

МАТЕМАТИКĂ ŞI INFORMATICĂ

СОЗДАНИЕ МНОГОФУНКЦИОНАЛЬНОЙ СИСТЕМЫ ДЛЯ ПЛАНИРОВАНИЯ ЗАДАЧ «SCHECOLER»

Виктория АБАЛИНА, студентка

CZU: 004.415.2

viktoria.zheneral@gmail.com

This article is intended to introduce the available methods for creating a desktop application for scheduling tasks using web development tools. The competitors in the market and the available tools were analyzed. As a result, the Electron framework and the Node.js platform were chosen as the main implementation tool. A list of the intended functionalities was then compiled. The application was created based on the analysis and the formulated requirement.

В современном мире, зависящим от компьютерных технологий и стремящемся к оптимизации всех процессов, важным аспектом является организация личного времени. В настоящее время на рынке компьютерных программ представлено огромное количество различных календарей с возможностью добавления заметок, а также приложений для планирования. Однако большая часть существующих на данный момент приложений обладают сложным и неудобным пользовательским интерфейсом, функционально ограничены или, напротив, содержат излишние функциональные особенности.

В дипломной работе были рассмотрены возможности создания простого в использовании и понятного пользователю, но многофункционального приложения для планирования задач и организации времени с поддержкой кроссплатформенности, то есть со способностью работать с несколькими операционными системами.

Первоначально требовалось изучить готовые решения данной задачи, а именно проанализировать конкурентов на рынке. Было выявлено, что одним из недостатков являлось оснащение приложений сложными для восприятия функциональностями, учитывая, что главные требования клиента к подобным программам – простота и удобство. Немаловажным критерием пользователей

при выборе приложения является графический интерфейс, то есть удобство использования и общий визуальный стиль.

Следующий подготовительный этап состоял в выборе подходящих инструментов реализации. Практически любой язык программирования (или, точнее, механизм выполнения языка программирования), обладающий способностью напрямую взаимодействовать с машинным кодом, может быть использован и для реализации десктопных приложений с графическим интерфейсом. Таким образом, спектр доступных инструментов включает в себя почти все языки программирования, созданные на протяжении последних 60-ти лет. Основываясь на результатах анализа по выбранным критериям (комфорт разработки, возможность настройки графического интерфейса и рациональность использования), выбор был сделан в пользу платформы Node.js, фреймворка Electron, инструментов веб-разработки HTML, CSS, языка программирования Javascript и СУБД SQLite. JavaScript – мультипарадигменный язык программирования, поддерживающий объектно-ориентированный, функциональный и императивный подходы. Изначально JavaScript задумывался как один из простых скриптовых языков, встраиваемых в HTML-код и предназначенных для объединения в единое целое различных компонентов веб-страниц [1]. Сейчас JavaScript в основном используется в качестве инструмента предоставления программного доступа к объектам различных приложений, и наиболее широкое применение находит в веб-браузерах для обеспечения интерактивности на веб-страницах [2], поскольку является однопоточным и асинхронным. Node.js – программная среда разработки, основанная на движке V8, транслирующем JavaScript в машинный код. Данная среда разработки выполняет JavaScript-код вне веб-браузера и, соответственно, предоставляет возможность использовать JavaScript для написания инструментов командной строки и сценариев, исполняемых на стороне сервера [3]. Electron (также известный как atom shell) – это фреймворк, позволяющий разрабатывать кроссплатформенные приложения с графическим интерфейсом для настольных операционных систем, используя инструменты веб-технологий [4].

Поскольку задача дипломной работы состояла в разработке многофункциональной системы для планирования задач, первым этапом планирования проекта было выбрано составление списка необходимых функциональностей, объясняемых, прежде всего, спецификой будущего приложения. Было принято решение реализовать три типа отображения календаря (день, неделя и месяц), добавление, редактирование, копирование и удаление нового плана и сортировка всех планов по стандартно заданным категориям. Отображение планов происходит в зависимости от их типа: безвременные задачи отсортированы по степени важности, мероприятия – по времени их начала. Далее была реализована регистрация и авторизация по уникальным логину и паролю для дифференцирования планов пользователей, совместно использующих программу на одном устройстве.

Перед началом процесса создания любого приложения необходимо подготовить и протестировать среду разработки. Изначально требовалось установить платформу Node.js и загрузить необходимые для разработки побочные модули менеджером пакетов npm. Важно помнить, что модули первоначально написаны для операционной системы Linux, поэтому потребовалась повторная сборка определенных компонентов, например, программного модуля для работы с базами данных SQLite. После того, как отдельные модули были разработаны в соответствии с ранее определенными требованиями к функциональным особенностям программы, их требовалось объединить в цельный проект. Для готового приложения была создана программа-инсталлятор, упрощающая процесс непосредственной установки.

Приложение, созданное как результат дипломной работы, идентично рыночным конкурентам, было проанализировано для выявления сильных и слабых сторон как с точки зрения самого приложения, так и с учетом использованных инструментов. Программа была протестирована на различных операционных системах, обнаруженные дефекты были успешно устранены.

Разработанная программа может быть применена в качестве электронной записной книжки деловыми людьми для орга-



низации планов и экономии личного времени. Схожим образом приложение может использоваться как календарь с возможностью добавления заметок.

Литература:

1. ФРИМЕН, Э., РОБСОН, Э. *Изучаем программирование на JavaScript*. Издательство “Питер”, 2018.
2. ФЛЭНАГАН, Д. *JavaScript. Карманный справочник. Сделайте веб-страницы интерактивными!* Москва: издательский дом “Вильямс”, 2015.
3. *Professional Node.js: создание масштабируемого программного обеспечения на основе JavaScript*. John Wiley & Sons, 2012.
4. БРАЙТ, П. *Microsoft's new Code editor is built on Google's Chromium* (англ.). Ars Technica.com, 2015.

*Рекомендовано
Георгий ЛАТУЛ, преподаватель*