**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Building a web-based application that manages activities of the maid service**

|  |  |
| --- | --- |
| **Group 6** | |
| **Group member** | Bùi Tiến Tuân - Leader - SE60824  Mạnh Quang Tuyến - Member - SE60890  Trương Hải Đăng - Member - SE60841 (Droped out)  Nguyễn Tấn Công - Member - SE60920 |
| **Supervisor** | Mr. Nguyễn Trọng Tài |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | MS-Website |

- Ho Chi Minh City, December 2014-

***ACKNOWLEDGEMENTS***

After three exhausted months of so many hardships and challenges with capstone project, we finally finish the trial to reach this moment of happiness. There are a lot of feelings we would like to express.

Firstly, our best gratitude to our capstone project's supervisor Mr.Nguyễn Trọng Tài for his support and guidance during this project work. Not only did he give us many technical supports but also he gave us mental supports as well. There were also hard times when all of us felt down and despair due to our incompetence but Mr.Tài gave us encouragement and the ways to solve our problem. We all love and respect him as much as we want to pass this course.

Next, we would like to give our thanks to other teacher who have been teaching us for four years of our university life. We cannot make it here today without useful knownledges they have taught us. We wish them the best of luck in life to keep working as educators who bring students like us to a bright future.

Our gratitude also expands to all the FPT University Committee and Department staff who provided us the best environment to study and develop ourself. It contributes important parts for us to improve our skill and knowledge as well as boost our creativity to the utmost.

We also want to give our grateful thanks to our fellow friends who have been with us during four years of studying and especially the capstone project's time to support us as well as disturb our work a lot too but thank to that, we have the enthusiasm and spirit to keep trying our best.

Lastly, we wish to thank our parents for their caring and supporting during our studying time generally and capstone project particularly. With this work, we have somehow repaid them partially with our growing up.

In conclusion, we are very proud of our work and we wish for its success in Capstone Project Defense Ceremony so that we will have nothing to regret.

# Table of Contents

[Table of Contents 4](#_Toc408421155)

[List of Tables 6](#_Toc408421156)

[List of Figures 7](#_Toc408421157)

[Definitions, Acronyms, and Abbreviations 8](#_Toc408421158)

[1 Introduction 9](#_Toc408421159)

[1.1 Project Information 9](#_Toc408421160)

[1.2 Introduction 9](#_Toc408421161)

[1.3 Current Existence Method 9](#_Toc408421162)

[1.4 System Overview 9](#_Toc408421163)

[1.5 Role and Responsibility 9](#_Toc408421164)

[2 Software Project Management Plan 10](#_Toc408421165)

[2.1 Problem Definition 10](#_Toc408421166)

[2.1.1 Name of this Capstone Project 10](#_Toc408421167)

[2.1.2 Problem Abstract 10](#_Toc408421168)

[2.1.3 Project Overview 10](#_Toc408421169)

[2.2 Project organization 11](#_Toc408421170)

[2.2.1 Software Process Model 11](#_Toc408421171)

[2.2.2 Roles and responsibilities 12](#_Toc408421172)

[2.2.3 Tools and Techniques 13](#_Toc408421173)

[2.3 Project Management Plan 14](#_Toc408421174)

[2.3.1 Task 14](#_Toc408421175)

[2.3.2 All Meeting Minutes 18](#_Toc408421176)

[2.4 Coding Convention 18](#_Toc408421177)

[3 Software Requirement Specification 18](#_Toc408421178)

[3.1 User Requirement Specification 18](#_Toc408421179)

[3.1.1 Guest requirement 18](#_Toc408421180)

[3.1.2 Authenticated User requirement 18](#_Toc408421181)

[3.1.3 Staff requirement 18](#_Toc408421182)

[3.1.4 Customer requirement 19](#_Toc408421183)

[3.1.5 Maid mediator requirement 19](#_Toc408421184)

[3.1.6 Webmaster requirement 20](#_Toc408421185)

[3.1.7 System requirement 20](#_Toc408421186)

[3.2 System Requirement Specification 20](#_Toc408421187)

[3.2.1 External Interface Requirement 20](#_Toc408421188)

[3.2.2 Functional Requirement 22](#_Toc408421189)

[3.2.3 Non-functional Requirement 45](#_Toc408421190)

[3.3 Entity Relationship Diagram 47](#_Toc408421191)

[3.3.1 Diagram 47](#_Toc408421192)

[3.3.2 Data Dictionary 47](#_Toc408421193)

[4 Software Design Description 49](#_Toc408421194)

[4.1 Design Overview 49](#_Toc408421195)

[4.2 System Architectural Design 49](#_Toc408421196)

[4.3 Component Diagram 49](#_Toc408421197)

[4.4 Detailed Description of Components 51](#_Toc408421198)

[4.4.1 Class Diagram 51](#_Toc408421199)

[4.4.2 Sequence Diagram 58](#_Toc408421200)

[4.5 Database Design 63](#_Toc408421201)

[4.5.1 Physical Database Design 63](#_Toc408421202)

[4.5.2 Data Dictionary 64](#_Toc408421203)

[4.6 Algorithms 71](#_Toc408421204)

[4.6.1 K-Means algorithms 71](#_Toc408421205)

# List of Tables

[Table 1: Roles and Responsibility Details 16](#_Toc408420153)

[Table 2: Task 20](#_Toc408420154)

[Table 3: Data Dictionary 149](#_Toc408420155)

# List of Figures

[Figure 1: Waterfall model (Introduction software engineering book) 11](#_Toc408421206)

[Figure 2: System Overview Use Case 22](#_Toc408421207)

[Figure 6: < Authenticated user >Advance Search 23](#_Toc408421208)

[Figure 17: <Webmaster> Add Staff 24](#_Toc408421209)

[Figure 25: <System> Suggest JobRequest 26](#_Toc408421210)

[Figure 26: <System> Send SMS 27](#_Toc408421211)

[Figure 28: <Customer> View JobRequest 29](#_Toc408421212)

[Figure 29: <Customer> Apply JobRequest 29](#_Toc408421213)

[Figure 30: <Customer> Comment JobRequest 31](#_Toc408421214)

[Figure 31: <Customer> Rate JobRequest 32](#_Toc408421215)

[Figure 32: <Customer> View Profile 34](#_Toc408421216)

[Figure 37: <Customer> Post Recruitment 34](#_Toc408421217)

[Figure 39: <Customer> Pay 36](#_Toc408421218)

[Figure 50: <Staff, MaidMediator> Post JobRequest 38](#_Toc408421219)

