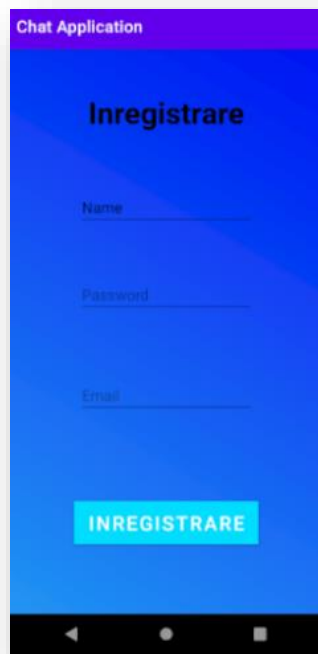


Chat Application – Functionalitati

Înregistrare utilizator:

Interfata utilizatorului pentru pagina de start/pagina de inregistrare:



```
package com.example.chatapplication;  
import ...
```

```
public class RegisterActivity extends AppCompatActivity {  
  
    EditText userDate, passDate, emailDate;  
    Button registerBut;  
    // Firebase  
    FirebaseAuth aut;  
    DatabaseReference ref;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_register);  
  
        // initialize  
        userDate = findViewById(R.id.userEditText);  
        passDate = findViewById(R.id.editTextTextPassword);  
    }  
}
```

```

emailDate = findViewById(R.id.emailEditText);
registerBut = findViewById(R.id.button);
aut = FirebaseAuth.getInstance();

// adding event to button register
registerBut.setOnClickListener(new View.OnClickListener(){

    @Override
    public void onClick(View v) {
        String username_text = userDate.getText().toString();
        String email_text = emailDate.getText().toString();
        String pass_text = passDate.getText().toString();

        // empty data register
        if (TextUtils.isEmpty(username_text) || TextUtils.isEmpty(email_text) ||
            TextUtils.isEmpty(pass_text)) {
            Toast.makeText(RegisterActivity.this, "Fill all fields",
                Toast.LENGTH_SHORT).show();
        } else {
            RegisterNow(username_text, email_text, pass_text);
        }
    }
});

private void RegisterNow(final String name, String email, String password) {
    aut.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {

                if (task.isSuccessful()) {
                    FirebaseUser firebaseUser = aut.getCurrentUser();
                    String userid = firebaseUser.getId();
                    ref = FirebaseDatabase.getInstance().getReference("MyUsers").child(userid);
                    //HashMaps
                    HashMap<String, String> hashMap = new HashMap<>();
                    hashMap.put("id", userid);
                    hashMap.put("username", name);
                    hashMap.put("imageUrl", "default");
                }
            }
        });
}



```

















```
// Opening the Main Activity after Success Regist.
        ref.setValue(hashMap).addOnCompleteListener(new OnCompleteListener<Void>())
    {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if (task.isSuccessful()) {
                Intent i = new Intent(RegisterActivity.this, MainActivity.class);
                i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(i);
                finish();
            }
        }
    });
} else {
    Toast.makeText(RegisterActivity.this, "Invalid Username or Password or
Email",
        Toast.LENGTH_SHORT).show();
}
}
});
}
```

Baza de date Firebase:

ChatApp ▾ Authentication

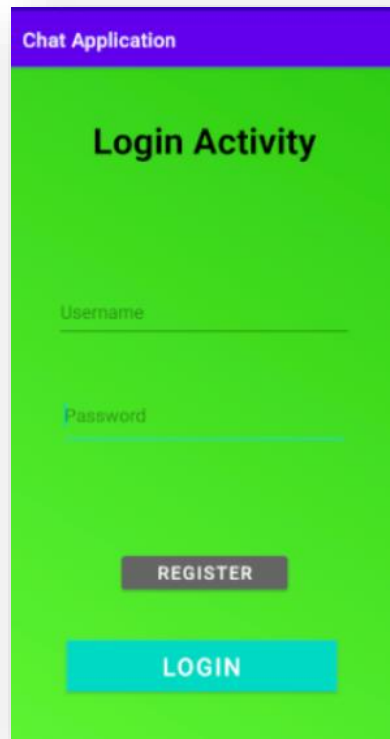
Go to docs

<div><div> Search by email address, phone number, or user UID</div><div><div>Add user</div><div> </div></div></div>				
Identifier	Providers	Created	Signed In	User UID 
usemou@email.com		May 24, 2021	May 24, 2021	FFSFOAq1zih9RQIAxg8JlwxJdZs2
danasimionesel@yahoo.co...		May 24, 2021	May 24, 2021	HNWL7N6qYAbnXU9ADK9irSFEL...
user1@email.com		May 24, 2021	May 24, 2021	KJ83mi73L0arbabminup945T6zd2
usern@email.com		May 24, 2021	May 24, 2021	aKtNXtn5xdf.JAskPPc2o02WMP8g1
user6@email.com		May 10, 2021	May 24, 2021	bvVp6SS6tHQ0zVRIYcJn0VNec873
danasimionesel3@gmail.c...		May 24, 2021	May 24, 2021	eJm3XfCroCMj73oKerQaAyFaWUn2
user@email.com		May 24, 2021	May 24, 2021	gKEp6AA6oI0ccGt90vdxYKR80Gf2
user9@email.com		May 22, 2021	May 24, 2021	mslI98oD9k4XsTvClrk3hRnkUjvS2
catalin.martin100@yahoo.c...		May 24, 2021	May 24, 2021	p1vOxdNOLnTMLYSqclIAIjIMM6Sb2
user5@email.com		May 10, 2021	May 22, 2021	q8EQ45cZRZeJ1QqUNelXeonND5...
newuser@email.com		May 11, 2021	May 11, 2021	t7ioC0Ap0FN0pgkHNkKSt2hUMSiz
asimioneseidaniel852@gm...		May 24, 2021	May 24, 2021	ySfQNLBtpvQJJsARISAKco0fsK3003

Funcția Login/Logout

Interfața utilizatorului pentru pagina de autentificare:



```
package com.example.chatapplication;

import ...
public class LoginActivity extends AppCompatActivity {

    EditText userDateLogin, passDateLogin;
    Button loginBut, registerBut;

    //Firebase
    FirebaseAuth aut;
    FirebaseUser firebase_user_name;

    @Override
    protected void onStart() {
        super.onStart();
        firebase_user_name = FirebaseAuth.getInstance().getCurrentUser();

        //check user exist + save
        if (firebase_user_name != null) {
            Intent i = new Intent(LoginActivity.this, MainActivity.class);
            startActivity(i);
            finish();
        }
    }
}
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    userDateLogin = findViewById(R.id.editTextLogin);
    passDateLogin = findViewById(R.id.editTextPassword2);
    loginBut = findViewById(R.id.button2);
    registerBut = findViewById(R.id.registerBut);

    //autenticare firebase
    aut = FirebaseAuth.getInstance();

    // register button
    registerBut.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(LoginActivity.this, RegisterActivity.class);
            startActivity(i);
        }
    });

    //login button
    loginBut.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String email_text = userDateLogin.getText().toString();
            String pass_text = passDateLogin.getText().toString();

            //empty field
            if (TextUtils.isEmpty(email_text) || TextUtils.isEmpty(pass_text)) {
                Toast.makeText(LoginActivity.this, "Please fill all the fields",
                    Toast.LENGTH_SHORT).show();
            } else {
                aut.signInWithEmailAndPassword(email_text, pass_text).addOnCompleteListener(new
                OnCompleteListener<AuthResult>() {

