Project Manager 2007

A Chapter Style Thesis

Submitted to

the Department of Computer Science

and the Faculty of the

University of Wisconsin-La Crosse

La Crosse, Wisconsin

by

Metin Tuzmen

in Partial Fulfillment of the

Requirements for the Degree of

Master of Software Engineering

December, 2007

Project Manager 2007

By Metin Tuzmen

We recommend acceptance of this manuscript in partial fulfillment of this candidate's requirements for the degree of Master of Software Engineering in Computer Science. The candidate has completed the oral examination requirement of the capstone project for the degree.

Dr. Kasi Periyasamy
Examination Committee Chairperson

Date

Dr. Thomas Gendreau
Examination Committee Member

Date

Dr. Mao Zheng
Examination Committee Member

Date

ABSTRACT

TUZMEN, METIN, “Project Manager 2007”, Master of Software Engineering, December, 2007, Advisors: Dr. Kasi Periyasamy, Dr. Thomas Gendreau

Every software employs certain level of project management system. Irrespective of the size of the team that builds the software, good project management skills and tools will always make developing the software easier, maintainable and within the proposed budget. However, when project management fails to perform the right tasks, the software development process results increasing costs, and serious problems, and fails in its promised value. This could even lead to the project being abandoned and resources being demoralized. Many tools have been developed to streamline project management. Some of them help project managers control project schedule, others do budget controls, and yet others allow addition of other components such as requirements management and/or code management. Project Management 2007 is a project which realizes the importance of project management, not just as a place to control a certain project, but as a place where software engineering activities can be managed and learned for better performance and refining these activities to fit the real world. Project Manager 2007 manages projects at an organizational level, where each project is defined in qualitative details based on the organization’s structure and software engineering techniques. Project managers as well as other mid-high level managers can learn about their organization, software development capacities and can give direction to new initiatives to improve quality of the software. Project Manager 2007 addresses project management, organization management, resource management, risk management, skill management, document management, metrics management and requirements management.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my project advisors Dr. Kasi Periyasamy and Dr. Thomas Gendreau for their valuable guidance. I like to thank the project sponsor, Firstlogic who provided the support for this project. I also would like to thank my supervisor Dan Bills and co-workers Scott Roellig and Lakshmi Vaimpalli for their input into various aspects of this project. Finally, I wish to thank my wife and my soon to be born kid for their patience and encouragement during the tenure of this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

.NET Framework
The Microsoft .NET Framework is a software component included in the Microsoft Windows operating system. It provides a large body of pre-coded solutions to common software development and manages the execution of programs written specifically for the framework. The .NET Framework is intended to be used by most new applications created for the Windows platform. [1]

AJAX
AJAX (Asynchronous JavaScript and XML), or Ajax, is a web development technique used for creating interactive web applications. [1]

ASP.NET
ASP.NET is a web application framework marketed by Microsoft that programmers can use to build dynamic web sites, web applications and XML web services. [1]

C#
C# is an object-oriented programming language developed by Microsoft as part of the .NET initiative and later approved as a standard by ECMA and ISO [1]

ISO 9126-1
ISO 9126 is an international standard for the evaluation of software quality. [1]

Java Script
JavaScript is a scripting language most often used for client-side web development. [1]

MS installer
Microsoft installer. This is the application that installs Project Manager 2007 to the server.

Process Management
Process Management is the application of knowledge, skills, tools, techniques and systems to define, visualize, measure, control, report and improve processes with the goal to meet customer requirements profitably. [1]

Project Management
Project management is the discipline of organizing and managing resources (e.g. people) in such a way that the project is completed within defined scope, quality, time and cost constraints. [1]

Prudentia
Codename given to Project Manager 2007 during development.

SQL
Development language developed for retrieval of data from relational databases.

SQL Server
Microsoft SQL Server is a relational database management system (RDBMS) produced by Microsoft. [1]

SQL Scripts
Text files containing SQL language for creating tables in SQL Server databases.

Thick client
A fat client is a computer (client) in client-server architecture networks which typically provides rich functionality independently of the central server [1]

Thin client
A thin client (sometimes also called a lean client) is a client computer or client software in client-server architecture networks which depends primarily on the central server for processing activities, and mainly focuses on conveying input and output between the user and the remote server. [1]

Visual Studio
Microsoft Visual Studio is Microsoft's flagship software development product for software developers. It centers on an integrated development environment which lets programmers create software applications (console and GUI applications including Windows Forms applications), web sites, web applications, and web services that run on any platforms supported by Microsoft's .NET Framework (for all versions after Visual Studio 6). [1]

1. Introduction to Software Project Management

Project management is a continuous process of managing resources and their activities in order to successfully complete a product and making it better. Every project goes through well-defined stages such as preparation, design, production, validation and completion. Each stage is completed by a collection of pre-defined activities and consumes the resources that are set aside by the project management. The effective utilization of the resources forms the core of project management. The primary goal of project management is to take quantitative measurements and analysis of the allocation, utilization and consumption of the resources based on their usage in previous projects.

The following case study illustrates the need for better project management: Imagine a construction company. A customer wants to build a house. In this scenario, the house becomes the project, the construction company's lead engineer is the project manager, and the customer stays in the role of the customer. The first action taken by the project manager would be to go through the preparation phase where the company staff sit down with the customer, and discuss about how their dream house will look like. Once the preparations are done, the project manager then creates the design of the house and starts gathering resources and activities for the project. These resources can be human or non human but they are all managed in the same way and perform the activities to get the house built. After the design is completed and activities are planned, the production stage starts. At this stage, the house is built with the instructions given by the project manager. During this stage, the house is validated continuously, to make sure that the design is being followed and the house is safe. At the end, when the house is completed, the keys are handed over to the customer who is very happy with the sheer beauty of a stunning house that was built within the budget. The most interesting part of the case study is when the project manager looks at all details of the project, analyzes what worked well and what not worked well for this project, and shares this information with other project managers so they document this experience. When the next time they build a house, this collective knowledge allows them to build a better house in less time, and possibly with a lower budget. It is this interesting part of the project manager that is addressed in the rest of this manuscript.

Software project management is one such type of project management. The only difference is that it applies only to software projects; accordingly, the activities and resources are different. Billions of dollars are being spent on software projects each year, and so even if we estimate a conservative amount of 5% of these projects fail, it would still amount to a very large dollar figure. At the end, only a handful of projects are considered to be good enough for customers to purchase and use without having to go through great pains.

Every software project has many controlling variables, and many different factors that cause this project to fail. Controlling each one of these sources of chaos is a feat by itself and every project suffers from this chaotic environment, whether you are building software or building a house. Project management's main goal should therefore be to balance and encounter this chaotic environment to bring order and stability, so that the project can progress on a smooth environment. Supplemental goals of project management should be to identify and learn from these chaotic environmental forces. The result of such effective project management would be that future projects will have less chances of running into problems. Standardized processes can be developed from this learning process and can be improved upon.

Project management is a complex concept, not necessarily because it is hard but because there are so many different variables that we need to be aware of. Adding on to this complexity is the fact that most companies while wanting to improve their processes only act when there is an emergency. They do not establish proper management from the beginning. Temporary fixes only solve the current problem but future problems of the same type still need emergency fixes. Most software companies run over budget, not on time to release software products and most of the time, throw quality to second priority due to market pressure. While this concept holds true for now, in the future, quality of software more than its price will be the main factor in selling a software product. The only way to achieve improvements in quality is to continuously re-evaluate development processes, use quantitative measures, get feedback from people who have gone through such an evaluative process before, and document all the analysis for future use.

So how can we achieve quality software? Well the answer to this question is continuous process improvement. As the processes are applied to projects, it should be the project manager’s job to follow up on them and find out from the departments who are working on these projects how the processes are being applied and what kind of results are being achieved. As this cycle churns, more and more people will find out what works and what doesn’t work. Each project will attain a better quality and processes will be more widely accepted by the workers as they will see that it makes their life easier and their job more fulfilling.

2. Development of Project Manager 2007

2.1. History of Project Manager 2007

Project Manager 2007 was developed from scratch for the most part. Some classes used in the application were harvested from previous projects by the same developer. Initial requirements gathering were started in 2004 and were completed in 2005. Design was completed in 2006 and development started in the same year. Beta version of Project Manager 2007 is available on November 27, 2007. Project Manager 2007 was developed using a combination of waterfall and spiral models. Waterfall structure and processes were used for requirements, testing and release but spiral development was used for design and implementation stages.

It took quite a while to write the requirements document for Project Manager 2007 due to the fact that the requirements kept changing. Another problem was that the requirements document looked more like a design document in the sense that it had more in-depth information. This triggered several meetings with the sponsors before the requirements were solidified. Once the requirements were reviewed and accepted by the sponsors and project advisors, the design of the application started. The design work composed of creating the use case diagram, design document, class diagram, navigation structure diagram and database design diagram. Many prototypes of the screens were developed and scrapped at this stage.

Because of the extensive work done during design stage, the implementation work was fairly straight forward. Only minor changes were made to the application coming from scope creep. Many new ideas had to be discarded at this stage because it made the application longer to develop and less stable as design wasn't available for them. However, some of them were implemented within the application. An example of this was the error handling mechanism. At one point, all classes were using individual error routines which caused long stretches of 'try' and 'catch' statements. Later on, it was decided to break the 'catch' statement into three components: one for application specific exceptions, one for data integrity errors, one for system exceptions. Sometimes a fourth one was needed for any other Exceptions thrown from the user interface backend code operations.

Release of Project Manager 2007 will be in two stages, a beta deployment and the release. Beta version was available on Nov 27 2007 and release will follow shortly after that based on beta results. An initial installer already exists to install the necessary files based on an MS installer. Along with this, SQL files will be created to generate the necessary database structure. These two installers will be separate for the time being but will be merged at a future release.

Project Manager 2007 was chosen as a thin client application vs. a thick client, due to several factors. Table 1 displays the reasons why a thin client approach was chosen.

	Factor	**Reason**
1	Minimize resource consumption on IT	Thin client applications have minimal footprint on the client systems, thus minimizing the workload on these systems so that these systems are available for other applications to utilize their processing power and CPU cycles.
2	Install once, run everywhere	Thin client applications can be made run on a web browser, rendering HTML to the user which enhances its availability and usability.
3	Scalability	In thin client applications, the application itself can be placed in either a single server or a network of servers, thus multiplying the processing power and scalability of the application.
4	Security	In thin client applications, there is only one machine that creates a connection to the database where the data is stored. This makes the application more secure, and easier to integrate.
5	Maintainability	Thin client applications are easier to update and maintain. If there is an update, the application can handle the update in a shorter time. On the other hand, thick client applications require more time for updates thus skyrocketing the cost for maintenance.

Table 1. Factors For Choosing Thin Client Model

Due to these reasons, Project Manager 2007 was chosen to be deployed as a web based thin client application.

