Requirements Document For Around The World In 80 Questions API

last updated :13.1.19

Table Of Contents:

Opening Page:

List Of Students:

1) Daniil Rolnik . ID: 334018009.

   Roles in the team: Product Manager, Product Owner, DevOps.

   Avatar: 

2) Elia Ben Anat . ID: 308048388.

   Role in the team: Database Administrator.

   Avatar: 

3) Eden Dupont . ID: 204808596.

   Role in the team: QA Engineer.

   Avatar: 

4) Eden Sharoni . ID: 315371906

   Role in the team: Scrum Master.

   Avatar: 

Course Information:

Course Name: Integrative Software Engineering
Project Name: Around The World In 80 Questions (ATW80Q - shortcut)

Project Introduction:

This project for the course Integrative Software Engineering, is about a quiz game which is intended to help teachers to check student's knowledge and expand the student's knowledge in various topics. Project will include two primary actors: Manager(teacher) and Player(student). Manager is able to edit or to upload elements. Player is able to perform Activities. Activities and Elements will be defined further in the requirements document.

Game Description:

Question will be displayed, the user fills his answer and the game tells him if he was right or wrong.

Vision :

We aim to have biggest database of questions and answers in various topics, which can be used for applications and websites easily, we will be recognised as best education platform for students.

Purpose of the system:

The system can be used as a game or a teaching tool, Manager can add questions and send the question to anyone for them to answer.

Business purpose of the System:
All the features in the games are under license, the game will be licensed yearly by schools.

Scope

1. In Scope:

The system will have a limited amount of stored questions.
The player can exit the program mid-game.
The system will allow skipping questions.
The system will hold verbal answer based questions in the database
Manager actor will have the ability to add their own questions.
Player will receive points depending on how much correct answers he had

2. Out of Scope:

The system will allow payments.
The system will automatically save the score.
The system will allow advertisements.
The player can select a different language.
The player can add questions.
The system will set timer on every question.
The player can easily customize UI of the service
The game will ask non-verbal questions. (pictures)
The system will return to the point where the player exited.

Current + Conclusion:

Currently, there are other games of this type on the web, but there is none for educational institutes which can be tailored for the teacher (Manager actor) such as “World Geography Game” for example.
Our competitors (www.quiz-tree.com) are not specific enough for the teacher’s (Manager actor) needs.
For example, adding questions and answers to the database.
In conclusion there is a need for a platform that can provide education in a fun and interactive way.

Actors and goals:

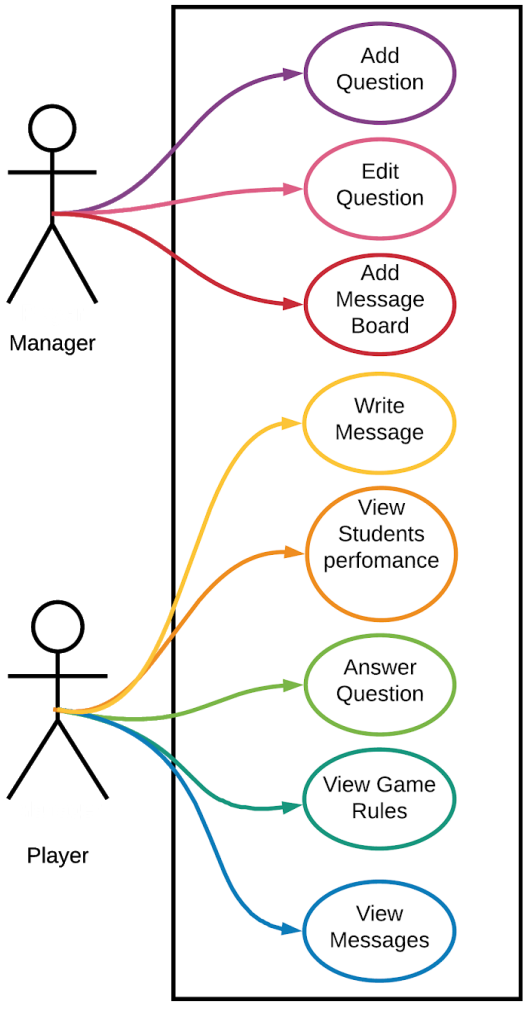
Actor	Primary/Secondary	Description	Goals
Player	Primary	A user that wants to improve their knowledge in history, geography etc.	Improve knowledge, do interactive homework, study for tests.
Manager	Primary/Secondary (depends on the use case)	A user that wants to update questions database or view their player's performance.	Improve ATW80Q, create more personal experience for players. Review how users use the ATW80Q.

