table of contents

General 2

Testing Team Members 3

About The Website 3

Manager Summary 4

Main Requirements 5

Non-Functional Testing 6

Requirements for Testing..... 7

Testing Strategies 13

Tests Organization..... 14

Testing Environment 15

Testing Tools 16

Success Metrics for the Project 16

Test Levels 16

Testing Plan 16

Settings Domain Testing 17

Waiters Calculator Domain Testing 33

Percentage Calculator Domain Testing 38

Staff Calculator Domain Testing 45

Points Calculator Domain Testing 52

Contact Us Domain Testing ....., ... 58

Testing Techniques we used 62

Bugs/Defects management 63

Urgency and hardware table level 63

Bag Status Table 63

Employees and responsibilities..... 64

Bug handling diagram..... 65

Bug lifecycle diagram..... 68

Test Results 69

Bugs Found 69

Real Bugs 69

Made Up Bugs 69

Details of bugs found ..... Tests document

Bug Report forms..... 70

Conclusions and recommendations 75

1. General:

Testing Team Members:

1. Lior Shilon
2. Elyasaf Sinvani
3. Sharon Angado
4. Shmuel Malikov

The Website:

- waiterscalculator.com

This website is used to help "mutual tips" restaurants distribute the tips between the employees faster.

The website has a “settings” system in which the user can define a percentage of the total tips that goes to different restaurant’s roles and set an affair to the restaurant based on the hours each waiter worked.

between the roles: kitchen staff, bartenders, shift manager, busser, expo, runners and hostess.

The website has 4 different calculators, each calculator provides a different method for a “tip sharing system”, said system will be used for distributing the money between the restaurant’s workers.

1. Waiters Calculator
   Gets the total tips and the waiters hours. calculates the affairs to the stuff and the restaurant based on the settings the user defined and the amount of money each waiter receives by their hours of work.

2. Percentage Calculator
   Does exactly what the Waiters Calculator does but has another feature, the user can associate a percentage to a specific waiter which gives the user the option to reward senior waiters with higher percentage for the same amount of time.
   For example in some Sheff restaurants new waiters earn 70% of the “per hour earnings” and senior waiters earn 100%.

3. Staff Calculator
   The calculator gets all the employees and their hours of work and presents the revenue per role per person in said role based on their hours of work.
   Citation: https://waiterscalculator.com/how-to-use.html#all-staff-calculator
   “Calculates the salary each role earned based on the settings the user defined , calculates the per hour earnings of each role by dividing the total role revenue by the

total role hours, and calculates how much an employee gets by multyplying the per hour earnings by the hours of work."

4. Points Calculator
   each employee is assigned with a number of points which his daily tips earnings will be calculated by. (divides the total tips by the total employees' points to calculate how much money the employees earned per point).

Manager Summary:

In the project we were asked to run test on the website "waiterscalculator.com", a website who helps mutual tips restaurants divide the total tips was made in a shift between the employees that work in the shift.

each team member was asked to perform at least 5 functional tests and the whole team was required to report at least 5 bugs found in the testing process and write conclusions and recommendations arising from the tests results.

To achieve said goals, we created a testing plan that includes:

- Grouping test by their priority and setting each group a percentage of the test that should pass in order to achieve success (See under header "Success Metrics For The Project").
- Predefined schedule for each section of the work with an earlier deadline in order to handle unexpected events which will cause a delay in the schedule and a predefined testing strategy (See under header "Testing strategies")
- Covering the risks that arising from the various limitations and ways to handle them (See under header "Tests Organization").

As a team, we divided the website to different modules, wrote user stories under each module and requirements under each user story, we created the tests and divided them among the team members.
The modules we chose to test: Settings, Waiters Calculator, Percentage Calculator, Staff Calculator, Contact Us.
Each team performed his share of the tests
Due to the fact that we do not have access to the site code, we ran only black box tests.
We used Test Areas, Equivalence Partitions and Boundary Value Analysis.

The clients of the project is the course staff and the website's developers who we contacted through the Contact Us section of the website about bugs we found.

Final Results:

- Even though the website stood with our testing goals, we found a crucial bug which presents the user with negative amount of money assigned to a waiter, we think it arise from a calculation error which is the main functionality of the website and therefore should be attended as soon as possible.
- The project met the predefined deadline

- We agreed that the design of the website looks a bit old and probably should be updated

Conclusions:

- We recommend improving the design of the website.
- Using the testing techniques helps to identify all the test cases that needed to be tested and helps defining a clear goals regarding the tests which contributes to the confidence in the testing process.
- Working together, as a team, on most of the project's parts kept us synchronized, very aware of the project progress and details and each team member felt he contributed his share and opinion to the project.
- Having one team member run it's tests and then assigning another team member to check his tests proven to be very efficient in detecting mistakes in the tests.

Main Requirements:

The models for testing:

- Settings
- Waiters Calculator
- Percentage Calculator
- Staff Calculator
- Points Calculator
- Contact Us

Module – Settings:

1. As a user I want to be able to set an hourly affair to the restaurant for waiters and a percentage of the total tips that goes to: kitchen staff, bartenders, shift manager, busser, expo, runners and hostess, so that I will able to create a "money affair system" that matches the restaurant I work in.
2. As a user I want all my defined settings to be saved, so that each time I visit the website or use different calculator I won't need to re-enter all the settings.

Module – Waiters Calculator:

3. As a user I want to be able to calculate the amount of money that goes to each restaurant's role and to the restaurant by the settings I defined, by entering all waiter's hours of work and optionally their names and by entering the total tips made, so that I will be able to spread the tips easily and quickly between the employees.
4. As a user I want to be able to easily add waiters to the calculation so that I will be able to spread the tips between the actual number of waiters that worked in a specific shift
5. As a user I want to easily remove waiters from the calculation so that if I accidently added one extra waiter, I wouldn't need to reset what I did and start form beginning again.

Module – Percentage Calculator:

6. As a user I want to be able to calculate the amount of money that goes to each restaurant's role and to the restaurant by the settings I defined, by the total tips made, by entering all waiter's hours of work and optionally their names and by assigning a percentage to each waiter, so that I will be able to easily and quickly spread the tips between the employees and reward senior waiters over new waiters for the same amount of time they worked.
7. As a user I want to be able to easily add waiters to the calculation so that I will be able to spread the tips between the actual number of waiters that worked in a specific shift.
8. As a user I want to easily remove waiters from the calculation so that if I accidently added one extra waiter, I wouldn't need to reset what I did and start form beginning again.

Module – Staff Calculator:

9. As a user I want to be able to calculate the amount of money that goes to each restaurant's employee and to the restaurant by the settings I defined, by the total tips made, by entering all employees hours of work and assigning each employee a role and optionally a name, so that I will be able to easily and quickly spread the tips between all employees based on the "percentage from total tips" assigned to them and their hours of work.
10. As a user I want to be able to easily add restaurant employees to the calculation so that I will be able to spread the tips between the actual number of employees that worked in a specific shift.
11. As a user I want to easily remove restaurant employees from the calculation so that if I accidently added one extra employee, I wouldn't need to reset the page and start form beginning again.

Module – Points Calculator:

12. As a user I want to be able to calculate the amount of money that goes to each restaurant's employee by entering the total tips made and by assigning each employee a specific amount of points and potentially a name, so that I will be able to easily divide the total tips received among the employees based on their role or contribution to the business.
13. As a user I want to be able to easily add restaurant employees to the calculation so that I will be able to spread the tips between the actual number of employees that worked in a specific shift.
14. As a user I want to easily remove restaurant employees from the calculation so that if I accidently added one extra employee, I wouldn't need to reset the page and start form beginning again.

Module – Contact Us:

15. As a user I want to be able contact the company built the website by providing them with my name, the message and optionally my email so that I will be able to report an error I found or suggest about ideas for modification that matches the way my restaurant handles tips sharing and get a response from the company.

Non-Functional Requirements:

- As a developer I want the website to have clear and thorough documentation so that I will be able to understand the code more easily when needed.

- As a product manager I want the website to be compatible with following web browsers: Safari, Google Chrome, Microsoft Edge, Mozilla Firefox, and following devices: desktop computers, laptops, and mobile devices. (tests document, test case ID: NF2)
- As a product manager I want the website to load in less than 2 seconds so that users will have a good experience when using the website. (tests document, test case ID: NF1)
- As a product manager I want the website to withstand 500,000 at a time So the website won't crash when in use.
- As a user I want the website to be secure so my information will remain private.(NF3)
- As a product manager I want the website to have an explanation about how to use each calculator so users will be able to use it according to their needs (NF4)

Requirements For Testing:

1. As a user I want to be able to set an hourly affair to the restaurant for waiters and a percentage of the total tips that goes to kitchen staff, bartenders, shift manager, busser, expo, runners, and hostess, so that I will be able to create a "money affair system" that matches the restaurant I work in.

1.1) Entering invalid value to the "Waiter pays the restaurant per hour" text input field and valid values to the percentage of tips that goes to the: kitchen staff, bartenders, shift manager, busser, expo, runners and hostess text input fields.

1.2) Entering invalid value to the "percentage of tips that goes to kitchen" text input field and valid values to the percentage of tips that goes to the: bartenders, shift manager, busser, expo, runners and hostess text input fields and "Waiter pays the restaurant per hour" text input field.

1.3) Entering invalid value to the "percentage of tips that goes to bartenders" text input field and valid values to the percentage of tips that goes to the: kitchen, shift manager, busser, expo, runners and hostess text input fields and "Waiter pays the restaurant per hour" text input field.

1.4) Entering invalid value to the "percentage of tips that goes to shift manager" text input field and valid values to the percentage of tips that goes to the: kitchen, bartenders, busser, expo, runners and hostess text input fields and "Waiter pays the restaurant per hour" text input field.

1.5) Entering invalid value to the "percentage of tips that goes to busser" text input field and valid values to the percentage of tips that goes to the: kitchen ,bartenders, shift manager, expo, runners and hostess text input fields and "Waiter pays the restaurant per hour" text input field.

1.6) Entering invalid value to the "percentage of tips that goes to expo" text input field and valid values to the percentage of tips that goes to the: kitchen, bartenders, shift manager, busser, runners and hostess text input fields and "Waiter pays the restaurant per hour" text input field.

1.7) Entering invalid value to the "percentage of tips that goes to runners" text input field and valid values to the percentage of tips that goes to the: kitchen, bartenders, shift manager, busser, expo and hostess text input fields and "Waiter pays the restaurant per hour" text input field.

1.8) Entering invalid value to the "percentage of tips that goes to hostess" text input field and valid values to the percentage of tips that goes to the: kitchen, bartenders, shift manager, busser, expo and runners text input fields and "Waiter pays the restaurant per hour" text input field.

1.9) Entering valid values to all the text input fields but the sum of all the percentages defined is 100 or greater.

1.10) Entering valid values to all the input fields when the sum of all percentages is smaller than 100.

2. As a user I want all my defined settings to be saved, so that each time I visit the website or use different calculator I won't need to re-enter all the settings.
   - 2.1) Entering valid values to all the text input fields and pressing the save button than exit website and reopen it again from the same browser.
   - 2.2) Entering valid values to all the text input fields and pressing the save button than switch to another calculator.
   - 2.3) Entering invalid value to a text input field without pressing the save button than exit the website and reopen it again from the same browser.

