

Grenoble INP - ENSIMAG

École nationale supérieure d'informatique et de mathématiques appliquées de Grenoble

RAPPORT DE STAGE ASSISTANT INGÉNIEUR

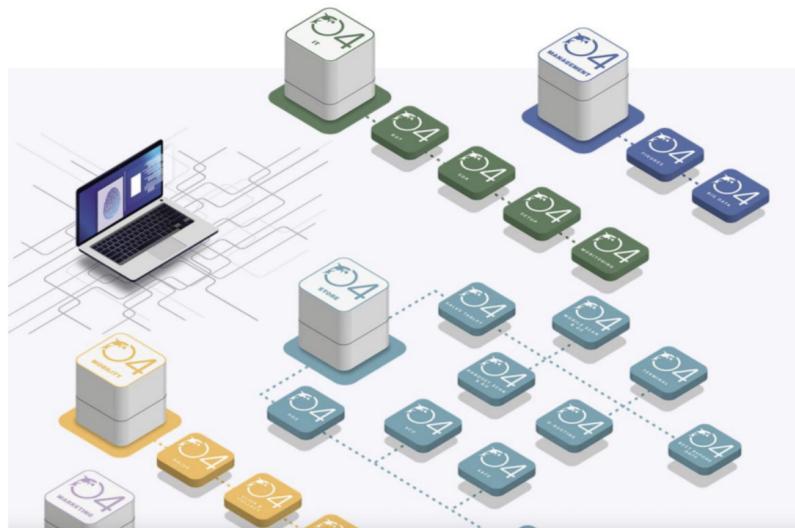
Effectué chez ORIKA

18 mai 2022 - 10 septembre 2022, 4 mois

Développement des applications web-mobiles pour le secteur d'encaissement : o4act , tuipos , o4iz

Zineb Et-tarraf

2 ème année, Option Modélisation Mathématique, Image, Simulation



ORIKA

Immeuble EUROPEEN sis 19
Boulevard Eugène Deruelle
69003 Lyon

Responsable de Stage

Ludovic HOARAU

Tuteur de l'école

Xavier Nicollin

Remerciements

Tout d'abord, je tiens à remercier chaleureusement mon maître de stage, **Mr Ludovic Hoarau**, responsable de l'équipe de Lyon, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Et surtout les discussions que nous avons eues sur ses missions ainsi que ses futurs projets. De ces échanges pleins d'enthousiasme et d'ambition je tire un exemple dans le professionnalisme. Un exemple d'une personne qui travaille avec passion, qui peut travailler jusqu'à des heures décalées, une personne qui travaille sur les projets de l'entreprise comme s'ils étaient les siens. Grâce aussi à sa confiance, j'ai pu m'accomplir pleinement dans mes missions.

Je tiens également à exprimer mes plus sincères remerciements à **l'équipe** à laquelle j'appartenais qui a facilité mes démarches et assuré un suivi tout au long de mon stage.

Enfin, un grand merci à **Fouad MAZOUZ, Rehian MAZOUZ, Alexandre ALGOEDT** pour m'avoir donné l'opportunité de rejoindre l'entreprise et pour leur flexibilité dans le processus de recrutement.

Table des matières

Introduction	1
1 Structure d'accueil	2
1.1 L'entreprise Orika	2
1.2 L'organisation de l'entreprise	3
2 Application web de saisie des rapports d'activité : o4act	3
2.1 La conception UX de o4act et la conception de sa base de données :	4
2.2 Le développement backend et frontend de o4act et la liaison entre les deux :	5
2.3 Déploiement de l'application o4act en mode production : Création du docker et l'automatisation des scripts :	8
2.4 La réalisation de l'application :	8
3 Développement d'un frontend de caisse en mode console : tuipos	9
3.1 Le développement de tuipos :	10
3.2 La réalisation de tuipos :	10
4 Application mobile pour la reconnaissance de produit : o4iz	11
4.1 La première version de o4iz :	11
4.2 La deuxième version de o4iz :	13
5 Impression personnelle	14
6 Conclusion	15
Bibliographie	17
A Liste des technologies adoptées par l'entreprise	18
B La fonction loadData	19
C Les écrans de o4act	20

Introduction

En proposant des solutions multicanales natives, robustes et globales, l'entreprise Orika démontre la force de son logiciel dans le domaine de la grande distribution. Son logiciel principal est la solution d'encaissement Oriane, il regroupe tous les logiciels de caisses, sco (caisses automatiques), gestion des stocks, gestion des commandes, analyse des chiffres d'affaires, etc.

L'évolution de ce logiciel ainsi que sa maintenance sont l'objectif principal de l'entreprise et dans le cadre du traitement des bugs remontés par les clients j'ai participé en équipe de trois au développement de l'outil **tuipos** qui facilite le débogage du côté moteur des caisses .

L'entreprise investit également dans la recherche et elle a réussi à développer un modèle de machine learning pour la reconnaissance de produits par caméra dans ce cadre j'ai développé le frontend d'une application appelée **o4iz** qui sert à montrer aux clients le bon fonctionnement de ce modèle. Pour ce faire, j'ai collaboré avec les personnes qui ont développé le modèle et le backend.

Avant de m'attaquer au développement de **tuipos** et **o4iz** et dans le cadre de la maîtrise des technologies adoptées par l'entreprise - Annexe A, j'ai développé une application appelée **o4act** à usage interne et qui sert à la saisie des rapports d'activité quotidiens .

Ces projets m'ont permis de me familiariser avec l'industrie et les réalités du métier d'ingénieur ainsi qu'avec les mécanismes et les valeurs de l'entreprise.

Ce rapport est organisé comme suit : Nous proposons d'abord une présentation de la structure d'accueil, Puis une présentation de mes trois sujets de stage et nous finalisons ce rapport par un bilan personnel de mon expérience ainsi qu'une conclusion générale.

