**[Project/Scenario Title]**

**IS213 Enterprise Solution Development**

***[Section-Team]***

**Assignment**

***[Team Members]***

# Introduction

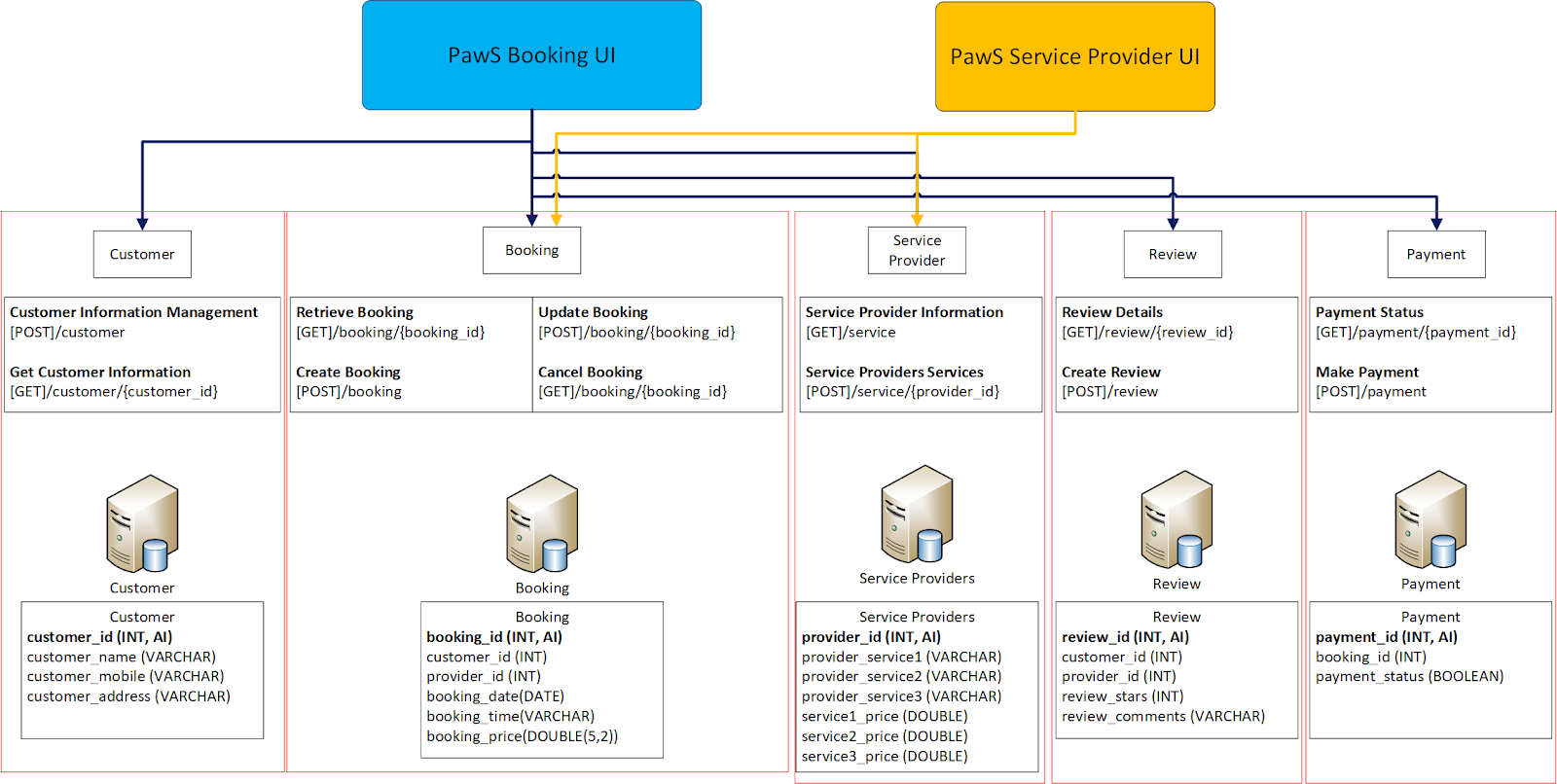
*[Short introduction about business scenario and the user scenarios you are covering]*

