

# UNIVERSITY OF MARYLAND UNIVERSITY COLLEGE (UMUC)

## SOFTWARE DESIGN DESCRIPTION PLAN

---

EMPLOYEE'S TIME MANAGEMENT SYSTEM VERSION 1.0

**Team Members:**

Justin Mullins  
Elvin Petrosy  
Ian Spooner  
Wendy Velasquez Ebanks

**Professor:** Christopher Howard

**CMSC 495:** Current Trends in Computer Science

April 9, 2017

## Revision History

Date	Version	Description	Author	Signature
04/03/2017	1.0	Document Creation, and drafted initial outline, and combining all teams collaboration sections	Wendy Velasquez Ebanks	WCVE

## Document Approval

Name	Date	Version Signed	Signature

## Table of Contents

1. Introduction.....	3
1.1 Purpose .....	3
1.2 Scope.....	3
1.3 Design Constraints.....	3
1.4 Document Outline .....	3
1.5 Definitions and Acronyms .....	4
2. References .....	4
3. Decomposition and Description .....	4
3.1 Module Decomposition.....	5
3.1.1 Topology Diagram .....	6
3.1.2 System Architecture .....	7
3.2 Data Description.....	8
3.3 Data Entity.....	8
3.2.2 Database Description .....	9
4. Class description .....	11
4.1 UML diagram of class structure .....	11
4.2 Class methods.....	12
5. Graphical User Interface .....	14
5.1 Login Screen.....	14
5.2 Timesheet.....	15
5.3 Employee Information.....	16
5.4 TimeSheet Submission/Management.....	17
5.5 User Privileges.....	18

## **1. Introduction**

The present document contains the Software Design Description Plan for the **Employee's Time Management System**.

### **1.1 Purpose**

The purpose of the Software Design Description Plan is to document and track the necessary information required to effectively define the approach used in user interface and internal structure design of the Employee Time Management System.

### **1.2 Scope**

The system is composed of a database linked to a software application, which will help to capture, store and later retrieve data inputted related to the employee's time sheet; it will also help in the generation of the needed reports and any other information relevant to time management of the employee.

### **1.3 Design Constraints**

The design constraints considered for the Employee Time System Management relate to the appropriate version of the software in order to have a good display of the Graphical User Interface (GUI) as well as the required speed connection in order to have a smooth connection display of the different features of the system. Another constraint noted in the design and implementation of the models described is the amount of time allocated for design and implementation of the system.

### **1.4 Document Outline**

The software design document is divided into five main sections with various subsections. The main sections of the Software Design Description Plan Document are:

1. Introduction
2. References

3. Decomposition and Description
4. Class Description.
5. Graphical User Interface

### **1.5 Definitions and Acronyms**

<b>Term</b>	<b>Definition</b>
<b>CSS</b>	Cascade Style Sheet
<b>DFD</b>	Data Flow Diagram
<b>ER</b>	Entity Relation
<b>ETMS</b>	Employee Time Management System
<b>GUI</b>	Graphical User Interface
<b>HTML</b>	Hypertext Markup Language

## **2. References**

- [1] IEEE Standard 1233-98: Guide for Developing System Requirements Specification
- [2] B. Schneiderman and C. Plaisant, "Designing the User Interface: Strategies for Effective Human-Computer Interaction," in Designing the User Interface: Strategies for Effective Human-Computer Interaction
- [3] Software Engineering Standards Committee of the IEEE Computer Society, "IEEE Recommended Practice for Software Design Descriptions", IEEE Std 1016-1998.
- [4] IEEE Standard 1058-1998: Standard for Software Project Management Plans
- [5] R. Darnall and J. Preston, Project management from simple to complex, 1st ed. The Saylor Academy, 2010.

