



Addis Ababa University
College of Natural Sciences
Department of Computer Science

System Design Document

Human Resource Management System (HRMS)

For

Addis Ababa Prison Administration (AAPA)

Table of Contents

List of Figures	III
List of Tables	IV
1. Introduction	1
1.1. Over view	1
1.2. Definitions, acronyms and abbreviations	1
1.3. References	2
1.4. System Overview.....	2
1.5. Design Goals.....	2
2. Proposed Software Architecture	7
2.1. Over view	7
2.2. Sub system decomposition	9
2.3. Hardware/Software mapping	12
2.4. Persistent data management.....	13
2.5. Access Control and Security.....	14
2.6. Global software control	15
2.7. Boundary conditions	15

List of Figures

Fig1: Logical software architecture of HRMS.....	8
Fig2: Component diagram for MySQL.....	9
Fig3: Component diagram for Key man	10
Fig 4: Component diagram for Chrome	10
Fig 5: Component diagram for Tomcat.....	10
Fig 6: Restructured components specification	12
Fig 7: Deployment diagram.....	13
Fig 8: Database Schema	14

List of Tables

Table1: Access control matrix 15

1. Introduction

Up to now, we were concentrating on the problem domain. We elicited the user's requirement and document them in the requirements specification document. After this has been done, the analysis was done using different UML diagrams.

System design, object design and implementation constitute the construction of the system. During these three activities, the gap between the system specification, produced during requirements elicitation and analysis, and the system that is delivered to the user is filled. System design is the first step to get into the solution domain.

1.1. Overview

The goal of the System design is to produce a design specification of HRMS in terms of a document so that it is understandable by our development team.

In this document we will describe the design aspect of HRMS. We will consider design issues such as, HRMS's system architecture, its decomposition, hardware and software mapping, persistent data management, access control and security issues, boundary conditions and subsystem services.

1.2. Definitions, acronyms and abbreviations

HRMS: Human Resource Management System

AAPA: Addis Ababa Prison Administration

CRUD: Create Read Update Delete

IS: Integrated System

ORM: Object Relational Mapping

MVC: Model-View-Controller

SDD: System Design Document

DBMS: Database Management System

TCP/IP: Transfer Control Protocol/Internet Protocol

LAN: Local Area Network.

RDBMS: Relational Database Management Software.

CMS: Clinic Management System

PRMS: Prison Record Management System

GUI: Graphical User Interface

WAR: Web Archive

IDE: Integrated Development Environment

1.3. References

Before reading this system design document, we recommend the prerequisite document, Requirements specification of HRMS to be read.

1.4. System Overview

At this point, we have identified that HRMS is web-based application software. It is a subsystem of the Integrated Systems. The Integrated System (IS) comprises the Clinic subsystem and the Prison Record Management subsystem. HRMS is a system that maintains employees' records and process information of the stored records.

1.5. Design Goals

The list of design goals describes the qualities of the [HRMS](#) system, i.e. that we need to optimize. The design goals are derived from the nonfunctional requirements. These goals guide the decision to be made by developers especially when trade-offs are needed. For example, we need to decompose the database tables to the 3rd normal form. In doing so, we

also need some de-normalization operations so that join from fewer tables increase the performance of our system.

The design goals can be generally grouped into five categories. These are:

- ✓ Performance Criteria
- ✓ Dependability Criteria
- ✓ Maintenance Criteria
- ✓ Cost Criteria
- ✓ End User Criteria

Performance Criteria

System responds for user requests with in the first 15 seconds (Query Latency). Because, JSP servlets are pre compiled, they run at reasonably fast amount of time. The system should also perform with high throughput. The number of users that could concurrently access the system without the gradual degradation in performance of the web server is up to 100.

Dependability Criteria

The dependability criteria considers the following bulleted design goals

- ✓ **Robustness:** The System needs to be robust. It must be able to survive invalid user inputs such as trying to print a report for an employee with a non-existing id. We need to customize error messages in the language of the users using red colors and bold fonts.
- ✓ **Reliability:** The reliability of a system checks whether there exists a difference between the specified and the observed behavior of the system. HRMS needs to output appropriate and expected results to requests. For example, when users request for report of employee information, HRMS need to show the report instead of updating or doing other irrelevant activities.
- ✓ **Availability:** The system should be available in 24/7.

