



# Introduction

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

# Technical Overview Diagram

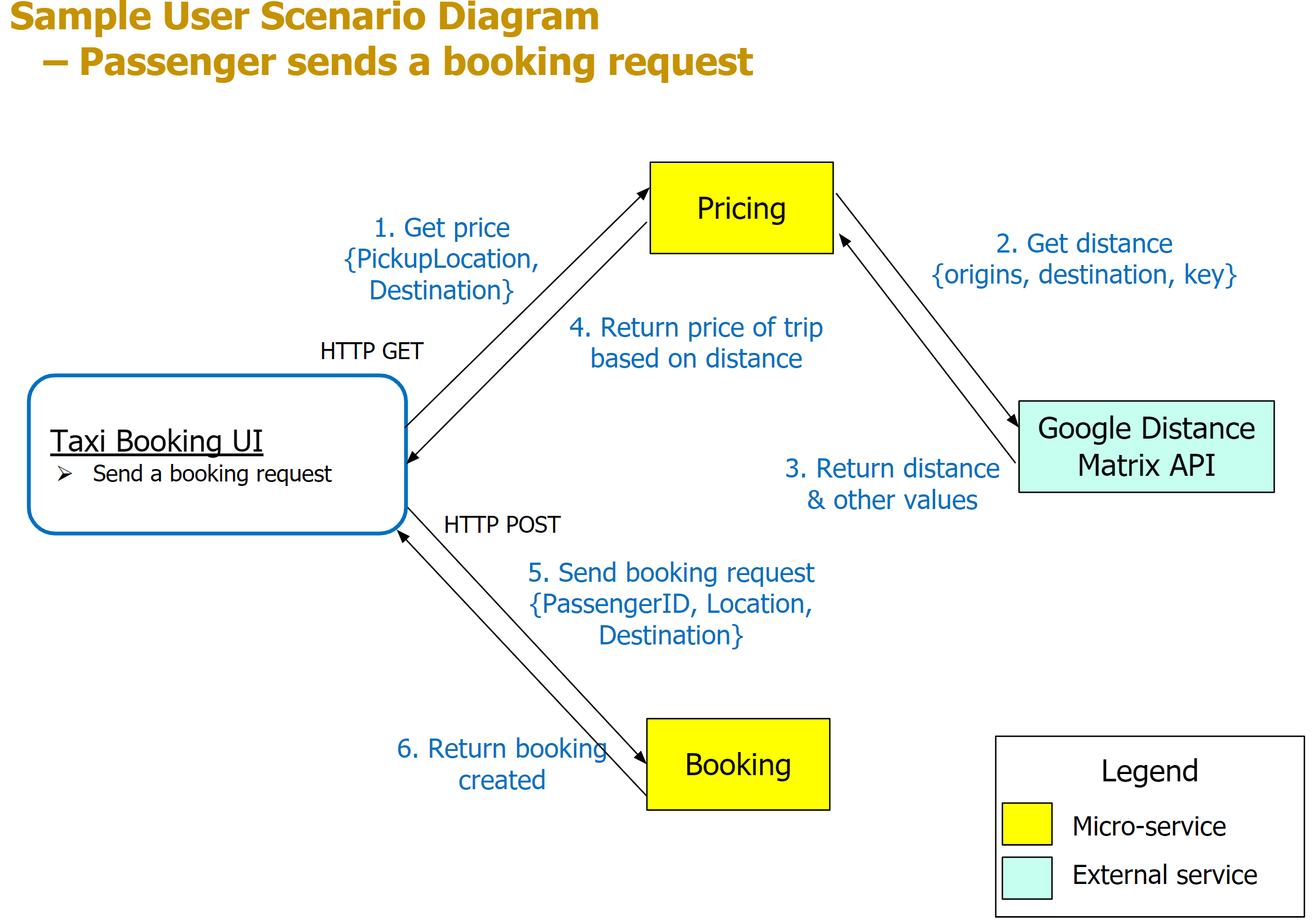
*[Technical Overview Diagram]*

*[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]*

*  
Sample - User Scenario Diagram*

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

e.g. Send a booking request:

1. Passenger starts the booking request by clicking on a button on the UI, then the UI invokes the Pricing service via HTTP GET to get the price quotes.
2. Upon receiving a UI request, the Pricing service invokes the external Google Distance Matrix API to calculate the distance of the potential taxi trip.
3. The Google API returns the calculated distance back to the Pricing service.
4. The Pricing service calculates the prices based on distance and returns the price quotes back to UI.
5. The UI then, upon user confirmation, invokes the Booking service via HTTP POST to add a new booking.
6. The Booking service saves the booking details into the Booking db, then returns a status code with the booking details back to the UI.

