

Scope Document

Bulk Bill Payment

GrameenPhone

Version 1.1

Table of Contents

Table of Contents.....	3
1 Introduction	5
1.1 Purpose	5
1.2 Scope.....	5
1.3 Audience	5
2 Bulk BillPayment Portal.....	6
2.1 Login Flow:	6
2.2 BillPayment Flow:.....	23
2.3 Password Management Flow:	24
2.4 Existing Flow Diagrams:	37
3 Technical Solution Requirement	40
3.1 Solution Overview	40
3.1.1 Tech stack to be used to deploy and run existing code	40
3.2 Deployment Specifications of existing module	40
3.2.1 DB Readiness.....	40
3.2.2 Application readiness	41
3.3 Project Specific Scope.....	41
3.3.1 Current Implemenation	41
3.3.2 Graphical User Interface.....	41
3.3.3 Training.....	42
3.3.4 Documentation	42
3.3.5 Postman Collection.....	42
3.4 Testing and Acceptance	42
3.4.1 System Integration Testing	43
3.4.2 User Acceptance Procedure.....	43
3.5 Risks, Impact and Mitigation	43
3.6 Responsibility Matrix.....	43

3.7	Out of Scope	43
3.8	Code management	43
3.9	Project handover	44
3.10	Project Timelines.....	44
3.11	Support and maintenance.....	44
4	Document Change History	45

1 Introduction

This document describes scope of work for Bulk BillPayment UI requirement for GrameenPhone

1.1 Purpose

It is the scope of work document for the changes needed in current Bulk BillPayment module of Grameenphone to integrate with upgraded mobiquity PayX Solution. This will give flexibility to admin users to fetch bill details of account holders in bulk and then pay all pending bills in bulk.

1.2 Scope

Grameenphone has already shared their code of existing Bulk BillPayment UI from old version and want same to be implemented in PayX solution provided by comviva in Phase B.

On existing feature, below are changes that need to be done:

- While fetching pending bill details of all bill account selected on UI : Change in existing integration logic and update to Mobiquity Integration API
- For Bill payment : Change in existing integration logic and update to Mobiquity Integration API
- Ambiguous payment handling scenarios(for enquiry and not reversal or refund) request will be routed to Mobiquity , for any dispute management
- Email and SMS handling should be there as per current setup : integration with Email server and SMSGW
- Login page to be created on WEB to enable user login with credentials. In old system, login page was from IDP. But in latest deployment Login feature should be created in Bulk UI itself.
- Password management (change and forgot password) feature has to be implemented on Bulk UI : Integration with Mobiquity for backend logic

Note : There are 2 UI for Bulk Bill Payment and Reporting. Current scope is just to give feature of Bulk Billpayment UI . Reporting UI requirements will be fulfilled from Pentaho which is part of Mobiquity solution.

1.3 Audience

Comviva technical teams, partner teams

2 Bulk BillPayment Portal

Admin user would be able to access Bulk BillPayment portal. Each of them will have their specific userId and password

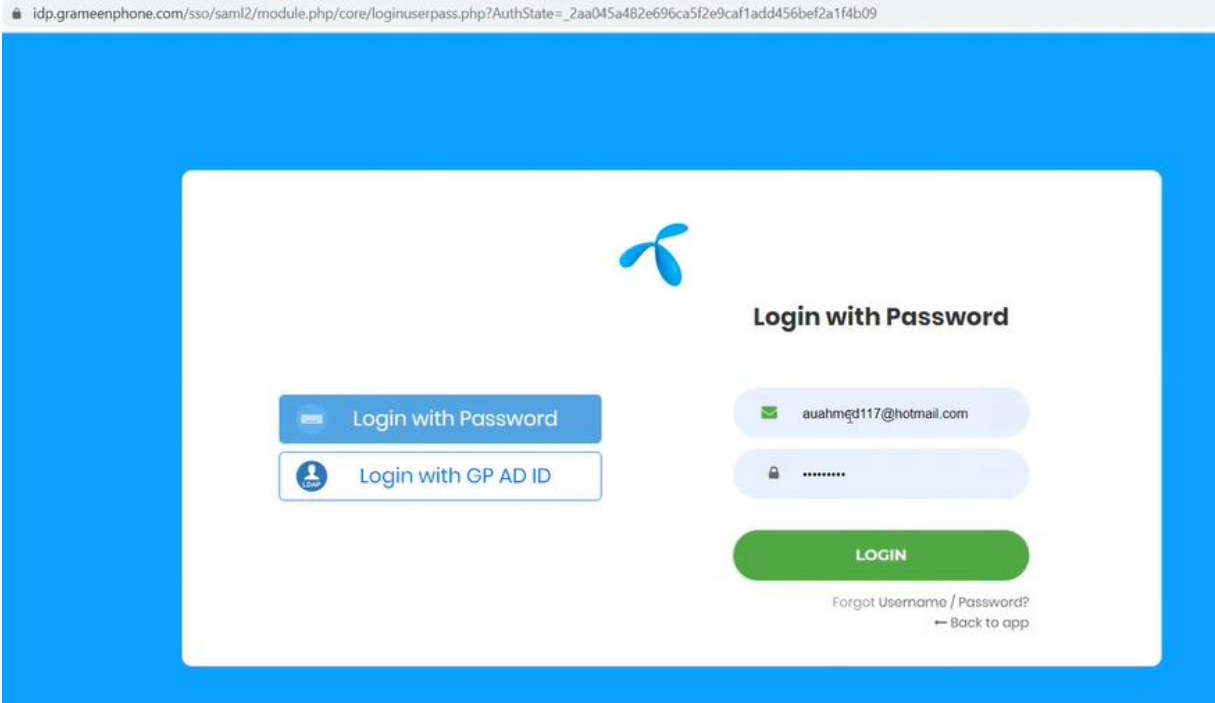
2.1 Login Flow:

Current Flow:

In old architecture, User authentication while Login is maintained by IDP. So whenever user is logging into the system, application routes user to login page of IDP. Where userid and password is asked from user.

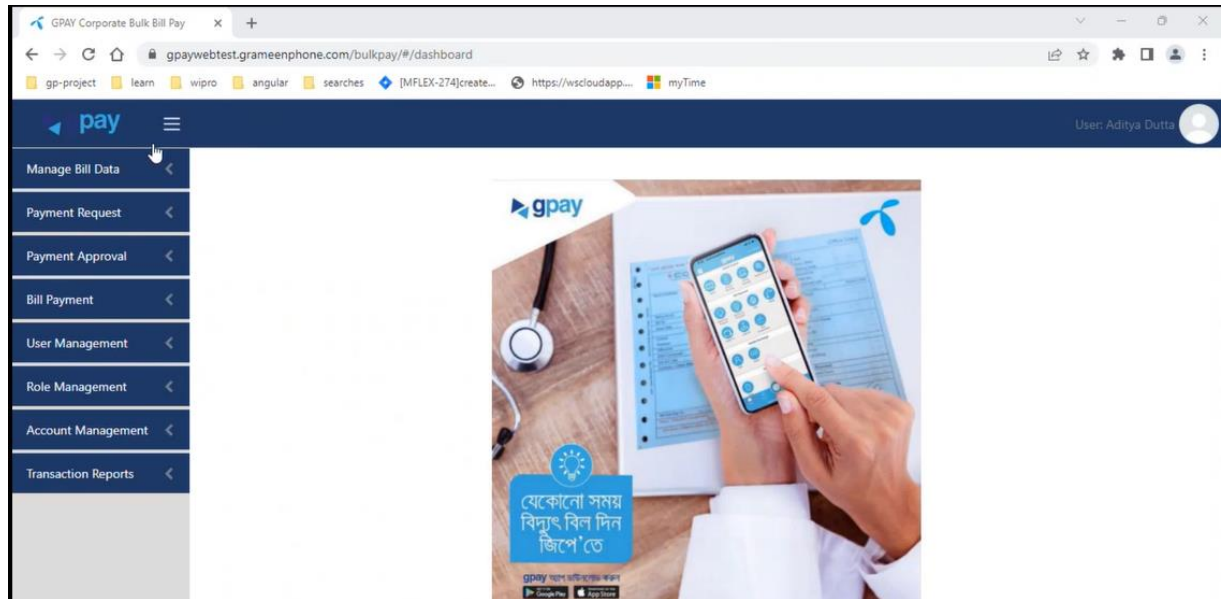
IDP validated credentials, on success it calls Billpay Dashboard page and give access to admin user.

Login Page of IDP :



The screenshot shows a web browser window with the URL `idp.grameenphone.com/sso/saml2/module.php/core/loginuserpass.php?AuthState=_2aa045a482e696ca5f2e9caf1add456bef2a1f4b09`. The page has a blue background and a white login box. Inside the box, there is a blue logo at the top center. Below the logo, there are two buttons: "Login with Password" (blue) and "Login with GP AD ID" (white with a blue border). To the right of these buttons, there are two input fields: one for the username (containing "auahmed117@hotmail.com") and one for the password (containing "*****"). Below the input fields is a green "LOGIN" button. At the bottom right of the login box, there are two links: "Forgot Username / Password?" and "Back to app".

BillPay Dashboard snapshot is below:



New Flow:

Partner need to create this Login page into Bulk UI code itself. And w.r.t. authentication, hit is send to Mobiquity for user validation. Since there is no password storage in Bulk Bill mflex DB. Only userId and email id of users are stored. If user is verified by Mobiquity, then dashboard will be shown.

