**COMP 3059 – Capstone Project I**

**Software Requirements Analysis and Design Assignment**

Table of Contents

[1.0 Introduction 3](#_Toc529219950)

[1.1 Purpose 3](#_Toc529219951)

[1.2 Scope 3](#_Toc529219952)

[2.0 System Overview 3](#_Toc529219953)

[2.1 Project Perspective 3](#_Toc529219954)

[2.2 System Context 4](#_Toc529219955)

[2.3 General Constraints 5](#_Toc529219956)

[2.4 Assumptions and Dependencies 5](#_Toc529219957)

[3.0 Functional Requirements 5](#_Toc529219958)

[3.1 Functional Requirement or Feature #1> 5](#_Toc529219959)

[3.2 Use Cases Diagrams 6](#_Toc529219960)

[3.2.1 Use Case Diagram - Client 6](#_Toc529219961)

[3.2.2 Use Case Diagram - Departments 7](#_Toc529219962)

[3.3 Data Modelling and Analysis 8](#_Toc529219963)

[3.3.1 Normalized Data Model Diagram 8](#_Toc529219964)

[3.3.2 Activity Diagrams 9](#_Toc529219965)

[3.3.3 Sequence Diagrams 9](#_Toc529219966)

[3.3.4 UML Class Diagram 10](#_Toc529219967)

[3.3.4.1 Method Contract: 10](#_Toc529219968)

[3.3.4.2 Method Specification 11](#_Toc529219969)

[3.3.5 Process Modelling 13](#_Toc529219970)

[3.3.5.1 Data Flow Diagram 13](#_Toc529219971)

[4.0 Non-Functional Requirements 14](#_Toc529219972)

[5.0 Logical Database Requirements 14](#_Toc529219973)

[5.1 Data Format 15](#_Toc529219974)

[5.2 Structure 15](#_Toc529219975)

[5.3 Selections 16](#_Toc529219976)

[5.4 Data Integrity 16](#_Toc529219977)

[6.0 Other Requirements 17](#_Toc529219978)

[7.0 Approval 18](#_Toc529219979)

# Introduction

## Purpose

This requirement specification document describes the functions and requirements for the Recruitment Boutique System It explains the purpose and features of the system.

## Scope

Recruitment Boutique is a work flow for JC Consulting designed to control the recruitment process. The system will be designed to optimize the flow between the JC Consulting areas, to measure SLA, and improve communication between areas, including feedback emails and providing a uniform process. The system will contain a database with the candidates resume and client’s opportunities.

The system will cover:

* flow control, job opportunity input for the clients, SLA control, status report, status changes, easy search, easy login, email notification, status alert.

# System Overview

## 

## Project Perspective

The Recruitment Boutique system will control the work flow between JC Consulting areas.

**Customer:**

**Customer description:**

It includes customer, code, name, address, phone number and contact. This information will be used to give feed back about the requested position and allocated candidates.

**Customer job position:**

It includes the description and requirements of job opportunity.

**Intake Department:**

It includes department code, responsible, activity status, activity start date, activity end date, activity responsible.

**Evaluate Department:**

It includes department code, responsible, activity status, activity start date, activity end date, activity responsible.

**Allocate Department:**

It includes department code, responsible, activity status, activity start date, activity end date, activity responsible, candidates allocated.

**Deployment Department:**

It includes department code, responsible, activity status, activity start date, activity end date, activity responsible, candidate deployed.

**Support Department:**

It includes department code, responsible, activity status, activity start date, activity end date, activity responsible, QA forms, QA results (graphics)

## System Context

## 



Figure - System Context

## General Constraints

* The system must be delivered by April 2019
* He system must be user-friendly
* Implement the database using a centralized database management system to the groups
* Each area will be able to login at a specific part of the process

## Assumptions and Dependencies

* The Scope may change as new information and issues are revealed
* It is assumed that the user is familiar with an internet browser and familiar with handling the keyboard and mouse.
* Failure to identify changes to draft deliverables within the time specified in the project timeline will result in project delays
* Since the application is a web-based application there is a need for the internet browser. It will be assumed that the users will possess decent internet connectivity.

