

A thick dark grey vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the text '2022-2023'. In the bottom left corner, several thin, curved lines in dark grey and light grey sweep upwards and to the right.

2022-2023

Project CYRYWEB

Application d'exercices en Anglais

*@Yoann Jeauneau, Romain Guyenot, Yael Benguigui, Corentin
Champion*

<i>Propriétés</i>	<i>Description</i>
Intitulé	Développement d'un logiciel permettant à des utilisateurs de réaliser des exercices ludiques d'anglais.
Description	CyryWeb est une application visant à proposer des exercices d'anglais, incluant une progression par un système de points et de niveaux permettant de débloquent des difficultés supplémentaires.
Durée estimée en heure	Environ 80 heures
Langages et outils Utilisés	<ul style="list-style-type: none"> • Java JEE • MySql • Java Servlet, JSP • JDBC connector • IDE : Eclipse • PhpMyAdmin • GitHub : https://github.com/CYRY-Project/CYRYWeb • Google drive • Trello • Whimsical

Savoir Faire SI Mobilisés en priorité	<p><i>Les savoir-faire de la phase d'étude du projet, auxquels s'ajoutent :</i></p> <p><i>SI3 - Extraire et modifier les données d'une base de données</i></p> <p><i>SI4 – Programmer à l'aide d'un langage de programmation structuré</i></p> <p><i>SI4 – Programmer en utilisant des classes d'objet</i></p> <p><i>SI4 – Appliquer des normes de développement</i></p> <p><i>SI6 - Concevoir une interface utilisateur</i></p> <p><i>SI6 - Valider et documenter une application</i></p> <p><i>SI6 - Rédiger une documentation d'utilisation</i></p> <p><i>SLAM1 – Modifier un schéma de données et l'implantation de la base de données correspondante</i></p> <p><i>SLAM1 – Utiliser un outil de génération et de rétro conception de base de données</i></p> <p><i>SLAM1 – Adapter une application exploitant une base de données à l'évolution de son schéma</i></p> <p><i>SLAM2 – Programmer à l'aide d'un langage de programmation objet</i></p> <p><i>SLAM2 – Configurer et utiliser un environnement de développement</i></p> <p><i>SLAM2 – Mettre au point un programme</i></p> <p><i>SLAM3 – Programmer dans l'environnement de développement associé à un SGBD</i></p> <p><i>SLAM4 – Programmer un composant logiciel</i></p> <p><i>SLAM4 – Exploiter une bibliothèque de composants</i></p>
Documents joints	<p>Cahier des charges, bases de donnée, code de l'application respectant le modèle MVC</p>
Modalité de réception	<p>Production d'un système opérationnel VI</p>
Equipe	<p>Travail réalisé majoritairement individuellement, avec quelques parties en binôme. Le projet global est un projet de 4 personnes pour lequel des outils de gestion de projet et de versioning ont été mis en place et utilisé.</p>
Comptes test	<p>Compte 1 (niveau 1) : utilisateur test1 ; mot de passe test1 Compte 2 (niveau 16) : utilisateur test2 ; mot de passe test2</p>

Cahiers des charges

- L'utilisateur

Il peut réaliser des exercices de niveaux et de type différents en fonction de leurs niveaux.

Il peut aussi :

- **Créer des comptes**
- **Modifier son pseudo**
- **Supprimer son compte**
- **Suivre leur progression sur leur profil**

- Les exercices

L'application comporte 5 type d'exercices au total, à savoir :

- **Question à Choix Multiples (QCM) – Multiple Choice Questions (MCQ)**

Une photo, une question sous la photo, et 4 choix de réponses possibles avec une seule bonne réponse possible.

- **Situation – Situation (Situation)**

Une phrase sera posée sous forme de question et 4 images seront proposées en réponse. Une seule réponse possible.

- **Texte à trous – Gap Fill Text (GapFillText)**

Une ou plusieurs phrases sont proposées et comportant des « trous ». Les choix possibles de réponse sont ici proposés dans un ComboBox. Pour valider l'ensemble de l'exercice, l'utilisateur devra la ou les bonnes réponses.

- **Traduction – Translation (Translation)**

5 mots en anglais et 5 mots en français sont proposés. L'utilisateur doit associer chaque mot anglais à sa version française. Pour valider l'exercice, l'utilisateur doit trouver les 5 associations.

