

Final project

Cinema City



Table of contents:

Team members.....	3
About our website	3
Executive Summary.....	4
Main requirements	5
Non-functional requirements	5
Test requirements.....	6
Traceability table	7
Testing strategy	8
Testing Organization	9
Testing environment.....	10
Testing tools	10
Metrics.....	10
Testing techniques.....	11
Severity and urgency tables	18
Bug status table	19
Table of Role Holders.....	20
Error handling diagrams.....	21
Real Bugs	22
Conclusions and recommendations	27

Team members:

NAME	ID
Adi Malka	325258499
Adiel Sinvani	211935424
Tom Oz Asulin	316387695
Matan Ofri	206919680
Nofar Duchan	322599424

About our website

The site we chose is: Cinema City

Cinema City is the name of an Israeli chain of movie theater complexes ("Megaplex") owned by the brothers Leon and Moshe Edri. The complexes it owns, one of the largest in Israel, are in Ramat Hasharon, north of the Gillot interchange, west of Rishon Lezion, in Jerusalem, in Kfar Saba, Netanya, Beer Sheva, Hadera, and Ashdod. And it is in the process of building additional cinema complexes.



It offers a quality movie viewing experience, using the most advanced technologies, both in the field of projection and in the field of sound.

Target Audience:

The Cinema City website is intended for anyone interested in watching high quality movies and enjoying an upgraded entertainment experience. The site is suitable for the whole family, from small children to adults.

When will the Cinema City website be used?

The Cinema City website can be used at any time, 24 hours a day, 7 days a week

When you want to order a ticket, view information about movies screened in the cinema or that will be screened soon, etc.

Executive Summary

Project requirements, actions taken and project client definition:

In the project, a testing program must be created on the site we chose – "Cinema City" site - this is a site that is a gateway to a whole world of entertainment offering a wide range of services and functions designed to upgrade the cinema experience of the general public.

The first steps we took in order to start the testing program are a consolidated review of the site and getting to know it in depth, building testing strategies and preparing a schedule so that we can manage our time better and make sure we have enough time to do each part of the testing program.

Moreover, we organized the tests by raising possible risks when carrying out the test plan, but we handled them by joint teamwork and proper time management.

We then wrote main requirements for testing on our site and did a full coverage of the tests and test cases arising from them. Each team member wrote 5 test cases which he ran in order to optimize times and so that everyone would be synchronized while acting on all the comprehensive tests done on the site. To carry out the tests, we used test areas and equivalence departments that helped us examine in more depth the end conditions of the inputs that can be inserted into the site.

Finally, we built a system to manage bug handling by creating code tables for bug management, a table of role holders as well as appropriate charts that visually communicate the handling of the bug in our system. On the "Cinema City" website we found 5 bugs so we didn't need artificial bugs.

The client - the "Cinema City" site, which can carry out a review through this project on the tests done on the site and can also use the system we built to manage the handling of errors found in the system.

Main results:

- We were able to perform comprehensive and quality tests on the site in an efficient and good manner which cover all the functionality of the system and ensure that it meets the main requirements.
- We received more information about the "Cinema City" website and its function.
- We were able to work in cooperation while meeting the deadlines we set.
- With the writing of the test plan we improved our testing skill.

Main conclusions:

- Cooperation between all team members facilitates the project execution process.
- Allocating time to carry out tasks is beneficial for receiving suppliers quickly and optimizing time.
- Thorough testing before the development process, prevents problems before they reach customers, which leads to a higher quality product.



Main requirements

Module- viewing a list of movies

1. As a user I would like to see the list of movies that appear in a specific venue and date according to the movie type filter.



Module- order movie tickets

2. As a user I would like to choose specific seats in the hall in order to ensure a good place to watch.
3. As a user, I would like to enter my personal details when ordering in order to verify my details and receive the tickets for the correct details.

Non-functional requirements

- As a user, I would like the colors when choosing the seats to be clear in order for the system to be clear and convenient.
- As a user, I would like to have one way to order tickets and not several ways so that I don't get confused and ordering tickets is more understandable.
- As a user, I would like the credit card payment security on the site to be high, so that I can purchase tickets without fear.
- As a user, I would like the site to respond quickly to the actions I want to perform so that I don't have to wait a long time for a simple action.
- As a product manager, I would like to know that the website can withstand the load of a million users at a time, in order to avoid website crashes during use.

