Projet 7 : Trouvez un restaurant pour déjeuner avec vos collègues

Projet 7 : Trouvez un restaurant pour déjeuner avec vos collègues

Sommaire :

Définir et appliquer un thème global :

```
<!-- Base application theme. -->
<style name="AppTheme" parent="Theme.AppCompat.DayNight.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
    <item name="android:textColorPrimary">@android:color/white</item>
    <item name="android:textColorSecondary">@android:color/white</item>


</style>

<style name="AppThemeNavigation" parent="Theme.AppCompat.DayNight.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
    <item name="android:textColorPrimary">@android:color/black</item>
    <item name="android:textColorSecondary">@android:color/black</item>

</style>
```

```
<!-- STYLE SIGN-IN ACTIVITY -->
<style name="LoginTheme" parent="AppTheme">
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
    <item name="colorButtonNormal">@color/colorAccent</item>
    <item name="colorControlNormal">@android:color/darker_gray</item>
    <item name="colorControlActivated">@android:color/darker_gray</item>
    <item name="colorControlHighlight">@android:color/darker_gray</item>
    <item name="android:textColor">@android:color/white</item>
    <item name="android:textColorPrimary">@android:color/darker_gray</item>
    <item name="android:textColorSecondary">@android:color/darker_gray</item>
    <item name="android:windowBackground">@android:color/white</item>
    <item name="windowActionBar">false</item>
    <item name="windowNoTitle">true</item>
    <item name="android:textColorHint">@android:color/white</item>
</style>

<!-- STYLE FOR MAIN ACTIVITY -->
<style name="NoTitle" parent="AppTheme">
    <item name="windowNoTitle">true</item>
    <item name="windowActionBar">false</item>
    <item name="actionBarTheme">@null</item>
</style>
```

Connexion : LoginActivity

```java
@Override
public int getFragmentLayout() { return R.layout.activity_login; }

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    this.handleResponseAfterSignIn(requestCode, resultCode, data);
}

@Override
protected void onResume() {
    super.onResume();
    this.updateUIWhenResuming();
}

@OnClick(R.id.main_activity_button_login)
public void onClickLoginButton() {
    if (this.isCurrentUserLogged()){
        this.startProfileActivity();
    } else {
        this.startSignInActivity();
    }
}

@OnClick(R.id.main_activity_button_hungry)
public void onClickChatButton() {
    if (this.isCurrentUserLogged()){
        sharedpref();
        this.startHungryActivity();
    } else {
        this.showSnackBar(this.coordinatorLayout, "Vous devez vous connecter !");
    }
}
```

```java
private void createUserInFirestore(){

    if (this.getCurrentUser() != null){

        if (!getAllUser().contains(getUserCurrent())) {

            String urlPicture = (this.getCurrentUser().getPhotoUrl() != null) ? this.getCurrentUser()
                    .getPhotoUrl().toString() : null;
            String username = this.getCurrentUser().getDisplayName();
            String uid = this.getCurrentUser().getUid();

            Restaurant resto = new Restaurant("Pas encore choisit",  address: "?",  id: "?");
            boolean notification = true;

            ArrayList<String> restoLike = new ArrayList<~>();


            UserHelper.createUser(uid, username, urlPicture, resto, notification, restoLike ).
                    addOnFailureListener(this.onFailureListener());
        }

    }
}

private void startSignInActivity(){
    startActivityForResult(
            AuthUI.getInstance() AuthUI
                    .createSignInIntentBuilder() SignInIntentBuilder
                    .setTheme(R.style.LoginTheme) SignInIntentBuilder
                    .setAvailableProviders(
                            Arrays.asList(new AuthUI.IdpConfig.EmailBuilder().build(), //EMAIL
                                    new AuthUI.IdpConfig.GoogleBuilder().build(), //GOOGLE
                                    new AuthUI.IdpConfig.FacebookBuilder().build())) // FACEBOOK
                    .setIsSmartLockEnabled( enableCredentials: false,  enableHints: true) SignInIntentBuilder
                    .setLogo(R.drawable.logo) SignInIntentBuilder
                    .build(),
            RC_SIGN_IN);
}
```

Connexion : LoginActivity

```
private void startProfileActivity(){
    Intent intent = new Intent( packageContext: this, ProfilActivity.class);
    startActivity(intent);
}


private void startHungryActivity(){
    Intent intent = new Intent( packageContext: this,HungryActivity.class);
    startActivity(intent);
}


private void showSnackBar(CoordinatorLayout coordinatorLayout, String message){
    Snackbar.make(coordinatorLayout, message, Snackbar.LENGTH_SHORT).show();
}

private void updateUIWhenResuming(){
    this.buttonLogin.setText(this.isCurrentUserLogged() ? "Afficher le profil" : "Connexion");
}


private void handleResponseAfterSignIn(int requestCode, int resultCode, Intent data){

    if (requestCode == RC_SIGN_IN) {
        if (resultCode == RESULT_OK) { // SUCCESS
            this.createUserInFirestore();
            showSnackBar(this.coordinatorLayout, "Authentification réussie !");
        }
    }
}
```

Ecran d'accueil : HungryActivity

```
private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener
        = new BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        fragmentManager = getSupportFragmentManager();

        switch (item.getItemId()) {
            case R.id.navigation_mapView: fragmentManager.beginTransaction().replace(R.id.content_frame,firstFragment ).commit();
                return true;
            case R.id.navigation_listView: fragmentManager.beginTransaction().replace(R.id.content_frame, secondFragment).commit();
                return true;
            case R.id.navigation_workmates: fragmentManager.beginTransaction().replace(R.id.content_frame, thirdFragment).commit();
                return true;
        }
        return false;
    }
};

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

    mLayout = findViewById(R.id.mlayout);

    getAllUser();getAllUserListResto();getAllUserWithoutMyself();

    showRestaurants();

    BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.navigation);
    navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelectedListener);

    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            activity: this, drawer, toolbar, "Mettre à jour le nom d'utilisateur", "Go4Lunch");
    drawer.setDrawerListener(toggle);
    toggle.syncState();

    NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
    navigationView.setNavigationItemSelectedListener(this);

    fragmentManager = getSupportFragmentManager();
    fragmentManager.beginTransaction().replace(R.id.content_frame, new FirstFragment()).commit();
```

Vue des restaurants sous forme de carte : FirstFragment

```java
@Nullable
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    mView = inflater.inflate(R.layout.map_layout, container,  attachToRoot: false);

    FloatingActionButton button = (FloatingActionButton) mView.findViewById(R.id.recentrer);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            if (ActivityCompat.checkSelfPermission(mContext, android.Manifest.permission.ACCESS_FINE_LOCATION)
                    != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(mContext,
                    android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
                return;
            }
            Location mLastLocation = LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
            LatLng here = new LatLng(mLastLocation.getLatitude(), mLastLocation.getLongitude());

            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(here,  v: 14));

        }
    });

    return mView;
}


@Override
public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);

    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment) getChildFragmentManager().findFragmentById(R.id.map);
    mapFragment.getMapAsync( onMapReadyCallback: this);
}
```

```java
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    // Get location
    if (mGoogleApiClient == null) {
        mGoogleApiClient = new GoogleApiClient.Builder(mContext)
                .addConnectionCallbacks(this)
                .addOnConnectionFailedListener(this)
                .addApi(LocationServices.API)
                .build();

        mGoogleApiClient.connect();
    }


    mMap.setOnInfoWindowClickListener(new GoogleMap.OnInfoWindowClickListener() {
        @Override
        public void onInfoWindowClick(Marker marker) {


            String snippet = marker.getSnippet();
            String name = marker.getTitle();
            String[] separated = snippet.split( regex: ";");


            Restaurant resto = new Restaurant(name, separated[0], separated[1]);


            int comp = name.compareTo("VOUS");

            if (comp != 0) {
                Intent i = new Intent(getActivity(), DetailRestaurantActivity.class)
                i.putExtra(RESTAURANT, resto);

                startActivity(i);
            }

        }
    });
```

