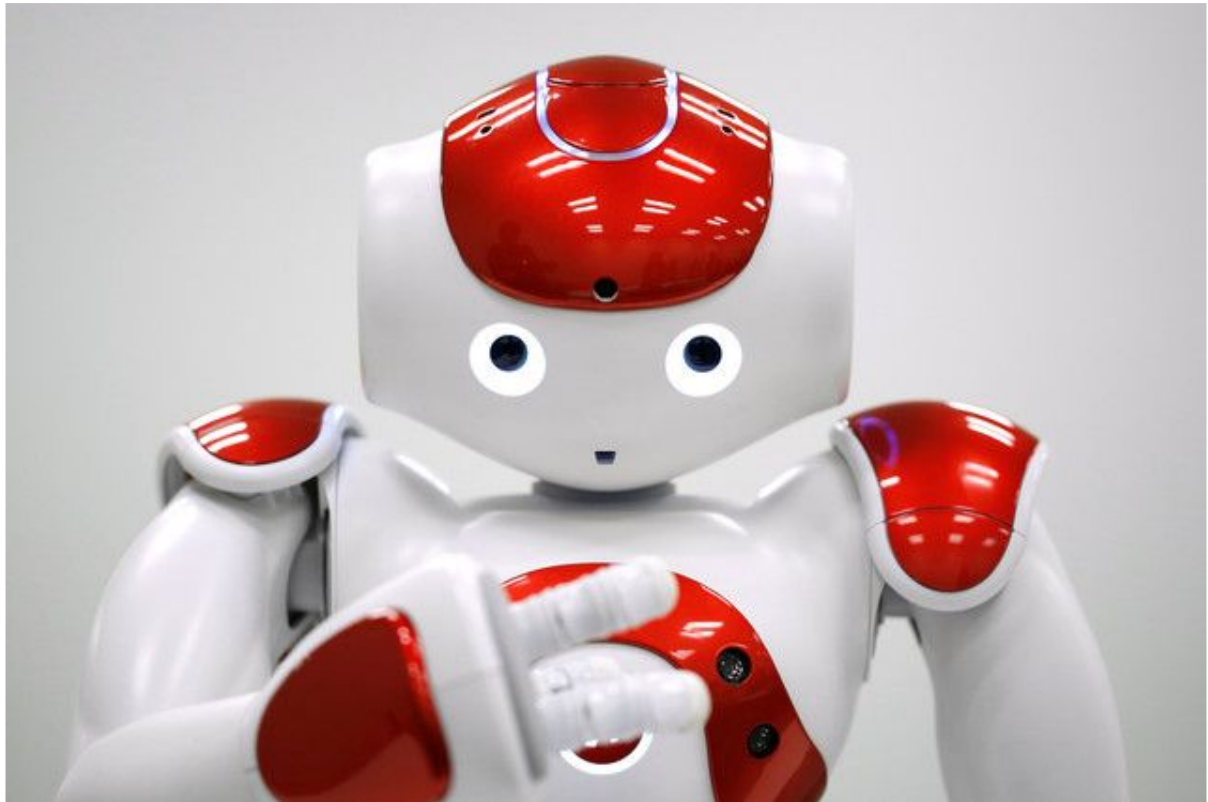


Projet PANC



Dossier d'analyse



Étudiants: ABITBOL Naomi
BOUNMA Adeline
CIOBANU Patricia

Professeur: Camille Salinesi
Licence 3 MIAGE Classique

Table des matières

Introduction	2
Qualité attendue	4
Système: vue fonctionnelle	5
Système: vue statique	6
Système: vue dynamique	7
Diagramme de séquence: S'authentifier	7
Diagramme de séquence: Création compte	8
Diagramme de séquence: Consultation score	9
Diagramme de séquence: S'exercer	10
Choix d'implémentation	11
Conduite du projet	14
Bibliographie	15

Introduction

De nos jours, l'intégration des robots dans la vie quotidienne transforme le secteur industriel. Les robots sont devenus une partie importante de l'économie mondiale. Ils aident à améliorer l'efficacité du travail et à réduire le coût de la main d'oeuvre.

Cependant, dans le contexte pédagogique, le potentiel des robots n'est pas pleinement exploité. Récemment, des entreprises aux Etats-Unis, en Europe et en Corée ont commencé à concevoir des robots qui ont un impact positif sur les enfants handicapés avec des troubles comme celui du spectre autistique (TSA) ou le Trouble Déficitaire de l'Attention avec ou sans Hyperactivité (TDA/H).

Le robot NAO constitue l'un des exemples de succès dans cette industrie. Il a été développé pour venir à la rencontre des besoins émotionnels des enfants qui ont des nécessités particulières en ce qui concerne l'environnement d'apprentissage. Il a été développé par une compagnie française "*Aldebaran Robotics*". En 2005, le fondateur, Bruno Maisonier, a créé cette compagnie. C'est la première entreprise française qui se consacre à la robotique humanoïde. Aujourd'hui, l'entreprise compte approximativement 180 employés.

NAO est le robot humanoïde le plus utilisé dans le monde pour des raisons pédagogiques. Entièrement interactif, divertissant et constamment évolutif, NAO est une plateforme sûre pour enseigner les étudiants aux niveaux différents.

Par suite, NAO qui ressemble plus à un jouet qu'à une machine, contribue positivement au désir de l'enfant de se concentrer et d'interagir avec ses camarades. Son succès est constitué de la réduction des gestes et des expressions faciales qui normalement ont lieu pendant une conversation homme-homme. Cette simplification aide les enfants à devenir plus à l'aise lors d'une conversation.



En effet nous mettrons l'accent sur la communication afin de rendre le dialogue simple et accessible: le robot pourrait donc être un égal de l'enfant, voire un ami, tout en jouant le rôle de professeur.

