**Analysis, Modelling and Design**

**ISMG – 6060, Fall 2018**

**Professional and Scientific Staff Management**

**New Temporary Staff Hiring System**

**Submitted by – Yash Nigam**

**Table of contents**

**Executive Summary …………………………………………………………..3**

**Project Identification and Systems Request……………………………...4**

**System Proposal……………………………………………………………….6**

**Requirements Definition and Concept Map……………………………..17**

**Functional Model……………………………………………………………..20**

**Structural Model………………………………………………………………27**

**Behavioral Models……………………………………………………………33**

**Moving on to Design…………………………………………………………37**

**Class and Method Design…………………………………………………..39**

**Data Management Layer…………………………………………………….53**

**User Interface Design………………………………………………………..59**

**Physical Architecture Layer Design………………………………………63**

**Conversion and Test Plan…………………………………………………..64**

# Executive Summary

Professional and scientific staff management (PSSM) specializes in high skilled temporary staffing placement for their clients. Their core competency is shortlisting, interviewing and getting right candidates for their clients, with clients, they negotiate the contracts, renew them and work on their specific staffing requests. The current staffing fulfillment system comprises of spreadsheets on company’s network, this creates inconsistency with the long hiring process and cause many errors and discrepancies, it is also inefficient for collecting client and candidate personal data. It is unable to provide a reliable, accessible and structured process for PSSM employees or the clients.

The new system hosts a secure consolidated database with data and visual analytic capabilities ensuring a safe and efficient system for the growth of PSSM. A basic web application configuration would be required to login to the system, the new system would have CRM and will be secured through firewall, the functional IT part will be outsourced so that PSSM staff can focus on their core business operation. The main process is divided in three parts- Opening a staffing request, placing a candidate and filling the staffing request. The basic responders to this process will be the clients, contract managers, placement and agreement departments. To achieve this functional model is created through use case and use case description associated with each of them, this describes the interaction between actors and system, along with this, detailed overall process in described through Activity Diagram and structural models of the system.   
Outputs from functional modelling will be used for structural modelling, CRC cards, class diagram shows the backbone of the solution model.

The last part of analysis will be modeling the behavior of the system, this shows how the structural models interact with each other with the help of sequence diagram, communication diagram, and behavioral state machine diagram.

# Project Identification and System Request

# **System Request**

|  |  |
| --- | --- |
| System Request: | Professional and science staff management (PSSM) |
| Project Sponsor: | Contract Manager, Placement manager **Professional and science staff management (PSSM)** |
| Business Need: | The project will enable multiple users to login into a common system simultaneously, one automated system will be implemented reducing the duplicate workload caused by paper requests and electronics requests and responses. Interactive dashboard will be implemented to keep a track of placement records. The contract manager, placement manager and arrangement manager should be able to communicate well with a consolidated database. |
| Business Requirements: | -Analytics to keep real time placement record on a dashboard -Real time interaction with clients for their requirements through interactive service portal -Immediate response system to alert the client and prospective employee during each step -Customer service unit to help clients and employees to resolve any doubts or clarifications -A client facing database will enable the clients to put in a request and as the request processes, the contract manager will get the notification in real time |
| Business Value: | This new consolidated system will reduce the manual effort and keep the real time information for all contracts, notify the contract manager and clients if the contract is coming to an end. It will also manage the candidate database and upgrade it continuously for clients to directly look for candidates of their own choice. With next phase an application will be implemented with this consolidated system enabling all the departments to fetch employee data, contract data and the placement status. It will give PSSM a competitive advantage over other staffing agencies. |
| Special Issues or Constraints: | -With the proposed system we will eliminate the redundancy and automate the entire information system for anyone with the PSSM login credentials and our clients -The integrated system will be deployed in modules, starting with the client facing portal to put in requests and answer any generic questions with automated customer service page. -For integration, technical expertise will be required for DevOps and analytics, along with that, customer service representatives will need training with the new integrated system. |

# System Proposal

**Summary**

Contract information from all the clients will be consolidated in a single database to reduce manual efforts and track real time status. This database will also have candidate section with all their information which will be updated continuously. Going forward employees from all the department will be working through the same database cutting out the dependency on each other. Real time interaction with the client will be enabled using analytics and dashboards so that the customer can see the transparency in the process. The new request system will enable the clients to directly submit the request in the system and it will be further taken care by PSSM, including contract manger, placement manager and arrangement manager.

The basic responders to this process will be the clients, contract managers, placement and agreement departments. To achieve this functional model is created through use case and use case description associated with each of them, this describes the interaction between actors and system, along with this, detailed overall process in described through Activity Diagram and structural models of the system.

**Evolutionary Work-plan and work breakdown structures**

|  |
| --- |
| **Enhanced Unified Process** |

|  | Duration | Dependency |
| --- | --- | --- |
| **I. Business Modeling** |  |  |
| a. Inception  1. Understanding current scenario 2.Uncover business process problems  3.Identification of potential projects | 1 day  0.5 day  0.5 day |  |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **II. Requirements** |  |  |
| a. Inception 1. Identification of requirement analysis techniques 2. Identification of requirements gathering techniques  3. Identification of functional and nonfunctional requirements  A. Perform JAD sessions  B. Perform document analysis  C. Conduct interviews  1. Interview project sponsor  2. Interview inventory system contact  3. Interview special order system contact  4. Interview ISP contact  5. Interview CD Selection Web contact  6. Interview other personnel  D. Observe retail store processes 4. Analyze current systems 5. Create requirements definition  A. Determine requirements to track  B. Compile requirements as they are elicited  C. Review requirements with sponsor | 0.5 day  0.5 day         3 days  5 days    0.5 day  0.5 day   0.5 day   0.5 day  0.5 day  1 day  1 day   4 days   1 day  5 days   2 days | II.a.1, II.a.2       II.a.3.A II.a.3.A  II.a.3.A II.a.1, II.a.2 II.a.3, II.a.4  II.a.5.A  II.a.5.B |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **III. Analysis** |  |  |
| a. Inception 1. Identify business processes 2. Identify use cases | 3 days  3 days | III.a.1 |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **IV. Design** |  |  |
| a. Inception 1. Identify potential classes | 3 days | III.a |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **V. Implementation** |  |  |
| a. Inception |  |  |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **VI. Test** |  |  |
| a. Inception |  |  |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **VII. Deployment** |  |  |
| a. Inception |  |  |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **VIII. Configuration and Change Management** |  |  |
| a. Inception 1. Identify necessary access controls for developed artifacts 2. Identify version control mechanisms for developed artifacts | 0.5 day  0.5 day |  |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **IX. Project Management** |  |  |
| a. Inception 1. Create workplan for the inception phase 2. Create system request 3. Perform feasibility analysis A. Perform technical feasibility analysis B. Perform economic feasibility analysis C. Perform organizational feasibility analysis  4. Identify project effort 5. Identify staffing requirements 6. Compute cost estimate 7. Create workplan for first iteration of the  elaboration phase 8. Assess inception phase | 1 day  1 day   1 day   2 days  2 days  0.5 day   0.5 day  0.5 day  1 day  1 day | IX.a.2        IX.a.3   IX.a.4  IX.a.5  IX.a.1  IX.a1 .a, II.a, III.a  IV.a, V.a, VI.a  VII.a, VIII.a,  IX.a, X.a, XI.a  XII.a |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **X. Environment** |  |  |
| a. Inception 1. Acquire and install CASE tool 2. Acquire and install programming environment 3. Acquire and install configuration and change management tool 4. Acquire and install project management tools | 0.5 day  0.5 day   0.5 day    0.5 day |  |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **XI. Operations and Support** |  |  |
| a. Inception |  |  |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |
| **XII. Infrastructure Management** |  |  |
| a. Inception 1. Identify appropriate standards and enterprise models 2. Identify reuse opportunities, such as patterns,  frameworks, and libraries 3. Identify similar past projects | 1 day   0.5 day      1 day |  |
| b. Elaboration |  |  |
| c. Construction |  |  |
| d. Transition |  |  |
| e. Production |  |  |

**Risk Assessment and feasibility analysis**

|  |  |
| --- | --- |
| **Risk** | The new consolidated database system might be a risk factor due to its introduction |
| **Likelihood of risk** | Medium probability |
| **Potential Impact** | Time will be consumed and delay in quality assurance |
| **Ways to address this risk** | The risk can be mitigated by appointing a systems analyst and an architect to design the database and train the team. |

**Feasibility Analysis Executive Summary**

**Technical Feasibility: Can We Build It?**

|  |
| --- |
| Familiarity with Functional area: Less familiarity generates more risk |
| * PSSM is familiar with the functional side of the proposed system, they are clear with its operation and implementation |
| Familiarity with Technology: Less familiarity generates more risk |
| * They are familiar with the technology, the database consolidation, retrieval, extraction and the dashboard analytical side is well planned and sorted out. With the new system, the technical specialty must be taken care by experts and external help will be required, medium familiarity. |
| Project Size: Large Projects have more risk |
| * PSSM has a large project which will be deployed in phases, this is a medium to high risk project. |
| Compatibility: The harder it is to integrate the system with the company’s existing technology, the higher the risk |
| * The system will be compatible as there is no such as-in system available, everything is created from scratch keeping up with business requirements and stakeholder demands.  **Economic Feasibility: Should We Build It?**  With the right amount of return on investment, the project is economically feasibleIntangible Costs and Benefits  -The system will be needing maintenance cost -PSSM will have to spend money to hire experienced architects to design the database -Biggest benefit will be the competitive edge over the competitors -Robust and flawless system will attract more clients and upfront cost can be increased, resulting in profit |

# **Organizational Feasibility: If We Build It, Will They Come?**

|  |
| --- |
| **Strategic Alignment** |
| Stakeholders |
| * Is the project strategically aligned with the business? Yes |
| * Project Champion(s) Clients, PSSM employees and stakeholders |
| * Senior Management |
| * Users |
| * Other stakeholders |

# **Additional Comments**

Cost benefit analysis  


|  |
| --- |
|  |



Requirements Definition and Concept Map

**Non-Functional Requirements  
  
1. Operational requirement**

* 1. The system will maintain a backup of all the placement records and candidates
  2. The system will be accessible through PSSM portal for employees and clients
  3. The dashboard will be supported 24\*7
  4. The system will be supported on all operating platforms

1.5 The customer support unit will be operational all year round.

**2. Performance requirement**

* 1. The system will provide analytics capabilities to search for the right candidate
  2. The consolidated database would let the clients fetch candidates and put a request associated with their selection.
  3. The system will alert the Contract manager, placement manager and arrangement manager whenever a request is put in and the dashboard will be updated simultaneously.
  4. The system will provide an update and alert about the changes, termination or negotiation in the contract with the client.
  5. Multiple users can use the database simultaneously.