Vue des restaurants sous forme de carte : FirstFragment

```
@Override
public void onConnected(@Nullable Bundle bundle) {
    if (ActivityCompat.checkSelfPermission(mContext, android.Manifest.permission.ACCESS_FINE_LOCATION)
            != PackageManager.PERMISSION_GRANTED && ActivityCompat
            .checkSelfPermission(mContext, android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
        return;
    }

    Location mLastLocation = LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
    LatLng here = new LatLng(mLastLocation.getLatitude(), mLastLocation.getLongitude());
    mMap.addMarker(new MarkerOptions().position(here).title("VOUS").icon(BitmapDescriptorFactory.
            defaultMarker(BitmapDescriptorFactory.HUE_BLUE)));
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(here,  v: 14));

    double latitude = mLastLocation.getLatitude();
    double longitude = mLastLocation.getLongitude();

    String url = getUrl(latitude, longitude);
    Object[] DataTransfer = new Object[2];
    DataTransfer[0] = mMap;
    DataTransfer[1] = url;
    GetNearbyPlacesData getNearbyPlacesData = new GetNearbyPlacesData(getAllUserListResto(), mContext);
    getNearbyPlacesData.execute(DataTransfer);


}



private String getUrl(double latitude, double longitude) {

    StringBuilder googlePlacesUrl = new StringBuilder("https://maps.googleapis.com/maps/api/place/nearbysearch/json?");
    googlePlacesUrl.append("location=" + latitude + "," + longitude);
    googlePlacesUrl.append("&radius=" + 10000);
    googlePlacesUrl.append("&type=" + "meal_takeaway");
    googlePlacesUrl.append("&sensor=true");
    googlePlacesUrl.append("&key=" + API_KEY);
    return (googlePlacesUrl.toString());
}
```

Vue des restaurants sous forme de carte : GetNearbyPlaceData

```
public class GetNearbyPlacesData extends AsyncTask<Object, String, String> {

    String googlePlacesData;
    GoogleMap mMap;
    String url;
    List<String> coworkersResto;
    Context context;

    public GetNearbyPlacesData ( List<String> coworkersResto, Context c ) {
        this.coworkersResto = coworkersResto;
        this.context = c;
    }

    @Override
    protected String doInBackground(Object... params) {
        try {
            Log.d( tag: "GetNearbyPlacesData",  msg: "doInBackground entered");
            mMap = (GoogleMap) params[0];
            url = (String) params[1];
            DownloadUrl downloadUrl = new DownloadUrl();
            googlePlacesData = downloadUrl.readUrl(url);
            Log.d( tag: "GooglePlacesReadTask",  msg: "doInBackground Exit");
        } catch (Exception e) {
            Log.d( tag: "GooglePlacesReadTask", e.toString());
            Toast.makeText(context, "Echec de la requête. Ré-essayer", Toast.LENGTH_SHORT).show();
        }
        return googlePlacesData;
    }

    @Override
    protected void onPostExecute(String result) {
        Log.d( tag: "GooglePlacesReadTask",  msg: "onPostExecute Entered");
        List<HashMap<String, String>> nearbyPlacesList = null;
        DataParser dataParser = new DataParser();
        nearbyPlacesList =  dataParser.parse(result);
        ShowNearbyPlaces(nearbyPlacesList);
        Log.d( tag: "GooglePlacesReadTask",  msg: "onPostExecute Exit");
    }

    private void ShowNearbyPlaces(List<HashMap<String, String>> nearbyPlacesList) {

        if (nearbyPlacesList.size() == 0){
            Toast.makeText(context, "Echec de la requête. Ré-essayer", Toast.LENGTH_SHORT).show();
        }
```

```
or (int i = 0; i < nearbyPlacesList.size(); i++) {
    Log.d( tag: "onPostExecute",  msg: "Entered into showing locations");
    MarkerOptions markerOptions = new MarkerOptions();
    HashMap<String, String> googlePlace = nearbyPlacesList.get(i);
    double lat = Double.parseDouble(googlePlace.get("lat"));
    double lng = Double.parseDouble(googlePlace.get("lng"));


    String placeName = googlePlace.get("place_name");
    String vicinity = googlePlace.get("vicinity");

    String id_place =  googlePlace.get("reference");



    LatLng latLng = new LatLng(lat, lng);
    markerOptions.position(latLng);
    markerOptions.snippet(vicinity + " \n :" + id_place);
    markerOptions.title(placeName );


    if (coworkersResto.contains(id_place)) {
        markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN));
    }else{

        if (getUserCurrent().getRestoLike().contains(id_place)){
            markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ROSE));

        }else {
            markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED));

        }
    }

    mMap.addMarker(markerOptions);

    //move map camera
    mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
    mMap.animateCamera(CameraUpdateFactory.zoomTo( v: 11));
```

Vue des restaurants sous forme de liste : SecondFragment

```
StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);

if (ActivityCompat.checkSelfPermission(mContext, Manifest.permission.ACCESS_FINE_LOCATION)
        == PackageManager.PERMISSION_GRANTED) {
}

LocationManager lm = (LocationManager) mContext.getSystemService(Context.LOCATION_SERVICE);
Location location = lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
lng = location.getLongitude();
lat = location.getLatitude();

int radius = 10000;

ArrayList<Restaurant> list = search(lat, lng, radius);

if (list.size() == 0){
    Toast.makeText(mContext, "Echec de la requête. Ré-essayer", Toast.LENGTH_SHORT).show();
}

mListView = (RecyclerView) myView.findViewById(R.id.list);
mListView.setLayoutManager(new LinearLayoutManager(getActivity()));

adapter = new RestaurantAdapter(getContext(), list, new LatLng(lat,lng) );

mListView.setAdapter(adapter);

mListView.smoothScrollToPosition(mListView.getAdapter().getItemCount() - 1);

return myView;


ublic static ArrayList<Restaurant> search(double lat, double lng, int radius) {

HttpURLConnection conn = null;
StringBuilder jsonResults = new StringBuilder();
try {
    StringBuilder sb = new StringBuilder(PLACES_API_BASE);
    sb.append(TYPE_SEARCH);
    sb.append(OUT_JSON);
    sb.append("location=" + String.valueOf(lat) + "," + String.valueOf(lng));
    sb.append("&radius=" + String.valueOf(radius));
    sb.append("&type=meal_takeaway");
    sb.append("&key=" + API_KEY);
```

```
    URL url = new URL(sb.toString());
    conn = (HttpURLConnection) url.openConnection();
    InputStreamReader in = new InputStreamReader(conn.getInputStream());

    int read;
    char[] buff = new char[1024];
    while ((read = in.read(buff)) != -1) {
        jsonResults.append(buff,  offset: 0, read);
    }
} catch (MalformedURLException e) {
    Log.e(LOG_TAG,  msg: "Error processing Places API URL", e);
    return resultList;
} catch (IOException e) {
    Log.e(LOG_TAG,  msg: "Error connecting to Places API", e);
    return resultList;
} finally {
    if (conn != null) {
        conn.disconnect();
    }
}

try {
    // Create a JSON object hierarchy from the results
    JSONObject jsonObj = new JSONObject(jsonResults.toString());
    JSONArray predsJsonArray = jsonObj.getJSONArray( name: "results");

    // Extract the descriptions from the results
    resultList = new ArrayList<Restaurant>(predsJsonArray.length());
    for (int i = 0; i < predsJsonArray.length(); i++) {

        Restaurant place = new Restaurant(predsJsonArray.getJSONObject(i).getString( name: "name"),
                predsJsonArray.getJSONObject(i).getString( name: "vicinity"), predsJsonArray.getJSONObject(i).
                getString( name: "reference"));

        resultList.add(place);

    }
} catch (JSONException e) {
    Log.e(LOG_TAG,  msg: "Error processing JSON results", e);
}

return resultList;
```