Stakeholders:

Name	Internal/ External	Description	Goal	Connection to the project	Supportive/ Opposed
Customer (private and public schools)	External	Wants to educate players about different subjects .	give knowledge to players	Utilizes the tool to teach	Supportive
Creditors (none profit)	External	Loaned money to the project	Their objective is to get their loan returned with interest.	No direct connection.	Supportive
Employees	Internal	Anyone who contributes work to the project with a salary or payment.	Job satisfaction, salary, job security, motivation and self-actualization.	Perform tasks the manager orders them to do.	Supportive
Government	External	Politicians, regulatory authorities	Get taxes, everything should be according to the law. Potential customer for the product for public schools	Regulator agent checks that the business meets the terms of the law	Supportive
Owners	Internal	People who own the company, may be investors	Interested in how much profit the business makes to get profits.	Owner, may make direct decisions.	Supportive
Competitors	External	Companies that provide similar services	Wants to get the biggest share of market.	Influences our profits and the decisions we make.	Opposed

Use Case Diagram + Use Cases

1. Use Case Diagram



2. Use Cases:

Use case	**Goal**
Add question	**To give the player access to question**
Edit Question	**To correct a question in the database**
Add Message Board	**To allow players to contact with the manager**
View Students performance	**To see the status of the high score players**
Write message	**To send a message to the manager**
Answer Question	**To the player to interact with the game**
View Game Rules	**To understand the rules of the game**
View Messages	**To see other player messages**

ATW80Q API

API definitions:

1. User - Player/Manager that registered and performed verification through API
2. Element - Elements which a user can access and execute activities on
3. Activity - An activity the user wants to execute on the system. Only user of type player user may do this in our system.

Supported Types of elements and their attributes

- **DEFAULT_TYPE**
- **ELEMENT_QUESTION_TYPE**

  Name of element property is the question title
  ELEMENT_QUESTION_KEY - Question body
  ELEMENT_QUESTION_KEY - Answer to question
  ELEMENT_POINT_KEY - point value of question

- **ELEMENT_MESSAGEBOARD_TYPE**

  MESSAGEBOARD_MESSAGE_COUNT - Number of messages on the board.
  Messages' superkeys are also in the attributes

Supported Activity types

- **GET_SCORES_ACTIVITY**
- **GET_MESSAGES_ACTIVITY**
  holds in ElementId property the superkey of the messageboard it wants to retrieve messages from.
- **MESSAGE_ACTIVITY**
  - ElementId property holds the messageboard superkey of the messageboard owner
    ACTIVITY_MESSAGE_KEY
- **QUESTION_READ_ACTIVITY**
    ElementId property holds the superkey for the question
- **QUESTION_ANSWER_ACTIVITY**
    ElementId property holds the superkey for the question
    ACTIVITY_USER_ANSWER_KEY - holds the answer the user's answer to the question

- **GET_GAME_RULES_ACTIVITY**

API Urls:

	HTTP Method	URL	input	Output
1	POST	/playground/users	NewUserForm	UserTO
2	GET	/playground/users/confirm/{playground}/{email}/{code}	--	UserTO
3	GET	/playground/users/login/{playground}/{email}	--	UserTO
4	PUT	/playground/users/{playground}/{email}	UserTO	--
5	POST	/playground/elements/{userPlayground }/{email}	ElementTO	ElementTO
6	PUT	/playground/elements/{userPlayground}/{email}/{playground}/{id}	ElementTO	--
7	GET	/playground/elements/{userPlayground}/{email}/{playground}/{id}	--	ElementTO
8	GET	/playground/elements/{userPlayground}/{email}/all	--	ElementTO[]
9	GET	/playground/elements/{userPlayground}/{email}/near/{x}/{y}/{distance}	--	ElementTO[]
10	GET	/playground/elements/{userPlayground}/{email}/search/{attributeName}/{value}	--	ElementTO[]
11	POST	/playground/activities/{userPlayground}/{email}	ActivityTO	Object

1) Register new user using NewUserForm
2) Confirm User with verification code
3) Check that verification code entered in url #2 was correct and from now one user is logged in
4) Update user information
5) Save Element
6) Update element
7) Get element by id
8) Get all elements
9) Get elements near (x,y) location
10) Get all elements with attribute name and attribute value
11) Send activity to server and execute

Workflows

(Basic Flows and Alternate Flows)

