COMP 354: INTRODUCTION TO SOFTWARE ENGINEERING

PROJECT MANAGEMENT APPLICATION
SOFTWARE REQUIREMENTS SPECIFICATION

Team B

Alex Huot
Dmitri Novikov
David Gurnsey
Jun Hui Chen
Philip Eloy
Robert Wolfstein
Nicholas Francoeur
Mehdi Jamal Mokhtar
Nivine Shebarou
Lee Kelly
Steven Di Lullo

Friday, June 6th, 2014

Revision History:

Date Created:	May 21st, 2014
Most Recent Update:	June 6th, 2014
Previous Updates:	June 5th, 2014 June 4th, 2014 June 2nd, 2014 June 1st, 2014 May 30th, 2014 May 28th, 2014

Table of Contents

Project Description:

Introduction:

Project Management Application, PMA, is a productivity application designed to allow Project Managers to model and analyze the various activities and Resources involved in their projects via a graphical user interface.

Context:

PMA is designed with the Project Manager foremost in mind. By modelling Projects as a set of Activities, the Project Manager can divvy up work among Team Members, monitor Cost and Schedule overruns, as well perform Critical Path, PERT, and Earned Value Analysis on their Projects.

Business Goals:

The successful outcome of this project will be an application that allows for Project Managers to organize their resources and analyze the health of their Projects.

Scope:

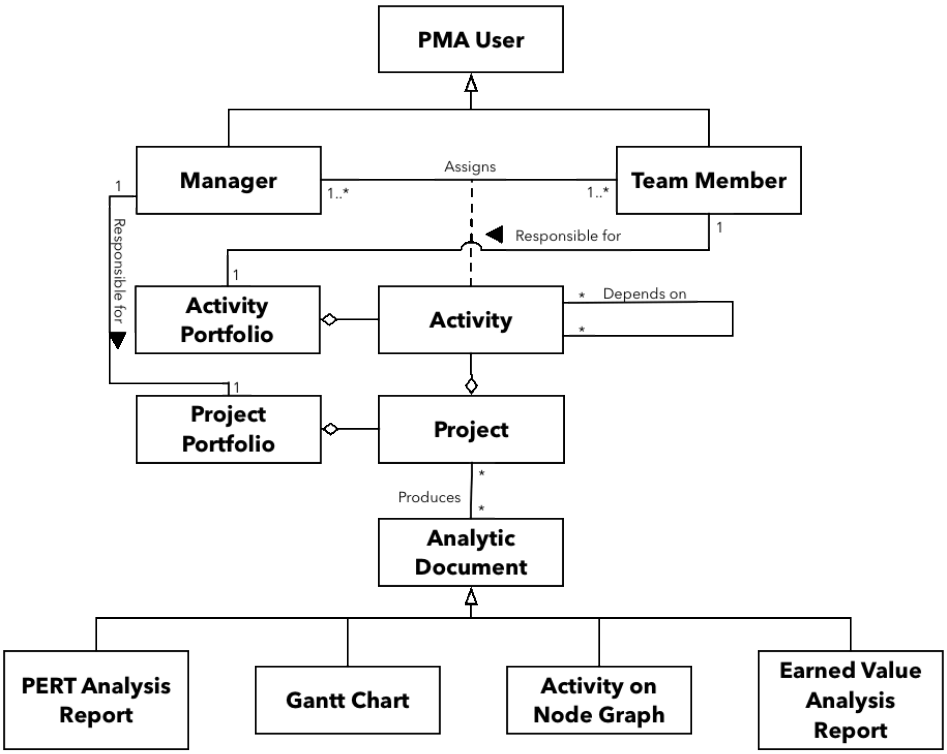
The scope of PMA must be tractable. Given the potential for complexity, certain features that are not specifically requested by the stakeholders are omitted. They include:

Network Connectivity: One might expect a real-time interconnected environment for users to update and be notified about updates as the application runs.

Multiple Concurrent Users: An extension of the above constraint is that no two users should be using the application at the same time. The reason for this is that our database implementation is shared and needs to be protected against updates that would lead to in-consistencies.

Managers as Extended Team Members: Intuitively, one might expect that Project Managers might treat one another as Team Members, assigning Activities to them. While this makes sense and would make for a robust implementation, it would also require a much more sophisticated application logic than would otherwise be the case. Again, given that there is no explicit stakeholder request for such functionality, we leave it out. Thus, the total extent of actions available to the Team Member is that they may view the Activities assigned to them.

Domain Model:



Activity
ID description status startDate optimisticTimeToCompletiton pessimisticTimeToCompletion realisticTimeToCompletion earliestStartTime latestStartTime earliestFinishTime latestFinishTime expectedDuration targetCompletionDate zScore probabilityOfCompletionByTarget plannedValue actualCost

Project
ID description status projectValue earnedValue budgetAtCompletion actualCost percentComplete costVariance scheduleVariance costPerformanceIndex schedulePerformanceIndex estimateAtCompletion estimateToCompletion startDate deadline

As we see from the model above, users of our PMA can one be of two types: Project Managers or Team Members. The association between these two is that a Manager *assigns* Activities to a Team Member. The Team Member is thus *responsible for* the *assigned* Activity, as they are for all of the Activities that *aggregate* to form their Activity Portfolio. Similarly, Managers are responsible for each of the Projects that exist within their Project Portfolio.

