VIT-TECHZOIDS

RFID BASED EMPLOYEE ATTENDANCE SYSTEM

Software Requirements Specification

Version 1.0

Team Guide: Kumar K.

Members: Anant Gupta, Dheeraj Joshi, Jitin Kohli

College Name: Vellore Institute of Technology,Vellore

Department: B.Tech Computer Science and Engg.

State: Tamil Nadu

Table of Contents

Description

Software Requirements Specification

1.0 Introduction

1.1 Purpose:

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to consumers. In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality.

1.2 Scope:

RFID Based Employee Attendance System makes use of the RFID detection system to calculate attendance of employees in an organization and do further calculations of their salaries based on it. This software has been designed to reduce the labor of manual attendance for employees in the organization. It also aims at building accuracy in taking attendance electronically and thus reducing human error so that the salaries are also calculated accurately.

The cost of implementation of the system is only one time as the RFID chips are light and cheap containing information up to a few binary digits. The chip is rewritable and so is the information that is stored in the database corresponding to the chip code.

The software system also gives a personalized working environment for each and every employee to track the working of the system, for social networking, and as an event reminder.

1.3 Definitions.Abbreviations and Acronyms:

- RFID – Radio Frequency Identifier
- SRS - Software Requirements Specification, a document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.

- HTML (Hyper Text Markup Language): It is used to create static web pages.

- JSP (Java Server Pages): It is used to create dynamic web content.

- J2EE (Java 2 Enterprise Edition): It is a programming platform, belonging to the Java platform, which is used for developing and running distributed java applications.

- WASCE (WebSphere Application Server Community Edition): It is an application server that runs and supports the J2EE and the web service applications.

- WSAD (WebSphere Studio Application Developer): It is a designer toolkit which is designed to develop more complex projects by providing a complete dynamic web service.

- DB2 (IBM Database 2): It is a database management system that provides a flexible and efficient database platform to raise a strong "on demand" business applications.

- HTTP (Hyper Text Transfer Protocol): It is a transaction oriented client/ server protocol between a web browser and a web server.

1.4 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.
IEEE Computer Society, 1998.

1.5 Technologies to be used:

1.5 Technologies to be used:

J2EE: (Servlet, JSP, JAXP, Java Beans) Application architecture.

JAVA: Application architecture.

WASCE: (WebSphere Application Server Community Edition) Web Server

DB2: IBM Database.

Ajax: Asynchronous Java Script and XML.

XML: Extension Markup Language. **Web 2.0:** RSS Feed 2.0.

RAD 7.0: Development tool.

Localization: English

1.6 Overview of Document:

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product.
Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed.
Section 3 gives specific requirements which the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given.

2. Overall Description

2.1 Product Perspective:

RFID Based Employee Attendance System is a web application that is being built as a part of IBM the Great Mind Challenge. This system would detect the presence of employees in the organization premises, from a RFID chip carried along by them, by a RFID detector and calculate their attendance and generate reports based on that.

This document covers all the functioning and features of the system right from the functional requirements and how they are implemented to the non-functional requirements, extra features and shortcomings possessed by the system.

2.2 Software Interface

Front End Client:

Web Server:

Data Base Server:
Back End:

2.3 Hardware Interface

Client Side:

Server Side:

2.4 Product functions:

- There are four basic users: a) Administrator b) Employee c) Operator d) Visitor.
- Every user of the software should be an employee of the organization (except for the visitor).
- The presences of the employee within the organization premises is calculated by detecting the code in the chip carried by them and thus calculate their working hours.
- Employees can view their attendance date and month wise and also their salaries calculated according to it.
- The users can also search for the employees' information stored in the organization database. The accessibility of information varies according to the type of user.
- This software is also facilitated for interaction between the administrator and employee via e-mail.
- The concerned authorities get a generated report of the attendance and salary of the employees after a uniform time interval.
- The software also caters to outsiders.

2.5 User characteristics:

- **Admin** – System administrator, who looks after the system and has access to all rights and permissions in the system. He is the back-end user of the system.
- **Employee** – Actor carrying the RFID code and a member of the organization. He is the front-end user of the system.
- **Visitor** – An outsider in the system who has very limited access to it. He is not a part of the organization.

