

3 Информация об игре. Данный пункт отображает на экран форму, на которой содержится ознакомительная информация об игре «Судoku» для новичков. На примере объясняются основные правила и советы по игре. Форма изображена на рисунке 6.



Рисунок 6 – Форма «Об игре»

Игровое приложение «Судoku» создано для операционной системы Windows, для написания кода использовался объектно-ориентированный язык C++ [3, 4]. Приложение реализовано в среде программирования C++ Builder 6 [5].

Литература

- 1 Правила Судoku [Электронный ресурс]. – Режим доступа: <https://sudoku.com/ru/pravila-sudoku/>. – Дата доступа: 08.01.2024.
- 2 Уроки по Судoku от чемпиона мира Томаса Снайдера [Электронный ресурс]. – Режим доступа: <https://sudoku.com/ru/kak-igrat/uroki-po-sudoku-ot-chempiona-mira-tomasa-snajdera/>. – Дата доступа: 11.01.2024.
- 3 Доусон, М. Изучаем C++ через программирование игр / М. Доусон. – Санкт-Петербург : Питер, 2021. – 352 с.
- 4 Лафоре, Р. Объектно-ориентированное программирование в C++ / Р. Лафоре. – Санкт-Петербург : Питер, 2022. – 928 с.
- 5 Архангельский, А. Я. C++ Builder 6. Справочное пособие. Книга 1. Язык C++ / А. Я. Архангельский. – Москва : БИНОМ, 2004. – 544 с.

УДК 004.91

Р. А. Ермоленко

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ЖУРНАЛА ПРИЕМОВ В СТОМАТОЛОГИЧЕСКОЙ ПОЛИКЛИНИКЕ

Статья посвящена созданию функционального приложения для учета приемов пациентов с широким спектром возможностей, включая управление пользователями,

поиск и отображение информации о пациентах и их приемах, а также возможность коммуникации с пациентами через телефонные звонки. В разработке предусмотрена система разграничения прав доступа между пользователями и администраторами. Приложение разработано с использованием серверной части, построенной на технологиях Node.js, Express, Express Validator, Axios, Lodash, а также базы данных MongoDB со специальной ODM-библиотекой Mongoose для взаимодействия с ней.

Современные информационные технологии играют ключевую роль в медицинской индустрии, предоставляя возможности для разработки эффективных инструментов учета и управления медицинскими данными.

Целью проекта является создание удобного и функционального приложения, которое позволит медицинским учреждениям эффективно управлять информацией о пациентах и их приемах. Основные задачи, решаемые приложением, включают авторизацию пользователей и управление их профилями; возможность поиска пациентов по ФИО и отображение их медицинских данных; добавление, обновление и удаление информации о пациентах и их приемах; реализацию функции звонка пациентам по контактному телефону; разграничение прав доступа между пользователями и администраторами, обеспечивая каждому из них определенные полномочия.

Для разработки серверной части приложения использовались современные технологии, такие как Node.js и Express, а также база данных MongoDB [1] с использованием Mongoose для взаимодействия с ней. Кроме того, для обеспечения безопасности и удобства использования были применены различные библиотеки и инструменты, включая Express Validator, Axios, Lodash и другие.

Первоначально была разработана серверная часть приложения с использованием платформы Node.js и фреймворка Express. Node.js, что обеспечило высокую производительность и возможность асинхронного взаимодействия с базой данных. Фреймворк Express позволил удобно организовать маршрутизацию и обработку HTTP запросов.

Для валидации входящих данных и обеспечения безопасности была использована библиотека Express Validator. Она позволила проводить проверку данных, поступающих от пользователей, на соответствие заданным правилам, что снизило вероятность ошибок и улучшило надежность приложения.

В качестве хранилища данных была выбрана MongoDB, NoSQL база данных из-за ее гибкости и простоты масштабирования. С помощью Mongoose [1], библиотеки для взаимодействия с MongoDB из Node.js, была реализована схема данных и организован доступ к базе данных.

