**Case Study: Library System**

1. **Problem Statement**

The Library System is a web-based application used to automate a library. It allows the librarian to maintain the information about books, magazines and CDs. It also allows the librarian to maintain the information about its users. It provides the facilities such as search for items, browse, checkout items, return items, make reservation, remove reservation etc. to its users.

To borrow the items from the library, the users must register in the system. The search option allows the users to search for any item in the library. If the user finds that the required item is available in the library, he/she can checkout the item from the library. If the item is currently not available in the library, the user can make reservation for the item. When the item becomes available the respective user who made the reservation for that item first is notified.

The reservation is canceled when the user checks out the item from the library or through an explicit cancellation procedure.

The system allows the librarian to easily create, update, and delete information about titles, borrowers, items and reservations in the system. The librarian is an employee of the library who interacts with the borrowers whose work is supported by the system.

The Library System can run on popular web-browser platforms like Windows Explorer, Netscape Navigator etc. It can be easily extended with new functionality.

1. **Vision Document**

A vision document describes the higher level requirements of the system specifying the scope of the system.

The vision document for the Library System might be

* It is a support system
* The library lends books, magazines and CDs to borrowers who are registered in the system
* The Library System handles the purchases of new titles for the library
* Popular titles are brought in multiple copies. Old books, magazines and CDs are removed when they are out of date or in poor condition
* The librarian is an employee of the library who interacts with the borrowers whose work is supported by the system
* A borrower can reserve a book, magazine or CD that is not currently available in the library so that when it is returned or purchased by the library, the borrower is notified
* The reservation is canceled when the borrower checks out the book, magazine or CD or through an explicit cancellation procedure
* The librarian can easily create, update, and delete information about titles, borrowers, items and reservations in the system
* The system can run on popular web-browser platforms like Windows Explorer, Netscape navigator etc.
* The system is easy to extend with new functionality

# Glossary

Key terms are denoted in italics within the use-case specifications.

**Item** - A tangible copy of a *Title*.**Title** - The descriptive identifying information for a book or magazine. Includes attributes like name and description.**Reservation** - Whenever a borrower wishes to checkout an *Item* that is not available due to previous checkout by a different borrower a request can be made (a reservation) that locks the borrower in as the next person able to checkout the *Item*.

# Actors

**Borrower** - Interactive actor who uses the library to search for *Titles*, make reservations, checkout, and return *Items*.

**Librarian** - Interactive actor responsible for maintenance of the inventory, acting on behalf of the borrowers, and general support of the library (non-automated as well).**Master Librarian** - Interactive actor, themselves a Librarian, who is also responsible for maintaining the set of librarians for the system.**Registered User** - Any interactive user for whom the system maintains a system account. This includes borrowers, librarians, and master librarians. Capabilities include basic login, browsing and searching for *Titles*.

1. **Supplementary Specification Document**

**Objective**

The purpose of this document is to define the requirements of the Library system. This document lists the requirements that are not readily captured in the use-cases of the use-case model. The supplementary specification and use-case model together capture a complete set of requirements of the system.

**Scope**

This supplementary specification defines the non-functional requirements of the system such as reliability, performance, supportability, and security as well as functional requirements that are common across a number of use-cases.

**Reference**

None

**Common Functionalities**

* Multiple users must be able to perform their work concurrently
* If the reserved item has been purchased or available, the borrower must be notified

**Usability**

The desktop user interface shall be Widows NT or Windows 2000 compliant

**Reliability**

The system shall be 24 hours a day, 7 days a week and not more than 10% down time

**Performance**

* The system shall support up to 2000 simultaneous users against the central database of any given data
* The system must be able to complete 80% of all transactions within 5 minutes

**Supportability:** None

**Security**

* The system must prevent borrowers from changing borrowers information, items information, titles information, and librarians information
* Only Librarian can modify borrowers information, items information, and titles information
* Only Master Librarian can modify librarians information

1. **Use – Case Model**

**Actors**

Actor is something external to the system and interacts with the system. Actor may be a human being, device or some other software system.

