

Projet NTIC (32C2068) – Enseignant : Luka
Nerima / Assistant : Asheesh Gulati

Mobilize your LMS

Commanditaires : Patrick
Roth et Laurent Opprecht,
Service NTICE, Université de
Genève

Vjollca Ahmeti

Contenu¹

Remerciement.....	1
1. Introduction.....	2
2. Analyse de l'existant et des solutions	3
3. Evaluation des besoins des utilisateurs.....	4
Use Cases.....	4
Interface utilisateur.....	6
4. Choix de la solution	7
5. Mise en place de la solution.....	8
6. Résultats	11
Limites et difficultés	11
7. Conclusion et perspectives.....	14
8. Références.....	15

Remerciement

L'auteure du projet remercie l'équipe enseignante, Luka Nerima et Asheesh Gulati, pour leur accompagnement et leur aide, ainsi que les commanditaires du projet, Patrick Roth et Laurent Opprecht, pour leur aide, leur soutien et leurs encouragements durant toute la durée du projet. Merci infiniment à tous !

¹ N.B. : L'usage du masculin générique est employé pour désigner tant les hommes que les femmes.

1. Introduction

Pour ce projet, messieurs Patrick Roth et Laurent Opprecht, du service NTICE (*Nouvelles Technologies de l'Information et de la Communication pour l'Enseignement*) de l'Université de Genève, ont demandé d'envisager la mise en place d'une solution mobile pour le prochain LMS (*Learning Management System*) qui sera installé à la rentrée 2013.

L'idée de ce projet est donc de proposer une solution mobile pour l'affichage des fonctionnalités les plus utilisées par les étudiants de l'LMS Chamilo. Dans un premier temps, ce projet se limite effectivement uniquement aux utilisateurs « étudiants » de la plateforme. Chamilo sera mis en place afin de remplacer le LMS actuel le plus utilisé : Dokeos. Toutes les fonctionnalités disponibles ne seront pas nécessairement implémentées, seules les plus fréquentes seront retenues.

Le projet propose donc de revisiter la structure web des pages de Chamilo afin de **proposer une solution mobile adaptée et destinée aux étudiants**. Pour cela, nous allons, dans un premier temps, évaluer les solutions existantes en étudiant les solutions génériques qu'il existe pour le passage d'un site web en version mobile. Dans un second temps, nous évaluerons les besoins des utilisateurs afin de dégager les fonctionnalités qui sont le plus couramment utilisées par les étudiants et qui devraient être utilisables sur un appareil mobile. Dans un troisième temps, il s'agira d'analyser les pages concernées afin de déterminer les modifications de la CSS des pages pour permettre un affichage mobile. Il y a aura probablement plusieurs solutions possibles; il faudra donc choisir la solution la plus appropriée et argumenter les choix décisifs.

Le projet est mandaté par le service NTICE du département DSTIC (Division du système et des technologies de l'information et de la communication) de l'Université de Genève (ci-après Unige).

2. Analyse de l'existant et des solutions

Après avoir observé ce qui se faisait sur la toile en termes de technologie mobile, nous avons repéré au moins trois différentes méthodes pour transposer du contenu web sur un support mobile.

La première solution est de manipuler directement le contenu, son architecture et son affichage en créant une application mobile. La solution est complexe puisqu'elle exige de bien connaître à la fois le système qu'on souhaite transposer et à la fois la programmation sur les systèmes d'exploitation mobile. De plus, si l'on veut s'assurer d'atteindre tous les utilisateurs du système, il faut prévoir de développer une application par système d'exploitation mobile (ex. une application iPhone, une application pour Android). Par exemple, les TPG (*Transports Publics Genevois*) ont développé une application iPhone pour ses usagers, mais concernant les autres Smartphones, ils se sont contentés de mettre en place la seconde solution : un site web mobile².

