The Great Mind Challenge'12
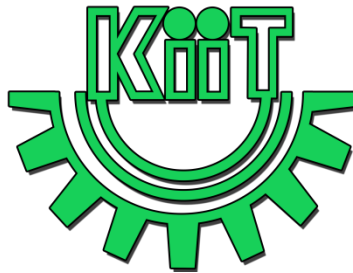### Initiate Collaborate Innovate

HR Operations Manager

Software Requirements Specification

KIIT University
Bhubaneswar - 751024

Team Members

Mayank K. Rastogi
Mayank Bajaj
Ayush Choubey
Manik Nandan

Project Guide

Mr. Prachet Bhuyan
Associate Professor
School of Computer Engineering

The Basilisks

Revision History

Date	Version	Description	Author
30-01-2013	1.0	Synopsis	The Basilisks

Table of Contents

1. Introduction:

2. Overall Description

3. Specific Requirements

1. Introduction

1.1 Purpose

In business environments filled with diverse forms of content, continuously changing and complex business processes, and an array of different line-of-business interfaces, companies are looking for ways to gain operational efficiencies, reduce risk, and improve quality through exception management and end-to-end process visibility.

The solution will be used to demonstrate how value can be delivered across any business process by linking business processes to relevant business content, and to the people that need to make business decisions.

1.2 Scope

- The system proposes a new HR hiring process.
- The hiring process is divided for applicants, HR heads, Managers and Interviewers. Each user will have their own privileges and cannot interfere in each other's work.
- The applicant will send his resume for a job opportunity to a particular email id, meant only for the hiring process.
- The system will process the applicant's email and forward it to the respective HR for shortlisting, or reject the application if it was received past the deadline.
- The HR will be able to view the applicants' resume and will shortlist them for interview.
- The Manager will schedule the interviews for the shortlisted applications and assign team members for the interviews.
- The interviewers will update the results after taking the interviews. The Manager will then review the results and select the applicants.
- The HR will then send offer letters to the selected applicants by email.
- The applicant will accept or reject the offer by clicking a link sent to him by email.
- A new Employee ID will be automatically generated if the applicant accepts the offer.

1.3 Definitions, Acronyms and Abbreviations

- **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.

- **RAD (Rational Application Developer for Websphere Software):** It helps Java developers to rapidly design, develop, assemble, test, profile and deploy high quality Java/Java EE, Portal, Web/Web 2.0, OSGi, Web services and SOA applications.

- **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.

- **UML (Unified Modeling Language):** It is a standard language for writing software blueprints. UML may be used to visualize, specify, construct and document.

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

- **XML (Extensible Markup Language):** It is a markup language that was designed to transport and store data.

- **Ajax (Asynchronous Java Script and XML):** It is a technique used in java script to create dynamic web pages.

- **Web 2.0:** It is commonly associated with web applications which facilitate interactive information sharing, interoperability, user-centered design and collaboration on the World Wide Web.

1.4 References

- Project specification requirement (provided by IBM).
- IBM Redbooks (http://www.redbooks.ibm.com)
- The Java EE 6 Tutorial (Oracle Corporation)

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 7.5:** Development Tool
- **Localization:** 2 Languages – English and Hindi

1.6 Overview

SRS includes two, sections overall description and specific requirements.

- Overall description will describe major role of the system components and interconnections.

- Specific requirements will describe the roles and functions of the actors.

2. Overall Description

2.1 Product Perspective

The web pages (XHTML/JSP) are present to provide the user interface on client side. Communication between client and server is provided through HTTP/HTTPS protocols.

The Client Software is to provide the user interface on system user client side and for this TCP/IP protocols are used.

On the server side web server is for JSP and database server is for storing the information.
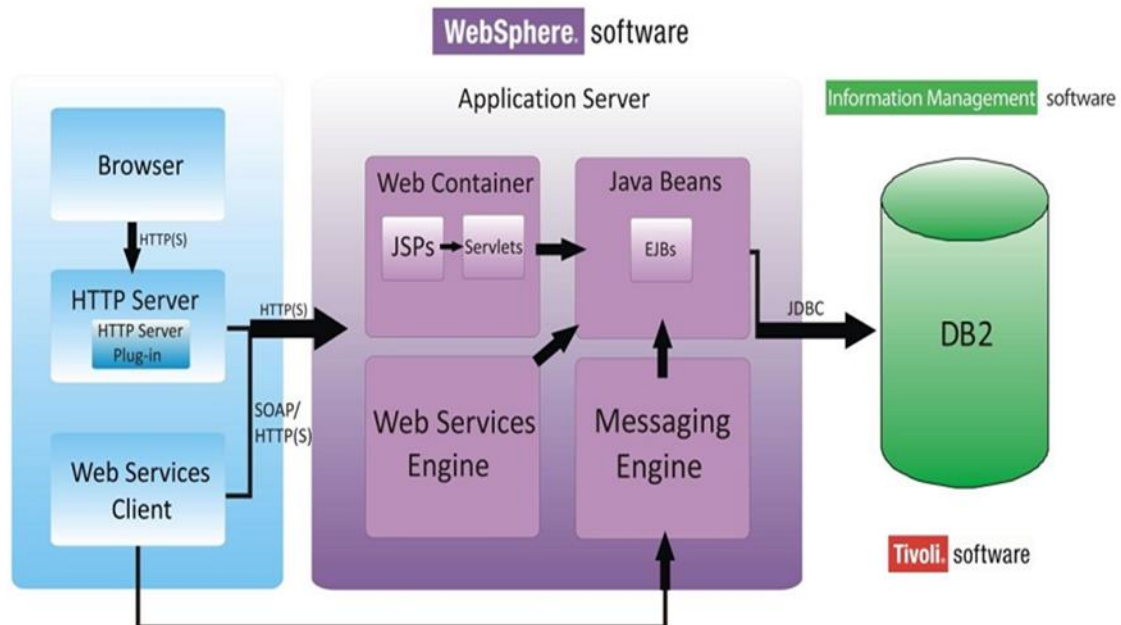
2.2 Software Interface