Vue des restaurants sous forme de liste : RestaurantAdapter

```
public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {

    // Define a Place ID.
    String placeId = list.get(position).getId();

Specify the fields to return.
    List<Place.Field> placeFields = Arrays.asList(Place.Field.ID, Place.Field.RATING,

Construct a request object, passing the place ID and fields array.
    FetchPlaceRequest request = FetchPlaceRequest.newInstance(placeId, placeFields);


Initialize Places.
    Places.initialize(context, API_KEY);

Create a new Places client instance.
    PlacesClient placesClient = Places.createClient(context);

    placesClient.fetchPlace(request).addOnSuccessListener((response) -> {
        Place place = response.getPlace();

        if (place.getRating()!= null) {

            notation = place.getRating();
            list.get(position).setNote((int)Math.round(notation*100));

        }

        if (notation>=4){
            holder.star1.setVisibility(View.VISIBLE);
            holder.star2.setVisibility(View.VISIBLE);
            holder.star3.setVisibility(View.VISIBLE);
        }else if (notation>=3 && notation <4){
            holder.star1.setVisibility(View.VISIBLE);
            holder.star2.setVisibility(View.VISIBLE);
            holder.star3.setVisibility(GONE);
        } else if (notation>=2 && notation <3){
            holder.star1.setVisibility(View.VISIBLE);
            holder.star2.setVisibility(GONE);
            holder.star3.setVisibility(GONE);
        }else if (notation<2){
            holder.star1.setVisibility(GONE);
            holder.star2.setVisibility(GONE);
            holder.star3.setVisibility(GONE);
        }
```

```
    double dist = CalculationByDistance(here,place.getLatLng())/10 + CalculationByDistance(here,place.getLatLng())%1
    BigDecimal bd = new BigDecimal(dist);
    bd= bd.setScale( newScale: 2,BigDecimal.ROUND_DOWN);
    dist = bd.doubleValue();
    holder.distance.setText(dist + " km");
    list.get(position).setDistance((int)Math.round(dist*1000));


    Calendar cal = Calendar.getInstance();
ay = 1 à Saturday = 7
    int dayOfWeek = cal.get(Calendar.DAY_OF_WEEK) - 2;

    if (dayOfWeek == -1){
        dayOfWeek = 6;
    }


    if (place.getOpeningHours()!= null)
    holder.opening.setText(place.getOpeningHours().getWeekdayText().get(dayOfWeek));


    if (place.getPhotoMetadatas()!=null) {


        PhotoMetadata photoMetadata = place.getPhotoMetadatas().get(0);
        FetchPhotoRequest photoRequest = FetchPhotoRequest.builder(photoMetadata).setMaxHeight(200).build();
        placesClient.fetchPhoto(photoRequest).addOnSuccessListener((fetchPhotoResponse) -> {
            Bitmap bitmap = fetchPhotoResponse.getBitmap();
            Glide.with(context) RequestManager
                    .load(bitmap) RequestBuilder<Drawable>
                    .centerCrop() RequestBuilder<Drawable>
                    .into(holder.image);


    holder.name.setText(list.get(position).getName());
    holder.descript.setText(list.get(position).getAddress());

    int nb = 0;
    for(int j = 0; j < getAllUserWithoutMyself().size(); j++){

        String restoUser = getAllUserWithoutMyself().get(j).getResto().getId();
        int comparison3 = restoUser.compareTo(list.get(position).getId());
        if (comparison3 == 0){ nb += 1; }
    }

    holder.nbWorkmate.setText(" " + nb);

    holder.descript.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent i = new Intent(context, DetailRestaurantActivity.class);
            i.putExtra(RESTAURANT, list.get(position));
            context.startActivity(i);

        }
    });
```

Fiche détaillée d'un restaurant : RestaurantDetailActivity

```
Intent i = getIntent();
resto = (Restaurant) i.getSerializableExtra(RESTAURANT);

// Define a Place ID.
String placeId = resto.getId();

ify the fields to return.
List<Place.Field> placeFields = Arrays.asList(Place.Field.ID, Place.Field.NAME, Place.Field.ADDRESS,Place.Fi

truct a request object, passing the place ID and fields array.
FetchPlaceRequest request = FetchPlaceRequest.newInstance(placeId, placeFields);

ialize Places.
Places.initialize(getApplicationContext(), API_KEY);

te a new Places client instance.
PlacesClient placesClient = Places.createClient( context: this);

placesClient.fetchPlace(request).addOnSuccessListener((response) -> {
    Place place = response.getPlace();

    text.setText(place.getName() );
    lieuTel.setText(place.getName() + " (" + place.getRating() + "/5)"  );
    loca.setText(place.getAddress());

    if (place.getPhoneNumber()!= null) { telephone.setText(place.getPhoneNumber()); }
    if (place.getWebsiteUri()!= null) { web.setText(place.getWebsiteUri().toString()); }

    configureCall( num: "tel:" + place.getPhoneNumber());
    configureWebsite(place.getWebsiteUri());

    if (place.getPhotoMetadatas()!= null) {

        PhotoMetadata photoMetadata = place.getPhotoMetadatas().get(0);
        String attributions = photoMetadata.getAttributions();
        FetchPhotoRequest photoRequest = FetchPhotoRequest.builder(photoMetadata).setMaxHeight(200).build();
        placesClient.fetchPhoto(photoRequest).addOnSuccessListener((fetchPhotoResponse) -> {
            Bitmap bitmap = fetchPhotoResponse.getBitmap();
            Glide.with( activity: this) RequestManager
                    .load(bitmap) RequestBuilder<Drawable>
                    .centerCrop() RequestBuilder<Drawable>
                    .into(image);

        });
    }
```

```
    RecyclerView recyclerView = (RecyclerView) findViewById(R.id.listCoworkers);
    recyclerView.setLayoutManager(new LinearLayoutManager( context: this));

    ArrayList<User>  listUser = new ArrayList<~>();

    for(int j = 0; j < getAllUser().size(); j++){

        String restoUser = getAllUser().get(j).getResto().getId();
        int comparison = restoUser.compareTo(resto.getId()
        );

        if (comparison == 0){ listUser.add(getAllUser().get(j)); }
    }

    MyAdapter adapter = new MyAdapter( c: this, listUser, getUserCurrent(),  s: false);

    recyclerView.setAdapter(adapter);


}

@Override
public int getFragmentLayout() { return R.layout.activity_detail_restaurant; }

private void configureBack() {
    back = findViewById(R.id.back);
    back.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { finish(); }
    });

}

private void configureCall(String num) {
    Button call = findViewById(R.id.call);
    call.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent appel = new Intent(Intent.ACTION_DIAL, Uri.parse(num));
            startActivity(appel);
```

Fiche détaillée d'un restaurant : RestaurantDetailActivity

```
private void configureWebsite(Uri url) {
    Button website = findViewById(R.id.website);
    website.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent( Intent.ACTION_VIEW,  url  );
            startActivity(intent);

        }
    });
}

private void configureLike() {
    like = findViewById(R.id.like);
    if(getUserCurrent().getRestoLike().contains(resto.getId())){
        like.setCompoundDrawablesWithIntrinsicBounds( left: 0, R.drawable.ic_baseline_star_24px,  right:
    }

    like.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            ArrayList<String> newRestoLike = getUserCurrent().getRestoLike();

            if(getUserCurrent().getRestoLike().contains(resto.getId())){

                like.setCompoundDrawablesWithIntrinsicBounds( left: 0,R.drawable.ic_baseline_star_borde
                newRestoLike.remove(resto.getId());
                /unchecked/
                Map<String, Object> updateMap = new HashMap();
                updateMap.put(RESTOLIKE, newRestoLike);

                if (getCurrentUser() != null) {
                    UserHelper.updateRestoLike(newRestoLike, getCurrentUser().getUid()).addOnFailurel
                }

            }else {

                like.setCompoundDrawablesWithIntrinsicBounds( left: 0, R.drawable.ic_baseline_star_24p
                newRestoLike.add(resto.getId());
                /unchecked/
                Map<String, Object> updateMap = new HashMap();
                updateMap.put(RESTOLIKE, newRestoLike);

                if (getCurrentUser() != null) {
                    UserHelper.updateRestoLike(newRestoLike, getCurrentUser().getUid())
```

```
ivate void configureFab() {
  fab = findViewById(R.id.fab);

  final int[] comp = {getUserCurrent().getResto().getId().compareTo(resto.getId())};
  if (comp[0] == 0) { fab.setImageResource(R.drawable.ic_baseline_done_24px); }

  fab.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {

          comp[0] = getUserCurrent().getResto().getId().compareTo(resto.getId());
          if (comp[0] != 0) {
              /unchecked/
              Map<String, Object> updateMap = new HashMap();
              updateMap.put(ADDRESS, resto.address);
              updateMap.put(NAME, resto.name);
              updateMap.put(ID, resto.id);

              fab.setImageResource(R.drawable.ic_baseline_done_24px);
              if (getCurrentUser() != null) {
                  UserHelper.updateResto(updateMap, getCurrentUser().getUid()).addOnFailur
              }

          }else{
              @SuppressWarnings("unchecked")
              Map<String, Object> updateMap = new HashMap();
              updateMap.put(ADDRESS,  v: "?");
              updateMap.put(NAME, "Pas encore choisit");
              updateMap.put(ID,  v: "?");

              fab.setImageResource(R.drawable.ic_baseline_done_outline_24px);
              if (getCurrentUser() != null) {
                  UserHelper.updateResto(updateMap, getCurrentUser().getUid()).addOnFailur
              }
```

```
ccessListener<Void> updateUIAfterRESTRequestsCompleted(final int origin){
ew OnSuccessListener<Void>() {
rride
ic void onSuccess(Void aVoid) {
switch (origin){
    case UPDATE_RESTO: Toast.makeText( context: DetailRestaurantActivity.this, "Vous avez choisi de manger à :" + " " + resto.getName(
    case UPDATE_RESTO2: Toast.makeText( context: DetailRestaurantActivity.this, "Vous avez choisi de ne plus manger à :" + " " + resto.
    case UPDATE_RESTO_LIKE:Toast.makeText( context: DetailRestaurantActivity.this, "Vous aimez" + " " + resto.getName(), Toast.LENGTH_S
    case UPDATE_RESTO_LIKE2: Toast.makeText( context: DetailRestaurantActivity.this, "Vous n'aimez plus" + " " + resto.getName(), Toast
    default: break;
}
```

