|  |
| --- |
| **FPT-aptech computer education** |
| eProject Document |
| [RECRUITMENT PROCESS SYSTEM] |
|  |
| |  |  | | --- | --- | | **FAT 1 – C1007L – SEM3 – GROUP 6** | | | **Group Member** | Vũ Hoàng Chiến | | **Instructor** | <Faculty name> | | **Batch** | <BatchName> | | **Semester** | SEM 3 | |
|  |

|  |
| --- |
| - Hanoi .10/ 06/2013 - |

# Introduction

*The thirst for learning, upgrading technical skills and applying the concepts in real*

*life environment at a fast pace is what the industry demands from IT professionals*

*today. However busy work schedules, far-flung locations, and unavailability of*

*convenient time-slots pose as major barriers when it comes to applying the concepts*

*into realism. And hence the need to look out for alternative means of*

*implementation in the form of laddered approach.*

*The above truly pose as constraints especially for our students too! With their busy*

*schedules, it is indeed difficult for our students to keep up with the genuine and*

*constant need for integrated application which can be seen live especially so in the*

*field of IT education where technology can change on the spur of a moment. Well,*

*technology does come to our rescue at such times!!*

*Keeping the above in mind and in tune with our constant endeavour to use*

*Technology in our training model, we at Aptech have thought of revolutionizing the*

*way our students learn and implement the concepts using tools themselves by*

*providing a live and synchronous eProject learning environment!*

# Problem Definition

## Problem Abstraction

This project is aimed at developing a web-based and central Recruitment Process System for the HR Group for a company (Mango bussiness). Some features of this system will be creating vacancies, storing Applicants data, Interview process initiation, Scheduling Interviews, Storing Interview results for the applicant and finally Hiring of the applicant. Reports may be required to be generated for the use of HR group.

## The Current System

*Life today to solve the problem of information job and virtual job seeker. Recruitment system We create thorough evaluation of the information security job postings and candidate personal information need not "hunt" candidates introduced earlier for other employers without permission...*

## The Proposed System

• In recent years, the issue of data security and information management effectively before 'whirlwind' internet and the risk of leakage of confidential information outside society is very focused company. The recruitment of personnel responsible for the management of information and now is the time when the most powerful field development.

• A successful recruitment system must meet: Privacy and reduce recruitment costs.

• Development of a web-based Recruitment Process System for the HR group for a company

## Boundaries of the System

To meet the needs of customers, the group has set requirements for the is to build a system so complete. Currently the program offers features to users as follows:

* **Deploy:**
  + Attach database
  + Deploy system
* **Login:**
  + User Name
  + Password
* **HR group:**
  + View, Create, edit, delete vacancy
  + View, Create, edit, delete applicant
  + Search
  + Attach Applicant to Vacancy
  + Schedule or remove interview
* **Interviewer:**
  + View scheduled, applicant, vacancy
  + Change scheduled
  + Update result for interview

## Development Environment

1. Minimum requirements

|  |
| --- |
| *Minimum requirements* |
| PC Pentium III |
| 128 MB RAM |
| 1 GB Hard Drive |

1. Recommended requirements

|  |
| --- |
| Recommended requirements |
| PC Pentium III 1800 MHz |
| 2G RAM |
| 2 GB hard drive |

1. *Software*
   * *ASP.NET 4.0*
   * *Visual studio 2012*
   * *IIS 7*
   * *MS SQL Server 2012*
   * *Window 7 or window 8*
   * *Telerik Q1 2013*

# Requirements and Business Flow

## Customer Requirement Specification

## 

**Applicant:**

* View home page
* Register & Login to website
* Edit individual information
* Look for and view a recruitment vacancy
* Apply for a vacancy

**Admin:**

* View and edit individual information
* Add HR and interview
* Reset password of HR and interview
* Search HR, interview
* View recruited applicant for a vacancy

**HR:**

* View applicant for a position
* Change applicant status
* Create/ delete interview schedule
* Create/ edit recruitment vacancy
* Look for applicant
* Look for recruitment vacancy

**Interviewer:**

* View applicant for a vacancy
* View/edit individual information
* View recruitment vacancy
* View applicant information
* Change the status of interview schedule
* Create interview schedule and result
* Make proposal of changing interview schedule
* Create/ edit recruitment vacancy
* Look for applicant, recruitment vacancy

## Activity Diagram

## Use Case Diagram

*<Put here the overall use case diagram. If the system can be partitioned into several sub-systems, you can use multiple diagrams to show the overall functionalities of the system>*

## Use Case Specification

*<Write down all non-trivial use cases. This should reflect what you get when your team does the system analysis. Use the template to write the detailed specification for use cases>*

