ABSTRACT

The project discusses the design and implementation of the Task Management System (TMS) for the University Services Center at Texas A&M University-Corpus Christi. The Task Management System is designed in SQL server2000 and ASP.NET. The purpose of the system is to automate the task management process. Tasks are set of activities performed by the Student workers in University Services Center. These activities start early in the morning, continue throughout the day and end at night. These tasks include opening the building, scheduling the rooms for seminars, conferences etc., and closing the building at the end of the day among many others. The current process in place is a tedious, manual process with no proper archiving system for data backup. The discussed TMS maintains a database of all available and future tasks and generate reports.
# TABLE OF CONTENTS

**ABSTRACT** ...................................................................................................................................................... ii

**TABLE OF CONTENTS** ......................................................................................................................................... iii

**LIST OF FIGURES** ............................................................................................................................................... v

**LIST OF TABLES** ............................................................................................................................................... vi

1. **INTRODUCTION AND BACKGROUND** ........................................................................................................ 2
   1.1 Task Fulfillment Personnel ............................................................................................................................ 2
   1.2 Tasks .............................................................................................................................................................. 3
     1.2.1 Beginning-of-the-day tasks .................................................................................................................. 4
     1.2.2 End-of-the-day tasks ............................................................................................................................ 4
     1.2.3 Recurring tasks ...................................................................................................................................... 4
     1.2.4 Tasks assigned on the fly .................................................................................................................... 4
     1.2.5 Tour tasks ............................................................................................................................................. 5
     1.2.6 Event management schedule tasks...................................................................................................... 5
   1.3 Present Day Task-Fulfillment Process ........................................................................................................... 5
   1.4 Alternative Task Management System ......................................................................................................... 5

2. **TASK MANAGEMENT SYSTEM (TMS)** .......................................................................................................... 7
   2.1 Administrator Interface .................................................................................................................................. 8
     2.1.1 Manage accounts ............................................................................................................................... 9
     2.1.2 Manage tasks ..................................................................................................................................... 11
   2.2 Event Management System .......................................................................................................................... 14
   2.3 Image Management System .......................................................................................................................... 18
     2.3.1 Functional aspects ............................................................................................................................ 18
     2.3.2 IMS login ........................................................................................................................................... 19
     2.3.3 Uploading event pictures .................................................................................................................. 20
     2.3.4 Viewing event pictures .................................................................................................................... 21
   2.4 Student Worker Interface .............................................................................................................................. 22

3. **SYSTEM DESIGN** .......................................................................................................................................... 26
   3.1 Initial Requirements Analysis ........................................................................................................................ 26
   3.2 Project Frame Work and Components ......................................................................................................... 27
   3.3 Project Architecture ...................................................................................................................................... 28
     3.3.1 Presentation tier ............................................................................................................................... 29
     3.3.2 Logical tier ........................................................................................................................................ 29
     3.3.3 Data tier ............................................................................................................................................ 30
   3.4 Working of Three-tier Architecture ............................................................................................................. 30
   3.5 Stages in a Three-tier Architecture Transaction .......................................................................................... 31
   3.6 The Design Phase ...................................................................................................................................... 33
   3.7 User Interface Design .................................................................................................................................. 33
   3.8 Image Management System Design (IMS) .................................................................................................. 34
   3.9 Database Design ......................................................................................................................................... 34
     3.9.1 Description of database tables ....................................................................................................... 37
   3.10 Data Flow Diagrams .................................................................................................................................. 38
LIST OF FIGURES

Figure 1.1 Hierarchical classification of USC Task Management personnel ............. 3
Figure 2.1 Login Screen.................................................................................................................. 8
Figure 2.2 Administrator Home page .......................................................................................... 9
Figure 2.3 Manage Accounts ........................................................................................................ 10
Figure 2.4 Creating a new user ..................................................................................................... 10
Figure 2.5 Manage tasks ............................................................................................................. 12
Figure 2.6 Manage Open/Close tasks ....................................................................................... 12
Figure 2.7 Manage Tour tasks .................................................................................................... 13
Figure 2.8 EMS tasks requestor form ....................................................................................... 14
Figure 2.10 Administrators Event Notifications ................................................................. 15
Figure 2.11 Administrators interface for Checking Events pending approval .................. 16
Figure 2.12 Event Approval Screen ...................................................................................... 17
Figure 2.12 IMS Home Page ........................................................ .................................................. 19
Figure 2.13 IMS add event pictures ......................................................................................... 20
Figure 2.14 upload event pictures ............................................................................................. 21
Figure 2.15 IMS View Pictures ............................................................................................... 21
Figure 2.16 Student workers Home page .................................................................................. 22
Figure 2.17 Student workers tasks view menu ........................................................................ 23
Figure 2.18 Open/Close tasks fulfillment form ........................................................................ 24
Figure 2.19 Open/Close Check list fulfillment form .............................................................. 24
Figure 2.20 Tour fulfillment form ............................................................................................ 25
Figure 3.1 Three-tier architecture [c-sharpcorner 2005] .................................................. 29
Figure 3.2 The client/server relationship .................................................................................. 31
Figure 3.3 Web Database Architecture .................................................................................... 31
Figure 3.4 High level DFD ......................................................................................................... 32
Figure 3.5 User Interface design ............................................................................................... 34
Figure 3.6 Database Design ..................................................................................................... 35
Figure 3.7 Level 0 DFD for TSM [USC 2004] ........................................................................... 39
Figure 3.8 Level 1 DFD for Student worker [USC 2004] .................................................. 40
Figure 3.9 Level 2 DFD for Student workers Open tasks [USC 2004] ........................... 41
Figure 3.10 Level 2 DFD for Student workers Close tasks [USC 2004] ................................ 42
Figure 3.11 Level 2 DFD for Student workers Tour tasks [USC 2004] ........................... 42
Figure 3.12 Level 2 DFD for EMS tasks [USC 2004] ...................................................... 43
Figure 3.13 Level 1 DFD for Administrative Tasks [USC 2004] ....................................... 44
Figure 3.14 Overview of TSM [USC 2004] ............................................................................. 44
LIST OF TABLES

