

**Создание веб-приложения для терминала самообслуживания кампусной системы Благовещенского государственного педагогического университета**

*Пюра Полина Константиновна*

*Благовещенский государственный педагогический университет*

*Студент*

*Антонов Антон Анатольевич*

*Благовещенский государственный педагогический университет*

*к.ф.-м.н., доцент кафедры информатики и методики преподавания информатики*

**Аннотация**

В данной работе приводится описание реализации веб-приложения для терминала самообслуживания кампусной системы ФГБОУ ВО «БГПУ» с целью автоматизации процессов

**Ключевые слова:** терминал самообслуживания, цифровизация, javascript, kotlin, vue.js

**Creation of a web application for the self-service terminal of the campus system of the Blagoveshchensk State Pedagogical University**

*Pyura Polina Konstantinovna*

*Blagoveshchensk State Pedagogical University*

*Student*

*Antonov Anton Anatolyevich*

*Blagoveshchensk State Pedagogical University*

*Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of Computer Science and Methods of Teaching Computer Science*

**Abstract**

This paper describes the implementation of a web application for the self-service terminal of the BSPU campus system in order to automate processes

**Keywords:** self-service terminal, digitalization, javascript, kotlin, vue.js

В последнее время автоматизация бизнес-процессов получила большую популярность и распространение. Выбор технологий для автоматизации основывается на задачах, которые должны быть решены. Так, автоматизация может быть реализована с помощью приложений на компьютере или на телефоне у конечного пользователя. Но не всегда эти устройства обладают всеми необходимыми характеристиками для решения поставленных задач. В

таких случаях отличным решением будет прибегнуть к использованию терминалов самообслуживания.

Терминалы самообслуживания – это оборудование для совершения самостоятельных платежей, выдачи чеков, квитанций и многих других действий. Их установка позволяет снизить нагрузку на кадровый состав магазина или другой организации [1].

В кампусной системе ФГБОУ ВО «БГПУ» привязка новых идентификаторов происходила в основном путём обращения в центр обслуживания клиентов, что значительно усложняло пользовательскую историю и приносило дополнительную нагрузку на работников центра. Исходя из этого возникла потребность вынести некоторые функциональные возможности, доступные только администратору системы, на самостоятельную обработку пользователем.

Цель исследования: разработка веб-приложения для терминала самообслуживания кампусной системы ФГБОУ ВО «БГПУ».

Веб-приложение разрабатывалось как одностраничное (англ. Single Page Application, SPA). Локализация приложения – русская.

Основная задача приложения – обработка запросов пользователя на привязку нового идентификатора и проверку существующего.

Веб-приложение разрабатывалось для сенсорного терминала со встроенным считывателем карт и QR-кодов.

Для создания модуля серверной части приложения применялись язык программирования Kotlin и фреймворк SpringBoot [2]. Клиентская часть была написана на языке JavaScript и его фреймворке Vue.js [3].

Разработанный модуль на сервере содержит в себе контроллер, который позволяет обрабатывать следующие запросы:

1. получение пользователя по логину и паролю;
2. получение пользователя по идентификатору;
3. получение списка всех правил доступа для пользователя;
4. добавление нового идентификатора и блокирование существующих;
5. запрос события со стороны сервера (long polling 20с);
6. отправка события серверу.

На стороне клиента была реализована необходимая графическая оболочка под существующие запросы с использованием CSS и HTML и необходимые функциональные возможности, в том числе для взаимодействия с сервером. В функциональные возможности входит:

1. Считывание и определение считавшегося идентификатора: карты или QR-кода.
2. Обработка статуса запросов с сервера с переходом на экран с сообщением об ошибке или об успешном завершении операции.
3. Привязка и проверка идентификатора.
4. Реализация поддержки сенсорной клавиатуры.
5. Реализация long polling для постоянной связи с сервером, и в случае существующего запроса с панели администратора – его обработка.

На рисунке 1 представлен главный экран веб-приложения.



Рисунок 1 – Главный экран веб-приложения

Для взаимодействия клиента и сервера используются HTTP(s) запросы. С помощью HTTP(s) был реализован процесс, который позволяет терминалу держать постоянную открытую связь с сервером – long polling [4].

Терминал с разработанным веб-приложением введен в эксплуатацию и находится на стадии поддержки. В дальнейшем приложение может легко расширяться и в него могут добавляться новые компоненты в соответствии с запросами.

### Библиографический список

1. Гаврилов Л. П. Инновационные технологии в коммерции и бизнесе: учебник для вузов. М.: Юрайт, 2023. 372 с.
2. Фелипе Г. Spring Boot 2: лучшие практики для профессионалов. СПб.: Питер, 2020. 464 с.
3. Документация Vue.js URL: <https://ru.vuejs.org>. (дата обращения: 1.09.2022)
4. Fain Y. et al. Enterprise Web Development: Building HTML5 Applications: From Desktop to Mobile. O'Reilly Media, 2014. 642 с.