Liste des collègues : ThirdFragment

```
public class ThirdFragment extends Fragment {

    View myView;
    MyAdapter adapter;

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

    }

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        myView = inflater.inflate(R.layout.user_recycler, container,  attachToRoot: false);


        FloatingActionButton button = (FloatingActionButton ) myView.findViewById(R.id.fab);
        button.setImageResource(R.drawable.ic_baseline_message_24px);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent i = new Intent(getActivity(), ChatActivity.class);
                startActivity(i);

            }
        });

        RecyclerView  recyclerView = (RecyclerView) myView.findViewById(R.id.list);
        recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));


        adapter = new MyAdapter(getContext(), getAllUserWithoutMyself(), getUserCurrent(),  s: true);
        recyclerView.setAdapter(adapter);

        return myView;

    }

}
```

Liste des collègues : MyAdapter

```
public MyAdapter(Context c, ArrayList<User> p, User u , boolean s){
    context = c;
    profiles = p;
    currentUser = u;
    showResto = s;
}

@NonNull
@Override
public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    return new MyViewHolder(LayoutInflater.from(context).inflate(R.layout.list_item,parent, attachToRoot:
}

@Override
public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {

    if (showResto == true) {
        holder.name.setText(profiles.get(position).getUsername());
        holder.descript.setText(profiles.get(position).getResto().getName());

        //an english device can have french coworkers so...
        int comparison =  profiles.get(position).getResto().getName().compareTo("has not yet chosen.");
        int comparison2 =  profiles.get(position).getResto().getName().compareTo("Pas encore choisit");

        if (comparison != 0 && comparison2 != 0 ) {


            holder.descript.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {

                    Intent i = new Intent(context, DetailRestaurantActivity.class);
                    i.putExtra(RESTAURANT, profiles.get(position).getResto());

                    context.startActivity(i);
                }
            });

        }else{
            holder.name.setTextColor(Color.rgb( red: 160, green: 160, blue: 160));
            holder.descript.setTextColor(Color.rgb( red: 200, green: 200, blue: 200));
            holder.descript.setText("Pas encore choisit");
        }
```

```
    }else{

        if (profiles.get(position) != currentUser) {
            holder.name.setText(profiles.get(position).getUsername() + ' ' + "mange ici !")
        }else{
            holder.name.setText("Je mange ici !");
        }
    }

    Glide.with(holder.avatar.getContext()) RequestManager
            .load(profiles.get(position).getUrlPicture()) RequestBuilder<Drawable>
            .apply(RequestOptions.circleCropTransform()) RequestBuilder<Drawable>
            .into(holder.avatar);


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

class MyViewHolder extends RecyclerView.ViewHolder {
    TextView name, descript ;
    ImageView avatar;


    public MyViewHolder(@NonNull View itemView) {
        super(itemView);
        name = itemView.findViewById(R.id.list_title);
        descript =  itemView.findViewById(R.id.list_desc);
        avatar = itemView.findViewById(R.id.list_image);
    }
```

-Fonctionnalité de recherche

```
    Places.initialize(getApplicationContext(), API_KEY);

    // Initialize the AutocompleteSupportFragment.
    AutocompleteSupportFragment autocompleteFragment = (AutocompleteSupportFragment)
            getSupportFragmentManager().findFragmentById(R.id.autocomplete_fragment);

pecify the types of place data to return.
    autocompleteFragment.setPlaceFields(Arrays.asList(Place.Field.ID, Place.Field.NAME));
    autocompleteFragment.setHint("Recherche restaurants");
    autocompleteFragment.setTypeFilter(ESTABLISHMENT);

    LocationManager lm = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    if (checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && checkSelfPermission(M
        return;
    }
    Location location = lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
    double longitude = location.getLongitude();
    double latitude = location.getLatitude();

    autocompleteFragment.setLocationRestriction(RectangularBounds.newInstance(
            new LatLng( v: latitude - 0.05 , v1: longitude - 0.05),
            new LatLng( v: latitude + 0.05,  v1: longitude + 0.05)));

et up a PlaceSelectionListener to handle the response.
    autocompleteFragment.setOnPlaceSelectedListener(new PlaceSelectionListener() {
        @Override
        public void onPlaceSelected(Place place) {
            if (place.getTypes()!= null) {
                if (place.getTypes().contains(Place.Type.RESTAURANT)) {

                    Intent i = new Intent( packageContext: HungryActivity.this, DetailRestaurantActivity.class);
                    i.putExtra(RESTAURANT, new Restaurant(place.getName(), place.getAddress(), place.getId()));
                    startActivity(i);

                } else { Toast.makeText( context: HungryActivity.this, "Ce n'est pas un restaurant", Toast.LENGTH_SHORT).show(); }

            }else{
                Intent i = new Intent( packageContext: HungryActivity.this, DetailRestaurantActivity.class);
                i.putExtra(RESTAURANT, new Restaurant(place.getName(), place.getAddress(), place.getId()));
                startActivity(i);
            }
        }
        @Override
        public void onError(Status status) { }
```

-Menu (HungryActivity)

```
@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) { drawer.closeDrawer(GravityCompat.START);
    } else { super.onBackPressed(); }
}

/StatementWithEmptyBody/
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();
    FragmentManager fragmentManager = getSupportFragmentManager();

    if (id == R.id.nav_my_resto) {

        int comp = getUserCurrent().getResto().getName().compareTo("Pas encore choisit");
        if (comp== 0) {
            Toast.makeText( context: this, "Vous n'avez pas encore choisit de restaurant.", Toast.LENGTH_SHORT).show();;
        }else {
            Intent i = new Intent( packageContext: this, DetailRestaurantActivity.class);
            i.putExtra(RESTAURANT, getUserCurrent().getResto());
            startActivity(i);
        }
    } else if (id == R.id.nav_profil) {

        Intent newPage = new Intent( packageContext: this, ProfilActivity.class);
        startActivity(newPage);
        this.finish();

    } else if (id == R.id.nav_logout) {
        AuthUI.getInstance().signOut( context: this).addOnSuccessListener( activity: this, this.updateUIAfterRESTRequestsCompleted());
    }

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
```

```
private OnSuccessListener<Void> updateUIAfterRESTRequestsCompleted(){
    return new OnSuccessListener<Void>() {@Override  public void onSuccess(Void aVoid) { finish(); }};
}

private void updateUIWhenCreating(){

    if ( getCurrentUser() != null){
        //Get picture URL from Firebase
        if ( getCurrentUser().getPhotoUrl() != null) {
            Glide.with( activity: this) RequestManager
                    .load( getCurrentUser().getPhotoUrl()) RequestBuilder<Drawable>
                    .apply(RequestOptions.circleCropTransform()) RequestBuilder<Drawable>
                    .into(imageViewProfile);
        }

        //Get email & username from Firebase
        String email = TextUtils.isEmpty( getCurrentUser().getEmail()) ? "Auncune adresse email" :  getCurrentUser().getEmail();

        //Update views with data
        this.textViewEmail.setText(email );

        // 5 - Get additional data from Firestore
        UserHelper.getUser( getCurrentUser().getUid()).addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                User currentUser = documentSnapshot.toObject(User.class);
                String username = TextUtils.isEmpty(currentUser.getUsername()) ? "Aucun nom d'utilisateur" : currentUser.getUsern
               textUsername.setText(username);
            }
        });
    }
```

