



Projet 3 :

-Développez une nouvelle fonctionnalité pour l'application Entrevoisins et testez-la



Entrevoisins

Cossu Denis -Projet 3-

Parcours Développeur Android

(model) Neighbour : Avant

```
public class Neighbour {  
  
    /** Identifier */  
    private Integer id;  
  
    /** Full name */  
    private String name;  
  
    /** Avatar */  
    private String avatarUrl;  
  
    /**  
     * Constructor  
     * @param id  
     * @param name  
     * @param avatarUrl  
     */  
    public Neighbour(Integer id, String name, String avatarUrl) {  
        this.id = id;  
        this.name = name;  
        this.avatarUrl = avatarUrl;  
    }  
  
    public Integer getId() { return id; }  
  
    public void setId(Integer id) { this.id = id; }  
  
    public String getName() { return name; }  
  
    public void setName(String name) { this.name = name; }  
  
    public String getAvatarUrl() { return avatarUrl; }  
  
    public void setAvatarUrl(String avatarUrl) { this.avatarUrl = avatarUrl; }  
  
    @Override  
    public boolean equals(Object o) {  
        if (this == o) return true;  
        if (o == null || getClass() != o.getClass()) return false;  
        Neighbour neighbour = (Neighbour) o;  
        return Objects.equals(id, neighbour.id);  
    }  
  
    @Override  
    public int hashCode() { return Objects.hash(id); }  
}
```

Cossu Denis -Projet 3-

Parcours Développeur Android

(model) Neighbour : Après

```
public class Neighbour {  
    /** Identifier */  
    private Integer id;  
  
    /** Full name */  
    private String name;  
  
    /** Avatar */  
    private String avatarUrl;  
  
    private String localisation;  
  
    private String tel;  
  
    private String description;  
    /**  
     * Constructor  
     * @param id  
     * @param name  
     * @param avatarUrl  
     */  
    public Neighbour(Integer id, String name, String avatarUrl, String localisation, String tel, String description) {  
        this.id = id;  
        this.name = name;  
        this.avatarUrl = avatarUrl;  
        this.localisation = localisation;  
        this.tel = tel;  
        this.description = description;  
    }  
  
    public Integer getId() { return id; }  
  
    public void setId(Integer id) { this.id = id; }  
  
    public String getName() { return name; }  
  
    public void setName(String name) { this.name = name; }  
  
    public String getAvatarUrl() { return avatarUrl; }  
  
    public void setAvatarUrl(String avatarUrl) { this.avatarUrl = avatarUrl; }  
  
    public String getLocalisation() { return localisation; }  
}
```

```
public String getTel() { return tel; }  
  
public String getDescription() { return description; }  
  
@Override  
public boolean equals(Object o) {  
    if (this == o) return true;  
    if (o == null || getClass() != o.getClass()) return false;  
    Neighbour neighbour = (Neighbour) o;  
    return Objects.equals(id, neighbour.id);  
}  
  
@Override  
public int hashCode() { return Objects.hash(id); }  
}
```

Parcours Développeur Android

Neighbour ApiService : Avant/Après

```
public interface Neighbour ApiService {  
  
    /**  
     * Get all my Neighbours  
     * @return {@link List}  
     */  
    List<Neighbour> getNeighbours();  
  
    /**  
     * Deletes a neighbour  
     * @param neighbour  
     */  
    void deleteNeighbour(Neighbour neighbour);  
}
```

```
/**  
 * Neighbour API client  
 */  
public interface Neighbour ApiService {  
  
    /**  
     * Get all my Neighbours  
     * @return {@link List}  
     */  
    List<Neighbour> getNeighbours();  
  
    /**  
     * Deletes a neighbour  
     * @param neighbour  
     */  
    void deleteNeighbour(Neighbour neighbour);  
  
    /**  
     * Get all my NeighboursStar  
     * @return {@link List}  
     */  
    List<Neighbour> getNeighboursStar();  
  
    /**  
     * Deletes a neighbourStar  
     * @param neighbour  
     */  
    void deleteNeighbourStar(Neighbour neighbour);  
  
    /**  
     * Add a neighbourStar  
     * @param neighbour  
     */  
    void addNeighbourStar(Neighbour neighbour);  
  
    /**  
     * Add a neighbour  
     * @param neighbour  
     */  
    void addNeighbour(Neighbour neighbour);  
}
```

Parcours Développeur Android

DummyNeighbourGenerator : Avant

```
public abstract class DummyNeighbourGenerator {

    public static List<Neighbour> DUMMY_NEIGHBOURS = Arrays.asList(
        new Neighbour( id: 1, name: "Caroline", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704d"),
        new Neighbour( id: 2, name: "Jack", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704e"),
        new Neighbour( id: 3, name: "Chloé", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704f"),
        new Neighbour( id: 4, name: "Vincent", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704a"),
        new Neighbour( id: 5, name: "Elodie", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704b"),
        new Neighbour( id: 6, name: "Sylvain", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704c"),
        new Neighbour( id: 7, name: "Laetitia", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026703d"),
        new Neighbour( id: 8, name: "Dan", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026703b"),
        new Neighbour( id: 9, name: "Joseph", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704d"),
        new Neighbour( id: 10, name: "Emma", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026706d"),
        new Neighbour( id: 11, name: "Patrick", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026702d"),
        new Neighbour( id: 12, name: "Ludovic", avatarUrl: "http://i.pravatar.cc/150?u=a042581f3e39026702d")
    );

    static List<Neighbour> generateNeighbours() { return new ArrayList<>(DUMMY_NEIGHBOURS); }
}
```

