**BestBenefits**

**www.best-benefits.com**

**Architecture / Workflow / Design Document**

**B***estBenefits* software allows companies to setup *Life Insurance*, *Medical Insurance* and other benefit policies they can provide to their employees.

- For each of these policies the contribution provided by employee will be different by age i.e. if the age of the Employee is between 30-35 then the rate is $10 per month

* How would you go about building this flexible policy configuration?

- Workflow diagrams are given and explained in one line statement based on the System and Application level.

**Application/System Level** (Front-end / UI based workflow - AWS/Azure etc. platform):

best-benefits

(application)

notification-service

MySQL

Modules/APIs i.e.

Auth

Users

Roles

**Life-Insurance**

**Med-Insurance**

Redis(i.e. caching)

Notifications

**Module Level** (Various Modules):

Other policies etc.

if ($emp\_age is between 30 and 35)

$life\_insu = $10 per month

if ($emp\_age is between 30 and 35)

$life\_insu = $10 per month

**Application Level** use case diagram:

We could understand the high level workflow of the *BestBenefits* application from the above diagrams, in which Authentication, Users, Roles, Policies, User Mgnt, Life Insurance, Medical Insurance and other policies are the modules. And as per the system all are playing a key role to manage entire back-end business to manage; which has a layered pattern i.e. application, presentation, database and infrastructure. Application has been integrated with notification services to send or view the notifications.

Application consists of the following components

* **MySQL Database** – The mid-tier storage component
* **Redis Catching** – The database catching component
* **Load Balancing and Elastic Search** - Cloud services etc.
* **Back-end Modules** – The main business application which provides all the business features
* **Front-end Application** – The frontend components to render the web application
* **API Gateway** – The API requests for the frontend application rendering being served through this gateway
* **Notification Services** - i.e. Twilio, SendGrid etc. are integrated for notification purpose

**Database Level**:

Assume that BestBenefit application has 'bestbenefit' database and hosted on AWS. Based on the user story we can come up with below specimen tables to satisfy the business rules or requirements.

- user

id p.k.

name

address

created\_at

updated\_at

- permissions

id p.k.

all

few

created\_at

updated\_at

- roles

id p.k.

admin

user

created\_at

updated\_at

- benefit\_policy i.e. Medical, Dental, Life etc.

id p.k.

benefit\_name

...

- benefit\_plan i.e. various plans based on category

id p.k.

plan\_name

...

- benefit\_plan\_option i.e. employee only, employee with family etc.

id p.k.

option\_name

...

As per the user story; these policies contribution provided by employee will be different by age i.e. if the age of the Employee is between 30-35 then the rate is $10 per month.

**Specimen** / **Pseudo code** for user story:

interface PolicyInterface () {

public ApplyPolicy() {}

...

}

abstract class PolicyPlan() {

public ApplyPlan() {}

...

}

class PolicyController extends Controller {

public $insurance\_policy;

public $insurance\_plan;

public $employee\_age;

public $employee\_policy;

public function \_\_construct() {

// --

}

public function ApplyPolicy() {

if ($this->employee\_age > 30 && $this->employee\_age < 35) {

$this->employee\_policy = $10;

$this->insurance\_policy = $10;

return "Policy Applied.";

}

}

}

**System & Application Configuration**: have to be configured at application level and which will be associated with the system i.e. AWS or Azure platform. Configuration contains all the resources to be used into the application based on conditions for various purposes.

-- end of the document --