Test requirements:

As a user, I would like to see the list of movies that appear in a specific complex and on a specific date by filtering the type of movie in order to view information about a certain movie

1. I will choose a cinema complex and a certain date without filtering films. [PA-23](#)
2. I will select a cinema complex and a certain date and filter by regular films. [PA-24](#)
3. I will select a cinema complex and a certain date and filter by VIP movies. [PA-25](#)
4. I will select a cinema complex and a certain date and filter by ONYX. [PA-22](#)

As a user, I would like to select specific seats when ordering movie tickets in order to ensure a good place to watch

5. Selection consecutive seats in the same row in the middle. [PA-13](#)
6. Choosing non-consecutive seats from different rows. [PA-14](#)
7. Choosing consecutive seats from different rows. [PA-15](#)
8. Choosing consecutive seats but leaving one of the side seats alone. [PA-16](#)
9. Choosing a pair of seats that are a sofa. [PA-17](#)
10. Selection of non-consecutive seats in the same row between which there is one single chair. [PA-18](#)
11. Choosing only one ticket in the sofa. [PA-19](#)
12. Selection of non-consecutive seats in the same row between which there is more than one seat. [PA-20](#)
13. Choosing a chair designed for the disabled. [PA-21](#)

As a user, I would like to enter my personal details in order to finish ordering the movie tickets and receiving the tickets for the correct details

14. Entering numbers in the first name field when all other fields before are correct. [PA-1](#)
15. Entering numbers in the last name field when all other fields before are correct. [PA-2](#)
16. Entering an ID number with more than 9 digits and all the fields before it are correct. [PA-3](#)
17. Entering an id number with less than 9 digits and all the fields before it are correct. [PA-4](#)
18. Entering valid fields and clicking on notifications confirmation. [PA-5](#)
19. Entering valid fields and proceeding to payment with acceptance of the terms and conditions. [PA-6](#)
20. Entering valid fields and switching to payment without approval of the regulations. [PA-7](#)
21. Entering a different number in the phone number verification field and everything else is normal. [PA-8](#)
22. Entering numbers in the field for the phone number, and ensuring all fields are filled correctly beforehand. [PA-9](#)
23. Entering a phone number with less than 10 digits and every previous field is valid. [PA-10](#)
24. Entering letters in the ID number field and all the fields before it are correct. [PA-11](#)
25. Entering more than 10 digits in the phone number field and all the fields before it are correct. [PA-12](#)

Traceability table:

Test requirement number	Test requirement code	Test case pages	Running - Google chrome pages	Running - Firefox pages
1	PA-23	32	38	83
2	PA-24	33	39	84
3	PA-25	34	40	85
4	PA-22	31	37	82
5	PA-13	21	42	87
6	PA-14	22	52	91
7	PA-15	23-24	53	92
8	PA-16	25	44	76
9	PA-17	26	54	93
10	PA-18	27	45	77
11	PA-19	28	46	78
12	PA-20	29	47	79
13	PA-21	30	48	80
14	PA-1	2	50	89
15	PA-2	3	51	90
16	PA-3	4	67	106
17	PA-4	5	68	107
18	PA-5	6-7	56-57	95-96
19	PA-6	8-9	58-59	97-98
20	PA-7	10-11	60-61	99-100
21	PA-8	12-13	62-63	101-102
22	PA-9	14-15	64-65	103-104
23	PA-10	16-17	69-70	108-109
24	PA-11	18	71	110
25	PA-12	19-20	72-73	111-112

Testing strategy

Our testing strategy will be carried out manually and will be based on an effective combination of diverse testing techniques. Our goal is to cover all the essential requirements while ensuring a high level of testing while reducing the required work time and resources.



The content of the tests:

- Carrying out comprehensive tests of all the main test requirements found.
- Major system requirements that are not functional, such as compliance with loads and information security conditions, will not be tested within the framework of this testing

The order of the tests:

After classifying the test requirements according to priority levels, the highest priority tests will be performed first, and then the rest of the tests will be performed according to the established order of priority.

Advantages of this strategy:

- Comprehensive coverage - the integration of diverse testing techniques ensures thorough coverage of all requirements.
- Efficiency - focusing on high priority test requirements first allows efficient utilization of time and resources.
- High quality - adherence to a high level of testing throughout the entire process guarantees a quality product.

We believe that this approach will allow us to perform thorough tests while using resources effectively, and will ensure the success of the project.

