

Об опыте разработки мобильного приложения вопросов и ответов

А. В. Данеев¹✉, Р. А. Данеев², И. А. Рыжов¹

¹ Иркутский государственный университет путей сообщения, г. Иркутск, Российская Федерация

² Восточно-Сибирский институт МВД России, г. Иркутск, Российская Федерация

✉ daneev@mail.ru

Резюме

В работе рассматривается опыт разработки мобильного приложения, предназначенного для вопросов и ответов. Актуальность исследования заключается в том, что интернет стал рассматриваться как большая база знаний, для работы с которой требуются специальные инструменты. Система вопросов и ответов представляет собой сообщество из пользователей, помогающих друг другу в поиске решения конкретной проблемы. Такие системы являются базой знаний, которые хранят в себе вопросы и их решение. Плюс таких систем в том, что, если пользователь ограничен в знаниях в какой-либо области, то ему может помочь человек, разбирающийся в этой сфере. В таких системах существует принцип разделения вопросов на категории, например, наука, философия, психология или автомобили. Современные системы вопросов и ответов – это, как правило, веб-сайт, предназначенный для людей, ищущих решение на свой вопрос. Такие сайты позволяют пользователям задать вопрос или помочь с решением на уже заданные вопросы. Таким образом, формируется социальная сеть, участником которой может стать каждый. Такие социальные сети не только помогают с решением проблемы, но и дают пользователям возможность обмениваться своими знаниями и опытом с другими, а в некоторых случаях искать новые пути решения. Разработанное мобильное приложение является таким инструментом. Оно позволяет организовать базу знаний пользователей и предоставляет удобный поиск вопросов и ответов. Также данное приложение организует небольшую социальную сеть с системой сообщений для общения пользователей. Пользователю не нужно тратить время на поиск интересующего его вопроса в интернете. Такое приложение соответствует установленным требованиям, поддерживается большим количеством устройств, работающих на операционной системе Android. Реализованная в мобильном приложении система аутентификации обеспечивает необходимую безопасность доступа к возможностям приложения.

Ключевые слова

мобильное приложение, сеть Интернет, поиск информации, база знаний, социальная сеть

Для цитирования

Данеев А. В. Об опыте разработки мобильного приложения вопросов и ответов / А. В. Данеев, Р. А. Данеев, И. А. Рыжов // Современные технологии. Системный анализ. Моделирование. – 2021. – № 3 (71). – С. 202–211. – DOI: 10.26731/1813-9108.2021.3(71).202-211

Информация о статье

поступила в редакцию: 22.07.2021, поступила после рецензирования: 19.08.2021, принята к публикации: 03.09.2021

On the experience in developing mobile application "questions and answers"

A. V. Daneev¹✉, R. A. Daneev², I. A. Ryzhov¹

¹ Irkutsk State Transport University, Irkutsk, the Russian Federation

² East Siberian Institute of the Ministry of Internal Affairs of Russia, Irkutsk, the Russian Federation

✉ daneev@mail.ru

Abstract

The paper discusses the experience of developing a mobile application of questions and answers. The relevance of this work lies in the fact that the Internet has come to be seen as a large knowledge base, which requires special tools. The system of questions and answers is a community of users helping each other in finding a solution to their problem. Such systems are a knowledge base that stores problems and their solutions. The advantage of such systems is that if the user is limited in knowledge in any area, then a person who understands this area can help him. In such systems, there is a principle of dividing questions into categories, for example, science, philosophy, psychology or cars. Modern question and answer systems are usually a website consisting of people looking for a solution to their questions. Such sites allow users to ask a question or help with the solution to questions already asked. This is how a social network is formed, which everyone can become a member of. Such social networks not only help with solving problems, but also give users the opportunity to share their knowledge and experience with others. And in some cases, look for new solutions. The developed mobile application is such a tool. It allows you to organize the user knowledge base and provides a convenient search for questions and answers. Also, this application organizes a small social network with a message system for users to communicate. Users do not need to waste time searching the Internet for a question of interest. Such an application meets the established requirements and is supported by a large number of devices running on the Android operating system. The authentication system implemented in the mobile application provides the necessary security for access to the application's resources.

Keywords

mobile application, Internet, information search, knowledge base, social network

For citation

Daneev A. V., Daneev R. A., Ryzhov I. A. Ob opyte razrabotki mobil'nogo prilozheniya voprosov i otvetov [On the experience in developing mobile application "questions and answers"]. *Sovremennye tekhnologii. Sistemy analiz. Modelirovanie* [Modern Technologies. System Analysis. Modeling], 2021, No. 3 (71), pp. 202–211. – DOI: 10.26731/1813-9108.2021.3(71).202-211

Article info

Received: 22.07.2021, Revised: 19.08.2021, Accepted: 03.09.2021

Введение

В последние годы наблюдается бурный рост объема общедоступной информации. Вследствие чего в современном обществе появилась проблема поиска и получения нужной информации. Эту проблему усугубляет и то, что в настоящее время информация, доступная в сети Интернет, имеет очень высокий уровень динамики. В каждый момент времени появляются новые материалы и факты. Постоянный рост объема информационных массивов и их обновление делают сложным, а зачастую практически невозможным учет всей информации. По причине этого данные, представляющие ценность, частую остаются невостребованными.

В XXI в. деятельность людей, коллективов, организаций и компаний в большей степени зависит от имеющейся у них информации, а также способности быстро ее найти. Имея доступ к представленной в сети Интернет информации, пользователю хотелось бы получать только нужную ее часть, в то время как поисковые системы представляют для этого малые возможности. Пользователю приходится самому продолжать искать информацию среди той, что ему предлагается, что затрудняет ее восприятие. Таким образом, получается противоречие между большим количеством доступной информации и ограниченными возможностями по ее поиску.

