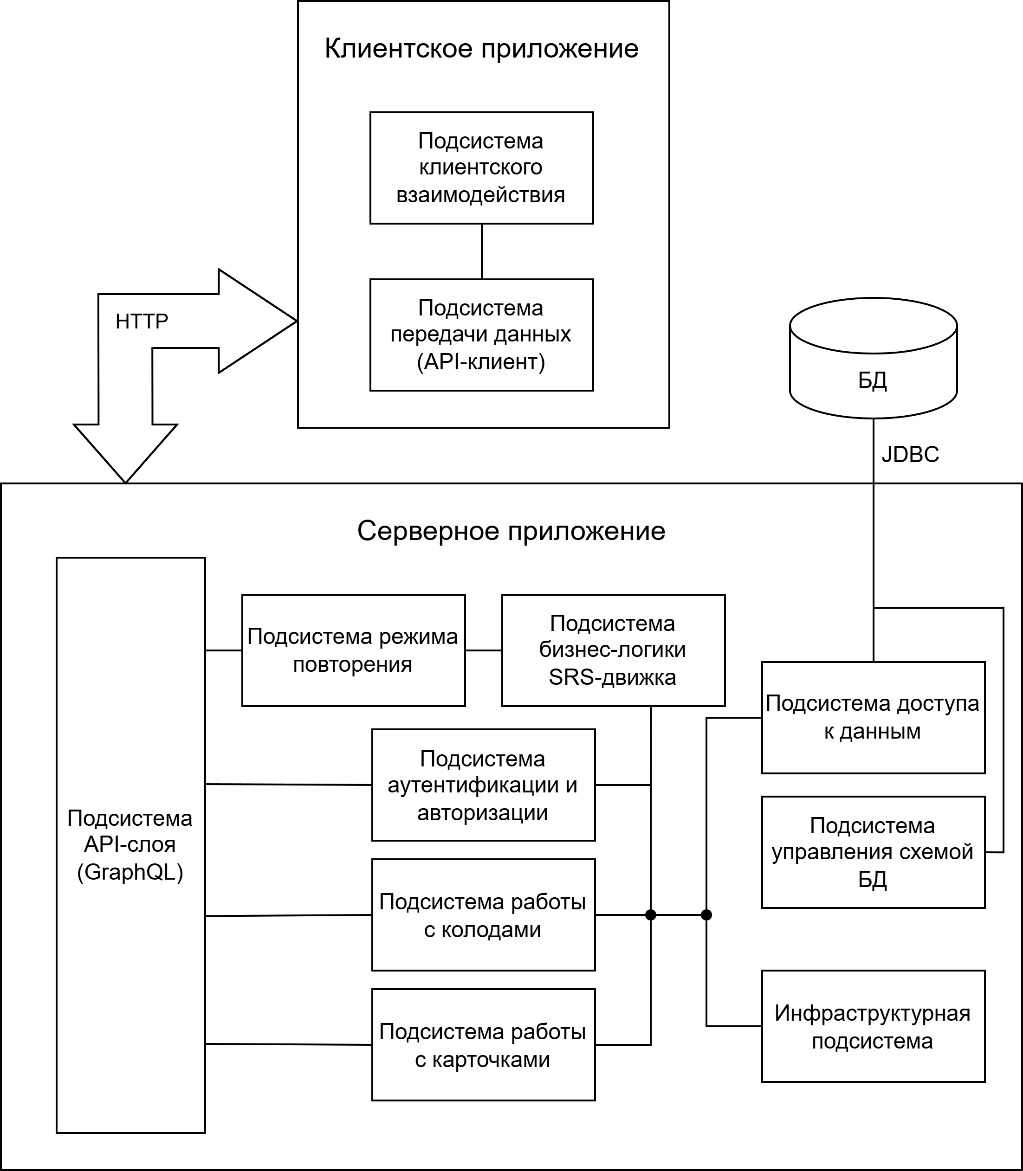


Рисунок 1 – Структурная схема системы

* подсистема бизнес-логики SRS-движка, которая обеспечивает реализацию алгоритма интервального повторения и управление процессом повторения карточек;
* подсистема обработки пользовательских запросов, принимающая и обрабатывающая запросы пользователей, и предоставляющая необходимые данные клиентской части;
* подсистема режима повторения, позволяющая пользователям повторять и закреплять изученный материал в соответствии с алгоритмом интервального повторения;
* подсистема управления карточками, предоставляющая пользователям возможность создавать, редактировать и удалять обучающие карточки;
* подсистема управления колодами, позволяющая объединять карточки в тематические группы для удобного изучения и организации материала;
* подсистема хранения и обработки данных, отвечающая за сохранение и предоставление информации, используемой в приложении;
* подсистема управления схемой базы данных, обеспечивающая инициализацию и поддержку актуальности структуры базы данных;
* инфраструктурная подсистема, отвечающая за базовые настройки и конфигурацию среды выполнения приложения.
  1. Диаграмма вариантов использования