2.6 Constraints :

The system shall include multiple members so conflict of interest may arose.
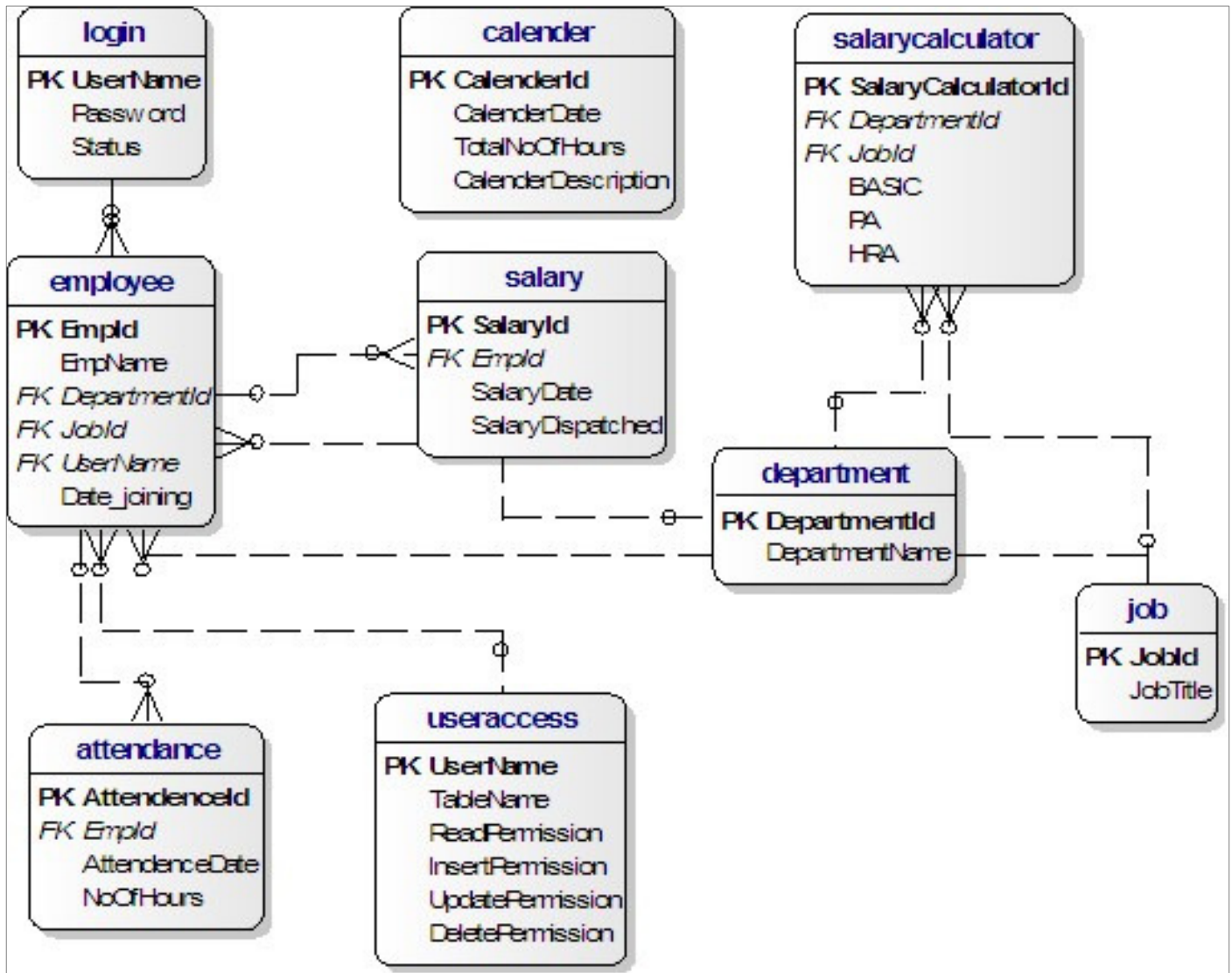
2.7 Use Case Diagram :
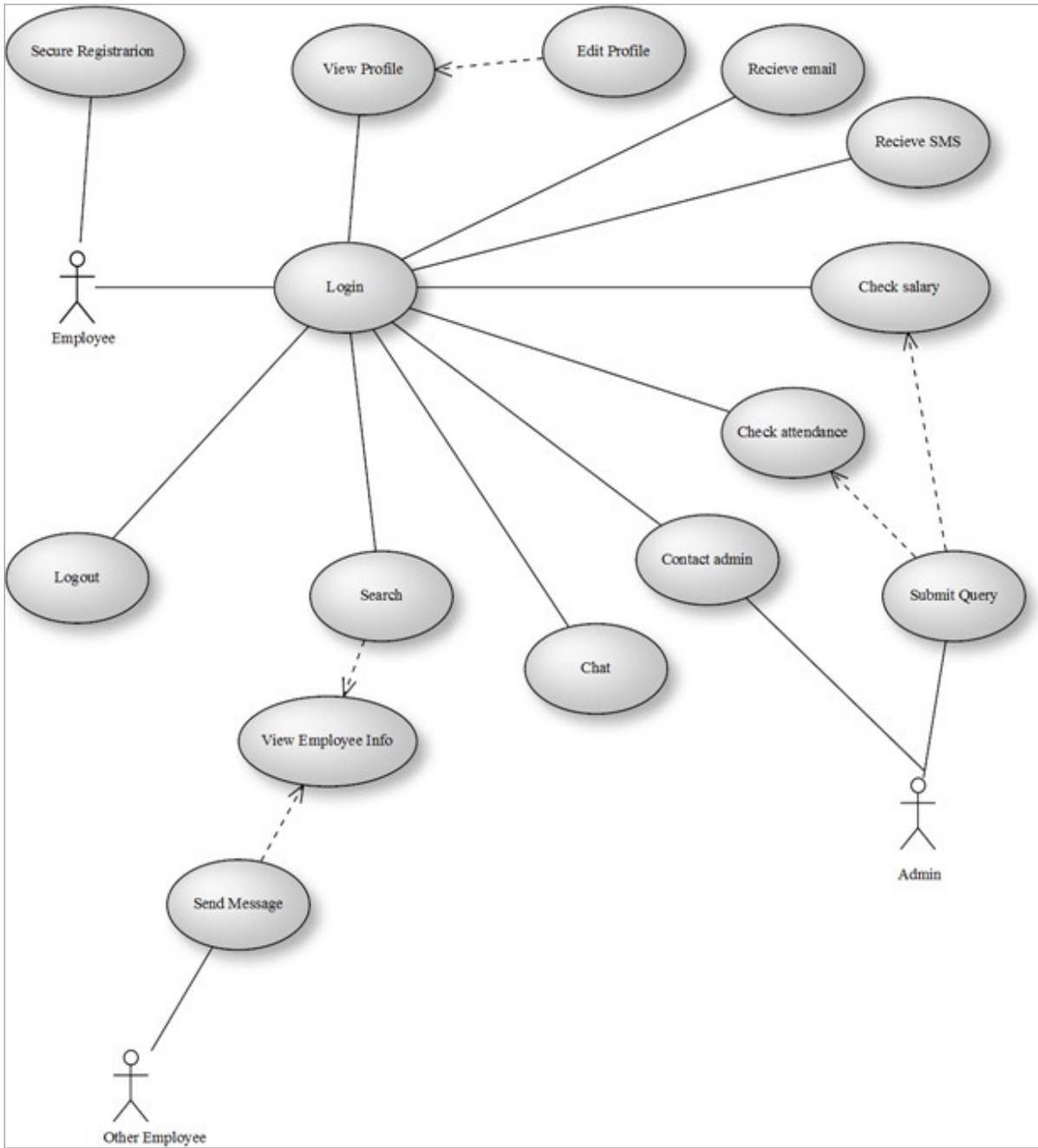
2.8 Sequence Diagrams:

2.8.1 Database Design :

2.8.2 Entity Relationship Diagram:
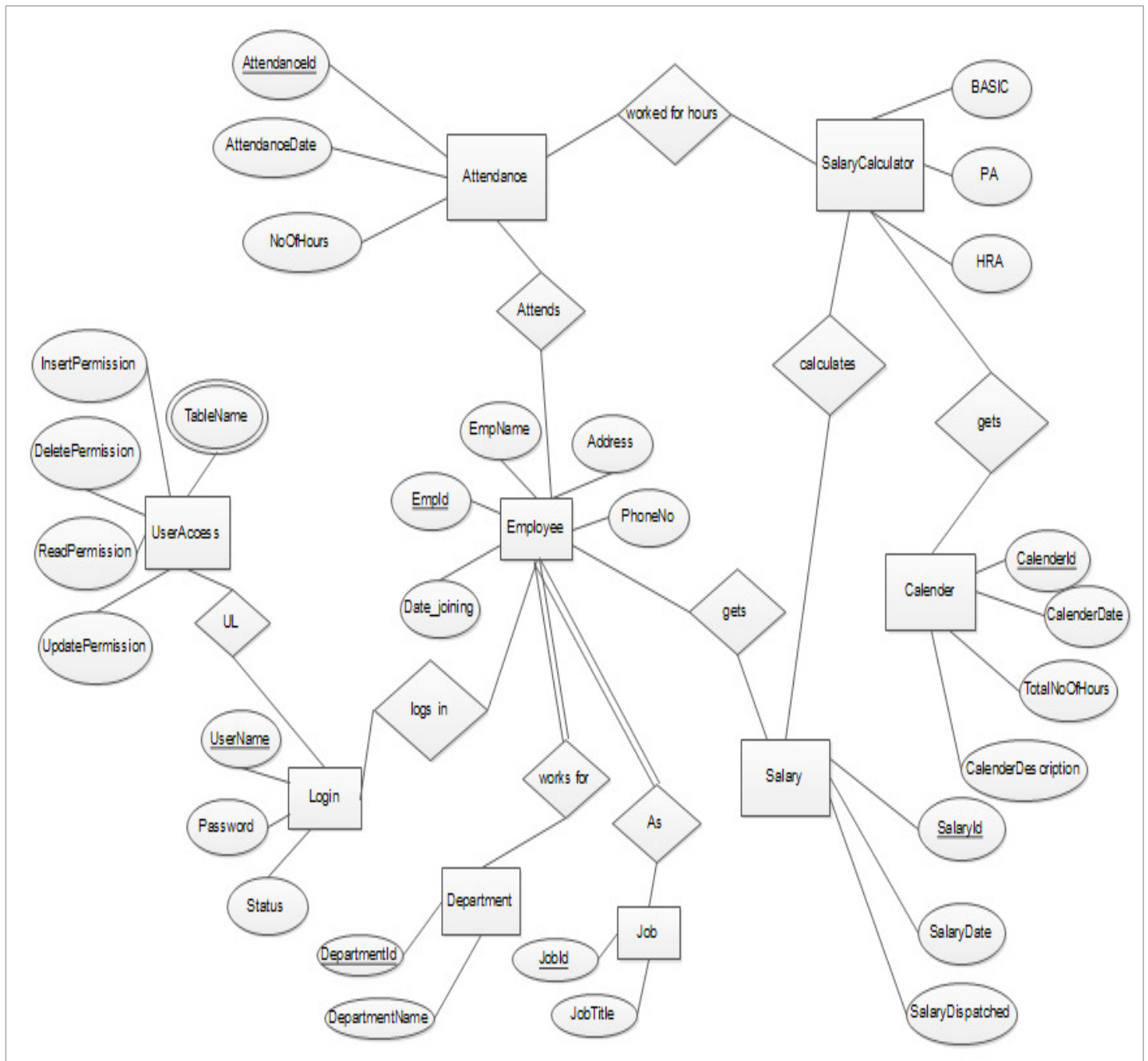
2.9 Assumptions and dependencies :

- All members of the organization have a unique employee ID and a pre-assigned unique code for first time registration.
- The authentication system is fed with the data to identify the admin.
- There is a salary dispatch system in the organization.