Each Project contains data relevant to its progression in terms of cost and schedule. Some of this data is input by the Manager while other data is derived from the values in the data fields of the Activities that make up the Project.

There is a many-to-many relationship between Activities that we call *depends on,* which indicates that a given Activity must wait for zero or more other Activities to be completed before it can commence. This Activity may in turn have Activities that depend on it, but the information is only necessary in a forward direction, that is, a given Activity is unconcerned with those Activities that depend upon it, only those upon which it depends.

Actors:

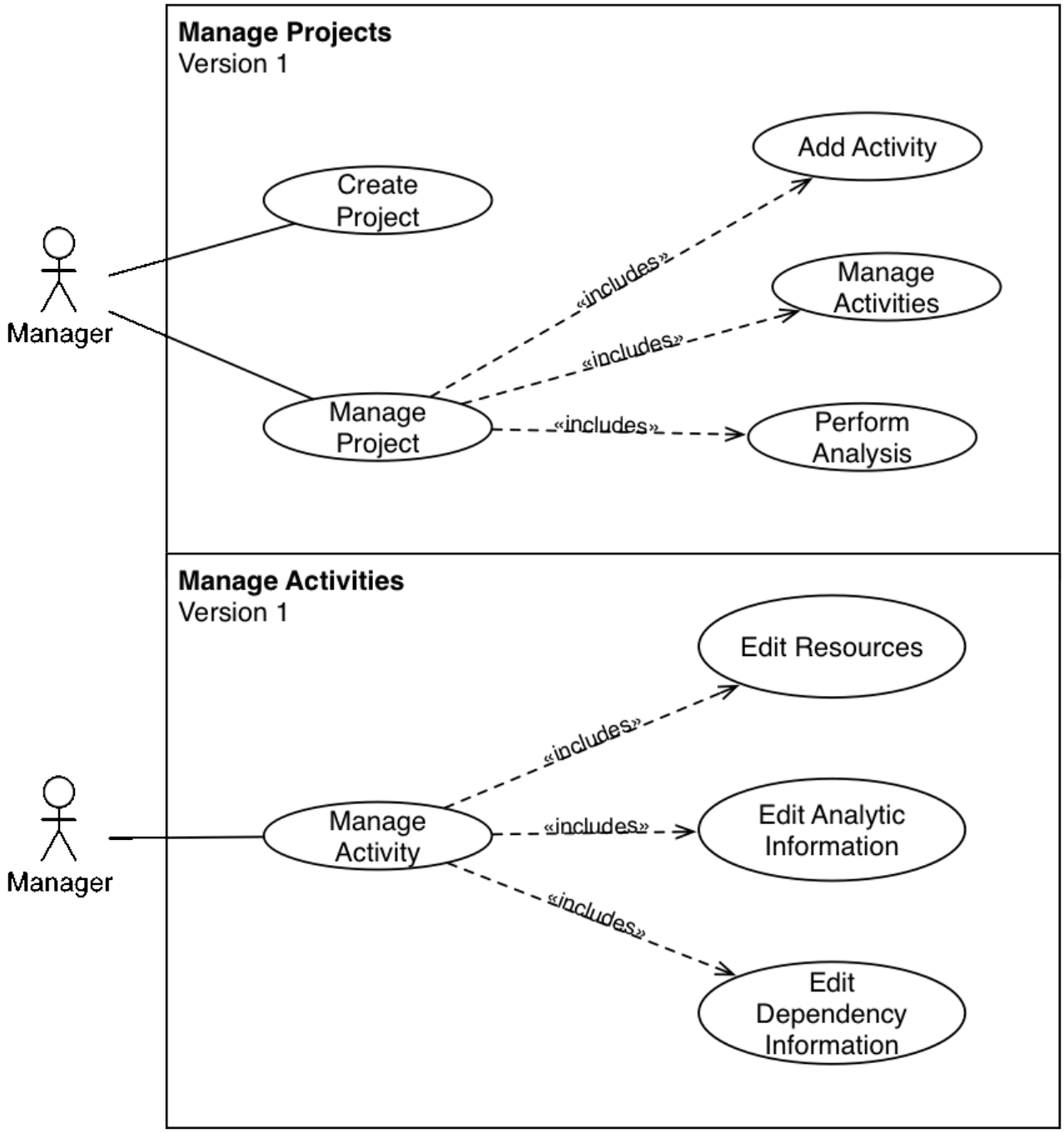
Project Managers are users who organize and manage Projects.

