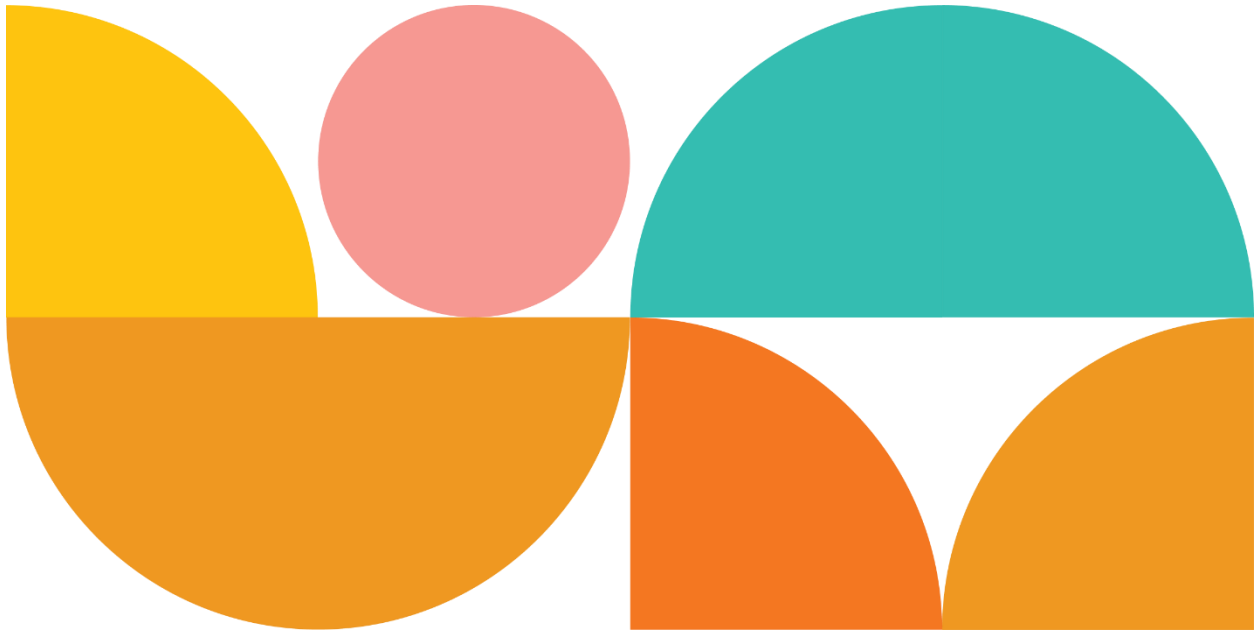


11/05/2023

SWASTHYA SURAKSHA

HEALTHCARE APPLICATION DEVELOPMENT

GROUP 22



IMT2019015	IMT2019030	IMT2019064	IMT2019084	IMT2019509
Ayush Tiwari	G Sri Harsha	Pratik R Ahirrao	Shrey Tripathi	Lalith Kumar Reddy G

Table of Contents

Swasthya Suraksha	4
The Gateway	4
Registration and Authentication.....	4
Consent Request from HIU.....	4
Data Request with Prior Consent	5
Send Data Request.....	5
Fetch Consents	5
Approve, Reject and Revoke Consent.....	5
The Structure	6
Flow Diagrams.....	7
HIU Request	7
HIU Request with Consent.....	8
HIP Send Request	9
Fetch Consents.....	9
Approve Consent.....	10
Revoke Consent.....	11
Reject Consent	12
List of All APIs.....	13
Gateway APIs.....	13
Request Consent for Data	13
Request for Data with Prior Consent.....	13
Request HIP to Send Data to HIU	14
Approve Consent Request.....	15
Verify Consent Request	16
Fetch Consents	17
Revoke Consent	17
Reject Consent.....	18
Send OTP	19

Verify OTP	19
Authenticate	20
Consent Manager APIs	21
Approve Consent Request	21
Verify Consent Request	22
Fetch Consents	23
Fetch Consents with Consent ID	24
Add Pending Consent	24
Revoke Consent	25
Reject Consent	25
Verify Pin	26
Hospital APIs	27
New Data Consent Request	27
Data Request with Prior Consent	27
HIP Send Data Request	28
Consent Update	29
Delete Consent	29
Patient APIs	31
Approve Consent Request	31
Fetch Consents	31
Revoke Consent	32
Reject Consent	33
Register	33
Verify OTP	34
Authenticate (Login)	34
Contribution	35
Links	35

Swasthya Suraksha

The Swasthya Suraksha framework aims to provide a trusted channel to connect patients, doctors, hospitals, and the consent manager. It connects the establishments to share the health data, given the patient's consent. Each entity is assigned a unique ID called the Swasthya Suraksha ID(SSID). The Swasthya Suraksha framework is primarily composed of the gateway, the database, the Patient App, the Hospital App and Consent Manager. The gateway is the sole communication medium for all the patient and hospital applications.

The database maintains the user and authentication details.

A patient/establishment registers in our app and gets a Swasthya Suraksha ID(SSID) and gets connected to our central gateway. In Patient App, patients can see the list of consents and then have a choice to approve/reject/revoke the consent initiated by doctor/establishment. In Hospital App, doctors can request a health record, fetch all consents, and create an EHR (Electronic Health Record).