For Library system, actors might be;

* Librarian
* Borrower

**Use – Case**

A use-case represents sequence of actions performed by the system that yields an observable result of value for a particular actor. Use-case represents a functional requirement of a system. For Library system, we can find the following use-cases;

* Login
* Search
* Browse
* Check out item
* Return item
* Make reservation
* Cancel reservation
* Manage titles
* Manage items
* Manage borrowers
* Manage librarians

**Use - Case Diagram**



**5.4 Use – Case Descriptions**

## 5.4.1 Use-Case Specification: Login

### **5.4.1.1 Description**

A registered user can log in and, upon verification, can initiate subsequent actions.

### **5.4.1.2 Flow of Events**

#### **5.4.1.2.1 Basic Flow**

1. Initiated when a registered user chooses to Login.
2. The system prompts for username and password.
3. The registered user enters a username and password and submits them.
4. The system authenticates the username and password combination.
5. The system authorizes the registered user according to the role(s) to which the registered user has been assigned.
6. The system displays the main page and awaits subsequent action.

#### **Alternative Flows**

* + **Invalid Username/Password**

1. The system displays the Authentication Failed message.
   * **Account Locked**
2. The system displays the <appropriate message>.
   * **Authentication Service Unavailable**
3. The system displays a Service Unavailable message and does not permit any further attempts to login.

### **Special Requirements**

1. Up to three consecutive failed tries to login with invalid username/password combination until locking an account.
2. Minimum password length is 8 characters, and must include a combination of characters including at least one non-alphabetic character.

### **Preconditions**

User has an account with the system

### **Post-conditions**

#### **Primary Success Post-condition**

The user is considered authenticated and is viewing the main page from which additional actions can be initiated.

#### **Login Failure**

If the Login fails as described in any of the alternatives above, an appropriate message is displayed and the user is not considered authenticated.

### **Notes**

1. So far we are not doing much with roles.
2. The “appropriate message” above is vague; we need to come up with how we report this to the user.
3. We need to talk to security people about how reasonable it is to lock the user account after some number of failed attempts. If we keep that rule, we’ll need an Unlock Account use case.

## Use-Case Specification: Browse

### **Description**

A registered user can browse the contents of the library as a precursor to other actions.

### **Flow of Events**

#### **Basic Flow**

1. Initiated when a registered user chooses to browse *Titles*.
2. The system responds by displaying all of the *Titles* in the system, along with topical descriptions.
3. The registered user selects a *Title* for further information.
4. The system displays *Title* detail along with the *Items* and the available action on each *Item*.

#### **Alternative Flows**

* **No records**

1. The system displays message indicating no *Titles* are in the system.

### **Special Requirements**

* 1. The *Titles* will be sorted alphabetically by the name.

### **Preconditions**

The user has been authenticated.

### **Post-conditions**

#### **Primary Success Post-condition**

The registered user is viewing a *Title* along with the related *Items*.

## Use-Case Specification: Search

### **Description**

A registered user can search the contents of the library as a precursor to other actions.

### **Flow of Events**

#### **Basic Flow**

1. Initiated when a registered user chooses to perform a search of *Titles*.
2. The system responds by providing the registered user a means to enter search criteria.
3. The registered user enters search criteria and initiates the query.
4. The system determines results and displays the matching *Titles*, along with topical descriptions.
5. The registered user selects a *Title* for further information.
6. The system displays *Title* detail along with the *Items* and the available action on each *Item*.

#### **Alternative Flows**

#### **No matches**

1. The system displays message indicating no *Titles* in the system match this criteria.

### **Special Requirements**

1. The search only searches based on the name of the *Item*, not description or any other field.
2. The system shall use the percent sign as a wildcard (in keeping with standard SQL idioms).
3. The results will be sorted alphabetically by the name.

### **Preconditions**

The user has been authenticated.

### **Post-conditions**

#### **Primary Success Post-condition**

The registered user is viewing a *Title* along with the related *Items*.

