**OCMS 50 – eService Payment for Offence**

**Based on Functional Document v1.4**

**Prepared** **by**



MGG **SOFTWARE**

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Updated By** | **Date** | **Changes** |
| v1.0 | Mubiyarto Wibisono, Mochamad Eka Ramadan | 29/04/2025 | Document Initiation |
| v1.1 | Mubi | 18/08/2025 | * Restructure Document * Add Payment Status * Revise flow for payment handling |
| v1.2 | Rafif | 20/08/2025 | * Section 1 - Remove eocms\_payment\_matrix from diagram flow * Add section 2 - Send outstanding parking offence notices to the payment channels according to Payment Matrix * Add section 3 - Outstanding Notice Search and Payment through AXS * Removing advantage and weakness sections |
| v1.3 | Rafif | 26/08/2025 | * Section 1.6: Add new Validate captcha search using Vehicle Number and Notice Number * Section 2.4: Modify payment matrix based on OCMS 43 feedback |
| v1.4 | Rafif | 08/09/2025 | * Remove high level bussines flow, design ratioanale * Separate flow search outstanding notices to 3 flows by each search param * Modify captcha flow to use turnstile |
| V1.5 | Mubi | 22/09/2025 | * Change payment matrix process * Change Singpass/Corppass URL to latest SPCP spec * Remove notice\_payment\_flag to use payment\_acceptance\_allowed * Update process search outstanding notice include 5 mins check * Add new field payment\_date\_and\_time in VON and eVON * Rename field in eocms\_user\_message |
| V1.6 | Mubi, Rafif | 14/11/2025 | * Change search by id\_no notice flow to check entityId * Added 1 more param to check outstanding notice idType * Change store URAPG response from java memory to variable * Change eocms\_user\_message to follow latest * Change response API just get necessary field * Combine payment flow into one big flow |

**Table of Content**

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Content** | | **Page** |
| 1 | Search Notices for Parking Fines | | 5 |
| 1.1 | Use Case | 6 |
| 1.2 | High Level Technical Flow | 7 |
| 1.3 | Motorist search notices for parking fines | 8 |
| 1.3.1 | Diagram Flow Image search using vehicle no | 8 |
| 1.3.2 | Diagram Flow Image search using notice no | 10 |
| 1.3.3 | Diagram Flow Image search using id no | 12 |
| 1.3.4 | API Specification | 14 |
| 1.3.4.1 | API for eService | 14 |
| 1.3.5 | Database and Data Mapping | 15 |
| 1.3.5.1 | Search Notice Parameter | 15 |
| 1.3.5.2 | UI Data Mapping Non-Payable Fines | 15 |
| 1.3.6 | Success Outcome | 15 |
| 1.3.7 | Error Handling | 16 |
| 1.3.7.1 | Application Error Handling | 16 |
| 1.3.7.2 | API Error Handling | 16 |
| 1.4 | Check 5 Mins Transaction | 17 |
| 1.4.1 | Diagram Flow Image | 17 |
| 1.4.2 | Database and Data Mapping | 19 |
| 1.4.2.1 | Get Today Transactions | 19 |
| 1.4.2.2 | Get User Message | 19 |
| 1.4.3 | Success Outcome | 19 |
| 1.4.4 | Error Handling | 20 |
| 1.5 | Validate Captcha Search using Vehicle Number or Notice Number | 21 |
| 1.5.1 | Diagram Flow Image | 21 |
| 1.5.2 | API Specification | 22 |
| 1.5.2.1 | API for eService | 22 |
| 1.5.3 | Success Outcome | 23 |
| 1.5.4 | Error Handling | 23 |
| 2 | Send outstanding parking offence notices to the payment channels according to Payment Matrix | | 24 |
| 2.1 | Use Case | 25 |
| 2.2 | Singpass Corppass Login | 26 |
| 2.2.1 | Diagram Flow Image | 26 |
| 2.2.2 | API Specification | 27 |
| 2.2.2.1 | API for eService | 27 |
| 2.2.2.2 | API consume | 29 |
| 2.2.2.3 | API Provide | 30 |
| 2.2.3 | Data Mapping | 30 |
| 2.2.3.1 | createAppTxnId | 30 |
| 2.2.3.2 | getAuthResponse | 30 |
| 2.2.4 | Successful Outcome | 31 |
| 2.2.5 | Error Handling | 31 |
| 2.2.5.1 | createAppTxnId | 31 |
| 2.2.5.2 | getAuthResponse | 31 |
| 2.3 | Payment Matrix | 32 |
| 2.3.1 | Diagram Flow Image | 32 |
| 2.3.2 | Data Mapping | 34 |
| 2.3.3 | Success Outcome | 34 |
| 2.3.4 | Error Handling | 35 |
| 2.4 | Sample Data User Message | 35 |
| 3 | Outstanding Notice Search and Payment through AXS | | 36 |
| 3.1 | Use Case | 37 |
| 3.2 | Get Outstanding Notice | 38 |
| 3.2.1 | Diagram Flow Image | 38 |
| 3.2.2 | API Specification | 40 |
| 3.2.2.1 | API Provide | 40 |
| 3.2.3 | Data Mapping | 42 |
| 3.2.4 | Success Outcome | 42 |
| 3.2.5 | Error Handling | 42 |
| 3.2.5.1 | Application Error Handling | 42 |
| 3.2.5.2 | API Error Specification | 43 |
| 3.3 | Payment Transaction | 43 |
| 3.3.1 | Diagram Flow Image | 43 |
| 3.3.2 | API Specification | 44 |
| 3.3.2.1 | API Provide | 44 |
| 3.3.2.2 | API Consume | 46 |
| 3.3.3 | Data Mapping | 47 |
| 3.3.3.1 | AXS Payment | 47 |
| 3.3.3.2 | URAPG Payment Transaction for AXS | 47 |
| 3.3.4 | Success Outcome | 48 |
| 3.3.5 | Error Handling | 48 |
| 3.3.5.1 | Application Error Handling | 48 |
| 3.3.5.2 | API Error Specification | 48 |
| 4 | Process Payment with Payment Gateway | | 50 |
| 4.1 | Use Case | 51 |
| 4.2 | High Level Technical Flow | 52 |
| 4.2.1 | Diagram Flow Image | 52 |
| 5 | Payment processing and integration with URA Payment Gateway | | 54 |
| 5.1 | Use Case | 55 |
| 5.2 | Diagram Flow Image | 56 |
| 5.3 | API Specification | 59 |
| 5.3.1 | API for eService | 59 |
| 5.3.1.1 | API Generate Transaction | 59 |
| 5.3.2 | API Consume | 60 |
| 5.3.2.1 | API Call URA PG | 60 |
| 5.4 | Database and Data Mapping | 61 |
| 5.4.1 | Payload for StorePG Json | 61 |
| 5.4.2 | Insert to eocms\_web\_txn\_detail | 61 |
| 5.4.3 | Patch eocms\_web\_txn\_detail | 62 |
| 5.5 | Success Outcome | 62 |
| 5.6 | Error Handling | 62 |
| 6 | Transaction Success and Failure Handling | | 64 |
| 6.1 | Use Case | 65 |
| 6.2 | Diagram Flow Image | 66 |
| 6.3 | API Specification | 69 |
| 6.3.1 | API Provide | 69 |
| 6.3.1.1 | API URA PG Callback Success | 69 |
| 6.3.1.2 | API URA PG Callback Failed | 69 |
| 6.3.2 | API Consume | 70 |
| 6.3.2.1 | API Check Payment Status from URA Payment Gateway | 70 |
| 6.4 | Database and Data Mapping | 71 |
| 6.4.1 | Patch data to eocms\_web\_txn\_detail | 71 |
| 6.4.2 | UI Data Mapping Transaction Payment Success | 71 |
| 6.4.3 | UI Patch eocms\_valid\_offence\_notice | 71 |
| 6.5 | Success Outcome | 71 |
| 6.6 | Error Handling | 72 |
| 7 | Receipt Download | | 73 |
| 7.1 | Use Case | 74 |
| 7.2 | Diagram Flow Image | 75 |
| 7.3 | API Specification | 76 |
| 7.3.1 | API for eService | 76 |
| 7.3.1.1 | API Get Transaction Details | 76 |
| 7.3.1.2 | API Download Receipt | 77 |
| 7.4 | Database and Data Mapping | 78 |
| 7.4.1 | Data in Generated Receipt | 78 |
| 7.5 | Success Outcome | 78 |
| 7.6 | Error Handling | 79 |

# **Section 1 - Search Notices for Parking Fines**

### **Use Case**

1. The Pay Parking Offence Notice eService has a search function that lets Motorists search for outstanding/unpaid Notices to make payment.
2. The function allows Motorists to search for Notices using the following search parameter types:
3. Vehicle number
4. Notice number.
5. NRIC, FIN or UEN by logging in to their Singpass or Corppass account
6. When a search is initiated, the OCMS backend retrieves the notice(s) from the database and checks each notice to determine:
   1. Whether it is payable or not
   2. Whether a payment attempt was made today, and if it occurred within the last 5 minutes
   3. Whether it has been paid within the past 6 months
7. The notice(s) will be returned to the eService Portal, and the search result will be classified and displayed onscreen:
   1. Notices that users can select to make payment
   2. Notices that are non-payable and cannot be selected for payment
8. User who login to the eService portal using their SingPass or Corppass will be able to access these additional functions:
   1. View paid notices from the past six months
   2. Download and print receipts for notices paid within the past 6 months



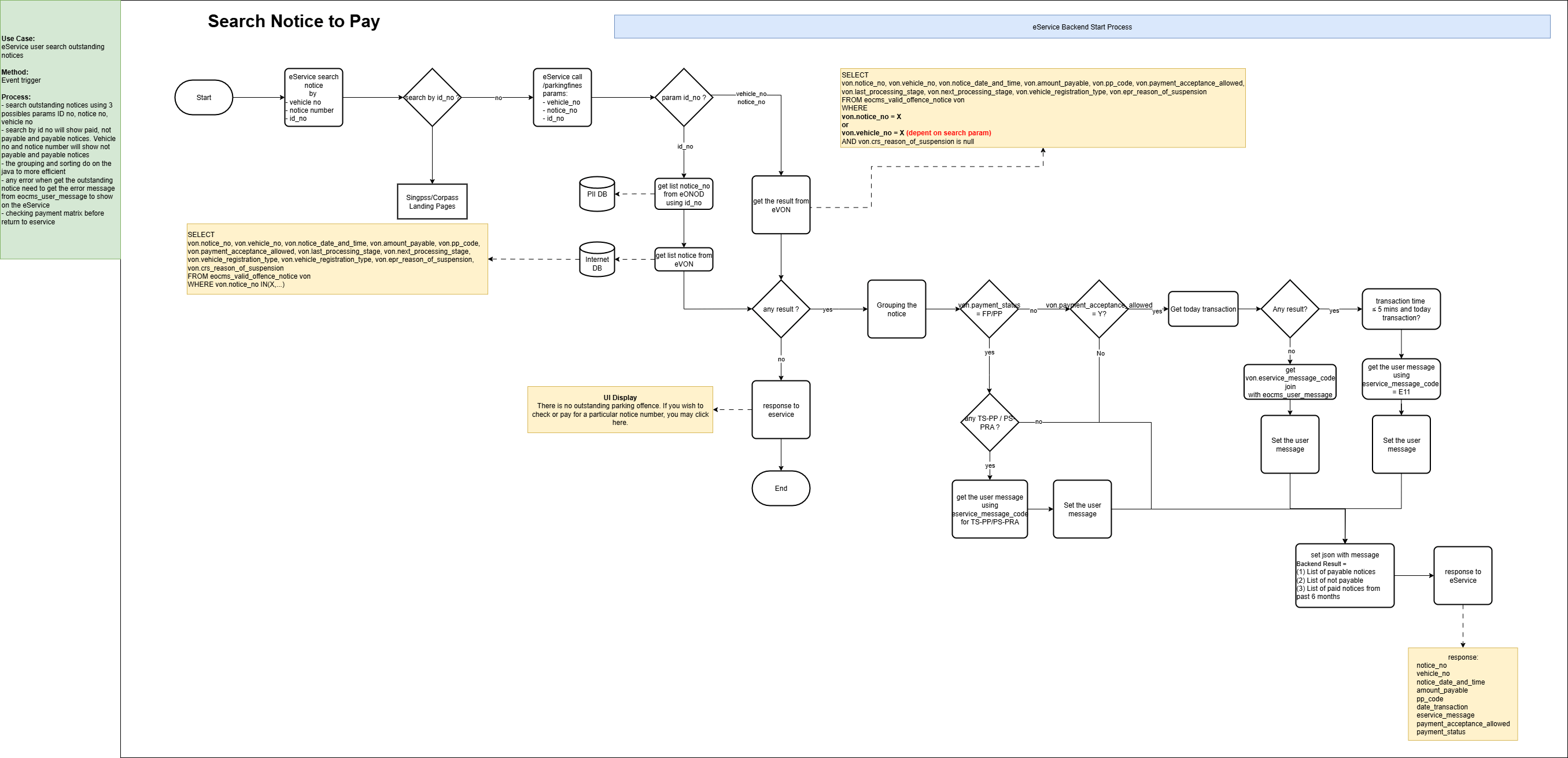








### **High Level Technical Flow**



NOTE: Due to page size limit, the full-sized image is appended.

