

# Airline reservation system ars software requirements specifications computer scie...

[Environment](#), [Air](#)



Airline reservations system (ARS) is a online software application used to reserve and retrieve information and perform transactions related to air travel. Originally designed and performed by airlines, ARSes were later developed for the use of travel agencies. Major ARS operations that book and sell tickets for multiple airlines are known as Global distribution systems (GDS). Airlines have divested most of their direct holdings to dedicated GDS companies, who make their systems accessible to consumers through Internet gateways. Modern GDSes are providing the services like booking hotel rooms and rental cars as well as airline tickets. They also provide access to railway reservations in some markets although these are not always integrated with the main system.

## REQUIREMENTS DOCUMENT

First we are developing a Software Requirements Specification (SRS) document that specifies what an airline reservation system should and should not do. The SRS document is divided into five sections namely

### System Objectives

Mainly we discuss the goals and objectives of the system categorized based on the viewpoint of the airline company and the customer. They help in a top-down development of the SRS.

### System Context

This section clearly depicts the environment and boundaries of the ARS and the entities with which it interacts. It helps us see how the system fits into

the existing scheme of things. What the system will do by itself and what it expects other entities to do is clearly delineated.

### Functional Requirements

These requirements states the functions of the system - what it should do and what it should not. This will includes the most common requirements of the customer in addition to some additional features. like reserving tickets, rescheduling tickets etc. Freedom from ambiguity and navigability were kept in mind while documentation. A consistent terminology has been followed throughout and the terms are explained in the appendix. The subsections follow a logical sequence that reflects the real world. For example, a customer cannot reschedule a ticket unless he has bought one earlier and cannot buy one unless he has checked its availability.

### Non-functional Requirements

These are quality requirements that stipulate the performance levels required of the system for various kinds of activities. Numerical lower and upper limits set conditions on the response times, access times etc of the system. Sometimes, tradeoffs are necessary among various non-functional requirements.

### Future Requirements

As technology improving day to day, users needs are also increasing. so we have to update our applications time to time in order to satisfy the customers. These are the specifications which are not provided for now in

the current version of ARS but which could be incorporated into future versions. Some of these need advanced technologies and interfaces with other systems. The ARS could be designed in future to enhance the existing capabilities or add entirely new ones.

The assumptions and limitations of the ARS have been interspersed in the SRS to present the same in their proper context.

## REQUIREMENTS ANALYSIS DOCUMENT

### 1. System Objectives

1. 1 The Airline Reservation System (ARS) is a software application to assist an airline with transactions related to making ticket reservations, which includes blocking, reserving, canceling and rescheduling tickets.

#### 1. 2 From the viewpoint of the airline -

1. 2. 1 Minimize repetitive work done by the system administrator and reservation clerks.

1. 2. 2 Maintain consistency among different access modes, e. g. by phone, by web, at the information desk and across different physical locations. The users should be basically taken through the same steps by the system as they go through in conventional desk-reservation systems.

1. 2. 3 Maintain customer information in case of emergency, e. g. flight cancellation due to inclement weather. The profile can also be used by the

airline company to track user preferences and travel patterns to serve them better, plan routes, for better marketing and efficient scheduling of flights.

1. 2. 4 Maximize the revenue of the airline company by various means:

1. 2. 4. 1 Increase awareness among frequent travelers about various special offers and discounts.

1. 2. 4. 2 Minimize the number of vacant seats on a flight and maximize flight capacity utilization.

1. 2. 4. 3 Maintain the capability to adopt a flexible pricing policy. The price of the tickets should be dynamically determined based on how early, before the date of departure, the customer buys the ticket.

1. 3 A survey conducted by airline companies shows that users of an existing reservation system would respond favorably to an ARS that satisfied or helped them satisfy the following objectives:

1. 3. 1 Reduce effort and frustration for travelers in scheduling a trip, especially by reducing the search effort for the flight they need to take.