3. As a user I want to be able to calculate the amount of money that goes to each restaurant's role and to the restaurant by the settings I defined, by entering all waiter's hours of work and optionally their names and by entering the total tips made, so that I will be able to spread the tips easily and quickly between the employees.
   - 3.1) Calculating tips spread by entering invalid total tips amount and valid waiters' hours and names.
   - 3.2) Calculating tips spread by entering valid total tips amount and waiters name, and invalid waiter's hour.
   - 3.3) Calculating tips spread with valid total tips amount and valid waiters hours when all the "name" text input fields are empty.
   - 3.4) Calculating tips spread with valid total tips amount and valid waiters hours when all the "name" text input fields are full.
   - 3.5) Calculating tips spread with valid total tips amount, waiters' hours and names when the total affairs to the restaurant staff and the restaurant, are bigger than the total tips that was made.
   - 3.6) Calculating tips spread with valid: total tips amount, waiters' hours and names after changing one of the settings to be an invalid number without saving the settings.
   - 3.7) Checking that after a calculation the "Total Hours" output field shows the sum of all the waiter's hours in the table.
   - 3.8) Checking that all the affairs to each role are calculated by the settings the user defined.
   - 3.9) Checking that waiter's earnings per hour calculation is as follows:
     waiters per hour = (total tips – total affairs) / total waiters' hours.
   - 3.10) Checking that the amount of money each waiter in the table receives is calculated by: "Per Hour" earnings * waiter's hours

3.11) Checking that: total waiters earnings + total affairs = total tips when the maximum error range allowed is 1 (currency)

3.12) Checking that the transparency of the “ADD ROW” and “DELETE ROW” button increase when pressing the calculate button GUI)

4. As a user I want to be able to easily add waiters to the calculation so that I will be able to spread the tips between the actual number of waiters that worked in a specific shift.
   - 4.1) Add a row to the table by clicking the add row button when not in the middle of a calculation.
   - 4.2) Add a row to the table by clicking the add row button when in the middle of a calculation.

5. As a user I want to easily remove waiters from the calculation so that if I accidently added one extra waiter, I wouldn’t need to reset what I did and start form beginning again.
   - 5.1) Remove a row from the waiters table when the number of table rows is bigger than 1 and not in the middle of calculation.
   - 5.2) Remove a row from the waiters table when the number of table rows is exactly 1.
   - 5.3) Remove a row from the waiters table when the number of table rows is bigger than 1 and in the middle of calculation.

6. As a user I want to be able to calculate the amount of money that goes to each restaurant’s role and to the restaurant by the settings I defined, by the total tips made, by entering all waiter’s hours of work and optionally their names and by assigning a percentage to each waiter, so that I will be able to easily and quickly spread the tips between the employees and reward senior waiters over new waiters for the same amount of time they worked.
   - 6.1) Calculating tips spread by entering invalid total tips amount and valid waiters’ names and percentages.
   - 6.2) Calculating tips spread by entering invalid waiter’s hour and valid total tips, names, and percentages.
   - 6.3) Calculating tips spread by entering invalid waiter’s percentage and valid waiters’ hours, names, and total tips.
   - 6.4) Calculating tips spread with valid total tips amount, waiters’ hours, and percentages when all the “name” text input fields are empty.
   - 6.5) Calculating tips spread with valid total tips amount, waiters’ hours and percentages when all the “name” text input fields are full.
   - 6.6) Calculating tips spread with valid total tips amount, waiters’ hours, percentages and names when the total affairs to the restaurant staff and the restaurant are bigger than the total tips that was made.
   - 6.7) Calculating tips spread with valid: total tips, waiter’s name, hours and percentage when there is only 1 waiter in the table and his percentage is smaller than 100% and checking the waiter gets all the remaining money from the total tips after affairs.

6.8) Checking that after a calculation the "Total Hours" output field shows the sum of all the waiter's hours in the table.

6.9) Checking that all the affairs to each role are calculated by the settings the user defined.

6.10) Checking that waiter's earnings per hour calculation is as follows:
waiters per hour = (total tips – total affairs) / sum(hour(i) * percentage(i)), $i \in \{1,...,n\}$, n is the number of waiters.

6.11) Checking that the amount of money each waiter in the table receives is calculated by: "Per Hour" earnings * waiter's hours * waiter's percentage

6.12) Checking that: total waiters earnings + total affairs = total tips when the maximum error range allowed is 1 (currency)

6.13) Checking that the transparency of the "ADD ROW" and "DELETE ROW" button increase when pressing the calculate button GUI)

7. As a user I want to be able to easily add waiters to the calculation so that I will be able to spread the tips between the actual number of waiters that worked in a specific shift.

7.1) Add a row to the table by clicking the add row button when not in the middle of a calculation.

7.2) Add a row to the table by clicking the add row button when in the middle of a calculation.

8. As a user I want to easily remove waiters from the calculation so that if I accidently added one extra waiter, I wouldn't need to reset what I did and start form beginning again.

8.1) Remove a row from the waiters table when the number of table rows is bigger than 1 and not in the middle of calculation.

8.2) Remove a row from the waiters table when the number of table rows is exactly 1.

8.3) Remove a row from the waiters table when the number of table rows is bigger than 1 and in the middle of calculation.

9. As a user I want to be able to calculate the amount of money that goes to each restaurant's employee and to the restaurant by the settings I defined, by the total tips made, by entering all employees hours of work and assigning each employee a role and optionally a name, so that I will be able to easily and quickly spread the tips between all employees based on the "percentage from total tips" assigned to them and their hours of work.

9.1) Calculating tips spread by entering invalid total tips amount and entering valid employee's name and hours of work and selecting a valid employee role.

9.2) Calculating tips spread by entering invalid employee's hours and entering valid total tips, and employee's name, and selecting a valid role.

9.3) Calculating tips spread by selecting an invalid employee's role and entering valid total tips and employee's hours of work.

9.4) Calculating tips spread by entering valid total tips amount, names and hours, and by selecting valid employees' roles when there are no waiters in the employees table.

9.5) Calculating tips spread by entering valid total tips amount, names and hours, and by selecting valid employees' roles when there is at least 1 waiter in the employee's table.

9.6) Calculating tips spread by entering valid total tips amount, names and hours, and by selecting valid employees' roles when the total affairs to the restaurant staff and the restaurant are bigger than the total tips that was made.

9.7) Calculating tips spread by entering valid total tips amount, names and hours, and by selecting valid employees' roles when the "name" text input fields are empty.

9.8) Checking that after a calculation the "Total Hours" output field of each role shows the sum of hours of all the employees in said role.

9.9) Checking that role's earnings per hour calculation is as follows:
role per hour = (role settings percentage * total tips) / total role hours
waiters per hour = (total tips – total affairs)/total waiters' hours

9.10) Checking that the amount of money each employee in the table receives is calculated by: "Role Per Hour" earnings * employee's hours (employee from the said role)

9.11) Checking that: total employee earnings = total tips when the maximum error range allowed is 1 (currency)

9.12) Checking that the transparency of the "ADD ROW" and "DELETE ROW" button increase when pressing the calculate button GUI)

10. As a user I want to be able to easily add restaurant employees to the calculation so that I will be able to spread the tips between the actual number of employees that worked in a specific shift.

10.1) Add a row to the table by clicking the add row button when not in the middle of a calculation.

10.2) Add a row to the table by clicking the add row button when in the middle of a calculation.

11. As a user I want to easily remove restaurant employees from the calculation so that if I accidently added one extra employee, I wouldn't need to reset the page and start form beginning again.

11.1) Remove a row from the employees table when the number of table rows is bigger than 1 and not in the middle of calculation.

11.2) Remove a row from the employees table when the number of table rows is exactly 1.

11.3) Remove a row from the employees table when the number of table rows is bigger than 1 and in the middle of calculation.

12. As a user I want to be able to calculate the amount of money that goes to each restaurant's employee by assigning each employee a specific amount of points, so that I will be able to easily divide the total tips received among the employees based on their role or contribution to the business.

12.1) Calculating tips spread by entering invalid employee's points and valid name and total tips.

12.2) Calculating tips spread by entering invalid total tips amount and valid name and points.

12.3) Calculating tips spread by entering valid: total tips amount and employees' points when the "name" text input field is empty.

12.4) Calculating tips spread by entering valid: total tips amount, employees' points and names.

12.5) Checking that after a calculation the "Total Points" output field shows the sum of points of all the employees in the table.

12.6) Checking that earnings per point calculation is as follows:
Per point = total tips / total points

12.7) Checking that the amount of money each employee in the table receives is calculated by: "Per Point" earnings * employee points.

12.8) Checking that: total employees' earnings = total tips when the maximum error range allowed is 1 (currency)

12.9) Checking that the transparency of the "ADD ROW" and "DELETE ROW" button increase when pressing the calculate button GUI)

13. As a user I want to be able to easily add restaurant employees to the calculation so that I will be able to spread the tips between the actual number of employees that worked in a specific shift.

13.1) Add a row to the table by clicking the add row button when not in the middle of a calculation.

13.2) Add a row to the table by clicking the add row button when in the middle of a calculation.

14. As a user I want to easily remove restaurant employees from the calculation so that if I accidently added one extra employee, I wouldn't need to reset the page and start form beginning again.

14.1) Remove a row from the employees table when the number of table rows is bigger than 1 and not in the middle of calculation.

14.2) Remove a row from the employees table when the number of table rows is exactly 1.

14.3) Remove a row from the employees table when the number of table rows is bigger than 1 and in the middle of calculation.

15. As a user I want to be able contact the company built the website by providing them with my name, the message and optionally my email so that I will be able to report an error I found or suggest about ideas for modification that matches the way my restaurant handles tips sharing and get a response from the company.

15.1) Sending a message with invalid name and valid email and message.

15.2) Sending a message with invalid message and valid mail and name.

15.3) Sending a message with valid name, message, and Email.

15.4) Sending a message with invalid email and valid message and name.

Side Notes for creating artificial bugs:

- We can change the hours section to be up to 20 hours (because employee hours can't surpass a certain number) and create a bug. (tests document, test case ID: N1)
- We can change the email (in the contact us section) to be only an exciting email and create a bug (tests document, test case ID: N3)
- In the "Staff Calculator" we can change the employee role in the table to be an input field in order to create a bug. (tests document, test case ID: N2)

Testing strategies:

Our testing strategy will be as follows:

We have a total of 6 modules for testing (Settings, Waiters Calculator, Percentage Calculator, Staff Calculator, Points Calculator and Contact Us) and a total of 83 requirements for testing

(See section: Requirements For Testing).

Because we can't divide the work between the team members by assigning each tester a module (we have 6 modules and each has different number of requirements for testing under it) we decided to divide the requirements for testing between the team members, meaning each member will have 20 requirements for testing and the 3 remaining requirements will be given to 3 random testers.

We will use different testing strategies to cover all needed requirements and after each member will finish the requirements assigned to him, each team member will check the tests of another team member in order to try detecting test errors.

What will be tested and what not:

- All the requirements under the "Requirements For Testing" section will be tested.
- Non-Functional Requirements under the Non-Functional requirements section who contain a test case ID near them will be tested, others won't be tested due to limitations
- The navigation bar of the website won't be tested because the models we chose to test already have enough requirements under them.

Schedule:

- 01/01/23 – Zoom meeting of all team member to go over the final project requirements and skeleton
- 02/01/23 – Frontal meeting of all team member for covering:
    1. The General section of the project – team members and website explanation.
    2. Division of the site into models for testing and user stories.
    3. Decision about the tests strategies
    4. Finishing the last subsections of the General section.
- 07/01/23 - Writing the requirements for testing under each model and dividing them between the team members.
-

- 10/01/23 – Zoom meeting of all team members (after each member has finished running his tests) discussing the tests and their results, and assigning each team member with a task to check another team member requirements and results.
- 11/01/23 - Frontal meeting of all team members for handling and reporting the bugs
- 12/01/23 – Frontal meeting of all team members for writing the conclusions and the manager summary
- 13/01/23 – Zoom meeting of all team members for making last finishes and submitting the project.

Testing Techniques:

We will preform black box testing on the website using Qase website/testing tables.

In our testing plan we used following testing techniques:

1. Equivalence partitions.
2. Borders testing.
3. Domain testing.

Tests Organization:

Risk Management:

- To prevent (As much as possible) test errors, meaning tests that had an error in the test itself, each team member will go over another's team member tests and will make sure there are no problems in the tests steps and results.

- In the short time provided us for making the tests, we won't be able to tests all website requirements and modules so we chose to focus on the most critical aspects of the website and cover all potential test cases in them.

- To avoid a lack of synchronization between the team members in the short time we was given to perform the tests we predefined a schedule containing the days we will meet on and the tasks to perform in each day/meeting.

- In order to avoid delay in the delivery of the task because of unplanned events we defined a "smart schedule", we defined a sooner schedule that will fulfill the assignment in a shorter amount of time so even if a meeting will get canceled because of unplanned events we will still have time to delay the meeting to.