Dans ce cadre nous avons eu l'idée de créer une application web qui initie les enfants à la programmation en leur faisant exécuter des commandes au robot. Le but de l'application est qu'un élève suive pas à pas des étapes afin d'aboutir à un "projet" i.e. un résultat final visible directement sur le robot.

Notre projet a pour objectif d'utiliser le robot NAO dans un cadre à la fois pédagogique et à la fois récréatif, pour un public essentiellement jeune (6-12 ans). Il serait donc intéressant d'intégrer cette application dans des classes ou tout simplement dans la vie quotidienne d'un enfant.

L'intérêt vient par exemple de la patience illimitée du robot, ce qui permettra à l'enfant de s'exercer autant de fois que voulues sur un même exercice ou encore de sentir une présence constante.

Il existe déjà des applications visant à apprendre la programmation aux jeunes enfants telles que Scratch ou encore Tynker, cependant notre application sera propre au robot NAO et l'utilisateur pourra voir un résultat en direct de son travail. Le robot fournira une sorte de vérification en temps réel du travail afin de guider au mieux l'utilisateur.

Ce projet s'inscrit dans un contexte social dans la mesure où il touche directement à l'éducation et la pédagogie. Le message que l'on désire transmettre est la sensibilisation de la société à l'égard de l'importance que l'informatique a de nos jours et donc la nécessité d'intégrer cet élément le plus tôt possible.

Nous pourrions imaginer que l'application soit vendue à des écoles ou encore à des particuliers car elle répond aux besoins des utilisateurs (enfants 6-12 ans) et se penche sur les enjeux actuels. En effet, la programmation devient un outil important dans la mesure où l'exercice apporte des stratégies pour résoudre des problèmes, désigner des projets ou encore communiquer des idées.

L'aspect transverse du projet par rapport à la formation MAGE est simple. Cela regroupera les matières de méthodes d'ingénierie des systèmes (ISI1), les fondements de l'algorithmique, la programmation orientée objet (INF1, INF2) ainsi que les techniques de tests et validation de logiciels (ISI6). Cela concernera les techniques de communication (TC), l'anglais (AN) et l'atelier d'outils de développement (PRO1) dans une moindre mesure.

Qualité attendue

Nous attendons de notre système que son implémentation soit le plus efficace et le plus simple possible. Pour ceci nous veillons à ce que les besoins soient bien déterminés et se concentrer sur les principales fonctionnalités du système (et en rajouter par la suite).

Après avoir déterminé les besoins nous avons choisi le langage dans lequel nous voulons développer notre application, html, php, css, javascript, mysql et python pour le serveur. Ce sont des langages accessibles qui permettent une implémentation performante.

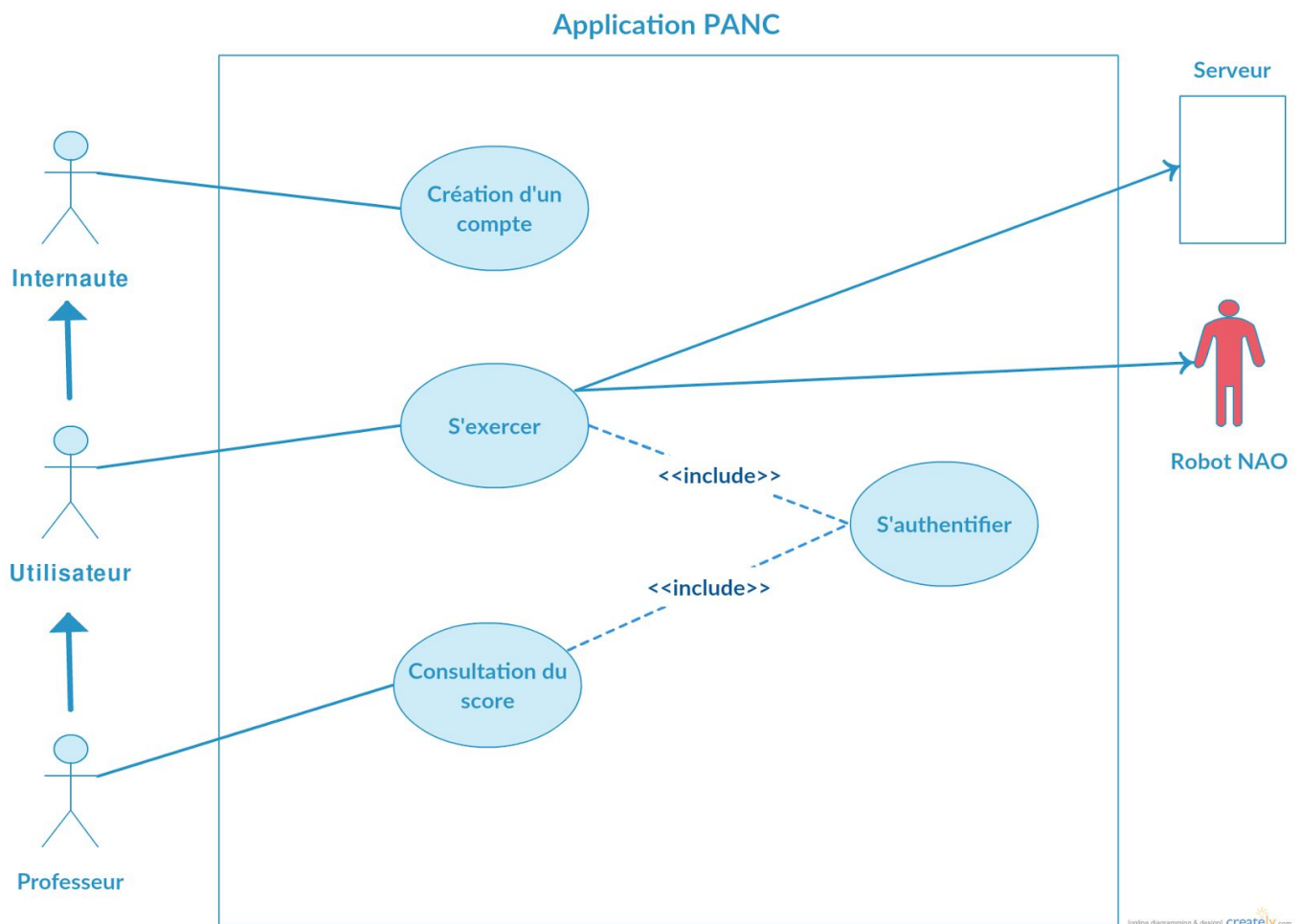
Notre choix d'implémentation assurera l'intégrité et la sécurité des données et fera en sorte que la maintenance du système hérite également de ces propriétés.

