
To-Do Liste

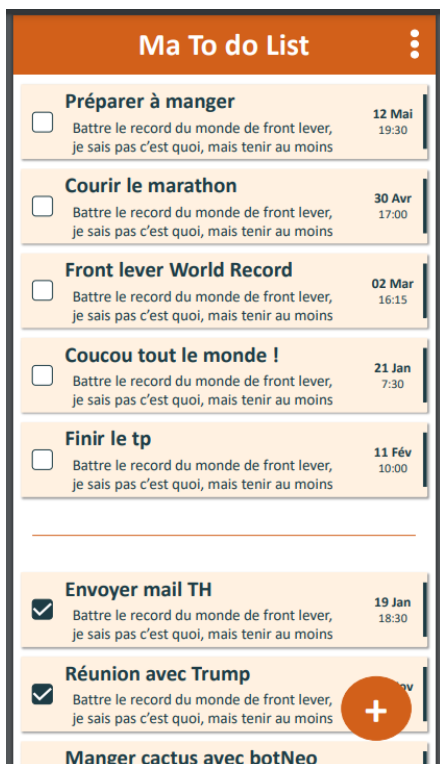
*Leboeuf Sullivan, Garcia Christophe
M. Le Lain, M. Kerbellec*

Table des matières

1. Interfaces initiales	2
2. Diagramme de classes initial	3
3. Interfaces finales	4
4. Diagramme de classes final	5
5. Diagrammes de séquence (finaux)	6
6. Le fonctionnement de l'application	8
7. Les cas limites	9
8. Améliorations possibles.....	9

1. Interfaces initiales

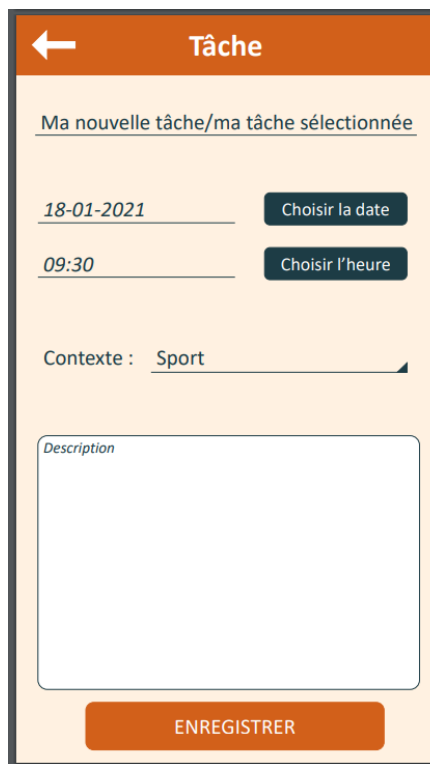
Voici les interfaces que nous avons créées au début du projet. Nous étions plutôt ambitieux par rapport à notre savoir-faire, mais finalement le projet s'est bien passé. Vous pourrez constater que certains éléments sont différents entre les interfaces initiales et finales.



Ma To do List

- ☐ **Préparer à manger** 12 Mai 19:30
Battre le record du monde de front lever, je sais pas c'est quoi, mais tenir au moins
- ☐ **Courir le marathon** 30 Avr 17:00
Battre le record du monde de front lever, je sais pas c'est quoi, mais tenir au moins
- ☐ **Front lever World Record** 02 Mar 16:15
Battre le record du monde de front lever, je sais pas c'est quoi, mais tenir au moins
- ☐ **Coucou tout le monde !** 21 Jan 7:30
Battre le record du monde de front lever, je sais pas c'est quoi, mais tenir au moins
- ☐ **Finir le tp** 11 Fév 10:00
Battre le record du monde de front lever, je sais pas c'est quoi, mais tenir au moins
- ☒ **Envoyer mail TH** 19 Jan 18:30
Battre le record du monde de front lever, je sais pas c'est quoi, mais tenir au moins
- ☒ **Réunion avec Trump** 19 Jan 18:30
Battre le record du monde de front lever, je sais pas c'est quoi, mais tenir au moins

