# Requirements – Student #4

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:** C2. 031 |
| **Repository:** https://github.com/aaronma300604/DP2-C01-31 |
| **Student #2**  **ID Number:** 5443\*\*\*\*\*  **UVUS:**  JCT6889  **Name:**  Terrón Hernández, Diego  **Roles:**  operator, developer, tester |
| **Date:** Sevillle July 2, 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

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

X Mismos problemas que los indicados en la parte grupal.

--- CORRECCIÓN ---

Se han añadido las tareas faltantes y homogenizado los nombres de las tareas.

# MANDATORY Deliverable D02: data models

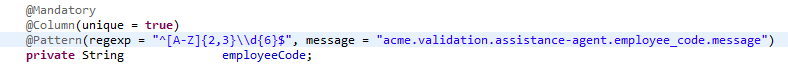
## Information requirements

1. The **assistance agents** are the people responsible for recording and managing post-flight incidents reported by passengers. The system must store the following data about them: an **employee code** (unique, pattern "^[A-Z]{2-3}\d{6}$", where the first two or three letters correspond to their initials), a list of **spoken languages** (no longer than 255 characters), the **airline** for which they work, the **moment** on which they began to work for that airline (in the past), and optionally, a **brief bio** (up to 255 characters), their **salary**, and a link to a **photo** that should be stored anywhere else.

X No ha usado ningún patrón para modelar el código del empleado, aunque lo ha implementado en el AgentValidator, se podría haber puesto en el modelo de dominio Java.

--- CORRECCIÓN ---

Se ha implementado el patrón en el modelo de dominio.



1. A **claim** is a formal request or complain made by a passenger or customer due to a problem or inconvenience experienced during a flight. They are registered by the **assistance agents**, and the data to store when registering a **claim** is the following: the **registration moment** (in the past), the **passenger email**, a **description** (up to 255 characters), a **type** (“FLIGHT ISSUES”, “LUGGAGE ISSUES”, “SECURITY INCIDENT”, “OTHER ISSUES”) and an **indicator** of whether the claim is accepted or not.

X

1. Claims need to be tracked through **tracking logs**. A tracking log records each step in the procedure followed to resolve or reject a claim, ensuring that all actions and decisions are documented. The system must store the following data about **tracking logs**: the **last update moment**, the **step** undergoing (up to 50 characters), a **resolution percentage**, and an **indicator** on whether the claim was finally accepted or not. When a claim is accepted or rejected, the system must store its **resolution** indicating the reason why was rejected or the compensation to offer (up to 255 characters).

X No ha incluido en el atributo “lastUpdate” que sea obligatorio:

A black and red text

AI-generated content may be incorrect.

¿Qué es o para que sirve el atributo “iteration”? no se ha indicado como parte de la entidad en el requisito:

A math equation with black text

AI-generated content may be incorrect.

--- CORRECCIÓN ---

Se ha añadido @Mandatory a “lastUpdate”.

Texto

El contenido generado por IA puede ser incorrecto.

El atributo “iteration” es interno y está relacionado con el requisito 9, que indica la posibilidad de crear un TrackingLog excepcional incluso existiendo ya uno al 100%. En el momento que diseñé el modelo, todavía había bastante ambigüedad en cuanto a dicho requisito. Por ejemplo, no se explica cuántas veces es posible crear un TrackingLog excepcional tras haber llegado al 100%, o qué porcentaje puede tener este nuevo TrackingLog excepcional.

El uso del atributo “iteration” fue una decisión deliberada pues hace que el código sea muy fácilmente expansible ante cualquier posible aclaración sobre este requisito. Cuando se crea el primer TrackingLog para un Claim, éste atributo toma el valor de “1”, y para el resto de TrackingLogs hasta el 100% se mantiene igual. Cuando se intenta crear un nuevo TrackingLog, se comprueba el TrackingLog con mayor porcentaje para el correspondiente Claim, y si es 100% y su “iteration” es “1”, el “iteration” del nuevo TrackingLog tomará el valor “2”. La query para ordenar los TrackingLog por porcentaje también tiene en cuenta el atributo, ordenándolos primero por “iteration” y luego por porcentaje.

Texto

El contenido generado por IA puede ser incorrecto.

Como posteriormente se aclaró que sólo puede existir un TrackingLog excepcional por Claim, ahí concluye la explicación de ejemplo. Pero en el caso de que se hubiera aclarado que podrían existir más, incluso de porcentajes menores a 100%, basta con eliminar la condición de que el TrackingLog con mayor porcentaje deba tener “iteration” a “1”. Con esto, se pueden crear TrackingLogs excepcionales infinitas veces y siempre aparecerá como el de mayor porcentaje el que mayor “iteration” tenga.

Se ha decidido mantener el atributo “iteration” incluso tras la aclaración. Aparte de cumplir con los requisitos actuales, representa una buena práctica en el diseño, pues favorece la mantenibilidad y escalabilidad del código en caso de modificaciones futuras a los requisitos, algo perfectamente factible en un entorno real.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

1. Produce assorted sample data to test your application informally. The data must include two **assistance agent** accounts with credentials “**agent1**/**agent1**” and “**agent2**/**agent2**”. Create an additional agent account with credentials “**manager3/manager3”** that accounts for a new agent with no associated data, except for his or her profile.

