

Rapport itération 3:

Pour le diagramme de classe les classes qui sont coloriés en vert sont celles qu'on a faites durant l'itération 3 (ou modifié).

affichage en liste (Maxime):

Nous nous sommes basés sur l'affichage en bureau que l'on a mis verticalement et agencé un peu différemment, nous avons aussi fait un système pour cacher ou montrer les tâche que possède une colonne car nous trouvions cela plus pratique et esthétique pour l'utilisateur.

affichage Bureau(Hugo):

Nous avons fait comme l'énoncé nous disait, pour le moment un affichage avec deux colonne qui contient des tâches, l'utilisateur peut ajouter des tâches dans la colonne voulu, il peut aussi cliquer sur les tâche pour les voir plus en détail. Les colonnes sont les unes à côté des autres contrairement à l'affichage en liste ou elles sont l'une en dessous de l'autre.

Visuel(Maxime et Hugo):

Pour le visuel choisi, nous sommes parties sur un style plutôt violet que nous trouvions agréable pour les yeux en plus d'être passe partout, nous sommes rester sur un même style pour toutes les pages pour ne pas agresser visuellement l'utilisateur. Pour les boutons afficher tâche, nous avons décidé de mettre des logo que nous trouvions plus jolie et en accord avec notre visuel.

Fonctionnalité changer d'affichage (Jawad, Timothée):

Pour que l'utilisateur change d'affichage c'est très simple il lui suffit de sélectionner l'affichage qu'il souhaite dans la combobox en haut à droite et l'affichage voulu par l'utilisateur va s'afficher.

Pour la partie code, nous avons ajouté à notre vue principal un attribut affichage qui est un attribut de type StratégieVisuel. Cette classe (qui hérite d'un pane) a pour l'instant deux classes enfant : VisuelBureau et VisuelListe chacun réalisant le visuel accordé à son nom. Nous avons décidé de partir sur un patron de type Stratégie pour les raisons suivantes:

- Si plus tard nous voulons faire d'autre affichage il est facile d'en ajouter un nouveau, juste faire une classe qui hérite de stratégieVisuel

- Le code est beaucoup plus maintenable comme ceci

- Nous avons décidé de faire de stratégie un Pane plutôt que de lui faire renvoyer un affichage avec une méthode car quand l'affichage est mis à jour, il suffit de mettre la stratégie à jour (Le reste de la vuePrincipale n'est jamais modifié) c'est donc plus économe (en mémoire) que de tout détruire et de tout redessiner. Avec cette méthode nous pouvons aussi changer directement ce qui a été modifié par exemple une tâche rajouté au lieu de tout redessiner nous allons juste redessiner la tâche elle même, chose que l'on ne pourrait pas faire si stratégie renvoyait un affichage au lieu d'être un affichage lui même.

Pour changer l'attribut de type strategieVisuel, nous avons fait un controller lié à la comboBox qui va changer directement dans VuePrincipal l'attribut Stratégie et l'affichage en conséquence.

Toutes les fonctionnalités ont été testé et debugué avant la fin de l'itération.