POCKETHEPIA

Maxime Alexandre Lovino

4 Septembre 2018

PLAN

- Présentation du projet
- Objectifs techniques
- Technologies utilisées
- Résultats obtenus
- Conclusion

PRÉSENTATION DU PROJET

- Carte d'étudiant NFC
- Porte-monnaie électronique
  - Paiements entre étudiants ou à la cafétéria
- Contrôle d'accès
- Inspiré de la carte d'étudiant EPFL
- Application mobile et web pour accéder aux données et administrer

NFC (NEAR FIELD COMMUNICATION)

- Communication sans fil à faible portée
- Induction électromagnétique
- 3 modes de fonctionnement
  - Mode lecteur (actif)
  - Mode carte (passif)
  - Mode peer-to-peer

OBJECTIFS TECHNIQUES DU PROJET

- Application mobile native Android avec données hors-ligne
- Projet structuré autour des données
- Backend commun pour l'application web et mobile
- Frontend structuré en utilisant un framework

ROLES

- Utilisateur
- Administrateur
- Personne acceptant des paiements

ROLES

- Utilisateur
- Administrateur
- Personne acceptant des paiements
- Auditeur
- Bibliothécaire
- Administrateur d’accès pour une zone

MODÈLES

Log
_id
category
date
triggeringUser
description
rawData
Area
_id
name
User
_id
name
email
isAdmin
isLibrarian
acceptsPayments
canInvite
isAuditor
cardId
virtualCard
adminForAreas
expiration
importBatch
Room
_id
name
area
Transaction
_id
title
date
from
to
amount
stripe
adminCharge
Access
_id
user
room
startDate
endDate
startTime
endTime

ARCHITECTURE UTILISÉE

- Données hors-ligne sur mobile ✓
- Synchronisation automatique ✓
- Pas de format propriétaire
- Backend Node.JS + MongoDB
- Utilisation des nouvelles librairies Android Jetpack

ARCHITECTURE UTILISÉE

ANDROID JETPACK

- *Room* pour la gestion de base de données SQLite
- *WorkManager* pour la synchronisation en arrière-plan
- *Navigation* pour simplifier le chargement et la navigation à l'aide de Fragments

ROOM DATABASE

ROOM DATABASE

NAVIGATION

- Gestion de la *back-stack* automatique
- Passage d'arguments entre les écrans de façon typée
- Génération de classes et méthodes pour se déplacer vers d'autres écrans

NAVIGATION

Sans Jetpack Navigation

```
val intent = Intent(this, VirtualCardActivity::class.java)
intent.putExtra("USER_ID", current.id) //this is not enforced
startActivity(intent)
```

Avec Jetpack Navigation

```
//compile-time check, we can't forget to pass the id
val directions = AdminFragmentDirections.adminToUserDetail(current.id)
view.findNavController().navigate(directions)
```

IMPLÉMENTATION SUR ANDROID

ANGULAR

- Framework frontend modulaire
- Composants réutilisables
- *TypeScript* rend la programmation plus agréable grâce au typage
- *Angular Material* comme librairie visuelle

IMPLÉMENTATION ANGULAR

BACKEND ET AUTHENTIFICATION

- API REST servant de CRUD (Create-Read-Update-Delete) pour l'accès et la modification des données
- Utilisation du framework Express pour gérer l'API REST
- Authentification via JSON Web Token
- Utilisation de la librairie Passport.JS pour l'authentification

BACKEND ET AUTHENTIFICATION

BACKEND ET AUTHENTIFICATION

DÉPLOIEMENT AVEC DOCKER

DÉMONSTRATION

Pockethepia

pockethepia.maximelovino.ch/admin/users

Maxime Lovino

PocketHepia

- Home
- Transactions
- Access

Admin

- Users
- Logs
- Access

New user

1 Personal infos — 2 Permissions Optional — 3 Confirmation

Full Name *

Email *

Password *

Next

Import users

Here you can import a CSV file containing a list of users to import with headers on the first line for the different properties of the users. You can use this template with the headers already inserted and fill in the users before uploading it back. For the boolean fields, fill in with 0 (false) and 1 (true).

Upload file

Filter

Name	Email	Permissions
Cafet Novae	cafet@hepia.ch	Accepts payments
Maja Stajic	maja@stajic.com	
Maxime Lovino	maximelovino@gmail.com	Admin
Tommy Schneider	thomas@java.com	

Pockethepia

pockethepia.maximelovino.ch/admin/users

Maxime Lovino

PocketHepia

- Home
- Transactions
- Access

Admin

- Users
- Logs
- Access

New user

1 Personal infos — 2 Permissions Optional — 3 Confirmation

Full Name *

Email *

Password *

Next

Import users

Here you can import a CSV file containing a list of users to import with headers on the first line for the different properties of the users. You can use this template with the headers already inserted and fill in the users before uploading it back. For the boolean fields, fill in with 0 (false) and 1 (true).

Upload file

Filter

Name	Email	Permissions
Cafet Novae	cafet@hepia.ch	Accepts payments
Maja Stajic	maja@stajic.com	
Maxime Lovino	maximelovino@gmail.com	Admin
Tommy Schneider	thomas@java.com	

AMÉLIORATIONS POSSIBLES

- Intégration au système d'authentification AAI
- Intégration avec NEBIS et la bibliothèque
- Réalisation d'une Progressive Web App ou application Electron pour la partie web

CONCLUSION

- Apprentissage de nouvelles technologies
- Perfectionnement dans le développement mobile
- Projet intéressant et qui peut être étendu par la suite

QUESTIONS