## **3. Decomposition and Description**

This section describes the entities, their relationship and how they are classified by modules and by the data each one of them collects. The global data structure of this application is described over the class-diagrams and ER-diagrams and the different diagrams presented throughout the entire document. A database

will store all data retrieved through the graphical user interface (GUI), and will be organized on the classes defined in the software application later. Based on the acquired information the database will generate the necessary timesheet reports that will be provided to the end user (employee or supervisor). The database structure diagram depicts all the data involved in the most internal side of the application, and it is consider a data-centric design style. The classes that handle the data will be isolated and accessed by way of a model-view controller system. The database, which is the permanent storage of information, will be implemented by using MySQL database, and the software application that will communicate with this database will be developed in Java, JavaScript, HTML and CSS.

### 3.1 Module Decomposition

The system works mainly through a website as the main source to retrieve and generate the information. The graphic user interface (GUI) of a computer system comprises the interaction metaphors, images, and concepts used to convey function and meaning on the computer screen [2].

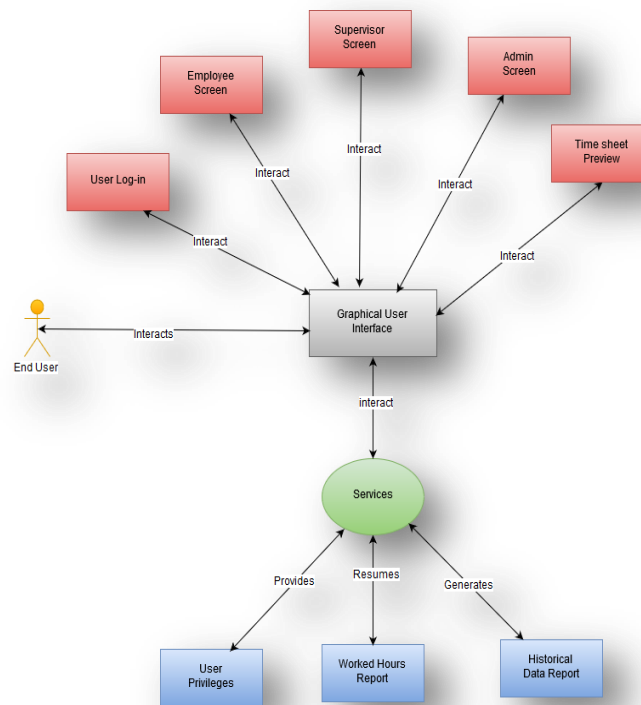


FIGURE 1 ARCHITECTURAL CONTEXT DIAGRAM – USER INTERFACE

The decomposition of the main areas of the system is as follows:

1. The Graphical user interface, which is the interface on which the user will enter the hours worked for a pre-set schedule agreed with the supervisor.
2. The description of the internal structure of the Database with its respective entities and relations defined.
3. The class structure that defines the record that will be stored by each Timesheet submission
4. The main methods that will help in the retrieval of information from the Web interface of the system.

### 3.1.1 Topology Diagram

The diagram below provides a topological illustration of the intended functionality of the Employee Time System Management with various system components that will be used.

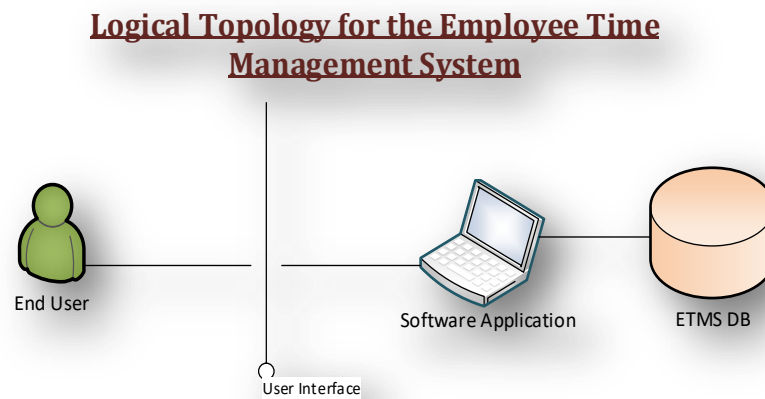


FIGURE 2 ETMS - LOGICAL TOPOLOGY

The Interaction of software components along with its responsibilities is explained below:

**HTTP/User Interface** - Hyper Text Transport Protocol is the communication protocol used to connect to servers on the World Wide Web. The primary function of HTTP is to establish a connection with a Web server and transmit HTML pages to the user's browser.