Когда мы хотим что-то узнать, мы спрашиваем – задаем вопрос, что, в общем-то, естественно в процессе познания. Большинство систем по поиску информации не имеют возможности отвечать на наши вопросы. Для поиска и получения человеку нужно сформировать запрос из ключевых слов и задать его поисковой машине.

В последнее время повысился интерес к разработке интеллектуальных и нетрадиционных механизмов поиска и получения информации. Интернет стал рассматриваться как потенциальный большой источник знаний, для работы с которой требуются специальные инструменты. Сегодня термин «информационный поиск» (англ. Information retrieval) включает в себя поиск текстовых документов, изображений, видео, многоязыковой поиск, географически-зависимый поиск. Помимо этого к информационному поиску можно причислить и поиск ответа на вопрос. В последние годы увеличилось количество

проектов таких систем в данной области [1–9], например, сервис вопросов и ответов «Яндекс.Кью».

Описание предметной области

Система вопросов и ответов представляет собой сообщество из пользователей, помогающих друг другу в поиске решения конкретной проблемы. Такие системы являются базой знаний, которые хранят в себе вопросы и их решение. Плюс таких систем в том, что, если пользователь ограничен в знаниях в какой-либо области, то ему может помочь человек, разбирающийся в этой сфере. В таких системах существует принцип разделения вопросов на категории, например, наука, философия, психология или автомобили.

Современные системы вопросов и ответов – это, как правило, веб-сайт, на котором люди ищут решение своих вопросов. Такие сайты позволяют пользователям задать вопрос или помочь с решением на уже заданные вопросы. Таким образом, формируется социальная сеть, участником которой может стать каждый. Такие социальные сети не только помогают с решением проблемы, но и дают пользователям возможность обмениваться своими знаниями и опытом с другими, а в некоторых случаях искать новые пути решения.

Для разработанного мобильного приложения необходимо, чтобы мобильное устройство соответствовало следующим техническим характеристикам:

- ОС Android 5.1 +;
- оперативная память не менее 1 Гб;
- объем встроенной памяти 100 Мб;
- постоянный доступ к Интернету.

Программа поставляется в виде установочного файла формата .apk. В будущем приложение может быть размещено в магазине приложений Play Market. Для разработки приложения была выбрана среда разработки Visual Studio Code – редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS.

Описание взаимодействия пользователя с приложением

Взаимодействие пользователя с приложением происходит в соответствии с рис. 1, 2.

Просмотр информации о приложении. Пользователь, находясь на экране авторизации, может

ознакомиться с информацией о приложении, нажав на соответствующую кнопку «О приложении». Если пользователь авторизован, с данной информацией можно ознакомиться через настройки профиля пользователя.

Просмотр справки «Как взаимодействовать с приложением». Пользователь, находясь на главном экране, может нажать на соответствующую кнопку – «знак вопроса» – для просмотра помощника по взаимодействию с приложением.

Создание поста с вопросом. Пользователь может создать пост с вопросом при нажатии на соответствующую кнопку – «знак плюса» – на нижней панели навигации приложения. На экране создания вопроса необходимо выбрать категорию и заполнить обязательные поля вопроса: тема вопроса, описание проблемы. Также пользователь может, если это необходимо, прикрепить к вопросу фотографию или документ. Когда вопрос сформирован, необходимо нажать на кнопку «Опубликовать». После этого созданный вопрос появится на главном экране приложения, и другие пользователи могут помочь с проблемой.

Редактирование поста с вопросом. Пользователь может отредактировать свой опубликованный вопрос, выбрав из списка своих вопросов необходимый и нажав на соответствующую кнопку – «знак

редактирования». На экране редактирования изменить необходимые элементы поста вопроса и нажать на кнопку «Сохранить».

Поиск поста с вопросом. Пользователь выполняет поиск необходимого вопроса по базе. При нажатии на соответствующую кнопку – «знак лупа» – на панели навигации в поле вводится необходимая информация, далее выводится список совпадений.

Просмотр постов по категориям. Пользователь, находясь на главном экране приложения, нажимает на кнопку «знак списка» со списком категорий и выбирает нужную. После чего знакомится с постами данной категории.

Просмотр поста с вопросом. Пользователь на главном экране выбирает интересующий его вопрос.

Добавление поста с вопросом в избранное. Пользователь, находясь на главном экране, может добавить в избранное пост, нажав на соответствующую кнопку – «знак сердца» – под ним.

Отметить ответ как решение вопроса. Пользователь при просмотре своего поста и ответов к нему, выбирает ответ как решение вопроса, нажав на соответствующую кнопку – «знак большой палец вверх». После чего на посте с вопросом появится индикатор, что вопрос решен. Пользователь может отменить действие, нажав на кнопку еще раз.