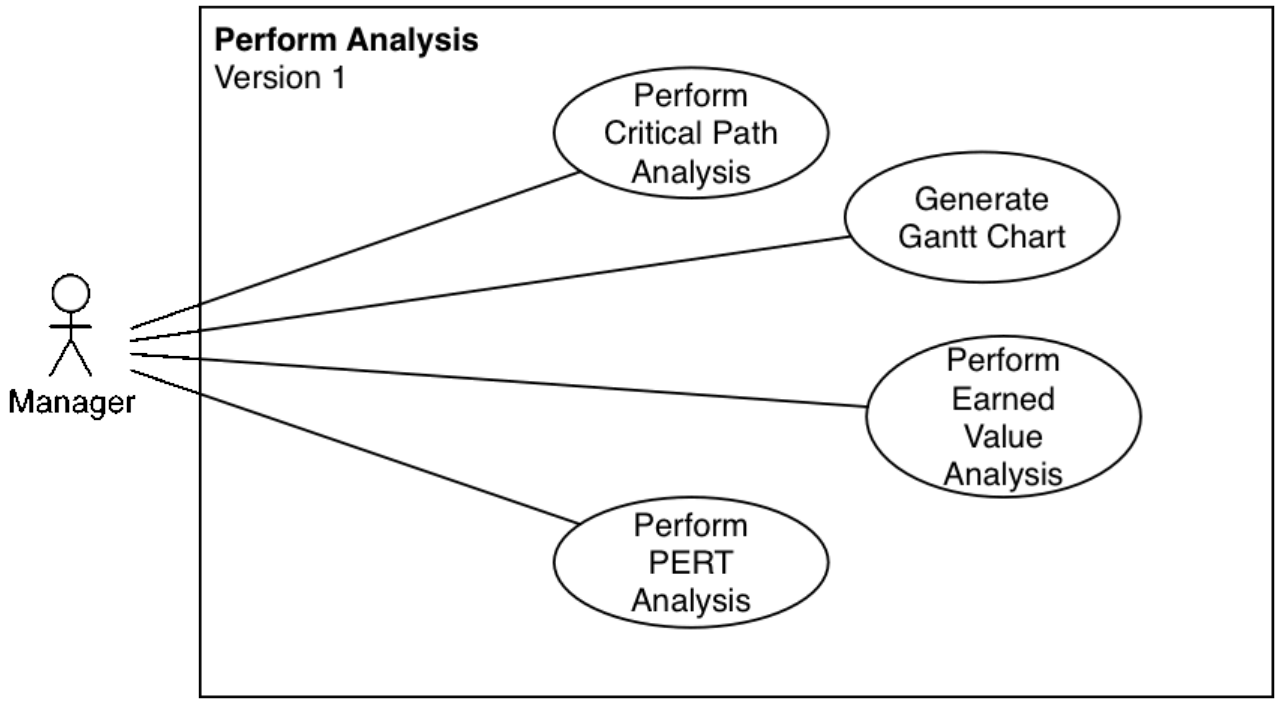
Project Members are users to whom Activities are assigned.

Functional Requirements:

Overview:

Please note that a complete treatment of the User Stories, converting them into explicit functional requirements can be found in the User Story Conversions section.





Use Cases:

Use Case: Create Project <<includes>> Initialize Project Details

Successful Outcomes: Primary Actor creates a new Project.

Use Case Package	General Management
ID	UC-GM-01
Use Case Goal	Primary actor successfully creates a new Project.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	Manager has successfully logged in as per {UC-GF-01}
Domain Entities	Project, Manager

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to create a Project.	
2	System prompts Primary Actor to enter new Project in-	

Step	Action	Notes
	formation.	
3	Primary Actor performs Initialize Project Details. {UC-PM-01}	
4	System generates a unique Project ID and creates a new Activity with the information gathered in*Main Success Scenario step 3*. Saves the new Project.	
5	*Use case ends successfully.*	

Alternative Flows:

3a. Initialize Project Details ends unsuccessfully.

Step	Action	Notes
3a.1	Primary Actor enters information that causes Initialize Project Details {UC-PM-01} to fail.	
3a.2	*System returns control to Main Success Scenario Step 2.*	

3b. Initialize Project Details ends unsuccessfully.

Step	Action	Notes
3b.1	Primary Actor enters information that causes Initialize Project Details {UC-PM-01} to fail.	
3b.2	User aborts.	
3b.2	*Use case ends unsuccessfully.*	

Use Case: Initialize Project Details

Successful Outcomes: Primary Actor enters the details required to initialize a Project.

Use Case Properties:

Use Case Package	Project Management
ID	UC-PM-01
Use Case Goal	Primary Actor enters valid Project Name, Project Description, Start Date, Target Completion Date, and Budget information required for a new Project.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	Primary Actor is engaged in Create Project {UC-GM-01} .
Domain Entities	Manager, Project, Start Dat, Target Completion Date, Budget

Main Success Scenario:

Step	Action	Notes
1	Primary Actor enters Project Name, Project Description, Start Date, Target_Completion Date, Budget informa-tion.	
2	System verifies the internal consistency of the entered information and prompts the user to confirm.	
3	Primary Actor confirms.	
4	System saves changes to the Project.	
5	*Use case ends successfully.*	

Alternative Flows:

2a. Inconsistent Date Information.

