[Focus France - Web/Mobile]

Functional Specification

Revision history

Date	Version	Description/ Type of modification
	0.1	First draft

[Focus France - Web/Mobile]

Functional specification

1. Introduction

This document contains the general functional requirements for the platform Focus France (web&mobile). This document contains general information, algorithm descriptions and non-functional requirements that is used as an addition to the Figma mockups where the platform functionalities are described. This document can be used by Noveo, the Customer and Auditors of the project.

2. General description of the project

The project aims the development of a platform Focus France (web/mobile).

2.1 Project context and goals

The project aims to develop a platform for quick and easy learning for undergraduate or high school students. The learning process is performed with quiz flashcards that show repeating questions to be answered by students via mobile app. Teachers create quizzes for students with a web app.

The platform consist of two modules:

- Teachers app (web) - the module for teachers where quizzes are created and students' performance is monitored. Teachers can use their PC, laptops or tablets (iPad is commonly used).
- Students app (mobile) - the mobile app (iOS +Android) where quizzes are passed and students own progress is monitored.

The general workflow:

- A teacher performs the following steps (using a web app):
    - Creates subject(s)
    - Creates quizzes
    - Assigns quizzes to the created subjects
    - Creates a class (a group of people connected by the same learning program)
    - For each semester of a class, a teacher assigns quizzes that are supposed to be passed/learned by students.
        - Subjects to which these quizzes are assigned are shown for teacher and students automatically
    - shares the link to join the class with students (QR-code or access code)
- Students:
    - scan the QR-code or input the access code to join the class
    - After joining the class, quizzes for each semester are available for all students in the class.
    - Students pass the quizzes during semesters using their mobile app and then see their results/score.
- Teacher:
    - Monitors the results in the teacher’s web app

2.2 Target user groups

- **Teachers app:** teachers and tutors who lead educational programs and courses before enrolling the high schools (Science PO).
- **Students app:** students of the age 17-18 who want to enroll in high schools and universities.

2.3 Roadmap

The following functionalities are expected for the MVP

- All students are registered, no anonymous users who can pass quizzes using pseudonyms.
- Teachers sign up / invitation is not expected. Only one teacher is created in the DB in advance by the dev team.
- Semester dates are fixed: 1/09 to 31/01 and from 01/02 to 31/08.
- The class entity is a group of students with the same learning program. The program is split into several semesters (max number of semesters is 6).
- Quizzes only study mode are expected (learning process).

For the future versions (approximately):

- Tournament mode.
- Anonymous quizzes passing using pseudonyms.
- Teachers invite and sign up.
- Semesters management: set the flexible dates, edit dates.
- Spaced repetitions for the study mode.

2.4 Browsers support

Chrome
Safari
Firefox
Versions are TBD

2.5 Markup requirements (web)

The following screen resolutions are expected to support:

- Narrow screens for tablets (iPad) - 1024-1440px - primary due to the fact that teachers mostly use their iPads.

2.6 Screen orientations support (mobile)

We assume only one screen orientation (portrait) is supported for the mobile app for the MVP. For future versions, landscape support can be added.

2.7 Languages support

Only one language support (French) is expected. Original mockups are provided in English for the development team convenience, and translations to French can be found here.

2.8 Animations

The custom animation is not required within the project.

2.9 Data flow description

Main entities

Subject

Subject is a knowledge domain.

Quiz properties:

- Name
- Quizzes - array of quizzes assigned to this subject

Quiz

Quiz is a list of initial questions with correct answers. From these initial questions the questions of different types are generated during the quiz.

Quiz is an array of objects (question cards) with the following properties:

- Initial question
- Correct answer

Quiz properties:

- Name
- Subject(s) - one quiz can have several subjects.

Quizzes are associated with a specific class semester and reused multiple times. One initial quiz can be assigned to several classes and several semesters with different deadlines.
The max number of questions for quizzes is limited (2500).

Class

Class is a group of users/students with the same learning program. They learn the same quizzes with the same deadlines.

Class properties:

- List of students enrolled in this class.
- List of class semesters:
    - Current
    - Finished
    - Upcoming

Class semesters (the first and the last) are defined on creation of the class and cannot be changed afterwards.

Class semester

Class semester is a list of quizzes to be learnt by students during a semester (half year) within the association to a specific class.