Les langages de programmation que nous avons choisi sont très répandus et utilisés. Ceci permettra de faire évoluer l'application sans problème (même par des personnes extérieures) et apportera au système le caractère d'opérabilité.

Il faudra intégrer des tests unitaires (puis d'intégration, fonctionnels) le plus tôt possible afin que notre futur système soit de bonne qualité et qu'il réponde au mieux aux besoins et à nos attentes. Nous utiliserons TestLink et Selenium pour ceci.

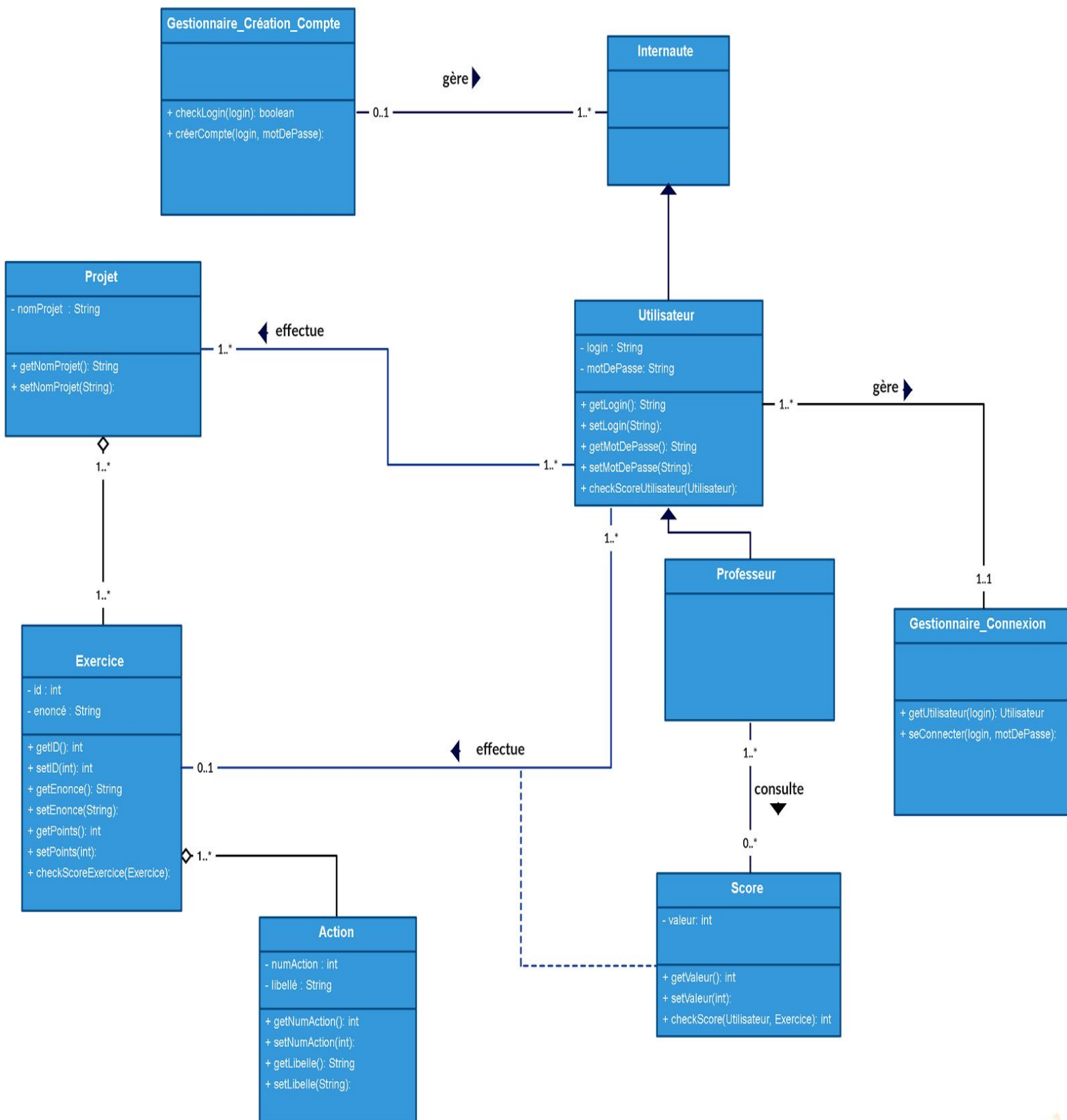
Afin de travailler en équipe dans les meilleures conditions possible, nous utiliserons GIT en tant que plate-forme collaborative et logiciel de gestion de versions. Pour communiquer, nous utiliserons Slack. Enfin, pour gérer la répartition des tâches et du planning, nous utiliserons Trello.