Schedule:

- **07/03/24** - review of the site and a deep understanding of its function + thinking about the testing strategy and schedule.
- **14/03/24** - division into teams + start of work on part I - organization of tests, risk management, test environment and test tools.
- **21/03/24** - writing main requirements and test areas.
- **28/03/24** - writing test requirements and test cases and we will run the tests - at least 5 each.
- **04/04/24** - Division into teams - creating a system for managing bugs that have been found (if they are not found, we will create artificial bugs) and preparing bug report forms.
- **11/04/24** - preparation of a document containing all the parts + use of indices + executive summary.
- **17/04/24** - review of the project file and its submission.



Testing Organization:

Management risk:

- Performing multiple tests and writing them by me may lead to many mistakes such as lack of necessary conditions for a particular test and errors in the test details in order to deal with this risk, we will divide the existing team into pairs and make a replacement: each partner will examine his partner's test cases and thus we will be able to identify mistakes that appeared due to the load.
- Mismatches between team members can lead to errors, double checks, and generally double work to deal with this problem. We will develop a common means of communication for all team members at the beginning of each day, we will assign tasks and responsibilities to each member of the team, we will work on shared documents, and at the end of each day we will send an update on what tasks each team member has accomplished, and through this we will coordinate the tasks for the next day.
- Due to the security situation in the country, unexpected delays may occur due to lack of electricity and communications, as well as the absence of one or more staff members due to reserve service. To solve this risk, we will take a large "security" time interval, to ensure compliance with times and targets.
- Due to a lack of experience and knowledge of one or more team members, there may be tasks that the team member does not know how to perform In order to solve this problem, we will open a common forum for all team members where each team member can ask questions related to the task and be assisted by the rest of the team members, and we will hold a weekly meeting to discuss and see how to optimize the team members by dividing them into tasks that are right for them.

Priority:

All tests are divided according to models, each team assigned to a particular model will classify the tests according to upper to lower importance according to the testing requirements, requirements of paramount importance will be requirements that may harm the conduct of the business, for example ordering a seat in a film that has already been caught and requirements that may harm the profitability of the business, for example ordering tickets without inserting a means of payment.

Human Resources:

The personnel at our disposal is five students in the third year of software engineering at Sammi-Shamoon College of engineering, whose names are indicated at the beginning of the document.

Testing environment

The project's testing environment is a personal computer with an Internet connection and browsers installed on it.

We will use the Windows 10/11 operating system and two different browsers:

1. Google Chrome Version 123.0.6312.88
2. Firefox Version 125.0.1

Testing tools

The testing tools used for this project are the website "Qase", on both google chrome and Firefox.

Metrics

Incorporating metrics in our project is critical in order for our tests to be effective and accurate.

- **Test Coverage** – our goal is to have Full Completion of all of the tests that we have defined in the section requirements for testing to ensure every functionality is tested and passes.
- **Defect priority** - After the first set of tests all the bugs that are opened should be handled, first we handle the highest priority level bug and then go down the priority levels till the lowest level, thus focusing on the more critical functionalities of the website.
- **Execution time** – an important aspect of the tests is the tests execution time. We want to be able to track the execution time of the tests in order to optimize the testing workflows and make sure that all the tests run smoothly.



Testing techniques:

Test area 1 - Amount of characters (numbers) in the ID field in order to complete the movie ticket order.

- The variable entering the area: X - amount of characters (numbers) in the ID field that a user entered { $X > 0$ }
- The requested test: $0 < X \leq 9$
- Class C1 (valid): $0 < X \leq 9$
It means that the user entered a valid input into the ID field and can continue completing the order (if the input is less than 9, added zeros automatically).
- Class C2 (invalid): $X > 9$ (Please note that it is not possible to enter a negative ID number)
It means that the user entered an incorrect input into the ID field and cannot continue completing the order.
- The classes are parallel because they represent the same test for a different range of values of variable X and are independent of each other.

(valid) C1	Test case	parameter	test	expected result	meaning
	1 (exceeds lower limit)	X_i (amount of characters in ID field) = 0	$0 < X_i \leq 9$	The user cannot continue to complete the order	The system will wait for proper input in order to complete the order
	2 (lower limit)	X_i (amount of characters in ID field) = 4	$0 < X_i \leq 9$	The user can continue to complete the order	The system will add zeros to the beginning of the ID card, and the user can continue to complete the order
	3 (upper limit)	X_i (amount of characters in ID field) = 9	$0 < X_i \leq 9$	The user can continue to complete the order	The user entered valid input, and the user can continue to complete the order

(invalid) C2	Test case	parameter	test	expected result	meaning
	1 (exceeds lower limit)	X_i (amount of characters in ID field) = 9	$X_i > 9$	The user cannot continue to complete the order	The user entered invalid input, and the user cannot continue to complete the order
	2 (lower limit)	X_i (amount of characters in ID field) = 10	$X_i > 9$	The user cannot continue to complete the order	The user entered invalid input, and the user cannot continue to complete the order

Test area 2 - A string of characters in the ID field in order to complete the movie ticket order.

