

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبو بكر بلقايد-تلمسان

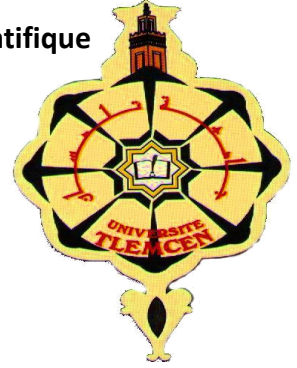
Université AbouBekr BELKAID

كلية العلوم – تيجاني هدام

Faculté des Sciences – Tidjani HAddam

قسم الإعلام الآلي

Département d'informatique



Rapport Mini-Projet - L3

Pour l'Obtention du Diplôme : LICENCE

Intitulé du mini projet

Système de Billetterie en ligne

Examiné le : 12/06/2022

Devant les Jurys :

Examineur : Mr. MERZOUG.

Examineur : Mr. FEKHAR.

Encadrant : Mr. BENZIANE

Réalisé par :

. AMMAR Abderrahmane.

. ABDELLAOUI Sid Ahmed.

. HAMIDI Mohamed.

. SENOUCI Mohamed Amine.

Année Universitaire: 2021-2022

Sommaire

1. Introduction :

- + Introduction générale et objectif cible.**
- + Etudes de faisabilités**
- + outils utilise dans :**
 - * phase de spécification.**
 - * phase de conception.**
 - * phase de développement.**

2. Spécification et analyse des besoins :

- + analyse des besoins**
- + spécification**
- + exigence fonctionnelle de l'application**
- + exigence non fonctionnelle de l'application**

3. Conception :

- + diagramme de case d'utilisation.**
- + diagramme de classe.**
- + diagramme de séquence système.**
- + modèle logique des données (MLD).**

1 - Introduction générale

L'informatique représente la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité ces dernières décennies. En effet, loin d'être un éphémère phénomène de mode, ou une tendance passagère, l'informatique vient nous apporter de multiples confort à notre mode de vie. Aucun domaine n'est resté étranger à cette stratégie qui offre tant de services aussi bien pour l'enseignement ou l'administration que pour le personnel et c'est dans ce cadre d'idées que s'inscrit notre mini-projet.

L'objectif ciblé dans notre mini-projet est la spécification, la conception, et l'implémentation de systèmes de billetterie qui vous aident à gérer la vente du billet de plusieurs types d'événement sportive sous forme d'un code QR. Notre travail consiste à informatiser modélise et organise cette idée et le fusionner avec la nouvelle technique code QR pour simplifier cette action et la transformée d'un point de gestion classique (tu dois déplacer et faire la chaîne pour acheter un billet et plusieurs conditions inappropriées) à une plateforme en ligne offre :

- Commodité pour les ventes et les paiements.
- Gestion des tickets et des stocks.
- Emission et contrôle automatiques de différents types de tickets et de lots.
- Disponibilité des sièges mise à jour en temps réel.
- Billets personnalisés avec une variété de supports d'accès innovants : billet électronique, code QR dynamique, etc.
- Un suivi en temps réel des statistiques.

Pour réaliser cette idée, on utilise ensemble d'activités successives, organisées en vue de la production d'un logiciel nommé : "les activités ou Processus du développement logiciel".

● Etude de faisabilité : (problème et solution)

C'est une longue phase, on discute l'idée et ça utilise son marché, sa population et plusieurs facteurs. Dans cette phase on s'intéresse de répondre à ces questions :

- + Pourquoi développer le logiciel ?
- + Comment procéder pour faire ce développement ?
- + Y a-t-il un marché pour le logiciel ?

D'après le dernier match de Algérie Vs Cameroun et tous les événements qui sont passés à cause de l'organisation catastrophique de processus de vente du billet et ces conséquences (violence et émeute) on a trouvé que c'est une très bonne idée pour l'informatiser et diffuser dans le marché de sport surtout que c'est une nouvelle idée n'existe pas dans le marché, n'a pas des concurrents, et elle lutte spéculation sur les prix dans le marché noir. L'application devrait tout d'abord être extrêmement fiable valide ergonomique, et son utilisation quotidienne

ne devra pas laisser place à l'éventuel point faible. L'objectif principal est la gestion des achats des billets sous forme d'un code QR, mais en autres, l'application devrait notamment :

- Permettre d'inscrire des nouveaux utilisateurs.
- Permettre de faire la gestion des matchs (ajout, suppression ...).
- Permettre de faire le paiement en ligne (ccp ou carte visa...).
- Permettre de faire une étude statistique sur le taux de vente des billets.

