



# Media Services

## **Media Services Online**

Mohammed Abukhiran

Report 13 on the work of Week 13

**Berea College**  
**Nov 30, 2010**

## ***Application Development***

### **Project Concept Proposal**

Media Services at Berea College uses Voyager (Database Management System) for keeping track of equipment, Pure Invoice for billing, and FileMaker for keeping track of projects. Working with three different systems at the same time becomes considerably annoying.

The Voyager system is usually used by libraries for keeping track of books and other items, when using for managing equipment many different problems occur. Also, the fact that all three systems mentioned about are completely disconnected from each other creates many problems. Media Services Online is a Database Management System that includes all three programs in one. Having the experience of working with Media Services for three years allows me to create a program that fulfills their needs.

The intended users are student workers at Media Services. And in the long run faculty, staff, students and other community members. My project will create a much easier working environment for Media Services student workers. Instead of going through the calendar day by day to find out where is our overdue equipment, it would take one click. Media Services faculty would find my program very helpful when trying to view a query of equipment used in the past month. There are many other very beneficial functionalities of my program.

Media Services Online will be able to keep track of projects, equipment issue, and billing. The program will include a database server and an interactive online interface. There are many similar programs that have the same functionality as mine but my program will be specifically built to fit the needs of Media Services.

#### Resources:

- Robert J. Lewis, Media Services Coordinator.
- Platform: Microsoft Windows
- Language:
  - MySQL 5.1: relational database management system, for designing the database
  - PHP 5.3.3: Hypertext Preprocessor, for designing the interface.
  - Apache Web Server: Serves as a local host machine. (update Sep 27)

- Tools:
  - Adobe Dreamweaver CS5: for designing the web page.
  - MySQL Workbench 5.2 CE: for designing the entity relationship diagram, and writing the MySQL code for the Database. (Updated Oct 26, 2010)
  - DBQwikiSite v.6: for designing the Interface. (Updated Oct 26, 2010)

Many different challenges will be faced during this project's development. One of the biggest challenges is to create an ER diagram for the database. Another challenge will be designing an interactive easy to use interface.

#### Measures

Week 2	Creating an ER diagram
Week 3 – Week 6	Building the Database
Week 6 – Week 7	Building the Website
Week 7 – Week 11	Building the Interface
Week 11 – Week 13	Testing
Week 13 – Week 15	Repairs and Modifications

If time permits future extensions can be made. One of the major extensions will be to allow students, faculty, or staff to access the program online and reserve equipment, put it orders for projects and many other features. This allows student workers to concentrate their time on doing projects, and makes it easier for customers to utilize our services.

#### Inspiration

I am a student worker at Media Services; the problems I run into on daily bases were my primary motivation to choose this project. I am very interested in database management and I think this project will improve my skill.

#### Vision and Scope

Media Services Online (MSO) will allow student workers to reserve equipment, process projects, and create bill slips for customers. Having three different programs in one provides a much easier working environment for the student workers, and answers many questions for the staff members. If time permits, MSO will be open to the public to reserve equipment and put in orders online.

Media Services Online will keep track of every piece of equipment, for example if a customer was to check out a Triflex sound system, MSO will provide the location, dates, condition of the equipment, due date and other information needed to keep track of the equipment. It will also send emails to customers with overdue equipment providing the charges that will be added to their account. MSO will keep track of projects and notify customers when their project is completed. Finally, MSO will automatically create bill slips for customers based on the amount of work spent. Adding the data about available equipment and project capabilities is considered to be out of scope.

### **Software Requirements Specifications**

Using your Scope Document as a guide, precisely and explicitly describe all of the primary software requirements using the following template for each specific requirement:

- Number: 1
  - Statement: MySQL based database
  - Evaluation: Database to hold the information about equipments, projects and customers.
  - Depends Upon: None
  - Priority: Essential
  - Requirement Revision History: Appropriately formatted September 15th.
- Number: 2
  - Statement: Web Based, Standards Compliant Interface
  - Evaluation: Interface can be accessed from any major web browser: Firefox, Internet Explorer, or Google Chrome.
  - Depends Upon: None
  - Priority: Essential
  - Requirement Revision History: Appropriately formatted September 15th.
- Number: 3
  - Statement: Query Processer
  - Evaluation: Allows requesting information from a database.
  - Depends Upon: 1
  - Priority: Essential
  - Requirement Revision History: Appropriately formatted September 15th.