Table 3.1 Listing of major scripts with their usage. ................................................................. 45
1. INTRODUCTION AND BACKGROUND

This project deals with the implementation of a Task Management System for assisting the maintenance personnel at the University Services Center (USC) at Texas A&M University–Corpus Christi. The USC is dedicated to providing goods and professional services to students, faculty, staff and visitors to the campus. Their goals are to assist in enhancing the quality of student life on campus and provide support to the educational, research and public service mission of the University. The USC has many offices and facilities, which cater and assist to the needs and requirements of students, faculty and staff. These facilities and services include Sand Dollar Office, Breaker’s Game room, a Post Office, Convenient Store, Food Court, Campus Copies etc.

Every day the USC building maintenance staff follows a specific set of task lists for ensuring proper functioning of these facilities and scheduling different events occurring at the USC. This process of task assignment and implementation forms an integral part of USC operations. A typical maintenance schedule at USC involves opening and closing the building every day, locking up different facilities at the end of the day, performing cleaning operations in the building hallways, restrooms and cafeteria, scheduling rooms for seminars, meetings, and career fairs and periodically taking headcount in some facilities for security and analysis. The entire course of completing these activities is termed as “task fulfillment.”

1.1 Task Fulfillment Personnel

The task fulfillment personnel at USC include a building manager, Student managers, and Student workers. They share the collective responsibility of ensuring
efficient functioning of USC. The nature of duties of these personnel varies depending on their respective roles. The hierarchical classification of their roles is shown below.

Figure 1.1 Hierarchical classification of USC Task Management personnel

1.2 Tasks

Tasks are a group of duties usually assigned to a Student worker by the Student manager. These tasks are fulfilled by different Student workers and are reported back to the Student manager. These tasks are broadly classified as:

- Beginning-of-the-day tasks
- End-of-the-day tasks
- Recurring tasks
Tasks assigned on the fly
Tour tasks
Event management schedule tasks

1.2.1 Beginning-of-the-day tasks

These tasks are performed by a Student worker at the start of the day (usually at 7:00 a.m.). These include opening the building, unlocking some of the facilities, switching on the lights in hallways and in rooms, checking the rest rooms for proper cleanliness and checking the facilities for proper trash disposal etc.

1.2.2 End-of-the-day tasks

These tasks are performed at the end of the day (usually at 10:00 p.m.). These include closing the building, closing and locking the facilities, switching off the lights etc.

1.2.3 Recurring tasks

These tasks are activated for a certain period and are completed at regular intervals.

1.2.4 Tasks assigned on the fly

These tasks are assigned by the Student manager to a Student worker who is working on that particular shift. These tasks are assigned on the fly to Student workers.

1.2.5 Tour tasks

These types of tasks are performed every hour by touring around the USC building to obtain status of rooms, which may be opened/closed/locked/occupied. Occasionally, this may also involve taking the headcount in some of the rooms for security and analysis purposes.
1.2.6 Event management schedule tasks

Event management tasks are requested by various departments at the University for performing their events. These may include tasks like arranging banquet halls, arranging for career fair, scheduling rooms and other business office operations during the start of semester in registering for classes etc.

1.3 Present Day Task-Fulfillment Process

The present day task-fulfillment process is based on manual record keeping using a paper-based system. In this system, a printed task sheet with a list of all “Open tasks” to be performed on a given day is handed out to Student workers. The Student worker subsequently marks off the tasks that have been completed including the date and time of completion. These tasks may be any of the ones that were discussed in the previous section.

However, the current system does not provide the flexibility of adding new tasks to the existing task form as some tasks may be added or scheduled for only a certain period. In this case, an unforeseen task is associated with a note on the existing form. On some occasions, an instant schedule of an assignment by a Student manager requires him/her to inform the Student worker in person. This method of task assignment has often been found to be tedious and inefficient.

1.4 Alternative Task Management System

Considering the limitations of the existing system, an alternative system based on automated task scheduling and maintenance streamlines the task management process by coordinating workflow across different levels of management. This system also simplifies the process by providing hands-on tools for proactive management.
The proposed system provides greater flexibility to users by providing customizable user interfaces for adding, deleting, and updating tasks. For example, if the Student manager needs to add a new task to the existing task list he/she can open the form and use the “Create New Task” tool from the interface. The Student worker can now view the updated task checklist and prioritize them accordingly. Similarly, an existing task can be edited or deleted from the task list. Through the new system, the Student managers can also monitor the status of tasks assigned to Student workers in real-time and provide tools for alerting the Student workers. This not only improves the visibility but also helps in keeping tasks on schedule. Finally, real-time reports can be generated at the end of the day or for any given period (for example, a report of all the pending tasks for the last three days can be obtained at the click of a button) through this new system.

The ensuing sections in the report discuss the TMS implementation from both the end-user and developers perspective. Section 2 deals with the user interface design and logical flow of the task management process. Accompanying screenshots are intended to aid in understanding the process flow. Section 3 explains analysis, design and interaction between the interface and database. Testing and evaluation of the TMS under a real-time environment are discussed in Section 4. Lastly, Section 5 concludes the project discussing about the possible incorporation of other features for its full-scale implementation in a real world environment.
2. TASK MANAGEMENT SYSTEM (TMS)

The main objective of Task Management System (TMS) is to design an Online Task Management System for University Services Center at Texas A&M University, Corpus-Christi. The TMS is about fulfilling tasks by building personnel (Administrator, Student manager, Student worker). The type of tasks and fulfillment tasks varies depending on the type of building personnel.

