
# Software Requirements Specification

# EmployeeManagement System

**Prepared by**

**JSS Team**

**17.07.2023**

# 1 Introduction

The Software Requirements Specification (SRS) document outlines the requirements and specifications for the development of an Automated Recruiting System that leverages Cognitive Analysis and Machine Learning techniques. This advanced system aims to streamline and enhance the recruitment process by intelligently analyzing candidate profiles, assessing their suitability, and recommending the most promising candidates for various positions within an organization.

## *1.1 Purpose*

The process of recruiting and hiring new personnel poses intricate challenges for organizations, necessitating a comprehensive solution that goes beyond traditional recruitment methods. Conventional approaches involve manual tasks such as posting job openings, reviewing resumes, and conducting interviews. However, these processes often lack the capability to assess candidates' cognitive abilities and personality traits, leading to hiring decisions that may not align optimally with the organization's requirements and culture.

## *1.2 Problem Definition*

The process of recruiting and hiring new personnel is a critical and often complex undertaking for any organization. Traditionally, recruitment involves manual tasks such as posting job openings, sifting through numerous resumes, scheduling interviews, conducting assessments, and managing communication with candidates. These processes are time-consuming, error-prone, and can lead to inefficiencies that hinder the overall effectiveness of the recruitment process.

### 1.2.1 Current Challenges

- **Limited Assessment:** Existing recruitment methods primarily focus on qualifications and experience, potentially overlooking candidates with the right cognitive abilities and personality traits.
- I**nconsistent Evaluation:** The absence of standardized cognitive and personality assessment tools can led to subjective evaluations and inconsistent hiring decisions.
- **Mismatched Candidates:** Hiring candidates without considering their cognitive fit for a role can result in poor job satisfaction and higher turnover rates.
- **Time-Consuming Analysis:** Manual cognitive and personality assessments are resource-intensive and can extend the recruitment process, causing delays in candidate selection.

### 1.2.2 Solution Overview

The Enhanced Automated Recruitment System aims to revolutionize the recruitment process by incorporating cognitive and personality analysis into the evaluation criteria. This system goes beyond conventional methods and offers the following key features:

- **Cognitive Assessment:** Integration of cognitive evaluation tools that measure candidates' problem-solving, critical thinking, and decision-making abilities.
- **Personality Profiling**: Utilization of personality assessment methodologies to gauge traits like teamwork, adaptability, and communication skills.
- **Objective Evaluation:** Standardized assessment methods that provide an objective basis for evaluating candidates' cognitive strengths and personality attributes.
- **Data-Driven Insights:** Collection and analysis of cognitive and personality data to predict candidates' suitability for specific roles and teams.

## *1.2 Objective*

The primary objectives of the Enhanced Automated Recruitment System with Cognitive and Personality Analysis are as follows:

- Develop a sophisticated platform that integrates automated cognitive and personality assessment tools within the recruitment process.
- Enable recruiters to evaluate candidates' cognitive aptitude and personality traits to ensure a holistic fit with organizational requirements.
- Offer an objective and standardized evaluation process that complements traditional qualifications-based assessments.
- Leverage data analytics to provide insights into the correlation between cognitive attributes, personality traits, and job performance.
- Facilitate well-informed recruitment decisions that align candidates with roles that suit their cognitive and personality profiles.

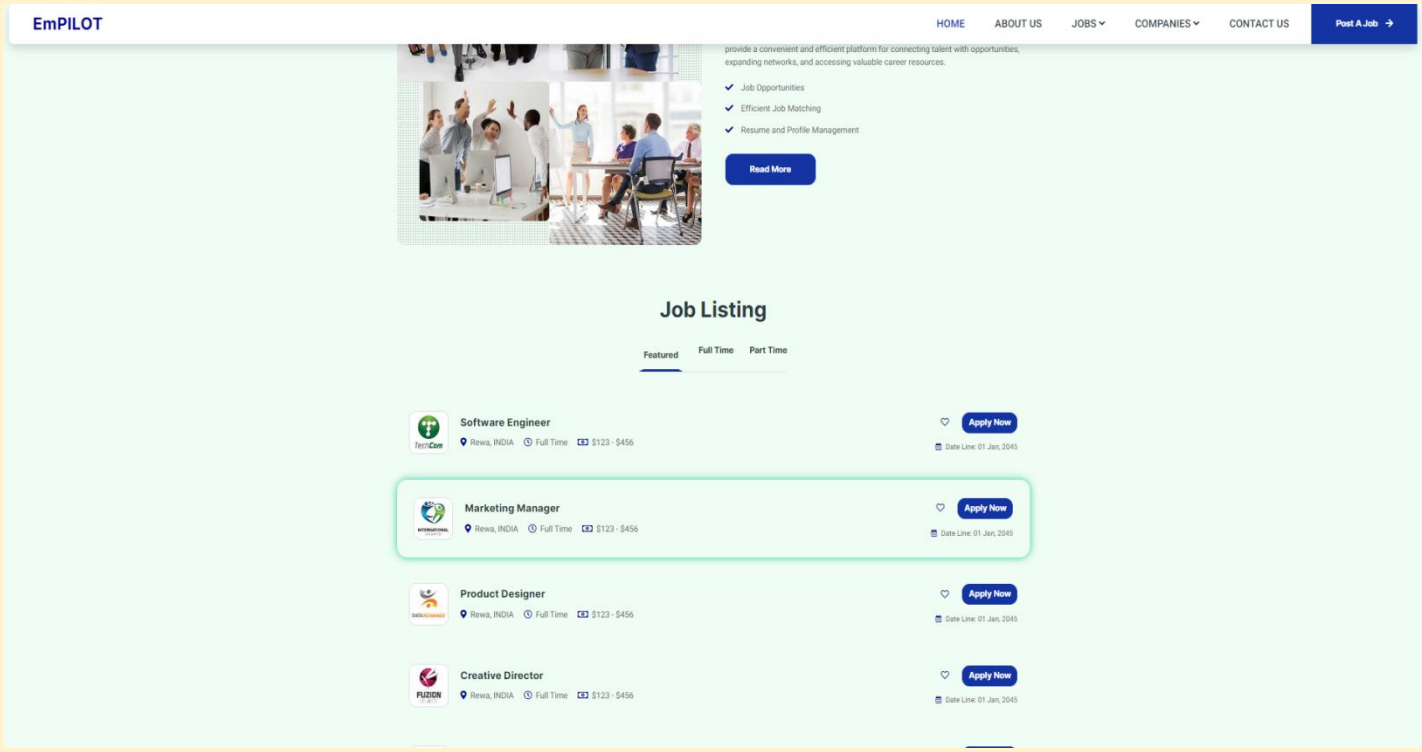
By addressing the limitations of traditional recruitment processes through the incorporation of cognitive and personality analysis, the Enhanced Automated Recruitment System aims to elevate the quality and precision of hiring decisions, resulting in more successful and fulfilling employee placements.This updated problem definition section highlights the challenges associated with assessing cognitive and personality attributes in recruitment and outlines the objectives of the Enhanced Automated Recruitment System with Cognitive and Personality Analysis. Adapt the content to accurately convey the specific goals and context of your project.

## *1.3 Scope*

This document covers the whole definition of the EMPLOYEE Management System (ERMS) project. It basically includes the requirements for managing the personal data, controlling authentication and authorization mechanism, and evaluating of employees' performance.

More specifically, our ERMS (EMPLOYEE Management System) controls and manages the personal database such that any user with different role types as manager, admin, employee, and Employee will be able to manipulate their personal data.

In addition to manipulating the personal data, our ERMS will provide authentication and authorization mechanism. Every user with any role type can be able to login to the system with his/her username and password.




## *1.4 User and Literature Survey*

Although most of the companies of the world have been using ERMS for a long time as a result of a need for reaching and managing the data of employees, ERMS projects are developed without the required properties they to be well-formed. Unfortunately, this situation is not different in Turkey. Due to the rapid progress in companies like increase in number of the employees, the increase in the expectations in employees' skills and the variations of the employee types, the existing ERMS becomes inadequate to meet these required needs.

We aim to develop a ERMS in such a manner that it would be able to fulfill the upcoming changes in the needs of Siemens Company. In other words, our ERMS will be a flexible system such that it could be improved according to the future needs of Siemens Company.

## *1.5 Definitions and Abbreviations*

- SRS: Software Requirements Specification

- ERMS: EmployeeManagement System

### *1.6 References*

[1] IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

[2] http://www.webdunia.net/CaseStudy/internal_caseStudies/CaseStudyERMS2.pdf

[3] http://www.ibm.com/developerworks/rational/library/769.html

[4] http://www.cs.iusb.edu/thesis/SLingareddy_thesis.pdf

[5] http://en.wikipedia.org/wiki/Eclipse_(software)

[6] http://www.eclipse.org/org/

Robin Schumacher, Arjen Lentz. "Dispelling the Myths' [7]MySQL AB. Retrieved 2007-02-10. .

