

CENG 491

D&D SOFTWARE

**SYSTEM REQUIREMENTS
SPECIFICATION REPORT**

Prepared by:

Firat Alpergin

Dogan Yazar

Tuncay Namli

Mehmet R. Dogar

TABLE OF CONTENTS

OVERVIEW.....	2
1. INTRODUCTION.....	2
1.1 PROBLEM DEFINITION	2
1.2 GOALS & OBJECTIVES	2
1.3 STATEMENT OF SCOPE.....	4
1.4 SOFTWARE CONTEXT	4
1.5 MAJOR CONSTRAINTS	5
1.6 HARDWARE & SOFTWARE REQUIREMENTS	5
1.7 USER & LITERATURE SURVEY	6
2. USAGE SCENARIO	9
2.1 USER PROFILES	9
2.2 USE-CASES.....	9
3. DATA MODEL AND DESCRIPTION	14
3.1 DATA OBJECT DESCRIPTION	14
3.2 DATA RELATIONSHIP DESCRIPTION	19
3.3 ENTITY RELATIONSHIP DIAGRAM (ERD)	23
3.4 DATA DICTIONARY	24
4. FUNCTIONAL MODEL AND DESCRIPTION	27
4.1 DESCRIPTION OF FUNCTIONS	27
4.1.1 Process Specification (PSPEC)	27
4.1.2 Functional Modeling	30
5. BEHAVIORAL MODEL AND DESCRIPTION	37
5.1 DESCRIPTION FOR SOFTWARE BEHAVIOR.....	37
5.1.1 States	37
5.1.2 Events	39
5.2 STATE TRANSITION DIAGRAM (STD)	42
5.3 CONTROL SPECIFICATION (CSPEC).....	42
6. PROJECT ESTIMATIONS, PROCESS MODEL & TEAM STRUCTURE.....	45
6.1 PROJECT ESTIMATIONS	45
6.2 PROCESS MODEL & TEAM STRUCTURE.....	47
7. RISK MANAGEMENT PLAN	48
7.1 RISK MANAGEMENT ORGANIZATIONAL ROLE.....	48
7.2 RISK DESCRIPTION	48
7.3 RISK TABLE	49
8. VALIDATION CRITERIA.....	49
9. PROJECT SCHEDULE	50

OVERVIEW

In this document, the software requirements specification of DProject can be found. In the Introduction section, the problem definition, goals & objectives and statement of scope of DProject is defined. Also the role of DProject in the software context, the hardware & software requirements of it are present. The major constraints on the project and the user & literature survey carried out can also be found in this section. The Usage Scenario section provides a usage scenario for DProject and organizes information collected during requirements elicitation into use cases. In the Data Model and Description section, the information domain of DProject is described. The Functional Model and Description section describes each major software function, along with a data flow hierarchy. The Behavioral Model and Description section presents a description of the behavior of DProject. The Project Estimations, Process Model & Team Structure section consists of the estimations made about DProject and also the process model and team structure strategy to be followed throughout the project. The Risk Management Plan Section describes the RMMM activities that are to be followed throughout the project. In the Validation Criteria section, the approach to software validation is described. The Project Schedule Section consists of the schedule that is to be followed throughout the project.

1. INTRODUCTION

1.1 PROBLEM DEFINITION

The aim of this project is to develop a project management tool whose ultimate goal is to ease the planning, management and supervision of a project. The tool lets the users of it (who have different access rights) to achieve task, meeting and resource management of their projects effectively; provide them features like statistics & report generation, export/import facilities, scheduling; it lets the users communicate effectively throughout the project by using features like forums, e-mail sending and notifications and it also makes it possible for the clients to see the overall progress of the project they are purchasing. The tool offers a multi-user and online environment so that it is largely available to all of its users.

The main functionalities, goals and objectives of the product can be found in the section that follows.

1.2 GOALS & OBJECTIVES

The ultimate goal of the project management tool, as stated earlier, is to ease the planning, management and supervision of a project. To be able to achieve this goal, the tool offers a lot of features to the users of it. The main objectives of the system can be stated as follows:

- The tool offers a multi-user environment which can be accessed online securely. By this way, the users of the system can keep track of the project progress as they want and different projects of different companies can be managed by the tool

simultaneously. The users of the system will be provided usernames and passwords so that the system can work securely.

- The tool lets the users of it be provided different access rights so that the hierarchical structures seen in real life projects also appear in it. For example, the project manager will have more sophisticated features than a developer in the project. Every member of the project will have different access rights depending on his/her place in the hierarchy.
- The tool lets its users communicate effectively throughout the project. To be able to achieve this, it will provide features like sending e-mails and notifications and also provides forums to its users so that the team members can communicate among themselves about any subject concerning them. The users are also provided the opportunity to send file attachments among themselves.
- The tool lets its users handle task management effectively. The users have the opportunity to create and assign tasks (if they have the proper rights, of course) with different priorities, keep track of the progress of tasks, the dependencies among the tasks and add comments and notifications about the statuses of tasks. All the tasks that are to be completed throughout the project can ultimately be examined and be kept track of by the task management features offered by the tool.
- The tool lets its users handle meeting management effectively. The users with enough access rights have the opportunity to arrange meetings and the tool will greatly help them in fixing the details of the meetings. The arranger of the meeting has the chance to provide different options to the possible attendants by using notifications and depending on the responses of them, the tool lets the arranger fix the details like the date, place and the attendants of the meeting.
- The tool lets its users handle human and resource management effectively. It will let the users define new materials and assign them to different projects. All the necessary information about a material is kept and managed in the system. The system will also keep track of the user salaries and wages and keeps track of the project management.
- The tool provides features for statistics and report generation. It will let the users define and use different filters (which can be saved for later use) for statistics and report generation. This feature makes it possible to keep track of several aspects of the project such as the time spent by each member on any task, the total amount of effort spent on a particular issue, etc.
- The tool provides features for importing and exporting reports. The reports can be exported in different formats such as XML, excel, etc. There will also be the importation and exportation of a project as a whole from one project to another.
- The tool provides features for project scheduling. It will have features for automatic Gantt chart construction and provides project calendars to its users. It will let the users examine the start and end dates for finished tasks, the time spent on a particular task and the statuses of unfinished tasks (when it is supposed to finish, how much of it is completed) with a graphical demonstration using different time periods (weekly,

monthly, etc.) so that the planning of the project can be achieved much more efficiently.

- The tool provides features for the clients of the project to see the overall progress of the project they invest money in. The clients will be able to see the necessary parts of the project so that they can keep track of the progress as they wish.

1.3 STATEMENT OF SCOPE

The following general requirements were laid of for our project named Dproject:

- A way that Dproject could define a new company to system with a default administrator.
- A way that administrator could add new users to system.
- A way that in which users could create new projects.
- A way that in which users could create new tasks for projects.
- A way that in which users could assign tasks to other users.
- A way that in which users could determine the dependencies between tasks.
- A way that users could handle resource management for projects.
- A way that users could assign materials to tasks
- A way to follow and report users working time on work items.
- A way to import/export project to some external files.
- A way to handle communication among users of the company
- A way that in which users are notified from the actions done in system.
- A way that users could visualize the statistics and reports by charts.

Staff members of Dproject have requested a lot of interface enhancement that will increase the usability of the product for staff.

1.4 SOFTWARE CONTEXT

All the current project management tools share a lot in common. Being one of the most popular software systems in the market, all the project management tools provide some basic and crucial features to its users. Our tool (DProject) also has these basic properties and some extra ones which make our tool more appealing to the customers.

First of all, DProject's online nature is one feature that distinguishes it from the others. One of the most important features of a project management tool is easy access and an online access is the ultimate solution to this problem. DProject is also a multi-user system, which is a feature that is shared by most of the project management tools in the market. Dproject provides basic features like task management, communication means for project members, project schedule, statistics and report generation, resource management. These basic properties are enough for the average customer and they will be satisfied with them. But DProject provides some other features that are designed for the sophisticated customer. It includes a meeting management system, which is much more detailed and different from the other project management tools. The meeting management system of DProject handles nearly

all the operations that are needed for arranging meetings and fixing meeting details. This extensive meeting management system will surely be highly praised by the customers. DProject also has some innovative features about import & export features. These include the importation or exportation of a project as a whole from one DProject system to another and also some more sophisticated import & export features in a various number of formats. These high level features will surely make the more sophisticated customers satisfied with DProject.

We believe that, DProject will make a large number of customers with different backgrounds and experiences satisfied because it consists of features that are appealing for different user types. It provides all the basic features that project management tools have and also some new ones that are lacking in most of them.

1.5 MAJOR CONSTRAINTS

There are two major constraints imposed on this project:

1.5.1 Time

The deadline of the project is strict and is no subject to change. The project is to be completed in about 2 months from now. Since no delay in the delivery date is permitted, the scheduling of the process should be carefully made and the people working in the team should obey the schedule as much as possible. In case of a risk of missing the deadline, some parts of the system may not be implemented (the ones that are not that essential) and the product may be delivered consisting of the essential functions.

1.5.2 People

The number of people working in the project is also fixed (3 people). To accomplish all the intended functionalities of the system, the work breakdown should be done carefully and the schedule should be tracked precisely by each member of the group. The deadlines for the milestones are exact and since the delay of one member will effect the others' completion too, all the members should work with responsibility and do the best of them in the intended time.

1.6 HARDWARE & SOFTWARE REQUIREMENTS

Throughout the project, our hardware and software requirements are as follows:

- **Hardware Requirements**

- Web Server: Depends on how many users will use this software management tool but in the development phase we will use an ordinary PC (e.g. with Pentium 4 CPU and 512 MB RAM).
 - For Development: We need 4 PCs with minimum configuration of P4 1.4 CPU and 256 MB RAM.
 - Client-Side: A computer with an internet/network connection.
- **Software Requirements**
 - Web Server: We will use Apache web server with Tomcat Servlet container. That is because, Apache is secure, stable, open source and it's widely used all over the world.
 - DBMS: Our options are Oracle, MySQL, DB2 and MSSQL Server. For now, we are planning to go with MySQL since it's open source and widely used.
 - Development Enviroment: We will use Java technology since it's very powerful, scalable and cross-platform. Also, it provides excellent APIs like JDBC and Swing. More precisely, we will use Servlet technology of J2EE platform to implement our project.
 - Client-Side: Any internet browser working on any environment.
 - Others: For documentation we will use MS Office or Open

1.7 USER & LITERATURE SURVEY

Large amount of Project Management Tools that exist in the market includes main project management concepts and features which are very similar to each other.

Although many of them try to approach to the problem in an innovative way, only few can achieve this. Therefore, we decided to do a large survey to understand the problem and current solutions which will give new ideas about our project. We divided our survey in two parts; interviewing with customers and examining current project management tools.

Customer Meetings

We had contacted an employee, Faysal Basci from Meteksan Sistem, who works on some big projects as a project manager. We sent an e-mail to him which includes some questions about project management, the tool they use and what features they need mostly

which does not exist in the current tool they use. The reply consists of four parts; task assignment, statistics, feedback management, and keeping time records for users.

Firstly, he mentions about task management and task categorization. He states that tasks must be categorized as demo, bug, improvement, etc. and some access rights must exist for task assignment. Tasks may consist of sub-works which are the required tasks to complete the main task and explanation of these (like document updates or code updates).

Secondly, he talks about the importance of statistics and graphical interfaces for showing these statistics. Effort ratios and effort distribution statistics are some examples that he gives. He also specifies that employee cost factor, type of employee (developer, debugger, etc), time that employee is trained must be kept for users to use in statistics.

Another feature that he points out is the feedback management which includes customer, tester and developer feedbacks. He thinks that this feature must be managed by project manager and includes some scenarios that feedbacks are evaluated, processed and archived.

