

HE
IG^{VD}

IICT
Institut des
Technologies de
l'information et de
la communication



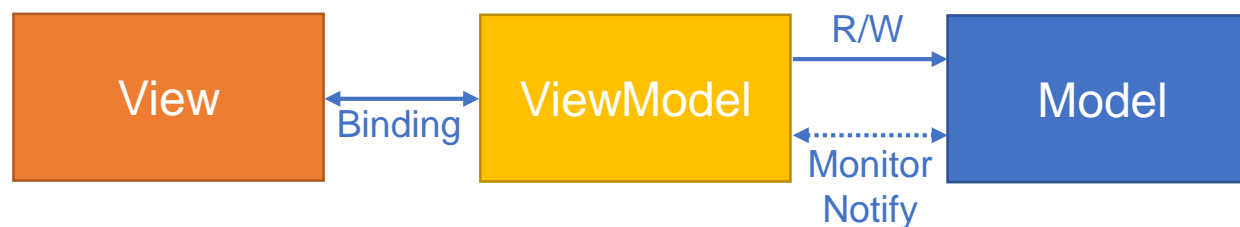
Développement d'applications *Android*

Laboratoire 3

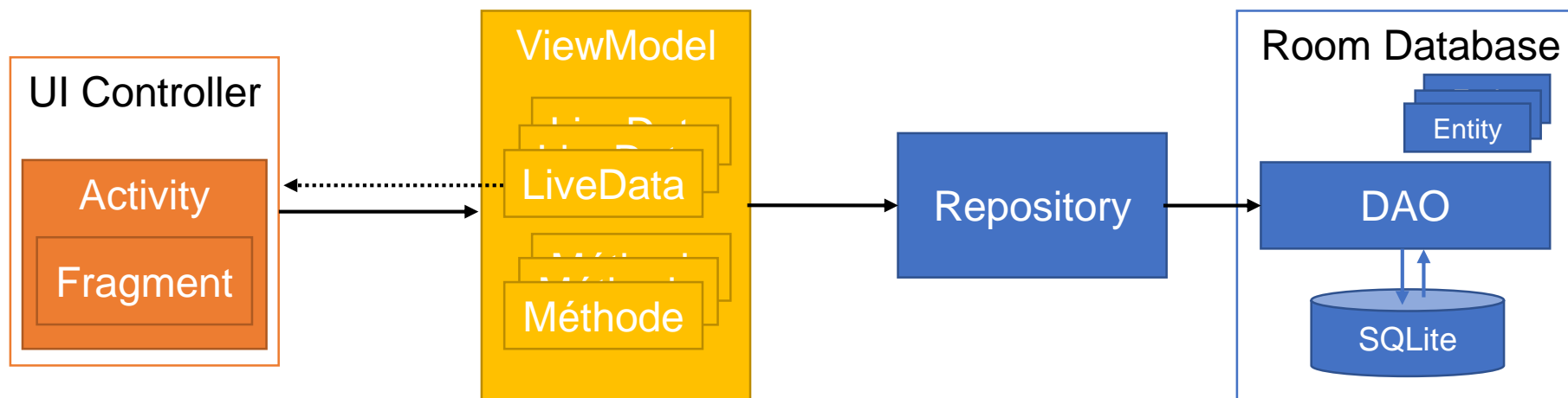
Fabien Dutoit

Rappel MVVM

- Architecture MVVM



- Appliquée sur *Android*



Base de données

- Utilisation d'*Android Room* pour gérer la base de données
- Les modèles sont fournis (Entities), relation One-to-One (0...1)

@Entity

```
data class Note (  
    @PrimaryKey(autoGenerate = true) var noteId : Long?,  
    var state : State,  
    var title : String,  
    var text : String,  
    var creationDate : Calendar,  
    var type : Type  
)
```

@Entity

```
data class Schedule (  
    @PrimaryKey(autoGenerate = true) var scheduleId : Long?,  
    var ownerId : Long,  
    var date : Calendar  
)
```

```
data class NoteAndSchedule (  
    @Embedded val note: Note,  
    @Relation(  
        parentColumn = "noteId",  
        entityColumn = "ownerId"  
    )  
    val schedule: Schedule?  
)
```

```
enum class State { IN_PROGRESS, DONE }
```