[8] "What is MySQL, MySQL 5.1 Reference Manual". MySQL AB. Retrieved 2011-08-26. [9]

http://www.microsoft.com/sqlserver/en/us/default.aspx

### *1.7 Overview*

This SRS is organized in a way that any user o can easily understand and use the ERMS. In other words, it constitutes a user guideline for ERMS.

Basically, this document starts with a brief explanation of the problem. Later on, it continues with a detailed solution we proposed. Also block diagrams of our solution to visualize the solution and system better, specific and functional requirements, interface requirements, constraints that may be confronted while developing the software, relationship between user types each of which is a data object. This document ends with the planning and estimating the basic schedule of our process.

## 2 Overall Description

The overall description of our project can be stated as creating and managing the database, developing a friendly user interface to manipulate the database, provide an authentication mechanism to safely accomplish tasks mentioned above.

### *2.1 Product Perspective*

Currently, There exists multiple ERMS's however, with this ERMS we will provide them with additional capabilities and properties organized neatly using some machine learning algorithms.

ERMS which is an online intranet System will be used by employees as well as recruiters.. These types who have different roles can be stated as; admin, manager, EMPLOYEE, employee. Every user enters the main authentication page and after that, system will grant them authorization. After being authorized according to their permissions (role type) users will basically query and edit the database via ERMS.

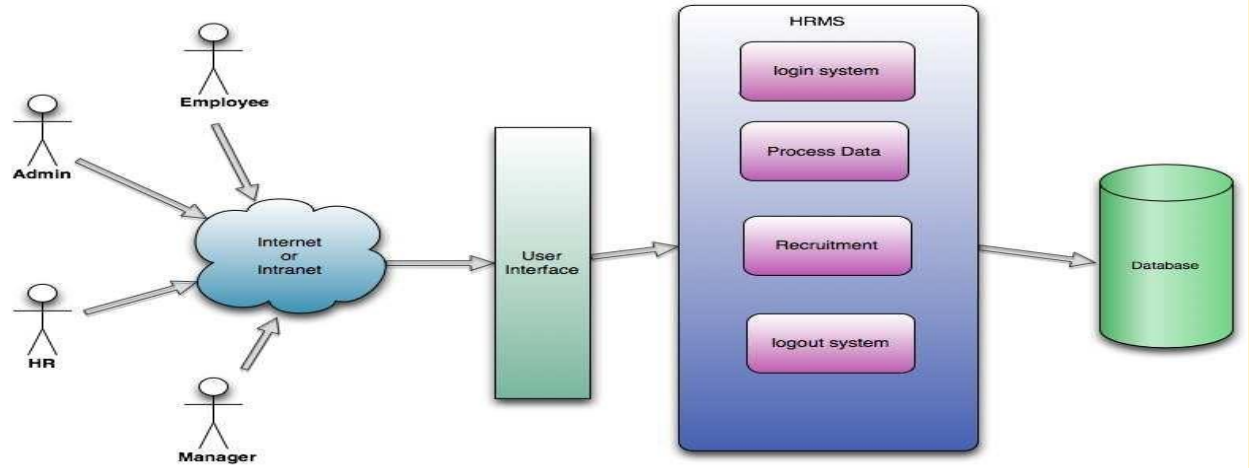



Figure 2.1.1: General Use Case Diagram [2]

## *2.2 Product Functions*

ERMS implements some major functions in order to accomplish required tasks. These functions constitute a basis for the whole system. These functions can be stated as:

### 2.2.1) Authentication and Authorization

Being connected to internet, users will be able to get into the system. In order to see the interface related to his/her role type, the users account should be authorized and also his/her user name and password should be authenticated. These tasks are basically held by the functions implemented under the header of Authentication and Authorization major function.

### 2.2.2) Process Data

These functions which can be examined in that process data major function are basically provides user to manage the database according to the desired task. These management tasks constitute the major feature of the ERMS. With the help of these functions a user can update some basic personal data like contact information,name,resume etc. In addition to update data, a user can also search the database in order to obtain the list of the users' which has the properties desired. Also a user may also see the specific information about a user or all users which can be named as report. In other words searching is the operation with rows of the database while reporting is operation with columns of the database.

### 2.2.3) Recruitment

Recruitment of a new person which means introducing a new user to the system can be accomplished in two steps. When it is needed to add a new user to the system, firstly, EMPLOYEE must create an employee account, by the way at this step ERMS automatically gives an id to that user. At the second step, admin creates a user related to that user id.