Priorities:

We decided that our main priorities will focus on the main target of the website, which is to help mutual tips restaurant distribute the total tips was made in a shift between the employees that worked in the shift by providing different affairs system customized by the user(percentage of total tips for each role and hourly amount of money that a waiter pays the restaurant per hour) and

different systems to distribute the money between the waiters(by hours, by hours and percentage assignment to a waiter)or employees(by assigning certain number of point for each employee, by providing the employees role and hours).

Given this target, the critical malfunctioning of the website will be:

1. An error of calculations in a specific calculator – can lead to money problems, an employee or the restaurant can be affected by those errors, and this will be a critical error.
2. Error in the contact us section – this is the only sections a user can contact the creators, there is no email address or contact details, so a failing "contact us" model can lead to a lot of users' bug reports which the creators of the website won't know about which will cause user to abandon the use of the website.

So, in our tests we decided to focus on each calculator the website presents and the contact us model which we consider as the critical sections of the website.

In our testing we included the Settings section of the website, because 3 out of 4 calculators use the settings the user defined to make the tips spread calculations between the employees.

Human recourses:

The testers team will be made of 4 software engineering students, who are in their 3rd year of the first degree, the testers fully covered a quality and testing course.

Team members:

1. Lior Shilon
2. Elyasaf Sinvani
3. Sharon Angado
4. Shmuel Malikov

Testing Environment:

We will use our personal computer and the internet in order to connect the website and run the tests.

We will use Qase/test tables to record each test and its steps.

Team operating systems:

1. Windows 10/11
2. MacOS

Browsers:

1. Microsoft Edge Version 108.0.1462.54 (Official build) (64-bit).
2. Google Chrome Version 108.0.5359.125 (Official Build) (64-bit).

We will create a testing data for each test that covers all possible scenarios that can happen, all valid and invalid possibilities.

Testing Tools:

We will use test tables to perform the tests on the website, we will create a test table for every requirement or set of requirements that represents a “test unit” , each table will contain the steps we made in order to perform the test.

Success Metrics For The Project:

To finish the project successfully we will need to meet below goals:

1. Completing all the tests we defined in the requirements for testing section.
2. Reporting about all the bugs that was found in the process of testing.
3. Meeting the schedule we defined in the testing strategies section.
4. Site test passed –
    - Tests of high priority - 95% passing
    - Tests of medium priority- 90% passing
    - Tests of low priority - 80% - passing

Test Levels:

1. system testing, which is the process of testing a complete software system to ensure that it meets the specified requirements.
2. Acceptance testing: Acceptance testing is the process of testing a software system to ensure that it meets the acceptance criteria specified by the client or end-user.

The website we tested is already a complete system so we used the above in order to test the technical aspect of the website, to check that each calculator does it’s jobs and calculations are correct, we used functional tests to check what are the system’s functions and some GUI tests.

2. Testing Plan:

Domain testing:

Domain Testing – Settings

---

1. As a user I want to be able to set an hourly affair to the restaurant for waiters and a percentage of the total tips that goes to: kitchen staff, bartenders, shift manager, busser, expo, runners and hostess, so that I will able to create a "money affair system" that matches the restaurant I work in.

- **Domain testing 1** – The amount of money a waiter pays for an hour of work to the restaurant.
  The variable entering the domain – W, the amount {W >= 0}
  The required test – W >= 0 and W is integer.
  A1: W>= 0 and W is integer (valid)
  A2: W < 0 (invalid)
  A3: W = nonnumeric, character(invalid)
  A4: W = floating point number(invalid)
  A5: W = NULL (invalid)

A1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	W= -1	W >= 0	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	W = 0	W >= 0	The number will appear in the box and will be valid	Other attributes of the settings module will be checked

A2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	W= -1	W < 0	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	W = 0	W < 0	The number will appear in the box and will be valid	Other attributes of the settings module will be checked

A3(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	W= "12dd3"	W = nonnumeric	Error message	The settings won't be updated, it

					will remain the same

A4(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	W= 12.1	W = floating point number	Error message	The settings won't be updated, it will remain the same

A5(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	W = NULL	W = NULL point number	Error message	The settings won't be updated, it will remain the same

- **Domain testing 2** – The percentage of total tips assigned to the kitchen staff.
  The variable entering the domain – K, the percentage for kitchen {0 =< K < 100}
  The required test – 0 =< K < 100 {range of values}
  A6: 0 =< K < 100 and K is integer (valid)
  A7: K < 0 (invalid)
  A8: K >= 100 (invalid)
  A9: K = nonnumeric, character(invalid)
  A10: K = floating point number(invalid)
  A11: K = NULL (invalid)

A6(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	K = -1	0 =< K < 100	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	K = 0	0 =< K < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	3(upper limit)	K = 99	0 =< K < 100	The number will appear in the box	Other attributes of the settings

				and will be valid	module will be checked
	4(exceeding upper limit)	K = 100	0 =< K < 100	Error message	The settings won't be updated, it will remain the same

A7(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	K = -1	K < 0	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	K = 0	K < 0	The number will appear in the box and will be valid	Other attributes of the settings module will be checked

A8(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(Upper limit)	K= 99	K >= 100	Error message	The settings won't be updated, it will remain the same
	2(exceeding upper limit)	K = 100	K >= 100	Error message	The settings won't be updated, it will remain the same

A9(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	K = "e"	K = nonnumeric	Error message	The settings won't be updated, it will remain the same

A10(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	K = 45.3	K = floating point number	Error message	The settings won't be

					updated, it will remain the same

A11(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	K = NULL	K = NULL point number	Error message	The settings won't be updated, it will remain the same

- **Domain testing 3** – The percentage of total tips assigned to the bartenders.
  The variable entering the domain – B, the percentage for bartenders {0 =< B < 100}
  The required test – 0 =< B < 100 {range of values}
  A12: 0 =< B < 100 and B is integer (valid)
  A13: B < 0 (invalid)
  A14: B >= 100 (invalid)
  A15: B = nonnumeric, character(invalid)
  A16: B = floating point number(invalid)
  A17: B = NULL (invalid)

A12(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	B = -1	0 =< B < 100	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	B = 0	0 =< B < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	3(upper limit)	B = 99	0 =< B < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	4(exceeding upper limit)	B = 100	0 =< B < 100	Error message	The settings won't be updated, it will remain the same

A13(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	B = -1	B < 0	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	B = 0	B < 0	The number will appear in the box and will be valid	Other attributes of the settings module will be checked

A14(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(Upper limit)	B = 99	B >= 100	Error message	The settings won't be updated, it will remain the same
	2(exceeding upper limit)	B = 100	B >= 100	Error message	The settings won't be updated, it will remain the same

A15(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	B = "e"	B = nonnumeric	Error message	The settings won't be updated, it will remain the same

A16(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	B = 4.3	B = floating point number	Error message	The settings won't be updated, it will remain the same

A17(invalid)	Test Case	Parameter	Test	Expected Result	Meaning

	1(invalid value)	B = NULL	B = NULL point number	Error message	The settings won't be updated, it will remain the same

- **Domain testing 4** – The percentage of total tips assigned to the shift manager.
  The variable entering the domain – S, the percentage for shift manager {0 =< S < 100}
  The required test – 0 =< S < 100 {range of values}
  A18: 0 =< S < 100 and S is integer (valid)
  A19: S < 0 (invalid)
  A20: S >= 100 (invalid)
  A21: S = nonnumeric, character(invalid)
  A22: S = floating point number(invalid)
  A23: B = NULL (invalid)

A18(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	S = -1	0 =< S < 100	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	S = 0	0 =< S < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	3(upper limit)	S = 99	0 =< S < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	4(exceeding upper limit)	S = 100	0 =< S < 100	Error message	The settings won't be updated, it will remain the same

A19(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	S = -1	S < 0	Error message	The settings won't be updated, it

					will remain the same
	2(lower limit)	S = 0	S < 0	The number will appear in the box and will be valid	Other attributes of the settings module will be checked

A20(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(Upper limit)	S = 99	S >= 100	Error message	The settings won't be updated, it will remain the same
	2(exceeding upper limit)	S = 100	S >= 100	Error message	The settings won't be updated, it will remain the same

A21(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	S = "e"	S = nonnumeric	Error message	The settings won't be updated, it will remain the same

A22(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	S = 45.3	S = floating point number	Error message	The settings won't be updated, it will remain the same

A23(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	S = NULL	S = NULL point number	Error message	The settings won't be updated, it will remain the same

- **Domain testing 5** – The percentage of total tips assigned to the bussers.
  The variable entering the domain – Z, the percentage for bussers {0 =< Z< 100}
  The required test – 0 =< Z < 100 {range of values}
  A24: 0 =< Z < 100 and Z is integer (valid)
  A25: Z < 0 (invalid)
  A26: Z >= 100 (invalid)
  A27: Z = nonnumeric, character(invalid)
  A28: Z = floating point number(invalid)
  A29: Z = NULL (invalid)

A24(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	Z = -1	0 =< Z < 100	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	Z = 0	0 =< Z < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	3(upper limit)	Z = 99	0 =< Z < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	4(exceeding upper limit)	Z = 100	0 =< Z < 100	Error message	The settings won't be updated, it will remain the same

A25(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	Z = -1	Z < 0	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	Z = 0	Z < 0	The number will appear in the box and will be valid	Other attributes of the settings module will be checked

A26(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(Upper limit)	Z = 99	Z >= 100	Error message	The settings won't be updated, it will remain the same
	2(exceeding upper limit)	Z = 100	Z >= 100	Error message	The settings won't be updated, it will remain the same

A27(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	Z = "e"	Z = nonnumeric	Error message	The settings won't be updated, it will remain the same

A28(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	Z = 45.3	Z = floating point number	Error message	The settings won't be updated, it will remain the same

A29(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	Z = NULL	Z = NULL point number	Error message	The settings won't be updated, it will remain the same

- **Domain testing 6** – The percentage of total tips assigned to the expos.
  The variable entering the domain – E, the percentage for expos {0 =< E< 100}
  The required test – 0 =< E < 100 {range of values}
  A30: 0 =< E < 100 and E is integer (valid)
  A31: E < 0 (invalid)
  A32: E >= 100 (invalid)
  A33: E = nonnumeric, character(invalid)

A34: E = floating point number(invalid)
A35: E = NULL (invalid)

A30(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	E = -1	0 =< E < 100	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	E = 0	0 =< E < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	3(upper limit)	E = 99	0 =< E < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	4(exceeding upper limit)	E = 100	0 =< E < 100	Error message	The settings won't be updated, it will remain the same

A31(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	E = -1	E < 0	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	E = 0	E < 0	The number will appear in the box and will be valid	Other attributes of the settings module will be checked

A32(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(Upper limit)	E = 99	E >= 100	Error message	The settings won't be updated, it will remain the same

	2(exceeding upper limit)	E = 100	E >= 100	Error message	The settings won't be updated, it will remain the same

A33(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	E = "e"	E = nonnumeric	Error message	The settings won't be updated, it will remain the same

A34(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	E = 99.9	E = floating point number	Error message	The settings won't be updated, it will remain the same

A35(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	E = NULL	E = NULL point number	Error message	The settings won't be updated, it will remain the same

- **Domain testing 7** – The percentage of total tips assigned to the runners.
  The variable entering the domain – R, the percentage for runners {0 =< R < 100}
  The required test – 0 =< R < 100 {range of values}
  A36: 0 =< R < 100 and R is integer (valid)
  A37: R < 0 (invalid)
  A38: R >= 100 (invalid)
  A39: R = nonnumeric, character(invalid)
  A40: R = floating point number(invalid)
  A41: R = NULL (invalid)

A36(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	R = -1	0 =< R < 100	Error message	The settings won't be

					updated, it will remain the same
	2(lower limit)	R = 0	0 =< R < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	3(upper limit)	R = 99	0 =< R < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	4(exceeding upper limit)	R = 100	0 =< R < 100	Error message	The settings won't be updated, it will remain the same

A37(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	R = -1	R < 0	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	R = 0	R < 0	The number will appear in the box and will be valid	Other attributes of the settings module will be checked

