**SMARTER WORK**

**MANAGEMENT SYATEM**

**INTRODUCTION**

Smarter Work Management System(SWMS) is an automated system that can be useful to Employees(developers) , team leads and managers(Administrator) in any functional organization. SWMS gives the facility to define the projects in the organization and also allows to track the efforts spent by the employees for that particular project related task. A report generation facility is supported in SWMS that allows the team leads and managers to analyze effort spent by employees to an assigned project. This tool can help managers for effort estimation per project.  This tool helps employees to document their efforts to analyze.

**Existing System:**

* This existing system is not providing secure registration and profile management of all the users properly.
* This system is not providing on-line Help.
* This system doesn’t provide any facility to track the effort spent by employees on a particular task.
* This system doesn’t provide any facility to maintain projects and it’s sub modules online.
* This manual system gives us very less security for saving data and some data may be lost due to mismanagement.
* The system is giving only less memory usage for the users.
* The system also not giving help to implement and tune the system.
* The system doesn’t have facility to generate requirement specific report.

**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.

* This system maintains user’s personal, address, and contact details.
* User friendliness is provided in the application with various controls provided by system rich user interface.
* This system makes the overall project management much easier and flexible.
* Various classes have been used for maintain the details of all the users and catalog.
* Authentication is provided for this application only registered users can access.
* Report generation features is provided using to generate different kind of reports.
* The system provides facilities to track effort spent by employees on a particular task.
* The system also provides facilities to manage different projects and it’s sub modules online efficiently.
* The system also provides facilities to manage project assignment to Team Leads by manager and sub module assignment to developers by Team leads.
* This system has also facility to manage different developer teams and their assignments to team leads.
* This system is providing more memory for the users to maintain data.
* This system is providing accessibility control to data with respect to users.
* This system also maintains the user hierarchy.

**FEASIBILITY REPORT**

**FEASIBILITY REPORT**

**TECHNICAL FEASIBILITY:**

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical

analysis.

1. **Understand the different technologies involved in the proposed system:**

Before commencing the project, we have to be very clear about what are the technologies that are to be required for the development of the new system.

1. **Find out whether the organization currently possesses the required technologies:**
   * Is the required technology available with the organization?
   * If so is the capacity sufficient?

For instance –

“Will the current printer be able to handle the new reports and forms required for the new system?”

**OPERATIONAL FEASIBILITY:**

Proposed projects are beneficial only if they can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

* Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
* Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.
* Have the user been involved in the planning and development of the project?
* Early involvement reduces the chances of resistance to the system and in
* General and increases the likelihood of successful project.

Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

**ECONOMIC FEASIBILITY:**

Economic feasibility attempts 2 weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

**SYSTEM REQUIREMENT SPECIFICATION**

**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 strategies.

**NUMBER OF MODULES**

The system after careful analysis has been identified to be presented with the following modules:

The Modules involved are

1. Authentication and Security Module
2. Admin(Project Manager) Module
3. Team Lead Module.
4. Employee (developers) Module.
5. Query and feedback Module.
6. Track Effort Module.
7. Reports Module.

**Modules Description:**

1. **Authentication and Security 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.

1. **Admin Module**

This module consists of the following sub modules:

1. **Registration Module**

Here admin register new employees ie teamleads, developers and store their details in Database. He also add new project details like project code, project name, number of modules etc and store their details in database.

1. **Users Management Module**

Here admin manages different type of users. He can add/view the employee details and also can delete them if requirement comes. He also assigns team members to team lead and can view the team details of the team lead.

1. **Project(Task) Management Module**

Here admin can add a new project and can add, view, modify and delete them according to requirement. Here admin also can assign these projects to team leads.

1. **Team Lead Module**

This module has following sub modules:

1. **Project-Module Management Module:**

Here team leads will let know about current working project details and then he creates sub modules of an assigned project and manages those modules according to the requirement.

1. **Task Management Module:**

One’s the modules are decided by the team lead then he divide those modules into possible no of subtasks. Here team leads manages all the sub tasks of the assigned Project by assigning them to the team members of the team for which he is the team lead.

1. **Employee Module**

Employees can update assigned task status then it will be visible

To the team leads and higher authorities. Employees can view current project information as well as current assigned tasks. Employees send queries to team leads and view replies. Employees can view their profile and can change password also.

1. **Query And Feedback Module:**

Using this module the system maintains a query and feedback sub system in between users. An user can make a query to his superiors regarding any matter And an user can reply the queries made to him.

1. **Track Effort Module:**

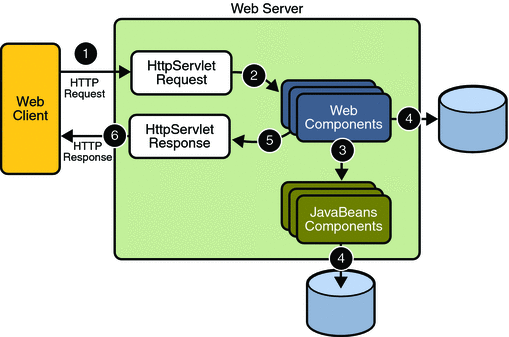
In this module tracking of effort is done automatically with respect to a project (assigned task) . An employee (developer) set his effort spends on a assigned task in a time unit. This effort can be tracked by the hierarchy of seniors with respect to a task(project).

1. **Reports Module:**

In this Module the User and Administrator can generate the different types of Reports according to their access.

**PROCESS FLOW**

**ARCHITECTURE DIAGRAM**



1. **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, etc.

1. **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

1. **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.

1. **THE DATABASE LAYER**

This layer comprises of the Database Components such as DB Files, Tables, and 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.

**SDLC METHODOLOGIES**

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

* The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
* A preliminary design is created for the new system.
* A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
* A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning an designing the second prototype.
4. Constructing and testing the second prototype.

* At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.
* The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
* The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
* The final system is constructed, based on the refined prototype.
* The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

**The following diagram shows how a spiral model acts like:**



**Fig 1.0-Spiral Model**

**ADVANTAGES**

* Estimates(i.e. budget, schedule SWMSc .) become more relistic as work progresses, because important issues discoved earlier.
* It is more able to cope with the changes that are software development generally entails.
* Software engineers can get their hands in and start woring on the core of a project earlier.