2.2. Project Management Structure

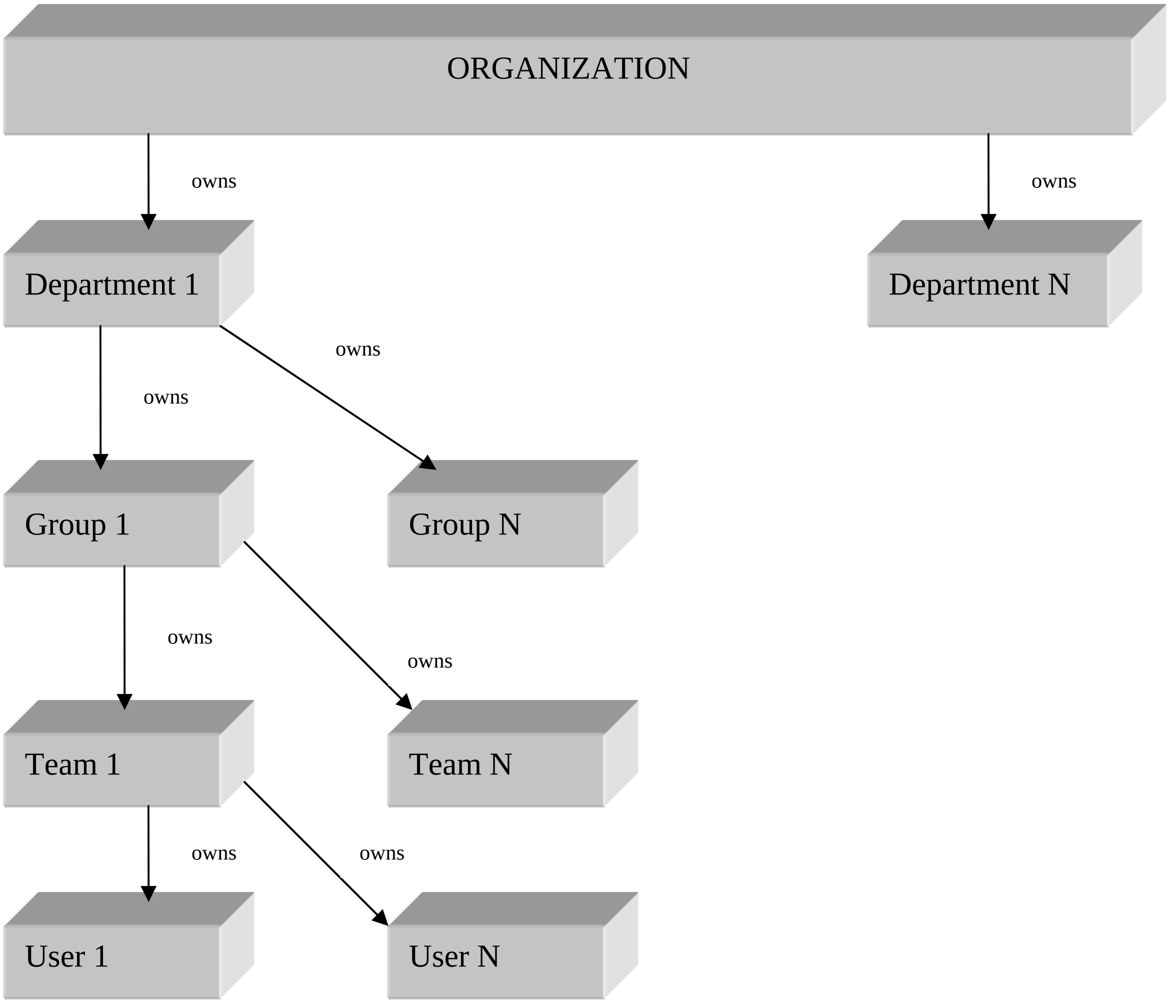


Figure 1. Organizational Structuring In Project Manager 2007

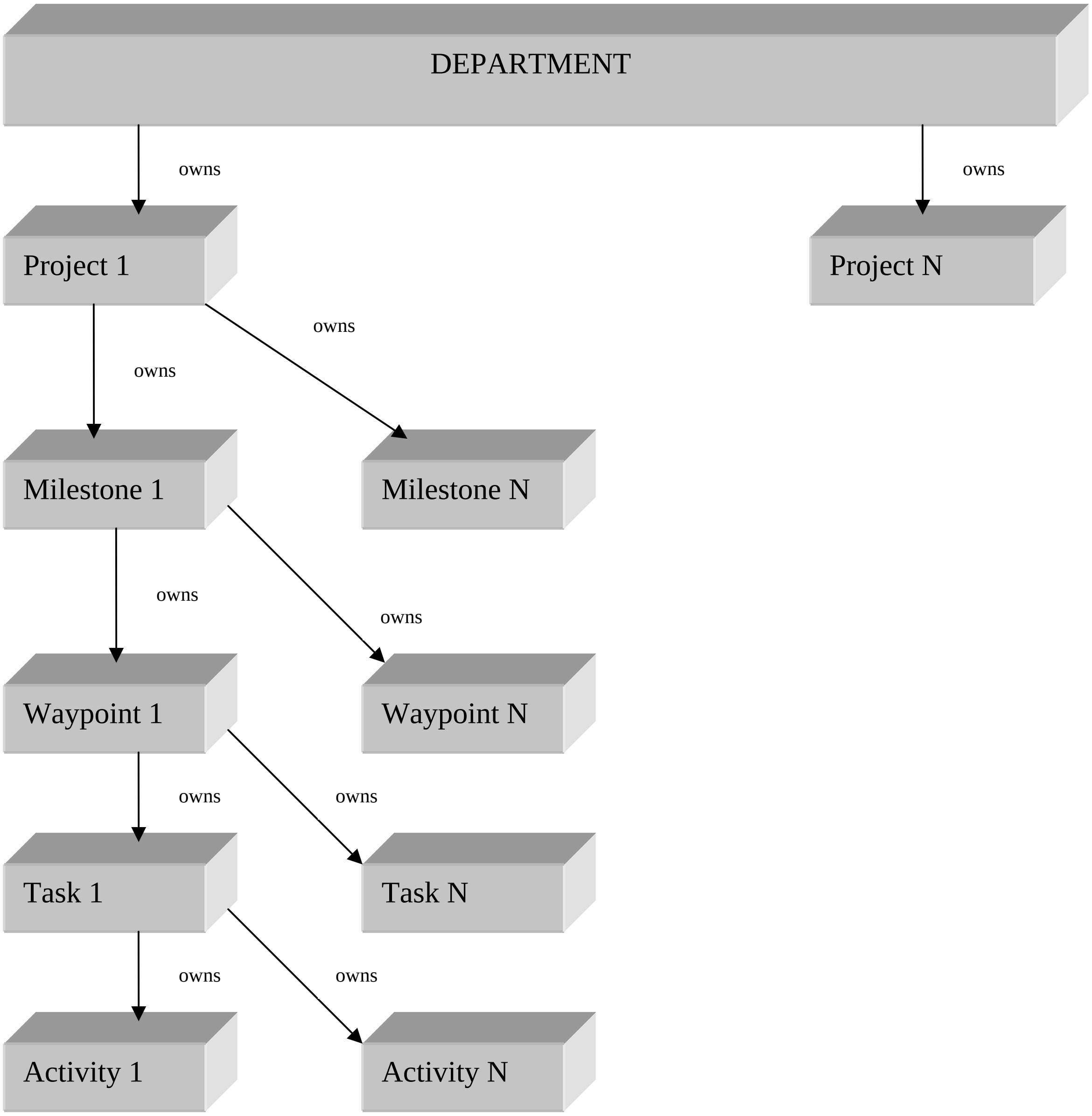


Figure 2. Project Structuring In Project Manager 2007

Figures 1 and 2 show two different models in an organization, Figure 1 representing the departmental view and Figure 2 representing the project view. In these models, the organization is the universe. An organization may have many departments, a department may have many groups, a group may have many teams and finally a team may have many resources. In essence, the hierarchy asserts that a resource ultimately belongs to the organization. One could explore the relationships between a project and its entities. For example, a project is tied to the department, the project contains milestones, each milestone contains waypoints, waypoints contain tasks and tasks contain activities. Activities are assigned to resources belonging to the department where the project is assigned.

This type of structure for an organization allows a project be divided into atomic components so that each component can be analyzed. Moreover, users can take on multiple roles to reflect real world scenarios. This approach is also useful for future expansion, where dependencies can be associated between different components.

2.2.1. Users

Project Manager 2007 supports three categories of users: administrators, managers and users. However, support for up to one hundred user types is built into Project Manager 2007 but are not currently utilized.

Administrators have the highest level of privileges. Their main task is to maintain the application. In most cases, administrators are not end users. An administrator only sees the world of project management from a database maintenance point of view. Therefore, while they can do anything and everything, they cannot see project status. They can, however, work within the application to fix any errors in the application reported by users.

Managers are high end users in the sense that they mostly handle status information rather than dealing with low level entities such as code, diagrams and other intrinsic details. However, manager type users are hierarchically ordered within themselves by the application. This hierarchy forces some activities to be performed only by managers at a certain level. Here is an example:

User Carlos Queiroz is a group manager in the department Manchester United which is managed by Alex Ferguson. This department is working on two different projects, English Premier League and Champions League. Carlos Queiroz manages the work for the Champions League. While Alex Ferguson can see the statistical information and make changes to both the groups, Carlos Queiroz can only make changes to Champions League project. He can however view the information about the English Premier League but cannot make changes to it.

It is therefore clear that managers can only change what is directly under their control. They can view the information under other managers' control but have no rights to make modifications. This way, any given project is controlled by the people directly involved with it based on the parts under their sphere of influence. Distinction between projects is respected and different views into the projects based on organizational roles become possible.

The lowest level is the user type. Users only have rights to do certain actions such as manipulating tasks and activities. Users are the core workers in a project. Notice that one user may work on several projects during a given time period depending on the work load.

2.2.2. Departments

Most users are indirectly assigned to a department. The only users directly assigned to a department are the department managers. This is because every department needs a person in charge and even if the position is vacant, someone is always playing a dual role in real world. Departmental ownership changes can be done at anytime by an administrator, provided that the current manager of the department does not have any pending activities to be completed within the departments' projects. If so these activities first needs to be assigned a new owner and then department manager may change. One or more projects are assigned to departments.

2.2.3. Groups

Groups form a department. A group is a smaller subset of a department and there can be many groups under a department. Groups can be involved with as many projects as the department is involved. Groups are can either be created by the manager of the department or can be created by the administrator. Groups can be edited or deleted, provided that they do not have any pending task in any project yet to be completed. Most users are indirectly assigned to a group. The only users directly assigned to a group are the group managers.

2.2.4. Teams

A team consists of a smaller subset of a group and there can be many teams under a group. Teams, like groups, can be involved in as many projects as their department is assigned to. Teams can either be created by a high level manager or can be created by the administrator.

Teams are special in the sense that, users are assigned directly to it. Team managers are assigned by either a high level manager or the administrator. The users in the team are then directly assigned by any of the managers (or administrator) thus forming the team.

2.2.5. Projects

Projects in Project Manager 2007 are created by department managers or the administrator. Each department manager is responsible for setting up the project that their departments are assigned to. As mentioned earlier, in Project Manager 2007, responsibilities lie within the people who have direct control over the object.

A project starts when its first lowest level atomic part (activity) starts. A project finishes when its last lowest level atomic part (activity) finishes. A project is completed manually, after the project or department manager reviews and verifies that all work assigned to the project have been completed.

A project type describes the unique characteristics of the project. In some sense, this is similar to a class in object-orientation and the projects are somewhat similar to the objects created under this project type. This allows future projects to be compared against completed projects. This way, department managers, can easily find completed projects of the same type (and composition) in order to get a glimpse of what kind of challenges they will be expecting. An administrator can create project types that fit the organization. The installation of Project Manager 2007 only comes with one type which is the default type. This is because it is assumed that organizations take time to define the types of projects they work on.

2.2.6. Milestones

Milestones in Project Manager 2007 can be created by any manager of a department who owns that project. This means that group managers or team managers as well as department managers can create milestones. The reason for this option is that parts of the project can be built in tandem with other parts of the same project. For example, one group may be working on the user interface while another group works on the back end code.

Milestones are the backbones of a project. A project can have a number of milestones. Milestone types are defined by the organization as the installer only comes with the default type. A milestone starts when the first activity that belongs to the milestone starts, and ends when the last activity that belongs to the milestone completes. Milestones are automatically manipulated by the software and do not require manual intervention for their status updates.

2.2.7. Waypoints

Waypoints in Project Manager 2007 can be created by any manager assigned to the project. Waypoint defines milestones and is meant to be concrete steps in the lifecycle of the milestone to be completed. A milestone can have an unlimited number of waypoints assigned to it.

As with projects and milestones, waypoints do not require human interaction for status updates. A waypoint starts when an activity related to it starts, and finishes when the last activity related to it completes.

The installation of Project Manager 2007 comes with only the default type of waypoints. Therefore an administrator needs to add the types of other waypoints related to an organization.

2.2.8. Tasks

Tasks in Project Manager 2007 are usually created by users. Managers can also create tasks if needed. Tasks are part of waypoints. A waypoint can have an unlimited amount of tasks assigned to it. Tasks do not start and end with human interaction but rather rely on its activities to start and finish for completion.

Installation of Project Manager 2007 only adds the default task type and therefore an administrator needs to create the types of tasks used in the organization.

2.2.9. Activities

Activities are the atomic component of a project. Activities constitute the tasks and as explained above, tasks constitute waypoints, waypoints constitute milestones and milestones constitute projects. One important aspect of activities is that activities are always assigned a resource. Activities, like tasks, are mostly created by users but managers can create them as well. The status of an activity is always changed by the user assigned to the activity (or user's manager or the department manager). The same strategy is used to control the status of other entities as well.

Like other entities, during initial installation, activities only come with the default type and therefore an administrator needs to setup the types of other activities based on the organization. Activities also contain complexity and priority fields. The priority field allows managers to determine which activities have higher priority over other activities. The complexity field allows managers to see how complex an activity is. These two parameters, working together, bring better schedule and resource control over a project. Notice that these fields also propagate upwards along the chain of projects and are calculated from 1 to 3 (3 being the highest level and riskiest and 1 being easiest and less risky). The following example illustrates this scenario.

Project 'A' contains milestone 'a1' which in turn contains waypoint 'a11'. The waypoint 'a11' contains tasks 'b' and 'c'. Task 'b' contains activities 'b1' and 'b2', while task 'c' contains activities 'c1' and 'c2'. The priority of activity 'b1' is 'High' and its complexity is also 'High'. The priority of activity 'b2' is 'High' and its complexity is 'Medium'. The priority of activity 'c1' is 'Low' and its priority is 'Low', while the priority of activity 'c2' is 'Medium' and its complexity is 'high'. Weighed value is the average calculated value.

When a manager looks at the task's statuses, he/she would see that

Task B Priority = 3 and complexity = 2.5 and weighed value = 2.75

Task C Priority = 1.5 and complexity = 2 and weighed value = 1.75

Waypoint a11 Priority = 2.25 and complexity = 2.25 and weighed value = 2.25

Milestone a1 Priority = 2.25 and complexity = 2.25 and weighed value = 2.25

Project a Priority = 2.25 and complexity = 2.25 and weighed value = 2.25

If the deadline of the project is fast approaching, the manager must take a decision on whether or not to finish task ‘b’ or ‘c’, or add resources if both needs to be finished.

These components also help define quantitative measures of a project (and all its components) so that when similar projects with somewhat similar measures are being created, a manager can make use of past experiences.

2.3. Process Management

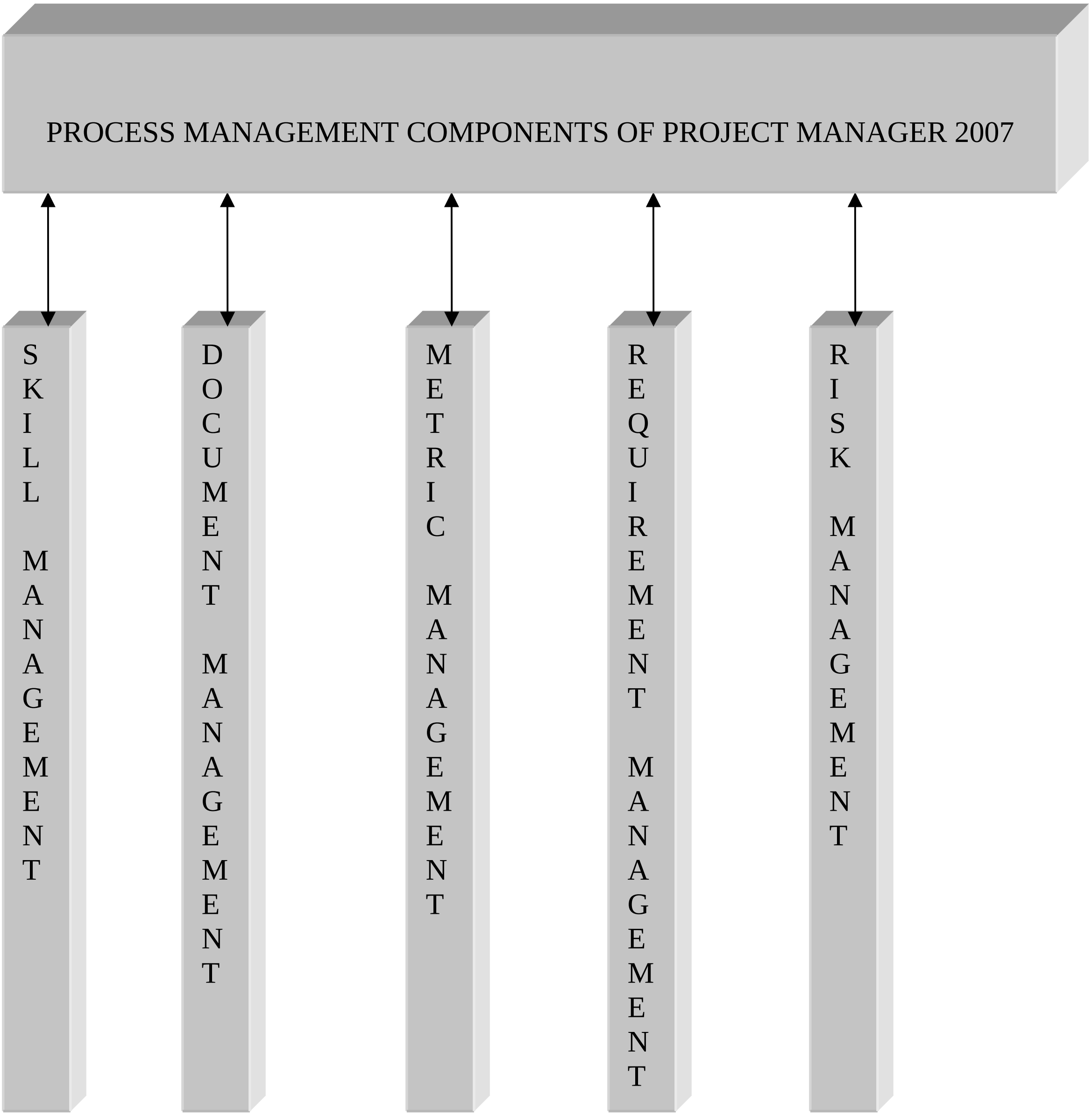


Figure 3. Process Management Components In Project Manager 2007

Figure 3 shows the Process Management components in Project Manager 2007. Process management compliments project management. Working together, these two management techniques allow for constant improvement in processes resulting in better applications with each iteration of the software life cycle. Process management is considered a vertical activity in Project Manager 2007 as they can be expanded and in the future will be turned into pluggable components. As companies adapt processes, they will be able to expand these verticals to fit into their organization. The current implementation includes the basics of process management in a simplified but extendable approach to fit the organization.

