

**PROJECT WORKSPACE: A COMPUTER SUPPORTED  
COLLORATIVE WORK TOOL FOR SOFTWARE  
PROJECT DEVELOPMENT**

A Thesis

Presented to the Institute of Science and Engineering

of Işık University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

in

The Department of Computer Engineering

by

Tuncay Çoruh

July 2005

Approval of the Institute of Science and Engineering

\_\_\_\_\_  
( Prof. Hüsnü Ata Erbay )

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of  
Master of Science.

\_\_\_\_\_  
( Prof. Selahattin Kuru )

Head of Department

This is to certify that I have read this thesis and that, in my opinion, it is fully  
adequate in scope and quality as a thesis for the degree of Master of Science.

\_\_\_\_\_  
( Prof. Selahattin Kuru )

Supervisor

Examining Committee Members.

.....

\_\_\_\_\_

.....

\_\_\_\_\_

.....

\_\_\_\_\_

.....

\_\_\_\_\_

.....

\_\_\_\_\_

## **ABSTRACT**

### **PROJECT WORKSPACE**

**Tuncay Çoruh**

Software project development is a very serious job. Most of the project budgets are very big and most of them do not finish. So, the need to improve our software development processes is very real. Development side is not the only source of the software project failures. The main problem is the communication between the sponsor and the development side. The project should have some critical points to start the project: customer needs, delivery time, budget, resources etc. Project Workspace can help you on this point to manage requirements, issues, and files.

Key words: CSCW, Workspace, release, requirement, issue.

## **ÖZET**

### **PROJE ÇALIŞMA ALANI**

**Tuncay Çoruh**

Yazılım projesi geliştirme çok önemli bir iştir. Projelerin büyük bir kısmının bütçesi çok fazladır ve bu projelerin büyük bir bölümü başarısızlıkla sonuçlanmaktadır. Bundan dolayı yazılım projesi geliştirmede ilerleme kaydetmemiz gerekir. Bu başarısızlıkların sorumlusu sadece geliştirme yapanlar değildir. Ana problem sponsor ve geliştirme takımı arasındaki iletişimden kaynaklanmaktadır. Projeye başlamadan önce bazı önemli kritik noktalar çok iyi bir şekilde tanımlanmalıdır: müşteri gereksinimleri, proje teslim tarihi, bütçe, kaynaklar vs. Proje Çalışma Alanı bu konuda müşteri isteklerinin, projede çıkacak sorunların ve proje dosyalarının yönetimi için yararlı olabilir.

Anahtar kelimeler: Bilgisayar destekli ortak çalışma, proje çalışma alanı, paket, istek, sorun.

## **ACKNOWLEDGEMENTS**

I would like to express sincere appreciation to Prof. Selahattin Kuru for his guidance and advice throughout the thesis work. Thanks to my project manager from my job Ercan Ardal for his suggestions and comments in project management. Special thanks to my friend Erhan Polat for his design help. Finally, I would like to thank to my family and to my friends Ekrem Çağlar, Uğur Güven and Yasemin Sun for their help and support.

## TABLE OF CONTENTS

ABSTRACT .....	III
ÖZET .....	IV
ACKNOWLEDGEMENTS .....	V
TABLE OF CONTENTS .....	VI
LIST OF FIGURES .....	VIII
LIST OF TABLES .....	X
CHAPTER 1 INTRODUCTION .....	1
1.1. CSCW .....	1
1.2. CRITICAL POINTS IN A PROJECT .....	1
1.3. THESIS OUTLINE .....	2
CHAPTER 3 COLLABORATIVE WORKING AND SHARED WORKSPACES ...	3
2.1. COLLABORATIVE WORKING.....	3
2.2. SHARED WORKSPACES.....	4
2.2.1. Groove.....	4
2.2.2. SpeeDEV.....	6
CHAPTER 4 PROJECT WORKSPACE.....	9
3.1. MODULES.....	10
3.1.1. Workspace.....	10
3.1.2. Users.....	10
3.1.3. Release .....	12
3.1.4. Requirement .....	13
3.1.5. Issue .....	15
3.1.6. File .....	17
3.1.7. Discussion .....	17
3.2. TECHNOLOGY AND ENVIRONMENT.....	18
CHAPTER 5 AN EXAMPLE USE .....	20
CHAPTER 6 EVALUATION .....	27
CHAPTER 7 CONCLUSION.....	30
6.1. SUMMARY.....	30
6.2. FUTURE WORK.....	30
REFERENCES.....	31
APPENDIX A. USE CASES .....	33

A.1.	UC1 – Login.....	33
A.2.	UC2 – Workspace .....	36
A.3.	UC3 – Releases .....	40
A.4.	UC4 – Requirements.....	44
A.5.	UC5 – Issues .....	48
A.6.	UC6 – Files .....	52
A.7.	UC7 – Discussion.....	56
A.8.	UC8 – Users .....	59
A.9.	UC9 – Admin.....	62
APPENDIX B. CD CONTENT .....		66

## LIST OF FIGURES

Figure 2.1 – Evolution of Interaction.....	3
Figure 2.2 - Groove Virtual Office .....	5
Figure 2.3 – SpeeDEV .....	7
Figure 3.1 – Workspace Page.....	11
Figure 3.2 – Users Page .....	12
Figure 3.3 – Releases Page .....	13
Figure 3.4 – Requirements Page .....	14
Figure 3.5 – Issues Page.....	16
Figure 3.6 - Files Page .....	18
Figure 3.7 – Discussion Page.....	19
Figure 4.1 – Add user.....	20
Figure 4.2 – Create workspace.....	21
Figure 4.3 – User selection.....	21
Figure 4.4 – Add release .....	22
Figure 4.5 – Add requirement .....	23
Figure 4.6 - Files .....	24
Figure 4.7 – Add Issue .....	25
Figure 4.8 - Discussion .....	26
Figure A.6.1 – Login.....	33
Figure A.6.2 - Login (Empty username and password error) .....	33
Figure A.6.3 – Login (Wrong username or password error) .....	34
Figure A.6.4 – Use case diagram of login.....	35
Figure A.6.5 - Workspace .....	37
Figure A.6.6 – Use case diagram of workspace.....	38
Figure A.6.7 - Releases .....	41
Figure A.6.8 – Use case diagram of releases .....	43
Figure A.6.9 - Requirements.....	45



Figure A.6.10 – Use case diagram of requirements .....	47
Figure A.6.11 - Issues .....	49
Figure A.6.12 – Use case diagram of issues .....	51
Figure A.6.13 - Files .....	53
Figure A.6.14 – Use case diagram of files .....	54
Figure A.6.15 – Discussion.....	56
Figure A.6.16 – Use case diagram of discussion .....	58
Figure A.6.17 - Users .....	60
Figure A.6.18 – Use case diagram of users.....	61
Figure A.6.19 - Administration.....	63
Figure A.6.20 – Use case diagram of administration.....	64

## **LIST OF TABLES**

Table 5.1 Project Workspace vs Groove.....	27
--	----

## **CHAPTER 1 INTRODUCTION**

### **1.1. CSCW**

Growth of the computer integration and communication produces the Internet age. The Internet and developing network technology make people work together and share information over imaginary world to cope with big projects. This working system was created by Computer Supported Collaborative Work(CSCW) idea. CSCW is a specific discipline that guides thoughtful and appropriate design and development of computer systems to support group work[1][2][3].

Collaboration is the collective work of two or more individuals where the work is undertaken with a sense of shared purpose and direction that is attentive, responsive, and adaptive to the environment[4].

Collaborated system work is not a simple system. Individual teams and groups from all over the world can be the partner of this group and can work on their tasks individually. To make the system work some rules should be set. This is a must to get the partners together, to provide them work conveniently and to make them understand each other.

### **1.2. Critical Points in a Project**

Today's software technology provides many advantages to make things easy. The companies all over the world want to benefit from these advantages to ease processes, to provide better services, to reduce costs etc. Because of that these companies give software projects to software companies or their own software development departments. In both there are a customer who wants the software project to be done and a team who will develop this project. The communication between these both sides should be well defined for the success of the project.

There should be some critical points to start the project: customer needs, delivery time, budget, resources etc. All these critical points should be well defined for healthy progress of the project development. Because customer always changes the requirements of the project but the time and budget are not changed. Furthermore, success is most often measured by short-term deliverables, not by long-term maintainability[5]. So, all possible deliverables and requirements should be planned and the critical points should be managed to prevent possible failures. This is a must for professional project development cycle. Project Workspace can help you on this point to manage requirements, issues and files.

### **1.3. Thesis Outline**

This thesis focuses on a shared workspace tool which provides computer supported collaborative work and communication between sponsor and development side. The rest of this thesis is organized as follows. Chapter 2 examines collaborative working and other shared workspaces. Chapter 3 presents a solution to a shared workspace: Project Workspace. Chapter 4 presents an example use of the Project Workspace. Chapter 5 is evaluation and Chapter 6 presents conclusion and future work.

## CHAPTER 2 COLLABORATIVE WORKING AND SHARED WORKSPACES

### 2.1. Collaborative Working

Working is the key of developing and always opens doors to more complicated new working areas. While the process is in advance, working individually becomes more difficult and at a point becomes impossible. Solution is collaboration; people to people interaction about data. This means that collaboration is more than just the ability for one or more people to access data or status information[6].

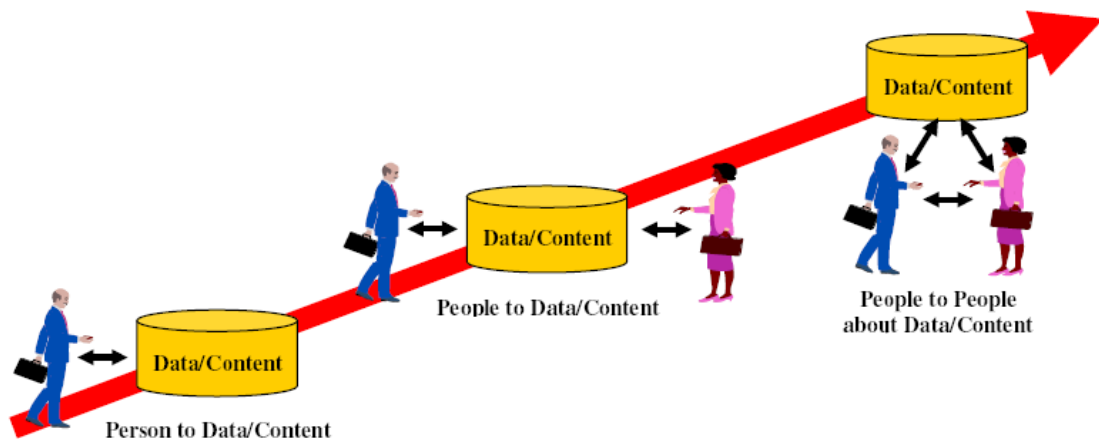


Figure 2.1 – Evolution of Interaction

Collaboration is becoming an important key element of designing and building complex systems. Collaboration is required to solve the branches of the complexity by collecting experts of different skills and to provide the customer requests for better, faster and cheaper delivery. In simple terms, collaborative working involves using information systems to enable individuals or groups of individuals to work concurrently on information, no matter whether they are dispersed or co-located[7].

## **2.2. Shared Workspaces**

A Shared workspace is a software tool to share information for a collaborated work. Today's computer and communication technology provide individuals and individual groups to share information and work together from independent places.

Benefits of a shared workspace:

- Communication environment for the members.
- Shared information and knowledge for collaboration work.
- Faster access for the latest version of the information.
- Time and cost savings

A shared workspace tool should have at least the following features to provide the collaboration work:

- Can be accessed from everywhere – Internet technology should be used.
- Should be secure to keep the information.
- Should have some set of rules to form the boundaries of sharing information. Because the members can be from different countries and cultures, their approaches to the projects cases can be different. Defined set of rules will provide the members to understand the correct meaning from the shared information.

### **2.2.1. Groove**

Groove Virtual Office (Figure 2.1) is a Windows based, peer to peer application. It is software that allows your team to work together as if you were all in the same physical location. Teams of people at organizations of all sizes and types are using Groove Virtual Office to speed the way they get work done - from simple file sharing, to formal and informal projects, to larger-scale business processes[8].

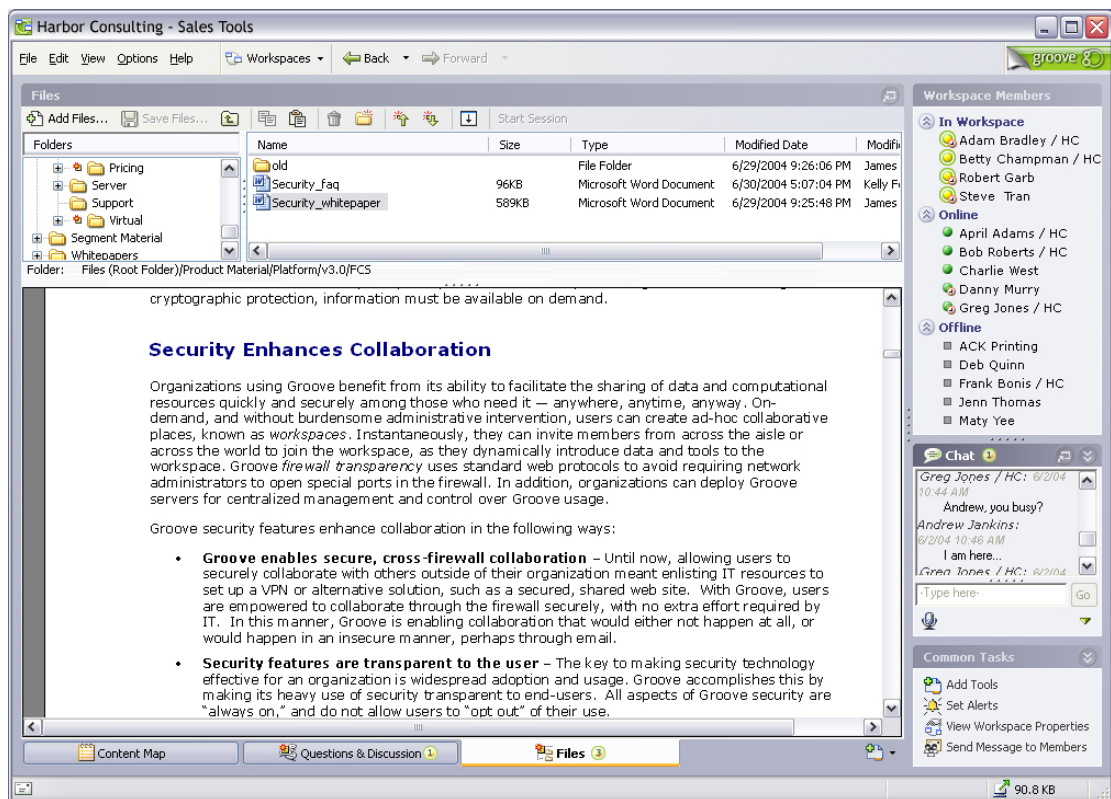


Figure 2.2 - Groove Virtual Office

After a user installs Groove Virtual Office, he/she can create a workspace and invite other Groove users to the workspace or join to a workspace. There are two types of users: manager and participant. Manager has read, write, change and delete rights whereas the participant has read, write and change rights. Modules have read/unread property of the items. If a new item is created or an item is changed then a star character Modules can be added to Groove as plug-in. The basic modules of Groove are contacts, file sharing, discussion and calendar:

- **Contacts:** User can search Groove user database and can add the users to the contact list. Additionally, the user can see the users of the workspace that he/she is joined.