3. Specific Requirements

3.1 Use Case Models

I. Employee Use Case
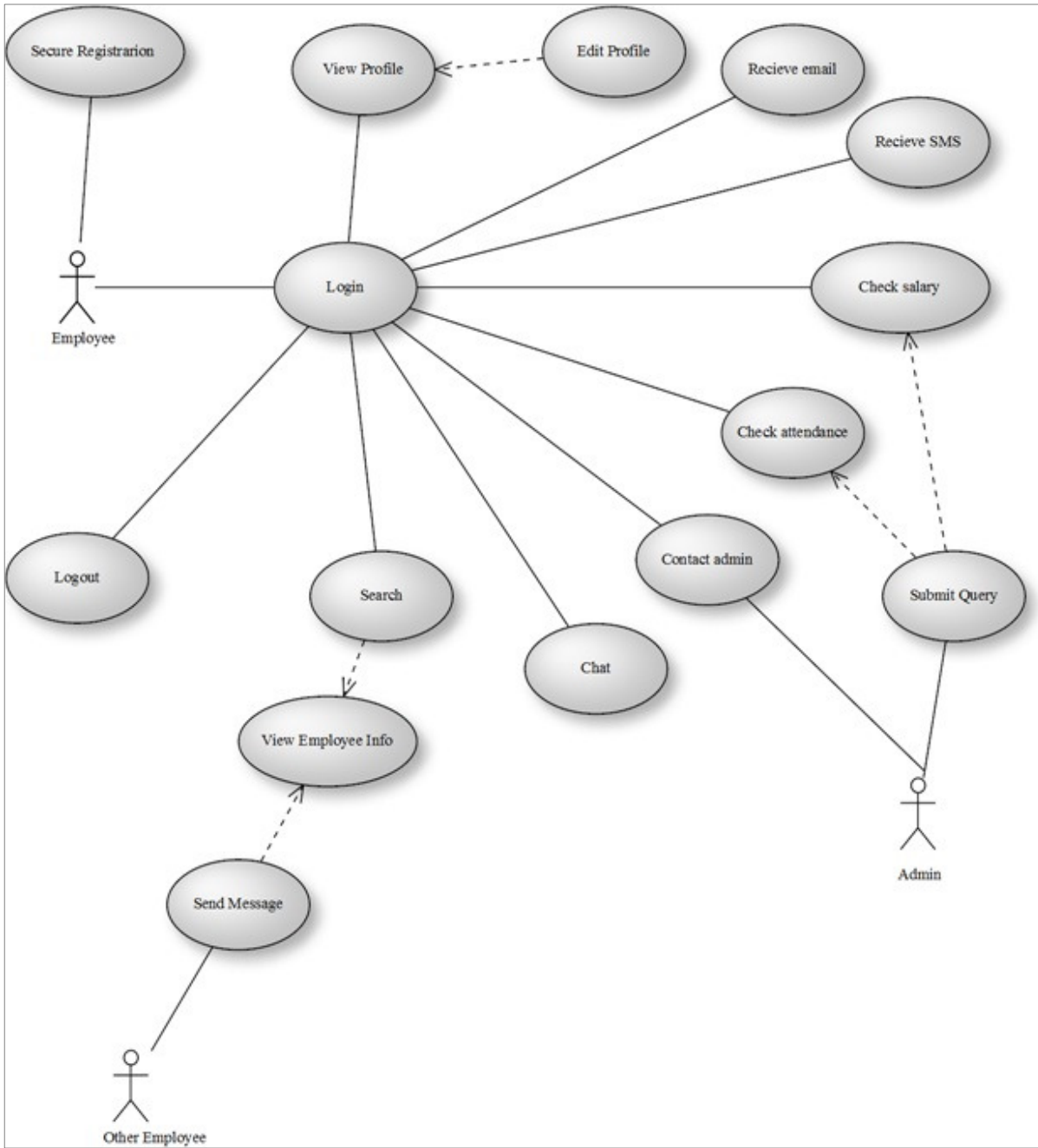
Use Case	Description
Secure Registration	The employee has to do one time registration into the system

|	
Login	The employee has to enter his credentials every time he wants to work on the system
View Profile	The employee has view his updated information
Edit Profile	The employee can add more information to his profile
Search	The employee can search for his colleague based on keywords
Colleague information	The employee can view the permitted information of the colleague he has searched for
Send message	The employee can send a direct email to this colleague
Contact admin	The employee can send a direct email to the admin of the system, visible by all admins
Check attendance	The employee can check for his attendance whenever the report is generated
Check salary	The employee can check how much salary he has received after certain periods
Submit Query	The employee can submit a query in case he feels there is some error in the generated report
Receive email	The employee receives email on any holidays, issue of salary and attendance report
Receive SMS	The employee receives a SMS on any holidays, issue of salary and attendance report
Chat	The employee can chat with any other users who are using this feature
Logout	The employee can logout of the session

II. Admin Use Case model

Use case	Description

Secure Registration	The admin has to do one time registration into the system
Login	The admin has to enter his credentials every time he wants to work on the system
View Profile	The admin has view his updated information
Edit Profile	The admin can add more information to his profile
Search	The admin can search for all users based on keywords
Colleague information	The admin can view all the information of the colleague he has searched for
Send message	The admin can send a direct email to this user of the system
Review Report	The admin handles automatically generated reports
Export	The admin can store and send the reports in different formats
Authenticate	The admin has to pass special authentication in order to perform some extremely authorized tasks
Check query	The admin looks into queries submitted by employees
Edit report	The admin can make changes in the generated report
Setup hierarchy	The admin has to set up the hierarchy the organization and roles of users
Setup calendar	The admin has to enter holiday list, number of working hours and salary calculation method
Contact employee	The admin has to revert back to any direct emails sent by any employee or in any other cases
Edit visitor page	The admin has to add and edit content of the visitor welcome page
Check visitor message	The admin has to check for feedback messages left by any visitor
Chat	The admin can chat with other users using this feature
Logout	The admin has to logout to end the session

