Student 2 Testing Report



**Group Number:** C2.037  
**Repository:** <https://github.com/DP2-C1-037/Acme-ANS-C2>

**Workgroup Members:**

Student 2: Adrián Chabrera Rubio - [adrcharub@alum.us.es](mailto:adrcharub@alum.us.es)

**Date: 02/07/2025**

Table of Contents

[1 Executive Summary 2](#_Toc202444873)

[2 Revision Table 3](#_Toc202444874)

[3 Introduction 4](#_Toc202444875)

[4 Contents 5](#_Toc202444876)

[4.1 Functional testing 5](#_Toc202444877)

[4.1.1 Passenger functionalities functional testing 5](#_Toc202444878)

[4.1.2 Booking functional testing 11](#_Toc202444879)

[4.1.3 Booking assignations (intermediate entity) functional testing 15](#_Toc202444880)

[4.1.4 Total coverage 16](#_Toc202444881)

[4.2 Performance testing 17](#_Toc202444882)

[4.2.1 Computer A results 18](#_Toc202444883)

[4.2.2 Computer B results 19](#_Toc202444884)

[4.2.3 Results comparison 20](#_Toc202444885)

[5 Conclusions 21](#_Toc202444886)

[6 Bibliography 21](#_Toc202444887)

# Executive Summary

This Testing Report outlines the methodology and structure for evaluating the functionality and performance of the project through systematic testing procedures. It is divided into two main chapters: functional testing and performance testing.

The functional testing chapter presents a comprehensive list of implemented test cases, systematically grouped by feature. Each test case includes a concise description and a clear assessment of its effectiveness in detecting software bugs. The generation of this chapter relies heavily on structured planning and tool-assisted execution using Eclipse, ensuring consistency and minimizing the risk of oversight or disorganization.

The performance testing chapter provides quantitative insights into the system’s responsiveness. It includes detailed charts and a 95% confidence interval for the wall time required to serve requests on two different computers. Furthermore, it performs a 95% confidence hypothesis test to determine which of the two machines exhibits superior performance.

This report serves as a foundation for evaluating both the reliability and efficiency of the software. By following a rigorous and reproducible approach, it ensures high-quality outcomes and provides meaningful insights into the system's behavior under test conditions.

# Revision Table

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date** | **Description** |
| 1.0 | 24/05/2025 | Initial Draft |
| 2.0 | 25/05/2025 | Functional testing chapter |
| 3.0 | 26/05/2025 | Performance testing chapter and conclusions |
| 4.0 | 02/07/2025 | Second call revision |

# Introduction

This Testing Report introduces the objectives, scope, and approach used to evaluate the project's functionality and performance. It sets the context for the testing process and explains the rationale behind the chosen methods and structure.

The report begins by outlining the goals of functional testing, which focus on verifying that each feature of the software operates as intended. It emphasizes the importance of systematic test case design, guided by planning tools and the Eclipse environment, to ensure comprehensive coverage and accurate detection of software bugs.

Next, the report introduces the purpose of performance testing, aimed at assessing how efficiently the system handles requests under different conditions. The analysis includes statistical metrics such as 95% confidence intervals and hypothesis testing to compare results from two different computing environments, or a simulated equivalent when necessary.

This introduction provides a roadmap for the report and establishes the foundational principles of the testing process. By defining the scope and methodology upfront, it ensures that the testing activities are purposeful, reproducible, and aligned with the project's quality assurance goals.

# Contents

## Functional testing

A complete test suite has been developed to ensure that requirements that involve booking and passenger functionalities are met. These test suites are divided into a long list of test cases that have been tested one by one. The list that contains all these tests is following up, classifying them whether they are positive, negative or illegal (a hacking technique had to be made to test it out).

Test cases are divided by entity (Passenger, Booking and AssignedTo) and by functionality (list, show, create, update, publish and delete).