In order to support this approach, we must ensure that admin users who are present in mobiquity systems can only access bulk billpay dashboard. And they must need mobiquity login id and credentials only to login after registration in Bulk Billpay application. When any mobiquity user want access on Bulk Pay dashboard, then they have to get registered in BulkPay UI also with same login ID.

So login ID of user in mobiquity and BulkUI DB should match.

BulkUI application will send request to generate system token at time of login, once system token is fetched, login request will be sent with system token and login credentials and in return if user is valid in mobiquity system then userToken is sent in response to Login API along with refresh Token.

On receiving successful login, Bulk UI dashboard must be made visible to user.

Expections from Partner:

- Create Login page
- Integrate with mobiquity to fetch system token and Login validation
- Basis mobiquity Login API response, route request to already existing Bulk dashboard in system.

2.1.1

API Details:

2.1.1.1 Fetching System Token

Action: GET

Endpoint: /ums/v1/user/auth/web/system-token

Description

This API is used by the DFS container to generate grant type credentials for the user to login/access mobile or web application. By default, one grant type (system token) is associated with the system. This API helps in creating multiple grant types based on requirement.

Response Body

Code: 200

Get System Token for Web Application.

Fields	Type	Description	Example/Allowed Values
LoginWithOTPResponseModel	Login response model with token		
language	String	Preferred language of the user. By default, system supports 3 languages English/French/Arabic.	En, ar, fr
lastLoginTime	String	Time stamp of last login of the user.	2020-10-13T02:47:56
message*	String	The API response message that shows the type of information returned. No Specific length	OTP validation is required. Please enter OTP to continue
serviceFlow	String	A unique code that is internal to Comviva and identifies the business process flow associated to the API.	OTPLLOGIN
serviceRequestId*	String	Whenever an API is called, Comviva generates a unique service request ID for the API request. This unique ID traverses' components in the business process flow. It is useful when the process flow is paused for the user input and needs to be resumed when user has provided the input. Length-36 characters (System generated)	f491f6b1-aa9b-43de-93b0-c85eda706a2c
status*	String	The status to show whether the API call is a success, failure, in progress, or paused.	SUCCEEDED, FAILED, INPROGRESS, or PAUSED
token	String	TokenData	

Fields	Type	Description		Example/Allowed Values	
		Fields	Type	Description	Example
		access_token *	String	Access token is used for token based authentication to allow an application to access an API. The application receives an Access Token after a user successfully authenticates and authorizes access. User can then pass this access token as a credential to perform any transactions in the application. Access tokens are short-lived tokens.	asdfqwrtwy
		expires_in	Integer (\$int32)	Validity period (in seconds) for accessing the token.	299
		refresh_token	String	Once the access token expires, you can use the refresh token to get the new access token. Refresh tokens also expire but they are supposed to live much longer than the access token. Max Length- 256 characters	asdfqwrtwy
transactionId *	String	The unique id generated by the Comviva platform for the		XX220316.0959.A15026	

Fields	Type	Description	Example/Allowed Values
		transaction. Length-21 character	
txnStatus	String	Defines the five different statuses of the transaction. Not Applicable for Non-Financial APIs TI – Transaction Initiated TS – Transaction Succeeded TF – Transaction Failure TP – Transaction Paused TA – Transaction Ambiguous	Success
userId	String	Unique ID generated by the system after successful user registration. Length-20 characters	US.k6GH1579603850092

Sample Response Body

```
{
  "serviceRequestId":"d69b89d8-a8e2-41a7-8ee2-8df7575ed4f9", "message":null, "transactionId":null, "txnStatus":null, "serviceFlow":"SYSTEMTOKEN", "status":"SUCCEEDED", "language":"en", "token":{"access_token":"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Imp3cy1rZXktcHVibG1jLTEifQ.eyJzY29wZSI6WyJHRVRfVFNFU19BTkRfQUNDt1VOVF9ERVRBSUxTIiwiaXNjaWUiOiJ1OTI0NTQ2LCJhdXRob3JpdGlscyI6WyJST0xFOX0NMSUVOVCI6LCJqdGkiOiI1Njg5YzhkMy01ZTFkLTJmZEtYmY2OC0zODE1MzFkMjZlYWQiLCJjbGllbnRfaWQiOiJDb3JlV2ViIn0.ElEwVfg0CkRcd2qTbbTGVo99KM6_ZhKkiVKsQ1lloFzZMUQbtCPRGK8hmHfmF2QBF1ty4qksA308qV-UyN4PyLt7oyGHFOhRmKdl0NTHndo0jaZm8IfpexjxiS0QOMZ3shZ5rXVoRxxkltbED1mJo9csO-SYIYKT6FXrxF7zsKY6b9hnIbJkm9ockRHeN_4OREvbH42vwSMPuulGbKyXeGiJ8nA5uRXYlU59pZ1y0eubAfmftsb9fwOsvKMiabZsHsJYyFxxqXYrIyHhdq_Ar0hxGvIx7sp6Ov9qZZCwwK9LEal51N7UcfdLsBq5LoyalfhOrhReWjYMKVXOnlA4N4g", "expires_in":2868, "refresh_token":null}, "lastLoginTime":null, "userId":null, "sessionIdList":null
}
```

Error Codes

Error Code	Description
400	Bad request
500	Internal Server Error

2.1.1.2 Validating User Credential and Generate Token for Web

Action: POST

Endpoint: /ums/v3/user/auth/web/login

Description

This API is used to validate user login credentials based on login policy and generates token to access web application.

Request body

Fields	Type	Description	Example/Allowed Values																				
LoginCriteriaRequest	String	Validate user login credential based on the criteria.																					
AuthenticationValue*	String	PIN/Password of the user Default length of pin is 4 Min password length is 5 & Max password length is 10	2468																				
bearerCode *	String	Access bearer channel of user which includes Web, USSD, Core Web, Mobile App	Web, USSD, Core Web, Mobile App																				
deviceInfo *	DeviceDetailDTO Description: Representing the device details from where request raised. <table border="1"> <thead> <tr> <th>Fields</th><th>Type</th><th>Description</th><th>Example/Allowed Values</th></tr> </thead> <tbody> <tr> <td>appName</td><td>String</td><td>Application name Length-40 character</td><td>mobiquity</td></tr> <tr> <td>appVersion</td><td>String</td><td>Version of the app that the user accessing. Length-20 character</td><td>10.2</td></tr> <tr> <td>browser</td><td>String</td><td>Browser compatibility of the application. Max Length-50 character</td><td>Chrome</td></tr> <tr> <td>deviceId *</td><td>String</td><td>Unique ID of the</td><td>excs-233-daca-312</td></tr> </tbody> </table>			Fields	Type	Description	Example/Allowed Values	appName	String	Application name Length-40 character	mobiquity	appVersion	String	Version of the app that the user accessing. Length-20 character	10.2	browser	String	Browser compatibility of the application. Max Length-50 character	Chrome	deviceId *	String	Unique ID of the	excs-233-daca-312
Fields	Type	Description	Example/Allowed Values																				
appName	String	Application name Length-40 character	mobiquity																				
appVersion	String	Version of the app that the user accessing. Length-20 character	10.2																				
browser	String	Browser compatibility of the application. Max Length-50 character	Chrome																				
deviceId *	String	Unique ID of the	excs-233-daca-312																				

Fields	Type	Description	Example/Allowed Values
		device from which the user is accessing the app. Max Length-50 character	
	isPublicDevice *	String	Is the device public, Y/N
	latitude	String	Latitude geography of the device. Max Length-30 character
	longitude	String	Longitude geography of the device. Max Length-30 character
	mac	String	MAC address of the device. Max Length-30 character
	model	String	Model number of the device. Max Length-150 character
	networkOperator	String	Network operator of the device. Max Length-30 character
	networkType	String	Network type of the device which includes 3G, 4G, 5G, etc. Max Length-30 character
	os	String	Operating system of the device. Max Length-20 character
	providerIpAddress	String	IP address of the

Fields	Type	Description	Example/Allowed Values
		service provider. Max Length-50 character	
identifierType *	String	Access identifier type of the user/transactor	mobileNumber
identifierValue	String	Access identifier value associated with the identifier type selected. Mobile Number:8-15 Email: Max length-40 characters LOGINID-min length is 3 & max length is 20	777XXXXXX
isTokenRequired *	String	If token is required to access the app, select Y else select N	Y
language *	String	Preferred language of the user. By default, system supports 3 languages English/French/Arabic.	En, ar, fr
workspaceId *	String	Workspace is the classification of all users at high level. All users are clubbed together under three workspaces SUBSCRIBER/BUSINESS/ADMIN	SUBSCRIBER/BUSINESS/ADMIN

Sample Request Body

```
{
  "bearerCode": "WEB",
  "language": "en",
  "workspaceId": "ADMIN",
  "identifierType": "LOGINID",
  "identifierValue": "System01",
  "authenticationValue": "Com@1357",
  "isTokenRequired": "Y",
  "deviceInfo": {
    "appName": "mobilePay",
    "appVersion": "V X.9",
    "deviceId": "ffed2d4608c5191f5086b2f2cf160afd",
    "browser": "Google Chrome",
    "isPublicDevice": "N",
    "latitude": "",
    "longitude": "",
    "mac": "",
    "model": "Desktop - Windows 10",
    "networkOperator": "",
    "networkType": "",
    "os": "",
    "providerIpAddress": "136.226.255.14"
  }
}
```

Response Body

Code: 200

Validates credentials based on the login policy and generate token if request came from existing mapped device.