1 Structure d'accueil

1.1 L'entreprise Orika

Orika est une entreprise française spécialisée dans le développement de logiciels pour le secteur de la grande distribution. Fondée en 2001 par Fouad MAZOUZ, Orika a développé une solution multicanale native, robuste, globale et standardisée à destination des entreprises du commerce quelle que soit leur taille . Orika est présente dans plusieurs pays à l'international notamment l'Italie, l'Afrique subsaharienne et la Maroc où elle prend environ 90 % de la part du marché.

Le siège social d'Orika est basé à la Réunion et dans le cadre d'une stratégie de croissance, elle implante un nouveau siège social à Lyon pour se rapprocher de ses clients métropolitains et pour en acquérir de nouveaux .

Depuis sa création Orika a installé plus de 17 000 POS dans environ 17 pays ou régions du monde , les principaux déploiements ont commencé en France, en Italie, en Belgique, en Pologne et au Maroc avant d'atteindre l'Afrique subsaharienne et l'Asie tout en collaborant avec des entreprises informatiques de premier plan comme NTT DATA et ATOS pour développer des solutions logicielles les mieux adaptées aux particularités du secteur de la grande distribution.

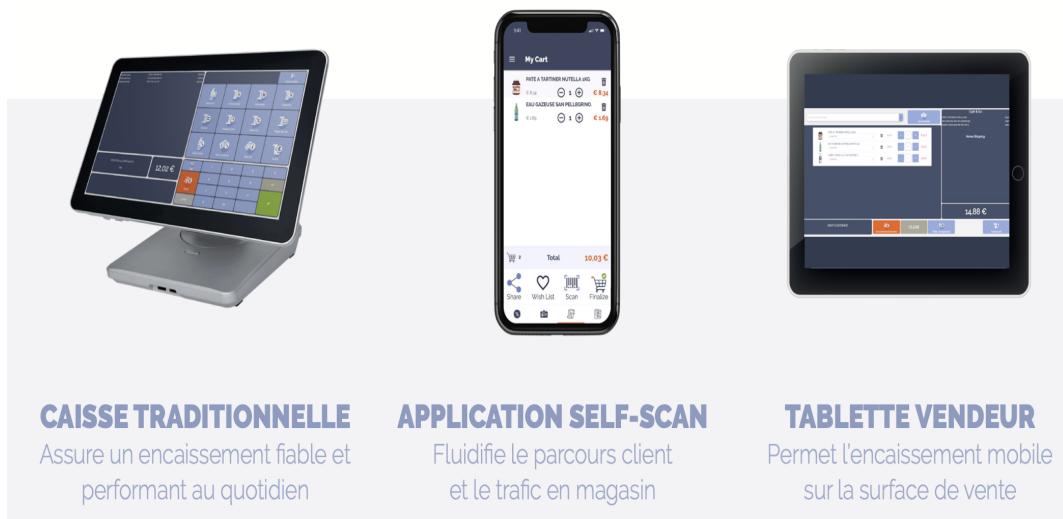


FIGURE 1 – Exemple de produits de l'entreprise .

1.2 L'organisation de l'entreprise

La structure de l'entreprise est basée sur quatre grands pôles :

- **Le pôle de développement** qui joue un rôle collectif pour l'amélioration et la maintenance des logiciels de l'entreprise ainsi qu'une valorisation de son potentiel d'innovation .
- **Le pôle projet** qui établit les relations entre le client et le pôle de développement.
- **Le pôle de recherche** qui est aujourd'hui axé principalement sur l'intégration de l'intelligence artificielle dans les solutions actuelles .
- **Le pôle du support** qui s'occupe du côté hardware à savoir l'installation des logiciels auprès des clients .

2 Application web de saisie des rapports d'activité : o4act

Objectifs

Dans l'objectif de bien assimiler les technologies adoptées par l'entreprise - Annexe A , j'étais amené à développer une application à usage interne .

En effet, les salariés, notamment les stagiaires, qui travaillent potentiellement sur différents sujets devront envoyer un rapport d'activité à leurs responsables à la fin de chaque journée. Un rapport récapitulant les tâches réalisées dans la journée ainsi qu'un plan d'action pour les prochaines tâches.

L'envoi de ces rapports par mail rend difficile leur suivi par les responsables, car ils sont mêlés à des dizaines voire des centaines d'autres mails. De là vient l'idée de créer une application web que nous avons nommée o4act et qui permet aux employés de saisir des rapports d'activité et aux responsables de suivre plus facilement les rapports ainsi que le retard et le non retard des saisies. Cette application propose également un filtre de recherche avancé. C'est une application qui offre quasiment les mêmes fonctionnalités que le webmail.

Ma contribution : au projet consiste essentiellement à réaliser toutes les étapes du projet depuis la définition du besoin, au développement du backend, frontend et base de données jusqu'à la mise en production et la maintenance de l'application , ceci avec l'aide

d'anciens projets de l'entreprise ainsi ainsi que le soutien de mon superviseur et de toute l'équipe.

au projet consiste essentiellement à s'occuper de toutes les parties backend et frontend de l'application avec l'aide d'anciens projets de l'entreprise ainsi que l'accompagnement de mon superviseur et de toute l'équipe.

2.1 La conception UX de o4act et la conception de sa base de données :

Afin d'offrir des expériences significatives et pertinentes à l'utilisateur . On a du commencer par la conception de l'ensemble du processus d'acquisition et d'intégration de l'application . Pour ce faire , le cahier de charges a été traduit en faisant la conception de la maquette de o4act qui représente le design des différents écrans .

