Testing Report for Erato-UoM-Backend

COMP90082: Project Group ER-1

COMP90082-2021-SM1-ER Project Github:
https://github.com/AVK95/Eratos-UoM-Backend.git
COMP90082-2021-SM1-ER Project Confluence:
https://confluence.cis.unimelb.edu.au:8443/display/COMP900822021SM1 ER

In charge of: En-Wen Tsai <tsaie@student.unimelb.edu.au>

Proposal:

The purpose of this document is to define and present the test cases for Eratos website, covering the test cases for all user stories.

Target users:

This document is mainly designed for those responsible for executing the test cases for back-end in this project.

Functional Requirements Summary:

User Story	User Story Title	Test ID	Test Case Title
1	Get all active modules	1	User front-end is able to get all active modules
2	Create tasks	2	User front-end is able to create a task
3	Get task information	3	User front-end is able to get the task information
4	Get order history and status	4	User/Admin front-end is able to get order information
5	Register new users	5	User/Admin front-end is able to register as a new user
6	Get user information and identity	6.1	User front-end is able to the get user information
		6.2	Admin front-end is able to the get all user information
7	Update user information and identity	7	User/Admin front-end is able to update their user information

8	Remove users with all related information	8	User/Admin front-end is able to remove the their user information
9	sync the back-end server with the Eratos server	9	User/Admin front-end is able to sync task/order information with server
10	Get all orders	10	Admin front-end is able to get all order information
11	Get all tasks	11	Admin front-end is able to get the task information
12	Remove tasks	12	Admin front-end is able to remove the specific task
13	Remove orders	13	Admin front-end is able to remove the specific order
14	Create and modify modules	14.1	Admin front-end is able to create a resource/modules
		14.2	Admin front-end is able to modify a resource/modules
15	Get all modules	15	Admin front-end is able to get all modules
16	Remove modules	16	Admin front-end is able to remove the specific module

Test case for Story 1:

This test case is only for the user front-end, before creating a task, users need to check all active modules which are the acceptable modules or not.

Test ID		1
Test Case Title		User front-end is able to get all active modules
Test Type		Functional
Objective		Verify that user front-end is able to get all active modules
Pre-conditions		
Steps		1. Send the request to the server
Expected Outcome		Output the success message with a string including EDescription, ModuleID, ModuleName, ModuleSchema, isActive. Each module is separated by ","
Acceptance Criteria	Given	Make the request
	When	I parse the request to the Eratos server
	Then	I should get the list of all active modules.
Test result		PASS
Notes		[1] Sample: https://eratosuombackend.azurewebsites.net/api/getActiveModules
Tester		En-Wen TSAI
Date		05/06/2021

Test case for Story 2:

This test case is for the user front-end to create a task for the back-end to process their requests.

Test ID		2
Test Case Title		User front-end is able to create a task
Test Type		Functional
Objective		Verify that the user front-end is able to create a task after finishing the payment.
Precondition		1. Get their user uri 2. Get the resource uri 3. Get the resource name 4. Get the payment ID & price from front-end 5. Get the geometry info from front-end
Steps		1. Take priority to this task 2. Send the request with these parameters
Expected Outcome		Output success message with TaskID and ResourceID
Acceptance Criteria	Given	Received user's data
	When	I create a task based on provided data
	Then	I should get the success message.
Test result		PASS
Notes		[1] Sample (use "Postman" to do the post request): https://eratosuombackend.azurewebsites.net/api/createTask? &userUri=https://staging.e-pn.io/users/xsrudeulfvtr2eiaiksic2jf&paymentID=https://staging.e-pn.io/users/xsrudeulfvtr2eiaiksic2jf2021-05-28 00:18:14&price=69.66&moduleType=https://schemas.eratos.ai/json/utas.climateinfo&taskType=GenerateClimateInfo&name=testing&geometry=POINT(140.3142799 -42.84756651)&priority=low&code=pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JXfd9itviGQ==
Tester		En-Wen TSAI
Date		04/06/2021

Test cases for Story 3:

This test case is for the user front-end. User front-end is able to get task status back to users. Both need to sync the latest status with the database and then they can get the latest task information.

story ID		3
Test title		User front-end is able to get the task information
Test Type		Functional
Objective		Verify that the users can get the latest task status.
Precondition		1. Call sync task status API (Story 9) 2. Get the user uri
Steps		1. Find the task information by user uri
Expected Outcome		Output the success message with a string including TaskID, EratosTaskID, CreateAt, LastUpdatedAt, StartedAt, EndedAt, Name, Priority, State, Type, Meta, Error, UserID, OrderID. Each task is separated by ","
Acceptance Criteria	Given	User uri
	When	I use the user uri as parameter to find task information
	Then	I can get the string of all tasks information.
Test result		PASS
Notes		[1] Sample: https://eratosuombackend.azurewebsites.net/api/getTasksOrdersOfUser?userUri=demouser3&code=pC04Se3EKKG7ls6PPO3az2ldtzg6gJEFrTVbvpOIKh0JXfd9itviGQ==
Tester		En-Wen TSAI
Date		04/06/2021

Test case for Story 4:

This test case is for the user & admin front-end to get the order history back to users. Both need to sync the latest status with the database and then they can get the latest order information.