1) Add Question

Basic Flow

Manager	System
Performs registration	
	Performs registration on Manager
Performs verification	
	Performs verification on Manager
Chooses to add question using url #5	
	Adds question to database

Alternate Flow

Player	System
Performs registration	
	Performs registration on Player
Performs verification	
	Performs verification on Player
Chooses to add question using add question activity and using url #5	
	System throws exception (request from none manager)

2) Update Question

Basic Flow

Manager	System
Performs registration	
	Performs registration on Manager
Performs verification	
	Performs verification on Manager
Chooses to update existing question using url #6	
	Updates existing question in database

Alternate Flow

Manager	System
Performs registration	
	Performs registration on Manager
Performs verification	
	Performs verification on Manager
Chooses to update non existing question using url #6	
	System throws exception

3) Add Message Board

Basic Flow

Manager	System
Performs registration	
	Performs registration on Manager
Performs verification	
	Performs verification on Manager
Chooses to add new message board using url #5	
	Adds new message board

Alternate Flow

Player	System
Performs registration	
	Performs registration on Manager
Performs verification	
	Performs verification on Manager
Chooses to add new message board using url #5	
	System throws exception (request from none manager)

4) View Students Performance

Basic Flow

Player	System
Performs registration	
	Performs registration on Player
Performs verification	
	Performs verification on Player
Chooses to view students performance using url #11	
	returns students performance to Player

Alternate Flow

Player	System
Performs registration	
	Performs registration on Player
Performs verification	
	Performs verification on Player
Chooses to view students performance using url #11	
	There is no performance results to return

5) Write Message

Basic Flow

Player	System
Performs registration	
	Performs registration on Player
Performs verification	
	Performs verification on Player
Chooses to add new message using url #11	
	Adds new message board

Alternate Flow

Manager	System
Performs registration	
	Performs registration on Manager
Performs verification	
	Performs verification on Manager
Chooses to add new message using url #11	
	System throws exception (request from none player)

6) Answer Question

Basic Flow

Player	System
Performs registration	
	Performs registration on Player
Performs verification	
	Performs verification on Player
Gives correct answer using element question id and url #11	
	Adds points to Player

Alternate Flow

Player	System
Performs registration	
	Performs registration on Player
Performs verification	
	Performs verification on Player
Gives not correct answer using element question id and url #11	
	Nothing happens

7) View Game Rules

Basic Flow

Player	System
Performs registration	
	Performs registration on Player
Performs verification	
	Performs verification on Player
Chooses to view game rules using view game rules activity and using url #11	
	Returns game rules as string

Alternate Flow

Player	System
Performs registration	
	Performs registration on Player
Performs verification	
	User not verified
	Throws exception

8) View Messages

Basic Flow

Player	System
Performs registration	
	Performs registration on Player
Performs verification	
	Performs verification on Player
Chooses to view messages using message board id and using url #11	
	returns all messages in chosen message board

Alternate Flow

Manager	System
Performs registration	
	Performs registration on Manager
Performs verification	
	Performs verification on Manager
Chooses to view messages using message board id and using url #11	
	System throws exception (request from none player)

Non-functional requirements:

sn	Requirement and Description	Requirement Type (U/R/P/S)
1	Adding a question will take no more than 3 seconds	FURPS: Performance
2	The server will be easy to translate to other languages (localizability)	FURPS: Supportability
3	list of the best players with high score will be limited by pagination from the whole playground	FURPS: Usability
4	The DataBase we use is MySQL .	FURPS: Supportability
5	The program is accessible to people with eyesight problems	FURPS: Supportability

User Stories:

AS a	I Want To	So That
Manager	add question to session	i could build a program for the class
Manager	moderate en existing question	I could change any mistake or fault in the question or answer
Manager	get high scores of my students	could understand the general status of my class
Manager	write message on the message board	my students will be informed of news regarding class
Manager	read messages from the message board	I will be able to communicate with my students
Manager	create different message board	I could separate different subjects in different boards
Player	answer questions	I could increase my score and learn the subject in class
Player	write message on the message board	I could inform my teacher about any problem or information i would like to shere
Player	read messages from the message board	get information from my teacher regarding class

List of used technologies:

1) Spring v2.1.1
    a. Spring Boot
    b. Spring Web
    c. Spring Data
2) RESTful Web Service with Spring Web
3) jdk 1.8
4) JUnit
5) Eclipse IDE
6) Github repository
7) Hibernate
8) JPA
9) Maven
10) MySql
11) AspectJ
12) Java Swing
13) Boot starter mail
14) H2