```

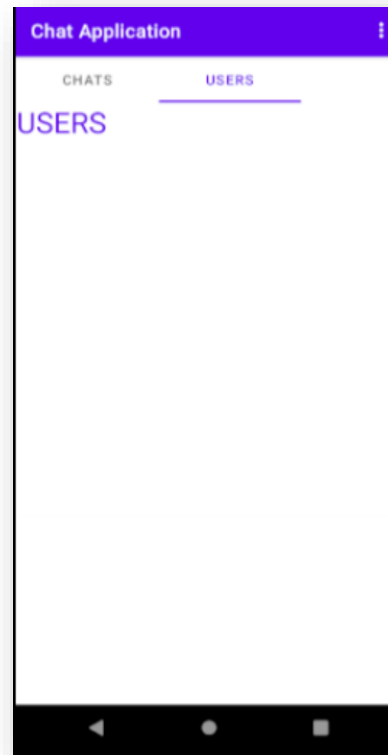
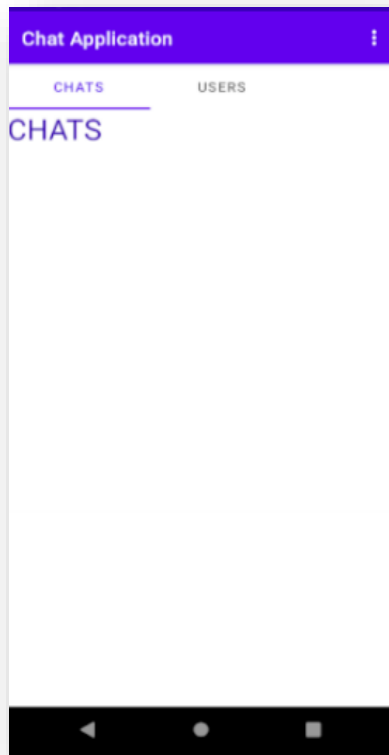
```

                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            Intent i = new Intent(LoginActivity.this, MainActivity.class);
                            i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
                                Intent.FLAG_ACTIVITY_NEW_TASK);
                            startActivity(i);
                            finish();
                        } else {
                            Toast.makeText(LoginActivity.this, "Login Failed!",
                                Toast.LENGTH_SHORT).show();
                        }
                    }
                });
            }
        }
    });
}
}
}
}
}

```

Funcția Meniu

Aplicatia are in aceasta faza doua meniuri: Chats (meniul conversatiilor) si Users (meniul utilizatorilor existenti in aplicatie):



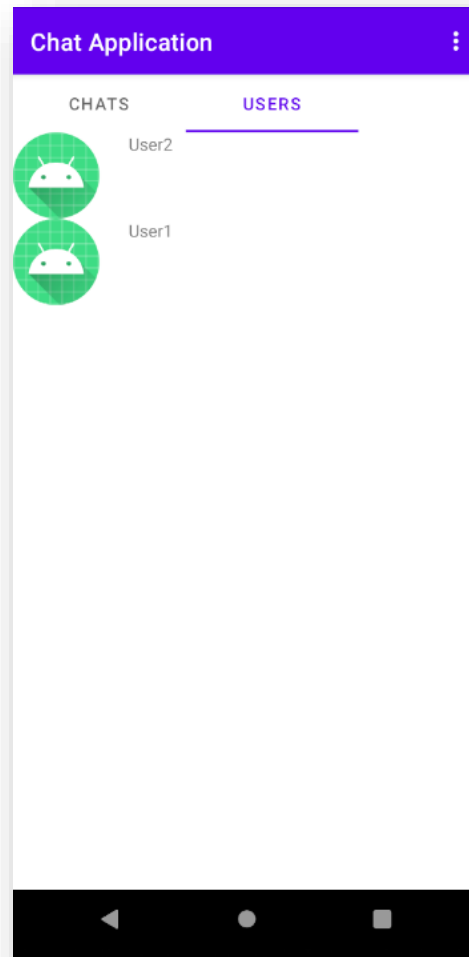
```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Fragments.ChatsFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="CHATS"
        android:textColor="@color/purple_700"
        android:textSize="30sp" />

</FrameLayout>
```

Afisare utilizatori:

In meniul 'Users' al aplicatiei sunt afisati toti utilizatorii inregistrati in aplicatie.



```
package com.example.chatapplication.Adapter;  
  
import ...  
  
public class UserAdapter extends RecyclerView.Adapter<UserAdapter.ViewHolder> {  
    private Context context;  
    private List<Users> mUsers;
```

```

// Constructor

public UserAdapter(Context context, List<Users> mUsers)
{
    this.context = context;
    this.mUsers = mUsers;
}

@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(context).inflate(R.layout.user_item,
        parent,
        false);
    return new UserAdapter.ViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    Users users = mUsers.get(position);
    holder.username.setText(users.getUsername());

    if (users.getImageURL().equals("default")){
        holder.imageView.setImageResource(R.mipmap.ic_launcher);
    }else{
        Glide.with(context)
            .load(users.getImageURL())
            .into(holder.imageView);
    }

    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(context, MessageActivity.class);
            i.putExtra("userid", users.getId());
            context.startActivity(i);
        }
    });
}

@Override
public int getItemCount() {
    return mUsers.size();
}

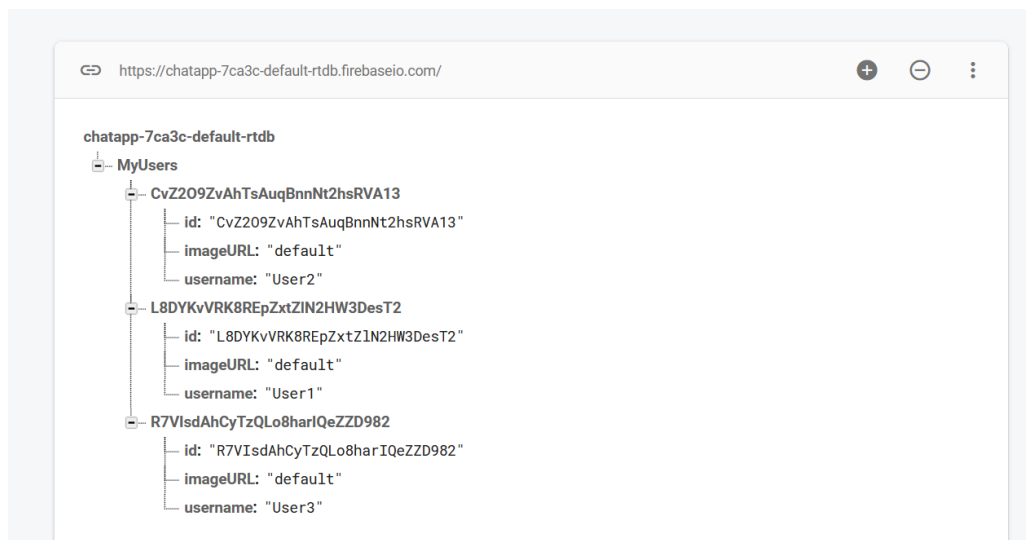
public class ViewHolder extends RecyclerView.ViewHolder{
    public TextView username;
    public ImageView imageView;

    public ViewHolder(@NonNull View itemView) {
        super(itemView);

        username = itemView.findViewById(R.id.textView3);
        imageView = itemView.findViewById(R.id.imageView);
    }
}
}

```

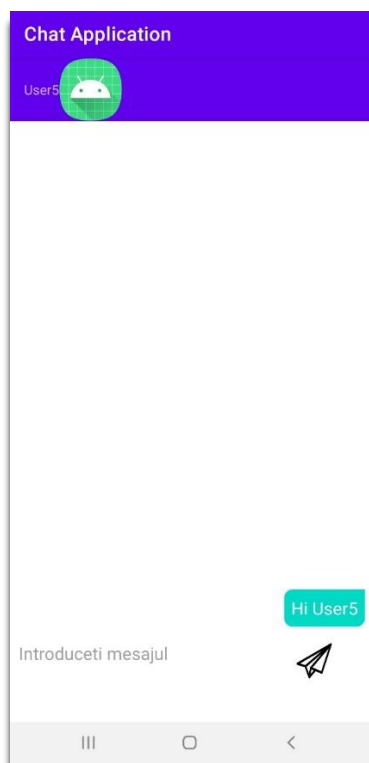

Utilizatorii sunt pastrati in baza de date Firebase in timp real astfel:



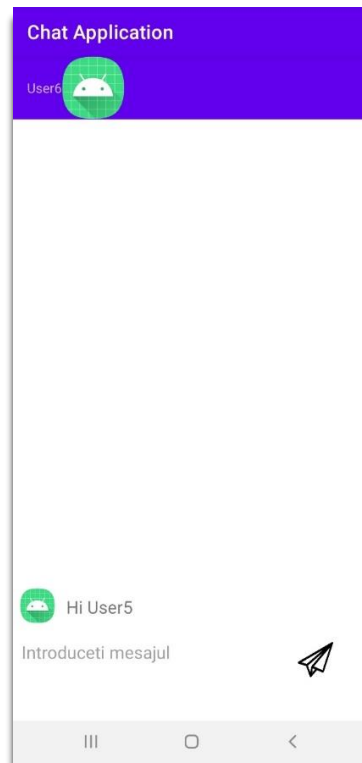
Trimitere mesaje:

S-a implementat functionalitatea de trimitere mesaje.