- Number: 4
  - Statement: Notification System
  - Evaluation: Notifies overdue customers, and completed projects via email.
  - Depends Upon: 1
  - Priority: High
  - Requirement Revision History: Appropriately formatted September 15<sup>th</sup>.
- Number: 5
  - Statement: Billing System
  - Evaluation: Creates bill slips for projects based on amount of work.
  - Depends Upon: 1
  - Priority: High
  - Requirement Revision History: Appropriately formatted September 15<sup>th</sup>.
- Number: 6
  - Statement: Web Server and interface
  - Evaluation: Interface can be accessed from any major web browser, which allows community members to reserve equipment and create projects.
  - Depends Upon: 1 and 2
  - Priority: If time permits
  - Requirement Revision History: Appropriately formatted September 15<sup>th</sup>.

## **System Design and Architecture**

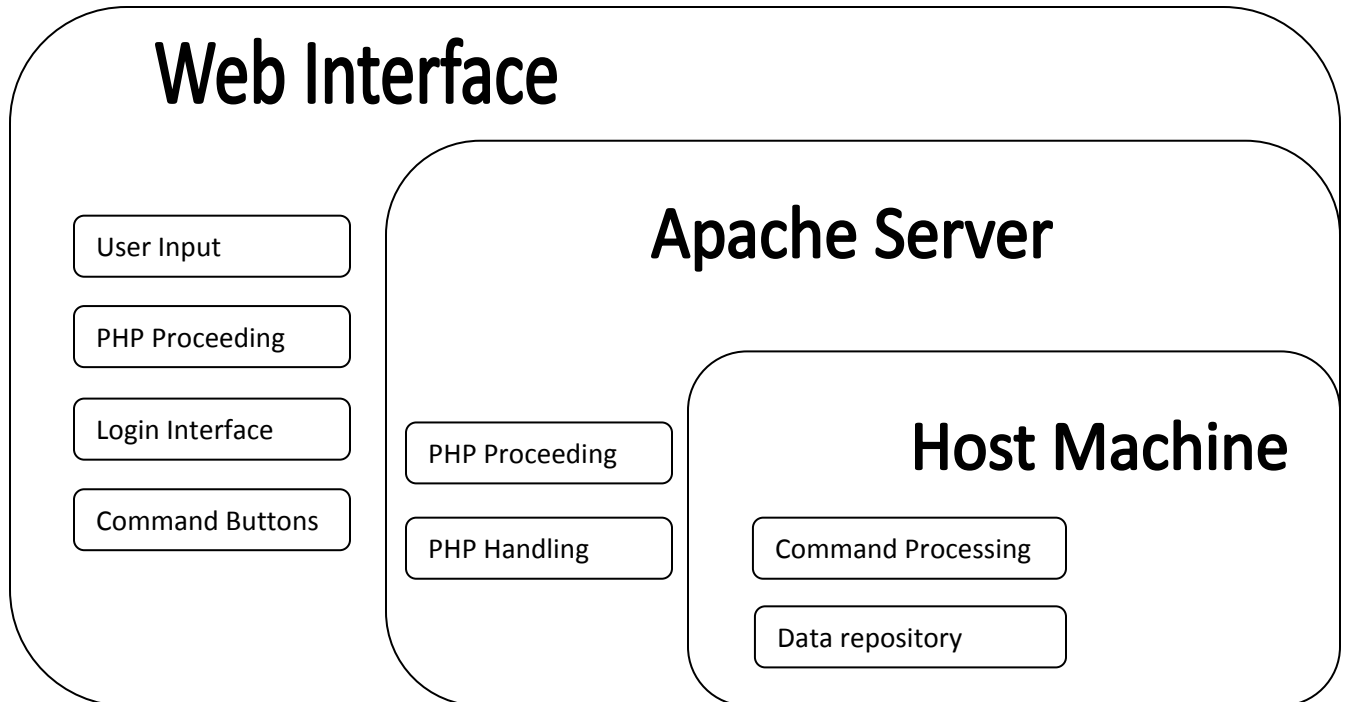
Media Services online is broken down to three main functionalities:

1. Equipment reservation: Allows student workers at Media Services to reserve equipment for customers using a friendly web-based user interface. Keeps track of late equipment and sends automated emails to overdue customers.
2. Project handling: Allows student workers at Media Services to put in orders for customers using a friendly web-based user interface. Keeps track of progress made on each individual project. Sends automated emails to the customer when their project is done.
3. Billing: Allows sending automated bill slips to customers. Keeps track of finance.

Media Services Online consists of three major software components:

1. Database: MySQL database contains all data related to available equipment, project capabilities, customer information.
2. Web server: Apache web server works as local server to grant communication between the database and the interface.

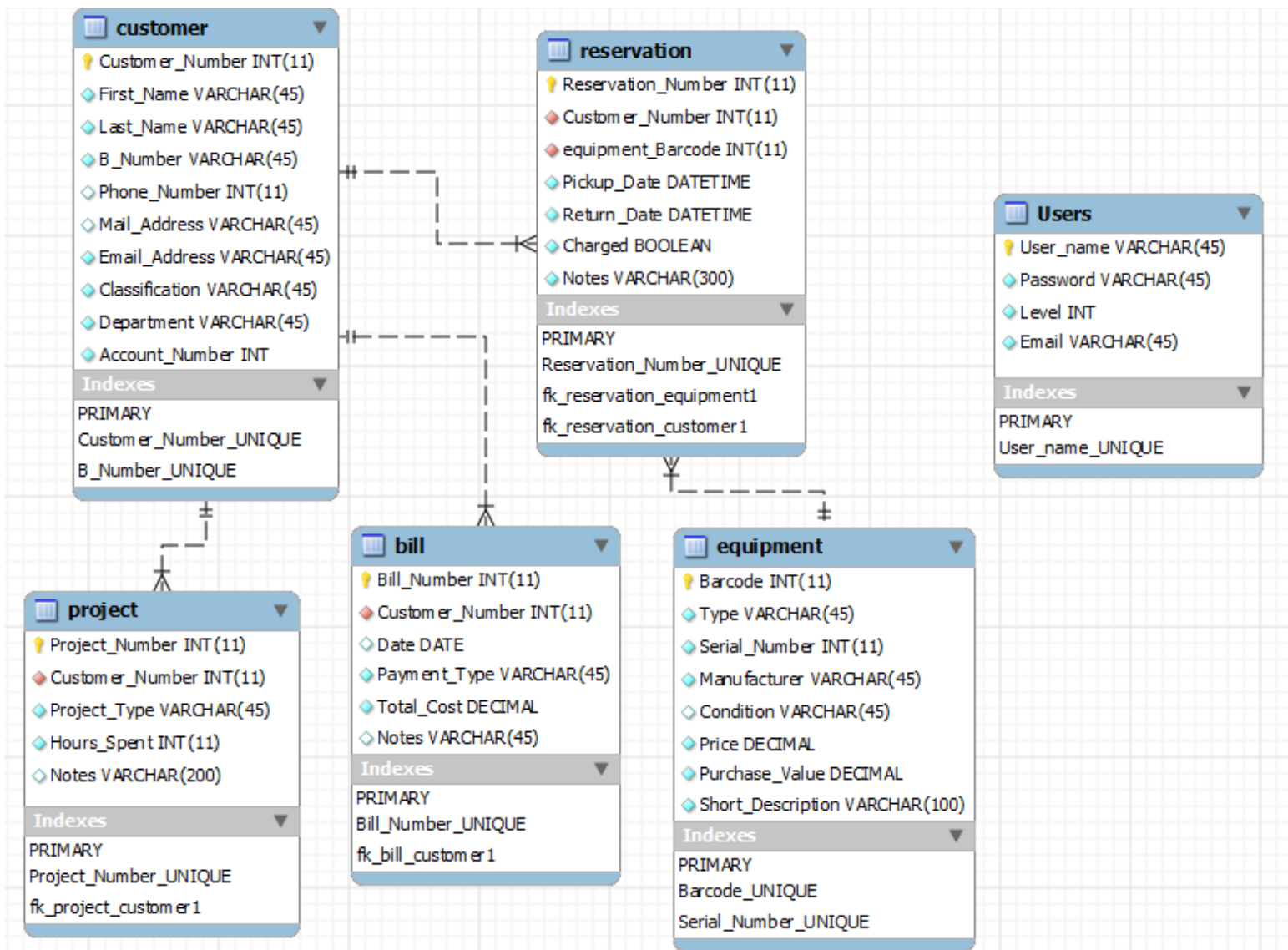
3. User interface: PHP based interface that creates a friendly environment to utilize the functionalities of MSO.



