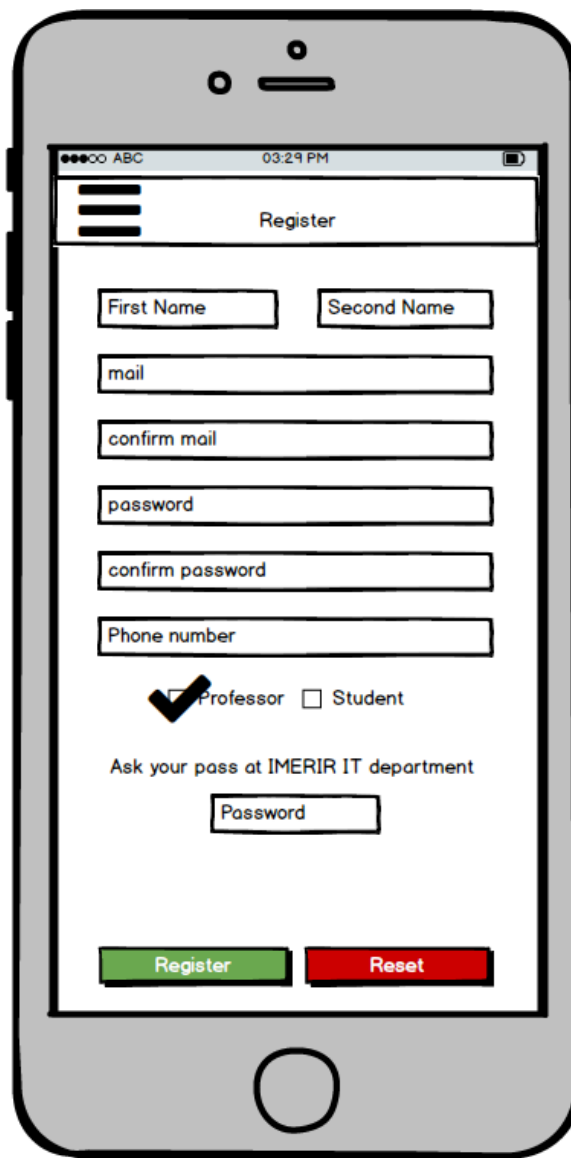


Page de connexion

L'utilisateur peut alors soit se connecter avec ses identifiants si il à déjà crée un compte sinon il peut se créer un compte à l'aide du bouton s'enregistrer.

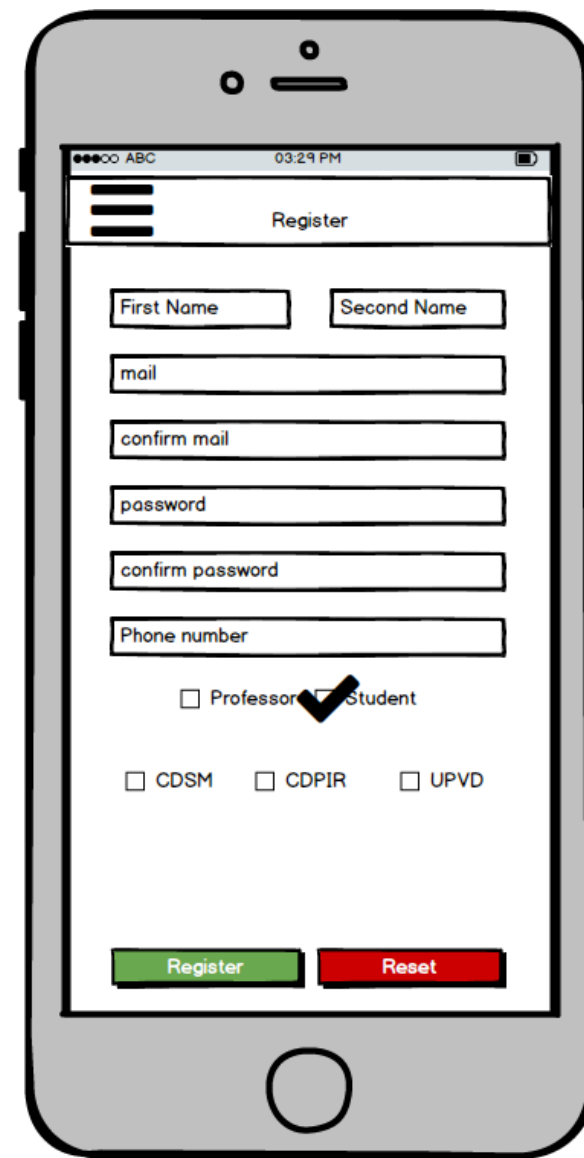


Page de création de compte

L'utilisateur peut se créer un compte en complétant les champs indiqués.

Seule une adresse mail @imerir.com est acceptée.

Si la case professeur est cochée un champ mot de passe apparait.
C'est un mot de passe qui permet la création d'un compte prof qui sera donné sous demande au service info pour éviter que les élèves aient les mêmes droits que les profs.



Page de création de compte

L'utilisateur peut se créer un compte en complétant les champs indiqués.

Seule une adresse mail @imerir.com est acceptée.

Si la case élève est cochée l'utilisateur devra entrer sa formation afin que les offres proposées soient filtrées en rapport avec la formation.

Le compte élève possède moins de droits que celui du professeur.

```

+ WebServiceUserClient
- fields
  - instance: WebServiceUserClient
  - context: Context
  - queue: RequestQueue
  - users: ArrayList<User>
  - userHash: HashMap<String, User>
  - userHashByld: HashMap<Integer, User>
- constructors
  - WebServiceUserClient (context: Context)
- methods
  + createInstance (appContext: Context): void
  + getInstance (): WebServiceUserClient
  + requestUsers (listener: OnUsersListListener): void
  + POSTUser (user: User): void
  + getUsers (): ArrayList<User>
  + getUser (mail: String): User
  + getUserByld (id: Integer): User

```

```

+ User
  implements Serializable
- fields
  - id: int
  - type: boolean
  - nom: String
  - prenom: String
  - mail: String
  - telephone: String
  - formation: int
  - cvID: int
  - lmlD: int
  - etiquette: String
  - dashBoard: String
  - participations: int
  - password: String
  - isValid: boolean
- constructors
  + User (json: JSONObject)
- methods
  + getId (): int
  + setId (id: int): void
  + getType (): boolean
  + setType (type: boolean): void
  + getNom (): String
  + setNom (nom: String): void
  + getPrenom (): String
  + setPrenom (prenom: String): void
  + getMail (): String
  + setMail (mail: String): void
  + getTelephone (): String
  + setTelephone (telephone: String): void
  + getFormation (): int
  + setFormation (formation: int): void
  + getCvID (): int
  + setCvID (cvID: int): void
  + getLmlD (): int
  + setLmlD (lmlD: int): void
  + getEtiquette (): String
  + setEtiquette (etiquette: String): void
  + getDashBoard (): String
  + setDashBoard (dashBoard: String): void
  + getParticipations (): int
  + setParticipations (participations: int): void
  + getPassword (): String
  + setPassword (password: String): void
  + isValid (): boolean
  + setValid (valid: boolean): void

