



المدرسة الوطنية للمهندسين بقرطاج

Ecole Nationale d'Ingénieurs de Carthage

Projet Développement Mobile

Réservation Des Spectacles

Réalisée par :

Mohamed Amine Guizani

2ème Année Informatique Group D

Encadrée par :

Madame Olfa Lamouchi

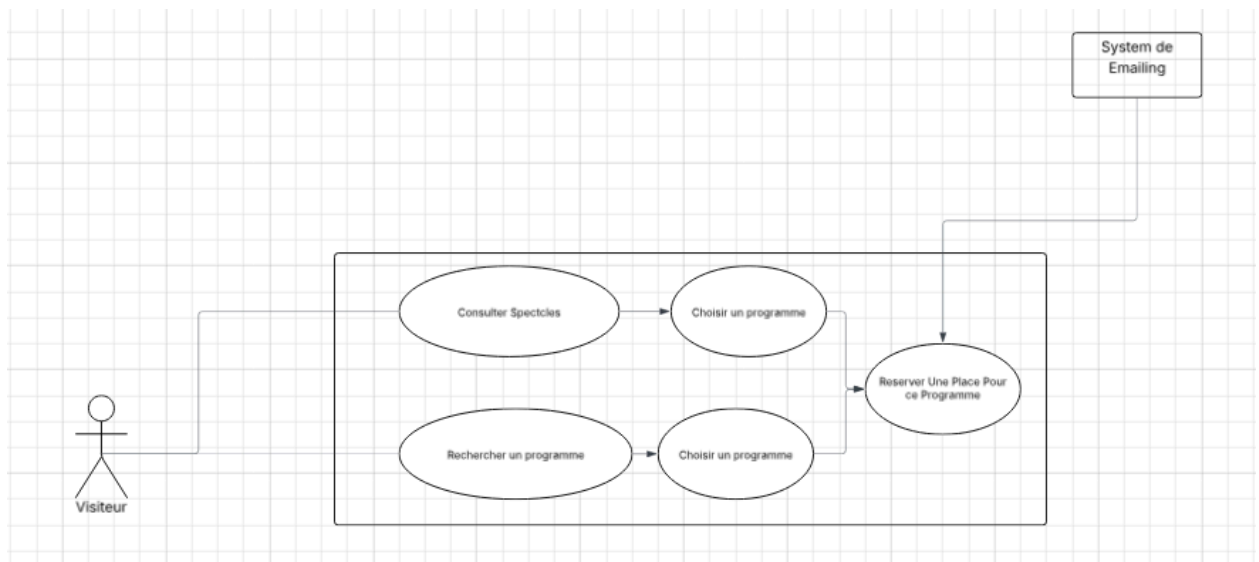
Date : 30/4/2025

1. Description du projet

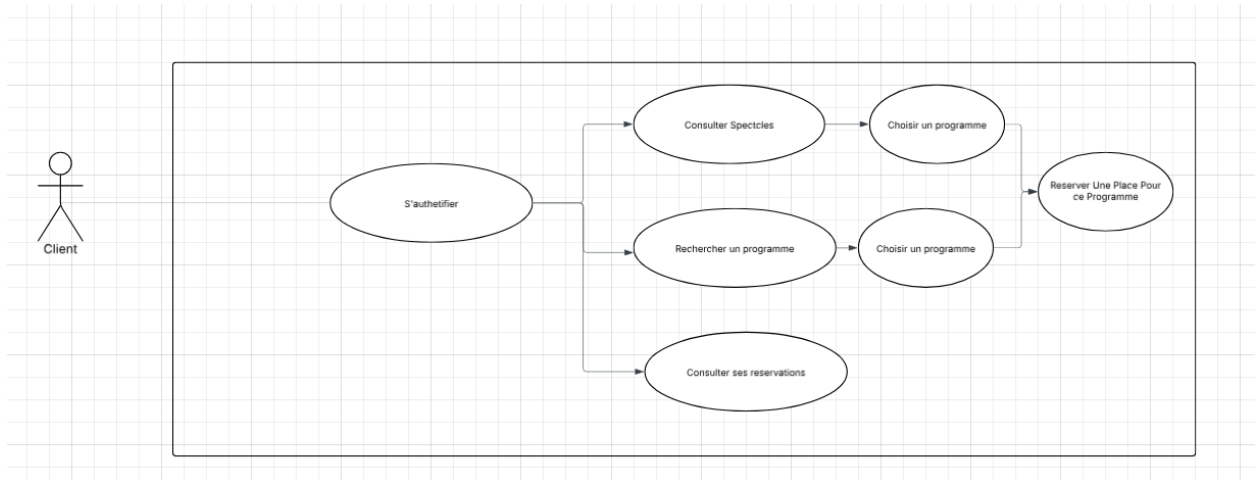
Le projet **SpectaBook** est une application mobile Android permettant aux utilisateurs de consulter les spectacles disponibles, de rechercher des programmes selon différents critères (date, lieu, type), et de réserver leurs places directement depuis leur smartphone. L'application propose une interface intuitive et moderne basée sur le composant **ConstraintLayout** ainsi que des éléments tels que **RecyclerView** et **CardView** pour une navigation fluide. Les utilisateurs peuvent accéder à un formulaire de réservation, et un e-mail de confirmation leur est automatiquement envoyé après validation. Côté technique, le projet est développé en **Java** avec une architecture **MVVM**, et communique avec une **API REST** pour la gestion des données (spectacles, réservations). Un système d'**envoi d'e-mails** est intégré pour assurer la confirmation des réservations. Cette application a été pensée pour offrir une expérience utilisateur simple et efficace, sans nécessiter de création de compte pour effectuer une réservation.