The above nested diagram shows the layout of the overall design of MSO. Functionally, the program will be database management system that requires direct communication among the Web Interface, Apache Web Server, and the Host machine.

The interface will allow access to the functionalities of the program. User input will be processed using PHP and send to the host machine through the web server. Apache web server will function as a link between the interface and the host machine. After processing the commands the host machine output the results to the interface through the web server. The interface will display a readable version of the output to the user.

(Update Nov 16, 2010) Below is an entity relationship diagram that provides a better understanding of the database management system.



## Implementation

- README.txt (*Completed Oct 5, 2010*) – This file holds information about MSO and system and hardware requirement instructions on how to install it.
- mso.sql (*Updated Nov 16, 2010*) – This file holds the MySQL code for the database.
- Interface (*Updated Nov 16, 2010*) – holds the code for the web interface.
  - Images – contains images used in the interface.
  - Js - contains different java functions such as functions to dynamically add/remove filter objects and save their contents
  - Cascading Style sheets designs for
    - Bill.css
    - Bill\_add.css
    - Bill\_delete.css
    - Bill\_edit.css
    - Bill\_more.css
    - Bill\_search.css
    - Customer.css
    - Customer\_add.css
    - Customer\_delete.css
    - Customer\_edit.css
    - Customer\_more.css
    - Customer\_search.css
    - Equipment.css
    - Equipment\_add.css
    - Equipment\_delete.css
    - Equipment\_edit.css
    - Equipment\_more.css
    - Equipment\_search.css
    - Project.css
    - Project\_add.css
    - Project\_delete.css
    - Project\_edit.css
    - Project\_more.css
    - Project\_search.css
    - Reservation.css
    - Reservation\_add.css
    - Reservation\_delete.css
    - Reservation\_edit.css
    - Reservation\_more.css
    - Reservation\_search.css
    - Users.css
    - Users\_add.css
    - Users\_delete.css
    - Users\_edit.css
    - Users\_more.css
    - Users\_search.css
    - Global\_group\_login.css - style sheet for the login page.
    - Global\_group\_passwords.css - style sheet for the login passwords page.



- Bill\_control\_panel.html
- Customer\_control\_panel.html
- Equipment\_control\_panel.html
- Project\_control\_panel.html
- Reservation\_control\_panel.html
- Users\_control\_panel.html
- Index.html – Home page.
- QSSiteNav.html – Navigation page.
- Site\_control\_panel.html – Website control panel.
- Qwikcalender.js – calendar design for entering dates.
- PHP code for:
  - Bill.php – contains definitions for the bill entity.
  - Bill\_add.php - allows adding data entries in the bill table.
  - Bill\_delete.php - allows deleting data entries in the bill table.
  - Bill\_edit.php – allows editing data entries in the bill table.
  - Bill\_more.php - allows viewing data entries in the bill table.
  - Bill\_search.php - allows search data entries in the bill table.
  - Customer.php - contains definitions for the customer entity.
  - Customer\_add.php – allows editing data entries in the customer table.
  - Customer\_delete.php – allows deleting data entries in the customer table.
  - Customer\_edit.php - allows editing data entries in the customer table.
  - Customer\_more.php – allows viewing data entries in the customer table.
  - Customer\_search.php – allows searching data entries in the customer table.
  - Equipment.php – contains definitions for the equipment entity.
  - Equipment\_add.php – allows adding data entries in the equipment table.
  - Equipment\_delete.php – allows deleting data entries in the equipment table.
  - Equipment\_edit.php – allows editing data entries in the equipment table.