**3. Security requirement**

* 1. Access to the client contract is visible to client and contract manager only.
  2. Only client can put in a request and further modifications can be done by approval of the arrangement manager.
  3. The candidate information is visible to only the clients and concerned departments.

**4. Cultural and political requirements**

* 1. The system will keep the candidate information and client contract information safe
  2. All the regulatory compliance and ethics best practices will be strictly applied.

**Functional Requirement**

**1. Creating Requests**

1.1 Clients can raise a staffing request through the consolidated centralized system

1.2 The system will receive the staffing request from the client and register it in the database.

1.3 The system will automatically check the contract and proceed further if it is valid.

1.4 The system will revert to the client about their request and they can track the entire process.

**2. Real time tracking** 2.1 the client and the PSSM staff can track a staffing request throughout its fulfillment

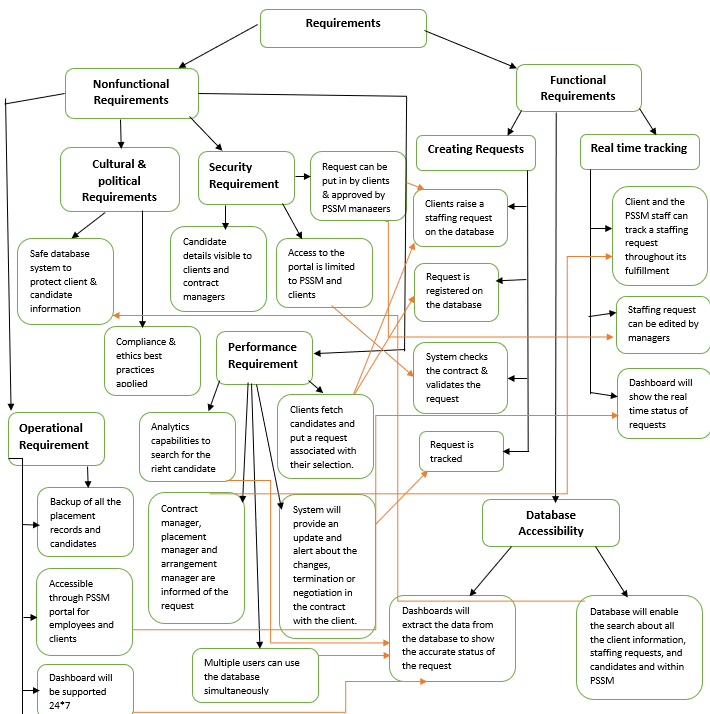
2.2 Staffing request can be edited real time by a single department only, other departments can only view it. Simultaneous editing will be locked.

2.3 Dashboard would show the real time status of applicants and no. of staffing request they are suited for.

**3. Database Accessibility**

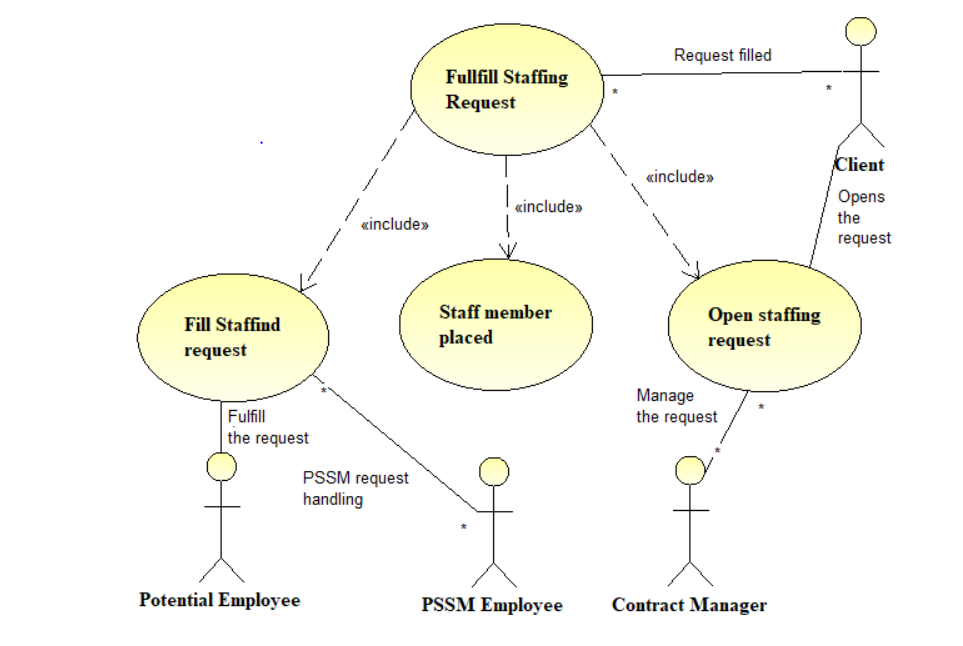
3.1 The dashboards will extract the data from the database, enabling the clients to see the real-time status of their request and they can also see the candidates suited for their requirements.

3.2 The database will enable the search about all the client information, staffing requests, and candidates and within PSSM

