

Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ И. В. Евдокимов
подпись
« ____ » _____ 20 __ г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 «Программная инженерия»

Разработка сервиса для повышения навыков владения иностранными языками
на основе техник «СуперПамяти» и захвата субтитров

Руководитель	_____	к. т. н., доцент	И. В. Евдокимов
	подпись, дата		
Выпускник	_____		М. И. Носов
	подпись, дата		
Выпускник	_____		А. Д. Чагин
	подпись, дата		
Выпускник	_____		Н. Е. Двоглазов
	подпись, дата		
Нормоконтролер	_____		О. А. Антамошкин
	подпись, дата		

Красноярск 2018

Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой

И.В. Евдокимов

подпись

« ____ » _____ 2018 г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студентам

А. Д. Чагин, М. И. Носов, Н. Е. Двоглазов

Группа КИ14-17Б Направление (специальность) 09.03.04
Программная инженерия

Тема выпускной квалификационной работы «Разработка сервиса для повышения навыков владения иностранными языками на основе техник «СуперПамяти» и захвата субтитров»

Утверждена приказом по университету № 7834/с от 2018-05-30
Об утверждении тем ВКР, назначении руков-й, консульт-в

Руководитель ВКР И. В. Евдокимов, кандидат технических наук, доцент

Исходные данные для ВКР _____

Перечень разделов ВКР Введение, обзор методов и средств разработки сервиса для повышения навыков владения иностранными языками, проектирование сервиса «EduJetPlayer» и анализ его аналогов, разработка сервиса «EduJetPlayer», заключение

Перечень графического материала Презентационные слайды PowerPoint

Руководитель ВКР

подпись

И. В. Евдокимов

Задание принял к исполнению

подпись

А. Д. Чагин

Задание принял к исполнению

подпись

М. И. Носов

Задание принял к исполнению

подпись

Н. Е. Двоглазов

« ____ » _____ 2018 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка сервиса для повышения навыков владения иностранными языками на основе техник «СуперПамяти» и захвата субтитров» содержит 91 страницу текстового документа, 3 таблицы, 1 приложение, 29 использованных источников.

СЕРВИС ДЛЯ ИЗУЧЕНИЯ ЯЗЫКОВ, ВЕБ ПРИЛОЖЕНИЕ, АНАЛИЗ АНАЛОГОВ СЕРВИСА, РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ СЕРВИСА «EDUJETPLAYER».

Цели:

Создать бесплатный сервис, который поможет пользователю в изучении иностранных языков, комбинируя современные подходы к запоминанию информации и обучению.

Задачи:

- анализ существующих аналогов и их возможностей;
- проектирование сервиса с учетом возможностей существующих решений;
- разработка сервиса;
- тестирование концепции;
- доработка функционала.

Важность нашего сервиса состоит в том, что он позволит с пользой проводить время за просмотром любимых сериалов или фильмов, попутно изучая желаемый язык. Эффективность методов нашего сервиса состоит в том, что он предоставляет удобный интерфейс для сохранения и перевода субтитров, и анализом уже изученных слов, для формирования списка слов необходимых для понимания конкретного видео пользователем.

Итогом нашей работы стал сервис «EduJetPlayer», способный предоставлять пользователям функционал для удобного изучения иностранных языков при методе захвата и перевода субтитров, с последующим их анализом и обработкой для формирования базы изученных пользователем слов.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 Обзор методов и средств разработки сервиса для повышения навыков владения иностранными языками	7
1.1 Инструменты проектирования.....	7
1.2 Языки программирования.....	8
1.3 Средства разработки	10
1.4 Используемые библиотеки.....	16
2 Проектирование сервиса «EduJetPlayer» и анализ его аналогов	22
2.1 Анализ существующих аналогов и их возможностей	22
2.2 Проектирование сервиса	27
2.3 Архитектура сервиса.....	32
3 Разработка сервиса «EduJetPlayer».....	35
3.1 Командное взаимодействие	35
3.2 Реализованный функционал	45
3.3 Методология работы с сервисом.....	68
3.4 Технические требования к серверу и установка.....	88
ЗАКЛЮЧЕНИЕ	91
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	92
ПРИЛОЖЕНИЕ А.....	95

ВВЕДЕНИЕ

Первым делом необходимо разработать веб-сервис, который будет предоставлять пользователям возможность загрузки своих видео файлов и файлов субтитров, для их последующего просмотра и изучения иностранного языка при помощи предоставляемого интерфейса. Сервис должен иметь возможность автоматического перевода субтитров и сохранения выбранных слов/фраз для изучения. Одной из главных функций будет являться анализ загруженных пользователем данных для формирования списка слов, необходимых к изучению. Также в качестве дополнительного функционала должна быть реализована система автоматического подбора подходящих субтитров по запросу пользователя. Благодаря функционалу для загрузки пользователем своих материалов, наш сервис может быть использован для изучения любых языков, которые поддерживает google-translator [1].

Наш сервис будет использовать модули и библиотеки с open-source лицензией, что позволит сделать сервис бесплатным для использования, не вкладывая деньги в разработку.

Все существующие на данный момент решения не предоставляют пользователю возможности загрузки своих материалов для изучения, а также не имеют функционала для анализа уже изученных пользователем слов. К тому же большинство существующих сервисов – платные и предоставляют ограниченное число видео материалов по платным подпискам, при этом поддерживая только английский язык.

1 Обзор методов и средств разработки сервиса для повышения навыков владения иностранными языками

В данном разделе описан этап проектирования нашего сервиса, используемые инструменты, языки программирования, среды разработки и их анализ.

1.1 Инструменты проектирования

В этом разделе мы рассмотрим инструменты проектирования, применявшиеся нами при разработке нашего сервиса для помощи в изучении иностранных языков.

Для проектирования нами был использован язык обширного профиля UML – открытый стандарт, который часто обновляется с целью удобства использования и оптимизации сложности понимания.

UML состоит из графических обозначений и содержит 16 видов диаграмм, для описания всех аспектов разработки нашего сервиса. UML это не язык программирования, но существуют технологии для генерации кода на основе UML-диаграмм.

Для описания бизнес-процессов мы использовали методологию функционального проектирования IDEF0.

Основные преимущества IDEF0 состоят в следующем:

- полнота описания бизнес-процесса;
- комплексность при декомпозиции;
- возможность агрегирования и детализации потоков данных и информации (разделение и соединение стрелок);
- жесткие требования методологии, обеспечивающие получение моделей процессов стандартного вида;
- простота документирования процессов; соответствие подхода к описанию процессов в IDEF0 стандартам ISO 9000:2000.

1.2 Языки программирования

В этом разделе мы рассмотрим языки программирования, которые могут быть использованы для программирования серверной или клиентской части нашего сервиса, а во второй главе будет представлен их анализ.

Рассмотрим языки программирования серверов. Это PHP [2], Javascript [3], Ruby [4], Python [5].

PHP (англ. PHP: Hypertext Preprocessor – «PHP: препроцессор гипертекста»; первоначально Personal Home Page Tools – «Инструменты для создания персональных веб-страниц») – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Язык и его интерпретатор (Zend Engine) разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией, несовместимой с GNU GPL.

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript (стандарт ECMA-262).

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки. При разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет какая-либо компания или

организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным товарным знаком компании Oracle Corporation.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).

Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные. Есть реализации интерпретаторов для JVM (с возможностью компиляции), MSIL (с возможностью компиляции), LLVM и других. Проект PyPy предлагает реализацию Python с использованием JIT-компиляции, которая значительно увеличивает скорость выполнения Python-программ.

Python – активно развивающийся язык программирования, новые версии (с добавлением (изменением) языковых свойств) выходят примерно раз в два с половиной года. Вследствие этого и некоторых других причин на Python отсутствуют стандарт ANSI, ISO или другие официальные стандарты, их роль выполняет CPython

Ruby – динамический, рефлексивный, интерпретируемый высокоуровневый язык программирования. Язык обладает независимой от операционной системы реализацией многопоточности, строгой динамической типизацией, сборщиком мусора и многими другими возможностями. По особенностям синтаксиса он близок к языкам Perl и Eiffel, по объектно-ориентированному подходу – к Smalltalk. Также некоторые черты языка взяты из Python, Lisp, Dylan и Clu.

Кроссплатформенная реализация интерпретатора языка является полностью свободной.

1.3 Средства разработки

В этом разделе нами будут рассмотрены различные средства разработки, которые используются при создании сервиса подобного нашему.

Node или Node.js – программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи NW.js, AppJS или Electron для Linux, Windows и macOS) и даже программировать микроконтроллеры (например, tessel и espruino). В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом-выводом.

В состав Node.js входит собственная система управления пакетами npm. Система управления пакетами – набор программного обеспечения, позволяющего управлять процессом установки, удаления, настройки и обновления различных компонентов программного обеспечения. Системы

управления пакетами активно используются в различных дистрибутивах операционной системы Linux и других UNIX-подобных операционных системах.

Программное обеспечение представляется в виде особых пакетов, содержащих помимо дистрибутива программного обеспечения набор определённых метаданных, которые могут включать в себя полное имя пакета, номер версии, описание пакета, имя разработчика, контрольную сумму, отношения с другими пакетами. Метаданные сохраняются в системной базе данных пакетов.

MongoDB является системой управления базой данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. СУБД управляет наборами JSON-подобных документов, хранимых в двоичном виде в формате BSON. Хранение и поиск файлов в MongoDB происходит благодаря вызовам протокола GridFS. Подобно другим документоориентированным СУБД (CouchDB и др.), MongoDB не является реляционной СУБД.

В MongoDB нет такого понятия, как «транзакция». Атомарность гарантируется только на уровне целого документа, то есть частичного обновления документа произойти не может.

Отсутствует понятие «изоляции». Любые данные, которые считываются одним клиентом, могут параллельно изменяться другим клиентом. MongoDB используется для быстрой разработки хорошо масштабируемых решений.

Теперь мы рассмотрим языки программирования клиентской части нашего сервиса.

JavaScript используется в клиентской части веб-приложений: клиент-серверных программ, в которых клиентом является браузер, а сервером – веб-сервер, имеющих распределённую между сервером и клиентом логику. Обмен информацией в веб-приложениях происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами.

Большинство фреймворков автоматизированного тестирования JavaScript-кода предполагают запуск тестов в браузере. Это осуществляется при помощи HTML-страницы, являющейся контекстом тестирования, которая, в свою очередь загружает всё необходимое для осуществления тестирования. Альтернатива – запуск тестов из командной строки. В этом случае используются окружения, отличные от браузера, например, Rhino. Одним из первых инструментов такого рода является Crosscheck, позволяющий тестировать код, эмулируя поведение Internet Explorer 6 и Firefox версий 1.0 и 1.5. Другой пример фреймворка автоматизированного тестирования JavaScript-кода, не использующего браузер для запуска тестов – библиотека env.js, созданная Джоном Резигом. Она использует Rhino и при этом содержит эмуляцию окружения браузера и DOM.

Главная проблема систем тестирования, не использующих браузеры, в том, что они используют эмуляции, а не реальные окружения, в которых выполняется код. Это приводит к тому, что успешное прохождение тестов не гарантирует того, что код корректно отработает в браузере. Проблемой систем тестирования, использующих браузер, является сложность работы с ними, необходимость осуществления рутинных неавтоматизированных действий. Для решения этого JsTestDriver [6], фреймворк автоматизированного тестирования, разрабатываемый Google, использует сервер, взаимодействующий с браузерами для осуществления тестирования. Сходным образом ведёт себя Selenium Remote Control, входящий во фреймворк автоматизированного тестирования Selenium [7]: он включает в себя сервер, запускающий и завершающий браузеры и действующий как HTTP-прокси для запросов к ним. Кроме того, в Selenium содержится Selenium Grid, позволяющий осуществлять одновременное тестирование JavaScript-кода на разных компьютерах с разными окружениями, уменьшая время выполнения тестов. Testswarm [8], имеющее поддержку фреймворков автоматизированного тестирования JavaScript-кода QUnit (библиотека jQuery), UnitTestJS (библиотека Prototype), JSSpec (библиотека

MooTools), JsUnit, Selenium и Dojo Objective Harness, представляет собой распределённое средство поддержки непрерывной интеграции.

Негативное свойство, которым может обладать фреймворк автоматизированного тестирования JavaScript-кода – наличие зависимостей. Это создаёт риск отказа в работе тестируемого кода, успешно проходящего тесты, в среде с отсутствием этих зависимостей. Например, исходная версия JsUnitTest, фреймворка, созданного и использовавшегося для тестирования библиотеки Prototype, зависела от самой Prototype, изменяющего свойства объектов в глобальной области видимости.

В качестве редакторов исходного кода могут использоваться рассмотренное далее ПО.

Visual Studio Code [9] – редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

Visual Studio Code основан на Electron – фреймворк, позволяющий с использованием Node.js разрабатывать настольные приложения, которые работают на движке Blink. Несмотря на то, что редактор основан на Electron, он не использует редактор Atom. Вместо него реализуется веб-редактор Monaco, разработанный для Visual Studio Online.

Многие возможности Visual Studio Code не доступны через графический интерфейс, зачастую они используются через палитру команд или JSON файлы (например, пользовательские настройки). Палитра команд представляет собой подобие командной строки, которая вызывается сочетанием клавиш.

Visual Studio так же позволяет заменять кодовую страницу при сохранении документа, символы перевода строки и язык программирования текущего документа.

Visual Studio Code является главным редактором и включает в себя функции, необходимые для высокопроизводительного редактирования исходного кода.

На рисунке 1 показан раздел редактирования исходного кода.

Преимущества:

- горячие клавиши;
- множественный выбор текста (мульти-курсорный);
- мульти-курсорный редактор;
- уменьшение (расширение) выделения текста;
- выбор столбца из обычного текста;
- поиск по файлам;
- технология IntelliSense.

Функции:

- загрузка файла gulpfile.js;
- интерфейс остановки и запуска задач.

Рассмотрим подходящие под язык программирования Javascript среды программирования: WebStorm [10], Notepad++ [11].

WebStorm – среда для разработки на JavaScript, которая подходит для client-side-разработки, создания приложений на Node.js и мобильных приложений на React Native.

Главное достоинство WebStorm – это удобный и умный редактор для JavaScript, HTML и CSS, который также поддерживает TypeScript, CoffeeScript, Dart, Less, Sass и Stylus и фреймворки, например, Angular, React и Vue.js.

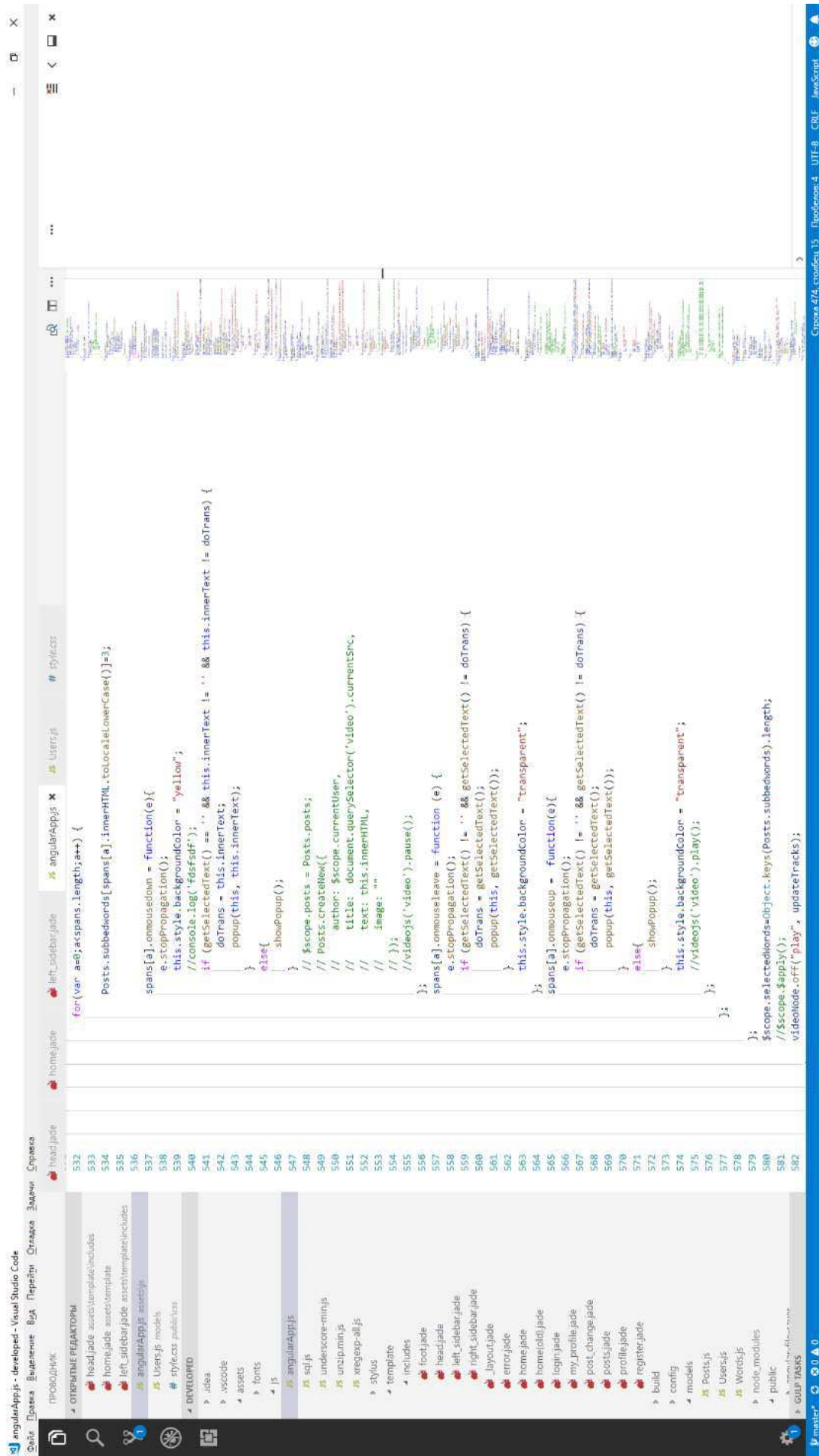


Рисунок 1 – Visual Studio Code

WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями. Важным преимуществом интегрированной среды разработки WebStorm является работа с проектами (в том числе, рефакторинг кода JavaScript, находящегося в разных файлах и папках проекта, а также вложенного в HTML). Поддерживается множественная вложенность (когда в документ на HTML вложен скрипт на Javascript, в который вложен другой код HTML, внутри которого вложен Javascript) – то есть в таких конструкциях поддерживается корректный рефакторинг.

WebStorm поддерживает отладку приложений в node.js. Также поддерживается полный набор функций редактирования приложений на javascript – как для исполнения на сервере, так и в браузере: автодополнение, навигация по коду, рефакторинг и проверка на ошибки.

Для вывода сообщений Node.js используется отдельная вкладка в IDE.

Notepad++ – свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки. Поддерживает открытие более 100 форматов. Базируется на компоненте Scintilla, написан на C++ с использованием STL, а также Windows API и распространяется под лицензией GNU General Public License. Базовая функциональность программы может быть расширена как за счёт плагинов, так и сторонних модулей, таких как компиляторы и препроцессоры.

1.4 Используемые библиотеки

В этом разделе мы приведем библиотеки, которыми пользовались при создании нашего сервиса для помощи в изучении иностранных языков и приведем краткое описание каждой.

Anki – универсальная программа для помощи в изучении иностранных слов и вообще любой информации, представленной в виде карточек. Перед аналогами она имеет широкий ряд преимуществ:

- большой выбор возможностей по настройке вида карточек;
- открытый исходный код;
- поддержка большого количества форматов мультимедиа;
- большое количество плагинов;
- настройка схемы обучения;
- наглядная и подробная статистика обучения;
- поддержка большинства современных платформ (linux, windows, mac, iphone, android, список можно продолжить!);
- синхронизация данных.

Однако благодаря столь широкому функционалу, пользователю поначалу может быть трудно с ней разобраться. Но в интернете есть большое количество обучающих роликов по работе с Anki.

Для работы с Anki могут быть использованы:

а) официальные версии программ:

- 1) Anki (на основе PyQt; также называется «AnkiDesktop») – основная версия, для настольного компьютера (Windows, Mac OS X, Linux, FreeBSD; можно запустить на Maemo);
- 2) AnkiMobile – мобильное приложение только для iOS с закрытым исходным кодом. Приложение платное (финансирует разработку Anki);
- 3) AnkiWeb – веб-приложение для повторений и хостинг колод и дополнений (плагинов);

б) совместимые:

- 1) AnkiDroid – бесплатное мобильное приложение с открытым исходным кодом для Android-устройств, от других авторов. Позволяет редактировать карточки, а также использовать другой сервер синхронизации;

2) AnkiUniversal – универсальное приложение Windows (UWP) для устройств на базе Windows 10. Его можно установить на ПК, приставки Xbox, смартфоны на Windows 10 Mobile, очки Hololens или Surface Hub.

Video.js – это библиотека JavaScript и CSS, которая облегчает работу по созданию HTML5-видео. Библиотеку можно также назвать HTML5-видеоплеером. Video.js обеспечивает управление и создание внешней оболочки плеера в HTML/CSS, а также исправляет несоответствия в разных браузерах и обладает дополнительным функционалом: полноэкранный режим и наличие субтитров, поддержка плагинов. Video.js позволяет переключаться на Flash тогда, когда недоступен HTML5, а также обеспечивает JavaScript API для взаимодействия с видео.

В создании сервера использовался модуль express.js [12].

Данный модуль является каркасом web-приложений для Node.js, реализован как свободное и открытое программное обеспечение под лицензией MIT. Он спроектирован для создания веб-приложений и API. Де-факто является стандартным каркасом для Node.js. Автор фреймворка, TJ Holowaychuk, описывает его как созданный на основе написанного на языке Ruby каркаса Sinatra, подразумевая, что он минималистичен и включает большое число подключаемых плагинов. Express может являться backend'ом для программного стека MEAN, вместе с базой данных MongoDB и каркасом AngularJS [13] для frontend'a, что и используется в нашем проекте.

Для обеспечения безопасности паролей пользователей используется пакет PassportJS [14]. Это middleware для авторизации под node.js. Passport поддерживает авторизацию с помощью огромного количества сервисов, включая «ВКонтакте».

Далее будут представлены вспомогательные библиотеки и их краткое описание:

- angular-file-saver – это сервис AngularJS, который использует файлы FileSaver.js и Blob.js для реализации интерфейса HTML5 W3C saveAs () в браузерах, которые не поддерживают его;

- `body-parser` – средство разбора `body` в `Node.js`, которое разбирает входящие запросы в промежуточном программном обеспечении перед обработчиками, доступными под свойством `req.body`;

- `Chardet` – модуль обнаружения символов для `NodeJS`, написанный на чистом `Javascript`, основан на проекте `ICU` <http://site.icu-project.org/>, который использует анализ случайности, чтобы определить наиболее вероятную кодировку;

- `Cheerio` – быстрая и гибкая реализация ядра `jQuery`, разработанного специально для сервера, `Cheerio` анализирует разметку и предоставляет API для обработки результирующей структуры данных;

- `cookie-parser` – разбирает заголовки `Cookie` и заполнит `req.cookies` с помощью объекта, связанного с именами файлов `cookie`, имеет возможность включить поддержку подписанного файла `cookie`, передав секретную строку, которая назначает `req.secret`, чтобы она могла использоваться другим промежуточным программным обеспечением;

- `cURL` – кроссплатформенная служебная программа командной строки, позволяющая взаимодействовать с множеством различных серверов по множеству различных протоколов с синтаксисом `URL`;

- `fs-extra` – добавление методов файловой системы, которые не включены в собственный модуль `fs` и добавляет поддержку `promise` в методы `fs`, также добавлена поддержка `graceful-fs` для предотвращения ошибок `EMFILE`;

- `google-translator` – модуль `Node.js` для работы с API `Google Cloud Translation`, который автоматически обрабатывает массовые переводы, превышающие лимит запросов API переводов `Google`;

- `Jade` – высокопроизводительный механизм шаблонов, сильно зависящий от `Haml` и реализованный с помощью `JavaScript` для узлов и браузеров;

- `Jsonwebtoken` – реализация `JSON Web Tokens`;

- `Mongoose` – инструмент моделирования объектов `MongoDB`, предназначенный для работы в асинхронной среде;

- `Morgan` – средство промежуточного буфера `HTTP`-запроса для `Node.js`;

- node-7zip – автономная оболочка NodeJs для 7zip, использует NodeJs и 7zip для обработки файлов и каталогов архивирования;

- Passport – Express-совместимое промежуточное программное обеспечение для Node.js, единственной целью которого является проверка подлинности запросов, которые он выполняет с помощью расширяемого набора плагинов, известных как стратегии;

- Relevancy – содержит базовый алгоритм сортировки (взвешивания), который может использоваться для взвешивания короткой строки относительно другой короткой строки, может отображать релевантность между двумя строками, но только однонаправленно;

- Request – разработан как самый простой способ сделать http-запросы, поддерживает HTTPS и перенаправления по умолчанию;

- Subsr – конвертер субтитров для JavaScript и инструмент командной строки;

- Gulp – инструмент для автоматизации трудоемких задач в рабочем процессе разработки;

- vjs.video – директива angular.js для video.js, с помощью которой можно легко включить видео и аудио в свои проекты, используя надежный видеоплеер HTML video.js. Директива vjs-video обрабатывает всю сложность, связанную с использованием video.js в приложении Single Pages (SPA) AngularJS, и включает в себя следующие функции: Начальная загрузка video.js после того, как сайт будет готов, правильное размещение видео, когда текущий угловой вид выходит за рамки, загрузка и горячая свопинг видео с использованием привязки угловых данных, поддержка аудио (для версий video.js > = 4.9), адаптивный контейнер (для версий video.js 4.x).

