



Projet 62 MyCampus:
Développement d'un chatbot à destination du campus
brestois de IMT Atlantique.

Rapport final du projet CODEV
Version 2

Écrit par:
DOUBLI Youssef, EL QORCHI Imane,
FAIZ Farouk, MASSAOUD Hamid
Élèves ingénieurs à IMT Atlantique, Campus de Brest

A l'attention de:

M. MALLET Julien, Encadrant du projet et Enseignant chercheur
dans le département informatique à IMT Atlantique,
et le comité de pilotage du projet CODEV

16 Juin 2019

Résumé

— *Écrit par : MASSAOUD Hamid*
— *Relecture par : FAIZ Farouk*

Ce rapport porte sur le développement d'une plateforme intitulée «MyCampus», qui permettra la bonne gestion des ressources de la Maisel (Maison des élèves) à savoir les cuisines, la laverie et le service de réception des colis. Ce projet est le fruit d'une réflexion personnelle portée par les membres de notre équipe, l'idée est venue suite à un réel besoin que nous avons mesuré grâce à un sondage. Ce rapport met en relief les résultats d'un travail qui a été réparti en deux mini-projets : un mini-projet pour le développement d'un chatbot et un autre pour l'interface web. Un chatbot est par définition, un robot logiciel pouvant dialoguer en langage naturel avec un individu. Notre choix de développer un tel robot a été déterminé par l'essor fulgurant que connaît ce secteur et ses nombreuses avancées technologiques notamment dans le domaine de l'intelligence artificielle, de l'analyse du langage et du Machine Learning.

L'idée de MyCampus vient donc s'inscrire dans un contexte de transformation numérique, elle vise à faciliter non seulement la vie des étudiants mais aussi le travail de responsable des colis à la Maisel. Pour l'étudiant, elle lui permettra d'être notifié de la disponibilité de ses colis, de connaître les offres de covoiturage proposées par les élèves brestois, de consulter le menu du RAK, ainsi que celui de la cafétéria et les emplois du temps. Pour le responsable, elle lui permettra de substituer le cahier des colis par une interface web afin de renseigner plus facilement les colis arrivés.

Table des matières

Introduction	3
1 Objectifs et gestion du projet	3
1.1 Motivation	3
1.2 Objectifs du projet	4
1.3 Gestion du projet	5
2 Développement logiciel du chatbot	5
2.1 Conception du chatbot	6
2.2 Solutions techniques	6
2.3 Réalisation du prototype zéro	10
2.4 Services implémentés dans le chatbot	10
3 Développement logiciel de l'application web	14
3.1 Solutions techniques	14
3.2 Fonctionnalités	16
4 Composante matérielle : solutions capteurs	19
4.1 Problématique	19
4.2 Première solution : capteur de courant	19
4.2.1 Prototypage	19
4.2.2 Devis de l'installation pour la laverie	20
4.3 Deuxième solution : capteur de courant	20
4.3.1 Prototypage	20
4.3.2 Devis de l'installation pour la laverie	20
4.4 Contraintes techniques et logistiques des installations	21
5 Tests et validation du projet	21
5.1 Contraintes	21
5.2 Bilan du projet	22
Conclusion	22
Glossaire	23
Annexes	25

Introduction

- *Écrit par : MASSAOUD Hamid*
- *Relecture par : DOUBLI Youssef*

Sur le campus de Brest de IMT Atlantique, les étudiants ont accès à plusieurs ressources mises à disposition par la Maisel (Maison des élèves) : les cuisines, la laverie et un service de réception des colis. Ce système mis en place pour 428 étudiants a besoin sans doute d'une meilleure gestion. En effet, plusieurs résidents sont contraints de faire des allers-retours pour aller vérifier la disponibilité des machines à laver, des colis au foyer, ou des cuisines est libre ou non. MyCampus est une plateforme composée d'un chatbot sous Facebook Messenger et une interface Web que nous développons pour optimiser cette gestion, et améliorer le quotidien des étudiants, il permettra aux utilisateurs francophones et anglophones du campus d'être notifiés de l'arrivée de leur colis, de vérifier la disponibilité des machines à laver et la disposition des cuisines. Il offre également d'autres services complémentaires, les élèves peuvent connaître les offres de covoiturages et consulter le menu du RAK, de la cafétéria et les emplois du temps. Nous sommes convaincus que l'utilisateur parviendra plus facilement et plus rapidement à trouver l'information en discutant avec un chatbot. Ces dernières années, l'utilisation des messageries instantanées a explosé au point de dépasser la fréquentation des réseaux sociaux. Messenger par exemple attire aujourd'hui plus de 1,4 milliard d'utilisateurs dans le monde, et demeure le plus utilisé parmi les résidents du campus, ce qui justifie le choix d'un chatbot implémenté sur Facebook. Désormais les chatbots contiennent une intelligence artificielle qui permet d'assimiler les besoins de l'interlocuteur et fournit l'information dont il a besoin. Nous essayerons dans un premier temps d'explicitier les objectifs de notre projet. Ensuite, nous détaillerons le développement technique. Enfin, nous présenterons les tests effectués pour la validation du projet.

1 Objectifs et gestion du projet

- *Écrit par : MASSAOUD Hamid*
- *Relecture par : EL QORCHI Imane*

1.1 Motivation

Les chatbots représentent sans doute la plus grosse révolution digitale pour les marques depuis l'avènement des réseaux sociaux. Ils vont remplacer 80% des applications d'ici 5 ans c'est ce qui affirme Phil Libin le cofondateur d'Evernote, un logiciel qui permet d'enregistrer des informations, sous forme de notes, images, vidéos, ou pages web.

Les chatbots sont déjà utilisés et employés par plusieurs entreprises. On peut citer par exemple celui de 20minutes qui permet d'envoyer un carrousel d'actualités ou celui de Domino's Pizza qui lui, propose de régler une commande et enregistrer des coordonnées de

l'utilisateur. L'entreprise ferroviaire SNCF elle aussi a lancé un chatbot en 2016 permettant de rechercher un train et de réserver un trajet. Notre motivation pour le développement de notre propre chatbot est donc venue suite à nos recherches sur le domaine en question et à notre volonté d'enrichir nos connaissances dans le domaine de l'intelligence artificielle. L'implémentation d'un chatbot nous offre un champ d'étude et de découverte plus large. Les chatbots peuvent offrir plusieurs avantages notamment en terme de temps, puisqu'ils sont plus rapides à fournir les informations, mais aussi par leur simplicité d'usage.

1.2 Objectifs du projet

Les 428 résidents du campus qui sont les potentiels utilisateurs ont accès à plusieurs ressources et services mis en place par la Maisel :

- Parmi les 12 bâtiments, 9 possèdent chacun une cuisine commune.
- 12 machines et 8 sèche-linge fonctionnels dans la laverie (bâtiment i4).
- Un service de réception des colis destiné aux résidents.

En revanche du fait de la grande superficie du campus, et du nombre limité des ressources, l'accès est généralement difficile et contraignant. Pour mesurer le besoin réel des résidents, nous avons choisi de lancer une enquête auprès de ces derniers afin de recueillir leur avis sur l'utilité de notre service (figure ci-dessous). Sur les 100 réponses collectées des élèves qui possèdent un logement sur le campus nous avons conclu que :

- 91.5% des résidents ont été déjà contraints à faire le trajet chambre-cuisine ainsi que le trajet chambre-laverie plusieurs fois.
- 92% des habitants estiment utile de savoir à l'avance la disponibilité de la cuisine et les machines à laver.
- 97% des résidents pensent que être notifiés de l'arrivée des colis est nécessaire à mettre en place, d'où le fait que ce soit notre fonction prioritaire.

En effet, les élèves ne connaissent pas exactement le jour de la réception de leurs biens, ils sont obligés de faire des allers-retours pour vérifier la présence ou non de leurs colis. La gestion des colis se passe aujourd'hui comme ceci : lorsqu'un colis arrive à la maison des élèves, le responsable des colis note sur un cahier les coordonnées du destinataire. Ensuite, vers 18h ce cahier et les colis sont transmis au foyer, là où les élèves peuvent les récupérer. Notre projet a donc pour objectif d'apporter des solutions techniques pour améliorer ce processus et ainsi promouvoir le bien-être des résidents du campus. Pour cette raison notre projet cible deux acteurs différents : le résident du campus et l'employé de la Maisel. L'utilisation de la plateforme diffère d'un acteur à l'autre, pour un habitant de la Maisel elle se fait uniquement via le chatbot qui lui présente les différentes informations sous forme de langage naturel. En revanche, pour un employé de la Maisel, l'utilisation se fait via l'interface web qui lui permet de renseigner les colis arrivés.

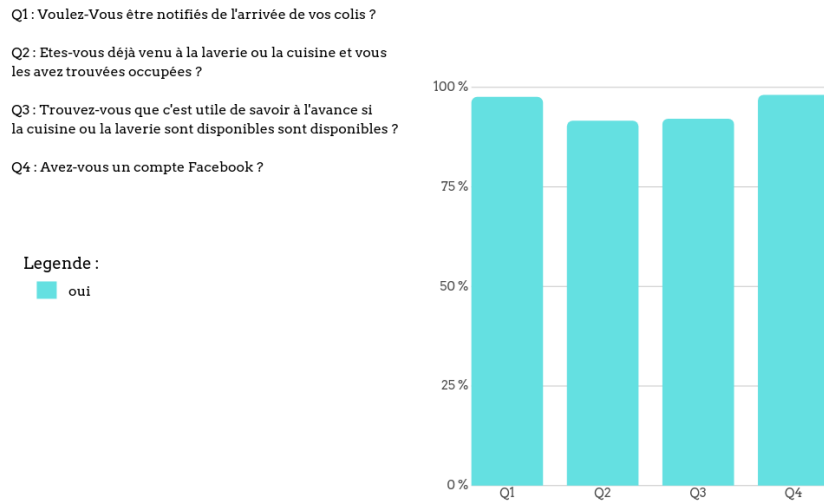


FIGURE 1 – Mesure du besoin des résidents de la Maisel

1.3 Gestion du projet

La collaboration entre les membres de l'équipe lors du développement se fait par le biais de l'outil Github. En effet, cet outil assure une synchronisation entre les différents collaborateurs et permet la gestion des différentes versions du code. Pour atteindre nos objectifs, nous avons également rédigé un planning pour la distribution des tâches et le bon suivi du déroulement du projet.

