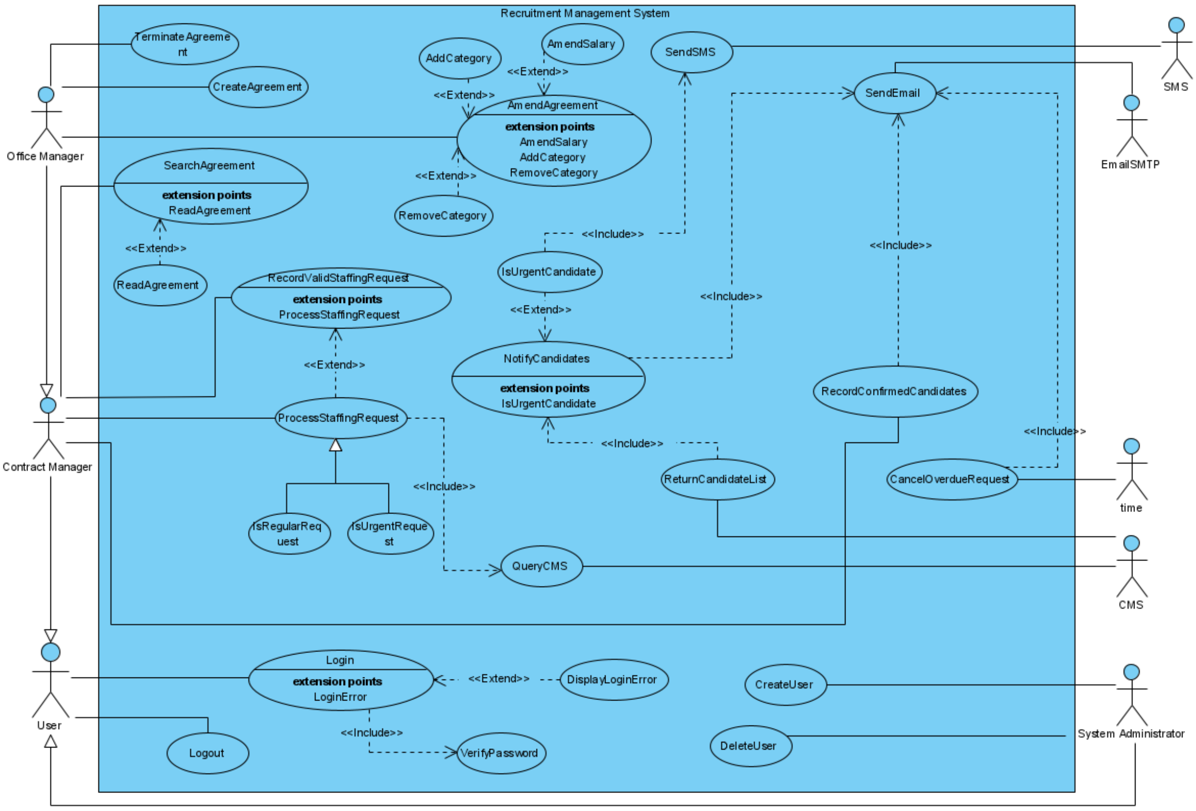
Object Oriented Analysis and Design

1)

