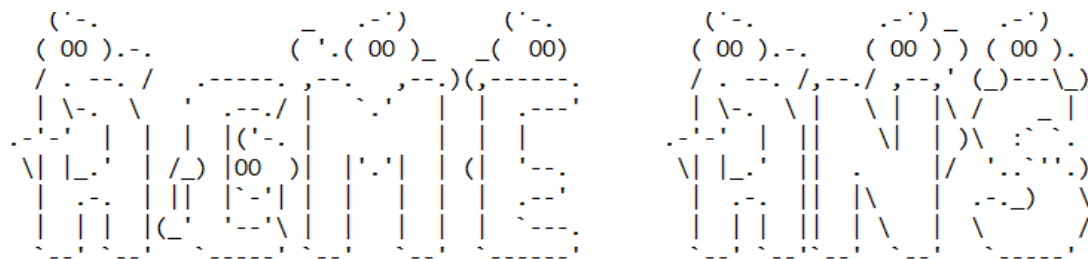


Portada

REPORTE DE ANÁLISIS

Acme-ANS-D03



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

Creado por el grupo C1.022, del G1

Participantes	
Nombres	Correos
Benito Merchán, Manuel Jesús	manbenmer1@alum.us.es
Fernández Noguero, Francisco	frafernog@alum.us.es
Gómez Navarro, Esteban	estgomnav@alum.us.es
Gutiérrez Arazo, Beatriz	beagutara@alum.us.es
Varo Vera, Juan	juavarver@alum.us.es

3 de Abril de 2025

Índice

[Portada](#) 1

[Índice](#) 2

[Resumen ejecutivo](#) 3

[Tabla de revisiones](#) 4

[Introducción](#) 5

[Contenidos](#) 6

[Decisión #1:](#) 6

[Decisión #2:](#) 7

[Conclusiones](#) 8

[Bibliografía](#) 9

Resumen ejecutivo

S.A. abreviado), la cual es una compañía ficticia especializada en ayudar a aeropuertos a organizar y coordinar sus operaciones a partir de soluciones desarrolladas en software. La logística de los vuelos (la programación de los vuelos, la organización de reservas y de tripulación, etc.) se gestionan mediante el desarrollo de un WIS. Con esto queda implícito que no solo será necesario desarrollar la aplicación en sí, sino también una documentación apropiada en la que se refleje la evolución de esta. En este caso, se trata de un reporte de análisis en el que describiremos toda la información relacionada con las decisiones de diseños tomadas por el grupo para la segunda entrega del proyecto.

Tabla de revisiones

[illegible]

Introducción

El análisis de un proyecto consiste en entender los requisitos dados para tener la capacidad de resolverlos. Esto conlleva tomar decisiones debido a que puede haber más de una alternativa a la hora de desarrollar soluciones, cada una con sus pros y sus contras, y no siempre habrá una correcta, sino una que se adaptará mejor al proyecto que debe de ser identificada y escogida.

Este reporte está organizado de la siguiente forma:

1. Resumen ejecutivo: Introducción breve sobre el reporte.
2. Tabla de revisiones: Historial de revisiones del documento.
3. Introducción: Contextualización del análisis además de su importancia.
4. Contenidos: Descripción de algunas decisiones tomadas.
5. Conclusiones: Resumen de los hallazgos y la importancia de este reporte.
6. Bibliografía: Fuentes consultadas durante la investigación.

Contenidos

Decisión #1: Ocultar los botones a los miembros de la tripulación que no estén disponibles.

Requisito asociado: Requisito individual 8 del estudiante 3:

- List the flight assignments separately, one for completed flight legs and another one for those planned but that have not taken place yet.
- Show the details of their flight assignments and the associated legs and flight crew members.
- Create, update, and publish their flight assignments. Only crew members with du-ty “LEAD ATTENDANT” can perform these operations. Please, note that to publish a flight assignment these cannot be linked to legs that have already occurred. Additionally, only flight crew members with an "AVAILABLE" status can be as-signed to a leg, and they cannot be assigned to multiple legs simultaneously. Lastly, each leg can only have one pilot and one co-pilot. The allocation of re-maining roles for other flight crew members is at the discretion of the “LEAD AT-TENDANT”. Flight assignments can be updated or deleted as long as they have not been published.

Problema encontrado: Para no permitir dichas operaciones

Soluciones posibles valoradas: Permitir que salga el botón pero que no interactue vs. Ocultar los botones.

Pros:

Es más fácil de programar.

Contras

Es menos intuitivo, ya que al ver el botón el usuario tendería a pinchar sobre él y esperar la creación de una nueva asignación.

Pros:

Es más intuitivo, y permite mostrar solamente la funcionalidad que ese miembro de la tripulación puede realizar.

Da lugar a menos errores.

Contras

El uso de un booleano para restringir que salga el botón no se hace con un dataset.put, sino que darle más vueltas.

Solución adoptada: Ocultar los botones haciendo uso de un atributo global booleano en el método unbind.

Validación del profesor: Proporcionada en clase presencial.

Decisión #2:

Requisito asociado: Requisito 8 individual / 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

Requisito 9 individual / 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.

Problema encontrado: no saber si la relación entre booking y passenger estará bien.

Soluciones posibles valoradas:

Mantenerlo como está ahora (oneToMany de booking a passenger).

Pros:

Simplicidad de código (no hay que cambiar nada de antes).

Familiaridad con el código.

Contras

Dificultad para añadir nuevas funcionalidades

Poner una manyToMany entre booking y passenger.

Pros:

Mejor adaptación a los requisitos.

Contras

Complejidad de código (habrá que editar código creado previamente).

Necesidad de crear una tabla intermedia y sus operaciones necesarias.

Solución adoptada: se ha optado por la primera opción, creando una relación entre customer y passenger para poder manejar el requisito 9.

Validación del profesor: Proporcionada en clase presencial.

Conclusiones

Como conclusión, a raíz de la creación del proyecto y el inicio del trabajo en él, han surgido numerosas decisiones, tal y como se han mencionado anteriormente. Gracias a este documento, cualquier decisión tomada se puede corroborar de manera sencilla, manteniendo la trazabilidad y fomentando la creación de más versiones de este documento. A modo de resumen, hemos definido que todos los integrantes serán desarrolladores, analistas y testers a la vez, pero los roles de manager y operador irán rotando entre entregables.

Bibliografía

Intencionalmente en blanco