NOTTINGHAM TRENT UNIVERSITY

College of Arts and Science
School of Science and Technology

ISYS10221: ISAD

SYSTEMS DEVELOPMENT PORTFOLIO

By

Group I4
N0305434: Robert Shippey
N0329708: Ian Jayes
N0362030: Justin Nsiko-Bankwa

JJTC Baby Care Ltd
Employee Recruitment & Development Project

Seminar Tutor: Ahmad Al-Shami

Table of Contents

I. Team Structure & Development Strategy 3

II. Team Concept Map 5

III. Project Work Plan 6
Work Breakdown Structure 6
Gantt Chart 7

IV. System Proposal 8

V. Evaluation 19

Appendix A. Minutes of Meetings 21

Appendix B. Stages of Normalisation 24

Appendix C. Detailed Use-Case Tables 26

I. Team Structure & Development Strategy

The Belbin's self-perception exercise is a test that each group member completes to find out the type of role the person will be best suited when working in a group. The exercise consists of 7 sections, each section has 8 statements that the person has to give a score and each section has to add up to 10. The highest statement mark will identify what the group member is best at. Every team member in our group (Ian, Robert and Justin) completed the Belbin's self-perception exercise and below is a summary of the outcome.

Justin:

The results came through as a College Worker. This means he is conservative, dutiful and predictable. He has the organisational skills with a practical mind and common sense. These add up to make him a hard working disciplined team member. The downside is that he will struggle seeing new ideas. He scored two marks for the statement "I find it difficult to get started unless the goals are clear" which proves he needs a leader or guidance to help complete a given task.

Robert:

The results came through as a Team Worker. This means he gets on with everyone, although can be sensitive to any comments given to him. He has the ability to respond to people and to situations when things are looking grim. Another positive point is that he has the ability to promote team spirit. A negative point is that he can be indecisive in a moment of crisis. He scored himself 4 marks for this statement "I find it difficult to lead from the front, perhaps because I am over-responsive to group atmosphere." To clarify, he means he feels he can't be an active leader because he's very sensitive to the group's atmosphere. He scored himself 1 for the following statement: "I would be ready to work with the person who showed the most positive approach." He obviously believes that he won't just work with the person who shows the most positive approach; he will instead work with everyone.

Ian:

The results came through as a Team Worker and College Worker. This means he's a bit of both as he "finds it difficult to get started unless the goals are clear" (college worker) and he has "a quiet interest in getting to know colleagues better" (team worker). All in all, Ian is practical, has common sense and has the ability to response to other group members.

In conclusion, we all had a similar result of Team Worker and College Worker. This shows that we're all hard working and should work well together. We do not have anyone who has been identified as a chair but we will try hard to each take a role of leading the group.

The waterfall model is a suitable development methodology tool to use as its widely used in most software development projects. The waterfall model is split up into 6 stages normally listed below:

Requirement Phase: this includes all the user requirements, system requirements and any specifications that the final product will need.

Specification Phase: this phase is where the exact requirements and specifications

are put together to output the way the system should turn out to be.
Design Phase: this phase is where the visual and the logical design are planned. Visual designs would be the front end where the end user will see and the logical designs are the databases and data storage plans.
Testing & Implementing Phase: this phase is where the developer tests the system, checks to see if there are any flaws or error and makes sure the system meets all the specifications. Once it's been tested then it's ready to be implemented.
Maintenance Phase: this phase is where the developer keeps the system updated with the changes made in the specification and provides support to the company if any errors happen.

As system developers we will use the Waterfall model, as it is easy to understand and follow, it makes documenting easy as the developer can use each phase as a template. Each phase relies on the previous phase so it's a good method to follow when system developing. The waterfall method is better for longer project that need to be highly accurate, which our project is. The waterfall method is also useful for developing systems that use familiar technology and project with clear requirements.

Divide and conquer is a simple and easy method to use to find when trying to identify the root causes and effects of a problem within a system. Fish bone diagrams are very basic and doesn't really help fix a problem. All it does is just shows the programmer the map of the program where as the divide and conquer method allows the programmer to break down the software and look at each section and solve one section at a time. This is a good way of testing the software as well.

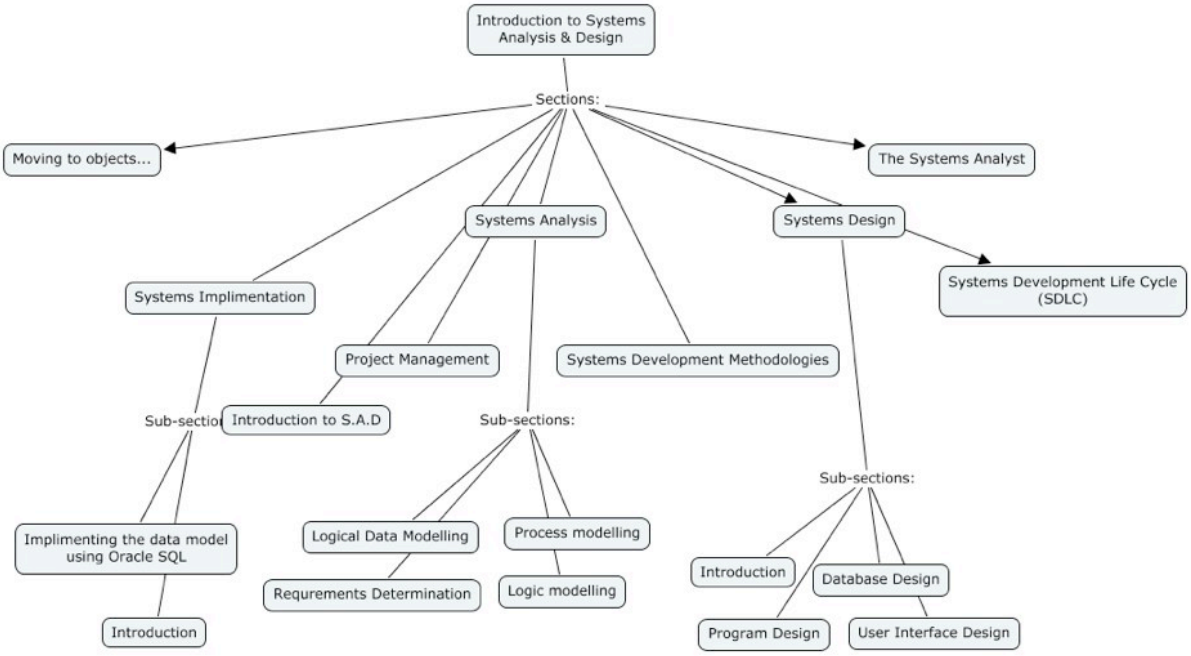
If an error occurs with a system, the system developer may investiage the problem by looking at the program and breaking it down into local sections which can be tested seperately to find out where the issue lies. For example, the developer may find that queries are giving an incorrect result. A logical way to assess this problem, using the divide and conqour method, would be to take the queriers, tables, relationships, and the data it's self as separate parts of the system. Then each can be tested indiidually.

II. Team Concept Map

A concept map is a diagram that is used to better understand and share knowledge. It breaks down a concept into hierarchical nodes and associates them with linking words. These nodes can contain text, images, and web links to further demonstrate understanding of the concept. An arrow that shows the direction of information accompanies the linking words. These linked arrows show the relationship between information and are intended to read like a sentence.

We used concept maps to illustrate our understanding of the ISAD module. Our initial concept map was very basic and had a small amount of nodes. We had limited knowledge of the subject and so out concept map was limited. After working further with the software, and gaining more knowledge of ISAD from the lectures we were able to make improvements to the concept map. We added new major topic nodes and elaborated existing nodes further to show new understanding.

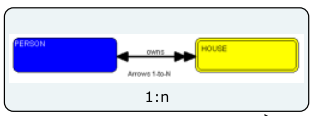
Our concept map from Review point 1:



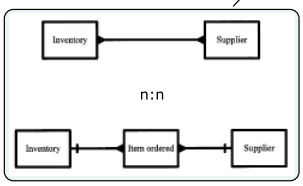
Final concept map:

Systems Analysis and Design

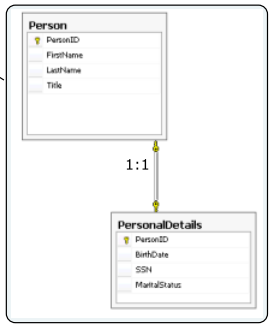
- How it's implimented
  - Systems Development Methodologies
    - Traditional
      - **Waterfall**
      - **V-model**
      - **Paralell**
      - **Spiral**
    - Modern
      - **Agile**
      - **RAD**
- What's Involved
  - Systems Development Lifecycle
    - Phases
      - Planning
      - Acceptance
      - Requirments
      - Analysis
        - consists of
          - Process modelling
            - is
              - Use cases
              - DFDs
                - levels
                  - concept
                  - level 1
                  - level 0
          - Logic modelling
            - is
              - decision tables
              - stuctured english
              - decision trees
          - Logical Data Modelling
            - is
              - data dictionaries
              - Entity Relationship Diagrams
                - relationships
                  - 1:n



                  - n:n



                  - 1:1



              - Functional Dependancy diagrams
              - Normalisation
                - contains
                  - 1st Normal Form
                  - 2nd Normal Form
                  - 3rd Normal Form
      - Design
        - Sub Sections:
          - Design Architecture
          - User Interface Design
          - Program Design
          - Database Design
            - techniques
              - CRUD Matrix
      - Implementation
        - stages are:
          - for updating System
          - for new Systems
      - Testing
        - types:
          - Black-Box Testing
          - White-Box Testing
      - Requirments
      - Phases
- Who's Involved
  - Stakeholders
    - are
      - someone who has an interest in a system
    - can be classified as
      - Users
      - Owners
      - Designers
      - Analysts
      - Programmers
      - IT Vendors and consultants

