Strategy - Project Planning System

SRS Document

Disclaimer

This Software Requirements Specification document is a guideline. The document details all the high level requirements. The document should be used as a guideline by the students to design the Solution Architecture for the project. The document also describes the broad scope of the project and high level logical object model. But while developing the solution if the developer has a valid point to add more details being within the scope specified then it can be accommodated after consultation.

Project Planning System - Strategy

Introduction

The purpose of this document is to define scope and requirements of a Project Planning System for service delivery officers, project managers and team members. The proposed system will carry out automated project planning, scheduling, tracking, and will facilitate resource assignment.

This document should be used by the development team to architect solution and subsequently develop, code and test the same.

Management Summary

Foundation of any successful project is an effective project plan. While simple project plans can be created manually, it is a tedious, laborious and an error prone task. With increasing number of tasks, the complexity of manual project scheduling, critical path determination and slack computation grows exponentially. Therefore, it is critical to have an automated tool to carry out project planning task.

To address the above challenge, an effective project planning system is envisaged. The proposed system - Strategy will deliver automated project planning, scheduling, tracking mechanisms; it will also facilitate resource assignment.

Strategy will be a web-based system and will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

Key Assumptions

To ensure timely completion of the project, following simplifications must be considered:

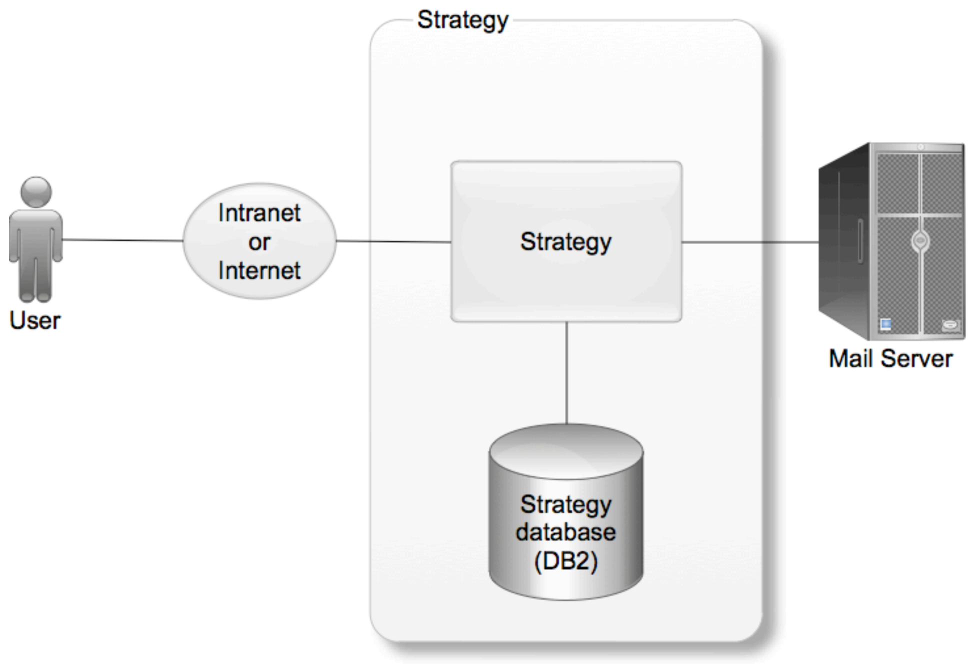
1. While there are many task dependencies like *Finish - Start*, *Start - Finish*, *Finish - Finish*, and *Start - Start*, but for this project only **Finish - Start** task dependency will be considered.
2. There are no circular task dependencies and therefore no such checks are required to be implemented.
3. One person can work on only one task at a time; or in other words a person is assigned to a task at 100% utilization level. On assignment, person gets blocked for the project duration.

Note: In reality one person may work on multiple tasks by allocating his time between them. Also the blocking may accordingly occur dynamically for the task duration.

High Level Architecture

Strategy's high level architecture is illustrated through the context diagram shown in figure 1 below. It will have following categories of users:

1. Service Delivery Officer (who create new projects and assigns to project manager)
2. Project Manager (who creates a project plan for the assigned projects)
3. Team Member (who is performs the assigned task(s) for a project)



Strategy Context Diagram

Strategy	The proposed web-based application, Strategy will be accessed by the users of a project organization. It will perform automated project planning, scheduling, and will facilitate resource assignment. It will be a web based system to allow unified access across project locations.
Strategy Database	This will hold all the Strategy data including the projects, their tasks, project organization users and their details, etc.
Mail Server	It is responsible for delivering notification e-mails to Strategy users.