### Passenger functionalities functional testing

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test case** | **Type** | **Bugs detected** |
| **LIST** | | | |
| P1 | List passengers | Positive | No bugs detected |
| P2 | Try to list passengers with an anonymous realm | Hacking | No bugs detected |
| **SHOW** | | | |
| P3 | Show a passenger information | Positive | No bugs detected |
| P4 | Try to show a passenger with an anonymous realm | Hacking | No bugs detected |
| P5 | Try to show a passenger with right realm but wrong user | Hacking | No bugs detected |
| P6 | Try to show a passenger that doesn´t exist | Hacking | No bugs detected |
| **CREATE** | | | |
| P7 | Create a passenger with null data | Negative | No bugs detected |
| P8 | Create a passenger with base full name length | Positive | No bugs detected |
| P9 | Create a passenger with minimum full name length | Positive | No bugs detected |
| P10 | Create a passenger with minimum + 1 full name length | Positive | No bugs detected |
| P11 | Create a passenger with maximum – 1 full name length | Positive | No bugs detected |
| P12 | Create a passenger with maximum full name length | Positive | No bugs detected |
| P13 | Create a passenger with maximum + 1 full name length | Negative | No bugs detected |
| P14 | Create a passenger with non-latin full name characters | Positive | No bugs detected |
| P15 | Create a passenger with an injection full name | Positive | No bugs detected |
| P16 | Create a passenger with a valid email | Positive | No bugs detected |
| P17 | Create a passenger with an invalid email | Negative | No bugs detected |
| P18 | Create a passenger with a correct passport number format | Positive | No bugs detected |
| P19 | Create a passenger with minimum passport number length | Positive | No bugs detected |
| P20 | Create a passenger with minimum + 1 passport number length | Positive | No bugs detected |
| P21 | Create a passenger with maximum – 1 passport number length | Positive | No bugs detected |
| P22 | Create a passenger with maximum passport number length | Positive | No bugs detected |
| P23 | Create a passenger with maximum + 1 passport number length | Negative | No bugs detected |
| P24 | Create a passenger with lower case passport number characters | Negative | No bugs detected |
| P25 | Create a passenger with invalid symbols in their passport number | Negative | No bugs detected |
| P26 | Create a passenger with a passport number already used by the customer | Negative | No bugs detected |
| P27 | Create a passenger with a correct birth date | Positive | No bugs detected |
| P28 | Create a passenger with minimum – 1 second birth date | Negative | No bugs detected |
| P29 | Create a passenger with minimum birth date | Positive | No bugs detected |
| P30 | Create a passenger with minimum +1 second birth date | Positive | No bugs detected |
| P31 | Create a passenger with maximum – 1 second birth date | Positive | No bugs detected |
| P32 | Create a passenger with maximum birth date | Positive | No bugs detected |
| P33 | Create a passenger with maximum + 1 second birth date | Negative | No bugs detected |
| P34 | Create a passenger with a wrong format birth date | Negative | No bugs detected |
| P35 | Create a passenger with empty special needs | Positive | No bugs detected |
| P36 | Create a passenger with base special needs length | Positive | No bugs detected |
| P37 | Create a passenger with minimum special needs length | Positive | No bugs detected |
| P38 | Create a passenger with minimum + 1 special needs length | Positive | No bugs detected |
| P39 | Create a passenger with maximum – 1 special needs length | Positive | No bugs detected |
| P40 | Create a passenger with maximum special needs length | Positive | No bugs detected |
| P41 | Create a passenger with maximum + 1 special needs length | Negative | No bugs detected |
| P42 | Create a passenger with non-latin special needs characters | Positive | No bugs detected |
| P43 | Create a passenger with an injection special needs | Positive | No bugs detected |
| P44 | Try to create a passenger with an anonymous realm | Hacking | No bugs detected |
| **UPDATE** | | | |
| P45 | Update a passenger with null data | Negative | No bugs detected |
| P46 | Update a passenger with base full name length | Positive | No bugs detected |
| P47 | Update a passenger with minimum full name length | Positive | No bugs detected |
| P48 | Update a passenger with minimum + 1 full name length | Positive | No bugs detected |
| P49 | Update a passenger with maximum – 1 full name length | Positive | No bugs detected |
| P50 | Update a passenger with maximum full name length | Positive | No bugs detected |
| P51 | Update a passenger with maximum + 1 full name length | Negative | No bugs detected |
| P52 | Update a passenger with non-latin full name characters | Positive | No bugs detected |
| P53 | Update a passenger with an injection full name | Positive | No bugs detected |
| P54 | Update a passenger with a valid email | Positive | No bugs detected |
| P55 | Update a passenger with an invalid email | Negative | No bugs detected |
| P56 | Update a passenger with a correct passport number format | Positive | No bugs detected |
| P57 | Update a passenger with minimum passport number length | Positive | No bugs detected |
| P58 | Update a passenger with minimum + 1 passport number length | Positive | No bugs detected |
| P59 | Update a passenger with maximum – 1 passport number length | Positive | No bugs detected |
| P60 | Update a passenger with maximum passport number length | Positive | No bugs detected |
| P61 | Update a passenger with maximum + 1 passport number length | Negative | No bugs detected |
| P62 | Update a passenger with lower case passport number characters | Negative | No bugs detected |
| P63 | Update a passenger with invalid symbols in their passport number | Negative | No bugs detected |
| P64 | Update a passenger with a passport number already used by the customer | Negative | No bugs detected |
| P65 | Update a passenger with a correct birth date | Positive | No bugs detected |
| P66 | Update a passenger with minimum – 1 second birth date | Negative | No bugs detected |
| P67 | Update a passenger with minimum birth date | Positive | No bugs detected |
| P68 | Update a passenger with minimum +1 second birth date | Positive | No bugs detected |
| P69 | Update a passenger with maximum – 1 second birth date | Positive | No bugs detected |
