

*Акишин Андрей Валерьевич
студент 2 курса магистратуры, факультета управления
Московский автомобильно-дорожный государственный технический
университет,
Россия, г. Москва
e-mail: luxlite_oleg@mail.ru*

*Научный руководитель: Баринов К.А., кандидат технических наук, доцент
кафедры «Автоматизированные системы управления»
Московский автомобильно-дорожный государственный технический
университет
Россия, г. Москва*

*Бизенков Олег Игоревич
студент 2 курса магистратуры, факультета управления
Московский автомобильно-дорожный государственный технический
университет,
Россия, г. Москва*

*Научный руководитель: Брыль В.Н., кандидат технических наук,
доцент кафедры «Автоматизированные системы управления»
Московский автомобильно-дорожный государственный технический
университет
Россия, г. Москва*

ОСОБЕННОСТИ РАЗРАБОТКИ, ТЕСТИРОВАНИЯ И ВНЕДРЕНИЯ КОМПЬЮТЕРНЫХ ПРИЛОЖЕНИЙ

Аннотация: в статье рассматривается логическая составляющая компьютерных приложений обязывает присутствия оператора (человека, который работает с программой), заключающие в себе всю совершенную функциональность и способность работать отдельно на любой машине обособленно от других приложений. Для их работы нужны лишь достаточные аппаратные ресурсы компьютера. Само приложение и набор библиотек, который включает в себя функции для работы с приложением. При анализе последних исследований, была поставлена проблема о необходимости внедрения компьютерных приложений.

Ключевые слова: компьютерное приложение, платформа nw.js и electron.js, интерфейс браузера, главные компоненты, настройка приложения, платформа Cuba, масштабируемость, универсальное решение, системное окно, кроссплатформа, визуальный конструктор.

Akishin Andrey Valeryevich
2nd year master student, faculty of management
Moscow automobile and road state technical University,
Moscow, Russia

Scientific adviser: Barinov K.A.,
candidate of technical sciences, associate professor,
associate professor of the Department "Automated control systems"
Moscow automobile and road transport state technical University
Russia, Moscow

Bizenkov Oleg Igorevich
2st year master student, faculty of management
Moscow automobile and road state technical University,
Moscow, Russia

Scientific adviser: Bryl V.N.,
candidate of technical sciences, associate professor,
associate professor of the Department "Automated control systems"
Moscow automobile and road transport state technical University
Russia, Moscow

FEATURES OF DEVELOPMENT, TESTING AND IMPLEMENTATION OF COMPUTER APPLICATIONS

Abstract: *The article discusses the logical component of computer applications requires the presence of an operator (a person who works with the program), which includes all the perfect functionality and the ability to work separately on any machine separately from other applications. They only need sufficient computer hardware resources to work. The app itself and a set of libraries that includes functions for working with the app. When analyzing the latest research, the problem was raised about the need to implement computer applications.*

Key words: computer application, platform nw.js and electron.js, browser interface, main components, application configuration, Cuba platform, scalability, universal solution, system window, cross-platform, visual designer.

Компьютерные приложения встречаются и многопользовательскими. К примеру, редактор файлов, который в зависимости от логина и пароля, введенных при запуске, будет давать доступ к многообразным файлам. И программа, и файлы находятся на одном компьютере, просто вырабатывается локальное разграничение доступа для различных пользователей. Компьютерное

приложение решает множество проблем, такие как, желание, чтобы веб-приложение имело доступ к файловой системе, могло обрабатывать различные системные процессы и имело доступ к буферу обмена с сочетанием клавиш. Таким образом, оно может работать в фоне, запускаться при старте системы и полностью оффлайн, и не спрашивать доступ к камере или микрофону, т.е. когда мы пользуемся приложением десктопным, все права определяются автоматически. Для того, чтобы в приложении возможно было использовать Web Jay, Web Audio и другие технологии, они должны работать на одной платформе. [1]

Компьютерное приложение позволяет создавать системные окна, т.е. не только браузерный интерфейс внутри браузера, но и элементы интерфейса самой системы, т.е. окошки, иконки трея, контекстные меню [2]. Тем самым позволяет совершенно полностью пользоваться унификацией и делать кросс-доменные запросы, менять User Agent, т.е. возможность пользователю представляться кем угодно, а главное работа в полноценном полноэкранный режиме, что является крайне полезным для программ, где используются терминалы. [3] На сегодняшний день есть несколько программ, которые также пытаются реализовать все вышеперечисленные моменты, это компании такие как Apple Plush, но используя данные программы, вероятность столкнуться с проблемой – длинная, тяжелая для восприятия документация, очень велика.