Manger cactus avec botNeo



Tâche

Ma nouvelle tâche/ma tâche sélectionnée

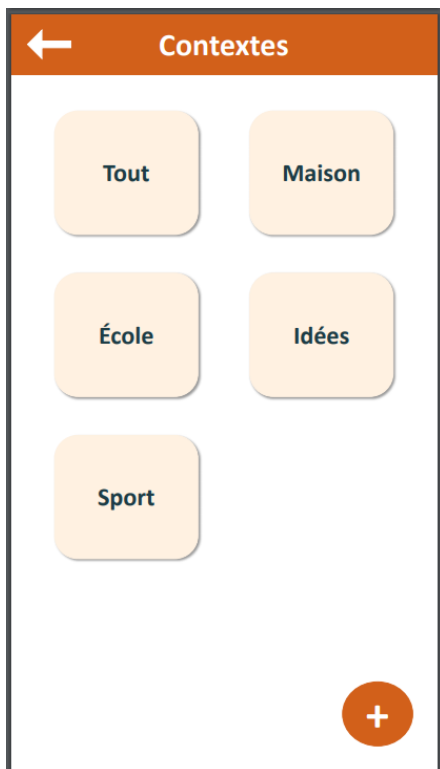
18-01-2021 Choisir la date

09:30 Choisir l'heure

Contexte : Sport

Description

ENREGISTRER



Contextes

Tout Maison

École Idées

Sport

+



Contextes

