

UE622 - Génie Logiciel

Auteurs
RULLAC Éloïse, FAUCON Léo

May 5, 2025

- **En groupe** : Cahier des charges, axe fonctionnel, statique, dynamique.
- **Éloïse** : IHM, UI, contrôleurs (front-end).
- **Léo** : Communication client-serveur (back-end).

Graphique de Gantt Projet GL

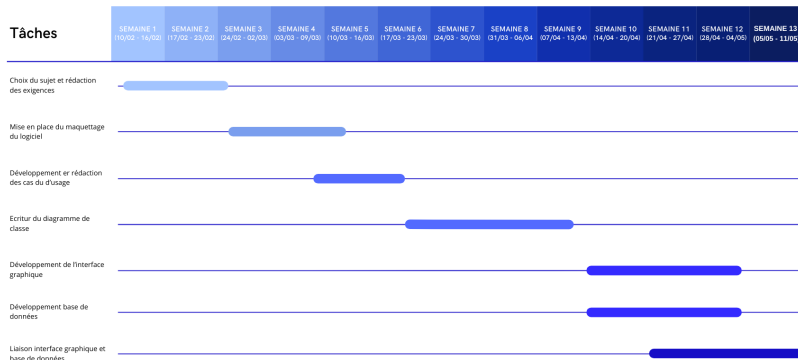


Figure 1: Graphique de Gantt

Exemple d'exigences

- AU010 : Le système doit permettre à l'utilisateur de se connecter via une adresse mail et un mot de passe.
- AU020 : Le système doit permettre à l'utilisateur de gérer son compte (adresse mail, mot de passe, nom, prénom).

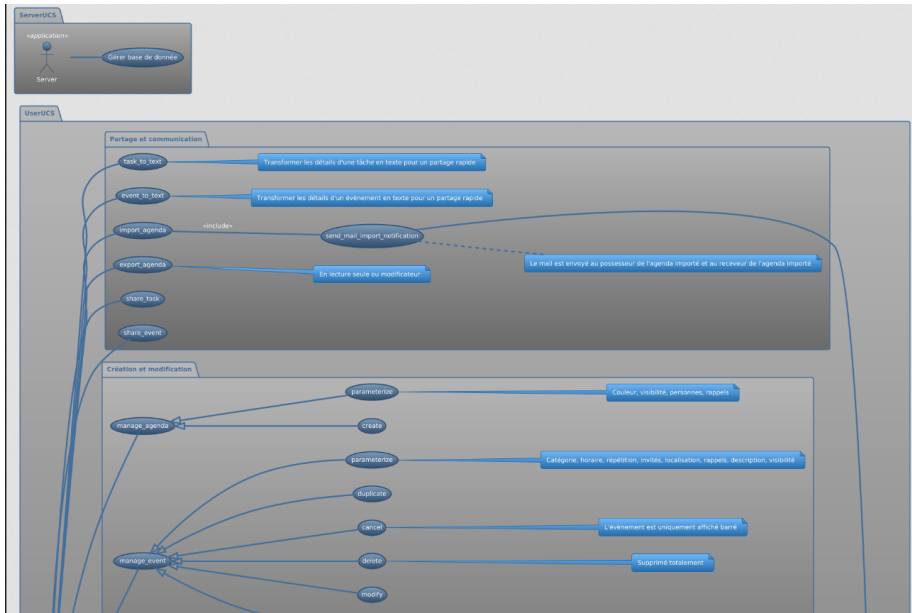


Figure 2: Cas d'usages (1/2)