The Gateway

The gateway is a complete backend service that all parties are required to communicate with. The gateway can register and authenticate patients, accept new consent request from HIU (Health Information User), accept data request with prior consent from HIU, place data request to HIP (Health Information Provider), handle accept, revoke, or reject consent request from patient, fetch consents given the SSID.

Registration and Authentication

The patients can register via the patient application. They are required to fill necessary demographic data along with their password and authentication pin. The user will receive an OTP on the registered mobile number. On entering the OTP, the gateway verifies the OTP and generates a unique 12-digit SSID.

The patients can login via the patient application. The patient enters the SSID and password for authentication. JWT (JSON Web Token) is used to maintain a user session.

Consent Request from HIU

The doctor can place a new consent request for patient data. The hospital application is required to provide the doctor SSID, HIU SSID, patient SSID, and HIP SSID. The gateway

validates the existence of all SSIDs. On validation, the gateway creates a new consent object with the shared fields and places a request to the consent manager to add a new pending consent.

Data Request with Prior Consent

If the hospital is already in possession of a consent object for the required data and the doctor wishes to fetch new data, they can place a request to the gateway to directly fetch data from the HIP without involving the patient.

The HIU provides the consent ID of the consent object associated with the data required. The gateway fetches the consent object from the CM (Consent Manager) and verifies it with the CM. On verification, the gateway places a request to the HIP to send the data.

Send Data Request

The gateway places the send data request to the HIP by sharing the consent object and the URL at which they can post the data to HIU.

Fetch Consents

The patient dashboard calls the gateway to fetch consents. The gateway requests the CM to share the consent objects given the patient SSID. This fetches all pending, approved and revoked consents.

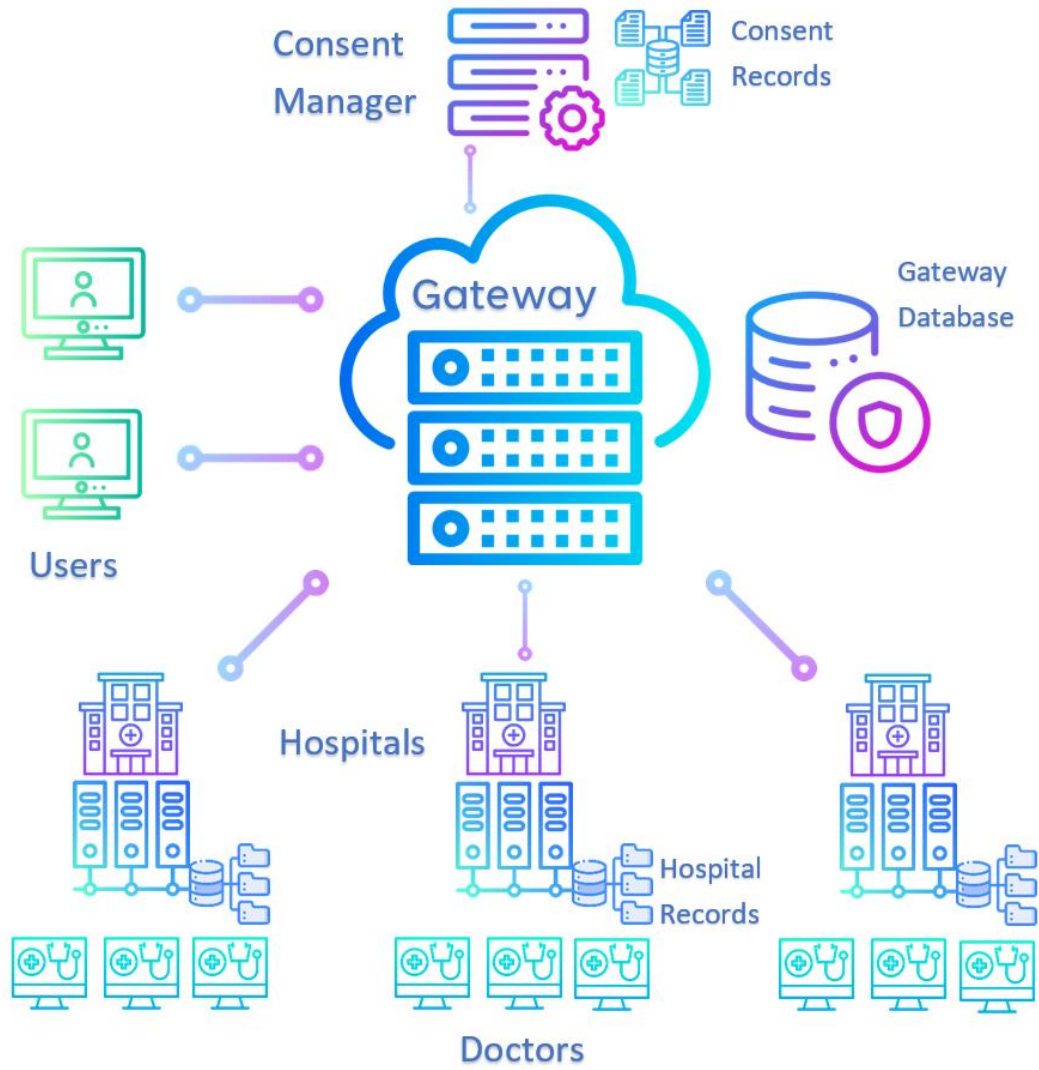
Approve, Reject and Revoke Consent

If a consent object is pending, i.e., the object is yet to be approved, the patient can approve it by filling the necessary fields and providing the authentication pin. The gateway shares this pending consent object with the filled fields and the authentication pin to the consent manager. The CM verifies the pin and saves the consent object, and is said to be approved. The gateway also shares this new consent object with the HIU and HIP.