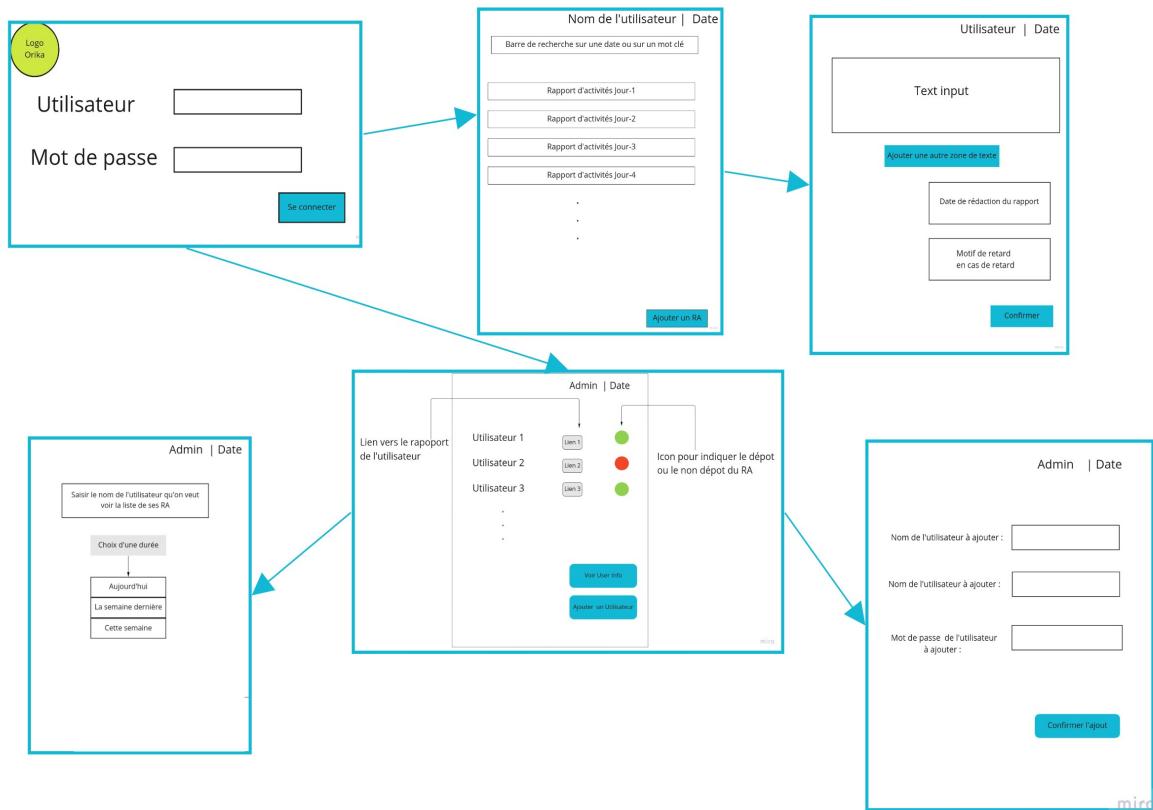


FIGURE 2 – La maquette de l'application o4act.

On a aussi fait la conception de la base de données après avoir fait l'analyse des dépendances fonctionnelles des relations ceci était dans le but d'éviter toute redondance .

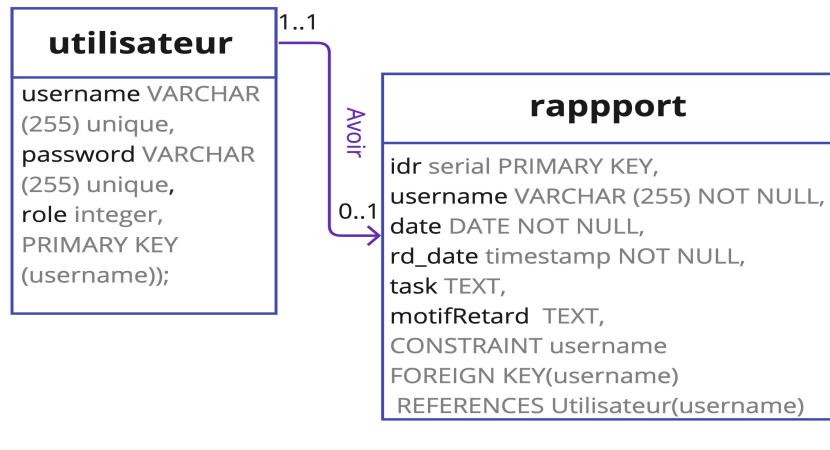


FIGURE 3 - Schémas de la base de données de o4act.

2.2 Le développement backend et frontend de o4act et la liaison entre les deux :

Sans entrer dans les détails nous expliquerons le développement backend et frontend d'o4act et la logique entre les deux.

Le frontend de o4act :

Les navigateurs sont de base des interpréteurs des langages web **html, css, js**. Et ils transforment suite à cette interprétation le code du site en pages interactives que les utilisateurs voient lorsqu'ils visitent un site Web.

Donc dans la partie frontend on code les pages et les objets graphiques en react native qui seront interprétés par un navigateur.

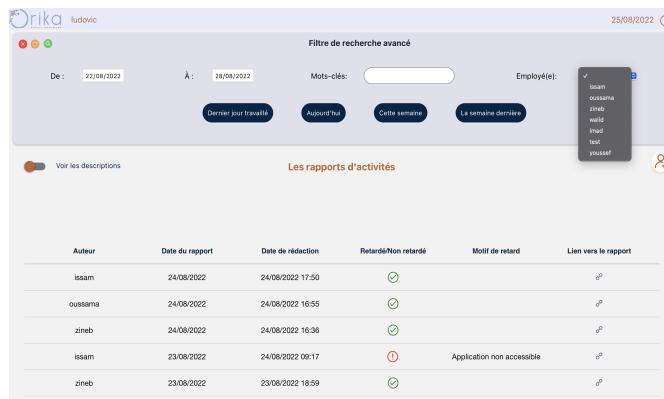


FIGURE 4 - Un navigateur est un simple interpréteur du point de vue du frontend .

Le backend de o4act :

Pour effectuer des actions sur les pages de l'application qu'on a codé dans le frontend et visualiser leurs effets comme les clicks sur les boutons ou l'effet de la barre de recherche... Le frontend interprété par le navigateur a besoin de communiquer avec un serveur web pour valider certaines actions ou pour récupérer certaines données .