Finally, keeping time records is the concept that he underscores mostly. He states that for all users, the time that user spent on tasks and time spent for specific tasks must be recorded since these are very important for statistics and future estimations.

After this e-mail interview and examining it, we arranged a meeting with a developer, Murat Yükselen, who works for SRDC. In this interview, he states that meeting arrangement is the most important problem that is frequently faced during the project management process. He also proposes a solution in which meeting arrangement is assigned to software. This solution contains some e-mail notifications and deciding algorithm. We discussed about the solution and decided on a meeting arrangement feature and its requirements.

Current Tools

We searched many current project management tools and tried to find the best ones to analyze deeply. We decided that aceproject and MSProject are the tools that include good designed main features and some innovative ideas. We try to explain main features and new approaches of these tools as below.

1. Aceproject

We can divide the main features of Aceproject as below:

-Task Management: This feature is one of the core concepts of all project management tools. Mainly task is created and assigned to other users. Firstly, user determines general information like summary, details, estimated time and id of the task. Also he specifies the status, group, type, priority, start and due-date of the task. Then he determines the reviewers and users that he assigns to. Aceproject's tasks also have comments, attachments and history trails. Users can view task lists in their account. These lists are formed according to their access rights. They have the user-friendly interfaces like incomplete tasks, complete tasks, my-tasks, file attachments for accessing the proper views easily. In addition to these interfaces, they also include filters that users can apply and form the view they want.

-Statistics: Statistics is the feature that allows some users to view some numerical values of projects according to some criteria. General statistics, user statistics, task group statistics, task type statistics, status statistics and priority statistics are the throughputs that aceproject provides.

-Reports and Charts: These features provide an easy way to create time reports, task reports and drawing some charts about projects. Aceprojects 'task report' feature allows user to view tasks according to related user, description and dates. In addition users can specify the data, the start and due-dates for project or tasks, select weekly or monthly option and generate the report by the Time Reports feature. Aceproject also gives a Gantt chart view to users for graphical showing of tasks and their start and due dates.

-Time sheets and approval: This is the most important feature for aceproject that distinguishes it from other tools. Users can create their weekly time sheets by selecting the project and the week. Time sheet includes the task, some of its properties and total hour worked on. This data can be entered by user or by the IN/OUT feature of the tool. An IN/OUT item is used to log your time automatically for a task. When you close the IN/OUT, the hours elapsed are added to your time sheet for that task. After creating time sheets, users can send these to the project managers for approval. Project managers can approve or reject these time sheets and if he rejected, elapsed times will not involved in statistics and reports. Also users and managers can generate reports from these time sheets.

-Access rights: Aceproject specify some access rights to simulate the hierarchy of projects and to restrict users to their rights. These rights for users are mainly divided into two parts; Administrator and Normal. These two has some interface specific to them. Time sheet access rights are about time sheets and permit users to create their time sheets by manually,by IN/OUT or both. User Directory access rights allow the administrator to specify what will appear on each user directory. Project access rights are classified into Project Manager ,Time Approval, Project Access and Edit Level Tasks which also includes many leveled rights.

-Calendar, Forum, Export and E-mail notification: Calendar shows the deadlines and start dates of tasks and offer a graphical view for tasks. Forum keeps the conversation between company users to discuss their problems. Export interface enables users to export their generated reports to MSExcel. E-mail notification system provides the communication between the system and user when the user is not logged in. This system helps the system to notify the user for some actions done in the system.

2. MSPProject:

In addition to these features, MSPProject provides some extra features for project management discussed as below:

-Task dependency: MSPProject enables users to link two tasks according to their dependencies. These dependencies are listed below;

F-S Predecessor finishes and the other starts

S-F Two tasks must start at the same time

F-F Two tasks must finish at the same time

S-S Start of the predecessor determines when the other starts

Also users can view critical tasks according to these dependencies.

-Resources and Cost: MSProject divides resources into two parts; Work Resources and Material resources. Material resources are supplies and stocks that are needed for the project. MSProject allows users to keep track of material resources and assign them to tasks. Work resources are people and equipment that works on tasks. For cost calculations, project multiplies the numbers of hours a resource works and hourly cost rate to give you the resource cost rate.

-Publishing projects on WEB and IMPORT/EXPORT: In MSProject user can save a project as html document that can be viewed on any browser. Also it supports some file formats to export and import (Project Database *.mpd, MSAccess, MSExcel, Hypertext Markup lang. *.htm)

2. USSAGE SCENARIO

2.1 USER PROFILES

The system must have three different user profiles that distinguish the objectives and rights of users. We can classify these users as:

- Administrator
- Normal User
- Client

2.2 USE-CASES

1) Administrator:

Administrator has all the rights which include the rights of Normal User and some administrative rights to manage the system for company. The system can have more than one administrator. Administrator scenarios must include editing company info, editing some preferences of company, creating / deleting / editing projects, importing / exporting projects, handling resource management of company, creating / deleting / editing user info, viewing all projects and tasks, viewing statistics, generating time reports, generating task reports, generating charts for statistics and tasks and handling time approval. We can divide these scenarios to Administrative scenarios and Normal scenarios (owned by normal users). In this part we only clarify Administrative scenarios and capabilities, others will be explained in Normal User scenarios and the slight difference will be underlined there.

1.1 Editing Company Info

Administrator should be allowed to edit company name, company preferences, contact information, and company address. Company preferences should include selecting time zone, date format and week management. Date format is the format that every user in company views the dates as selected. In week management admin can choose the start of the week.

1.2 Preference of Time Types & Project Types

In this scenario, admin can add new time types to default ones (Regular, Training, Overtime) and some project types. The project type is used for categorizing the projects in company.

1.3 Handling Project

1.3.1 Creating Project

Firstly, admin must supply some general information to the system. This information should include name, description, project type, priority, start-due dates, budget and contact information. Also he can select some project preferences as enabling task history, enabling e-mail notification and some default values for tasks. In addition to these, the system must save the creator and creation date.

After creation of the project administrator can assign users to the project, define new task groups, define new task types, define new task statuses, and define new task priorities for the project. These will be clarified in Normal User profile since Normal user can be given the necessary rights for these scenarios.

1.3.2 Deleting Project

For this operation admin must select if the entire project or some parts are deleted. These parts must be classified into groups and after selection, they must be deleted from the system.

1.4 Handling Users

1.4.1 Creating New User

The scenario starts with the supplying account information for the new user. Account information includes the user_id and password information. Admin also can specify the user's contact information which should include name, phone and e-mail information. Access rights and user preferences are other important information that can be supplied by administrator. User profile (client, normal, admin), time approval type and user directory restriction are the general access rights that can be specified by the administrator for the users. Admins also should determine the initial preferences for the user that implies specifying email notification cases and other scenarios attribute preferences. The system must store the user creation time for every new user and show this information to admin.

After creation of user, administrator also can assign user to projects and tasks which will be deeply cleared up in normal user scenarios.

1.4.2 Deleting User and Deleting User Items

Admin must have the right that enables him to delete user completely or delete some items of user. User items must be categorized clearly for selecting and deleting them properly.

1.5 Import/Export Projects and Export Reports

This capability makes the projects transportable between separate systems. Administrator should have the right that enables him to select a project and export it to a file

that can be used later to construct the same project again. Also administrator is able to select a report and export it to files with external formats.

2) Normal User

The system must provide a Normal User profile that can be easily categorized by giving some different access rights to the profile. This can be done by specifying all access rights and corresponding capabilities clearly.

2.1 Project Editing

The user must have the Project Manager right to edit the projects. We can divide the project editing into below categories:

2.1.1 General Information Editing

In this scenario project manager can edit the information that administrator supply when creating the project.

2.1.2 Assigning Users

Project manager can assign users to project and specify the rights of the user in project. These rights are all explained access rights for normal user which are giving project manager right, time approval right, project access right, and specifying task editing level right. After this assignation, the system must inform user about this assignation which must be by e-mail or some other option.

2.1.3 Creating Task Groups

Project Manager should be able to determine some task groups for the project to add tasks to these groups. Any task that does not belong to any task group is added to general group.

2.1.4 Creating Task Types

This scenario gives project manager the capability to define new task types for categorizing the tasks for his project.

2.1.5 Creating New Status

Project manager should be able to define new task statuses that are used for specifying the status of the task.

2.1.6 Creating New Priorities

Project manager can specify the priorities of the project.

2.1.7 Creating User Groups

Project manager should define teams by specifying team_id, team_name and team description. Then he can add users to these teams.

2.2 Task Editing

Access to this scenario is restricted by Task Editing Levels access right. We can categorize task editing as below for different access rights.

2.2.1 Creating New Task

For user to create a new task, he should have the necessary right. The scenario starts with supplying description, estimated time and details. User also specifies the task group, task type and task priority for the task which are defined before. User must also give the start and due dates to the system.

2.2.2 Assigning Task

If the user has the necessary right, user can assign the task to other users that work in the project. System must inform the user that will work on task by e-mail or some other notification.

2.2.3 Add/Delete Attachment

Users should have the capability to attach files to tasks. This can be done by specifying the path of the attach file and description of the attached file.

2.2.4 Viewing History Trail

Users that have the necessary right can view the history of the task which includes date, moderator, old value and new value.

2.2.5 Open/Close In-Out

Users can open or close their In-Out if they were given this right. When opened by the user, the system calculates the elapsed time that passes while user is working on task. This will generate the time sheet of the user automatically.

2.2.6 Allocating Material

User can allocate materials that exist in the project by selecting material type and quantity of the material. If the material does not exist, user can define new material type and purchase some quantity. These will be explained in Resource Management case.

2.2.7 Handling Task Dependency

User can insert dependency between two tasks by clarifying the dependency type and data if he has the necessary right. These dependencies may restrict some operation on tasks like editing some fields. Dependencies must include finish-start in which one task waits the other to be finished, start-start in which one task waits some time after the start of the other, etc. System must inform the user that works on tasks when dependency constraint is lifted off.

2.3 Resource Management

Every project can have some resources. These resources are defined and purchased by some user that has the necessary right. These purchased items' cost and human resources' cost must be automatically decreased from the project budget.

2.3.1 Defining Material

User defines a new material by specifying the material attributes like; material_id, material_name, material_unit_cost, etc.

2.3.2 Purchasing Material to Project

User that has the right can purchase materials for the project that have been defined before. User specifies the quantity that is purchased and the cost will be decreased from budget automatically. Purchasing information must be saved for later operations.

2.4 Time Approval & Time Sheets

The system must define the necessary rights for user to fill their time sheets manually or automatically. After the generation of time sheets, they must be sent to another user that has the necessary right to approve this time sheet.

2.4.1 Generating Time Sheets

User specifies the week that time sheet is owned and add the work items to this time sheet. Then he specifies the time that he works on the work item for every day in the week. User must select another user that has the right and send time sheet for approval.

2.4.2 Approval of Time Sheet

User selects the time sheet that is sent to him and approves or rejects it. If rejected, user that has sent the time sheet must generate new time sheet for that week.

2.5 Statistics and Time/Task Reports

2.5.1 Viewing Statistics of Project

Users that have the necessary right can view the statistics of project which includes general project statistics, projects user statistics, task groups' statistics, task types' statistics, task priorities' statistics and task statuses' statistics. These statistics must include the number of items that they include for some categorization.

2.5.2 Generating Time/Task Reports

User can select the filtering data and generate the time report that includes the work items and working time on that item according the selected filter. Task report contains only work items and related information about that work item that is filtered by user.

2.6 Generating Charts

User should have the capability to generate some charts for viewing statistics and time / task reports graphically.