**JDBC/ Software Application**– Java Database Connectivity is an application program interface (API) specification for connecting programs written in

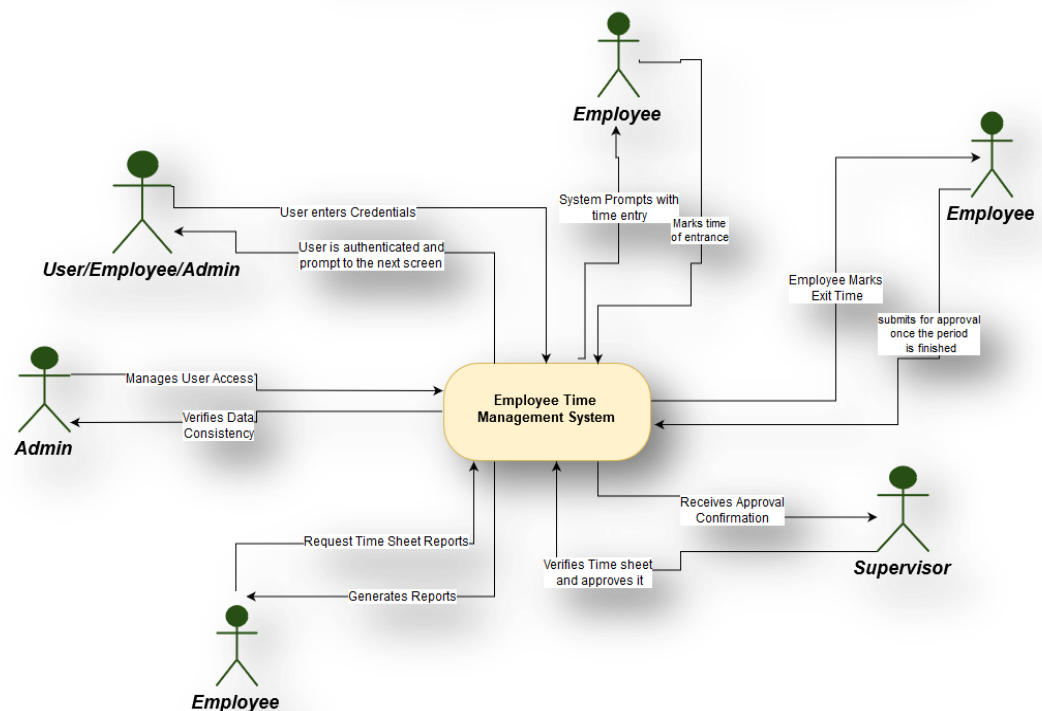
Java to the data in popular databases. The application program interface allows encode access request statements in structured query language (SQL) that are then passed to the program that manages the database. It returns the results through a similar interface.

**MySQL Database** – stores the employee's data, from the software application in relational format.

### 3.1.2 System Architecture

The system architecture is depicted in the following with the context diagram below, through the different processes that the system is required and intended to do. The activities will be processed and performed based on the type of user and the input provided.

**Use Case Diagram for Employee's Time Management System**



**FIGURE 3 ARCHITECTURAL CONTEXT DIAGRAM - USE CASE**



### 3.2 Data Description

This section provides the main components of the system represented in different diagrams to show case the relationship and traceability of the stated requirements, as shown in the data flow below:

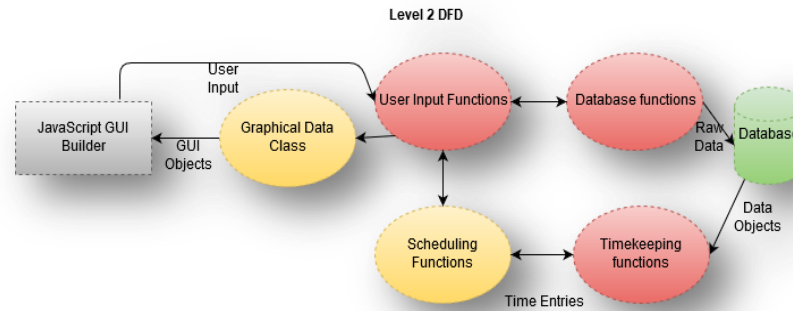


FIGURE 4 ETMS - DATA FLOW DIAGRAM LEVEL 2

### 3.3 Data Entity

The E-R diagram below displays the proposed structure of the database system and how it will store the most important information from the main sections that are required and reflecting the requirements previously specified.

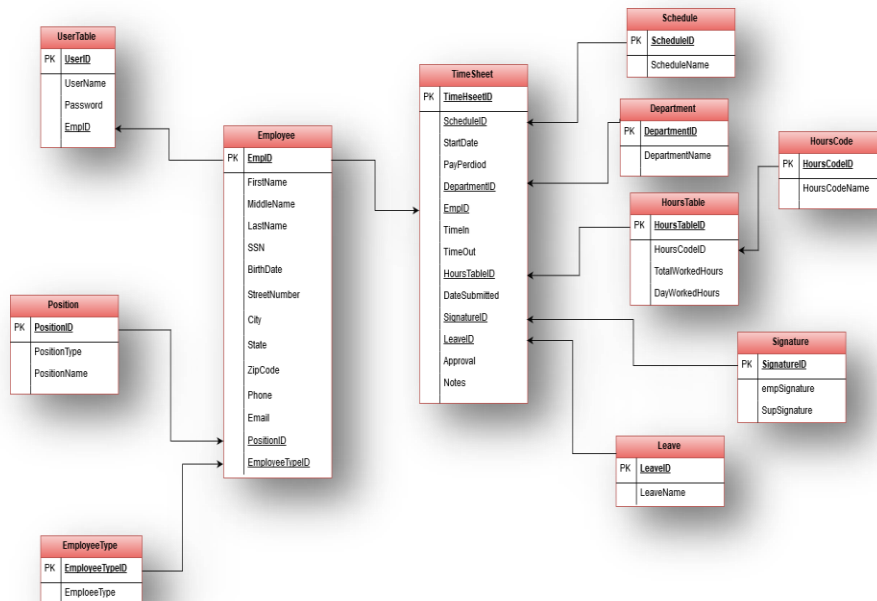


FIGURE 5 ETMS - ENTITY RELATIONSHIP DIAGRAM

### 3.2.2 Database Description

Each Entity represents a portion of information that will be relevant in the generation of the employee's time sheet.

**EMPLOYEE-** stores the personal information related to an employee, which later parts of it will be used in the time sheet when generating the different reports for the employee.

```
EMPLOYEE(  
  EmpID      Integer primary key auto_increment,  
  FirstName   VarChar(20),  
  MiddleName  VarChar(20),  
  LastName    VarChar(50),  
  SSN         VarChar(12),  
  StreetNumber VarChar(50),  
  City        VarChar(20),  
  ZipCode     VarChar(7),  
  Telephone   VarChar(16),  
  Email       VarChar(50),  
  Birthdate   Date,  
  PositionID  Integer references POSITION(PositionID));
```

**USERTABLE-** Creates the table that manages all users that have access to ETMS

```
USERTABLE(  
  UserID      Integer primary key auto_increment,  
  username    VarChar(20),  
  password    VarChar(16),  
  EmpID       Integer references EMPLOYEE(EmpID));
```

**POSITION-** Defines the type of position of the user

```
POSITION(  
  PositionID Integer primary key auto_increment,  
  PositionName VarChar(20),  
  PositionType VarChar(20));
```