### **Notes**

1. We might want to combine this with the Search use case. The combined use case could be called Select *Title* and one of the original use cases could be the basic flow and the other would be the alternative.
   * 1. **Use-Case Specification: Make Reservation**
        1. ***Description***

This use-case starts when the user wants to make a reservation for an item

* + - 1. ***Flow of Events***
         1. **Basic flow**

1. The system prompts the borrower to enter the item information for which he wants reservation
2. The borrower submits the item information
3. The system marks the item as reserved and associates the borrower with the reservation
   * + - 1. **Alternative Flow**

None

* + - 1. ***Special requirements***

None

* + - 1. ***Pre-conditions***

The borrower is viewing a particular title with an item that is not currently available

* + - 1. ***Post-conditions***

The item is marked as reserved and the reservation is saved in the database

* + - 1. ***Notes***

1. So far there is no nice way to figure out what a borrower has reserved.

**5.4.5 Use-Case Specification: Remove Reservation**

***5.4.5.1 Description***

The borrower can remove an existing reservation for an item.

* + - 1. ***Flow of events***
         1. **Basic Flow**

1. The system prompts the borrower for the item information for which the reservation is removed
2. The borrower enters the item information and submits
3. System marks the item as no longer reserved
   * + - 1. **Alternative Flows**

None

* + - 1. ***Special requirements***

None

* + - 1. ***Pre-conditions***

The borrower is viewing a particular *Title* with an *Item* that is reserved by the borrower.

* + - 1. ***Post-conditions***

The previously reserved *Item* is no longer reserved.

* + 1. **Use-Case Specification: Check out Item**
       1. ***Description***

This use-case starts when the borrower wishes to check out an item from the library

* + - 1. ***Flow of Events***
         1. **Basic Flow**

1. The borrower performs a search for the desired titles
2. The system prompts the borrower to enter search criteria
3. The borrower specifies the search criteria and submits
4. The system locates matching titles and displays them to the borrower
5. The borrower selects titles to check out
6. The system displays the details of titles as well as whether or not there is an available item to be checked out
7. The borrower confirms the check out
8. the system checks out the item
9. Steps 1-8 can be repeated as often as needed by the borrower
10. The borrower completes the check out
11. The system notifies the Librarian that the borrower has concluded the check out item session and displays instructions for the borrower to collect the items
    * + - 1. **Alternative Flows**

None

* + - 1. ***Special requirements***
      2. ***Pre-conditions***

The borrower is viewing a particular *Title* with an *Item* that is currently available.

* + - 1. ***Post-conditions***

The *Item* is demarked as checked out to the borrower.

* + 1. **Use-Case Specification: Return Item**
       1. ***Description***

This use-case starts when the borrower wishes to return an item

* + - 1. ***Flow of Events***
         1. **Basic Flow**

1. The system prompts the borrower to enter the item information he wants to return
2. The borrower enters the item information and submits
3. The system marks the item as available
   * + - 1. **Alternative Flows**

None

* + - 1. ***Special requirements***

None

* + - 1. ***Pre-conditions***

The borrower is viewing a particular *Title* with an *Item* that is checked out by the borrower.

* + - 1. ***Post-conditions***

The *Item* is demarked as available.

* + - 1. ***Notes***

A reasonable future enhancement would be to notify anyone with a reservation on the *Item*.

**5.5 Activity Diagram**

****

**6. Design Model**

**6.1 Class Diagram**



* 1. **Sequence Diagram and Collaboration diagrams**
     1. **Sequence Diagram for Login Use-case**



**6.2.2 Collaboration Diagram for Login Use-case**



**6.2.3 Sequence Diagram for Search Use-case**



**6.2.4 Collaboration Diagram for Search Use-case**



**6.2.5 Sequence Diagram for Browse Use-case**



**6.2.6 Collaboration Diagram for Browse Use-case**



**6.2.7 Sequence Diagram for Make Reservation Use-case**



**6.2.8 Collaboration Diagram for Make Reservation use-case**



**6.2.9 Sequence Diagram for Remove Reservation Use-case**



**6.2.10 Collaboration Diagram for Remove Reservation Use-case**



**6.2.11 Sequence Diagram for Check Out Item Use-case**



**6.2.12 Collaboration Diagram for CheckOut Item Use-case**



**6.2.13 Sequence Diagram for Return Item Use-case**



**6.2.14 Collaboration Diagram for Return Item Use-case**



**6.3 State chart Diagram for Title Class**



1. **Deployment Model**

**7.1 Component Diagram**

****

**7.2 Deployment Diagram**

****

**Point of Sale System**

**ABSTRACT:**

**Definition:**

“Point of sales can mean a retail shop or the location where transaction occurs “. These are intended to draw the customers attention to products

OR

Point of sale or POS as it is more commonly refers to the capturing the date and customers payment information at a physical location when goods are sold.