III. Visitor Use Case model
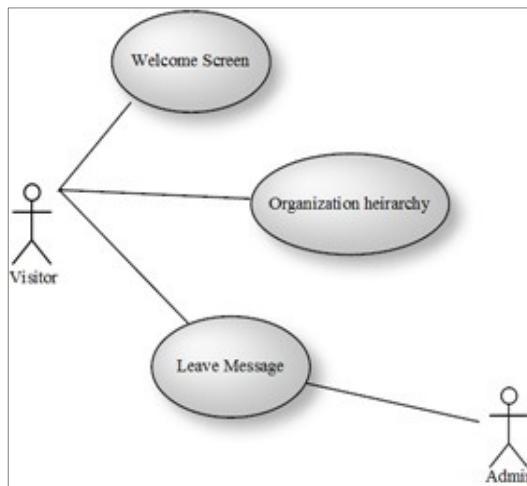
Use Case	Description
Welcome Screen	The visitor can view the welcome screen of the organization
Organization hierarchy	The visitor can view the hierarchy of the organization
Leave Message	The visitor can leave a feedback message

3.2 Functional Requirements

I. Secure registration

1. Description: When a new user enters his details for the first time a registration process commences to generate a new user id for him so as to interact with thy company work in the future.

2. Actor: Employee, Admin, Operator (front end actor).

3. Input: Employee id and unique secret code (for first time access).

4. Output: Employee account created with email id.

5. Flow of events:

Basic flow:

1) Employee opens registration page.

2) The user clicks 'login' button.

3) The login screen is opened.

4) The user enters employee id and unique secret code which was given at time of joining the company.

5) If the entry is correct then proceed.

6) Prompt for user for username and password. Create account.

7) User logs in again and fills his/her details.

8) Email account is accessible within the user login with a system generated welcome mail.

Alternate flow:

Employee id is wrong.

Secret code doesn't match.

If any of the two fields in the login page is left unfilled it will show invalid employee id and password.

If the employee id and the password are wrong it will display invalid user id and the password.

Precondition:

1. Login button is clicked.

2. The employee should have joined the company and received id and code.

Post condition:

Email id created with its home page displayed.

II. Search

1. Description: Information of all employees can be searched based on different criteria based on the type of keyword that is entered in the search box.

2. Actor: Employee, Admin, Operator (front end actor).

3. Input: A keyword, category of keyword (optional).

4. Output: List of all employees related to the keyword, and their information.

5. Flow of events

Basic Flow:

1) A keyword is entered in the search box.

2) The category/categories of the keyword are selected from a drop-down list (optional).

3) If a category (table column title) is selected, the keyword is compared to the entire content of all entries under that column(s).

4) If none of the categories are selected, the entire table content is compared with the keyword.

5) The comparison is non-case sensitive.

6) If the keyword matches with any entry, the whole row is selected and displayed.

7) The entry list is displayed in a sorted manner according to the number of occurrences of the keyword in the row.

Alternate Flow:

1) In step 1, the search box is left empty.

VIT-Techzoids / VIT University

2) An error message is displayed when the search button is clicked.

3) If the entered keyword does not match any entry, an error message is displayed.

Pre-condition:

1) The user using this function is already logged in.

2) The types of employee information to be displayed depend on the type of user using this function.

Post-condition:

1) The information can only be viewed, not edited.

III. Welcome Screen

I. Description: The visitor is greeted with a welcome page giving a brief introduction about the organization, the hierarchy and feedback facility.

II. Actor: Visitor

III. Input: Organization URL

IV. Output: Static page

V. Flow of events:

Basic flow:

1) The visitor navigates to the home page of the organization.

2) A page is displayed providing the basic information about the website.

3) The visitor goes to display organization hierarchy section.

4) The hierarchy of the organization is displayed, along with employees' information, belonging to the hierarchy.

5) The visitor goes to Contact Us section in the page.

6) A feedback message is left by the visitor.

Pre-condition:

1) The visitor cannot be an employee of the organization.

2) The welcome page and hierarchy structure may vary from time to time, as it is customizable by the admin.

Post-condition:

1) The feedback message is sent to admin via email.

IV. Receive automatic emails:

1. Description: Employees would receive emails regarding salaries, attendance and   holidays. The emails would be generated automatically.

2. Actor: Employee

3. Input: Open inbox.

4. Output: Emails.

5. Flow of events:

Basic Flow:

1) Whenever salaries are issued, the email sending client (application) retrieves salary information from the employee database.

2) It sends the information via an email template to the email address of the employee, given in the database.

3) The system checks the calendar for any holidays within the next 5 days. If any is found, it retrieved the holiday information from the calendar.

4) It is then sent via a template email for that to the email address of all employees, extracted from the database.

5) At the end of every month, the attendance database of each and every employee is checked and the total number of working hours is calculated.

6) This calculated information of each employee, is sent to that employee only via a template email to the respective email addresses.

Alternate Flow:

1) Any entry in the employee database is not present accidentally.