2.3.1. Skill Management

Skills in Project Manager 2007 mean any knowledge, specialty, certifications or education a user may have that will make the user a better candidate to be assigned to a certain project. According to Richard Murch [2], skill management is necessary for organizations “to align strategies and work force competencies with enterprise vision is to create a road map from vision to execution”.

The first time a user logs in, the user is expected to create a virtual personality of the user into the system. This is done with the use of skills. By default, Project Manager 2007 comes only with a default skill. An administrator or a manager in the system can enter more skills based on organization. Each skill must contain a skill type and administrators or managers can also create these skill types.

The use of skills is explained in the following example. Consider the situation where manager sets up a skill type called ‘Six Sigma Certification’. The manager then can include skills of ‘Green Belt’, ‘Black Belt’ and ‘Master Black Belt’ all assigned to the skill type of ‘Six Sigma Certification’. So when a user chooses skills, they can choose the certain level of certification they have on Six Sigma Certification.

As shown in the above example, a user can virtually display all their knowledge based on the skills the user has. This way, when a manager is looking to add resources to a department, he/she can ensure that the resources are compatible with the project being developed. This also helps identify levels of training needed for the resources and allows managers to have firmer hold on their staff. In addition, it also helps in making decisions on whether or not outside resources need to be hired.

2.3.2. Document Management

Every project contains some documents and hence document management is an important part of project management. Therefore, a separate entity on document management has been implemented within Project Manager 2007. A document is assigned to a project and has an owner.

Every document belongs to a unique document type. Project Manager 2007 comes only with a default project type. An administrator or a manager can enter more document types. A document contains a location indicating where the document can be found.

2.3.3. Metric Management

According to Stephen H. Kan [3], “It is undisputed that measurement is crucial to the progress of all sciences”. Metrics can be used to evaluate how an organization grades itself. A manager can gather, learn and improve upon processes using applicable metrics. Project Manager 2007 includes two types of metrics – internal and external metrics. .

An internal metric gathers project information from the database using an SQL statement. For example, a manager may want to find out how many tasks in the project ended up with activity successful completion percentage of over %75. The manager will use an internal metric to gather this information.

An external metric is composed of information captured through a different application that the manager makes use of to gather. This metric is usually used when a manager is compiling metrics to show to the higher level managers. For example, a team manager can create a code coverage metric and enter it into Project Manage 2007 metric. When the department manager runs the metrics related to the project, he/she will see this metric and can make use of it to influence their decisions. Another example of an external metric is bug counts. This metric could reveal activities in certain tasks regarding how many bugs are found with each activity. Because Project Manager 2007 does not include a bug reporting tool, an external tool is probably being used. The owner of this metric would collect this information and enter it into Project Manager 2007 as an external metric. In the future, external metrics can be extended to be able to log into other databases and run SQL statements against those databases and return back the results thus eliminating human intervention.

All metrics have owners so that each metric has a person responsible for it. All metrics also contain types so that they can be categorized correctly. Project Manager 2007 only comes with the default metric type but managers and administrators can enter more types.

2.3.4. Requirement Management

Requirement management is implemented within Project Manager 2007. Successful verification and validation of the requirements is what makes quality software.

Every requirement is assigned to a project and must contain a priority. The prioritization of a requirement allows the development team to identify riskier parts of a project, and to implement and validate them early on during the lifecycle of a project. The priority of a requirement also enables Project Manager 2008 (discussed later) to generate automatic risks based on the status of the requirement.

As mentioned, the status of the requirement is changed manually by its owner. Requirements can be initialized, partially covered or fully covered. For example if a requirement has a high priority and it is still in initialized state while the project schedule is 75% completed, Project Manager 2007 will automatically generate a risk.

Requirements are categorized based on their types. Project Manager 2007 comes with only the default type but an administrator or a manager can add more types.

2.3.5. Risk Management

Risk management is another component implemented in Project Manager 2007. Projects who realize risks before they happen and manages/mitigates them when they do happen will always attain higher quality. According to Steve McConnell [4], “Not practicing active risk management on a software project is tantamount to ignoring decades of experience in which the software industry has learned thousands of times that software development is a high-risk activity.” Project Manager 2007 allows for risks to be defined by users; in addition, risks can also be created based on hard-coded settings.

The manual addition of a risk to a process can be done by any user of the tool. When someone finds out there is a risk to the project, it can be added and assigned to it. A risk is uniquely assigned to its owner (the managers can re-assign ownership). A creator or owner of a risk can also enter more details of the risk, details to avoid it as well as mitigation details for the same.

The automatic addition of a risk occurs when the Project Manager 2007 application finds out that there is a risk to the project based on hard-coded settings. An example of this is in schedule. When a user edits an activity such as ‘actual effort’, the application calculates (based on the estimated end date) if the remaining effort is equal or greater than the remaining time for that activity. If the user has more work than possible hours allocated on desired end date of the activity, then automatically a risk is created and added to the project. The risk is assigned to either the project manager or the owner of that activity.

Risks in Project Manager 2007 must always have an origination. This way after the project is over, a manager can take a look to see where most risks are originating from. Project Manager 2007 only comes with default risk type during installation so administrators or managers can enter more risk types to fit the organization.

2.4. Development Goals of Project Manager 2007

Project Manager 2007’s main goal is not only project management but also help process engineering. With information being stored in a quantitative manner, it is easy to understand how processes affected projects. According Scott Berkun this behavior is found in many successful organizations. He states that[5] “…The Boeing Company, one of the largest airplane design and engineering firms in the world, keeps a black book of lessons it has learned from design and engineering failures. Boeing has kept this document since the company was formed, and it uses it to help modern designers learn from past attempts." Table 2 displays the reasons why Project Manager 2007 might be chosen for an organization for their project management needs.

	Factor	Reason
1	Fits into any organization	Project Manager 2007 fits into any organization irrespective of the number of projects the organization handles. Because it is a web-based application, upgrading or maintaining the application is a snap as well. Organizational structure within the company can be accurately simulated within Project Manager 2007, giving the clearest picture about the organization on a virtual world.
2	Fits into any development model	One can define any development process within Project Manager 2007. This functionality also allows for any project of any size to be represented correctly in Project Manager 2007.
3	Real world project management	Project Manager 2007 lets define a project with milestones, waypoints, tasks and activities going to the atomic level. In addition, Project Manager 2007 also allows creation and controls other components that go in parallel to project management such as risk management, resource management, document management, skill management and requirements management.
4	Story Mode	Story mode is a functionality of Project Manager 2007 where every action ever performed on the project by any user is kept in a chronological order. This way no matter what happens in the project, it is always kept in a detailed account and process engineers can analyze them later to find problems as they occur to make sure they do not occur in the future. According to Pankaj Jalote [6], "…the project does not end until a postmortem has been done to uncover what went wrong and why, and what worked and why. This analysis will enable the project manager and the team members to cull out key lessons on project execution."
5	Discover hidden risks automatically	In Project Manager 2007, there are some pre-existing conditions that automatically generate a risk associated with a project. For example, project components whose desired end dates have passed while component is in an initialized state would generate a risk. These hidden risks,

		which may go unnoticed to users, is caught by the application and will automatically generate a risk in the project which it is associated with. These way managers can take appropriate actions before risks are encountered.
6	Security	With a state of the art security system, Project Manager 2007 is secured against unauthorized actions by its users or third parties. Any and all actions taken in the system is recorded with an IP address and user id captured. If a user is found to be trying to do things they are not supposed to, the application can automatically lock that user out. Another aspect of security is the database integrity. In case the application realizes a breach of integrity is occurring, it can automatically take action from locking the system by sending out an email to an administrator.
7	Built in debug mechanism	Project Manager 2007 comes with a built in debug mechanism that administrators can use. This is especially helpful in identifying bugs or problem with Project Manager 2007 so that they can be fixed in a shorter amount of time by the support group. This mechanism also allows the administrator to sort active and resolved issues, making it easier for them to find and fix problems with the application.
8	Value	Project Manager 2007 is free to any and all. It is distributed via a GPU license. Anyone from students to multinational organizations can use Project Manager 2007 free of charge. Hence one of the marketing slogans for Project Manager 2007 is “Salvation is free”. In the future, a dedicated web page will be in place.

Table 2. Reasons for choosing Project Manager 2007

3. Design Process of Project Manager 2007

Project Manager 2007 is a web-based project management software. It is built using ASP.NET and C# programming language for the backend code. It uses a SQL Server 2000 database as its data repository; however, it can be adjusted to use SQL Server 2005 as well. It uses the .NET Framework version 1.1.4322. The installer is built on the MS Installer. Visual Studio 2003 was used to implement this project.

3.1. High Level Overview of Project Manager 2007

Project Manager 2007 is a web based application. The web server has the information on how to find the backend database in the web.config file, which it makes use of during user interaction. The user interacts directly with the client who sends the information to the server where server connects to the database to get information and then process it. Once the result is ready, it is sent back to the client where the user can see it. Figure 4 displays the high level architecture for Project Manager 2007.

Project Manager 2007 web-pages have two components. The client code defines the interactions with the server, the objects found on the server side for rendering and java script code for client based interaction with the page. Examples of the Java Script code are the clock on the pages, the messages that gets displayed to the user and disabling of the back forward buttons on the pages. The server code is the part where the logic component is built. All the actions associated with the user controls are encoded in here as well as all the database and helper class operations.

For best performance, the database and the web server should be in separate machines. However, they can run under the same server if needed. Figure 4 shows the high level architecture for Project Manager 2007.

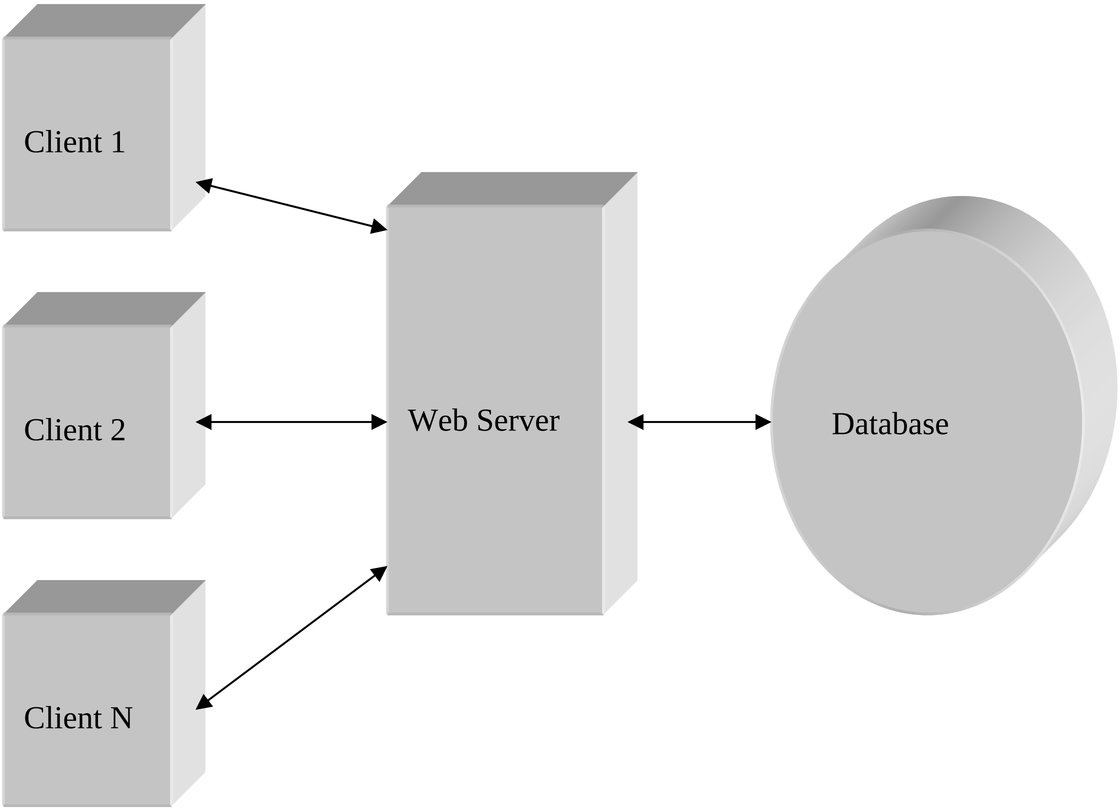


Figure 4. High Level Architecture For Project Manager 2007

3.2. *Class Architecture of Project Manager 2007*

Project Manager 2007 is made up of many classes. These classes are organized in two groups. The first group consists of the UI classes which implement project and process engineering components and are the classes user directly interacts with. The second group consists of the helper classes (called the function classes) which support the UI classes by implementing the logic behind the application. This class architecture was derived from the use case diagram as seen on Figure 5.

