

Cahier des charges de projet GL02 - Équipe 3MR

Version 0.1 (péface done, à vérifier) Matthieu Soliman, Maxence Barbançon, Maxime Huyghe, Ronan Gasselin

Préface (1 page)

Ce document s'adresse à l'équipe qui entreprendra le développement de ce projet ainsi qu'à notre responsable de TD.

Il liste d'abord brièvement les exigences fonctionnelles et non-fonctionnelles, avant de donner plus de détails sur ces exigences et sur les formats de données utilisés.

Ce document sera structuré en huit parties distinctes qui seront énumérées et définies dans la préface. La première partie est la préface, dans cette partie nous expliquons aux interlocuteurs comment lire ce document afin de comprendre pleinement ce qui y est spécifié. La partie suivante est l'introduction dans laquelle sera rappelé le contexte d'écriture de ce cahier des charges et du développement du projet dans son ensemble. Elle permettra aussi de rappeler et d'éclaircir les objectifs de ce projet logiciel. Le cahier des charges présentera ensuite un glossaire où seront réunis tous les termes techniques et expressions qui nécessitent une redéfinition ou un éclaircissement supplémentaire pour aider à la compréhension des lecteurs. Y seront définis seulement les termes ressortant du projet, les termes techniques nécessitant des connaissances ou recherches annexes sont laissés à la discrétion du lecteur. La partie quatre de ce document mettra l'accent sur la **spécification générale des exigences**. Ce chapitre listera les **exigences fonctionnelles et non fonctionnelles** dans une brève introduction et les exprimera en langage naturel. A chaque exigence sera associé un numéro d'identification unique permettant d'y faire référence dans la section suivante. Dans la cinquième partie les lecteurs pourront y retrouver les **spécifications détaillées** qui préciseront chacune des exigences de manière structurée. Chaque exigence sera composée d'un identifiant, un titre, une liste d'objectifs, des pré et postconditions, la liste des entrées, des traitements effectués et des sorties. Ces exigences pourront être accompagnées d'une description ou d'une maquette illustrant les sorties visuelles si cela est pertinent. Elle détaillera également une section **format de données** où sera spécifié les différents formats utilisés dans l'application en respectant le formalisme **Augmented Backus-Naur Form** (ABNF). Enfin une section orientée sur la **sémantique des données** permettra aux lecteurs d'appréhender l'objet principal de ce projet logiciel, les e-mails, grâce à une **spécification algébrique**. Y seront détaillés les opérateurs temporels et d'équivalence de ce type d'objet, une **spécification algébrique d'une collection** permettant de le contenir ainsi que les opérations et axiomes associés. Une conclusion permettant de mettre en perspective les objectifs du projet et les exigences détaillées plus haut dans le cahier des charges ainsi que des conseils pour l'équipe de développement qui implémentera ce projet sera rédigée

en sixième partie. Enfin une bibliographie et une annexe viendront compléter ce dossier.

Ce document sera à la fois versionné sur un système de contrôle de version comme Git ou SVN et par un numéro de version sous le titre qui devra être incrémenté à chaque changement, sauf s’il ne consiste que de corrections orthographiques ou similaire.

Introduction (1 page)

Le cabinet de conseil UIConsult fait appel à notre équipe car il souhaite se doter d’un outil d’aide à l’analyse des communications. Leur demande fait suite au besoin de donner aux responsables de secteur la possibilité de produire un rapport d’analyse des échanges par e-mails des différents collaborateurs. Ce qui leur permettra un suivi individuel, ceci étant pertinent pour la préparation des entretiens d’évaluation réguliers sur leurs performances. Un tel rapport est appelé “RCom” au sein du cabinet.

a besoin d’analyser des données issues des e-mails envoyés et reçus par ses collaborateurs

L’équipe de développement devra répondre aux principaux objectifs détaillés ci-dessous. - extraire les e-mails. - extraire les informations des contacts des collaborateurs au format VCard (RFC6350). - développer un outils en ligne de commande

Glossaire (si nécessaire)

- RCom : rapport d’analyse des échanges par email des collaborateurs

Liste exigences fonctionnelles et non-fonctionnelles

L’objectif est d’extraire les données importantes des boîtes emails et préparer les graphiques au format SVG (et/ou PNG) à ajouter dans les RCom. Un autre enjeu est aussi d’extraire des emails certaines données personnelles d’un collaborateur (adresse email, contenu, nom, fonction) sous format VCard (RFC6350).

[10 max fonctionnelles, quelques non-fonctionnelles] - SPEC1 Compter les emails échangés sur une période - SPEC2 Retourner la liste des “buzzy days” (email écrit le week-end ou la nuit) - SPEC3 Donner le top10 des interlocuteurs d’un collaborateur - SPEC4 Donner le top10 des termes les plus utilisés dans les objets d’une boîte mail - SPEC5 Visualiser le nombre d’échanges entre collaborateurs pour une boîte mail donnée (nuage de points) - SPEC6 Retrouver des emails avec différent critères de recherche : nom, prénom, email - SPEC7 Extraire les contacts d’un collaborateur au format VCard (RFC6350) - SPEC_NF1 Retourner les

données sous forme de phrases compréhensibles et lisibles par un lecteur d'écran
//?