- **Construction de phrase - Sentence Construction (SentenceConstruction)**

Plusieurs mots sont proposés sous la forme de bouton, et l'utilisateur doit reconstituer une phrase cohérente en anglais. Pour valider l'exercice, l'utilisateur doit reconstruire la phrase attendue.

Les niveaux

L'application comporte 3 niveaux de difficulté, à savoir : débutant, intermédiaire et avancé. Chaque difficulté comporte 10 exercices (2 QCM, 2 traductions, 2 situations, 2 textes à trou, 2 construction de phrase).

Pour débloquent les niveaux de difficulté supplémentaire, l'utilisateur doit pour cela valider des exercices et augmenter son niveau de compte. A chaque exercice validé, l'utilisateur gagne 500 points. Tous les 1000 points l'utilisateur gagne 1 niveau de compte. Les différents niveaux de compte requis pour débloquent chaque niveau de difficulté :

- Débutant : accessible dès la création du compte
- Intermédiaire : niveau de compte supérieur à 5
- Avancé : niveau de compte supérieur à 10

Spécificités techniques

- Langage de l'application et graphique JAVA
- Base de donnée : MySQL
- Structure de l'application : POO en MVC

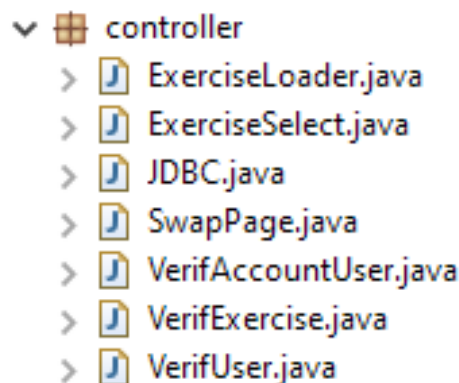
Model View Controller

- Controller

Définis dans le diagramme de classe. Permet de voir toutes les interactions entre les objets.

3 rôles principaux pour nos controllers :

- Contrôle de saisie (sécurité)
- Vérification des exercices
- Aiguillage vers les différentes vues



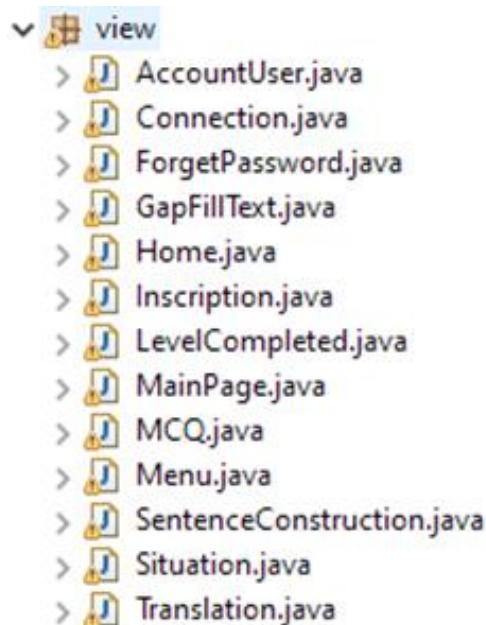
- View

On définit toutes les différentes pages/vue possibles.

Toutes les vues auront en commun une JFrame (la fenêtre), ainsi qu'en haut de chaque vue, une fois connecté sur un compte utilisateur, un bouton permettant d'afficher un menu ainsi que l'avatar de l'utilisateur qui lui permettra d'accéder aux diverses informations de son compte.

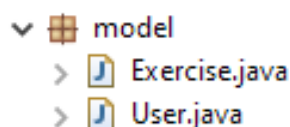
Et des fenêtres à modifier selon les actions :

- Affichage des exercices en fonctions de la difficulté
- Affichage du menu
- Affichage du profil utilisateur
- Affichage de la page d'inscription
- Affichage de la page de connexion
- Affichage de la page mot de passe perdu



- Model

Les modèles nous permettent de créer des instances d'objets prédéfinies qui seront ensuite utilisées ou modifiées au sein de l'application. Ils nous permettent de créer des instances d'exercices et d'utilisateur.