Class semester properties:

- Start and end dates - for the MVP these dates are fixed (from 01 Sep to 31 Jan, and from 01 Feb to 31 Aug), but in future versions class semester dates can be changed by a teacher.
- List of quizzes added to the semester. Each quiz added to the class semester has additional properties:
    - Due date - the deadline when the quiz shall be passed. If not defined, the semester end date is taken.
- List of subjects - quizzes added to the semester are grouped by subjects within the semester.
    - Quizzes are assigned to subjects on platform level (teacher assigns them on subject page)
    - Quizzes that are assigned to no subject are grouped under “No subject”
    - Each subject added to the semester have additional editable attribute:
        - Study time - defined by a teacher, estimated time needed for learning a subject during the semester.

Student's progress

Students pass quizzes during a semester and achieve results.

For each quiz passed by a student during a semester, the following parameters are recorded:

- Score - number of points earned.
- Quiz grade - the grade 0-20 derived from the quiz score
- Study time - time spent passing the quiz.
- Status - Not started, In progress, Complete, Overdue

During a semester, the list of subjects with student's results is collected:

- Subject score:
    - Subject study time - is a sum of study time of all quizzes of this subject during a semester
    - Projected time grade
        - = (study time / study time needed to reach grade=20) / (number of days passed / number of days in semester) * 20
    - Subject grade
        - = (study time / study time needed to reach grade=20) * 20

Teachers app (web) functional requirements

3.1 Users

For the MVP only 1 teacher will be created on the backend. Sign up will be developed in the future versions.

3.2 Notifications

Push-notifications for the teacher's app are not expected.

3.3 Offline mode description

Offline mode for the teacher's app is not expected.

Students app (mobile) functional requirements

4.1 User roles

Only one role is expected for the students app. All functionalities are available for all users, no split or restricted access is expected.

4.2 Work with unauthorized users

For the MVP, all users must register to the app. Work with unauthorized users is not supported. In future versions, anonymous users who can pass quizzes using pseudonyms are expected.

4.3 Notifications

Push-notifications are not expected.
The following notifications is in scope of the MVP:

- In-app notifications (toasts).

4.4 Offline mode description

Students might have unstable internet connections, but they still want to keep their learning process ongoing.

According to that, the following conditions applied for the application:

- To start the quiz, an internet connection is needed to load all the questions. We expect **up to 2500** initial questions to be loaded on a device without performance issues.
- When a quiz is started, all questions are already loaded to the device, so they can be shown offline.
- When the quiz is finished or stopped, the results are sent to the server to be available for teachers.
- If the server is unavailable (a student is offline), the results are saved on the device.
- When the connection is restored on the opened app, the corresponding data is sent to the server.
- A student can close the app, but their progress is saved on the device. When the app is opened next time and online, all the unsaved changes and student's progress are sent to the server.
- Questions for all the quizzes in progress are saved on the device. Once a quiz is complete and the result are saved to the server, its questions can be deleted from the device.

- If a student with quizzes in progress has installed an app on a new device (no questions/quizzes are saved on this device), the questions for quizzes started/resumed shall be uploaded from the server/backend when a quiz is started or resumed, so the internet connection is required in this case.

For the future versions with tournaments:

- Offline and back to online can be shown for students during the learning process and during tournaments (toast notifications, icons, etc).
- Offline students should not stop the tournament and keep other students awaiting for them.
- Once students are offline during tournaments, they are marked as "offline" for a teacher, and their results cannot be updated.
- When offline students have their connection restored, their results are updated.
- Team mode sync: TBD later.

4.5 Students accounts data collection and storage

For the MVP, only registered users can access the application functionality.

The following conditions are applied for the data collected during the registration process and application usage:

1. The data collected:
    a. User's name;
    b. User's email.
2. Purposes of the data collection:
    a. Account registration;
    b. Studen's learning progress storage.
3. Storage conditions:
    a. Users' accounts data is saved to the internal system database.
    b. Server's location - France (where the system will be deployed).
4. Data retention and deletion policy:
    a. After the account deletion is initiated by a user, it has to be marked as inactive in the database with the possibility to restore it by a user's inquiry during 1 month after the deletion request.
    b. After 1 month of an account being inactive, the account is deleted forever from the database.

For the MVP, the user profile for the student app is out of scope (update from 18/09/2023), so the account deletion from the app UI is not expected. However, the "Privacy Policy" or "Terms of use" text will have a notice with an email where users can enquire for the account deletion.

The following consequences are applied for inactive/deleted accounts:

- Student's name is removed from all quiz results: teachers cannot see the deleted student's name.
- Student's name is removed from all classes.
- When an account is restored from being inactive, all the learning progress is restored for this user, a user is returned back to all classes and quizzes they attended before deletion.