**SOFTWARE REQUIREMENT AND**

**HARDWARE REQUIREMENT**

**SOFTWARE REQUIREMENTS**

Operating System : Windows XP/2003 or Linux

User Interface : HTML, CSS

Client-side Scripting : JavaScript

Programming Language : Java

Web Applications : JDBC, Servlet, JSP

IDE/Workbench : My Eclipse 6.0

Database : Oracle 10g

Server Deployment : Tomcat 5.x

**HARDWARE REQUIREMENTS**

Processor : Pentium IV

Hard Disk : 40GB

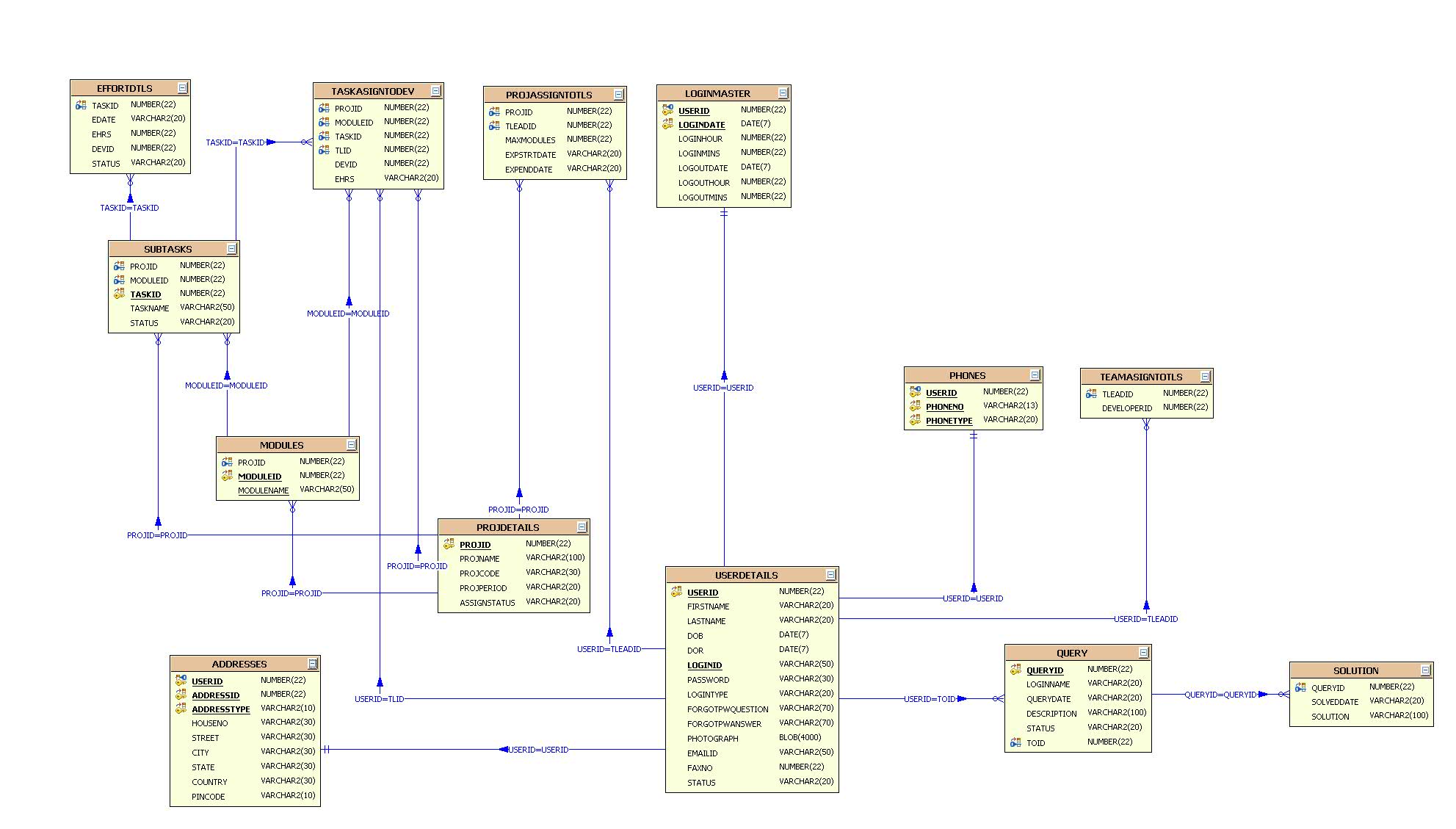
RAM : 1GB or more

**SYSTEM DESIGN**

**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.

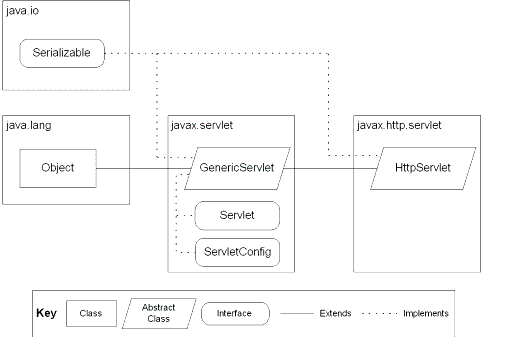
Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

