Version Delivery Scheduling Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Wave | Phase/Product Version | Bundle | Iteration | Notes |
| (Mandatory) What wave is this version delivered to? | (Mandatory) What Phase or Product Release is this version delivered to? | (Optional) What Bundle is this version delivered to? | (Optional) What Iteration is this version delivered to? | Any other notes relating to the scheduling of this version |

Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Approved By | Signed | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Author:

Document version:

Document Date:

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 2012-Apr-24 | 0.0a | Initial Draft | James Ellison |
| 2012-May-10 | 0.0b | Update from Peer Review | James Ellison |
| 2012-May-17 | 0.0c | Update from Peer Review | James Ellison |
| 2012-May-21 | 0.0d | Update from Peer Review | James Ellison |
| 2012-May-22 | 0.0e | Update from Peer Review | James Ellison |
| 2012-May-22 | 0.0f | Update from Peer Review | James Ellison |
| 2012-Jul-27 | 0.0g | Added story breakdown and acceptance tests | Andrey Golovachev |
| 2012-Aug-30 | 0.0h | Update from Peer Review | Rick Winstel |
| 2012-Sep-06 | 1.0a | Added elaboration details for calculation of full numeric availabilityUsecase for Simulate Availability  Removed advanced flow 2 - view of availability detail.  Merged exception flows 1 and 2 which were different for basic and advanced search though the contents were the same. | Andrey Golovachev |

Contents

1. Brief Description 5

1.1 Background 5

1.2 References 5

1.3 Assumptions 6

2. Actors 6

3. General Preconditions 6

4. Basic Flow of Events 6

4.1 Basic Flow 1 – Basic Simulate Availability Request 6

4.1.1 Specific Preconditions 6

4.1.2 Steps 6

4.1.3 Specific Post Conditions 6

5. Alternate Flows 6

5.1 Alternate Flow 1 – Advanced Simulate Availability Request 6

5.1.1 Specific Preconditions 6

5.1.2 Steps 7

5.1.3 Specific Post Conditions 7

6. Exception Flows 7

6.1 Exception Flow 1 – Validation Error for Basic Request 7

6.1.1 Specific Preconditions 7

6.1.2 Steps 7

6.1.3 Specific Post Conditions 7

7. Sub Flows 7

8. General Post Conditions 8

9. Extension Points 8

10. Special Requirements 8

10.1 Navigation Requirements 8

10.2 Usability Requirements 8

11. Additional Information 8

12. Changes to Reviewed Use Cases 8

13. Future Use Case Considerations 8

14. Issues 8

15. Use Case Elaboration & Story Breakdown 8

15.1 Basic Availability Simulation GUI – INV.11 8

15.2 Full Availability Simulation GUI – AVAIL.25a 9

15.3 Full Availability Simulation GUI – AVAIL.25b 9

# Brief Description

This use case defines the Horizon Inventory Availability and Schedules (HIAS) Simulate Availability functionality. This function allows the user to see current availability levels, test the impact of new business rules on availability and to test any Point-of-Sale-based restrictions set via Booking Limt buckets.

## Background

Flight availability returned by NGI is the result of a financial availability calculation, allocation availability calculation and rules (influences) that manipulate the calculations and the response.

Calculation of financial availability from bid prices and rules and the effect of influences combined can make the NGI availability response obscure. In some cases, it will not be clear why NGI returns any given flight availability.

To provide visibility and aid diagnosis of problems, NGI must provide the Inventory Analyst or central site administrator proving support with the capability to initiate and trace availability requests from different points of sale including the influences that were activated in computing the response.

