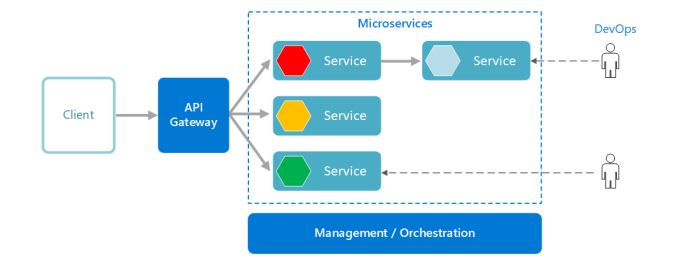
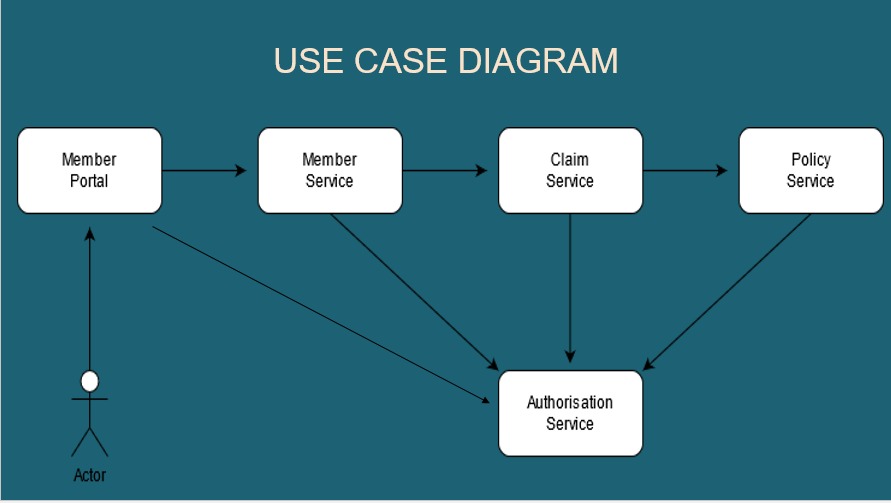
**High Level Design Document**