2. Diagramme cas utilisation

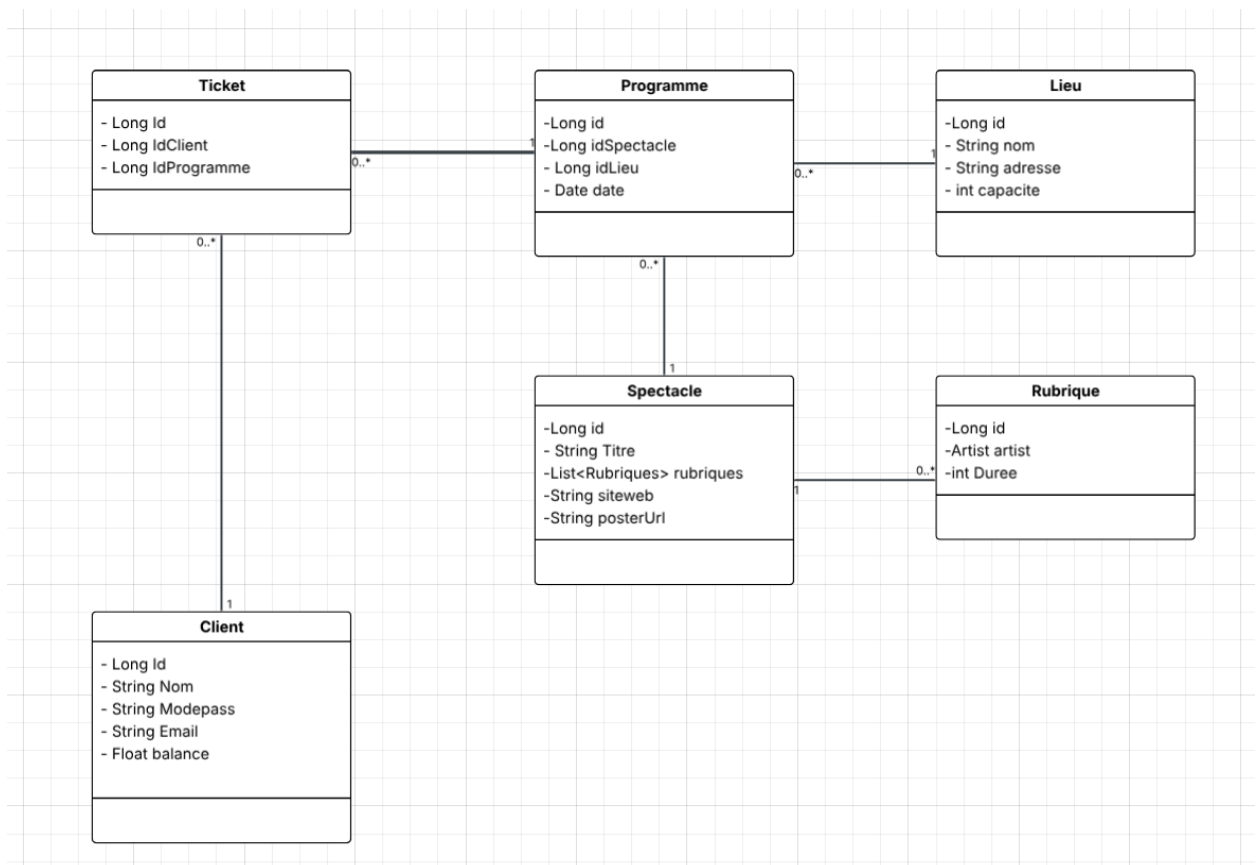
1. Diagramme cas utilisation pour un visiteur