- sql.js – SQLite скомпилирован в javascript, это порт SQLite для JavaScript, путем компиляции кода SQLite C с помощью Emscripten. Он использует файл виртуальной базы данных, хранящийся в памяти, и, таким образом, не сохраняет изменения, внесенные в базу данных. Тем не менее, он позволяет вам импортировать любой существующий файл sqlite и экспортировать созданную

базу данных в виде массива, заполненного javascript. Здесь нет связей C или компиляции node-gyp, sql.js – это простой файл javascript, который может использоваться как любая традиционная библиотека javascript. Если вы создаете собственное приложение в javascript (например, с помощью Electron) или работаете в node.js. SQLite является общедоступным, sql.js лицензируется MIT.

- underscore.js – является библиотекой утилиты для JavaScript, которая обеспечивает поддержку обычных функциональных возможностей (карта, сокращение, фильтр ...) без расширения любых основных объектов JavaScript.

- unzip.min.js – это реализация ZLIB (RFC1950), DEFLATE (RFC1951), GZIP (RFC1952) и PKZIP в JavaScript.

- xregexp-all.js – обеспечивает расширенные (и расширяемые) регулярные выражения JavaScript. Вы получаете современный синтаксис и флаги, помимо того, что браузеры поддерживают изначально. XRegExp также представляет собой пояс службы регулярных выражений с инструментами, облегчающими парсинг выражений, освобождая от непоследовательности перекрестного браузера regex и других неприятностей. XRegExp поддерживает все собственные синтаксисы регулярных выражений ES6. Он поддерживает браузеры ES5 +, с возможностью использовать его с Node.js или как модуль RequireJS. XRegExp компилируется в собственные объекты RegExp. Поэтому регулярные выражения, созданные с помощью XRegExp, выполняются так же быстро, как и собственные регулярные выражения.

- Videojs-overlay – плагин для отображения простых наложений – аналогичный по функционалу YouTube «Аннотации» и по внешнему виду - во время воспроизведения видео. Он не встроен в video.js в качестве компонента ModalDialog.

2 Проектирование сервиса «EduJetPlayer» и анализ его аналогов

В данном разделе нами были рассмотрены 3 аналога сервисов для помощи в изучении иностранных языков, а также описаны его этапы проектирования.

2.1 Анализ существующих аналогов и их возможностей

В данном разделе нами будут рассмотрены 3 аналогичных сервиса для помощи в изучении иностранных языков, где будет проведён предварительный и отдельный анализ сервисов, сделан вывод.

Основная идея предварительного анализа состоит в выявлении аналогичных решений на основе списка функций:

- видеоплеер с поддержкой интерактивных субтитров;
- автоматический перевод субтитров;
- сохранение выбранных слов и фраз;
- интерфейс для изучения сохраненных материалов;
- формирование базы изученных слов;
- анализ новых субтитров на предмет неизученных слов;
- загрузка собственных видео и субтитров;
- поддержка больше 2х языков.

В данном разделе будут проанализированы следующие решения:

- «ororo.tv» [15];
- «learnenglishfeelgood.com» [16];
- «animelon.com» [17].

Первым мы рассмотрим сервис «learnenglishfeelgood.com» (рисунок 2), однако на него необходимо оформлять подписку на год, и она стоит 60\$.

Плюсы:

- простота использования;
- возможность просмотра и изучения сохраненных субтитров в личном кабинете;

- большой набор видео по платной подписке.

Минусы:

- нет возможности загрузки своих видео;
- поддержка только русского и английского языка;
- дорогая подписка;
- наличие неисправленных ошибок.

Далее мы рассмотрим сервис ororo.tv, он имеет возможность бесплатного использования, но в таком случае есть лимит на 1 серию в день. Сервис работает только с русским и английским языками. Большая часть материалов для просмотра не доступна по бесплатной подписке, однако стоит заметить, что продуманно реализован просмотр перевода субтитров в видеоплеере и их сохранения.

На рисунке 3 представлен пользовательский интерфейс видеоплеера сервиса «ororo.tv».

Плюсы:

- простота использования;
- возможность просмотра и изучения сохраненных субтитров в личном кабинете;
- большой набор видео по платной подписке.

Минусы:

- нет возможности загрузки своих видео;
- поддержка только русского и английского языка;
- дорогая подписка;
- наличие неисправленных ошибок.

Лучшим существующим аналогом является сервис «animelon.com», однако он обладает некоторыми существенными недостатками, такими как поддержка только английского и японского языка и отсутствие возможности загрузки своих видео. На рисунке 4 представлен пользовательский интерфейс сервиса animelon.com.

Минусы:

- нет возможности загрузки своих видео;
- поддержка только японского и английского языка.

Плюсы:

- простота использования;
- возможность просмотра и изучения сохраненных субтитров в личном кабинете;
- сервис абсолютно бесплатный;
- большой выбор сериалов для просмотра;
- удобная реализация интерфейса для работы с японским языком.

При анализе аналогов мы рассмотрели самые известные сервисы, предоставляющие интерфейс для изучения иностранных языков по субтитрам.

Среди них оказался только один бесплатный сервис, способный предоставить полноценный функционал, однако все рассмотренные решения не предоставляют возможности загрузки собственных видео и субтитров, что делает наше решение наиболее актуальным.



Рисунок 2 – сервис «learnenglishfeelgood.com»

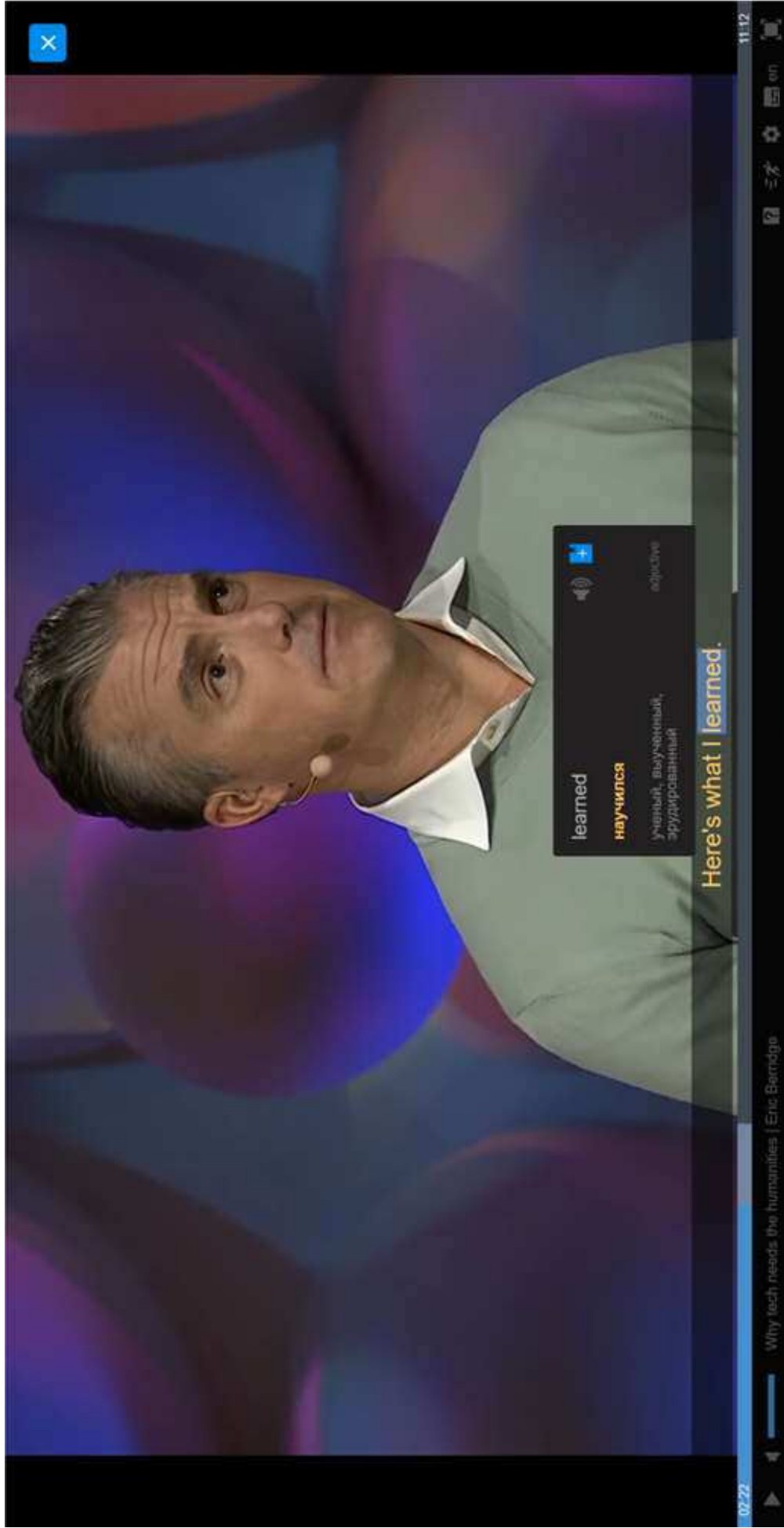


Рисунок 3 – Сервис «огого.tv»

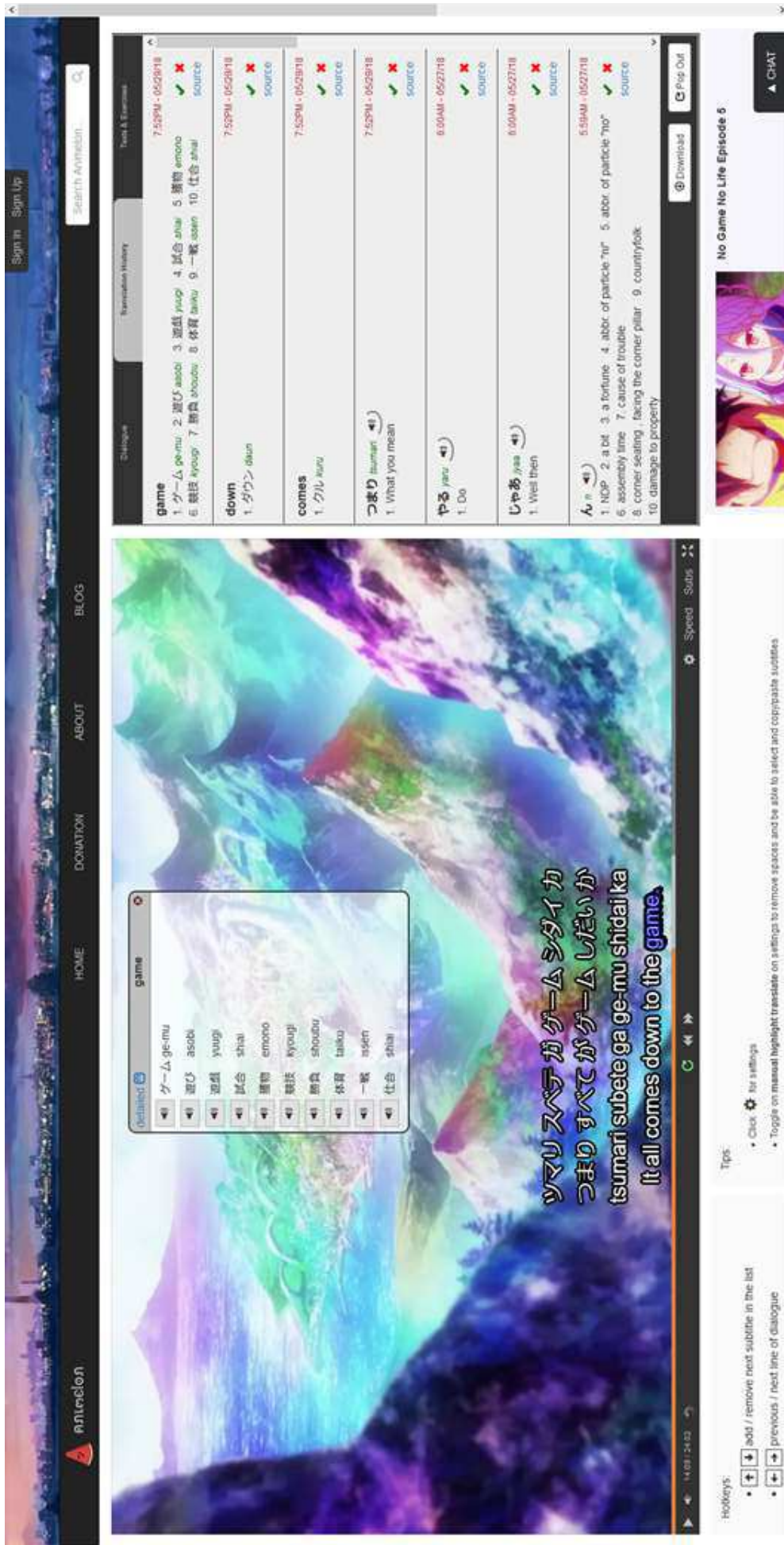


Рисунок 4 – сервис «animelon.com»

2.2 Проектирование сервиса

В данном разделе будут рассмотрены преимущества сервиса «EduJetPlayer», а также шаги, которые мы предприняли при его проектировании.

Благодаря тому, что наш сервис не хранит видеоматериалы, существенно снижается нагрузка на сервер, таким образом, потребности сервера может обеспечить даже бесплатный хостинг, также это позволяет пользователям запускать свои видеофайлы без рекламы и не платить за подписку онлайн кинотеатрам и другим сервисам для просмотра видео.

Нашей целевой аудиторией являются люди, заинтересованные в изучении иностранных языков. Концепция нашего сервиса в помощи обучению позволяет с пользой проводить время за просмотром новых фильмов и сериалов, что может заинтересовать и того, кто не собирался в серьёз заниматься изучением какого-либо языка, если его беспокоит количество бесцельно проведенного времени за просмотром сериалов.

Наш сервис будет распространяться в виде открытого исходного кода, что позволит сторонним разработчикам без труда подключить новый модуль из прт [18] и расширить функционал.

Так же наш сервис имеет интуитивно понятный интерфейс, который позволяет пользователям быстро начать работать, не задумываясь о сложной настройке каких-либо параметров, также есть видео инструкция с демонстрацией возможностей сервиса.

Для моделирования вариантов использования и графического описания взаимодействия модулей мы выбрали язык UML [19]. На рисунке 5 представлена схема вариантов использования нашего сервиса со стороны пользователя. Участие в сервисе администратора, модераторов и прочих управляющих должностей в сервисе не предусмотрено за ненадобностью. Пользователь для работы с сервисом должен зарегистрироваться либо войти в неё. После авторизации ему становятся доступны следующие действия: загрузить видео с субтитрами для просмотра, сгенерировать коллекцию Anki [20]. После загрузки

видео пользователь продолжает движение в основном потоке действий, и может выбрать непонятные субтитры для изучения (путем генерации Anki файлов).

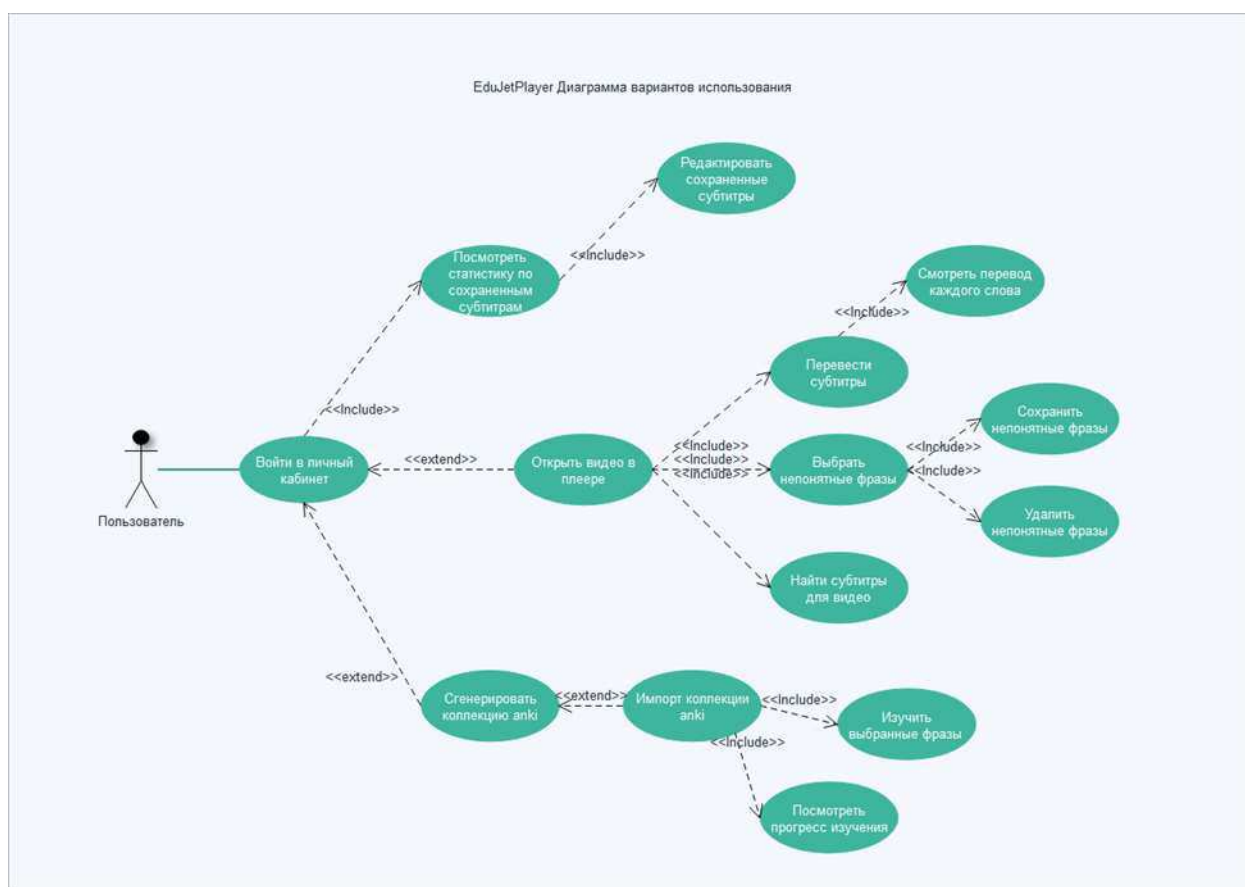


Рисунок 5 – Диаграмма вариантов использования

На рисунке 6 указана схема бизнес-процессов. С начала пользователь указывает видео, которое хочет посмотреть. Сейчас в проекте реализована загрузка видео и субтитров с компьютера пользователя. Затем у пользователя есть возможность выбрать фразу либо отдельное слово, которое пользователю неясно. Обработка полученных материалов включает в себя перевод при помощи API google-translator, а также генерацию карточек Anki, что позволяет изучать иностранный язык по методу super memo [21].

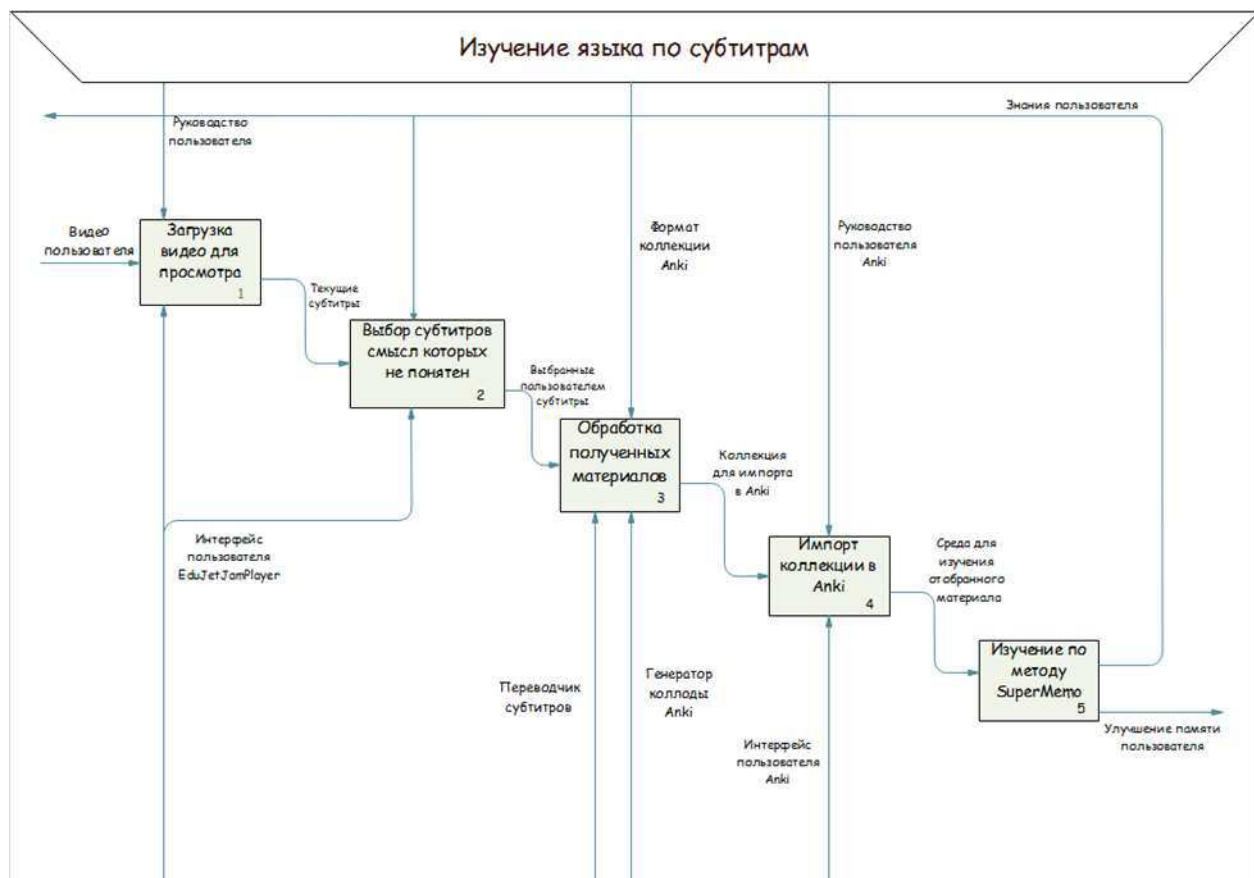


Рисунок 6 – IDEF0 схема бизнес-процессов

Что бы выбрать один из представленных в первой главе языков программирования мы составили таблицу для их анализа (Таблица 1).

Таблица 1 – Отрицательные качества языков PHP, Ruby и Python.

