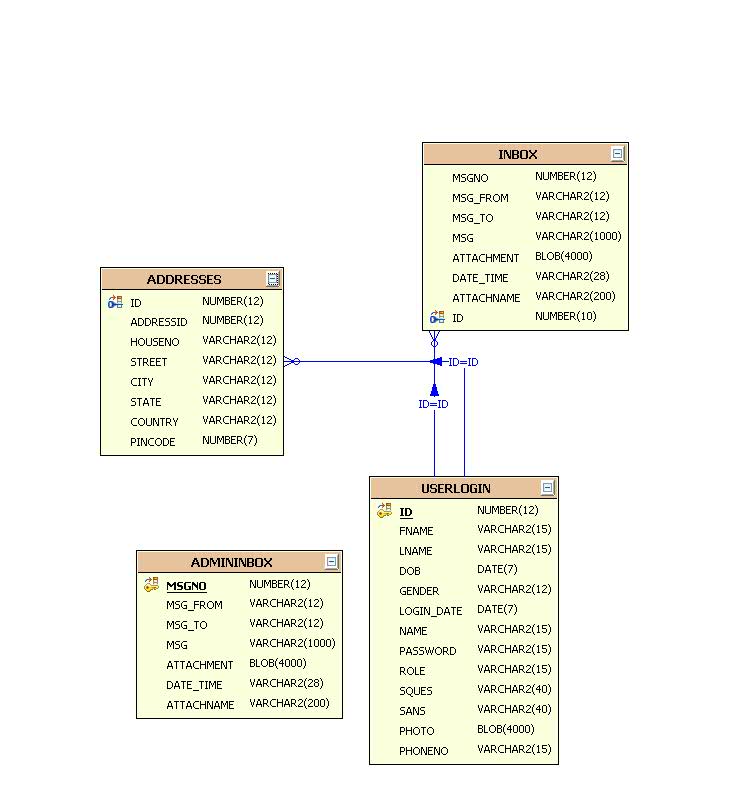
**SYSTEM DESIGN**

**E-R DIAGRAM**

**E - R Diagrams**

****

**UML DIAGRAMS**

**UNIFIED MODELING LANGUAGE DIAGRAMS**

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**

This view represents the system from the users perspective.

The analysis representation describes a usage scenario from the end-users perspective.

**STRUCTURAL MODEL VIEW**

In this model the data and functionality are arrived from inside the system.

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:

UML Analysis modeling, which focuses on the user model and structural model views of the system.

UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

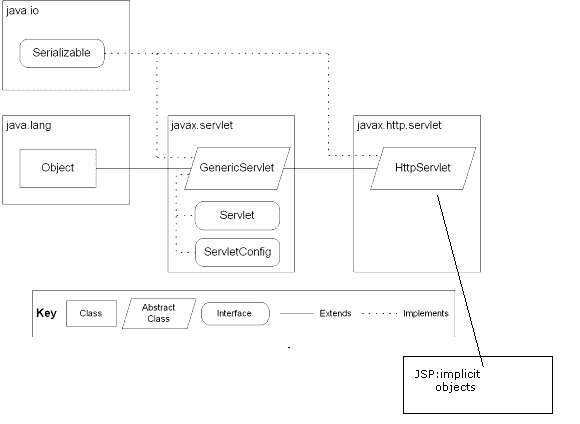
Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

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.