Parcours Développeur Android

DummyNeighbourGenerator : Après

```
ublic abstract class DummyNeighbourGenerator {

    public static List<Neighbour> DUMMY_NEIGHBOURS = Arrays.asList(
        new Neighbour( id: 2, name: "Jack", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704e", localisation: "Aurice à 0km", tel: "+33 6 45 62 90 77", description: "Bonsoir ! Fraichement sorti de la prison de la Santé, je suis à la recherche d'un travail dans le secteur de l'assassinat en série. Je suis également à la recherche d'une personne pour me faire une petite缠", ),
        new Neighbour( id: 3, name: "Chloé", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704f", localisation: "Le Ley à 1km", tel: "+33 6 56 21 98 31", description: "Salut tout le monde, je suis une jeune femme de 22 ans et je recherche un travail dans le secteur de l'assassinat en série. Je suis également à la recherche d'une personne pour me faire une petite缠", ),
        new Neighbour( id: 4, name: "Vincent", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704a", localisation: "Lamothe à 2km", tel: "+33 7 35 36 23 90", description: "Bonjour, photo de profil de Vincent, un jeune homme de 25 ans qui recherche un travail dans le secteur de l'assassinat en série. Il est également à la recherche d'une personne pour me faire une petite缠", ),
        new Neighbour( id: 5, name: "Elodie", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704b", localisation: "Cauna à 2km", tel: "+33 6 45 62 38 90", description: "Elodie 26 ans, mère de deux enfants, recherche un travail dans le secteur de l'assassinat en série. Elle est également à la recherche d'une personne pour me faire une petite缠", ),
        new Neighbour( id: 6, name: "Sylvain", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704c", localisation: "Bas-Mauco à 2km", tel: "+33 6 12 21 34 23", description: "Gentil Barbou, 30 ans, cherche un travail dans le secteur de l'assassinat en série. Il est également à la recherche d'une personne pour me faire une petite缠", ),
        new Neighbour( id: 7, name: "Laetitia", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026703d", localisation: "Haut-Mauco à 2km", tel: "+33 6 94 53 48 63", description: "Je m'ennuie dans ma ville natale, je recherche un travail dans le secteur de l'assassinat en série. Je suis également à la recherche d'une personne pour me faire une petite缠", ),
        new Neighbour( id: 8, name: "Dan", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026703b", localisation: "Campagne à 3km ", tel: "+33 6 21 95 46 32", description: "Beau mec bien dans sa peau, recherche un travail dans le secteur de l'assassinat en série. Il est également à la recherche d'une personne pour me faire une petite缠", ),
        new Neighbour( id: 9, name: "Joseph", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704d", localisation: "Saint-Perdon à 4km", tel: "+33 6 97 86 70 86", description: "Bonjour je suis Joseph, 28 ans, je recherche un travail dans le secteur de l'assassinat en série. Je suis également à la recherche d'une personne pour me faire une petite缠", ),
        new Neighbour( id: 10, name: "Emma", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026706d", localisation: "Banquet à 4km ", tel: "+33 7 26 43 10 02", description: "Jeune directrice de l'école primaire de la ville, recherche un travail dans le secteur de l'assassinat en série. Je suis également à la recherche d'une personne pour me faire une petite缠", ),
        new Neighbour( id: 11, name: "Patrick", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026702d", localisation: "Saint-Sever à 4km ", tel: "+33 7 56 10 83 67", description: "Bon jour à tous, je suis Patrick, 32 ans, je recherche un travail dans le secteur de l'assassinat en série. Je suis également à la recherche d'une personne pour me faire une petite缠", ),
        new Neighbour( id: 12, name: "Ludovic", avatarUrl: "http://i.pravatar.cc/150?u=a042581f3e39026702d", localisation: "Bretagne de Marsan à 5km", tel: "+33 6 23 56 03 77", description: "Bo", ),
        new Neighbour( id: 1, name: "Caroline", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704d", localisation: "Saint pierre du mont à 5km", tel: "+33 6 86 57 90 14", description: "Bo"
    );

    public static List<Neighbour> DUMMY_NEIGHBOURS_STAR = Arrays.asList(
        new Neighbour( id: 1, name: "Caroline", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026704d", localisation: "Saint pierre du mont à 5km", tel: "+33 6 86 57 90 14", description: "Bo",
        new Neighbour( id: 11, name: "Patrick", avatarUrl: "http://i.pravatar.cc/150?u=a042581f4e29026702d", localisation: "Saint-Sever à 4km ", tel: "+33 7 56 10 83 67", description: "Bon jour à tous, je suis Patrick, 32 ans, je recherche un travail dans le secteur de l'assassinat en série. Je suis également à la recherche d'une personne pour me faire une petite缠"
    );

    static List<Neighbour> generateNeighbours() { return new ArrayList<>(DUMMY_NEIGHBOURS); }
    static List<Neighbour> generateNeighboursStar() {
        return new ArrayList<>(DUMMY_NEIGHBOURS_STAR);
    }
}
```

Parcours Développeur Android



DummyNeighbour ApiService : Avant/Après

```
public class DummyNeighbour ApiService implements Neighbour ApiService {  
  
    private List<Neighbour> neighbours = DummyNeighbourGenerator.generateNeighbours();  
  
    /**  
     * {@inheritDoc}  
     */  
    @Override  
    public List<Neighbour> getNeighbours() { return neighbours; }  
  
    /**  
     * {@inheritDoc}  
     */  
    @Override  
    public void deleteNeighbour(Neighbour neighbour) { neighbours.remove(neighbour); }  
}
```

```
public class DummyNeighbour ApiService implements Neighbour ApiService {  
  
    private List<Neighbour> neighbours = DummyNeighbourGenerator.generateNeighbours();  
    private List<Neighbour> neighboursStar = DummyNeighbourGenerator.generateNeighboursStar();  
  
    /**  
     * {@inheritDoc}  
     */  
    @Override  
    public List<Neighbour> getNeighbours() { return neighbours; }  
  
    /**  
     * {@inheritDoc}  
     */  
    @Override  
    public void deleteNeighbour(Neighbour neighbour) { neighbours.remove(neighbour); }  
  
    /**  
     * {@inheritDoc}  
     */  
    @Override  
    public List<Neighbour> getNeighboursStar() { return neighboursStar; }  
  
    /**  
     * {@inheritDoc}  
     */  
    @Override  
    public void deleteNeighbourStar(Neighbour neighbour) { neighboursStar.remove(neighbour); }  
  
    @Override  
    public void addNeighbourStar(Neighbour neighbour) { neighboursStar.add(neighbour); }  
  
    @Override  
    public void addNeighbour(Neighbour neighbour) { neighbours.add(neighbour); }  
}
```

