

Requirements – Student #2

Please, fill in the following form, make sure that you have ticked the requirements that you consider fulfilled, save this document, **and attach it in its original format (.docx)** to every deliverable. Regarding your ID, please keep only four random digits and mask the others using an asterisk. **Please**, note that this document must be edited with the desktop version of Word since the web version does not properly support forms. Attaching this document entails that you are the authors of the work delivered, you have not cheated in any way, and you have read and understood the information delivered regarding the subject, with a special emphasis on the methodological guidelines and how your work is going to be graded. Make sure that your project works well with the latest version of the development framework.

Group: C1.022

Repository: <https://github.com/FranciscoFernandezN/Acme-ANS>

Student #2

ID Number: 7**4*4*7*

UVUS: BPP4634

Name: Gutiérrez Arazo, Beatriz

Roles: Analista, Desarrolladora, Tester

Date: Sevilla Abril 26, 2025

MANDATORY Deliverable D01: introduction

Information requirements

Intentionally blank.

Functional requirements

- 1) Modify the anonymous menu so that it shows an option that takes the browser to the home page of your favourite web site. The title must read as follows: “<id-number>: <surname>, <name>”, where “<id-number>” denotes your DNI, NIE, or passport number, “<surname>” denotes your surname/s, and “<name>” denotes your name/s.

X

Non-functional requirements

Intentionally blank.

Testing requirements

Intentionally blank.

Managerial requirements

- 2) Provide a link to your planning dashboard in GitHub to review the tasks, their current status, and your schedule.

X

MANDATORY

Deliverable D02: data models

Information requirements

- 3) **Customers** are the people who purchase flights. The system must store the following data about them: an **identifier** (unique, pattern "`^[A-Z]{2-3}\d{6}$`", where the first two or three letters correspond to their initials), a **phone number** (pattern "`^\+?\d{6,15}$`"), a **physical address** (up to 255 characters), plus a **city** and a **country** (both up to 50 characters). Optionally, customers may have some **earned points** (up to 500k points).

X

- 4) A **booking** is a reservation made by a **customer** to purchase a **flight**, guaranteeing some seats on a specific itinerary and associating some **passengers'** details with the trip. The system must manage the following information for each **booking**: a **locator code** (unique, pattern "`^[A-Z0-9]{6,8}$`"), a **purchase moment** (in the past), a **travel class** ("ECONOMY", "BUSINESS"), and a **price**. Optionally, the system should record the **last nibble** of the credit card used for payment.

X

- 5) A **passenger** is an individual who takes a flight and he or she must be registered in the corresponding booking. The system must store the following data about passengers: a **full name** (shorter than 256 characters), an **email**, a **passport number** (pattern "`^[A-Z0-9]{6,9}$`"), a **date of birth**, and, optionally, his or her **special needs** (shorter than 51 characters).

X

Functional requirements

Intentionally blank.

Non-functional requirements

Intentionally blank.

Testing requirements

- 6) Produce assorted sample data to test your application informally. The data must include two **customer** accounts with credentials "**customer1/customer1**" and "**customer2/customer2**". Create an additional customer account with credentials "**customer3/customer3**" that represents a customer with only profile data.

X

Managerial requirements

- 7) Provide a link to your planning dashboard in GitHub to review the tasks, their current status, and your schedule.

X

MANDATORY

Deliverable D03: implementing features

Information requirements

Intentionally blank.

Functional requirements

8) Operations by **customers** on **bookings**:

- List their bookings.
- Show the details of their bookings and the associated passengers, if any.
- Create or update their bookings. Bookings can be updated as long as they have not been published. A booking can be published only when the last credit card nibble has been stored.

X

9) Operations by **customers** on **passengers**:

- List the passengers in their bookings.
- Show the details of their passengers.
- Create a passenger and record the information related to that passenger.
- Update a passenger as long as it has not been published.

X

Non-functional requirements

Intentionally blank.

Testing requirements

Intentionally blank.

Managerial requirements

- 10) Provide a link to your planning dashboard in GitHub to review the tasks, their current status, and your schedule.

X

MANDATORY

Deliverable D04: formal testing

Information requirements

- 11) Create appropriate indices for your entities, if required.

X

Functional requirements

Intentionally blank.

Non-functional requirements

Intentionally blank.

Testing requirements

- 12) Produce a test suite for Requirements #8 and #9.