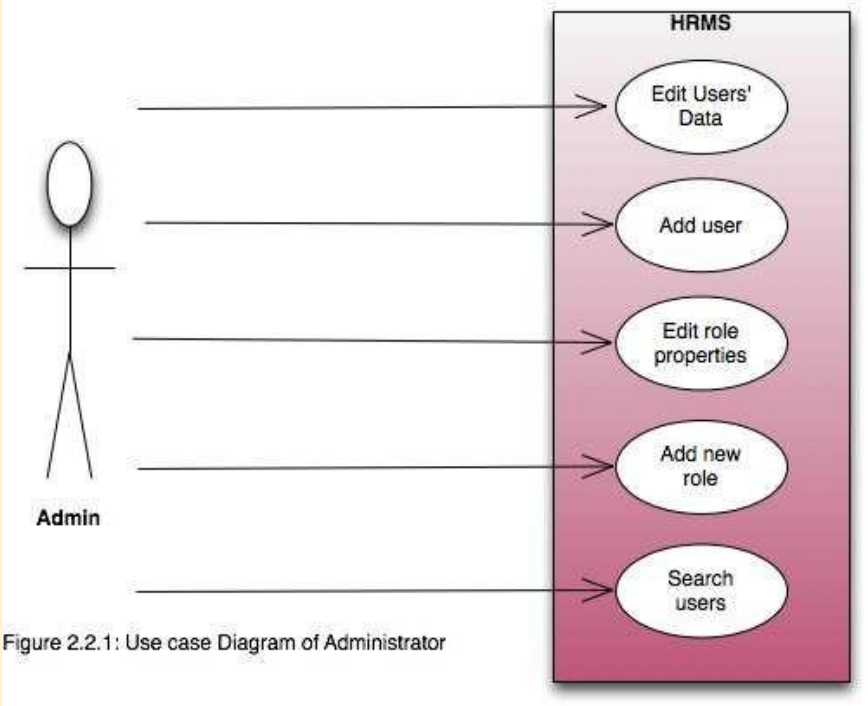



Figure 2.2.1: Use case Diagram of Administrator

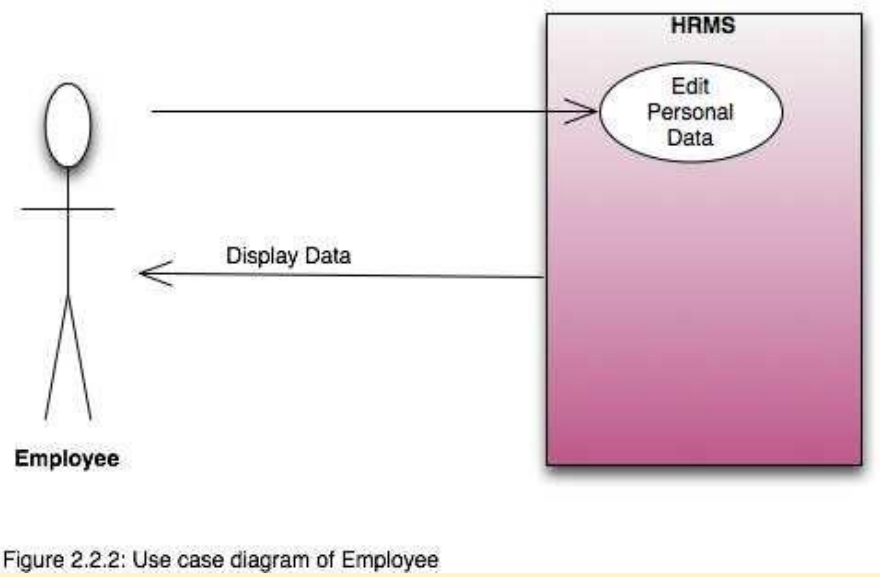



Figure 2.2.2: Use case diagram of Employee

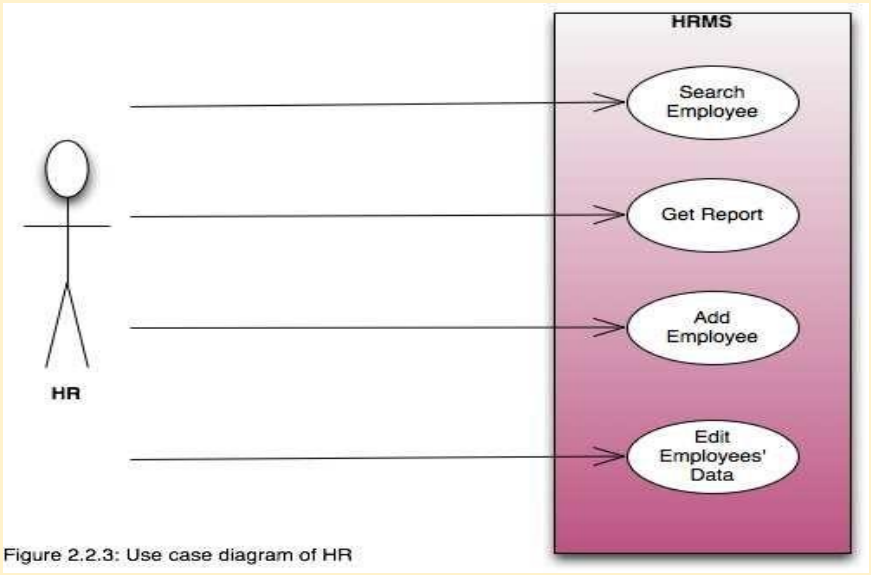


Figure 2.2.3: Use case diagram of HR

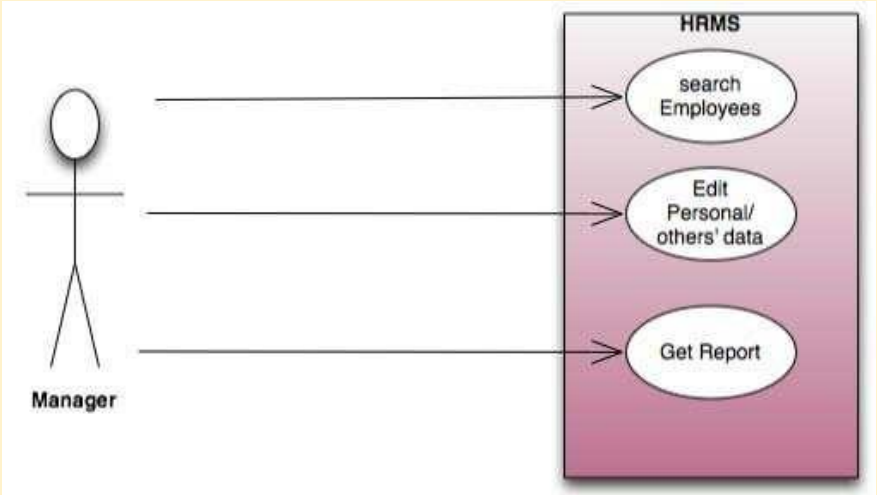


Figure 2.2.4: Use case diagram of Manager

## *2.3 Constraints, Assumptions and Dependencies*

- **Regularity Policies:**

Each user must be an registered as employee or as the recruiter. In other words, each user has account created by EMPLOYEE and authenticated by admin.

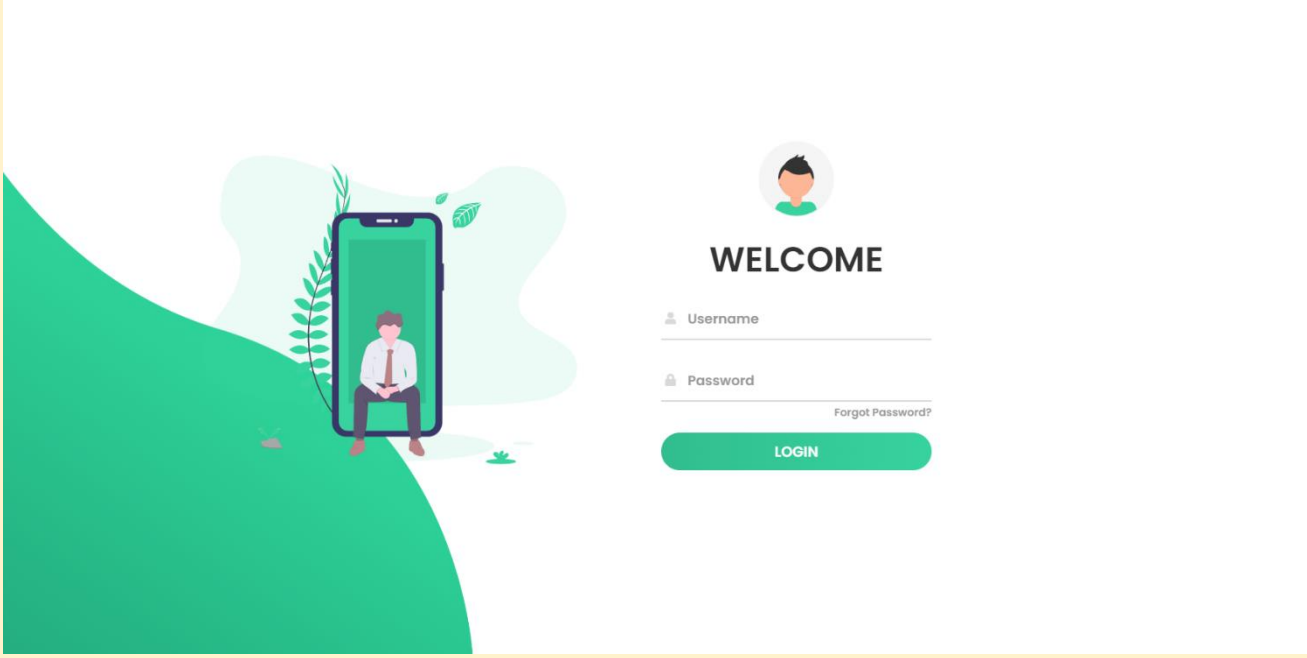
- **Hardware Limitations**

There is no limitation in the operating system in which ERMS will work. However, the ERMS system and the database will work on a server that needs to be always online. Users can access the system with any internet browser and machine learning algorithms.

# 3 Specific Requirements

## *3.1 Interface Requirements*

All the users will see the same page when they enter ERMS. This page asks the users a username and a password.




After being authenticated users will see the interface containing the information of the first tab of the user role types. This interface include different tabs according to their role types determined by admin at the authentication phase. By the way, users may have more than one role. Then a user who has more than one role will be able to see all the tabs that are related to his/her role types. These tabs can be named as; Personal Data Tab, Employee List Tab, Add New Employee Tab, Add New User Tab, Manage Users Tab, Users List Tab and Arrange Roles Tab.

User who have employee role have authorization to see only Personal Data Tab, EMPLOYEE role gives right to see Employee List Tab and Add New Employee Tab.Admin role gives user right to see User List Tab and Arrange Roles Tab. These authorizations are default ones but an admin can change these authorizations by Arrange Roles Tab. These tabs can be explained detailed as:

- **HTML5**
- **SCSS**
- **BOOTSTRAP**
- **JAVASCRIPT**
- **DJANGO**

- **REACTSJS**
- **SQLite**

### 3.1.1 Personal Data Tab

With the help of this tab, employees will be able to see their personal information which appears in a user-friendly design and also by means of this tab they may edit, update some information in other words manage some personal information which are updatable such as contact information, training information. This tab will only seen by the users who has a role of Employee.

### 3.1.2 Employee List Tab

This tab gives the list of all employees as selectable format (there will be a check box near each employee). This tab includes two function buttons namely report and search.

When clicked on report button, a window will be opened which enables the user to select any column that he/she wants to see the specified employees' (Employee list is selectable so EMPLOYEE can specify users by selecting them from the list) or all employees' information under preferred column/s.

When clicked on search button, again a window will be opened in which the user can enter the field name and the desired value. There can be more than one entry and user can choose to OR/AND them.

### 3.1.3 Add New Employee Tab

With this tab EMPLOYEE can add new employee (when a new employee is recruited) to the employee database with filling personal master data of this new employee. This tab adds new employee without any authenticated or authorized user attached to it. In other words only being added via this interface by EMPLOYEE is not enough to access this system. This account must be validated by admins.

Clicking on this tab an admin will see a window like:

### 3.1.4 Add New User Tab

With the help of his tab, Admins can add new user to the system. Admin can add user with user ID and arrange authentication and authorization to it. This user will be created by syncEmployeeonized to the employee with the same ID from the employee database.