A38(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(Upper limit)	R = 99	R >= 100	Error message	The settings won't be updated, it will remain the same
	2(exceeding upper limit)	R = 100	R >= 100	Error message	The settings won't be updated, it will remain the same

A39(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	R = “e”	R = nonnumeric	Error message	The settings won’t be updated, it will remain the same

A40(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	R = 15.5	R = floating point number	Error message	The settings won’t be updated, it will remain the same

A41(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	R = NULL	R = NULL point number	Error message	The settings won’t be updated, it will remain the same

- **Domain testing 8** – The percentage of total tips assigned to the hostess.
  The variable entering the domain – H, the percentage for hostess {0 =< H< 100}
  The required test – 0 =< H < 100 {range of values}
  A42: 0 =< H < 100 and H is integer (valid)
  A43: H < 0 (invalid)
  A44: H >= 100 (invalid)
  A45: H = nonnumeric, character(invalid)
  A46: H = floating point number(invalid)
  A47: H = NULL (invalid)

A42(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	H = -1	0 =< H < 100	Error message	The settings won’t be updated, it will remain the same
	2(lower limit)	H = 0	0 =< H < 100	The number will appear in the box and will be valid	Other attributes of the settings

					module will be checked
	3(upper limit)	H = 99	0 =< H < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	4(exceeding upper limit)	H = 100	0 =< H < 100	Error message	The settings won't be updated, it will remain the same

A43(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	H = -1	H < 0	Error message	The settings won't be updated, it will remain the same
	2(lower limit)	H = 0	H < 0	The number will appear in the box and will be valid	Other attributes of the settings module will be checked

A44(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(Upper limit)	H = 99	H >= 100	Error message	The settings won't be updated, it will remain the same
	2(exceeding upper limit)	H = 100	H >= 100	Error message	The settings won't be updated, it will remain the same

A45(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	H = "e"	H = nonnumeric	Error message	The settings won't be updated, it will remain the same

A46(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	H = 45.3	H = floating point number	Error message	The settings won't be updated, it will remain the same

A47(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	H = NULL	H = NULL point number	Error message	The settings won't be updated, it will remain the same

- **Domain testing 9** – The sum of all role's percentages.
  The variable entering the domain – SP = H + R + E + Z + S + B + K, total sum of percentages {SP < 100}
  The required test – SP < 100 ,the total sum of percentages is less than 100
  Note that SP can't do below 0 because it's the total sum of variables that can't be negative and their min value is 0(see above domain variables)
  A48: SP < 100 (valid)
  A49: SP >= 100 (invalid)

A48(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(Upper limit)	SP = 99	SP < 100	The number will appear in the box and will be valid	Other attributes of the settings module will be checked
	2(exceeding upper limit)	SP = 100	SP < 100	Error message	The settings won't be updated, it will remain the same

A49(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(Upper limit)	SP = 99	SP >= 100	The number will appear in the box and will be valid	Other attributes of the settings

					module will be checked
	2(exceeding upper limit)	SP = 100	SP >= 100	Error message	The settings won't be updated, it will remain the same

---

2.As a user I want all my defined settings to be saved, so that each time I visit the website or use different calculator I won't need to re-enter all the settings.

- **Domain testing 1** – When the user opens the website on the web the last settings that he saved should appear in the settings section.
  The variable entering the domain – S, the state of the settings when opening the page (state means the actual values the settings contain), Boolean(pass/fail)
  The required test – S = S' (the last settings that the user saved) {must be}
  B1: S = S' (valid)
  meaning the settings of the user are identical to the settings he saved last.
  B2: S != S' (invalid)
  meaning there is at least 1 field in the settings that differ from the settings the user saved last.

B1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	S	Opening the website in the browser after user defined the settings already	S = the settings the user has defined last	The settings was saved correctly in the database

B2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	S	S != the settings the user has defined last	Error message saying something went wrong with the settings	There is a problem with the database

Domain Testing – Waiters Calculator

3.As a user I want to be able to calculate the amount of money that goes to each restaurant's role and to the restaurant by the settings I defined, by entering all waiter's hours of work and optionally their names and by entering the total tips made, so that I will be able to spread the tips easily and quickly between the employees.

Note: Although the "Settings" are mentioned above, because we tested them in a different module we won't test validity of the settings, we will assume they are valid and test all other variables in the domain.

- **Domain testing 1** – Hours input.
  The variable entering the domain – H, waiter hours {H> 0}
  The required test – ∀i [(i ∈ {1,...,n}) → (H(i) > 0)] | (n - number of waiters)
  C1: ∀i [(i ∈ {1,...,n}) → (H(i) > 0)] (valid)
  C2: ∃i [H(i) <= 0] (invalid)
  C3: ∃i [H(i) = nonnumeric] (invalid)
  C4: ∃i [hour(i) = NULL] (invalid)

C1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	$H_i = 0$	$H_i > 0$	Error message	The calculations process won't proceed.
	2(lower limit)	$H_i = 0.1$	$H_i > 0$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

C2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	$H_i = 0$	$H_i <= 0$	Error message	The calculations process won't proceed.
	2(lower limit)	$H_i = 0.1$	$H_i < 0$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

C3(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	H_i = “e”	H_i = nonnumeric	Error message	The calculations process won’t proceed.

C4(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	H_i = NULL	H_i = NULL	Error message	The calculations process won’t proceed.

- **Domain testing 2** – Total tips input.
  The variable entering the domain – T, total tips {T > 0}
  The required test – T > 0
  C5: T > 0 (valid)
  C6: T <= 0 (invalid)
  C7: T = nonnumeric (invalid)
  C8 : T = NULL (invalid)

C5(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	T = 0	T > 0	Error message	The calculations process won’t proceed.
	2(lower limit)	T = 0.1	T > 0	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

C6(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	T = 0	T <= 0	Error message	The calculations process won’t proceed.
	2(lower limit)	T = 0.1	T <= 0	The number will appear	The calculations

				in the box and will be valid	process will proceed. Other fields will be checked

C7(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	T = “e”	T = nonnumeric	Error message	The calculations process won’t proceed.

C8(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	T = NULL	T = NULL	Error message	The calculations process won’t proceed.

- **Domain testing 3** – Name input.
  The variable entering the domain – N, waiter’s name
  The required test – Because name is an optional field the required test is to check that every value of N is accepted. (We inferred from the checks we ran on the website that this field is only to provide the user with an option to add a name to a waiter so the user will be able to identify him easily, otherwise than that the name field has no effect on the system)
  C9: N = NULL (valid)
  C10: N = numeric (valid)
  C11: N= String (valid)

C9(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = NULL	N = NULL	Name field will be valid	The calculations process will proceed. Other fields will be checked

C10(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = 11	N = numeric	The name will appear in the box and will be valid	The calculations process will proceed.

					Other fields will be checked

C11(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = “Omer”	N = String	The name will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

---

4.As a user I want to be able to easily add waiters to the calculation so that I will be able to spread the tips between the actual number of waiters that worked in a specific shift.

- **Domain testing 1** – Test add row button during calculation.
  The variable entering the domain – is_calculating = Boolean, indicating if we are in the middle of the calculations while pressing the add button.
  The required test – is_calculating = False, meaning we are not in the middle of a calculation.
  D1: is_calculating = False (valid)
  D2: is_calculating = True (invalid)

D1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(not in the middle of a calculation)	is_calculating = False	is_calculating = False	A new row, containing 3 columns will be added to the bottom of the table	The table will have 1 more row than the previous table.

D2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(in the middle of a calculation)	is_calculating = True	is_calculating = True	A new row won’t be added to the tale	The table will have the same number of row as the previous table.

5.As a user I want to easily remove waiters from the calculation so that if I accidently added one extra waiter, I wouldn't need to reset what I did and start form beginning again.

- **Domain testing 1** – Test remove row button during calculation.
  The variable entering the domain – is_calculating = Boolean, indicating if we are in the middle of the calculations while pressing the add button.
  The required test – is_calculating = False, meaning we are not in the middle of a calculation.
  E1: is_calculating = False (valid)
  E2: is_calculating = True (invalid)

E1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(not in the middle of a calculation)	is_calculating = False	is_calculating = False	Continue for checking the number of rows the table has	The action is valid but depend on the number of table rows

E2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(in the middle of a calculation)	is_calculating = True	is_calculating = True	The last row of the table won't be deleted	The table will have the same number of row as the previous table.

- **Domain testing 2** – The number of rows the table has.
  The variable entering the domain – N, the number of rows {N > 1}
  The required test – N > 1, meaning the user will be allowed to delete a table row only if the table has 2 rows or more.
  E3: N > 1 (valid)
  E4: N = 1 (invalid)

  Note: The number of rows (N) can't get below 1.

E3(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	N = 1	N > 1	Error message	The table will have the same number of

					row as the previous table.
	2(lower limit)	N = 2	N > 1	The last row of the table will be deleted	The table will have 1 less row than the previous table.

E4(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 The number of rows in the table is 1	N = 1	N = 1	Error message	The table will have the same number of row as the previous table.

Domain Testing – Percentage Calculator:

6.As a user I want to be able to calculate the amount of money that goes to each restaurant's role and to the restaurant by the settings I defined, by the total tips made, by entering all waiter's hours of work and optionally their names and by assigning a percentage to each waiter, so that I will be able to easily and quickly spread the tips between the employees and reward senior waiters over new waiters for the same amount of time they worked.

Note: Although the "Settings" are mentioned above, because we tested them in a different module we won't test validity of the settings, we will assume they are valid and test all other variables in the domain.

- **Domain testing 1** – Hours input.
  The variable entering the domain – H, waiter hours {H> 0}
  The required test – ∀i [(i ∈ {1,...,n}) → (hour(i) > 0)] | (n - number of waiters)
  F1: ∀i [(i ∈ {1,...,n}) → (H(i) > 0)] (valid)
  F2: ∃i [H(i) <= 0] (invalid)
  F3: ∃i [H(i) = nonnumeric] (invalid)
  F4: ∃i [H(i) = NULL] (invalid)

F1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	H_i= 0	H_i > 0	Error message	The calculations process

					won't proceed.
	2(lower limit)	$H_i = 0.1$	$H_i > 0$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

F2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	$H_i = 0$	$H_i <= 0$	Error message	The calculations process won't proceed.
	2(lower limit)	$H_i = 0.1$	$H_i < 0$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

F3(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	H_i = "e"	H_i = nonnumeric	Error message	The calculations process won't proceed.

F4(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	H_i = NULL	H_i = NULL	Error message	The calculations process won't proceed.

- **Domain testing 2** – Total tips input.
  The variable entering the domain – T, total tips {T > 0}
  The required test – T > 0
  F5: T > 0 (valid)
  F6: T <= 0 (invalid)
  F7: T = nonnumeric (invalid)
  F8: T = NULL (invalid)

F5(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	T = 0	T > 0	Error message	The calculations process won't proceed.
	2(lower limit)	T = 0.1	T > 0	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

F6(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	T = 0	T <= 0	Error message	The calculations process won't proceed.
	2(lower limit)	T = 0.1	T <= 0	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

F7(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	T = "e"	T = nonnumeric	Error message	The calculations process won't proceed.

F8(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	T = NULL	T = NULL	Error message	The calculations process won't proceed.

- **Domain testing 3** – Name input.
  The variable entering the domain – N, waiter's name

The required test – Because name is an optional field the required test is to check that every value of N is accepted. (We inferred from the checks we ran on the website that this field is only to provide the user with an option to add a name to a waiter so the user will be able to identify him easily, otherwise than that the name field has no effect on the system)
F9: N = NULL (valid)
F10: N = numeric (valid)
F11: N= String (valid)

F9(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = NULL	N = NULL	Name field will be valid	The calculations process will proceed. Other fields will be checked

F10(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = 11	N = numeric	The name will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

F11(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = “Omer”	N = String	The name will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

- **Domain testing 4** – Percentage input.
  The variable entering the domain – P, the percentage assigned to a waiter {0 <P< 100}
  The required test – ∀i [(i ∈ {1,...,n}) → (0 < P(i) < 100)] | (n - number of waiters) {range of values}
  F12: ∀i [(i ∈ {1,...,n}) → (0 < P(i) < 100)] (valid)
  F13: ∃i [P(i) <= 0] (invalid)
  F14: ∃i [P(i) >= 100] (invalid)
  F15: ∃i [P(i) = nonnumeric, character] (invalid)

F16: ∃i [P(i) = NULL] (invalid)

F12(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	$P_i = 0$	$0 < P_i < 100$	Error message	The calculations process won't proceed.
	2(lower limit)	$P_i = 0.1$	$0 < P_i < 100$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked
	2(upper limit)	$P_i = 99.9$	$0 < P_i < 100$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked
	2(exceeding upper limit)	$P_i = 100$	$0 < P_i < 100$	Error message	The calculations process won't proceed.