Système: vue fonctionnelle



Notre application présentera divers profils. Tout d’abord il y aura le profil “Internaute” qui pourra créer un compte. Ensuite, il aura le profil “Utilisateur” qui hérite de toutes les fonctionnalités de “Internaute” et pourra en plus s’exercer. Pour pouvoir s’exercer, il faudra obligatoirement s’authentifier et les informations envoyées lors des exercices seront reçues par le serveur. Finalement, il y aura le profil “Professeur” qui hérite de toutes les fonctionnalités de “Utilisateur” et qui pourra en plus consulter le score des utilisateurs. Pour pouvoir faire cela, il devra aussi s’authentifier obligatoirement.

Système: vue statique



Système: vue dynamique

Diagramme de séquence: S'authentifier

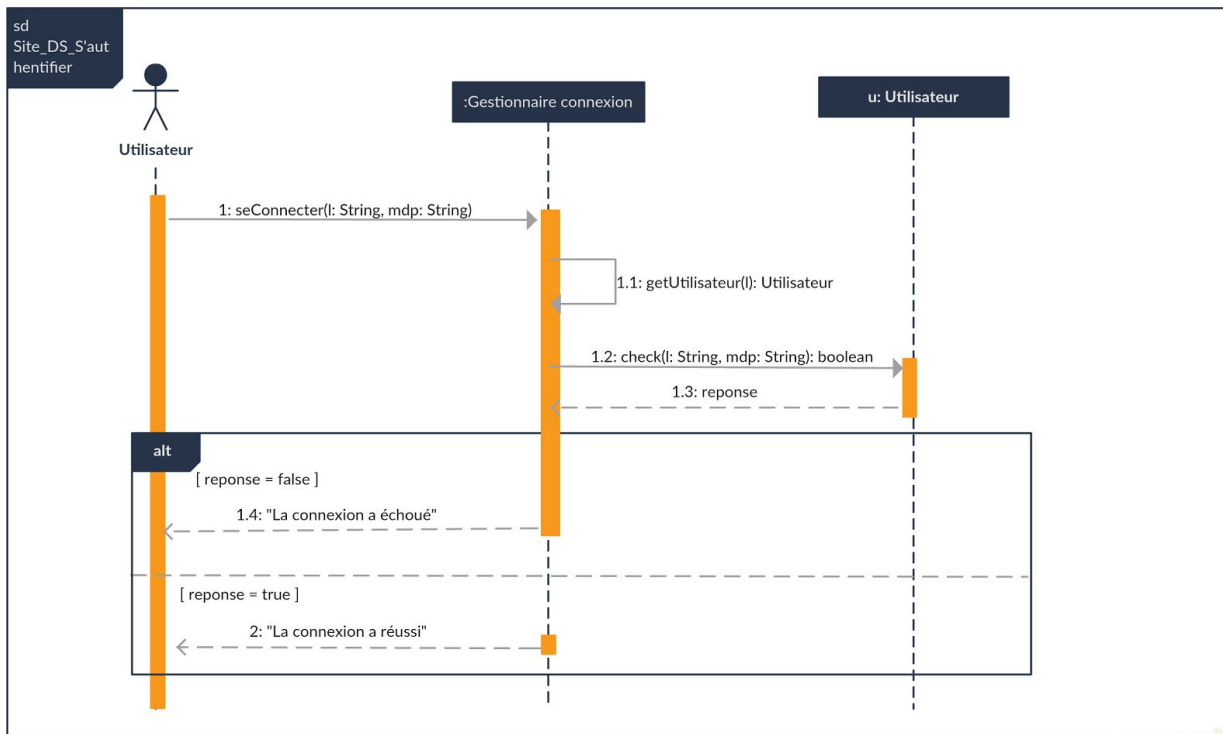


Diagramme de séquence: Création compte

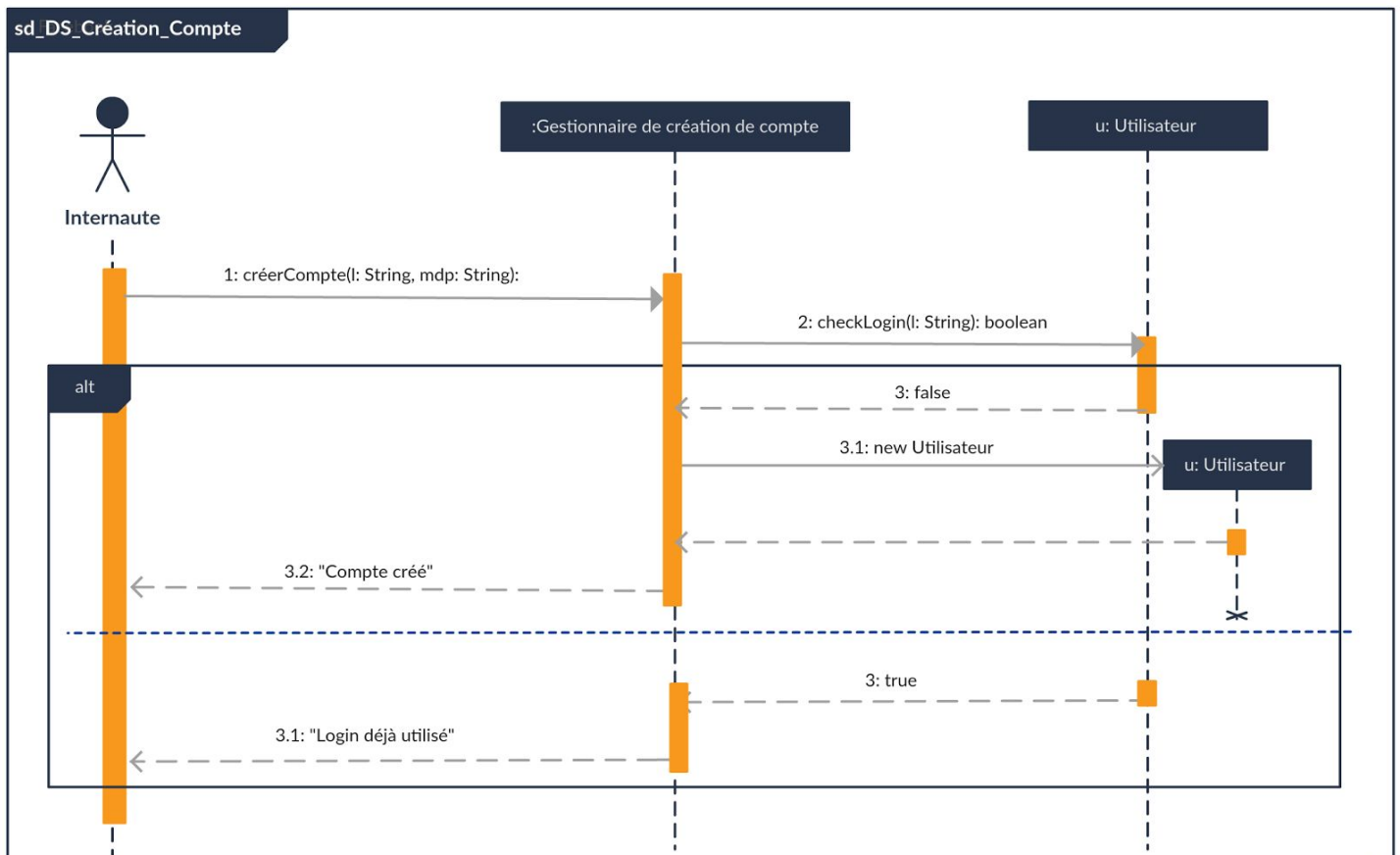
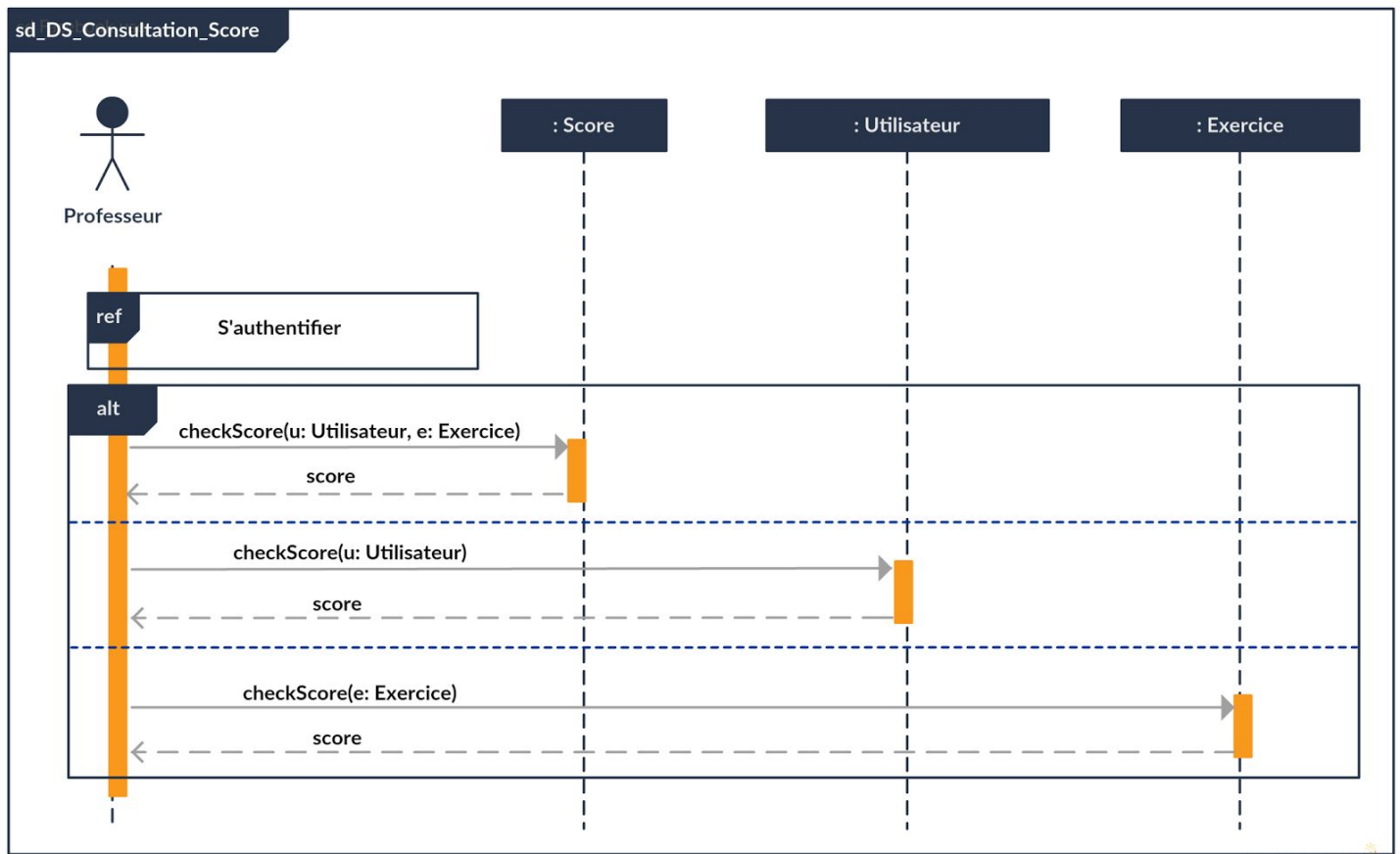
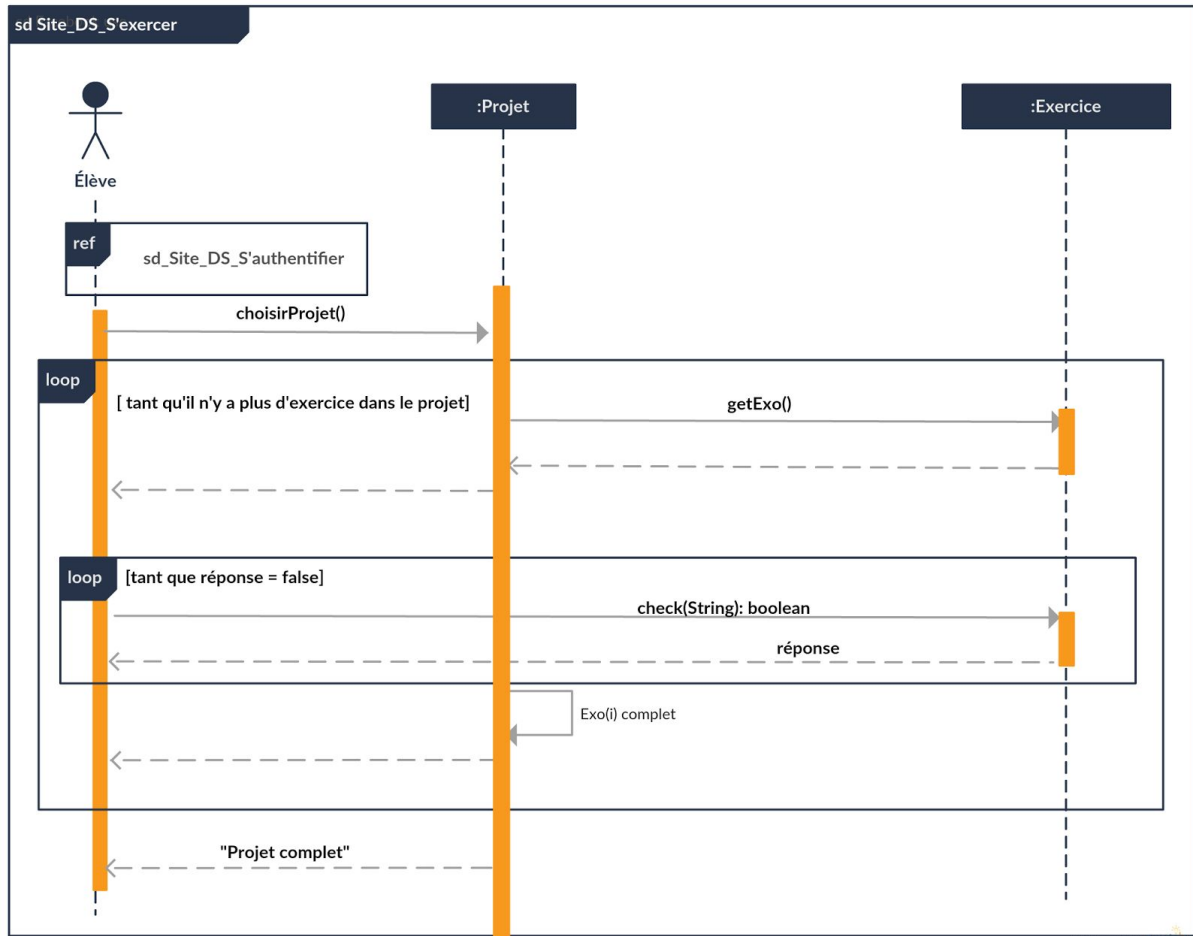