2. Diagramme cas utilisation pour un Client



3. Diagramme de Class



4. BD et Api de connection

On a utiliser Mysql pour la base de donnée

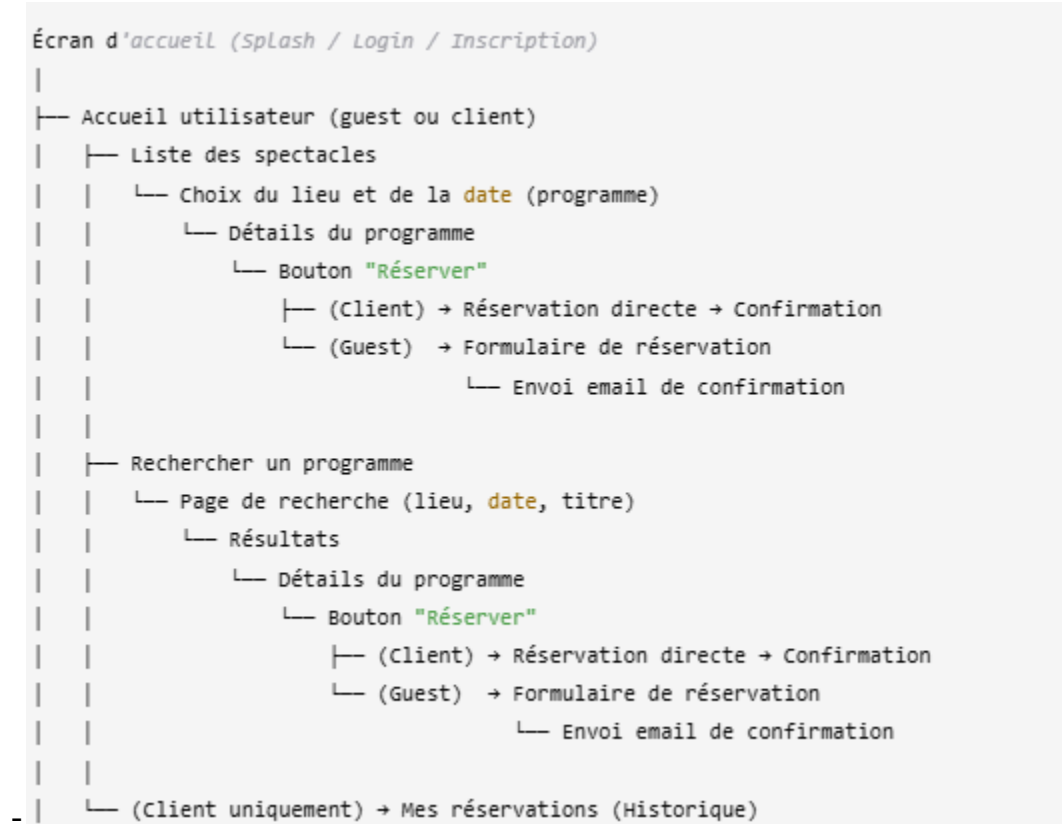


On a utiliser Spring Boot pour la création de REST API



Pour le développement du backend de l'application, j'ai choisi **Spring Boot** car il s'agit d'un framework robuste, moderne et très répandu dans l'écosystème Java. Il permet de créer rapidement des applications web sécurisées et évolutives grâce à sa structure modulaire, son intégration facile avec JPA/Hibernate et son excellent support de la gestion des dépendances via Spring Initializr. Quant à la base de données, mon choix s'est porté sur **MySQL** en raison de sa fiabilité, sa performance et sa large compatibilité avec les outils Java. C'est une base de données relationnelle open source bien adaptée aux applications nécessitant une gestion structurée et cohérente des données comme les réservations, les utilisateurs et les programmes dans SpectaBook.

5. Arbre des interfaces



6. Outils utilisés et Difficultés rencontrées

- **Android Studio** : pour le développement de l'application mobile Android.
- **Java / Kotlin** : pour la logique métier et l'implémentation des interfaces utilisateurs.
- **Spring Boot** : framework backend pour la création de l'API RESTful.
- **MySQL** : base de données relationnelle utilisée pour stocker les utilisateurs, les spectacles, les programmes et les réservations.

- **Postman** : pour tester les requêtes HTTP envoyées au backend.
- **Git & GitHub** : pour le versionnement du code et le travail collaboratif.
- **Firebase (optionnel)** : pour l'envoi d'e-mails de confirmation ou l'authentification (si utilisé).
- **PlantUML** : pour la modélisation des cas d'utilisation, arbres d'interface et diagrammes de classes.

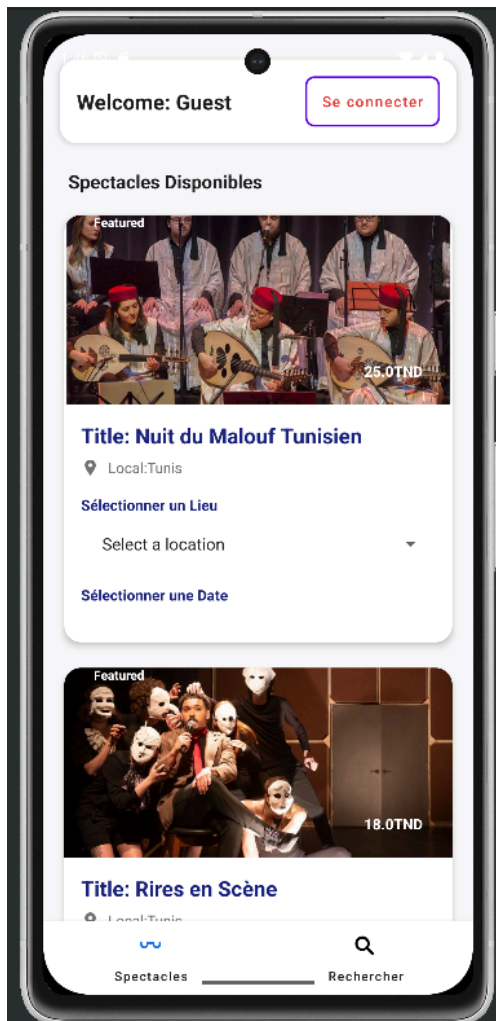
Les Difficultés rencontrées :

- **Gestion des utilisateurs (invité vs client)** : il a fallu mettre en place une logique conditionnelle pour distinguer les actions accessibles selon le type d'utilisateur (ex. formulaire pour les invités).
- **Navigation complexe dans l'app** : permettre un accès fluide aux détails de spectacles via deux entrées (recherche ou liste principale) tout en gardant une expérience utilisateur cohérente.
- **Connexion entre le frontend Android et le backend Spring Boot** : configurer correctement les appels réseau, gérer les erreurs et assurer la communication fluide avec l'API REST.
- **Validation du formulaire de réservation** : surtout pour les invités, avec vérification des champs obligatoires avant l'envoi de l'email de confirmation.