- **File Sharing:** Groove has a file management module that users can share files. User can synchronize these files with their local files.
- **Discussion:** Users can follow the discussions on this module. They can open new discussion and reply to a discussion.

### 2.2.2. SpeeDEV

SpeeDEV (Figure 2.2) Software Developer Life Cycle (SDLC) is a web based shared workspace. SpeeDEV provides a Collaborative Development Lifecycle Management solution for software development teams and engineering organizations to manage the entire process of defining, designing, developing and deploying technology applications[9].

SpeeDEV's approach encompasses certain known islands of technology that are necessary for engineering management success. The independent point solutions for Requirements Management, Issue and Defect Tracking, Test Case and Test Plan Management, Project and Task Management, and Process Automation are rolled into a modular solution, that can expand as needed and integrate with other technologies.

- **Requirements Management:** Requirements are the logical starting point for any SDLC. It is only recently that organizations are realizing the importance and critical nature of Requirements Management. The complete SDLC can be seen as the aggregate of lifecycles of all the different kinds of Requirements from their inception to closure.



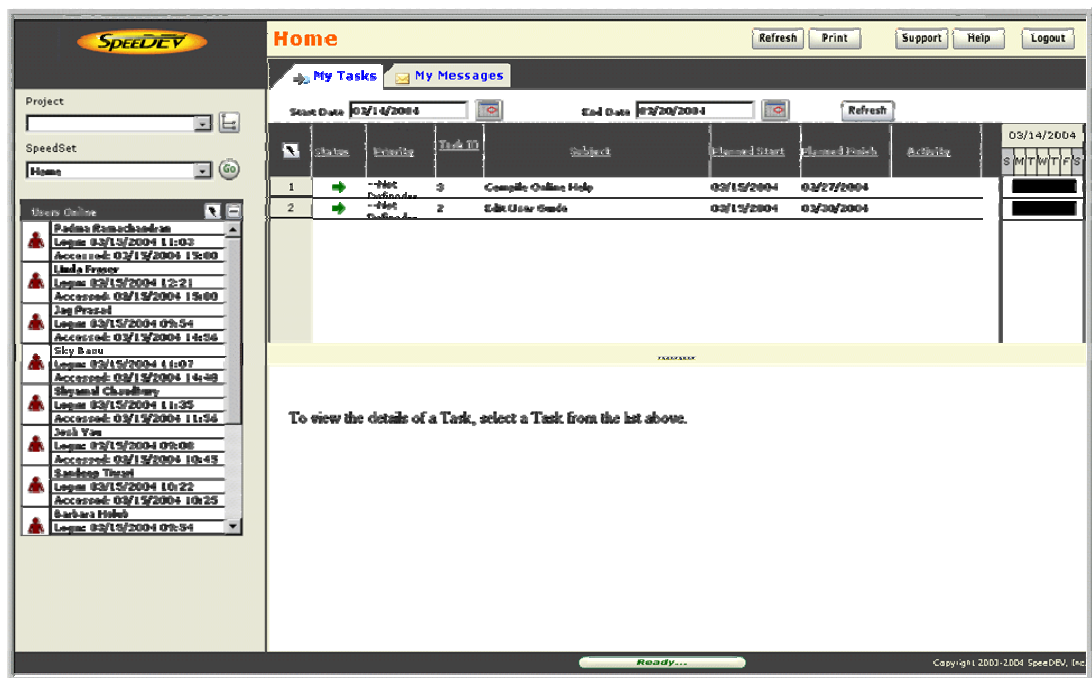


Figure 2.3 – SpeedDEV

- Risk Management:** The ever-increasing complexity of software products introduces various risk factors. Recognition, management and mitigation of these risks are often the key to success of a software project. Stand-alone Risk Management applications are inadequate and normally divorced from the notion of a Risk Process and often the importance of interrelationships between Requirements, Issues and Risks are overlooked.
- Issue Management:** One of the earliest entities recognized to be synonymous to quality, or rather lack of it, is defect or bugs. It is no surprise that the software industry has tried to address this issue for that last 30 years by developing various bug and defect tracking tools. However, with the realization that Quality is a process, rather than an end product has been the prime mover in accepting the SDLC as the solution

rather than stand-alone defect tracking tools. In SDLC, Issues are of various kinds, not just technical defects. It is important to track them the beginning of a project rather than in the development-test cycle only, as is customary for defect tracking tools. The interrelationship and traceability of Issues to Requirements, Tests and Risks has increased the value of Issue Tracking manifold over standalone defect tracking systems.

- **Release Management:** In recent years, rarely is software development is done in a linear or sequential fashion. Following the Rapid Development model, developers try to release the more important features as soon as possible. Multiple releases occur in the lifetime of a software, sometimes even with overlapping lifecycles. Release management has similar needs as project management, but typically a single Project may include a number of releases with entities like different versions of Requirements or Tests. Also, as later releases fix issues from previous ones, they become interrelated through their common Issues.

## CHAPTER 3 PROJECT WORKSPACE

The Web based Project Workspace enables development team and customer to work together and communicate each other regardless of the location. The Project Workspace system is secure and available from anywhere without installing any client applications. The structure of the system is based on extreme programming (XP) method. Extreme programming has 4 main steps:

- **Simplicity:** Development should be as simple as possible, but not simpler. From the side of the Project Workspace System, just meet all the requirements and deliver the release.
- **Communication:** Most projects have more than one person involved, and communication among these individuals is critical to project success, whether they are developers or some other kind of stakeholders. Extreme Programming emphasizes face-to-face, person-to-person communication[10]. Project Workspace System provides a reliable communication platform to follow the progress in project development.
- **Feedback:** Complex efforts such as software projects work far better if we steer them frequently. Feedback is very important to determine the direction of the project; it resolves issues and unclear points. Feedback can be done by using Issues and Discussion module in Project Workspace System.
- **Courage:** Because it is rich in communication and feedback, XP fills project stakeholders with courage and confidence about how things are going. But XP also calls for courage in the first place: the courage to do simple things, to communicate openly and frequently, and to trust our fellow stakeholders on

the project[10]. Both customer and development side can easily follow the progress of the project from Project Workspace System. By the way both sides feel confident about progress of the project.

### 3.1. Modules

#### 3.1.1. Workspace

A workspace (Figure 3.1) is a place where people with common purposes and goals can share and access project information. This module is an electronic shared workspace which enable selected group of users to share information, files and to discuss about the project[11]. A workspace begins with the person who creates it and then invites others to join.

Workspace is the top level of this system. A set of people, who have access to the workspace, can benefit from the system, such as creating release, requirements, downloading project files etc. Project communication platform starts with the creation of a workspace of the project. Please see the Appendix A (A.2 UC3 - Workspace) for the usage of this module.

#### 3.1.2.Users

User is the one who uses the workspace, in other words node of the workspace communication platform. Workspace users are determined from Users module (Figure 3.2). Please see the Appendix A (A.8 UC8 - Users) for the usage of this module. When a workspace is created by a workspace manager, he/she adds the user(s) of the workspace by giving workspace roles to the system users. The roles are:

- **Participant:** Read right
- **User:** Read and add rights
- **Leader:** Read, add and change rights

- **Manager:** Read, add, change and delete rights

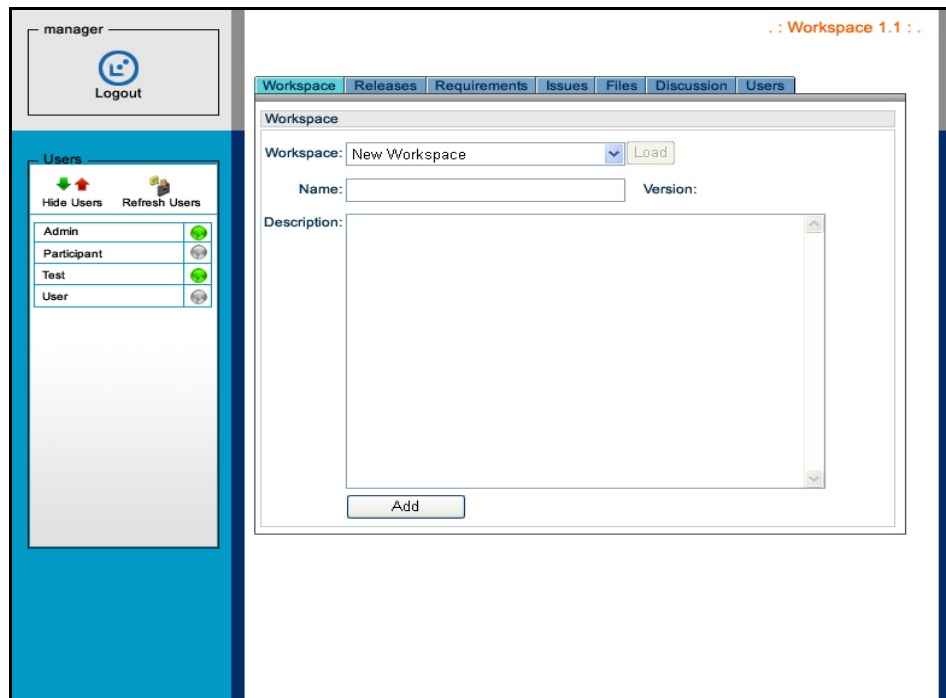


Figure 3.1 – Workspace Page

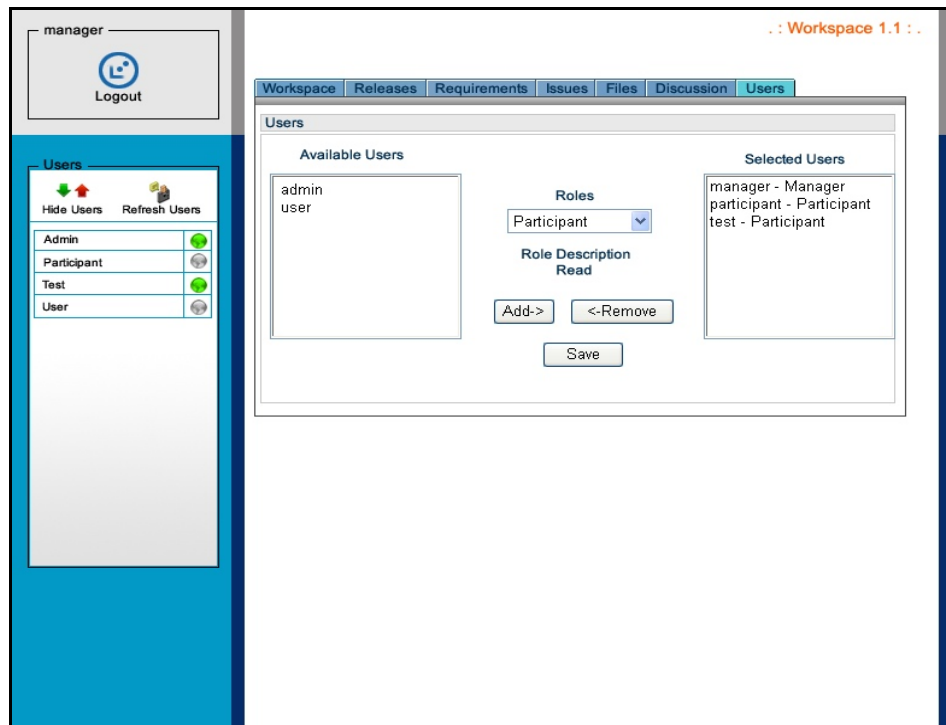


Figure 3.2 – Users Page

### 3.1.3.Release

Release is a software package which meets the requirements. All workspaces should have one or more releases (Figure 3.3). After the creation of the workspace users define necessary release(s). This part shows the general points of the projects. It includes requirements for detailed information. Please see the Appendix A (A.3 UC3 - Releases) for the usage of this module.

#### Key Features:

- All releases have status of “Not Started”, “In progress” or “Completed”.
- If a release is changed then a new version of the release is created.
- Differences between the versions are highlighted to follow the changes.
- New or changed releases can be determined by unread icon.

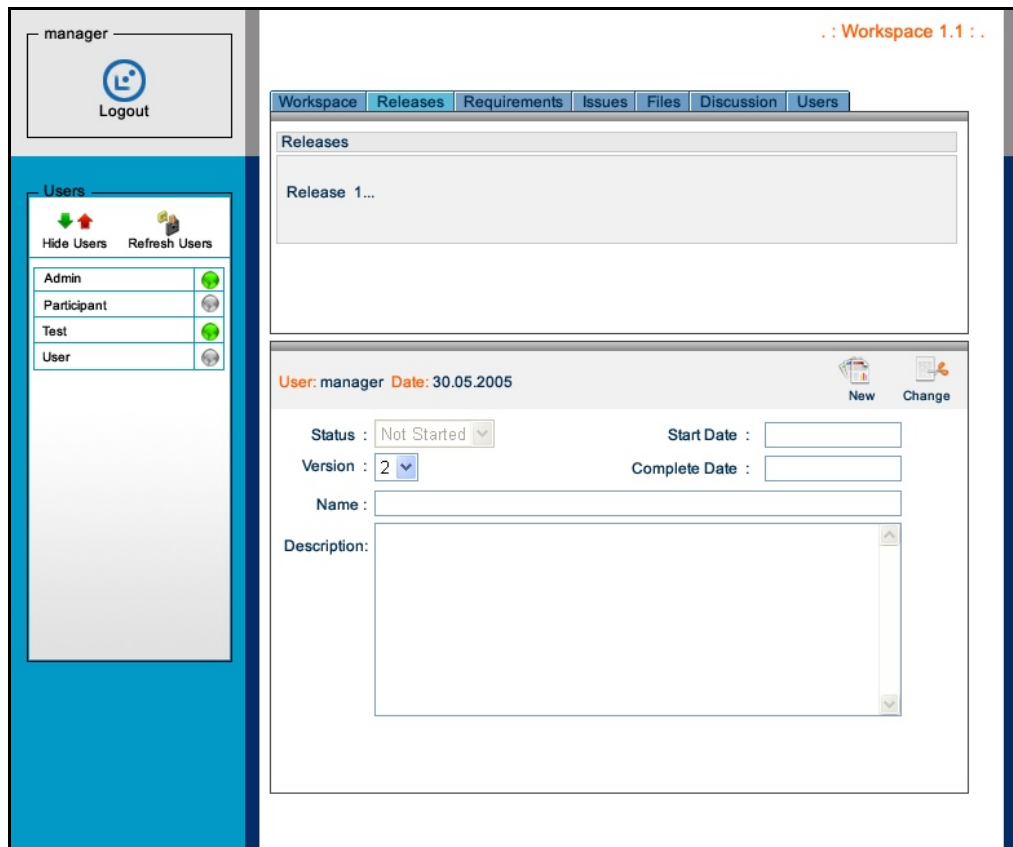


Figure 3.3 – Releases Page

### 3.1.4.Requirement

Requirements are the things that the project must deliver, are the objectives to fulfill the goals of the sponsor, the person who wants the project(Figure 3.4). For instance customer has an objective of monthly report services for the performance of their workers. This is a requirement for a project but it is not clear enough. So, the requirements are also the detailed view of the objectives, answers to the questions, such as “What type of report do you want?”.