● Spécification : (quoi ?)

Dans cette phase on s'intéresse :

- + d'analyse des besoins.
- + Établir une description claire de ce que doit faire le logiciel (fonctionnalités détaillées, exigences de qualité, interface...)
- + Définir les exigences fonctionnelles et non fonctionnelles

● Conception : (comment ?)

Dans cette phase on s'intéresse d'élaborer une solution concrète réalisant la spécification par la :





- + Conception globale : description architecturale en composants (avec interface et fonctionnalités)
- + Conception détaillé : Détailler le fonctionnement des composants

On utilise dans cette étape la modélisation par le langage UML (Unified Modeling Language) c'est un langage graphique pour visualiser, spécifier, construire et documenter un logiciel

Le logiciel utilisé dans cette étape est **modelio** . 

● Programmation :

La dernière étape de notre processus du développement logiciel c'est l'implantation de la solution en code, les outils utilise dans cette étape :

- + l'environnement de développement : **Visual studio code**. 
- + Framework : **Google Flutter**.  Flutter
- + langage de programmation : **Dart**.  Dart
- + Backend : **Google Firebase**.  Firebase

2 – Spécification et Analyse des besoins

● Analyse des besoins :

Le marché (clients) besoin d'une application qui gère l'achat en ligne avec une performance et disponibilité et utilisation facile.

● Spécification :

A l'aide de ces besoins on a établi une description claire de ce que doit faire le logiciel (Côté de fonctionnement avec l'utilisateur) :

- * Le nom de l'application est "INSTICKET" signifie : INST (instantanée) TICKET (billet).
- * Cliquez sur l'Icon de l'application va nous afficher une interface d'authentification contient deux boutons :
 - Identifier : l'utilisateur écrit son nom, adresse mail pour l'identifier à l'application.
 - S'inscrire : pour les nouveaux utilisateurs permettent d'inscription par l'écriture des informations nécessaires.
- * Après l'application va nous afficher une autre page "Accueil" contient les matchs disponibles et un champ de recherche, le clic sur un match va nous afficher une interface contient toutes les informations sur le match et deux boutons pour l'achat de billet de prix différent.
- * L'interface "MES BILLETS" contient les billets achetés et aussi historique des billets expire, chaque billet a son propre information et code QR, grâce à ce code l'utilisateur accède au stade par scanner son QR code une fois le code est scanné, le billet va être expiré et coloré en rouge, et l'utilisateur ne peut pas le réutiliser.
- * On a une autre interface nommée "Profil" contient l'ID de l'utilisateur (c'est son adresse mail) et un champ pour changer le nom, bouton pour effectuer ce changement et autre bouton "Tableau de bord d'administrateur" qui apparaît si l'utilisateur est admin avec cette fonctionnalité on peut ajouter un nouveau match, ajouter une nouvelle équipe, scanner le code QR, contient aussi les statistiques sur la vente des billets.
- * On a aussi un "Menu Bar" (pop-up fenêtre) :
 - A propos.
 - Contactez – nous.
 - Thème sombre (change thème).

● Exigences fonctionnelles :

- + Permettre d'inscrire des nouveaux utilisateurs.
- + Permettre de consulter les matchs disponibles.
- + Voir des informations détaillées sur le match.
- + Voir les informations sur les billets achetés.
- + Télécharger code QR.
- + Voir l'historique d'achat des billets expirés et des billets non expirés.
- + Changer le nom.
- + Changer le thème.

● Exigences non fonctionnelles :

