|  |
| --- |
| FPT-aptech computer education |
| eProject Document |
| Recrutiment Process System |
|  |
| |  |  | | --- | --- | |  | | | **Group Member** | <Hoang Do Phu><C00473><1689>  <Tung Trinh Minh><C00194><6196>  <Luong Bach Van><A03593><6109> | | **Instructor** | Vinh Duong Le | | **Batch** | C0812L | | **Semester** | 4 | |
|  |

|  |
| --- |
| - Hanoi, 4/2012 - |

*<The Table of Contents goes here>*

# Introduction

*<Introduction to this document, how it is organize, what it contains>*

# 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. 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

There are HR group who will create vacancies, create applicants (persons applying for a vacancy), initiate interviews and close vacancy. There are interviewers who will be intimated about the interview schedule and finally enters the result.

## The Proposed System

The requirements are as follows:

1. **Generic:**
   1. Login to the system through the first page of the application – *the welcome message should display the Name and Employee number of the person.*
   2. Change the password after login to the application
   3. See his/her details and change it.
   4. Help from the system
2. **HR group:**
   1. Should be able to create a new vacancy.
   2. Should be able to change any of the editable details for the vacancy.
   3. Should be able to create a new applicant.
   4. Should be able to change any of the editable details for the applicant.
   5. Should be able to search on Applicant Number and Vacancy Number.
   6. Attach an applicant to a vacancy - *The relationship between applicant and vacancy should be many:many.*
   7. Should be able to schedule the interview and enter details of the interviewer and date/time.
   8. Should be able to postpone or remove the interview.

.

1. **Interviewer:**
   1. Should be able to view all interviews scheduled to be taken.
   2. Should be able to view the details of the applicants details
   3. Should be able to view the details of the vacancy.
   4. Should be able to search on Interview Date, Applicant Number and Vacancy Number.
   5. Should have the access to change the Interview details – Date/Time, Status (Selected/Rejected).
2. **Other details:**
   1. The vacancy should have the following details:
      1. System should generate a Vacancy number sequentially. *This should be unique and cannot be changed later on.*
      2. Date of the creation of the vacancy should default as today – *This cannot be changed later on.*
      3. Status of the Vacancy – Open/Close/Suspended. *This should default as Open but can be changed later by HR.*
      4. Title of the Vacancy (like Java Developer etc.)
      5. Number of job opening under that vacancy (like 5 numbers of Java Developers required). *Once all these job openings are filled i.e. 5 people are hired, the vacancy should close itself.*
      6. Department in the company – *this should come from a drop-down box which have the Department list.*
      7. Date by which the vacancy should be fulfilled/gets closed – *this can be left blank otherwise the vacancy will close on that day by default.*
      8. List of all the applicants already ‘Selected’ for a job openings in the vacancy.
   2. Applicant details should have the following:
      1. System should generate an Applicant number sequentially. *This should be unique and cannot be changed later on.*
      2. Date of the creation of the applicant should default as today – *This cannot be changed later on.*
      3. Status of the Applicant – Not in Process/In Process/Hired/Banned.
      4. The Applicant Status should default as ‘Not in Process’ on creation of the applicant but as soon as a single vacancy is attached, this should default as ‘In Process’. No more vacancy can be attached to the applicant, if the status is either ‘Hired’ or ‘Banned’ - *The status can be changed manually by HR Group.*
   3. Applicant-Vacancy data should have the following:
      1. This should display the Applicant Number and Name.
      2. This should display the Vacancy Number and Title
      3. Display the date on this the Applicant is attached to the vacancy
      4. Status of the Applicant/Vacancy – Interview Scheduled/Selected/Rejected/Not Required.
      5. Checkbox stating – ‘Schedule Interview’. This should open the page with following details.
         1. Display the Employee Number of the Interviewer – *The interviewer should be from the same department.*
         2. Display the Name of the Interviewer – this should default if the number is selected.
         3. Display the date of the scheduled interview – *Date should only be in future.*
         4. Display the Start and End time of the scheduled interview – *Date/Time of the interview for the same Interviewer should not conflict with any other interview previously scheduled for the interviewer/applicant.*
   4. As soon as all the number of job openings is filled, the vacancy status should change to ‘Close’ by default.
   5. HR group should not be able to attach an applicant to vacancy if the status of the Vacancy is ‘Close/Suspended’.
   6. Once closed, a vacancy can not be reopened or suspended in any case.
   7. If Open, the status of the vacancy can be changed to ‘Close’ or ‘Suspended’ by the HR.
   8. If ‘Suspended’, the status of the vacancy can be changed to ‘Close’ or ‘Open’ by the HR.
   9. Once the status of the Applicant-Vacancy becomes ‘Selected’, the status of the Applicant should change by default to ‘Hired’.
   10. The required security needs to be implemented.
   11. HR group is the Employees from ‘HRD’ department.