La seconde solution est orientée vers l'affichage du contenu. Il s'agit de reconstruire la totalité de l'affichage du système afin d'en faire une version entièrement « mobile ». Peu importe le type d'appareil utilisé (PC, tablettes, liseuses, Smartphones), l'affichage ressemble toujours à celui qu'on obtient sur les appareils mobile, du moins, il respecte les conventions de design mobile. Le système est totalement revisité et, s'il s'agit d'un site web, il possède même une adresse url spécifique (généralement contenant un « m » supplémentaire). Le contenu est essentiellement modifié via son affichage, mais parfois, l'architecture peut subir des changements également pour optimiser l'affichage (ex. utiliser des images spécifique pour des icônes). A titre d'exemple, nous pouvons reprendre l'exemple des TPG qui ont donc développé un site web mobile³ dont l'url comporte un « m » pour désigner cette version mobile. L'Université de Fribourg a également mis en place une solution similaire pour son espace d'information⁴ (l'url commence également par « m »). Que l'on consulte ce type de sites web avec un Smartphone ou un ordinateur portable de 17 pouces, l'affichage est le même : il est clairement orienté mobile (ex. avec des icônes en guise de menu).

La troisième solution consiste à manipuler uniquement l'affichage du système, sans jamais toucher au contenu ou à son architecture profonde, afin d'adapter l'affichage en fonction de l'appareil qu'utilise l'utilisateur lorsqu'il consulte le système. Ainsi, lorsque l'utilisateur consulte le système sur un ordinateur portable de 15 pouces, il le voit affiché comme à son habitude. En revanche, en visualisant le même système (même url) à partir de son Smartphone, l'utilisateur va voir l'affichage du système modifié de sorte à ce qu'il s'adapte à son petit écran, et ceci en n'ayant fait aucune manipulation particulière.

Nous discuterons ci-après de notre choix parmi ces solutions, mais pour pouvoir faire un choix judicieux, il est important de définir les besoins des utilisateurs de Chamilo.

² <http://www.tpg.ch/communique-du-25.07.2011>

³ <http://m.tpg.ch/index.htm>

⁴ <http://m.unifr.ch/home/>

3. Evaluation des besoins des utilisateurs

Dans la mesure où Chamilo n'a pas encore été implémenté à l'Unige, il était évidemment impossible de consulter les étudiants à ce sujet. En revanche, nous pouvions les interroger sur le système déjà en place afin d'évaluer leur usage générique d'un LMS, d'autant plus que Chamilo est en quelque sorte une version améliorée de Dokeos (c'est un descendant de celui-ci).

Nous n'avons pas entrepris une analyse des besoins à grande échelle, car Bonhôte et Pellet (2011) ont réalisé ce travail il y a peu de temps auparavant et ont permis de dégager les principales fonctionnalités de Dokeos qu'utilisaient les étudiants de l'Unige. Leur travail visait à définir l'interfaçage sur les téléphones mobiles pour répondre au mieux aux besoins des utilisateurs. Il ne s'agissait donc pas de mettre en place un système, mais uniquement de définir l'interface utilisateurs. Cependant, grâce à leur travail, ils ont découvert que les étudiants avaient principalement une activité de consultation en ce qui concerne :

- Les documents des cours
- L'information générale sur le cours
- Les notifications sur les activités (cours ajouté, annonces ajoutées, etc.)