The POS transaction is captured using a variety of devices which include computers, cash registers, optical and barcode scanners, magnetic card readers etc.

**Present system:**

1. In the present system the people go to the market manually and buy the products. It is time consuming
2. In the shop products are given the by the sales person to the customers. So number of sales person are maintained.
3. The stock details are also maintained manually
4. The billing and updations are also done manually
5. The regular customers details are also maintained. But it is difficult in the present system
6. The discounts and the special offers in the registers
7. The roles are
   1. owner
   2. customer
   3. sales person
   4. stock holders
   5. others

**Proposed system**

1. The proposed system is computerized system
2. In the phone to delivery are also provided
3. The stock details, the billing and the customer information stored in computer
4. If the product is purchased it is automatically deducts from the stock
5. The products are scanned through scanners then the cost will be displayed
6. the roles are
   1. Administrator
   2. Customer
7. we have to provide debit card facility
8. We have to analyse the sales based upon season and product
9. We provide the warrenty for the products and after the warranty period the details of the products will be automatically removed

**Functional requirements:**

inventory():

\* current stock()

\* customer orders()

\*required stock()

\*total sales()

Billing()

\*scaning()

\*display()

Payments()

\*cash

\*card

**Nonfunctional requirements:**

\*scanner

\*printer

\*building

\*furniture

Software requirements : \*frontend:java,html

\*backend:sql

Operating system : \*windows xp

Hardware requirements : \*hard disk 160GB

\*ram 2GB

\*processor speed 3.2GHZ

Login

Administrator

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Datetype | Size | Constraints |
| Adminid  Password | Varchar2  Varchar2 | 20  10 | Primarykey  Notnull |

Customer

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Datetype | Size | Constraints |
| Cname  Custid  Address  Billid | Varchar2  Varchar2  Varchar2  Varchar2 | 20  20  20  10 | Notnull  Primarykey  Notnull  Foreignkey |

Billing

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Datetype | Size | Constraints |
| Billid  Productname  Date  Warrenty  Tax  Discount  Itemcost  Total  No.ofItems | Varchar2  Varchar2  date  number  number  number  number  number  number | 10  20.  20  10  10  10  10  10  10 | Primarykey  Notnull  Notnull  Notnull  Notnull  Notnull  Notnull.  Notnull  Notnull |

Inventory

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Datetype | Size | Constraints |
| Stockitemid  Stockoutid  Reqid  Totid | Varchar2  Varchar2  Varchar2  Varchar2 | 20  20  20  20 | Foreignkey  Foreignkey  Foreignkey  Foreignkey |

Stockout

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Datetype | Size | Constraints |
| Reqid  Ordersid | Varchar2  Varchar2 | 20  20 | Primarykey  Foreignkey |

Totalsales

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Datetype | Size | Constraints |
| Totid  Itemid  No.ofitems  Cost  Total | Varchar2  Varchar2  Number  Number  Number | 20  20  10  10  10 | Primarykey  Notnull  Notnull  Notnull  Notnull |

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Datetype | Size | Constraints |
| Cardid  Name  Bankname  Amount  Orgid | Varchar2  Varchar2  Varchar2  Varchar2  Varchar2 | 20  20  20  20  20 | Primarykey  Notnull  Notnull  Notnull  Notnull |

Supervisor

Controller

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Datetype | Size | Constraints |
| Conid  Conname  Totid | Varchar2  Varchar2  Varchar2 | 20  20  20 | Primarykey  Notnull  Foreignkey |

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Datetype | Size | Constraints |
| Sname  Sid  Custid  Empid  Stockitemid | Varchar2  Varchar2  Varchar2  Varchar2  Varchar2 | 20  20  20  20  20 | Notnull  Primarykey  Foreignkey  Foreignkey  Foreignkey |

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Datetype | Size | Constraints |
| Rollid  Ename  Esal  Custid |  |  |  |

Class diagram



































**JOB PORTAL**

**Abstract:**

**Definition:**

It is the corner for the jobseekers , in which there is an exchange of information between the employers in the organization and jobseekers who are applying for the job. Here the employers are looking for the skilled persons who are suitable for that post. The organization specify their requirements , no. of vacancies and categories and the qualification for that area. In the same manner , the jobseekers are also specify their qualification and skill set. By the exchange of information between the employer and jobseeker, they attain the suitable person for the particular job which is used for the growth of the organization .