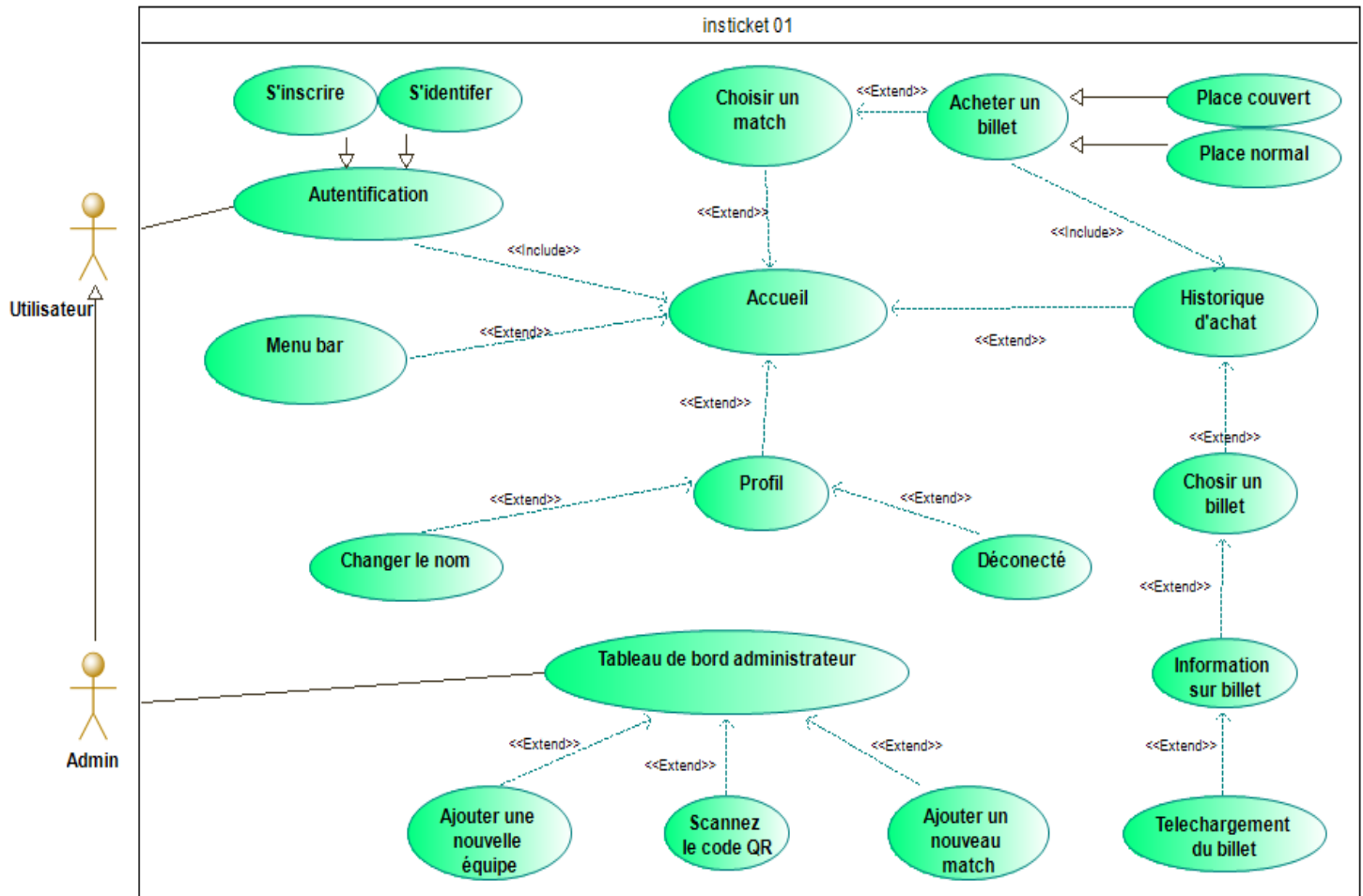
- + Validité : réponse aux besoins des utilisateurs (achat, téléchargement code QR ...)
- + Facilité d'utilisation(ergonomie) : l'interface est simple et interactive.
- + Performance : l'app est léger (11Mo), et temps de réponse rapide, ne consomme pas beaucoup de ressource.
- + Fiabilité : l'application fiable exemple si l'utilisateur chercher à un nom d'équipe et le nom est incorrect ou introuvable l'app ne casse pas de fonctionnement (ne bugue pas).
- + Sécurité : intégrité des données et protection des accès.
- + Maintenabilité : facilité à corriger ou transformer le logiciel.
- + Extensible : on peut développer plusieurs versions évaluer.

● Modèles de cycle de vie de l'application :

- + On a choisi le "Modèle incrémental" consiste à développer pas à pas :
 - Sous-ensemble minimal et fonctionnel du système.
 - Ajouts d'incréments jusqu'à la fin du processus.
 - les intégrations sont progressives.
 - possibilité de livraisons et de mises en service après chaque incrément.

