FlightPub
Requirements Model Document

Team No: 4

Team Members:

Tobias Colson

Samuel Brackenrig

Adam Fleming

Benjamin Collins

Jordan Cork

Document Change Control

Version	Date	Authors	Summary of Changes
0.8	04/05/18	Tobias Colson	Collating use cases
0.9	05/05/18	Tobias Colson	Inserting more detail
1.0	06/05/18	Tobias Colson	Formatting, finalisation
2.0	6/06/18	Jordan Cork	Make Recommended Changes, Reformat, Incorporate User Categories

Document Sign-Off

Name (Position)	Signature	Date
Benjamin Collins	~	06/05/18
Tobias Colson	~	06/05/18
Adam Fleming	~	06/05/18
Jordan Cork	~	06/05/18
Samuel Brackenrig	~	06/05/18

Please note: This document is based on Requirement Model of Process Mentor Uni-Sep 3.0.

Contents

1 Introduction

This report contains the Use Case specification for the FlightPub online flight booking system.

This system aims to provide a fully featured web-application for the booking of flights. It will implement a digital storefront allowing users to effectively search and select flights that fit their requirements. Upon selection of their chosen flight, the user is then able to complete a booking process, all within the application. A review system will allow users to provide feedback on both the application and flights themselves. The system will also include a user account facility to allow users to create and manage a profile, providing a more tailored experience should they choose to use it.

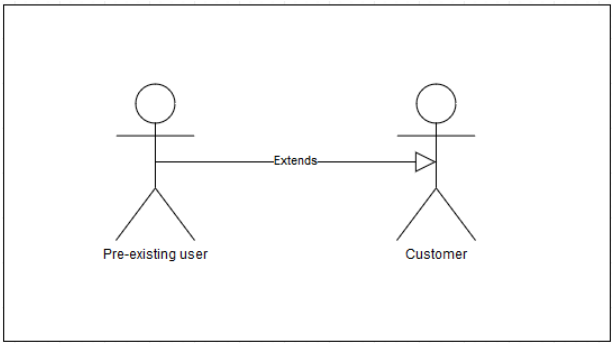
This tailored expierence, which can also be accessed by non logged in customers via the Assisted Search, is based on a User Category selected by the user. There are 6 categories, Standard, High Class, Leisure, Business, Budget, and Family, each with their own unique recommendation strategy. This strategy is a priority based ordering system of the search results. Each category has a specific priority ordering of recommendation parameters which tailors the order of the search results. If no User Category has been selected then the default is the 'Standard' Category.

This report should be used as a basis for the implementation of the system and is to be used to ensure project goals are met.

2 Actors

2.1 Actor Diagram

The main actor in the system is the customer. This customer may exist as a new user or as a pre-existing user. The pre-existing user extends the customer role, able to use the same features as well as account specific ones.



2.2 Actor Definitions

1.2.1 A01 <Customer>

Description	*The person using the system to search, select and book flights. This is the primary actor in all use cases, as the system is designed to provide a service specifically for them.*
Inherits	*None*
Actor Type	*Active - Person*

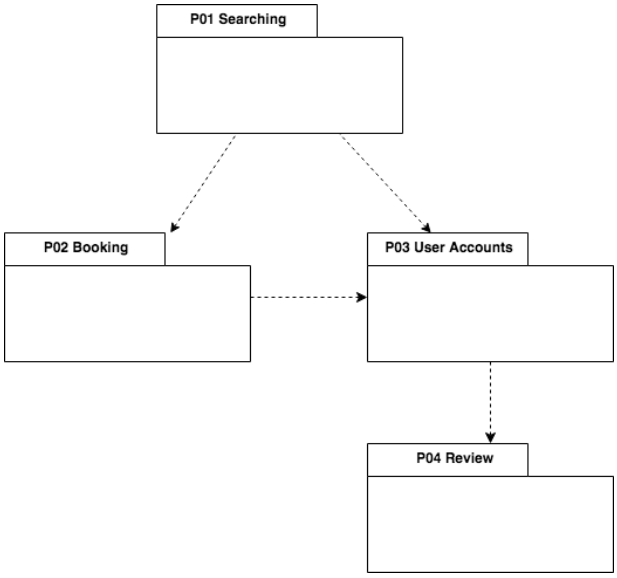
1.2.2 A02 <Pre-existing Customer>

Description	*A user with an existing account. Has access to services that require a user account, such as flight history.*
Inherits	*Customer*
Actor Type	*Active - Person*

1.2.3 A03 <External Payment Services>

Description	*An external payment processing service. Handles the user's payment in the final phase of booking*
Inherits	*None*
Actor Type	*Passive - System*

3 Use Case Model



2.1 P001 <Searching>

This package involves the customer searching for flights using either the specific or assisted search. Then the customer will be displayed with their filtered results which they are able to sort the flight list by price, stop overs, etc. If the customer is not happy with these results they are to do a new search and return to the start again. Pre-existing users are able to search for a similar flight that they booked before.

The initial order of search results presented to a user is based off of the User Category selected. This category can be selected as part of the Assisted Search parameters or defined in the personal information of logged in users. This ordering is achieved through a priority based system. Each category has a specific priority ordering of recommendation parameters which tailors the order of the search results. If no User Category has been selected then the default is the 'Standard' Category.

<u>Recommendation Parameters</u>

Cost: Flights with lowest price

Flight Rating: Highest rated flights

No. Stop Overs: Flights with the least stopovers

<u>User Categories</u>

Recommendation Parameters shown in order of highest to lowest priority.