Step	Action	Notes
2a.1	System informs Primary Actor that the Date information is invalid (e.g. Target Completion Date is before the Start Date or Current Date).	
2a.2	*System returns control to Main Success Scenario step 1.*	

2b. Inconsistent Date Information.

Step	Action	Notes
2b.1	System informs Primary Actor that the Date information is invalid (e.g. Target Completion Date is before the Start Date or Current Date).	
2b.2	Primary Actor aborts.	
2b.3	*Use case ends unsuccessfully.*	

Use Case: Edit Project Details

Successful Outcomes: Primary Actor updates the details of the existing Project.

Use Case Properties:

Use Case Package	Project Management
ID	UC-PM-02
Use Case Goal	Primary Actor successfully updates any of the following: Project Name, Project Description, Start Date, Target Completion Date, and Budget.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	Primary Actor has access to their Portfolio as the result of having successfully completed Login{UC-GF-01}.
Domain Entities	Manager, Project, Start Date, Target Completion Date, Budget Portfolio

Main Success Scenario:

Step	Action	Notes
1	Primary Actor edits any of the following: Project Name, Project Description, Start Date, Target Completion Date, Budget information.	
2	System verifies the internal consistency of the entered information. Then prompts the user to confirm.	
3	Primary Actor confirms.	
4	System saves changes to the Project.	
5	*Use case ends successfully.*	

Alternative Flows:

2a. Inconsistent Date Information.

Step	Action	Notes
2a.1	System informs Primary Actor that the Date information is invalid (e.g. Target Completion Date is before the Start Date or Current Date).	

Step	Action	Notes
2a.2	*System returns control to Main Success Scenario step 1.*	

2b. Inconsistent Date Information.

Step	Action	Notes
2b.1	System informs Primary Actor that the Date information is invalid (e.g. Target Completion Date is before the Start Date or Current Date).	
2b.2	Primary Actor aborts.	
2b.3	*Use case ends unsuccessfully.*	

Use Case: Create Activity <<includes>> Initialize Activity Details

Successful Outcomes: Primary Actor creates a new Activity.

Use Case Properties:

Use Case Package	Activity Management
ID	UC-AM-01
Use Case Goal	Primary actor successfully creates a new Activity.
Actor(s)	Primary Actor: User
Level	User-Goal
Precondition	Manager is logged in to their Project Portfolio as per Login {UC-GF-01}.
Domain Entities	Activity, Manager, Portfolio, Activity ID

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to create an Activity.	
2	System prompts Primary Actor to enter new Activity information.	
3	Primary Actor performs Initialize Activity Details {UC-AM-02}	
4	System generates a unique Activity ID and creates a new Activity with the information gathered in step 3.	
5	*Use case ends successfully.*	

Alternative Flows:

3a. Initialize Project Details ends unsuccessfully.

Step	Action	Notes
3a.1	Primary Actor enters information that causes Initialize Activity Details {UC-AM-02} to fail.	
3a.2	*System returns control to Main Success Scenario step 1.*	

3b. Initialize Project Details ends unsuccessfully.

Step	Action	Notes
3b.1	Primary Actor enters information that causes Initialize Activity Details {UC-AM-02} to fail.	
3b.2	User Aborts.	
3b.3	*Use case ends unsuccessfully.*	

Use Case: Initialize Activity Details

Successful Outcomes: Primary Actor enters the details required to initialize a Activity.

Use Case Properties:

Use Case Package	General Management
ID	UC-AM-02
Use Case Goal	Primary Actor enters valid Activity Name, Activity Description, Start Date, Target Completion Date, and Budget information required for a new Activity.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	Primary Actor is engaged in Create Activity {UC-AM-01} .
Domain Entities	Manager, Activity, Activity Name, Activity Description, Start Date, Target Completion Date, Budget

Main Success Scenario:

Step	Action	Notes
1	Primary Actor enters Activity Name, Activity Description, Start Date, Target Completion Date, Budget information.	
2	System verifies the internal consistency of the entered information and prompts the user to confirm.	
3	Primary Actor confirms.	
4	System saves changes to the Activity.	
5	*Use case ends successfully.*	

Alternative Flows:

2a. Inconsistent Date Information.

Step	Action	Notes
2a.1	System informs Primary Actor that the Date information is invalid (e.g. Target Completion Date is before the Start Date or Current Date).	
2a.2	*System returns control to Main Success Scenario step 1.*	

2b. Inconsistent Date Information.

Step	Action	Notes
2b.1	System informs Primary Actor that the Date information is invalid (e.g. Target Completion Date is before the Start Date or Current Date).	
2b.2	Primary Actor aborts.	
2b.3	*Use case ends unsuccessfully.*	

Use Case: Add Resource

Successful Outcomes: Primary Actor adds a Resource to an Activity.