[Figure 63: Entity Relationship Diagram 47](#_Toc408421220)

[Figure 64: MVC Pattern 49](#_Toc408421221)

[Figure 65: Component Diagram 50](#_Toc408421222)

[Figure 66: Class Diagram 51](#_Toc408421223)

[Figure 71: <All User> Search 58](#_Toc408421224)

[Figure 74: <Customer> Apply Job Request 59](#_Toc408421225)

[Figure 79: <Customer> Post Recruitment 60](#_Toc408421226)

[Figure 82: <Customer> Rate Job Request 60](#_Toc408421227)

[Figure 83: <Customer> Rate Job Request 61](#_Toc408421228)

[Figure 92: <Staff, Maid Mediator> Post Job Request 61](#_Toc408421229)

[Figure 87: <Staff, Maid Mediator> Register Job Request 62](#_Toc408421230)

[Figure 99: Physcial Database Diagram 63](#_Toc408421231)

[Figure 94: K-Means algorithm flow chart 72](#_Toc408421232)

# Definitions, Acronyms, and Abbreviations

|  |  |  |
| --- | --- | --- |
| **No.** | **Abbreviation & Acronym** | **Definition** |
| 1 | MS | Maid Service |
| 2 | OS | Operation System |
| 3 | Admin | Administrator |
| 4 | HTTP | Hyper Text Transfer Protocol |

# Introduction

## Project Information

* **Project name:** Maid Services
* **Project Code:** MS-Website
* **Product Type:** Web-based Application
* **Start Date:** September 9, 2014
* **End Date:** December 20, 2014

## Introduction

The main goal of MS Website is to help users to easily find suitable person for taking care of household chores as well as help maids to quickly find a suitable job.

## Current Existence Method

Currently, most of job centers are supplying maid service with many functions such as posting maid’s information, maid searching by criteria (expected salary, age, experiences …) but the customers can’t post their private requests and give rating, comment about the maid that they have recruited. Moreover, maid’s information is not detailed and complete. The system can’t match and suggest the maid to the customer, negotiation and signing the contract are time-consuming.

## System Overview

The system has two main targets:

* **Customer:** busy with works and want to find people who can take care of houseworks for them. Customer can post a recruitment with demanded skills and the system will automatically match that recruitment with suitable job requests. Customer can choose one of them to apply.
* **Maid:** has experience in doing houseworks and want to find a job but doesn’t have knownledge about IT. They can come to talk directly with staff to post a job request or ask a person with knownledge about IT to become a mediator. The mediator can post new job request but still has to pay staff directly to active the posted job request.

## Role and Responsibility

|  |  |  |
| --- | --- | --- |
| **Role and Responsibilities** | **Full name** | **E-mail** |
| Instructor | Nguyễn Trọng Tài | taint@fpt.edu.vn |
| Leader | Bùi Tiến Tuân | tuanbtse60824@fpt.edu.vn |
| Member | Mạnh Quang Tuyến | tuyenmqse60890@fpt.edu.vn |
| Member | Trương Hải Đăng (drop out) | dangthse60841@fpt.edu.vn |
| Member | Nguyễn Tấn Công | congntse60920@fpt.edu.vn |

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* The system is for maid and the people who are searching for maid.
* The language is Vietnamese.
* The end product includes: the website and documents involved with the system.

### Problem Abstract

* Nowadays, the maid services aim to two main characters: customer and maid.
  + *Customer:* they can search maid, apply maid and payment.
  + *Maid:* they can search customer and post job request to apply themselves to the job.
* Moreover, there are not many websites can automatic suggest suitable maid for customer, and those websites require that the maids have to have knowledge about information technology.
* So, this website system is developed and focus on using algorithsm to automatic suggest suitable maid for customer. This website also not require maids have knowledge about informatioon technology because this website has a role call “maid mediator”, they can post job request for maid.

### Project Overview

#### The Current System

There are have many website about work such as http://timviec.com, timviecnhanh.com, vietnamworks.com …. But they have some problem:

* **Advantages:** Provider many information about works in Vietnam.
* **Disadvantages:**
  + There is so much information.
  + There is no centralized information about Maid.

#### The Proposed System

Our site focuses on those who are busy with their works so they need to find maids to help them with household chores and the maids who are searching for jobs. Maids can quickly find job for themselves and it also makes it easy for people who are in need of finding maids to choose and recruit. With small amount of money, the website can connect those two together. The maids are people who don’t have much knowledge about IT, so our purpose is to create a website with ease of use and simple.

There are three main users whom the website interacts with including:

* **Maid mediator:** Because of lacking of knowledge in IT, they can ask someone has knowledge to become their representation and register for them. Manage requested job, Search jobs, Apply job, View maid statistic.
* **Customer:** Manage recruitment, View customer statistic, Comment, Search maids, Rating, Apply maid.
* **Staff:** Can help maid to register and post info in place of them. Manage post time, Confirm request, Manage fee.

#### Boundaries of the System

* The system is for maid and the people who are searching for maid.
* The language is Vietnamese.
* The end product includes: the website and documents involved with the system.

#### Development Environment

##### Hardware requirements

**For Server**

|  |  |  |
| --- | --- | --- |
| Windows | Minimum Requirements | Recommended |
| Internet Connection | 512Kbs | 1Mbs |
| Operating System | Window 7 | Window 8 |
| Computer Processor | Intel® Core 2 Duo | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| Computer Memory | 2GB RAM | 3GB or more |

##### Software requirements

* Microsoft Windows 7: Operating System.
* SQL Server 2008 R2: used for managing the database for system.
* StarUML 5.0: designing models and diagrams tools.
* Visual Studio 2012: used to implement website and web service.
* Google Code & TortoiseSVN: used for source control.

