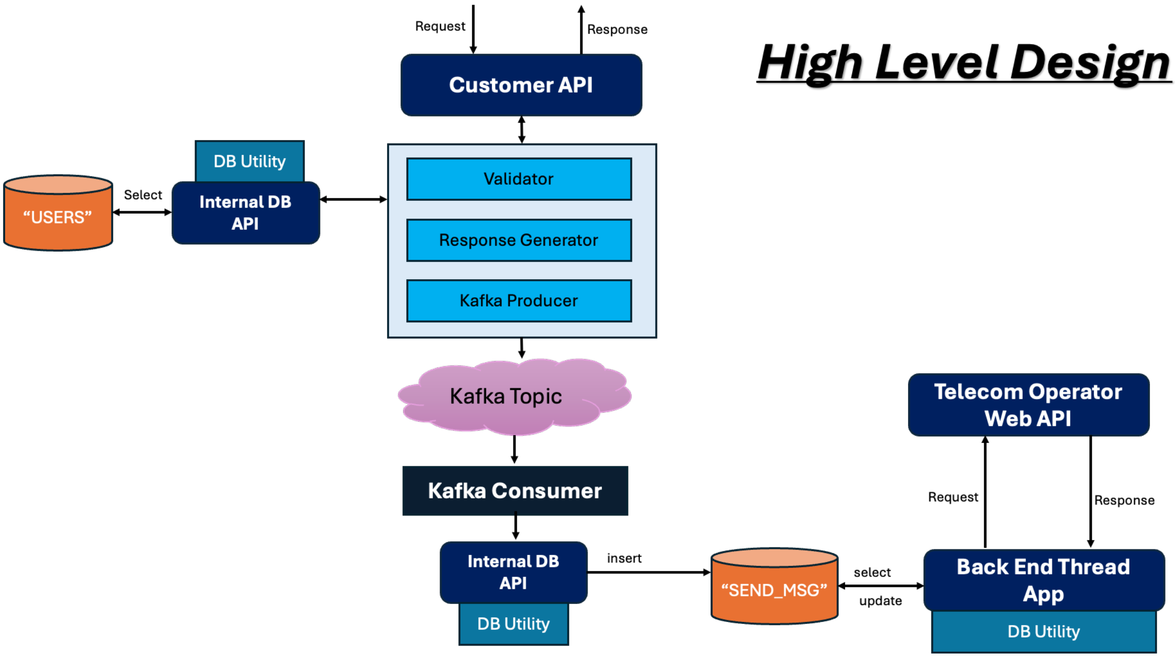
Demo Project – Scope of Work

This demo project is a miniature version of simulating our SMS Platform. This project has five modules,

I. **Customer API** (to receive SMS from our customers)
II. **Kafka Consumer** (pushing the request to Kafka for further processing)
III. **Internal DB API** (for few DB Operations)
IV. **Telecom Operator Web API** (simulate our platform to send SMS to Telecom Operator)
V. **Back End Thread App** (Process the SMS requests received from Customers)

High Level Design:



(I) Customer API

This is the simulation of the Customer API module of our SMS Platform, which can help our platform to receive SMS messages from our customers.

http://localhost:<port>/telco/sendmsg?username={0}&password={1}&mobile={2}&message={3}

Request Parameters:

a) username: String
b) password: String

c) mobile: integer (10 digit)
d) message: String (max length should not exceed 160 characters)

Response:

Success Response: STATUS: ACCEPTED~~RESPONSE_CODE: SUCCESS~~Unique Acknowledgement ID

Unique Acknowledgement ID: It can be UUID.randomUUID() or System.currentTimeMillis() value

Failure Response: STATUS: REJECTED~~ RESPONSE_CODE: FAILURE (when the "mobile" is empty or less than 10 characters length or "message" parameter is empty, or username/password validation failed)

Validate the username & password received using the Internal DB API's 2nd API "GET ACCOUNT ID API" and get the account_id. After the successful validation, form a JSON object with those parameters and produce it to the Kafka Topic.

JSON Object Sample:

{ "ack_id": "12345ABC", "account_id": 1001, "mobile": 9876543210, "message": "Test Message" }

Explanation:

- ack_id: A string representing the acknowledgment ID.
- account_id: An integer representing the account ID.
- mobile: An integer representing the mobile number.
- message: A string representing the message.

(II) Kafka Consumer

This consumer will pick the object from Kafka topic, parse it and using the Internal DB API's first API "INSERT SEND MSG TABLE API" to insert this object information into send_msg table.

(III) Internal DB API

This is the simulation of the Internal DB API module, which can help other module developers to call this API to do few DB operations in the database.

