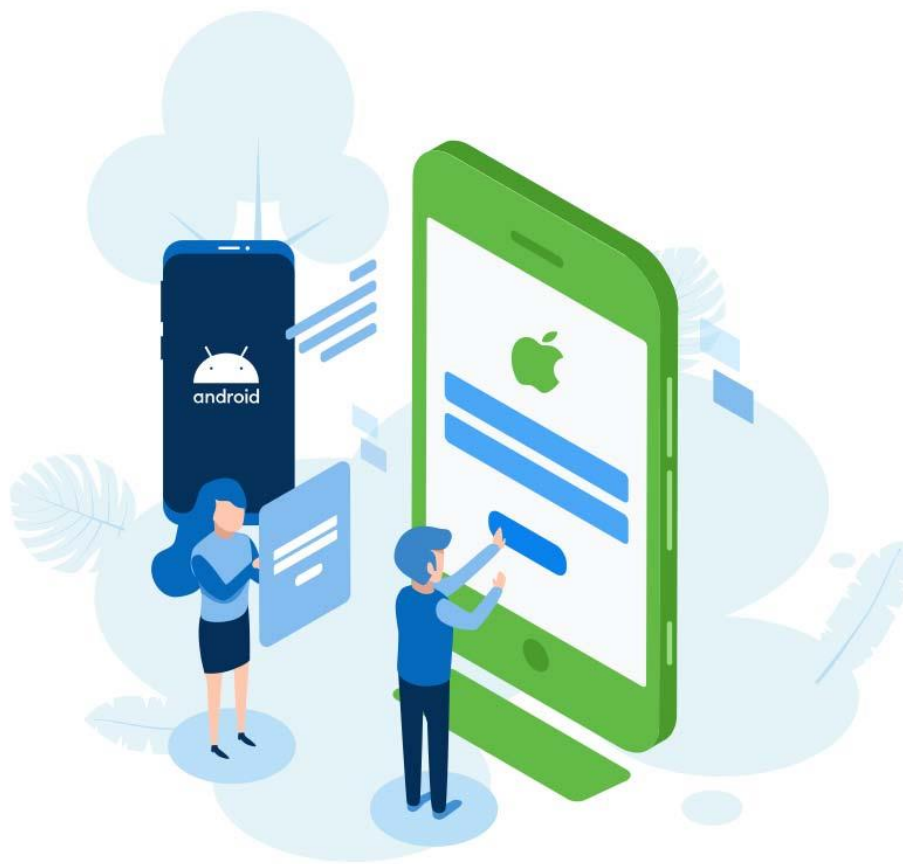


Développement d'une application mobile multiplateforme



Koetschet Thierry
Chemin du Perrey 22
1670 Ursy
thierry.koetschet.1998@gmail.com

cpnv
Centre professionnel du Nord vaudois
École d'Informatique

SI-CA2a

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	3
1.2	Organisation	4
1.3	Méthode de gestion de projet	5
1.4	Objectifs.....	5
1.5	Planification initiale	6
1.6	Structure du dossier.....	7
2	Analyse / Conception.....	7
2.1	Description des frameworks	7
2.1.1	Apache Cordova	7
2.1.2	React Native	7
2.1.3	NativeScript	8
2.1.4	Flutter	8
2.2	Analyse des frameworks.....	8
2.3	Choix du framework.....	9
2.4	Concept	10
2.4.1	Modèle conceptuel de données	10
2.4.2	Modèle logique de données.....	11
2.5	Stratégie de test.....	12
2.6	Risques techniques	12
2.7	Planification	13
2.8	Dossier de conception	15
3	Réalisation.....	16
3.1	Dossier de réalisation	16
3.2	Installation de NativeScript	17
3.3	Aperçu de l'application.....	22
3.4	Description des tests effectués	24
3.5	Erreurs restantes	26
4	Conclusions	27
5	Annexes.....	28
5.1	Cahier des charges.....	28
5.2	Résumé du travail.....	31
5.3	Sources – Bibliographie.....	32
5.4	Journal de travail	33
5.5	Glossaire	34
5.6	Manuel d'installation	34

1 Analyse préliminaire

1.1 Introduction

Le but de ce travail consiste tout d'abord à rechercher différents frameworks permettant de faire du développement d'application mobile, puis de les analyser et enfin de les tester afin de choisir l'outil le plus pertinent selon mes critères. Ce choix est important car le framework que je choisirai sera utilisé pour réaliser mon TPI.

Dans un deuxième temps, je programmerai une petite application avec le framework choisi afin de me préparer pour l'application que je développerai pour le TPI. L'idée de cette application est de permettre aux habitants de Sainte-Croix de commander des repas dans les différents restaurants de la région et de se les faire livrer à domicile. Cette application s'inspirera des géants du domaine tels que Uber eats, Just Eat ou encore Deliveroo.

La raison de mon choix de développer une application mobile s'explique par le fait que je suis très intéressé par la programmation mais j'avais également envie de faire quelque chose de nouveau, que l'on n'a pas vu en cours. J'ai alors réalisé que de créer une application mobile remplissait les deux conditions et j'ai donc rempli ma fiche signalétique en conséquence.

1.2 Organisation

Projet	Date début	Participants		Activités	Tâches
		Nom	Prénom		
Développement d'une application mobile multiplateforme	31.01.2023	Koetschet	Thierry	1 Choix framework, langage et IDE	11 Installation Cordova 12 Création projet test Cordova 13 Installation React Native 14 Création projet test React Native 15 Installation Native Script 16 Création projet test Native Script 17 Installation Flutter 18 Création projet test Flutter 19 Comparaison et choix du meilleur framework
				2 Modélisation de la base de données	21 Conception MCD 22 Conception MLD 23 Création de la DB 24 Hébergement de la DB sur un serveur 25 - 26 - 27 - 28 - 29 - 30 -
				3 Développement de l'application	31 Structure des pages du projet 32 Connexion à la base de données 33 Page d'accueil 34 Page d'affichage des restaurants 35 Page d'affichage des catégories de plats 36 Rédaction des tests 37 Application des tests et considération des résultats 38 - 39 - 40 -
				4 Documentation et présentation	41 Organisation personnelle 42 Rédaction de la documentation 43 Préparation de la présentation 44 - 45 - 46 - 47 - 48 - 49 -
				5 Autres	51 Absence 52 - 53 - 54 - 55 - 56 - 57 - 58 -

Cette image est tirée du fichier « planification_pre-TPI_tkt.xlsm » disponible dans le répertoire git de mon projet pré-TPI.

L'organisation de mon projet c'est fait de la manière suivante : J'ai tout d'abord partager mon travail en use cases pour définir les quatre majeures parties. La première partie concerne tout ce qui s'apparente à la l'analyse, les tests et le choix entre les frameworks pour continuer la suite du projet. Le deuxième use case englobe toutes les tâches relevant de la modélisation de la base de données. Le troisième concerne la partie pratique du projet, c'est-à-dire la création et le développement de l'application mobile. Enfin le dernier use case va composer du travail de rédaction de la documentation, de mon organisation personnelle et de la partie présentation du projet.

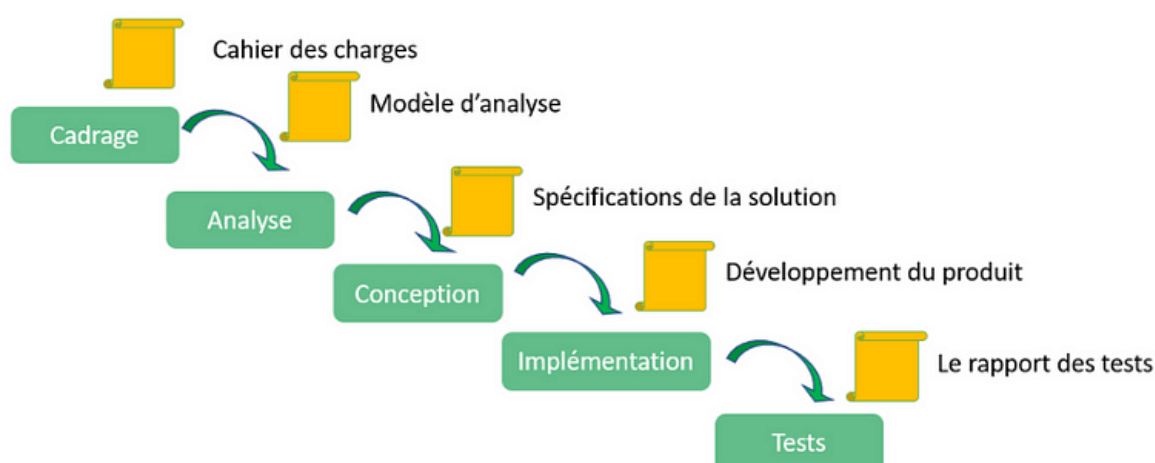
1.3 Méthode de gestion de projet

J'ai choisi la méthode en cascade pour la gestion de mon projet. Je trouve que pour un projet en solo, cette méthode répond parfaitement à mes besoins en terme de gestion de projet.



Blog Gestion de Projet

MÉTHODE EN CASCADE (WATERFALL)



Les cinq étapes de la méthode en cascade correspondent parfaitement à la vision que j'ai pour la réalisation d'un tel projet. En effet, le cadrage en début de projet est primordial afin de structurer notre travail et de ne pas trop s'éparpiller. La partie analyse correspond à ce que j'ai fait en me documentant sur les différents frameworks. Après la partie analyse, j'ai réalisé la conception et l'implémentation de mon application et terminé mon projet par la rédaction de la méthode de tests et réalisation des différents tests par rapport à l'installation du framework et le fonctionnement de la l'application.

1.4 Objectifs

Il est important pour moi d'atteindre certains objectifs lors du pré-TPI : j'aimerais tout d'abord choisir un framework avec lequel je suis assez familier pour pouvoir développer une application mobile d'une certaine envergure. Afin de me familiariser avec le framework en question, j'aimerais programmer une plus petite application me permettant de mieux comprendre le langage de programmation, le fonctionnement du framework et de l'IDE et enfin de pouvoir tester le produit fini.

J'aimerais également acquérir certaines capacités par rapport à la planification et l'organisation d'un tel travail mais aussi concernant la rédaction d'une documentation conforme aux attentes de l'école et du monde professionnel plus généralement.

1.5 Planification initiale

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Projet			Planification																	
	Développement d'une application mobile multiplateforme			31.01.23	07.02.23	20.02.23	27.02.23	06.03.23	13.03.23	20.03.23	27.03.23	03.04.23									
2		Total																			
3	Prévu	57 h 45		9 h 00	9 h 00	9 h 00	6 h 45	1 h 45	1 h 45	5 h 45	5 h 45	9 h 00									
4	Koetschet	27 h 45		9 h 00	9 h 00	9 h 00	0 h 45														
5																					
6			SEM	5	6	8	9	10	11	12	13	14									
7	1 Choix framework, language et IDE																				15 h 30
8																					9 h 00
9		11 Installation Cordova	Prévu		1 h 00																1 h 00
10			Koetschet		1 h 30																1 h 30
11		12 Création projet test Cordova	Prévu		2 h 30																2 h 30
12			Koetschet		2 h 15																2 h 15
13		13 Installation React Native	Prévu		1 h 00																1 h 00
14			Koetschet		1 h 30																1 h 30
15		14 Création projet test React Native	Prévu		2 h 30																2 h 30
16			Koetschet		3 h 45																3 h 45
17		15 Installation Native Script	Prévu			1 h 30															1 h 30
18			Koetschet																		
19		16 Création projet test Native Script	Prévu			2 h 30															2 h 30
20			Koetschet																		
21		17 Installation Flutter	Prévu				1 h 30														1 h 30
22			Koetschet																		
23		18 Création projet test Flutter	Prévu				2 h 30														2 h 30
24			Koetschet																		
25		19 Comparaison et choix du meilleur framework	Prévu				0 h 30														0 h 30
26			Koetschet																		

Le cadre bleu représente les use cases de mon projet. Le cadre rouge montre les tâches à réaliser. Les cadres jaunes indiquent le temps planifier pour chaque tâche et les valeurs temporelles surlignées en vert représente le temps réel utilisé pour chaque tâche. Le reste du tableau contient des formules afin de représenter de manière plus visuelle les différentes données saisies.