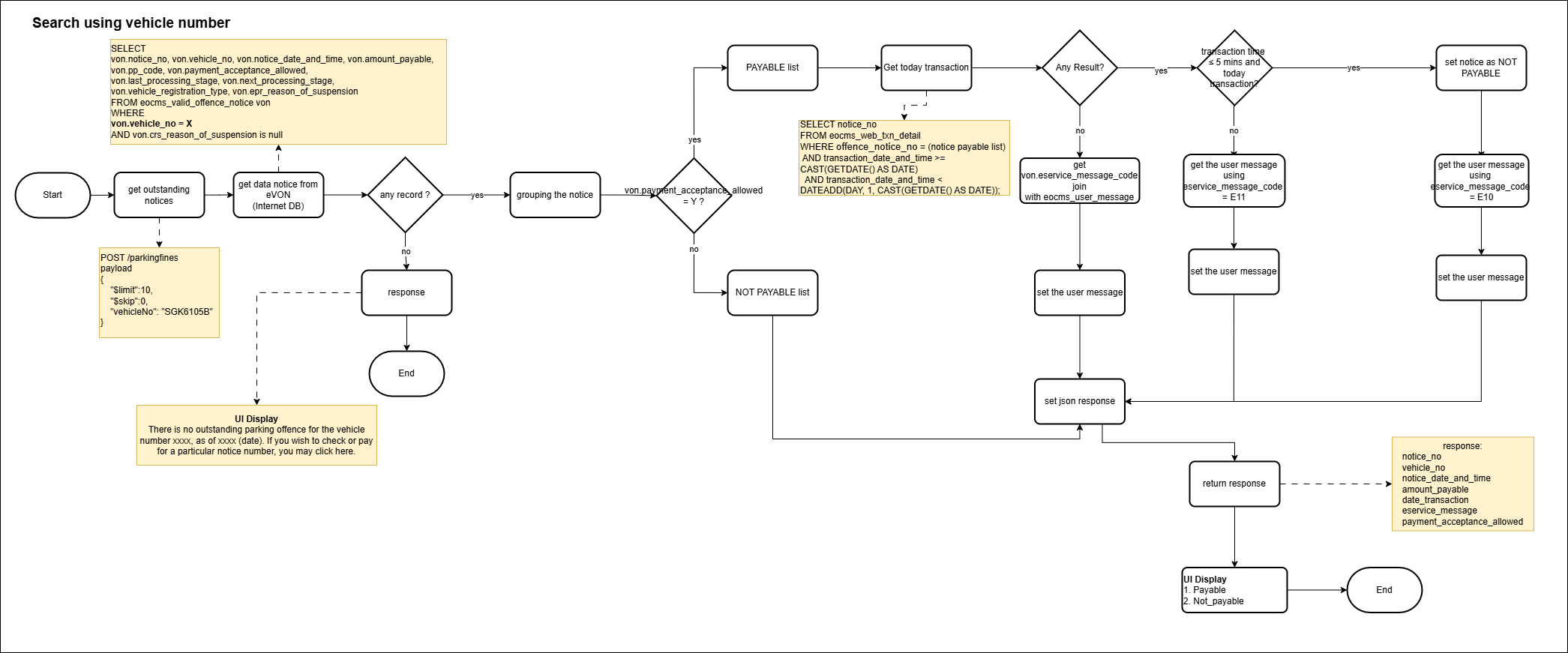


|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Start | Initialization | Flow begins for “search notice to pay”. |
| Eservice Search Page | System Navigation | *Refer to section 1.3* |
| Grouping notices | System Action | *Refer to Section 1.3* |
| Check transaction time  ≤ 5 mins | System Action | *Refer to Section 1.3* |
| Respond To Eservice | API Response | Return final json payload to front‑end. |
| End | Termination | Flow completes. |



### **Motorist search notices for parking fines**

#### **Diagram Flow Image search using vehicle no**



NOTE: Due to page size limit, the full-sized image is appended.





|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Definition** | **Brief Description** | |
| Get outstanding notices | System Interaction | User search notice using vehicle\_no | |
| Get notice data from eVON | System Interaction | Backend queries eVON to retrieve matching notices for the vehicle. | |
| Any record? | Decision | If no, return “There is no outstanding parking offence for the vehicle number xxxx “.  If yes, continue. | |
| Group notices | System Interaction | Backend classifies each notice and group by  payment\_acceptance\_allowed | |
| Check payment\_acceptance\_allowed | System Interaction | Check flag for grouping | |
| PAYABLE list (if Yes) | System Interaction | Collect notices flagged PAYABLE for further checks. | |
| Any result? | Decision | If no, proceed to payment matrix and set json.  If yes, evaluate transaction time. | |
| Transaction time < 5 mins? | Decision | If < 5 minutes, mark the notice NOT\_PAYABLE (still processing) and add the <5-min message; if ≥5 minutes, keep as PAYABLE and add the >5-min message. | |
| NOT\_PAYABLE list (if No) | System Interaction | Collect notices flagged NOT\_PAYABLE (will be shown but cannot be paid). | |
| Set JSON response | System Interaction | Construct response list with fields:  notice\_no  vehicle\_no  notice\_date\_and\_time  amount\_payable  date\_transaction  eservice\_message  payment\_acceptance\_allowed | |
| Return response | System Interaction | Send the JSON to the eService. | |
| UI Display | Presentation | Frontend shows two sections:  Payable  Not payable (with reasons/messages). | |
| End | Termination | Flow completes. | |

#### **Diagram Flow Image search using notice no**

A diagram of a work flow

AI-generated content may be incorrect.

NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Definition** | **Brief Description** | |
| Get outstanding notices | System Interaction | User search notice using notice\_no | |
| Get notice data from eVON | System Interaction | Backend queries eVON for the specified notice numbers. | |
| Any record? | Decision | If no, return “no outstanding parking offence for the notice”.  If yes, continue. | |
| Group notices | System Interaction | Backend classifies each notice and group by | |
| payment\_acceptance\_allowed | System Interaction | Check flag for grouping | |
| PAYABLE list (if Yes) | System Interaction | Collect notices flagged PAYABLE for duplicate-payment checks. | |
| Check today’s payments for PAYABLE | System Interaction | For each PAYABLE notice, look up today’s transactions in web\_txn\_detail to prevent duplicate payment. | |
| Any result? | Decision | If no, proceed to payment matrix and set json.  If yes, evaluate transaction time. | |
| Transaction time < 5 mins? | Decision | If < 5 minutes, mark the notice NOT\_PAYABLE (still processing) and add the <5-min message; if ≥5 minutes, keep as PAYABLE and add the >5-min message. | |
| NOT\_PAYABLE list (if No) | System Interaction | Collect notices flagged NOT\_PAYABLE (will be shown but cannot be paid). | |
| Set JSON response | System Interaction | Construct response list with fields:  notice\_no  vehicle\_no  notice\_date\_and\_time  amount\_payable  date\_transaction  eservice\_message  payment\_acceptance\_allowed | |
| Return response | System Interaction | Send the JSON to the eService. | |
| UI Display | Presentation | Frontend shows two sections:  Payable  Not payable (with reasons/messages). | |
| End | Termination | Flow completes. | |

#### **Diagram Flow Image search using id no**

A diagram of a diagram

AI-generated content may be incorrect.

NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Login using Singpass/Corppass | User Action | User authenticates (individual via Singpass; company via Corppass/UEN). |
| Check entityType = UEN? | Decision | Check response entityType = UEN?  If yes, get outstanding notice using entityId,  If not UEN, get outstanding notice using nric. |
| ~~Get outstanding notices~~ | ~~System Interaction~~ | ~~eService calls backend~~  ~~POST /parkingfines payload {     "$limit":10,     "$skip":0,~~  ~~"idNo": "S3558450A" }~~ |
| Get list of notice\_no from eONDD (PII DB) | System Interaction | Backend maps the authenticated ID to related notice numbers from PII (owner/driver mapping). |
| Any record? | Decision | If no notices are linked to the ID, return “no outstanding parking offence for the ID”.  If yes, continue. |
| Get data notice from eVON (Internet DB) | System Interaction | Retrieve details for those notice numbers from eVON. |
| Any record? | Decision | If no notices, return “no outstanding parking offence for the ID”.  If yes, continue. |
| Group the notice | System Interaction | Backend classifies each notice and group by |
| Check payment\_status | System Interaction | Check payment\_status = FP/PP |
| Check payment\_acceptance\_allowed (If No) | System Interaction | Check flag for grouping |
| PAYABLE list (if Yes) | System Interaction | Collect notices flagged PAYABLE for duplicate-payment checks. |
| Check today’s payments for PAYABLE | System Interaction | For each PAYABLE notice, look up today’s transactions in web\_txn\_detail to prevent duplicate payment. |
| Any result? | Decision | If no, proceed to payment matrix and set json.  If yes, evaluate transaction time. |
| Transaction time < 5 mins? | Decision | If < 5 minutes, mark the notice NOT\_PAYABLE (still processing) and add the <5-min message; if ≥5 minutes, keep as PAYABLE and add the >5-min message. |
| PAID list | System Interaction | Collect notices flagged PAID (already paid). |
| ~~PAID list – TS-PP / PS-PRA present?~~ | ~~Decision~~ | ~~If any PAID notice involves TS-PP or PS-PRA, fetch the corresponding user error text and attach as error\_message; include in response.~~ |
| NOT\_PAYABLE list | System Interaction | Collect notices flagged NOT\_PAYABLE (will be shown but cannot be paid). |
| Set JSON response | System Interaction | Construct the response for UI |
| Return response | System Interaction | Send the JSON to the eService. |
| UI Display | Presentation | Frontend shows three sections:  Paid (past 6 months)  Payable  Not\_payable (with messages). |



#### **API Specification**

##### **API for eService**

###### **API Get Notice Details**

|  |  |
| --- | --- |
| caAPI Name | parkingfines |
| URL | UAT: https://parking2.uraz.gov.sg/ocms/v1/parkingfines  PRD: https://parking.uraz.gov.sg/ocms/v1/parkingfines |
| Description | The API to get notice for payment |
| Method | POST |
| Header | {  "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  "Content-Type": "application/json"  } |
| Payload | {      "$limit":10,      "$skip":0,      "vehicleNo":" SGK6105B"  } |
| Response | {      "total": 1,      "limit": 9999,      "skip": 0,      "data": [          {              "noticeNo":"441000236X",              "vehicleNo":"SGK6105B",              "noticeDateAndTime":"2025-05-01T14:30:00",              "amountPayable":"100.00",              "ppCode": "M0028",              "dateTransaction":"2025-05-10T14:30:00",              "errorMessage": null,              "show":"Y",              "noticePaymentFlag":"Y"          }      ]  } |
| Response Failure | {      "data": {          "appCode": "OCMS-5000",          "message": "The request took too long to process. Please try again later."      }  } |

#### **Database and Data Mapping**

##### **Search Notice Parameter**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Zone** | **Database Table** | **Field Name** | **UI Field** | **Description** |
| Internet | eocms\_valid\_offence\_notice | vehicle\_no | Vehicle No | For search when input parameter vehicle\_no |
| Internet | eocms\_valid\_offence\_notice | notice\_no | Notice No | For search when input parameter notice\_no |
| PII | eocms\_offence\_notice\_owner\_driver | id\_no | NRIC/UEN/FIN | For search when input parameter id\_no, get the notice\_no and get the notice details from eVON |

##### **UI Data Mapping Non-Payable Fines**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Zone** | **Database Table** | **Field Name** | **UI Field** | **Description** |
| Internet | eocms\_valid\_offence\_notice | vehicle\_no | Vehicle No | For search when input parameter vehicle\_no |
| Internet | eocms\_valid\_offence\_notice | notice\_no | Notice No | For search when input parameter notice\_no |
| Internet | eocms\_valid\_offence\_notice | notice\_date\_and\_time | Date & Time Offence | Date and time of notice |

#### **Success Outcome**

* + - 1. The system queries the backend for the parking notice using the provided owner information and finds a valid offence notice.
      2. The system successfully retrieves all required offence notice and owner details from the backend databases.
      3. Notices can find and response to eService

#### **Error Handling**

##### **Application Error Handling**

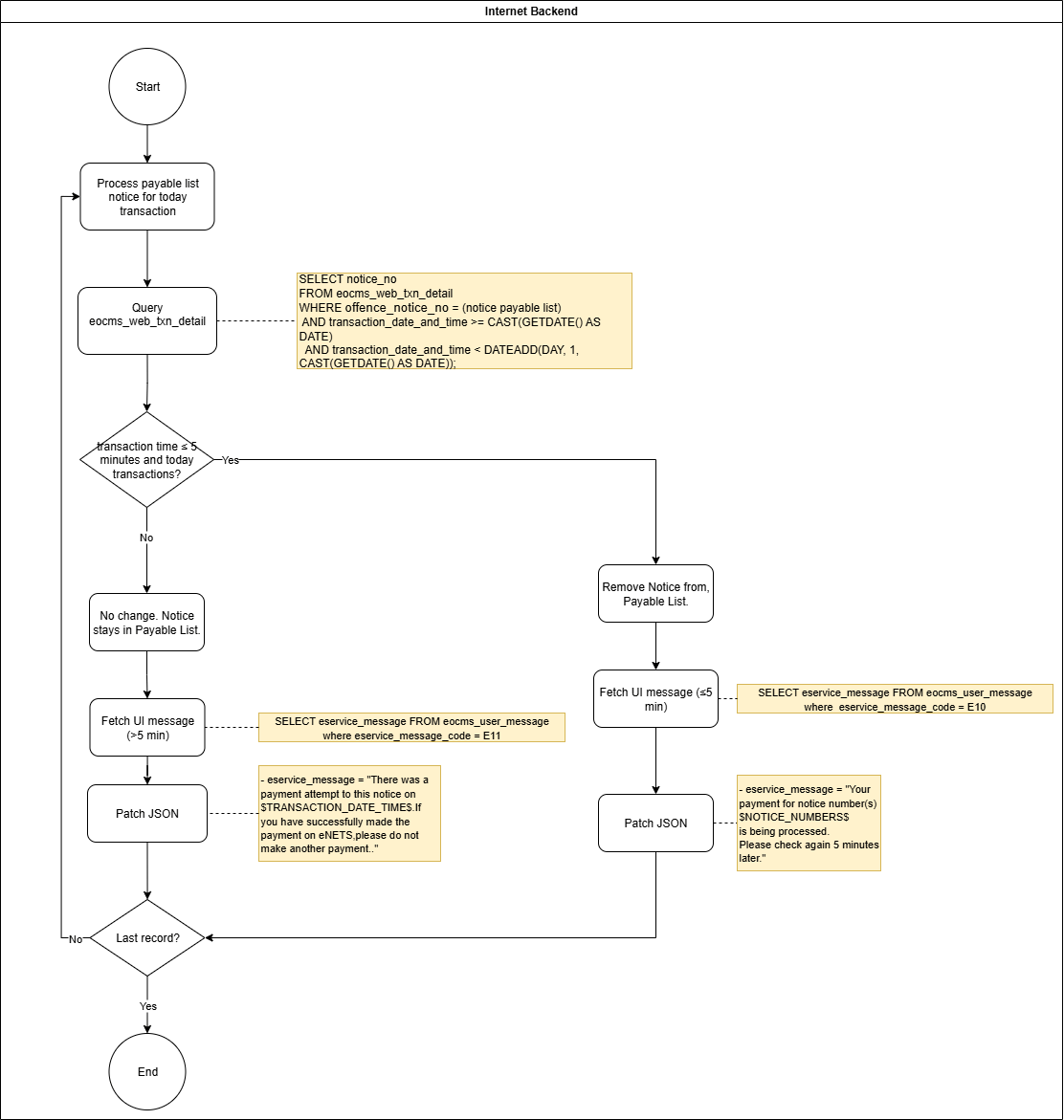
|  |  |  |
| --- | --- | --- |
| **Error Scenario** | **Definition** | **Brief Description** |
| SPCP Login Failed | Authentication Failure | User could not authenticate using SPCP, user will redirect to home pages. |
| No Result when search notices | Data Not Found | Query did not return any offence notices. Will return error messages to inform data not found |