PHP	Ruby	Python
Имеет посредственную обработку ошибок	Сложен в изучении	Ограничен на слое доступа к БД
Глобальные параметры конфигурации могут менять семантику языка, усложняя разработку и перенос на новые платформы	Большее время вычисления по сравнению с остальными языками программирования	Не имеет полной поддержки многоядерных вычислений
Не подходит для создания настольных приложений	Мало информационных ресурсов	Не имеет коммерческой поддержки
Имеет меньшую защиту в сравнении с остальными языками	Медленно обновляется и разрабатывается	Обыкновенно медленней чем Java

В таблице указаны качества, которые нас не устраивали в конкретных языках, в следствие чего мы приняли решение остановится на языке Javascript для серверной разработки. Сервер для нашего сервиса будет работать на фреймворке NodeJS [22], так как это единственный серверный фреймворк для выбранного нами языка программирования, который, однако, удовлетворяет нашим требованиям, как показано в первой главе.

На рисунке 7 показан пример кода, создающего и запускающего HTTP-сервер на Node.js, который выдает «Hello, world!»

```
// Загружаем модуль http
const http = require('http');

// Создаем web-сервер с обработчиком запросов
const server = http.createServer((req, res) => {
  console.log('Начало обработки запроса');
  // Передаем код ответа и http-заголовки
  res.writeHead(200, {
    'Content-Type': 'text/plain; charset=UTF-8'
  });
  res.end('Hello world!');
});

// Запускаем web-сервер
server.listen(2002, '127.0.0.1', () => {
  console.log('Сервер запущен http://127.0.0.1:2002/');
});
```

Рисунок 7 – пример кода сервера.

В качестве базы данных была выбрана не реляционная база данных MongoDB [23], преимущества которой описаны в первой главе.

Что бы выбрать среду программирования сервера из рассмотренных в первой главе инструментов разработки мы провели сравнительный анализ, показанный в таблице 2. И исходя из приведенного сравнения выбор остановился на WebStorm.

Таблица 2 – Сравнительный анализ WebStorm и Notepad++

WebStorm	Notepad++
Имеет бесплатную лицензию для студентов	Открытый исходный код
Рефакторинг и проверка на ошибки	Не поддерживает отладку
Поддержка VCS	Не имеет поддержки VCS
Анализ кода	Подсветка синтаксиса

Теперь рассмотрим проектирование видео плеера. Так как пользователю потребуется возможность сохранять перевод функционал интерфейса был реализован следующим образом: после выбора необходимого видео и субтитров пользователь находит их по мере просмотра видео и может воспользоваться интерактивным окном, на котором, при нажатии, фраза целиком отправляется в модуль перевода. Если пользователю потребуется только перевод конкретного слова, то при нажатии на слово оно будет переведено с возможными синонимами.

Сам видеоплеер был разработан, используя web-технологии (HTML5 [24], JS, CSS)

Выбор Video.js обоснован тем, что:

- осуществима возможность разработки видеоплеера, воспроизводящего видео с различными видами субтитров и источников видео (удаленный, локальный);

- пользователь получает возможность самостоятельно выбирать слова или фразу из субтитров во время воспроизведения видео для их дальнейшего перевода. В данной дипломной работе применялись дополнения, для расширения функционала, такие, как Gulp Tasks v0.0.12, которое позволяет визуализировать и выполнять задачи gulp, найденные в рабочем пространстве.

Мы провели анализ требований, сформировали диаграмму вариантов использования и схему бизнес-процессов, что позволило увидеть полную картину принципа работы нашего сервиса.

2.3 Архитектура сервиса

Для разработки сервиса была использована архитектура клиент-серверного ПО:

а) онлайн плеер:

- 1) воспроизведение видео пользователя;
- 2) интерфейс управления воспроизведением;
- 3) сохранение субтитров для изучения;

б) сервер:

- 4) перевод субтитров;
- 5) генерация коллекции Anki из сохраненных субтитров;
- 6) генерация рекомендуемых к изучению фраз на основании статистики пользователя;
- 7) работа с базой данных;

в) личный кабинет:

- 8) отображение статистики по сохраненным субтитрам;
- 9) интерфейс загрузки сгенерированной коллекции Anki;
- 10) отображение рекомендуемых к изучению фраз;

г) база данных:

- 11) хранение данных сайта;
- 12) хранение данных пользователя.

Приложение делится на несколько компонентов, каждый из которых выполняет свою часть работы. Сервер управляет работой остальных компонентов, осуществляет работу с базой данных (обеспечивая сохранность данных). Личный кабинет и онлайн плеер выступают в качестве интерфейса для общения с сервером и в свою очередь базой данных. На рисунке 8 представлена диаграмма компонентов.

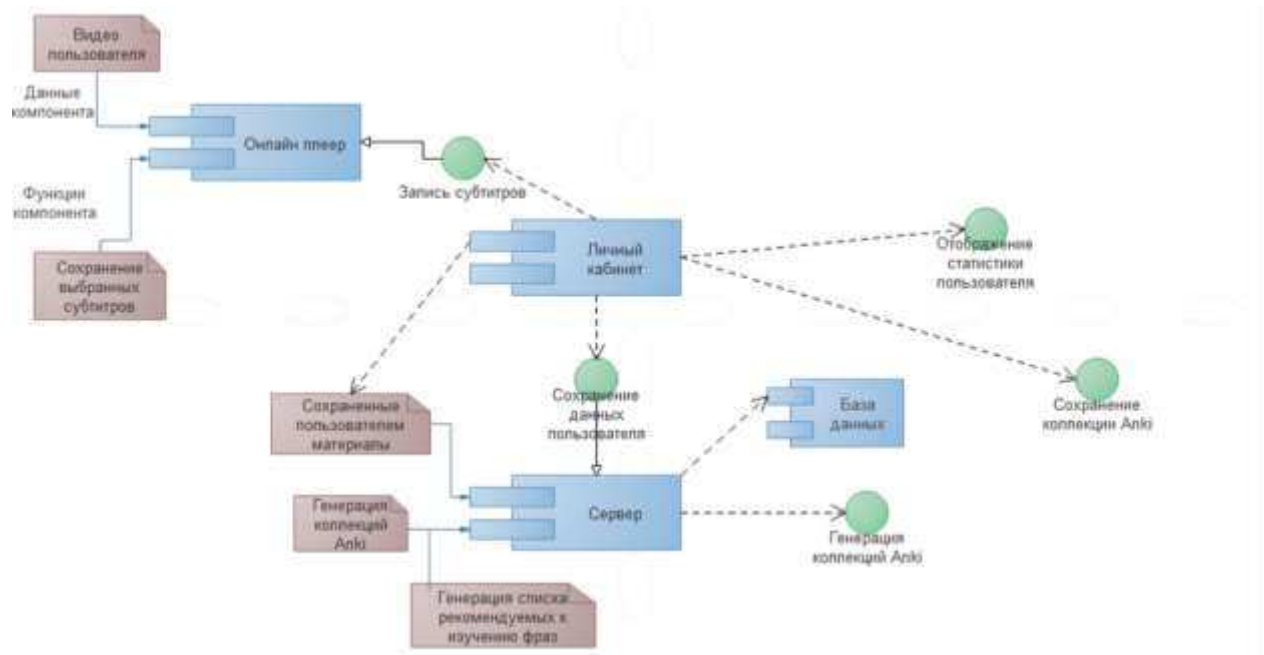


Рисунок 8 – Диаграмма компонентов «EduJetPlayer»

На рисунке 9 показан макет интерфейса нашего сервиса, состоящий из четырех панелей:

- панель 1 – навигационная часть интерфейса;
- панель 2 – панель выбора источников;
- панель 3 – видеоплеер;
- панель 4 – сохраненные фразы.

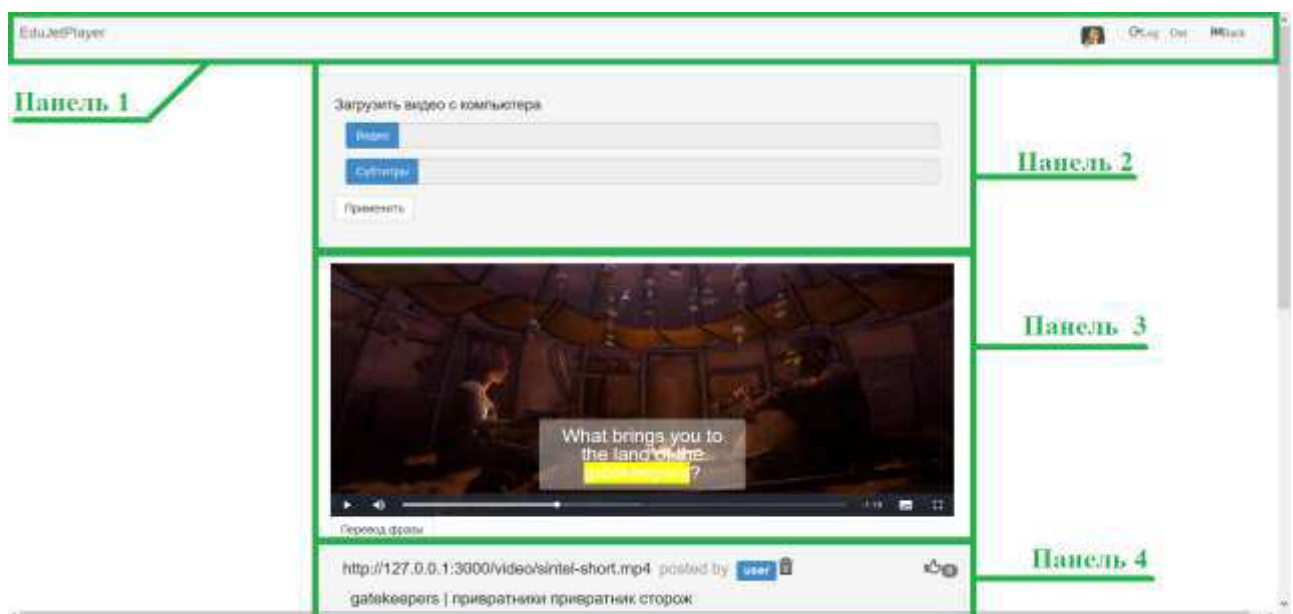


Рисунок 9 – Макет интерфейса

С помощью панели 1 осуществляется навигация по нашему сервису, слева присутствует кнопка перехода на главную страницу, а справа кнопка перехода в личный кабинет. Далее следуют кнопки завершения сеанса (либо входа и регистрации, если пользователь не зарегистрирован в сервисе) и кнопка перехода к предыдущей странице.

На панели 2 представлены две кнопки выбора видео для плеера, одна для выбора источника видеопотока, а другая для выбора файла субтитров. Ниже расположена кнопка «Применить» которая передает источники плееру для воспроизведения.

На панели 3 находится видеоплеер, в нижней его части присутствует панель управления с кнопкой начала либо остановки воспроизведения, слайдером, показывающим текущую позицию воспроизведения, временной меткой, кнопкой выбора субтитров и кнопкой перехода в полноэкранный режим.

Панель 4 представляет собой сохраненный пользователем перевод фразы. На панели хранится информация о источнике видео, пользователе который сохранил эту фразу, и сама фраза с переводом. Присутствуют кнопки удаления и увеличения рейтинга перевода (переводы сортируются по их рейтингу, назначенному пользователем).

3 Разработка сервиса «EduJetPlayer»

В данном разделе будут рассмотрены компоненты нашего сервиса, такие как: интерфейс, видеоплеер и серверное взаимодействие.

3.1 Командное взаимодействие

В ходе командной работы были использованы следующие инструменты:

- BitBucket [25],
- Trello [26].

Далее следует описание некоторых примеров использования вышеперечисленных инструментов в ходе разработки программного продукта «EduJetPlayer».

Проект был инициализирован в командном репозитории на хостинге BitBucket (рисунок 10). Интерфейс репозитория состоит из трех панелей:

- панель 1 – навигационная панель;
- панель 2 – описание проекта;
- панель 3 – файлы репозитория.

На панели 1 находятся название репозитория и его эмблема, а под ней 8 кнопок перехода к различному функционалу:

- код – страница с файлами репозитория, позволяет посмотреть, когда изменялись конкретные файлы в последний раз, а также увидеть комментарии к коммиту этих файлов;
- изменения – история коммитов, их слияния и ветвления;
- ветки – показывает все параллельные ветки кода;
- pull-запросы – запросы разработчиков на принятие их кода в ветку;
- pipelines – позволяет запускать поступающий в репозиторий код;
- загрузки – с этой страницы можно загрузить весь репозиторий;
- boards – доски инструмента для назначения задач Trello;
- настройки – настройки репозитория.

developed

developed **Панель 2** Invite Clone ...

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, [add a description to your repository](#).

master Filter files

Панель 3

Name	Size	Last commit	Message
.idea		2018-05-27	Загрузка нескольких подходящих субтитров с "горячей заменой"
.vscode		2018-05-27	Исправил баг со спанами на всю страницу. Исправил: перевод выделяет слово без точек...
assets		позавчера	Починил баг с переводом слов. Изза колбека слово не менялось, в отличии от перевода
build		2018-05-26	home page redesign
config		2018-02-11	First commit
models		позавчера	Починил баг с переводом слов. Изза колбека слово не менялось, в отличии от перевода
public		позавчера	Починил баг с переводом слов. Изза колбека слово не менялось, в отличии от перевода
routes		позавчера	Починил баг с переводом слов. Изза колбека слово не менялось, в отличии от перевода
.gitignore	41 B	2018-05-27	Исправил баг со спанами на всю страницу. Исправил: перевод выделяет слово без точек...
UnRAR.exe	361.21 KB	2018-04-17	merged files #2
deplist.txt	444 B	2018-02-11	First commit
developed.7z	65.41 MB	2018-03-25	added anki file gen

Панель 1

- developed
- Код
- Изменения
- Ветки
- Pull-запросы
- Pipelines
- Загрузки
- Boards
- Настройки

Рисунок 10 – Интерфейс репозитория проекта

На панели 2 находится никнейм разработчика, название репозитория и его краткое описание. Присутствуют такие опции как приглашение разработчика к участию в работе над данным репозиторием (так как по умолчанию все репозитории на платформе BitBucket приватные), копирование репозитория, выбор основной ветки кода и фильтр файлов.

На панели 3 присутствуют файлы и директории репозитория, в таблице с заголовками имя (name), размер (size), последний коммит (last commit), комментарий (message).

Была создана задача по изменению кода программы, которая затем была назначена на другого разработчика (рисунок 11).

Интерфейс создания задачи состоит следующих элементов:

- а) описание – подробное описание задачи;
- б) вложения – добавление вложений к задаче, таких как картинки, текстовые файлы и прочее;
- в) добавление комментария;
- г) действия:
 - 1) перемещение;
 - 2) копирование;
 - 3) подпись – подписка на оповещения о статусе задачи;
 - 4) архивация;
- д) добавить:
 - 1) участников;
 - 2) метки – цветные настраиваемые метки, например «Срочно»;
 - 3) чеклист – определенная последовательность действий, которую необходимо выполнить для решения задачи;
 - 4) срок – срок выполнения задачи;
 - 5) вложение;
- е) улучшения – синхронизация задачи с Bitbucket.

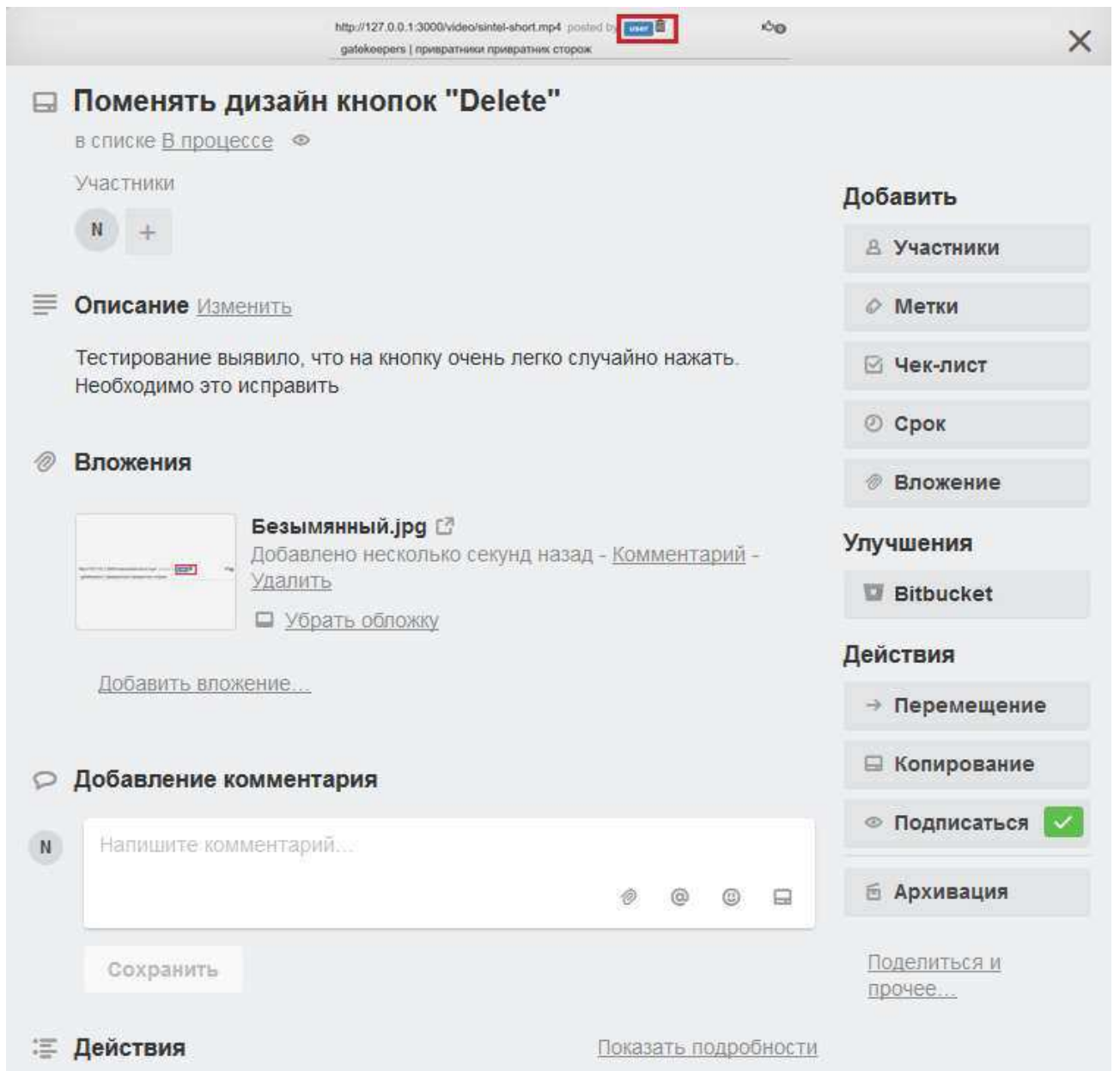


Рисунок 11 – Создание задачи

Затем созданную задачу мы переводим в доску «В процессе» и приступаем к выполнению задания (рисунок 12).

Нами были проведены необходимые изменения в соответствии с задачей (рисунок 13).

Anton Smith / developed

Boards BETA

Feedback ?

EduJetPlayer ☆

ООО "РазРабы" Free

Командная



Bitbucket

Нужно сделать

Реализовать прогнозирование слов
необходимых для понимания видео
по сохраненным в БД словам

Добавить карточку...

В процессе

Поменять дизайн кнопок "Delete"

Добавить карточку...

Готово

Переработать дизайн страницы
на новый

Добавить выбор языка перевода

Добавить подсветку выбранных
субтитров

Исправить баг с отображением
плеера

Сделать поиск субтитров по
пользовательскому запросу

Обновить пакет google-translator

Добавить поддержку видео
формата 16:9

Добавить отображение
сохраненных слов с цветом по
знанию слова

Добавить японских субтитров
для ковбой Бибопа

Добавить карточку...

Меню

Сменить фон

Фильтр карточек

 Улучшения 1

Календарь, Google Drive и другое...

Стикеры

Ещё

Действия

некоморф добавил(а) карточку
[Добавить японских субтитров для
ковбой Бибопа](#) в список Готово
несколько секунд назад

некоморф добавил(а) карточку
[Добавить отображение
сохраненных слов с цветом по
знанию слова](#) в список Готово
минуту назад

некоморф добавил(а) карточку
[Добавить поддержку видео
формата 16:9](#) в список Готово
минуту назад

некоморф добавил(а) карточку

Открыть в новом окне

Рисунок 12 – Интерфейс работы с задачами



Рисунок 13 – История коммитов

В систему контроля версий были внесены изменения, затем коммит был отправлен в удаленное хранилище (рисунок 14). В левом верхнем углу интерфейса создания коммита можно увидеть список измененных файлов и выбрать только нужные для коммита, также при выборе файла из этого списка. В нижней части интерфейса будет отображен список измененных строк, в которых можно выбрать, какие строки оставить, а какие отменить. В центре есть поле Commit message в которое пишется описание коммита и спец фразы, которые могут автоматически закрывать issue или менять их статус. В нижней части интрефейса есть синяя кнопка commit, при её нажатии все изменения сохраняются в локальный git. Чтобы отправить изменения в удаленный репозиторий, необходимо нажать на стрелочку на этой кнопке и выбрать опцию commit and push.

В правой части рисунка представлены следующие опции панели Git:

- автор;
- исправить коммит;
- отменить коммит.

Панель «До коммита»:

- переформатировать код;
- реорганизовать код;
- оптимизировать импорт;
- анализировать код;
- найти комментарии TODO;
- произвести очистку.

Панель «После коммита»:

- загрузить файлы;
- всегда использовать выбранный сервер.

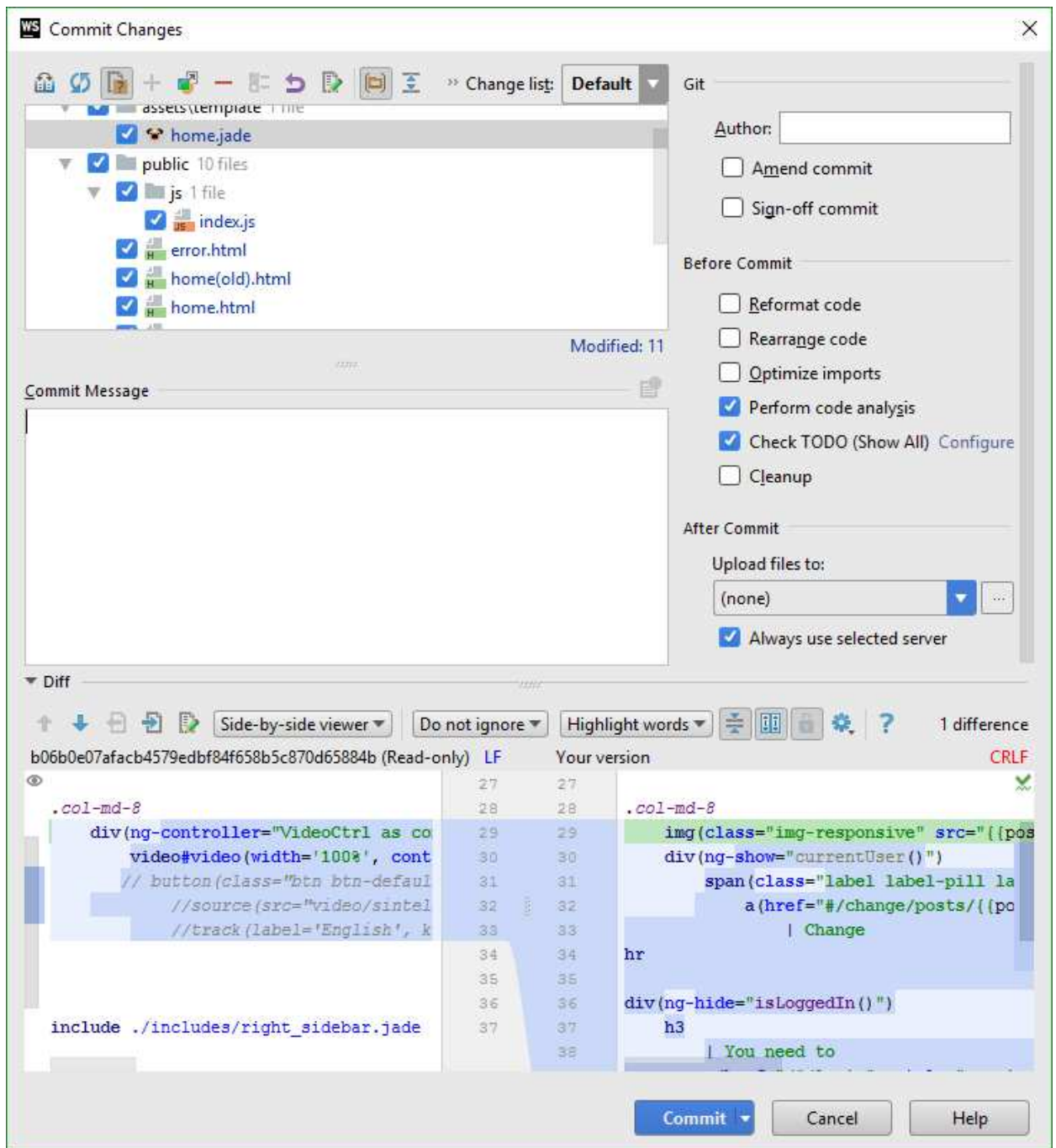


Рисунок 14 – Интерфейс создания коммита в WebStorm

Тикет был переведён на инициатора для выполнения проверки (рисунок 15). Внесённые изменения были проверены и выполнение тикета было подтверждено (рисунок 16).

