Software Architecture Design

Project: Air Booking

Author: Igor Podsekin, [wonderu@wonderu.com](mailto:wonderu@wonderu.com)

Moscow, 2016

Content

[Revision History 4](#_Toc453494407)

[1. Introduction 5](#_Toc453494408)

[1.1 Purpose 5](#_Toc453494409)

[1.2 Scope 5](#_Toc453494410)

[1.3 Definitions, Acronyms and Abbreviations 5](#_Toc453494411)

[2. Architectural Goals and Constraints 6](#_Toc453494412)

[2.1 Technical Platform 6](#_Toc453494413)

[2.2 Transaction 6](#_Toc453494414)

[2.3 Security 6](#_Toc453494415)

[2.4 Persistence 6](#_Toc453494416)

[2.5 Reliability/Availability (failover) 6](#_Toc453494417)

[2.6 Performance 6](#_Toc453494418)

[2.7 Internationalization (i18n) 6](#_Toc453494419)

[3. Requirements 7](#_Toc453494420)

[3.1 Functional Requirement 7](#_Toc453494421)

[3.2 Nonfunctional Requirements 7](#_Toc453494422)

[3.3 Limitations 7](#_Toc453494423)

[4. Use Case Model 8](#_Toc453494424)

[5. Data Model 9](#_Toc453494425)

[6. Layouts Model 10](#_Toc453494426)

[7. Components Model 11](#_Toc453494427)

[7.1 API-service 11](#_Toc453494428)

[7.2 Public module 11](#_Toc453494429)

[7.3 Staff module 11](#_Toc453494430)

[7.4 Scheduler 11](#_Toc453494431)

[7.5 HTML-application 11](#_Toc453494432)

[7.6 SMTP Relay 12](#_Toc453494433)

[8. Implementation 12](#_Toc453494434)

[9. Deployment Model 14](#_Toc453494435)

[10. Process Models 17](#_Toc453494436)

[10.1 Booking Business Process 17](#_Toc453494437)

[10.2 Check-In Business Process 19](#_Toc453494438)

[10.3 Public User Authentication Process 20](#_Toc453494439)

[11. User Interface mockups 21](#_Toc453494440)

[11.1 Public website user interface 21](#_Toc453494441)

[11.1.1 Public front page 22](#_Toc453494442)

[11.1.2 Public login 22](#_Toc453494443)

[11.1.3 Search flight form 23](#_Toc453494444)

[11.1.4 Search flight result 24](#_Toc453494445)

[11.1.5 Passenger’s info and payment 25](#_Toc453494446)

[11.1.6 Payment confirmation 26](#_Toc453494447)

[11.1.7 My bookings 26](#_Toc453494448)

[11.1.8 Booking details 27](#_Toc453494449)

[11.1.9 Cancel booking 27](#_Toc453494450)

[11.1.10 Cancel booking confirmation 28](#_Toc453494451)

[11.1.11 Check-in 28](#_Toc453494452)

[11.1.12 Select seat 29](#_Toc453494453)

[11.1.13 Check-in confirmation 29](#_Toc453494454)

[11.2 Staff website user interface 30](#_Toc453494455)

[11.2.1 Staff login 30](#_Toc453494456)

[11.2.2 Search flight 31](#_Toc453494457)

[11.2.3 Flight details 31](#_Toc453494458)

[11.2.4 Cancel ticket confirmation 32](#_Toc453494459)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 2016-06-10 | 0.1 | Initial version | Igor Podsekin |
| 2016-06-11 | 0.2 | System context and components | Igor Podsekin |
| 2016-06-12 | 0.3 | Architecture draft | Igor Podsekin |

# Introduction

## Purpose

The Software Architecture Design (SAD) provides a comprehensive architectural overview of the Air Booking Website. It presents a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

## Scope

The scope of this SAD is to depict the architecture of the Air Booking website.