Diagramme de séquence: Consultation score



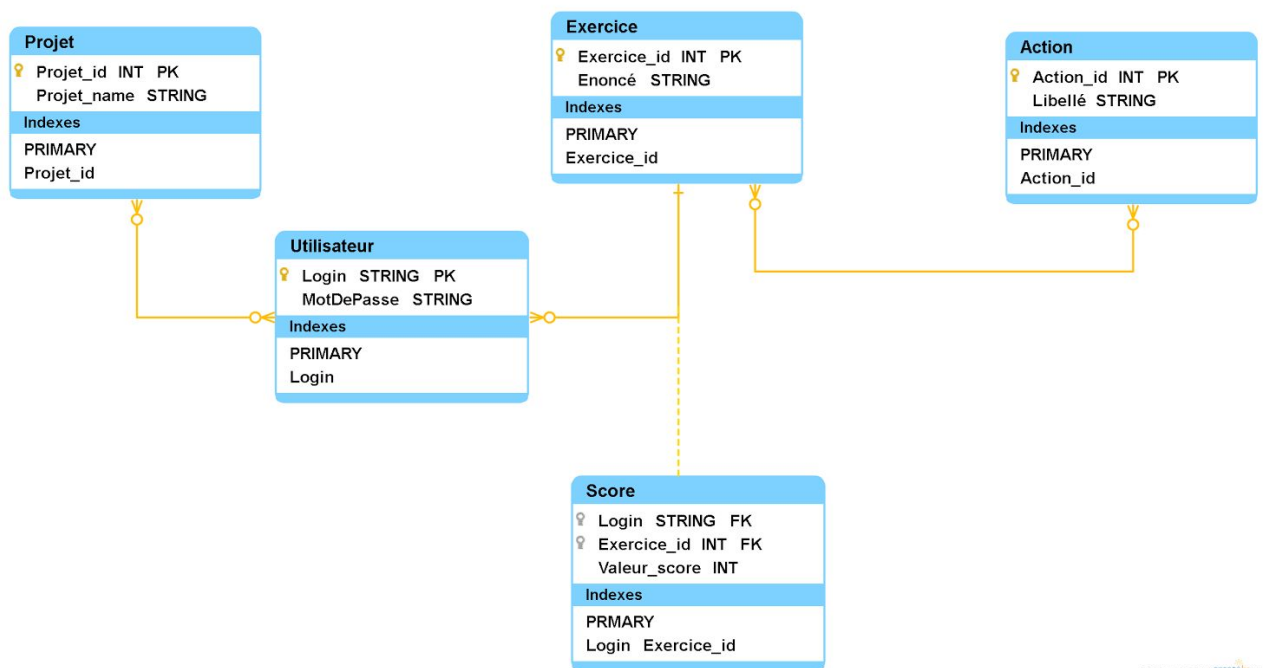
On peut consulter un score pour un utilisateur et un exercice donné, pour un utilisateur et donc tous ses scores ainsi que pour un exercice donc tous les scores obtenus à cet exercice.