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**

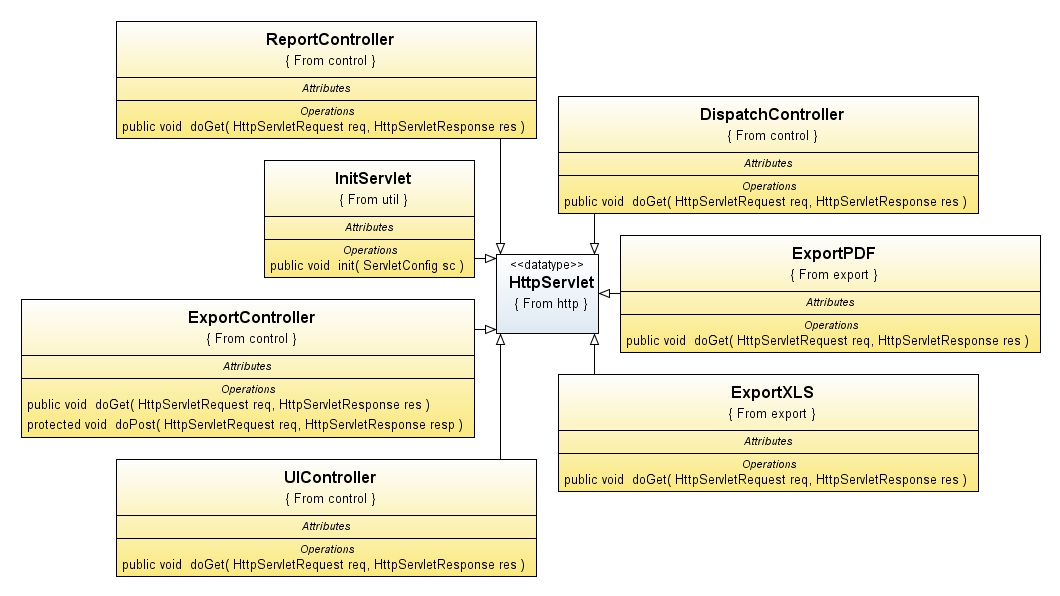
Class diagrams describe the structure of the system in terms of classes and objects. The servlet api class diagram will be as follows.