X

## 

## Managerial requirements

1. 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

1. Operations by **assistance agents** on **claims**:

* List their completed claims, that is, the ones that have been accepted or rejected and show their details.
* List the undergoing claims and show their details including the leg to which they are linked.
* Create, update, publish, and delete their claims. Claims must be linked to legs that occurred. Additionally, claims can be updated or deleted as long as they have not been published.

X Si no puedo asignar a una claim que estoy creando una leg que aún no ha ocurrido, ¿por qué se carga en el desplegable? En el desplegable se deberían de cargar sólo aquellas legs que estén publicadas y que ya hayan tenido lugar.

La fecha de una claim es un valor que debería de calcularse de manera automática. Es la fecha en la que se registra la claim. ¿Tiene sentido crear una claim en el año 2000? No, sólo se deberían de crear claims en el momento actual, que es la fecha que tenga configurada el sistema:

A screenshot of a computer

AI-generated content may be incorrect.

Hay problemas de internacionalización:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Un usuario puede eliminar las claims de otro usuario mediante POST hacking. Muestro a continuación los pasos que he seguido. Me logueo en la ventana de la izquierda como agent3/agent3 y en la derecha como agent1/agent1. Se puede apreciar que en las dos interfaces hay agents distintos logueados porque tienen diferentes claims en sus listados:

A screenshot of a computer

AI-generated content may be incorrect.

Hago clic en la ventana de la izquierda, en la única claim que tengo y en la de la derecha para ver algún ID de alguna claim (que están todas publicadas):

A screenshot of a computer

AI-generated content may be incorrect.

En la ventana de la izquierda pongo el id del formulario de la derecha (id = 236) en el código html que aparece (fondo negro):

A screenshot of a computer

AI-generated content may be incorrect.

Hago clic en delete, y se puede apreciar como en la ventana de la derecha hay menos elementos en el listado, lo que significa que el agent3 ha eliminado una reclamación publicada del agent1.

A screenshot of a computer

AI-generated content may be incorrect.

--- CORRECCIÓN ---

Se han actualizado los services para que los legs que se muestren en los formularios relativos a Claims sean sólo los válidos.

Se han actualizado los services para que las fechas de los Claims se asignen automáticamente.

Se ha corregido la fución “authorise” del AgentClaimsDeleteService para que compruebe que el Claim pertenece al agent que intenta realizar la acción.

Se han corregido los problemas de internacionalización.

1. Operations by **assistance agents** on **tracking logs**:

* List and show the tracking logs associated to their claims.
* Create, update, publish, and delete a tracking log. A tracking log cannot be published until its corresponding claim is published. Once published, tracking logs cannot be updated or deleted. In exceptional cases, a new tracking log may be created even after the last one has been published (the one with a 100% resolution percentage). This additional tracking log is generated when the customer expresses dissatisfaction, prompting agents to review their claims.

X

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. 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

1. Create appropriate indices for your entities, if required.

X

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

1. Produce a test suite for Requirements #8 and #9.

X

## Managerial requirements

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

X

1. 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

1. The system must handle **assistance agent** **dashboards** with the following **indicators**:

* The ratio of claims that have been resolved successfully.
* The ratio of claims that have been rejected.
* The top three months with the highest number of claims.
* The average, minimum, maximum, and standard deviation of the number of logs their claims have.
* The average, minimum, maximum, and standard deviation of the number of claims they assisted during the last month.

X

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Produce a UML domain model regarding the information requirements in your project.

X

# SUPPLEMENTARY I Deliverable D03: implementing features

## Information requirements

Intentionally blank.

## Functional requirements

1. Operations by **anonymous principals** on **user accounts**:

* Sign up to the system and become an assistance agent.

1. Operations by **assistance agents** on **user accounts**:

* Update their profiles.

1. Operations by **administrators** on **claims**:

* List the claims in the system that are published.
* Show the details of the claims that they can list (including their tracking logs).

1. Operations by **assistance agents** on **dashboards**:

* Show their dashboards.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. 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.

# SUPPLEMENTARY I Deliverable D04: formal testing

## Information requirements

Intentionally blank.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

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

## Managerial requirements

1. Produce a lint report.

# SUPPLEMENTARY II Deliverable D01: introduction

## Information requirements

Intentionally blank.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Produce an analysis report.

X

1. Produce a planning and progress report.

X

# SUPPLEMENTARY II Deliverable D02: data models

## Information requirements

1. The system is required to store **flight status** or **delays** that assistance agents can consult to help them with some claims. A web service must be used to populate this entity with information about flight statuses/delays. 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.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Produce an analysis report.

1. Produce a planning and progress report.

# SUPPLEMENTARY II Deliverable D03: implementing features

## Information requirements

Intentionally blank.

## Functional requirements

1. Operations by **assistance agents** on **flights status/delays**:

* List the flights statuses/delays available in the system.
* Show the details of the flight statuses/delays registered in the system.

1. Operations by **administrators** on **flights statuses/delays**:

* Populate the database with flights status/delay data.

## Non-functional requirements

Intentionally blank.

## Testing requirements

Intentionally blank.

## Managerial requirements

1. Produce an analysis report.

1. Produce a planning and progress report.

# SUPPLEMENTARY II Deliverable D04: formal testing

## Information requirements

Intentionally blank.

## Functional requirements

Intentionally blank.

## Non-functional requirements

Intentionally blank.

## Testing requirements

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

## Managerial requirements

1. Produce an analysis report.

1. Produce a planning and progress report.