III. Project Work Plan

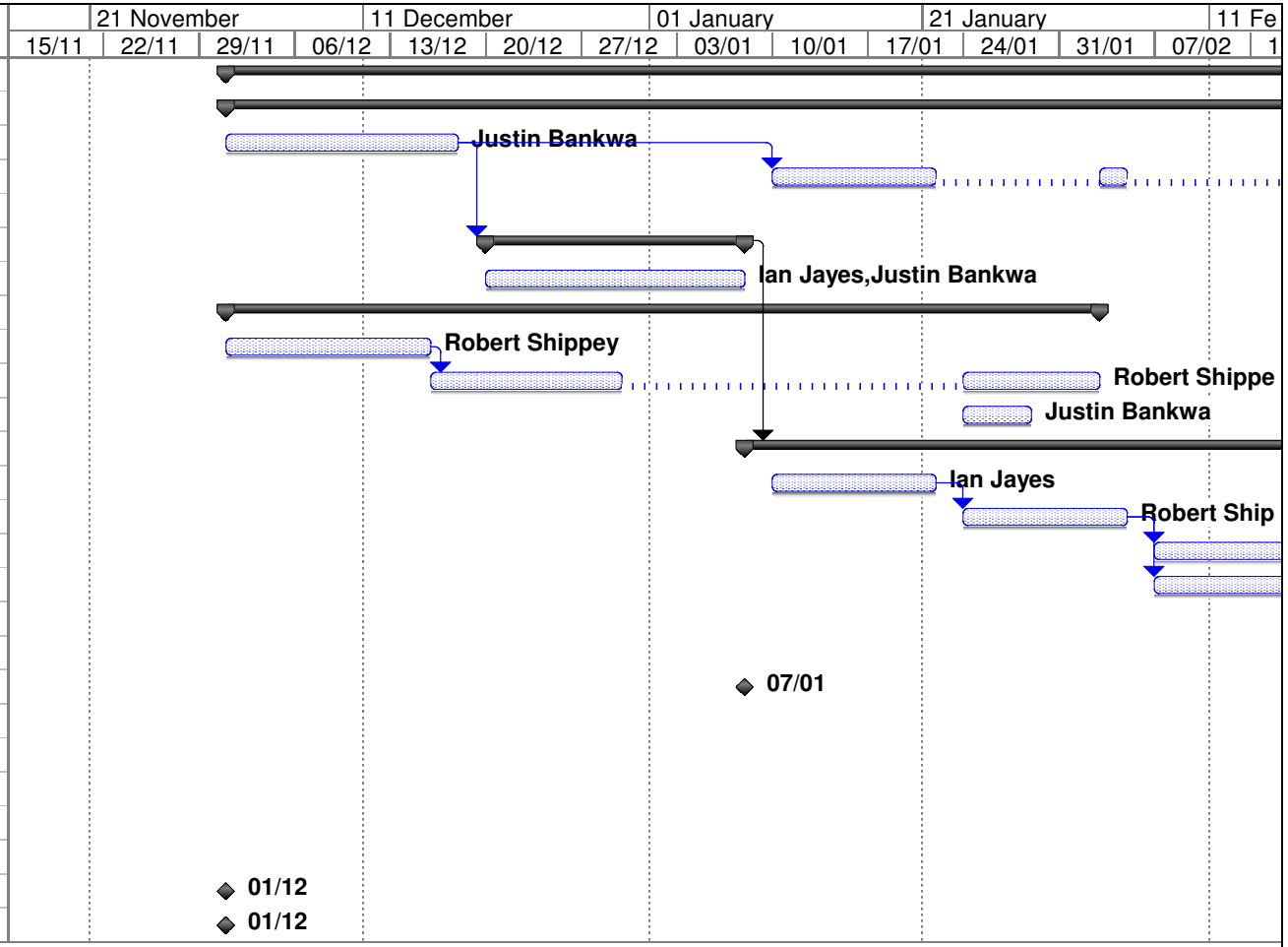
Work Breakdown Structure

Task Name	Duration	From	To	Assigned
ANALYSIS PHASE	112d	01/12/10	06/05/11	
Create requirements definition: • Determine requirements to track • Compile requirements as they are gathered • Review requirements with sponsor	32d: • 10d • 10d • 10d	01/12/10: • 01/12/10 • 20/12/10 • 03/01/11	14/01/11: • 17/12/10 • 31/12/10 • 14/01/11	Justin Justin, Ian Justin
Gather requirements: • Perform document analysis	15d: • 3w	20/12/10: • 20/12/10	07/01/11: • 07/01/11	Ian, Justin
Conduct interviews: • Interview project sponsor • Interview other project stakeholders • Observe retail store processes	20d: • 10d • 10d • 5d	01/12/10: • 01/12/10 • 16/12/10 • 01/12/10	29/12/10: • 15/12/10 • 29/12/10 • 08/12/10	Robert Ian, Robert Justin

Analyse current system: • Draw context diagram • Create use cases • Draw process models • Draw data models • Identify improvements • Conduct 3 JAD sessions • Analysis of current system is complete	45d: • 10d • 10d • 5d • 10d • 10d • 3w • 0d	07/01/11: • 10/01/11 • 24/01/11 • 07/02/11 • 07/02/11 • 21/02/11 • 21/02/11 • 07/01/11	11/03/11: • 21/01/11 • 04/02/11 • 11/02/11 • 18/02/11 • 04/03/11 • 11/03/11 • 07/01/11	 Ian Robert Justin, Ian Robert Ian, Justin
Develop concept for new system: • Draw context diagram • Create use cases • Draw process models • Draw data models	40d: • 10d • 3w • 1.5w • 3w	14/03/11: • 14/03/11 • 28/03/11 • 18/04/11 • 18/04/11	06/05/11: • 25/03/11 • 15/04/11 • 27/04/11 • 06/05/11	 Robert, Ian Robert, Robert, Robert, Ian, Justin
Design Phase	0d	01/12/10	01/12/10	
Implementation Phase	0d	01/12/10	01/12/10	

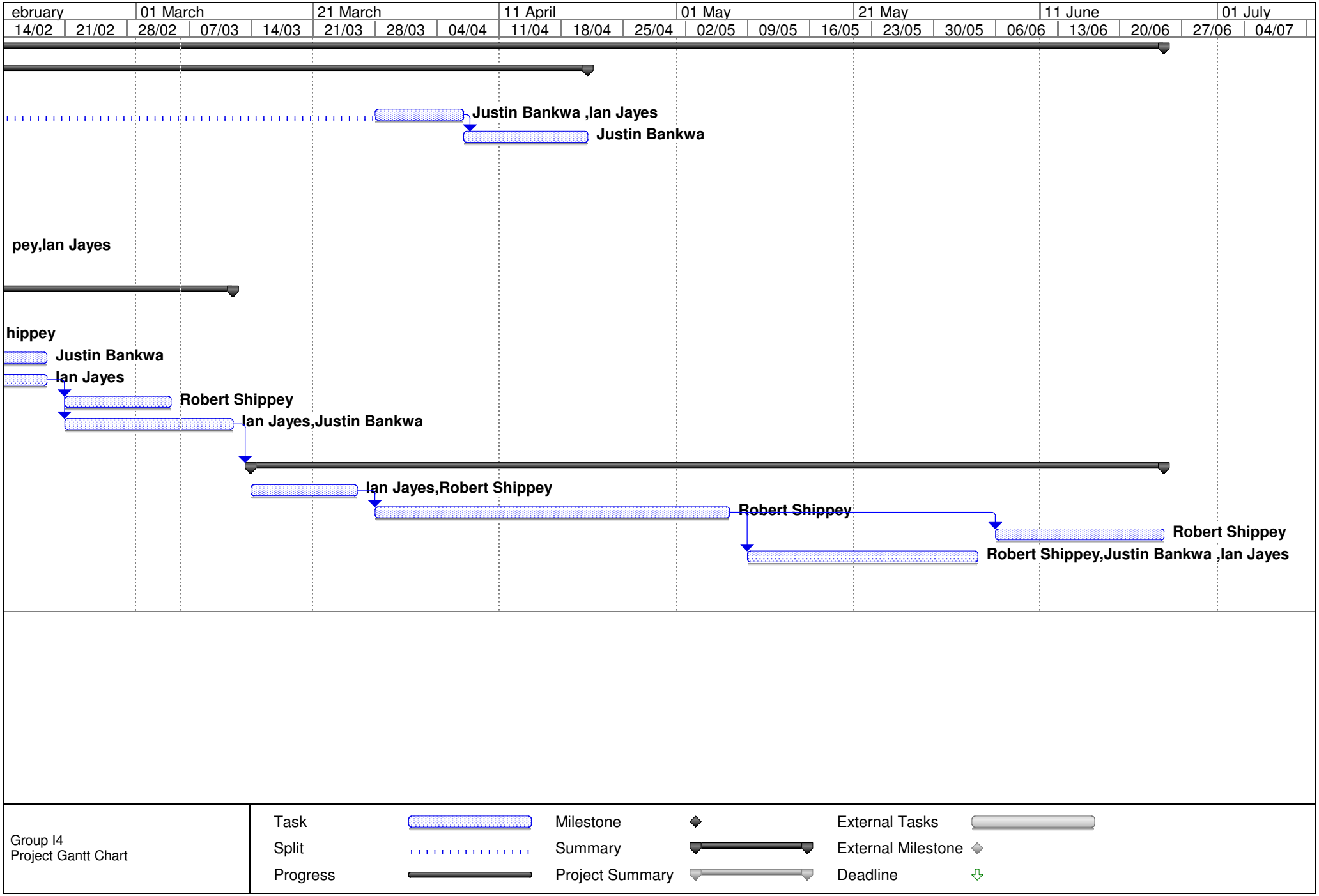
Gantt Chart

ID	Task Name
1	**Analysis Phase**
2	**Create requirements definition**
3	Determine requirements to track
4	Compile requirements as they are gathered
5	Review requirements with sponsor
6	**Gather requirements**
7	Perform document analysis
8	**Conduct interviews**
9	Interview project sponsor
10	Interview other project stakeholders
11	Observe retail store processes
12	**Analyse current system**
13	Draw context diagram
14	Create use cases
15	Draw process models
16	Draw data models
17	Identify improvements
18	Conduct 3 JAD sessions
19	Analysis of current system is complete
20	**Develop concept for new system**
21	Draw context diagram
22	Create use cases
23	Draw process models
24	Draw data models
25	Design Phase
26	Implementation Phase