We have created a Pet Grooming System (PawSystem). There are two main actors – (i) Customer (with pet) and (ii) Service Provider. There are 5 user scenarios: (i) Customer manage Booking, (ii) Service Provider manage Booking, (iii) Customer creates Review, (iv) Customer makes Payment, (v) Service Provider acknowledges Payment

# Technical Overview Diagram

*[Technical Overview Diagram]*

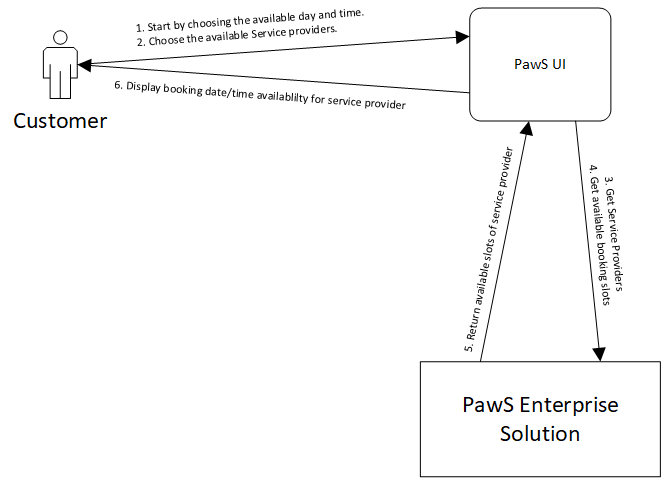
*[If the database portion for any microservice cannot fit into one diagram, have a separate diagram and refer to it]*



# User Scenarios

## [User Scenario 1]

*[User Scenario Diagram – Customer manage Booking]*

 *Sample - User Scenario Diagram*

*[Write-up of this user scenario(s) with reference to the User Scenario Diagram(s).]*

1. Passenger starts the booking request by selecting his available day and time as shown on the PawS UI. Thereafter, the UI will show the passenger a list of available Service Providers he can choose from based on his availability. Passenger chooses amongst the list of Service Providers.
2. Upon receiving this UI request, PawS UI then sends a GET request to PawS Enterprise Solution to get the list of service providers as well as the available booking slots.
3. PawS Enterpise Solution then sends a reply message back to PawS UI containing the list of available slots of service provider.
4. The PawS UI then displays the booking date and time availability for service provider to the passenger.

### (Micro)Services

*[This is where you put in the details of the (micro)service(s) and operations including any external service(s) that are* ***used in this User Scenario ONLY****]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service Name | Operational information (e.g., HTTP URL or AMQP exchange type and keys, if any) | Description of the functionality | Input (if any) | Output (if any) |
| *Booking* | *Add new booking [POST] /booking* | *[No additional description required for this operation as the Operation, Input & Output are clear enough for example]* | *{PassengerID, Location, Destination}* | *{status of the booking} and booking details if any {bookingID, PassengerID, DriverID, DateTime, PickUpLocation, Destination, Price}* |
| *Pricing* | *Get price*  *[GET] /pricing?{PickupLocation}&{Destination}* | *[No additional description required for this operation as the Operation, Input & Output are clear enough for example]* | *{PickupLocation, Destination}* | *{Price}* |
| *Google Distance Matrix API* | [*https://maps.googleapis.com*](https://maps.googleapis.com)  *GET /maps/api/distancematrix /****outputFormat?parameters*** | *Obtain the distance in km between two points for the use of price calculation for a Taxi trip.* | *origins, destinations, key* | *distance (Note: you may put only the data fields that are meaningful to what you are doing, instead of all the data fields)* |