2 Développement logiciel du chatbot

— *Écrit par : DOUBLI Youssef*

— *Relecture par : MASSAOUD Hamid*

Un chatbot est un agent qui dialogue avec l'utilisateur. C'est une interface homme-machine qui permet à l'utilisateur d'envoyer des commandes et des requêtes à travers le langage naturel. Un chatbot est en général un moyen très ergonomique et facile d'utilisation. Nous avons décidé d'utiliser Facebook Messenger comme plateforme pour notre chatbot vu sa popularité et sa compatibilité avec plusieurs machines (téléphones, ordinateurs, tablettes...) et systèmes d'exploitation (Android, iOS...).

2.1 Conception du chatbot

L'utilisateur est l'unique acteur face au chatbot. Il peut poser des questions en langage naturel à travers son téléphone, ordinateur, ou toute plateforme supportée par l'application Facebook. À travers l'API Facebook Messenger, ce message est transmis pour être traité par la partie logique du chatbot. Il interagit simultanément avec la base de données contenant les informations sur les colis, et avec des services extérieurs comme le site web RAK et EDT pour récupérer le menu du restaurant et les emplois du temps des élèves. Pour optimiser la communication entre le chatbot et les utilisateurs, l'API DialogFlow est intercalé entre Facebook Messenger et la logique du chatbot : les messages sont transmis à DialogFlow pour les classer et comprendre l'intention de l'utilisateur. Le schéma ci-dessous présente le principe de fonctionnement de MyCampus.

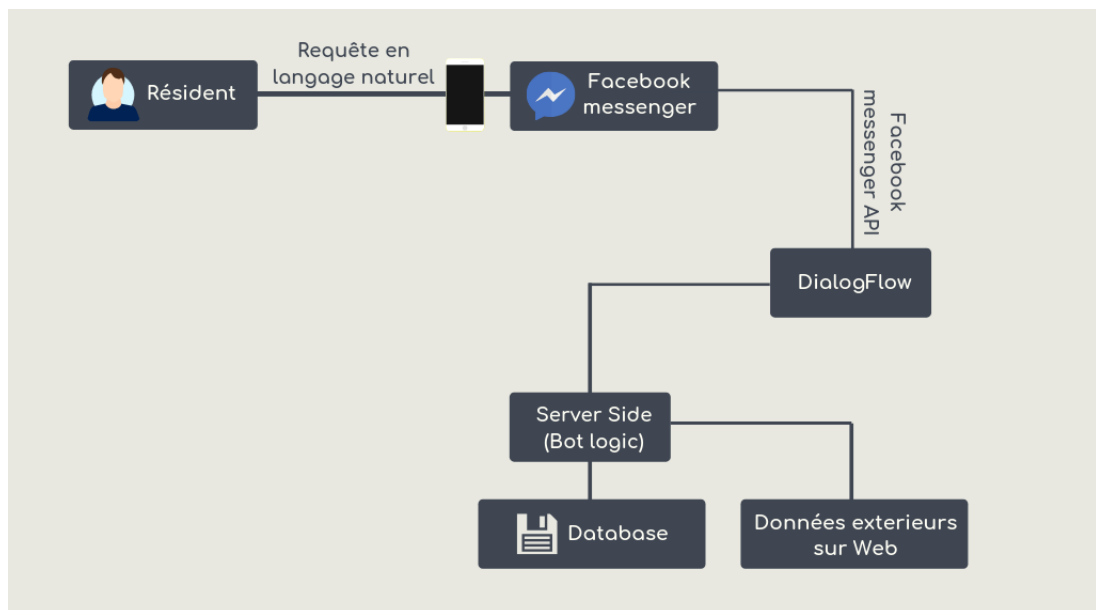


FIGURE 2 – Schéma de fonctionnement de MyCampus

2.2 Solutions techniques

NodeJS & frameworks : NodeJs s'est avéré le meilleur outil pour coder la logique du chatbot. Sa flexibilité, la diversité de ses librairies (packages) et la riche documentation disponible nous ont encouragé à le choisir comme outil principal. L'utilisation de ce langage nous a mis face à un vrai défis vu sa nature dite "event-driven".

L'introduction de cette notion et sa compréhension nous a facilité la tâche du développement et nous a permis de mieux architecturer la logique du chatbot comme montré sur la figure

ci-dessous. Nous avons eu recours à plusieurs librairies :

- Express : est une librairie qui permet le traitement des requêtes HTTP (GET, POST, PUT, DELETE ..) et de faire le routage sur une application NodeJs.
- Body-parser : historiquement une partie de Express, body-parser actuellement est une librairie indépendante qui permet le "parsing" des données passées dans une requête POST et expose ces données sur une application NodeJs.
- Cheerio : une librairie de web-scraping, qui récupère le contenu d'une page, fait le 'parsing', et permet le traitement de ces données.
- Dotenv : une librairie qui gère les variables de l'environnement sur un serveur d'exécution et permet d'importer des fichier .env. Bien que nous ne l'utilisions pas sur l'environnement de production vu que Heroku prend en charge la gestion des variables, "dotenv" a facilité le développement local.
- Ical : est une librairie qui manipule les fichier .ical (apple calendar) et permet la gestion de leurs contenus.
- Nodemailer : permet d'envoyer des emails par un serveur SMTP depuis NodeJs.
- Pg : permet de communiquer et interroger une base de données Postgres.
- Request : permet d'envoyer des requêtes HTTP et de récupérer leurs réponses.
- Uuid : génère des uuid (ie. universally unique identifier, appelé également GUID).
Utilisé pour générer le lien de vérification.

Le schéma ci-dessous est un bilan de ces solutions techniques :

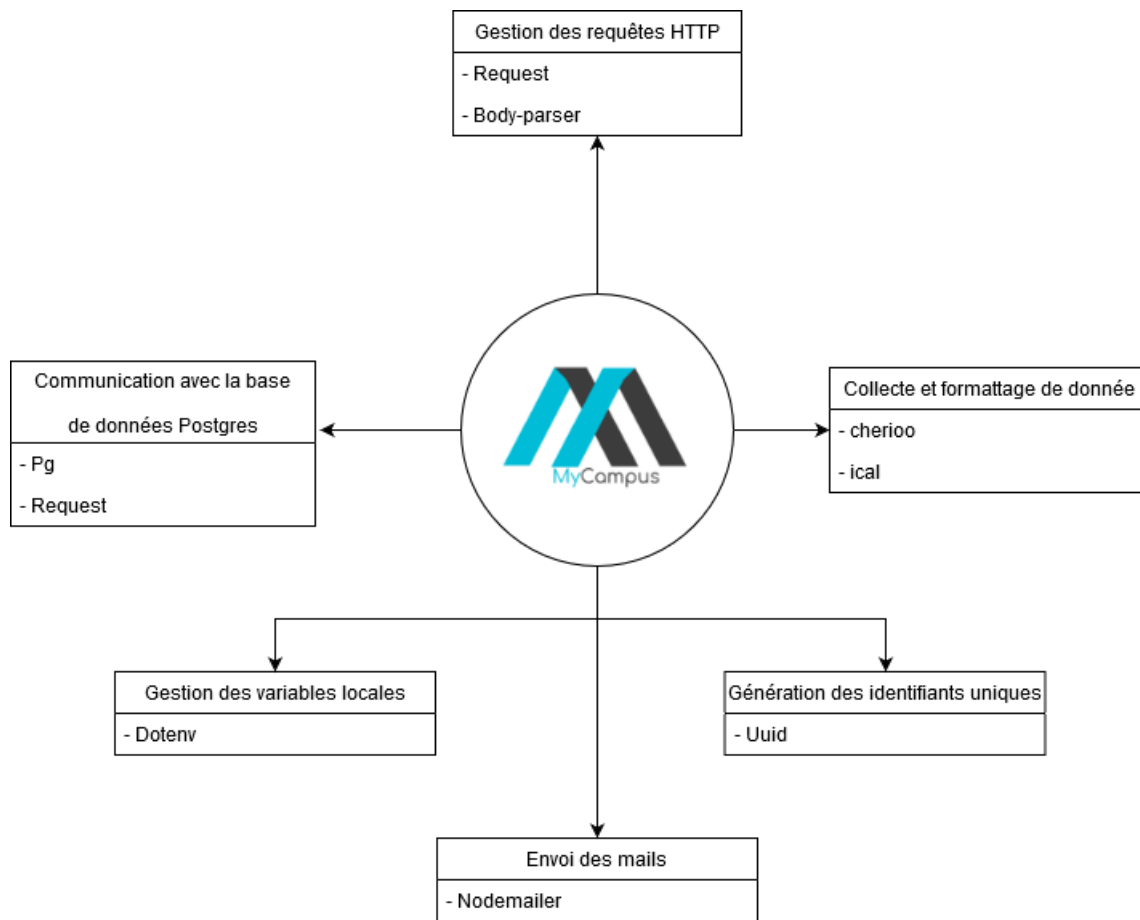


FIGURE 3 – Schéma illustrant les rôles des dépendances utilisées

Heroku : Heroku est un service d'hébergement qui permet le déploiement des applications web sur des serveurs en plusieurs langages de programmation (NodeJs, Python, PHP ...). Nous avons décidé d'y héberger la logique du chatbot qui nécessite un serveur d'exécution disponible de façon permanente pour traiter les requêtes des utilisateurs.