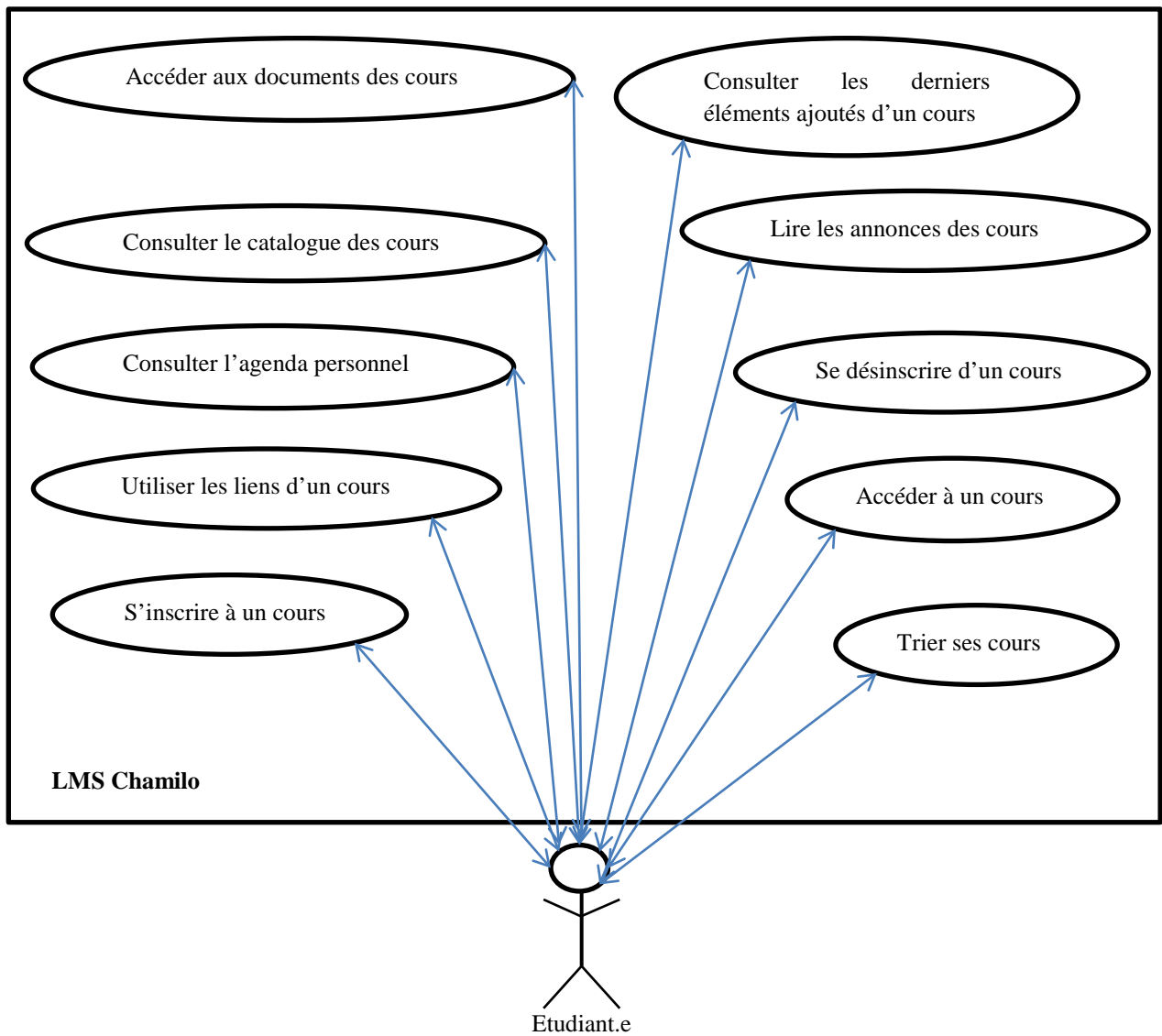
Les étudiants faisaient souvent également :

- La recherche d'un cours
- Des impressions de document ou diapositives
- Une inscription à un cours

Nous nous sommes donc basés sur l'analyse de ces auteurs pour relever les fonctionnalités que nous souhaitions spécifiquement mettre en avant pour un usage mobile. Nous avons maintenu toutes les fonctionnalités qui concernent les activités de consultation ou de recherche. En réalité, nous avons uniquement abandonné la mise en place de la fonctionnalité d'impression. Le système d'impression mis en place à l'Unige est en effet relativement complexe (il nécessite l'installation d'un programme pour pouvoir imprimer depuis son propre ordinateur) et n'est pas encore développé pour pouvoir lancer des impressions depuis un appareil mobile.

Use Cases

Partant de ce travail d'analyse des besoins, nous avons dégagé toutes les interactions que les utilisateurs seraient amenés à faire avec le système une fois que nous y aurons apporté les modifications nécessaires. Nous avons représenté ces interactions dans le diagramme de cas d'utilisation ci-dessous.