## Definitions, Acronyms and Abbreviations

**UML –** Unified Modeling Language

**AD** – Active Directory

**AD FS** – Active Directory Federated Services

**API** - Application programming interface

**FR** – Functional requirement

**NFR** – Nonfunctional requirement

**OAuth** – is an open standard for authorization, commonly used as a way for Internet users to log in to third party websites using their Microsoft, Google, Facebook, Twitter, One Network etc. accounts without exposing their password

**PNR** - passenger name record is a record in the database of a computer reservation system that contains the itinerary for a passenger, or a group of passengers travelling together

**SMTP** – Simple Mail Transfer Protocol is an Internet standard for electronic mail (email) transmission

**3D Secure** – protocol designed to be an additional security layer for online credit and debit card transactions.

**DAL** – Data Access Layer

**BLL** – Business Logic Layer

**PL** – Presentation Layer

**JS** – JavaScript

**REST** – representational state transfer

**CRUD** – create, read, update and delete. They are the four basic functions of persistent storage

**WebAPI / Web API** - application programming interface (API) for either a web server or a web browser

**VPN** – virtual private network

**DMZ** – demilitarized zone. It is a physical or logical subnetwork that contains and exposes an organization's external-facing services to untrusted network, usually the Internet.

**IIS** – Internet Information Server

**SAD** – Software Architecture Design

# Architectural Goals and Constraints

This section describes the software requirements and objectives that have some significant impact on the architecture.

## Technical Platform

Air Booking online application will be deployed onto a IIS server as a .NET-application.

## Transaction

Air Booking online application is transactional, leveraging the technical platform capabilities. Transaction management model of the .NET platform will be reused intensively.

## Security

The system must be secured, so that a customer can make online payments.

The application must implement basic security behaviors:

* Authentication:
  + Login using user name and a password;
  + Login using OAuth provider (Google, Twitter and LinkedIn);
  + Login using special one-time URL for accessing from emails.
* Authorization: public user and staff roles

For internet access, the following requirements are mandatory

* Confidentiality: sensitive data and channels must be encrypted (credit card payments)
* Data integrity : Data sent across the network cannot be modified by a tier
* Auditing: Every sensitive action can be logged
* Non-repudiation : gives evidence a specific action occurred

.NET security model will be reused.

## Persistence

Data persistence will be addressed using a relational database MS SQL Server.

## Reliability/Availability (failover)

The availability of the system is a key requirement by nature, as it is a selling system. The candidate architecture must ensure failover capabilities. Reliability/Availability will be addressed through the .NET platform and load balancing.

Targeted availability is 24/7: 24 hours a day, 7 days a week

## Performance

The payment process (credit card authorization and confirmation) must be under 10 seconds.

## Internationalization (i18n)

The application doesn’t have internationalization capabilities and should be support only English language.

# Requirements

Describes the requirements with a significant impact on the architecture.

## Functional Requirement

|  |  |
| --- | --- |
| ReqNo | Description |
|  | Public user should be able to login to the site |
|  | Public user should be able to login to the site using Google Account |
|  | Public user should be able to login to the site using Twitter Account |
|  | Public user should be able to login to the site using LinkedIn Account |
|  | Public user should be able to search air tickets |
|  | Public user should be able to search air tickets by selecting departure and return days |
|  | Public user should be able to search one-way and return air tickets |
|  | Public user should be able to search air tickets by setting departure and arrival airports |
|  | Public user should be able to search air tickets for multiple people |
|  | Public user should be able to reserve air tickets |
|  | Public user should be able to make a payment for reservation |
|  | Public user should receive tickets as PDF via email. This email should contain useful links like “My Bookings”, “Check-In Now”, “Cancel booking” that will not require the user to login. |
|  | Public user should be able to perform online check-in/cancellation, from 48 hrs and upto 4 hrs from departure, choose seats, and print boarding passes |
|  | Public user should receive email alert at 48 hrs from departure, regarding their reservation, to encourage them for early online checkout. This email should contain a “Check-In Now” link that will not require the user to login |
|  | Airline staff should be able to login using their active directory credentials, even from public computers |
|  | Airline staff should be able to receive reservation chart as PDF at 1 hrs before departure |
|  | Airline staff should be able to cancel any ticket, generating email alert and refund to the public user |
|  | Airline staff should be able to cancel the whole flight, generating email alerts and refunds to the public users |