Le document complet de planification sera remis en annexe à cette documentation.

1.6 Structure du dossier

Ce dossier est composé de 4 parties principales.

La première partie représente l'analyse préliminaire de ce travail et se compose d'une introduction, d'une explication de l'organisation du projet, une description des objectifs personnels et puis un bref aperçu de ma planification initiale.

La deuxième se compose de toute la partie analyse et conception du projet avec l'élaboration du concept, de la stratégie de test, un compte rendu des risques techniques, une révision de la planification initiale du projet et enfin la partie conception du projet.

La troisième partie représente toute la réalisation pratique du projet commençant par lister tous les fichiers du dossier de réalisation, puis une description des tests effectués et des erreurs restantes et finalement une énumération des documents fournis à la remise du projet.

La dernière décrit les conclusions auxquelles je suis arrivé à la fin de ce projet telles que les objectifs atteints ou non, mon ressenti au fil du projet, les difficultés rencontrées et les améliorations que je pourrais apporter si je devais refaire un tel projet.

2 Analyse / Conception

2.1 Description des frameworks

Un framework est un ensemble de composants logiciels servant à gérer et créer la structure et le corps d'une application. Les différents framework analysés dans ce travail sont spécifiquement utiles à la création d'applications mobiles. Cette analyse, en plus des différents tests effectués sur les frameworks, va permettre faire un choix qui semble le plus adéquat pour la réalisation d'une petite application mobile.

2.1.1 Apache Cordova

Cordova est un framework gratuit et open-source permettant la création d'application mobile multiplateforme avec un seul code de base composé de HTML, CSS et JavaScript. Les premières version d'Apache Cordova en 2011 sous le nom de Fondation Apache, puis Apache PhoneGap.

2.1.2 React Native

ReactNative est tout comme Cordova un framework open-source créer par Meta Platforms (anciennement Facebook). Il permet le développement de logiciel pour plateforme mobile pour des systèmes d'exploitation tels qu'Android ou iOS.

2.1.3 NativeScript

Le framework NativeScript a été créé et mis à disposition par Progress gratuitement. Il sert à coder des applications mobiles avec le langage de programmation JavaScript ou des langages dérivés tels que TypeScript. La particularité de ce framework est qu'il contient une multitude d'API natifs, c'est-à-dire propre à NativeScript

2.1.4 Flutter

Le dernier framework que j'ai testé est Flutter. Il s'agit d'un SDK open-source créé et mis à disposition par Google pour le développement d'applications Android, iOS, Linux, MacOS, Windows, etc. Contrairement aux autres frameworks, Flutter possède son propre langage de programmation : Dart.

2.2 Analyse des frameworks

Au début de ce projet, j'ai décidé de commencer par Apache Cordova. J'ai réalisé l'installation en suivant le guide d'installation mis à disposition sur le site officiel d'Apache Cordova¹. Les premiers problèmes sont apparus lors de l'installation de l'environnement de développement Android Studio. En effet, j'ai rencontré à ce moment des soucis pour créer un AVD nécessaire pour l'affichage de l'application. L'interface de gestion des appareils virtuels me refusait catégoriquement de créer un émulateur. Je me suis rendu compte plus tard dans mon travail en installant d'autres frameworks que le guide d'installation d'Apache Cordova manquait de détails au niveau de la modification des variables d'environnement et que cela en était certainement la cause. De plus, le build de l'application a tout simplement échoué sans pour autant que je ne comprenne la raison, dû certainement à mon manque d'expérience dans le domaine.

J'ai donc décidé de tester l'installation d'un autre framework, React Native. J'ai également suivi la documentation d'installation de l'environnement disponible sur le site de React Native². J'ai alors rencontré les mêmes problèmes qu'avec Cordova. J'ai donc essayé de suivre un tutoriel sur Youtube³. Grâce à la vidéo, je me suis rendu compte de deux problèmes principaux dans mon setup : le premier était que je n'avais pas ajouté de variables d'environnement ce qui empêchait le framework de trouver tous les chemins nécessaires pour fonctionner correctement. Et en configurant ces variables d'environnement, j'ai rencontré un deuxième problème. En effet, j'ai remarqué que le dossier Sdk, donc le dossier contenant le kit de développement de software Android ne se situait pas dans le même dossier que l'IDE Android Studio ce qui rendait la configuration des variables d'environnement relativement compliquée.

La décision qui me semblait alors la plus logique était de refaire une installation de zéro en prenant soin de mettre tous les dossiers concernés dans un même dossier appelé Android à la racine de mon C:\. Pour ce faire, j'ai donc désinstallé Android Studio et supprimé tous les dossiers restants contenant le Android dans leur nom. Puis

¹ Site officiel d'Apache Cordova : <https://cordova.apache.org/docs/en/latest/guide/cli/index.html>

² Documentation d'installation de React Native : <https://reactnative.dev/docs/environment-setup>

³ Lien vidéo d'installation React Native : <https://www.youtube.com/watch?v=oZFCt69Bccc&ab>

j'ai refait toute l'installation en faisant attention à ce que tous les dossiers aient la même racine. Après toute cette procédure, j'ai refait des tests avec React Native. Cette fois-ci, j'ai réussi à créer un AVD et à construire une application vide avec React Native.

J'ai décidé de poursuivre mon analyse et mes tests sur le framework NativeScript⁴. Son installation est similaire aux autres frameworks avec le téléchargement de la librairie, le setup de l'IDE et du SDK qui a déjà été fait précédemment. Après avoir téléchargé la librairie NativeScript avec la commande **npm install -g nativescript**, une autre commande très utile montre un diagnostic très complet des éléments que NativeScript arrive à trouver et les erreurs qu'il rencontre. Cette commande est **ns doctor android** et elle est une des raisons qui a influencé mon choix. Une fois que le diagnostic est totalement fonctionnel, j'ai pu créer, construire et lancer mon application test.

Le dernier framework que j'ai décidé de tester est Flutter⁵. Tout comme NativeScript, son installation a été relativement simplifiée par toutes les étapes d'installation de l'IDE et du SDK fait au début du projet. J'ai pu créer une application test et la lancer sur l'émulateur Android et elle fonctionnait bien malgré quelques latences au niveau de la machine hôte probablement dues au grand nombre de ressources demandé par l'AVD. La seule raison qui a poussé mon choix sur NativeScript est que le langage de programmation avec Flutter est complètement nouveau pour moi et j'ai préféré me diriger sur quelque chose que je connais déjà avec du JavaScript.

2.3 Choix du framework

Comme je l'ai déjà un peu abordé au chapitre précédent, j'ai choisi NativeScript comme framework pour programmer mon application.

La première raison s'explique par le fait qu'il s'agit du framework que j'ai eu le moins de problèmes à installer et à configurer. En effet, sa documentation était particulièrement détaillée par rapport aux autres et comme je l'ai expliqué, la commande **ns doctor android** a rendu le debug de l'installation beaucoup plus aisé.

Deuxièmement, JavaScript est un langage de programmation que je connais assez bien grâce au cours ICT-120 ou encore ICT-306 reçus pendant ma première année de formation.

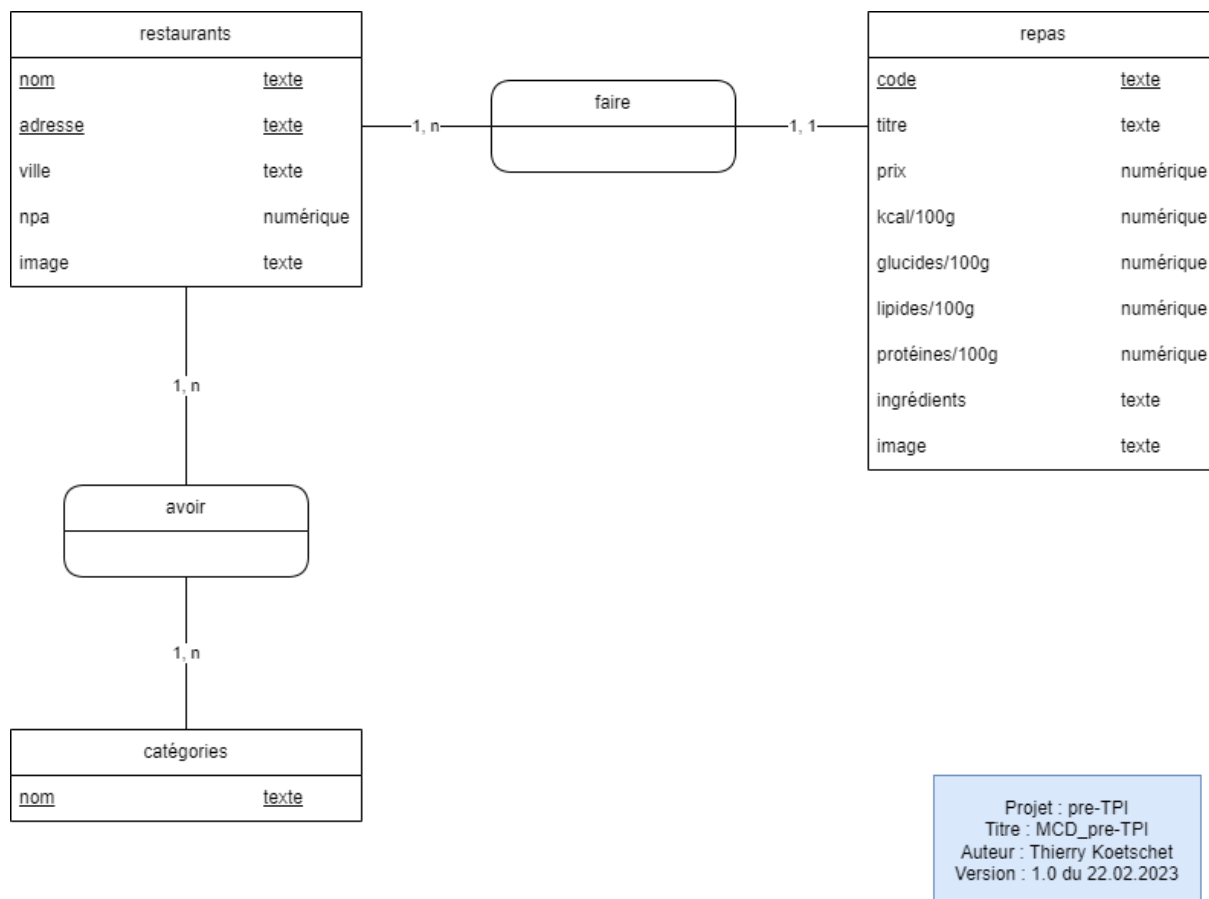
Finalement, NativeScript a répondu à toutes les attentes que j'avais pour la réalisation de mon programme : créer une application multiplateforme, visualiser des tutoriels compréhensibles et disposer d'une documentation d'installation complète et fonctionnelle.

