# PROJECT BRIEF

An Insurance Management System that manages the customers, policies bought and claims made. The application is expected to have the following 3 services –

* Customer Management
  + Add New Customers
  + View Existing Customers
* Policy Management
  + Register a new Policy
  + Renew an Existing Policy
* Claim Management
  + Registering the Claim

## Functional Requirements

1. Add Customer
   1. Admin and Manager will be able to add customer
   2. Customer info along with the policy no should be added.
   3. Validity - Policy Start date and End date should also be added.
   4. Customer ID should be generated automatically.
2. View Customers
   1. Admin or manager can view the Customers.
   2. Customer data should be displayed in table along with status.
   3. Status should be updated based on the validity of the policy they registered to.
3. Register Policy
   1. All policy data will be added. Policy No will be auto generated.
   2. Policy ID will be generated automatically and will set to active.
4. Renew Policy
   1. Policy can be both Renewed and cancelled.
   2. On Renewal, the policy has to be extended to one year from the date of renewal.
5. Register Claims
   1. Customers can land to the page.
   2. Input customer id, policy id and bill amount to be claimed.

## Technical Requirements

* Backend – Java Spring Boot
* Database – MYSQL or H2
* ORM – Hibernate
* Language – Java
* API Testing – Postman or Chrome plug-in.

## Mandatory Requirements

1. All REST API should have proper responses, Exception Handling and Logging Mechanism.
2. Follow the best coding principles and Naming conventions as required.
3. Implement JWT token or OAuth2 or Basic authentication at API side
4. All the application related settings (like connection string, port,...) should come from application.properties or YML file.
5. Use ORM or JPQL for query and use Native Query as required.
6. Create Entity , Repository and controller as required.
7. Do Validations of fields.
8. Use Comments as required.

**Note** – Make sure all the best practices such as Normalization and SOLID principles are followed.