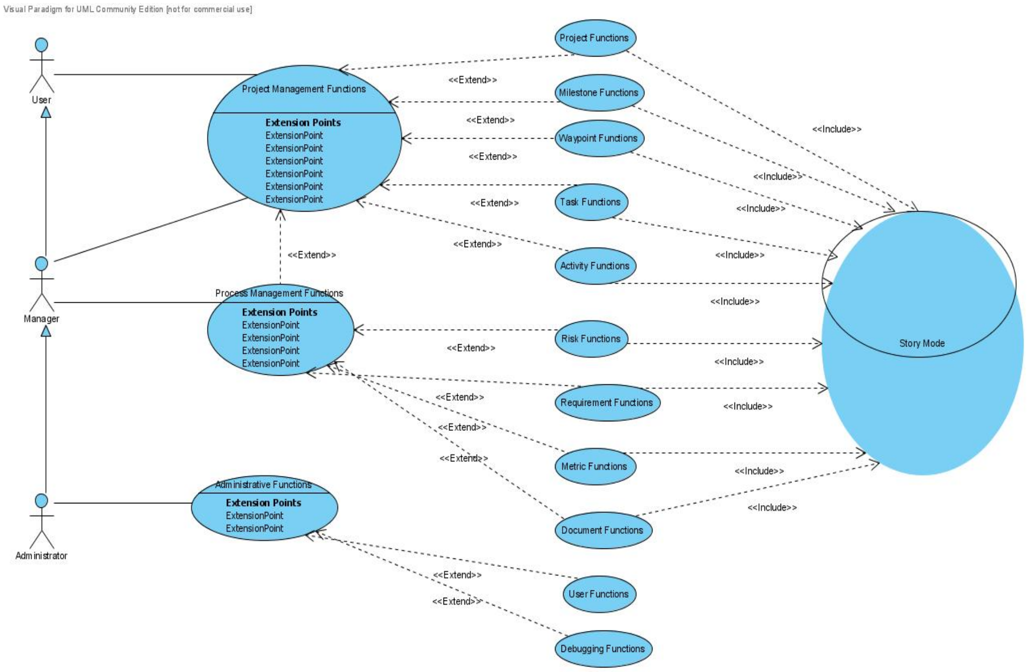


Figure 5. Use Case Diagram For Project Manager 2007

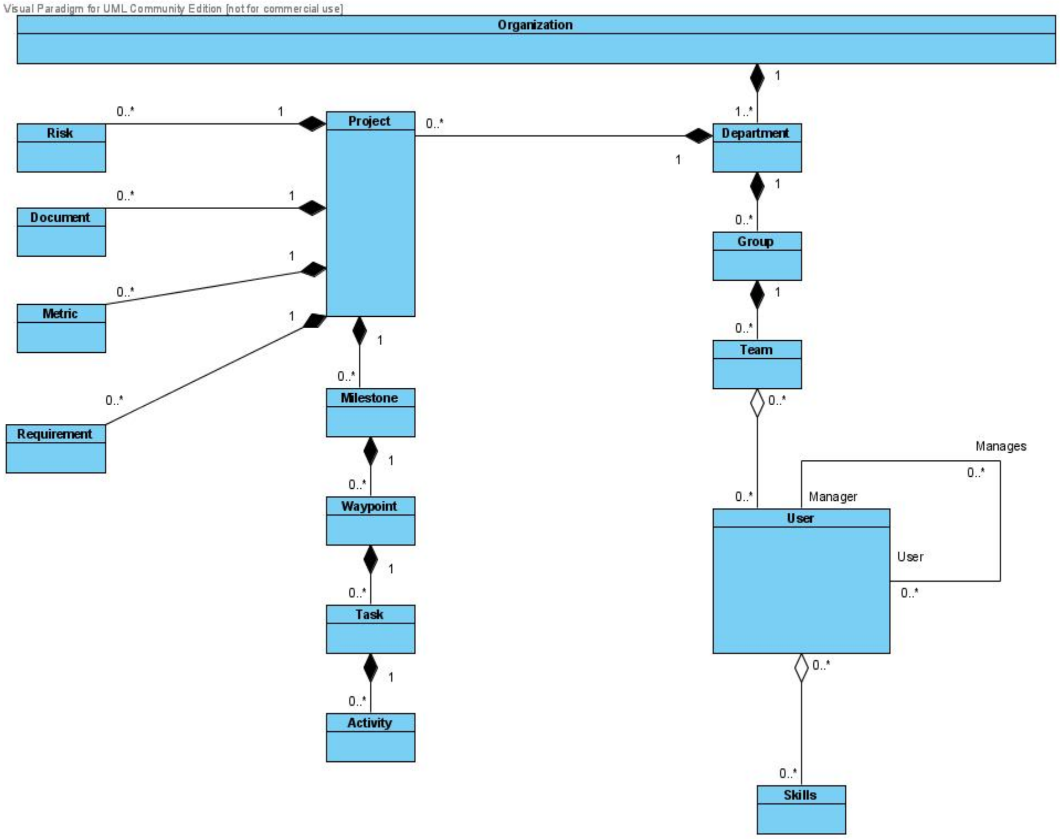


Figure 6. Class Diagram For Project Manager 2007

Each one of the project and process component classes also contains a helper class as mentioned. For example, the Requirement class in Figure 6 is implemented in the UI as AddRequirement class and at the back end as RequirementFunctions class. Using the AddRequirement class, a user of the application can add, edit, and delete requirements. AddRequirement class is associated with the helper class of RequirementFunctions class. It is this class which controls if the requirement already exists or if it generates a risk to the project, etc. The helper classes are associated with other helper classes (for example GeneralFunctions or DB_Utility or PrudentiaExceptions) which perform system wide operations or database operations or exception handling. Other helper classes complete the user interface but do not have anything to do with project or process management such as login, debugger and administrative pages.

Project Manager 2007 was designed to be highly cohesive and loosely coupled. Each class in Project Manager 2007 performs a specific tasks and only that task. They are aided with helper functions which perform tasks related to the part of the application while making use of a public interface for data access. As a loosely coupled application, changes to a component do not break other parts of the application. Due to this, Project Manager 2007 is very easy to maintain and expand.

Project Manager 2007 achieves five of the six identifiers of quality characteristics of software as defined by ISO 9126-1. The only requirement not met completely is portability; Project Manage 2007 can only run on SQL Server type databases. Other five requirements (functionality, reliability, usability, efficiency and maintainability) have been met and improvements will be made based on the Beta version.

3.2.1. Implementation of Project Manager 2007

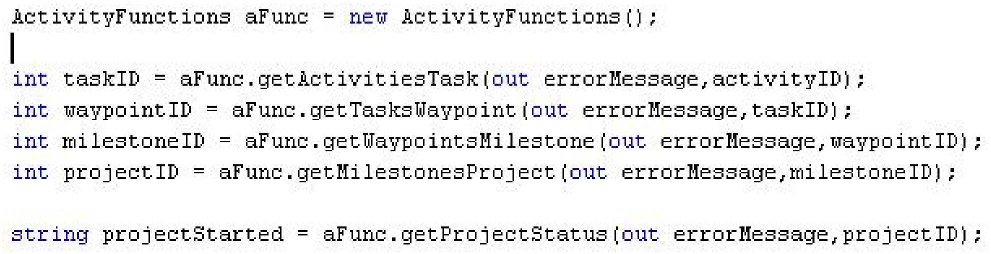
The first part of Project Manager 2007 was the database itself. Using the database schema diagrams created during design, database and its tables were created, along with the relationships between the tables.   Finally SQL Scripts were created to be used in the installer.

Once the database structure was in place, the next section was the security subsystem. Security subsystem was built as an improved version of a previous projects security system that was designed for a MSE class, Independent Study (Scheduling System for Physical Therapy Department). The security system was iterated numerous times throughout implementation, due to the fact that in implementation more scenarios were found than what was thought of at design time.

After the security system was built, administrator functionalities were built into the system. Project Manager 2007's own built-in bug capture/reporting tool was built so that all errors encountered during the application can be analyzed and fixed. This proved to be very helpful during beta stages as the application did not need to run in a debug environment. The debugging system was left in the application itself so that, in the future, administrators and people who are expanding this application can make use of it as well. This built-in error capture and reporting tool allows for all errors to be displayed in their entirety as well as exactly showing which line of code is the culprit. It functions as if the application is being run in a debug environment and has only a minimal footprint during processing.

Next, each one of the major classes was developed. For each one of the components, a function class was introduced according to the design document and was implemented. Many of these classes inherit each other. For example ActivityFunctions inherits TaskFunctions, TaskFunctions inherits WaypointFunctions, WaypointFunctions inherits MilestoneFunctions and MilestoneFunctions inherits ProjectFunctions. Due to this design, it became easy to implement against the database design and get information out of the database without having to initialize many classes.

Figure 7 shows how just having the activities id allows us to start every component up to the project level start when that activity starts. This is possible with the inheritance and in future it is easy for expansion. If inheritance was not used, each one of the classes would have to initialize taking more system resources.



```
ActivityFunctions aFunc = new ActivityFunctions();

int taskID = aFunc.getActivitiesTask(out errorMessage,activityID);
int waypointID = aFunc.getTasksWaypoint(out errorMessage,taskID);
int milestoneID = aFunc.getWaypointsMilestone(out errorMessage,waypointID);
int projectID = aFunc.getMilestonesProject(out errorMessage,milestoneID);

string projectStarted = aFunc.getProjectStatus(out errorMessage,projectID);
```

Figure 7. Inheritance In Project Manager 2007

Once the functions class for each component was built, user interface work began. While this was easy compared to other tasks, due to the dependency to .NET version 1.1, AJAX like functionalities were hard to implement. At this time, the menus were built along with the UI's of each component. While at design time there were three windows per component, aptly named add, edit and delete, later this was changed so that only one window was needed per component which could do all the three functions depending on how the user is navigating to the page.

After the UI was completed, the functionalities behind each UI element was implemented. This is the part where a lot of change and, unfortunately, some scope creep occurred. An example of this is, trying to implement different variations of the code to see their performances, memory footprints and database utilizations improvements. During the prototyping period, a lot of time was spent trying to improve the performance and interactivity of the application. Some page behind code was streamlined but, overall, the gain was marginal at best. Next, all the My[Component] pages were built, replacing the temporary buttons to navigate between pages.

3.3. Database Structure of Project Manager 2007

Project Manager 2007 database has gone through normalization process and due to this process it is classified as in first normal form. Each table contains its own data along with a primary key to uniquely identify each row. Further normalization was not required for this stage of the application but maybe performed in the future once usage and data collections are better analyzed. Figure 8, shows the database schema for project and process components.

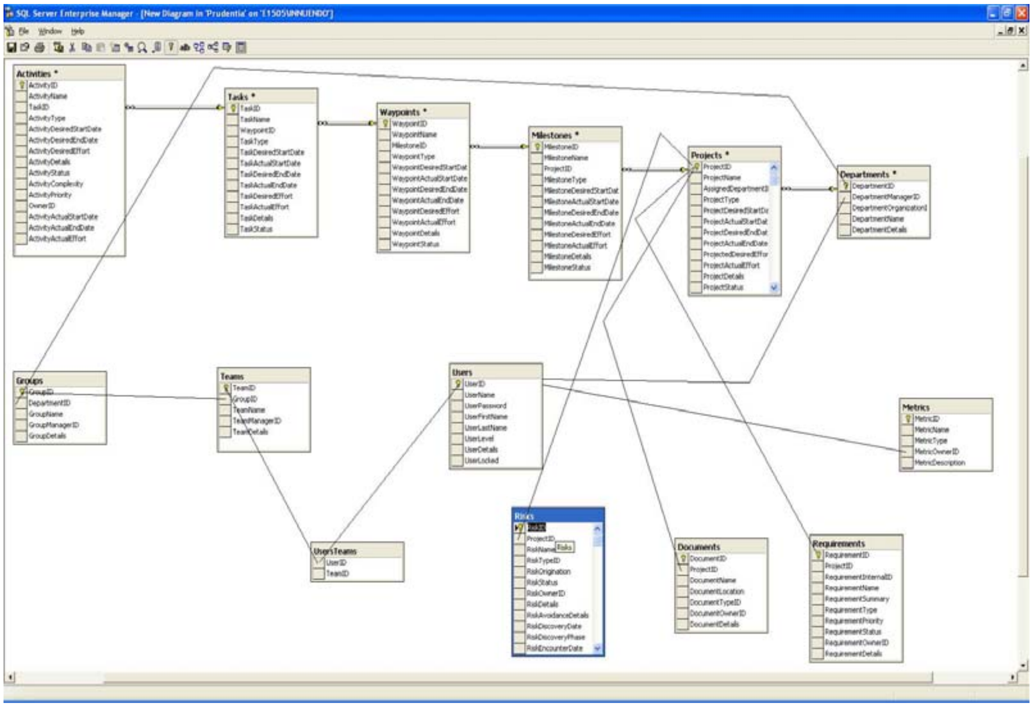


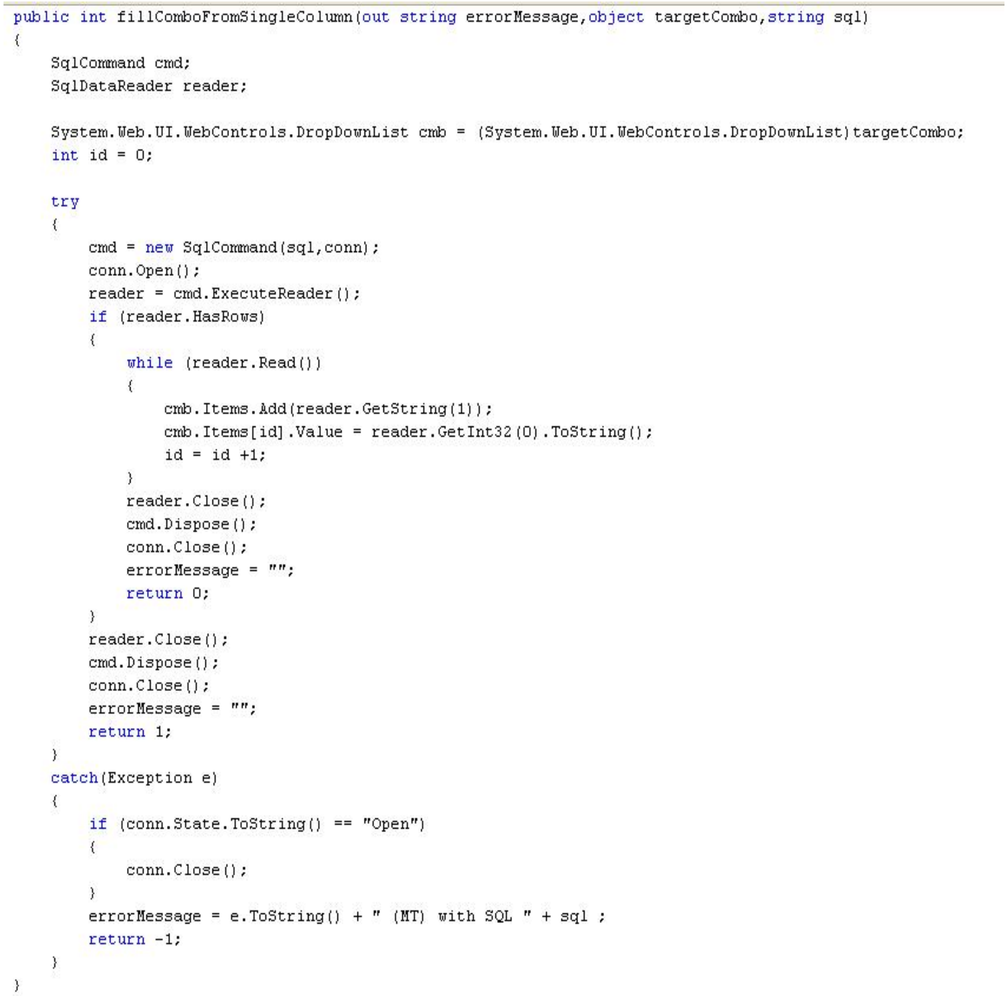
Figure 8. Database Schema For Project And Process Management

Nearly all calculations from the database are done directly at the database level. There is very little programming logic put into result sets, as most of the retrieval and analysis are done by the SQL Server itself. This frees up the CPU cycle of the web server and application itself; hence more users will be able to get their information faster.

Projects themselves do not contain any user information. They are assigned to departments and departments are broken down into groups, teams and users. Now imagine the scenario if it was decided to implement this in code rather then in database.

First application would have to get all users in a data structure (a dataset) and then, one by one, find out the team they are part of, then find out and then the department who owns that project. It would be an expensive and invasive operation; moreover, the application would also have to handle all the error scenarios. It would be done at the server level hence CPU cycles would be used in getting this information and another user could be waiting for the server to respond to their request.