Tout Maison

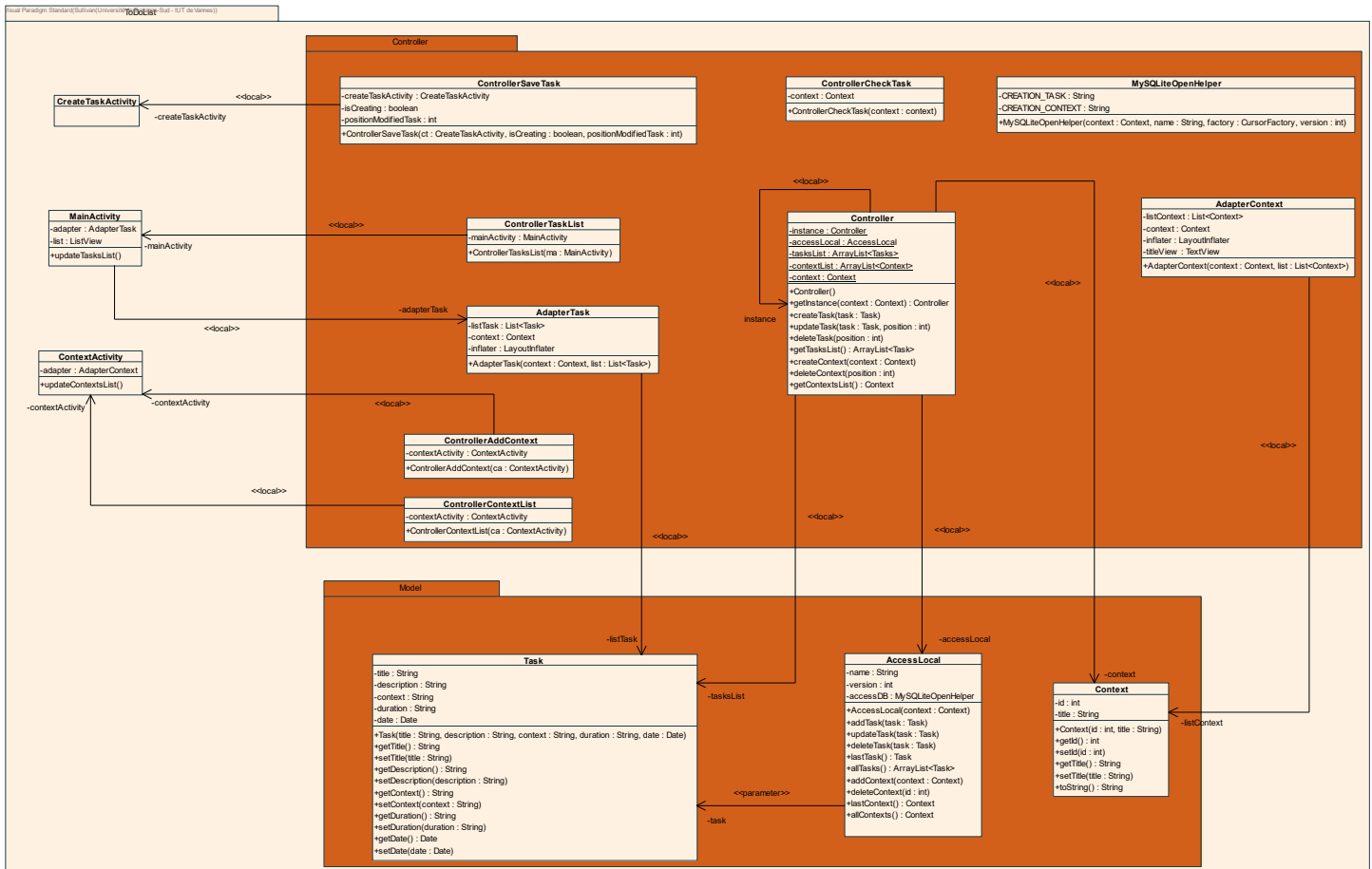
École Idées

Mon nouveau contexte Ajouter

Clavier

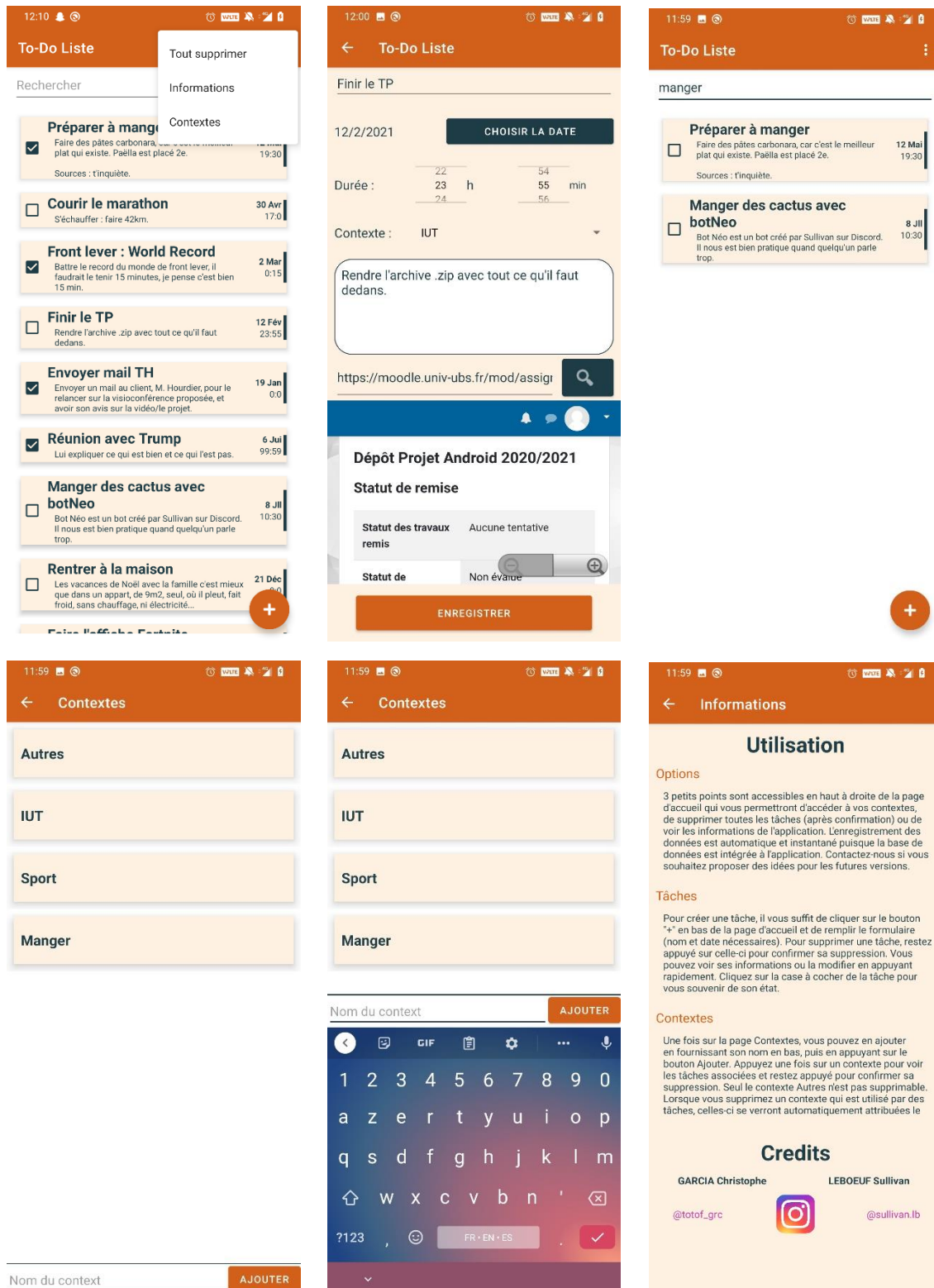
2. Diagramme de classes initial

Le diagramme a été réalisé lors de la première séance sur Visual Paradigm, le voici :