Le diagramme représente donc la classe d'utilisateur « Etudiant.e », seule classe dont nous nous préoccupons pour ce travail, qui interagit avec le système (le grand cadre), Chamilo. Les interactions que l'étudiant développe sont essentiellement des activités de consultation auxquelles le système répond par un affichage du contenu. Il n'y a aucune interaction complexe qui mériterait un développement approfondi. Même si nous ne les avons pas insérées dans cette représentation (car cela dépasse le cadre de notre travail), il est évident que pour le bon fonctionnement du système, les classes d'utilisateurs « Administrateur » et « Enseignant.e » sont nécessaires, car ce sont ces utilisateurs qui insèrent le contenu dans le LMS. L'interaction des étudiants commence lorsque le contenu est déjà créé, d'où cette représentation des cas d'utilisation.

Interface utilisateur

En préambule à la conception de l'interface utilisateur, nous avons étudié une proposition d'interface mobile pour Dokeos élaborée par deux étudiants du cours Interface personne-machine à l'Université de Genève. Dans leur travail, Bonhôte et Pellet (2011) répondent aux exigences des interfaces mobiles, puisqu'ils suivent les conventions ergonomiques pour ce type d'interface⁵. Par exemple, ils proposent un affichage avec des icônes à la place des menus pour le prototype mobile. Nous avons également réfléchi à l'affichage, car nous nous trouvions dans une situation de dilemme. En effet, nous avons le premier choix de suivre le travail de Bonhôte et Pellet (2011) qui respectait les conventions d'affichage mobile, mais qui réformait l'architecture sémantique de Dokeos. Par exemple, les auteurs proposaient de placer dans le menu d'accueil, au même niveau, une icône-bouton pour les cours et une icône-bouton les annonces, ce qui répondait bien aux besoins des utilisateurs. Mais il faut savoir que dans la hiérarchie du contenu, les annonces sont contenues dans les cours et ne sont donc pas au même niveau d'un point de vu sémantique. Le second choix que nous avons était donc de respecter l'architecture du système et de garder les mêmes niveaux de menus et sous-menus.

Nous avons donc opté pour un affichage classique des menus en version mobile (en ligne) et maintenu l'usage des icônes pour le contenu des cours. Ce choix est également ergonomique puisque, par soucis de consistance, il respecte le critère d'homogénéité et de cohérence de Bastien et Scapin (1998)⁶. L'affichage classique de Chamilo se présente en effet avec des menus en ligne et du contenu en blocs les uns sous les autres. Les icônes sont des éléments centraux dans le LMS puisqu'elles servent d'identificateur à toutes les activités disponibles pour un cours donné. Il nous semblait donc important de garder cette identité iconique des activités de cours qui en font la spécificité du système. A titre d'illustration, les premiers écrans pourraient donc ressembler aux layouts ci-dessous :

Ecran 1

Ecran 2

Ecran 3

Ecran 4



⁵ <http://www.ergonomie-interface.com/mobile-tactile-nomade/conception-site-web-mobile/> ;

<http://www.usability.gov/articles/newsletter/pubs/052010news.html>

⁶ <http://tecfa.unige.ch/~mireille/CriteresB&S.html>

4. Choix de la solution

Chamilo est un LMS relativement complexe dans son architecture (comme la plupart des LMS). Il nous paraissait donc difficile de mettre en place une application dédiée ou d'envisager composer avec la structure de Chamilo pour en faire un accès mobile. Nous avons donc opté pour la troisième solution : manipuler l'affichage CSS de Chamilo afin d'adapter l'affichage à des appareils mobiles. La solution est d'autant plus intéressante qu'avec CSS3, il est désormais possible de faire ce genre de manipulations relativement facilement grâce à la technologie Media Queries⁷. En effet, il suffit d'adresser les instructions CSS3 que l'on souhaite définir à un type de média particulier pour qu'elles s'appliquent lors de l'usage de ce média. L'adressage peut être combiné avec une expression supplémentaire qui indique par exemple la taille de l'écran. En CSS, cela donne :

```
@media (max-width: 480px) {  
}
```

La règle `@media` va adresser les instructions à tous types de média, mais il existe d'autres règles qui permettent d'adresser les instructions à un média particulier (ex. `@media projection` va adresser les instructions aux projecteurs). L'expression `(max-width: 480px)` permet d'adresser les instructions uniquement aux médias qui ont des écrans d'une largeur (*width*) maximale (*max*) de 480 pixels (`480px`). Cela correspond donc aux petits écrans de Smartphones.