Cossu Denis -Projet 3-

Parcours Développeur Android



DI et DeleteNeighbourEvent : pas de changement.

```
/**  
 * Dependency injector to get instance of services  
 */  
  
public class DI {  
  
    private static Neighbour ApiService service = new DummyNeighbour ApiService();  
  
    /**  
     * Get an instance on {@link Neighbour ApiService}  
     * @return  
     */  
    public static Neighbour ApiService getNeighbour ApiService() { return service; }  
  
    /**  
     * Get always a new instance on {@link Neighbour ApiService}. Useful for tests, so  
     * @return  
     */  
    public static Neighbour ApiService getNewInstance ApiService() {  
        return new DummyNeighbour ApiService();  
    }  
}
```

```
/**  
 * Event fired when a user deletes a Neighbour  
 */  
  
public class DeleteNeighbourEvent {  
  
    /**  
     * Neighbour to delete  
     */  
    public Neighbour neighbour;  
  
    /**  
     * Constructor.  
     * @param neighbour  
     */  
    public DeleteNeighbourEvent(Neighbour neighbour) { this.neighbour = neighbour; }  
}
```

Cossu Denis -Projet 3-

Parcours Développeur Android



Ajout de 4 events :

- ActualiseEvent
- AddNeighbourEvent
- AddNeighbourStarEvent
- DeleteNeighbourStarEvent

```
public class AddNeighbourEvent {  
  
    public Neighbour neighbour;  
  
    /**  
     * Constructor.  
     * @param neighbour  
     */  
    public AddNeighbourEvent(Neighbour neighbour) {  
  
        this.neighbour = neighbour;  
    }  
  
}
```

```
public class AddNeighbourStarEvent {  
  
    public Neighbour neighbour;  
  
    /**  
     * Constructor.  
     * @param neighbour  
     */  
    public AddNeighbourStarEvent(Neighbour neighbour) {  
  
        this.neighbour = neighbour;  
    }  
  
}
```

```
public class ActualiseEvent {  
  
    public ActualiseEvent() {  
  
    }  
}
```

```
public class DeleteNeighbourStarEvent {  
  
    /**  
     * Neighbour to delete  
     */  
    public Neighbour neighbour;  
  
    /**  
     * Constructor.  
     * @param neighbour  
     */  
    public DeleteNeighbourStarEvent(Neighbour neighbour) {  
  
        this.neighbour = neighbour;  
    }  
  
}
```

Cossu Denis -Projet 3-



Parcours Développeur Android

ListNeighbourActivity : Avant

```
public class ListNeighbourActivity extends AppCompatActivity {

    // UI Components
    @BindView(R.id.tabs)
    TabLayout mTabLayout;
    @BindView(R.id.toolbar)
    Toolbar mToolbar;
    @BindView(R.id.container)
    ViewPager mViewPager;

    ListNeighbourPagerAdapter mPagerAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list_neighbour);
        ButterKnife.bind( target: this);

        setSupportActionBar(mToolbar);
        mPagerAdapter = new ListNeighbourPagerAdapter(getSupportFragmentManager());
        mViewPager.setAdapter(mPagerAdapter);
        mViewPager.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(mTabLayout));
        mTabLayout.addOnTabSelectedListener(new TabLayout.ViewPagerOnTabSelectedListener(mViewPager));
    }
}
```

Parcours Développeur Android

ListNeighbourActivity : Après

```
public class ListNeighbourActivity extends AppCompatActivity {

    // UI Components
    @BindView(R.id.tabs)
    TabLayout mTabLayout;
    @BindView(R.id.toolbar)
    Toolbar mToolbar;
    @BindView(R.id.container)
    ViewPager mViewPager;

    ListNeighbourPagerAdapter mPagerAdapter;
    FloatingActionButton fab;
    String avatar ;
    int id ;
    String login;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list_neighbour);
        ButterKnife.bind( target: this);

        setSupportActionBar(mToolbar);
        mPagerAdapter = new ListNeighbourPagerAdapter(getSupportFragmentManager());
        mViewPager.setAdapter(mPagerAdapter);
        mViewPager.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(mTabLayout));
        mTabLayout.addOnTabSelectedListener(new TabLayout.ViewPagerOnTabSelectedListener(mViewPager));

        id = 13;
        configureFab();
    }

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

        fab.setOnClickListener(view -> {
            login = "Denis";
            id += 1;
            //avatar = "https://api.adorable.io/AVATARS/512/" + id + ".png";
            avatar = "http://i.pravatar.cc/150?u=a042581f4e290267"+ id +"e";
            String numero = "+33 7 50 25 51 " + id;
        });
    }
}
```

```
fab.setOnClickListener(view -> {
    login = "Denis";
    id += 1;
    //avatar = "https://api.adorable.io/AVATARS/512/" + id + ".png";
    avatar = "http://i.pravatar.cc/150?u=a042581f4e290267"+ id +"e";
    String numero = "+33 7 50 25 51 " + id;

    new AlertDialog.Builder(view.getContext())
        .setView(R.layout.dialog_choix_sujet)
        .setPositiveButton( text: "Valider", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                EditText etSujet = (EditText) ((AlertDialog) dialog).findViewById(R.id.sujet);
                if (etSujet.getText().toString().length()>0) {
                    login = etSujet.getText().toString();
                }
                String description = "Bonjour je suis " + login + ". Venez parler !";
                EventBus.getDefault().post(new AddNeighbourEvent(new Neighbour(id, login,
                    avatar, localisation: "Localisation indisponible", numero, description)));
            }
        })
        .setNegativeButton( text: "Annuler", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        })
        .show();
});

@Override
protected void onStart() {
    super.onStart();
    EventBus.getDefault().post(new ActualiseEvent());
}
```

Parcours Développeur Android

ListNeighbourPagerAdapter : Avant/Après

```
public class ListNeighbourPagerAdapter extends FragmentPagerAdapter {  
  
    public ListNeighbourPagerAdapter(FragmentManager fm) { super(fm); }  
  
    /**  
     * getItem is called to instantiate the fragment for the given page.  
     * @param position  
     * @return  
     */  
    @Override  
    public Fragment getItem(int position) { return NeighbourFragment.newInstance(); }  
  
    /**  
     * get the number of pages  
     * @return  
     */  
    @Override  
    public int getCount() { return 1; }  
}
```

```
public class ListNeighbourPagerAdapter extends FragmentPagerAdapter {  
  
    public ListNeighbourPagerAdapter(FragmentManager fm) { super(fm); }  
  
    /**  
     * getItem is called to instantiate the fragment for the given page.  
     * @param position  
     * @return  
     */  
    @Override  
    public Fragment getItem(int position) {  
  
        if (position != 0) {  
            return NeighbourFragmentStar.newInstance();  
        } else {  
            return NeighbourFragment.newInstance();  
        }  
    }  
  
    /**  
     * get the number of pages  
     * @return  
     */  
    @Override  
    public int getCount() { return 2; }  
}
```