- ✓ **Security:** The system should implement firewall since the system is to be implemented in Intranet network. System shouldn't allow non-authorized users to access the system. This will be assured by implementing user accounts and account types for users of the system.

Maintenance Criteria

The maintenance criteria considers the following bulleted design goals

- ✓ **Extensibility:** The system should be extensible to further entertain new functionalities in further increments
- ✓ **Modifiability:** The system should be easy to apply refactoring options on to it. We need to make our subsystems loosely coupled so that modification to one subsystem at a later time, shouldn't affect the whole system.
- ✓ **Portability:** The system should be able to run on whatever kind of platform of personal computers and servers.
- ✓ **Readability:** The code for the system should be readable to add functionalities as modules of java project at a later time. System should incorporate comments in code.
- ✓ **Traceability of requirements:** The system should implement all the required functional requirements.

Cost Criteria

We will try to minimize development cost. Deployment cost is to be covered by AAPA to install the appropriate server and client machines.

End User Criteria

System should implement understandable and simple GUI.

The followings also need to be considered:

Assumptions and Dependencies

HRMS, being a web based application software is highly dependent on the existence of web server application software.

To store data, we need a persistent data management (DBMS) software.

A local area network must be available within AAPA to deploy our application.

End-Users need to have basic knowledge on how to use computer applications. They need to know Amharic since our system is to be developed using it.

End-Users must use off-the-shelf components such as key man or power geez to interact with the interface of our system. (We recommend key man).

General Constraint

The system should be used in parallel with the existing paper based HRM before it is tested that it could work independently. After it is tested that it works independently and if a network failure happens, the existing paper based HRM should be used as an alternative.

Software Environment

There is no restriction regarding the machine on which the application is to be deployed. Since the system is developed using ~~Java~~java, we get one benefit. I.e. ~~Platform~~platform portability.

HRMS will be developed using the following:

- ✓ Java script and JQuery are used to deliver dynamic content.
- ✓ Twitter bootstrap is used to draw stylized user controls.
- ✓ CSS is used to style pages.
- ✓ JSP is used to design pages.
- ✓ Java beans are used to model database and form objects.
- ✓ Java Controllers are used to control page navigation.
- ✓ MySQL workbench is used to create relationships between tables visually.

- ✓ MySQL database management system.
- ✓ Spring frame work which implements MVC architecture.
- ✓ Hibernate ORM frame work which does object relational mapping.
- ✓ Glass fish server is an application server.

To use java and design our pages, we will use IntelliJIDEA IDE because it has got good refactoring options (one of the many important features of the IDE). In order to use [Java](#), we need the appropriate compatible JDKs and JVMs.

In order to design Jsp pages whose controls are compatible with the back end java bean, we need special tag libraries.

Standards Compliance

We shall adhere to the standards set for every resource we are going to use. When using Java for example, such standards include:

- ✓ Naming Conventions for classes, methods and variables and database tables.
- ✓ We need to include comments.

Memory and Capacity Limitations

To proceed with the effective management of server's database, HRMS shall be deployed on a server machine which has at least 3 GB RAM and more than 250GB hard disk space.

2. Proposed Software Architecture

2.1. Over view

AAPA has no previous computerized HRM system. Since this is a green field engineering project which is done from scratch, we don't mention previous software architecture. But here is the proposed software architecture for our system.

Logical Software Architecture

The logical software architecture of HRMS is MVC architecture. Our system is to be divided into different layers that depict this typical architecture.

- In the Data Access Layer (repository layer), we implement classes for basic CRUD database operations. We also need to include different custom queries at this layer.
- In the Domain layer, we implement java beans used to model form and database table objects.
- In the Controller layer, we implement classes that control page navigation.
- In the Service layer, we implement classes of the actual work. I.e. we implement our use cases.
- In the view layer, we design JSP pages that client browser renders to the user.

In implementation we are strictly following the logical software architecture.

The following figure depicts the logical software architecture of our system, HRMS.