Fields	Type	Description	Example/Allowed Values
LoginWithOTPResponseModel	Login response model with token		
language	String	Preferred language of the user. By default, system supports 3 languages	En, ar, fr

Fields	Type	Description	Example/Allowed Values								
		English/French/Arabic.									
lastLoginTime	String	Time stamp of last login of the user.	2020-10-13T02:47:56								
message *	String	The API response message that shows the type of information returned.	OTP validation is required. Please enter OTP to continue								
serviceFlow	String	A unique code that is internal to Comviva and identifies the business process flow associated to the API.	OTPLGIN								
serviceRequestId *	String	Whenever an API is called, Comviva generates a unique service request ID for the API request. This unique ID traverses' components in the business process flow. It is useful when the process flow is paused for the user input and needs to be resumed when user has provided the input.	f491f6b1-aa9b-43de-93b0-c85eda706a2c								
status *	String	The status to show whether the API call is a success, failure, in progress, or paused.	SUCCEEDED, FAILED, INPROGRESS, or PAUSED								
token	String	<div>TokenData</div> <table> <tr> <th>Fields</th><th>Type</th><th>Description</th><th>Example/Allowed Values</th></tr> <tr> <td>access_token *</td><td>String</td><td>Access token is used for token based authentication to allow an application to access an API. The application receives an Access Token after a user successfully authenticates and authorizes access. User can then pass this access token as a</td><td>asdfqwrtwy</td></tr> </table>		Fields	Type	Description	Example/Allowed Values	access_token *	String	Access token is used for token based authentication to allow an application to access an API. The application receives an Access Token after a user successfully authenticates and authorizes access. User can then pass this access token as a	asdfqwrtwy
Fields	Type	Description	Example/Allowed Values								
access_token *	String	Access token is used for token based authentication to allow an application to access an API. The application receives an Access Token after a user successfully authenticates and authorizes access. User can then pass this access token as a	asdfqwrtwy								

Fields	Type	Description	Example/Allowed Values
			credential to perform any transactions in the application. Access tokens are short-lived tokens.
		expires_in	Integer (\$int32) Validity period (in seconds) for accessing the token. 299
		refresh_token	String Once the access token expires, you can use the refresh token to get the new access token. Refresh tokens also expire but they are supposed to live much longer than the access token. Length-256 characters asdfqwrtwy
transactionId *	String	The unique id generated by the Comviva platform for the transaction. Length-20 character	XX220316.0959.A15026
txnStatus	String	Defines the five different statuses of the transaction. Not Applicable for Non-Financial APIs TI – Transaction Initiated TS – Transaction Succeeded TF – Transaction Failure TP – Transaction Paused	Transaction Success


```
6iwNI9WuL9KuUvIB_cMtk2MkPa4R6dTGw"
  },
  "lastLoginTime": "2023-02-08T14:54:14",
  "userId": "US.78281675624511391"
}
```

Below is the response body when two factor authentication is enabled:

```
{
  "serviceRequestId": "d8b7d414-b4d0-4d02-9f42-d1c3edb6459b",
  "message": "OTP validation is required. Please enter OTP to continue",
  "transactionId": null,
  "txnStatus": null,
  "serviceFlow": "LOGIN_POLICY",
  "status": "PAUSED",
  "language": "en",
  "code": "otp.validation.required",
  "mfsTenantId": "mfsPrimaryTenant",
  "transactionTimeStamp": "2023-02-09T10:18:56"
}
```

Error Codes

Error Code	Description
400	Bad request
500	Internal Server Error

Error scenario:

Http error code: 400.

Error Code: Authen01- Invalid credentials.

Response:

```
{ "txnStatus": "TF", "status": "FAILED", "language": "en", "mfsTenantId": "mfsPrimaryTenant", "errors": [ { "code": "AUTH_06", "message": "Invalid login credentials. Please try again.", "componentName": "user-authentication" } ], "transactionTimeStamp": "2023-07-06T20:30:00", "errorCode": "Authen01", "traceId": "19422597-1711-4bc6-bcab-baf6734549a8", "step": "get.userid.service", "errorUserMsg": "Invalid credentials.", "httpErrorCode": "400" }
```

2.1.1.3 Confirm OTP and Generate Token

Action: POST

Endpoint : /ums/v3/user/auth/login-confirm

Description

This API is used to confirm **OTP** and mapped device information and then generate token. This is used when the user login authentication requires **OTP** to access Web Application.

Request body

Fields	Type	Description	Example/Allowed Values
loginValidateRequest	API to validate OTP and generate token for device login.		
	Fields	Type	Description
	otp *	String	OTP received on registered mobile number. Length-6 digit
	resumeServiceRequestId *	String	Whenever an API is called, Comviva generates a unique service request ID for the API request. This unique ID traverses' components in the business process flow. It is useful when the process flow is paused for the user input and needs to be resumed when user has provided the input. Length-36 characters

Sample Request Body

```
{
  "otp": "343566",
  "resumeServiceRequestId": "8f96dbc4-edd1-45cb-b6b1-1e3a76f42e0c"
}
```

Response Body

Code: 200

Access token generated successfully.

Fields	Type	Description	Example/Allowed Values
LoginWithOTPResponseModel	Login response model with token		
language	String	Preferred language of the user. By default, system supports 3 languages English/French/Arabic.	En, ar, fr
lastLoginTime	String	Time stamp of last login of the user.	2020-10-13T02:47:56
message *	String	The API response message that shows the type of information returned.	OTP validation is required. Please enter OTP to continue
serviceFlow	String	A unique code that is internal to Comviva and identifies the business process flow associated to the API.	OTPLGIN
serviceRequestId *	String	Whenever an API is called,	f491f6b1-aa9b-43de-93b0-

Fields	Type	Description	Example/Allowed Values												
		<p>Comviva generates a unique service request ID for the API request. This unique ID traverses' components in the business process flow. It is useful when the process flow is paused for the user input and needs to be resumed when user has provided the input.</p> <p>Length-36 characters</p>	c85eda706a2c												
status *	String	The status to show whether the API call is a success, failure, in progress, or paused.	SUCCEEDED, FAILED, INPROGRESS, or PAUSED												
token	String	<div>TokenData</div> <table> <tr> <th>Fields</th><th>Type</th><th>Description</th><th>Example/Allowed Values</th></tr> <tr> <td>access_token *</td><td>String</td><td>Access token is used for token based authentication to allow an application to access an API. The application receives an Access Token after a user successfully authenticates and authorizes access. User can then pass this access token as a credential to perform any transactions in the application. Access tokens are short-lived tokens.</td><td>asdfqwrtwy</td></tr> <tr> <td>expires_in</td><td>Integer (\$int32)</td><td>Validity period (in seconds) for accessing the token.</td><td>299</td></tr> </table>		Fields	Type	Description	Example/Allowed Values	access_token *	String	Access token is used for token based authentication to allow an application to access an API. The application receives an Access Token after a user successfully authenticates and authorizes access. User can then pass this access token as a credential to perform any transactions in the application. Access tokens are short-lived tokens.	asdfqwrtwy	expires_in	Integer (\$int32)	Validity period (in seconds) for accessing the token.	299
Fields	Type	Description	Example/Allowed Values												
access_token *	String	Access token is used for token based authentication to allow an application to access an API. The application receives an Access Token after a user successfully authenticates and authorizes access. User can then pass this access token as a credential to perform any transactions in the application. Access tokens are short-lived tokens.	asdfqwrtwy												
expires_in	Integer (\$int32)	Validity period (in seconds) for accessing the token.	299												

Fields	Type	Description		Example/Allowed Values	
		refresh_token	String	Once the access token expires, you can use the refresh token to get the new access token. Refresh tokens also expire but they are supposed to live much longer than the access token. Max Length-256 characters	asdfqwrtwy
transactionId *	String	The unique id generated by the Comviva platform for the transaction. Length-20 characters		XX220316.0959.A15026	
txnStatus	String	Defines the five different statuses of the transaction. Not Applicable for Non-Financial APIs TI – Transaction Initiated TS – Transaction Succeeded TF – Transaction Failure TP – Transaction Paused TA – Transaction Ambiguous		Transaction Success	
userId	String	Unique ID generated by the system after successful user registration. Length-20 characters		US.k6GH1579603850092	