Use Case Properties:

Use Case Package	Activity Management
ID	UC-AM-03
Use Case Goal	The Primary Actor adds a Resource to the Activity.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	Primary Actor is a Manager, and the selected Activity has been initialized {UC} and is not completed, and the selected Resource is available.
Domain Entities	Resource, Activity, Manager

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates their intention to add a new Resource to the selected Activity.	
2	System responds by presenting a list of available Resources.	
3	Primary Actor selects a valid Resource from the list.	
4	System prompts Primary Actor for confirmation about the addition of the Resource.	
5	Primary Actor confirms.	
6	*Use case ends successfully.*	

Use Case: Remove Resource

Successful Outcomes: Primary Actor removes a Resource from an Activity.

Use Case Properties:

Use Case Package	Activity Management
ID	UC-AM-04
Use Case Goal	The Primary Actor removes a Resource from the Activity.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	Primary Actor is a Manager, that the selected Activity has been initialized {UC} and is not completed, and the selected Resource is assigned to the selected Activity.
Domain Entities	Resource, Activity, Manager

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention remove a specific Resource from the selected Activity.	
2	System responds by presenting the list of assigned Resources for the Activity.	
3	Primary Actor selects a Resource from the list.	
4	System prompts Primary Actor for confirmation about the removal of the Resource.	
5	Primary Actor confirms.	
6	*Use case ends successfully.*	

Use Case: Edit Cost

Successful Outcomes: Primary Actor updates the Cost information for the selected Activity.

Use Case Properties:

Use Case Package	Activity Management
ID	UC-AM-05
Use Case Goal	Primary User successfully edits any of the following Activity values: AC.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	The Primary Actor has at least one active Activity in the current Project.
Domain Entities	Activity, Manager, AC.

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to edit the AC value of the Activity.	
2	System validates correctness of new data values and prompts the Primary Actor to confirm the update.	
3	User confirms.	
4	*Use case ends successfully.*	

Alternative Flows:

4a. Invalid entry.

Step	Action	Notes
4a.1	Data entered by the Primary Actor fails consistency check (i.e. Wrong data type, negative value ...).	
4a.2	System informs Primary Actor of the error and prompts re-try.	
4a.3	*Use case resumes at Main Success Scenario step 1.*	

4b. Invalid entry.

Step	Action	Notes
4b.1	Data entered by the Primary Actor fails consistency check (i.e. Wrong data type, negative value ...).	
4b.2	System informs Primary Actor of the error and prompts re-try.	
4a.3	User aborts.	
4a.4	*Use ends unsuccessfully.*	

Use Case: Edit Dependancies

Successful Outcomes: The list of Activities that the currently selected Activity depends on is updated to reflect the correct sequence of the Project.

Use Case Properties:

Use Case Package	Activity Management
ID	UC-AM-06
Use Case Goal	Primary Actor updates list of preceding Activities.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	The Primary Actor is a Manager and has at least 2 Activities that are active in the current Project.
Domain Entities	Manager, Project, Activity

Main Success Scenario:

Step	Action	Notes
1	Primary Actor selects an Activity whose dependancy information will be updated.	
2	System responds by presenting the dependancy information of the Activity.	
3	Primary Actor makes the change to the dependancy information.	
4	System verifies that the update is valid and prompts the Primary Actor to confirm.	
5	Primary Actor confirms the change.	
6	*Use case ends successfully.*	

Alternative Flows:

4a. Invalid update.

Step	Action	Notes
4a.1	Verification of the update fails and the System informs the user of the error and prompts a retry.	

Step	Action	Notes
4a.2	*System returns control to Main Success Scenario step 3.*	

4b. Invalid update.

Step	Action	Notes
4b.1	Verification of the update fails and the System informs the user of the error and prompts a retry.	
4b.2	Primary Actor aborts.	
4b.3	*Use Case ends unsuccessfully.*	

6a. Primary has more changes to make.

Step	Action	Notes
6a.1	Primary Actor indicates intention to make more changes.	
6a.2	*System returns control to Main Success Scenario step 1.*	

Use Case: Generate Critical Path

Successful Outcomes: System calculates and displays a visual representation of the Project time table, it's interdependent sequence of Activities as an Activity on Node with the Critical Path highlighted.

Use Case Properties:

Use Case Package	Data Analysis
ID	UC-DA-01
Use Case Goal	The Primary Actor has a Activity on Node compiled and displayed for them with the Critical Path highlighted.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	Primary Actor has an active Project with a logically consistent schedule of Activities
Domain Entities	Activities, Manager, Activity on Node, Project, Critical Path, ES, EF, LS, LF, FLOAT

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to have a Gantt Chart displayed.	
2	System responds calculating the ES, EF, LS, LF, FLOAT for the Project then generates and displays the Chart.	
3	*Use case ends successfully.*	

Use Case: Generate Gantt Chart

Successful Outcomes: System calculates and displays a visual representation of the Project time table, it's interdependent sequence of Activities.