-Notifications : NotificationHelper

```
public void createNotification()
{
    String restoID = getUserCurrent().getResto().getId();

    ArrayList<User> listUser = new ArrayList<~>();

    for(int j = 0; j < getAllUserWithoutMyself().size(); j++){

        String restoUser = getAllUserWithoutMyself().get(j).getResto().getId();

        int comparison = restoUser.compareTo(restoID);

        if (comparison == 0){

            listUser.add(getAllUserWithoutMyself().get(j));
        }
    }

    String coworkers = "Personne d'autre n'y mange.";

    if (listUser.size()>0) {
        coworkers = "Vous mangez avec :" + " ";
    }

    for(int j = 0; j < listUser.size(); j++){

        if (j!= 0){
            coworkers += ", ";
        }
        coworkers += listUser.get(j).getUsername();

    }

    Intent intent = new Intent(mContext , LoginActivity.class);

    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

    PendingIntent resultPendingIntent = PendingIntent.getActivity(mContext,
            requestCode: 0 /* Request code */, intent,
            PendingIntent.FLAG_UPDATE_CURRENT);
```

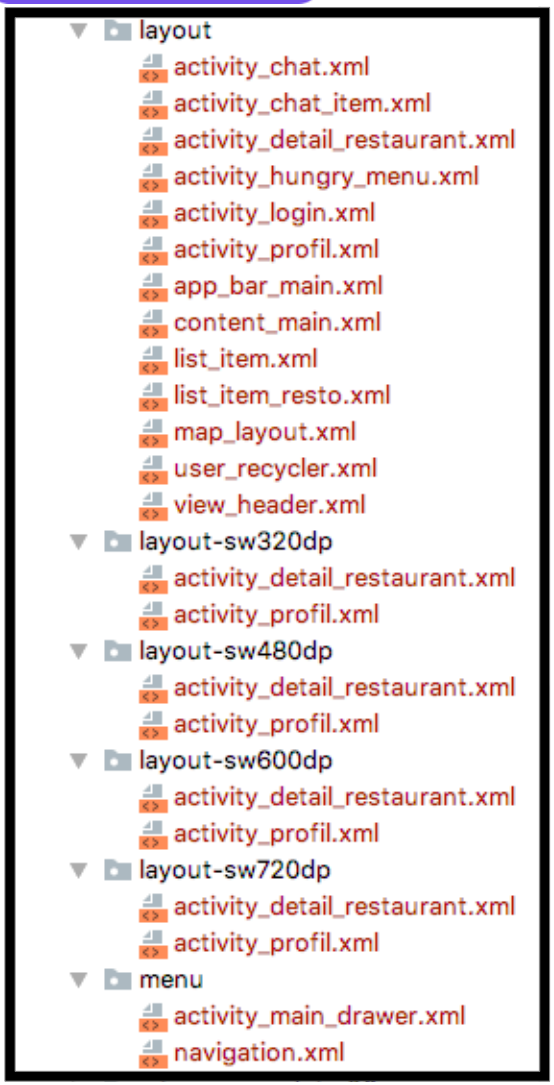
```
NotificationCompat.Builder mBuilder = new ~~NotificationCompat.Builder~~(mContext);
mBuilder.setSmallIcon(R.mipmap.ic_launcher);
mBuilder.setContentTitle("Vous avez choisi de manger à" + " " + getUserCurrent().getResto().getName())
        .setContentText(coworkers)
        .setAutoCancel(true)
        .setSound(Settings.System.DEFAULT_NOTIFICATION_URI)
        .setContentIntent(resultPendingIntent);

NotificationManager mNotificationManager = (NotificationManager) mContext.getSystemService(Context.NOTIFICATION_SERVICE);

if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O)
{
    int importance = NotificationManager.IMPORTANCE_HIGH;
    NotificationChannel notificationChannel = new NotificationChannel(NOTIFICATION_CHANNEL_ID,  name: "NOTIFICATION_CHANNEL_
    notificationChannel.enableLights(true);
    notificationChannel.setLightColor(Color.RED);
    notificationChannel.enableVibration(true);
    notificationChannel.setVibrationPattern(new long[]{100, 200, 300, 400, 500, 400, 300, 200, 400});
    assert mNotificationManager != null;
    mBuilder.setChannelId(NOTIFICATION_CHANNEL_ID);
    mNotificationManager.createNotificationChannel(notificationChannel);
}
assert mNotificationManager != null;
mNotificationManager.notify( id: 0 /* Request Code */, mBuilder.build());
```

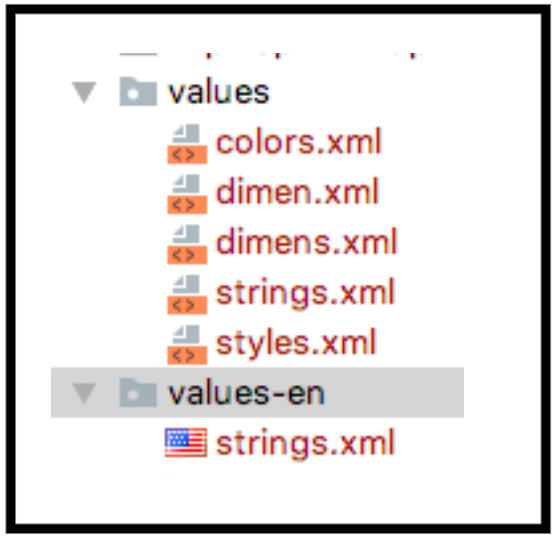
-Notifications : NotificationReceiver Et lancement dans LoginActivity

```java
public class NotificationReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {


        //if (intent.getAction().equals(Intent.ACTION_BOOT_COMPLETED)) {
        NotificationHelper notificationHelper = new NotificationHelper(context);


        if (getUserCurrent().getNotification() == true) {
            int comparison = getUserCurrent().getResto().name.compareTo("Pas encore choisit");
            if (comparison != 0) {
                notificationHelper.createNotification();
            }
        }
```

```java
public void sharedpref(){

    SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(this);
    if (!prefs.getBoolean(FIRST,  b: false)) {
        Calendar calendar = Calendar.getInstance();
        calendar.set(Calendar.HOUR_OF_DAY,12);
        calendar.set(Calendar.MINUTE,00);
        if (calendar.getTime().compareTo(new Date()) < 0) calendar.add(Calendar.DAY_OF_MONTH,  i1: 1);
        Intent intent = new Intent(getApplicationContext(), NotificationReceiver.class);
        PendingIntent pendingIntent = PendingIntent.getBroadcast( context: this, requestCode: 0,intent, flags: 0);
        AlarmManager alarmManager = (AlarmManager)getSystemService(this.ALARM_SERVICE);
        if (alarmManager != null) {
            alarmManager.setInexactRepeating(AlarmManager.RTC_WAKEUP,System.currentTimeMillis(), INTERVAL_DAY,pendingIntent);
        }
        SharedPreferences.Editor editor = prefs.edit();
        editor.putBoolean(FIRST,  b: true);
        editor.apply();
    }
}
```