Sample Response Body

```
{
  "serviceRequestId": "8dd2d378-21d2-4bf8-a363-f023792e15bb",
  "message": "Login Successfully",
}
```

```

    "transactionId": null,
    "txnStatus": null,
    "serviceFlow": "LOGIN_POLICY",
    "status": "SUCCEDED",
    "language": "en",
    "token": {
      "access_token":
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Imp3cy1rZXktcHVibGljLTEifQ.eyJzZXJ2aWNlUmVxdWVzdElkIjoiOGY5NmRiYzQtZWRRkMS00NWNiLWI2YjEtMWUzYTc2ZjQyZTBjIiwidXNlcl9uYW1lIjoiOGY5NmRiYzQtZWRRkMS00NWNiLWI2YjEtMWUzYTc2ZjQyZTBjIiwiaXV0aG9yaXphdGlvdjByb2ZpbGVDb2RlIjoiU3Vic0RlZmFlbHQiLCJpZGVudGlmaWVyVmFsdWUiOiI3Nzc0Njg0MzU5IiwiaWRlbnRpbmZmlcllR5cGUiOiJNU01TRE4iLCJjYXRlZ29yeUNvZGU0IjVTVUJTIiwidXNlcklkIjoiVVMuNTMwMTY3NTM5OTE5ODU5NCIsImRldmVjZUlkIjpdWxsLCJhdXRob3JpdGllcyI6WyJST0xFX1VTRVlXSwiY2xpZW50X2lkIjoiQ29yZVdlYiIsInNjb3BlIjpbIkdFVF9VU0VSX0FORF9BQ0NPVU5UX0RFVEFJTfMiLCJTRUxGU0VUQVVU5EzBQ1RPUiIsIkZfVENIX1VTRVJfUUVFU1RJT04iLCJUE5DT1JSRUNUIiwiQVRNQ0FTSE9VVCIsIkFUTUNBU0hPVVRfVjQiLCJBRE1UWE5SRUZJRCJdLCJuYW1lIjoiU3Vic2NyaWJlciBIYWdlbmVzIiwiaWVhcmVhY29kZSI6IldFQiIsImV4cCI6MTY3NTkyMTU2NiwiianRpIjoiMzZM5ODRmYTYtNzFhYi00MmVmLTgyZDYtMTY1ZWUwZDRlMmZlIn0.WFcZiMSapOAMdouIKj2bKkmUdtWST78bY71-gN8OIpGIFmJXFJMX9AwwhEPtB6vsqJ5RQebVyn26OlMopD_IzRKdSxx4V5hfaFgx1kcIQocvlgmGx14aW-QcUJxcqVoIU3hPddj0iZ1Hz0Tlab8czJ5edOkk0qIDTh8BvutwVKLAOYHl6dpKwxn6_PHSzFpoH9qi9jb6fgEFXBDIKhDbj9CqPUU17iAvbEBD42wkJQyemqu4-9eETbsbugX5PVjo12_CYUdbLAJgmGwwlEYlFOQ2GTk6bPdfzG5ImhNRigv9P9kK5aeCdb2rM3qZcqQ2GI_FvLF1UQS0zDVhNKWnA",
      "expires_in": 2999,
      "refresh_token":
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Imp3cy1rZXktcHVibGljLTEifQ.eyJzZXJ2aWNlUmVxdWVzdElkIjoiOGY5NmRiYzQtZWRRkMS00NWNiLWI2YjEtMWUzYTc2ZjQyZTBjIiwidXNlcl9uYW1lIjoiOGY5NmRiYzQtZWRRkMS00NWNiLWI2YjEtMWUzYTc2ZjQyZTBjIiwiaXV0aG9yaXphdGlvdjByb2ZpbGVDb2RlIjoiU3Vic0RlZmFlbHQiLCJpZGVudGlmaWVyVmFsdWUiOiI3Nzc0Njg0MzU5IiwiaWRlbnRpbmZmlcllR5cGUiOiJNU01TRE4iLCJjYXRlZ29yeUNvZGU0IjVTVUJTIiwidXNlcklkIjoiVVMuNTMwMTY3NTM5OTE5ODU5NCIsImRldmVjZUlkIjpdWxsLCJhdXRob3JpdGllcyI6WyJST0xFX1VTRVlXSwiY2xpZW50X2lkIjoiQ29yZVdlYiIsInNjb3BlIjpbIkdFVF9VU0VSX0FORF9BQ0NPVU5UX0RFVEFJTfMiLCJTRUxGU0VUQVVU5EzBQ1RPUiIsIkZfVENIX1VTRVJfUUVFU1RJT04iLCJUE5DT1JSRUNUIiwiQVRNQ0FTSE9VVCIsIkFUTUNBU0hPVVRfVjQiLCJBRE1UWE5SRUZJRCJdLCJhdGkiOiIzMzk4NGZhNi03MWFjLTQyZWYtODJkNi0xNjVlZTBkNGUyZmUiLCJuYW1lIjoiU3Vic2NyaWJlciBIYWdlbmVzIiwiaWVhcmVhY29kZSI6IldFQiIsImV4cCI6MTY3NTk0ODU2NiwiianRpIjoiZmUzNDFlYmItN2NlYS00MTBjLWFKZmItZmJmYjA2MmI0NjFlIn0.UrYcUPmNEjr78tegJ6nGt897SVPF1_MRz5asJ-1Epj62SqV3dj97fgivov5tvcSWxbDFNeT4cXINKHnIuI6rZUuu5G6q2E_fFI3DH4b4PIifvTJ58BQnvG1DlL6XD14I3y4YQ3n4Cm4WkmMzRFtyDiyvzSC2Yg3lqx9bqLW4G7_w80A6vn0SYZqFd2jEpt7WQCTspceK952vp9OWg6JFzVMZaZ49YuounMBLgBEBwxQaMrnfGSbI4_ODBwrb29bus0I4EG_SMroyrZMjkX-lqkVISXKZJjPdu3ZvULniY54wxjN2uUUEYr1MK1R7X6qXkE11rO5lBeCaPzCQzuKNg"
    },
    "lastLoginTime": null,
    "userId": "US.5301675399198594",
    "sessionIdList": null
  }

```

Error Codes

Error Code	Description
400	Bad request
500	Internal Server Error

Error scenario:

Http error code: 400.

Error Code: Generic06- Invalid input.

Response:

```
{
  "status": "FAILED",
  "errors": [
    {
      "code": "sfm.errors.invalid.resume.service.request.id",
      "message": "resume request id mandatory cannot have null or empty
value",
      "componentName": "SFM"
    }
  ],
  "errorCode": "Generic06",
  "errorUserMsg": "Invalid input.",
  "httpErrorCode": "400"
}
```

2.2 BillPayment Flow:

Current Flow:

- On UI (named as CBP / web-portal-application) : to fetch bill details of accounts and to pay pending bills in bulk, request is initiated from UI and it is stored in CBP DB for further processing. At this point request is just queued, no fetching or processing has take place.
- Second component scheduler will then pick these request from DB on timely basis and communicates with communicator component which will make hit to backend financial system(mobiquity) for Payment and third party system(Biller platform) for actual processing

New Flow:

Process flow will be similar to existing approach , but only change in integration.

- On UI (named as CBP / web-portal-application) : to fetch bill details of accounts and to pay pending bills in bulk, request is initiated from UI and it is stored in CBP DB for further processing. At this point request is just queued, no fetching or processing has take place.
- Second component scheduler will then pick these request from DB on timely basis and communicates with communicator component which will make hit to backend financial system(mobiquity) to fetch bill details or Payment request instead of Biller Platform.
- In fetch bill details it is simple API hit to mobiquity to fetch bill details instead of biller platform.
- For Payment also, communicator will send payment hit to just mobiquity. Not to biller. Mobiquity will in turn send hit to biller platform for any kind of third party integration. Communicator will not interact with biller platform.
- While sending hit to mobiquity, CBP has to ensure a unique reference ID is created, stored in mflex DB corresponding to each bill request and is sent to mobiquity to manage for any disputes in future.
- Communicator will just wait for Mobiquity response of Payment API. Response handling is mentioned below:
 - If it is Failed, then payment is also marked failed in mflex DB.
 - If success/pending or fulfilment initiated : then it is assumed that request is properly posted in mobiquity and now to check status of payment etc. mobiquity DB/ reports has to be inspected. All mobiquity reports will show corresponding unique ID created by CBP so as to track the transaction in mobiquity.
 - If no response from mobiquity for payment request, then all such transactions will be enquired in mobiquity with unique reference ID created at time of payment.
 - If success/fail response of enquiry then status is cleared in mFlex DB.
 - But if no response of enquiry, for a predefined configurable retries, then those are left uncleared in mflex DB, and all such transactions can be checked in Mobiquity reports.

Note :

Scheduler will pick only those records for enquiry for which they didn't get response back from mobiquity platform. No retry feature will be available in this scenario. If scheduler gets response back for payment API, then also any reference to the payments and third party response should be from mobiquity reports. It should not be checked from Bulk Billpay UI.

Expectations from partner :

- Communicator code need to be updated so it can now integrate with new mobiquity platform with updated API specifications to fetch bill details and for payment. These will be routed to mobiquity instead of direct biller integration.
- Logic of unique ID creation to be sent in mobiquity payment APIs

- Mobiquity response handling and Ambiguous payments settlement logic (enquiry , **not refund**) need to be updated accordingly in BillPay code.
- Email and SMS handling should be there as per current setup : integration with Email server and SMSGW

2.3 Password Management Flow:

When any user forgets his password then user should have flexibility to get these reset via forget password feature on login page. Also once he login, then user can change his password on his own.

Current Flow:

In current platform, all password management is done at IDP page.