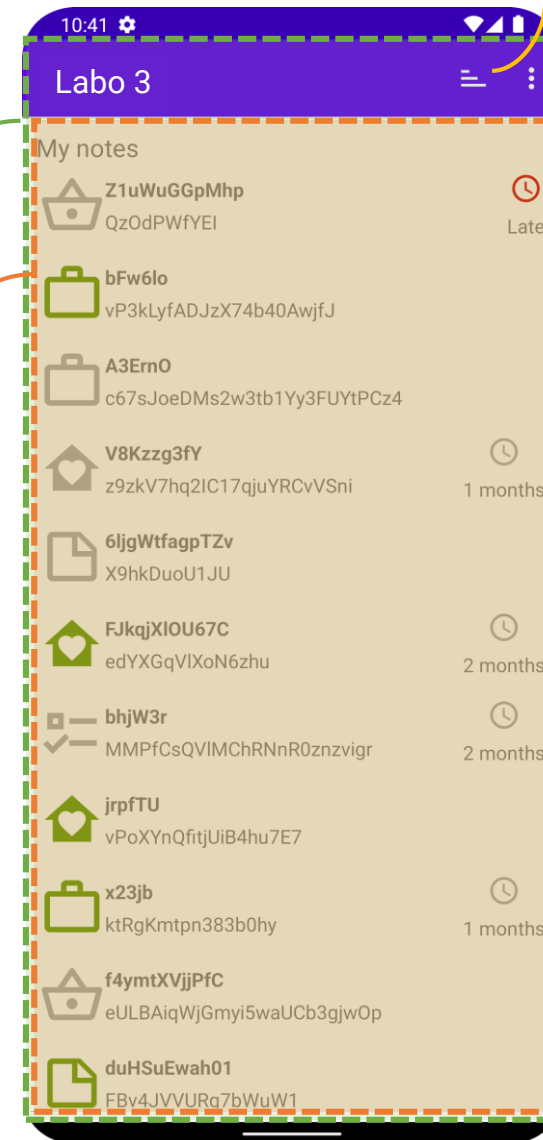
```
enum class Type { NONE, TODO, SHOPPING, WORK, FAMILY }
```

Interface souhaitée – Smartphone

- MainActivity, accueillant :
 - Le *Fragment* avec la *RecyclerView* affichant les Notes
 - Un Menu avec 4 options:
 - Tri des Notes
 - Par date de création
 - Par date prévue (Schedule)
 - Générer une note (aléatoire)
 - Supprimer toutes les notes
 - Le ViewModel offrant
 - L'accès aux *LiveData* alimentant la *RecyclerView*
 - Les méthodes permettant le tri et les actions (ajout/suppression) sur les Notes