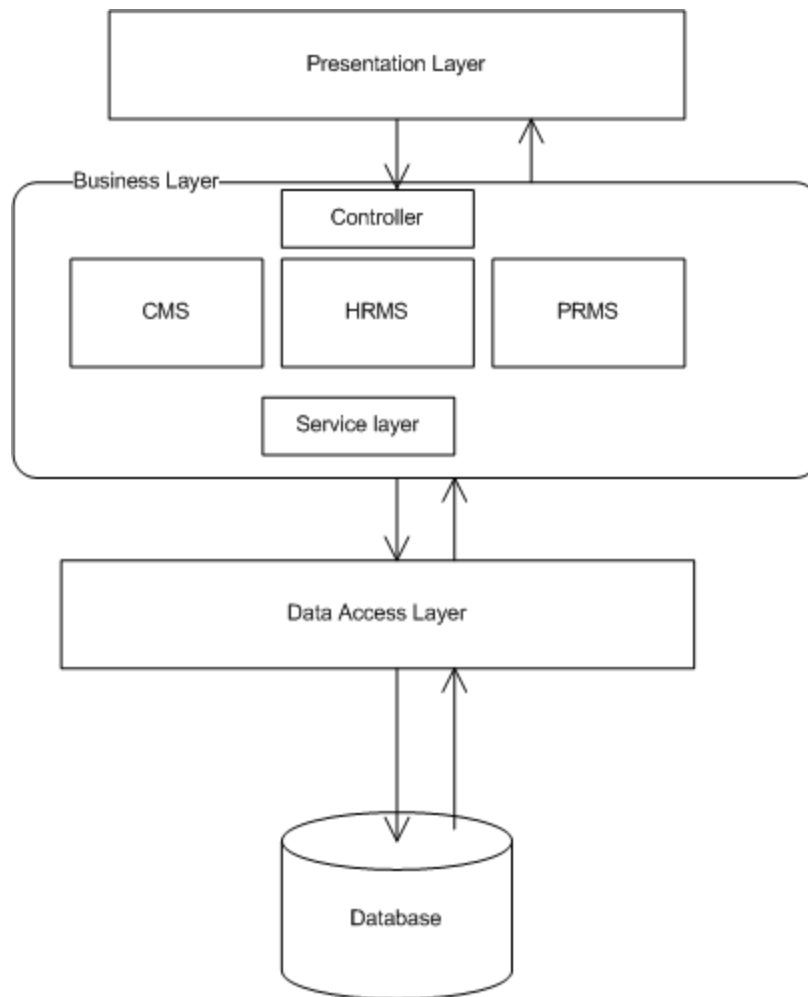


Fig1: Logical software architecture of HRMS.

Physical Software Architecture

This architecture describes how the different components and nodes are arranged within the system. HRMS follows 3-tier architecture. Because,

- We need a presentation layer for user interface. The user interface component, i.e. the JSP page is implemented for users to access IS through simpler user interface controls.
- The Application logic which is to be deployed at the Application server's side contains the subsystems and the services provided by them. It also accepts user requests and processes them accordingly.

- Finally we need a persistent data management for data storage. This layer contains a DBMS which can be viewed as one single subsystem.

2.2. Sub system decomposition

The following are subsystems identified within our system. Our system also needs the services provided by different off-the-shelf components. For example, we are using the following off-the-shelf components.

- Key man: Amharic software providing fonts for Amharic typing on web pages.
- Tomcat: An application server on which the application (business logic) of our system is to be deployed.
- MySQL: A Database Server on which the database of our system is to be deployed.
- Chrome: Browser in which users will interfere with the presentation layer of the system.

The following component diagrams illustrate the services provided by the above off-the-shelf components.

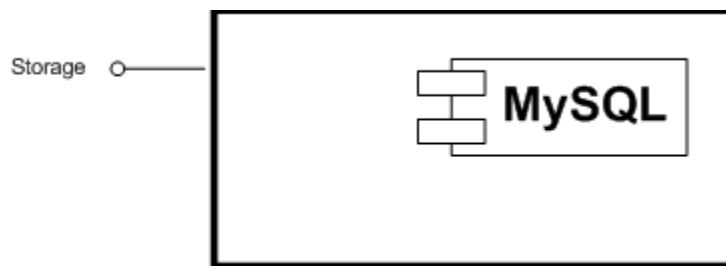


Fig2: Component diagram for MySQL.