The users of TMS are Building manager, Student manager, and Student worker. Currently the users of the system are using manual processing for task fulfillment. Therefore, the new system is designed keeping in view the existing users and the functionality in mind. This system is developed with simple and robust user-friendly features that include the ability to navigate using any web browser. The user activates a button that initiates an action such as directing the user to a new page or retrieving data from the database. The user enters the system by typing the designated Universal Resource Locator (URL) in a Web browser like Internet Explorer or Netscape Navigator. This takes the user to the main page of the system.

Figure 2.1 gives a screenshot of the Login page of the system.
On this page, the user of the system can enter his/her login information and activate the login button. This processes the login information of the user and allows him/her to access different features of their account. Depending on the type of User-Login information (privileges), corresponding User interfaces are displayed.

TMS has three main interfaces:

- Administrator interface
- Student manager interface
- Student worker interface

### 2.1 Administrator Interface

Each interface has different sets of functions. The functions of Student manager are similar to that of the administrator except that only an administrator can create a Student manager account. The following are the different features in the Administrator interface.
• Manage User Accounts
• Manage tasks
• View Reports
• View/Approve Event tasks (EMS tasks)

Figure 2.2 illustrates the administrator interface.

![Administrator Home page](image)

Figure 2.2 Administrator Home page

Each of the Administrative tasks is further elaborated in the sections following.

2.1.1 **Manage accounts**

Administrator as a part of managing accounts has the privileges of adding/editing/deleting/updating User accounts. Administrator can also view all the existing users of the system.

Figure 2.3 illustrates a screenshot of Manage Accounts interface.
Figure 2.3 Manage Accounts

Figure 2.4 illustrates a screenshot for creating a new user. This is one of the privileges of the administrator.

Figure 2.4 Creating a new user

The user has to provide First Name, Last Name, User Name, Password, and select an Account Type from the drop-down menu, which includes Administrator, Student.
manager, Student worker and other contact information such as email address, and phone number. Upon successful entry of all the information, a new account is created for the user. An email is generated and sent to the user with his/her login information.

2.1.2 Manage tasks

Managing tasks are one of the important functions of the Administrator. The manage tasks menu gives a listing of all the types of tasks like Open/Close tasks, EMS tasks (Events), Tour tasks etc. Upon clicking any of the listed tasks, the administrator is provided with several options regarding the operations that can be done on these tasks. The administrator can create a new task or edit an existing task. Similarly, the administrator can delete a task. The administrator upon clicking EMS Tasks can view all the EMS tasks pending for approval. Under the Assign task button the administrator can assign a Task to the Student worker.

Figure 2.5 illustrates a screenshot for Manage tasks.
Figure 2.5 Manage tasks

Manage Open/Close tasks: Open/Close tasks are fulfilled by Student worker at 7:00 a.m. and at 10:00 p.m. every day. The Administrator can create or delete these tasks.

Figure 2.6 gives a screenshot of Managing Open/Close tasks.

![Figure 2.6 Manage Open/Close tasks](image)

Figure 2.6 Manage Open/Close tasks

Tour tasks are also performed by Student worker every day. Administrator can add a new tour task or delete an existing tour task. Figure 2.7 gives a screenshot for managing tour tasks by administrator.
Figure 2.7 Manage Tour tasks
2.2 Event Management System

Event Management System is an extended module of TMS. Often times there are several events occurring on campus, which may fund raising events for Scholarships. There may also special event requests coming from various departments to the Administrator of USC. These requests may be for rooms for holding seminars, or from Office of Career Services Center for conducting career fair etc. These requests are put forward through a user interface form in which the event requestor provides all the information needed, for instance, reserving a room with the requestor contact information such as Email address or Phone number. Upon submitting a request, the request is forwarded to Administrator pending approval. The following describes the user interface for requester and furnishes the information needed while booking.

Figure 2.8 describes User interface form for event tasks provided for a requestor

![EMS Events Form](image)

Figure 2.8 EMS tasks requestor form
When the event request is submitted over the above form an email is automatically generated and sent to the administrator with a subject “Event To Be Approved (TaskMS Application)”. Once the administrator logs into the system, views all the pending event requests for approval. The following screens illustrate the event approval sequence for administrator.

Figure 2.9 gives a screenshot of Administrator User interface form for checking Event Tasks (EMS)

![Task Management System](image)

Welcome Chau Meakala
You can use the top menu links for the following:

**Manage Tasks:**
Create Open Close Tasks, CheckLists for Open/Close Tasks, Create Your Tasks, Modify the tasks’ contents and Delete
Tasks. Check events and approve the events

**Manage Accounts:**
Create a user, modify the user, see list of all users.

**Assign Tasks:**
Assign various tasks such as Open/Close, Your Tasks and various events to student workers.

![Notifications & Comments](image)

You have 1 event(s) to be approved. [Click Here To Approve]

Figure 2.10 Administrators Event Notifications

The administrator clicks the events under Notifications and comments for detailed description about the type of requests.
Figure 2.11 Administrators interface for Checking Events pending approval

Upon clicking the events, which are submitted by the requestor, the administrator can approve the events by simply changing the event status to ‘approved’ and assigning a room location for the event. The following screen shot illustrates event approval process.
Once the event is approved based on available resources an email is automatically sent to the requestor for confirmation.
2.3 Image Management System

2.3.1 Functional aspects

Various events occur at the USC and as part of these events people collect images for future reference and other purposes. Managing these images for common viewing by interested parties may be a tedious process. Hence, this system provides an easy-to-use tool to collect images from the patrons and load them on the USC server for a standard viewing experience.

After each event, the event managers or participants can send their images to the Building manager. The existing TMS provides the Building managers a module for collecting, storing and uploading multiple images for managing image libraries from different sources. A web link on the public domain is provided for the attendees of a particular event. These images remain on the server for a period of one month and at the end of that period, the Building manager has the privilege to remove the images from the server.