<table>
<tr><td colspan="2"><b>Requirement ID: 1</b></td><td colspan="2"><b>Requirement Type: FR</b></td><td colspan="2"><b>Event/Use Case #1</b></td></tr>
<tr><td colspan="6"><b>Description:</b> The system will allow approved requests to be recorded onto the RMS with details of the client.</td></tr>
<tr><td colspan="6"><b>Rationale:</b> The system will be able query CMS and find jobs most suitable for the client’s skill set.</td></tr>
<tr><td colspan="6"><b>Source:</b> The initial statement of requirements of the Recruitment Management Software.</td></tr>
<tr><td colspan="6"><b>Fit Criteria:</b> The system shall correctly record each approved clients’ staffing requests including type of employee, job description, key skills, the starting date and the duration of employment with the salary for the vacant position.</td></tr>
<tr><td colspan="3"><b>Customer Satisfaction: 5</b></td><td colspan="3"><b>Customer Dissatisfaction: N/A</b></td></tr>
<tr><td colspan="3"><b>Priority:</b> Essential</td><td colspan="3"><b>Conflicts:</b> None</td></tr>
<tr><td colspan="4"><b>Supporting Material:</b> The initial statement of requirements of the Recruitment Management Software.</td><td colspan="2" rowspan="2">Volere
Source: Atlantic Systems Guild</td></tr>
<tr><td colspan="4"><b>History:</b> New Requirement</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Requirement ID: 2</b></td><td colspan="2"><b>Requirement Type: NFR</b></td><td colspan="2"><b>Event/Use Case #1</b></td></tr>
<tr><td colspan="6"><b>Description:</b> Clients’ details shall be kept secure and unshared.</td></tr>
<tr><td colspan="6"><b>Rationale:</b> The Data Protection Act states that data must “handled in a way that ensures appropriate security, including protection against unlawful or unauthorised processing, access, loss, destruction or damage”.</td></tr>
<tr><td colspan="6"><b>Source:</b> Data Protection Act 2018</td></tr>
<tr><td colspan="6"><b>Fit Criteria</b>: The development and use of the system is compliant with ISO 27001/ISO27005</td></tr>
<tr><td colspan="3"><b>Customer Satisfaction: 5</b></td><td colspan="3"><b>Customer Dissatisfaction: N/A</b></td></tr>
<tr><td colspan="3"><b>Priority:</b> Highly Important</td><td colspan="3"><b>Conflicts:</b> None</td></tr>
<tr><td colspan="4"><b>Supporting Material:</b> Data Protection Act 2018; https://www.gov.uk/data-protection</td><td colspan="2" rowspan="2">Volere
Source: Atlantic Systems Guild</td></tr>
<tr><td colspan="4"><b>History:</b> New Requirement</td></tr>
</table>

2)



3)

ID: UC11	**Use case:** ProcessStaffingRequest (ProcessStaffingOrder)

Brief description:

The contract manager processes staffing request which can either be regular or urgent. The system will query CMS to find suitable candidates based upon the clients needs in the request.

Primary actors:

Office Manager and Contract Manager

Secondary actors:

None (some of the extensions will have secondary actors)

Preconditions:

1. RMS is operational, the Contract Manager or Office Manager has logged in.
2. Client must have sent a staffing request.
3. Staffing request has been validated by a Contract Manager.

Flow of events:

1. The contract manager records the validated staffing requests which includes the job description, key skills required for the job, start date, the duration of

employment and the salary. 2. The staffing request can either be urgent in which candidates have 3 days to respond or a regular request in which candidates have 7 days to respond. 3. CMS is then queried, include(queryCMS)
Postconditions: 1.The system stores data of the staffing requests.
Alternative flow: None

ID: UC12	**Use case:** QueryCMS
Brief description: CMS is queried by RMS to find suitable candidates given the clients requirements.	
Primary actors: None	
Secondary actors: CMS	
Preconditions: 1. RMS is operational 2. Staffing request must be stored in RMS.	
Flow of events: 1. RMS will query the CMS. 2. Key skills in the staffing request are matched to that of the candidates. 3. Once the query is complete CMS returns a list of candidates to the RMS. 4. RMS will order the list of candidates in order of relevance and begin notifying candidates.	
Postconditions: 1. An ordered list is stored onto RMS.	
Alternative flow: EmptyCandidateList	

Alternative flow: QueryCMS : EmptyCandidateList
ID: UC12.1
Brief description: CMS returns a list without candidates to RMS.
Primary actors: None

Secondary actors: CMS

Preconditions:

1. RMS is operational
2. CMS is operational and has access to RMS
3. Staffing request must be stored in RMS.

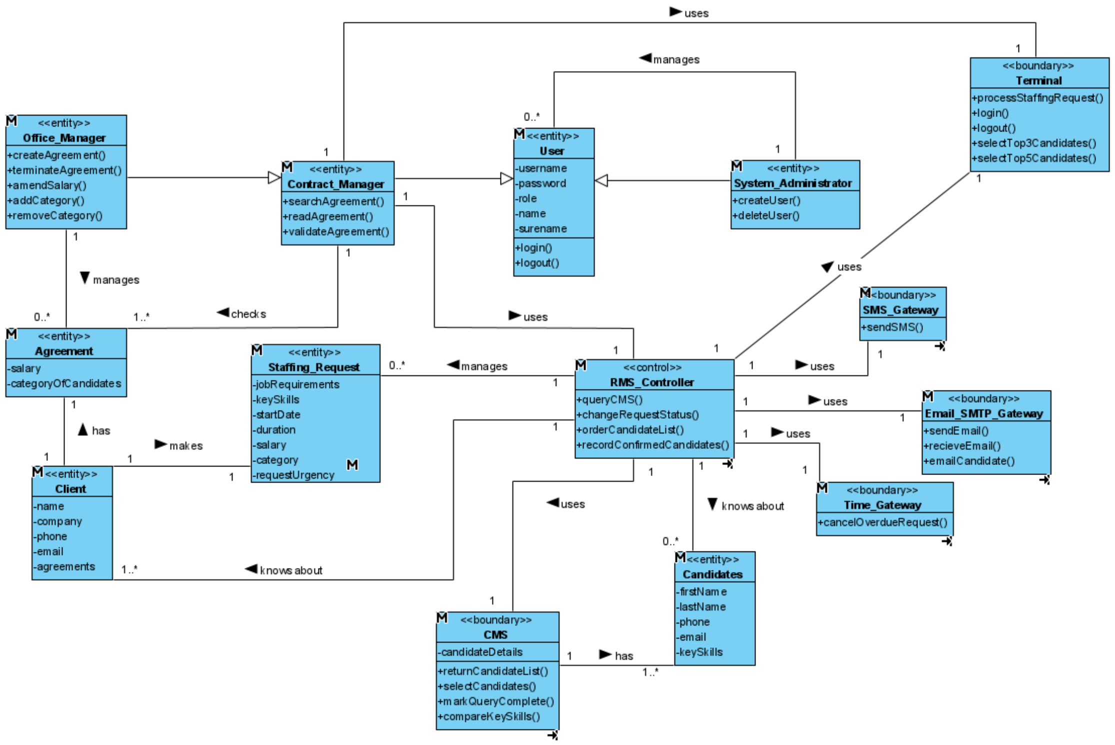
Alternative flow:

2. CMS does not find any suitable candidates using the requirements in the staffing request.
3. CMS returns an empty candidate list.
4. RMS marks the staffing request as failed.
5. RMS uses Email SMTP to send email notifying the client of the failed request.

Postconditions:

1. Empty candidate list is returned.
2. Staffing request is marked as failed.

4a)



4b.1)

VERBS

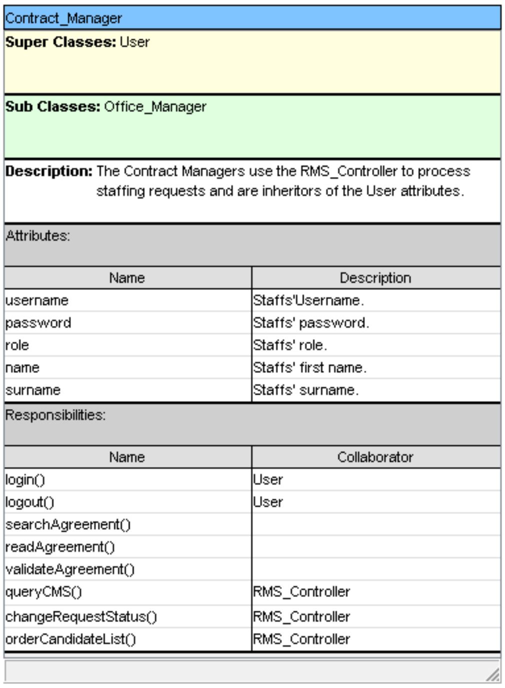
NOUNS

“Each client would have an agreement with RMA for the provision of personnel on short term contracts. An agreement is drawn up between RMA and the respective client defining the categories of personnel, which RMA will be providing to the client and the range of monthly salaries for each category. Each agreement is recorded in RMS and may be subsequently amended by the Office Manager at RMA. The amendments may involve adding or removing categories of staff to the agreement or amending the range of monthly salaries for the existing categories of personnel. The Agreement is valid until it is terminated by the Office Manager.”