- The variable entering the area: Y - a string in the ID field entered by a user { $0 < \text{len}(Y) \leq 9$ }
- The requested test: Y- a string containing only numbers
- Class C3 (valid): Y- a string containing only numbers
It means that the user entered a valid input into the ID field and can continue completing the order.
- Class C4 (invalid): Y- a string containing characters that are not necessarily numbers
It means that the user entered an incorrect input into the ID field and cannot continue completing the order.
- The classes are parallel because they represent the same test for a different range of values of the Y variable and are independent of each other.

(valid) C3	Test case	parameter	test	expected result	meaning
	1 (required value)	Y (string in the ID field) = '033245666' string containing only numbers	Y = string containing only numbers	The user can continue to complete the order	The user entered valid input, and the user can continue to complete the order

(invalid) C4	Test case	parameter	test	expected result	meaning
	2 (wrong value)	Y (string in the ID field) = '#bjt' -string containing characters that are not numbers	Y = string containing only numbers	The user cannot continue to complete the order	The user entered invalid input, and the user cannot continue to complete the order

Test area 3 - A string of characters in the first name field in order to complete the entry of personal details and order the movie ticket.

- The variable entering the area: Z – a string of characters in the first name field entered by a user { $0 < \text{len}(Z)$ }
- The requested test: Z- a string containing only characters that are letters
- Class C5 (valid): Z- a string containing only characters that are letters
It means that the user entered a valid input in the first name field and can continue completing the order.
- Class C6 (invalid): Z- a string containing characters that are not necessarily just letters
It means that the user entered an incorrect input in the first name field and cannot continue completing the order.
- The classes are parallel because they represent the same test for a different range of values of the Z variable and are independent of each other.

(valid) C5	Test case	parameter	test	expected result	meaning
	1 (required value)	Z (string in the name field) = 'Nofar'- string containing only letters	Z = string containing only letters	The user can continue to complete the order	The user entered valid input, and the user can continue to complete the order

(invalid) C6	Test case	parameter	test	expected result	meaning
	2 (wrong value)	Z (string in the name field) = '1#67'- string containing characters that are not letters	Z = string containing only letters	The user cannot continue to complete the order	The user entered invalid input, and the user cannot continue to complete the order

Test area 4 - A string of characters in the last name field in order to complete the entry of personal details and order the movie ticket.

- The variable entering the area: N – a string of characters in the last name field entered by a user { $\text{len}(N) > 0$ }
- The requested test: N- a string containing only characters that are letters
- Class C7 (valid): N- a string containing only characters that are letters
It means that the user entered a valid input in the last name field and can continue completing the order.
- Class C8 (invalid): N- a string containing characters that are not necessarily just letters
It means that the user entered an invalid input in the last name field and cannot continue completing the order.
- The classes are parallel because they represent the same test for a different range of values of variable N and are independent of each other.

(valid) C7	Test case	parameter	test	expected result	meaning
	1 (required value)	N (string in the last name field) = 'Duchan'- string containing only letters	N = string containing only letters	The user can continue to complete the order	The user entered valid input, and the user can continue to complete the order

(invalid) C8	Test case	parameter	test	expected result	meaning
	2 (wrong value)	N (string in the last name field) = '6%72' -string containing characters that are not letters	N = string containing only letters	The user cannot continue to complete the order	The user entered invalid input, and the user cannot continue to complete the order

Test area 5 - Amount of characters (numbers) in the phone number field in order to complete the movie ticket order.

- The variable entering the area: W - amount of characters (numbers) in the phone number field that a user entered { $W \geq 0$ }
- The required test: $W = 10$
- Class C9 (valid): $W = 10$
It means that the user entered a valid input in the phone number field and can continue completing the order
- Class C10 (invalid): $W \neq 10$
It means that the user entered an incorrect input into the ID field and cannot continue completing the order.
- The classes are parallel because they represent the same test for a different range of values of the W variable and are independent of each other.