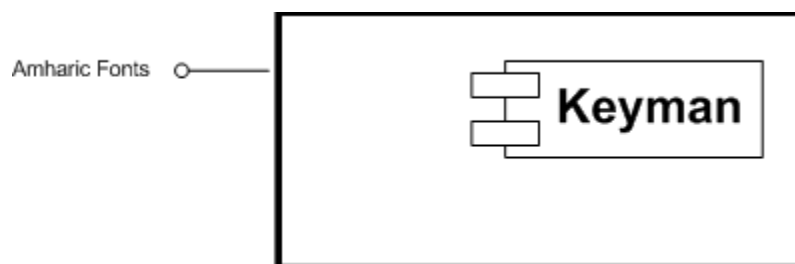


Fig3: Component diagram for Key man

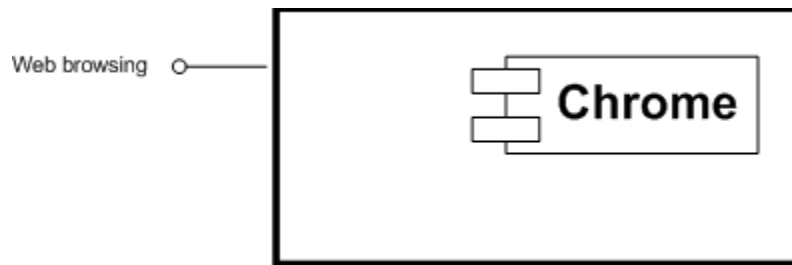


Fig 4: Component diagram for Chrome

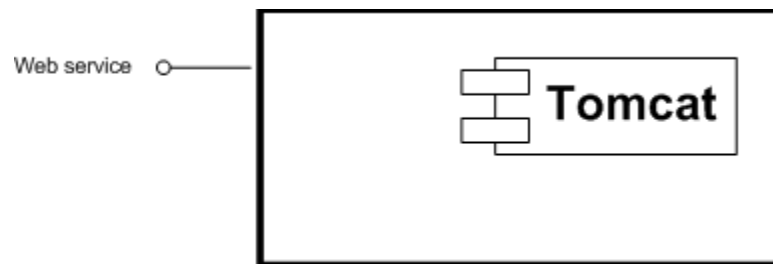


Fig 5: Component diagram for Tomcat

Registration subsystem

The services provided by this subsystem are as follows:

- ✓ Basic registration of employees.
- ✓ Full registration of employees.
- ✓ Update the information of employees.

This subsystem requires services from other subsystems such as the Authentication subsystem which is discussed below. It also interacts with the DBMS subsystem.

Authentication Subsystem

The Authentication subsystem makes sure that users of the system are eligible or not. This subsystem provides security as a service. It requires the service provided by the Database subsystem.

Administration Subsystem

The Administration subsystem provides the following services:

- ✓ Options for registering, adding, deleting and updating database lookup objects such as list of Cities, list of countries, list of country languages and so on.
- ✓ Registering work items in the database.

It interacts with the Authentication subsystem and the database subsystem.

File Tracking Subsystem

The basic service provided by this subsystem is registering file record places and telling where they are located at current time.

Permission Handling Subsystem

This subsystem is responsible for handling permission of employees of AAPA. It interacts with the database subsystem.

Transfer Handling Subsystem

This subsystem is responsible for handling the transfer from one department to the other for employees working at AAPA(internal transfer). It interacts with the database subsystem.

Restructured Subsystem Specification

The following figure illustrates the relationship among subsystems of HRMS. All subsystems need to read/write/update the database subsystem. They also have interaction with the off-the-shelf components described above.

