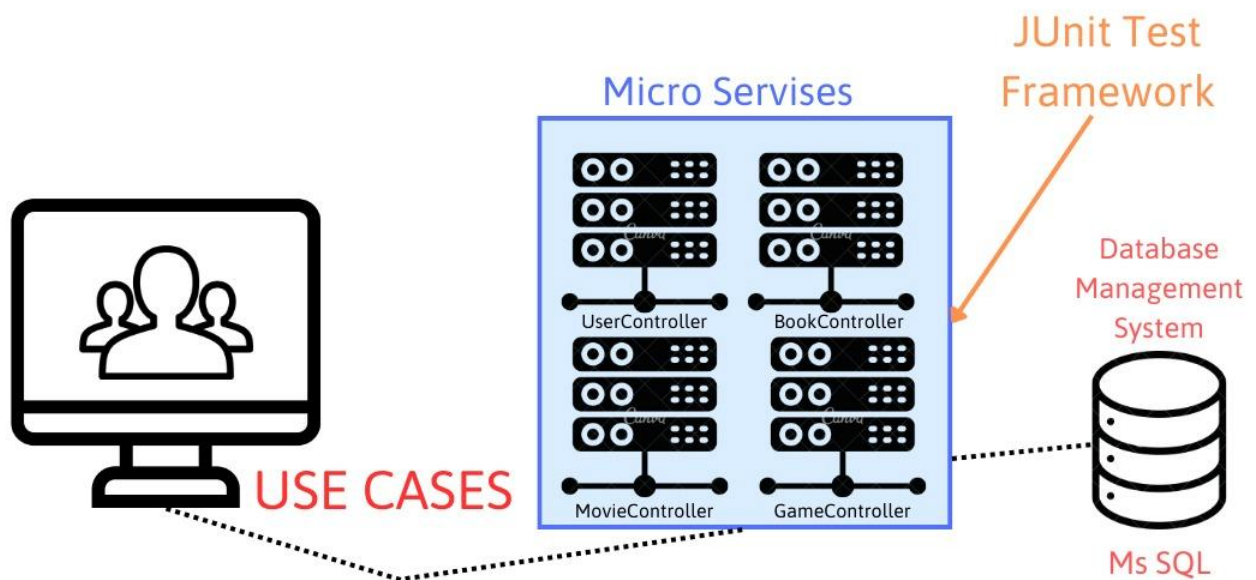# Systemtestdokument

Take A Side

INF202 MS#4

Verantwortliche/r:

▪ Merve Damla İnce, e200503039@stud.tau.edu.tr

▪ Damla Ersoy, e200503005@stud.tau.edu.tr

Stakeholder:

▪ Dl. Ömer Karacan, omer.karacan@tau.edu.tr

1) Systemüberblick

Micro Servises

JUnit Test Framework

UserController

BookController

MovieController

GameController

USE CASES

Database Management System

Ms SQL

2) Systemtestfälle

System Test Case ID Name	Sys_TC_1 Test provide report of USE CASE 1 via <Log in and vote events> service
Preconditions	Log in and vote web service is given: • Request: GET Log in (String email, String password) | PUT Vote(event_type Event) • Response: Verification of the user account with the relevant parameters | Updating the voting rates of events • System is up and running
Test Steps	1. GET Log in(String email, String password) is called with User email and password and PUT Vote() is called to update vote_rate. 2. Response data is parsed, and the expected data is selected.
Post-Condition	none
Test Data	none
Expected Result	Event.vote_rate = 63
Actual Result	Event.vote_rate = 63
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	REQ1.1/log in , REQ1.2/vote events

System Test Case ID Name	Sys_TC_2 Test provide report of USE CASE 2 via <Search(with parameters) and view events> service

Preconditions	Search and view web service is given:
	• Request: GET getbyAttribute(attr_type attr_name)
	• Response: A list of events with specifed attributes
	   - int eventID
	   - String event_name
	   - String event_year
	   - String event_genre
	   - float event_vote_rate
	
	   - where event;
	     1 <= eventID <= 32767
	
	• System is up and running

Test Steps	1. GET getbyAttribute(attr_type attr_name) is called with attributes of specifed event. 2. Response data is parsed, and the expected data is selected
Post-Condition	none
Test Data	none
Expected Result	Movie.getByYear(1982) = "Blade Runner"
Actual Result	Movie.getByYear(1982) = "Blade Runner"
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	REQ2/search and view events

System Test Case ID Name	Sys_TC_3 Test provide report of USE CASE 3 via <Organize events and users> service
Preconditions	Organize web service is given: • Request: POST addObject(obj_type obj_name) | DELETE deleteByEmail(String email) • Response: Adding new events with given parameteres and deleting users by specifed emails – Boolean ifExists() • System is up and running
Test Steps	1. POST addObject(obj_type obj_name) is called with relevant object name , DELETE deleteByEmail(String email) is called with User email. 2. Response data is parsed, and the expected data is selected
Post-Condition	none
Test Data	none
Expected Result	Book.ifExists("Nasıl Ölünür") = True;
Actual Result	Book.ifExists("Nasıl Ölünür") = True;
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	REQ3.1/organize events, REQ3.2/organize users