The business needs addressed by this Use Case include:

* Verifying the core availability calculation logic is working as expected including the validation that the needed inputs are flowing through as expected
* Confirming that Influences are having their desired impact and not causing any unintended side effects
* Testing the impact of new Influences prior to deploying them
* Verifying the responses for upper compartment requests which may not be using Financial Availability but can be affected by influences

## References

|  |  |  |
| --- | --- | --- |
| **Ref. ID** | **Reference** | **Document/ Reference Version** |
| [1] |  | 1.0b |
| [2] | System BDD for POS Values | 0.0f |
| [3] | IAS Story Detail INV.11 Availability GUI – story description, the first iteration of Availability GUI |  |
| [4] | release2\SAD\IAS-SAD-Drawings.vsd – Architecture overview diagram |  |

# Assumptions

* Display the most restrictive availability of leg, segment or POS-specific
* Display in Cabin Order – RBD within Cabin (hierarchy that is defined in the schedule)

# Actors

* **Primary Actor:** The primary actor is an Inventory Analyst

# General Preconditions

* Actor is authorized to perform the functionality and has successfully logged into the system
* The system initialization has taken place with all appropriate data.

# Basic Flow of Events

## Basic Flow 1 – Basic Simulate Availability Request

### Specific Preconditions

* Availability information exists for certain flights

### Steps

1. Actor initiates Horizon Inventory AS Availability Simulation Search
2. System responds with Simulation Search GUI Screen
3. Actor provides the **Search parameters from BDD Simulate Availability – section 2.1**
4. Actor submits search criteria
5. System validates entered search criteria
6. System notifies Actor of success with display of results screen

### Specific Post Conditions

* System displays availability information for travel options and segments matching the search criteria entered by the user

# Alternate Flows

## Alternate Flow 1 – Advanced Simulate Availability Request

### Specific Preconditions

Basic flow step 2 has been completed

### Steps

1. Actor initiates Horizon Inventory AS Availability Simulation Search
2. System responds with Simulation Search GUI Screen
3. Actor selects the Advanced Panel on the Simulation Search GUI Screen
4. Actor provides the data **for the advanced search from BDD Simulate Availability – section 2.2**
5. Actor submits information
6. System validates entered search criteria
7. System invokes Use case for POS Data Propagation (AVAIL.24 story) to apply POS restrictions.
8. System notifies Actor of success.

### Specific Post Conditions

* System displays availability information for travel options and segments matching the search criteria entered by the user. The system takes into account any existing POS restrictions that match the user’s request.

# Exception Flows

## Exception Flow 1 – Validation Error

### Specific Preconditions

* User has defined input for Simulate Availability request and attempted to proceed with execution

### Steps

1. System receives error response during processing the Simulate Availability request
2. System indicates the field which caused the error
3. System retains the information entered
4. User has the opportunity to correct the data and resubmit the Simulate Availability request

### Specific Post Conditions

* If another is found, the system responds with appropriate error message.
* If no error occurs during processing, system displays requested availability

# Sub Flows

None

# General Post Conditions

* Users will have the option to export the Search Results to an external comma-separated values (CSV) file.

# Extension Points

None.

# Special Requirements

None.

# Future Use Case Considerations

* Ability to display ISSR availability
* IAM will manage the users that will have the capability of requesting and viewing:
  + Full Numeric Availability
  + Group Availability

# Use Case Elaboration & Story Breakdown

### Basic Availability Simulation GUI – INV.11

Story description is available on SVN. The scope included Host only availability simulation returning status availability information.

### Availability simulation GUI (resolve gaps) – AVAIL.25a

The scope of this story is to identify and resolve gaps from the original implementation of INV.11 story.