1. 3. 2 Show all possible combinations and itineraries available for a pair of origin-destination cities.

1. 3. 3 Reduce redundancy in the information required from the customers in order for them to buy tickets, create user accounts etc.

1. 3. 4 Check the validity of input data and give a feedback to the user in case of errors or inconsistency.

1. 3. 5 Provide flexible access modes to users - internet, telephone, PDA.

1. 3. 6 Protect customers' privacy concerns.

1. 3. 7 Make it easy for travelers to check the ticket status or make changes to their trip.

## 2. System Context

2. 1 The ARS will provide the following types of easy-to-use, interactive, and intuitive graphical and telephonic interfaces.

2. 1. 1 The ARS will provide an easy-to-use, intuitive Graphical User Interface (GUI) as part of the Clerk/Administrator's working desktop environment.

2. 1. 2 The ARS will also provide an interactive GUI, on the World Wide Web for the general customers.

The above two ARS interfaces shall help provide the following functionalities to the users - access to the ARS to check the flight schedule, availability of seats, ticket price and to block, reserve, cancel, and reschedule tickets.

The ARS will also provide an easy-to-use, simple telephonic user interface, which can be accessed by the customers through telephone or cell phone from anywhere. This interface shall provide access, only to the following functionalities, namely, check flight schedule and check ticket status

including any change in the flight timings. The functionality available through this telephonic interface is limited because of security constraints.

2. 2 The system and its environment and the interactions between them are depicted in the diagram below.

DB-Reservations

Flight Schedule Database

Customer

Via Web

DB-User

DB-Schedule

I

N

T

E

R

F

A

C

E

' CW'

DB-Geography

ARS software

INTERFACE ' Cp'

Customer

Via Phone

INTERFACE ' A'

Administrator

The closed boundary above clearly delineates the system and the environment. The diagram shows the interactions between the ARS software and the databases inside the system. There are three databases internal to the system and which the system maintains. DB-user is the database containing all the personal information of the registered users of the ARS. This can be updated by the user by logging in to the system. Information from this database is used during transactions like charging the credit card etc. DB-schedule is a copy of the flight schedule database. The latter exists independently and is updated by a flight scheduler system which is out of scope of the ARS. DB-schedule is updated with the latest status of the flight



schedule database whenever there is any change in the latter. For example, if a flight has been added to the schedule between two cities on Tuesdays, DB-schedule gets updated with this change through a process with which we are not concerned. It is external to the system and is out of the scope of this SRS. DB-schedule also contains the base prices of tickets for various flight numbers. DB-reservations are a database containing information regarding the number of seats available on each class on different flights. It has provision for marking how many of the reserved seats have been blocked but not yet bought. DB-reservations should update itself using DB-schedule, for example, if a new flight is added. DB-geography is a database, which contains information about the cities and towns serviced by the airline. The distance between all cities and towns is contained in a matrix form. There are three interfaces, one for the administrator, one for the customer via web and another for the customer via phone. The administrator can update DB-schedule with any changes in the base prices of flight tickets. The system uses a pricing algorithm and dynamically determines the actual price from this base price depending on the date of reservation vis-a-vis date of departure. The customer interfaces (web and phone) enable multiple functions which are described in the following section - section 3.

### 3. Functional Requirements

#### User Accounts

#### Registration and creation of user profile

#### Checking Availability

Making Reservations/Blocking/Confirmation

Confirm Ticket

Reschedule Ticket

Cancellation

Update Profile

View Ticket Status

Query Flight Details

Telephone access

User Accounts

The passenger, who will henceforth be called the 'user', will be presented with 3 choices by the reservation system, as the first step in the interaction between them. A user can choose one of these and his choice would be governed by whether he is a guest or a registered user and whether he wants to check the availability of tickets or also block/buy them. The terms 'registered user' and 'guest' are described below.