Благодаря использованию MongoDB приложение эффективно хранит информацию о пациентах и их приемах, а также оперативно выполняет запросы к базе данных, обеспечивая быструю работу приложения.

Одной из важнейших задач была реализация системы аутентификации и разграничения прав доступа. Для этого были использованы сессии и токены аутентификации. При успешной аутентификации пользователю выдавался токен, который затем использовался для доступа к защищенным ресурсам сервера. Система разграничения прав доступа позволяла каждому пользователю видеть только свои данные и оперировать только своими ресурсами. Администраторам предоставлялись дополнительные права на просмотр и управление всеми данными.

Помимо основных технологий, описанных выше, для разработки были также использованы дополнительные инструменты, такие как Axios для выполнения HTTP запросов, Lodash для упрощения манипуляций с данными и другие.

Для создания интерфейса приложения, предназначенного для учета приемов пациентов и управления медицинскими данными, был выбран фреймворк React Native [2, 3]. Этот выбор обусловлен его возможностью создания кроссплатформенных приложений,

которые могут запускаться как на устройствах под управлением iOS, так и на устройствах под управлением Android, с общим кодом. Интерфейсная часть приложения была разработана с использованием компонентов React Native [4, 5]. Это включает в себя элементы ввода данных, кнопки, списки пациентов и приемов, модальные окна для добавления и редактирования информации и т. д. Компоненты были организованы в соответствии с логической структурой приложения, чтобы обеспечить четкость и удобство навигации пользователя. Для управления навигацией между экранами приложения была использована библиотека React Navigation [4, 5].

При запуске программного приложения “Dental App” открывается экран загрузки и главная страница, на которой представлен отсортированный по ближайшей дате журнал приемов, раздел для поиска, категории и вкладки с выводом приемов, а также присутствует подвал с навигацией перехода по вкладкам (рисунок 1). Главная вкладка является основным меню для выбора дальнейших действий пользователя. При активации ФИО пациента осуществляется переход на вкладку с картой пациента, где содержится актуальная информация о приемах – описан номер зуба (с возможностью посмотреть формулу зубов), который необходимо лечить, диагноз, врач (пользователь), который записал пациента, комментарий о приеме, время приема, дата приема, цена услуги и другая необходимая информация о пациенте. В карте пациента приемы отображаются в определенном порядке: сначала на сегодняшний день, затем, после подведенной черты, идут все остальные приемы в отсортированном порядке. Если прием прошел, то он затемняется и пропадает возможность его редактировать.

