Projet TER M1 IC

Development plan

Descotils Juliette
Gattaciecca Bastien

Table of content

This development plan details the general design of the web application. We divide the parts of this document on all the different features described in the specification.
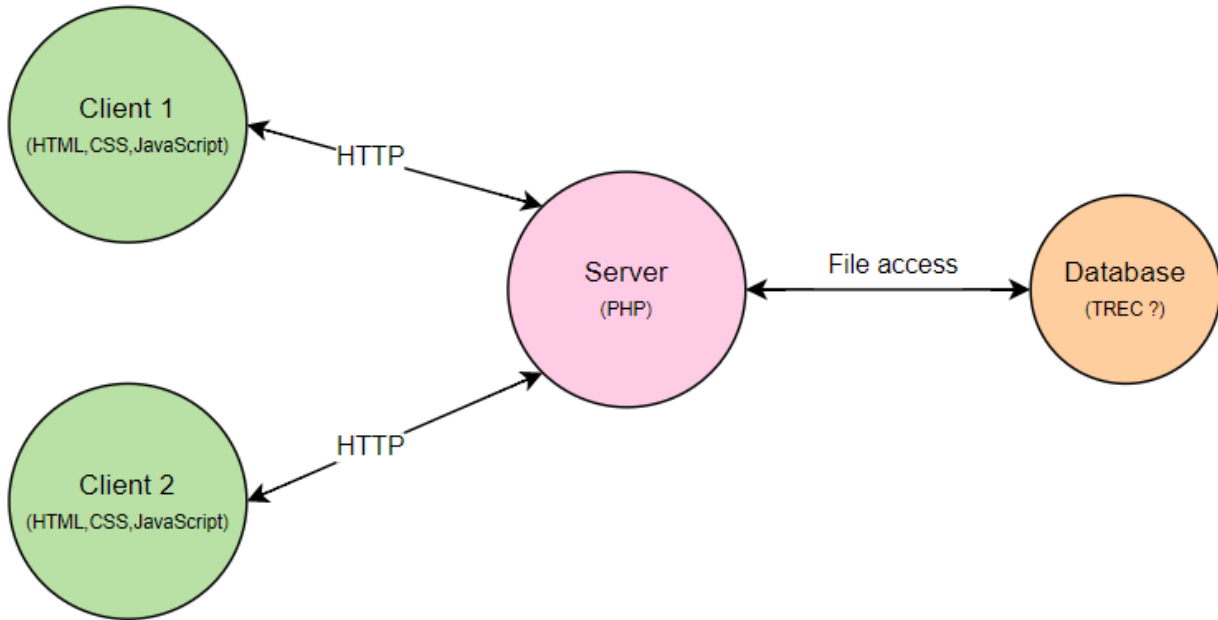
1. Global architecture

