Software Requirements Specification

(SRS)

ISTM-Insurance System Tracking Manager

Author(s):Suresh

Version :2.0

Date First Issued:

Updated:

OVERVIEW

This software provides five types of Insurance services, which includes Life Insurance, medical Insurance, and Motor Insurance. Home Insurance, Travel Insurance. This software provides loan facility for Motor Purchasing. The details can be viewed and updated by the officials of the company.

STUDY OF THE SYSTEM

In the flexibility of uses the interface has been developed a graphics concepts in mind, associated through a browser interface. The GUI's at the top level has been categorized as follows

1. Administrator Interface Design.
2. User Interface.
3. Security Authentication.
4. Reports.
5. General end-users.

The administrative user interface will maintain the different users details, the interface helps the administration with all the transactional states like which users sending the mails, and which users receiving whishing mails, users details information history. And the statistics of the system in difference stratagies.

Proposed System:

- The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach by system Rich User Interface.
- The user information files can be stored in centralized database which can be maintained by the system.
- A User can view the details of various policies and schemes offered by the Insurance Company.
- New Users can register with the site so that he can get information online.
- An existing policyholder can view his policy details and calculate the premium.
- The web site provides information about the new strategies and subsidiary schemes of the company.
- Provides loan facility for policyholders and online payments.
- Provides Loan EMI calculator
- Provides Interest calculator
- Administrator gives the approval for the new users
- Administrator edit modify and delete, upload certain information
- Provide the facility to send the statements and loan EMI details in pdf format to users mail ids.

No of Modules:

1. Registered Users
2. Company Officials
3. Administrator
4. Reports
5. Security and Authentication

1. Registered Users

- A User can view the details of various policies and schemes offered by the Insurance company.
- New Users can register with the site so that he can get information online.
- An existing policyholder can view his policy details and calculate the premium.
- The web site provides information about the new strategies and subsidiary schemes of the company.
- Provides loan facility for policyholders and online payments.
- Provides Loan EMI calculator
- Provides Interest calculator

2. Company Officials

- Company officers can provide different types of Insurances details.
- Officials can update policy holder's information like premiums, installments etc.
- Official can approval user policy.
- Officials can upload user's policy details into company database.

3. Administrator

- Administrator gives the approval for the new users
- Administrator edit modify and delete, upload certain information.
- Administrator can add Company officials.
- Admin can provide policy types and sub type through this portal.

4. Reports

- Interest Calculation and EMI
- Customer Insurance Policy Payment Reports
- Customer Loan Payment Reports.

5. Security and Authentication Module:

The user details should be verified against the details in the user tables and if it is valid user, they should be entered into the system. Once entered, based on the user type access to the different modules to be enabled / disabled and individual user can change their default password or old password.

Data Flow Diagrams

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also know as a data flow graph or a bubble chart.

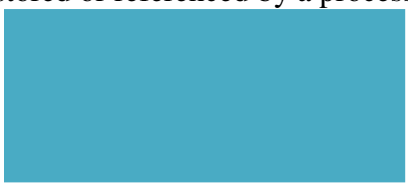
DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts. The Basic Notation used to create a DFD's are as follows:

1. Dataflow: Data move in a specific direction from an origin to a destination.

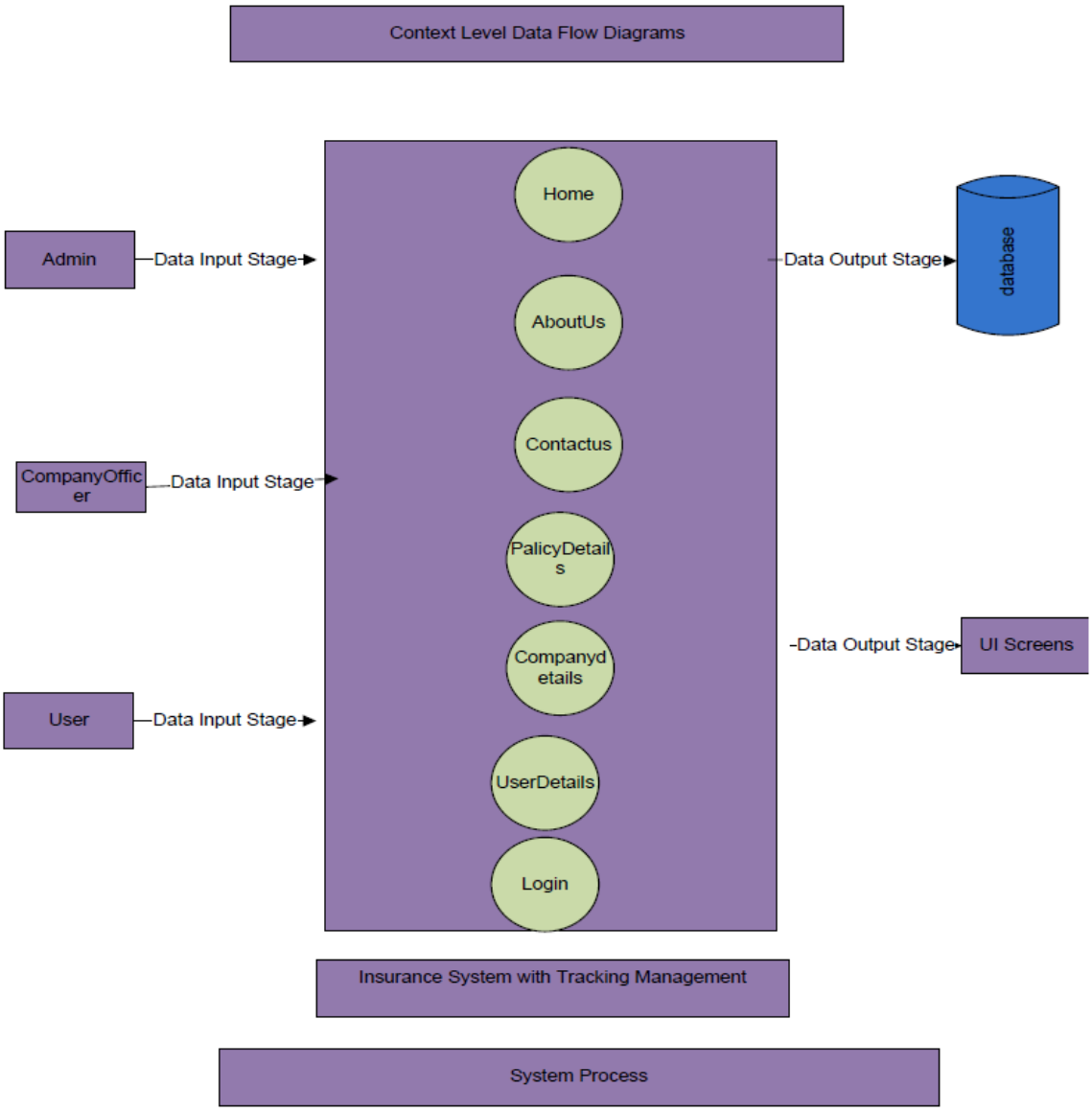
2. Process: People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

3. Source: External sources or destination of data, which may be People, programs, organizations or other entities.

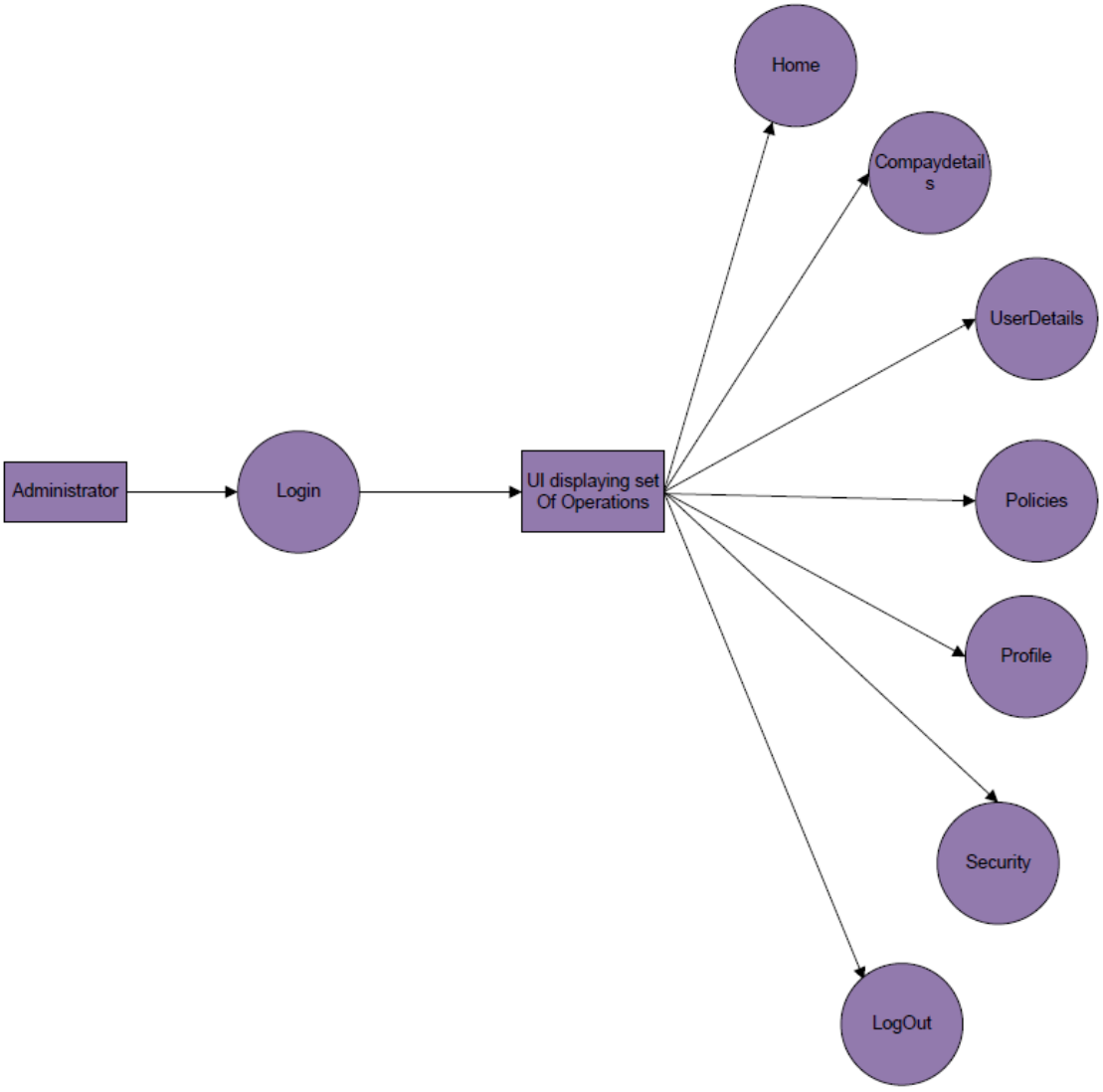
4. Data Store: Here data are stored or referenced by a process in the System.