*<Use case temlpate:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | | |
| **Use-case No.** | <UC001> | **Use-case Version** | | <1.0> |
| **Use-case Name** | <Name> | | | |
| **Author** | <Members> | | | |
| **Date** | Dd/mm/yyyy | **Priority** | <High\Normal\Low> | |
| **Actor:**  <Lit all actors>  **Summary:**  *<Briefly describe the use case>*  **Goal:**  *<Briefly describe the goal of use case>*  **Triggers**  *<What leads this use case?>*  **Preconditions:**  *<List the required pre-conditions for this use case>*  **Post Conditions:**  *<List the required post-condition for this use case>*  **Main Success Scenario:**  *<List the main steps for this use case to reach the goal successfully >*    **Alternative Scenario:**  *<List the other steps for this use case to reach the goal in some alternatives condition >*  **Exceptions:**  *<list the exceptions of this use case>*  **Relationships:**  *<List the relationships that use case relates to>*  **Business Rules:**  *<Any concern about the business>* | | | | |

## Other Concerns<Optional>

*<You can list here all other concerns about the business or the requirements if needed>*

# Design

## System Architecture

* Admin table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | Admin\_Id | Int | PK, auto increment, Not null |  |
| 2 | Admin\_FullName | Nvarchar(50) |  |  |
| 3 | Admin\_Account | Nvarchar(50) |  |  |
| 4 | Admin\_Password | Nvarchar(50) |  |  |
| 5 | Admin\_Role | Nvarchar(10) |  |  |
| 6 | Admin\_Status | bit |  |  |

* Applicant table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | Applicant\_Id | Int | PK, auto increment, Not null |  |
| 2 | Vacancy\_Id | Int | FK with Vacancy table |  |
| 3 | Applicant\_FullName | Nvarchar(50) |  |  |
| 4 | Applicant\_Email | Nvarchar(50) |  |  |
| 5 | Skill | Ntext |  |  |
| 6 | PositionApply | Nvarchar(50) |  |  |
| 7 | Cetificate | Ntext |  |  |
| 8 | SuggestSalary | Ntext |  |  |
| 9 | Applicant\_Address | Nvarchar(50) |  |  |
| 10 | Applicant\_IP | Nvarchar(50) |  |  |
| 11 | Applicant\_Description | Ntext |  |  |
| 12 | Applicant\_DateRegister | Nvarchar(50) |  |  |
| 13 | Applicant\_Admin\_Accept | Nchar(10) |  |  |
| 14 | Applicant\_Result | Nchar(10) |  |  |
| 15 | Applicant\_Client\_Confirm | Nchar(10) |  |  |

* Interview table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | Interviewer\_Id | Int | PK, auto increment, Not null |  |
| 2 | Schedule\_Id | Int | FK with Schedule table |  |
| 3 | Interviewer\_Result | Nvarchar(50) |  |  |
| 4 | Interviewer\_Note | Ntext |  |  |

* Schedule table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | Schedule\_Id | Int | PK, auto increment, Not null |  |
| 2 | DateTime\_Create | Nvarchar(50) |  |  |
| 3 | Schedule\_Date | Nvarchar(50) |  |  |

* Vacancys table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| 1 | Vacancy\_Id | Int | PK, auto increment, Not null |  |
| 2 | Schedule\_Id | Int | FK with schedule table |  |
| 3 | Vacancys\_WorkAddress | Nvarchar(200) |  |  |
| 4 | Vacancy\_TypeTime | Nchar(10) |  |  |
| 5 | Vacancy\_Salary | Nvachar(50) |  |  |
| 6 | Vacancy\_Position | Nvarchar(50) |  |  |
| 7 | Vacancy\_Numberpeople | Int |  |  |
| 8 | Vacancy\_Skill | Nvarchar(200) |  |  |
| 9 | Vacancys\_Exp | Nvarchar(50) |  |  |
| 10 | Vacancy\_Gender | Nvarchar(50) |  |  |
| 11 | Vacancy\_Age | Nvarchar(50) |  |  |
| 12 | Vacancys\_Description | Ntext |  |  |
| 13 | Vacancy\_DateStart | Nvarchar(50) |  |  |
| 14 | Vacancy\_DateEnd | Nvarchar(50) |  |  |
| 15 | Vacancy\_DateUp | Nvarchar(50) |  |  |
| 16 | Vacancy\_DateInterViewer | Nvarchar(50) |  |  |
| 17 | Vacancy\_TimeInterViewer | Nvarchar(50) |  |  |
| 18 | Vacancy\_Status | Nvarchar(50) |  |  |
| 19 | Admin\_Id | Int | FK with Admin table |  |