Test ID	4
Test Case Title	User/admin front-end is able to get the order history
Test Type	Functional
Objective	Verify that the users can get the latest order history.
Precondition	1. Call sync order status API (Story 9) 2. Get the user uri
Steps	1. Find the order information by user uri
Expected Outcome	Output the success message with a string including OrderID, OrderTime, PaymentID, Price, Status, UserID. Each task is separated by ","
Acceptance Criteria	Given	User uri
	When	I use the user uri as parameter to find order information in the database
	Then	I can get the string of all order information.
Test result	PASS
Notes	[1] Sample: https://eratosuombackend.azurewebsites.net/api/getTasksOrdersOfUser?userUri=demouser3&code=pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JXfd9itviGQ==
Tester	En-Wen TSAI
Date	04/06/2021

Test case for Story 5:

This case is for the user & admin front-end, back-end needs to store their user data for the further operation because we cannot get the data from Eratos directly.

Test ID		5
Test Case Title		User/admin front-end is able to register as a new user
Test Type		Functional
Objective		Register new users for our back-end database
precondition		1. Get user data from front-end
steps		1. Insert these data to the back-end database.
Expected Outcome		Output the success message with username.
Acceptance Criteria	Given	Received user data from front end
	When	I get the user info from the front-end and then insert it to our database.
	Then	I should get the successful message.
Test result		PASS
Notes		[1] Sample payload: {auth0Id:"auth0|60585179790525006901be39",id:"https://staging.e-pn.io/users/xsrudeulfvtr2eiaiksic2jf", info: "https://staging.e-pn.io/resources/gu4km2nnimvyqepmaw3mfcwg",resourcePolicy:"https://staging.e-pn.io/policies/3ylpwlaj6o3wb7p4ui4kkeie", termLastAcceptedDate:"2021-03-25T03:13:39.912183Z",termsLastAcceptedVersion: 12}
Tester		En-Wen TSAI
Date		04/06/2021

Test cases for Story 6:

These cases are for the user & admin front-end, user front-end can get their user information and admin front-end can access all user information. There are two sub-cases, one is for users to be able to get their user information and the other is admin can get all user information.

Test ID	6.1	
Test Case Title	User front-end is able to get the user information	
Test Type	Functional	
Objective	Verify that user front-end can get their user details from the database	
Precondition	1. Get the user uri	
Steps	1. Query with user uri to their user information.	
Expected Outcome	Output the success message with a string including EratosUserID, Email, Auth0ID, UserID, and name.	
Acceptance Criteria	Given	Login to get user uri
	When	I query with valid user uri to the database
	Then	I should get the message about the user details.
Test result	PASS	
Notes	[1] Sample: https://eratosuombackend.azurewebsites.net/api/getUserInfo?userUri=demouser3&code=pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JXfd9itviGQ==	
Tester	En-Wen TSAI	
Date	04/06/2021	

Test ID	6.2	
Test Case Title	Admin front-end is able to get all user information	
Test Type	Functional	
Objective	Admin front-end can get all user information	
Precondition	1. Start index of user table 2. End index of user table 3. The range between start and end must be less than 100.	
Steps	1. Query with start point and end point of the user ids.	
Expected Outcome	Output the success message with a string including eratosUserId, email, Auth0ID, UserID, and name. Each user is separated by ",".	
Acceptance	Given	Put the given start and end point

Criteria	When	I send the query to the database
	Then	I should get the message about specific user details.
Test result		PASS
Notes		[1] If an admin wants to get more than a hundred user information, he/she should query more than one time. [2] Sample: https://eratosuombackend.azurewebsites.net/api/getUserInfo?userUri=all&start=1&end=10&code=pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JXfd9itviGQ==
Tester		En-Wen TSAI
Date		04/06/2021

Test case for Story 7:

This case is for the user & admin front-end, both can update their own information in the database. However, only the admin can edit the isAdmin value.