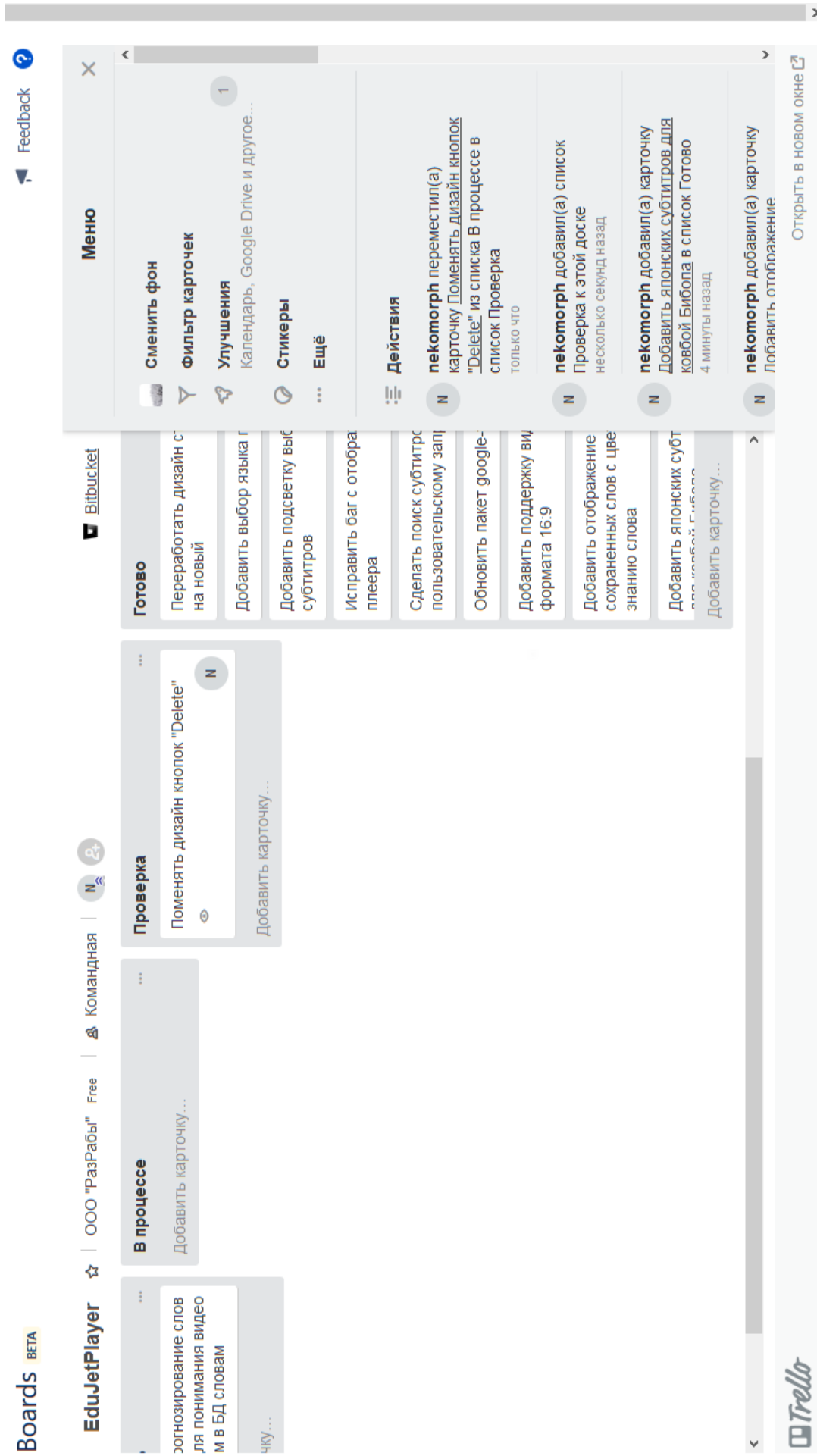


Рисунок 15 – Работа с задачей

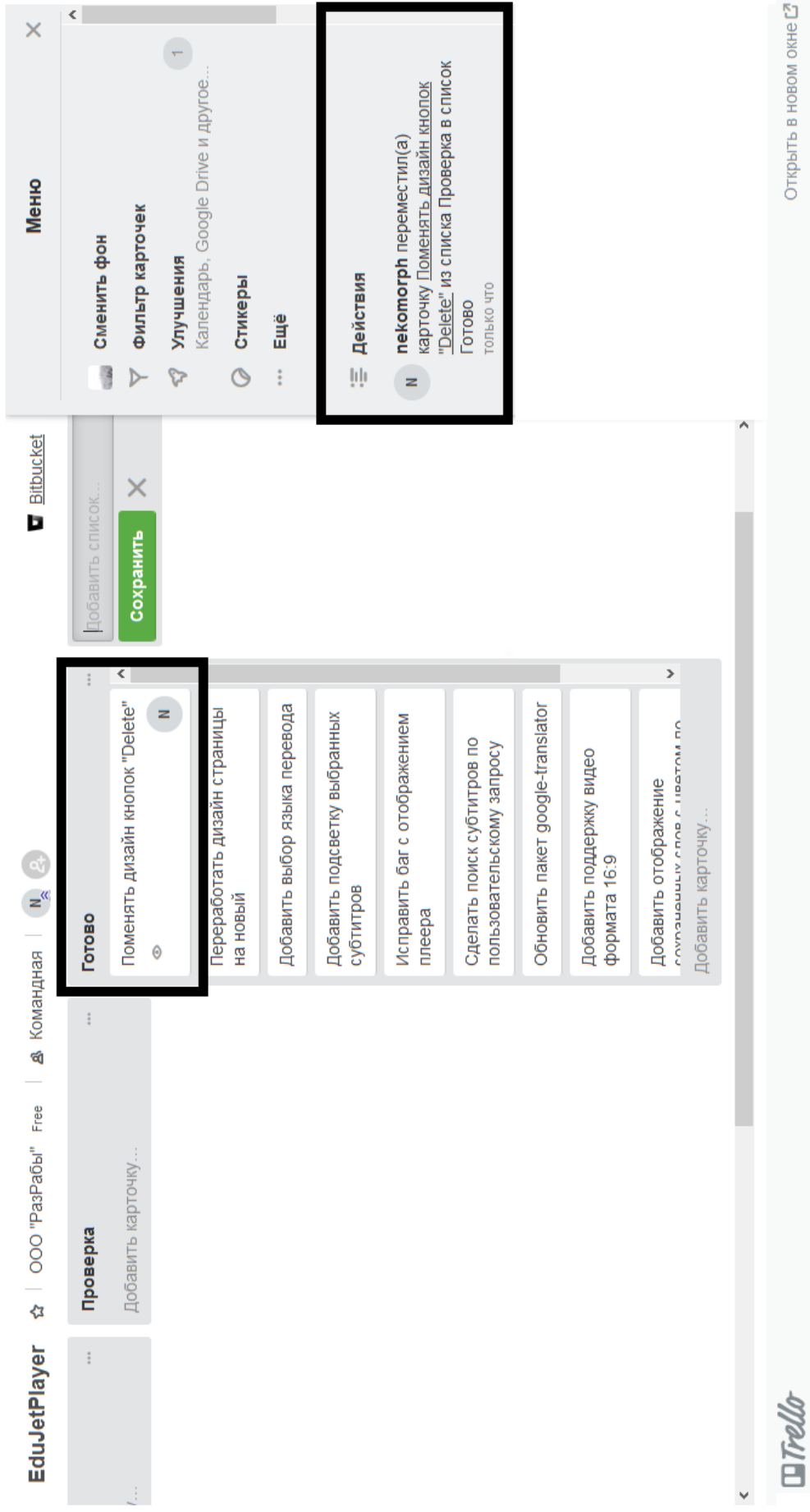


Рисунок 16 – Закрытие задачи

3.2 Реализованный функционал

В данном разделе будет описана разработка:

- основного функционала видеоплеера;
- загрузки видео и субтитров с устройства пользователя;
- дизайна страницы;
- перевода субтитров по слову или фразе;
- перевода субтитров с неизвестного языка;
- генерации коллекции Anki;
- импорта коллекции Anki в базу данных;
- поиска субтитров со стороннего ресурса;
- разархивация субтитров;
- поиска по релевантности;
- конвертации субтитров.

Рассмотрим компоненты видеоплеера и их функционал:

а) онлайн плеер:

- 1) воспроизведение видео пользователя;
- 2) интерфейс управления воспроизведением (полноэкранный режим, громкость);
- 3) отправка субтитров на перевод (слово, фраза);

б) сервер:

- 1) перевод по слову или фразе;
- 2) перевод с неизвестного языка.

На рисунке 17 показано окно интерфейса видеоплеера с субтитром (внизу) и его переводом (сверху):

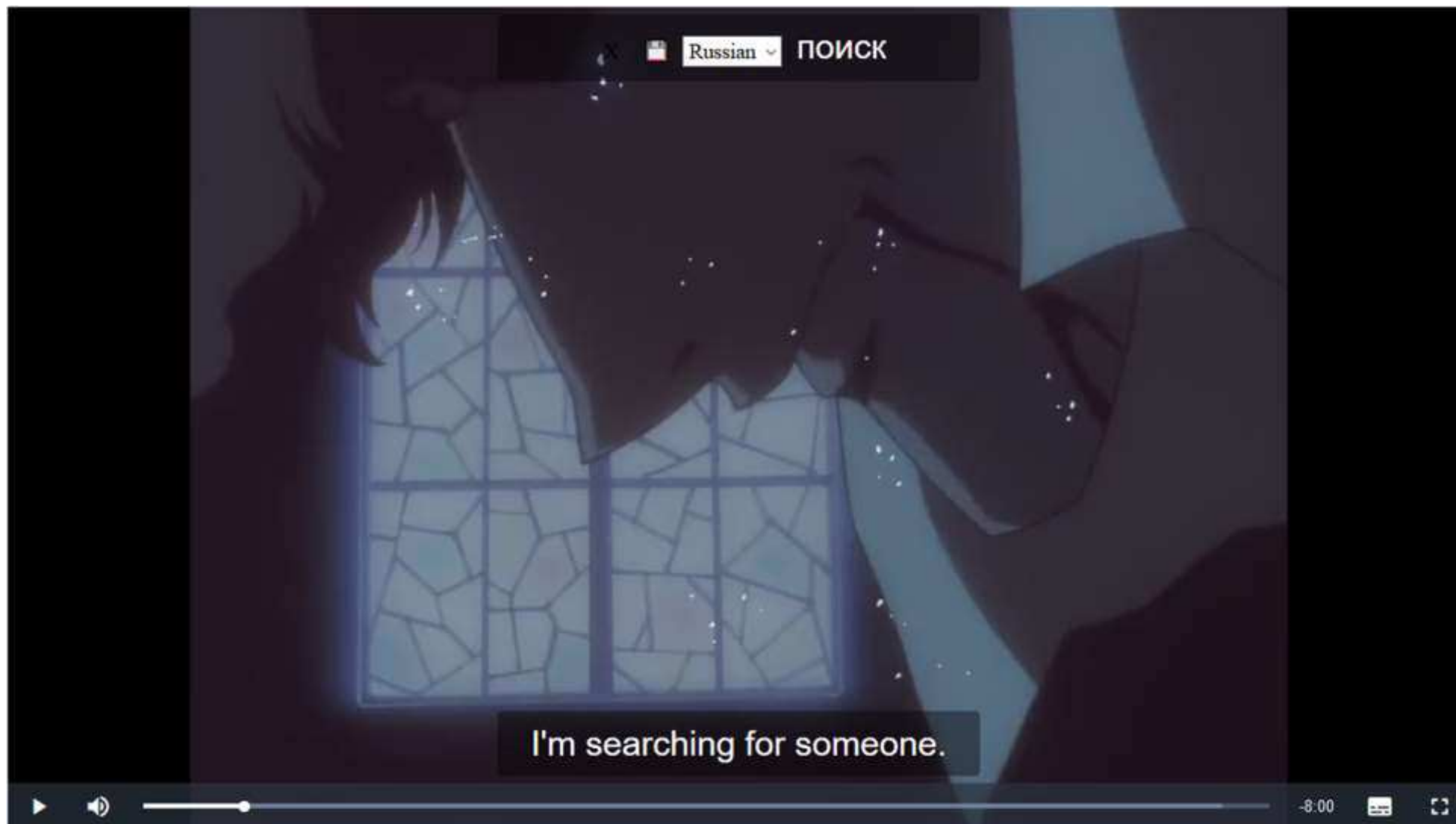


Рисунок 17 – Функционал видеоплеера

На рисунке 18 показано окно видеоплеера при первом посещении страницы.

Начать воспроизведение видео можно без выбора источника видеофайла (по умолчанию) нажав на прямоугольную кнопку со скруглёнными краями и треугольным символом по центру.

После нажатия кнопки пользователь сможет увидеть субтитры сразу либо спустя некоторое время (где предусмотрено время их появления) и воспользоваться функцией перевода (рисунок 19 и 20).

Пользователь может отправить субтитры на перевод нажатием на затемнённую область вокруг субтитров или перевести слово с его синонимами нажатием на него.

Для воспроизведения видео пользователь может выбрать видеофайл и файл субтитров для воспроизведения. Чтобы выбрать файл пользователь должен нажать на соответствующие кнопки вверху страницы (рисунок 20) и после нажатия кнопки применить видео будет практически сразу доступно для просмотра.

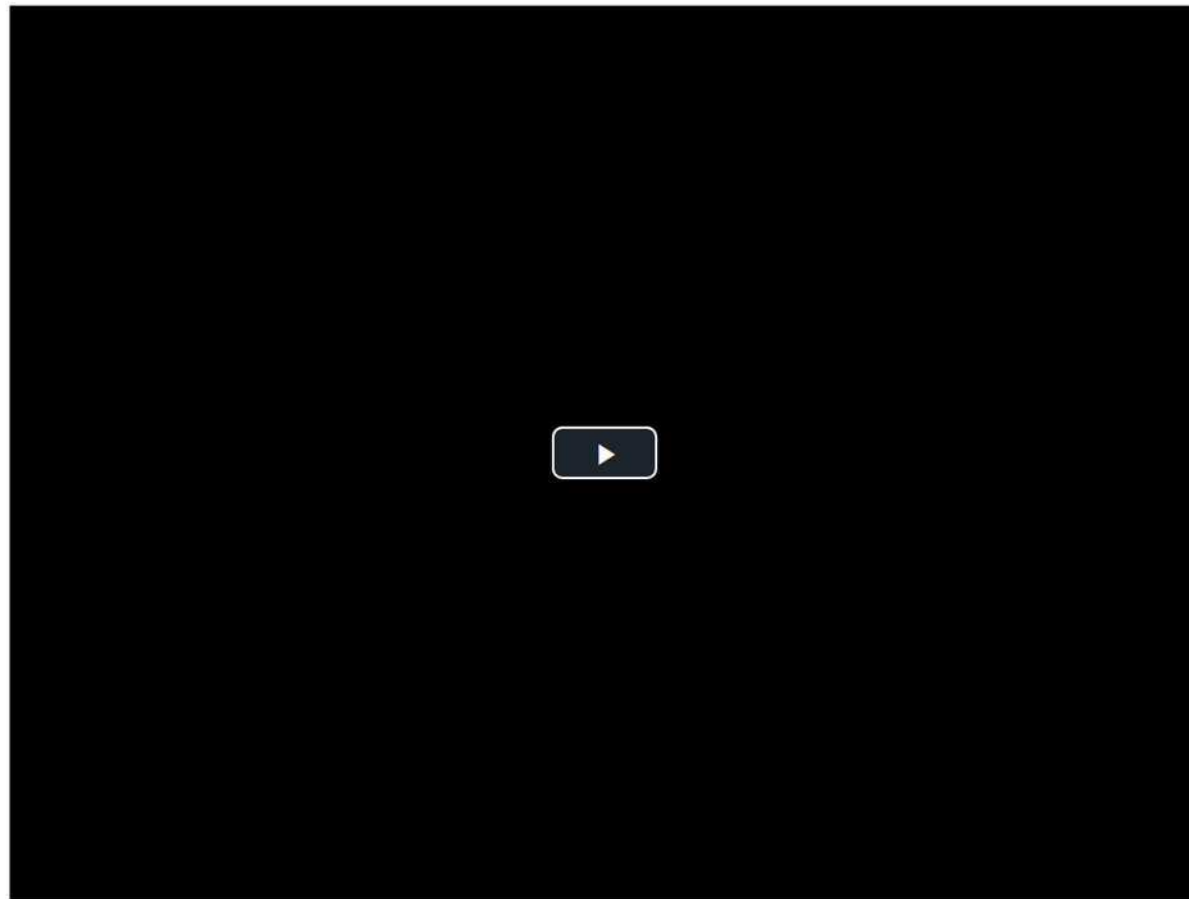
Планируется в будущем добавить функцию воспроизведения видео не только с локальных носителей информации, но и онлайн источников. Пользователь может выбирать с какого момента воспроизводить видео используя временную шкалу внизу по центру.

Останавливать или продолжать воспроизведение видео используя треугольную кнопку слева внизу. И открывать видео на полный экран кнопкой справа внизу.

После нажатия на затемнённую область вокруг субтитров они отправляются для перевода на сервер, а, чтобы не загромождать окно видеоплеера аннотациями, перевод будет выведен сразу сверху страницы, не прерывая воспроизведения видео.

Загрузить видео с компьютера

Выбор субтитров



[<< Prev](#) [Next >>](#)

You need to [Log In](#) or [Register](#) before you can add a post.

Рисунок 18 – Видеоплеер на сайте

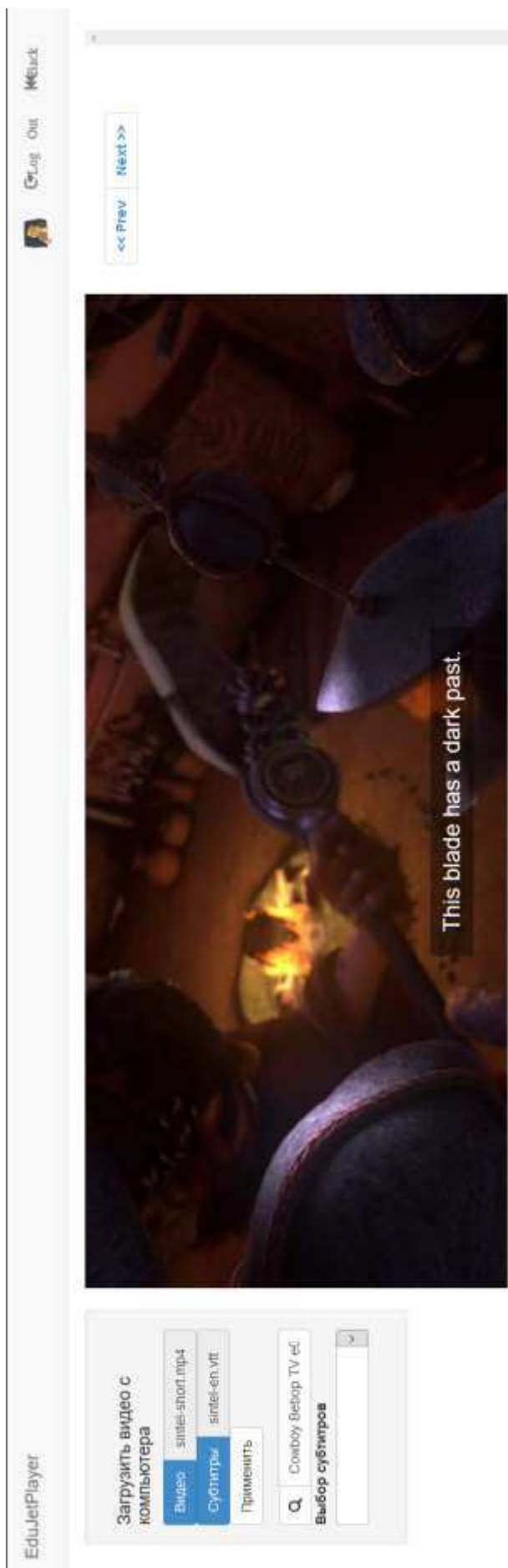


Рисунок 19 – Окно с субтитрами

EduJetPlayer Log Out Back

Загрузить видео с компьютера

Видео: sintel-short.mp4

Субтитры: sintel-en.vtt

Применить

🔍 Cowboy Bebop TV еС

Выбор субтитров

Russian Вы - дурак, путешествующий один, так совершенно неподготовленный.

You're a fool for traveling alone, so completely unprepared.

blob:http://localhost:3000/6f90dab4-a088-49ba-b2e7-2e88ea3adfe3 pos

by admin

This blade has a dark past. | У этого клинка темное прошлое.

blob:http://localhost:3000/6f90dab4-a088-49ba-b2e7-2e88ea3adfe3 pos

by admin

fool | дурачить дурак
глупец дура шут
кисель

blob:http://localhost:3000/6f90dab4-a088-49ba-b2e7-2e88ea3adfe3 pos

by admin

You're a fool for traveling alone, so completely unprepared. | Вы -

Рисунок 20 – Переведённые субтитры

С помощью backend-технологии Node.js была добавлена возможность JavaScript-коду взаимодействовать с основным сайтом через свой API (написанный на C++), подключив внешнюю библиотеку google-translator (рисунок 21), обеспечивая вызовы к ней из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера. В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом.

(google-translator) Google Translator API for FREE

Translate with google for FREE without API key

Installation

```
npm install google-translator
```

Usage

```
var translator = require('google-translator');

/*
  Translate 'Hello'
  from   :   English
  to     :   French
*/

translator('en', 'fr', 'Hello', response => {
  console.log(response);
});
```

Рисунок 21 – GitHub проект google-translator

Если воспользоваться интерактивным окном, на котором, при нажатии, фраза целиком отправляется в модуль перевода, то сервер, получив запрос в текстовом виде, ретранслирует его модулю google-translator, а в конце translate.google.com.

Если слово в субтитре одно, то к переведенной фразе добавятся синонимы на родном языке.

Если пользователю неизвестен язык субтитров, то он всё равно может воспользоваться модулем перевода: язык подберётся автоматически. Если же язык не был определён, то фраза вернётся на сайт в исходном виде.

Нами было визуально обновлено окно вывода субтитров так, что при наведении на него оно изменяло цвет для его интерактивности, но оно было незаметно, что было скорректировано в коммите. Так же было визуально обновлено окно вывода субтитров так, что при наведении на него оно изменяло цвет для его интерактивности.

Во время тестирования нами было замечено, что сайт отображается не так как должен. Нами было замечено, что если убрать рекурсивную зависимость, то код нашего сервиса продолжал выполняться без ошибок.

Эта ошибка возникает, когда «\$injector» пытается получить функцию, которая зависит от самой себя, прямо или косвенно. Чтобы исправить это, было необходимо создать такую цепочку зависимостей, чтобы не было круговой.