F13(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	$P_i = 0$	$P_i < 0$	Error message	The calculations process won't proceed.
	2(lower limit)	$P_i = 0.1$	$P_i < 0$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

F14(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	2(upper limit)	$P_i = 99.9$	$0 < P_i < 100$	The number will appear in the box	The calculations process will proceed.

				and will be valid	Other fields will be checked
	2(exceeding upper limit)	P_i = 100	0 < P_i < 100	Error message	The calculations process won't proceed.

F15(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	P_i = "e"	P_i = nonnumeric	Error message	The calculations process won't proceed.

F16(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	P_i = NULL	P_i = NULL	Error message	The calculations process won't proceed.

7.As a user I want to be able to easily add waiters to the calculation so that I will be able to spread the tips between the actual number of waiters that worked in a specific shift.

- **Domain testing 1** – Test add row button during calculation.
  The variable entering the domain – is_calculating = Boolean, indicating if we are in the middle of the calculations while pressing the add button.
  The required test – is_calculating = False, meaning we are not in the middle of a calculation.
  G1: is_calculating = False (valid)
  G2: is_calculating = True (invalid)

G1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(not in the middle of a calculation)	is_calculating = False	is_calculating = False	A new row, containing 4 columns will be added to the bottom of the table	The table will have 1 more row than the previous table.

G2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(in the middle of a calculation)	is_calculating = True	is_calculating = True	A new row won't be added to the tale	The table will have the same number of row as the previous table.

8.As a user I want to easily remove waiters from the calculation so that if I accidently added one extra waiter, I wouldn't need to reset what I did and start form beginning again.

- **Domain testing 1** – Test remove row button during calculation.
  The variable entering the domain – is_calculating = Boolean, indicating if we are in the middle of the calculations while pressing the add button.
  The required test – is_calculating = False, meaning we are not in the middle of a calculation.
  H1: is_calculating = False (valid)
  H2: is_calculating = True (invalid)

H1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(not in the middle of a calculation)	is_calculating = False	is_calculating = False	Continue for checking the number of rows the table has	The action is valid but depend on the number of table rows

H2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(in the middle of a calculation)	is_calculating = True	is_calculating = True	The last row of the table won't be deleted	The table will have the same number of row as the previous table.

- **Domain testing 2** – The number of rows the table has.
  The variable entering the domain – N, the number of rows {N > 1}

The required test – N > 1, meaning the user will be allowed to delete a table row only if the table has 2 rows or more.
H3: N > 1 (valid)
H4: N = 1 (invalid)

Note: The number of rows (N) can't get below 1.

H3(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	N = 1	N > 1	Error message	The table will have the same number of rows as the previous table.
	2(lower limit)	N = 2	N > 1	The last row of the table will be deleted	The table will have 1 less row than the previous table.

H4(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 The number of rows in the table is 1	N = 1	N = 1	Error message	The table will have the same number of rows as the previous table.

Domain Testing– Staff Calculator:

9.As a user I want to be able to calculate the amount of money that goes to each restaurant's employee and to the restaurant by the settings I defined, by the total tips made, by entering all employees hours of work and assigning each employee a role and optionally a name, so that I will be

able to easily and quickly spread the tips between all employees based on the "percentage from total tips" assigned to them and their hours of work.

Note: Although the "Settings" are mentioned above, because we tested them in a different module we won't test validity of the settings, we will assume they are valid and test all other variables in the domain.

- **Domain testing 1** – Hours input.
  The variable entering the domain – H, employee's hours of work {H> 0}
  The required test – ∀i [(i ∈ {1,...,n}) → (hour(i) > 0)] | (n - number of waiters)
  I1: ∀i [(i ∈ {1,...,n}) → (H(i) > 0)] (valid)
  I2: ∃i [H(i) <= 0] (invalid)
  I3: ∃i [H(i) = nonnumeric] (invalid)
  I4: ∃i [H(i) = NULL] (invalid)

I1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	$H_i = 0$	$H_i > 0$	Error message	The calculations process won't proceed.
	2(lower limit)	$H_i = 0.1$	$H_i > 0$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

I2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	$H_i = 0$	$H_i <= 0$	Error message	The calculations process won't proceed.
	2(lower limit)	$H_i = 0.1$	$H_i < 0$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

I3(invalid)	Test Case	Parameter	Test	Expected Result	Meaning

	1(invalid value)	H_i = "e"	H_i = nonnumeric	Error message	The calculations process won't proceed.

I4(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	H_i = NULL	H_i = NULL	Error message	The calculations process won't proceed.

- **Domain testing 2** – Total tips input.
  The variable entering the domain – T, total tips {T > 0}
  The required test – T > 0
  I5: T > 0 (valid)
  I6: T <= 0 (invalid)
  I7: T = nonnumeric (invalid)
  I8: T = NULL (invalid)

I5(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	T = 0	T > 0	Error message	The calculations process won't proceed.
	2(lower limit)	T = 0.1	T > 0	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

I6(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	T = 0	T <= 0	Error message	The calculations process won't proceed.
	2(lower limit)	T = 0.1	T <= 0	The number will appear in the box	The calculations process will proceed.

				and will be valid	Other fields will be checked

I7(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	T = “e”	T = nonnumeric	Error message	The calculations process won’t proceed.

I8(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	T = NULL	T = NULL	Error message	The calculations process won’t proceed.

- **Domain testing 3** – Name input.
  The variable entering the domain – N, employee’s name
  The required test – Because name is an optional field the required test is to check that every value of N is accepted. (We inferred from the checks we ran on the website that this field is only to provide the user with an option to add a name to a waiter so the user will be able to identify him easily, otherwise than that the name field has no effect on the system)
  I9: N = NULL (valid)
  I10: N = numeric (valid)
  I11: N= String (valid)

I9(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = NULL	N = NULL	Name field will be valid	The calculations process will proceed. Other fields will be checked

I10(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = 11	N = numeric	The name will appear in the box and will be valid	The calculations process will proceed.

					Other fields will be checked

I11(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = “Omer”	N = String	The name will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

- **Domain testing 4** – employee’s role.
  The variable entering the domain – R, employee’s role.
  The required test –
  ∀i [(i ∈ {1,...,n}) → (R(i) = Waiter, Kitchen, Bartender, Shift Manager, Busser, Expo, Runner, Hostess)] | (n - number of employees)
  Role is one of {Waiter, Kitchen, Bartender, Shift Manager, Busser, Expo, Runner, Hostess}, (set of input values).
  I10: ∀i [(i ∈ {1,...,n}) → (R(i) = Waiter | Kitchen | Bartender | Shift Manager | Busser | Expo | Runner | Hostess)] (valid)
  I11: ∃i [R(i) != Waiter ^ Kitchen ^ Bartender ^ Shift Manager ^ Busser ^ Expo ^ Runner ^ Hostess] (invalid)

I10(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 One of the valid values	R_i = “Waiter”	R_i = (Waiter | Kitchen | Bartender | Shift Manager | Busser | Expo | Runner | Hostess)	The role will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

I11(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	H_i = Mechanic	H_i != Waiter ^ Kitchen ^ Bartender ^ Shift Manager	Error message	The calculations process won’t proceed.

			^ Busser ^ Expo ^ Runner ^ Hostess		

- **Domain testing 5** – the number of waiters in the table.
  The variable entering the domain – NW, number of waiters in the table.
  The required test – NW >= 1, there is a row in the table that it's role is "Waiter", the number of waiters can't be zero
  I12: NW >= 1 (valid)
  I13: NW = 0 (invalid)

I12(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	NW = 0	NW >= 1	Error message	The calculations process won't proceed.
	2(lower limit)	NW = 1	NW >= 1	Calculation process will be proceeded	The number of waiters is valid
I13(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	NW = 0	NW = 0	Error message	The calculations process won't proceed.

10.As a user I want to be able to easily add restaurant employees to the calculation so that I will be able to spread the tips between the actual number of employees that worked in a specific shift.

- **Domain testing 1** – Test add row button during calculation.
  The variable entering the domain – is_calculating = Boolean, indicating if we are in the middle of the calculations while pressing the add button.
  The required test – is_calculating = False, meaning we are not in the middle of a calculation.
  J1: is_calculating = False (valid)
  J2: is_calculating = True (invalid)

J1(valid)	Test Case	Parameter	Test	Expected Result	Meaning

	1(not in the middle of a calculation)	is_calculating = False	is_calculating = False	A new row, containing 4 columns will be added to the bottom of the table	The table will have 1 more row than the previous table.

J2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(in the middle of a calculation)	is_calculating = True	is_calculating = True	A new row won't be added to the table	The table will have the same number of rows as the previous table.

11.As a user I want to easily remove restaurant employees from the calculation so that if I accidently added one extra employee, I wouldn't need to reset the page and start form beginning again.

- **Domain testing 1** – Test remove row button during calculation.
  The variable entering the domain – is_calculating = Boolean, indicating if we are in the middle of the calculations while pressing the add button.
  The required test – is_calculating = False, meaning we are not in the middle of a calculation.
  K1: is_calculating = False (valid)
  K2: is_calculating = True (invalid)

K1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(not in the middle of a calculation)	is_calculating = False	is_calculating = False	Continue for checking the number of rows the table has	The action is valid but depend on the number of table rows

K2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(in the middle of a calculation)	is_calculating = True	is_calculating = True	The last row of the table	The table will have the same

				won't be deleted	number of row as the previous table.

- **Domain testing 2** – The number of rows the table has.
  The variable entering the domain – N, the number of rows {N > 1}
  The required test – N > 1, meaning the user will be allowed to delete a table row only if the table has 2 rows or more.
  K3: N > 1 (valid)
  K4: N = 1 (invalid)

K3(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	N = 1	N > 1	Error message	The table will have the same number of rows as the previous table.
	2(lower limit)	N = 2	N > 1	The last row of the table will be deleted	The table will have 1 less row than the previous table.

K4(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 The number of rows in the table is 1	N = 1	N = 1	Error message	The table will have the same number of rows as the previous table.

Domain Testing – Points Calculator:

---

12. As a user I want to be able to calculate the amount of money that goes to each restaurant's employee by entering the total tips made and by assigning each employee a specific amount of points and potentially a name, so that I will be able to easily divide the total tips received among the employees based on their role or contribution to the business.

- **Domain testing 1** – Name input.
  The variable entering the domain – N, employee's name
  The required test – Because name is an optional field the required test is to check that every value of N is accepted. (We inferred from the checks we ran on the website that this field is only to provide the user with an option to add a name to a waiter so the user will be able to identify him easily, otherwise than that the name field has no effect on the system)
  L1: N = NULL (valid)
  L2: N = numeric (valid)
  L3: N= String (valid)

L1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = NULL	N = NULL	Name field will be valid	The calculations process will proceed. Other fields will be checked

L2(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = 11	N = numeric	The name will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

L3(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1	N = "Omer"	N = String	The name will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

- **Domain testing 2** – Total tips input.
  The variable entering the domain – T, total tips {T > 0}
  The required test – T > 0
  L4: T > 0 (valid)

L5: T <= 0 (invalid)
L6: T = nonnumeric (invalid)
L7: T = NULL (invalid)

L4(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	T = 0	T > 0	Error message	The calculations process won't proceed.
	2(lower limit)	T = 0.1	T > 0	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

L5(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	T = 0	T <= 0	Error message	The calculations process won't proceed.
	2(lower limit)	T = 0.1	T <= 0	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

L6(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	T = "e"	T = nonnumeric	Error message	The calculations process won't proceed.