Front End:

- **Client on Internet:** Web Browser, Operating System (any)
- **Client on Intranet:** Web Browser, Operating System (any)

Back End:

- J2EE, Java, Servlets, AJAX, XML
- **Web Server:** WebSphere Application Server
- **Email Server:** Any (preferably Apache James)
- **Database Server:** IBM DB2
- **Operating System:** Any (preferably Linux)

2.3 Hardware Interface

Minimum Requirements:

Client Side			
	Processor	**RAM**	**Disk Space**
Internet Explorer - 6	Intel Pentium III or AMD-800 MHz	128 MB	100 MB

Server Side			
	Processor	**RAM**	**Disk Space**
RAD	Intel Pentium III or AMD 800 MHz	1 GB	3.5 GB
DB2 9		256 MB	500 MB (Excluding database size)

Recommended Requirements:

Client Side			
	Processor	**RAM**	**Disk Space**
Internet Explorer - 6	All Intel or AMD - 1 Hz	256 MB	100 MB

Server Side			
	Processor	**RAM**	**Disk Space**
RAD	All Intel or AMD - 2 Hz	2 GB	3.5 GB
DB2 9		512 MB	500 MB (Excluding database size)

2.4 Communication Interface

- Applicant on Internet will be using HTTP/HTTPS protocol.
- Employees on Intranet will be using HTTP/HTTPS protocol.

2.5 Constraints

- Applicant must mention the Opportunity Id in the subject field in correct format while sending his resume for proper processing of his application.

2.6 Product Functionalities

- **Administration**: Each employee will have his own privilege. The entire system is divided for the employees of the organization such as HR head, Manager, Interviewer, etc.

- **Security**: The employees are required to authorize themselves by giving their password, to login into the system.

- **Job Search:** Applicants will be able to search for a job opening and will be able to send their applications from the website itself.

- **Hiring Process:** Applicants will be able to send their resume by email. The applications will be shortlisted for interviews and then selected based on the interview result.

- **Online Applicant Search:** HR will be able to search for applicants having the desired skill sets and work experience, who have uploaded their resumes on popular job seeking sites like LinkedIn.

- **Notification:** Applicants will be notified at each step of the hiring process by email.

- **Employee Attendance:** Employee's attendance will be managed based on his login and logout activity on the system.

- **Private Messages:** Employees will be able to share information with each other instantly using private messages.

- **Employee Leave Application:** Employees will be able to request for leave. The supervisor will be able to grant or reject the leave application.

