

способы передачи информации. С появлением электронных примеров также были улучшены способы не только обмена данными, но также их запись и хранение.

Разработанное приложение предоставляет пользователю возможность удобного хранения, публикации и просмотра книг, а также других текстовых файлов, хранящихся в виде электронных документов. Помимо этого, при наличии аудиоверсии книги она также может быть опубликована в данной программе и прослушана. При этом доступ к файлам имеют только авторизованные пользователи, которые при желании могут выдать доступ иным пользователям.

Сервер приложения был реализован на языке Typescript. Typescript – расширение языка Javascript, которое вводит поддержку статической типизации, а также позволяет улучшить обработку исключительных ситуаций, возникающих при работе программ, и ускорить время выполнения разрабатываемого программного обеспечения. При разработке используется платформа Node.js, созданная в 2009 году Райаном Далем, также используются фреймворки Nest.js [1] и GraphQL [2], а также иные сторонние библиотеки.

Сам текст электронных документов располагается в заранее подготовленных файлах, хранящихся на сервере, а ссылки на них и дополнительная информация расположены в базе данных. В качестве СУБД используется MongoDB.

Интерфейс веб-приложения реализован на языке JavaScript с использованием библиотеки React [3]. Оформление компонентов приложения выполнено при помощи расширения языка CSS – препроцессора SASS. Прототип интерфейса выполнен в среде Figma.

Тестирование HTTP-запросов выполнялось с помощью HTTP-клиента Postman.

Литература

1 Официальная документация Nest.js [Электронный ресурс]. – Режим доступа : <https://docs.nestjs.com>. – Дата доступа : 04.03.2021.

2 Официальная документация GraphQL [Электронный ресурс]. – Режим доступа : <https://graphql.org/learn>. – Дата доступа : 05.03.2021.

3 Официальная документация React [Электронный ресурс]. – Режим доступа : <https://ru.react.js.org/docs/getting-started.html>. – Дата доступа : 20.04.2021.

А. В. Долженко

Науч. рук. Н. Б. Осипенко,

канд. физ.-мат. наук, доцент

ПРОЕКТИРОВАНИЕ ПРОТОТИПА ГОЛОСОВОГО ПОМОЩНИКА «ОМИКРОН»

Голосовой ассистент – это личный помощник человека, который внедряется в смартфоны, компьютеры и другие устройства и готов помогать найти интересующую пользователя информацию или выполнять различные задачи. Согласно опросу, проведенному PwC (Price waterhouse Coopers), около 75 % людей пользуются голосовым помощником, а также около 20 % используют его для навигации в автомобиле.

Для реализации прототипа голосового помощника задействован язык программирования Python, его обширные библиотеки и модули. Команды, осуществляемые голосовым помощником, реализуются как при помощи Интернета, так и без него, а также используется база данных MS SQL Server. Схема работы прототипа голосового помощника «Омикрон» приведена на рисунке 1.

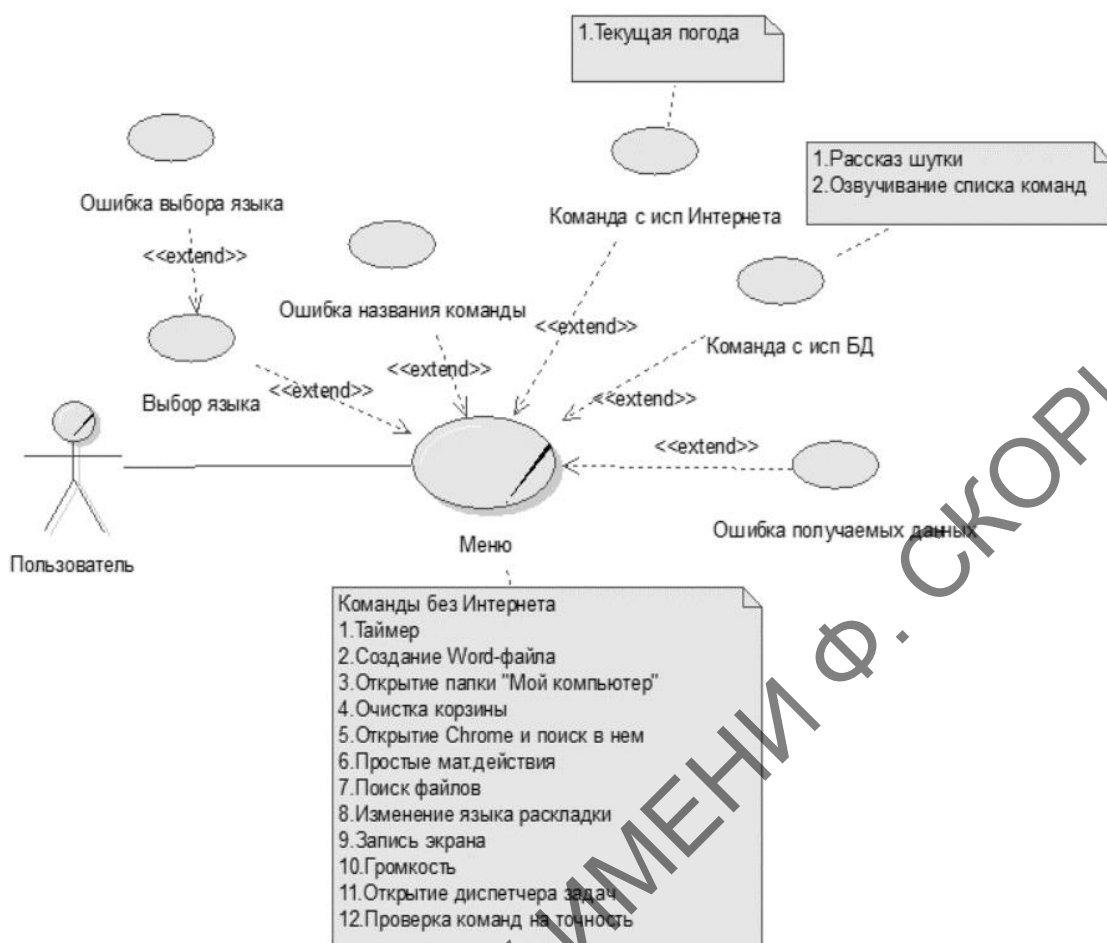


Рисунок 1 – Схема работы прототипа голосового помощника «Омикрон»

Данный прототип голосового помощника функционирует на русском и английском языках. Предусмотрена обработка таких проблем, как выбор языка, на котором будет функционировать голосовой помощник, нераспознанная речь и получение некорректных данных для выполнения команды. При возникновении одной из выше перечисленных команд голосовой помощник уведомит о конкретной проблеме и будет ожидать последующего запроса от пользователя.

А. С. Зайцев
 Науч. рук. *Н. Б. Осипенко*,
 канд. физ.-мат. наук, доцент

КОНЦЕПЦИЯ ТЕСТИРОВАНИЯ РЕШЕНИЯ ЗАДАЧИ ВЕБ-ПРИЛОЖЕНИЕМ ПО ИЗУЧЕНИЮ ЯЗЫКА ПРОГРАММИРОВАНИЯ JAVASCRIPT

Материалы посвящены описанию концепции проверки выполненного практического задания разработанным клиент-серверным приложением по изучению языка программирования JavaScript. Онлайн-обучение программированию без практики невозможно. Решив задачу, пользователь отправляет свой ответ; при тестировании находятся все тесты для этой задачи из коллекции тестов по ID задачи, пришедшей на тестирование; проверяются все присланные решения задачи путем получения