Spécification détaillée (8-15 pages)

Exigences

format : Identifiant | Titre | Objectifs | Précondition(s) | Postcondition(s) |
Entrées | Traitements | Sorties | Gestion des erreurs

SPEC1

Titre | Compter les emails échangés sur une période Objectifs | on veut savoir le
nombre de mails échangés soit au total soit sur une période temporelle donnée
Préconditions Postconditions Entrées Traitements Sorties Gestion des erreurs |
il n'y a pas de mails échangés dans la période donnée -> retourner 0

SPEC2

Titre | Retourner la liste des “buzzy days” (email écrit le weekend ou la nuit)
Objectifs | on cherche à obtenir toutes les fois où un mail a été envoyé sur un
créneau horaire informel Préconditions |néant Postconditions | néant Entrées |
structure et contenu de la boîte mail d'un collaborateur Traitements | demander
si on veut retourner les mails envoyés le weekend et/ou la nuit ou les deux
Sorties| liste de mails, nombre de résultats Gestion des erreurs | si aucun résultat
correspondant, afficher un message le notifiant

SPEC3

Titre | Donner le top10 des interlocuteurs d'un collaborateur Objectifs | on
aimerait avoir la liste des 10 interlocuteurs privilégiés d'un collaborateur (ceux
qui interagissent le plus avec lui) Préconditions | néant Postconditions | néant
Entrées | structure et contenu de la boîte mail d'un collaborateur Traitements
| Sorties | liste de contacts Gestion des erreurs | cas de figure rare où un
collaborateur a interagi avec moins de 10 destinataires différents

SPEC4

Titre | Donner le top10 des termes les plus utilisés dans les objets d'une boîte mail
Objectifs | on désire obtenir les 10 mots les plus récurrents dans les objets des
mails (reçus ou envoyés) du collaborateur Préconditions | néant Postconditions
| néant Entrées | structure et contenu de la boîte mail d'un collaborateur
Traitements | exclure du comptage les mots communs et ordinaires (conjonction
“et”, “d” apostrophe, etc) Sorties | liste de 10 mots Gestion des erreurs | cas de
figure (très rare) où peu de mails ont été reçus/envoyés

SPEC5

Titre | Visualiser le nombre d'échanges entre collaborateurs pour une boîte mail donnée (nuage de points) Objectifs | il s'agit de produire un retour graphique (SVG/PNG) illustrant la fréquence des échanges du collaborateur avec les autres collaborateurs Préconditions | liste des collaborateurs de UIConsult connue Postconditions | néant Entrées | structure et contenu de la boîte mail d'un collaborateur Traitements | inclure ou exclure certains collaborateurs de l'analyse Sorties | graphique sous format SVG ou PNG Gestion des erreurs | néant

SPEC6

Titre | Retrouver des emails avec différent critères de recherche : nom, prénom, email Objectifs | on veut obtenir une liste de mails en fonction du nom/prénom/mail du destinataire ou de l'émetteur Préconditions | néant Postconditions | néant Entrées | critères de recherche sur l'émetteur ou le destinataire (au choix) Traitements | Sorties | liste de mails, nombre de résultats Gestion des erreurs | si aucun résultat correspondant, afficher un message le notifiant

SPEC7

Titre | Extraire les contacts d'un collaborateur au format VCard (RFC6350) Objectifs | on veut pouvoir donner la liste de contacts d'un collaborateur selon le format de coordonnées personnelles imposé Préconditions | néant Postconditions | néant Entrées | structure et contenu de la boîte mail d'un collaborateur Traitements | Sorties | liste de contacts au format VCard (RFC6350) Gestion des erreurs |

SPEC_NF1

Titre | Retourner les données sous forme de phrases compréhensibles et lisibles par un lecteur d'écran //pas sûr Objectifs | Préconditions | Postconditions | Entrées | Traitements | Sorties | Gestion des erreurs |

Formats de données

Email

```
; Some headers could be validated in the grammar.
; Not doing it because they are not relevant to this software.
email =
    "Message-ID" hsep message-id-line
    "Date" hsep date-line
    "From" hsep email-line
    "To" hsep email-line
    "Subject" hsep line
    "Mime-Version" hsep line
```

```

"Content-Type" hsep line
"Content-Transfer-Encoding" hsep line
"X-From" hsep line
"X-To" hsep line
"X-cc" hsep text *(", " text) crlf
"X-bcc" hsep text *(", " text) crlf
"X-Folder" hsep line
"X-Origin" hsep line
"X-Filename" hsep line
*line

; Header separator
hsep = ": "

line = text crlf

text = *(vchar/wsp)

message-id-line = "<" text "@" text ">" crlf

date-line = day-of-week ", " date wsp time crlf
day-of-week = "Mon" / "Tue" / "Wed" / "Thu" / "Fri" / "Sat" / "Sun"
date = day wsp month wsp 4digit
day = [(1-3)] digit
month = "Jan" / "Feb" / "Mar" / "Apr" / "May" / "Jun" / "Jul" / "Aug" / "Sep"
      / "Oct" / "Nov" / "Dec"
time = hour ":" min-or-sec ":" min-or-sec wsp ("-" / "+") 4digit "(" 3(A-Z) ")"
hour = (0-2) digit
min-or-sec = (0-5) digit

email-line = email crlf
; We shouldn't assume the host part of an email address is a URL
email = text "@" text

```

Format VCard

Sémantique des données

Conclusion

Bibliographie

Annexes