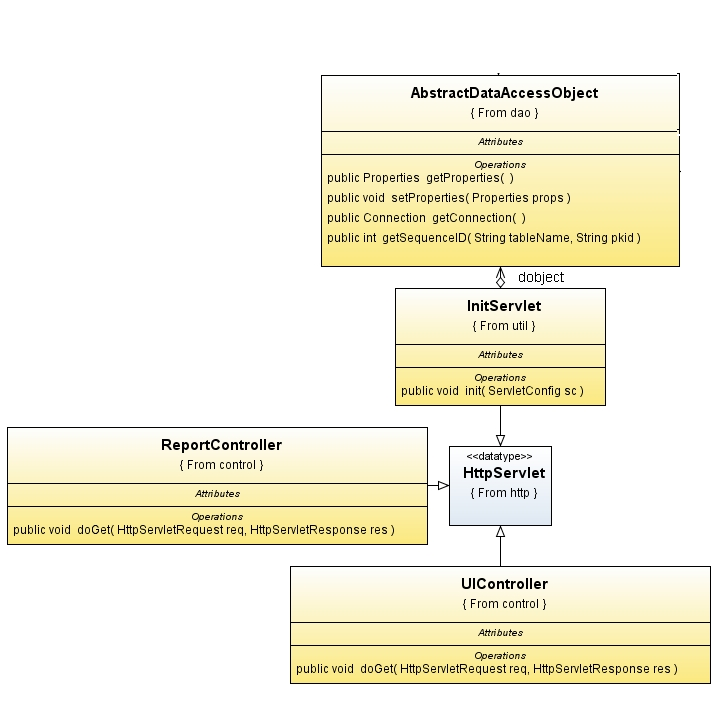


JSP: Implicit Objects

**Class Collaboration Diagrams**

**Class Collaboration Diagram**



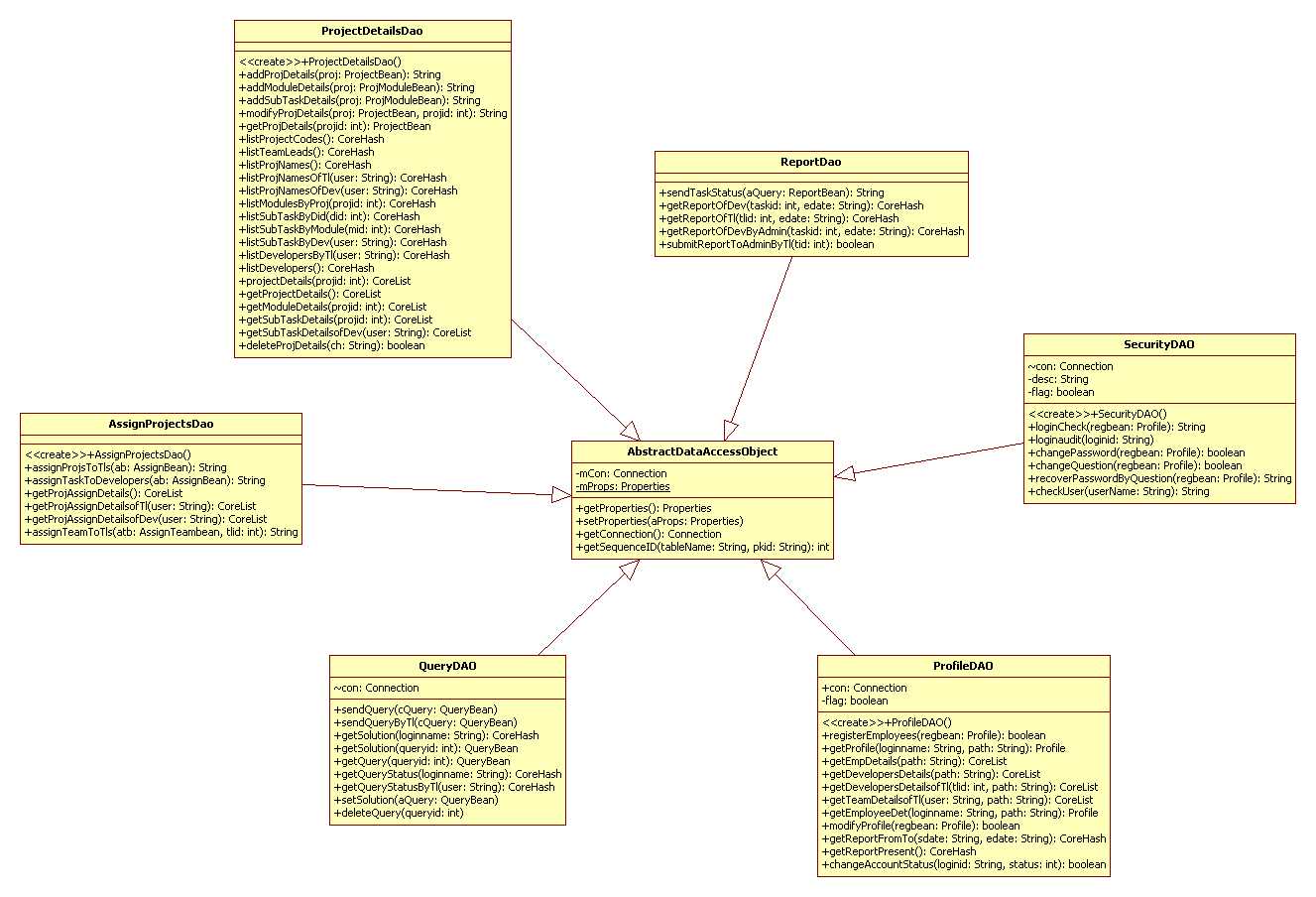












**Use Case Diagrams**

**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

**SmarterWorkManagement**

**System**

**Admin**

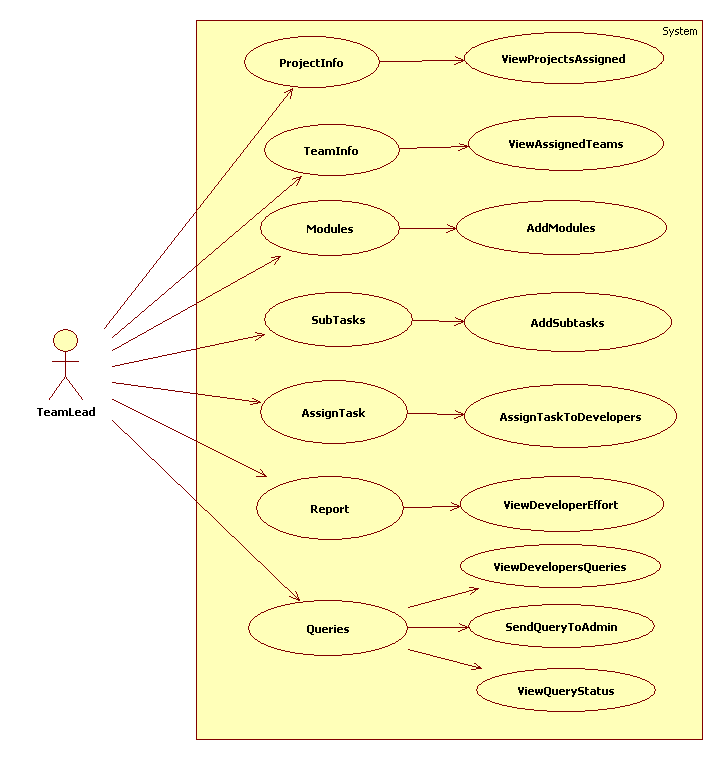