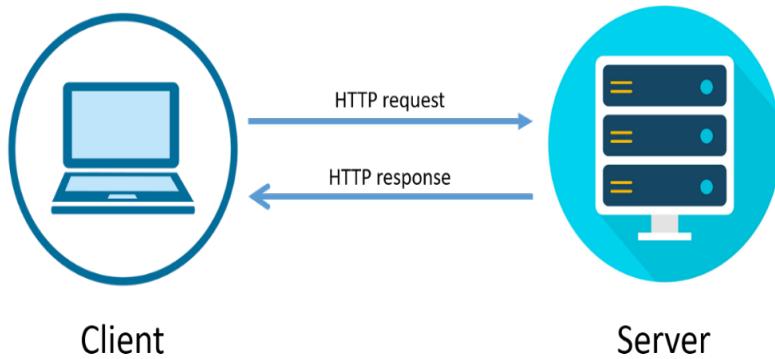


FIGURE 5 - Un navigateur est du point de vue du backend un simple client effectuant des requêtes HTTP.

Pour cela, nous avons implémenté une fonction `loadData` - Annexe B dans le frontend qui prend en paramètre l'url du serveur à interroger et les paramètres à passer au serveur ainsi que la méthode http (post, get ...) à utiliser et la fonction qui va agir sur la réponse du serveur . La fonction `loadData` est appelée à chaque action effectuée dans le frontend et qui nécessite une communication avec le serveur .

Traitement des requêtes HTTP reçues auprès du serveur :

Le navigateur, client, en envoyant une requête HTTP préparée en appelant `loadData` - Annexe B. Il attend un serveur qui comprendra cette requête et qui répondra également en retours. Pour ce faire on crée un serveur web avec tornado .

```

def main():

    settings = dict(
        debug=True,
        static_path=os.path.join(ROOTPATH, "web"),
        template_path=os.path.join(ROOTPATH, "web"),
        cookie_secret="u5SXVuerTfyQTT7uTbu7HjqiqHnh8UsBm37J4Y5lwto=",
    )

    opts = dict(db=db)
    application = tornado.web.Application([
        (r"/(.*)", IndexHandler, opts),
    ], **settings)

    http_server = tornado.httpserver.HTTPServer(application)
    http_server.listen(4000)
    logging.info('Listening on {}'.format(4000))
    tornado.ioloop.IOLoop.instance().start()

```

FIGURE 6 - Création du serveur web qui répond aux actions effectuées dans le frontend.

On crée une application avec le framework python tornado `tornado.web.Application` auquel nous passons en paramètre un dictionnaire `opts` contenant les informations sur le serveur de base de données psql et un gestionnaire `IndexHandler` que nous implémentons pour ce serveur afin qu'il puisse interpréter les requêtes http envoyées en fonction de son étiquette et envoyer la requête psql correspondant au webservice interrogé .

Puis nous créons à partir de cette application un serveur web en écoute sur le port 4000 `tornado.httpserver.HTTPServer` .

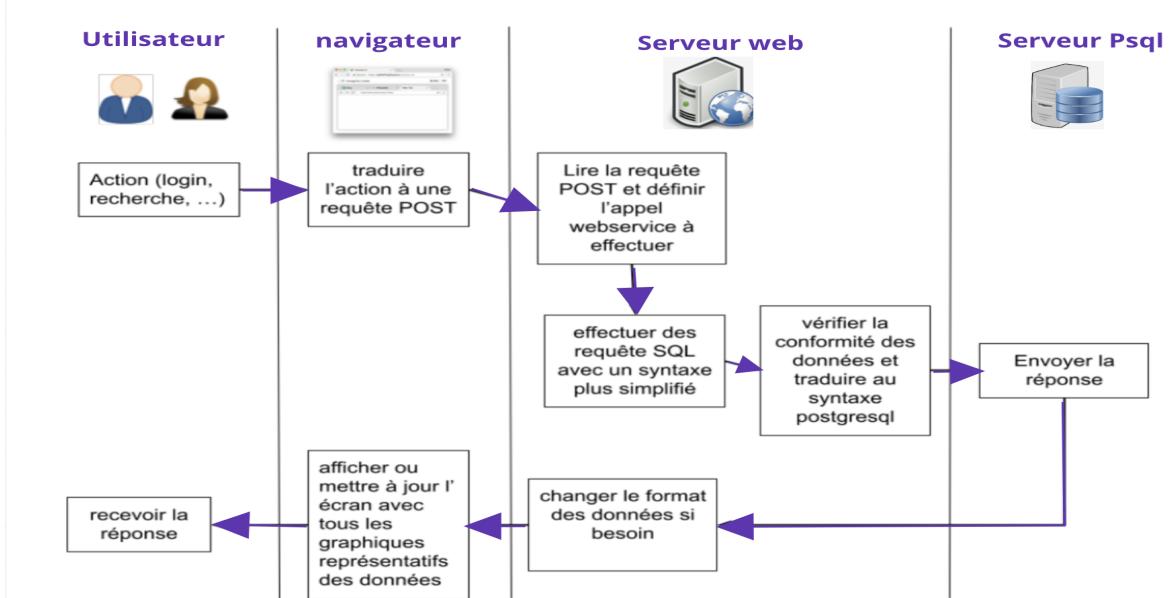


FIGURE 7 - Exemple résumant ce qui se passe dans le frontend, le backend et entre les deux.

2.3 Déploiement de l'application o4act en mode production : Création du docker et l'automatisation des scripts :

Afin de réduire le temps de mise en place d'un environnement pour pouvoir faire tourner l'application, nous avons utilisé docker qui permet de créer une machine virtuelle avec uniquement les technologies utilisées afin que les ressources utilisées par l'application soient minimale et le déploiement de l'application soit simple. Pour générer le tar de production à partir des fichiers de développement de l'application, on a suivit les étapes suivantes :

- Générer le web-build qui transforme l'ensemble du frontend en un index.html et qui prend soin de réduire la taille ; des centaines de Mo à des dizaines de Mo .
- Ajouter la méthode get dans le serveur du backend .

```
@gen.coroutine
def get(self, path = ''):
    if path == 'o4act':
        self.render("index.html")
```

- Créer le docker build le docker compose-up .
- Faire un script qui automatise toutes les étapes précédentes et génère le tar de production .

Après la génération du tar de production il doit être testé sur plusieurs machines virtuelles vierges avant de lancer son déploiement dans le serveur .