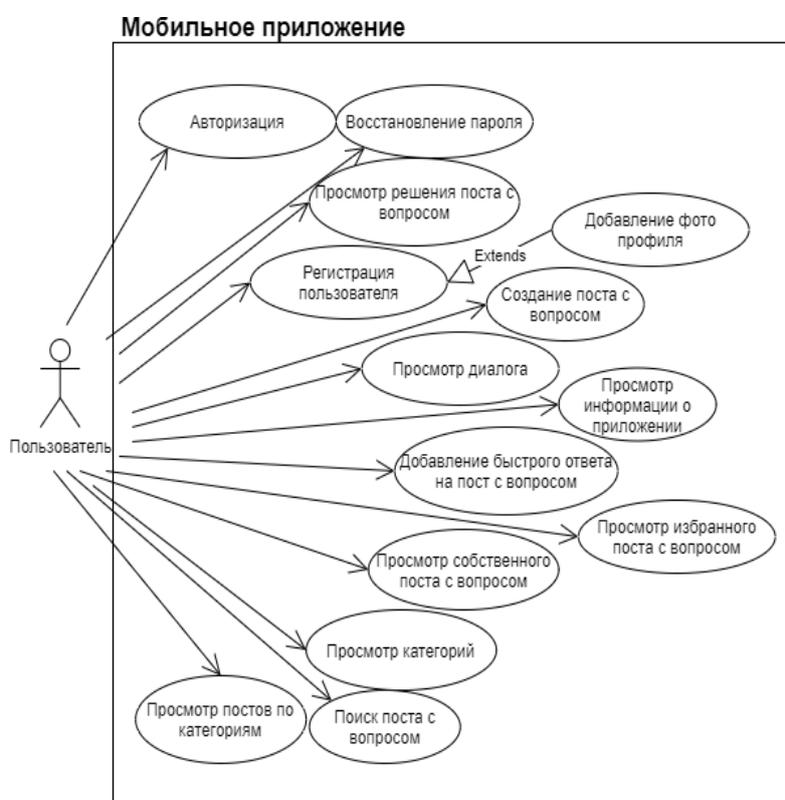


Рис. 1. Диаграмма сценариев (первая часть)

Fig. 1. Scenario diagram (first part)

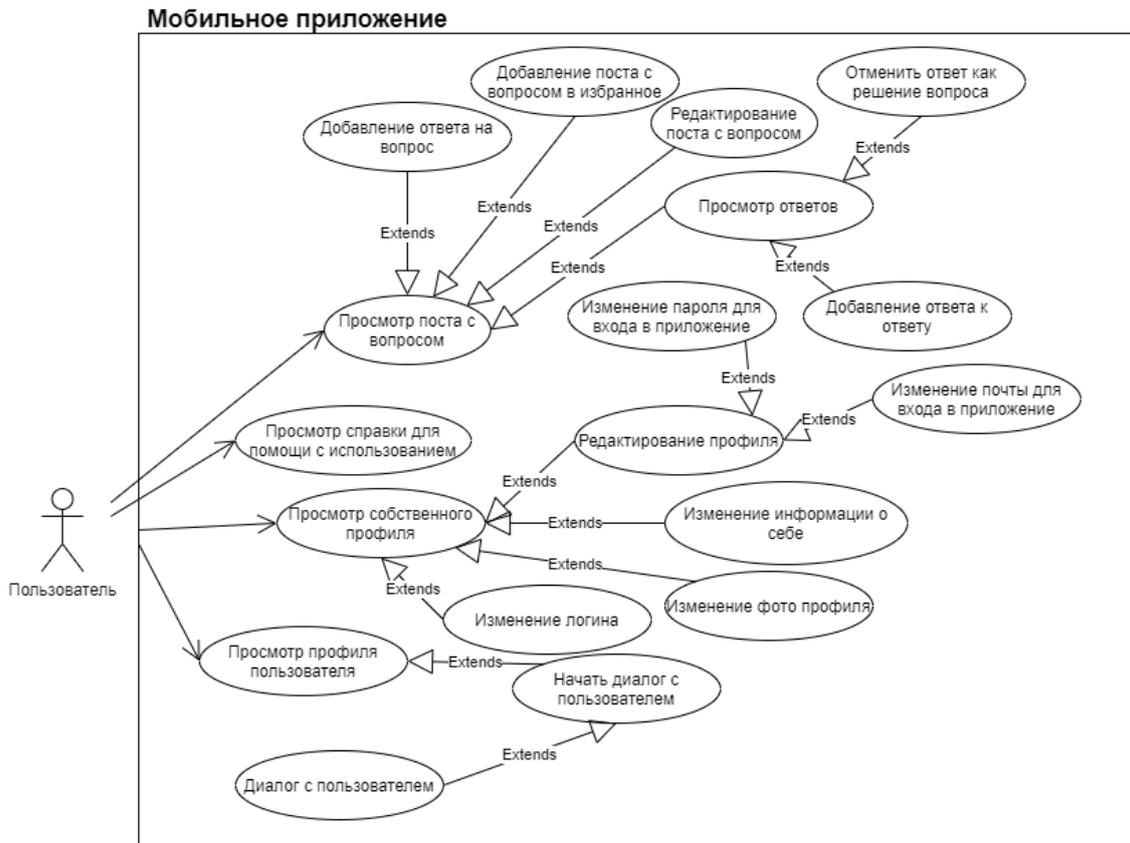


Рис. 2. Диаграмма сценариев (вторая часть)

Fig. 2. Scenario diagram (second part)

Просмотр решения поста с вопросом. Пользователь при просмотре постов может нажать кнопку статуса поста, если она имеется. И на экране будет отображен пост с решением.

Просмотр избранного поста с вопросом. Пользователь может нажать в верхней части главного экрана на кнопку избранных постов – «знак сердца». И на экране со списком избранных постов, выбрать тот, который его интересует.

Просмотр собственного поста с вопросом. Пользователь в своем профиле может нажать на кнопку «собственные посты». И на экране со списком постов, выбрать необходимый для просмотра пост.

Добавление ответа на вопрос. Пользователь при просмотре вопроса может помочь другому пользователю с вопросом при нажатии на соответствующую кнопку – «Ответить». На экране создания ответа необходимо заполнить поле ответа. И при необходимости можно прикрепить фотографию или документ. После формирования ответа необходимо нажать на кнопку «Добавить» и ответ на вопрос появится в комментариях к посту вопроса.