**Present System:**

In the present system , first the employers advertise their vacancies of jobs through the newspapers. The persons who are seeking for the job search through newspaper . If the jobseeker qualifications are suitable to the organizational requirements then they forward their resume through post or by hand.

After the completion of this process , the organization sends call letter to the jobseekers who are suitable for thejob.

**Drawbacks:**

1. It is a time taking process .
2. There may be a chance of postal delay.
3. It is a risky process between the jobseeker and organization.
4. There is no acknowledgement whether they received the resume or not.
5. Limited no.of jobseekers are applying for the job.

**Proposed System:**

In the proposed system , the portal is divided into two blocks. One block for theorganisation andanother block for the jobseekers.In theO rganisational

Block ,each organization contains its own profile and its requirements.In the jobseekers block, there is a register field which is useful for the jobseekers to register into the portal. There is a chance for the jobseekersto update their resume at any time.The updated resumes are viewed by the webmaster and forwarded to the organization.

A single person may have an opportunity to apply for the no.of jobs by using this portal.

**Requirements:**

1.functional requirements

2. non-functional requirements

3. software requirements

**Functional requirements:**

1. Job seeker functional requirements:

* Visiting the portal
* Registering to the job portal
* Login to his own id
* Knowing the information about the portal
* Uploading the details and resume
* Updating the resume at any time
* Receiving the information from the web master
* Logout from the portal

2. Web master functional requirements:

* Examining the job seekers block
* Communicating with the organizations
* Keeping the details about the job and vacancies in the organizational block
* Categorize the resumes based on qualification
* Forwarding resumes to the organization
* Forwarding the updated resumes
* Getting the information from the organization
* Sending the information to the job seekers
* Getting the feedback from the job seekers

3. Employer functional requirements:

* Communicating with the web master
* Specifying their requirements
* Receiving the resumes from the web master
* Selected candidates are intimated to the web master

**Non-functional requirements:**

The basic requirements to approach this site are a computer, phone, an internet connection and site address for the job seeker.

**Software requirements:**

Frontend: java

Backend: oracle 8i

Os: windows XP

**Hardware specifications:**

Ram: 2GB

Hard disk: 500GB

Mother board: intel

Processor: intel dual core

**Community portal(job site)**

**Data Dictionary**

**Table Name: Country (Master Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| CountryId | Number | 3 | Primary Key |
| Country Name | Varchar2 | 15 | Not null |

**Table Name: State (Master Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| StateId | Number | 3 | Primary Key |
| CountryId | Number | 3 | Foreign Key |
| State Name | Varchar2 | 10 | Not null |

**Table Name: City (Master Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| CityId | Number | 3 | primary key |
| City Name | Varchar2 | 10 | Not null |
| StateId | Number | 3 | Foreign Key |
| CountryId | Number | 3 |  |

**Table Name: Skills (Master Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| SkillId | Number | 3 | primary key |
| Skill Description | Varchar2 | 100 | Not null |

**Table Name: UserType (Master Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| UserTypeId | Number | 15 | primary key |
| User Description | Varchar2 | 20 | Not null |

**Table Name: Registration (Master Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| UserTypeId | Varchar2 | 15 | Foreign Key |
| Username | Varchar2 | 15 | Foreign Key |
| User Password | Varchar2 | 20 | Not null |
| UserFirstName | Varchar2 | 15 | Not null |
| UserDOB | Date | 10 |  |
| User address | Varchar2 | 10 |  |
| CountryId | Varchar2 | 10 | Not null |
| CityId | Varchar2 | 15 | Not null |
| UserPhoneNo | Number | 3 | Not null |
| User email | Varchar2 | 20 |  |
| UserLastName | Varchar2 | 20 | Not null |
| StateId | Varchar2 | 20 | Not null |