# Schematic Diagram

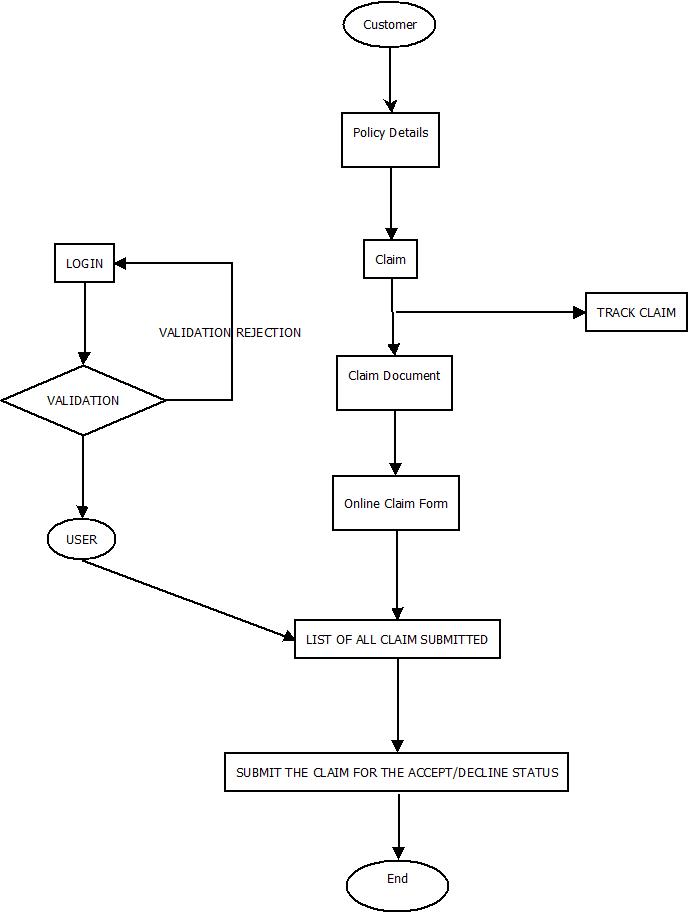
# 3.1 System Architecture Diagram



# 3.2 Use Case Diagram

****

# 3.3 Flow Chart

****

# Functional Requirements and High Level Design

## Individual Components of the System

### Member Microservice

|  |  |
| --- | --- |
| **Claims Management System** | **Member Microservice** |
| **Functional Requirements**  Can assume that Member Portal App is the only client to this Microservice. An authorized member view the premium bills, submit the claim, can view the claim status, which is already submitted.  Post Authorization, basic Member based validation are performed in this Microservice and then it communicates to the below Microservices for retrieving necessary information.  Member Microservice will interact with the Claim Microservice for the following functionalities:   * The Microservice will interact with Claims Module, to check the eligibility of the member and the claim, and then action the claim to set for processing. * To get the status of an already submitted claim | |
| **Entities**   1. **Member**   <Details of Member>   1. **Member Policy**   <Details of Policy subscribed by every member, Premium Details, Top-up Summary>   1. **Member Claim**   <Details of Member, Policy, Claim Status, Claim requested and settled details>   1. **Member Premium**   <Details of Primary Member, Policy Details, Premium Due and Payment Details>  **REST End Points**  **Claim Microservice**   * GET: /view Bills (Input: Member\_ID, Palicid | Output: Last Premium Paid Date, Premium\_Amount\_Due, Details of Late Payment Charges if applicable, Due Date etc.) * GET: /getClaimStatus (Input: Claim\_ID, Policy\_ID, Member\_ID | Output: Claim Status, Claim Status Description) * POST: /submitClaim (Input: Policy\_ID, Member\_ID, Claim\_Details (Hospital ID, Benefits Availed, Total Billed Amount, Total Claimed Amount) | Output: Claim Status, Claim Status Description) | |
| **Trigger** – Can be invoked from Member Portal (local MVC app) | |
| **Steps and Actions**   1. Member Portal will request for any of the 4 operations as per the business logic 2. For all the 3 operations basic Member Profile will be verified before interacting with other Microservices. 3. At any point in time, the premium bills must be viewed by the Member Portal Client. Hence the Member Microservice must expose the Premium Details through /view Bills REST End Point. (Assume that the Premium Payment is not done in the portal, wherein it gets updated to the system as flat file). 4. If /getClaimStatus end point is invoked the Claims Microservice has to be invoked to get the claim status. The response returned by Claims has to be cascaded as the response in this end point. 5. If /submitClaim end point is invoked, then Claims Microservice has to be invoked with Claim Details. The new Claim submission status returned by Claims Microservice will be forwarded to the Member Portal by the Member Microservice | |
| **Non-Functional Requirement:**   * Only Authorized Member can access these REST End Points * If the view Bill request is received multiple times for the same member and policy, there must not be multiple database hits leading to performance issues. | |

### Claims Microservice

|  |  |
| --- | --- |
| **Claims Management System** | **Claims Microservice** |
| **Functional Requirements**  Member Microservice interacts with Claims Microservice. Post authorization of request, Claim Microservice allows the following operations:  To view the status of submitted claim:   * Retrieve the claim status from database and return   To verify claim eligibility by interacting with Policy Microservice and action settlement:   * View the Claim details and check the following:   i. If the Claimed Amount is applicable under the subscribed policy  ii. If the Claimed benefit is applicable under the subscribed policy  iii. If the Hospital in which benefits are availed is a permissible Health Care Provider (Hospital).  If the above 3 conditions are satisfied, update the claim as “Pending Action” else “Claim Rejected”. If any information is not available or found to be invalid, then update status as “Insufficient Claim Details”. If any contradictory details found, update status as “Under Dispute” | |
| **Entities**  **1. Claim**  <Claim details like Claim Number, Status, Remarks, Policy/Benefit Details, Hospital Details, Benefits Availed, Amount Claimed, Settled etc.>  **REST End Points**  **Claims Microservice**  • GET: /getClaimStatus (Input: Claim\_ID, Policy\_ID, Member\_ID | Output: Claim Status, Claim Status Description).  • POST: /submitClaim (Input: Policy\_ID, Member\_ID, Claim\_Details (Hospital ID, Benefits Availed, Total Billed Amount, Total Claimed Amount) | Output: Claim Status, Claim Status Description | |
| **Trigger** – Can be invoked from Member Microservice. | |
| **Steps and Actions:**  1. Claims Microservice will have 2 End Points exposed to Member Microservice  2. If /getClaimStatus end point is invoked by Member Microservice, the Claims Microservice will check the status in database and will return the response back to Member Microservice.  3. If /submitClaim end point is invoked, then the Claims Microservice will invoke the Policy Microservice for retrieving the permissible Provider Details (Hospital), eligible benefits of a policy and the eligible claim amoun for the benefits.   * Based on the details retrieved, the Claims Microservice will decide on any of the following   actions: sanctioning / rejecting / requesting further information / raising a dispute | |
| **Non-Functional Requirement:**  **•** Hitting the Policy Microservice for 3 different details must happen in parallel to improve efficiency | |