```

```

+ WebServiceUserClient.OnUsersListListener
- fields
- methods
  + onUsersReceived (users: ArrayList<User>): void
  + onUsersFailed (error: String): void

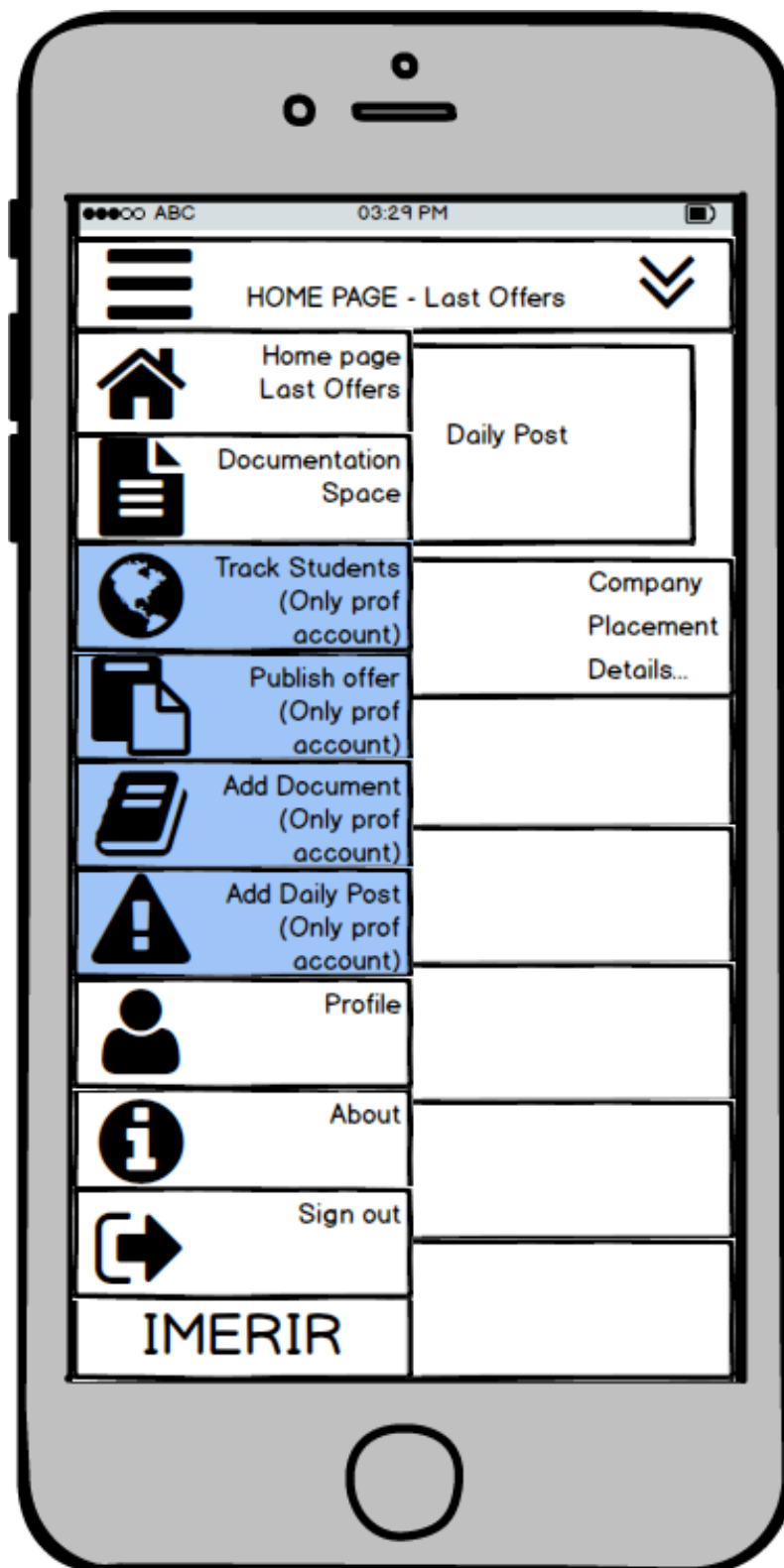
```

```

+ LoginActivity extends AppCompatActivity
  implements View.OnClickListener
  WebServiceUserClient.OnUsersListListener
- fields
  - _user: ArrayList<User>
  ~ user: User
  - cr: boolean
  ~ floatingActionMenu: FloatingActionMenu
  ~ fbaStudent: FloatingActionButton
  ~ fbaResponsable: FloatingActionButton
  ~ mail: EditText
  ~ password: EditText
  ~ accountNotValid: CardView
  ~ connexion: Button
  ~ sendMail: Button
  ~ loadingDialog: ProgressDialog
  - progressStatus: int
  - handler: Handler
  - listener: Listener
  ~ internet: int
  ~ cbRemember: CheckBox
  - email: String
  - loginPreferences: SharedPreferences
  - loginPrefsEditor: Editor
  - saveLogin: Boolean
  - mToolbar: Toolbar
  - mSwipeRefreshLayoutLogin: SwipeRefreshLayout
- constructors
- methods
  # onCreate (savedInstanceState: Bundle): void
  + onClick (v: View): void
  - setupRefreshSwipe (): void
  - checkUser (mail: String, password: String): boolean
  + onUsersReceived (users: ArrayList<User>): void
  + onUsersFailed (error: String): void
  + onCreateOptionsMenu (menu: Menu): boolean
  + onOptionsItemSelected (item: MenuItem): boolean
  + showLoadingDialogData (): void
  - checkConnectivity (): int

```





Slide menu de l'application

L'utilisateur aura accès à différentes fonctions indiquées sur le schéma.

Les champs en bleus sont ceux uniquement accessibles par les professeurs.

Le menu sera consultable dans toutes les fenêtres de l'application.

Page d'accueil - Dernières offres

Voici la page d'accueil l'utilisateur se verra accueilli par le dernier message laissé par les responsables ARE

Ensuite il pourra voir les dernières offres qui ont été publiées sous forme de liste en ayant la possibilité de voir quelques informations concernant l'offre.

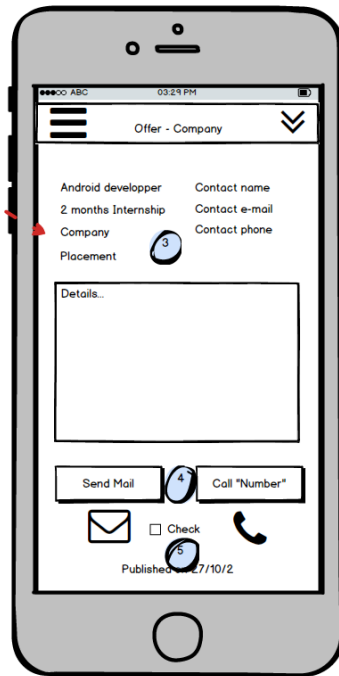
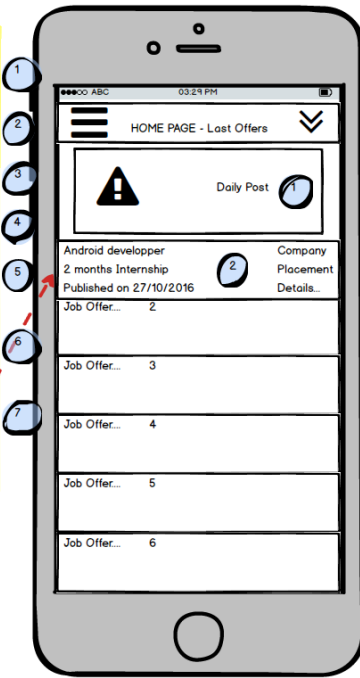
L'utilisateur pourra sélectionner une offre ce qui l'emmènera sur une page où l'offre sera présentée avec plus de détails.

L'utilisateur pourra si l'offre l'intéresse envoyer un email direct à l'entreprise (si email disponible) et/ou appeler directement l'entreprise si numéro disponible.

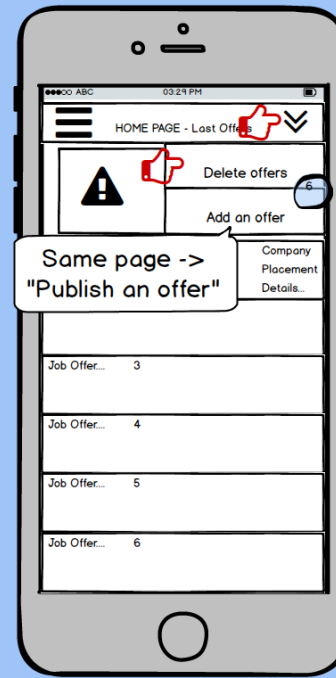
Si l'élève a participé à l'offre il pourra l'indiquer sur la checkbox cela permettra aux professeurs d'avoir un suivi des élèves si oui ou non ils sont actifs à la recherche d'emploi.