L7(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	T = NULL	T = NULL	Error message	The calculations process won't proceed.

- **Domain testing 3** – employee's points.
  The variable entering the domain – P, employee's points. (P > 0)
  The required test –
  ∀i [(i ∈ {1,...,n}) → (P(i) > 0)] | (n - number of employees)
  L8: ∀i [(i ∈ {1,...,n}) → (P(i) > 0)] (valid)
  L9: ∃i [P(i) < 0] (invalid)
  L10: ∃i [P(i) = nonnumeric, character] (invalid)
  L11: ∃i [P(i) = NULL] (invalid)

L8(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	$P_i = 0$	$P_i > 0$	Error message	The calculations process won't proceed.
	2(lower limit)	$P_i = 0.1$	$P_i > 0$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

L9(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	$P_i = 0$	$P_i < 0$	Error message	The calculations process won't proceed.
	2(lower limit)	$P_i = 0.1$	$P_i < 0$	The number will appear in the box and will be valid	The calculations process will proceed. Other fields will be checked

L10(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	P_i = "e"	P_i = nonnumeric	Error message	The calculations process won't proceed.

L11(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(invalid value)	P_i = NULL	P_i = NULL	Error message	The calculations process won't proceed.

13.As a user I want to be able to easily add restaurant employees to the calculation so that I will be able to spread the tips between the actual number of employees that worked in a specific shift.

- **Domain testing 1** – Test add row button during calculation.
  The variable entering the domain – is_calculating = Boolean, indicating if we are in the middle of the calculations while pressing the add button.
  The required test – is_calculating = False, meaning we are not in the middle of a calculation.
  M1: is_calculating = False (valid)
  M2: is_calculating = True (invalid)

M1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(not in the middle of a calculation)	is_calculating = False	is_calculating = False	A new row, containing 3 columns will be added to the bottom of the table	The table will have 1 more row than the previous table.

M2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(in the middle of a calculation)	is_calculating = True	is_calculating = True	A new row won't be added to the table	The table will have the same number of rows as the previous table.

14.As a user I want to easily remove restaurant employees from the calculation so that if I accidently added one extra employee, I wouldn't need to reset the page and start form beginning again.

- **Domain testing 1** – Test remove row button during calculation.
  The variable entering the domain – is_calculating = Boolean, indicating if we are in the middle of the calculations while pressing the add button.
  The required test – is_calculating = False, meaning we are not in the middle of a calculation.
  N1: is_calculating = False (valid)
  N2: is_calculating = True (invalid)

N1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(not in the middle of a calculation)	is_calculating = False	is_calculating = False	Continue for checking the number of rows the table has	The action is valid but depend on the number of table rows

N2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(in the middle of a calculation)	is_calculating = True	is_calculating = True	The last row of the table won't be deleted	The table will have the same number of row as the previous table.

- **Domain testing 2** – The number of rows the table has.
  The variable entering the domain – N, the number of rows {N > 1}
  The required test – N > 1, meaning the user will be allowed to delete a table row only if the table has 2 rows or more.
  N3: N > 1 (valid)
  N4: N = 1 (invalid)

N3(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1(exceeding lower limit)	N = 1	N > 1	Error message	The table will have the same number of rows as the previous table.
	2(lower limit)	N = 2	N > 1	The last row of the table will be deleted	The table will have 1 less row than the previous table.

N4(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 The number of rows in the table is 1	N = 1	N = 1	Error message	The table will have the same number of rows as the previous table.

Domain Testing – Contact Us:

15.As a user I want to be able contact the company built the website by providing them with my name, the message and optionally my email so that I will be able to report an error I found or suggest about ideas for modification that matches the way my restaurant handles tips sharing and get a response from the company.

- **Domain testing 1** – Name input.
  The variable entering the domain – N, the name of the sender {N > 1}

The required test – N = String (a set of characters, only spaces and letters allowed)
O1: N = String containing only letters and spaces (valid)
O2: N = String containing invalid characters, such as: !#@2 etc… (invalid)
O3: N = NULL (invalid)

O1(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 Valid name	N = Or Barak	N = String containing only letters and spaces	The value will be valid, Email input will be checked	The sending of the message will depend on the validity of the next input field (the Email)

O2(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 Invalid value	N = dfr555h	N = String containing invalid characters	Error message	The process of sending the message will not proceed

O3(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 Invalid value	N = dfr555h	N = NULL	Error message	Message sending process will not proceed

- **Domain testing 2** – Email input.
  The variable entering the domain – E, the Email of the sender (a String) {N > 1}
  The required test – E = NULL or (E = *+ "@" + * and 0 < E.length_after_@ < 64) (if the user chooses to provide his email the input must contain a @, at least 1 character from the right of the @, at least 1 character from the left of the @ and no more than 64 characters from the left of the @)
  O4: (E = *+ "@" + *) and (0 < E.length_after_@ < 64) (valid)
  O5: E = NULL (valid)
  O6: E = String that doesn't contain "@" (invalid)
  O7: (E = *+ "@" + *) and (E.length_after_@ >= 64) (invalid)
  O8: E = "@" + * , There are no letters from the left of the @ (invalid)
  O9: E = * + "@" , There are no letters from the right of the @ (invalid)

O4(valid)	Test Case	Parameter	Test	Expected Result	Meaning

	1 Valid Email	E = lior199707@walla .com	(E = *+ “@” + *) and (0 < E.length_after_@ < 64	The Email will be valid, message input will be checked	The sending of the message will depend on the validity of the next input field (the message)
	2 (upper limit) E.length_after_@ = 63	E = lior@wallaaaaaaa aaaaaaaaaaaaaa aaaa wallaaaaaaaaaaaa aaaaaaaaaaaaaa aa	(E = *+ “@” + *) and (0 < E.length_after_@ < 64	The Email will be valid, message input will be checked	The sending of the message will depend on the validity of the next input field (the message)
	3 Exceeding upper limit	E = lior@wallaaaaaaa aaaaaaaaaaaaaa aaaa wallaaaaaaaaaaaa aaaaaaaaaaaaaa aar	(E = *+ “@” + *) and (0 < E.length_after_@ < 64	Error message	The process of sending the message will not proceed

O5(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 Valid Email	E = NULL	E = NULL	The Email will be valid, message input will be checked	The sending of the message will depend on the validity of the next input field (the message)

O6(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 Invalid Email	E = lior199707walla.com	E = String that doesn’t contain “@”	Error message	The process of sending the message will not proceed

O7(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	2 (upper limit) E.length_after_@ = 63	lior@wallaaaaaaa aaaaaaaaaaaaaa aaaa wallaaaaaaaaaaaa aaaaaaaaaaaaaa aa	(E = *+ “@” + *) and (E.length_after_@ >= 64)	The Email will be valid, message input will be checked	The sending of the message will depend on the validity of the next input field (the message)

	3 Exceeding upper limit	lior@wallaaaaaaa aaaaaaaaaaaaaaa aaaa wallaaaaaaaaaaaa aaaaaaaaaaaaaaa aar	(E = *+ “@” + *) and (E.length_after_ @ >= 64)	Error message	The process of sending the message will not proceed

O8(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 Invalid Email	E = @gmail.com	E = “@” + * , There are no letters from the left of the @	Error message	The process of sending the message will not proceed

O9(invalid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 Invalid Email	E = lior@	E = * + “@” , There are no letters from the right of the @	Error message	The process of sending the message will not proceed

- **Domain testing 3** – Message input.
  The variable entering the domain – M, the message of the sender {N > 1}
  The required test – M != NULL
  O10: M != NULL (valid)
  O11: M = NULL(invalid)

O10(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 Valid message	M = “Hi”	M != NULL	The message will be sent	If we have reached to the message check it means that all previous variables (name and Email) are valid so the message will be checked

O11(valid)	Test Case	Parameter	Test	Expected Result	Meaning
	1 Invalid message	M = NULL	M = NULL	Error message	The message won’t be sent

General Note:

A note regarding all test domain that belong to a user story, for each user story, for each test domain under it, all the variables entering the domain need to be valid in order to perform the task the user story provides the user with.

It means if one of the variables belongs to a domain under a user story is invalid than the action won't perform even if all other environment variables under the same use story are valid.

Testing Techniques We Used:

We used GUI tests and input tests, we focused more on tests that serves the website's main goal which are the inputs for the calculation and the ability to add employees/waiters, each calculator has it's own customized table on it's web page, so in order to really be able to spread the tips between employees there must be a way to add and remove employees from each table.

We tested the validity of the calculations, we read the "How To Use" page of the website and leaned how the calculators work so we could go into how the calculators really calculate the final result, and tested the final results on the calculators based on the formulas we were able to infer from the explanation about the calculations.

We used:

- **Division into Domain test** areas - Its function is to divide the whole test into different test areas, so that each area constitutes one functional unit. The division into test areas was according to input or action in the system , When we referred to several fields / inputs behaving the same in one test area.

- **Division of equivalence classes** - Dividing the product's features into equivalence classes helps to identify test cases that can be used to validate the product's behavior. An

equivalence class is a set of input or output values that are expected to behave in the same way for a given unit of functionality.
By creating equivalence classes, we can create a set of test cases that cover a wide range of functionality with a relatively small number of tests. This can be particularly useful in cases where it is infeasible to test every possible input or output value for a given unit of functionality.
We divided the departments according to the valid, invalid classification.

- **End conditions** - Using end conditions in testing can help to identify and fix problems with software more efficiently by focusing on the parts of the software that are most likely to be problematic, The assumptions are that the behavior in the end values is more incorrect and problematic than the values in the center.
  We will separate tests that are in the valid range, which will give us positive results and values that are out of the valid range and according to the characterization should give us an error message.

Bugs/Defects management:

1.Urgency and hardware table level.

usually the data in urgency table is matching accordingly to the data in the hardware table as follows:

Immediate and High go together.

Soon and Medium go together.

No Rush and Low go together.

Urgency level	Code
immediate	I
soon	S
No rush	N

Hardware level	Code
High ↑	1
Medium ◎	2
Low ↓	3

2.Bug status table.

Status	Code
New	Ne
Open	op
Assigned	As

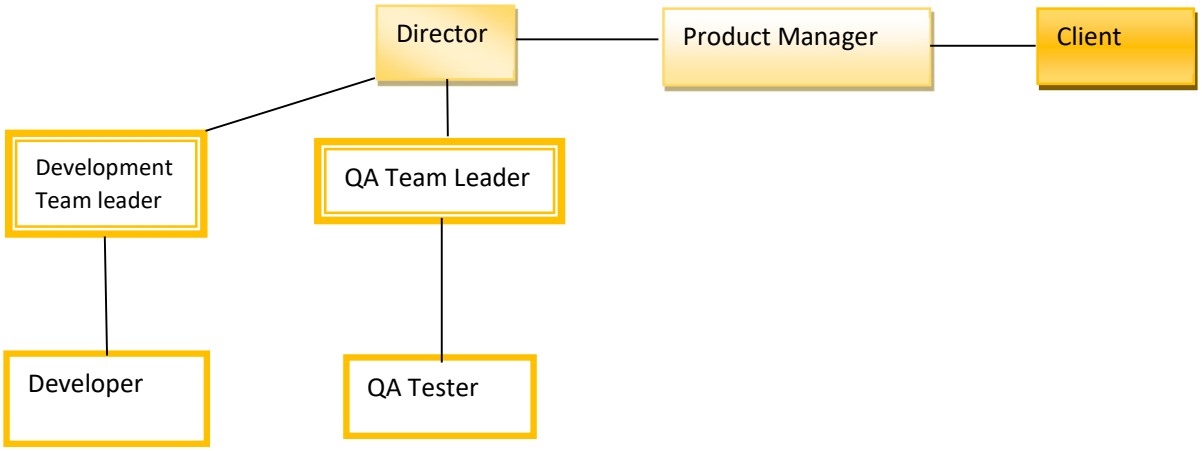
investigate	In
Fixed	Fi
Retest	Ret
deferred	Def
Closed	cl
Not bug	Nb
Rejected	Rej
Known	Kn
Pass	Ps
Reopen	Reop

3.Employees and responsibilities.