Project Manager 2007 makes use of a class called DBUtility to interact with the database. The purpose of this class is to make DB interaction easier and more manageable. All connections to the database are handled by this one class. This class also handles all interactions with GUI objects and database operations. Here is an example:

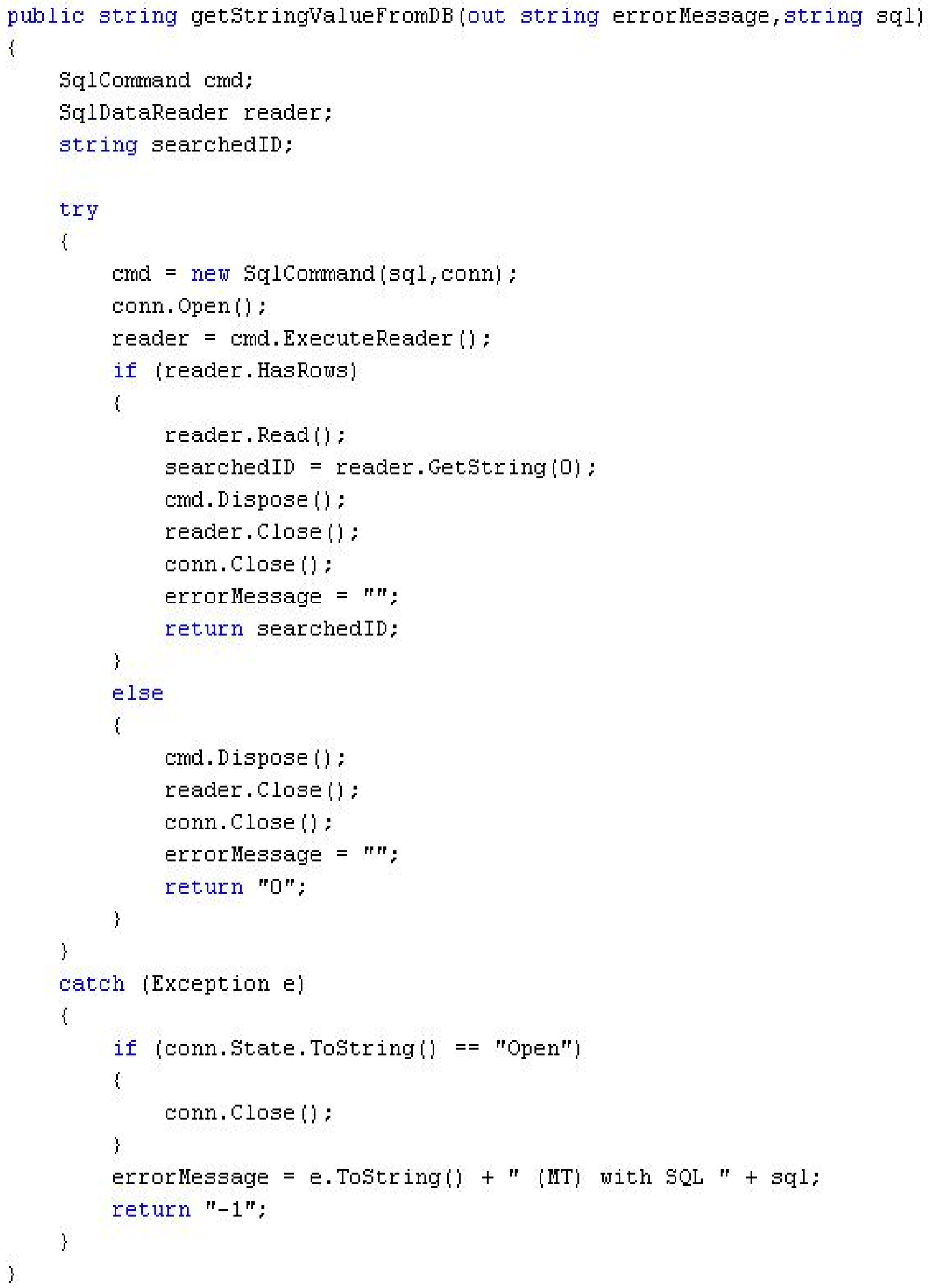


```
public int fillComboFromSingleColumn(out string errorMessage,object targetCombo,string sql)
{
    SqlCommand cmd;
    SqlDataReader reader;

    System.Web.UI.WebControls.DropDownList cmb = (System.Web.UI.WebControls.DropDownList)targetCombo;
    int id = 0;

    try
    {
        cmd = new SqlCommand(sql,conn);
        conn.Open();
        reader = cmd.ExecuteReader();
        if (reader.HasRows)
        {
            while (reader.Read())
            {
                cmb.Items.Add(reader.GetString(1));
                cmb.Items[id].Value = reader.GetInt32(0).ToString();
                id = id +1;
            }
            reader.Close();
            cmd.Dispose();
            conn.Close();
            errorMessage = "";
            return 0;
        }
        reader.Close();
        cmd.Dispose();
        conn.Close();
        errorMessage = "";
        return 1;
    }
    catch(Exception e)
    {
        if (conn.State.ToString() == "Open")
        {
            conn.Close();
        }
        errorMessage = e.ToString() + " (MT) with SQL " + sql ;
        return -1;
    }
}
```

Figure 9. GUI Actions Based On Database Data

In Figure 9, ‘fillComboFromSingleColumn’ method allows the GUI (or another class) to call the DB_Utility class and fill a combo box on the User Interface with information stored in the database. For example, in the ‘AddAnActivity’ window, the combo box containing users and combo box containing the activity types are populated in this way.

In addition to interacting with user objects, DB_Utility class is also used for getting values out of the database. Here is an example



```
public string getStringValueFromDB(out string errorMessage,string sql)
{
    SqlCommand cmd;
    SqlDataReader reader;
    string searchedID;

    try
    {
        cmd = new SqlCommand(sql,conn);
        conn.Open();
        reader = cmd.ExecuteReader();
        if (reader.HasRows)
        {
            reader.Read();
            searchedID = reader.GetString(0);
            cmd.Dispose();
            reader.Close();
            conn.Close();
            errorMessage = "";
            return searchedID;
        }
        else
        {
            cmd.Dispose();
            reader.Close();
            conn.Close();
            errorMessage = "";
            return "0";
        }
    }
    catch (Exception e)
    {
        if (conn.State.ToString() == "Open")
        {
            conn.Close();
        }
        errorMessage = e.ToString() + " (MT) with SQL " + sql;
        return "-1";
    }
}
```

Figure 10. Getting Values Out Of The Database Example

In Figure 10, the method ‘getStringValueFromDB’ allows any caller to reach to a string value in db and returns it back. This method is used in getting string values out of a database such as when retrieving an activity name that is already stored in database.

A benefit of having the DB_Utility function is that, if the database changes to a different one (lets say Oracle), it will be easy for maintenance to adapt to the Oracle data types. The only update will be to the DB_Utility class and database specific sql can be used within the application.

As explained above database operations in Project Manager 2007 is always done using DB_Utility. However other classes construct SQL statements before passing it on to the DB_Utility for executionDatabase operations are not reversible in Project Manager 2007. The only way to reverse transactions is either to restore the database, do it at a database administrator level or undo actions manually with the help of an administrator. The reason for not employing the concept of transactions was due to the fact there are alternate methods to protect data integrity and user actions. Transactions usually have a larger footprint on database operations and CPU cycles. Because of this reason, transactions were left out of Project Manager 2007 although they may make a comeback in future versions depending on feedback from users.

All database connections opened from Project Manager 2007 is closed at the end of the operation. This decreases the workload of the database server.

Database information in Project Manager 2007 is stored in the web.config file. While password for the DB is not encrypted, this security flaw is very hard to find. This is due to .NET framework 1.1 not supporting encrypted web.config files and will be fixed in Project Manager 2008.

3.4. Security in Project Manager 2007

A web application gains benefit from the fact that anyone can access it. However this benefit works as a two-edged sword. A web application uses more resources and has more points of vulnerability than a desktop application. Adding a database to the equation just adds up to its complexity.

Most of the time, the threats come from users within the application, doing things that they are not supposed to. They can access secret information, change data they are not supposed to change, delete information etc.

Due to these reasons, Project Manager 2007 uses multiple levels of security. In this section we will explore the security system of Project Manager 2007.

3.4.1. Logging in Users and Creating tickets

Every user must login to the Project Manager 2007. Each user has an assigned username and password. If the user is successful with the login mechanism, an active ticket is created for the user and the user is allowed to access the database. This ticket expires depending on the time set by the administrator in the web.config file. Ticket expiration only occurs when a user is inactive.

When the ticket is created, the user is also captured in the activity logs along with the IP address of the client from which the user has logged in and entered into the active users list. If the user is not active (due to an administrator forcing the user out), he/she will be taken out of the application.

3.4.2. Verifying User Levels

Every user is always checked for his/her credentials. As explained in the database tables guide, NavigationalPermissions table defines what user levels are allowed at which pages. Whenever a user navigates to a page, the user's level is retrieved and then checked against the NavigationalPermissions table. If the user is on a page he /she should not be, then the user is shown a default page and returned to the home page.

3.4.3. Navigational Security

As discussed before, the only way for users to navigate within the Project Manager 2007 application is to use either the buttons or menus on a page. For most of the pages in Project Manager 2007, three different menus exist for three different user types. Depending on the user level gathered during page load, the appropriate menu loads up, thus crippling the user's ability to navigate to areas they are not suppose to see.

Another expectation of navigational security is the use of session variables. Session variables are used in each page during transfer and page load as static variables while the page is active. The session variables contain, the purpose for which the user has opened this page, return address once the users work on the page is completed and various variables for selecting the item to edit or delete (if the users purpose is edit or delete). In case the system realizes that the user enters a page without these variables, the user is taken to an error page and then back to his/her homepage.

Another aspect of navigational security is the locking of the back and forward buttons. By simple java script code, the browser is locked from using the back and forward buttons. So the user cannot try to throw off navigational security, nor can it cause problems for the application.

3.4.4. Execution Security

Because security also includes disabling the user's ability to enter invalid information to the database, in case of fatal runtime errors, Project Manager 2007 has the ability to lock a page. Each GUI class in Project Manager 2007 contains a function called 'disableControls'. When this function is called, all objects that cause a post back of the page are disabled except for the navigational option to go back to where the navigation started . This stops users from seeing multiple errors and also from database being written with invalid data.

3.4.5. Activity Security

Every action in Project Manager 2007 is recorded. There are two immediate benefits to this: the first being is that everything is easily traceable and secondly it allows for projects to be replayed back in details. Even the administrator actions are recorded and there is no way to delete these actions from the application. One has to delete them from the database directly.

3.5. Error Handling in Project Manager 2007

Project Manager 2007 handles and throws its own errors. There are two error types and both are derived from the system ApplicationException class.

Most errors in Project Manager 2007, originate from DB_Utility class. When a class is interacting with the DB_Utility class, and the expected result does not exist or an expected value is not returned or an exception is thrown due to a database or sql error, DB_Utility returns an error value, signaling the caller class that an exception has occurred. The caller class, by looking at the return value, can raise an exception which gets handled within the code.

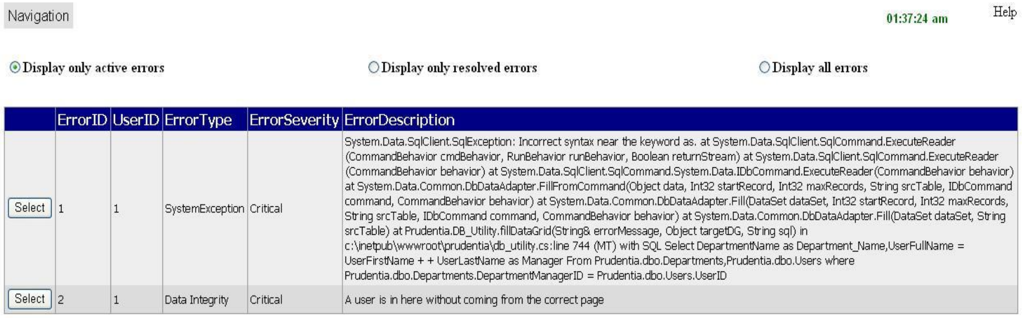
If the value was not expected to be null and yet it is returned as such from the DB_Utility, that means that a data integrity error is in progress. A specific PrudentiaException (with type of DataIntegrity) is thrown and the original caller catches this exception, and handles the processing on the page accordingly.

Sometimes the value returned from DB_Utility can be empty but it is not an error. At this time, a specific PrudentiaException is thrown (with type of administrative) and the user is given a pop-up warning message explaining what the problem is.

Generally, if an application error is encountered, processing should not continue. At this time, a SystemException is raised from the DB_Utility and the caller class passes this along to the GUI class. The GUI class handles this error by locking the page and showing the specific message back to the user. These errors usually come from SQL Server database (such as a malformed SQL statement).

From time to time, GUI control exceptions are thrown as well from the application. These may or may not come from the DB_Utility class. If it’s from the DB_Utility class, it is handled as a SystemException. If it’s coming from the GUI class, then they are handled as a standard exception. At both cases, the page is locked and the user can only go back.

Lastly, Project Manager 2007 employs a powerful internal debugger type utility as seen in Figure 11 and 12.



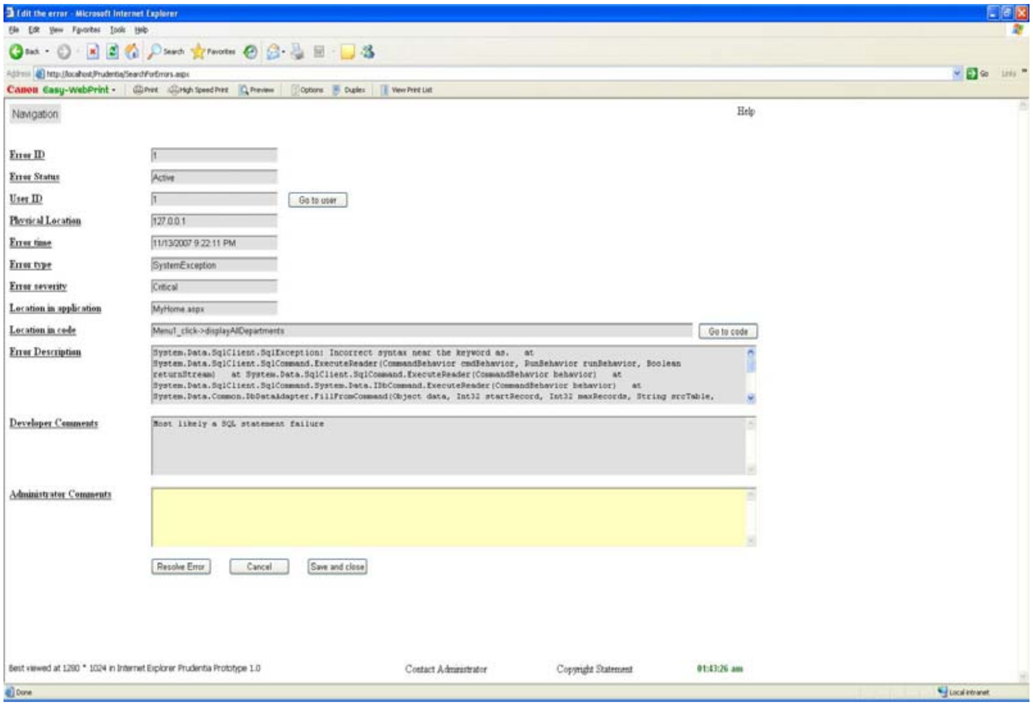
● Display only active errors ○ Display only resolved errors ○ Display all errors

	ErrorID	UserID	ErrorType	ErrorSeverity	ErrorDescription
Select	1	1	SystemException	Critical	System.Data.SqlClient.SqlException: Incorrect syntax near the keyword as. at System.Data.SqlClient.SqlCommand.ExecuteReader (CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream) at System.Data.SqlClient.SqlCommand.ExecuteReader (CommandBehavior behavior) at System.Data.SqlClient.SqlCommand.System.Data.IDbCommand.ExecuteReader(CommandBehavior behavior) at System.Data.Common.DbDataAdapter.FillFromCommand(Object data, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior) at System.Data.Common.DbDataAdapter.Fill(DataSet dataSet, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior) at System.Data.Common.DbDataAdapter.Fill(DataSet dataSet, String srcTable) at Prudentia.DB_Utility.fillDataGrid(String& errorMessage, Object targetDG, String sql) in c:\inetpub\wwwroot\prudentia\db_utility.cs:line 744 (MT) with SQL Select DepartmentName as Department_Name,UserFullName = UserFirstName + + UserLastName as Manager From Prudentia.dbo.Departments,Prudentia.dbo.Users where Prudentia.dbo.Departments.DepartmentManagerID = Prudentia.dbo.Users.UserID
Select	2	1	Data Integrity	Critical	A user is in here without coming from the correct page

Figure 11. Error Generated During Usage Of Project Manager 2007

As seen on Figure 11, the administrator can filter what type of errors to view, active errors are the default for this page. Figure 12 shows the screen when the administrator selects the error. More information about the error is displayed at this page.