Для иллюстрации функциональных требований к системе на концептуальном уровне используется диаграмма вариантов использования, позволяющая наглядно отразить ключевые сценарии взаимодействия пользователей и компонентов приложения.

Диаграмма вариантов использования (use case diagram) — это поведенческая диаграмма в языке UML, отображающая отношения между акторами (пользователями или внешними системами) и прецедентами (вариантами использования), что позволяет описать функциональные требования системы на концептуальном уровне [3]. Такая диаграмма показывает, какие сервисы предоставляет система и каким образом они используются внешними сущностями, не вдаваясь в детализацию внутренней реализации [4].

На рисунке 2 представлена диаграмма прецедентов (use case), демонстрирующая виды взаимодействия акторов с Java-приложением и определяющая основные варианты использования.

* 1. Диаграмма деятельности

Диаграмма активностей является одним из ключевых инструментов визуального моделирования поведения системы в нотации UML (Unified Modeling Language). Она используется для описания потоков управления и деятельности в системе, демонстрируя, какие действия выполняются и в каком

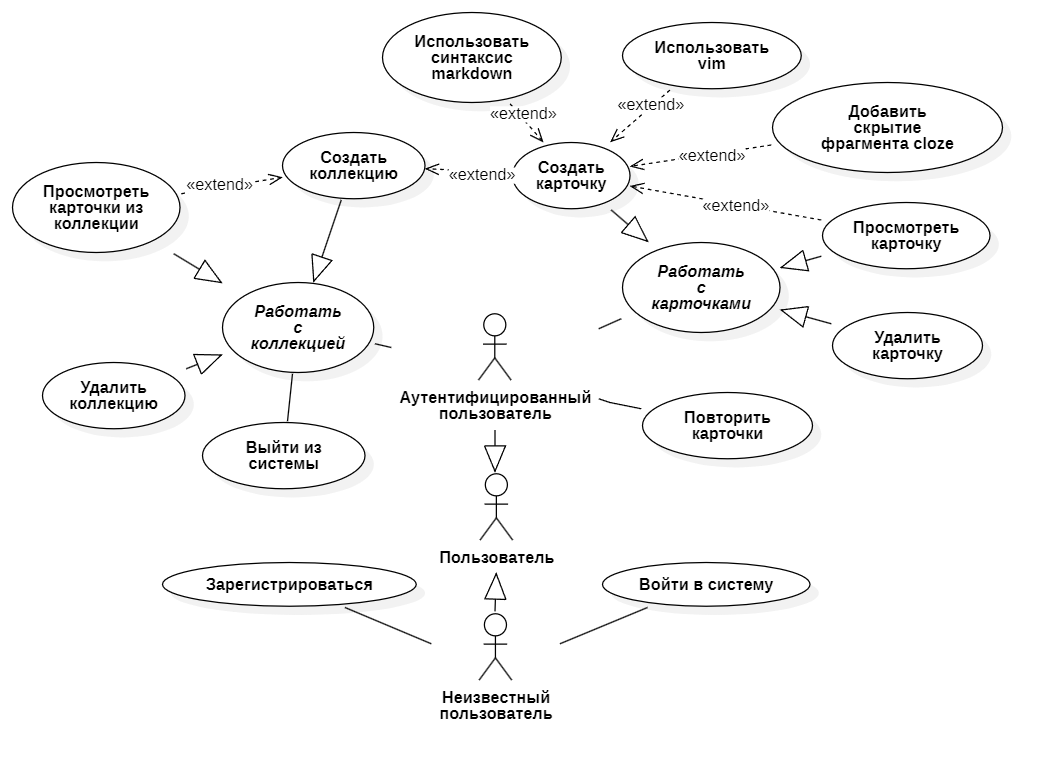


Рисунок 2 – Диаграмма вариантов использования

порядке. Такие диаграммы помогают понять, как ведёт себя система в ответ на определённые события или сценарии использования, что особенно полезно на этапе проектирования программного обеспечения.