Diagramme de séquence: S'exercer

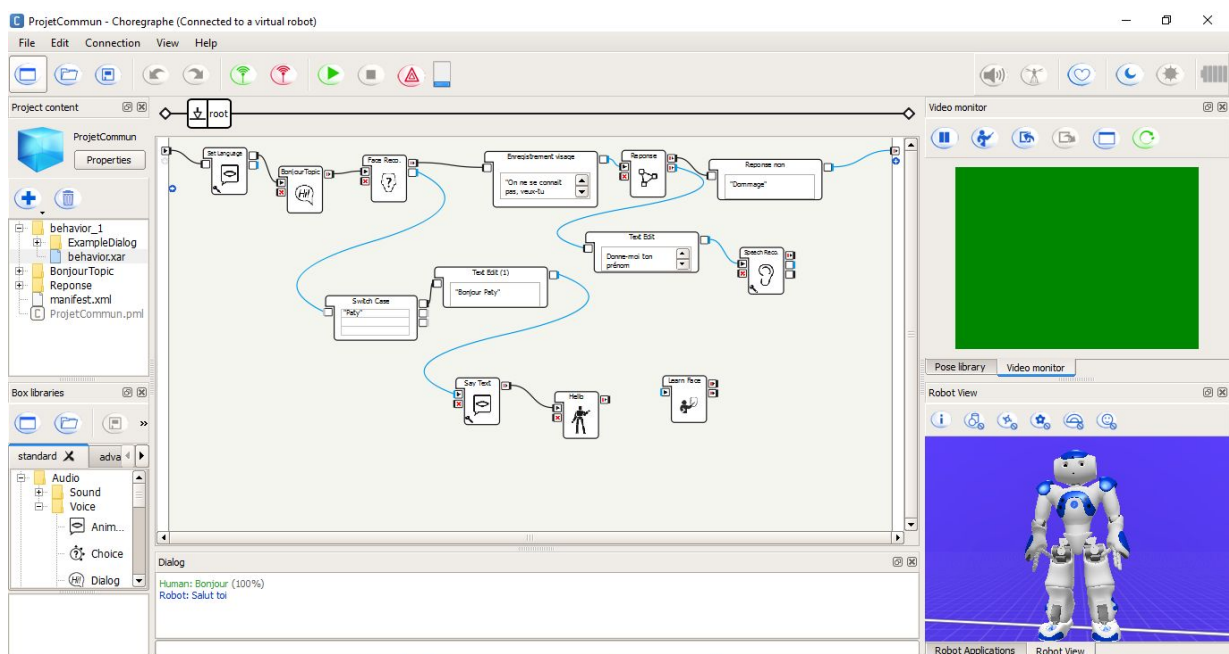


Choix d'implémentation

❖ Schéma relationnel: Base de données



❖ L'environnement de développement **Choregraphe**: c'est un outil de programmation qui permet de modifier les mouvements et les comportements de NAO.



- *“L'interface présente un espace de travail sur lequel il est possible de glisser-déposer des composants graphiques appelés boîtes. Chacune représente une action, un comportement ou un traitement. Il est possible de composer et d'associer des boîtes afin de créer des composants personnalisés de plus en plus complexes tendant, en général, vers un comportement. L'association des boîtes est représentée par des flèches qui orientent leur enchaînement ; cela permet de décrire aisément le flot d'exécution.”*

❖ **Python:** un des langages utilisés pour la programmation du robot NAO.

Python possède les caractéristiques suivantes qui le rendent différent des autres langages de programmation:

- C'est un langage open-source.
- Le langage est très facile à apprendre car son code source ressemble au pseudo-code
- La syntaxe permet aux utilisateurs de lire le code du programme facilement et le rend approprié pour le développement de prototypes et les différentes tâches de programmation ad hoc.
- Le code écrit en Python est incorporable dans des applications ayant utilisé d'autres langages comme C, C ++ et Java.
- Python est livré avec un environnement de développement intégré appelé IDLE et offre une grande bibliothèque standard.
- En mode interactif, Python peut facilement tester même de petites portions de code
- On peut facilement créer un serveur