**EMPLOYEEETYPE-** Defines the type of employee accessing the system either supervisor or non-supervisor

```
EMPLOYEEETYPE(  
  EmployeeTypeID Integer primary key auto_increment,  
  EmployeeType   VarChar(20));
```

**DEPARTMENT-** Stores the department on which an employee works

```
DEPARTMENT (
  DeptID      Integer primary key auto_increment,
  DepartmentName  VarChar(40));
```

**SCHEDULE**-Defines the different schedules that are available to the employee to arrange with the supervisor

```
SCHEDULE (
  ScheduleID  Integer primary key auto_increment,
  Schedulecategory  VarChar(40));
```

**HOURSCODE**- This table manages the type of hours an employee is working either regular or overtime

```
HOURSCODE (
  HoursCodeID Integer primary key auto_increment,
  HoursCodeName  VarChar(20));
```

**HOURLSTABLE**- This table is mainly formed by calculated fields that will depend on the type of hours coded by period

```
HOURLSTABLE (
  hoursTableID      Integer primary key auto_increment,
  hoursWorkedperDay Integer,
  hoursWorkedbyWeek Integer);
```

**SIGNATURETABLE**- This table holds the signature of the employee and supervisor as part of the Timesheet approval

```
SIGNATURETABLE (
  SignatureID Integer primary key auto_increment,
  payperiodEndDate  Date,
  empSignature      Varchar(30),
  empSuperSignature Varchar(30),
  EmpID Integer references EMPLOYEE(EmpID));
```

**LEAVE**-Defines the different types of leave to which the employee may qualify in a pay period

```
LEAVE (
  LeaveID Integer primary key auto_increment,
  LeaveName Varchar(30));
```

**TIMESHEET**- This table contains all the information related to the employee and the corresponding summary of times during a period of two weeks.

```

TIMESHEET (
TimeSheetID Integer primary key auto_increment,
StartDate Date,
TimeIn      date,
TimeOut     date,
EmpID  Integer references EMPLOYEE (EmpID) ,
DeptID  Integer references DEPARTMENT (DeptID) ,
ScheduleID  Integer references SCHEDULE (ScheduleID) ,
EmployeeTypeID (Integer references
EMPLOYEETYPE (EmployeeTypeID) ,
Payperiod  Integer,
LeaveID Integer references LEAVE (LeaveID) ,
EndDate    Date,
dateSubmitted    Date,
EmployeeSignature Varchar(30) ,
Approved    Varchar(30) ,
Notes Varchar(60) );

```

## 4. Class description

The model below covers the entities identified that will hold key data to perform a function in the ETMS, and they are referred as domain objects.

### 4.1 UML diagram of class structure

Initial UML Diagram

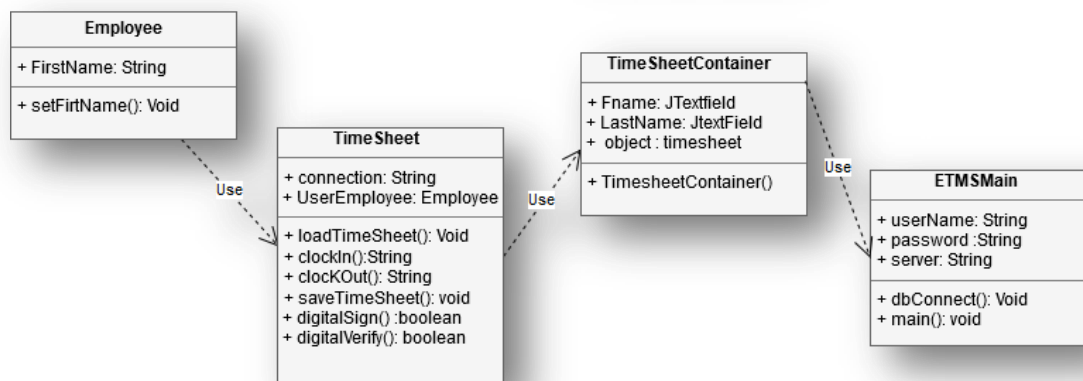


FIGURE 6 INITIAL UML DIAGRAM

## 4.2 Class methods

**dbConnect** - Connects the MySQL client to the server using a secure connection. It runs on each startup.

**Parameters:** username (String), password (String), server (String)

**Returns:** Connection

**loadTimesheet** - Pull data from the database to fill in the web GUI with timesheet data. This function might also be able to be used to pass this data to the digitalSign function.