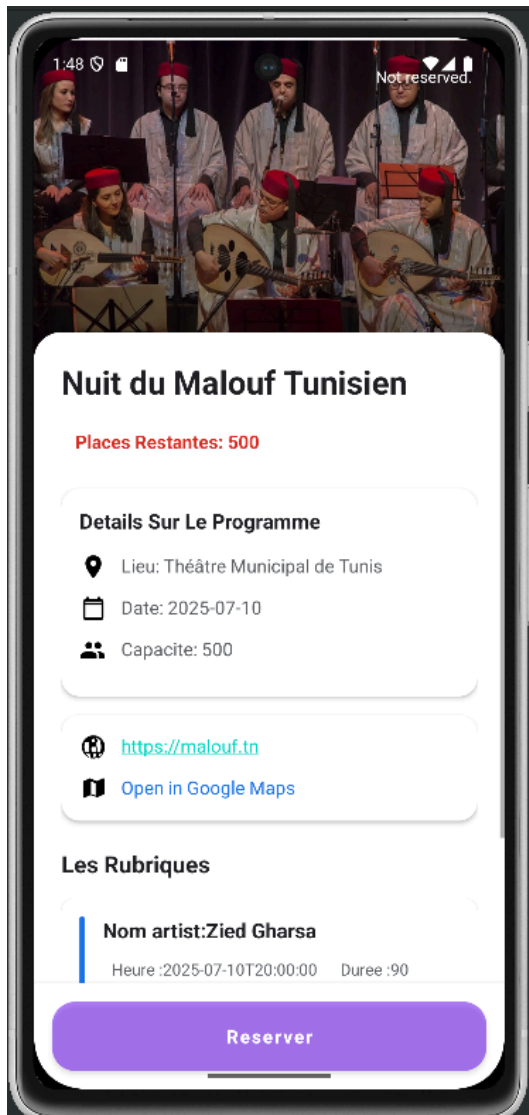
7. Les Interfaces

7.1 les Interfaces :

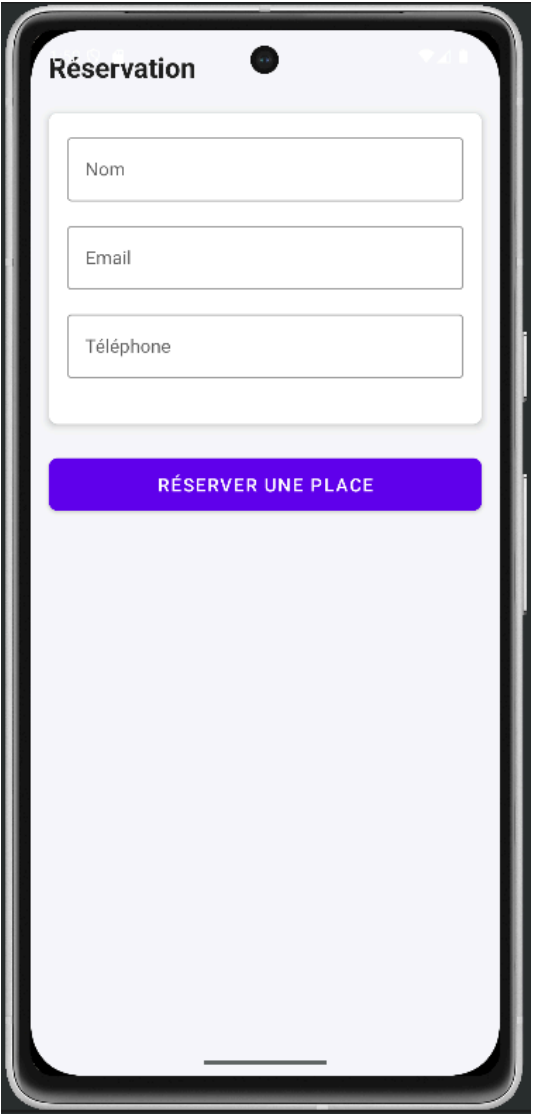
Interface d'accueil :



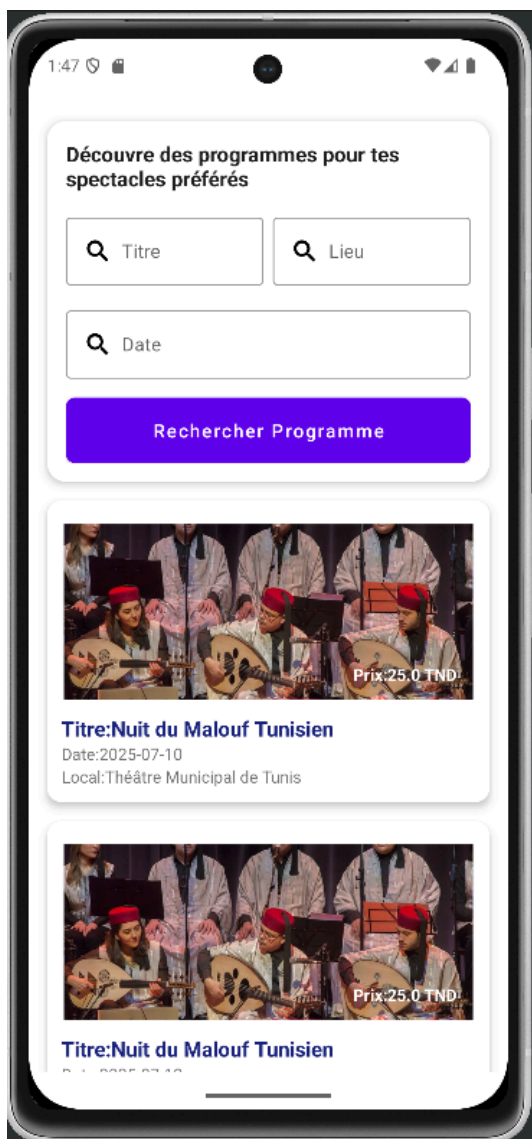
Interface de détails sur un programme :