Si l'utilisateur possède un compte professeur il aura accès à un menu supplémentaire qui lui permettra soit d'ajouter une offre (lien vers l'activité ajouter une offre) ou soit supprimer une sélection d'offres (checkbox qui s'affichent et une poubelle pour valider la suppression).

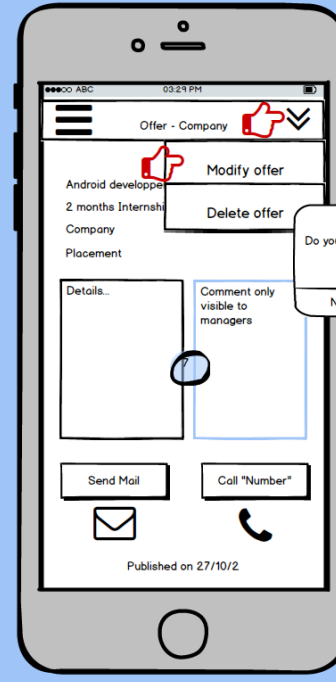
Si l'utilisateur possède un compte professeur en consultant une offre il aura accès à une information supplémentaire (Détails destinés aux professeurs). Et accès à un menu permettant l'action de supprimer l'offre en question et permettant l'action modifier l'offre.



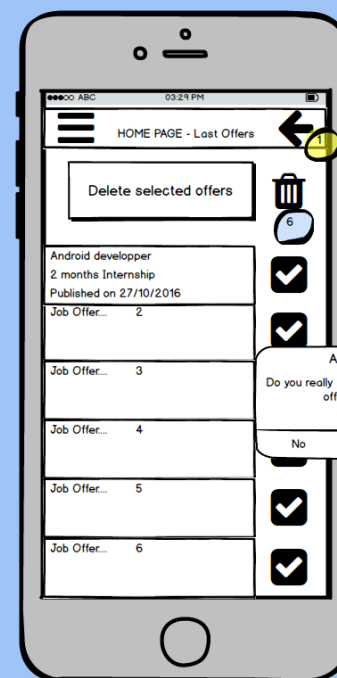
Only professor account



Same page -> "Publish an offer"



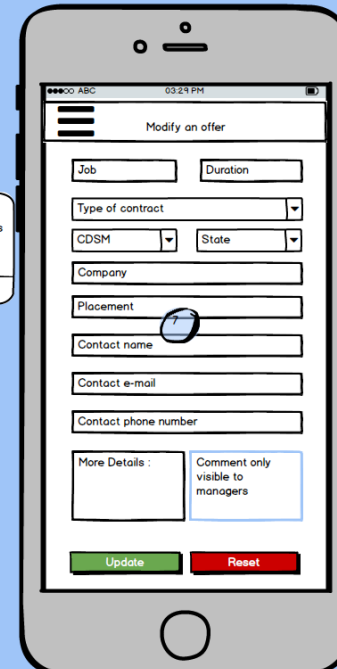
Alert
Do you really want delete this offer ?
No Yes



Alert
Do you really want delete this offers ?
No Yes

Helpful
1 Back Button
2 Consult on offer

Only prof account can access this parts




```

+ Offer
- fields
- offerID:int
- typeContrat:int
- formationAssociee:int
- etat:int
- details:String
- detailsResponsables:String
- datePublicaiton:String
- contactID:int
- nomContact:String
- nomEntreprise:String
- lieuEntreprise:String
- mailContact:String
- telephoneContact:String
- titre:String
- dureeContrat:String
- constructors
+ Offer (json:JSONObject)
- methods
+ getOfferID():int
+ setOfferID(offerID:int):void
+ getTitre():String
+ setTitre(titre:String):void
+ getDureeContrat():String
+ setDureeContrat(dureeContrat:String):void
+ getTypeContrat():int
+ setTypeContrat(typeContrat:int):void
+ getFormationAssociee():int
+ setFormationAssociee(formationAssociee:int):void
+ getEtat():int
+ setEtat(etat:int):void
+ getDetails():String
+ setDetails(details:String):void
+ getDetailsResponsables():String
+ setDetailsResponsables(detailsResponsables:String):void
+ getDatePublicaiton():String
+ setDatePublicaiton(datePublicaiton:String):void
+ getContactID():int
+ setContactID(contactID:int):void
+ getNomContact():String
+ setNomContact(nomContact:String):void
+ getNomEntreprise():String
+ setNomEntreprise(nomEntreprise:String):void
+ getLieuEntreprise():String
+ setLieuEntreprise(lieuEntreprise:String):void
+ getMailContact():String
+ setMailContact(mailContact:String):void
+ getTelephoneContact():String
+ setTelephoneContact(telephoneContact:String):void

```

```

+ WebServiceOfferClient
- fields
- instance:WebServiceOfferClient
- context:Context
- queue:RequestQueue
- offers:ArrayList<Offer>
- offerHash:HashMap<Integer, Offer>
- constructors
- WebServiceOfferClient (context:Context)
- methods
+ createInstance (appContext:Context):void
+ getInstance():WebServiceOfferClient
+ requestOffers (listener:OnOffersListListener):void
+ getOffers():ArrayList<Offer>
+ getOffer(id:int):Offer
+ POSTOffer (offer:Offer):void
+ PUTOffer (offer:Offer, id:int):void
+ DELETEOffer(id:int):void

```

```

+ OfferDetailActivity extends AppCompatActivity
implements View.OnClickListener
- fields
- final CALL_PHONE_PERMISSION:int
- mToolbar:Toolbar
~ Titre:TextView
~ Duree:TextView
~ NomEntreprise:TextView
~ Lieu:TextView
~ NomContact:TextView
~ Mail:TextView
~ Telephone:TextView
~ DatePublish:TextView
~ Contrat:TextView
~ tvDetailsResponsables:TextView
~ Details:TextView
~ DetailsResponsables:TextView
~ callBtn:FloatingActionButton
~ mailBtn:FloatingActionButton
~ offer:Offer
- constructors
- methods
# onCreate (savedInstanceState: Bundle):void
+ onClick(v:View):void
+ showCallDialog():void
+ onRequestPermissionsResult(requestCode:int, permissions:String[], grantResults:int[]):void
+ onCreateOptionsMenu (menu:Menu):boolean
+ onOptionsItemSelected (item:MenuItem):boolean

```



ABC 03:29 PM

Publish an offer

Job Duration

Type of contract

2 CDSM State 1

Company

Placement

Contact name

Contact e-mail

Contact phone number

More Details : Comment only visible to managers

Publish Reset 1

L'état de l'offre se décompose en 3 états :

- 1
- Urgent
 - En recherche
 - Plus valide (Archive l'offre)

Liste pour filtrer le public cible de l'offre :

- 2
- CDSM
 - CDPIR
 - UPVD

Type de contrat :

- 3
- Stage
 - Apprentissage
 -

Publier une offre

Activité accessible par prof uniquement

Formulaire permettant de proposer une offre aux étudiants

Boutons permettant de publier l'offre ou alors de remettre à 0 tous les champs.