| P70 | Update a passenger with maximum birth date | Positive | No bugs detected |
| P71 | Update a passenger with maximum + 1 second birth date | Negative | No bugs detected |
| P72 | Update a passenger with a wrong format birth date | Negative | No bugs detected |
| P73 | Update a passenger with empty special needs | Positive | No bugs detected |
| P74 | Update a passenger with base special needs length | Positive | No bugs detected |
| P75 | Update a passenger with minimum special needs length | Positive | No bugs detected |
| P76 | Update a passenger with minimum + 1 special needs length | Positive | No bugs detected |
| P77 | Update a passenger with maximum – 1 special needs length | Positive | No bugs detected |
| P78 | Update a passenger with maximum special needs length | Positive | No bugs detected |
| P79 | Update a passenger with maximum + 1 special needs length | Negative | No bugs detected |
| P80 | Update a passenger with non-latin special needs characters | Positive | No bugs detected |
| P81 | Update a passenger with an injection special needs | Positive | No bugs detected |
| P82 | Try to update a passenger with an anonymous realm | Hacking | No bugs detected |
| P83 | Try to update a passenger with right realm but wrong user | Hacking | No bugs detected |
| P84 | Try to update a published passenger with right realm | Hacking | No bugs detected |
| P85 | Try to update a published passenger with wrong realm | Hacking | No bugs detected |
| P86 | Try to update a passenger that doesn’t exist | Hacking | No bugs detected |
| P87 | Try to update a passenger legally but using the URL | Hacking | No bugs detected |
| **PUBLISH** | | | |
| P88 | Publish a passenger with null data | Negative | No bugs detected |
| P89 | Publish a passenger with base full name length | Positive | No bugs detected |
| P90 | Publish a passenger with minimum full name length | Positive | No bugs detected |
| P91 | Publish a passenger with minimum + 1 full name length | Positive | No bugs detected |
| P92 | Publish a passenger with maximum – 1 full name length | Positive | No bugs detected |
| P93 | Publish a passenger with maximum full name length | Positive | No bugs detected |
| P94 | Publish a passenger with maximum + 1 full name length | Negative | No bugs detected |
| P95 | Publish a passenger with non-latin full name characters | Positive | No bugs detected |
| P96 | Publish a passenger with an injection full name | Positive | No bugs detected |
| P97 | Publish a passenger with a valid email | Positive | No bugs detected |
| P98 | Publish a passenger with an invalid email | Negative | No bugs detected |
| P99 | Publish a passenger with a correct passport number format | Positive | No bugs detected |
| P100 | Publish a passenger with minimum passport number length | Positive | No bugs detected |
| P101 | Publish a passenger with minimum + 1 passport number length | Positive | No bugs detected |
| P102 | Publish a passenger with maximum – 1 passport number length | Positive | No bugs detected |
| P103 | Publish a passenger with maximum passport number length | Positive | No bugs detected |
| P104 | Publish a passenger with maximum + 1 passport number length | Negative | No bugs detected |
| P105 | Publish a passenger with lower case passport number characters | Negative | No bugs detected |
| P106 | Publish a passenger with invalid symbols in their passport number | Negative | No bugs detected |
| P107 | Publish a passenger with a passport number already used by the customer | Negative | No bugs detected |
| P108 | Publish a passenger with a correct birth date | Positive | No bugs detected |
| P109 | Publish a passenger with minimum – 1 second birth date | Negative | No bugs detected |
| P110 | Publish a passenger with minimum birth date | Positive | No bugs detected |
| P111 | Publish a passenger with minimum +1 second birth date | Positive | No bugs detected |
| P112 | Publish a passenger with maximum – 1 second birth date | Positive | No bugs detected |
| P113 | Publish a passenger with maximum birth date | Positive | No bugs detected |
| P114 | Publish a passenger with maximum + 1 second birth date | Negative | No bugs detected |
| P115 | Publish a passenger with a wrong format birth date | Negative | No bugs detected |
| P116 | Publish a passenger with empty special needs | Positive | No bugs detected |
| P117 | Publish a passenger with base special needs length | Positive | No bugs detected |
| P118 | Publish a passenger with minimum special needs length | Positive | No bugs detected |
| P119 | Publish a passenger with minimum + 1 special needs length | Positive | No bugs detected |
| P120 | Publish a passenger with maximum – 1 special needs length | Positive | No bugs detected |
| P121 | Publish a passenger with maximum special needs length | Positive | No bugs detected |
| P122 | Publish a passenger with maximum + 1 special needs length | Negative | No bugs detected |
| P123 | Publish a passenger with non-latin special needs characters | Positive | No bugs detected |
| P124 | Publish a passenger with an injection special needs | Positive | No bugs detected |
| P125 | Try to publish a passenger with an anonymous realm | Hacking | No bugs detected |
| P126 | Try to publish a passenger with right realm but wrong user | Hacking | No bugs detected |
| P127 | Try to publish a published passenger with right realm | Hacking | No bugs detected |
| P128 | Try to publish a published passenger with wrong realm | Hacking | No bugs detected |
| P129 | Try to publish a passenger that doesn’t exist | Hacking | No bugs detected |
| P130 | Try to publish a passenger legally but using the URL | Hacking | No bugs detected |
| **DELETE** | | | |
| P131 | Delete a passenger which isn’t assigned to any booking | Positive | No bugs detected |
| P132 | Delete a passenger which is assigned to more than one booking | Positive | No bugs detected |
| P133 | Try to delete a passenger with an anonymous realm | Hacking | No bugs detected |
| P134 | Try to delete a passenger with right realm but wrong user | Hacking | No bugs detected |
| P135 | Try to delete a published passenger with right realm | Hacking | No bugs detected |
| P136 | Try to delete a published passenger with wrong realm | Hacking | No bugs detected |
| P137 | Try to delete a passenger that doesn’t exist | Hacking | No bugs detected |
| P138 | Try to delete a passenger legally but using the URL | Hacking | No bugs detected |