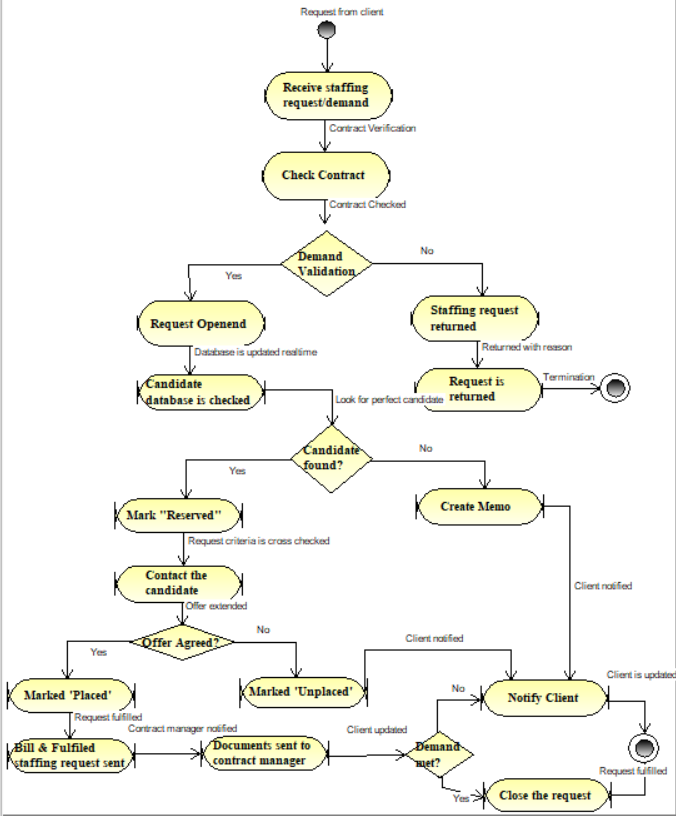


# Functional Model

**Use Case Diagram for PSSM**



**Activity Diagram**



**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Fulfill staffing request | | ID: 1 | Importance Level: High |
| Primary Actor:  Client | Use Case Type: Overview/Essential | | |
| Stakeholders and Interests:  Client | | | |
| Brief Description: Contract manager at PSSM receives the client demand and verifies the client contract, placement department investigates the consolidated database to match the requirements and the arrangement department negotiates placement terms and send the candidate to the client. | | | |
| Trigger: Client puts in a request to fulfill their demand for a staff member  Type: External | | | |
| Relationships:  Association: Contract manager, placement manager and arrangement manager  Include: Client  Extend:  Generalization: Fulfill staffing requests and update the database | | | |
| Normal Flow of Events:   1. Contract manager opens the staffing request submitted by the client 2. Placement department finds a suitable fit for the request from their database 3. Arrangement department negotiates the terms and condition of the opportunity | | | |
| SubFlows:   1. Contract with the client is verified first, then the request is forwarded 2. The client is notified if the request is accepted or denied 3. The client has the access to candidate database where they can look the candidate of their own choice 4. With real-time tracking and notification, updates are sent to all departments about the process and fulfillment | | | |
| Alternate/Exceptional Flows: When a request is not fulfilled, the client is notified with reason, all the other processes are carried out by the contract department at PSSM. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Open staffing request | | ID: 2 | Importance Level: High |
| Primary Actor: Client, Contract Manager | Use Case Type: Detailed/Real | | |
| Stakeholders and Interests:  Contract manager, candidate, placement department | | | |
| Brief Description: A new staffing request is received by PSSM, contract manager reviews the terms with clients  And puts the request in the database to find a suitable candidate for the role. If the demand is not valid, the request is sent back to the client with the explanation. | | | |
| Trigger: Client puts the request in the database system or via phone, email and requests a demand  Type: External and Internal | | | |
| Relationships:  Association: Contract manager, placement manager  Include: Client  Extend: Valid request  Generalization: Update the database with new request and contract verification with client | | | |
| Normal Flow of Events:   1. Client demands a new temporary staff, they put in a request for the same 2. Contract manager reviews the contract deal with the client 3. Contract manager validates the request and request proceeds in the database 4. Placement department is notified of the new requirement. | | | |
| SubFlows:   1. The contract is matched with the contract no. and the details that the client has put in the request 2. Database flags and acknowledges the request and proceeds to notify the placement department 3. Placement department works on the request and verifies the requirements and the available candidates | | | |
| Alternate/Exceptional Flows:  If the request is invalid i.e the contract details do not match the request details, a letter is generated and sent to the client notifying about the reason and measure to eliminate it. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Place staff member | | ID:3 | Importance Level: High |
| Primary Actor:  Placement Department | Use Case Type: Detailed/Essential | | |
| Stakeholders and Interests:  Placement department, contract manager, arrangement department, candidate | | | |
| Brief Description: The placement department verifies the position, qualification, and experience of the candidate mentioned on the request and matches it in the database. | | | |
| Trigger: After approval from the contract manager, the database sends the staffing request to the placement department to look for a suitable candidate to meet the client’s requirement.  Type: Internal | | | |
| Relationships:  Association: Placement department, candidate  Include: Client  Extend:  Generalization: Find the right candidate | | | |
| Normal Flow of Events:   1. An employee from the placement department checks the position, qualification, and experience specified in the staffing request in the database 2. If the perfect candidate is found, they are marked as ‘Reserved’ | | | |
| SubFlows:   1. The database gives all the available candidates matching the client's request. 2. The database forwards these details to the arrangement department notifying them about the possible matches | | | |
| Alternate/Exceptional Flows:  If a candidate is not found in the database, an ‘unable to fill’ memo is generated and sent to the arrangement department to further pass it on to the client or the contract manager. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Fill Staffing request | | ID: 4 | Importance Level: High |
| Primary Actor: Arrangement Department, Candidate | Use Case Type: Detailed/Real | | |
| Stakeholders and Interests:  Client, candidate, arrangement department | | | |
| Brief Description: A candidate is contacted, and the details and negotiations are discussed and agreed upon. The candidate details are forwarded to the client and they report to the client. The staffing request is closed | | | |
| Trigger: Possible candidate marked as ‘reserved’ in the database and the arrangement department is notified  Type: Internal / External | | | |
| Relationships:  Association: Candidate, Clients and arrangement department  Include:  Extend:  Generalization: Put the right candidate on the job and close the client request | | | |
| Normal Flow of Events:   1. Arrangement department contacts the candidate 2. Details are settled and discussed with negotiations 3. The candidate is placed with the client by the PSSM employee 4. The staffing request is closed. | | | |
| SubFlows:   1. Database marks the candidate as placed 2. Database closes the staffing request 3. Copy of bill and staffing request is sent to the client 4. If the candidate is not found, the additional memo is sent to the client | | | |
| Alternate/Exceptional Flows: If the staffing request is not fulfilled, the memo is filed by the contract manager. | | | |

**Functional Model Verification and Validation**

-One event of a use case description is present in each activity in the activity diagram

-In the activity diagram, all objects show as object nodes are in events of the use-case diagram.

-The orders of events in the use case diagram are the same as the order in the activity diagram.

-Each use case diagram has a use case description.

-All the actors in the use case description are matching the use case diagram.

-In the use case description, the stakeholders are shown as actors for use case diagram.

-All the relationships from use case description are shown in the use case diagram.

-All the UML standards are followed by the use case diagram.

Structural Model

**CRC Card**

For PSSM staffing solution, there are three classes  
1.Contract

2.Candidate

3.StaffingRequest

**Front:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class Name:** Contract | **ID:** 1 | | **Type:** Concrete |
| **Description:** This class describes the contracts negotiated between the client and PSSM staffing agency | | | **Associated Use Cases:** 3 |
| **Responsibilities**  Responsibility1  Addition to the consolidated contract table  Assigned to the staffing request  Assign to PSSM candidate (another department)  Add the candidate | | **Collaborators**  Collaborator1  Database  StaffingRequest  Candidate  Candidate | |

**Back:**

|  |
| --- |
| **Attributes:**  Attribute1  Contract\_ID (Integer)  Contract\_Date (Date)  Client\_Name (String)  Client\_Address (String)  Client\_Phone (String)  Client\_Email (String)  CandidateDetail (String array) |
| **Relationships**:  **Generalization (a-kind-of):**  **Aggregation (has-parts):**  **Other Associations:** Candidate,StaffingRequest |

**Front:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class Name:** Candidate | **ID:** 2 | | **Type:** Concrete |
| **Description:** This class represents the available candidate details visible to the client and PSSM using consolidated database. | | | **Associated Use Cases:** 3 |
| **Responsibilities**  Responsibility1  Added\_Employee Added to contract  Hold/reserve Place | | **Collaborators**  Collaborator1  Database  Contract  StaffingRequest  StaffingRequest | |

**Back:**

|  |
| --- |
| **Attributes:**  Attribute1  EmpID (Integer)  EmpStatus (String)  EmpName (String)  EmpAddr (String)  EmpPhone (String)  EmpEmail (String)  Experience (Integer)  Qualification (Free text field)  Expected Salary (Float)  ContractID (Integer array) |
| **Relationships**:  **Generalization (a-kind-of):**  **Aggregation (has-parts):**  **Other Associations:** Contract, StaffingRequest |

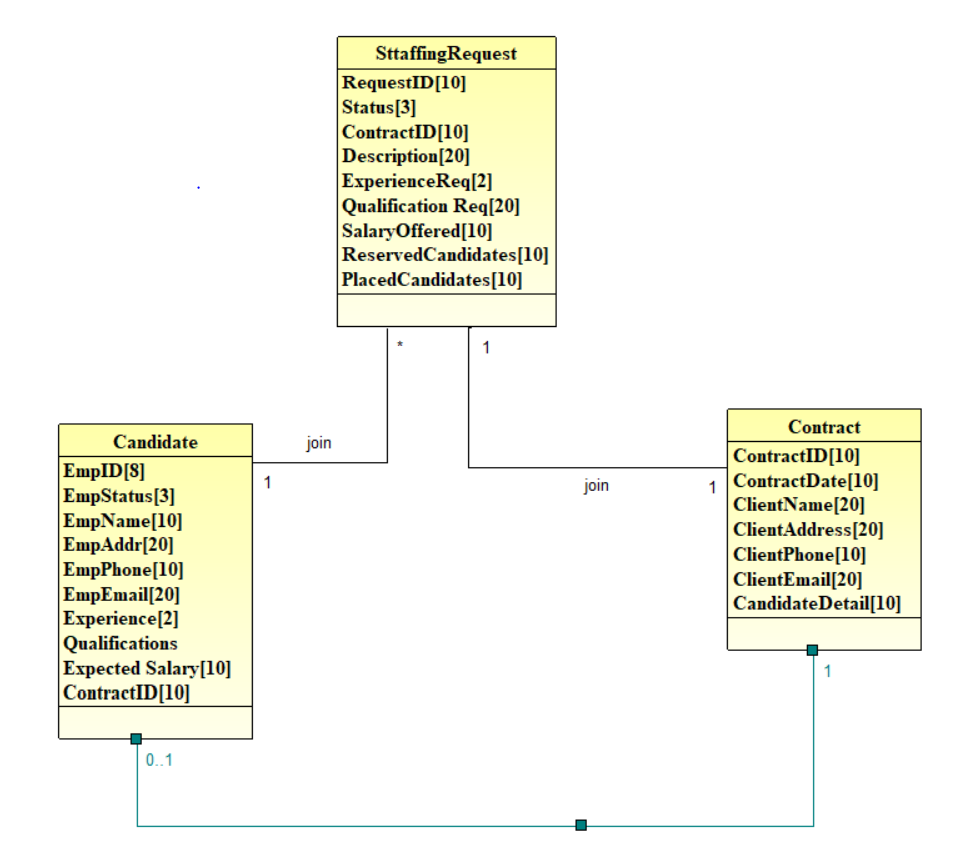
**Front:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class Name:** StaffingRequest | **ID:** 3 | | **Type:** Concrete |
| **Description:** This class represents staffing request used to manage placements for PSSM clients and candidates. | | | **Associated Use Cases:** |
| **Responsibilities**  Responsibility1  Add to table  Assign to contract  Reserve candidate  Place candidate  Close | | **Collaborators**  Collaborator1  Database  Contract  Candidate  Candidate  Database | |

**Back:**

|  |
| --- |
| **Attributes:**  Attribute1  RequestID (Integer)  Status (String)  ContractID (Integer)  Description (String)  Experience reqst (Integer)  Qualification reqst (String)  Salary Offered (Float)  Reserved Candidate (Integer Array)  Placed Candidate (Integer Array) |
| **Relationships**:  **Generalization (a-kind-of):**  **Aggregation (has-parts):**  **Other Associations:** Contract, Candidate |

