ON-LINE PROJECT MANAGEMENT SYSTEM

by

Qian Sha
Bachelor of Economics, Capital University of Economics and Business, 1996

A Project

Submitted to the Graduate Faculty

of the

University of North Dakota

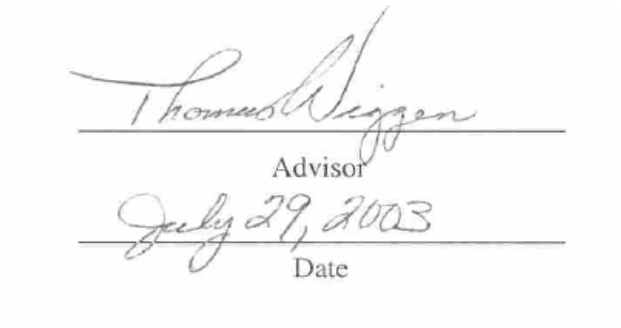
in partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota
August
2003

This report, submitted by Qian Sha in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota, has been read by the Faculty Advisor under whom the work has been done and is hereby approved.

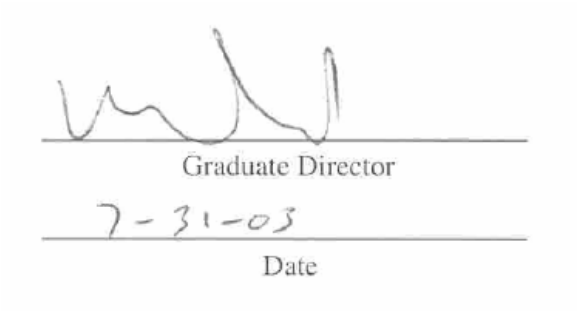


Advisor

July 29, 2003

Date

This project document meets the standards for appearance, conforms to the style and format requirements of the Computer Science Department of the University of North Dakota, and is hereby approved.



Graduate Director

7-31-03

Date

PERMISSION

Title: ON LINE PROJECT MANAGEMENT SYSTEM

Department: Department of Computer Science

Degree: Master of Science

In presenting this report in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that Department of Computer Science shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my work or, in his absence, by the Chairperson of the Department.

Signature Qian Sha

Date 07/31/03

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The editor express sincere appreciation to Tom Wiggen, my advisor, who helped me so much for my final report, and Xiaodong Zhang, my supervisor in Upper Midwest Aerospace Consortium, who provided the project requirements and gave me suggestions about the design of the database and user interface.

ABSTRACT

The project is developed for the on-line management of projects in UMAC (Upper Midwest Aerospace Consortium). It is a web based development, implemented on Redhat Linux platform. The web server is Apache-2; the database server is MySQL; the script languages I used are PHP and JavaScript. Those are all open sources and you can get them for free from internet.

The final product has a nice interface, a well-designed database. It is easy to use and maintain. Right now people in UMAC (Upper Midwest Aerospace Consortium) are using it.

I did the design, implementation and test all by myself. This report records all the things I did for the project in a software engineering way.

CHAPTER I

Introduction

1.1 Problem Statement

In UMAC (Upper Midwest Aerospace Consortium), there are lots of projects going on every year. People in UMAC hire lots of student employees to help them do the projects. How to manage all the projects efficiently on line is a problem for the project leaders and their employees. Also an employee might be involved in several different projects, he/she might have different roles in different projects. For example, an employee might be a team leader for one project; while at the same time, he/she might be a team member for another project. Right now they do not have software to manage their projects yet.

By using an on-line project management system, without having a meeting, the employee could know the tasks assigned to them by their project leaders; the project leader could know the progress of the current project; and the supervisor will know how many projects are going on concurrently and who is responsible for each of the project.

Also, sometimes people need to modify their projects. For example, the goal of a project can be changed; the duration and extension of a project can be changed too. The supervisor can even change the project leader, although it does not happen quite often. All of these could be done by using project management software. The design and implementation of such a project management system is the goal of this project.

All the information of current employees will be stored in a database. Without managing the employee, we cannot manage projects. Old employee leave and new employee come. Different people have different roles. All together in UMAC (Upper Midwest Aerospace Consortium), there are about 40 people, including the student employees.

1.2 System Functions Overview

The whole project management system can be partitioned into several sub-systems. For each of the subsystem, it has several functions.

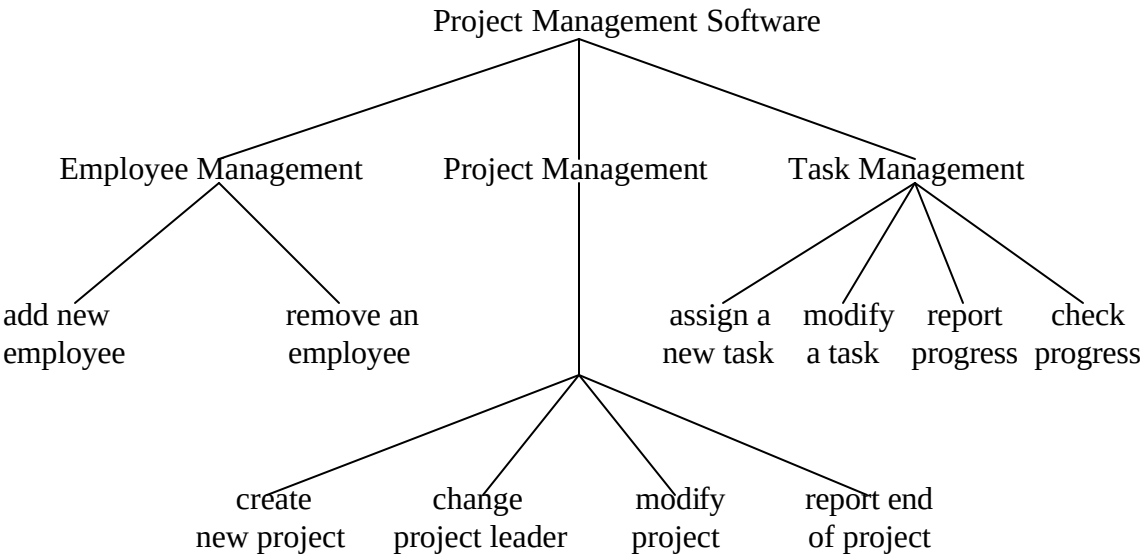


Figure 1. Vertical Partition of Function

CHAPTER II

Requirements and Specifications

Requirements were provided by Xiaodong Zhang, research associate and system administrator in UMAC (Upper Midwest Aerospace Consortium).

2.1 Types of Users

There will be three different kinds of users for the on-line project management system.

The supervisor is the only person can manage his employees. He/she can add a new employee into the database; remove an employee from the database. He/she also is the only person that can create a project and assign it to a project team leader. To change project leader, project duration and project extension is another special privilege for the supervisor. Finally when a project is done, the supervisor could prove it and report the end of the project. The supervisor is not responsible to one specific project; he has to know how many projects are currently going on and how many projects are completed in the whole organization.

There is only one supervisor in UMAC (Upper Midwest Aerospace Consortium), the chairperson of the organization.

A project team leader is the person who will be responsible for one or more on-going projects, send project report to the supervisor, and assign specific tasks under a certain project to other employees. Of course he/she needs to know all the on-going projects under his/her control and the progress of each task under a specific project, once

the date passes the due date, he/she should see some kind of alert. A project team leader can also change the content and due day of a task and browse those completed projects that he/she used to be the leader for. Once there are no on-going projects he/she is the leader for, then he/she is not a project leader anymore.

A project team member is the person who needs to check the tasks assigned by the project team leader and reports his/her progress of the task he/she is working on.

An employee's privileges totally depend on his or her status in the database. Except the supervisor, other employees might have two statuses. He/she can be a project team leader, at the same time; he/she can be a team member of another project.

Once a user logs into the system, the system will check the database about the employee's status, then his or her privileges are decided and shown in the interface. Once a user logs out the system, his or her cookie for the session should be killed.

2.2 Documentation Requirements

For the documentation purpose, important date should be time-stamped by calling system function. For instance, when a task is assigned, the assign date will be automatically stored into the database.

2.3 Interface Requirements

Generally speaking, the user interface should be simple and easy to use.

CHAPTER III

Software Design

3.1 High Level Design

A state transition diagram shows the state space of a given context, the events that cause a transition from one state to another, and the actions that result. It is widely used in the design phase of software engineering process.

I began the software design description with a state transition diagram (see figure 2). This diagram displays all the possible states of the system as rounded rectangles. An arrow connects one state (state A) to another (state B) if a transition from state A to state B is possible. Each arrow is labeled in some way to indicate the particular event(s) that can trigger this state transition. In the on-line project management system, I have identified four states.

Because for each of different users, he/she can only log in or log out of the system by using his/her account in UMAC (Upper Midwest Aerospace Consortium). Once logged in, he/she can stay in or log out the system. Three different types of users, so we have three logged in states. If log in failed, a user will stay in the not logged in state until he/she passes the account verification.

