

Payment- LOT 2

SYSTEM DESIGN DOCUMENT

Version 1.0

Confidentiality Statement

**This document contains information that is proprietary and confidential to Tata Consultancy Services Limited, which shall not be disclosed outside Transport for London, transmitted, or duplicated, used in whole or in part for any purpose other than its intended purpose. Any use or disclosure in whole or in part of this information without express written permission of Tata Consultancy Services is prohibited. Any other company and product names mentioned are used for identification purposes only and may be trademarks of their respective owners.**

**© Copyright <2021>, Tata Consultancy Services (TCS)**

TATA Code of Conduct

**We, in our dealings, are self-regulated by a code of conduct as enshrined in the ‘TATA Code of Conduct’. We request your support in helping us adhere to the code in letter and spirit. We request that any violation or potential violation of the code by any person be promptly brought to the notice of the Local Ethics Counsellor or the Principal Ethics Counsellors or the CEO of TCS. All communication received in this regard will be treated and kept as confidential.**

**Table 1: DOCUMENT REVISION LIST**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision No. | Revision Date | Author | Revision Description |
| 1.0 | 22-Nov-2021 | Pranav Gandhi | This document covers Payment Scenarios which will be handled by Lot 2 Team |

Contents

[1 Introduction 7](#_Toc88564106)

[1.1 Purpose 7](#_Toc88564107)

[1.2 List of Abbreviations 7](#_Toc88564108)

[1.3 Audience 8](#_Toc88564109)

[2 Process Description 9](#_Toc88564110)

[2.1 Lookup Process 10](#_Toc88564111)

[2.1.1 Pre-Requisite 10](#_Toc88564112)

[2.1.2 Business Process Map 10](#_Toc88564113)

[2.1.3 Structure of Data 11](#_Toc88564114)

[2.2 Process Payment Transaction 11](#_Toc88564115)

[2.2.1 Pre-Requisite 11](#_Toc88564116)

[2.2.2 Business Process Map 12](#_Toc88564117)

[2.3 Status Update to Service System 12](#_Toc88564118)

[2.3.1 Pre-Requisite 12](#_Toc88564119)

[2.3.2 Business Process Map 13](#_Toc88564120)

[2.3.3 Structure of Data 13](#_Toc88564121)

List of Figures

[Figure 1: Business Process Map: Lookup Process 10](#_Toc88564122)

[Figure 2: Business Process Map: Process Payment Transaction 12](#_Toc88564123)

[Figure 3: Business Process Map: Status Update to Service System 13](#_Toc88564124)

List of Tables

[Table 1: DOCUMENT REVISION LIST 3](#_Toc88564125)

[Table 2: List of Abbreviations 7](#_Toc88564126)

[Table 3: Process List 9](#_Toc88564127)

# Introduction

## Purpose

The objective of this document is to outline design for Payment process that will be initiated from customer via Lot 2 IVR. This document also specifies the details of the process of collecting payment at Inspection Centres via Virtual Terminal. This document specifies Pre-Requisites, Business Process Maps associated with process, Form Specifications, Actions and Stakeholder details, Business Rules and Validations, Alerts and Notification, and Integration. Following are the list of Process/ Subprocesses which are covered under this document.

* Payment Processing through IVR
* Payment Processing through Virtual Terminal from Inspection Centres

## List of Abbreviations

**Table 2: List of Abbreviations**

|  |  |
| --- | --- |
| Abbreviation | Expanded Form |
| TfL | Transport for London |
| TCS | Tata Consultancy Services Ltd. |
| TPH | Taxi and Private Hire |
| PHV | Private Hire Vehicle |
| IVR | Interactive Voice Response |
| CVV | Card Verification Value |

## Audience

This document is intended to provide an understanding on business functions to different teams:

* TfL Project Management
* TfL SMEs
* TCS Project team for Designing Payment Process
* Lot 2 Supplier
* KeyIVR team

# Process Description

This section will cover processes and subprocesses related to collecting payment at the time customer calls to TfL/ Lot 2 contact centre and confirms payment through IVR. Then agent will make a lookup based on input from customer and communicate amount to be paid. If customer agrees to pay then, agent will divert the call to KeyIVR. The process of completing payment transaction is explained in further sections.