##### **API Error Handling**

|  |  |  |  |
| --- | --- | --- | --- |
| **Error Scenario** | **App Error Code** | **User Message** | **Brief Description** |
| General Server Error | OCMS-5000 | Something went wrong on our end. Please try again later. | A generic error indicating the server encountered an unexpected condition. |
| Bad Request | OCMS-4000 | The request could not be processed due to a syntax error. Please check and try again. | The server could not understand the request due to invalid syntax. |
| Unauthorized Access | OCMS-4001 | You are not authorized to access this resource. Please log in and try again. | Authentication is required and has failed or has not been provided. |

### **Check 5 Mins Transaction**

#### **Diagram Flow Image**



NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Process payable list notice for today transaction | System Process | Starts processing each notice in the payable list to check for any payment transactions made today. |
| Query eocms\_web\_txn\_detail | Database Query | Runs a query to retrieve any transactions time <5 Mins and today transaction on web\_txn\_detail |
| Transaction time ≤ 5 minutes? | System Decision | Check transaction time ≤ 5 minutes or not |
| If No | System Decision | If No, the payment was made more than 5 minutes ago |
| No change. Notice stays in Payable List. | System Process | If the payment was made more than 5 minutes ago, keeps the notice in the payable list. |
| Fetch UI message (>5 min) | Database Query | Retrieves the UI message from eocms\_user\_message |
| Patch JSON | System Process | eservice\_message = "There was a payment attempt to this notice on $TRANSACTION\_DATE\_TIME$.If you have successfully made the payment on eNETS,please do not make another payment.." |
| If Yes | System Decision | If Yes, determines that the payment attempt is too recent (within 5 minutes). |
| Remove Notice from Payable List | System Process | The payment was within the last 5 minutes, removes the notice from the payable list and move to not payable list. |
| Fetch UI message (≤ 5 min) | Database Query | Retrieves the UI message from eocms\_user\_message. |
| Patch JSON | System Process | eservice\_message = "Your payment for notice number(s) $NOTICE\_NUMBERS$ is being processed. Please check again 5 minutes later." |
| Last record? | System Decision | Checks if this is the last notice in the payable list being processed. |
| If No | System Decision | Process payable list notice for today transaction |
| If Yes | System Decision | Finish check transaction time |
| End | Termination | Ends the transaction processing loop. |



#### **Database and Data Mapping**

##### **Get Today Transactions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Zone** | **Database Table** | **Field Name** | **Description** |
| Internet | eocms\_web\_txn\_detail | offence\_notice\_no | Reference to offence notice number |
| Internet | eocms\_web\_txn\_detail | transaction\_date\_and\_time | Date of the transaction |

##### **Get User Message**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Zone** | **Database Table** | **Field Name** | **Value** | **Description** |
| Internet | eocms\_user\_message | eservice\_message\_code | E11 | The unique identifier for representing a specific user-level error. |
| Internet | eocms\_user\_message | eservice\_message | There was a payment attempt to this notice on $TRANSACTION\_DATE\_TIME$.If you have successfully made the payment on eNETS,please do not make another payment. | A clear message for user about the error happen |

#### **Success Outcome**

1. Prevented duplicate payments within 5 minutes to avoid accidental multiple transactions.
2. Accurately displayed payment status for each notice (payable or not payable).
3. Provided clear and timely messages to users about their payment status and next steps.
4. Ensured the process completes only after checking all relevant notices, maintaining data integrity.

#### **Error Handling**

|  |  |  |
| --- | --- | --- |
| **Error Scenario** | **Definition** | **Brief Description** |
| Transaction timestamp missing | Payment transaction has no valid timestamp | A transaction record exists but the timestamp is null or invalid |
| Transaction time calculation error | Unable to calculate time difference correctly | System fails to determine if transaction was within the 5-minute window |
| Database query failure | Failure in fetching data from the database | System cannot retrieve transactions or messages due to DB errors |
| UI message retrieval failure | Unable to get appropriate user message | The system cannot fetch or generate the message to display to the user |
| JSON patch/update failure | Failure to update the JSON response | System encounters an error when modifying the JSON with new info |

### **Validate Captcha Search using Vehicle Number or Notice Number**

#### **Diagram Flow Image**

A screenshot of a computer

AI-generated content may be incorrect.

NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Definition** | **Brief Description** | |
| Show Turnstile widget | UI Rendering | Client renders the Turnstile widget. | |
| User validate turnstile | User Action | User completes the Turnstile challenge. | |
| Receive token | System Interaction | Browser receives Turnstile token from widget. | |
| Submit token | API Call | Client sends POST /captcha/v1/validate with  {  "token": "<XXXXXXX>"  } | |
| Call Turnstile verify | System Interaction | Backend calls Turnstile verify endpoint | |
| Success? | Decision | If Turnstile says the token is valid, proceed; otherwise handle error. | |
| Success response | System Interaction | Backend returns  {  appCode: "OCMS-2000",  message: "Success"  }  to the UI. | |
| Display success | UI Feedback | UI shows success message and allows the user to continue the search process. | |
| Error response | System Interaction | If invalid, backend returns  {  appCode: "OCMS-4000",  message: "Captcha not correct. Please try again."  }. | |
| Display error | UI Feedback | UI shows an error pop-up with the message. | |
| User confirms pop-up | User Action | User acknowledges the error. | |
| Refresh page | UI Action | Page refreshes (widget reloads and a new token will be generated). | |
| End | Termination | Flow ends (either after success message or after refresh awaiting a new attempt). | |

#### **API Specification**

##### **API to Validate Captcha**

|  |  |
| --- | --- |
| API Name | Validate captcha |
| URL | UAT: https://parking2.uraz.gov.sg/ocms/captcha/v1/validate  PRD: https://parking.uraz.gov.sg/ocms/captcha/v1/validate |
| Description | The API to validate captcha |
| Method | POST |
| Header | {  "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  "Content-Type": "application/json"  } |
| Payload | {     "token": "XXXXXX"  } |
| Response | {  “data”: {  "appCode": "OCMS-2000",  "message": "Success"  }  } |
| Response Failure | {      "data": {          "appCode": "OCMS-4000",          "message": "Captcha not correct. Please try again."      }  } |

#### **Success Outcome**

1. Turnstile loads successfully and generates a one-time token for the session.
2. User completes the widget and submits the form with the generated token.
3. eService success backend calls Turnstile /siteverify with secret + token.
4. Turnstile confirms validity, backend returns {appCode: "OCMS-2000", message: "Success"}.
5. UI shows a success message and allows the user to proceed with the search process.

#### **Error Handling**

|  |  |  |
| --- | --- | --- |
| **Error Scenario** | **Definition** | **Brief Description** |
| Invalid token | System Interaction | Backend receives a failed response from Turnstile /siteverify. |
| Error response | System Interaction | Backend returns {appCode: "OCMS-4000", message: "Captcha not correct. Please try again."}. |
| Display error | UI Feedback | UI displays an error pop-up showing the failure message. |
| User confirms pop-up | User Action | User acknowledges the error message by confirming/closing the pop-up. |
| Refresh page | UI Action | Page reloads, widget re-renders, and a new token is generated for retry. |

# **Section 2 – Send outstanding parking offence notices to the payment channels**

### **Use Case**

1. When the Pay for Parking Fines eService or AXS sends a backend request to search for notices based on the motorist’s search parameter, OCMS validates the search results against a pre-defined Payment Matrix before returning the search results.
2. The Payment Matrix is a standard checklist of rules that determines:
   1. Whether a notice can be displayed on the payment portal
   2. Whether a notice is payable
   3. The corresponding user message that the eService portal should display based on the notice’s

attributes

1. The Matrix will be implemented as a new table in the OCMS Internet database.
2. Each row in the matrix maps a unique combination of attributes to corresponding outcomes that guide the system’s display logic, payment eligibility validation, and messages to end-users.

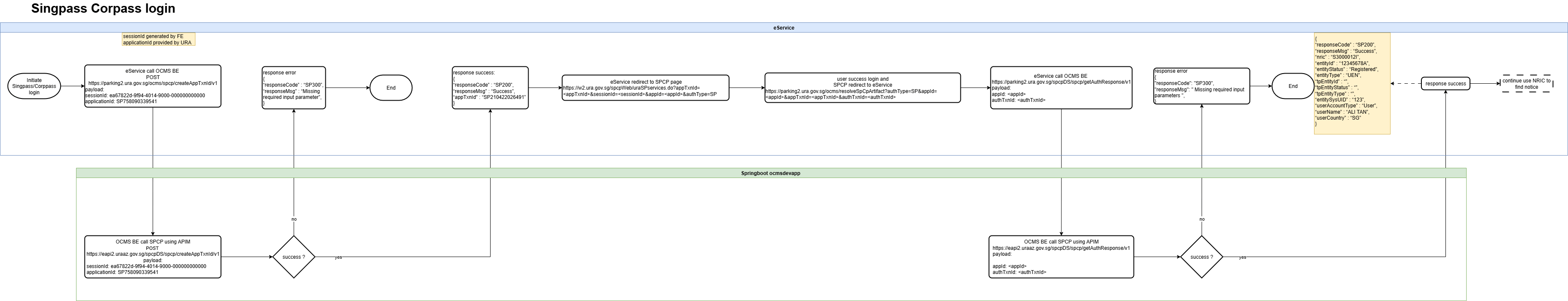






### **Singpass Corppass Login**

#### **Diagram Flow Image**

NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Initiate Singpass/Corppass login | Start | Offender will click Login buttons images for Individual(SP) / Business(CP) Users |
| eService call OCMS BE — POST | Request from eService to OCMS BE | eService will call OCMS BE |
| OCMS BE call SPCP using APIM — POST | OCMS BE → SPCP via APIM | Than OCMS BE will call SPCP |
| success ? | Decision | Branch on result of createAppTxnId. |
| response error | Error path | {  "responseCode": "SP300",  "responseMsg": "Missing required input parameter"  } → End. |
| response success | Success path | {  "responseCode": "SP200",  "responseMsg": "Success",  "appTxnId": "SP210422026491"  }. |
| eService redirect to SPCP page | Redirect URL | https://w2.ura.gov.sg/spcpWeb/uraSPservices.do?appTxnId=<appTxnId>&sessionId=<sessionId>&appId=<appId>&authType={SP/CP}. |
| user success login and SPCP redirect to eService | Callback to eService | https://parking2.ura.gov.sg/ocms/resolveSpCpArtifact?authType={SP/CP}&appId=<appId>&appTxnId=<appTxnId>&authTxnId=<authTxnId>. |
| eService call OCMS BE | Request from eService to OCMS BE | https://parking2.ura.gov.sg/spcpDS/spcp/getAuthResponse/v1  Payload: appId: <appId>, authTxnId: <authTxnId>. |
| OCMS BE call SPCP using APIM | OCMS BE → SPCP via APIM | https://eapi2.uraaz.gov.sg/spcpDS/spcp/getAuthResponse/v1  Payload: appId: <appId>, authTxnId: <authTxnId>. |
| success ? | Decision | Branch on result of getAuthResponse. |
| response error | Error path | {  "responseCode": "SP300",  "responseMsg": " Missing required input parameters "  } → End. |
| response success | Success path (sample payload shown) | Yellow note shows example: responseCode SP200, responseMsg Success, and identity/user fields |
| continue use NRIC to find outstanding notice |  |  |



#### **API Specification**

##### **API for eService**

###### **Generate app txn id**

|  |  |
| --- | --- |
| API Name | Gemerate app txn id |
| URL | UAT : https://parking2.ura.gov.sg/ocms/spcp/createAppTxnId/v1  PRD : https://parking.ura.gov.sg/ocms/spcp/createAppTxnId/v1 |
| Description | The API to generate app txn id from SPCP |
| Method | POST |
| Query Parameter | sessionId=<sessionId>  appId=<appId> |
| Header | {  "Authorization": "Bearer [AUTH\_TOKEN]",  "Content-Type": "application/json"  } |
| Response | {  “responseCode” : “SP200”,  “responseMsg” : “Success”,  “appTxnId” : “SP210422026491”  } |
| Response Failure | {  “responseCode” : “SP300”,  “responseMsg” : “Missing required input parameter”,  } |

###### **Redirect to SPCP page**

|  |  |
| --- | --- |
| API Name | SPCP page |
| URL | UAT : https://w2.ura.gov.sg/spcpWeb/uraSPservices.do  PRD : https://w.ura.gov.sg/spcpWeb/uraCPservices.do |
| Description | The endpoint to redirect user to SPCP login page |
| Query Parameter | appTxnId=<appTxnId>  sessionId=<sessionId>  appId=<appId>  authType=SP |

###### **Get user information from SPCP**

|  |  |
| --- | --- |
| API Name | Get user SPCP information |
| URL | UAT : https://parking2.ura.gov.sg/ocms/spcp/getAuthResponse/v1  PRD : https://parking.ura.gov.sg/ocms/spcp/getAuthResponse/v1 |
| Description | The API to get SPCP user detail information |
| Method | POST |
| Header | {  "Authorization": "Bearer [AUTH\_TOKEN]",  "Content-Type": "application/json"  } |
| Payload | {  “appId”: <appId>,  “authTxnId”: <authTxnId>  } |
| Response | { “responseCode” : “SP200”, “responseMsg” : “Success”, “nric” : “S3000012I”, “entityId” : “12345678A”, “entityType” : “UEN”, } |
| Response Failure | {  “responseCode” : “SP300”,  “responseMsg” : “Missing required input parameter”,  } |