⁴ Documentation d'installation de NativeScript : <https://docs.nativescript.org/environment-setup.html>

⁵ Documentation d'installation de Flutter : <https://docs.flutter.dev/get-started/install>

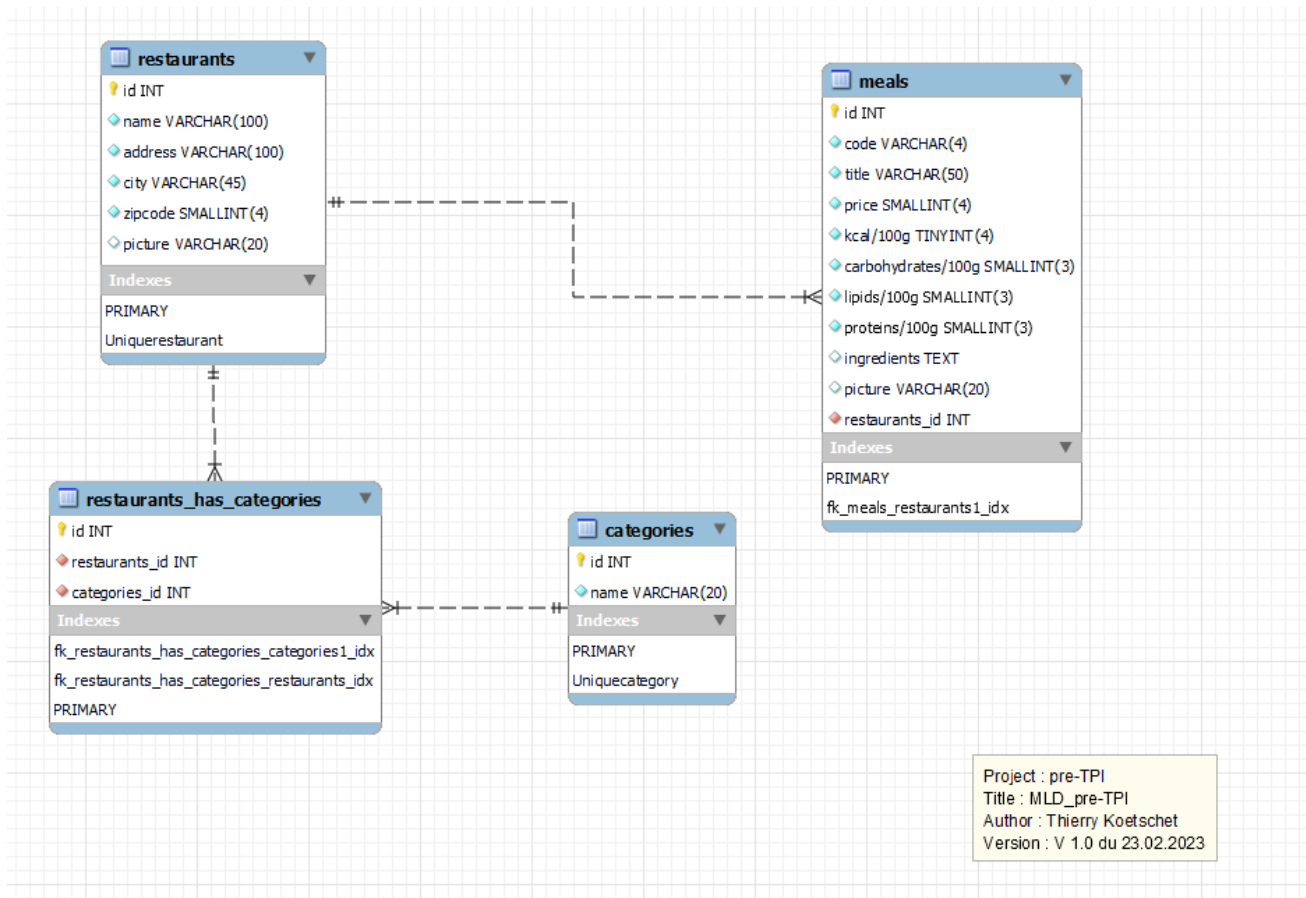
2.4 Concept

2.4.1 Modèle conceptuel de données



Le MCD a été réalisé avec l'application draw.io

2.4.2 Modèle logique de données



Le MLD a été réalisé avec MySQL Workbench 8.0 CE

Celui-ci a permis de générer le script de création de la base de données que j'ai exécuté sur HeidiSQL.

2.5 Stratégie de test

Dans un premier temps, les tests effectués vont me permettre vérifier si l'installation des frameworks est fonctionnelle.

Ma stratégie de test va d'abord se composer de test unitaires. Ceux-ci vont me permettre de vérifier l'état des composants nécessaires et au fonctionnement du framework :

- Le Software Development Kit
- L'environnement de développement
- Les variables d'environnement
- L'émulateur

Ces composants sont communs aux quatre framework mais d'autres composants sont propres à chaque framework, et nécessaire pour leur bon fonctionnement, qu'il faudra aussi pouvoir tester.

Si chaque test unitaire est concluant, je pourrai passer au tests d'intégration. Ceux-ci permettront de contrôler si tous les modules testés individuellement fonctionnent ensemble.

Enfin, je terminerai la réalisation de mes tests par les tests fonctionnels. Ils me permettront de déterminer si les frameworks installés et mon application mobile fonctionnent correctement.

2.6 Risques techniques

Le principal risque technique que j'ai rencontré est le manque de temps. Plus le projet avançait, plus je réalisais que l'installation de framework était chronophage et que j'allais vraisemblablement manqué de temps à la fin du projet. Malheureusement, entre la partie analyse, la partie pratique et la rédaction de la documentation, j'ai décidé de coupé court à la partie pratique afin d'avoir assez de temps pour rendre un document convenable.

Le deuxième risque technique auquel je me suis heurté est mon manque de compétence dans le domaine du développement d'applications mobiles. En effet, l'ajout d'un framework à un projet de développement apporte une toute nouvelle dimension que je n'ai jamais eu l'occasion d'apprendre dans ma formation. C'est ce nouvel aspect qui m'attirait au début du projet mais je me suis vite rendu compte que j'ai été un peu trop ambitieux peut-être que j'aurais sûrement dû me contenter des connaissances acquises en cours afin de garder un cadre connu autour de mon projet.

2.7 Planification

Projet		Planification										
Développement d'une application mobile multiplateforme			31.01.23	07.02.23	20.02.23	27.02.23	06.03.23	13.03.23	20.03.23	27.03.23	03.04.23	
	Total											
Prévu	81 h 00		9 h 00	9 h 00	9 h 00	9 h 00	9 h 00	9 h 00	9 h 00	9 h 00	9 h 00	
Koetschet	67 h 30		9 h 00	9 h 00	9 h 00	9 h 00	9 h 00	9 h 00	9 h 00	4 h 30		
		SEM	5	6	8	9	10	11	12	13	14	
1 Choix framework, language et IDE												15 h 30
												19 h 15
11 Installation Cordova	Prévu		1 h 00									1 h 00
	Koetschet		1 h 30									1 h 30
12 Création projet test Cordova	Prévu		2 h 30									2 h 30
	Koetschet		2 h 15		1 h 30							3 h 45
13 Installation React Native	Prévu		1 h 00									1 h 00
	Koetschet		1 h 30									1 h 30
14 Création projet test React Native	Prévu		2 h 30									2 h 30
	Koetschet		3 h 45									3 h 45
15 Installation Native Script	Prévu			1 h 30								1 h 30
	Koetschet			2 h 15								2 h 15
16 Création projet test Native Script	Prévu			2 h 30								2 h 30
	Koetschet			1 h 30		1 h 00						2 h 30
17 Installation Flutter	Prévu				1 h 30							1 h 30
	Koetschet				2 h 00							2 h 00
18 Création projet test Flutter	Prévu				2 h 30							2 h 30
	Koetschet				1 h 30							1 h 30
19 Comparaison et choix du meilleur	Prévu				0 h 30							0 h 30
	Koetschet						0 h 30					0 h 30
2 Modélisation de la base de données												2 h 45
												3 h 00
21 Conception MCD	Prévu				0 h 45							0 h 45
	Koetschet				0 h 45							0 h 45
22 Conception MLD	Prévu				0 h 45							0 h 45
	Koetschet				0 h 45							0 h 45
23 Création de la DB	Prévu				0 h 45							0 h 45
	Koetschet				1 h 30							1 h 30
24 Hébergement de la DB sur un serveur	Prévu					0 h 30						0 h 30
	Koetschet											
25 -	Prévu											
26 -	Prévu											
27 -	Prévu											
28 -	Prévu											
29 -	Prévu											
3 Développement de l'application												16 h 30
												9 h 30
31 Structure des pages du projet	Prévu					1 h 00						1 h 00
	Koetschet					0 h 45		0 h 45				0 h 45
32 Connexion à la base de données	Prévu					1 h 15	3 h 00					4 h 15
	Koetschet						2 h 30	0 h 45				3 h 15
33 Page d'accueil	Prévu						2 h 15					2 h 15
	Koetschet						3 h 00	1 h 30				4 h 30
34 Page d'affichage des restaurants	Prévu							2 h 30				2 h 30
	Koetschet											
35 Page d'affichage des catégories de plats	Prévu							2 h 30				2 h 30
	Koetschet											

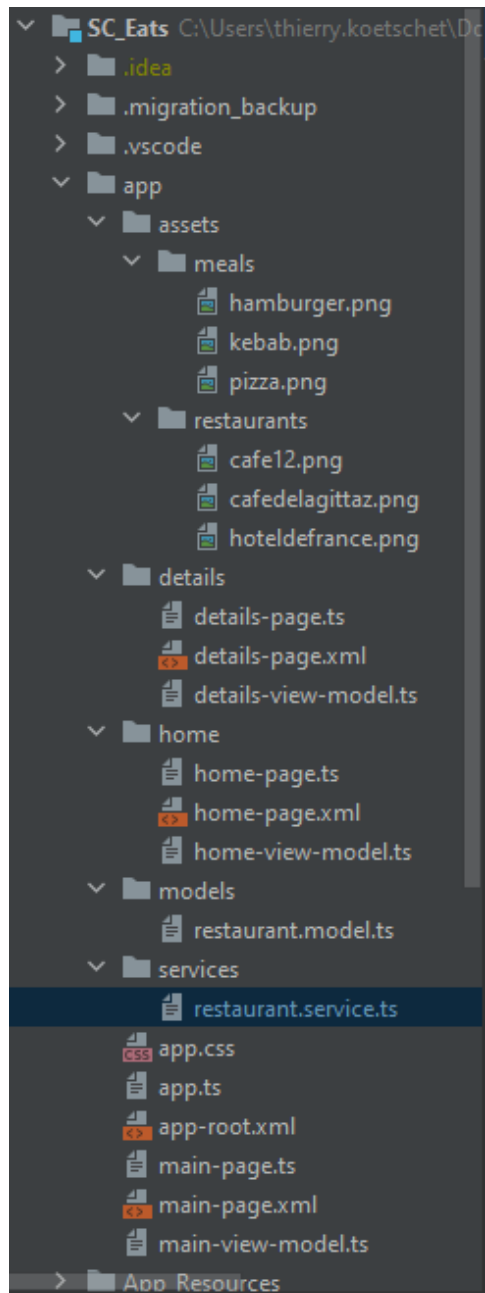
[illegible]

Toute la planification est contenue dans le fichier « planification_pre-TPI_tkt.xlsm » qui englobe également toute la partie organisation des tâches et le journal de travail.