(valid) C9	Test case	parameter	test	expected result	meaning
	1 (required value)	W (amount of characters in phone number field) = 10	$W = 10$	The user can continue to complete the order	The user entered valid input, and the user can continue to complete the order

(invalid) C10	Test case	parameter	test	expected result	meaning
	2 (wrong value)	W (amount of characters in phone number field) = 10	$W \neq 10$	The user cannot continue to complete the order	The user entered invalid input, and the user cannot continue to complete the order

Test area 6 - A string of characters in the field phone number in order to complete the entry of the personal details and order the movie ticket.

- The variable entering the area: T – string of characters in the phone number field that a user entered { len(T) = 10 }
- The requested test: T- a string containing only characters that are numbers
- Class C11 (valid): T- a string containing only characters that are numbers
It means that the user entered a valid input in the phone number field and can continue completing the order.
- Class C12 (invalid): T- a string containing characters that are not necessarily just numbers
It means that the user entered an incorrect input in the phone number field and cannot continue completing the order.
- The classes are parallel because they represent the same test for a different range of values of the variable T and are independent of each other.

(valid) C11	Test case	parameter	test	expected result	meaning
	1 (required value)	T (string in the phone number field) = '0524567892' - string containing only numbers	T = string containing only numbers	The user can continue to complete the order	The user entered valid input, and the user can continue to complete the order

(invalid) C12	Test case	parameter	test	expected result	meaning
	2 (wrong value)	T (string in the phone number field) = 'ydtbkłpi%#' - string containing characters that are not numbers	T = string containing only numbers	The user cannot continue to complete the order	The user entered invalid input, and the user cannot continue to complete the order

Test area 7 - A string of characters in the phone number verification field in order to complete the entry of the personal details and order the movie ticket.

- The variable entering the area: S – string of characters in the phone number verification field that a user entered { len(S) = 10 }
- The requested test: S- a string that matches the string entered in the phone number field
- Class C13 (valid): S- a string that matches the string entered in the phone number field
It means that the user entered a valid input in the phone number verification field and can continue to complete the order.
- Class C14 (invalid): S- a string that does not match the string entered in the phone number field
It means that the user entered an incorrect input in the phone number verification field and cannot continue completing the order.
- The classes are parallel because they represent the same test for a different range of values of variable S and are independent of each other.

(valid) C13	Test case	parameter	test	expected result	meaning
	1 (required value)	S (string in the phone number verification field) = '0524567892'- string containing only numbers	S = string matches the string entered in the phone number field	The user can continue to complete the order	The user entered valid input, and the user can continue to complete the order

(invalid) C14	Test case	parameter	test	expected result	meaning
	2 (wrong value)	S (string in the phone number verification field) = '0524567894' – string does not match the string entered in the phone number field	S = string matches the string entered in the phone number field	The user cannot continue to complete the order	The user entered invalid input, and the user cannot continue to complete the order

Severity and urgency tables:

Severity table:

severity level	code
Urgent	1
necessary	2
negligible	3

Urgency table:

level of urgency	code
Immediate	A
No urgent	B
Marginal	C

Explanations of the different ranks:

Severity levels:

- 1) Urgency** - most significant failures that cause the site to crash, loss of important data or failures that cause important functionality in the system to not work.
- 2) Necessary**- errors that deal with improper input checks, weak runtimes, and negligible bugs that can be bypassed.
- 3) Negligible** - the website's visibility, the user's discomfort, functionality that requires a long process compared to simple and short functionality.

level of urgency:

- 1) Immediate** - as soon as possible.
- 2) Not urgent** - if there is nothing more urgent that belongs to the previous level (immediate).
- 3) Marginal** - if there is time left and there are no more important things.

Bug status table:

status	code
Report an error	Report
Error verification	Verification
Canceling a test	Cancel
Correction confirmation	Confirmation
Error classification	Classification
Error fixing	Fixed
Check again	Check
Close an inspection	Close

Explanations of the different statuses:

Report an error-(checker) The checker found an error in the system and reported it and forwarded it for verification by the head of the checker team.

Error verification - (head of the testing team) the head of the testing team received the test and verifies the test.

If the test was not relevant, the head of the testing team cancels it and thus it is close.

If the test is relevant, the head of the testing team forwards it to the manager for approval.

Canceling a test - (head of the team of testers/manager) the test was not relevant and therefore canceled by the head of the team of testers or the manager decided to cancel for reasons of lack of budgets, etc...

Correction confirmation - (Manager) The manager received the inspection after it was verified or after it was re-inspected by the head of the inspection team and decides whether to invest in the repair or cancel it.

Error classification - (developer team leader) The developer team leader received the test and forwards it to the developer responsible for this part of the system.