Например (рисунок 22), когда создается экземпляр MyCtrl, служба myService будет создана инжектором myService с зависимостью от самого себя, что заставляет «\$injector» обнаруживать циклическую зависимость и выдавать ошибку:

```
angular.module('myApp', [])
.factory('myService', function (myService) {
  // ...
})
.controller('MyCtrl', function ($scope, myService) {
  // ...
});
```

Рисунок 22 – Пример круговой зависимости

Нами был добавлен video.js – основной плагин для воспроизведения видео – который, по неизвестным на тот момент причинам, не мог изменить источник воспроизведения видео.

Как только код плеера был встроен нами на сайт, то были оставлены файлы «package.json», которые при выполнении команды «npm install» сохраняли в папке сайта ненужные файлы в большом количестве. Была добавлена рекомендация – не воспроизводить команду.

Дальнейшее тестирование показало, что перевод по слову работает не так как должен. Нами было замечено, что если фраза была написана с опечаткой, то наш сервис возвращал исправленный вариант, а не перевод. Данная ошибка была исправлена. Нами был изменён участок кода, вызывающий рекурсивную зависимость «auth – users – auth» ещё раз для возврата прежнего функционала сервиса. А перевод по слову был назначен через подписку на событие видеоплеера «.on("play", ...)» таким образом, что при нажатии на слово оно отправлялось на сервер для перевода и возвращалось в виде поста на основном сайте сервиса.

Нами была добавлена возможность перевода фразы путём отключения окна вывода субтитров и добавления своего окна с целью внесения функциональных и визуальных изменений (подсвечивание слов при нажатии, отправка слов на сервер). Мы провели процедуру ручного редактирования кода, после автоматической процедуры слияния кода средствами Git, так как возникали ошибки автоматического слияния. Произошло это потому, что система Git не могла разобраться с тем, какой код необходимо оставить, а какой убрать.

Так как перевод по слову был назначен нами через подписку на событие видеоплеера «.on("play", ...)» таким образом, что при нажатии на слово оно отправлялось на сервер для перевода и возвращалось в виде поста на основном сайте нашего сервиса, то мы решили не загромождать пространство постов, которое необходимо для сохранения фраз. Мы решили выводить их в отдельном окне, а сохранить переведённое уже по требованию пользователя, через кнопку в окне.

Само окно перевода было добавлено нами для выбора языка (по умолчанию русский) и последующим переводом в отдельном пространстве окна вверху видеоплеера (рисунок 23). На рисунке видно кнопку сохранения перевода и выбора языка.

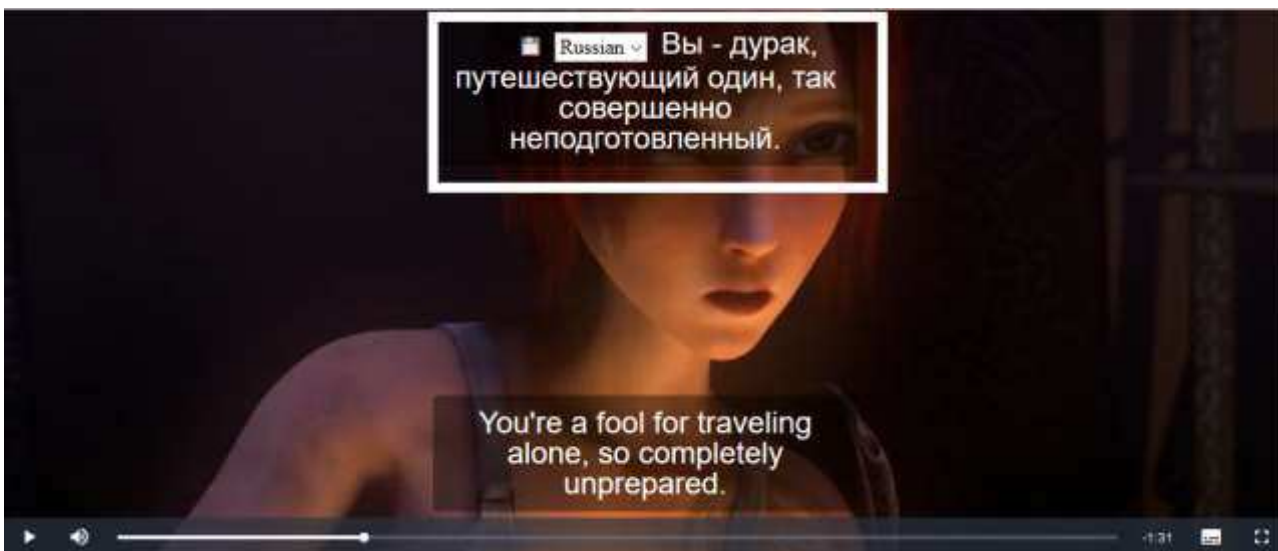


Рисунок 23 – Окно перевода и выбора языка

Рассмотрим реализованные функции исходного кода с комментариями:

- function trans(el, _text, _lang) – передача фразы для перевода на сервер, возвращает переведённый текст в элемент DOM, если сервер отправил ответ. Иначе остаётся сообщение «Переводим...»;
- function languages(el) – наполняет DOM элемент «select» данными о поддерживаемых на данный момент языках с сервера;
- function getSelectedText() – возвращает выделенный на странице текст;
- function changeWidth(e) – задаёт ширину DOM элемента «select» на ширину выбранного внутри текста;
- function showPopup() – уникальная функция, задающая всем DOM элементам с классом «tooltip1» классы оверлея субтитров;
- function popup(el, text) – иницирует элемент оверлея и наполняет его переведённым текстом;
- function popdown() – скрывает элемент;
- var subtran = function (el, _lang) – переводит фразу с указанным языком через сервер;
- var updateTracks = function(data) – создаёт интерактивные субтитры из имеющихся;
- \$scope.addSub = function () – сохраняет фразу на сайте;

- `$scope.$on('vjsVideoReady', function (e, data) { ... })` – событие, которое возникает, когда плеер начинает воспроизводить видео;

- `$scope.changedProgress = function(event, options)` – событие: при выборе anki-файла распаковывает его, и отправляет необходимые данные на сервер, чтобы сохранить пользовательскую информацию в базе данных;

- `function getUnknownWords()` – выгрузка малоизученных слов в anki-файл.

Нами был изменён код, отображающий субтитры таким образом, что при открытии страницы нашего сервиса текст соответственно увеличивался, и наоборот при уменьшении страницы.

Также нами был исправлен баг вызывающий изменение встроенного окна субтитров на собственные: при нажатии на кнопку воспроизведения, функция вызывается каждый раз, но теперь однократно, при замене видеоисточника.

Сохраненные пользователем переводы слов и фраз хранились в отдельных документах шаблона «Post». Он хранил название видео, текст и его перевод.

Для того что бы реализовать импорт сохраненных фраз в программу Anki сервер формирует все сохраненные переводы в следующий формат: «фраза ; перевод фразы» и записывает построчно в файл, который сохраняет во временную директорию с помощью ранее указанного модуля fs-extra. Затем сервер отправляет пользователю этот файл для скачивания с помощью сервиса angular-file-saver. Для загрузки коллекции Anki пользователю необходимо в личном кабинете нажать на кнопку «download Anki» которую можно заметить на рисунке 24.

url of avatar

Avatar

Change avatar

download Anki

Рисунок 24 – Личный кабинет пользователя.

Сервер формирует все сохраненные переводы в формат пригодный для интеграции с программой Anki и отправляет пользователю файл для скачивания. Этот файл для использования необходимо импортировать в программу для работы с Anki файлами.

Что бы сервер мог находить субтитры по пользовательскому запросу нами был реализован следующий функционал: сервер обращается к площадке с субтитрами с запросом пользователя, обрабатывает выдачу с площадки и скачивает во временную директорию архивы с субтитрами.

На рисунке 25 представлен фрагмент кода обработки поисковой выдачи, где видно, что сервер обращается к площадке с субтитрами (например, <http://subs.com.ru>), и пользуясь системой поиска при помощи модуля request получает страницу с поисковой выдачей, которая обрабатывается при помощи модуля cheerio. Все элементы страницы, которые могут содержать ссылку на страницу с архивом отбираются, и осуществляется переход по этим ссылкам. Далее выдача так же обрабатывается, и находится ссылка собственно на файл с субтитрами.

```
router.get('/caption', function(req, res, next) {
  let _name = req.query.name;

  var URL = 'http://subs.com.ru/index.php?e=search&sq='+_name;

  request(URL, function(error, response, body) {
    if(error) { return console.error('There was an error!'); }

    var $ = cheerio.load(body);
    var i=0;
    $('a').each(function() {
      var name = $(this).text();
      var link = $(this).attr('href');

      if(link.includes('highlight')) {
        link = "http://subs.com.ru/" + link.substring(0, link.indexOf('&'));
        request(link, function(error, response, body) {
          if(error) { return console.error('There was an error!'); }

          var banana = cheerio.load(body);

          banana('td[class=even]').each(function() {

            var text = banana(this).text();
```

Рисунок 25 – Фрагмент кода обработки поисковой выдачи.

Извлечение субтитров из архива проводится с помощью модуля `node-7zip` [27], который является оболочкой для библиотеки архиватора `7zip`, с помощью которого можно извлекать файлы из таких архивов как `.rar` и `.zip`. После того, как загруженные субтитры извлечены, по файлам субтитров проводится релевантный поиск с целью наилучшего совпадения результата пользовательскому запросу. Данная операция выполняется с помощью модуля `relevancy` [28]. Этот модуль осуществляет сортировку списка файлов по запросу с клиентской части нашего сервиса по определенным настройкам, показанным на рисунке 26.

```
let mySorter=relevancy;
mySorter.weights={matchInSubjectLength: 1,
  matchInSubjectIndex: 1,
  matchInValueLength: 0,
  matchInValueIndex: 0};
let tmp=mySorter.sort(results, _name).slice(0,10);
```

Рисунок 26 – Настройка релевантной сортировки

- `matchInSubjectLength` – (0-1) Доля наибольшего совпадения подстроки найденной в пределах фактического запроса. Итак, если объект «Gu», но мы только сопоставляем «G» (например, в «Grape»), тогда доля будет равна 0,5.

- `matchInSubjectIndex` – (0-1) Близость матча к началу темы. Например, учитывая предмет "A grape" против "Grapelicious", "grape" в "A grape" совпадает с началом значения, имеет индекс 2. Самый высокий возможный индекс этого значения равен 2 (учитывая длину совпадения: 5), поэтому `matchInSubjectIndex` в этом примере получает значение 0.

- `matchInValueLength` – (0-1) Доля наибольшего соответствия подстроки найденной в пределах целевого значения. Например, мы подобрали подстроку «King» (длина: 4) против значения «United Kingdom» (длина: 14) и $4/14 = \sim 0,29$.

- `matchInValueIndex` – (0-1) Близость совпадения к началу целевого значения. Например, совпадение «dom» в «Kingdom» – «dom» сопоставляется с

индексом 4. 4, делится на общую длину 7, вычитается из 1 и дает нам наш `matchValueIndex` примерно 0,43.

Субтитры очень часто хранятся в различных форматах, тогда как наш плеер может использовать только субтитры формата `.vtt`. Разрешена эта проблема была следующим образом: подключен модуль `subsr` [15], который предназначен для перекодирования субтитров в необходимый формат. Во время тестирования после подключения этого модуля в нем были найдены блокирующие ошибки трансляции нескольких форматов субтитров (таких как `.srt` и `.ass`). Пример исправленного кода приведен на рисунке 27.

```
var helper = {
  toMilliseconds: function(s) {
    var match = /^%s*(%d+)?(%d{1,2}):(%d{1,2})([.,]-?(%d{1,3}))?%s*$/ .exec(s);
    var hh = match[1] ? parseInt(match[1].replace(":", "")) : 0;
    var mm = parseInt(match[2]);
    var ss = parseInt(match[3]);
    var ff = match[5] ? parseInt(match[5]) : 0;
    var ms = hh * 3600 * 1000 + mm * 60 * 1000 + ss * 1000 + ff * 10;
    return ms;
  },
  toTimeString: function(ms) {
    var hh = Math.floor(ms / 1000 / 3600);
    var mm = Math.floor(ms / 1000 / 60 % 60);
    var ss = Math.floor(ms / 1000 % 60);
    var ff = Math.floor(ms % 1000 / 10); //2 digits
    var time = hh + ":" + (mm < 10 ? "0" : "") + mm + ":" + (ss < 10 ? "0" : "") + ss + "." + (ff < 10 ? "0" : "") + ff;
    return time;
  }
};
```

Рисунок 27 – Фрагмент исправленного кода преобразования времени из формата `.ass`

На рисунке на третьей строчке можно видеть регулярное выражение для определения, является ли строка с текстом закодированным временем. В нем автором была допущена ошибка, он не учитывал то, что в некоторых видах формата `.ass` время в миллисекундах может указываться в отрицательном виде, из-за чего вся процедура перекодировки прекращалась. Кроме того, в аналогичном регулярном выражении для преобразования времени уже в формате `.vtt` автор не добавил точку как вариант десятичного разделителя.

Разберем регулярное выражение, используемое в данном фрагменте кода:

«/^\s?(\d+)?(\d{1,2}):(\d{1,2})([.,]-?(\d{1,3}))?\s?\$/»

- «^» утверждает положение в начале строки;
- «\s» соответствует любому пробельному символу (равному [\ r \ n \ t \ f \ v]);
- «()» выражение в скобках соответствует определенной группе;
- «?» квантификатор - совпадение между нулем и одним разом, как можно больше раз, при необходимости возвращаясь;
- «\d» совпадает с цифрой (равной [0-9]);
- «+» квантификатор - Соответствует одному и неограниченному количеству раз, при необходимости возвращаясь;
- «:» соответствует символу «:» буквально (с учетом регистра);
- «{1,2}» Квантификатор - Соответствует совпадениям от одного до двух раз, при необходимости возвращаясь;
- «[]» Сопоставляет один символ, присутствующий в скобках [.,];
- «-» соответствует символу «-» буквально (с учетом регистра);
- «\$» утверждает позицию в конце строки.

В итоге регулярное выражение принимает строки вида «0:00:01:-120».

Сервер посылает один, самый релевантный файл с субтитрами на клиент, и плеер подключает его для воспроизведения.

Нами была добавлена функция импорта файлов изучения Anki, которая выводится в браузерной консоли после обработки (рисунок 28):

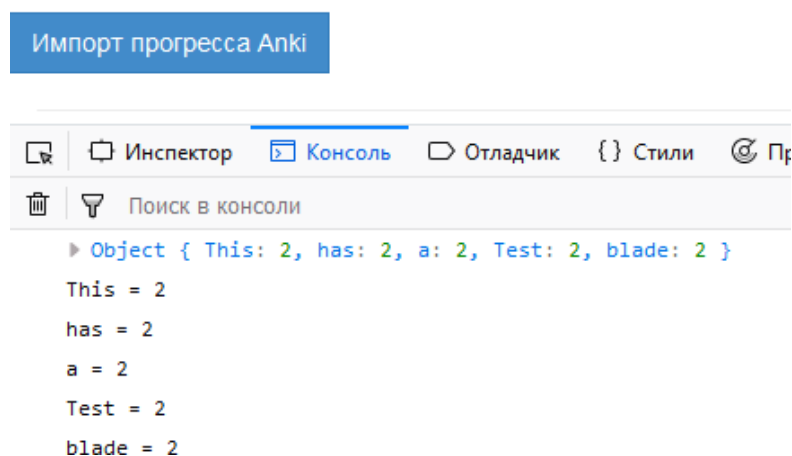


Рисунок 28 – Пример вывода в консоль браузера

Также нами были добавлены соответствующие библиотеки для работы с базой данных внутри файла импорта, и для распаковки файла. Так же были добавлены команды, которые помогали в разработке нашего сервиса, которые были удалены.

Мы, путём тестирования, выявили баг, который приводил окно с субтитрами в нерабочее состояние, также нами было исправлено регулярное выражение для правильного нахождения слов в субтитрах.

Сервер получает от клиента импортированную коллекцию `anki`, в которой указана степень изученности пользователем того или иного слова от 1 до 3, где 1 – изучено плохо, 3 – изучено отлично. Для каждого пользовательского профиля хранится список его изученных слов, с помощью которого осуществляется анализ субтитров для нового видео. Те слова, которые присутствуют в субтитрах, но не найдены в базе изученных слов переводятся и добавляются в колоду `anki` для изучения. Данная колода затем может быть запрошена пользователем для загрузки и последующего изучения. Чем чаще пользователь будет пользоваться данным функционалом, тем точнее будет прогноз нашего сервиса по неизвестным для пользователя словам.

Мы добавили кодирование пересылаемых имен файлов, так как в некоторых названиях субтитров содержались специальные переменные, которые необходимо было экранировать для нормального функционирования сервера.

Что бы решить эту проблему мы воспользовались функцией `encodeURIComponent`. Для оптимизации места выделяемого серверу мы добавили удаление скачанных архивов с субтитрами и сами субтитры, кроме перекодированных. В целях тестирования работы нашего сервиса с субтитрами на символьных языках были добавлены японские субтитры. Тестирование показало, что система разбивки субтитров на отдельные слова не может выполнять свою работу из-за особенностей строения предложений в японском языке, а именно отсутствие пробелов между разными словами.

Для удобства пользователя нами было решено предоставить ему выбор субтитров, поэтому мы внесли следующие изменения: для загрузки субтитров на

клиент сервер стал посылать список из десяти перекодированных файлов, что позволило пользователю нашего сервиса самому выбрать какие субтитры подключить. При выборе клиент запрашивает у сервера файл из списка. В планах дальнейшей разработки сделать поиск субтитров в первую очередь по уже сконвертированным субтитрам с нашего сервера, а затем уже в случае неудовлетворительного поиска прибегнуть к площадкам размещения субтитров. Подключение выбранных пользователем субтитров может осуществляется на ходу, то есть не нужно останавливать воспроизведение видео в плеере.

Уже конвертированные субтитры хранятся на нашем сервере, их текущий вид указан на рисунке 29.



Рисунок 29 – Конвертированные субтитры в файловой системе сервера

В дальнейшем планируется расширить число поддерживаемых площадок для поиска субтитров, добавление площадок с другими языками субтитров (японский, китайский, испанский), что позволит увеличить целевую аудиторию.

Также нами была добавлена система хранения неизученных пользователем слов, с целью анализа и формирования файлов для экспорта в программу Anki. Теперь слова неизвестные пользователю сами при переводе добавляются в базу с индексом наименьшей изученности. Всего индексов три, 1 – плохо известны, 2 – хорошо известны и 3 – известны отлично. Слова хранятся в документе пользователя в виде массива, со значениями слово, перевод и индекс изученности. Так же был добавлен механизм импорта в систему хранения слов

нашего сервиса из файлов программы Anki. Для того что бы добавить все слова, которые пользователь не отметил как изученные необходимо нажать на кнопку «добавить слова» находящуюся в левой части пользовательского интерфейса (рисунок 30).

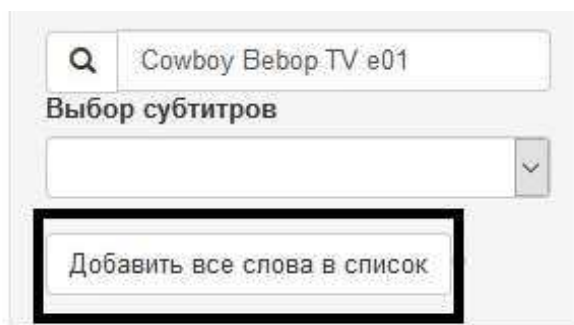


Рисунок 30 – Кнопка добавления слов

Первоначально для подключения модуля google-translator мы использовали его при репозиторий, но как оказалось он несвоевременно обновлялся и было принято решение подключать модуль напрямую из Github, это реализуется добавлением ссылки на репозиторий в package.json: «"google-translator": "github:731MY/google-translator"». Так же это дает возможность для копирования репозитория и последующего его редактирования, так например для модуля subsrt мы сделали собственную копию, чтобы исправить ошибки при конвертации некоторых типов субтитров.

На этом этапе нами был выполнен редизайн домашней страницы сервиса «EduJetPlayer». В прошлой версии дизайна не было сайд баров, а сохраненные субтитры отображались под плеером. Внешний вид старой версии дизайна предоставлен на рисунке 31. Как можно заметить, пользователю не видно сохраненного текста и для его просмотра приходится пролистывать страницу вниз. Так же сам проигрыватель находится не в центре экрана, а половину интерфейса занимает интерфейс для загрузки видео и субтитров. В то же время мы работали над модулем автоматической загрузки субтитров, поэтому нужно было решить каким образом расположить интерфейс для их поиска.

Мы добавили возможность воспроизведения видео и субтитров из разных источников в тестовом режиме, где ссылка на видео и ссылка с субтитрами может быть указана в левой части окна.

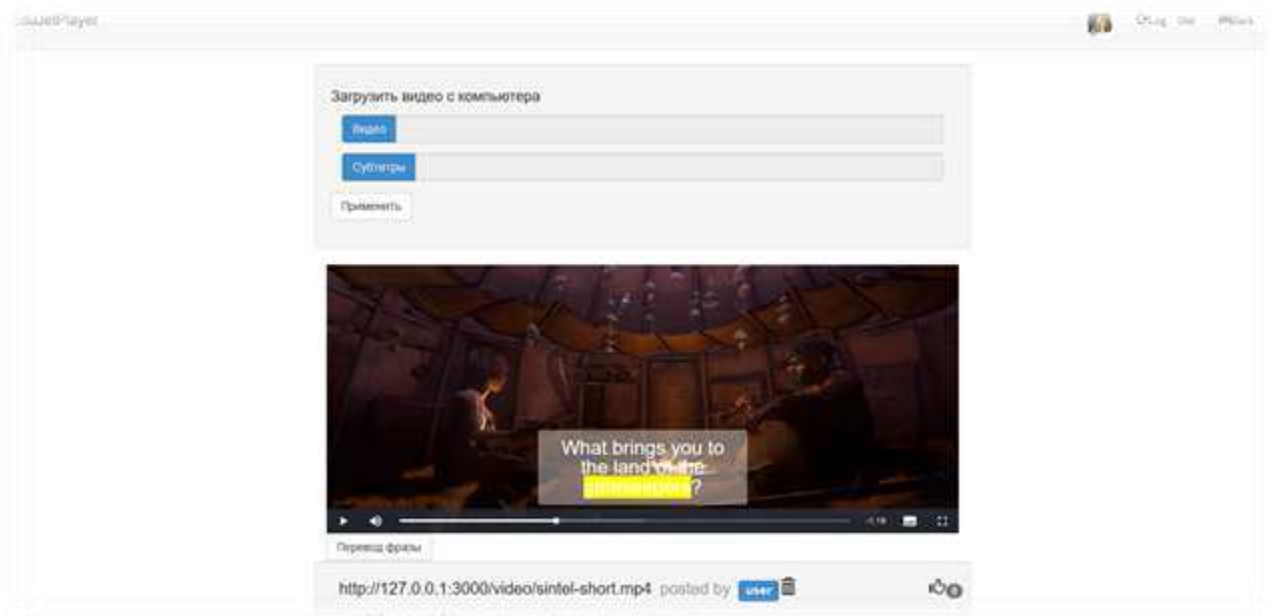


Рисунок 31 – Старый дизайн интерфейса

В данном коммите мы произвели рефакторинг шаблонов и разбили части страницы на разные файлы jade, что было необходимо для удобной работы с разными блоками пользовательского интерфейса нашего сервиса. На рисунке 32 предоставлен список измененных шаблонов и стилей.