Ce projet a débuté le 31 janvier 2023 et la documentation ainsi que le produit fini, c'est-à-dire l'application mobile, ont été rendus le 29 mars 2023. Le reste du temps prévu pour le projet pré-TPI sera consacré à la présentation orale.

Les valeurs temporelles surlignées en gris représentent la planification initiale et les valeurs surlignées en vert démontrent le temps réel accordé à chaque tâche.

2.8 Dossier de conception

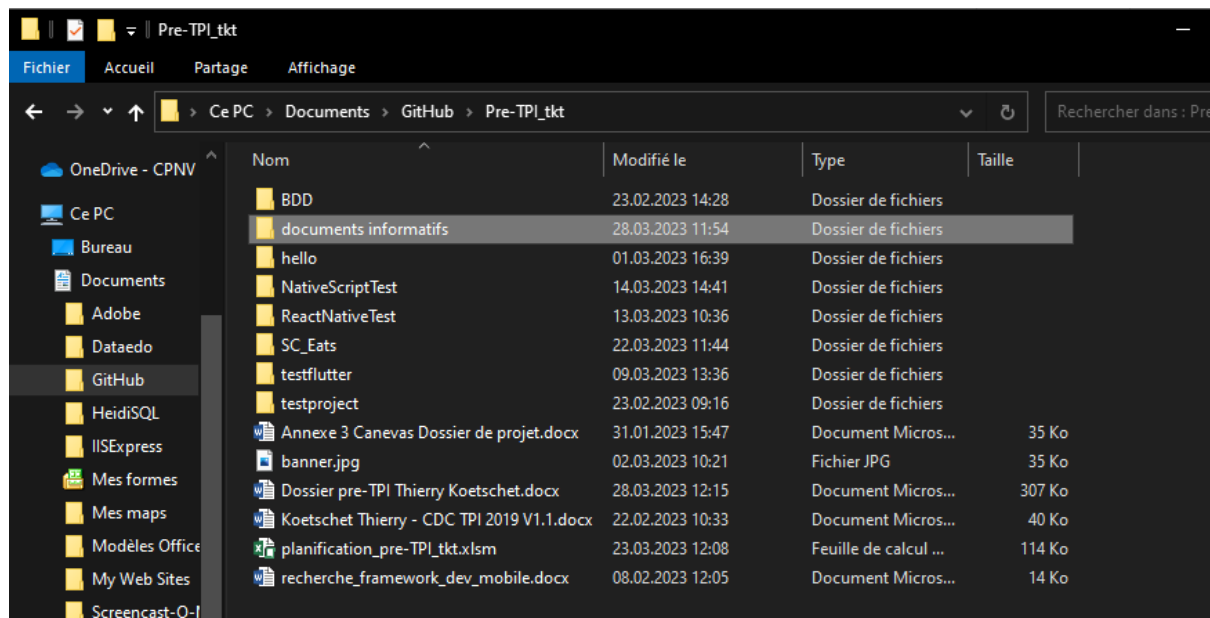


L'image ci-contre démontre les fichiers nécessaires au fonctionnement de l'application mobile.

Tout d'abord, le dossier « assets » contient toutes les images utilisées dans le projet. Le dossier « details » englobe les fichiers nécessaires à l'affichage de la page de détails des restaurants. Le dossier s'appelant « home » contient les fichiers permettant d'afficher la page d'accueil au lancement de l'application. Le fichier « restaurant.model.ts » contient la classe RestaurantModel permettant la création d'objets selon les principes de la programmation orientée objet. Le fichier « restaurant.service.ts » gère la création des objets restaurants et contient toutes les données affichées dans la page d'accueil et la page de détails. Enfin, le reste des fichiers présents dans le dossier « app » servent à la mise en page de l'application.

3 Réalisation

3.1 Dossier de réalisation



La totalité du projet se trouve sur un répertoire GitHub ouvert au public (https://github.com/ThierryKoetschet/Pre-TPI_tkt). L'organisation du répertoire a été fait de la manière suivante :

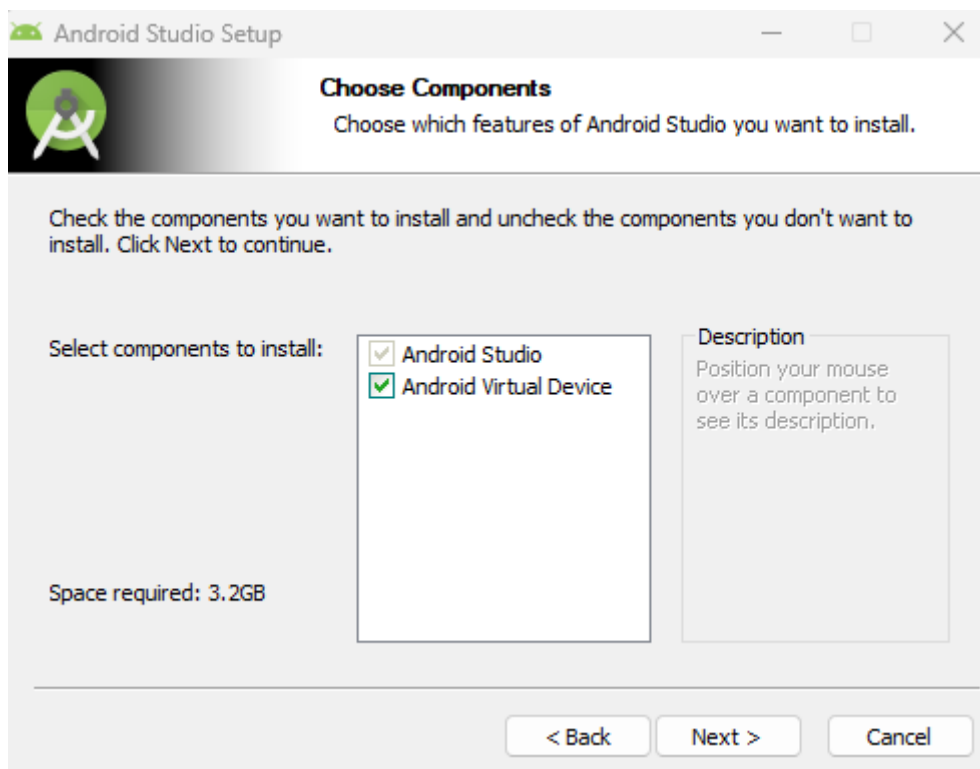
- Tout ce qui concerne la base de données (MCD, MLD, scripts, backups, ...) se trouve dans le dossier « BDD »
- Le dossier « documents informatifs » contient des documents concernant l'évaluation du projet par le responsable de projet et les experts
- Le dossier « hello » contient une tentative de création d'application avec Cordova
- Le dossier « testproject » contient également une tentative de création de projet avec React Native
- Le fichier « planification_pre-TPI_tkt.xlsm » sert à l'organisation et la planification des tâches et contient également le journal de travail.
- Le fichier « Koetschet Thierry – CDC TPI 2019 V1.1.docx » est la première version du cahier des charges transmis par M. Saison pour la réalisation du pré-TPI.
- Ce fichier de documentation se situe également dans le dossier de réalisation sous le nom de « Dossier Pre-TPI Thierry Koetschet.docx »
- Le dossier SC_Eats contient l'application de réalisation du projet
- Le fichier « recherche_framework_dev_mobile.docx » est un travail de recherche effectué sur les différents framework proposés par M. Saison avant que je reçoive mon cahier des charges.

3.2 Installation de NativeScript

Cette marche à suivre ne traite que de l'installation et de la configuration de NativeScript pour Windows 10.

Tout d'abord, il faut télécharger Chocolatey afin de pouvoir installer les dépendances Nodejs et Java Development Kit. Pour ce faire nous devons télécharger la licence sur le site de Chocolatey⁶. Nous pouvons alors saisir la commande **choco install -y nodejs.install temurin11** pour installer les dépendances citées.

L'étape suivante consiste à installer Android Studio et tous ses composants complémentaires tels que Android SDK, Android SDK Platform, Android Virtual Device et Performance (Intel ® HAXM). Ces composants complémentaires sont absolument nécessaires pour le fonctionnement de la future application. Nous pouvons commencer par installer Android Studio en téléchargeant l'exécutable sur le site officiel de Android Studio⁷. Exécutez le fichier téléchargé, puis continuez l'installation.

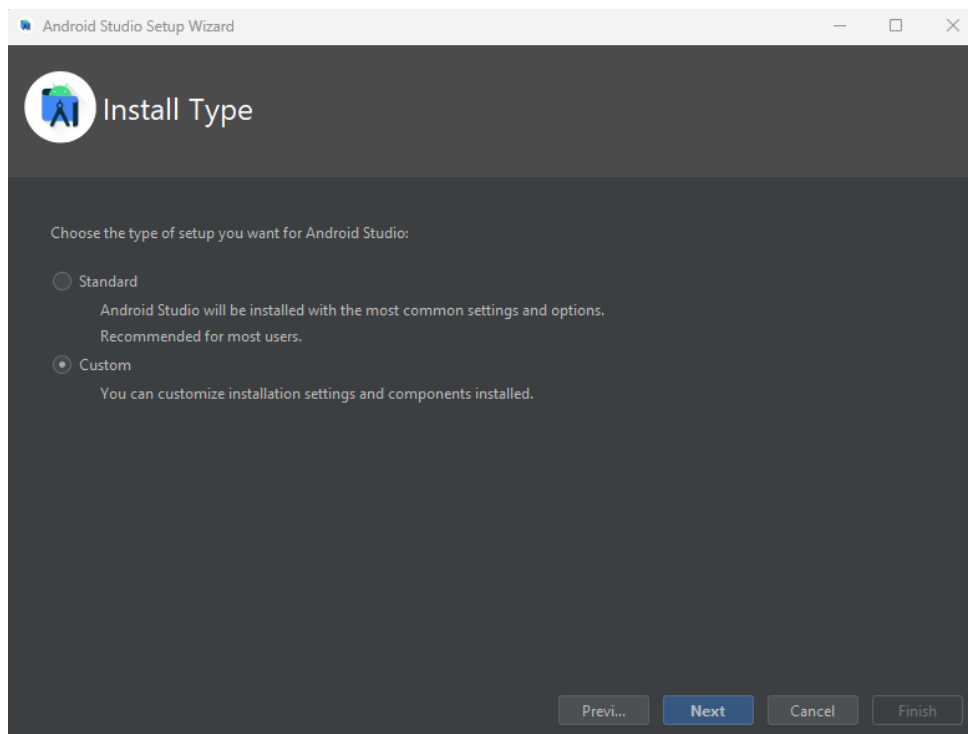


La checkbox Android Virtual Device doit être cochée pour continuer l'installation. Choisissez un dossier d'installation et gardez ce dossier dans un coin de votre mémoire car tous les composants complémentaires devront être installés dans le même dossier afin de nous faciliter la tâche avec les variables d'environnement. Enfin, procédez à l'installation d'Android Studio et lancez l'application. Il faut ensuite passer