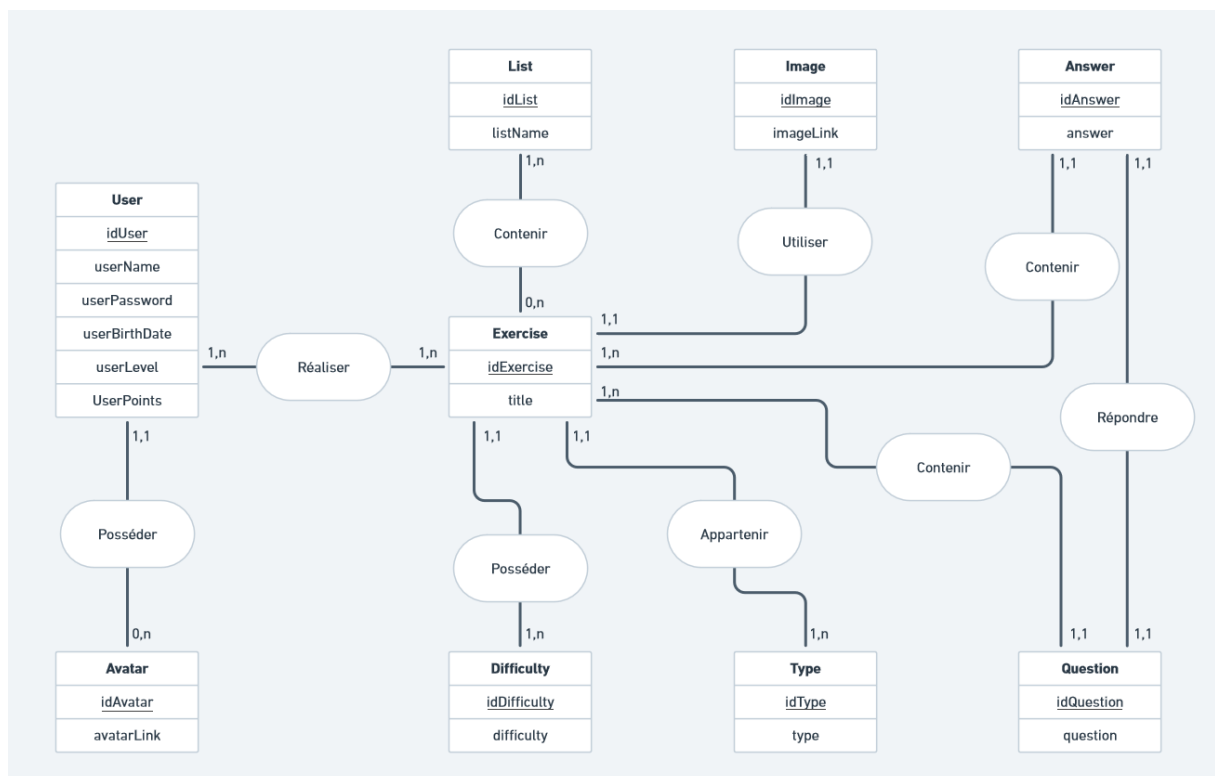


Base de donnée

- Dictionnaire de Données

Nom	Description	Type	Commentaire	Entité
userName		VARCHAR(20)	nom d'utilisateur	USER
userPassword		VARCHAR(20)	mot de passe de l'utilisateur	USER
userBirthDate	JJ-MM-AAAA	VARCHAR(10)	date de naissance de l'utilisateur	USER
userLevel		int	niveau de l'utilisateur	USER
userPoints		int	points de l'utilisateur	USER
avatarLink	%.%	VARCHAR(255)	nom et extention de l'avatar	AVATAR
title		VARCHAR(255)	titre de l'exercice	EXERCISE
type		VARCHAR(255)	type de l'exercice	TYPE
difficulty		VARCHAR(255)	difficulté de l'exercice	DIFFICULTY
question		VARCHAR(255)	question (phrase, mot, ...)	QUESTION
answer		VARCHAR(255)	réponse (phrase, mot, ...)	ANSWER
imageLink	%.%	VARCHAR(255)	nom et extention de l'image	IMAGE
listName		VARCHAR(20)	nom de la liste personnalisée	LIST

- Modèle Conceptuel de Données (MEA)



- Modèle Logique de Données (SRD)