1. **INSERT SEND MSG TABLE API:** API to send the SMS message (insert into send_msg table with given parameters and received_ts as current timestamp)

2. **GET ACCOUNT ID API:** API to get the account_id and validate the username and password matching with database entries in "users" table (if the validation is success, then return account_id)

(IV) Telecom Operator Web API

This is the simulation of Telecom Operators HTTP API which will accept the message from Back End (Thread based application) Module.

Following is the sample HTTP API URL,

http://localhost:<port>/telco/smsc?account_id={0}&mobile={1}&message={2}

Request Parameters:

a) account_id: integer
b) mobile: integer (10 digit)
c) message: String (max length should not exceed 160 characters)

Response:

Success Response: STATUS: ACCEPTED~~RESPONSE_CODE: SUCCESS~~Unique Acknowledgement ID

Unique Acknowledgement ID: It can be UUID.randomUUID() or System.currentTimeMillis() value

Failure Response: STATUS: REJECTED~~ RESPONSE_CODE: FAILURE (when the "mobile" or "accountID" or "message" parameters are empty or "mobile" parameter length less than or greater than 10 characters length)

(V) Back End Thread App

- This is a thread-based listener or scheduler component, which will keep on listening the send_msg table for every 1 second.
- Whenever, there is a new insert happening (status='NEW'), it will pick the record and update that records status="INPROCESS".
- Then call the Telecom Operator Web API with the necessary parameters and get the response
- Update the telco_response column with that response and status='SENT' and sent_ts='current time stamp')

Database Table Structure

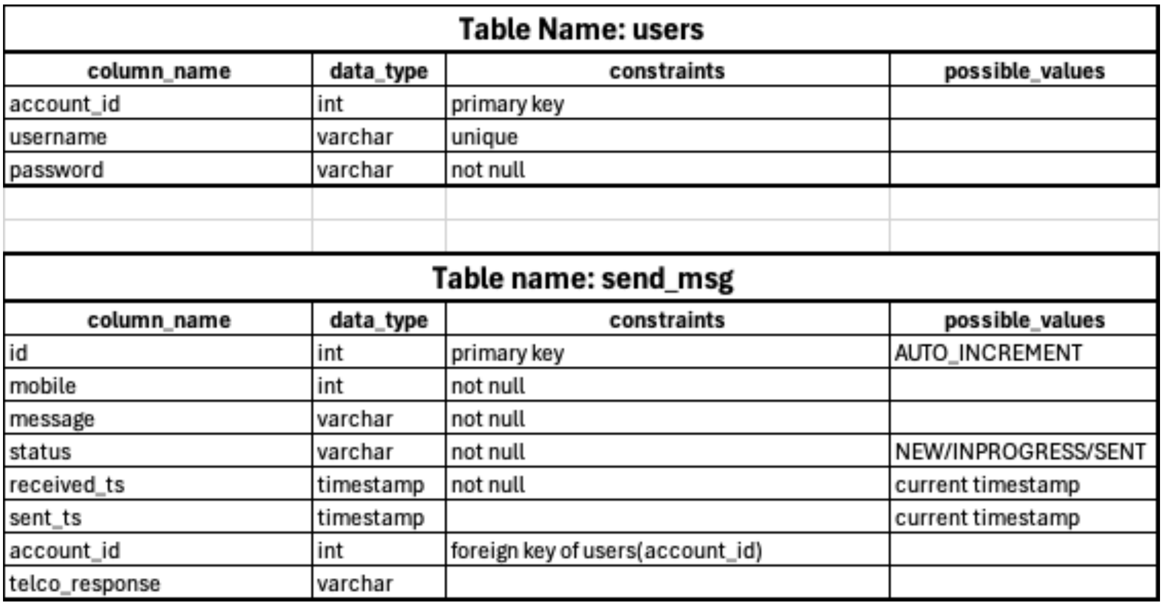


Table Name: users

column_name	data_type	constraints	possible_values
account_id	int	primary key	
username	varchar	unique	
password	varchar	not null	

Table name: send_msg

column_name	data_type	constraints	possible_values
id	int	primary key	AUTO_INCREMENT
mobile	int	not null	
message	varchar	not null	
status	varchar	not null	NEW/INPROGRESS/SENT
received_ts	timestamp	not null	current timestamp
sent_ts	timestamp		current timestamp
account_id	int	foreign key of users(account_id)	
telco_response	varchar		

Reference Notes:

It’s only for example and reference purpose, not actual.

For “users” table, you need to insert few data manually for testing and demonstration purpose. Following are sample data,

Insert into users (account_id, username, password) values
(‘1001’,’user1’,’password1’);
Insert into users (account_id, username, password) values
(‘1002’,’user2’,’password2’);

Few sample Queries for send_msg table:

Insert into send_msg(mobile, message, status, received_ts, account_id) values (?,?,’NEW’,now(),?)

Update send_msg set status=’INPROCESS’ where id=?

Update send_msg set status=’SENT’,telco_response=? Where id=?