Просмотр ответов. Пользователь при просмотре вопроса знакомится с ответами, нажав на кнопку «Комментарии». Также, в случае необходи-

мости, предусмотрено добавление ответа к ответу, что позволяет пользователям вступать в диалог с другими пользователями.

Добавление ответа к ответу. Пользователь при необходимости может ответить на ответ, нажав на соответствующую кнопку – «Ответить», и предварительно заполнив поле для добавления ответа.

Редактирование профиля. Пользователь, находясь на экране профиля, может отредактировать свой профиль, нажав на соответствующую кнопку – «знак редактирования» рядом с элементом, который он хочет изменить, например, логин, информацию о себе или фото профиля.

Просмотр собственного профиля. Пользователь может открыть свой профиль и ознакомиться со своей информацией при нажатии на соответствующую кнопку на панели навигации.

Просмотр профиля пользователя. Пользователь при просмотре постов, комментариев нажимает на фото пользователя, знакомясь с профилем после перехода на новый экран.

Добавление быстрого ответа на пост с вопросом. Пользователь может добавить быстрый ответ к посту с вопросом, нажав на соответствующую кнопку – «Ответить» – в блоке поста.

Изменение почты для входа в приложение. Пользователь в настройках профиля может изменить почту для входа в приложение. Предварительно подтвердив старую почту и введя новую.

Изменение пароля для входа в приложение. Пользователь в настройках профиля изменяет пароль для входа в приложение. Введя почту профиля и получив письмо со сбросом пароля на почту.

Начать диалог с пользователем. Пользователь при просмотре профиля другого пользователя может начать диалог с ним, нажав на соответствующую кнопку – «Написать сообщение».

Просмотр диалога. Пользователь, перейдя через панель навигации на экран списка диалогов, выбирает необходимый диалог и переходит к просмотру.

Диалог с пользователем. Пользователь при просмотре диалога с другим пользователем может вводить в соответствующее поле текст сообщения и вести диалог с ним.

Авторизация. Пользователю для входа в приложение необходимо ввести почту и пароль.

Восстановление пароля. Если пользователь по какой-либо причине потерял пароль, он восстанавливает его, перейдя на соответствующий экран.

Регистрация пользователя. Если пользователь использует приложение впервые, ему необходимо зарегистрироваться, перейдя на форму регистрации, нажав на соответствующую кнопку. На форме регистрации заполняются обязательные поля: логин, почта, пароль. При желании можно также установить фото профиля.

Архитектура мобильного приложения

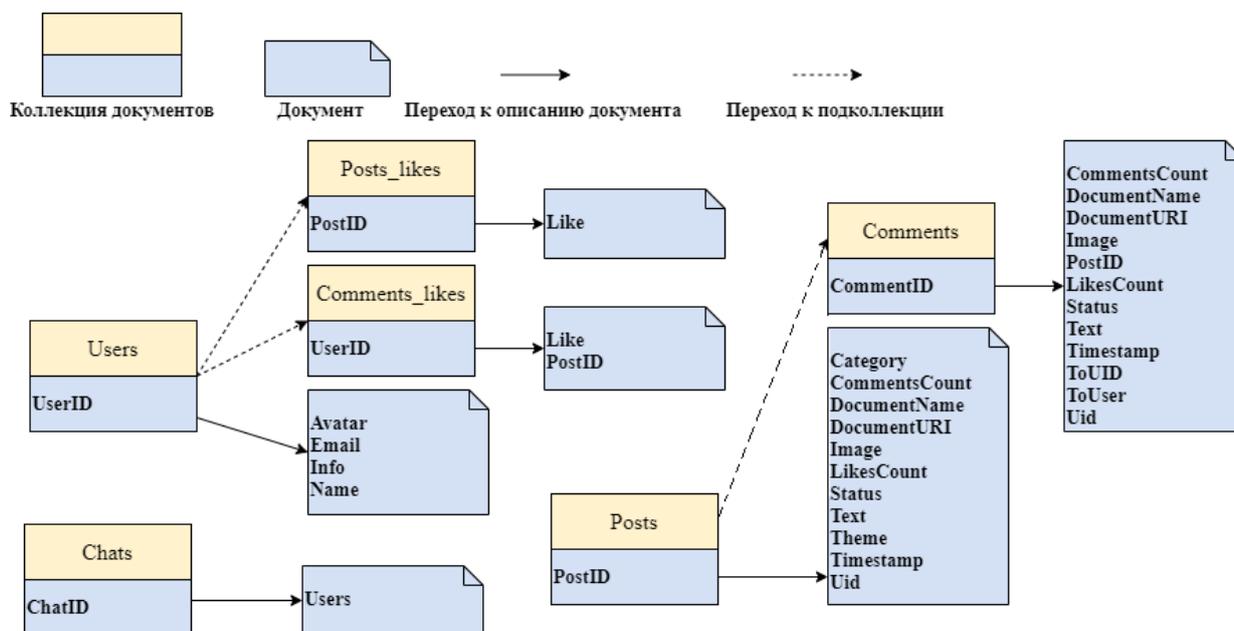


Рис. 3. Схема базы данных Firestore

Fig. 3. Firestore Database Schema

Приложение вопросов и ответов является клиент-серверным приложением. В качестве клиента выступает приложение для Android. Клиент не взаимодействует с базой данных напрямую, так же он не содержит большого количества бизнес-логики. Клиент отвечает за получение и отображение данных, полученных с сервера. Так же клиент отвечает за отправку полученных от пользователя данных, или выполняет запрос к данным по команде пользователя. Взаимодействие с сервером осуществляется через HTTP запросы и ответы. На сервере сосредоточена основная часть бизнес-логики. Он предоставляет клиенту набор методов API, через которые можно передавать и получать данные. Также он осуществляет обработку получаемой информации от клиента и обеспечивает взаимодействие между разными пользователями приложения.