## Nonfunctional Requirements

|  |  |
| --- | --- |
| ReqNo | Description |
|  | The system should be created as a Single-page Application (SPA) |
|  | The system should support all modern browsers (the browser list should be defined in Technical Specification) |
|  | The system should have responsive design |
|  | System core must be ready to serve mobile apps in future w/o any modification |
|  | System should be installed on cloud |
|  | All the alerts sent should be guaranteed (not lost due to some technical errors) |
|  | Only one airline is supposed to use this system for its own customers |
|  | All communications should be encrypted using SSL/TLS-encryption |

## Limitations

|  |  |
| --- | --- |
| ReqNo | Description |
|  | The system should not be able to postpone flights, change gates etc. |
|  | The system should be allow to book only direct flights |
|  | The system will have no flight classes |
|  | The system will not track the baggage and its options |
|  | This document doesn’t describe user registration use case |
|  | The system will have only one language and one currency |

# Use Case Model

Describes the set of scenarios and/or use cases that represent some significant, central functionality of the system.

C:\Users\ipodsekin\Downloads\Public user use cases (4).png

Picture 1. Public user use cases diagram

C:\Users\ipodsekin\Downloads\Staff Use Cases (4).png

Picture 2. Staff use cases diagram

Use cases description should be provided as separate document “Use Cases Decription.pdf”. The description should contain name, description, number, actors, triggers, preconditions, postconditions, main success scenario and exceptions.

# Data Model

Describes the architecturally significant persistent elements in the data model

C:\Users\ipodsekin\Downloads\DataModel (7).png

Picture 3. Data model diagram

Picture 3 shows the high-level data model of air booking system. The table describes entities and their relationships below.

|  |  |
| --- | --- |
| Entity | Description (if needs) |
| Airport[[1]](#footnote-1) | Information about airport |
| Airplane1 | Has an image of airplane for “Select seat” UI |
| Seat1 | Seats in airplane. It has seat coordinates (X, Y) mapped to Airplane image (for “Select seat” UI) |
| FlightInfo1 | Information about airport, gate and time |
| Flight1 | Contains flight code and FlightInfo for departure and arrival |
| FlightSchedule1 | FlightSchedule represents the scheduled flight: when, airplane and status |
| User | The public user |
| Booking | The combination of ordered outbound and return tickets |
| Ticket | Ticket represents the physical electronic ticket or boarding pass |
| Passenger | Information about a passenger on the ticket |

# Layouts Model

The system will have several layers.

C:\Users\ipodsekin\Downloads\Layers (5).png

Picture 4. Layouts diagram

|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Purpose | Technologies | Realization |
| Data Access (DAL) | The direct access to the database. It’s the only layer responsible to communicate with the database. | Entity Framework | If some other layer wants to access the database, then this will be done through some of the classes (repositories). These repositories connect to database using special DbContext. The system has only one DAL |
| Business Logic (BLL) | Contains business rules |  | The layer has services – special classes for implementing business rules and access to DAL. |
| API | Exposes WebAPI-enpoints for the RESTful services. | ASP.NET Web API | API has two parts: public users API and staff API.  The same API should be used in Mobile Applications |
| Presentation (PL) | End user interface | Bootstrap, AngularJS | Contains client HTML, JS and CSS code, which communicates with API by REST-protocol. It also has two parts for public and staff users. |

# Components Model

C:\Users\ipodsekin\Downloads\ComponentsView (3).png

Picture 5. Components diagram

The picture describes the subsystems of the application. External systems are marked by yellow color.

There are two main subsystems: HTML-application in web browser and API-service. Web Browser contains Presentation Layer. API-service is the combination of Business Logic Layer, Data Access Layer and Models.

## API-service

API-service divided in two modules: public module and staff module.

## Public module

Public module contains booking and check-in module. It also has several support modules such as authentication (using OAuth-providers: Google, Twitter and LinkedIn), PDF-documents generator etc. This module is connected to Price Service a part of the customer’s booking system.

## Staff module

Staff module provides flight management services. Staff authenticates using AD FS service.

## Scheduler

The scheduler executes periodically at fixed times, dates, or intervals scheduled jobs. The scheduler checks time intervals of check-in start, reservation chart generation etc.

## HTML-application

Provides user interface and hosts within the web browser. It communicates with API-service by REST-protocol. The solution has two separated HTML-applications the first is for public users and the second for the staff.

## SMTP Relay

SMTP Relay is the single point for email sending and enables the email message queue. Generally, SMTP Relay attempts to deliver emails to the Internet within a few seconds of each request. ISP might be unable to deliver the email to the recipient because of a temporary condition such as "mailbox full." In these cases, SMTP Relay attempts to retry the message for a length of time. If the error is permanent, such as "mailbox does not exist," SMTP Relay does not retry the delivery attempt.

# Implementation

C:\Users\ipodsekin\Downloads\LayerInteration.png

Picture 6. Class diagram

Class diagram (Picture 6) shows the example of layers implementation. The diagram contains only one model – Airplane. The logic for other models should be implemented in the same way.

About the classes:

* Models – domain model (Airplane, Airport, Ticket etc.);
* \*Repository – CRUD database operations;
* \*Service – business logic which using \*Repository classes for DB-manipulations. These classes implement Dependency Container design pattern;
* \*Controller – WebAPI controllers;
* \*ViewModel – classes represent the data that you want to send/receive via RESTful service.

There is no the straightway to create \*Service-classes via “new” keyword. It could be realized by using AutoFac (<http://autofac.org/>). The self explanatory code below:

private static void SetAutofacContainer()

{

var builder = new ContainerBuilder();

builder.RegisterControllers(Assembly.GetExecutingAssembly());

builder.RegisterType<UnitOfWork>().As<IUnitOfWork>().InstancePerRequest();

builder.RegisterType<DbFactory>().As<IDbFactory>().InstancePerRequest();

// Repositories

builder.RegisterAssemblyTypes(typeof(AirplaneRepository).Assembly)

.Where(t => t.Name.EndsWith("Repository"))

.AsImplementedInterfaces().InstancePerRequest();

// Services

builder.RegisterAssemblyTypes(typeof(AirplaneService).Assembly)

.Where(t => t.Name.EndsWith("Service"))

.AsImplementedInterfaces().InstancePerRequest();

IContainer container = builder.Build();

DependencyResolver.SetResolver(new AutofacDependencyResolver(container));

}

Presentation layer should be built using Bootstrap and AngularJS libraries. It will allow us to create responsive single page application.

# Deployment Model

Describes the mapping of the software onto the hardware and shows the system's distributed aspects