### Booking functional testing

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test case** | **Type** | **Bugs detected** |
| **LIST** | | | |
| B1 | List bookings | Positive | No bugs detected |
| B2 | Try to list bookings with an anonymous realm | Hacking | No bugs detected |
| **SHOW** | | | |
| B3 | Show a booking information | Positive | No bugs detected |
| B4 | Try to show a booking with an anonymous realm | Hacking | No bugs detected |
| B5 | Try to show a booking with right realm but wrong user | Hacking | No bugs detected |
| B6 | Try to show a booking that doesn’t exist | Hacking | No bugs detected |
| **CREATE** | | | |
| B7 | Create a booking with null data | Negative | No bugs detected |
| B8 | Create a booking with a correct locator code format | Positive | No bugs detected |
| B9 | Create a booking with minimum locator code length | Positive | No bugs detected |
| B10 | Create a booking with minimum + 1 locator code length | Positive | No bugs detected |
| B11 | Create a booking with maximum – 1 locator code length | Positive | No bugs detected |
| B12 | Create a booking with maximum locator code length | Positive | No bugs detected |
| B13 | Create a booking with maximum + 1 locator code length | Negative | No bugs detected |
| B14 | Create a booking with lower case locator code characters | Negative | No bugs detected |
| B15 | Create a booking with invalid symbols in its locator code | Negative | No bugs detected |
| B16 | Create a booking with a locator code that already exists | Negative | No bugs detected |
| B17 | Create a booking with “ECONOMY” travel class | Positive | No bugs detected |
| B18 | Create a booking with “BUSINESS” travel class | Positive | No bugs detected |
| B19 | Create a booking with a 4-digits credit card last nibble | Positive | No bugs detected |
| B20 | Create a booking with a 3-digits credit card last nibble | Negative | No bugs detected |
| B21 | Create a booking with a 5-digits credit card last nibble | Negative | No bugs detected |
| B22 | Create a booking with characters on its credit card last nibble | Negative | No bugs detected |
| B23 | Create a booking with a valid flight | Positive | No bugs detected |
| B24 | Try to create a booking with an anonymous realm | Hacking | No bugs detected |
| B25 | Try to create a booking with an unpublished flight | Hacking | No bugs detected |
| B26 | Try to create a booking with a flight that doesn´t exist | Hacking | No bugs detected |
| B27 | Try to create a booking with a travel class that doesn’t exist | Hacking | No bugs detected |
| **UPDATE** | | | |
| B28 | Update a booking with null data | Negative | No bugs detected |
| B29 | Update a booking with a correct locator code format | Positive | No bugs detected |
| B30 | Update a booking with minimum locator code length | Positive | No bugs detected |
| B31 | Update a booking with minimum + 1 locator code length | Positive | No bugs detected |
| B32 | Update a booking with maximum – 1 locator code length | Positive | No bugs detected |
| B33 | Update a booking with maximum locator code length | Positive | No bugs detected |
| B34 | Update a booking with maximum + 1 locator code length | Negative | No bugs detected |
| B35 | Update a booking with lower case locator code characters | Negative | No bugs detected |
| B36 | Update a booking with invalid symbols in its locator code | Negative | No bugs detected |
| B37 | Update a booking with a locator code that already exists | Negative | No bugs detected |
| B38 | Update a booking with “ECONOMY” travel class | Positive | No bugs detected |
| B39 | Update a booking with “BUSINESS” travel class | Positive | No bugs detected |
| B40 | Update a booking with a 4-digits credit card last nibble | Positive | No bugs detected |
| B41 | Update a booking with a 3-digits credit card last nibble | Negative | No bugs detected |
| B42 | Update a booking with a 5-digits credit card last nibble | Negative | No bugs detected |
| B43 | Update a booking with characters on its credit card last nibble | Negative | No bugs detected |
| B44 | Update a booking with a valid flight | Positive | No bugs detected |
| B45 | Try to update a booking with an anonymous realm | Hacking | No bugs detected |
| B46 | Try to update a booking with right realm but wrong user | Hacking | No bugs detected |
| B47 | Try to update a published booking with right realm | Hacking | No bugs detected |