New Flow and Expectations from Partners:

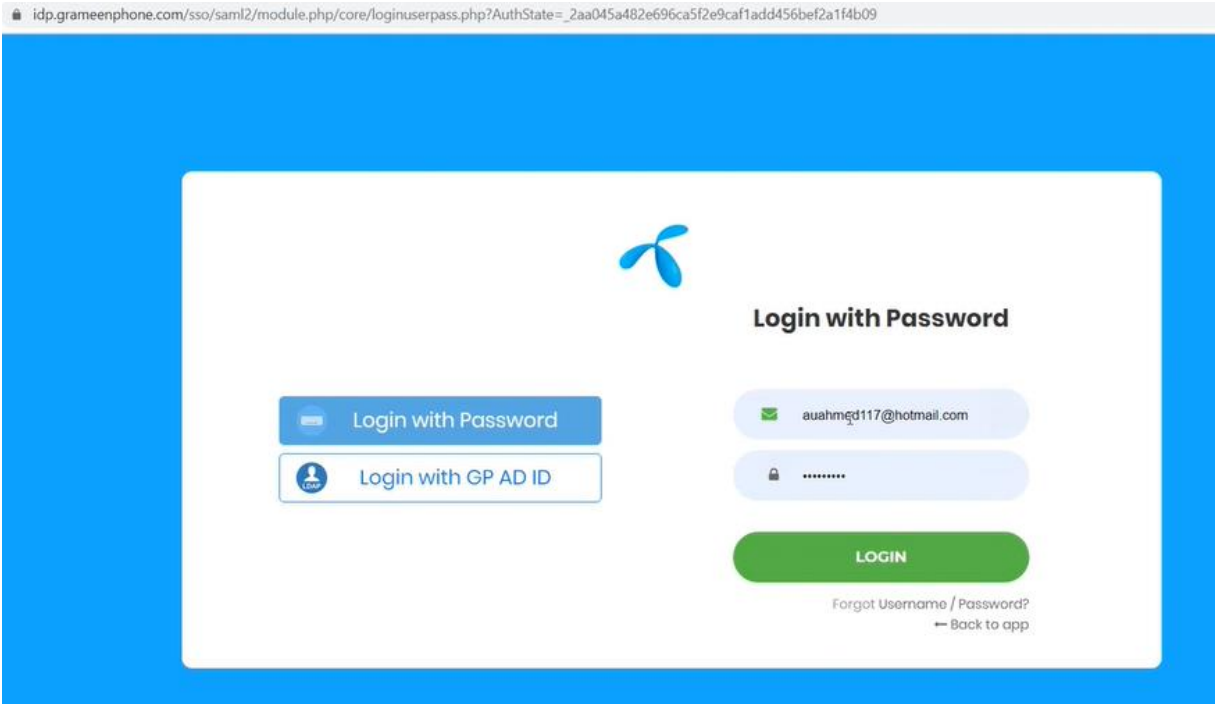
Password management to be incorporated on Bulk UI portal in 2 ways:

1. Give forgot password option on login page.
2. User himself once logged in, can be given with Change password option

All 2 APIs need to have direct integration with Mobiquity for forgot password and change password.

Mobiquity will share APIs to achieve same. Partner can design pages.

Forgot password option may come like below:



Change Own password or reset password pages are new.

2.3.1

API Details:

2.3.1.1 Change Authentication Factor

Action: POST

Endpoint: /ums/v2/user/auth/change-credential

Description

Self-initiated change authentication (Pin/Password). User can change his old password to new password without answering security questions. This API is usually called for changing the default password given by mobility system for the first login after successful user registration.

Request body

Fields	Type	Description	Example/Allowed Values
Authorization	String	Authorization Token to pass header	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9ImltpZC6Iimp3cy1r
changeAuthenticationFactorModel	String	Change Authentication service body. Cannot be empty.	
ChangeAuthenticationFactorModel	String	Change Authentication Request Model	
requestedBy*	String	SELF for self request else unique ID of Requestor from Mobility System.	SELF
workspaceId*	String	Workspace is the classification of all users at high level. All users are clubbed together under three workspaces SUBSCRIBER/BUSINESS/ADMIN	SUBSCRIBER/BUSINESS/ADMIN
identifierType*	String	Access identifier type of the user like MSISDN/EMAIL/LOGINID/OTHRID to perform reset .	MSISDN/EMAIL/LOGINID/OTHRID
identifierValue*	String	Access identifier value associated with the identifier type selected. Mobile Number:8-15 Email: Max length-40 characters LOGINID-min length is 3 & max length is 20	7766990546
language	String	Preferred language of the user. By default, system supports 3 languages	En, Fr, Ar

Fields	Type	Description	Example/Allowed Values
		English/French/Arabic.	
oldauthenticationValue*	String	Old authentication value (PIN/Password) Of the user. Default length of pin is 4 Min password length is 5 & Max password length is 10 Pin/password also can be configured using security profile	4568
newauthenticationValue*	String	New authentication value (PIN/Password) Of the user. Default length of pin is 4 Min password length is 5 & Max password length is 10 Pin/password also can be configured using security profile	1357
confirmedauthenticationValue*	String	Re-Enter new authentication value to confirm. Default length of pin is 4 Min password length is 5 & Max password length is 10 Pin/password also can be configured using security profile	1357

Sample Request Body

```
{ "confirmedAuthenticationValue": "Com@135", "identifierType": "LOGINID", "identifierValue": "BUSADM7772569881", "language": "en", "newAuthenticationValue": "Com@135", "oldAuthenticationValue": "000000", "requestedBy": "SELF", "workspaceId": "ADMIN" }
```

Response Body

Code: 200

Service Response for Change Authentication Factor.

Fields	Type	Description	Example/Allowed Values
AuthenticationResponse	String	User Authentication Response Model	
identifierType*	String	Access identifier type of the user like MSISDN/EMAIL/LOGINID/OTHERID to perform reset .	MSISDN/EMAIL/LOGINID/OTHERID

Fields	Type	Description	Example/Allowed Values
identifierValue*	String	Access identifier value associated with the identifier type selected. Mobile Number:8-15 characters Email: Max length-40 characters LOGINID-min length is 3 & max length is 20	7766990546
message*	String	The API response message that shows the type of information returned. <i>No Specific length</i>	Your password is changed successfully.
serviceFlow	String	A unique code that is internal to Comviva and identifies the business process flow associated to the API.	CHANGEAUTHFACTOR/ RESETAUTHFACTOR/ RESETAUTHBYADMIN
serviceRequestId*	String	Whenever an API is called, Comviva generates a unique service request ID for the API request. This unique ID traverses' components in the business process flow. It is useful when the process flow is paused for the user input and needs to be resumed when user has provided the input. <i>Length-36 characters</i>	f491f6b1-aa9b-43de-93b0-c85eda706a2c
status*	String	The status to show whether the API call is a success, failure, in progress, or paused.	SUCCEEDED, FAILED, INPROGRESS, or PAUSED
workspaceId*	String	Workspace is the classification of all users at high level. All users are clubbed together under three workspaces SUBSCRIBER/BUSINESS/ADMIN	SUBSCRIBER/BUSINESS/ADMIN
userId*	String	Unique ID generated by the system after successful user registration. Length-20(System generated)	US.k6GH1579603850092
userName	String	Name of the user.	Not set in response

Fields	Type	Description	Example/Allowed Values
	g		
language	String	Preferred language of the user. By default, system supports 3 languages English/French/Arabic.	En, Fr, Ar

Sample Response Body

```
{
  "serviceRequestId": "ddc143ec-eeab-4b7d-b151-3831fee0f562",
  "message": "Authentication factor is successfully changed",
  "serviceFlow": "CHANGEAUTHFACTOR",
  "status": "SUCCEEDED",
  "userId": "US.7611675338672421",
  "userName": null,
  "workspaceId": "ADMIN",
  "identifierType": "LOGINID",
  "identifierValue": "BUSADM7772569881"
}
```

Error Codes

Error Code	Description
400	Bad request
500	Internal Server Error

Error scenario:

Http error code: 400

Error Code: Generic04- Mandatory field can't be empty.

Response:

```
{
  "txnStatus": "TF",
  "status": "FAILED",
  "language": "en",
  "mfsTenantId": "mfsPrimaryTenant",
  "errors": [
    {
      "code": "WORK02",
      "message": "Workspace is mandatory",
      "componentName": "user-authentication"
    }
  ],
  "transactionTimeStamp": "2023-03-30T01:23:13",
  "errorCode": "Generic04",
  "traceId": "352e62e0-7053-497d-8f2d-0ce13d1ac446",
  "step": "get.userid.service",
  "errorUserMsg": "Mandatory field can't be empty",
  "httpErrorCode": "400"
}
```

2.3.1.2 Forget Authentication API

This section will tell you all APIs to be called in sequence to reset login credentials

1. Initiate reset authentication via OTP
2. Validate OTP
3. Confirm new authentication values

2.3.1.2.1 Validate Self-Set Authentication Value with OTP

Action: POST

Endpoint: /v2/ums/user/auth/**self-set**-auth/initiate

Description

This API is used to initiate change authentication value (Pin/Password) with two factor authentication (OTP).