+5	-0	M	assets/stylus/style.css
+6	-3	M	assets/template/_layout.jade
+114	-0	A	assets/template/home(old).jade
+35	-85	M	assets/template/home.jade
+33	-0	A	assets/template/includes/left_sidebar.jade
+57	-0	A	assets/template/includes/right_sidebar.jade
+1	-1	M	assets/template/login.jade
+1	-1	M	assets/template/register.jade

Рисунок 32 – Измененные шаблоны и стили

Рассмотрим для примера шаблон правого сайдбара, в котором должны отображаться сохраненные пользователем фразы. Исходный код правого сайдбара приведен на рисунке 33.

```
1 .col-md-2.pull-right
2 .sidebar.affix.pre-scrollable(style='height:100%; min-height:100%; width:15%; min-width:15%')
3   div(ng-repeat="post in posts | orderBy:'-upvotes'", style="font-size:20px")
4     div(class="well row")
5       span(style="margin:10px;" ng-show="post.title")
6         | {{post.title}}
7       span(ng-show="post.author" style="margin:10px; opacity: 0.5;")
8         | posted by
9       span(class="label label-pill label-primary" ng-show="post.author" )
10        a(href="#/profile/{{post.author}}" style="color: white;") {{post.author}}
11
12        div(style = "display:inline; position: absolute;right: 0;margin-right:10px;")
13          span(class="glyphicon glyphicon-thumbs-up", ng-click="incrementUpvotes(post)")
14          span(class="badge")
15            | {{post.upvotes}}
16        div(style = "display:inline; position: absolute;right: 10;margin-right:10px;")
17          span(class="glyphicon glyphicon-trash", ng-click="delete(post)")
18        img(class="img-responsive" ng-src="{{post.image}}" alt="{{post.image}}" style="margin:10px;")
19        div(style="margin-left:20px;" ng-bind-html="post.text")
20
21
22    ul(class="pagination" ng-controller="NavCtrl" style="font-weight: bold;")
23      li
24        a(ng-click="prev()") << Prev
25      li
26        a(ng-click="next()") Next >>
27    div(ng-hide="isLoggedIn()")
28      h3
29        | You need to
30        a(href="#/login", style="margin:10px;") Log In
31        | or
32        a(href="#/register", style="margin:10px;") Register
33        | before you can add a post.
```

Рисунок 33 – Исходный код правого сайдбара

Первая строка кода создает div класса col-md-2 и pull-right что определяет его ширину в две колонки bootstrap стиля и смещает его к правому краю. Далее на строках 3-19 происходит загрузка сохраненных пользователем фраз и описание их интерфейса. На строчке 19 происходит непосредственно загрузка самого текста с сервера. Так же была добавлена возможность для сохранения картинок, чтобы в будущем пользователь мог вместе с фразой сохранять и момент на котором была фраза, это поспособствует запоминанию материала, так как подключится ассоциативная и зрительная память.

В итоге после большого количества исправлений и экспериментов удалось создать новую версию интерфейса, которая показана на рисунке 34.



Рисунок 34 – Новый дизайн интерфейса

Раньше плеер изменял свои параметры отображения по размеру видео и при открытии видео формата 4:3 плеер занимал слишком большое пространство и для его полного охвата приходилось пролистывать страницу, поэтому мы решили сделать фиксированные пропорции видеоплеера. На рисунке 35 показан пример отображения плеера с видео 4:3 до исправления.

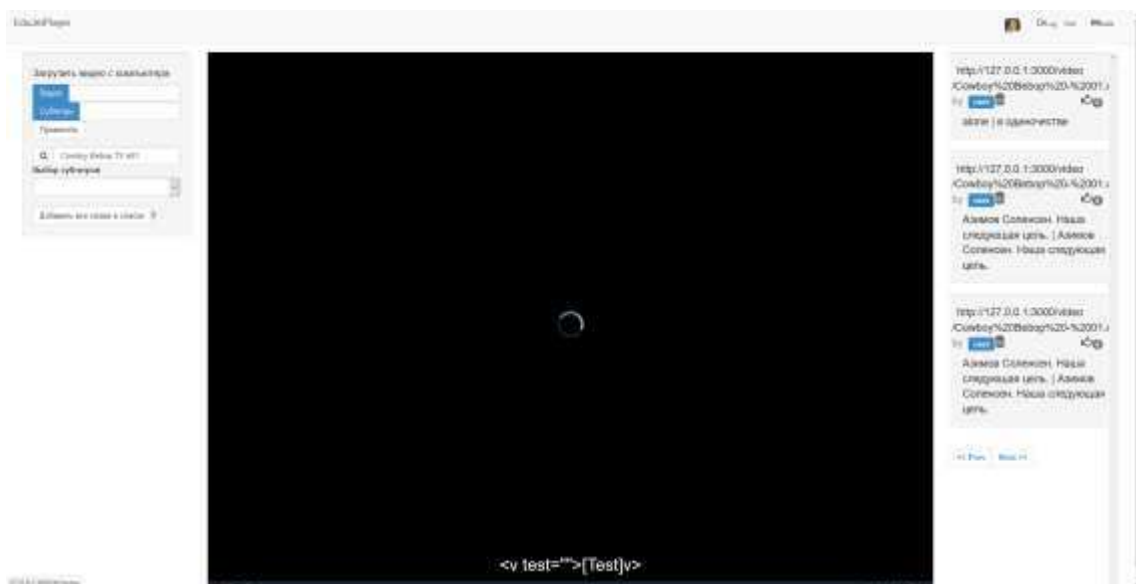


Рисунок 35 – Пример отображения плеера

Сохраненные пользователем и импортированные из коллекции Anki слова отображаются на странице профиля пользователя. Их вид представлен на рисунке 36. Красным цветом отмечен индекс знания 1 (самый низкий), желтым индекс 2, а красным индекс 3 (самый высокий).

Сохраненные слова:

астероидах : asteroids asteroid planetoid a : has : имеет blade : лопать past : мимо dark : темно This : Эта
 Test : Контрольная работа shhh : Вы shhh : тсс shhh : это shhh : путешествие shhh : мы shhh : Дракон
 shhh : запомнить shhh : shhh : родственный shhh : Это shhh : кровь you're : Вы brings : приносит i : я
 lucky : везучий much : много fool : дурачить so : так v : в been : было test : контрольная работа what : какие
 almost : почти can : Можно very : очень completely : полностью i'm : я we're : мы unprepared : неподготовленный
 for : для dragon : Дракон land : земельные участки hunter : охотник spirit : дух long : длинный of : из
 flowing : текущий thank : спасибо kindred : родственный the : lone : одинокий shed : сарай quest : поиск
 blood : кровь someone : кто то searching : поиск you : вы to : в it : Это innocent : невинный your : ваш
 alone : в одиночестве gatekeepers : привратники as : в виде still : все еще traveling : путешествие blood's : Блада
 dangerous : опасно done : сделанный dear : Уважаемые remember : запомнить this : это i've : я

Рисунок 36 – Сохранённые в базе слова

По итогу работ был сформирован функционал и основные модули программы, после чего построили макет интерфейса. Для командной разработки был использована система контроля версий Git. Были выбраны наиболее подходящие средства и среды разработки. В ходе разработки были реализованы основные функции нашего сервиса.

Для тестирования мы использовали универсальную библиотеку автоматизации веб браузера Selenium Web Driver. Selenium поддерживает некоторые из самых популярных браузеров, которые предпринимают шаги, чтобы сделать Selenium родной частью своего браузера. Это также основная технология в бесчисленных других средствах автоматизации браузеров, API-интерфейсах и инфраструктурах. Многие, возможно, большинство программных приложений сегодня написаны как веб-приложения для запуска в интернет-браузере. Эффективность тестирования этих приложений сильно различается среди компаний и организаций. В эпоху высоко интерактивных и гибких программных процессов, в которых многие организации используют методологию Agile, автоматизация тестирования часто становится требованием для программных проектов. Автоматизация тестирования часто является ответом. Автоматизация тестирования означает использование программного инструмента для запуска повторяющихся тестов для тестируемого приложения.

Существует множество преимуществ для автоматизации тестирования. Большинство из них связано с повторяемостью тестов и скоростью, с которой тесты могут быть выполнены. Существует ряд коммерческих и открытых инструментов, доступных для оказания помощи в разработке автоматизации тестирования. Selenium является наиболее широко используемым решением с открытым исходным кодом.

Тестирование статичного контента мы производили при помощи Selenium Web Driver на Python, так как у нас есть опыт работы именно с ним. Рассмотрим принцип работы тестирования на примере кода, представленном на рисунке 37.

```

1  from selenium import webdriver
2  from selenium.webdriver.common.keys import Keys
3
4  driver = webdriver.Firefox()
5  driver.get("http://127.0.0.1:3000")
6  assert "EduJetPlayer" in driver.title
7  elem = driver.find_element_by_name("SelectSubtitles")
8  elem.send_keys("Cowboy Bebop")
9  elem.send_keys(Keys.RETURN)
10 assert "No results found." not in driver.page_source
11 driver.close()

```

Рисунок 37 – Пример кода теста для проверки результатов выбора субтитров

В первой строке (рисунок 37) мы подключаем непосредственно сам webdriver, необходимый для запуска браузера в автоматическом режиме, во второй строке мы добавляем класс Keys, который требуется для взаимодействия с командами клавиатуры, такими как Enter, Shift, Alt итд. Далее на 4 строке мы создаем элемент класса Firefox WebDriver. Метод driver.get на 5 строке используется для открытия заданного url в браузере. Следующая строка утверждение того, что заголовок содержит слово «EduJetPlayer». Assert позволяет проверять предположения о значениях произвольных данных в произвольном месте программы). По своей сути assert это утверждение, встроенное в код программы. В случаях, когда заданное утверждение не верно, assert вызывает исключение. Это позволяет контролировать выполнение программы в заданных рамках. От обычных условий assert отличается тем, что программа с assert не примет иного хода событий. На строке 7 мы получаем элемент с названием «SelectSubtitles» и далее на следующей строке передаем ему значение субтитров, выбор которых хотим протестировать. Assert на 10 строке сработает, если субтитры не подключатся, и страница пришлет ответ «No results found.». Последней строкой мы закрываем окно браузера.

3.3 Методология работы с сервисом

Наш сервис, способен предоставлять пользователям функционал для удобного изучения иностранных языков при методе захвата и перевода

субтитров, с последующим их анализом и обработкой для формирования базы изученных пользователем слов.

Для примера рассмотрим интерфейс приложения Anki для Windows. В главном меню программы пользователю предоставлен список сохраненных коллекций карточек, краткая статистика, отображающая активность пользователя по дням и интерфейс навигации. На рисунке 38 представлен интерфейс главного меню программы.

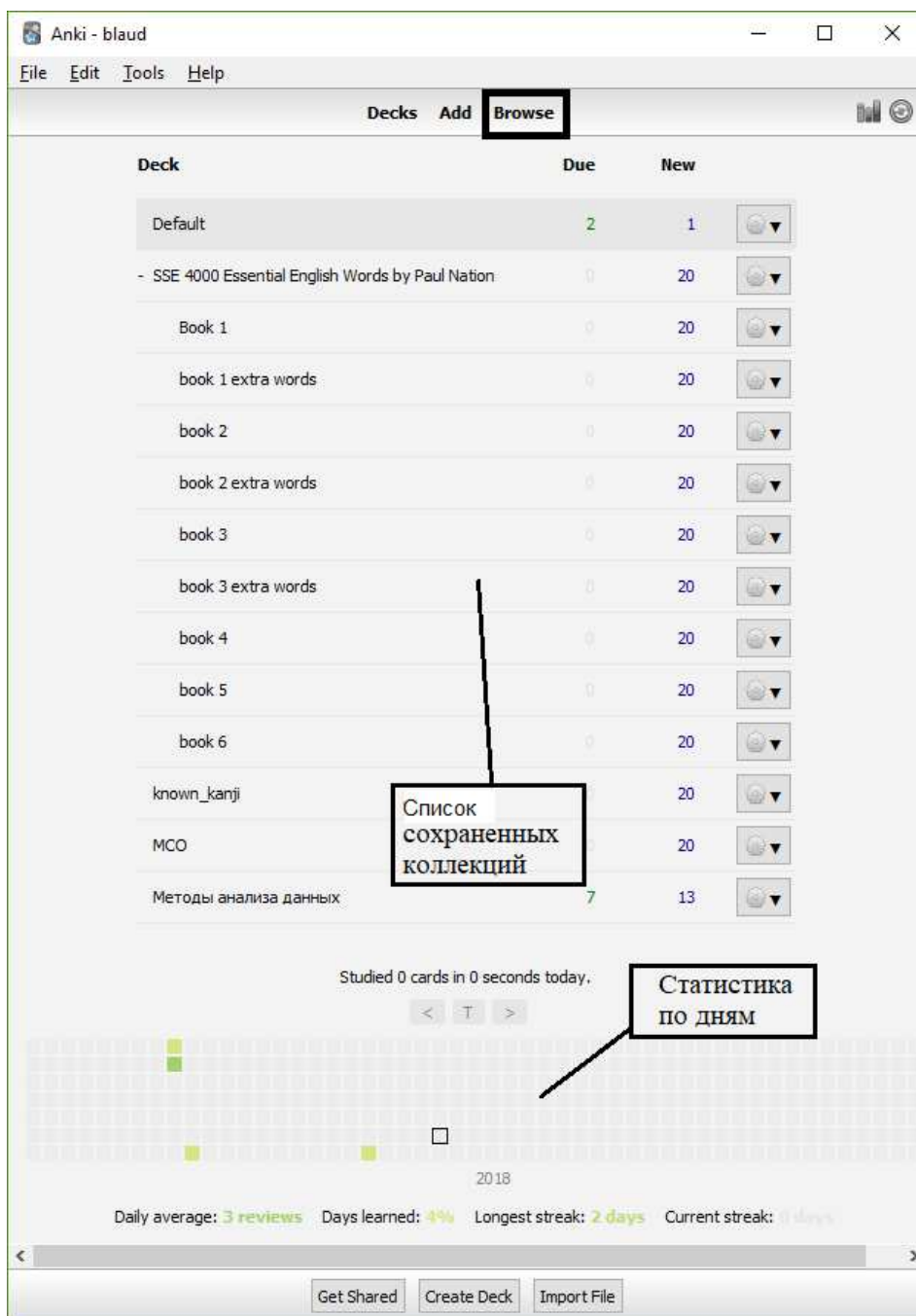


Рисунок 38 – Главное меню Anki Desktop

Чтобы перейти в меню просмотра сохраненных данных и их редактирования, нужно нажать кнопку Browse в середине верхней части интерфейса (рисунке 38). На рисунке 39 представлен интерфейс обзора сохраненных карточек.

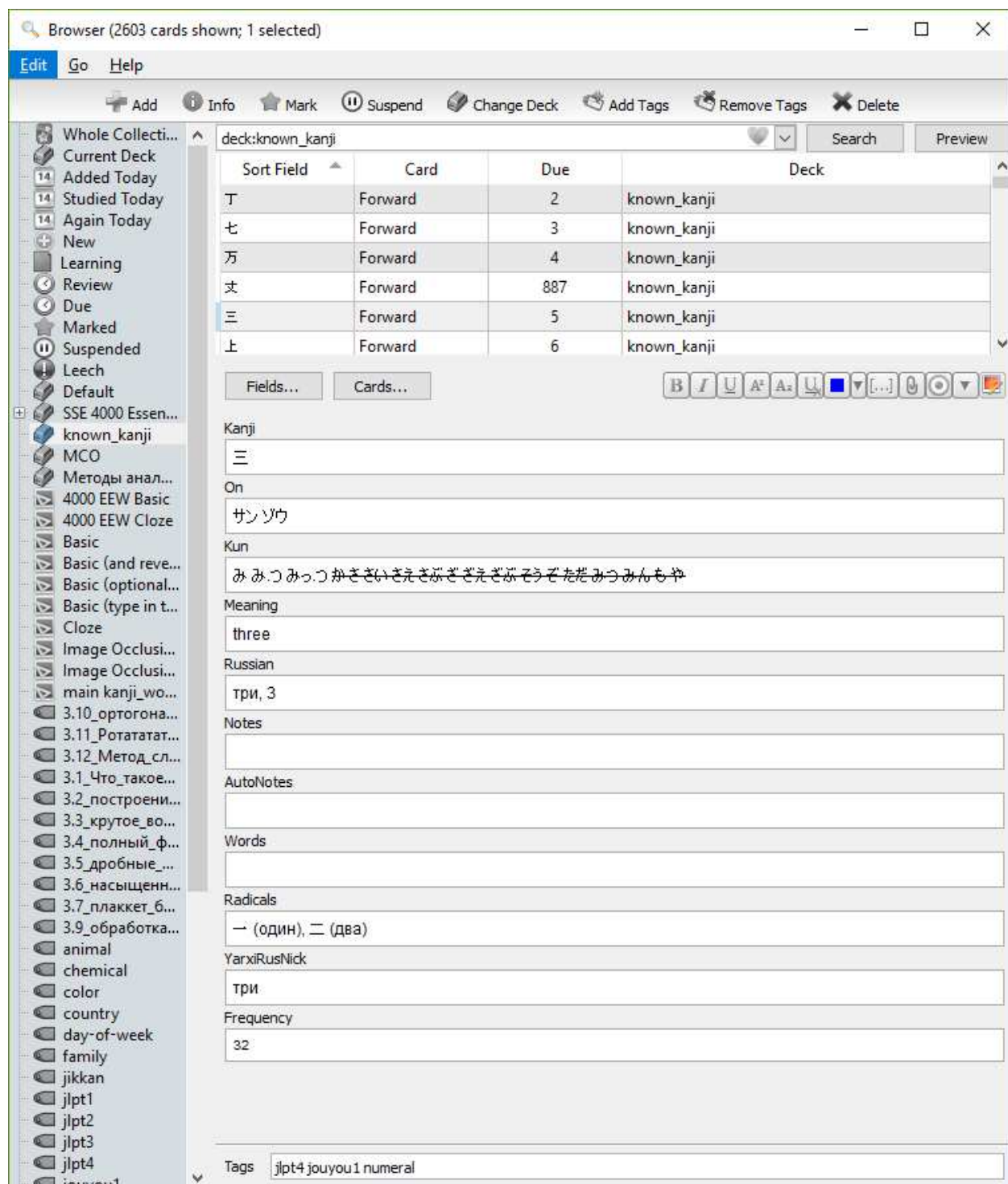


Рисунок 39 – Интерфейс обзора сохраненных карточек

На рисунке 39 справа находится интерфейс предоставляющий выбор коллекции для просмотра содержащихся в ней карточек. Так же можно выбрать карточки из нескольких коллекций по тегу. После панели инструментов сверху идет таблица со списком карт содержащихся в выбранной коллекции. При нажатии на одну из строчек таблицы, ниже отображается содержание карточки и поля для редактирования. Поля карточек определяются при их создании. Для разметки информации внутри карточки используется встроенный язык разметки. Для нашего сервиса мы первоначально использовали базовый тип карточек, однако в будущем планируется добавить в карточки звуковые и фотоматериалы, поэтому мы сделали собственный тип, который позволяет создавать более сложные карточки с расширенным функционалом. Интерфейс разметки карточек представлен на рисунке 40.

Для написания разметки карточки используются поля `Front template` и `Back Template`, расположенные слева интерфейса создания шаблона карточки (рисунок 40). Так же можно задавать стили CSS для карточки в поле `Styling`. Справа в реальном времени отображается внешний вид карточки, которая будет сгенерирована из шаблона. Так же в программе есть интерфейс для ручного создания карточек, в котором можно по заданному шаблону добавлять текстовые и медиа данные в карточку, но в рамках нашего проекта мы не будем рассматривать этот интерфейс, так как генерация карточек у нас будет происходить на сервере в автоматическом режиме.

Далее мы рассмотрим непосредственно сам процесс изучения слов в программе Anki. После выбора коллекции на главном меню пользователю представляется интерфейс с краткой статистикой по изученным карточкам данной коллекции, в котором есть кнопка `Study now` (рисунок 41), после нажатия которой пользователю показывается первая в очереди на изучение карточка из коллекции (рисунок 42). Сперва пользователю показывается только иностранное слово или фраза, которую он должен вспомнить и проговорить про себя, после чего нажать кнопку `Show Answer`, тогда ему отобразится обратная сторона карточки с переводом (рисунок 43).