Серверная часть приложения

Для реализации серверной части приложения использовалась облачная платформа Firebase. В разработанном приложении используются следующие компоненты Firebase:

- Firebase Authentication – для аутентификации пользователей в приложении;
- Cloud Firestore – NoSQL база данных – необходима для хранения вопросов, ответов, информации пользователей (логина, персональных данных);
- Firebase Realtime – это облачная база данных (данные хранятся в формате JSON и синхронизируются в режиме реального времени с каждым подключенным клиентом), используется для хранения диалогов пользователей;

– Cloud Storage – облачное хранилище – используется для хранения фотографий, документов и фотографий профилей пользователей.

База данных Cloud Firestore. В схеме базы данных приложения представлены четыре коллекции: category, posts, chats, users, которые содержат документы с информацией о постах с вопросами, их категориях, диалогов и пользователей соответственно (рис. 3).

База данных Realtime Database. В схеме базы данных приложения представлена коллекция chats, которая содержит коллекцию диалогов (рис. 4). Коллекция Chats содержит документы диалогов, в которых хранятся документы сообщений. Внутри каждого документа сообщения хранится объект User с информацией о пользователе-отправителе.

Хранилище файлов Cloud Storage. В схеме хранилища файлов приложения представлены четыре папки для хранения: avatars, comments, documents, photos, которые содержат документы и фотографии, публикуемые пользователями приложения (рис. 5).

Реализация мобильного приложения

Подключение к серверу Firebase. Для начала работы с аутентификацией, базами данных, облач-

ным хранилищем и облачными функциями необходимо установить соединение с сервером Firebase. В разрабатываемом приложении с помощью пакетного менеджера Node.js нужно установить Firebase JavaScript SDK через следующую команду: `npm i firebase`. После чего импортировать его в проект через кодовую строку: `require('firebase')`.

В веб-интерфейсе Firebase необходимо создать проект разрабатываемого приложения. После чего в созданном проекте получить объект конфигурации для создания соединения с Firebase (рис. 6). Данный объект конфигурации внедряется программным кодом мобильного приложения (рис. 7).

Аутентификация. В приложении используется аутентификация по электронной почте и паролю. Для включения функции аутентификации по почте необходимо активировать эту опцию в веб-сервисе Firebase. Когда пользователь входит в приложение или регистрируется, он становится текущим пользователем экземпляра Auth. Экземпляр сохраняет состояние пользователя, поэтому при перезапуске приложения информация пользователя не теряется. Когда пользователь выходит из приложения, экземпляр Auth перестает хранить ссылку на пользовательский объект и больше не сохраняет его состояние.

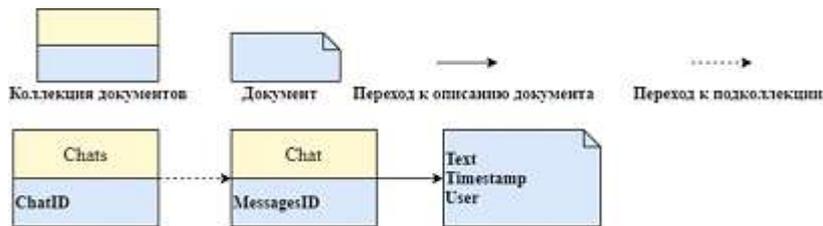


Рис. 4. Схема базы данных Realtime Database

Fig. 4. Real time Database Schema

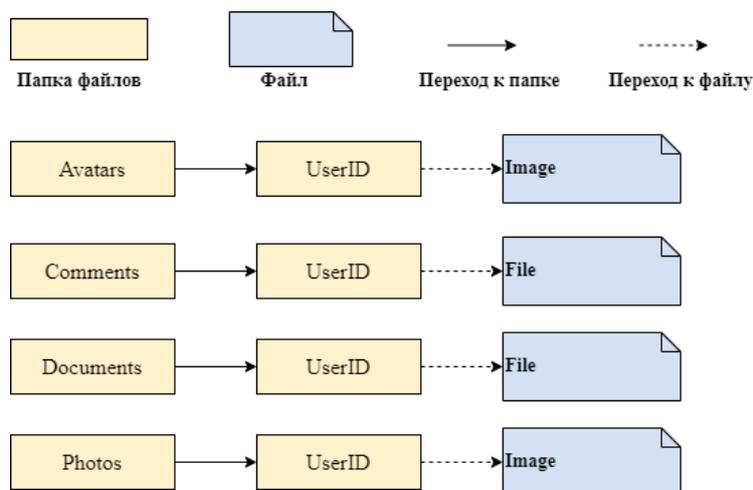


Рис. 5. Схема хранилища файлов

Fig. 5. File storage scheme

```

var firebaseConfig = {
  apiKey : "AIzaSyDOCAbC123dEf456GhI789jKl01-Mn0" ,
  authDomain : "myapp-project-123.firebaseio.com" ,
  databaseURL : "https://myapp-project-123.firebaseio.com" ,
  projectId : "myapp-project-123" ,
  storageBucket : "myapp-project-123.appspot.com" ,
  messagingSenderId : "65211879809" ,
  appId : "1: 65211879909: веб: 3ae38ef1cdcb2e01fe5f0c" ,
  Идентификатор измерения : "G-8GSGZQ44ST"
};

```

Рис. 6. Объект конфигурации Firebase

Fig. 6. Firebase Configuration Object

```

if (!firebase.apps.length) {
  firebase.initializeApp(FirebaseKeys);
}

```

Рис. 7. Программный код инициализации подключения к серверу Firebase

Fig. 7. Program code for initializing the connection to the Firebase server

Авторизация. Для авторизации в приложении используется метод `firebase.auth().signInWithEmailAndPassword` (`email`, `password`). При нажатии на кнопку «Войти» вызывается метод, который отправляет введенные данные: почту и пароль. Firebase Authentication проверяет правильность полученных данных и возвращает ответ, токен доступа или ошибка входа.