If a consent object is approved, the patient can revoke the consent. The CM changes the status of the consent object to pending (sets approved to false). The gateway also updates the status of consent in HIU and HIP.

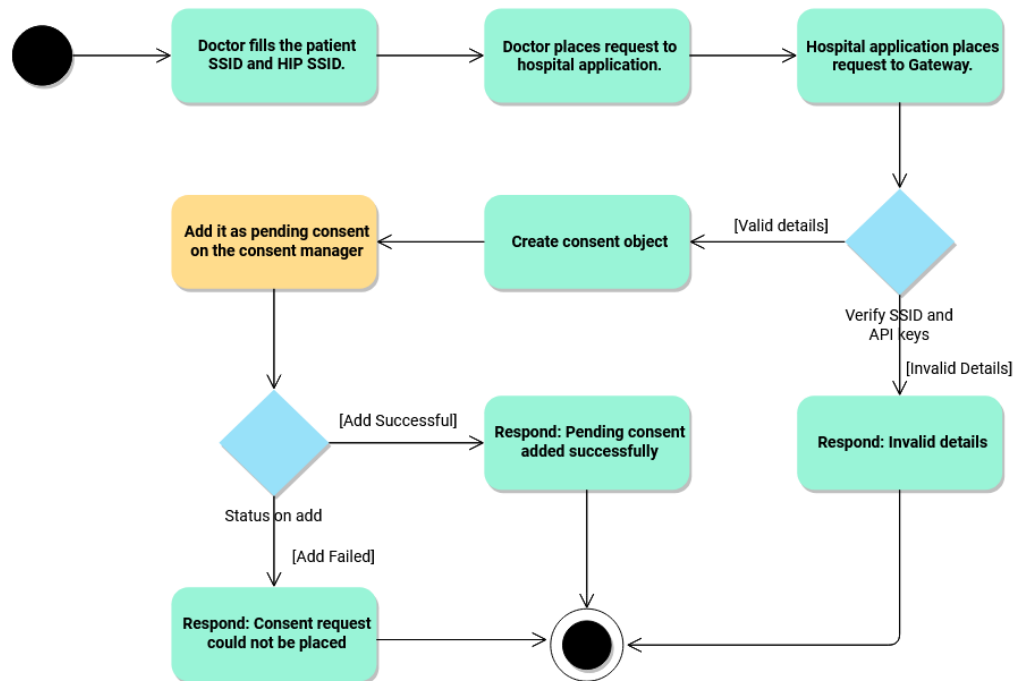
The patient can also reject the consent, which deletes the consent object.