This section will also cover the details of the process which will be carried out at Inspection Centres to process the payment via Virtual Terminal of Payment Gateway.

**Table 3: Process List**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Process** | **Remarks** |
| 1 | Lookup Process | Customer calls TfL/ Lot 2 Contact Centre and request for payment information |
| 2 | Process Payment Transaction | Agent will transfer the call to KeyIVR for payment processing |
| 3 | Status Update to Service System | Provide transaction information to Lot 1 Service System |
| 4 | Process Payment via Virtual Terminal | Lot 2 user will select particular request of Vehicle Inspection and chooses option for Payment through Virtual Terminal |

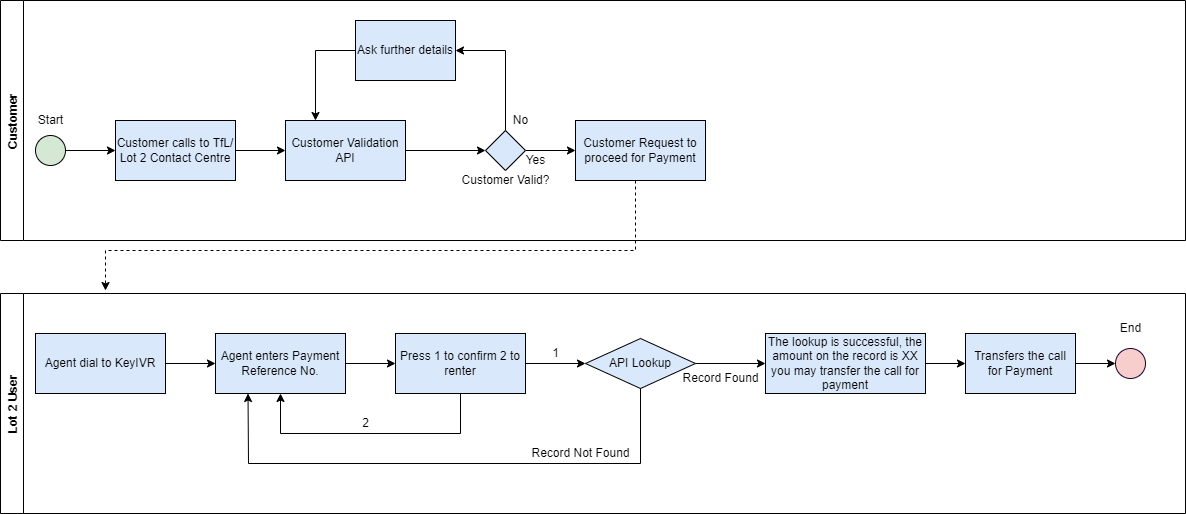
## Lookup Process

This process will be used by TfL/ Lot 2 agent to check the pending payment information for customer. To retrieve these details, agent will enter Payment Reference Number. Based on that information will be fetched and displayed.

### Pre-Requisite

* TfL/ Lot 2 IVR should have valid account and access privilege to access this application.

### Business Process Map



**Figure 1: Business Process Map: Lookup Process**

**Process Flow Description: Lookup Process**

* Customer calls to TfL/ Lot 2 Contact Centre
* Agent will receive call and customer validation API will be called. If customer record is valid and customer requests for payment through IVR then Agent dials to KeyIVR system
* If customer is not valid then agent asks some more details to validate customer
* Once call id diverted to Key IVR then, agent enters Payment Reference No.
* Then agent will be asked to confirm the details. Agent will press 1 to confirm and press 2 for renter the number.
* Upon confirmation API for lookup process will be called from Service System.
* If record is available then, message with amount will be communicated agent and agent forwards call to customer for making payment.
* If record is not found during lookup process then, agent will be asked to renter the details.

### Structure of Data

This section covers what details are passed while requesting details from service system during Lookup process

Input format

**GET Invoice: -** KeyIVR API will get Invoice Details by calling

**Endpoint URL: -** https://{tfl\_dns}/accountLookup/

**Request Body: -** {requestType, requestNo}

requestType – will hold identification of process like Apply for new Driver Licence, Apply for New Operator Licence etc.

requestNo – will hold reference number of the process for which payment needs to be made

Output format

**If Successful**

{

"status": "Ok"

"reference": "{invoice reference}"

"amount":"{amount to be paid}"

}

**If Failed**

{

"status": "Failed"

''description": "{if status = Failed, then description

of the error(Unauthorized/Internal Server

Error/Invalid data/etc.)"