**TeamLead**

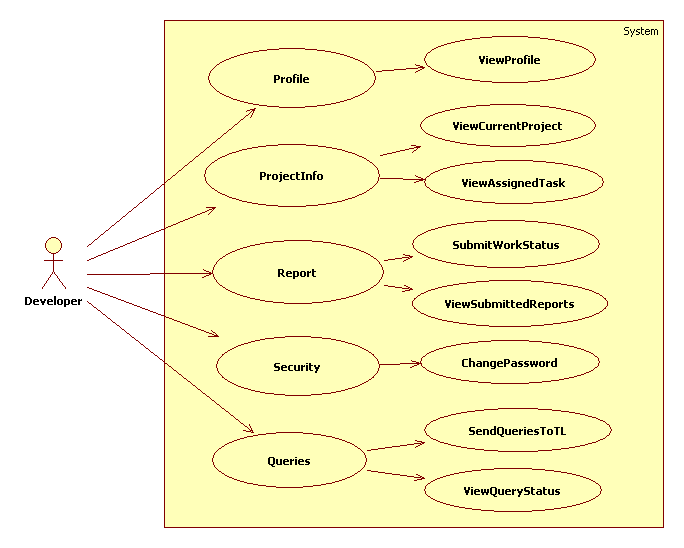
**Developer**

1. **Administrator Use Case Diagram**



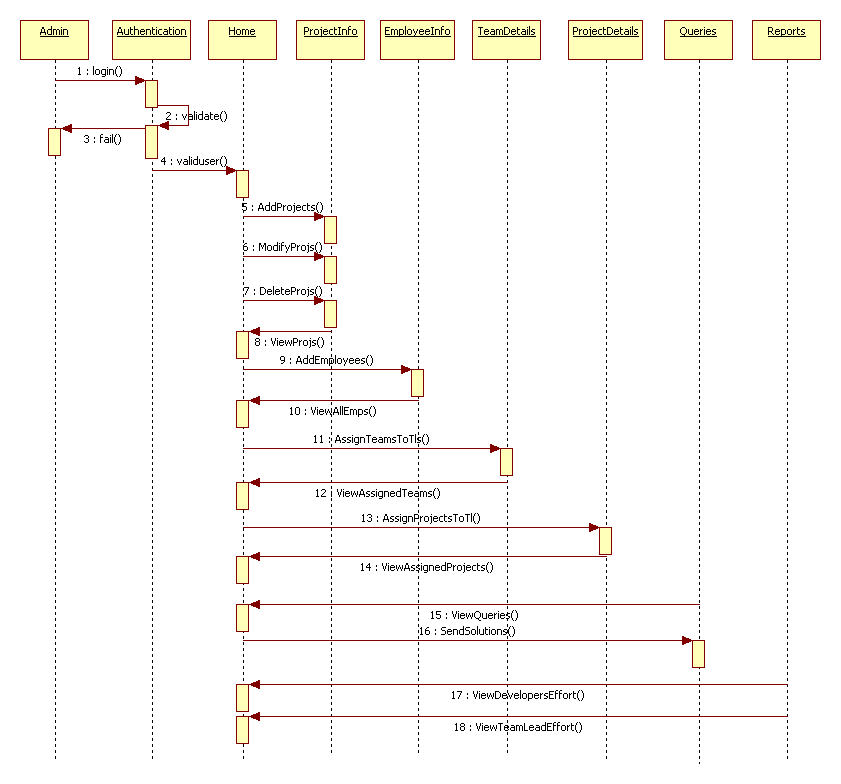
1. **Teamlead Use Case Diagram**



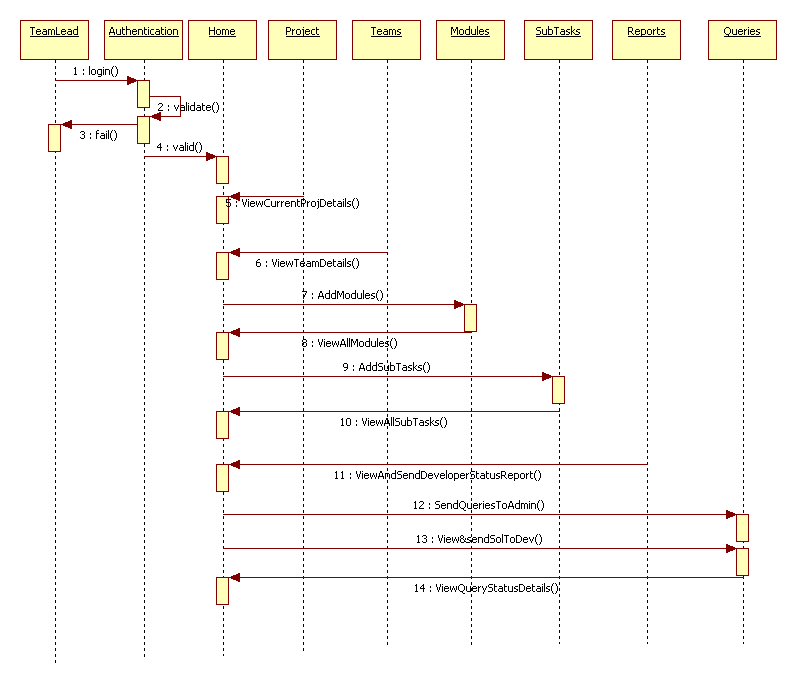
1. **Developer Use Case Diagram**
2. 