Работа с файлами через мобильное приложение. Для возможности добавить картинку используется компонент `ImagePicker`, который импортирует через программный код: `import * as ImagePicker from 'expo-image-picker'`. При нажатии на иконку добавления фотографии вызывается функция `pickImage`. Данный компонент имеет настройки типа медиа файлов, возможность редактирования, соотношение сторон и другие. `ImagePicker`, предоставляет пользователю выбрать, через какое приложение добавить фотографию.

Для добавления файла используется компонент `DocumentPicker`, который импортирует через программный код: `import * as DocumentPicker from 'expo-document-picker'`. При нажатии на иконку добавления документа вызывается функция `pickDocument`. `DocumentPicker` предоставляет пользователю выбрать необходимый файл из памяти телефона. После выбора файла компонент `DocumentPicker` возвращает объект с информацией о файле, такие параметры как имя и путь записываются в состояние компонента приложения для загрузки в облачное хранилище.

При нажатии на кнопку «Написать сообщение» на сервер отправляется запрос вида: `collection('chats').where('users', 'array-contains', userID).get()`.

Данный запрос отправляется в базу данных Cloud Firestore в коллекцию чатов для проверки существования диалогов пользователя. Если диалоги существуют, то он вернет список из уникальных идентификаторов диалогов. Запрос к Cloud Firebase не позволяет делать сложную выборку из коллекции, поэтому список диалогов обрабатывается на стороне клиента. Список фильтруется на соответствие состава из двух конкретных пользователей диалога. Если он существует, то список сообщений загрузится из базы данных. Для создания диалога нужно чтобы один из пользователей отправил сообщение собеседнику. После отправки сообщения будет сгенерирован уникальный документ диалога с массивом пользователей: `collection('chats').doc().set({users: [userID, user Companion]})`.

Для реализации поиска постов с вопросами в приложении используются Cloud Functions – облачные функции Firebase и интеграция в приложение поисковой системы Algolia.

Для подключения необходимых пакетов используется пакетный менеджер Node.js. Для реализации поиска потребуется скачать пакет для взаимодействия с Cloud Functions с помощью команды: `npm firebase-functions`. Для интеграции поисковой системы Algolia также потребуется пакет, который скачивается с помощью `npm algoliasearch`.

За отображение списка постов отвечает компонент `Flatlist`. Этот компонент имеет множество свойств, такие как `data`, `initialNumToRender`, `renderItem`, `onRefresh`. `Data` – данные для отрисовки, список элементов. `InitialNumToRender` – список отрисовывается после получения определенного количества элементов, используется для оптимизации,

чтобы не дожидаться полного получения списка элементов. `RenderItem` – задается разметка для рендеринга элемента списка. Для обновления списка используется жест «потягивание списка вниз», который запускает функцию обновления списка, указанную в свойстве `OnRefresh`.

Описание работы интерфейса мобильного приложения

Регистрация, восстановление доступа и аутентификация пользователей. При первом запуске приложения пользователя встречает привет-

ственный экран с формой входа (рис. 8). Если у пользователя уже существует аккаунт, и он авторизован, то он увидит экран главной страницы приложения без повторной авторизации. Для новых пользователей существует регистрация (рис. 8). При регистрации пользователь может установить фото профиля. Для восстановления доступа к приложению нужно заполнить поле привязанной почты и подтвердить действие. После чего на почту придёт письмо со ссылкой сброса пароля. Если при входе, восстановлении доступа или регистрации возникнут проблемы, приложение выдаст ошибку.

