

Interface et Multimédia  
**IFT-215**

Travail Pratique 5

Amandine Fouillet - 14 130 638  
Frank Chassing - 14 153 710

7 avril 2015



# Table des matières

<b>Introduction</b>	<b>4</b>
<b>1 Présentation générale de l'interface</b>	<b>4</b>
1.1 Choix de l'interface . . . . .	4
1.1.1 Objectifs réalisés . . . . .	4
1.1.2 Respect des lois de Gestalt . . . . .	4
1.1.3 Choix des composants . . . . .	4
1.2 Aide à l'utilisation . . . . .	5
1.2.1 Gestion des erreurs . . . . .	5
1.2.2 Gestion de l'aide . . . . .	5
<b>2 Explication du code</b>	<b>5</b>
2.1 Accueil, Connexion et Inscription . . . . .	5
2.2 Base de la navigation . . . . .	5
2.3 Implémentation des fonctionnalités . . . . .	6
2.3.1 Similarités . . . . .	6
2.3.2 Messagerie et Contacts . . . . .	6
2.4 Aide et paramètres . . . . .	6
<b>3 Tests de notre application</b>	<b>6</b>
3.1 Protocole . . . . .	7
3.2 Résultats . . . . .	7
<b>4 Amélioration</b>	<b>8</b>
4.1 Ce qu'il reste à implémenter . . . . .	8
4.2 Idées nouvelles . . . . .	8
<b>Conclusion</b>	<b>8</b>

# Introduction

Après avoir réfléchi sur la conception de notre interface par le biais des Tps précédents, nous avons implémenté notre interface en suivant les objectifs et les contraintes que nous nous étions fixés. Dans un premier temps, ce rapport présente l'interface générale de notre application et les différents éléments d'interface qu'elle possède. Dans un second temps, il met en avant les phases de tests que nous avons été amené à réaliser pour obtenir différents retours. Enfin, nous verrons comment il est possible d'envisager des améliorations de notre application et nous expliciterons le code qui a permis d'implémenter notre application.

## 1 Présentation générale de l'interface

### 1.1 Choix de l'interface

#### 1.1.1 Objectifs réalisés

Pour réaliser notre interface nous nous sommes basés sur les objectifs que nous avons établis dans les Tps précédents. Nous voulions créer une interface permettant d'accueillir toutes les fonctionnalités de communication entre un étudiant et sa famille. Notre but principal était de réunir l'ensemble de ces fonctionnalités et de pouvoir naviguer rapidement entre elles. Dans notre interface finale, nous avons donc repris le menu vertical permettant cette navigation rapide. Nous retrouvons également les informations sur chacun des contacts que l'on a ajouté. Ceci permet à l'utilisateur de connaître l'actualité de ses contacts en temps réel. Cependant la fonction principale reste la communication. L'utilisateur a accès à une liste de communication et peut entamer à tout moment une conversation écrite, audio ou vidéo avec un ses contacts.

#### 1.1.2 Respect des lois de Gestalt

En faisant une premier ébauche de notre interface dans le Tp précédent, nous avons fait attention à ce que celle-ci respecte les lois de Gestalt. Nous retrouvons beaucoup la notion de fermeture dans notre interface par le biais de zone créée avec des contours noirs. Nous retrouvons également la notion de similarité et de proximité permettant de regrouper les éléments allant ensemble, notamment au niveau de la liste des communications et la liste des contacts, où chaque contact ou communication est présenté de la même façon. Nous pouvons également voir que plusieurs éléments respectent la loi de continuité, comme par exemple le menu vertical où les bulles suivent une même colonne.

#### 1.1.3 Choix des composants

Pour réaliser notre interface, nous avons utilisé plusieurs composants. Pour les fenêtres d'inscription et de connexion, nous avons implémenter des champs textes libre permettant à l'utilisateur de rentrer les informations qu'il souhaite. De plus, pour le champ réservé au mot de passe, nous avons utilisé un champ texte spécialisé permettant de ne pas percevoir le mot de passe tapé. En ce qui concerne la liste des contacts et des communications, nous avons choisi d'utiliser des listes déroulantes avec ascenseur permettant ainsi d'avoir une fenêtre assez petite (taille Smartphone) et de placer de nombreux contacts ou de créer de nombreuses conversations. Les différentes boutons de notre application sont des images ayant un évènement particulier. Nous voulions personnaliser davantage notre application et nous ne pouvions donc utiliser de simples boutons qui ne nous permettaient pas d'obtenir le rendu que l'on voulait.

## 1.2 Aide à l'utilisation

Nous avons implémenté plusieurs éléments d'interface qui permettent d'avertir les éventuelles erreurs que peut commettre l'utilisateur. Nous avons également ajouté des fonctionnalités permettant d'aider l'utilisateur à utiliser simplement notre application.