}

## Process Payment Transaction

This process will be initiated if Lookup process is successful, and agent diverts the call to customer to enter details for processing payment transactions.

### Pre-Requisite

* Lookup process should be successful

### Business Process Map

**Figure 2: Business Process Map: Process Payment Transaction**

**Process Flow Description: Process Payment Transaction**

* This process starts once call is transferred to customer for payment processing.
* Customer will have to confirm whether they are card holder or not by pressing 1 or 2 respectively.
* If customer press 2 then call will be transferred back to agent as customer is not a card holder in this case.
* If customer press 1 then customer will have to enter card details followed by \*.
* Then customer needs to enter expiry date as first 2 digits denotes month, and last 2 digits denotes year.
* Then Customer needs to enter CVV number for payment authorization. After that customer needs to wait till response from payment gateway is received.
* If payment is successful, then KeyIVR will call API from service system to update the payment status and customer will be communicated with response of the payment.
* If payment fails, and attempts are less than 3 then customer can re-enter the details or use a different payment method. And if attempts are more than 3 then, call will be transferred to agent.

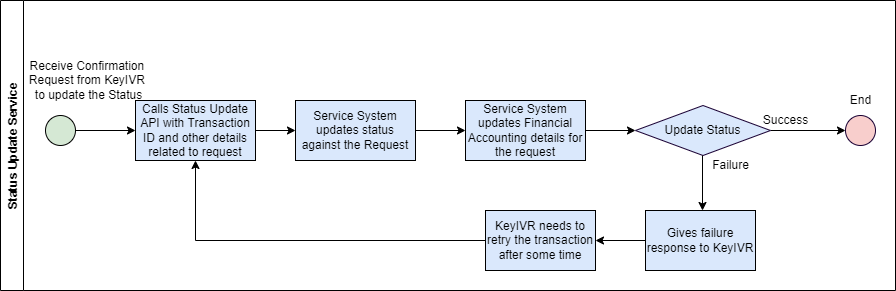
## Status Update to Service System

Once payment is successful, KeyIVR system calls the API from Service System which will update the status of the request in service system and next steps will be followed to complete the process of the requests.

### Pre-Requisite

* Process Payment Transaction should be successfully completed

### Business Process Map



**Figure 3: Business Process Map: Status Update to Service System**

**Process Flow Description: Status Update to Service System**

* KeyIVR will call API from service system to update the status in the service system
* Status of the request will be updated in the service system
* Financial accounting related details also will be updated in the service system
* If updates to status and other details are successful, then customer will be notified accordingly
* If updates to status or other details fails, then service system will give failure response to KeyIVR and KeyIVR needs to call the same service after some time.

### Structure of Data

This section covers what details are passed while requesting details from service system during Status update process

Input format

**POST Request: -** KeyIVR API will send payment Details by calling

Endpoint URL: - https://{tfl\_dns}/paymentUpdate/

by using Authentication type:

API Key:

key: Authorization

value: {secret code}

**Request body: -**

{ transactionId

reference

amount

currency

merchantReturn

status

reference2

createdAt

}

Output format

**If Successful**

{

"status": "OK"

"statusCode": "200"

''description": "empty string"

}}

**If Failed**

{

"status": "FAILED"

"statusCode": "{400/401/404/409/etc.}"

