



UNIVERSITÉ DE NANTES



# Horasis Project

---

## *Cahier des Charges*

---

### *Membres de l'équipe*

Quentin LEVAVASSEUR

Nicolas GAILLARD

Anthony GRIFFON

Valentin BOUCHEVREAU

### *Client / Professeur tuteur*

*Mr. Marcus BARKOWSKY*

*Département Informatique*

*Université de Nantes - Polytech Nantes*

*Avril 2016*

## Sommaire

I - Définition du projet .....	3
I.1 - Contexte .....	3
I.2 - Objectifs .....	3
I.3 - Procédure et critères d'acceptation .....	3
II - Expression du besoin.....	4
II.1 - Besoins fonctionnels .....	4
II.2 - Besoins techniques.....	4
II.3 - Scénario d'utilisation.....	4
II.4 - Liste des fonctionnalités.....	5
III - Contraintes .....	8
III.1 - Contraintes générales .....	8
III.2 - Coûts.....	8
III.3 - Risques .....	8
IV - Rendu .....	9
IV.1 - Maquette de l'application.....	9
IV.2 - Planning.....	10

## I - Définition du projet

### I.1 - Contexte

Lors d'un déplacement, il est parfois difficile de travailler dans des espaces réduits ou publics où le manque de place et le bruit nuisent à la productivité. Plus particulièrement, la lecture d'un document important sur un support physique ou numérique devient vite laborieuse dans ce genre de situations, lorsqu'il s'agit d'y accéder ou encore pour une question de confort.

### I.2 - Objectifs

Notre projet s'inscrit dans un contexte de validation de nos acquis de programmation JAVA au travers d'un projet de fin d'année. L'objectif étant la réalisation d'une application Android (support téléphone) permettant l'affichage d'un document au format PDF (Portable Document Format) d'une manière agréable sur un bureau "physique", le tout en utilisant une paire de Google Cardboard. On parle alors de bureau virtuel, le but étant de s'échapper de la réalité afin de travailler dans de meilleures conditions. La restitution devra se faire avec la meilleure résolution visuelle possible.

Notre première conception s'inscrit dans un domaine d'expérimentation et se veut avant tout flexible. De plus, l'application devra s'adresser à tous types d'utilisateurs souhaitant manipuler et consulter des fichiers PDF d'une manière novatrice et plus dynamique.

### I.3 - Procédure et critères d'acceptation

L'application doit être en accord avec le cahier des charges ainsi que les besoins clients. Elle se doit d'implémenter chaque fonctionnalités, interface ainsi que de respecter les contraintes de délai. De plus un manuel utilisateur expliquant les différentes interfaces et manipulations devra être produit.

Le projet sera validé une première fois à mi-parcours pour s'assurer que nous sommes sur la bonne voie et que nous maîtrisons le SDK Android lié à la Cardboard. Le projet devra être compatible avec les exigences des fonctionnalités Java pour Android ainsi que le besoin client.

## II - Expression du besoin

### II.1 - Besoins fonctionnels

- Application Android ( $\geq 5.0$ )
- Application utilisant la Google Cardboard.
- L'application doit permettre de visualiser un fichier PDF dans un environnement 3D contenant un bureau avec la Cardboard.
- Le bureau est un objet 3D, il s'agit d'une table avec le fichier pdf posé dessus.
- L'application doit permettre de tourner les pages du document PDF.
- L'application doit permettre de zoomer le pdf ou de le dézoomer, 5 distances d'affichage différentes.
- L'affichage s'effectue en mode PORTRAIT, toujours.
- Aucun pré-traitement du document.

Acteur : L'utilisateur qui souhaite consulter ses fichiers PDF sur un bureau virtuel.

