



## TP2 – UI/UX

### Objectifs

Au terme de ce laboratoire, l'étudiant :

- Sera familiarisé avec les logiciels utilisés pour concevoir les interfaces utilisateur (draw.io) ;
- Sera capable d'évaluer une proposition d'interface utilisateur imparfaite.
- Sera capable de proposer et justifier des interfaces utilisateur sur base d'un diagramme de cas d'utilisations ou d'un discours client décrivant une situation réelle.

### Durée

2 séances (0.5+1+0.5) de 2h en présentiel pour réaliser les exercices (à éventuellement terminer à domicile pour la séance suivante).

### Modalités

Exercice 1 : Réalisé en commun pour permettre à tout le monde de se familiariser avec le logiciel draw.io et d'appréhender les règles de conception d'UI.

Exercice 2 : Travail individuel visant à évaluer une proposition d'interface utilisateur correspondant à des cas d'utilisation.

Exercice 3 : Travail en groupe avec accompagnement de l'enseignant en charge du laboratoire. Au terme du laboratoire (ou avant la séance suivante), un étudiant déposera sur HELMo Learn un document PDF reprenant les interfaces utilisateur créées par son groupe.

Les travaux des étudiants seront analysés et critiqués en commun lors de la séance de laboratoire suivante.

### Exercice 1 : eBulletin

#### Consignes

On demande de proposer une UI ergonomique répondant aux cas d'utilisation repris dans l'énoncé (et qui respectera les critères listés au cours théorique).

#### Énoncé

On veut développer un module permettant aux étudiants de consulter leurs notes pour les UE figurant à leur PAE.

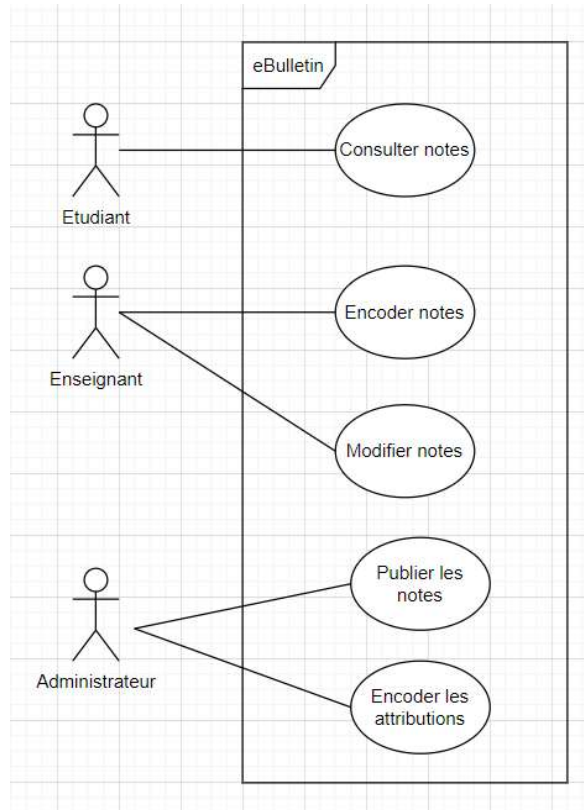
Les notes des UE sont représentées par des nombres entiers entre 0 et 20.

Les codes suivants sont utilisés pour les cas particuliers :

- PP : l'étudiant ne s'est pas présenté à son examen
- PR : l'étudiant a signé son examen sans défendre ses chances
- FR : l'étudiant a écopé d'une sanction disciplinaire pour fraude lors de l'examen

On considère ici que les PAE des étudiants sont encodés via une procédure semi-automatisée qui est en-dehors du périmètre de cet exercice.

Voici le diagramme des cas d'utilisation pressentis :



### **Consulter les notes**

Événement déclencheur : Un étudiant veut consulter ses notes pour une session.

Description : L'étudiant peut voir un tableau avec ses notes par UE.

Acteur(s) impliqué(s) : Etudiant

Préconditions : L'étudiant est authentifié ; Le PAE de l'étudiant est encodé ; Les notes sont publiées

Postconditions : /

### **Encoder notes**

Événement déclencheur : Un enseignant veut encoder ses notes pour une session.