## Class Diagram

*<Provide class diagrams for the project>*

## Class Diagram Explanation

*<Provide brief explanation about the class diagram above. You do not need to explain “obvious” parts of your class diagram. For example, I know what a “Login” class is. Don’t say “The login class was created to store login information.”>*

## Sequence Diagram (Optional)

*<for important and complex interactions, protocols or algorithms, sequence diagrams should be drawn for clearing the details and supporting the system implementation. This section is optional>*

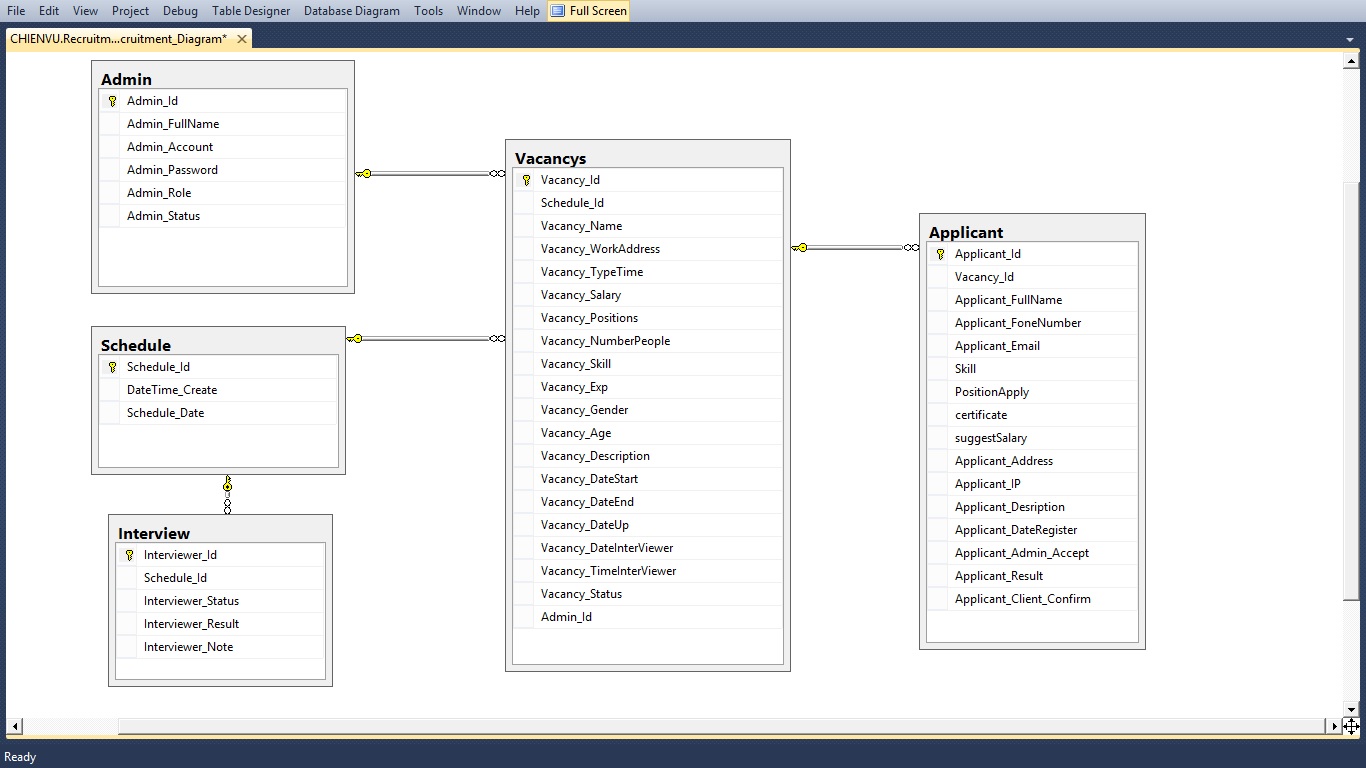
## Collaboration Diagram (Optional)

*<for important and complex interactions, collaboration diagrams should be drawn for clearing the details and supporting the system implementation. This section is optional>*

## State Diagram (Optional)

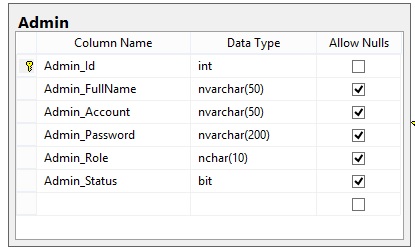
*<put all state diagrams here>*

## Entity Relationship Diagram

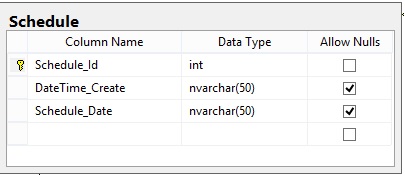


## Database Design

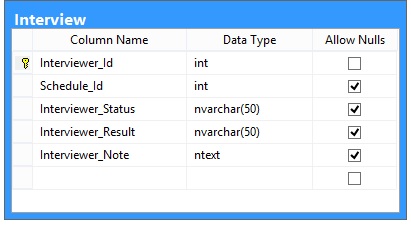
* Admin table



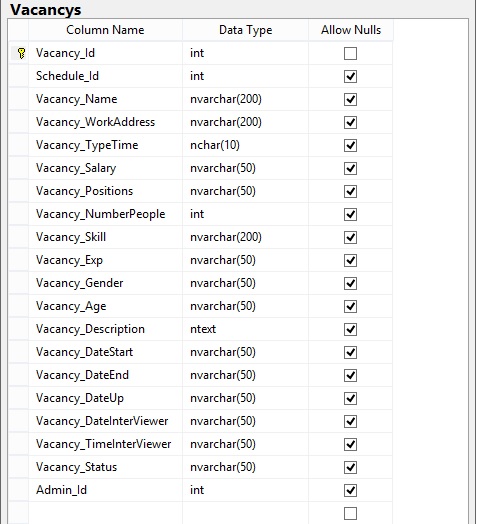
* Schedule table



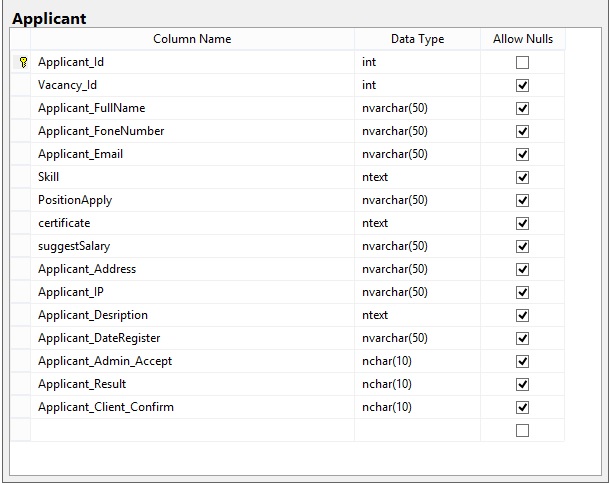
* Interview table



* Vacancys table



* Applicant table



## Algorithms (optional)

*<Provide the detailed description about algorithms used in the system. You can use Flow Chart or Activity Diagram to represent algorithms. Focus on the important and complex algorithms>*

## Others (optional)

*<Any design concerns or diagrams can be put here>*

# System Prototype

*<Put the system prototype or mock UI here. Focus on* ***important forms*** *and the* ***screen flows*** *between forms. If you use RAD and .NET, prototyping really help you to reach your development goals quickly>*

# Management and Project Planning

## Management Approach

Our team is half self-managed and half managed by one leader. We plan to become Scrum team. Therefore, we still need one leader to observer the overall project process. The assign task process is self-managed. Each member in Scrum team takes the appropriate task that suitable for him. We plan to do the meeting every day. 4 hours per day. 15 minutes of summarize what each member did, didn’t last meeting and what are they going to do.

## Project Plan

*<The detailed project plan is put here. You can use WBS Excel sheet, Sprint Backlog (see Sprint Backlog\_Template.xls), Task sheet, Gantt chart, etc. to present your team’s plan. You can capture the Gantt chart in PMS if you use it to plan your project>*

## Task Sheet

*<Write down the tasks in Task Sheet-compatible format, this Task Sheet works as the activity report of the projec or the plan of the project (not recommended); see eProject Guide for detailed Task Sheet>*

## Meeting Minutes (Optional)

*<Put all minutes of your team meetings here>*

# Checklists

## Check List of Validation

*< Put the checklist here; describe how it is used and the resulted checklist>*

## Submission Checklist

*< Put the checklist here; describe how it is used and the resulted checklist>*

# Screenshots

*<Capture some intuitive and main screens of the software and put them here>*

# Coding Convention

*<Provide the coding convention for your team. If you simply want to use the existing code standard(s) such as ‘Java Code Convention’, you can refer to it\them by name or URL>*

# Other Concerns<Optional>

*<If you have any other information you want to add to this document, place it here. This could include thoughts on the eProject, improvements, etc.>*

# Appendix

## Glossary [Optional]

*<Place all definitions or abbreviation used in this document >*

## References [Optional]

*<Place all referenced materials used in this document >*

## Others<Optional>