The Structure

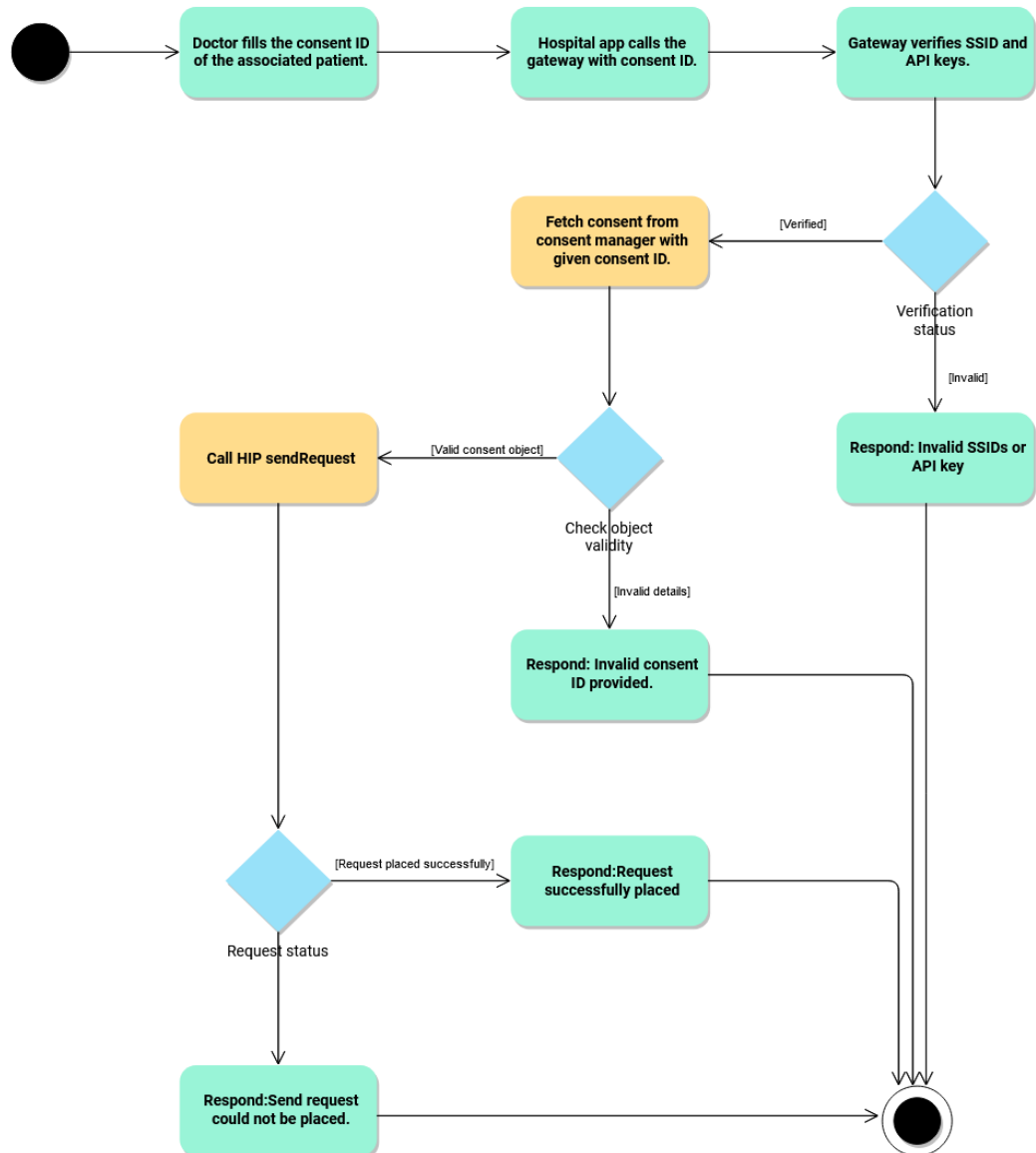


Flow Diagrams

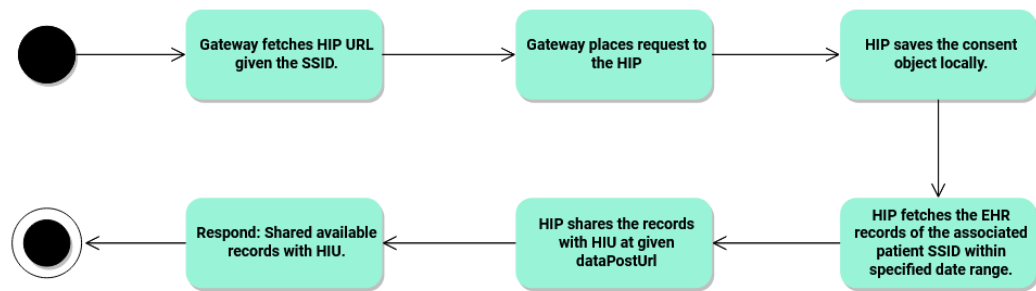
HIU Request



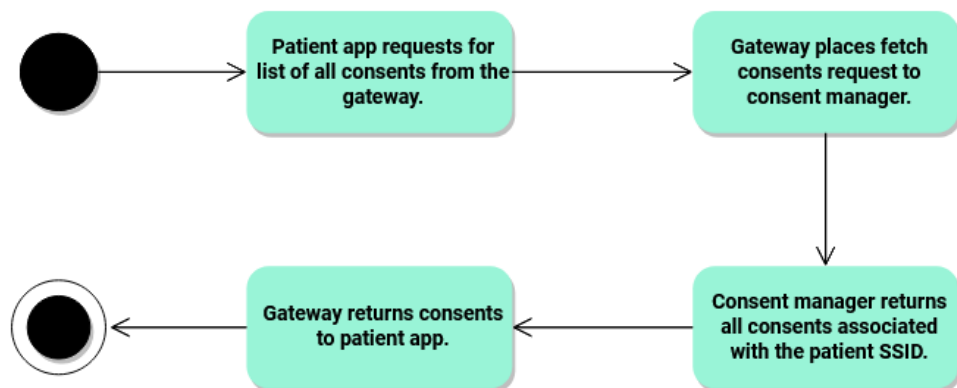
HIU Request with Consent



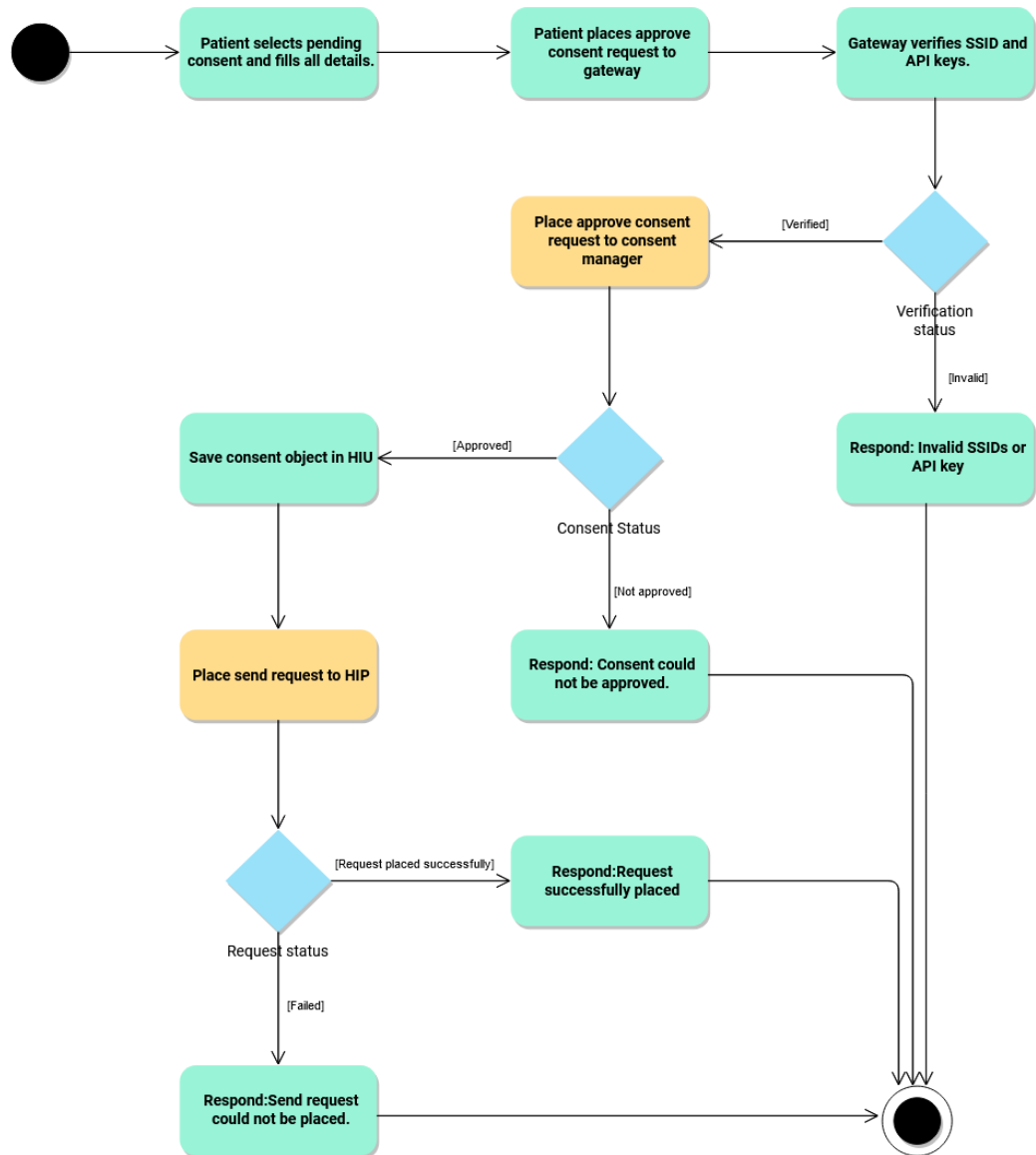
HIP Send Request



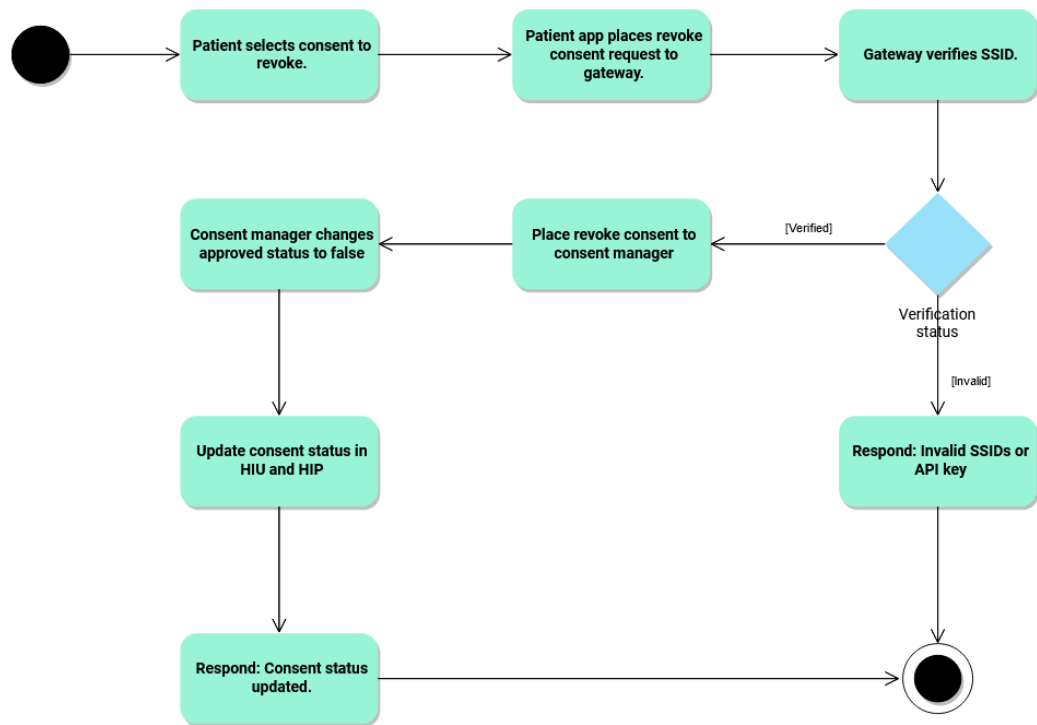
Fetch Consents



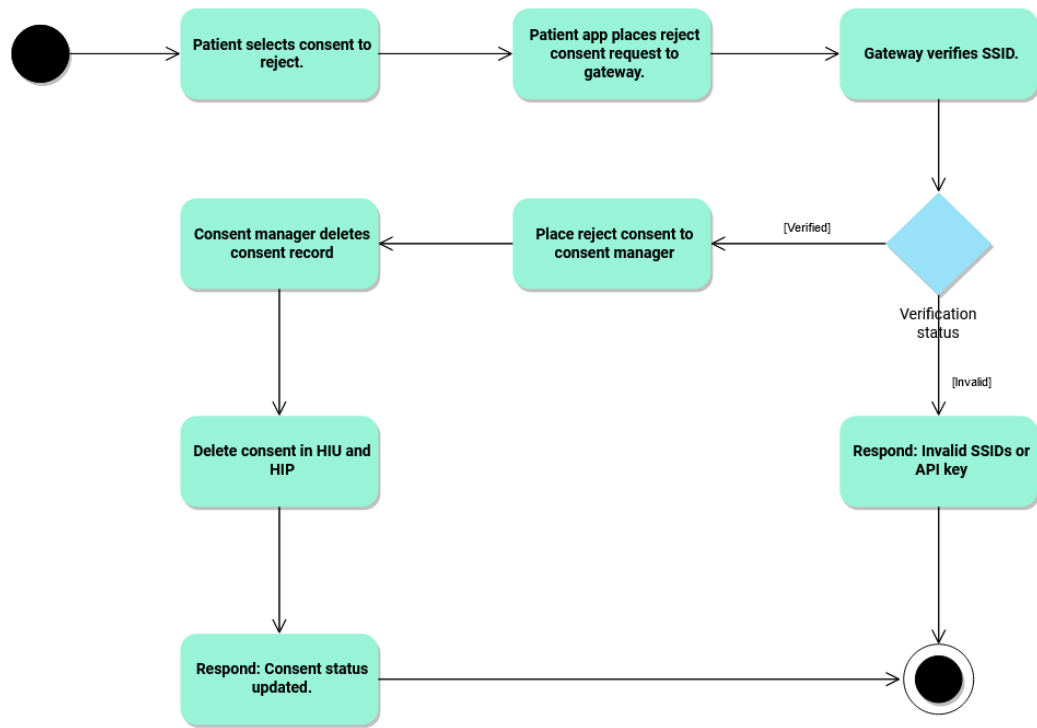
Approve Consent



Revoke Consent



Reject Consent



List of All APIs

Gateway APIs

Request Consent for Data

Path: /gateway/request/hiu/request

Call: HIU places the call to Gateway

Request Body:

```
doctorSSID: String  
hiuSSID: String  
patientSSID: String  
hipSSID: String
```

Request Response:

```
response: String  
doctorSSID: String  
hiuSSID: String
```

Function:

1. Validate API key and all SSIDs.
2. On validation, request to create a pending consent to the consent manager.
3. Respond to the HIU.

Request for Data with Prior Consent

Path: /gateway/request/hiu/requestWithConsent

Call: HIU places the call to Gateway

Request Body:

```
consentId: String
```

Request Response:

```
response: String  
doctorSSID: String  
hiuSSID: String
```

Function:

1. Validate API keys and SSID.
2. Verify consent object with the consent manager by fetching with given consent ID.
3. On verification, fetch HIU dataPostURL.
4. Then place the send request to HIP.
5. Respond to the HIU.

Request HIP to Send Data to HIU