**Class Diagram**



**Verification and Validation for Functional model**

-CRC cards🡪Classes on the class diagram

-CRC card responsibility🡪Operation on class diagram

-CRC card attributes🡪Class diagram attributes

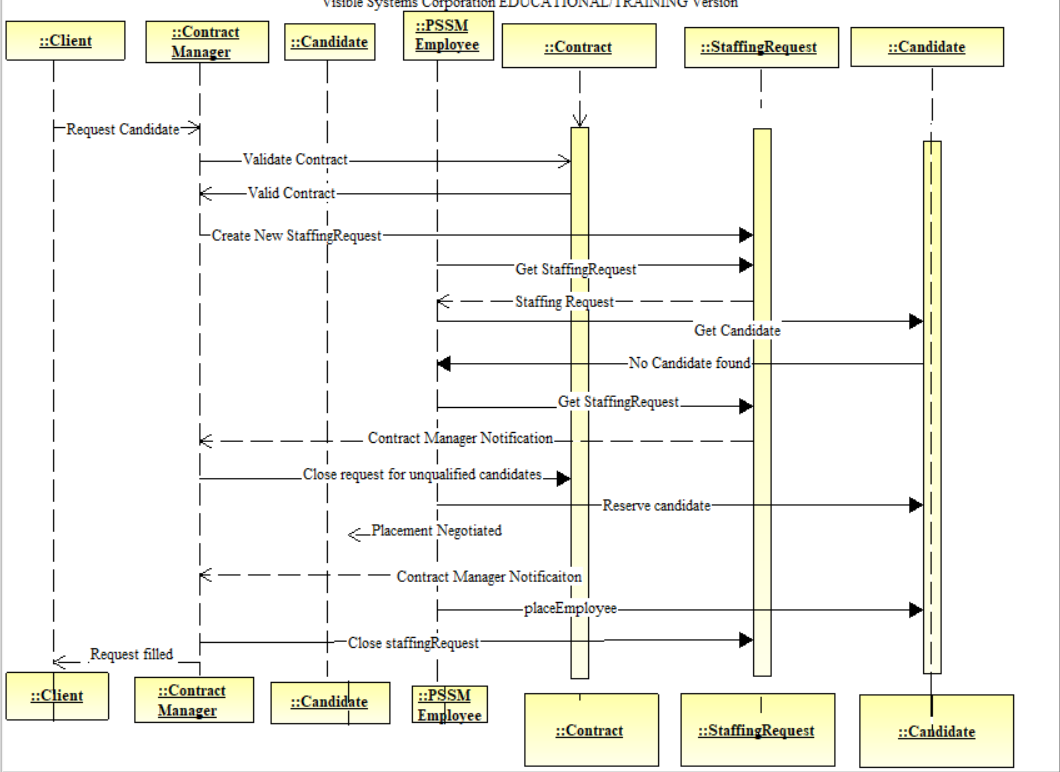
-CRC relationships🡪Notations are appropriate

-CRC card collaborators🡪CRC card relationship and class diagram association

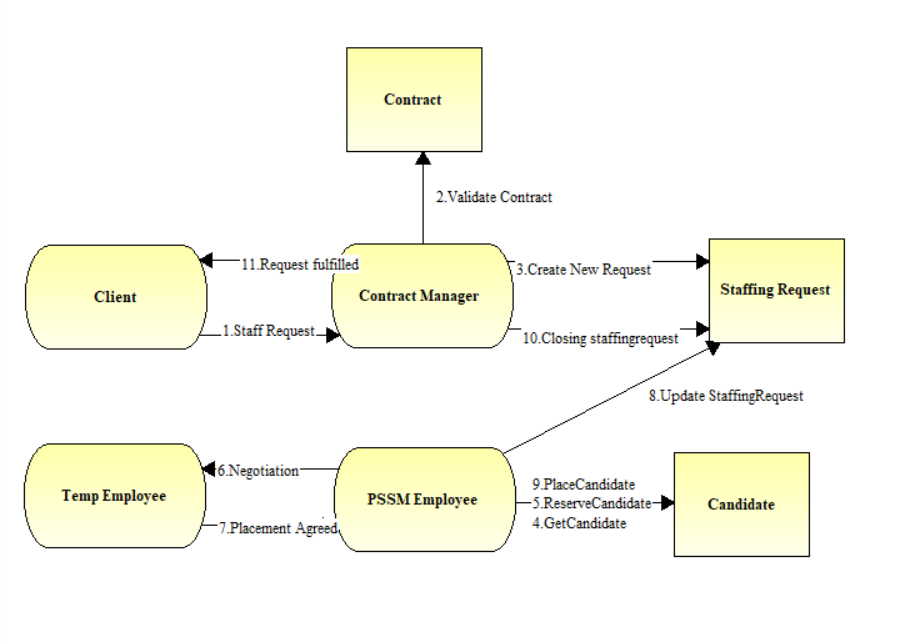
-Classes reflect unique characteristics about intersection of connecting classes.

Behavioral Models

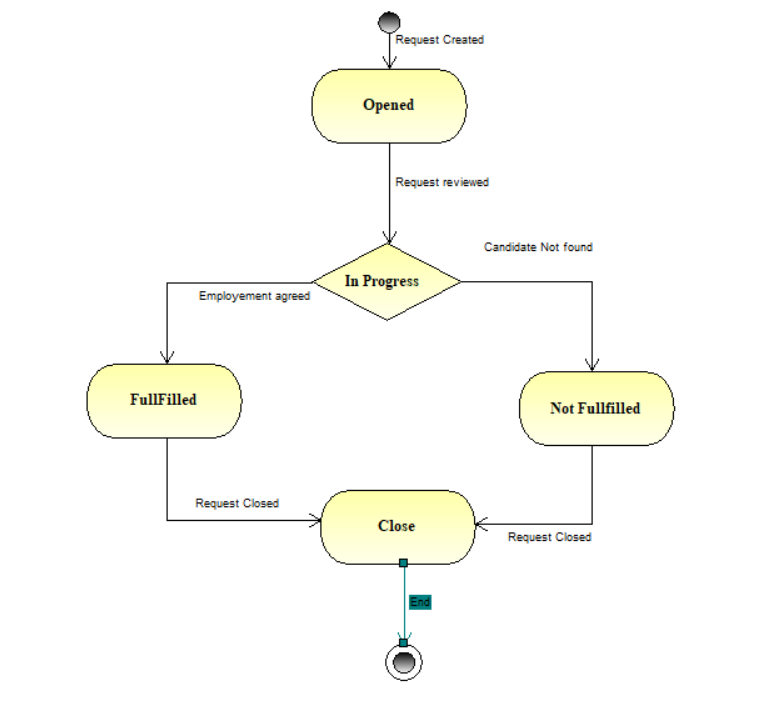
**Sequence Diagram**

****

**Communication Diagram**



**Behavioral state machine**

****

**Crude Analysis**



**Verification and validation of behavioral model**

-All actors and objects in sequence diagrams included in communication diagrams.

-Sequence diagram messages → communication diagram associations

-Sequence diagram messages → communication diagram messages

-Sequence diagram guard conditions → communication diagram guard conditions

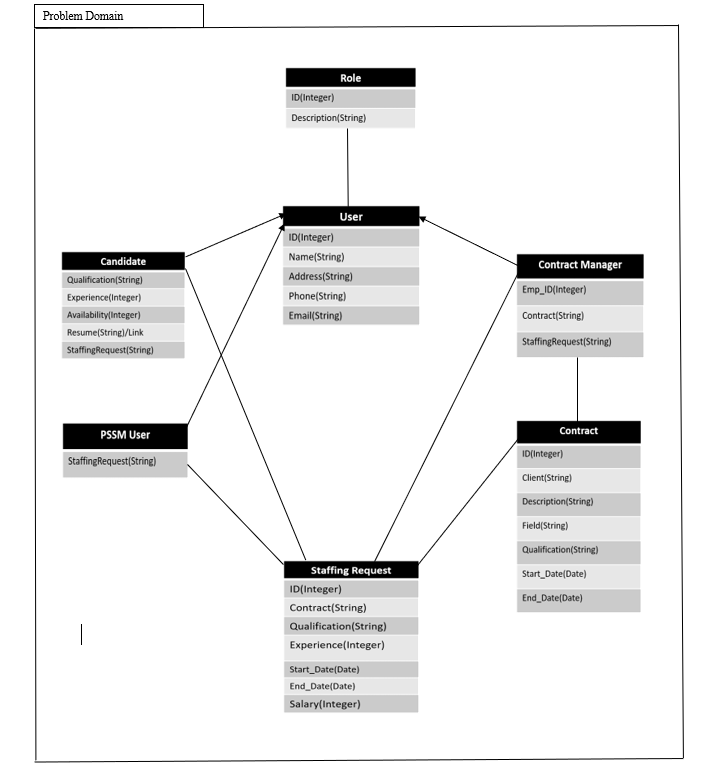
-Communication diagram sequence numbers reflect sequence diagram order

-Behavior state machine transitions → sequence and communication diagram messages

-Behavior state machine transitions → CRUDE matrix classification

-CRUDE matrix entries → messages between actors and objects, behavior state machine transitions

Moving on to Design

**Package Diagram**

**Validation and Verification walkthrough**

-Derived from the Class Diagram and Crud Matrix

-The problem domain is created based on identified packages from the previous classes and reviews

-All the dependency matrices are based on relationships of the communication diagram, Crude matrix, and association with the Class Diagram.

**Design Strategy**

According to the current functionality and understanding of the system, I would recommend Packaged Software strategy because, PSSM does not have a dedicated Information Technology department which supports the organization, earlier they were doing a lot of manual and paperwork, resulting in a system with slow processing and less involvement of automated processes. With the proposed system we are trying to add a consolidated database and analytical capabilities to the entire process, which requires IT expertise and capabilities, with the current staff, in-house development is not possible, the cost of hiring developers, testers and IT managers would result in a very costly affair. With the packaged software, PSSM has a variety of systems to browse and select the one which makes PSSM’s goals and requirements. It is more efficient to buy a product which is already created, tested and proven, along with this, it is easier to install in a short period of time, the biggest advantage with this system is the conflict resolution and expertise provided by the vendor who developed the software/system. Systems like ERP, enterprise resource planning, provides the advantage of automation to the entire company. They also provide modifications to the software according to the needs of the customer.