### Beyond the Labs

*[List, describe and explain the things you have done that are not taught in the labs* ***for this User Scenario only (if any)****.]*

1. Invoke Google Distance Matrix API.

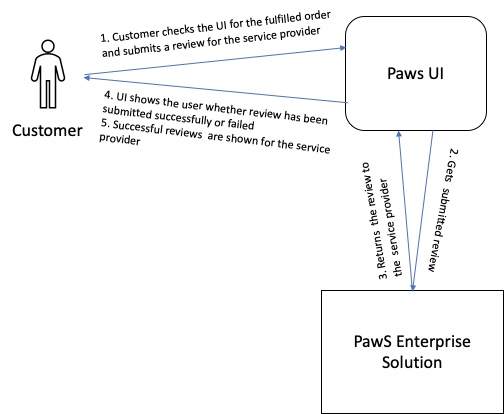
*…*

If Nil for this user scenario, no need to have this subsection.

## 

## [User Scenario 2]

*[User Scenario Diagram – Customer Creates Review]*



*[Write-up of this user scenario(s) with reference to the User Scenario Diagram(s).]*

* + - 1. The customer checks the UI for the fulfilled order and submits a review to the service provider.
      2. The PawS UI then gets the submitted review from the PawS Enterprise Solution. The PawS Enterprise Solution returns the review to the service provider.
      3. The UI then shows the customer the submit status of the review. Successful reviews are shown for the service provider.

### (Micro)Services

*[This is where you put in the details of the (micro)service(s) and operations including any external service(s) that are* ***used in this User Scenario ONLY****- exclude those that have been mentioned in the previous user scenarios.]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service Name | Operational information (e.g., HTTP URL or AMQP exchange type and keys, if any) | Description of the functionality | Input (if any) | Output (if any) |
| *Booking* | *[GET] /booking?status={open}* | *Get a list of bookings based on status* | *{PassengerID, Location, Destination}* | *{list of bookings}* |
| *[PUT] /booking/{bookingID}* | *Update booking status* | *{bookingID, driverID}* | *(status of the update result} and {passenger id, name, address, phone} if any}* |
| *Direct Exchange with*  *[BKEY] passenger.reply*  *[RKEY] passenger.request*  *(BKEY means binding key; RKEY means routing key)* | *Send a request using [RKEY] passenger.request to get passenger information* | *{PassengerID}* | *{id, name, address, phone} of a passenger* |
| *Passenger* | *Direct Exchange with*  *[BKEY] passenger.request*  *[RKEY] passenger.reply* | *Retrieve passenger information and send it back via [RKEY] passenger.reply* | *{PassengerID}* | *{id, name, address, phone} of a passenger* |