2.7 Database Design

2.7.1 E-R Diagram
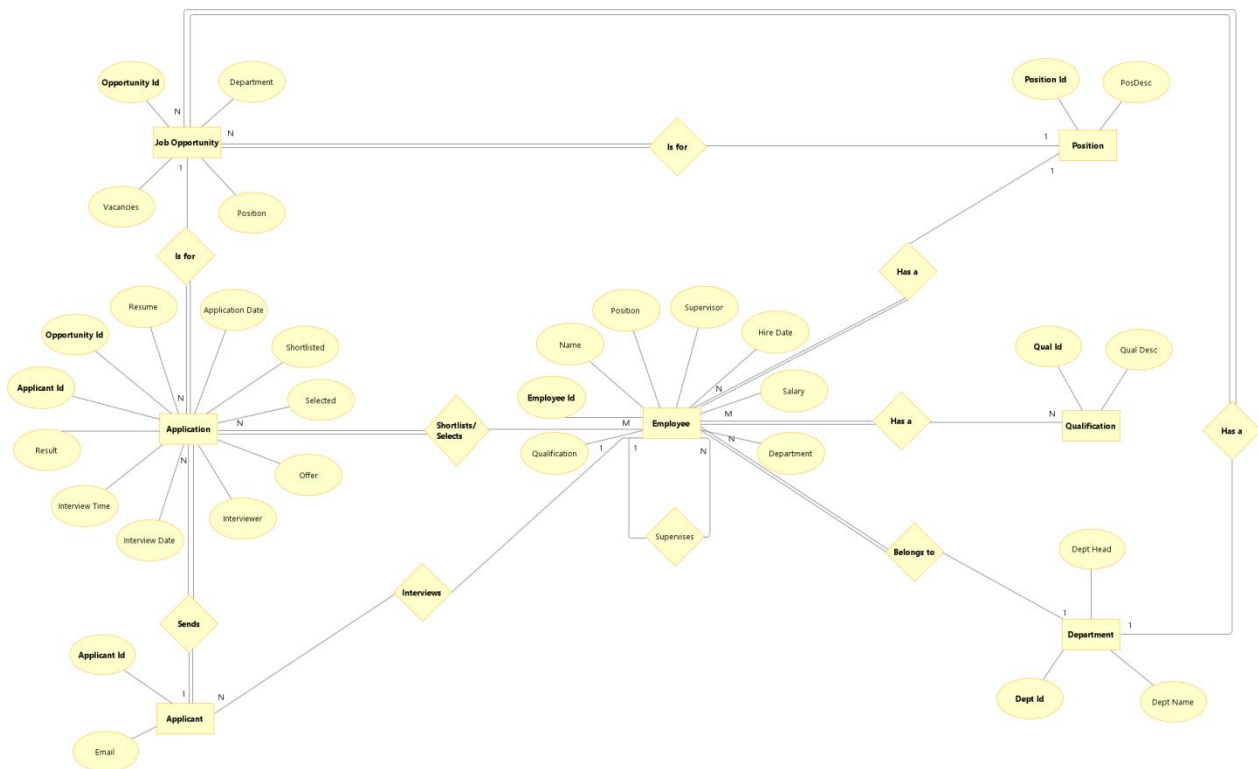
2.7.2 Schema Diagram

APPLICANT
- APPLICANT_ID
- EMAIL

DEPARTMENT
- DEPT_ID
- DEPT_NAME
- DEPT_HEAD

POSITION
- POSITION_ID
- POS_DESC

APPLICATION
- APPLICANT_ID [FK]
- OPPORTUNITY_ID [FK]
- RESUME
- APPLICATION_DATE
- SHORTLISTED
- SELECTED
- OFFER
- INTERVIEWER
- INTERVIEW_DATE
- INTERVIEW_TIME
- RESULT

EMPLOYEE
- EMPLOYEE_ID
- PASSWORD
- POSITION_ID [FK]
- SUPERVISOR
- HIREDATE
- SALARY
- DEPT_ID [FK]
- QUAL_ID [FK]

JOB_OPPORTUNITY
- OPPORTUNITY_ID
- DEPARTMENT_ID
- POSITION_ID
- NO_OF_VACANCIES

QUALIFICATION
- QUAL_ID
- QUAL_DESC

2.8 Use Case Model Survey
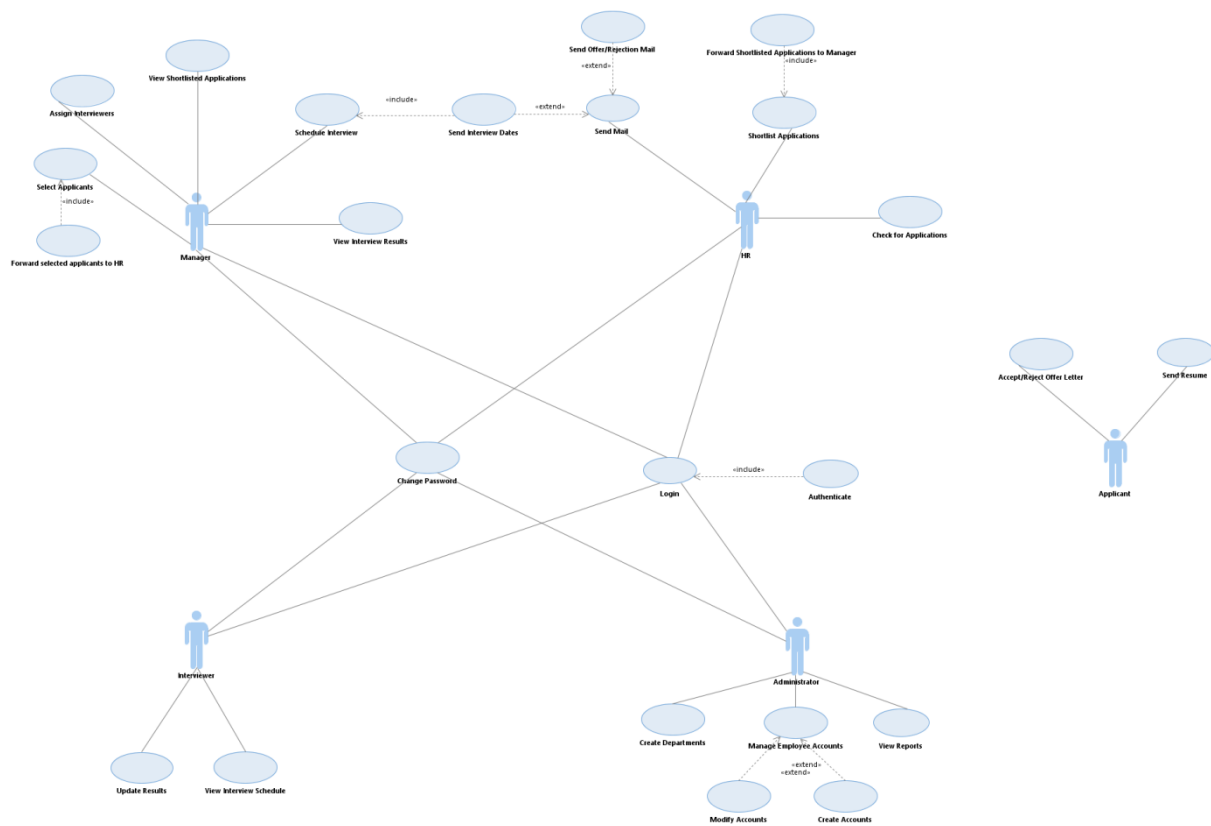
Applicant: An applicant's main job is to apply in the company by providing the job opportunity id. He/she can either send his/her resume by email to the email-id provided by the company HR. After the selection process, if they get selected, they will be sent a link in an email. The applicant will click on this link if he chooses to accept the offer.