2.4 La réalisation de l'application :

The screenshot shows the main dashboard of the o4act application. At the top, there is a header with the logo 'Orika' and the name 'ludovic'. Below the header, there is a search bar labeled 'Filtre de recherche avancé' with fields for 'De' (22/08/2022), 'À' (28/08/2022), 'Mots-clés', and 'Employé(e)'. A dropdown menu is open under 'Employé(e)' showing names: issam, oussama, zineb, wald, imad, test, and youssef. There are also buttons for 'Dernier jour travaillé', 'Aujourd'hui', 'Cette semaine', and 'La semaine dernière'. Below the search bar, there is a button 'Voir les descriptions' and a link 'Les rapports d'activités'. The main content area displays a table of activity reports:

Auteur	Date du rapport	Date de rédaction	Retardé/Non retardé	Motif de retard	Lien vers le rapport
issam	24/08/2022	24/08/2022 17:50	✓		🔗
oussama	24/08/2022	24/08/2022 16:55	✓		🔗
zineb	24/08/2022	24/08/2022 16:36	✓		🔗
issam	23/08/2022	24/08/2022 09:17	⌚	Application non accessible	🔗
zineb	23/08/2022	23/08/2022 18:59	✓		🔗

FIGURE 8 - Écran d'accueil de o4act - voir la suite des écrans dans l'annexe C

3 Développement d'un frontend de caisse en mode console : tuipos

Objectifs

En effet, il y a plusieurs équipes dans la maintenance des caisses qui s'occupent des bugs signalés par les clients. Dans l'équipe moteur dont je faisais partie et qui est spécialisée dans les bugs et les problèmes liés au moteur de caisse. L'ancienne méthode de débogage consistait à commencer par une analyse des traces de communication entre le client **caisse** et le serveur **moteur** ^a. Ensuite on conclut si le problème vient du moteur ou non, puis on rejoue le scénario du bug dans un test et on procède au débogage.

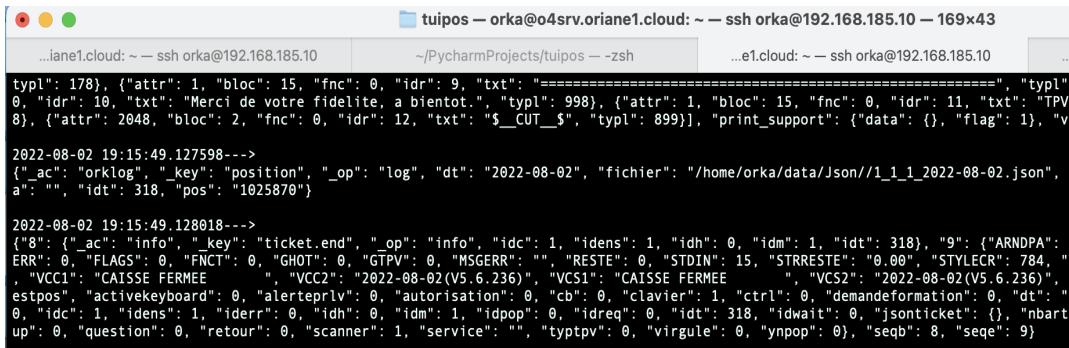
Malheureusement dans de nombreux cas le problème signalé ne vient pas du moteur. Parfois l'origine du problème se situe dans le backend ou le frontend ce qui entraîne une grande perte de temps . De là vient l'idée de développer un outil que nous avons nommé tuipos qui permet de s'affranchir de toutes les couches logicielles complexes et matériels (frontend web, backend, périphérique de caisse). Le but est de se connecter directement au Moteur d'encaissement afin de faire des tests, de la supervisions et diagnostique d'anomalie très rapidement et ainsi en rejouant le bug signalé dans le tuipos on va pouvoir conclure rapidement et avec certitude si le bug vient du moteur ou non et on se lancera directement dans le débogage.

^a. Les caisses graphiques sont reliées par plusieurs backends qui gèrent l'affichage, qui communiquent avec la base des produits du magasin... Dans ce qui suit, nous désignons par moteur de caisse la partie du backend qui est indépendante de la gestion de l'affichage.

Ma contribution : Ce projet était un travail d'équipe de trois : un stagiaire dans sa dernière année, notre superviseur et moi . Nous nous sommes répartis les tâches tout en travaillant sur un même dépôt git.

3.1 Le développement de tuipos :

Dans ce projet, nous avons choisi de travailler avec le langage Go pour plusieurs raisons : Go est facile à apprendre, il est minimal et rapide, et permet un meilleur traitement multithread . Nous avons codé avec Go une interface en mode console simulant une caisse normale. Nous sommes ensuite passés à la mise en action de cette console (effets des touches, communication avec le moteur, etc.). Pour ce faire, nous avons travaillé en parallèle avec les caisses disponibles dans l'entreprise et celles disponibles à distance avec une connexion vnc .En effet, dans les fichiers de trace de la communication caisse-moteur nous analysons la requête envoyée au moteur depuis une caisse normale suite à une action et ensuite nous envoyons la même requête depuis notre console avec golang. Ainsi jusqu'à ce que nous ayons terminé toutes les actions entre caisse-moteur.



```

tuipos - orka@o4srv.oriane1.cloud: ~ -- ssh orka@192.168.185.10 - 169x43
...lane1.cloud: ~ -- ssh orka@192.168.185.10 - /PycharmProjects/tuipos -- zsh ...e1.cloud: ~ -- ssh orka@192.168.185.10 ...

typ": 178}, {"attr": 1, "bloc": 15, "fnc": 0, "idr": 9, "txt": "=====, "typ": 0, "idr": 10, "txt": "Merci de votre fidelite, a bientot.", "typ": 998}, {"attr": 1, "bloc": 15, "fnc": 0, "idr": 11, "txt": "TPV 8}, {"attr": 2048, "bloc": 2, "fnc": 0, "idr": 12, "txt": "$_CUT$_", "typ": 899}], "print_support": {"data": {}, "flag": 1}, "v
2022-08-02 19:15:49.127598-->
{"_ac": "orklog", "_key": "position", "_op": "log", "dt": "2022-08-02", "fichier": "/home/orka/data/Json//1_1_1_2022-08-02.json", "a": "", "idt": 318, "pos": "1025870"}}

2022-08-02 19:15:49.128018-->
{"8": {"_ac": "info", "_key": "ticket.end", "_op": "info", "idc": 1, "idens": 1, "idh": 0, "idm": 1, "idt": 318}, "9": {"ARNDPA": ERR: 0, "FLAGS": 0, "FNCT": 0, "GHOT": 0, "GTPV": 0, "MSGERR": "", "RESTE": 0, "STDIN": 15, "STRRESTE": "0.00", "STYLECR": 784, "VCC1": "CAISSE FERREE ", "VCC2": "2022-08-02(V5.6.236)", "VCS1": "CAISSE FERREE ", "VCS2": "2022-08-02(V5.6.236)", "estpos", "activekeyboard": 0, "alerteprlv": 0, "autorisation": 0, "cb": 0, "clavier": 1, "ctrl": 0, "demandeformation": 0, "dt": 0, "idc": 1, "idens": 1, "iderr": 0, "idh": 0, "idm": 1, "idpop": 0, "idreq": 0, "idt": 318, "idwait": 0, "jsticket": {}, "nbartup": 0, "question": 0, "retour": 0, "scanner": 1, "service": "", "typtpv": 0, "virgule": 0, "ynpop": 0}, "seqb": 8, "seqe": 9}

