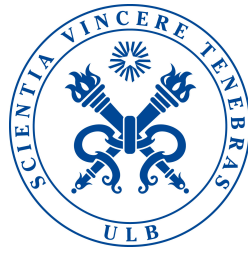


UNIVERSITÉ LIBRE DE BRUXELLES



INFO-F209: PROJET D'INFORMATIQUE II  
WIZARD POKER

---

# Software Requirement Document

---

*Auteurs:*

Yasin ARSLAN  
Youcef BOUHARAOUA  
Jalal NAH  
Sacha MEDAER

Youssef SITIL  
Miguel TEROL ESPINO  
Jacky TRINH

*Titulaires:*

Joël GOSSENS  
Christian HERNALSTEEN

*Assistants:*

Keno MERCKX  
François GÉRARD  
Jacopo DE STEFANI

December 14, 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	But du projet . . . . .	2
1.2	Glossaire . . . . .	2
1.3	Historique du document . . . . .	2
<b>2</b>	<b>Besoins de l'utilisateur</b>	<b>3</b>
2.1	Exigences fonctionnelles . . . . .	4
2.1.1	Gestion des utilisateurs . . . . .	4
2.1.2	Gestion de la collection . . . . .	4
2.1.3	Duel . . . . .	4
2.1.4	Gestion des défis . . . . .	5
2.2	Exigences non fonctionnelles . . . . .	5
2.3	Exigences du domaine . . . . .	5
<b>3</b>	<b>Besoins du système</b>	<b>6</b>
3.1	Exigences fonctionnelles . . . . .	6
3.1.1	Use case view . . . . .	6
3.2	Exigences non fonctionnelles . . . . .	6
3.3	Design et fonctionnement du système . . . . .	6

# Chapter 1

## Introduction

1.1 But du projet

1.2 Glossaire

1.3 Historique du document

## Chapter 2

# Besoins de l'utilisateur

Le *Wizard-Poker* est un jeu de cartes *fantastique* dans lequel des joueurs s'affrontent dans des **duels 1-vs-1**. Chaque joueur possède une collection de cartes à jouer lui permettant de former des paquets appelés **decks**.