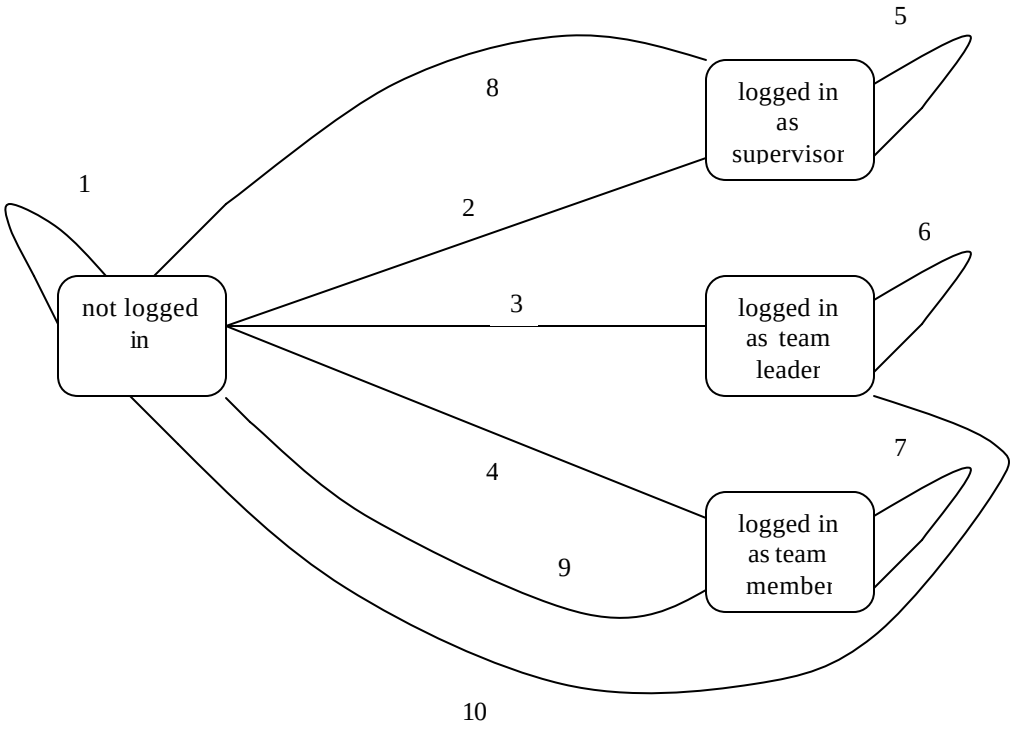


Figure 2. State Transition Diagram

Note: 1. user name and password does not match, stay in the not logged in state.

2. user name and password matches, go to the logged in as supervisor state.

3. user name and password matches, go to the logged in as team leader state.

4. user name and password matches, go to the logged in as team member state.

5. stay in the logged in as supervisor state.

6. stay in the logged in as team leader state.

7. stay in the logged in as team member state.

8. log out from the logged in as supervisor state.

9. log out from the logged in as team leader state.

10. log out from the logged in as team member state.

3.2 Database Design

The core of this project is the database design. Basically for management software, users only communicate with database through interface. Whether your database is designed reasonable and sufficient will have a direct effect on the quality of the product.

Conceptual modeling is an important phase in designing a successful database application. My design just followed the traditional approach of concentrating on the database structures and constraints. After talked to Xiaodong Zhang about the information that people in UMAC (Upper Midwest Aerospace Consortium) need to manage their projects, I came up with the Entity Relation Diagram (figure 3). All together there are three entities: employee, project and task, represented by rectangles. Each entity has its own attributes, represented by eclipses; the key attribute is represented by a bolded eclipse. Those entity attributes are indispensable information that is necessary to manage projects.

There are relations between entities, represented by diamonds. There are two kinds of relations in the ER diagram. One is one-to-many (1-n); the other is many-to-many (m-n).

Below is the diagram for all the entities and relations for the database of the on-line project management system. Because being the developer, I do not know for sure

how people in UMAC (Upper Midwest Aerospace Consortium) will manage their projects, I asked some suggestions from Xiaodong, and then make the final design.

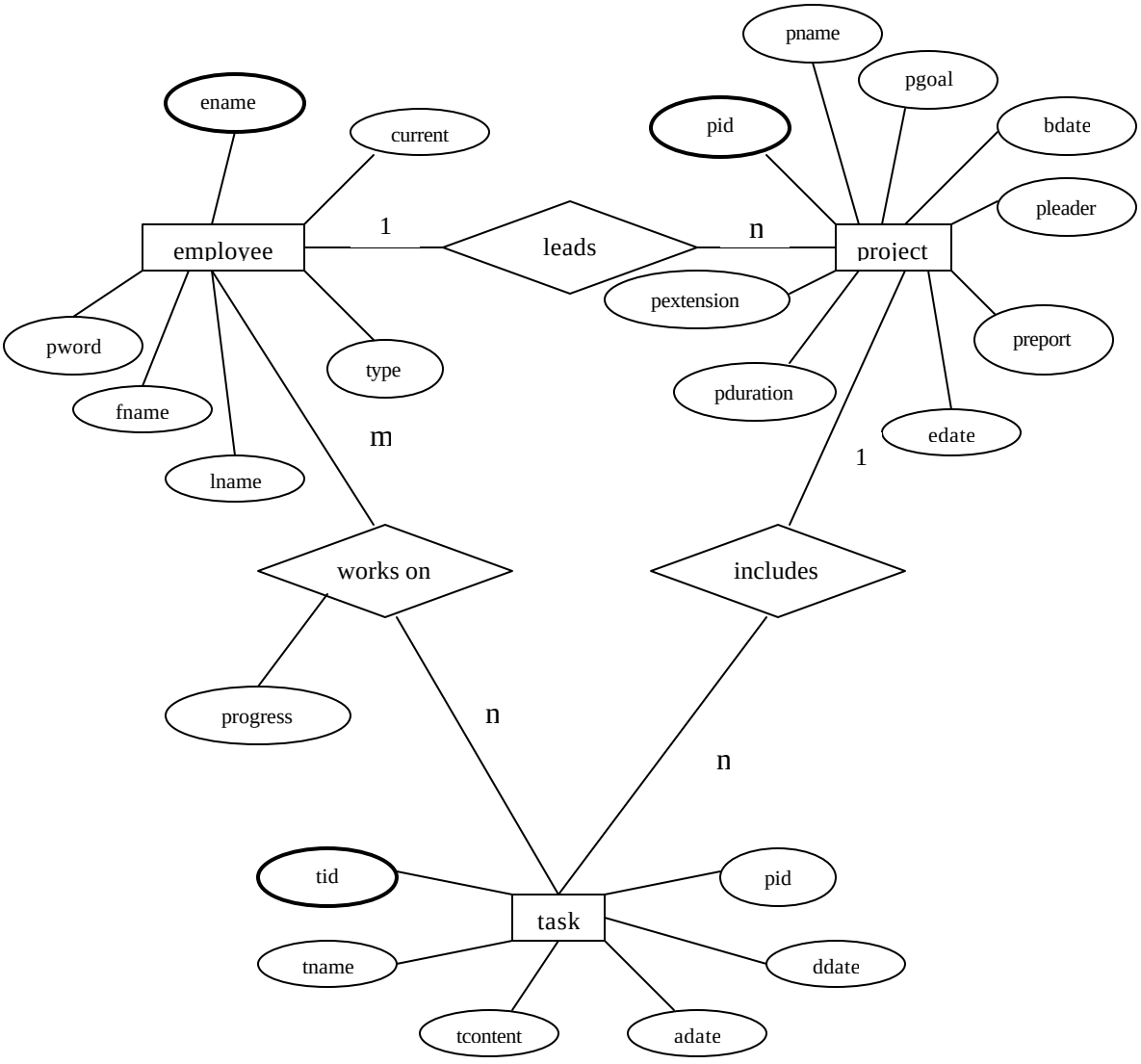


Figure 3. Entity Relation Diagram

Once the ER diagram has been drawn, that means the conceptual design of the database is finished and the conceptual schema is created. Then the next step was the actual implementation of the database, using MySQL. So the conceptual schema was translated from the high-level data model into the implementation data model. This step is called logical design, and its result is a database schema in the implementation data model of the DBMS (Database Management System). Normalization technique was used to reduce the redundancy of tables. The final database is made up of four tables. Three entities ended up to three tables, and one table is for the many-to-many relation. Those tables can hooked up together for complicated SQL (Structure Query Language) queries.

Table 1. Project Table

Field	Type	Null	Key	Default	Extra	Description
pid	int(5)		PRI	0	auto_incre	project id
pname	varchar(25)					project name
pgoal	text					project goal
bdate	date					beginning date
pleader	varchar(25)					project leader's ename
preport	text	yes		null		final report
edate	date	yes		null		ending date
pduration	varchar(25)					project duration
pextension	varchar(25)	yes		null		project extension

Table 2. Employee Table

Field	Type	Null	Key	Default	Extra	Description
ename	varchar(25)		PRI			last name + first name initial, user name
pword	varchar(25)					password
fname	varchar(20)					first name
lname	varchar(20)					last name
type	varchar(20)					single or multiple, represented by number
current	varchar(1)					whether the user is current employee

Table 3. Task Table

Field	Type	Null	Key	Default	Extra	Description
tid	int(5)		PRI	null	auto_incre	task id
tname	varchar(25)					task name
tcontent	text					specific task content
adate	date					assigned date
ddate	date					due date
pid	int(5)			0		the project ID which the task belongs to

Table 4. works_on table

Field	Type	Null	Key	Default	Extra	Description
tid	int(5)		PRI	0		task ID which the employee is working
ename	varchar(25)		PRI			employee's ename
tprogress	text	yes		null		task progress

There is something I would like to make clear. For employee table, ename is the primary key. It is made up of last name plus first name initial. This may not be unique, although the chance of having two employees have the same first name and last name is small. So when the supervisor tries to add an ename that is already existed in the database, by using PHP, I made a pop-up message to tell the supervisor that ename has already been used and suggest to change to another one. For project and task table, Id is the primary key. But there should be no duplicates on project name and task name, otherwise it will cause trouble. A user will see projects or tasks having same names from the interface, there is no way for him/her to figure out the difference. Because neither project id nor task id will show to users, ids do not make sense to users, only the names. For works_on table, the primary key is a combination of ename and tid.