Activité

Fragment
Notes



Creation date

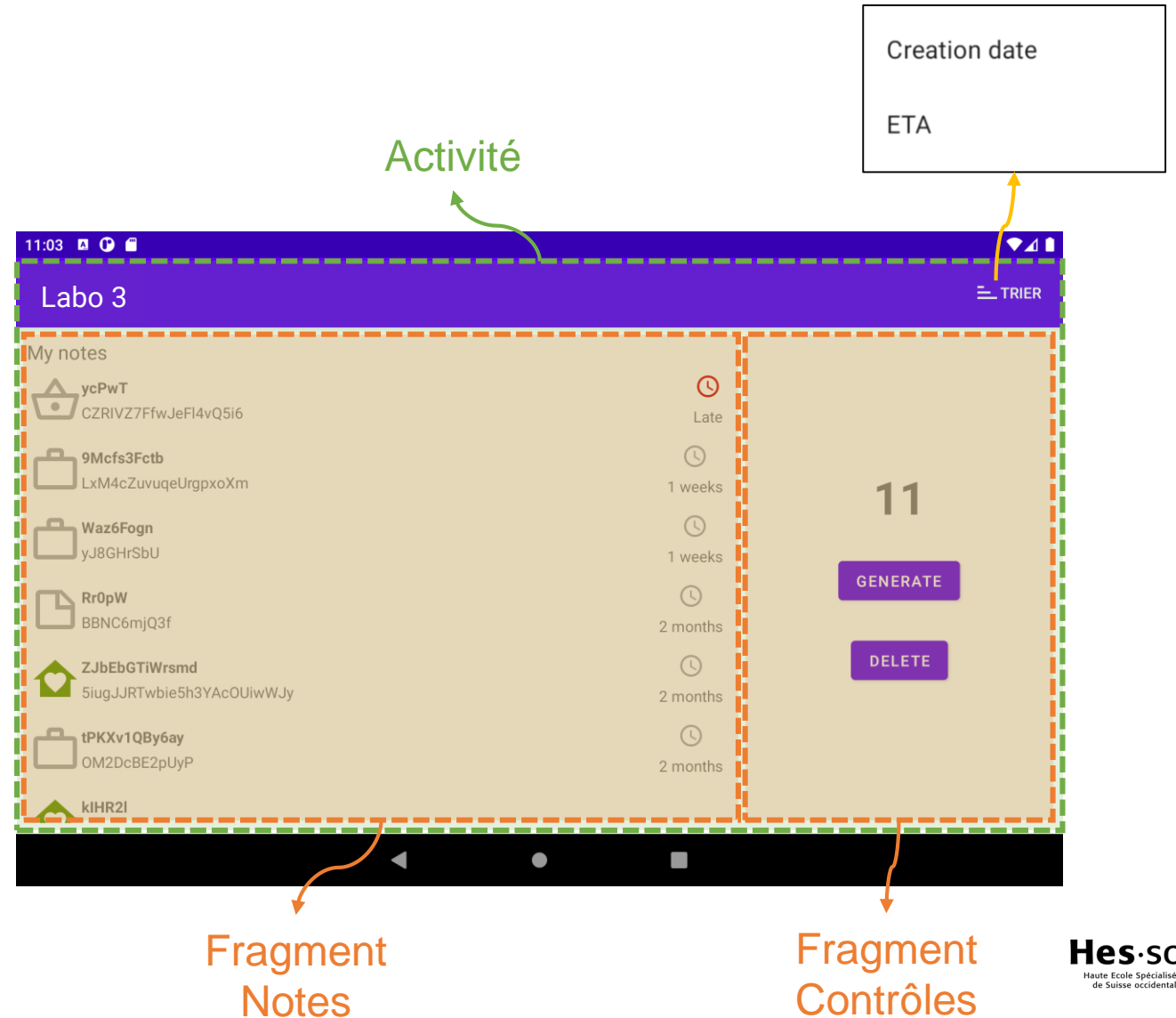
ETA

Generate

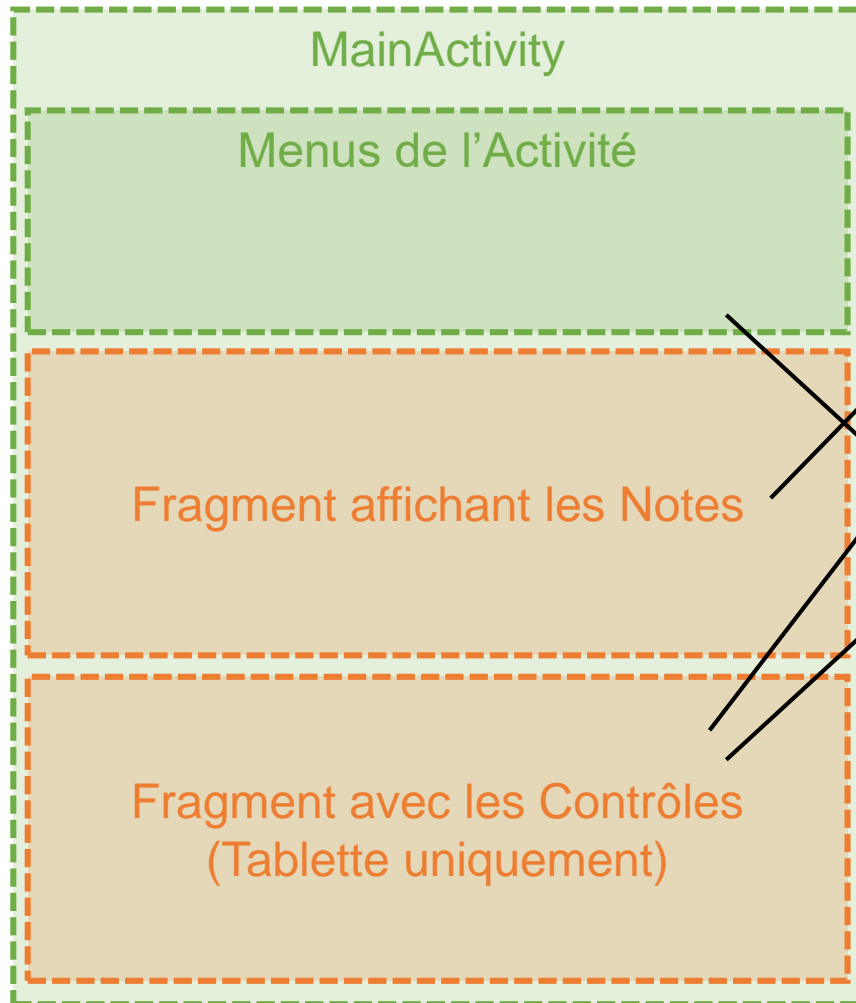
Delete all

Interface souhaitée – Tablette (configuration large)

- MainActivity, accueillant :
 - Le *Fragment* avec la *RecyclerView* affichant les Notes
 - Un *Fragment* avec un compteur du nombre de Notes et deux actions:
 - Générer une note (aléatoire)
 - Supprimer toutes les notes
 - Un Menu avec 2 options:
 - Tri des Notes
 - Par date de création
 - Par date prévue (Schedule)
 - Le ViewModel



ViewModel – API



```
class NotesViewModel(private val repository: DataRepository) : ViewModel() {  
    val allNotes = repository.allNotes //: LiveData<List<NoteAndSchedule>>  
    val countNotes = repository.countNotes //: LiveData<Int>  
    {  
        fun generateANote() { ... }  
        fun deleteAllNote() { ... }  
    }  
}
```

HEIG
VD
HAUTE ÉCOLE
D'INGÉNIEURIE
ET DE GESTION
DU CANTON
DE VAUD

HE^{VD}
IG

HAUTE ÉCOLE
D'INGÉNIEURIE
ET DE GESTION
DU CANTON
DE VAUD