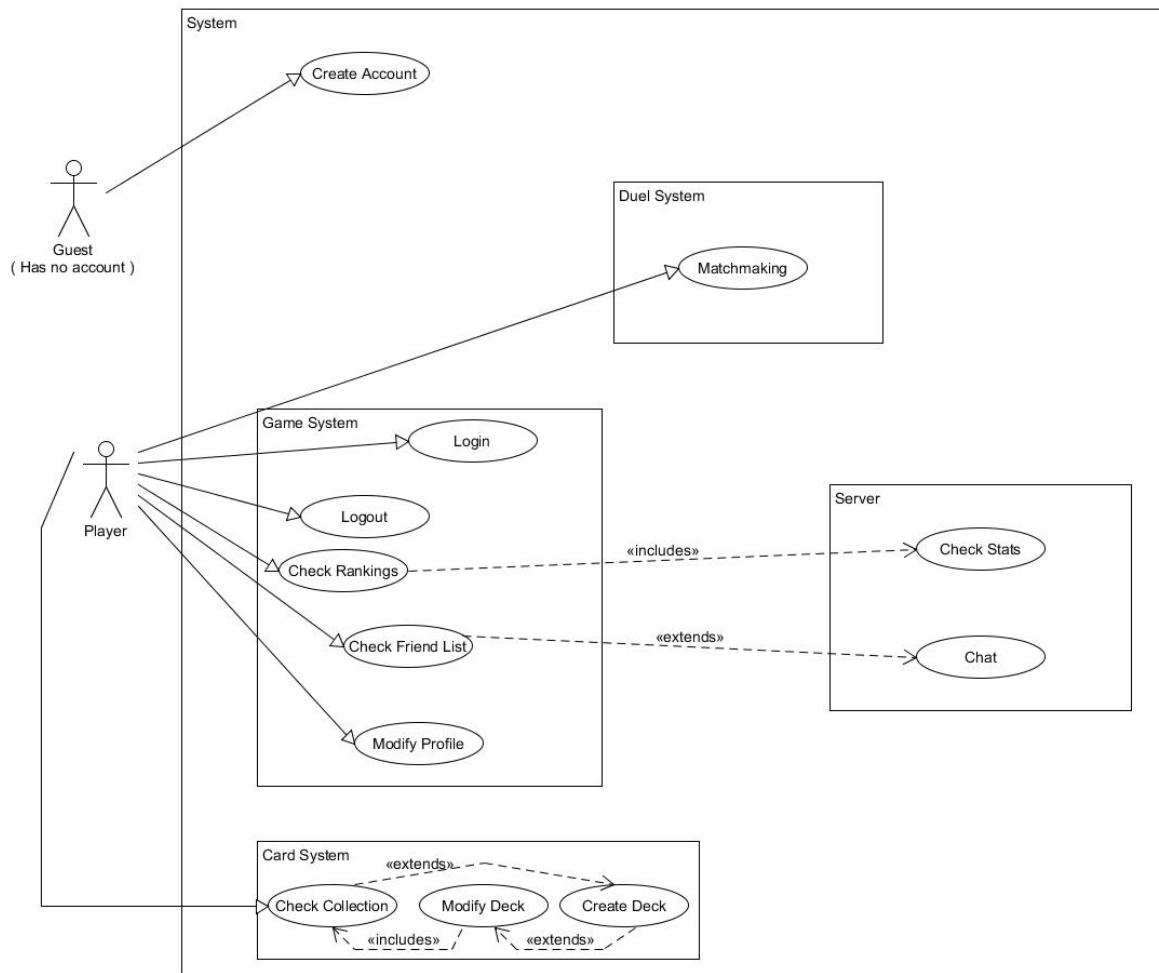


Figure 2.1: Système du jeu.

## 2.1 Exigences fonctionnelles

### 2.1.1 Gestion des utilisateurs

**Cas général :** L'*utilisateur* doit pouvoir s'inscrire (s'enregistrer) sur le *serveur* avec un nom d'*utilisateur* unique et un mot de passe.

**Pré Condition :** L'*utilisateur* n'est pas déjà connecté

**Post Condition :** L'*utilisateur* aura fait une demande d'inscription.

**Cas Exceptionnel :** L'inscription échouera et l'*utilisateur* sera invité à recommencer si le nom de compte est déjà enregistré sur le *serveur*.

#### Connection

**Cas général :** L'*utilisateur* doit pouvoir se connecter en s'authentifiant auprès du *serveur* avec son nom d'*utilisateur* et son mot de passe.

**Pré Condition :** L'*utilisateur* n'est pas déjà connecté.

**Post Condition :** L'*utilisateur* aura fait une demande de connexion.

**Cas Exceptionnel :** L'authentification échouera et l'*utilisateur* sera invité à recommencer si le mot de passe ne correspond pas au nom d'*utilisateur*, ou cet *utilisateur* n'existe sur le *serveur*.

### 2.1.2 Gestion de la collection

#### Decks

### 2.1.3 Duel

#### Jouer un tour

**Cas général :** L'*utilisateur* doit pouvoir jouer un tour lors d'un match, c'est à dire placer ses *cartes* et/ou effectuer des actions.

**Pré Condition :** L'*utilisateur* doit être dans un duel.

**Post Condition :** L'*utilisateur* aura joué ses actions.

**Cas Exceptionnel :** La perte de connexion avec un *utilisateur* maintenue, durant un certain timeout, [entraînera son remplacement pour ce duel par une IA fournie par le système. Le match continuera sans changement.]

### Déclarer forfait

**Cas général :** L'*utilisateur* peut déclarer forfait durant un duel (il sera considéré comme perdant).

**Pré Condition :** L'*utilisateur* doit être dans un duel.

**Post Condition :** L'*utilisateur* sera déclaré comme perdant.

### 2.1.4 Gestion des défis

#### Matchmaking

**Cas général :** L'*utilisateur* peut initialiser un recherche avec le Matchmaking.

**Pré Condition :** L'*utilisateur* ainsi que l'autre utilisateur doivent être connectés et ne pas être dans un duel.

**Post Condition :** L'*utilisateur* l'utilisateur adverse aussi en Matchmaking se connectera.

## 2.2 Exigences non fonctionnelles

## 2.3 Exigences du domaine

Le domaine de ce projet étant celui des jeux-video, les exigences du domaine restent pas suffisamment netes comme pour les citer explicitement comme contraintes absolues.

## Chapter 3

# Besoins du système

### 3.1 Exigences fonctionnelles

#### 3.1.1 Use case view

Dans notre diagramme use case, nous avons repris dans les grandes lignes toutes les manières d'utiliser le Wizard Poker. Aucun détail n'est illustré, seulement une idée générale des différentes exécutions. Il existe, pour cette application, 2 types d'utilisateurs :

Le **guest**, qui sera un nouvel utilisateur découvrant le jeu, celui-ci n'aura accès à aucun contenu si ce n'est la **création d'un compte**. Nous l'avons privé de toute autre action, même l'accès au classement.

Le **Player** a déjà un compte et peut donc **se connecter**. Une fois connecté, il y a donc divers plate-formes. Il peut **modifier son profil**, **consulter sa liste d'amis**, **lancer une partie** et également **se déconnecter**. Le système de **chat** est accessible depuis la liste d'amis du joueur et **l'analyse de sa dernière partie** se trouve via le classement. Ceux-ci seront principalement des actions qui apparaissent dans toutes les applications de ce type. Les actions propres au jeu Wizard Poker sont liés aux cartes. Le Player peut donc **voir sa collection**, **modifier son deck** et également **créer un nouveau deck**. Lors de la modification d'un deck, il y a des ajouts de cartes, celles-ci doivent être disponibles dans la collection du joueur, nous avons donc décidé de rendre la collection accessible depuis la modification d'un deck.

### 3.2 Exigences non fonctionnelles

### 3.3 Design et fonctionnement du système