**THE USE-CASE:**

To provide a clear and unambiguous description of how the end-user and the system interact with one another.

Use-cases are defined from an actor’s point of view. An actor is a role that people (users) or devices play as they interact with the software. In the following proposed system the use-case will be:

**3.1 Use Case diagrams**

Many people are introduced to use cases via UML, which defines a graphical notation for representing use cases called the use case model. UML does not define standards for the written format to describe use cases, and thus many people have the misapprehension that this graphical notation defines the nature of a use case; however, a graphical notation can only give the simplest overview of a use case or set of use cases.The UML standard is the most popular standard for graphical notation of use cases. However, there are a number of alternative standards.Use Case diagrams identify the functionality provided by the system (use cases), the users who interact with the system (actors), and the association between the users and the functionality. Use Cases are used in the Analysis phase of software development to articulate the high-level requirements of the system. The primary goals of Use Case diagrams include:

* Providing a high-level view of what the system does
* Identifying the users ("actors") of the system
* Determining areas needing human-computer interfaces
* Use Cases extend beyond pictorial diagrams. In fact, text-based use case descriptions are often used to supplement diagrams, and explore use case functionality in more detail.

**3.1.1 Graphical Notation**

The basic components of Use Case diagrams are the Actor, the Use Case, and the Association.

|  |  |  |
| --- | --- | --- |
| **Actor** | An Actor, as mentioned, is a user of the system, and is depicted using a stick figure. The role of the user is written beneath the icon. Actors are not limited to humans. If a system communicates with another application, and expects input or delivers output, then that application can also be considered an actor. | ../../../Farid/13%20March/usecase.aspx_files/usecase-actor.png |
| **Use Case** | A Use Case is functionality provided by the system, typically described as verb+object (e.g. Register Car, Delete User). Use Cases are depicted with an ellipse. The name of the use case is written within the ellipse. | ../../../Farid/13%20March/usecase.aspx_files/usecase-usecase.png |
| **Association** | Associations are used to link Actors with Use Cases, and indicate that an Actor participates in the Use Case in some form. Associations are depicted by a line connecting the Actor and the Use Case. | ../../../Farid/13%20March/usecase.aspx_files/usecase-association.png |

**Relationships**

There are several types of relationships that may appear on use case diagram.

* an association between an actor and a use case
* an association between two use cases
* a generalization between two actors
* a generalization between two use cases

Associations are depicted as lines connecting two modeling elements with an optional open headed arrowhead on one end of the line indicating the direction of the initial invocation of the relationship .Generalizations are depicted as a close headed arrow with the arrow pointing towards the more general modeling element.

**SYSTEM BOUNDRY BOXES (optional)**

A rectangle can be drawn around the use case, called the system boundary box, to indicate the scope of the system. Anything within the box functionality that is in scope and anything outside the box is not. System boundary boxes are rarely used,

Packages (optimal) packages are UML contracts that enable you to organize model element into groups. Packages are depicted as file folders and can be use don any of the UML diagrams including both use case diagrams and class diagrams. Packages are used only when diagrams become unwieldy, which generally implies they can not printed on a single page to organize a large diagram in to smaller ones.

**3.2 Activity diagrams**

Activity diagrams represent the dynamics of the system. They are flow chart that is used show the work flow of a system. That is they show the flow of a system, what activities can be done parallel, and any alternative paths through the flow. An activity is the execution of a task whether it is a physical activity or the execution of code. Simply put the activity diagram shown the sequence of activities.

Like the simple flow chart, activity diagrams have support for conditional behavior, but have added support for parallel execution as well.

**Start:** each activity diagram has non start (above) at which the sequence of actions begins.

**End:** each activity diagrams has one finish at which the sequence of actions ends

**Activity**: are connected tougher by transition. Transitions are directed arrows flowing from the previous activity to the next activity. They are optionally accompanied by a textual label of the form:

Activity

**Label**

The **guard** is a conditional expression that when true indicates that transition is taken. The label is also optional and freeform.

To show conditional behavior uses a branch and a merge. The top diamond is a branch and has only one transaction flowing into it and any number of mutually exclusive transitions flowing out. That is the guards on the outgoing transitions must resolve themselves so that only one is followed. The merge is used to end the conditional behavior .there can be any number of incoming and only one outgoing ,transition ..