''description": "{description of the

error(Unauthorized/Internal Server Error/Invalid

data/etc.)"}

}

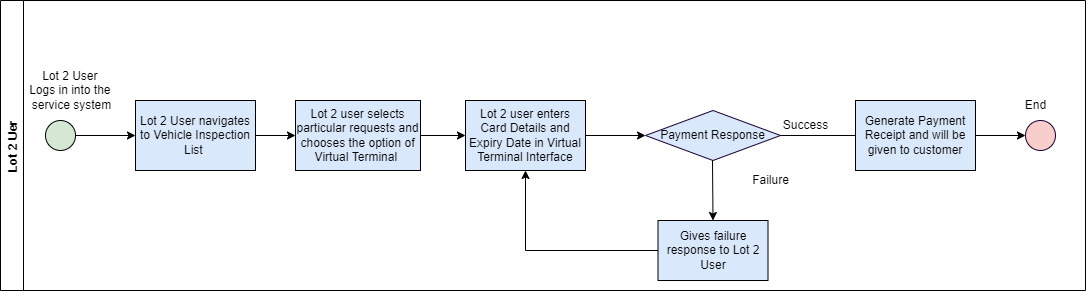
## Process Payment via Virtual Terminal

This process will be available to Lot 2 authorised user for processing payment at Inspection Centre. Lot 2 user will search the request of Vehicle Inspection based on inputs from customer and chooses the option of payment through Virtual Terminal. After payment is successfully completed Lot 2 Inspection centre will complete further steps to complete the processing of Vehicle Inspection request.

### Pre-Requisite

* Customer should have valid Vehicle Inspection request
* Lot 2 user should have privilege to use Virtual Terminal

### Business Process Map



**Figure 2: Process Payment via Virtual Terminal**

**Process Flow Description: Process Payment via Virtual Terminal**

1. Customer visits Vehicle Inspection Centre and agrees to pay at inspection centre
2. Lot 2 user will login into the service system and search the Vehicle Inspection request
3. Lot 2 authorised user will get an option to process the payment through Virtual Terminal
4. On selection of Virtual Terminal Payment option, user will be redirected to Global Payment’s Virtual Terminal page
5. Lot 2 user will enter Card Number and Expiry Date of card to process the payment
6. After processing the payment, Lot 2 user will receive the successful or failure response
7. If payment is successful, then Lot 2 user will communicate to customer and process the next steps. If payment is failed, then Lot 2 user will confirm the details or request for another payment method.

### Form Specifications

This section describes form specifications for raising a refund request. Form specification contains following attributes:

**Field Name** – Indicates name of the field

**Field Type** – Indicates Data Type of field e.g., Textbox, Label, Datebox, Dropdown etc.

**Field Length** – Indicates Length of the field

**Description** – Details like brief description, condition, information etc.

**M/ O/ C/ A** – Indicates field is Mandatory, Optional, Conditional or Auto-populated

#### Form: Virtual Terminal Payment

Lot 2 user will go to Virtual Payment screen and provide following details. This screen will not be developed in service system and it will be part of Global Payments solution.

**Table 8: Form Specifications: Virtual Terminal Payment**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sub-Section** | **Field Name** | **Field Type** | **Field Length** | **Description** | **M/O/C/A** | **Remarks** |
| ***Payment Details*** | Card Number | Numeric | 16 | User enters Card Number in this field | M |  |
| Expiry Date | Datebox |  | User enters expiry Date in this field | M |  |
| Amount | Decimal | 20 | Amount will be auto populated based in request | A |  |

##### **Actions & Stakeholders**

**Stakeholder**: Lot 2 User

**Actions**:

**Submit:** - On click of submit button, Payment will be processed, and response will be displayed on the screen

**Cancel: -** On click of this button, payment processing will be stopped

##### **Alert and Notifications**

**Notification**

**Table 9- Alert and Notifications – Initiate Refund Screen**

|  |  |  |  |
| --- | --- | --- | --- |
| **Channel** | **Notification** | **Trigger Event** | **Notified To** |
| Virtual Terminal User Interface | Payment is processed successfully | After successful processing of payment | Lot 2 User |
| Virtual Terminal User Interface | Payment is failed | After processing of payment fails | Lot 2 User |

### Validations & Business Rules

**Table 14: Validations & Business Rules: Initiate Refund**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Business Rules | Remarks |
| 1 | Customer should select valid request for Payment Processing |  |
| 2 | Multiple Payment under same request should not be allowed |  |

### Outcome of the Process

* Request number will be generated on submission of the request.
* Refund receipt will be generated on successful payment.

**Open Points**

* Format for Refund Receipt on successful process of Refund
* Format for each notification mentioned in Section 2.4.2