Group I4
Project Gantt Chart

Task		Milestone		External Tasks	
Split		Summary		External Milestone	
Progress		Project Summary		Deadline	


ebruary
01 March
21 March
11 April
01 May
21 May
11 June
01 July
14/02
21/02
28/02
07/03
14/03
21/03
28/03
04/04
11/04
18/04
25/04
02/05
09/05
16/05
23/05
30/05
06/06
13/06
20/06
27/06
04/07
Justin Bankwa ,Ian Jayes
Justin Bankwa
pey,Ian Jayes
hippey
Justin Bankwa
Ian Jayes
Robert Shippey
Ian Jayes,Justin Bankwa
Ian Jayes,Robert Shippey
Robert Shippey
Robert Shippey
Robert Shippey,Justin Bankwa ,Ian Jayes
Group I4
Project Gantt Chart
Task
Split
Progress
Milestone
Summary
Project Summary
External Tasks
External Milestone
Deadline

IV. System Proposal

Functional Requirements

A functional requirement is something that a system must be able to actively achieve. It may be a receiving a certain input, to processes something, or provide a specified output.

The following is a list of the functional requirements of the Employee Recruitment & Development system. Labels showing whether a requirement is found in the new or current system can be found in the brackets.

- FR1: Receive Applications (new)
- FR2: Receive Job Descriptions (new/current)
- FR3: Select for Interview (new/current)
- FR4: Evaluate and Recruit (new/current)
- FR5: Purge 1 Year Old Job Descriptions (new/current)
- FR6: Create Employee Record (new)

Non-functional Requirements

A non-functional requirement is something that is asked of the system, but is not required for the system to function. The Employee Recruitment & Development system has a number of non-functional requirements, as follows.

- NR1: The system should not allow automatic updates to the Graduate Vacancies database (new)
- NR2: Graduates should be notified of acceptance/rejection within 2 months of application (new)
- NR3: The system should be in place within 18 months (new)
- NR4: Prototype should be in place within three months (new)
- NR5: Application software should be compatible with current network infrastructure
- NR6: All communication between the company and the graduates should be transmitted under secure conditions.

Summary of Use Cases

To clearly show the details of the main functional requirements that were investigated with Use Case tables, we have produced a summary of the main features found for each process. This summary table outlines the purpose, inputs, and outputs of the requirements.

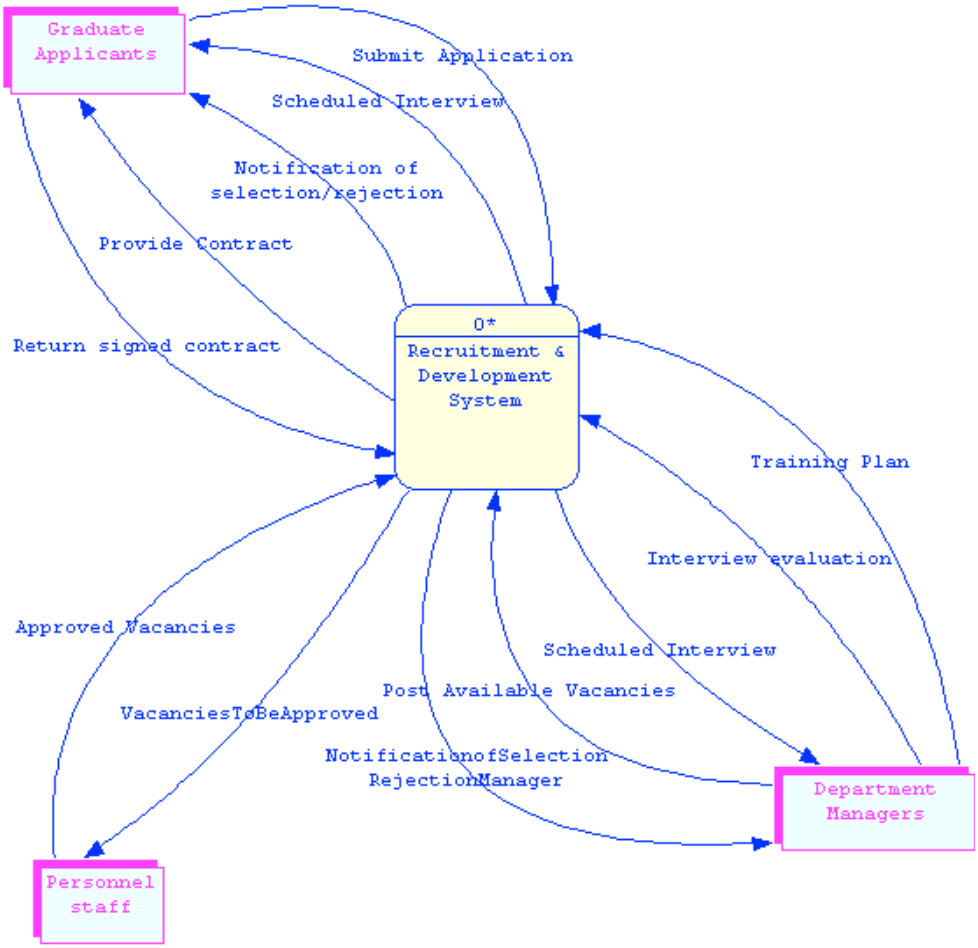
Process No.	1	2	3	4	5	6
Name	**Receive Applications**	**Receive Job Description**	**Select for Interview**	**Evaluate and Recruit**	**Purge 1 year old applications**	**Create employee record**

Purpose	To add applications into the pool	To add vacancies to the file	To select three candidates	To select a new recruit	To keep the database up to date	To store employee data
Input	Application form	Job Description	Graduate Applications, Job Descriptions	Interview evaluation, returned signed contract	All applications	New recruits' application form
Source of input	Graduate applicant	Department manager	Application pool, graduate vacancies	Department managers, recruited graduate	Application pool	Evaluate and recruit process
Output	Application data	Vacancies data	Scheduled interview, selected graduates	Notification of selection and provide unsigned contract, notification of rejection, rejected applications	Delete year old applications	New employee data
Output recipient	Application pool	Graduate vacancies	Graduate and department manager, evaluate and recruit process	Recruited graduate, rejected graduate, application pool	Application pool	Employee file