A partir de là, il suffit simplement de placer toutes les instructions CSS que l'on souhaite appliquer à l'intérieur des crochets pour qu'elles soient prises en charge. Voici un exemple qui permet de redéfinir la taille du texte pour les titres *h3* et *h4* :

```
@media (max-width: 480px) {  
  h3 {  
    font-size: 18px;  
  }  
  
  h4 {  
    font-size: 16px;  
  }  
}
```

⁷ www.w3.org/TR/css3-mediaqueries/

5. Mise en place de la solution

Pour la mise en place concrète de cette solution, nous avons dans un premier temps installé la dernière version beta de Chamilo (version 1.9.0-beta) sur notre propre machine afin d'accéder aux fichiers CSS et les modifier en conséquence. Pour pouvoir visualiser nos modifications de façon effective, nous avons commencé par créer du contenu et des profils d'utilisateurs.

Après avoir observé la structure de Chamilo, nous nous sommes aperçu que la majorité des modifications pouvaient être réalisées à partir des fichiers CSS des thèmes graphiques de la plateforme. Il n'était donc pas nécessaire de s'attaquer aux fichiers centraux qui concernent toute la structure de Chamilo. Nous avons donc repris le thème « chamilo » tel quel et nous avons créé un thème « mobile ». Nous avons ensuite modifié les fichiers CSS à l'intérieur du thème. Cette première tentative a présenté deux problèmes majeurs :

1. La création d'un thème impose à l'administrateur de choisir le thème pour que les modifications soient visibles.
2. Malgré de nombreuses tentatives, nous ne pouvions visualiser le résultat de nos modifications depuis un appareil mobile à cause de l'accès à l'url (à partir de l'adresse IP).

Pour palier avant tout à ce second problème, nous avons installé Chamilo sur le serveur latIntic.unige.ch qui avait l'avantage de permettre un affichage directement en ligne. Après cette installation, nous avons eu une surprise majeure : l'affichage sur les appareils mobiles semblait avoir déjà été pris en charge dans cette version beta de Chamilo. Nous avons donc recherché les fichiers qui permettaient cet affichage. Nous avons trouvé à la racine du répertoire CSS, des fichiers qui contenaient effectivement des commandes Media Queries, ce qui a rendu notre première initiative (créer un thème mobile) complètement obsolète. Nous sommes donc repartis sur l'idée de s'attaquer directement aux fichiers concernés.

Notre travail ayant été grandement facilité par les nombreuses modifications orientées mobile déjà définies par les concepteurs, nous avons entrepris d'élargir le nombre de fonctionnalités disponibles pour les étudiants et avons donc gardé le menu « agenda » qui n'était pas prévu au départ. Par soucis de concision, nous ne développerons pas dans ce rapport toutes les modifications réalisées, cependant, elles sont disponibles dans le fichier [resonsive.css](#) avec nos commentaires qui indiquent ce que fait concrètement chaque modification. Par ailleurs, nous avons également modifié l'affichage graphique en reprenant le thème « mobile » créé et en le renommant « unige ». Nous avons notamment remplacé le logo de la plateforme par celui de l'Unige. Le reste des modifications graphiques « Unige » s'est fait dans le fichier [resonsive.css](#).

Cependant, à titre d'exemple, voici ci-dessous quelques éléments qui montrent la difficulté principale de notre travail : sélectionner correctement les éléments pour que seuls ceux-ci soient modifiés. La combinaison de sélecteurs multiples constituait en effet la majeure partie de notre travail. Nous nous sommes aidé pour cela de l'extension *Firebug*⁸ du navigateur *Mozilla Firefox*.