Requirements are very important for the achievement of the project; meeting all the requirements will finish the project successfully[12]. According

to Steve McConnell in *Software Project Survival Guide*, “The most difficult part of requirements gathering is not documenting what the users 'want'; it is the effort of helping users figure out what they 'need' that can be successfully provided within the cost and schedule parameters available to the development team.”[13]. This module provides to understand and keep all requirements together.

Users can see all requirements of a release from the requirement list and can follow the status and the change history of a requirement from requirement properties below the requirement list(Figure 3.4). Please see the Appendix A (A.4 UC4 - Requirements) for the usage of this module.

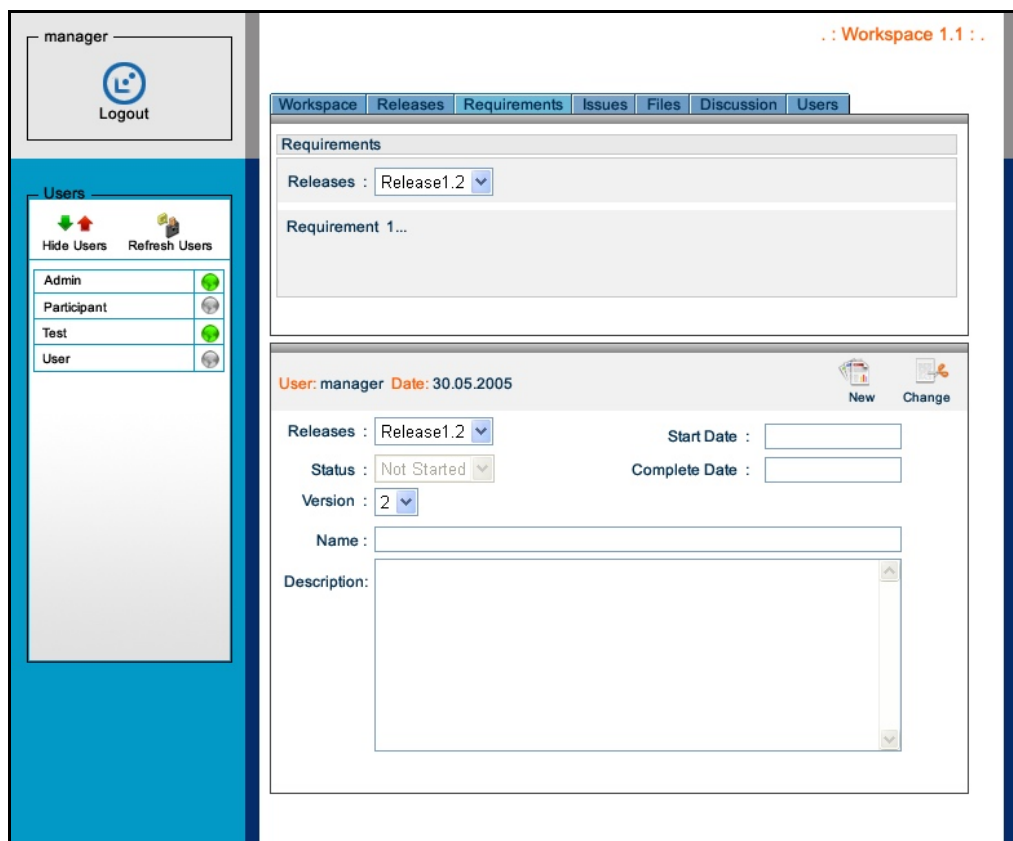


Figure 3.4 – Requirements Page



**Key Features:**

- All requirements have status of “Not Started”, “In progress” or “Completed”.
- If a requirement is changed then a new version of the requirement is created.
- Differences between the versions are highlighted to follow the changes.
- New or changed requirements can be determined by unread icon.

**3.1.5. Issue**

An issue exists when there is a gap between stakeholder expectations and an organization’s policies, performance, products or public commitments. Issue management is the process used to close that gap[14]. This module helps to manage the issues of the system.

Issues are unclear and/or wrong parts of the defined requirements. If there are questions on a requirement or it is not clear enough then you cannot meet that requirement. This module is very important to understand the requirements(Figure 3.5).

Issues do not have an impact on the defined projects product; scope, definition, or specification. Issues are identified in the form of questions, problems, or suggestions raised by the project team, management, or contractor. They can often affect the status of tasks or deliverables, which would result in changes to cost and schedule[15]. Please see the Appendix A (A.5 UC5 - Issues) for the usage of this module.

### Key Features:

- All issues have status of “Not Started”, “In progress” or “Completed”.
- If an issue is changed then a new version of the issue is created.
- Differences between the versions are highlighted to follow the changes.
- New or changed issues can be determined by unread icon.

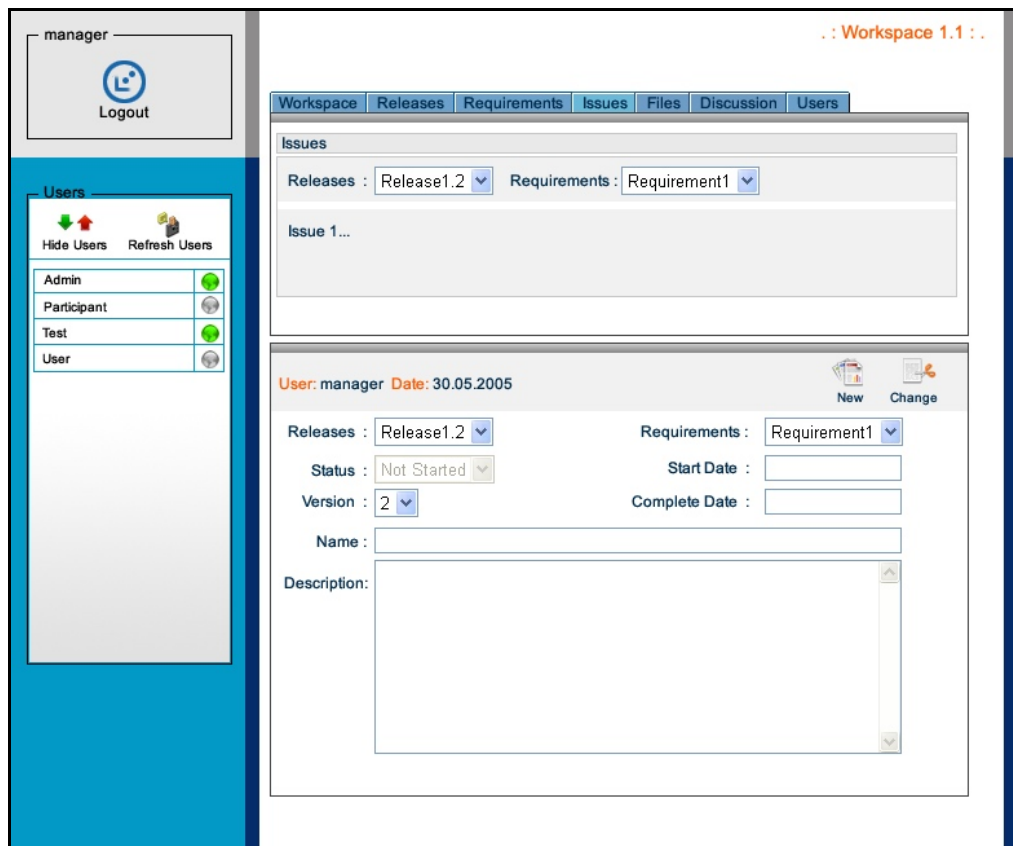


Figure 3.5 – Issues Page

### **3.1.6.File**

This module provides file sharing to the workspace users, such as project documentation, release packages, images etc(Figure 3.6). Please see the Appendix A (A.6 UC6 - Files) for the usage of this module.

#### **Key Features:**

- Users can add new folders and files with description.
- Users can delete files and/or folders.
- Users can download files.

### **3.1.7.Discussion**

Workspace users can discuss everything under the discussion sections: workspace, releases, requirements, and issues(Figure 3.7). Please see the Appendix A (A.7 UC7 - Discussion) for the usage of this module.

#### **Key Features:**

- Users can add discussions to the workspace, release, requirement, Issue sections.
- Users can reply to an added discussion.
- Discussions are ordered as tree form.

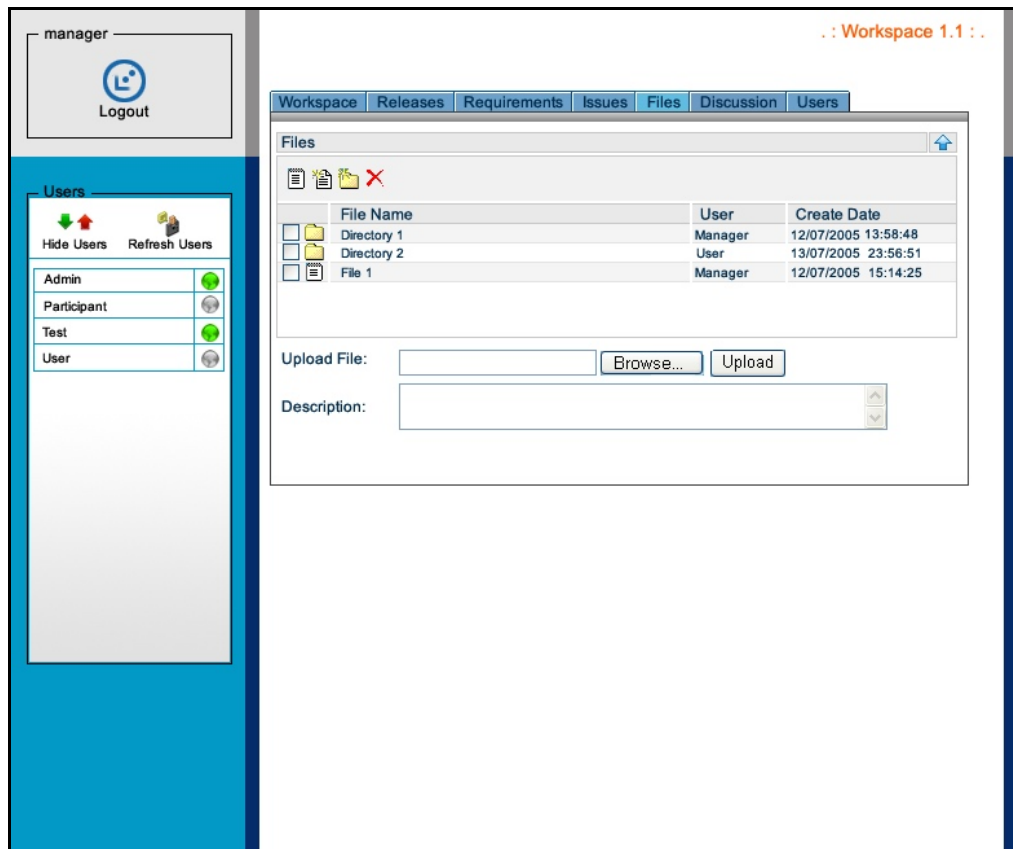


Figure 3.6 - Files Page

### 3.2. Technology And Environment

Project Workspace system is a Web based application. No client software installation is required. ASP.NET and C# are used to develop this application. Microsoft SQL 2000 is used for database side. The following environment is required for the system:

- **Server:**
  - Windows 2000 or up.
  - IIS 5 or up.
  - .NET Framework 1.1
  - Microsoft SQL 2000

- **Client:**
  - Internet Explorer 5.5 or up

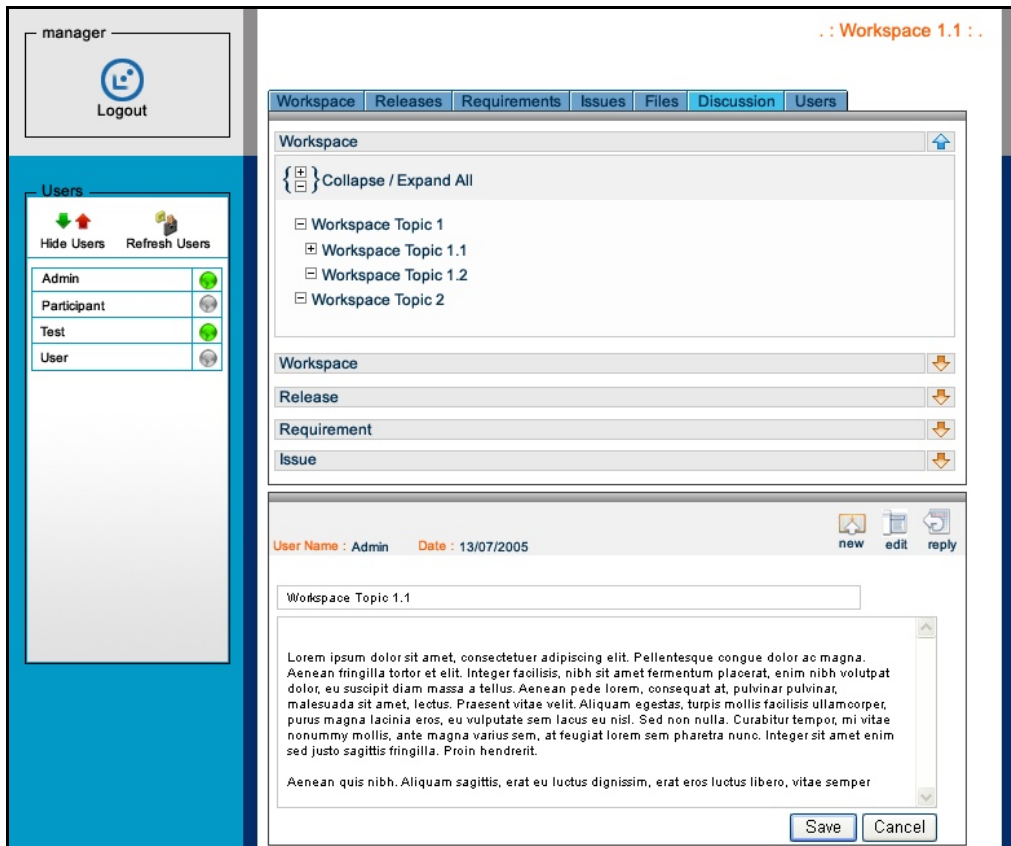
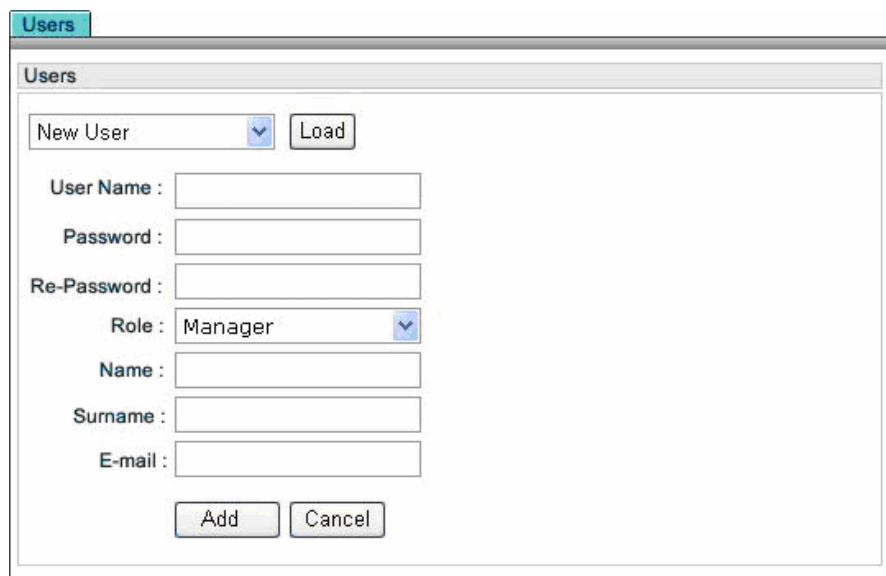


Figure 3.7 – Discussion Page

## CHAPTER 4 AN EXAMPLE USE

In this scenario we consider that Project Workspace will be used for a software project development. There will be two sides: customer and development team. Customer, named as Depo, is a large clothing store which has many branch stores. Customer wants a web page for its branch stores. After project analysis and contract the project starts.

