INSURANCE PROJECT

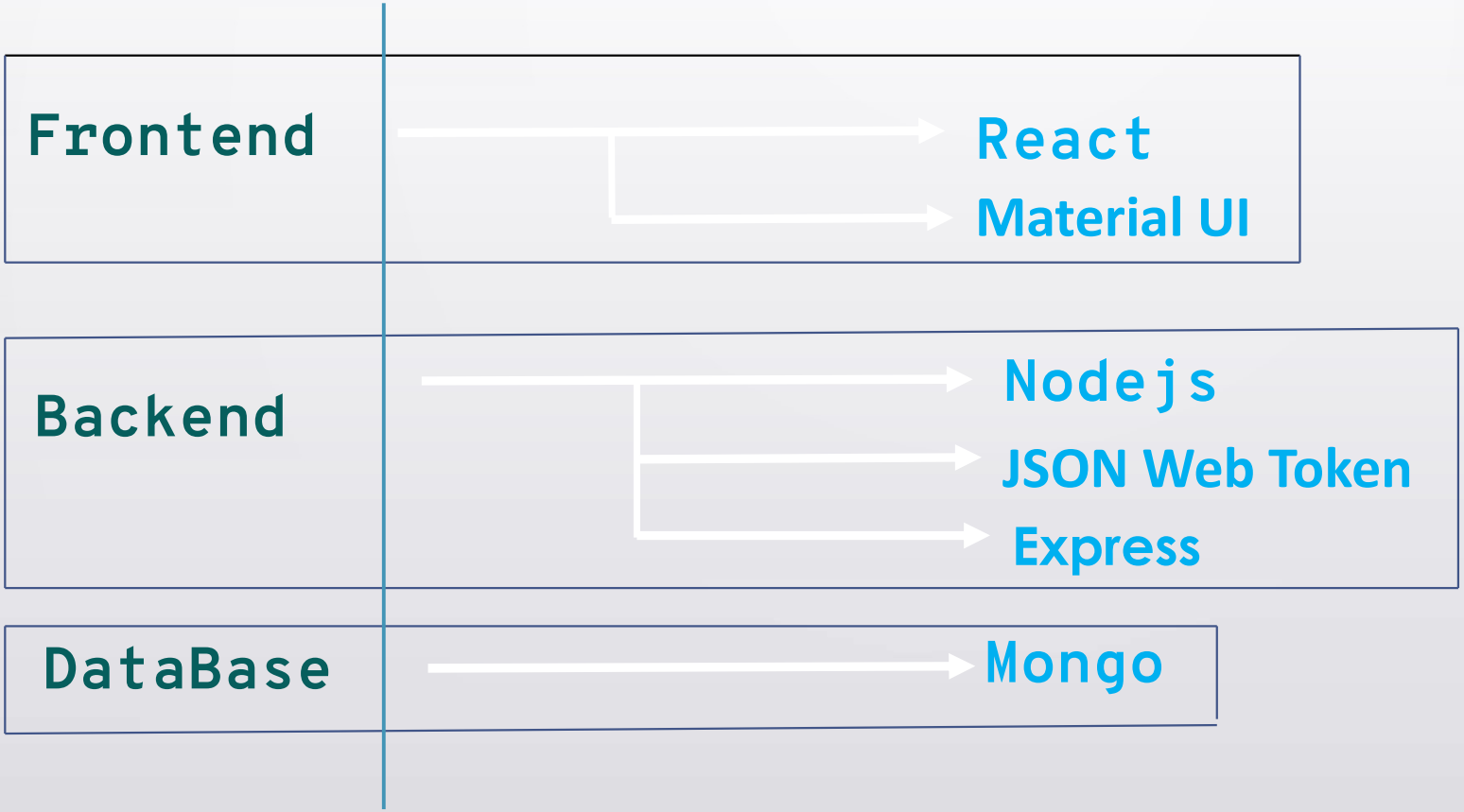
TOPICS TO DISCUSS

- The Challenge
- Technologies used
- The Process
- Control Flow Diagram and ER Diagram
- Outputs

ABOUT THE CHALLENGE :

- Admin , Employee ,Agent and Customer four roles are defined.
- Admin is a super user So, Admin can add Agent, Employee, States, Cities and also view the details of all roles, approve some of the requests of withdrawn Commission and Policies.
- Agent is marketing by sending mail to various customers, for this Agent also get some commission.
- Customer can browse through various insurance plans and schemes and buy whichever plan suits them.

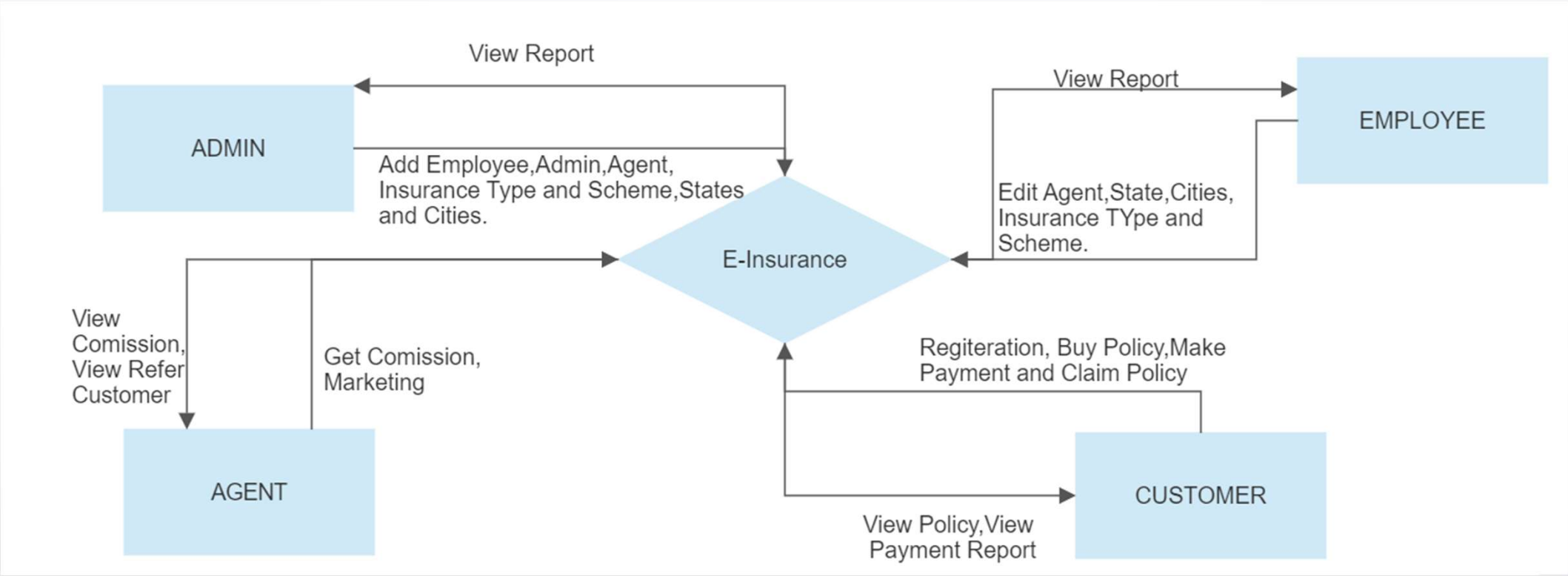
TECHNOLOGIES USED :