Description : L'enseignant choisit l'UE pour laquelle il désire encoder les notes et il encode les notes. Il doit être en mesure de voir facilement les étudiants pour lesquels aucune note n'est encodée.

Acteur(s) impliqué(s) : Enseignant

Préconditions : L'enseignant est authentifié ; Le PAE des étudiants est encodé ; Les attributions de l'enseignant sont encodées

Postconditions : Les notes sont sauvegardées et prêtes à être publiées.



### **Modifier notes**

Événement déclencheur : Un enseignant veut modifier une ou des notes pour une session.

Description : L'enseignant choisit l'UE pour laquelle il désire modifier la/les notes et il encode la/les nouvelle(s) note(s).

Acteur(s) impliqué(s) : Enseignant

Préconditions : L'enseignant est authentifié ; Le PAE des étudiants est encodé ; Les attributions de l'enseignant sont encodées ; des notes ont préalablement été encodées.

Postconditions : Les nouvelles notes sont sauvegardées et prêtes à être publiées.

### **Publier les notes**

Événement déclencheur : la date officielle de publication des notes est atteinte.

Description : L'administrateur choisit la session pour laquelle il veut publier les notes et il confirme la publication des notes.

Acteur(s) impliqué(s) : Administrateur

Préconditions : L'administrateur est authentifié ; Le PAE des étudiants est encodé ; Les attributions de l'enseignant sont encodées ; Toute les notes de la session ont préalablement été encodées.

Postconditions : Les notes sont publiées à destination des étudiants.

### **Encoder les attributions**

Événement déclencheur : les attributions des enseignants pour l'année académique sont connues et stables.

Description : L'administrateur encode les attributions pour toutes les UE.

Acteur(s) impliqué(s) : Administrateur

Préconditions : L'administrateur est authentifié ;

Postconditions : Les attributions sont encodées ce qui permet aux enseignants d'encoder les notes.



## Exercice 2 : RideLink (Covoiturage)

### Consignes

On vous demande d'évaluer la proposition d'interface utilisateur sur base de l'énoncé ci-dessous. Pour évaluer cette production, vous utiliserez les grilles d'évaluation fournies sur Learn (sous forme de tests).

### Énoncé

L'association étudiante de HELMo souhaite déployer une application web nommée **RideLink** pour faciliter le covoiturage entre étudiants depuis leur domicile/kot vers leur Campus. Le fonctionnement serait le suivant :

- Les **conducteurs** publient un trajet en renseignant l'origine, le campus de destination, la date/heure, le nombre de places disponibles et des informations complémentaires (bagages, pauses...). La publication se fait depuis l'application conducteurs ; le trajet devient visible immédiatement.
- Les **passagers** peuvent rechercher des trajets par origine/destination/date via la page de recherche, consulter les détails d'un trajet et demander une réservation d'une ou plusieurs places.
- Quand une réservation est effectuée, le **conducteur** reçoit une notification et peut accepter ou refuser la réservation avant le départ ; en cas de refus, le passager est notifié automatiquement. Si le conducteur accepte, la place est définitivement attribuée.
- Les **administrateurs** du service gèrent les comptes (validation/désactivation), consultent l'historique des trajets et peuvent traiter les signalements (comportement, litiges).
- Le système envoie automatiquement des e-mails (confirmation de réservation, rappel de départ, annulation) via un service de messagerie.

### Contraintes métier importantes :

- Seuls les utilisateurs inscrits et authentifiés peuvent publier un trajet ou demander une réservation.
- Le nombre de places proposées par un conducteur est un entier strictement positif.
- Les réservations ne peuvent dépasser le nombre de places disponibles ; le système applique la première règle du premier arrivé / premier servi.