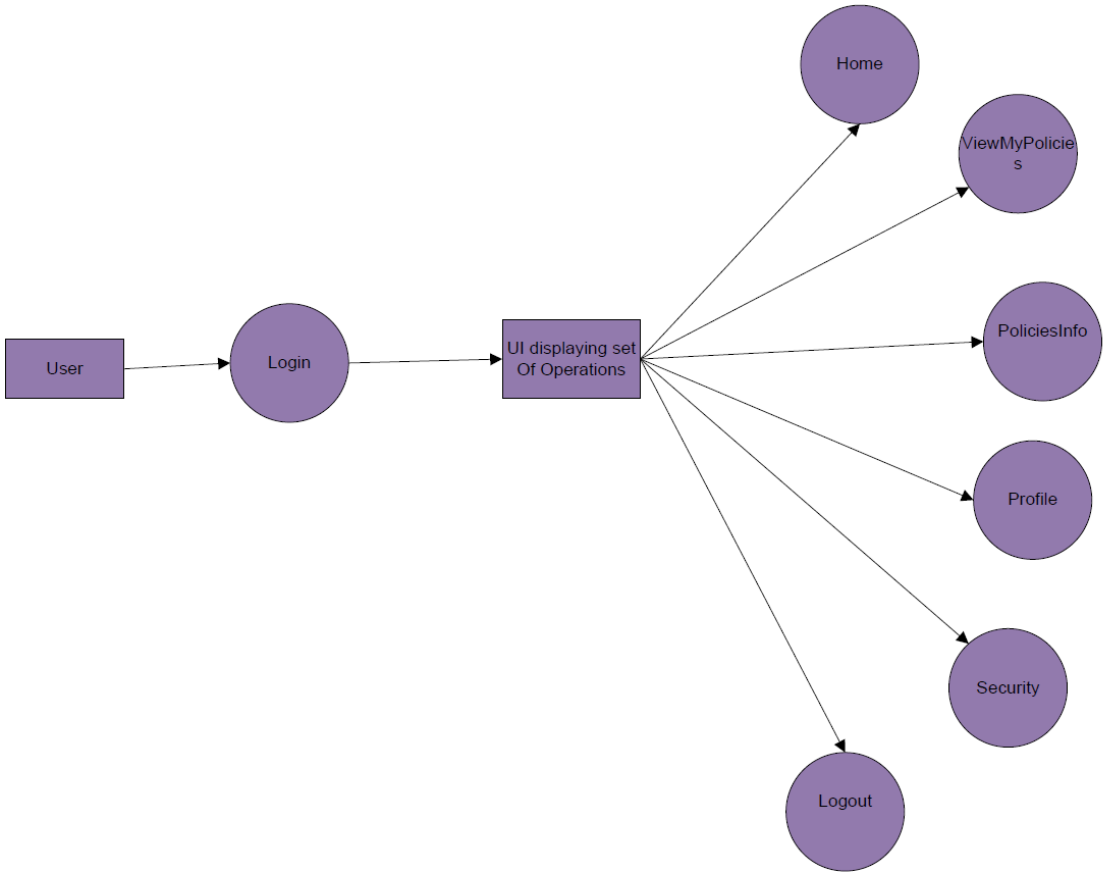
Context Level Data Flow Diagram



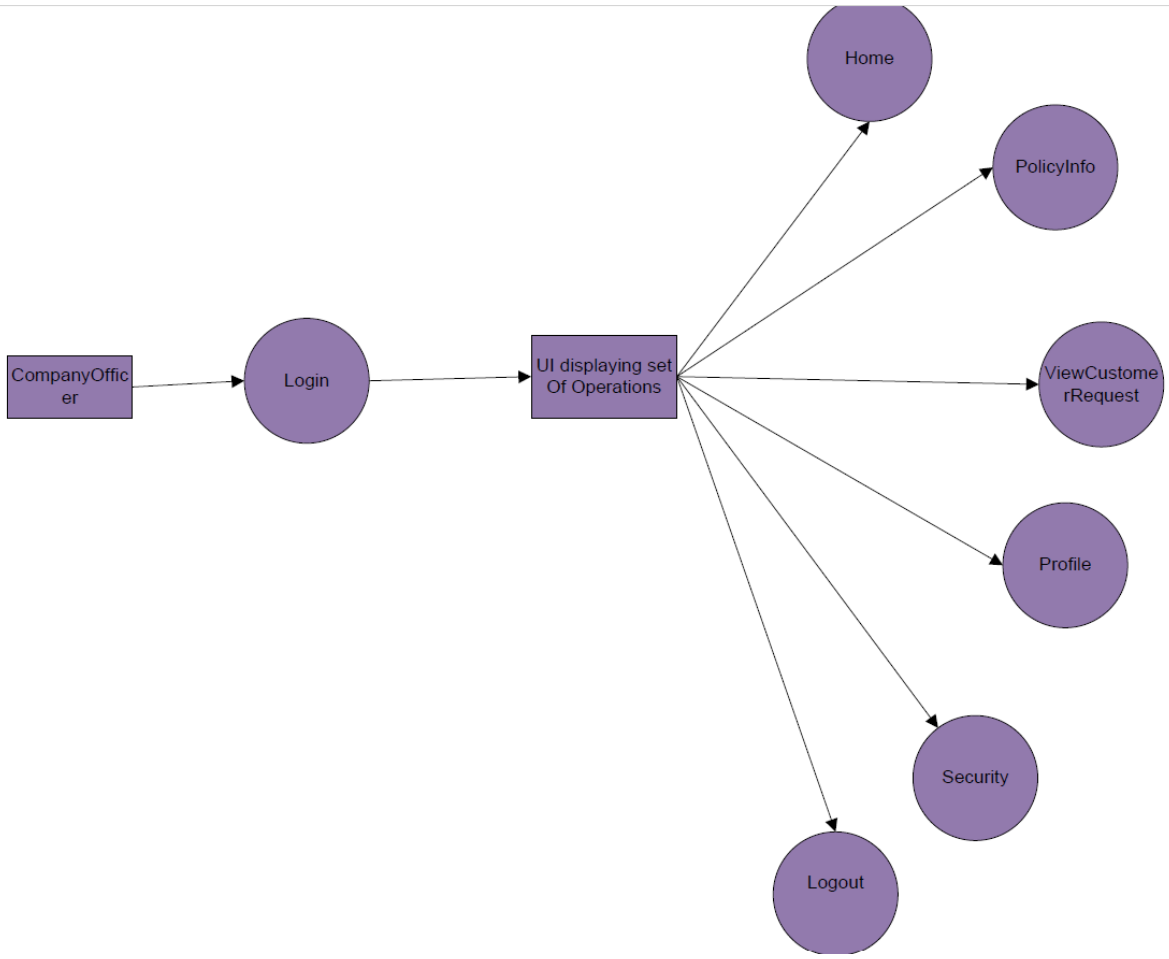
Level1 Data Flow Diagram for Administrator:



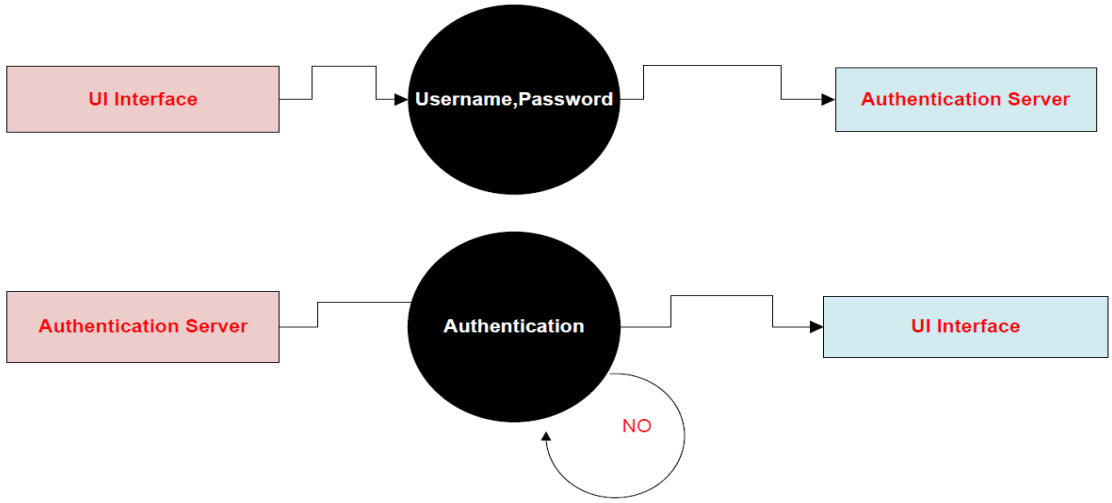
Level1 Data Flow Diagram for User:



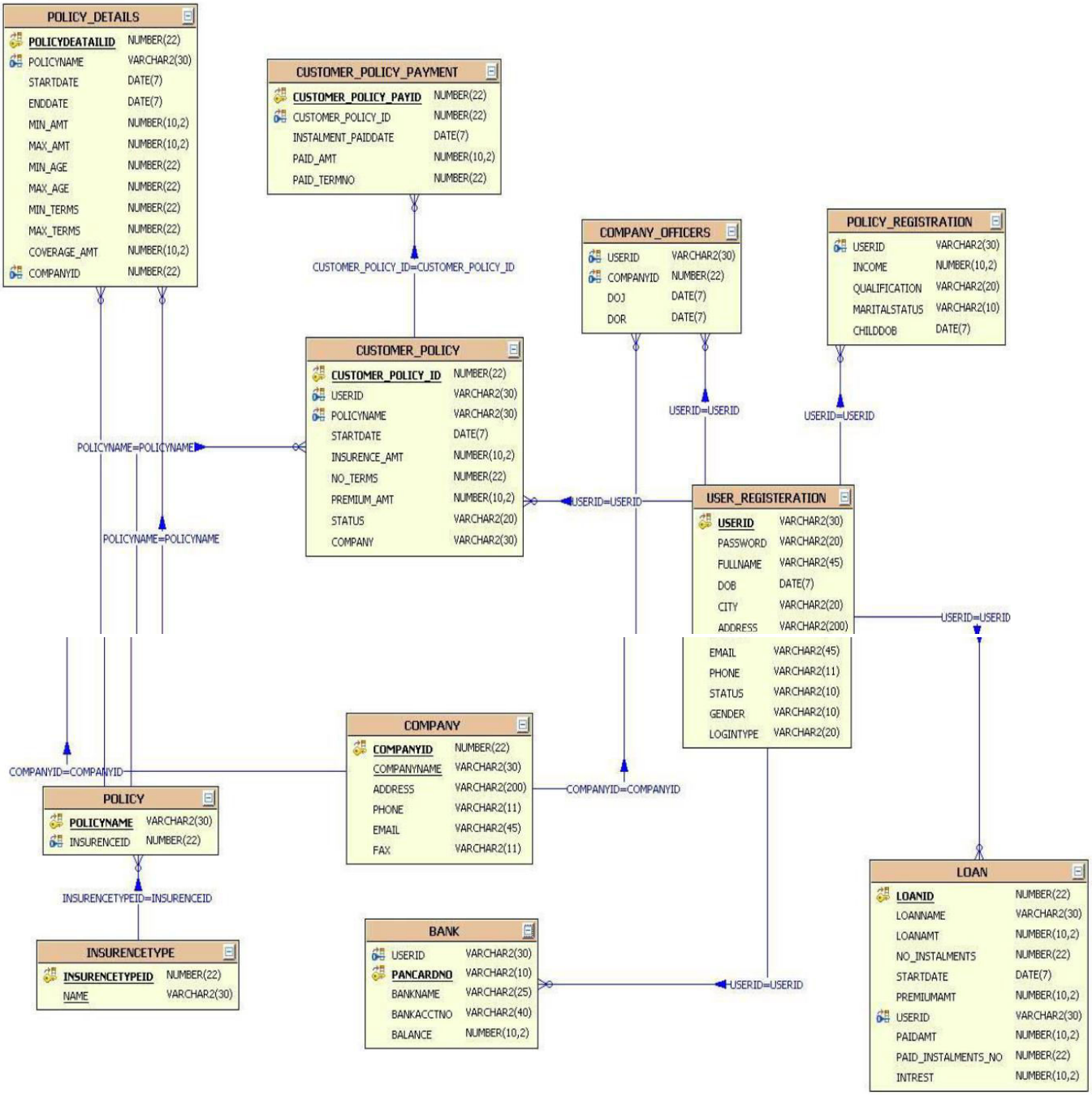
Level1 Data Flow Diagram for Company Officer:



Authentication Data Flow Diagram:

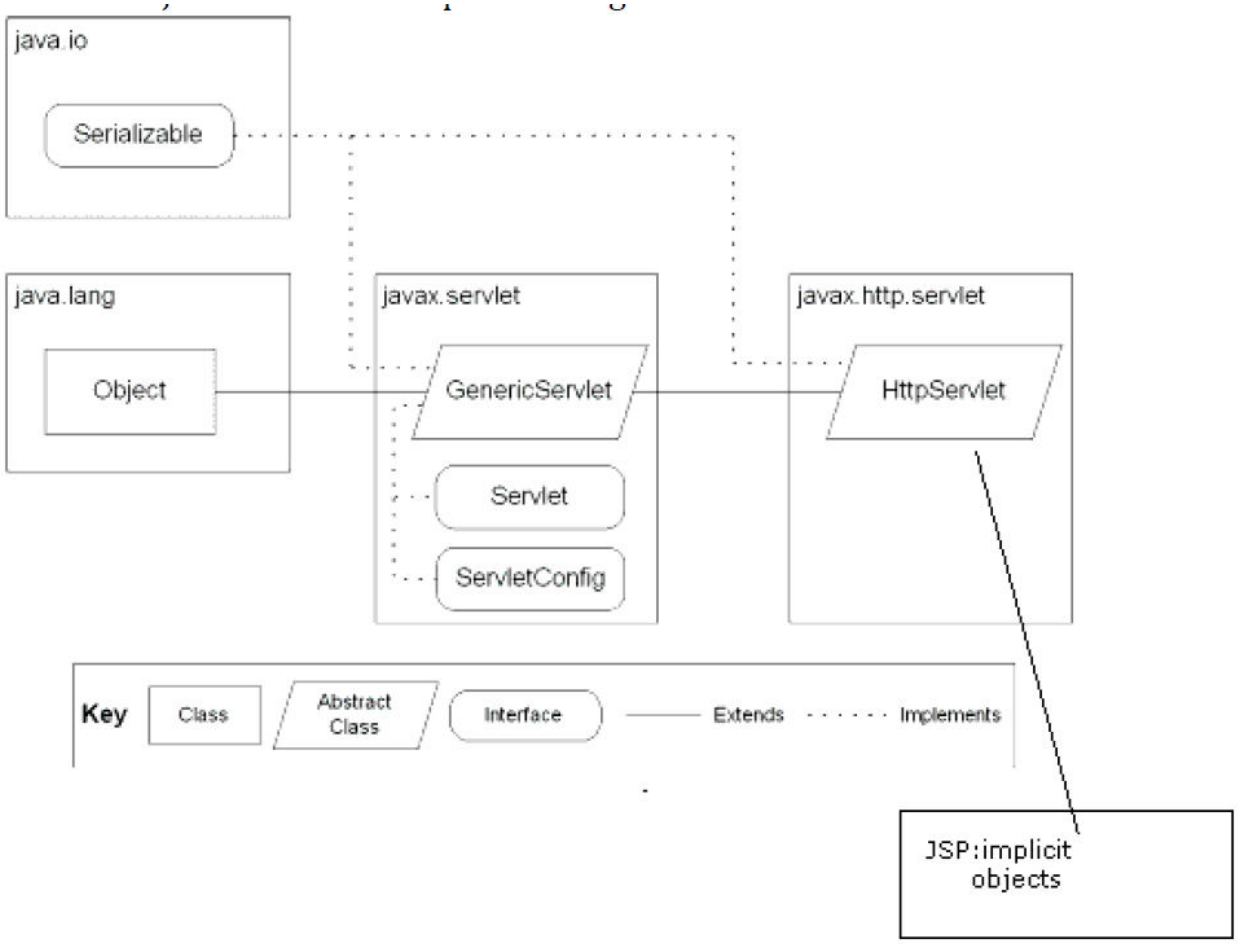


E - R Diagrams

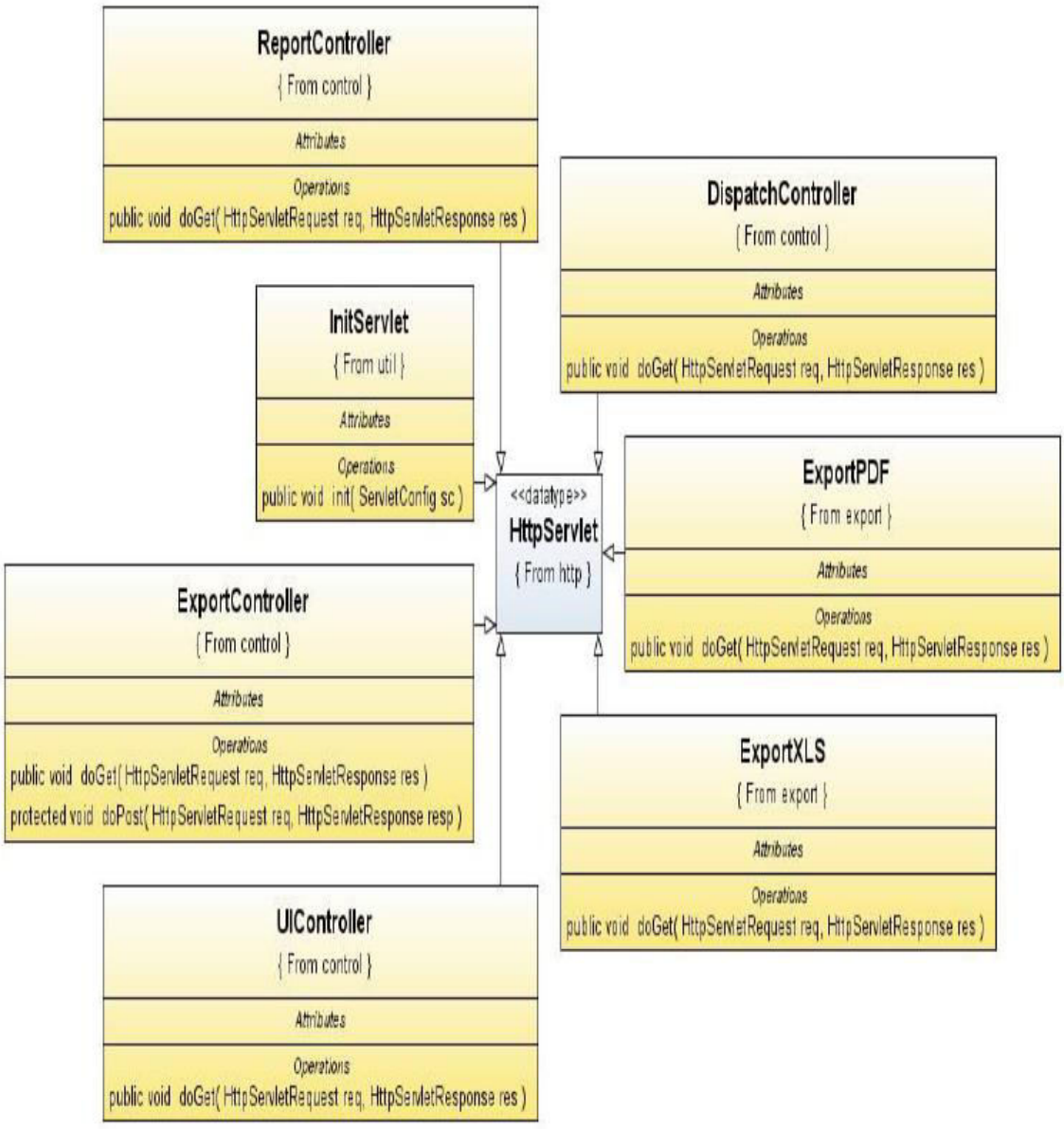


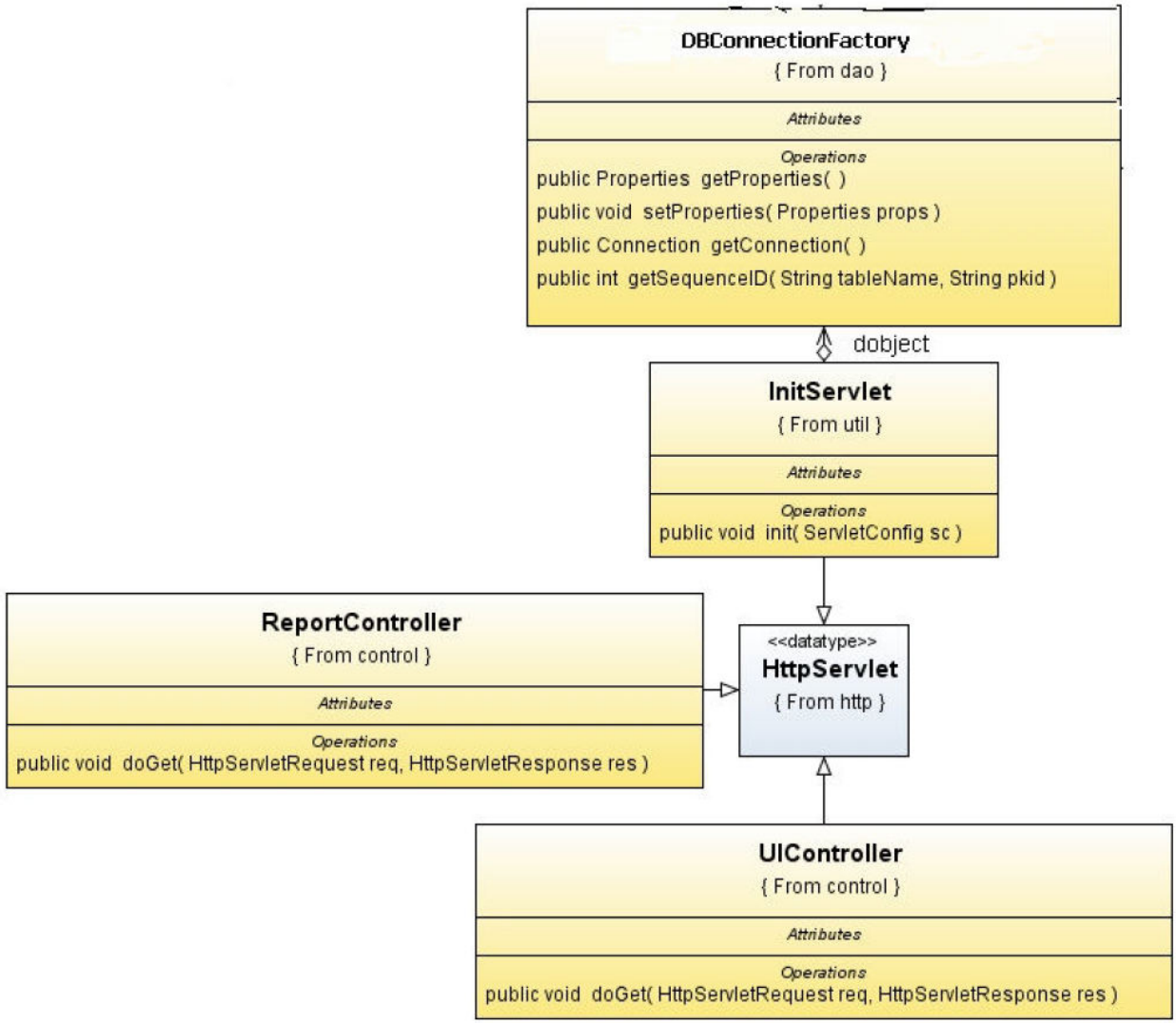
CLASS DIAGRAM

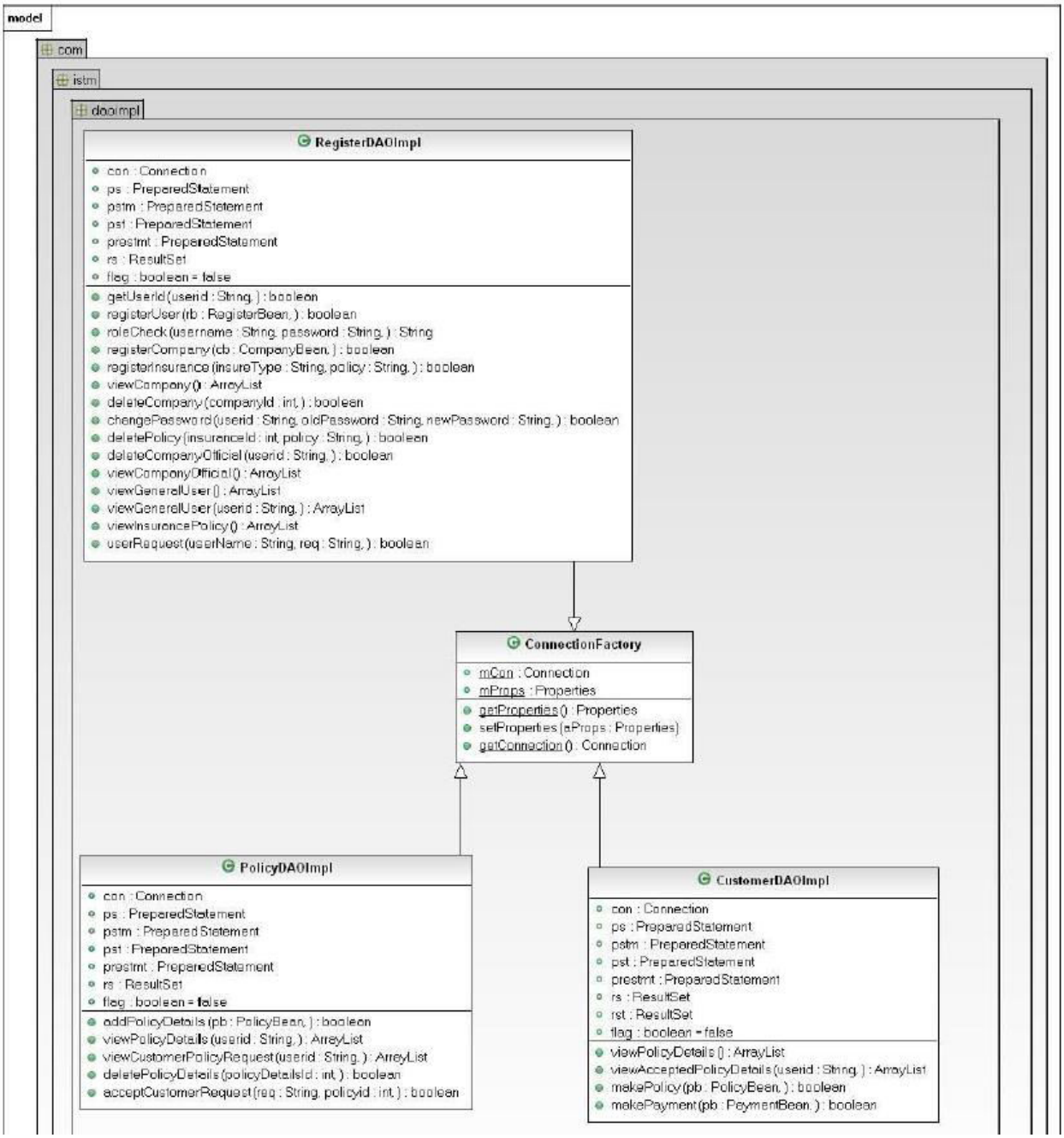
Class diagrams describe the structure of the system in terms of classes and objects. The servlet api class diagram will be as follows.



Class Collaboration Diagram







UML Diagrams

Unified Modeling Language:

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

•User Model View

1. This view represents the system from the users perspective.
2. The analysis representation describes a usage scenario from the end-users perspective.

•Structural model view

1. In this model the data and functionality are arrived from inside the system.
2. This model view models the static structures.

•Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

•Implementation Model View :

In this the structural and behavioral as parts of the system are represented as they are to be built.