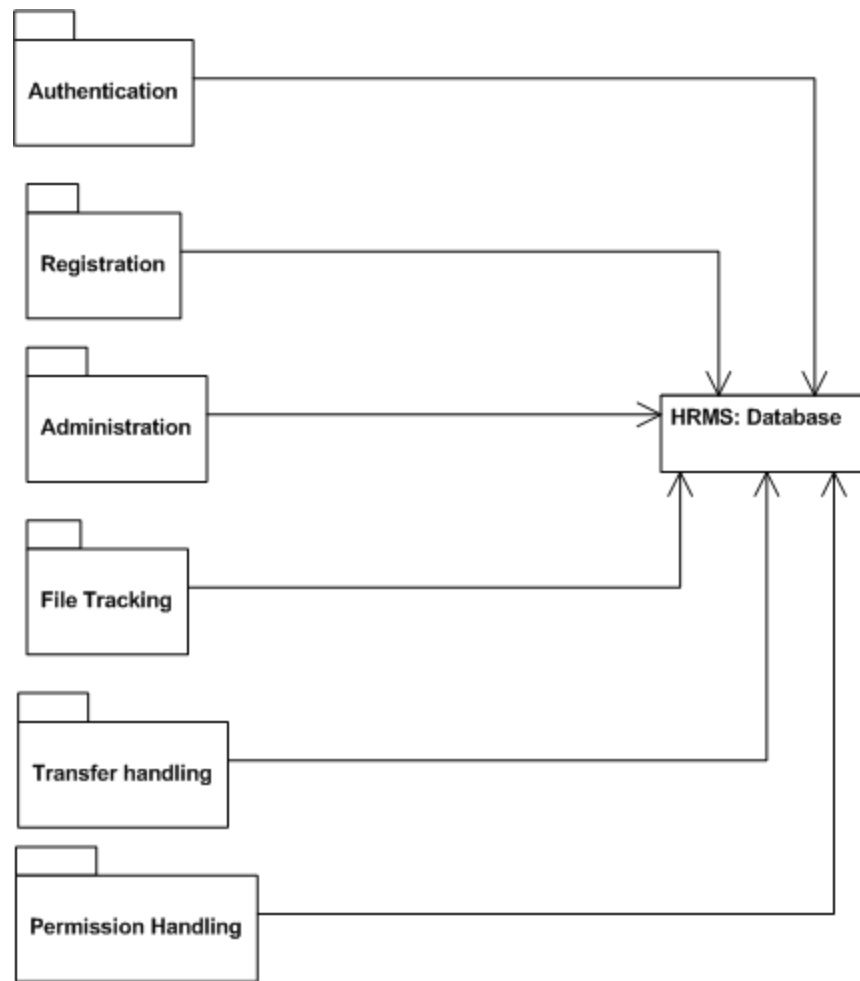


Fig 6: Restructured components specification

2.3. Hardware/Software mapping

Each subsystem has to be mapped to components and each component needs to be mapped to different nodes. The above subsystems with their list of services can be regarded as components and in turn they are to be mapped to nodes.

The hardware configuration of the system will be using a client machine and a server machine. For the effective communication between these nodes, we use a LAN under TCP/IP infrastructure.

Off-the-shelf components, Key man and browsers Chrome, fire fox and so on are to be deployed at the client side's node. The application logic of HRMS comprises the subsystems

listed above that are to be deployed on the off-the-shelf component, Application server Tomcat on the server side's node. The Database of our system is to be deployed on the off-the-shelf component Database Server MySQL on the server side's node. Our system as said earlier forms a three-tier architecture which is described in terms of figure 7 below.