First of all, administrator of the Project Workspace adds the required users to the system(Figure 4.1). A user who has a “Manager” right creates a workspace with a name of “Depo Branch Services” and enters a general definition of the project(Figure 4.2). In Project Workspace system a project always starts with the creation of the workspace. After the creation of the workspace the workspace manager (owner) should add the users to the workspace by assigning workspace roles(Figure 4.3). Now, the base of the project communication environment was completed.



The image shows a software dialog box titled "Users". At the top left, there is a tab labeled "Users". Below the title bar, the word "Users" is repeated. The main area contains a "New User" dropdown menu and a "Load" button. Below these are several input fields: "User Name", "Password", "Re-Password", "Role" (with a dropdown menu currently showing "Manager"), "Name", "Surname", and "E-mail". At the bottom of the dialog are two buttons: "Add" and "Cancel".

Figure 4.1 – Add user

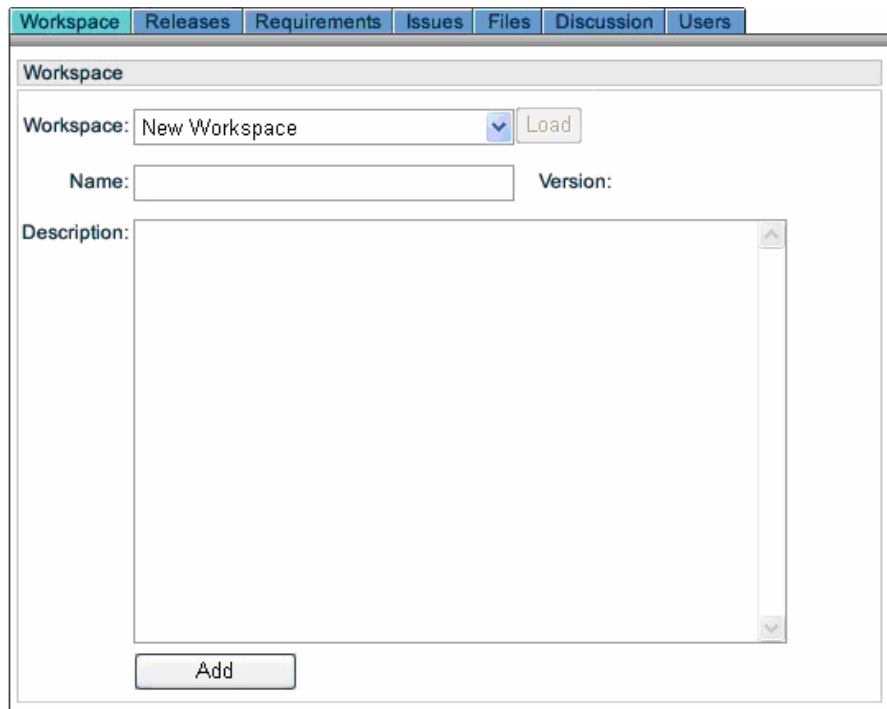


Figure 4.2 – Create workspace

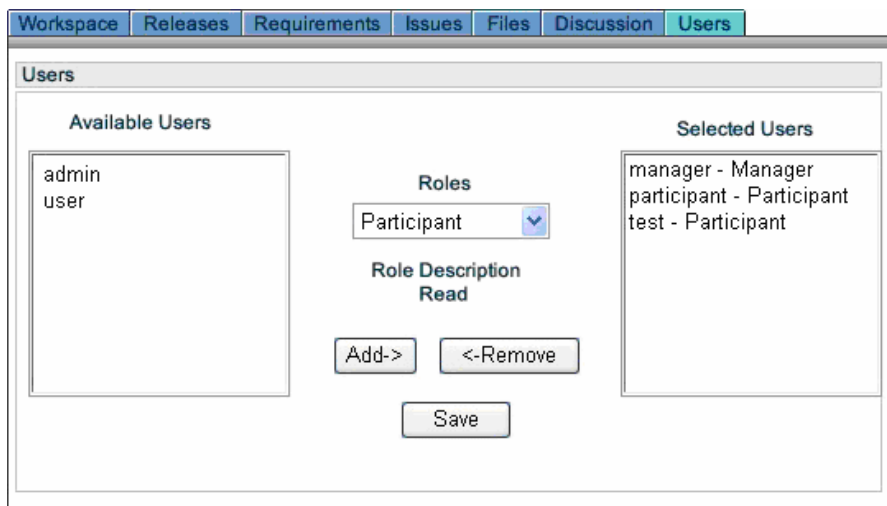


Figure 4.3 – User selection

A user, from the customer side, who has “Add” right, should create a new release for the first part of the project(Figure 4.4). Customer wants to provide weekly sales reports to their branch offices via a web page for the first part of the project.

The screenshot shows a software application window with a menu bar at the top containing 'Workspace', 'Releases', 'Requirements', 'Issues', 'Files', 'Discussion', and 'Users'. Below the menu is a 'Releases' section with a list containing 'Release 1...'. The main form area has a header with 'User: manager Date: 30.05.2005' and 'New Change' buttons. The form fields are: Status (Not Started), Start Date, Version (2), Complete Date, Name, and Description.

Figure 4.4 – Add release

Then requirements for the release should be created by the user who is from customer side(Figure 4.5). Some requirements can be as below:

- Requirement Name: Environment and Technology  
Description: ASP.NET and MSSQL 2000 will be used on Windows 2000 server.
- Requirement Name: Login System  
Description: Passwords will not be kept in database.  
After the three wrong login tries, the IP of the client will be blocked.
- Requirement Name: Report  
Description: A weekly sales report for the branch stores. All branch reports can only see their own report. Reports will be generated on demand in PDF format; they will not be kept in server. Data will be gotten from database in XML format.



Schema of the XML file, reportData.xsd, is in Files module under “Weekly Sales Report\Required Files” directory (Figure 4.6).

Figure 4.5 – Add requirement

After the creation of the requirements, the release works can start. Development team should meet all the requirements, so they should completely understand the requirements. If they are obscure then issues should be created for the requirements which are unclear (Figure 4.7). Some issues can be as below:

- Issue Name: Passwords  
Description: Where should we keep the passwords, in Web.config, in a file or in database as hashed?  
Requirement: Login System

- Issue Name: Design of the report  
Description: Is there any example report?  
Requirement: Report

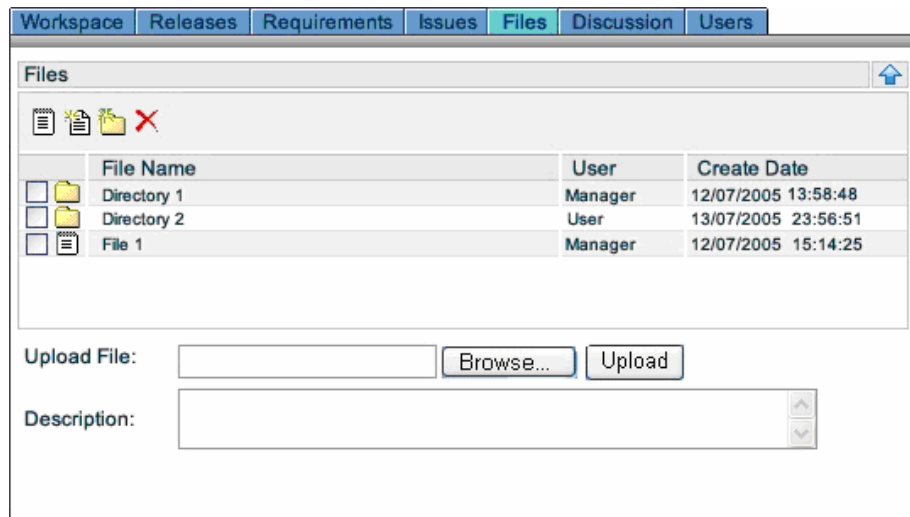


Figure 4.6 - Files

Because of the issues the related requirements will not be met. So the users from the customer side should solve the issues. Maybe some of the issues should be discussed; they can be added to the Discussion module(Figure 4.8). After the agreement on the discussed item the solution is added to the related issue and the issue status is set to “Completed”.

After all requirements are met the release will be completed. Details of the modules are explained with use cases in Appendix A.

Workspace Releases Requirements **Issues** Files Discussion Users

Issues

Releases : Release1.2 Requirements : Requirement1

Issue 1...

---

User: manager Date: 30.05.2005 New Change

Releases : Release1.2 Requirements : Requirement1

Status : Not Started Start Date :

Version : 2 Complete Date :

Name :

Description:

Figure 4.7 – Add Issue

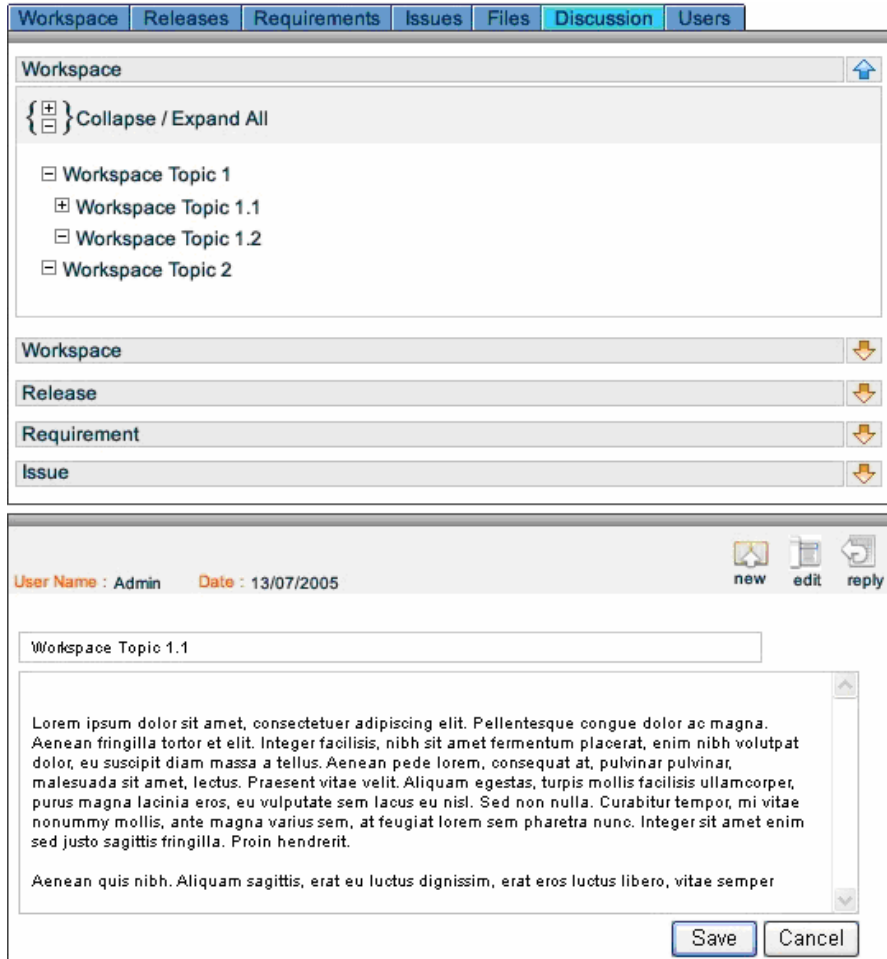


Figure 4.8 - Discussion

## CHAPTER 5 EVALUATION

This system was developed to provide a shared workspace for management of a collaborative work for the software projects in a simple way, with the following features:

The system is location independent. Users can access the workspaces via the internet. This is a web based application, so no extra client software installation is required; just internet explorer 5 or up is required as browser.

Security is very important for the web applications which are open to the internet. This system was developed to be ready for possible attacks. System is ready for SQL injection attacks. Also user passwords are kept in database in encrypted form and HTTP query strings (information in browser address bar) are sent to the server as encrypted.

The general web site vulnerabilities (SQL injection, cross-site scripting and unrestricted directory listings) were tested. SQL injection vulnerability could lead to a site's entire back-end database being downloaded by a hacker. Cross-site scripting occurs when hackers embed malicious JavaScript code into a site's dynamically generated pages, affecting the machine of any user that views that site. Unrestricted directory listings can be exploited by attackers to gain access to data that was not intended to be viewable to unauthenticated users[16, 17].

Generally the shared workspaces, which are based on software project management, provide a freeze property for the release, requirement and issue modules. This property freezes the related item and prevents the users from updating the item or from entering a new item. In Project Workspace system versioning is used instead of freezing. By the way, users can still update the item or enter a new

item, but actually a new version of that item is created. So, the users can see all versions of the item with change date and the name of the user who performs the update process.

Users can also follow the item changes by using the versions. Differences between the versions can be determined by the highlighted changed information. Also new or changed items are determined by new icon.

This system provides an easy project management, communication and information sharing platform with compare to the others(Table 5.1). On the other hand it has some disadvantages because of the following lack of features:

A task management module can be added to the system to assign requirements, issues to the members and to follow the status of the tasks. Task management is the management of the collaborative work. The project is split into small parts and assigned to the users. Task management in software development involves, among others, decomposing the project into smaller units, assigning tasks to developers according to their expertise, and creating a development schedule[18].

A messaging module can be added to provide an easy communication between the members. By the help of this feature, users can send messages to other users to acquaint them for the quick information, or issues.