1

```

+ WebServiceOfferClient
  fields
  - instance :WebServiceOfferClient
  - context: Context
  - queue: RequestQueue
  - offers :ArrayList<Offer>
  - offerHash :HashMap<Integer, Offer>
  constructors
  - WebServiceOfferClient (context: Context)
  methods
  + createInstance (appContext: Context):void
  + getInstance ():WebServiceOfferClient
  + requestOffers (listener: OnOffersListListener ):void
  + getOffers ():ArrayList<Offer>
  + getOffer (id:int): Offer
  + POSTOffer (offer: Offer ):void
  + PUTOffer (offer: Offer , id:int):void
  + DELETEOffer(id:int):void

```

```

+ AddOfferActivity extends AppCompatActivity
  fields
  - _formation:int
  - _typeContrat:int
  - nomOffre: EditText
  - duree: EditText
  - nomEntreprise: EditText
  - lieu: EditText
  - nomContact: EditText
  - mailContact: EditText
  - telephoneContact: EditText
  - details: EditText
  - detailsResponsable: EditText
  - typeContrat: Spinner
  - formation: Spinner
  - ajouterOffre: Button
  - mToolbar: Toolbar
  constructors
  methods
  # onCreate (savedInstanceState: Bundle):void
  - _castAndSendOffer ():void

```

```

+ Offer
  fields
  - offerID: int
  - typeContrat: int
  - formationAssociee: int
  - etat: int
  - details: String
  - detailsResponsables: String
  - datePublicaiton: String
  - contactID: int
  - nomContact: String
  - nomEntreprise: String
  - lieuEntreprise: String
  - mailContact: String
  - telephoneContact: String
  - titre: String
  - dureeContrat: String
  constructors
  + Offer (json:JSONObject)
  methods
  + getOfferID ():int
  + setOfferID (offerID: int):void
  + getTitre ():String
  + setTitre (titre: String):void
  + getDureeContrat ():String
  + setDureeContrat (dureeContrat: String):void
  + getTypeContrat ():int
  + setTypeContrat (typeContrat: int):void
  + getFormationAssociee ():int
  + setFormationAssociee (formationAssociee: int):void
  + getEtat ():int
  + setEtat (etat: int):void
  + getDetails ():String
  + setDetails (details: String):void
  + getDetailsResponsables ():String
  + setDetailsResponsables (detailsResponsables: String):void
  + getDatePublicaiton ():String
  + setDatePublicaiton (datePublicaiton: String):void
  + getContactID ():int
  + setContactID (contactID: int):void
  + getNomContact ():String
  + setNomContact (nomContact: String):void
  + getNomEntreprise ():String
  + setNomEntreprise (nomEntreprise: String):void
  + getLieuEntreprise ():String
  + setLieuEntreprise (lieuEntreprise: String):void
  + getMailContact ():String
  + setMailContact (mailContact: String):void
  + getTelephoneContact ():String
  + setTelephoneContact (telephoneContact: String):void

```



```

+ WebServiceOfferClient
└─ fields
  - instance :WebServiceOfferClient
  - context :Context
  - queue :RequestQueue
  - offers :ArrayList<Offer>
  - offerHash :HashMap<Integer, Offer>
└─ constructors
  - WebServiceOfferClient (context: Context)
└─ methods
  + createInstance (appContext: Context):void
  + getInstance ():WebServiceOfferClient
  + requestOffers (listener: OnOffersListListener ):void
  + getOffers ():ArrayList<Offer>
  + getOffer (id:int): Offer
  + POSTOffer (offer: Offer ):void
  + PUTOffer (offer: Offer , id:int):void
  + DELETEOffer(id:int):void

```

```

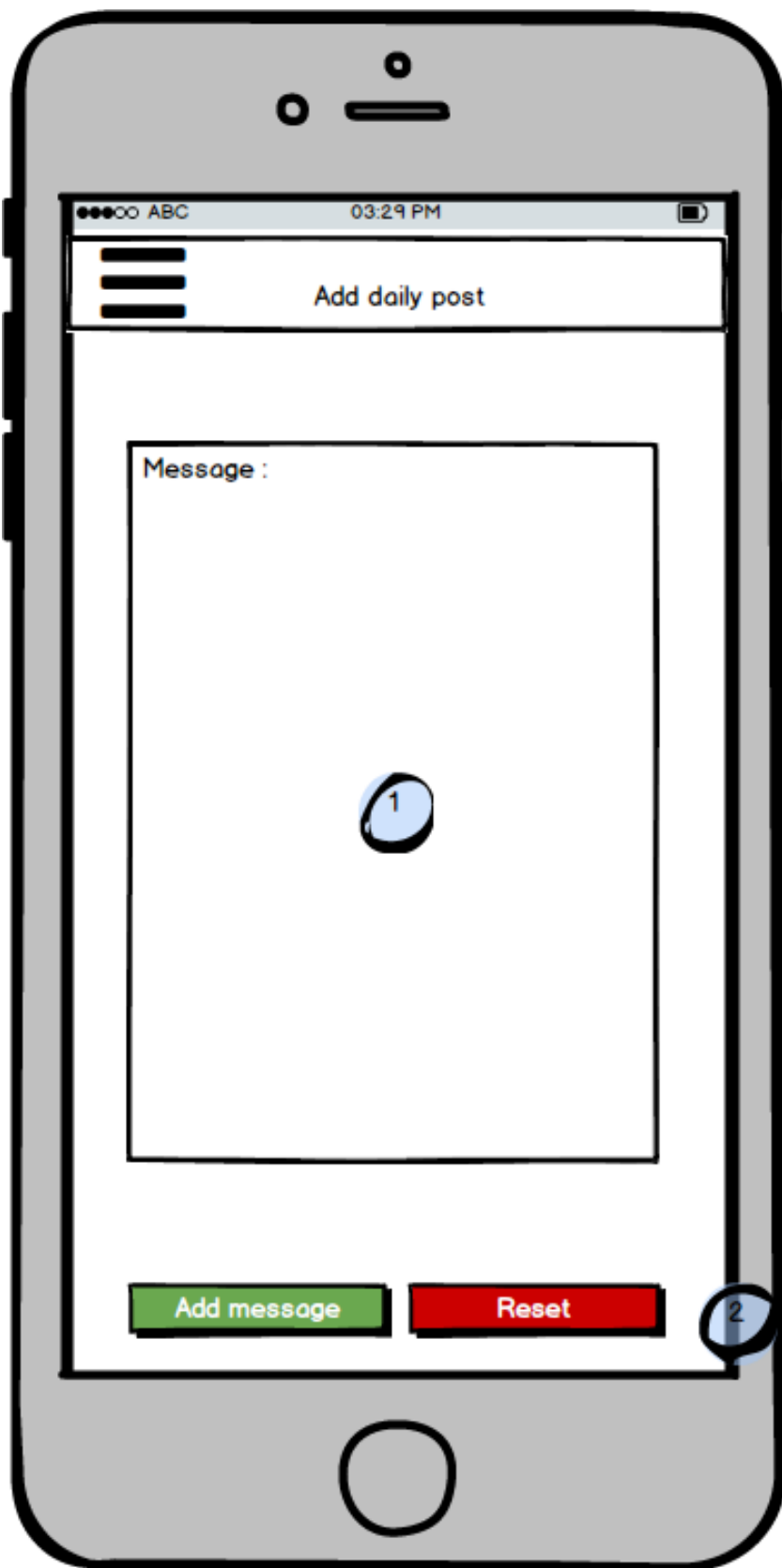
+ AddOfferActivity extends AppCompatActivity
└─ fields
  - _formation:int
  - _typeContrat:int
  - nomOffre: EditText
  - duree: EditText
  - nomEntreprise: EditText
  - lieu: EditText
  - nomContact: EditText
  - mailContact: EditText
  - telephoneContact: EditText
  - details: EditText
  - detailsResponsable: EditText
  - typeContrat: Spinner
  - formation: Spinner
  - ajouterOffre: Button
  - mToolbar: Toolbar
└─ constructors
└─ methods
  # onCreate (savedInstanceState: Bundle):void
  - _castAndSendOffer ():void