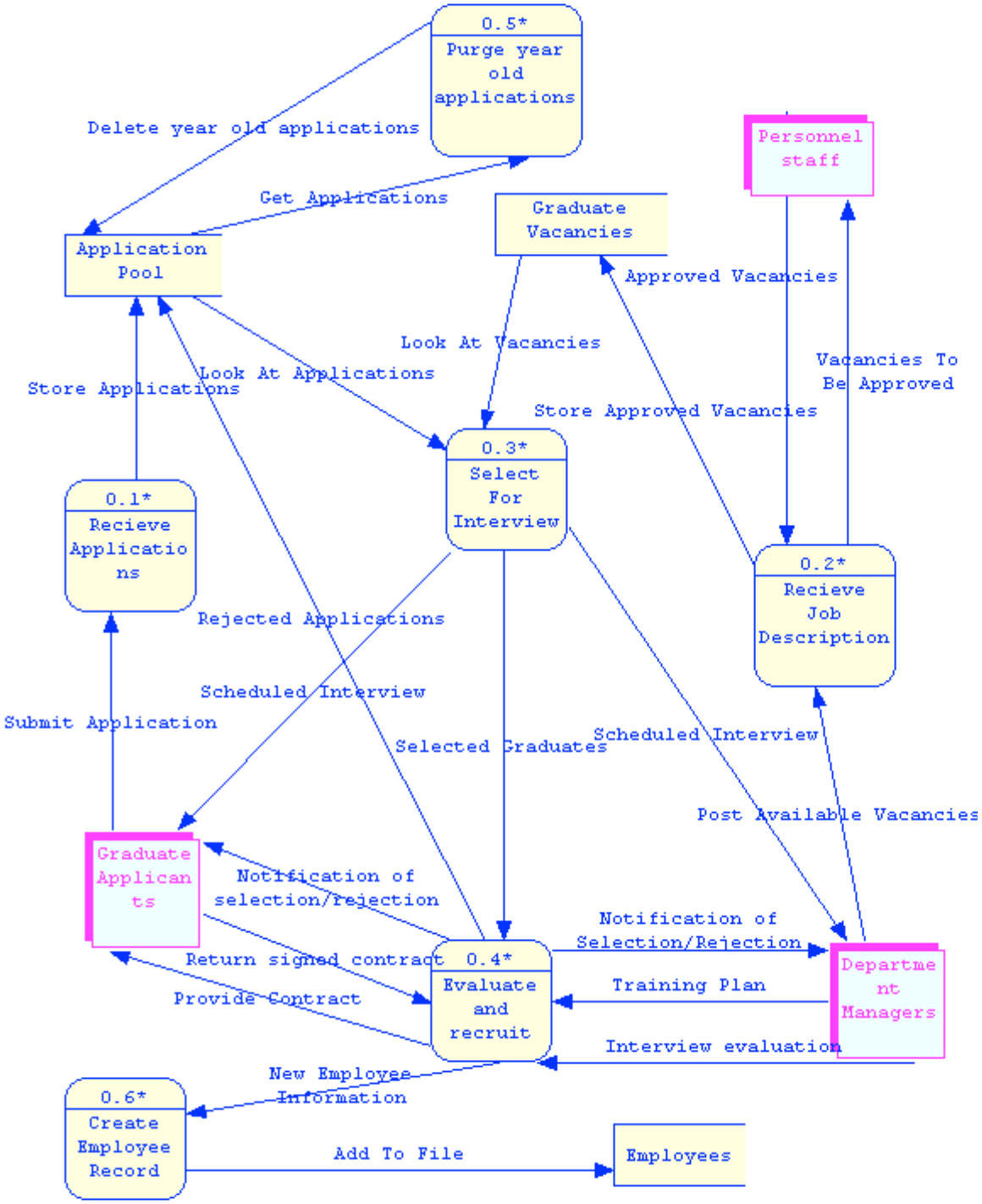
Logical and Physical Process Models

To model the system, we used Data Flow Diagrams using the Gane and Sarson method. We generated DFDs using WinA&D, a CASE tool.

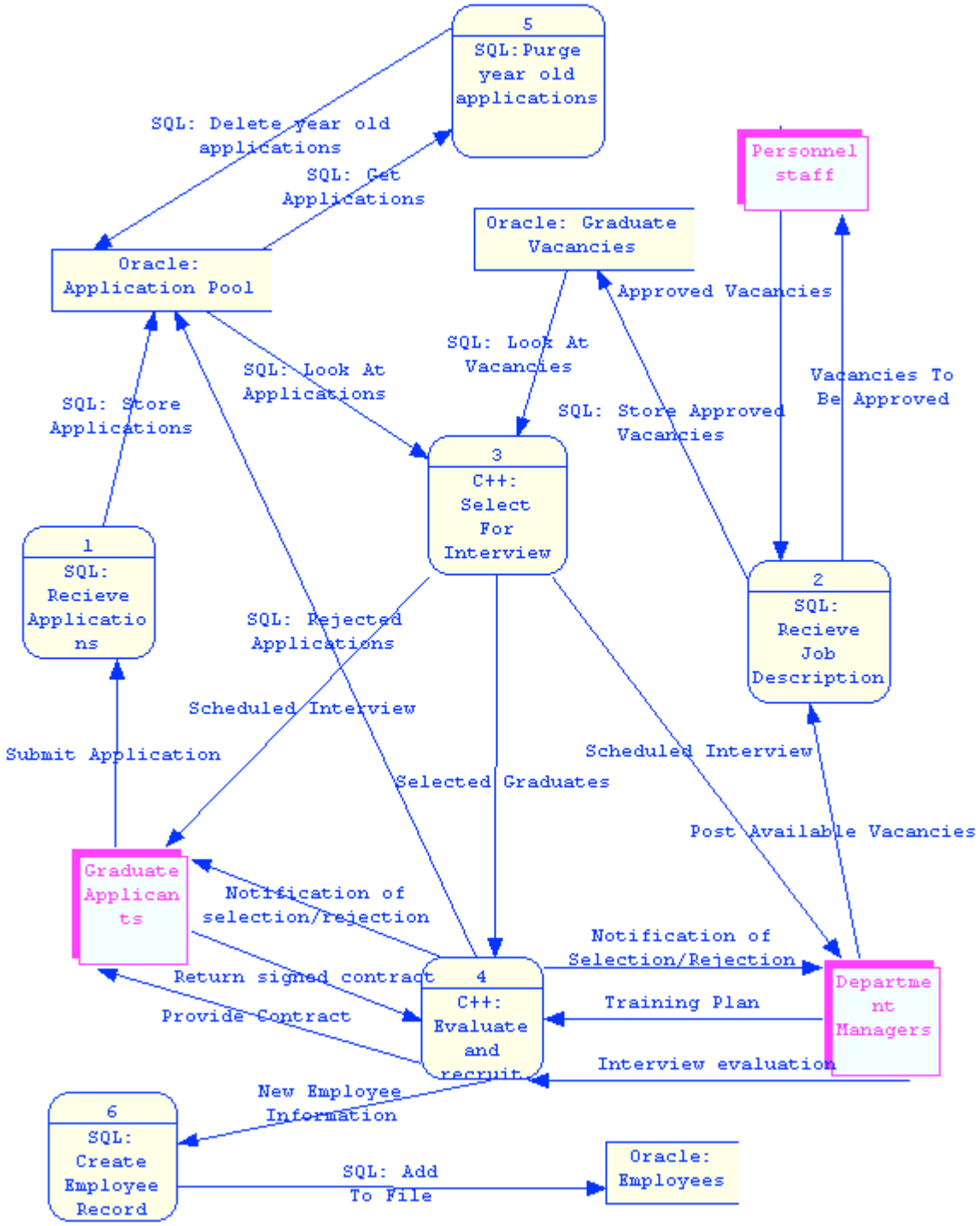
- Logical context level DFD. This shows the system as a whole. It depicts how externals entities interact with the system by showing what data flows from an entity into the system.



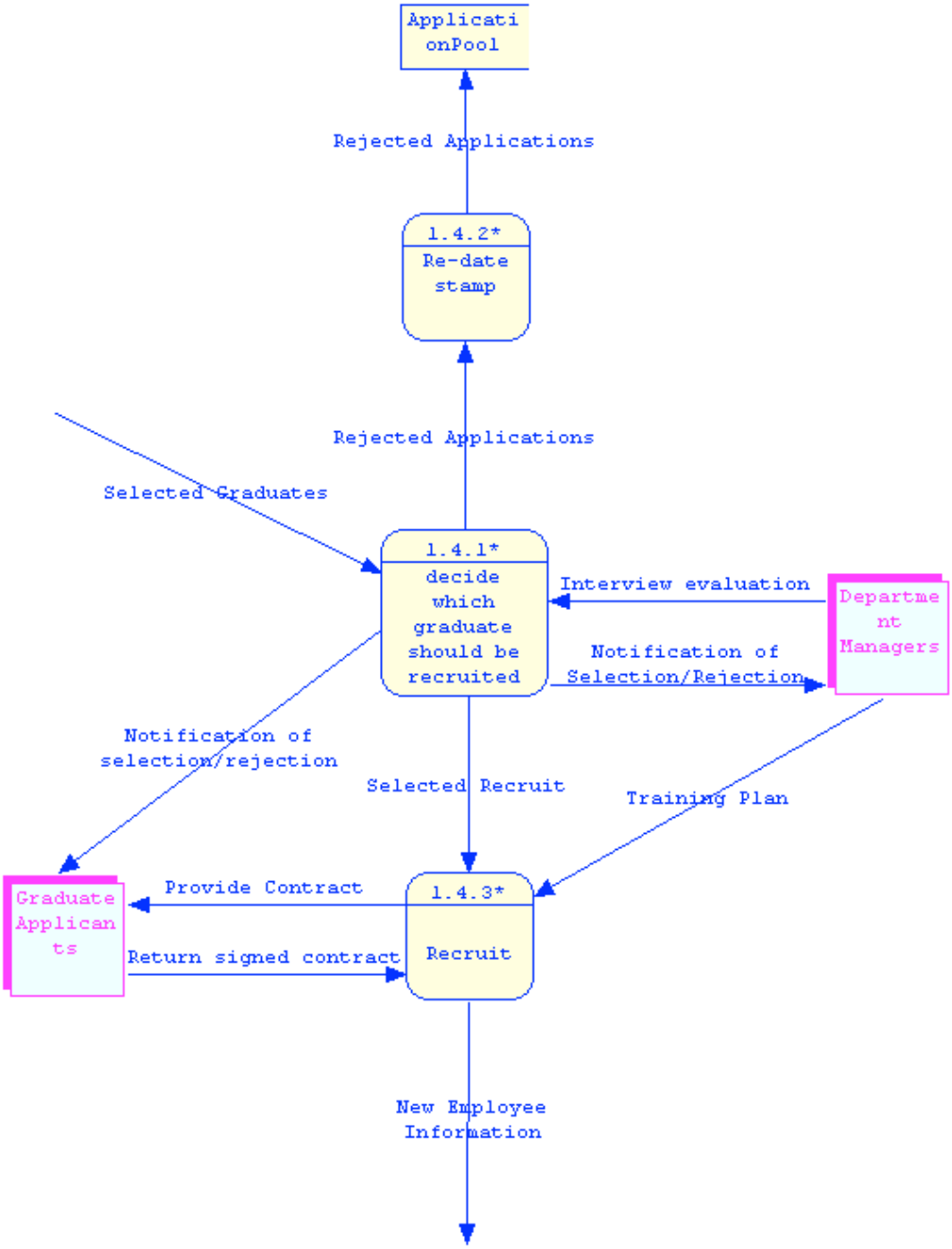
- Logical level 0 DFD. The logical level 0 DFD takes the system, and opens it up to reveal the main functional requirement processes. The model shows how data flows from and to entities. At level 0, we also include data stores and show what data will be passed to and from those.



- Physical level 0 DFD. The physical DFD at level 0 defines what technologies would be implemented on the processes and data stores.



- Logical level 1 DFD for a level 1 process: Evaluate and recruit process. This level 1 DFD gives a decomposed view of the 'Evaluate and recruit' process. It shows sub-processes and data flows inside the level 0 process.