Administrator: An administrator can login into the system and change his password. Apart from this, administrator's job will be to create new departments, manage employee accounts like promoting an employee or changing his department or creating new account, and to view various reports regarding the system.

Interviewer: An interviewer will have his own username and password to log into the system. The main job of interviewer is to view his schedule for various interviews and after the interview it's his job to update the results in database and using those results, the manager will select the employees.

Manager: Managers too will have their login id and password by which they will log into the system. His main job is to view the applications that are shortlisted by HR-Head and on the basis of number of shortlisted applicants; he will schedule the interview and appoint an interviewer for it. After interviewer updates the results, it's manager's job to select the potential employees while keeping in mind the number of vacancies.

HR-Head: HR-Head can log into his account and can perform various tasks. His main job is to view the applications and shortlist the applicants and thus completing first level of selection process. The shortlisted applications are forwarded to managers. Other than this, HR-Head will have to send the rejection mail stating the reason of rejection and job application.

2.9 Class Diagram

«Java Class»
HR
- checkApplications ()
- shortlistApplications ()
- sendMail ()

«Java Class»
Interviewer
- viewSchedule ()
- updateResults ()

«Java Class»
Manager
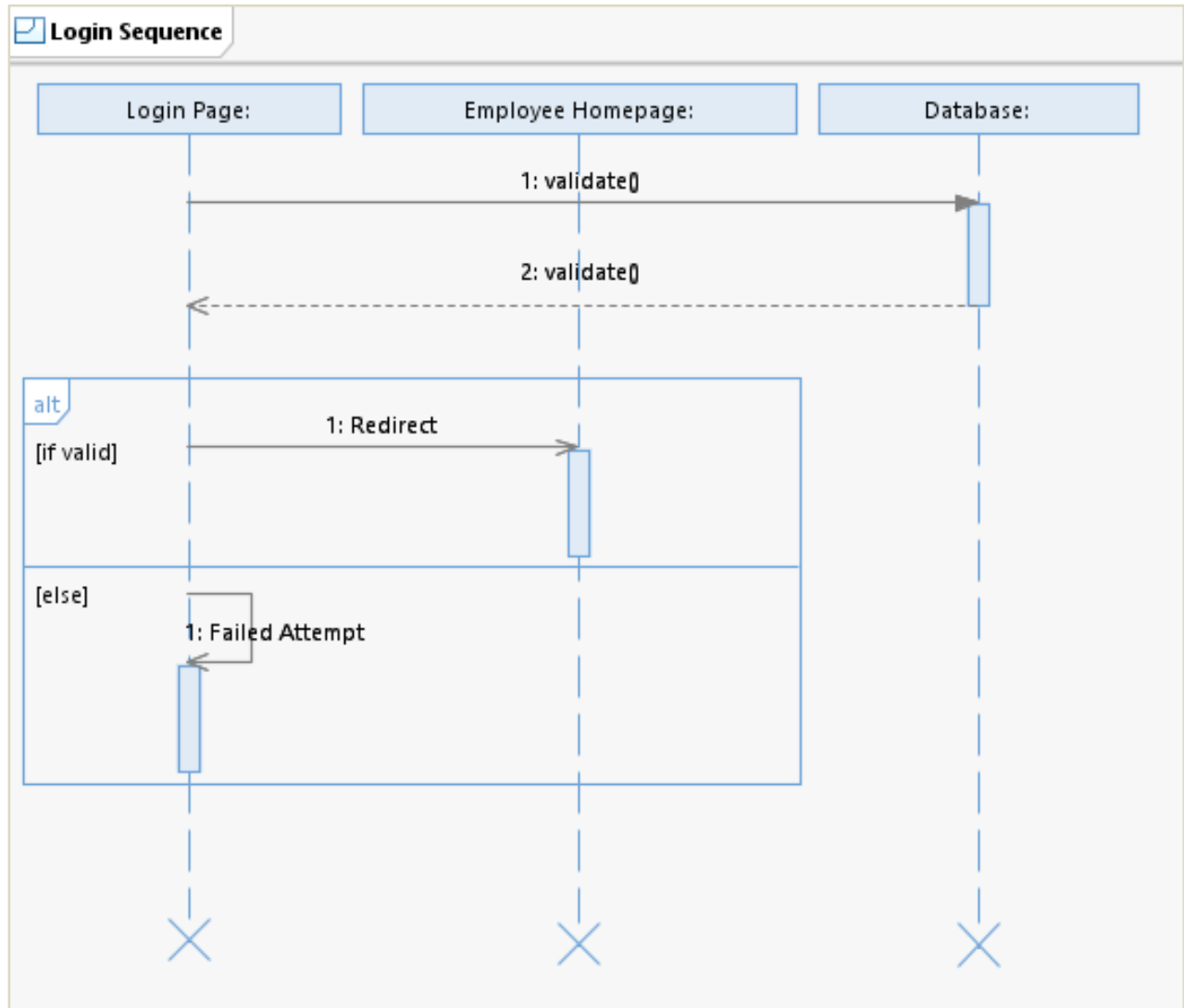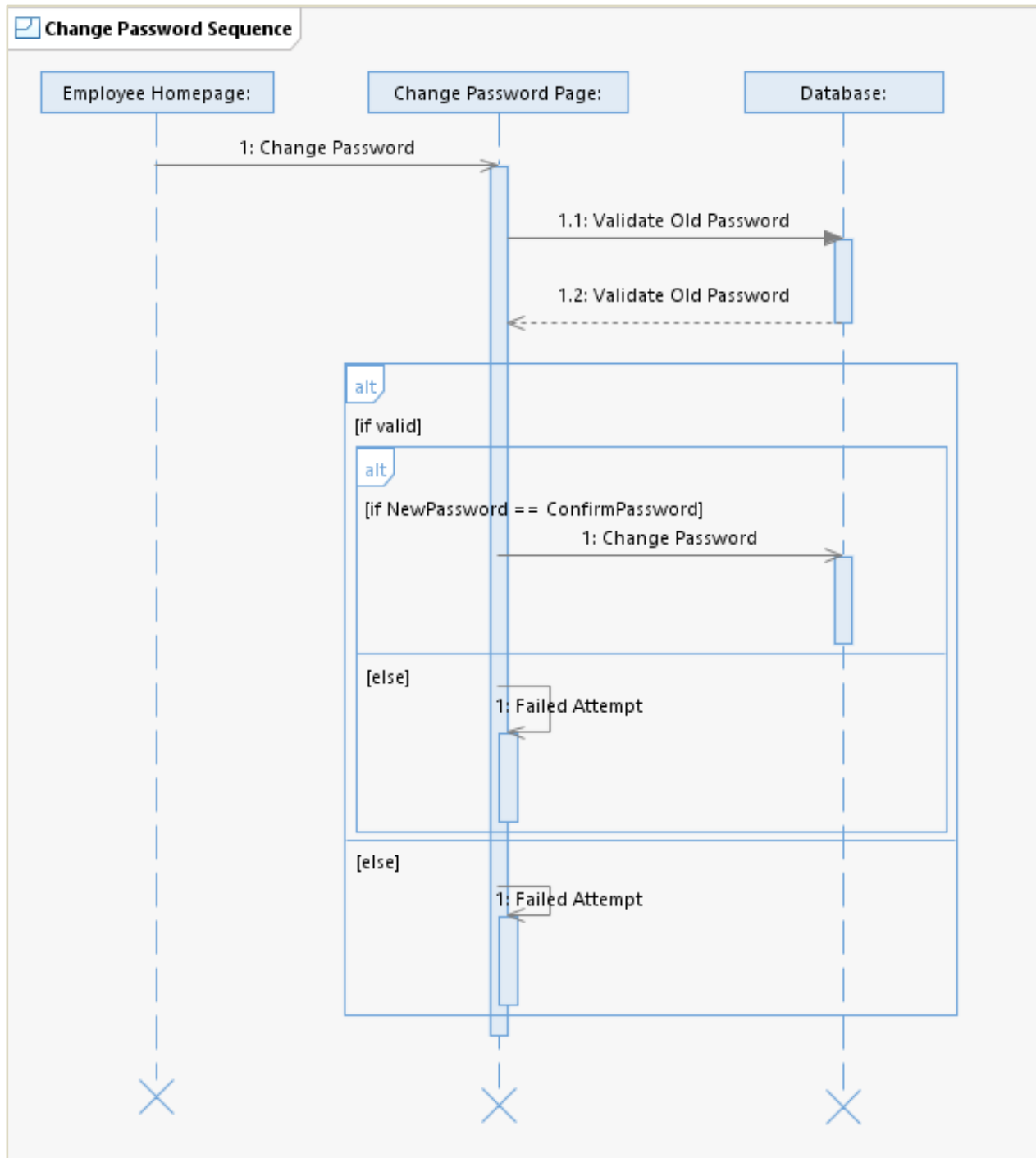- checkShortlistedApplications ()
- scheduleInterview ()
- sendMail ()
- updateResults ()