In employee table, I set an attribute called current. This is the flag to show whether the employee has been removed from the database. Because an employee leaves, but his/her work is still in the database. We cannot just delete the employee's record from the database, it will violate the referential integrity constraint. The referential integrity constraint is specified between two relations and is used to maintain the consistency among tuples of the two relations. Informally, the referential integrity constraint states

that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation. So if we just simply delete an employee's record from employee table, some records in works_on table reference to the record will refer to nothing.

Also in employee table, I set an attribute called type. It is a varchar. Remember, employees may have more than one status in UMAC (Upper Midwest Aerospace Consortium). Employee's status is represented as "type" in database. Each of the type is represented by a number. "1" represents supervisor, "2" represents project team leader, "3" represents project team member. If an employee has more than one status, then those statuses are stored in the database as numbers separated by comma.

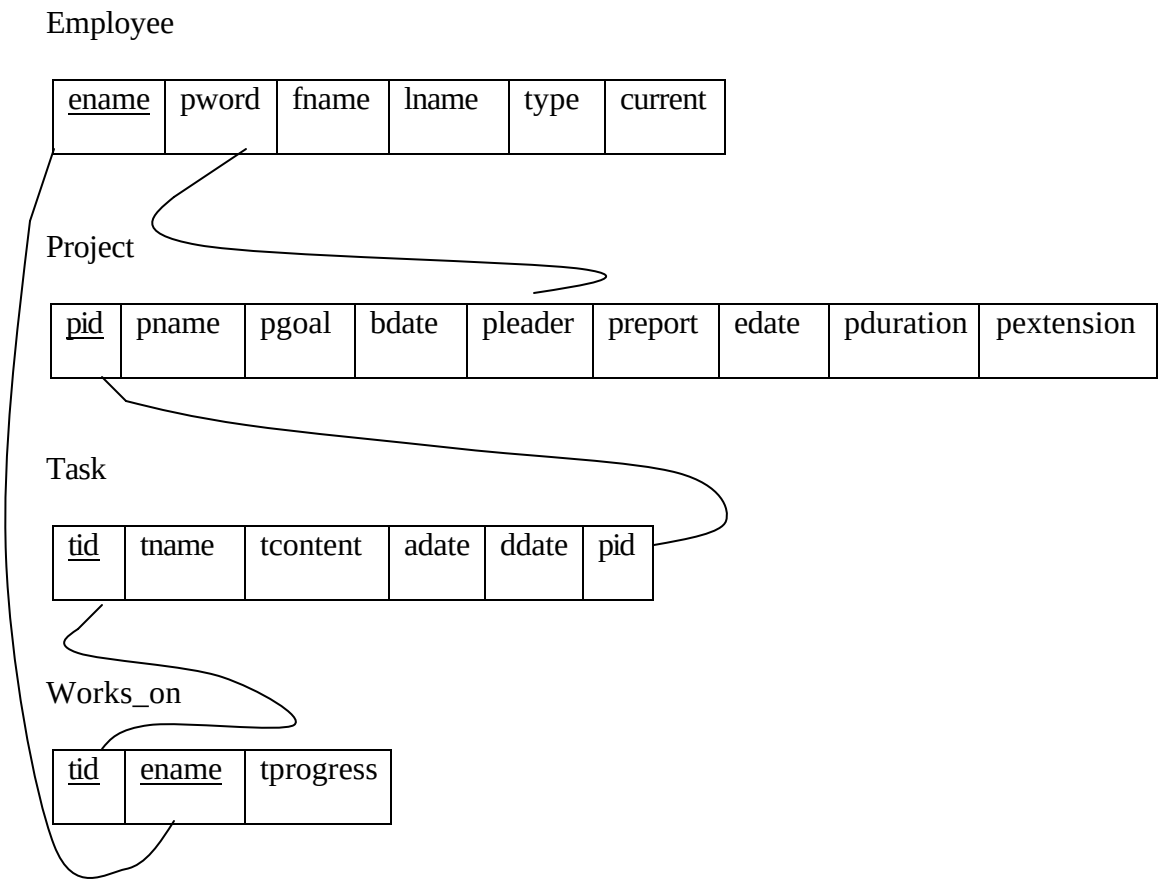


Figure 4. Referential Integrity Constraints on Relational Database Schema Diagram

Figure 4 describes the relations between the four tables in the database and how they are connected.

3.3 Low Level Design

After I finished the database design, the next step was the architectural design. The method I chose was to draw the DFD (Data Flow Diagram).

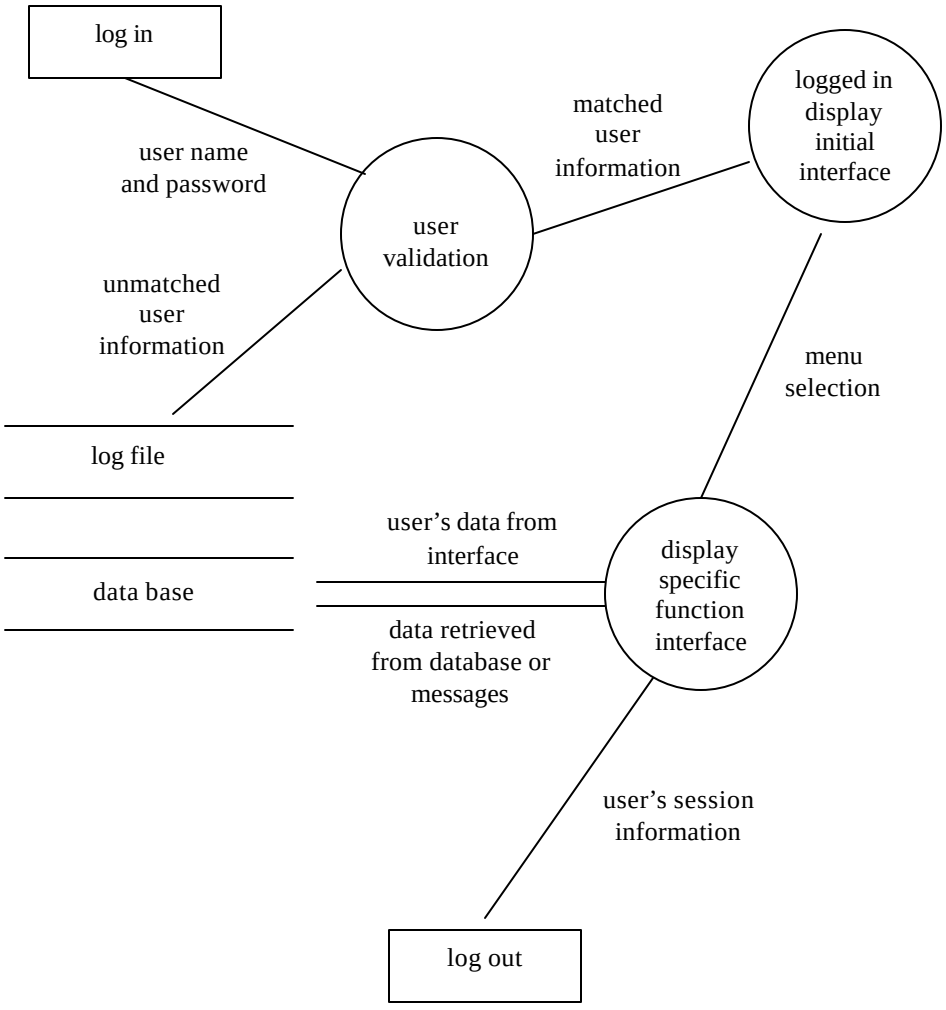


Figure 5. Level 1 DFD

Figure 5 is the data flow diagram for all the employees, it does not matter whether the user is a supervisor, team leader or team member.

Following figures are the data flow diagrams for different users.

