**Sign up using Phone Number**

In today’s world, a user interacts with various apps/websites which require them to login or sign up via using their email address. There are a high chances of a person forgetting the username and the associated password to login or just completely giving up the sign up process as it asks a lot of information. Furthermore, from the app developer point of view, they will need to perform extra steps to verify the user’s email address.

This leads to another solution of using phone number as a way to login to the service. It is easy to remember the phone number and we can implement a mechanism to verify that the user is in possession of this number via text messages. In order to sign up with phone number the user Birth Date, Name, Gender ,Email this all need to be collected first.

* Birth Date: The user Date Of Birth must be collected that should be in dd/mm/yyyy format.
* Name: The complete name of the user is collected.
* Family Name: The complete Family details is necessary and that to be stored.
* Gender: The user gender to be collected as male, Female.
* Email: The user’s associated email address to be collected.

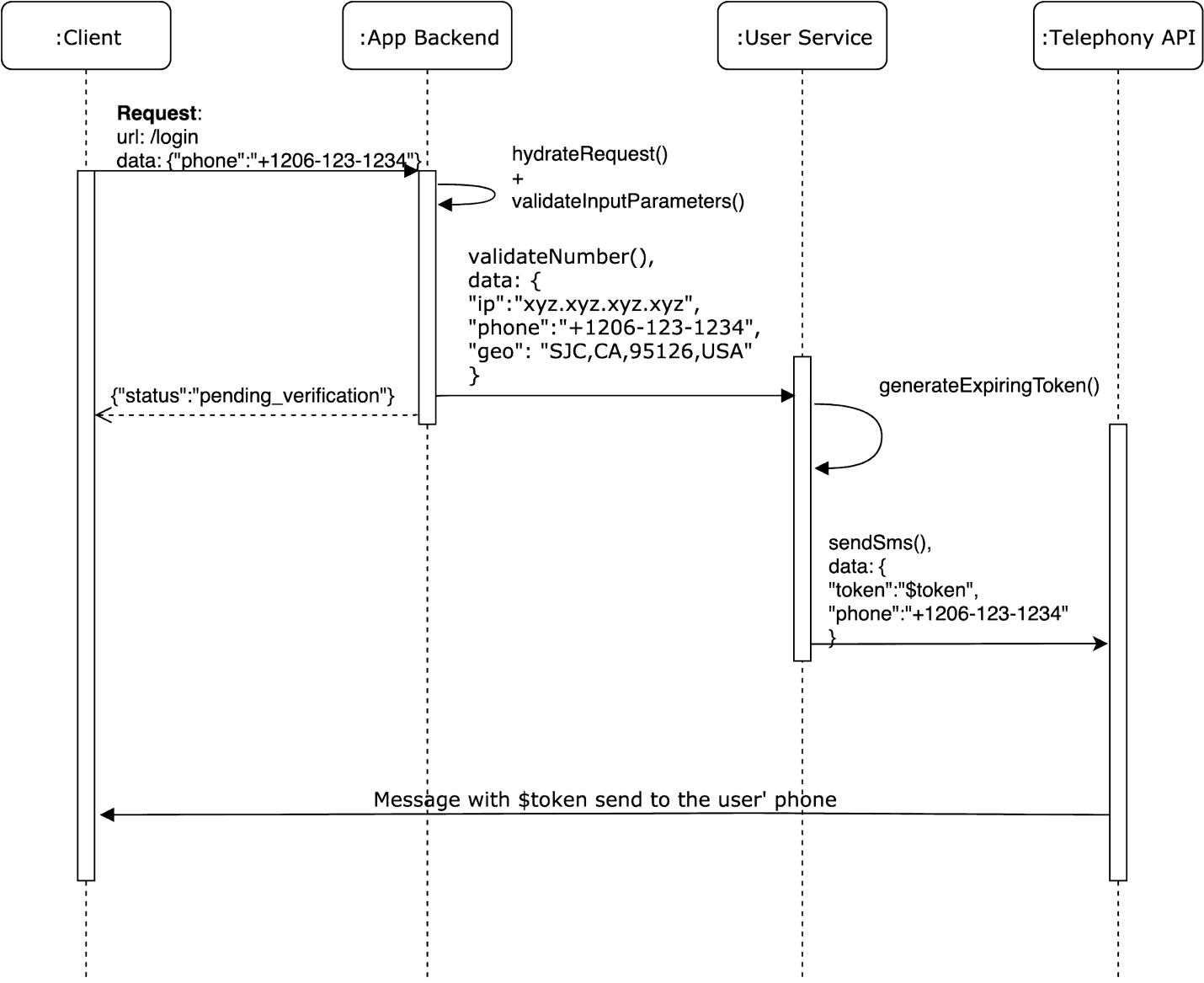
**sign up process is divided into two parts:**

1. Login Flow : In this phase, user submits the phone number to app backend and receives a token.
2. Verify Flow : In this phase, user submits the received token in the previous phase and sends to backend. On validation, user is logged in and a session cookie is issued.
3. **Login Flow:** In order to authenticate the user account, user submits the phone number to the App Backend . On submission, user is shown a screen where a token can be submitted. App backend validates the input parameter and enrichies/ hydrates the request with extra information like (IP address, geo information, device information etc). All this information can be used to generate a fingerprint of a user request which can be used for later security measures.This enriched and validated request is submitted to user service which generates a token and associate with these request. Token generation is an important process here.

**Token Generation**

The aim of the token is to make sure we can

* Expire the token so that same token cannot be used multiple times.
* Invalidate a token so that if a user enters the phone number multiple times then we can invalidate the previously generated token issue a new one.
* as the key and a randomly generated 6 digit token as the value which expires in 5 minutes.{“Key” : “+1206-123-1234”, “Value”:”123456”}



**Send SMS:** User service then calls a telephony API to send this token to the user’s phone number. There are various companies which provides this service (ex: Twilio, Plivo etc).

1. **Verify Phone Number Flow**

user enters the token on the form. This form consists of a hidden phone number field which is submitted to an endpoint (for ex: /verify\_phone) with the the token. Once again the app backend hydrates the request and validates the input parameters. The user service does a look up with the phone number as the key from the memcache data store. A session cookie is issued If the token exists and matches with the input token. This session cookie is sent as response header to the client which can persist it for later requests.

* **Sign up vs Login:** On successful token match, user service should do a look up in the user database to see if there is an already existing account with the same phone number, if that phone number is present then the user should be logged into that account. Another issue is to consider what happens if a user changes the phone number. This involves implementing a similar mechanism of verifying the new phone number and updating the user account if that is successful.
* **Advantage:** This definitely removes the mental load on the user to remember the email and passwords across various services. This is the best way.