A user who has traveled by the airline earlier would have been given a user id and a password. He would have his personal information stored in the database referred earlier as 'DB-user'. This 'personal information' would be henceforth referred to as 'profile'. Such a user with a profile in DB-user shall

be called a 'registered user'. A registered user will be able to check the availability of tickets as well as block/buy a ticket by logging into the system.

A new user, on the other hand, would either have to

register himself with the system by providing personal information or

log into the system as a guest.

In case of 'a', the new user becomes a registered user.

In case of 'b', the new user would remain a guest.

A guest can only check the availability of tickets and cannot block or buy tickets.

But a registered user can also act as a guest if he only wants to check the availability of tickets. 'Availability of tickets' always refers to viewing the flight schedule for given days, the price of tickets and any discount offers.

The system shall present the user with an option to exit from the system at any time during the following processes.

Registration and creation of user profile

The system shall require a user to register, in order to carry out any transactions with it except for checking the availability of tickets. It will ask the user for the following information at the least - a user id, a password, first name, last name, address, phone number, email address, sex, age,

preferred credit card number. The system will automatically create a ' sky miles' field and initialize it to zero in the user's profile.

### Checking Availability

After logging in a user (either a registered user or a guest), the system shall request him to enter the following details - origin city and destination city. " City' is a generic term and refers to a city or town as the case may be. The origin and destination cities would be entered as text. The system shall now refer to the flight schedule database, referred to as ' DB-geography' earlier, and check if there is any ambiguity with the names of the cities. In case there are more than two cities with same name as entered by the user, the system shall list all of them (with more qualifications) and ask the user to select one of them. In case, either the origin or destination cities are not listed in DB-geography as being directly serviced by the airline, the system shall suggest the nearest city to which service is available, including the distance of the destination city from this nearest city.

After the origin and destination cities are ascertained, the system shall now access the flight schedule database, referred to as ' DB-schedule' , and checks if there is a direct operational service between the two cities. If not, the system shall suggest possible routes and transfer points using a ' route selection algorithm'. The user shall now be presented with a choice of either selecting one of the routes. In case he selects a route, the system shall fill in the intermediate stop over points and create a multiple trip itinerary for the user.

The system shall now ask the user to enter the following details - class, one-way or

round trip, departure date and the number of adult passengers, children and senior

citizens.

' Class' refers to business class/first class/club class/smoking/non smoking.

This choice shall be made by the user through a drop down menu indicating all the possible combinations of choices. One-way/round trip shall be either a drop down menu or a check box selection. ' Departure date' refers to either a single date or a range of dates, entered through a calendar-like menu. This menu shall not show dates in the past or those dates that are too ahead in the future(as determined by the airline policy). In case, the trip is a round trip, the system shall also ask the user to enter the departure date on the return trip. Having taken all the above input from the user, the system checks for any false entries like the departure date on the return trip being earlier than the departure date on the onward trip. In case of incompatibility, the system shall display a suitable error message and prompt the user to enter the information correctly. Having taken all of the information , the system shall now access the flight schedule database ' DB-schedule' and queries it using the input provided by the user. The system queries the reservation database ' DB-reservations' to check which of the flights on the schedule have seats available. The system displays the results in a suitable form (a tabular form) with the following information depicted - for each flight

number – the flight number, departure time in origin city, arrival time in destination city, the duration of the flight (taking into account the possibility of a change of time zone) and the number of seats available on that flight.

There can be several flights between two cities and all of them will be listed for the particular date that the user wants to depart from the Origin City. In case, the user has entered a range of dates, the system shall display all the flights for all those dates in the range.

If the user has requested a round trip, the system shall display two tables – one for the onward trip and one for the return trip. There will be a check box in front of each line in the table representing a flight with available seats.

The user is now asked to check one of the boxes reflecting a choice of a flight number and time. In case of a round trip, the user is asked to check one box each in the two tables.

The system shall now display the price of the ticket for the trip. This will be the sum of the prices for all the members of the travel party being represented by the user.

The system shall also list any rules regarding the cancellation of tickets – what percentage of the price will be refunded within what date ranges. This will be displayed as a table.

**Making Reservations/Blocking/Confirmation**

After having taken the user through the, Checking Availability, The system will now ask the user if he wishes to block/buy the ticket. If yes, and if the user has been a guest, he will have to first register and become a registered user and then log onto the system.

If the user is already a registered user, and if he has logged on already, he can block/buy the ticket, but if he has been acting as a guest, he will have to log on.

Having ensured that the user is logged on validly according the system compares the departure date with the system date. If the departure date falls within 2 weeks of the system date, the system informs the user that he has no option to block the ticket and asks him if he would like to buy it.

If the difference between the departure date and system date is more than 2 weeks, the system asks the user if he would like to block or buy the ticket. The system informs the user that he can block the ticket at no cost now. It also informs him that if he chooses to block the ticket, he should make a final decision before 2 weeks of the departure date. The system shall send an email to the user, 3 weeks before the departure date as a reminder, in case he decides to block the ticket now.

Having taken the input from the user, the system shall now proceed to update the reservation database DB-reservation. It will decrement the number of available seats on the particular flight for the particular class by the number of travelers being represented by the user.

In case of a blocking, the system makes a note of it in the database - to be used if the user doesn't turn up before 2 weeks of the departure date. It generates a blocking number and displays it for the user to note down.

In case the user buys the ticket, the system accesses his profile and charges the price of the ticket to his credit card number. It simultaneously generates a confirmation number and displays it to the user for him to note down. The ticket has been reserved.

It adds the mileage of the trip (accounting for the number of travelers) to the skymiles in his profile.

#### Confirm Ticket

A user who has earlier blocked a ticket after going through the previous steps required to either confirm the ticket before two weeks of the departure date or the ticket stands cancelled.

To let the user confirm a ticket, the system shall first log him on and ask for his blocking number. Then it accesses DB-reservation and removes the check mark, which so far represented a blocked seat. The seat is now confirmed and reserved for the user.

The system accesses DB-user and charges the price of the ticket to the credit card number of the user. It simultaneously generates a confirmation number and displays it for the user to note down. The ticket has been reserved.



It adds the mileage of the trip (accounting for the number of travelers) to the skymiles in his profile.

### Reschedule Ticket

The system shall present the user with an option to re-schedule his travel party's trip. In order to do this, the system first logs on the user and requests his confirmation number. It will not allow a user to reschedule a blocked ticket but only a confirmed ticket. Using this, it queries DB-reservation and presents the details of the trip to the user, including but not limited to origin city, destination city, date of departure and date of arrival (in case the trip is a round trip).

The system shall now ask the user to select new dates from the calendar-menu.

In case, there are no available tickets for the dates entered, it displays a suitable message informing him that rescheduling to that date is not possible.

In case there are tickets available, the system asks the user to select the flight number for the trip (another for the return trip if the trip is a round trip) and proceeds to update the database.

The system accesses DB-reservation and decrements the number of available seats on the flight(s) by the number of members in the user's travel party. It then increments the entry for the previous flight by the same

number to reflect an increase in the available seats on it as a result of the rescheduling.

The system now checks if there is any difference in the prices of the tickets. If so, it accesses DB-user and charges or credits the credit card as the case may be. The system generates a new confirmation number and displays it to the user.

### Cancellation

The system shall also give the user an option to cancel a confirmed ticket or a blocked ticket.

The latter case is simpler and will be dealt with first - the system shall first log on the user and request the blocking number. Then it accesses DB-reservation and updates it by incrementing the number of available seats by the number of people in the user's travel party.

In the former case, i. e., for a confirmed ticket, it asks for the confirmation number and accesses DB-reservation and presents the details of the trip.

It then lists the applicable rules for cancellation of tickets and depending on the system date and the departure date, it displays the % of the amount that would be refunded if the user cancels the ticket.