C:\Users\ipodsekin\Downloads\PhysicalView (11).png

Picture 7. Deployment model diagram

|  |  |  |  |
| --- | --- | --- | --- |
| Zone | Component | Purpose | Software/Hardware |
| Intranet (private cloud) | Firewall / VPN Gateway | Network security system that monitors and controls the incoming and outgoing network traffic. VPN-tunneling | Server: Cisco Adaptive Security Virtual Appliance [link](http://www.cisco.com/c/en/us/products/security/virtual-adaptive-security-appliance-firewall/index.html) |
| Database server | Manage databases | Server: virtual server  Software:   * Operating system: Windows Server * MS SQL Server Enterprise |
| Price service server | Calculates current ticket prices according to request conditions | n/a  (it’s not the part of this booking system) |
| Active Directory Federation Services (AD FS) server | Provides users with single sign-on access to systems and applications located across organizational boundaries. It is connected to AD server | Server: virtual server  Software:   * Operating system: Windows Server * AD FS server role |
| Active Directory (AD) server | Stores information about members of the domain, including devices and users, verifies their credentials and defines their access rights. | Server: virtual server  Software:   * Operating system: Windows Server * AD server role |
| DMZ (Amazon AWS) | Firewall | Network security system that monitors and controls the incoming and outgoing network traffic | Amazon AWS WAF (Web Application Firewall) Service <https://aws.amazon.com/waf/> |
| Load balancer | Automatically distributes incoming application traffic across multiple public web servers | Amazon AWS Elastic Load Balancing Service [https://aws.amazon.com/ elasticloadbalancing](https://aws.amazon.com/elasticloadbalancing) |
| Public web-server 1..N | Distributes content and reverse-proxying requests to API server | Server: Amazon AWS EC2 instance <https://aws.amazon.com/ec2/>  Software:   * Operating system: Ubuntu Server * Web-server and reverse proxy: nginx   (In the future it could be changed to Amazon API Gateway for proxying and Amazon Cloudfront for distributing static content) |
| API server | Exposes RESTful-services for staff and public web clients (mobile applications) | Server: Amazon AWS EC2 instance <https://aws.amazon.com/ec2/>  Software:   * Operating system: Windows Server * Web-server: IIS |
| Staff web server | Distributes content and reverse-proxying requests to API server | Server: Amazon AWS EC2 instance <https://aws.amazon.com/ec2/>  Software:   * Operating system: Ubuntu Server * Web-server and reverse proxy: nginx   (In the future it could be changed to Amazon API Gateway for proxying and Amazon Cloudfront for distributing static content) |
| Active Directory Federation Services (AD FS) proxy server | Brokers a connection between external users and internal AD FS server. It acts as a reverse proxy. | Server: Amazon AWS EC2 instance <https://aws.amazon.com/ec2/>  Software:   * Operating system: Windows Server * AD FS Proxy |
| SMTP relay server | Middle layer between system’s servers and external SMTP-server. It allows change external SMTP-server without configuring all our servers. SMTP Relay have the queue that guarantees the email delivery. | Server: Amazon AWS EC2 instance <https://aws.amazon.com/ec2/>  Software:   * Operating system: Windows Server * IIS with SMTP Relay role   or we can use  Amazon Simple Email Service (Amazon SES) <https://aws.amazon.com/ses> |
| Internet | User computer | It hosts web browser | Any modern browser |
| User mobile device | It hosts booking mobile application | Apple iOS or Android phone or tablet |
| 3D Secure Bank Service | A service for extra security check by asking customers to confirm the payment with a one-time password they receive by SMS. | External Service |
| Google OAuth2 provider | OAuth2-provider for authentication using Google account | External Service |
| Twitter OAuth2 provider | OAuth2-provider for authentication using Twitter Account | External Service |
| LinkedIn OAuth2 provider | OAuth2-provider for authentication using LinkedIn account | External Service |

# Process Models

The system has several processes:

1. Authentication (described below)
2. Booking (described below)
3. Check-In (described below)
4. Cancel booking by user
5. Reservation chart generation
6. Cancel tickets by staff
7. Cancel flight by staff

## Booking Business Process

C:\Users\ipodsekin\Downloads\BookingBP.png

Picture 8. Booking Business Process diagram (part 1).

C:\Users\ipodsekin\Downloads\BookingBP2.png

Picture 9. Booking Business Process diagram (part 2)

## Check-In Business Process

C:\Users\ipodsekin\Downloads\CheckInBP.png

Picture 10. Check-in business process diagram

## Public User Authentication Process

The diagram below shows the attempt to enter to “My bookings” page as unauthorized user and the process of authentication using Google Account.

C:\Users\ipodsekin\Downloads\OAuth2.png

Picture 11. Public user authentication process sequence diagram

# User Interface mockups

The system has two different UIs:

1. Public website user interface;
2. Staff website user interface.

## Public website user interface

C:\Users\ipodsekin\Downloads\Public Sitemap.png

Picture 12. Public website screen flow diagram

### Public front page

C:\Users\ipodsekin\Downloads\Public front.png

Picture 13. Public front page mockup

|  |  |
| --- | --- |
| Page name | Public front page |
| Page description | Main information page |
| Authentication | No |
| Incoming pages | All |
| Outgoing pages | Public login,  Search flight form, My bookings, Check-in |
| Elements | No |