2.6.1 Gantt Chart

User can view the Gantt chart which shows the tasks in a time table. This chart must show the start-due dates of tasks and progression on tasks. Also user can filter data for the Gantt chart.

2.6.2 Dependency Chart

User can view the dependencies of tasks graphically by applying filter and generating this chart. Chart must show the dependency type and dependency data for every dependency. In addition to these, chart must highlight the critical path for the project.

2.6.3 Other Charts

User can generate some other charts like pie charts, line charts, etc for visualization of statistics and time reports for some filter.

2.7 Forum

User can create new thread in the Forum and write the message to it or write the message to another created thread.

2.8 User Preferences & User Directory

User can edit his preferences for the system. These preferences are applied to the system for the user until they are changed. Users also can view other users' information but this is restricted by some access right.

3) Client

The clients will log into the system using user names and the passwords provided to them and they will monitor the development phase of the project.

3. DATA MODEL AND DESCRIPTION

3.1 DATA OBJECT DESCRIPTION

1) User:

'User' is the entity that constitutes the internal system representation of a person using DProject. It can represent an ordinary developer, a project manager, or an administrator. It interacts with the system to accomplish some tasks, communicate, arrange a meeting, and get information. It has a *user_id*, that uniquely determines it, a *password* that secures its data, a

name, surname, birth_date, sex, address, e-mail_address, phone_number, and photo. Other than this identifying information, 'User' entity also has properties that specifies it in a project concept; like *speciality, payment_policy*(regular salary or hourly wage), and *payment_amount*. The 'User' also can have *preferences* in using the program to configure and use it in the most effective way. Every user *is part of a company*, and can *work in a project*, in which it has certain *access-rights*.

2) Task:

'Task' is the entity in the system that represents a real-world task, which requires time, resources, and labor to be completed. A 'Task' entity has all the detail to define a certain task, so that its whole scope can be specified by just inspecting the entity. The tasks are created, modified, and deleted by the system *users*, who use them to divide the work into small and non-overlapping parts, so that the starting, developing, and completing phases all can be viewed clearly and consistently. The 'Task' entity has a *task_id* that uniquely determines it, a *task_name, task_description, start_date, deadline, and type* which are set from the beginning of the life-time of the entity, and other attributes such as the *percentage_done, status, priority, worked_time_on_task, individual_work_times, finish_date, and comments* which can change during the life-time of the process. It is also possible that tasks are placed inside a *task_group*. Other than being in a task group, the tasks may be arranged so that one *depends on another task*. For the *users*, a task can be one that *is assigned to him*, or one that he *is the reviewer of* it. Every change made on a file is stored in a history trail, so each file *has a history trail*. Other than its internal properties, externally a *file may be attached to a 'Task'*.

3) User Group:

Users can come together to form groups so that they can be assigned a specific task as a group, and has a certain hierarchy in it. By this way, a 'User Group' entity forms. This entity has a *user_group_id* which is used to identify the user group uniquely; and a *user_group_name*. A user group *has_users* inside it, and also it can be *assigned_to a task*.

4) Project:

A 'Project' is an entity which is the main constituting part of DProject system. The entity 'Project' represents a real-life project, for which the task arrangements, meeting arrangements, resource management, and communication is facilitated by the system. In this context, every project has a *project_id* which is used to identify a project uniquely, a *project_name, a description* what the project is aiming, and a *start_date* which are generally set at the beginning of the project. Other attributes like the *estimated_duration* of the project, and the *budget* of the project are mostly updated as the project proceeds. Every project also has a *client* as usual. The client has certain rights to get informed about the progress of the project. Also, as usual every project needs some material resource to proceed. These resources consist of the *material types which the project has as defined*.

5) Material:

The entity 'Material' forms the base for the material resources that will be used in a task; and so, in a project. They are definitions of actual materials in abstract terms, so that

they can be realized in a project and be used as resources. Every 'Material' entity has a unique *material_id*, a *material_name* and the *material_unit_cost*. The address where the material is *purchased_from* can also be kept along with a description of the material.

6) Task Resource:

The entity 'Task Resource' is an entity that consists of organized materials so that it can be used in a task. It has a *material_id* to determine which material this resource *consists of*. The *quantity* is another attribute of the 'Task Resource', which determines how much (in units) of the material (the one which has the *material_id* as its id) is required for the task to be accomplished. The 'Task Resource' is *assigned to* a specific task.

7) Project Resource:

The entity 'Project Resource' represents the current resources of the project, so that they can be assigned to tasks. The project resources are created by purchasing materials. It has a *material_id* to determine which material this resource *consists of*. It also has the attribute *quantity* showing how much of that material is available in the project. A project resource entity is *assigned to* a specific project.

8) Client:

The entity 'Client' represents the client of the project. It simply has a *company_name*, a *login_name* and *password* for the project, an *e-mail address* and *contact info* for communication, and the *name of the person in charge*. This entity is used to identify who the project is being done for; and facilitate communication between the project developers and project client.

9) Company:

The entity 'Company' represents the company using DProject, and developing projects with it. Every company has certain attributes such as *compant_name*, *address*, *contact info*, *e-mail server*, *webpage*, and *timezone/format*. The e-mail server info will be used if the members of a company have their own company e-mail addresses and want to use it. Also a company can supply its *logo* to be used in the visual environment of DProject. Another important issue about a company is its *week management*, that is on which day its working days in a week start and when it ends. Every company has at least one user as an *administrator*. Also every *company has its own forum* pages so that company specific issues can be discussed.

10) To_Be_Arranged_Meeting:

The 'To_Be_Arranged_Meeting' entity represents a meeting that is in the process of being arranged. That is, the *time options*, *potential participants*, *the place* and *the description* of the meeting is set, and it is scheduled to be arranged. In the process of being arranged the time options are also used to form a *priority list*, that holds the current choices of the potential participants that made their choices. Usually a *user arranges* a meeting. A to be arranged meeting can also have *attached files* that gives information relevant to the future meeting.

11) Arranged_Meeting:

The 'Arranged Meeting' entity represents a meeting that is now arranged and has its exact *date*, *place*, *description* and *participants*. It is also *arranged by* a user, and can have *attached_files*, that are either attached before the meeting was arranged, or after the meeting was arranged or even after the meeting actually took place

12) History Trail:

Every task has a 'History Trail', that keeps the major modifications done to the task, so that when a user wants to see this information they can be presented. A history trail of a task *has history trail items* as its constituent parts.

13) History Trail Item:

A 'History Trail Item' is an entity that stores one modification done on a task, so that when these are grouped together they can show all the changes made on a task. A 'history trail item' has a *task_id* as its attribute, which shows the specific task this item is for. It also stores the id of the *modifier* (i.e. the user that made the changes). The trail item also keeps the name of the *field* the change was made upon, its *old value*, and *its new value*.

14) Forum:

The entity 'Forum' represents a forum facility that is supplied to every company in DProject. Generally speaking, the forum *consists of forum threads*.

15) Forum Thread:

The entity 'Forum Thread' constitutes the actual Forum entity. It has a specific *subject*, and it in turn *consists of individual forum messages*.

16) Forum Message:

'Forum Message' entity is the smallest entity that the whole forum consists of. It consists of a *sender* (who is another user of the system from the same company), *subject* of the message, *date* of the message, and finally the actual *message body*.

17) E-Mail:

The entity 'E-mail' is the representation of an e-mail, which one of the users of DProject may want to send to the outer internet. For this aim it *has a sender* (one of the users), a *receiver* of the mail (which may or may not be a user of the system), a *subject* for the mail, a *message body* for the mail, and the *date* that it is sent.

18) Access Right:

The entity 'Access Right' is owned by every user so that it determines what the user can and can not do using DProject. Firstly, the *user_directory_rights* determine whether the user is able to see the other users' information from the project, or if he/she can see only the

project managers' info, or can not see nobody's info. Secondly, the *timesheet_rights* determines how the user records the work time, either by using the in/out facility or by filling in the timesheet by him. Thirdly, the *time_approval_right* simply determines if the user has the right to approve working time of another user. The *task_modify_access_rights* determines what can a user do on a task. He can read, edit, delete, create according to the *task_modify_access_right* type he has. Lastly, the global rights determine if a user is an Administrator or not. If the user is an Administrator it has the highest level of rights that we counted above. But if he is not an Administrator then he is subject to those restrictions.

19) Filter:

'Filter' entity is the one that materializes what a user wants to see in a report, statistics, or chart. Using the filter entity reports, statistics, and charts are generated that corresponds to the user needs. A user may want to see data about a user using the *user_option* of a filter, may want to see data about a specific task using the *task_option* of a filter, may want to see data from a specific time interval using the *time_interval_option* of a filter, may want to see data from a specific project using the *project_option* of a filter, may want to see data from a task group which have specific attributes like *priority_option*, *status_option*, *task_type_option*, *task_group_option*; and may want to see data about a specific user group using the *user_group_option*, and may want to see data about a specific material type using the *material_option* of a filter. Reports, statistics and chart generation will be done according to the choices made in these options. Another aspect of the 'Filter' entity is that it can be saved and used later, so that a user using the same kind of filter every time he needs a report does not have to construct one for every report. These require a filter to have a *filter_id*, and a *filter_name* so that it can be identified uniquely and be named.

20) Reports:

'Reports' are entities that are generated by a user to present to another user, to get an approval. Every report *has a filter*, and report generating is done according to this filter and data is parsed according to it. Users can also specify *formats* for a report, which specifies the visual attributes of a report.

21) Statistics:

'Statistics' are entities that are generated by a user to see data in an organized way so that inferences may be made. The organization of the data is done according to a user specified filter. So every 'Statistics' entity *has filter*. It also has an attribute named *format*, which is also specified by the user to determine its visual form.

22) Charts:

'Charts' are presentation of data to the user in a format that is one of the formats that the system has as defined. These are the *types* of a chart (e.g. a Gantt chart). Also a chart *has filter* to determine the time interval, specific users and etc. according to which the chart will be produced.

23) Notification:

'Notification' entity is used to notify users of certain events. That can be a notification about a task assigned to the user, a notification about a meeting that he/she is chosen as a

participant, or a direct notification from another user including a message. These constitute the *notification type*. Every notification also has a *notification_id* that uniquely determines it, a *message_body*, and a *date* attribute. They may also have *attached_files*. Every notification has a user as the *receiver of* it, and some notifications may have a user as the *sender of* it, while some other may be system generated notifications.

24) File Directory:

The entity 'File Directory' is a real directory, in which the files attached to tasks, meetings, and notifications are held. So, the 'File Directory' *includes files* in it.

25) File:

A 'File' entity is a real file that may be attached to a task, a meeting or a notification. It has a unique *filename*.

26) Database:

The 'Database' entity represents the data storage facility of DProject. All the information about projects, companies, tasks, users, meetings etc. will be held in the database. According to the data type it will hold database will consist of a *notifications database*, a *meetings database*, a *tasks database*, a *user accounts database*, a *project database*, a *company database*, a *material database*, and a *filter database*.

3.2 DATA RELATIONSHIP DESCRIPTION

1) works_in:

The relationship 'works_in' is between the 'user' and 'project' entities. It means that the specified user is working in the specified project. One user may work in many projects, and one project may have many users in it. Also, for every project there must at least be one user in it, but there may be users who does not work in a project.

2) has_access_right:

This relationship is between the entities 'user', 'project' and 'access rights' and determines what access right the user has in the project. A user has to have a specified access right in a project, and it has to be unique.

3) work_in_company:

The relationship 'work_in_company' is between the 'user' and 'company' entities. It means that the specified user is working in the specified company. One user must work in exactly one company, but one company may have many users in it. Also, for every company there must at least be one user in it.