| B48 | Try to update a published booking with wrong realm | Hacking | No bugs detected |
| B49 | Try to update a booking that doesn’t exist | Hacking | No bugs detected |
| B50 | Try to update a booking modifying its purchase moment read-only input | Hacking | No bugs detected |
| B51 | Try to update a booking modifying its price read-only input | Hacking | No bugs detected |
| B52 | Try to update a booking with an unpublished flight | Hacking | No bugs detected |
| B53 | Try to update a booking with a flight that doesn´t exist | Hacking | No bugs detected |
| B54 | Try to update a booking with a travel class that doesn’t exist | Hacking | No bugs detected |
| B55 | Try to update a booking legally but using the URL | Hacking | No bugs detected |
| **PUBLISH** | | | |
| B56 | Publish a booking with null data | Negative | No bugs detected |
| B57 | Publish a booking with a correct locator code format | Positive | No bugs detected |
| B58 | Publish a booking with minimum locator code length | Positive | No bugs detected |
| B59 | Publish a booking with minimum + 1 locator code length | Positive | No bugs detected |
| B60 | Publish a booking with maximum – 1 locator code length | Positive | No bugs detected |
| B61 | Publish a booking with maximum locator code length | Positive | No bugs detected |
| B62 | Publish a booking with maximum + 1 locator code length | Negative | No bugs detected |
| B63 | Publish a booking with lower case locator code characters | Negative | No bugs detected |
| B64 | Publish a booking with invalid symbols in its locator code | Negative | No bugs detected |
| B65 | Publish a booking with a locator code that already exists | Negative | No bugs detected |
| B66 | Publish a booking with “ECONOMY” travel class | Positive | No bugs detected |
| B67 | Publish a booking with “BUSINESS” travel class | Positive | No bugs detected |
| B68 | Publish a booking with a 4-digits credit card last nibble | Positive | No bugs detected |
| B69 | Publish a booking with a 3-digits credit card last nibble | Negative | No bugs detected |
| B70 | Publish a booking with a 5-digits credit card last nibble | Negative | No bugs detected |
| B71 | Publish a booking with characters on its credit card last nibble | Negative | No bugs detected |
| B72 | Publish a booking with a valid flight | Positive | No bugs detected |
| B73 | Publish a booking without a credit card last nibble stored | Negative | No bugs detected |
| B74 | Publish a booking with no passengers assigned | Negative | No bugs detected |
| B75 | Publish a booking with at least one unpublished passenger assigned | Negative | No bugs detected |
| B76 | Try to publish a booking with an anonymous realm | Hacking | No bugs detected |
| B77 | Try to publish a booking with right realm but wrong user | Hacking | No bugs detected |
| B78 | Try to publish a published booking with right realm | Hacking | No bugs detected |
| B79 | Try to publish a published booking with wrong realm | Hacking | No bugs detected |
| B80 | Try to publish a booking that doesn’t exist | Hacking | No bugs detected |
| B81 | Try to publish a booking modifying its purchase moment read-only input | Hacking | No bugs detected |
| B82 | Try to publish a booking modifying its price read-only input | Hacking | No bugs detected |
| B83 | Try to publish a booking with an unpublished flight | Hacking | No bugs detected |
| B84 | Try to publish a booking with a flight that doesn´t exist | Hacking | No bugs detected |
| B85 | Try to publish a booking with a travel class that doesn’t exist | Hacking | No bugs detected |
| B86 | Try to publish a booking legally but using the URL | Hacking | No bugs detected |
| **DELETE** | | | |
| B87 | Delete a booking which isn’t assigned to any passenger | Positive | No bugs detected |
| B88 | Delete a booking which is assigned to more than one passenger | Positive | No bugs detected |
| B89 | Try to delete a booking with an anonymous realm | Hacking | No bugs detected |
| B90 | Try to delete a booking with right realm but wrong user | Hacking | No bugs detected |
| B91 | Try to delete a published booking with right realm | Hacking | No bugs detected |
| B92 | Try to delete a published booking with wrong realm | Hacking | No bugs detected |
| B93 | Try to delete a booking that doesn’t exist | Hacking | No bugs detected |
| B94 | Try to delete a booking legally but using the URL | Hacking | No bugs detected |

