Testing

Testing aims at assuring the functionalities of the product fit our client's expectation and the product being delivered is of high quality.

Refer to the following pages for details of our test procedure, and test results:

Back-End Unit Testing/Integration Testing

Back-End Test Plan

Goal:

Testing of the features and overall performance on the Back-End side. The elements to test are server and database.

Post-condition:

Report to the back-end lead about the test results. Approve a pull request to the develop branch if the test is successful.

Testing framework:

Jest framework is used for the unit testing of the software. The information on how to use the framework can be found here.

Class	API	Method	Pre-Route	Route
game-controller.js	newGame	post	/play	/new
	joinGame	post		/join/:gameID
	getPlayerStatus	get		/wait/:gameID
	getGameStatus	get		/:gameID/status/
	getBackground	get		/background/:gameID
	getRolesInfo	get		/:gameID/roles/
	showAllRoles	get		/choose-role/:gameID
	chooseRoles	post		/choose-role/:gameID
	fetchDecision	get		/make-decision/:gameID
	makeDecision	post		/make-decision/:gameID
	readyToDiscuss	post		/:gameID/start-discussion/
	getRoundDescription	get		/:gameID/round/
	getRoundStatus	get		/:gameID/round-status
	getRoundOutcome	get		/:gameID/round-outcome
option-controller.js	getOption	get	/options	/
	newAllOptions	post		/
	dropOption	post		/drop
question-controller.js	getQuestion	get	/questions	/
	newQuestion	post		/
	dropQuestion	post		/drop
roles.js	getRole	get	/roles	/
	newRole	post		/
	getRolesIntro	get		/introduction

Back-End Test Results

Test suite	Test case - functionality	Sprint 3 Results (Pass/Fail)	Sprint 4 Results (Pass/Fail)
game_model.test.js	create & save game successfully	Pass	Pass
	insert game successfully	Pass	Pass
play.test.js	connect to server	Pass	Pass
	insert options	Pass	Pass
	create new game	Pass	Pass
	choose available role	Pass	Pass

	choose unavailable role	Pass	Pass
	make decision	Pass	Pass
	get role information	Pass	Pass
	get outcome	Pass	Pass
role_model.test.js	create & save role successfully	Pass	Pass
	insert role successfully	Pass	Pass
	create & save user successfully	Pass	Pass
	insert user successfully	Pass	Pass

Front-End Unit Testing

Test results

ID	Test Goal	Test Suite Path	Test Result	Code Status
1	check whether title and button list classes exist	Home.spec.js	pass	uploaded
2	check if home page can render first title correctly	Home.spec.js	pass	uploaded
3	check if home page can render second title correctly	Home.spec.js	pass	uploaded
4	check if home page can render New Game button and it is clickable	Home.spec.js	pass	uploaded
5	check if home page can render Join Game button and it is clickable	Home.spec.js	pass	uploaded
6	check if home page can render Help button	Home.spec.js	pass	uploaded
7	check if home page can render Help button content correctly	Home.spec.js	pass	uploaded
8	check if home page can render Help Got it button and check whether it is clickable	Home.spec.js	pass	uploaded
9	check whether title and form classes exists	JoinGame.spec.js	pass	uploaded
10	check if JoinGame page can render main title correctly	JoinGame.spec.js	pass	uploaded
11	check if JoinGame page can render submit button correctly	JoinGame.spec.js	pass	uploaded
12	check if JoinGame page can display a left arrow button and it is clickable	JoinGame.spec.js	pass	uploaded
13	check whether title and form classes exists	NewGame.spec.js	pass	uploaded
14	check if NewGame page can render main title correctly	NewGame.spec.js	pass	uploaded
15	check if NewGame page can render submit button correctly	NewGame.spec.js	pass	uploaded
16	check if NewGame page can display a left arrow button and it is clickable	NewGame.spec.js	pass	uploaded
17	check whether title and button list classes exists	ChooseYourRole.spec.js	pass	uploaded
18	check if ChooseYourRole page can render main title correctly	ChooseYourRole.spec.js	pass	uploaded
19	check if ChooseYourRole page can render Boeing Executive button correctly	ChooseYourRole.spec.js	pass	has been removed because of updating code
20	check if ChooseYourRole page can render Aeronautical Engineer button correctly	ChooseYourRole.spec.js	pass	has been removed because of updating code
21	check if ChooseYourRole page can render Software Developer button correctly	ChooseYourRole.spec.js	pass	has been removed because of updating code
22	check if ChooseYourRole page can render Pilot button correctly	ChooseYourRole.spec.js	pass	has been removed because of updating code
23	check if ChooseYourRole page can render FAA Official button correctly	ChooseYourRole.spec.js	pass	has been removed because of updating code
24	check whether ChooseYourRole page title render correctly and make sure its class exists	ChooseYourRole.spec.js	pass	uploaded
25	check ChooseYourRole page can get role object and check the backend role name and its description are correctly	ChooseYourRole.spec.js	pass	uploaded
26	check Next button going to the chapterBackground page	ChooseYourRole.spec.js	pass	uploaded
27	check whether Background page title render correctly and make sure its class exists	BackGround.spec.js	pass	uploaded
28	check Background page can get correct data from backend	BackGround.spec.js	pass	uploaded
29	check Next button going to the Choose Your role page	BackGround.spec.js	pass	uploaded
30	check whether ChapterBackground page title render correctly and make sure its class exists	ChapterBackground.spec.js	pass	uploaded
31	check Next button going to the IndividualBackground page	ChapterBackground.spec.js	pass	uploaded
32	check whether IndividualBackground page title render correctly and make sure its class exists	IndividualBackground.spec.js	pass	uploaded
33	check Next button going to the Discussion page	IndividualBackground.spec.js	pass	uploaded

