

SAE 3.01 – Développement d'une application

LOGICIEL D'ORGANISATION DE TACHES PERSONNELLES

Description du sujet

L'application à réaliser dans cette SAE est un **organisateur de tâches**. De nombreux sites proposent ce type d'outils (Trello, Asana, ...) avec différentes approches.

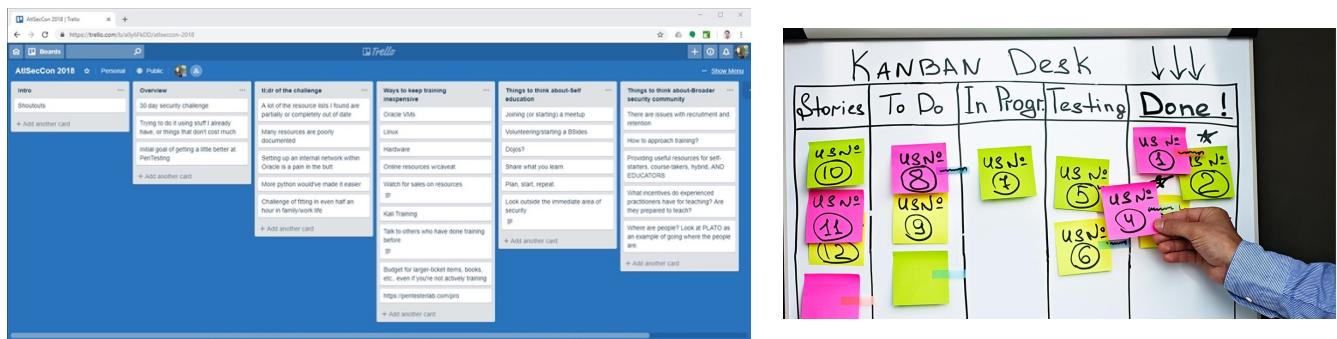


Figure 1 : A gauche, Trello (<https://trello.com>) permet une visualisation par colonnes contenant les tâches à réaliser et facilitant le pilotage de projet par une méthode agile. A droite un tableau Kanban, les tâches sont sur des post-its qui peuvent glisser d'une colonne à la suivante en fonction de l'état de leur réalisation.

The figure shows two screenshots of the Asana interface. On the left, a 'Liste de tâches' view is shown, displaying a grid of tasks with columns for 'Nom de la tâche', 'Responsable', 'Échéance', 'Type de travail', and 'Priorité'. On the right, a 'Tableau' (Timeline) view is shown, displaying tasks across days (Lundi, Mardi, Mercredi, Jeudi) with columns for 'To Do', 'In Progress', 'Testing', and 'Done!'. Each task is represented by a card with details like priority, due date, and assignee.

Figure 2 – Asana (<https://asana.com>), permet une visualisation des tâches par listes (à gauche), par colonnes (à droite)

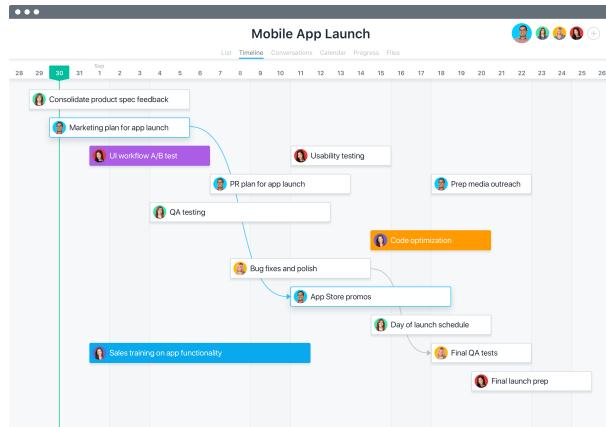


Figure 3 - Diagramme de Gantt sur Asana

Dans l'application que vous allez réaliser, vous pourrez faire preuve d'imagination et ajouter des fonctionnalités non demandées.

Ce projet a pour objectif de développer un logiciel permettant de gérer des tâches personnelles pouvant être dépendantes les unes des autres et de visualiser l'organisation de la progression de leur réalisation dans une interface graphique restant simple d'utilisation et ergonomique.

Objectifs à atteindre

L'application à développer doit pouvoir

- **créer des tâches** qui peuvent dépendre d'autres tâches, par exemple une tâche inclue dans une autre tâche ou encore une tâche ne pouvant démarrer que si une autre est terminée ;
- **archiver des tâches** une fois celles-ci réalisées ;
- **visualiser l'ensemble des tâches** sous forme d'un **bureau** avec des colonnes regroupant des tâches, une tâche pouvant être glissée déposée d'une colonne à une autre. La description d'une tâche (et de ses sous-tâches) pourra être visualisée après sélection avec la souris ;
- **visualiser l'ensemble des tâches** sous forme de **listes** et de sous-listes si le déploiement des sous-tâches est demandé ;
- **générer le diagramme de Gantt** d'une sélection de tâches afin de visualiser la chronologie de la réalisation de celles-ci.

Le développement se fera en **JavaFx** en respectant l'architecture MVC et en utilisant des patrons de conceptions adaptés.