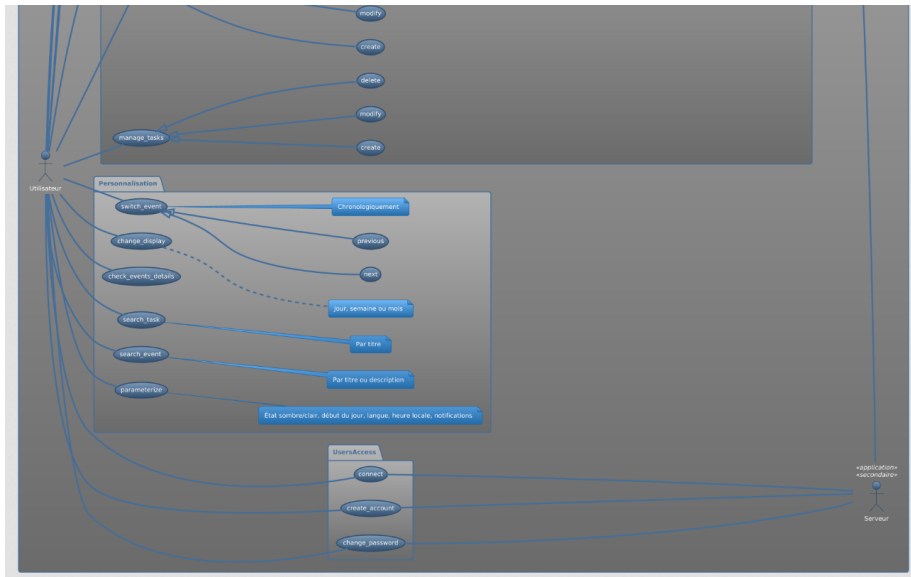
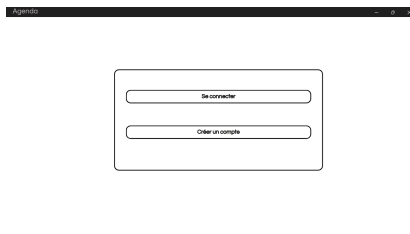
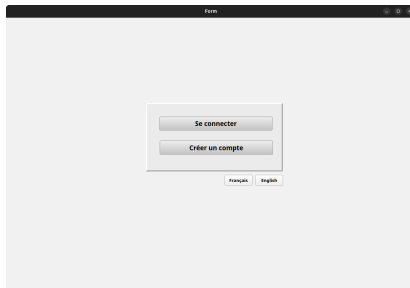


Figure 3: Cas d'usages (2/2)



(a) Maquettage



(b) home_page.py

Agenda

Connectez vous

Adresse mail

Mot de passe

Rechercher connecté

(a) Maquettage

Fern

Connectez-vous

Adresse mail

Mot de passe

Valider

Retour tester connecté

(b) login_page.py

A wireframe of a web form titled "Créer votre compte" (Create your account). The form is centered within a window titled "Agenda". It contains three input fields stacked vertically, each with a label above it: "Adresse mail", "Mot de passe", and "Confirmer mot de passe". The form is enclosed in a simple rectangular border.

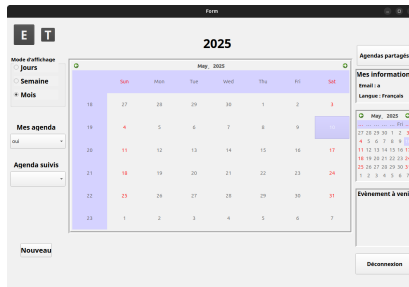
(a) Maquettage

A mockup of the same "Créer votre compte" form, now rendered with a more realistic design. It is displayed in a window titled "Form". The form has a light gray background and a thin border. The input fields are now styled with a light blue border and a white background. The labels "Adresse mail", "Mot de passe", and "Confirmer mot de passe" are in a bold, sans-serif font. At the bottom of the form, there are two buttons: "Retour" (Return) on the left and "Valider" (Validate) on the right. Below the "Valider" button, there is a small link that says "Je suis déjà connecté" (I am already connected).

(b) sign_in_page.py

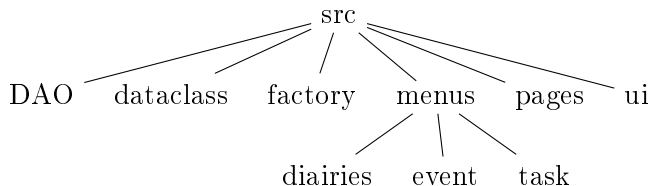


(a) Maquettage



(b) event_page.py

Arborescence du projet



- Communication client-serveur: socket.
- Communication serveur-BDD : sqlite3.