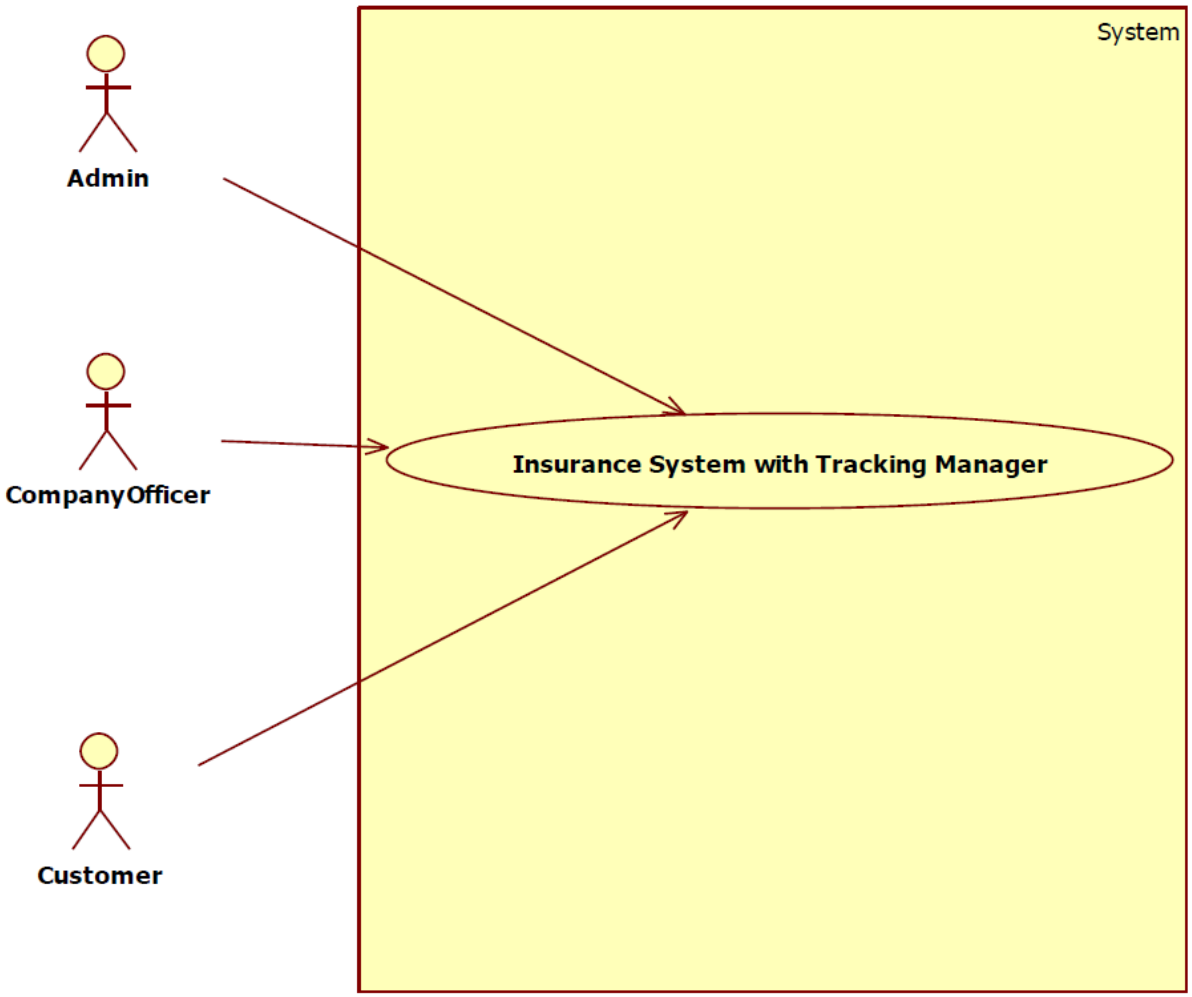
•Environmental Model View

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

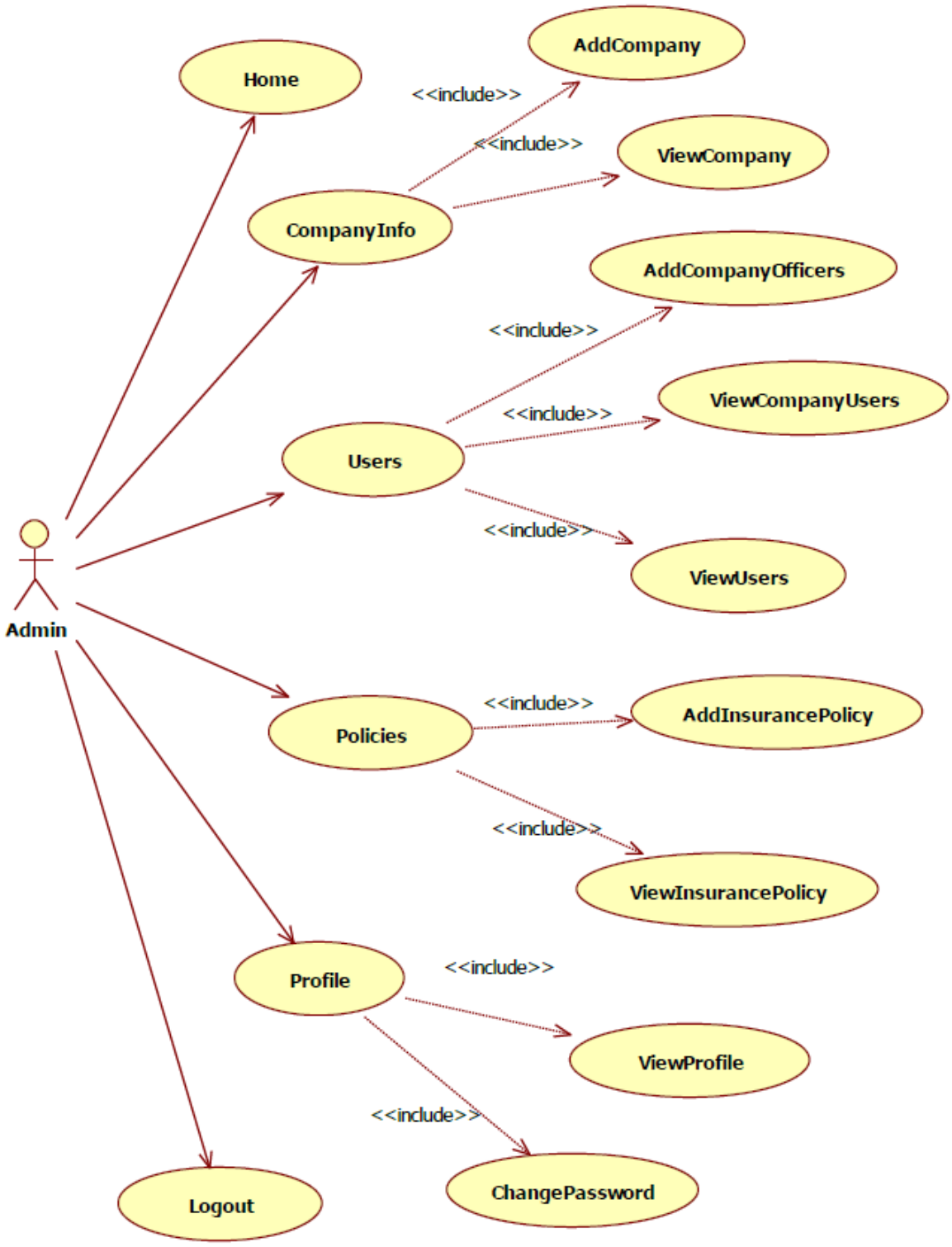
UML is specifically constructed through two different domains they are:

1. UML Analysis modeling, this focuses on the user model and structural model views of the system.
2. UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.
3. Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.
4. Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