The user can also see the error type, error severity and the error message as well as the user who encountered the error.


Edit the error - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Prudentia/SearchForErrors.aspx
Go Links
Canon Easy-WebPrint Print High Speed Print Preview Options Duplex View Print List
Navigation
Help
Go to user
Go to code
Resolve Error
Cancel
Save and close
Best viewed at 1280 * 1024 in Internet Explorer Prudentia Prototype 1.0
Contact Administrator
Copyright Statement
01:43:26 am
Done
Local intranet

Error ID	1
Error Status	Active
User ID	1
Physical Location	127.0.0.1
Error time	11/13/2007 9:22:11 PM
Error type	SystemException
Error severity	Critical
Location in application	MyHome.aspx
Location in code	Menu1_click->displayAllDepartments
Error Description	System.Data.SqlClient.SqlException: Incorrect syntax near the keyword as. at System.Data.SqlClient.SqlCommand.ExecuteReader(CommandBehavior cmdBehavior, RunBehavior runBehavior, Boolean returnStream) at System.Data.SqlClient.SqlCommand.ExecuteReader(CommandBehavior behavior) at System.Data.SqlClient.SqlCommand.System.Data.IDbCommand.ExecuteReader(CommandBehavior behavior) at System.Data.Common.DbDataAdapter.FillFromCommand(Object data, Int32 startRecord, Int32 maxRecords, String srcTable,
Developer Comments	Most likely a SQL statement failure
Administrator Comments	

Figure 12. Detailed Error Display Window In Project Manager 2007

In this page, the administrator can see the error in its full details. The administrator at this time can navigate to see the logs for the user who encountered this error or the code that encountered the error. The administrator can also see the error description as given by the system (OS, DB, etc..) as well as any developer comments as to what to look for when this error is encountered. Lastly, the administrator can type his own comments (for example how the error was resolved or how this was not a real error message because the DB was down at the moment) and resolve or save changes to the error.

3.5.1. PrudentiaException Class

PrudentiaException class extends the System.ApplicationException class. When an exception is encountered and this class is initialized, it is constructed with an exception number. For example, when the user is trying to login to the system, if the user name and password do not match, login class calls raises a new PrudentiaException(3), which is userNameAndPasswordDoesNotMatchException. When this exception is raised, PrudentiaException class, internally searches for a method of the same name as the exception number. When the method is found, it is called. The method then sets the error description and error type. At the end of this process a new PrudentiaException object is formed and active.

As talked about earlier, there are two exception types. The first type is "Administrative" and when an exception is thrown as this type, Project Manager 2007 displays a warning message and the user is given instructions on what is wrong. These kind of "Administrative" errors are thrown when user forgets to type a necessary field, if the expected type is not the entered type or if a simple error has occurred.

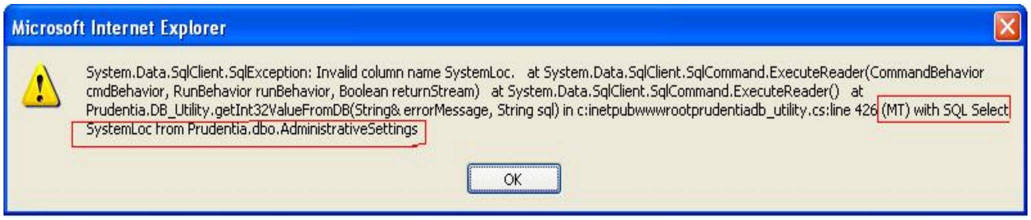
The second type of error is the dreaded "Data Integrity" error. When a data integrity type error is discovered, it means that the data in the database is not dependable anymore. An automatic email to the administrator is sent. A data integrity error works the same way as an administrative error except when the GUI class realizes the type of error, it will automatically lock that page to further use by the user.

Project Manager 2007 defines its own PrudentiaException class error messaging system for a single purpose. It is to make sure that all logical errors are captured (PrudentiaExceptions do not contain any application errors) thus helping administrators and support personnel access to better error messaging, Because these error messages are not application errors but instead logical errors, they need to be explained during development as much as possible so that in the future fixes will be easier.

3.5.2. SystemException Class

SystemException class extends the System.ApplicationException. It adds a string value after the actual error message thrown by the database, operating system or the .Net framework.

The SystemException class is initialized with a string value and adding it to the derived base classes message value. Figure 13 shows an example of a SQL error that occurred and shown back to the user.



```
System.Data.SqlClient.SqlException: Invalid column name SystemLoc.   at System.Data.SqlClient.SqlCommand.ExecuteReader(CommandBehavior
cmdBehavior, RunBehavior runBehavior, Boolean returnStream)   at System.Data.SqlClient.SqlCommand.ExecuteReader()   at
Prudentia.DB_Utility.getInt32ValueFromDB(String& errorMessage, String sql) in c:inetpubwwwrootprudentiadb_utility.cs:line 426 (MT) with SQL Select
SystemLoc from Prudentia.dbo.AdministrativeSettings
```

Figure 13. An Error Message Displayed To The User

This simple but powerful approach allows passing extra information in the error message thrown by the application. This extra information is useful in seeing the syntax that is causing trouble and more informative for an administrator and support person so that issues can be resolved much quicker.

3.6. User Interface Design of Project Manager 2007

Project Manager 2007 makes use of server side controls of Visual Studio .NET for displaying controls. It makes use of Java Script supplementing C#.

Project Manager 2007 is made up off three types of pages. These are component pages, homepages and administrative pages. A page is considered a component page if it's used to add/edit/delete any part of project or process management components. A page is considered a homepage if it's used to navigate to the component pages. Homepages contains appropriate actions and lists that shows the components accessible from that page. A page is considered an administrative page if only an administrator (or sometimes a manager) can access it and change or view system values. .

Each page in Project Manager 2007 follows the same UI interface standards. Table 4 displays UI standards that were designed and followed in the implementation of Project Manager 2007

	UI Standard	**Reason**
1	Optimize page for viewing	Each page is optimized for viewing at 1280 * 1024 resolution in Internet Explorer 6.x
2	Have a clock on each page	Each page contains a clock showing the current system time.
3	Access to help at each page	Each page contains a link to the help system. When this link is clicked, it displays the help for

		the current page in a separate window.
4	Give feedback to the user	When the user adds, edits or deletes a component, show success or failure messages to the user
5	Separate menus for different user types	Each page contains a separate menu for users to control their actions.

Table 3. UI Standards of Project Manager 2007

3.7. Testing Process of Project Manager 2007

The testing types executed were, GUI testing, unit testing (for the component which was used during a MSE class), integration testing and some system testing. GUI testing consisted of verifying that the component built followed the UI standards of defined in design and of Microsoft (applicable concepts). Unit testing was performed for only one component, which is the DB_Utility class, and it was performed because a driver already existed to run unit tests for that class. This driver was enhanced during development of Project Manager 2007. Integration tests occurred when the backend implementation was completed for the component that was being built. For example, for component pages, this included testing the adding, editing and deleting of the components during different scenarios. The test cases were pre-built but were expanded during the testing process. Finally, some system tests occurred for non-project\process management components such as the error handling mechanism or the debugger. System tests could occur for these components because they were finished and no changes except for bug fixes were expected for these components.

Towards the end of the development, system, security, usability and acceptance tests occurred. The project and process management systems were tested using pre-designed scenarios. During security testing, system as whole and as individual pages was tested. Once again, test cases were expanded as more scenarios were discovered during testing. Usability testing was performed during beta process by meeting with beta users and learning from their experiences. Finally, during acceptance testing, pre-existing tests cases were run on the application to validate and verify that it matched the requirements and design. Once the acceptance tests passed, the application was ready for release.

3.8. Installation Process of Project Manager 2007

Project Manager 2007 is installed via a MS Installer. The installer was created using Visual Studio 2003 and the installer is actually an application in itself. During the install process, Project Manager 2007 is fully implemented to the target machine, meaning users who have access to the actual server machine can actually make changes to the code by opening the project. This was decided because Project Manager 2007 is a free application and because it allows faster bug fixes as well as debugging environment.

The database of Project Manager 2007 is available in .sql files which creates the database tables. The database must be created by an administrator before these files can be loaded into it.

4. Future work in Project Manager 2007

4.1. Next Step

Project Manager 2007 will be in beta stage by November 27, 2007. Once version 1.0 is out, work will start on bugs that have not been fixed and at the same time, a host will be found to upload the application and its dedicated webpage.

Once feedback is received from users, work on Project Manager 2008 will begin. This release will upgrade the development environment to Visual Studio 2008 and .NET Framework 3.0. Main goal of Project Manager 2008 will be to implement AJAX components to make the application more interactive.

Future versions will include a pluggable component design that will integrate with the application without any compilation. It will also have more interactive AI structure, more statistical information about projects and processes, virtual project scenarios, visual tutorial, a scenario based game where a user manages projects with changing problems, visual graphs, remote management, running as a service and a natural language query support system.

5. Competitors of Project Manager 2007

There are quite a few competitors in the field of project management against Project Manager 2007. Some of them are quite good while some of them are crude such as Excel spreadsheets sold as products. However, only a select few fuses software engineering principles with project management. Commercial applications are expensive and organizations have to adapt to these applications instead of the application fitting the organization. Table 5 displays the competitors [1] of Project Manager 2007.

	Competitor	**Description**
1	dotProject	Most extensively used open sourced web application for project management. Supports basic project management activities and has a plug and play module design. Supports basic process management such as risk management.
2	Project.net	A commercial open source web based application which includes project management as one of its components as well as many others such as vendor tracking. No process management functionality was found.
3	ProjectPier	A free open source web based application implements a project as a grouping of tasks. No process management functionality was found.
4	Trac	Trac is a free, open source, web based software development tool. It incorporates project management as part of the development process but not process management.
5	Microsoft Office Project	Microsoft Office Project Standard 2007 is a commercial web based application used in project management. It only allows for limited process management functionality such as resource management.
6	VPMi	VPMi is a commercial, web based project and process management software. It is the leader in the field of project management.
7	24SevenOffice	24SevenOffice is a commercial, web based CRM software which includes project management as one of its components.
8	AtTask	AtTask is a web based, commercial, project and process engineering software.

9	JIRA	JIRA is a commercial, web based project management software. It implements project management by mapping the business's processes into the lifecycle of the software being developed. It does not contain process management components.
10	Project Insight	ProjectInsight is a commercial, web based project management software. It does not account for process management components.

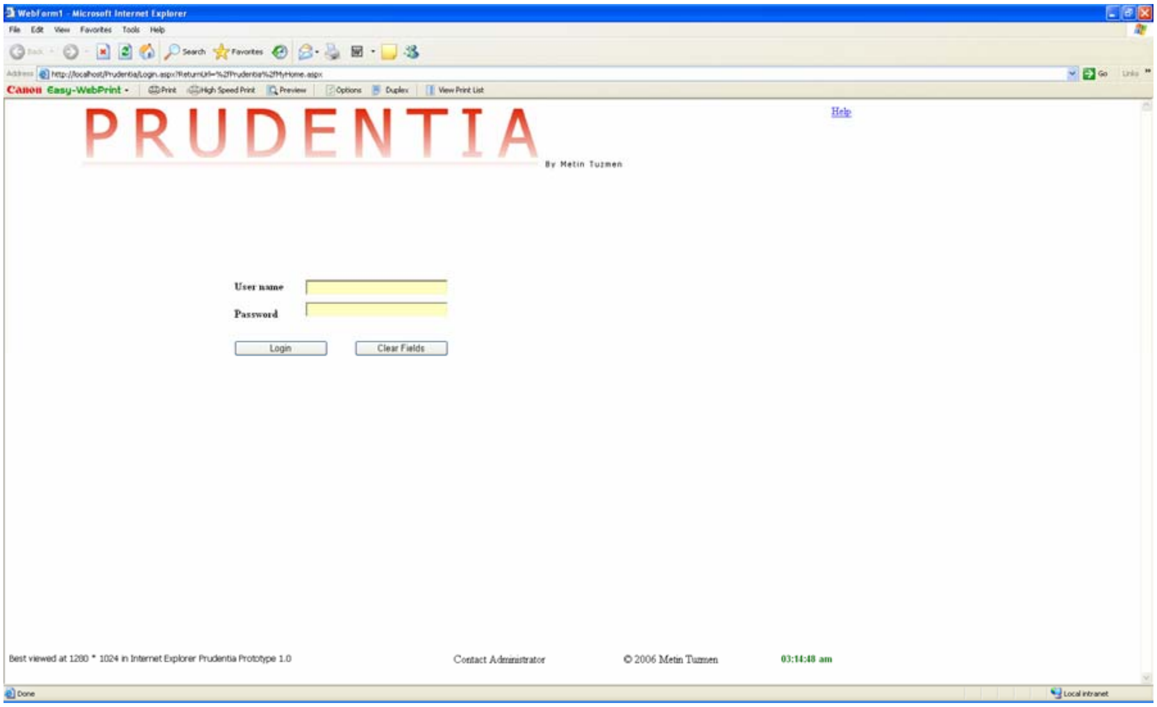
Table 4. Competitors of Project Manager 2007

Overall Project Manager 2007 stacks very well against the current applications available to organizations. Project Manager 2007 differentiates itself in terms of process management and easy adaptability to companies as well as its ease of use. Being free and source code being given allows organizations to better fit the application to their specifications and grow as the organization grows. Project Manager 2007 is an organic and dynamic application which in a few future versions would be fit to compete against commercial products in the area of project management. In the area of process management, it currently has only two competitors to catch up to which are WPMi and AtTask and can surpass them in the future.

