# 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:** C2.021 |
| **Repository:** https://github.com/DP2-C1-021/Acme-ANS-C2 |
| **Student #4**  **ID Number:** \*\*\*\*0943\*  **UVUS:**  pausangom1  **Name:**  Sánchez Gómez, Paula  **Roles:**  Manager, Developer, Tester |
| **Date:** Sevilla February 18, 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.

Enlace primera convocatoria: https://github.com/orgs/DP2-C1-021/projects/1/views/1?filterQuery=assignee%3Apaulasanchezg+D01, Nuevo tablero para la segunda convocatoria: https://github.com/orgs/DP2-C1-021/projects/2/views/1?filterQuery=assignee%3Apaulasanchezg

# 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

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

## 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.

Enlace primera convocatoria: https://github.com/orgs/DP2-C1-021/projects/1/views/1?filterQuery=assignee%3Apaulasanchezg+D02, Nuevo tablero para la Segunda convocatoria: https://github.com/orgs/DP2-C1-021/projects/2/views/1?filterQuery=assignee%3Apaulasanchezg

# 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 Es posible actualizar el registration moment de un claim mediante POST hacking.

**Correción del error**

Para proteger el atributo registrationMoment de la entidad Claim frente a posibles ataques de POST hacking, he aplicado las siguientes medidas conforme a la teoría y buenas prácticas:

**1. Exclusión del campo registrationMoment en la fase de enlace de datos (binding).** En el método bind(), he evitado mapear el campo registrationMoment desde la petición hacia el objeto entidad. Esto significa que, aunque el cliente intente enviar un valor modificado para este campo a través del formulario, el sistema simplemente ignora esa información y no la asigna al objeto que se va a guardar.

**2.** **Asignación segura y controlada en la fase de ejecución (perform)** Como registrationMoment no es enlazado durante el binding, es necesario asignar manualmente su valor en el método perform(), estableciéndolo con la fecha y hora actual del servidor (usando MomentHelper.getCurrentMoment()). Así se garantiza que este campo siempre refleje el momento real y legítimo en que se crea el objeto Claim.

Esta estrategia coincide con la recomendación teórica de tratar los atributos de solo lectura como campos que no deben ser modificados a través de POST hacking, evitando su manipulación en la capa de binding y controlando su asignación internamente. Así se previene la alteración indebida de datos críticos sin necesidad de rechazar la petición en la fase de autorización, facilitando un flujo de control más limpio y seguro. **En conclusión,** la mejor práctica para proteger atributos sensibles como registrationMoment es **no incluirlos en el binding**, para evitar que puedan ser modificados por los datos enviados desde el cliente. En su lugar, se asignan directamente en la capa de negocio (perform()). Esto protege contra ataques de POST hacking, garantiza la integridad del dato y cumple con la teoría y requisitos de seguridad planteados.

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 Es posible crear varios caros excepcionales. Es possible modificar el porcentaje de resolución al publicar el caso excepcional mediante POST hacking.

**Corrección del error - Prevención de acceso y creación ilegal de tracking logs tras publicar uno con resolución 100% -> Es posible crear varios casos excepcionales**

El problema detectado era que, tras publicar un TrackingLog con un porcentaje de resolución igual a 100% para una determinada reclamación (Claim), la funcionalidad para crear nuevos TrackingLog seguía estando accesible mediante GET hacking, y además permitía crear nuevos logs, aunque no pudieran ser publicados. Esto incumple el requisito de seguridad, ya que se debería impedir totalmente el acceso y creación en este escenario, generando un error de autorización. Para corregirlo, he aplicado las siguientes medidas:

**1. Restricción en la fase de autorización (authorise) para la creación (create):** En el método authorise() del servicio AssistanceAgentTrackingLogCreateService se comprueba, al recibir una petición GET o POST para crear un tracking log, si ya existe algún TrackingLog para la reclamación con resolución publicada al 100%. Si esto es así, se deniega la autorización para la creación, evitando que el usuario acceda a la URL o cree nuevos logs. Esto implica que la ruta y acción quedan protegidas contra ataques GET hacking y POST hacking, ya que se bloquea desde el inicio.