Processing Logic

Below is a decision table demonstrating the processing logic behind the increase in salary for a graduate. It selects a percentage increase based on the type and level of degree the graduate has earned.

3rd class degree?	Y	-	-	-	-	-	-	-
Degree with Hons?	-	N	Y	Y	Y	Y	Y	Y
Science degree?	-	-	Y	Y	Y	N	N	N
1st class degree?	-	-	Y	-	-	Y	-	-
Upper second-class degree?	-	-	-	Y	-	-	Y	-
Lower second-class degree?	-	-	-	-	Y	-	-	Y
No Increase	X	X						
10%			X					
7%				X				
5%					X		X	
8%						X		
2%								X

Normalisation and Logical/Physical Data Modelling

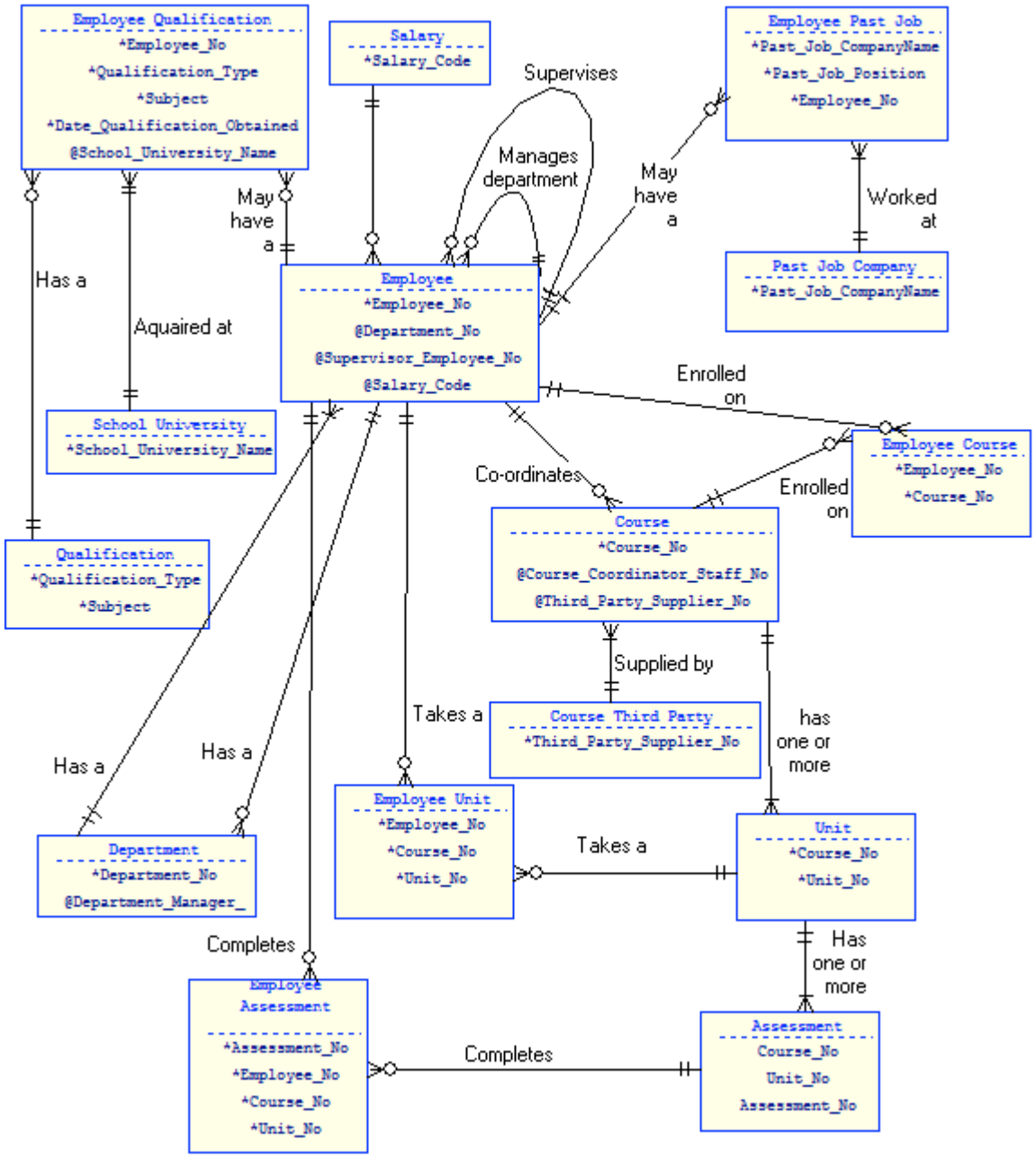
After taking the un-normalised attributes listed in the system description of the business profile and normalising them to Third Normal Form we reached a state of data entities that we feel are suitably normalised. Below are all the tables normalised to Third Normal Form. The stages of normalisation can be found in Appendix B.

Employee table: ***Employee_No** Employee_Name Employee_Address Job_Description Status Start_Date #Supervisor_Employee_No *#Salary_Code* *#Department_No*	Salary table: ***Salary_Code** Salary_Amount	Department table: ***Department_No** Department_Name Department_Location #Dept_Manager_Employee_No
Employee Qualification table: ***Employee_No** ***Qualification_Type** ***Subject** ***Date_Qualification_Obtained** Grade #School_University_Name	Qualification table: ***Qualification_Type** ***Subject** Awarding_Body Qualification_Description	School University table: ***School_University_Name** School_University_Address

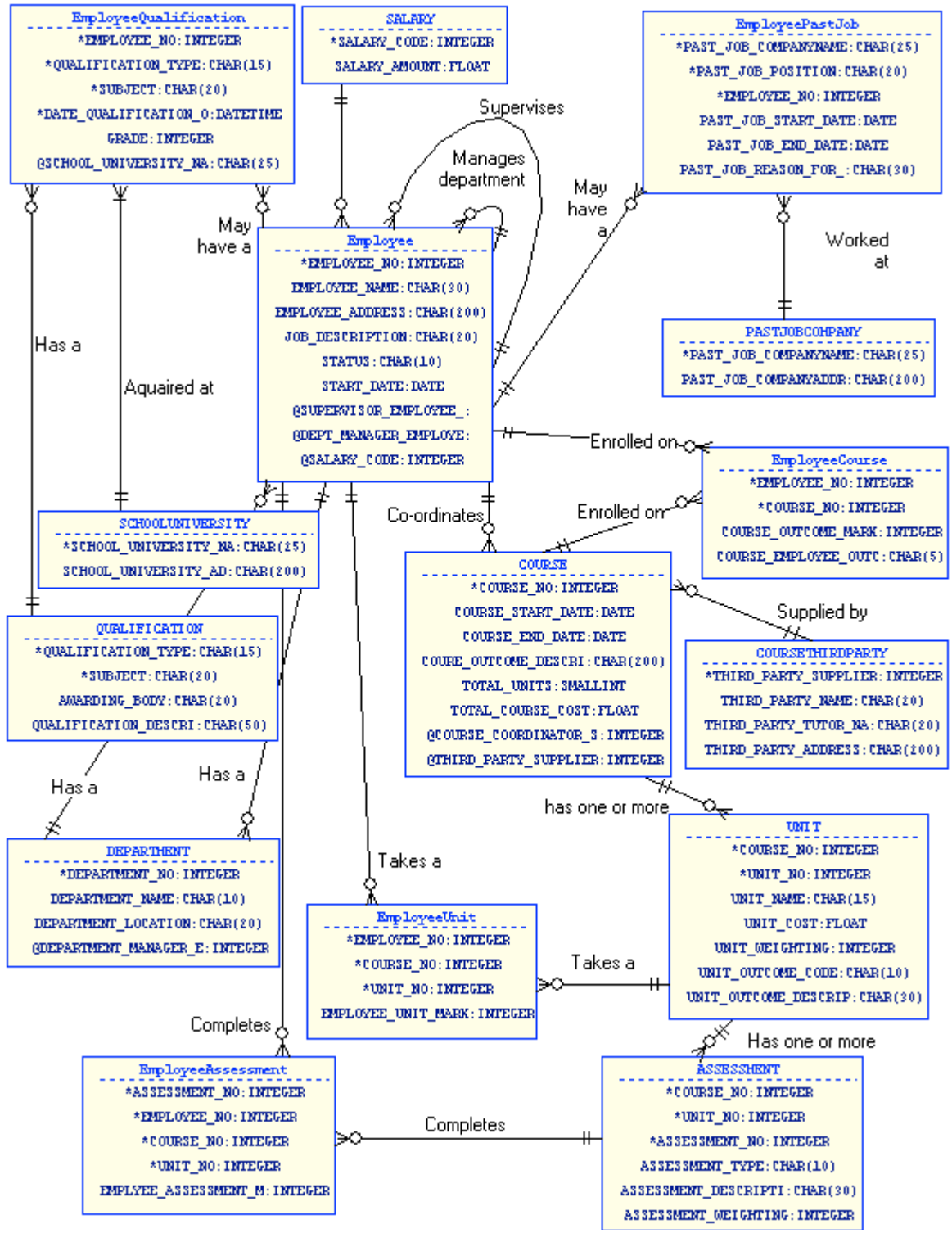
Employee Past Job table: ***Employee_No** ***Past_Job_CompanyName** ***Past_Job_Position** Past_Job_Start_Date Past_Job_End_Date Past_Job_Reason_for_Leaving	Past Job Company table: ***Past_Job_Company Name** Past_Job_CompanyAddress	Course table: ***Course_No** Course_Name Course_Start_Date Course_End_Date Course_Outcome_Description Total_Units Total_Course_Cost #Third_Party_SupplierNo #CourseCoordinator_Staff_No
Third Party Supplier table: ***Third_Party_SupplierNo** Third_Party_Name Third_Party_Tutor_Name Third_Party_Address	Employee Course table: ***Course_No** ***Employee_No** Course_Employee_Mark Course_Employee_Outcome_Code	Unit table: ***Unit_No** ***Course_No** Unit_Name Unit_Cost Unit_Weighting Unit_Outcome_Code Unit_Outcome_Description
Employee Unit table: ***Unit_No** ***Course_No** ***Employee_No** Employee_Unit_Mark	Assessment table: ***Assessment_No** ***Course_No** ***Unit_No** Assessment_Type Assessment_Description Assessment_Weighting	Employee Assessment table: ***Assessment_No** ***Unit_No** ***Course_No** ***Employee_No** Employee_Assessment_Mark

To demonstrate our knowledge of normalisation, we created a new attribute at First Normal Form to be used as part of a compound key for attributes relating to assessments. We found that there was not a good enough attribute to uniquely define an assessment of a unit, even when we created a compound key using Employee_No, Course_No and Unit_No. Assessment_Type would seem like a viable solution but we found this does not adequately identify examples of a unit that has more than one exam, for instance. To properly define an assessment of a unit, we created the Assessment_No field. This will give each assessment within a unit a unique number, allowing multiple exams (or indeed any other type of assessment) with different weightings to be assigned to a unit.