```

FIGURE 9 - Exemple de fichier de trace de la communication entre caisse-moteur

3.2 La réalisation de tuipos :

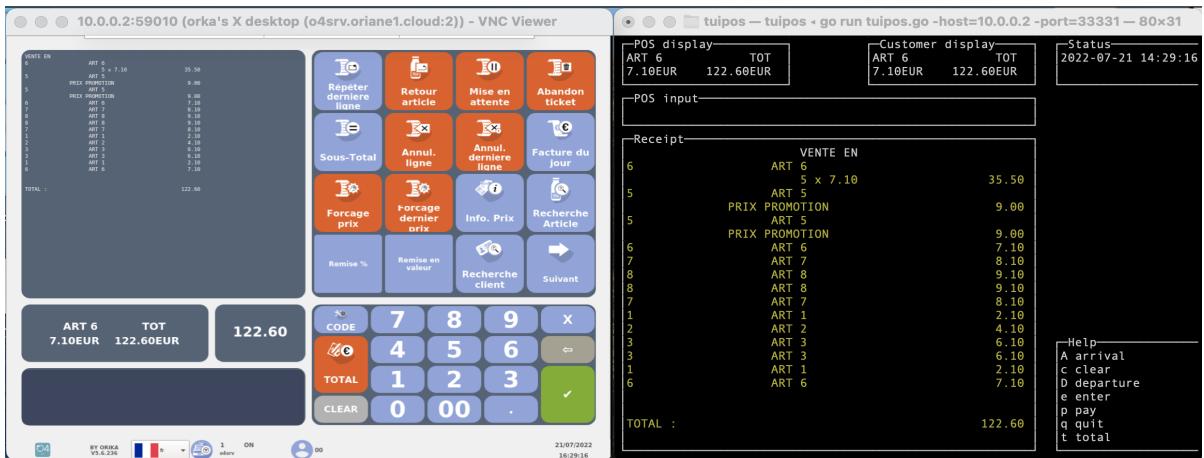


FIGURE 10 - Exemple d'un ticket joué en même temps¹ par une caisse normale et par la caisse en mode console que nous avons mis en place .

- Grace à une connexion web-socket en mode broadcast entre le moteur et les deux caisses .

4 Application mobile pour la reconnaissance de produit : o4iz

4.1 La première version de o4iz :

Objectifs

Dans le cadre de la recherche et du développement, l'entreprise a développé un modèle de machine learning qu'elle a nommé modèle de fraude et qui permet la reconnaissance des produits par caméra.

Dans le but de tester le bon fonctionnement de ce modèle ainsi que de fournir à l'équipe commerciale un outil puissant et simple pour montrer aux clients le bon fonctionnement du modèle de fraude. J'ai développé une application mobile qui permettra aux utilisateurs d'identifier l'article contenu dans une image.

Ma contribution : Ayant déjà développé le backend de cette application par l'équipe de recherche. Et tout en collaborant avec cette dernière, ma mission dans ce projet était de développer le frontend d'o4iz.

Le développement et réalisation de o4iz-version1 :

Nous avons développé la première version d'o4iz de manière à ce qu'elle permette aux utilisateurs de configurer l'API à interroger (choix de l'adresse IP et du port de l'API) et de choisir la taille de l'image dans la compression que nous allons faire .

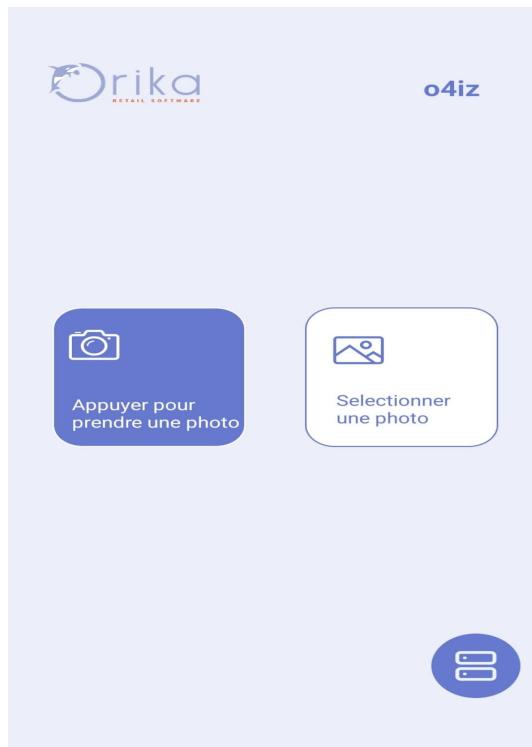


FIGURE 11 - Pour la prédiction, l'utilisateur doit choisir l'image à prédire soit depuis la galerie de son téléphone soit en prenant une photo rapide.