⁸ <https://getfirebug.com>

La modification suivante montre comment nous avons supprimé dans le premier écran, toutes les informations concernant les cours les plus actifs, ainsi que d'autres fonctionnalités pour ne limiter le premier écran qu'à l'activité de login. Remarquez notamment, les différents usages de sélection des éléments en combinant des sélecteurs `id` et `class`.

```
@media (max-width: 480px) {  
  
    /*Suprime l'affichage des cours les plus actifs au centre de l'écran*/  
    #hot_courses {  
        display: none;  
    }  
  
    /*Supprime le choix de la langue -> NITCE pourra choisir de laisser la langue  
ou pas*/  
    #lang_form {  
        display: none;  
    }  
  
    /*Affichage plus grand des étiquettes du block login*/  
    #login_block .control-label {  
        font-size: 18px;  
    }  
  
    /*Agrandi le label du bouton pour qu'il soit bien visible*/  
    #login_block .btn {  
        font-size: 20px;  
        margin: center;  
    }  
  
    /*Agrandi aussi les labels*/  
    #login_block>ul {  
        font-size: 16px;  
    }  
}
```

Certains éléments demandaient une combinaison de sélecteurs plus complexe :

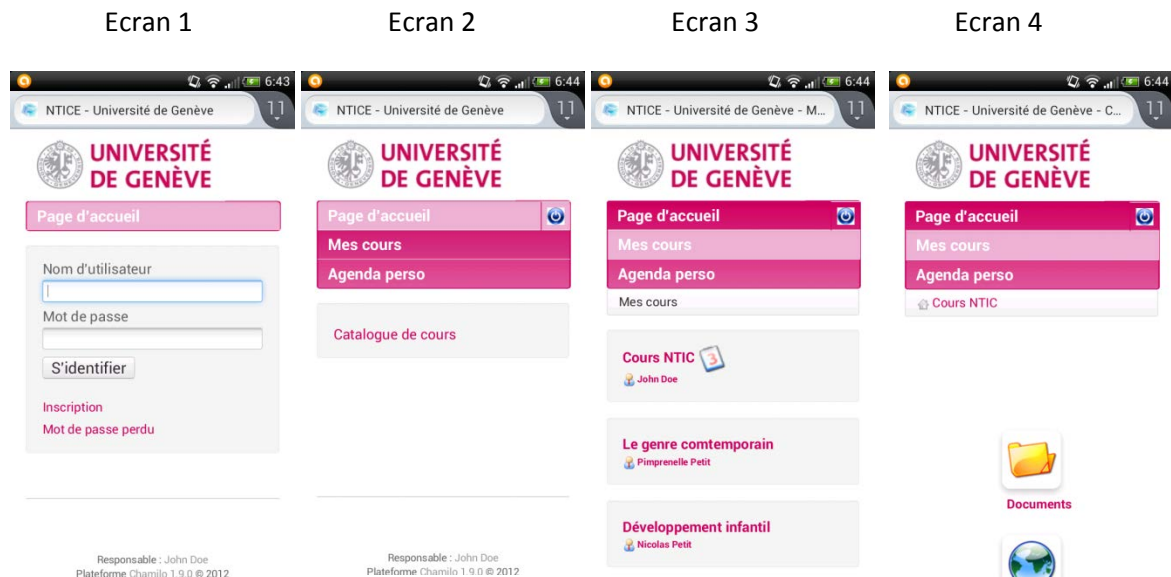
```
@media (max-width: 480px) {  
  /*Fait disparaître l'onglet "Ma progression"*/  
  .subnav ul:nth-child(2)>li:nth-last-child(2) {  
    display: none;  
  }  
  
  /*Fait disparaître l'onglet "Réseau social"*/  
  .subnav ul:nth-child(2)>li:last-child {  
    display: none;  
  }  
  /*Dans les documents des cours, supprime l'icône de téléchargement zippé  
  -> inutile sur téléphone mobile et empêche de cliquer sur le document*/  
  .data_table td:nth-child(2) img {  
    display: none;  
  }  
}
```

Pour mieux comprendre, par exemple, il faut lire la première modification de la manière suivante : « Sélectionner le second élément en partant de la fin [`nth-last-child(2)`] dans l'élément de liste [`li`] qui est l'enfant direct de la liste [`ul`], qui est elle-même le second élément [`nth-child(2)`] ayant pour parent le menu de navigation de classe [`subnav`]. » Plus concrètement, cela sélectionne l'onglet progression sans modifier aucun autre élément du même menu.