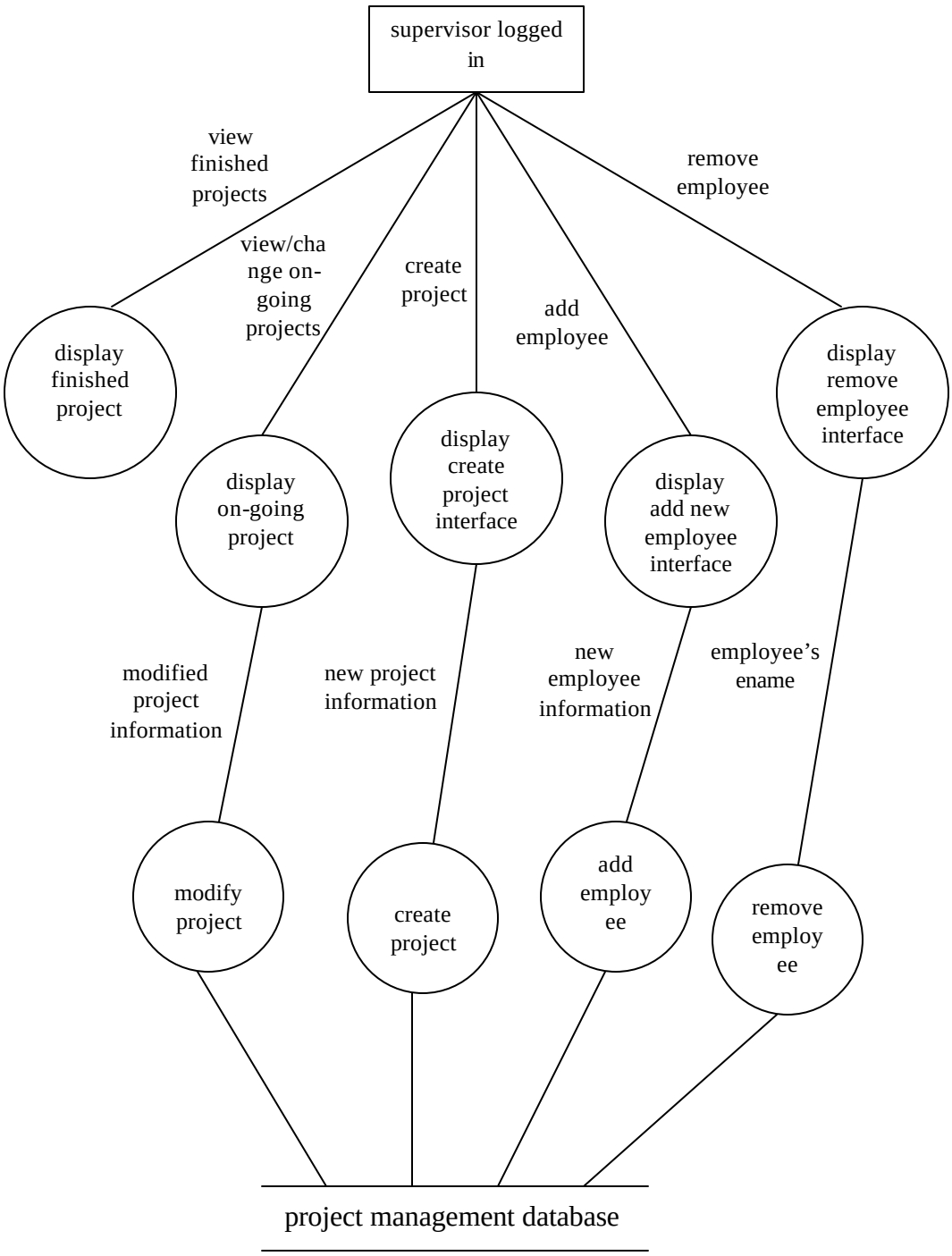


Figure 6. Level 2 DFD for Supervisor

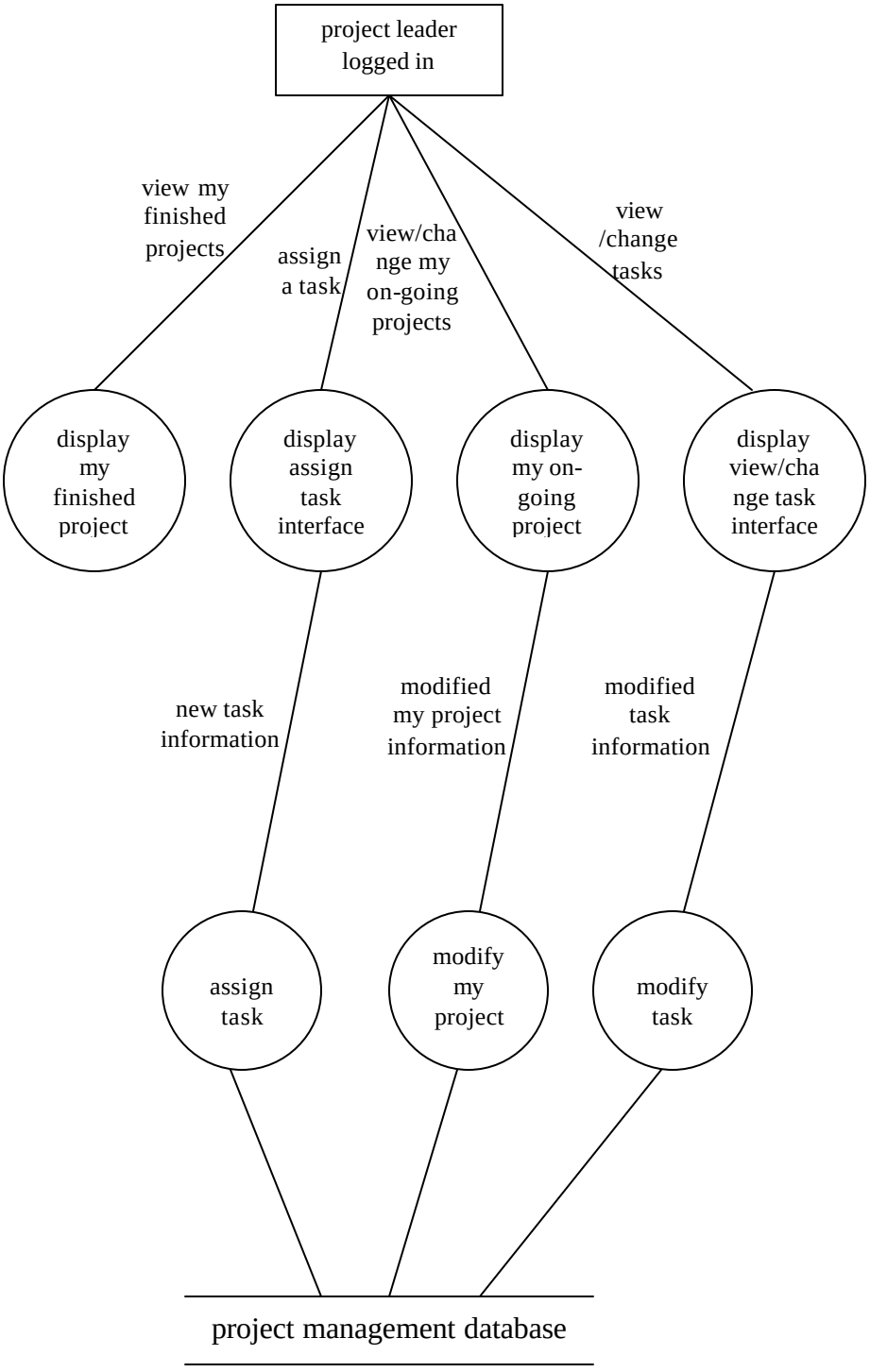


Figure 7. Level 2 DFD for Team Leader

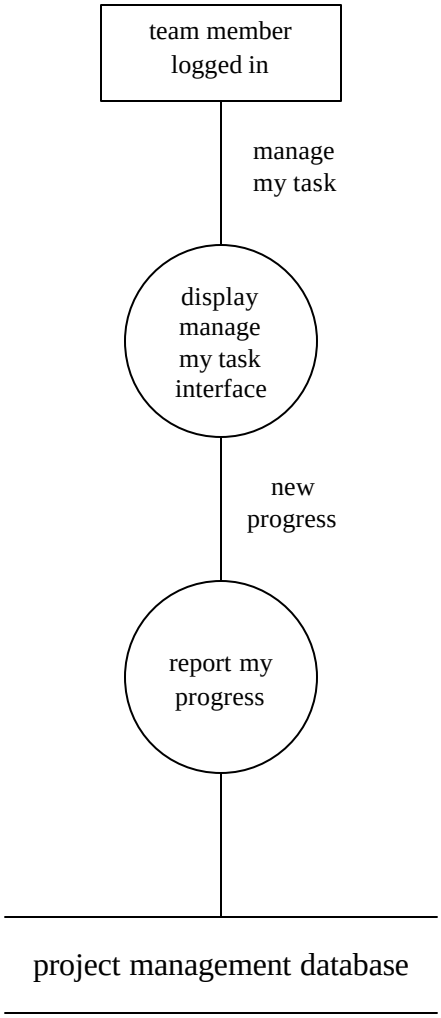


Figure 8. Level 2 DFD for Team Member

By following this design, every bubble matched to a function or process. It is easy to maintain if a project is designed in this way. Actually in my code, I wrote all the functions in a .INC file. In the main PHP program, I just include the .INC file, and then I could call all the functions I defined there and by passing them corresponding parameters. This makes your program easy to understand and avoid code duplicates.

CHAPTER IV

Software Validation & Verification

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. For best testing result, there should be a third group responsible for testing. But this is not a big project, so Xiaodong Zhang and I were the two testers.

4.1 Development Process Model

Testing process is always bundled with developing process. For the on-line project management system, prototype model was used during the development process. It begins with requirements gathering. Before I started, I talked with Xiaodong about the overall requirements. Then I started a "quick design". The quick design focuses on a representation of those aspects of the software that will be visible to the user, it leads to the construction of a prototype. Ideally, the prototype serves as a mechanism for identifying software requirements. At the beginning, neither of us knew what the product should be like. After I built several versions prototypes, we finally decided what kind of interface we want. After each version was built, Xiaodong and I tested it, and then he gave me more suggestions and requirements to improve it.

Prototyping paradigm is widely used. The advantage of this paradigm is that user get a feel for the actual system and developers get to build something immediately.

4.2 Testing Plan

White-box testing and black-box testing were used. White-box testing is predicated on close examination of procedural detail. Black-box testing is used to demonstrate that software functions are operational. For the white-box testing, I used basis path testing. From the flow chart, I figured all the paths.