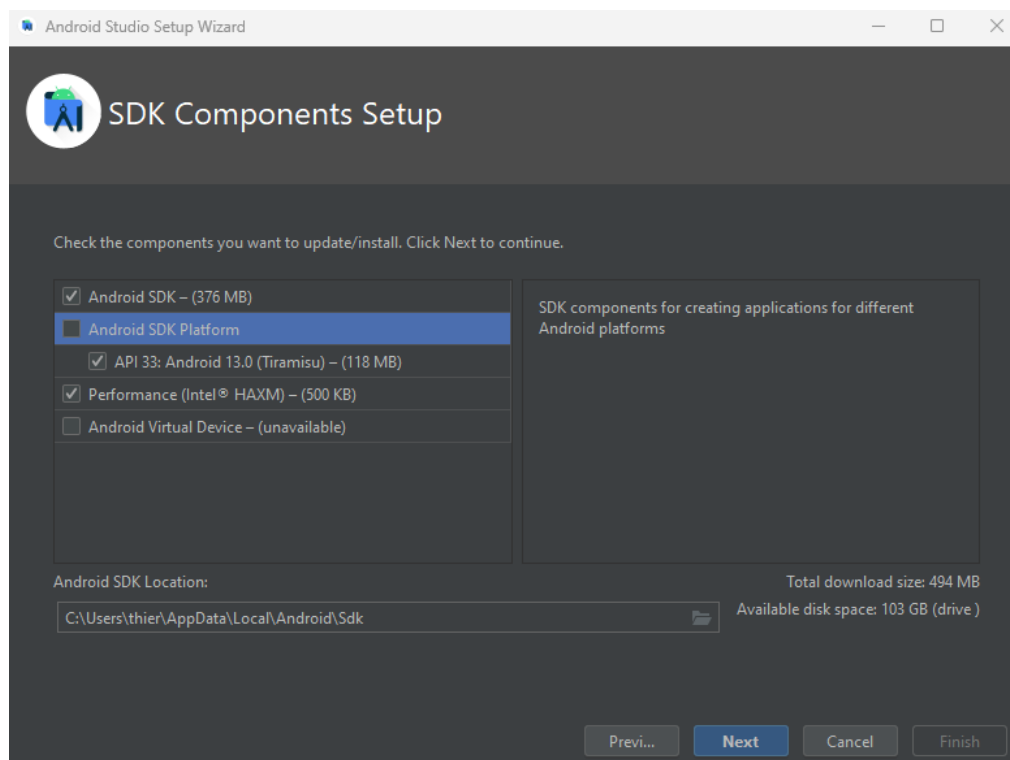
⁶ Site officiel de Chocolatey : <https://chocolatey.org/install#individual>

⁷ Site officiel de Android Studio : <https://developer.android.com/studio>

les prochaines étapes jusqu'à ce que l'on doive choisir le type de configuration que l'on veut faire pour Android Studio.

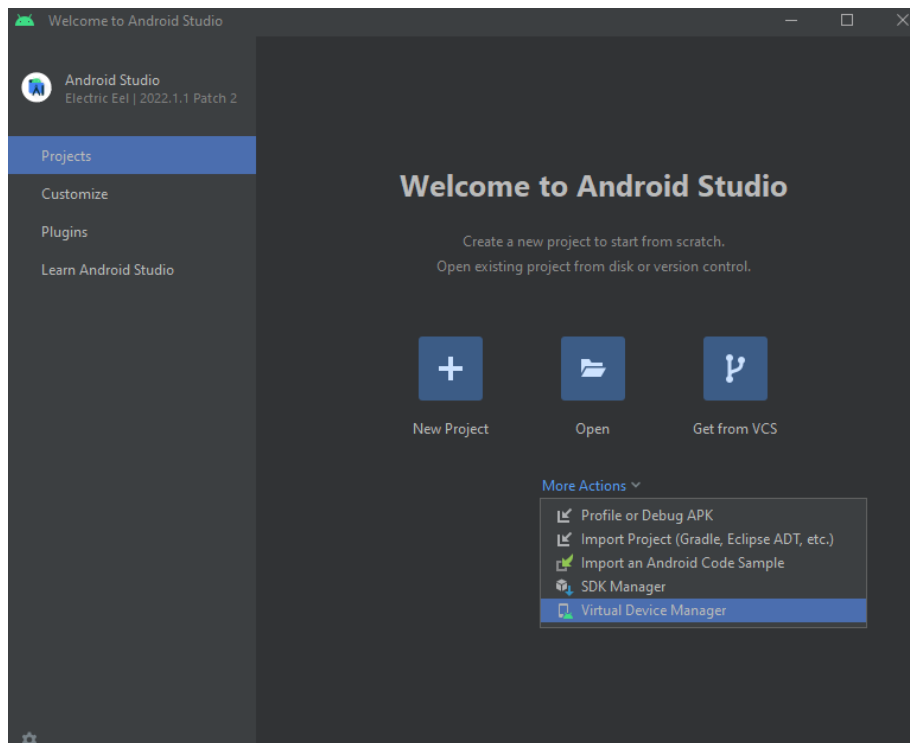


Choisissez l'installation personnalisée. Vérifiez que le dossier contenant le JDK se situe bien dans le dossier créer précédemment et passez à l'étape suivante. Sélectionnez ensuite le thème selon votre préférence et passez.

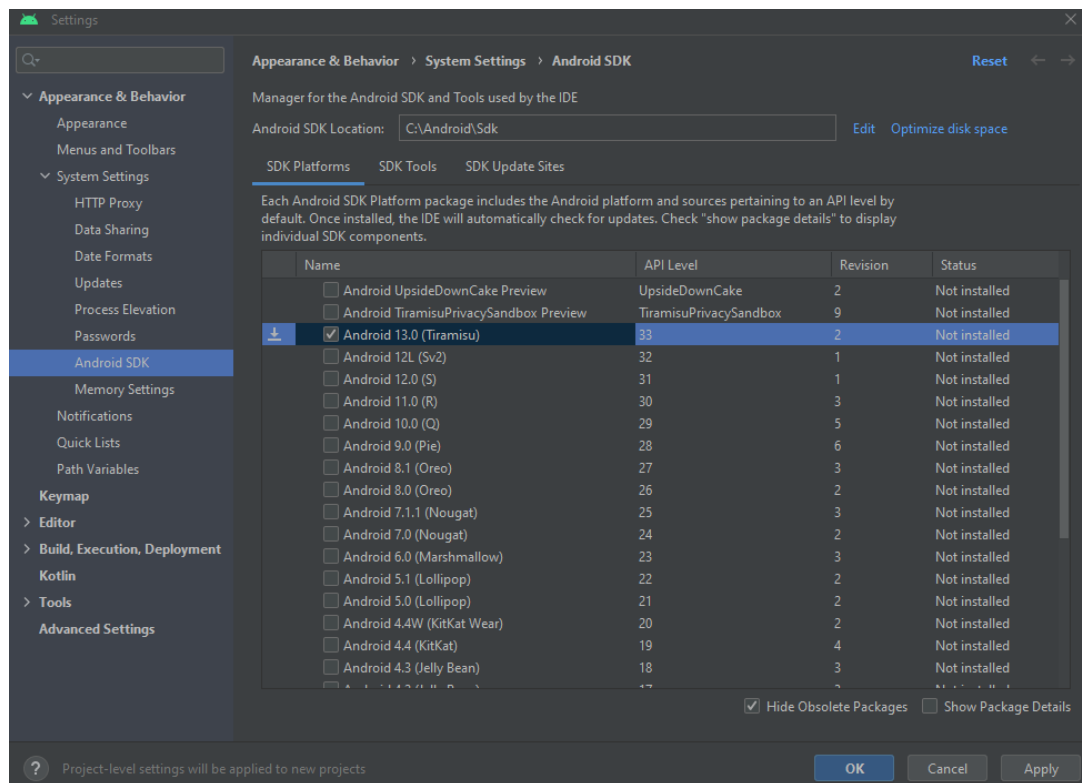


Développement d'une application mobile multiplateforme

Une fois que vous arrivez à cette étape, faites attention à installer tous les composants sélectionnez ci-dessus dans le dossier Android où se situe le dossier Android Studio et procédez à l'installation.



Ensuite, accédez à SDK Manager et installez Android 13.0 (Tiramisu) de l'onglet SDK Platforms s'il n'est pas déjà installé. Il servira d'API pour l'émulateur.



Nous allons maintenant configurer les variables d'environnement pour permettre au framework NativeScript de trouver Android SDK. Pour ce faire, éditez les variables d'environnement système dans la barre recherche Windows, cliquez sur « Variables d'environnement ... » et, sous variables utilisateur, créez une nouvelle variable nommée **ANDROID_HOME** avec comme valeur le dossier Sdk situé dans le dossier Android que nous avons créé au début de la marche à suivre. Modifiez ensuite la variable « Path » et ajoutez une nouvelle variable **[Dossier Android]\Sdk\platform-tools**.