```

```

+ Offer
└─ fields
  - offerID :int
  - typeContrat :int
  - formationAssociee :int
  - etat :int
  - details :String
  - detailsResponsables :String
  - datePublicaiton :String
  - contactID :int
  - nomContact :String
  - nomEntreprise :String
  - lieuEntreprise :String
  - mailContact :String
  - telephoneContact :String
  - titre :String
  - dureeContrat :String
└─ constructors
  + Offer (json:JSONObject)
└─ methods
  + getOfferID ():int
  + setOfferID (offerID: int):void
  + getTitre ():String
  + setTitre (titre: String):void
  + getDureeContrat ():String
  + setDureeContrat (dureeContrat: String):void
  + getTypeContrat ():int
  + setTypeContrat (typeContrat: int):void
  + getFormationAssociee ():int
  + setFormationAssociee (formationAssociee: int):void
  + getEtat ():int
  + setEtat (etat: int):void
  + getDetails ():String
  + setDetails (details: String):void
  + getDetailsResponsables ():String
  + setDetailsResponsables (detailsResponsables: String):void
  + getDatePublicaiton ():String
  + setDatePublicaiton (datePublicaiton: String):void
  + getContactID ():int
  + setContactID (contactID: int):void
  + getNomContact ():String
  + setNomContact (nomContact: String):void
  + getNomEntreprise ():String
  + setNomEntreprise (nomEntreprise: String):void
  + getLieuEntreprise ():String
  + setLieuEntreprise (lieuEntreprise: String):void
  + getMailContact ():String
  + setMailContact (mailContact: String):void
  + getTelephoneContact ():String
  + setTelephoneContact (telephoneContact: String):void

```



Ajouter un message d'accueil

Activité accessible par prof uniquement

Formulaire permettant de publier un message qui sera affiché sur la page d'accueil de l'application et envoie une notification sur le téléphone.

Bouton envoyer ou reset le formulaire

```

+ AddMessageActivity extends AppCompatActivity
    implements TextWatcher
[-] fields
- mToolbar: Toolbar
- nbCarRestants: TextView
- sendMessage: Button
- etMessage: EditText
[-] constructors
[-] methods
# onCreate (savedInstanceState: Bundle): void
+ afterTextChanged (editable: Editable): void
+ beforeTextChanged (s: CharSequence, start: int, count: int, after: int): void
+ onTextChanged (s: CharSequence, start: int, before: int, count: int): void

```

```

+ WelcomeMessage
[-] fields
- id: int
- publisherId: int
- message: String
[-] constructors
+ WelcomeMessage (json: JSONObject)
[-] methods
+ getId(): int
+ setId(id: int): void
+ getPublisherId(): int
+ setPublisherId(publisherId: int): void
+ getMessage(): String
+ setMessage(message: String): void

```

```

+ WebServiceMessageClient
[-] fields
- instance: WebServiceMessageClient
- context: Context
- queue: RequestQueue
- messages: ArrayList<WelcomeMessage>
- messageHash: HashMap<Integer, WelcomeMessage>
[-] constructors
- WebServiceMessageClient (context: Context)
[-] methods
+ createInstance (appContext: Context): void
+ getInstance(): WebServiceMessageClient
+ requestMessages (listener: OnMessagesListListener): void
+ requestAllMessages (listener: OnMessagesListListener): void
+ POSTMessage (message: WelcomeMessage): void
+ getMessages(): ArrayList<WelcomeMessage>
+ getMessage(id: String): WelcomeMessage
+ DELETEMessage(id: int): void

```



Page consulter des documents

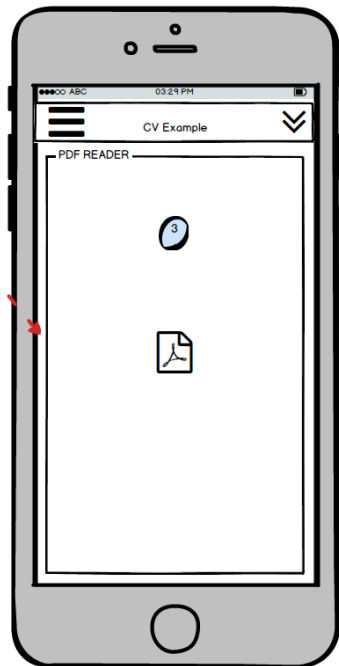
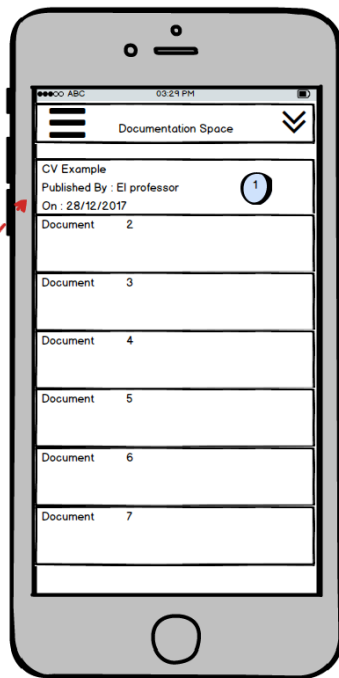
Cet espace documentation est un espace où les professeurs déposent des documents types CV's exemples LM exemples ... Les élèves peuvent alors les consulter afin de s'en servir de support pour rédiger leurs propres documents.

Même architecture que consulter des offres (liste et cliquer sur un élément pour le voir)

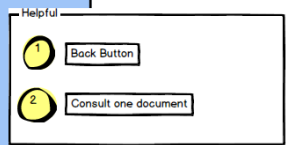
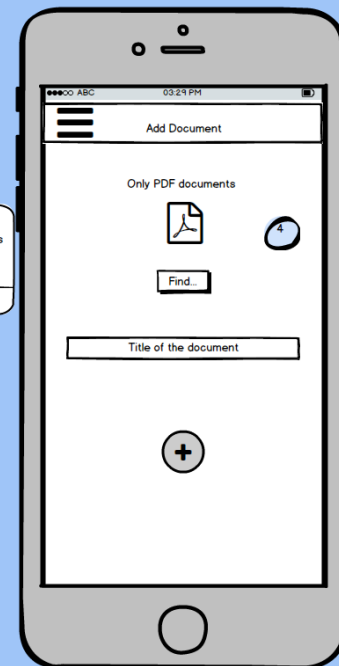
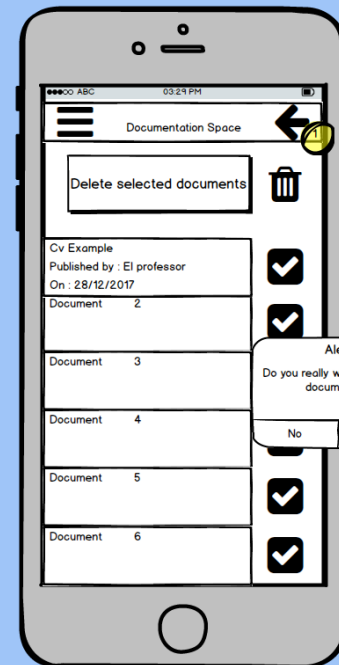
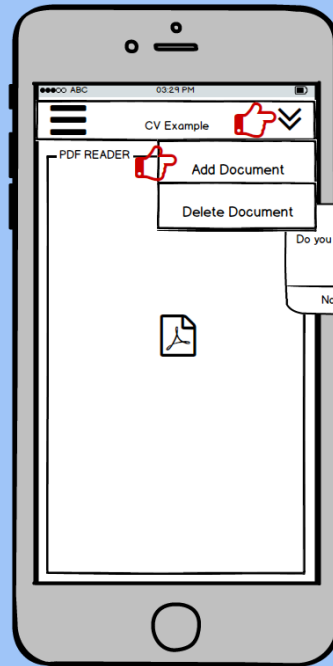
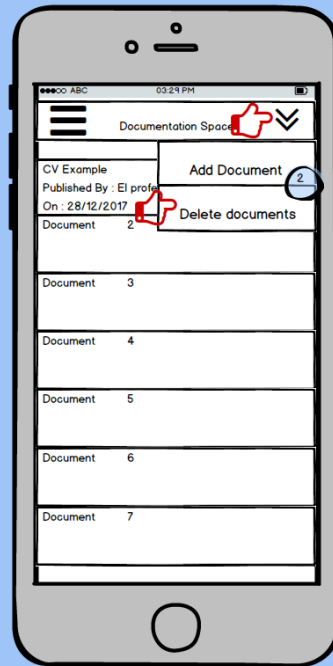
Les profs peuvent aussi supprimer ou ajouter un document directement dans le menu de l'activité.