### Policy Microservice

|  |  |
| --- | --- |
| **Claims Management System** | **Policy Microservice** |
| **Functional Requirements**  Claims Microservice interacts with Policy Microservice. Post authorization of request, Claim Microservice allows the following operations:   * Provide the permissible providers in which healthcare services can be offered. (Return the * chain of hospitals) * Provide the list of benefits which the member is eligible to, under a subscribed policy. * To provide the eligible claim amount, for the given benefit under a subscribed policy. | |
| **Entities**   1. **Policy**   <Policy details like Policy Number, Benefits, Premium, Tenure etc.>   1. **Member\_Policy**   <Member Policy details like Policy Number, Member Details, Subscription Date, Tenure, Benefits, Cap Amount for benefits etc.>   1. **Provider\_Policy**   <Provider/Hospital details which is allowed to avail benefits from, for each policy in each location.>  **REST End Points**  **Policy Microservice**   * GET: /getChainOfProviders (Input: Policy\_ID | Output (Provider List, location wise) * GET: /getEligibleBenefits (Input: Policy\_ID, Member\_ID | Output: Benefits List for the Member, for the policy) * GET: /getEligibleClaimAmount (Input: Policy\_ID, Member\_ID, Benefit\_ID | Output: Eligible Amount that can be claimed) | |
| **Trigger** – Can be invoked from Claims Microservice | |
| **Steps and Actions**   1. Claims Microservice will have 3 End Points exposed to Claims Microservice 2. For all the 3 End Points, straight forward details will be retrieved from the database | |

### Authorization Microservice

|  |  |
| --- | --- |
| **Claims Management System** | **Authorization Microservice** |
| **Security Requirements**   * Service to Service communication has to happen using JWT * Pass End User Context across Microservice * Have the token expired after specific amount of time say 15 minutes. * Have this service configured in the cloud along with other services | |