After the user cancels the ticket, the system generates a cancellation number and displays it for the user to note down. It accesses DB-reservation and updates it by incrementing the number of available seats on that flight

by the number of travelers in the user's party. It accesses DB-user and credits the refund amount to his credit card number. The system then deducts the mileage of the trip (taking into account the number of travelers in his party) from the sky miles in his profile.

#### Update Profile

The system shall enable the user to update his profile at any time. Changes can be made in fields including but not limited to address, phone number and preferred credit card number.

#### View Ticket Status

The system shall allow a user to view all information about his trip. After logging him on, it asks for his blocking number or his confirmation number. It accesses DB-reservation and retrieves the details of the trip and presents them to the user in a convenient format, including any last minute changes to the flight timings etc. Such changes will be highlighted.

#### Query Flight Details

The system shall allow any user (registered or non registered) to access the details about the arrival and departure times of a flight by requesting the user to input the flight number and date. The system accesses DB-schedule and presents the time of arrival and departure.

#### Telephone access

The system shall be accessible through a touch-tone telephone. The telephonic interface shall, at the least, provide the customer with the facility to check availability of tickets and query flight details. The system shall walk the customer exactly through steps 3. 3 and 3. 9 respectively but through a telephonic interface.

#### Non-functional Requirements Performance

Response time of the Airline Reservation System should be less than 2 second most of the time. Response time refers to the waiting time while the system accesses, queries and retrieves the information from the databases (DB-user, DB-schedule etc) (A local copy of flight schedule database is maintained as DB-schedule to reduce this access time)

ARS shall be able to handle at least 1000 transactions/inquiries per second.

ARS shall show no visible deterioration in response time as the number of users or flight schedule data increases

#### Reliability

ARS shall be available 24 hours a day, 7 days a week

ARS shall always provide real time information about flight availability information.

ARS shall be robust enough to have a high degree of fault tolerance. For example, if the user enters a negative number of passengers or a value too

large, the system should not crash and shall identify the invalid input and produce a suitable error message.

ARS shall be able to recover from hardware failures, power failures and other natural catastrophes and rollback the databases to their most recent valid state.

### Usability

ARS shall provide a easy-to-use graphical interface similar to other existing reservation system so that the users do not have to learn a new style of interaction.

The web interface should be intuitive and easily navigable Users should be able to understand the menu and options provided by ARS.

Any notification or error messages generated by ARS shall be clear, succinct, polite and free of jargon.

### Integrity

Only system administrator has the right to change system parameters, such as pricing

policy etc. The system should be secure and must use encryption to protect the

databases.

Users need to be authenticated before having access to any personal data.

## Interoperability

ARS shall minimize the effort required to couple it to another system, such as flight schedule database system.

## Future Requirements

### Support for waiting list functionality

ARS shall be made more flexible in ticket reservation handling, and shall accept waiting list for reservation. The waiting list handling capability of ARS shall be made more advanced, by enabling it to send requests to the Flight Scheduler to schedule extra flights, depending on the demand in a particular corridor, and providing the wait listed passengers with a new flight.

The telephonic interface of the ARS shall be improved to support more functionality like

allowing the customers to cancel a ticket etc., by incorporating security measures.

ARS shall be made more dynamic and helpful to the users by enabling it to send instant

messages to the passengers, of a cancelled or rescheduled flight, through email, phone,

Fax etc., informing them about the change, and providing them with other feasible

alternatives.

Information about the kind of meals served in a flight and the type of entertainment offered on a flight should be incorporated into the system.

Provide service

integration with auto rental agencies and hotel chains.

Interface for the travel agents shall be provided in the future versions with additional features like informing them of any availability of seats on a flight which was earlier booked to capacity.

Choices like aisle or window seats shall be provided to the users.

The ARS shall be able to handle the situation where flight services are available to multiple airports in a single city.