- Equipment\_more.php – allows viewing data entries in the equipment table.
  - Equipment\_search.php – allows searching data entries in the equipment table.
  - Project.php – contains definitions for the project entity.
  - Project\_add.php – allows editing data entries in the project table.
  - Project\_delete.php – allows adding data entries in the project table.
  - Project\_edit.php – allows editing data entries in the project table.
  - Project\_more.php – allows viewing data entries in the project table.
  - Project\_search.php – allows searching data entries in the project table.
  - Reservation.php – contains definitions for the reservation entity.
  - Reservation\_add.php – allows adding data entries in the reservation table.
  - Reservation\_delete.php – allows deleting data entries in the reservation table.
  - Reservation\_edit.php – allows editing data entries in the reservation table.
  - Reservation\_more.php – allows viewing data entries in the reservation table.
  - Reservation\_search.php – allows searching data entries in the reservation table.
  - Users.php – contains definitions for the Users entity.
  - Users\_add.php – allows adding data entries in the Users table.
  - Users\_delete.php – allows deleting data entries in the Users table.
  - Users\_edit.php – allows editing data entries in the Users table.
  - Users\_more.php – allows viewing data entries in the Users table.
  - Users\_search.php – allows searching data entries in the Users table.
- 
- Global\_group\_login.php – control the login page.
  - Global\_group\_password.php – users passwords
  - Global\_group\_users.php – usernames.
  - Qs\_connection.php – connecting to the MySQL database.
  - Qs\_functions.php – contains functionalities for the interface.

## **Mohammed Abukhiran, Lead Developer**

Mohammed is a senior Mathematics and Computer and Information Sciences major at Berea College, and is also a student worker for Berea College Media Service. Originally, his interest in computers was primarily focused on computer programming and artificial intelligence, but after working at Media Service Mohammed understood the need of a better database management system than the one currently being used. Problems caused by the current database system were the main motivation leading to this project.

### **Known Bugs and Other Issues – in *Media Services Online 0.1 beta***

- Adding equipment in the equipment table is not functioning properly.
- There is an internal service error when logging in to the database.

### **Known Bugs and Other Issues – in *Media Services Online 0.2 beta***

- Some functions are deprecated. Meaning they used to work in older versions of PHP and not in PHP 5.3
- Function `ereg_replace()` should be replaced with a better function such as `preg_replace()`, and the parameters need to be fixed.
- Email functionality not working correctly.

### **Known Bugs and Other Issues – in *Media Services Online 0.2 beta***

- Email Functionalities are not working.

## Preliminary Test Plan and Preliminary Test Cases

<b>Name</b>	<b>PHP Processing</b>
<b>Requirements</b>	None
<b>Preconditions</b>	Both PHP and Apache are installed and setup on the host machine
<b>Steps</b>	<ol style="list-style-type: none"><li>1. Go to <a href="http://localhost">http://localhost</a> or 127.0.0.1 on the host machine's web browser.</li><li>2. The page should display the interface.</li></ol>
<b>Expected Results</b>	Everything works correctly

<b>Name</b>	<b>Standards Compliant Web Programming</b>
<b>Requirements</b>	PHP Processing
<b>Preconditions</b>	The root http folder must be setup correctly on the host machine Both PHP and Apache must be installed on the host machine The host machine must have a standards compliant browser
<b>Steps</b>	<ol style="list-style-type: none"><li>1. Go to <a href="http://localhost">http://localhost</a> or 127.0.0.1 on the host machine's web browser.</li><li>2. Visit the page in Firefox, IE, and Chrome</li></ol>
<b>Expected Results</b>	The page displays correctly in all 3 different browsers