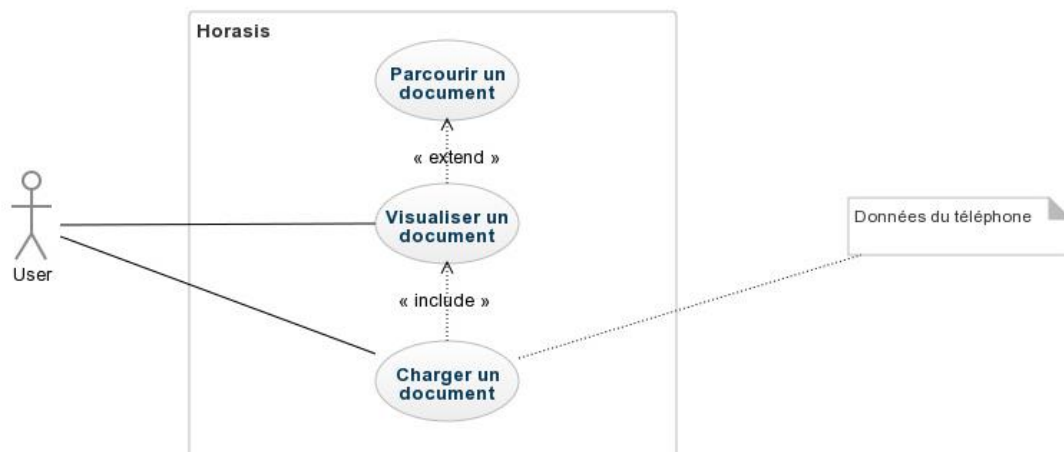


Diagramme de cas d'utilisation lié à l'application

### II.2 - Besoins techniques

- Le téléphone doit adopter une résolution Full HD afin de proposer une lecture agréable.
- L'application peut fonctionner sur une résolution inférieure, nous ne garantissons pas que la lecture sera possible avec une résolution moindre.

### II.3 - Scénario d'utilisation

L'utilisateur est dans un espace confiné, par exemple passer dans un train ou bien dans un avion. L'espace étant restreint, il ne peut étaler ses documents devant lui ou bien déployer son ordinateur. C'est alors qu'il enfle son Google Cardboard et lance l'application. Projeté dans un espace virtuel, l'utilisateur peut alors visualiser et lire ses différents documents sans gêne avec confort sans pour autant être perturbé par le manque réel de place.

## II.4 - Liste des fonctionnalités

N°	Fonctionnalités générales
FG01	Choisir un PDF
FG02-A	Lever la feuille
FG02-B	Baisser la feuille
FG03-A	Retourner à la page précédente
FG03-B	Aller à la page suivante
FG04	Zoomer
FG05	Dézoomer
FG06	Fermer le PDF actuel
FG07	Recentrer le bureau

FG01	Choisir un PDF
<p>Scénario:</p> <p>Lors du lancement de l'application, l'utilisateur voit une liste de ses dossiers et de ses fichiers PDFs. Il peut alors naviguer dans les différents dossiers, revenir sur le dossier parent.</p> <p>L'utilisateur ne voit que les dossiers et les fichiers PDFs, il ne peut alors que choisir l'un des deux, s'il choisit un dossier, il va naviguer dans le dossier, s'il choisit un fichier PDF, alors le programme va passer à l'état: <b>AppStartCardboard</b></p>	

FG02-A	Lever la feuille
<p>Scénario:</p> <p>L'application est à l'état: <b>AppStartCardboard</b>, elle est dans l'environnement 3D, avec la feuille posée sur le bureau.</p> <p>Déclencher le trigger en regardant la feuille sur le bureau va faire lever la feuille. La feuille va alors changer de position et se trouver en face de l'utilisateur.</p>	

## FG02-B Baisser la feuille

## Scénario:

L'application est à l'état: **AppStartCardboard**, elle est dans l'environnement 3D, avec la feuille face à l'utilisateur.

Déclencher le trigger en regardant la feuille va faire baisser la feuille sur le bureau. La feuille va alors changer de position et se trouver sur le bureau.

## FG03-A Tourner la page à gauche

## Scénario:

L'application est à l'état: **AppStartCardboard**, elle est dans l'environnement 3D, avec la feuille en face de l'utilisateur.

Déclencher le trigger en regardant le bouton "Page suivante" va changer la page actuelle par la page suivante s'il existe une page suivante dans le pdf.