##### **API consume**

###### **SPCP create app txn id**

|  |  |
| --- | --- |
| API Name | Get user SPCP information |
| URL | UAT : https://eapi2.uraaz.gov.sg/spcpDS/spcp/createAppTxnId/v1  PRD : https://eapi.uraaz.gov.sg/spcpDS/spcp/createAppTxnId/v1 |
| Description | API Generate app txn id |
| Method | POST |
| Request parameters | “sessionId”: <authTxnId>  “appId”: <appId> |
| Response | {  “responseCode” : “SP200”,  “responseMsg” : “Success”,  “appTxnId” : “SP210422026491”  } |
| Response Failure | {  “responseCode” : “SP300”,  “responseMsg” : “Missing required input parameter”,  } |

###### **Get Authentication Response**

|  |  |
| --- | --- |
| API Name | Get user SPCP information |
| URL | UAT : https://eapi2.uraaz.gov.sg/spcpDS/spcp/getAuthResponse/v1  PRD : https://eapi2.uraaz.gov.sg/spcpDS/spcp/getAuthResponse/v1 |
| Description | The API to get SPCP user detail information |
| Method | POST |
| Request Parameters | “appId”: <appId>,  “authTxnId”: <authTxnId> |
| Response | { “responseCode” : “SP200”, “responseMsg” : “Success”, “nric” : “S3000012I”, “entityId” : “12345678A”, “entityStatus” : “Registered”, “entityType” : “UEN”, “tpEntityId” : “”, “tpEntityStatus” : “”, “tpEntityType” : “”, “entitySysUID” : “123”, “userAccountType” : “User”, “userName” : “ALI TAN”, “userCountry” : “SG” } |
| Response Failure | {  “responseCode” : “SP300”,  “responseMsg” : “Missing required input parameter”,  } |



#### **Data Mapping**

##### **createAppTxnId**

|  |  |
| --- | --- |
| **Payload** | **Description** |
| sessionId | Generated by FE |
| applicationId | Get from URA SPCP when register the application |

##### **getAuthResponse**

|  |  |
| --- | --- |
| **Payload** | **Description** |
| appId | applicationId |
| auhTxnId | appTxnId from createAppTxnId response |

#### **Successful Outcome**

1. createAppTxnId succeeds → eService receives: { "responseCode": "SP200", "responseMsg": "Success", "appTxnId": "SP210422026491" }.
2. eService redirects to SPCP page using: appTxnId, sessionId, appId, authType=SP/CP.
3. User successfully logs in and SPCP redirects back to eService callback with: authType=SP/CP, appId, appTxnId, authTxnId.
4. eService calls /spcpDS/spcp/getAuthResponse (with appId, authTxnId) and OCMS BE calls /spcpDS/spcp/getAuthResponse/v1.
5. getAuthResponse succeeds → eService receives success payload with "responseCode":"SP200","responseMsg":"Success" and identity/user fields.

#### **Error Handling**

##### **createAppTxnId**

|  |  |
| --- | --- |
| **Response Code** | **Response Message** |
| SP200 | Success |
| SP300 | Missing required input parameters. |
| SP301 | Application transaction id already created for the session id. |
| SP302 | Application Id is not a registered id. |
| SP303 | Fail to create transaction record. |
| SP502 | System Error |

##### **getAuthResponse**

|  |  |
| --- | --- |
| **Response Code** | **Response Message** |
| SP200 | Success |
| SP300 | Missing required input parameters. |
| SP302 | Missing configuration - application Id is not a registered id |
| SP304 | There is no response record for the specified transaction id |
| SP502 | System Error |

### **Preset eservice\_message\_code**

#### **Diagram Flow Image**

A screenshot of a diagram

AI-generated content may be incorrect.

NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Definition** | | **Brief Description** |
| **Create Notice Type O and E** | | | |
| Create notice | System Interaction | | Create the notice; set von\_payment\_acceptance\_allowed = Y (ref: OCMS 5). |
| Detect vehicle reg type | Decision | | Check If veh\_reg\_type = F |
| (If Yes) set message | System Interaction | | For veh\_reg\_type = F, set von\_service\_message\_code = E6 |
| Notice created | Result | | Notice created |
| If No (S, I D, or V) | System Interaction | | Process to next step |
| ANS check | System Interaction | | Refer to Technical Document OCMS 10 |
| an\_flag = Y ? | Decision | | If Yes, set payment acceptance to N  If No set service message E1. |
| Notice created | Result | | Notice created (ANS branch). |
| End | Termination | | Flow ends. |
| **Create Notice Type U** | | | |
| Create notice | System Interaction | | Refer to Technical Document OCMS 5 |
| Detect vehicle reg type | Decision | | Check If veh\_reg\_type = F |
| (If Yes) set message | System Interaction | | For veh\_reg\_type = F, set von\_service\_message\_code = E6 |
| Notice created | Result | | Notice created |
| If No (S, I D, or V) | System Interaction | | Process to next step |
| Notice created | Result | | Notice created. |
| Detect vehicle reg type (post-creation) | System Interaction | | Refer to Technical Document OCMS 13 |
| OIC decide compoundable? | Decision | | If Yes, set von.payment\_  acceptance\_allowed = Y  If No set E4 and disallow payment. |
| **~~Change Stage end of RR3 (go to CPC/CFC~~)**  **Court Processing** | | | |
| Cron processing | Initiate | | Cron processing run |
| Change stage | System Interaction | | Notice will change stage |
| Stage inside list can’t pay? | Decision | | Change stage to court list stage can’t pay  set von.eservice\_message\_code = E14  and  von.payment\_acceptance\_allowed = N |
| **Apply Permanent Suspend** | | | |
| Apply PS | System Interaction | | Refer to Technical Document OCMS 18 |
| Disallow payment | System Interaction | | Set von\_payment\_acceptance\_allowed = N. |
| Any revival applied? | Decision | | If Yes, revival applied  If No, do nothing |
| ~~Revival applied~~ | ~~System Interaction~~ | | ~~Set von\_payment\_acceptance\_allowed = Y.~~ |
| Check any other active suspension | Decision | | If yes, do nothing.  If no, set von.eservice\_message\_code = E6  and  von.payment\_acceptance\_allowed = Y |
| End | Termination | | Flow ends. |











#### **Data Mapping**

|  |  |  |  |
| --- | --- | --- | --- |
| **Zone** | **Database Table** | **Field Name** | **Description** |
| Internet | eocms\_user\_message | eservice\_message\_code | For join result from payment matrix |
| Internet | eocms \_valid\_offence\_notice | payment\_acceptance\_allowed | Indicates if payment is acceptable |

#### **Success Outcome**

1. Create Notice – offence type O & E

* On create, von.payment\_acceptance\_allowed = Y.

1. Create Notice – offence type U

* On create, von.payment\_acceptance\_allowed = Y.
* After creation, OIC decides compoundable?
  + Yes → confirm von.payment\_acceptance\_allowed = Y.
  + No → set von.eservice\_message\_code = E4 and von.payment\_acceptance\_allowed = N.

1. Change stage – end of RR3 (→ CPC/CFC)

* On RR3 expiry cron → set von.eservice\_message\_code = E4 and von.payment\_acceptance\_allowed = N.
* If a payment activation exists → flip to von.eservice\_message\_code = E1 and von.payment\_acceptance\_allowed = Y.

1. Apply Permanent Suspend (PS)

* On PS apply → von.payment\_acceptance\_allowed = N.
* If revival later applied → reset von.payment\_acceptance\_allowed = Y.

#### **Error Handling**

|  |  |  |
| --- | --- | --- |
| **Error Scenario** | **Definition** | **Brief Description** |
| DB write/transaction error | Any update to VON fields fails | Abort current branch; return error to caller/cron; retry with idempotency. |

### **Sample Data User Message**

|  |  |  |
| --- | --- | --- |
| **eService message Code** | **eService Message** | **Status** |
| E1 | Pay your parking fine(s) now to avoid higher penalties and court action. (Notice number: $NOTICE\_NUMBERS$) | Current |
| E2 | There are other outstanding notices/court summonses against the vehicle. To check if the notices/court summonses are against you, please search using your ID number. Alternatively, you may chat with us or call us at 6329 3434 for advice. | Current |
| E4 | Parking fine amount is not available yet. Please chat with us or call us at 6329 3434 for advice. (Notice number: $NOTICE\_NUMBERS$) | Current |
| E5 | This is an Advisory Notice. No payment is required. | Current |
| E6 | Pay your parking fine(s) now to avoid your vehicle being towed or clamped if the offence is not settled. (Notice number: $NOTICE\_NUMBERS$) | Current |
| E10 | Your payment for notice number(s) $NOTICE\_NUMBERS$ is being processed.  Please check again 5 minutes later. | Current |
| E11 | There was a payment attempt to this notice on $TRANSACTION\_DATE\_TIME$.  If you have successfully made the payment on eNETS,  please do not make another payment. | Current |
| E12 | There is no requirement payment parking offence for the notice number xxxx, as of xxxx (date) already paid with reduced amount  note: waiting stanley, still KIV | New |
| E13 | Please make payment for the notice before [DD MMM YYYY] *(2 weeks before Court date)*. After the payment due date, you will have to settle the summons at a higher fine before [DD MMM YYYY, 5pm] *(Court date, 5pm)* .Once payment is made, you do not need to attend State Court 4AN on [DD MMM YYYY]. | New |
| E14 | The notice is no longer payable. Please attend court on the given court date to make your plea. If you have missed your court date, please report to the Warrant Enforcement Unit (WEU) to obtain a new court date. | New |
| E15 | The fine amount is not payable yet. The vehicle owner is required to furnish the driver's particulars to us using the Furnish Driver's Particulars e-Service. | New |
| E16 | There is no outstanding parking offence for the ID $ID\_NUMBER$, as of $CURRENT\_DATE$. If you wish to check or pay for a particular notice number, you may click here. | New |
| E17 | There is no outstanding parking offence for the notice $NOTICE\_NUMBERS$, as of $CURRENT\_DATE$. | New |
| E18 | There is no outstanding parking offence for the vehicle number $VEHICLE\_NUMBERS$, as of $CURRENT\_DATE$ (date). If you wish to check or pay for a particular notice number, you may click here. | New |

# **Section 3 – Outstanding Notice Search and Payment through AXS**

### **Use Case**

* 1. OCMS shall enable the retrieval of outstanding Parking Offence Notices via AXS to facilitate payment by motorists, covering both payable and not-payable notices with the appropriate display of payment eligibility and guidance.
  2. To support this use case, OCMS shall maintain timely and accurate synchronisation of Parking Offence Notices and payment statuses between the Intranet and Internet Zones.
  3. The synchronisation shall work by:

Allowing AXS to query OCMS for outstanding notices using approved search criteria.

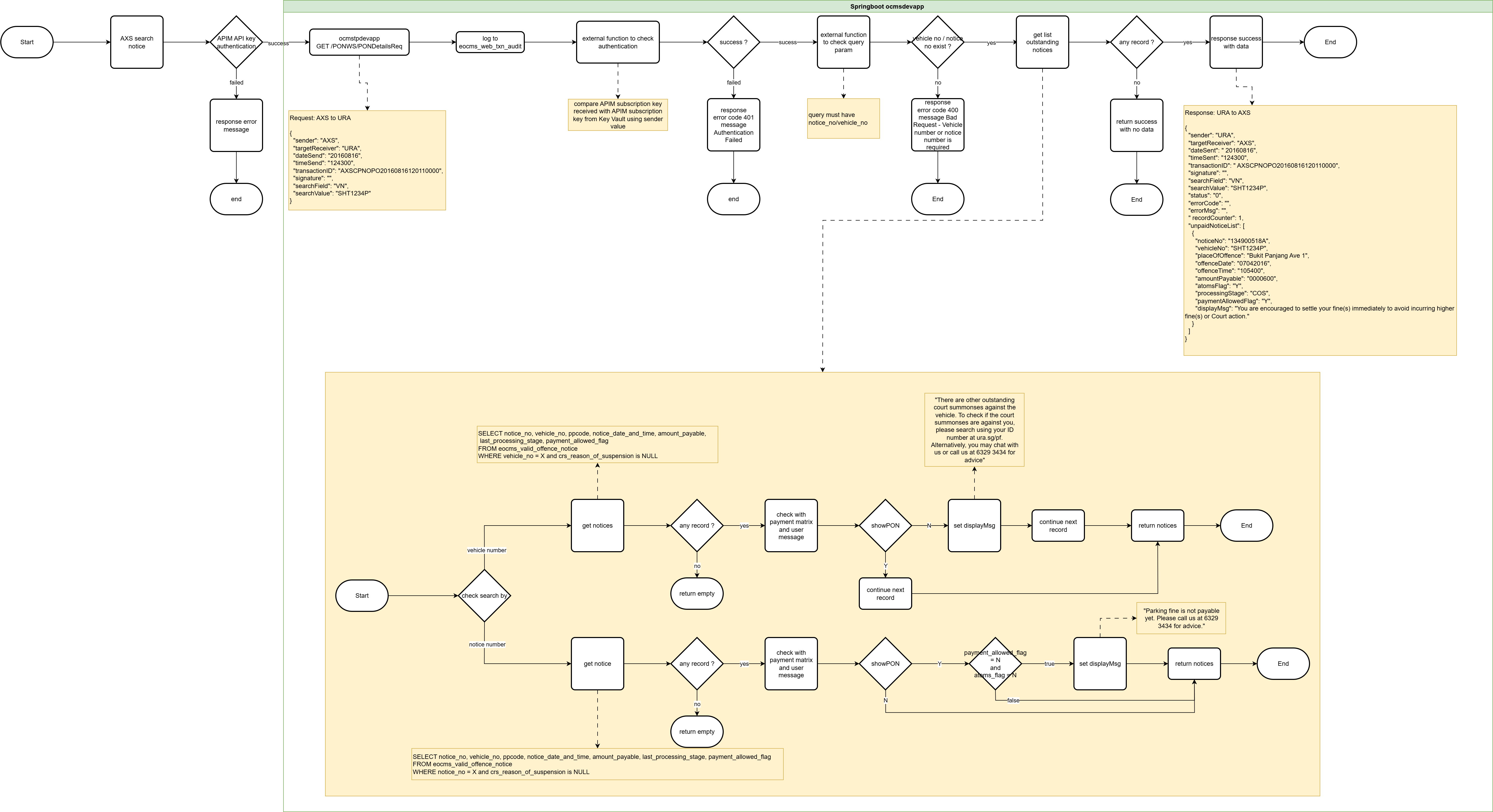
Applying a set of business rules in OCMS to determine whether each notice is payable or not payable, and returning the appropriate result and guidance.