Request body

Fields	Type	Description	Example/Allowed Values
Authorization	String	Authorization Token to pass header	eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Impr3cy1r
adminResetAuthenticationFactor Model	String	Self-initiated change authentication with otp service body. Cannot be empty.	
SelfSetAuthenticationFactorWith OtpModel	String	Self-Set Authentication with OTP Request Model	
bearerCode *	String	Access bearer channel from which the request is raised.	Web, USSD, Core Web, Mobile App
identifierType*	String	Access identifier type of the user like MSISDN/EMAIL/LOGINID/OTHERID to perform reset.	MSISDN/EMAIL/LOGINID/OTHERID
identifierValue*	String	Access identifier value associated with the identifier type selected. Mobile Number:8-15 Email: Max length-40 characters LOGINID-min length is 3 & max length is 20	7766990546
language	String	Preferred language of the user. By default, system supports 3 languages English/French/Arabic.	En, Fr, Ar
workspaceId*	String	Workspace is the classification of all users at high level. All users are clubbed together under three workspaces	SUBSCRIBER/BUSINESS/ADMIN

Fields	Type	Description	Example/Allowed Values
		SUBSCRIBER/BUSINESS/ADMIN	

Sample Request Body

```
{
  "requestedBy": "SELF",
  "workspaceId": "BUSINESS",
  "identifierType": "MSISDN",
  "identifierValue": "7779555603",
  "language": "en",
  "bearerCode": "MOBILE",
  "deviceInfo": {
    "appName": "MobiquityPayChannel",
    "appVersion": "10.03.0.01",
    "deviceId": "17cf27d2-871b-42d2-aa4c-2930bed0f5e6",
    "isPublicDevice": "N",
    "model": "Google sdk_gphone_x86",
    "os": "ANDROID"
  }
}
```

Response Body

Code: 200

Self-initiated change authentication with OTP response.

Fields	Type	Description	Example/Allowed Values
SelfSetAuthenticationFactorWithOtpResponse	String	Self-Set Authentication with OTP Response model.	
Code	String	Step code where the request is paused for user input.	otp.validation.required OR security.answer.required
language	String	Preferred language of the user. By default, system supports 3 languages English/French/Arabic.	En, Fr, Ar
message*	String	The API response message that shows the type of information returned.	OTP validation is required. Please enter "OTP" to continue OR "Answer" for security question is required. Please enter the correct answer to continue.
mfsTenantId	String	MFS Tenant ID of the user	mfsPrimaryTenant
originalServiceRequestId*	String	The Original Service Request ID Length-36 characters	da371e6e-db02-472b-94c3-2ad2f79dfd09
serviceFlow	String	A unique code that is internal to Comviva and identifies the business process flow associated to the API.	SELFSETAUTHMFA
serviceRequestId*	String	Whenever an API is called, Comviva generates a unique service request ID for the API request. This unique ID traverses' components in the business process flow. It is useful when the process flow is paused for the user input and needs to be resumed when user has provided the input.	f491f6b1-aa9b-43de-93b0-c85eda706a2c

Fields	Type	Description	Example/Allowed Values
		Length-36 characters	
status*	String	The status to show whether the API call is a success, failure, in progress, or paused.	SUCCEEDED, FAILED, INPROGRESS, or PAUSED
transactionTimeStamp *	String	The date and time of transaction in the format YYYY-MM-DD and HH:MM:SS respectively.	2023-01-31T15:49:45

Sample Response Body

```
{
  "serviceRequestId": "796e4d22-426f-46e3-bae5-1fe116988a95",
  "mfsTenantId": "mfsPrimaryTenant",
  "language": "en",
  "serviceFlow": "SELFSETAUTHMFA",
  "transactionTimeStamp": "2023-02-02T11:38:20",
  "originalServiceRequestId": "796e4d22-426f-46e3-bae5-1fe116988a95",
  "code": "otp.validation.required",
  "message": "OTP validation is required. Please enter OTP to continue",
  "status": "PAUSED"
}
```

Error Codes

Error Code	Description
400	Bad request
500	Internal Server Error

Error scenario:

Http error code: 400.

Error Code: invalid.value- Unexpected value 'QR_CODE_TYPE_DYNAMIC'.

Response:

```
{ "traceId": null, "errorUserMsg": "Bad Request", "errors": [ { "code": "invalid.value", "message": "Unexpected value 'LOGINI'" } ], "step": null, "referenceId": null, "status": "FAILED", "httpErrorCode": "400" }
```

2.3.1.2.2 Validate OTP for Self-Set Authentication Value

Action: POST

Endpoint: /v2/ums/user/auth/self-set-auth/validate-otp

Description

This API is used to validate the OTP entered for self-initiated change authentication value (Pin/Password).

Request body

Fields	Type	Description	Example/Allowed Values
adminResetAuthenticationFactorModel	String	Validate OTP for self-initiated change authentication with OTP service body. Cannot be empty.	
ValidateOtpSelfSetAuthenticationFactorWithOtpModel	String	Self Set Authentication with OTP Request Model	
language *	String	Preferred language of the user. By default, system supports 3 languages English/French/Arabic.	En, Fr, Ar
otp *	String	OTP received on registered mobile number. Length-6	213131
resumeServiceRequestId *	String	Whenever an API is called, Comviva generates a unique service request ID for the API request. This unique ID traverses' components in the business process flow. It is useful when the process flow is paused for the user input and needs to be resumed when user has provided the input. Length-36 characters (System generated)	f491f6b1-aa9b-43de-93b0-c85eda706a2c

Sample Request Body

```
{ "resumeServiceRequestId": "796e4d22-426f-46e3-bae5-1fe116988a95", "otp": "704812", "language": "fr" }
```

Response Body

Code: 200

Self-set authentication with OTP response.

Fields	Type	Description	Example/Allowed Values
ValidateOtpSelfSetAuthenticationFactorWithOtpResponse	String	Self-Set Authentication with OTP Response model.	
Code	String	Step code where the request is paused for user input.	new.authentication.value.required
language	String	Preferred language of the user. By default, system supports 3 languages English/French/Arabic.	En, Fr, Ar
message*	String	The API response message that shows the type of information returned.	new authentication value is required. Please enter new & confirmation values to continue.
mfsTenantId	String	MFS Tenant ID of the user	mfsPrimaryTenant
originalServiceRequestId*	String	The Original Service Request ID Length-36 characters	da371e6e-db02-472b-94c3-2ad2f79dfd09
serviceFlow	String	A unique code that is internal to Comviva and identifies the business process flow associated to the API.	SELFSETAUTHMFA
serviceRequestId*	String	Whenever an API is called, Comviva generates a unique service request ID for the API request. This unique ID traverses' components in the business process flow. It is useful when the process flow is paused for the user input and needs to be resumed when user has provided the input. Length-36 characters	f491f6b1-aa9b-43de-93b0-c85eda706a2c
status*	String	The status to show whether the API call is a success, failure, in progress, or paused.	SUCCEEDED, FAILED, INPROGRESS, or PAUSED
transactionTimeStamp*	String	The date and time of transaction in the format YYYY-MM-DD and HH:MM:SS respectively.	2023-01-31T15:49:45

Sample Response Body

```
{
  "serviceRequestId": "e4e90030-e4a5-41e6-8bcd-9ac0e8128034",
  "mfsTenantId": "mfsPrimaryTenant",
  "language": "fr",
```

```

    "serviceFlow": "SELFSETAUTHMFA",
    "transactionTimeStamp": "2023-02-02T11:38:45",
    "originalServiceRequestId": "796e4d22-426f-46e3-bae5-1fe116988a95",
    "code": "new.auth.value.required",
    "message": "Une nouvelle valeur d'authentification est requise.
    Veuillez entrer de nouvelles valeurs et des valeurs de confirmation pour
    continuer",
    "status": "PAUSED"
  }

```

Error Codes

Error Code	Description
400	Bad request
500	Internal Server Error

Error scenario:

Http error code: 400.

Error Code: Generic05- No data found.

Response:

```

{
  "status": "FAILED",
  "errors": [
    {
      "code": "sfm.errors.resume.service.request.id.not.found",
      "message": "resume request id not found",
      "componentName": "SFM"
    }
  ],
  "errorCode": "Generic05",
  "errorUserMsg": "No data found",
  "httpErrorCode": "400"
}

```

2.3.1.2.3 Confirm Authentication Values after OTP Validation

Action: POST

Endpoint: /v2/ums/user/auth/self-set-auth/confirm

Description

This API is used to confirm new authentication value for self-initiated change authentication value (Pin/Password) after OTP validation.

Request body

Fields	Type	Description	Example/Allowed Values
adminResetAuthenticationFactorModel	String	set authentication value for Self-set authentication with otp service body. Cannot be empty.	
SetAuthenticationValueWithOtpModel	String	Self-Set Authentication with OTP Request Model	

Fields	Type	Description	Example/Allowed Values
confirmedAuthenticationValue*	String	<p>Re-enter new password for confirmation.</p> <p>Default length of pin is 4</p> <p>Min password length is 5 & Max password length is 10</p> <p>Pin/password also can be configured using security profile</p>	1234
newAuthenticationValue*	String	<p>Enter new password/PIN</p> <p>Can be configured using security profile</p> <p>Default length of pin is 4</p> <p>Min password length is 5 & Max password length is 10</p> <p>Pin/password also can be configured using security profile</p>	1234
language	String	Preferred language of the user. By default, system supports 3 languages English/French/Arabic.	En, Fr, Ar
resumeServiceRequestId*	String	<p>System Generated Unique Id which determines the service flow</p> <p>Length- 36 characters (System generated)</p>	f491f6b1-aa9b-43de-93b0-c85eda706a2c

Sample Request Body

```
{ "resumeServiceRequestId": "e4e90030-e4a5-41e6-8bcd-9ac0e8128034", "confirmedAuthenticationValue": "1358", "newAuthenticationValue": "1358", "language": "fr" }
```

Response Body

Code: 200

Self-set authentication with OTP response.