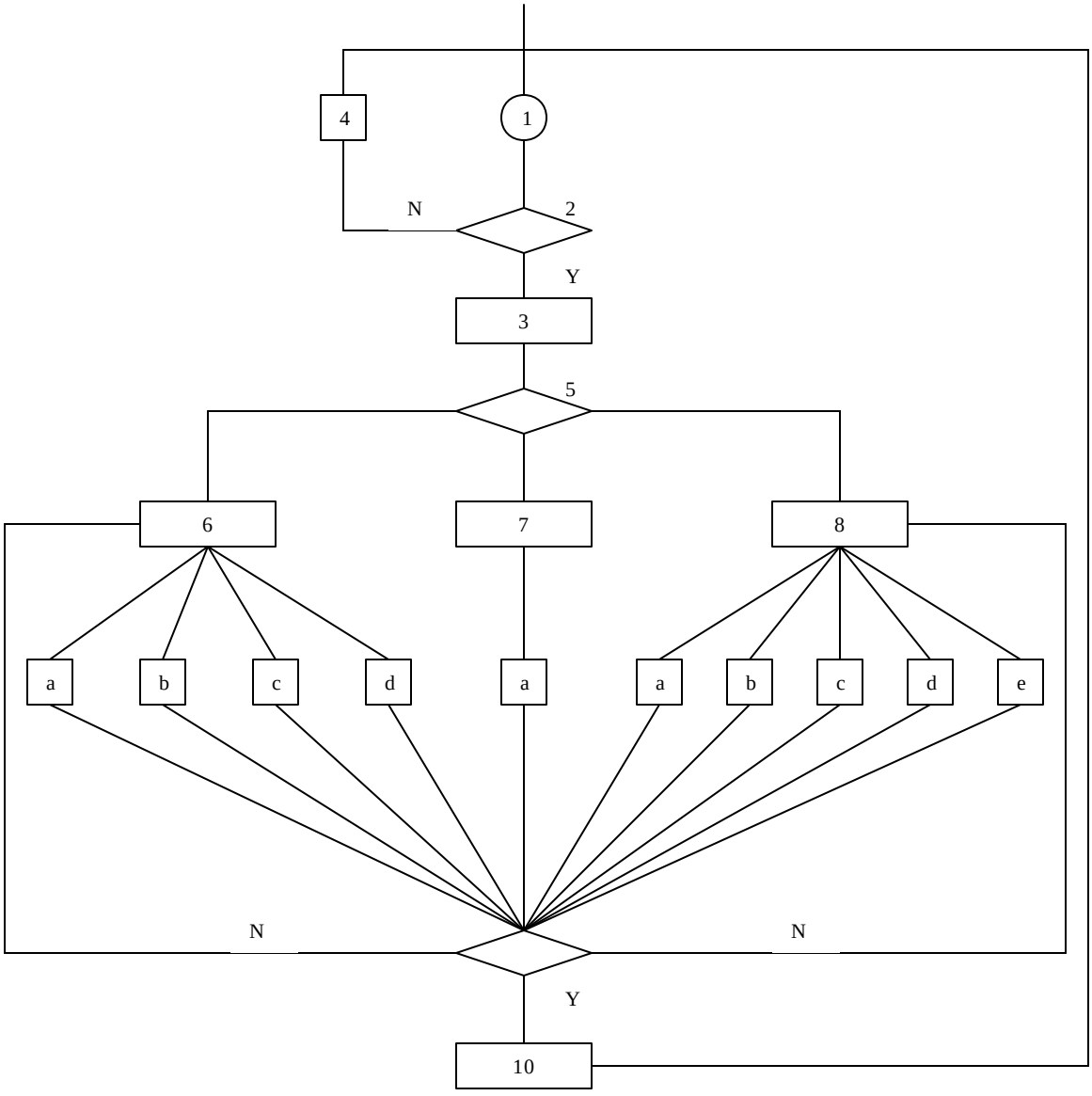


Figure 9. Flow Chart

Notes:

1: user log in

2: if password and user name matches

then 3: get the user's status from database

else 4: display error message

go back to the log in page

end if

5: case user's status

team leader 6: display corresponding interface

team member 7: display corresponding interface

supervisor 8: display corresponding interface

end case

6a: view my finished projects

6b: view my on-going projects

6c: assign task under a project

6d: view tasks under a project

7a: manage my current tasks

8a: view all finished projects

8b: view all on-going projects

8c: create a new project

8d: add a new employee

8e: remove an employee

9: log out

10: kill session cookies and redirect to initial log in page

When I did the white-box testing, first I just input some test data into the database through MySQL API, and then I just followed each path to make sure by using all my test cases. It turned out to be fine.

I did the black-box testing through the interface. I input some sample data into the database before I started testing. After I input more data through the interface, I checked them through MySQL API to see whether they have been stored in the right tables. To make sure the output from the software is correct, I checked the output by using SQL (structured query language) from the database. If the results match, that means the software is working properly. People do not need to worry about the software structure, to the tester, the whole software is like a "black box".

4.2 Testing Strategies

Not all the testing works are done at last minute. Some of them are parallel with your coding process. I tested each function separately. Each function in the software is like a module, which is the smallest unit of the software. Basically, all the functions that user can use are linked to the interface. Once the link has been clicked, the corresponding function will be called. The interface is acting as the driver for the unit testing, and there is no stub needed in this case.

Because the units in this project are not based on each other, so no integration testing was applied.

After I finished my own testing for each version, Xiaodong tested the software as a user. He used alpha test, which means I was recording the usage problems and errors

when he did the testing. Then I corrected the errors and solved the usage problems. Then we did the same process again until the final version.

The process can be described by using the figure below:

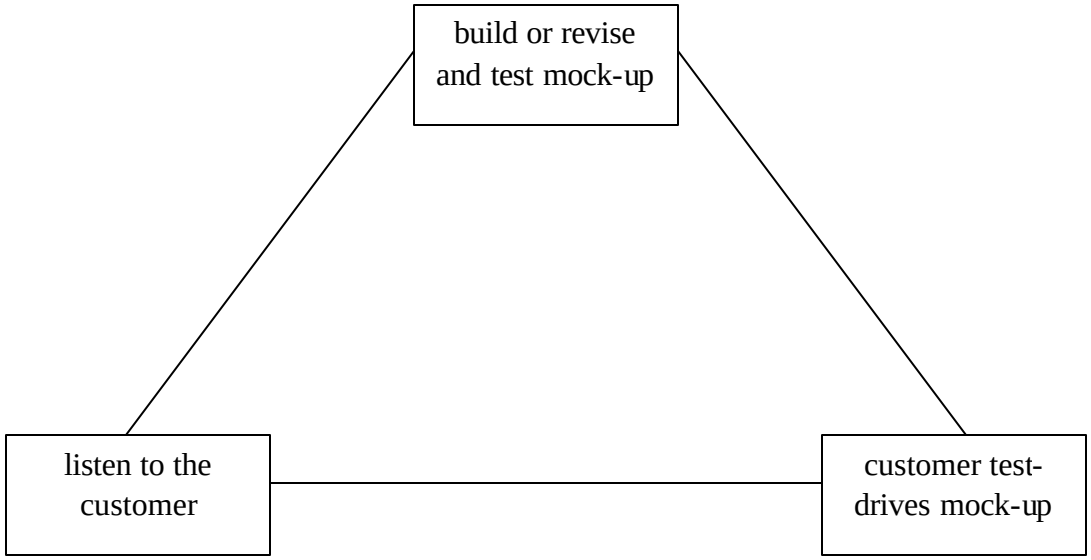


Figure 10. Developing and Testing Process

4.3 Security Testing and Stress Testing

The security issue is mainly focused on the database access privilege. The MySQL server is located on the UMAC (Upper Midwest Aerospace Consortium) server. To access it, you must pass the user account and password verification. I wrote a function for connecting to the database. From the user interface, once the user passes the password and user account verification, then he/she can access the database according to his or her privileges. But a user must log out after he/she is done, otherwise, there might be a chance that other people break into the system and mess up the data in the database. The log out function works fine; it kills all the user's cookies for his/her session.

Stress testing was used too. Stress tests are designed to confront programs with abnormal situations. When I tested my final version, I used a user with all statuses. That means the user is a supervisor, a project team leader and a project team member. By doing that, I could test all the system functions. But in the real world, this is not the case.

When I tested the robustness of the product, I tried to violate the size limit definition of the attribute in the table of the project management database. For example, the definition of the size of ename attribute from employee table is varchar(25), I input an ename with a length of 30. But no error messages generated. MySQL just truncated it to fit. This may cause trouble. Because the user does not know the length limit, he/she will think the input has been accepted. But when later he/she uses the ename to log in, the system will not allow him/her to log in. My solution is to limit the input size from the interface. For instance, if the definition of fname attribute from employee table is varchar(20), then the size of the input text from the interface is limited to 20. So it will not cause trouble anymore. For all the textareas, the definition in the MySQL database is LONGTEXT, it can hold up to 4294970000 bytes. This should be enough even for a project final report.

CHAPTER V

Ideas

The team members in a group may need to communicate on-line. For this on-line project management system, we do not have this function. Because when Xiaodong told me the specific requirements, he did not mention that. Actually the project group in UMAC will have regular meetings; they can also communicate by sending emails, so they just ignored this function. But I still think if people can know other member's process in one project team, it will be good.

Another thing is if in the future when the MySQL database server is upgraded, probably it will support views. So I can create views in the database, then the query efficiency will be improved.

APPENDIX I

User Manual

1. System Configuration

The development is based on Linux Redhat 8.0, and the language and database I used are all open source. After the installation of Redhat Linux, you can download MySQL database server from www.mysql.com, the version we are using in UMAC (Upper Midwest Aerospace Consortium) right now is 3.23.56, and one of the drawbacks is it does not support view (virtue table). But the coming new version will support view. You also need to download PHP from www.php.net, and configure it with the support of MySQL. The Apache2 web server comes with Redhat 8.0, just choose the default system configuration when you install the operating system. One thing I would like to mention here, Redhat 8.0 has a version of PHP in its installation package. I am using newer version of PHP, so I did not choose the system option of PHP from the operating system.

The combination of MySQL and PHP with Apache probably is the most popular choice for web-based database developments.

2. Product Installation

The only thing you need to do is to copy all the files into the default www service directory. On Linux Redhat 8.0 it is /var/www/html. The product must be installed on a Linux Redhat 8.0 or higher server with MySQL server 3.23.56 or higher and Apache-2 server installed

3. Initialization

A supervisor will be the first person that uses the system. First, he/she should log in by using user name “root”, password “root”. This is the initial user created by the developer. Then the supervisor should add him/herself into the user list after logs in as root, and then log out root and log in as him/herself, and remove user “root” from the user list.