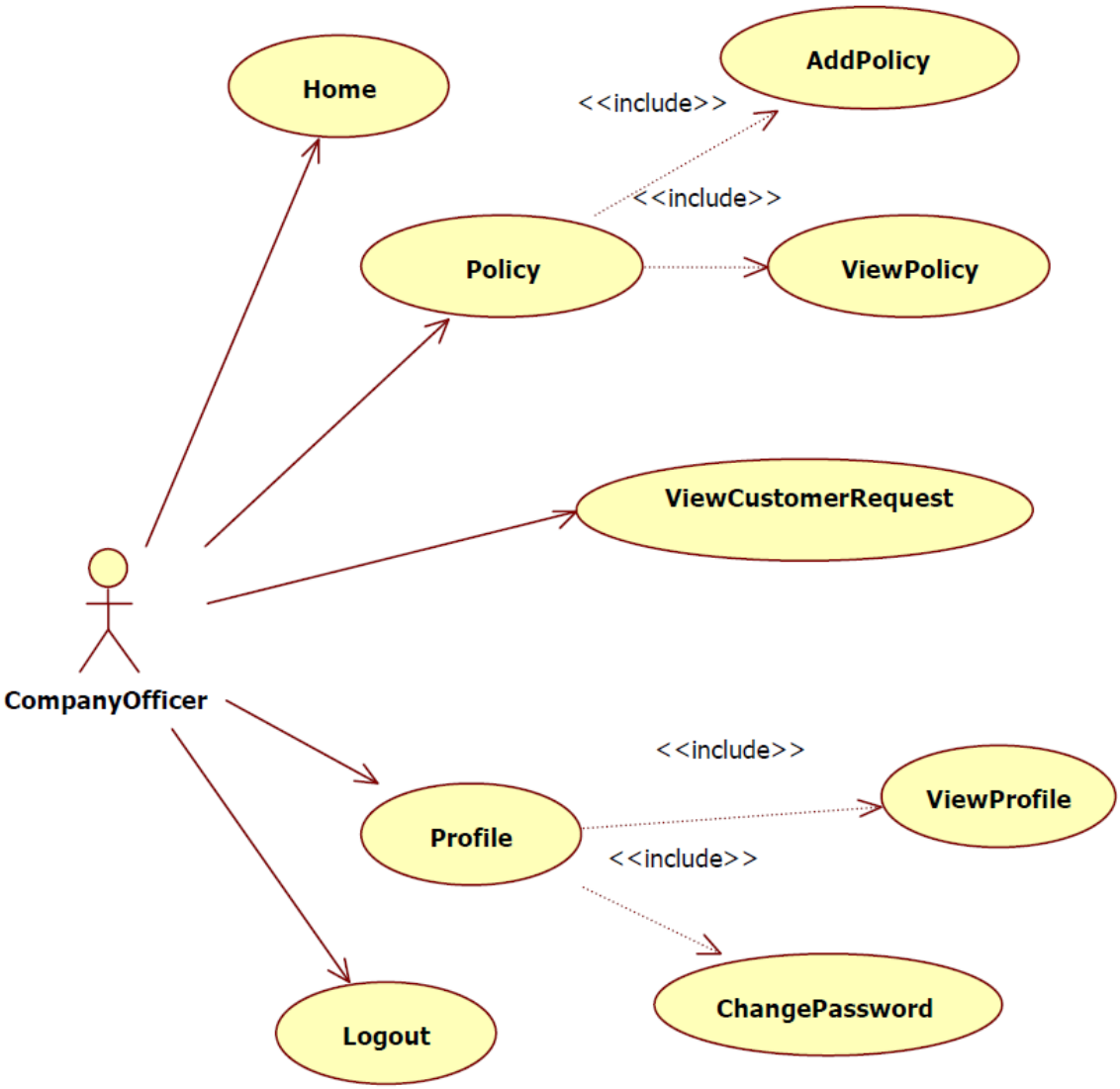
1. **System Use Case Diagram**



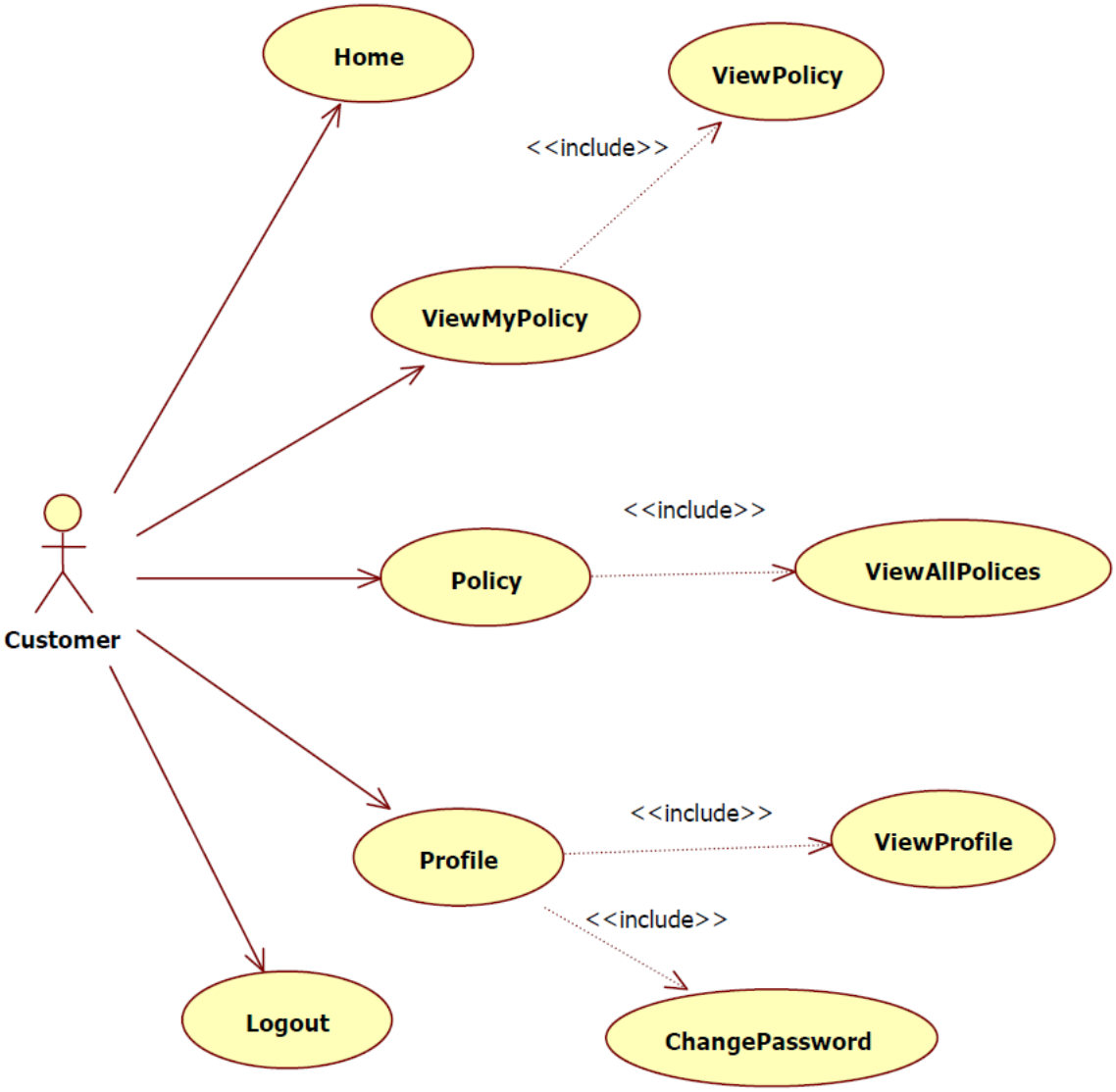
Admin Use Case



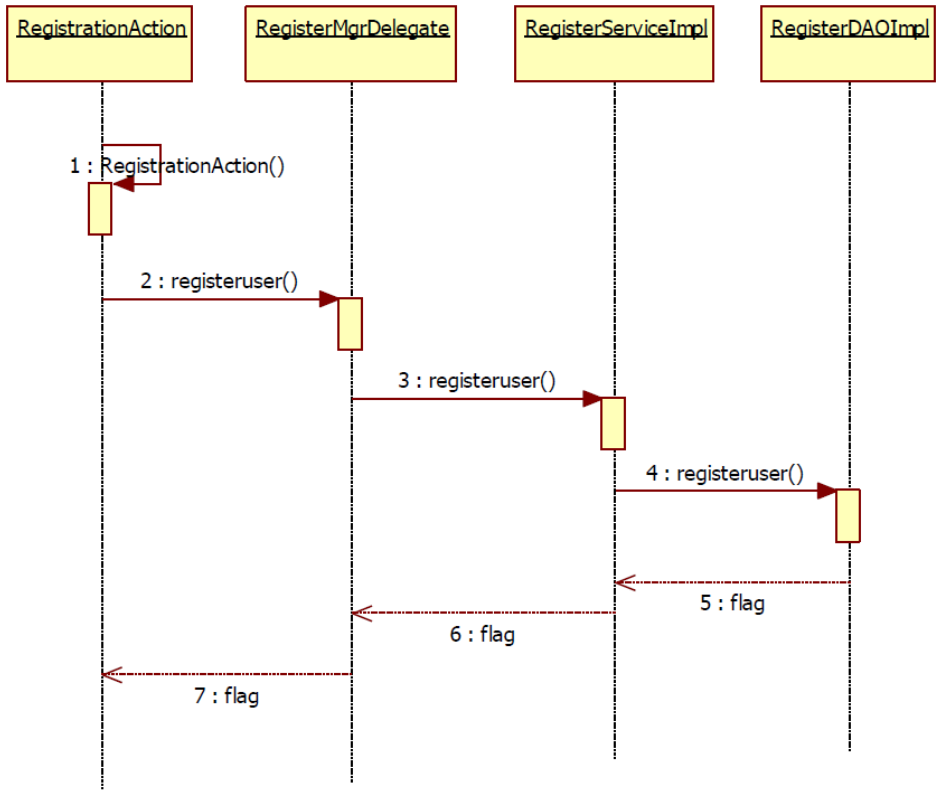
Company Officer Use case



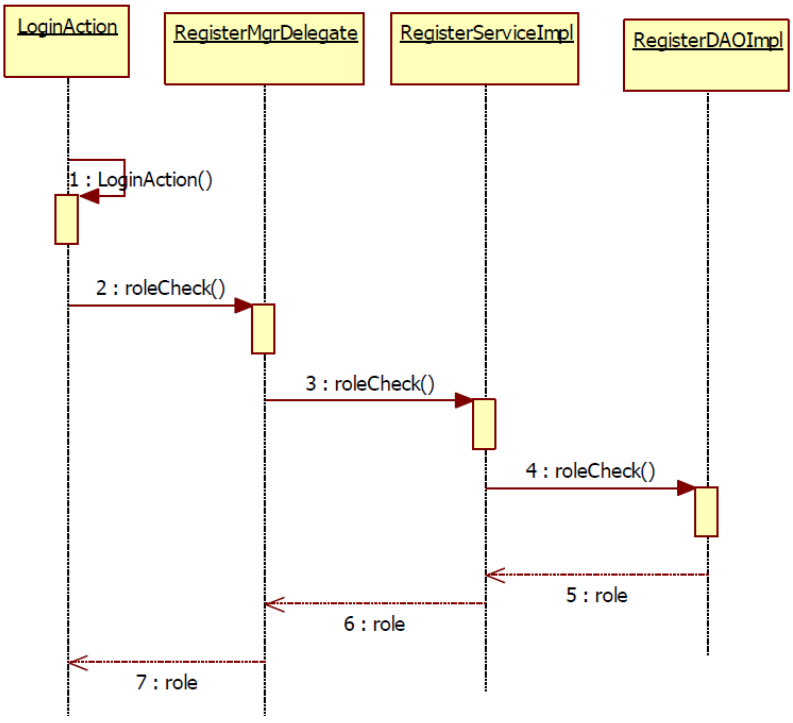
Customer Use Case



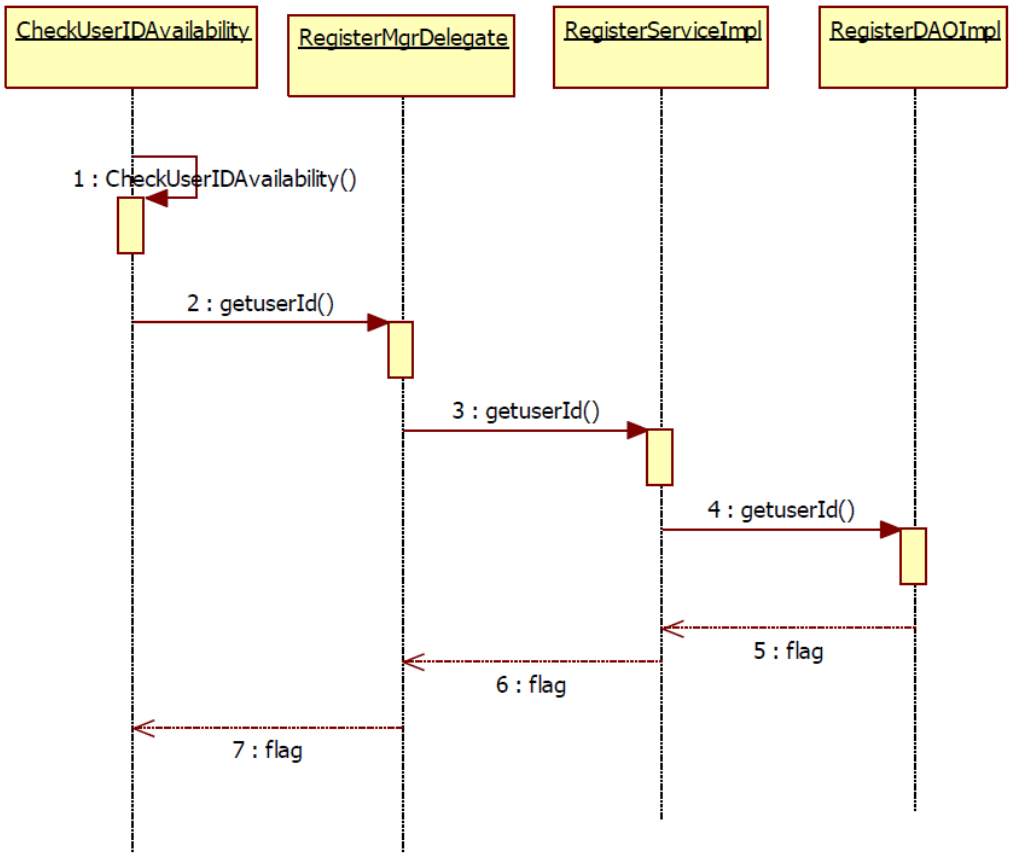
Sequence Diagrams : Register Action Sequence