**Parameters:** conn (Connection), payperiod (int) date (int)

**Returns:** String

**saveTimesheet** - Saves data in the web GUI timesheet to the database.

**Parameters:** conn (Connection)

**Returns:** void

**verifyTimesheet** -Verify data entered in web GUI timesheet is valid before saving to the database.

**Parameters:** Text field content to ensure the GUI is not sending empty values where is required to have one

**Returns:** boolean

**digitalSign** - Digitally sign the timesheet, first hash the contents, then encrypt using the user's private key.

**Parameters:** conn (Connection), timesheet (String), privateKey (PrivateKey)

**Returns:** digital signature (byte[])

**digitalVerify** - Verify the digital signature is legitimate using the user's public key to decrypt the digital signature and match the hash of the content.

**Parameters:** conn (Connection), digital signature (byte[]), public key (PublicKey), timesheet (String)

**Returns:** Boolean

**clockIn** - adds a timestamp for the user at the current time.

**Parameters:** timesheet (int), conn (Connection)

**Returns:** query, (String)

**clockOut** - adds a timestamp for the user at the current time.

**Parameters:** timesheet (int), conn (Connection)

**Returns:** query, (String)

**printSummary** - This one should display the summary of hours worked by the employee

**Parameters:** (it should be some type of record)

**Returns:** String

**printDailyReport** - provides a report of the hours worked by day.

**Parameters:** (Query type)

**Returns:** String

**printWeeklyReport** - provides a report of hours worked by week

**Parameters:** (Query Type)

**Returns:** String

**printReport** - it should provide the cumulative timesheets for the entire year

**Parameters:** (query type)

**Returns:** String

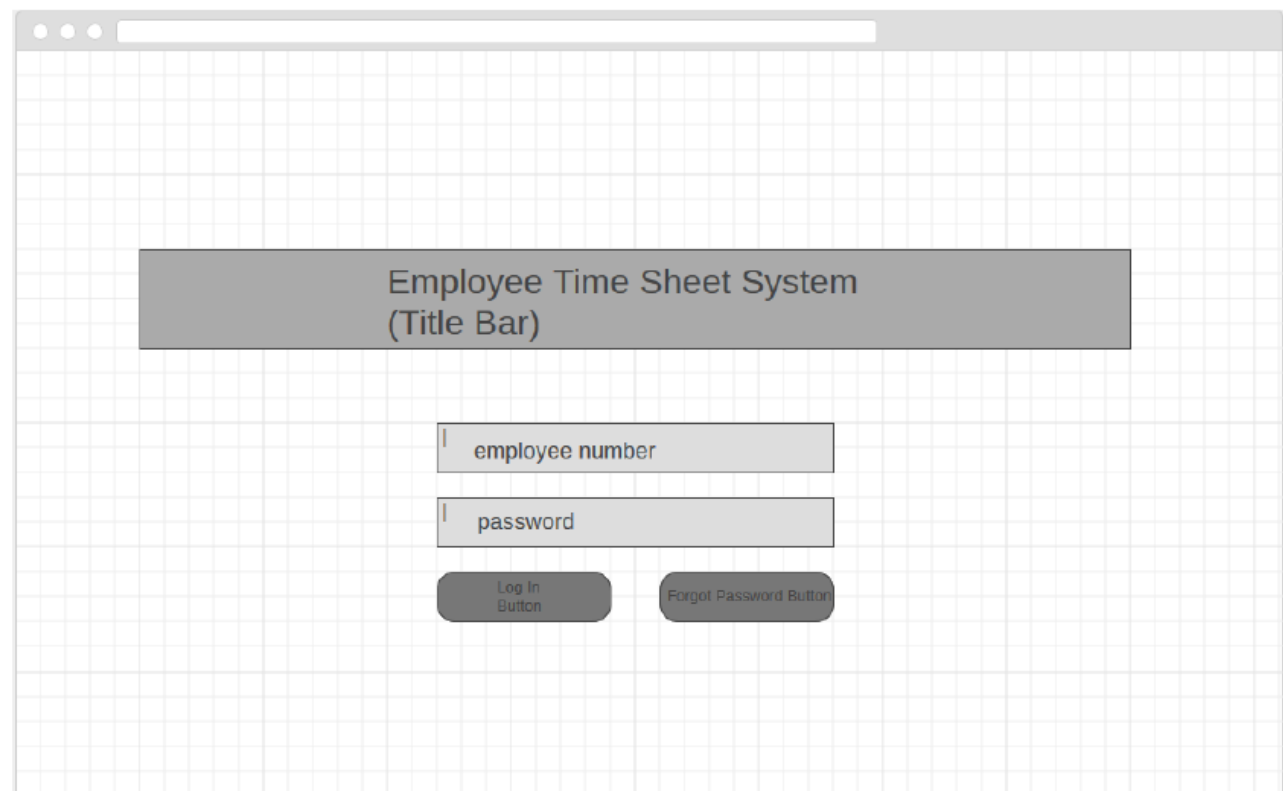