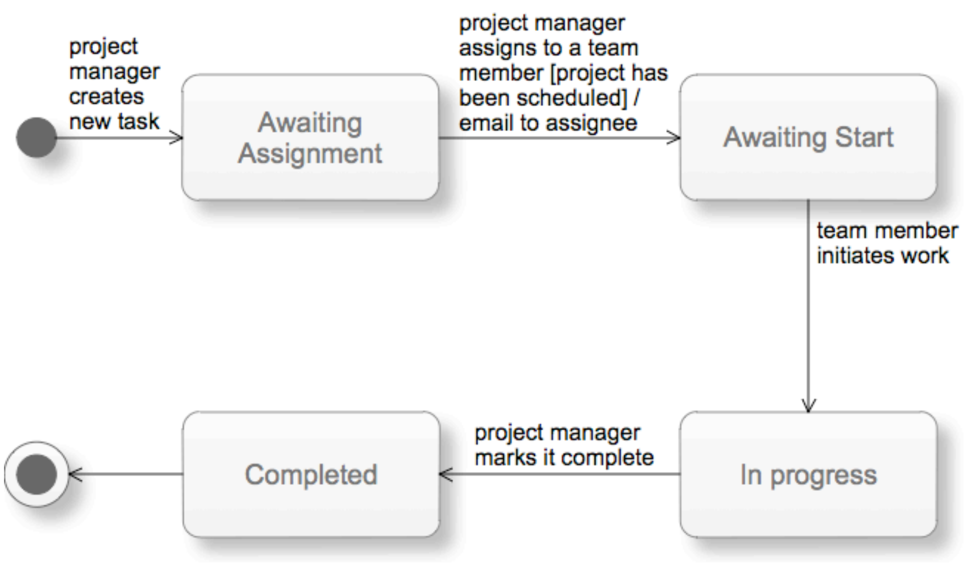
Functional Requirements

The high level functional requirements for the Strategy system are outlined in the State and Use Case diagram described in this section.

Strategy will provide a secure user-id/password based secured login mechanism to access its services. The details of this are not outlined here. The development team is expected to create these keeping in mind the general practices followed by the web applications.

State Diagram

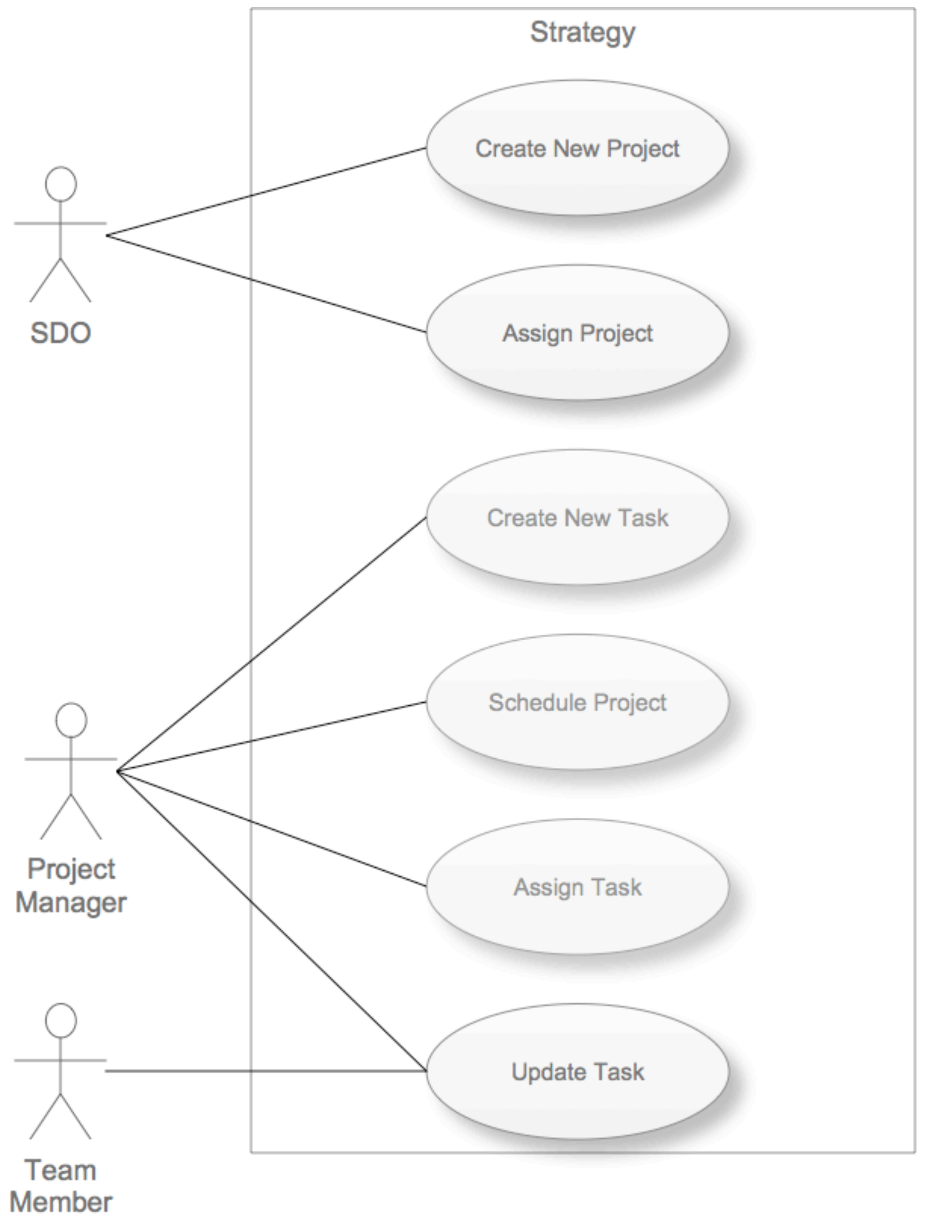
The following figure outlines the state diagram for a task in the system.