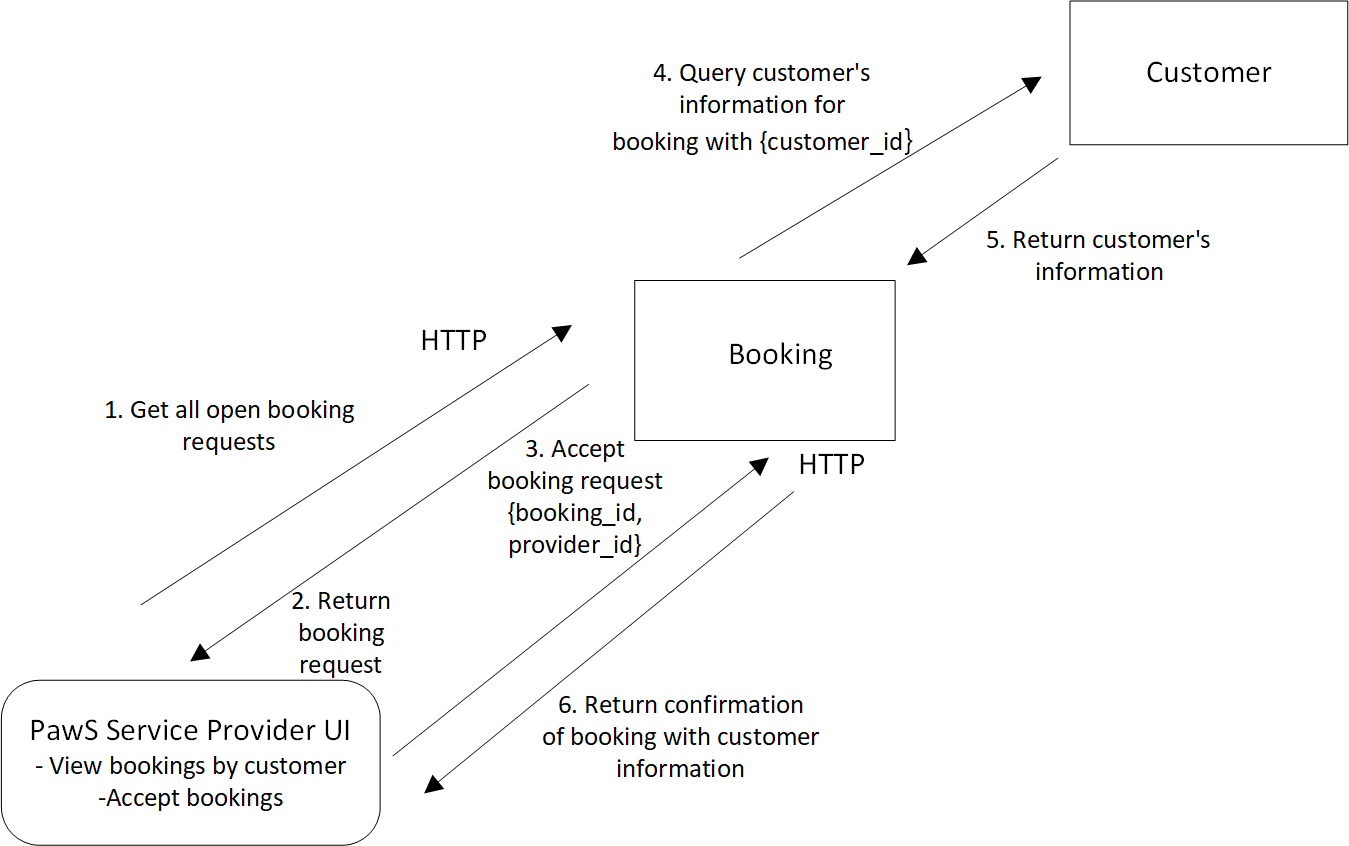
### Beyond the Labs

*[List, describe and explain the things you have done that are not taught in the labs* ***for this User Scenario only (if any)****.]*

If Nil for this user scenario or the beyond-the-labs has been mentioned in the previous User Scenario(s), no need to have this subsection.

## [User Scenario 3]

*[User Scenario Diagram –Service Provider manage Booking]*



*[Write-up of this user scenario(s) with reference to the User Scenario Diagram(s).]*

1.

### (Micro)Services

*[This is where you put in the details of the (micro)service(s) and operations including any external service(s) that are* ***used in this User Scenario ONLY****- exclude those that have been mentioned in the previous user scenarios.]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service Name | Operational information (e.g., HTTP URL or AMQP exchange type and keys, if any) | Description of the functionality | Input (if any) | Output (if any) |
| *Booking* | *[GET] /booking?status={open}* | *Get a list of bookings based on status* | *{PassengerID, Location, Destination}* | *{list of bookings}* |
| *[PUT] /booking/{bookingID}* | *Update booking status* | *{bookingID, driverID}* | *(status of the update result} and {passenger id, name, address, phone} if any}* |
| *Direct Exchange with*  *[BKEY] passenger.reply*  *[RKEY] passenger.request*  *(BKEY means binding key; RKEY means routing key)* | *Send a request using [RKEY] passenger.request to get passenger information* | *{PassengerID}* | *{id, name, address, phone} of a passenger* |
| *Passenger* | *Direct Exchange with*  *[BKEY] passenger.request*  *[RKEY] passenger.reply* | *Retrieve passenger information and send it back via [RKEY] passenger.reply* | *{PassengerID}* | *{id, name, address, phone} of a passenger* |