# Functional Requirements

## Functional Requirement or Feature #1>

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement Statement | Must / Want | Comments |
| 1 | New users shall be able to register to the system clicking on Register Link | Must |  |
| 2 | Register users shall be able to Login to system by providing email & password | Must |  |
| 2.1 | Upon successful login, user shall be redirected to client page in the system | Want |  |
| 2.2 | If email or password are not corrected, then Login Error message shall be prompted | Must |  |
| 3 | User shall be able to recover their password by clicking in the “Forgot Password” link | Must |  |
| 4 | Users should be able to search for their job requests and create a report with the status of all open positions | Want |  |
| 5 | The system shall have a home pate with the purpose of the organization | Mus |  |
| 6 | The system shall list the members of JC Consulting | Want |  |
| 7 | A “Contact US” page shall provide a form to send an email | Must |  |
| 9 | Client shall be able to request to cancel a job position | Must |  |
| 10 | Client shall be able to alter a job position | Must |  |
| 11 | Areas shall be able to change job position status | Must |  |
| 12 | Each area shall be able to input information in a specific field | Must |  |
| 13 | System shall be able to record all inputs (Areas) | Must |  |
| 14 | System shall be able to send an email to the next Area | Must |  |
| 15 | **Admin Powers**: Read and write authorization to individual resources and sets of resources must be separate | Must |  |
| 16 | A User should not be allowed to overwrite another user’s changes | Must |  |
|  |  |  |  |

Table - Functional Requirements

## Use Cases Diagrams

## Use Case Diagram - Client

Assumptions:

* + - Client can create an account
    - Only User can sign
    - Only registered client can solicitate job position

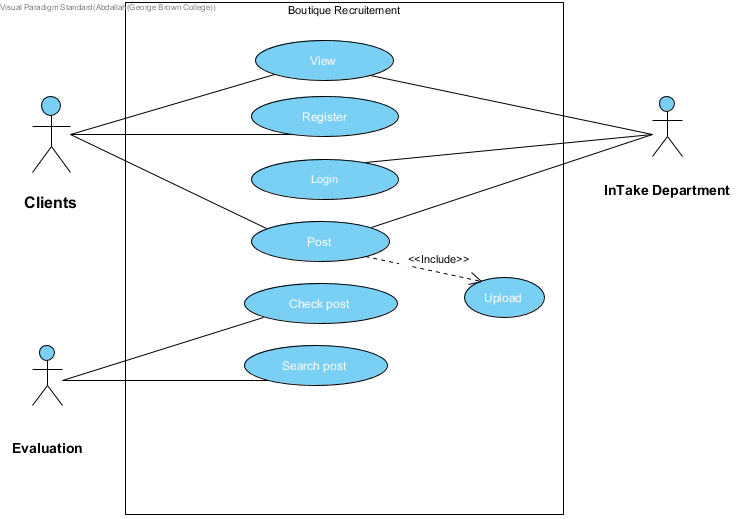


Figure - Use Case Diagram Clients

## Use Case Diagram - Departments

Assumptions:

* + - Only User can sign
    - All Department can search for a post
    - All Department will change post status
    - Intake Department can cancel a post