Error fixing - (developers) The developer received the test from the head of the team and fixes it.

Check again - (head of the testing team) after the error has been corrected, it is forwarded to the head of the testing team to verify that it has indeed been corrected.

Error closing - (head of the testing team) the head of the testing team closes the test and marks it as a fixed error.

Table of Role Holders:

	test run	opening error	Error fixing	Error closing	ביטול בדיקה	שינוי סטטוס
QA Engineer	X	X				X
QA Team Lead	X			X	X	X
A developer			X			X
Development Team Lead						X
Manager					X	X

QA Engineer - his job is to carry out the system tests given to him after finding an error, opens an error form and passes it on.

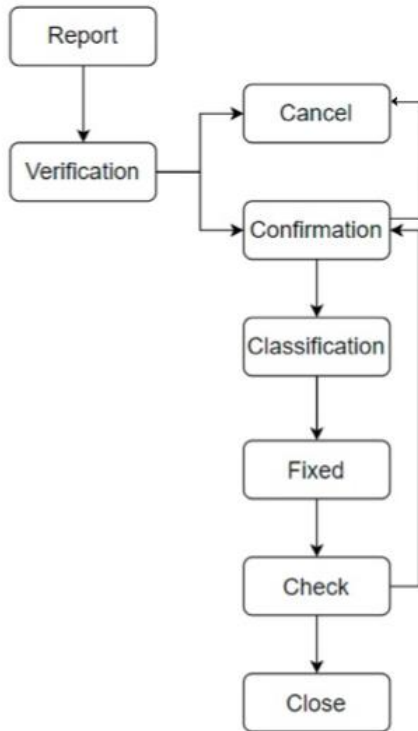
QA Team Lead - his job is to verify the test performed by the tester, he must cancel it or pass it on to the manager, and in addition he must check once more after the repair of the key in order to close it or return it again for confirmation of continued care.

A developer - his job is to create the functionality in the system and fix them according to the error forms.

Development Team Lead - his job is to classify the test to the key relevant to correcting the error.

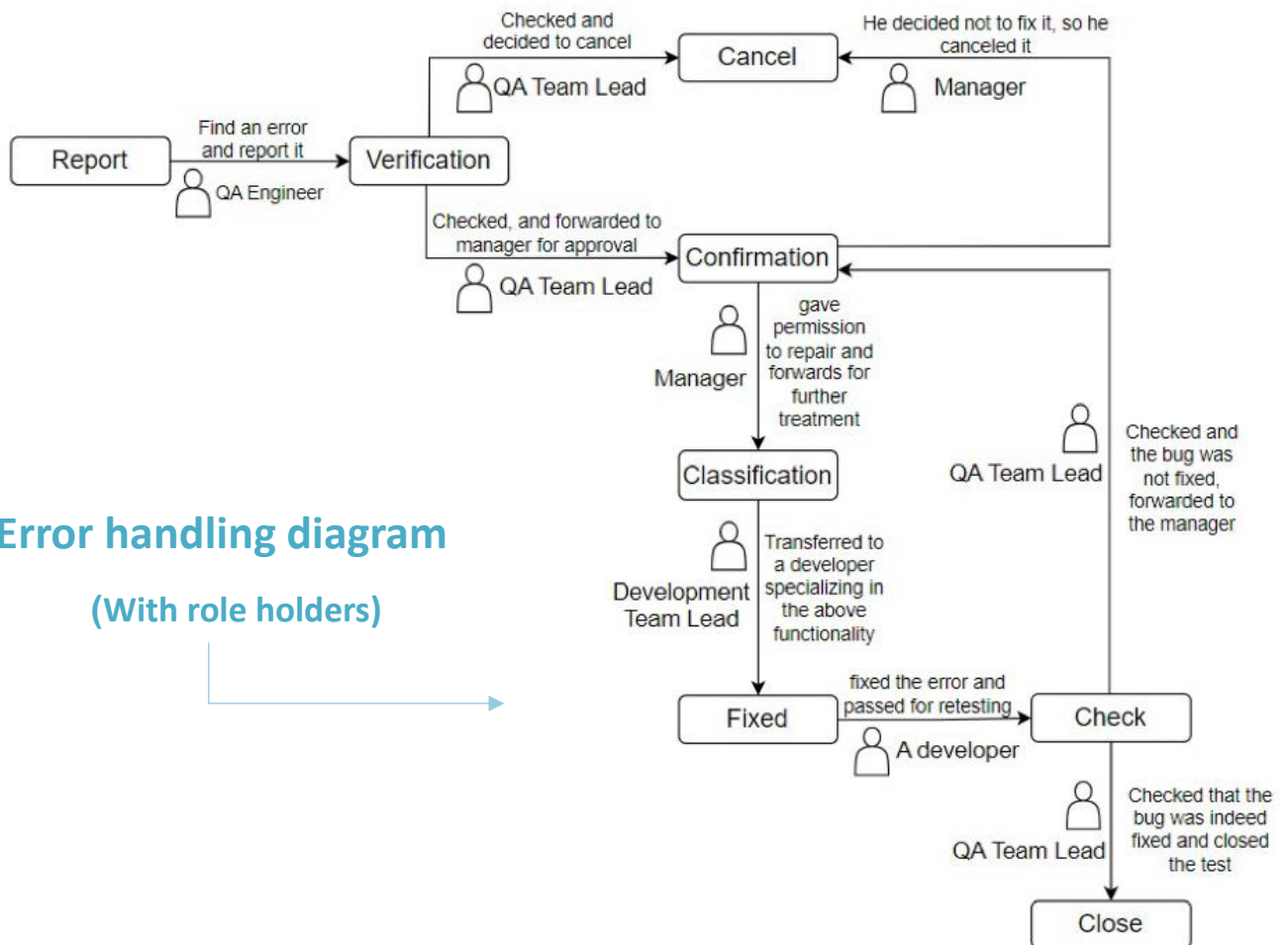
Manager - his role is to unite all the factors operating in the system, and to decide to cancel tests in such and other cases (such as: lack of budgets, etc.), to approve the continuation of the error handling.