Одна из немногих платформ, которая способна предоставить разработку кроссплатформенных компьютерных приложений, это платформа NW.js (ранее node-webkit), придуманная Roger Wang, разработчиком из Китая в 2011 году и Electron.js. Основная идея данного проекта, показать, что компьютерное приложение запускается на любой машине и там же может делать изменения. Два важных компонента nw.js, благодаря которым и осуществляется данная разработка – это браузер Chrome и IO, объединённые в единый контекст V8 - javascript-движок, у которого открытый исходный код. [4], [5]

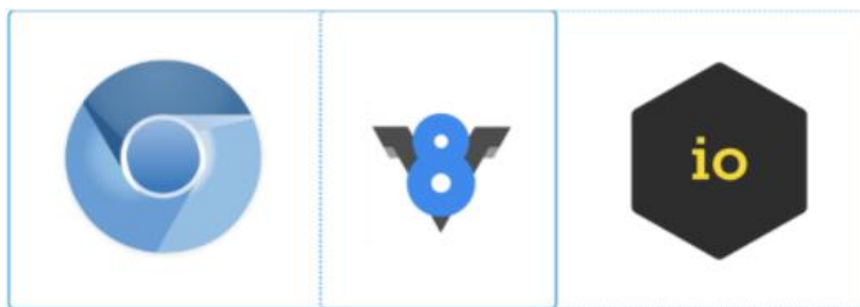


Рис.1 – Изображение компонентов компьютерного приложения

Для того, чтобы выбрать платформу, на основе которой будет приведен пример внедрения и использования компьютерного приложения, проведем сравнительный анализ nw.js и electron.js. Данные оформлены в виде таблицы, представленной ниже [6].

Таблица 1- Сравнительный анализ

	Nw.js	Electron.js
Время работы браузера	chromium	libchromiumcontent
Механизм макета	Blink/Webkit537	Webkit537
Chrome версия	41	44
Входная точка	Html/javascript	javascript
Chrome тех.поддержка	+	-
Windows Apps Store/XP	+	-
Mac Apps Store	+	+

Из данной таблице можно сделать вывод, что Electron.js является вытекающей платформой из nw.js. Рассмотрим поэтапную реализацию создания и тестирования компьютерного приложения на платформе Electron.js. Принцип работы Электрона основан на двух типах процессов:

Первый тип – основной процесс, который отвечает за интеграцию и взаимодействие с GUI операционной системы. Под этим понятием скрывается интеграция в док на OS X или панель задач на Windows, а также сворачивание в

трей, полноэкранный режим и прочие обыденные для ОС вещи. Такой процесс может быть запущен только один раз на всю жизнь приложения [7], [8].

Второй тип – процесс рендеринга, отвечающий за отображение окна браузера, в котором с помощью одной магической строчки может быть открыта страница приложения или любая другая веб-страница. Таких процессов может быть произвольное число. За создание процесса рендеринга отвечает основной процесс [9]. Для того, чтобы породить основной процесс используется следующая схема рождения приложения:

- Электрон читает `package.json` и ищет в нём секцию `main`, в которой определен основной файл приложения. Этот файл далее в статье я буду называть «точкой входа».

- Затем происходит обработка «точки входа» и создаётся основной процесс, который в свою очередь, при желании разработчика, открывает какую-либо страницу или страницы, то есть создаёт окно браузера. А если говорить точнее, то порождает процесс или процессы рендеринга [9]. Если попытаться графически изобразить жизнь приложения, построенного на Электроне, то она будет иметь следующий вид:

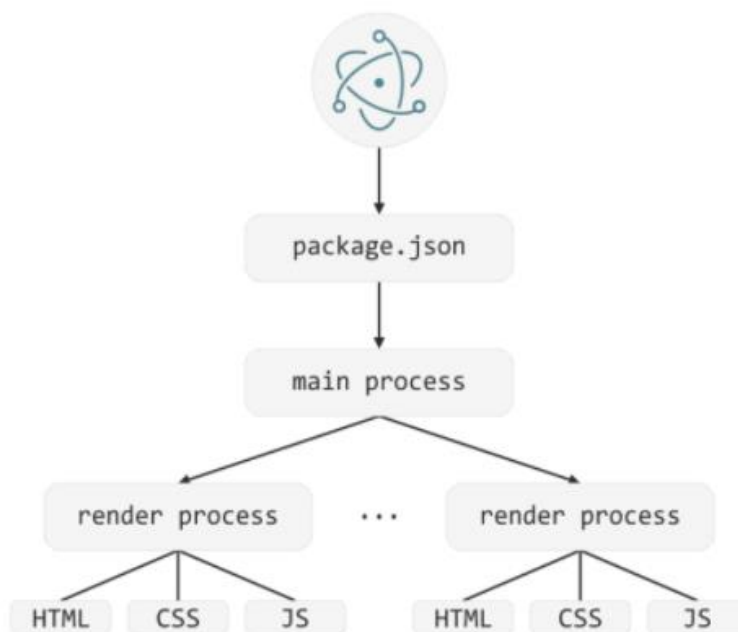


Рис.2 – Построение Electron

В дальнейшем происходит настройка окружения, WebStorm, и после создания приложения появляется окно, которое имеет служебное меню с небольшим набором стандартных пунктов. Служебным оно называется из-за того, что после сборки проекта, как вы увидите позднее, оно исчезает, далее осуществляется распространение приложения [10].