## Project organization

### Software Process Model

With the schedule of weekly reports for every stage, the software will be developed by using waterfall model it is very simple and require minimal resource for implementation. With waterfall model, developing process will include five main phases:

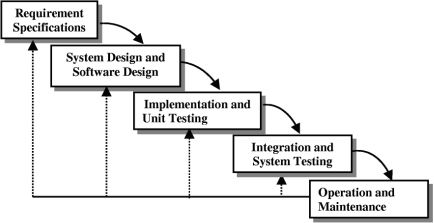


Figure 1: Waterfall model (Introduction software engineering book)

* **Requirement Specifications**: The first phase is also the most important phase of the process which may cause great effect to other phases. This phase includes gathering as much as possible requirements from customer, or other information source like World Wide Web, and producing a most detail and accurate software definition.
* **System Design and Software Design**: This phase is fundamental for implementation phase. Based on customer’s requirements to create logical modules, and definite their inter relations. Using algorithm and diagram to describe implementation of those modules.
* **Implementation**: This phase consists of actually constructing the product as per the design specification(s) developed in the previous step. Typically, this step is performed by a development team consisting of programmers, interface designers and other specialists, using tools such as compilers, debuggers, interpreters and media editors.
* **Testing:**
  + **System implementation and Unit testing**: Developing software modules follow detail designs, and doing unit testing for each module.
  + **Integration and System testing**: Testing output, performance in modules integrating process, and retests all functions of whole system.
  + **User acceptance testing**: A proper execution of all the preceding stages ensures an application as per the provided requirements and most importantly, it ensures a satisfied client.
* **Operation and Maintenance**: After testing completely, the software is handle over client, developing team will respond for maintenance of the system.

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Nguyễn Trọng Tài | Project manager | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | Bùi Tiến Tuân | Team Leader, BA, DEV, Tester | * Managing process * Assign task for team member * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **3** | Mạnh Quang Tuyến | Team Member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **4** | Trương Hải Đăng | Team Member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **5** | Nguyễn Tấn Công | Team Member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |

Table 1: Roles and Responsibility Details

### Tools and Techniques

#### For Development

* Front-end technologies: HTML5, CSS3, JavaScript, jQuery, AJAX.
* Back-end: ASP .NET MVC4, Entity Framework.
* Data Management System: SQL Server 2008 R2.
* IDE: Visual Studio 2012
* System design tool: StartUML
* IIS: web server

#### For Management

-Process and code tracking: Team Foundation Server.

-Tortoise SVN: Document version control and tracking.

## Coding Convention

C#: Using to develop MS-website.

Summary:

* Naming Convention.
* Indentation.
* Declaration.
* Code Examples.

All “Code Conventions for the C# Programming Language, by Microsoft

<http://msdn.microsoft.com/en-us/library/ff926074.aspx>

# Software Requirement Specification

## System Requirement Specification

### External Interface Requirement

#### User Interface

* The general interface of website is simple and simplifies user interfaces.
* The design is simple and monochrome is more prefer.
* The layout of information in page is showing simple but full of functions.
* The error, warning and notification messages must be simple, neat, and easy to understand. Error warning does not discomfort to the user.
* The working layout of user is spacious.
* Member can contact with manager and admin easily.

#### Hardware Interface

There is no extra hardware interfaces are needed. The system will use the standard hardware and data communications resources of a standard computer.

#### Software Interface

Browser: Chrome, Firefox

#### Communication Protocol

MS website uses HTTP/HTTPS protocol for communication with the web browser and the web server. In addition, TCP/IP network protocol for communication with HTTP protocol.

### Functional Requirement

#### System Overview Use Case

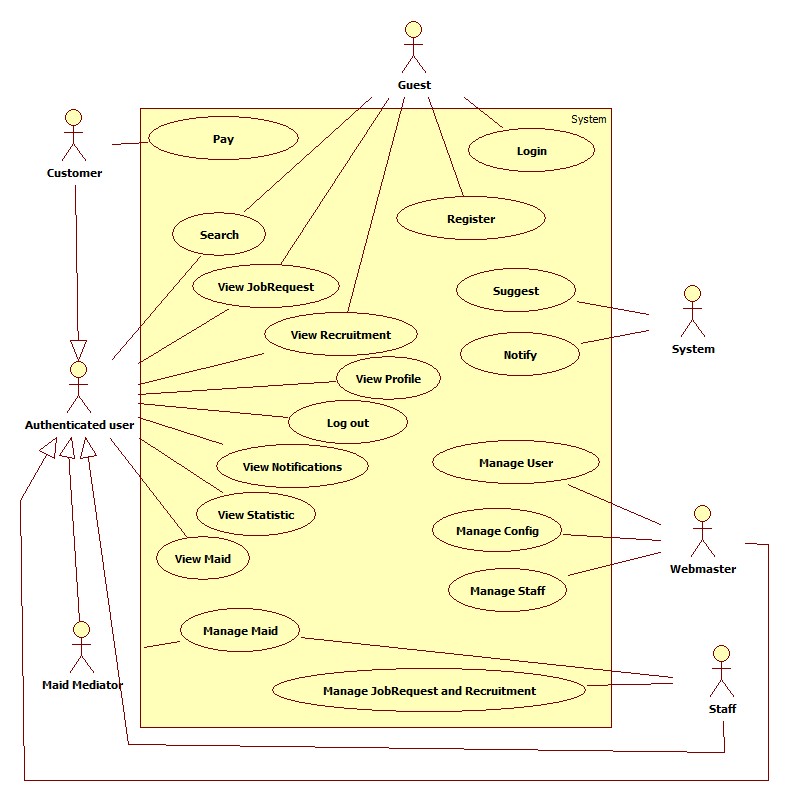


Figure 2: System Overview Use Case

### Non-functional Requirement

#### Usability

##### Graphic User Interface

* All the text, label and image in user page should be in Vietnamese.
* All the text, label and image in admin page should be Vietnamese.

##### Usability

* Website admin, supervisor and roundman only need more than one hour to train.

##### Installation

* The system must be easy to deploy. Customer can deploy successfully and learn to configure, maintain the system within one day of training.
* The attached manual guide must be clear. User can read and do themselves without developer’s help.

#### Reliability

* There is no requirement for system maintenance task from the user.
* Mean Time Between Failures (MTBF): more than 6 months.
* Accuracy: 100%.
* Maximum Bugs and Defect Rate: 0.3 bugs per thousand lines of code (0.3bugs/KLOC).
* Critical bugs:
* Loss of data: not any

#### Availability

* The server shall be working 24 hours per day and 7 days per week.

#### Security

* *All sensitive* information (password, etc.) must be hashed when storing in database and during transmission over networks using MD5 hash.
* Validate input data in SQL query before execute to avoid SQL Injection, XSS
* The role of user and member is clearly.

#### Maintainability

* All code shall fully document. All program files shall include comments concerning authorship and date of last change.
* The code shall be modular to permit future modifications.

#### Portability

* N/A

#### Performance

* Large tables and indexes must be partitioned data into smaller, more manageable sections by using partition in SQL Server 2008 R2.