Interfata expeditorului



Interfata destinatarului



```

package com.example.chatapplication;

import ...

public class MessageActivity extends AppCompatActivity {

    TextView username;
    ImageView imageView;

    RecyclerView recyclerView;
    EditText msg_editText;
    ImageButton sendBtn;

    FirebaseUser fuser;
    DatabaseReference reference;
    Intent intent;

    MessageAdapter messageAdapter;
    List<Chat> mchat;

    RecyclerView recyclerView;
    String userid;

    @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_message);

        // Widgets
        imageView = findViewById(R.id.imageview_profile);
        username = findViewById(R.id.username);
        sendBtn = findViewById(R.id.btn_send);
        msg_editText = findViewById(R.id.text_send);

        //RecyclerView
        recyclerView = findViewById(R.id.recycler_view);
        recyclerView.setHasFixedSize(true);

        LinearLayoutManager linearLayoutManager = new LinearLayoutManager(getApplicationContext());
        linearLayoutManager.setStackFromEnd(true);
        recyclerView.setLayoutManager(linearLayoutManager);

        intent = getIntent();
        userid = intent.getStringExtra("userid");

        fuser = FirebaseAuth.getInstance().getCurrentUser();
        reference = FirebaseDatabase.getInstance().getReference("MyUsers").child(userid);

        reference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                Users user = snapshot.getValue(Users.class);
                username.setText(user.getUsername());

                if(user.getImageURL().equals("default")){
                    imageView.setImageResource(R.mipmap.ic_launcher);
                }else{
                    Glide.with(MessageActivity.this)
                        .load(user.getImageURL())
                        .into(imageView);
                }

                readMessages(fuser.getUid(),userid,user.getImageURL());
            }
        })
    }

```

```

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    });

    sendBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String msg = msg_editText.getText().toString();
            if (!msg.equals("")){
                sendMessage(fuser.getId(), userid, msg);
            }else{
                Toast.makeText(MessageActivity.this, "Nu puteti trimite un mesaj gol",
                    Toast.LENGTH_SHORT).show();
            }

            msg_editText.setText("");
        }
    });
}

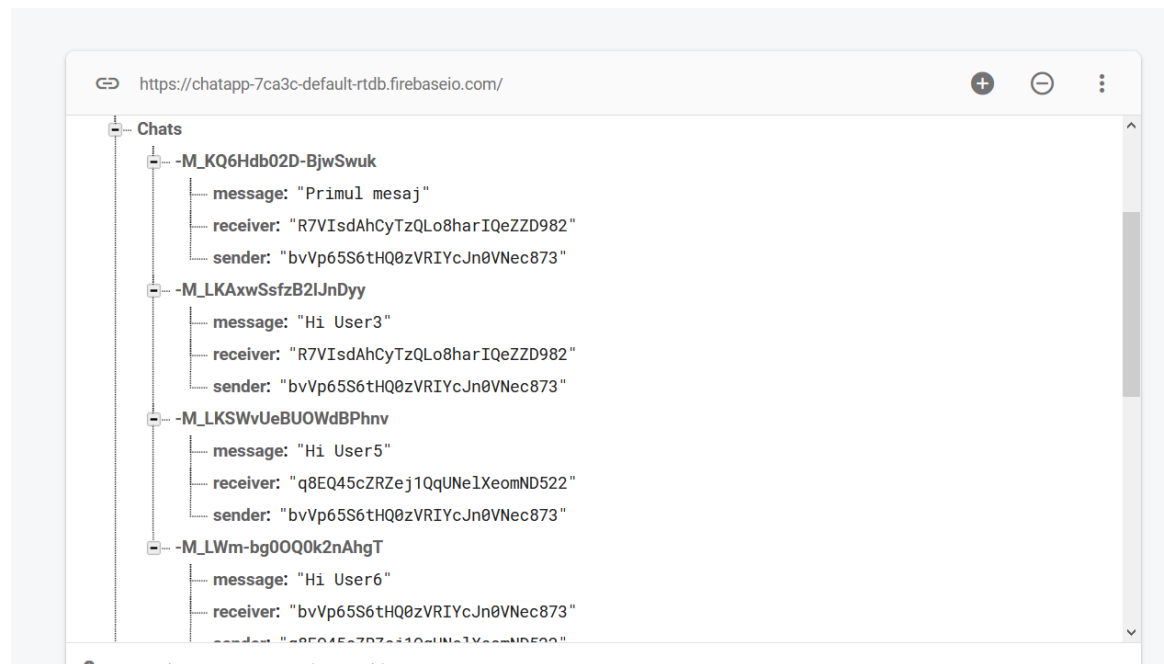
private void sendMessage(String sender, String receiver, String message){
    DatabaseReference reference = FirebaseDatabase.getInstance().getReference();

    HashMap<String, Object> hashMap = new HashMap<>();
    hashMap.put("sender", sender);
    hashMap.put("receiver", receiver);
    hashMap.put("message", message);

    reference.child("Chats").push().setValue(hashMap);
}
}

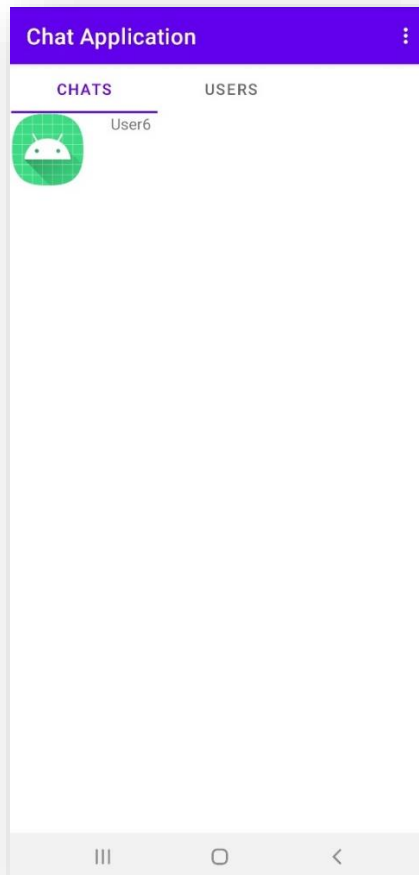
```