<b>Name</b>	<b>Login Processing</b>
<b>Requirements</b>	PHP processing
<b>Preconditions</b>	User has authentication to access the database
<b>Steps</b>	<ol style="list-style-type: none"><li>1. Go to <a href="http://localhost">http://localhost</a> or 127.0.0.1 on the host machine's web browser.</li><li>2. Enter the Username and password</li></ol>
<b>Expected Results</b>	<ul style="list-style-type: none"><li>• The user can login if he passes the right parameters</li><li>• The user will have no access to the database if not.</li></ul>

<b>Name</b>	<b>Permissions</b>
<b>Requirements</b>	PHP processing
<b>Preconditions</b>	User has Permission to access specific functionalities of database
<b>Steps</b>	<ol style="list-style-type: none"><li>1. Login as a "student" or "admin"</li><li>2. Try to access "User" data.</li></ol>
<b>Expected Results</b>	<ul style="list-style-type: none"><li>• The user can access the User data if he is an "admin"</li><li>• The user will have no access to the data if he is "student".</li></ul>

<b>Name</b>	<b>Adding data</b>
Requirements	PHP processing
Preconditions	User has Permission to access specific functionalities of database
Steps	<ol style="list-style-type: none"> <li>1. Go to add customer.</li> <li>2. Pass the right parameter</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>• The data will be added if the user passes the right parameters.</li> <li>• If not and error message will show up.</li> </ul>

<b>Name</b>	<b>Editing data</b>
Requirements	PHP processing
Preconditions	User has Permission to access specific functionalities of database
Steps	<ol style="list-style-type: none"> <li>1. Go to edit bill.</li> <li>2. Pass the right parameter</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>• The data will be edited if the user passes the right parameters.</li> <li>• If not and error message will show up.</li> </ul>

<b>Name</b>	<b>Removing data</b>
Requirements	PHP processing
Preconditions	User has Permission to access specific functionalities of database
Steps	<ol style="list-style-type: none"> <li>1. Go to remove project.</li> <li>2. Pass the right parameter</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>• The data will be removed if the user passes the right parameters.</li> <li>• If not and error message will show up.</li> </ul>

<b>Name</b>	<b>Viewing more data</b>
Requirements	PHP processing
Preconditions	User has Permission to access specific functionalities of database
Steps	Go to more customer.
Expected Results	More specific data about the customer will show up.

<b>Name</b>	<b>Logging out</b>
Requirements	PHP processing
Preconditions	User has authentication to access the database
Steps	Logout
Expected Results	The user will be logged out

<b>Name</b>	<b>Password recovery</b>
<b>Requirements</b>	PHP processing
<b>Preconditions</b>	User forgot his password
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. Click "Forgot password" in the login screen</li> <li>2. Enter your email address</li> </ol>
<b>Expected Results</b>	The user will receive an email with his current password

<b>Name</b>	<b>Data Varification</b>
<b>Requirements</b>	PHP processing
<b>Preconditions</b>	User enters wrong type of data in a field
<b>Steps</b>	Enter words in an integer field for example
<b>Expected Results</b>	The user will receive an error message with an explanation

<b>Name</b>	<b>Foreign key specifications</b>
<b>Requirements</b>	PHP processing
<b>Preconditions</b>	User enters a non-existing value
<b>Steps</b>	<p>When making a reservation</p> <p>Enter a "barcode" that is not in the "Equipment" table</p>
<b>Expected Results</b>	The user will receive an error with an explanation

## Software demo

My demo explains the purpose of the software. Then it goes on to explain the administrator functionalities and the student functionalities. It shows how to add, edit, remove, and view data in more detail.

The link to my demo is <http://www.youtube.com/watch?v=cYihOtSeCBO>.