Test ID		7
Test Case Title		User/admin front-end is able to update their user information
Test Type		Functional
Objective		Users and admins can update user information
Precondition		1. Had already registered as an user/admin.
Steps		1. Make some changes in their personal information. ex: name, email, info or isAdmin.
Expected Outcome		Output the success message "The user info is up to date.".
Acceptance Criteria	Given	Make some changes
	When	I update these changes to the database
	Then	I should get the success message.
Test result		PASS
Notes		[1] Only admins can change the isAdmin parameter to true or false. [2] Sample payload: {"Auth0ID":"demoauth0","CreatedAt": "5\/17\/2021 8:11:33 PM","Email":"demoemail","EratosUserID": "demouser","Info": "ModifiedDemoInfo", "Name": "demoname", "UserID":1, "isAdmin":false}
Tester		En-Wen TSAI
Date		04/06/2021

Test case for Story 8:

This case is for the user & admin front-end, both can delete their own information in the back-end database.

Test ID		8
Test Case Title		User/admin front-end is able to remove user
Test Type		Functional
Objective		Verify that the user/admin front-end is able to remove the existing user.
precondition		1. User had already existed in the database
Steps		1. Get the user ID
Expected Outcome		Get the success message
Acceptance Criteria	Given	Put user ID as parameter
	When	I send this parameter to the server
	Then	I should get the success message from the server
Test result		PASS
Notes		[1] Sample: https://eratosuombackend.azurewebsites.net/api/removeUser?userID=2&code =pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JX fd9itviGQ==
Tester		En-Wen TSAI
Date		03/06/2021

Test case for Story 9:

This case is for the user & admin front-end, both are able to update the back-end server by fetching the tasks and orders info and syncing the two servers.

Test ID		9
Test Case Title		User/Admin front-end is able to sync task/order information with servers.
Test Type		Functional
Objective		Verify that the user/admin front-end can sync the back-end server by fetching the tasks and orders info with Eratos server.
Precondition		1. The task status is incomplete (Queued or Processing) 2. The order status is incomplete (Queued or Processing)
Steps		1. Connect to back-end database 2. Check the task status is "Queued" or "Processing" 3. Check the order status is "Queued" or "Processing" 4. Send the request to Eratos server.
Expected Outcome		Output the success message "Database is up to date"
Acceptance Criteria	Given	Check the order status and connect to the database
	When	I send the request to Eratos server
	Then	I can get the success message.
Test result		PASS
Notes		[1] Sample: https://eratosuombackend.azurewebsites.net/api/syncTasksAndOrders
Tester		En-Wen TSAI
Date		05/06/2021

Test case for Story 10:

This case is only for the admin front-end, admin front-end is able to get all order information for a range between one to hundred.

Test ID	10
Test Case Title	Admin front-end is able to get all order information
Test Type	Functional
Objective	Verify that the admin can get all the latest order information.
Precondition	1. Start index of order id 2. End index of order id 3. The range between start and end must be less than 100.
Steps	1. Query with start point and end point of order id
Expected Outcome	Output the success message with a string including OrderID, OrderTime, PaymentID, Price, Status, UserID. Each order is separated by ","

Acceptance Criteria	Given	Put the given start and end point
	When	I send the query to the database
	Then	I should get the message about orders' details.

Test result	PASS
Notes	[1] If an admin wants to get more than a hundred order information, he/she should query more than one time. [2] Sample: https://eratosuombackend.azurewebsites.net/api/getAllOrders?start=1&end=10&code=pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JXfd9itviGQ==
Tester	En-Wen TSAI
Date	04/06/2021

Test case for Story 11:

This case is only for the admin front-end, admin front-end is able to get all task information for a range between one to hundred.

Test ID		11
Test Case Title		Admin front-end is able to get all task information
Test Type		Functional
Objective		Verify that the admin can get all the latest task information.
Precondition		1. Start index of task id 2. End index of task id 3. The range between start and end must be less than 100.
Steps		1. Query with start point and end point of task id
Expected Outcome		Output the success message with a string including EratosTaskID, Error, LastUpdatedAt, Meta, Name, OrderID, Priority, StartedAt, State, TaskID, Type, UserID. Each task is separated by ","
Acceptance Criteria	Given	Put the given start and end point
	When	I send the query to the database
	Then	I should get the message about tasks' details.
Test result		PASS
Notes		[1] If an admin wants to get more than a hundred task information, he/she should query more than one time. [2] Sample: https://eratosuombackend.azurewebsites.net/api/getAllTasks?start=1&end=10&code=pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JXfd9itviGQ==
Tester		En-Wen TSAI
Date		04/06/2021

Test case for Story 12:

This case is only for the admin front-end, admin front-end is able to remove the specific order information based on their needs.

Test ID		12
Test Case Title		Admin front-end is able to remove the specific task
Test Type		Functional
Objective		Verify that the admin front-end is able to remove the existing task.
precondition		1. User had already created this task
Steps		1. Get the Task ID
Expected Outcome		Get the success message "Task removed successfully. Updated order information is as following"
Acceptance Criteria	Given	Put Task ID as parameter
	When	I send this parameter to the server
	Then	I should get the success message from the server
Test result		PASS
Notes		[1] Sample: https://eratosuombackend.azurewebsites.net/api/removeTask?taskID=2&code =pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JX fd9itviGQ== [2] when the task has been removed, the status of the related order will be update as "deleted"
Tester		En-Wen TSAI
Date		03/06/2021

Test case for Story 13:

This case is only for the admin front-end, admin front-end is able to remove the specific order information based on their needs.