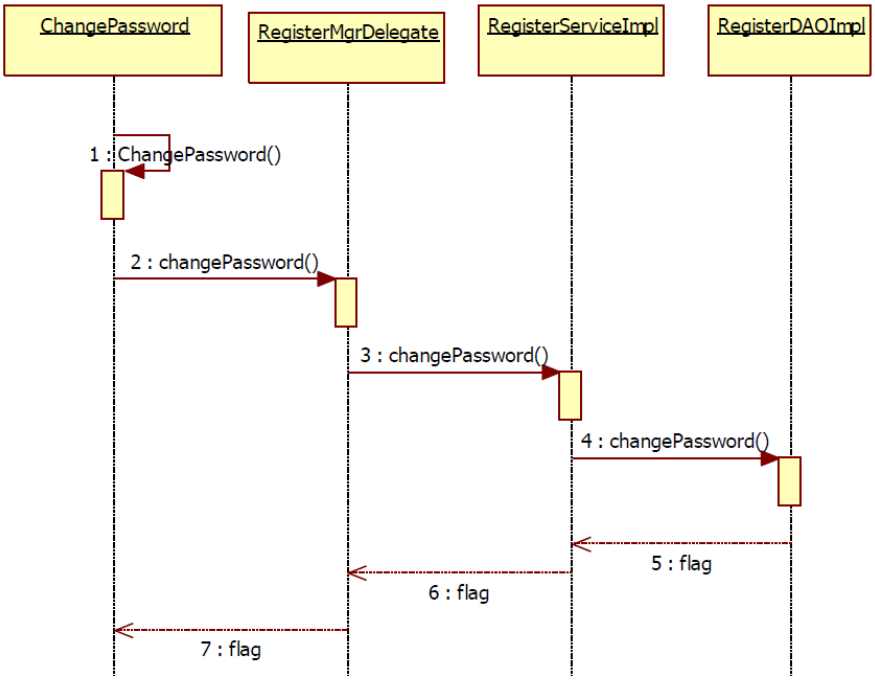
Login Action Sequence



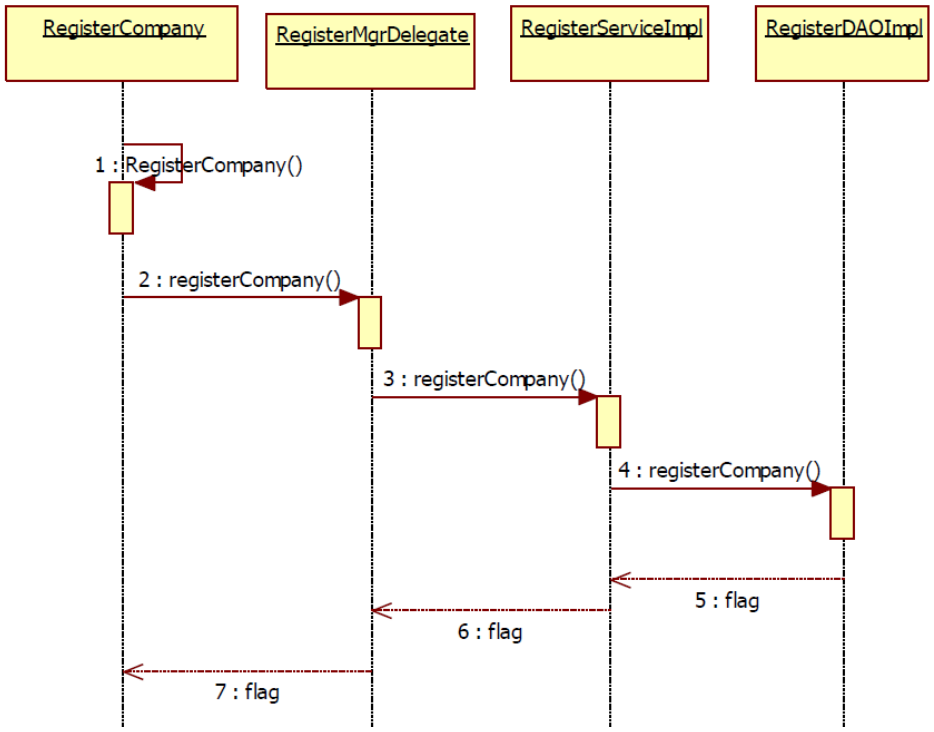
Check User ID Availability Sequence



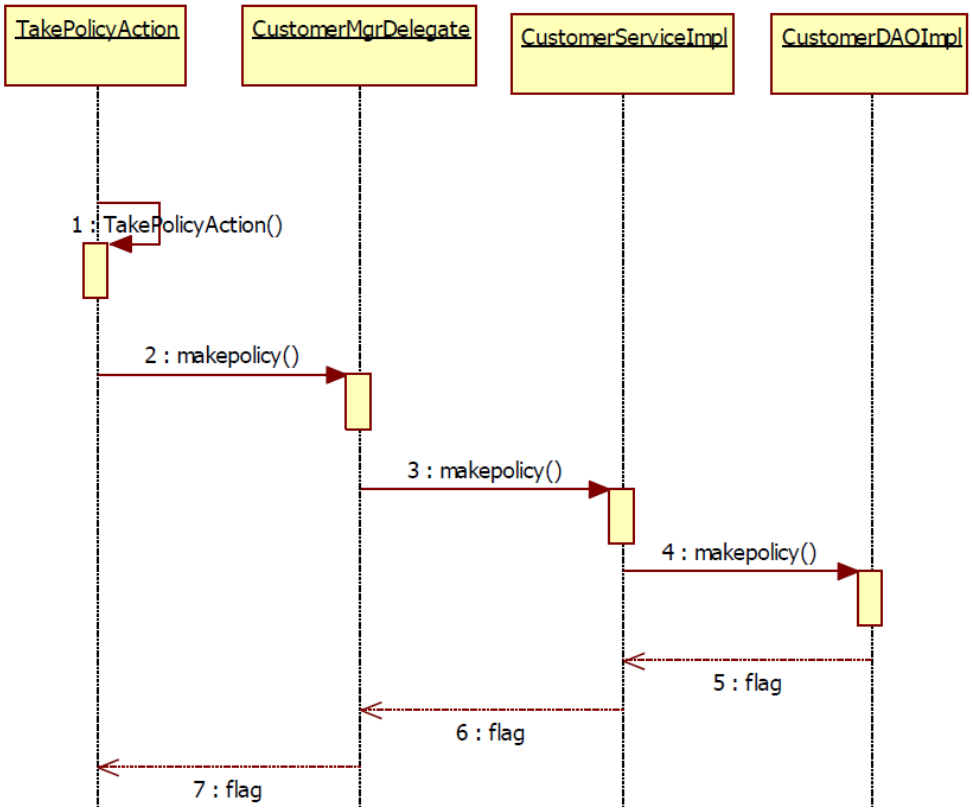
Change Password Sequence



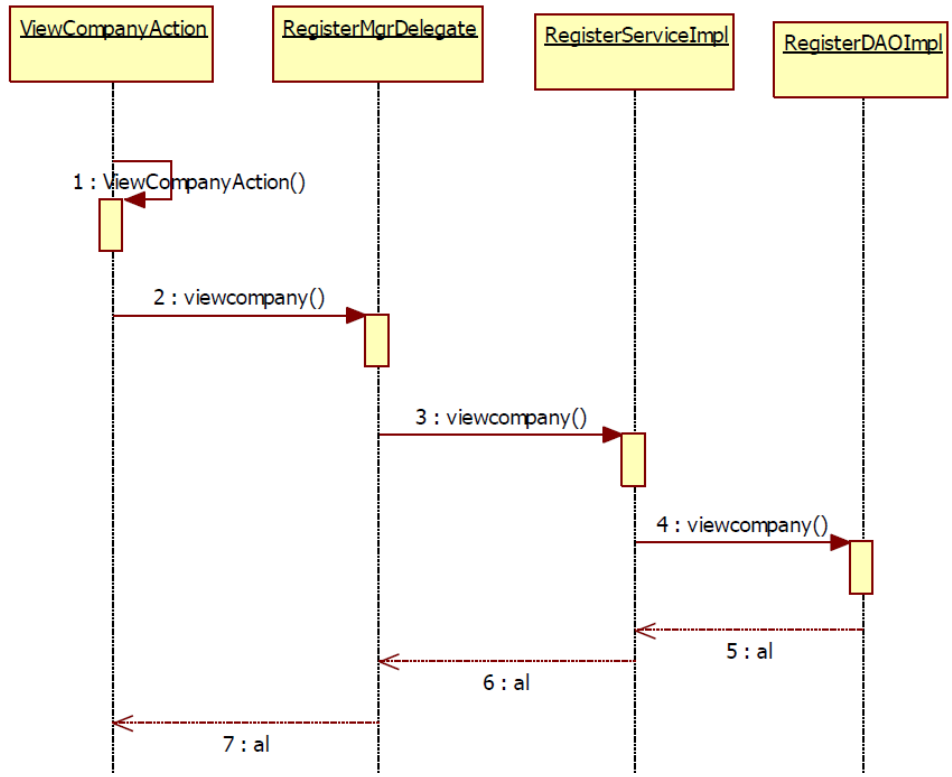
Register Company Sequence



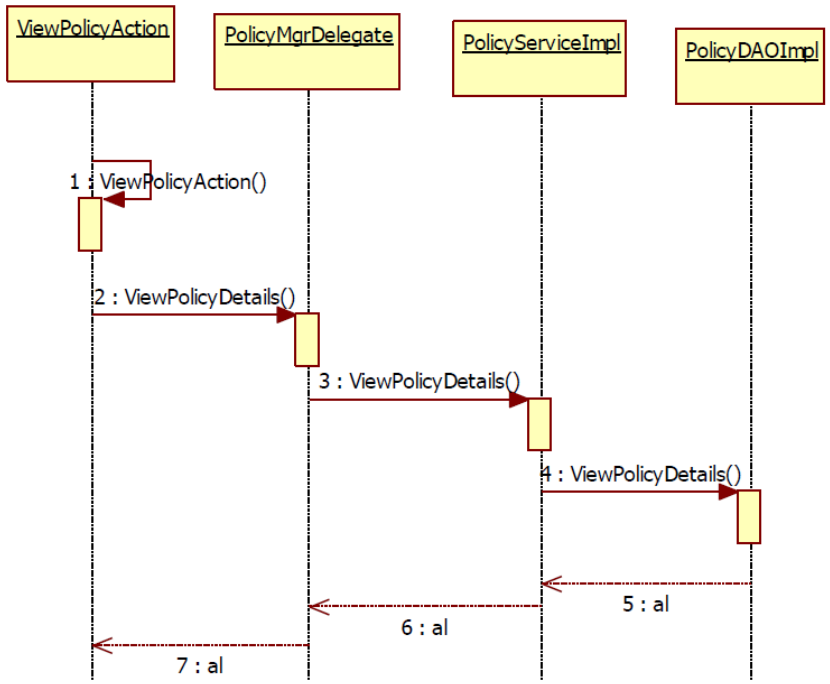
Make Policy Action Sequence



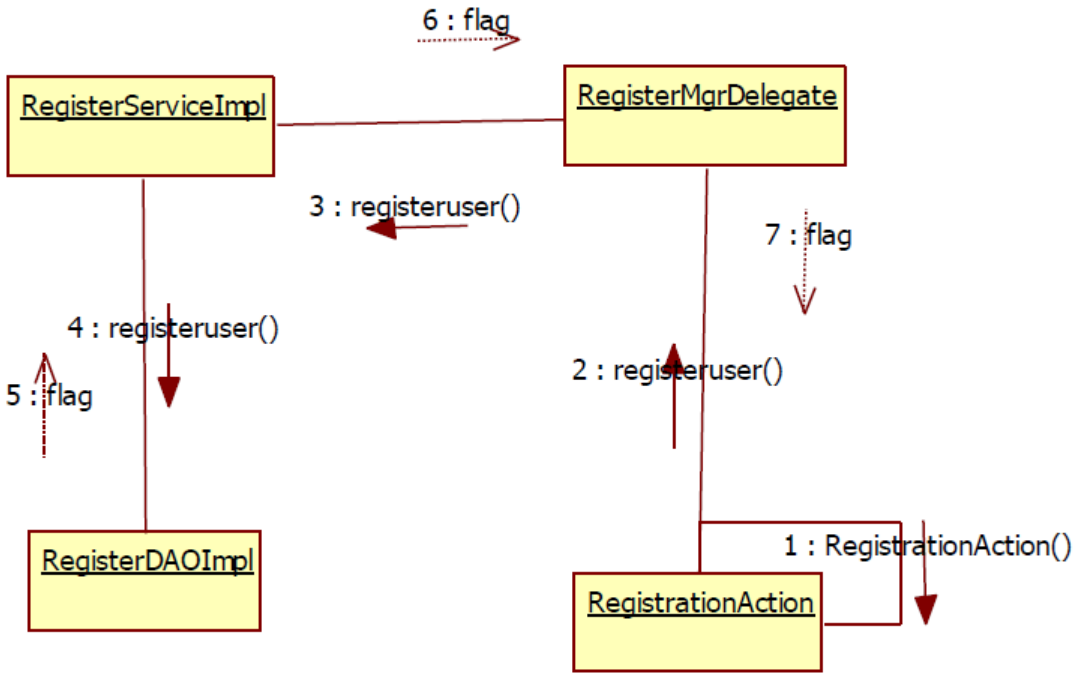
View Company Action Sequence



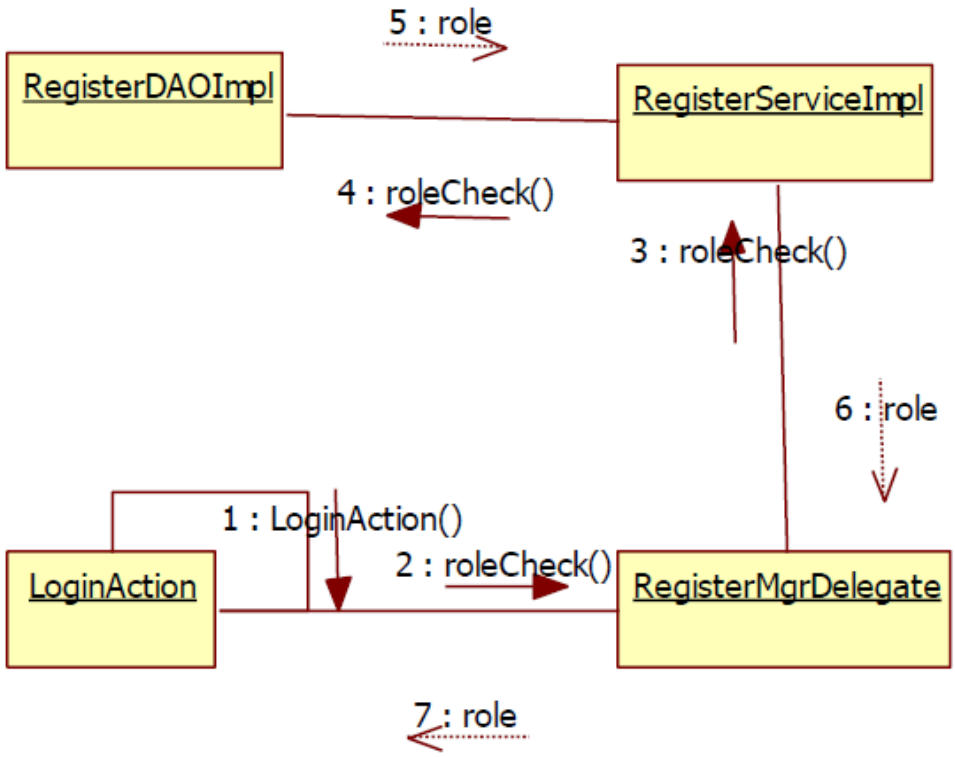
View Policies Action Sequence



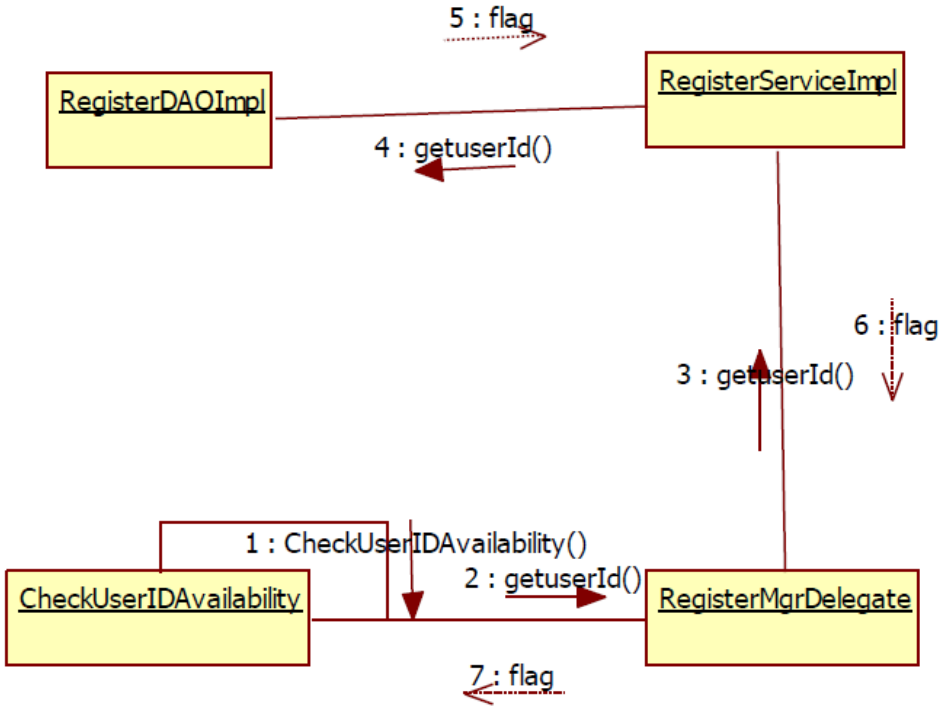
Collaboration Diagrams : Register Action Collaboration



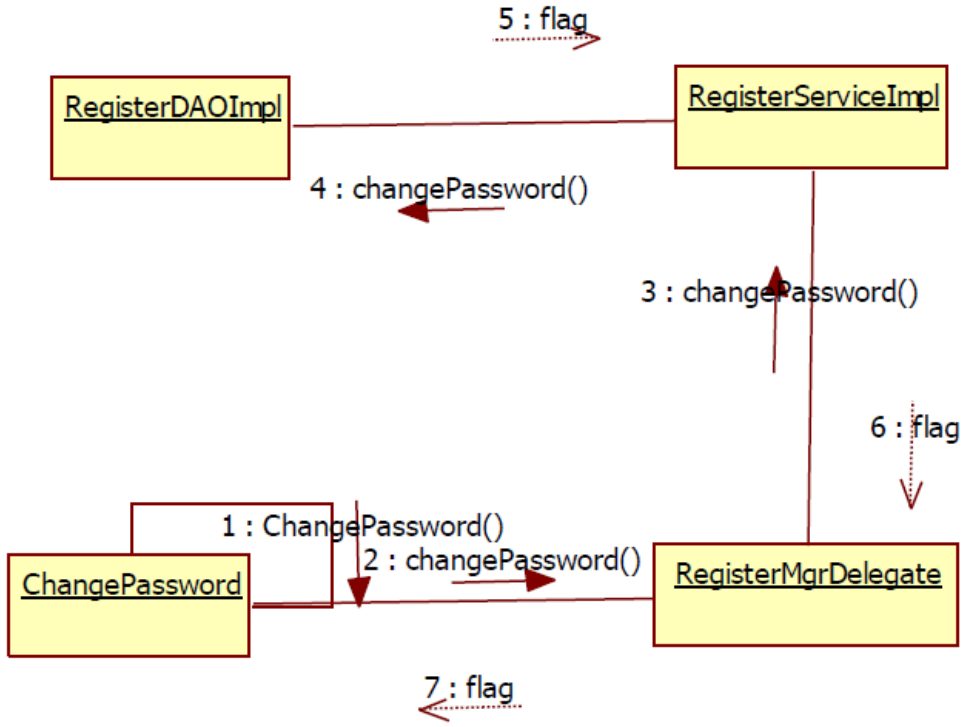
Login Action Collaboration



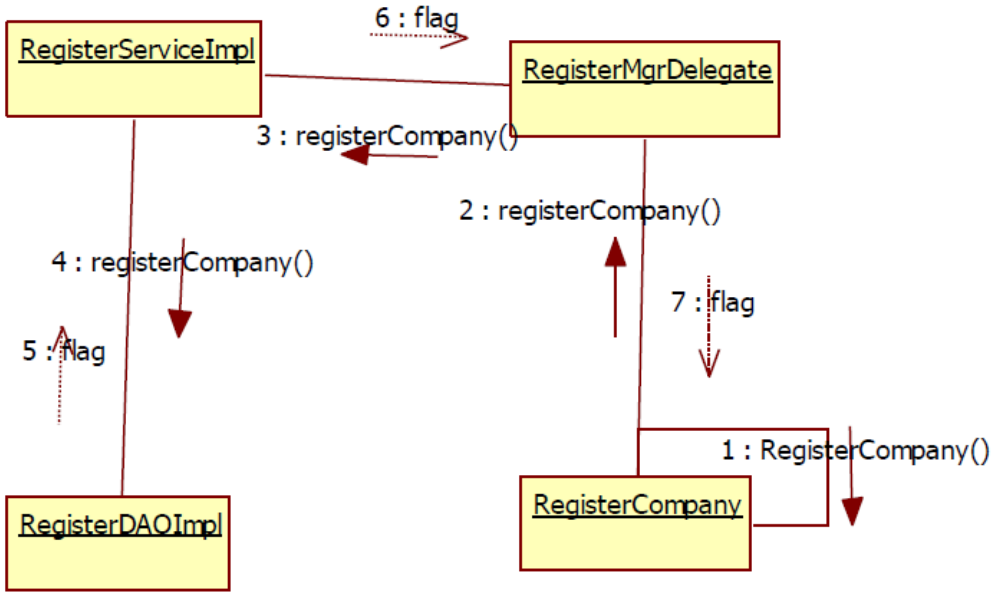
Check User ID Availability Collaboration