4) includes_file:

The relationship 'includes_file' is between the 'file' and 'file directory' entities. For a file there must be exactly one file directory, and a file directory may have zero or many files in it.

5) sends_notification:

The relationship 'sends_notification' is between the 'user' and 'notification' entities. For a notification there must be exactly one user to send it, and a user may have zero or many notifications sent.

6) receives_notification:

The relationship 'receives_notification' is between the 'user' and 'notification' entities. For a notification there may be at least one (or more) user to receive it, and a user may have zero or many notifications received.

7) is_attached

The relationship 'is_attached' is between the 'file' and 'notification' entities. For a notification there may be zero or many files attached to it, and a file must be attached to zero or more notifications.

8) report_has_filter:

The relationship 'report_has_filter' is between the 'report' and 'filter' entities. For a report there must be exactly one filter, and a filter may have zero or many reports.

9) statistics_has_filter:

The relationship 'statistics_has_filter' is between the 'statistics' and 'filter' entities. For a statistics there must be exactly one filter, and a filter may have zero or many statistics.

10) chart_has_filter:

The relationship 'chart_has_filter' is between the 'chart' and 'filter' entities. For a chart there must be exactly one filter, and a filter may have zero or many charts.

11) has_history_trail:

The relationship 'has_history_trail' is between the 'task' and 'history_trail' entities. For a task there must be exactly one history trail, and a history trail must have exactly one task for which it stands.

12) depends_on:

The relationship 'depends_on_tasks' is between the 'task' and 'task' entities. For a task there may be one or zero tasks to depend on.

13) assigned_to_user:

The relationship 'assigned_to_user' is between the 'user' and 'task' entities. For a task there may be zero or many users that it is assigned, and a user may have zero or many tasks to complete.

14) reviewer_of:

The relationship 'reviewer_of' is between the 'user' and 'task' entities. For a task there may be zero or many reviewers, and a user may have zero or many tasks to review.

15) task_has_attachment:

The relationship 'task_has_attached_files' is between the 'task' and 'file' entities. For a file there may be zero or many tasks that it is attached, and a task may have zero or many files attached to it.

16) has_item:

The relationship 'has_item' is between the 'history trail' and 'history trail items' entities. For a history trail item there must be exactly one history trail that it is in, and a history trail may have zero or many history trail items.

17) sends_email:

The relationship 'sends_email' is between the 'user' and 'e-mail' entities. For a e-mail there must be exactly one user to send it, and a user may have zero or many e-mails sent.

18) has_threads:

The relationship 'has_threads' is between the 'forum' and 'forum thread' entities. For a forum thread there must be exactly one forum, and a forum may have zero or many forum threads.

19) has_messages:

The relationship 'has_messages' is between the 'forum threads' and 'forum messages' entities. For a forum message there must be exactly one forum thread, and a forum thread may have zero or many forum messages in it.

20) user_arranges:

The relationship 'user_arranges' is between the 'user' and 'arranged_meeting' entities. For a meeting there must be exactly one arranger, and a user may have zero or many meetings that are arranged by him.

21) user_participates:

The relationship 'user_participates' is between the 'user' and 'arranged_meeting' entities. For a user there zero or more arranged meetings that he is one of the participants, and an arranged meeting may have zero or many participants.

22) has_files:

The relationship 'has_files' is between the 'arranged_meeting' and 'file' entities. For a file there may be zero or more arranged meetings that it is attached to, and an arranged meeting may have zero or many files.

23) user_arranges_2:

The relationship 'user_arranges_2' is between the 'user' and 'to_be_arranged_meeting' entities. For a meeting there must be exactly one arranger, and a user may have zero or many meetings that are arranged by him.

24) user_participates_2:

The relationship 'user_participates_2' is between the 'user' and 'to_be_arranged_meeting' entities. For a user there zero or more to-be-arranged meetings that he is one of the participants, and a to-be-arranged meeting may have zero or many participants.

25) has_files2:

The relationship 'has_files2' is between the 'to_be_arranged_meeting' and 'file' entities. For a file there may be zero or more to-be-arranged meetings that it is attached to, and a to-be-arranged meeting may have zero or many files.

26) is_admin_of:

The relationship 'is_admin_of' is between the 'user' and 'company' entities. For a company there must be at least one (or more) users that are administrators for the company, and a user may have zero or one company that he is the Administrator of.

27) has_forum:

The relationship 'has_forum' is between the 'company' and 'forum' entities. For a company there must be exactly one forum, and a forum has exactly one company that it belongs to.

28) is_client_of:

The relationship 'is_client_of' is between the 'client' and 'project' entities. For a project there must be exactly one client, and a client may have zero or many projects.

29) has_material:

The relationship 'has_material' is between the 'project' and 'material' entities. For a project there zero or more materials defined, and a material must have exactly one project that it is defined in.

30) g_assigned_to:

The relationship 'g_assigned_to' is between the 'user group' and 'task' entities. For a user group there may be zero or more tasks that are assigned, and a task may be assigned to zero or many user groups.

31) g_assigned_to_p:

The relationship 'g_assigned_to_p' is between the 'user group' and 'project' entities. For a user group there is exactly one project that it is assigned to, and a project may have zero or many user groups.

32) has_user:

The relationship 'has_user' is between the 'user group' and 'user' entities. For a user there may be zero or more user groups that he is a member of, and a user group may have zero or many users in it.

33) is_assigned_to_task:

The relationship 'is_assigned_to_task' is between the 'task resource' and 'task' entities. For a task resource there must be exactly one task that it is assigned to, and a task may have zero or many task resources assigned to it.

34) task_resource_consists_of:

The relationship 'task_resource_consists_of' is between the 'task_resource' and 'material' entities. For a task resource there must be exactly one material that it consists of, and a material may have zero or many task resources that it is a part of.

35) is_assigned_to_project:

The relationship 'is_assigned_to_project' is between the 'project resource' and 'project' entities. For a project resource there must be exactly one project that it is assigned to, and a project may have zero or many project resources assigned to it.

36) project_resource_consists_of:

The relationship 'project_resource_consists_of' is between the 'project_resource' and 'material' entities. For a project resource there must be exactly one material that it consists of, and a material may have zero or one project resources that it is a part of.

3.3 ENTITY RELATIONSHIP DIAGRAM (ERD)

The Entity Relationship Diagram did not fit in an A4 paper, so it can be found in the Appendix presented in an A3 paper layout.

3.4 DATA DICTIONARY

Name: User

Aliases: User info

Where used/how used: create new user

Description:

user = user_id + password + name + surname + middle name + birth date + photo
+ speciality + payment_policy + payment_amount + phone_number + address + sex +
e-mail addr. + preferences + project id + access right type + company id

Name: Notification

Aliases: task notification, meeting notification

Where used/how used: arrange meeting, manage tasks

Description:

Notification = notification_id + notification type + message + date + sender id
+ receiver id + attached file names

Name: Report

Aliases: None

Where used/how used: generate report

Description:

Report = format + filter id

Name: Statistics

Aliases: None

Where used/how used: generate Statistics

Description:

Statistics = format + filter id

Name: Chart

Aliases: None

Where used/how used: generate chart

Description:

Chart = type + filter id

Name: Task

Aliases: task info, assigned task info, new tasks info, reviewed task info, deleted task

Where used/how used: manage tasks

Description:

Task = task_id + task_name + task_description + start date + deadline +
finished-date + priority + type + status + percentage_done + task group +
history_trail_id + dependent_task_ids + assigned_user_ids + reviewer_ids +
attached_file_names + comments + worked-time on task
+ individual worked times

Name: History Trail item

Aliases: Task edit info

Where used/how used: Manage Tasks

Description:

History trail item = modifier_id + field name + old value + new value

Name: History Trail

Aliases: None

Where used/how used: Manage Tasks

Description:

History Trail = History Trail item ids + Task id

Name: File

Aliases: Attached file

Where used/how used: manage tasks, arrange meeting

Description:

File = file name + file info

Name: E-mail

Aliases: None

Where used/how used: arrange meeting, manage tasks

Description:

e-mail = subject + receiver + message body + sender_id

Name: Forum

Aliases: None

Where used/how used: process forum

Description:

forum = thread ids

Name: Forum thread

Aliases: None

Where used/how used: process forum

Description:

forum threads = forum message ids + subject

Name: Forum message

Aliases: None

Where used/how used: process forum

Description:

forum message = sender id + subject + message body

Name: Access Rights

Aliases: Rights for users

Where used/How used: create new user, manage tasks, manage resources, meeting arrangement

Description:

Access Rights = global rights + user directory rights + time sheet rights + edit task + level rights + time approval rights

Name: Arranged Meeting

Aliases: meeting details,

Where used/How used: meeting arrangement

Description:

arranged meeting = date + place + description + arranger_id + participator_id
+ attached file names

Name: to be arranged meeting

Aliases: None

Where used/How used: Meeting Arrangement

Description:

to be arranged meeting = time options + priority + list + place + description +
attachments + arranger_id + potential_participator_id +
attached file names

Name: Company

Aliases: Company info

Where used/How Used: Initialize company

Description:

company = company name + address + contact info + logo + e-mail server +
webpage + timezone/format + week management + admin_id +
forum_id

Name: Project

Aliases: project info

Where used/How Used: create new project

Description:

project = project name + project id + description + start date + estimated
duration + budget + client_id + material_ids

Name: Client

Aliases: customer

Where used/How Used: manage tasks

Description:

client = client company name + login name + e-mail + name of person in charge +
password + contact info + monitored_project_id

Name: User Group

Aliases: team

Where used/How Used: manage tasks

Description:

user group = assigned_project_id + user_ids + user group id + user group
name

Name: Filter

Aliases: filtered info, filtered data, saved filter info

Where used/How Used: statistics&report generation, manage tasks

Description:

filter = filter_id + filter_name + user_option + task_option +
time_interval_option + project_option + priority_option + status_option +
task_type_option + task_group_option + user_group_option + material_option

Name: Material

Aliases: Material resources

Where used/How Used: manage tasks, manage resources

Description:

material = material name + material id + material unit cost + address of
purchase + description

Name: Task Resources

Aliases: None

Where used/How Used: manage resources, manage tasks

Description:

task resources = material id + quantity + assigned task_id

Name: Project Resources

Aliases: None

Where used/How Used: manage resources

Description:

project resources = material id + quantity + assigned project_id

Name: .dpr files

Aliases: None

Where used/How Used: Export/Import

Description:

.dpr files = file_name

Name: xml files

Aliases: None

Where used/How Used: Import/Export

Description:

xml files = file_name

4. FUNCTIONAL MODEL AND DESCRIPTION

4.1 DESCRIPTION OF FUNCTIONS

4.1.1 Process Specification (PSPEC)

1. process new login

The process new login takes the password of the user according to its userid from database and take the decision by comparing it with the entered password. The decision may be reject login or accept login. If the process select reject case process will start again.

2. create new user

This process takes the user info from the user and stores them to the database. If user data with same userid exist in database it rejects the creation of new user which is decided by the new user info status retrieved from database. User info includes user-name, user-id, passwd, etc. and will be explained deeply in data description model in this document.

3. initialize company

The initialize company process takes the company info from the user and stores this information to database. Company info includes company-name, address, etc. and will be explained clearly in data description model.

4.1 create tasks

This process initially takes the access rights of the user (owner of the session) from the database by sending the user-id to database and takes the decision that if that user can create task and to whom this task can be assigned. If user has the right to create task, process takes the task info and attach files from user and stores them to database. If also user has the task assignation right, user specifies the assigned persons. Then process save the information and task assign notification to database. Also process sends e-mail to these assigned persons as e-mail notification.