La décision d'utiliser Heroku s'est basée principalement sur la qualité de l'offre gratuite qu'il propose. Bien qu'il pose des limitations sur la machine (trafic, puissance et mémoire), il convient à notre projet. Heroku offre une interface graphique et une interface en ligne de commande (Heroku CLI) qui permettent la gestion de la machine et sa configuration.

Postgres SGBD : Notre premier réflexe a été d'utiliser une base de données MySQL vu notre familiarité avec ce SGBD. Au cours du développement du prototype zéro, on

a basculé vers l'utilisation de Postgres car notre hébergeur Heroku propose des bases de données Postgres gratuitement. Cela était parfaitement convenable et très adapté à ce qu'on cherchait : rassembler les données sur le même service permettra de mieux les gérer, les sécuriser, et surtout éviter des coûts supplémentaires.

L'accès à la base de données se fait par Heroku CLI, qui donne accès à l'interface en ligne de commande de Postgres classique. La conception de la base de données a été faite suivant le schéma conceptuel suivant :

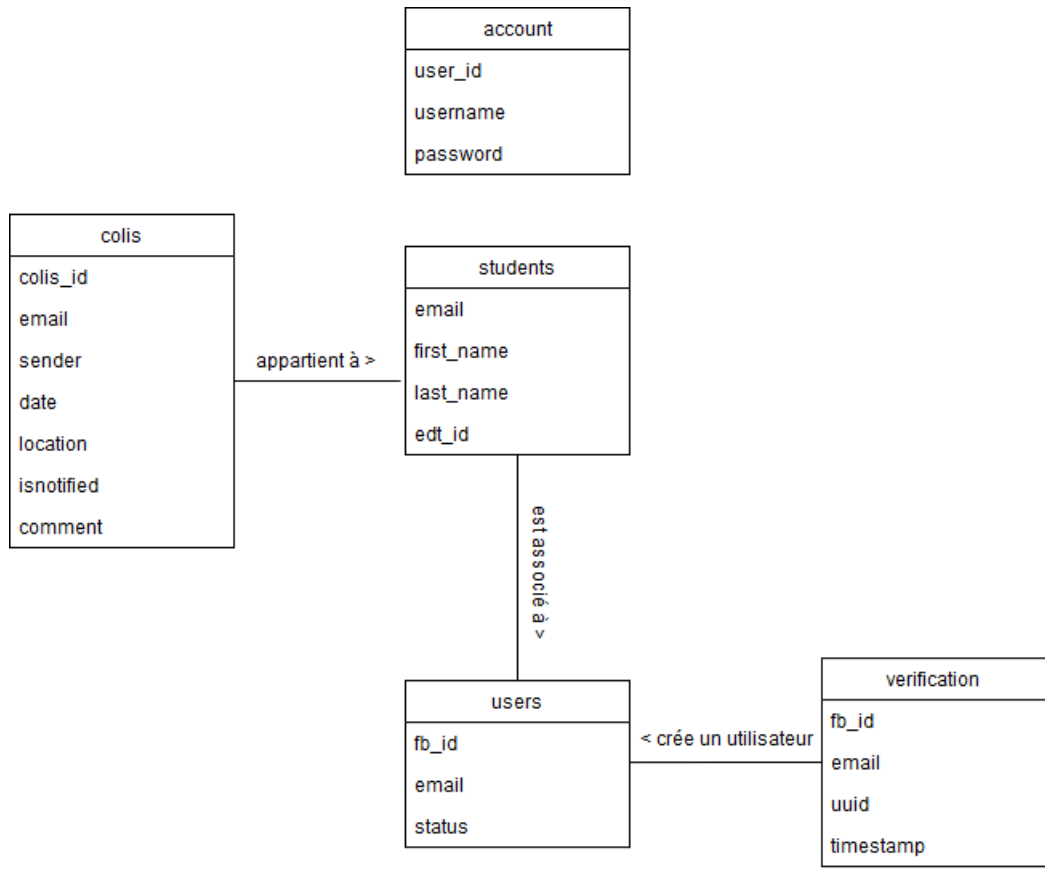


FIGURE 4 – Schéma conceptuel de la base de données de MyCampus

Facebook Messenger API : Facebook Messenger API est une API de Facebook qui fait partie des APIs de “Facebook For Developers”. Facebook a mis en place cette dernière pour encourager les développeurs à proposer des applications sur Facebook, notamment des chatbots fonctionnant sur Messenger avec lesquels les utilisateurs peuvent interagir directement. Facebook Messenger exige une page publique qui représente la façade publique du

chatbot, et une application sur "Facebook For Developers" qui permet de lier les différents composants et interagir avec la partie logique du chatbot.

DialogFlow : DialogFlow est une API de NLP (Natural Language Processing) qui permet le traitement du langage naturel. C'est un outil Google, à la base payant, mais qu'on utilise gratuitement avec certaines restrictions. DialogFlow permet à l'utilisateur d'avoir une discussion plus fluide et dynamique avec le chatbot. Elle introduit une notion de (Intent) où le développeur définit des intentions et entraîne l'agent à les reconnaître lors d'une discussion avec un utilisateur. Une fois reconnue, on peut envoyer ce résultat à la logique du chatbot pour effectuer le traitement classique et répondre à l'utilisateur. On peut configurer certaines intentions pour avoir des réponses "statique" et prêtes qui ne nécessitent pas d'être traitées par la logique du chatbot.

2.3 Réalisation du prototype zéro

Étant donné que nous avons nous-mêmes initiés ce projet, nous avons besoin d'un prototype sur lequel on pouvait s'appuyer pour convaincre l'équipe de pilotage de CODEV d'accepter notre proposition. Ce prototype était également une opportunité pour se familiariser avec les différents outils, et pour lire la documentation disponible sur "Facebook For Developers" et "Facebook Messenger API".

Bien que le code a été repris à zéro après la validation initiale de notre projet, le prototype zéro était un bon point de départ pour l'avancement du développement.

2.4 Services implémentés dans le chatbot

Vérification d'identité : Le service de vérification était le premier service à implémenter dans le chatbot. Afin de vérifier qu'un compte Facebook appartient à un élève de IMT Atlantique, nous avons décidé d'utiliser l'adresse @imt-atlantique.net. Par conséquent, dès que l'utilisateur initie une conversation avec le chatbot, celui-ci vérifie si son compte est déjà authentifié. Sinon, il demande une adresse @imt-atlantique vers laquelle il envoie un email de vérification contenant un lien d'activation du compte. Cette opération n'a lieu que pendant la première utilisation. Une fois authentifié, l'utilisateur peut accéder aux autres services librement.

L'envoi du mail de vérification s'effectue en deux étapes : La génération du lien de vérification, et l'envoi du mail. Une fois le mail de l'utilisateur renseigné, le chatbot crée un compte temporaire et génère un UUID qui apparaît sur le lien de vérification. Grâce au package "nodemailer", un mail est envoyé par le serveur SMTP de Gmail par l'adresse mycampusimt@gmail.com. Dès lors que l'utilisateur clique sur le lien, le serveur compare le UUID y figurant avec celui généré précédemment et active le compte. Par la suite, le compte temporaire est supprimé et l'utilisateur est désormais actif.

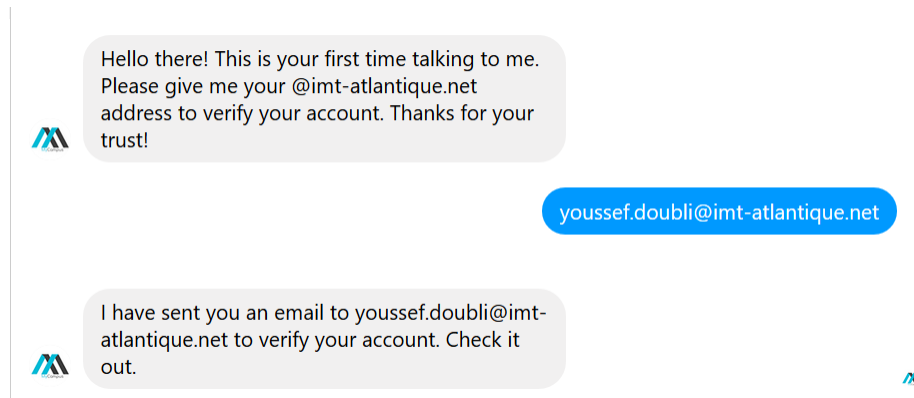


FIGURE 5 – Processus de vérification d'identité

Notification colis : Comme évoqué précédemment, recevoir des notifications pour les colis était une priorité. Cependant, la partie chatbot pour implémenter ce service n'était pas plus compliquée techniquement. La grande partie du travail était dédiée à l'application web, cela justifie le retard arrivée dans l'implémentation de ces services. Une fois qu'un colis est saisi, l'utilisateur recevra une notification l'informant de l'arrivée de son colis. Il peut également demander au chatbot à tout moment s'il a des colis non récupérés. La notification est envoyée sur Facebook, mais il peut également la recevoir par mail. Ceux qui n'ont pas activé leurs comptes peuvent toutefois bénéficier de cette alternative. Les utilisateurs actifs peuvent également demander de recevoir leurs notifications par mail et non pas via Facebook Messenger s'ils le souhaitent.