Quand une offre est consultée cela ouvre un lecteur PDF directement inclus dans l'application, pas besoin de télécharger le doc pour le lire.

Voir ajouter document dans le chapitre ajouter document



Only professor account



Only prof account can access this parts

```

+ DocumentAdapter extends RecyclerView.Adapter
  implements WebServiceUserClient.OnUsersListListener
fields
- documents:ArrayList<Document>
~ cardView:CardView
- listener:OnDocumentClickListener
~ user:User
constructors
+ DocumentAdapter(documents:ArrayList<Document>, listener: OnDocumentClickListener)
methods
+ onCreateViewHolder (parent: ViewGroup, viewType: int):DocumentViewHolder
+ onBindViewHolder (holder:DocumentViewHolder, position:int):void
+ getItemCount():int
+ onUsersReceived (users:ArrayList<User>):void
+ onUsersFailed (error:String):void

```

```

+ User
  implements Serializable
fields
- id:int
- type:boolean
- nom:String
- prenom:String
- mail:String
- telephone:String
- formation:int
- cvID:int
- lmlD:int
- etiquette:String
- dashBoard:String
- participations:int
- password:String
- isValid:boolean
constructors
+ User(json:JSONObject)
methods
+ getId():int
+ setId(id:int):void
+ getType():boolean
+ setType(type:boolean):void
+ getNom():String
+ setNom(nom:String):void
+ getPrenom():String
+ setPrenom(prenom:String):void
+ getMail():String
+ setMail(mail:String):void
+ getTelephone():String
+ setTelephone(telephone:String):void
+ getFormation():int
+ setFormation(formation:int):void
+ getCvID():int
+ setCvID(cvID:int):void
+ getLmlD():int
+ setLmlD(lmlD:int):void
+ getEtiquette():String
+ setEtiquette(etiquette:String):void
+ getDashBoard():String
+ setDashBoard(dashBoard:String):void
+ getParticipations():int
+ setParticipations(participations:int):void
+ getPassword():String
+ setPassword(password:String):void
+ isValid():boolean
+ setValid(valid:boolean):void

```

```

+ Document
fields
- DocId:int
- PublisherId:int
- PublishDate:String
- DocumentURL:String
constructors
+ Document(json:JSONObject)
methods
+ getDocId():int
+ setDocId(docId:int):void
+ getPublisherId():int
+ setPublisherId(publisherId:int):void
+ getPublishDate():String
+ setPublishDate(publishDate:String):void
+ getDocumentURL():String
+ setDocumentURL(documentURL:String):void

```

```

+ WebServiceDocumentClient
fields
- instance:WebServiceDocumentClient
- context:Context
- queue:RequestQueue
- documents:ArrayList<Document>
- documentHash:HashMap<Integer, Document>
constructors
- WebServiceDocumentClient(context:Context)
methods
+ createInstance (appContext:Context):void
+ getInstance():WebServiceDocumentClient
+ requestDocuments(listener: OnDocumentsListListener):void
+ getDocuments():ArrayList<Document>
+ getDocument(id:int):Document
+ DELETEDocument(id:int):void

```

```

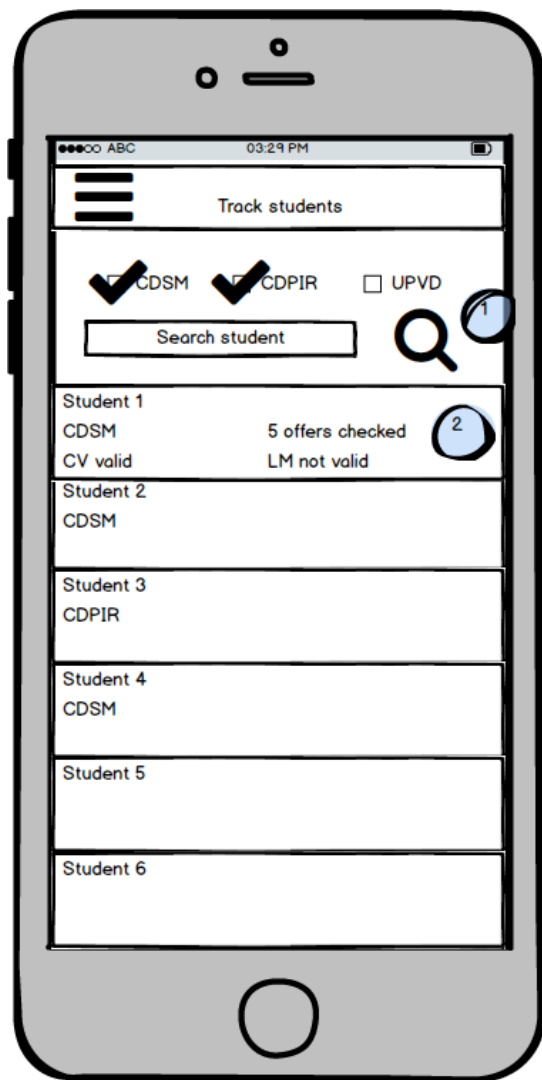
+ WebServiceDocumentClient.OnDocumentsListListener
fields
methods
+ onDocumentsReceived(Documents:ArrayList<Document>):void
+ onDocumentsFailed(error:String):void

```

```

+ DocumentListFragment extends Fragment
  implements View.OnClickListener
  WebServiceDocumentClient.OnDocumentsListListener
  DocumentAdapter.OnDocumentClickListener
fields
- final PICK_FILE_REQUEST:int
- final TAG:String
- selectedFilePath:String
- SERVER_URL:String
~ recyclerView:RecyclerView
~ fbaAddDocument:FloatingActionButton
~ dialog:ProgressDialog
~ mSwipeRefreshLayoutDocument:SwipeRefreshLayout
~ loadingDialog:ProgressDialog
~ progressStatus:int
~ handler:Handler
~ internet:int
~ _document:ArrayList<Document>
constructors
methods
+ onCreateView (inflater:LayoutInflater, container:ViewGroup, savedInstanceState:Bundle):View
+ onViewCreated (view:View, savedInstanceState:Bundle):void
+ onClick(v:View):void
+ onDocumentClicked(document:Document):void
+ onDocumentsReceived(Documents:ArrayList<Document>):void
+ onDocumentsFailed(error:String):void
+ onActivityResult(requestCode:int, resultCode:int, data:Intent):void
- showFileChooser():void
- setupRefreshSwipe():void
- showLoadingDialog():void
- checkConnectivity():int
+ uploadFile(selectedFilePath:String):int

```

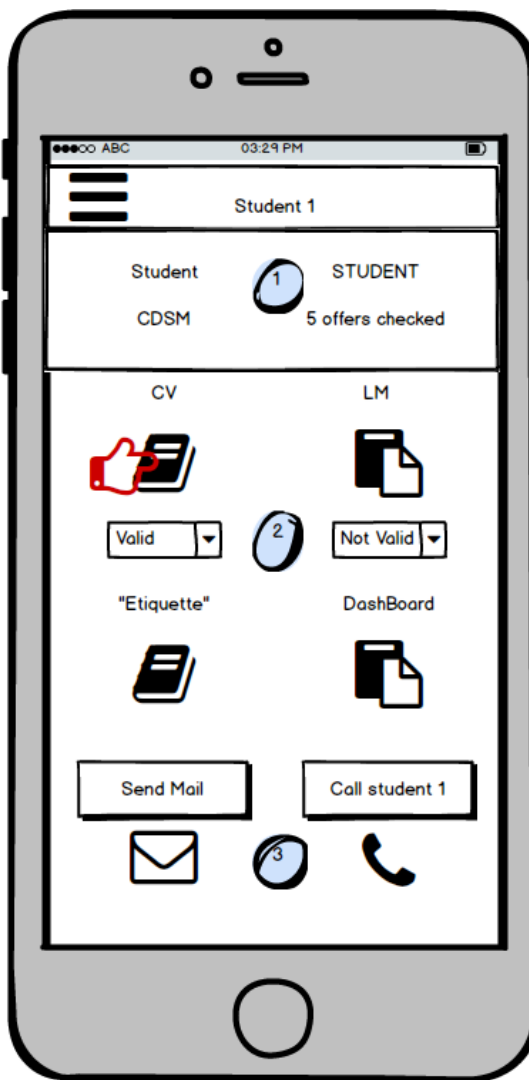



Liste des étudiants

Activité en accès uniquement par les professeurs permettant de consulter le profil d'un étudiant inscrit.