The known gaps include:

1. The order of RBDs in availability display must match the order of RBDs for this segment in schedules. For reference only, RBDs in schedule legs:  
   tns:ShowFlightSchedulePeriod/tns:ScheduledFlightOperationPeriod/sch:Legs/sch:Leg/sch:Cabin/sch:BookingClassList  
     
   Segment RBD overrides:  
   tns:ShowFlightSchedulePeriod/tns:ScheduledFlightOperationPeriod/sch:LegOverride/sch:BookingClassList  
     
   Note: inclusion of specific schedule RBDs in inventory is determined by Assign Nesting Template and dressing functionality which has been already implemented.
2. Within availability response string, i.e.   
   *ShowFlightAvailability/tns:OriginDestination/inv:ODOptions/inv:ODOption/inv:Segment/inv:RbdList*  
   and  
   *tns:ShowFlightAvailability/tns:OriginDestination/inv:ODOptions/inv:ODOption/inv:Segment/inv:Cabin/inv:RbdList*

colon must be used to delimit RBD and availability value, and commas must be used to delimit RBD-value pairs, for example: Y:5,B:C,HN:A  
This is required to support 2-character PRBMs (“night flight indicators”) and enable the GUI client to parse availability response for multi-column display in Availability Simulation screen.

1. “EQP” string should be displayed in Aircraft type column of search results if aircraft type varies across the segment’s legs (as indicated by inv:Segment/ inv:EquipmentType/inv:EquipmentChangeOfGuageInd)
2. Number of stops column needs to be displayed, see BDD for schema mapping.
3. Flight number must include suffix if it is present in schedules. Suffix should be kept in grid’s copy of schedules.

Other gaps should be reported if identified with regression testing. Once this story is complete, INV.11 will become deprecated, UC and BDD will become the master.

### Full Availability Simulation GUI – AVAIL.25b

**Details**

The scope of the story includes:

1. Advanced availability simulation:  
   - Section 5 of this use case, section 2.2 of the BDD. The existing functionality of AVAIL.24 (POS Data Propagation) is used to apply POS restrictions to availability during simulation.
2. GUI layout changes and enhancements (see BDD and wireframes for details):  
   - Inventory Status fieldset should be removed from search criteria.  
   - The radio button in basic search needs to be changed.  
   - Collapsible “Tree tables” layout needs to be used in search results.  
   - Add segment number and airline columns  
   - Make Date Range a radio button  
   - Remove max routes field  
   - Hovering over the values of “Stops” should list stop locations within a segment. (note: for stop locations to be returned in response, the following field must be set to True in the request: SearchForFlightAvailability)/tns:SearchInfo/inv:ReturnStopLocationsInd)  
   - Default sort order should be Departure time  
   - support of full numeric availability, groups and waitlists  
   - each RBD is a separate column, the availability value is zero filled. Please note that the GUI client will need to parse the availability response string returned by *InventoryEnquirer*, strip colons and treat commas as column delimiters; and also zero-fill full numeric values up to 3 digits.
3. Displaying of full numeric availability for groups upon user’s request (see BDD sections 2.2 and 2.3 for details, supported with AVAIL.26 story on the server side)
4. Display status and full numeric availability for waitlist (see BDD section 2.3 for details, supported with AVAIL.26 story on the server side)
5. Displaying of full numeric availability upon user’s request (see below).

**Non-functional requirements**

As per supplementary specification, in 99.5% of cases the response time should not exceed 2000ms for a set of 20 travel options with 2 segments each displayed on a single search results screen. Worst-case scenario needs to be assumed, i.e. full numeric availability specific to POS with multiple attributes was requested; while 20 POS templates are set up, each of 10 items (test data of AVAIL.24 story can be re-used).

