

ПРОГРАММНАЯ ПОДДЕРЖКА ПРОЦЕССА ОБУЧЕНИЯ ПЕРСОНАЛА И АГРЕГАЦИИ ИНФОРМАЦИОННЫХ РЕСУРСОВ КОМПАНИИ В СФЕРЕ ИТ

Ставер И.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Космыкова Т.С. – ст. преп.

Информационные технологии поглощают и овладевают всеми сферами жизнедеятельности человека: начиная с сайтов-реклам и заканчивая проектированием сложных проектов.

Каждая фирма старается автоматизировать свой процесс с ростом предприятия. Сейчас сложно вспомнить, где бы не использовались ИТ, поэтому все большее число предприятий задумывается об автоматизации своей деятельности: своевременное определение спроса на те или иные товары, определение возможности производства товара с максимальной скоростью, высокой степенью качества, по лучшим ценам и в то же время с достаточной рентабельностью – это всё очень влияет на нормальное функционирование предприятия.

Но также с развитием ИТ на предприятии, появляется потребность того же самого предприятия в персонале, который знает и умеет выполнять бизнес задачи в сфере ИТ. Многие предприятия создают различные очные и онлайн курсы и тренинги для своего персонала чтобы повысить уровень квалификации в сфере ИТ. Но иногда такой подход не оказывается удобным для всех из-за того, что для некоторого персонала иногда проще прочитать какую-нибудь интересную и полезную статью на мобильном телефоне нежели посещать различные курсы. Предприятия понимают эту проблему и пытаются ее решить.

Одно из правильных решений - это создать мобильное приложение, где находились бы разные тренинги и курсы для персонала, которое бы позволяло обучаться «на лету». Приложение должно быть создано для актуальных мобильных платформ (iOS, Android). Одним из важных критериев успешности мобильного приложения это его дизайн. Соответственно дизайн должен быть максимально интуитивным и понятным пользователю. Помимо все различных тренингов и курсов, предприятие может внедрить туда и различные другие свои сервисы. Следовательно, приложения должно быть легко масштабируемым. Контент приложения должен быть актуальным, ибо если будет обратное, то смысл, обучать устаревшими технологиями персонал, нет.

Целью работы является создание мобильного приложения с помощью которого персонал может обучаться. Одной из главных составляющих данного приложения должен стать понятный и интуитивный дизайн. Данное приложение должно быть легко масштабируемым, а также иметь актуальный контент.

Приложение должно предоставлять:

- возможность навигироваться между сервисами и ресурсами компании;
- иметь технологию «чат-бота» для осуществления быстрых запросов;
- предоставлять сканнер баркодов;
- предоставлять интерактивную карту мира;
- возможность обучения различным программам компании.
- актуальные тренинги и программы для обучения персонала
- ссылки на первоисточники информации по тренингам и программам

В результате было создано клиент-серверное мобильное приложение, для актуальных мобильных платформ (iOS, Android) чтобы помочь персоналу компании в обучении различными тренингами и программами, а также доступом к различным ресурсам компании в удобном виде.

Для разработки использовалась программная среда WebStorm. Все изменения, внесенный в базу данных, осуществляются с помощью ODM Mongoose. Для создания графического интерфейса использовалась декларативная возможность описание интерфейса фреймворка React Native. В ней мы работали с JSX-файлами. К компонентам можно подключать CSS-файлы, а контролировать выполнение действий, добавление слушателей на таблицы и т.д. происходит с помощью привязки функции обработчика на компонент.

Для того, чтобы реализовать программу, которая сможет использоваться на предприятии, необходимо создать обмен данными между различными устройствами и базой данных. Поэтому было разработано клиент-серверное работающее с помощью протокола HTTP.

HTTP — это протокол, позволяющий получать различные ресурсы, например HTML-документы. Протокол HTTP лежит в основе обмена данными в Интернете. HTTP является протоколом клиент-серверного взаимодействия, что означает инициирование запросов к серверу самим получателем, обычно веб-браузером (web-browser). Полученный итоговый документ будет (может) состоять из различных поддокументов являющихся частью итогового документа: например, из отдельно полученного текста, описания структуры документа, изображений, видеофайлов, скриптов и многого другого.

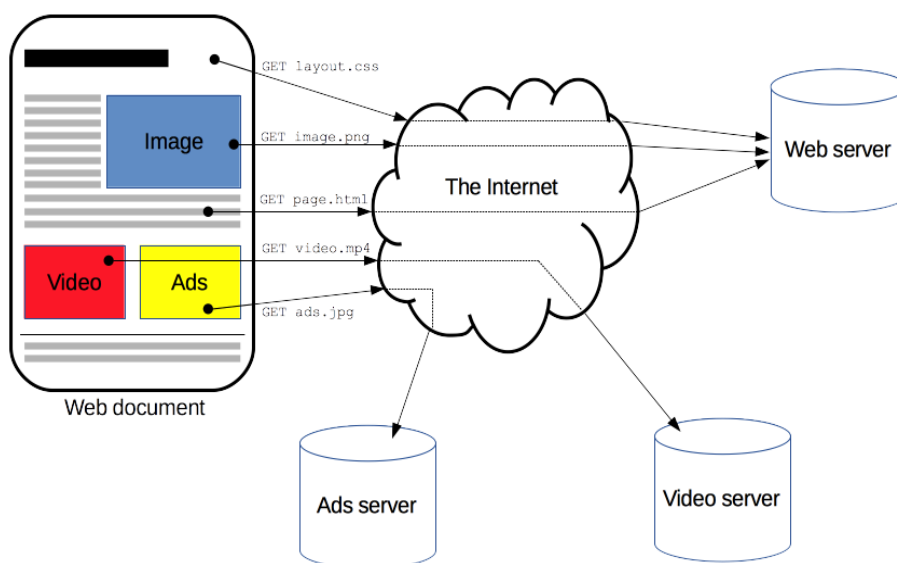


Рисунок 1 – Схема организации работы HTTP протокола

Клиенты и серверы взаимодействуют, обмениваясь одиночными сообщениями (а не потоком данных). Сообщения, отправленные клиентом, обычно веб-браузером, называются *запросами*, а сообщения, отправленные сервером, называются *ответами*.