Un filtrage entre formations est possible pour une meilleure lisibilité.
Et un champ de recherche disponible pour rechercher un étudiant en particulier.

Une liste d'étudiants s'affiche en fonction des critères de recherche le professeur peut alors sélectionner un étudiant



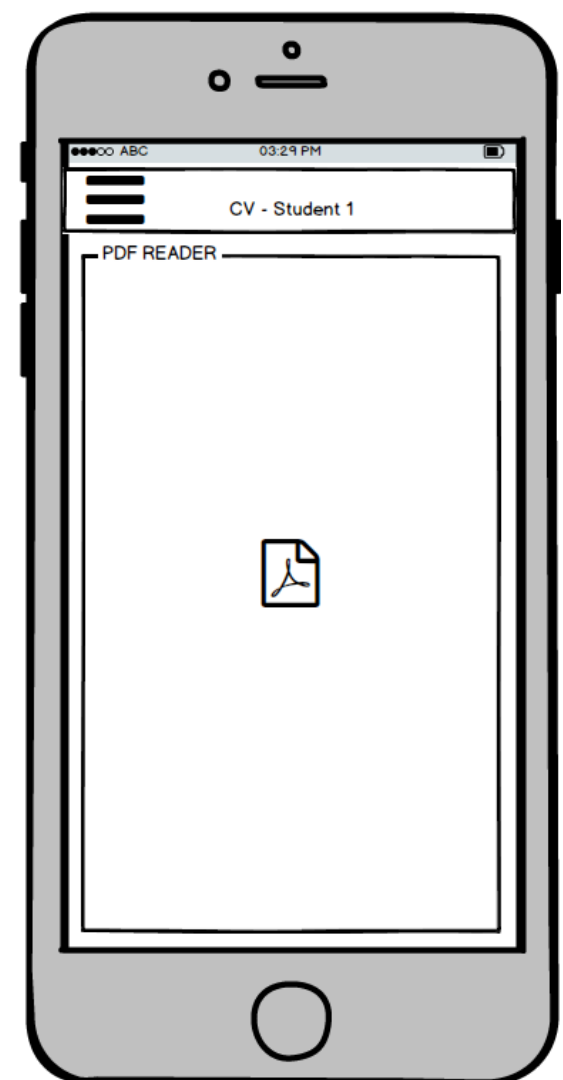
Profil d'un étudiant

Activité en accès uniquement par les professeurs permettant de consulter le profil d'un étudiant inscrit.

Le profil contient les informations de l'étudiant ainsi que les documents qu'il a déposés et le nombre d'offres auquel il a postulé.

Le professeur peut alors valider la qualité des documents de recherche d'entreprise de l'étudiant. Vérifier son étiquette et son tableau de bord si ils sont disponibles.

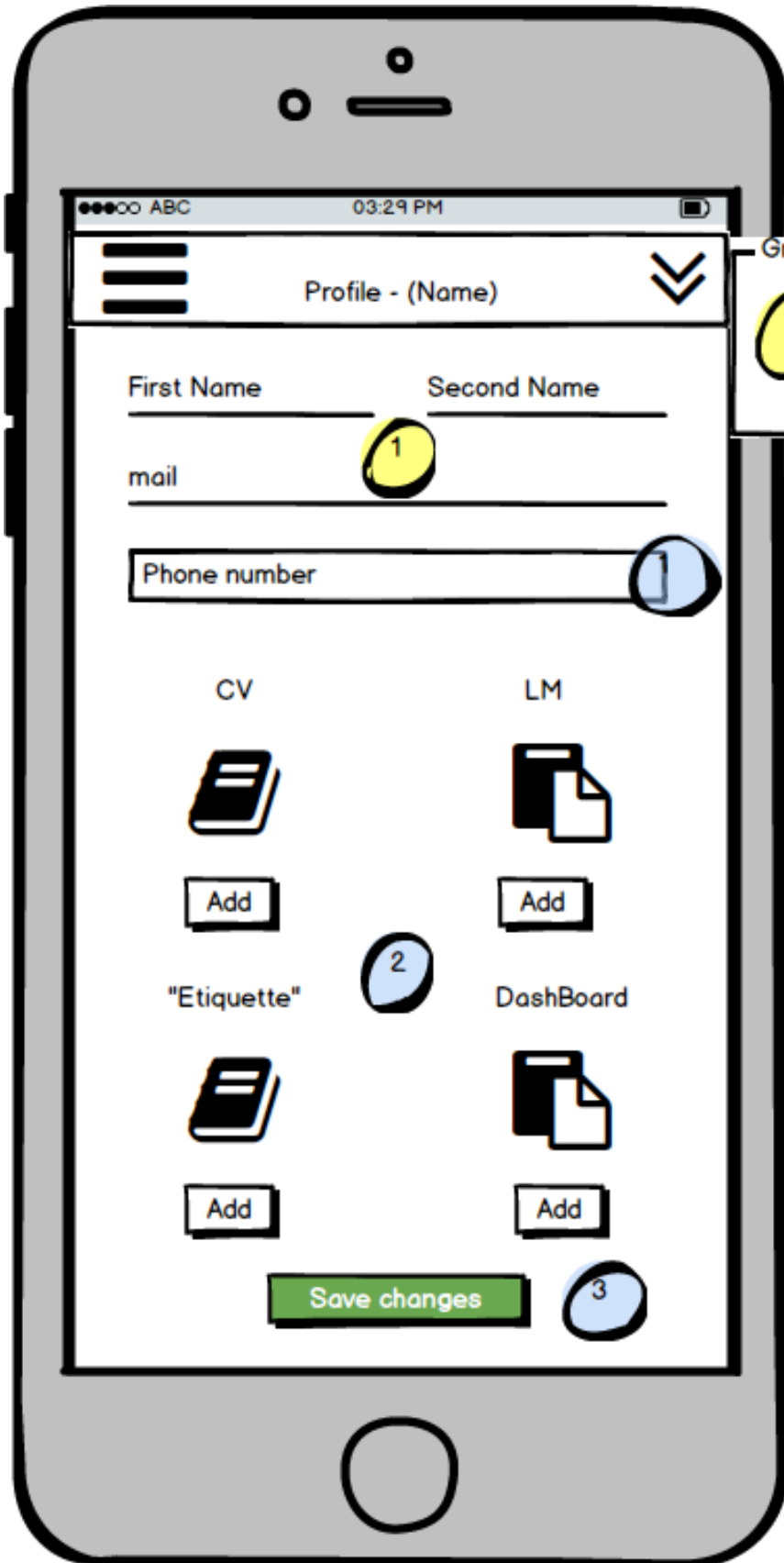
L'appeler ou lui envoyer un mail.



Profil d'un étudiant

Activité en accès uniquement par les professeurs permettant de consulter le profil d'un étudiant inscrit.

Si le professeur clique sur un document le lecteur pdf intégré l'affiche comme pour l'espace documentation.