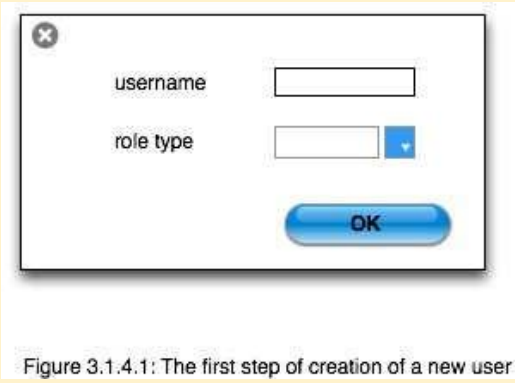
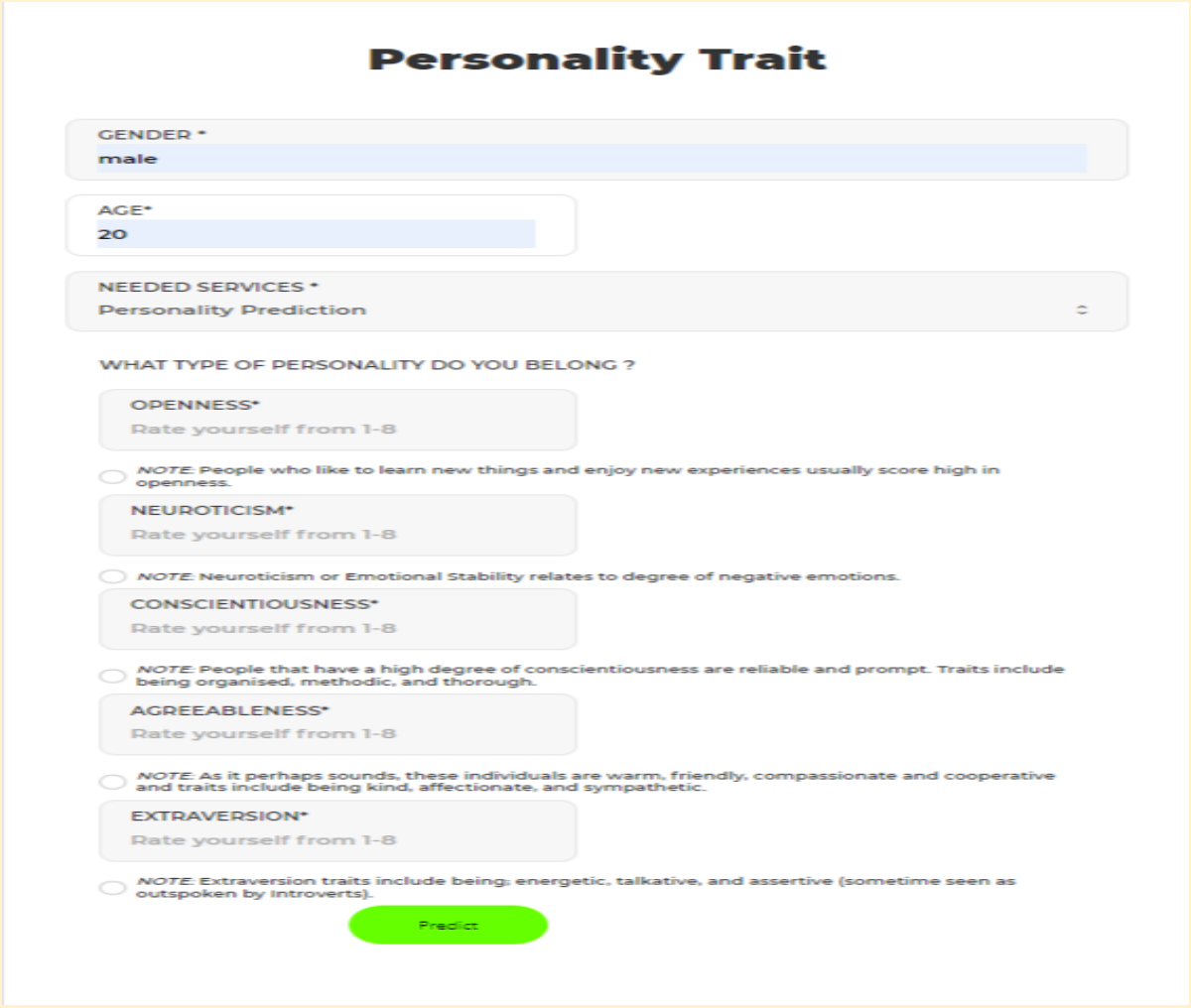



Figure 3.1.4.1: The first step of creation of a new user

### 3.1.5 User List Tab

In this tab, an Admin can list all users. Also there exist a search button as in Manage Users Tab which lets the admin to select user from the list by searching. After selecting the user, a new window is opened with information of that user. Admin can edit all information of that user including roles information (roles part of a user can be changed only by admin). When admin wants to change role of a user, a list came up of all roles determined in the system. Admin can select/deselect roles from this list; this selection determines authorization of this user. If any of these roles are selected then automatically this user becomes unauthenticated.



## *3.2 Functional Requirements*

In this section, we will explain the major functions of ERMS along with the data flow. So the major functionality of the project such as authentication mechanism, personal data processing, recruitment, report, and graphical user interface unit will explained step by step.

### 3.2.1. Functional requirement 1..n

#### 3.2.1.1 Authentication

| | |
|---|---|
| Login | User can login to the ERMS system with his/her username and password. |
| Logout | User can logout from the ERMS system. |
| Login failure | If the user does not exists in the database or the user did not get authorized by the ERMS admin yet. |

#### 3.2.1.2 Authorization

| | |
|---|---|
| User role check | After logging in, the user role will be checked from the database and the user interface will be created according to that role/roles. |