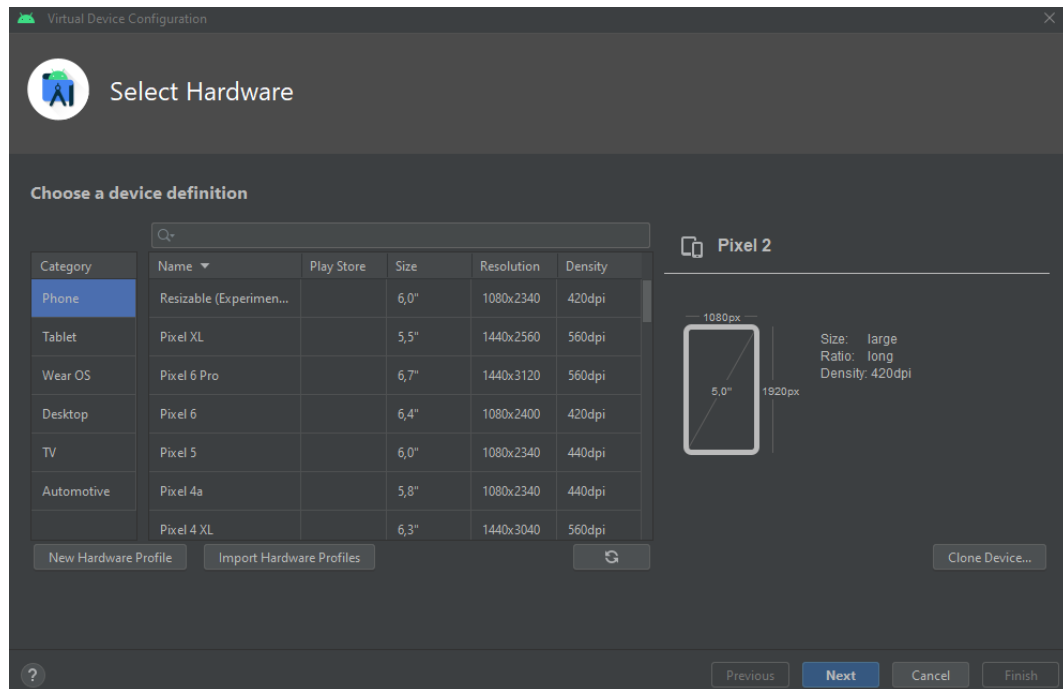
Nous pouvons maintenant installer NativeScript CLI en saisissant la commande **npm install -g nativescript** dans un invite de commande. La commande **ns doctor android** permet de vérifier si tous les composants nécessaires au fonctionnement de NativeScript sont installés ou configurés.

```
C:\Users\thier>ns doctor android
✓ Do you want to help us improve NativeScript by automatically sending anonymous usage statistics? We will not use this
information to identify or contact you. ... no
✓ Getting environment information

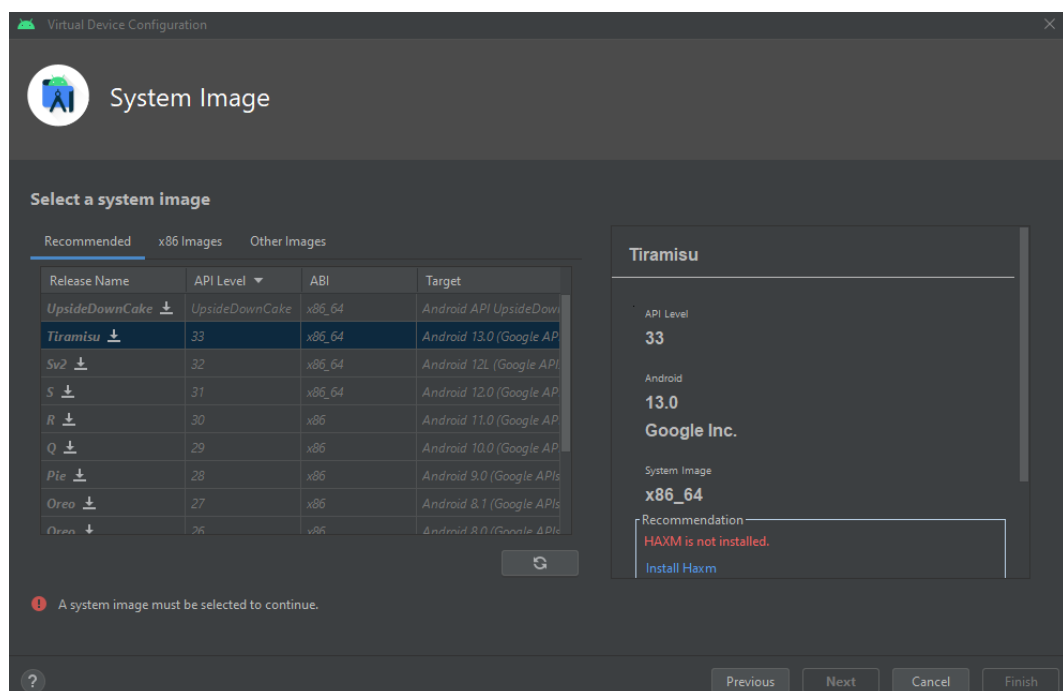
No issues were detected.
✓ Your ANDROID_HOME environment variable is set and points to correct directory.
✓ Your adb from the Android SDK is correctly installed.
✓ The Android SDK is installed.
✓ A compatible Android SDK for compilation is found.
✓ Javac is installed and is configured properly.
✓ The Java Development Kit (JDK) is installed and is configured properly.
✓ Local builds for iOS can be executed only on a macOS system. To build for iOS on a different operating system, you can
use the NativeScript cloud infrastructure.
✓ Getting NativeScript components versions information...
✓ Component nativescript has 8.5.0 version and is up to date.
```

Si vous avez respecté la marche à suivre jusqu'ici, toutes les vérifications de la commande **ns doctor android** devraient être validées comme sur la capture ci-dessus.

L'ultime étape consiste à créer un AVD (Android Virtual Device) pour pouvoir visualiser l'application mobile en taille réelle. Pour ce faire, accédez au Virtual Device Manager de Android Studio et cliquez sur « Create device ».



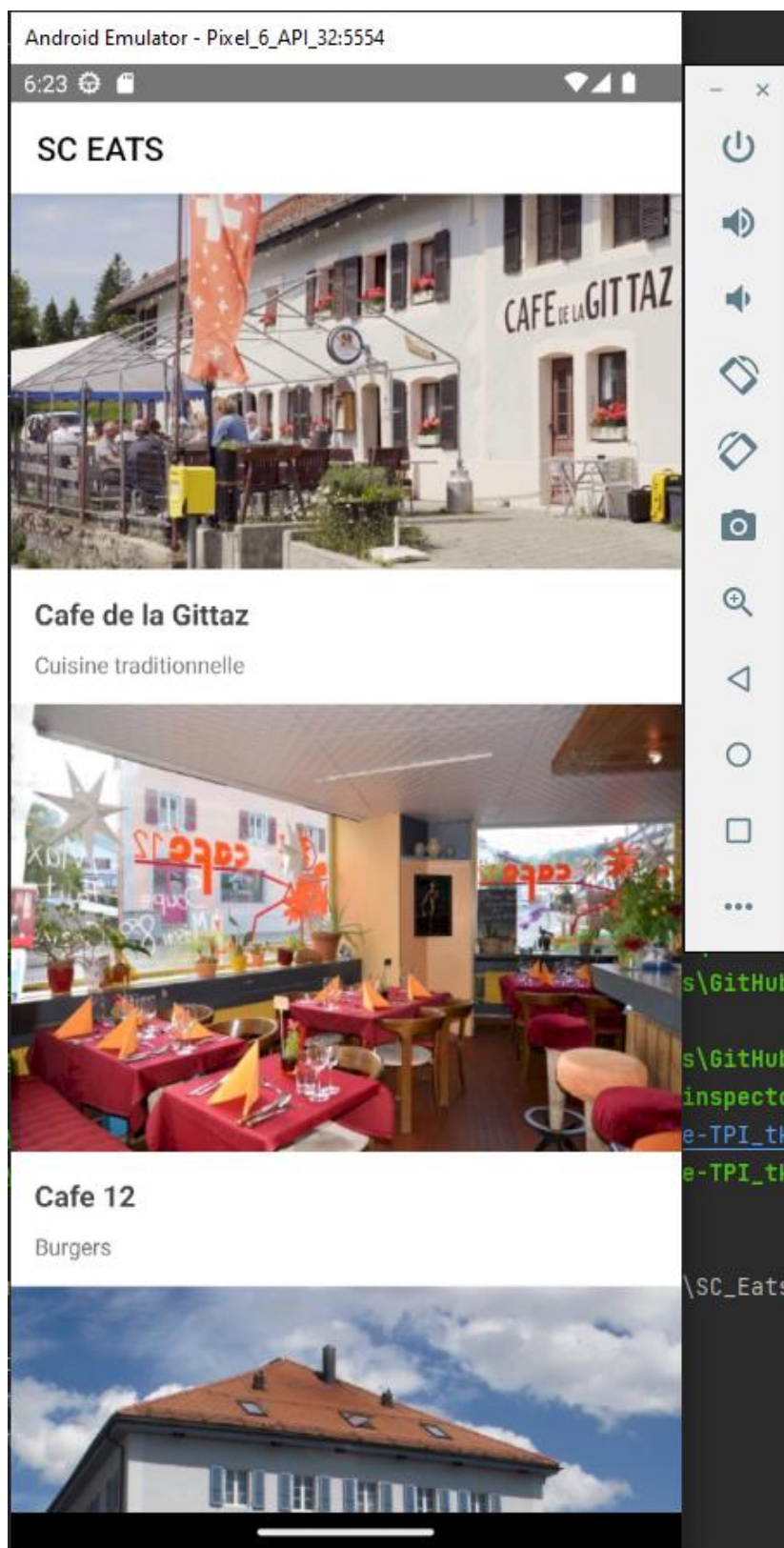
Sélectionnez ensuite le téléphone que vous souhaitez émuler. J'ai personnellement choisi le Pixel 6.



Sélectionnez alors l'image système Tiramisu pour correspondre au choix de la plateforme SDK et créez l'appareil virtuel.

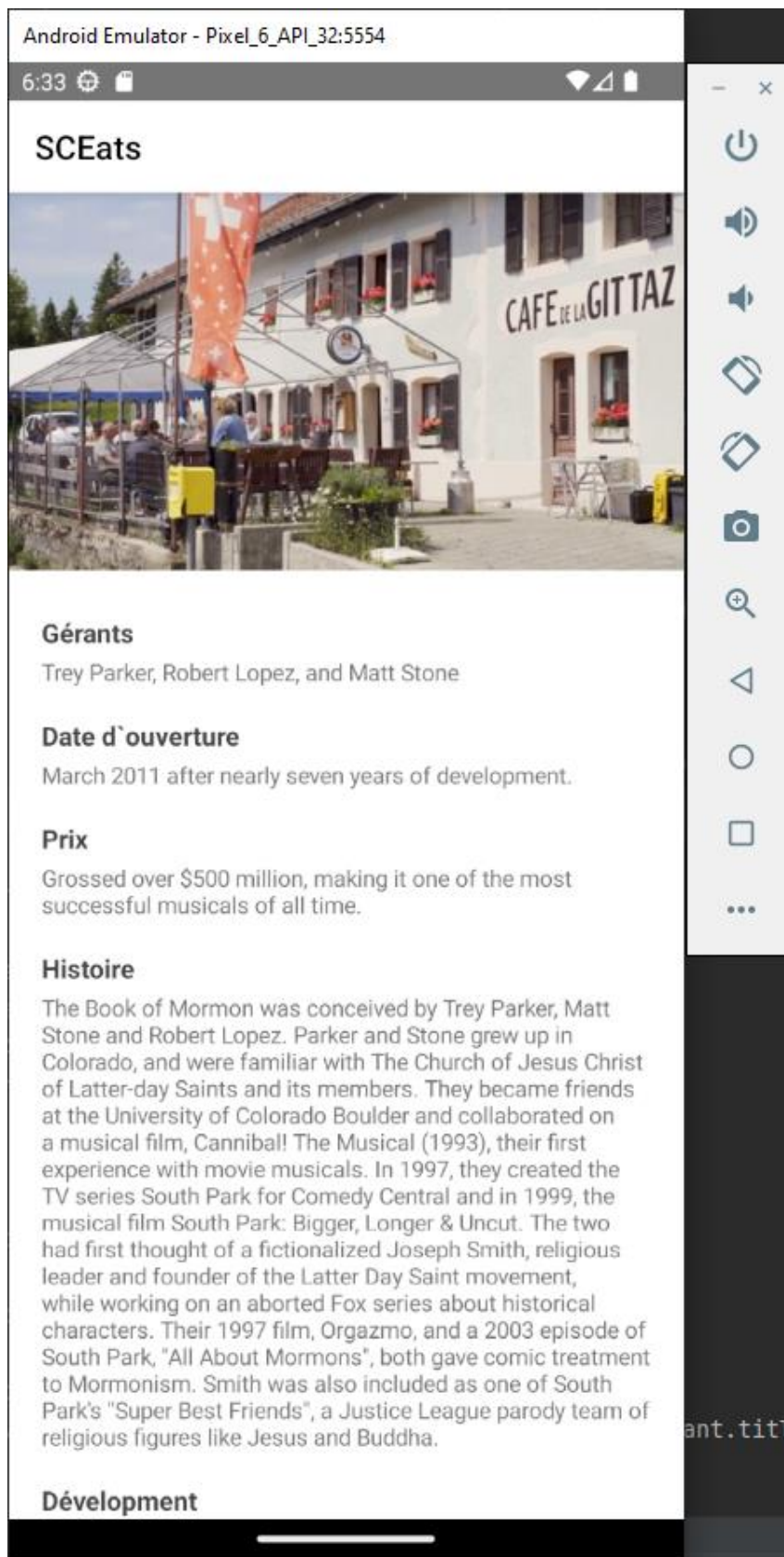
L'installation et la configuration de l'environnement de développement sont terminées et l'application devrait pouvoir tourner sur l'émulateur en exécutant la commande **ns run android** dans le dossier contenant l'application mobile.