## 5. Graphical User Interface

This section is specifically for those who will interact with the Employee Time management system (users, supervisors, and administrators). They all will be able to use the system through a graphical user interface (GUI); access to the GUI does not require any external interface other than a keyboard and mouse.

### 5.1 Login Screen

Category	Use Case ID	Req. ID	Test Case ID	Req. Description	Test Case Description	Pass /Fail
UI	SRS 9.2.1-1	IR-1	ETMS-IR-01	The user shall be authenticated by inputting its credentials when log-in to the system	Validate that user is authenticated when logging the system	

Main Login Screen:



The image shows a mockup of the Main Login Screen for the Employee Time Sheet System. The screen is displayed within a window frame with a title bar. The title bar contains the text "Employee Time Sheet System (Title Bar)". Below the title bar, there are two input fields: "employee number" and "password". Below these fields are two buttons: "Log In Button" and "Forgot Password Button". The background of the screen is a light gray grid.

FIGURE 7 ETMS - LOGIN SCREEN

## 5.2 Timesheet

Category	Use Case ID	Req. ID	Test Case ID	Req. Description	Test Case Description	Pass/Fail
UI	SRS 9.2.1-2	IR-2	ETMS-IR-02	The user shall be able to input his/her hours according to the schedule previously set and submit them for review.	Validate that the user inputs his/her hours and submit his/her timesheet	

Timesheet:

Main Title Banner

Welcome, [USER] Sign Out Button

Account Timesheet Manage

Timesheet entry

Timesheet Status:

Validate/Save Submit

FIGURE 8 ETMS - TIMESHEET INTERFACE



### 5.3 Employee Information

Category	Use Case ID	Req. ID	Test Case ID	Req. Description	Test Case Description	Pass/Fail
UI	SRS 9.2.1-4	IR-4	ETMS-IR-04	The system shall request the necessary information from the user in order to generate historical and current reports for the user.	Assess completeness of required user data in the DB.	