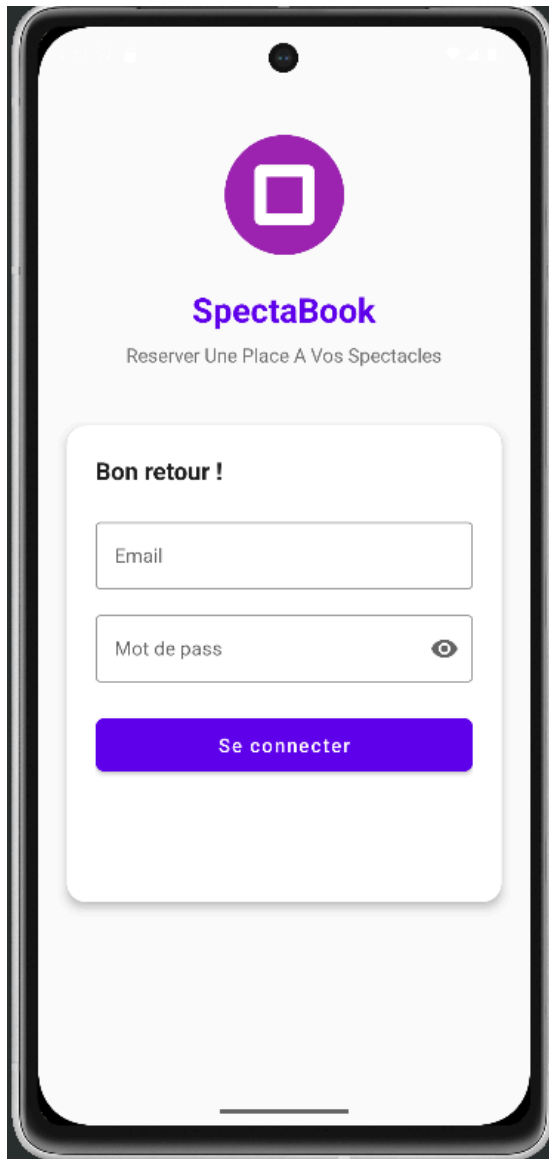
Interface Formulaire de réservation :



Interface de Recherche des Programmes :



Interface Se connecter :



8. Annex : Code des parties importantes

ApiService :

```
package com.example.spectacleappfront;
```

```
import java.util.List;

import retrofit2.Call;
import retrofit2.http.Body;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
import retrofit2.http.GET;
import retrofit2.http.POST;
import retrofit2.http.Path;
import retrofit2.http.Query;

public interface ApiService {

    @GET("/spectacles")
    Call<List<Spectacle>> getSpectacles();

    @GET("/programmes")
    Call<List<Programme>> getProgrammes();

    @POST("/login")
    Call<User> loginUser(@Body LoginRequest loginRequest);

    @POST("/reserved/{spectacleId}/{userId}")
    Call<Boolean> getReserved();

    @GET("/programmes/reservations")
    Call<List<Programme>> getReservedProgrammes(
        @Query("userId") Long userId
    );

    @GET("/spectacles/{specId}")
    Call<List<Lieu>> getLieuxBySpectacleId(@Path("specId") Long specId);

    @GET("/spectacles/{specId}/{lieuId}")
    Call<List<String>> getDatesBySpectacleIdAndLieu(@Path("specId") Long specId,
        @Path("lieuId") Long lieuId);

    @GET("/programmes/date")
    Call<Programme> getProgramme(@Query("date") String date,
        @Query("specId") Long specId,
        @Query("lieuId") Long lieuId);

    @POST("/send-confirmation-email")
    @FormUrlEncoded
```

```

        Call<Void> sendConfirmationEmail(@Field("email") String email ,
@Field("name") String name,@Field("programmeId") Long programmeId);
    }
}

```

ApiClient :

```

package com.example.spectacleappfront;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class ApiClient {
    private static final String BASE_URL = "http://10.0.2.2:9090"; // use
10.0.2.2 for localhost in emulator
    private static Retrofit retrofit = null;

    public static Retrofit getClient() {
        if (retrofit == null) {
            retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();
        }
        return retrofit;
    }
}

```

Login Activity :

```

package com.example.spectacleappfront;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;

```

```

import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class LoginActivity extends AppCompatActivity {
    private EditText editTextEmail, editTextPassword;
    private Button buttonLogin;

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

        editTextEmail = findViewById(R.id.editTextEmail);
        editTextPassword = findViewById(R.id.editTextPassword);
        buttonLogin = findViewById(R.id.buttonLogin);

        buttonLogin.setOnClickListener(v -> loginUser());
    }

    private void loginUser() {
        String email = editTextEmail.getText().toString().trim();
        String password = editTextPassword.getText().toString().trim();

        if (email.isEmpty() || password.isEmpty()) {
            Toast.makeText(this, "Please fill all fields",
Toast.LENGTH_SHORT).show();
            return;
        }

        LoginRequest request = new LoginRequest(email, password);
        ApiService apiService = ApiClient.getClient().create(ApiService.class);
        Call<User> call = apiService.loginUser(request);

        call.enqueue(new Callback<User>() {
            @Override
            public void onResponse(Call<User> call, Response<User> response) {
                if (response.isSuccessful() && response.body() != null) {