Согласно определению, диаграмма активностей — это поведенческая диаграмма, которая моделирует динамический аспект системы, отображая последовательность действий, управляющие переходы, условия ветвления и параллельные процессы [2]. Она может использоваться для описания как бизнес-процессов, так и поведения отдельных компонентов программной системы. В UML 2.5 диаграммы активностей считаются разновидностью state-machine диаграмм, адаптированных для моделирования рабочих процессов и потоков управления в программных системах.

Использование диаграммы активностей особенно оправдано в задачах, связанных с описанием логики пользовательского взаимодействия или бизнес-логики процессов. Благодаря её наглядности и последовательному отображению шагов выполнения, такие диаграммы позволяют эффективно коммуницировать между аналитиками, разработчиками и другими участниками проекта, минимизируя риск недопонимания требований и поведения системы.

На рисунке 3 представлена диаграмма деятельности системы.

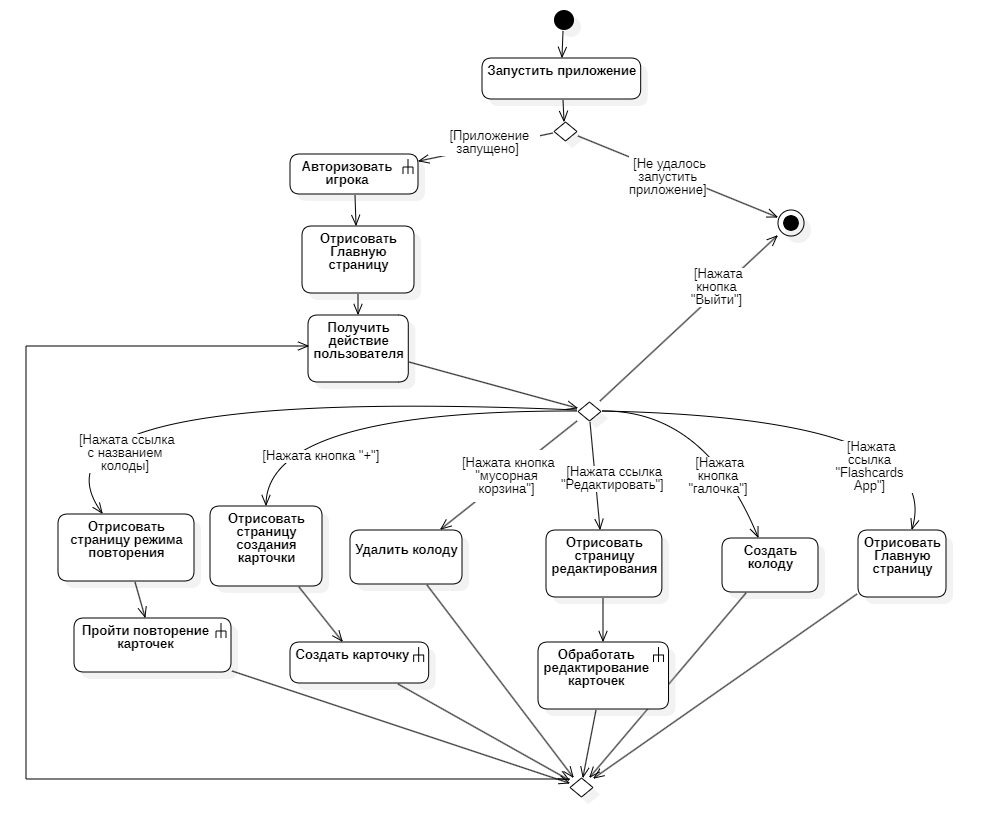


Рисунок 3 – Диаграмма деятельности

* 1. Диаграмма последовательности

Диаграмма последовательностей — это один из основных типов поведенческих диаграмм UML, предназначенный для отображения взаимодействия между объектами в определённой временной последовательности. Она показывает, какие сообщения (вызовы методов, сигналы и т.п.) передаются между участниками системы в рамках конкретного сценария. Такой тип диаграммы помогает понять, как реализуется логика взаимодействия компонентов и как протекает выполнение функциональности во времени.