-Traduction et adaptation aux différentes tailles d'écran



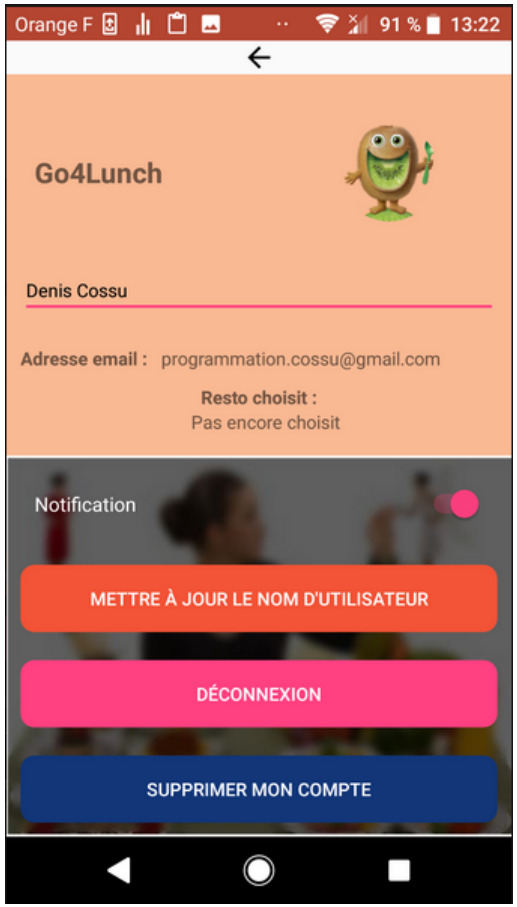



```
<string name="where_do_you_eat">Vous avez choisi de manger à </string>
<string name="nobody">Personne d\'autre n\'y mange.</string>
<string name="who_eat_with_you">Vous mangez avec : </string>
```

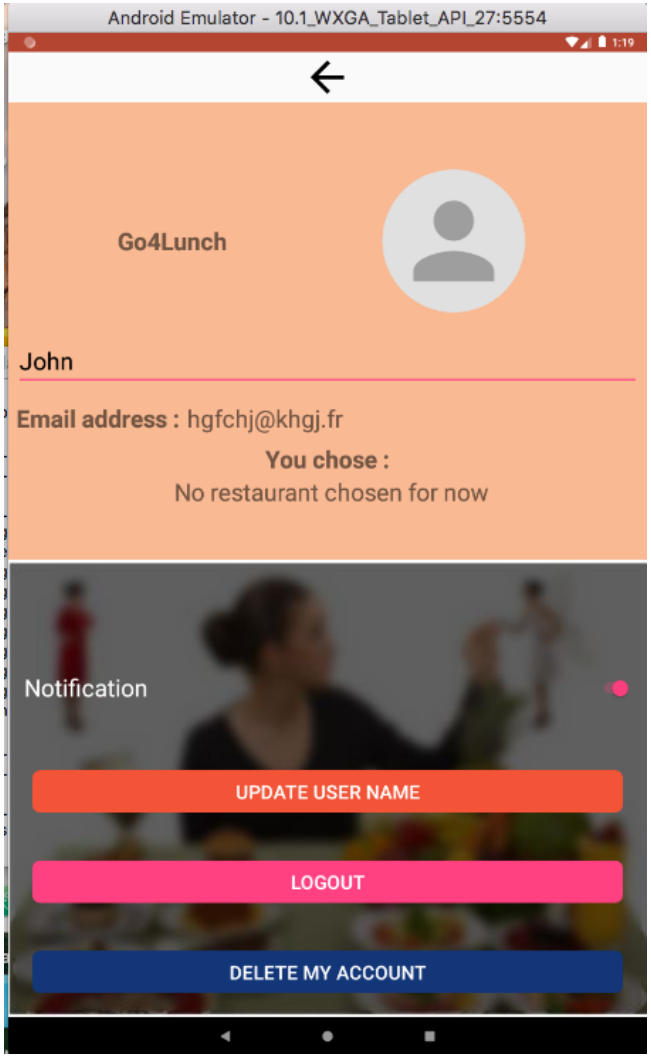
```
<string name="where_do_you_eat">You chose to go to </string>
<string name="nobody">Nobody else eat here.</string>
<string name="who_eat_with_you">You eat with : </string>
```

-Traduction et adaptation aux différentes tailles d'écran

Mon portable en français

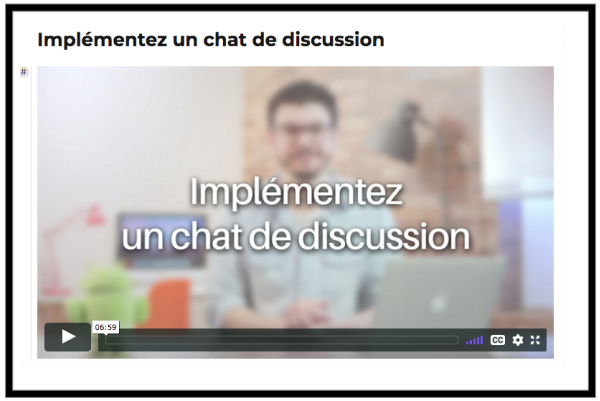


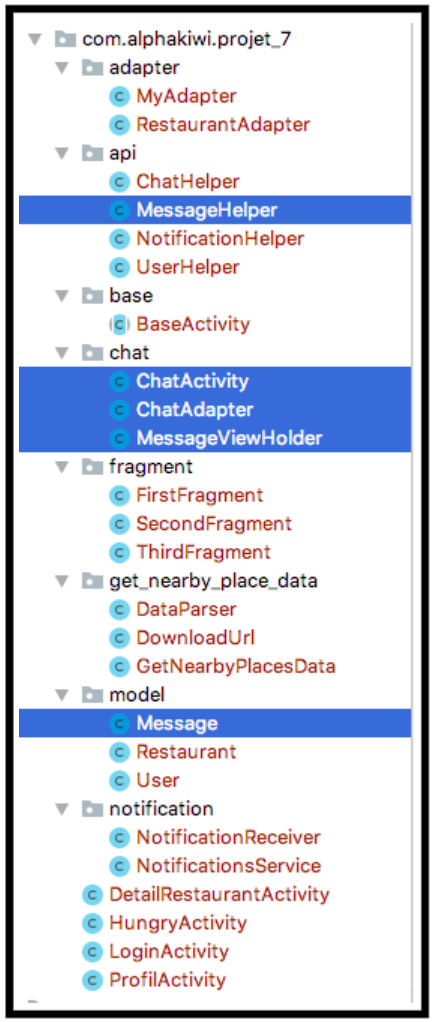
Une grande tablette en anglais



-Fonctionnalité complémentaire : inscription par E-mail + messagerie

```
private void startSignInActivity(){
    startActivityForResult(
            AuthUI.getInstance() AuthUI
                    .createSignInIntentBuilder() SignInIntentBuilder
                    .setTheme(R.style.LoginTheme) SignInIntentBuilder
                    .setAvailableProviders(
                            Arrays.asList(new AuthUI.IdpConfig.EmailBuilder().build(), //EMAIL
                                    new AuthUI.IdpConfig.GoogleBuilder().build(), //GOOGLE
                                    new AuthUI.IdpConfig.FacebookBuilder().build())) // FACEBOOK
                    .setIsSmartLockEnabled( enableCredentials: false,  enableHints: true) SignInIntentBuilder
                    .setLogo(R.drawable.logo) SignInIntentBuilder
                    .build(),
            RC_SIGN_IN);
}
```