X

Managerial requirements

- 13) Provide a link to your planning dashboard in GitHub to review the tasks, their current status, and your schedule.

X

- 14) Produce a testing report.

X

SUPPLEMENTARY I

Deliverable D01: introduction

Information requirements

Intentionally blank.

Functional requirements

Intentionally blank.

Non-functional requirements

Intentionally blank.

Testing requirements

Intentionally blank.

Managerial requirements

Intentionally blank.

SUPPLEMENTARY I

Deliverable D02: data models

Information requirements

- 15) The system must handle **customers dashboards** with the following **indicators**:
- The last five destinations.
 - The money spent in bookings during the last year.
 - Their number of bookings grouped by travel class.
 - Count, average, minimum, maximum, and standard deviation of the cost of their bookings in the last five years.
 - Count, average, minimum, maximum, and standard deviation of the number of passengers in their bookings.

X

Functional requirements

Intentionally blank.

Non-functional requirements

Intentionally blank.

Testing requirements

Intentionally blank.

Managerial requirements

- 16) Produce a UML domain model regarding the information requirements.

X

SUPPLEMENTARY I

Deliverable D03: implementing features

Information requirements

Intentionally blank.

Functional requirements

17) Operations by **anonymous principals** on user **accounts**:

- Sign up to the system and become a customer.

X

18) Operations by **customers** on user **accounts**:

- Update their profiles.

X

19) Operations by **administrators** on **bookings**:

- List the bookings in the system that are published.
- Show the details of the bookings (including the passengers).

X

20) Operations by **customer** on **dashboards**:

- Show their customer dashboards.

X

Non-functional requirements

Intentionally blank.

Testing requirements

Intentionally blank.

Managerial requirements

21) Provide a link to a video in which you informally test requirement #8 and #9. Videos should not exceed 10 minutes in length and must be stored at the USE's facilities.

X

SUPPLEMENTARY I

Deliverable D04: formal testing

Information requirements

Intentionally blank.

Functional requirements

Intentionally blank.

Non-functional requirements

Intentionally blank.

Testing requirements

Intentionally blank.

Managerial requirements

22) Perform five mutations in your code and report on the results.

X

Managerial requirements

23) Produce a lint report.

X

SUPPLEMENTARY II

Deliverable D01: introduction

Information requirements

Intentionally blank.

Functional requirements

Intentionally blank.

Non-functional requirements

Intentionally blank.

Testing requirements

Intentionally blank.

Managerial requirements

24) Produce an analysis report.

X

25) Produce a planning and progress report.

X

SUPPLEMENTARY II

Deliverable D02: data models

Information requirements

- 26) The system must include a board to recommend something in the city and/or country of a given airport. **Recommendations** can be about experiences, activities, restaurants, accommodation or any other thing that a person may find interesting at the destination. A web service must be used to populate this entity with information about recommendations. Thus, the exact data to store depends on the chosen service, and it is the students' responsibility to define them accordingly. It is also the students' responsibility to find the appropriate service; no implicit or explicit liabilities shall be covered by the University of Seville or their individual affiliates if the students contract pay-per-use services! The students are strongly advised to ensure that the service they choose is free of charge.

X

Functional requirements

Intentionally blank.

Non-functional requirements

Intentionally blank.

Testing requirements

Intentionally blank.

Managerial requirements

- 27) Produce an analysis report.

X

- 28) Produce a planning and progress report.

X

SUPPLEMENTARY II

Deliverable D03: implementing features

Information requirements

Intentionally blank.

Functional requirements

29) Operations by **customers** on **recommendations**:

- List recommendations related to final destinations in their bookings.

X

30) Operations by **administrators** on **recommendations**:

- Populate the database with recommendations somehow related to locations (e.g., related to cities or countries).

X

Non-functional requirements

Intentionally blank.

Testing requirements

Intentionally blank.

Managerial requirements

31) Produce an analysis report.

X

32) Produce a planning and progress report.

X

SUPPLEMENTARY II

Deliverable D04: formal testing

Information requirements

Intentionally blank.

Functional requirements

Intentionally blank.

Non-functional requirements

Intentionally blank.

Testing requirements

- 33) Produce as a complete test suite as possible for Requirement #29 ensuring that the API is properly mocked.

X

Managerial requirements

- 34) Produce an analysis report.

X

- 35) Produce a planning and progress report.

X