**Class Diagram**

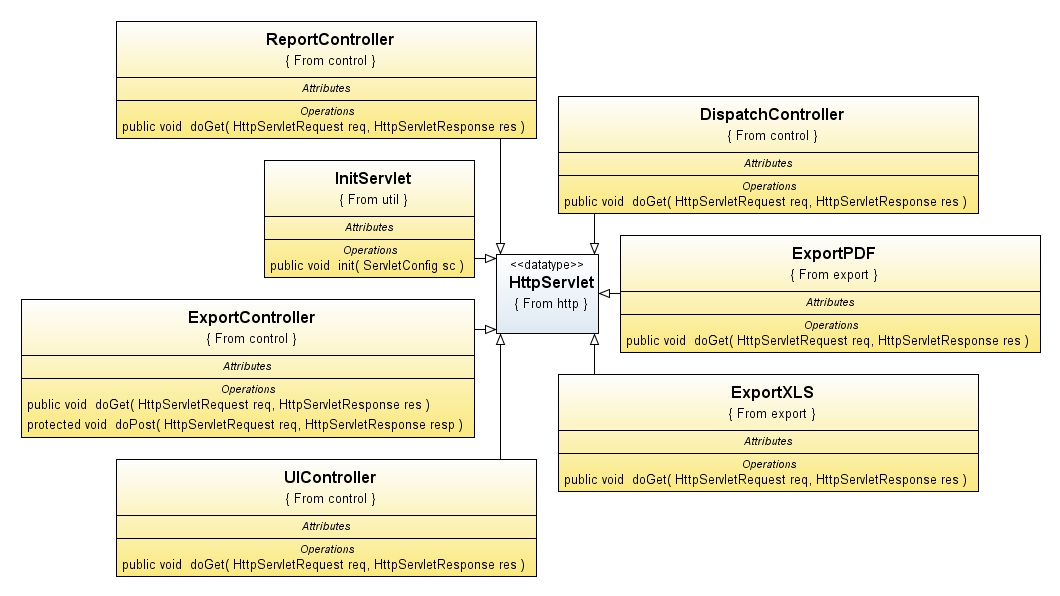
**CLASS DIAGRAM**

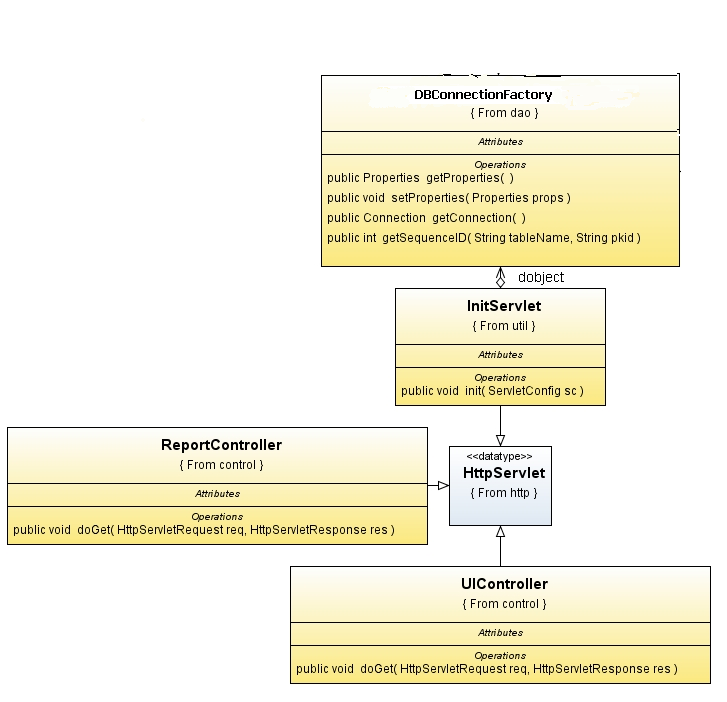
Class diagrams describe the structure of the system in terms of classes and objects. The servlet api class diagram will be as follows.