**Acceptance Tests**

| **Nr.** | **Test** | **Pre-condition** | **Action** | **Post-condition** |
| --- | --- | --- | --- | --- |
|  | Search for “host only” travel options with POS restrictions applied. | An ICR record exists with an associated booking limit bucket, which in turn has a POS template associated. The Seats Available counter calculated for that BL bucket within ICR is less than a nested bucket’s SA for a certain RBD(s) | The user submits availability simulation request providing POS details (as per section 2.2 of the BDD) which match the POS details of the BL bucket associated to one of the existing ICR records.  The “airline code” radio button is in its default “host only” position. | The availability display SA levels in simulation results account for BL bucket limits: SA for BL bucket is displayed for the RBD(s) in question. Multiple travel options with several segments within each need to be demonstrated.  Note: application of Min/Max booked passengers limit and Upline/Downline BL conditions will be subject of future stories as well as application of POO restrictions.  The returned travel options include host-to-host (online) connections only. |
|  | Search for “Host&any OA” travel options with POS restrictions applied. | Same as above | Same as above, the “airline code” radio button is in “Host&any OA” position.  The “sort by” dropdown is in its default “departure time” position. | Same as above.  The returned travel options include online as well as interline connections and are sorted by departure time.  Availability for OA segments should be based on schedules (open by default) and account for received AVS messages (which may close availability). This was implemented as part of TECH.15b story in Stage 4.  Hovering over the values of “stops” columns shows stop locations (airport codes, comma separated) |
|  | POS-specific request for full numeric availability for multiple dates sorted by arrival time. | Same as above | Same as above, the Date Range radio button is in “+/- 3 days” position.  The “sort by” dropdown is in “arrival time” position.  The availability type is set to “full numeric”. | Same as above.  The returned travel options are sorted by arrival time and include those within 3 days of the specified departure date.  Full numeric availability is displayed in accordance with the BDD and wireframes.  Note: performance needs to be measured and reported. |
|  | Display group availability | Same as above | The user selects to display group availability in availability simulation. | Full numeric availability for groups is displayed in search results in accordance with the BDD and wireframes |
|  | Display waitlist availability | Same as above | The user selects to display waitlist availability in availability simulation. | Full numeric availability for waitlist is displayed in search results in accordance with the BDD and wireframes |
|  | Invalid POS attributes in search request | None specific | The user attempts availability simulation providing POS attributes which are incorrect in accordance with the System BDD for POS values, for example, user type A is combined with IATA number. | An error message is displayed. |

### Support full numeric availability – AVAIL.25c

**Details**

Please note that unless otherwise stated, the changes listed below must be supported for both *InventoryEnquirer*’s ports: *SearchForFlightAvailability* and *SearchForFlightSpecificAvailability*

Also note that within *SearchInfo* the *FullNumericAvailabilityInd, GroupAvailabilityInd* and *WaitlistAvailabilityInd* elements are mutually exclusive.

The scope of the story includes:

* 1. Return full numeric availability upon request and apply POS restrictions when specified

A new optional Boolean field has been added to availability request:  
*tns:SearchInfo/tns:FullNumericAvailabilityInd* If set to true, then full numeric availability (i.e. numbers not capped by 9) should be returned in the response.

Please refer to SAD architecture overview diagram [4] for location of the components mentioned below. For regular numeric availability requests (0-9 seats, A, C and L statuses in response) pre-calculated NAVN availability data from *FlightNumericAvailabilityStatus* (part of *FlightManager* in grid) is normally used.

* However for full numeric availability requests the system should bypass NAVN data and instead interrogate *InventoryControlRecord* and calculate full numeric availability on the fly, exactly how it is already done when NAVN is created/updated. The only difference is that full numeric availability response can return more than 9 seats available, and “A” status is never returned in such case. For reference only, the following usecase describes availability calculation: NGI UC for Basic Inventory Adjustment-CFT 0.2.docx section 4.2.2  
    
  An example response to full numeric availability request would be (assuming AVAIL.26 story is implemented and the waitlist status is returned):

Y:112,B:34,M:9,H:L,Q:C

* AVAIL.24 story (NGI UC Data Propagation to POS) describes application of POS restrictions to regular numeric availability. However for full numeric POS-specific availability requests POS-NAVN should be bypassed and instead full numeric availability should be calculated exactly how it is prescribed by AVAIL.24 when POS-NAVN is created/updated. I.e. create a list of POS templates matching POS attributes of the availability request, determine booking limit buckets associated with those POS templates, and for every RBD select the most restrictive of leg, segment or applicable BL constraints. The only difference is that the response could return more than 9 seats available, and “A” status is never returned in such case.