4.2.1 check user access rights

The process starts with editing request and takes the user-id, task-id information from the session. Check the user right for editing task by querying with user-id from the database. After this evaluation the process gives the edit right to user or rejects it. If edit right is given, then the process sends the user-id and task-id to the check dependency process.

4.2.2 check dependency

The process starts with taking the user-id and task-id from the check user access rights process. Then it takes the dependent task status from the status according to task-id. Dependent task status includes the dependent tasks to the task which is identified by the given task-id and the process evaluates this information to decide if there is a dependency constraint that restricts the task editing operation. After the evaluation, process takes the decision that if task is editable or not. If task is editable then it sends the task-id and user-id to the makes the changes process.

4.2.3 make the changes

The makes the changes process takes the user-id and task-id from the check dependency process and edited fields from user and stores these edited fields to database

which is specified by task-id and user-id. Editable fields will be clarified in data description model.

4.3 delete tasks

The process starts with the delete request and queries the access rights of user by user-id if the user has the deleting right. If deletion is accepted, the task specified by task-id is deleted from the database.

4.4 request approval of work done on tasks

The process starts with approval request on task and sends the task-id to the check reviewer process.

4.5 check reviewer

This process takes the task-id from the request approval process and takes the user-id of reviewer from the database by using the task-id. Then it stores reviewer notification to database. This notification helps the reviewer to see the approval request of user.

4.6 notify tasks to assign and approval

The process starts with login and sends the user-id which is the login input to database to user task notifications. These notifications include assign and approval notifications that are sent to user.

9.1 obtain project data

The process takes the project-id that is to be exported from the user and takes the project data from the database owned by this project-id. Then it sends these data to the change to sql statements process.

9.2 change to sql statements

This process takes the project data from the obtain project data process and project files from the file directory and creates one file with .dpr extension and one directory which contains attached files. The file with .dpr extension contains the sql statements to build the project.

9.3 import .dpr file

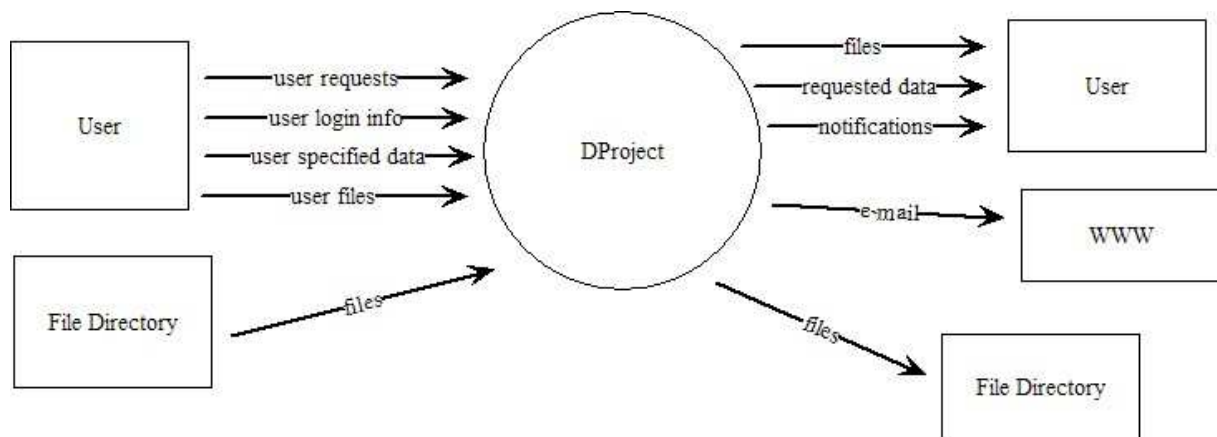
The process takes the file with .dpr extension and stores the project information and the project that file contains to database.****Attached files***

9.4 convert to XML

This process takes the report generated by user and related files from the database and converts them into xml file.

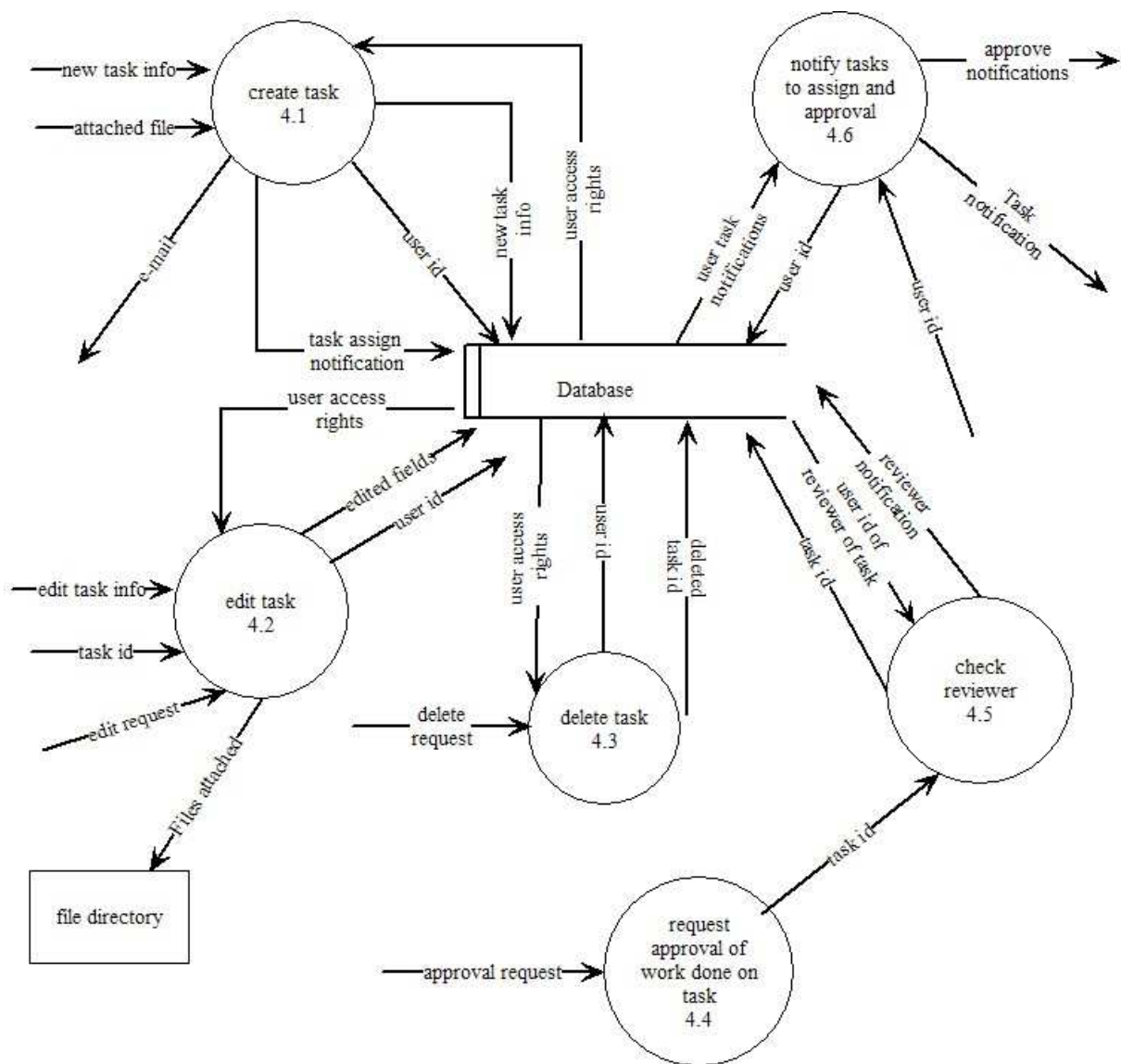
4.1.2 Functional Modeling

1) Level-0 DFD:

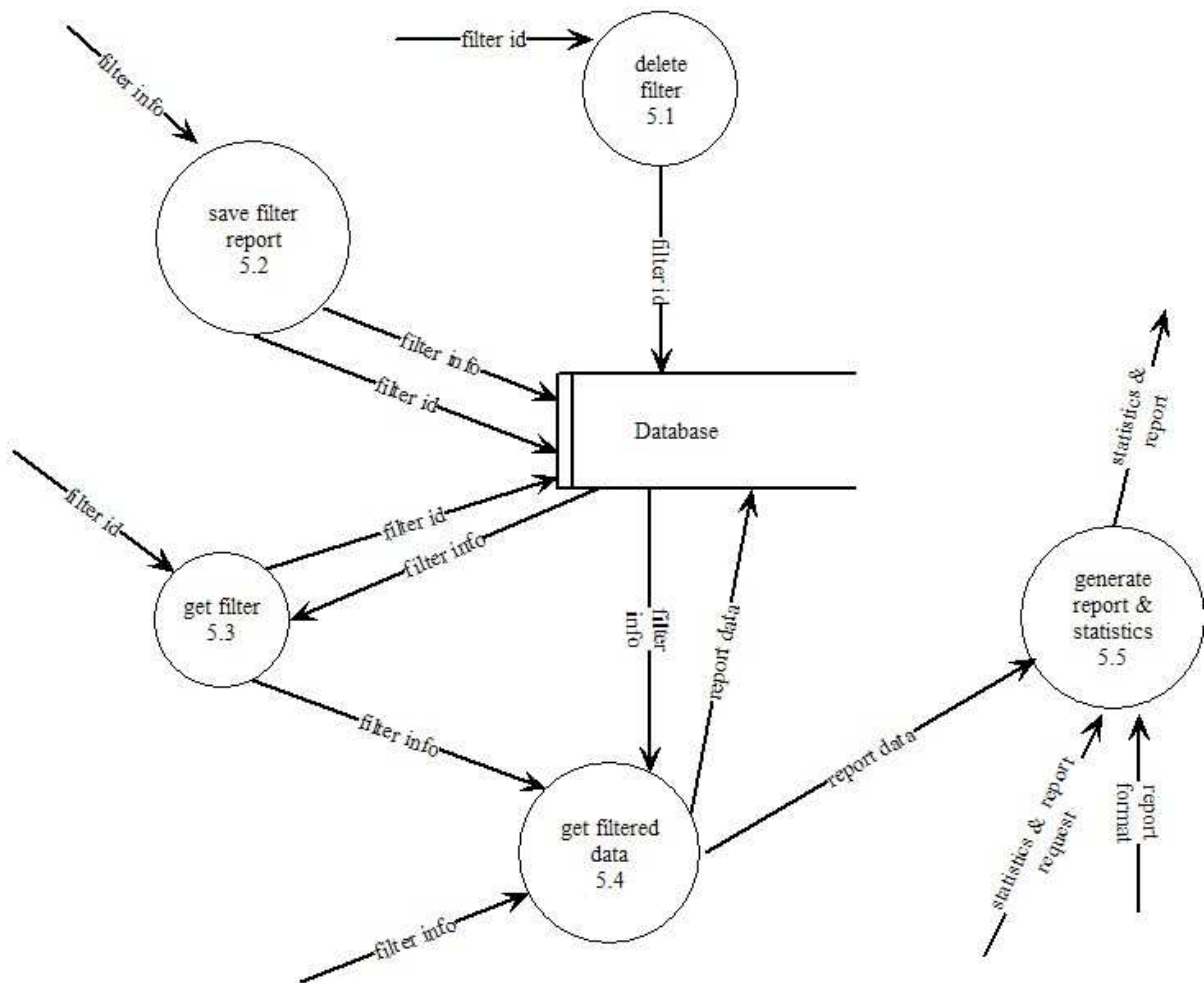


3) Level-2 DFD:

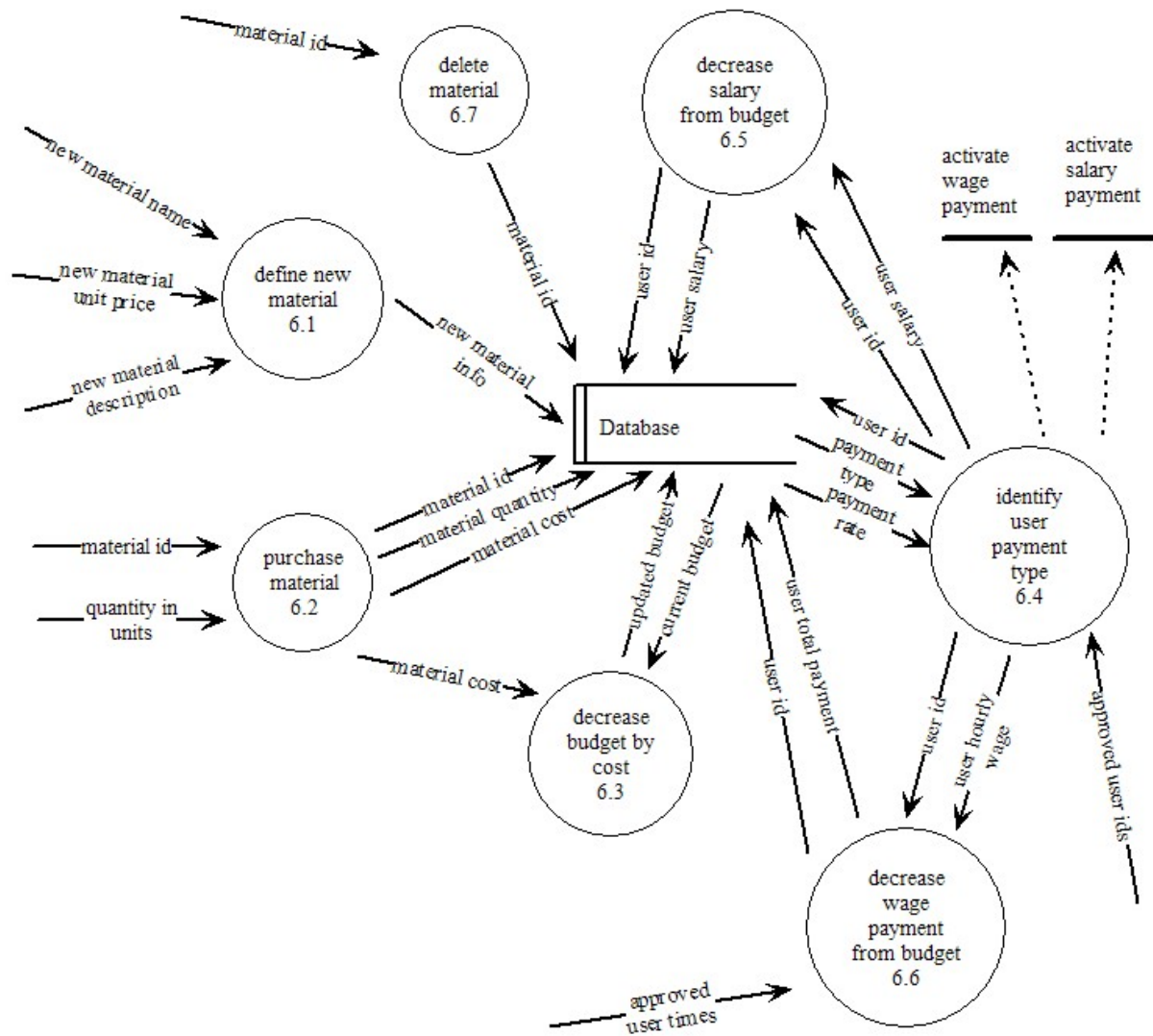
i) Manage Tasks



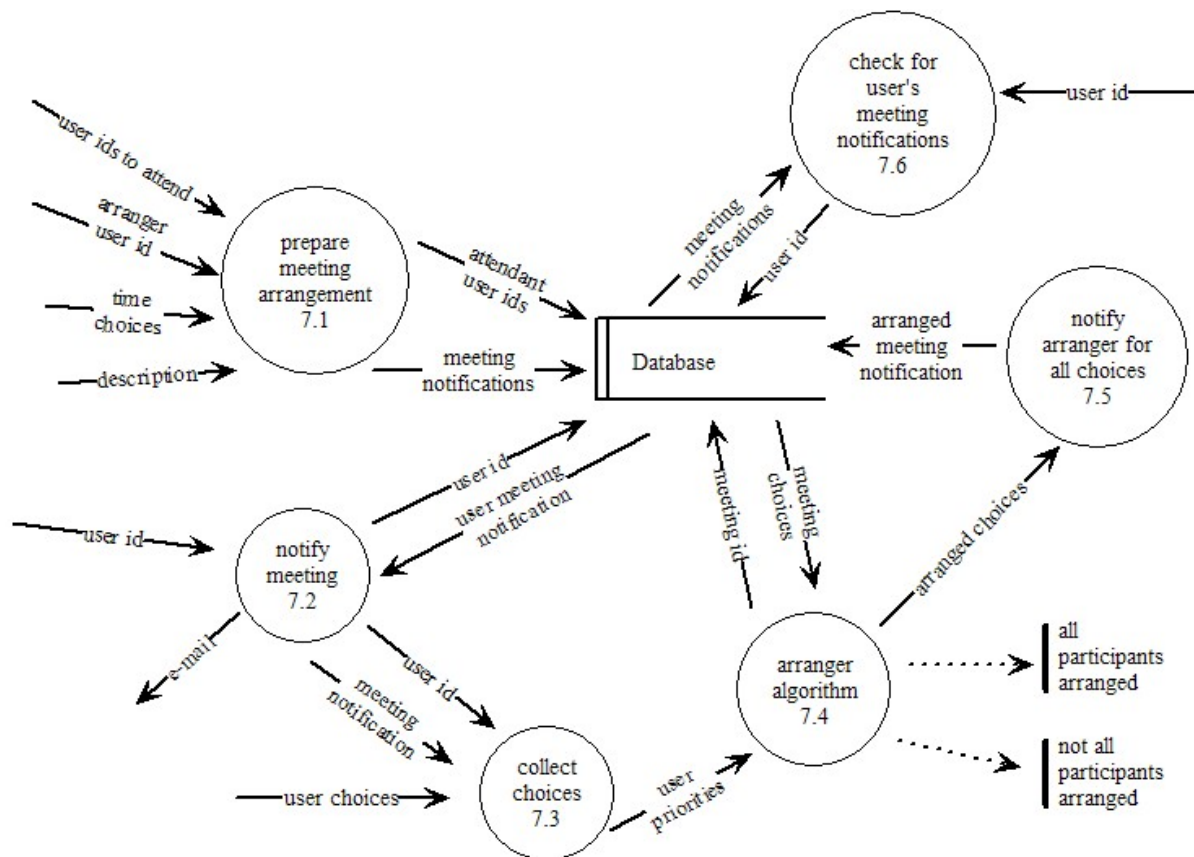
ii) Statistics & Report Generation



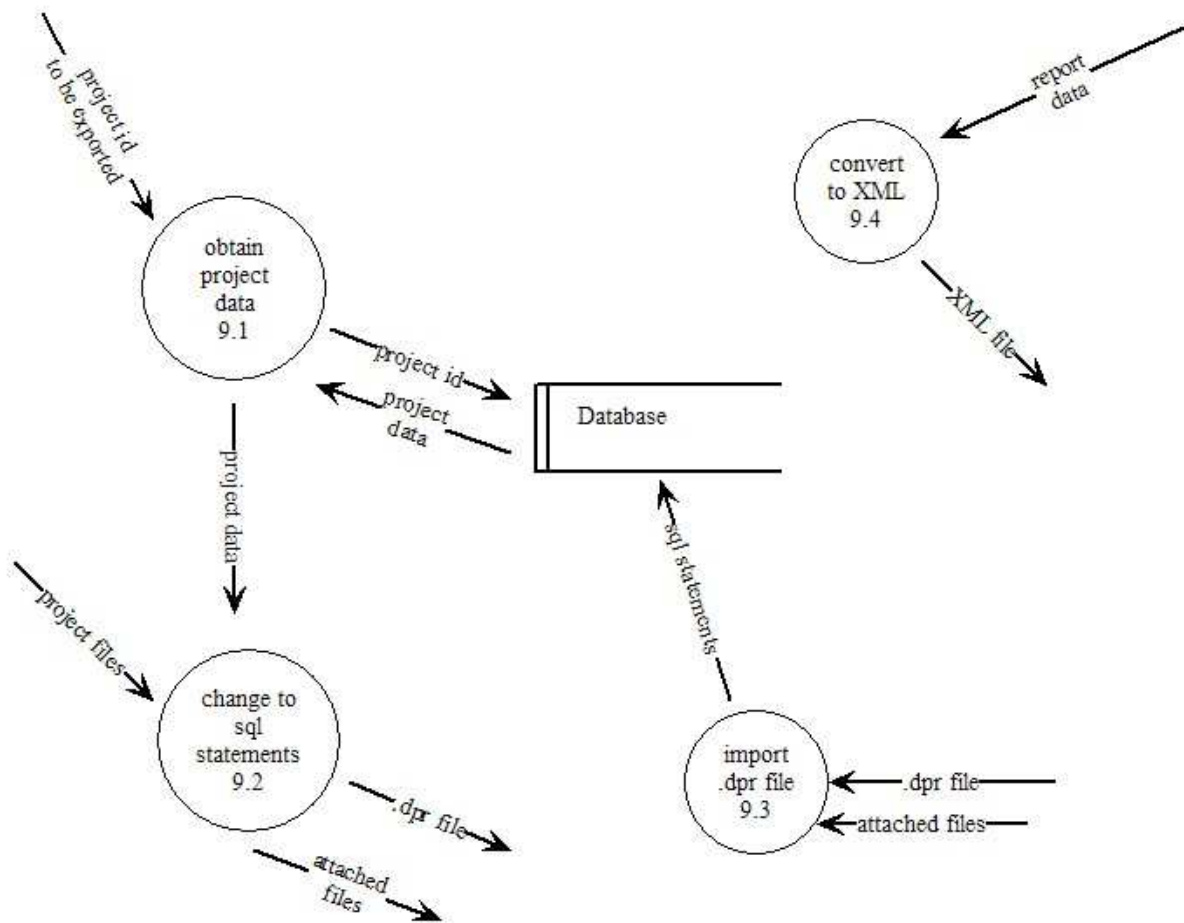
iii) Manage Resources



iv) Meeting Arrangement

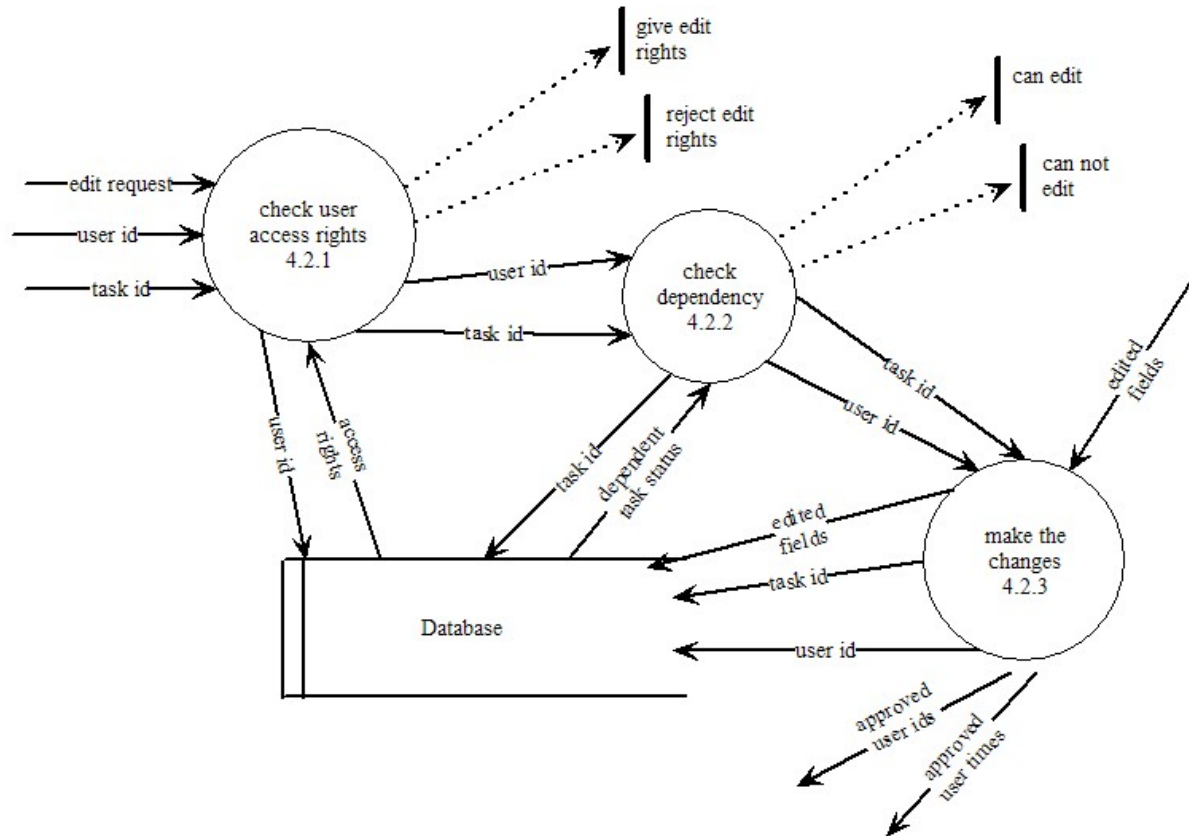


v) Import & Export



4) Level-3 DFD

Edit Task



5. BEHAVIORAL MODEL AND DESCRIPTION

5.1 DESCRIPTION FOR SOFTWARE BEHAVIOR

5.1.1 States

The major states of the system are as follows:

1 - Waiting for login state

Waiting for login state is the initial state of the system in which the system is waiting for a login request from the user. The system is idle in this step waiting for an action. Depending on the request from the user, the system either moves into the initializing new

company step (user wants to initialize new company information) or the waiting for user request state (user logs in to the system).

2 - Initializing new company state

In the initializing new company state, the system serves the user's requests for setting up the new company information. In this state, the user enters all the necessary information for a new company into the system. After the initialization is successfully completed, the system moves into the waiting for user request state to further respond to the user needs.

3 - Waiting for user request state

The waiting for user request state is the state from which the system moves into different states depending on the request from the user. In this state, the system carefully parses the user selections and move into the next state accordingly. Depending on the request from the user, the system can move into creating new user, handling task operations, handling report & statistics generation, handling meeting management, handling resource management, handling export & import and handling forum operations state. Also, if the user makes a log out request, the system again returns to the waiting for login state.

4 - Creating new user state

In the creating new user state, the system responds to user needs for creating a new user account. The user enters all the necessary information for a new user in this state. After the new user account is successfully created, the system returns to the waiting for user request state to respond to new user requests.

5 - Handling task operations state

In the handling task operations state, the system responds to all user requests related to task management. The user performs all task operations (creating, modifying accordingly, deleting, working on, assigning to users, assigning to reviewers, etc.) in this state. After the user requests are served, the system returns to the waiting for user request state to respond to new user requests.

6 – Handling statistics & report generation state

In the handling statistics & report generation state, the system satisfies the user needs related to statistics & report generation. The user generates reports and filters using the filters created instantly or created before and saved in the system. After the user completes the operations he wants, the system returns to the waiting for user request state to respond to new user requests.

7 – Handling meeting management state

In the handling meeting management state, the user performs all the operations related to meeting management. The arranger fixes meeting details (like sending options to possible attendants, deciding on inviting whom according to the responses from them, etc.) and the attendants respond to the options send to them from the arranger (like choosing the possible options, giving excuses, etc.). After the operations related to meeting arrangement are

completed, the system returns to the waiting for user request state to respond to new user requests.

8 – Handling resource management state

In the handling resource management state, the user performs all the operations related to resource management like setting up new material info (quantity, unit price, etc.), attaching needed material info to newly created tasks, sharing resources among current projects, etc. After the operations related to resource management are completed, the system returns to the waiting for user request state to respond to new user requests.

9 – Handling export & import

In the handling export & import state, the user performs importation and exportation to and from the system. The user imports and exports new files, reports and even a whole project in that state. After the operations related to importation and exportation are performed, the system returns to the waiting for user request state to respond to new user requests.

10 – Handling forum operations