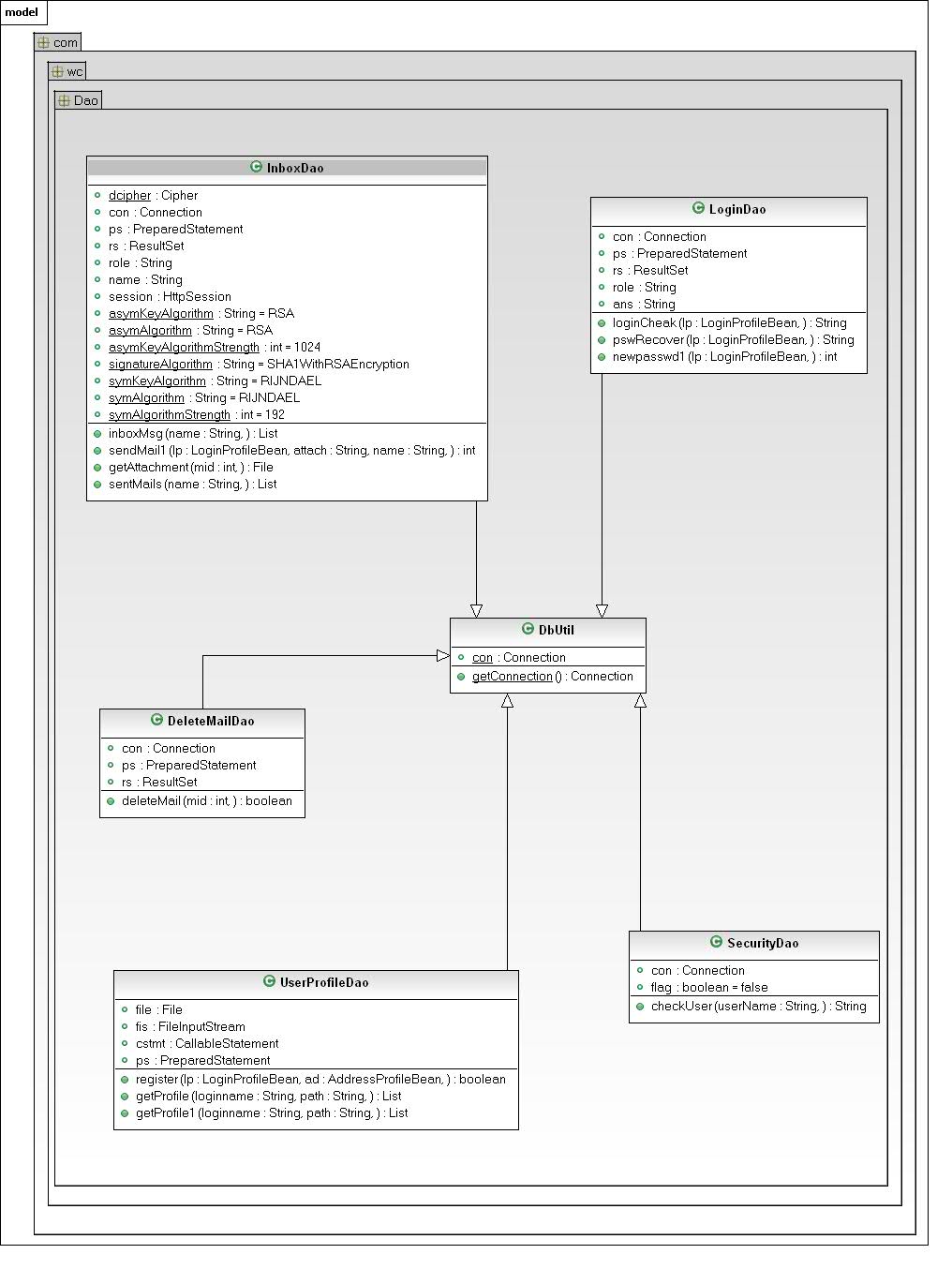


**Class Collaboration Diagrams**

**Class Collaboration Diagram**







**Use Case Diagrams**

**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:

* UML Analysis modeling, this focuses on the user model and structural model views of the system.
* UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

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.

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**

System

**Admin**

**User**

A Cryptology-based Approach for

Web Information Security

1. **Administrator Use Case Diagram**

****

User Use case Diagram



**Sequence Diagrams**

**User-Level Sequence Diagrams**

**1. Administrator Sequence Diagram**



User Sequance Diagram



**Operation-Level Sequence Diagram**

1. **Login Sequence Diagram**

login

Administrator

User

1: login

2: login

3: validate

4: validlogin

5: validlogin

**Login Collaborative Diagram**

Administ

rator

login

User

1: login

2: login

3: validate

4: validlogin

5: validlogin

1. **Present Login User Report Sequence Diagram**

**Admin**

**Login**

**Report**

**DataBase**

**login**

**validate**

**validLogin**

**getPresentUsingReport**

**selectInfo**

**ReportInfo**

**ReportOutput**

**Present Login User Report Collaborative Diagram**

**Login**

**Admin**

**Report**

**Data**

**Base**

**1: login**

**2: validate**

**3: validLogin**

**4: getPresentUsingReport**

**5: selectInfo**

**6: ReportInfo**

**7: ReportOutput**

**Collaboration Diagrams**

**Admin Collaboration diagram**



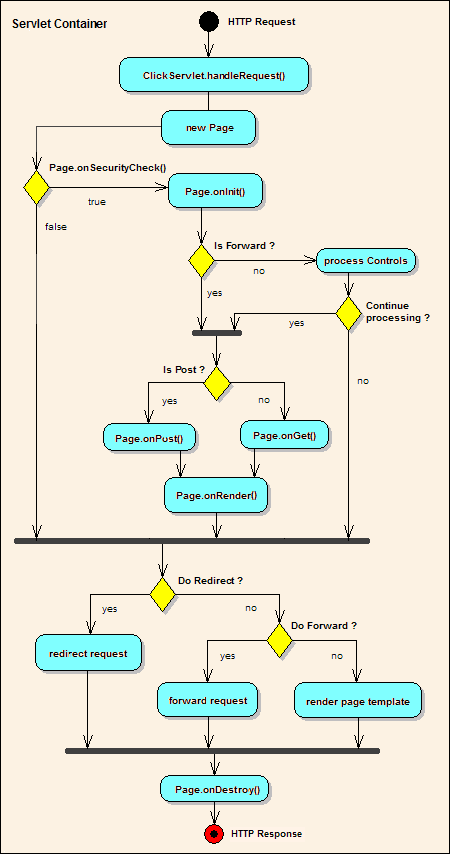
User Collaboration diagram



**ACTIVITY DIAGRAMS**

**ACTIVITY DIAGRAMS**

1. **Servlet Container**

****

##### Administrator Activity Diagram

****

User Activity Diagram



**Component Diagram**

**Component Diagram :**

****

**Deployment Diagram**

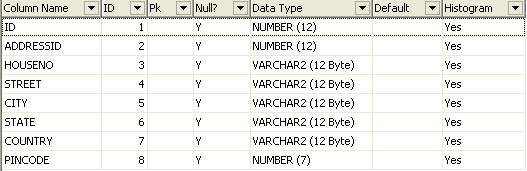
**Deployment Diagram:**

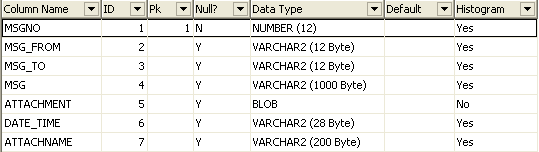
****

**Data Dictionary**

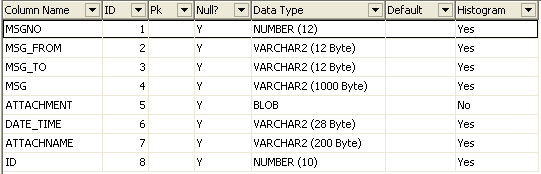
**Data Disctinory**

ADDRESSES



ADMININBOX

INBOX



USERLOGIN

