Design

Document

AGENCY MANAGEMENT SYSTEM

**Index**

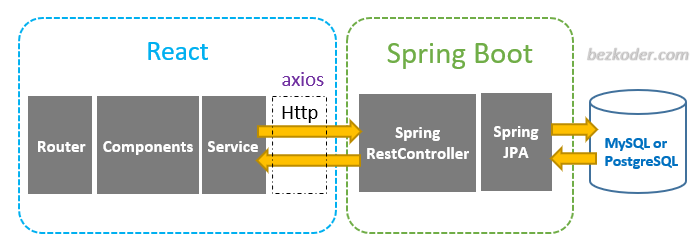
|  |  |  |
| --- | --- | --- |
| Sr No | Content | Page No |
| 1 | Introduction | 3 |
| 2 | Architectural Design | 3 |
| 3 | High Level Design | 6 |
|  | 3.1 E-R Diagram | 6 |
|  | 3.2 Page Navigation Diagram | 7 |
|  | 3.3 Data Flow Diagram | 9 |
|  | 3.4 Deployment Diagram | 12 |
| 4 | Low Level Design | 13 |
|  | 4.1 Database Design | 13 |
|  | 4.2 Stored Procedure | 16 |
|  | 4.3 Details Of Page Navigation | 17 |

* **Introduction:**

This document is meant for the description of the structure and the database which we are using in this project. This document gives brief description about Architecture of the system, E-R diagram of the system and the table descriptions, the page navigation diagrams and the detail description for the page navigation.

* **Architecture Design:**

Following diagram shows the details of the Agency Management System system architecture.



This System consist of three tiers as listed below,

* First tier
* Second tier
* Third tier

**First Tier:**

This tier is used for user interface and it is called as client tier. In this tier we are using ReactJs because of it provides better interactivity, easier navigation, compact. The use of java script facilities us for the client side validation. That’s why in first tier we are using the java script. We are using ReactJs for the presentation purpose.

**Second Tier:**

Second Tier is comprises of two parts listed below,

* **Server UI**

In this part of second tier we are using JSP, because it provides better UI to system, as well as it provides the dynamically designing of pages.

* **Server Process**

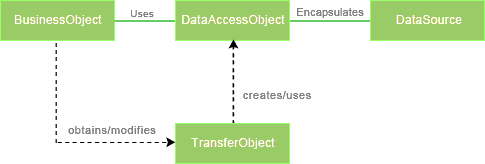
Servlet API is standard and freely available on the internet (like JSPs) servlets have the advantages like ease of development & platform independence (like Java) they can access all the J2SE and J2EE APIs can take the full advantage & capabilities of the Java programming language.

**Third Tier:**

Third tier consist of a Data Access Object (DAO) and the back end i.e. the database of Online PG Management system.

**Data Access Object (DAO):**

Data access object layer has proven good in separate business logic layer and persistent layer. The DAO design pattern completely hides the data access implementation from its clients.