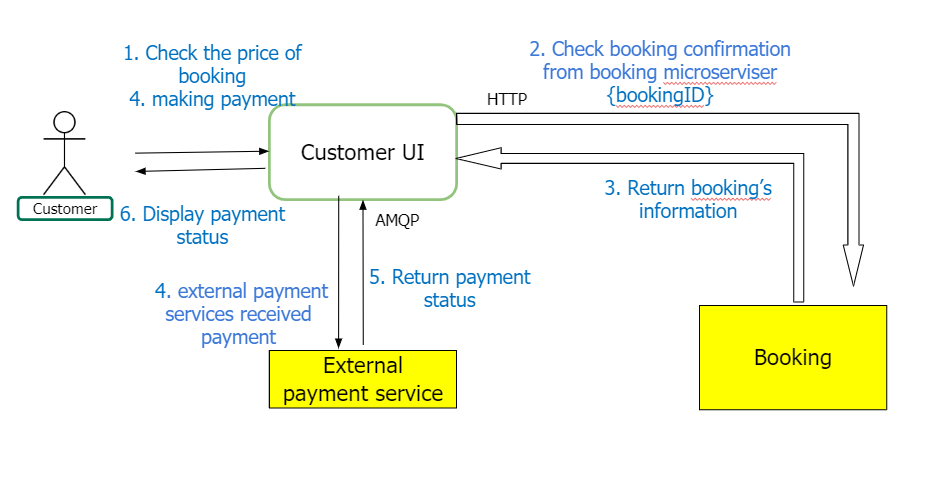
### Beyond the Labs

*[List, describe and explain the things you have done that are not taught in the labs* ***for this User Scenario only (if any)****.]*

If Nil for this user scenario or the beyond-the-labs has been mentioned in the previous User Scenario(s), no need to have this subsection.

## [User Scenario 4]

*[User Scenario Diagram – Customer makes payment]*



*[Write-up of this user scenario(s) with reference to the User Scenario Diagram(s).]*

1. The customer checks the price of his booking on the Customer UI.

2. The customer UI sends a HTTP request message to the booking microservice to get the booking confirmation status for the customer’s booking. The message contains the customer’s booking ID.

3. The booking microservice returns a reply message consisting of the customer’s booking information.

4. Next, the customer proceeds to make payment on the Customer UI.

5.Using AMQP protocol, the Customer UI sends a request message to the external payment service informing the service about the customer’s payment.

6.The external payment service then sends back a reply message containing the payment status of the customer.

7.The Customer UI displays the payment status to the customer.

### (Micro)Services

*[This is where you put in the details of the (micro)service(s) and operations including any external service(s) that are* ***used in this User Scenario ONLY****- exclude those that have been mentioned in the previous user scenarios.]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service Name | Operational information (e.g., HTTP URL or AMQP exchange type and keys, if any) | Description of the functionality | Input (if any) | Output (if any) |
| *Booking* | *[GET] /booking?status={open}* | *Get a list of bookings based on status* | *{PassengerID, Location, Destination}* | *{list of bookings}* |
| *[PUT] /booking/{bookingID}* | *Update booking status* | *{bookingID, driverID}* | *(status of the update result} and {passenger id, name, address, phone} if any}* |
| *Direct Exchange with*  *[BKEY] passenger.reply*  *[RKEY] passenger.request*  *(BKEY means binding key; RKEY means routing key)* | *Send a request using [RKEY] passenger.request to get passenger information* | *{PassengerID}* | *{id, name, address, phone} of a passenger* |
| *Passenger* | *Direct Exchange with*  *[BKEY] passenger.request*  *[RKEY] passenger.reply* | *Retrieve passenger information and send it back via [RKEY] passenger.reply* | *{PassengerID}* | *{id, name, address, phone} of a passenger* |