## Boundaries of the System

The system will be used by employees of the company:

* Administrator
* HR Group
* Interviewer

## Development Environment

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | \* Ram 512 MB or larger  \* CPU 2.0 Ghz or larger  \* HDD 80 GB or larger  \* Connect to internet | \* Ram 128 Mb or larger  \* CPU 266 MHZ or larger  \* Connect to internet |
| Software | \* Windows Server 2000 or later  \* SQL Server 2005 or later  \* IIS 6 or later | \* Internet Explorer  \* Firefox  \* Chrome  \* Safari |

*Developer tools*

* Microsoft Windows 2000 SP4 or higher
* Net Bean 6.9 or higher
* Microsoft SQL Server 2005 Standard Edition or higher
* j2sdk1.4.1\_02 (or later)
* Microsoft Office XP
* Microsoft Internet Explorer 7.0
* Adobe Dreamweaver CS5

# Requirements and Business Flow

*<In this section you will provide the requirements you developed against. This section should be useful, so only provide enough information (requirements, diagrams) to help with the implementation of the software. Focus on the “non-obvious” aspects. If you are dealing with a business process, activity diagrams are good. Making a set of requirements that is compact, simple and covers all the key aspects of the software can. Don’t do the work if you don’t need it!>*

## Customer Requirement Specification

*<Summarize the customer requirement into the compact form>*

## Activity Diagram

*<Business processes should be modeled carefully. Use activity diagrams to show important businesses. Focus on non-trivial ones>*

## 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

*[This section shows design of the system. This could be a part of the Developers Manual]*

## System Architecture

*<Explain and present the architecture of the system using texts or diagams>*

## 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>*

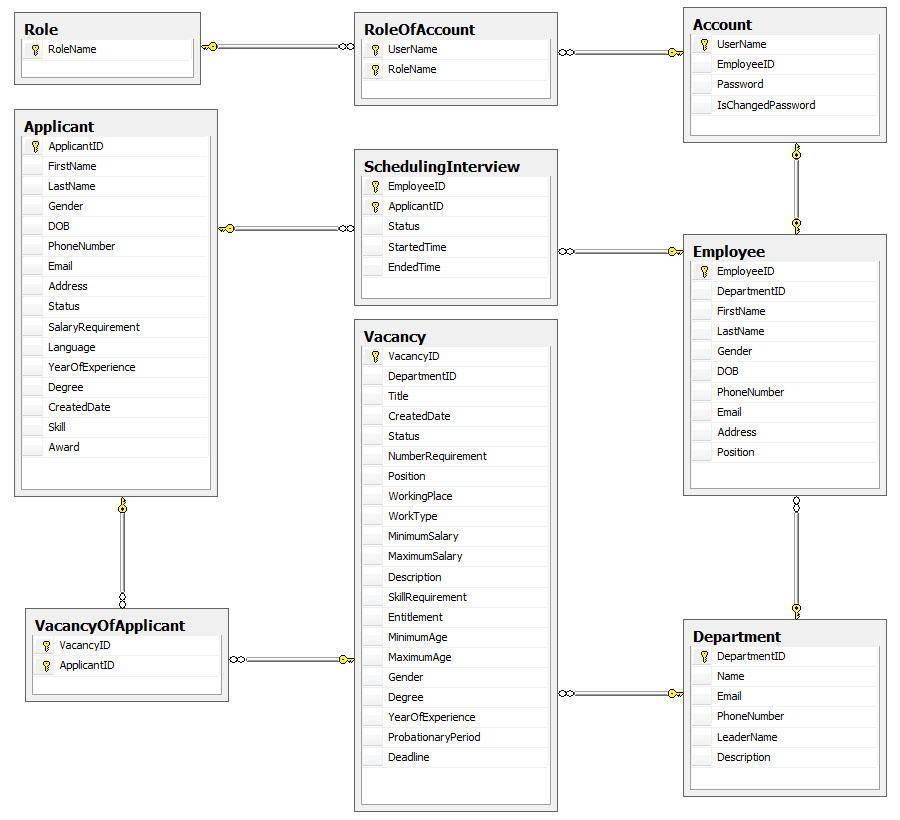
## Collaboration Diagram (Optional)