project manager creates new task
Awaiting Assignment
project manager assigns to a team member [project has been scheduled] / email to assignee
Awaiting Start
team member initiates work
In progress
project manager marks it complete
Completed

1. Project manager creates a new task, whose initial state is “awaiting assignment”.
2. Once a “awaiting assignment” task is assigned a team member on the basis of skill needed for the task, its state is transitioned to “awaiting start” and the assignee is informed via an e-mail. However, a task can be assigned only after a project has been scheduled to determine all planned dates (and slack etc).
3. The assigned team member initiates the work on task, by updating its actual start date. The task now transitions to “in progress state”.
4. Project manager marks a task completed by updating its actual completion date in consultation with the assigned team member.

Use Case Diagrams

The following figure illustrates the Use Case diagram for the system.



Use Case Diagram

Please note that the master maintenance use case diagram is not explained here as they are very simple and straight forward. There are 3 scenarios viz. skill, role, and user master maintenance. Please refer to section on Logical Object Model to build a detailed design for the same. An administrator will be responsible for maintaining all the masters. To simplify the development, you may only complete the DB design and insert test data in the required master tables directly using DB2.

Login as a separate use case is assumed to be included in other use cases. Other simple use cases like **view tasks** (*by project*, *assigned to me*), **view projects** (*assigned to me* for project manager, *working on* for team member who has one or more task assigned in a project), **view critical path** (tasks of a project who have *zero slack*) are not listed. Such use cases can be implemented by developing appropriate SQL query and displaying the results via JSPs.

Use Cases

Create New Project

Use Case Element	Description
Number	UC.01
Application	Creation of a new project by the SDO
Use Case Name	Create New Project
Primary Actor	SDO User
Secondary Actor	None
Pre-condition	None
Trigger	SDO clicks on the **Create New Project** link on the home page; this menu item/ operation is accessible to the users with SDO role only
Basic Flow	• System displays the new project creation form. • User enters the required information and clicks save button. The new project record is saved.
Alternate Flow	• If the cancel button is clicked, the data is discarded. No record is created.
Output	None

Assign Project

Use Case Element	Description
Number	UC.02
Application	A newly created project is assigned to a project manager for execution.
Use Case Name	Assign Project
Primary Actor	SDO User
Secondary Actor	None
Pre-condition	None
Trigger	SDO clicks on the **Assign Project** link on the home page; this menu item/ operation is accessible to the users with SDO role only
Basic Flow	• System presents a list of all projects that have not been assigned a project manager so far; SDO clicks on the project s/he wishes to assign a project manager. • System displays the list of project managers; SDO selects the project manager s/he wishes to assign to the project. • SDO clicks on save. Project record is updated and project manager is assigned. A mail to the assigned project manager is sent informing him/her of the assignment.
Alternate Flow	• If the cancel button is clicked, the operations is cancelled.
Output	e-mail to the assigned project manager.