A report module can be added to get detailed reports to follow the release, requirement, issue, task and time statuses. By the help of these reports, users can follow the whole project status by querying, for instance, completed requirements, open issues, not started releases etc.

**Table 5.1 Project Workspace vs Groove**

<b>Feature</b>	<b>Project Workspace</b>	<b>Groove</b>
Client software	Internet explorer 5 or up is required(web based application)	Client software is required (Windows based application)
Offline usage	No	Yes
Workspace, release, requirement and issue management	Yes	No
File sharing	Yes	Yes
Discussion	Yes	Yes
New item alert	Yes	Yes
Task Management	No	With add-in
Messaging	No	Yes

## **CHAPTER 6 CONCLUSION**

### **6.1. Summary**

Collaboration is an important element of success for complex projects. A number of skills can be required to finish a project. To collecting these skills set defines collaborated work. But collaborated work generates new problems: communication between the members, management of the communication and management of the shared information.

Today's computer and communication technology provide sharing and accessing information from all over the world. This causes to arise new ideas to bring collaborative work and sharing information together via the help of computer technology: Computer Supported Collaborative Work.

This thesis focuses on communication between sponsor and development side, critical points on software project development cycle and managing these critical points by using a Web based Project Workspace system. This system considers workspace, release, requirement, issue hierarchy and file management.

### **6.2. Future Work**

This project works on the critical points: customer needs, problems and file sharing. Workspace, release, requirement, issue and file modules were developed to manage these critical points. Further research can be done on the critical points resource, delivery time and messaging by developing task and scheduling management modules and messaging system for the members. Also a report module can be added to get detailed reports of status of the project.



## REFERENCES

- [1] Saul Greenberg, Computer Supported Cooperative Work and Groupware, Academic Press, London, 1991.
- [2] Masood Masoodian, Human to Human Communication Support for Computer Based Shared Workspace Collaboration,  
<http://www.cs.waikato.ac.nz/~masood/thesis/thesis.pdf>
- [3] Jonathan Grudin,  
<http://www.ics.uci.edu/~grudin/Papers/IEEE94/IEEEComplastsb.html>
- [4] Michael M. Beyerlein, Cheryl Harris, Guiding the Journey to Collaborative Work Systems: A Strategic Design Workbook (Collaborative Work Systems Series) Pfeiffer; (September 5, 2003).
- [5] Tyson Gill, Project Planning for Successful Software Development,  
<http://www.phptr.com/articles/article.asp?p=26397&rl=1>
- [6] David Coleman and Lewis Ward, Realizing the Potential of E-Collaboration, August 2001,  
[http://www.collaborate.com/mem/white\\_papers/new\\_CSAnalystwhitepaper-draft5.pdf](http://www.collaborate.com/mem/white_papers/new_CSAnalystwhitepaper-draft5.pdf)
- [7] Symon Lydiard, Defence e-Business: A Guide to Commercial Issues,  
[http://www.contracts.mod.uk/dc/public/ebusiness/collaborative\\_working.htm#2](http://www.contracts.mod.uk/dc/public/ebusiness/collaborative_working.htm#2)
- [8] <http://www.groove.net/home/index.cfm>
- [9] <http://www.speedev.com>
- [10] Ron Jeffries, Extreme Programming Adventures in C#
- [11] Secure Electronic Environment Workspace,  
<http://www.e-government.govt.nz/docs/workspace-3/chapter1.html>
- [12] Martin Burns, Introducing Project requirements,  
[http://www.evolt.org/article/Introducing\\_Project\\_Requirements/20/60393/](http://www.evolt.org/article/Introducing_Project_Requirements/20/60393/)
- [13] McConnell, Steve. *Software Project Survival Guide*. Redmond, Wash.: Microsoft Press, 1998.

[14] What is Issue Management,

[http://www.issuemanagement.org/documents/im\\_details.html#clarification%20of%20terms](http://www.issuemanagement.org/documents/im_details.html#clarification%20of%20terms)

[15] Project Management Methodology,

<http://www.dof.ca.gov/HTML/IT/PMM/OPTIMIZED/PM5.5%20Project%20Execution%20Change%20and%20Issue%20Mgmt.pdf>

[16] <http://www.baselinemag.com/article2/0,1540,666906,00.asp>

[17] <http://www.spidynamics.com/>

[18] Anita Sarma, A Survey of Collaborative Tools in Software Development,

[http://www.ics.uci.edu/~asarma/tech\\_report.pdf](http://www.ics.uci.edu/~asarma/tech_report.pdf)

## Appendix A. Use Cases

### A.1. UC1 – Login

#### A.1.1. Purpose

The purpose of UC1 is to provide login process of the system.

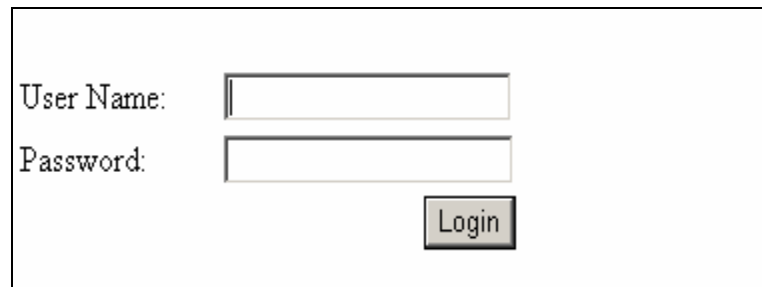
#### A.1.2. Description

This is a basic login page of the system. User is required to enter username and password to login to the system. After login process the user is redirected to the requested page based on the user role access.

#### A.1.3. User Interface

##### A.1.3.1. Page Mockup(s)

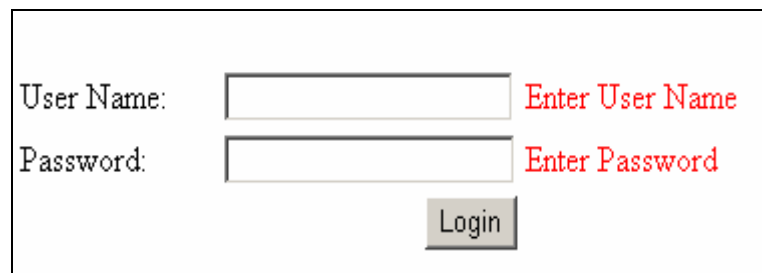
###### A.1.3.1.1. Login page



The mockup shows a login form with two input fields. The first field is labeled "User Name:" and the second is labeled "Password:". Below the password field is a button labeled "Login".

Figure A.6.1 – Login

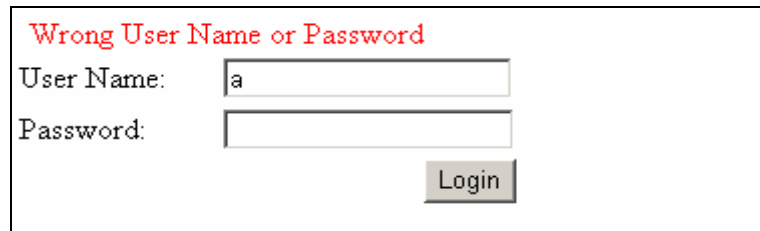
###### A.1.3.1.2. Error messages if username and/or password are empty.



The mockup shows the same login form as Figure A.6.1, but with error messages. The "User Name:" field has a red error message "Enter User Name" to its right. The "Password:" field has a red error message "Enter Password" to its right. The "Login" button is still present below the password field.

Figure A.6.2 - Login (Empty username and password error)

**A.1.3.1.3.** Error message if username and/or password are wrong.



Wrong User Name or Password

User Name:

Password:

Login

Figure A.6.3 – Login (Wrong username or password error)

### **A.1.3.2. Fields**

**A.1.3.2.1. User Name:** Field for user name.

**A.1.3.2.2. Password:** Field for password.

### **A.1.3.3. Buttons**

**A.1.3.3.1. Login:** Button to verify the username and password.

### **A.1.3.4. Links**

### **A.1.3.5. Other Active UI Elements**

**A.1.4. Actors:** Following actors are out of UC1 scope and they are going to be defined as INPUTs and OUTPUTs

**A.1.4.1. Admin:** Administrator of the system.

**A.1.4.2. User:** User of the system.

### **A.1.5. Preconditions**

**A.1.5.1.** Some users should be defined in the system.

### A.1.6. Use Case Diagram

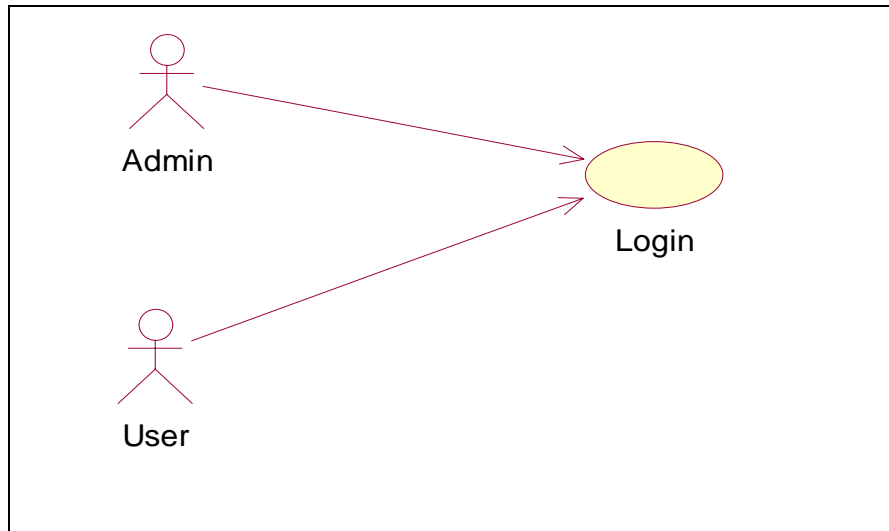


Figure A.6.4 – Use case diagram of login

### A.1.7. Basic Flow of Events

- A.1.7.1. User opens the Project Workspace.
- A.1.7.2. User enters user name and password.
- A.1.7.3. User clicks the Login button.

### A.1.8. Business Rules

- A.1.8.1. There are three types of system users.
  - A.1.8.1.1. **Admin:** Administrator of the system.
  - A.1.8.1.2. **User:** One who uses the Project Workspace system.
    - A.1.8.1.2.1. **Manager:** Has right to create workspace
    - A.1.8.1.2.2. **Participant**
- A.1.8.2. Password is kept in DB as hashed

### A.1.9. Error Conditions

- A.1.9.1. User Name is not entered: Error message is “Enter User Name”.
- A.1.9.2. Password is not entered: Error message is “Enter Password”.

**A.1.9.3.** User entered wrong user name or password: Error message is “Wrong User Name or Password”.

**A.1.10. Post Conditions**

**A.1.10.1.** Admin is redirected to administration page.

**A.1.10.2.** User is redirected to the page that he/she has selected.

**A.1.11. Related Use Cases**

**A.1.11.1.** UC2 – Workspace

**A.1.11.2.** Admin Page

**A.2. UC2 – Workspace**

**A.2.1. Purpose**

The purpose of UC2 is managing workspace properties: creating and modifying workspace.

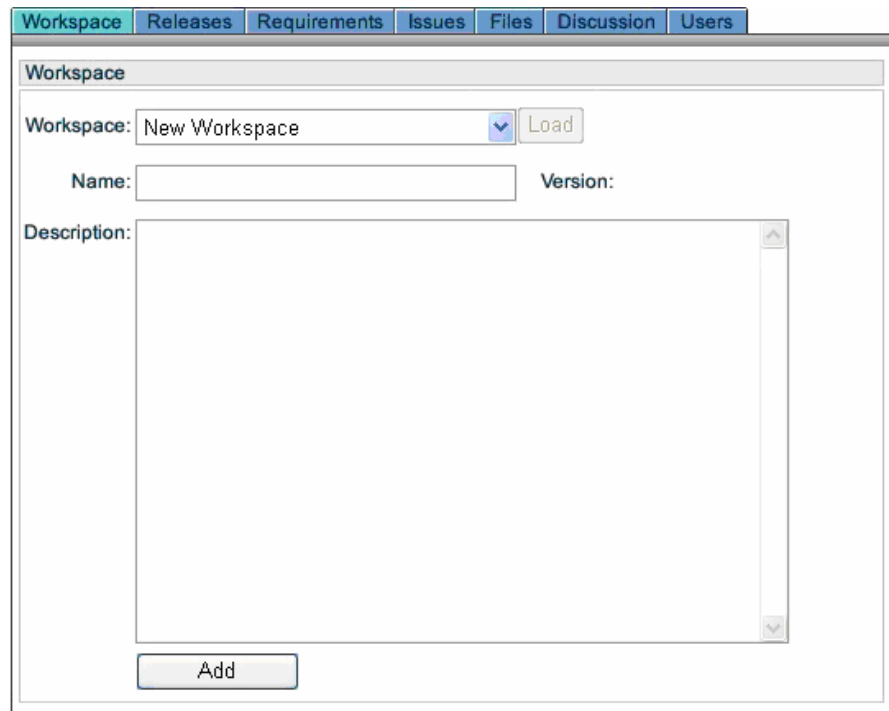
**A.2.2. Description**

A workspace is a place where people with common purposes and goals can share ideas and work together. A workspace begins with the person who creates it and then invites others to join. A user, who has access to create a workspace, defines a workspace and enters information about the workspace by using this module.

## A.2.3. User Interface

### A.2.3.1. Page Mockup(s)

#### A.2.3.1.1. Workspace page



The screenshot shows a web application interface with a navigation bar at the top containing tabs for 'Workspace', 'Releases', 'Requirements', 'Issues', 'Files', 'Discussion', and 'Users'. The 'Workspace' tab is active. Below the navigation bar is a form titled 'Workspace'. The form contains a dropdown menu labeled 'Workspace:' with 'New Workspace' selected, a 'Load' button, a 'Name:' text input field, a 'Version:' text input field, and a large 'Description:' text area with a vertical scrollbar. At the bottom of the form is an 'Add' button.