Error handling diagrams



Error handling diagram

(Without role holders)



Error handling diagram

(With role holders)

Real Bugs:

Real bug 1:

Choosing consecutive seats but leaving one of the side seats alone

Error handling form:

Error ID code: 01 **Error title:** Discontinuous seat selection with a single seat left.

The date of opening the error: 17/04/24

Identity of the error opener: Adi (QA Engineer)

Identity of the approver: Adiel (QA Team Lead)

Degree of severity: 2 **Priority for treatment:** C

current status: Classification. **from date:** 24/04/24 **Current warranty:** Matan (Development Team Lead)

Planned end date for status: 29/04/24.

Error description: When the user wants to order movie tickets when selecting the seats, he selects several non-consecutive seats and leaves one single seat in between, so no he cannot continue to the payment process.

Detail of the test environment: On the Cinema City website (both in the Chrome and Firefox browsers) after coordinating the desired movie, venue, and date, the user has to choose seats.

Description of current treatment: The classification of the test and its belonging to the developer responsible for the above functionality.

Final status for error: Fixed **Date:** 29/04/24 **Confirm:** Matan (Development Team Lead)

Free text accompanying the error: In terms of convenience for the user who wants to order separate seats that did not sit next to him, therefore leaving a single chair.

Real bug 2:

Selection of non-consecutive seats in the same row between which there is one single chair.

Error handling form:

Error ID code: 02 **Error title:** Choosing consecutive seats but leaving one of the side seats alone

The date of opening the error: 17/04/24

Identity of the error opener: Adi (QA Engineer)

Identity of the approver: Adiel (QA Team Lead)

Degree of severity: 2 **Priority for treatment:** C

current status: Verification **from date:** 25/04/24 **Current warranty:** Adiel (QA Team Lead)

Planned end date for status: 01/05/24

Error description: When the user wants to order movie tickets when selecting the seats, he selects several consecutive seats and leaves one single seat aside, so no he cannot proceed to the payment process.

Detail of the test environment: On the Cinema City website (both in the chrome and firefox browsers) after coordinating a movie, a complex, and how they are desired, the user has to choose seats.

Description of current treatment: The QA Team Lead received the error from the QA Engineer and verifies it again.

Final status for error: Cancel **Date:** 01/05/24 **Confirm:** Adiel (QA Team Lead)

Free text accompanying the error: In terms of convenience for the user who wants to order seats without the side seat.

Real bug 3:

Entering an ID number with less than 9 digits and all the fields before it are correct

Error handling form:

Error ID code: 03 **Error title:** Entering an ID with less than 9 digits and all the fields before that are correct.