3 – Conception

- **Diagramme de Cas d'utilisation :**

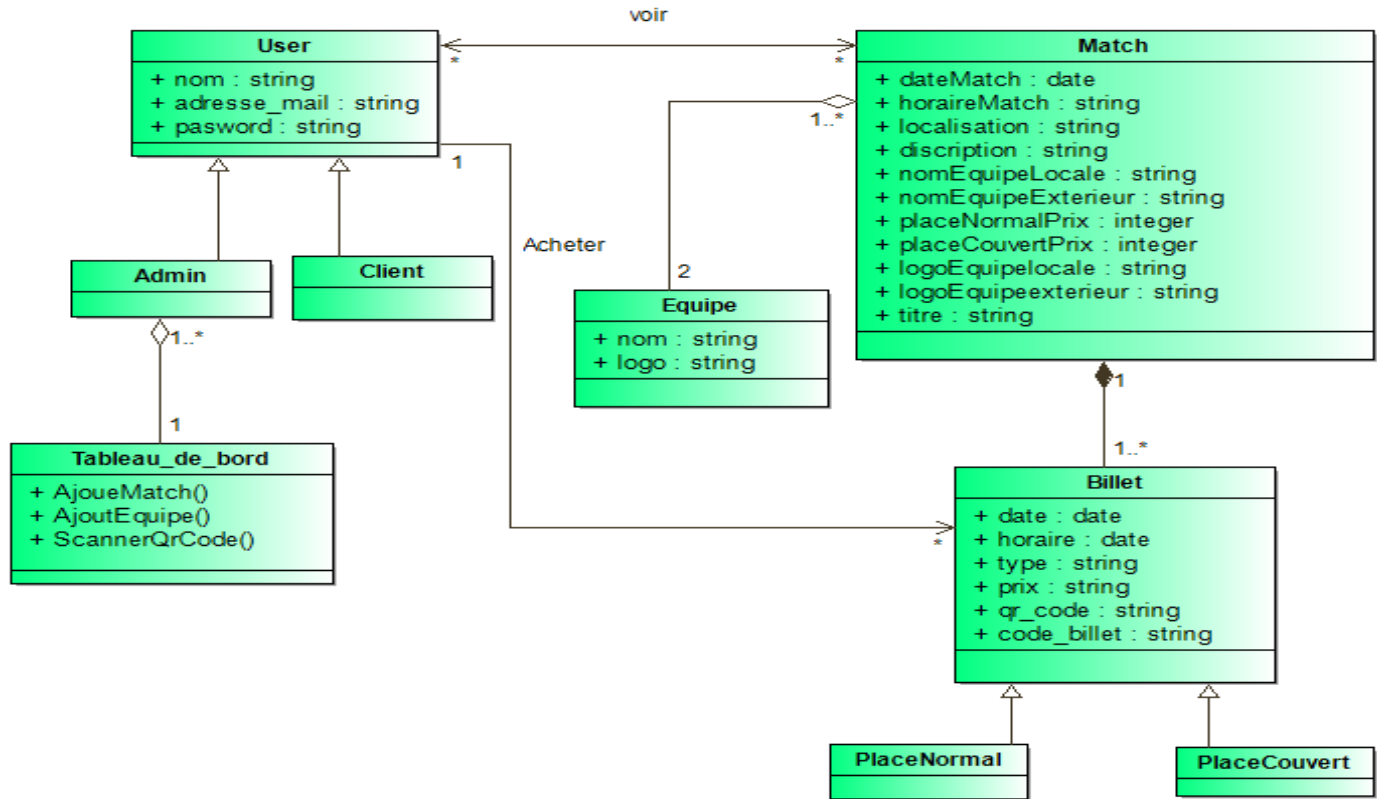


Description : on a l'acteur utilisateur déclenche C.U "authentification", ce C.U inclut le C.U de base "Accueil", on peut aller vers toutes les autre C.U liée direct (Historique d'achat, profil, choisir un match ...) ou indirect (acheter un billet, tableau de bord administrateur, information sur billet ...) Depuis ce C.U "accueil", on a d'autre acteur qui hérite toutes c'est fonctionnalité depuis l'acteur utilisateur et a en plus cas d'utilisation "tableau de bord administrateur" qui sert à gérer l'application (ajoute suppression des matchs, scannez QR code, ajout des équipes).

* Quelques notions :

- + Système : nom d'application.
- + Acteur : déclencher le cas d'utilisation.
- + Cas d'utilisation (C.U) : fonctionnalité de système.
- + Relation : Héritage, inclusion, extension, association simple.