Согласно спецификации UML, диаграмма последовательностей описывает поведение объектов, упорядоченных по горизонтали, и их взаимодействие по времени, представленному по вертикали. Основными элементами диаграммы являются объекты (участники взаимодействия), линии жизни и сообщения. Она наглядно демонстрирует последовательность вызовов, создание и удаление объектов, а также может отображать условные ветвления и циклы [2].

На рисунке 4 отображена диаграмма последовательности для варианта использования «Повторить карточки».

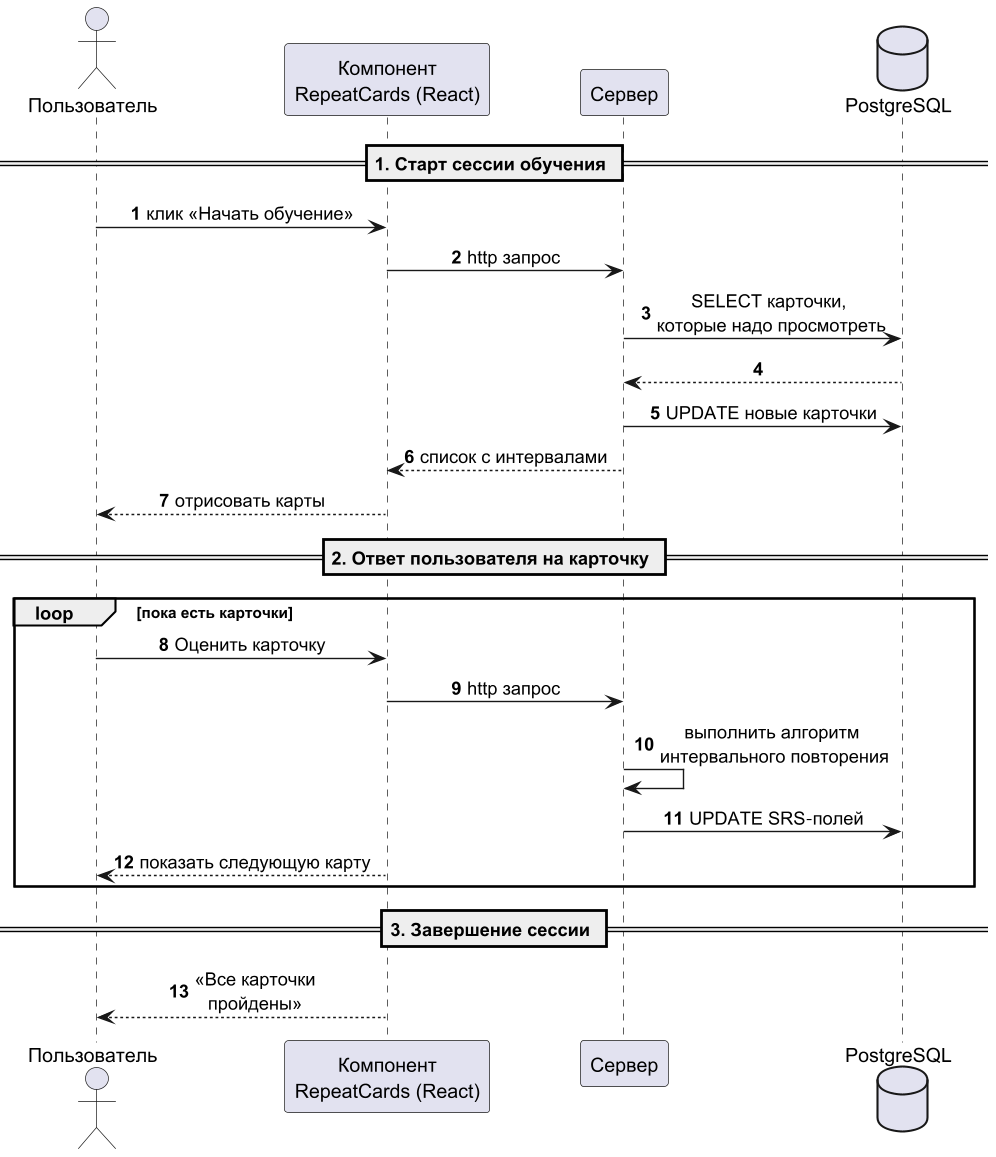


Рисунок 4 – Диаграмма последовательности для варианта использования «Повторить карточки»

1. Руководство пользователя
   1. Назначение системы

Данная система предназначена для обучения по системе Лейтнера с реализацией алгоритма интервального повторения.