Diagram 1 - architecture diagram between the client and the server

As we can see on the architecture diagram, several clients can interact with the server at the same time. The client-server communication is done via an HTTP protocol. The server allows the client to download and import the files stored in the database.
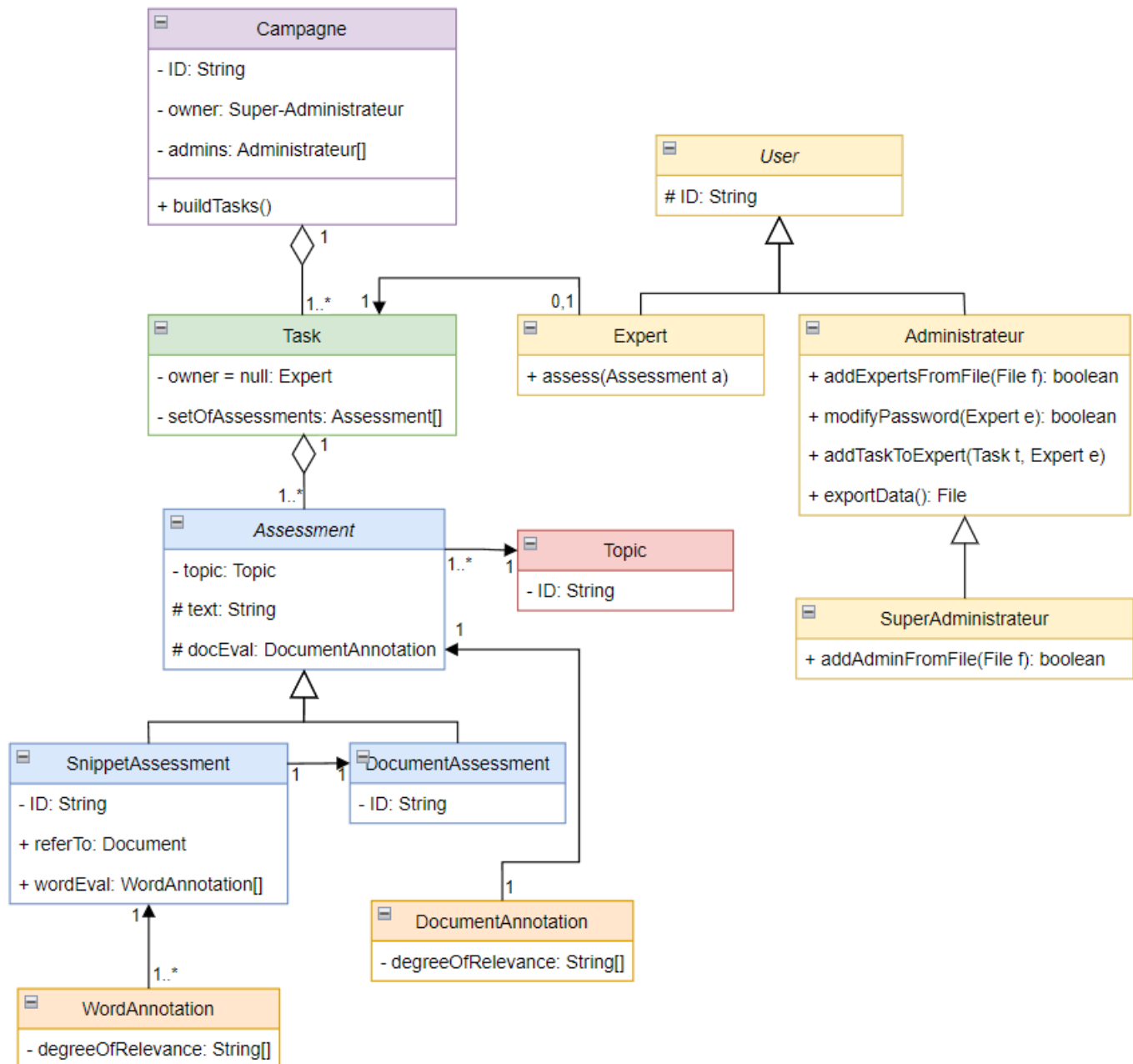
Campagne
- ID: String
- owner: Super-Administrateur
- admins: Administrateur[]
+ buildTasks()

User
ID: String

Task
- owner = null: Expert
- setOfAssessments: Assessment[]

Expert
+ assess(Assessment a)

Administrateur
+ addExpertsFromFile(File f): boolean
+ modifyPassword(Expert e): boolean
+ addTaskToExpert(Task t, Expert e)
+ exportData(): File

Assessment
- topic: Topic
text: String
docEval: DocumentAnnotation

Topic
- ID: String

SuperAdministrateur
+ addAdminFromFile(File f): boolean

SnippetAssessment
- ID: String
+ referTo: Document
+ wordEval: WordAnnotation[]

DocumentAssessment
- ID: String

DocumentAnnotation
- degreeOfRelevance: String[]

WordAnnotation
- degreeOfRelevance: String[]

Diagram 2 - class diagram, application data model

Below, we present the class diagram as the data model of the application. In yellow represents the inheritance hierarchy of the users. The User class is an abstract class for users and has an ID.

A Campaign has an ID. At initialization, it has a SuperAdministrator who must add administrators to this campaign. And again, the SuperAdministrator and Administrators add Experts to the campaign.

When all the experts are added, the Task tasks are built according to the number of experts added. At initialization, a Task is not assigned to any Expert. Once all the tasks are built, they are assigned to the different experts added to the campaign.

Once assigned, the Task Assessments are annotated by the Experts associated with these Tasks.

The annotations made by the Experts are modelized by the WordAnnotation and DocumentAnnotation classes. WordAnnotation refers to the annotation of each word of a Snippet and DocumentAnnotation refers to the annotation of a document or a snippet as a whole.

2. Mockup

Here we present a mockup of the experts' dashboard. E1 and "start assessing" are buttons. The "start assessing" button is disabled when there are no documents/snippets left to assess. E1 is a menu which contains only the logout button.