**Sequence Diagrams**

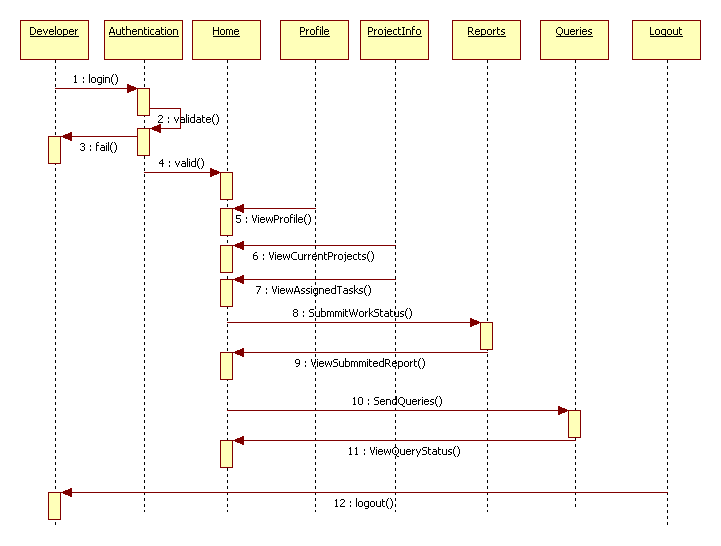
1. **Administrator Sequence Diagram**



1. **Teamlead Sequence Diagram**



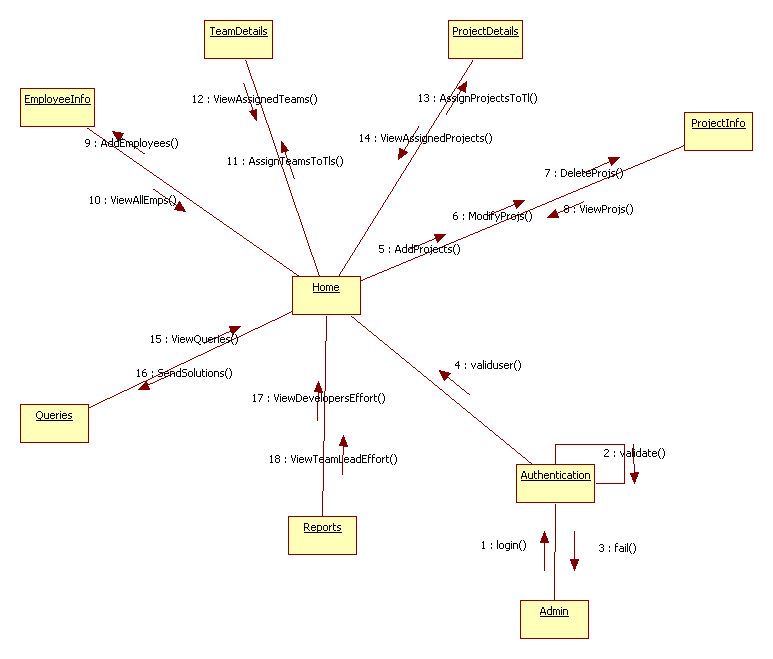
1. **Developer Sequence Diagram**



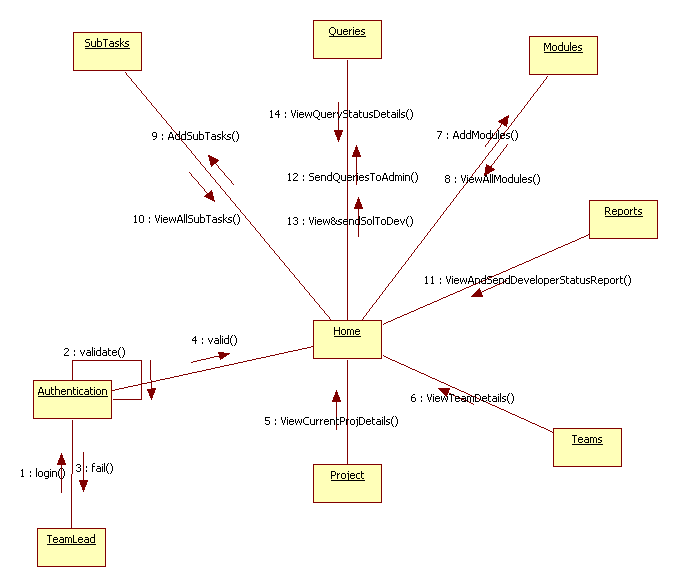
**Collaboration Diagrams**

\

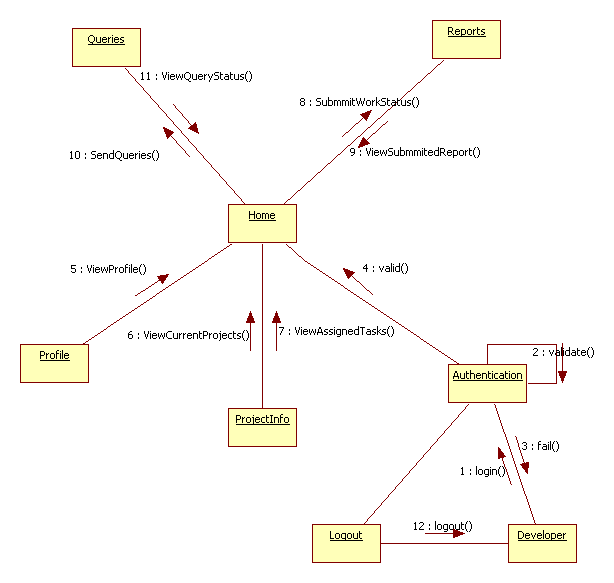
1. **Admin Level CollobrativeDiagram**



1. **Teamlead Level CollobrativeDiagram**



1. **Developer Level CollobrativeDiagram**



**Operation-Level Sequence Diagram**

1. **Login Sequence Diagram**

****

**Login Collaborative Diagram**

****

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

****

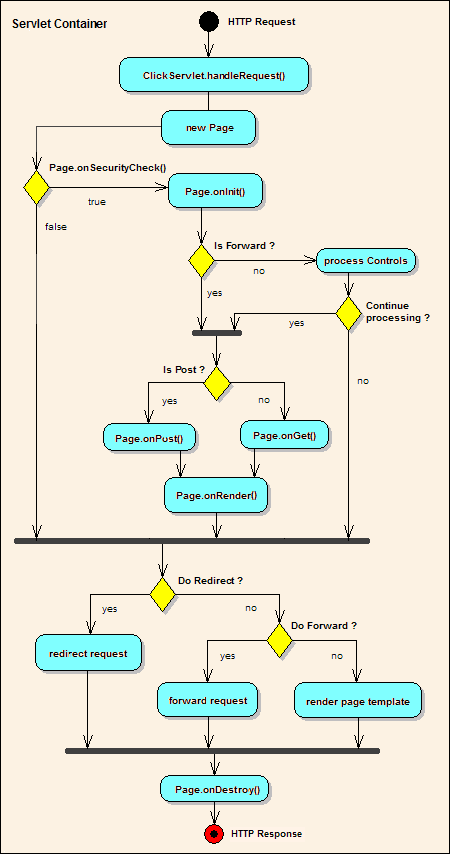
**Present Login User Report Collaborative Diagram**