From the Third normal form entities, we created a logical data model that shows all key attributes (both primary and foreign). We also linked related entities with each other using a relevant relationship, which is determined by our use of primary and foreign keys.



To demonstrate the physical data model, we have produced an Entity Relationship Diagram that lists the entity tables, their relationships, and all attributes with their data types.



CRUD (Create Read Update Delete) Matrix

To demonstrate how each of the processes affects the data stores in the system, we created a CRUD matrix. This shows which processes create, read, update or delete information from certain data stores.

Process / Data store	1. Receive Applications	2. Receive Job Descriptions	3. Select for Interview	4. Evaluate and Recruit	5. Purge 1 year old applications	6. Create Employee Record
Application Pool	C		R	U	RD	
Graduate Vacancies		C	R			
Employees						C

Proposal for Change

To make the models meet the system request for the Employee Recruitment & Development project, they would need to be updated.

To enable graduates to post their applications online, a website would have to be created. Data from a form on this website would flow into the Receive Applications process. In the physical model, the process would be composed of HTML and PHP, the data flows would be SQL to enable the website to interface with the Application pool. As the new system is required to maximise paperless contact between entities within and external to the company, it would be necessary to gather graduate applicants email addresses along with their application. This would enable the company to electronically notify graduates of their selection/rejection. In the physical model, the data flows to the Graduate Applicant entity could all be implemented through email.

The system request also states that department managers want to be able to document and recommend salary increases to be considered by senior management. As this is required to be implemented over the companies' intranet, another process in HTML and PHP would be needed to facilitate SQL data flows to the Graduate Vacancies data store. Once the suggested salary increase had been added to the system, a flag would need to be set to allow senior management to see which vacancies had salary increases to be considered.

Finally, the system request asks that the new system should enable department managers to forward training plans to Human Resources before a new employee starts. The training plan will be tailored to the new employee's needs. Currently this happens after an employee signs their contract but if a department manager is able to get the plan ready before the employee has joined the team then the employee will be able to integrate faster and easier into the team. To enable this in the DFDs, information about the potential employees will need to be sent to department managers so that they can tailor the plan to them. Once this is done, there will need to be a new process to facilitate transferring the training plan from the department managers and to Human Resources. This will be using the companies' intranet so it will be HTML and PHP based using SQL for the interfacing with data stores.

V. Evaluation

As the above work demonstrates, we have experienced the systems development processes. In this evaluation we will reflect on this experience and we will discuss successful aspects of our group work, and which parts have room for improvement.

An integral part of any project is planning. We found planning and organising our project difficult, our group is the result of a two-group merge, which occurred because of a number of people leaving in the early stages of this module. This meant that we had to quickly adjust to the new environment and our new colleagues. Initially we set out to have meetings once every two weeks, but this didn't happen. We ended up having meetings monthly and communicating mostly through email. This method worked well for us although we may have been more organised if we did have more meetings.

Our team worked well collaboratively. Once work was divided equally we all carried out our selected tasks and then used meetings to splice the work together to create our deliverables. To assist our collaborative working, we used www.dropbox.com to create a cloud-based folder that we can access from anywhere with an Internet connection. A bonus over the NOW-based file sharing provided by the University is that with a small application provided by Dropbox, a folder on our personal computers can be made to be constantly in synchronisation with the cloud-based folder.

From taking part in this project and working through all the tasks we feel that the process of this project could be improved by implementing a fairer system for grouping members.

We found that concept mapping was not helpful in assisting us to learn. We concluded that it might be useful for people with different learning styles to our own but our example-based learning style was not aided by the abstract and conceptual nature of concept maps.

We found Microsoft Project to be a very interesting and useful tool in laying out a roadmap of a project. Although we found estimating times for each piece of work to be difficult, once we had included task dependencies we could see how the project would progress.

The CASE tool that we used throughout this project, WinA&D, was one of great discussion. On the one hand, it seems to have a huge learning curve, is very difficult to master, and seems to have a few limitations and bugs. However, once we understood how to carry out the tasks that we needed, it is powerful and has the ability to create the types of models that we require.

One feature that we think would be very useful in WinA&D is automatic relationships in Entity Relationship Diagrams. As the attributes for each table are stored in the

program, once we assign primary and foreign keys to entities, the program could have a function that automatically links tables based on the keys that we have entered. This would speed up the creation of ERDs and would also ensure that the type of relationship is always accurate.

Appendix A. Minutes of Meetings

Date: 16th December 2010

Where: CIB
Who attended? : Justin Bankwa, Robert Shippey and Ian Jayes
Meeting leader: Ian Jayes
Starting time: 01:30pm
Finishing time: 02:15pm

Contents:

- Group 3 and 4 merge and prepare for the review point 2 on 17th January 2011.
- We decided to subdivided task to work in during the Christmas period
- Set up a Dropbox account

Decision:

- Tasks subdivide and set a deadline
- Ian Jayes: put in order all the earlier version of the concept map
- Robert Shippey: uses cases and project plan and draft process models
- Justin Bankwa: project plan
- Set a meeting every fortnight

Meeting taker: Justin Bankwa

Date: 10th January 2011

Where: CIB
Who attended? : Justin Bankwa, Robert Shippey and Ian Jayes
Meeting leader: Robert Shippey
Starting time: 02:15
Finishing time: 03:40pm

Contents:

- Review point 2
- Everyone's work done
- Improvement
- Final version

Decision:

- An emergency meeting on 13th January 2011 to evaluate the work for the 17th January 2011's review point

Meeting taker: Justin Bankwa

Date: 13th January 2011
Where: CIB
Who attended? : Justin Bankwa, Robert Shippey and Ian Jayes
Meeting leader: Robert Shippey
Starting time: 12:00pm
Finishing time: 01:00pm

Contents:
- Review point 2
- Final version

Decision:
- Project plan and work breakdown by Justin Bankwa
- Concept map by Ian Jayes
- Use case and (data flow diagrams) by Robert Shippey
- Next meeting on 27th January 2011

Meeting taker: Justin Bankwa

Date: 27th January 2011
Where: CIB
Who attended? : Justin Bankwa, Robert Shippey and Ian Jayes
Meeting leader: Robert Shippey
Starting time: 12:00pm
Finishing time: 12:35

Contents:
- Review point 3
- Task subdivision
- Justin Bankwa: research about the Cmap and prepare for the introduction
- Ian Jayes: put in order all the earlier version of the concept map
- Robert Shippey: finalise the latest version of the concept map

Decision:
- Tasks subdivide and set a deadline
- Final version on the 07th February 2011 in the group Dropbox
- Next meeting on 10th February 2011

Meeting taker: Justin Bankwa

Date: 10th February 2011

Where: CIB
Who attended? : Justin Bankwa, Robert Shippey and Ian Jayes
Meeting leader: Robert Shippey
Starting time: 12:00pm
Finishing time: 14:35

Contents:

- Review point 3
- Justin Bankwa: finalise and bring the research about the Cmap and prepare for the introduction
- Ian Jayes: finalise and bring put in order all the earlier version of the concept map
- Robert Shippey: finalise and bring finalise the latest version of the concept map