NOUN	CLASS?	ATTRIBUTE?	NOT MODELLED
Client	✓		
Agreement	✓		
RMA	✓		
Provision			✓
Personnel			✓
Term			✓
Contracts		✓	
Categories		✓	
Range			✓
Salaries		✓	
Category		✓	
Office Manager	✓		
Amendments		✓	
Staff	✓		

VERB	OPERATION	ASSOCIATION	NOT MODELLED
Would have		✓	
Is Drawn		✓	
Defining	✓		
Will Be Providing			✓
Recorded	✓		
May be			✓
Amended/Amending	✓		
Involve	✓		
Adding	✓		
Removing	✓		
Existing			✓
Terminated	✓		

4b.2)



Contract_Manager

Super Classes: User

Sub Classes: Office_Manager

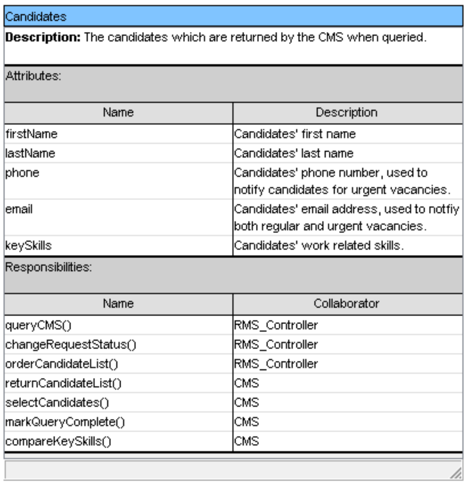
Description: The Contract Managers use the RMS_Controller to process staffing requests and are inheritors of the User attributes.

Attributes:

Name	Description
username	Staffs'Username.
password	Staffs' password.
role	Staffs' role.
name	Staffs' first name.
surname	Staffs' surname.

Responsibilities:

Name	Collaborator
login()	User
logout()	User
searchAgreement()	
readAgreement()	
validateAgreement()	
queryCMS()	RMS_Controller
changeRequestStatus()	RMS_Controller
orderCandidateList()	RMS_Controller



Candidates

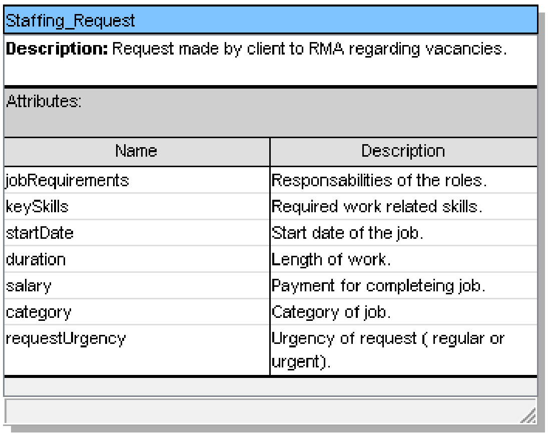
Description: The candidates which are returned by the CMS when queried.

Attributes:

Name	Description
firstName	Candidates' first name
lastName	Candidates' last name
phone	Candidates' phone number, used to notify candidates for urgent vacancies.
email	Candidates' email address, used to notfiy both regular and urgent vacancies.
keySkills	Candidates' work related skills.

Responsibilities:

Name	Collaborator
queryCMS()	RMS_Controller
changeRequestStatus()	RMS_Controller
orderCandidateList()	RMS_Controller
returnCandidateList()	CMS
selectCandidates()	CMS
markQueryComplete()	CMS
compareKeySkills()	CMS



Staffing_Request

Description: Request made by client to RMA regarding vacancies.