BIBLIOGRAPHY

1. http://en.wikipedia.org, Wikipedia, Wikimedia Foundation, Inc., November 2007
2. R. Murch, Project Management: Best Practices for IT Professionals, Prentice Hall, 2000
3. S. H. Kan, Metrics and Models in Software Quality Engineering, Addison Wesley Professional, 2002
4. S. McConnell, Software Project Survival Guide, Microsoft Press, 1997
5. S. Berkun, The Art of Project Management, O'Reilly, 2005
6. P. Jalote, Software Project Management in Practice, Addison Wesley Professional, 2002
7. E. Dustin, Effective Software Testing: 50 Specific Ways to Improve Your Testing, Addison Wesley Professional, 2002

APPENDIX A: Selected Functionality Screenshots of Project Manager 2007


WebForm1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Canon Easy-WebPrint Print High Speed Print Preview Options Duplex View Print List
PRUDENTIA
By Metin Tuzmen
Help
User name
Password
Login
Clear Fields
Contact Administrator
Done
Local intranet

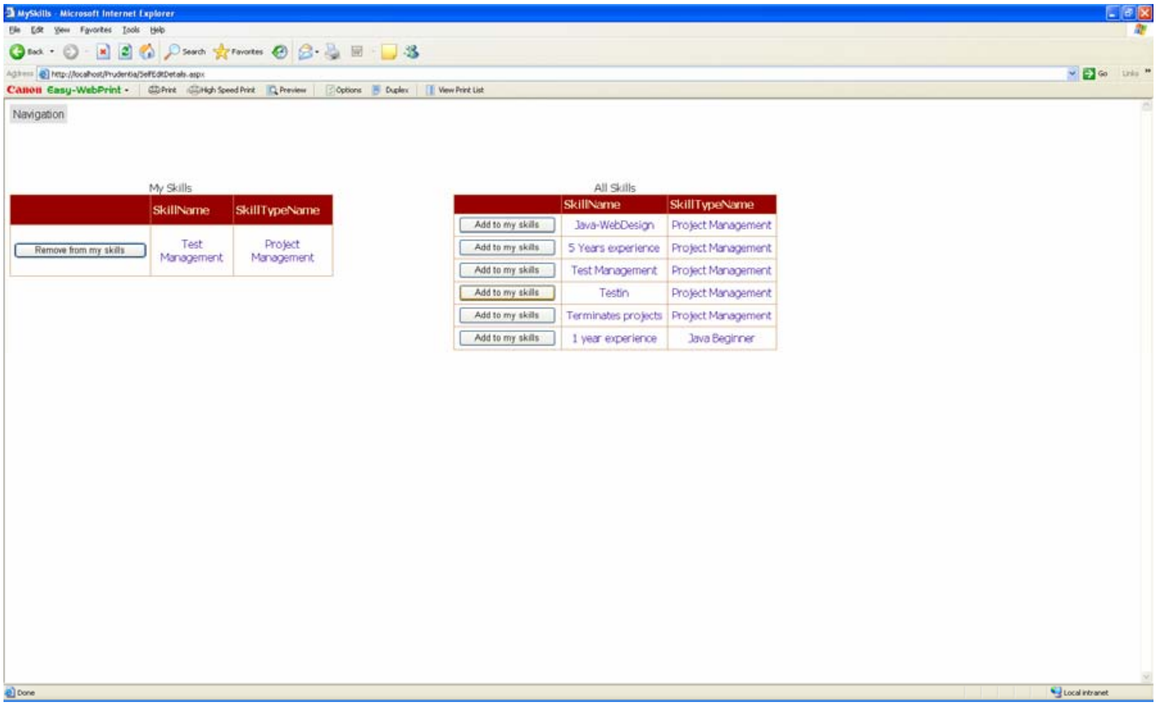
http://localhost/Prudentia/Login.aspx?ReturnUrl=%2fPrudentia%2fMyHome.aspx

Best viewed at 1280 * 1024 in Internet Explorer Prudentia Prototype 1.0

© 2006 Metin Tuzmen

03:14:48 am

Figure 14. Login Screen of Project Manager 2007



MySkills - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Prudentia/SelfEditDetails.aspx
Go Links
Canon Easy-WebPrint Print High Speed Print Preview Options Duplex View Print List
Navigation
Done
Local intranet

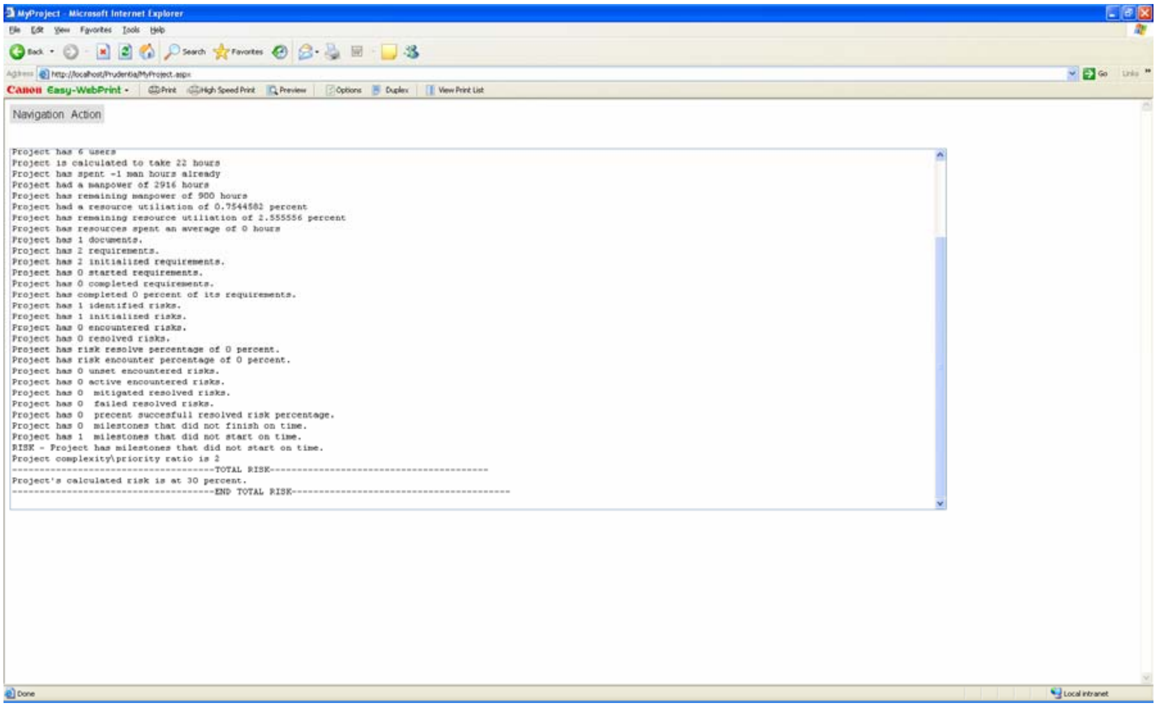
My Skills

	SkillName	SkillTypeName
Remove from my skills	Test Management	Project Management

All Skills

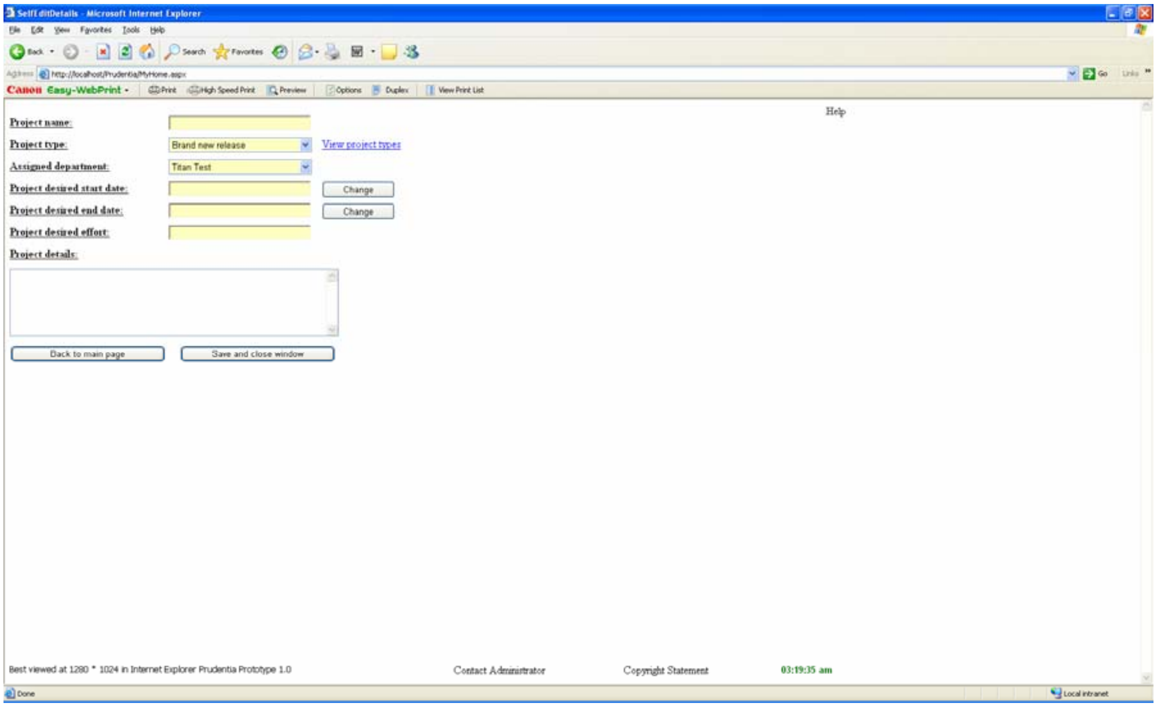
	SkillName	SkillTypeName
Add to my skills	Java-WebDesign	Project Management
Add to my skills	5 Years experience	Project Management
Add to my skills	Test Management	Project Management
Add to my skills	Testin	Project Management
Add to my skills	Terminates projects	Project Management
Add to my skills	1 year experience	Java Beginner

Figure 15. User Skill Management screen



```
Project has 6 users
Project is calculated to take 22 hours
Project has spent -1 man hours already
Project had a manpower of 2916 hours
Project has remaining manpower of 900 hours
Project had a resource utiliation of 0.7544582 percent
Project has remaining resource utiliation of 2.555556 percent
Project has resources spent an average of 0 hours
Project has 1 documents.
Project has 2 requirements.
Project has 2 initialized requirements.
Project has 0 started requirements.
Project has 0 completed requirements.
Project has completed 0 percent of its requirements.
Project has 1 identified risks.
Project has 1 initialized risks.
Project has 0 encountered risks.
Project has 0 resolved risks.
Project has risk resolve percentage of 0 percent.
Project has risk encounter percentage of 0 percent.
Project has 0 unset encountered risks.
Project has 0 active encountered risks.
Project has 0  mitigated resolved risks.
Project has 0  failed resolved risks.
Project has 0  precent succesfull resolved risk percentage.
Project has 0  milestones that did not finish on time.
Project has 1  milestones that did not start on time.
RISK - Project has milestones that did not start on time.
Project complexity\priority ratio is 2
-------------------------------------TOTAL RISK----------------------------------------
Project's calculated risk is at 30 percent.
-------------------------------------END TOTAL RISK----------------------------------------
```

Figure 16. Project Status Screen


SelfEditDetails - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Prudentia/MyHome.aspx
Go Links
Canon Easy-WebPrint Print High Speed Print Preview Options Duplex View Print List
Help
Change
Change
Back to main page
Save and close window
Contact Administrator
Copyright Statement
Done
Local intranet

Project name:

Project type: Brand new release View project types

Assigned department: Titan Test

Project desired start date:

Project desired end date:

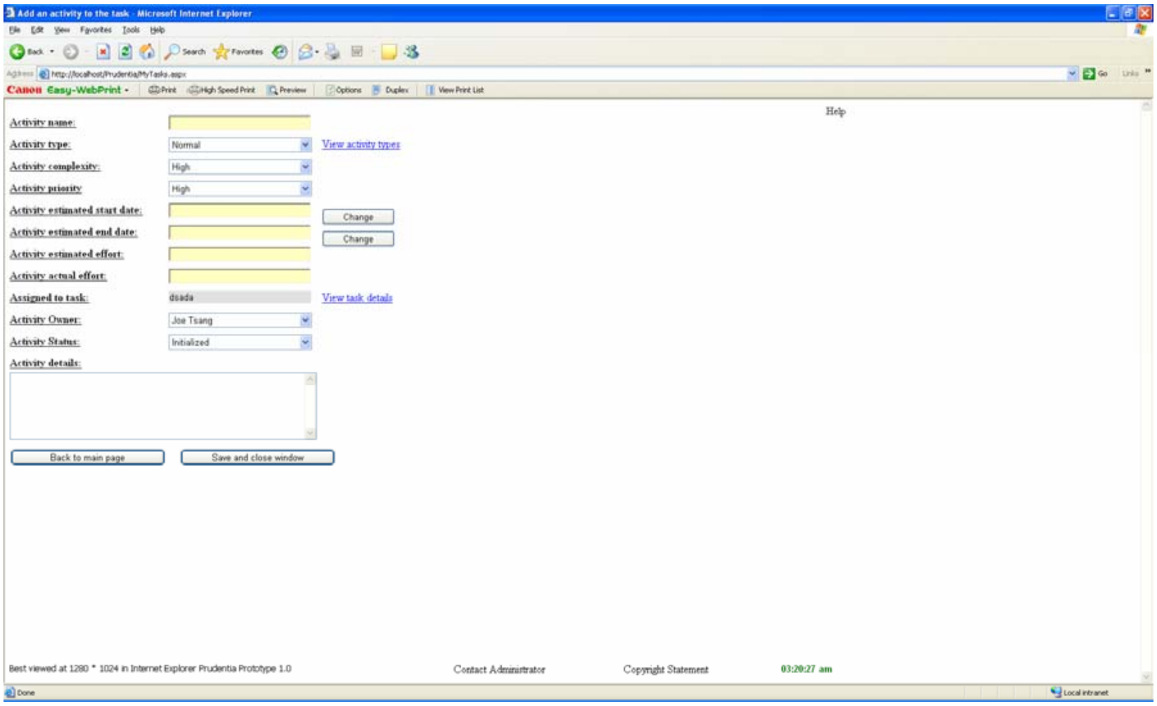
Project desired effort:

Project details:

Best viewed at 1280 * 1024 in Internet Explorer Prudentia Prototype 1.0

03:19:35 am

Figure 17. Add Project Screen

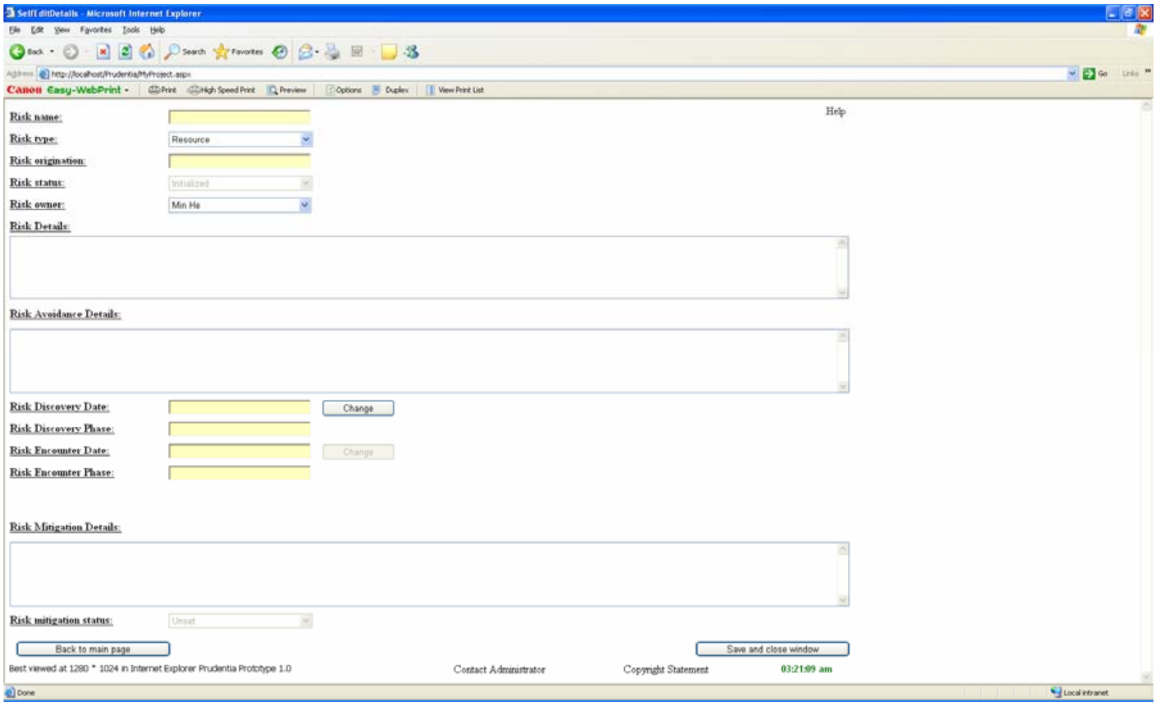


Add an activity to the task - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Prudentia/MyTasks.aspx
Go Links
Canon Easy-WebPrint Print High Speed Print Preview Options Duplex View Print List
Help
Change
Change
Back to main page
Save and close window
Done
Local intranet