```

```

        User user = response.body();
        Log.d("LoginCheck", "isUserLoggedIn = " + user);

        // Save login state and user info
        SharedPreferences prefs = getSharedPreferences("MyPrefs",
MODE_PRIVATE);

        SharedPreferences.Editor editor = prefs.edit();
        editor.putBoolean("isLoggedIn", true);
        editor.putLong("userId", user.getId());
        editor.putString("userEmail", user.getEmail());
        editor.putString("userName", user.getName());
        editor.putFloat("userBalance", user.getBalance());

        editor.apply();

        Toast.makeText(LoginActivity.this, "Login successful",
Toast.LENGTH_SHORT).show();

        // Redirect to MainActivity
        Intent intent = new Intent(LoginActivity.this,
MainActivity.class);
        startActivity(intent);
        finish();
    } else {
        Toast.makeText(LoginActivity.this, "Invalid credentials",
Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onFailure(Call<User> call, Throwable t) {
    Toast.makeText(LoginActivity.this, "Login failed: " +
t.getMessage(), Toast.LENGTH_LONG).show();
}
});
}
}

```

Main Activity (Page des spectacles) :

```

package com.example.spectacleappfront;

import android.content.Intent;

```

```
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.List;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class MainActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private SpectacleAdapter adapter;
    private List<Spectacle> spectaclesList = new ArrayList<>();

    private Button logoutButton;
    private TextView welcomeTextView;

    @Override
    protected void onStart() {
        super.onStart();
    }

    private boolean isUserLoggedIn() {
        SharedPreferences prefs = getSharedPreferences("MyPrefs", MODE_PRIVATE);
        return prefs.getBoolean("isLoggedIn", false);
    }

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

        SharedPreferences prefs = getSharedPreferences("MyPrefs", MODE_PRIVATE);
        String username = prefs.getString("userName", "Guest");
        boolean isLoggedIn = prefs.getBoolean("isLoggedIn", false);
```



```

// UI Setup
recyclerView = findViewById(R.id.recyclerView);

logoutButton = findViewById(R.id.btnLogout);
logoutButton.setText(isLoggedIn ? "Se déconnecter" : "Se connecter");

welcomeTextView = findViewById(R.id.tvWelcome);
welcomeTextView.setText("Welcome: " + username);

recyclerView.setLayoutManager(new LinearLayoutManager(this));
adapter = new SpectacleAdapter(MainActivity.this, new ArrayList<>());
recyclerView.setAdapter(adapter);

fetchSpectacles();

logoutButton.setOnClickListener(v -> {

    if (isLoggedIn) {

        SharedPreferences.Editor editor = prefs.edit();
        editor.clear();
        editor.apply();
        logoutButton.setText("Se connecter");

    } else {

        Intent intent = new Intent(MainActivity.this,
LoginActivity.class);
        startActivity(intent);
    }
});

View btnSpectacles = findViewById(R.id.btn_spectacles);
View btnSearch = findViewById(R.id.btn_search);
View btnReservations = findViewById(R.id.btn_reservations);

btnReservations.setVisibility(isLoggedIn ? View.VISIBLE : View.GONE);
btnSpectacles.setOnClickListener(v -> {
    startActivity(new Intent(MainActivity.this, MainActivity.class));
});

```

```

        btnSearch.setOnClickListener(v -> {
            startActivity(new Intent(MainActivity.this,
RechercherProgramme.class));
        });

        btnReservations.setOnClickListener(v -> {

            startActivity(new Intent(MainActivity.this,
ProgrammesReservees.class));
        });
    }

    private void fetchSpectacles() {
        ApiService apiService = ApiClient.getClient().create(ApiService.class);
        Call<List<Spectacle>> call = apiService.getSpectacles();

        call.enqueue(new Callback<List<Spectacle>>() {
            @Override
            public void onResponse(Call<List<Spectacle>> call,
Response<List<Spectacle>> response) {
                Log.d("data fetch result", "working ");
                if (response.isSuccessful() && response.body() != null) {
                    spectaclesList = response.body();
                    adapter.setSpectacleList(spectaclesList);
                } else {
                    Log.e("MainActivity", "Failed to fetch data. Response code:
" + response.code());
                }
            }

            @Override
            public void onFailure(Call<List<Spectacle>> call, Throwable t) {
                Log.e("MainActivity", "Error fetching data: " + t.getMessage());
            }
        });
    }
}