Parcours Développeur Android

NeighbourFragment : Avant

```
public class NeighbourFragment extends Fragment {

    private Neighbour ApiService mApiService;
    private List<Neighbour> mNeighbours;
    private RecyclerView m.RecyclerView;

    /**
     * Create and return a new instance
     * @return {@link NeighbourFragment}
     */
    public static NeighbourFragment newInstance() {
        NeighbourFragment fragment = new NeighbourFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mApiService = DI.getNeighbour ApiService();
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_neighbour_list, container, false);
        Context context = view.getContext();
        m.RecyclerView = (RecyclerView) view;
        m.RecyclerView.setLayoutManager(new LinearLayoutManager(context));
        m.RecyclerView.addItemDecoration(new DividerItemDecoration(getContext(), DividerItemDecoration.VERTICAL));
        initList();
        return view;
    }
}
```

```
/*
 * Init the List of neighbours
 */
private void initList() {
    mNeighbours = mApiService.getNeighbours();
    m.RecyclerView.setAdapter(new MyNeighbourRecyclerViewAdapter(mNeighbours));
}

@Override
public void onStart() {
    super.onStart();
    EventBus.getDefault().register(subscriber: this);
}

@Override
public void onStop() {
    super.onStop();
    EventBus.getDefault().unregister(subscriber: this);
}

/*
 * Fired if the user clicks on a delete button
 * @param event
 */
@Subscribe
public void onDeleteNeighbour(DeleteNeighbourEvent event) {
    mApiService.deleteNeighbour(event.neighbour);
    initList();
}
```

Parcours Développeur Android

NeighbourFragment : Après

```
public class NeighbourFragment extends Fragment {

    private Neighbour ApiService;
    private List<Neighbour> mNeighbours;
    private List<Neighbour> mNeighboursStar;
    private RecyclerView mRecyclerView;

    /**
     * Create and return a new instance
     * @return {@link NeighbourFragment}
     */
    public static NeighbourFragment newInstance() {
        NeighbourFragment fragment = new NeighbourFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ApiService = DI.getNeighbour ApiService();
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_neighbour, container, false);
        Context context = view.getContext();
        mRecyclerView = (RecyclerView) view;
        mRecyclerView.setLayoutManager(new LinearLayoutManager(context));
        mRecyclerView.addItemDecoration(new DividerItemDecoration(getContext(), DividerItemDecoration.VERTICAL));
        initList();
        return view;
    }

    /**
     * Init the List of neighbours
     */
    private void initList() {
        mNeighbours = ApiService.getNeighbours();
        mNeighboursStar = ApiService.getNeighboursStar();
        mRecyclerView.setAdapter(new MyNeighbourRecyclerViewAdapter(mNeighbours, mNeighboursStar, star: true));
    }
}
```

```
@Override
public void onStart() {
    super.onStart();
    EventBus.getDefault().register( subscriber: this);
}

@Override
public void onStop() {
    super.onStop();
    EventBus.getDefault().unregister( subscriber: this);
}

/**
 * Fired if the user clicks on a delete button
 * @param event
 */

@Subscribe
public void onDeleteNeighbour(DeleteNeighbourEvent event) {
    ApiService.deleteNeighbour(event.neighbour);
    initList();
}

@Subscribe
public void onDeleteNeighbourStar(DeleteNeighbourEvent event) {
    ApiService.deleteNeighbourStar(event.neighbour);
    initList();
}

@Subscribe
public void onAddNeighbourEvent(AddNeighbourEvent event) {
    ApiService.addNeighbour(event.neighbour);
    initList();
}

@Subscribe
public void onActualise(ActualiseEvent event) { initList(); }
```

Parcours Développeur Android

On ajoute NeighbourFragmenStart

```
public class NeighbourFragmentStar extends Fragment {

    private Neighbour ApiService mApiService;
    private List<Neighbour> mNeighbours;
    private RecyclerView mRecycler View;

    /**
     * Create and return a new instance
     * @return {@link NeighbourFragmentStar}
     */
    public static NeighbourFragmentStar newInstance() {
        NeighbourFragmentStar fragment = new NeighbourFragmentStar();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mApiService = DI.getNeighbour ApiService();
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_neighbour_list_star
            , container, false);
        Context context = view.getContext();
        mRecycler View = (RecyclerView) view;
        mRecycler View.setLayoutManager(new LinearLayoutManager(context));
        mRecycler View.addItemDecoration(new DividerItemDecoration(getContext()
            , DividerItemDecoration.VERTICAL));
        initList();
        EventBus.getDefault().register( subscriber: this);
        return view;
    }
}
```

```
/*
 * Init the List of neighbours
 */
private void initList() {
    mNeighbours = mApiService.getNeighboursStar();
    mRecycler View.setAdapter(new MyNeighbourRecyclerViewAdapter(mNeighbours, mNeighbours, star: false));
}

@Override
public void onDestroy() { // changer onStop par onDestroy
    super.onDestroy();
    EventBus.getDefault().unregister( subscriber: this);
}

/**
 * Fired if the user clicks on a delete button
 * @param event
 */

@Subscribe
public void onDeleteNeighbour(DeleteNeighbourStarEvent event) {
    mApiService.deleteNeighbourStar(event.neighbour);
    initList();
}

@Subscribe
public void onAddNeighbourStarEvent(AddNeighbourStarEvent event) {
    mApiService.addNeighbourStar(event.neighbour);
    initList();
}

@Subscribe
public void onActualise(ActualiseEvent event) {
    initList();
}
```