### (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

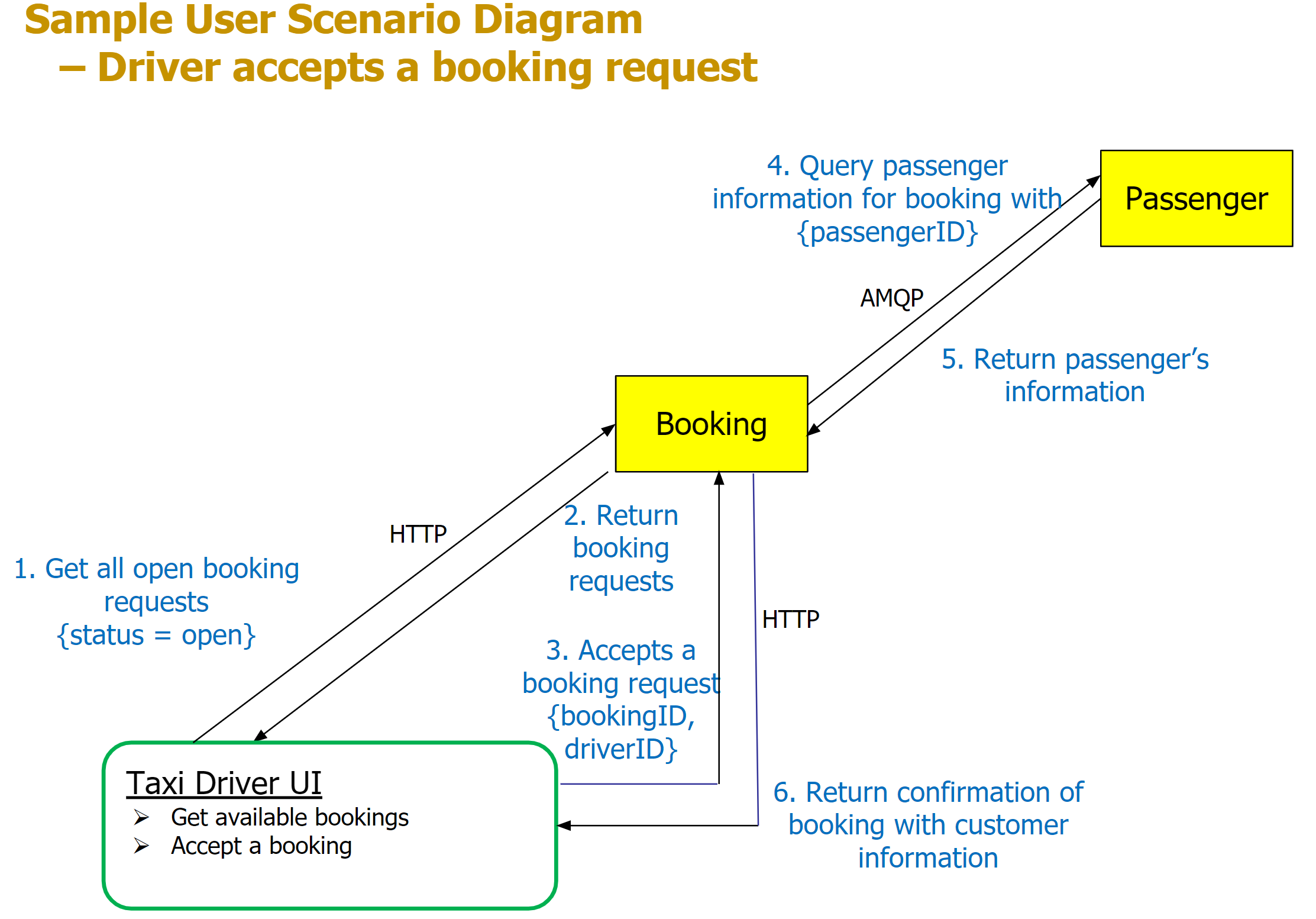
*[List, describe and explain the things you have done that are not taught in the labs* ***for this User Scenario only (if any)****.]*

1. Invoke Google Distance Matrix API.

*…*

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

## [User Scenario 2]

**

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

e.g. Taxi driver uses UI to find a booking

1. The driver uses the UI to invoke the Booking service, via HTTP GET, to get all open bookings.
2. Return all bookings that are currently with the status = open, and the UI displays all bookings.
3. The taxi driver selects a specific booking and triggers the UI to invoke Booking service, via HTTP PUT, to update the status of the booking in the book db.
4. Before returning the response to the Taxi Driver UI from step 3, Booking service sends a message to obtain Passenger information.
5. Passenger service receives the message and returns the requested passenger information from the Passenger db.
6. The Booking service upon receiving passenger information returns the confirmation of the booking to the Taxi Driver UI from step 3 invocation.

### (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 n]

…

# Remaining Beyond the Labs not covered above

*[List, describe and explain beyond the labs component(s) that are not already mentioned in the previous User Scenarios section.]*

e.g.

* Deploy a docker container on a cloud platform (e.g., Azure) that can interact with another part of your solution.
* Deploy separate Docker containers for more than 1 microservices that can interact with each other.