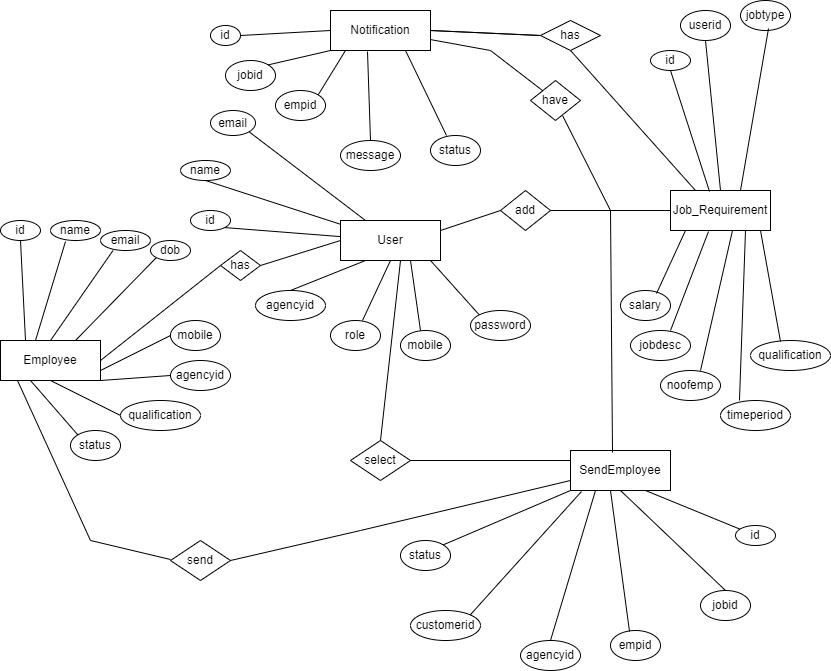


*Fig Data Access Object (DAO) Mechanism*

**Java Database Connectivity (JDBC):**

JDBC is used to provide database connectivity from java to database. Using Java database connectivity we can update/retrieve data to/from database with java programs. The main advantage of using JDBC is we can execute database queries by the program so that we can utilize the functionality provided by the database (with the queries). More over we can use triggers too. JDBC provides much other functionality (like the functions provided by CallableStatemtent class) to manage the data. Additionally, loading the driver will be different to different databases.

* **High Level Design:**
* **E-R Diagram:**



Above E-R Diagram shows that database of Agency Management system consist of following entities:

* **User**

This entity contains the id,name,emailed,agencyId,role,mobile,password .

* **Employee**

This entity contains the id, name, emailId, dob, mobile, agencyId, Qualification, status.

* **SendEmployee**

This entity contains the status, CustomerId , agencyId, empid, jobId,id.

* **Notification**

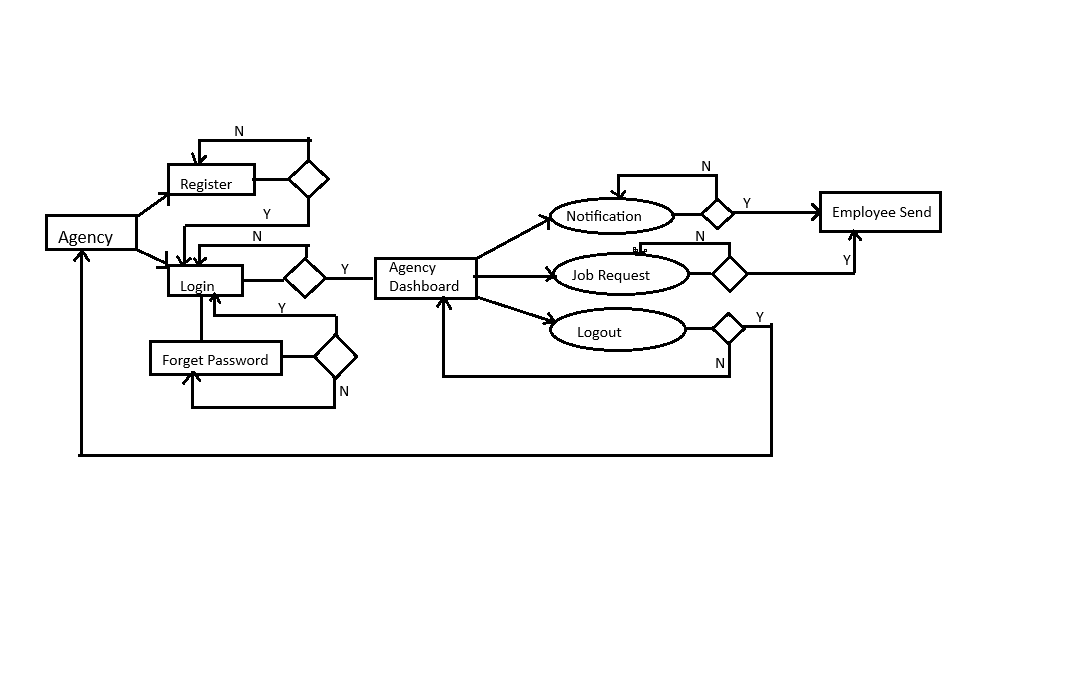
This entity contains the id,jobId,empid,message,status.