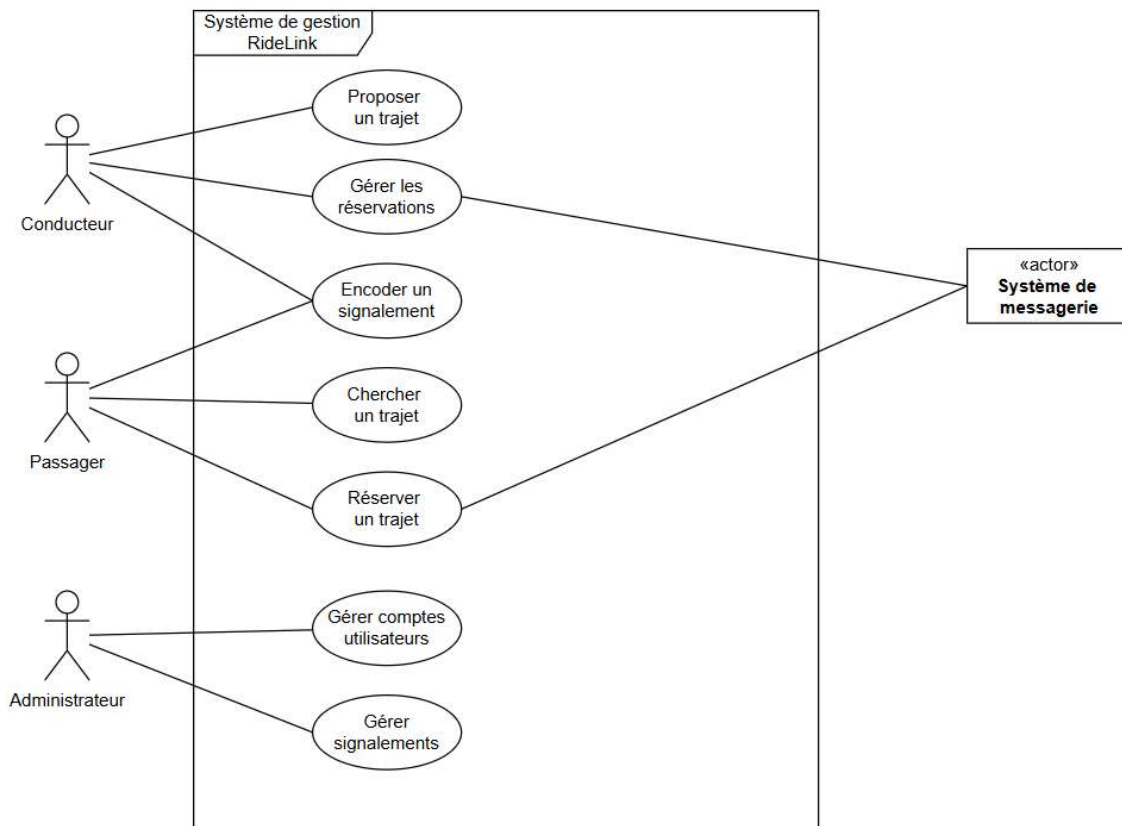


Figure 1 Diagramme de UC pour le système RideLink

## Descriptions textuelles des UC

### 1. Proposer un trajet

- **Acteur(s) bénéficiant du UC** : Conducteur
- **Événement déclencheur du UC** : Le conducteur souhaite proposer un trajet pour transporter des passagers jusqu'à son campus.
- **Description brève** : Le conducteur complète le formulaire de proposition de trajet : origine, campus de destination, date/heure, nombre de places et informations complémentaires (bagages, fumeur / non-fumeur, pauses prévues). Le système vérifie la validité des champs puis publie le trajet et le rend visible dans la liste de recherche. Si un champ obligatoire est manquant, la publication échoue et le système affiche un message d'erreur afin que l'utilisateur puisse corriger son encodage.
- **Précondition(s)** : L'utilisateur est inscrit, authentifié en tant que conducteur et possède au moins une place disponible (nombre de places renseigné > 0).
- **Postcondition(s)** : Le trajet est enregistré dans le système et visible par les passagers.

### 2. Chercher un trajet

- **Acteur(s) bénéficiant du UC** : Passager



- **Événement déclencheur du UC** : Le passager veut trouver un trajet correspondant à son déplacement.
- **Description brève** : Le passager saisit origine, destination et date/heure (optionnellement des filtres : nombre de places, bagages autorisés...). Le système affiche les trajets correspondants triés (par proximité). Le passager voit la liste et peut consulter les détails d'un trajet et voir le nombre de places restantes. Si aucun trajet ne correspond, un message d'information est affiché.
- **Précondition(s)** : L'utilisateur est inscrit, authentifié en tant que passager.
- **Postcondition(s)** : Le passager a accès à une liste de trajets filtrée correspondant à sa recherche.