Activity name:		
Activity type:	Normal	View activity types
Activity complexity:	High	
Activity priority	High	
Activity estimated start date:		
Activity estimated end date:		
Activity estimated effort:		
Activity actual effort:		
Assigned to task:	dsada	View task details
Activity Owner:	Joe Tsang	
Activity Status:	Initialized	
Activity details:		

Best viewed at 1280 * 1024 in Internet Explorer Prudentia Prototype 1.0 Contact Administrator Copyright Statement 03:20:27 am

Figure 18. Add Activity Screen



SelfEditDetails - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Prudentia/MyProject.aspx
Go Links
Canon Easy-WebPrint Print High Speed Print Preview Options Duplex View Print List
Help
Change
Change
Back to main page
Save and close window
Contact Administrator
Copyright Statement
Done
Local intranet

Risk name:

Risk type: Resource

Risk origination:

Risk status: Initialized

Risk owner: Min He

Risk Details:

Risk Avoidance Details:

Risk Discovery Date:

Risk Discovery Phase:

Risk Encounter Date:

Risk Encounter Phase:

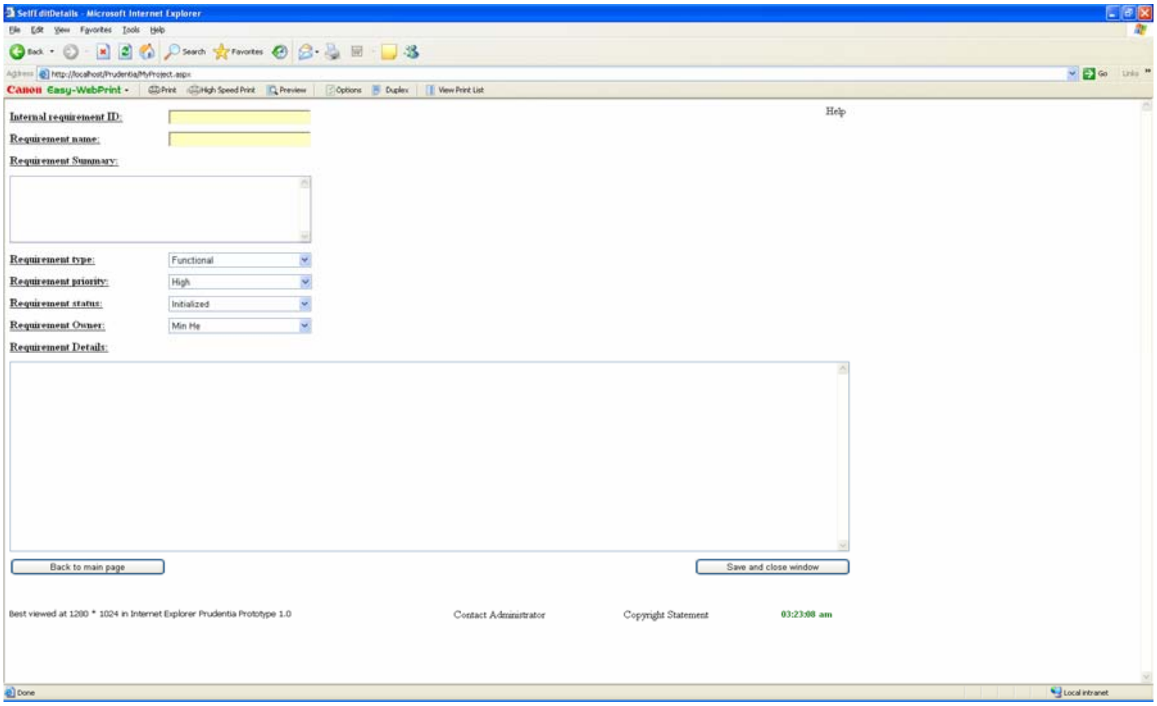
Risk Mitigation Details:

Risk mitigation status: Unset

Best viewed at 1280 * 1024 in Internet Explorer Prudentia Prototype 1.0

03:21:09 am

Figure 19. Add Risk Screen


SelfEditDetails - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Prudentia/MyProject.aspx
Go Links
Canon Easy-WebPrint Print High Speed Print Preview Options Duplex View Print List
Help
Back to main page
Save and close window
Contact Administrator
Copyright Statement
Done
Local intranet

Internal requirement ID:

Requirement name:

Requirement Summary:

Requirement type: Functional

Requirement priority: High

Requirement status: Initialized

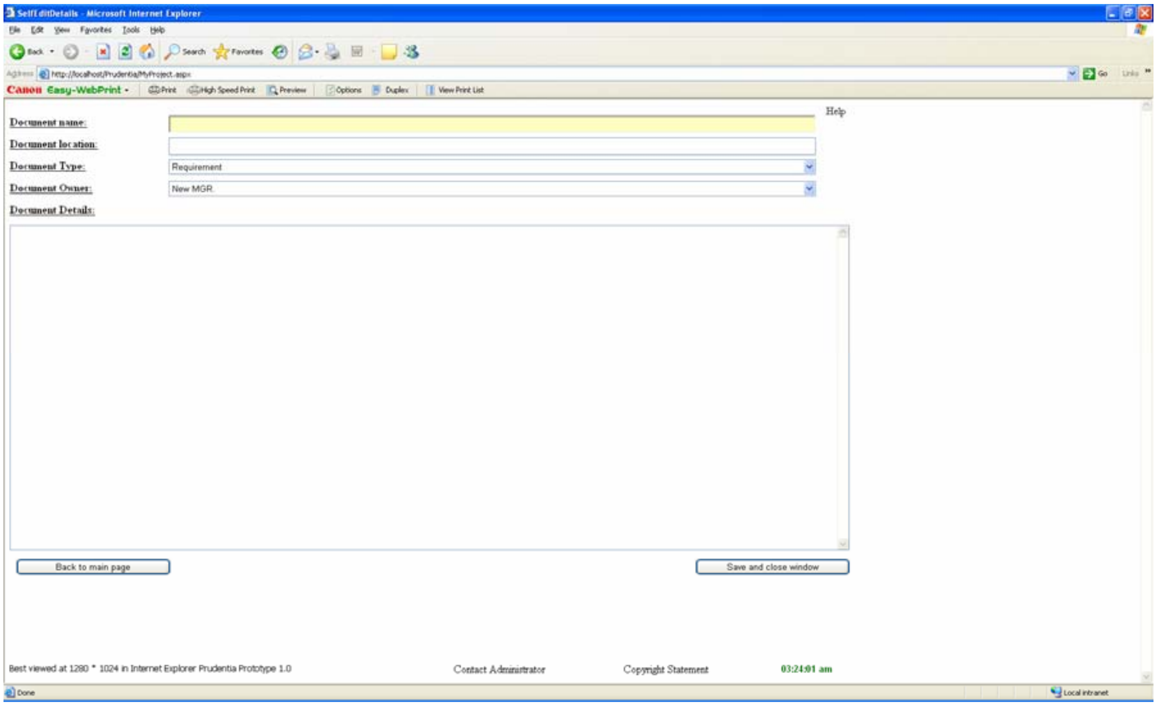
Requirement Owner: Min He

Requirement Details:

Best viewed at 1280 * 1024 in Internet Explorer Prudentia Prototype 1.0

03:23:08 am

Figure 20. Add Requirement Screen


SelfEditDetails - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Prudentia/MyProject.aspx
Go Links
Canon Easy-WebPrint Print High Speed Print Preview Options Duplex View Print List
Help
Back to main page
Save and close window
Contact Administrator
Copyright Statement
03:24:01 am
Done
Local intranet

Document name:

Document location:

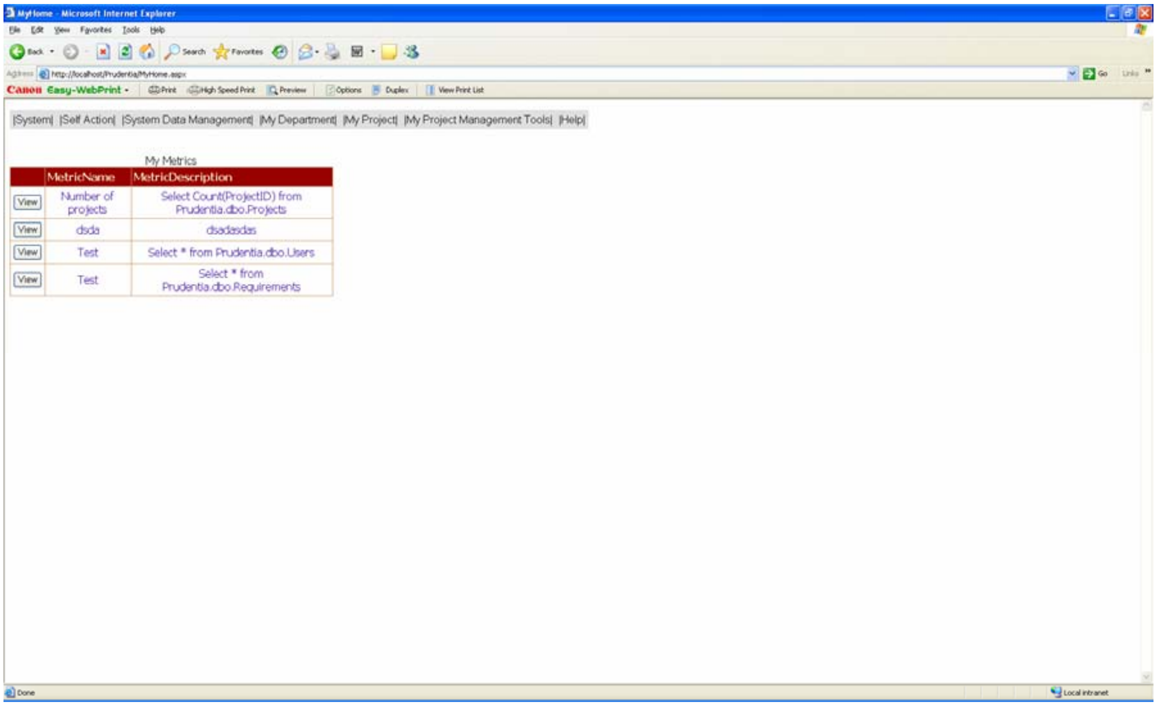
Document Type: Requirement

Document Owner: New MGR

Document Details:

Best viewed at 1280 * 1024 in Internet Explorer Prudentia Prototype 1.0

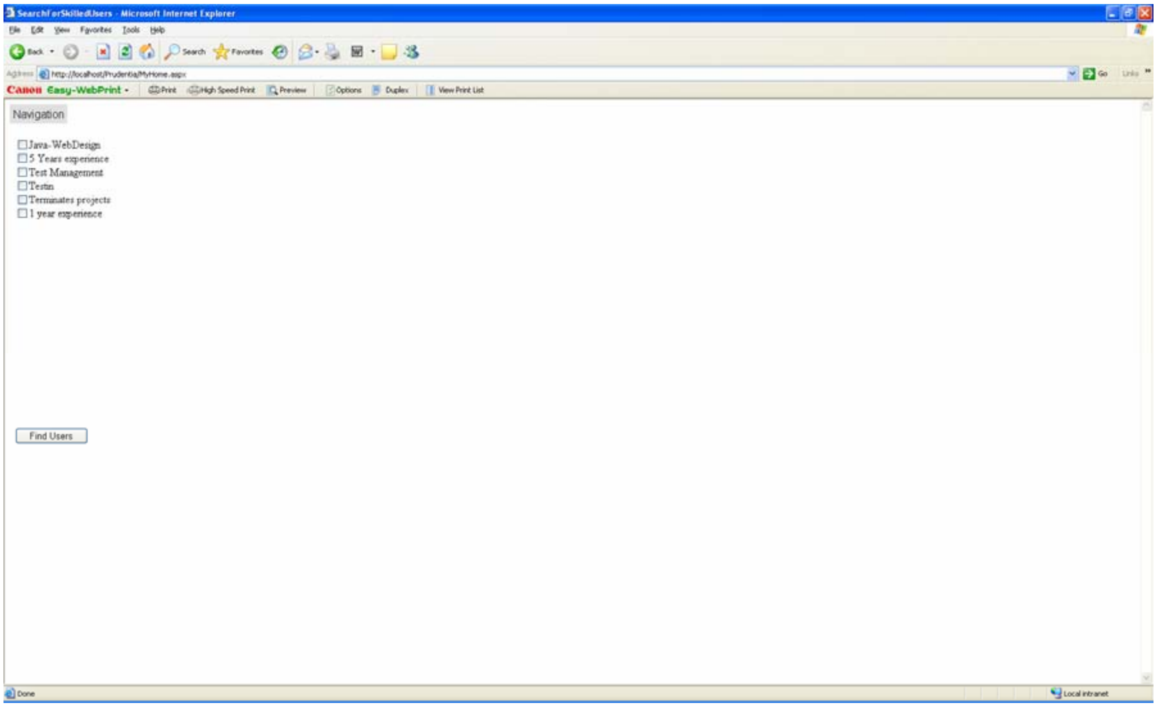
Figure 21. Add Document Screen


MyHome - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Prudentia/MyHome.aspx
Go Links
Canon Easy-WebPrint Print High Speed Print Preview Options Duplex View Print List
|System| |Self Action| |System Data Management| |My Department| |My Project| |My Project Management Tools| |Help|
Done
Local intranet

My Metrics

	MetricName	MetricDescription
View	Number of projects	Select Count(ProjectID) from Prudentia.dbo.Projects
View	dsda	dsadasdas
View	Test	Select * from Prudentia.dbo.Users
View	Test	Select * from Prudentia.dbo.Requirements

Figure 22. Metrics Screen


SearchForSkilledUsers - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Prudentia/MyHome.aspx
Go Links
Canon Easy-WebPrint Print High Speed Print Preview Options Duplex View Print List
Navigation
Find Users
Done
Local intranet

□ Java-WebDesign
□ 5 Years experience
□ Test Management
□ Testin
□ Terminates projects
□ 1 year experience

Figure 23. Search For Skilled Users Screen

APPENDIX B: Comparison of Project Manager 2007 to A Project Manager’s Assistant

A Project Manager’s Assistant is an application that was developed by a fellow MSE graduate student, Ben Garbers. This application is a client server based JAVA application, built on top of an existing framework (ABLE). These two applications were built separately from each other however share some similarities such as project management, skills management and resource allocation. However there are some differences in the way they implement project management. Project Manager 2007 is more applicable for large organizations where management is further away from actual development teams. A Project Manager’s Assistant is more applicable for direct managers to use on their teams for smaller, non-dependant projects. Project Manager 2007 relies on data collected for automatic project progress making while the A Project Manager’s Assistant application needs the project manager to be on top of the development process for project progress. Project Manager 2007 employs a Process Management framework for future expandability while A Project Manager’s Assistant makes use of other applications that was developed during the MSE program.

Overall, both applications have their uses and complement each other. In a perfect role, they would work in tandem. Project Manager 2007 tracks the larger project, the process management components and the dependency information. A Project Managers Assistant controls the task and activity management and cost estimation components.