When the supervisor adds new user into the user list, the user name should be the employee’s last name plus first name initial. Just for simplicity, no upper case letter needed.

4. How to use

The user interface is friendly and easy to follow. First thing is to log in.

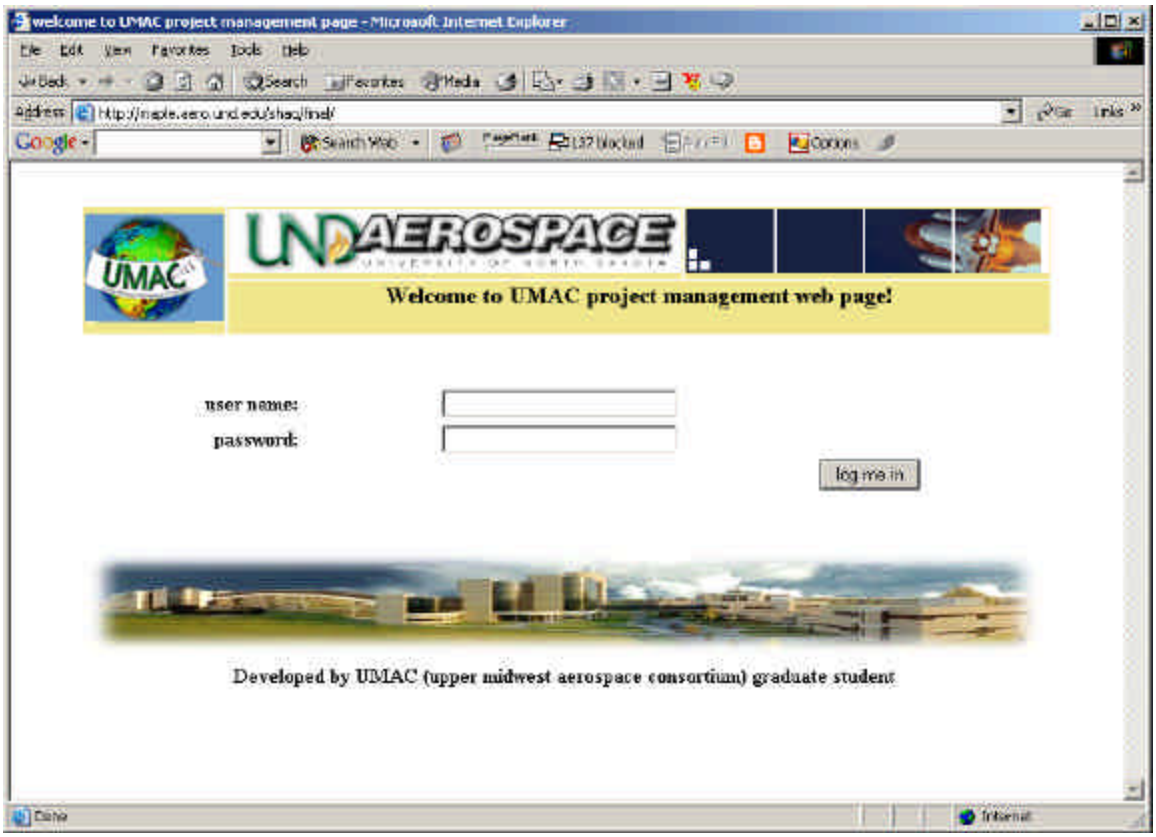


Figure 11. User Log in Page

If log in fails, the user has to go back and try again.

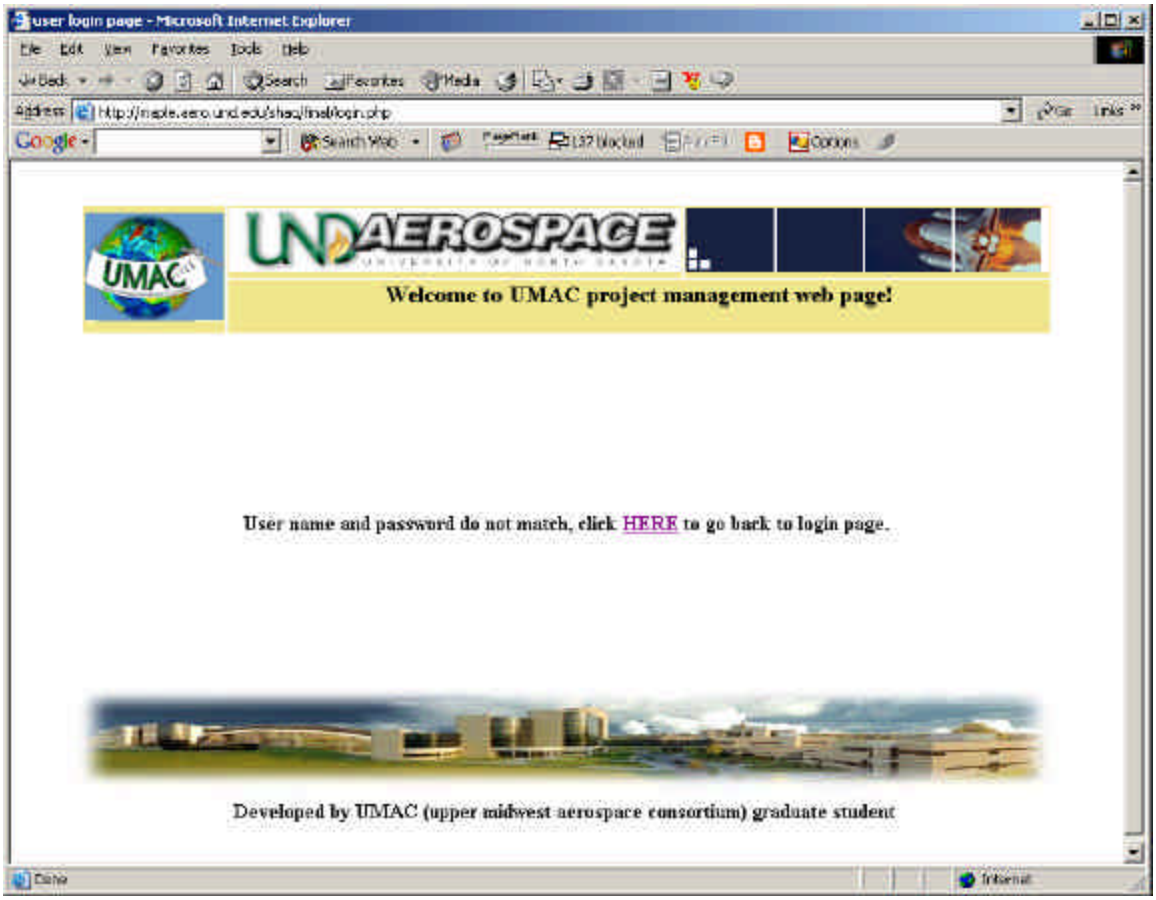


Figure 12. Failed Log in Page

4.1 Supervisor

When a supervisor logs in, the default interface will show current on-going progresses. He/she can change project's goal, duration, extension and project leader and submit the changes. The project report is submitted by the project leader, once the project is proved by him/her, he/she can check the checkbox down the right corner, then the project is finished, which means it is not a on-going project anymore and nothing can be changed. If the supervisor would like to see another on-going project, he/she can click the "select to see another project" drop down list, and then he/she will see other on-going

projects (if there is any). Then he/she can choose one by clicking the name; the whole page will be refreshed to show the project he/she has picked.