Fields	Type	Description	Example/Allowed Values
SetAuthenticationValue WithOtpResponse	String	Self Set Authentication with OTP Response model	
language	String	Preferred language of the user. By default, system supports 3 languages English/French/Arabic.	En, Fr, Ar
message*	String	The API response message that shows the type of information returned.	Authentication factor is successfully changed
mfsTenantId	String	MFS Tenant ID of the user	mfsPrimaryTenant
originalServiceRequestId *	String	The Original Service Request ID Length-36 characters	da371e6e-db02-472b-94c3-2ad2f79dfd09
serviceFlow	String	A unique code that is internal to Comviva and identifies the business process flow associated to the API.	SELFSETAUTHMFA
serviceRequestId*	String	Whenever an API is called, Comviva generates a unique service request ID for the API request. This unique ID traverses' components in the business process flow. It is useful when the process flow is paused for the user input and needs to be resumed when user has provided the input. Length-36 characters	f491f6b1-aa9b-43de-93b0-c85eda706a2c
status*	String	The status to show whether the API call is a success, failure, in progress, or paused.	SUCCEEDED, FAILED, INPROGRESS, or PAUSED
transactionTimeStamp *	String	The date and time of transaction in the format YYYY-MM-DD and HH:MM:SS respectively.	2023-01-31T15:49:45

Sample Response Body

```
{
  "serviceRequestId": "19ce96ff-af45-40f7-a7d4-7e094292aa53",
  "mfsTenantId": "mfsPrimaryTenant",
  "language": "fr",
  "serviceFlow": "SELFSETAUTHMFA",
  "transactionTimeStamp": "2023-02-02T11:38:48",
  "originalServiceRequestId": "796e4d22-426f-46e3-bae5-1fe116988a95",
  "code": "AUTH_04",
  "message": "Le facteur d'authentification a bien été modifié",
  "status": "SUCCEEDED"
}
```

Error Codes

Error Code	Description
400	Bad request
500	Internal Server Error

Error scenario:

Http error code: 400

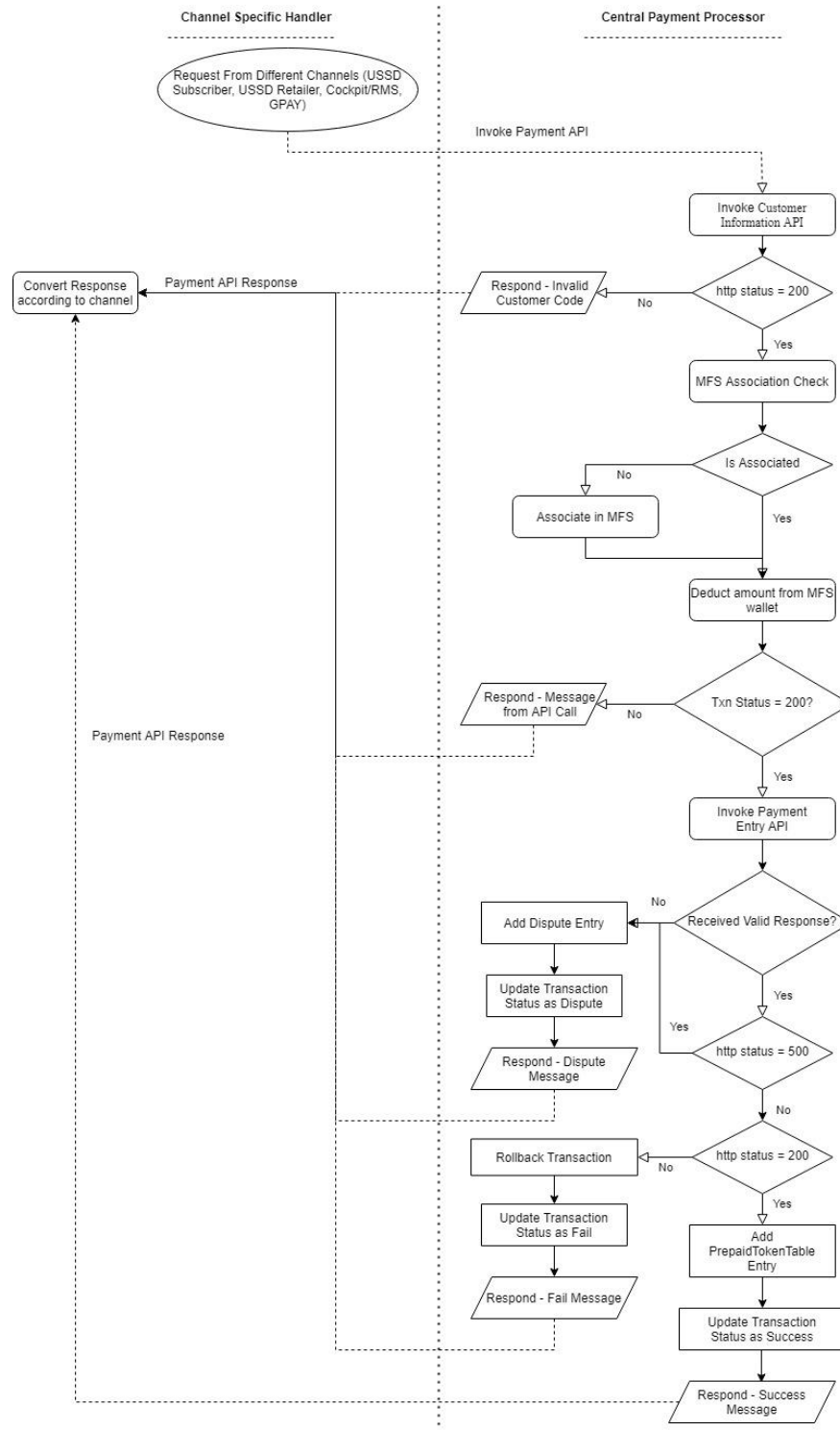
Error Code: Generic05- No data found.

Response:

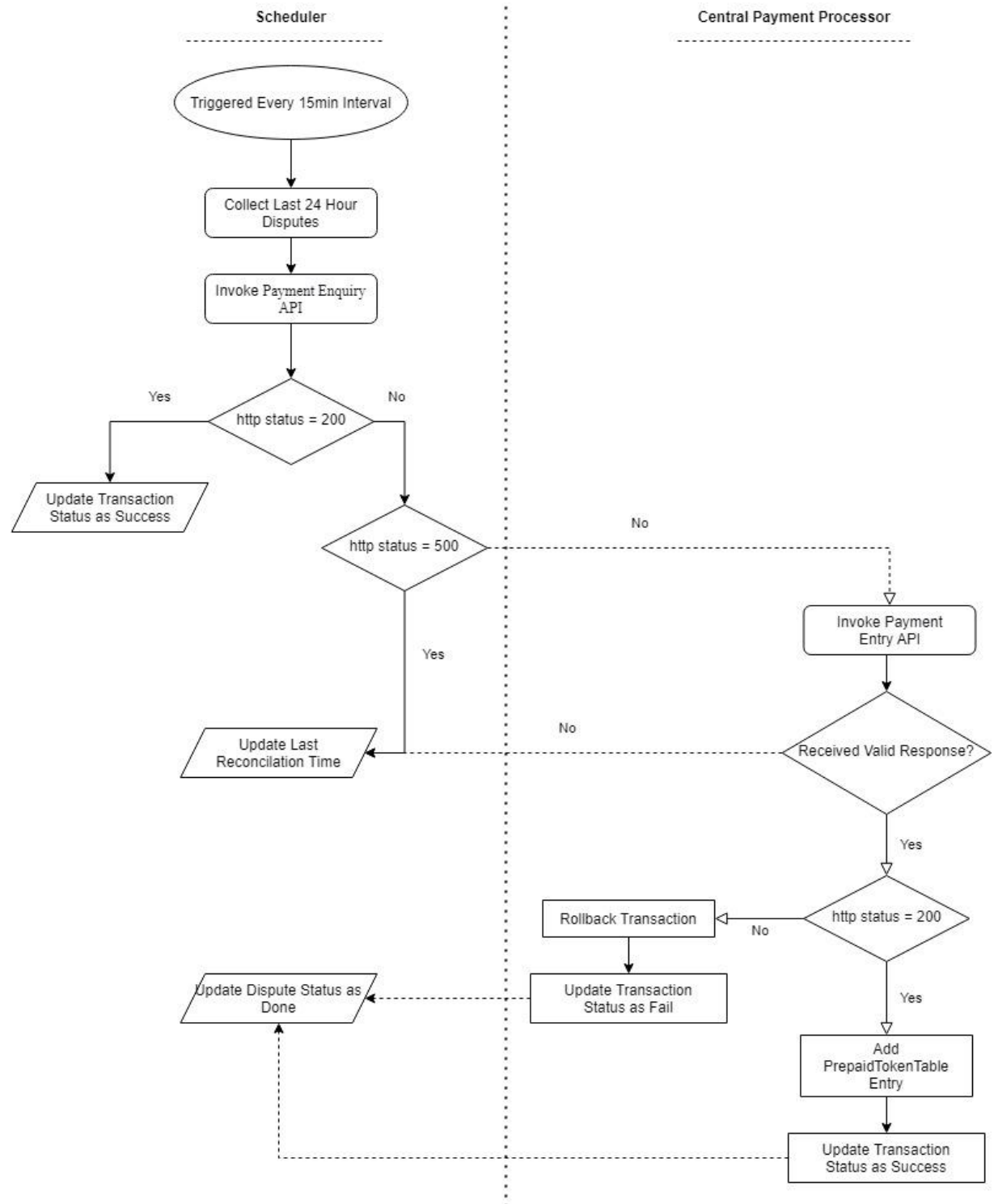
```
{
  "status": "FAILED",
  "errors": [
    {
      "code": "sfm.errors.resume.service.request.id.not.found",
      "message": "resume request id not found",
      "componentName": "SFM"
    }
  ],
  "errorCode": "Generic05",
  "errorUserMsg": "No data found",
  "httpErrorCode": "400"
}
```

2.4 Existing Flow Diagrams:

Flow diagram of Billpayment Request



Call flow of scheduler : to pick bulk request(fetch bill and payment) initiated from UI



Note:

- All technical training session happened with Wipro will be shared with partner that will give detailed view of current implemented request flow .
- Postman collection of existing system APIs is also shared.
- Git code link of existing system will be shared by Comviva to partner

3 Technical Solution Requirement

3.1 Solution Overview

the solution must be created using open-source technologies widely accepted and used by the industry.