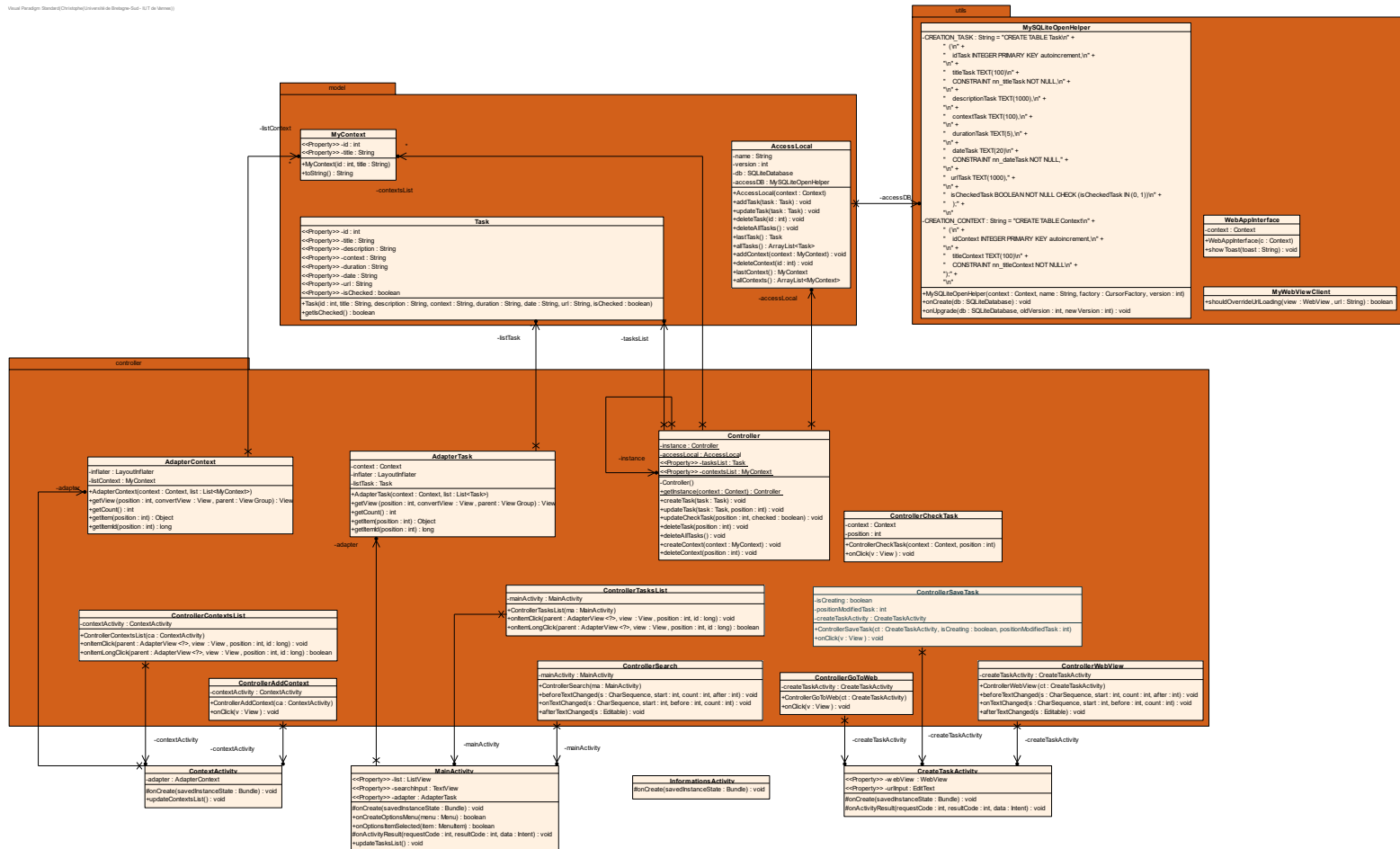
3. Interfaces finales

Et voici les interfaces finales. Ce sont des screens d'un téléphone ayant téléchargé et installé l'apk.