Class and Method Design

**Front:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class Name:** Contract | **ID:** 1 | | **Type:** Concrete |
| **Description:** This class describes the contracts negotiated between the client and PSSM staffing agency | | | **Associated Use Cases:** 3 |
| **Responsibilities**  Responsibility1  Addition to the consolidated contract table  Assigned the staffing request  Assign to PSSM candidate (another department)  Hire the candidate | | **Collaborators**  Collaborator1  Database  StaffingRequest  Candidate  Candidate | |

**Back:**

|  |
| --- |
| **Attributes:**  Attribute1  Contract\_ID (1..1) (integer)  Contract\_Date (0..1) (Date)  Client\_Name (1..1) (String)  Client\_Address (1..1) (string)  Client\_Phone (1..1) (string)  Client\_Email (1..1) (string) CandidateDetail (0..1) (String array) |
| **Relationships**:  **Generalization (a-kind-of):**  **Aggregation (has-parts):**  **Other Associations:** Candidate,StaffingRequest |

**Front:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class Name:** Candidate | **ID:** 2 | | **Type:** Concrete |
| **Description:** This class represents the available candidate details visible to the client and PSSM using consolidated database. | | | **Associated Use Cases:** 3 |
| **Responsibilities**  Responsibility1  Added\_Employee Added to contract  Hold/reserve Place | | **Collaborators**  Collaborator1  Database  Contract  StaffingRequest  StaffingRequest | |

**Back:**

|  |
| --- |
| **Attributes:**  Attribute1  EmpID (0..1) (integer)  EmpStatus (1..1) (string) [available, reserved, placed]  EmpName (1..1) (string)  EmpAddr (1..1) (string)  EmpPhone (1..1)(string)  EmpEmail (1..1) (string)  Experience (0..1) (integer)  Qualification (0..1)(free text field) [bachloers, masters, doctorate]  Expected Salary (1..1) (float)  ContractID (1..1) (integer) |
| **Relationships**:  **Generalization (a-kind-of):**  **Aggregation (has-parts):**  **Other Associations:** Contract, StaffingRequest |

**Front:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class Name:** StaffingRequest | **ID:** 3 | | **Type:** Concrete |
| **Description:** This class represents staffing request used to manage placements for PSSM clients and candidates. | | | **Associated Use Cases:** |
| **Responsibilities**  Responsibility1  Add to table  Assign to contract  Reserve candidate  Place candidate  Close | | **Collaborators**  Collaborator1  Database  Contract  Candidate  Candidate  Database | |

**Back:**

|  |
| --- |
| **Attributes:**  Attribute1  RequestID (1..1) (integer)  Status (0..1) (string) [Open, closed]  ContractID (1..1) (integer)  Description (0..1) (free text field)  Experience reqst (0..1) (integer)  Qualification reqst (0..1) (string) [bachleors, masters, doctorate]  Salary Offered (1..1) (float)  Reserved Candidate (1..1) (int array)  Placed Candidate (1..1) (int array) |
| **Relationships**:  **Generalization (a-kind-of):**  **Aggregation (has-parts):**  **Other Associations:** Contract, Candidate |

**Contract for each method in that class**, used OCL to specify the preconditions and the post-condition.

|  |
| --- |
| **Method Name**: Addition to contract table **Class Name**: Contract **ID**: 1 |
| **Clients (Consumers) :** Client, PSSM |
| **Associated use cases :** AddCandidateDetail |
| **Description of responsibilties:**  Adding a new staffing request from the client to the consolidated database which would enable the employees of PSSM and the client to look at the progress of the staffing request throughout the process of hiring a candidate. |
| **Arguments Received :**  Staffing requirements checked with the existing contract with the client |
| **Type of value returned :**  Staffing request number/ID to track the request and add or edit anything with the candidate detail |
| **Pre-conditions:**  Client should have an active contract with PSSM and all the terms are to be agreed on beforehand |
| **Post-conditions:**  The staffing request should be kept in database for the ease of tracking and reporting & for documentation purposes. |

|  |
| --- |
| **Method Name**: Assign staffing request **Class Name**: Contract **ID**: 2 |
| **Clients (Consumers) :** PSSM |
| **Associated use cases :** StaffingRequest |
| **Description of responsibilties:**  Once the staffing request is put in the database, the request goes to employees responsible for finding and contacting the candidates with the most suitable profile given on the description. |
| **Arguments Received :**  Staffing requirement requirement should be aligned to the contract description |
| **Type of value returned :**  The shared database will provide the list of candidates to interview and shortlist based on the staffing request and the client will be notified |
| **Pre-conditions:**  Staffing request should be valid and PSSM also checks for the valid contract and details related to the candidate requirement |
| **Post-conditions:**  The staffing request is passed on to the placement department for the selection of the right candidate. |

|  |
| --- |
| **Method Name**: Assign candidate to placement department **Class Name**: Contract **ID**: 3 |
| **Clients (Consumers) :** PSSM |
| **Associated use cases :** CandidateDetail |
| **Description of responsibilties:**  The candidates would be shortlisted from the database and the requirement from the contract will be matched |
| **Arguments Received :**  None |
| **Type of value returned :**  Candidates from the consolidated database, shortlisted on the basis of the client requirements |
| **Pre-conditions:**  Candidate must be available for the time of contract and the client holds a valid contract with PSSM |
| **Post-conditions:**  The candidates list is passed onto the arrangement department for contacting the candidates and hiring them. |

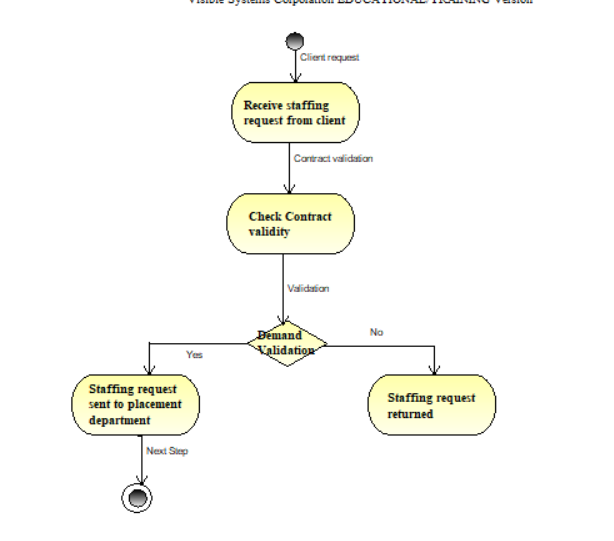
|  |
| --- |
| **Method Name**: Hiring the candidate **Class Name**: Contract **ID**: 4 |
| **Clients (Consumers) :** Client, PSSM |
| **Associated use cases :** CandidateDetails |
| **Description of responsibilties:**  The arrangment department hires the candidates, settle the benefits and negotiations and report the hired candidate profile to the client |
| **Arguments Received :**  If there is no candidate, the client is informed and the search is continued for a suitable candidate |
| **Type of value returned :**  Candidates that match the requirement of the client and agree to the terms and conditions of the employment |
| **Pre-conditions:**  Candidate list is provided by the placmeent department and the selected candidate is available to the tenure. |
| **Post-conditions:**  Client is sent the hiring details of the candidate and the candidate is marked as placed by the manager. |

**Method specification for each method in the class with** **Structured English** **and Activity Diagrams for the algorithm specification**

Addition to contract table

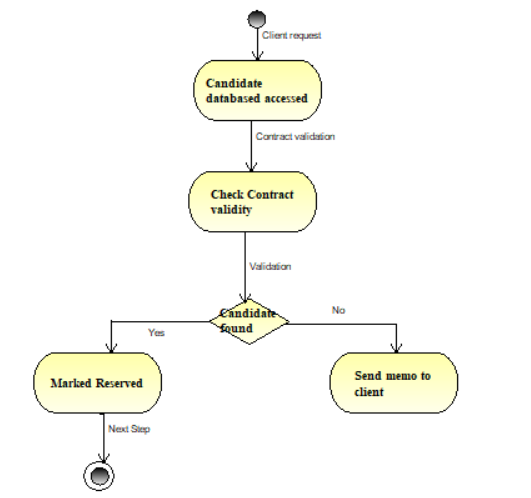
|  |  |  |
| --- | --- | --- |
| **Method Specification Form** | | |
| Method Name: **Addition to contract table** | Class Name:  **Contract** | ID: 1 |
| Contract ID:1 | Programmer: Jerry Seinfeld | Date due: 12/31/2018 |
| Language used: Java, SQL | | |
| Arguments received | | |
| Data Type: | Notes: | |
| Messages sent & arguments passed | | |
| ClassName.MethodName | Data Type | Notes: |
| Contract.ContID  Contract.ContDate  Contract.ClientDetail  Contract.CandidateInfo | Integer  Date  String  String Array | ContractID gives unique key to track the request  To track the project  Client information for staffing request  Candidate details in the consolidated request |
| Arguments returned | | |
| Data Type: | Notes: | |
| Algorithm specification:  Check the client’s contract with PSSM before inserting the staffing request If the contract is active, proceed else Return the request with reason slip and notify the contract manager. For all valid contracts Update the database with a new entry as ContractID | | |
| Miscellaneous notes: | | |
|  | | |

Activity Diagram



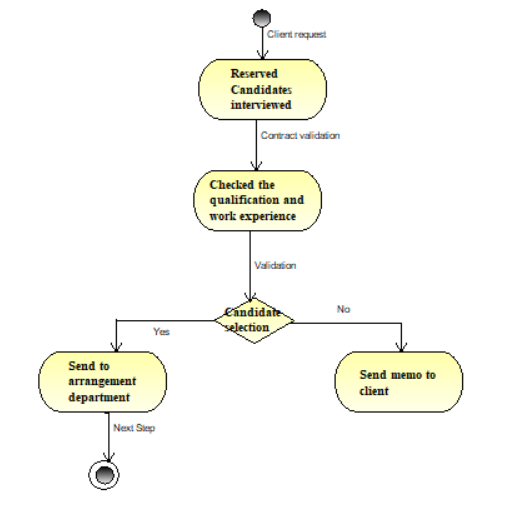
Assign to staffing request

|  |  |  |
| --- | --- | --- |
| **Method Specification Form** | | |
| Method Name:  Assign staffing request | Class Name:  **Contract** | ID: 2 |
| Contract ID: 1 | Programmer: George Castanza | Date due:  12/31/2018 |
| Language used: Java, SQL | | |
| Arguments received | | |
| Data Type: | Notes: | |
| Messages sent & arguments passed | | |
| ClassName.MethodName | Data Type | Notes: |
| Contract.ContID  Contract.ContDate  Contract.ClientDetail  Contract.CandidateInfo | Integer  Date  String  String Array | ContractID gives unique key to track the request  To track the project  Client information for staffing request  Candidate details in the consolidated request |
| Arguments returned | | |
| Data Type: | Notes: | |
| Algorithm specification:  The placement department starts looking for candidates from the consolidated database and verirfying the contract  If contract is valid, return  Else, search for the criteria from the staffing request and contract.  List of eligible and available candidates sent to the arrangement department.  If, candidate is not found, report the client with acknowledgment slip | | |
| Miscellaneous notes: | | |
|  | | |