```

Page Recherche des Programmes pour un spectacle :

```
package com.example.spectacleappfront;

import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.List;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class RechercherProgramme extends AppCompatActivity {

    private EditText editTextTitle, editTextLocation, editTextDate;
    private Button searchButton;
    private RecyclerView recyclerView;
    private ProgrammeAdapter adapter;
    private List<Programme> programmesSearched = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_rechercher_programme);

        // Inset handling...
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets sys = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(sys.left, sys.top, sys.right, sys.bottom);
            return insets;
        });
    }
}
```

```

// 1) find views
editTextTitle    = findViewById(R.id.editTextTitle);
editTextLocation = findViewById(R.id.editTextLocation);
editTextDate     = findViewById(R.id.editTextDate);
searchButton     = findViewById(R.id.searchButton);
recyclerView     = findViewById(R.id.recyclerViewProgrammes);

// 2) setup RecyclerView + Adapter
adapter = new ProgrammeAdapter(this, programmesSearched);
Log.d("ps dans adapterinit", programmesSearched.toString());
recyclerView.setLayoutManager(new LinearLayoutManager(this));
recyclerView.setAdapter(adapter);

// 3) fetch all programmes from backend
fetchProgrammes();

// 4) filter on button click
searchButton.setOnClickListener(v -> {
    String t = editTextTitle.getText().toString().trim().toLowerCase();
    String l =
editTextLocation.getText().toString().trim().toLowerCase();
    String d = editTextDate.getText().toString().trim().toLowerCase();

    List<Programme> filtered = new ArrayList<>();
    for (Programme p : programmesSearched) {
        boolean byTitle    = t.isEmpty() ||
p.getSpectacle().getTitle().toLowerCase().contains(t);
        boolean byLocation = l.isEmpty() ||
p.getLieu().getNom().toLowerCase().contains(l);
        boolean byDate     = d.isEmpty() ||
p.getDate().toLowerCase().contains(d);
        if (byTitle && byLocation && byDate) {
            filtered.add(p);
        }
    }
    adapter.updateData(filtered);
});
}

private void fetchProgrammes() {
    ApiService api = ApiClient.getClient().create(ApiService.class);
    api.getProgrammes().enqueue(new Callback<List<Programme>>() {
        @Override
        public void onResponse(Call<List<Programme>> call,
Response<List<Programme>> res) {
            Log.d("ps dans RP", res.body().toString());
            if (res.isSuccessful() && res.body() != null) {
                programmesSearched = res.body();
                adapter.updateData(programmesSearched);
            }
        }
    });
}

```

```

        } else {
            Log.e("RechercherProgramme", "Fetch failed: " + res.code());
        }
    }
    @Override
    public void onFailure(Call<List<Programme>> call, Throwable t) {
        Log.e("RechercherProgramme", "Error: " + t.getMessage());
    }
});
Log.d("programme list :",programmesSearched.toString());
}
}

```

Programme Details activity :

```

package com.example.spectacleappfront;

import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.bumptech.glide.Glide;

```

```

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class SpectacleDetailActivity extends AppCompatActivity {
    private TextView tvTitle, tvLocation, tvDate, tvCapacity, tvWebsite,
spotsLeftTextView, reservationInfo, tvGoogleMaps;
    private Button reserveButton;
    private LinearLayout rootLayout;
    private RecyclerView rubriqueRecyclerView;
    private RubriqueAdapter rubriqueAdapter;
    private ImageView imagePoster;

    private long userId;
    private String userName;
    private long spectacleId , programmeId;
    private int capacity;
    private String website;
    private RequestQueue queue;

    private String posterurl;

    private String gloc;

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

        // Initialize views
        tvTitle = findViewById(R.id.tvTitle);
        tvLocation = findViewById(R.id.tvLocation);
        tvDate = findViewById(R.id.tvDate);
        tvCapacity = findViewById(R.id.tvCapacity);
        tvWebsite = findViewById(R.id.tvWebsite);
        spotsLeftTextView = findViewById(R.id.spotsLeft);
        reservationInfo = findViewById(R.id.tvReservationInfo);
        reserveButton = findViewById(R.id.reservationButton);
        rootLayout = findViewById(R.id.spectacleRootLayout);
        imagePoster = findViewById(R.id.imagePoster);
        tvGoogleMaps= findViewById(R.id.tvGoogleMaps);
        rubriqueRecyclerView = findViewById(R.id.rubriqueRecyclerView);
        queue = Volley.newRequestQueue(this);

        // Get user info from SharedPreferences
        SharedPreferences prefs = getSharedPreferences("MyPrefs", MODE_PRIVATE);
        userId = prefs.getLong("userId", -1);
        Log.d("LoginCheck", "in spectacle detail page = " + userId);
    }
}

```

```

        userName = prefs.getString("userName", "Unknown");

        // Get spectacle data from Intent
        Intent intent = getIntent();
        programmeId = intent.getLongExtra("progid", -1);
        spectacleId = intent.getLongExtra("specid", -1);
        String title = intent.getStringExtra("title");
        String location = intent.getStringExtra("location");
        String date = intent.getStringExtra("date");
        website = intent.getStringExtra("website");
        capacity = intent.getIntExtra("cap", 0);
        posterurl = intent.getStringExtra("posterurl");
        gloc = intent.getStringExtra("gloc");
        ArrayList<Rubrique> rubriques = (ArrayList<Rubrique>)
getIntent().getSerializableExtra("rubriques");
        if (rubriques != null) {
            rubriqueAdapter = new RubriqueAdapter(rubriques);
            rubriqueRecyclerView.setLayoutManager(new
WrapContentLinearLayoutManager(this, LinearLayoutManager.VERTICAL, false));
            rubriqueRecyclerView.setAdapter(rubriqueAdapter);
        }
        Glide.with(this).load(posterurl).into(imagePoster);

        // Set static texts
        tvTitle.setText(title);
        tvLocation.setText("Lieu: " + location);
        tvDate.setText("Date: " + date);
        tvCapacity.setText("Capacite: " + capacity);
        tvWebsite.setText(website);

        // Check reservation status and spots left
        checkReservationStatus();
        fetchSpotsLeft();
        tvGoogleMaps.setOnClickListener(v -> {
            if (gloc != null && !gloc.isEmpty()) {
                String url = "https://www.google.com/maps/search/?api=1&query="
+ Uri.encode(gloc);
                Intent mapIntent = new Intent(Intent.ACTION_VIEW,
Uri.parse(url));
                startActivity(mapIntent);
            } else {
                Toast.makeText(this, "Location is missing",
Toast.LENGTH_SHORT).show();
            }
        });

        // Handle website click
        tvGoogleMaps.setOnClickListener(v -> {
            if (gloc != null && !gloc.isEmpty()) {