The date of opening the error: 17/04/24

Identity of the error opener: Adi (QA Engineer)

Identity of the approver: Adiel (QA Team Lead)

Degree of severity: 2 **Priority for treatment:** B

current status: Fixed **from date:** 25/04/24 **Current warranty:** Nofar (Developer)

Planned end date for status: 11/05/24

Error description: When the user wants to order movie tickets when entering his personal information, he enters a PIN with less than 9 digits and all the previous fields are correct and he does not receive an error message or complete zeroes.

Detail of the test environment: On the Cinema City website (both in the Chrome browser and in the Firefox browser), after coordinating a movie, place and date, choosing the desired seats, the user had to enter his personal information, enter all the fields up to the valid identity card. but entered less than 9 digits in the ID.

Description of current treatment: The developer received the error from the Development Team Lead and needs to be fixed.

Final status for error: Check **Date:** 11/05/24 **Confirm:** Nofar (Developer)

Free text accompanying the error: If the certificate is the correct size but less than 9 digits, it can be completed with zeros, but if it is not the correct size, an error message will be given instead.

Real bug 4:

Entering a different number in the phone number verification field and everything else is normal

Error handling form:

Error ID code: 04 **Error title:** You entered a different number in the phone number verification field, and all fields before that are correct.

The date of opening the error: 17/04/24

Identity of the error opener: Adi (QA Engineer)

Identity of the approver: Adiel (QA Team Lead)

Degree of severity: 2 **Priority for treatment:** B

current status: Classification **from date:** 25/04/24 **Current warranty:** Matan (Development Team Lead)

Planned end date for status: 30/05/24

Error description: When the user wants to order movie tickets while entering his personal information, he enters a phone number and in the matching field a different phone number is entered and he does not receive an error message about this instead.

Detail of the test environment: On the Cinema City website (both in the chrome browser and in the firefox browser) after coordinating a movie, complex, and date, choosing the desired seats, the user had to enter his personal information, entered all the fields up to the phone number and, in verification, enters a different phone number from the field.

Description of current treatment: QA Team Lead received an error and it needs to be classified to the developer responsible for its domain.

Final status for error: Fixed **Date:** 30/05/24 **Confirm:** Matan (Development Team Lead)

Free text accompanying the error: Add an error message instead of the phone not being the same.

Real bug 5:

Entering a phone number with less than 10 digits and every previous field is valid.

Error handling form:

Error ID code: 05 **Error title:** You entered a phone number with less than 10 digits and all the fields before that are correct.

The date of opening the error: 17/04/24

Identity of the error opener: Adi (QA Engineer)

Identity of the approver: Adiel (QA Team Lead)

Degree of severity: 2 **Priority for treatment:** B

current status: Classification **from date:** 25/04/24 **Current warranty:** Matan (Development Team Lead)

Planned end date for status: 30/05/24

Error description: When the user wants to order movie tickets when entering his personal information, he enters a phone number with less than 10 digits (invalid number) and instead receives an error message The phone is not valid.

Detail of the test environment: On the Cinema City website (both in the chrome browser and in the firefox browser) after coordinating a movie, venue, and date, choosing the desired seats, the user had to enter his personal information, entered all the fields up to the phone number and in this field he entered an incorrect .phone number

Description of current treatment: QA Team Lead received an error and it needs to be classified to the developer responsible for its domain.

Final status for error: Flixed **Date:** 30/05/24 **Confirm:** Matan (Development Team Lead)

Free text accompanying the error: Add an error message instead of the phone not working.

Conclusions and recommendations:

Website recommendations:

The site meets most of the customer's needs and the goals of the tests we performed. Overall, this site is user-friendly, but we would still recommend the site to be clearer, more accessible and to pay attention to the small details. For example: error messages when entering incorrect values, this will help the user "Average" to complete the ticket purchase process successfully and without unnecessary delays.

Recommendations and conclusions for the method of testing:

Using the various test techniques helped us to implement the process correctly, detailing the tests and dividing each test into the different stages and values allowed us to "get down" to the user's level and understand his need, our method of working as team members by dividing into pairs and dividing daily tasks helped us to optimize the process Writing the requirements for testing correctly by thinking together and going down to details. The group chat/question forum helped us answer each other and carry out the entire testing process in the most accurate way.

Learning from the process:

From this project we learned how to cooperate correctly between all the team members personally, everyone learned to help each other, we learned how division into teams and division of tasks can optimize the process and make it simpler and more successful, in addition each team member understood how to divide the sub-tasks in the way that suits him and is correct for him, the clear conclusion for all of us from this process is that proper teamwork will lead to the simplicity of the task and the maximum utilization of the abilities of the team members.