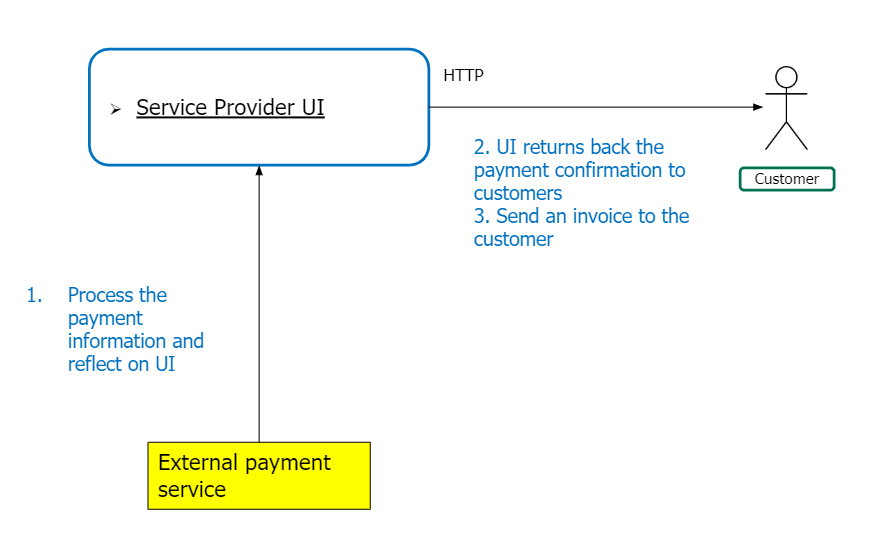
### Beyond the Labs

*[List, describe and explain the things you have done that are not taught in the labs* ***for this User Scenario only (if any)****.]*

If Nil for this user scenario or the beyond-the-labs has been mentioned in the previous User Scenario(s), no need to have this subsection.

## [User Scenario 5]

*[User Scenario Diagram – Service Provider Collects Payment]*



*[Write-up of this user scenario(s) with reference to the User Scenario Diagram(s).]*

1.The external payment service processes the payment system processes the payment information and this is being reflected on the UI.

1. The UI then sends a HTTP request message to the customer containing the payment confirmation details, as well as an invoice.

### (Micro)Services

*[This is where you put in the details of the (micro)service(s) and operations including any external service(s) that are* ***used in this User Scenario ONLY****- exclude those that have been mentioned in the previous user scenarios.]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service Name | Operational information (e.g., HTTP URL or AMQP exchange type and keys, if any) | Description of the functionality | Input (if any) | Output (if any) |
| *Booking* | *[GET] /booking?status={open}* | *Get a list of bookings based on status* | *{PassengerID, Location, Destination}* | *{list of bookings}* |
| *[PUT] /booking/{bookingID}* | *Update booking status* | *{bookingID, driverID}* | *(status of the update result} and {passenger id, name, address, phone} if any}* |
| *Direct Exchange with*  *[BKEY] passenger.reply*  *[RKEY] passenger.request*  *(BKEY means binding key; RKEY means routing key)* | *Send a request using [RKEY] passenger.request to get passenger information* | *{PassengerID}* | *{id, name, address, phone} of a passenger* |
| *Passenger* | *Direct Exchange with*  *[BKEY] passenger.request*  *[RKEY] passenger.reply* | *Retrieve passenger information and send it back via [RKEY] passenger.reply* | *{PassengerID}* | *{id, name, address, phone} of a passenger* |

### Beyond the Labs

*[List, describe and explain the things you have done that are not taught in the labs* ***for this User Scenario only (if any)****.]*

If Nil for this user scenario or the beyond-the-labs has been mentioned in the previous User Scenario(s), no need to have this subsection.