* **Job\_Requirement**

This entity contains the Property id,userId,jobtype,salary,jobDesc.

* **Page Navigation Diagram:**
* **Agency**

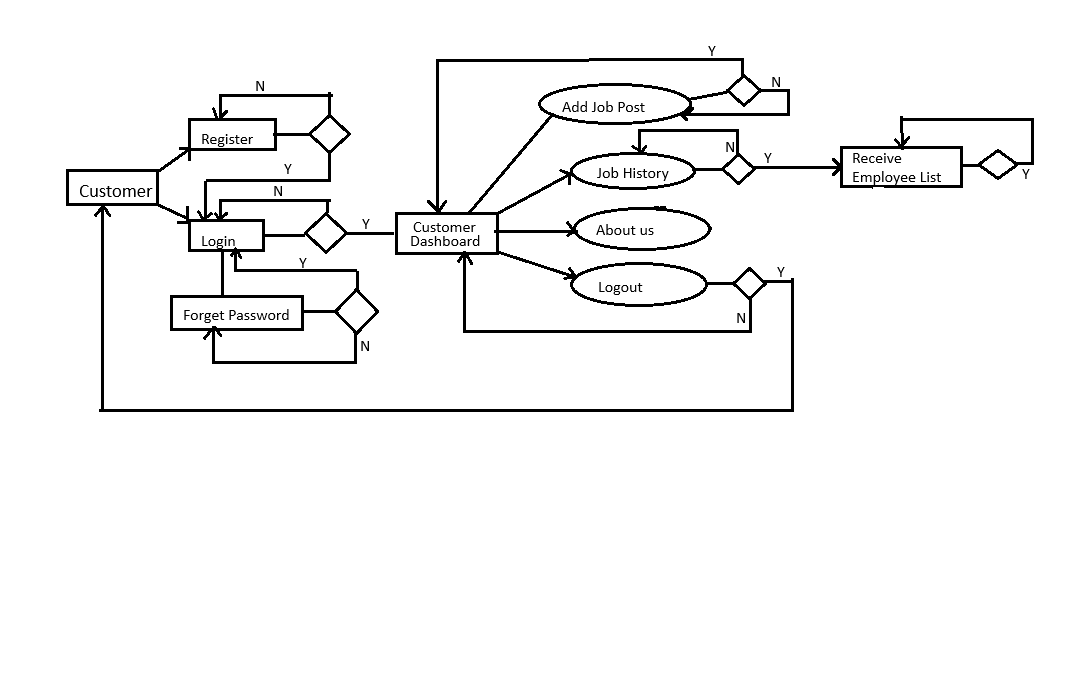
Following diagram explains the page navigation for the Agency module:



**Fig. Page Navigation for Agency**

* **Customer**

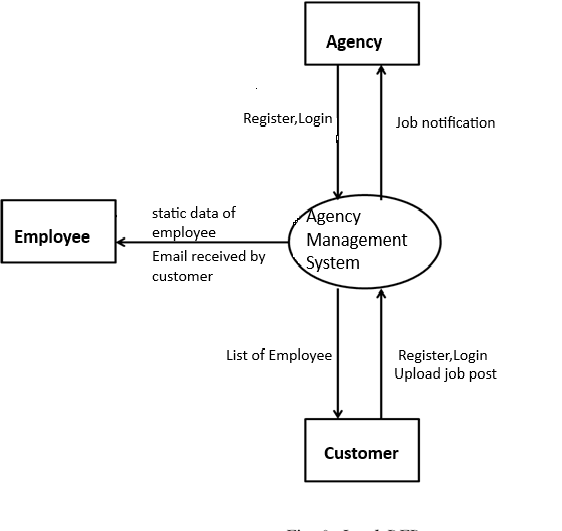
Following diagram explains the page navigation for the Customer module:



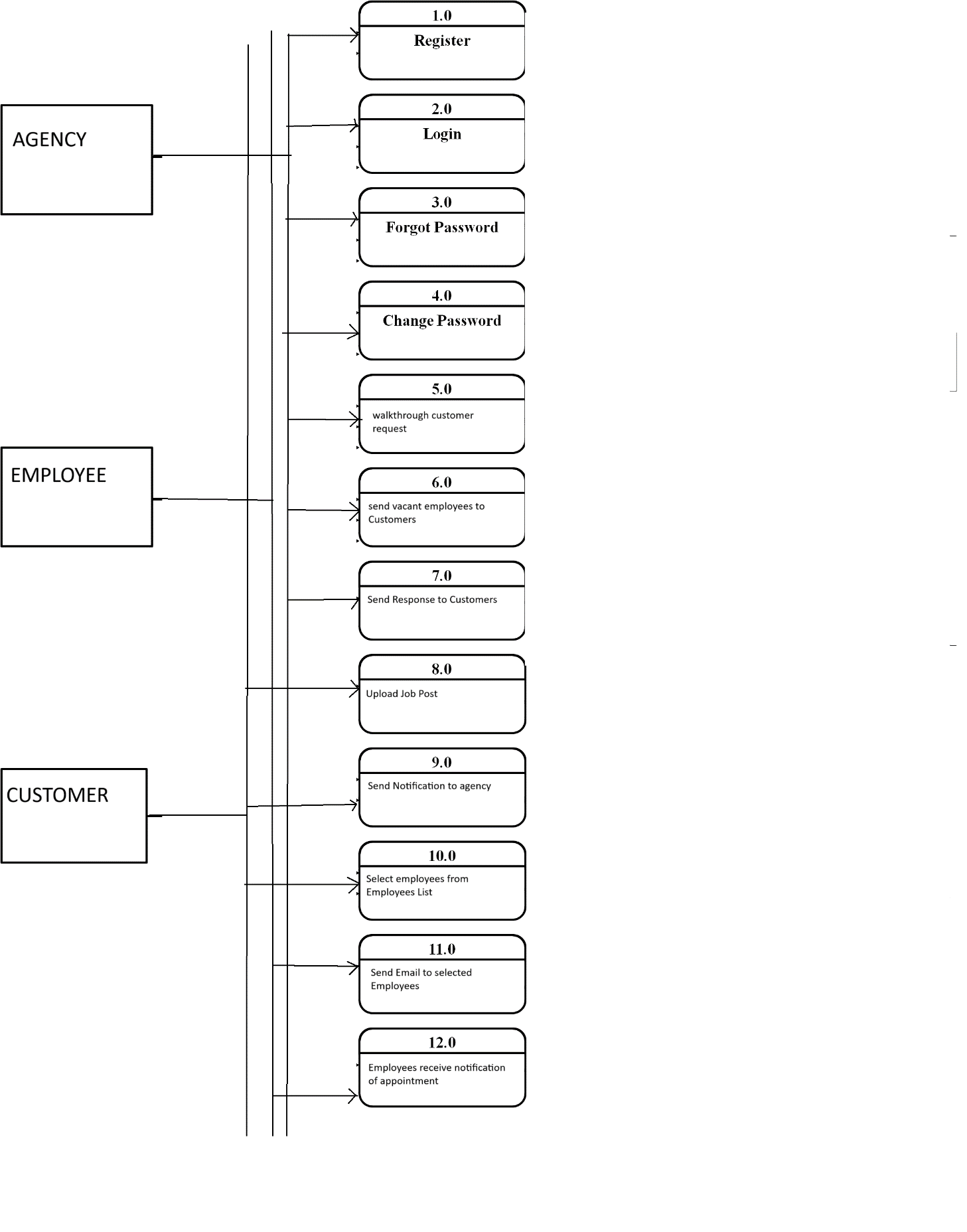
**Fig. Page Navigation for Customer**

**Data Flow Diagram:**

* **0-Level DFD:**

****

* In 0-Level DFD, there are three Entities:
* Agency
* Customer
* Employee
* **1-Level DFD:**

*Fig. 1-Level DFD*

In 1-Level DFD, Agency Entity having following processes:

* Register(Process 1.0)
* Login(Process 2.0)
* Forgot Password(Process 3.0)
* Change Password(Process 4.0)
* Walkthrough Customer Request(Process 5.0)
* Send valid Vacant Employees(Process 6.0)
* Send Response to Customer Request(Process 7.0)

Customer Entity having following processes:

* Register(Process 1.0)
* Login(Process 2.0)
* Forgot Password(Process 3.0)
* Change Password(Process 4.0)
* Upload Job Post(Process 5.0)
* Send notification Message to Agency(Process 6.0)
* Select Employees from Employee List(Process 11.0)
* Send email to all selected Employees(Process 12.0)
* Employee Entity:
* Employee Entity will have static data.
* Selected Employees will received notification through email.
* **Low Level Design:**
* **Database Design:**

**1]Employee Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| Id | int | Not Null | Primary key | Auto\_increment | Employee ID |
| name | Varchar(20) | yes |  | NULL | Employee name |
| Email | Varchar(20) | yes | unique | NULL | Employee email |
| dob | date | yes |  | NULL | Employee DateOfBirth |
| qualification | Varchar(30) | yes |  | NULL | Employee qualification |
| agencyid | int | yes |  | NULL | Employee agency id |
| status | Varchar(10) | yes |  | vacant | Employee status |
| mobile | Varchar(11) | No |  | NULL | Employee mobile number |