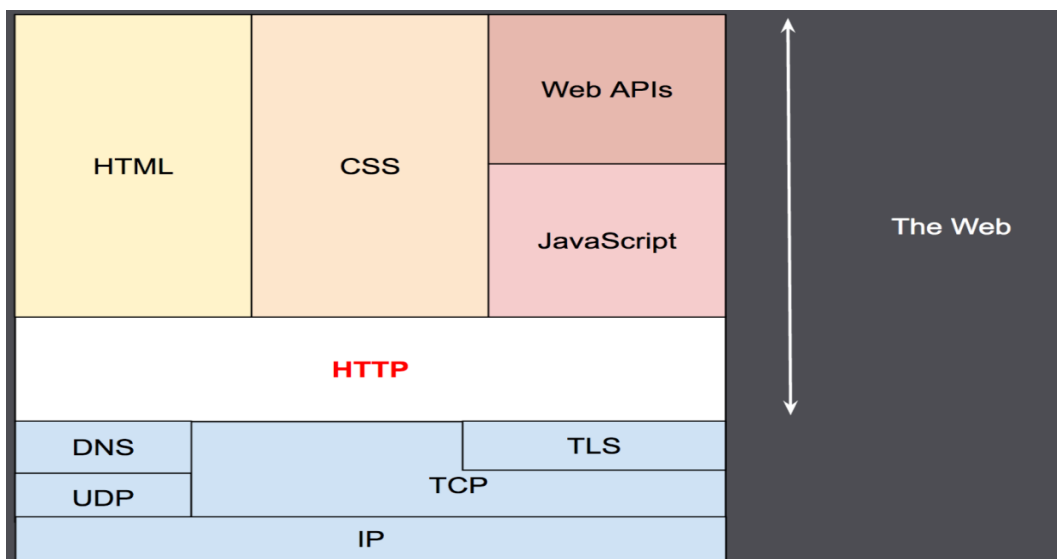


Рисунок 2 – Схема размещения HTTP протокола

На рисунке 1 и 2 показаны размещение HTTP протокола в стеке, а также примерная организация работы.

При написании кода приложения, а также при его компиляции и запуске мы будем придерживаться схемы изображенной на рисунке 3.

Как видно из схемы основным составляющим будет библиотека React. React с помощью взаимодействия с Virtual DOM будет эффективно обновлять состояние приложения.

Для большей гибкости в управлении состоянием приложения мы будем использовать библиотеку Redux. Она оперирует такими понятиями как: store, state, actions, reducers. Эта библиотека призвана объединить все состояние приложения в один большой объект.

Для гибкости в формировании запросов к сторонним API мы будем использовать библиотеку Redux-Saga. Это библиотека помогает делать так называемые «side effects» более предсказуемыми, а также позволяет формировать эти эффекты в синхронном стиле.

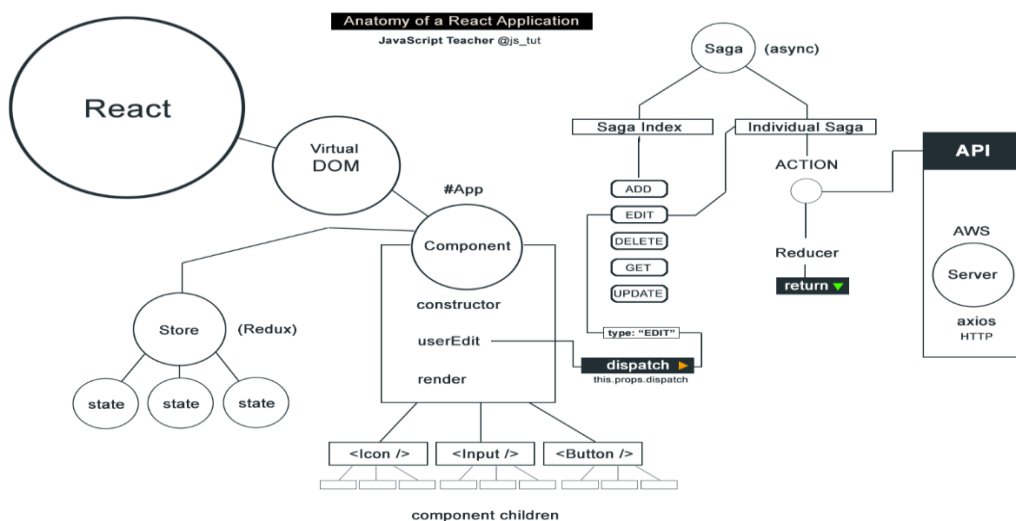


Рисунок 3 – Схема работы приложения

56-я научная конференция аспирантов, магистрантов и студентов БГУИР, 2020 г.

На схеме представлен также и сервер. При этом сервер будет работать на технологии Node.JS.

Список использованных источников:

1. Официальный сайт React Native. [Электронный ресурс] - Режим доступа: <https://facebook.github.io/react-native/>