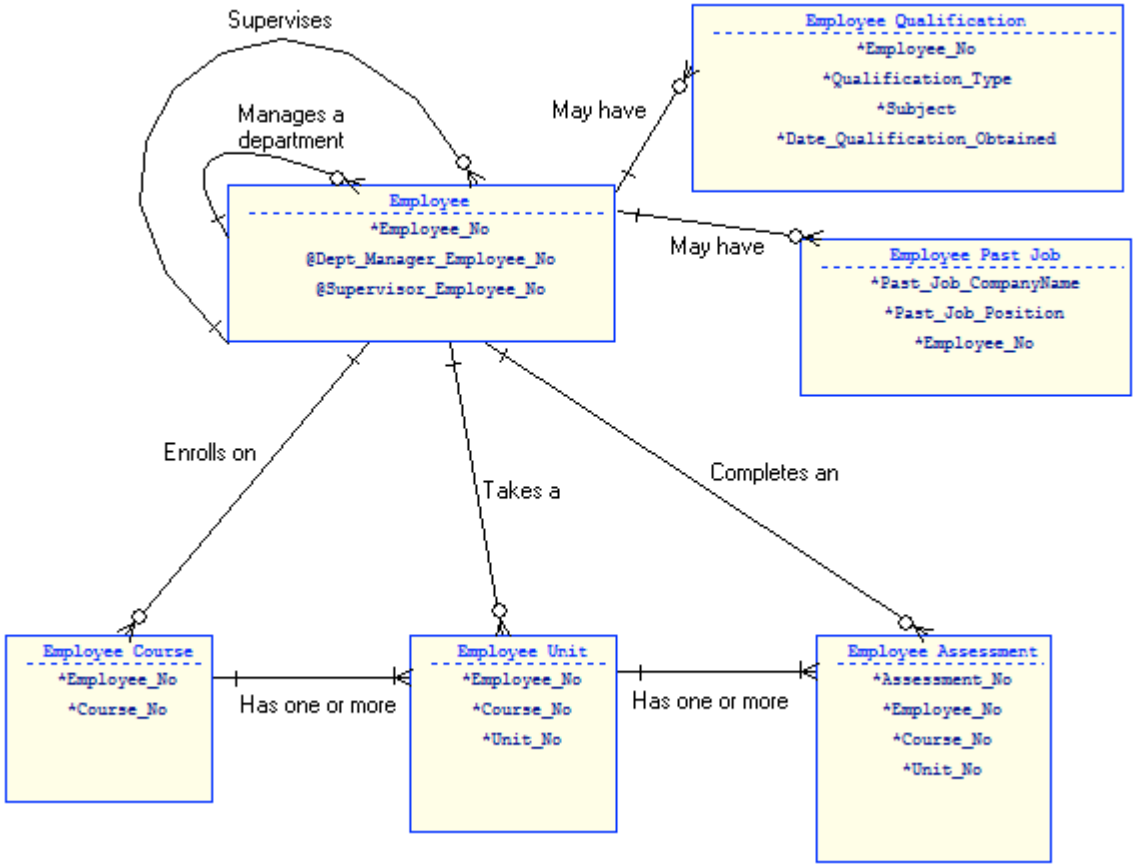
Decision:

- Justin Bankwa will start the PowerPoint presentation
- Ian Jayes will present all the earlier version of the concept map
- Robert Shippey: will present the latest version of the concept map and the Cmap features

Meeting taker: Justin Bankwa

Appendix B. Stages of Normalisation

First Normal Form ERD:



Second Normal Form ERD:

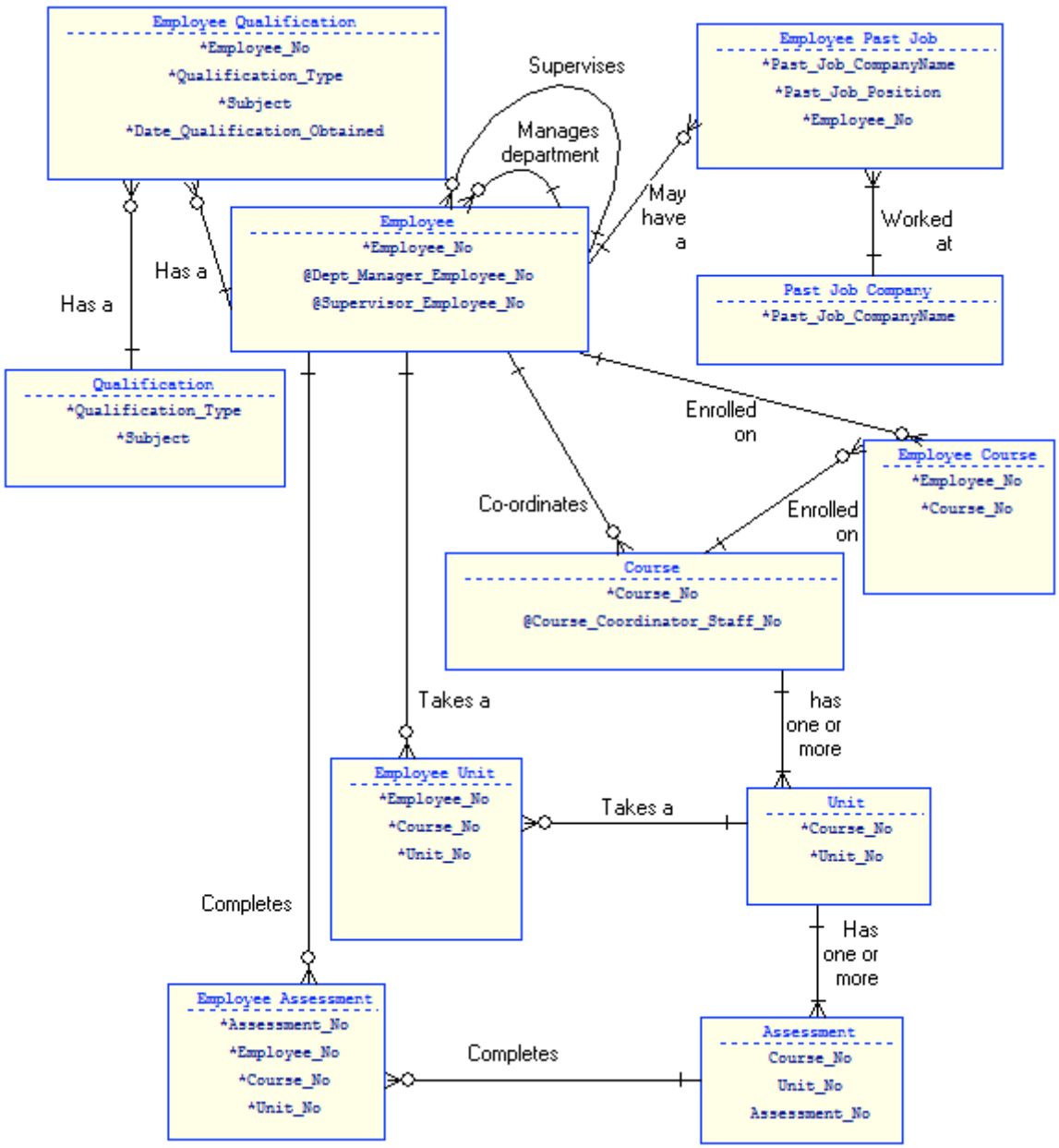


Table showing UNF (including levels) to 3NF:

UNF		1NF	2NF	3NF
Employee_No	1	***Employee_No**	***Employee_No**	Employee table:
Employee_Name	1	Employee_Name	Employee_Name	***Employee_No**
Employee_Address	1	Employee_Address	Employee_Address	Employee_Name
Job_Description	1	Job_Description	Job_Description	Employee_Address
Status	1	Status	Status	Job_Description
Start_Date	1	Start_Date	Start_Date	Status
Salary_Code	1	Salary_Code	Salary_Code	Start_Date
Salary_Amount	1	Salary_Amount	Salary_Amount	#Supervisor_Employee_No
Department_No	1	Department_No	Department_No	*#Salary_Code*
Department_Name	1	Department_Name	Department_Name	*#Department_No*
Department_Location	1	Department_Location	Department_Location	
Dept_Manager_Employee_No	1	*#Dept_Manager_Employee_N*	*#Dept_Manager_Employee_N*	Salary table:
Supervisor_Employee_No	1	*o*	*o*	***Salary_Code**
Qualification_Type	2	*#Supervisor_Employee_No*	*#Supervisor_Employee_No*	Salary_Amount
Qualification_Description	2			
Subject	2	***Employee_No**	***Employee_No**	Department table:
Grade	2	***Qualification_Type**	***Qualification_Type**	***Department_No**
School_University_Name	2	***Subject**	***Subject**	Department_Name
School_University_Address	2	***Date_Qualification_Obtaine**	***Date_Qualification_Obtaine**	Department_Location
Awarding_Body	2	**d**	**d**	#Dept_Manager_Employee_N
Date_Qualification_Obtained	2	Qualification_Description	Grade	o
Past_Job_CompanyName	2	Grade	School_University_Name	
Past_Job_CompanyAddress	2	School_University_Name	School_University_Address	Employee Qualification table:
Past_Job_Position	2	School_University_Address		***Employee_No**
Past_Job_Start_Date	2	Awarding_Body	***Qualification_Type**	***Qualification_Type**
Past_Job_End_Date	2		***Subject**	***Subject**
Past_Job_Reason_for_Leavin	2	***Employee_No**	Qualification_Description	***Date_Qualification_Obtaine**
g	2	***Past_Job_CompanyName**	Awarding_Body	**d**
Course_No	2	***Past_Job_Position**		Grade
Course_Name	2	Past_Job_CompanyAddress	***Employee_No**	#School_University_Name

