

ОРГАНИЗАЦИЯ СОВМЕСТНОГО ПОЛЬЗОВАТЕЛЬСКОГО ВЗАИМОДЕЙСТВИЯ В РАМКАХ ОБРАЗОВАТЕЛЬНОЙ ПЛАТФОРМЫ ПРОЕКТА "ЦИФРОВОЙ ФАКУЛЬТЕТ"

А. Д. Жалнерчик

Белорусский государственный университет, г. Минск;

alexzhalnerchikz@mail.ru;

науч. рук. – А. В. Курочкин, преп.

Организация совместного доступа к онлайн-ресурсам – одна из важнейших составляющих в широком диапазоне систем коллаборативного интерактивного взаимодействия, актуальность которых значительно возросла в последние годы в связи с более активным развитием технологий для удалённой работы и дистанционного обучения. Ярким представителем таких систем являются онлайн-доски – общее пространство на веб-ресурсе с организацией совместного доступа нескольких пользователей для совместной работы, визуализации информации, поддержки обсуждений в обучении и бизнесе. В рамках работы рассматривается проектирование архитектуры такой системы на базе учебного облачного центра обработки данных платформы “Цифровой факультет”. Приведен анализ основных технологий и требований, спроектирована архитектура системы и рассмотрены основные сценарии взаимодействия.

Ключевые слова: онлайн-доска, цифровые технологии в образовании, websocket, системы совместного визуального доступа.

СТЕК ТЕХНОЛОГИЙ И ОСОБЕННОСТИ РЕАЛИЗАЦИИ

Ключевой особенностью любых систем коллаборативного взаимодействия является необходимость организации двухстороннего соединения между клиентским устройством. **WebSocket** – это протокол двунаправленной передачи данных прикладного уровня через длительное соединение при помощи стека протоколов TCP/IP, обеспечивающий обмен сообщениями между клиентом и сервером в режиме реального времени.

Коммуникация между клиентом и сервером осуществляется асинхронно, путём передачи сообщений как с сервера на клиент, так и с клиента на сервер, в любое время, пока соединение активно, в отличие от традиционной модели “запрос-ответ”, характерной для протокола HTTP.

Для создания легкорасширяемого сервиса без ограничения на количество пользователей была выбрана технология NodeJs для серверной части, а также библиотека React для клиентской части приложения. Выбор данных технологий обуславливается широкой поддержке событийной модели при помощи функций обратного вызова,

что значительно упрощает коммуникацию между серверной и клиентской частью приложения в асинхронном режиме при помощи WebSocket.

АРХИТЕКТУРА ПРИЛОЖЕНИЯ СОВМЕСТНОГО ВИЗУАЛЬНОГО ДОСТУПА

Для реализации поставленной задачи были выделены основные программные сущности:

- Пользователь (User)
- Доска (Canvas)
- Инструмент для редактирования (Tool)
- Слушатели событий для доски и инструментов (Observers)
- WebSocket

На рисунке 1 представлена схема архитектуры приложения, включающая в себя основные компоненты клиентской и серверной части.

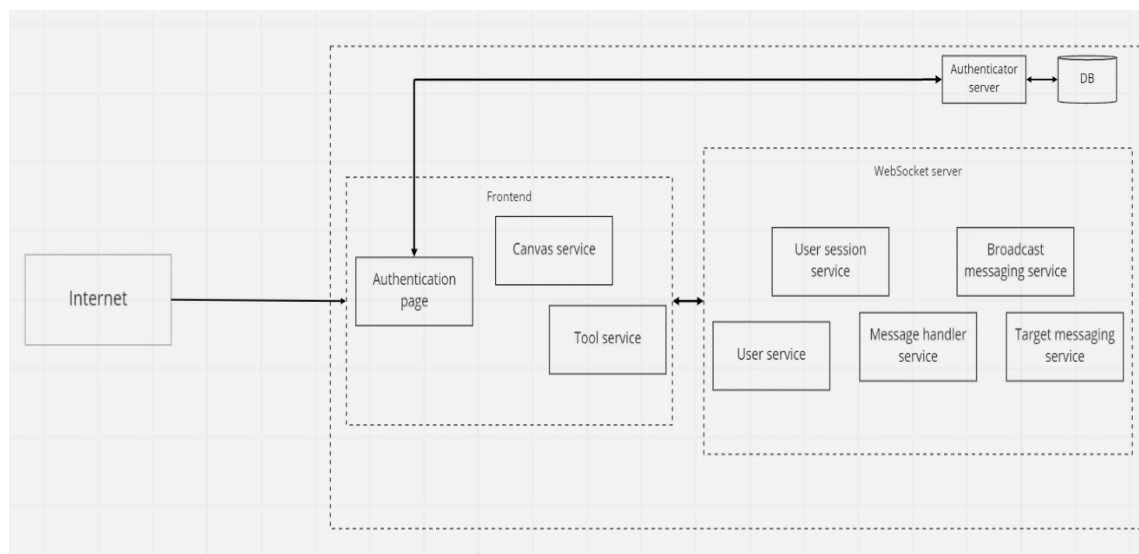


Рис. 1. Архитектура приложения

Страница аутентификации с помощью протоколов OAuth2.0 и OpenId Connect отправляет запрос на сервер аутентификации с использованием implicit схемы, которое предоставляет клиентскому приложению информацию о пользователе в идентификационном токене.

Сервисы canvas service и tool service отвечают, соответственно, за обработку событий по визуализации текущего состояния доски с помощью функций обратного вызова, и модификации этого состояния при помощи некоторых инструментов при действиях одного из пользователей с уведомлением сервера.