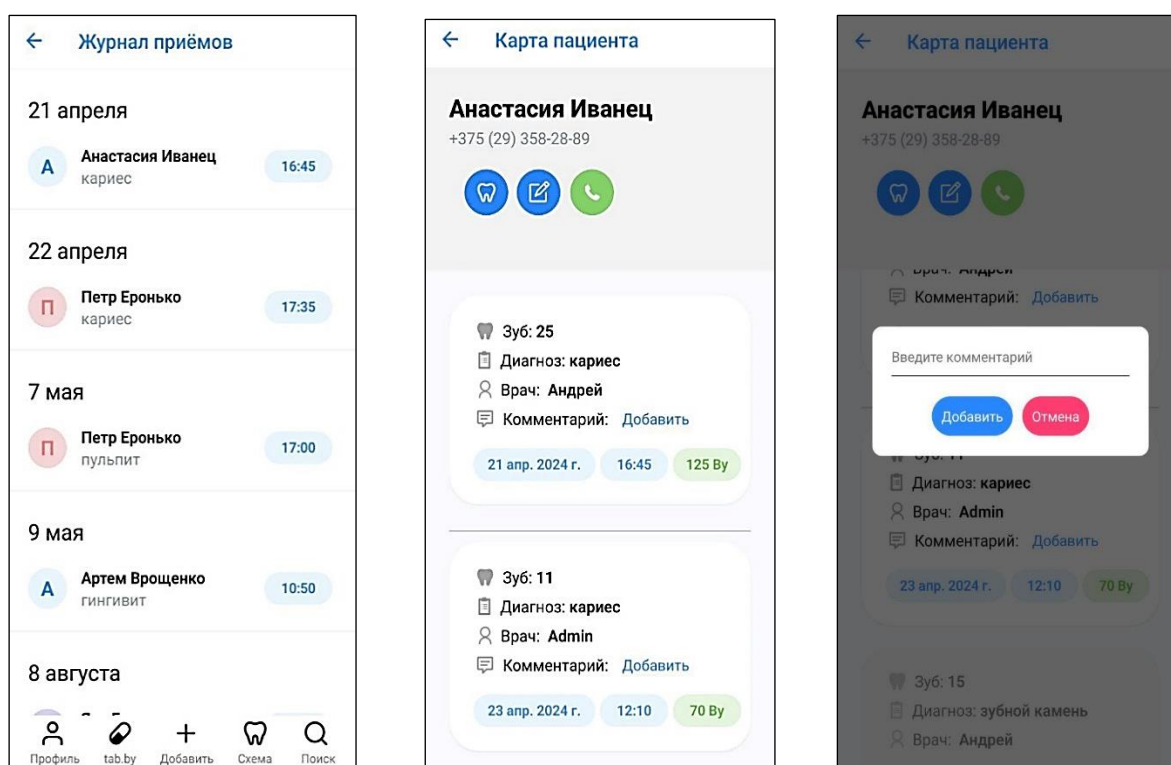


Рисунок 1 – Вкладка журнала приемов (главная вкладка),
вкладка карта пациента и возможность добавления комментария

Система разграничения прав доступа является ключевым элементом приложения, предназначенного для учета приемов пациентов и управления медицинскими данными. Ее целью является обеспечение безопасности данных, конфиденциальности информации и эффективного управления доступом к различным функциям и данным в приложении.

Каждый пользователь должен пройти процесс аутентификации при входе в приложение, предоставляя свои учетные данные (рисунок 2). После успешной аутентификации каждому пользователю назначается соответствующий уровень доступа в соответствии с его ролью в системе. Обеспечение безопасности данных осуществляется путем выдачи токенов аутентификации, которые используются для доступа к защищенным ресурсам сервера.

The figure shows three mobile application screens related to user authentication:

- Вход в аккаунт (Login):** Features a title 'Вход', a subtitle 'Введите данные для входа', input fields for 'Электронная почта' and 'Пароль', a 'Войти' button, and a link 'Нет аккаунта? Зарегистрируйтесь'.
- Регистрация (Registration):** Features a title 'Регистрация', a subtitle 'Введите данные для регистрации', input fields for 'Имя аккаунта', 'Почта', and 'Пароль', a 'Регистрация' button, and a link 'Есть аккаунт? Войдите'.
- Профиль (Profile):** Features a title 'Информация о пользователе', a subtitle 'Данные пользователя', and displays user details: 'Логин: Admin', 'Электронная почта: admin@mail.ru', and 'Роль: ADMIN'. It includes 'Администрирование' and 'Выйти' buttons.

Рисунок 2 – Процесс аутентификации при входе в приложение

Функциональность приложения также зависит от роли пользователя. Обычные пользователи имеют доступ только к функциям, связанным с их собственными данными, такими как просмотр информации о себе и о своих пациентах. Администраторы же имеют право на просмотр и редактирование всех данных в системе, а также на управление учетными записями пользователей. Возможность добавления комментария о приеме предусмотрена только пользователю (врачу), у которого этот прием проходит (рисунок 3).

The figure shows three mobile application screens related to administrative functions:

- Администрирование (Admin Management):** Displays a list of users with their login, email, and role. For each user, there are 'Изменить роль' and 'Удалить пользователя' buttons. The users listed are Андрей (USER), Admin (ADMIN), Евгений (ADMIN), Артем (USER), and Антон.
- Администрирование (Admin Management with modal):** Shows the same list, but with a modal dialog open for 'Изменить роль пользователю: Артем'. The modal allows selecting a role from a dropdown (currently showing 'USER') and includes 'Обновить роль' and 'Отмена' buttons.
- Карта пациента (Patient Card):** Displays patient information for 'Петр Еронько' (+375 (29) 568-36-58). It includes icons for various services and a list of appointments with details like 'Зуб: 46', 'Диагноз: зубной камень', 'Врач: Admin', and 'Комментарий: Добавить'. Each appointment entry also shows the date, time, and duration.