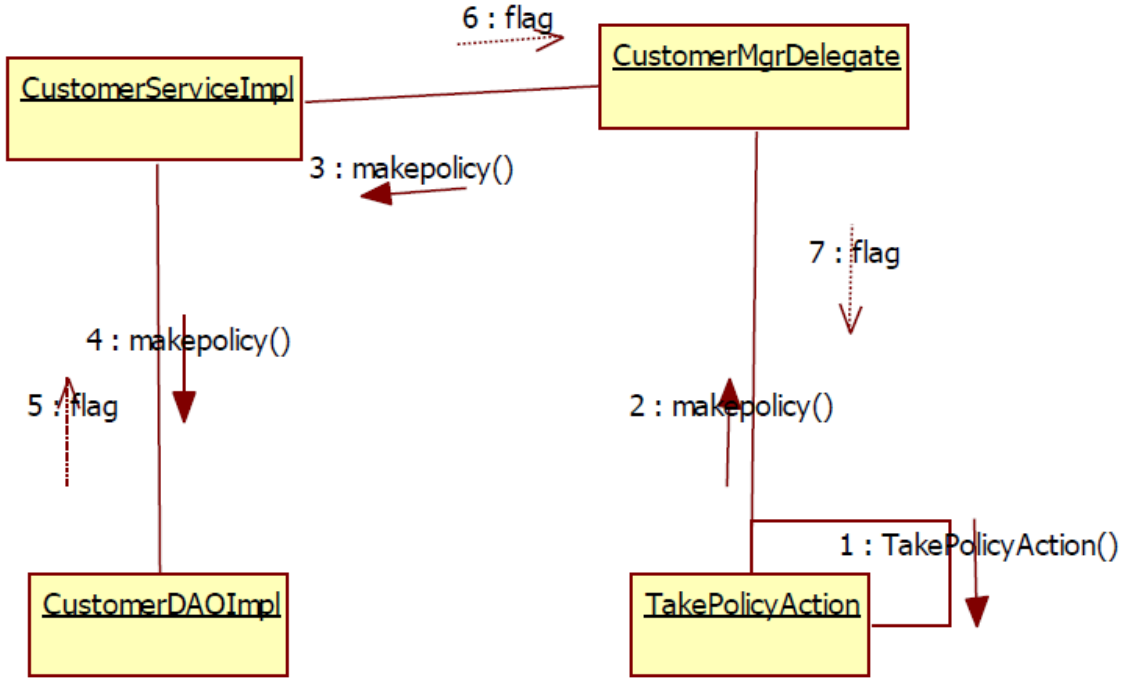
Change Password Collaboration



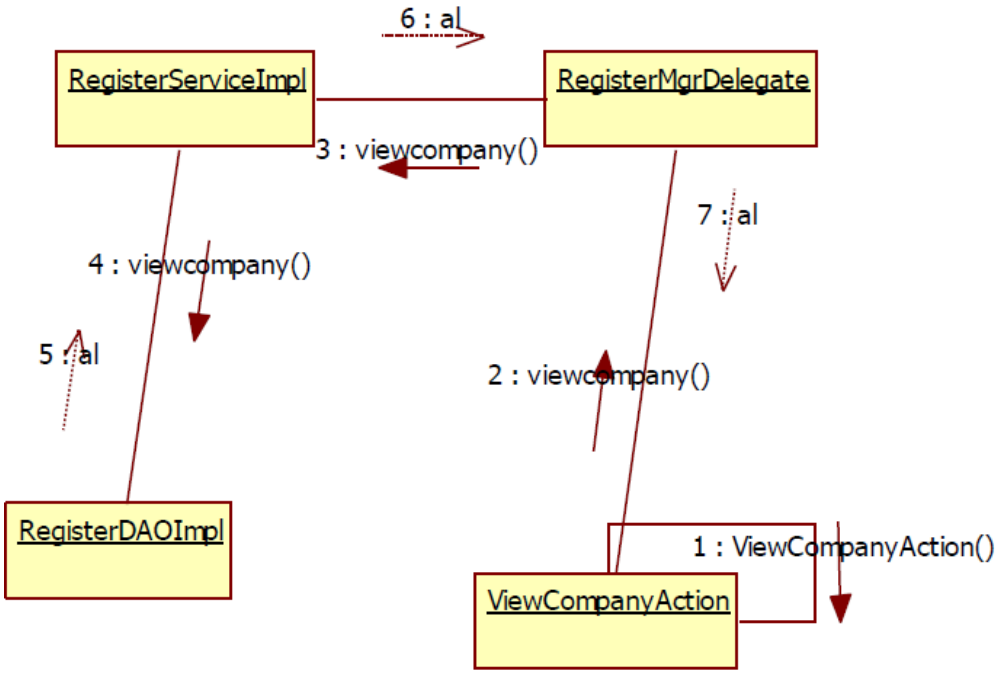
Register Company Collaboration



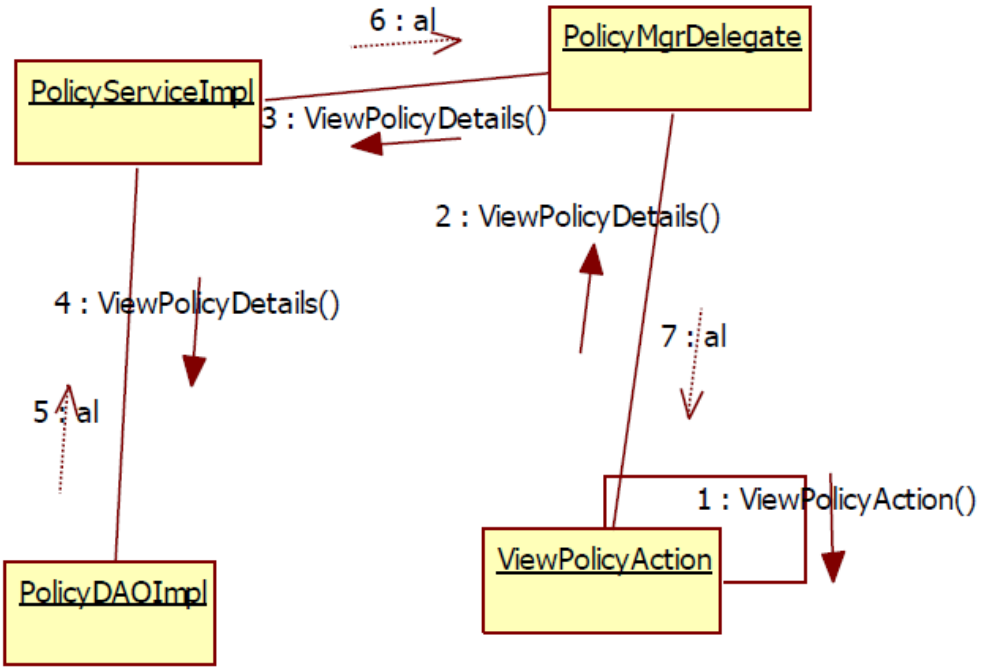
Make Policy Action Collaboration



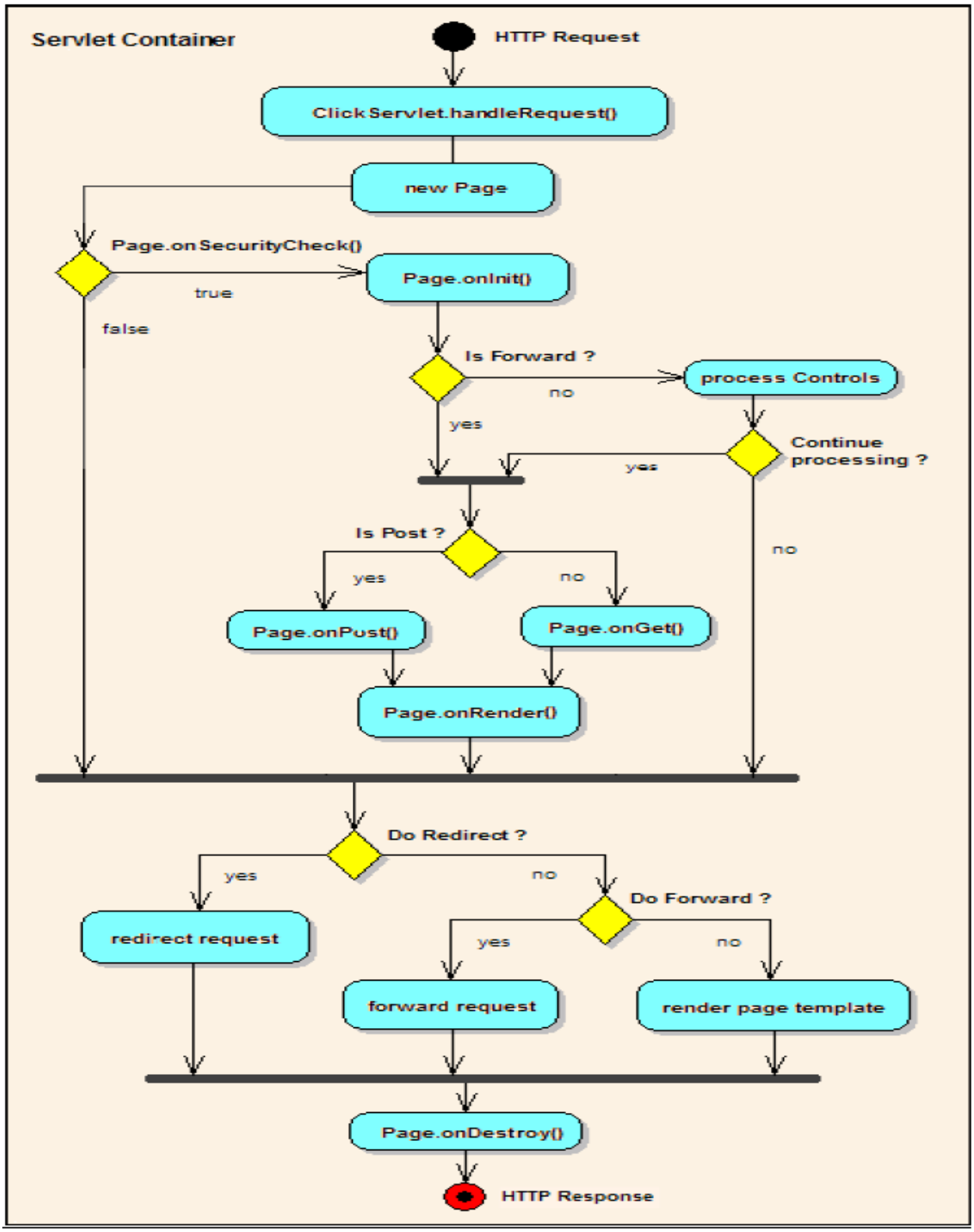
View Company Action Collaboration



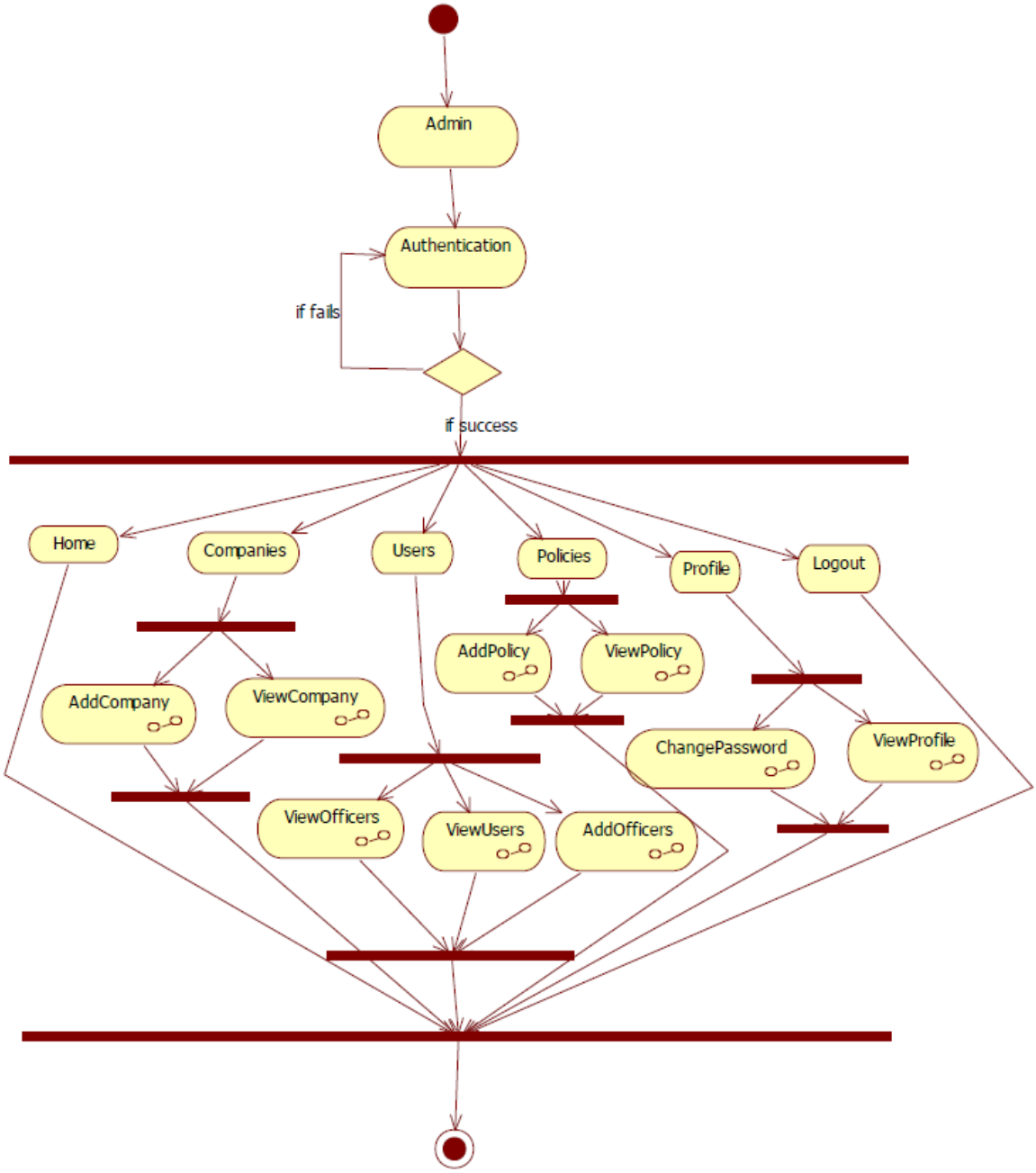
View Policies Action Collaboration



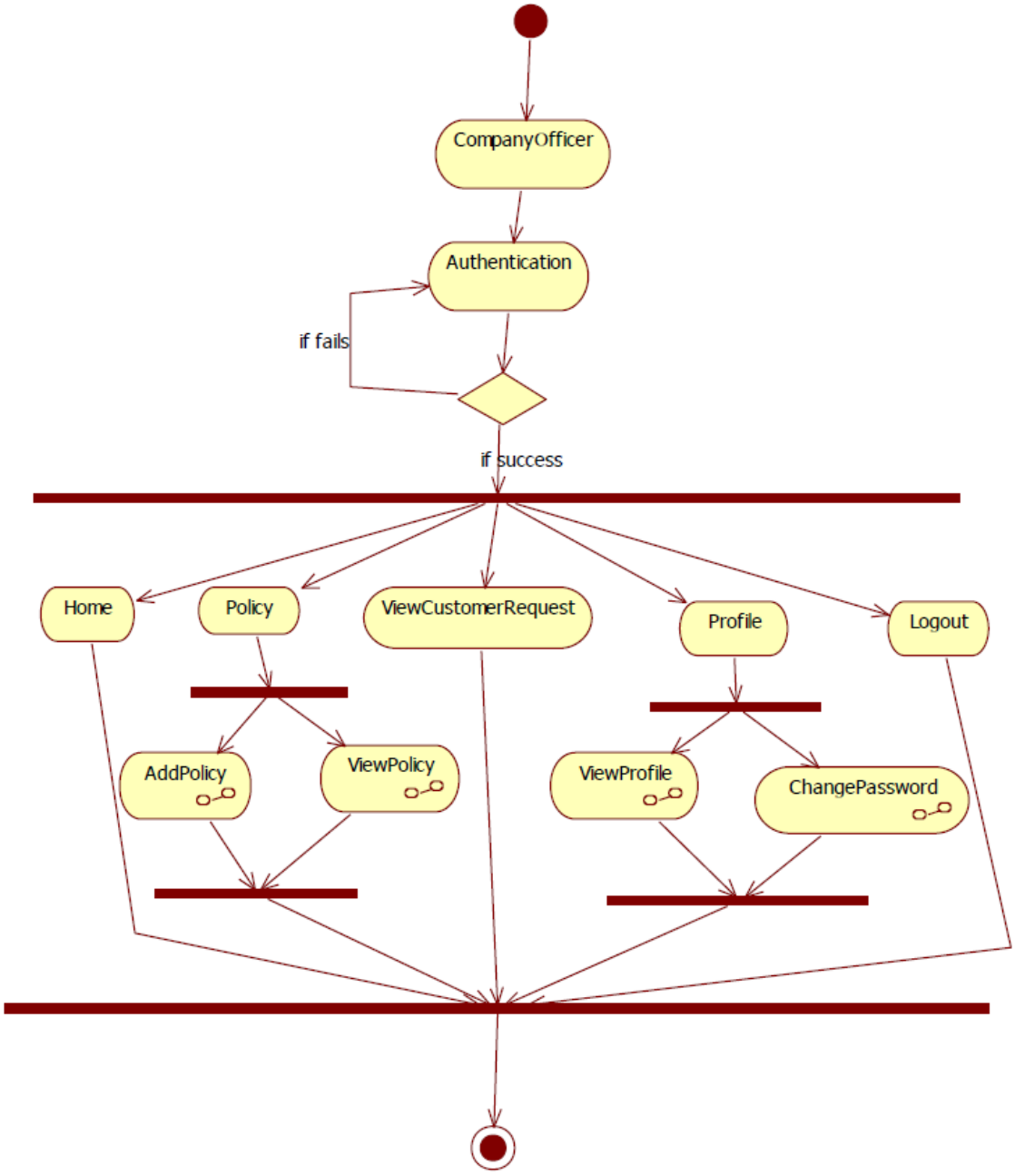
ACTIVITY DIAGRAMS : Servlet Container



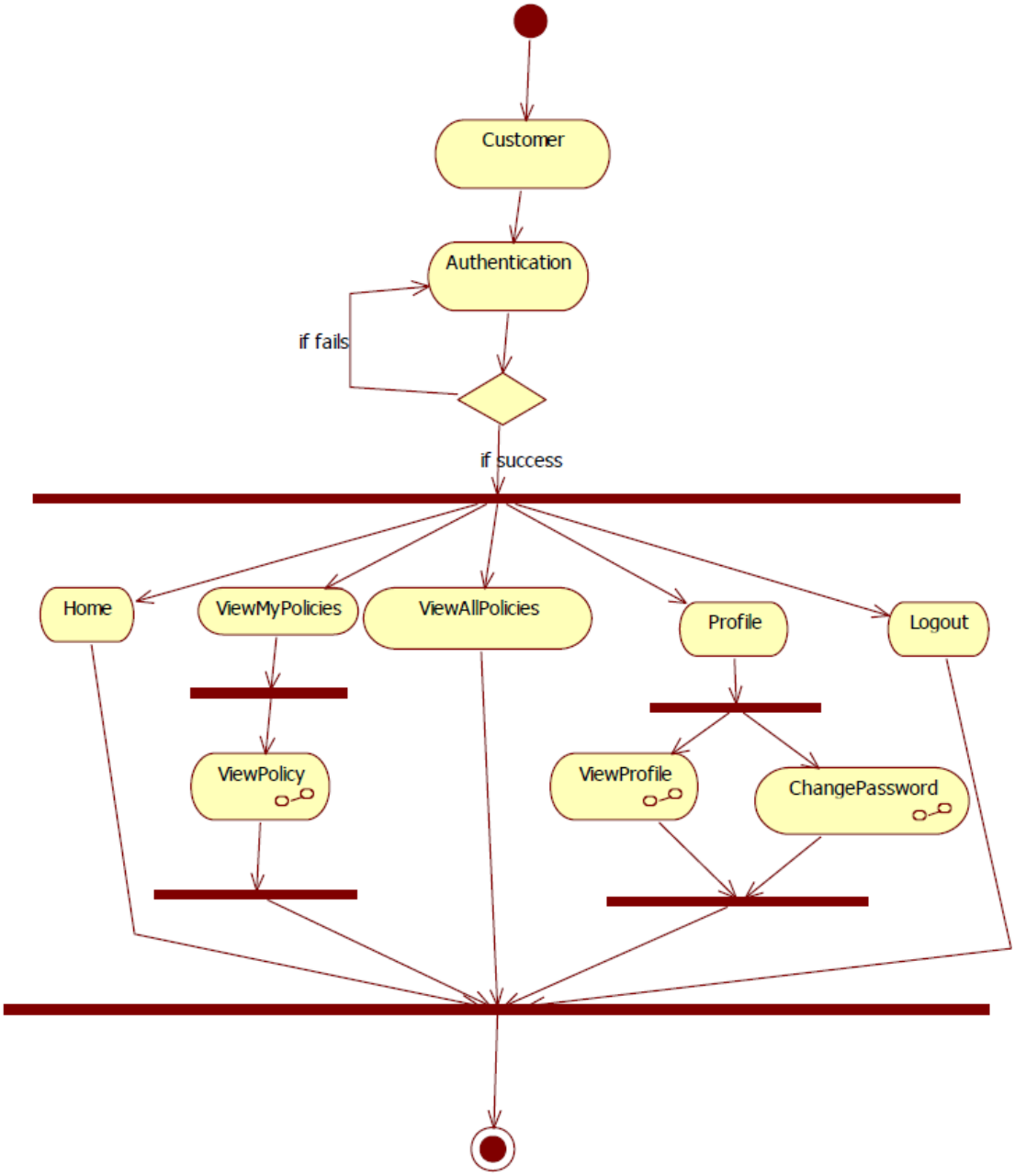
Admin Activity



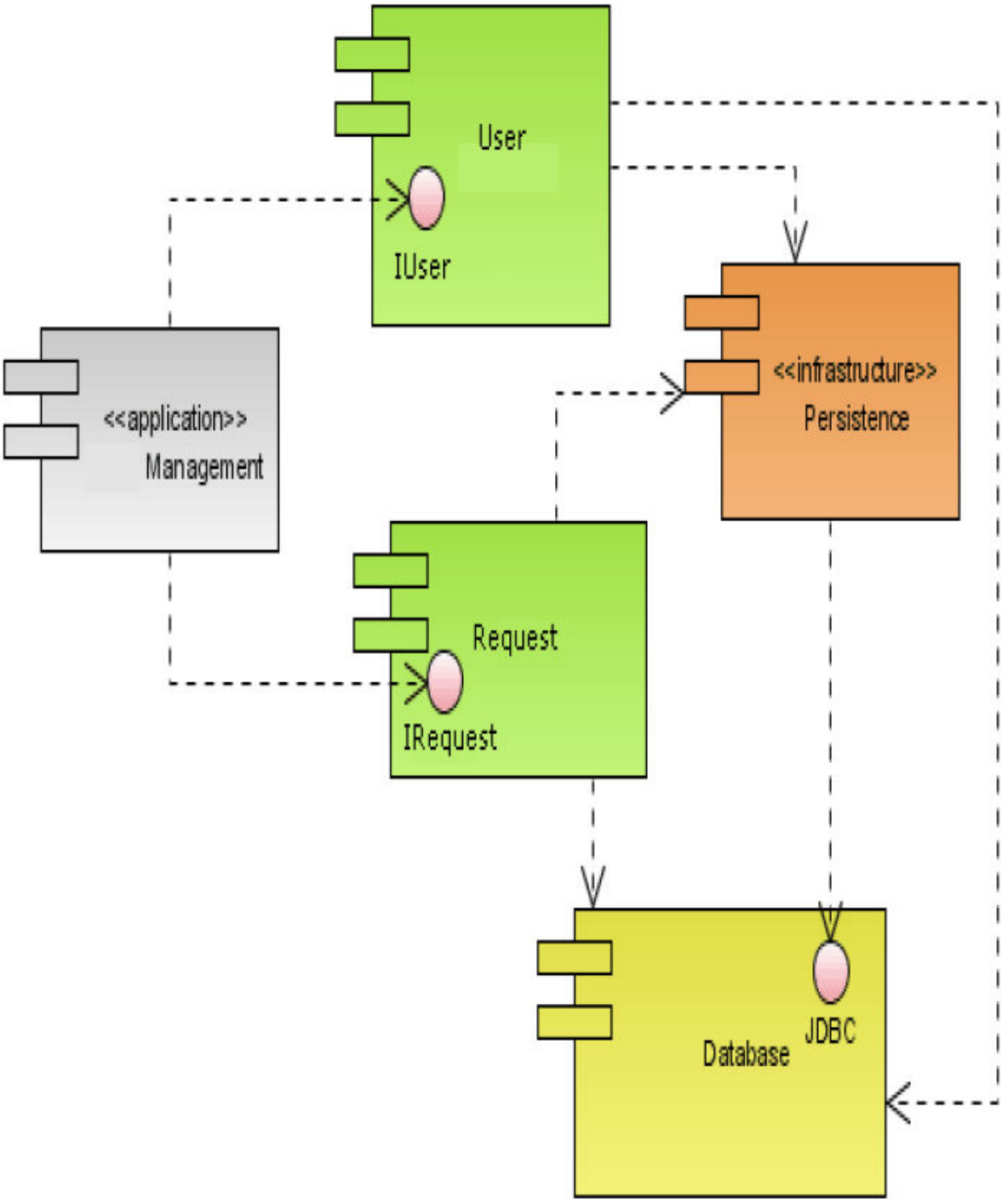
Company Officer Activity



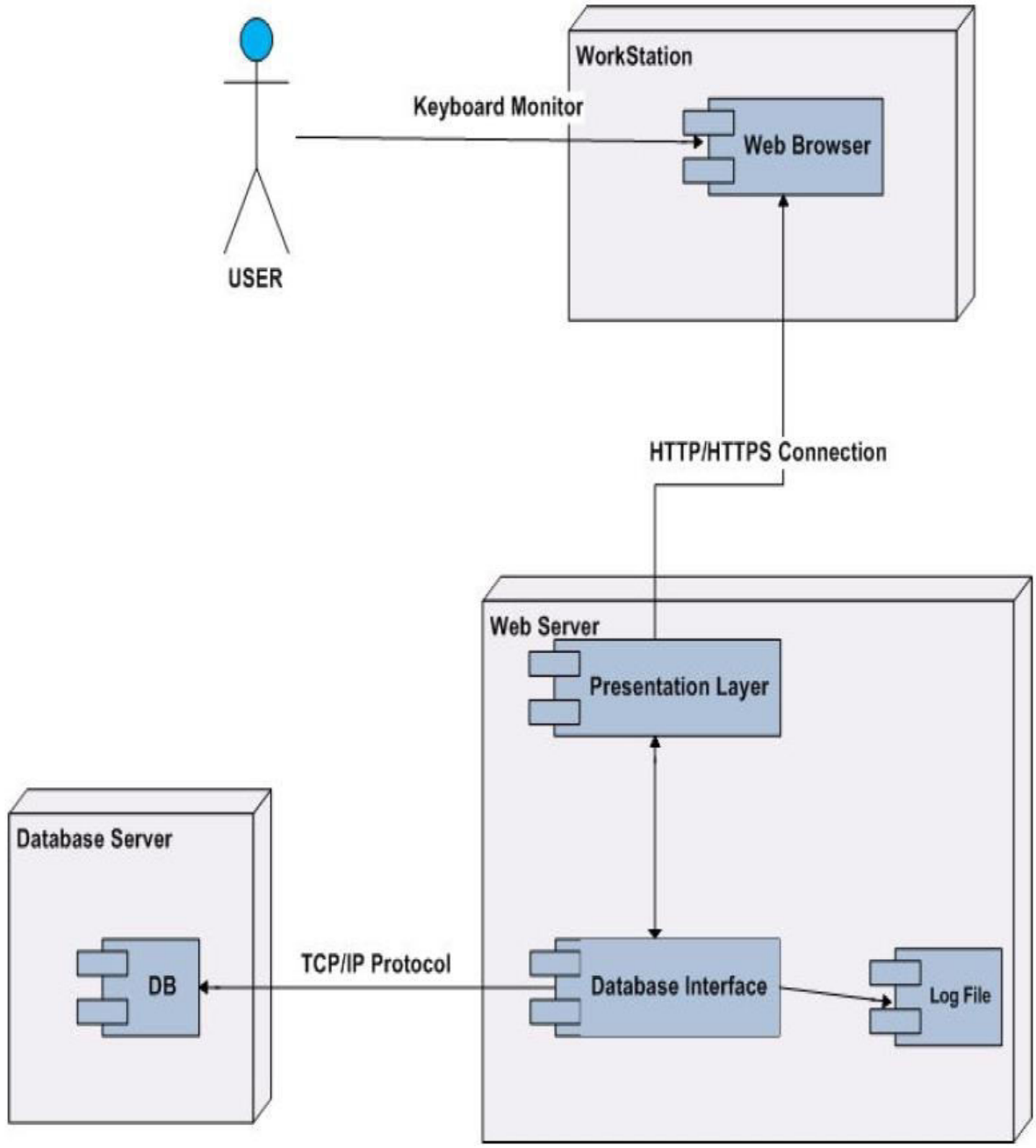
Customer Activity



Component Diagram:



Deployment Diagram:



Process Flow

THE PRESENTATION LAYER: Also called as the client layer comprises of components that are dedicated to presenting the data to the user. For example: Windows/Web Forms and buttons, edit boxes, Text boxes, labels, grids.

THE BUSINESS RULES LAYER: This layer encapsulates the Business rules or the business logic of the encapsulations. To have a separate layer for business logic is of a great advantage. This is because any changes in Business Rules can be easily handled in this layer. As long as the interface between the layers remains the same, any changes to the functionality/processing logic in this layer can be made without impacting the others. A lot of client-server apps failed to implement successfully as changing the business logic was a painful process

THE DATA ACCESS LAYER: This layer comprises of components that help in accessing the Database. If used in the right way, this layer provides a level of abstraction for the database structures. Simply put changes made to the database, tables, etc do not affect the rest of the application because of the Data Access layer. The different application layers send the data requests to this layer and receive the response from this layer.

THE DATABASE LAYER: This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database could be created using SQL Server, Oracle, Flat files, etc. In an n-tier application, the entire application can be implemented in such a way that it is independent of the actual Database. For instance, you could change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected.

Software Requirements:

Technology	: J2SE and J2EE
Web-Technologies	: HTML ,JavaScript ,CSS
Web Server	:Tomcat
Java Version	:JDK
Backend Database	:Oracle
IDE	: My Eclipse

Approvals:

Name	Signature	Date