To show parallel behavior uses a fork and a join. The fork has one transition entering and any number of transition exiting .all of which will be take. The join represents the end of the parallel behavior and has any number of transitions entering an only one leaving.

**STATE DIAGRAM**

The state diagram shows the change of an object through time. Based upon events that occur the state diagram shown how the object changes from start to finish.

State another

Do/active state

States are represented as a rounded rectangle with the name of the state shown. Optionally you can include an activity that represented a longer running task during that state. Connecting states together are transitions. These represented the events that cause the object to change from one state to another. The guard clause of the label is again mutually exclusive and must resolve itself to be either true or false. actions represent tasks that run causing the transitions actions are different from activates in that action cannot be interrupted, while an activity can be interrupted by and incoming event .both ultimately represent and operation on the object being studied. For example an operation that sets an attribute would be considered and action, while a long calculations might be an activity. The specific separation between the two depends on the object and the system being studied. Like activity diagrams state diagrams, state diagram have one start and one end format which the state transitions start and end respectively.

**3.3 Sequence diagrams**

Sequence diagrams are used to show the flow of functionality through a use case. Actors and objects are shown at the top of the diagram. Each arrow represents a message passed between actor and object to or object and object to perform the needed functionality .it depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the object needed to carry out the functionality of scenario.

Sequence diagrams describe interaction among class in terms of an exchange of messages over time.

**Basic sequence diagram symbols and notations class roles**

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don’t list object attributes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Object** | Objects are instances of classes, and are arranged horizontally. The pictorial representation for an Object is a class (a rectangle) with the name prefixed by the object name (optional) and a semi-colon. | ../../../Farid/13%20March/sequence.aspx_files/sequence-object.png |  |
| **Actor** | Actors can also communicate with objects, so they too can be listed as a column. An Actor is modeled using the ubiquitous symbol, the stick figure. | ../../../Farid/13%20March/sequence.aspx_files/sequence-actor.png |
| **Lifeline** | The Lifeline identifies the existence of the object over time. The notation for a Lifeline is a vertical dotted line extending from an object. | ../../../Farid/13%20March/sequence.aspx_files/sequence-lifeline.png |
| **Activation** | Activations, modeled as rectangular boxes on the lifeline, indicate when the object is performing an action. | ../../../Farid/13%20March/sequence.aspx_files/sequence-activation.png |
| **Message** | Messages, modeled as horizontal arrows between Activations, indicate the communications between Objects. | ../../../Farid/13%20March/sequence.aspx_files/sequence-message.png |

**Loops** A repetition or loop within a sequence diagram is depicted as a rectangle .Place the condition for exiting the loop at the bottom left corner in square brackets [].

**3.4 Use Case Diagrams**

**Use Case Diagram for User/Job seeker**

Sign up

Visit site

View jobs

Search jobs

User/Job seeker

Apply for job

View NEWS

Sign out

**Use Case Diagram for Company**

Sign Up

Visit site

Add jobs

View posted Jobs

View applicants

View profile

Company/Employer

Edit profile

View news

Sign out

**Use Case Diagram for Administrator**

Log in Registered User

Sign Up

View user

Edit user

Delete user

View Company

Edit Company

Delete Company

Company/Employer

View jobs

Edit jobs

Post jobs

Delete jobs

Sign Out

**Use Cases for job seeker visit site**

**User Visit Site**

|  |  |
| --- | --- |
| **Name** | **Visit Site** |
| **Description** | This Use case is used when user wants to visits the site and wants to get the Information about the companies that are providing the jobs. And view the opportunities to Apply. |
| **Precondition** | The Keyword for the site must be entered carefully. |
| **Primary Flow** | When User opens the site (Main Page) All Information Affiliated with the site) is displayed to the user.  General Information of the jobs in that company is also provided. |
| **Post Condition:** | On Demand the required page is displayed. |