Updating the payment status in OCMS promptly upon receiving confirmation from the Payment Gateway when a notice has been paid through AXS.

* 1. This process shall ensure that the information available to AXS is always current, enabling motorists to view accurate offence details, amounts payable, and payment eligibility.
  2. OCMS shall ensure that all payments made via AXS are recorded accurately with key transaction details such as amount paid, payment date, receipt number, payment channel, and notice reference.
  3. OCMS shall maintain appropriate security and integrity controls to ensure data accuracy, prevent duplicate payment updates, and protect communication between systems.

### **Get Outstanding Notice**

#### **Diagram Flow Image**



NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Start | Start | Entry point of the process. |
| AXS search notice | Search initiation | AXS initiates a notice search request. |
| APIM API key authentication | Authentication check | Validates the API key; failure returns error message and ends process; success continues. |
| ocmstpdevapp GET /PONWS/PONDetailsReq | Service request | Sends request to endpoint, logs transaction to eocms\_web\_txn\_audit. |
| External function to check authentication | Key verification | Compares APIM subscription key received with Key Vault subscription key based on sender value. |
| Success? | Authentication decision | If authentication fails, returns 401 Authentication Failed; if successful, proceeds. |
| External function to check query param | Parameter validation | Checks if query has notice\_no or vehicle\_no. |
| Notice no / vehicle no exist? | Query completeness check | If missing, returns 400 Bad Request; if present, proceeds. |
| Get list outstanding notices | Retrieve data | Queries outstanding notices for the search input. |
| Any record? | Record check | If no records, returns success with no data; if found, proceeds. |
| Response success with data | Response delivery | Returns result data to AXS and ends process. |

Get list outstanding notices using vehicle number

|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Start | Entry point | Starts vehicle-number search path. |
| Check search by → vehicle number | Search routing | Chooses search type. |
| Get notices | Notice retrieval | Runs SQL to select matching notices with vehicle\_no and no suspension reason. |
| Any record? | Record check | If no records, returns empty; otherwise continues. |
| Check with payment matrix and user message | Payment validation | Evaluates payment rules and prepares user messages. (refer to section 2.4) |
| showPON = Y | Display setup | Prepares the PON display details. |
| showPON = N | Display setup | Override the displayMsg value |
| Continue next record | Loop handler | Moves to the next notice record. |
| Return notices | Data return | Sends compiled notice data back. |
| End | Exit | Ends branch. |

Get list outstanding notices using notice number

|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Start | Entry point | Starts notice-number search path. |
| Check search by → notice number | Search routing | Chooses search type. |
| Get notice | Notice retrieval | Runs SQL to select the notice matching notice\_no with no suspension reason. |
| Any record? | Record check | If no records, returns empty; otherwise continues. |
| Check with payment matrix and user message | Payment validation | Evaluates payment rules and prepares user messages. (refer to section 2.4) |
| showPON = N | Display setup | Prepares the PON display details. |
| showPON = Y | Check Payment allowed & atoms flag check | If true = override displayMsg  If false = Prepares the PON display details. |
| Return notices | Data return | Sends notice data back. |
| End | Exit | Ends branch. |



#### **API Specification**

##### **API Provide**

|  |  |
| --- | --- |
| API Name | Get outstanding notices |
| URL | UAT : https://parking2.uraz.gov.sg/ocms/PONWS/v1/PONDetailsReq  PRD : https://parking.uraz.gov.sg/ocms/PONWS/v1/PONDetailsReq |
| Description | The API to get outstanding offence notice information |
| Method | GET |
| Header | {  "Content-Type": "application/json",  “Ocp-Apim-Subscription-Key”: “[APIM secret value]”  } |
| Payload | {    "sender": "AXS",    "targetReceiver": "URA",    "dateSend": "20160816",    "timeSend": "124300",    "transactionID": "AXSCPNOPO20160816120110000",    "signature": "",    "searchField": "VN",    "searchValue": "SHT1234P"  } |
| Response | {  "sender": "URA",  "targetReceiver": "AXS",  "dateSent": " 20160816",  "timeSent": "124300",  "transactionID": " AXSCPNOPO20160816120110000",  "signature": "",  "searchField": "VN",  "searchValue": "SHT1234P",  "status": "0",  "errorCode": "",  "errorMsg": "",  " recordCounter": 1,  "unpaidNoticeList": [  {  "noticeNo": "134900518A",  "vehicleNo": "SHT1234P",  "placeOfOffence": "Bukit Panjang Ave 1",  "offenceDate": "07042016",  "offenceTime": "105400",  "amountPayable": "0000600",  "atomsFlag": "Y",  "processingStage": "COS",  "paymentAllowedFlag": "Y",  "displayMsg": "You are encouraged to settle your fine(s) immediately to avoid incurring higher fine(s) or Court action."  }  ]  } |
| Response Failure | {  "sender": "URA",  "targetReceiver": "AXS",  "dateSent": " 20160816",  "timeSent": "124300",  "transactionID": " AXSCPNOPO20160816120110000",  "signature": "",  "searchField": "VN",  "searchValue": "SHT1234P",  "status": "0",  "errorCode": "",  "errorMsg": "",  " recordCounter": 1,  "unpaidNoticeList": []  } |

#### 

#### **Data Mapping**

|  |  |
| --- | --- |
| **Response** | **Description** |
| searchField | VN(vehicle number)/NN(notice number) |
| searchValue | Get from request payload |
| recordCounter | Sum list of payable notices |
| noticeNo | von.notice\_no |
| vehicleNo | von.vehicle\_no |
| placeOfOffence | von.pp\_name |
| offenceDate | von.notice\_date\_and\_time |
| offenceTime | von.notice\_date\_and\_time |
| amountPayable | von.amount\_payable |
| atomsFlag |  |
| processingStage | von.last\_processing\_stage |
| paymentAllowedFlag | ~~von.payment\_allowed\_flag~~ von.payment\_accepted\_allowed |
| displayMsg | eocms\_user\_message.eservice\_message |

#### **Success Outcome**

1. Valid API key authentication
2. Authentication check passed
3. Valid query parameters provided
4. Records found for given vehicle number
5. Records found for given notice number
6. Response with data returned to AXS

#### **Error Handling**

##### **Application Error Handling**

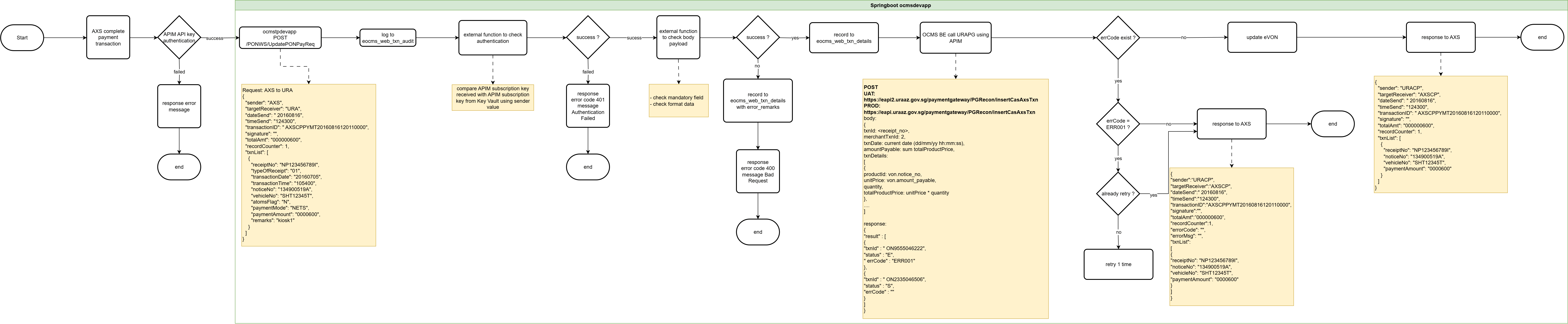
|  |  |  |
| --- | --- | --- |
| **Error Scenario** | **Definition** | **Brief Description** |
| API key authentication failed | APIM API key authentication | Returns a response error message and ends the flow when the API key validation fails. |
| Authentication check failed | External function to check authentication → decision “success?” = No | Returns response error code 401 – Authentication Failed and ends the process. |
| Missing required query parameter | External function to check query param → “notice\_no / vehicle\_no exist?” = No | Returns response error code 400 – Bad Request (vehicle number or notice number is required) and ends the process. |
| No records found (vehicle number path) | Decision “any record?” = No after Get notices | Returns empty response and ends the process. |
| No records found (notice number path) | Decision “any record?” = No after Get notice | Returns empty response and ends the process. |
| Payment not allowed & not in ATOMS | At showPON when payment\_allowed\_flag = N and atoms\_flag = N | Sets displayMsg: “Parking fine is not payable yet. Please call us at 6329 3434 for advice.” Then returns notices. |
| Other outstanding court summons | At showPON (vehicle number path) when court-summons condition triggers | Sets displayMsg advising to search by ID at ura.sg/pf or call 6329 3434; continues processing remaining records. |

##### **API Error Specification**

|  |  |
| --- | --- |
| **Error Code (For Fail Status)** | |
| 400 | Bad Request |
| 401 | Unauthorized |
| 500 | Internal Server Error |

### **Payment Transaction**

#### **Diagram Flow Image**



NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| AXS complete payment transmission | Payment request initiation | AXS sends a payment completion transmission to the service. |
| APIM API key authentication | API key validation | Validates the API key; failure returns an error message and ends process; success continues. |
| /PONWS/UpdatePONPayReq request | Payment update request | Sends payment update request to ocmsdevapp endpoint; logs request to eocms\_web\_txn\_audit. |
| External authentication check | Subscription key verification | Compares APIM subscription key with Key Vault subscription key using sender value. |
| Authentication success? | Authentication decision | If failed, returns 401 Authentication Failed; if successful, continues. |
| External body payload check | Request data validation | Verifies all mandatory fields and field formats according to specification. |
| Payload check success? | Payload validation decision | If failed, logs request with error remarks in eocms\_web\_txn\_details, returns 400 Bad Request; if successful, continues. |
| Record to eocms\_web\_txn\_details | Transaction detail logging | Records request details without error remarks. |
| OCMS BE call to URA PG | External payment gateway call | Sends POST request to URA PG API (insertCasAxsTxn) via APIM. |
| errCode exist? | URA response evaluation | Checks if URA response contains an error code. |
| errCode = ERR001? | Specific error handling | If ERR001, proceed to retry logic; if other error, return error response to AXS and end. |
| Already retry? | Retry attempt check | If no prior retry, wait 1 minute and retry once; if already retried, return error response to AXS and end. |
| Update eVON | Data update | Updates eVON record after successful URA processing. |
| Response to AXS (success) | Final acknowledgment | Sends final success response with payment details to AXS. |
| End | End | Terminates the payment completion process. |



#### **API Specification**

##### **API Provide**

|  |  |
| --- | --- |
| API Name | AXS Payment Transaction |
| URL | UAT : https://parking2.uraz.gov.sg/ocms/PONWS/v1/UpdatePONPayReq  PRD : https://parking.uraz.gov.sg/ocms/PONWS/v1/UpdatePONPayReq |
| Description | The API to do the payment transaction outstanding offence notice |
| Method | POST |
| Header | {  "Content-Type": "application/json",  “Ocp-Apim-Subscription-Key”: “[APIM secret value]”  } |
| Payload | {    "sender": "AXS",    "targetReceiver": "URA",    "dateSend": " 20160816",    "timeSend": "124300",    "transactionID": " AXSCPPYMT20160816120110000",    "signature": "",    "totalAmt": "000000600",    "recordCounter": 1,    "txnList": [      {        "receiptNo": "NP123456789I",        "typeOfReceipt": "01",        "transactionDate": "20160705",        "transactionTime": "105400",        "noticeNo": "134900519A",        "vehicleNo": "SHT12345T",        "atomsFlag": "N",        "paymentMode": "NETS",        "paymentAmount": "0000600",        "remarks": "kiosk1"      }    ]  } |
| Response | {  "sender": "URA",  "targetReceiver": "AXS",  "dateSend": " 20160816",  "timeSend": "124300",  "transactionID": " AXSCPPYMT20160816120110000",  "signature": "",  "totalAmt": "000000600",  "recordCounter": 1,  "txnList": [  {  "receiptNo": "NP123456789I",  "noticeNo": "134900519A",  "vehicleNo": "SHT12345T",  "paymentAmount": "0000600"  }  ]  } |
| Response Failure | {  "sender": "URACP",  "targetReceiver": "AXSCP",  "dateSend": "20160816",  "timeSend": "124300",  "transactionID": "AXSCPPYMT20160816120110000",  "signature": "",  "totalAmt": "000000600",  "recordCounter": 1,  "errorCode": "",  "errorMsg": "",  "txnList": [  {  "receiptNo": "NP123456789I",  "noticeNo": "134900519A",  "vehicleNo": "SHT12345T",  "paymentAmount": "0000600"  }  ]  } |

##### **API Consume**

|  |  |
| --- | --- |
| API Name | URAPG Payment Transaction for AXS |
| URL | UAT: https://eparking2.uraz.gov.sg/paymentgateway/PGRecon/insertCasAxsTxn  PROD: https://eparking.uraz.gov.sg/paymentgateway/PGRecon/insertCasAxsTxn |
| Description | The API to do the payment transaction outstanding offence notice to URAPG |
| Method | POST |
| Header | {  "Content-Type": "application/json",  “Ocp-Apim-Subscription-Key”: “[APIM secret value]”  } |
| Payload | {      "txnId": "<receipt\_no>",      "merchantTxnId": 2,      "txnDate": "current date (dd/mm/yy hh:mm:ss)",      "amountPayable": "sum totalProductPrice",      "txnDetails": [          {              "productId": "von.notice\_no",              "unitPrice": "von.amount\_payable",              "quantity": "quantity",              "totalProductPrice": "unitPrice \* quantity"          }      ]  } |
| Response | {  "result": [  {  "txnId": "ON2335046506",  "status": "S",  "errCode": ""  }  ]  } |
| Response Failure | {  "result": [  {  "txnId": "ON9555046222",  "status": "E",  "errCode": "ERR001"  }  ]  } |