«Java Class»
Application
- applicantId : int
- opportunity : String
- resumeloc : String
- applicationDate : Date
- email : String
- shortlisted : String
- selected : String
- offer : String
- getApplication ()
- insertApplication ()
- getApplicantId ()
- setApplicantId ()
- getOpportunity ()
- setOpportunity ()
- getResumeloc ()
- setResumeloc ()
- getApplicationDate ()
- setApplicationDate ()
- getEmail ()
- setEmail ()
- getShortlisted ()
- setShortlisted ()
- getSelected ()
- setSelected ()
- getOffer ()
- setOffer ()

«Java Class»
Employee
- employeeId : int
- password : String
- name : String
- position : String
- supervisor : int
- hireDate : Date
- salary : double
- department : int
- qualification : int
- getEmployeeId ()
- setEmployeeId ()
- getPassword ()
- setPassword ()
- getName ()
- setName ()
- getPosition ()
- setPosition ()
- getSupervisor ()
- setSupervisor ()
- getHireDate ()
- setHireDate ()
- getSalary ()
- setSalary ()
- getDepartment ()
- setDepartment ()
- getQualification ()
- setQualification ()
- createEmployee ()

«Java Class»
DBConnection
- dbHost : String
- dbUser : String
- dbPass : String
- getConnection ()

«Java Class»
MailProcessor
- checkMail ()
- sendMail ()
- fetchMails ()
- createEmployee ()

«Java Class»
Position
- positionId : int
- positionDesc : String
- getPositionId ()
- setPositionId ()
- getPositionDesc ()
- setPositionDesc ()
- getPositions ()
- createPosition ()

«Java Class»
Qualification
- qualId : int
- qualDesc : String
- getQualId ()
- setQualId ()
- getQualDesc ()
- setQualDesc ()
- getQualifications ()
- createQualifications ()

«Java Class»
Localization
- LOCALE_DIR : String
- domFactory : DocumentBuilderFactory
- builder : DocumentBuilder
- doc : Document
- xpath : XPath
- path : String
- Localization ()
- setLocale ()
- getAvailableLocales ()
- getLocaleName ()
- getString ()

«Java Class»
Department
- deptId : int
- deptName : String
- deptHead : String
- getDeptId ()
- setDeptId ()
- getDeptName ()
- setDeptName ()
- getDeptHead ()
- setDeptHead ()
- getDepartments ()
- createDepartment ()

«Java Class»
JobOpportunity
- opportunityId : int
- position : String
- department : String
- numOfVacancies : int
- getOpportunityId ()
- setOpportunityId ()
- getPosition ()
- setPosition ()
- getDepartment ()
- setDepartment ()
- getNumOfVacancies ()
- setNumOfVacancies ()
- getJobOpportunities ()
- createJobOpportunity ()