**Use Case of Sign Up**

**User Sign Up**

|  |  |
| --- | --- |
| **Name** | **Sign Up** |
| **Description:** | This Use case is used when user is wants to sign up and wants to be registered. Because to apply for a job. |
| **Precondition** | The keyword for the site must be opened. |
| **Primary Flow** | When user is new to the site and to get benefit from the site, user wants to be registered then he/she opens the registration form.  User Fills this form. User gives all necessary information which is required.  After completing this form, user submits this form and Gets registered. |
| **Post Condition:** | User is registered to the site and becomes an authorized user. |

**Use case for View Jobs**

**User View jobs**

|  |  |
| --- | --- |
| **Name:** | **View Jobs** |
| **Description** | This Use Case is used when user wants to view job of multiple categories. User start view for the required Job. |
| **Precondition** | The User must click on the link Current Jobs for finding required job. He/She may also see the jobs of different countries. |
| **Primary Flow** | When user clicks on the current jobs.  All related/required jobs are retrieved and displayed.  All detailed Information about the job and the job offering company is also displayed.  User Selects the required job.  The detailed Information about the job and job offering company is displayed. |
| **Post Condition:** | All related and vacant jobs are displayed to the user. |

**Use case for Searching Jobs**

**User Search for job**

|  |  |
| --- | --- |
| **Name:** | **Search for Job** |
| **Description** | This Use Case is used when user wants to search for a specific job and wants to apply for that job. User start search for the required Job. |
| **Precondition** | The User must write the title of specific job in a box and then click on button Search for job. |
| **Primary Flow** | When user clicks on the search for jobs.  All related/required jobs are retrieved and displayed.  All detailed Information about the job and the job offering company is also displayed.  User Selects the required job.  The detailed Information about the job and job offering company is displayed. |
| **Post Condition:** | All related and vacant jobs are displayed to the user. |

**Use Case to Apply for Job**

**User Apply for Job**

|  |  |
| --- | --- |
| **Name** | **Apply for Job** |
| **Description:** | This use case is used when user wants to apply for a job. User has to submit his application form. |
| **Precondition:** | The user must Sign in or Register to apply and then the User must click on the link Apply for applying specific job. |
| **Primary Flow** | After filling the provided application form, user wants to apply for job.  User selects a required job.  User submits his/her application for that job. |
| **Post Condition** | After validating the user’s information given in application. User is given response. |

**Use Case to View NEWS**

**View news**

**User**

|  |  |
| --- | --- |
| **Name** | **View news** |
| **Description** | This Use case is used when user want to see the NEWS. User can see news of different channels such as Google news, MSNBC news and Geo news. |
| **Precondition** | User must click on RSS news and must select the channel to which he wants to see news. |
| **Primary Flow** | When User click on RSS news, three channels are displayed to the user. He/she select one from which he/she wants to see news. Updated RSS files are fetched from the database of that channel site directly and displayed to the user. |
| **Post Condition** | On Demand the required page is displayed. |