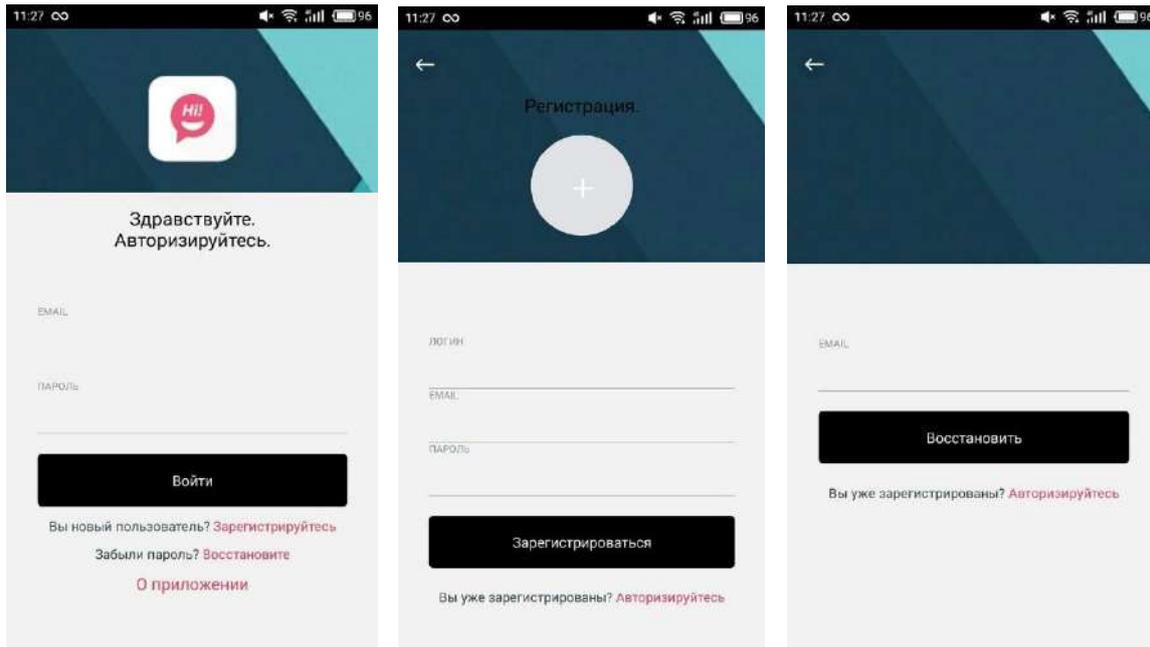


Рис. 8. Начальные экраны приложения

Fig. 8. Application Start Screens

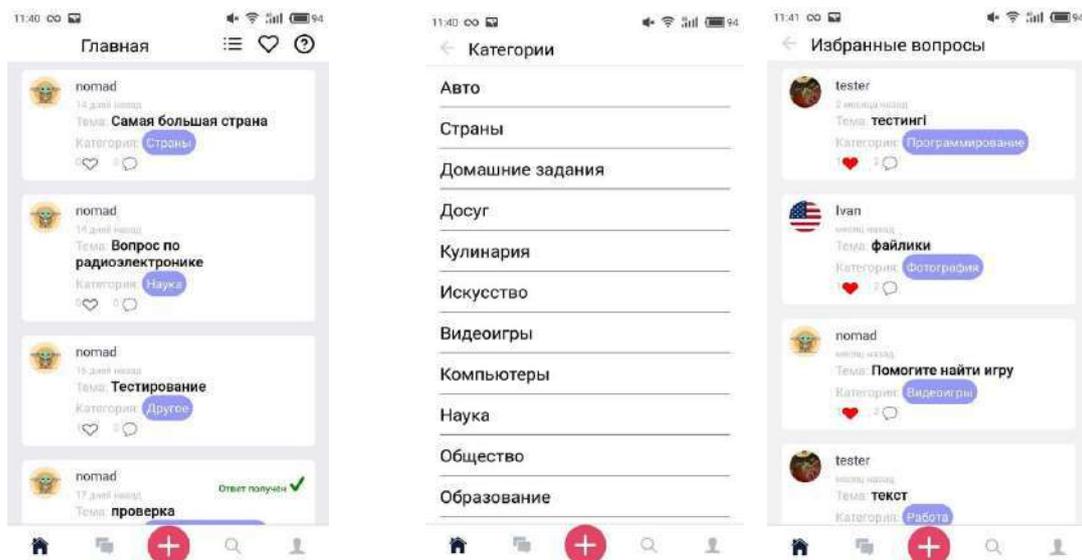


Рис. 9. Главный экран приложения и его работа

Fig. 9. The main screen of the application and its operation

Главный экран приложения. После успешного входа в приложение пользователя приветствует главный экран приложения (рис. 9). В верхней части приложения (заголовок) слева направо располагаются название экрана приложения, кнопка просмотра категорий, кнопка просмотра постов с вопросами добавленных в избранное, кнопка просмотра краткой информации как взаимодействовать с приложением. Для добавления поста с вопросом в избранное нужно просто нажать на кнопку в виде «сердца», после добавления «сердце» поменяется свой цвет на красный. Для удаления достаточно проделать те же самые действия. Посты с вопросами имеют счетчики с количеством добавлений в избранное и количеством ответов. Для добавления ответа достаточно нажать на соседнюю кнопку – «комментарий» – рядом с «сердцем».

Заключение

В статье дано описание мобильного приложения для создания и поиска вопросов на различные темы и получения ответов для операционной системы Android. Приложение поддерживается большинством устройств, является актуальным и полезным в повседневных задачах пользователей. Реализованная в мобильном приложении система аутентификации обеспечивает необходимую безопасность доступа к возможностям приложения. В работах [9–18] приведены авторские свидетельства, учебные пособия и статьи, в которых рассматриваются смежные вопросы.

Список литературы

1. Хэррон Д. Разработка серверных веб-приложений на Javascript / Хэррон Дэвид. ДМК Пресс, 2012. 144 с.
2. Томас М.Т. React в действии / М.Т. Томас, Питер, 2019. 368 с.
3. React Native Documentation // React Native. URL: <https://reactnative.dev/docs/getting-started> (дата обращения 02.05.2020).
4. Flux: Архитектура приложения на React.js – всестороннее исследование // Medium. URL: <https://medium.com/@marina.kovalyova/flux-the-react-js-application-architecture-773f515d068d> (дата обращения 02.05.2020).
5. Firebase Authentication Documentation // Firebase. URL: <https://firebase.google.com/docs/auth> (дата обращения 02.05.2020).
6. Firebase Realtime Database Documentation // Firebase. URL: <https://firebase.google.com/docs/database> (дата обращения 02.05.2020).
7. Firebase Cloud Firestore Documentation // Firebase. URL: <https://firebase.google.com/docs/firestore> (дата обращения 02.05.2020).
8. Firebase Storage Documentation // Firebase. URL: <https://firebase.google.com/docs/storage> (дата обращения 02.05.2020).
9. Firebase Cloud Functions Documentation // Firebase. URL: <https://firebase.google.com/docs/functions> (дата обращения 02.05.2020).
10. Algolia Search Documentation // Algolia. URL: <https://www.algolia.com/doc/guides/building-search-ui/installation/react/> (дата обращения 02.05.2020).
11. Node.js Documentation // Node.js. URL: <https://nodejs.org/en/docs/> (дата обращения 10.05.2020).
12. NPM Documentation // npmjs. URL: <https://docs.npmjs.com/> (дата обращения 10.05.2020).
13. Паттерны разработки: MVC vs MVP vs MVVM // Habr. URL: <https://habr.com/ru/post/344184/> (дата обращения 10.05.2020).
14. Мобильное приложение для Multi-Vendor: Вопросы и ответы. URL: https://www.cs-cart.ru/docs/latest/user_guide/mobile_app/faq.html (дата обращения 10.05.2020).
15. Данеев А.В., Данеев Р. А., Диомидов И.А. Разработка геоинформационной системы в сфере лесоустройства // Информационные технологии и математическое моделирование в управлении сложными системами. 2020. № 3 (8). С. 89–98.
16. А.С. № 2020662976 от 21.10.2020 Рос. Федерация. Мобильное приложение вопросов и ответов / Рыжов И.А., Данеев А.В., Данеев Р.А.
17. А.С. № 2020663581 от 29.10.2020 Рос. Федерация. Геоинформационная система для предприятия в сфере лесоустройства «ГИСЛЕС» / Диомидов И.А., Данеев А.В., Данеев Р.А.
18. Данеев А.В., Ермакова Н.М. Программа тестирования и обучения // Информационные технологии и проблемы математического моделирования сложных систем. Вып. 19. Иркутск: ИрГУПС, 2017. С. 142–146.