Наглядным примером инновационного развития и карьерного успеха благодаря десктопным приложениям является крупнейшая частная служба такси Addison Lee, находящиеся в Лондоне, но активно сотрудничающая с Российскими инновационными центрами. Высокоуровневая платформа CUBA, позволяющая создавать бизнес-приложения на Java, предоставляет богатый набор UI компонентов, визуальный конструктор форм, функции аудита, контроля доступа и генератор отчетов. На основе данной платформы были созданы такие приложения, как Sherlock – полнофункциональное решение для служб такси, которое автоматизирует все аспекты управления бизнесом - от заказа такси и назначения машин до биллинга и приложений для клиентов [11].



Рис.3 – изображение CUBA.platform

Sherlock совмещает высокую гибкость и масштабируемость решения с низкой стоимостью внедрения и программой непрерывной технической поддержки. Система построена на платформе CUBA в сочетании с такими компонентами как Google Maps, а также мобильными клиентами на Android и iOS, что позволило создать наиболее комплексное и целостное на текущий момент решение для управления такси в мире [11]. Данное приложение

существует, как и на лондонском сайте addisonlee.com, так и на русифицированном haulmont.ru. предоставление таких удобств с помощью приложений, позволило компании на сравнительно небольшой срок, стать одним из крупнейших инновационным проектом. Данные выводы были сделаны на основе детального рассмотрения публикаций компании Addison Lee, запустившая в работу революционную систему бронирования авто «Shamrock», разработанную Magenta Technology [11].

Подводя итог, можно сказать, что кроме вышеперечисленного, у компьютерных приложений нет никаких проблем с поддержкой старых версий программ и обратной совместимостью. И наконец, данные приложения позволяют своим пользователям быть по-настоящему мобильными. Можно работать в сети, сохранять результаты своей работы на сервере и, в случае необходимости, иметь к ним доступ отовсюду (где есть выход в Интернет, естественно).

Список литературы:

1. Свентицкий П., Иванова Н.А. Инструменты кроссплатформенной разработки мобильных приложений. Новосибирск: СибАК, 2014.
2. Савин, Р. Тестирование Дот Ком. Минск: Дело, 2007. 312 с.
3. Appcelerator Titanium: официальный сайт разработчиков Appcelerator [Электронный ресурс] // Режим доступа. URL: <http://www.appcelerator.com> (дата обращения: 28.05.2020 г.).
4. Kony: официальный сайт компании Kony [Электронный ресурс] // Режим доступа. URL: <http://www.kony.com> (дата обращения: 28.05.2020 г.).
5. PhoneGap: официальный сайт разработчиков PhoneGap - [Электронный ресурс] // Режим доступа. URL: <http://www.phonegap.com> (дата обращения: 28.05.2020 г.).
6. DailyComm. Коммуникации в ИТ-бизнесе: [Электронный ресурс] // Режим доступа. URL: <http://www.dailycomm.ru> (дата обращения: 28.05.2020 г.).

7. Компьютерное приложение / создание – 2016 г.г. [Электронный ресурс] // Режим доступа URL: https://dev.1c-bitrix.ru/learning/course/index.php?&COURSE_ID=52&LESSON_ID=5374 (дата обращения: 28.05.2020 г.).
8. Винченко, И. Автоматизация процессов тестирования. СПб.: Питер, 2008. 211 с.
9. Безъязыкова Н.А., Яковлева М.С. Актуальные проблемы авиации и космонавтики / насыщенные интернет-приложения – Том 1 004.5-2015 [Электронный ресурс] // Режим доступа: URL: <http://cyberleninka.ru/article/n/nasyschennye-internet-prilozheniya> (дата обращения: 28.05.2020 г.).
10. HAULMONT / передовые решения Addicon Lee 2016г.г [Электронный ресурс] // Режим доступа: URL: <https://www.haulmont.ru/blog/mar-02-2017/679> (дата обращения: 28.05.2020 г.).
11. Построение Electron приложения / Введение – 2015 г.г. [Электронный ресурс] // Режим доступа: URL: <https://canonium.com/articles/electron-desktop-app-introduction> (дата обращения: 28.05.2020 г.).