2.10 Sequence Diagram

2.10.1 User Login
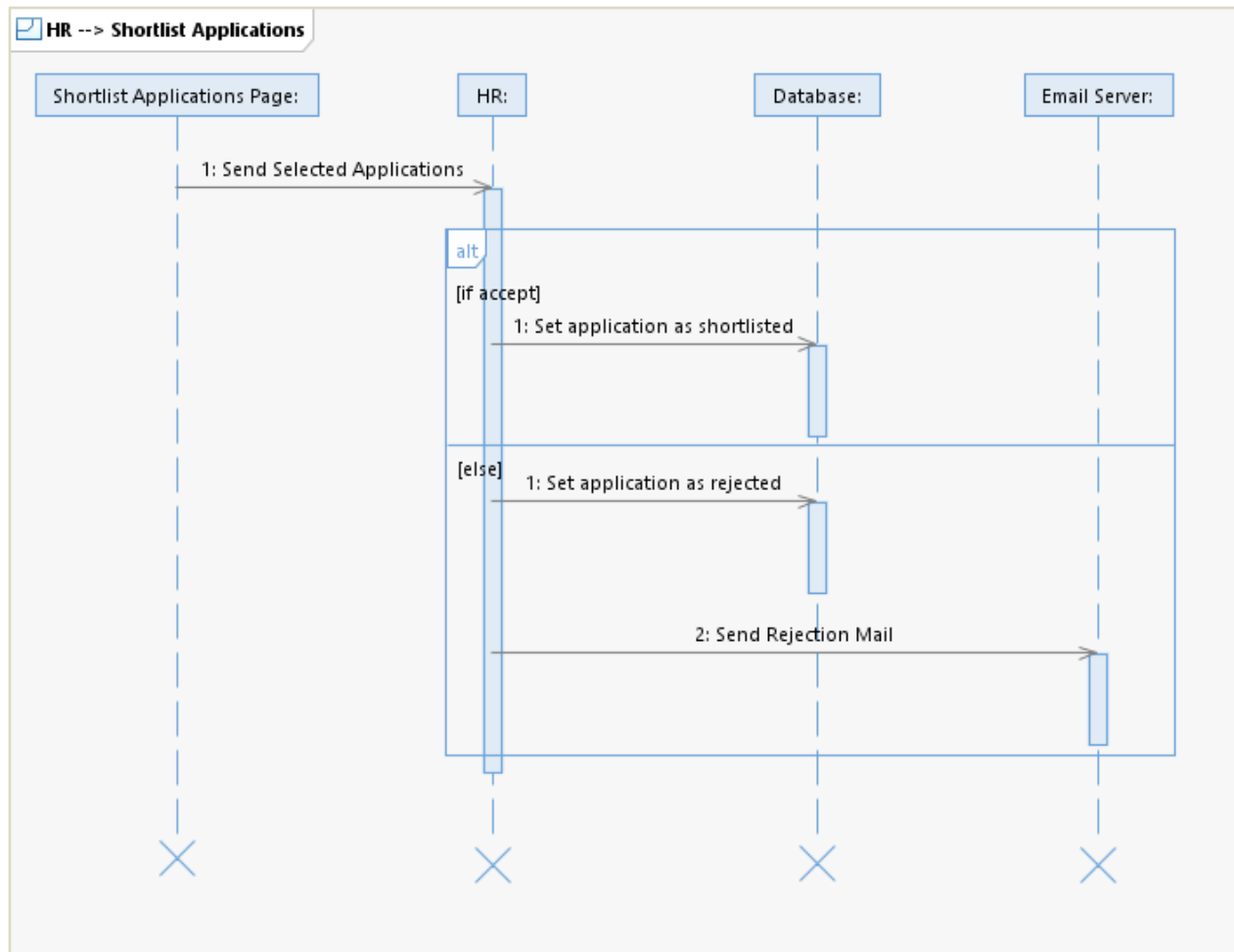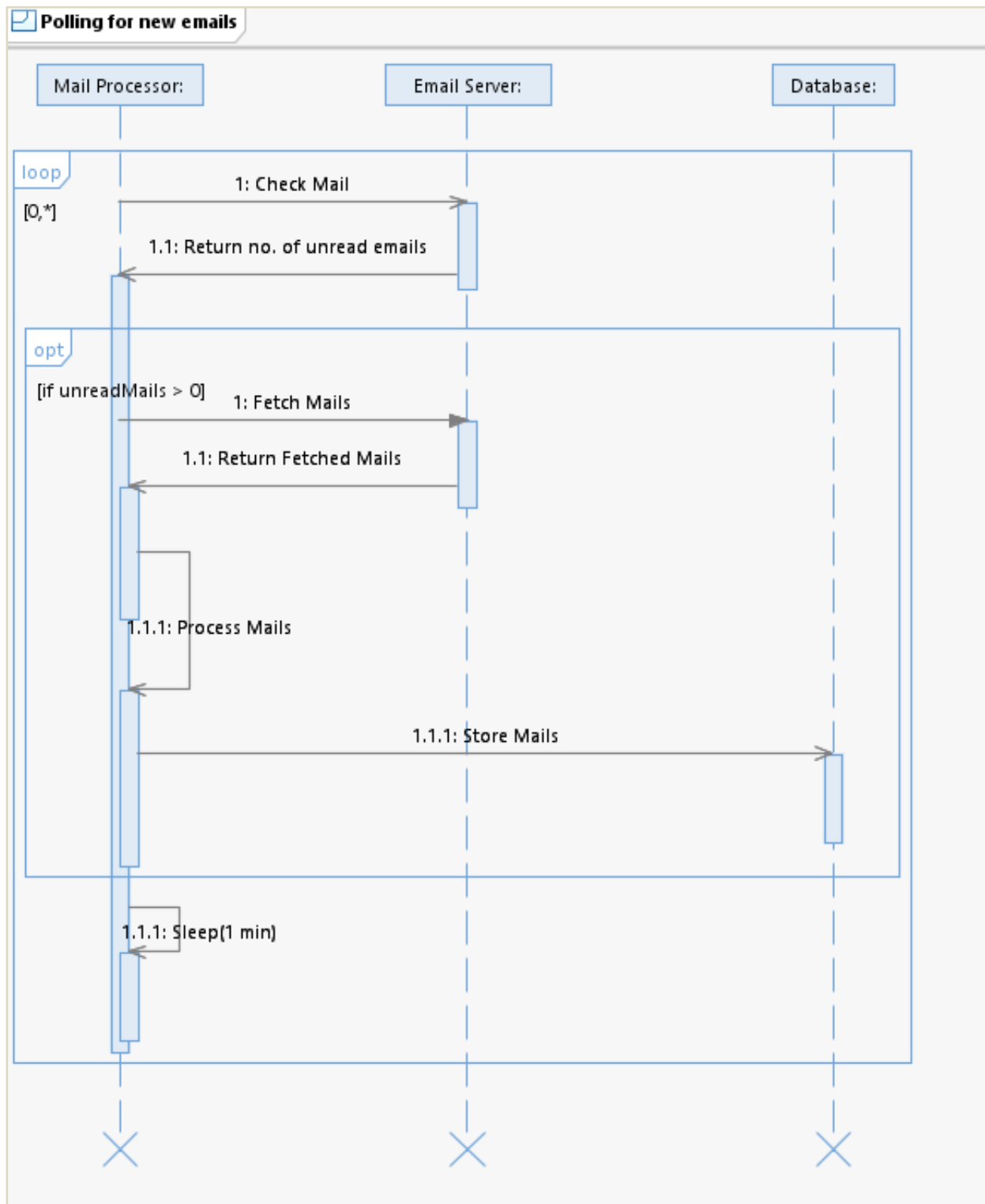
2.10.2 Change Password