Additional features in IMS include adding, editing and deleting image records to and/or from the existing system. Moreover, image categories and subcategories can be created to group common images. These fields are managed by the Building manager and can be set up to suit the needs of different event types. The IMS interface also contains windows for viewing thumbnails and larger views of selected images. A provision for viewing all the images as a slide show in a category or subcategory is also provided.

The ensuing sections illustrate the various functionalities incorporated in IMS with accompanying screen shots.
2.3.2 IMS login

IMS is a secure and password protected file system. On the administrator’s main menu there is a link provided for IMS system. When the link is clicked, the system asks for login and password information. Upon authentication of the login information, the main page of IMS is displayed. The main page consists of links to upload events, view events where in the event pictures can be viewed as thumbnails or slideshow, ‘EMS events’ link which gives a listing of all events. The following screen shot is the main page of IMS with all the accompanying links.

![IMS Home Page](image)

Figure 2.12 IMS Home Page

Once the EMS events are clicked, there is a listing of past events with the number of pictures uploaded for that event. It also gives the administrator an option of deleting the events or adding a new event.
2.3.3 Uploading event pictures

New events can be uploaded to the server as individual files or folders. The administrator creates a new event name under which all the pictures are uploaded. Upon clicking the add button the administrator is navigated to the folders on the computer. The administrator can then select the pictures to be uploaded. The selected pictures are displayed in the Java applet window along with their size, and last modified date. At this point, the administrator also has the option of deleting the selected pictures. The following screen shots illustrate the upload process.

Figure 2.13 IMS add event pictures
2.3.4 Viewing event pictures

The uploaded event pictures can be viewed as thumbnails or slide shows.
2.4 Student Worker Interface

Student workers use TMS to accomplish daily tasks. Student workers have fewer privileges in terms of creating user accounts, creating tasks, adding or deleting tasks when compared to Administrator and Student manager. Student workers can view their daily schedule for task fulfillment. The following screenshots explain in detail different options available for Student workers in Task fulfillment process.

Figure 2.16 gives a screenshot of Student workers home page.

![Student workers Home page](image)

The following broadly describes the features available for a Student worker upon logging into the system. The Student worker can

- View tasks assigned to him/her.
- Submit tasks for task fulfillment (day-to-day activities).
- View tasks history for analysis
- Change password option.
As part of a Student worker’s daily schedule, the Student worker retrieves the tasks assigned to him/her. The tasks are retrieved based on task types (open/close/tour/events) or tasks waiting for completion. The tasks are also retrieved between specified dates; for example, Open Tasks between 08/1/05 to 08/7/05 and schedule for whole week is displayed for the Student worker.

Figure 2.17 illustrates the view tasks menu for Student worker.

![Task Management System](image)

**Task Management System**

*Texas A&M University - Corpus Christi*

**View Tasks**

- Task Type: 
- From Date: [ ]/ [ ]/ [ ]  
- To Date: [ ]/ [ ]/ [ ]  
- View Schedule

**View this week's schedule**

**Today's Tasks**

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Start Time</th>
<th>End Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open/Close Task</td>
<td>7:00 AM</td>
<td>12:00 AM</td>
</tr>
<tr>
<td>Tour Task</td>
<td>7:00 AM</td>
<td>12:00 AM</td>
</tr>
</tbody>
</table>

**Events**

<table>
<thead>
<tr>
<th>Location</th>
<th>Event</th>
<th>Start Time</th>
<th>End Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cayo Room 210</td>
<td>Orientation Computer Service</td>
<td>1:00 PM</td>
<td>5:00 PM</td>
</tr>
</tbody>
</table>

Figure 2.17 Student workers tasks view menu

Figure 2.18 illustrates the screenshot for Open/Close tasks fulfillment.

Open/Close tasks are regular tasks fulfilled by the Student worker. These tasks are retrieved daily early in the morning at 7:00 a.m. and at 10:00 p.m. during night. Open/Close tasks include opening building floors, hallways and noting the open/close times against them. Upon completion of these tasks, the tasks are submitted.
Figure 2.18 Open/Close tasks fulfillment form

Figure 2.19 is the screenshot for open/close check listed tasks. These tasks are check marked upon completion. These are part of Open/Close tasks. Some of the Check listed tasks include checking rest rooms for trash disposal, turning lights on/off depending on the open/close shifts etc.

Figure 2.19 Open/Close Check list fulfillment form
Tour tasks are also type of regular tasks fulfilled every day. During tour process, the Student worker tours the building and checks for status of the rooms. The status can be opened/closed/locked/occupied etc. Tours are conducted every hour. Some of the tour tasks include taking the count in certain rooms for security and analysis purposes. For example, count is taken in Cafeteria for analyzing business and arriving at business decisions. Based on the occupancy of the cafeteria USC administration expands food counters in cafeteria. Figure 2.20 illustrates the screenshot for tour-task fulfillment process.

![Figure 2.20 Tour fulfillment form](image)

The Student worker also has the option of taking notes or adding comments while touring the building corresponding to a location. These are all the types of tasks a Student worker performs as part of his/her daily schedule.
3. SYSTEM DESIGN

This section describes the design, analysis, and implementation phases of the Task Management System (TMS).

3.1 Initial Requirements Analysis

This project was inspired by the limitations of the USC’s existing task management system. A meeting with Mr. Bob Martin, Systems Programmer at the University Service Center and Mr. Paul Reynolds, Assistant Director of University Center has initiated the idea about automating the existing task management system to make it more efficient and robust.