In the handling forum operations state, the user performs operations related to the forums in the system. The user performs thread operations (creating, modifying, deleting), message operations (creating, sending, modifying, deleting) and all other possible forum operations. After the user's requests are responded successfully, the system returns to the waiting for user request state to respond to new user requests.

5.1.2 Events

Some major events that puts the system into one from another can be stated as follows:

1 – Login request

When the user wants to login to the system, he makes a login request and this invokes the process login process. The login request causes the system to move from waiting for login state to the waiting for user request state provided that the user enters a valid id and password to the system.

2 – Company initialization request

When the user wants to enter the information for a new company, he makes a company initialization request, which causes the system to move from the waiting for login state to the initializing new company state provided that the user which signs up a new company has entered a valid id and password to the system.

3 – Company initialization successful

When the user enters the information for a new company successfully (that is, all the required fields are filled, a non-existing id has been selected, etc.), the system moves from the initializing new company state to the waiting for user request.

4 – Create new user request

When the user wants to sign up a new user to the system, he makes a create new user request. This invokes the create new user process and causes the system to move from waiting for user request into the creating new user state.

5 – Task operation request

When the user wants to perform operations related to the task management capabilities of the system, he makes a task operation request. This request invokes the manage tasks process and causes the system to move from the waiting for user request state into the handling task operations state.

6 – Statistics & report request

When the user wants to perform certain operations related to statistics & reports, he makes a statistics & report request. This request invokes the statistic & report generation and causes the system to move from the waiting for user request into the handling statistics & report generation state.

7 – Meeting management request

When the user wants to perform operations related to the meeting management capabilities of the system, he makes a meeting management request. This request invokes the arrange meeting process and causes the system to move from the waiting for user request state to the handling meeting management state.

8 – Resource management request

When the user wants to perform operations related to the resource management capabilities of the system, he makes a resource management request. This request invokes the manage resources process and causes the system to move from the waiting for user request state to the handling resource management state.

9 – Export & import request

When the user wants to perform exportation and importation operations, he makes a export & import request. This request invokes the export & import process and causes the system to move from the waiting for user request state to the handling export & import state.

10 – Forum request

When the user wants to perform operations related to the forums of the system, he makes a forum request. This request invokes the process forum process and causes the system to move from the waiting for user request state into the handling forum operations state.

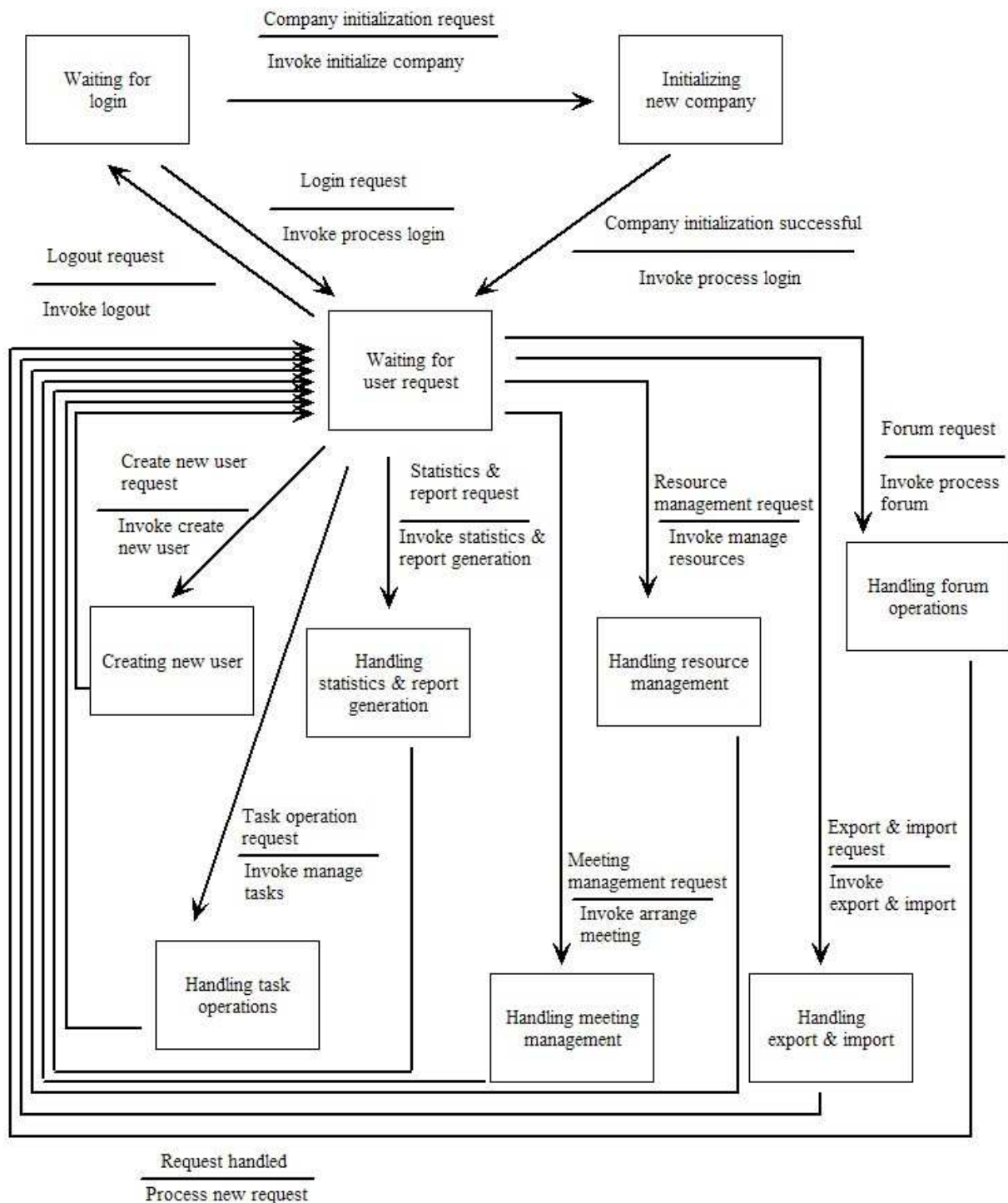
11 – Request handled

When the user's requests are responded accordingly and successfully, the system moves into the waiting for user request state from any request handling state to be able to respond to the new user requests.

12 – Logout request

When the user is done with the system and wants to log out, he makes a logout request. This request causes the system to move from the waiting for user request state into the waiting for login state.

5.2 STATE TRANSITION DIAGRAM (STD)



5.3 CONTROL SPECIFICATION (CSPEC)

1 – accept login

The accept login control signal is generated by the process login process. It is asserted when the user provides a valid id and password upon the login process. When it is asserted, the user is accepted to log in the system.

2 – reject login

The reject login control signal is generated by the process login process. It is asserted when the user provides a non-valid id and/or password while trying to log in the system. When it is asserted, the system waits for the user to provide a valid id and/or password.