Create New Task

Use Case Element	Description
Number	UC.03
Application	New task creation by the assigned project manager for the project.
Use Case Name	Create New Task
Primary Actor	Project Manager
Secondary Actor	None
Pre-condition	None
Trigger	Project Manager clicks on the **Create New Task** link on the desired project page; this menu item/operation is accessible only to the user with project manager role and have been assigned the opened project.
Basic Flow	• System displays the new task form is displayed. • Project Manager fills the new task form with necessary details like task description, duration, dependencies (if any), and skill required etc. • Project Manager clicks on save button; the data is saved and a new task record for the selected project is created.
Alternate Flow	• If the cancel button is clicked, the operations is cancelled. • In even of error while saving due to field level validations failure, appropriate error message(s) is displayed; form remains open for correction and subsequent save (or cancel).
Output	Error Message in the event of validation failure(s)

Schedule Project

Use Case Element	Description
Number	UC.04
Application	Automatic scheduling of project by the assigned project manager to determine earliest/latest start/finish dates for all the tasks.
Use Case Name	Schedule Project
Primary Actor	Project Manager
Secondary Actor	None
Pre-condition	None
Trigger	Project Manager clicks on the **Schedule Project** link on the desired project page; this menu item/operation is accessible only to the user with project manager role and have been assigned the opened project.
Basic Flow	The project is scheduled to determine the earliest/latest start and finish dates of each task by traversing the network forward and backward. Each task and the project records are updated accordingly.
Alternate Flow	None
Output	None

Assign Task

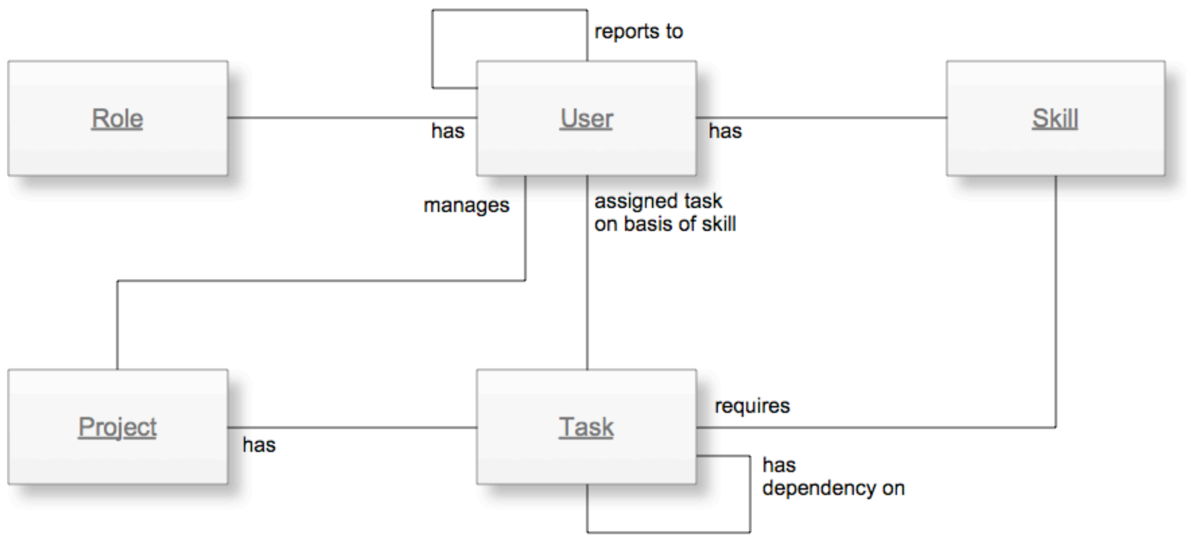
Use Case Element	Description
Number	UC.05
Application	Task assignment by the project manager to a team member.
Use Case Name	Assign Task
Primary Actor	Project Manager
Secondary Actor	None
Pre-condition	Project has already been scheduled
Trigger	Project Manager clicks on the **Assign Task** link on the desired project page; this menu item/operation is accessible only to the user with project manager role and have been assigned the opened project.
Basic Flow	• System presents a list of all tasks that have not been assigned a team member so far; project manager clicks on the project s/he wishes to assign a team member. • System displays the list of team members who have task's skill and are available for the project duration; project manager selects the team member s/he wishes to assign to the task. • SDO clicks on save. Task record is updated and team member is assigned. A mail to the assigned team member is sent informing him/her of the assignment.
Alternate Flow	• If the cancel button is clicked, the operations is cancelled.
Output	e-mail to the assigned team member