### Booking assignations (intermediate entity) functional testing

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test case** | **Type** | **Bugs detected** |
| **LIST** | | | |
| A1 | List booking assignations from a booking | Positive | No bugs detected |
| A2 | Try to list bookings assignations from a booking with an anonymous realm | Hacking | No bugs detected |
| A3 | Try to list bookings assignations from a booking with right realm but wrong user | Hacking | No bugs detected |
| A4 | Try to list booking assignations from a booking that doesn’t exist | Hacking | No bugs detected |
| **SHOW** | | | |
| A5 | Show a booking assignation information | Positive | No bugs detected |
| A6 | Try to show a booking assignation with an anonymous realm | Hacking | No bugs detected |
| A7 | Try to show a booking assignation with right realm but wrong user | Hacking | No bugs detected |
| A8 | Try to show a booking assignation that doesn’t exist | Hacking | No bugs detected |
| **CREATE** | | | |
| A9 | Add a booking assignation | Positive | No bugs detected |
| A10 | Add a booking assignation for an already assigned passenger | Negative | No bugs detected |
| A11 | Add a booking assignation for a null passenger | Negative | No bugs detected |
| A12 | Try to create a booking assignation for a booking with an anonymous realm | Hacking | No bugs detected |
| A13 | Try to create a booking assignation for a booking with right realm but wrong user | Hacking | No bugs detected |
| A14 | Try to create a booking assignation for a booking with a passenger from another user | Hacking | No bugs detected |
| A15 | Try to create a booking assignation for a booking with a passenger that doesn´t exist | Hacking | No bugs detected |
| A16 | Try to create a booking assignation for a published booking | Hacking | No bugs detected |
| A17 | Try to create a booking assignation for a booking that doesn’t exist | Hacking | No bugs detected |
| **DELETE** | | | |
| A18 | Delete a booking assignation | Positive | No bugs detected |
| A19 | Try to delete a booking assignation with an anonymous realm | Hacking | No bugs detected |
| A20 | Try to delete a booking assignation with right realm but wrong user | Hacking | No bugs detected |
| A21 | Try to delete a booking assignation from a published booking | Hacking | No bugs detected |
| A22 | Try to delete a booking assignation that doesn’t exist | Hacking | No bugs detected |
| A23 | Try to delete a booking assignation legally but using the URL | Hacking | No bugs detected |

### Total coverage

All these test cases resulted in an almost 100% coverage for the customers functionalities requested, shown below:

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

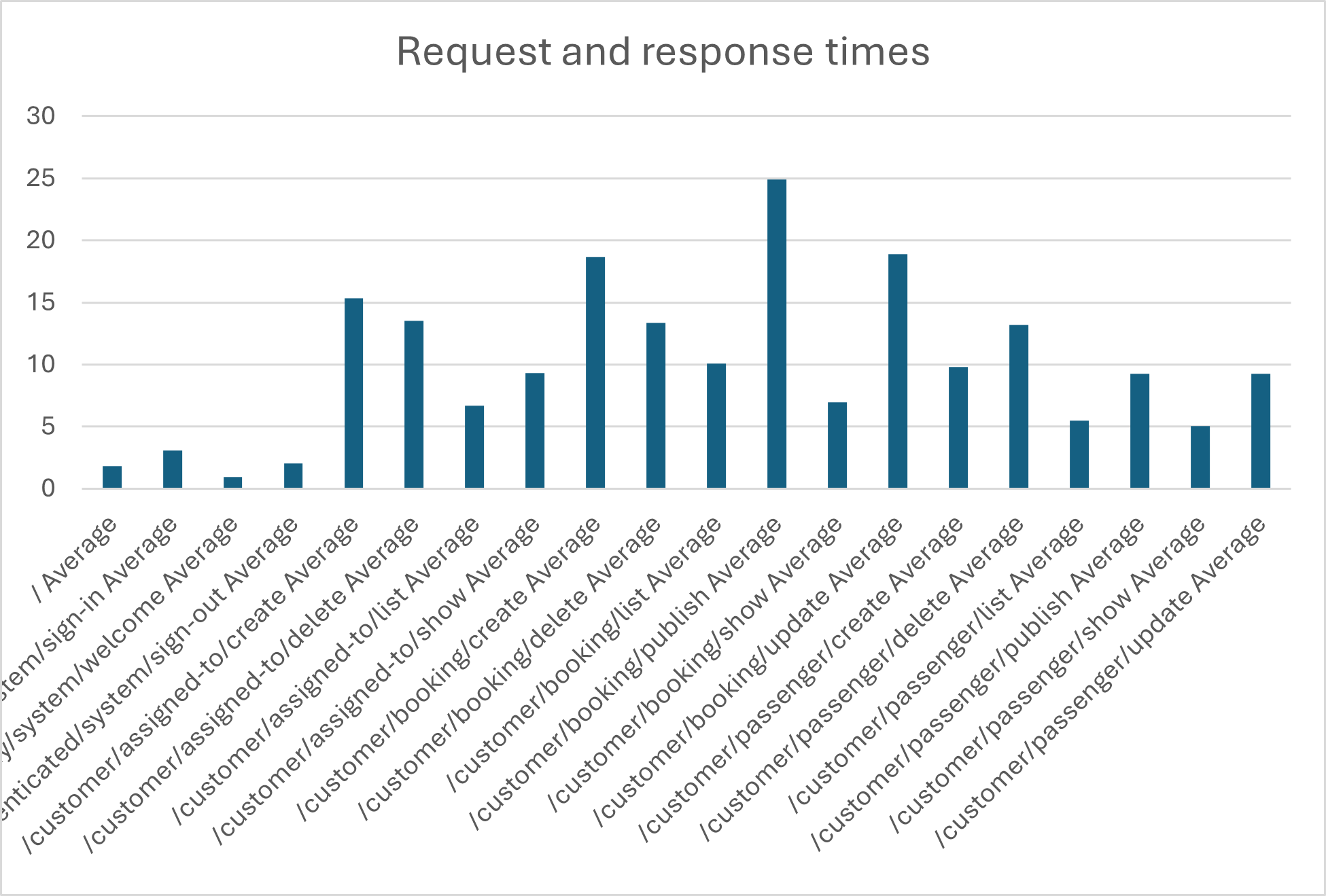
AI-generated content may be incorrect.