Figure A.6.5 - Workspace

### A.2.3.2. Fields

**A.2.3.2.1. Name:** Name of the workspace.

**A.2.3.2.2. Description:** Description of the workspace.

**A.2.3.2.3. Version:** Version number of the workspace.

### A.2.3.3. Buttons

**A.2.3.3.1. Add:** Button to add new workspace.

**A.2.3.3.2. Change:** Button to change the workspace.

**A.2.3.3.3. Load:** Button to load a workspace.

### A.2.3.4. Links

### A.2.3.5. Other Active UI Elements

**A.2.3.5.1. Workspace:** Drop down box for available workspaces.

**A.2.4. Actors:** Following actors are out of UC2 scope and they are going to be defined as INPUTs and OUTPUTs

**A.2.4.1. Manager:** System manager, can add new workspace.

**A.2.4.2. Workspace Manager:** has the change right on workspace.

**A.2.4.3. Workspace Participant:** can only load the workspace.

### A.2.5. Preconditions

**A.2.5.1.** User should login to the system.

### A.2.6. Use Case Diagram

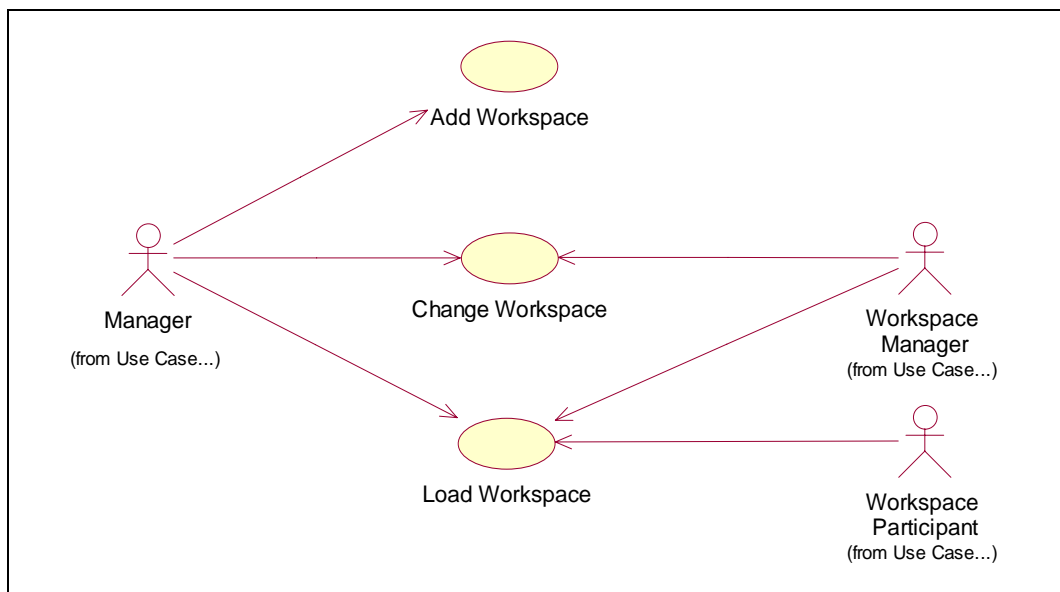


Figure A.6.6 – Use case diagram of workspace

### A.2.7. Basic Flow of Events

**A.2.7.1.** Add workspace:

**A.2.7.1.1.** User selects New Workspace from Workspace drop down box.



**A.2.7.1.2.** User enters workspace name and description.

**A.2.7.1.3.** User clicks Add button.

**A.2.7.2.** Load workspace:

**A.2.7.2.1.** User selects a workspace from available workspaces.

**A.2.7.2.2.** User clicks Load button.

**A.2.7.3.** Change workspace:

**A.2.7.3.1.** User changes workspace name and/or description.

**A.2.7.3.2.** User clicks Change button.

#### **A.2.8. Business Rules**

**A.2.8.1.** User should have right to add and/or change workspace.

**A.2.8.2.** Workspace name is unique.

**A.2.8.3.** Workspaces cannot be deleted.

**A.2.8.4.** If a workspace is changed, actually a new version of that workspace is created.

**A.2.8.5.** Changes between versions are determined by red colored labels.

#### **A.2.9. Error Conditions**

**A.2.9.1.** Workspace is not entered: Error message is “Enter Workspace Name”.

**A.2.9.2.** Workspace name exists: Error message is “Workspace name exists. Please enter a different workspace name”.

#### **A.2.10. Post Conditions**

**A.2.10.1.** User can access to sub modules (Releases, requirements, issues, files, discussion and user) by loading a workspace.

#### **A.2.11. Related Use Cases**

**A.2.11.1.** UC1 – Login

**A.2.11.2.** UC3 – Releases

**A.2.11.3.** UC3 – Requirements

**A.2.11.4.** UC4 – Issues

**A.2.11.5.** UC5 – Files

**A.2.11.6.** UC6 – Discussion

**A.2.11.7.** UC7 - Users

### **A.3. UC3 – Releases**

#### **A.3.1. Purpose**

The purpose of UC3 is managing workspace releases: creating and modifying releases.

#### **A.3.2. Description**

Release is a program package which provides the requested requirements. Releases of the project are defined by using this module.

### A.3.3. User Interface

#### A.3.3.1. Page Mockup(s)

##### A.3.3.1.1. Releases page

The screenshot shows a web application interface for managing releases. At the top, there is a navigation bar with tabs for 'Workspace', 'Releases', 'Requirements', 'Issues', 'Files', 'Discussion', and 'Users'. The 'Releases' tab is active. Below the navigation bar, there is a header area with the text 'Releases' and a list item 'Release 1...'. The main content area is divided into two sections. The top section displays user information: 'User: manager' and 'Date: 30.05.2005', along with 'New' and 'Change' buttons. The bottom section contains a form with the following fields: 'Status' (a dropdown menu set to 'Not Started'), 'Start Date' (an empty text input), 'Version' (a dropdown menu set to '2'), 'Complete Date' (an empty text input), 'Name' (an empty text input), and 'Description' (a large text area with a vertical scrollbar).

Figure A.6.7 - Releases

#### A.3.3.2. Fields

**A.3.3.2.1. Start Date:** Start date of the selected release.

**A.3.3.2.2. Complete Date:** Complete date of the selected release.

**A.3.3.2.3. Release Name:** Name of the release.

**A.3.3.2.4. Release Description:** Description of the release.

### **A.3.3.3. Buttons**

**A.3.3.3.1. Change Release:** Button to pass change mode.

**A.3.3.3.2. New Release:** Button to pass new release mode.

**A.3.3.3.3. Save:** Button to save changed or new release.

### **A.3.3.4. Links**

**A.3.3.4.1. Release List:** List of the available releases.

### **A.3.3.5. Other Active UI Elements**

**A.3.3.5.1. Status:** Status of the release(Not started, In Progress, Completed).

**A.3.3.5.2. Version:** Version number of the release.

**A.3.4. Actors:** Following actors are out of UC3 scope and they are going to be defined as INPUTs and OUTPUTs

**A.3.4.1. Workspace Manager:** has the change right on workspace.

**A.3.4.2. Workspace Participant:** can only load the workspace.

### **A.3.5. Preconditions**

**A.3.5.1.** User should login to the system.

**A.3.5.2.** User should load a workspace.

### A.3.6. Use Case Diagram

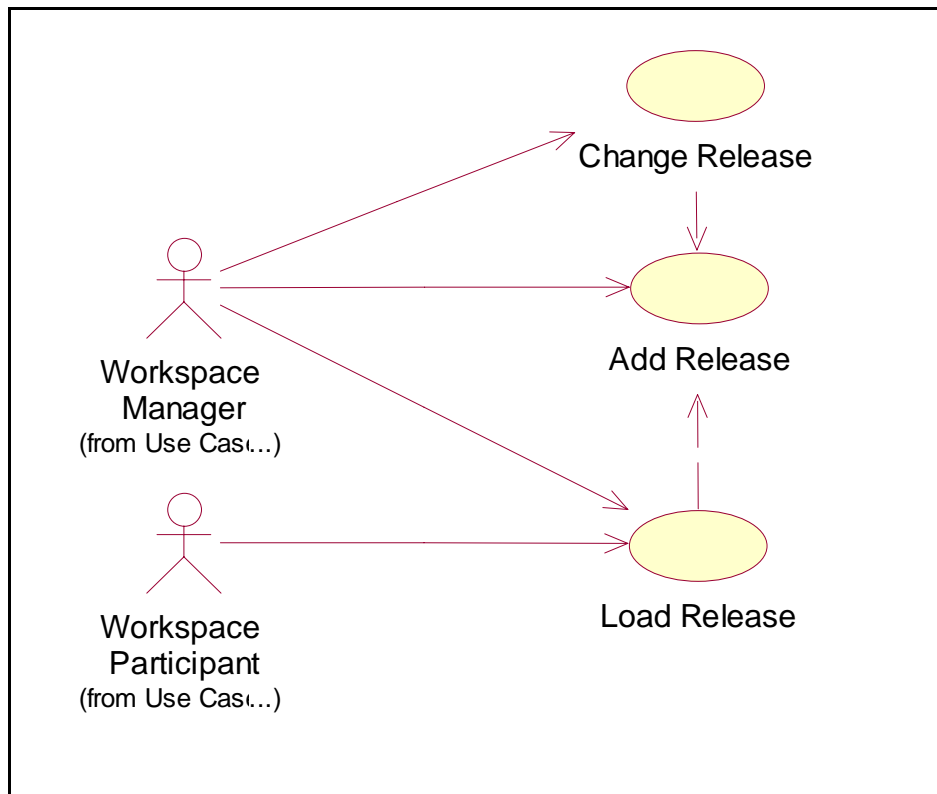


Figure A.6.8 – Use case diagram of releases

### A.3.7. Basic Flow of Events

#### A.3.7.1. Add release:

**A.3.7.1.1.** User clicks New Release button.

**A.3.7.1.2.** User enters release properties.

**A.3.7.1.3.** User clicks Save button.

#### A.3.7.2. Change release:

**A.3.7.2.1.** User clicks a release link from Release list.

**A.3.7.2.2.** User clicks Change button.

**A.3.7.2.3.** User changes release properties.

**A.3.7.2.4.** User clicks Save button.

### **A.3.8. Business Rules**

**A.3.8.1.** User should have right to add and/or change release.

**A.3.8.2.** If a release is changed, actually a new version of that release is created.

**A.3.8.3.** Changes between versions are determined by red colored labels.

### **A.3.9. Error Conditions**

#### **A.3.10. Post Conditions**

**A.3.10.1.** User can add requirements for the created release.

#### **A.3.11. Related Use Cases**

**A.3.11.1.** UC2 – Workspace

**A.3.11.2.** UC4 - Requirements

## **A.4. UC4 – Requirements**

### **A.4.1. Purpose**

The purpose of UC4 is managing release requirements: creating and modifying requirements.

### **A.4.2. Description**

Requirements are the things that the project must deliver. Requirements of a release are defined by using this module.

### A.4.3. User Interface

#### A.4.3.1. Page Mockup(s)

##### A.4.3.1.1. Requirements page

Workspace Releases Requirements Issues Files Discussion Users

Requirements

Releases : Release1.2

Requirement 1...

User: manager Date: 30.05.2005

New Change

Releases : Release1.2

Status : Not Started

Version : 2

Name :

Start Date :

Complete Date :

Description:

Figure A.6.9 - Requirements

#### A.4.3.2. Fields

**A.4.3.2.1. Start Date:** Start date of the selected requirement.

**A.4.3.2.2. Complete Date:** Complete date of the selected requirement.

**A.4.3.2.3. Requirement Name:** Name of the requirement.

**A.4.3.2.4. Requirement Description:** Description of the requirement.

### **A.4.3.3. Buttons**

**A.4.3.3.1. Change Requirement:** Button to pass change mode.

**A.4.3.3.2. New Requirement:** Button to pass new requirement mode.

**A.4.3.3.3. Save:** Button to save changed or new requirement.

### **A.4.3.4. Links**

**A.4.3.4.1. Requirement List:** List of the available requirements.

### **A.4.3.5. Other Active UI Elements**

**A.4.3.5.1. Release:** Drop down box for release list.

**A.4.3.5.2. Status:** Status of the requirement (Not started, In Progress, Completed).

**A.4.3.5.3. Version:** Version number of the requirement.

**A.4.4. Actors:** Following actors are out of UC4 scope and they are going to be defined as INPUTs and OUTPUTs

**A.4.4.1. Workspace Manager:** has the change right on workspace.

**A.4.4.2. Workspace Participant:** can only load the workspace.

### **A.4.5. Preconditions**

**A.4.5.1.** User should login to the system.

**A.4.5.2.** User should load a workspace.

**A.4.5.3.** There should be at least one release.



#### A.4.6. Use Case Diagram

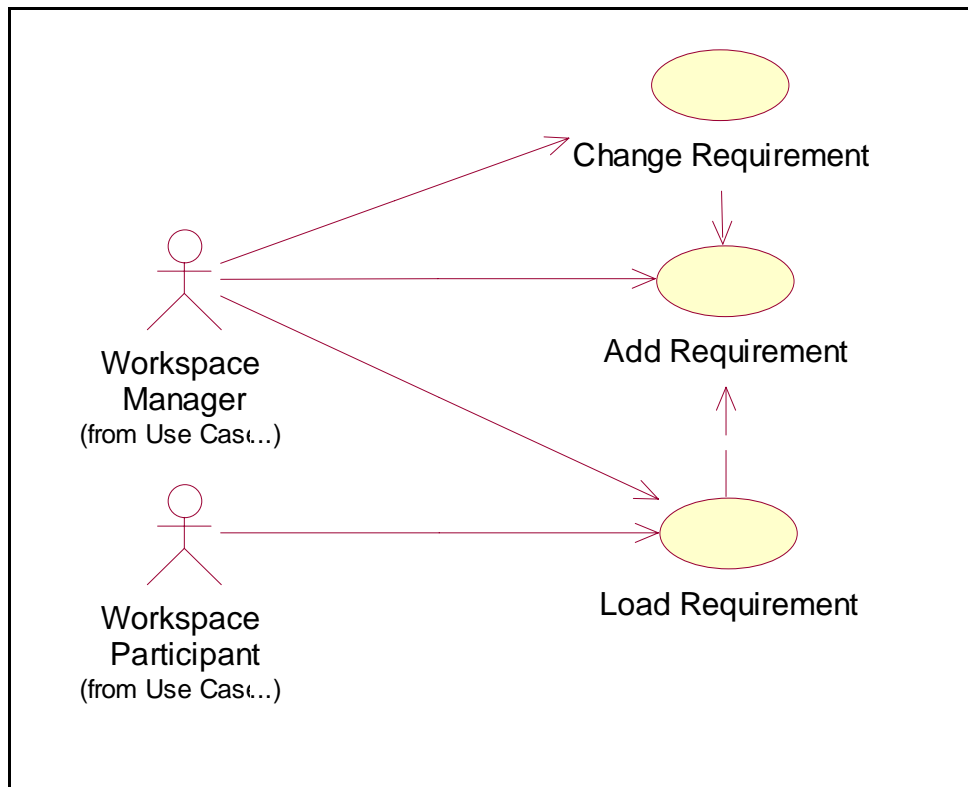


Figure A.6.10 – Use case diagram of requirements

#### A.4.7. Basic Flow of Events

##### A.4.7.1. Add requirement:

A.4.7.1.1. User clicks New Requirement button.

A.4.7.1.2. User enters requirement properties.

A.4.7.1.3. User clicks Save button.

##### A.4.7.2. Change requirement:

A.4.7.2.1. User clicks a requirement link from Requirement list.

A.4.7.2.2. User clicks Change button.

A.4.7.2.3. User changes requirement properties.

A.4.7.2.4. User clicks Save button.