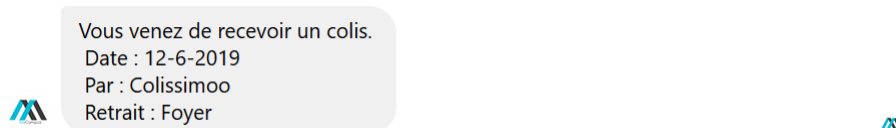


FIGURE 6 – Exemple de notification envoyée à l'utilisateur

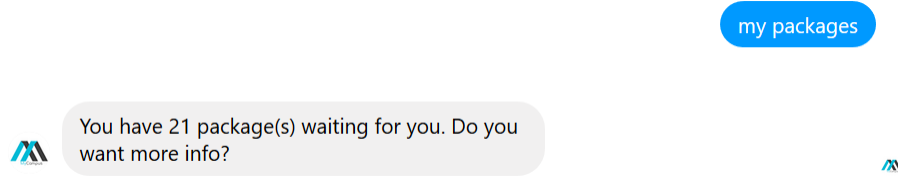


FIGURE 7 – L'utilisateur demande au chatbot des informations sur ses colis

Menu RAK : Le service menu RAK permet à l'utilisateur de demander des informations sur le menu du jour, et de recevoir la réponse instantanément sans besoin de consulter le site du RAK. Notre approche était d'utiliser le web scraping pour récupérer les informations, les organiser, les reformuler, et les présenter dans un format compatible avec une application de messagerie. La librairie cherioo a rendu cette tâche possible, et nous a permis d'implémenter facilement ce service.

L'utilisateur demande alors "dinner" ou "what's for dinner" (respectivement "lunch" ou "what's for lunch") et le chatbot répond. Après des alpha tests, des utilisateurs ont proposé d'ajouter les menus du cafétéria à ce service, ce qui est actuellement accessible via une commande comme "cafeteria" ou "menu cafétéria".

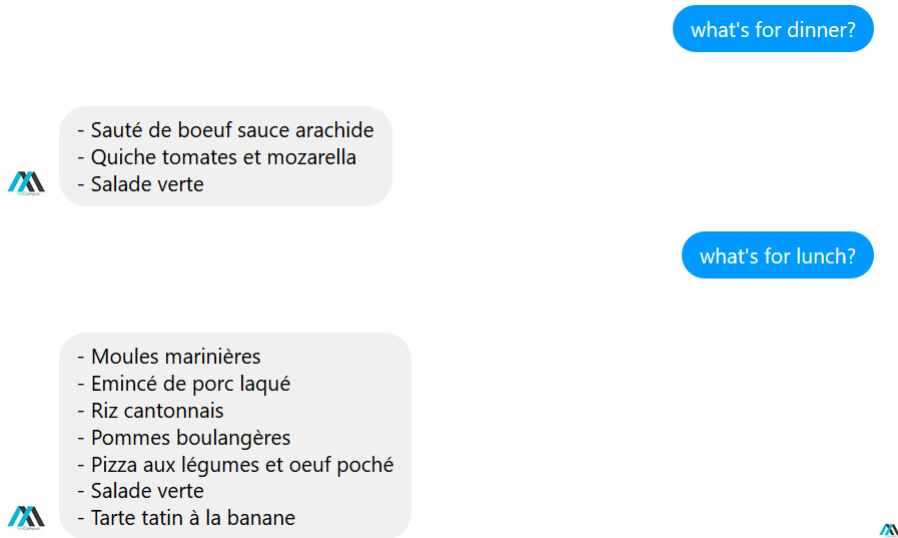


FIGURE 8 – L'utilisateur demande les menus du RAK

Accès à l’emploi du temps : Donner au chatbot l’accès au emplois du temps a été la partie la plus difficile à réaliser d’un point de vue technique. Edt.int-atlantique.net qui rassemble tous les emplois du temps des étudiants repose sur la plateforme ADEsoft. Après beaucoup de recherches, nous ne sommes pas parvenus à trouver une API qui nous permettra de communiquer avec ce service. Il était indispensable donc de faire manuellement ce travail et d’opter pour une solution alternative.

Notre approche était d’utiliser les liens d’exportation disponibles sur la plateforme qui exporte le calendrier des élèves en un fichier .ical (format apple calendar) mais qui contient un identifiant unique à chaque étudiant. Alors il fallait récupérer ces identifiants grâce à un programme Python. Le choix de python pour la réalisation de cette tâche a été basé sur des raisons d’optimisation. Ce script est la seule composante du projet écrite en Python, et qui synchronise les identifiants des élèves dans la base de données.

L’utilisateur peut demander s’il a des cours (e.g : “my classes” ou “my classes next friday” ou “my classes 05/06/2019”) et recevra une réponse instantanément.

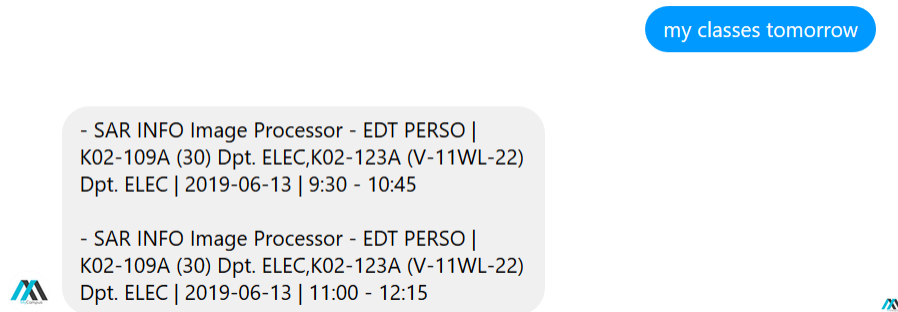


FIGURE 9 – L’utilisateur demande au chatbot son emploi de temps

Covoiturage : En collaboration avec le projet DDRS “Atlantic-Express”, nous avons intégré le service covoiturage qui permet aux utilisateurs de demander au chatbot s’il existe des covoitages disponibles vers une destination précise sur la plateforme Atlantic-Express. Nous avons suivi deux approches différentes : La première approche reposait sur le web-scraping pour collecter les données directement du site web de atlantic-express. La deuxième approche propose une solution optimale : avoir un accès directe à la base de données MySQL du service hébergée par le ResEl. L’utilisateur peut demander par exemple au chatbot “covoiturage”, “carpool” ou “covoiturage brest lyon” et le chatbot répond suivant les trajets disponibles.

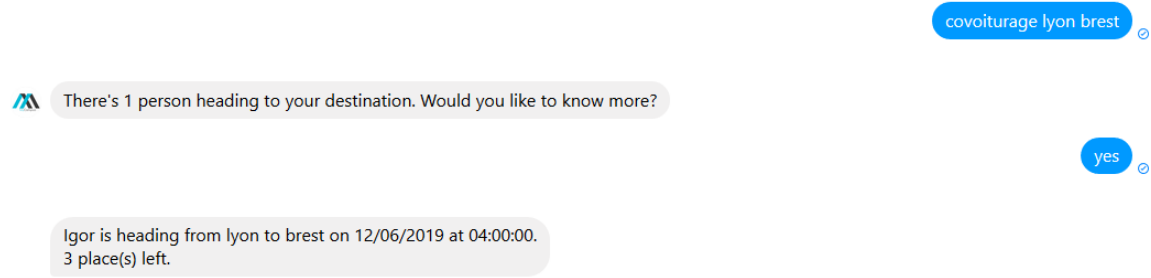


FIGURE 10 – L'utilisateur vérifie la disponibilité d'un covoiturage

Enfin, le développement du chabot terminé, nous avons besoins pour le service de notification de colis d'une interface pour les employés de la Maisel afin de faciliter la saisie des informations concernant les colis.

Dans la partie qui suit, nous allons examiner les solutions techniques ainsi que les fonctionnalités du site web.

3 Développement logiciel de l'application web

- *Écrit par : FAIZ Farouk*
- *Relecture par : DOUBLI Youssef*

L'application Web est un site destiné au personnel de la Maisel. À l'aide de leurs identifiants (nom d'utilisateur et mot de passe), ils peuvent interagir avec la base de données en ajoutant, modifiant ou supprimant des colis, ou tout simplement en parcourant la liste des colis arrivés avec la possibilité d'imprimer un PDF de cette liste.

3.1 Solutions techniques

Modèles HTML : Les modèles (Templates) permettent de générer des fichiers HTML suivant une structure bien précise. Ici, un modèle en EJS permet de remplir le tableau en parcourant la liste des colis et en remplissant chaque ligne par leurs informations respectives (nom, prénom, date ...etc.).

CSS : Bootstrap a été le meilleur choix pour un framework CSS. Couplé avec du CSS personnalisé, il a permis de créer un site web adaptatif avec un bon résultat visuel en peu de temps.

Front Js : Pour la partie Front, on a utilisé JQuery comme librairie de base. L'appel au code JS de Bootstrap a aussi été nécessaire, notamment pour remplir les "Modals".

Deux autres librairies secondaires ont été importées dans le code. La première a permis d'intégrer la fonctionnalité d'autocomplétion "Typeahead" pour les emails avec un résultat visuel convaincant pour les inputs de Bootstrap. La deuxième est une librairie permettant d'intégrer des masques de saisie "Input mask", elle a été utilisée pour avoir le champ de saisie de la date.