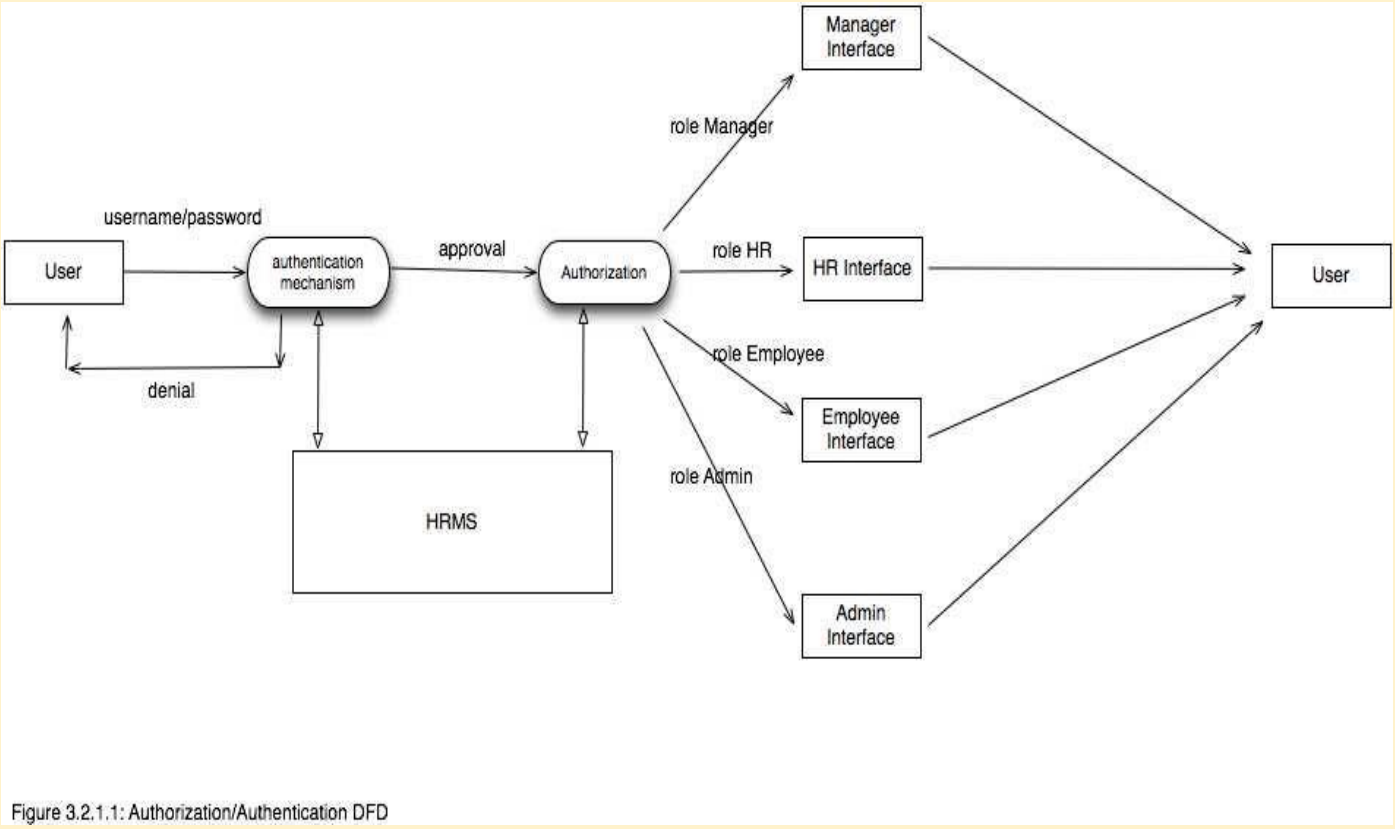



Figure 3.2.1.1: Authorization/Authentication DFD

### 3.2.1.3 Process Data

| | |
|---|---|
| Display | User with defined roles can display the content of the database. Being more specific, employee can only view his/her personal information. Manager can see not only his/her personal information but also employees' information who are working under his/her coverage. Admin and EMPLOYEE can display their personal information and all employees' information. |
| Edit | A user with employee role can edit his/her specific personal information. Manager can only edit employees' personal information that is under his/her coverage except user role type. EMPLOYEE can edit all employees' information except user role type. Admin can edit all information related to all employees' including their user role type. |
| Search | User with manager role type can search the content of database for the employees who are under his/her coverage. EMPLOYEE and admin roles can search all the employees' information in the database. Search feature works on specific keywords showing employees' characteristics, peculiarities, skills, features, and etc. For example, EMPLOYEE wants to find employees who are well trained in "Java Programming Language". He/she will write the specific keyword in the search bar and press the available search button. Afterwards, he/she will find a list of all the employees' who know "Java Programming Language". |
| Report | This feature is basically used to filter the contents of the search mechanism. For instance, as we mentioned in the above search feature. The EMPLOYEE wants to get a report of some specific employees who know "java programming Language". The list of employees obtained from the result of search feature he/she can get the specific report by selecting the corresponding checkbox available for each employee. Or a manager role type can get a report of some or all employees' who are working under his/her coverage by selecting the checkbox. Except employee role type, all other role types such as admin, EMPLOYEE, and manager can use this feature. |
| Update authentication | This feature can be used only by admin role type. Admin can update the role type of a specific user. For example, an employee got promotion and his role type will be changed from employee role to manager role. Admin will be able to update this authentication mechanism. |

### 3.2.1.4 Recruitment

| | |
|---|---|
| Add a new employee | EMPLOYEE role type is able to add a new employee to the database. The new employee will have all the required personal information related to his/her. The new created employee will have an id. |

| | |
|---|---|
| Add a new user | After being created a new employee by EMPLOYEE role, admin role is responsible for creating a new user by the specified id assigned in the "Add a new employee" feature. The unique id will be given by the system. Admin will assign a new role such as employee, manager, EMPLOYEE, and admin to the new created user. |

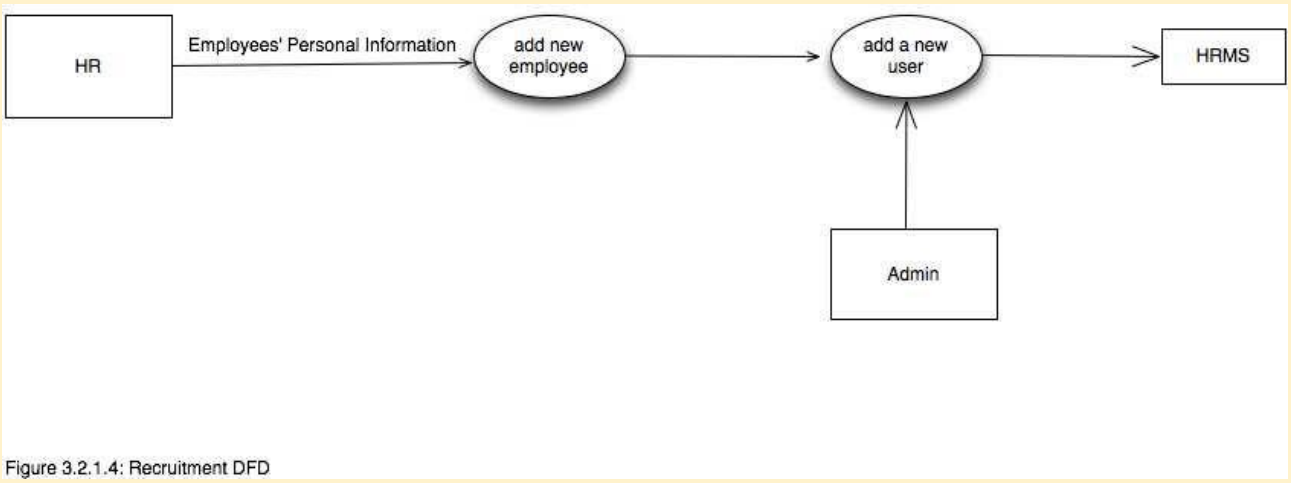



Figure 3.2.1.4: Recruitment DFD

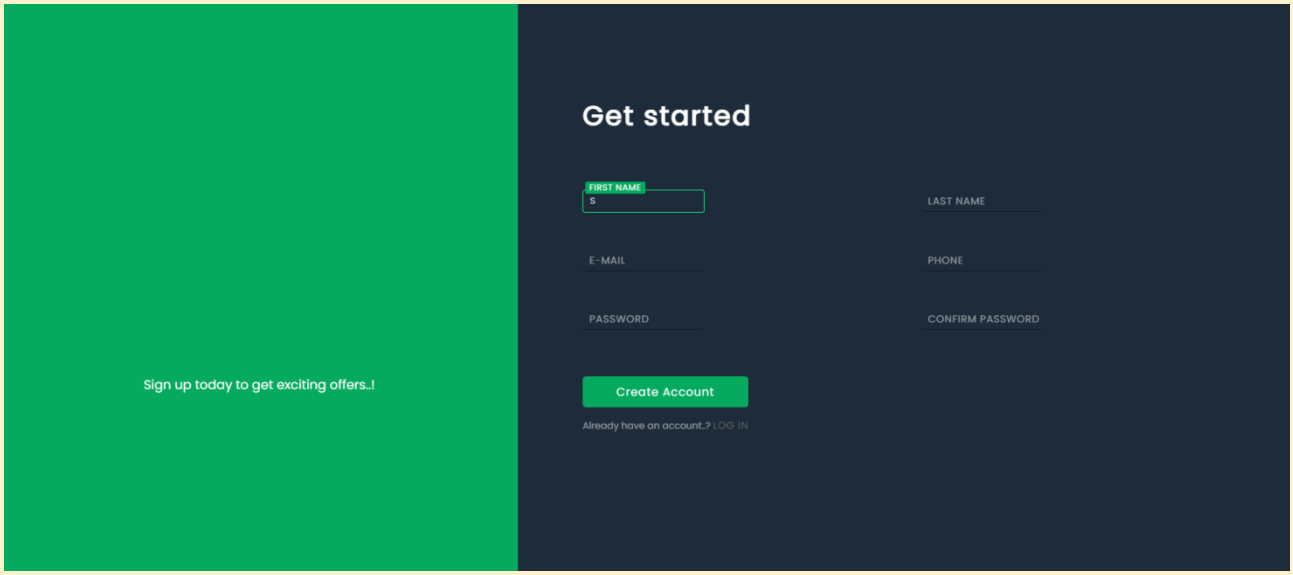
## *3.3 Non-functional Requirements*

### 3.3.1 Performance requirements

**3.3.1.1)** The number of the online user of the ERMS can be estimated as 50 at most.

**3.3.1.2)** There is no restriction on the number of the users to be added to the database.

### 3.3.2 Design constraints



### 3.3.2.1 Hardware Requirements

 The ERMS application will be storing employees' personal data. Roughly 50000 MB of storage capacity is needed.

### 3.3.2.2 Software Requirements

- Since ERMS application is a web-based application, internet connection must be established.
- The ERMS software will be used on PCs and will function via internet or interanet in any web browser.
- The ERMS application interface will be developed by Django framework.s  The ERMS software will support Django environment.
- The ERMS software personal database model will support SQLite environment as DBMS.
- Application will run on 256MB or higher of RAM.

## 3.3.2.3 Development Environment Requirments

This section outlines the necessary development environment requirements for setting up and developing the automated recruitment system using Django. These requirements are essential for the successful development and testing of the application.

### 2.3.1 Python and Virtual Environment

The development environment must have Python installed, preferably version 3.6 or higher. A virtual environment should be created and activated for isolating project-specific dependencies.

### 2.3.2 Django Framework

The Django web framework is fundamental for building the recruitment system. Install Django within the project's virtual environment using the following command:

bash

pip install django

### 2.3.3 Database System

Choose and install a database system compatible with Django. Common options include PostgreSQL, MySQL, and SQLite. Install the corresponding Python packages and configure the database connection settings.