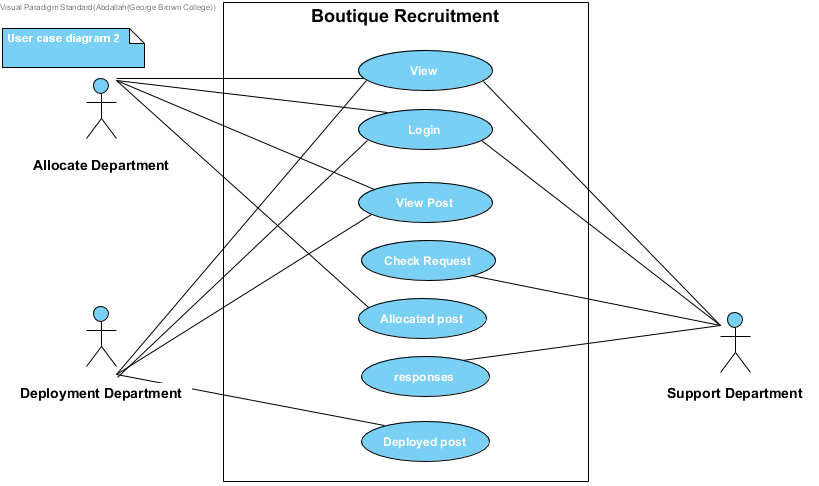


Figure - Use Case Diagram Departments

## Data Modelling and Analysis

## Normalized Data Model Diagram

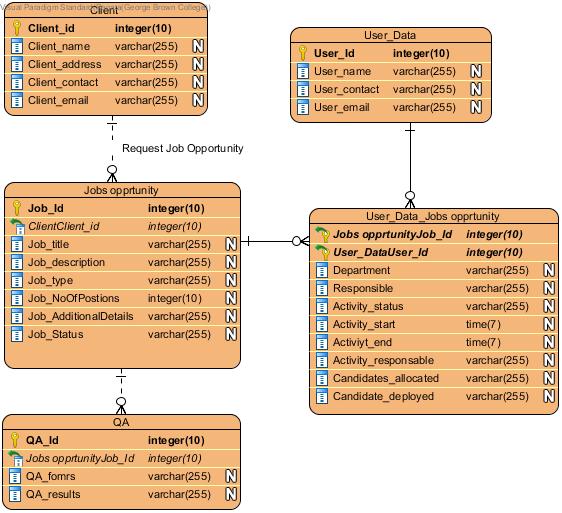


Figure - Normalized Data Model Diagram

## Activity Diagrams

Figure - Activity Diagram

## Sequence Diagrams

Figure - Sequence Diagram

## UML Class Diagram

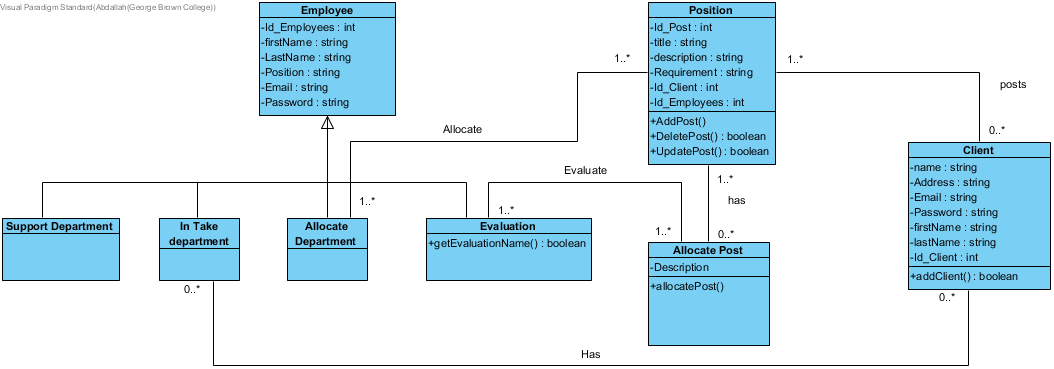


Figure - UML Class Diagram

## Method Contract:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Contract: AddPost() | | | | | |
| Method Name: | AddPost() | Class Name: AddPost | Course | ID: | 1 |
| Clients: Clients | | | | | |
| Associated Use Cases: Client | | | | | |
| Description of Responsibilities: Add the position | | | | | |
| Arguments Received: Client\_Id, Job\_Id, Job\_title, Job\_description, Job\_type, job\_NoOfPositions,Job\_additionalDetails | | | | | |
| Type of Value Returned: Status of the position | | | | | |
| Pre-Conditions: Fill the form with required field  Register before | | | | | |
| Post-Conditions: Approved Post | | | | | |

Table - AddPost() Method Contract

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Contract: AddClients() | | | | | |
| Method Name: | AddClients() | Class Name: AddClients | Course | ID: | 2 |
| Clients (consumers): Clients | | | | | |
| Associated Use Cases: Clients | | | | | |
| Description of Responsibilities: Clients register in the system | | | | | |
| Arguments Received: Client\_Id, Client\_name, Client\_address, Client\_contact, Client\_email | | | | | |
| Type of Value Returned: None | | | | | |
| Pre-Conditions: Fill the form with required field | | | | | |
| Post-Conditions: None | | | | | |

Table - AddClients() Method Contract

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method Contract: JobsFlow() | | | | | |
| Method Name: | Jobsflow() | ClassName: flow | Course | ID: | 3 |
| Clients (consumers): Departments | | | | | |
| Associated Use Cases: Departments | | | | | |
| Description of Responsibilities: Departments allowed to view and change jobs | | | | | |
| Arguments Received: Client\_Id, Client\_name, Job\_ID, Job\_description | | | | | |
| Type of Value Returned: None | | | | | |
| Pre-Conditions has a job posted | | | | | |
| Post-Conditions: None | | | | | |