Standard

1. Cost
2. Flight Rating
3. No. Stop Overs

High Class

1. Flight Rating
2. No. Stop Overs
3. Cost

Business

1. No. Stop Overs
2. Flight Rating
3. Cost

Leisure

1. Flight Rating
2. Cost
3. No. Stop Overs

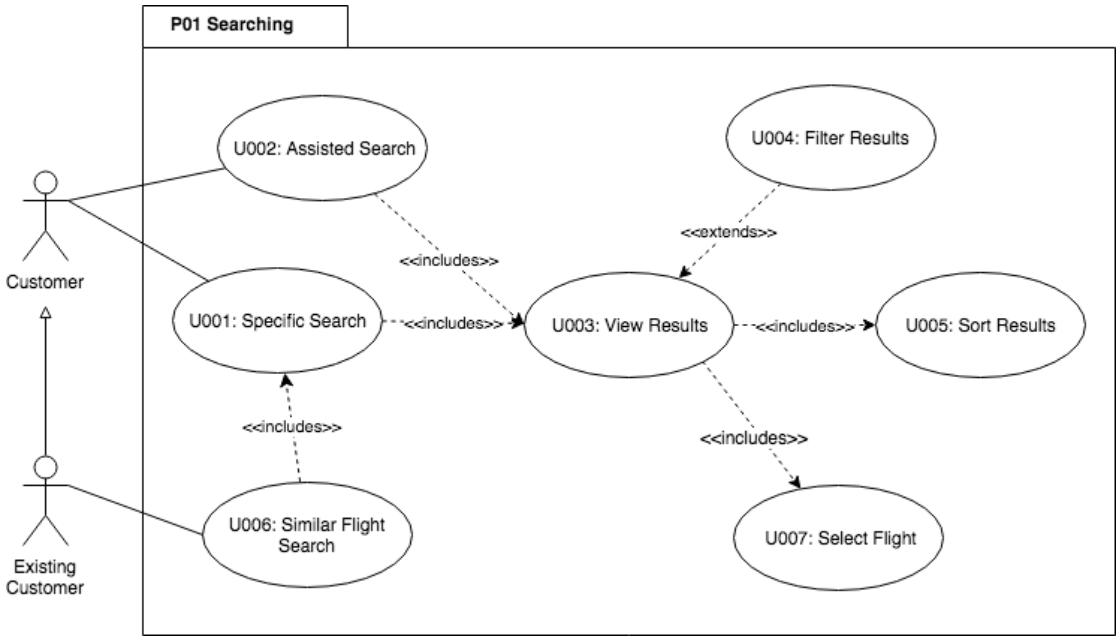
Budget

1. Cost
2. No. Stop Overs
3. Flight Rating

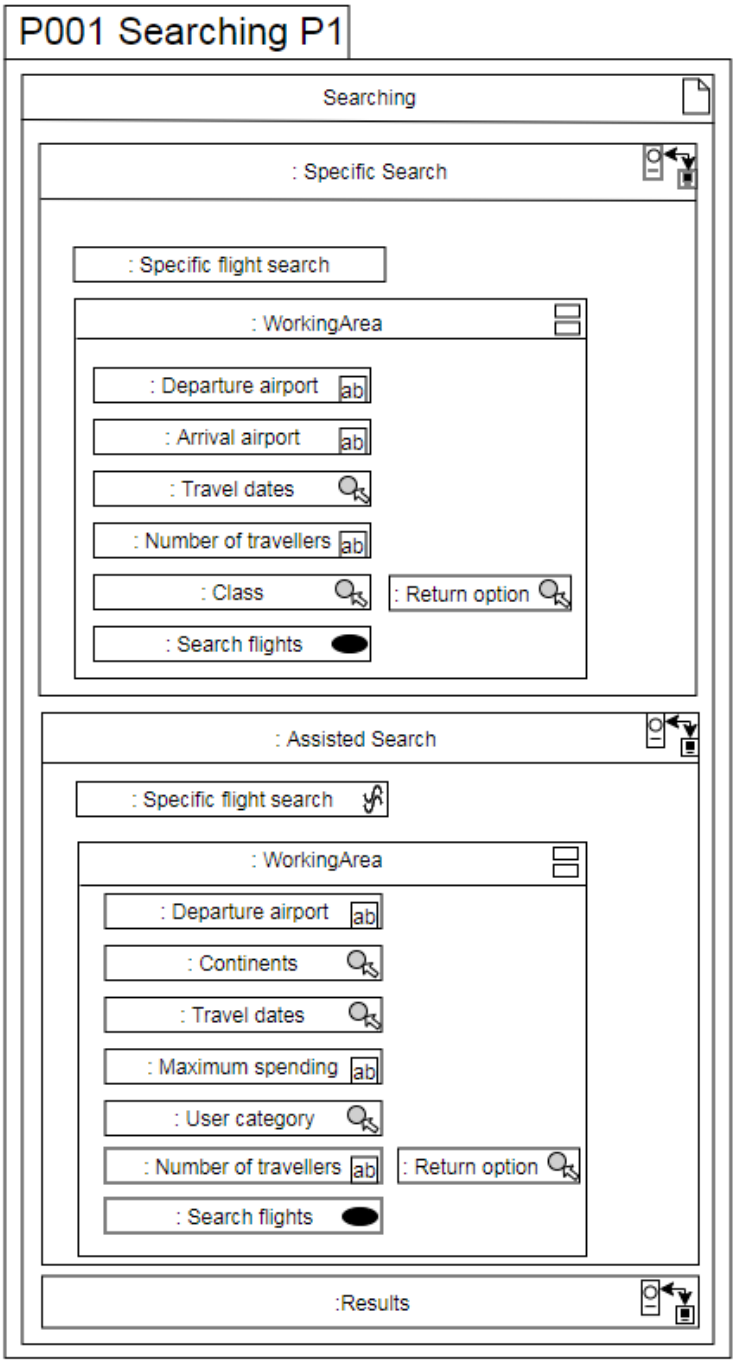
Family

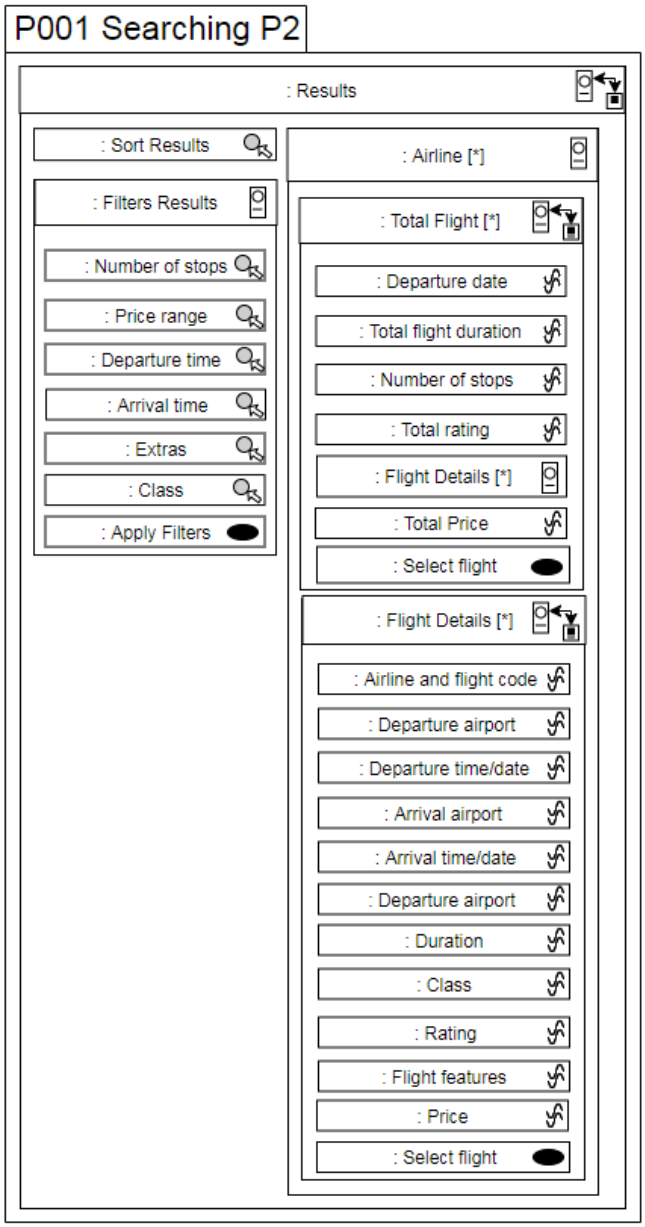
1. No. Stop Overs
2. Cost
3. Flight Rating

