

Login

Use case	Login
Summary	Visitor log ins into the system
Actor	Visitor
Precondition	The customer already has an account registered
Postcondition	The customer is logged in the system
Main scenarios	1. The visitor provides username and password 2. The system validates the customer's username and password
Alternative scenarios	2a The systems fails to validate credentials 1. The system signals an error and prompts for new credentials 2. The use case proceeds from (1)

Register

Use case	Register
Summary	The visitor signs up into library system
Actor	Visitor
Precondition	Visitor has a unique username, password, passport/id numbers and expiry date.
Postcondition	The visitor's account was created, and visitor can log into the system as a customer
Main scenarios	1. Visitors provides his/her information as a CPR, password, name, address and credit card information 2. System validates data 3. Visitor's account was created
Alternative scenarios	At any time during step 1 visitor cancels 1. The use case ends 2a The account already exists 1. System informs visitor, that system already exists 2b visitor's information are invalid 1. System informs visitor, that his/her data are not valid

Book flight

Use case	Book flight
-----------------	-------------

Summary	The customer can book a flight
Actor	Customer
Precondition	Customer must be logged into their account and the flight must have available seats
Postcondition	The chosen flight is booked
Main scenarios	<ol style="list-style-type: none"> 1. The customer chooses the flight that they want to book. 2. The system shows details regarding the flight. 3. The customer books seats. 4. The customer selects what type of luggage they want to bring. 5. The customer purchases the ticket. 6. The system generates an unique code for the ticket purchase.
Alternative scenarios	<p>At any time during step 1 customer cancels</p> <ol style="list-style-type: none"> 1. The use case ends <p>3a The customer does not want to reserve desired seats</p> <ol style="list-style-type: none"> 1. System chooses a random seat for the customer.

View available flights

Use case	View available flights
Summary	View flights available for customer
Actor	Customer
Precondition	Flights must be added by operator into the system and dates of the flights are in the future
Postcondition	Flights are viewed to the customer
Main scenario	<ol style="list-style-type: none"> 1. Customer inputs departure from 2. Customer inputs flight destination 3. Customer inputs departure date 4. Customer inputs 5. System shows a list of all available flights with those constrains
Alternate Scenario	If there are no data stored, continue from 1.

	If the system cannot find item with specific attribute, system will show an error message.
--	--

Cancel booked flights

Use case	Cancel booked flights
Summary	Customer cancels their booked flights
Actor	Customer
Precondition	Flight must be booked by the customer
Postcondition	Flight is no longer booked for the specific customer
Main scenario	<ol style="list-style-type: none"> 1. Customer views booked flights 2. Customer selects the booked flight to be cancelled 3. Customer cancels the flight 4. System shows confirmation prompt 5. Customer will confirm that the customer wants to cancel that particular booked flight 6. System will notify user that the flight is no longer booked 7. System will delete booked code, generated for customer
Alternate Scenario	*a At any time during step 1,2,3 & 4 Customer cancels 1. The use case ends

Edit user info

Use case	Edit user info
Summary	The customer manages their account
Actor	Customer
Precondition	Customer must have account
Postcondition	Changes made by customer are saved
Main scenario	<ol style="list-style-type: none"> 1. The customer chooses to change the value for: <ol style="list-style-type: none"> a. Full name b. Password c. Address

	d. Date of birth e. Passport/Id number f. Passport/Id expiration date g. Nationality h. Gender i. Phone number j. Email address k. Credit card info 2. System will confirm that info has been edited
Alternate Scenario	1a The entered value is the same as previous value 1.No change has been made

View booked flights

Use case	View booked flights
Summary	The customer views booked flights
Actor	Customer
Precondition	Flight must be booked by customer in Book flights and the flight did not happen yet
Postcondition	Flights are shown to a customer
Main scenarios	1. Customer open booked flights 2. Booked flights are shown
Alternative scenarios	If there are no booked flights, system will show an error

View previous flights

Use case	View previous flights
Summary	The customer views booked flights
Actor	Customer
Precondition	Flight must be booked by customer in Book flights and the flight already happened
Postcondition	Previous flights are shown to a customer
Main scenarios	1. Customer open booked flights 2. Previous flights are shown
Alternative scenarios	If there are no previous flights, system will show an error

Create flights

Use case	Create flights
Summary	The operator creates a new flight
Actor	Operator
Precondition	Operator must have all details about new flight
Postcondition	The flight is created
Main scenarios	<ol style="list-style-type: none">1. The operator chooses to create new flight2. The operator needs to insert information about flight3. The system validates data of the new flight4. The flight is created and stored in the database
Alternative scenarios	<ol style="list-style-type: none">2a The flight information is invalid<ol style="list-style-type: none">1. System informs operator, that information is invalid

Cancel flights

Use case	Cancel flight
Summary	The operator cancel flight
Actor	Operator
Precondition	Flight must be created in database
Postcondition	The flight is cancelled
Main scenarios	<ol style="list-style-type: none">1. The operator chooses an upcoming flight to cancel2. Flight is cancelled
Alternative scenarios	If flight already happened, system will show an error

View information about customers

Use case	View information about customers
Summary	The operator views information about customers
Actor	Operator
Precondition	Database needs at least one registered customer
Postcondition	The information is showed
Main scenarios	<ol style="list-style-type: none">1. Operator filters customers by attribute he demands2. Operator chooses a customer3. Full information about a customer will show
Alternative scenarios	<p>If there are no data stored, continue from 1.</p> <p>If the system cannot find item with specific attribute, continue from 1.</p>

Edit flight information

Use case	Edit flight information
Summary	The operator edits information about flight
Actor	Operator
Precondition	Flight must be created in database
Postcondition	The flight is updated
Main scenarios	<ol style="list-style-type: none"> 1. The operator chooses an upcoming flight to edit 2. Operator edits the information 3. The system validates the new data of the flight 4. The flight is updated and customers will see the new data
Alternative scenarios	<p>*a At any time during step 1 operator cancels</p> <ol style="list-style-type: none"> 1. The use case ends <p>2a The flight information is invalid</p> <ol style="list-style-type: none"> 2. System informs operator, that information is invalid <p>2b Entered value is the same as previous value</p> <ol style="list-style-type: none"> 1. No change has been made

Delay flights

Use case	Delay flights
Summary	The operator delays a chosen flight
Actor	Operator
Precondition	There is at least one created flight in database
Postcondition	The departure and arrival time for the flight is updated, and notification is sent to customers who booked this flight
Main scenarios	<ol style="list-style-type: none"> 1. The operator chooses a flight to delay 2. A message asking to confirm the delay 2. The flight is delayed 3. Customers get notification about delay
Alternative scenarios	<p>At any time during step 1 and 2 operator cancels</p> <ol style="list-style-type: none"> 1. The use case ends