● Diagramme de Classe :



Description :

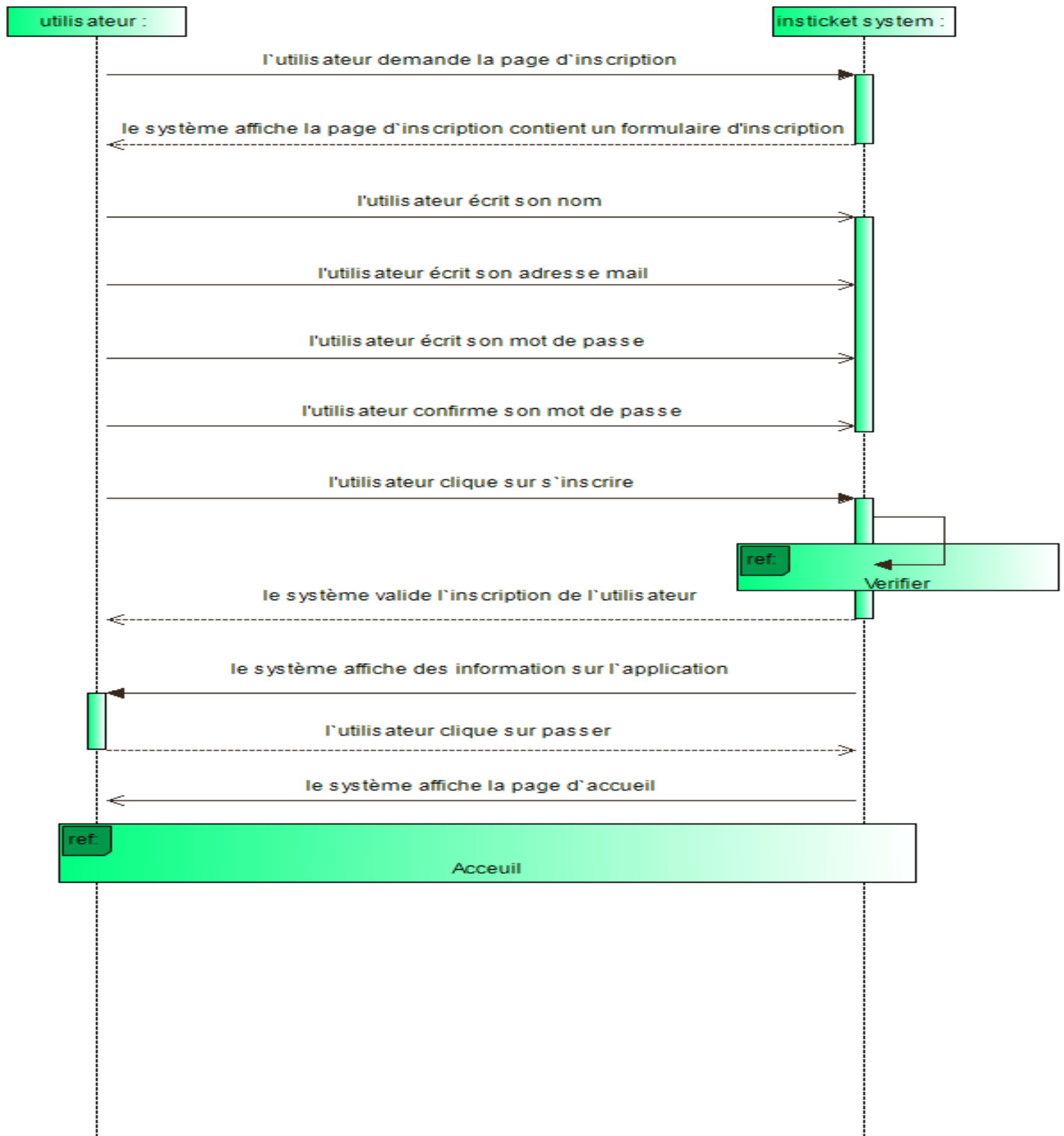
- * la classe user en relation de type association simple avec classe match, user voit plusieurs matchs, le match visionner par plusieurs utilisateurs.
- * la classe user aussi à une relation d'association simple avec la classe billet user peut acheter plusieurs billets.
- * La classe billet en relation de type composition avec la classe match (le match possède plusieurs billets si on supprime le match les billets seront supprimé aussi).
- * La classe équipe en relation avec la classe match de type agrégation (le match compose de deux équipes la différence entre la composition et l'agrégation et la dernière si on supprime le match les équipes ne seront pas supprimé juste disparition de la relation).
- * les deux classe admin et client héritent de classe user qui est plus généralisé et les deux classe (admin, client) sont plus spécialisé.
- * la classe tableau de bord en relation avec la classe admin de type agrégation, cette classe contient les méthodes de gestion des matchs des équipes ... etc.
- * la classe billet est une classe généralise, les deux classe place normal et place couvert héritent de la classe billet pour plus de spécialisation dans le type et le prix de billet.