Path: /gateway/request/hip/sendRequest

Call: Gateway places call to HIP

Request Body:

```
consentObj: {  
  doctorSSID: String  
  hiuSSID: String  
  patientSSID: String  
  hipSSID: String  
  dataAccessStartDate: LocalDate  
  dataAccessEndDate: LocalDate  
  dataAccessStartDate: LocalDate  
  requestInitiatedDate: LocalDate  
  consentApprovedDate: LocalDate  
  consentEndDate: LocalDate  
  consentID: String  
  selfConsent: boolean  
  isApproved: boolean  
}  
dataPostUrl: String
```

Request Response:

```
response: String
```

Function:

1. Fetch HIP address from SSID mapping.
2. Place request to HIP to send data to HIU.

Approve Consent Request

Path: /gateway/request/approveConsent

Call: Patient app places call to Gateway

Request Body:

```
patientSSID: String
encPin: String
consentObj: {
  doctorSSID: String
  hiuSSID: String
  patientSSID: String
  hipSSID: String
  dataAccessStartDate: LocalDate
  dataAccessEndDate: LocalDate
  dataAccessStartDate: LocalDate
  requestInitiatedDate: LocalDate
  consentApprovedDate: LocalDate
  consentEndDate: LocalDate
  consentID: String
  selfConsent: boolean
  isApproved: boolean
}
```

Request Response:

```
response: String
```

Function:

1. Verify all SSIDs.
2. Request consent manager to verify pin and approve consent.
3. On approval, send consent object to HIU.
4. On approval, place send request to HIP.