Table - JobsFlow() Method Contract

## Method Specification

|  |  |  |
| --- | --- | --- |
| Method Specification: AddClient() | | |
| Method Name:  AddClient() | Class Name:  InsertClient | ID: 1 |
| Contract ID: 1 | Programmer: Renata Moura, Abdallahman Habyarimana, Anushaka | Data Due:27/04/2019 |
| Programing Language:HTML, CSS, JAVASCRIPT, SQL, MEAN STACK | | |
| Triggers/Events: New client registration | | |
| Arguments Received:  Data Type: | Notes: | |
| Client\_Id - int | Primary key | |
| Client\_name - String |  | |
| Client\_address - String |  | |
| Client\_contact – String |  | |
| Client\_email - String |  | |
| Client\_status - Boolean | Active or not | |
| Messages Sent & Arguments Passed:  ClassName, MethodName: | Data Type: | Notes |
| Register.Client | Boolean | Add new client at database |
| Arguments Returned:  Data Type: | Notes: | |
| Client\_status – Boolean | Return if the client is successfully registered | |
| Algorithm Specification: | | |
| Execute register page form  If client input required information  Submit form  Else  Show the needed corrections  Return status object | | |

Table - AddClient() Method Specification

|  |  |  |
| --- | --- | --- |
| Method Specification: AddPost() | | |
| Method Name:  AddPost() | Class Name:  Position | ID: 2 |
| Contract ID: 1 | Programmer: Renata Moura, Abdallahman Habyarimana, Anushaka | Data Due:27/04/2019 |
| Programing Language:HTML, CSS, JAVASCRIPT, SQL, MEAN STACK | | |
| Triggers/Events: Client post a position | | |
| Arguments Received:  Data Type: | Notes: | |
| Job\_Id - int | Primary Key | |
| Job\_title - String | Foreign Key | |
| Client\_Id: int |  | |
| Job\_description – String |  | |
| Job\_Type - String |  | |
| Job\_NoOfPositions - int |  | |
| Job\_AdditionlaDetais - String |  | |
| Job\_status:boolean |  | |
| Job\_status\_post:boolean |  | |
| Messages Sent & Arguments Passed:  ClassName, MethodName: | Data Type: | Notes |
| Request.Job | Boolean | Request Evaluate Department to review and write a job position |
| Post.Job | String | Post Job request on website |
| Arguments Returned:  Data Type: | Notes: | |
| Job\_status – Boolean | Return if the job position is active or not | |
| Job\_status\_post -Boolean | Returns if the job was post successfully | |
| Algorithm Specification: | | |
| Execute register page form  If client input required information  Submit form  Else  Show the needed corrections  Return status object | | |

Table - AddPost() Method Specification

## Process Modelling

## Data Flow Diagram

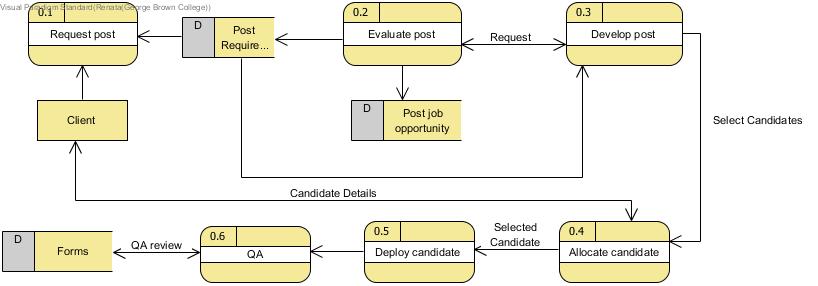


Figure - Data flow Diagram

# Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| ID | Requirement Type | Requirement Statement |
| 1 | Performance requirements | Normalization – The objective of normalization is to reduce redundancy, which means that the information is to be stored only once. If database is not properly designed, it can give rise to modification anomalies. |
| 2 | Performance requirements | Real time access to date- The system must provide real time access to database |
| 3 | Security | Access – Any user who uses the system shall have a login ID and Password |
| 4 | Security | Modification – Any modification (insert, delete, upgrade) for the Database shall be done by a logged user |
| 5 | Security | Administrators’ Rights - Administrators shall be able to view and modify all information |
| 6 | Security | Authentication – Correct identification of parties attempting to access the system, and protection of the system form unauthorised parties |
| 7 | Security | Data Confidentiality – Data needs to be protected and kept confidential |
| 8 | Qualities Attributes | Availability – The system shall be available all the time |
| 9 | External requirements | The system shall not disclosure any personal information about clients and candidates. |
| 10 | System goal requirements | The system should be easy to use by the users and should be organized to minimize user errors. |