Schema path to POS attributes in availability request (for reference only):  
*tns:SearchForFlightAvailability/tns:POS*

* Note: Availability for OA segments should still be based on Schedules and received OA AVS messages as implemented previously within TECH.15bc stories.
* Note: Full numeric availability support is currently required only for the purposes of Availability Simulation in IAS GUI.
* Full numeric availability requests will incur a small performance overhead since they have to interrogate the ICR to determine actual seat counts. The additional performance overhead should be no more than 100ms as compared to NAVN-based regular availability request and 300ms if POS restrictions are specified.
  1. Apply POS restrictions to Group availability
* AVAIL.26 story describes how Groups availability should be returned if specifically requested for. To enable full support of Simulation GUI, the current story must apply any POS restrictions specified in the group availability request similarly to 1)
* Same performance requirements as for 1)
  1. Return Waitlist Seats Available upon request.

A new optional Boolean field has been added to availability request:  
*tns:SearchInfo/tns:WaitlistAvailabilityInd*

* If set to true, then full numeric Waitlist Seats Available counts (WSA) should be returned instead of regular availability (i.e. SA counts). In such scenario WSAs should be returned regardless of SA levels or Waitlist Allowed (WLA) indicator.  
    
  For reference only, below is the path to WSA within the ICR schema:  
  *tns:ShowICR/tns:InventoryControlRecord/tns:Legs/tns:FlightCabin/tns:TreeTopBucket/tns:NestedBucket/tns:AllocationCounters/tns:WaitlistSeatsAvailable*   
  and to WLA indicator:  
  *tns:ShowICR/tns:InventoryControlRecord/tns:Legs/tns:FlightCabin/tns:TreeTopBucket/tns:NestedBucket/tns:AllocationControls/tns:WaitlistAllowedInd*
* If the user is requesting for waitlist availability by setting WaitlistAvailabilityInd to true, then “C” should be appended to WSA levels in the availability response for those RBDs with WLA indicator=false, and “L” when WLA=”true”. This will allow the user to see if waitlist has been forced closed, for instance via ICR GUI.
* Any POS restrictions specified in the waitlist availability request must be applied similarly to 1)
* Note: Full numeric waitlist availability support is currently required only for the purposes of Availability Simulation in IAS GUI.
* Same performance requirements as for 1)

An example availability response to waitlist availability request would be:

Y:23L,B:8C,M:0L,H:0C

**Acceptance Tests**

| **Nr.** | **Test** | **Pre-condition** | **Action** | **Post-condition** |
| --- | --- | --- | --- | --- |
|  | POS-specific full numeric availability request | An ICR record exists with two or more associated booking limit buckets, which in turn have a POS template associated. The Seats Available counter for each of those BL buckets within ICR is less than a nested bucket’s SA for a certain RBD(s) | The user submits availability simulation request providing POS details (as per section 2.2 of the BDD) which match the POS details of the BL bucket associated to one of the existing ICR records.  Full numeric availability is requested by setting *FullNumericAvailabilityInd* to true. | Full numeric availability accounting for POS constrains is returned.  Need to demonstrate:   * + - Multiple travel options returned, with several segments in each     - Multiple POS and BL templates set up in the system     - Two or more BL buckets of the ICR match POS attributes of the availability request, the most restrictive is selected     - Match of the request’s POS attributes with a POS template occurs when all values of at least one POS item match |
|  | POS-specific Group availability request | Same as above | Same as above, but Groups availability is requested by setting *GroupBookingInd* to true | POS-constrained full numeric availability for Groups is returned as per AVAIL.26 story. |
|  | POS-specific Waitlist availability request | Same as above | Same as above, but Groups availability is requested by setting *WaitlistAvailabilityInd* to true | POS-constrained full numeric availability for Waitlists is returned. RBDs for which waitlists are not allowed are denoted with “C”, all other RBDs – with “L”. |