2.10.3 Shortlist Applications

2.10.4 Polling for new emails

3. Specific Requirements

3.1 Use-Case Reports

3.1.1 HR-Head Use-Case Report

Use Case	Description
Login	The HR-Head can login using his login id and password
Change password	The HR-Head can change his password
View Application	HR-Head has the facility to view all the applications
Shortlist Application	HR-Head will have to shortlist few applications and thus completing first level of selection process
Send Rejection Mail	If an Applicant is not selected then HR-Head will have to send mail stating the reason of rejection
Send job Letter	If an applicant gets selected, he/she is sent a job letter with a link which has to be clicked to accept the job.
Send Interview dates	If an applicant is shortlisted then he/she is given a date and time of the interview which is sent to him by the HR-Head

3.1.2 Manager Use-Case Report

Use Case	Description
Login	The Manager can login using his username and password
Change password	The manager can change his password
Schedule Interview	It's Manager's job to schedule an interview and the schedule depends upon the number of shortlisted applications
Assign Interviewers	Manager has to assign an interviewer for a particular schedule
View Interview Results	Managers can view interview results.
Select Applications	The final and most important task of the Manager is to select the right applicants depending upon the marks of the marks of the interview process and number of vacancies in the department

3.1.3 Administrator Use-Case Report

Use Case	Description
Login	The Administrators can login using his username and password
Change password	The Administrators can change his password
Database Backup	Administrator can take Database Backup
View Reports	Administrators can View Reports regarding the system
Add New Departments	Administrators is the only person responsible for creating new department
Manage User Accounts	Administrator can add new employees in the system and even manage and modify the existing employees, like department change or change in salary, position.

3.1.4 Interviewer Use-Case Report

Use Case	Description
Login	An Interviewer can login using his username and password
Change password	An Interviewer can change his password
Check Interview Schedule	An Interviewer will have to check interview schedule and if there is a problem will have to notify the same to manager
Mark applicants after interview	After he takes an interview, he has to mark the applicants and on the basis of those marks, the manager will select the potential employees

3.1.5 Applicant Use-Case Report

Use Case	Description
Send Job application	An applicant will have to send his resume along with an opportunity id in a particular format.
Accept/Reject Job Offer	After the shortlisting process the applicants will be sent a job letter along with a link which has to be clicked to confirm job acceptance.

3.2 Supplementary Requirements

- **24 x 7 Availability:** The world never sleeps. So 24 hours availability becomes critical. Because system can be an automated Process, so it can stay open for 24 hours a day. So UPS support must be on server site for at least 8 hours in case of power failure. System will remain inaccessible to users between 2:00 am to 4:00 am for backup and maintenance purpose which will be done by the administrator.

- **Dynamic Website:** Most of the Web implementations consisted of static HTML pages. This becomes very difficult to manage if the number of pages gets too large. An effective system should be largely dynamic, taking advantage of technology that automates this process rather than relying on manual processes. Application should serve dynamic user based customized web pages to its clients from server.

- **Provide good performance and the ability to scale the server:** The Web application Server should provide good performance and the ability to manage performance with techniques, such as support for caching, clustering, and load balancing.

- **Providing session management capability:** Web application developers should not spend valuable time worrying about how to maintain sessions within the application. The Web Application Server should provide these services.

- **User friendly:** The site should be very user friendly and must provide help and support at each step. The help and support system is both online and off-line through the phone-line.