***Executive Section***



TO: Dr. Jan Pearce, Project Director

FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Sep 09, 2010

Deciding which project to choose was very hard and required a great amount of research. I had an interest in Artificial Intelligence but all the projects I was interested in required a great amount of work making it impossible for me to accomplish during one semester. Taking into consideration the amount of time allowed for this project I decided to do Media Services Online. I have done research to determine which programming language would be best for my needs. Also, I did research to create a programming life-cycle for my project.

The biggest challenge was deciding which project to do. I spent a long time researching different projects and determining which would be possible to do. Finding a best-fit programming language was a bit challenging but after some research it became relatively easy.

Deciding which project to do took about five hours, and finding a language took about one hour. Deciding a life cycle for the project took about 2 hours. The total time spent on this project was about 8 hours. The project goals for the following week is to create a working ER diagram and have the needed software installed and ready to use.

# Media Services

TO: Dr. Jan Pearce, Project Director

FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Sep 15, 2010

Writing down the vision and scope was more complicated than I thought. It required a great amount of careful thinking and evaluation, but after meeting with Rob everything went good. After creating the scope and vision, writing down the preliminary software development specifications was much easier. My general understanding of the project is now very complete and I feel like I am ready to move to the next step.

The biggest challenge was creating the scope document. I spent a long time researching different functionalities and determining which would be possible to do in the time frame we have. Creating the preliminary software development specifications was a bit challenging but after some research and looking at previous projects it became relatively easy.

Creating the scope and vision documents took about four hours, and creating the preliminary software development specifications took about two hour. The total time spent on this project was about 14 hours. The project goal for the following week is to refine the Software Requirements Specifications section.



TO: Dr. Jan Pearce, Project Director



FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Sep 21, 2010

I felt very confident with my Software Requirements Specifications, so I decided to start on creating a preliminary design of my software architecture. I started by doing some research about entity-relationship (ER) diagram.

I did not face any challenges so far in designing the software architecture. Building the ER diagram is always challenging but I believe that I have a good idea of how each entity is related to each other. Researching the different types of entity relations was very helpful to my understanding.

Refining the preliminary software development specifications took about one hour. Research about entity relations took about two hours. The process of creating a preliminary design for the software architecture took about three hours. The total time spent on this project was about 20 hours. The project goal for the following week is to refine the Software Requirements Specifications section.

TO: Dr. Jan Pearce, Project Director



FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Sep 27, 2010

Building the system design and architecture was a bit more complicated than I thought. It required a great amount of careful thinking and evaluation. After creating the system architecture, I accomplished a better understanding of my project and how each component flows with each other. My general understanding of the project is now very complete and I feel like I am ready to move to coding.

The biggest challenge was creating the system design and architecture. I spent a long time researching different functionalities and determining which would be possible to do.

Building the system design and architecture took about five hours, and building an entity relationship diagram took about two hour. The total time spent on this project was about 27 hours. The project goal for the following week is to refine the ER-diagram and start coding the database in MySQL.

TO: Dr. Jan Pearce, Project Director



FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Oct 05, 2010

This week I refined my ER-diagram which was much easier now that I had it all set up. I downloaded all the needed software to start coding the database. I created the SQL code for the database based on my ER-diagram.

The biggest challenge was converting the ER-diagram to code. I spent a long time researching and relearning SQL code which helped me greatly during the implementation. Exporting the SQL code to a file was very challenging but after some research it was successful.

Refining the ER-diagram took about two hours, writing the SQL code for the database took about four hours, and exporting the SQL to a file took about one hour. The total time spent on this project was about 34 hours. The project goal for the following week is to refine the SQL code for the database, and start working on the interface architecture.

TO: Dr. Jan Pearce, Project Director



FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Oct 19, 2010

During the past two weeks I refined my ER-diagram and rewrote my SQL code for the database. I downloaded all the needed software to start coding the interface. I created the PHP code for the database web interface based on my SQL code.

The biggest challenge was linking PHP to my Apache server. I spent a long time researching and relearning PHP and Java code which helped me greatly during the implementation. Creating a connection between the interface the SQL database was a bit challenging but after hours of work it went successfully.