Client – Mutual Tips Restaurants who use the website .

Director – The project manager, responsible for monitoring and planning of the project and leading the team.
Development Team Leader – Responsible for the developer team.
QA Team Leader – Responsible for the QA team.
QA Tester – Responsible for testing the product.
Developer – Responsible for developing the product.
Product Manager – Responsible for translating the client's request to actual requirements for the project, and keep in touch with the client about the product.

Employee	Open	Close	Fix	Check	Status update
Client	+	-	-	-	-
Director	+	+	+	+	+
Development Team leader	-	+	+	+	+
QA Team Leader	+	+	-	+	+
QA Tester	+	-	-	+	+
Developer	-	-	+	-	+
Product Manager	+	+	-	-	+



4.Bug handling diagram.

New - This state represents a new bug detected by the client,tester or product manager, this bug than awaits for approval by the Director(project manager).

Open – The bug is identified, by the QA team leader, as a problem that needs to be fixed and awaits for assignment by the developer team leader.

Assigned – The bug is assigned to a developer by the developer team manager, after the developer team manager approves the bug.

investigate – The bug is investigated by a developer to determine its cause and potential solutions. This may involve reproducing the bug, analyzing log files, and consulting with other team members.

Fixed: The bug is fixed by a developer and awaits for retesting.

Retest: The bug is tested again by the QA tester to ensure that it properly resolves the issue and does not introduce any new problems.

Reopen: If the retest is unsuccessful, the process returns to the investigation step.

Pass: There is no longer a bug , awaits for QA team leader approvement.

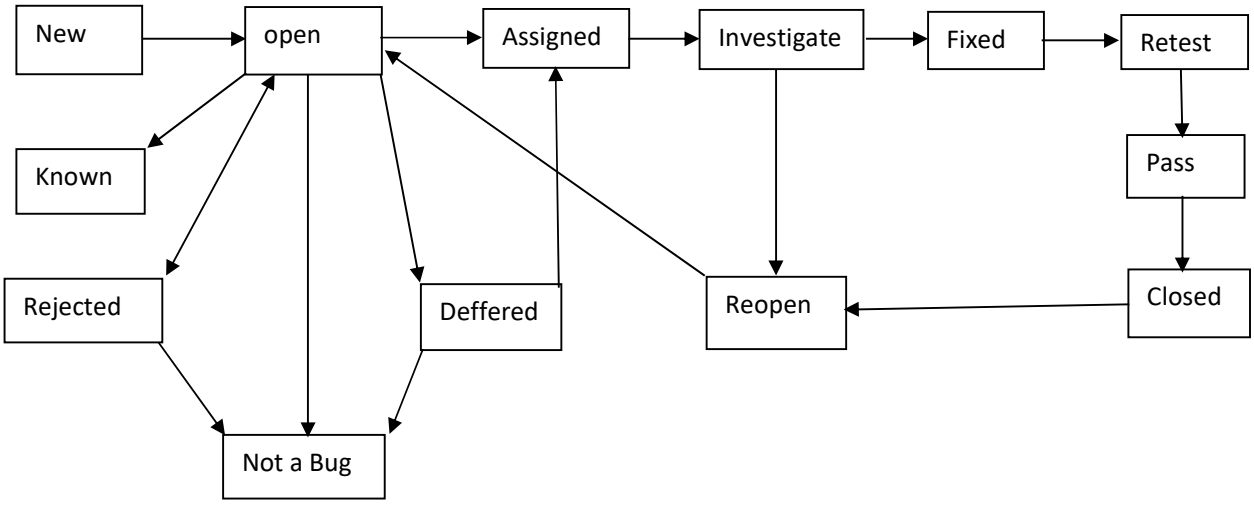
Closed: The bug is marked as resolved and closed in the bug tracking system by the QA team leader.

Not bug : The bug isn't really a bug according to system's requirements, only the director(project manager) will update it's status to "Not Bug".

Deferred : The bug is not urgent so we can handle it later.

Rejected: The QA team leader or the developer team leader thinks the bug isn't really a bug, awaits for director approval.

Known: The bug is already known and reported.

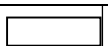


Bug handling diagram explained.

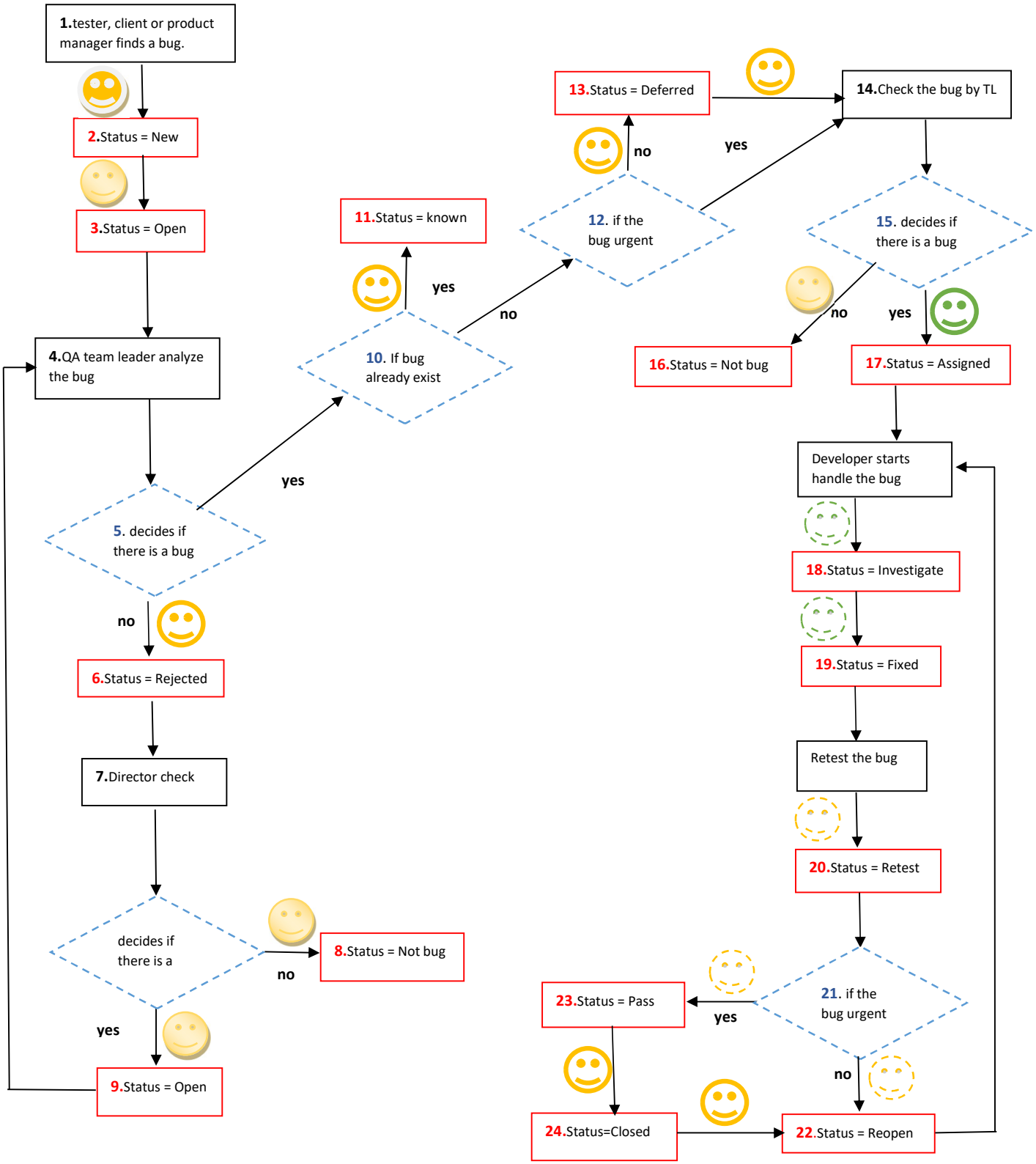
1. A tester, client or product manager finds a bug.
2. The status of the bug is updated to New.
3. The director approves the bug and the status of the bug is updated to Open.
4. The QA team leader will analyze the bug.
5. The QA team leader decides if there is a bug or not.
6. If the QA team leader decides that there is no bug he will update the bug status to Rejected.
7. The director will check the rejected bug.
8. If the director agrees with the QA team leader he will update the bug status to Not Bug.
9. If the director disagrees with the QA team leader he will update the bug status to Open.
10. If the QA team leader decides that there is a bug he will check if the bug is already in the bug tracking system.
11. If the bug is already in the bug tracking system the QA team leader will change its status to Known.

12. If the QA team leader decides the bug isn't registered in the tracking system he will check if the bug is urgent.
13. If the bug is not urgent to deal with, the QA team leader will change its status to Deferred, when the QA team leader decides that the bug should be handled he will contact the developer team leader and will wait for the developer team leader to check the bug and assign it to a developer.
14. If the bug was found urgent by the QA team leader or the QA team leader decides it's time to handle the bug (the status of the bug was deferred) , he will contact the developer team leader about the bug.
15. The developer team leader checks the bug.
16. If the developer team leader decides that the bug isn't really a bug he will contact the director for changing the status of the bug to Not Bug.
17. After the bug is approved by the developer team leader, he will give the bug to the development team and the status of the bug will update to Assigned.
18. The development team will start investigating the bug, the bug's status will be updated to Investigate.
19. After the development team fixed the bug, the bug's status will be updated to Fixed.
20. The development team will send the bug to the tester for a retest, the tester will update the bug's status to Retest.
21. The tester checks if the bug is fixed or if there are new bugs occurred from the change of the code.
22. If the bug wasn't fixed or there are new bugs found, the tester will send the bug to the development team again and the tester will update the status of the bug to Reopen.
23. If the tester finds that the bug was fixed, and that there are no new bugs he will change the status of the bug to Pass and will await for the QA team leader confirmation.
24. The QA team leader confirms that the bug was fixed and changes the bug's status to Close. Another case is when the bug is closed and for some reason we the closed bug was detected again so instead of going through the chain of responsibilities from the start, the QA team leader will change the bug's status to Reopen.

*****Employee – all the characters.**

									
Employee	Director	TL QA	QA	TL Developer	Developer	Action	Bug Status	decision	Next step

Bug lifecycle diagram



3. Tests Results:

- The test tables are all in the second documents called “tests”.

Bugs Found:

Real Bugs:

9.6) Calculating tips spread by entering valid total tips amount, names and hours, and by selecting valid employees’ roles when the total affairs to the restaurant staff and the restaurant are bigger than the total tips that was made.

Table test case ID: 9.1 – 9.7, step 33.

15.1) Sending a message with invalid name and valid email and message.

Table test case ID: 15.1 – 15.4, step 6.

Made Up Bugs:

- All tables and new user stories for creating the bugs can be seen in the tests document containing all the tests tables, under the header “Made Up Bugs” at the bottom of the document

N1) Calculating tips spread by entering valid total tips amount and waiters name, and invalid waiter’s hour which is above 12 hours.

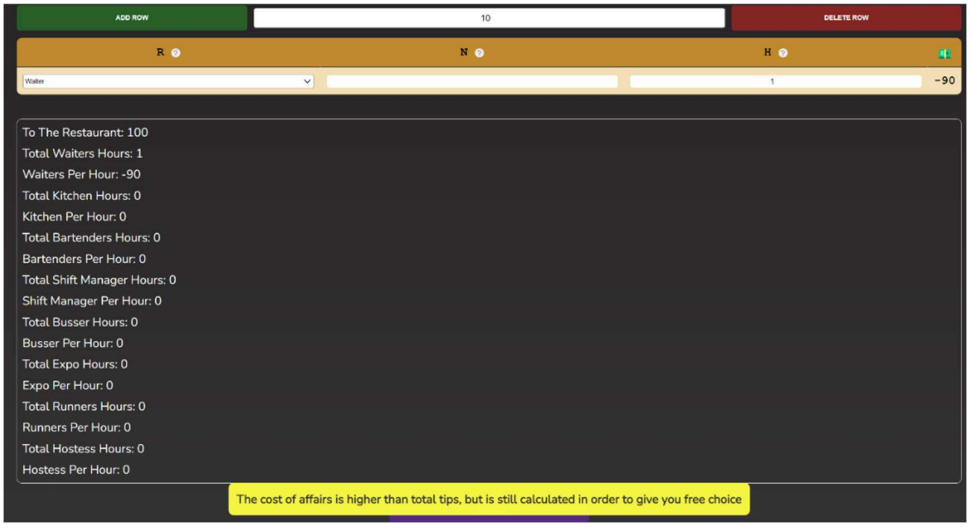
N2) Editing the employee’s role by typing it’s new role.