- **1**: création de compte
- **2**: connexion à un compte
- **3**: requête du client avec comme spécification la catégorie *requestType*
- **4**: retour serveur positif
- **5**: retour serveur négatif avec message d'erreur dans la catégorie *errmsg*

- **"createAgenda"**: création d'un agenda
- **"updateAgenda"**: mise à jour d'un agenda
- **"getAgendaList"**: récupération de la liste des agendas pour un utilisateur
- **"deleteAgenda"**: suppression d'un agenda
- ...

```

1 class Request:
2     def __init__(self, query: str, func: Callable[[sqlite3.Cursor], str]):
3         self.query = query
4         self.func = func
5
6     def send(self, *args) -> bytes:
7         return self.func(con.execute(self.query, args)).encode()
8
9 def createAgenda(cur: sqlite3.Cursor) -> str:
10     return "{\\"data\\":{\\"agenda_id\\":\" + str(cur.lastrowid) + \",\\"op\\":4}"
11
12 requestType_to_query: dict[str, Request] = {
13     "createAgenda": Request(
14         "insert into agenda (user_id, name) values (?, ?);",
15         createAgenda
16     )
17 }
18
19 # if m_json["op"] == 3:
20 try:
21     data: bytes =
22         requestType_to_query[m_json["requestType"]].send(*m_json["data"].values())
23 except sqlite3.Error as e:
24     conn.sendall(("{\\"errmsg\\":\"\" + str(e) + "\",\\"op\\":5}").encode())
25 else:
26     conn.sendall(data)

```

```

1 class AgendaDAO:
2     def delete(self, agenda: Agenda):
3         self.dbcom.sendall({
4             "data":{
5                 "agenda_id": agenda.id
6             },
7             "requestType": "deleteAgenda",
8             "op": 3
9         })
10        return self.dbcom.recv()
11
12    def get_list(self, user: User) -> list[Agenda]:
13        self.dbcom.sendall({
14            "data": {
15                "user_id": user.id
16            },
17            "requestType": "getAgendaList",
18            "op": 3
19        })
20        r: list[Agenda] = []
21        data: dict = self.dbcom.recv()
22        for agenda in data["data"]["agendaList"]:
23            r.append(Agenda(agenda["id"], agenda["name"]))
24
25        return r

```

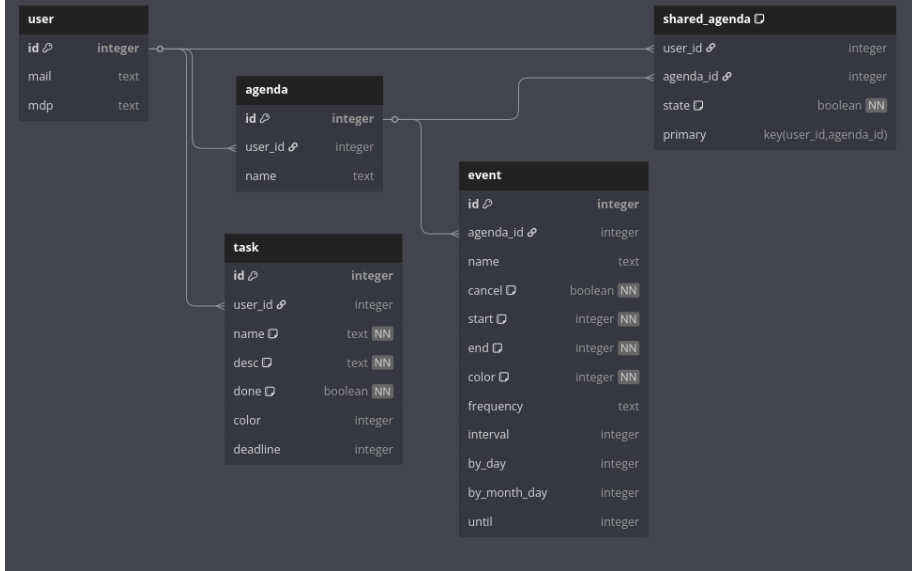



Figure 8: MEA