4.6 Quiz algorithms

4.6.1 Question generation algorithm

Questions are uploaded (using copy+paste or upload the xls-file) to the systems in the following format:

- Initial question text that usually has spaces ___ where the expected answer is supposed to be. E.g.: *Insanité d'___ : altération des facultés mentales.*
- Initial answer, usually it's a short string, one word (but not necessary). E.g.: *esprit*.

From these *initial questions* and *answers* the *generated questions and answers* of each type are shown during the quiz.

4.6.1.1 Question type True/False generation

The question text is generated the following way:

%Initial question text% - taken from the initial question with no changes
%Answer% - taken from the same initial question if the expected correct answer is Yes or from another initial question (random) if the expected correct answer is No.

E.g:

Insanité d'___ : altération des facultés mentales.
Esprit

The user is supposed to answer this generated question with buttons Yes or No.
If the generated answer is taken from the same question (it's correct), the correct answer for the True/False type of question is Yes.
If the generated answer is taken from another question (it's incorrect), the correct answer for the True/False type of question is No.

NB: Taking into account that some initial questions may have the same initial answer, the generated answer taken from another initial question must not be the same as the initial answer for this question.
E.g:
Initial question: TMG = ___ / PA

Initial answer: MBU
Another initial question with the same answer MBU: ___ = PVHT - PA
This answer from this initial question must not be taken to make the generated question with the correct answer "No".

4.6.1.2 Question type MCQ generation

The generated multiple choice questions are shown with 4 answer options: one of them is correct, others are incorrect. Users choose one correct answer by clicking on it.

The question text is generated the following way:

%Initial question text%
%Answer option 1%
%Answer option 2%
%Answer option 3%
%Answer option 4%

Generated answers are taken from answers on other questions from the initial list randomly.

E.g.:
The question for which the answer options shall be generated:
Acte sous seing ___ : acte qui n'est pas rédigé par un officier public. Il ne bénéficie donc pas des avantages attachés à l'acte authentique.
Correct answer: *privé*

Other questions with answers from the same list (taken randomly):

Société, La ___civile immobilière permet de structurer et d'organiser l'acquisition et la gestion de biens immobiliers à plusieurs, et ainsi d'éviter les inconvénients d'une indivision.

Bail, Acte d'administration : acte de gestion courante du patrimoine, qui permet d'en conserver la valeur et de le faire fructifier (par exemple la conclusion d'un ___ d'habitation).

Régime, Le ___ de séparation des biens distingue deux types de biens : les biens propres de l'époux et ceux de l'épouse.

So, the generated question with answer options shown for a user will be:

Acte sous seing ___ : acte qui n'est pas rédigé par un officier public. Il ne bénéficie donc pas des avantages attachés à l'acte authentique.

Régime
Bail

Privé
Société

NB: Taking into account that some initial questions may have the same initial answer, answers taken from other initial question must not be the same as the correct answer for this question.
E.g:
Initial question: TMG = ___ / PA
Correct answer: MBU
Another initial question with the same answer MBU: ___ = PVHT - PA
So, this initial question must be excluded from the list of questions from which answer options can be taken from.

4.6.1.3 Question type Input generation

For the input type, only the initial question is shown for a user with an input field where the answer is typed.

%Initial question text%
%Input field for the answer%

The answer input by a user is compared with the initial correct answer.

The following rules are applied for checking the input answer:

- Letters register has not to be taken into account (no matter if an answer is given using capital letters or not).
- Spaces at the beginning and at the end of the input string have to be excluded and ignored.

E.g:
Question: *Insanité d'___ : altération des facultés mentales.*
Typed answer: *Esprit*
Correct answer given in the initial file: *esprit*
The typed answer started with a space and with a capital letter is correct.

4.6.2 Question show algorithm

Questions are shown according to their complexity levels:

- 1 - true / false (easy)
    - Questions with answers true and false are generated randomly (approx 50/50).
- 2 - MCQ (moderate)
    - select the correct answer among 4 variants
- 3 - inputs (hard)
    - no answer displayed; student types the expected answer.

To have a quiz complete, generated questions of each type shall be answered correctly 2 times in a row. The quiz is complete when each question from the initial file is "*learnt*" which means at least 2 correct answers for generated questions of each type were given.
For the flawless quiz completion, a user answer **`(number of initial questions) x3`** questions (e.g. for 20 questions in the initial file, at least 60 questions are generated during the quiz).

Questions are generated "on fly" from the initial list. That means, if the same question is asked several times, the suggested answers might be different because they are generated randomly according to the generation algorithm.
E.g. the correct answer is "Yes" for the 1st question show, and "No" for 2nd, etc.
For the MCQ, one set of answers is generated when a question is asked for the 1st time, another answer options are generated if this question is shown again, etc.