References

1. Harron D. Development of server-side web applications in Javascript. DMK Press, 2012. 144 p.
2. Thomas M.T. React in action. Peter, 2019. 368 p.
3. React Native Documentation. React Native. URL: <https://reactnative.dev/docs/getting-started> (accessed 05/02/2020).
4. Flux: React.js Application Architecture – A Comprehensive Study // Medium. URL: <https://medium.com/@marina.kovalyova/flux-the-react-js-application-architecture-773f515d068d> (date circulation 05/02/2020).
5. Firebase Authentication Documentation. Firebase. URL: <https://firebase.google.com/docs/auth> (date of treatment 05/02/2020).

6. Firebase Realtime Database Documentation. Firebase. URL: <https://firebase.google.com/docs/database> (accessed 02.05.2020).
7. Firebase Cloud Firestore Documentation. Firebase. URL: <https://firebase.google.com/docs/firestore> (accessed 02.05.2020).
8. Firebase Storage Documentations. Firebase. URL: <https://firebase.google.com/docs/storage> (accessed 02.05.2020).
9. Firebase Cloud Functions Documentation. Firebase. URL: <https://firebase.google.com/docs/functions> (accessed 02.05.2020).
10. Algolia Search Documentation. Algolia. URL: <https://www.algolia.com/doc/guides/building-search-ui/installation/react/> (date of access 05/02/2020).
11. Node.js Documentation. Node.js. URL: <https://nodejs.org/en/docs/> (date of access 05/10/2020).
12. NPM Documentation. npmjs. URL: <https://docs.npmjs.com/> (accessed 10.05.2020).
13. Development patterns: MVC vs MVP vs MVVM // Habr. URL: <https://habr.com/ru/post/344184/> (accessed 10.05.2020).
14. Mobile application for Multi-Vendor: Questions and Answers. URL: https://www.cs-cart.ru/docs/latest/user_guide/mobile_app/faq.html (accessed 10.05.2020).
15. Daneev A.V., Daneev R.A., Diomidov I.A. Razrabotka geoinformatsionnoj sistemy v sfere lesoustrojstva [Development of a geoinformation system in the field of forest management]. *Informatsionnye tekhnologii i matematicheskoe modelirovanie v upravlenii slozhnymi sistemami* [Information technologies and mathematical modeling in the management of complex systems], 2020, No. 8, pp. 89–98.
16. A.S. No. 2020662976 of 21.10.2020 Ros. Federation. Mobil'noe prilozhenie voprosov i otvetov [Mobile application of questions and answers]. Ryzhov I.A., Daneev A.V., Daneev R.A.
17. A.S. No. 2020663581 of 29.10.2020 Ros. Federation. Geoinformatsionnaya sistema dlya predpriyatiya v sfere lesoustrojstva «GISLES» [Geoinformation system for the enterprise in the field of forest management "GISLES"]. Diomidov I. A., Daneev A.V., Daneev R. A.
18. Daneev A.V., Ermakova N. M. Programma testirovaniya i obucheniya [Program of testing and training]. *Informatsionnye tekhnologii i problemy matematicheskogo modelirovaniya slozhnykh sistem* [Information technologies and problems of mathematical modeling of complex systems], 2017, Vol. 19, pp. 142–146.

Информация об авторах

Данеев Алексей Васильевич – д-р техн. наук, проф. кафедры информационных систем и защиты информации, Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: daneev@mail.ru

Данеев Роман Алексеевич – канд. техн. наук, доц. кафедры информационно-правовых дисциплин, Восточно-Сибирский институт МВД России, г. Иркутск, e-mail: romasun@mail.ru

Рыжов Иван Алексеевич – Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: nomad38note@gmail.com

Information about the authors

Alexey V. Daneev – Doctor of Engineering Science, Professor, the Subdepartment Information systems and information security, Irkutsk State Transport University, Irkutsk, e-mail: daneev@mail.ru

Roman A. Daneev – Ph. D. in Engineering Science, Associate Professor, the Subdepartment Subdepartment Information-Legal disciplines, East Siberian Institute of the Ministry of Internal Affairs of Russia, Irkutsk, e-mail: romasun@mail.ru

Ivan A. Ryzhov – the Subdepartment Information systems and information security, Irkutsk State Transport University, Irkutsk, e-mail: nomad38note@gmail.com