Use Case Properties:

Use Case Package	Data Analysis
ID	UC-DA-02
Use Case Goal	The Primary Actor has a Gantt Chart compiled and displayed for them.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	Primary Actor has an active Project with a logically consistent schedule of Activities
Domain Entities	Activities, Manager, Gantt Chart, Project, Critical Path, ES, EF, LS, LF, FLOAT

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to have a Gantt Chart displayed.	
2	System responds calculating the ES, EF, LS, LF, FLOAT, for the Project then generates and displays the Chart.	
3	*Use case ends successfully.*	

Use Case: Generate PERT Analysis Report

Successful Outcomes: System calculates and displays a visual representation of the important metrics associated with the Project complete with a probabilistic assessment of the likelihood of it finishing on time and on budget respectively.

Use Case Properties:

Use Case Package	Data Analysis
ID	UC-DA-03
Use Case Goal	The Primary Actor has a PERT Analysis Report compiled and displayed for them.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	Primary Actor has an active Project with a logically consistent schedule of Activities
Domain Entities	Activities, Manager, PERT Analysis

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to have a PERT Analysis Report displayed.	
2	System responds by calculating the relevant values for the Project then generates and displays the a report.	
3	*Use case ends successfully.*	

Use Case: Generate Earned Value Analysis Report

Successful Outcomes: System calculates and displays a visual representation of the important metrics associated with a Project complete with a probabilistic assessment of the likelihood of it finishing on time and on budget respectively.

Use Case Properties:

Use Case Package	Data Analysis
ID	UC-DA-04
Use Case Goal	The Primary Actor has a Earned Value Report compiled and displayed for them.
Actor(s)	Primary Actor: Manager
Level	User-Goal
Precondition	Primary Actor has an active Project with a logically consistent schedule of Activities
Domain Entities	Activities, Manager, Activity on Node, Earned Value

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to have a Earned Value Report displayed.	
2	System responds by calculating the relevant values for the Project then generates and displays the a report.	
3	*Use case ends successfully.*	

Use Case: View Assigned Activities

Successful Outcomes: Primary Actor accesses descriptions and deadline information of Activities' assigned to them.

Use Case Properties:

Use Case Package	Team Member
ID	UC-TM-01
Use Case Goal	The Primary Actor accesses the details of the Activities assigned to them.
Actor(s)	Primary Actor: Team Member
Level	User-Goal
Precondition	Primary Actor has successfully logged in to their Portfolio {UC-GF-01} and Primary Actor has at least assigned Activity.
Domain Entities	Team Member, Activity, Portfolio

Main Success Scenario:

Step	Action	Notes
1	Primary Actor selects an Activity to investigate from the list of all Activities assigned to them.	
2	System responds by presenting the details of the selected Activity.	
3	Primary Actor reviews information.	
4	*Use case ends successfully.*	

Alternative Flows:

4a. Primary Actor investigates another Activity.

Step	Action	Notes
4a.1	Primary Actor wishes to continue viewing Activities.	
4a.2	*Use case restarts at step 1.*	

Use Case: Login
Successful Outcomes: Primary Actor accesses their appropriate Portfolio.

Use Case Properties:

Use Case Package	Global Functionality
ID	UC-GF-01
Use Case Goal	Primary Actor gains access to their Portfolio
Actor(s)	Primary Actor: User
Level	User-Goal
Precondition	User has launched the PMA
Domain Entities	PMA, Portfolio

Main Success Scenario:

Step	Action	Notes
1	Primary Actor indicates intention to log in to Portfolio.	
2	System prompts Primary Actor for ID and Password.	
3	Primary Actor supplies login information.	
4	System verifies the Primary Actor and presents them with their Portfolio appropriate to their User type (i.e. Team Member or Manager).	
5	*Use case ends successfully.*	

Alternative Flows:
4a. Unrecognized User.

Step	Action	Notes
4a.1	System finds no matching ID and Password combination.	
4a.2	*Returns to Main Success Scenario step 2.*	

4b. Unrecognized User.

Step	Action	Notes
4b.1	System finds no matching ID and Password combination.	

Step	Action	Notes
4b.2	User aborts the program.	
4b.3	*Returns to Main Success Scenario step 2.*	

User Story Conversions:

User Story 1: As a project manager, I want to manage all my projects with the application.

1. The application will allow users to be of the type 'Project Manager'.
2. The application will support the creation of 'Project' entities.
3. The application will support 'Project Managers' having multiple 'Projects' in Project Portfolios.

User Story 2: As a project manager, I want to associate multiple interdependent activities with my project.

4. The application will support the creation of 'Activities' within a 'Project'.
5. The application will support the interaction of 'Activity' entities.

User Story 3: As a project manager, I want to assign resources (project members) and project-relevant properties to my activities.

6. The application will allow users to be of the type 'Team Member'.
7. The application will allow 'Managers' to assign 'Team Members' to 'Activities'.

User Story 4: As a project member, I want to be able to see which tasks have been assigned to me.

8. The application will allow 'Team Members' to view the 'Activities' to which they are assigned.

User Story 5: As a project manager, I want to generate GANTT charts.

