

## Слайд 1. Титульный

Здравствуйте, уважаемые члены Государственной экзаменационной комиссии.

Меня зовут **Александр Алёнушка**, я обучающийся группы 6401-020302

Института информатики и кибернетики Самарского университета. Представляю вашему вниманию выпускную квалификационную работу на тему **«Веб-приложение для обучения по системе Лейтнера с реализацией алгоритма интервального повторения»**.

Научный руководитель — доцент кафедры программных систем, к. т. н. **Гордеева О.А.**

---

## Слайд 2. Цель и задачи

**Цель работы** — разработать полнофункциональное веб-приложение, которое внедряет методику Лейтнера и современный алгоритм интервального повторения (SRS). Для достижения цели надо решить задачи представленные на слайде

(Можно не читать)

1. Провести анализ предметной области;
  2. Сделать обзор существующих систем-аналогов
  3. Изучить реализации интервальных алгоритмов;
  4. Спроектировать архитектуру системы (модули, их взаимодействие, выбор технологий);
  5. Реализовать информационное и программное обеспечение;
  6. Провести тестирование и отладку итогового решения.
- 

## Слайд 3. Актуальность

Так как система для обучения, рассмотрим этот процесс детальнее. Процесс обучения обычно состоит из:

1. Первое знакомство с информацией (чтение или просмотр) и конспектирование
2. Возвращение к забытому или не до конца понятому материалу (перечитывание)

Обычная ситуация, когда для изучения последующего материала надо владеть пройденными темами и довольно часто нет прозрачного и ясного понимания какие именно предыдущие темы следует знать для возможности изучения нового материала.

Поэтому надо стараться знать как можно больше. Из-за этого возникает следующие сложности:

- сложность мониторинга степени усвоения предыдущих тем
  - сложность определения что ты реально понял и запомнил, а что нет
  - медленный процесс создания конспектов
  - неэффективность простого перечитывания всего подряд
  - неясность как часто следует повторять материал
- 

## Слайд 4. Актуальность

Разработанная система предлагает следующие решения. Алгоритм интервального повторения решает проблему с количеством повторений и их интервалами. Применение ЭВМ позволяет автоматизировать мониторинг степени усвоения информации. Cloze-тесты, пример есть на слайде. Это обычное сокрытие информации. Пользователь вспоминает например теорему Пифагора, и тут же видит правильный ответ, поэтому он может сразу проанализировать правильно ли он запомнил. Для быстрого создания карточек применяется язык разметки markdown и редактор близкий к редактору кода. Формат флэш карточек позволяют структурировать информацию. Формат веб-приложения позволяет работать с системой на любом устройстве на котором есть веб-браузер и доступ в интернет.

Данную систему можно применять для:

- подготовка к собеседованию
  - прохождение курсов
  - подготовка сотрудников предприятий и организаций к сдаче квалификационных экзаменов
  - подготовка сотрудников органов внутренних дел к прохождению профессиональных аттестаций и экзаменов
- 

## Слайд 5. Система Лейтнера и интервальное повторение

Ядром приложения является алгоритм интервального повторения. Его основой является метод Лейтнера, который заключается в распределении карточек по «ящикам»: успех — переход дальше, ошибка — откат в начало, у каждой коробки свой интервал повторения. Также алгоритм интервального повторения основывается на кривой Эббингауза, которая показывает процент запомненных слов в зависимости от интервала с последнего повторения. А сам алгоритм сделан для автоматического регулирования промежутков между повторениями.

---

## Слайд 6. Анализ аналогов. Anki

Рассмотрим системы аналоги.

Одной из самых популярных систем использующей флэш-карточки и алгоритм интервальных повторений является Anki. Это мультиплатформенное приложение, с богатой экосистемой плагинов и максимально гибкой настройкой алгоритма интервальных повторений. Но Anki имеет сложный интерфейс, требует от пользователя техническую квалификацию и имеет неудобный процесс создания карточек.