The meeting has also provided a basis for designing the project with end-user expectations and future requirements in mind. The final system’s delivery model was built after a comprehensive review of the existing system. This was instrumental in developing the design phase of this project, as there was a clear understanding of the final user expectations and the new system’s delivery model. Discussions were also held with Krishana Bhaskar Vommi, Analyst programmer USC along with Naresh Shroff who implemented TMS on a pocket PC platform about the data model and data flow diagrams (DFD). The data flow diagrams followed in the succeeding sections are the excerpts from those discussions. [USC 2004]

After a series of discussions with the management personnel at USC, we have decided to implement Microsoft’s .NET technologies using MS SQL Server as the database server for this project. The scope, the number of users and the security issues were also discussed in these meetings.
3.2 Project Frame Work and Components

The Task Management System (TMS) for the University Service Center was developed using ASP.NET Frame work, JavaScript and MS SQL Server as the database. Image Management System is developed as a separate module in Java.

ASP.NET makes building real world Web applications easier. ASP.NET-server controls enable an HTML-like style of declarative programming that helps in building great web pages with far less code unlike classic ASP.

The reasons for using ASP.NET were:

- **Easy Programming Model** as displaying data, validating user input, and uploading files are easy. Moreover, ASP.NET pages are compatible across a wide range of web browsers.

- **Easy Deployment** As .NET redirects all new requests to the new code, server restarting is not required to deploy or replace compiled code.

- **Flexible Language Options** ASP.NET supports more than 25 .NET languages, including built-in support for VB.NET, C#, and JScript.NET. As additional tools are not required in leveraging existing programming language skills, ASP.NET provides unprecedented flexibility in any choice of language [gotdotnet 2005].

SQL Server database was used as the backend. The following were the reasons for using SQL Server.

- SQL Server is generally easier to install, use and manage.

- SQL Server is highly compatible with .NET framework. Unlike Microsoft Access database, SQL Server provides enhanced capability to the
developer to write Stored Procedures [Shepker 2000]. Stored Procedures help in separating client application from the underlining structure of the database. This further improves the stability of the application through simplified coding on the client end.

- SQL Server is well supported by the University Services Center.
- Crystal Reports are used to generate reports. Crystal reports are embedded in .NET framework and are a very user-friendly tool. End users can generate reports at ease.

### 3.3 Project Architecture

Task Management System is implemented as three-tier architecture. Three-tier architecture is a client-server architecture in which the user interfaces, functional process logic ("business rules"), data storage and data access are developed and maintained as independent modules, most often on separate platforms [Wikipedia 2005]. The following illustrates the components of three-tier architecture:

- Presentation Layer (Front end)
- Logical Layer (Middleware)
- Data Layer (Back end)

In some of the three-tier architecture implementations, logical layer is further divided into Business and Data Access Tiers in order to increase scalability and transparency [Zhao and Kearney 2003]. The tiers can be deployed on physically separated machines. The characteristic of the tier communication is that the tiers will communicate only to their adjacent neighbors. For example, the Presentation Layer tier
will interact directly with the Business Tier and not with Data Access or Data Tiers. Figure 3.1 gives a pictorial representation of three-tier architecture.

![Three-tier architecture](image)

**Figure 3.1 Three-tier architecture [c-sharpcorner 2005]**

### 3.3.1 Presentation tier

This tier is responsible for the presentation of data, receiving user events and controlling the user interface. This tier is responsible for communication with the users and web service consumers and it will use objects from Business Layer to respond to GUI raised events.

### 3.3.2 Logical tier

This is the brain of any three-tier application. This tier is again divided into Business Tier and Data Access Tiers. Although some architects do not make any distinction between Business Tier and Data Access Tiers, owing to slower down
performance, the advantages of separating logical tier into business and data access tiers are two-fold. This separation increases code transparency and helps in modifying data layers at a later stage without interacting with the business layer.

Business Tier: This sub-tier contains classes to calculate aggregated values such as total revenue, cash flow and debit and this tier is independent of GUI controls and database accessibility. The classes of Data Access Tier will supply the information needed from the databases to this sub-tier.

Data Access Tier: This tier acts as an interface to the Data Tier. This tier deals with retrieving and storing information from the database.

3.3.3 Data tier

This tier is responsible for retrieving, storing and updating information; therefore, this tier can be ideally represented through a commercial database. Stored procedures are a part of the Data Tier. Usage of stored procedures increases the performance and code transparency of an application.

3.4 Working of Three-tier Architecture

Implementing three-tier architecture needs client workstations with a web browser, application server (IIS, Apache etc) and a Database. A communication link is established between Client machine’s web browser and applications server. The Web browser makes a request of the application server. The server sends back a response.
3.5 Stages in a Three-tier Architecture Transaction

A typical three-tier architecture transaction consists of the following stages:

- A user’s Web browser issues an HTTP request for a particular Web page.
- The Web server receives the request, separates the HTML tags, and passes the script to the ASP.NET engine for processing.
- The ASP.NET engine begins parsing the script. Inside the script is a command to connect to the database and execute a query (perform the search for the pending tasks). ASP.NET opens a connection to the Database server and sends on the appropriate query (Figure 3.3).

SQL Server receives the database query and processes it, and sends the result back to the ASP.NET engine.

The ASP.NET engine finishes running the script and formats the query results in HTML. It then returns the resulting HTML to the Web server.

The Web server passes the HTML back to the client machine, where the user can view the list of pending tasks that has been requested.
Figure 3.4 illustrates the high level flow of data across the three-tier Architecture.
3.6 The Design Phase

After the initial analysis of the project requirements, the design phase was initiated by consulting with Bob Martin a systems programmer at the University Services Center for the technical design and development of the required components. These discussions mainly centered on the structure of the database and the design of the various user interfaces. Once the technical design was agreed, Reynolds the Assistant Director for University Service Center was consulted again for a functionality review to make sure that there were no deviations from the original functional requirements.