### 1.2.1 Gestion des erreurs

Afin que l'utilisateur comprenne le sens de l'erreur qu'il a commise, nous avons décidé de l'avertir en utilisant des boîtes de dialogue explicite. Par exemple, dans la fenêtre permettant à l'utilisateur de s'inscrire, celui-ci doit absolument remplir les champs "Pseudo" et "Mot de passe" pour que l'inscription soit valide, sinon une boîte de dialogue l'avertit que le champ "Pseudo" ou "Mot de passe" est vide. De plus, le champ "Email" devra correspondre à une adresse Email valide sinon l'utilisateur verra une boîte de dialogue s'ouvrir l'avertissant que le format de l'Email est invalide, et cette boîte de dialogue donnera également à l'utilisateur la forme de l'Email attendue. En ce qui concerne la connexion à l'application, si les informations rentrées n'existent pas dans la base de données, une boîte de dialogue s'ouvrira indiquant à l'utilisateur que les informations de connexion ne sont pas exactes.

### 1.2.2 Gestion de l'aide

Plusieurs éléments ont été intégrés à l'application permettant d'obtenir de l'aide à tout moment. Tout d'abord, nous avons ajouté un bouton d'aide dans chaque fenêtre. Ce bouton ouvre une nouvelle fenêtre recueillant diverses informations permettant d'aider l'utilisateur à comprendre et utiliser les fonctions dans la fenêtre où il se trouve. Ce bouton existe à tout moment et empêche l'utilisateur de rester bloqué sur l'application.

Dans un second temps, nous avons décidé d'utiliser des `ToolTipText`. Un `ToolTipText` correspond au texte suivant notre curseur lors du survol d'un élément. Nous avons implémenté ces éléments sur les items qui ne sont pas explicites comme par exemple les boutons ayant seulement un icône. Bien que nous ayons choisi des icônes plus ou moins explicites, nous avons renforcé la compréhension de ces boutons en ajoutant des `ToolTipText`.

## 2 Explication du code

### 2.1 Accueil, Connexion et Inscription

À l'ouverture de l'application, l'utilisateur est en présence d'une première fenêtre que nous avons nommée *PanelAwaï*. Cette fenêtre se veut simple et est composée de seulement trois boutons : connexion, inscription et à propos. Chacun de ces trois boutons mènent vers un panel ayant respectivement pour nom *PanelConnexion*, *PanelInscription* et *PanelAPropos*. Afin de simuler la connexion et l'inscription d'un utilisateur, nous avons mis en place une classe *Utilisateur* et une classe *UtilisateurManager*. Ainsi, avec des identifiants pré-enregistrés dans un tableau, on peut simuler une connexion et une inscription en enregistrant de nouveaux identifiants dans ce tableau le temps d'un essai. Une fois la connexion établie, l'utilisateur atterrit sur le menu central de l'application. Dans le code, ce menu est implémenté dans la classe *PanelMenu*.

### 2.2 Base de la navigation

Afin de mettre en place une navigation entre les différentes fonctionnalités de notre application et respecter notre modélisation initiale (voir TP3), nous avons mis en place une base qui se trouve dans

la classe *PanelBase*. Cette classe construit les deux barres horizontales et verticales permanentes de l'application et accueille un emplacement vide afin d'y intégrer les différentes fonctionnalités. C'est aussi elle qui gère les événements pour passer d'une fonctionnalité à l'autre : lors du clic sur une des icônes de la barre verticale, l'emplacement vide est remplacé par les données de la fonctionnalité.

## 2.3 Implémentation des fonctionnalités

### 2.3.1 Similarités

Une classe a été créée pour chaque fonctionnalité de notre application : *PanelCommunication*, *PanelPhotos*, *PanelCalendrier*, *PanelDepenses*, *PanelCarte* et *PanelContacts*. Comme nous l'avons expliqué précédemment, chacun de ces panels est amené à être inclus dans l'emplacement vide de la classe *PanelBase*. Cependant, afin de respecter leurs rôles respectifs ces panels sont tous construits différemment. La carte et les dépenses n'ont pas été implémentés, les photos et le calendrier sont implémentés partiellement tandis que les contacts et la communication sont pratiquement fonctionnels.

### 2.3.2 Messagerie et Contacts

Afin de mettre en place le système de messagerie et des contacts plusieurs autres panels ont dû être réalisés. Pour la messagerie, la classe *PanelListe* a été créée afin de construire l'affichage de chaque discussion avec un contact différent. Il existe ensuite trois classes afin d'accéder aux différents modes de communication : la classe *PanelChat* construit l'échange de messages textuels entre un utilisateur et un contact, la classe *PanelAudio* simule un appel téléphonique et la classe *PanelVideo* simule un appel vidéo. Pour les contacts, de même que pour la messagerie, la classe *PanelContact* a été créée afin de construire l'affichage d'un contact dans une liste qui est construite dans *PanelContacts*. À la sélection d'un contact dans cette liste, on accède à ses informations qui sont codées dans la classe *PanelBaseContact* qui est implémentée de la même manière que la classe *PanelBase* mais qui donne accès, cette fois-ci, aux informations d'un contact et non de l'utilisateur.