Сервис пользовательских сессий серверной части отвечает за хранение информации о сессии подключенных пользователей, а также за присоединение этого пользователя к некоторому экземпляру доски.

Сервис обработки сообщений серверной части отвечает за логику обработки действий пользователей в зависимости от типа сообщения. Эти сообщения актуализируют централизованное состояние экземпляра доски на сервере.

Сервис широковещательной передачи сообщений отвечает за рассылку сообщений, содержащих актуальное состояние доски, или последовательности операций по её актуализации, с серверной части на клиентскую.

Сервис целевой передачи сообщений позволяет осуществить передачу сообщений между отдельными клиентами напрямую.

Для развертывания приложения в рамках учебного облачного центра платформы “Цифровой факультет” описаны и спроектированы спецификации контейнеров для системы контейнеризации Docker.

ПРОЕКТИРОВАНИЕ МНОГОПОЛЬЗОВАТЕЛЬСКОГО ВЗАИМОДЕЙСТВИЯ

На рисунке 2 приведена sequence-диаграмма совместного взаимодействия двух пользователей со спроектированной системой.

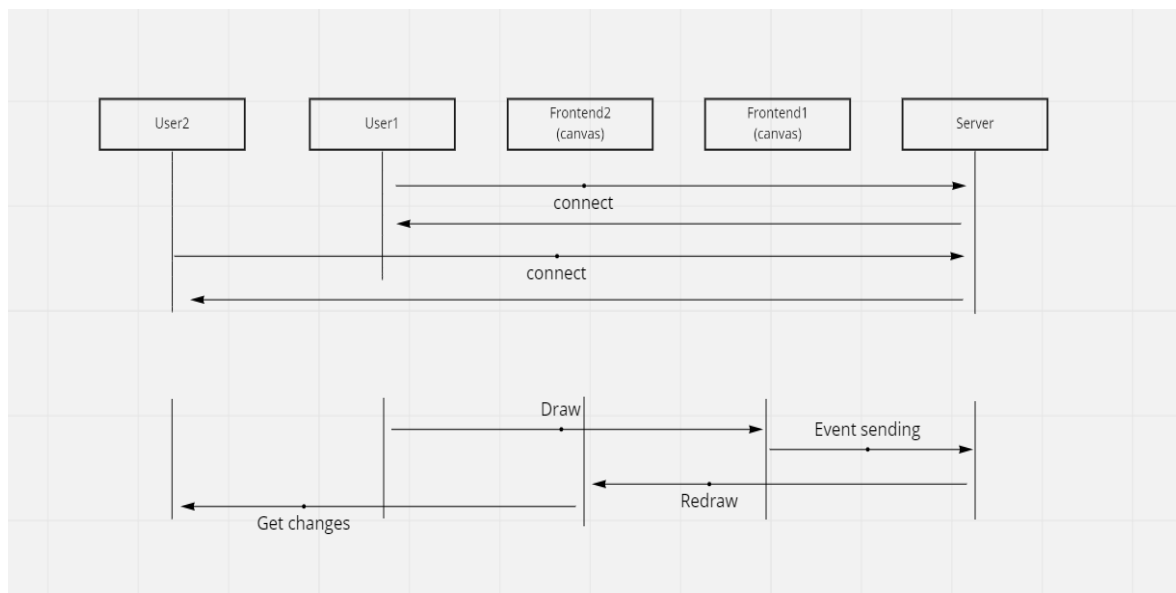


Рис. 2. Sequence диаграмма подключения двух пользователей к серверу и рисования на одной доске в режиме реального времени

На начальном этапе оба пользователя отправляют HTTP запрос серверу с целью установить соединение по протоколу WebSocket. После

того, как сервер одобрил переход на использование этого протокола, создаётся сессия, к которой присоединяются оба пользователя.

Когда один из пользователей совершает некоторое действие на своей доске, инструмент, с помощью которого осуществляется модификация, генерирует и отправляет на сервер сообщение, передавая в нем всю необходимую для повторения этой модификации информацию. Например, для инструмента “прямоугольник” сообщение содержит тип инструмента, координаты левого верхнего угла, ширину, высоту, цвет.

Сервер, получив сообщение, осуществляет актуализацию централизованного состояния со списком событий, и производит широковещательную рассылку пользователям, подключенным к данной сессии, тем самым сообщая, что произошли изменения, которые нужно применить.

Canvas каждого пользователя получает от сервера сообщение с указанными типом модификаций и параметрами, и применяет их на локальную копию доски, визуализированную у конкретного пользователя в веб-браузере.

Таким образом, представленная система позволяет поддерживать централизованное состояние доски на сервере и актуализировать и визуализировать это состояние у подключенных клиентов на основании их действий.

ЗАКЛЮЧЕНИЕ

В рамках проведенного исследования исследована технология Web-Sockets, разработана архитектура приложения для коллаборативного взаимодействия с онлайн-доской, спроектированы сценарии аутентификации с помощью внешнего источника, а также описан сценарий многопользовательского взаимодействия. Представленные результаты легли в основу прототипного приложения, подготовленного к развертыванию в рамках учебного облачного центра платформы “Цифровой факультет”.

Библиографические ссылки

1. *Andrew Lombardi* WebSocket: Lightweight Client-Server Communications, 2015 Mystic coders, LLC
2. *Varun Chopra* WebSocket Essentials: Building Apps with HTML5 WebSockets, 2015 Packt Publishing Ltd.
3. *Danny Coward* Java WebSocket Programming, 2014 McGraw-Hill Education (Publisher)