Parcours Développeur Android



MyNeighbourRecyclerViewAdapter : Avant

```
public class MyNeighbourRecyclerViewAdapter extends RecyclerView.Adapter<MyNeighbourRecyclerViewAdapter.ViewHolder> {

    private final List<Neighbour> mNeighbours;

    public MyNeighbourRecyclerViewAdapter(List<Neighbour> items) { mNeighbours = items; }

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

    @Override
    public void onBindViewHolder(final ViewHolder holder, int position) {
        Neighbour neighbour = mNeighbours.get(position);
        holder.mNeighbourName.setText(neighbour.getName());
        Glide.with(holder.mNeighbourAvatar.getContext())
            .load(neighbour.getAvatarUrl())
            .apply(RequestOptions.circleCropTransform())
            .into(holder.mNeighbourAvatar);

        holder.mDeleteButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EventBus.getDefault().post(new DeleteNeighbourEvent(neighbour));
            }
        });
    }

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

    public class ViewHolder extends RecyclerView.ViewHolder {
        @BindView(R.id.item_list_avatar)
        public ImageView mNeighbourAvatar;
        @BindView(R.id.item_list_name)
        public TextView mNeighbourName;
        @BindView(R.id.item_list_delete_button)
        public ImageButton mDeleteButton;

        public ViewHolder(View view) {
            super(view);
            ButterKnife.bind( target: this, view);
        }
    }
}
```

Cossu Denis -Projet 3-

Parcours Développeur Android

MyNeighbourRecyclerViewAdapter : Après

```
public class MyNeighbourRecyclerViewAdapter extends RecyclerView.Adapter<MyNeighbourRecyclerViewAdapter.ViewHolder> {

    private final List<Neighbour> mNeighbours;
    private final List<Neighbour> mNeighboursStar;
    private final Boolean mStar;

    public MyNeighbourRecyclerViewAdapter(List<Neighbour> items,
                                         List<Neighbour> itemsStar, Boolean star) {
        mNeighbours = items;
        mNeighboursStar = itemsStar;
        mStar = star;
    }

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

    @Override
    public void onBindViewHolder(final ViewHolder holder, int position) {
        Neighbour neighbour = mNeighbours.get(position);

        holder.mNeighbourName.setText(neighbour.getName());
        Glide.with(holder.mNeighbourAvatar.getContext())
            .load(neighbour.getAvatarUrl())
            .apply(RequestOptions.circleCropTransform())
            .into(holder.mNeighbourAvatar);

        holder.mDeleteButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (mStar == true) {
                    EventBus.getDefault().post(new DeleteNeighbourEvent(neighbour));
                } else {
                    EventBus.getDefault().post(new DeleteNeighbourStarEvent(neighbour));
                }
                EventBus.getDefault().post(new ActualiseEvent());
            }
        });
        holder.mStarButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (mNeighboursStar.contains(neighbour)){
                    EventBus.getDefault().post(new DeleteNeighbourStarEvent(neighbour));
                    holder.mStarButton.setImageResource(R.drawable.ic_star_border_yellow_24dp);
                    EventBus.getDefault().post(new ActualiseEvent());
                } else{
                    EventBus.getDefault().post(new AddNeighbourStarEvent(neighbour));
                    holder.mStarButton.setImageResource(R.drawable.ic_star_yellow_24dp);
                    EventBus.getDefault().post(new ActualiseEvent());
                }
            }
        });
    }

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

    public class ViewHolder extends RecyclerView.ViewHolder {
        @BindView(R.id.item_list_avatar)
        public ImageView mNeighbourAvatar;
        @BindView(R.id.item_list_name)
        public TextView mNeighbourName;
        @BindView(R.id.item_list_delete_button)
        public ImageButton mDeleteButton;
        @BindView(R.id.item_list_star_button)
        public ImageButton mStarButton;
    }
}
```

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

        Intent nextPage = new Intent(holder.mNeighbourAvatar.getContext(),
                                      PresentationActivity.class);
        nextPage.putExtra( name: "nom",neighbour.getName());
        nextPage.putExtra( name: "avatar",neighbour.getAvatarUrl());
        nextPage.putExtra( name: "id",neighbour.getId());
        nextPage.putExtra( name: "localisation",neighbour.getLocalisation());
        nextPage.putExtra( name: "tel",neighbour.getTel());
        nextPage.putExtra( name: "description",neighbour.getDescription());
        holder.mNeighbourAvatar.getContext().startActivity(nextPage);

    }
});

if (mStar == false){
    holder.mStarButton.setVisibility(GONE);
}

if (mNeighboursStar.contains(neighbour)){
    holder.mStarButton.setImageResource(R.drawable.ic_star_yellow_24dp);
} else{
    holder.mStarButton.setImageResource(R.drawable.ic_star_border_yellow_24dp);
}

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

public class ViewHolder extends RecyclerView.ViewHolder {
    @BindView(R.id.item_list_avatar)
    public ImageView mNeighbourAvatar;
    @BindView(R.id.item_list_name)
    public TextView mNeighbourName;
    @BindView(R.id.item_list_delete_button)
    public ImageButton mDeleteButton;
    @BindView(R.id.item_list_star_button)
    public ImageButton mStarButton;

    public ViewHolder(View view) {
        super(view);
        ButterKnife.bind( target: this, view);
    }
}
```

Parcours Développeur Android

La nouvelle activité de la nouvelle fonctionnalité : PresentationActivity

```
public class PresentationActivity extends AppCompatActivity {

    private TextView text = null;
    private TextView lieuTel = null;
    private TextView presentation = null;
    private TextView loca = null;
    private TextView telephone = null;
    private TextView facebook = null;
    private ImageView image = null;
    private List<Neighbour> mNeighboursStar;
    private FloatingActionButton fab;
    private Button back;
    private Neighbour neighbour = null;
    private Neighbour ApiService mApiService;

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

        ApiService = DI.getNeighbour ApiService();
        mNeighboursStar = ApiService.getNeighboursStar();

        text = (TextView) findViewById(R.id.text);
        lieuTel = (TextView) findViewById(R.id.lieuTel);
        presentation = (TextView) findViewById(R.id.description);
        loca = (TextView) findViewById(R.id.loca);
        telephone = (TextView) findViewById(R.id.tel);
        facebook = (TextView) findViewById(R.id.facebook);
        image = (ImageView) findViewById(R.id.imageAvatar);