### 3. Réserver une place

- **Acteur(s) bénéficiant du UC** : Passager
- **Événement déclencheur du UC** : Le passager décide de réserver une ou plusieurs places sur un trajet.
- **Description brève** : Le passager initie une demande de réservation depuis la fiche du trajet en indiquant le nombre de places souhaitées. Le système vérifie que le nombre de places demandées est  $\leq$  places disponibles. Si OK, la réservation est enregistrée en statut "en attente d'approbation" et le système envoie une notification au conducteur. Si le nombre demandé dépasse la disponibilité, la réservation est refusée et une suggestion d'autres trajets est proposée.
- **Précondition(s)** : Le passager est inscrit et authentifié en tant que passager ; le trajet existe et présente au moins une place libre.
- **Postcondition(s)** : Une réservation provisoire ("en attente d'approbation") est créée pour le passager et le conducteur concerné est prévenu via une notification par mail.

### Propositions d'UI à évaluer

#### UC 'Proposer un trajet'

Voici une proposition d'interface utilisateur pour couvrir le cas d'utilisation 'Proposer un trajet'. On vous demande d'évaluer cette proposition via le test 'RideLink : évaluation de l'interface pour le UC 'Proposer un trajet' que vous trouverez dans l'espace Learn du cours.



RideLink - Proposer un itinéraire

[Encoder trajet](#)
[Gérer réservations](#)
[Encoder signalement](#)

RM Logout

**Encodez votre proposition d'itinéraire**

Votre prénom et nom

Détails du trajet proposé (origine - destination)

Date

Nombre de places

☐ 1
 ☐ 2
 ☐ 3
 ☐ 4
 ☐ 5
 ☐ 6
 ☐ 7
 ☐ 8

Figure 2 : proposition d'interface pour le UC 'Proposer un trajet'

### Enchaînement des UC 'Chercher un trajet' et 'Réserver un trajet'

Voici une proposition d'interface utilisateur pour couvrir l'enchaînement des cas d'utilisation 'Chercher un trajet' et 'Réserver un trajet'. On vous demande d'évaluer cette proposition via le test 'RideLink : évaluation des interfaces pour 'Chercher + réserver un trajet' que vous trouverez dans l'espace Learn du cours.



RideLink - Chercher un trajet

Chercher un trajet
Réserver un trajet
Encoder un signalement
René Passager
Logout

Encodage des critères de recherche

Adresse de départ

Date

Nombre de places

Campus d'arrivée

Heure d'arrivée

☐ Bagages possibles

Rechercher

Date	Arrivée	Voir détails
12/10/2025	Campus des Coteaux	
12/10/2025	Campus des Coteaux	
12/10/2025	Campus des Coteaux	

Fonctionnement de la recherche :

- Les seuls champs obligatoires sont le Campus et la date.
- Il est possible d'échanger le départ et l'arrivée (retour du campus vers le domicile)
- Le champ avec l'adresse proposera des suggestions d'adresses valides en fonction des premiers caractères encodés.
- Le nombre de places est un entier > 0

En fonction des critères encodés, la liste des trajets correspondant s'affichera (un message adéquat s'affichera s'il n'y a aucun résultat).

En cliquant sur l'icône de la colonne 'Voir détails', on peut accéder à une nouvelle page donnant accès aux détails du trajet proposé.

RideLink - Réserver un trajet

Chercher un trajet
Réserver un trajet
Encoder un signalement
René Passager
Logout

Trajet proposé par :

Date / heure du départ :

Adresse de départ :

Adresse d'arrivée :

Nombre de places disponibles :

Bagages possibles :

Informations complémentaires :

Nathalie Conducteur

12/10/2025 à 7h30

Rue des pissenlits, 45 4000 Liège

Campus des Coteaux

2

☒

On ne fume pas et on ne mange pas dans ma voiture.

Figure 3 : Propositions d'interface utilisateur pour les UC Rechercher et réserver un trajet





### Exercice 3 : IntenseBeauty (Salon de beauté)

#### Consignes

On demande de proposer une UI ergonomique répondant aux cas besoins exprimés dans l'énoncé (et qui respectera les critères listés au cours théorique). Vous travaillerez en groupe de 2 ou 3 afin de vous partager les différents use case mais en restant attentif à la cohérence du tout.