### Swagger

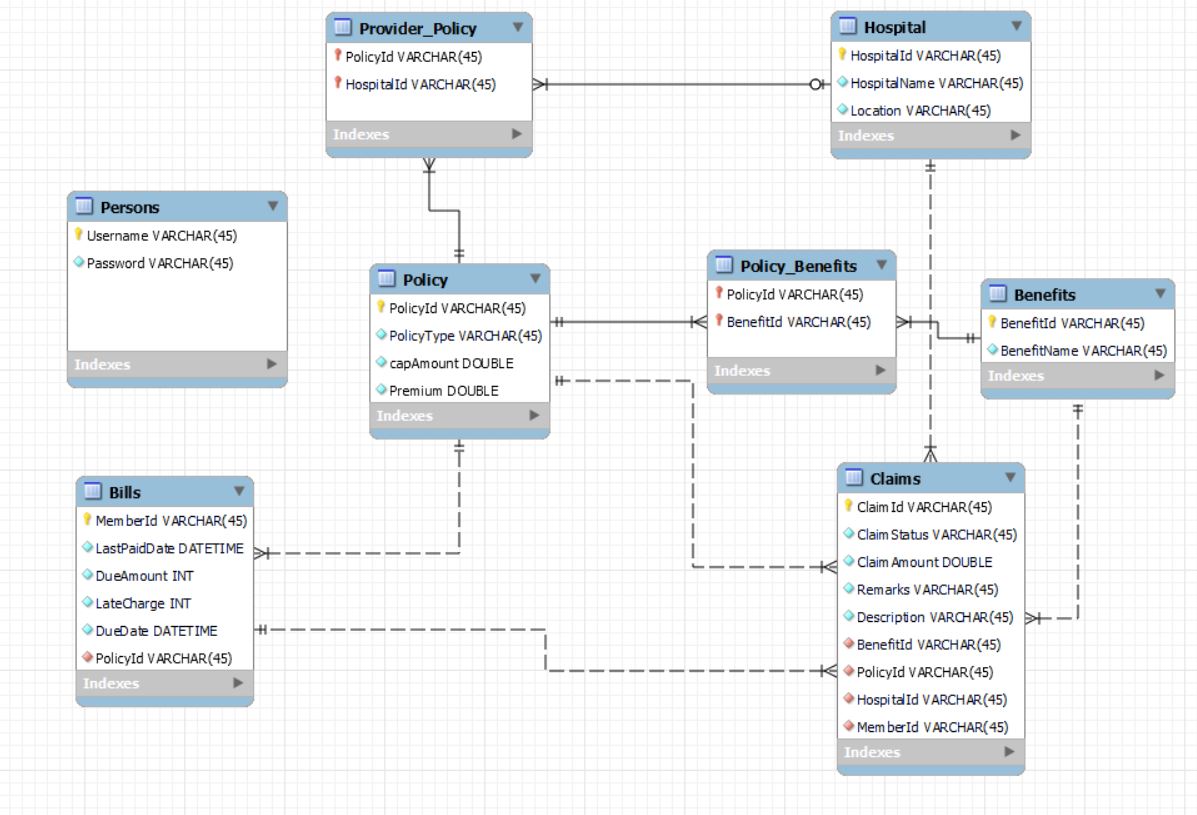
|  |  |
| --- | --- |
| **Claims Management System** | **Swagger** |
| **Documentation Requirements(Java)**   * All the Microservices must be configured with Swagger for documentation * Register the swagger resources in the Swagger Microservice and enable them as REST end points * Configure this service along with other services in the cloud | |

### Member Portal (MVC)

|  |  |
| --- | --- |
| **Claims Management System** | **Member Portal** |
| * Member Portal must allow a member to Login. Once successfully logged in, the member does the following operations: * View Claim Status * Submit a Claim * View Bill Status of the subscribed policies * Each of the above operations will reach out to the middleware Microservices that are hosted in cloud. | |

# Database Design

## Data Model



## Table Structure

* + 1. **Persons**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| Username | VARCHAR | 45 | No |
| Password | VARCHAR | 45 | No |

* + 1. **Bills**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| MemberId | VARCHAR | 45 | No |
| LastPaidDate | DATETIME |  | No |
| DueAmount | INT |  | No |
| LateCharge | INT |  | No |
| DueDate | DATETIME |  | Yes |
| PolicyId | VARCHAR | 45 | No |

* + 1. **Policy**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| PolicyId | VARCHAR | 45 | No |
| PolicyType | VARCHAR | 45 | Yes |
| capAmount | DOUBLE |  | Yes |
| Premium | DOUBLE |  | Yes |

* + 1. **Provider Policy**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| PolicyId | VARCHAR | 45 | No |
| HospitalId | VARCHAR | 45 | No |

* + 1. **Policy Benefits**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| PolicyId | VARCHAR | 45 | No |
| BenefitId | VARCHAR | 45 | No |

* + 1. **Hospital**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| HospitalId | VARCHAR | 45 | No |
| HospitalName | VARCHAR | 45 | No |
| Location | VARCHAR | 45 | No |

* + 1. **Claims**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| ClaimId | VARCHAR | 45 | No |
| ClaimStatus | VARCHAR | 45 | No |
| ClaimAmount | DOUBLE |  | Yes |
| Remarks | VARCHAR | 45 | Yes |
| Description | VARCHAR | 45 | Yes |
| BenefitId | VARCHAR | 45 | No |
| PolicyId | VARCHAR | 45 | No |
| HospitalId | VARCHAR | 45 | No |
| MemberId | VARCHAR | 45 | No |

* + 1. **Benefits**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| BenefitId | VARCHAR | 45 | No |
| BenefitName | VARCHAR | 45 | Yes |