## Entity Relationship Diagram

### C:\Users\USER\Desktop\ms-website\03.Documents\ERD\MS-ERD_1.0.pngDiagram

Figure 63: Entity Relationship Diagram

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| Entity Name | Description |
| Account | Describe all account of MS - Website system. |
| Staff | Describe all staff of website. |
| Customer | Describe all customer of website. |
| MaidMediator | Describe all maid mediator of website. |
| Maid | Describe all maid of website. |
| JobRequest | Describe all job request, with was posted by staff and maid mediator. |
| Recruitment | Describe all recruitment, withc was posted by customer. |
| SkillCategory | Describe all category of skill. |
| SkillInstance | Describe all instance of skill. |
| SkillReference | Describe all reference of skill of all job requests and recruitments. One job request or recruitment refers to one skill reference and that reference have all informations of that recruitment or job request. Each column of skill in skill reference is a skill instance. |

# Software Design Description

## Design Overview

* This document describes the technical and UI design of the MS Wesite. It includes the architectural design and the detailed design of common functions and business functions. It also includes the design of database model.
* The architectural design describes the overall architecture of the system, and the architecture of each main component and subsystem. It will describe the patterns being used, the role of each component and the role of the system in the working environment.
* The detailed design describes static and dynamic structure for each component and function. It includes class diagrams, class explanations, and sequence diagrams of the main use cases.UML 2.0.
* The database design describes the relationship between entities, and details of each entity.
* The user interface design describes the layout of the system, and some design for the screens.

## System Architectural Design

In MVC pattern, the functions of model, view, and controller are defined as in following figure.