4. Diagramme de classes final

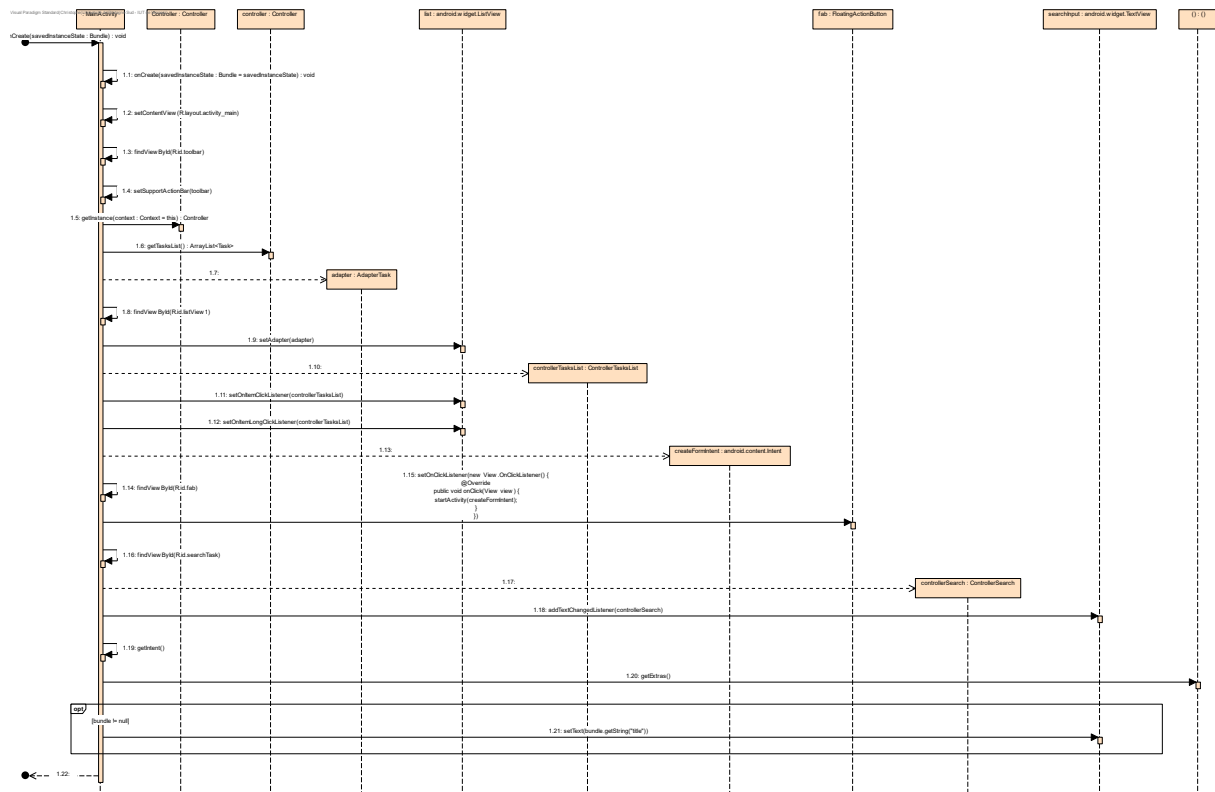
Nous avons pu ensuite générer le diagramme de classes depuis Visual Paradigm en important le code source de l'application, le voici :