5. Respond with status.

Verify Consent Request

Path: /gateway/request/verifyConsent

Call: Any entity (other than CM) can place call to Gateway

Request Body:

```
reqSSID: String
consentObj: {
  doctorSSID: String
  hiuSSID: String
  patientSSID: String
  hipSSID: String
  dataAccessStartDate: LocalDate
  dataAccessEndDate: LocalDate
  dataAccessStartDate: LocalDate
  requestInitiatedDate: LocalDate
  consentApprovedDate: LocalDate
  consentEndDate: LocalDate
  consentID: String
  selfConsent: boolean
  isApproved: boolean
}
```

Request Response:

```
response: String
```

Function:

1. Verify all SSIDs.
2. Request consent manager to verify consent object.
3. Respond with status.

Fetch Consents

Path: /gateway/request/fetchConsents

Call: Patient can place call to Gateway

Request Body:

```
reqSSID: String
```

Request Response:

```
response: String
patientSSID: String
consentObjs: [
  consentObj: {}
  ...
]
```

Function:

1. Authenticate requesting user.
2. Place fetch consents request to consent manager.
3. Respond with status.

Revoke Consent

Path: /gateway/request/revokeConsent

Call: Patient can place call to Gateway

Request Body:

```
reqSSID: String
consentId: String
```

Request Response:

```
response: String
```

Function:

1. Verify all SSIDs.
2. Place revoke consent request to consent manager.
3. On revoke, update to HIU and HIP.
4. Respond with status.