3.7 User Interface Design

Task Management System broadly consists of interfaces classified as per the types of users. The Administrator, Student manager and Student workers are the users of the system. Depending on their privileges, each user interface has different functionality. The Functions of Administrator and Student manager are almost the same except that the administrator has privileges to add Student manager. Both the administrator and the Student manager oversee the functions of Student workers.

Figure 3.5 gives an overview of the Interface design and describes the functions of each user.
3.8 Image Management System Design (IMS)

Image Management system is developed using Java and ASP.NET technologies. When a new event is created for uploading pictures, a corresponding folder is created in the ‘wwwroot’ directory. An user of type ‘ASP.NET’ is automatically created and has write permissions to the event folder. During the upload process, a Java applet is loaded in the browser, which enables to upload pictures. Image Management System does not have a database but the pictures are stored in file folder, which has write permissions. Image Management System is typically a two-tier architecture with client machine interacting with application server (IIS). The folders are created in the ‘inetpub’ directory on the IIS server. IMS works on variety of platforms like Windows XP, MAC e.t.c.

3.9 Database Design

Figure 3.6 describes the database design for the Task Management System. The database is designed comprising eight tables. A user can be Administrator, Student worker, Student manager. A Student worker can have multiple tasks assigned. The
database tables are designed keeping in view the broad functions of each type of users in the system. First, the entities are identified, which are Administrator, Student manager, Student worker. The possible functions for each entity are identified; for example, creating user accounts, creating tasks etc in the case of administrator. Tables are designed linking entities with functions. This formed the basis for database design. For example, the table `tour_Student_Tasks` has information about all the Tour tasks for a Student worker.

![Database Design Diagram](image)

**Figure 3.6 Database Design**

UserAccounts, OpenClose_Student_Tasks, Tour_Student_Tasks, Student_AssignedTask

ask UserAccounts table is the central table of database which has User details, like LastName, FisrtName, Address etc. Each User can have multiple tasks assigned. The
relation between *UserAccounts* and *Student_AssignedTask* is 1 to many. For example, to retrieve all open/close the tasks for a user the flow starts from *Student Assigned Task* table and selects the task type as Open/Close and follows the Open/Close table, which has location information, and open/close timestamp.
3.9.1 Description of database tables

User Accounts: User Accounts table stores User Information. A record in the User Accounts table gives comprehensive information about the User’s of the System. The columns of the user table are User ID (Primary Key), First Name, Last Name, Password, Account Type, (Account types are Administrator, Student manager, Student worker), Phone Number, Email.

OpenClose_Student_Tasks: This table stores the information about Open/Close tasks assigned to the particular User. This table holds the information about the Open/Close task time stamp, and the particular Student worker that has fulfilled these tasks for a given day. The columns in the OpenClose_Student_Tasks table are: User ID, Location ID (Primary Keys), OpenTime, CloseTime, Date.

OpenCloseTasks: This table gives the location information of the Open/Close tasks. The location information is the name of the Floor, Room Name where the Open/Close tasks are performed. The columns in this table are Location ID (Primary Key), Location.

OpenClose_CheckList_Student: As a part of the Open/Close tasks the Student workers checks the tasks for fulfillment or incompletion. That information is stored in OpenClose_CheckList_Student: The columns in this table are User ID, CheckList ID (Primary Key), Open_Flag, Close_Flag, Date.

OpenCloseCheckList: This table stores the names of the tasks that can be Check Listed.

The columns in this table are Checklist ID (Primary Key), Checklist (Name of the location/task that is Check marked)
**Student Assigned Tasks:** This table stores the information about Task types and number of tasks that are assigned to Student worker. A user can be assigned any number of tasks.

The primary key is TID, which is nothing but the Event ID from the Events table. The columns in the Student Assigned Task table are User ID, Task Type (Open/Close, Dynamic tasks, Tour tasks etc), Start Date (Task Activation Date), End Date (Task End Date), Shift, Start Time (Task Start Time), End Time (Task completion time), Notes (Any comments that are entered when a task is accomplished).

**Events:** This table holds the information about Event Management Tasks. The Primary Key is the Event ID. The other columns in the Event table are ResStart (Event Reservation Start Time), ResEnd (Event Reservation End time), Event Start (Event Start Time), EventEnd (Event End Time), Location (Location of the event), Setup Type, Event Type, Contact Name (Requestor Name), Department (Requestor Department), Phone Number.

**TourStudentTasks:** This table stores the information about the Student worker Tour Tasks. User ID, Area ID (Primary Keys), Area ID is the location in the building where the Student worker tours. The other columns in TourStudentTasks table are Count (is the number of occupants in the respective tour locations, Status (It is the status of the rooms during tour process which are Open/Closed/Occupied, Locked) Tour Notes (Comments against tour tasks), Date (Date and time of the Tour).

### 3.10 Data Flow Diagrams

Data Flow Diagrams (DFDs) examine how data flows into, out of, and with in the system. The Level 0 DFD for the TMS is illustrated in Figure 3.7. When a User logs into
the TMS, he or she is authenticated, and based on the account privileges the corresponding tasks are displayed. If the user authentication is failed, it prompts the user to reenter the username and password. Administrator tasks are different from those of Student worker tasks. A Student worker is directed to UC Task Management System and administrator is directed to an Administrative tasks.