---

## Слайд 7. Интерфейс Anki

На слайде показаны окна для создания карточек в anki. Пользователь при помощи языков HTML, CSS и JavaScript создает шаблон для карточек и дальше заполняя форму слева вносит саму информацию. Создавая шаблон пользователь не имеет предпросмотра и вынужден либо создавать текст для карточки в стороннем приложении либо редактировать его внутри формы. Многие пользователи не смогут пользоваться Anki по этим причинам.

---

## Слайд 8. Анализ аналогов. StudyStack

Другой популярный аналог это веб-приложение StudyStack. Оно имеет простой веб-интерфейс, игровые режимы для обучения, однако нет поддержки алгоритма интервального повторения.

---

## Слайд 9. Интерфейс StudyStack

Стади стэк подходит для небольших фрагментов информации и как дополнительная система к обучающему процессу, т.е. не является самостоятельным решением, а больше подходит как сопровождение после урока или доклада.

Наше же приложение имеет доступ из браузера, удобный процесс создания карточек, и поддержку алгоритма интервального повторения.

---

## Слайд 10-14. Как вычисляются интервалы

Детальнее рассмотрим алгоритм интервального повторения. Интервалы вычисляются по простой рекуррентной формуле. В ней участвует параметр И ЭФ, или параметр легкости,

который в свою очередь также вычисляется по рекуррентной формуле с учетом оценки пользователя КЮ.

---

## **Слайд 15. Практическая схема алгоритма**

Диаграмма иллюстрирует жизненный цикл карточки: Создание карточки -> при первичном изучении всегда используются жестко заданные интервалы в 10 минут и 1 день -> после чего происходит перевод в стадию повторения, где интервалы вычисляются по алгоритму интервального повторения. Ошибка «Again» возвращает карточку в режим переобучения.

---

## **Слайд 16. Структурная схема системы и используемые технологии.**

Сделаем обзор на реализацию веб-приложения. Система разделена на три слоя: Клиентское приложение реализовано с использованием языка программирования javascript и фреймворка react.js. Серверная часть с использованием языка java и фреймворка spring boot, используется graphql для взаимодействия серверного и клиентского приложений. База данных postgresql и система управления миграциями liquibase с ORM фреймворком hibernate. Для запуска приложения используется система контейнеризации Docker.

---

## **Слайд 17. Диаграмма вариантов использования**

На диаграмме вариантов использования можно наглядно увидеть какие есть возможности у пользователя. В основном это работа с карточками, коллекциями и повторение карточек.

---

## **Слайд 18. Диаграмма последовательности режима повторения**

На диаграмме последовательности режима повторения показан обмен сообщениями: фронтенд запрашивает карточки, после отправляет оценки пользователя на сервер, где рассчитываются новые интервалы.

И перейдем к интерфейсу приложения.

---

## **Слайды 19–20. Интерфейс приложения**

Страницы и формы для регистрации и входа имеют классический набор полей.

Для создания карточек используется редактор близкий к редакторам кода. В правой части сразу видно как будет выглядеть итоговая карточка. Используется язык разметки markdown.

При повторении пользователь видит карточку со скрытыми фрагментами. Нажимая клавишу таб эти фрагменты раскрываются и пользователь может проверить свою память.

Сверху главная страница с таблицей колод. Внизу страница редактирования колод и карточек.

---

## Слайд 21. Выводы

В результате выполнены все поставленные задачи:

- Проанализирована предметная область и исследованы аналоги;
- Изучены и реализованы алгоритмы интервального повторения SM-2 с адаптацией под разные режимы карточек;
- Спроектирована модульная архитектура и выбраны технологии React + Spring Boot + GraphQL;
- Разработан удобный UI с поддержкой Vim-режима и Markdown;
- Проведены юнит-, интеграционные и нагрузочные тесты — приложение успешно обслуживает 500 одновременных сессий;
- Подготовлена документация и оформлена ВКР.

Таким образом создано современное, расширяемое и пользовательски-дружественное решение, повышающее эффективность самообучения за счёт адаптивных интервалов повторения.

**Благодарю за внимание!** Готов ответить на ваши вопросы.