3.1.1 Tech stack to be used to deploy and run existing code

Comviva suggest to use below listed technology stack for the development of the application:

Angular 9.1.1
npm 6.13.4
Node 12.16.1
tomcat 9 or above
Jdk 14
MySql

3.2 Deployment Specifications of existing module

The solution must be deployed with below steps in order.

3.2.1 DB Readiness


- Install MySql in your local system.
- Install MySql Workbench in your local system.
- execute Below all the files in MySQL Workbench for setup database in your local system.



 mflex_bulk_bill.sql



 mflex_core.sql



 mflex_report.sql
 Z

3.2.2

Application readiness

- There are Four Modules which need to be deployed in same order
- 1. web application
- 2. async job
- 3. mfs-communication
- 4. gpay-C

Steps to deploy:

- (Web Application – master)
 - Go in this folder
 1. cd src/ng
 2. open command prompt or git bash
 3. npm init -y
 4. npm install
 5. npm run dev
 - Now, open the project and do some changes in .properties (DB configuration, etc...)
 - run main class which have @SpringBootApplication annotations.
 - open any Browser and hit : <http://localhost:8080/>

3.3

Project Specific Scope

3.3.1

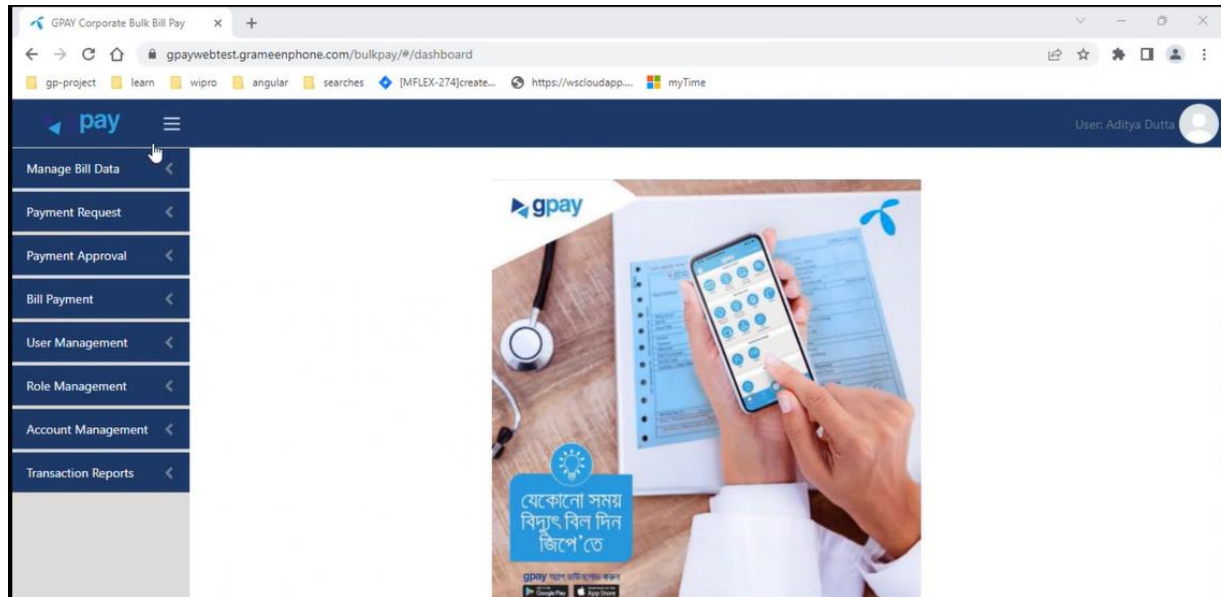
Current Implemenation

All technical hands-on recording will be shared with Parner

3.3.2

Graphical User Interface

GUI should be as below:



Rest can be referred from Recordings:

3.3.3 Training



Once the application is delivered on site, the partner must conduct technical training to the Comviva development team and provide necessary information and code walk through for changes made in existing Code.

3.3.4 Documentation

Flow diagram has been attached above to explain process flow of Request

3.3.5 Postman Collection

Attached is postman collection is existing system APIs to integrate within system and with biller platform



 Communicator Desco-postpaid.po
 copy.postman_collestan_collection.js

Note: Comviva will share it's API (Fetch bill, payment, ambiguous settlement)

3.4 Testing and Acceptance

3.4.1 System Integration Testing

Application must pass all the test cases defined by partner's internal QA. Comviva QA will audit the report and also may execute their own designed cases as per the functionalities if required, Partner must fix all issues raised by Comviva QA.

Note: Any bug identified during QA testing which is present in existing/new code must be fixed by Partner.

3.4.2 User Acceptance Procedure

Once the application is cleared by Comviva QA, it must be offered to customer for their testing and acceptance. User acceptance test cases must be shared by partner and passed in order to get user acceptance certificate. All UAT defects must be fixed by Partner.

3.5 Risks, Impact and Mitigation

Partner must identify and define all the risks and share its impact as well as mitigation plan.

3.6 Responsibility Matrix

Partner must prepare and share responsibility matrix for various tasks required to be executed during the lifecycle of this project.

3.7 Out of Scope

Third party (Biller)APIs integration would be out of scope for partner, though partner must define their own list of out of scope items and share with us.

All third party integrations will be under mobility scope. Partner has to just integrate with Mobility Payment APIs.

Reporting UI shown in demo is out of scope of partner . Mobility will support all report related need through existing reports present in product.

3.8 Code management

Partner must commit and update all the codes, designs, and documents into Comviva version control systems (GIT) and provide necessary access information for Comviva development team. Partner should adhere to all the Comviva software development guidelines and follow GDPR processes.

3.9 Project handover

Once the project is completed, partner must handover updated software to Comviva and conduct various workshops (technical, functional, non-functional etc.) for teams and provide them hands on experience and KT to enable them to take care of future change requests as well as bug fixes. This will be for changes done by Partner on top of existing code.

3.10 Project Timelines

This project must be completed within the defines and agreed timelines. The project is expected to be completed within 30 days.

3.11 Support and maintenance

- Partner will be responsible for UAT as well as production defects
- Partner must provide technical support and maintain the project upto 1 year after delivery.
- Partner must assign a dedicated team to develop and support this project
- Production defects must be fixed with in defined SLA.

4 Document Change History

Version	Change Type (A/M/D)	Change Description	Prepared By	Reviewer	Approved by	Date
1.0	A	Initial version	Anshika Aggarwal	Pranay Agrawal	Kamal Kishor Arya	27-July-2023

Disclaimer

Copyright © 2021: Comviva Technologies Limited 5th, 7th & 8th floor, Capital CyberScape, Village Ullahwas, Sector 59, Golf Course Extn. Road, Gurugram – 122102, Haryana, India.

All rights about this document are reserved and shall not be, in whole or in part, copied, photocopied, reproduced, translated, or reduced to any manner including but not limited to electronic, mechanical, machine readable, photographic, optic recording or otherwise without prior consent, in writing, of Comviva Technologies Ltd (the Company).

The information in this document is subject to changes without notice. This describes only the product defined in the introduction of this documentation. This document is intended for the use of prospective customers of the Company Products Solutions and or Services for the sole purpose of the transaction for which the document is submitted. No part of it may be reproduced or transmitted in any form or manner whatsoever without the prior written permission of the company. The Customer, who/which assumes full responsibility for using the document appropriately. The Company welcomes customer comments as part of the process of continuous development and improvement.

The Company, has made all reasonable efforts to ensure that the information contained in the document are adequate, sufficient and free of material errors and omissions. The Company will, if necessary, explain issues, which may not be covered by the document. However, the Company does not assume any liability of whatsoever nature, for any errors in the document except the responsibility to provide correct information when any such error is brought to company's knowledge. The Company will not be responsible, in any event, for errors in this document or for any damages, incidental or consequential, including monetary losses that might arise from the use of this document or of the information contained in it.

This document and the Products, Solutions and Services it describes are intellectual property of the Company and/or of the respective owners thereof, whether such IPR is registered, registrable, pending for registration, applied for registration or not.

The only warranties for the Company Products, Solutions and Services are set forth in the express warranty statements accompanying its products and services. Nothing herein should be construed as constituting an additional warranty. The Company shall not be liable for technical or editorial errors or omissions contained herein.

The Company logo is a trademark of the Company. Other products, names, logos mentioned in this document, if any, may be trademarks of their respective owners.

Copyright © 2021: Comviva Technologies Limited. All rights reserved.

Thank You

Visit us at www.comviva.com