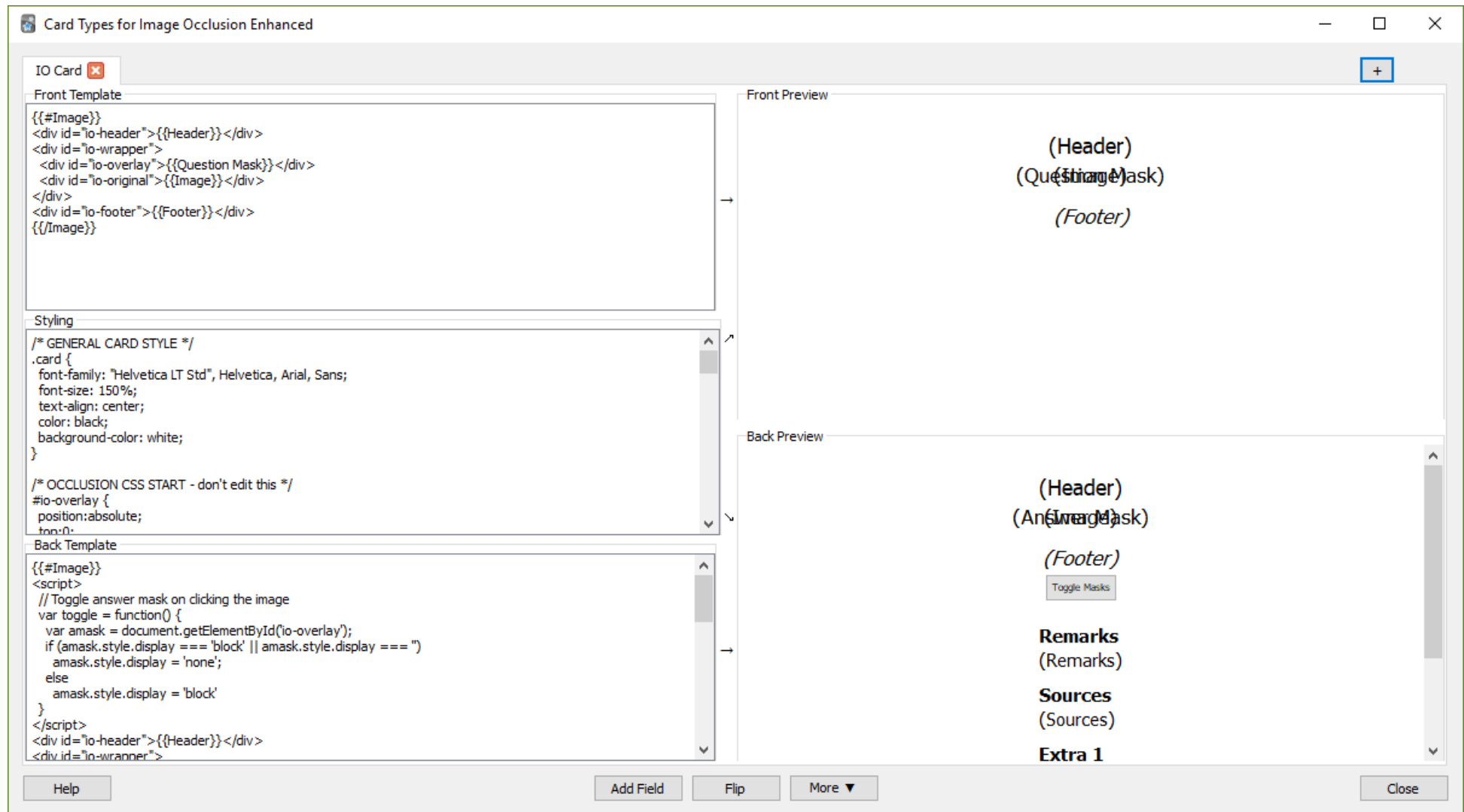


Рисунок 40 – Интерфейс разметки карточек

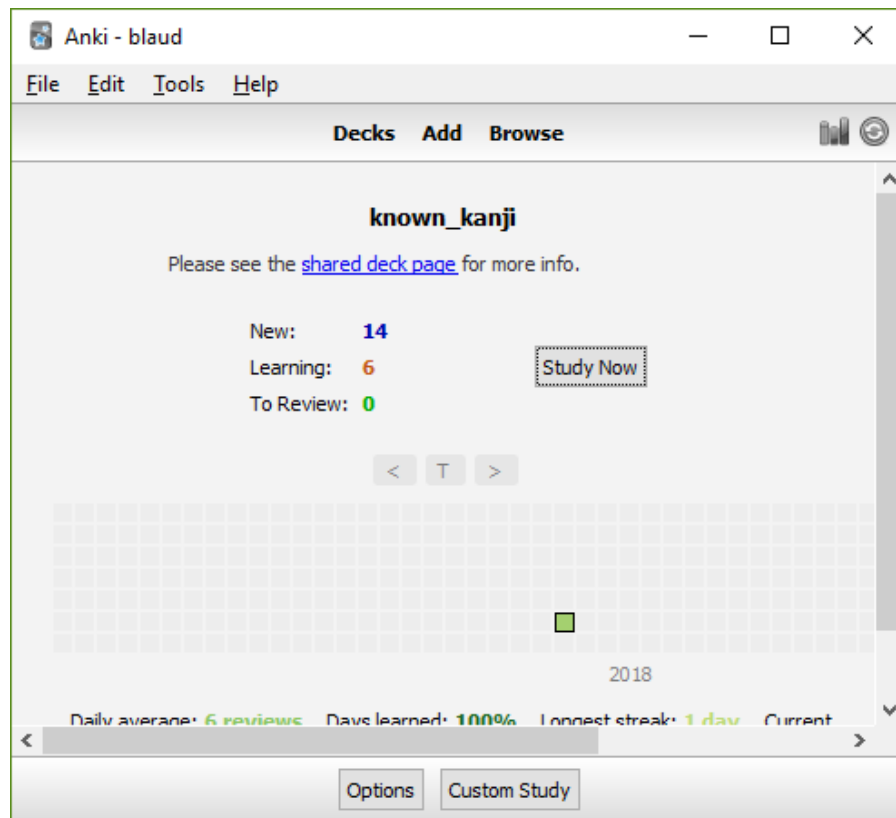


Рисунок 41 – Интерфейс обзора статистики выбранной коллекции

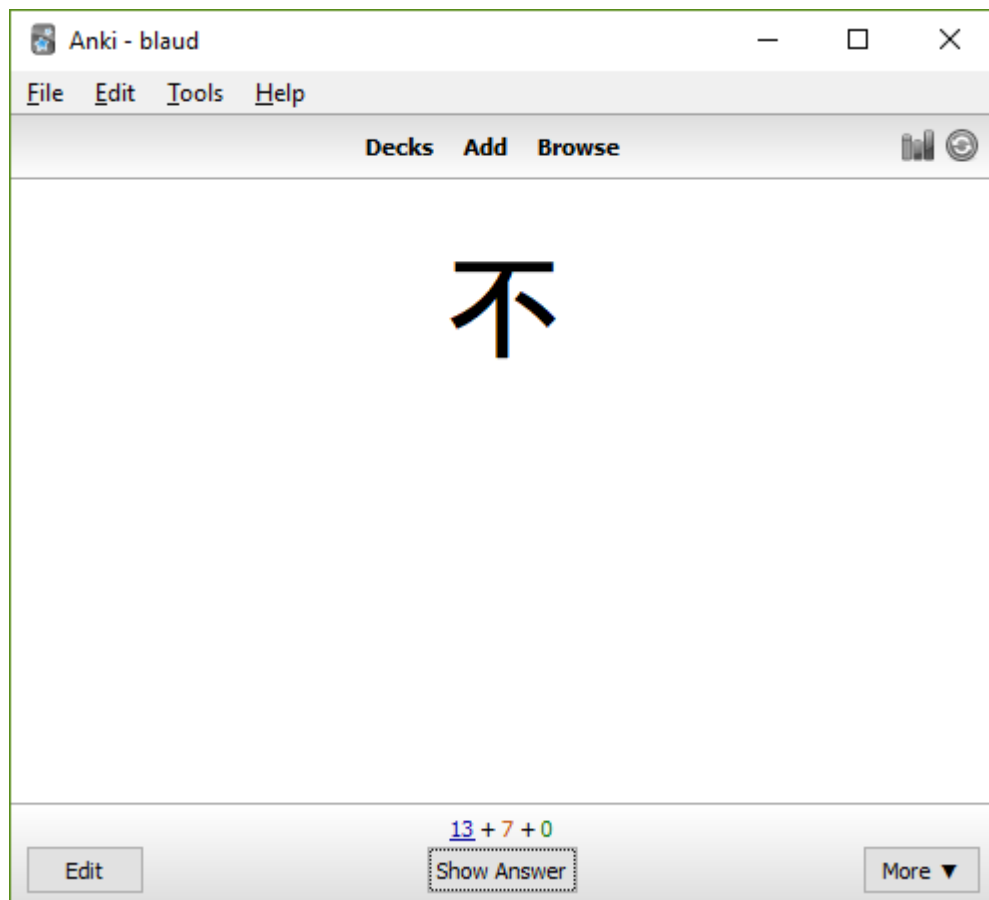


Рисунок 42 – Интерфейс изучения карточки



Рисунок 43 – Обратная сторона карточки

После того, как пользователь посмотрел ответ, он должен оценить насколько правильно он ответил и в зависимости от результата выбрать одну из трех кнопок в центре нижней части интерфейса, которые отражают степень изученности карточки и в зависимости от выбора пользователя будет выбран оптимальный интервал повторения карточки на основе техники SuperMemo. После нажатия на любую из трех кнопок, будет показана следующая в очереди на изучение карточка.

SuperMemo (Супер память) - это метод обучения, разработанный SuperMemo World и SuperMemo R & D с Петром Войняком в Польше с 1985 года по настоящее время. Он основан на исследованиях долговременной памяти и представляет собой практическое применение метода интервального повторного обучения, который был предложен для эффективного обучения ряда психологов еще в 1930-х годах. По словам разработчиков SuperMemo и некоторых других

сторонников разнесенного обучения повторения, этот процесс может оптимизировать долгосрочное приобретение знаний.

Далее мы приведем общий алгоритм работы техники запоминания. SuperMemo начинает попытки вычислить оптимальные интервалы между повторениями, сохраняя запись оценки пользователя при обучении. Эта запись используется для оценки текущей характеристики памяти пользователя и сложности лежащего в ее основе предмета. Сложность предмета выражает сложность запоминания и отражает усилия, необходимые для создания однозначных и стабильных следов памяти. SuperMemo берет запрошенный коэффициент возврата в качестве критерия оптимизации (например, 95%) и вычисляет интервалы, удовлетворяющие этому критерию. Функция оптимальных интервалов представлена в матричной форме (матрица OF) и может быть изменена на основе результатов учебного процесса. Хотя выполнение критерия оптимизации довольно легко, сложность алгоритма обусловлена необходимостью получения максимальной скорости сходимости в свете известных моделей памяти.

Теперь рассмотрим то, как пользоваться нашим сервисом. «EduJetPlayer» позволит с пользой проводить время за просмотром любимых сериалов или фильмов, попутно изучая желаемый язык. Эффективность методов сервиса состоит в том, что он предоставляет удобный интерфейс для сохранения и перевода субтитров, и анализом уже изученных слов, для формирования списка слов необходимых для понимания конкретного видео пользователем

Для того что бы пользоваться сервисом «EduJetPlayer» необходимо что бы пользователь имел навыки работы хотя бы с одним интернет браузером из списка:

- «Internet Explorer»;
- «Edge»;
- «Mozilla Firefox»;
- «Opera»;
- «Yandex Browser»;

- «Google Chrome».

Функции сервиса «EduJetPlayer» которыми может воспользоваться пользователь:

- воспроизведение видео пользователя;
- интерфейс управления воспроизведением;
- сохранение субтитров для изучения;
- перевод субтитров;
- генерация коллекции Anki из сохраненных субтитров;
- интерфейс загрузки сгенерированной коллекции Anki;
- отображение рекомендуемых к изучению фраз.

Доступ пользователей к сервису «EduJetPlayer» осуществляется с применением технологии «тонкого клиента» при помощи интернет-браузера.

Персональные компьютеры пользователей должны удовлетворять следующим минимальным требованиям:

- память – не менее 1 Гб, рекомендуется 4 Гб;
- операционная система Windows 7 и старше, Ubuntu 14.04 и старше, OS X 10.10 и старше;
- браузер MS IE 9.0 и выше, либо Google chrome, Mozilla Firefox, Opera, Edge;

- канал доступа в Интернет – 1 Мбит/с и выше.

Для запуска сервиса «EduJetPlayer» пользователю необходимо перейти по электронному адресу «edujetplayer.blaud.ru». Что бы убедиться в том, что сервис работоспособен, пользователю необходимо удостовериться в том, что на экране браузера отобразилось главное окно клиентского приложения без выдачи пользователю сообщений о сбое в работе.

Операции доступные пользователю:

- воспроизведение видео пользователя;
- интерфейс управления воспроизведением;
- сохранение субтитров для изучения;
- перевод субтитров;

- генерация коллекции Anki из сохраненных субтитров;
- интерфейс загрузки сгенерированной коллекции Anki.

Для загрузки видео необходимо в главном окне выбрать источник видео и субтитров, и нажать кнопку применить (рисунок 44):



Рисунок 44 – Загрузка видео

Для перевода субтитров наведите указатель мыши на окно с субтитрами. Видео остановится, и вы сможете либо.

Нажать по интересующему вас слову (рисунок 45) либо по всей фразе (рисунок 46), выделить интересующий фрагмент текста (рисунок 48). Перевод отображается в окне в верхней части плеера. Что бы сохранить его для последующего изучения нажмите на кнопку сохранения (рисунок 48).



Рисунок 45 – Выбор одного слова

EduJetPlayer Log Out Back

Загрузить видео с компьютера

Видео:

Субтитры:

Выбор субтитров



<< Prev Next >>

Рисунок 46 – Перевод всей фразы

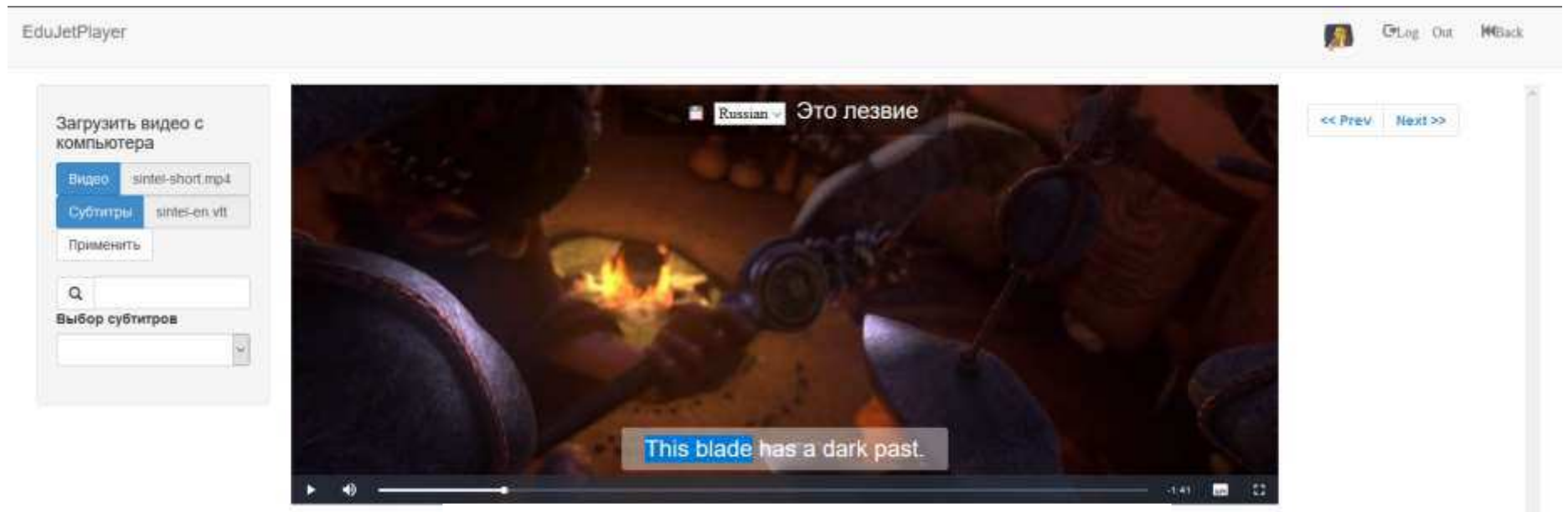


Рисунок 47 – Выделение конкретного фрагмента

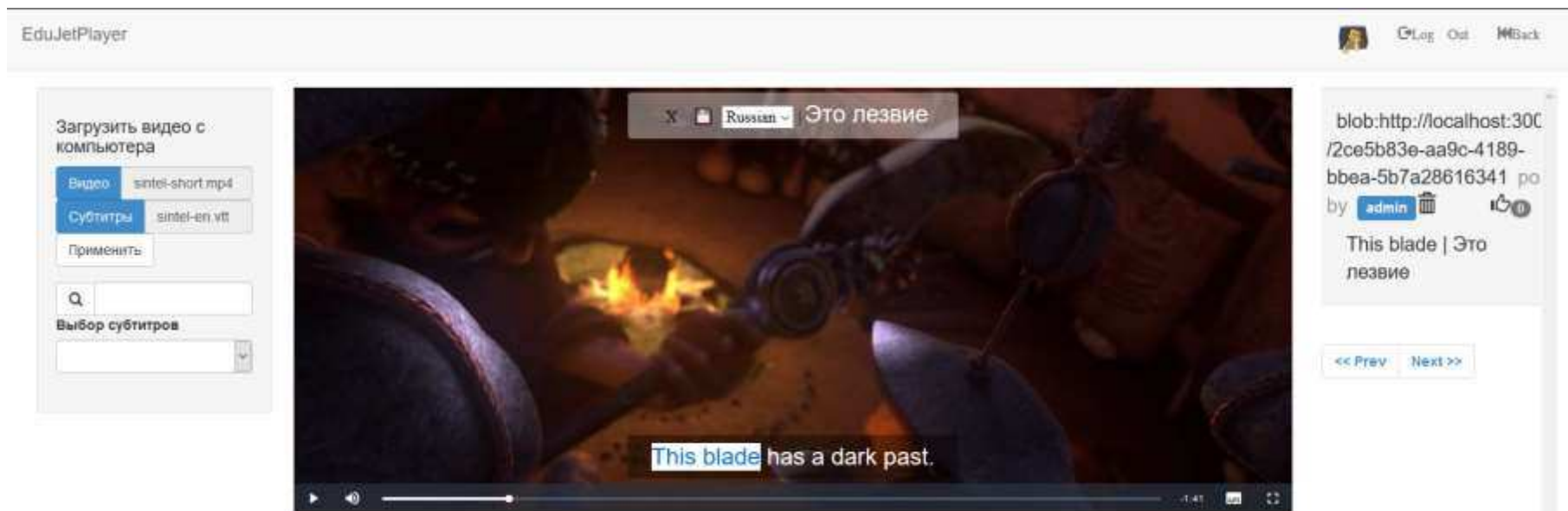


Рисунок 48 – Сохранение перевода

Для загрузки файла anki пользователю нужно перейти в раздел личного кабинета нажав по аватару профиля в верхней части экрана (рисунок 49)

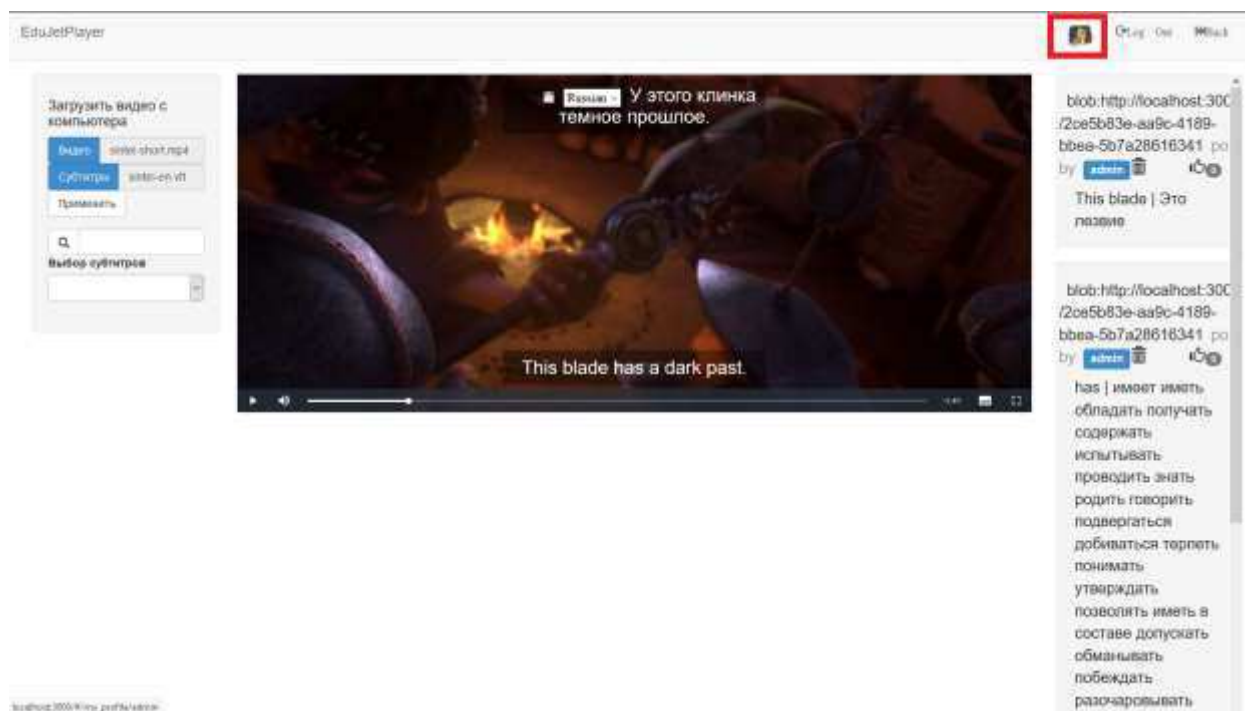


Рисунок 49 – Аватар профиля

В профиле необходимо нажать кнопку download anki (рисунок 50)



Рисунок 50 – Кнопка download Anki

Браузер предложит выбрать место для сохранения импортированной коллекции anki (рисунок 51).

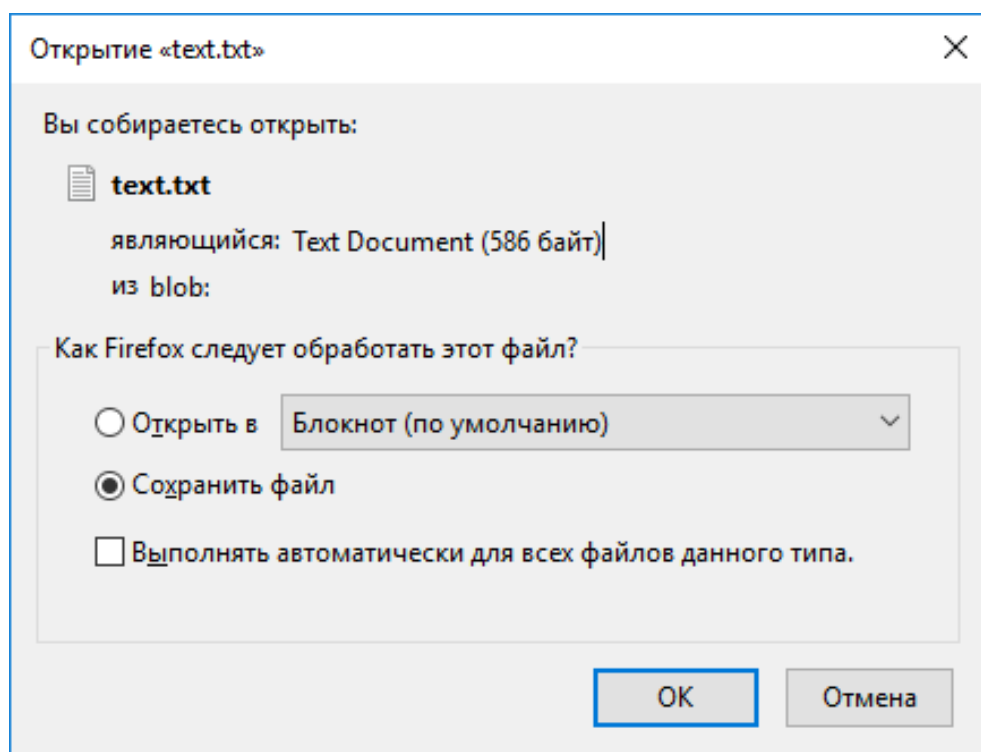


Рисунок 51 – Коллекция для импорта

При сбое в работе аппаратуры восстановление нормальной работы сервиса «EduJetPlayer» должно производиться после:

- перезагрузки страницы с сервисом;
- очистки кэша браузера.

При ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции сервиса «EduJetPlayer» возлагается на ОС. При ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

При неверных действиях пользователей, неверных форматах или недопустимых значениях входных данных, сервис «EduJetPlayer» выдает пользователю соответствующие сообщения, после чего возвращается в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

Ниже рассмотрен пример контрольный пример работы с сервисом «EduJetPlayer», начиная с ее запуска и заканчивая регистрацией:

- запустите браузер;

- введите в нем адрес сервиса «EduJetPlayer» (рисунок 52);
- перейдите по нему. Сервис «EduJetPlayer» предложит вам создать аккаунт для возможности сохранения ранее просмотренных переводов субтитров (рисунок 53);
- зарегистрируйтесь, введя предпочитаемый логин и пароль, где можно указать ссылку на аватар для вашего профиля (рисунок 54);
- вы должны увидеть, что теперь сервис «EduJetPlayer» больше не предлагает вам войти в свою учетную запись или зарегистрироваться (рисунок 55).