        Intent i = getIntent();
        String nom = i.getStringExtra("nom");
        int id = i.getIntExtra("id", defaultValue: 0);
        String avatar = i.getStringExtra("avatar");
        String localisation = i.getStringExtra("localisation");
        String tel = i.getStringExtra("tel");
        String description = i.getStringExtra("description");

        neighbour = new Neighbour(id,nom,avatar,localisation,tel,description);
    }

    configureFab();
    configureBack();
}
```

```
Glide.with( activity: this)
    .load(avatar)
    .override( width: 420, height: 350) // resizes the image to these dimensions
    .centerCrop()
    .into(image);

text.setText(nom );
lieuTel.setText(nom );
presentation.setText(description);
loca.setText(localisation);
telephone.setText(tel );
facebook.setText("www.facebook.fr/" + nom.toLowerCase());
presentation.setMovementMethod(new ScrollingMovementMethod());

}

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

    if (mNeighboursStar.contains(neighbour)){
        fab.setImageResource(R.drawable.ic_star_yellow_24dp);
    }else{
        fab.setImageResource(R.drawable.ic_star_border_yellow_24dp);
    }

    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (mNeighboursStar.contains(neighbour)){
                EventBus.getDefault().post(new DeleteNeighbourStarEvent(neighbour));
                fab.setImageResource(R.drawable.ic_star_border_yellow_24dp);
            }else{
                EventBus.getDefault().post(new AddNeighbourStarEvent(neighbour));
                fab.setImageResource(R.drawable.ic_star_yellow_24dp);
            }
        }
    });
}

private void configureBack() {
    back = findViewById(R.id.back);

    back.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { finish(); }
    });
}
```

Parcours Développeur Android

Les tests unitaires de l'application :

```
// ...
 * Unit test on Neighbour service
 */
@RunWith(JUnit4.class)
public class NeighbourServiceTest {

    private Neighbour ApiService service;

    @Before
    public void setup() { service = DI.getNewInstance ApiService(); }

    @Test
    public void getNeighboursWithSuccess() {
        List<Neighbour> neighbours = service.getNeighbours();
        List<Neighbour> expectedNeighbours = DummyNeighbourGenerator.DUMMY_NEIGHBOURS;
        assertThat(neighbours, IsIterableContainingInAnyOrder.containsInAnyOrder(expectedNeighbours.toArray()));
    }

    @Test
    public void deleteNeighbourWithSuccess() {
        Neighbour neighbourToDelete = service.getNeighbours().get(0);
        service.deleteNeighbour(neighbourToDelete);
        assertFalse(service.getNeighbours().contains(neighbourToDelete));
    }

    // Mes tests que j'ai créé
    @Test
    public void getStarUser() {
        List<Neighbour> neighboursStar = service.getNeighboursStar();
        List<Neighbour> expectedNeighboursStar = DummyNeighbourGenerator.DUMMY_NEIGHBOURS_STAR;
        assertThat(neighboursStar, IsIterableContainingInAnyOrder.containsInAnyOrder(expectedNeighboursStar.toArray()));
    }

    @Test
    public void deleteStarUser() {
        Neighbour neighbourToDeleteStar = service.getNeighboursStar().get(0);
        service.deleteNeighbourStar(neighbourToDeleteStar);
        assertFalse(service.getNeighboursStar().contains(neighbourToDeleteStar));
    }

    @Test
    public void addStarUser() {
        Neighbour neighbourToAddStar = service.getNeighbours().get(0);
        service.addNeighbourStar(neighbourToAddStar);
        assertTrue(service.getNeighboursStar().contains(neighbourToAddStar));
    }

    @Test
    public void addUser() {

        Neighbour neighbourToAdd = service.getNeighbours().get(0);
        service.addNeighbour(neighbourToAdd);
        assertTrue(service.getNeighbours().contains(neighbourToAdd));
    }
}
```

Cossu Denis -Projet 3-

Parcours Développeur Android

Les tests instrumentalisés de l'application : ViewActions et ViewAssertion (déjà fait avant sauf LaunchActivityAction)

```
public class DeleteViewAction implements ViewAction {  
    @Override  
    public Matcher<View> getConstraints() { return null; }  
  
    @Override  
    public String getDescription() { return "Click on specific button"; }  
  
    @Override  
    public void perform(UiController uiController, View view) {  
        View button = view.findViewById(R.id.item_list_delete_button);  
        button.performClick();  
    }  
}
```

```
public class LaunchActivityAction implements ViewAction {  
    @Override  
    public Matcher<View> getConstraints() { return null; }  
  
    @Override  
    public String getDescription() { return "Click on specific button"; }  
  
    @Override  
    public void perform(UiController uiController, View view) {  
        View button = view.findViewById(R.id.item_list_name);  
        button.performClick();  
    }  
}
```

```
public class RecyclerViewItemCountAssertion implements ViewAssertion {  
    private final Matcher<Integer> matcher;  
  
    public static RecyclerViewItemCountAssertion withItemCount(int expectedCount) {  
        return withItemCount.Matchers.is(expectedCount));  
    }  
  
    public static RecyclerViewItemCountAssertion withItemCount(Matcher<Integer> matcher) {  
        return new RecyclerViewItemCountAssertion(matcher);  
    }  
  
    private RecyclerViewItemCountAssertion(Matcher<Integer> matcher) { this.matcher = matcher; }  
  
    @Override  
    public void check(View view, NoMatchingViewException noViewFoundException) {  
        if (noViewFoundException != null) {  
            throw noViewFoundException;  
        }  
  
        RecyclerView recyclerView = (RecyclerView) view;  
        RecyclerView.Adapter adapter = recyclerView.getAdapter();  
        Assert.assertThat(adapter.getItemCount(), matcher);  
    }  
}
```

Cossu Denis -Projet 3-

Parcours Développeur Android

Les tests instrumentalisés de l'application : (déjà fait)

```
/*
 * Test class for list of neighbours
 */
@RunWith(AndroidJUnit4.class)
public class NeighboursListTest {

    // This is fixed
    private static int ITEMS_COUNT = 12;
    private static int ITEMS_COUNT_STAR = 2;

    private ListNeighbourActivity mActivity;

    @Rule
    public ActivityTestRule<ListNeighbourActivity> mActivityRule =
        new ActivityTestRule(ListNeighbourActivity.class);

    @Before
    public void setUp() {
        mActivity = mActivityRule.getActivity();
        assertThat(mActivity, notNullValue());
    }