**Sequence Diagrams for Job seeker**

**Sequence Diagram of Use Case Visit Site**

**VISIT PAGES**

**SITE MAIN PAGE**

**User**

REQUEST URL

VISIT SITE

EXIT

**Sequence Diagram of Use case Sign Up**

**WEB SERVER**

**USER MAIN PAGE**

**MEMBERSHIP**

**SITE MAIN PAGE**

**USER**

REQUEST URL

SIGN UP

PROMOT FOR INPUT

SUBMIT DATA

SUBMIT

PROMPT

**Sequence Diagram of Use Case View jobs**

**WEB SERVER**

**Current jobs list page**

**SITE MAIN PAGE**

**User**

REQUEST URL

Current jobs

Jobs Query

Exit

**Sequence Diagram of Use Case Search jobs**

**WEB SERVER**

**Current jobs list page**

**SITE MAIN PAGE**

**User**

REQUEST URL

Search jobs

Prompt for Input job title

Submit job title

Job search Query

Exit

**Sequence Diagram of Use Case Apply For Job**

**JOBS**

**WEB SERVER**

**USER**

REQUEST APPLY

SUBMIT

Exit

**Sequence Diagram of Use Case View news**

**Web server**

**SITE MAIN PAGE**

**NEWS PAGES**

**User**

**NEWS PAGES**

**NEWS PAGES**

REQUEST URL

RSS NEWS

PROMPT FOR SELECT CHANNEL

SELECT CHANNEL

PROMPT NEWS PAGES

**Activity Diagram for Job seeker**

**Activity Diagram for Visit Site Use Case**

**Site Main**

**Page**

**Display Visit**

**Page**

**Select Visit**

**Site**

**Activity Diagram for Sign Up**

**Save**

**Information**

**Submit Data**

**Input Data**

**Display Form**

**Select Apply**

**For Job**

**Site Main**

**Page**

**Activity Diagram for View Jobs Use Case**

**Site Main**

**Page**

**Display Error**

**Display List**

**Select Job**

**Illegal**

**Request**

**Activity Diagram for Search Jobs Use Case**

**Site Main**

**Page**

**Input job title**

**Submit**

**Illegal**

**Request**

**Display List**

**Display Error**

**Activity Diagram Apply for Jobs Use Case**

**Jobs**

**Select Apply**

**Submit**

**Activity Diagram for View NEWS Case**

**Site Main**

**Page**

**Display NEWS**

**Page**

**Select RSS NEWS**

**Select News channel**

**Use Cases for Company**

**Use Case for Visit Site**

Company Visit Site

|  |  |
| --- | --- |
| **Name** | **Visit Site** |
| **Description:** | This Use Case is used when company is seeking for an online portal to place their site and wants to visits this site. |
| **Precondition:** | The keyword for the site must be entered carefully. |
| **Primary Flow** | When company opens the site (Main Page) All Information Affiliated with the site is retrieved from the Database and displayed to the user. |
| **Post Condition:** | On Demand the required page is displayed. |

**Use Cases for Sign Up**

Company Sign Up

|  |  |
| --- | --- |
| **Name** | **Sign Up** |
| **Description:** | This use case is used when company is not registered and wants to be registered. Because to place a job, company must be registered to the site. |
| **Precondition:** | The keyword for the site must be entered carefully and registration form must be opened. |
| **Primary Flow** | When company is new to the site and to place its jobs, company wants to be registered then its opens the registration form company fills this form. And gives all necessary information which is required. After completing this form, it submits this form and Gets registered. |
| **Post Condition:** | Company is registered to the site and becomes and authorized user. |

**Use case for View Applicants**

**User View applicants**

|  |  |
| --- | --- |
| **Name:** | **View applicants** |
| **Description** | This Use Case is used when user wants to view applicants of a specific job. User start view for the required applicant. |
| **Precondition** | The User must click on the link APPLICANTS for finding required applicant. Employer may also see the profile of applicant. |
| **Primary Flow** | When user clicks on the applicants.  All related/required applicants are retrieved and displayed.  All detailed Information about the applicant is also displayed.  User Selects the required applicant detail  The detailed Information about the applicant is displayed. |
| **Post Condition:** | All related and applicants are displayed to the user. |