Use Case Diagram

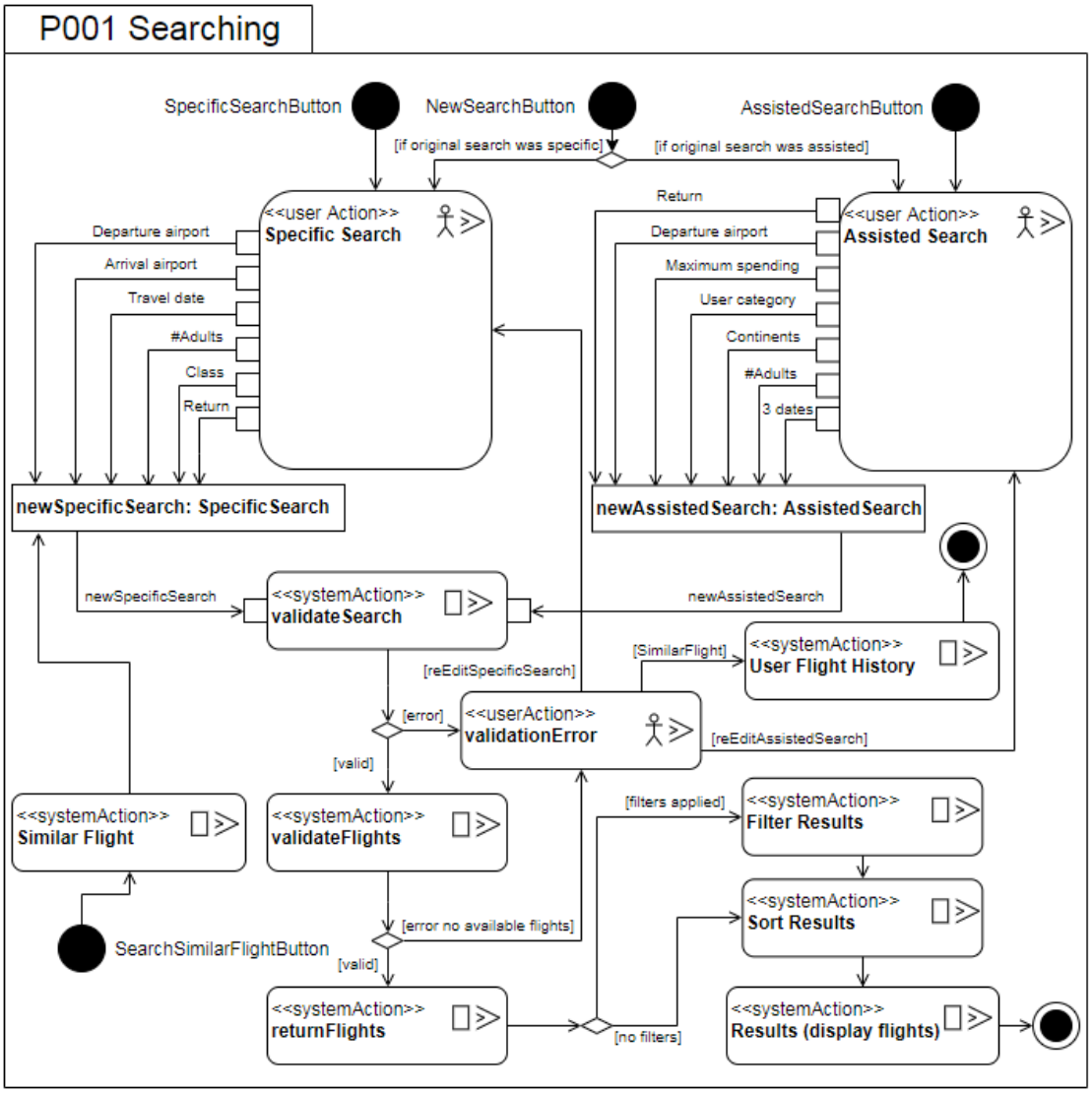


Presentation Model




P001 Searching P2
: Results
: Sort Results
: Filters Results
: Number of stops
: Price range
: Departure time
: Arrival time
: Extras
: Class
: Apply Filters
: Airline [*]
: Total Flight [*]
: Departure date
: Total flight duration
: Number of stops
: Total rating
: Flight Details [*]
: Total Price
: Select flight
: Flight Details [*]
: Airline and flight code
: Departure airport
: Departure time/date
: Arrival airport
: Arrival time/date
: Departure airport
: Duration
: Class
: Rating
: Flight features
: Price
: Select flight

Process Model



2.1.1 P001-U001 <Specific Search>

Description:

The customer wants to search for their desired flight based on departing/arriving airports, class, number of travellers, and select travel dates.

Actors:

Customer

Preconditions:

Selected specific search for flights

Main Flows:

1. Customer fills out the following inputs
   1.1 Departure airport (text) (Required)
   1.2 Arrival airport (text) (Required)
   1.3 Travel dates (select dates on a calendar) (Required)

1.4 Number of travellers (text) (Required)
1.5 Class (select from a list) (Optional)
1.6 Return option (Required)(radio button)

2. Customer clicks the Search flights button
3. System checks if the user validates user input
   3.1 Alternative 1: *Input is invalid*

Alternative Flows:

1. Input is invalid
   1.1 The system will prompt the user to provide correct input with error message

Post-conditions:

Main Flow: Once the system check is considered all clear the customer is redirected to the Filter Results page. (Filter Results use case)

Alternative 1 Flow: Customer will remain on the Specific search page and will have to put valid inputs in each input field they failed

Extends:

Results

Notes:

None

Presentation Model and Process Model:

None

Author:

Adam Fleming

2.1.2 P001-U002 <Assisted Search>

Description:

The customer wants to search for a flight where they don't really have an idea on where to go, but they are able to give some information.

Actors:

Customer

Preconditions:

1. Selected assisted search for flights

Main Flows:

1. Customer fills out the following inputs
   1.1. User category (Standard, High Class, Leisure, Business, Budget, Family) (select from a list) (Default: Standared)
   1.2. Continents they might want to visit (Drop down list) (Default: Oceania) (Required)
   1.3. Maximum spending on a single flight (text) (Optional)

1.4. Departure Date (Required) (Default: current date)

1.5. Number of travellers (Drop Down selection, Default: 1) (Required)

1.6. Return Or One Way Option (Required)(radio button)

1.7. Departure airport (Required)

2. Customer clicks the Search flights button
3. System checks if the user put valid and required inputs in

Alternative 1: *Customer puts in invalid inputs and/or no inputs*

Alternative Flows:

1. Customer puts in invalid inputs and/or no inputs

1.1. The system will prompt the user to put a input in or a valid input for each input field that they failed

Post-conditions:

Main Flow: Once the Search flights button is clicked, customer is redirected to the Filter Results page. (Filter Results use case)

Alternative 1 Flow: Customer will remain on the Assisted search page and will have to put valid inputs in each input field they failed

Extends:

View Results

Notes:

None

Author:

Adam Fleming

Presentation Model and Process Model:

None

2.1.3 P001-U003 <View Results>

Description:

The customer will be presented with a list of flights that are specific to their inputs in the specific/assisted search function. The flights will be initilally ordered based on the User Category system

Actors:

Customer

Preconditions:

Customer has put in correct inputs in specific/assisted search and clicked the search flights button

Main Flows:

1. The system pulls a list of flights matching the specific search parameters from the database.

Alternative 1: *If there are no available flights*

Alternative 2: *If the customer did assisted search*

2. System sorts flights list by default User Category ordering i.e. Standard Category

Altenative 3: *Customer is logged in to a User account*

3. System displays top 50 sorted flights from the list of flights

Alternative Flows:

1. If there are no available flights
    1.1. The system will prompt the customer that we are sorry for the inconvenience
2. If the customer did an assisted search
    2.1. System sorts flights list by User Category ordering selected in Assisted Search parameters
    2.2. System displays top 50 sorted flights from the list of flights
3. Customer is logged into a User account
    3.1. System sorts flight list by the ordering specified by the User Category selected in Users account information
    3.2. System displays top 50 sorted flights from the list of flights

Post-conditions:

Main Flow: Top 50 flights based on Standard ordering will be displayed to the customer

Alternative 1 Flow: The customer will be redirected to the specific/assisted search page where they can search for another flight

Alternative 2 Flow: Top 50 flights based on selcted User Category (in Assisted Search) ordering will be displayed to the customer

Alternative 3 Flow: Top 50 flights based on selcted User Category (in User information) ordering will be displayed to the customer

Extends:

None

Notes:

None

Author:

Adam Fleming, Jordan Cork

Presentation Model and Process Model:

None

2.1.4 P001-U004 <Filter Results>

Description:

The customer can select from a list to filter out certain flights they don't want to see. These options will include number of stops, price range, take off/arrival time, extras, and class.

Actors:

Customer

Preconditions:

1. Customer is on the Result Page

Main Flows:

1. Customer can fill out the follow inputs:
    1.1. Number of stops (range slider)
    1.2. Price range (range slider)
    1.3. Take off time (range slider)
    1.4. Arrival time (range slider)
    1.5. Extras (Wi-Fi, power, entertainment) (checkboxes)
    1.6. Class (select from a list)
2. Customer clicks the apply filters button
3. The system will return flights that are specific to the customer's filtered requests.