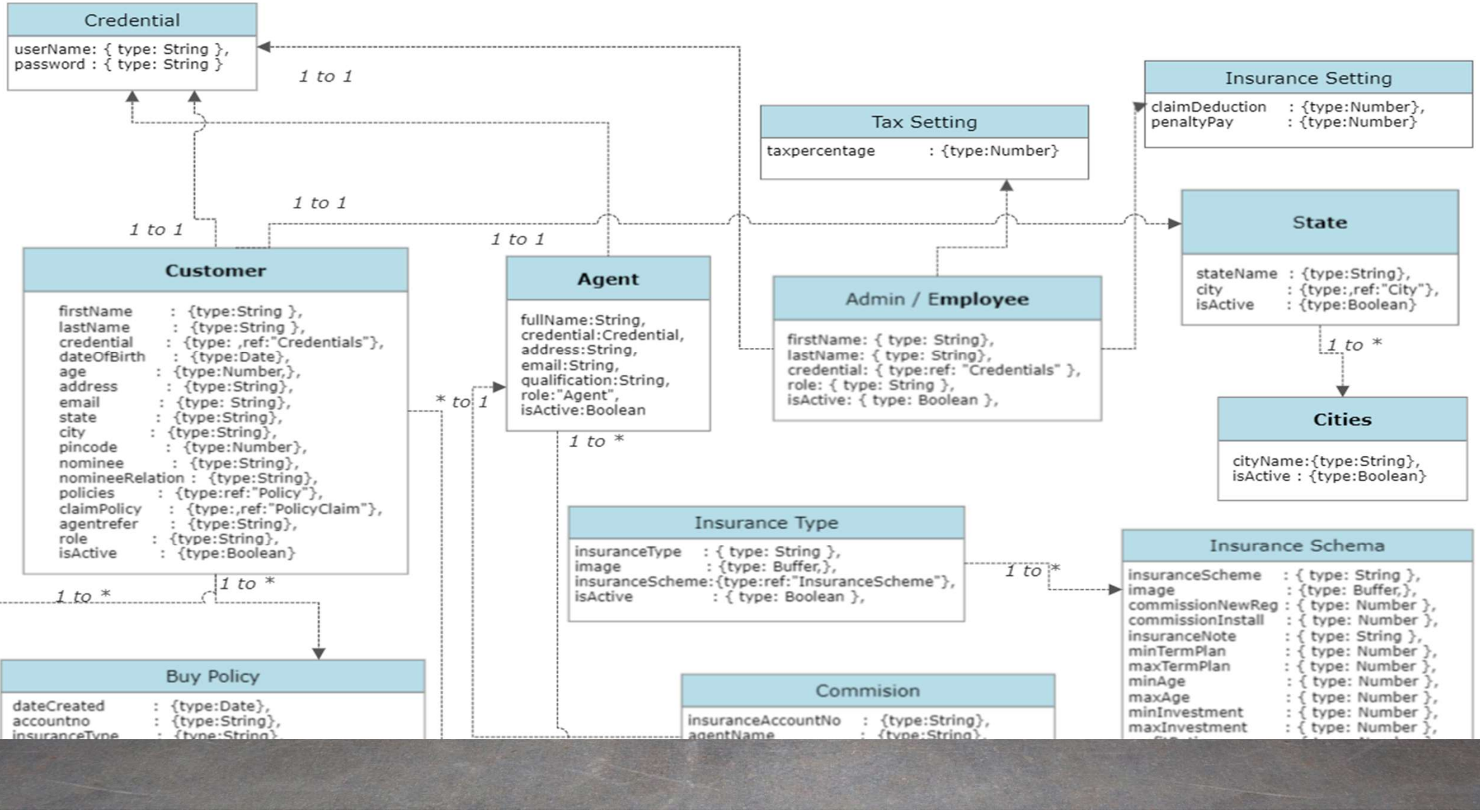
THE PROCESS:

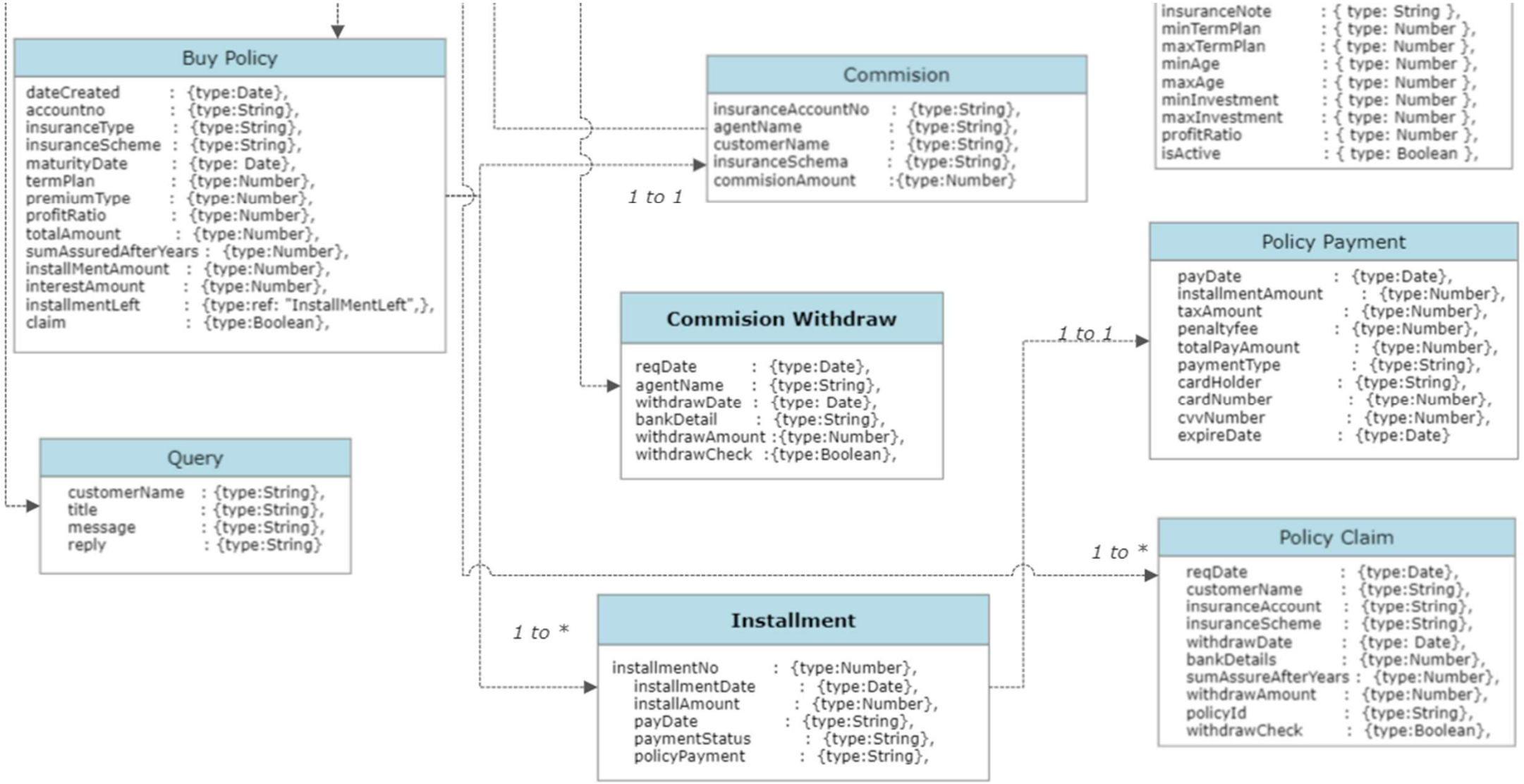
- Identify the Entities
- Control Flow Diagram
- Decide Flow Of Each Role
- Created Class Diagram
- Building REST API
- Making Front-End Modules
- Connecting the Front-End project to database using API
- Testing and Finalising features