Campaign name
Assessment type : snippets
Remaining assessments : 392
Remaining topics : 8

N°	Topics (9)	Completed assessments (58)	Total assessments (450)
1	Topic's title	50	50
2	Topic's title	8	50
3		0	50
4		0	50
5		0	50
6		0	50
7		0	50
8		0	50
9		0	50

Start assessing

3. Page structure (HTML and CSS)

Views	Elements	Library, package, methods
Login	Login textfield Password textfield Connection button	input button
Extended dashboard from expert view	Parameters button "start assessing" button The reduced dashboard is set on the page, it is always visible even when the expert scrolls down.	tr/td/th for the table button Using position css attribute at 'fixed'

4

Log out	"deconnection" menu item	button
Assessing a snippet/document	Display of "number of remaining assessments :" Display of the instructions Reduced dashboard content: - instructions - the snippet/document Abandon button Validate button (both button are rooted to the end of the document)	button text text button button Using position css attribute at 'fixed'
Admin dashboard	parameters button "add experts from file" button "add task" button located in the table (referring to an expert) Change view from expert view to topic view	button tabs/tab

4. Front-end development (JavaScript)

We decided to choose VueJS as our framework because it is one of the most commonly used, it is an open-source JavaScript framework made for building web UIs and single-page applications. It is known as an easy to learn framework while being very complete and allowing to produce high-quality interfaces. It facilitates the implementation of a MVC model.

Views	Element	Library, package, methods
Login	Display dashboard if id correct Display error message if id incorrect	@click to add an event on the connection button. The event is displaying the expert or the administrator dashboard according to the type of credential. If the credential is not existing in the experts dictionary display a warning message.
Extended dashboard from expert view	"start assessing" button The "start assessing" button becomes unclickable when the number of assessments is 0. Display data (number of completed assessments, total, etc.) from the dashboard.	Use @click to add an event on the "start assessing" button. The event is the display of the next assessment uncompleted. bind the disabled attribute of the "start assessing" button to "isDisabled" The data is stored in a View and displayed with the v-for directive.
Assessment page	One button on each word of the assessment to set the degree of relevance (left click increments it, right click sets it to the max level) Change the background color of a word according	Each document/snippet is seen as a string. Generate a button for each word of the string. Use @click to add an event on each button. Use event handler to associate right click with the event of setting the highest relevance degree. Use prevent default to prevent the contextual menu from being displayed. Use v-bind:style'{backGroundColor : color} on

	to its relevance degree.	each button.
	Button validate assessment and go to the next one.	Use @click to add an event on the "validate assessment" button. Displays next assessment's interface.
	Button abandon assessment and go back to dashboard	Use @click to add an event on the "validate assessment" button. Displays dashboard's interface.
	Parameters button → deconnection	Use @click to add an event on the "deconnection" button. Displays connection page.
Admin dashboard	Download data	Use @click to add an event on the 'Download data' button. The output files are then downloaded to the machine.
	Open menu (déconnection)	
	Add task to an expert	Use @click to add an event on the menu button then we open the drop-down menu.
	"Add expert from file" button	
		Display the new assessment in the list of assessment.
		Use @click to add an event on the button to open a file choser.

5. Backend development (PHP)

Element	Sub-elements	Library, package, methods
Login page	Check the value of the credential	With the global variables $_POST
Admin dashboard	The administrator adds a task to an expert.	We edit the file in which the assessments are distributed to the experts (associating expert ID - doc/snippet ID) The amount of unfinished assessments and the total number of assessments are updated.
	The administrator adds some credential to the system	Notifying the server that an expert has been added to the system.
	The administrator imports the data.	Constructs a file with the local data from finished assessments. The file is created locally on the admin's pc.
	Campaign name Remaining topics	Variables retrieved from the server that are used to update the admin dashboard.

	Remaining assessments
	
	Table of topics including :
	
	    Total number of topics
	
	    Total number of completed assessment
	
	    Total number of assessment	

Assessing a snippet/document	The expert imports the snippet/document he has to assess from the server	Import the snippet/document from the server with his ID.
	Expert sends file after finishing assessing a snippet/document	Construct a file containing first the IDs that define the campaign, the topic, the snippet/document, followed by the list of annotated words and their associated relevance level. The file is then sent to the server.