**Sequence Diagrams for Visit site**

**SITE MAIN PAGE**

**VISIT PAGES**

**Employer**

REQUEST URL

VISIT SITE

EXIT

**Sequence Diagram of Use case Sign Up**

**WEB SERVER**

**COPANY MAIN PAGE**

**MEMBERSHIP**

**SITE MAIN PAGE**

**Employer**

REQUEST URL

SIGN UP

PROMPT FOR INPUT

SUBMIT DATA

SUBMIT

PROMPT

**Sequence Diagram of Use Case Add Job**

**WEB SERVER**

**EMPLOYER MAIN PAGE**

**JOBS FORM**

**EMPLOYER**

REQUEST NEW/

PROMPT FOR INPUT

INPUT JOB

SAVE/UPDATE QUERY

**Sequence Diagram of Use Case View Applicant**

**COMPANY MAIN PAGE**

**APPLICANT**

**WEB SERVER**

**Employer**

REQUEST APPLICANT

REQUEST QUERY

PROMPT

**Activity Diagram for Employer**

**Activity Diagram for Visit Site Use Case**

**Site Main**

**Page**

**Display Visit**

**Page**

**Select Visit**

**Site**

**Activity Diagram for Sign Up**

**Save**

**Information**

**Submit Data**

**Input Data**

**Display Form**

**Select new user**

**Site Main**

**Page**

**Activity Diagram for Post Jobs Use Case**

**Employer Main**

**Page**

**Select Save/Post**

**Ask for Data**

**Select Post job**

**Activity Diagram for View applicants Use Case**

**Employer Main**

**Page**

**Display Applicants list**

**Select View applicants**

**Sequence Diagram of Use Case Post Jobs**

**WEB SERVER**

**ADMIN VIEW JOBS PAGE**

**JOBS FORM**

**ADMIN**

REQUEST NEW/EDIT

PROMPT FOR INPUT

INPUT JOB

SAVE/UPDATE QUERY

**Sequence Diagram of Use Case Edit/Update Information**

**WEB SERVER**

**INFORMATION**

**ADMIN VIEW PAGES**

**ADMIN**

REQUEST EDIT/UPDATE

PROMPT FOR INPUT

INPUT DATA

SUBMIT

**Sequence Diagram of Use Case Delete Information**

**WEB SERVER**

**ADMIN VIEW PAGES**

**ADMIN**

REQUEST DELETE

SAVE/DELETE QUERY

**Activity Diagram for Visit Site Use Case**

**Site Main**

**Page**

**Display Visit**

**Page**

**Select Visit**

**Site**

**Activity Diagram for Post Jobs Use Case**

**Admin Main**

**Page**

**Ask for Data**

**Select Post job**

**View jobs**

**Job**

**Select Save**

**Activity Diagram for Edit Information Use Case**

**User Main**

**Page**

**Input data**

**Display**

**Information**

**Select**

**Edit/Update**

**Select View**

**Page**

**Save Query**

**Activity Diagram for Delete Information Use Case**

**User Main**

**Page**

**Select View**

**Page**

**Display**

**Information**

**Select**

**DELETE**

**Save Query**

**DATABASE DESIGN**

**3.5.1 Logical Database Design and Relational Model**

Logical Database is a high performance fault tolerance database management system, especially designed for online transaction processing and large database applications. The database is mostly maintained by Structured Query Language (SQL), which is considered as a heart of database management system. Its popularity is due to case of use, flexibility and capability**.**

**Advantages**

* Provides easy access to all data.
* Flexibility in data modeling.
* Reduced data storage and redundancy.
* Independent of physical storage and logical database design.
* Provides a high-level data manipulation language

.

**3.6 Entity Relationship Diagram**

The data model consists of three interrelated pieces of information: the data object, the attributes that describe the data object, and the relationship that connect the data object to one another.

The object/relationship pair is the cornerstone of the data model. These pairs can be represented graphically using the entity / relationship diagram. The ERD was originally proposed by Peter Chen [CHE01] for the design of relational database systems and has been extended by the other. A set of primary components is identified for ERD: data objects, attributes, relationships, and various type indicators. The primary purpose of ERD is to represent the data object and their relationships [Ref1].

**3.7 Relations**

Relation defines in which way data objects are connected to the other object.

There are three types of relationship.

* One To One
* One To Many
* Many To Many

The relations in the tables can be defined as;

Online User and Customer have One-To-One relationship because each online user has only one entry in the customer file in that branch.

The relationship between the Customer and Account has One-To-Many relationship because one customer can be more than Introducer.

The relationship between the Online Customer and Transaction has One-To-Many relationship because one customer can be more than one Transaction.