34	check whether RoleBackground page title render correctly and make sure its class exists	RoleBackground.spec.js	pass	uploaded
35	check Next button going to the ChapterBackground page	RoleBackground.spec.js	pass	uploaded
36	check whether Waiting page title render correctly and make sure its class exists	Waiting.spec.js	pass	uploaded
37	check whether Waiting page can render correctly guide to end users	Waiting.spec.js	pass	uploaded
38	check Next button going to the background page	Waiting.spec.js	pass	uploaded
39	check if menu button gp to IndividualBackground page	Discussion.spec.js	pass	uploaded
40	check if discussion information can render correctly	Discussion.spec.js	pass	uploaded
41	check whether Discussion Content class show correctly	Discussion.spec.js	pass	uploaded
42	check whether title and class of Discussion page shows correctly	Discussion.spec.js	pass	uploaded
43	check whether title of drawer of the game menu content show correctly	Outcome.spec.js	pass	uploaded
44	check whether GameOutcome Text class show correctly	Outcome.spec.js	pass	uploaded
45	check whether title of drawer of the game menu content show correctly2	Outcome.spec.js	pass	uploaded
46	check if RoleOutcome Text class can show correctly	Outcome.spec.js	pass	uploaded
47	check if Outcome page router works well	Outcome.spec.js	pass	uploaded
48	check whether title and content exists	Ready.spec.js	pass	uploaded
49	check whether title and content exists	Ready.spec.js	pass	uploaded
50	check whether title and class of reflection page shows correctly	Reflection.spec.js	pass	uploaded
51	check whether All Decisions Table class show correctly	Reflection.spec.js	pass	uploaded
52	check if exit game button works well	Reflection.spec.js	pass	uploaded

Acceptance Testing

Detailed test table and results

User Story ID	User story	Acceptance Criteria ID	Given	When	Then	Acceptance Test ID	Test*	Expected Result
US_01	As a teacher I want to be relatively simple for the decision-making process so the game is fast-paced, which should improve student engagement and outcomes.	AC_01	The teacher control the game time.	The student Plays the game	The game is fast-paced, which should improve student engagement and outcomes.	AT_01	Play the whole game and test the gaming time.	The game process is in 20 minutes.
US_02	As a student I want to submit my option successfully so I can play the game without getting stuck.	AC_02	The student submit the option.	Submit the option of each questions	The game continues to next question fluently without stuck.	AT_02	All Users click the submit bottom in question page.	Next question page is displayed
US_03	As a student I want to get a overall report that shows decisions made by other users for each questions throughout the game so I can clearly see whether or not I achieved my character goals or not, and also reflect upon whether personal success, if attained, came at any cost to plot outcomes.	AC_03	The student submit the option.	Submit the option of each questions.	The student can see overall report of other players submitted.	AT_03	All Users click the submit bottom in question page.	Display whether game characters goals achieved and the overall report of other users decision for the question.
US_04	As a student/teacher I want to be able to rejoin the game session so I can continue the game if it is interrupted.	AC_04	The student want to rejoin the game.	The game interrupted and return to the index page.	The game can continue and locate the latest process.	AT_04	Quit the game and click the rejoin bottom to rejoin the game in the home page.	Rejoin the game and transform to the latest game process.
US_05	As a student I want to be able to see all chosen decisions made for all of my personal and group decisions. so I can get some insight into what other group members decided, and recall any forgotten decisions made earlier in the game.	AC_05	The student sees all the options chosen.	The student reaches the outcome page.	The student can get some insight into what other group members decided, and recall any forgotten decisions made earlier in the game.	AT_05	Submit the final question and reach the outcome page	All chosen decisions are shown in the outcome page
US_06	As a teacher I want to add a timer to limit the discussion time for each decision in game so I can make students put more focus on the contradictive information and force them to make decisions, which will improve their critical thinking of whether their decisions are right to do.	AC_06	The student checks how much time they left to answer the questions.	The student answers the questions.	The student puts more focus on the contradictive information and forced to make decisions in limited time.	AT_06	Enter the questions pages	A timer is working and shows how many times left

Bug Report

List of bugs

ID	Description	Fixed
1	The backend server is not available by the frontend. The program can not go any further after users put in their names and click on the start button.	Yes
2	The database is null after the server is launched. When a query of questions is sent at the beginning of the game, the backend program crashes immediately, leaving frontend website stuck at the waiting page.	Yes
3	After the last question is answered by every user, the program fails to give an outcome as expected and leaves the webpage stuck at the waiting status.	Yes
4	At the end of a round when everyone has answered the question and the program is expected to switch to the next question, there is a chance that the backend stuck in the waiting process.	No