

Mingo - Air ticket reservation system

In software architecture speak, this is a classic transaction processing system that requires one important ACID compliant transaction aware module(ticket reservation) surrounded with other support systems(such as check-in and notification aspects).

Requirements as documented in trial project screen

Public user requirements

- Ability to login using google+, linked in or twitter accounts - support for either one of them(using OAuth)
- Ability to select tickets based on date, location (if possible also account for multi-city and round trips)
- Allow users to use credit card (enable payment gateway integration support, paypal and may be our own version of wallet)
- Email notification for ticket confirmation, checkin notificatoin, checkin confirmation and cancellation confirmation(both user initiated and air personnel initiated)
- Online check-in support
- Cancellation support (between 48 hours to 4 hours of boarding time)

Airline staff requirements

- Reservation chard review - Email(stated) and in-app view(unstated)
- Reservation - unstated
- Cancellation and Refund
- Flight plan cancellation

Based on this understanding, I have conceived an approach that proposed following sub-modules(sub-systems) required to fulfill the stated needs,

1. Ticket pool management

A transactionally aware sub-system that helps in allowing reservation prospects to successfully block, reserve and book the tickets without over-stepping on each other. This sub-system needs to understand the intricacies of a scheduled(pre-defined) flight plan based on which the ticket pool needs to be populated for user selection.

2. Last mile integration/Payment gateway integration

This sub-system needs to offer support for external integration(should be proper web service) that enables the system under consideration to talk to 3rd party payment gateways. This should should offer the ability to let the user pick the choice of payment gateway to interact with. This sub-system and Ticket pool management sub-system will have handshaking to do.

3. Check-in management

This system is a ancillary system made available for the users as a value add and has no side-effect on ticket reservation system. It is independent of the above two systems and has isolated CRUD/Service layer needs.

4. Notification management

This system requires a certain set of batch processes to push email notifications to all the relevant parties about the bookings, reservation chart, online check-in invitation, cancellation notification, cancellation warning notification, flight plan cancellation notification.

5. User management

This system should offer support for google+ authentication(or OAuth) and also offer captive user authentication support.