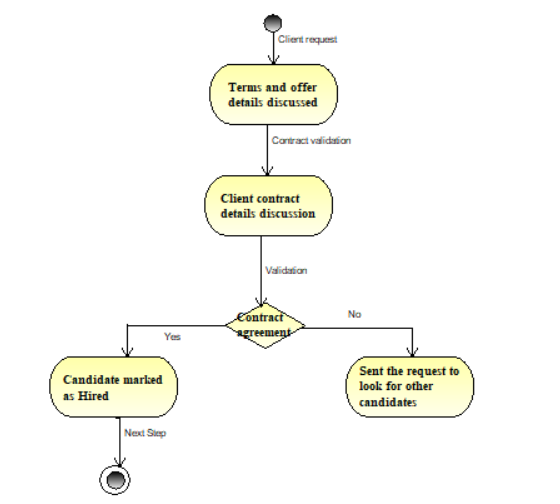
Assign candidate to placement department

|  |  |  |
| --- | --- | --- |
| **Method Specification Form** | | |
| Method Name:  Assign candidate to placement department | Class Name:  **Contract** | ID: 3 |
| Contract ID: 1 | Programmer: George Castanza | Date due:  12/31/2018 |
| Language used: Java, SQL | | |
| Arguments received | | |
| Data Type: | Notes: | |
| Messages sent & arguments passed | | |
| ClassName.MethodName | Data Type | Notes: |
| Contract.ContID  Contract.ContDate  Contract.ClientDetail  Contract.CandidateInfo | Integer  Date  String  String Array | ContractID gives unique key to track the request  To track the project  Client information for staffing request  Candidate details in the consolidated request |
| Arguments returned | | |
| Data Type: | Notes: | |
| Algorithm specification:  Placement department verifies the candidate profiles and work experience  If matched send the profiles for interview  If selected, send the profile to arrangemnt department  Else notifiy the client about the shrotage on candidate or requirements adjusment by them | | |
| Miscellaneous notes: | | |
|  | | |



Hiring the candidate

|  |  |  |
| --- | --- | --- |
| **Method Specification Form** | | |
| Method Name:  Hiring the candidate | Class Name:  **Contract** | ID: 4 |
| Contract ID: 1 | Programmer: Jerry Seinfeld | Date due:  12/31/2018 |
| Language used: Java, SQL | | |
| Arguments received | | |
| Data Type: | Notes: | |
| Messages sent & arguments passed | | |
| ClassName.MethodName | Data Type | Notes: |
| Contract.ContID  Contract.ContDate  Contract.ClientDetail  Contract.CandidateInfo | Integer  Date  String  String Array | ContractID gives unique key to track the request  To track the project  Client information for staffing request  Candidate details in the consolidated request |
| Arguments returned | | |
| Data Type: | Notes: | |
| Algorithm specification:  The shortlisted candidates are contacted to discuss terms and offer  If, the candidate agrees to the term and contract details  The candidate is given the offer letter and the candidate profile is sent to the client  Else, the candidate does not aceept the offer, the request is on hold and sent to ID = 2  The staffing request is closed and candidate is marked as Hired | | |
| Miscellaneous notes: | | |
|  | | |



Data Management Layer

Apply the rules of normalization to the class diagram to check the diagram for processing efficiency.

**First normal form**

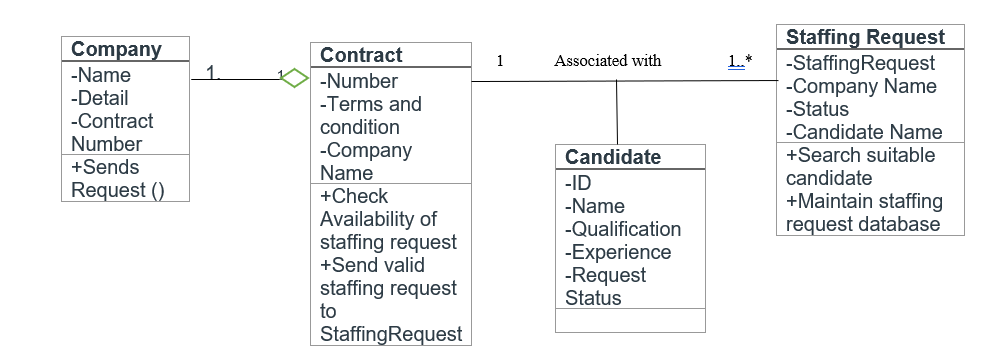
A table is said to be in 1NF, if there is no multivalued field in the table, with this diagram, there are no classes representing tables field with multi valued attributes, so they are all in first normal form

**Second normal form**

This table is in 2NF because, its already in 1NF and non-key attributes are dependent on the primary key, for instance two companies cannot have same names or contract numbers, so name is the primary key.

**Third normal form**

The table is in 1NF and 2NF as well the non-key attributes should not be dependent on any non-primary key, transitive dependency cannot be present. With the above diagram, no attributes of any tables are dependent on non-primary key.

****

**Develop a clustering and indexing strategy**

Clustering is one of the methods to improve the speed of accessing data from the relational database. In clustering, similar records are collected together physically so that similar records are together. With our tables for PSSM-DAM, Contract table, request table and staff table, it is easier to store data in these tables. Therefore, whenever a query is run to look for records, by their keys, types or any other attribute, the similar data can be pulled. With Interfile clustering, similar records/data is pulled from multiple sources/tables. With each table, there can be only clustering strategy, we will cluster data according to staff table, request table and PSSM table.

Index in a database is like an index in the book. It contains values from many columns in a table. With a query, indexing can be useful to find locations and details of records associated with the query, adding to it, a table can have unlimited number of indexes. In a decision support system, using many indexes can increase the response time. To perform join function, index keys can be created based on their foreign keys. With a large dataset, creating an index can be helpful for fields in grouping and sorting.

**Strategy will improve the performance of the database.**

With the introduction of tables at data management layer and various User interfaces like contract manager UI, Placement specialist UI and arrangement specialist UI, at the human computer interaction layer, the proposed system will enhance the hiring process at PSSM. The relational database requires an architectural design which is fulfilled by this system to database access and manipulation services. The basic function performed by the database are open, close, read, update, store, create and delete tables. These functions will be incorporated in staff database, request database and the contract database. With functions like real time updating will be supported by clustering and indexing resulting in a fast and efficient database system.