AVATAR (idAvatar, avatarLink)

clé primaire : idAvatar

USER (idUser, userName, userPassword, userBirthDate, userLevel, userPoints, idAvatar)

clé primaire : idUser

clé étrangère : idAvatar en référence à idAvatar de la table AVATAR

EXERCISEDONE(idUser, idExercise)

clé primaire : idUser, idExercise

clé étrangère : idUser en référence à idUser de la table USER

idExercise en référence à idExercise de la table EXERCISE

EXERCISE (idExercise, title, idType, idDifficulty)

clé primaire : idExercise

clé étrangère : idType en référence à idType de la table TYPE

idDifficulty en référence à idDifficulty de la table DIFFICULTY

IMAGE (idImage, imageLink, idExercise)

clé primaire : idImage

clé étrangère : idExercise en référence à idExercise de la table EXERCISE

TYPE (idType ,type)

clé primaire : idType

DIFFICULTY (idDifficulty, difficulty)

clé primaire : idDifficulty

QUESTION (idQuestion, question, goodAnswer, idExercise)

clé primaire : idQuestion

clé étrangère : goodAnswer en référence à idAnswer de la table ANSWER

idExercise en référence à idExercise de la table EXERCISE

ANSWER (idAnswer, answer, idExercise)

clé primaire : idAnswer

clé étrangère : idExercise en référence à idExercise de la table EXERCISE

EXERCISELIST (idList, idExercise)

clé primaire : idList, idExercise

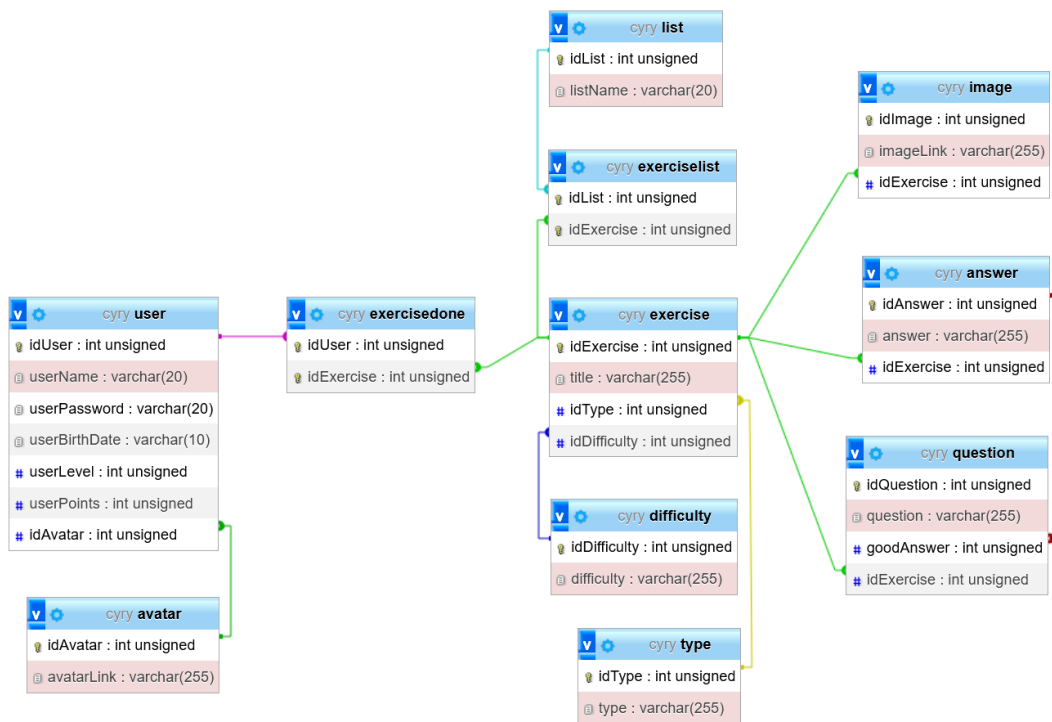
clé étrangère : idList en référence à idList de la table LIST