Concernant l'icône de téléchargement, comme elle n'avait pas d'identifiant spécifique (id), il fallait également trouver une combinaison de sélecteurs qui n'affectaient pas les autres éléments du tableau dans lequel elle était contenue.

6. Résultats

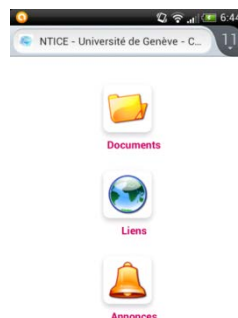
Le résultat final de toutes nos modifications correspond plutôt bien à nos exigences de départ. Nous avons en effet réussi à obtenir un menu en liste verticale pour les fonctionnalités que nous avons retenues et les icônes de cours sont également utilisées comme nous le souhaitons. Les copies d'écran ci-dessous montrent les quatre premiers écrans pour arriver au contenu du cours.



Limites et difficultés

Si nous sommes assez satisfaits du résultat final, il n'est cependant pas parfait. Il reste en effet quelques problèmes, des « bugs », que nous n'avons pas réussi à résoudre.

Le premier problème concerne l'affichage des icônes de cours, une fois à l'intérieur d'un cours. En effet, chaque icône est comprise dans un espace bloc défini par une classe `span4`. Dans le fichier original, il a été défini que ces blocs devaient s'afficher les uns sous les autres et leur largeur a également été fixée (en dur, directement dans le code HTML généré, et non dans la CSS) (Voir image ci-dessous).



L'affichage en colonne de toutes les icônes de cours fait sens lorsqu'il y en a beaucoup pour pouvoir cliquer dessus plus facilement, mais lorsqu'il n'y en a plus que trois, cela ne fait plus sens. Nous souhaitons donc afficher ces icônes dans une seule ligne. Or, il est difficile de modifier ces blocs sans perturber l'affichage, car il y a d'autres éléments important qui détiennent également la classe `span4`, notamment la liste des cours. Nous avons découvert par ailleurs qu'il est impossible de sélectionner un élément en CSS en fonction de son élément parent⁹ (idée de modifier l'affichage des icônes en passant par la classe `span4`, mais sans modifier les autres éléments ayant la même classe). De plus, il était impossible de sélectionner ces blocs en sélectionnant le n^{ème} élément, car le nombre d'élément varie en fonction du nombre de fonctionnalité que l'enseignant décide de mettre à disposition des étudiants ou non. Malgré de nombreux essais pour ne sélectionner que les icônes de cours et changer leur positionnement sur la page, nous n'avons pas réussi à obtenir le résultat que nous souhaitons. La solution la moins « bricolée » serait de directement modifier l'affichage en dur dans le code HTML généré. Ce premier problème témoigne bien de la problématique des LMS lorsqu'il s'agit de modifier leur affichage : l'architecture du code n'est pas toujours des plus parfaites ! Temporairement, nous avons opté pour la solution de n'afficher uniquement les icônes de cours qui nous intéressaient (les documents, les liens et les annonces), en supprimant toutes les autres grâce à leur id.

```
@media (max-width: 480px) {
    /*Supprime toutes les icones de cours, même si l'enseignant les affiche,
    sauf les icones de "Documents", de "Liens" et d'"Annonces"*/
    .span4 #tooldesc_1,#tooldesc_2,#tooldesc_4,#tooldesc_6,
    #tooldesc_8,#tooldesc_9,#tooldesc_10,#tooldesc_11,#tooldesc_12,#tooldesc_1
    3,#tooldesc_14,#tooldesc_15,#tooldesc_16,#tooldesc_17,
    #tooldesc_18,#tooldesc_19,#tooldesc_20 {
        display: none;
    }

    /*Fait la même chose pour leurs étiquettes*/
    .span4
    #istooldesc_1,#istooldesc_2,#istooldesc_4,#istooldesc_6,#istooldesc_8,#istoolde
    sc_9,#istooldesc_10,#istooldesc_11,#istooldesc_12,#istooldesc_13,#istooldesc_
    14,#istooldesc_15,#istooldesc_16,#istooldesc_17,#istooldesc_18,#istooldesc_19,
    #istooldesc_20 {
        display: none;
    }
}
```

⁹ <http://css-tricks.com/parent-selectors-in-css/>

Le second problème important que nous avons eu concerne le bouton « Rechercher » que l'on retrouve à la fois dans les documents de cours et dans le catalogue des cours. En effet, sur un écran d'appareil mobile, ce bouton s'affiche sous son champ de saisie associé, mais par la même occasion, il se superpose au texte qui vient immédiatement dessous (voir image ci-dessous).