**2.** **Cumplimiento con la teoría y buenas prácticas:** Según la teoría, las acciones ilegales (como GET o POST hacking que permiten acceder o modificar recursos que no deberían) deben ser bloqueadas en la fase de autorización. Mi corrección aplica esta recomendación, asegurando que el sistema devuelva una denegación de acceso inmediata para peticiones ilegítimas, mejorando la seguridad y la robustez.

**En resumen:** He reforzado la capa de autorización para que, tras publicar un TrackingLog con resolución 100%, no se permita ni acceder a la URL de creación ni generar nuevos logs para esa reclamación. Esto previene tanto el acceso indebido (GET hacking) como la creación ilegal (POST hacking), cumpliendo los requisitos y buenas prácticas de seguridad indicados en la teoría.

**Corrección del error - Es posible modificar el porcentaje de resolución al publicar el caso excepcional mediante POST hacking.**

El problema identificado era que, en la funcionalidad para crear un TrackingLog como caso excepcional (con resolución al 100%), el usuario podía modificar el valor del porcentaje de resolución mediante POST hacking, alterando así un dato que debería ser fijo y controlado por el sistema. Para solucionar este problema, he aplicado las siguientes acciones, siguiendo buenas prácticas de seguridad:

**1. Exclusión de los atributos sensibles del binding:** En el método bind() del servicio AssistanceAgentTrackingLogCreateExceptionalCaseService he eliminado la asignación automática del atributo resolutionPercentage (y otros sensibles como indicator o lastUpdateMoment) desde los datos enviados en la petición. En concreto, sólo dejo enlazar el campo resolution que es el único editable.  
Esto significa que, aunque un atacante intente modificar el porcentaje de resolución enviando un valor diferente en el formulario, el sistema no lo asignará al objeto entidad, ignorando ese dato malicioso.

**2. Asignación controlada en la fase load() y perform():** El valor resolutionPercentage se establece explícitamente y de forma segura en load() a 100.00 para el caso excepcional y se actualiza la fecha lastUpdateMoment en perform() con el momento actual, garantizando que estos campos reflejan siempre la información legítima y no manipulada.

De acuerdo con la teoría, los campos de sólo lectura o controlados deben evitarse en el binding para prevenir POST hacking. Esta práctica protege contra la modificación maliciosa de datos críticos sin necesidad de bloquear toda la petición en authorise(), permitiendo un flujo seguro y controlado.

**En conclusión:** He corregido el error quitando del bind() los atributos resolutionPercentage, indicator y lastUpdateMoment, de modo que no puedan ser alterados desde el cliente. Su valor se fija y controla en la lógica interna, evitando modificaciones ilegales por POST hacking y asegurando la integridad de los datos para el caso excepcional, en línea con la teoría y los requisitos de seguridad.

## 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.

Enlace primera convocatoria: https://github.com/orgs/DP2-C1-021/projects/1/views/1?filterQuery=assignee%3Apaulasanchezg+D03, Nuevo tablero para la segunda convocatoria: https://github.com/orgs/DP2-C1-021/projects/2/views/1?filterQuery=assignee%3Apaulasanchezg

# 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.

Enlace primera convocatoria: https://github.com/orgs/DP2-C1-021/projects/1/views/1?filterQuery=assignee%3Apaulasanchezg, Nuevo tablero para la Segunda convocatoria: https://github.com/orgs/DP2-C1-021/projects/2/views/1?filterQuery=assignee%3Apaulasanchezg

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.

X

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

* Update their profiles.

X

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).

X

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

* Show their dashboards.

X

## 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.

X

# 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.

X Está en el testing report

## Managerial requirements

1. 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

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.