****

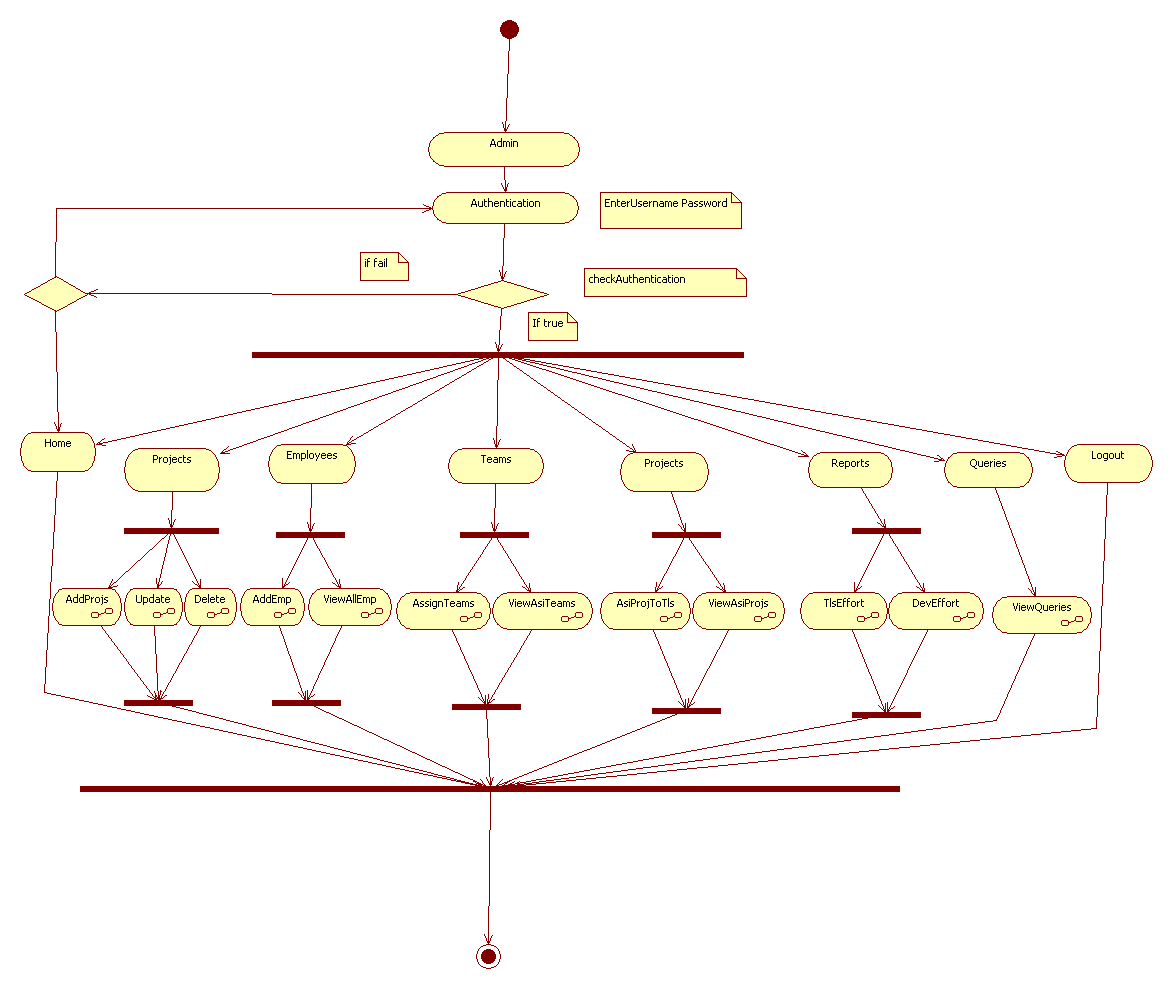
**ACTIVITY DIAGRAMS**

**ACTIVITY DIAGRAMS**

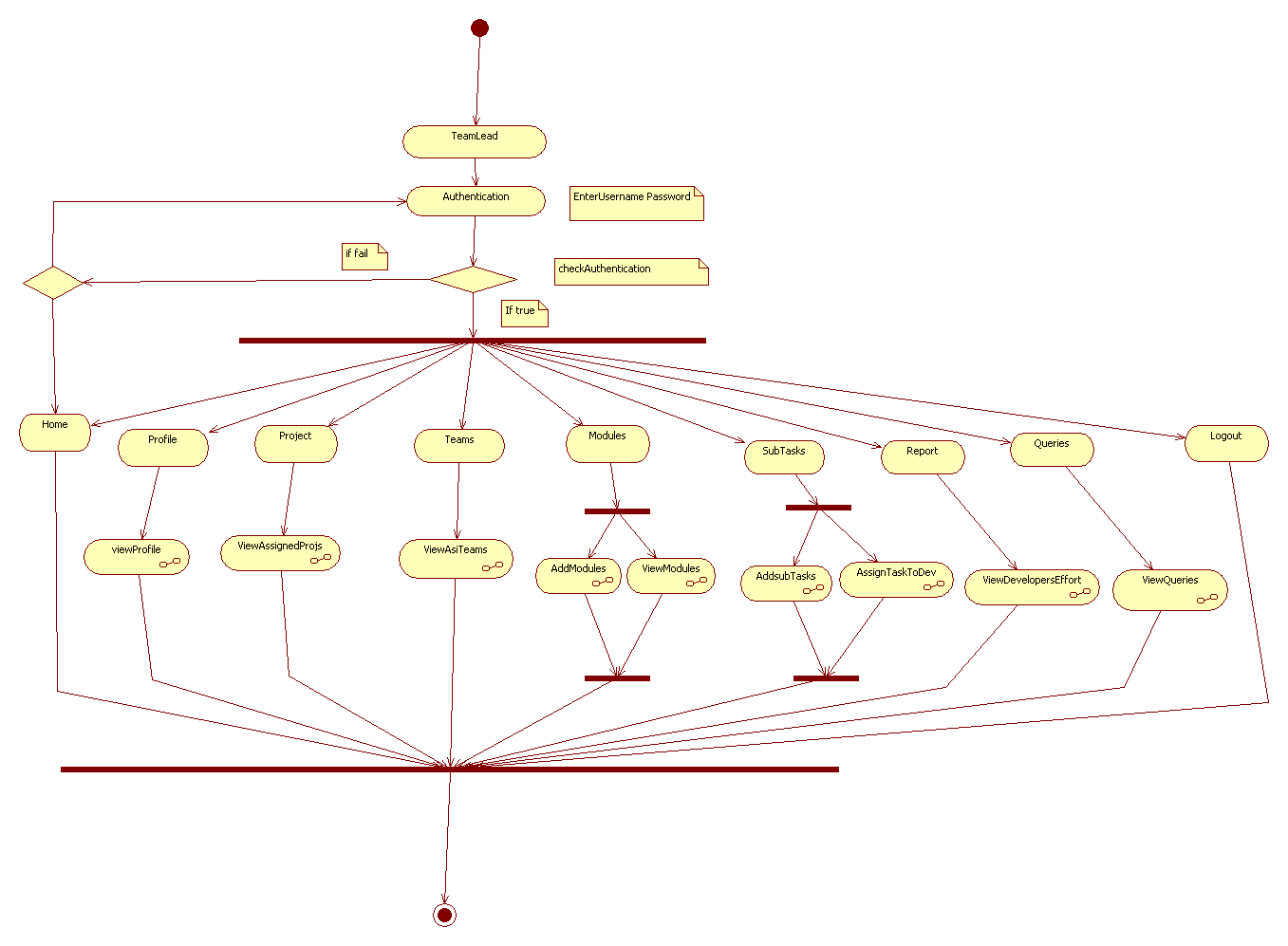
1. **Servlet Container**

****

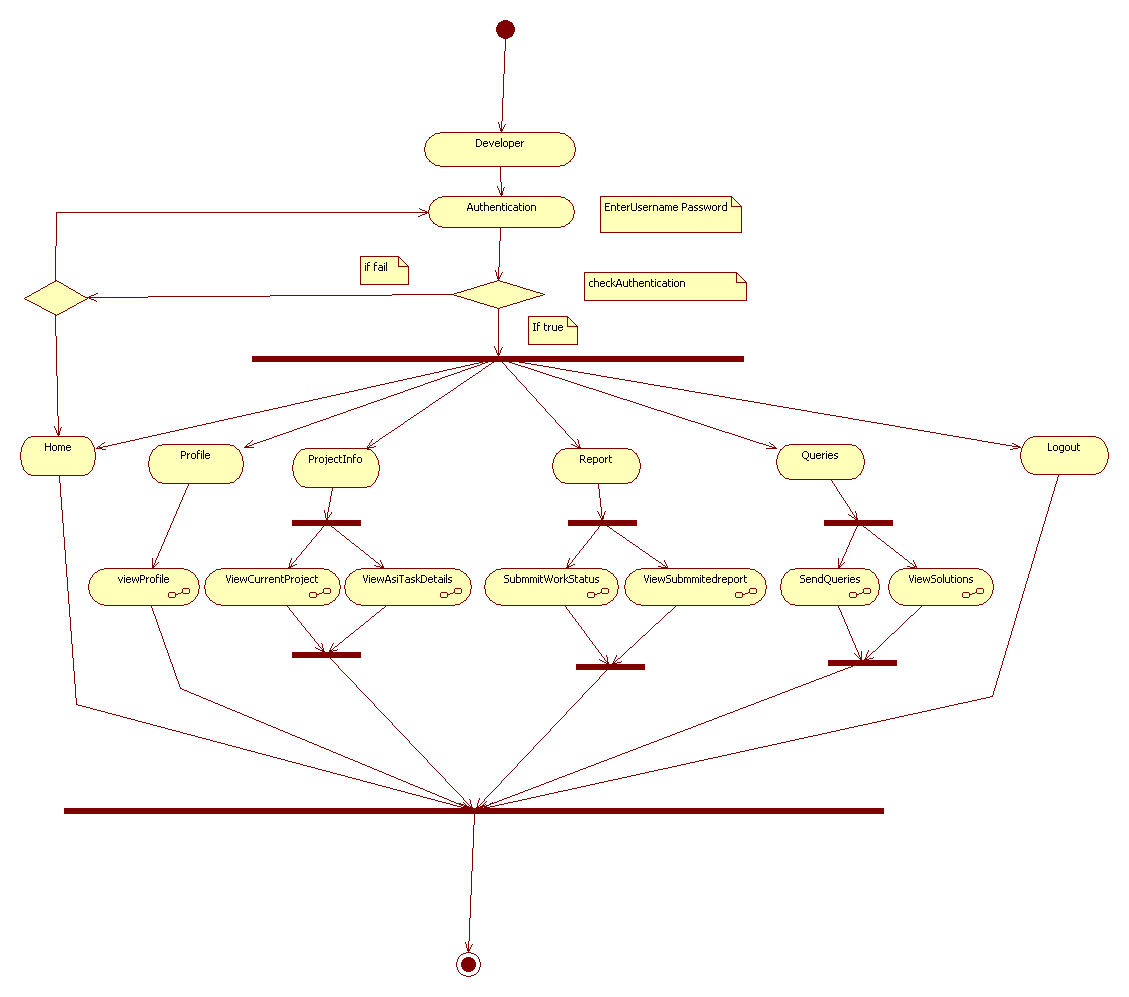
##### Administrator Activity Diagram



##### TeamLead Level Activity Diagram



##### Developer Level Activity Diagram

****

**Component Diagram**

**Component Diagram :**

****

**Deployment Diagram**

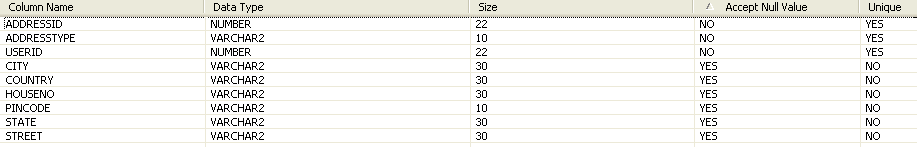
**Deployment Diagram:**

****

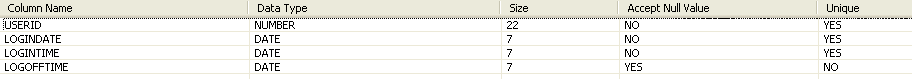