**2] SendEmployee Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| id | Integer | Not Null | Primary key |  | Employee ID |
| Customerid | Integer | Yes |  | NULL | Customer ID |
| empid | Integer | Yes |  | NULL | Employee ID |
| jobid | Integer | Yes |  | NULL | Job ID |
| agencyid | Integer | Yes |  | NULL | Agency ID |
| status | Varchar(10) | Yes |  | Vacant | Status of Employee |

**3] Notification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| id | Integer | Not Null | Primary key | Auto\_increment | Notification id |
| Message | Varchar(150) | Yes |  | Null | Notification Message |
| jobid | Integer | Yes |  | Null | Job id |
| empid | Integer | Yes |  | 0 | Employee id |

**4] User Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| User\_id | Integer | Not Null | Primary key | Auto\_Increment | User id |
| name | Varchar(50) | yes |  | Null | User name |
| email | Varchar(45) | yes |  | Null | User email |
| mobile | Varchar(11) | yes | Unique | Null | User mobile |
| password | Varchar(20) | yes |  | Null | User password |
| role | Varchar(11) | Yes |  | Null | User role |
| agencyid | integer | Yes |  | 0 | Agency id |

**5] Job\_requirement Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| id | Integer | NOT NULL | Primary key | AUTO\_INCREMENT | Job id |
| Job\_type | Varchar(30) | yes |  | NULL | Type of the job |
| No\_of\_emp | Integer | yes |  | NULL | Number of employee |
| Job\_desc | Varchar(45) | yes |  | NULL | Job description |
| qualification | Varchar(10) | yes |  | NULL | Qualification of employee |
| Time\_period | Varchar(30) | Yes |  | No Bond | Time Period |
| salary | Integer | Yes |  | Null | Salary of employee |
| userid | Integer | Yes |  | Null | Id of the user |