5. Diagrammes de séquence (finaux)

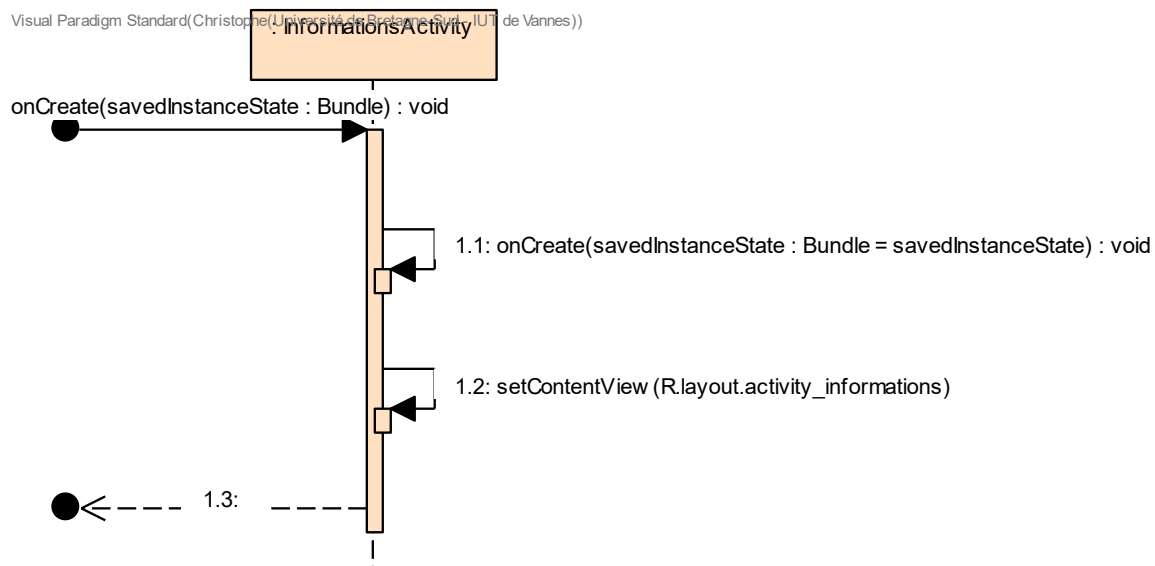
Voici le diagramme de séquence de chacune des méthodes *onCreate()* des activités :

MainActivity



InformationsActivity

Visual Paradigm Standard(Christophe(University of Bretagne Sud, IUT de Vannes))



```
sequenceDiagram
    participant ContextActivity
    participant Controller as Controller : Controller
    participant Controller as controller : Controller
    participant list as list : ListView
    participant button as button : Button
    participant AdapterContext as this.adapter : AdapterContext
    participant ControllerContextsList as controllerContextsList : ControllerContextsList

    ContextActivity->>ContextActivity: onCreate(savedInstanceState : Bundle) : void
    activate ContextActivity
    ContextActivity->>Controller: 1.1: onCreate(savedInstanceState : Bundle = savedInstanceState) : void
    deactivate ContextActivity
    ContextActivity->>Controller: 1.2: setContentView(R.layout.activity_context)
    deactivate ContextActivity
    ContextActivity->>Controller: 1.3: getInstance(context : Context = this) : Controller
    activate Controller
    Controller->>controller: 1.4: getContextsList() : ArrayList<MyContext>
    activate controller
    controller-->>AdapterContext: 1.5:
    deactivate controller
    ContextActivity->>Controller: 1.6: findViewById(R.id.contextsList)
    deactivate ContextActivity
    ContextActivity->>list: 1.7: setAdapter(adapter)
    activate list
    ContextActivity->>button: 1.8: findViewById(R.id.addContextButton)
    deactivate ContextActivity
    ContextActivity->>button: 1.9: setOnClickListener(new ControllerAddContext(this))
    activate button
    button-->>ControllerContextsList: 1.10:
    deactivate button
    ContextActivity->>list: 1.11: setOnItemClickListener(controllerContextsList)
    activate list
    ContextActivity->>list: 1.12: setOnItemLongClickListener(controllerContextsList)
    activate list
    ContextActivity-->>ContextActivity: 1.13:
    deactivate ContextActivity
```

```

sequenceDiagram
    participant User
    participant Controller
    participant Service
    participant Repository
    participant Mailer
    participant PasswordGenerator
    participant Validator
    participant Logger
    participant TransactionManager

    User->>Controller: createNewUser()
    activate Controller
    Controller->>Service: validateUser()
    activate Service
    Service->>Repository: saveUser()
    activate Repository
    Repository->>Mailer: sendWelcomeEmail()
    activate Mailer
    Mailer->>PasswordGenerator: generatePassword()
    activate PasswordGenerator
    PasswordGenerator->>Validator: validatePassword()
    activate Validator
    Validator->>Logger: logUserCreation()
    activate Logger
    Logger->>TransactionManager: commitTransaction()
    activate TransactionManager
    TransactionManager->>Controller: returnUser()
    deactivate TransactionManager
    deactivate Logger
    deactivate Validator
    deactivate PasswordGenerator
    deactivate Mailer
    deactivate Repository
    deactivate Service
    Controller->>User: returnConfirmation()
    deactivate Controller
  
```

6. Le fonctionnement de l'application

Lorsque l'utilisateur arrive sur l'application, reconnaissable par son **logo**, il voit la liste de toutes ses tâches. L'application contient une **base de données** SQLite pour sauvegarder les tâches et les contextes.