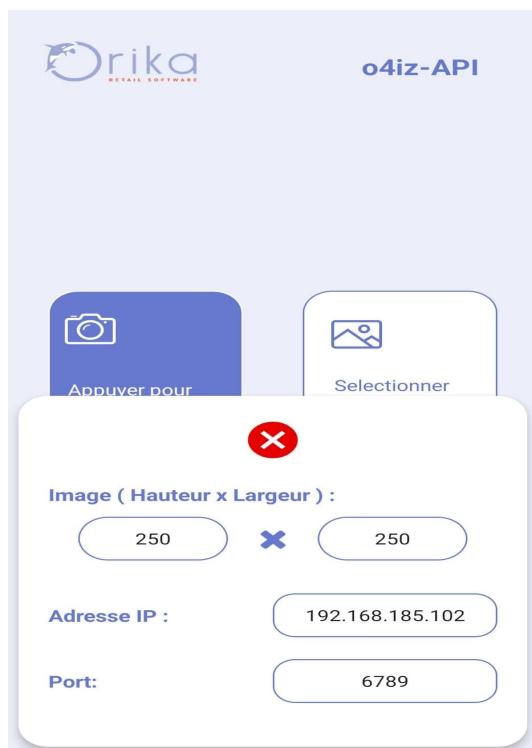


FIGURE 12 - L'utilisateur a à sa disposition un bouton de configuration de l'API à interroger .



FIGURE 13- Exemples de prédiction de l'image d'un cahier prise par camera et de l'image d'une canette sélectionnée depuis la galerie

4.2 La deuxième version de o4iz :

Objectifs

Afin de se détacher de l'API nous prévoyons une deuxième version pour o4iz qui permettra une prédiction et un téléchargement de modèle dans la locale de l'application.

Ma contribution : est de m'occuper de toutes les modifications à apporter à la première version.

Le développement et la réalisation de o4iz-version2 :

Dans cette version d'o4iz, l'utilisateur a été autorisé à entrer le lien à partir duquel on récupère le modèle de prédiction et il a à sa disposition un bouton pour le téléchargement et un bouton pour importer le modèle dans locale de l'application.

Ensuite nous avons travaillé sur l'intégration de tensorflow lite la librairie avec laquelle le modèle a été réalisé dans expo.



FIGURE 14 - L'utilisateur doit faire la configuration pour l'intégration du modèle dans l'application.

5 Impression personnelle

Mon stage de quatre mois chez Orika a été une expérience pleine d'apprentissage et de mise en pratique des connaissances acquises. J'ai eu la chance de participer à deux gros projets de production et un autre projet à usage interne.

J'ai commencé le premier sujet avec peu de connaissances dans le domaine du développement web-mobile. J'ai donc dû apprendre au fur et à mesure, que ce soit par des recherches personnelles ou en échangeant avec d'autres personnes de l'équipe ainsi qu'avec mon référent professionnel. Ce faisant, j'ai grandement développé mes capacités d'adaptation et d'autonomie dans mon travail.

Ensuite j'ai participé dans des projets de production qui se différenciaient grandement des projets académiques vu qu'ils touchent directement les clients et ainsi l'exigence sur les

moindres détails du projet est très élevée. Pour Orika, la caisse est un composant critique de leur solution d'encaissement. Une anomalie du logiciel peut faire perdre de l'argent à leur Client. Par exemple, dans un magasin, si il y a une anomalie bloquante en caisse avec une longue file d'attente de client, l'encaissement ne se fera plus ou sera ralenti.

En ayant déjà de solides capacités de communication et d'écoute. J'ai pu développer de plus en plus tout au long de mon stage ma capacité à apprendre rapidement. En devenant capable de comprendre de nouvelles informations à un rythme très rapide. Un apprentissage qui va au-delà d'une simple compréhension vers une application rapide voire immédiate. Cela se voit clairement dans le fait que j'ai pu accomplir pleinement mes missions tout en travaillant avec des langages et des frameworks avec lesquels je n'avais jamais travaillé et dont l'apprentissage était parallèle aux tâches assignées.

Enfin, je peux confirmer que cette expérience a été beaucoup plus enrichissante que je ne l'aurais pensé. En effet j'ai eu la chance d'évaluer rapidement et positivement dans un environnement favorable, avec une très bonne ambiance. C'était une bonne préparation au monde professionnel.

6 Conclusion

L'objectif de ce rapport était de présenter les trois projets : o4act, tuipos, o4iz dans lesquels je suis investi ainsi qu'un bilan de mon expérience personnelle durant les quatre mois de stage.

Enfin, je peux conclure que cette expérience de stage a été très constructive. En effet, j'ai eu la chance de développer une connaissance globale du domaine de la grande distribution et de l'organisation administrative d'un magasin. J'ai aussi eu la chance de travailler avec une équipe sympathique et très professionnelle. Parmi les tâches constructives qui m'ont le plus impressionné, il y a la tâche de débogage côté moteur, où j'ai rencontré les bugs les plus difficiles de ma vie. En effet, j'étais censé déboguer côté serveur (moteur) avec juste un accès au code côté client (caisse) et aux traces de communication client-serveur. Le bug dont je parle était une rupture anormale dans une connexion web-socket client-serveur. Ainsi j'ai pu développer mes compétences en débogage.

En vrai ,ce stage était une expérience très réussie et une confirmation de mes choix de carrière.