Back End : Pour le Back, NodeJS s'est avéré être un bon choix vu que c'était le langage d'écriture du chatbot. Express, framework très populaire pour écrire du NodeJS en Back end, était donc une nécessité. De nombreuses librairies/packages ont été utilisés pour la partie Back :

- Body-parser : Permet de récupérer les données provenant d'une requête POST comme objet JSON.
- Request : Utilisée pour envoyer une requête POST au serveur Heroku hébergeant le chatbot pour le mettre en marche et pouvoir envoyer des notifications aux utilisateurs si besoin.
- Pg : Sert d'intermédiaire pour pouvoir interagir avec la base de données
- Dotenv : Pour pouvoir stocker des variables d'environnement qui peuvent changer selon si l'on est en développement (localement) ou en production (Heroku).
- Passport et passport-local : Ces deux librairies gèrent la plus grosse partie de l'authentification. Le "local" est une "stratégie" qui consiste à utiliser un nom d'utilisateur et un mot de passe stockés dans une base de données.
- Express-session : Permet de gérer la notion de session d'un utilisateur à travers les cookies.
- Bcrypt : Permet de hacher les mots de passe lors de la création des comptes et de comparer les hashes lors de l'authentification

Le schéma ci-dessous représente les relation entre l'application et ces librairies :

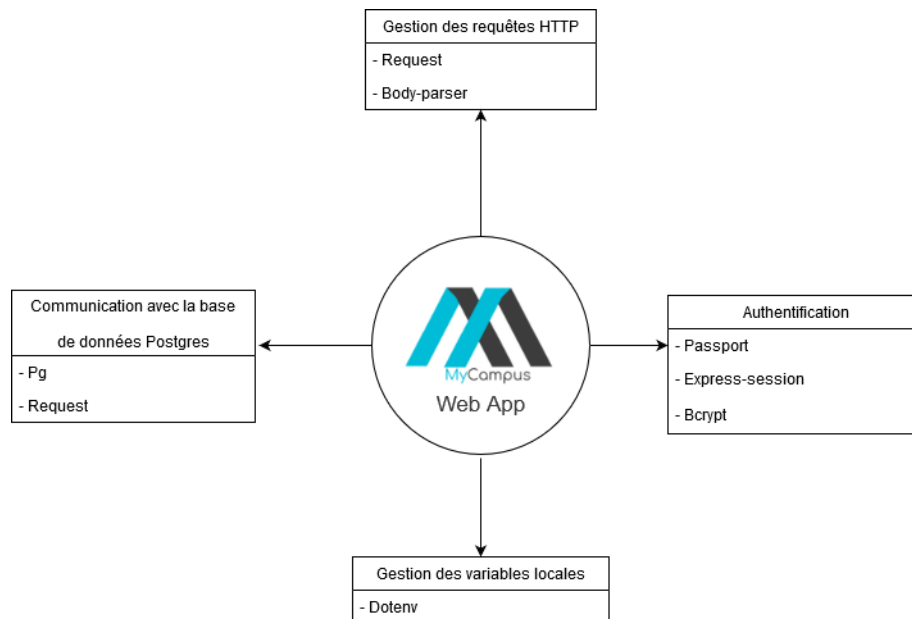


FIGURE 11 – Relations entre l'application et ses dépendances

3.2 Fonctionnalités

Authentification Pour pouvoir accéder à l'interface de gestion de colis, l'utilisateur doit être authentifié. Cela a pour but d'empêcher toute personne de modifier la base de données. C'est là où l'interface d'authentification entre en jeu : (Voir figure ci-dessous). Si les identifiants renseignés ne sont pas corrects, un message s'affiche et permet à l'utilisateur de réessayer. Dans le cas contraire, la redirection vers l'interface de gestion des colis est automatique.



FIGURE 12 – Interface d'authentification

Interface CRUD L'interface est le coeur de l'application, elle permet aux employés de la Maisel de modifier la base de données sans connaissance du langage SQL. Pour cela, l'interface est la plus intuitive possible avec un formulaire à remplir pour ajouter des colis, un tableau avec des pages de navigation pour accéder à la liste des données, et deux boutons pour modifier ou supprimer un colis (Voir figure ci-dessous).

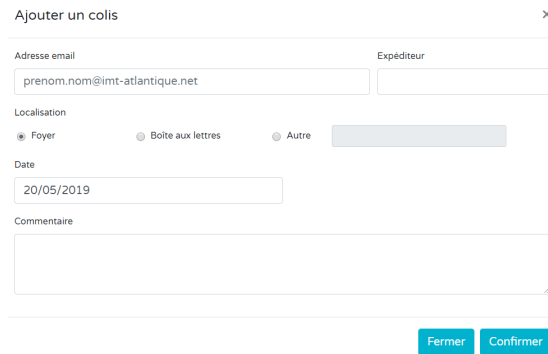


FIGURE 13 – Formulaire d'ajout d'un colis

Autocomplétion Afin de faciliter l'ajout et la modification des colis, l'autocomplétion permet de remplir l'email du récepteur du colis plus rapidement en entrant que les premières lettres (Voir figure ci-dessous). Le programme se charge ensuite de remplir les champs nom et prénom. Dans le cas d'insertion d'une adresse email ne figurant pas dans la base de données IMT Atlantique, un message d'erreur s'affiche.

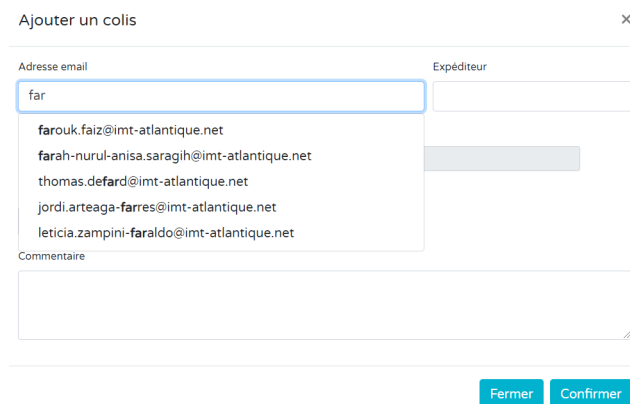


FIGURE 14 – Autocomplétion du champ adresse email

Personnalisation de l’affichage Afin de rendre la navigation plus agréable pour l’utilisateur, des options de personnalisation d’affichage sont disponibles.

- Le nombre de colis par page est configurable directement depuis un champ “select”. La navigation dans la liste des colis se fait ensuite en parcourant les pages disponibles en bas.
- L’utilisateur a la liberté de choisir entre un affichage compact du tableau ou un affichage large avec plus d’espaces.

Liste des colis							Télécharger la liste en PDF	Ajouter un colis
Id	Nom	Prénom	Email	Expéditeur	Localisation	Date	Commentaire	
58	FAIZ	Farouk	farouk.faiz@imt-atlantique.net	Colissimo	Foyer	28/4/2019		
57	FAIZ	Farouk	farouk.faiz@imt-atlantique.net	Colissimo	Boîte aux lettres	28/4/2019		
56	FAIZ	Farouk	farouk.faiz@imt-atlantique.net	Colissimo	Maisel	28/4/2019		
55	FAIZ	Farouk	farouk.faiz@imt-atlantique.net	Colissimo		28/4/2019		
54	FAIZ	Farouk	farouk.faiz@imt-atlantique.net	Colissimo	Foyer	28/4/2019		
53	FAIZ	Farouk	farouk.faiz@imt-atlantique.net	Colissimo	Boîte aux lettres	28/4/2019		
52	FAIZ	Farouk	farouk.faiz@imt-atlantique.net	Colissimo		28/4/2019		
51	FAIZ	Farouk	farouk.faiz@imt-atlantique.net	Colissimo	Foyer	28/4/2019		
50	DOUBLI	Youssef	youssef.doubli@imt-atlantique.net	Colissimoo		28/4/2019		
49	DOUBLI	Youssef	youssef.doubli@imt-atlantique.net	UPS		28/4/2019		

10 résultats sur 89

Précédent 4 5 6 7 8 Suivant

☒ Compact ☐ Large

10

FIGURE 15 – Interface principal de navigation des colis

Génération de PDF Une fois les colis du jour ajoutés, les étudiants ont besoin de signer pour affirmer la récupération de leurs colis. Pour cela, rien ne change. Ils auront toujours un papier à disposition avec la liste des colis et signeront dans la case de leur colis une fois récupéré. Pour faciliter la tâche au personnel de la Maisel, la génération de la feuille se fait automatiquement en cliquant sur un bouton “Télécharger la liste en PDF”. Ils n’auront ensuite qu’à l’imprimer et l’envoyer avec les colis aux foyer comme d’habitude. (Voir figure ci-dessous)

Liste des colis



Nom	Prénom	Expéditeur	Date	Signature
Farouk	FAIZ	Colissimo	17/5/2019	
Julien	MALLET	dsdsd	15/5/2019	
Farouk	FAIZ	Colissimo	9/5/2019	
Farouk	FAIZ	UPS	7/5/2019	
Youssef	DOUBLI	sdf	6/5/2019	
Youssef	DOUBLI	hh	6/5/2019	
Youssef	DOUBLI	l	6/5/2019	
Farouk	FAIZ	Colissimo	4/5/2019	

FIGURE 16 – Extrait du PDF généré

Après le développement logiciel de l'application, nous allons aborder une solution matérielle qui aura pour but de remonter les informations sur les états des machines à laver et les équipements des cuisines.

4 Composante matérielle : solutions capteurs

- *Écrit par : EL QORCHI Imane*
- *Relecture par : FAIZ Farouk*

4.1 Problématique

Réaliser un prototype électronique lié aux dispositifs de la laverie et des cuisines afin de capter les états de ces dispositifs (machine allumée ou éteinte), et ainsi créer une base de données versatile et dynamique de ces états qui permettra au chatbot de les transmettre à un instant précis aux utilisateurs. Ceci avec une assez bonne estimation et minimisation du coût total. Deux solutions de mesure de grandeur physique s'imposent :

- La mesure du courant électrique des plaques chauffantes.
- La mesure des vibrations du châssis des machines à laver .

4.2 Première solution : capteur de courant

4.2.1 Prototypage

Le choix le plus judicieux se porte sur des capteurs de courants non intrusifs commercialisés sur le site Amazon à un prix unitaire de 9,99 euros, chaque machine devrait

disposer de son capteur inductif branché sur un Arduino nano, le tout dans une boîte de dérivation qui n'est pas neutre électriquement, pour neutraliser l'ensemble il faudrait un boîtier-raccord pour deux machines. L'alimentation passe par un convertisseur 12V – 5V pour la carte Arduino qui fonctionne sous 5V.

La carte Arduino mesure la valeur de la tension en sortie du capteur et envoie la valeur. Un programme Arduino pourrait permettre de discrétiser ces résultats sous forme de 0 dans le cas d'un courant nul, puis la valeur 1 sinon. Les données seront par la suite remontées par un module RaspBerry en ethernet ou un module Bluetooth Arduino, et déployées dans la base de données.

4.2.2 Devis de l'installation pour la laverie

ci-dessous est un devis de l'installation en laverie (20 machines) uniquement à titre indicatif.