Update Task

Use Case Element	Description
Number	UC.06
Application	To update the task status - started by team member and completed by project manager.
Use Case Name	Update Task
Primary Actor	Project Manager
Secondary Actor	Team Member
Pre-condition	Task is already assigned a team member
Trigger	Project Manager clicks on Mark Task Completed link on the task detail page of his/her project; or Team Member clicks on Mark Task Initiated link on his assigned task's detail page.
Basic Flow	Task state is updated & saved according to the button clicked.
Alternate Flow	None
Output	None

Logical Object Model

A high level logical object model of the system is shown below. During technical design it will be transformed into a physical model covering all system entities. Such a diagram will include their relationship and its cardinality.



Logical Object Model

1. Each user has a role and one or more skills. User is assigned one task at a time; in other words, user dedicates 100% of his/her time to perform the task at hand. S/he can not initiate another task until task at hand is completed.
2. Roles are service delivery officer, project manager, and team member.
3. Team member(s) reports to a project manager. Project manager(s) report to a service delivery officer.
4. Project manager manages one or more project(s). A project has a description, planned & actual start date, planned & actual finish date.
5. Each project has many tasks. Task may depend on one or more tasks. The dependency is always assumed to be "Finish - Start". Each task has a planned & actual start date, and planned & actual finish date. Planned dates will be of two types viz. earliest and latest. Each task requires a skill to perform it. Task also has a milestone flag.
6. User (team member) needs to have a skill required by a task in order to perform it.

Database Design Guidelines

This involves the transformation of the use cases, state diagrams, and logical object model into detailed and optimized physical database table designs.

Typically persistent classes will map to table(s) with their attributes as columns of the table. In some cases a high level object may map in to a master-child table. Invoice is one such example where it maps in to "invoice_header" and "invoice_line_item" table.

Associations between two persistent objects are realized as foreign keys to the associated objects. A foreign key is a column in one table that contains the primary key value of the associated object.

Similarly, a standard technique in relational modeling is to use an intersection entity to represent many-to-many associations. Following is a broad checklist for physical database database design:

1. Database must be properly normalized except those instances where de-normalization help improves performance. This option must be used with special care.
2. All persistent classes that use the database for persistency must map to database structures.
3. Many-to-many relationships must have an intersecting table.
4. Primary keys should be defined for each table, unless there is a performance reason not to define a primary key.
5. Indexes should be defined to optimize access.
6. Data and referential integrity constraints should be defined.

Testing Approach

Quality of the software can be achieved with basic hygiene and consistency followed during design and development of User Interface(UI), Navigation, Validations as per the business process requirement.

To ensure the project delivers acceptable quality to the customer, its important to create a checklist of the conventions to be followed across. Common checks as below are for your reference during design and development:

Common Checks	Validation Type
Page Title is valid for the feature being provided on the page	UI
Order of the Data Entry Fields is logical as per the functionality being provided by the feature	UI
Order of the Display only Fields makes viewing and understanding easy for the user	UI
Spellings and Correctness of Label for the Data Entry and Display fields	UI
The labels are not wrapping onto another row thereby adding a blank row on the page	UI
The fields with drop down are displayed in single row instead of drop down coming on the next row	UI
Data Entry field basic validations are working i.e Text field /Numbers / Dates allow data for their type only	Functional
The dates are following a standard format dd/mmm/yyyy on all forms	UI
The color scheme of all forms i.e headers labels , alerts, entry fields are uniform throughout the application	UI
The action buttons for a New Data Entry Form are uniform for all forms that is allowing data entry	UI
The action buttons are performing the desired action e.g. "submit" is creating a new record if there are no errors and recording all the input fields, whereas 'cancel' is not creating a new record in the database	Functional
The links provided on the forms are opening correctly.	Functional
The data feed mechanism for Read and Write files is generating a log with count of entries.	Navigation

Suggested Technical Reading

The project is aimed at making the student understand concepts of Design and Development using IBM Rational tools, Web Sphere Application Server and DB2 Database. The following reading reference is easy to understand and should be read to get a clear understanding of capabilities of the tools and how you would leverage them to execute a project.

Technical Reference	URL to access
RAD - Tackling challenges of software development with Rational Application Developer for WebSphere Software	http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html
IBM Education Assistant - Rational Application Developer 7.5	http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html
RSA-Overview of Rational Software Architect for WebSphere Software Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html
Using the new features of UML Modeler in IBM Rational Software Architect Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html
Rational Technical Library	http://www.ibm.com/developerworks/rational/library/