### 2.3.5 Version Control System

A version control system is essential for tracking code changes and collaborating effectively. Install Git and establish a repository for the project.

### 2.3.6 Integrated Development Environment (IDE) or Text Editor

Select a suitable IDE or text editor for coding. Options include Visual Studio Code, PyCharm, Sublime Text, and Atom.

### 2.3.7 Dependencies Management

Maintain a requirements.txt file to manage project dependencies. This file should be generated using the following command:

bash

Copy code

pip freeze > requirements.txt

### 2.3.8 Static Files and Media Handling

Plan for managing static files (e.g., CSS, JavaScript) and media uploads. Leverage Django's built-in mechanisms, including the collectstatic management command and relevant settings.

### 2.3.9 Application Design

Create a structured plan for the Django app's architecture. Consider breaking down functionalities into separate apps within the project.

#### 2.3.10 Authentication and Security

Detail how user authentication and security will be implemented. Utilize Django's authentication features while customizing them as necessary.

#### 2.3.11 Frontend Framework (Optional)

If integrating a frontend framework such as React, Vue.js, or Angular, outline the integration strategy and dependencies.

#### 2.3.12 Testing

Define the approach for testing the application, including unit tests and integration tests. Utilize Django's testing framework to ensure code quality.

#### 2.3.13 Deployment Strategy

Although not directly related to the development environment, consider outlining how the application will be deployed. Explore deployment options such as AWS, Heroku, DigitalOcean, etc.

# 4. Data Model and Description

This section describes attributes of database objects and relationship between them with a data table dictionary and tables to overcome confusions. These data objects are made under the consideration of getting rid of unnecessary attributes and normalization factors. The ERMS application consists of two main database groups. First one is storing information for Personal Master Data module and the second one is for Authorization and Authentication Module.

## *4.1 Data Description of Personal Master Data Module*

In this section we mainly describe each table of the Personal Master Data Module in details. We determine each table and its responsibility in the module. Each table keeps many fields related to the specific data object. Then in the following sections we will explain the relationships of each database module table with each other.

- TBL_Employee
- TBL_APPL_User
- TBL_Employee_Projects
- TBL_Employee_TechnicalSkills
- TBL_Employee_Educatioin ➢ TBL_Employee_Languages
- TBL_Employee_Trainings
- TBL_Employee_Certificates
- TBL_Employee_Attachments

### 4.1.1 Data objects

### 4.1.1.1 TBL_Employee Table

| Name: | **TBL_Employee Table** |
|---|---|
| Description: | Data table for Employee consists of many fields which are responsible for storing the specific employee personal information. Some important fields in the table can be said as employee id, name, surname, email, phone, date of birth, and many more which can be seen in the above database table. Fields in this table will be filled with the specific data of the Employee. At the time a new Employee is added to the ERMS system, these fields keeps the data manually provided by the EMPLOYEE. After the first creation of the Employee account, employee him/herself, manager, EMPLOYEE, and admin have authorization to modify this information. |

#### 4.1.1.2 TBL_APPL_User

| Name: | **TBL_APPL_User** |
|---|---|
| Description: | The TBL_APPL_User table consists of basic information about a user. The detailed information about this table will be given in part (4.2), in Authorization and Authentication module part. |

#### 4.1.1.4 TBL_Employee_TechnicalSkills

| Name: | **TBL_Employee_TechnicalSkills** |
|---|---|
| Description: | This table stores information about an employee technical skills. Information such as employee Id, name, level, confirmation, notes, and levelId will be kept. |

#### 4.1.1.5 TBL_Employee_Educatioin

| Name: | **TBL_Employee_Educatioin** |
|---|---|
| Description: | This table keeps information about an employee education. Basically, information such as employee id, department, degree, institution, thesis, notes, still studying, entrance date, graduation, and etc. |

#### 4.1.1.6 TBL_Employee_Languages

| Name: | **TBL_Employee_Languages** |
|---|---|
| Description: | Employee languages table stores data fields such as employee id, languages, written and speaking level of the languages, native language, notes, and confirmation. |

#### 4.1.1.7 TBL_Employee_Trainings

| Name: | **TBL_Employee_Trainings** |
|---|---|
| Description: | This table gives information about an employee trainings. It keeps employee id, training name, location, duration, provider, confirmation, star and end dates, and notes. |

#### 4.1.1.8 TBL_Employee_Certificates

| Name: | **TBL_Employee_Certificates** |
|---|---|
| Description: | An employee may have certificates tEmployeeoughout his/her education timeline or he/she will get any other certificates during his working duration in the company. Employee certificate table stores data field such as employee id, name of the certificate, its provider, date, notes and confirmation. |

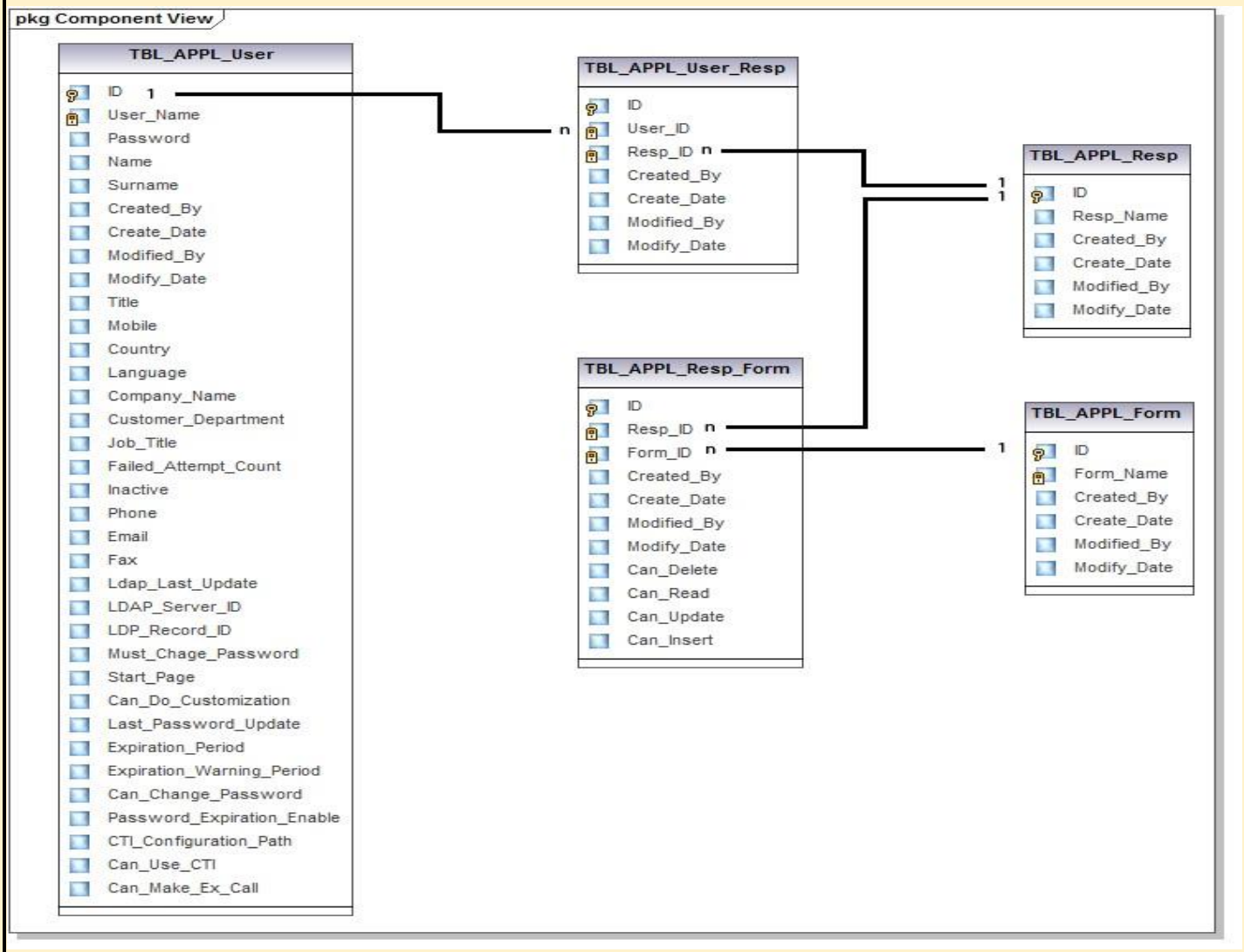
### 4.1.2 Relationships

Relationships between tables are made Employe enough a unique field which is employee id. Employee table creates the basic information table about an employee such as his/her id, name, surname, email, and etc. This provided employee id is a primary key for the employee table, but it will be a foreign key for all other tables using the specific employee id. All table mentioned above will manipulate employee personal that id. Therefore, employee id, the primary key, is like bridge between a unique employee and his/her relation to other data fields and tables.