Рисунок 3 – Возможности администратора

Для контроля данных были реализованы вкладки для добавления и обновления пациента, в которых актуализируется имя и фамилия, телефон пациента. Вкладка для добавления, обновления приема включает в себя данные о приеме: номер зуба, диагноз, цену и дату с временем приема. В них реализована система отображения старых данных в поле ввода, для того чтобы не вводить повторно ту информацию, которую пользователь не хочет менять. К приему появляется возможность редактирования комментария (рисунок 4).

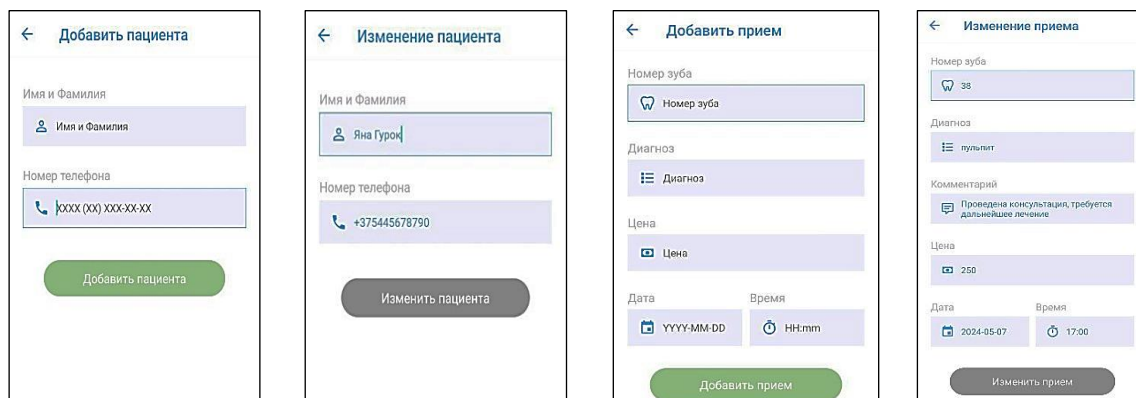


Рисунок 4 – Вкладки добавления и обновления пациента и приема

В приложении реализована возможность поиска лекарственных средств в аптеках Беларуси с помощью сайта <https://tabletka.by>. Целью разработки этой функции являлось удобство для пользователей в случае, если врач выписал какое-либо лечение или лекарственное назначение. Функция предоставляет возможность эффективного поиска нужных препаратов, сравнение их цен, ближайших аптек, что значительно облегчит доступ к необходимым медикаментам.

Литература

- 1 Bush, E. Node.js, MongoDB, React, React Native Full-Stack Fundamentals and Beyond / E. Bush. – Blue Sky Productions Inc., 2018. – 392 p.
- 2 Fullstack React Nativ. The Complete Guide to React Native / D. Abbott [et. al.]. – Published in San Francisco, California by Fullstack.io, 2017. – 690 p.
- 3 Boduch, A. React and React Native / A. Boduch, R. Derks. – Third Edition. – Pact Pabl, 2020. – 526 p.
- 4 Lebensold, J. React Native Cookbook / J. Lebensold. – United States of America, 2018. – 200 p.
- 5 Dabit, N. React Native in Action / N. Dabit. – Manning, 2019. – 320 p.

УДК 519.2

А. А. Зданович

ПОСТРОЕНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ ПРОГНОЗИРОВАНИЯ ЦЕНЫ АКЦИИ SHELL

Статья посвящена разработке нейронной сети для прогнозирования цены акций Shell. Исследован временной ряд цены акции, проведен спектральный анализ Фурье для выявления периодичности, а также построены и обучены пять моделей нейронных сетей. По наибольшей производительности, наименьшей ошибке обучения и контрольной ошибке выбрана более эффективная модель.