**Connectivity among the classes on the problem domain and data management layers**



**Communication diagram**



**Connectivity between Human computer interaction layer, Problem domain layer and Data management layer**

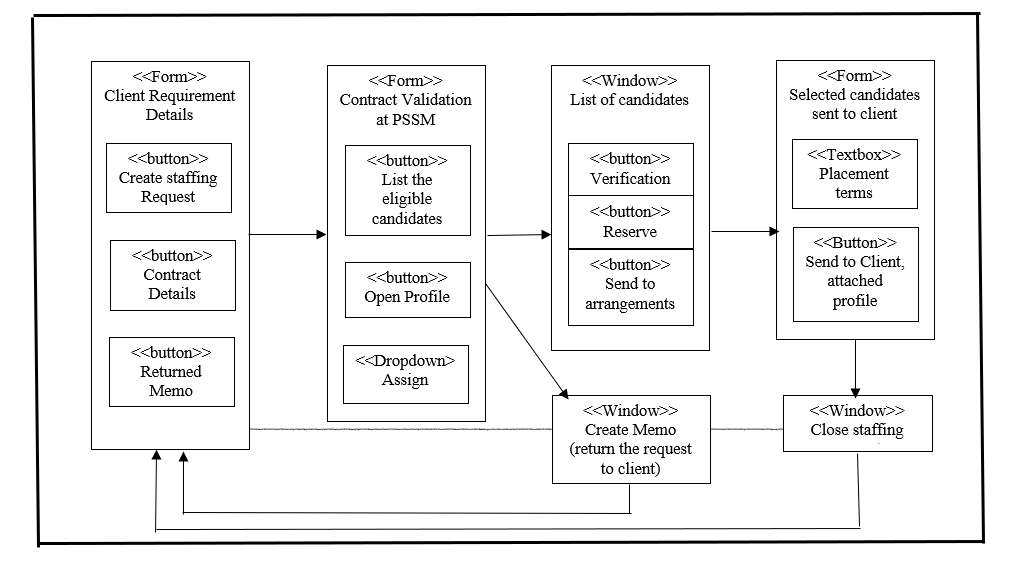


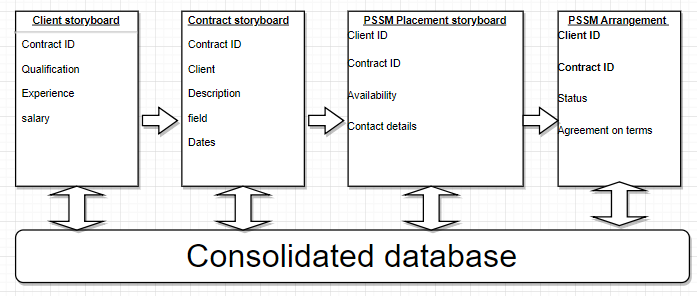
User Interface Design

**Windows Navigation Diagram, and design a storyboard**

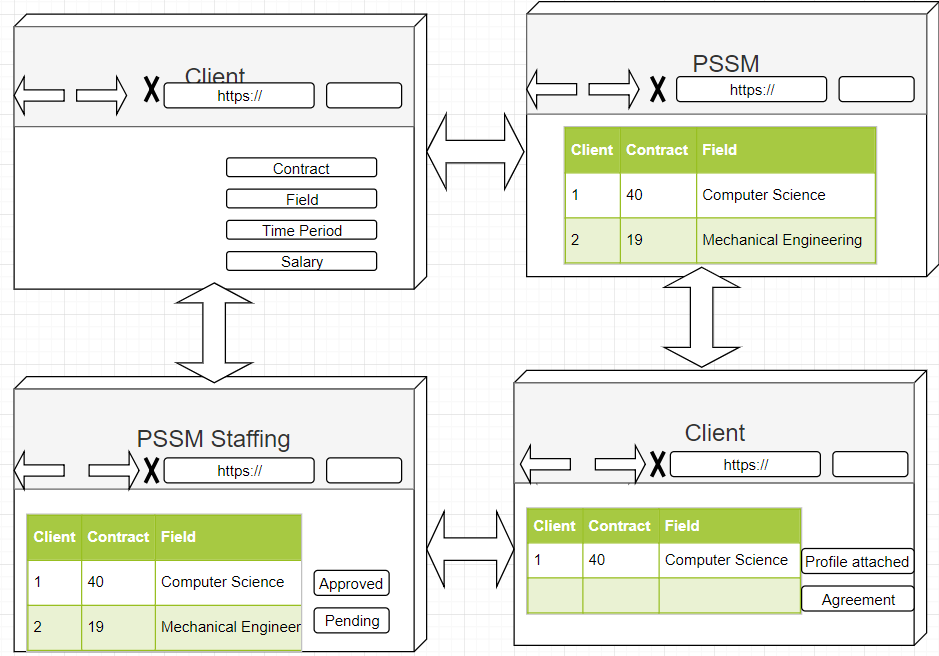
|  |  |
| --- | --- |
| Use Scenario: Client generates a staffing request to hire a new employee | Use Scenario: Candidates are shortlisted and verified according to the contract |
| 1. Client creates a StaffingRequest to hire a new employee  2. Contract Manager verifies the contract details with the client and approves the request for further process.  3. If the contract is not valid, the request is returned to the client along with the memo for the reason of decline.  4. The validated requests are inserted in the database for the placement department to start looking for eligible candidates and interview them | 1. The validated requests are stored in the database.  2. Based on the contract, the candidate’s qualification, experience, and other details are checked  3. Shortlisted candidates are interviewed by the placement department  4. Selected candidates are marked ‘reserved’ and the arrangement department contacts them and agrees to the contract.  5. The placed candidate detail is sent to the client |

**Windows navigation diagram**

****



**Windows Layout Diagrams for the user interface**



How would your user interface design have to be modified if you were to deploy it on a tablet? What about a smartphone?

For user interface design to deploy in a tablet or smartphone, the important point is to consider space limitation, the kind of input (touch/manual key), the target audience, in this case, clients and PSSM employee’s needs are to evaluate for a tablet or smartphone version, with smart devices capacities like GPS and accelerometers, the functioning of an interface can be enhanced. One of the major implications is the screen orientation, which is scrollable vertically, with touchscreen devices, factors like tapping, pinching, spreading, flicking and scrolling should be considered in designing.

What, if any, social media sites should PSSM con­sider?

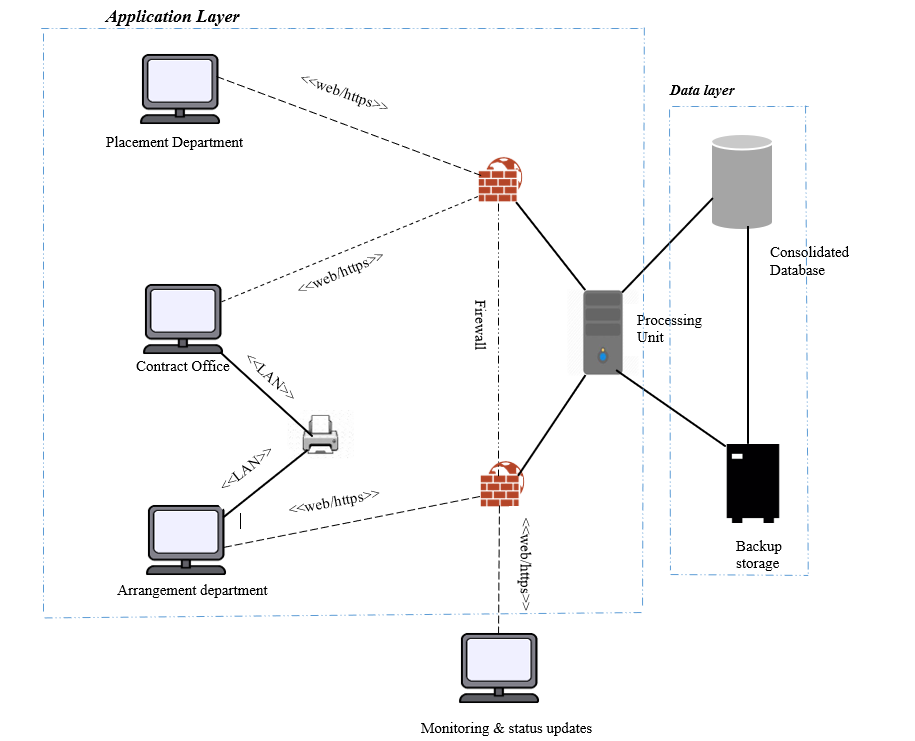
PSSM can consider Facebook and Twitter for their branding, marketing, and even operations, they can differentiate and innovate with linking LinkedIn profiles of all the candidates in the database along with their application, company details can also be added here to help the candidate learn about the company through their website and LinkedIn page.

How would your answers change if you were developing the system for a global audience?

PSSM would serve global clients and candidates, for this basic requirement like multilingual capabilities, cultural and sociological factors will be considered. For a global reach, most of the interface design would remain the same, just the target audience will change as soon as the user allows the user detection or applies the filter themselves.

Physical Architecture Layer Design

**Deployment Diagram**

****

PSSM consolidated Database

PSSM consolidated Server

Client Interface

<<HCI Layer>>

PSSM System

<<Data Management Layer>>

PSSM System

<<Problem Domain Layer>>

PSSM System

<<TCP/IP>> <<TCP/IP>>

Conversion and Test Plan

**Unit Testing**

|  |
| --- |
| *Test case #:* 1 *Test Case Name:* Reserve, place, close a Staffing Request.  *Designed by:* Cosmo Kramer *Design Date: 12/01/2018*  *Executed by:* Jerry Seinfeld *Execution Date: 12/01/2018*  *Short Description:* Test the candidate reservation, employee placement, and close functions of a Staffing Request. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | *Action* | *Expected System Response* | *Pass/*  *Fail* | *Comment* |
| 1 | Test Reserve: Click the “Reserve Candidate” button | The system presents a list of Employee as well as Staffing Request search input fields: Employee ID, Request ID, Employee Status, Request Status, ContractID, Employee Name, Employee Status, Expected Pay, Pay Offered, etc. |  |  |
| 2 | Fill in desired search fields, e.g. Employee Name or Request ID | The system displays a list of items that match. |  |  |
| 3 | Select the desired Staffing Request and click “Candidates” | A list of available candidates is presented. |  |  |
| 4 | Select a suitable candidate and click Reserve | A message is displayed: “Candidate <EmployeeID> has been reserved for Staffing Request <RequestID>.” Arrangements department has been notified. |  |  |
| 5 | *Check post-condition 1.* |  |  |  |
| 6 | *Repeat Test Reserve (1, 2) as described above.* |  |  |  |
| 7 | Select the desired Candidate and click “Staffing Requests” | A list of open Staffing Requests is presented. |  |  |
| 8 | Select an appropriate Staffing Request and click “Reserve” | A message is displayed: “Candidate <EmployeeID> has been reserved for Staffing Request <RequestID>. Arrangements department has been notified.” |  |  |
| 9 | *Check post-condition 2.* |  |  |  |
| 10 | Click the “Edit” button | The Staffing Request fields become editable, except for the Request ID field, which can’t be changed. |  |  |
| 11 | Enter a change for the Pay Offer field. Click Save. | The system presents a pop-up message “Staffing Request #<RequestID>” has been saved. |  |  |
| 12 | *Check post-condition 2.* |  |  |  |
| 13 | Test Place: From the home screen, click “Place Employee” | A list of reserved candidates is listed. |  |  |
| 14 | Select the desired candidate and click “Staffing Requests” | The system displays a list of open Staffing Requests. |  |  |
| 15 | Select an appropriate Staffing Request and click the “Place” button. | A message is displayed: “Employee <EmployeeID> has been placed for Staffing Request <RequestID>, Contract <ContractID>.” The Contract Manager is notified |  |  |
| 16 | *Check post-condition 3.* |  |  |  |
| 17 | From the home screen, select “Search Staffing Requests” | Several Staffing Request search fields are presented. |  |  |
| 18 | Enter the Request ID for the placed Staffing Request from step 15 above. | The Staffing Request is displayed. |  |  |
| 19 | Select the Staffing Request and choose “Close Request” | A message appears, “Staffing Request <RequestID> was filled and is now closed. The Client has been notified.” |  |  |
|  | *Check post-condition 4.* |  |  |  |

|  |
| --- |
| *Post-conditions:*  1. The Candidate is marked as “reserved” in the database, and the associated Staffing Request now includes the EmployeeID in the ReservedEmployee field. A notification email regarding the candidate reservation was received by Arrangements.  2. The Candidate is marked as “reserved” in the database, and the associated Staffing Request now includes the EmployeeID in the ReservedEmployee field. A notification email regarding the candidate reservation was received by Arrangements.  3. The Candidate, Staffing Request, and Contract records have been updated in the database. The billing department was notified to generate a bill for the client. A copy of the Staffing Request was emailed to the Client.  4. A notification email was sent to the Client indicating that the Staffing Request was closed. |