2) Service not being provided by the third party application, if it is used.

Pre-condition:

1) A template email should be present for each event with mail sending application.

2) The application should be able to retrieve data from various databases.

Post-condition:

1) Employees must not send a reply to the automatically generated email.

V. Receive SMSs:

1. Description: Employees would receive SMSs regarding absence for a day, on issuing of salaries and on holidays. The SMS would be generated automatically.

2. Actor: Employee.

3. Input: None.

4. Output: SMS on employees' phone.

5. Flow of events:

Basic Flow:

1) The SMS sending application interacts with email sending application and gets the amount of salary issued to each employee, when the email sending application extracts that information from the database.

2) It sends this amount information via a SMS template to the mobile number of the employee, given in the database.

3) The application checks the calendar each day for a holiday and extracts the information, if there is any holiday.

4) It is then sent via a SMS for that to the phone numbers of all employees, extracted from the database.

5) The application checks the database for the number of hours worked, at the end of each day.

6) If the figure is zero, it sends a SMS saying so.

Alternate Flow:

1) Any entry in the employee database is not present accidentally.

2) Service not being provided by the third party application, if it is used.

Pre-condition:

1) The application should interact with the email sending application, or they must both work under the same workspace.
2) The application should have access to employee database and calendar database.

Post-condition:

1) The employee must not reply to the SMS.

VI. View and Edit Profile Information:

1. Description: The employee can view his profile information and update this information whenever he wants.
2. Actor: Employee
3. Input: Entry relevant to the type of information (when updating).
4. Output: Employee information.
5. Flow of events:

Basic Flow:

1) The Employee goes to his profile after logging in.
2) He can see all the updated information about him that is stored in the database.
3) He clicks the edit button to add/remove/change current information.
4) All the non- editable entries become editable and changes are made.
5) On pressing the save button, the changes are saved in the same entry and old information is over-written.

Alternate Flow:

1) The Employee tries to edit his employee ID, which he cannot.
2) The Employee edits the old information, but doesn't press the save button, then the new information is not stored.

Pre-condition:

1) The Employee must be logged in.

Post-condition:

1) The employee cannot recover the previous information.
2) All functionalities of the system would use the new information.

VII. View Attendance and Salary

1. Description: The employee can view his attendance, every day, and salary, at the end of month.
2. Actor: Employee.
3. Input: Entries from attendance and salary database for each employee.
4. Output: Attendance and salary report.
5. Flow of events:

Basic Flow:

1) The employee goes to attendance panel within his account.

2) The attendance is in a table format with number of date, hours worked and total number of working hours for that day.

3) Attendance for a complete month is in a report format.

4) The employee goes to salary panel.

5) The salary received is shown month wise along with a brief description about its calculation method and other details, e.g. Bank account no. etc.

6) The employee can file a query in case he examines that either the attendance or salary or both are nor correct in the report.

Pre-condition:

1) Salary for the current month is not displayed.

2) Attendance report is generated only for a complete month.

3) The employee can view only his own attendance and salary.

Post-condition:

1) The employee must not file a complaint unnecessarily.

2) The reports are in a downloadable format.

VIII. Setup number of working hours and working days.

1. Description: The admin decides how many hours each employee should work per month to get the regular salary.

2. Actor: Admin

3. Input: Secure authentication

4. Output: Change in the calculation of salaries

5. Flow of events:

Basic Flow:

1) The admin authenticates through a secure verification.

2) Number of working hours is setup month wise.

3) This is further set up according to each day in a week, i.e., for each day of the week , number of hours is set up.

4) The criteria for calculation of salary is set up, in which only number of hours worked is to be calculated by the system.

5) Add/Edit a holiday in the calendar database.

Alternate Flow:

1) Secure authentication failed.

2) The number of hours in the month and the day wise broken down hours do not match.

3) A field is left blank.

Pre-condition:

1) Authentication should be secure.

Post-condition:

1) Total number of working hours per month is excluding holidays.

IX. Setup hierarchy of the organization:

1. Description: The admin sets up virtually how the organization is handled by different roles allotted in the hierarchy, including other admins as well.
2. Actor: Admin
3. Input: Secure Authentication
4. Output: Change in the structure of organization
5. Flow of events:

Basic Flow:
1) The admin authenticates through a secure verification.
2) He picks employees to be assigned special roles in the organization.
3) He can also make other users as admin.
4) He designs the hierarchy through a graphical tree structure.

Alternate Flow:
1) The secure authentication fails.
2) Inappropriate changes are made.
3) A field is left blank.