Composant	Quantité	Prix total
Boîte de Dérivation Étanche 113 x 113 x 60 mm	x 20	= 40 euros
Carte microcontrôleur Arduino Nano 65250 A000005	x 20	= 300 euros
Convertisseur 12V cc, 5 V c.c	x 1	= 10 euros
Capteur de courant alternatif non intrusifs	x 20	= 180 euros
Boitier raccord	x 10	= 200 euros
module bluetooth HC-05	x 20	= 60 euros
Total		= 790 euros

4.3 Deuxième solution : capteur de courant

4.3.1 Prototypage

Cette solution de mesure de vibrations se porte sur des capteurs piézoélectriques qui envoient des pics de courant sous contrainte mécanique, branchés sur une carte Arduino Nano dans un boîtier soudé au châssis de la machine. Nous pourrions même pas la suite, grâce aux vibrations, déterminer le mode de fonctionnement de la machine et donc la durée restante (si la machine est en mode pré-lavage, essorage ...) grâce à un étalonnage.

Les données seront également remontées par un module RaspBerry en ethernet ou un module bluetooth Arduino et par la suite déployées dans la base de données.

4.3.2 Devis de l'installation pour la laverie

Ci-dessous est un devis de l'installation en laverie (20 machines) uniquement à titre indicatif.

Composant	Quantité	Prix total
Carte microcontrôleur Arduino Nano 65250 A000005	x 20	= 300 euros
Capteur de vibrations piézoélectrique	x 20	= 20 euros
module bluetooth HC-05	x 20	= 60 euros
Total		= 420 euros

4.4 Contraintes techniques et logistiques des installations

Ces installations bien que cohérentes sur le plan technique, sont difficilement réalisables. La solution des capteurs de courant est très coûteuse et nécessite de lourds aménagements dans la laverie qui devraient être effectués par un professionnel, ceci sans perturber l'utilisation des machines.

En ce qui concerne la deuxième solution, elle s'avère également difficilement réalisable, puisque les vibrations des machines sont extrêmement faibles.

En conclusion, le projet est difficilement réalisable à ce stade-ci, néanmoins plusieurs autres pistes peuvent être étudiées.

Une fois la réalisation technique de notre projet complète, nous allons fixer dans la section qui suit, un processus de tests nécessaires pour la validation du projet.

5 Tests et validation du projet

- *Écrit par : MASSAOUD Hamid*
- *Relecture par : FAIZ Farouk*

5.1 Contraintes

Remarque : Les contraintes -C- mentionnées ci-dessous sont ordonnées par ordre de priorité.

Pour valider notre projet, il faudrait que les acteurs principaux, à savoir les résidents et l'employé de la Maisel, puissent valider l'efficacité des deux composantes logicielles (chatbot et site web).

Pour cela il convient de définir un certain nombre de critères de validation, on cite :

- C1 : Gérer les ressources limitées du serveur d'exécution. Un serveur gratuit n'offre que des ressources très limitées au chatbot au niveau de stockage et de mémoire.
- C2 : Maintenir une disponibilité permanente du service. Des pannes éventuelles ou un grand trafic entrant ne doivent pas interrompre le bon fonctionnement du chatbot.

- C3 : Garder un temps de réponse inférieur à 2s pour le chatbot. En effet, le temps d’interrogation de la base de données dépend de la taille de cette dernière. Plus le nombre d’utilisateurs accroit, plus le temps de traitement augmente.
- C4 : Disponible en plusieurs langues : au minimum français et anglais.
- C5 : Ergonomie du site web.

5.2 Bilan du projet

Il est important de prévoir un processus de validation. La plateforme bénéficiera des périodes de tests où l’on va mesurer son impact sur la vie des étudiants et sur le travail du responsable des colis à la Maisel. Ces tests vont nous permettre de l’adapter au mieux aux besoins réels. En effet, les personnes qui utiliseront le chatbot pourront communiquer leurs retours via un formulaire préalablement rédigé. Ainsi, une fois les retours des utilisateurs s’avèrent positifs, le chatbot serait accessible pour l’ensemble de étudiants.

Conclusion

- *Écrit par : MASSAOUD Hamid*
- *Relecture par : EL QORCHI Imane*

L’intérêt d’une telle plateforme est donc d’apporter une amélioration au niveau de la gestion des ressources de la Maisel. Grâce au travail organisé et la bonne répartition des tâches, nous avons réussi finalement à avoir un résultat satisfaisant. Notre interface Web pour l’employé de la Maisel est déjà prête pour l’utilisation. Bien que les services des cuisines et de la laverie n’ont pas pu être implémentés suite à des complications au niveau technique et financier, le chatbot répond bien à notre exigence fonctionnelle prioritaire définie dans le cahier des charges à savoir : Vérifier l’arrivée des colis pour les résidents du campus.

D’autres fonctions complémentaires ont donc pu être rajoutées à savoir le service covoiturage, le service RAK et cafétéria et enfin le service emploi du temps.

En revanche, notre projet présente une base susceptible d’être améliorée surtout en ce qui concerne les deux fonctions de service prioritaires : vérifier la disponibilité des machines à laver et les cuisines dans le campus. En effet, la possibilité d’implémenter ces services et équiper les dispositifs demeure toutefois envisageable, et constitue une réelle projection pour notre projet.

Glossaire

- *Écrit par : DOUBLI Youssef*
- *Relecture par : EL QORCHI Imane*

- ADEsoft : un outil de planification et de gestion utilisé par des établissements pour gérer les emplois temps.
- API : permet de rendre disponibles les données ou les fonctionnalités d'une application existante afin que d'autres applications les utilisent.
- Back End : la partie serveur qui gère les chemins vers les pages et l'interaction avec la base de données.
- Boîtier de dérivation : boîtier électrique située en aval d'un dispositif électrique. il est situé au plus près des installations et permet de centraliser les départs vers les différents appareils électriques.
- Bootstrap : framework (Ensemble de code écrit par d'autres développeurs) qui permet d'avoir des résultats visuels intéressants sans avoir à trop écrire son propre CSS.
- Capteur de courant inductif : un appareil utilisant le principe de l'induction électromagnétique pour détecter les variations du courant électrique.
- Capteur piézoélectrique : un capteur utilisant l'effet piézoélectrique (propriété que possèdent certains corps de se polariser électriquement sous l'action d'une contrainte mécanique et réciproquement de se déformer lorsqu'on leur applique un champ électrique) afin de mesurer des grandeurs physiques.
- Chatbot : un robot logiciel pouvant dialoguer avec un individu ou consommateur.
- CLI : une interface en ligne de commande. c'est une interface homme-machine dans laquelle la communication entre l'utilisateur et l'ordinateur s'effectue en mode texte.
- CSS : Langage qui sert à décrire la présentation et le style d'un document HTML.
- Dialogflow : une interface qui permet d'utiliser l'intelligence de Google. Il permet de reconnaître des phrases envoyées par l'utilisateur.
- EJS : générateur de code HTML à partir d'une template remplie avec des données fournies.
- Hash : un hash cryptographique est une séquence de chiffres et lettres générée par une fonction cryptographique.
- HTML : langage de balisage utilisé pour la création de pages web.
- Http : protocole de transmission permettant à l'utilisateur d'accéder à des pages web par l'intermédiaire d'un navigateur.
- IHM : une interface Homme-Machine permet à l'utilisateur humain d'interagir avec une machine.
- Intent(intention en français) : une notion de l'API DialogFlow qui représente une intention de l'utilisateur et entraîne l'agent à reconnaître cette intention dans une

- discussion.
- Javascript : langage de programmation qui permet d'implémenter les mécanismes dynamiques d'une page web.
 - JQuery : librairie javascript qui facilite l'interaction avec la partie HTML/CSS.
 - Modal : composante du framework Bootstrap. C'est une petite fenêtre qui s'affiche sur la page actuelle pour éviter de changer de page.
 - Module Bluetooth HC05 : permet d'ajouter une liaison Bluetooth à un micro-contrôleur via une communication série TTL.
 - MySQL : un système de gestion de bases de données relationnelles.
 - NLP : Natural Language Processing est une branche très importante du Machine Learning qui s'intéresse à la capacité d'un programme à comprendre le langage humain.
 - NodeJS : Langage permettant d'écrire du code pour contrôler la partie serveur (Back end). Sa syntaxe est celle même du javascript.
 - Parsing : consiste à analyser des données et les segmenter en éléments plus petits et caractéristiques.
 - SGBD : un système de gestion de base de données est un logiciel système servant à stocker, à manipuler ou gérer, et à partager des informations dans une base de données.
 - SMTP : un protocole de transfert de message.
 - UUID : Universally Unique IDentifier. c'est une série de lettres et chiffres générés par la machine de telle sorte qu'elle soit garantie d'être unique.
 - Web-scraping : une technique d'extraction du contenu de sites web dans le but de le transformer pour permettre son utilisation dans un autre contexte.

Annexes

Vu la grande taille des plannings (prévisionnel et réel), nous les avons découpé pour améliorer la lisibilité, néanmoins l'impression sur A4 peut toujours les rendre peu lisible.

Les **Fonctions de service Principales** que vous allez rencontrer en lisant les deux plannings, sont des fonctions qui ont été définies dans l'analyse fonctionnelle de notre cahier des charges. Voici donc, les trois objectifs que nous voulons atteindre lorsque nous avons commencé à travailler sur MyCampus :

- **FP1** : Vérifier l'arrivée des colis pour les résidents du campus. En effet, une fois le responsable des colis à la Maisel déclare un colis arrivé en son nom, le résident reçoit une notification.
- **FP2** : Vérifier la disponibilité des machines à laver dans le campus.
- **FP3** : Vérifier la disponibilité des cuisines dans le campus.

Pour planifier l'ensemble des activités du projet et pour piloter de façon à respecter aux mieux les engagements initiaux décrits dans le cahier des charges, nous avons mis en place un planning prévisionnel sur lequel nous estimons les durées à respecter et aussi les tâches à réaliser. L'outil utilisé est le Diagramme de Gantt, car il permet une visualisation globale des diverses missions qui composent notre projet. Il est le fruit d'une construction collective.