Alternative 1: *If there are no available flights*

4. Flights sorted based on currently selected sorting parameter (see below for sorting)
5. System displays top 50 flights based on ordering above

Alternative Flows:

1. If there are no available flights
    1.1. The system will prompt the customer that we are sorry for the inconvenience

Post-conditions:

Main Flow: The list of flights is now displayed with the filtered flights removed from the list and they are viewing it on the Result page

Alternative 1 Flow: The customer will be redirected to the Results page with their filters removed.

Extends:

Results

Notes:

None

Author:

Adam Fleming

Presentation Model and Process Model:

None

2.1.5 P001-U005 <Sort Results>

Description:

The customer can select from a list to sort the flights in a particular order. The flights can be sorted by price, number of stopovers, take off/arrival time, extra, cabin choices, airway, and User Category ordering.

Actors:

Customer

Preconditions:

Customer is on the Result Page

Main Flows:

1. Customer can click a list of sorting options to sort by these are: price, number of stopovers, take off/arrival time, extra, cabin choices, airway and User Category ordering.
2. Once they click the method they wish to sort, the list will be immediately sorted by that method

Alternative Flows:

None

Post-conditions:

Main Flow: The list of flights is now displayed in the order the customer selected and they are viewing in on the Results page

Extends:

Results

Notes:

None

Author:

Adam Fleming

Presentation Model and Process Model:

None

2.1.6 P001-U006 <Search Similar Flight>

Description:

Pre existing customers are able to view their previous flight selections. They are able to select a 'Search Similar Flights' option along with a departure date. This will carry out a search based on parameters in line with the previous flights parameters i.e. departure / arriving airport, etc.

Actors:

Pre-existing Customer

Preconditions:

The customer is viewing *Previously Booked Flights* page

Main Flows:

1. Customer selects a departure date (Default: current date)
2. Customer clicks the “Search Similar Flights” button
3. System carries out Specific Search (see above) with selected departure date (or default) and previous flight details (Departure Airport, Arriving Airport, Class, Passengers)

Alternative Flows:

None

Post-conditions:

Customer redirected to Search Results page

Extends:

None

Notes:

None

Author:

Samuel Brackenrig, Jordan Cork

Presentation Model and Process Model:

None

2.2.1 P001-U007 <Select Flight(s)>

Description:

The Customer selects a flight from the search results.

Actors:

Customer

Preconditions:

1. The search results are being displayed

Main Flows:

1. Customer selects a flight
   1.1. Customer determines their preferred flight.

Alternative 1: *Customer needs to alter search results*

   1.2. Customer clicks the ‘Select’ button corresponding to the flight
2. System stores the selected flight
3. System displays Return flights (top 50 based on currently selected sorting parameter)

Alternative 2: *One Way is selected*

4. Customer selects a return flight
   1.2. Customer determines their preferred flight

Alternative 3: *Customer needs to alter search results*

Alternative 4: *Customer wants to return to departure flight selection*

1.3. Customer clicks the 'Select' button corresponding to the flight

5. System stores the selected flight
6. Continue booking process proceed to View Selected Flight(s) use case

Alternative Flows:

1. Customer needs to alter search results
    1. Customer selects 'Back to Search'
    2. System returns to Search page
2. One way is selected
    1. Continue booking process proceed to View Selected Flight(s) use case
3. Customer needs to alter search results
    1. Customer selects 'Back to Search'
    2. Selected Departure flight is dropped by the system as the booking process will begin over again
    3. System returns to Search page
4. Customer wants to return to departure flight selection
    1. Customer selects the 'Change Departure Flight'
    2. System returns to Select Flight use case

Post-conditions:

Main Flow: Both flights have been selected and stored and *Customer* is redirected to the View Selected Flight(s) page (*View Selected Flight(s)* use case)

Alternative Flow 1: *Customer* has returned to Search page

Alternative Flow 2: One flight has been selected and stored and *Customer* is redirected to the View Selected Flight(s) page (*View Selected Flight(s)* use case)

Alternative Flow 3: Customer has returned to Search page. The booking process is restarted, and the previously selected departure flight is forgotten by the system

Alternative Flow 4: As for the pre-conditions for current use case. The previously selected departure flight is forgotten by the system and booking process has begun over again. The search results are displaying the prospective departure flights

Extends:

View Selected Flight(s)

Notes:

None

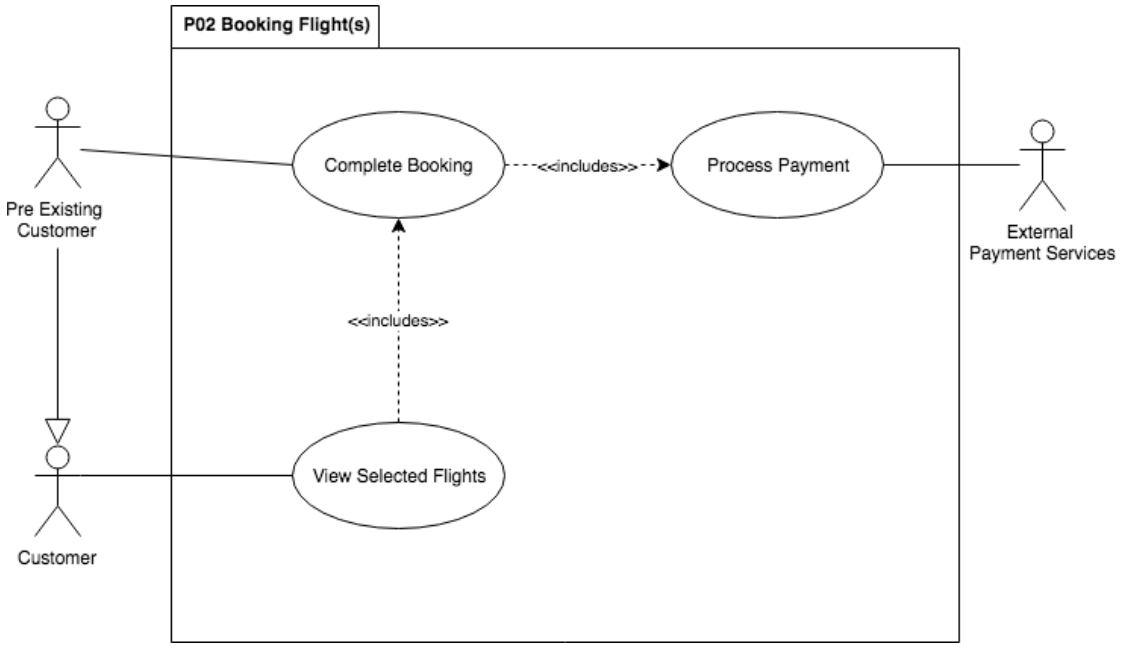
Author:

Jordan Cork

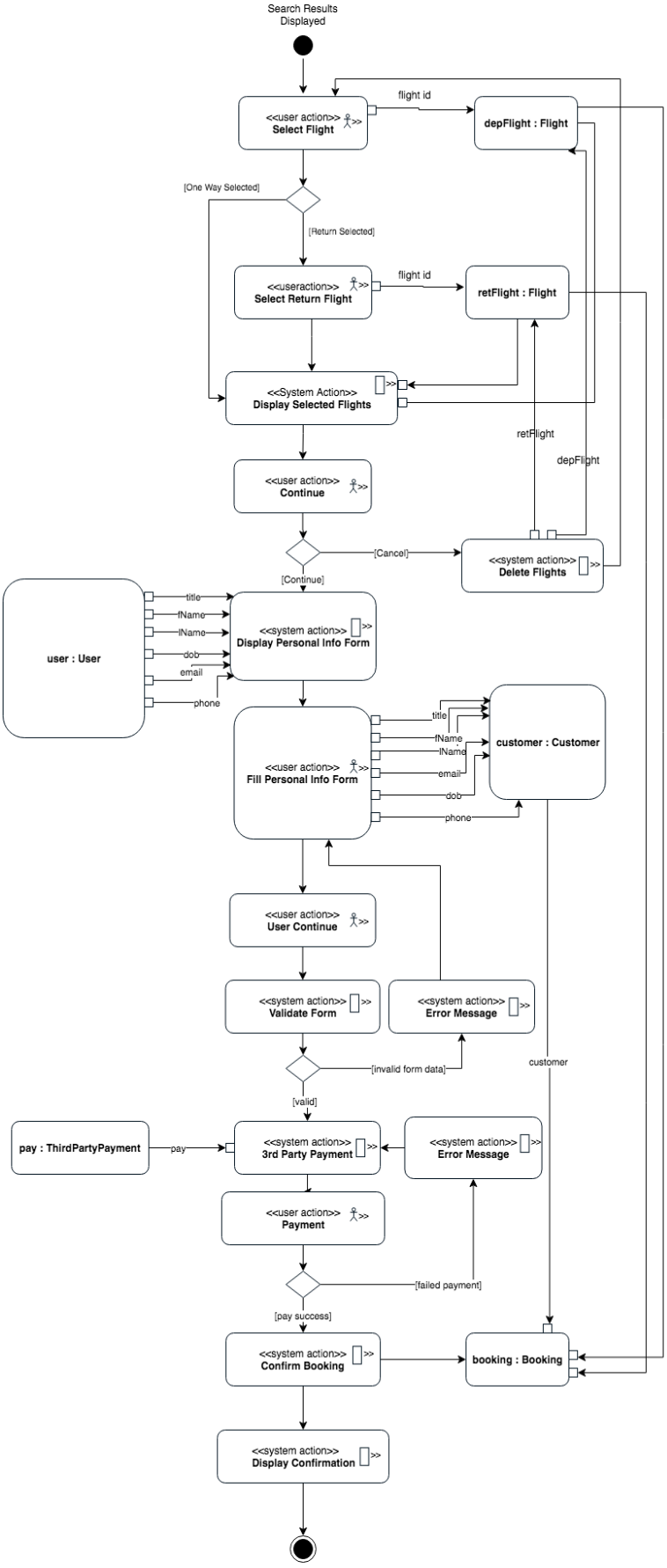
2.2 P002 <Booking>

This package involves the process of booking a one-way flight or return flights. It begins when the *Customer* is viewing their search results and is ready to select a flight and ends once the transaction, including all payments, has been completed or the *Customer* cancels the booking process.

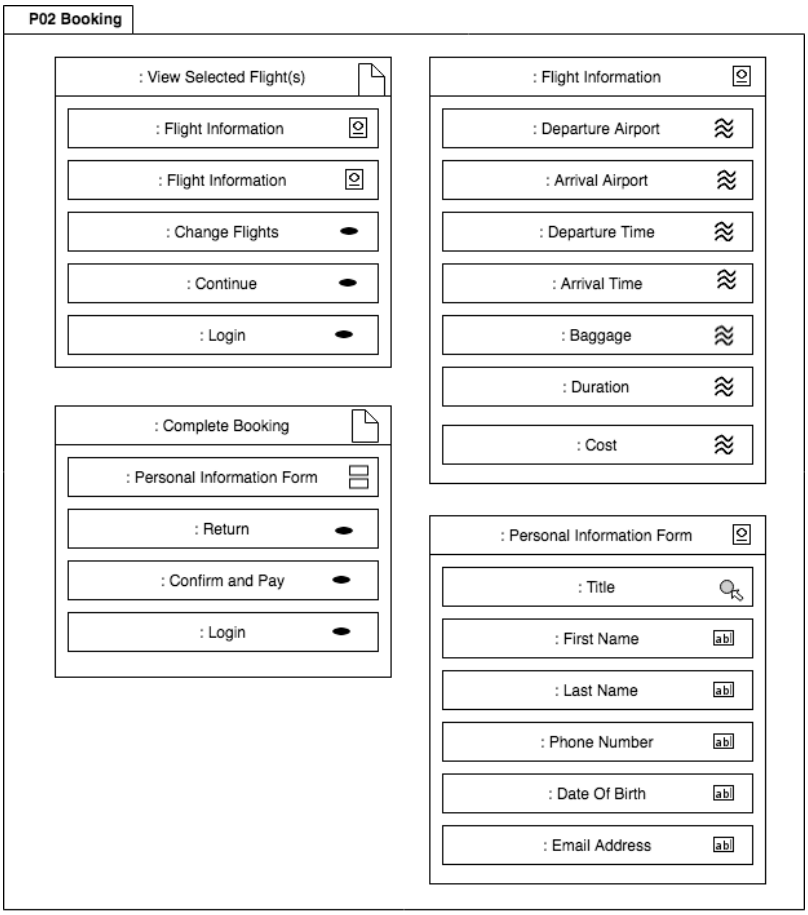
Use Case Diagram



Process Flow Diagram



Presentation Model



2.2.2 P002-U001 <View Selected Flights>

Description:

Relevant information on the selected flight(s) are presented to the Customer for them to review their selection before continuing with the booking process

Actors:

Customer

Preconditions:

1. The flight(s) have been selected by the Customer and stored by the system

Main Flows:

1. The system displays all relevant information for each flight selected
2. Customer is ok with their selection and selects ‘Continue’.

Alternative 1: Customer needs to change their selection

3. System proceeds to Complete Booking use case

Alternative Flows:

1. Customer needs to change selection

1.1. Selects ‘Change Flight(s)’ button
1.2. System removes the previously selected flight(s)
1.3. Returns to original search results
1.4. Proceeds with Select Flight use case

Post-conditions:

Main Flow: The selected flight(s) is stored and system proceeds with booking process, Complete Booking use case.

Alternative Flow 1: As for the pre-conditions for Select Flight use case. The previously selected flight(s) is forgotten by the system and booking process has begun over again. The search results are displaying the prospective departure flights

Extends:

Complete Booking

Notes:

None

Author:

Jordan Cork

2.2.3 P002-U002 <Complete Booking>

Description:

The Customer continues their booking process after confirming their selected flight(s). Involves confirming personal details and confirming booking.

Actors:

Customer

Pre Existing Customer

Preconditions:

1. The flight(s) has/have been selected by the Customer and stored by the system
2. The Customer has confirmed their selection

Main Flows:

1. System displays a form for the Customers personal information as well as brief details on selected flights and costs
2. Customer fills out personal information form. Alternative 1: User is logged in
3. Customer selects ‘Confirm and Pay’ button. Alternative 2: Customer selects ‘Return’ button
4. System proceeds to ‘Process Payment’ use case. Alternative 3: Incorrect personal details.

Alternative Flows:

1. Customer is logged in

1.1. System retrieves user information
1.2. System pre-fills out the personal detail form
1.3. Customer reviews personal information and makes necessary changes
1.4. Return to Main Flow step 3

2. Customer selects 'Return' button
   2.1. System returns to View Selected Flight(s) page
3. Improperly filled out personal information form
   3.1. System displays error message and prompts Customer to re-do form
   3.2. Return to Main Flow step 2

Post-conditions:

Main Flow: Flight(s) and personal information has been confirmed. Now ready for the payment process.

Alternative Flow 2: As for the pre-conditions for Select Flight use case. The previously selected flight(s) is forgotten by the system and booking process has begun over again. The search results are displaying the prospective departure flights.

Extends:

Process Payment

Notes:

None

Author:

Jordan Cork

2.2.4 P002-U003 <Process Payment>

Description:

The Customer continues there booking process after confirming their booking and personal details. This is the final step of the process where the Customer pays for the booking through an external payment service.