- Diagrammes de séquence système :

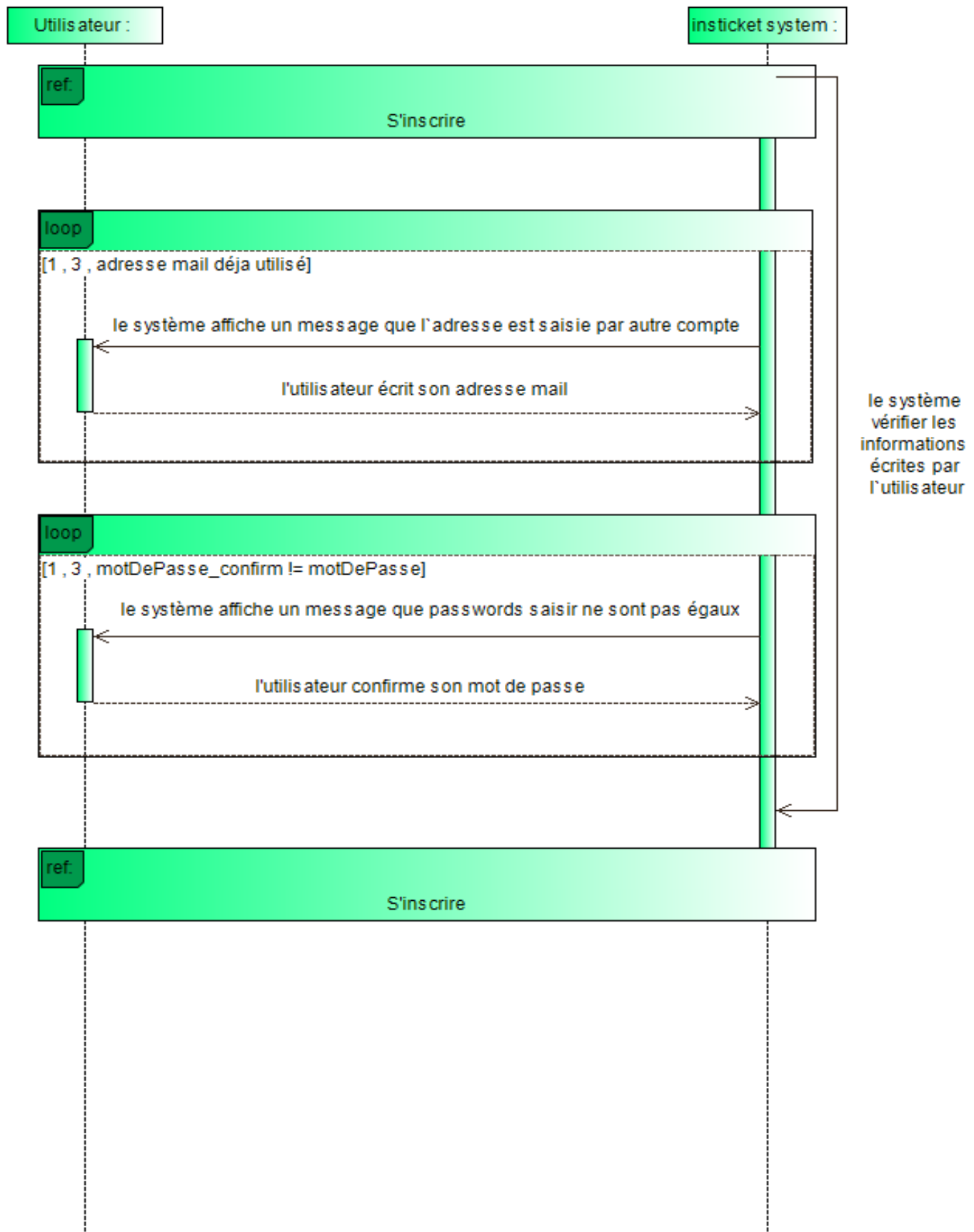
- 1 – C.U “s’inscrire” :

- 1.2 - diagramme de séquence système de C.U « s'inscrire »

- Scénario nominal :