idExercise en référence à idExercise de la table EXERCISE

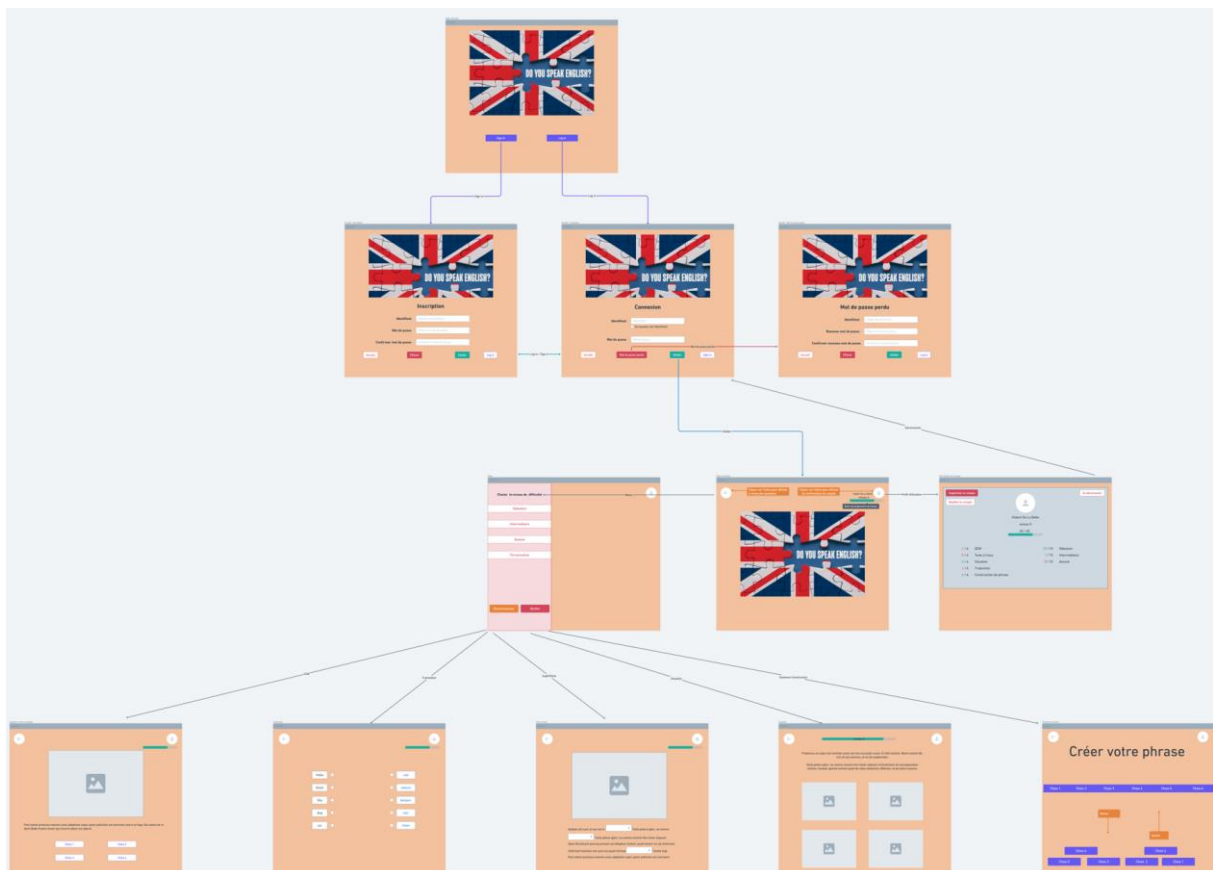
LIST (idList, listName)

clé primaire : idList

- Vue concepteur



- Diagramme des vues



- Diagramme de classe

Idées d'amélioration possible

- Ajouter un compte administrateur qui aurait comme droit la gestion des comptes (création, modification, suppression), ainsi que la possibilité d'ajouter ou de supprimer des exercices par le biais de l'application.
- La création de niveau personnalisé, où l'utilisateur peut mixer plusieurs exercices de différents types et de difficulté différente.
- Ajouter une rubrique « Grammaire » et « Conjugaison » pour apprendre les règles de grammaire et de conjugaison.
- Ajouter un système de changement d'avatar dans le profil utilisateur
- Améliorer l'interface graphique.
- Ajouter un système de notation des exercices pour un retour expérience des utilisateurs.
- Ajouter un système d'achat de bonus en dépensant des points.