#### **A.4.8. Business Rules**

**A.4.8.1.** User should have right to add and/or change requirement.

**A.4.8.2.** If a requirement is changed, actually a new version of that requirement is created.

**A.4.8.3.** Changes between versions are determined by red colored labels.

#### **A.4.9. Error Conditions**

#### **A.4.10. Post Conditions**

**A.4.10.1.** User can add issues for the created release.

#### **A.4.11. Related Use Cases**

**A.4.11.1.** UC2 – Workspace

**A.4.11.2.** UC3 – Releases

**A.4.11.3.** UC5 - Issues

### **A.5. UC5 – Issues**

#### **A.5.1. Purpose**

The purpose of UC5 is managing requirement issues: creating and modifying issues.

#### **A.5.2. Description**

Manage of requirement issues. Issues of a requirement are defined by using this module.

### A.5.3. User Interface

#### A.5.3.1. Page Mockup(s)

##### A.5.3.1.1. Issues page

The image shows two screenshots of a web application's 'Issues' page. The top screenshot shows a navigation menu with 'Issues' selected. Below the menu, there are two dropdown menus: 'Releases' (set to 'Release1.2') and 'Requirements' (set to 'Requirement1'). Below these is a text input field labeled 'Issue 1...'. The bottom screenshot shows a detailed form for an issue. At the top, it displays 'User: manager' and 'Date: 30.05.2005' with 'New' and 'Change' buttons. The form includes dropdowns for 'Releases' (Release1.2), 'Requirements' (Requirement1), 'Status' (Not Started), and 'Version' (2). It also has input fields for 'Start Date' and 'Complete Date', a 'Name' text field, and a large 'Description' text area.

Figure A.6.11 - Issues

#### A.5.3.2. Fields

**A.5.3.2.1. Start Date:** Start date of the selected issue.

**A.5.3.2.2. Complete Date:** Complete date of the selected issue.

**A.5.3.2.3. Issue Name:** Name of the issue.

**A.5.3.2.4. Issue Description:** Description of the issue.

#### A.5.3.3. Buttons

**A.5.3.3.1. Change Issue:** Button to pass change mode.

**A.5.3.3.2. New Issue:** Button to pass new issue mode.

**A.5.3.3.3. Save:** Button to save changed or new issue.

#### **A.5.3.4. Links**

**A.5.3.4.1. Issue List:** List of the available issues.

#### **A.5.3.5. Other Active UI Elements**

**A.5.3.5.1. Release:** Drop down box for release list.

**A.5.3.5.2. Requirement:** Drop down box for requirement list.

**A.5.3.5.3. Status:** Status of the issue (Not started, In Progress, Completed).

**A.5.3.5.4. Version:** Version number of the issue.

**A.5.4. Actors:** Following actors are out of UC5 scope and they are going to be defined as INPUTs and OUTPUTs

**A.5.4.1. Workspace Manager:** has the change right on workspace.

**A.5.4.2. Workspace Participant:** can only load the workspace.

#### **A.5.5. Preconditions**

**A.5.5.1.** User should login to the system.

**A.5.5.2.** User should load a workspace.

**A.5.5.3.** There should be at least one requirement.

### A.5.6. Use Case Diagram

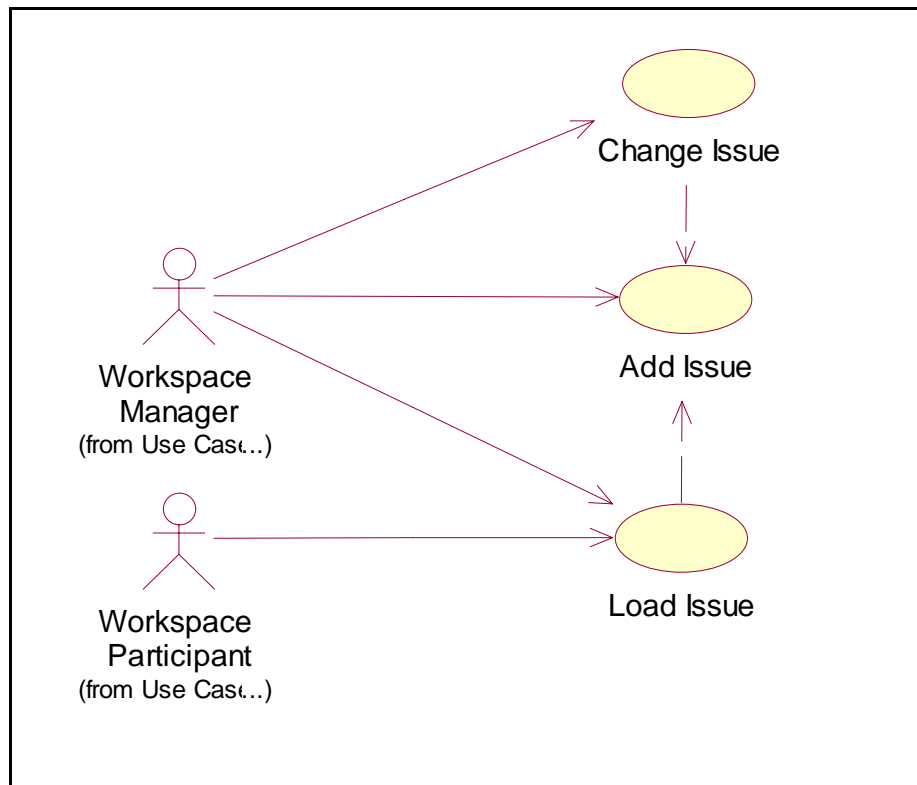


Figure A.6.12 – Use case diagram of issues

### A.5.7. Basic Flow of Events

#### A.5.7.1. Add issue:

**A.5.7.1.1.** User clicks New Issue button.

**A.5.7.1.2.** User enters issue properties.

**A.5.7.1.3.** User clicks Save button.

#### A.5.7.2. Change issue:

**A.5.7.2.1.** User clicks a issue link from Issue list.

**A.5.7.2.2.** User clicks Change button.

**A.5.7.2.3.** User changes issue properties.

**A.5.7.2.4.** User clicks Save button.

### **A.5.8. Business Rules**

**A.5.8.1.** User should have right to add and/or change requirement.

**A.5.8.2.** If an issue is changed, actually a new version of that issue is created.

**A.5.8.3.** Changes between versions are determined by red colored labels.

### **A.5.9. Error Conditions**

### **A.5.10. Post Conditions**

### **A.5.11. Related Use Cases**

**A.5.11.1.** UC2 – Workspace

**A.5.11.2.** UC4 - Requirements

## **A.6. UC6 – Files**

### **A.6.1. Purpose**

Purpose of UC6 is to share files to the workspace users, such as project documentation, release packages, images etc.

### **A.6.2. Description**

Sharing project files between workspace users is done by using this module.

## A.6.3. User Interface

### A.6.3.1. Page Mockup(s)

#### A.6.3.1.1. Files page

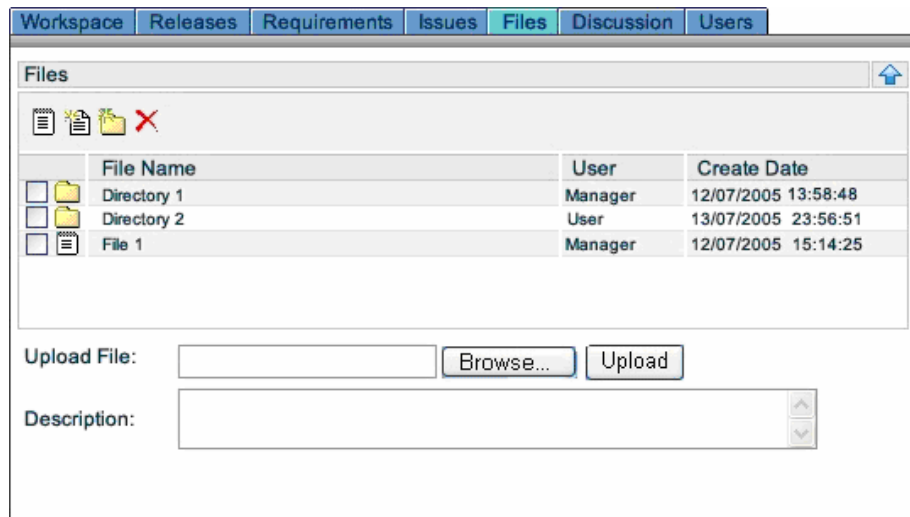


Figure A.6.13 - Files

### A.6.3.2. Fields

**A.6.3.2.1. Description:** Description of the file that will be uploaded.

### A.6.3.3. Buttons

**A.6.3.3.1. Browse:** button to get the local path of the file that will be uploaded.

**A.6.3.3.2. Upload:** button to upload the selected file.

### A.6.3.4. Links

**A.6.3.4.1. Parent Path link:** link to go to parent directory.

**A.6.3.4.2. File name link:** link to download file or go to the directory.

### A.6.3.5. Other Active UI Elements

**A.6.3.5.1. Select File:** checkbox to select files or directories.

**A.6.3.5.2. New Directory:** to create new directory.

**A.6.3.5.3. Delete:** deletes selected files and directories.

**A.6.4. Actors:** Following actors are out of UC6 scope and they are going to be defined as INPUTs and OUTPUTs

**A.6.4.1. Workspace Manager:** has the change right on workspace.

**A.6.4.2. Workspace Participant:** can only load the workspace.

**A.6.5. Preconditions**

**A.6.5.1.** User should login to the system.

**A.6.5.2.** User should load a workspace.

**A.6.6. Use Case Diagram**

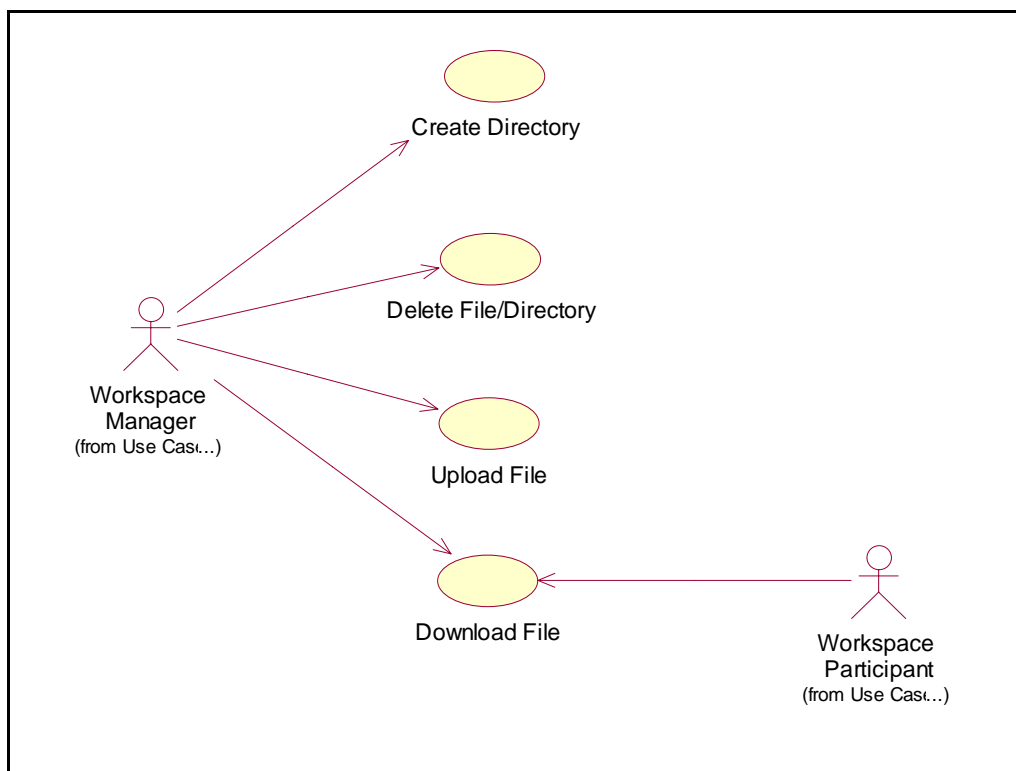


Figure A.6.14 – Use case diagram of files



### **A.6.7. Basic Flow of Events**

#### **A.6.7.1. Create Directory:**

**A.6.7.1.1.** User selects the path that will be the parent directory of the new directory.

**A.6.7.1.2.** User New Directory icon.

**A.6.7.1.3.** User enters the directory name.

#### **A.6.7.2. Delete file/directory:**

**A.6.7.2.1.** User selects the files and/or directories.

**A.6.7.2.2.** User clicks Delete icon.

#### **A.6.7.3. Download file:**

**A.6.7.3.1.** User clicks the file link that will be downloaded.

**A.6.7.3.2.** User clicks Save button.

**A.6.7.3.3.** User selects the download path.

#### **A.6.7.4. Upload file:**

**A.6.7.4.1.** User clicks Browse button.

**A.6.7.4.2.** User selects the file that will be uploaded.

**A.6.7.4.3.** User enters description for the new file.

**A.6.7.4.4.** User clicks upload button.

### **A.6.8. Business Rules**

**A.6.8.1.** User should have right to delete file/directory.

### **A.6.9. Error Conditions**

### **A.6.10. Post Conditions**

### **A.6.11. Related Use Cases**

**A.6.11.1.** UC2 - Workspace

## A.7. UC7 – Discussion

### A.7.1. Purpose

The purpose of UC7 is managing forum part for the modules.

### A.7.2. Description

Workspace users can discuss everything under the discussion sections: workspace, releases, requirements, and issues.