Group Name

1

Non modifiable

Consulter son profil

Permet de modifier son numéro de téléphone

1

Formulaire permettant d'ajouter ses documents afin que les professeurs puissent faire un suivi de la recherche d'emploi

2

Bouton valider les changements

3

CV



Add

"Etiquette"



Add

LM



Add

DashBoard



Add

Save changes

3

```

+ User
  implements Serializable
  fields
  - id:int
  - type:boolean
  - nom:String
  - prenom:String
  - mail:String
  - telephone:String
  - formation:int
  - cvID:int
  - lmlD:int
  - etiquette:String
  - dashBoard:String
  - participations:int
  - password:String
  - isValid:boolean
  constructors
  + User(json:JSONObject)
  methods
  + getId():int
  + setId(id:int):void
  + getType():boolean
  + setType(type:boolean):void
  + getNom():String
  + setNom(nom:String):void
  + getPrenom():String
  + setPrenom(prenom:String):void
  + getMail():String
  + setMail(mail:String):void
  + getTelephone():String
  + setTelephone(telephone:String):void
  + getFormation():int
  + setFormation(formation:int):void
  + getCvID():int
  + setCvID(cvID:int):void
  + getLmlD():int
  + setLmlD(lmlD:int):void
  + getEtiquette():String
  + setEtiquette(etiquette:String):void
  + getDashBoard():String
  + setDashBoard(dashBoard:String):void
  + getParticipations():int
  + setParticipations(participations:int):void
  + getPassword():String
  + setPassword(password:String):void
  + isValid():boolean
  + setValid(valid:boolean):void

```

```

+ WebServiceUserClient
  fields
  - instance:WebServiceUserClient
  - context:Context
  - queue:RequestQueue
  - users:ArrayList<User>
  - userHash:HashMap<String, User>
  - userHashById:HashMap<Integer, User>
  constructors
  - WebServiceUserClient(context:Context)
  methods
  + createInstance(appContext:Context):void
  + getInstance():WebServiceUserClient
  + requestUsers(listener:OnUsersListListener):void
  + POSTUser(user:User):void
  + getUsers():ArrayList<User>
  + getUser(mail:String):User
  + getUserById(id:Integer):User

```

```

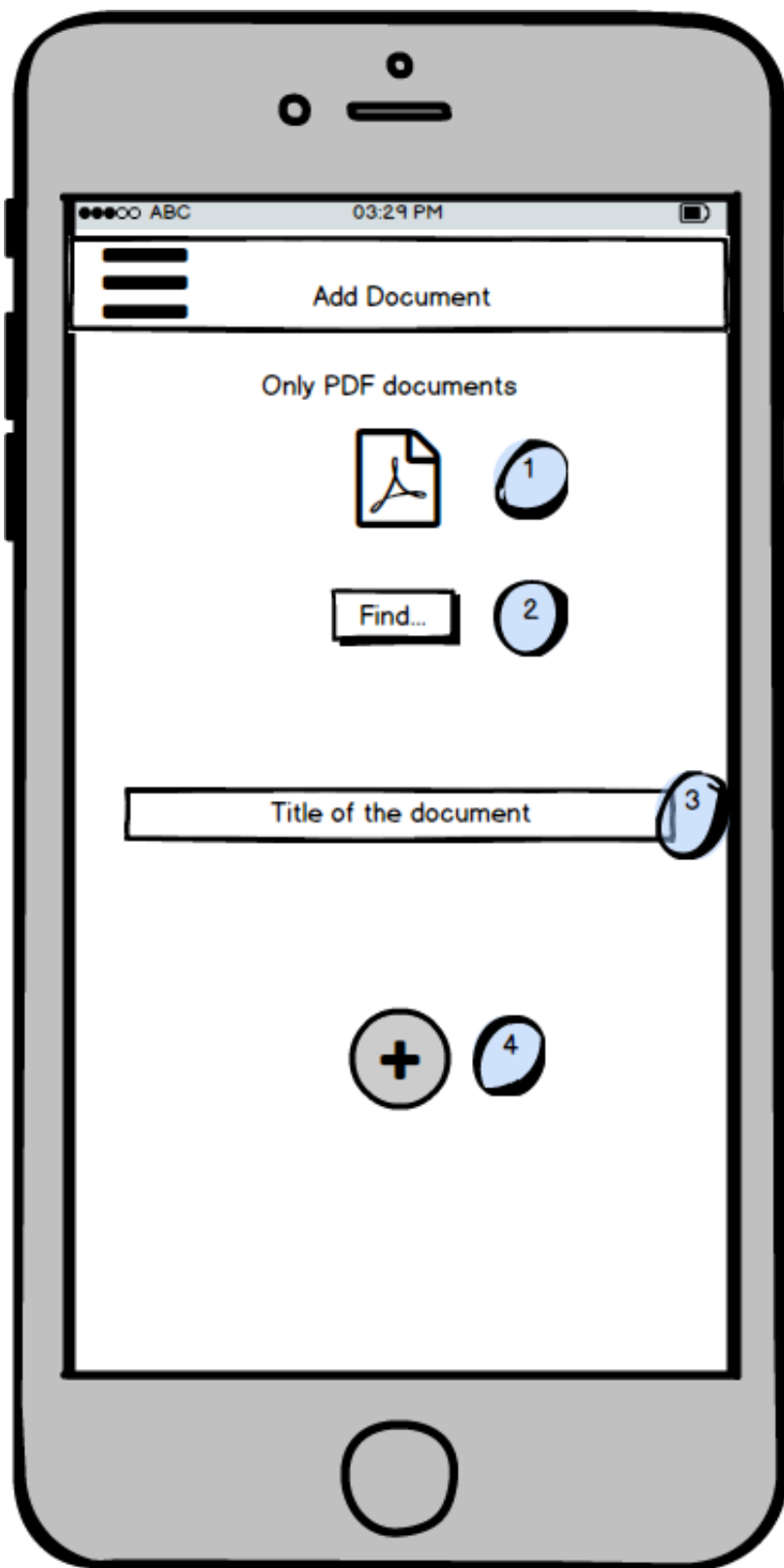
+ WebServiceUserClient.OnUsersListListener
  fields
  methods
  + onUsersReceived(users:ArrayList<User>):void
  + onUsersFailed(error:String):void

```

```

+ ProfilStudentFragment extends Fragment
  implements View.OnClickListener
  WebServiceUserClient.OnUsersListListener
  fields
  ~ user:User
  ~ _user:ArrayList<User>
  - final PICK_FILE_REQUEST_CV:int
  - final TAG:String
  ~ selectedFilePath:String
  ~ dialog:ProgressDialog
  ~ firstname:TextView
  ~ secondname:TextView
  ~ mail:TextView
  ~ phoneNumber:TextView
  ~ formation:TextView
  ~ valide:TextView
  ~ ibCV:ImageButton
  ~ ibLM:ImageButton
  ~ ibEtiquette:ImageButton
  ~ ibDB:ImageButton
  ~ floatingActionButtonEdited:FloatingActionButton
  ~ loadingDialog:ProgressDialog
  ~ progressStatus:int
  ~ handler:Handler
  ~ internet:int
  ~ mSwipeRefreshLayoutEtudiant:SwipeRefreshLayout
  constructors
  methods
  + onCreateView(inflater:LayoutInflater, container:ViewGroup, savedInstanceState:Bundle):View
  + onViewCreated(view:View, savedInstanceState:Bundle):void
  - showFileChooser():void
  - setupRefreshSwipe():void
  + onClick(view:View):void
  + onUsersReceived(users:ArrayList<User>):void
  + onUsersFailed(error:String):void
  + showLoadingDialog():void
  - checkConnectivity():int
  - showAlertNotImplemented():void

```



Publier un document

Activité accessible par prof uniquement

Formulaire permettant de proposer un document aux étudiants

Seulement les .pdf sont acceptés.

Bouton de recherche de doc

Affiche le titre du doc après téléchargement

Bouton pour publier le doc

1

2

3

4

Permettrait à un étudiant d'indiquer
aux profs si une offre n'est plus
valable