## 2.4 Aide et paramètres

Sur chacune des fenêtres de notre application, il est possible d'accéder à l'aide ou aux paramètres de la fonctionnalité. Pour implémenter cela, nous avons créé deux classes *PanelAide* et *PanelParametre* qui jouent leur rôle en affichant un texte différent en fonction de l'endroit où ils ont été appelés. Par exemple, si l'utilisateur a besoin d'aide dans la fonctionnalité de dépenses, la fenêtre aide qui va s'ouvrir contiendra un texte d'information sur le panel dépenses.

## 3 Tests de notre application

Afin de mettre en pratique les enseignements vus en cours et surtout tester l'utilisabilité de notre application, nous avons mis en place un protocole de test. L'idéal aurait été de tester notre application sur une population large et sur plusieurs générations mais nous avons dû improviser avec les personnes disponibles : deux camarades d'une vingtaine d'années qui n'étudient pas dans le parcours informatique. Cette partie décrira le protocole mis en place pour la réalisation des tests ainsi que les résultats obtenus.

### 3.1 Protocole

Les deux tests se sont déroulés de la même façon. Après leur avoir expliqué le principe de notre application, nous avons installé les deux volontaires devant un ordinateur avec une liste de tâches à réaliser. La liste des tâches qu'ils avaient à réaliser est présentée ci-dessous, avec en dessous de chacune des tâches le niveau de difficulté que nous avons estimé auparavant ainsi que ce que nous attendions de l'utilisateur lors de la réalisation de la tâche.

- S'inscrire avec les identifiants de votre choix  
Facile - *L'utilisateur comprend l'interface d'accueil et les modalités d'inscription.*
- Se connecter avec vos identifiants  
Facile - *L'utilisateur se sert de ses nouveaux identifiants pour se connecter.*
- Se rendre dans l'onglet communication  
Facile - *L'utilisateur navigue intuitivement et comprend les icônes du menu*
- Envoyer un message à maman  
Moyen - *L'utilisateur peut communiquer avec un contact donné sans grande difficulté.*
- Faire un appel video avec Nina  
Difficile - *Avec les seules indications iconiques, l'utilisateur parvient à lancer une conversation vidéo.*
- Consulter ses photos de New York  
Moyen - *L'utilisateur arrive à naviguer dans l'interface et peut se rendre sur ses photos.*
- Ajouter un contact  
Difficile - *L'utilisateur parvient à trouver le moyen d'ajouter un contact.*
- Aller consulter les informations du contact Clément  
Moyen - *En suivant les icônes, l'utilisateur arrive à accéder aux informations d'un contact.*
- Aller consulter le calendrier de maman  
Difficile - *L'utilisateur a compris la différence entre son calendrier et les calendriers de ses contacts.*
- Consulter l'aide  
Facile - *L'utilisateur est capable de trouver l'icône d'aide sur toutes les pages de l'application.*
- Se déconnecter  
Facile - *L'utilisateur est capable de trouver le bouton de déconnexion rapidement.*

Outre la compréhension globale de l'interface et la réalisation rapide des tâches demandées, nous voulions surtout observer la capacité de l'utilisateur pour comprendre la subtilité de notre application : savoir faire la différence entre les informations qui sont propres à l'utilisateur et les informations qui sont propres à ses contacts. Une fois les tâches réalisées, nous avons pu échanger avec nos deux testeurs afin de recueillir leurs impressions et leurs remarques ainsi que leur poser quelques questions.

### 3.2 Résultats

Nos deux testeurs sont parvenus sans grande difficulté à réaliser toutes les tâches que nous leur avons demandé. blabla.. Malgré que ces tests aient été pensés pour contrôler l'utilisabilité de notre application, nous sommes conscients qu'il ne sont pas suffisants pour contrôler la qualité de notre interface. En effet, nos deux testeurs étant étudiants, ils font parti d'une génération qui a grandi avec les nouvelles technologies et ils sont habitués à découvrir et s'habituer à de nouvelles interfaces. Cela rend la tâche plus facile pour eux et fausse nos tests, même si cela les rend aussi exigeants quant à la qualité de navigation et d'utilisation de l'application. De plus, les tests ont été réalisés sur un ordinateur alors qu'une application avec des objectifs de communication comme la notre est destinée à être développée en tant qu'application mobile. Or, les contraintes et les exigences ne sont

pas les mêmes sur un ordinateur avec un clavier et une souris que devant un téléphone portable tactile.

## **4 Amélioration**

### **4.1 Ce qu'il reste à implémenter**

Les fonctionnalités calendrier, photos, carte et dépenses La base de données Le serveur

### **4.2 Idées nouvelles**

## **Conclusion**