Notre projet débute réellement en Septembre dans le cadre du projet DDRS (Développement Durable et Responsabilité Sociétale). Il nécessite un développement technique considérable, pour cette raison nous l'avons proposé comme un projet CODEV au mois de Décembre. Nous avons respecté le planning prévisionnel dans son intégralité, les durées à respecter ont été légèrement modifiées.

Vous trouverez ci-dessous une analyse complète des écarts entre les des deux plannings :
Pour **FP1** :

Planning Prévisionnel	Planning Réel	Commentaire
Date Début : 15/03/2019 Date Fin : 16/03/2019	Date Début : 15/03/2019 Date Fin : 01/05/2019	Nous avons fini plus tôt notre fonction de service prioritaire. En effet, nous avons pu profiter des vacances d'Avril et notre séjours sur le campus de Nantes pour pouvoir avancer dans notre projet.

Pour les deux fonctions prioritaires **FP2** et **FP3**, nous avons décidé de ne pas les mettre en place, la section 4 de notre rapport (page 19) explique que les deux fonctions sont difficilement réalisables. Pour cette raison, nous avons opté de les substituer par trois

nouvelles fonctions complémentaires :

- **FC1** : Connaître le menu du RAK et celui de la cafétéria.
- **FC2** : Connaître l'emploi du temps.
- **FC3** : Savoir les offres de covoiturage proposées par les élèves brestois.

Pour **FC1** :

Planning Prévisionnel	Planning Réel	Commentaire
Date Début : 10/05/2019 Date Fin : 23/05/2019	Date Début : 01/05/2019 Date Fin : 14/05/2019	En lisant, notre planning prévisionnel, vous constatez que les deux dates dans le planning prévisionnel correspondent à celles de la FP2. Comme notre FP1 est déjà fonctionnelle, nous avons pu en profiter pour pouvoir réaliser la FC1.

Pour **FC2** :

Planning Prévisionnel	Planning Réel	Commentaire
Date Début : 10/05/2019 Date Fin : 23/05/2019	Date Début : 01/05/2019 Date Fin : 14/05/2019	Les deux dates dans le planning prévisionnel correspondent à celles de la FP3. Nous avons pu réaliser la FC2 rapidement car son implémentation est similaire à celle de la FC1.

Pour **FC3** : *Cette fonction a été ajoutée suite à notre échange avec le projet Covoiturage. En effet, l'idée d'une collaboration nous tentait dès le jour de notre Forum DDRS qui était sur Brest.*

Planning Prévisionnel	Planning Réel	Commentaire
Date Début : - Date Fin : -	Date Début : 23/05/2019 Date Fin : 01/05/2019	Pour réaliser cette fonction, nous avons contacté les élèves qui développent le projet covoiturage pour une collaboration collective. Nous avons demandé leur permission d'utiliser leur site Web pour avoir les différentes informations concernant les offres.

Le mois de Mai était aussi une occasion pour prendre rendez-vous avec la Maisel. Notre rencontre avec la responsable des colis Mme Collet était nécessaire. D'ailleurs, l'échange que nous avons eu avec elle, était intéressant car il nous a permis de mieux adapter notre interface à ses besoins.