Course_Start_Date	2	Past_Job_Start_Date	***Past_Job_CompanyName**	
Course_End_Date	2	Past_Job_End_Date	***Past_Job_Position**	Qualification table:
CourseCoordinator_Staff_No	2	Past_Job_Reason_for_Leavin	Past_Job_Start_Date	***Qualification_Type**
Course_Employee_Mark	2	g	Past_Job_End_Date	***Subject**
Total_Units	2		Past_Job_Reason_for_Leavin	Awarding_Body
Course_Employee_Outcome_	2	***Employee_No**	g	Qualification_Description
Code	2	***Course_No**		
Course_Outcome_Description	2	Course_Name	***Past_Job_CompanyName**	School University table:
Total_Course_Cost	2	Course_Start_Date	Past_Job_CompanyAddress	***School_University_Name**
Third_Party_SupplierNo	2	Course_End_Date		School_University_Address
Third_Party_Name	2	CourseCoordinator_Staff_No	***Employee_No**	
Third_Party_Tutor_Name	3	Course_Employee_Mark	***Course_No**	Employee Past Job table:
Third_Party_Address	3	Total_Units	Course_Employee_Mark	***Employee_No**
Unit_No	3	Course_Employee_Outcome_	Course_Employee_Outcome_	***Past_Job_CompanyName**
Unit_Name	3	Code	Code	***Past_Job_Position**
Unit_Cost	3	Course_Outcome_Description		Past_Job_Start_Date
Unit_Weighting	3	Total_Course_Cost	***Course_No**	Past_Job_End_Date
Unit_Outcome_Code	3	Third_Party_SupplierNo	Course_Name	Past_Job_Reason_for_Leavin
Unit_Outcome_Description	4	Third_Party_Name	Course_Start_Date	g
Employee_Unit_Mark	4	Third_Party_Tutor_Name	Course_End_Date	
Assessment_Type	4	Third_Party_Address	Total_Units	Past Job Company table:
Assessment_Description	4		Course_Outcome_Description	***Past_Job_CompanyName**
Assessment_Weighting		***Employee_No**	Total_Course_Cost	Past_Job_CompanyAddress
Employee_Assessment_Mark		***Course_No**	Third_Party_SupplierNo	
		***Unit_No**	Third_Party_Name	Course table:
		Unit_Name	Third_Party_Tutor_Name	***Course_No**
		Unit_Cost	Third_Party_Address	Course_Name
		Unit_Weighting	*#CourseCoordinator_Staff_No*	Course_Start_Date
		Unit_Outcome_Code		Course_End_Date
		Unit_Outcome_Description		Course_Outcome_Description
		Employee_Unit_Mark	***Employee_No**	Total_Units

		***Employee_No** ***Course_No** ***Unit_No** ***Assessment_No** Assessment_Type Assessment_Description Assessment_Weighting Employee_Assessment_Mark	***Course_No** ***Unit_No** Employee_Unit_Mark ***Course_No** ***Unit_No** Unit_Name Unit_Cost Unit_Weighting Unit_Outcome_Code Unit_Outcome_Description ***Employee_No** ***Course_No** ***Unit_No** ***Assessment_No** Employee_Assessment_Mark ***Course_No** ***Unit_No** ***Assessment_No** Assessment_Type Assessment_Description Assessment_Weighting	Total_Course_Cost #Third_Party_SupplierNo #CourseCoordinator_Staff_No <u>Third Party Supplier table:</u> ***Third_Party_SupplierNo** Third_Party_Name Third_Party_Tutor_Name Third_Party_Address <u>Employee Course table:</u> ***Course_No** ***Employee_No** Course_Employee_Mark Course_Employee_Outcome_ Code <u>Unit table:</u> ***Unit_No** ***Course_No** Unit_Name Unit_Cost Unit_Weighting Unit_Outcome_Code Unit_Outcome_Description <u>Employee Unit table:</u> ***Unit_No** ***Course_No** ***Employee_No**

				Employee_Unit_Mark <u>Assessment table:</u> ***Assessment_No** ***Course_No** ***Unit_No** Assessment_Type Assessment_Description Assessment_Weighting <u>Employee Assessment table:</u> ***Assessment_No** ***Unit_No** ***Course_No** ***Employee_No** Employee_Assessment_Mark

Appendix C. Detailed Use-Case Tables

This appendix contains the full details for each use-case for the six functional requirements.

<table>
<tr><td colspan="3">Use Case name: Receive Applications</td><td>ID: FR1</td></tr>
<tr><td colspan="4">Short Description: Information from the graduates application form is inserted into the application pool.</td></tr>
<tr><td colspan="4">Trigger: Graduates post an application to the website or a postal paper application is received.</td></tr>
<tr><td>Type:</td><td colspan="3">External/Temporal</td></tr>
<tr><td colspan="2">Major Inputs</td><td colspan="2">Major Outputs</td></tr>
<tr><td><u>Description</u></td><td><u>Source</u></td><td><u>Description</u></td><td><u>Destination</u></td></tr>
<tr><td>Application form</td><td>Graduate applicant</td><td>Application data</td><td>Application pool</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Major Steps Performed</td><td colspan="2">Information Required:</td></tr>
<tr><td colspan="2">• Online application:
○ Data from the online application is date stamped
○ Then inserted into the application pool
• Postal application:
○ The data from the paper application is typed into an online application, by an employee
○ This is then date stamped
○ And inserted into the application pool
○ The paper application form is then archived</td><td colspan="2">• Date the application has been received
• Graduate’s data, such as:
○ Title
○ Name
○ Address
○ Qualifications
○ Previous work experience
○ Skills
○ Courses attended</td></tr>
</table>

<table>
<tr><td colspan="3">Use Case name: Receive Job Description</td><td>ID: FR2</td></tr>
<tr><td colspan="4">Short Description: Receive a job description from a department manager and write it into the ‘Graduate vacancies’ file.</td></tr>
<tr><td colspan="4">Trigger: A department manager posts a job description to the system.</td></tr>
<tr><td>Type:</td><td colspan="3">External/Temporal</td></tr>
<tr><td colspan="2">Major Inputs</td><td colspan="2">Major Outputs</td></tr>
<tr><td>Description</td><td>Source</td><td>Description</td><td>Destination</td></tr>
<tr><td>Job description</td><td>Department manager</td><td>Vacancies data</td><td>Graduate vacancies</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Major Steps Performed</td><td colspan="2">Information Required:</td></tr>
<tr><td colspan="2">• The incoming job description is date stamped
• The job description data is written to the graduate vacancies file</td><td colspan="2">• Job description
• Essential skills and qualifications
• Desirable skills and qualifications
• Number of positions available
• Recommendation for increase in base salary for successful applicants</td></tr>
</table>

<table>
<tr><td colspan="3">Use Case name: Select for Interview</td><td>ID: FR3</td></tr>
<tr><td colspan="4">Short Description: Compares graduates in the pool of applicants with essential and desirable requirements of each job description, then schedules interviews between graduates and the corresponding managers.</td></tr>
<tr><td colspan="4">Trigger: This process will be run on the first day of the recruiting window</td></tr>
<tr><td colspan="4">Type: External/Temporal</td></tr>
<tr><td colspan="2">Major Inputs</td><td colspan="2">Major Outputs</td></tr>
<tr><td>Description</td><td>Source</td><td>Description</td><td>Destination</td></tr>
<tr><td>Graduate applications</td><td>Application pool</td><td>Scheduled interview notification</td><td>Graduate</td></tr>
<tr><td>Job descriptions</td><td>Graduate vacancies</td><td>Scheduled interview notification</td><td>Department manager</td></tr>
<tr><td></td><td></td><td>Selected graduates</td><td>Evaluate and recruit</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Major Steps Performed</td><td colspan="2">Information Required:</td></tr>
<tr><td colspan="2">• Takes all graduates in the application pool
• Compares applications to the requirements of a job description
• Selects three graduates that best fit the description
• Notifies the department manager of the three selected graduates
• Notifies the three selected graduates that they have been chosen
• Schedules interviews between each selected graduate and the corresponding manager
• Passes the three selected graduates into the evaluate and recruit process</td><td colspan="2">• Essential and desirable requirements
• Qualifications, work experience, and skills of the graduates
• Selected graduates contact information
• Department managers contact information and calendar</td></tr>
</table>

Use Case name: Evaluate and Recruit			**ID:** FR4
Short Description: Of the three selected graduates, one is selected based on their application and the evaluation of the candidates interview with the manager. The one selected candidate is then recruited.			
Trigger: Receiving the evaluation of the three selected graduates interviews.			
Type:	***External***/Temporal		
Major Inputs		**Major Outputs**	
Description	Source	Description	Destination
Interview evaluation	Department managers	Notification of selection	Recruited graduate
Returned signed contract	Recruited graduate	Notification of rejection	Rejected graduates
		Provide unsigned contract	Recruited graduate
		Rejected applications	Application pool
Major Steps Performed		**Information Required:**	
• Receive interview evaluation • Make a decision on which graduate should be recruited • Notify the department manager which graduate has been recruited • Notify the recruited graduate they have been selected • Notify the rejected graduates that they have not been chosen for this position (but their applications have been put back into the pool) • Rejected applications are re-date stamped • Rejected applications are stored back into the application pool.		• Interview evaluation • Contract • Graduates contact information • Department managers contact information • Selected applications	

Use Case name: Purge 1 year old applications			**ID:** FR5
Short Description: Delete all applications in the application pool that are a year old.			
Trigger: This process is started everyday to ensure that the pool is regularly kept up to date.			
Type:	External/***Temporal***		
Major Inputs		**Major Outputs**	
Description	Source	Description	Destination
Look up all applications	Application pool	Delete all year old applications	Application pool
Major Steps Performed		**Information Required:**	
• Look up applications that have a date stamp that is a year old • Delete these applications		• Todays date • Date stamp of applications	

<table>
<tr><td colspan="3">Use Case name: Create employee record</td><td>ID: FR6</td></tr>
<tr><td colspan="4">Short Description: Creates a new employee record every time a graduate is recruited.</td></tr>
<tr><td colspan="4">Trigger: A new employee is recruited.</td></tr>
<tr><td>Type:</td><td colspan="3">External/Temporal</td></tr>
<tr><td colspan="2">Major Inputs</td><td colspan="2">Major Outputs</td></tr>
<tr><td>Description</td><td>Source</td><td>Description</td><td>Destination</td></tr>
<tr><td>New recruits' application form</td><td>Evaluate and recruit process</td><td>New employee data</td><td>Employee file</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Major Steps Performed</td><td colspan="2">Information Required:</td></tr>
<tr><td colspan="2">• Make sure all required information is present
• Store the data in the employee file</td><td colspan="2">• Title
• Name
• Address
• Qualifications
• Previous work experience
• Skills
• Courses attended
• Current department</td></tr>
</table>