Memoria AIRDSS

Entrega 1

En esta primera entrega se dedicó a elegir qué proyecto se iba a implementar, que después de diversas propuestas, se decidió de forma unánime por la de AIRDSS.

A continuación, se empezó el desarrollo del diagrama de clases, en este debía haber mínimo 3 clases, aunque se realizaron un diagrama de 6 clases, aunque luego el equipo se centró en el desarrollo de 4 de ellas, pero debido a problemas se hizó la implementación de 3 de ellas de forma intensiva. Para el desarrollo del diagrama de clases trabajaron de forma conjunta todos los miembros del equipo de desarrollo.

Después se procedió a la implementación del diagrama de clases, donde previamente se creó el proyecto de Laravel y la BBDD, en el cual se repartió cada uno una clase:

- Cliente: Abraham
- Ticket: Alejandro
- Tarjeta de Embarque: Berta
- Vuelo: Daniel

Siguiendo con este reparto de tareas, cada uno realizo las migraciones y seeders de sus clases, para así tener creada la estructura de la BBDD así como unos datos de prueba.

Por último, se procedió a la realización de los test para comprobar que tanto los datos como la bbdd funcionaran correctamente.

Entrega 2

En esta segunda entrega se siguió con el desarrollo del proyecto.

Lo primero fue añadir las clases Admin, User , Package y Planes; aunque las dos primeras no se terminaron de incluir definitivamente. El grupo se dedicó a corregir errores que teníamos en el diagrama antes de eso. Esta parte la realizó conjuntamente todo el equipo.

Luego se procedió a la realización de los mockups, que aunque todos ofrecieron ideas, el desarrollo de los mismos los llevaron a cabo principalmente Abraham con ayuda de Alejandro, puesto que la herramienta que se usó(ninjamock) no permitía más componentes del proyecto en su versión gratuita.

A la hora de la implementación, cada uno siguió con cada una de sus clases y algunos también implementaron las clases auxiliares nuevas introducidas para esta entrega o que no se habían implementado en la entrega anterior. En el diseño, participaron todos los miembros, pero en gran medida Daniel y Alejandro. Cada uno realiza las siguientes clases con sus respectivas vistas:

- Abraham: Cliente
- Alejandro: Ticket y Package
- Berta: Boarding pass y Plane
- Daniel: Flight y Airport

Se ha de añadir que tuvimos un grave problema con la bbdd al añadir las nuevas clases, puesto que al hacer el rollback teníamos que borrar la base de datos todo el rato y volver a lanzar las migraciones y los seeders, porque daba problemas de integridad; este problema se debió a que los encargados de hacer esas clases realizaron modificaciones en las migraciones de la entrega anterior. Después de una tarde entera dedicada a solucionar este problema, se llego a la conclusión anterior, así como a la solución del problema. La solución fue extraer todas las nuevas claves ajenas y sus dependencias a nuevas migraciones en las que se modificara la estructura de las tablas. Esta corrección corrió a cargo de Abraham.

En cuanto a las funcionalidades pedidas para esta entrega, se expondrá las realizadas por cada componente del grupo en sus respectivas vistas:

- Mostrar listados para todos los objetos del modelo de dominio: Cada uno implementó esto en sus vistas.
- Borrar cualquier objeto en la base de datos: Todos los integrantes del grupo implementaron esto en sus vistas
- Crear y modificar objetos de 3 tipos distintos (para los demás se puede poblar la base de datos con Seeders): Se implementaron para todos los objetos excepto para Ticket.
- Paginación de listados: Cada uno implementó esto en sus vistas, usando la funcionalidad que ofrece laravel para esta tarea.
- Ordenar listados por algún campo: Cada uno implementó esto en sus vistas, usando el order by en las consultas. Para poder implementar esta funcionalidad en el buscador, se tuvo que hacer uso de session para tener almacenado los criterios de búsqueda y realizar las ordenaciones de esas búsquedas.
- Realizar búsquedas de una de las clases con 2 criterios de búsqueda distintos: La realizó Abraham en la vista de Listar clientes.

Entrega Final

Para esta entrega final se nos pide la entrega del proyecto funcional, añadiendo una serie de requisitos extra como son la validación de formularios, autenticación y registro de usuarios tanto administrador como cliente, capa de servicios, también se continuo con mejoras relacionadas con la interfaz.

En primer lugar se procedió a realizar los mockups de las nuevas vistas a implementar así como modificación de las vistas según el tipo de usuario sin registrar o registrado(cliente o admin), para facilitar el diseño e implementación de las vistas del proyecto se implemento Bootstrap mediante CDN (Content Delivery Network), para que estas fueran tanto user-friendly como responsive.

Además, se ha creado las nuevas vistas necesarias para la compra de un vuelo: una ventana para configurarlo (elegir asiento libre y si se tiene equipaje) y otra para el pago de este.

En relación a la autenticación de usuarios se procedió al traspaso de datos almacenados en la tabla Client a la tabla User, añadiendo un campo booleano que nos indique si el usuario es cliente o administrador, posteriormente se procedió a la refactorización relacionada con client y eliminación de dicha tabla.

A continuación aplicando el middleware Auth que integra Laravel, junto con otro creado para el proyecto que se encarga de verificar si el usuario es administrador o no, se procedió a la autenticación de las diferentes vistas.

En cuanto a la capa de servicio, se ha implementado la lógica que permite la compra de un determinado ticket de parte de un usuario concreto(eligiendo un asiento). Además, posibilita la opción de indicar si lleva equipaje o no. Todos los servicios con mucha lógica de negocio utilizan debugging, permitiendo conocer qué ocurre en cada momento.

Comprueba si el cliente, el vuelo y el asiento son valores correctos. Es decir, si el cliente existe, el vuelo existe y el asiento dentro de ese vuelo se encuentra disponible. Una vez realizado esta comprobación se procede a la compra(simulación de una compra real 50% de funcionar) y se actualiza la BBDD acordemente (nuevo ticket, nuevo boardinPass, actualización de flight y user).

Todo este proceso se realiza con transacciones, por lo que si ocurre un error es posible de realizar un rollBack y obtener el estado anterior a la compra.

Además, todo esto viene asociado a unas vistas que permiten configurar y realizar la compra propiamente dicha.

Tareas de entrega final

- Abraham: Diseño mockups de nuevas vistas y adaptación de las anteriores. Creación de las vistas contacto, perfil y modificación del perfil. Añadir a la vista vuelos un buscador con carrusel de imágenes. Ayuda en la creación final del UML, junto con el resto del grupo de trabajo.

- Alejandro: Diseño mockups nuevas vistas, implementación de Bootstrap, mejoras en diseño de las vistas, rediseño página de inicio de administración, autenticación en las vistas Ticket y Package, colaborar en solución de fallo en búsqueda de vuelo.

- Berta: Autenticación de vistas Plane y BoardingPass, adecuación al proyecto de la vista de Registro, Home y Reset Password. Notificaciones al administrador de vuelos cancelados, los vuelos se cancelan automáticamente al iniciar sesión el administrador (un vuelo aleatorio cada vez que inicia sesión) y aparece un aviso en la página de inicio al estar logeado con todos los vuelos cancelados.

- Daniel: Modificación del web.php de manera que solo se puede acceder a determinado grupo de rutas por diferentes middlewares. Creación de las vistas de configuración y compra de un ticket, junto a la lógica necesaria que permita la compra(BuyServices).

Valoración del grupo

Realmente, todos hemos concluido que el trabajo de equipo ha sido más o menos balanceado. Estamos totalmente satisfechos. Nos hemos involucrado todos y se ha realizado una coordinación eficiente sin conflictos.