## FG03-B Tourner la page à droite

## Scénario:

L'application est à l'état: **AppStartCardboard**, elle est dans l'environnement 3D, avec la feuille en face de l'utilisateur.

Déclencher le trigger en regardant le bouton "Page précédente" va changer la page actuelle par la page précédente s'il existe une page précédente dans le pdf.

## FG04 Zoomer

## Scénario:

L'application est à l'état: **AppStartCardboard**, elle est dans l'environnement 3D, avec la feuille en face de l'utilisateur.

Déclencher le trigger en regardant le bouton "Zoom" va avancer la page vers l'utilisateur si nous ne sommes pas déjà au zoom maximal. Sinon cela ne va rien changer.

**FG05 Dézoomer****Scénario:**

L'application est à l'état: **AppStartCardboard**, elle est dans l'environnement 3D, avec la feuille en face de l'utilisateur.

Déclencher le trigger en regardant le bouton "Dézoom" va reculer la page de l'utilisateur si nous ne sommes pas déjà au zoom minimal. Sinon cela ne va rien changer.

**FG06 Fermer le PDF actuel****Scénario:**

L'application est à l'état: **AppStartCardboard**, elle est dans l'environnement 3D, avec la feuille en face de l'utilisateur.

Appuyer sur le bouton "retour" d'Android va passer l'application en état: **AppStart** et nous allons pouvoir choisir parmi la liste de pdf, un pdf à afficher, comme au lancement de l'application.

**FG07 Recentrer le bureau****Scénario:**

L'application est à l'état: **AppStartCardboard**, elle est dans l'environnement 3D.

Appuyer sur le trigger, en visant une zone vide (qui ne vise ni le bureau, ni la feuille, ni un des boutons), l'application va recentrer le bureau devant l'utilisateur.

### III - Contraintes

#### III.1 - Contraintes générales

Peu de contraintes ont été spécifiées par le client. Cependant, le document affiché se doit d'être lisible pour l'utilisateur. De plus, le document ne doit subir aucun prétraitement afin de ne pas entacher sa mise en page initiale. Enfin, il doit être possible de visualiser les images qui composent le document d'une manière intuitive.

#### III.2 - Coûts

	Anthony G.	Quentin L.	Nicolas G.	Valentin B.
<b>Coûts (en heures)</b>	50	50	50	50

#### III.3 - Risques

##### **L'affichage du PDF s'effectue mal avec la méthode proposée.**

Détection du risque: Affichage de plusieurs types de PDF. On vérifie s'ils sont bien affichés.

Solution 1: Changer de librairie utiliser et tenter avec une seconde librairie.

Solution 2: Faire la conversion avant par un programme tiers, stocker les images sur le téléphone et les utiliser pour l'affichage des pages.

Solution 3: Passer par un service en ligne.

##### **L'interface est trop sensible, les boutons s'activent tout seul.**

Détection du risque: Pendant les tests de l'application.

Solution: Demander à cliquer plusieurs fois sur un bouton pour effectuer l'action.

##### **La conversion du PDF n'est pas faisable**

Détection du risque : Affichage de plusieurs documents

Solution 1: Changer de librairie

Solution 2: Passer par un service de conversion externe

##### **Non maîtrise du SDK Cardboard et OpenGL**

Détection du risque: Échec de modélisation d'un bureau

Solution 1: Représentation en 2D que l'on divisera sur deux parties de l'écran pour simuler un environnement virtuel.

##### **Impossibilité de tester l'application**

Détection du risque : l'émulateur d'Android Studio n'est pas capable d'afficher des environnements virtuels.