The Delete services are an exception, none of them is 100% covered because they have each an unbind method that is never really used; because of the requirements, deletions won’t have any errors that make the services unbind the data so it can be shown in the edition form, so it will never be possible to visit these methods, and they will never be covered by the test cases.

## Performance testing

For the performance testing chapter, data has been extracted and analysed from two different computers.

### Computer A results

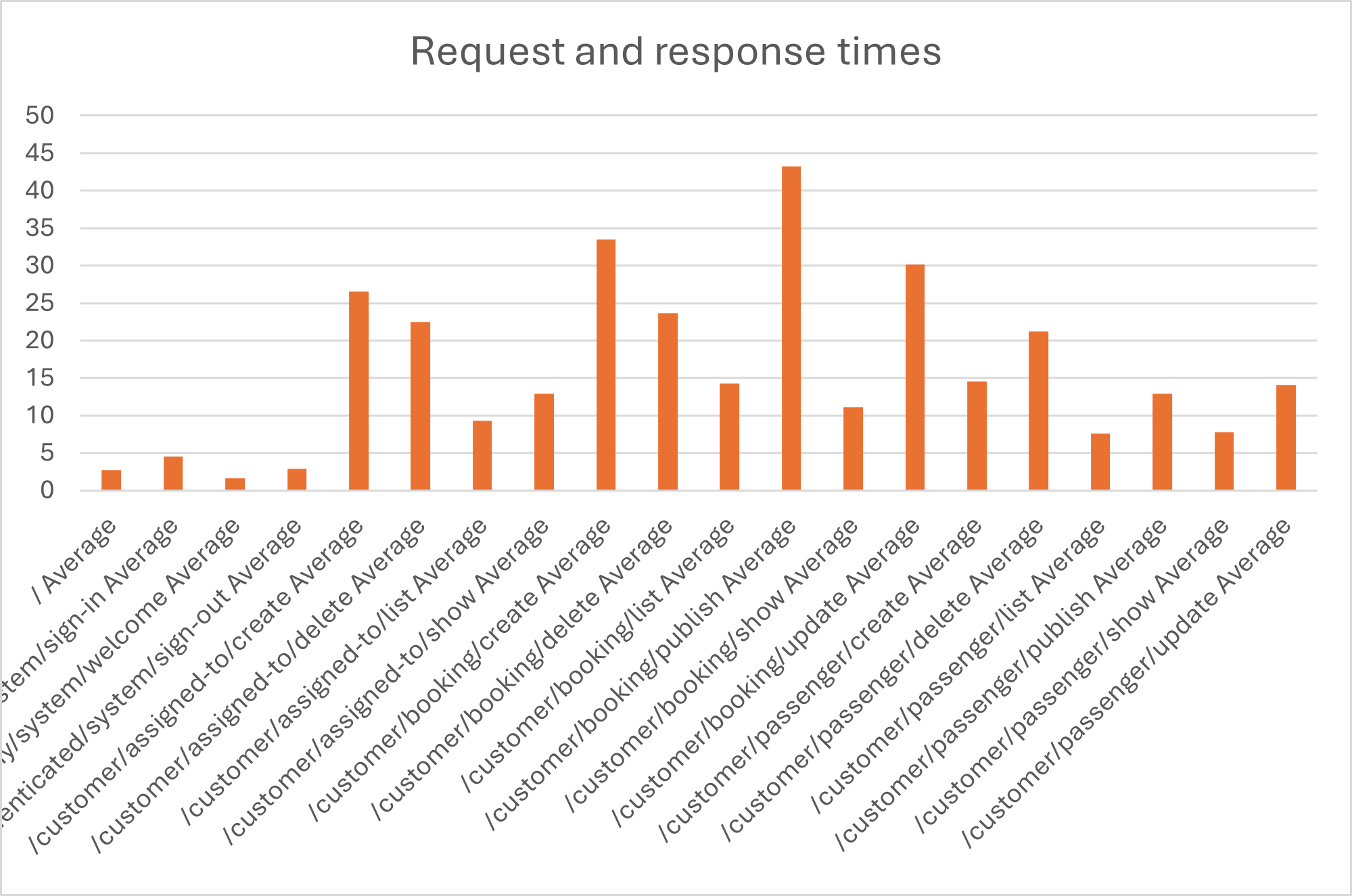


As we can see, booking create, update and publish functionalities are the ones which take the most time to be completed, this is because bookings are the most complex entities of all three, having a wide range of previous validations in comparison of the other entities, and also having longer authorise() and validate() methods than Passengers and AssignedTos