Table - Non-functional Requirements

# Logical Database Requirements

The following is a logical classification of various data entities as well as their attributes

|  |  |  |
| --- | --- | --- |
| Data | Attributes | Use |
| Client Data | * Client\_Id * Client\_name * Client\_address * Client\_contact * Client\_email * Client\_status | This data is used to identify the clients who will post jobs to the system. |
| Jobs Data | * Client\_Id * Job\_Id * Job\_title * Job\_description * Job\_type * No\_of\_positions * Additional\_details | This data will include a full detail on each job posted by the clients. |
| User Data | * Type\_of\_user * Dept\_Id * User\_Id * User\_name * User\_contact * User\_email | This data identifies the various system administrators, analysts etc who will have access to the database. |
| Workflow process Data | * Job\_Id * Dept * Responsible * Activity\_status * Activity\_start * Activity\_end * Activity\_responsible * Candidates\_allocated * Candidates\_deployed * Job\_satus | This data will include all the activities and statuses of different departments during the whole process flow. Data from here will be fetched according to requirement for the different views of the different departments. |
| QA | * QA\_forms * QA\_results | This data will include the QA to each job |
| Status Updates | To be specified in design. | This data is used to inform the clients of the workflow status. |

Table - Database Requirements

# Data Format

The system will store all the user account information as well as their job postings’ information. Furthermore, data generated by the various departments in the flow-process and passed on will also be stored. All data will be stored in logical SQL based tables. Each new table generated will be connected to the previous one through a foreign key. The tables and dependencies will form a tree-like hierarchical structure.

# Structure

The structure will define the data view of the logical database. It will adopt the hierarchy of the database tables defined by their foreign key relationships. This will also control the order in which the tables are accessed. It will determine the structure of the other components and the behavior of the logical database at runtime.

# Selections

The selections define a selection screen, which will form the user interface of the executable programs that use the logical database. Its layout will be determined by the structure.

# Data Integrity

Data integrity will be ensured through the usage of constraints. When an integrity constraint will be applied to a table, all data in the table must conform to the corresponding rule. When data is modified, or new data entered, the database will ensure that it the new data satisfies the integrity constraint, without any need to do checking within the program.

|  |
| --- |
| Client\_Id |
| Client\_name |
| Client\_address |
| Client\_contact |
| Client\_email |

Here, each client will have a unique ID which will be of type integer and length not more than 4. Client\_ contact will contain the phone nos., being all numeric in nature, and not more than 10 digits’ length. Client\_name will be alpha numeric and not more than 70 characters. Client\_email will be alphanumeric and not more than 40 characters’ length.

|  |
| --- |
| Job\_Id |
| Job\_title |
| Job\_description |
| Job\_type |
| NoOfPositions |
| Additional\_details |

Here, Client\_name is the foreign key linking to previous table. Job\_Id is alphanumeric in nature and not more than 20 characters. Job\_ description has a limit of 500 characters. Job\_type will be alphabetic and of length 2. Number\_of\_positions is numeric in nature and not exceeding length of 3.

|  |
| --- |
| Type\_of\_user |
| Dept\_Id |
| User\_Id |
| User\_name |
| User\_contact |
| User\_email |
| User\_active |

Dept will be numeric, not exceeding 2 digits. User\_Id and User\_contact will be numeric not exceeding 10 digits. User\_name will be alphabetic not exceeding 25 characters.

|  |
| --- |
| Dept\_Id |
| Responsible |
| Activity\_status |
| Actvity\_start |
| Activity\_end |
| Activity\_responsible |
| Candidates\_allocated |
| Candidates\_deployed |
| QA\_forms |
| QA\_results |

The Job\_Id is the foreign key linking to Job Data. Dept\_Id foreign key from User Data. Candidates\_allocated and Candidates\_deployed are numeric and not more than 3 digits each. The different columns will be formatted to be viewed as required in the different views pertaining to each department, e.g. Allocation department will have the view including department code, responsible, activity status, activity start date, activity end date, activity responsible, candidates allocated.

# Other Requirements

# Output Designs

The system will use reports to output information. The reports shall be created from tables.

# Client report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Client\_Id | Client\_name | Job\_Id | Job\_title | Status |
|  |  |  |  |  |  |

# Open opportunities report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Job\_Id | Job\_title | Status | Dept\_Id | SLA |
|  |  |  |  |  |  |

# Candidate Allocated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Job\_Id | Job\_title | Status | Candidate\_Id | Candidate\_name |
|  |  |  |  |  |  |

# Department SLA’s

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Dept\_Id | Qty Activities | SLA |
|  |  |  |  |

# Activities per Department

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Dept\_Id | Qty Activities | Status |
|  |  |  |  |

# Approval

The signatures below indicate their approval of the contents of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Role | Name | Signature | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |