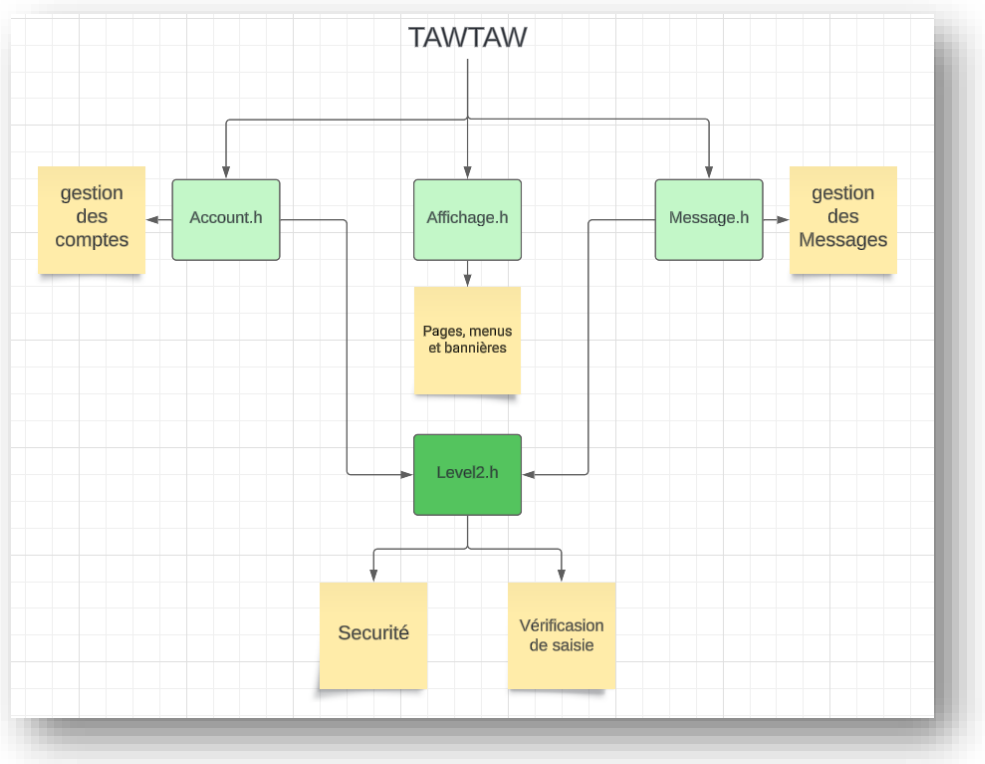
RAPPORT PROJET ATELIER PROGRAMMATION

UNIVERSITE DE SOUUSE
ISITCOM
RAYEN DRIRA
1-IOT-1
2023/2024

TABLE DE MATIÈRES

1. Architecture de l'application :



1.1. Gestion des comptes (Account.h)

1.1.1. Connection :

Permet de se connecter à l'application.

1.1.2. Create account :

Permet de créer un compte sur l'application.

1.1.3. Modification :

Permet de modifier les informations de l'utilisateur (login, mot de passe).

1.1.4. Supprimer_compte :

Permet de supprimer le compte de l'utilisateur.

1.1.5. Gestion_fichier, add_line, supprimer_account, modify_messages:

Permet d'effectuer des opérations liées aux fichiers (ajouter une ligne, supprimer une ligne, modifier une information).

1.2. Gestion des Messages (Message.h)

1.2.1. SendMessage:

Permet d'envoyer un message a un autre Utilisateur.

1.2.2. Viewbox:

Permet de consulter la boite de messagerie de l'utilisateur.

L'utilisateur peut filtrer les messages par :

1. Voir tous.
2. Voir par sujet.
3. Voir une Conversation avec un seul utilisateur.
4. Voir l'INBOX.
5. Voir l'OUTBOX.
6. Voir par date.

1.2.3. SendMessage:

Permet d'envoyer un message a un autre Utilisateur.

1.2.4. deleteMessage:

Permet de supprimer un ou des messages de la boite de messagerie.

L'utilisateur peut filtrer les messages comme « SendMessage » avec une option de plus :

Supprimer un seul message (par nombre).

1.2.5. count_message,AFFICHE_MESSAGE:

Permet de compter le nombre de messages dans la boite de messagerie.

1.2.6. AFFICHE_MESSAGE:

Permet d'Afficher un Message sous une structure claire.

1.3. Interface (Affichage.h)

Ensemble des fonctions d'affichage qui affiche 2 Bannières, 6 Menus, 10 Pages titres et une fonction d'affichage de la page « Plus sur l'application ».

1.4. (Level2.h)

1.4.1. Saisie:

1. « saisir-account » Permet de saisir le login et le mot de passe de l'utilisateur.
2. « create-Filename » Permet de créer le nom de la boîte de messagerie de l'utilisateur à partir de son login.

1.2.6. Control de Saisie:

1. « existe-login » Permet de vérifier si le login est unique.
2. « secure_login » Permet de vérifier si le login est écrite correctement.
3. « secure_mot_de_passe » Permet de vérifier si le mot de passe est fort.
4. « secure_content » Permet de vérifier si le contenue d'un message est écrit correctement.

1.2.6. Sécurité

1. « hash » Permet de Sécuriser le mot de passe (caché dans les fichiers).
2. « verify_password » Permet de vérifier i le mot de passe entré par le clavier est le même que celui de l'utilisateur.

2.Structures de données :

2.1. Compte :

Structure pour représenter un utilisateur :

Login : chaine de caractères (un seul mot).

Mot_de_passe : chaine de caractères.

Filename : chaine de caractères (Login+ ''.txt'').

2.2. Lettre :

Structure pour représenter un Message :

Nbre : entier (numéro du message).

Source : chaine de caractères (nom d'Utilisateur).

Destinataire : chaine de caractères (nom d'Utilisateur).

Sujet : chaine de caractères (un seul mot).

Content : chaîne de caractères (le contenu du message).

Month : entier (Le mois lorsque le message a été envoyé).

Day : entier (Le jour lorsque le message a été envoyé).

Hour : entier (L'heure lorsque le message a été envoyé).

Min : entier (La minute lorsque le message a été envoyé).

Pour utiliser automatiquement l'heure du système, vous devez importer la bibliothèque **"**time.h**"** et utiliser ce code :

```
//Get the time
time_t currentTime;
struct tm *localTime;
currentTime= time(NULL);
localTime =localtime(&currentTime);

newMessage.month=(localTime->tm_mon)+1;
newMessage.day=localTime->tm_mday;
newMessage.hour=localTime->tm_hour;
newMessage.min=localTime->tm_min;
```

2.Interface :

L'application utilise un système de pages et sous-pages ainsi que des menus pour la navigation. Elle est aussi dotée de bannières et ne peut être exécutée que sur le terminal.

Elle utilise :

1. Le caractère "_" pour créer des lignes et des cadres.
2. Des couleurs pour différencier entre les différents types de texte :
    - Blanc : Message d'entrée (INPUT).
    - Rouge : Message d'erreur.
    - Jaune : Message de réussite.
    - Noir : Pour les cadres, les lignes et les messages d'information générale.

Pour utiliser les couleurs, il faut déclarer chaque couleur comme suit :

```
#define ANSI_COLOR_RED     "\x1b[31m"
#define ANSI_COLOR_BLACK   "\x1b[90m"
#define ANSI_COLOR_BLUE    "\x1b[34m"
#define ANSI_COLOR_YELLOW  "\033[1;33m"
#define ANSI_COLOR_RESET   "\x1b[0m"
```

3.Sécurité :

L'application intègre plusieurs mesures de sécurité pour protéger les comptes des utilisateurs. Parmi ces mesures, elle utilise le hachage de mots de passe (fonction hash) pour stocker de manière sécurisée les informations d'identification des utilisateurs. De plus, elle vérifie l'identité des utilisateurs lorsqu'ils tentent de modifier leur compte ou de le supprimer (fonction verify_password). Ces vérifications garantissent que seuls les utilisateurs légitimes peuvent accéder aux fonctionnalités sensibles de l'application, renforçant ainsi la sécurité et la confidentialité des données des utilisateurs.

Pour Hacher les mots de passe l'application utilisent l'algorithme djb2 :

```
//Hash
unsigned long hash(compte* Tawtaw) {
    unsigned long hash = 5381;
    int c;
    char *mot_de_passe = Tawtaw->mot_de_passe;
    while ((c =*mot_de_passe++)) {
        hash = ((hash << 5) + hash) + c; // Hash * 33 + c
    }
    return hash;
}
```