The confidence level in computer A is computed from the following data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Computer A statistics* | |  |  |  |  |
|  |  |  | Interval (ms) | 6,1615889 | 7,20439632 |
| Mean | 6,682992612 |  | Interval (s) | 0,00616159 | 0,007204396 |
| Standard Error | 0,265657094 |  |  |  |  |
| Median | 4,437 |  |  |  |  |
| Mode | 1,0163 |  |  |  |  |
| Standard Deviation | 7,84026054 |  |  |  |  |
| Sample Variance | 61,46968533 |  |  |  |  |
| Kurtosis | 7,627434982 |  |  |  |  |
| Skewness | 2,347747053 |  |  |  |  |
| Range | 59,2236 |  |  |  |  |
| Minimum | 0,4909 |  |  |  |  |
| Maximum | 59,7145 |  |  |  |  |
| Sum | 5820,886565 |  |  |  |  |
| Count | 871 |  |  |  |  |
| Confidence Level(95,0%) | 0,521403708 |  |  |  |  |

### Computer B results



Computer B has an overall similarity to computer A results, being Booking services that write in the database the ones which take longer again. One notorious difference that these data show is the time that all requests take in general, which is longer than the one measured in computer A.

The confidence level in computer B is computed from the following data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Computer B statistics* | |  |  |  |  |
|  |  |  | Interval (ms) | 9,605738 | 11,4454 |
| Mean | 10,52557093 |  | Interval (s) | 0,009606 | 0,011445 |
| Standard Error | 0,468658415 |  |  |  |  |
| Median | 6,085401 |  |  |  |  |
| Mode | 1,4468 |  |  |  |  |
| Standard Deviation | 13,83137949 |  |  |  |  |
| Sample Variance | 191,3070586 |  |  |  |  |
| Kurtosis | 11,3853822 |  |  |  |  |
| Skewness | 2,881028125 |  |  |  |  |
| Range | 108,5455 |  |  |  |  |
| Minimum | 0,739999 |  |  |  |  |
| Maximum | 109,285499 |  |  |  |  |
| Sum | 9167,772282 |  |  |  |  |
| Count | 871 |  |  |  |  |
| Confidence Level(95,0%) | 0,919833278 |  |  |  |  |

### Results comparison

If we look closely at the graphs, we can see, as it was mentioned, that computer B has taken more time in resolving its requests, but if we analyse the intervals obtained, this assumption makes itself clear; just seeing that computer A’s top limit is less than computer B’s bottom limit is enough to confirm that every request made from computer A will be resolved faster than its homologues made from computer B.

But this information isn’t conclusive. So, to get a satisfactory conclusion, we will find the p-value executing a z-test, which ‘Before’ data was taken from computer B, and ‘After’ data from computer A.

|  |  |  |
| --- | --- | --- |
|  | *Before* | *After* |
| Mean | 10,52954 | 6,685874 |
| Known Variance | 191,3071 | 61,46969 |
| Observations | 873 | 873 |
| Hypothesized Mean Difference | 0 |  |
| z | 7,143055 |  |
| P(Z<=z) one-tail | 4,56E-13 |  |
| z Critical one-tail | 1,644854 |  |
| P(Z<=z) two-tail | 9,13E-13 |  |
| z Critical two-tail | 1,959964 |  |

The p-value obtained was 9,13E-13, which is incredibly low and below the alpha value (0.05), so thanks to this, we can ensure that the assumptions that we have been making are true. We will finally compare both computers mean times to confirm this hypothesis and declare one of them as the most powerful.

Computer A mean time: 6,685874 ms

Computer B mean time: 10,52954 ms

This difference, not only at their means, but also at their whole charts and confidence intervals as well; all of these computed before, makes the hypothesis clear and allow us to confirm that, certainly, computer A is more powerful than computer B.

# Conclusions

The testing process carried out in this report demonstrates a thorough and structured approach to validating both the functional correctness and performance efficiency of the software. Through an extensive suite of functional test cases covering positive, negative, and hacking scenarios, the system was shown to behave as expected across all tested features without revealing any bugs. This reflects a robust implementation and a careful attention to requirements.

In performance testing, empirical data was gathered from two computing environments. Statistical analysis, including confidence intervals and hypothesis testing, revealed a significant performance difference between the two systems. Specifically, Computer A consistently processed requests faster than Computer B, a conclusion supported by both interval comparison and p-value analysis.

Overall, the combination of exhaustive functional verification and rigorous performance evaluation confirms the system’s readiness and stability. The testing methods applied, such as automated execution and statistical validation, serve as a best-practice model for future testing efforts within similar development environments.

# Bibliography

Intentionally blank.