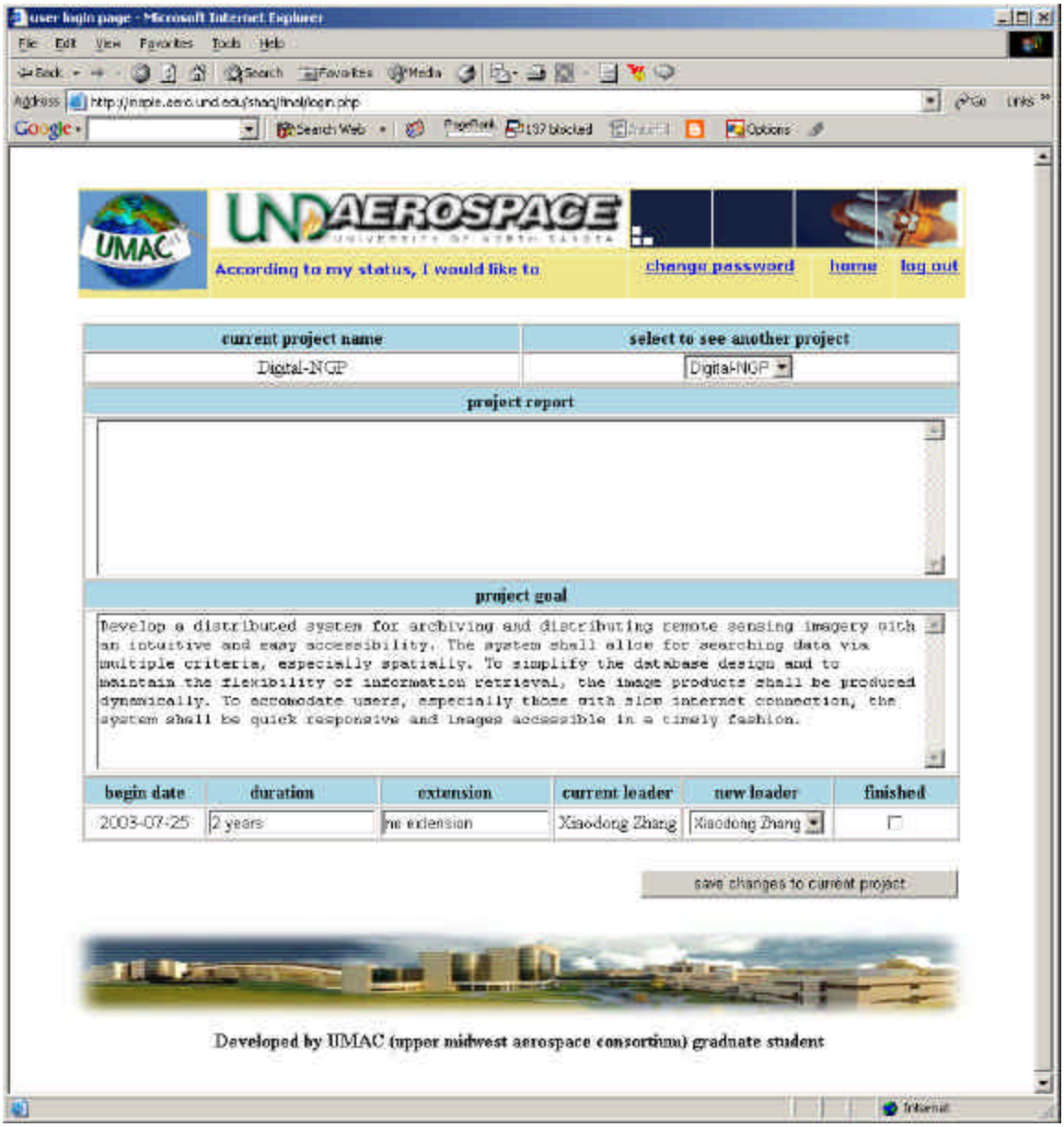


Figure 13. Supervisor's Default Initial Interface

The other options are available from the menu. Just put the mouse on the header "According to my status, I would like to", then options will show up according to the user's status. For the supervisor, there are four of them.

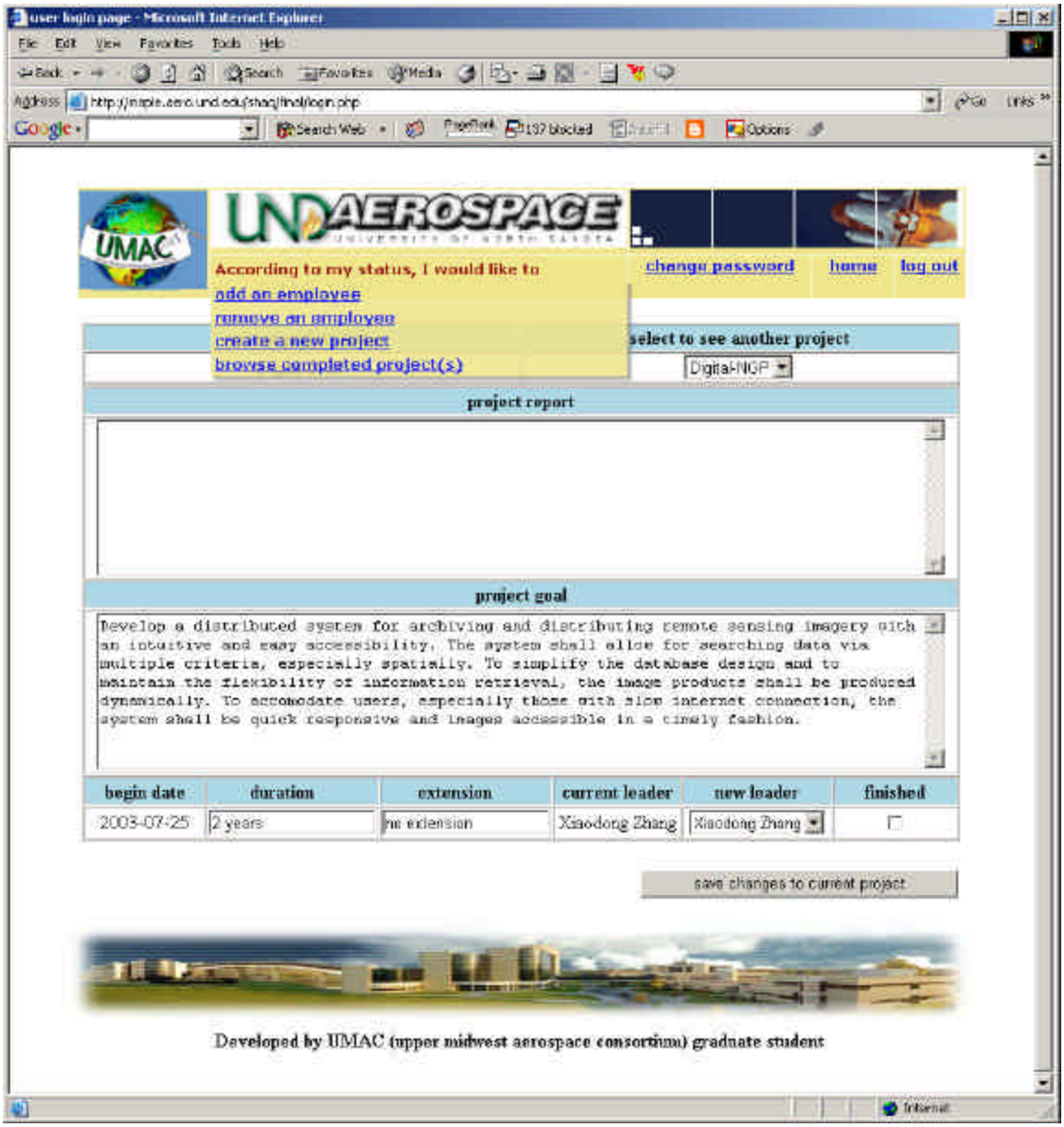


Figure 14. Supervisor's Other Options

The other options are "add an employee", "remove an employee", "create a new project" and "browse completed project(s)". A supervisor could choose one of those options from the menu, the interface for those four options are pretty easy to follow, here I would not bother to describe.

The three links on the header are the same for all the users despite of their status. They are used for changing password, returning to the initial page and logging out.

4.2 Project Team Leader

The initial interface for a team leader will show the on-going projects he is the leader for.

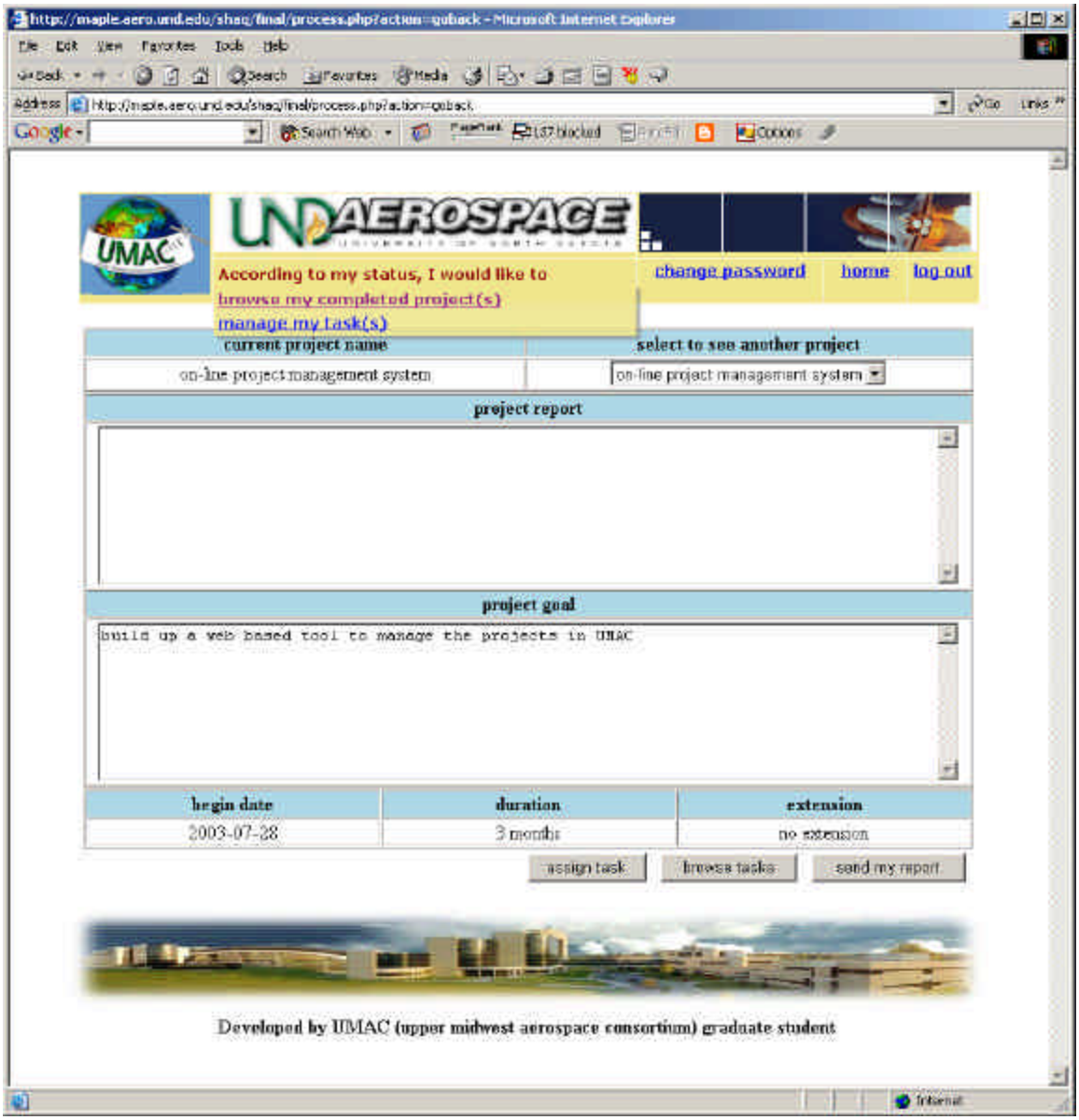


Figure 15. Team Leader's Initial Interface with Other Options

Compare to the supervisor's options from the menu, a team leader only get two options (the second option exists if the team leader is a team member of some other project groups concurrently). If he/she wants to see other project under his direction, just click the drop down list on the top right corner.