* 1. Условия работы системы

Для корректной работы системы необходимо наличие соответствующих программных и аппаратных средств.

1. Требования к техническому обеспечению:
   * ЭВМ типа IBM PC;
   * процессор типа x86 или x64 тактовой частоты 1400 МГц и выше;
   * клавиатура или иное устройство ввода;
   * мышь или иное манипулирующее ввода;
   * дисплей с разрешением не менее 1280 × 768 пикселей;
   * широкополосное подключение к сети Интернет, не менее 1 Мб/сек.
2. Требования к программному обеспечению:
   * операционная система Windows 10 и выше;
   * подсистема линукс для windows (WSL);
   * docker for desktop.
   1. Установка системы

Для локального запуска системы достаточно иметь исходный код проекта. Необходимо запустить терминал и перейти командой cd в директорию с исходным кодом. В данной директории необходимо прописать команду docker compose up. Дождаться запуска приложения. После данных манипуляций система будет доступна в браузере по ссылке http://localhost:3000.

* 1. Работа с системой
     1. Начало работы: регистрация и вход

Для работы с веб‑приложением пользователь должен пройти процедуру регистрации, указав email, пароль и подтверждение пароля. Требования к регистрации:

* email должен соответствовать формату электронной почты (проверка через регулярное выражение: ^\S+@\S+\.\S+$);
* пароль должен содержать не менее 6 символов;
* при успешной регистрации отображается сообщение: «Пользователь успешно зарегистрирован!».

На рисунке 5 приведена форма регистрации.

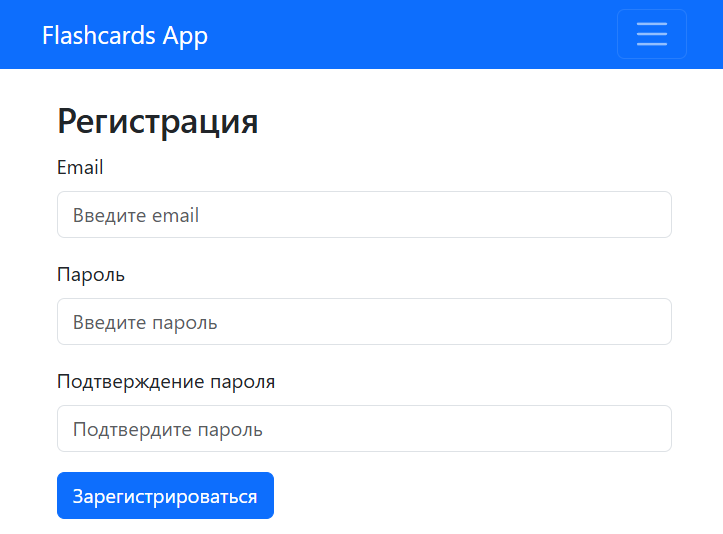


Рисунок 5 – Страница регистрации

После успешной регистрации можно выполнить вход в систему указав логин и пароль. На рисунке 6 приведена форма для входа.

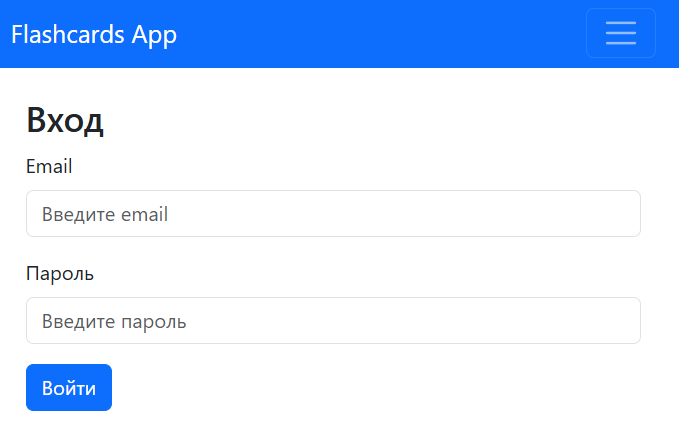


Рисунок 6 – Страница входа

* + 1. Работа с коллекциями карточек

После входа пользователь попадает на главную страницу, где отображается список всех его коллекций карточек. Здесь отображаются следующие данные для каждой коллекции:

* название коллекции;
* количество всех карточек;
* количество новых;
* количество карточек в стадии «обучения»;
* количество карточек; готовых к повторению.