N3) Sending a message with a non-existing Email.

Bug Report forms:

9.6) Calculating tips spread by entering valid total tips amount, names and hours, and by selecting valid employees' roles when the total affairs to the restaurant staff and the restaurant are bigger than the total tips that was made.

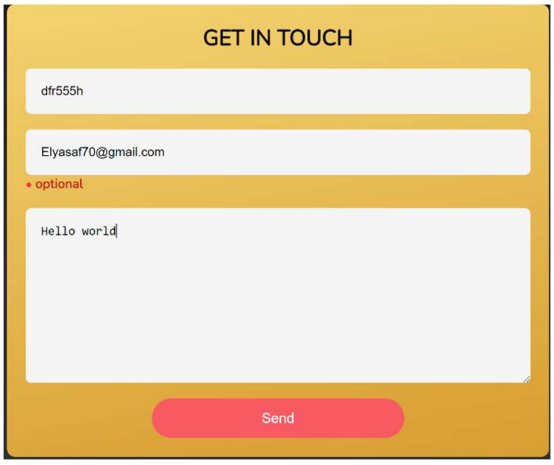
Error Title	Negative amount of money assigned to a waiter during calculation
Error ID	E1
Opener Identity	Shmuel Malikov (Tester)
Opening Date	12/01/23
Approver Identity	SCE quality and testing course lecturer team
Current Status	New(ne)
Handling Priority	Immediate(I)
Severity	High(3)
Details regarding the testing environment	The Staff Calculator page of the website, using google chrome
Error Description	In the Staff Calculator when the total sum of the affairs to each restaurant role is higher than the total tips value the per hour earnings of the waiters calculated as a negative value which results a negative amount of money assigned to waiters in the table
Final Status	Assigned(As)
Deadline	20/01/23



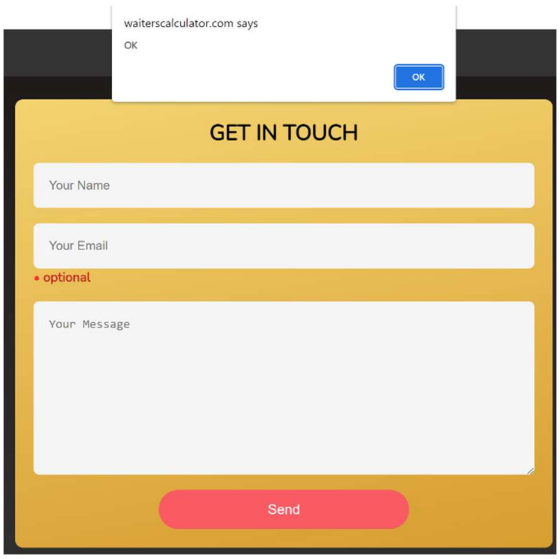
15.1) Sending a message with invalid name and valid email and message.

Error Title	Succeeding sending a mail from the contact us form with invalid name
Error ID	E2
Opener Identity	Elyasaf Sinvani (Tester)
Opening Date	12/01/23
Approver Identity	SCE quality and testing course lecturer team
Current Status	New(ne)
Handling Priority	Soon(S)
Severity	Medium (2)
Details regarding the testing environment	The Contact Us page of the website, using google chrome
Error Description	In the Contact Us page, under the name input field we entered : “dfr555h” and succeeded to send a mail using an invalid name input
Final Status	Not Bug (Nb)
Deadline	29/01/23

Before pressing the send button

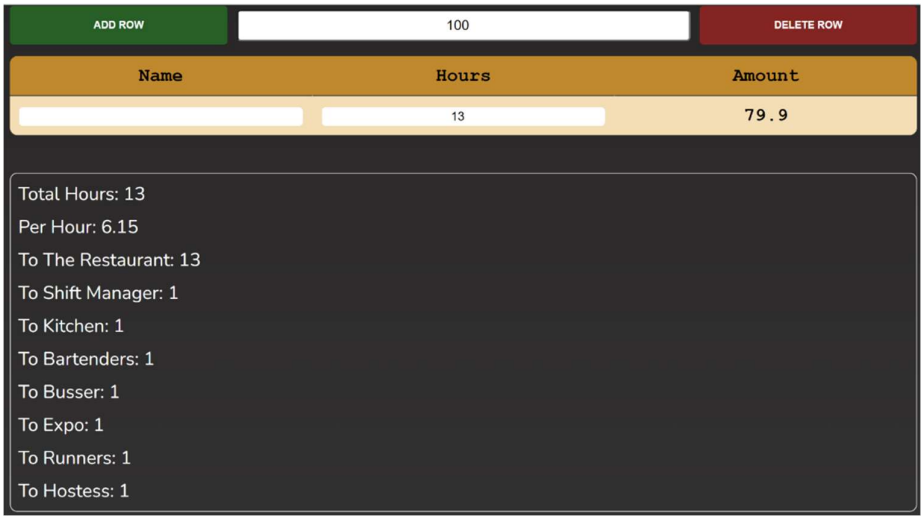


After pressing the send button



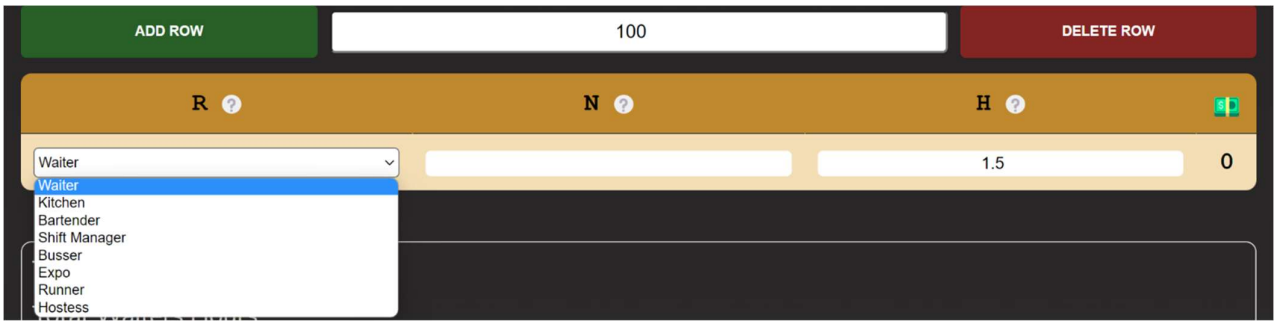
N1)Calculating tips spread by entering valid total tips amount and waiters name, and invalid waiter's hour which is above 12 hours.

Error Title	Hours sections larger than 12 is considered valid by the waiters calculator
Error ID	E3
Opener Identity	Sharon Angado (Tester)
Opening Date	12/01/23
Approver Identity	SCE quality and testing course lecturer team
Current Status	New(ne)
Handling Priority	Soon(S)
Severity	Medium (2)
Details regarding the testing environment	The Waiters Calculator page of the website, using Microsoft Edge
Error Description	In the Waiters Calculator page, in the hours section of the table we entered the value 13, when the maximum valid hours should be 12
Final Status	Open (op)
Deadline	29/01/23



N2) Editing the employee's role by typing it's new role.

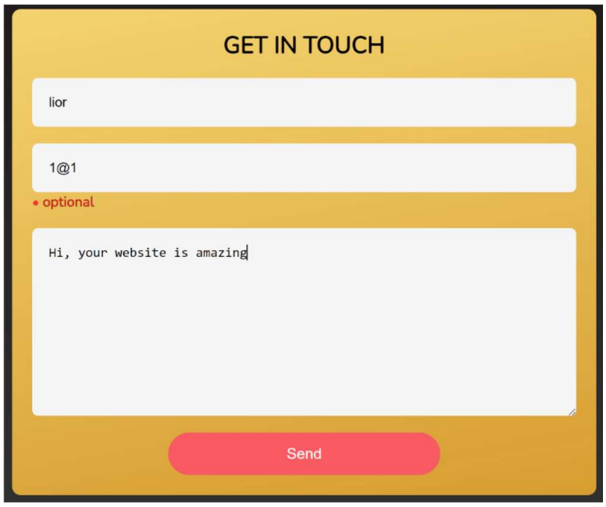
Error Title	Employee's role can't be inserted by input
Error ID	E4
Opener Identity	Lior Shilon (Tester)
Opening Date	12/01/23
Approver Identity	SCE quality and testing course lecturer team
Current Status	New(ne)
Handling Priority	Immediate (I)
Severity	High (3)
Details regarding the testing environment	The Staff Calculator page of the website, using google chrome
Error Description	The role selection in the table is presented as a choice form a combo box and not as a free input field
Final Status	Assigned(As)
Deadline	20/01/23



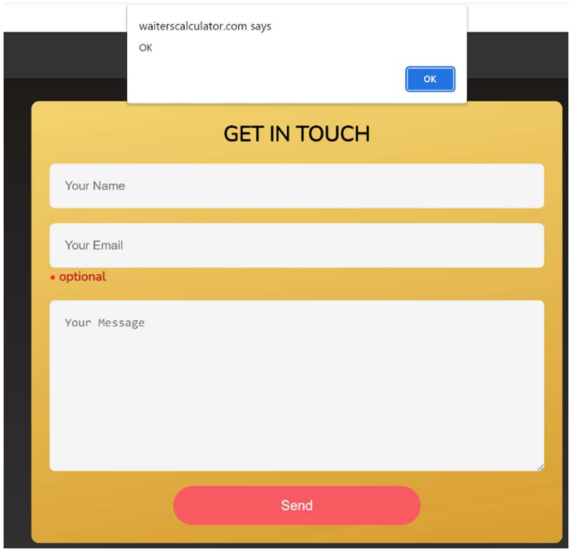
N3) Sending a message with a non-existing Email.

Error Title	Succeeding sending a mail from the contact us form with non-existing Email address
Error ID	E5
Opener Identity	Lior Shilon (Tester)
Opening Date	12/01/23
Approver Identity	SCE quality and testing course lecturer team
Current Status	New(ne)
Handling Priority	Immediate (I)
Severity	High (3)
Details regarding the testing environment	The Contact Us page of the website, using Microsoft Edge
Error Description	In the Contact Us page, under the Email input field we entered : "1@1" and succeeded to send a mail using an non-existing Email
Final Status	Assigned (As)
Deadline	20/01/23

Before pressing the send button



After pressing the send button



4. Conclusions and Recommendations:

Website Recommendations:

We found that the website is compliance with the testing requirements and meets the testing goals we defined.

The percentage of the High priority tests that passed is: 24/25 * 100 = 96%

The percentage of the Medium priority tests that passed is: 9/10 * 100 = 90%

The percentage of the Low priority test that passed is: 4/4 * 100 = 100%

We suggest to fix as soon as possible the bug: E1 (Bug ID) we found.

We suggest to work on the website's design with an emphasis on all the calculator tables, their design looks a bit old.

Conclusions and Recommendations for the testing strategies:

- The main functionality of the website is the calculators themselves and the settings, which are all based on user input fields and a dependency between them (because each input field holds a value that is a part of the calculation).
  Therefore, the test areas technique proved to be very helpful in covering all the tests cases in each module.

- Because each tester checked another's tester tests, we were able to identify issues in the tests and fix them.

- Using the end conditions strategy helped us to avoid a lot of possible tests combinations.

Personal Lessons:

- Using the test strategies we mentioned helped us finish the tests very fast and achieve goof test results, we even found some real bugs.

- Scheduling the meetings in a very short period of delays helped us stay on the same line by updating each other on the progress of each team member.

- By doing the major part of the work together and dividing only the tests between the team members everyone felt they contributed their share to the project.