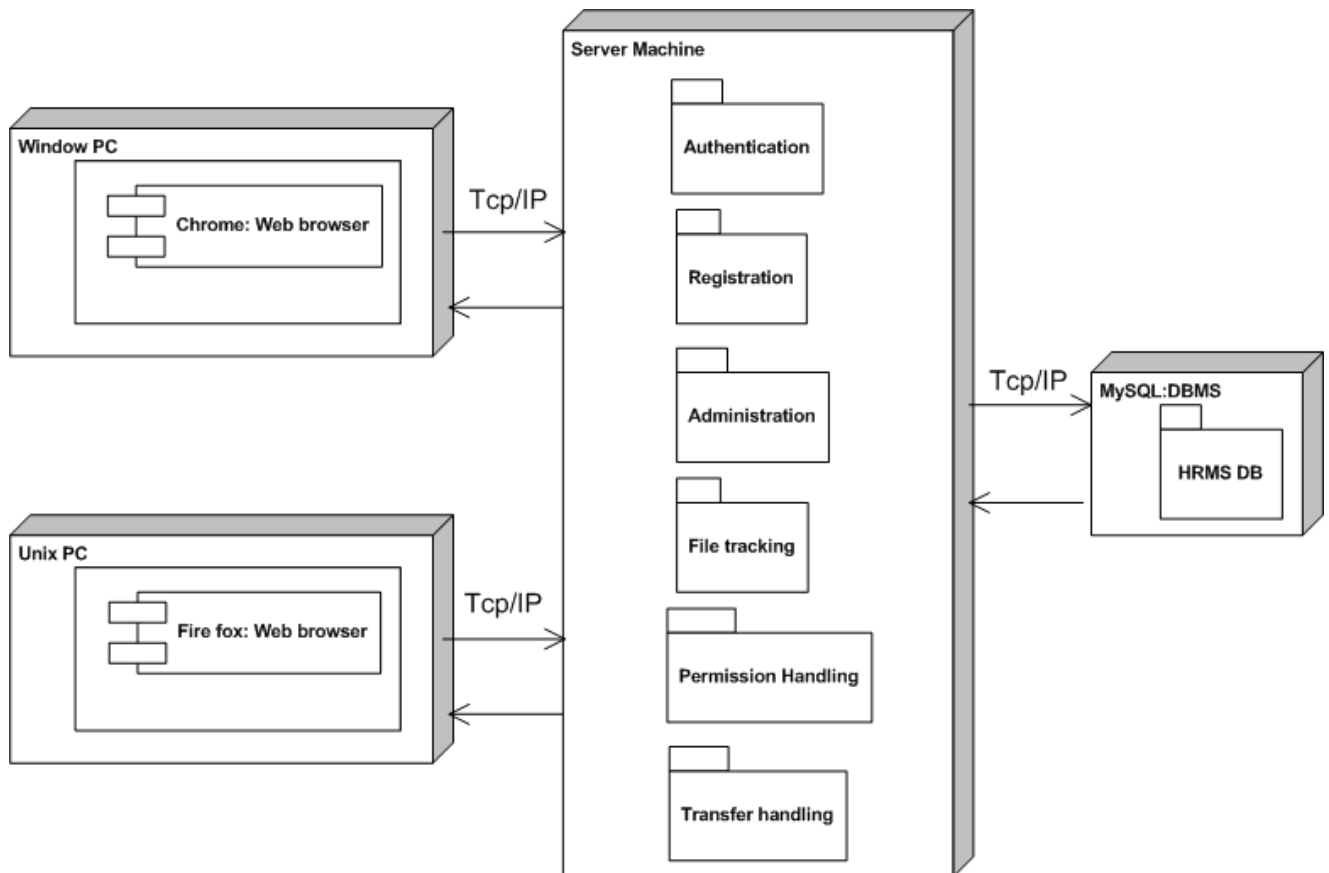


Fig 7: Deployment diagram

2.4. Persistent data management

For persistent data management, we use MySQL, a popular open source RDBMS. This database keeps track of basic and detailed information of employees working at AAPA. HRMS shall implement some authentication mechanism for the access requests that came to the database.

Mapping

In order to store information persistently, we map objects into tables of MySQL and their attributes to the columns of the table. Relationships among tables have to be implemented. We implement the three types of relationships 1 to 1, 1 to Many, Many to Many. The following tables, their attributes and the relationship among tables have been identified so far.