9. The application will support 'Managers' entering real 'Time Line' details about the 'Activities' in their 'Projects'. (Start Date and Due Date)
10. The application will support 'Managers' entering projected 'Time Line' estimates to the 'Activities' in their 'Projects'. (Expected duration)
11. The application will allow the 'Manager' to enter 'Dependancy' information relating the 'Activities' in the 'Project' to one another, indicating which 'Activities' must be completed before another may begin.
12. The application will allow 'Managers' to generate 'GANTT Charts'. (Visual depiction of the 'Project's' projected progression)

User Story 6: As a project manager, I want to perform a critical path analysis.

13. The application will allow the 'Manager' to identify the series of 'Activities' in a 'Project' that constitute its 'Critical Path'.
14. The application will be able to calculate the sequence of 'Activities' whose dependencies are most critical and whose delayed completion would postpone the 'Project's' completion.

User Story 7: As a project manager, I want to be able to perform PERT analysis.

15. The application will support calculations to determine the 'Float' values of 'Activities', that is, the degree of flexibility (in terms of the preferred time scale) allowable in completing the 'Activity' without disturbing the 'Project's' completion schedule.
16. For every 'Activity' the application will encode the 'Earliest Start Time (ES)', 'Earliest Finish Time (EF)', 'Latest Start Time (LS)', and 'Latest Finish Time (LF)'.
17. The application will allow the 'Manager' to view graphical charts expressing the information derived from the inter-dependance of these values (in 'Activities') across the 'Project'.
18. For every 'Activity' the application will encode the 'Most Likely Time to Completion (m)', the 'Optimistic Time to Completion (o)', and the 'Pessimistic Time to Completion (p)'.
19. The application will derive the 'Expected Duration (t)', and 'Standard Deviation (s)' for each 'Project' from the '(o), (m), and (p)'' values.
20. The application will allow the 'Manager' to set a 'Target Date of Completion (T)' for each 'Project'.
21. The application will calculate a 'z-score', calculate the probability of a 'Project' meeting its '(T)' based on that 'z-score'.

User Story 8: As a project manager, I want to be able to perform earned value analysis.

22. The application will encode a 'Project Value (ProV)' for each 'Project' that will be calculated as the sum of the 'Planned Values (PV)' of each 'Activity' in the 'Project' that should have been completed by the time of the calculation.
23. The application will encode the 'Earned Value (EV)' of the 'Project' using the 'Percentage Technique'.
24. The application will encode the 'Budget at Completion (BAC)' as the sum of all the

'Planned Values (PV)' for the 'Activities' in the 'Project'.

25. The application will calculate the '(EV)' of each 'Project' by dividing the 'Planned Value' of the 'Project' by the 'Budget at Completion'.
26. The application will encode the 'Actual Cost (AC)' of a project as the total cost of all the 'Activities' completed up to the point of the calculation.
27. The application will compute the 'Project's' 'Percent Complete (PC)' value by dividing the 'Earned Value (EV)' by the '(BAC)'.
28. The application will encode the values 'Cost Variance (CV)', 'Schedule Variance (SV)', 'Cost Performance Index (CPI)', and 'Schedule Performance Index (SPI)'.
29. The application will calculate the 'Cost Variance (CV)' as 'Earned Value (EV)' - 'Actual Cost (AC)' with positive values indicating that the 'Project' is on 'Budget' and negative ones indicating the opposite.
30. The application will calculate 'Schedule Variance (SV)' as '(EV)' - '(PV)' with positive results indicating that the 'Project' is ahead of schedule, while negative values indicate that it is behind schedule.
31. The application will calculate a 'Project's' 'Cost Performance Index' as the '(EV)'/'(AC)' with results of value greater than one indicating that the 'Project' is likely to cost less than what was planned and conversely, values less than one indicating the opposite.
32. The application will calculate the 'Schedule Performance Index (SPI)' as the '(EV)'/'(PV)', again, with values greater than one being positive, and those less than one being unfavourable.
33. The application will encode the 'Estimate at Completion (EAC)' for each 'Project'.
34. The application will calculate the '(EAC)' as the '(BAC)'/'(CPI)'.
35. The application will calculate the 'Estimate to Complete (ETC)' as the 'Estimate at Completion (EAC)' - the 'Actual Cost (AC)' which will indicate how much is projected to be spent between the current moment and the completion of the 'Project'.

Business Rules:

PERT Analysis Rules:

Number	Label	PERT Analysis Rule	Calculation	Notes
PR01	ES – Earliest Start	The earliest time at which an Activity can start given that its preceding Activities must finish first.	MAX (Latest Finish) of preceding Activities	
PR02	EF – Earliest Finish	The Activity's earliest start plus its duration.	ES + D	
PR03	LF – Latest Finish	Latest time the Activity can finish without delaying the entire Project.	Calculated during the backwards pass.	
PR04	LS – Latest Start	Latest finish time minus the duration.	LF - D	
PR05	D - Duration	Length of time to complete the Activity.		
PR06	F - Float	Degree of flexibility for the duration of the Activity without effecting the Critical Path.		
PR07	CP – Critical Path	The sequence of Activities, that if any one were to be delayed, the Project would also be delayed.		
PR08	m – Most likely Time to Completion	Best guess at the remaining time to complete the Activity. Subject to revision.		
PR09	a – Optimistic Time to Completion	Wishful estimate of time to complete the Activity.		
PR10	b – Pessimistic Time to Completion	Pessimistic estimate of time to complete the Activity.		