-Fonctionnalité complémentaire : Trie Des restaurants

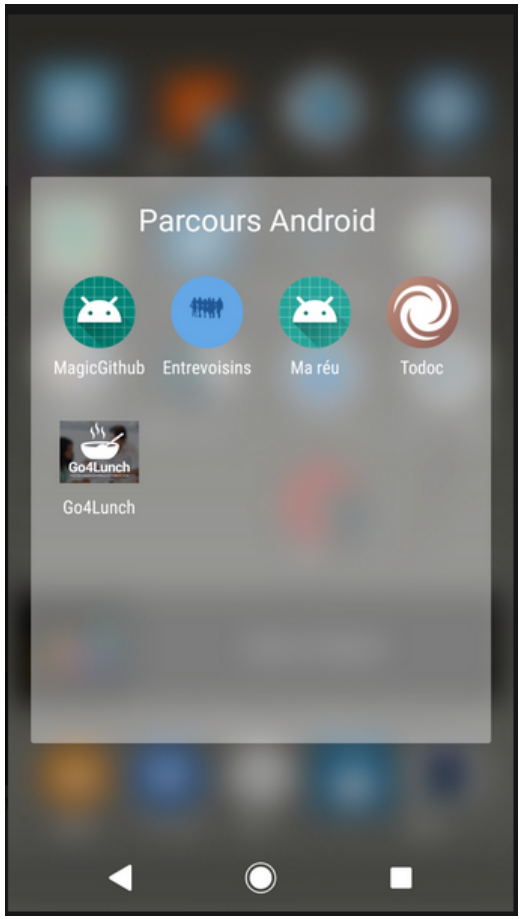
SecondFragment

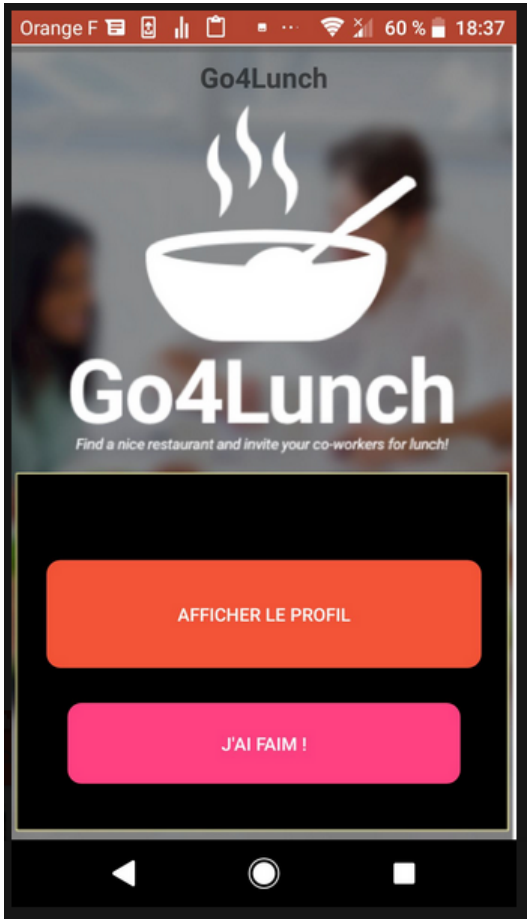
```
FloatingActionButton button = (FloatingActionButton ) myView.findViewById(R.id.fab);
button.setImageResource(R.drawable.ic_filter_list);

button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if (resultList != null) {

            switch(counter){
                case 0:
                    Collections.sort(resultList, new Restaurant.RestaurantDistanceComparator());
                    Toast.makeText(mContext, "Trie par distance", Toast.LENGTH_SHORT).show(); break;
                case 1:
                    Collections.sort(resultList, new Restaurant.RestaurantMarksComparator());
                    Toast.makeText(mContext, "Trie par note", Toast.LENGTH_SHORT).show();        break;
                case 2:
                    Collections.sort(resultList, new Restaurant.RestaurantAZComparator());
                    Toast.makeText(mContext, "Trie par ordre alphabétique", Toast.LENGTH_SHORT).show(); break;
                case 3:
                    Collections.sort(resultList, new Restaurant.RestaurantZAComparator());
                    Toast.makeText(mContext, "Trie par ordre alphabétique inversé", Toast.LENGTH_SHORT).show()

                default: //For all other cases, do this        break;
            }

        }

        adapter.updateRestaurants();
        mListView.smoothScrollToPosition(0);

        if (counter>=3){
            counter = 0;
        }else {
            counter += 1;
        }
    }
});
```

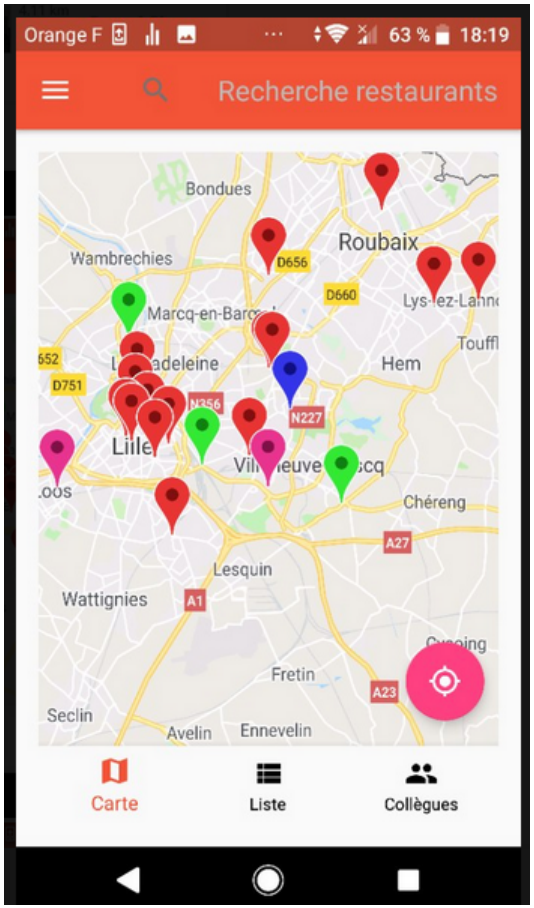
(Model) Restaurant

```
public static class RestaurantAZComparator implements Comparator< Restaurant> {
    @Override
    public int compare( Restaurant left,  Restaurant right) {
        return left.name.compareTo(right.name);
    }
}


public static class  RestaurantZAComparator implements Comparator< Restaurant> {
    @Override
    public int compare( Restaurant left,  Restaurant right) {
        return right.name.compareTo(left.name);
    }
}

public static class RestaurantMarksComparator implements Comparator<Restaurant> {
    @Override
    public int compare(Restaurant left, Restaurant right) { return  (right.note- left.note); }
}

public static class RestaurantDistanceComparator implements Comparator<Restaurant> {
    @Override
    public int compare(Restaurant left, Restaurant right) {
        return  (left.distance- right.distance);
    }
}
```

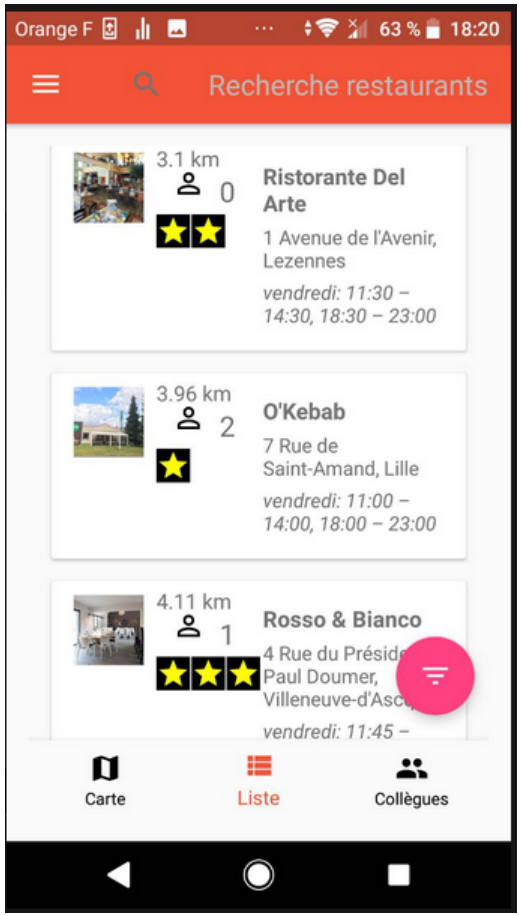
Fin

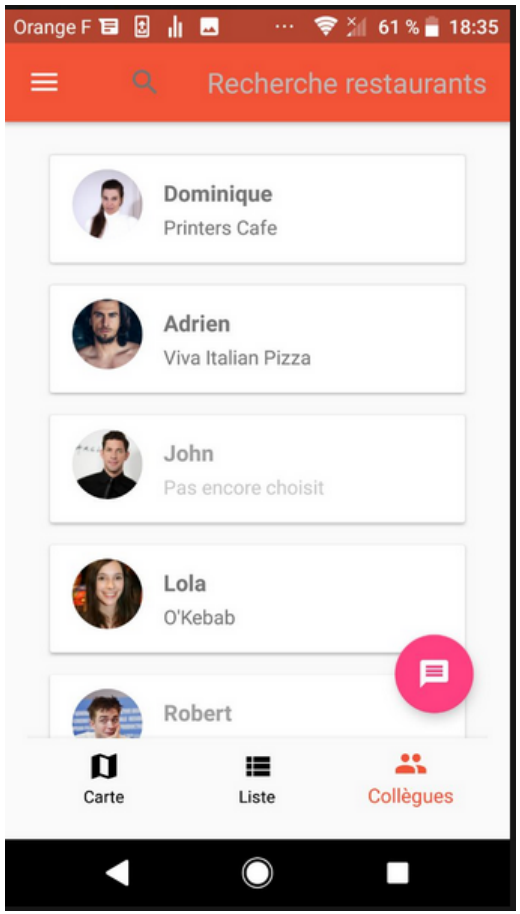

Parcours Android
MagicGithub
Entrevoisins
Ma réu
Todoc
Go4Lunch
Go4Lunch


Orange F
60 %
18:37
Go4Lunch
Go4Lunch
Find a nice restaurant and invite your co-workers for lunch!
AFFICHER LE PROFIL
J'AI FAIM !