#### **Data Mapping**

##### **AXS Payment**

|  |  |
| --- | --- |
| **Response** | **Description** |
| Sender | URA |
| targetReceiver | AXS |
| dateSend | Current date |
| timeSend | Current time |
| transactionID | Take from request payload |
| totalAmt | Sum from txnList paymentAmount |
| recordCounter | Total list data from txnList |
| receiptNo | Take from request payload |
| noticeNo | Take from request payload |
| vehicleNo | Take from request payload |
| paymentAmount | Take from request payload |

##### **URAPG Payment Transaction for AXS**

|  |  |
| --- | --- |
| **Response** | **Description** |
| txnId | eocms\_web\_txn\_details.receipt\_no |
| merchantTxnId | “2” |
| txnDate | eocms\_web\_txn\_details.transaction\_date\_and\_time |
| amountPayable | Total amount paid |
| productId | eocms\_web\_txn\_details.offence\_notice\_no |
| unitPrice | eocms\_web\_txn\_details.payment\_amount |
| quantity | 1 |
| totalProductPrice | unitPrice \* quantity |

#### **Success Outcome**

* + 1. API key authentication passes
    2. Subscription key verification succeeds
    3. Request body passes all mandatory field and format checks
    4. URA insertCasAxsTxn returns with no errCode
    5. ERR001 occurs once but succeeds on the single retry after 1 minute
    6. Transaction details recorded to eocms\_web\_txn\_details (no error remarks)
    7. eVON is updated successfully
    8. Final success response with payment details is returned to AXS

#### **Error Handling**

##### **Application Error Handling**

|  |  |  |
| --- | --- | --- |
| **Error Scenario** | **Definition** | **Brief Description** |
| API key authentication failed | APIM API key authentication | Return a response error message and end the flow. |
| Subscription key verification failed | External function to check authentication → decision “success?” = No | Return 401 Authentication Failed and end. |
| Body payload validation failed | External function to check body payload → decision “success?” = No | Record request to eocms\_web\_txn\_details with error\_remarks, return 400 Bad Request, and end. |
| URA PG response contains non-ERR001 error | After URA insertCasAxsTxn call, errCode exists and is not ERR001 | Return error response to AXS and end. |
| URA PG response ERR001 after retry exhausted | errCode = ERR001 and the request was already retried | Return error response to AXS and end. |
| URA PG response ERR001 on first attempt | errCode = ERR001 and not yet retried | Wait 1 minute, retry once, then re-evaluate outcome. |

##### **API Error Specification**

###### **AXS Payment transaction**

|  |  |
| --- | --- |
| **Error Code (For Fail Status)** | |
| 400 | Bad Request |
| 401 | Unauthorized |
| 500 | Internal Server Error |

###### **URAPG Payment transaction from AXS**

|  |  |
| --- | --- |
| **Error Code (For Fail Status)** | |
| ERR001 | Unexpected system error |
| ERR002 | All Fields are Mandatory |
| ERR003 | Data Type are INcorrect and exceed max length |
| ERR004 | txnDate is incorrect format |
| ERR005 | TotalProductPrice not correct |
| ERR006 | totalProductPrice in the txnDetails Array not equal to amountPayable |

# **Section 4 - Process Payment with Payment Gateway**

### **Use Case**

1. This Pay Parking Offence Notice eService is integrated with the URA payment gateway to enable payment processing.
2. The eService allows Motorists to select one or more Notices from the Payable to make payment using one of the following payment methods:
3. eNets
4. Credit card
5. PayNow
6. The eService interfaces with the URA Payment Gateway to initiate the payment session and re-direct the Motorist to the Payment Gateway webpage to perform and complete the payment transaction.

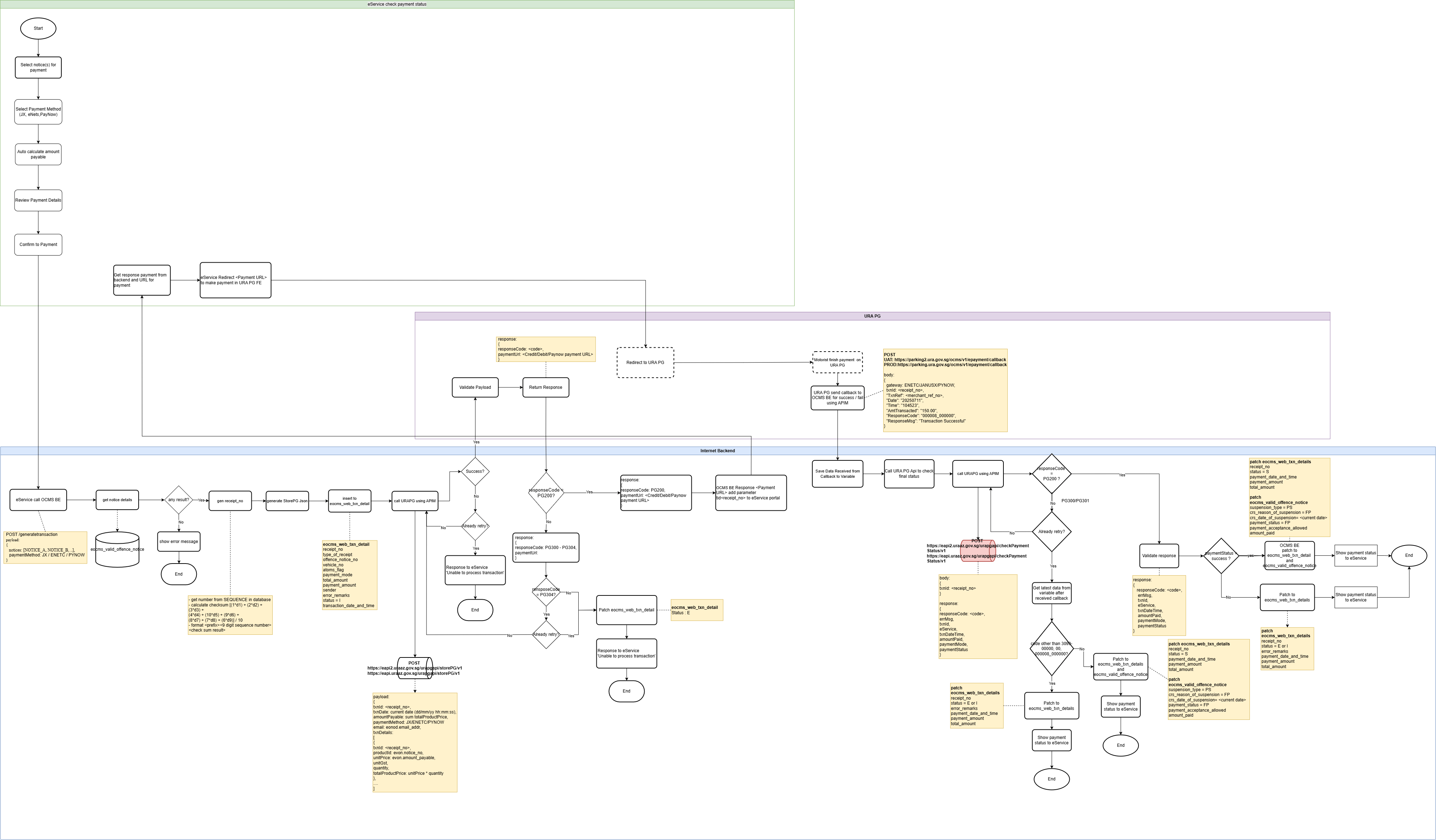






### **High Level Technical Flow**

#### **Diagram Flow Image**



NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Select notice(s) for payment | User action | Choose one or more notices to pay. |
| Select Payment Method (JX / eNets / PayNow) | User action | Pick the payment channel. |
| Auto calculate amount payable | System process | Compute total payable for selected notices. |
| Review Payment Details | User action | Review amounts, items, and method. |
| Confirm to Payment | User action | User confirms to proceed. |
| eService call OCMS BE | API entry | eService call OCMS BE to process |
| get notice details | Data lookup | Pull notice details from eocms\_valid\_offence\_notice. |
| any result? | System Decision | System validate result |
| If No | System Decision | If no matching notices, stop with error. |
| show error message → End | Error path | Return error to FE and terminate. |
| If Yes | System Decision | If yes, continue generate receipt\_no |
| gen receipt\_no | Generator | Get next sequence |
| generate StorePG JSON | Build payload | Construct URA PG storePG |
| insert to eocms\_web\_txn\_detail | Persist draft | Insert new data on eocms\_web\_txn\_detail table |
| call URAPG using APIM | Outbound call | Call URAPG to process the request |
| Success? | System Decision | Did APIM call succeed? |
| If No | System Decision | If no retry 1 time |
| Already retry? | System Decision | Validate for failure process already retry or not |
| If No | System Decision | Retry Process one time |
| If Yes | System Decision | If already retry one time, still error than stop process response to eService  “Unable to process payment” |
| If APIM Call Sucess | System Decision | If yes continue call URAPG |
| Validate Payload | Gateway step | URA PG validates the incoming storePG payload. |
| Return Response | Gateway step | Returns responseCode and paymentUrl to OCMS BE. |
| responseCode = PG200? | System Decision | Validate responseCode from URAPG |
| If No | System Decision | Response will be response:  {  responseCode: PG300 - PG304,  paymentUrl:  } |
| response: PG300–PG304? | Alt path | Handle non‑PG200 response set. |
| rensposeCode = PG304? | System Decision | If PG304, retry 1 time |
| Already retry? (on failure) | System Decision | If not retried yet, try call again; else return “Unable to process transaction”. |
| Patch eocms\_web\_txn\_detail → Status E | Update DB | Mark transaction error in DB. |
| Response to eService “Unable to process transaction” → End | Error path | Send failure message to FE and terminate. |
| response: PG200 + paymentUrl | Success path | Return success and URL to BE. |
| OCMS BE Response add parameter tid=<receipt\_no> to eService portal | System Process | Append tid query param (receipt no) when returning the payment URL to eService. |
| eService Redirect <Payment URL> to make payment in URA PG FE | Navigation | Redirect user to URA Payment Gateway page. |
| Redirect to URA PG | FE navigation | FE uses returned paymentUrl to land on URA PG page. |
| User makes payment at URA PG | User Action | Payer completes the payment on URAPG. |
| URAPG sends callback to OCMS BE | System Interaction | URAPG calls OCMS BE with result: |
| Save callback to java memory | Data Update | Persist the raw callback payload for later reconciliation. |
| Call URAPG “checkPaymentStatus” | API Call | OCMS BE calls |
| Success? | System Decision | Did the API call succeed? |
| Call URAPG Failed | System Action | Failed call URAPG for check final status payment get responseCode PG300/PG301 |
| Already retry? | System Decision | If the call failed, check if already retried once. |
| If not retried | System Interaction | If not retried yet, call “checkPaymentStatus” again. |
| If already retried | System Decision | If retried and still no success, fall back to the callback data saved in “Save callback to java memory.” |
| Response code code other than 3099-00000, 00, 000008\_000000? | System Interaction | Validate response code from memory |
| If yes | System Action | patch eocms\_web\_txn\_details |
| Response to eService | System Action | Show payment status to eService Payment Success |
| If no | System Action | patch eocms\_web\_txn\_details  and also  patch eocms\_valid\_offence\_notice |
| Response to eService | System Action | Show payment status to eService Payment Failed/Success |
| Call URAPG success | System Action | Success call URAPG for check final status, get responseCode PG200 |
| Validate response | System Interaction | Validate fields from final-check |
| Check paymentStatus | Decision | If paymentStatus = success |
| Patch txn + notice (success) | Data Update | patch eocms\_web\_txn\_details  and  patch eocms\_valid\_offence\_notice |
| paymentStatus not success | Decision | paymentStatus = Error/Incomplete |
| Patch txn (failure) | Data Update | patch eocms\_web\_txn\_details |
| Response to eService | System Action | Show payment status to eService Payment Failed/Success |
| End | End | Process completes. |
| ~~Verify / Check Status~~ | ~~External API Call~~ | ~~OCMS BE may call URA PG checkPaymentStatus/v1 API to confirm transaction result. +++~~ |
| ~~Update Transaction Record~~ | ~~Database Update~~ | ~~Update eocms\_web\_txn\_detail:~~   * ~~S (Success) → patch with txn date/time.~~ * ~~E (Error/Fail) → patch with error remarks.~~   ~~For success: also update eocms\_valid\_offence\_notice~~   * ~~payment\_status=FP~~ * ~~suspension\_type=PS~~ * ~~crs\_reason\_of\_suspension=FP.~~ |