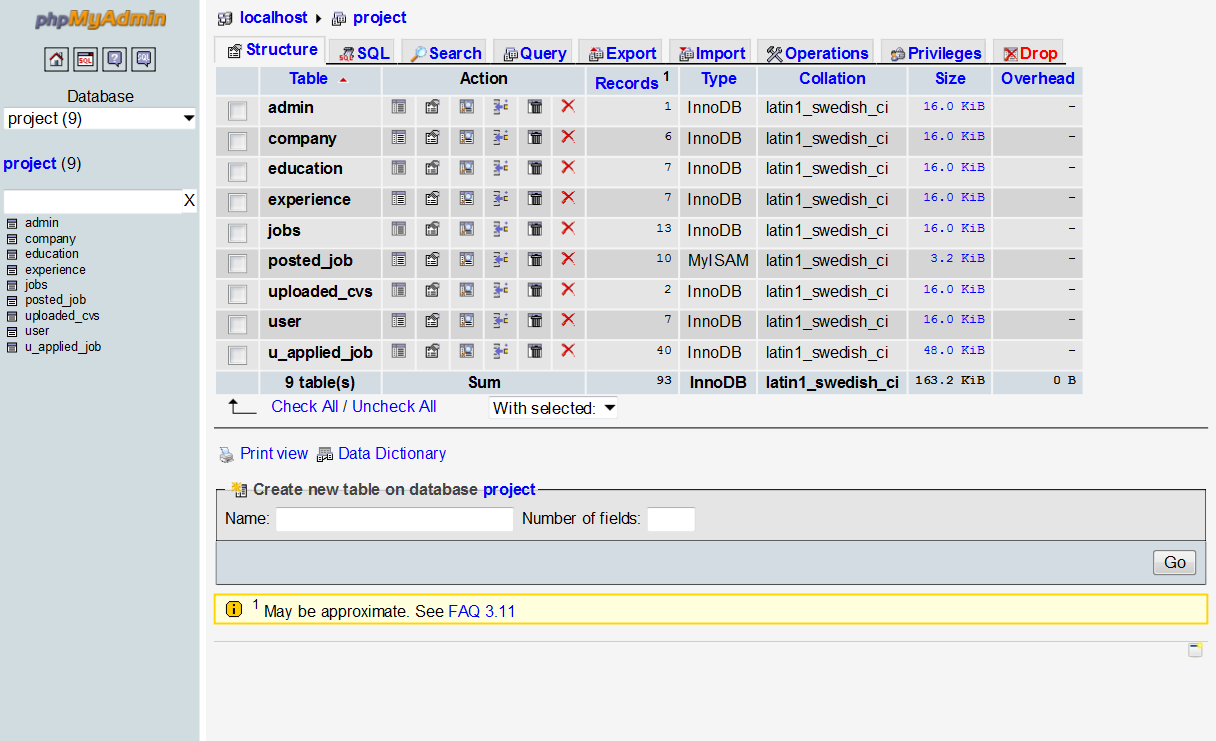
**Database design**

**3.8 Tables**

As a result, the System, Online jobs seeking uses a number of database files. The database is a collection of interrelated data about an enterprise with multiple users. The project, online jobs seeking, includes the following database files:

**Over all Database View**

Here all Tables of Database Are Visible.

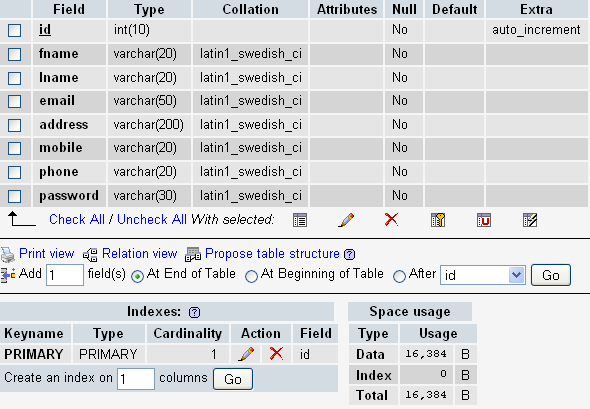


**Table Name:** **Admin**

**Primary Key:** **ID  
Description:**

This table is used to keep the personal data of Administrator. This table is used to create the account of an administrator and verify the authorization of administrator.

In this table ID is used as Primary Key.

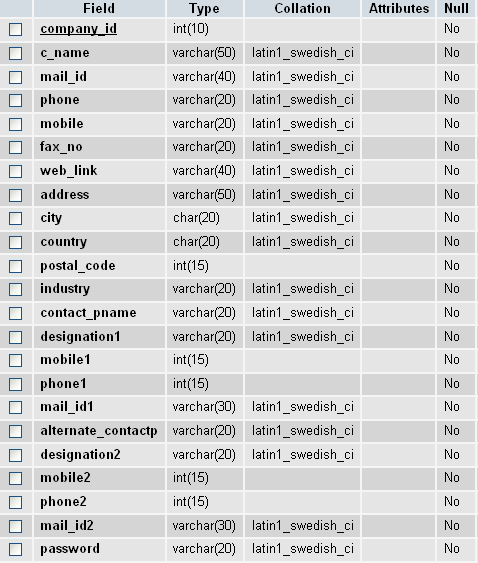


**Table Name: Company**

**Primary Key:** **company\_id**

**Description:**

This table is used to keep record of companies. This keeps the record of that of those companies who have registered themselves and issued jobs for the job seekers. In this table company\_id is used as primary key.