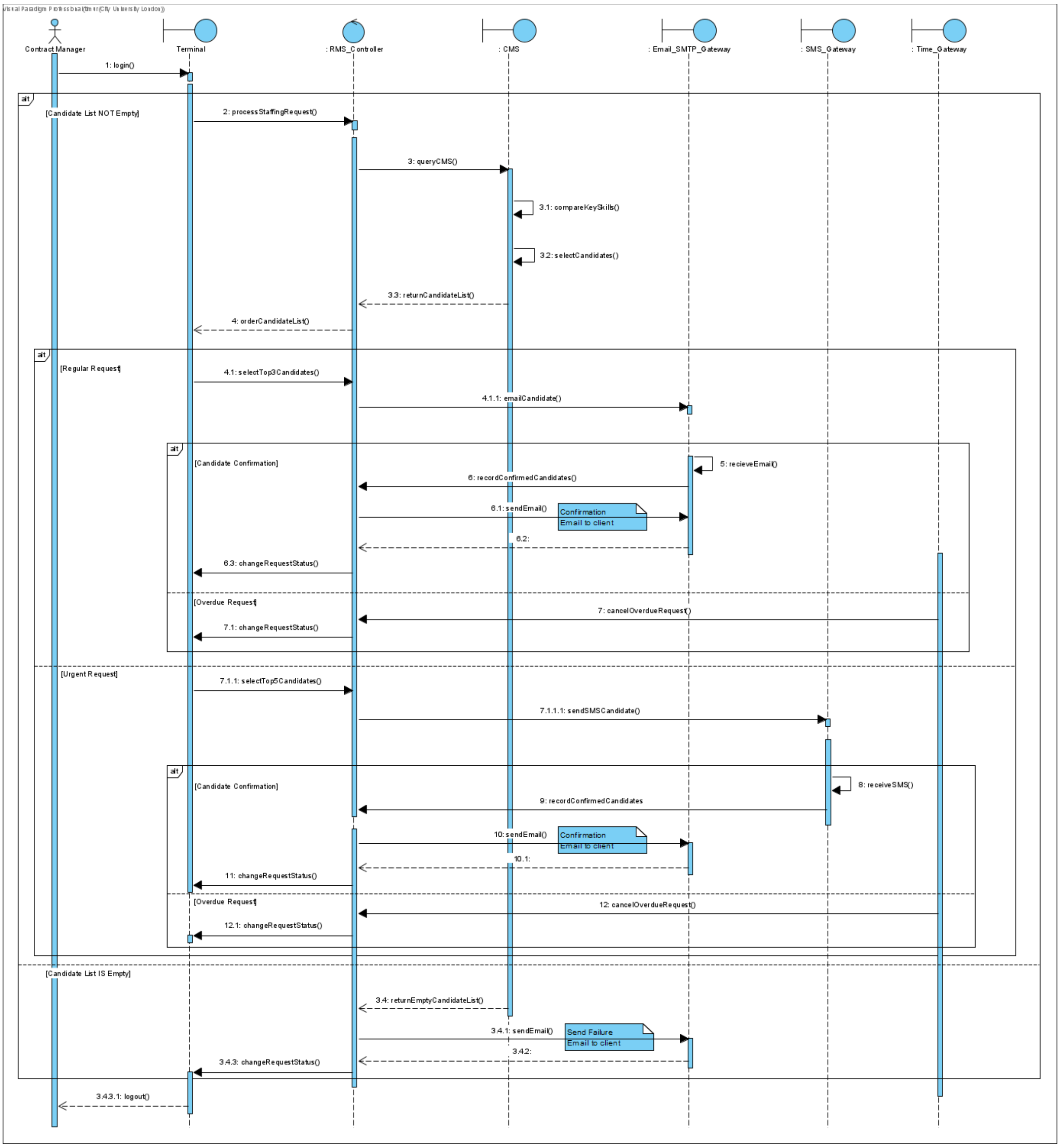
Attributes:

Name	Description
jobRequirements	Responsabilities of the roles.
keySkills	Required work related skills.
startDate	Start date of the job.
duration	Length of work.
salary	Payment for completeing job.
category	Category of job.
requestUrgency	Urgency of request (regular or urgent).

4b.3)

Associations between classes with different stereotypes confirm the rules of robustness analysis. A reasonable number of boundry classes have been added so that each actor type (time_gateway, sms_gateway, email_smtp_gateway, and CMS) has its own boundry class. Each class was assigned a stereotype that indicates its role in robustness analysis.

5)


Contract Manager
Terminal
: RMS_Controller
: CMS
: Email_SMTP_Gateway
: SMS_Gateway
: Time_Gateway
1: login()
alt
[Candidate List NOT Empty]
2: processStaffingRequest()
3: queryCMS()
3.1: compareKeySkills()
3.2: selectCandidates()
3.3: returnCandidateList()
4: orderCandidateList()
alt
[Regular Request]
4.1: selectTop3Candidates()
4.1.1: emailCandidate()
alt
[Candidate Confirmation]
5: recieveEmail()
6: recordConfirmedCandidates()
6.1: sendEmail()
Confirmation Email to client
6.2:
6.3: changeRequestStatus()
[Overdue Request]
7: cancelOverdueRequest()
7.1: changeRequestStatus()
[Urgent Request]
7.1.1: selectTop5Candidates()
7.1.1.1: sendSMSCandidate()
alt
[Candidate Confirmation]
8: receiveSMS()
9: recordConfirmedCandidates
10: sendEmail()
Confirmation Email to client
10.1:
11: changeRequestStatus()
[Overdue Request]
12: cancelOverdueRequest()
12.1: changeRequestStatus()
[Candidate List IS Empty]
3.4: returnEmptyCandidateList()
3.4.1: sendEmail()
Send Failure Email to client
3.4.2:
3.4.3: changeRequestStatus()
3.4.3.1: logout()