**Table Name: EducationQualification(Master Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| Course | Varchar2 | 10 | Not null |
| Aggregate | Number | 10 | Not null |
| Year of passing | Number | 10 | Not null |
| Backlogues | Number | 10 | Not null |

**Table Name:ExperienecDetails (Master Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| ExpId | Number | 8 | Primary Key |
| ExpCompany | Varchar2 | 15 | Not null |
| ExpDescription | Number | 8 | Not null |
| ExpNoOfYears | Varchar2 | 2 | Not null |
| ExpDetails | Varchar2 | 15 |  |
| Username | Varchar2 | 15 | Foreign Key |

**Table Name: SkillTransaction (Transaction Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| SkillId | Number | 15 | Not null |
| Username | Varchar2 | 20 | Not null |

**Table Name: EmploymentDetails(Master Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| Username | Number | 8 | Foreign Key |
| NameOfPost | Varchar2 | 8 | Not null |
| NumberOfPosts | Number | 8 | Not null |
| ExpDetails | Varchar2 | 2 | Not null |
| NoOfYearsExp | Number | 15 | Not null |
| ApplyLastDate | Date | 10 | Not null |
| Interview Date | Date | 10 | Not null |
| SkillId | Number | 10 | Foreign Key |
| Interview Time | Varchar2 | 3 | Not null |

**Table Name: ProfessionalDetails(Master Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| ProfId | Varchar2 | 15 | Primary Key |
| Username | Varchar2 | 10 | Foreign Key |
| DateOfBirth | Date |  |  |
| SkillId | Number | 8 | Foreign Key |
| EduTransId | Number | 8 | Foreign Key |
| ExperienceId | Number | 3 | Foreign Key |
| Marital Status | Varchar2 | 8 |  |
| Passport No | Number | 10 |  |
| Present Salary | Number | 8 |  |
| Passport No | Number | 15 |  |
| PptExpiryDate | Number | 10 |  |
| ExpectedSalary | Number | 8 | Not null |

**Table Name: EducationTransaction (Transaction Table)**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Width | Validation |
| EduTranId | Number | 15 | Primary Key |
| Username | Varchar2 | 10 | Foreign Key |
| UnivName | Varchar2 | 15 |  |
| Branch Name | Varchar2 | 15 |  |
| Division | Varchar2 | 10 |  |
| PassOutYear | Number | 8 |  |
| Percentage | Number | 4.2 |  |

**CLASS DIAGRAM FOR JOB PORTAL:**

****

**OBJECT DIAGRAM FOR JOB PORTAL**:

c:Companydetails

s:State

co:Country

ci:City

r: Registration

w:Webmaster

e:Edutrans

p:Professionaldetails

ex:Experience

ss:Skillset

q:Qualification

**USE CASE DIAGRAM FOR JOB FORTAL**:



**USE CASE DIAGRAM FOR JOB SEEKER** :



**USE CASE DIAGRAM FOR WEB MASTER :**



**USE CASE DIAGRAM FOR EMPLOYER:**



**MAIN SEQUENCE DIAGRAM FOR JOB PORTAL**:



**SEQUENCE DIAGEAM FOR JOBSEEKER**:



**SEQUENCE DIAGRAM FOR LOGIN**:



**SEQUENCE DIAGRAM FOR WEBMASTER CHECK**:



**SEQUENCE DIAGRAM FOR ORGANIZATION**:



**SEQUENCE DIAGRAM FOR WEBMASTER AND ORGANIZATION**:



**COLLABORATION DIAGRAM FOR JOB PORTAL**:



**COLLABORATION DIAGRAM FOR JOBSEEKER**:



**COLLABORATION DIAGRAM FOR LOGIN**:



**COLLABORATION DIAGRAM FOR WEBMASTER CHECK**:



**ACTIVITY DIAGRAM FOR JOB SEEKER**:



**ACTIVITY DIAGRAM FOR WEB MASTER**:



**ACTIVITY DIAGRAM FOR ORGANIZATION**:



**SWIMLAME DIAGRAM FOR JOBPORTAL**:



**COMPONENT DIAGRAM FOR JOBPORTAL:**

****