❖ **HTML:**

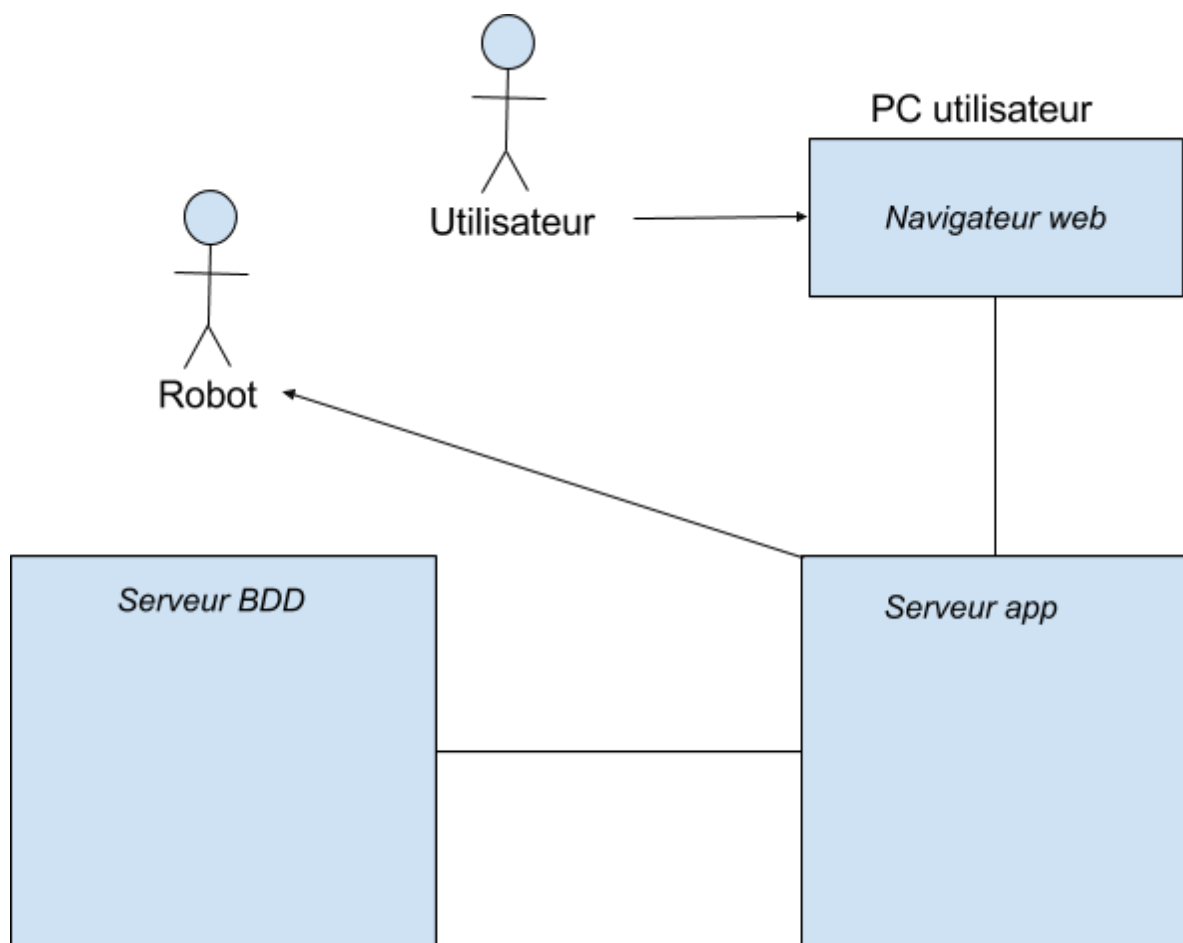
- développement de l'application web
- facilite les liens d'un serveur vers un autre, le texte HTML peut être écrit avec un éditeur standard. Cependant, le langage HTML n'est pas assez complet, les possibilités de mise en forme sont limitées et parfois complexes à mettre en oeuvre et mélange les éléments du contenu et de design d'un site.

❖ **PHP:**

- développement de l'application web
- disponible sur la grande majorité des hébergements grand public, permet facilement d'incorporer des composants dynamiques dans un site statique.

- ❖ **MYSQL**: permet d'avoir accès et de manipuler des bases de données
 - Le serveur MySQL est très rapide
 - facile d'utilisation
 - On peut effectuer diverses opérations sur une base MySQL en utilisant des interfaces écrites en C, Perl, C++, Java, Python, PHP
- ❖ **CSS**:
 - c'est un langage qui permet de décrire un document HTML
 - il permet de décrire comment les éléments HTML doivent être affichés

❖ Diagramme de déploiement



Conduite du projet

❖ Emploi du temps général

➤ *semaine 44-46* : découverte des possibilités de NAO , se renseigner sur le codage du robot, maîtrise basique du langage Python, utilisation de Chorégraphe
Nous avons travaillé séparément afin de se renseigner et de se familiariser au mieux avec l'environnement. Ensuite nous nous sommes concertées afin de définir le but et concept de notre projet.

➤ *semaine 47-50* : nous nous sommes renseignées sur le fonctionnement d'un serveur (Python), la façon de l'implémenter et de le faire communiquer avec le robot. Nous avons fait les premiers essais de serveurs et de communication avec l'application web.
Définition de notre application, concept, fonctionnalités...

semaine 51-8 : codage Python, développement de l'application web et du serveur, tests de la communication entre le serveur et l'application et entre le serveur et le robot. Répartition des tâches de développement.

semaine 9-13: codage Python, développement de l'application web et du serveur, dossier d'architecture. Amélioration de l'application et du contenu existant. Internationalisation de l'application.

semaine 14-fin : finitions, tests END to END et UAT (user acceptance tests), rendu final

❖ Outils pour mener à bien le projet

- **Slack**: communication au sein du groupe (<https://projetpanc.slack.com/>)
- **Trello**: répartition des tâches et gestion du planning
- **TestLink/Selenium**: création et exécution des tests
- **GIT**: plateforme collaborative et gestion de versions

Bibliographie

- <https://www.epi.asso.fr/revue/articles/a1206f.htm>
- <http://www.1776.vc/insights/robotic-integration-into-our-daily-lives-now-and-the-future/>
- <http://hackerspace.cs.rutgers.edu/library/NAO/NAO-programming.pdf>