**Data Dictionary**

**Data Disctinory**

ADDRESSES



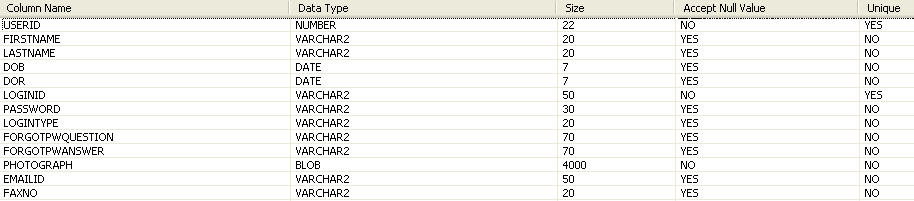
LOGINMASTER



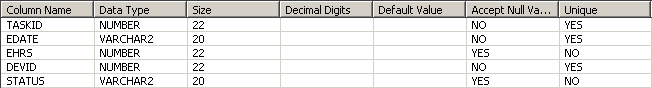
PHONES



USERDETAILS



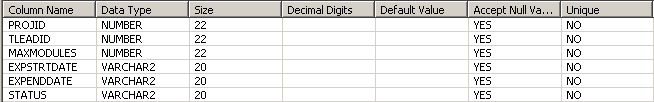
EFFORTDTLS

****

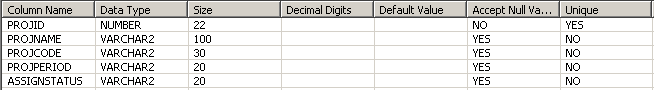
MODULES



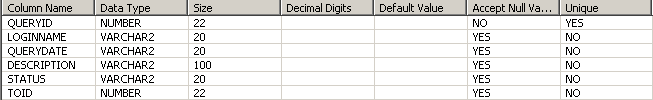
PROJASSIGNTOTLS



PROJDETAILS



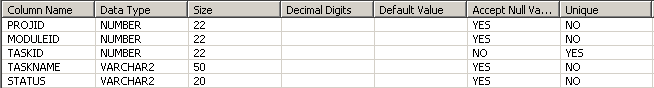
QUERY



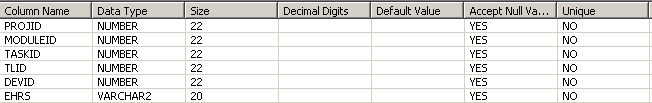
SOLUTION



SUBTASKS



TASKASIGNTODEV



TEAMASIGNTOTLS

****