System Test Case ID Name	CI_TC_Books Test provide Report of database interface
Preconditions	Database Table "Books" incluedes a row for - book_id = <generated> - b_name = "Nasıl Ölünür" - release_year ="2019" - autor = "Emilie Zola" - page_number = "47" - publisher = "Can Yayınevi" - vote_rate = <generated> CS Web service call is mocked!
Test Steps	1. Ensure that CS Web service call is mocked! 2. Junit Test Case calls provide ifExistBookObjectReportId((String b_name) with book b_name 3.Response data is parsed, and the expected data is selected.
Post-Condition	none
Test Data	"Nasıl Ölünür","2019","Emilie Zola","47","Can Yayınevi"
Expected Result	Book.book_id = "1"
Actual Result	Book.book_id = "1"
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	REQ3.1/organize events

System Test Case ID Name	CI_TC_Games Test provide Report of database interface

Preconditions	Database Table "Games" incluedes a row for
	    ○ game_id = <generated>
	    ○ g_name = "Elden Ring"
	    ○ release_year ="2022"
	    ○ developer = "FromSoftware"
	    ○ duration = "100"
	    ○ vote_rate = <generated>
	CS Web service call is mocked!

Test Steps	1. Ensure that CS Web service call is mocked! 2. Junit Test Case calls provide ifExistGameObjectReportId((string g_name) with game g_name 3. Response data is parsed, and the expected data is selected.
Post-Condition	none
Test Data	"Elden Ring","2022","FromSoftware","100"
Expected Result	Game.game_id = "1"
Actual Result	Game.game_id = "1"
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	REQ3.1/organize events

System Test Case ID Name	CI_TC_WebMaster Test provide Report of database interface

Preconditions	Database Table "WebMaster" incluedes a row for
	  ○ id = "21212121212121212"
	  ○ first_name = "Nihat"
	  ○ last_name ="Doğan"
	  ○ gender= "0"
	  ○ tNo= "0541 414 4141"
	CS Web service call is mocked!

Test Steps	11. Ensure that CS Web service call is mocked! 2. Junit Test Case calls provide ifExistWebmasterObjectReportId((string tNo) with webmaster tNo 3. Response data is parsed, and the expected data is selected.
Post-Condition	none
Test Data	"Nihat","Doğan","0","0541 414 4141"
Expected Result	Webmaster.id = "21212121212121212"
Actual Result	Webmaster.id = "21212121212121212"
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	REQ3.2/organize users

System Test Case ID Name	CI_TC_Users Test provide Report of database interface

Preconditions	Database Table "Users" incluedes a row for
	    o email = user123@gmail.com
	    o first_name = "Fatih"
	    o last_name = "Terim"
	    o gender ="0"
	    o age= "69"
	    o u_password= "***********"
	CS Web service call is mocked!

Test Steps	1. Ensure that CS Web service call is mocked! 2. Junit Test Case calls provide ifExistUserObjectReportUser((string email) with user email, 3. Response data is parsed, and the expected data is selected.
Post-Condition	none
Test Data	"user123@gmail.com","Fatih","Terim","0","69","***********"
Expected Result	User.first_name = "Fatih"
Actual Result	User.first_name = "Fatih"
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	REQ3.2/organize users

System Test Case ID Name	CI_TC_Movies Test provide Report of database interface

Preconditions	Database Table "Movies" incluedes a row for
	  ○ movie_id = \<generated>
	  ○ m_name = "Blade Runner"
	  ○ release_year ="1982"
	  ○ director= "Ridley Scott"
	  ○ producer= "Micheal Dealey"
	  ○ vote_rate = \<generated>
	CS Web service call is mocked!

Test Steps	1. Ensure that CS Web service call is mocked! 2. Junit Test Case calls provide ifExistMovieObjectReportId((string m_name) with movie m_name 3. Response data is parsed, and the expected data is selected.
Post-Condition	none
Test Data	"Blade Runner","1982","Ridley Scott","Micheal Dealey"
Expected Result	Movie.movie_id = "1"
Actual Result	Movie.movie_id = "1"
Verdict (Pass/Fail)	Pass
Verified UC & Req. IDs	REQ3.1/organize events

3) Rückverfolgbarkeit

DBMS

CI Test Case

USE CASE 1 <Log in and vote events>

System Test Case

GET Log in | PUT Vote

UserController

USE CASE 2 <Search(with parameters) and view events>

GET getbyAttribute

MovieController

BookController

POST addObject | DELETE deleteByEmail

GameController

USE CASE 3 <Organize events and users>