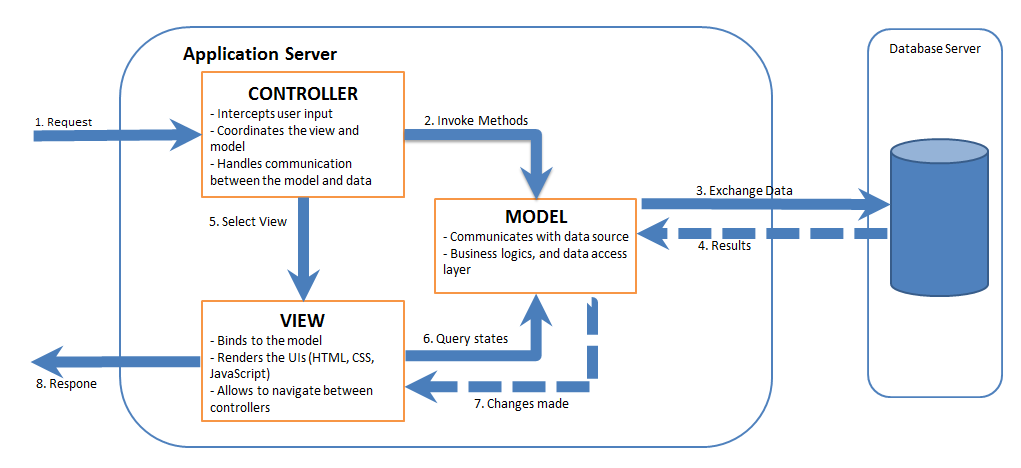


Figure 64: MVC Pattern

## Component Diagram

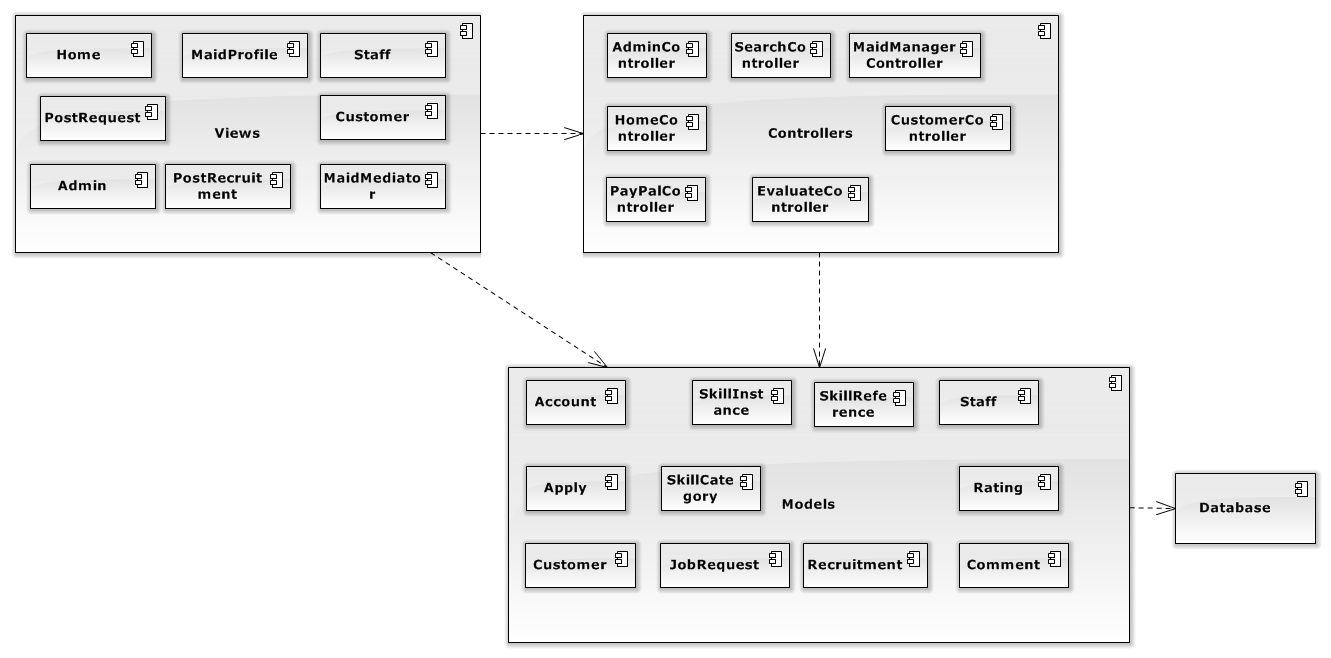


Figure 65: Component Diagram

## Detailed Description of Components

### Class Diagram

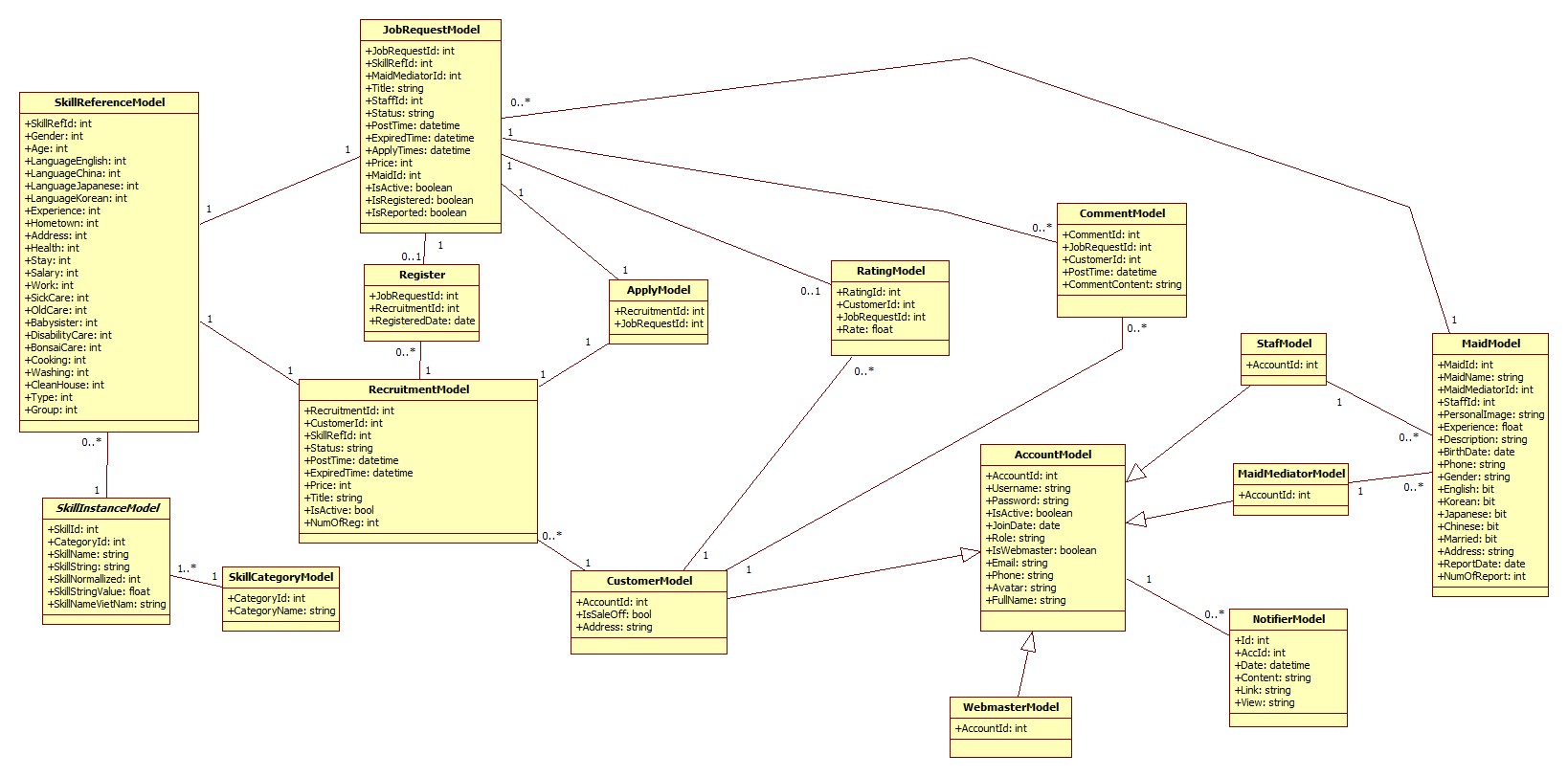


Figure 66: Class Diagram

#### Class Diagram – Entity Framework Model

##### AccountModel

This class is mapped with entity Account

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| AccountId | Integer | Public | Id of Account |
| Username | String | Public | Username of user |
| Password | String | Public | Password of user |
| IsActive | Boolean | Public | Check account is active |
| JoinDate | Date | Public | Date that user create an account |
| Role | String | Public | Role of User |
| IsWebmaster | Boolean | Public | Check account is an admin |
| Avatar | String | Public | Image of user |
| Email | String | Public | Email of user |
| Phone | String | Public | Phone of user |
| FullName | String | Public | Fullname of user |

* **Methods** None

##### ApplyModel

This class is mapped with relation Apply

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| RecruitmentId | Integer | Public | Id of recruitment |
| JobRequestId | Integer | Public | Id of job request |

* **Methods** None

##### CommentModel

This class is mapped with relation Comment

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| CommentId | Integer | Public | Id of comment |
| CustomerId | Integer | Public | Id of customer |
| JobRequestId | Integer | Public | Id of job request |
| PostTime | Datetime | Public | Time that comment was posted |
| CommentContent | String | Public | Content of comment |

* **Methods** None

##### CustomerModel

This class is mapped with entity Customer

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| AccountId | Integer | Public | Id of field |
| Address | String | Public | Address of customer |
| IsSaleOff | Bool | Public | Sale off |

* **Methods** None

##### JobRequestModel

This class is mapped with entity JobRequest

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| JobRequestId | Integer | Public | Id of job request |
| SkillRefId | Integer | Public | Id of skill reference |
| MaidMediatorId | Integer | Public | Id of maid mediator |
| StaffId | Integer | Public | Id of staff |
| Status | String | Public | Status of job request |
| PostTime | Datetime | Public | Time that job request was post |
| ExpiredTime | Datetime | Public | Time that job request was expired |
| ApplyTime | Datetime | Public | Time that job request was applied |
| MaidId | Integer | Public | Id of maid |
| IsActive | Boolean | Public | Check that job request is active |
| Price | Integer | Public | Price of jobrequest |
| Title | String | Public | Name of jobrequest |
| IsRegistered | Boolean | Public | Check that job request is registered |
| IsReported | Boolean | Public | Check that job request is reported |

* **Methods** None

##### MaidModel

This class is mapped with entity Maid

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| MaidId | Integer | Public | Id of maid |
| MaidName | String | Public | Name of maid |
| MaidMediatorId | Integer | Public | Id of maid mediator |
| StaffId | Integer | Public | Id of staff |
| PersonalImage | String | Public | Image of maid |
| Experience | Float | Public | Experience of maid |
| Description | String | Public | Description of maid |
| BirthDate | Date | Public | Birthdate of maid |
| Phone | String | Public | Phone of maid |
| Gender | String | Public | Gender of maid |
| English | Bit | Public | English |
| Korean | Bit | Public | Korean |
| Japanese | Bit | Public | Japanese |
| Chinese | Bit | Public | Chinese |
| Married | Bit | Public | Married |
| Address | String | Public | Address of maid |
| ReportDate | Date | Public | Date report |
| NumOfReport | Int | Public | Number of reports |

* **Methods** None

##### StaffModel

This class is mapped with entity Staff

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| AccountId | Integer | Public | Id of maid manager |

* **Methods** None

##### MaidMediatorModel

This class is mapped with entity MaidMediator

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| AccountId | Integer | Public | Id of maid manager |

* **Methods** None

##### NotifierModel

This class is mapped with entity Notifier

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| Id | Integer | Public | Id of notifier |
| AccId | Integer | Public | Id of account |
| Date | Datetime | Public | Date of notifier |
| Content | String | Public | Content of notifier |
| Link | String | Public | Link notify |
| View | Bool | Public | Confirm that view yet |

* **Methods** None

##### RegisterModel

This class is mapped with relation Register

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| RecruitmentId | Integer | Public | Id of recruitment |
| JobRequestId | Integer | Public | Id of job request |
| RegisteredDate | Date | Public | Date that job request is registered |

* **Methods** None

##### RatingModel

This class is mapped with relation Rate

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| RatingId | Integer | Public | Id of rating |
| CustomerId | Integer | Public | Id of customer |
| JobRequestId | Integer | Public | Id of job request |
| Rate | String | Public | Rate |

* **Methods** None

##### RecruitmentModel

This class is mapped with entity Recruitment

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| RecruitmentId | Integer | Public | Id of recruitment |
| SkillRefId | Integer | Public | Id of skill reference |
| CustomerId | Integer | Public | Id of customer |
| Status | String | Public | Status of recruitment |
| PostTime | Datetime | Public | Time that recruitment was posted |
| ExpiredTime | Datetime | Public | Time that recruitment was expired |
| Title | String | Public | Name of recruitment |
| Price | Int | Public | Price of recruitment |
| IsActive | Bool | Public | Check recruitment is active yet |
| NumOfReg | Int | Public | Number of register |

* **Methods** None

##### SkillCategoryModel

This class is mapped with entity SkillCategory

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| CategoryId | Integer | Public | Id of category |
| CategoryName | String | Public | Name of category |

* **Methods** None

##### SkillInstanceModel

This class is mapped with entity SkillIntance

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| SkillId | Integer | Public | Id of skill |
| CategoryId | Integer | Public | Id of category |
| SkillName | String | Public | Name of skill |
| SkillString | String | Public | The charater value of skill |
| SkillNormallied | Int | Public | The range value of skill |
| SkillStringValue | Float | Public | The number value of skill name |
| SkillNameVietNam | String | Public | The name of skill in Vietnamese |

* **Methods** None

##### SkillReferenceModel

This class is mapped with entity SkillReference

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| SkillRefId | Integer | Public | Id of Skill reference |
| Gender | Integer | Public | The value gender of skill |
| Age | Integer | Public | The value age of skill |
| LaguageEnglish | Integer | Public | The value language english of skill |
| LanguageChina | Integer | Public | The value language china of skill |
| LanguageJapanese | Integer | Public | The value language japanese of skill |
| LanguageKorean | Integer | Public | The value language korean of skill |
| Experience | Integer | Public | The value experience of skill |
| Hometown | Integer | Public | The value hometown of skill |
| Address | Integer | Public | The value address of skill |
| Health | Integer | Public | The value health of skill |
| Stay | Integer | Public | The value stay of skill |
| Salary | Integer | Public | The value salary of skill |
| Work | Integer | Public | The value work of skill |
| SickCare | Integer | Public | The value sickcare of skill |
| OldCare | Integer | Public | The value oldcare of skill |
| BabySister | Integer | Public | The value babysister of skill |
| DisabilityCare | Integer | Public | The value disabilitycare of skill |
| BonsaiCare | Integer | Public | The value bonsaicare of skill |
| Cooking | Integer | Public | The value cooking of skill |
| Washing | Integer | Public | The value washing of skill |
| Cleanhouse | Integer | Public | The value cleanhouse of skill |
| Type | Integer | Public | Skill of job request or skill of recruitment |
| Group | Integer | Public | Divide group in algorithm |

* **Methods** None

##### WebMasterModel

This class is mapped with entity WebMaster

* **Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Visibility** | **Description** |
| AccountId | Integer | Public | Id of account |

* **Methods** None

## Database Design

### Physical Database Design

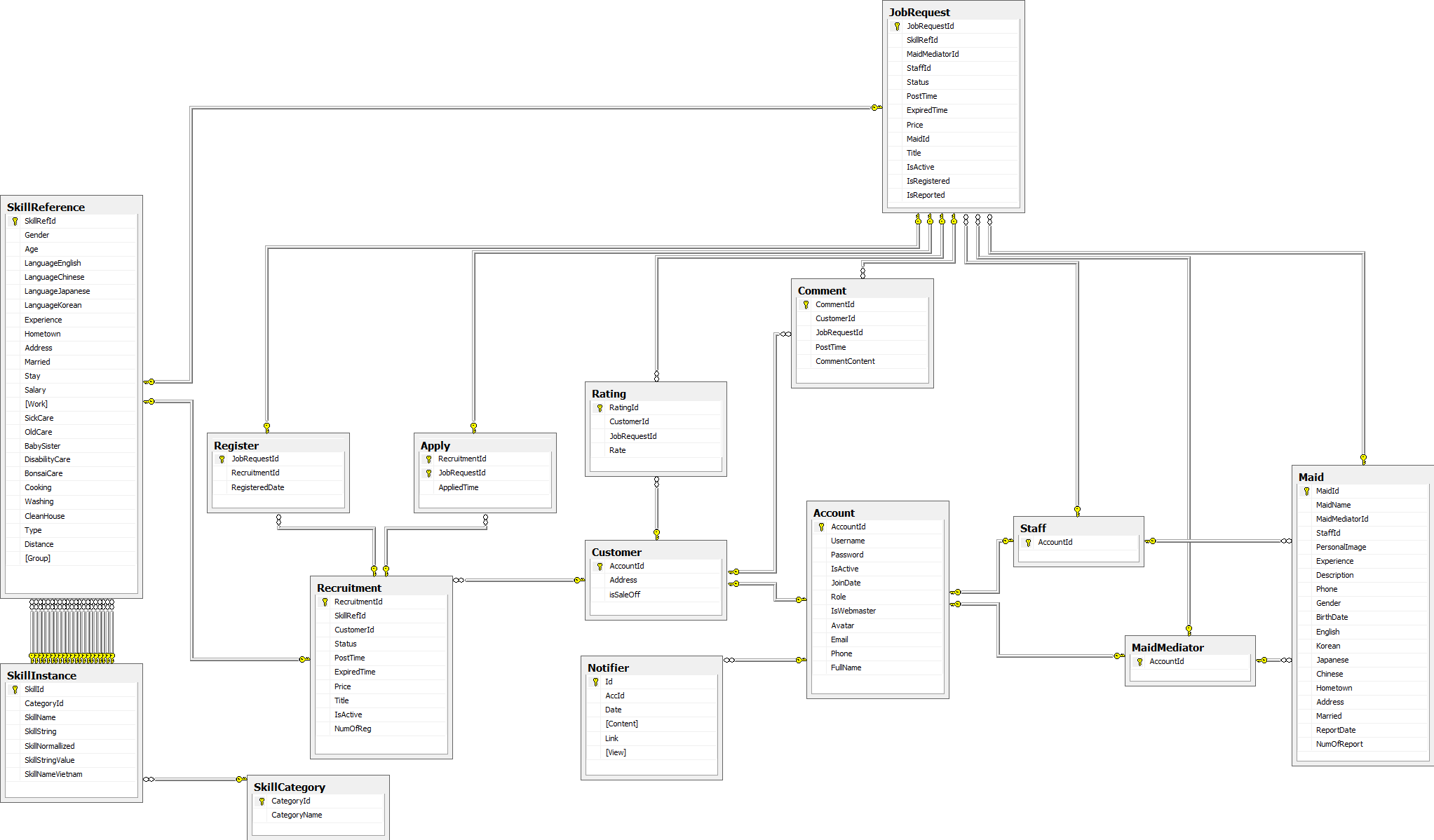


Figure 99: Physcial Database Diagram

### Data Dictionary

|  |  |  |
| --- | --- | --- |
| **No.** | **Table name** | **Description** |
| 1 | Account | Describe about users in system |
| 2 | Apply | Describe Apply |
| 3 | Comment | Describe all comment |
| 5 | Customer | Describe all customers informations |
| 6 | JobRequest | Describe all job requests |
| 5 | MaidMediator | Describe all maid mediators information |
| 7 | Maid | Describe all maids information |
| 8 | Rating | Describe all rating |
| 9 | Recruitment | Describe all recruitments |
| 10 | SkillCategory | Describe all SkillCategory |
| 11 | SkillIntance | Describe all SkillIntance |
| 12 | SkillReference | Describe all SkillReference |
| 13 | Staff | Describe all staffs information |
| 14 | Notifier | Describe all notifications |

Table 3: Data Dictionary

#### Acount

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| AccountId | int |  | PK |  | The index of this account |
| Username | nvarchar(50) |  |  |  | The username of the account |
| Password | nvarchar(50) |  |  |  | The password of the password |
| IsActive | bit |  |  |  | Admin can ban account |
| JoinDate | date |  |  |  | The date when this user join |
| Role | nvarchar(50) |  |  |  | The role of the user |
| IsWebmaster | bit |  |  |  | Check acount is admin |
| Avatar | nvarchar(50) | Yes |  |  | The link refers to an image, present for a user |
| Email | nvarchar(50) | Yes |  |  | The email of user |
| Phone | nvarchar(50) | Yes |  |  | Phone of user |
| Fullname | nvarchar(50) | Yes |  |  | The fullname of user |
| Unique: Username, Email, Avatar | | | | | |

#### Apply

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| RecruitmentId | int |  | PK |  | The index of this recruitment |
| JobRequestId | Int |  | PK |  | The index of this job request |

#### Comment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| CommentId | int |  | PK |  | The index of this comment |
| CustomerId | int |  | FK |  | The index of customer which contain this comment |
| JobRequestId | int |  | FK |  | The index of job request which is contain this comment |
| PostTime | datetime |  |  |  | The time when the comment posted |
| CommentContent | nvarchar(500) |  |  |  | The content of the comment |
| FK: CustomerId REF Customer (CustomerId), JobRequestId REF JobRequest(JobRequestId) | | | | | |

#### Customer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| AccountId | int |  | PK |  | The index of user |
| Address | nvarchar(100) |  |  |  | The address of customer |

#### JobRequest

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| JobRequestId | int |  | PK |  | The index of this job request |
| SkillRefId | int |  | FK |  | The SkillRef of this job request |
| MaidMediatorId | int | Yes | FK |  | The maid mediator of this job request |
| StaffId | int | Yes | FK |  | The staff of this job request |
| Status | nvarchar(50) |  |  |  | The status of this job request |
| PostTime | datetime |  |  |  | The time that this job request is posted |
| ExpiredTime | datetime |  |  |  | The time that this job request is expired |
| ApplyTimes | datetime |  |  |  | The time that this job request is applied |
| MaidId | int |  | FK |  | The maid of this job request |
| IsActive | bit |  |  |  | Check that this job request has payment |
| Title | nvarchar(100) |  |  |  | Name of jobrequest |
| Price | int |  |  |  | Price of jobrequest |
| IsReported | bool |  |  |  | Check that this job request has report |
| FK: SkillRefId REF SkillReference(SkillRefId)  MaidMediatorId REF MaidMediator(MaidMediatorId)  StaffId REF Staff(StaffId) | | | | | |

#### Maid

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| MaidId | int |  | PK |  | The index of maid |
| MaidName | nvarchar(50) |  |  |  | The name of maid |
| MaidMediatorId | int |  | FK |  | The maid mediator of maid |
| StaffId | Int |  | FK |  | The staff mediator of maid |
| PersonalImage | nvarchar(50) |  |  |  | The image of maid |
| Experience | float |  |  |  | The years experience of maid |
| Description | nvarchar(500) |  |  |  | The description of maid |
| Birthdate | date |  |  |  | Birthdate of maid |
| Phone | nvarchar(50) |  |  |  | The phone of maid |
| Gender | nvarchar(10) |  |  |  | The gender of maid |
| English | bit |  |  |  | Skill english of maid |
| Korean | bit |  |  |  | Skill korean of maid |
| Chinese | bit |  |  |  | Skill chinese of maid |
| Japanese | bit |  |  |  | Skill japanese of maid |
| Married | bit |  |  |  | Marital status of maid |
| Address | nvarchar(100) |  |  |  | Address of maid |
| NumOfReport | int |  |  |  | Number of reports |
| FK: MaidMediatorId REF MaidMediator(MaidMediatorId)  StaffId REF Staff(StaffId) | | | | | |

#### MaidMediator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| AccountId | int |  | PK |  | The index of this maid mediator |

#### Notifier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| Id | int |  | PK |  | The index of notification |
| AccId | int |  |  |  | The index of account which has notification |
| Date | datetime |  |  |  | Date of notification |
| Content | nvarchar(500) |  |  |  | Content of notification |
| Link | nvarchar(100) | Yes |  |  | Link notify |
| View | bit |  |  |  | Confirm that view yet |

#### Rating

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| RatingId | int |  | PK |  | The index of this rating |
| CustomerId | int |  | FK |  | The customer who rate |
| JobRequestId | int |  | FK |  | The job request which is rated |
| Rate | float |  |  |  | The value of this rate |
| FK: CustomerId REF Customer(CustomerId)  JobRequestId REF JobRequest(JobRequestId) | | | | | |

#### Register

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| RecruitmentId | int |  | FK |  | The recruitment which is job request register into |
| JobRequestId | int |  | PK |  | The job request which is register |
| RegisteredDate | date |  |  |  | The value of this rate |
| FK: RecruitmentId REF Recruitment(RecruitmentId) | | | | | |

#### Recruitment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| RecruitmentId | int |  | PK |  | The index of this recruitment |
| SkillRefId | int |  | FK |  | The index of skill which contains in this recruitment |
| CustomerId | int |  | FK |  | The customer who posts this recruitment |
| Status | nvarchar(50) |  |  |  | The status of the recruitment |
| PostTime | datetime |  |  |  | The time when the recruitment is posted |
| ExpiredTime | datetime |  |  |  | The time when the recruitment is expired |
| Price | int |  |  |  | Price of recruitment |
| Title | nvarchar(100) |  |  |  | Name of recruitment |
| IsActive | bit |  |  |  | Check that this recruitment has payment |
| NumOfReg | int |  |  |  | Number of register |
| FK: CustomerId REF Customer(CustomerId)  SkillRefId REF SkillReference(SkillRefId) | | | | | |

#### SkillCategory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| CategoryId | int |  | PK |  | The index of this category |
| CategoryName | nvarchar(50) |  |  |  | The name of category |

#### SkillInstance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| SkillId | int |  | PK |  | The index of this skill |
| CategoryId | int |  | FK |  | The category which contains this skill |
| SkillName | nvarchar(50) |  |  |  | The name of skill |
| SkillString | nvarchar(50) | Yes |  |  | The character value of skill name |
| SkillNormallized | int |  |  |  | The range value of skill |
| SkillStringValue | float |  |  |  | The number value of skill name |
| SkillNameVietnam | nvarchar(50) |  |  |  | The name of skill in Vietnamese |
| FK: CategoryId REF SkillCategory(CategoryId) | | | | | |

#### SkillReference

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| SkillRefId | int |  | PK |  | The index of this skill reference |
| Gender | int | Yes |  |  | The value gender of skill |
| Age | int | Yes |  |  | The value age of skill |
| LaguageEnglish | int | Yes |  |  | The value language english of skill |
| LanguageChina | int | Yes |  |  | The value language china of skill |
| LanguageJapanese | int | Yes |  |  | The value language japanese of skill |
| LanguageKorean | int | Yes |  |  | The value language korean of skill |
| Experience | int | Yes |  |  | The value experience of skill |
| Hometown | int | Yes |  |  | The value hometown of skill |
| Address | int | Yes |  |  | The value address of skill |
| Health | int | Yes |  |  | The value health of skill |
| Stay | int | Yes |  |  | The value stay of skill |
| Salary | int | Yes |  |  | The value salary of skill |
| Work | int | Yes |  |  | The value work of skill |
| SickCare | int | Yes |  |  | The value sickcare of skill |
| OldCare | int | Yes |  |  | The value oldcare of skill |
| BabySister | int | Yes |  |  | The value babysister of skill |
| DisabilityCare | int | Yes |  |  | The value disabilitycare of skill |
| BonsaiCare | int | Yes |  |  | The value bonsaicare of skill |
| Cooking | int | Yes |  |  | The value cooking of skill |
| Washing | int | Yes |  |  | The value washing of skill |
| Cleanhouse | int | Yes |  |  | The value cleanhouse of skill |
| Type | int |  |  |  | Skill of job request or skill of recruitment |
| Group | int | Yes |  |  | Divide group in algorithm |

#### Staff

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field name** | **Type** | **Null** | **P/F key** | **Default** | **Description** |
| AccountId | int |  | PK |  | The index of this staff |

## Algorithms

### K-Means algorithms

#### Definition

* k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.
* The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms. Additionally, they both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes.
* This is automatically function, help system classify each job request and recruitment from maid and customer. Function give all request become n-group. From there, system can be suggest for user about, Job requests same RecruitmentApply.

#### Problem

* Customer have Recruitment and MaidMediator or Staff have Job request, they need to find a Job request or Recruitment. System will suggest for them some job request or recruitment.
* With a job request or recruitment, MS-Website database save many attribute (20 attribute).
* System group all job request to n-group. Each group have the same element of a few attributes.

#### Solution

MS-Website use K-Means algorithm. Each job request or recruitment consider as a point in 20 dimensions space.

* First, normalize value. From qualitative values like age, address... to quantitative values. Intervals dividing is implemented by choosing an attribute which has the most number of values, to be particular it's Address attribute with 20 values in database. Therefore we choose the largest value 100 and divide into intervals: 5, 10, 15...
* Second, choose a number of cluster (K). In MS-Website, system suggest each group at job request and recruitment should have 10 element. Therefore, if job request have 100 rows we should choose k equal 10. However, admin can choose a another number K.
* Third, run K-Means algorithm. New group based on similar of value from column Normalize of data. A file (Mean.txt) contain k mean, with k is number of group. This file will be save on server of website.

#### Complexity

the complexity of this algorithm is O(n \* k \* I)

* n : number of points (JobRequest and Recruitment)
* k : number of cluster
* I : number of iteration

#### Flowchart

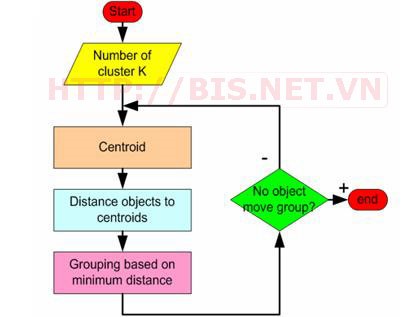


Figure : K-Means algorithm flow chart