## *4.2 Data Description of Authentication and Authorization Module*

In this section all authentication and authorization module database tables will be described in details. We describe each table and its responsibility in the module. Each authentication and authorization database tables keep many fields related to the specific data object. Relationship between each table will be explained as well.

- TBL_APPL_User
- TBL_APPL_User_Resp
- TBL_APPL_Resp
- TBL_APPL_Resp_Form ➢ TBL_APPL_Form




### 4.2.1 Data Objects

#### 4.2.1.1 TBL_APPL_User

| Name: | **TBL_APPL_User** |
|---|---|
| Description: | User application table is used for determining user information. When a user logs into the ERMS, user's username and password is checked from this table and directed to the system. Also it contains other data fields such as user id, name, surname, when this user is created and modified by whom, title of the user, password history, phone, email, fax, and many more. |

#### 4.2.1.2 TBL_APPL_User_Resp

| Name: | **TBL_APPL_User_Resp** |
|---|---|

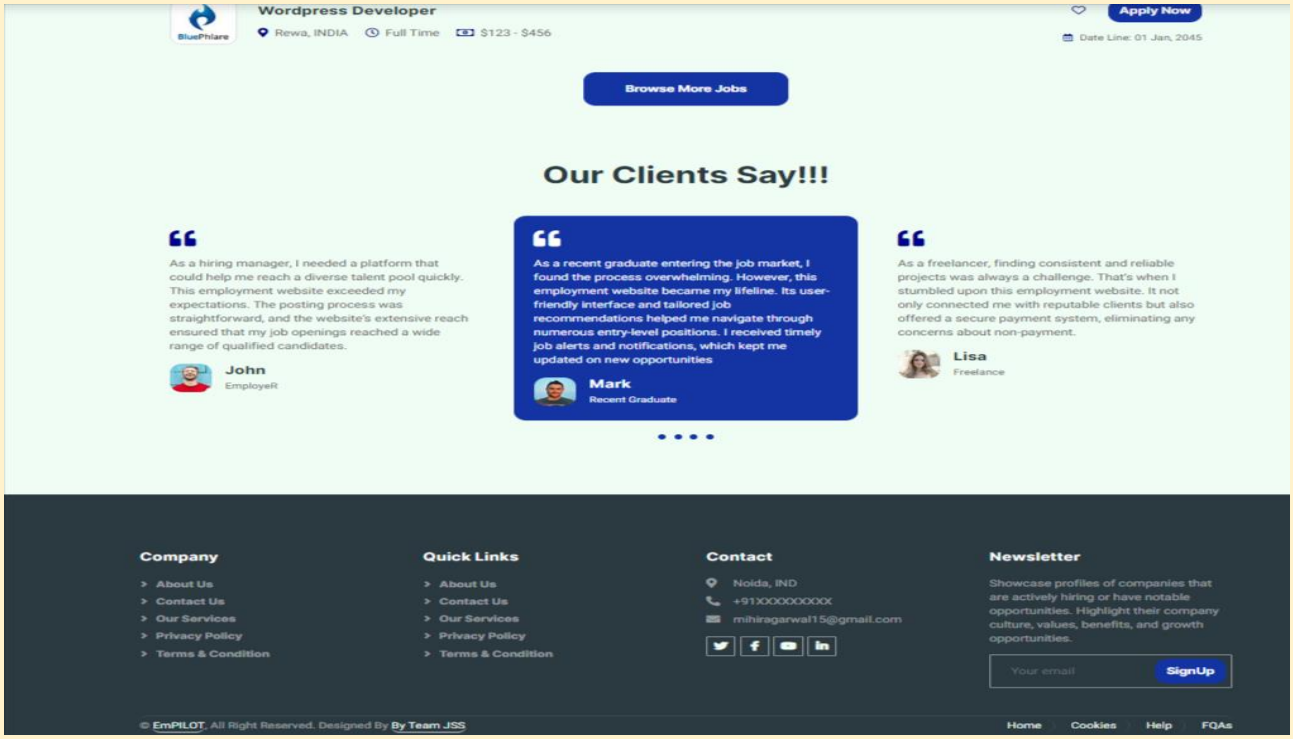
| | |
|---|---|
| Description: | This table is basically used to determine the assigned role types for a user. It also keeps creation and modification dates, and id of the users who created and modified these role assignments. |

### 4.2.1.4 TBL_APPL _Form

| | |
|---|---|
| Name: | **TBL_APPL_Form** |
| Description: | This table stores a list of functions that are used in the application. The basic data fields are function id, function name, creator, and modifier. |

### 4.2.1.5 TBL_APPL_Resp _Form

| | |
|---|---|
| Name: | **TBL_APPL_Resp _Form** |
| Description: | This is table is used as a bridge between **TBL_APPL_Resp** and **TBL_APPL_Form.** By means of this table we can determine which role types are authorized to which sets of functions. That is why when a user is logged in into the ERMS; according to the user role type the interface will be displayed. |



### 4.2.2 Relationships

The relationship between these tables is not prone to a single id. Nevertheless, there exists a process of checking a user at each step and determine his responsibility and authority. By the help of user id, user responsibility will be checked and interface will displayed according to the list of roles or responsibilities. After being checked the role types, related authorized functions will be called in the system.

# 5 Behavioral Model and Description

## *5.1 Description for software behavior*

Each employee will be added to the system by entering the personal data of an employee by EMPLOYEE. At the creating state, ERMS will give an id to the newly added employee.

After being added by EMPLOYEE, each employee has a right to be introduced to the system by admin. At this state, admin determines the role type of the user and this user is given a password automatically.

At the entering state of the system, each user sees the same interface which asks a username and a password.

After entering state, each user sees the same interface with different tabs related to their role type/s (manager, EMPLOYEE, admin, employee). At this stage:

- Managers can
  - Search employees under his responsibility or search all the users he/she desires, i.e a manager can search rows of the database.
  - Edit the employees' data who are under his responsibility.
  - Get report of information related to the employees he/she desires to learn, i.e. a manager can search columns of the database.
- Employees can
  - Edit their personal data
  - see their data in editable form.
- EMPLOYEE can
  - Search employees who has the properties he/she desires.
  - Get report about an employee, or employees.
  - Add employee by entering some specific personal data of that employee.
  - Edit employees' data

Administrator can

- Edit any user' s (manager, admin, EMPLOYEE, employee) data
- Add user, in other words create a valid account to login.
- Edit role properties, for example there can be a need to change the attributes of each role type such that the type of tabs each user sees may change. For that kind of situation there may be need to change the properties of each role type.

- Add a new role type to the system. Since, our ERMS is going to be a generic program there can be an addition to the role types and our system will perfectly welcome that kind of need.
- Search the users who have the desired capabilities or properties specified.

## *5.2 State Transition Diagram*

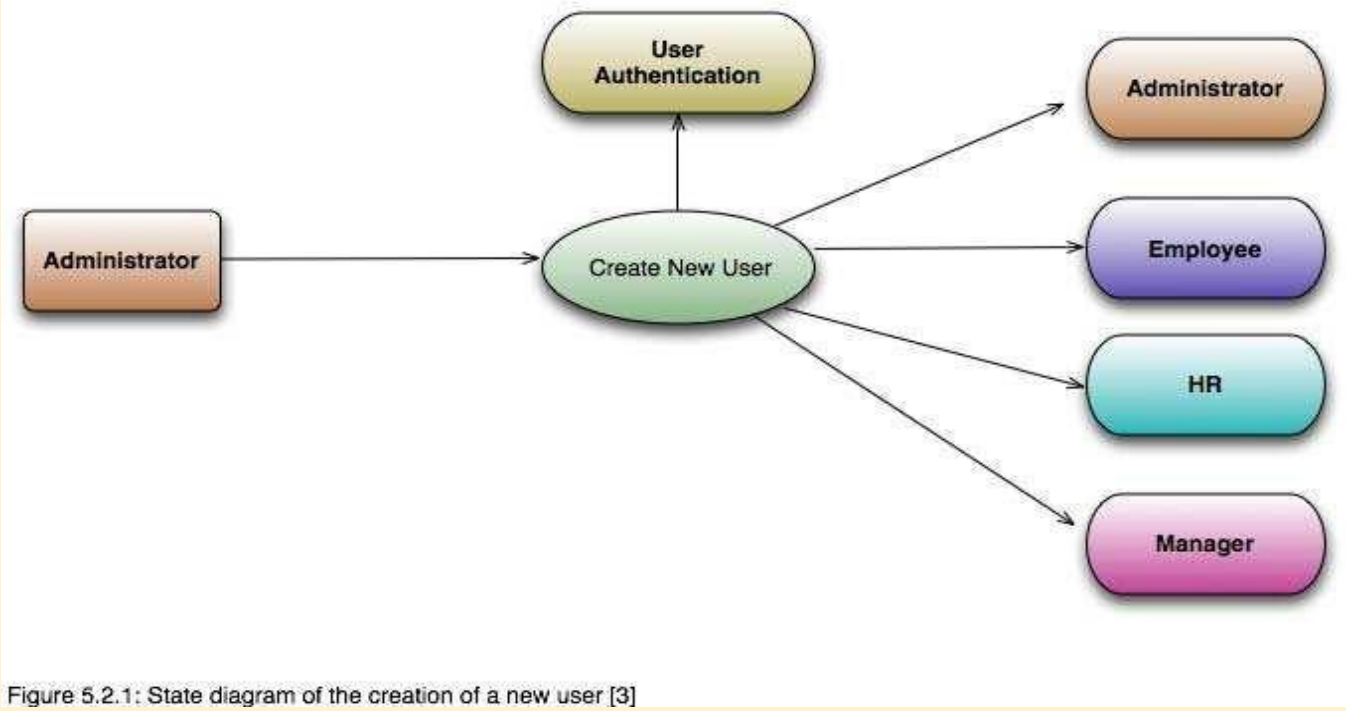



Figure 5.2.1: State diagram of the creation of a new user [3]