![Diagram](image)

**Figure 3.7 Level 0 DFD for TSM [USC 2004]**

Figure 3.8 describes Level 1 data flow with respect to Student worker. A Student worker upon logging into the TMS is presented with the listing of all types of tasks/processes. These tasks are Open tasks, Close tasks, and Tour tasks and EMS tasks.
Figure 3.8 Level 1 DFD for Student worker [USC 2004]

Figure 3.9 describes the Level 2 DFD for Student workers Open tasks. Open tasks fulfillment starts with opening of the building. It checks for the processing of the previous day Close tasks. If the previous day’s Close tasks are not processed the system prompts for processing of previous day Close tasks. Once the Close tasks are processed and entered into the system, Open tasks are processed next.
Figure 3.9 Level 2 DFD for Student workers Open tasks [USC 2004]

Figure 3.10 describes the Level 2 DFD for Student workers Close tasks. Close tasks fulfillment starts with closing the building. It checks for the processing of current day Open tasks. If the current day Open tasks are not processed the system prompts for processing of those tasks. Once the Open tasks are processed and entered into the system Close tasks are processed.
Figure 3.10 Level 2 DFD for Student workers Close tasks [USC 2004]

Figure 3.11 describes Level 2 DFD for Student workers Tour tasks. Tour tasks fulfillment starts with touring the building. Before touring the building, the system checks for the fulfillment of tour tasks from the previous shift. If the previous shift tour tasks are not fulfilled, system prompts the user for fulfilling those tasks. Once these tasks are fulfilled, the user is allowed to continue with the current tour tasks.

Figure 3.11 Level 2 DFD for Student workers Tour tasks [USC 2004]
Figure 3.12 describes Level 2 DFD for EMS tasks. EMS tasks fulfillment usually starts at the beginning of the day. It checks for the processing of previous day EMS tasks. If the previous day EMS tasks are not processed, the system prompts for processing of those tasks. Once the previous day EMS tasks are processed and entered into the system, current day EMS tasks are processed.

![Flowchart: Level 2 DFD for EMS tasks](image)

Figure 3.12 Level 2 DFD for EMS tasks [USC 2004]

Figure 3.13 describes Level 1 DFD for Administrator. Administrative tasks are creating User Accounts, Creating tasks which include edit/update/delete tasks, generating reports. Reports can be generated based on certain criterion. For example, an Administrator can view all the pending tasks for a given day or for a specified period of time.
Figure 3.13 Level 1 DFD for Administrative Tasks [USC 2004]

Figure 3.14 describes Overview of High Level DFD for the Entire Task Management System.

Figure 3.14 Overview of TSM [USC 2004]
3.11 Listing of Major Scripts

Table 3.1 Listing of major scripts with their usage.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Script Name</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>login.aspx</td>
<td>Lets the users to login into TMS</td>
</tr>
<tr>
<td>2</td>
<td>AdminHome.aspx</td>
<td>Administrators Home page</td>
</tr>
<tr>
<td>3</td>
<td>ManageAccounts.aspx</td>
<td>To add/delete/update/accounts</td>
</tr>
<tr>
<td>4</td>
<td>CreateUser.aspx</td>
<td>To create users</td>
</tr>
<tr>
<td>5</td>
<td>ManageTasks.aspx</td>
<td>To add/delete/update/create tasks</td>
</tr>
<tr>
<td>6</td>
<td>EMSTasks_create.aspx</td>
<td>To create EMS tasks</td>
</tr>
<tr>
<td>7</td>
<td>AllEvents.aspx</td>
<td>To View all EMS tasks by administrator</td>
</tr>
<tr>
<td>8</td>
<td>StudentWorkerHome.aspx</td>
<td>Student workers Home page</td>
</tr>
<tr>
<td>9</td>
<td>St_change_pwd.aspx</td>
<td>To Change Password for Student</td>
</tr>
<tr>
<td>10</td>
<td>St_view_tasks.aspx</td>
<td>To view all the Student worker tasks</td>
</tr>
<tr>
<td>11</td>
<td>Student_sd_opcltasks.aspx</td>
<td>Open/Close tasks Submit form</td>
</tr>
<tr>
<td>12</td>
<td>St_submit_checklist.aspx</td>
<td>Open/Close Check List tasks</td>
</tr>
<tr>
<td>13</td>
<td>St_submit_tour.aspx</td>
<td>Student Tour tasks submit form</td>
</tr>
</tbody>
</table>
4. TESTING AND EVALUATION

“Usability evaluation is any analysis of an empirical study of the usability of a prototype or system. The goal of the evaluation is to provide feedback in software development. Despite best efforts and sound practices, the original goals for the system may not in fact be achieved. More profoundly, the original project goals may have been successfully achieved, but they may turn out to be the wrong goals. Usability evaluation helps designers recognize that there is a problem, understand the problem and its underlying causes in the software, and plan changes to correct the problem” [Rosson 2002].

Testing is an important aspect of any project development. Task Management System is tested at various levels as shown below.

- Interface Level
- Database Level
- Implementation Level
- Usability Testing

The forms were used for data input. Testing was done to validate the forms in terms of proper data input. For example, a character input field can only accept characters as input and when the user enters a number in the character field, the form prompts a message to the user to enter only characters. While designing forms security checks are enforced by creating session variables for having a time-out session when the user is idle for an extended period on the forms while entering data.
4.1 Database Level

Sound database design is the most important issue in a database project. The database was tested for many scenarios. The database was normalized to reduce data redundancies. The queries were tested both at standalone level and after integrating with application for proper data retrieval, which was possible because of normalization and good entity relationships. The database was also tested when a user uploads or creates a new task, so that the task was inserted into the Task Table. The data types in the database were also checked for attributes to maintain consistency.

4.2 Implementation Level

The project was implemented simultaneously with the original paper-based model, which is currently used for some time to ensure proper back up. Any issues at this point were rolled back and corrected again in terms of data input or retrieval. At this level, the Task Management System was tested at a technical standpoint and any implementation constraints and errors were checked.

Testing was mainly performed to test functionality, performance, and accuracy. Tests were also performed on the interfaces at regular intervals to make sure all the scripts worked accurately. The following are the factors that were tested:

- Script accuracy – are the scripts that query and update the database accurate?
- Security – are all the interfaces properly authenticated to ensure security?