Reject Consent

Path: /gateway/request/rejectConsent

Call: Patient can place call to Gateway

Request Body:

```
reqSSID: String  
consentId: String
```

Request Response:

```
response: String
```

Function:

1. Verify all SSIDs.
2. Place reject consent request to consent manager.
3. On rejection, update to HIU and HIP.
4. Respond with status.

Send OTP

Path: /api/v1/auth/send-otp

Call: Patient app calls the gateway

Request Body:

```
phoneNum: String
```

Request Response:

```
void
```

Function:

1. Send OTP on phone number provided for registration.

Verify OTP

Path: /api/v1/auth/verify-otp

Call: Patient app calls the gateway

Request Body:

```
phoneNum: String  
otp: String
```

Request Response:

```
response: AuthResponse
```

Function:

1. Verify OTP sent for registration.
2. If valid, generate SSID.
3. Login patient by sharing JWT token.

Authenticate

Path: /api/v1/auth/authenticate

Call: Patient app calls the gateway

Request Body:

```
SSID: String  
password: String
```

Request Response:

```
response: AuthReponse
```

Function:

1. Verify SSID and password for authentication.
2. If valid, login patient by sharing JWT token.

Consent Manager APIs

Approve Consent Request

Path: /cm/consents/approveConsent

Call: Gateway places call to Consent Manager

Request Body:

```
patientSSID: String
encPin: String
consentObj: {
  doctorSSID: String
  hiuSSID: String
  patientSSID: String
  hipSSID: String
  dataAccessStartDate: LocalDate
  dataAccessEndDate: LocalDate
  dataAccessStartDate: LocalDate
  requestInitiatedDate: LocalDate
  consentApprovedDate: LocalDate
  consentEndDate: LocalDate
  consentID: String
  selfConsent: boolean
  isApproved: boolean
}
```

Request Response:

```
response: String
consentObj: {
  doctorSSID: String
  hiuSSID: String
  patientSSID: String
  hipSSID: String
  dataAccessStartDate: LocalDate
  dataAccessEndDate: LocalDate
  dataAccessStartDate: LocalDate
  requestInitiatedDate: LocalDate
  consentApprovedDate: LocalDate
  consentEndDate: LocalDate
  consentID: String
  selfConsent: boolean
  isApproved: boolean
}
```

Function:

1. Verify pin.
2. On verification, save new consent object.
3. Return approved consent object (or rejected status).

Verify Consent Request

Path: /cm/consents/verifyConsent

Call: Gateway places call to consent manager

Request Body:

```
reqSSID: String
consentObj: {
  doctorSSID: String
  hiuSSID: String
  patientSSID: String
  hipSSID: String
  dataAccessStartDate: LocalDate
  dataAccessEndDate: LocalDate
  dataAccessStartDate: LocalDate
  requestInitiatedDate: LocalDate
  consentApprovedDate: LocalDate
  consentEndDate: LocalDate
  consentID: String
  selfConsent: boolean
  isApproved: boolean
}
```

Request Response:

```
response: String
reqSSID: String
consentObj: {
  doctorSSID: String
  hiuSSID: String
  patientSSID: String
  hipSSID: String
  dataAccessStartDate: LocalDate
  dataAccessEndDate: LocalDate
  dataAccessStartDate: LocalDate
  requestInitiatedDate: LocalDate
  consentApprovedDate: LocalDate
  consentEndDate: LocalDate
  consentID: String
  selfConsent: boolean
  isApproved: boolean
}
```

Function:

1. Verify consent object exists and match all fields.
2. On verification, respond with status.

Fetch Consents

Path: /cm/consents/fetchConsents

Call: Gateway places call to consent manager

Request Body:

```
reqSSID: String
```

Request Response:

```
patientSSID: String
consentObjs: [
  consentObj: {}
  ...
]
```

Function:

1. Look for consent records matching the patient SSID.
2. Return list of matching SSIDs.

Fetch Consents with Consent ID

Path: /cm/patient/auth/verifyPin

Call: Gateway places call to consent manager

Request Body:

```
consentId: String
```

Request Response:

```
consentObj: {}
```

Function:

1. Search for consent object given consent ID.

Add Pending Consent

Path: /cm/consents/addPendingConsent

Call: Gateway places call to consent manager

Request Body:

```
consentObj: {}
```

Request Response:

```
consentObj: {}
```

Function:

1. Add pending consent to consent manager records.

Revoke Consent

Path: /cm/consents/revokeConsent

Call: Gateway places request to consent manager

Request Body:

```
reqSSID: String  
consentId: String
```

Request Response:

```
consentObj: {}
```

Function:

1. Change consent status and save object.
2. Respond with new consent object.

Reject Consent

Path: /cm/consents/rejectConsent

Call: Gateway places call to consent manager

Request Body:

```
reqSSID: String  
consentId: String
```

Request Response:

```
response: boolean
```

Function:

1. Delete existing consent object from consent manager.

Verify Pin

Path: /cm/patient/auth/verifyPin

Call: Consent manager places an internal private call

Request Body:

```
SSID: String  
encPin: String
```

Request Response:

```
response: boolean
```

Function:

1. Verify auth pin for corresponding patient SSID.

Hospital APIs

New Data Consent Request

Path: /hospital/requests/newRequest

Call: The doctor app makes the call to the hospital backend

Request Body:

```
docSSID: String
patientSSID: String
hipSSID: String
```

Request Response:

```
response: String
docSSID: String
hiuSSID: String
```

Function:

1. Place consent request for new data to gateway.
2. Respond with new consent request response from gateway.

Data Request with Prior Consent

Path: /hospital/requests/requestWithConsent

Call: The doctor app makes the call to the hospital backend.

Request Body:

```
docSSID: String
patientSSID: String
consentID: String
```

Request Response:

```
response: String
docSSID: String
hiuSSID: String
```

Function:

1. Place data request to gateway with prior consent.
2. Respond with new consent request response from gateway.

HIP Send Data Request

Path: /hospital/requests/hip/sendRequest

Call: The gateway makes the call to HIP

Request Body:

```
dataPostUrl: String
consentObj: {}
```

Request Response:

```
response: String
```

Function:

1. Save consent object.
2. Fetch EHR for the associated patient SSID and within specified date range.
3. Send all EHR to the HIU via the dataPostUrl.

Consent Update

Path: /hospital/requests/consentUpdate

Call: The gateway makes the call to Hospital

Request Body:

```
consentObj: {}
```

Request Response:

```
response: boolean
```

Function:

1. Update with new consent object.
2. Respond on update status.

Delete Consent

Path: /hospital/requests/deleteConsent

Call: The gateway makes the call

Request Body:

```
consentObj: {}
```

Request Response:

```
response: boolean
```

Function:

1. Save consent object.
2. Fetch EHR for the associated patient SSID and within specified date range.
3. Send all EHR to the HIU via the dataPostUrl.

Patient APIs

Approve Consent Request

Path: /patient/approveConsent

Call: Patient places request to patient backend app

Request Body:

```
patientSSID: String
encPin: String
consentObj: {
  doctorSSID: String
  hiuSSID: String
  patientSSID: String
  hipSSID: String
  dataAccessStartDate: LocalDate
  dataAccessEndDate: LocalDate
  dataAccessStartDate: LocalDate
  requestInitiatedDate: LocalDate
  consentApprovedDate: LocalDate
  consentEndDate: LocalDate
  consentID: String
  selfConsent: boolean
  isApproved: boolean
}
```

Request Response:

```
response: String
```

Function:

1. Patient fills form and clicks on approve.
2. Places call to patient app backend, which calls the gateway.

Fetch Consents

Path: /patient/fetchConsents

Call: Patient places call to patient backend app

Request Body:

```
reqSSID: String
```

Request Response:

```
patientSSID: String
consentObjs: [
  consentObj: {}
  ...
]
```

Function:

1. Place fetch consents record to gateway.

Revoke Consent

Path: /patient/revokeConsent

Call: Patient calls the patient backend app

Request Body:

```
reqSSID: String
consentId: String
```

Request Response:

```
consentObj: {}
```

Function:

1. Place revoke consent request to gateway.

Reject Consent

Path: /patient/rejectConsent

Call: Patient calls the patient backend app

Request Body:

```
reqSSID: String  
consentId: String
```

Request Response:

```
response: boolean
```

Function:

1. Place reject consent request to gateway.

Register

Path: /patient/register

Call: Patient calls the patient backend app

Request Body:

```
phoneNum: String
```

Request Response:

```
void
```

Function:

1. Place register request to gateway.

Verify OTP

Path: /patient/verify-otp

Call: Patient calls the patient backend app

Request Body:

```
phoneNum: String  
otp: String
```

Request Response:

```
response: AuthResponse
```

Function:

1. Place verify OTP request to gateway.

Authenticate (Login)

Path: /patient/authenticate

Call: Patient calls the patient backend app

Request Body:

```
SSID: String  
password: String
```

Request Response:

```
response: AuthResponse
```

Function:

1. Place authenticate request to gateway.

Contribution

- Ayush Tiwari: JWT Authentication, API Keys, services in front end and backend
- G Sri Harsha: Registration and authentication with Twilio, Internationalization, Notifications
- Pratik Ratnadeep Ahirrao: Major Front end UI/UX, integrating front end and backend
- Shrey Tripathi: Minor Front end UI/UX, integrating front end and backend, services in frontend and backend
- Lalith Kumar Reddy G: Backend structure setup, major services in backend, documentation

Every member contributed in almost every aspect of the project.

Links

Github Repos

- Frontend: <https://github.com/Pratik-ahirrao/swaksha-frontend>
- Backend: <https://github.com/lalith-krg/swaksha-backend>

OneDrive Link

- Video: https://iiitborg-my.sharepoint.com/:f/g/personal/shrey_tripathi_iiitb_org/Etj4usVtwWFIpTOY4JBOWjwBPpgzJ1c9f9MtSSvlziYF2w?e=no6KEm