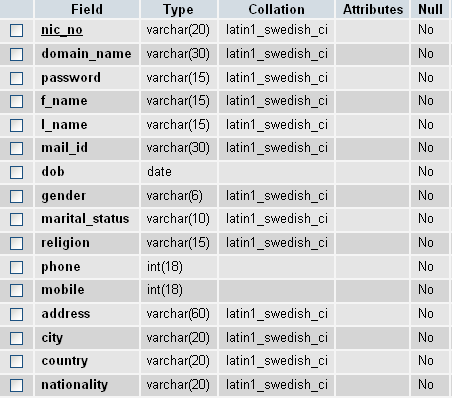


**Table Name:** **User**

**Primary key: Nic\_no**

**Description:**

This table is used to keep the information of each user who wants to apply for the job. There is required the necessary personal information. Here nic\_no is used as primary key.

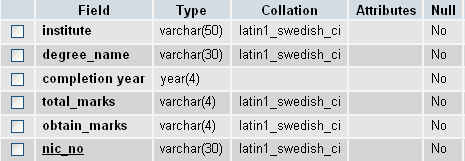


**Table name**: **Education**

**Primary key: nic\_no**

**Description:**

This table is used to keep the educational profile of user. This table is used for employer to view academic information of applicant. Nic\_no is used as primary key.

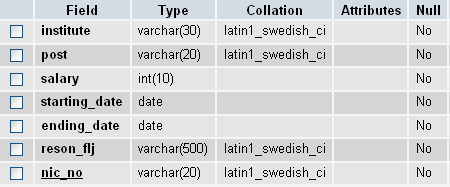


**Table Name:** **Experience**

**Primary key**: **Nic\_no**

**Description:**

This table is used to keep the experience profile of the job seekers which is also used for employer to view experience profile of applicants. Nic\_no is used as primary key.

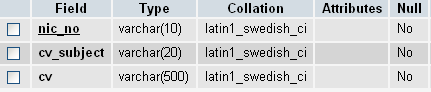


**Table name:** **Upload\_cvs**

**Primary key: nic\_no**

**Description:**

This table keeps the path CV’s uploaded by the jobseekers. Nic\_no is the primary key of that table.

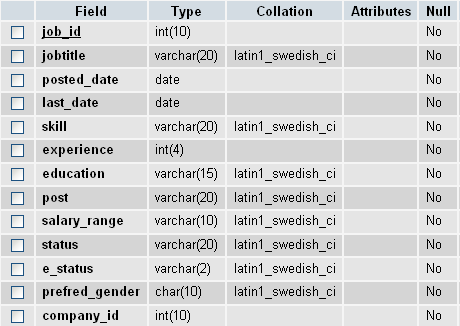


**Table name:** **jobs**

**Primary key**: **job\_id**

**Description:**

This table is used to save the detail of all the jobs posted by the Employers that is used by administrator to view that job’s detail. Here is job\_id is set as primary key.

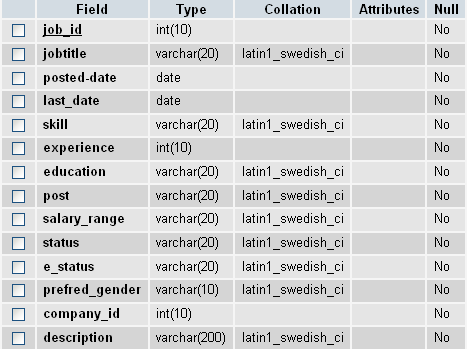


**Table name:** **posted**\_**jobs**

**Primary key**: **job\_id**

**Description:**

This table is used to save the detail of all the jobs activated by the administrator that is used by jobseeker to view that job’s detail. Here is job\_id is set as primary key.



**Table name:** u\_applied\_job

**Description:**

This table is used to keep the record of jobseekers (which jobseeker applied for which job).