Refining the ER-diagram and writing SQL code took about three hours, writing the PHP and Java code for the database took about eight hours, and exporting creating a connection between the database and the interface took about two hours. The total time spent on this project was about 47 hours. The project goal for the following week is to refine the PHP code for the database web interface, and fix some connection problems.

TO: Dr. Jan Pearce, Project Director



FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Oct 26, 2010

During the past week I refined my code for the web interface for the database. I rewrote the PHP code for the database web interface based on my SQL code. I produced Media Services Online 0.1beta. Also, I did some partial testing.

I faced a challenge in rewriting the code for the web interface. The biggest challenge was linking PHP to my Apache server again. Creating a connection between the interface the SQL database was partially successful.

Writing the PHP and Java code for the database took about seven hours, and creating a connection between the database and the interface took about two hours. The total time spent on this project was about 56 hours. The project goal for the following week is to refine the PHP code for the database web interface, and fix some connection problems though extensive testing.

TO: Dr. Jan Pearce, Project Director



FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Nov 02, 2010

During the past week I refined my ER-diagram and rewrote my SQL code for the database based on some feedback I received from the customer. I refined my code for the web interface for the database. I rewrote the PHP code for the database web interface based on my SQL code. I fixed some connections problems and added new features to the interface. I produced Media Services Online 0.2beta. Also, I did some partial testing.

I faced a challenge in rewriting the code for the web interface. The biggest challenge was linking PHP to my Apache server again. Security was a bit challenging but after some research it became much easier. Creating a connection between the interface the SQL database was fully successful.

Writing the SQL, PHP and Java code for the database took about six hours, and creating a connection between the database and the interface took about one hour. Testing took about two hours. The total time spent on this project was about 65 hours. The project goal for the following week is to refine the PHP code for the database web interface, fix some known bugs though extensive testing and start producing different pages in the interface to fit the needs of my customer. Also define a validate class to ensure that user input conforms to specific standards before used for database submissions.

TO: Dr. Jan Pearce, Project Director



FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Nov 16, 2010

During the past week I refined my ER-diagram and rewrote my SQL code for the database based on some feedback I received from my co-workers at Media Services. I refined my code for the web interface for the database. I rewrote the PHP code for the database web interface based on my SQL code. I fixed some connections problems and added new features to the interface. I produced Media Services Online 0.3beta. Also, I did some testing and created a validate class to ensure that user input conforms to specific standards before used for database submissions.

I faced a challenge in rewriting the code for the web interface. The biggest challenge was getting the email functionality to work. Creating a connection between the interface the SQL database was fully successful.

Writing the SQL, PHP and Java code for the database took about seven hours, and creating a connection between the database and the interface took about one hour. Testing took about two hours. The total time spent on this project was about 65 hours. The project goal for the following week is to refine the PHP code for the database web interface, fix some known bugs though extensive testing and start producing different pages in the interface to fit the needs of my customer. And Fix the email functionality.

TO: Dr. Jan Pearce, Project Director



FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Nov 23, 2010

During the past week I refined my ER-diagram and rewrote my SQL code for the database based on some feedback I received from my boss at Media Services. I refined my code for the web interface for the database. I rewrote the PHP code for the database web interface based on my SQL code. I fixed some email problems and added new features to the interface. I produced Media Services Online 0.4beta. Also, I did some testing on the validate class to ensure that user input conforms to specific standards before used for database submissions. Also, I created a video demo of how to use my software.

I faced a challenge in creating the demo, but after picking the right software to do it, it went smooth.

Refining my program took about two hours. Testing took about two hours. Creating the demo video took about three hours. The total time spent on this project is about 73 hours. The project goal for the following week is to video demo based on peer reviews.



TO: Dr. Jan Pearce, Project Director



FROM: Mohammed Abukhiran

SUBJECT: Media Services Online

DATE: Nov 30, 2010

During the past week I was not able to get much done. I watch the demo videos made by my peers and evaluated them.

I faced not challenges.

Watch the videos and evaluating them took about one hour. The total time spent on this project is about 74 hours. The project goal for the following week is to refine video demo based on peer reviews. And create my poster.