CONTROL FLOW DIAGRAM

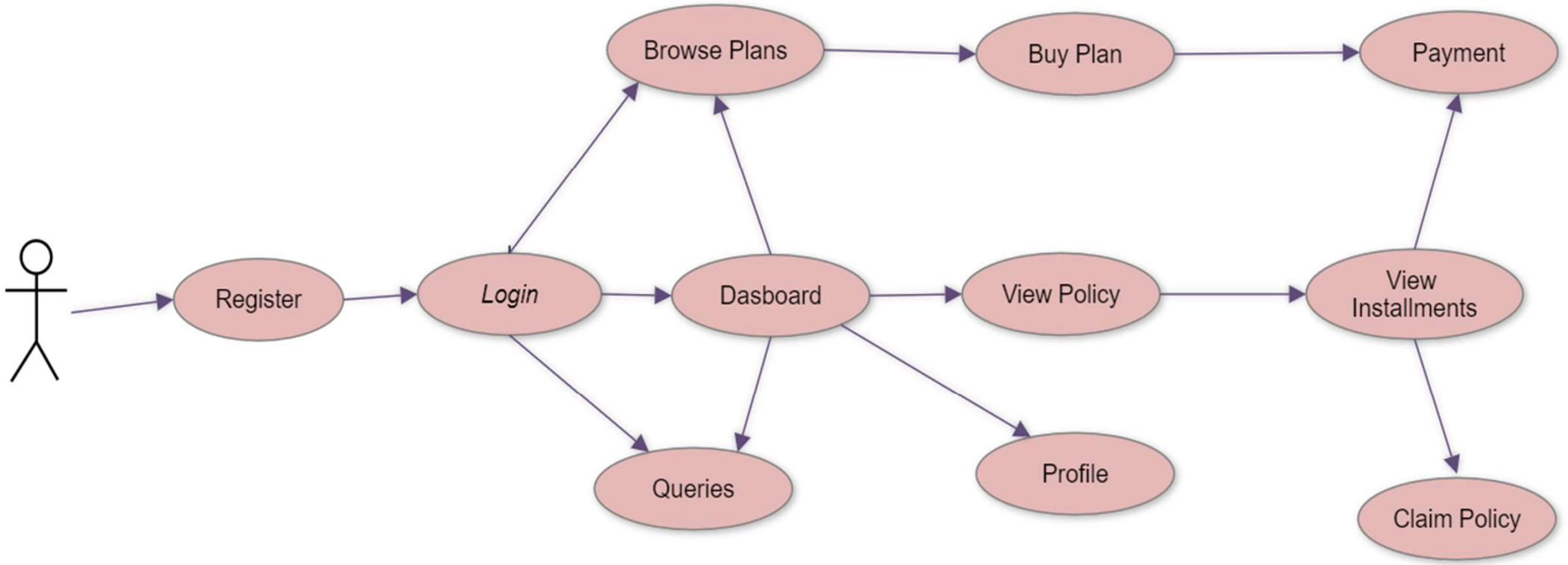


CLASS DIAGRAM

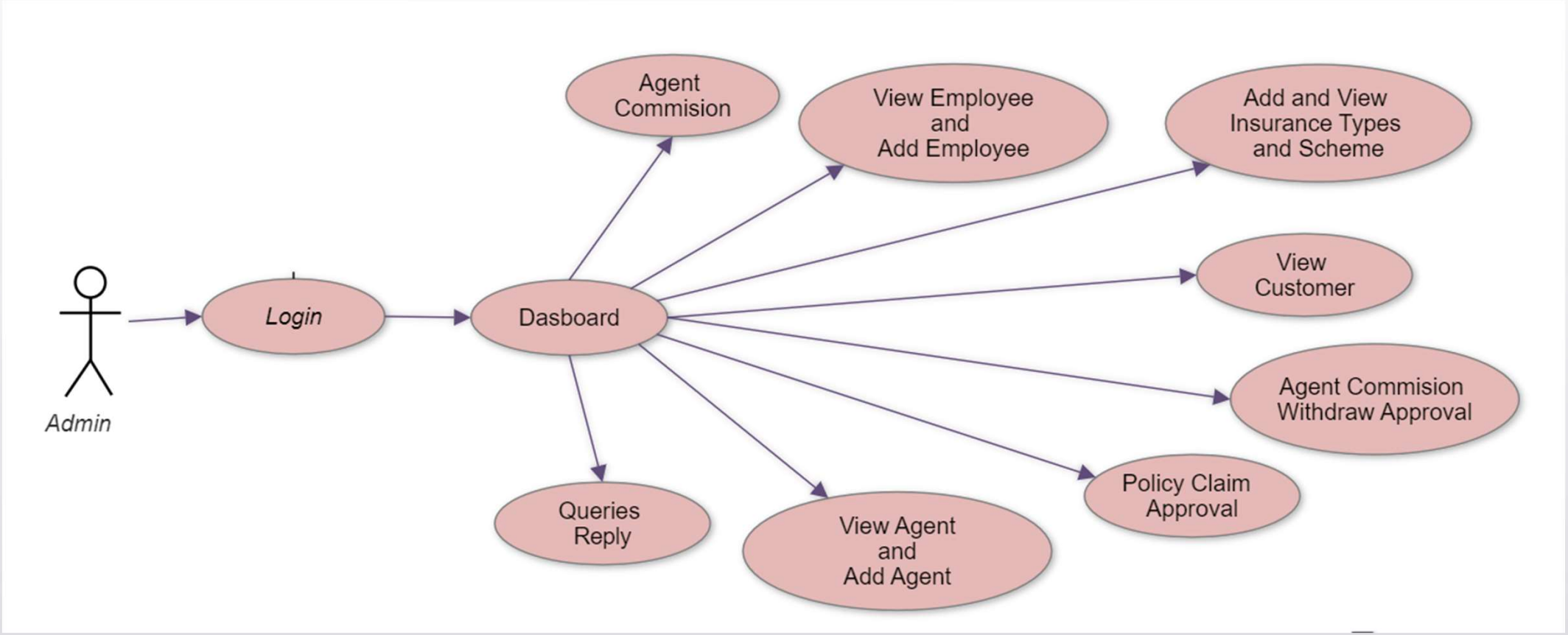

Credential
userName: { type: String },
password : { type: String }
1 to 1
Tax Setting
taxpercentage : {type:Number}
Insurance Setting
claimDeduction : {type:Number},
penaltyPay : {type:Number}
1 to 1
1 to 1
1 to 1
State
stateName : {type:String},
city : {type:,ref:"City"},
isActive : {type:Boolean}
Customer
firstName : {type:String },
lastName : {type:String },
credential : {type: ,ref:"Credentials"},
dateOfBirth : {type:Date},
age : {type:Number,},
address : {type:String},
email : {type: String},
state : {type:String},
city : {type:String},
pincode : {type:Number},
nominee : {type:String},
nomineeRelation : {type:String},
policies : {type:ref:"Policy"},
claimPolicy : {type:,ref:"PolicyClaim"},
agentrefer : {type:String},
role : {type:String},
isActive : {type:Boolean}
Agent
fullName:String,
credential:Credential,
address:String,
email:String,
qualification:String,
role:"Agent",
isActive:Boolean
Admin / Employee
firstName: { type: String},
lastName: { type: String},
credential: { type:ref: "Credentials" },
role: { type: String },
isActive: { type: Boolean },
1 to *
Cities
cityName:{type:String},
isActive : {type:Boolean}
* to 1
1 to *
1 to *
1 to *
Insurance Type
insuranceType : { type: String },
image : {type: Buffer,},
insuranceScheme:{type:ref:"InsuranceScheme"},
isActive : { type: Boolean },
1 to *
Insurance Schema
insuranceScheme : { type: String },
image : {type: Buffer,},
commissionNewReg : { type: Number },
commissionInstall : { type: Number },
insuranceNote : { type: String },
minTermPlan : { type: Number },
maxTermPlan : { type: Number },
minAge : { type: Number },
maxAge : { type: Number },
minInvestment : { type: Number },
maxInvestment : { type: Number },
Buy Policy
dateCreated : {type:Date},
accountno : {type:String},
Commision
insuranceAccountNo : {type:String},


Buy Policy
dateCreated : {type:Date},
accountno : {type:String},
insuranceType : {type:String},
insuranceScheme : {type:String},
maturityDate : {type: Date},
termPlan : {type:Number},
premiumType : {type:Number},
profitRatio : {type:Number},
totalAmount : {type:Number},
sumAssuredAfterYears : {type:Number},
installMentAmount : {type:Number},
interestAmount : {type:Number},
installmentLeft : {type:ref: "InstallMentLeft",},
claim : {type:Boolean},
Commision
insuranceAccountNo : {type:String},
agentName : {type:String},
customerName : {type:String},
insuranceSchema : {type:String},
commisionAmount :{type:Number}
1 to 1
insuranceNote : { type: String },
minTermPlan : { type: Number },
maxTermPlan : { type: Number },
minAge : { type: Number },
maxAge : { type: Number },
minInvestment : { type: Number },
maxInvestment : { type: Number },
profitRatio : { type: Number },
isActive : { type: Boolean },
Policy Payment
payDate : {type:Date},
installmentAmount : {type:Number},
taxAmount : {type:Number},
penaltyfee : {type:Number},
totalPayAmount : {type:Number},
paymentType : {type:String},
cardHolder : {type:String},
cardNumber : {type:Number},
cvvNumber : {type:Number},
expireDate : {type:Date}
Commision Withdraw
reqDate : {type:Date},
agentName : {type:String},
withdrawDate : {type: Date},
bankDetail : {type:String},
withdrawAmount :{type:Number},
withdrawCheck :{type:Boolean},
1 to 1
Query
customerName : {type:String},
title : {type:String},
message : {type:String},
reply : {type:String}
1 to *
1 to *
Installment
installmentNo : {type:Number},
installmentDate : {type:Date},
installAmount : {type:Number},
payDate : {type:String},
paymentStatus : {type:String},
policyPayment : {type:String},
Policy Claim
reqDate : {type:Date},
customerName : {type:String},
insuranceAccount : {type:String},
insuranceScheme : {type:String},
withdrawDate : {type: Date},
bankDetails : {type:Number},
sumAssureAfterYears : {type:Number},
withdrawAmount : {type:Number},
policyId : {type:String},
withdrawCheck : {type:Boolean},