For each initial question, the following workaround is applied:

- An initial question is marked as "todo"
- 2 true/false questions in a row must be answered correctly to move forward to the MCQ questions.
- If a wrong answer is given for a TF answer, new TF questions (at least 2) are generated based on the same initial questions.
- After 2 correct answers in a row on TF questions, the user moves to the MCQ level where 2 correct answers in a row shall be given.
- If a user gives the wrong answer on the MCQ level, they are moved down to the TF level to give 2 correct answer in a row to move back to the MCQ questions.
- After giving 2 correct answers in a row for the MCQ level, the Input type of questions are shown for a user.
- The initial question is marked as "passed" when 2 correct answers for the input type of questions are given.
- If wrong answer is given, MCQ questions are shown where 2 correct answers in a row are expected. A user can be moved down to TF questions when wrong answer is given for these MCQ questions.

Questions generated from initial questions are shown randomly. The quiz is complete when all initial question are learnt by passing the flow 2TF->2MCQ->2input type questions answered correctly in a row.

Example of a question show based on 2 initial questions (Q1 and Q2):
Q1 TF1 - correct
Q2 TF1 - correct
Q1 TF2 - incorrect
Q2 TF2 - correct
Q2 MCQ1 - correct
Q1 TF3 - correct

Q2 MCQ2 - correct
Q1 TF4 - correct
Q1 MCQ1 - correct
Q2 input1 - incorrect
Q1 MCQ2 - correct
Q2 MCQ3 - correct
Q1 input1 - correct
Q2 MCQ4 - incorrect
Q1 input2 - correct - Question 1 is passed and learnt, no more questions based on this initial question will be shown for a user.
Q2 TF3 - correct
Q2 TF4 - correct
Q2 MCQ5 - correct
Q2 MCQ6 - correct
Q2 input2 - correct
Q2 input3 - correct - Question 2 is passed and learnt, no more questions based on this initial question will be shown for a user.

4.6.3 Score calculation algorithm

4.6.3.1 Points calculation

The maximum number of points gained for a quiz is:
(Number of questions) x 3

2 correct answers on TF questions in a row: +1 point given to the final quiz score (proceed with the MCQ questions then).
2 correct answers on MCQ questions in a row: +1 point given to the final quiz score (proceed with the input questions then).
2 correct answers on input questions in a row: +1 point given to the final quiz score, and the initial question is marked as "passed", no more questions are generated.
Incorrect answer is given on a TF question: points are not decreased.
Incorrect answer is given on a MCQ question: -1, the final score is decreased and a user proceeds with TF questions.
Incorrect answer is given on an input question: -1, the final score is decreased and a user proceeds with MCQ questions.

4.6.3.2 Grades calculation

Grades are always in the range between 0 (worst) and 20 (best).

The grade calculation formula:
(Number of points gained) / (Max number of points) * 20
The grade earned is rounded mathematically: when the result is 7.257, the grade earned is 7. When a result is 7.6834 the grade earned is 8.

If a quiz is not complete before the deadline, their grade is frozen when the deadline is approached (at 00:00). However, students can continue learning and passing quizzes but their grade won't be incremented.

Example:
The deadline is 05 Nov.
A student earned 21 points of 60 and the grade 7/20 to the date 04 Nov. At 00:00 05 Nov the student result is frozen (7/20), but they can continue passing a quiz without incrementing their grade.

4.7 Deep links workflow and generation

Deep links are expected for use for the following cases:

- Email verification on a new user registration
- Password reset
- Joining the new class
- Start a quiz tournament (future versions)

For the email verification cases or for the password reset, a link is sent on a user email. When a link is proceeded on a device with the Focus France app installed, it's opened and the corresponding action is performed (email is verified or new password is generated).

QR-codes for joining the class contain the URL-links to be opened in the mobile app. After following the deeplink from the QR-code, the action to sign up to a class is performed from the application (if installed).

Deep links can be generated in any 3rd party service (e.g. Firebase), and the following workflow is expected:

- When opened on the mobile device with the Focus France app installed, the corresponding app location is opened.
- When a link is opened on a device with no Focus France installed, the application page on the Google Play / Appstore is opened.
- All the deeplinks generated must be unique in order to lead to explicitly defined app locations.

Email verification is processed in a different way:

- When a deeplink is opened on a mobile device with the Focus France app installed, the corresponding location is opened (Your email is verified)
- When a deeplink is opened on a device without the Focus France app installed (mobile or desktop), the Focus France webpage is opened with the confirmation that the user's email is verified.