Comme pour le problème précédent, cet élément bouton possède une commande d'affichage directement dans le code HTML à laquelle il est difficile de toucher [``]. Nous n'avons toujours pas réussi à résoudre le problème.

Enfin, il reste encore probablement des petits problèmes d'affichage que nous avons peu investigués et qui mériterait un examen supplémentaire. Par exemple, il serait utile de se pencher sur l'affichage de fenêtres pop-up lorsqu'on clique sur un événement d'agenda ou le nom d'un enseignant.

7. Conclusion et perspectives

Le bilan de ce projet est plutôt positif puisque nous avons pu apporter un début de solution à la demande du service NITCE. Bien entendu, notre travail a largement été facilité par le travail préalable des concepteurs qui s'étaient déjà penchés sur la problématique de l'affichage mobile, c'est pour cela que nous avons élargi notre champ d'actions. Cependant, certains bugs persistent et nous pensons qu'ils mériteraient une investigation supplémentaire pour obtenir le résultat attendu.

Des améliorations sont donc possibles, et si nous avons en partie répondu à la demande des mandataires de ce projet, nous souhaitons néanmoins leurs proposer une série de recommandations afin de poursuivre ce travail et mené à bien le projet dans son intégralité.

Nous faisons part de cinq recommandations importantes :

1. **Implémenter la version 1.9.0 de Chamilo qui est désormais stable.** Elle contient en effet probablement le même type de fichiers destinés à l'affichage mobile.
2. **Y ajouter nos modifications CSS**
3. **Chercher à réparer les bugs restants**
4. **Tester les modifications auprès de la population cible** et refaire une analyse des besoins des utilisateurs en deux temps : avant la mise en place de la plateforme et après un an d'utilisation. Certaines fonctionnalités sont tout à fait nouvelles dans Chamilo par rapport à Dokeos (ex. les fonctionnalités de réseautage). Il est important de vérifier l'usage qu'en font les utilisateurs.
5. **Faire un travail similaire pour les utilisateurs enseignants.** Ce travail n'a pas de sens si des modifications similaires ne sont pas également étudiées pour les enseignants. En effet, si l'on souhaite inciter les étudiants à utiliser la plateforme également sur leur appareil mobile, il est important de faire le même travail en amont avec les enseignants, les deux groupes de population étant dynamiquement liés. Un mauvais usage de la plateforme par les enseignants pourrait péjorer fortement l'usage des étudiants.

Pour terminer, nous tenions à préciser que ce travail a été réalisé à travers un système de portfolio électronique (portfolio.unige.ch) qui permet le suivi de projet. L'avantage de cette plateforme est sa richesse d'outils. Nous avons pu en effet réaliser notre travail, le commenter au fur et à mesure à travers un journal de bord, le présenter aux enseignants et aux mandataires qui pouvaient également commenter le travail, poser des questions sur un forum, etc., et le tout à travers une plateforme unique permettant le suivi de projet et les interactions avec toute l'équipe de travail. Le projet s'étant déroulé durant la période d'été, les interactions n'étaient pas très riches, mais il nous parut intéressant d'essayer de travailler avec cette méthode qui encourage le suivi de projet. Le contenu du travail est donc disponible sur portfolio.unige.ch, moyennant une demande d'invitation à l'auteur du projet.

8. Références

- Bastien, C., Leulier, C., Scapin, D., Thouin, C., Lynch, P., Horton, S., & Faulkner, X. (1998). L'ergonomie des sites web. Créer et maintenir un service web. *Cours INRIA*, 27. Consulté de http://www.metrics.it/ELERCO/fr/E2_4350_IAL_F_01_V.doc
- Bonhôte, J., & Pellet, C. (2011). *Projet Dokeos Mobile*. Université de Genève.