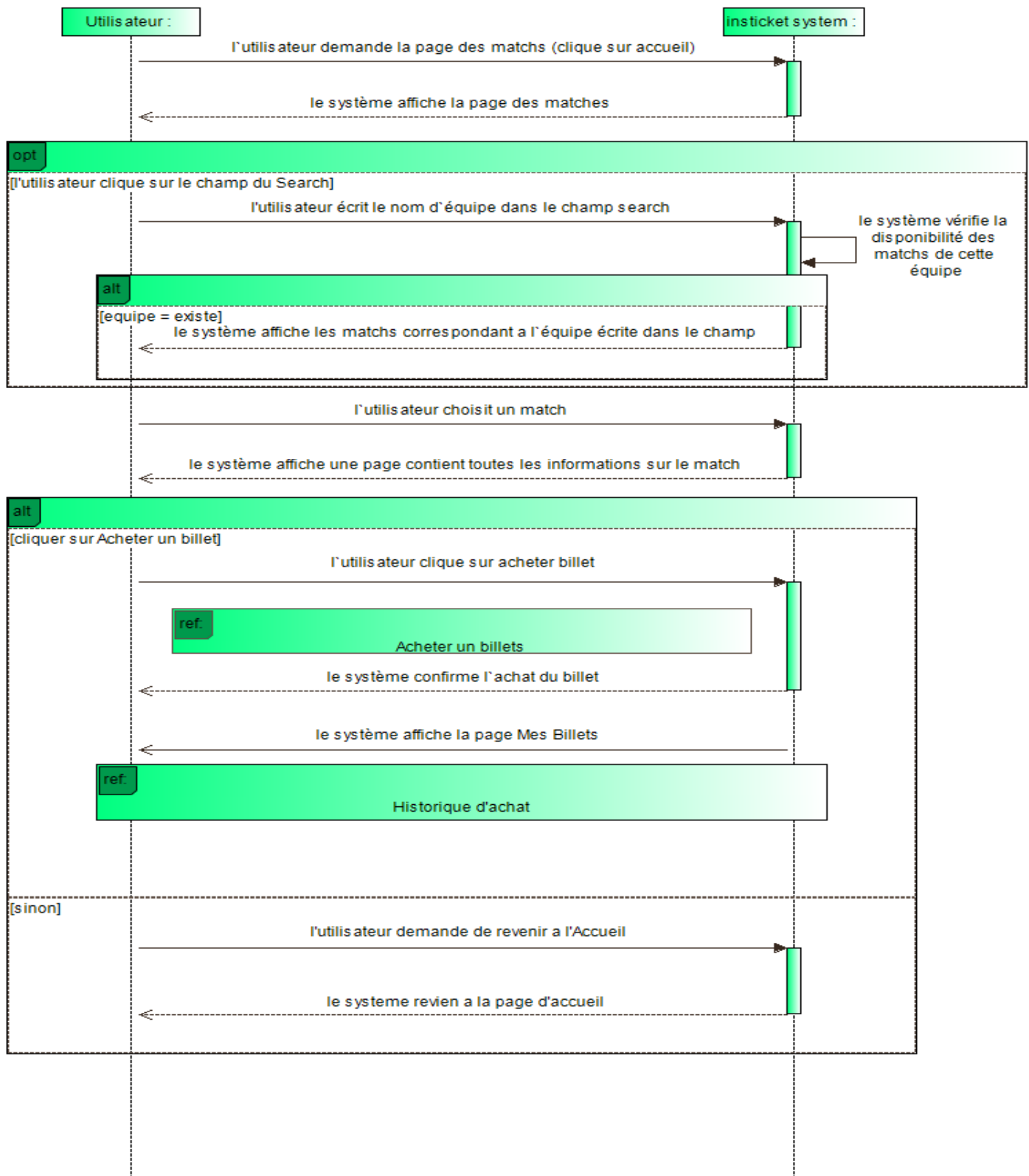
• Scénario alternatif :



2 – C.U “Choisir un match” :

2.2 - Diagramme de séquence système de C.U « Choisir un match »