Рисунок 52 – ввод адреса

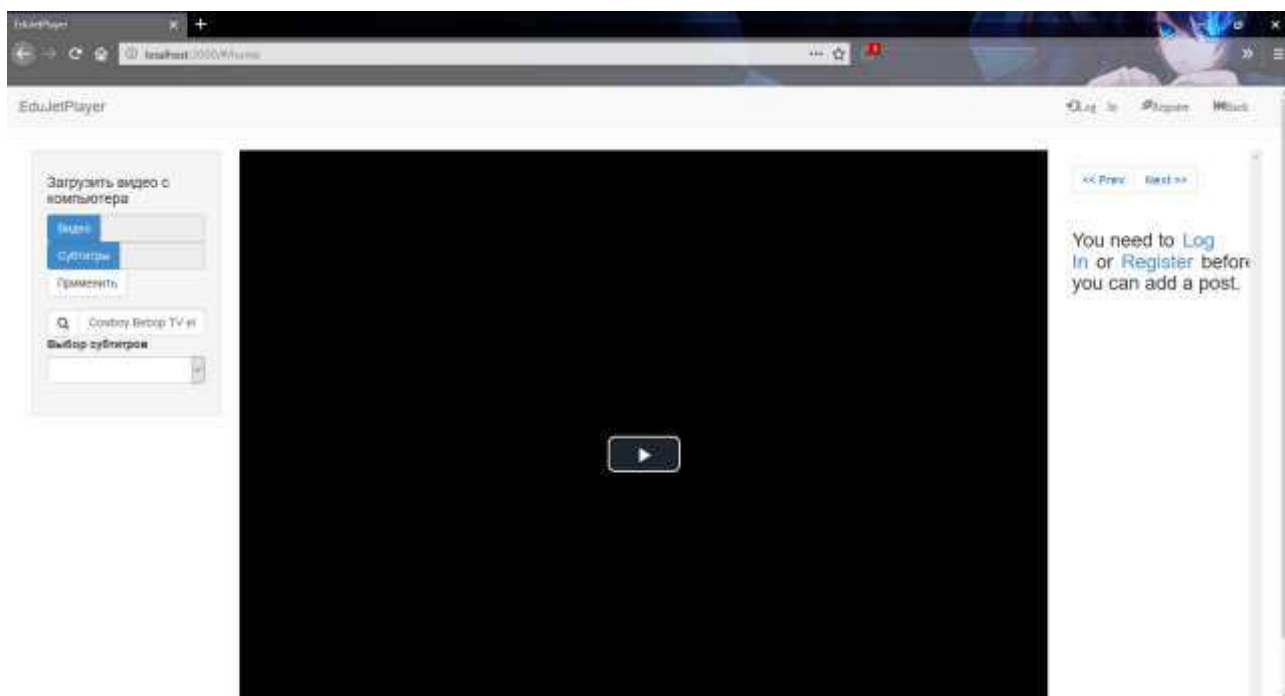
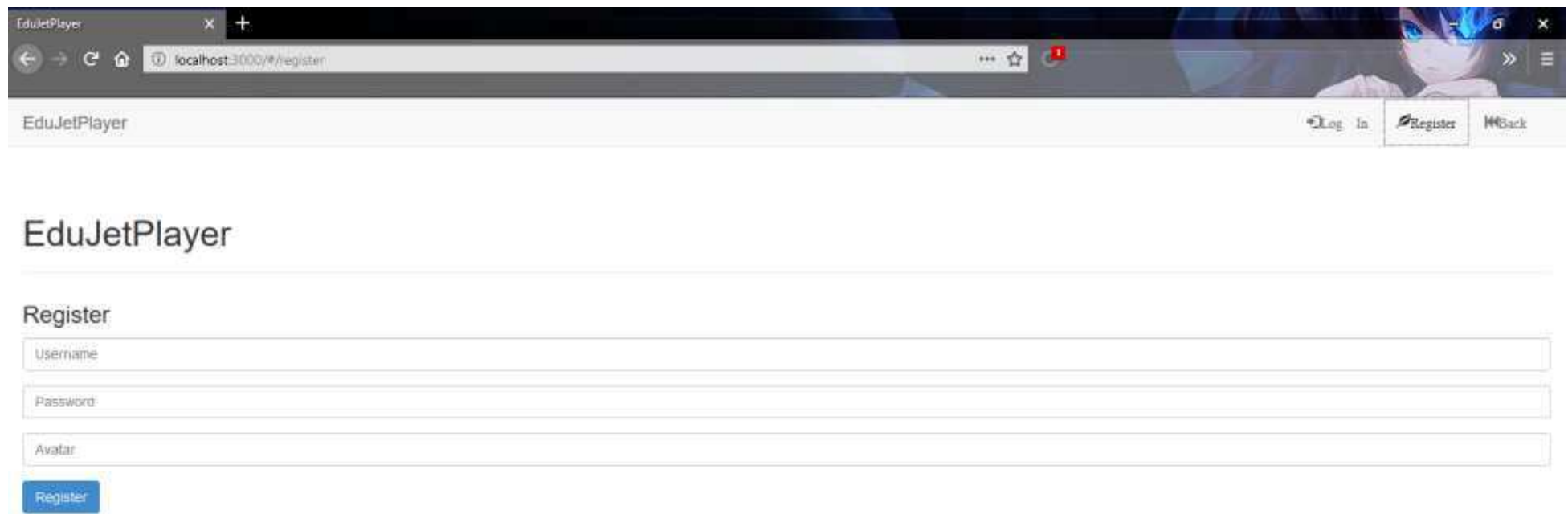


Рисунок 53 – Главный экран сервиса



EduJetPlayer

localhost:3000/#/register

EduJetPlayer

Log In Register Back

EduJetPlayer

Register

Username

Password

Avatar

Register

Рисунок 54 – Окно с регистрацией

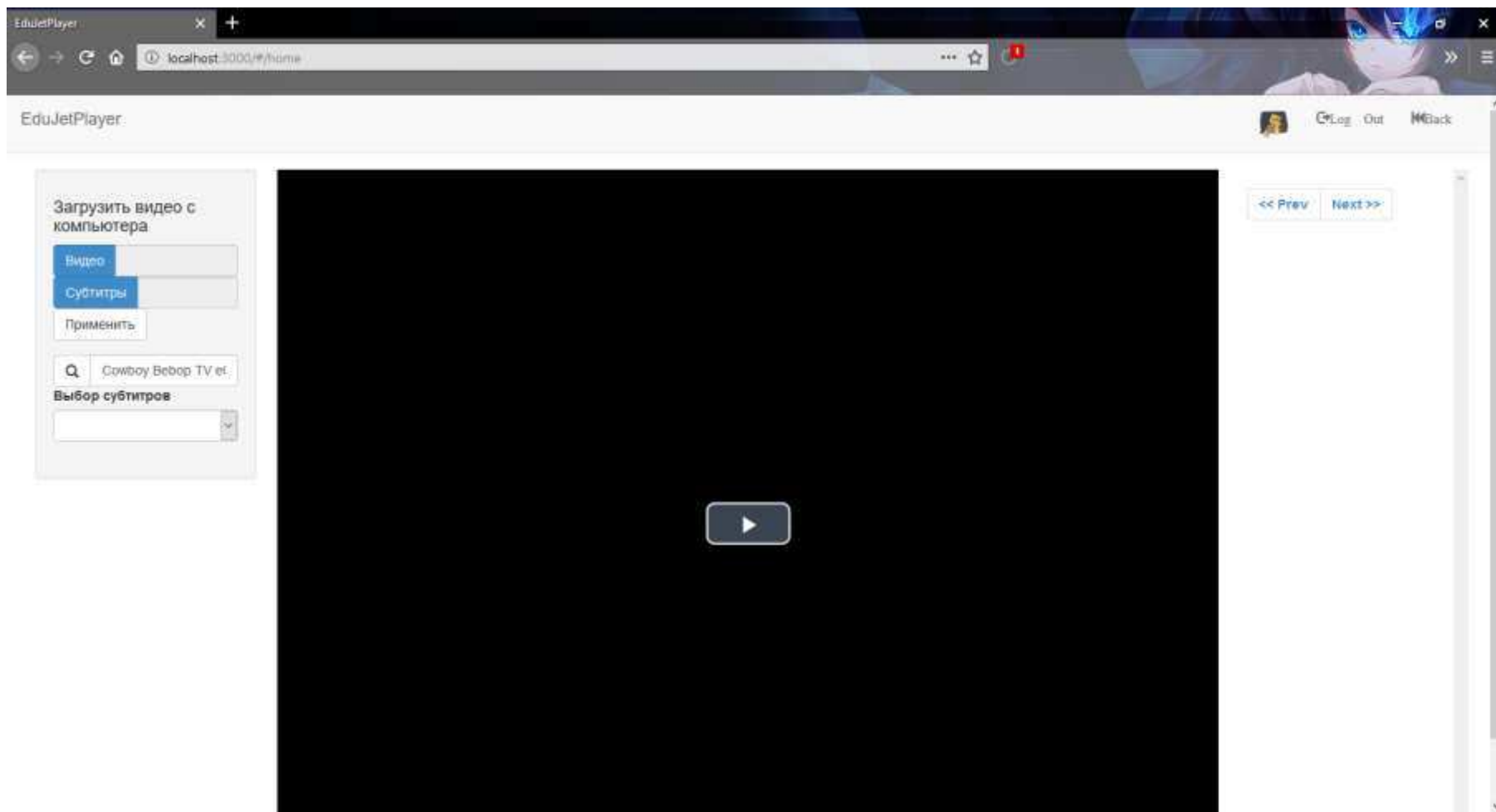


Рисунок 55 – Вход после регистрации

После процесса разработки мы протестировали эффективность нашего сервиса для помощи в изучении иностранных языков. В качестве испытуемого был выбран один из участников нашей команды разработчиков, испытание проводилось в течении семи дней.

Языком для изучения был выбран английский, так как испытуемый имел достаточный опыт владения языком для понимания общего смысла субтитров. Главной целью испытания было расширение словарного запаса, и наш сервис идеально подходил для этого.

В качестве учебного материала для испытаний мы использовали трилогию фильма «Матрица» и трилогию фильма «Властелин колец». В рамках испытания испытуемый смотрел одну часть трилогии в день, после чего изучал в течении двадцати минут сохраненные материалы.

После просмотра первой части трилогии фильма «Матрица» испытуемый добавил в базу 1313 новых слов, из них 37 неизвестных для него. На рисунке 56 изображена часть списка добавленных в базу слов. Далее испытуемый экспортировал неизученные слова в Anki и прошел процесс изучения в течении 20 минут. В результате он выучил полностью 8 слов, 12 слов он выучил наполовину, и оставшиеся 17 слов программа обучения рекомендовала оставить на следующий раз.

Затем испытуемый повторял данную процедуру в течении 7 дней, добавляя каждый день новые слова для изучения. В ходе испытания сервиса им было обнаружено несколько незначительных багов, мешающих удобству работы, и мы исправляли их по мере проведения эксперимента.

Результатом эксперимента стало повышение словарного запаса испытуемого на 109 ранее им неизвестных слов и формирование его базы знаний, в которой на момент окончания эксперимента содержалось 3842 слова



v : в landlady : помещица boundaries : границы brought : привел equipment : оборудование consumed : потребляются loco : сумасшедший
 hypocrites : лицемеры obviousness : очевидность impulses : импульсов bugged : прослушивают traveling : путешествие it : Это been : было
 the : you're : Вы i've : я almost : почти we're : мы still : все еще i : я completely : полностью i'm : я innocent : невинный shed : сарай
 fool : дурачить thank : спасибо blade : лопасть blood : кровь unprepared : неподготовленный very : очень for : для past : мимо
 much : много a : as : в виде searching : поиск lucky : везучий you : вы dark : темно lone : одинокий dragon : Дракон
 kindred : родственник land : земельные участки dear : Уважаемые long : длинный can : Можно spirit : дух test : контрольная работа
 has : имеет dangerous : опасно of : из quest : поиск to : в remember : запомнить flowing : текущий your : ваш blood's : Блада what : какие
 shhh : тсс someone : кто то brings : приносит so : так alone : в одиночестве hunter : охотник done : сделанный gatekeepers : привратники
 this : это think : думать be : быть adams : Адамса late : поздно problem : проблема minutes : минут here : Вот these : эти oh : ой
 simulations : моделирование main : главный fate : судьба doesn't : не struck : пораженный flat : квартира dreaming : сновидение
 mgnav : मग्न watched : смотрели unplugged : отключен от сети understand : Понимаю supposed : предполагаемый me : меня under : под
 dick : Дик cut : порез ridiculous : смешной make : делать yes : да whole : все deep : глубоко once : один раз great : Великий
 relieve : освобождать else : еще you'd : вы бы born : Родился it's : его lake : озеро over : над dream : мечта used : используемый
 replication : копирование please : пожалуйста desk : стол письменный companies : компании ready : готов unable : не в состоянии
 alice : Алиса team : команда fed : кормили slowly : медленно ass : жопа perfectly : в совершенстве brief : краткое sees : видит had : имел
 mean : имею в виду looking : Ищу an : matrix : матрица goes : идет carrier : перевозчик knew : знал crime : преступление exactly : в точку
 office : офис lock : замок more : Больше soon : скоро church : церковь is : является signals : сигналы nanban : Nanban everything : все
 respectable : респектабельный line : линия state : государство believe : верить mistaken : ошибочный core : ядро shift : сдвиг like : как
 don't : не morpheus : Морфей clothes : одежда everywhere : везде taste : вкус then : тогда before : до power : мощность gave : дал
 expecting : ожидая operations : операции she : она phone : Телефон doubt : сомнение lot : много cerebrum : мозг touch : потрогать
 lied : солгал apply : подать заявление taking : принятие roof : крыша nebuchadnezzar : Навуходоносор spawned : породил show : показать

3.4 Технические требования к серверу и установка

Технические требования к серверу:

- а) 4 гигабайта оперативной памяти (рекомендуется 8 гигабайт)
- б) 500 гигабайт места на жестком диске для записи.
- в) пропускная способность канала сервера не менее;
 - 1) входящий канал – 8 Мбит/с;
 - 2) исходящий канал – 4 Мбит/с;
- г) требования к портам:
 - 1) TCP порты 80, 443 и 1935 должны быть доступны;
 - 2) TCP порт 7443 должен быть доступен, при условии настройки SSL, иначе должен быть доступен порт 5066;
 - 3) UDP порты 16384 - 32768 должны быть доступны.

Рассмотрим процедуру установки серверной части. На сервере должен быть установлен MongoDB. Для этого необходимо убедиться, что используемая платформа есть в списке (таблица 3) поддерживаемых платформ.

Далее будет рассматривается установка MongoDB для Ubuntu 64-разрядных версий LTS (долгосрочная поддержка), так как именно на такой системе мы гарантируем успешный результат, по причине того что нами была использована именно такая конфигурация. Для установки следуйте инструкции:

- Импортируйте открытый ключ, используемый системой управления пакетами, выполнив следующую команду для импорта основного ключа GPG MongoDB: «`sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80`»;

- Создайте файл списка для MongoDB. Создайте файл списка «`/etc/apt/sources.list.d/mongodb-org-3.6.list`», используя команду: «`echo "deb [arch=amd64,arm64] https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.6.list`»;

- Перезапустите локальную базу данных пакетов. Выполните следующую команду для перезагрузки локальной базы данных пакетов: «`sudo apt-get update`»;

- Установите пакеты MongoDB. Установите последнюю стабильную версию MongoDB. Выполните следующую команду: «sudo apt-get install -y mongodb-org»;

Установка MongoDB завершена. Теперь требуется запустить MongoDB.

Таблица 3 – Список поддерживаемых платформ

Платформа	3.6 Community & Enterprise	3.4 Community & Enterprise	3.2 Community & Enterprise
Amazon Linux 2013.03 and later	+	+	+
Debian 7	+	+	+
Debian 8	+	+	+
Debian 9	3.6.5 и поздние		
RHEL/CentOS 6.2 and later	+	+	+
RHEL/CentOS 7.0 and later	+	+	+
SLES 11	до 3.6.4.	до 3.4.15	до 3.2.20
SLES 12	+	+	-
Solaris 11 64-bit	-	только Community	только Community
Ubuntu 12.04	до 3.6.4	до 3.4.15	до 3.2.20
Ubuntu 14.04	+	+	+
Ubuntu 16.04	+	+	+
Windows Vista	+	+	+
Windows 7/Server 2008 R2 and later	+	+	+
macOS 10.7 and later	+	+	+

Рассмотрим запуск MongoDB. Большинство Unix-подобных операционных систем ограничивают системные ресурсы, которые может использовать сеанс. Эти ограничения могут негативно повлиять на работу MongoDB. Экземпляр MongoDB хранит свои файлы данных в «/var/lib/mongodb»

и его файлы журнала в «/var/log/mongodb» по умолчанию и запускается с использованием учетной записи пользователя `mongodb`. Вы можете указать альтернативные каталоги файлов журналов и данных в файле «/etc/mongod.conf». Для получения дополнительной информации см. файл «systemLog.path» и «storage.dbPath». Если вы измените пользователя, который запускает процесс `MongoDB`, вы должны изменить права доступа в каталогах «/var/lib/mongodb» и «/var/log/mongodb», чтобы дать этому пользователю доступ к этим каталогам.

- Для запуска «`mongod`» выполните следующую команду «`sudo service mongod start`»;

- Убедитесь, что `MongoDB` успешно запущен. Убедитесь, что процесс «`mongod`» успешно запущен, проверяя содержимое файла журнала «/var/log/mongodb/mongod.log» на следующую строку «`[initandlisten] waiting for connections on port 27017`».

«порт» - это порт, который слушает «`mongod`». Если вы изменили настройку «`net.port`» в файле конфигурации «/etc/mongod.conf», порт может отличаться. Если вы изменили параметр файла конфигурации «`systemLog.path`», найдите файл журнала в том месте, которое вы указали для этого параметра. Вы можете увидеть некритические предупреждения в выводе «`mongod`».

Затем следует установить и запустить систему управления пакетами `NPM`. Для этого следует выполнить следующую команду «`sudo apt-get install npm`».

Теперь установите файлы сервера в каталог, и запустите в нем команду «`npm install`»

Для запуска сервера требуется выполнить команду в директории сервера «`npm start`»

Для того что бы убедиться в том, что сервер работает вы можете найти в командной строке следующую запись «`Node server started on :3000 ...`»

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы проанализировали аналоги разрабатываемого сервиса для повышения навыков владения иностранными языками на основе техник «СуперПамяти» и захвата субтитров и выявили общие существенные недостатки:

- отсутствие возможности для загрузки собственных материалов для просмотра;

- дороговизна подписки;

- поддержка только двух языков;

- отсутствие функционала для формирования списка слов, необходимых для понимания видео.

Для решения выявленных недостатков было решено создать сервис для помощи в изучении иностранных языков. Для удобства пользователя мы решили сделать наш сервис в формате веб приложения, так как оно не требует установки.

В ходе разработки мы реализовали сервис для повышения навыков владения иностранными языками на основе техник «СуперПамяти» и захвата субтитров и опробовали метод изучения иностранных слов по субтитрам на себе. Однако остался не обязательный к разработке функционал, такой как расширение базы поиска по субтитрам и его оптимизация, плагин для браузера.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Github - Google Translator API for FREE [Электронный ресурс] // GitHub – платформа разработки. – Режим доступа: <https://github.com/731MY/google-translator> (дата обращения 08.06.2018).

2 PHP: что такое PHP? - Manual [Электронный ресурс] // Язык программирования общего назначения. – Режим доступа: <http://php.net/manual/ru/intro-what-is.php> (дата обращения 08.06.2018).

3 Веб-технологии для разработчиков - JavaScript [Электронный ресурс] // Ресурсы для разработчиков, от разработчиков. – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата обращения 08.06.2018).

4 Язык программирования Ruby [Электронный ресурс] // Динамический язык программирования. – Режим доступа: <https://www.ruby-lang.org/ru/> (дата обращения 08.06.2018).

5 Python programming language [Электронный ресурс] // Язык программирования. – Режим доступа: <https://www.python.org/> (дата обращения 08.06.2018).

6 JsTestDriver project on Google Code [Электронный ресурс] // Выполнение тестов JavaScript. – Режим доступа: <https://code.google.com/archive/p/js-test-driver/> (дата обращения 08.06.2018).

7 Selenium home page [Электронный ресурс] // Автоматизация веб-приложений для целей тестирования. – Режим доступа: <https://www.seleniumhq.org/> (дата обращения 08.06.2018).

8 Home – Testswarm Wiki – Github [Электронный ресурс] // Распределенное тестирование непрерывной интеграции для JavaScript. – Режим доступа: <https://github.com/jquery/testswarm/wiki> (дата обращения 08.06.2018).

9 Документация Visual Studio Code [Электронный ресурс] // Среда разработки Visual Studio Code. – Режим доступа: <https://code.visualstudio.com/docs> (дата обращения 08.06.2018).

10 Справка по WebStorm 2018,1 [Электронный ресурс] // Среда разработки WebStorm. – Режим доступа: <https://www.jetbrains.com/help/webstorm/meet-webstorm.html> (дата обращения 08.06.2018).

11 Домашняя страница Notepad++ [Электронный ресурс] // Бесплатный редактор исходного кода. – Режим доступа: <https://notepad-plus-plus.org/> (дата обращения 08.06.2018).

12 express.js home page [Электронный ресурс] // Веб-структура для Node.js. – Режим доступа: <http://expressjs.com/> (дата обращения 08.06.2018).

13 AngularJS guide [Электронный ресурс] // Guide to AngularJS Documentation. – Режим доступа: <https://docs.angularjs.org/guide> (дата обращения 08.06.2018).

14 PassportJS home page [Электронный ресурс] // Аутентификация для Node.js. – Режим доступа: <http://www.passportjs.org/> (дата обращения 08.06.2018).

15 ORORO.TV - английский с удовольствием [Электронный ресурс] // Сервис для помощи в изучении англ. языка. – Режим доступа: <https://ororo.tv/ru> (дата обращения 08.06.2018).

16 ESL/ELL-English Grammar and Vocabulary Tests, Worksheets, Free ESL Resources [Электронный ресурс] // Сервис для помощи в изучении англ. языка. – Режим доступа: <http://learnenglishfeelgood.com/> (дата обращения 08.06.2018).

17 Animelon - Watch Anime, Learn Japanese [Электронный ресурс] // Сервис для помощи в изучении японского языка. – Режим доступа: <https://animelon.com/> (дата обращения 08.06.2018).

18 npm - package manager for Node.js [Электронный ресурс] // Менеджер пакетов для языка программирования JavaScript. – Режим доступа: <https://www.npmjs.com/> (дата обращения 08.06.2018).

19 The Unified Modeling Language Specification [Электронный ресурс] // Графический язык для визуализации, определения, построения и документирования распределенных объектных систем. – Режим доступа: <https://www.omg.org/spec/UML> (дата обращения 08.06.2018).

20 Anki - Powerful, intelligent flash cards. [Электронный ресурс] // Программа для облегчения запоминания. – Режим доступа: <https://apps.ankiweb.net/> (дата обращения 08.06.2018).

21 SuperМемо - учитесь быстро и забудьте о забывании [Электронный ресурс] // Метод обучения «СуперПамяти». – Режим доступа: <https://www.supermemo.com/ru/frontpage> (дата обращения 08.06.2018).

22 Node.js home page [Электронный ресурс] // Среда выполнения JavaScript. – Режим доступа: <https://nodejs.org/en/> (дата обращения 08.06.2018).

23 MongoDB home page [Электронный ресурс] // Система управления базой данных. – Режим доступа: <https://www.mongodb.com/> (дата обращения 08.06.2018).

24 Документация HTML5 [Электронный ресурс] // Ресурсы для разработчиков, от разработчиков. – Режим доступа: <https://developer.mozilla.org/ru/docs/HTML/HTML5> (дата обращения 08.06.2018).

25 BitBucket product features [Электронный ресурс] // Git для профессиональных команд. – Режим доступа: <https://bitbucket.org/product/features> (дата обращения 08.06.2018).

26 Trello home page [Электронный ресурс] // Управление проектами и их организация. – Режим доступа: <https://trello.com/home> (дата обращения 08.06.2018).

27 node-7zip package [Электронный ресурс] // Обертка Node.js для 7-Zip. – Режим доступа: <https://www.npmjs.com/package/node-7z> (дата обращения 08.06.2018).

28 relevancy package [Электронный ресурс] // Алгоритм сортировки (взвешивания). – Режим доступа: <https://www.npmjs.com/package/relevancy> (дата обращения 08.06.2018).

29 Github – subtitle file converter Node.js library and command line tool [Электронный ресурс] // GitHub – платформа разработки. – Режим доступа: <https://github.com/papnkukn/subsrt> (дата обращения 08.06.2018).

ПРИЛОЖЕНИЕ А

Распределение задач при разработке сервиса EduJetPlayer

В данном приложении рассмотрены глобальные задачи, выполненные каждым членом команды. Ниже приведен список задач для каждого члена команды:

а) Двоеглазов Николай:

- 1) Управление разработкой;
- 2) Проектирование архитектуры;
- 3) Проектирование интерфейса;
- 4) Тестирование сервиса;

б) Чагин Антон:

- 1) Разработка серверной части;
- 2) Разработка базы данных;
- 3) Выбор средств разработки серверной части;


в) Носов Максим:

- 1) Разработка видеоплеера;
- 2) Разработка клиентской части;
- 3) Выбор средств разработки клиентской части.

Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ

Заведующий кафедрой

 И. В. Евдокимов

подпись


«18» июня 2018 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.04 «Программная инженерия»

Разработка сервиса для повышения навыков владения иностранными языками
на основе техник «СуперПамяти» и захвата субтитров


Руководитель

 15.06.18
подпись, дата

к. т. н., доцент


И. В. Евдокимов

Выпускник

 15.06.18
подпись, дата

М. И. Носов

Выпускник

 15.06.18
подпись, дата

А. Д. Чагин

Выпускник

 15.06.18
подпись, дата

Н. Е. Двоглазов

Нормоконтролер

 15.06.18
подпись, дата

О. А. Антамошкин

Красноярск 2018

Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

УТВЕРЖДАЮ
Заведующий кафедрой



И.В. Евдокимов

подпись

«30» мая 2018 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студентам

А. Д. Чагин, М. И. Носов, Н. Е. Двоеглазов

Группа КИ14-17Б Направление (специальность) 09.03.04
Программная инженерия

Тема выпускной квалификационной работы «Разработка сервиса для повышения навыков владения иностранными языками на основе техник «СуперПамяти» и захвата субтитров»

Утверждена приказом по университету № 7834/с от 2018-05-30
Об утверждении тем ВКР, назначении руков-й, консульт-в

Руководитель ВКР И. В. Евдокимов, кандидат технических наук, доцент

Исходные данные для ВКР

Информация из открытого источника

Перечень разделов ВКР Введение, обзор методов и средств разработки сервиса для повышения навыков владения иностранными языками, проектирование сервиса «EduJetPlayer» и анализ его аналогов, разработка сервиса «EduJetPlayer», заключение

Перечень графического материала Презентационные слайды PowerPoint

Руководитель ВКР


подпись

И. В. Евдокимов

Задание принял к исполнению


подпись

А. Д. Чагин

Задание принял к исполнению


подпись

М. И. Носов

Задание принял к исполнению


подпись

Н. Е. Двоеглазов

«30» 05 2018 г.