3.3 Aperçu de l'application

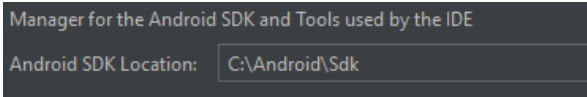

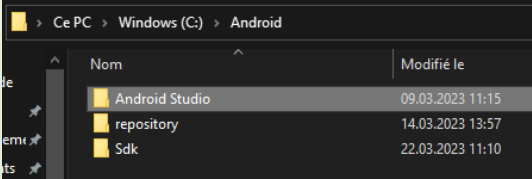

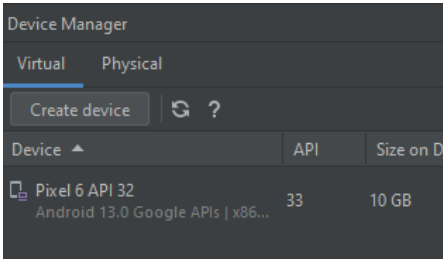

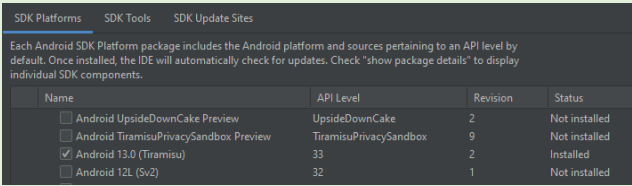

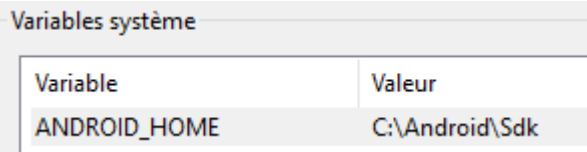

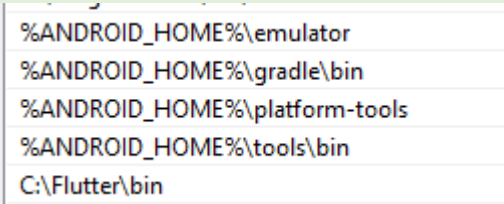






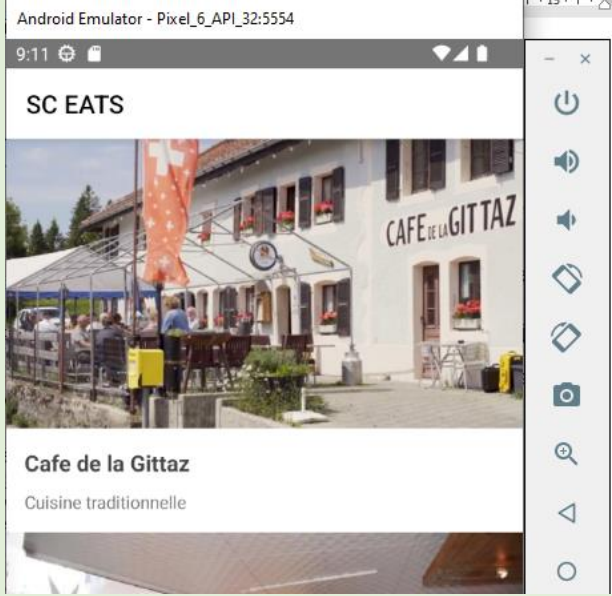



Voici la page d'accueil de l'application présentant trois restaurants de Sainte-Croix avec leur nom et le type de cuisine qu'ils servent.




En cliquant sur un des trois restaurants, il est possible d'avoir les détails sur le restaurant en question.



3.4 Description des tests effectués

Date	Test	Résultat	Validation
Tests unitaires			
28.03.2023	Android SDK est installé		
28.03.2023	Android Studio est installé		
28.03.2023	L'émulateur Android est installé		
28.03.2023	L'API de l'émulateur est installée		
28.03.2023	La variable d'environnement ANDROID_HOME est configurée		
28.03.2023	La variable d'environnement Path a été modifiée		

Tests d'intégration			
28.03.2023	La commande ns doctor android permet de vérifier si tous les composants testés précédemment et nécessaires au fonctionnement du framework sont trouvables	<pre>PS C:\Users\thierry.koetschet\Documents\GitHub\Pre-TPI_tkt\SC_Eats> ns doctor android ✓ Getting environment information No issues were detected. ✓ Your ANDROID_HOME environment variable is set and points to correct directory. ✓ Your adb from the Android SDK is correctly installed. ✓ The Android SDK is installed. ✓ A compatible Android SDK for compilation is found. ✓ Javac is installed and is configured properly. ✓ The Java Development Kit (JDK) is installed and is configured properly. ✓ Local builds for iOS can be executed only on a macOS system. To build for iOS on a d ✓ Getting NativeScript components versions information... ✓ Component nativescript has 8.5.0 version and is up to date.</pre>	
Tests fonctionnels			
28.03.2023	L'application se lance	La commande ns run android se lance sans accroche	
28.03.2023	L'application arrive à effectuer le build	Project successfully built.	
28.03.2023	L'application s'affiche sur l'émulateur de téléphone Android		
28.03.2023	L'application est responsive	L'application s'adapte à la taille de l'appareil virtuel donc elle est responsive	
28.03.2023	L'application a un menu de navigation	Par manque de temps, je n'ai pas pu implémenter cette fonctionnalité	

28.03.2023	L'application exploite des données	Le fichier restaurant.service.ts génère les objets <i>restaurants</i> et chaque objet contient des données spécifiques	
28.03.2023	L'application se connecte à une base de données	Par manque de temps et de connaissances, je n'ai pas pu implémenter cette fonctionnalité	
28.03.2023	L'application a une partie gestion de mémoire	Par manque de temps et de connaissances, je n'ai pas pu implémenter cette fonctionnalité	

3.5 Erreurs restantes

Il arrive que l'émulateur de téléphone Android ne fonctionne pas et je sais pas pour quelles raisons et je ne sais pas comment le déboguer. Parfois, le fait de redémarrer PC règle le problème mais cela n'explique rien sur la cause.

Grâce à mon analyse et ma recherche poussée sur l'installation des frameworks, j'ai pu réglé la majorité des problèmes persistants mais cela m'a pris un temps considérable au dépens de la réalisation pratique de mon projet.

4 Conclusions

Pour conclure ce travail, je pense que les objectifs que je m'étais fixés en début de projet sont loin d'avoir été atteints. Certes, j'ai réussi à choisir un framework qui correspond à mes attentes pour la réalisation d'une application mobile multiplateforme, cependant j'ai aussi réalisé que le développement mobile nécessite une formation spécifique et qu'il ne suffit pas d'apprendre les bases en autodidacte.

A propos de mon application, je suis vraiment déçu du produit fini. J'ai tellement perdu de temps en tentant d'installer les différents frameworks et je ne me rendais pas encore compte au début du semestre le temps que cela me prendrait et toutes les difficultés que j'allais rencontrer. Effectivement, je n'ai pas réalisé qu'installer un environnement de développement fonctionnel, avec un framework particulier à chaque fois et en plus de faire tourner l'application sur un émulateur pouvait être aussi laborieux. Régulièrement, lorsque j'essayais de visualiser le rendu de mon application sur l'AVD, il refusait de fonctionner et le lendemain il fonctionnait à nouveau correctement.

Je pense que de manière générale, ma frustration due à la répétition de problèmes rencontrés au cours du travail a eu un impact sur ma motivation et fait que je rends un projet dont je suis loin d'être fier et cela n'est pas facile moralement. De plus, j'ai réalisé que mes connaissances étaient loin d'être suffisantes pour ce qui me sera demandé au TPI. C'est pour ces raisons que j'ai décidé de demander mon chef de projet, M. Saison, de changer mon cahier des charges pour le TPI et de partir sur la réalisation d'un site web en PHP avec un CRUD et une architecture en MVC. Après discussion avec M. Saison, il a accepté ma demande et m'a confirmé qu'il adapterait le cahier des charges en conséquences.

Pour terminer sur une note positive, je pense que ce pré-TPI m'a tout de même permis de rendre compte de tout ce qu'il fallait ou ne pas faire en terme d'organisation, de documentation et de réalisation d'un projet de telle envergure. Cela reste une expérience qui à mon avis portera ces fruits et la différence sera notable entre mon pré-TPI et mon TPI.

5 Annexes

5.1 Cahier des charges

1 TITRE

Développement d'une application mobile multiplateformes

2 MATÉRIEL ET LOGICIEL À DISPOSITION

1 ordinateur type CPNV

Environnement de développement Windows

Outil de modélisation de base de données

Logiciels de la suite Microsoft Office pour la rédaction du rapport et la présentation

3 PRÉREQUIS

- Programmation C
- Développement Web (HTML5, CSS, PHP, JavaScript)
- Modélisation et gestion de base de données (MySQL)

4 DESCRIPTIF DU PROJET

Le but de ce projet consiste en la réalisation d'une application mobile multiplateforme : iOS, Android.

Dans un premier temps, le candidat devra analyser les différentes solutions existantes :

- Applications natives
 - iOS : Xcode (VM MacOS sur Windows), avec programmation en Objective-C vs Swift
 - Android : Android Studio ou équivalent, avec programmation en Java.
 - (Windows : Visual Studio, avec programmation en C#)
- Applications hybrides
 - Flutter (Google), avec programmation en Dart
 - React Native (Facebook) : React.js
 - Native Script : Angular.js, Vue.js
 - Autres ?
- Web views :
 - PhoneGap, Cordova, Capacitor, Ionic, ...

L'application produite est laissée libre au candidat, mais devra respecter certaines contraintes :

- Design : responsive, pages multiples et menu de navigation
- Exploitation de données, typiquement xml (web service, flux rss, ...)
- Gestion de la mémoire (*session* vs *local storage*)
- Si le temps le permet : Système de notifications, écran de chargement (*splash screen*), ...

5 LIVRABLES

Le candidat est responsable de livrer à son chef de projet et aux deux experts :

- Une planification initiale sous forme électronique au format PDF le 8 février 2023.
- Un rapport de projet sous forme électronique au format PDF 1x par semaine, le vendredi soir.
- Un journal de travail sous forme électronique au format PDF 1x par semaine, le vendredi soir.
- A la fin du Pré-TPI,
 - le rapport de projet et son journal de travail imprimés,
 - le rapport de projet final et son journal de travail sous forme électronique au format PDF,
 - un fichier « archive » contenant :
 - Un script de création de la base de données SQL ;
 - Un dossier contenant l'application complète ;
 - Une procédure d'installation et de mise en service de l'application.
- Un livrable de l'application web chaque vendredi déposé sur un dépôt distant de type Github ou Bitbucket.