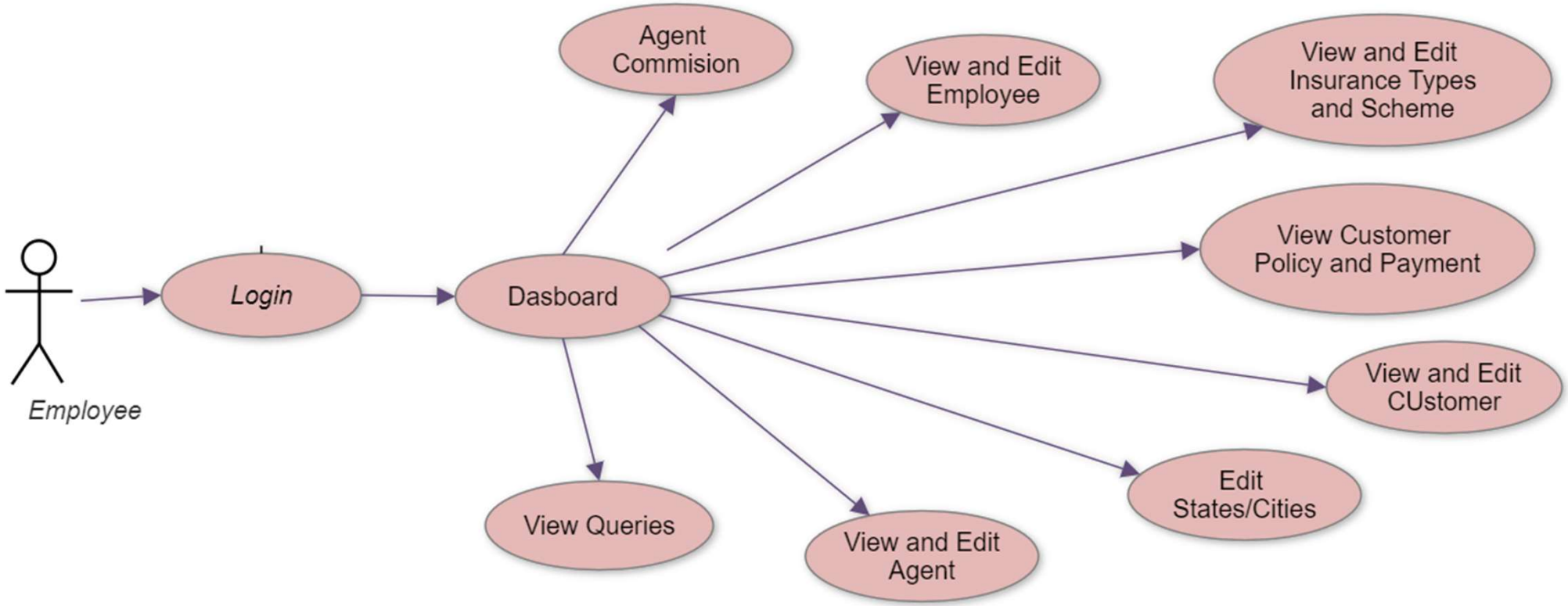
CUSTOMER FLOW



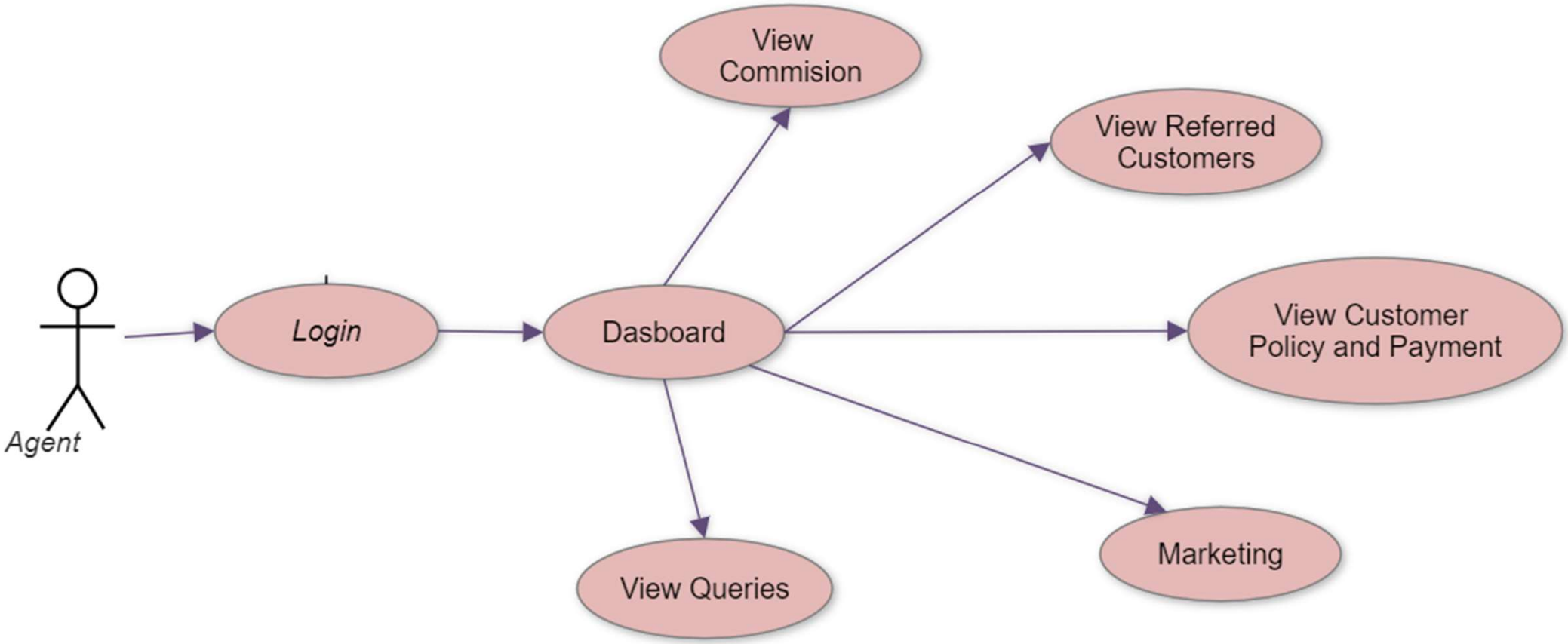
ADMIN FLOW


Admin
Login
Dasboard
Agent Commision
View Employee and Add Employee
Add and View Insurance Types and Scheme
View Customer
Agent Commision Withdraw Approval
Policy Claim Approval
View Agent and Add Agent
Queries Reply