Solution 1: Utiliser un émulateur alternatif (type GenyMotion)

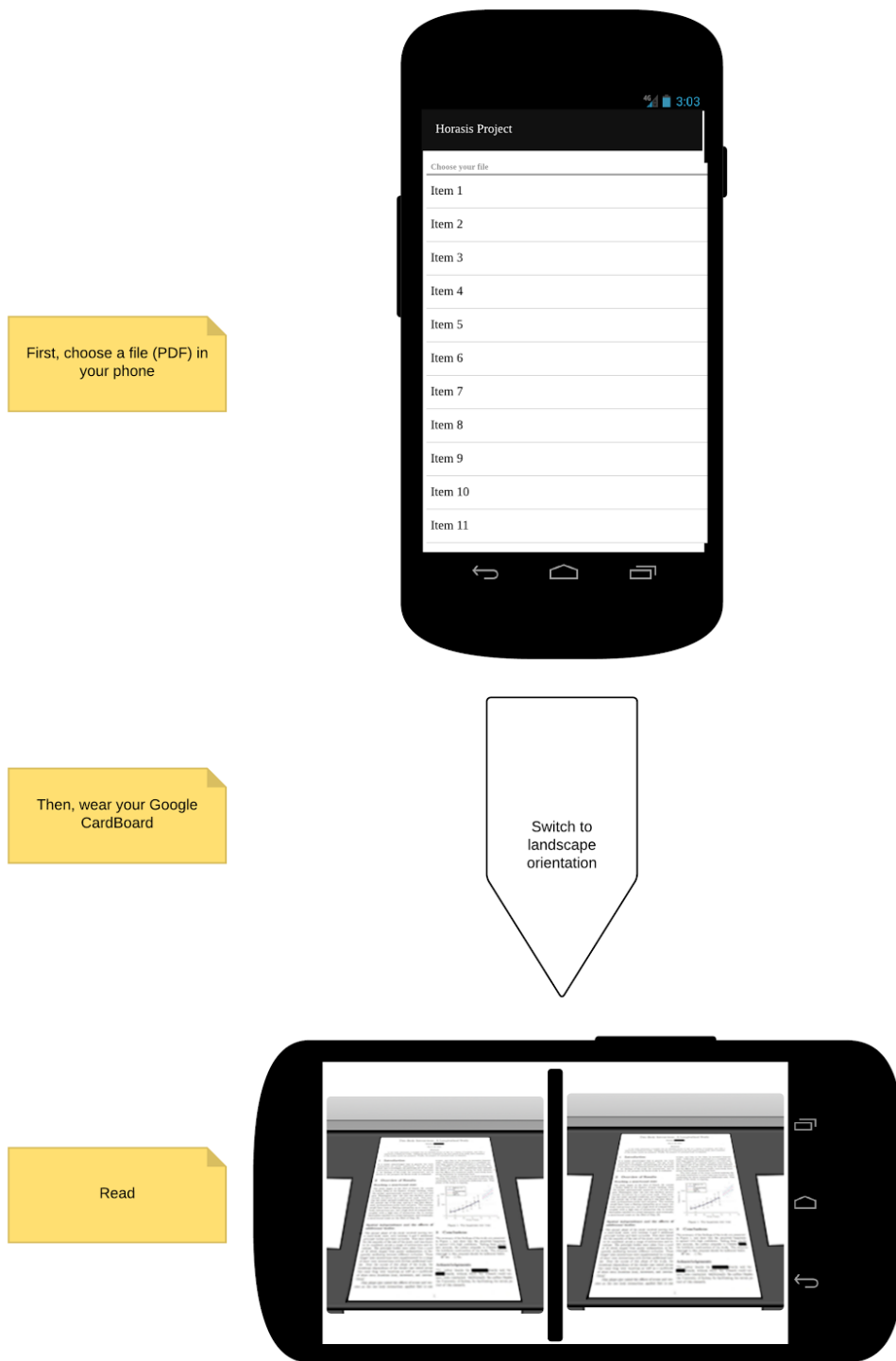
Solution 2: Demander un prêt de téléphone Android

Solution 3: Revoir la base du projet avec le client



## IV - Rendu

### IV.1 - Maquette de l'application



*Maquette de l'application faite à partir du logiciel web [Lucidchart](#)*

## IV.2 - Planning

Date de livraison	Rendu
Vendredi 15 avril 2016	Cahier des Charges
Vendredi 13 mai 2016	Conception Générale (UML)
Vendredi 3 juin 2016	Livraison de l'application et bilan

Une réunion se tiendra le mercredi 11 mai 2016 entre l'équipe du projet et le client pour faire état de l'avancée de l'application ainsi que de sa conception générale.