Le **bouton « + »** permet de créer une nouvelle tâche, il suffit de renseigner :

- Le nom (nécessaire)
- La date (nécessaire)
- Une durée : les heures, les minutes
- Un contexte : depuis ceux disponibles
- Une description
- Un lien : visualisable juste en dessous dans la **WebView** si le lien existe, et dans le **navigateur web par défaut** si l'utilisateur clique sur le bouton *Loupe*

En cliquant normalement sur une de ses **tâches**, il peut voir et modifier celle-ci. En cliquant longuement, il peut la supprimer (après confirmation). L'utilisateur peut **cocher ou décocher** une case (action instantanément enregistrée) pour se souvenir de l'état de la tâche sans vouloir la supprimer.

Une **barre de recherche** est disponible pour chercher une tâche par son nom, sa description, sa date, sa durée et son contexte. A chaque fois que le texte de recherche est changé, l'application met à jour la liste des tâches visibles.

Des **options** sont accessibles avec les 3 petits points en haut à droite de la page :

- « **Tout supprimer** » permet de supprimer l'entièreté de ses tâches (après confirmation)
- « **Informations** » permet de voir l'utilisation de l'application et les crédits
- « **Contextes** » permet d'accéder à ses contextes

En cliquant sur « Contextes », l'utilisateur voit la liste de tous ses contextes. Il peut en créer un en fournissant un titre puis en appuyant sur le bouton ajouter. Il ne peut pas créer un contexte dont le nom est déjà utilisé. En appuyant normalement sur un contexte, il verra sur la page principale la liste des tâches correspondant à celui-ci. En appuyant longuement, il peut supprimer le contexte (après confirmation). Seul le contexte « Autres » n'est pas supprimable. Ce contexte « Autres » est le contexte par défaut des tâches. Il est également attribué automatiquement aux tâches lorsque leur contexte est supprimé.

Des **toasts** apparaissent dans l'application :

- « Veuillez préciser le titre et la date. », lorsque l'utilisateur oublie de fournir le titre et/ou la date lors de la création d'une tâche
- « Veuillez saisir une URL valide. », lorsque l'utilisateur souhaite cliquer sur le bouton permettant d'accéder au lien de la tâche dans son navigateur web par défaut
- « Aucune tâche trouvée. », lorsque aucune tâche ne correspond à la recherche de l'utilisateur
- « Ce contexte n'est pas supprimable. », lorsque l'utilisateur souhaite supprimer le contexte « Autres »
- « Toutes les tâches ont été supprimées. », lorsque l'utilisateur a supprimé toutes les tâches de l'application

7. Les cas limites

Quelques cas limites sont présents dans notre application.

Déjà, nous sommes limités par le nombre de caractères introduisables dans la base de données :

- 100 : titre d'une tâche et d'un contexte
- 1000 : description et url d'une tâche
- 99 : nombre d'heures d'une tâche

Le retour en arrière, d'une activité à la précédente, n'est pas toujours parfait. Parfois nous pouvons retourner en arrière alors qu'on ne devrait pas, et inversement.

Lorsque l'utilisateur clique sur le bouton *Loupe* (charger l'url dans le navigateur web par défaut), l'application aurait dû vérifier si l'url existe, et pas seulement sa validité (ce qu'on fait).

8. Améliorations possibles

- Notification** : notifier l'utilisateur de la tâche le jour J (c'était en cours, projet annulé par manque de temps)
- Thème** : offrir la possibilité de passer sur un mode sombre
- Images** : permettre l'insertion d'images dans une tâche et un contexte
- Langues** : langue anglaise disponible
- Partage** : partager une tâche avec une autre personne ayant l'application

Sullivan et moi utilisons réellement l'application sur notre téléphone, c'est pourquoi nous améliorerons, dans tous les cas, la To-Do liste pour notre utilisation personnelle.