Pre-condition:
1) The authentication should be highly secure.

Post-condition:
1) The users having roles in the hierarchy don't come under the role of employee.
2) The setting up of hierarchy must be done in a proper and approved way.
3) Further changes can be made to the hierarchy.

3.3 External Interface Requirements

3.3.1 User Interfaces:

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

The users must have a basic knowledge on use of web forms, which will have a professional GUI. Each web form has AJAX implementation. The system can work in any Internet browser, provided it supports dynamic web pages, including AJAX .The pages are well designed and light to reduce load time. The forms and buttons are intended to be fast. Each page is as user-friendly as possible.

The admin must know how to read strategic data through graphical representations and implement on it. The admin must also be able to handle secure authentication system.

All web pages would have a help button which would lead to software documentation.

3.3.2 Software Specification

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

During the system development, assigning of both static and dynamic pages will be done. Creation of website functions and database systems along with interaction with third-party software.

The following are needed requirements:

Web browser- Mozilla Firefox
OS- Ubuntu
IDE for web page development- Eclipse
Database Server- DB2

3.4 Hardware Specification:

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

The system needs a close interaction with hardware of various kinds, since the detection of employee is done through a scanner. Also for secure authentication of admin, a biometric device is used.

RFID scanner

RFID chip

Biometric device

3.5 communication interface

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

The system shall use http/ftp protocol for communication over the internet.

4. Other functionalities

4.1 Security Requirements:

Secure access of confidential data (user's details). SSL can be used.
Not all data should be accessible to all the stake holders in the system.
Accessibility is predefined.

4.2 Availability Requirements:

24 X 7 availability
when the system is not required for full access i.e. not all the data and members are required, the system may be partially switched off for :
1) maintenance
2) save resources
3) allow the needed attributes to be accessible all the time without any fail
4) increase system efficiency

4.3 Performance Requirements:

Better component design to get better performance at peak time. Use of AJAX in web forms for fast response and feedback. Avoidance of heavy CSS content to make pages light and professional.

4.4 Software Quality Attributes:

Flexible service based architecture will be highly desirable for future extension.

4.5 Design constraints:

Software implementation is a tedious task.

4.6 Software system attributes:

The necessary qualities of software products are :

4.6.1 Security: The application is password protected and also any updation of new product entries and order processing is done by only privileged users. Not all st6akeholders can change data. Through proper verification and channel , data can be modified/added.

4.6.2 Maintainability: The application is to be designed so that it is easily maintained. Also it should allow incorporating new requirements in any module of system.
According to the type of institute the proper module can be designed based on pre=existing templates.
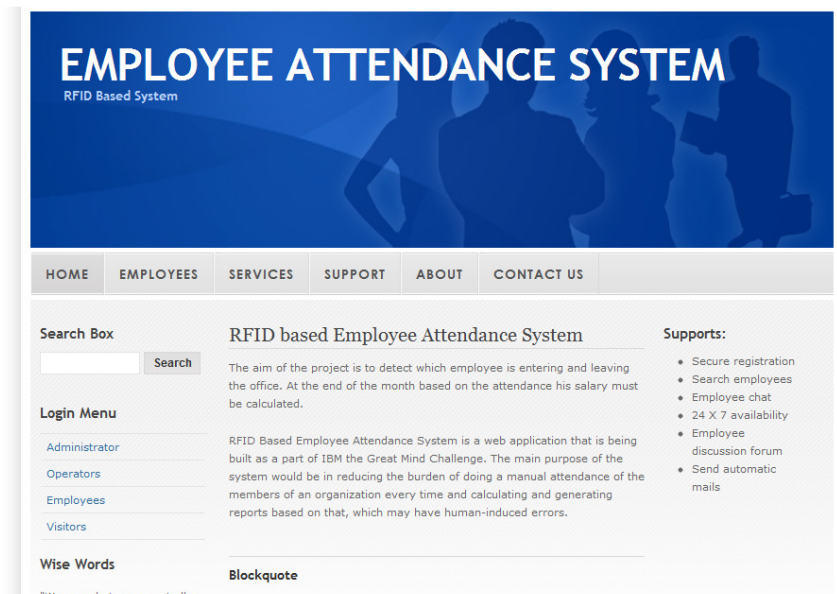when data is not required, the data can be switched off and thus system efficiency increased.

4.6.3 R e l i a b i l i t y
The system is reliable due to proper checks and balances provided through the entire process.

4 . 6 . 4 P o r t a b i l i t y : The application will be easily portable on any window based system.

Other requirenents

Screen Shots

App
end
ix

VI1

index