Actors:

Customer

External Payment Services

Preconditions:

1. The flight(s) have been selected by the Customer and stored by the system
2. The Customer has confirmed their selection
3. The Customer's personal information has been confirmed

Main Flows:

1. System redirects Customer to External Payment Service.
2. Customer pays through External Payment Service

3. External Payment Service returns a confirmation message to the system.

Alternative 1: Failed payment

4. System receives confirmation of payment
5. System stores booking information (Flight details) in users account
6. System emails Customer with booking information
7. System displays successful booking message
8. Customer selects 'Return to Home' button
9. System displays Home Page

Alternative Flows:

1. Failed Payment
   1.1. System receives a failed payment message from External Payment Service
   1.2. System displays failed payment message
   1.3. Customer selects 'Return to Booking'
   1.4. Returns to Complete Booking page

Post-conditions:

Main Flow: Booking has been confirmed and paid for. An email with all booking information has been sent to the Customer. Flight details have been stored in users account

Alternative Flow 1: As for the pre-conditions for Complete Booking. The flight(s) has/have been selected, stored in the system and confirmed

Extends:

None

Notes:

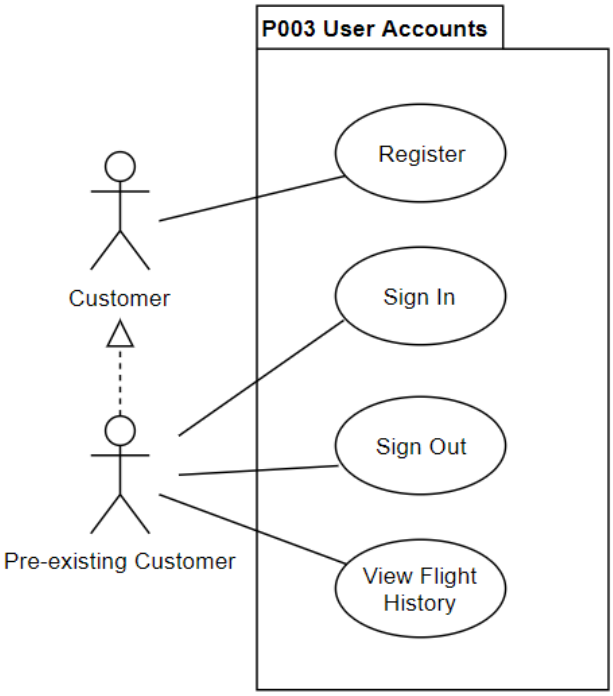
None

Authors:

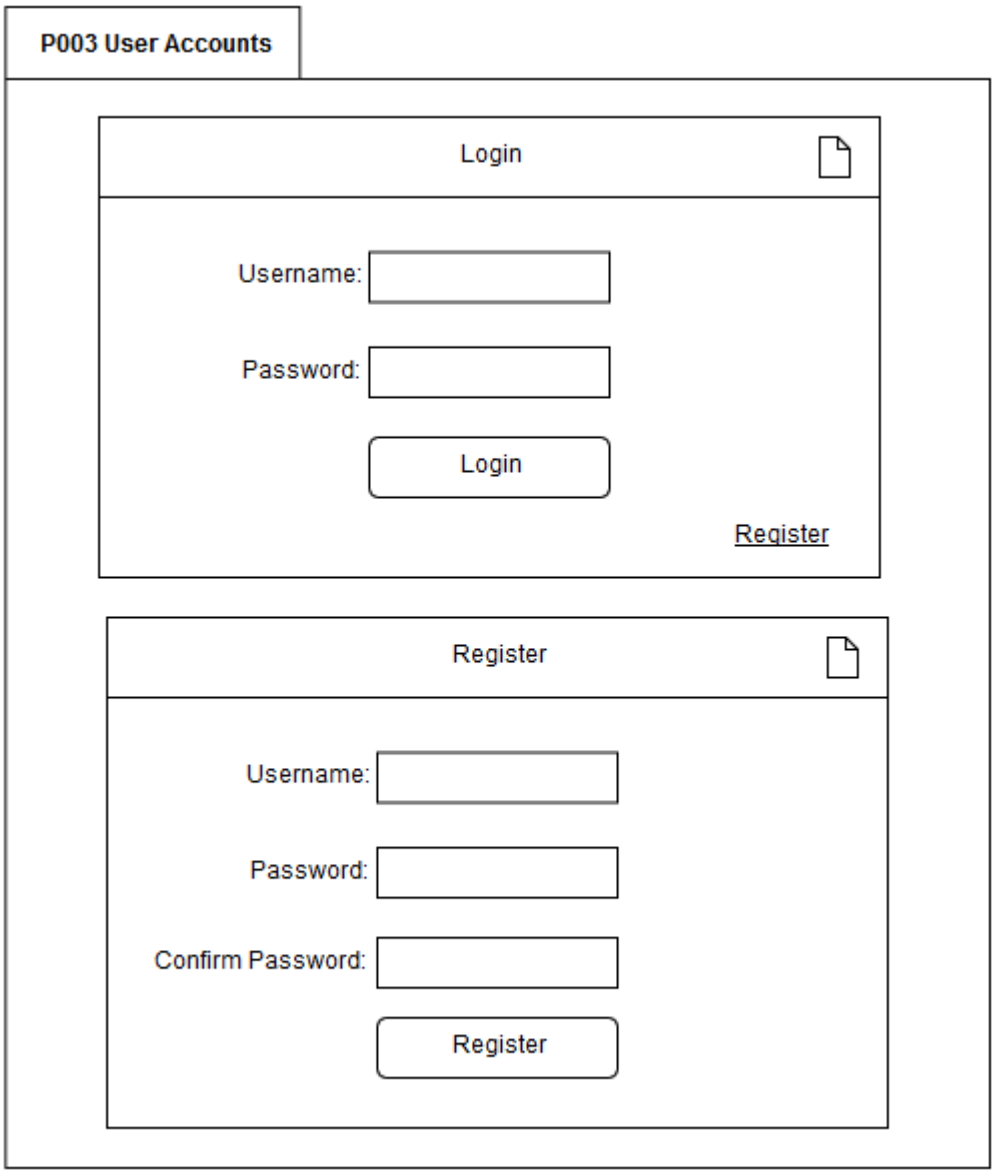
Jordan Cork

2.3 P003 <User Accounts>

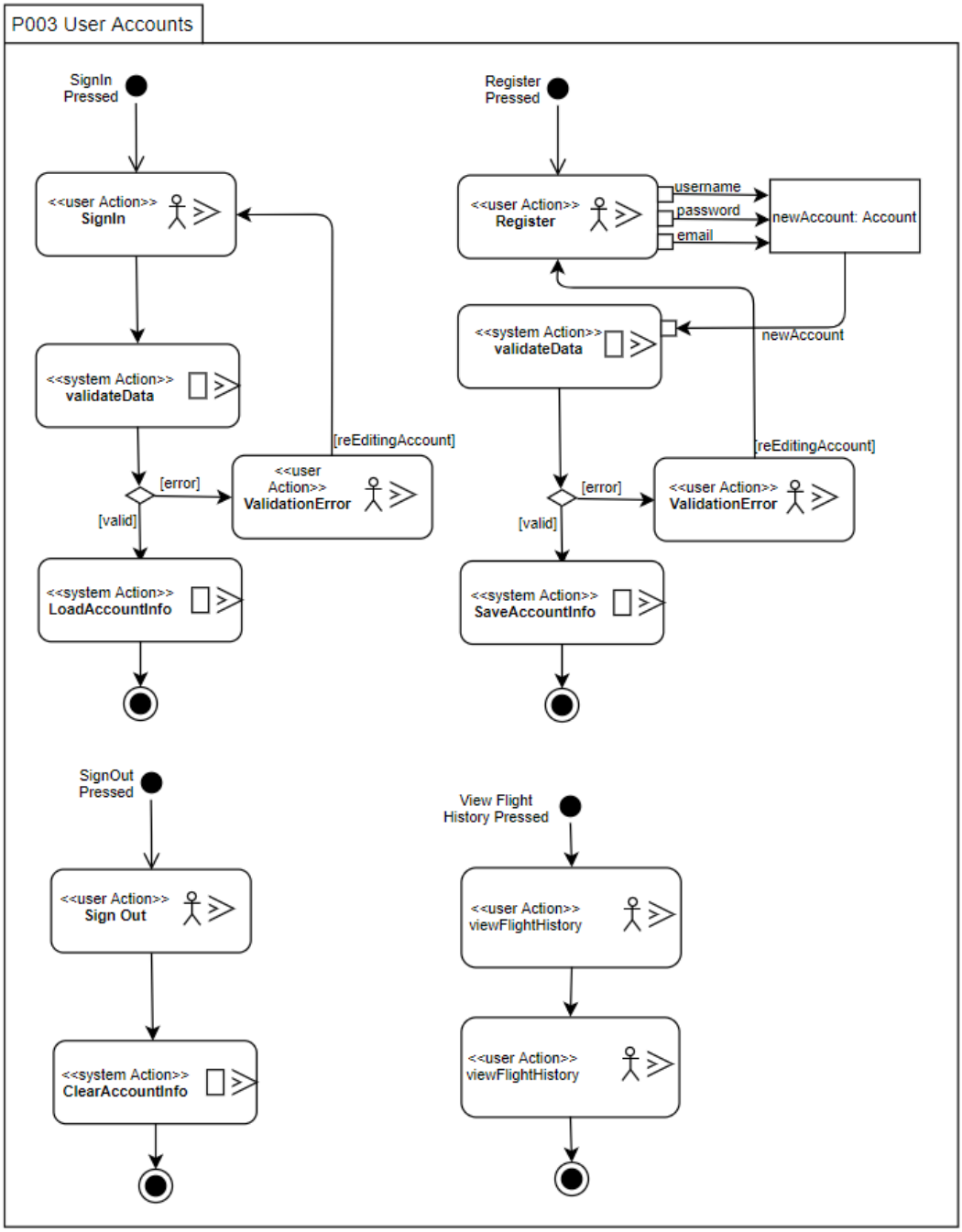
This package contains the features relevant to the user's account and their personal data which is to be stored in the system. This package handles User Maintenance (Register, Sign in, Sign out) as well as allowing the user to view their flight history.



Presentation Model



Process Flow Model



2.3.1 P003-U001 <Sign in>

Description:

Customer will be prompted with a portal to sign in using their pre-existing username and password. This customer account will store the email address of the customer, full name and flight history.

Actors:

Pre-existing Customer

Preconditions:

No customer is currently logged in on the host's session

Main Flows:

1. Customer will click the "Sign In" button on the home page or on the persistent menu bar
2. Customer enters their username and password
3. System checks customers entered details
    3.1. System sanitizes the input
    3.2. Query database for matching username and password
    3.3. Return success message to customer
4. Customer is returned to the previous page they were on

Alternative Flows:

1. Customer inputs an incorrect username and/or password and is prompted by the system to enter a correct username and password.

    1.1. System also prompts user with 'Forgetten Password?' option

        1.1.1. User selects 'Forgotten Password?'

        1.1.2. System emails user email address with password recorvery email

    1.2. After 3 unsuccessful attempts, system locks account and sends an email to user email address where the user can unlock the account. System also displays error message and instructions for unlocking account.

2. Customer inputs invalid or malicious text into the inputs. The system sanitizes all input and if an error occurs the user is notified and allowed to enter appropriate inputs

Post-conditions:

Customer is logged into the system and may now book flights. If input invalid, customer is prompted to enter new input.

Extends:

None

Notes:

None

Authors:

Benjamin Collins

2.3.2 P003-U002 <Sign out>

Description:

Customer will be signed out of their account.

Actors:

Pre-existing Customer

Preconditions:

Customer is currently logged into their account

Main Flows:

1. Customer will click the “Sign Out” button on the persistent menu bar
2. Customer is returned to home page

Alternative Flows:

None

Post-conditions:

Customer is logged out of the system.

Extends:

None

Notes:

None

Authors:

Benjamin Collins

2.3.3 P003-U003 <Register>

Description:

Customer will be prompted with a portal to register an account with the FlightPub booking system. This customer account will store the email address of the customer, full name and flight history.

Actors:

Customer

Preconditions:

No customer is currently logged in on the host's session

Main Flows:

1. Customer will click the "Register" button on the home page or on the persistent menu bar
2. Customer enters their desired username and password, as well as their full name and email address.
3. System checks customers entered details
    3.1. System sanitizes the input
    3.2. Query database for existing username
    3.3. Return success message to customer
4. Customer is returned to the previous page they were on

Alternative Flows:

1. Customer inputs an already existing username and is prompted by the system to enter a different username.
2. Customer inputs invalid or malicious text into the inputs. The system sanitizes all input and if an error occurs the user is notified and allowed to enter appropriate inputs

Post-conditions:

Customer is logged into the system and may now book flights. If input invalid, customer is prompted to enter new input.

Extends:

None

Notes:

None

Authors:

Benjamin Collins

2.3.4 P003-U004 <View Flight History>

Description:

Customer will be able to view all previous flights they have booked on their account, and clicking on them will return their corresponding details. If no previous bookings have been made, then that will be simply displayed

Actors:

Pre-existing Customer

Preconditions:

The customer is signed in

Main Flows:

1. Customer will click the “Account” button on the home page or on the persistent menu bar if not already on this page
2. Customer selects “View Flight History”
3. System checks customer’s previous flights
   3.1. System sanitizes the input
   3.2. Query database for flights under the account
   3.3. Return list of previous flights or “No flights have been previously booked”

Alternative Flows:

None

Post-conditions:

Customer is shown a list of previous flights if any

Extends:

None

Notes:

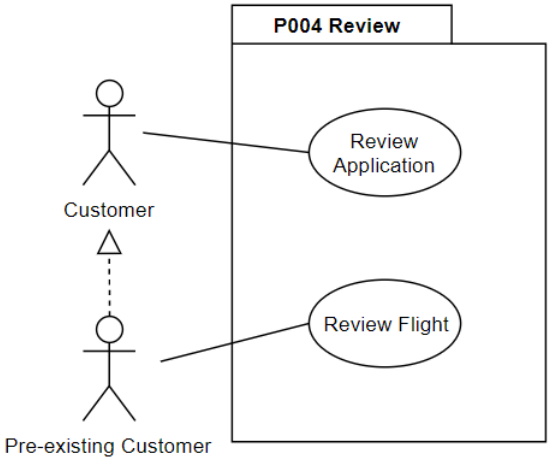
None

Authors:

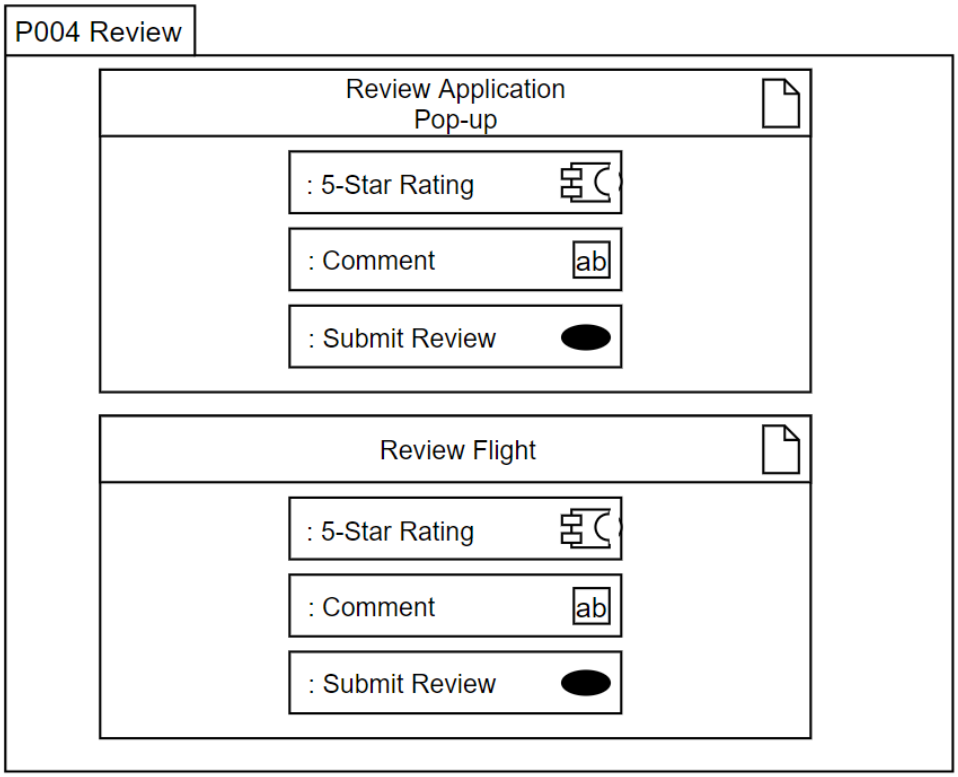
Samuel Brackenrig

2.4 P004 <Review>

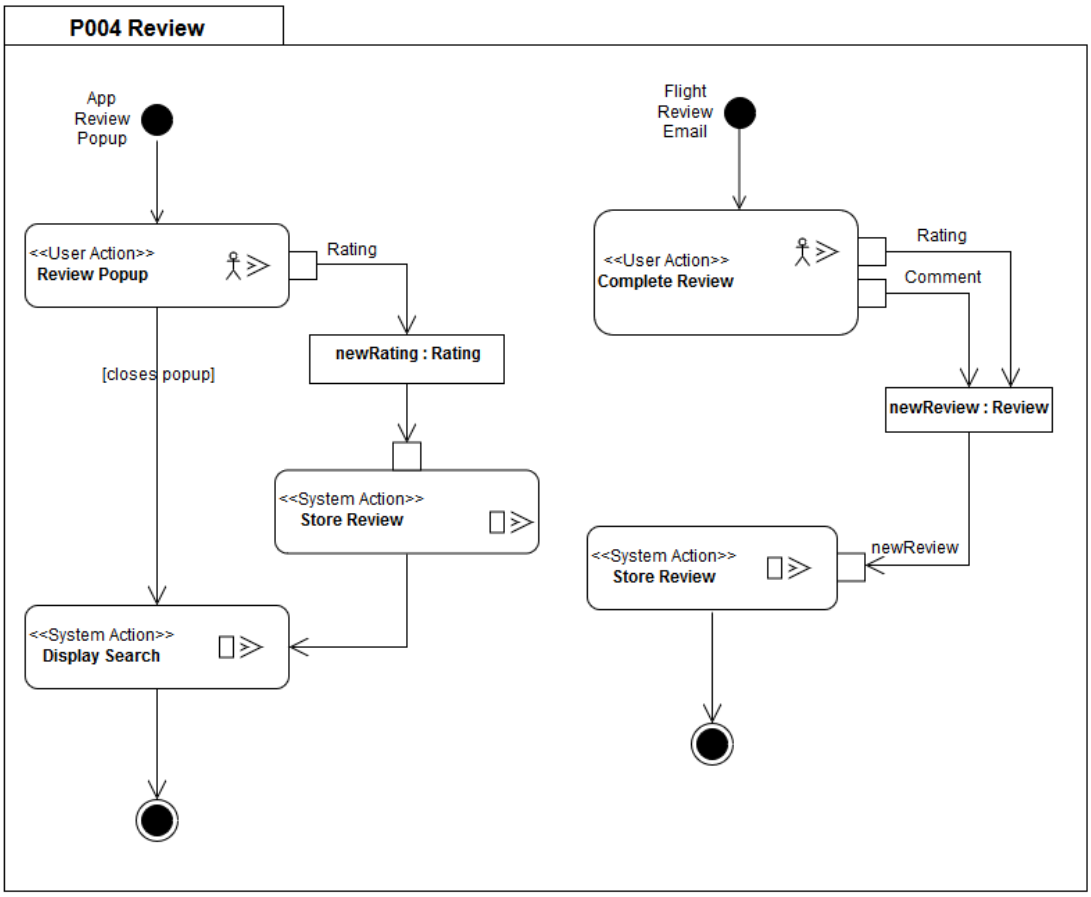
This package encompasses the optional review process provided to the user for reviewing the booking web-application as well as any flights they travel on.



Presentation Model



Process Flow Model



2.4.1 P004-U001 <Review Application>

Description:

Customer will see a pop-up on the screen allowing them to review their experience with the FlightPub booking web-application and leave any comments the customer may have. These reviews are then able to be viewed by the management staff in order to improve on the experience of the website

Actors:

Customer

Preconditions:

Customer has made a flight search and the results are on screen.

Main Flows:

1. Pop-up shown to customer allowing them to rate their experience with the flight booking system
   1.1. Customer selects a star-rating out of 5
   1.2. Customer leaves a comment describing their reason for rating and any additional feedback

2. Customer submits feedback
3. Pop-up closes leaving the search results page

Alternative Flows:

1. Customer chooses to not submit a review and instead closes the pop-up using the 'X' displayed in the corner of the box
2. Customer inputs invalid or malicious text into the input. The system sanitizes the input and stores it as-is

Post-conditions:

Customer has left a review. If customer has chosen to not leave a review, no action is taken.

Extends:

None

Notes:

None

Authors:

Benjamin Collins, Tobias Colson

2.4.2 P004-U002 <Review Flight>

Description:

Customer will receive an email after booking their flight and given a chance to review their experience with the airline they flew with and leave any comments they may have.These Reviews are then able to be viewed in Extra Flight Details of flights with same airline and aircraft on Flight Results page.

Actors:

Pre-existing Customer

Preconditions:

Customer has booked a flight.

Main Flows:

Alternative 1: *Customer is viewing booking history in account page*

1. Customer is taken to a portal via a link in the email allowing them to rate their experience with the airline

Alternative 2: *Ignores email*

1.1. Customer selects a star-rating out of 5

1.2. Customer leaves a comment describing their reason for rating and any additional feedback

2. Customer submits feedback

Alternative 3: *Malicious input*

3. Redirected to a page where they are shown a success message and a link to the FlightPub home page
4. Option to review flight from Accounts page is removed

Alternative Flows:

1. Viewing account history page
    a. Selects Review Flight next to a previous booking
    b. Then as for Main Flow 1.
2. Customer chooses to not submit a review and ignores the email.
3. Customer inputs invalid or malicious text into the input. The system sanitizes the input and stores it as-is

Post-conditions:

Customer has left a review. If customer has chosen to not leave a review, no action is taken.

Extends:

None

Notes:

None

Authors:

Benjamin Collins, Tobias Colson

4 Domain Dictionary

3.1 Terms and Abbreviations

Term	Definition
Presentation Model Key	**stereotype-names and their icons** presentationGroup presentationPage text textInput anchor fileUpload button image inputForm customComponent presentationAlternatives selection
Process Model Key	**stereotype-names and their icons** userAction systemAction

5 Mapping between Requirements and Use Cases

	P001-U001	P001-U002	P001-U003	P001-U004	P001-U005	P001-U006	P004-U001	P004-U002
R1	X							
R2								
R3	X	X						
R4		X						

R5			X			X		
R6			X					
R7								
R8			X					
R9								
R10				X				
R11					X			
R12		X			X			
R13								
R14								
R15								
R16					X			X
R17								X
R18								
R19								

	P002-U001	P002-U002	P002-U003	P002-U004	P003-U001	P003-U002	P003-U003	P003-U004
R1								
R2								
R3								
R4								
R5								
R6								
R7								
R8								
R9								
R10								
R11								
R12								
R13	X	X	X	X				
R14			X					
R15								
R16							X	
R17								
R18					X	X	X	
R19								X