3 – accept new user

The accept new user control signal is generated by the create new user process. It is asserted as a result of a successful creation (e.g. non-existing id, all the required fields are filled, etc.) of a new user account. When it is asserted, the system notifies the user that the new account is successfully created.

4 – reject new user

The reject new user control signal is generated by the create new user process. It is asserted as a result of an unsuccessful attempt (e.g. existing id, not all the required fields are created, etc.) to create a new user account. When it is asserted, the system prompts the user that the new account could not be created and requires the user to provide valid information to be able to create the new account.

5 – accept company initialization

The accept company initialization control signal is generated by the initialize company process. It is asserted as a result of a successful company initialization (e.g. vital fields that do not match with the existing ones, all the required fields are filled, etc.) in the system. When it is asserted, the user is notified that the new company information is saved successfully.

6 – reject company initialization

The reject company initialization control signal is generated by the initialize company process. It is asserted as a result of an unsuccessful attempt (e.g. vital fields that match with existing ones, not all the required fields are filled, etc.) to initialize a new company in the system. When it is asserted, the user is prompted that the new company could not be initialized successfully and requires the user to provide valid information to be able to create the new company in the system.

7 - activate wage payment

The activate wage payment control signal is generated by the identify user payment type process. It is asserted when the user to be paid is an employee that works on an hourly basis. When it is asserted, the system calculates the total amount of money that should be paid to the employee by the multiplication of the total working hours and hourly wage and decreases the budget by that amount.

8 – activate salary payment

The activate salary payment control signal is generated by the identify user payment type process. It is asserted when the user to be paid is an employee that works on a monthly basis. When it is asserted, the system decreases the budget by the amount of that employee's salary.

9 – all participants arranged

The all participants arranged control signal is generated by the arranger algorithm process. It is asserted when all the potential attendants of the meeting stated that they can be present at the meeting. When it is asserted, the arranger of the meeting is notified that all the invited employees will be present at the meeting

10 – not all participants arranged

The not all participants arranged control signal is generated by the arranger algorithm process. It is asserted some of the potential attendants of the meeting stated that they can not be present at the meeting. When it is asserted, the arranger of the meeting is notified some of the invited employees' schedules did not work out for that meeting.

11 – give edit rights

The give edit rights control signal is generated by the check user access rights process. It is asserted when the user has enough access rights to accomplish the desired task operation. When it is asserted, the user is let continue with the operation he wants.

12 – reject edit rights

The reject edit rights control signal is generated by the check user access rights process. It is asserted when the user does not have enough access rights to continue with the task operations he wants. When it is asserted, the user is prompted that his access rights are not enough to be allowed to accomplish the operations wanted.

13 – can edit

The can edit control signal is generated by the check dependency process. It is asserted when the modification request of a specific task does not violate the task dependency specifications among tasks. When it is asserted, the user is allowed to continue with the modification.

14 – can not edit

The can not edit control signal is generated by the check dependency process. It is asserted when the modification request of a specific task violates the task dependency specifications among tasks. When it is asserted, the user is prompted that the desired operation is not permitted because of the violation of the task dependency specification.

6. PROJECT ESTIMATIONS, PROCESSSS MODEL & TEAM STRUCTURE

6.1 PROJECT ESTIMATIONS

6.1.1 LOC-Based Estimation

The following estimates are based on “normalized-effort” estimation from previous programming experiences.

Functions	Estimated LOC
• Database Management	1000
• User Interface Development	5000
• Computer Graphics Facilities	1000
• Control Facilities	2200
Total Estimated Lines of Codes	9200

The estimates for LOC are plugged into the COCOMO formula for effort and duration estimation. The basic COCOMO model is used, for which

- Effort $E = a \text{ KLOC}^b$
- Duration $D = c E^d$

The project is classified as an organic project, using default values $a = 2.4$, $b = 1.05$, $c = 2.5$ and $d = 0.38$.

$$\begin{aligned} E &= 2.4(\text{KLOC})^{1.05} \\ &= 2.4(9.2)^{1.05} \\ &\approx 25 \text{ person-months} \end{aligned}$$

$$\begin{aligned} D &= 2.5E^{0.38} \\ &= 2.5(25)^{0.38} \\ &\approx 8 \text{ months} \end{aligned}$$

$$\begin{aligned} N &= E/D \\ &= 25/8 \\ &= 3 \end{aligned}$$

The results indicate that the project can be completed in 8 months with a project team of 3 people. Since we have 4 members in our team, the project duration will be shorter. It will take approximately 6 months.

6.1.2 FP-Based Estimation

Information Domain Value	Optimistic	Likely	Pessimistic	Est. Count	Weight	FP-count
Number of inputs	15	18	21	18	4	72
Number of outputs	12	15	18	15	5	75
Number of inquiries	7	10	14	10	4	40
Number of files	3	4	5	4	10	40
Number of external Interfaces	10	12	15	12	7	84
Count-total						311

Complexity Adjustment values

1. Backup and recovery	2
2. Data communications	4
3. Distributed processing	0
4. Performance critical	3
5. Existing operational environment	2
6. On-line data entry	4
7. Input transaction over multiple screens	3
8. Master files updated on-line	3
9. Information domain values complex	2
10. Internal processing complex	2
11. Code designed for reuse	1
12. Conversion/installation in design	2
13. Multiple installations	2
14. Application designed for change	2
Total	32

$$\begin{aligned}
 FP_{\text{estimated}} &= \text{total-count} \times (0.65 + 0.01 \times \sum F_i) \\
 &= 311 \times (0.65 + 0.32) \\
 &= 301.67
 \end{aligned}$$

Based on our estimation that the average productivity is 10.5 FP/ pm, we can say that the project will require an approximate effort of 28.7 person-months and with our team of 4 members it will take approximately 7 months.

6.2 PROCESS MODEL & TEAM STRUCTURE

6.2.1 Process Model

We will use linear sequential model while developing DProject Project Management Tool. This decision is supported by the following facts:

- We have approximately seven months to complete the project and along this time, the analysis, design, code generation, and testing phases all have to be completed sequentially.
- We have to produce our product once and for all, which makes prototyping disadvantageous.

Using the linear sequential model, we will have to complete the following tasks to complete the project:

- Requirements Analysis (this document), in which the nature of the program to be built will be described in detail. Specifically the information domain, required function, behavior, performance, and interface for the software will be described.
- Design phase, in which the data structures, software architecture, interface representations, and algorithmic detail will be specified using appropriate modeling techniques. Also, at the end of the design phase, a prototype for interface representation will be produced. Design phase will be accomplished until the end of December, 2004.
- Code Generation phase, in which the actual program will be produced. Code generation will be completed until the end of April, 2005.
- Testing phase, in which the software will be tested to identify whether it functions as it should. In this phase the System Requirements Specification will help us in validating the output product. The system will also be tested for potential bugs.
- Support phase, in which the produced software will undergo necessary changes. Although this is a necessity for general software development, in our case we do not plan a distinct support phase for our software. This is because the team members will not anymore be together after the project is complete, and it will be impossible to maintain support.

6.2.2 Team Structure

We will use democratic decentralized model as our team model. That is, all decisions will be made by group consensus, and the communication between group members will be horizontal. All team members will be responsible to all other team members. The communication will be realized by meetings, e-discussions, and in-group presentations. The facts supporting our decision for democratic decentralized model are as follows:

- Number of the team members is small, which makes communication easy.
- The team members have previous acquaintance, which makes decision taking processes more rapid.

- High volume of communication between team members is needed to complete the project which makes horizontal communication a must.

7. RISK MANAGEMENT PLAN

In order to come up with an error free system which addresses its initial goals, we must define some potential risks and devise methods to overcome them. Hence we should have a risk management plan to overcome the possible risks that may have different impacts on the project.

The main goal should be identifying and understanding possible risks, monitoring them, managing them and dealing with them with them. To be able to achieve this effectively, some methodology should be defined and used.

7.1 RISK MANAGEMENT ORGANIZATIONAL ROLE

Every member of the team should be responsible in managing risks. Every one should pay close attention to the project and all members of the team should maintain their effort to avoid possible risks. These duties can simply be states as follows:

- The estimations made at the beginning of the process should be revised frequently to avoid any miscalculations that can result in failure.
- The schedule should be traced carefully and must be strictly obeyed so that the product can be delivered on time.
- The team members should carefully examine the tools to be used and must be familiar with them so that there will not be any negative effects that are caused because of the inexperienced team members

7.2 RISK DESCRIPTION

- Development Risks:

The development risks are the ones that can occur throughout the development process and have negative effects on the development. To be able to have an efficient development process, there must be enough resources allocated to the team and the team should be well trained about the unclear technical issues.

- Employee Risks:

These risks depend on the technical ability, experience and enthusiasm of the team members. If the team members are not experienced enough about the subject or if there will be some staff leaving the project, this will have negative effects on the project.

- Product Size:

These risks depend on the team members incorrect estimations about the size of the final product and duration of the project. If the product size is not estimated correctly or the estimated time needed for the project is miscalculated, these will cause the risk of missing the deadlines or designing not a full-functionality product.

7.3 RISK TABLE

The risk table shows the risks that may causes a negative effect on the project. The risks are categorized according to the risk types defined before; also, their impacts and probabilities are stated. The risks are sorted according to their probabilities and their impacts. The Risk Mitigation, Monitoring and Management Plan (RMMM) will be prepared and the risks that are above the cut line will be considered in this plan. For the risks that are below the cut line, some additional plans (although not complicated as the RMMM) will also be developed so that their effects can be minimized.

Category	Risks	Probability	Impact	RMMM
Employee Risks	Lack of training and experience	30%	1	X
Product Size	Where size estimates could be wrong	30%	2	X
Development Risks	Insufficient resources	20%	2	
Employee Risks	Staff turnover	10%	1	
Development Risks	Lack of training on tools	10%	2	
Technology Risk	Obsolete technology	10%	3	

Impact Values

1
2
3
4

Description

Catastrophic
Critical
Marginal
Negligible

The table above shows all the possible risks, their chances of occurrence, their impacts and their categories. The impacts values are defined below the table. The risks are sorted after an evaluation of their possible effects on the project and sorted accordingly. The risks that are above the cut line that are marked with a (X) should be considered seriously and involved in the RMMM.

8. VALIDATION CRITERIA

Our project management tool DProject will be a web based tool (the only thing needed is a browser and a internet / network connection) so it won't load any burden to the user's

computer in that it won't use memory or cpu too much. The system will process incoming data and send relevant feedback within a few milliseconds of receiving it. However, since the users will use the system over the internet or their local area network, at some times users may not have such fast responses because of the bandwidth limitations. The system also will be able to deal with a maximum number of users (approximately 150) because it relies heavily on a database in which all the information will be kept. Also our tool will have the expected response time even at the times of heavy usage. For these reasons, the amount of data in web pages (especially size of the graphics) will be kept as small as possible so that it will load onto the user's computer very fast.

To validate our project the following tests (blackbox tests) must be done and the results should be satisfying. In other words our project has to conform the following specifications:

- New user creation and all the operations that can be done by user must be satisfied.
- Having easy and fast data entry through the browser.
- Quick response to the user request (so quick response to database queries)
- Checking whether file export creates correct file outputs and file import creates correct project objects.
- Checking if forum facility gives a reliable means of communication among users
- Checking whether meetings are arranged according to the user requests
- Validating that resources of the projects are allocated correctly
- Validating that the tasks can be assigned, edited, managed, and given account of reliably and securely.

9. PROJECT SCHEDULE

The project schedule did not fit in an A4 paper, so it can be found in the Appendix presented on a A3 paper layout.