Bibliographie

- <https://javascript.info/websocket>
- <https://fr.wikipedia.org/wiki/JavaScript>
- <https://w3techs.com/technologies/details/cp-javascript>
- <https://www.npmjs.com/package/@switt/react-native-tensorflow-lite>
- <https://blog.tensorflow.org/2020/02/tensorflowjs-for-react-native-is-here.html>
- <https://docs.expo.dev/versions/latest/sdk/filesystem/>
- <https://docs.expo.dev/versions/latest/sdk/document-picker/>
- <https://docs.expo.dev/versions/latest/sdk/imagepicker/>
- <https://docs.docker.com/engine/reference/commandline/build/>
- <https://docs.docker.com/compose/gettingstarted/>
- <https://oblador.github.io/react-native-vector-icons/>
- <https://docs.expo.dev/versions/latest/sdk/imagemanipulator/>
- <https://www.npmjs.com/package/@react-native-async-storage/async-storage>
- <https://docs.expo.dev/versions/latest/sdk/asset/>
- <https://www.psycopg.org/>
- <https://go.dev/doc/>
- <https://gobyexample.com/goroutines>
- https://en.wikipedia.org/wiki/Virtual_machine

Annexe A

Liste des technologies adoptées par l'entreprise

Système d'exploitation :

- Ubuntu Linux
- LVM

Outils :

- Sed , Vim , Git
- Réseau , Iptables ,Systemd

Services :

- Nginx , PXE , DHCP

Développement python :

- Python3 , Psycopg2 , Tornado

Développement web :

- JSON
- ReactJS

IDE :

- PyCharm ,
- Visual code

Base de données :

- PostgreSQL

Virtualisation :

- VirtualBox
- Docker ,Docker compose

Annexe B

La fonction loadData

```
export const loadData = (server , url, method, param, finalFunction) => [
  let Url = url;
  let reponse = {};
  let port = server.port
  let adress = server.adress

  fetch("http://"+adress +":"+port+ Url, {
    method: method,
    Accept: "application/json",
    "Content-Type": "application/json",
    body: (method == "POST") || (method == "PUT") || (method == "DELETE") ? JSON.stringify(param) : null,
  })
    .then((response) => response.json())
    .then((data) => {
      reponse = data;
    })
    .catch((error) => {
      throw error
    })
    .finally(() => {
      if (finalFunction) finalFunction(reponse);
    });
];
```

FIGURE 15 - L'implémentation de la fonction loadData

Annexe C

Les écrans de o4act

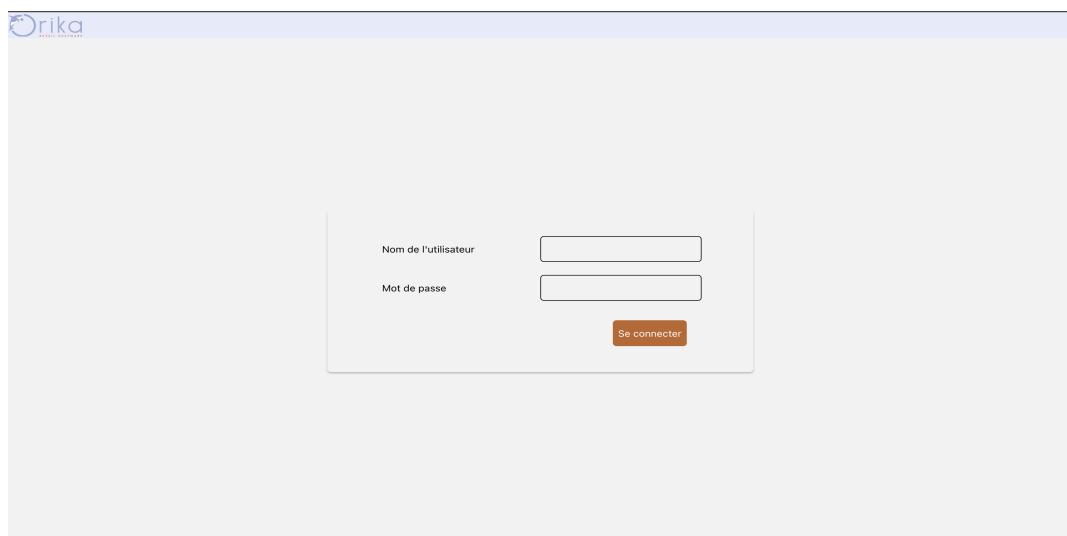


FIGURE 16 - L'écran de connexion à o4act

A screenshot of the o4act dashboard. At the top, there's a search bar and a date range from "22/08/2022" to "28/08/2022". Below that is a "Filtre de recherche avancé" section with buttons for "Dernier jour travaillé", "Aujourd'hui", "Cette semaine", and "La semaine dernière". A dropdown menu for "Employé(e)" shows names: issam, oussama, zineb, walid, imad, test, and youssef. Below this is a section titled "Les rapports d'activités" with a "Voir les descriptions" button. The main content area displays a table of reports:

FIGURE 17 - L'écran d'accueil

FIGURE 17 - L'écran du saisie du rapport d'activité

FIGURE 18 - L'écran pour ajouter un utilisateur , cet écran n'est accessible que dans le compte admin

Les rapports d'activités

Date du rapport	Date de rédaction	Retardé/Non retardé	Motif de retard	Lien vers le rapport
19/08/2022	19/08/2022 17:20	✓		🔗
<p>Bonjour</p> <p>Aujourd'hui j'ai ajouté les infos de l'image mais je trouve toujours pas comment faire pour la taille. les infos stockées par expo ne contiennent pas la taille .Mais j'ai fait la résolution et le format . J'ai généré l'apk et j'ai rencontré un pb c'est qu'en mode prod la caméra ne fonctionne pas comme prévu je l'ai résolu et je vous ai envoyé l'apk par mail .</p> <p>Bon weekend Zineb Et-tarraf</p>				
17/08/2022	17/08/2022 05:23	✓		🔗
<p>Bonjour</p> <p>Aujourd'hui j'ai résolu le problème de l'encodage de l'image .J'ai résolu aussi quelques bug d'installation des packages nécessaires pour le backend et j'ai discuté avec Oussama sur les paramètres à passer au backend . Ayant un backend et un frontend qui fonctionnaient bien j'ai rencontré un problème où la requête n'arrive pas au serveur .On a passé pas mal de temps moi et Issam pour trouver l'erreur . Maintenant l'application marche sauf qu'il y a quelques détails que je vais voir demain et qui concerne l'éteint de la caméra et quelques autres détails .</p> <p>Cordialement Zineb Et-tarraf</p>				

FIGURE 19 - Masquer et afficher rapidement les rapports à l'aide du bouton de haut à gauche .