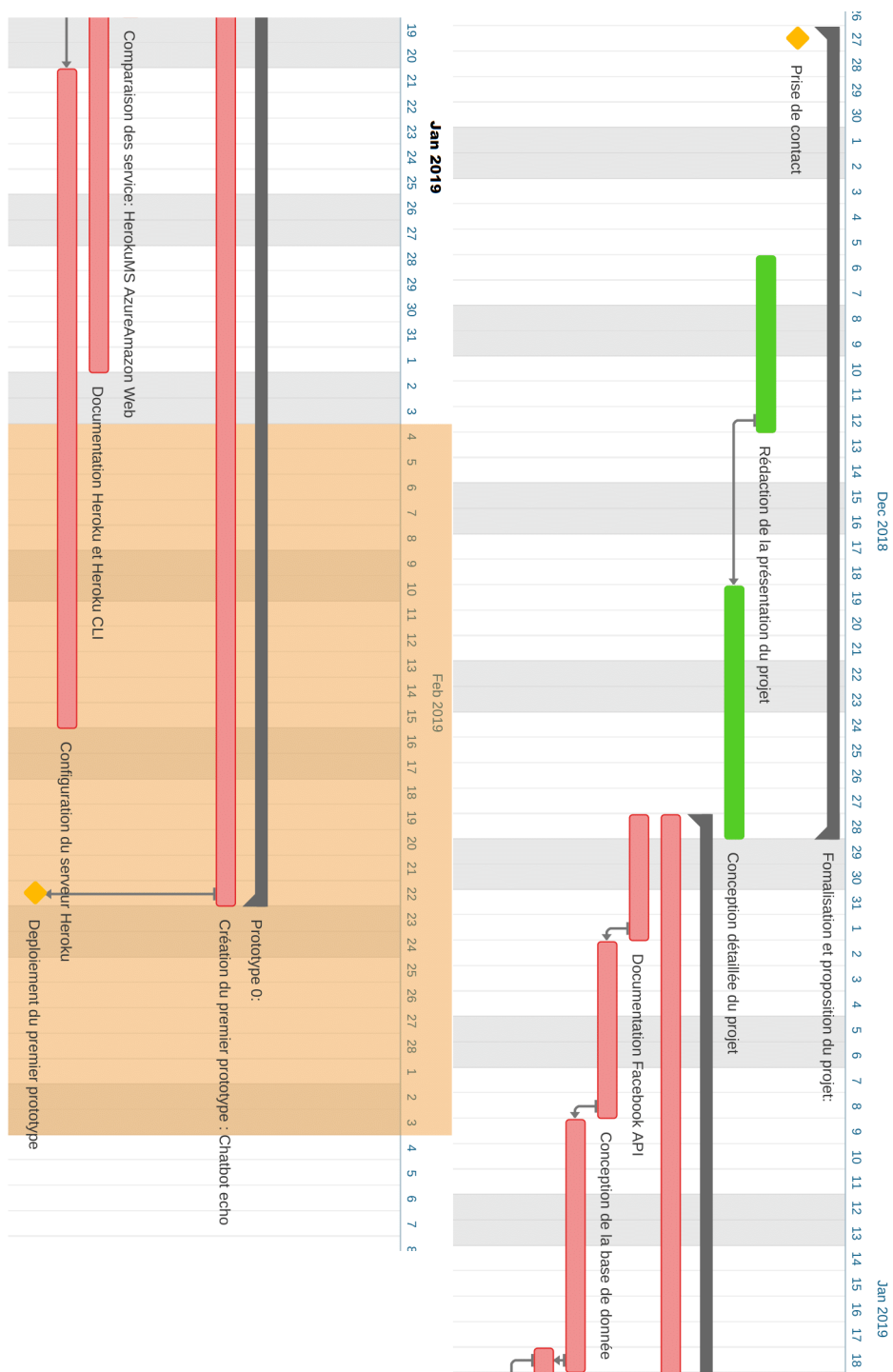


FIGURE 17 – Planning prévisionnel : Partie 1

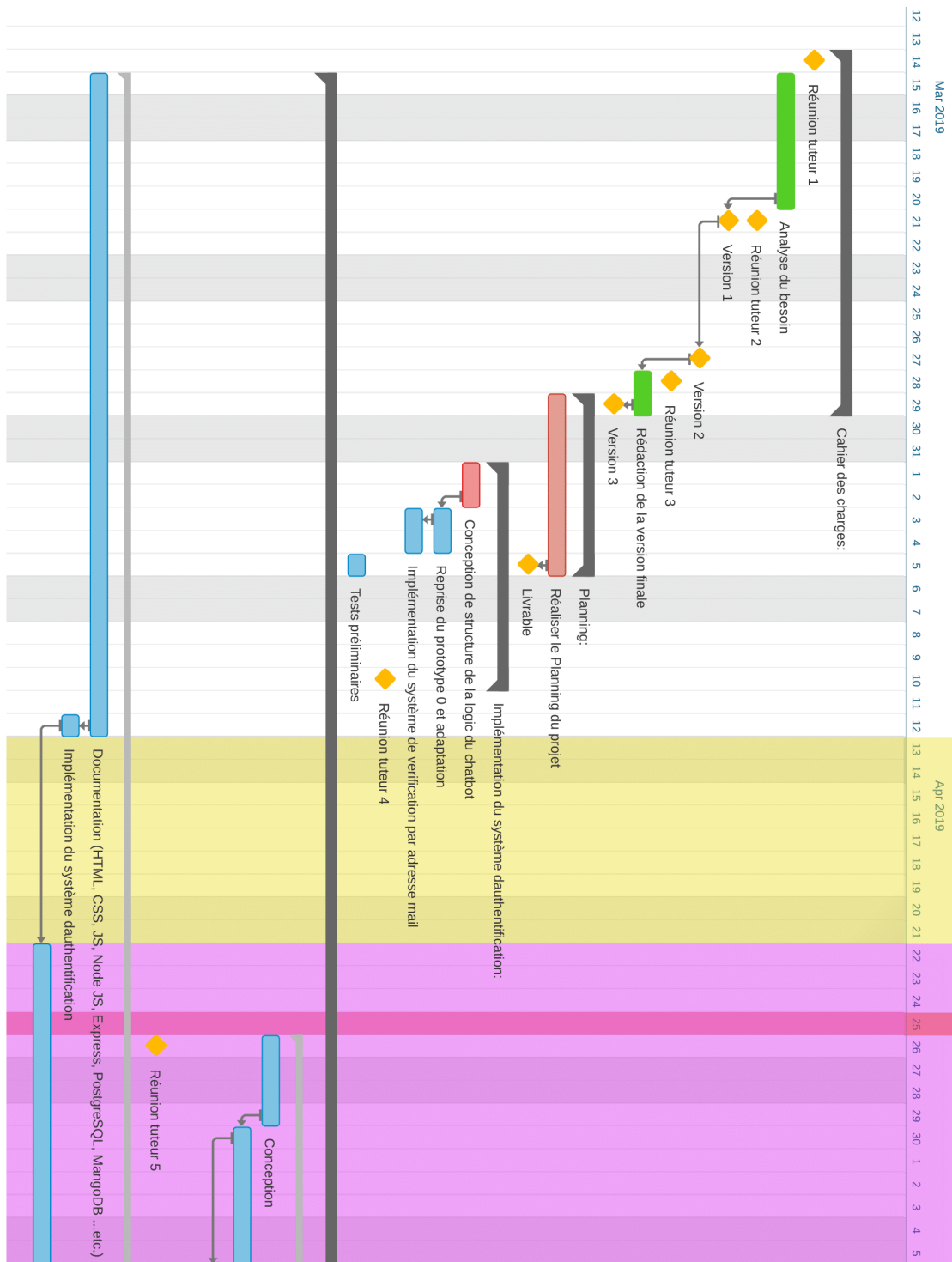


FIGURE 18 – Planning prévisionnel : Partie 2

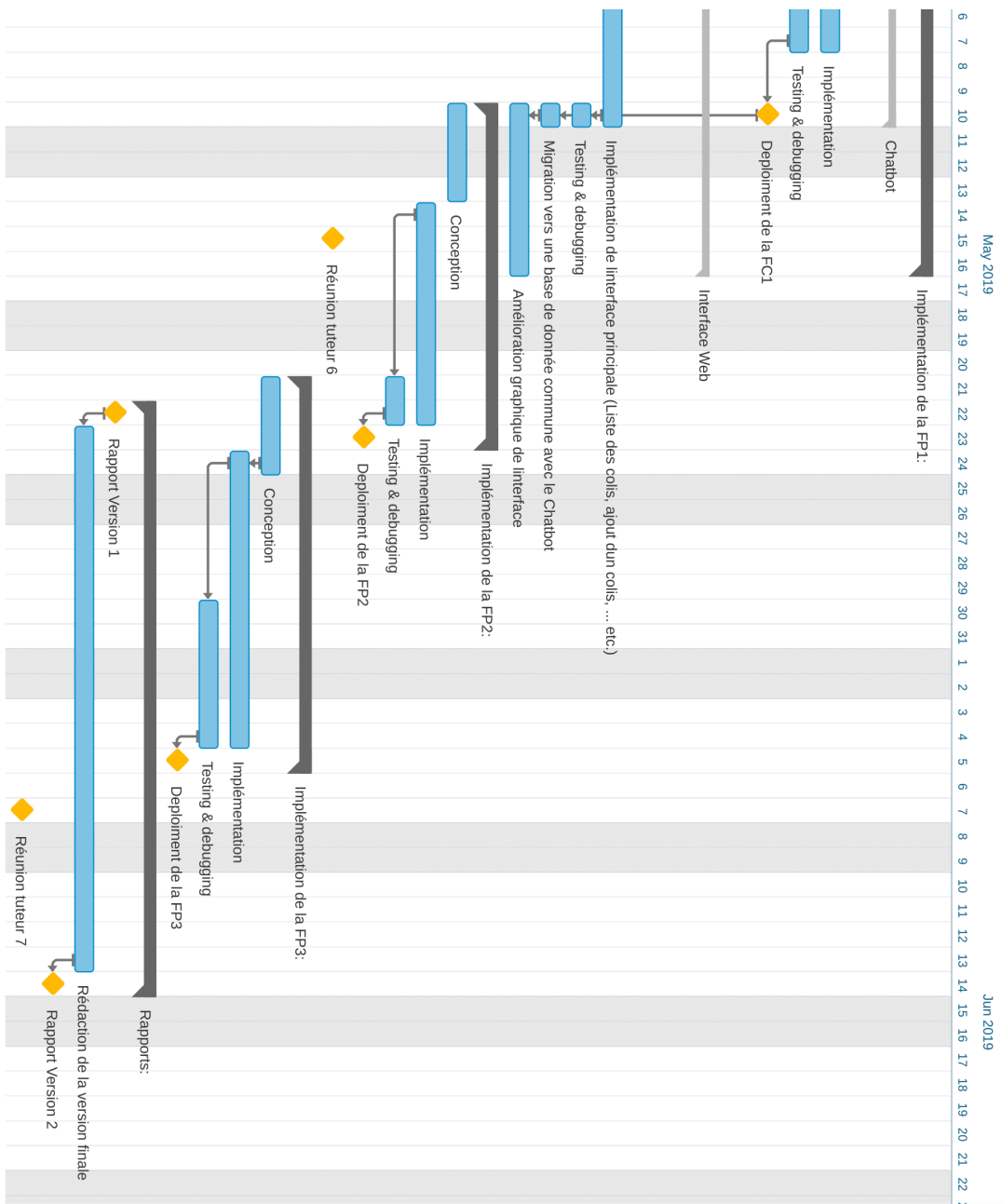


FIGURE 19 – Planning prévisionnel : Partie 3

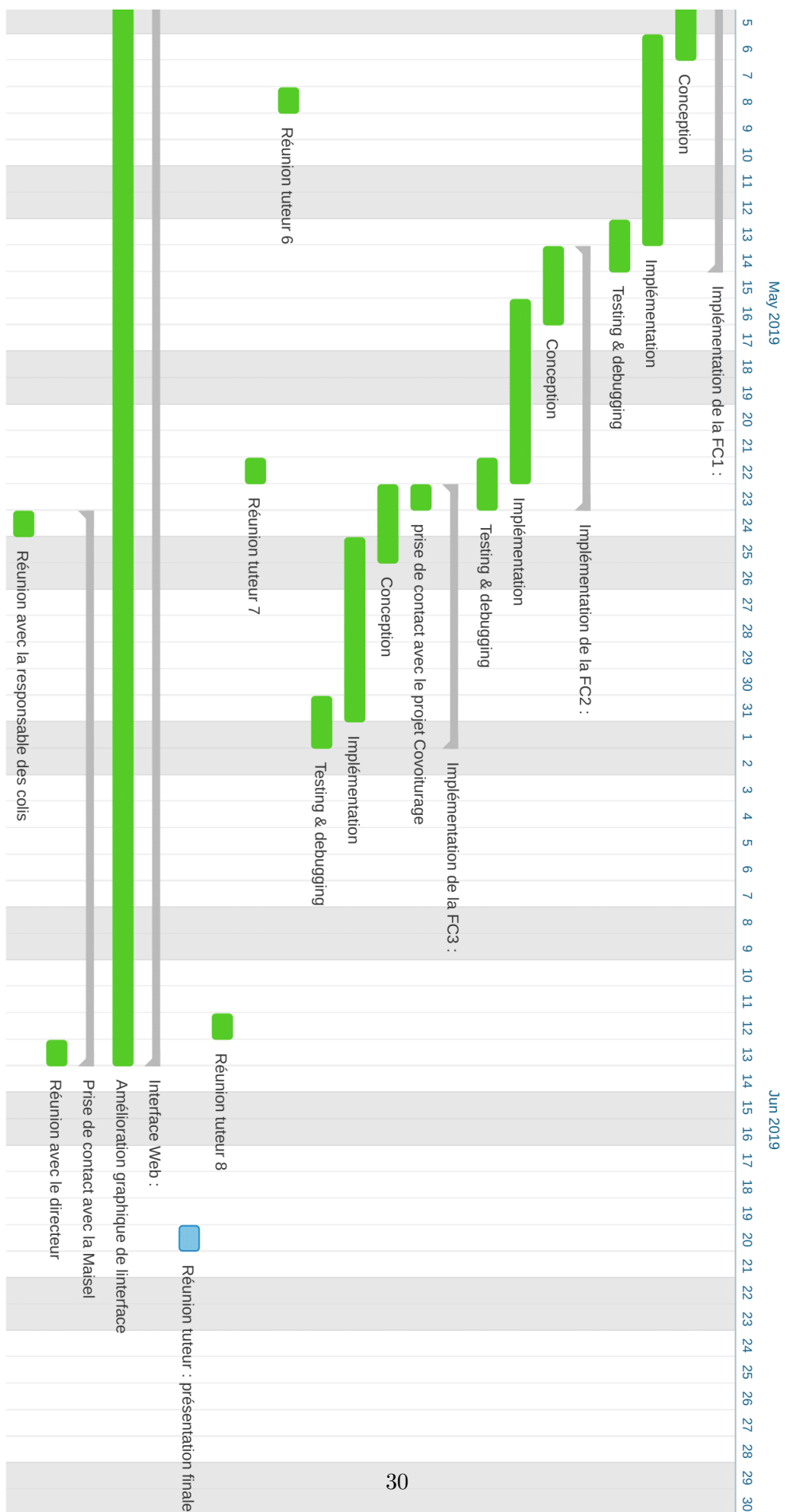


FIGURE 20 – Planning réel

Bibliographie

- [1] SAMUEL RONCE, *Comprendre et développer un Chatbot*, Eyrolles, 07/12/2017, 328, 978-2-212-67453-8.
- [2] NodeJS - The Complete Guide (incl. MVC, REST APIs, GraphQL), consulté le 05/03/2019.
<https://www.udemy.com/course/nodejs-the-complete-guide/>
- [3] Documentation Facebook For Developers, consulté le 25/12/2018.
<https://developers.facebook.com/docs>
- [4] The Web Developer Bootcamp, consulté le 11/03/2019.
<https://www.udemy.com/the-web-developer-bootcamp/learn/lecture/3861710>
- [5] Learn about basic Dialogflow concepts, consulté le 12/04/2019.
<https://dialogflow.com/docs>
- [6] Documentation Facebook For Developers, consulté le 25/12/2018.
<https://developers.facebook.com/docs>