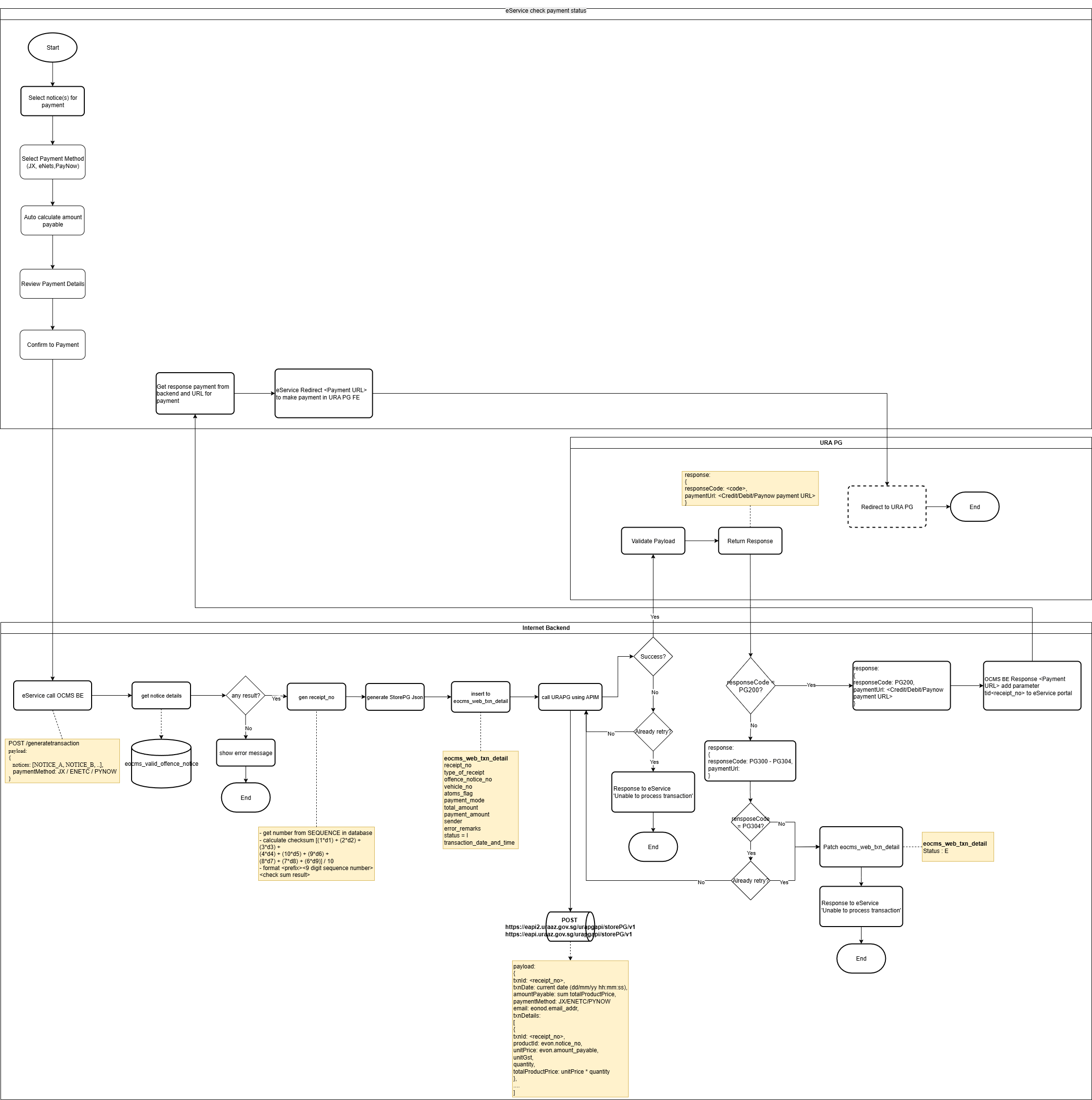


# **Section 5 - Payment processing and integration with URA Payment Gateway**

### **Use Case**

1. This Pay Parking Offence Notice eService is integrated with the URA payment gateway to enable payment processing.
2. The eService allows Motorists to select one or more Notices from the Payable to make payment using one of the following payment methods:
3. eNets
4. Credit card
5. PayNow
6. The eService interfaces with the URA Payment Gateway to initiate the payment session and re-direct the Motorist to the Payment Gateway webpage to perform and complete the payment transaction.

### **Diagram Flow Image**



NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Select notice(s) for payment | User action | *Refer to Section 4.2* |
| Confirm to Payment | User action | *Refer to Section 4.2* |
| eService call OCMS BE | API entry | *Refer to Section 4.2* |
| gen receipt\_no | Generator | *Refer to Section 4.2* |
| generate StorePG JSON | Build payload | *Refer to Section 4.2* |
| call URAPG using APIM | Outbound call | *Refer to Section 4.2* |
| Validate Payload | Gateway step | *Refer to Section 4.2* |
| Return Response | Gateway step | *Refer to Section 4.2* |
| OCMS BE Response add parameter tid=<receipt\_no> to eService portal | System Process | *Refer to Section 4.2* |
| eService Redirect <Payment URL> to make payment in URA PG FE | Navigation | *Refer to Section 4.2* |
| Redirect to URA PG | FE navigation | *Refer to Section 4.2* |



### **API Specification**

#### **API for eService**

##### **API Generate Transaction**

|  |  |
| --- | --- |
| API Name | Generate Transaction |
| URL | UAT : https://parking2.uraz.gov.sg /ocms/v1/generatetransaction  PRD : https://parking.uraz.gov.sg/ocms/v1/generatetransaction |
| Description | The API to Generate Transaction |
| Method | POST |
| Header | {  "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  "Content-Type": "application/json"  } |
| Payload | {     “noticeNo”: “[823000001G, 500500272B],  “paymentMethod”: “PYNOW”  } |
| Response | response:  {  “responseCode: “PG200”,  “paymentUrl”: “<Credit/Debit/Paynow payment URL>?tid=<recipt\_no>”  } |
| Response Failure | {  “responseCode”: “PG300”,  “paymentUrl”: “”  } |

#### **API Consume**

##### **API Call URA PG**

|  |  |
| --- | --- |
| API Name | Call URAPG using APIM |
| URL | UAT: https://eparking2.uraz.gov.sg/urapgapi/storePG/v1  PRD: https://eparking.uraz.gov.sg/urapgapi/storePG/v1 |
| Description | Call URAPG using APIM |
| Method | POST |
| Header | {  “Ocp-Apim-Subscription-Key”: “[APIM secret value]”,  "Content-Type": "application/json"  } |
| Payload | {  "txnId": "INV0000123450",  "txnDate": "2025-05-01T14:30:00",  "amountPayable": "10.00",  "paymentMethod": “PYNOW",  "email": "test@test.com",  "txnDetails": [  {  "txnId": "INV0000123450",  "productId": "823000001G",  "unitPrice": "10.00",  "unitGst": 0,  "quantity": 1,  "totalProductPrice": "10.00"  }  ]  } |
| Response Success | {  “responseCode: “PG200”  “paymentUrl”: “https://www.ura.gov.sg/paymentgateway/processPayNow.do"  } |
| Response Failure | {  “responseCode”: “PG300”,  “paymentUrl”:  } |

### **Database and Data Mapping**

#### **Payload for StorePG Json**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Zone** | **Database Table** | | **Field Name** | **Payload** |
| Internet | eocms\_web\_txn\_detail | | receipt\_no | txnId |
| Internet | eocms\_web\_txn\_detail | | transaction\_date\_and\_time | txnDate |
|  |  | | Total amount\_payable in transaction detail | amountPayable |
| Internet | eocms\_web\_txn\_detail | | payment\_mode | paymentMode |
| Internet | eocms\_offence\_notice\_owner\_driver | | email\_addr | Email |
| Transaction Detail | | | | |
| Internet | | eocms\_web\_txn\_detail | receipt\_no | txnId |
| Internet | | eocms\_valid\_offence\_notice | notice\_no | productId |
| Internet | | eocms\_valid\_offence\_notice | amount\_payable | unitprice |
|  | |  | 0 | unitGst |
|  | |  | 1 | quantity |
| Internet | | eocms\_valid\_offence\_notice | amount\_payable | totalProductPrice |

#### **Insert to eocms\_web\_txn\_detail**

|  |  |  |  |
| --- | --- | --- | --- |
| **Zone** | **Database Table** | **Field Name** | **Description** |
| Internet | eocms\_web\_txn\_detail | receipt\_no | Receipt number for the transaction (same as transaction id) |
| Internet | eocms\_web\_txn\_detail | type\_of\_receipt | Type code of the receipt |
| Internet | eocms\_web\_txn\_detail | offence\_notice\_no | Reference to offence notice number |
| Internet | eocms\_web\_txn\_detail | vehicle\_no | Vehicle registration number |
| Internet | eocms\_web\_txn\_detail | atoms\_flag | Flag indicating ATOMS integration status |
| Internet | eocms\_web\_txn\_detail | payment\_mode | Mode of payment used |
| Internet | eocms\_web\_txn\_detail | payment\_amount | Amount Paid |
| Internet | eocms\_web\_txn\_detail | sender | ID of the sender system or user |
| Internet | eocms\_web\_txn\_detail | error\_remarks | Error message if any issue during processing |
| Internet | eocms\_web\_txn\_detail | status | Status of the transaction  (S = success/E = error/I = Incomplete |
| Internet | eocms\_web\_txn\_detail | transaction\_date\_and\_time | Date of the transaction |

#### **Patch eocms\_web\_txn\_detail**

|  |  |  |  |
| --- | --- | --- | --- |
| **Zone** | **Database Table** | **Field Name** | **Description** |
| Internet | eocms\_web\_txn\_detail | status | E = Error |

### **Success Outcome**

1. The user selects valid notice(s) for payment and successfully goes through amount calculation, payment method selection, and payment confirmation.

2. The backend verifies that the selected notice(s) are valid and retrievable.

3. The backend generates a unique receipt number for the transaction.

4. The user is redirected to the payment gateway using the provided paymentURL.

5. The user completes the payment process at the URA/PG webpage.

6. URA/PG sends a callback to the application’s success URL, indicating that payment was successful.

### **Error Handling**

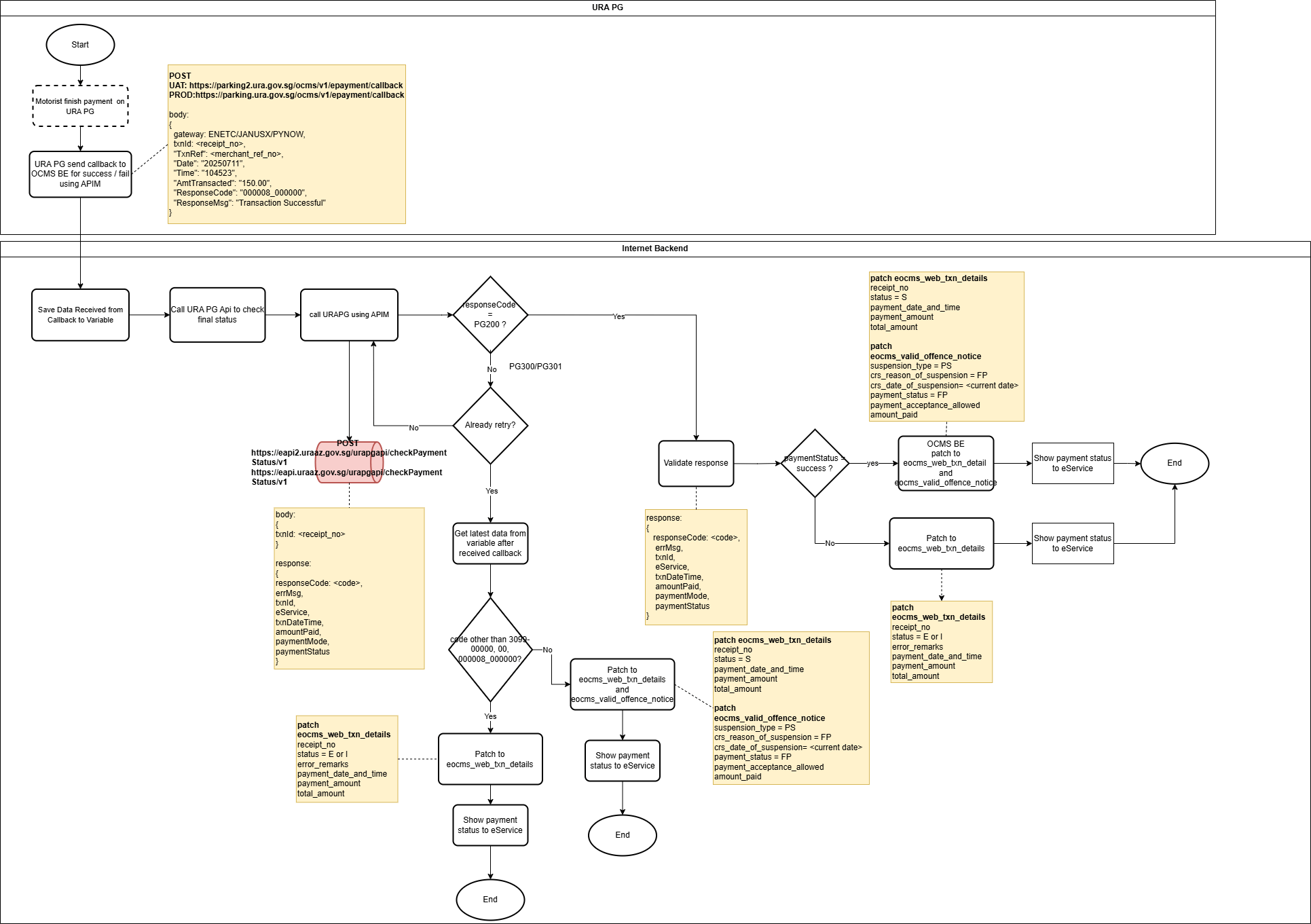
|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Check Return Response | Response Validation | The system checks the response from the URA PG API. If the payment gateway returns an error, the system displays an error message. |
| Show Error Message | |  | | --- | |  |   Error Display | If the payment fails at any point, such as during credit card validation or transaction submission, an error message is shown to the user, explaining the failure. |
| Error Handling on Payment Method | Payment Validation | If the user selects an unsupported or invalid payment method, the system alerts the user with an appropriate error message. |
| Error on Transaction Reference Generation | Data Generation Error | If there is an issue generating the transaction reference number, the system displays an error and prompts the user to try again or contact support. |
| Error During API Call to URA PG | System Interaction Error | If there is an issue with the API call to the URA Payment Gateway, there is a retry. |

# **Section 6 - Transaction Success and Failure Handling**

### **Use Case**

1. After payment has been processed, URA Payment Gateway will respond to OCMS Internet BE by calling either the success URL or fail URL with the transaction data.
2. When the OCMS Internet Backend receives the data, it will initiate a check status query to URA Payment Gateway to retrieve the final payment status.
3. When the OCMS Internet Backend receives the final status, it will respond to the eService Portal with the final payment status.
4. When the eService Portal receives the payment status, it will display the success or fail message on the UI to notify the Motorist of the payment status.