Test ID		13
Test Case Title		Admin front-end is able to remove the specific order
Test Type		Functional
Objective		Verify that admin front-end is able to remove the order
precondition		1. Call remove task API (Story 12)
Steps		1. Get this order ID
Expected Outcome		Get the success message
Acceptance Criteria	Given	Put order ID as parameter
	When	I send this parameter to the server
	Then	I should get the success message from the server
Test result		PASS
Notes		[1] Sample: https://eratosuombackend.azurewebsites.net/api/removeOrder?orderID=2&code =pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JX fd9itviGQ==
Tester		En-Wen TSAI
Date		03/06/2021

Test cases for Story 14:

These cases are only for the admin front-end, admin is able to add new modules, modify modules or toggle them on and off based on their needs. There are two cases, one is for admins to create new modules and the other is for admins to modify modules including activating or deactivating them.

Test ID	14.1
Test Case Title	Admin front-end is able to create a module
Test Type	Functional
Objective	Verify that admins are able to create a new module for users.
precondition	
Steps	1. Create a module name 2. Create a module schema 3. Make this module is active or not 4. Add description of the module
Expected Outcome	Get the success message
Acceptance Criteria	Given	Put needed parameter
	When	I send these parameters to the server
	Then	I should get the success message from the server
Test result	PASS
Notes	[1] Sample (use "Postman" to do the post request): https://eratosuombackend.azurewebsites.net/api/createModifyModule?moduleSchema=https://schemas.eratos.ai/json/person&moduleName=Person&isActive=false&Description=testing&code=pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JXfd9itviGQ==
Tester	En-Wen TSAI
Date	04/06/2021

Test ID	14.2
Test Case Title	Admin front-end is able to modify a module
Test Type	Functional
Objective	Verify that admins are able to modify the existing module.
precondition	1. Admin had already created a module
Steps	1. Change the module name 2. Change the module schema 3. Change the module status to inactive. 4. Change the content of module's description
Expected Outcome	Get the success message

Acceptance Criteria	Given	Put changed parameters
	When	I send these parameters to the server
	Then	I should get the success message from the server
Test result		PASS
Notes		[1] Sample (use "Postman" to do the post request): https://eratosuombackend.azurewebsites.net/api/createModifyModule?moduleSchema=https://schemas.eratos.ai/json/person&moduleName=Person&isActive=false&Description=testing&code=pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JXfd9itviGQ==
Tester		En-Wen TSAI
Date		04/06/2021

Test case for Story 15:

This case is only for the admin front-end, admin front-end is able to get modules for a range between one to hundred.

Test ID		15
Test Case Title		Admin front-end is able to get all modules
Test Type		Functional
Objective		Verify that admin front-end is able to get the all modules
Precondition		1. Start index of module id 2. End index of module id 3. The range between start and end must be less than 100.
Steps		1. Query with start point and end point of module id
Expected Outcome		Output the success message with a string including Moduleid, Module name, Module schema, and IsActive or not. Each module is separated by ","
Acceptance Criteria	Given	Put the given start and end point
	When	I send the query to the database
	Then	I should get the message about modules' details.
Test result		PASS
Notes		[1] If an admin wants to get more than a hundred user information, he/she should query more than one time. [2] Sample: https://eratosuombackend.azurewebsites.net/api/getAllModules?start=1&end=10&code=JtAQchbEMtFZ6wX2Cef1FAnlxy6vfPo9o06eNqMaEKKjGoUZIDJ8Cw==
Tester		En-Wen TSAI
Date		04/06/2021

Test case for Story 16:

This case is only for the admin front-end, admin front-end is able to remove the specific module based on their needs.

Test ID		16
Test Case Title		Admin front-end is able to remove the specific module
Test Type		Functional
Objective		Verify that the admin is able to remove the existing module.
precondition		1. Admin had already created this module
Steps		1. Get the Module ID
Expected Outcome		Get the success message
Acceptance Criteria	Given	Put module ID as parameter
	When	I send this parameter to the server
	Then	I should get the success message from the server
Test result		PASS
Notes		[1] Sample: https://eratosuombackend.azurewebsites.net/api/removeModule?moduleID=2&code =pC04Se3EKKG7Is6PPO3az2Idtzg6gJEFrTVbvpOIKh0JX fd9itviGQ==
Tester		En-Wen TSAI
Date		03/06/2021