```

```

        Uri gmmIntentUri = Uri.parse("geo:0,0?q=" + Uri.encode(gloc));
        Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
        // mapIntent.setPackage("com.google.android.apps.maps"); //
REMOVE THIS LINE

        // Let Android handle the best app (Maps app or browser)
        if (mapIntent.resolveActivity(getPackageManager()) != null) {
            startActivity(mapIntent);
        } else {
            Toast.makeText(this, "No app available to handle maps",
Toast.LENGTH_SHORT).show();
        }
    }

});

// Reserve button logic
reserveButton.setOnClickListener(v -> {
    boolean isLoggedIn = prefs.getBoolean("isLoggedIn", false);

    if (isLoggedIn) {
        sendReservationRequest();
    } else {

        Intent intent1 = new Intent(SpectacleDetailActivity.this,
GuestReservationForm.class);
        intent1.putExtra("programmeId", programmeId);
        startActivity(intent1);
    }
});
}

private void checkReservationStatus() {
    String url = "http://10.0.2.2:9090/reservation/" + programmeId + "/" +
userId;

    StringRequest request = new StringRequest(Request.Method.GET, url,
        response -> {
            Log.d("ReservationCheck", "Response: " + response);
            if (Boolean.parseBoolean(response.trim())) {

rootLayout.setBackgroundColor(Color.parseColor("#C8E6C9"));
                reservationInfo.setText("Programme est deja Reservee
Par: " + userName);
                reserveButton.setVisibility(View.GONE);
            } else {
                reservationInfo.setText("Not reserved.");
            }
        },
        error -> {

```



```

        Toast.makeText(this, "Failed to check reservation",
Toast.LENGTH_SHORT).show();
        Log.e("ReservationCheck", "Error: " + error.toString());
    }

);

queue.add(request);
}

private void fetchSpotsLeft() {
    String url = "http://10.0.2.2:9090/spectacles/spots/" + programmeId;
    StringRequest request = new StringRequest(Request.Method.GET, url,
        response -> spotsLeftTextView.setText("Places Restantes: " +
response),
        error -> spotsLeftTextView.setText("Spots left: N/A")
    );
    queue.add(request);
}

private void sendReservationRequest() {
    String url = "http://10.0.2.2:9090/reserve";

    StringRequest request = new StringRequest(Request.Method.POST, url,
        response -> {
            Toast.makeText(this, response, Toast.LENGTH_SHORT).show();
// Assuming string like "Reservation successful!"
            checkReservationStatus();
            fetchSpotsLeft();
        },
        error -> {
            if (error.networkResponse != null &&
error.networkResponse.data != null) {
                String errorMsg = new
String(error.networkResponse.data);
                Toast.makeText(this, "Error: " + errorMsg,
Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(this, "Failed to reserve",
Toast.LENGTH_SHORT).show();
            }
        }
    ) {
        @Override
        protected Map<String, String> getParams() {
            Map<String, String> params = new HashMap<>();
            params.put("userId", String.valueOf(userId));
            Log.d("reserve request is : ", "user id : " + userId +
"programmeId:" + programmeId);
            params.put("programmeId", String.valueOf(programmeId));

```

```

        return params;
    }
    @Override
    public String getBodyContentType() {
        return "application/x-www-form-urlencoded; charset=UTF-8";
    }
};

queue.add(request);
}
}

```

Historique des Réservations pour un client :

```

package com.example.spectacleappfront;

import android.os.Bundle;
import android.util.Log;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.List;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class ProgrammesReservees extends AppCompatActivity {

    private RecyclerView recyclerView;
    private ProgrammeAdapter adapter;
    private List<Programme> programmeList = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

setContentView(R.layout.activity_programmes_reserves);

// 1) wire up RecyclerView + adapter
recyclerView = findViewById(R.id.recyclerViewProgrammes);
recyclerView.setLayoutManager(new LinearLayoutManager(this));
adapter = new ProgrammeAdapter(this, programmeList);
recyclerView.setAdapter(adapter);
// 2) fetch the userId (e.g. from SharedPreferences)
long userId = getSharedPreferences("MyPrefs", MODE_PRIVATE)
    .getLong("userId", -1L);
// 3) call the endpoint
ApiService api = ApiClient.getClient().create(ApiService.class);
api.getReservedProgrammes(userId)
    .enqueue(new Callback<List<Programme>>() {
        @Override
        public void onResponse(Call<List<Programme>> call,
                               Response<List<Programme>> response) {
            if (response.isSuccessful() && response.body() != null)
            {
                List<Programme> list = response.body();
                adapter.updateData(list);
                Log.d("list of reservations :", list.toString());
            } else {
                Log.e("MyReservations", "Failed: " +
response.code());
            }
        }

        @Override
        public void onFailure(Call<List<Programme>> call, Throwable
t) {
            Log.e("MyReservations", "Error: " + t.getMessage());
        }
    });
}
}

```