### Public login

C:\Users\ipodsekin\Downloads\Public Login.png

Picture 14. Public login mockup

|  |  |
| --- | --- |
| Page name | Public login |
| Page description | Page contains two forms: authentication using login/password and authentication using social networks |
| Required Authentication | No |
| Incoming pages | All (from menu), Passenger’s info and payment, My bookings |
| Outgoing pages | Public front page, Passenger’s info and payment, My bookings |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | | Login | Textbox | string | Yes | 255 |  | | Password | Password | string | Yes | 255 |  | | Login | Button |  |  |  |  | | Google | Image Link |  |  |  | Performs Google OAuth 2 authentication | | Twitter | Image Link |  |  |  | Performs Twitter OAuth 2 authentication | | LinkedIn | Image Link |  |  |  | Performs LinkedIn OAuth 2 authentication | |

### Search flight form

C:\Users\ipodsekin\Downloads\Search Form.png

Picture 15. Search flight form mockup

|  |  |
| --- | --- |
| Page name | Search flight form |
| Page description | It allows public users to search and book flight |
| Required Authentication | No |
| Incoming pages | All (from menu) |
| Outgoing pages | Search flight result |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Req | LeN | Description | | From | Textbox with autocomplete | int | Yes | 255 | Value: airport.id. Format: “<Airport.Name> (<Airport.Code>)” | | To | Textbox with autocomplete | int | Yes | 255 | Value: airport.id  Format: “<Airport.Name> (<Airport.Code>)” | | Return/ One-way | Group of radio buttons | bool | Yes |  | RETURN-option enables ENDDATE | | startdate | Date picker | date | Yes |  | Range [Today, Today+(1 Year)] | | ENDdate | Date picker | date | No\* |  | \* Required if RETURN-option selected. Range [STARTDATE, STARTDATE +(1 Year)] | | Adults | Combobox | int | Yes |  | Range [1, 5] | | children | Combobox | int | Yes |  | Range [0, 5] | | Infants | Combobox | int | Yes |  | Range [0, 5] | | search | Button |  |  |  |  | |

### Search flight result

C:\Users\ipodsekin\Downloads\SearchFormResults.png

Picture 16. Search result mockup

|  |  |
| --- | --- |
| Page name | Search flight result |
| Page description | The page shows the list of flight options with prices, enables to select the option and shows a summary of the selected flight and total price |
| Required Authentication | No |
| Incoming pages | Search flight form |
| Outgoing pages | Passenger’s info and payment, Public login |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Passenger’s info and payment

C:\Users\ipodsekin\Downloads\Passengers Info (1).png

Picture 17. Passenger’s info and payment mockup

|  |  |
| --- | --- |
| Page name | Passenger’s info and payment |
| Page description | Shows form fields for passengers data to fill in, flight information and payment details |
| Required Authentication | Yes |
| Incoming pages | Search flight result |
| Outgoing pages | 3D Secure, Payment confirmation |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Payment confirmation

C:\Users\ipodsekin\Downloads\PaymentConfirmation (1).png

Picture 18. Payment confirmation

|  |  |
| --- | --- |
| Page name | Payment confirmation |
| Page description | Shows that the payment is finished successfully and enables to download PDF tickets and go to “My bookings” page |
| Required Authentication | Yes |
| Incoming pages | 3D Secure, Passenger’s info and payment |
| Outgoing pages | My bookings |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### My bookings

C:\Users\ipodsekin\Downloads\MyBookings.png

Picture 19. My bookings mockup

|  |  |
| --- | --- |
| Page name | My bookings |
| Page description | Shows booking list |
| Required Authentication | Yes |
| Incoming pages | All (from menu), Payment confirmation |
| Outgoing pages | Booking details |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Booking details

C:\Users\ipodsekin\Downloads\BookingDetails.png

Picture 20. Booking details mockup

|  |  |
| --- | --- |
| Page name | Booking details |
| Page description | Shows the details of a certain booking: PNR, status, flight details, passengers details and enables to print, save as PDF, cancel and perform check-in |
| Required Authentication | Yes |
| Incoming pages | My bookings |
| Outgoing pages | Cancel booking, Check-in, My bookings |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Cancel booking