|  |
| --- |
| *Test case #:* 2 *Test Case Name:* Candidate  *Designed by:* George Costanza *Design Date: 12/01/2018*  *Executed by:* Jerry Seinfeld *Execution Date: 12/01/2018*  *Short Description:* Test the Creation, Reading, Updating, and Deleting of a client Contract |

|  |
| --- |
| *Pre-conditions:*  The Customer has established a new relationship with a staff candidate.  The Customer has logged into the web application. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | *Action* | *Expected System Response* | *Pass/*  *Fail* | *Comment* |
| 1 | Test Create: Click the “Create Employee” button | The system presents a list of input fields. Fields surrounded by a red box are required. |  |  |
| 2 | Fill in required Employee fields: ID, Name, Address, Phone, Status, Years of Experience, Degree, and Expected Pay | The system enables the Save button at the bottom of the input fields |  |  |
| 3 | Click Save. | The system presents a pop-up message “Employee #<EmployeeID>” has been saved. The list of existing contracts is displayed. |  |  |
| 4 | *Check post-condition 1.* |  |  |  |
| 5 | Test Read: Click the “Find Employee” button | The system presents a list of Employee search input fields: ID, Name, Phone, Status, Years of Experience, Degree, Expected Pay |  |  |
| 6 | Type the desired Employee ID into the “ID” field. Click “Search” | The system presents a list of matching employees. |  |  |
| 7 | Select the desired Employee and click the “Open” button | The system displays the Employee details. The fields are read-only. |  |  |
| 8 | *Repeat Test Read (5) for Name, Phone, Status, Years of Experience, Degree, and Expected Pay* |  |  |  |
| 9 | Test Update: Open an Employee as described in Test Read (5) above. | The system displays the Employee details. The fields are read-only. |  |  |
| 10 | Click the “Edit” button | The Employee fields become editable, except for the Contract ID and Name fields, which can’t be changed. |  |  |
| 11 | Enter a change for the Phone field. Click Save. | The system presents a pop-up message “Employee #<EmployeeID>” has been saved. |  |  |
| 12 | *Check post-condition 2.* |  |  |  |
| 13 | *Repeat Test Update (9) for Name, Address, Phone, Status, Years of Experience, Degree, and Expected Pay* |  |  |  |
| 14 | Test Delete: Open an Employee as described in Test Read (5) above. | The system displays the Employee details. The fields are read-only. |  |  |
| 15 | Click the “Edit” button | The Employee fields become editable, except for the Employee ID and Name fields, which can’t be changed. |  |  |
| 16 | Click the “Delete Employee” button | The system displays a pop-up message “Are you sure you want to delete Employee #<EmployeeID>? You cannot undo this action.” Two buttons are presented below: “Cancel” and “Delete” |  |  |
| 17 | Click Delete. | The system displays a message, “Employee #<EmployeeID> has been deleted.” |  |  |
| 18 | *Check post-condition 3.* |  |  |  |

|  |
| --- |
| *Post-conditions:*  1. The new Candidate is saved in the database with the entered data.  2. The new Candidate is saved in the database with the updated data.  3. The Candidate has been removed from the database. |

**Use case testing**

|  |
| --- |
| *Test case #:* 1 *Test Case Name:* Fulfill a new staffingRequest  *Designed by:* Tim Whatley *Design Date: 12/01/2018*  *Executed by:* Elaine Benes *Execution Date: 12/01/2018*  *Short Description:* Test the options when a staffing request is initiated. |

|  |
| --- |
| *Pre-conditions:*  The user has opened a web browser. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | *Action* | *Expected System Response* | *Pass/*  *Fail* | *Comment* |
| 1 | Enter the new system URL in the browser. | The System displays theweb page. The user is prompted for a username and password. |  |  |
| 2 | Enter the assigned username and password. Click the “Login” button. | The System displays the home screen. |  |  |
| 3 | At the home screen, click Logout. | The user is prompted for a username and password. |  |  |
| 4 | Below the login fields, click the “Forgot password?” link. | The System displays an input field labeled, “Please enter your email address to reset your password” |  |  |
| 5 | Enter the new email address. Click “Reset my password”. | The System sends an email to the user. |  |  |
| 6 | The user checks their email and locates the email. The user clicks on the reset password URL included in the email. | The System opens a new browser window, connects to the application, and displays two input fields, for entering and confirming the new password. |  |  |
| 7 | Enter and confirm the new password. Click Save. | A message is displayed, “Your password has been successfully reset. You may now log into the system. |  |  |
| 8 | *Check post-condition 1.* |  |  |  |

|  |
| --- |
| *Post-conditions:*  1. The user can log in with the newly-reset password. |

|  |
| --- |
| *Test case #:* 2 *Test Case Name:* Contract fulfillment  *Designed by:* Cosmo Kramer *Design Date: 12/01/2018*  *Executed by:* George Costanza *Execution Date: 12/01/2018*  *Short Description:* Test the Creation, Reading, Updating, and Deleting of a client Contract |

|  |
| --- |
| *Pre-conditions:*  The Customer has established a contract with a client.  The Customer has logged into the  web application. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Step* | *Action* | *Expected System Response* | *Pass/*  *Fail* | *Comment* |
| 1 | Test Create: Click the “Create Contract” button | The system presents a list of input fields. Fields surrounded by a red box are required. |  |  |
| 2 | Fill in required Contract fields: ID, Expiration Date, Company Name, Address, Phone, Terms | The system enables the Save button at the bottom of the input fields |  |  |
| 3 | Click Save. | The system presents a pop-up message “Contract #<ContractID>” has been saved. The list of existing contracts is displayed. |  |  |
| 4 | *Check post-condition 1.* |  |  |  |
| 5 | Test Read: Click the “Find Contract” button | The system presents a list of Contract search input fields: ID, Expiration Date, Name, Phone |  |  |
| 6 | Type the desired Contract ID into the “ID” field. Click “Search” | The system presents a list of matching contracts. |  |  |
| 7 | Select the desired Contract and click the “Open” button | The system displays the Contract details. The fields are read-only. |  |  |
| 8 | *Repeat Test Read (5) for Expiration Date, Name, and Phone.* |  |  |  |
| 9 | Test Update: Open a Contract as described in Test Read (5) above. | The system displays the Contract details. The fields are read-only. |  |  |
| 10 | Click the “Edit” button | The Contract fields become editable, except for the Contract ID field, which can’t be changed. |  |  |
| 11 | Enter a change for the Phone field. Click Save. | The system presents a pop-up message “Contract #<ContractID>” has been saved. |  |  |
| 12 | *Check post-condition 2.* |  |  |  |
| 13 | *Repeat Test Update (9) for Expiration Date, Name, and Phone.* |  |  |  |
| 14 | Test Delete: Open a Contract as described in Test Read (5) above. | The system displays the Contract details. The fields are read-only. |  |  |
| 15 | Click the “Edit” button | The Contract fields become editable, except for the Contract ID field, which can’t be changed. |  |  |
| 16 | Click the “Delete Contract” button | The system displays a pop-up message “Are you sure you want to delete Contract #<ContractID>? You cannot undo this action.” Two buttons are presented below: “Cancel” and “Delete” |  |  |
| 17 | Click Delete. | The system displays a message, “Contract #<ContractID> has been deleted.” |  |  |
| 18 | *Check post-condition 3.* |  |  |  |

|  |
| --- |
| *Post-conditions:*  1. The new Contract is saved in the database with the entered data.  2. The updated Contract is saved in the database with the updated data.  3. The Contract has been removed from the database. |

**User Acceptance Testing**

Acceptance training consists of two types,

* Alpha testing – It is conducted by the users to ensure that they accept the system, the system tests are the test plan source and it is used for normal acceptance testing. Alpha testing can be repeated but are done by users to accept the system. With PSSM, the Client and PSSM staff will have to conduct alpha testing to understand the system’s working and its interaction with the common database system. By using the test data, it is determined if the solution meets the customer requirements. After each class and process, alpha testing is conducted. If an Agile method is performed, the customer can use it as soon as possible and get feedback for improvement or changes.
* Beta testing – It uses real data instead of test data, the system requirement in this case is the test plan source. In beta testing, the users closely monitor the system for errors or improvements. This is also be conducted by clients and PSSM staff.

**Implementation**

For a smooth transition from the existing PSSM system to the new improved system, a general change management system can be introduced which would focus on keeping up the ‘As-is’ system and transitioning to the new system, it can be done by ‘Unfreeze, Change, Refreeze’ method, this will facilitate any initial feedbacks or suggestions and doubts related to the transition to the new system. With the change, the system will be well accustomed to it as the unfreezing was conveyed and agreed upon. Lastly, with refreeze, all the proposed changes will be included in the current system, the refreezing process gives an opportunity for the staff and clients to be trained in the new system.

With their current system, PSSM does not have any standard operating procedure, the new system will provide faster processing of the request, ability to add new clients and candidates to the consolidated database system, create timely reports to check the status of placed, waited and unplaced candidates, with data analytics capabilities, the system will encourage PSSM to get more candidates profile based on the staffing request and contract. With live dashboards and reports, client information can be leveraged, and most profitable and least profitable clients can be identified.

With this new system, security of data will be enhanced compared to earlier manual requests and risk of losing data, tasks will be better coordinated, and all the departments will be able to keep a track of the existing and past requests in real time.  
With maintaining a database for clients and candidates, the cost of this datacenter can be considered a long-time investment, as CRM portals and Dashboards provide analytical capabilities, chances of getting ROI is higher than the investment.

One of the main and important aspect of this new system will be training for the staff on operations and maintenance. High quality design documents and user manuals would be provided for all the steps in the process, this will result in the increase in efficiency and productivity of PSSM.