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

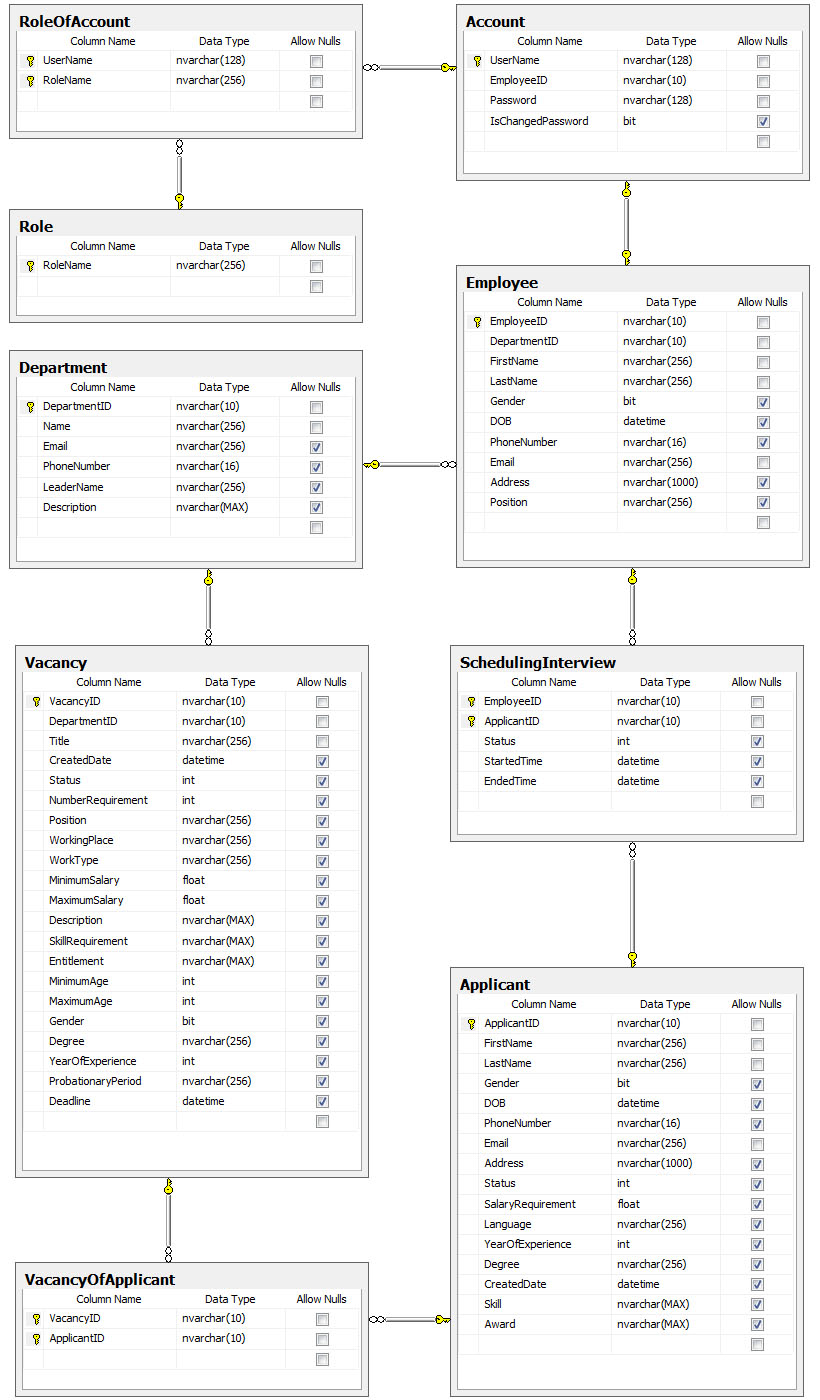
## State Diagram (Optional)

*<put all state diagrams here>*

## Entity Relationship Diagram

**

## Database Design



## 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>*

# Management and Project Planning

## Management Approach

*< Briefly describe the management approach that your team selects. Is your team self-managed or managed by one leader? How do you assign tasks to team members? How often do you meet? What do you do during meeting? Etc.>*

## Project Plan

*<The detailed project plan is put here. You can use WBS Excel sheet, Sprint Backlog, 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 maner; 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>