PR11	T – Target Date of Completion	Manager set due date for a project.		
PR12	t – Expected Duration	Estimated duration of the task.	(a+4m+b) ÷ 6	
PR13	z - z-score	Calculated value, to be compared against a table to return a probability.	(b - a) ÷ 6	
PR14	P – Probability of meeting Target Date of Completion	End result of the	Found from the z-score lookup table.	
PR15	FPR – Forward Pass Rule	Used to determine the ES and EF of each Node in the Network. The rule sweeps forward through the Network, updating the values of each Node.	The ES of a given Node *n* is the MAX(LS) of all preceding Nodes upon which *n* depends.	
PR16	BPR – Backwards Pass Rule	Used to determine the LS and LF of each Node in the Network. The rule sweeps backward through the Network, updating the values of each Node.	The LF of a given Node *n* is the MIN(LS) of all preceding (*backward pass*) Nodes.	

Earned Value Analysis Rules :

Number	Label	Earned Value Rule	Calculation	Notes
EV01	PV - Planned Value (Project)	Sum of PVs of Activities in the Project.	Sum (PV_k) for all tasks k.	
EV02	PV - Planned Value (Activity)	Budget planned for the Activity based on the projected Resource costs.	Estimated by Manager.	
EV03	EV - Earned Value	Sum of PVs of Activities in the Project that have been completed at the time of analysis.	Sum (PV_k) for all tasks k that have been completed.	Uses the Percentage Technique
EV04	BAC – Budget at Completion	Sum of PVs of Activities in the Project that remain	Sum (PV_k) for all tasks k.	
EV05	PSC – Percent Scheduled for Completion	Measure of the amount of work that should be done by a certain time given the Project's estimates.	PV(Project) ÷ BAC	
EV06	AC – Actual Cost (Project)	Actual tally of expenditures incurred by completed Activities.	Sum (Debts against Completed Activities)	
EV06	AC – Actual Cost (Activity)	Expenditures incurred by the Activity.		
EV07	PC - Percent Complete	Percentage of the Project completed at a given time.	EV ÷ BAC	
EV08	CV – Cost Variance	Difference between what should have been paid for completed work and what actually was paid for that work.	EV - AC	
EV09	SV – Schedule Variance	Difference between the value of completed work and value of the planned work.	EV - PV	

EV10	CPI – Cost Performance Index	Earned Value over Actual Cost. Values over 1 indicate the cost has been less than projected.	EV ÷ AC	
EV11	SPI – Schedule Performance Index	Earned Value over Planned value. Values over 1 indicate the Project is ahead of schedule.	EV ÷ PV	
EV12	EAC – Estimate at Completion	Indicates the projected cost of the Project at completion.	BAC ÷ CPI	
EV13	ETC – Estimate to Completion	Indicates projected cost overrun or underrun.	EAC - AC	

Non Functional Requirements:

The application must be intuitive to use and the interface must be clean and consistent.

Design Constraints:

The application must be written in Java using the Swing library and the Database must be done using SQLite. Testing is to employ JUint.

Glossary:

Business Entities:

Business Entity	Description
PMA	Project Management Application.
Project Manager	An application user with the abilities to create and manage projects and activities.
Team Member	An application user to whom activities are assigned.
Project Portfolio	A collection of projects.
Activity Portfolio	A collection of activities.
Resource	Another term for a team member.
Project	A collection of ordered activities.
Activity	A task to be completed in the larger setting of a project.
Project ID	A unique identifier for a project.
Activity ID	A unique identifier for an activity.
Project Name	A name chosen by the manager for their project.
Activity Name	A name chosen by the manager for their activity.
Project Description	A description given by the manager for their project.
Activity Description	A description given by the manager for their activity.
Cost	The actual monetary expenditure of an activity or project.
Schedule	The sequence of activities required to complete a project. These activities each have projected durations allowing the project to gauge wether it is progressing on target or not.
Critical Path	The sequence of activities for which if any one finishes later than their latest finish, the entire project will be delayed.
Gantt Chart	A visual depiction of the dependency driven schedule of a project.
PERT Analysis	A complex analysis tool for projecting project cost overruns and schedule delays. More information can be found here.
Earned Value Analysis	A tool for measuring the progression of a project throughout

	its execution. More information can be found here.
Activity on Node Graph	A visual representation of the dependancy chain of activities in a project.

References:

Martin Fowler *UML Distilled: A Brief Guide to the Standard Object Modelling Language*.
Rodger S. Pressman *Software Engineering: a Practitioner's Approach*
Daniel Sinnig PhD: *Lecture Slides, COMP 354*

Wikipedia
http://en.wikipedia.org/wiki/Program_evaluation_and_review_technique_(PERT)
http://en.wikipedia.org/wiki/Gantt_chart