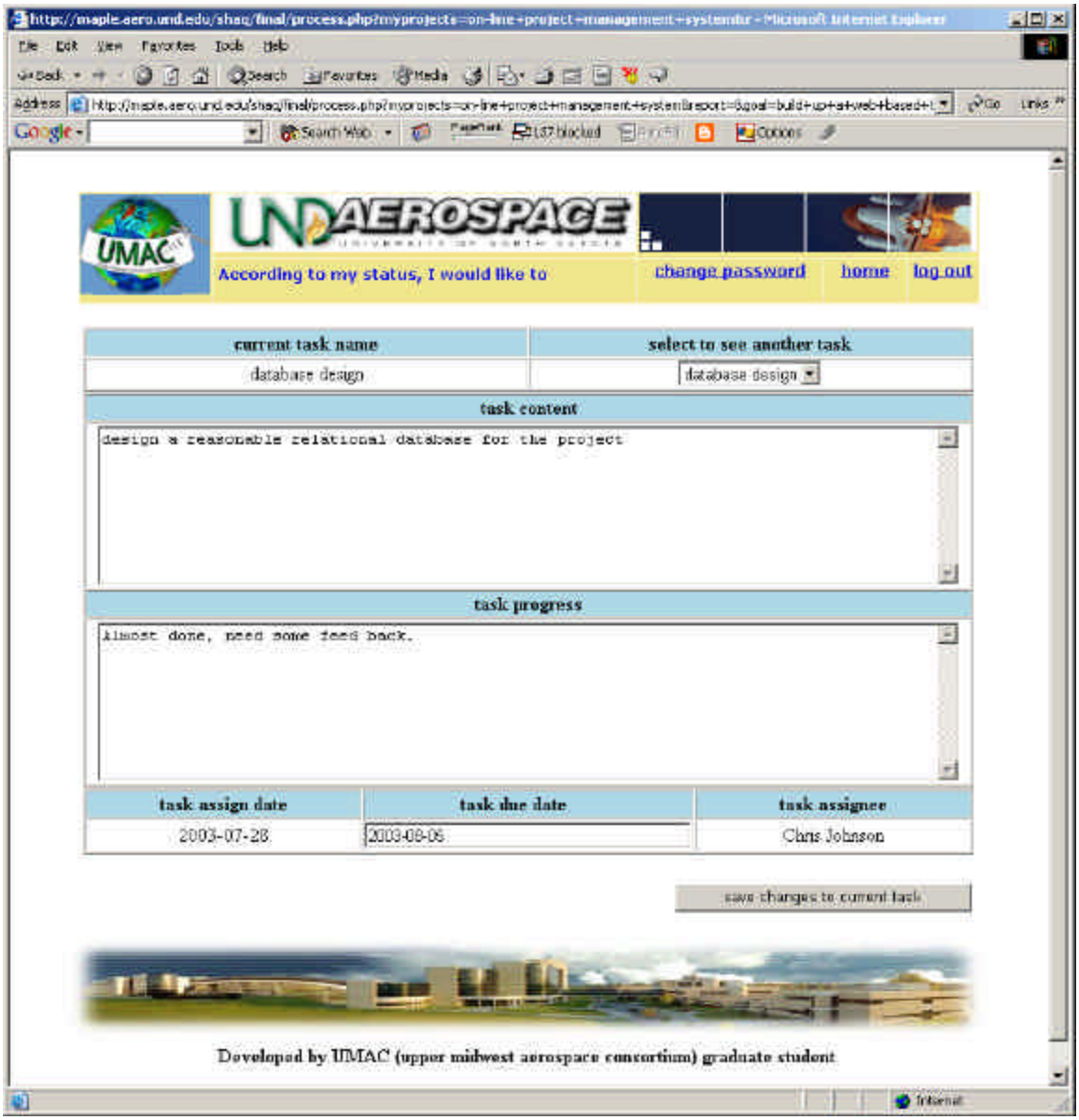


Figure 16. Team Leader's Interface to Browse Tasks Under a Project

A team leader could assign a task under current project. He/she could also send report of his/her project. Those two functions' interfaces are easy to follow, just click the

buttons. There is one more button called “browse tasks”. If a team leader clicks this button, he/she will see the tasks under the current project, see figure 15. Task’s due date and content can be changed by a team leader. To see other tasks under the same project, click the drop down list on the top right corner.

4.3 Project Team Member

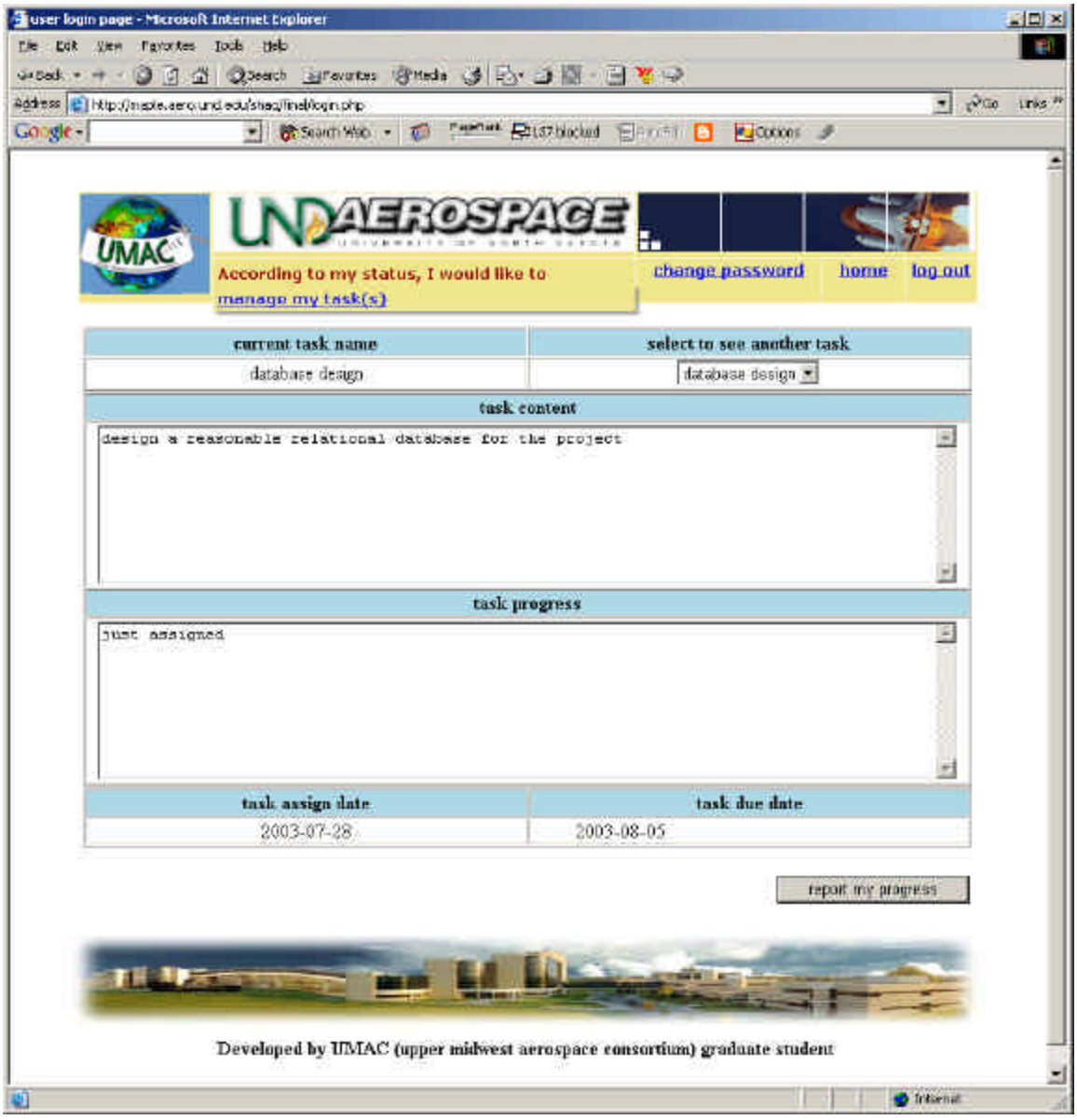


Figure 17. Team Member’s Initial Interface with Only One Options

The interface for a team member is rather simple compare to a supervisor or team leader's. Because the only option for a team member is to manage his/her tasks and report the processes. When a new task was assigned, the progress will be "just assigned". It can be modified by the logged in team member. If he/she would like to see other tasks have been assigned to him/her, just click the drop down list on the top right corner.

Finally, the user must log out the system when he or she is done with the session. Periodically change user password is recommended.

5. Maintenance

All the functions are in functions.inc file, for the easy maintenance in the future. All the codes are with comments for future maintainer to understand. There are two main PHP programs. One is for the login process; the other is for user's action processing.

REFERENCES

[1] Pressman, Roger S., Software Engineering, fifth edition, McGraw-Hill Higher Education, 2001.

[2] Elmasri | Navathe, Fundamentals of Database Systems, third edition, Addison-Wesley, 2000.