Orange F
63 %
18:19
Recherche restaurants
Bondues
Roubaix
Wambrechies
Marcq-en-Barœul
Hem
Lille
Chéreng
Lesquin
Wattignies
Fretin
Seclin
Avelin
Ennevelin
Carte
Liste
Collègues

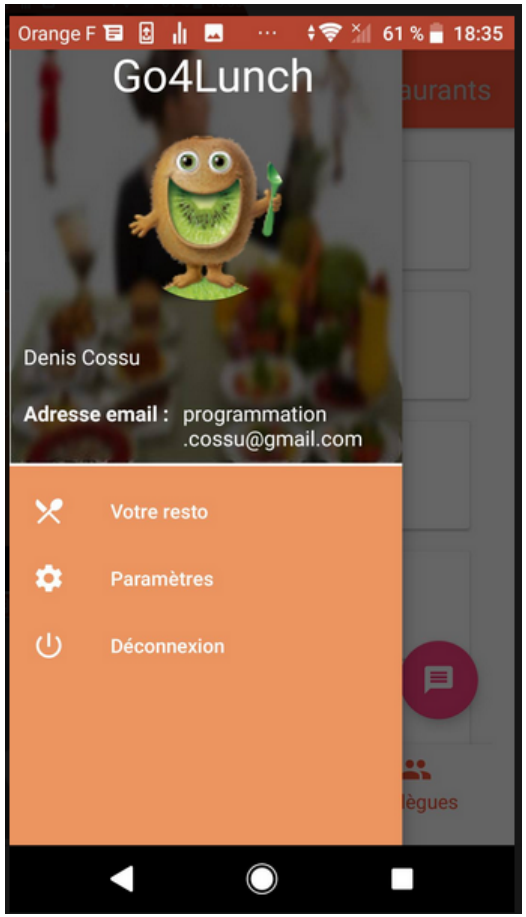
Fin

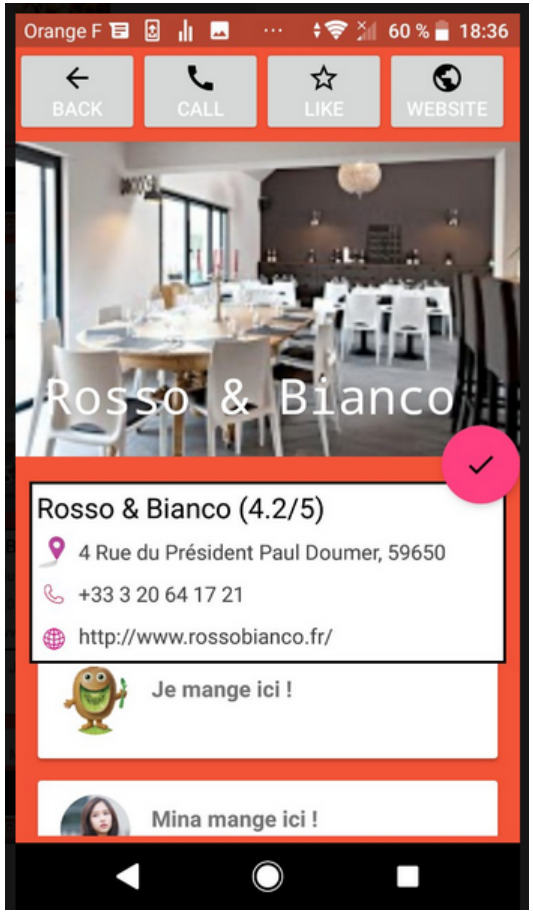

Orange F
63 % 18:20
Recherche restaurants
3.1 km
0
Ristorante Del Arte
1 Avenue de l'Avenir, Lezennes
vendredi: 11:30 – 14:30, 18:30 – 23:00
3.96 km
2
O'Kebab
7 Rue de Saint-Amand, Lille
vendredi: 11:00 – 14:00, 18:00 – 23:00
4.11 km
1
Rosso & Bianco
4 Rue du Préside
Paul Doumer,
Villeneuve-d'Asc
vendredi: 11:45 –
Carte
Liste
Collègues

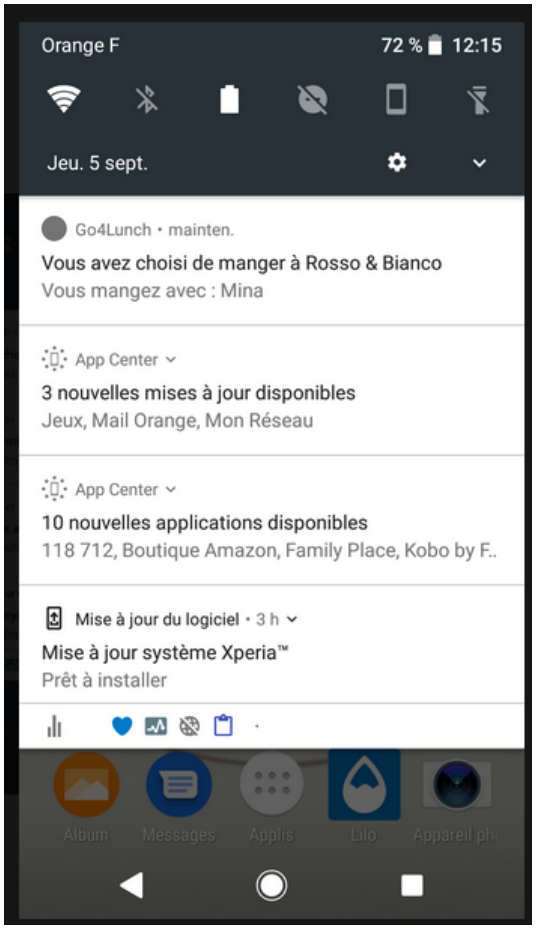

Orange F
61 % 18:35
Recherche restaurants
Dominique
Printers Cafe
Adrien
Viva Italian Pizza
John
Pas encore choisit
Lola
O'Kebab
Robert
Carte
Liste
Collègues


Orange F
61 % 18:34
14:10
Vous allez manger où ?
17:15
Je sais pas trop...
17:33
J'ai envie de pizza... ^^
17:33
On va à Pizza Napoli ?
18:34
Composer votre message
ENVOYER

Fin


Orange F
61 % 18:35
Go4Lunch
Denis Cossu
Adresse email : programmation .cossu@gmail.com
Votre resto
Paramètres
Déconnexion


Orange F
60 % 18:36
BACK
CALL
LIKE
WEBSITE
Rosso & Bianco
Rosso & Bianco (4.2/5)
4 Rue du Président Paul Doumer, 59650
+33 3 20 64 17 21
http://www.rossobianco.fr/
Je mange ici !
Mina mange ici !


Orange F
72 % 12:15
Jeu. 5 sept.
Go4Lunch • mainten.
Vous avez choisi de manger à Rosso & Bianco
Vous mangez avec : Mina
App Center
3 nouvelles mises à jour disponibles
Jeux, Mail Orange, Mon Réseau
App Center
10 nouvelles applications disponibles
118 712, Boutique Amazon, Family Place, Kobo by F..
Mise à jour du logiciel • 3 h
Mise à jour système Xperia™
Prêt à installer