EMPLOYEE FLOW

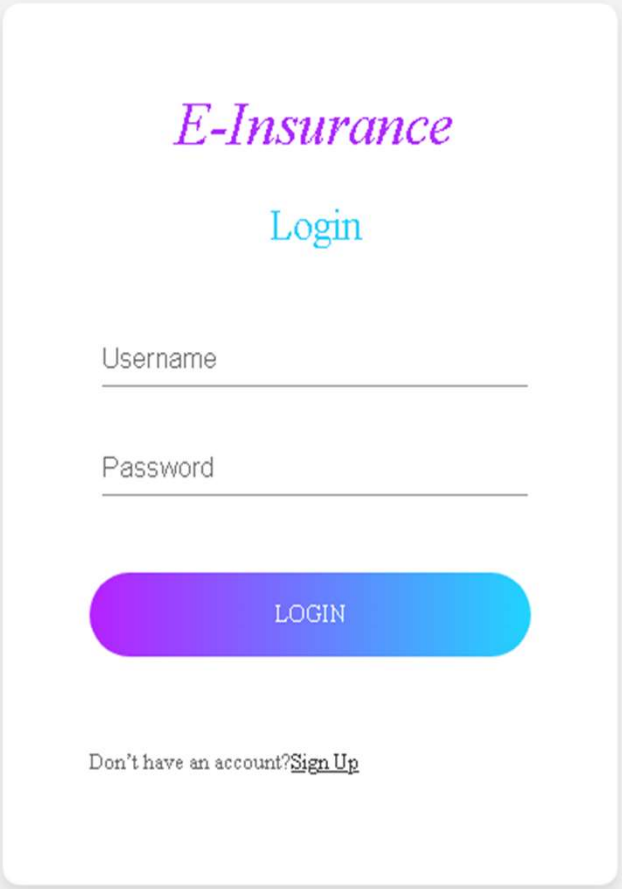


AGENT FLOW


View
Commision
View Referred
Customers
Agent
Login
Dasboard
View Customer
Policy and Payment
Marketing
View Queries

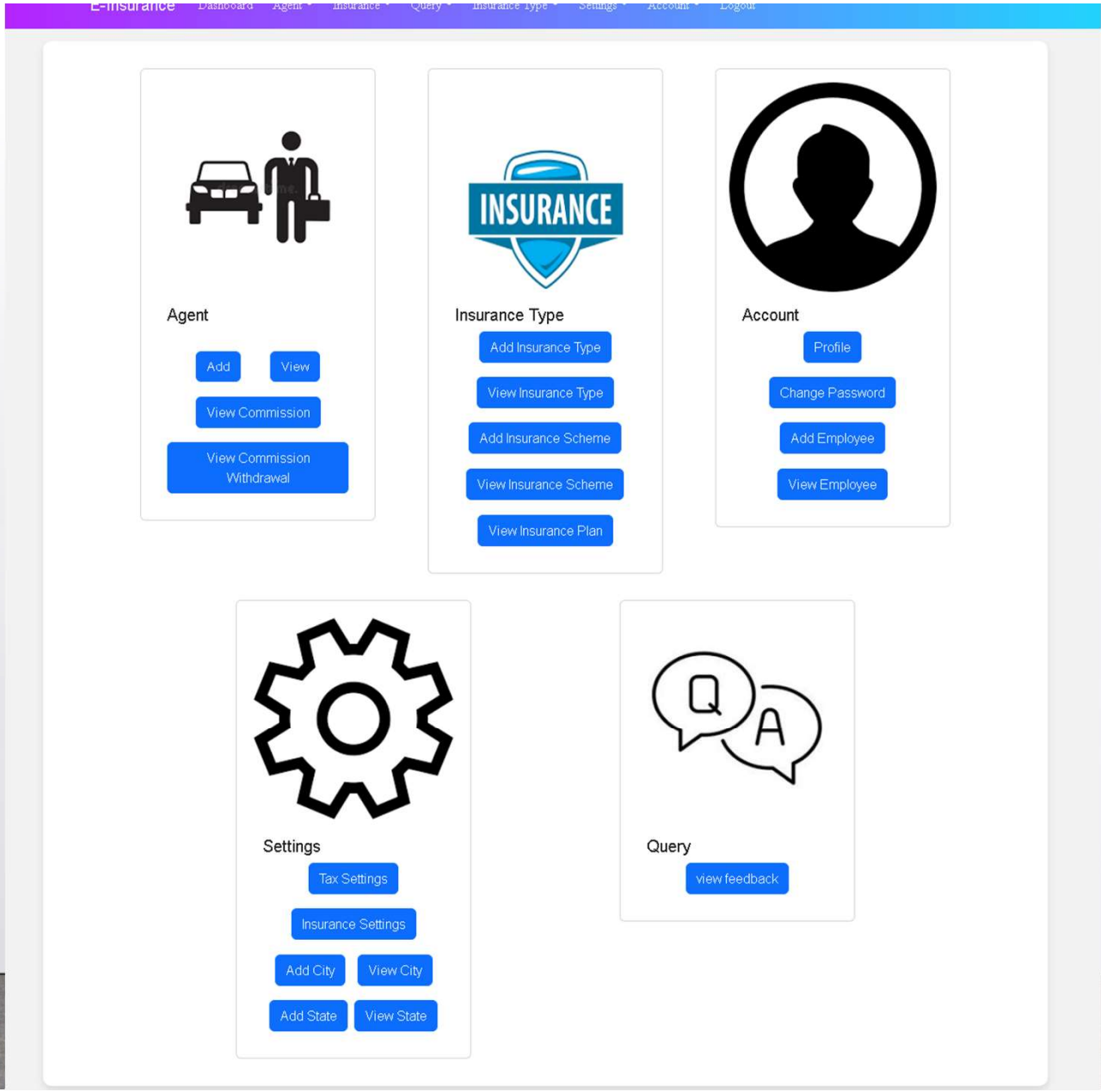
OUTPUT:

Login Page:

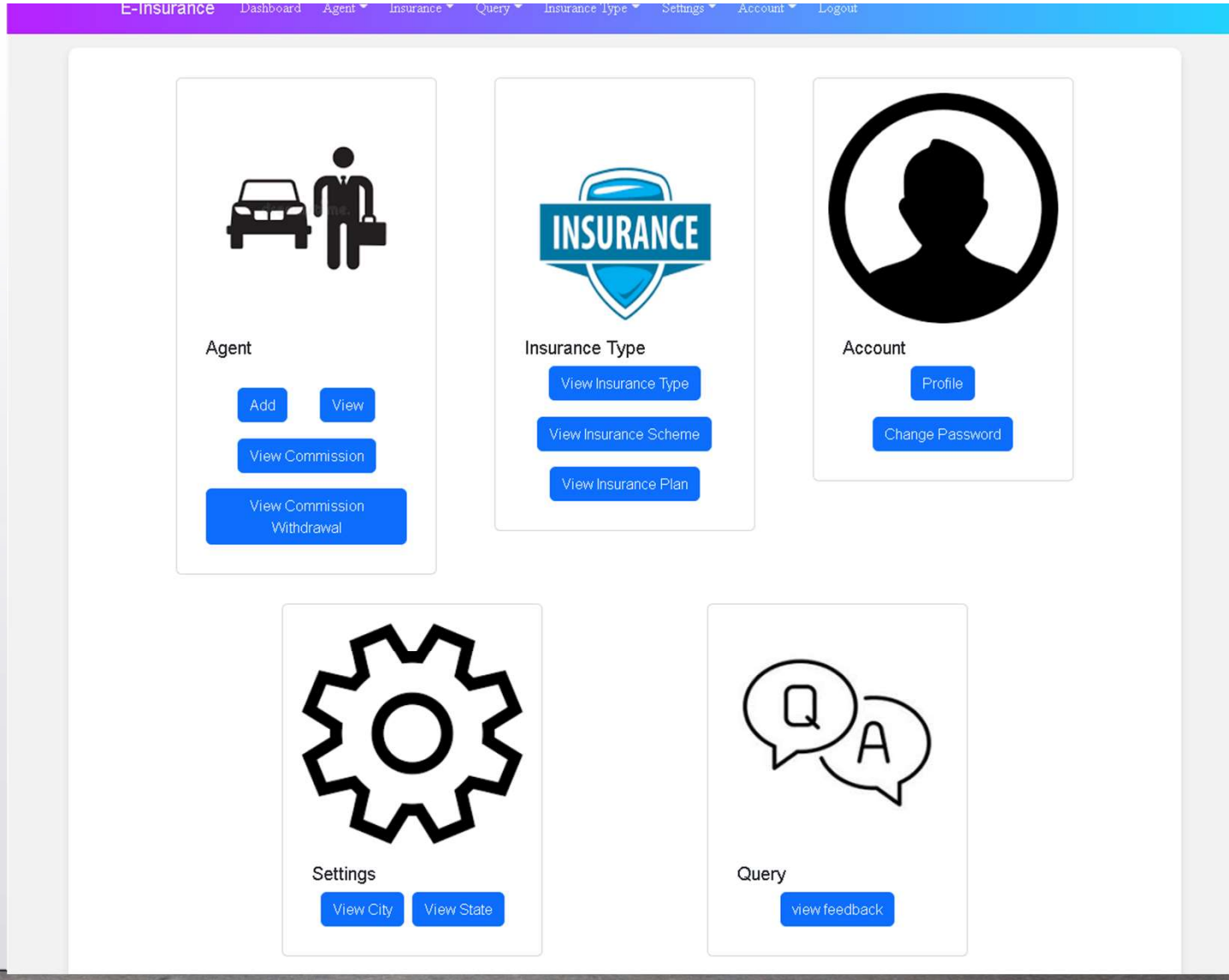


ADMIN

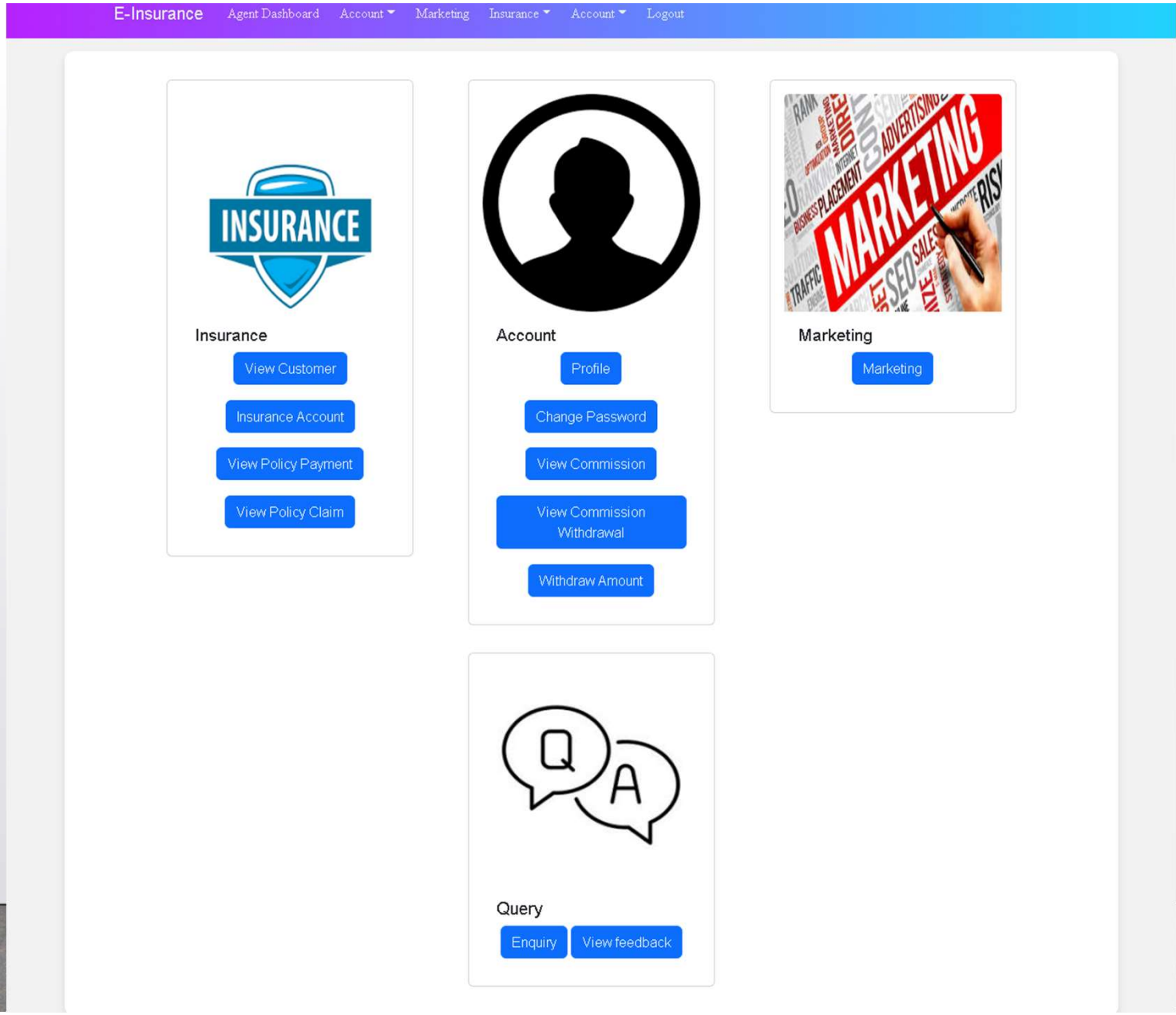
Dashboard :



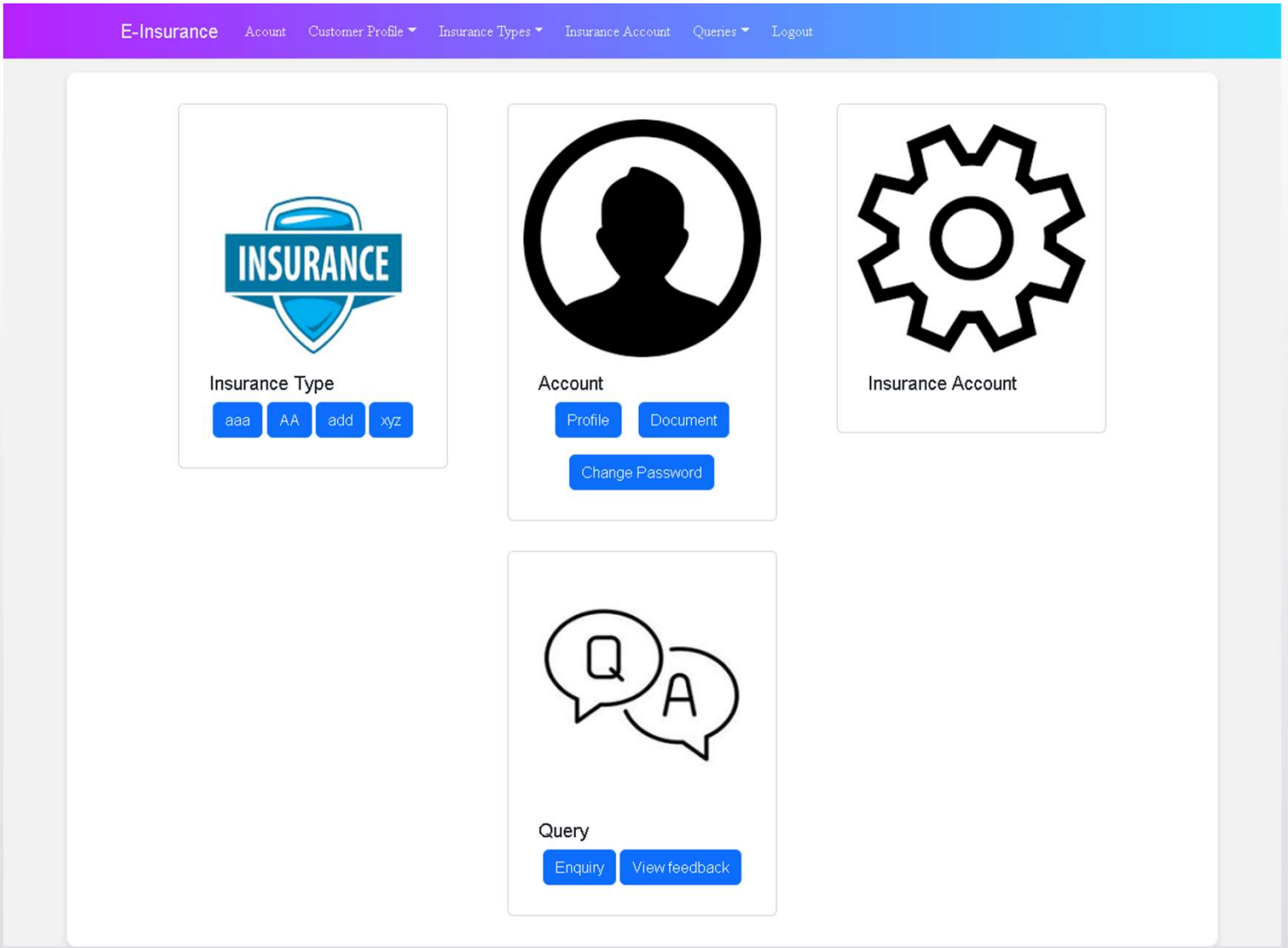
EMPLOYEE DASHBOARD :



AGENT DASHBOARD :



CUSTOMER DASHBOARD :



Module used for :

- Backend Validation : Validator
- Hashing Password : Bcrypt
- Convert to pdf : Canvas and Jspdf
- Encode Image : Multer

THANK YOU