На рисунке 7 изображена главная страница.

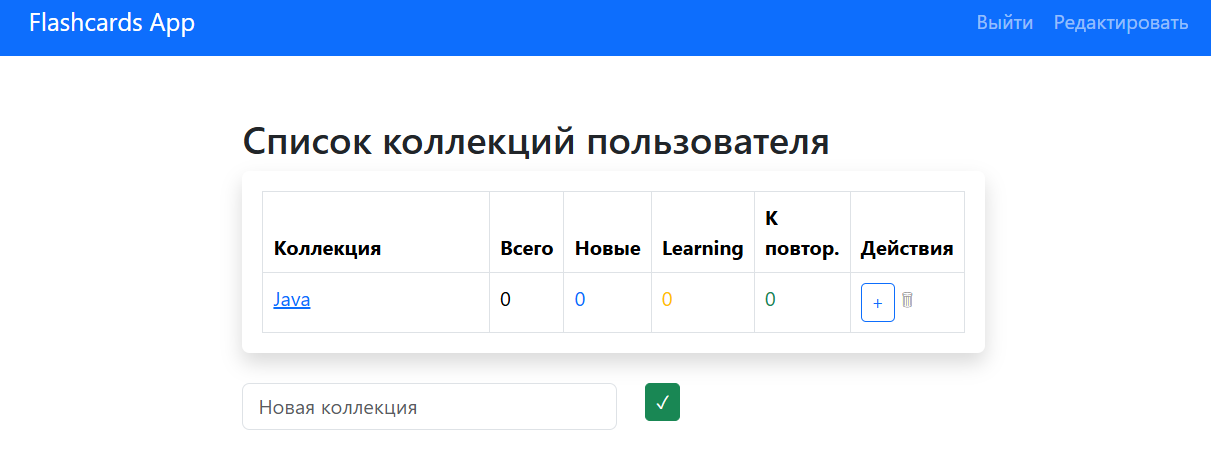


Рисунок 7 – Главная страница

Также на этой странице реализованы следующие функции:

* создание новой коллекции;
* удаление существующих коллекций;
* переход к добавлению карточек в коллекцию;
* запуск режима повторения для выбранной коллекции.
  + 1. Добавление и редактирование карточек

Выбрав коллекцию, пользователь может добавить в неё новые карточки. Редактор поддерживает:

* ввод текста карточки с использованием Markdown-разметки;
* выделение фрагментов для последующего скрытия («cloze»-режим) — позволяет создавать карточки с пропусками, которые надо вспомнить при повторении;
* предпросмотр карточки в реальном времени;
* включение Vim-режима для продвинутых пользователей.

На рисунке 8 изображена страница для создания карточек.