### A.7.3. User Interface

#### A.7.3.1. Page Mockup(s)

##### A.7.3.1.1. Discussion page

Workspace Releases Requirements Issues Files Discussion Users

Workspace

{ } Collapse / Expand All

- Workspace Topic 1
  - Workspace Topic 1.1
  - Workspace Topic 1.2
- Workspace Topic 2

Workspace

Release

Requirement

Issue

User Name : Admin Date : 13/07/2005 new edit reply

Workspace Topic 1.1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque congue dolor ac magna. Aenean fringilla tortor et elit. Integer facilisis, nibh sit amet fermentum placerat, enim nibh volutpat dolor, eu suscipit diam massa a tellus. Aenean pede lorem, consequat at, pulvinar pulvinar, malesuada sit amet, lectus. Praesent vitae velit. Aliquam egestas, turpis mollis facilisis ullamcorper, purus magna lacinia eros, eu vulputate sem lacus eu nisl. Sed non nulla. Curabitur tempor, mi vitae nonummy mollis, ante magna varius sem, at feugiat lorem sem pharetra nunc. Integer sit amet enim sed justo sagittis fringilla. Proin hendrerit. Aenean quis nibh. Aliquam sagittis, erat eu luctus dignissim, erat eros luctus libero, vitae semper

Save Cancel

Figure A.6.15 – Discussion

### **A.7.3.2. Fields**

**A.7.3.2.1. Topic:** Name of the Topic.

**A.7.3.2.2. Content:** Content of the topic.

### **A.7.3.3. Buttons**

**A.7.3.3.1. Edit:** Button to change discussion item.

**A.7.3.3.2. New:** Button to add new discussion.

**A.7.3.3.3. Save:** Button to save changed or new item.

### **A.7.3.4. Links**

**A.7.3.4.1. Item List:** List of the available discussions.

### **A.7.3.5. Other Active UI Elements**

**A.7.4. Actors:** Following actors are out of UC7 scope and they are going to be defined as INPUTs and OUTPUTs

**A.7.4.1. Workspace Manager:** has the change right on workspace.

**A.7.4.2. Workspace Participant:** can only load the workspace.

### **A.7.5. Preconditions**

**A.7.5.1.** User should login to the system.

**A.7.5.2.** User should load a workspace.

## A.7.6. Use Case Diagram

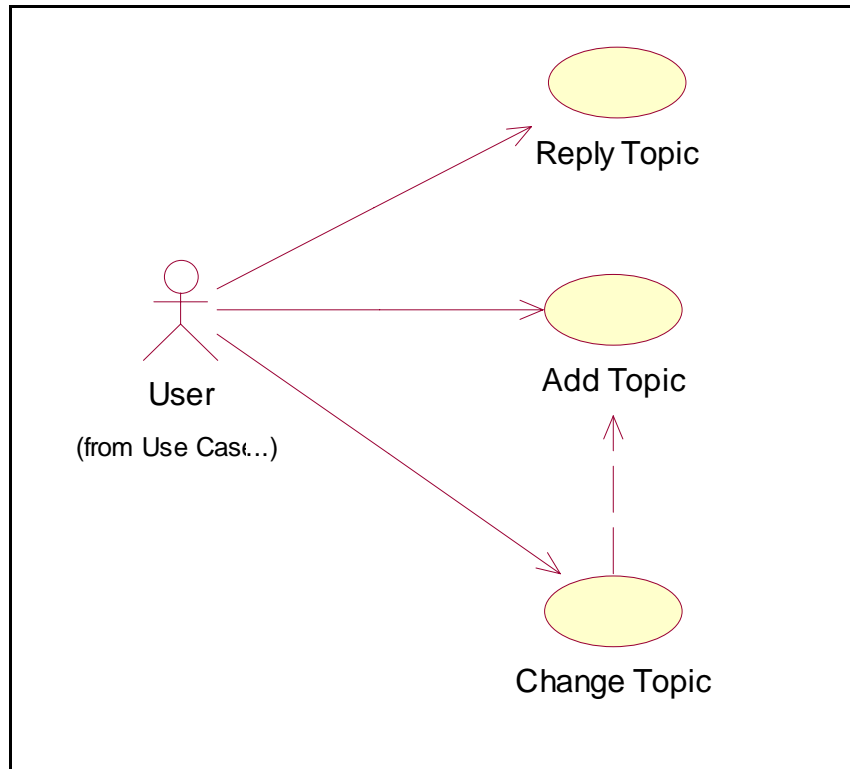


Figure A.6.16 – Use case diagram of discussion

## A.7.7. Basic Flow of Events

### A.7.7.1. Add discussion:

- A.7.7.1.1. User clicks New button.
- A.7.7.1.2. User enters topic and content.
- A.7.7.1.3. User clicks Save button.

### A.7.7.2. Reply discussion:

- A.7.7.2.1. User clicks Reply button.
- A.7.7.2.2. User enters topic and content.
- A.7.7.2.3. User clicks Save button.

### A.7.7.3. Change requirement

- A.7.7.3.1. User clicks a discussion link from discussion list.
- A.7.7.3.2. User clicks Edit button.

**A.7.7.3.3.** User changes discussion properties.

**A.7.7.3.4.** User clicks Save button.

**A.7.8. Business Rules**

**A.7.8.1.** User can only change his/her discussions.

**A.7.8.2.** Discussions are listed as tree structure.

**A.7.9. Error Conditions**

**A.7.10. Post Conditions**

**A.7.11. Related Use Cases**

**A.7.11.1.** UC2 - Workspace

**A.8. UC8 – Users**

**A.8.1. Purpose**

The purpose of UC8 is to manage workspace users.

**A.8.2. Description**

Workspace manager can add and remove workspace users by using this module.

## A.8.3. User Interface

### A.8.3.1. Page Mockup(s)

#### A.8.3.1.1. Users page

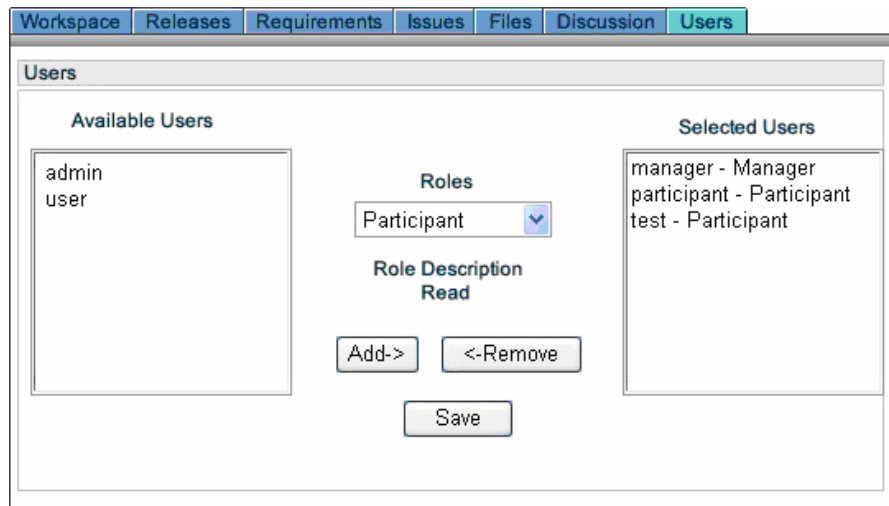


Figure A.6.17 - Users

### A.8.3.2. Fields

### A.8.3.3. Buttons

**A.8.3.3.1. Add:** Button to add users to the workspace.

**A.8.3.3.2. Remove:** Button to remove users from the workspace.

**A.8.3.3.3. Save:** Button to save the changes.

### A.8.3.4. Links

### A.8.3.5. Other Active UI Elements

**A.8.3.5.1. Available Users:** List box for available users.

**A.8.3.5.2. Selected Users:** List box for selected users for the workspace.

**A.8.3.5.3. Roles:** Roles to assign to the users.

**A.8.4. Actors:** Following actors are out of UC8 scope and they are going to be defined as INPUTs and OUTPUTs

**A.8.4.1. Workspace Manager:** has the change right on workspace.

#### **A.8.5. Preconditions**

**A.8.5.1.** User should login to the system.

**A.8.5.2.** User should load a workspace.

#### **A.8.6. Use Case Diagram**

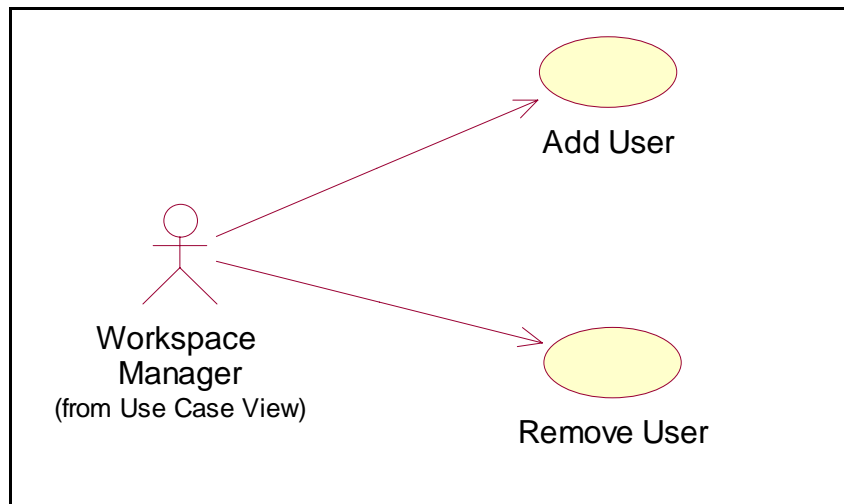


Figure A.6.18 – Use case diagram of users

#### **A.8.7. Basic Flow of Events**

**A.8.7.1.** Add user:

**A.8.7.1.1.** User selects user(s) from Available Users list box.

**A.8.7.1.2.** User clicks Add button.

**A.8.7.1.3.** User clicks Save button.

**A.8.7.2.** Remove user:

**A.8.7.2.1.** User selects user(s) from Selected Users list box.

**A.8.7.2.2.** User clicks Remove button.

**A.8.7.2.3.** User clicks Save button.

### **A.8.8. Business Rules**

**A.8.8.1.** User should be manager of the workspace.

**A.8.8.2.** There four types of workspace user roles:

**A.8.8.2.1. Participant:** Has only read right.

**A.8.8.2.2. User:** Has read and add rights.

**A.8.8.2.3. Leader:** Has read, add and edit rights.

**A.8.8.2.4. Manager:** Has read, add edit and delete rights.

### **A.8.9. Error Conditions**

#### **A.8.10. Post Conditions**

**A.8.10.1.** Added user can access to the workspace.

**A.8.10.2.** Removed user cannot access to the workspace.

#### **A.8.11. Related Use Cases**

**A.8.11.1.** UC2 - Workspace

### **A.9. UC9 – Admin**

#### **A.9.1. Purpose**

The purpose of UC9 is to add and change system users.

#### **A.9.2. Description**

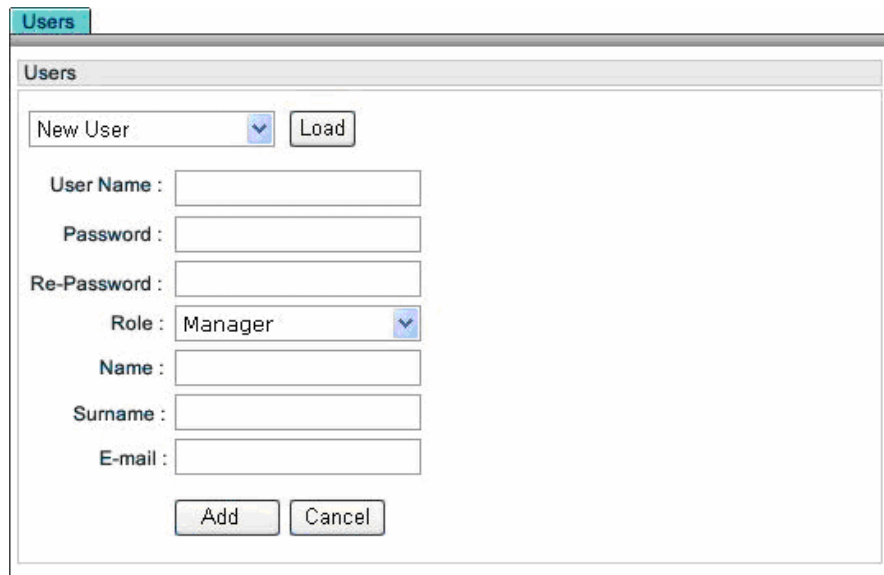
System administrator can add and remove system users by using this module.



### A.9.3. User Interface

#### A.9.3.1. Page Mockup(s)

##### A.9.3.1.1. Admin page



The screenshot shows a web application window titled 'Users'. Inside the window, there is a sub-header 'Users' and a form for adding a new user. The form includes a 'New User' dropdown menu and a 'Load' button. Below these are text input fields for 'User Name', 'Password', and 'Re-Password'. There is also a 'Role' dropdown menu currently set to 'Manager'. Further down are text input fields for 'Name', 'Surname', and 'E-mail'. At the bottom of the form are 'Add' and 'Cancel' buttons.

Figure A.6.19 - Administration

#### A.9.3.2. Fields

**A.9.3.2.1. User Name:** Text box for user's system user name.

**A.9.3.2.2. Password:** Text box for password.

**A.9.3.2.3. Confirm Password:** Text box for confirm password.

**A.9.3.2.4. Name:** Text box for name of user.

**A.9.3.2.5. Surname:** Text box for surname of user

**A.9.3.2.6. Email:** Text box for email of the user.

#### A.9.3.3. Buttons

**A.9.3.3.1. Load:** Button to load user info.

**A.9.3.3.2. Add:** Button to add new user.

**A.9.3.3.3. Change:** Button to change user information.

**A.9.3.3.4. Cancel:** Button to cancel add or change process.

#### A.9.3.4. Links

#### A.9.3.5. Other Active UI Elements

**A.9.3.5.1. User List:** Dropdown list box for available system users.

**A.9.3.5.2. Roles:** Roles to assign to the users.

**A.9.4. Actors:** Following actors are out of UC9 scope and they are going to be defined as INPUTs and OUTPUTs

**A.9.4.1. Admin:** has access right to the administrator page.

#### A.9.5. Preconditions

**A.9.5.1.** Admin should login to the system.

#### A.9.6. Use Case Diagram

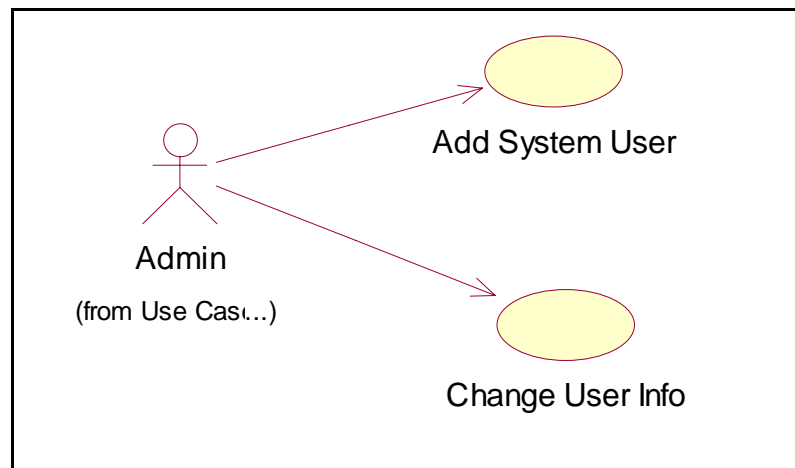


Figure A.6.20 – Use case diagram of administration

#### A.9.7. Basic Flow of Events

**A.9.7.1.** Add user:

**A.9.7.1.1.** User selects “New User” from Available Users list box.

**A.9.7.1.2.** User clicks Load button.

**A.9.7.1.3.** User enters new user information.

**A.9.7.1.4.** User clicks Add button.

**A.9.7.2.** Change user:

**A.9.7.2.1.** User selects user(s) from Selected Users list box.

**A.9.7.2.2.** User clicks Load button.

**A.9.7.2.3.** User changes the user information.

**A.9.7.2.4.** User clicks Change button.

### **A.9.8. Business Rules**

**A.9.8.1.** User should be administrator of the system.

**A.9.8.2.** There three types of system user roles:

**A.9.8.2.1. Participant:** Has only read right.

**A.9.8.2.2. Manager:** Has read and add workspace rights.

**A.9.8.2.3. Admin:** Has add and change system user rights.

### **A.9.9. Error Conditions**

### **A.9.10. Post Conditions**

**A.9.10.1.** Added workspace user can be invited to a workspace.

**A.9.10.2.** Added manager can create a new workspace.

### **A.9.11. Related Use Cases**

**A.9.11.1.** UC1 – Login

## **Appendix B. CD Content**