### **Diagram Flow Image**



NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| User makes payment at URA PG | User Action | *Refer to Section 4.2* |
| URAPG sends callback to OCMS BE | System Interaction | *Refer to Section 4.2* |
| ~~Save callback to java memory~~ | ~~Data Update~~ | ~~Persist the raw callback payload for later reconciliation.~~ |
| Response code code other than 3099-00000, 00, 000008\_000000? | System Interaction | *Refer to Section 4.2* |
| If yes | System Action | *Refer to Section 4.2* |
| Response to eService | System Action | *Refer to Section 4.2* |
| If no | System Action | *Refer to Section 4.2*  ~~patch eocms\_valid\_offence\_notice~~ |
| Response to eService | System Action | *Refer to Section 4.2* |
| Call URAPG success | System Action | *Refer to Section 4.2* |
| Validate response | System Interaction | *Refer to Section 4.2* |
| Check paymentStatus | Decision | *Refer to Section 4.2* |
| Response to eService | System Action | *Refer to Section 4.2* |



### **API Specification**

#### **API Provide**

##### **API URA PG Callback Success**

|  |  |
| --- | --- |
| API Name | Success Callback from URA Payment Gateway |
| URL | UAT : https://parking2.uraz.gov.sg /ocms/epayment/callback/v1/success  PRD : https://parking.uraz.gov.sg/ocms/epayment/callback/v1/success |
| Description | Get Success Callback from URA Payment Gateway |
| Method | POST |
| Header | {  "Content-Type": "multipart/form-data",  “Ocp-Apim-Subscription-Key”: “[APIM secret value]”  } |
| Query Params |  |
| Response | {    "txnID": "INV0000123450",    "txnRef": "TXN6789",    "gateway": "ENETC",    "amtTransacted": 100.0,    "date": "2025-06-10",    "time": "15:30:00",    "responseCode": "00",    "responseMsg": "Transaction successful"  } |

##### **API URA PG Callback Failed**

|  |  |
| --- | --- |
| API Name | Failed Callback from URA Payment Gateway |
| URL | UAT : https://parking2.uraz.gov.sg /ocms/epayment/callback/v1/failed  PRD : https://parking.uraz.gov.sg/ocms/epayment/callback/v1/failed |
| Description | Get Failed Callback from URA Payment Gateway |
| Method | POST |
| Header | {  "Content-Type": "multipart/form-data",  “Ocp-Apim-Subscription-Key”: “[APIM secret value]”  } |
| Query Params |  |
| Response | {    "txnID": "INV0000123450",    "txnRef": "TXN6789",    "gateway": "ENETC",    "amtTransacted": 100.0,    "date": "2025-06-10",    "time": "15:30:00",    "responseCode": "01",    "responseMsg": "Transaction failed"  } |

#### **API Consume**

##### **API Check Payment Status from URA Payment Gateway**

|  |  |
| --- | --- |
| API Name | Check Payment Status from URA Payment Gateway |
| URL | https://eapi2.uraaz.gov.sg/urapgapi/checkPaymentStatus/v1  https://eapi.uraaz.gov.sg/urapgapi/checkPaymentStatus/v1 |
| Description | Call URA Payment Gateway API to Check Status |
| Method | POST |
| Header | {  "Content-Type": "multipart/form-data",  “Ocp-Apim-Subscription-Key”: “[APIM secret value]”  } |
| Payload | {     “txnId”: "INV0000123450"  } |
| Response Success | {    “responseCode”: “PG200”,     "txnID": "INV0000123450",      “eService”: “Payment For Parking Offence”,      “txnDateTime”: "2025-05-01T14:30:00",      “amountPaid”: 100.0,      “paymentMode”: "PYNOW",      “paymentStatus”: "S"  } |
| Response Failure | {     “responseCode”: “PG300”,      “errMsg”: “error payment”,      “txnId”: “INV0000123450”,      “eService”: “Payment For Parking Offence”,,      “txnDateTime”: "2025-05-01T14:30:00",,      “amountPaid”: 100.0,      “paymentMode”: "PYNOW",      “paymentStatus”: "F"  } |

### **Database and Data Mapping**

#### **Patch data to eocms\_web\_txn\_detail**

|  |  |  |  |
| --- | --- | --- | --- |
| **Zone** | **Database Table** | **Field Name** | **Source** |
| Internet | eocms\_web\_txn\_detail | receipt\_no | txnId |
| Internet | eocms\_web\_txn\_detail | payment\_date\_and\_time | txnDateTime |
| Internet | eocms\_web\_txn\_detail | error\_remark | errMsg |
| Internet | eocms\_web\_txn\_detail | status | paymentStatus |
| Internet | eocms\_web\_txn\_detail | payment\_amount | amtTransacted |
| Internet | eocms\_web\_txn\_detail | total\_amount | amtTransacted |

#### **UI Data Mapping Transaction Payment Success**

|  |  |  |  |
| --- | --- | --- | --- |
| **Zone** | **Database Table** | **Field Name** | **UI Field** |
| Internet | eocms\_web\_txn\_detail | receipt\_no | Transaction Reference No |
| Internet | eocms\_web\_txn\_detail | payment\_date\_and\_time | Transaction Date & Time |
| Internet | eocms\_web\_txn\_detail | receipt\_no | Official Receipt Number |
| Internet | eocms\_web\_txn\_detail | payment\_mode | Payment Method |
| Internet | eocms\_web\_txn\_detail | payment\_amount | Amount Paid |

#### **UI Patch eocms\_valid\_offence\_notice**

|  |  |  |  |
| --- | --- | --- | --- |
| **Zone** | **Database Table** | **Field Name** | **Value** |
| Internet | eocms\_valid\_offence\_notice | crs\_reason\_of\_suspension | FP |
| Internet | eocms\_valid\_offence\_notice | crs\_date\_of\_suspension | Current Date |
| Internet | eocms\_valid\_offence\_notice | suspension\_type | PS |
| Internet | eocms\_valid\_offence\_notice | payment\_status | FP |

### **Success Outcome**

* + - 1. System shows a success alert, User is notified of the successful transaction.
      2. User can download the receipt, The receipt is generated using a template and made available for download.
      3. Data in eVON system is updated, record created in suspended\_notice table
      4. Payment details are patched/inserted to eocms\_web\_txn\_detail and ocms\_web\_txn\_detail.
      5. Response is displayed, system show success or failure reponse to the user

### **Error Handling**

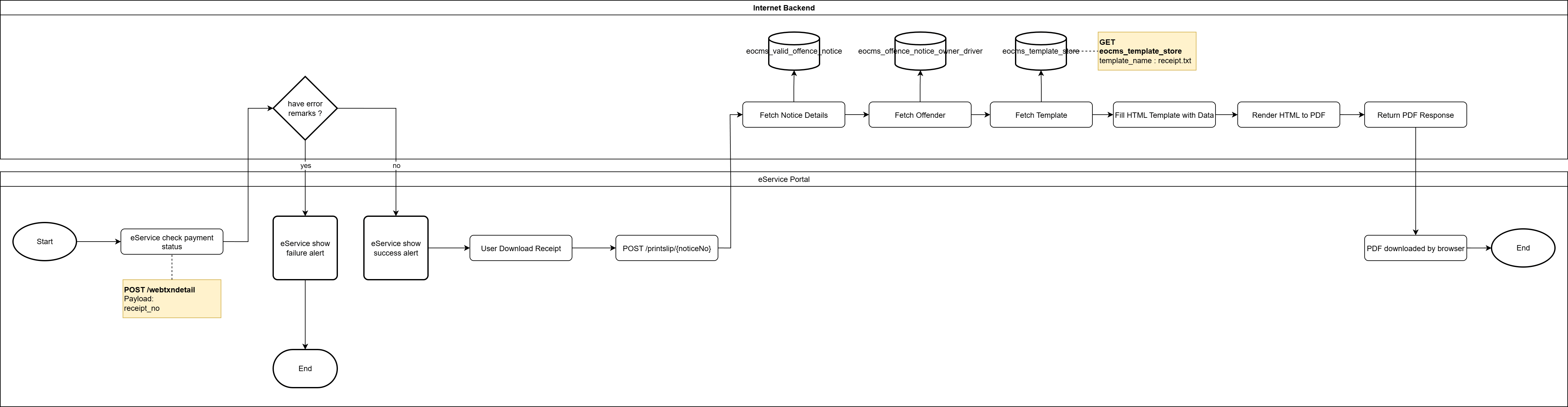
|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Failed Response from URA PG Callback | Error Handling | If the callback response from the URA Payment Gateway indicates a failure, the system handles it by logging the error and preparing a response. |
| Store to eocms\_web\_txn\_detail with errMsg | Data Storage with Error Details | The error message (errMsg) from the failed callback response is stored in the eocms\_web\_txn\_detail table to maintain a record of the failure. |
| URA Payment Gateway API Check Status Failed Response | Error Handling | If the status check from the URA Payment Gateway API fails, an error response is received and handled accordingly. |
| Checking Have Error Response | Error Check | The system checks if the response from the URA Payment Gateway API or the status check has an error. If it does, the system proceeds with the error handling flow. |

# **Section 7 - Receipt Download**

### **Use Case**

1. The motorists can download receipts for their paid notices by first logging into the eService with their SingPass or Corppass.parking
2. Once logged in, they can access a list of all payments made (both online and offline) within the past six months. From this list, they can select any notice and download its corresponding receipt.

### **Diagram Flow Image**



NOTE: Due to page size limit, the full-sized image is appended.



|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Start | Initialization | The process begins. |
| eService check payment | System Action | eService check for payment status to Backend |
| If have failed remark | System Response | eService will show failure alert |
| If didn’t have failed remark | System Response | eService show success alert |
| User Download Receipt | User Action | User can download the receipt. |
| POST /printslip/ | System Interaction | System sends a POST request to the backend to generate a receipt.  Payload:  {  "$limit”: “9999”,  “$skip”: “0”,  “offenceNoticeNo”: ”N123456789”  } |
| Fetch Notice Data | Data Retrieval | Backend fetches notice data from  eocms\_valid\_offence\_notice |
| Fetch Offender | Data Retrieval | Backend fetches offender details from eocms\_offence\_notice\_owner\_driver. |
| Fetch Template | Data Retrieval | Backend fetches the template from eocms\_template\_store. |
| Fill Template with Data | Data Formatting | System fills the template with relevant data. |
| Render HTML to PDF | Data Generation | System renders the filled template as a PDF document. |
| Return PDF Response | System Response | Backend returns the PDF as a response to the client. |
| PDF downloaded by browser | System Action | The PDF receipt is downloaded by the user's browser. |
| End | System Navigation | The process ends. |



### **API Specification**

#### **API for eService**

##### **API Get Transaction Details**

|  |  |
| --- | --- |
| API Name | Get Web Transaction Detail |
| URL | UAT : https://parking2.uraz.gov.sg/ocms/v1/webtxndetail  PRD : https://parking.uraz.gov.sg/ocms/v1/webtxndetail |
| Description | This API to get web txn detail |
| Method | POST |
| Header | {  "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  "Content-Type": "application/json"  } |
| Payload | {  "$limit”: “9999”,  “$skip”: “0”,  “receiptNo”: ”ON0000000019”  } |
| Response | {  "total": 1,  "limit": 10,  "skip": 0,  "data": [  {  "receiptNo":”ON0000000019”,  "typeOfReceipt": null,  "remarks": null,  "transactionDate":”2024-04-04”,  "transactionTime": “10:00:00”,  "offenceNoticeNo":”N123456789”,  "vehicleNo": “XD4545X”,  "atomsFlag": "N",  "paymentMode": “PYNOW”,  "paymentAmount": "70.00",  "sender": “URAPG”,  "status": "S",  "creUserId": “USER1”,  “amountPayable”: “70.00”,  “amountPaid”: “70.00”  }  ]  } |
| Response Failure | {  “data”:  {  "appCode": "OCMS-5002",  "message": "The request timed out. Please try again later"  }  } |

##### **API Download Receipt**

|  |  |
| --- | --- |
| API Name | Download Receipt |
| URL | UAT: https://api2.uraez.gov.sg/ocms/v1/printslip  PRD: https://api.uraez.gov.sg/ocms/v1/printslip |
| Description | The API to Download Receipt |
| Method | POST |
| Header | {      "Authorization": "Bearer <your\_access\_token>",      "Content-Type": "application/json"  } |
| Payload | {  noticeNo : <noticeNo>  } |
| Response | {      "data": {          "appCode": "OCMS-2000",          "message": "Success Download Receipt."      }  } |
| Response Failure | {      "data": {          "appCode": "OCMS-5000",          "message": "The request took too long to process. Please try again later."      }  } |

### **Database and Data Mapping**

#### **Data in Generated Receipt**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Zone** | **Database Table** | | | **Field DB Name** | **Field Name** | |
| Internet |  | | |  | User Message | |
| Internet | eocms\_web\_txn\_detail | | | transaction\_date\_and\_time | Transaction Date & Time | |
| Internet | eocms\_web\_txn\_detail | | | Receipt\_no | Transaction Reference No. | |
| Internet | eocms\_web\_txn\_detail | | | payment\_mode | Payment Method | |
| **Transaction Summary (List of Notices that have been paid, 1 Notice per row)** | | | | | | |
|  | |  |  | | | S/N |
| Internet | | eocms\_offence\_notice\_owner\_driver | notice\_no | | | Notice Number |
| Internet | | eocms\_web\_txn\_detail | vehicle\_no | | | Vehicle Number |
| Internet | | eocms\_valid\_offence\_notice | notice\_date\_and\_time | | | Date of Offence |
| Internet | | eocms\_web\_txn\_detail | payment\_amount | | | Amount (S$) |

### **Success Outcome**

1. System enables the user to download a payment receipt.
2. The system fills the template with data, renders it as a PDF, and prepares it for delivery.
3. The PDF receipt is returned to the user and successfully downloaded via their browser.
4. All steps are executed smoothly with proper data retrieval, validation, formatting, and user notification, ending the process as intended.

### **Error Handling**

|  |  |  |
| --- | --- | --- |
| **Step** | **Definition** | **Brief Description** |
| Show Response | Data retrieval failure or timeout | Displays error if the backend fails to return results due to network/server/database issues. |
| Download the Receipt | Client-side or browser download failure | Alerts user if the file could not be saved due to browser restrictions, pop-up blocks, or disk space. |
| Network/Session Timeout | User session timeout or disconnection | Warns user and prompts them to restart the process if idle or disconnected too long. |
| System Error | Unexpected or unhandled exception | Catches general failures with a generic error message and logs incident for review. |