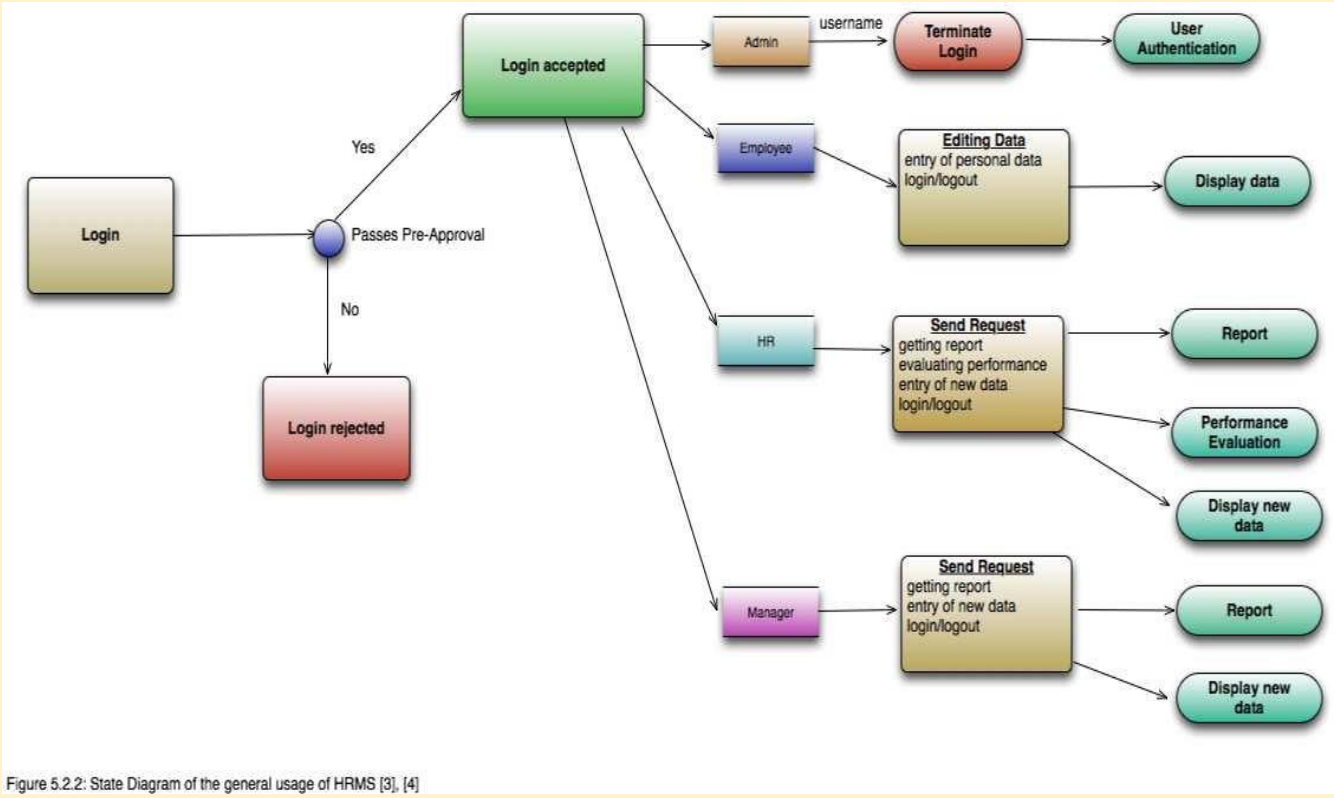


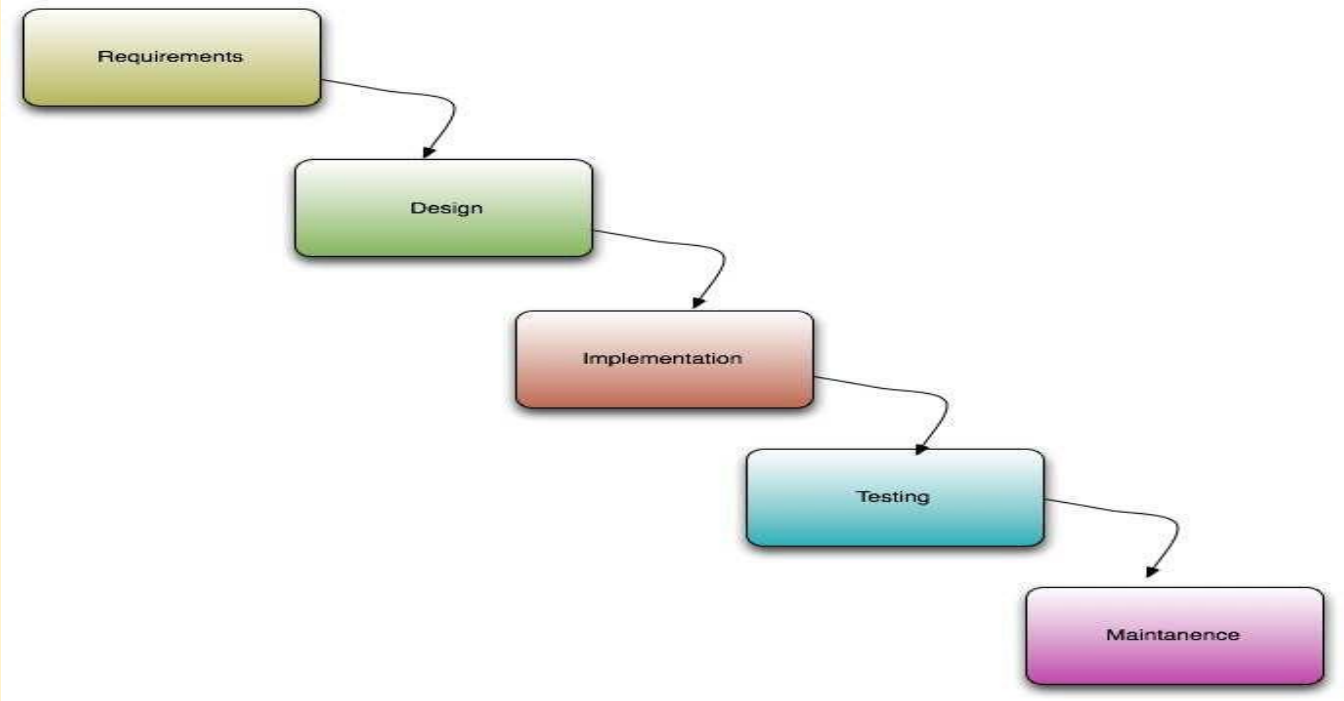

Figure 5.2.2: State Diagram of the general usage of HRMS [3], [4]

# 6 Planning

## *6.1 Team Structure*

We JSS Team arranged meetings to accomplish the task of writing SRS. In each time we met we arranged the tasks randomly. In other words all of us worked on both design and documentation.

## *6.3Process Model*



## 7 Conclusion

This document states the design level approach taken by the JSS Team for the project ERMS. After giving a basic information about what the ERMS is, the document briefly describes the problem and the solution we proposed to the problem with the figures to visualize better and steps taken to solve the problem. In other words this document introduces the technical details of the ERMS.

In the first part of the technical design, the major functions needed to develop an ERMS are introduced. Later on, these major functions and their sub-functions are visualized with the use case diagrams. In the second part, user interfaces are described in a detailed manner with figures. Lastly, data modules and their relationships are discussed.

To conclude, this document constitutes a base for the development of an ERMS.

**THANKYOU**