    /**
     * We ensure that our recyclerview is displaying at least one item
     */
    @Test
    public void myNeighboursList_shouldNotBeEmpty() {
        // First scroll to the position that needs to be matched and click on it.
        onView(withId(R.id.list_neighbours))
            .check(matches(hasMinimumChildCount( minChildCount: 1)));
    }

    /**
     * When we delete an item, the item is no more shown
     */
    @Test
    public void myNeighboursList_deleteAction_shouldRemoveItem() {
        // Given : We remove the element at position 4
        onView(withId(R.id.list_neighbours)).check(withItemCount(ITEMS_COUNT));
        // When perform a click on a delete icon
        onView(withId(R.id.list_neighbours))
            .perform(RecyclerViewActions.actionOnItemAtPosition( position: 3, new DeleteViewAction()));
        // Then : the number of element is 11

        ITEMS_COUNT = ITEMS_COUNT - 1;
        onView(withId(R.id.list_neighbours)).check(withItemCount(ITEMS_COUNT));
    }
}
```

Cossu Denis -Projet 3-

Parcours Développeur Android

Les tests instrumentalisés de l'application : (fait par moi)

```
// mes tests que j'ai créé :  
  
@Test  
public void myNeighboursList_LaunchDetailActivity() {  
  
    Intents.init();  
  
    onView(withId(R.id.list_neighbours))  
        .perform(RecyclerViewActions.actionOnItemAtPosition( position: 1, new LaunchActivityAction()));  
  
    intended(hasComponent(PresentationActivity.class.getName()));  
}  
  
@Test  
public void myNeighboursList_DetailActivity_withGoodName() {  
  
    UiDevice mDevice = UiDevice.getInstance(InstrumentationRegistry.getInstrumentation());  
  
    onView(withId(R.id.list_neighbours))  
        .perform(RecyclerViewActions.actionOnItemAtPosition( position: 0, new LaunchActivityAction()));  
  
    ViewInteraction textConfirmJack = onView(withId(R.id.lieuTel));  
    textConfirmJack.check(matches(withText("Jack")));  
  
    mDevice.pressBack();  
  
    onView(withId(R.id.list_neighbours))  
        .perform(RecyclerViewActions.actionOnItemAtPosition( position: 1, new LaunchActivityAction()));  
  
    ViewInteraction textConfirmChloé = onView(withId(R.id.lieuTel));  
    textConfirmChloé.check(matches(withText("Chloé")));  
}  
}
```

```
@Test  
public void myNeighboursList_ScreenStarWithOnlyFavorite() {  
  
    onView(withId(R.id.list_neighbours_star))  
        .check(withItemCount(ITEMS_COUNT_STAR))  
        .check(matches(hasDescendant(withText("Caroline"))))  
        .check(matches(hasDescendant(withText("Patrick"))));  
}  
  
// pas demandé  
  
@Test  
public void myReunionsList.AddItem() {  
  
    onView(withId(R.id.list_neighbours)).check(withItemCount(ITEMS_COUNT));  
  
    onView(withId(R.id.fab)).perform(click());  
  
    onView(withText("Valider")).perform(click());  
  
    ITEMS_COUNT = ITEMS_COUNT + 1;  
  
    onView(withId(R.id.list_neighbours)).check(withItemCount(ITEMS_COUNT));  
}
```

Parcours Développeur Android



les 2 rapports d'exécution (unitaire et instrumentalisé) des tests finaux qui réussissent :

NeighbourServiceTest: 6 total, 6 passed

```
&quot;/private/var/folders/mb/87l0xvrn54z1558tw56_6m500000gn/T/AppTrar  
Studio.app/Contents/jre/jdk/Contents/Home/bin/java&quot; -ea -Didea.test.cy  
Didea.launcher.bin.path=/private/var/folders/mb/87l0xvrn54z1558tw56_6m50  
Studio.app/Contents/bin&quot; -Dfile.encoding=UTF-8 -classpath &quot;/priv  
/F88C4349-7E2B-488B-AA20-0B29E08A3532/d/Android Studio.app/Content  
/T/AppTranslocation/F88C4349-7E2B-488B-AA20-0B29E08A3532/d/Android  
/mb/87l0xvrn54z1558tw56_6m500000gn/T/AppTranslocation/F88C4349-7E2
```

NeighboursListTest: 6 total, 6 passed