Each module was tested independently for data integrity, data consistency, consistency of data input and output formats and to check if it had achieved the
requirements. Standard coding guidelines were adopted along with proper documentation to facilitate easy maintenance.

4.3 Usability Testing

The application was tested for usability by getting feedback from the users. The usability scenarios included ease of use, performance, interface appearance and navigation. The users used the application simultaneously with paper-based system.

4.3.1 Usability testing goals

The main objective of the usability testing is to determine the ease of use of the system by the intended users. Feedback was obtained from users based on the following questions:

- Was there any difficulty in navigating through the system?
- Is there any difficulty in understanding the terminology in menus?
- Is the content in the menus legible?
- Are the colors in the menu appropriate and pleasing?
- Is there consistency in menu layout throughout the navigation?

4.4 Evaluation

The evaluation of TMS is mainly based on the user reaction and user satisfaction obtained from usability testing scenarios.

4.4.1 Evaluation criterion

Apart from the usability feedback, some of the evaluation criterions for TMS are listed below:

- The average time to complete each task
• The percentage of success in finding the information correctly
• The level of satisfaction achieved by the participant
• The users were also asked to evaluate the security of the TMS system. This includes trying to access the Web page, which the staff is not authorized to use.
• The ease of understanding the terminology used on the Web pages
• Completeness with which the site’s subject is treated
• Appearance of the site
• Consistency in Layout, Menus, and Visual Cues
5. RESULTS AND CONCLUSIONS

The Task Management System is designed to reduce overhead on the workers and the managers. There is a proper database for data archiving and generate reports. Reports can be generated for a day, on a weekly basis or even on a monthly basis. This gave an overview for the task managers and administrators to plan and to formulate future tasks effectively. New tasks are assigned by the manager at any instant without the need by the manager to tell the worker in person. Unlike the current system, the new system has proper data backup and archiving. There is proper division of labor between the workers, managers and administrators to perform duties.

The tasks are more organized and performed. It is easier to track the tasks because at every stage of performance there is a time stamp whether or not the task is duly performed on time. This reduced the scope for cheating and increased productivity and performance.

The major accomplishment of this project was complete automation of the existing paper based Task Management System. Proper User interface forms were created to input the tasks into the database. User interface forms were generated automatically for the Student workers to accomplish their duties in the form of tasks.
6. FUTURE WORK

At present, the TMS is developed for only certain facilities in the University Services Center. Some of the facilities such as Wells Fargo Bank are working independently with their own staff and tasks. These faculties can also be integrated into the existing system. The present system does not have any inventory functionality.

Student worker checks out equipment like Radios and Walkie-talkies, which are used in task fulfillment process. There is no process in place for inventory-tracking. This can be added as an additional module to TMS.
7. BIBLIOGRAPHY AND REFERENCES


[Date 1995] Date, C J. An Introduction to Database systems. Addison-Wesley Publications., 1995


APPENDIX A: REQUIREMENT ANALYSIS MEETINGS

Meetings with Mr. Paul Reynolds

Mr. Paul Reynolds is the Assistant Director for University Service Center.

Following are the excerpts from his meeting:

- Provided background information of the existing system needed for designing the new TMS; also explained the objectives that are expected of the new TMS.
- Gave a tour of the building and explained in detail how the current implementation of TSM.
- Explained the nature and scope of the project and drew parallels with existing system to the proposed system in designing the new TMS.
- Explained different types and number of users of the system.
- Discussed security issues and ways in making user’s accountable for work with the possibility of time stamp during logins.
- Meetings with Mr. Bob Martin, a systems programmer at University Services Center.
- Gave an overview of the existing system and explained the expected outcome from the new TMS from a technical standpoint.
- Discussed the possible technologies for implementing new TMS and was agreed upon to implement using .NET framework and SQL Server.
APPENDIX B: DATA DICTIONARY OF HIGH-LEVEL DFD

- **Actual Database Files** – The actual database files as stored inside the database system.
- **Create Email Request** – The request by ASP.NET engine to the mail server to send an email.
- **Database Server** – The backend database that will process all queries and give appropriate results.
- **Email** – The email that is sent to the user.
- **HTML document** – The resulting HTML document from server processing.
- **Host** – The whole system with the web server, the PHP scripts and the Mail Server.
- **Input** – The Universal Resource Locator of the host.
- **Mail Server** – This is a mail server of any standard mail system.
- **SQL Server** – This is the database management system going to be used in this project.
- **Output** – The various results that the user sees, prints, or saves.
- **Query Results** – The results from the queries.
- **Request** – The request the web server gives to the ASP.NET engine.
- **SQL Query** – Any legal query in Structured Query Language to the SQL server.
- **URL containing parameters** – The parameters that were input by user in the various forms.
• User – The different users of this system.

• Web Browser – The application that allows to access the system through the Internet.

• Web Server – The server side application that processes the scripts or compiled C# code. It is responsible for all client requests.
APPENDIX C: SYSTEM USABILITY FEEDBACK FORM

Answer any or all of the questions below. You are encouraged to include any questions, comments, or suggestions that you believe will enhance the capabilities of this system.

In general, how would you rate this system?

<table>
<thead>
<tr>
<th>Ease of navigation</th>
<th>Excellent</th>
<th>Good</th>
<th>Fair</th>
<th>Poor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web site structure</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System organization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consistency of Web page layout</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access speed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How do you compare this system with the existing paper-based system?

<table>
<thead>
<tr>
<th>Ease of use</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1 = Very easy to 4 = Very difficult)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time consumption</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1 = More time to 4 = Less time)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminology</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1 = Very similar to 4 = Very dissimilar)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1 = Very easy to 4 = Very difficult)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If given a choice, which system would you prefer?
Task management system
Paper-based system
Please suggest additional features you would like to see on the TMS Web site (please limit your suggestions or comments to 250 words).