C:\Users\ipodsekin\Downloads\CancelBooking.png

Picture 21. Cancel booking mockup

|  |  |
| --- | --- |
| Page name | Cancel booking |
| Page description | Enables user either to cancel booking or return to the booking details |
| Required Authentication | Yes |
| Incoming pages | Booking details |
| Outgoing pages | Booking details, Cancel booking confirmation |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Cancel booking confirmation

C:\Users\ipodsekin\Downloads\CancelConfirmation.png

Picture 22. Cancel booking confirmation mockup

|  |  |
| --- | --- |
| Page name | Cancel booking confirmation |
| Page description | Shows the cancelation is finished successfully and enables to go to “My bookings” page |
| Required Authentication | Yes |
| Incoming pages | Cancel booking |
| Outgoing pages | My bookings |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Check-in

C:\Users\ipodsekin\Downloads\Check-In.png

Picture 23. Check-in mockup

|  |  |
| --- | --- |
| Page name | Check-in |
| Page description | Shows a check-in form and enables to enter passengers last name and PNR or electronic ticket number |
| Required Authentication | No |
| Incoming pages | All (from menu), Booking details |
| Outgoing pages | Select seat |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Select seat

C:\Users\ipodsekin\Downloads\SelectSeat (1).png

Picture 24. Select seats mockup

|  |  |
| --- | --- |
| Page name | Select seat |
| Page description | Shows the layout pf the airplane and enables to choose a seat and continue with check-in process |
| Required Authentication | No |
| Incoming pages | Check-in |
| Outgoing pages | Check-in confirmation |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Check-in confirmation

C:\Users\ipodsekin\Downloads\CheckInConfirmation.png

Picture 25. Check-in confirmation mockup

|  |  |
| --- | --- |
| Page name | Check-in confirmation |
| Page description | Informs that check-in process is successfully finalized and enables to print a boarding pass and/or download boarding pass as PDF-document |
| Required Authentication | No |
| Incoming pages | Select seat |
| Outgoing pages | Public front page |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

## Staff website user interface

C:\Users\ipodsekin\Downloads\Stuff Sitemap (4).png

Picture 26. Staff website screen flow diagram

### Staff login

C:\Users\ipodsekin\Downloads\StuffLogin.png

Picture 27. Staff login mockup

|  |  |
| --- | --- |
| Page name | Staff login |
| Page description | Shows staff login form and enables login |
| Required Authentication | No |
| Incoming pages | n/a |
| Outgoing pages | Search flight |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Search flight

C:\Users\ipodsekin\Downloads\SearchFlight.png

Picture 28. Search flight mockup

|  |  |
| --- | --- |
| Page name | Search flight |
| Page description | Shows a search flight form |
| Required Authentication | Yes |
| Incoming pages | Staff login, Cancel ticket confirmation |
| Outgoing pages | Flight details |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Flight details

C:\Users\ipodsekin\Downloads\FlightDetails.png

Picture 29. Flight details mockup

|  |  |
| --- | --- |
| Page name | Flight details |
| Page description | Shows the flight details: departure and destination, date, time, passenger details, statuses and seat numbers. Enables to cancel tickets or whole flight |
| Required Authentication | Yes |
| Incoming pages | Search flight |
| Outgoing pages | Cancel ticket confirmation |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

### Cancel ticket confirmation

C:\Users\ipodsekin\Downloads\CancelTicketCancelation.png

Picture 30. Cancel ticket confirmation

|  |  |
| --- | --- |
| Page name | Cancel ticket confirmation |
| Page description | Shows the successful cancelation data: passenger details, statuses and seats. Enables to go back to flight search |
| Required Authentication | Yes |
| Incoming pages | Flight details |
| Outgoing pages | Search flight |
| Elements | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Element | Type | Required | Length | Description | |  |  |  |  |  |  | |

1. Entity has no management UI in the Air Booking and should be populated by external system. This process out of the scope of the current document [↑](#footnote-ref-1)