```
com.openclassrooms.entrevoisins.neighbour_list.NeighboursListTest  
myNeighboursList_shouldNotBeEmpty  
myNeighboursList_deleteAction_shouldRemoveItem  
myNeighboursList_DetailActivity_withGoodName  
myReunionsList_addItem  
myNeighboursList_LaunchDetailActivity  
myNeighboursList_ScreenStarWithOnlyFavorite
```



Parcours Développeur Android

Le fichier README.md :

OpenClassrooms

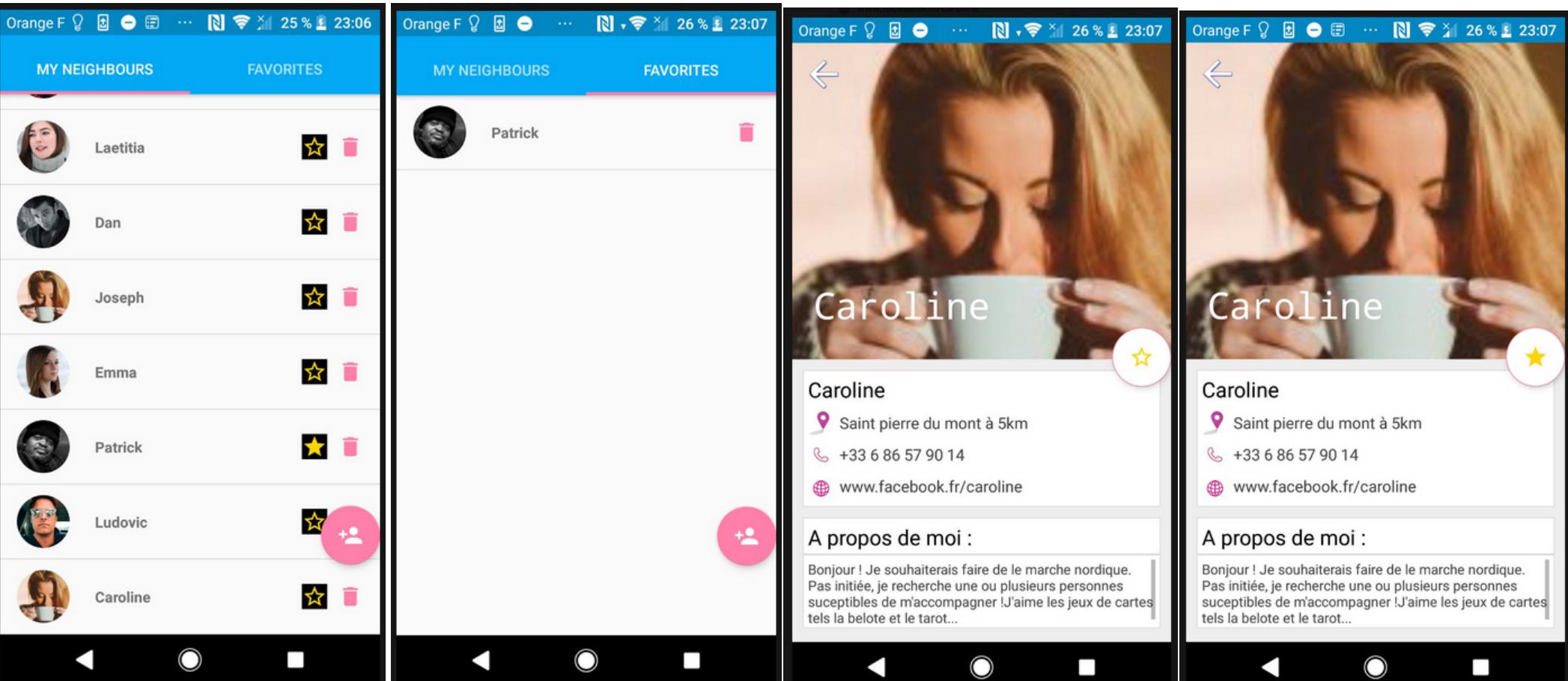
Ce dépôt contient une mini-application pour le P3 du parcours **Grande École du Numérique**. Pour utiliser ce projet vous devez cliquer sur le bouton 'clone or download' vous pouvez le télécharger en fichier zip ou faire un git clone avec le lien donné. Cliquez pour ouvrir Android Studio puis allez dans l'onglet 'File', puis 'New' et enfin 'Import Project' et vous devez sélectionnez le dossier contenant votre projet. Une fois le projet ouvert dans Android studio vous pouvez l'exécuter et le compiler en appuyant sur le petit marteau vert puis sur la flèche verte dans la barre du menu en haut de l'écran.

Cossu Denis -Projet 3-

Parcours Développeur Android



Conclusion :

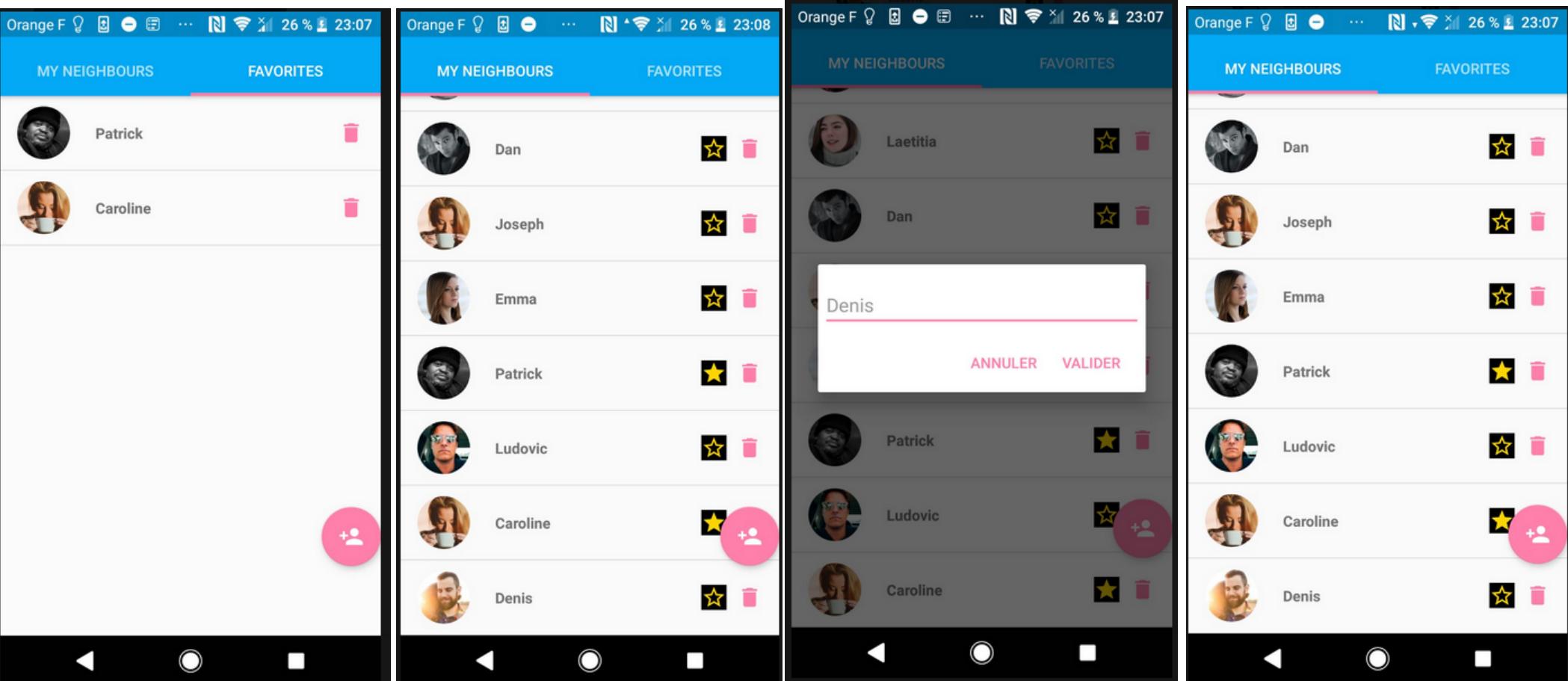


Cossu Denis -Projet 3-

Parcours Développeur Android



Conclusion :



Cossu Denis -Projet 3-