Conversatiile sunt, de asemenea, pastrate in baza de date Firebase in meniul “Chats”.



Istoric conversatii:

In meniul 'Chats' al aplicatiei se pot vizualiza conversatiile recente.



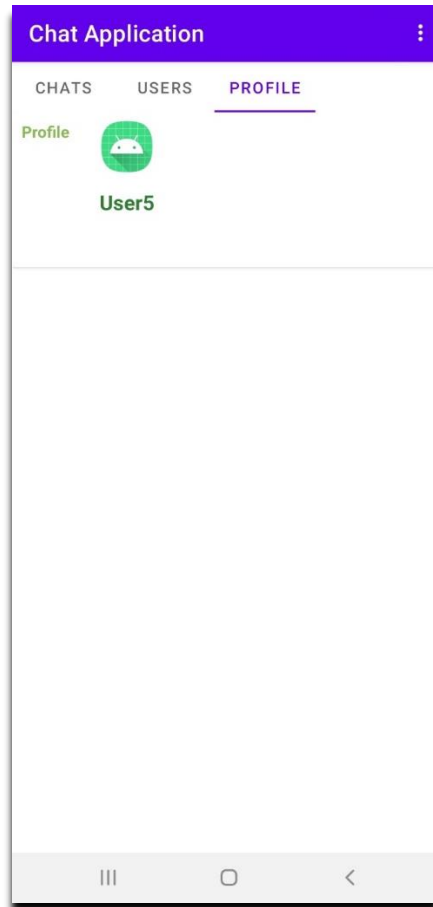
```
final DatabaseReference chatRef = FirebaseDatabase.getInstance()
    .getReference("ChatList")
    .child(fuser.getUid())
    .child(userid);

chatRef.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        if (!dataSnapshot.exists()){
            chatRef.child("id").setValue(userid);
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});
```

Vizualizare Profil:

Utilizatorul conectat isi poate vizualiza profilul.



```
package com.example.chatapplication.Fragments;

import ...

public class ProfileFragment extends Fragment {

    TextView username;
    ImageView imageView;

    DatabaseReference reference;
    FirebaseUser fuser;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment

        View view = inflater.inflate(R.layout.fragment_profile, container, false);

        imageView = view.findViewById(R.id.profile_image);
    }
}
```

```

        username = view.findViewById(R.id.username);

        fuser = FirebaseAuth.getInstance().getCurrentUser();
        reference = FirebaseDatabase.getInstance().getReference("MyUsers")
                .child(fuser.getId());

        reference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {

                Users user = snapshot.getValue(Users.class);
                username.setText(user.getUsername());

                if (user.getImageURL().equals("default")){
                    imageView.setImageResource(R.mipmap.ic_launcher);
                }else{
                    Glide.with(getContext()).load(user.getImageURL()).into(imageView);
                }

            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }

        });

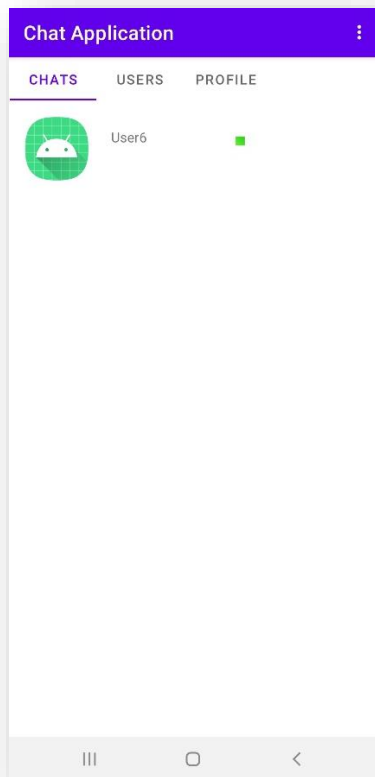
        return view;
    }
}

```

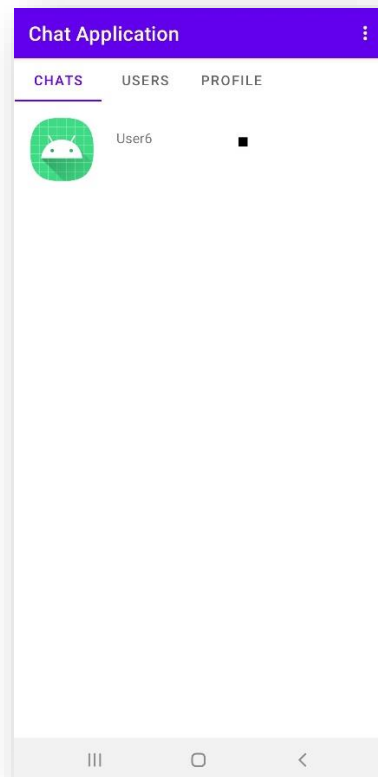
Afisare status utilizator:

Fiecare utilizator din lista de mesaje va fi marcat cu verde daca este 'online' sau cu negru daca este 'offline'.