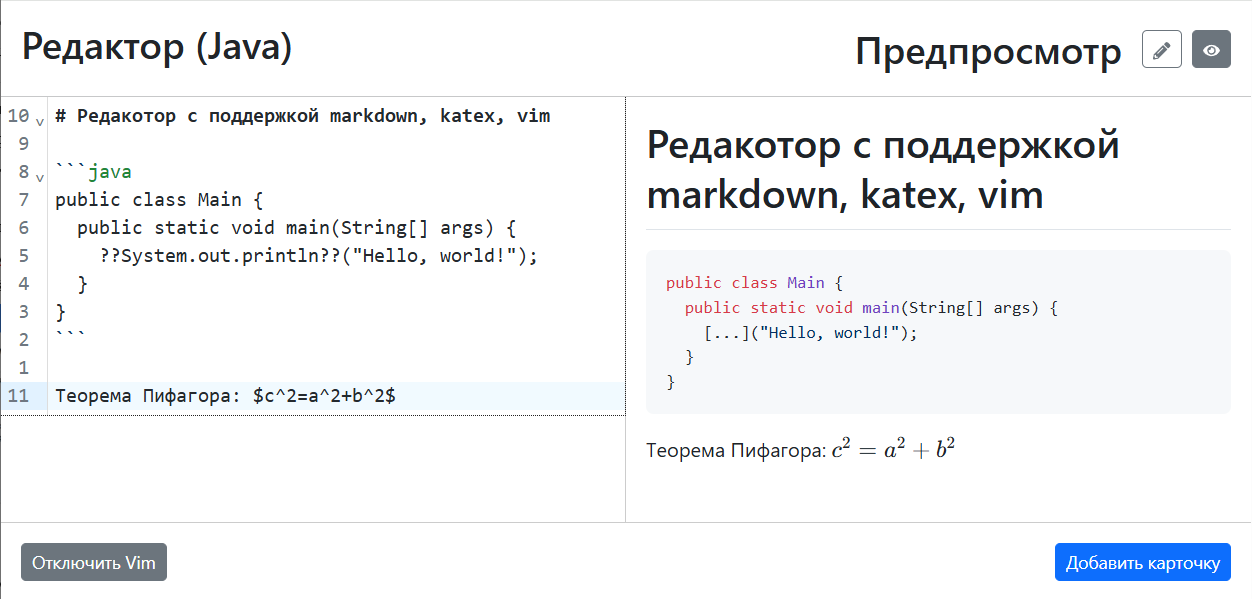


Рисунок 8 – Страница создания карточек

После ввода текста карточки пользователь сохраняет её в коллекции нажатием кнопки «Добавить карточку».

* + 1. Редактирование коллекций и поиск карточек

В разделе редактирования коллекций пользователь может:

* просматривать и фильтровать все свои коллекции;
* просматривать список всех карточек в выбранной коллекции;
* выполнять сортировку и поиск по содержимому карточек;
* просматривать содержимое отдельной карточки в Markdown-формате.

На рисунке 9 представлена страница редактирования коллекций.

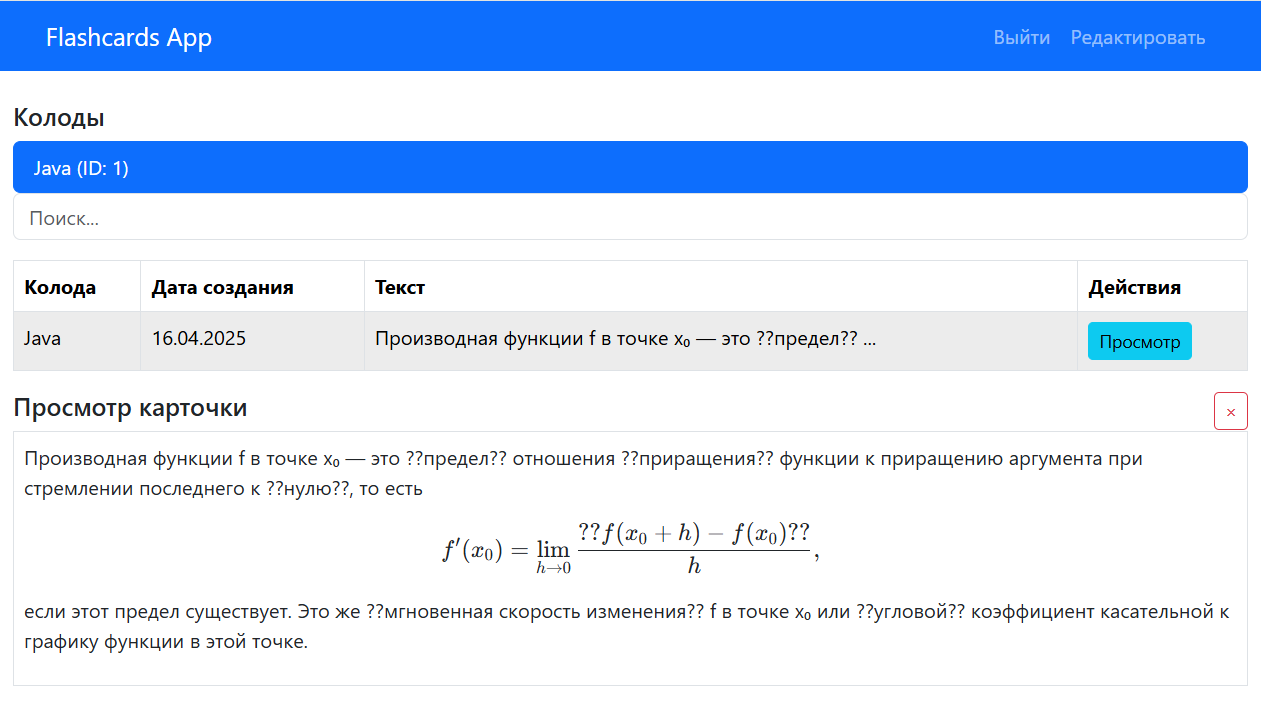


Рисунок 9 – Страница редактирования коллекций

* + 1. Повторение карточек

Основная функция системы — проведение интервального повторения по алгоритму Лейтнера. При запуске повторения:

* пользователю последовательно показываются карточки из выбранной коллекции;
* карточки автоматически сортируются по приоритету показа: сначала новые; затем карточки, находящиеся в стадии обучения, и далее — карточки на повторение,
* для cloze-карточек часть информации скрыта и открывается по нажатию Tab.

После просмотра карточки пользователь выбирает оценку ответа (кнопки: «Again»; «Hard», «Good», «Easy»), что влияет на дату следующего показа по алгоритму интервального повторения.

На рисунке 10 изображена страница режима повторения.

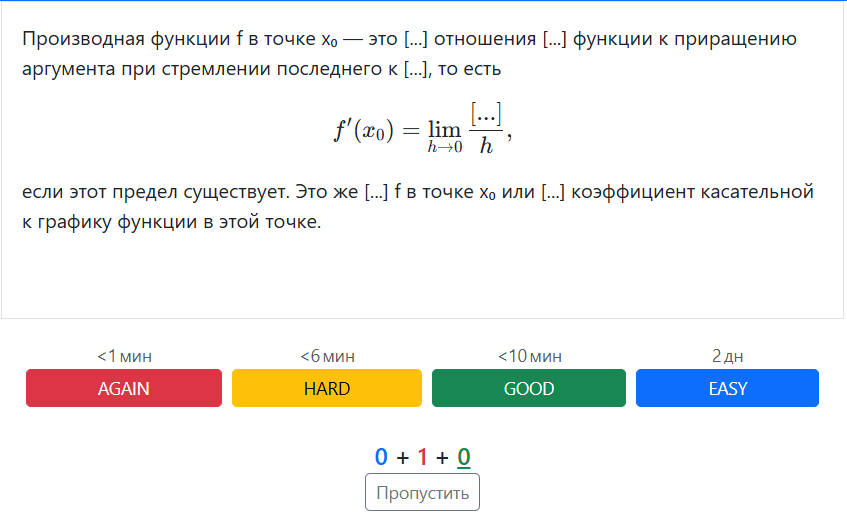


Рисунок 10 – Страница режима повторения

ЗАКЛЮЧЕНИЕ

В результате выполнения преддипломной практики выполнены все поставленные задачи:

* разработан проект системы;
* разработать клиентскую и серверную часть веб-приложения и провести тестирование;
* составить диаграммы, описывающие систему в нотации UML;
* написать руководство пользователя;
* подготовлен и оформлен письменный отчет по практике.

Таким образом, в процессе выполнения преддипломной практики были освоены все необходимые индикаторы (ОПК-1.1, ОПК-1.2, ОПК-1.3, ОПК-2.1 ОПК-2.2; ОПК-2.3 ОПК-3.1, ОПК-3.2 ОПК-3.3, ОПК-4.1 ОПК-4.2, ОПК-4.3, ОПК-5.1, ОПК-5.2, ОПК-5.3, ОПК-6.1, ОПК-6.2, ОПК-6.3) компетенций   
(ОПК-1, ОПК-2, ОПК-3, ОПК-4, ОПК-5, ОПК-6).

Список использованных источников

1. Структурная схема [Электронный ресурс] // Википедия: свободная энциклопедия. URL: https://ru.wikipedia.org/wiki/Структурная\_схема (дата обращения: 20.05.2025).
2. OMG Unified Modeling Language (OMG UML), Version 2.5.1 [Электронный ресурс]. URL: https://www.omg.org/spec/UML/2.5.1 (дата обращения: 20.05.2025).
3. Диаграмма прецедентов [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Диаграмма\_прецедентов (дата обращения: 12.12.2024).
4. Использование диаграммы вариантов использования UML при проектировании программного обеспечения [Электронный ресурс]. URL: https://habr.com/ru/articles/566218/ (дата обращения: 12.12.2024).
5. Буч Г. Язык UML. Руководство пользователя: Пер. с англ. / Г. Буч, Д. Рамбо, Б. Джекобсон. М.: ДМК-Пресс, 2001. 432 с.