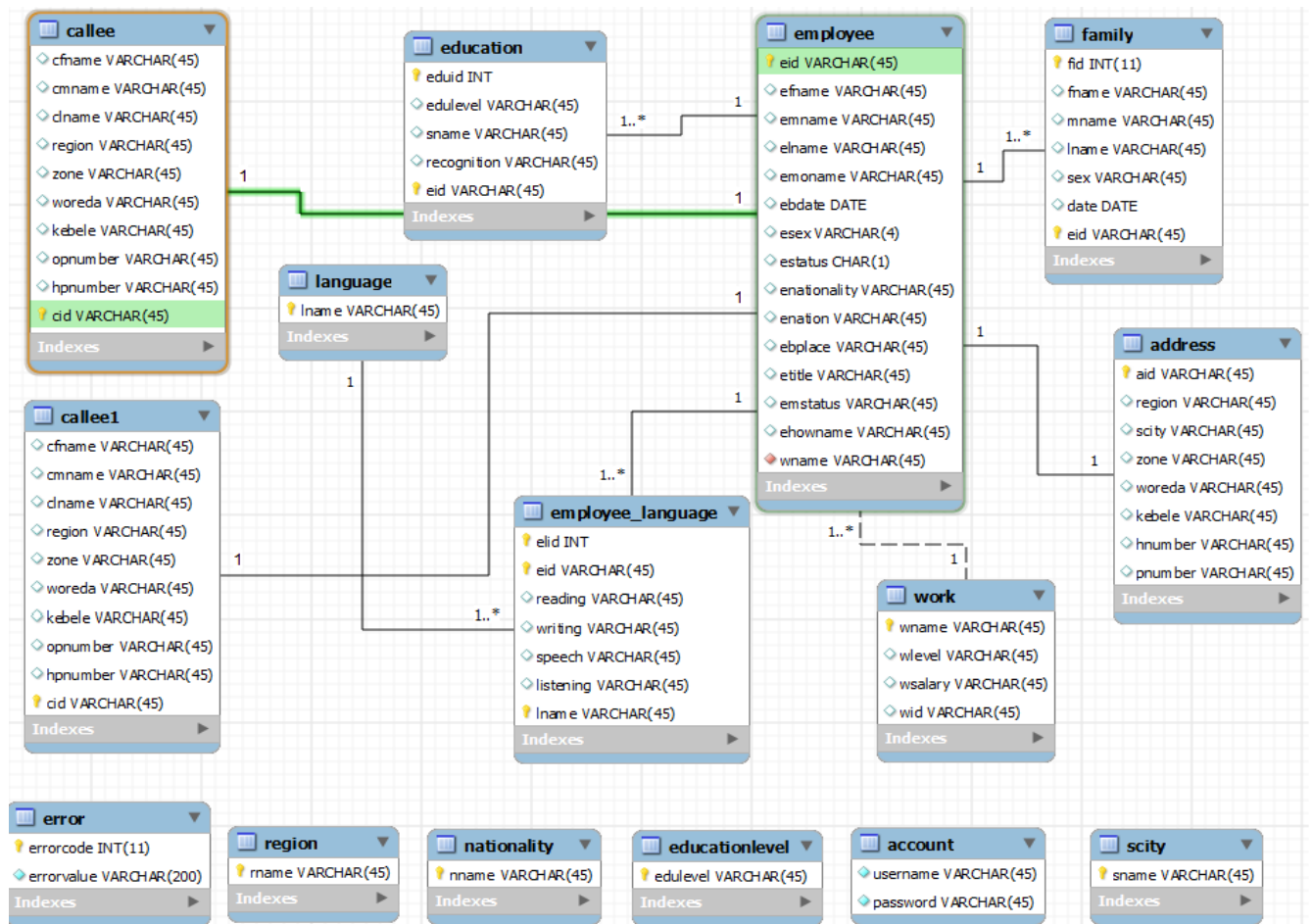


Fig 8: Database Schema

2.5. Access Control and Security

One way to show, which can access what is by using an access control matrix. The following figure depicts an actor who is involved in the scene, and the possible privileges assigned to the actor.

Table1: Access control matrix

Actor	Privilege
HRM Personnel	<ul style="list-style-type: none">➤ Registers employee information➤ Process Update➤ Handles transfer➤ Handles permission
HRM Leader	<ul style="list-style-type: none">➤ Authenticates forms➤ Approves forms
Administrator	<ul style="list-style-type: none">➤ Fills, Updates and deletes database lookup items.

2.6. Global software control

The control flow of our system is thread-based. I.e. the web server on which our application is to be deployed can handle more than one user request at a time. HRM personnel of the ministry have a collective job as said in the requirement analysis document. So we need to ensure interaction so that operations in the critical region shouldn't affect data consistency negatively. This can be assured by having lock variables. When one worker updates the entries of employee records, the lock variable is set so that the other waits until the operation ends.

2.7. Boundary conditions

The final output of the application is in IS.war file form. It is this web archive file that is to be deployed on Glass fish server. This web archive file is precompiled before the first request to the home page of the IS is made. In earlier cases, the web archive is compiled when the first HTTP Request arrives. This needs some compilation time and may result in significant delay of HTTP Response. But after the second request, since the servlet is precompiled, response will be fast. Pre compiled IS.war even makes the response of the first request fast.

System initialization begins when the web server, Glass fish is run. Users may issue requests to IS by typing the URL of IS in the Address bar of their browser.

System shutdown is when administrators explicitly close the web server.

We shall load database look up items when the first request to the login page of IS is issued.

When the system exists the data structures used to hold look up items will be freed.

Off-the-shelf component key man needs to be started before the system starts up.