Le fruit de votre travail sera déposé dans un document PDF sur HELMo Learn. Les interfaces utilisateur seront regroupées par use case.

#### Énoncé

« Intense Beauty » est un institut de beauté situé en région liégeoise. L'institut est composé de 9 esthéticiennes (dont 3 sont spécialisées dans les soins du visage par cryothérapie et 1 est spécialisée en gommage laser), deux hôtesse d'accueil, une comptable (à temps partiel) et la gérante.

L'institut propose une gamme complète de soins repris sur leur carte (épilation jambes (15€), massage temporal (12€), gommage paupières au laser (18€), soins 'peau-jeune' – cryothérapie (22€), masque aux concombres (16€), ...). Les soins se font uniquement sur rendez-vous, du mardi au samedi (de 10h00 à 18h00 avec interruption de 12h30 à 13h30).

Les rendez-vous se prennent exclusivement par téléphone. Lorsqu'un client désire prendre rendez-vous, il appelle l'institut et une des hôtesses d'accueil prendra son appel en charge. S'il s'agit d'un nouveau client, une nouvelle fiche client sera créée (nom, prénom, adresse, date de naissance, email, numéro de gsm). Ensuite, suivant le type de soin demandé, l'hôtesse d'accueil recherchera les disponibilités d'une employée adéquate et proposera au client la première date possible. Si cette date n'est pas possible, elle proposera la suivante jusqu'à trouver un rendez-vous qui convient. Une fois qu'une date conviendra au client, le rendez-vous sera consigné dans le carnet des rendez-vous de l'employée sélectionnée. Eventuellement, si aucune date ne convient la prise de rendez-vous est abandonnée.

La prise de rendez-vous pour un client existant sera similaire si ce n'est que l'hôtesse d'accueil commencera par rechercher la fiche du client et qu'elle y trouvera la liste des soins déjà reçus. Certains clients ont des habitudes et ils demandent explicitement que ce soit une esthéticienne particulière qui s'occupe d'eux.

Le jour avant le rendez-vous, une hôtesse d'accueil enverra un sms au client afin de lui rappeler la date et l'heure du rendez-vous.

Pour la bonne organisation du salon, chaque employée doit communiquer 15 jours à l'avance ses dates de congé à la gérante. Celle-ci vérifie le planning avant de les approuver et ensuite elle les consigne dans le carnet des rendez-vous. Une demande de congé peut être refusée si le planning des rendez-vous ne le permet pas.

Chaque mois, la comptable doit faire les comptes et donc compiler la liste des rendez-vous et les montants perçus.

La gérante désire vous confier la mission de concevoir un logiciel de gestion pour son institut. Le cahier des charges est le suivant :



- Les hôtesse d'accueil doivent pouvoir accéder facilement aux plannings des différentes employées et y ajouter, modifier ou supprimer des rendez-vous.
- Les hôtesse d'accueil doivent pouvoir créer de nouveaux dossiers pour les clients.
- Les hôtesse d'accueil doivent pouvoir rechercher la fiche d'un client sur base de son nom et/ou prénom.
- Les SMS de rappel doivent pouvoir être envoyés automatiquement.
- Les hôtesse d'accueil doivent pouvoir mentionner le montant à payer pour chaque type de soin et indiquer s'il a été payé ou pas.
- Au 10ème rendez-vous, une réduction de 10€ est accordée pour le soin suivant.
- Les esthéticiennes doivent pouvoir consulter facilement leur planning de la journée, de la semaine ou du mois en cours (sans avoir accès au planning de leurs collègues).
- Les employés doivent pouvoir encoder leurs demandes de congés qui devront pouvoir être approuvées ou rejetées par la gérante.
- Le comptable doit pouvoir établir différents rapports financiers :
  - Chiffre d'affaire de l'institut pour une période donnée
  - Chiffre d'affaire généré par une employée pour une période donnée
  - Détails des soins non-payés pour une période donnée
- Outre l'approbation des congés, la gérante veut pouvoir aussi s'occuper de l'accueil.