Utilizator ONLINE

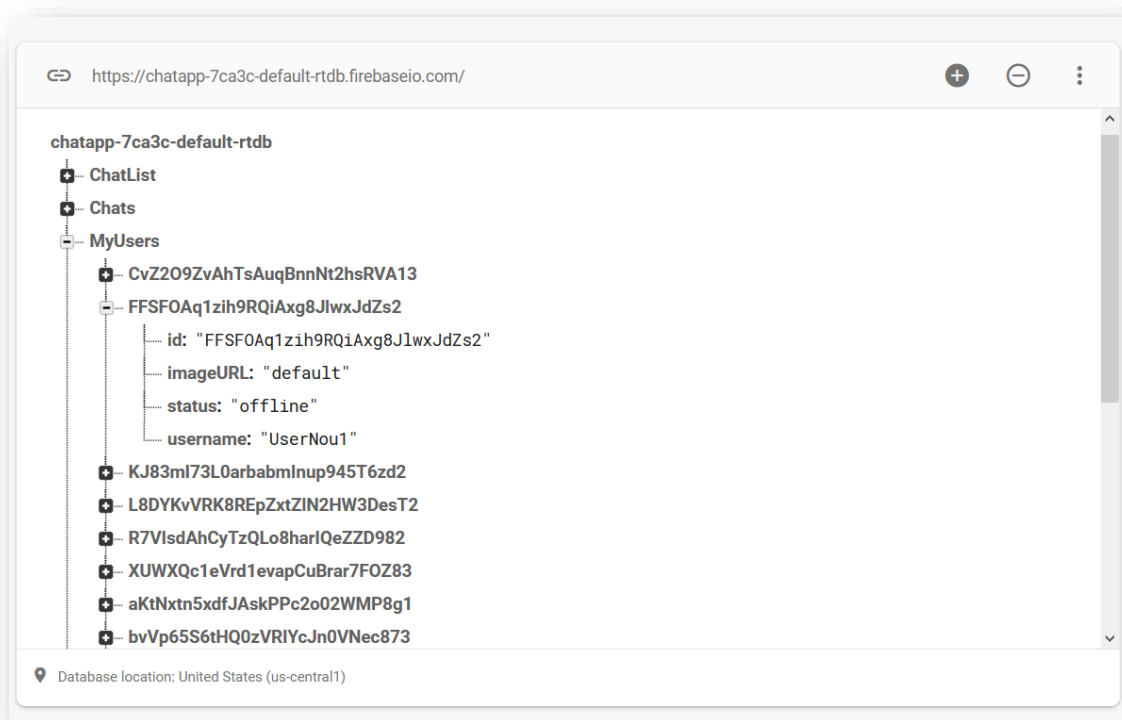


Utilizaor OFFLINE



```
private void CheckStatus(String status){  
  
    ref = FirebaseDatabase.getInstance().getReference("MyUsers").child(fireBaseUser.getId());  
  
    HashMap<String, Object> hashMap = new HashMap<>();  
    hashMap.put("status", status);  
  
    ref.updateChildren(hashMap);  
}  
@Override  
protected void onResume() {  
    super.onResume();  
    CheckStatus("online");  
}  
  
@Override  
protected void onPause() {  
    super.onPause();  
    CheckStatus("Offline");  
}
```

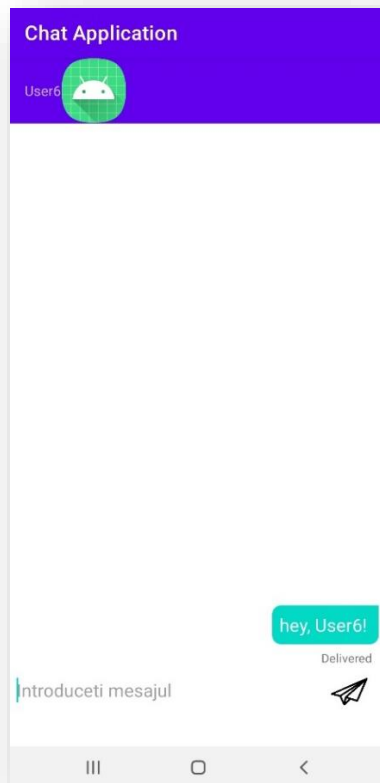
Fiecare utilizator nou creat va avea acum un nou parametru, 'status', care este stocat in baza de date Firebase astfel:



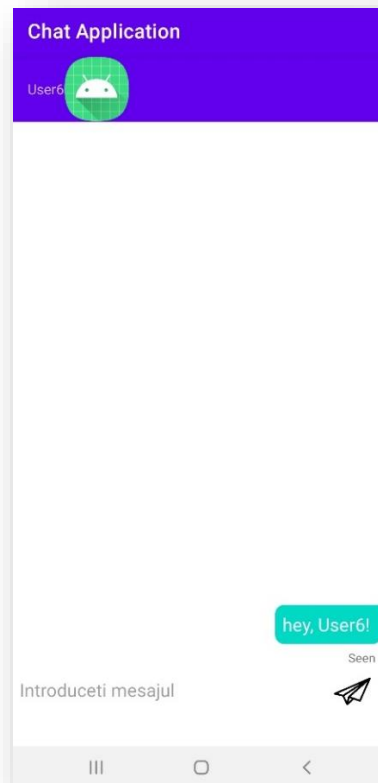
Afisare status mesaj

Dupa actionarea butonului de trimitere, in partea de jos a mesajului va aparea 'delivered' dupa ce acesta a fost trimis si 'seen' dupa ce acesta a fost deschis de catre destinatar.

Mesaj trimis



Mesaj deschis de utilizator



```
private void SeenMessage(String userid){  
    reference = FirebaseDatabase.getInstance().getReference("Chats");  
    seenListener = reference.addValueEventListener(new ValueEventListener() {  
        @Override  
        public void onDataChange(@NonNull DataSnapshot snapshot) {  
            for(DataSnapshot snapshot1 : snapshot.getChildren()){  
                Chat chat = snapshot1.getValue(Chat.class);  
                if(chat.getReceiver().equals(fuser.getId()) && chat.getSender().equals(userid)){  
                    HashMap<String, Object> hashMap = new HashMap<>();  
                    hashMap.put("isseen", true);  
                    snapshot1.getRef().updateChildren(hashMap); }}            }  
        }  
    });  
}
```

Ierarhia de fisiere:

Ierarhia de fisiere a aplicatiei va arata in final astfel:

