

Таким образом, для работы с шаблонами графических файлов можно использовать GDI+, который может работать с любым типом изображения.

УДК 004.42

Ивашко С.П., Капуста Е.В.

## **СИСТЕМА ОНЛАЙН-БРОНИРОВАНИЯ**

*БНТУ, Минск*

*Научный руководитель Круглик Т.М.*

Наша работа посвящена созданию системы для осуществления бронирования мест в маршрутных такси регулярного следования.

Основной задачей комплекса программ является автоматизация функций в части клиента, водителя, менеджера и диспетчера в процессе бронирование мест, отмена брони и её редактирование, регистрации клиентов, водителей и диспетчеров в системе, составление расписания движения, графика работы водителей, статистический учет и прочее.

К данной системе предъявляются требования актуализации данных в режиме реального времени и бесперебойной работы 24/7 под нагрузкой.

Нами были созданы две версии системы.

Первая версия полностью разработана и запущена. Необходимость проектирования второй версии возникла в связи с осознанием возможности оптимизации, уточнение некоторого функционала системы.

При разработке первой версии нами использовался следующих набор компьютерных технологий:

- серверная платформа NodeJS;
- документно-ориентированная база данных MongoDB;
- язык программирования C# и WindowsForms для разработки Desktop приложений;
- Xamarin.Android и Xamarin.iOS для мобильных приложений.

Разработанный на базе описанных выше технологий комплекс обладает рядом преимуществ и недостатков.

Недостатки выявленные в процессе использования системы: изначально не гибко спроектированная база, которая в последствии не смогла обеспечить внедрение и выполнение некоторых новых функций; ошибки в работе (баги) связанные с взаимодействием клиента и сервера, а также некоторыми не самыми дружелюбными к пользователю моментами; множество «лишних» обращений к серверу; отсутствие автоматического обновления данных в режиме реального времени, требуется постоянно обновлять данные вручную или вызывая методы обновления после определенных операций.

Наличие перечисленных недостатков привело к мысли о необходимости разработке более гибкой и универсальной системы, обладающей большим функционалом и скоростью работы. Было принято решение о том, что необходимо использовать такой стек технологий, который бы позволял использовать максимально идентичный код на разных платформах.

Таким образом для второй версии системы был выбран следующей стек технологий: JavaScript и TypeScript как основные языки программирования логики как на сервере, так и на клиенте; серверная платформа NodeJS; Sails.js – JavaScript фреймворк реального времени. Разработчики проекта вдохновились изяществом архитектуры Ruby on Rails и постарались воспроизвести ее на JavaScript.

Результатом труда стал перспективный MVC инструмент для создания современных web-приложений. Лучше всего фреймворк проявляет себя в приложениях реального времени, то есть там, где требуется поддержка данных в самом актуальном состоянии; SocketIO – JavaScript-библиотека для веб-приложений позволяющая осуществлять двунаправленную связь в режиме реального времени. Она работает на каждой платформе, браузере или устройстве, в равной степени ориентируясь на надежность и скорость; MongoDB – документно-ориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Классифицирована как NoSQL, использует JSON-подобные документы и схему базы данных; Angular – JavaScript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений; NativeScript – открытый фреймворк для создания нативных мобильных приложений с

помощью Angular, TypeScript или JavaScript; Electron – фреймворк для создания кроссплатформенных десктопных приложений на JavaScript, HTML, и CSS.

На наш взгляд перечисленный набор технологий обеспечивает выполнение всех требований, предоставляемых к системе как изначально, так и после анализа первой версии.

Код из Web приложения на Angular оборачивается с помощью Electron и на выходе мы имеем Desktop приложение. Используя NativeScript мы привязываем всю логику описанную с помощью Angular к кроссплатформенным компонентам интерфейса для Android и iOS. Sails.js позволяет, используя сокет поддерживать данные во всех приложениях в актуальном состоянии и моментально синхронизировать их.

Таким образом разрабатываемая в настоящее время система позволит исключить перечисленные выше недостатки первой версии системы. Используя перечисленные технологии, система обеспечивает выполнение предъявляемых к ней требований, обладает высоким функционалом и является востребованной.

УДК 004.42

Ивашко С.П., Капуста Е.В.

## **РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА NATIVESCRIPT И ANGULAR**

*БНТУ, Минск*

*Научный руководитель Дробыш А.А.*

В настоящее время почти у каждого человека в кармане лежит смартфон с доступом к интернету. И в каждом из этих миллионов устройств находится множество предустановленных производителем или скаченных и установленных самим пользователем из специализированного магазина приложений.

Двумя самыми популярными мобильными платформами являются Android и iOS. Каждая организация или сервис стараются помимо сайта и приложения под персональный компьютер, предоставить еще и приложения под эти платформы. Для разработки приложений под Android используется язык