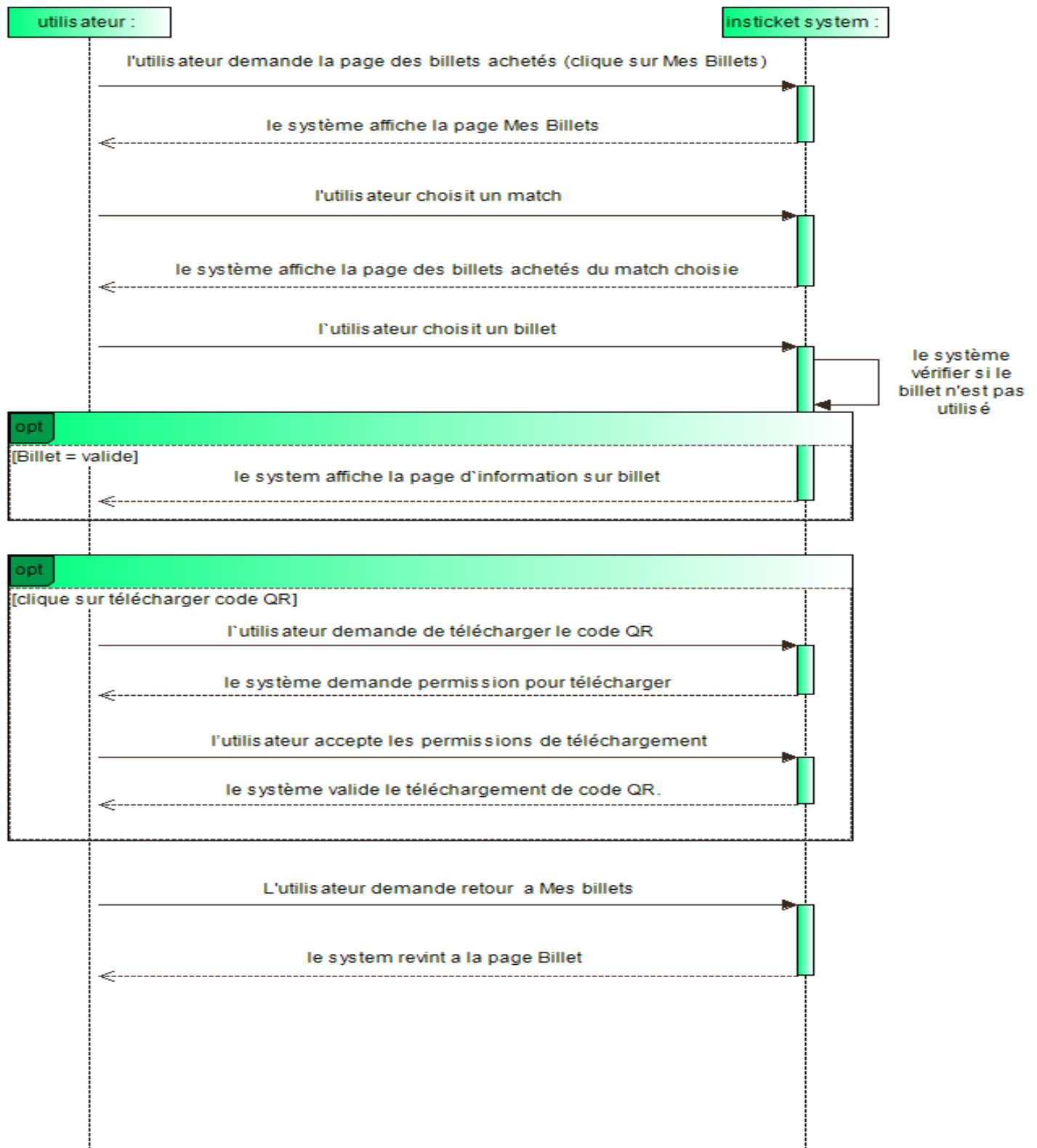
- Scénario nominal et alternatif :



3 – C.U “Historique d’achat” :

3.2 - Diagramme de séquence système de C.U « Historique d’achat »

- Scénario nominal et alternatif :



● Modèle Logique des Données :

+ Description :

Dans notre base de données on a quatre collections des tables sont : Game, team, ticket, user

Les relations entre les collections (MCD) :

Match (1) --- (nom de relation : compose) --- (*) Team

Match (1) --- (nom de relation : posséder) --- (*) Ticket

User (*) --- (nom de relation : consulter) --- (*) Match

Ticket (*) --- (nom de relation : acheter) --- (1) User

La table contient plusieurs attribue deux sont important :

PrimaryKey: attribue souligner.

ForeignKey: **attribue_fini_par #**.

+ MLD :

USER (uid, mail, motpass, nom, est_admin).

MATCH (match_id, titre, description, stade, date, nombre_place_couvert, courant_place_couvert, prix_place_couvert, nombre_place_normal, courant_place_normal, prix_place_normal, **nom_equipe_locale#**, **equipe_locale_image_url#**, **nom_equipe_exterieur#**, **equipe_exterieur_image_url#**).

TICKET (code, Qr_code, est_valide, type, prix, **uid#**, **match_id#**).

TEAM (nom, img_url).

La relation n-aire Match (*) --- (nom de relation : consulter) --- (*) User en MLD :

CONSULTER (**uid#**, match_id#).

référence :

Site web d'application : insticket.netlify.app.