6 POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

1 La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

2 En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20) :

1. Qualité de la modélisation de l'application
2. L'analyse comparative des solutions existantes
3. La pertinence des choix effectués
4. Description et qualité des tests effectués

5. La compatibilité de l'application sur les divers OS / versions
6. La transmissibilité des codes produits
7. Qualité de la documentation

7 HORAIRE DE TRAVAIL

	Lundi	Mardi	Mercredi	Jeudi	Vendredi
08:00 – 08:45					
08:50 – 09:35					
09:50 – 10:35					
10:40 – 11:25					
11:30 – 12:15					
Pause midi					
13:30 – 14:15					
14:20 – 15:05					
15:20 – 16:05					
16:10 – 16:55					

5.2 Résumé du travail

Le principal but de ce travail était d'analyser plusieurs frameworks utiles au développement d'application mobile. Cette analyse a permis de choisir un seul de ces frameworks par rapport à mes besoins personnels et aux avantages qu'il propose. Celui-ci sera utilisé pour la réalisation du projet de TPI. Le cahier des charges fourni par le chef de projet demandait également de réaliser une application mobile répondant à certaines caractéristiques.

J'ai alors débuté la réalisation du projet selon la méthode de gestion en cascade. C'est-à-dire que j'ai commencé par posé un cadre autour de mon projet afin que j'évite de trop divaguer. Le fichier de planification et le journal de travail m'ont permis de garder un historique de ce qui a été fait et de ce qui est à faire. Ensuite, j'ai pu passer à la partie analyse du travail. Cette section représente la majeure partie de mon travail car elle consistait à réaliser un rapport sur les différents framework étudiés et testés afin de faire un choix pour la réalisation du TPI. L'étape de conception peut s'apparenter au choix du framework et à la création d'une application test. L'avant-dernière étape de la méthode en cascade est l'implémentation qui correspond à l'écriture du programme et au développement de l'application. J'ai alors terminé mon projet par l'écriture et la réalisation des tests en respectant ma stratégie de tests.

Les résultats de ce projet sont les suivants : l'application réalisée est, à mon avis, loin de répondre aux attentes du cahier des charges mais je pense que ce projet consistait d'abord à un travail de recherche sur les différents framework proposés par le chef de projet afin d'acquérir les connaissances suffisantes pour la réalisation du TPI avec le framework choisi. Ce travail m'a aussi permis d'avoir aperçu sur mes lacunes en terme de réalisation de projet de manière autonome telles que ma gestion du temps qui sera cruciale lors du TPI, mon manque de discipline a probablement aussi été problématique et représente une zone d'amélioration à revoir. Mais je pense tout de même avoir acquis une certaine expérience en gestion de projet, en rédaction de documentation et en analyse préliminaire à la réalisation d'un gros projet.

5.3 Sources – Bibliographie

Source d'information pour la stratégie de tests :

<https://www.atlassian.com/fr/continuous-delivery/software-testing/types-of-software-testing>

Installation d'Apache Cordova :

<https://cordova.apache.org/docs/en/latest/guide/cli/index.html>

https://fr.wikipedia.org/wiki/Apache_Cordova

Installation de React Native :

<https://reactnative.dev/docs/environment-setup>

<https://www.youtube.com/watch?v=oZFCt69Bccc&ab>

Installation de NativeScript :

<https://docs.nativescript.org/environment-setup.html>

<https://fr.wikipedia.org/wiki/NativeScript>

Installation de Flutter :

<https://docs.flutter.dev/get-started/install>

[https://fr.wikipedia.org/wiki/Flutter_\(logiciel\)](https://fr.wikipedia.org/wiki/Flutter_(logiciel))

5.4 Journal de travail

Journal de travail					
Date	Sem	Activité	Heures	Description	Remarques
31.01.2023	5	41 Organisation personnelle	0:30	Présentation du cours par Mme Andolfatto	
31.01.2023	5	41 Organisation personnelle	1:45	Consultation des documents mis à disposition sur Moodle, tpivd.ch et Teams	J'ai essayé de regrouper tous les documents nécessaires pour le projet et à les organiser dans le même dossier
01.02.2023	5	41 Organisation personnelle	0:30	Discussion avec M. Saison à propos des détails importants à aborder pour la suite du pré-TPI et du TPI	J'ai proposé une nouvelle idée pour le TPI (application type UberEats) et M. Saison m'a proposé de réaliser le pré-TPI en faisant une analyse de différents framework pour le développement d'application mobile afin de déterminer le meilleur pour réaliser l'application du TPI
01.02.2023	5	42 Rédaction de la documentation	2:30	Recherche sur les différents frameworks proposé par M. Saison Création du repository github	Frameworks proposés : Apache Cordova, Ionic, React Native, NativeScript, Flutter, VS Fichier : recherche_framework_dev_mobile.docx
02.02.2023	5	51 Absence	3:45	Absence	
06.02.2023	6	11 Installation Cordova	1:30	Installation du framework Apache Cordova	Problème par rapport à la création d'un émulateur de téléphone Android : impossible de trouver un répertoire permettant d'enregistrer l'émulateur
07.02.2023	6	12 Création projet test Cordova	2:15	Test de création d'un projet avec Apache Cordova	Problème par rapport à la création d'un émulateur de téléphone Android : impossible de trouver un répertoire permettant d'enregistrer l'émulateur
08.02.2023	6	13 Installation React Native	1:30	Installation du framework React Native	
09.02.2023	6	14 Création projet test React Native	3:45	Tests du framework React Native	
21.02.2023	8	51 Absence	2:15	Absence	
22.02.2023	8	41 Organisation personnelle	0:45	Discussion avec M. Saison à propos de cahier des charges qu'il m'a transmis le 20.02.2023	M. Saison a répondu à mes questions concernant la remise hebdomadaire de documents et ses attentes par rapport à l'application à réaliser
22.02.2023	8	41 Organisation personnelle	1:30	Création et modification du fichier de planification et journal de travail	
22.02.2023	8	21 Conception MCD	0:45	Réalisation du modèle conceptuel de données pour la création de la base de données	
23.02.2023	8	41 Organisation personnelle	0:45	Finition sur le fichier de planification et préparation de mon planning de la semaine	
23.02.2023	8	22 Conception MLD	0:45	Réalisation du modèle logique de données pour la création de la base de données	
23.02.2023	8	23 Création de la DB	1:30	Création de la base de données avec le SGBD mariadb et enregistrement de quelques données	
23.02.2023	8	42 Rédaction de la documentation	0:45	Rédaction de l'introduction du projet	
28.02.2023	9	42 Rédaction de la documentation	0:45	Rédaction de la partie objectifs et planification	
28.02.2023	9	15 Installation Native Script	1:30	Tentative d'installation de Native Script sans grand succès pour l'instant	
01.03.2023	9	12 Création projet test Cordova	1:30	Tentative de création de projet et de construire l'application	Problème au niveau du "build" de l'application
01.03.2023	9	42 Rédaction de la documentation	1:30	Rédaction de la partie organisation	
02.03.2023	9	42 Rédaction de la documentation	0:45	Rédaction de la partie dossier de réalisation	
02.03.2023	9	15 Installation Native Script	0:45	Installation de Native Script	
02.03.2023	9	16 Création projet test Native Script	1:30	Test de création d'un projet avec Native Script	
02.03.2023	9	41 Organisation personnelle	0:45	Planification de la semaine 10	
07.03.2023	10	51 Absence	1:00	Participation à la présentation de l'Ecole Supérieure par le CPNV	
07.03.2023	10	42 Rédaction de la documentation	1:15	Ajout du cahier des charges à la documentation et modification de la mise en page	
08.03.2023	10	51 Absence	3:00	Absence pour cause de maladie	
09.03.2023	10	17 Installation Flutter	2:00	Installation du framework flutter et réinstallation de zéro de Android Studio pour cause de quelques problèmes	
09.03.2023	10	18 Création projet test Flutter	1:30	Tentative de création de projet avec Flutter	L'application fonctionne bien mais impossible de la faire tourner sur l'émulateur Android
09.03.2023	10	41 Organisation personnelle	0:15	Mise à jour du journal de travail et de la planification	
14.03.2023	11	16 Création projet test Native Script	1:00	Approfondissement du projet test avec Native Script	
14.03.2023	11	19 Comparaison et choix du meilleur framework	0:30	Choix du framework Native Script pour la réalisation de mon application	
14.03.2023	11	31 Structure des pages du projet	0:45	Création du projet et mise en place de la structure des pages de l'application	
15.03.2023	11	33 Page d'accueil	3:00	Création de la page d'accueil en suivant le tutoriel du site officiel de Native Script (https://docs.nativescript.org/)	Quelques erreurs sont apparues au niveau de l'accessibilité des différentes pages mais l'aide de M. Hurni m'a permis de les résoudre
16.03.2023	11	32 Connexion à la base de données	2:30	Commencement de la connexion à la base de données	
16.03.2023	11	42 Rédaction de la documentation	0:45	Rédaction de la partie structure du dossier dans ma doc	
16.03.2023	11	41 Organisation personnelle	0:30	Remplissage de mon journal de travail pour la semaine	
21.03.2023	12	33 Page d'accueil	1:30	Tentative de modification de la page d'accueil de l'application	
21.03.2023	12	32 Connexion à la base de données	0:45	Tentative de création d'un fichier de connexion à la base de données	Mon émulateur a décidé de cesser de fonctionner
22.03.2023	12	42 Rédaction de la documentation	2:15	Rédaction de la partie description des frameworks	
22.03.2023	12	41 Organisation personnelle	0:45	Discussion avec M. Saison pour définir certains détails concernant la réalisation du TPI	
23.03.2023	12	42 Rédaction de la documentation	2:15	Rédaction de la partie analyse	
23.03.2023	12	42 Rédaction de la documentation	1:30	Rédaction de la partie choix du framework	
28.03.2023	13	42 Rédaction de la documentation	2:15	Rédaction de la documentation	
29.03.2023	13	42 Rédaction de la documentation	2:15	Rédaction de la documentation	
	0				

Le journal de travail est également disponible dans le fichier « planification_pre-TPI_tkt.xlsm » présent dans le répertoire git.

5.5 Glossaire

Framework	Ensemble de composants logiciels servant à la programmation d'application.
Use case	Description écrite de la réalisation de tâches utilisé dans le développement logiciel.
IDE	Environnement de développement
HTML	Hypertext Markup Language est un langage de programmation très utilisé dans le développement web.
CSS	Cascading Style Sheets est langage de programmation utile à la mise en page d'un site web.
OS	Système d'exploitation
AVD	Android Virtual Device permet d'émuler un téléphone portable Android.
SDK	Software Development Kit
API	Application Programming Interface est un ensemble de composants logiciels facilitant le développement d'application.
CRUD	Acronyme pour « Create Read Update Delete » représentant la relation entre la base de données et l'application.
MVC	Model View Controller est une manière d'organiser la structure d'un projet de développement.

5.6 Manuel d'installation

Pour l'installation du framework, veuillez suivre le point 3.3 de cette documentation.