Un **trello** devra être utilisé pour chaque groupe qui permettra de suivre la progression du travail et la répartition des tâches, ceci pour l'ensemble du projet.

Un dépôt **github** devra être créé pour chaque projet et les différents documents et codes sources devront y figurer.

Déroulement de la SAE

La SAE, d'une **durée de 63 heures** par étudiant, se déroulera sur 4 semaines : les semaines 49 (8h) et 50 (16h) et 51 (16h) de 2025 ainsi que la semaines 2 (23h) de 2026. Les étudiants travailleront par **groupes de 4 ou 3** (groupe de 26 = 5*4 + 2*3) qui auront été constitués dans le cours de QDev. Des explications sur l'application javaFx à réaliser seront données dans le dernier cours de **QDev**. Des explications sur l'utilisation de Trello seront données dans le cours d'**Analyse**.

A. PARTIE ANALYSE

Semaine 49 et début 50 – 12h + 2h, dépôt d'un document d'étude préalable à la fin de la seconde séance de SAE de la semaine 50 et présentation de l'étude préalable aux enseignants d'analyse (avec éventuellement la présence de l'enseignant de Qualité de Développement du groupe) la séance suivante :

Chaque groupe fera **l'étude préalable du projet** et rédigera un **document** dans lequel figureront :

- La liste des fonctionnalités ;
- les cas d'utilisation de l'application et les diagrammes de cas d'utilisation si cela est nécessaire, des descriptions textuelles, des DSS et/ou scénarios pourront les compléter,
- un diagramme d'activités de l'application et un diagramme d'état de l'objet central de l'application (la tâche) ;
- conception : un diagramme de classe en y précisant les patrons de conception et d'architecture que vous prévoyez d'utiliser ;
- une maquette de l'application ;
- le planning des 6 itérations prévues et les objectifs de chacune (en terme de cas d'utilisations) avec identification des risques.

Le trello du groupe devra être rempli selon le planning prévu.

Evaluation en analyse : soutenance de 10 minutes par groupe et évaluation du trello et du document d'étude préalable.

B. PARTIE QUALITE DU DEVELOPPEMENT

Semaine 50, 51 et 2 – 6*8h + 1h de DS. Développement de l'application en javaFx par itérations successives de 8h (6 itérations).

Evaluation en Qualité du Développement: selon le barème figurant dans le fichier *BaremeSAE_3.01* sur arche : la note obtenue tiendra compte des réunions effectuées à chaque itération (cf. fichier *reunion_iteration* sur arche), de la soutenance finale et d'une épreuve individuelle sur feuille qui se déroulera le **jeudi 8 janvier**.

B.1. DOCUMENTS ATTENDUS

Un répertoire **Sources** et un répertoire **Documents** devront figurer dans le dépôt git du projet. Pour chaque itération, un sous répertoire devra apparaître dans le répertoire **Documents** (par exemple *Documents/Iteration3* pour la troisième itération) qui comportera :

- Un fichier *fonctionnalitesCU.txt* décrivant les fonctionnalités et les cas d'utilisations développés en précisant le rôle des membres du groupe;
- un diagramme de classe et des diagrammes de séquence pour chaque fonctionnalité décrivant le fonctionnement de votre application, vous indiquerez aussi vos choix de conception, si des patrons sont mis en œuvre et si ces choix sont conformes à votre étude préalable ;
- un fichier bilan présentant un état de votre version (bug, validation, ...);
- un tag de version finalisant cette version.

Dans le répertoire **Sources** :

- Un code mis à jour qui fonctionne conformément à vos choix de conception ;
- un ensemble de classes de tests qui vérifient les critères de validation des différentes fonctionnalités ;
- pas de document inutile (.class, .project, fichiers temporaires).

De plus, pour chaque itération le trello devra être mis à jour et évoluer pendant l'itération. Chaque itération de 8h de la SAE 3.01 devra fonctionner sur ce principe.

B.2. CONSIGNES

Fonctionnalités

- Choisissez vos fonctionnalités par priorité
- Ne faites que ce qui est utile pour les fonctionnalités de l'itération en cours
- N'ajoutez pas d'attributs ni de méthodes inutiles
- Ne pensez pas à l'itération suivante, vous y réfléchirez en temps utile
- Ne passez pas à de nouvelles fonctionnalités tant que celles en cours ne sont pas validées

Conception et code java

- Prenez le temps de réfléchir avant d'écrire du code
- Faites des classes simples (5 attributs max, limiter le nombre de lignes)
- Pas de copier-coller (réfléchir et changer la conception si besoin)
- Testez et faites des push régulièrement

B.3. CONTENU D'UNE ITERATION (8h)

(30 min) Choisir les fonctionnalités à ajouter

- Préciser ce qui doit être fait
- Adapter les critères de validation et les tests à réaliser

(1h30) Concevoir l'application

- Reprendre la conception de l'itération précédente
- Diagramme de classe, DSS ou scénarios des nouveautés
- Ajouter les éléments utiles
- Partage des tâches à partir du diagramme de classes

(4h30) Réalisation

- Écriture du code et des tests

(1h30) Intégration et validation

- Validation des tests, debuggage, mise en forme, tag