Account:

Main Title Banner

Welcome, [USER] Sign Out Button

Account Timesheet Manage

[First Name] [MI] [Last Name]  
[employee ID]

Street Address:  State:

City:  Zipcode:

Phone:  New Password:

Email:  Confirm Pass:

Save Discard

FIGURE 9 ETMS - EMPLOYEE PERSONAL INFO

## 5.4 TimeSheet Submission/Management

Category	Use Case ID	Req. ID	Test Case ID	Req. Description	Test Case Description	Pass/Fail
UI	SRS 9.2.2-2	OR-2	ETMS-OR-02	The system shall be able to provide a confirmation of work hour's submission.	Evaluates that a confirmation screen is presented after submission	

Dashboard:

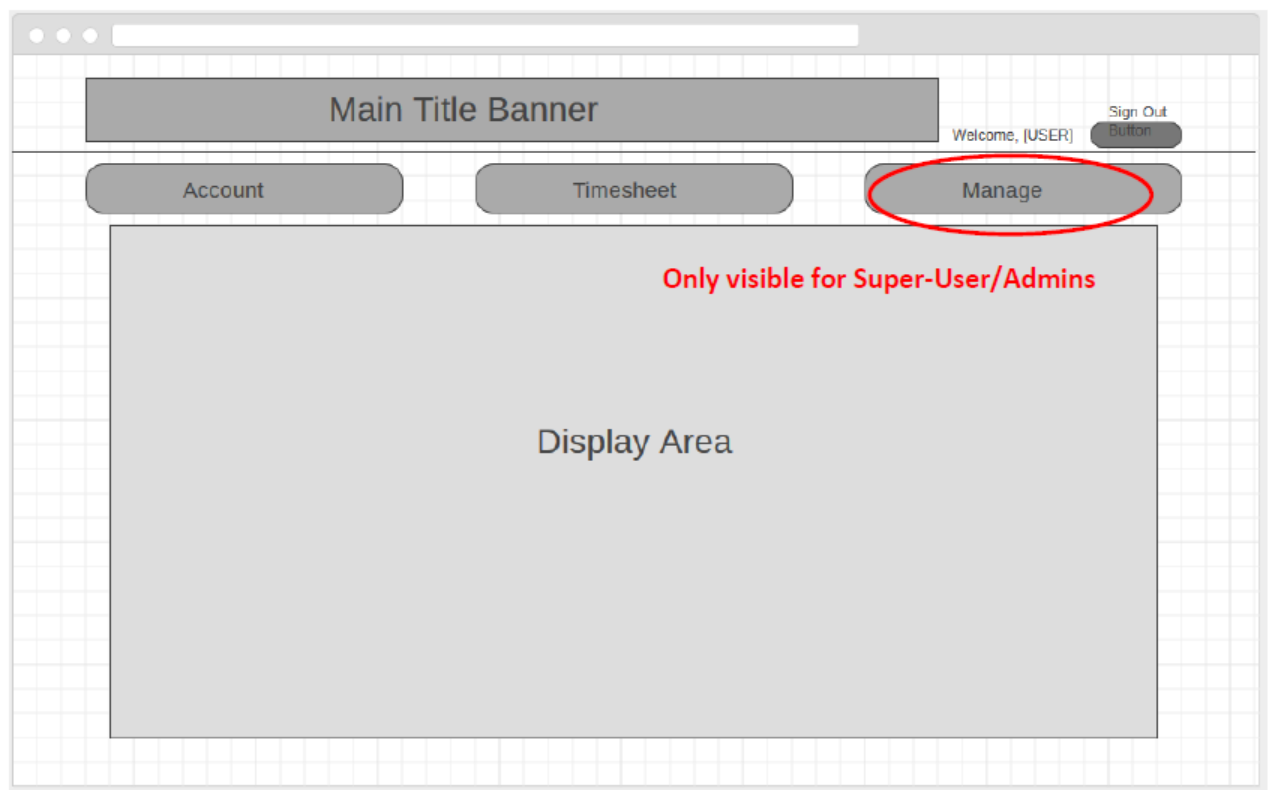


FIGURE 10 ETMS - MANAGEMENT DASHBOARD

## 5.5 User Privileges

Category	Use Case ID	Req. ID	Test Case ID	Req. Description	Test Case Description	Pass/Fail
Services	SRS 9.3.1-6	SR-6	ETMS-SR-06	The system shall assure a specific privilege level can only access data required for the job role.	Validates that a user is only able to access data related to their job role.	

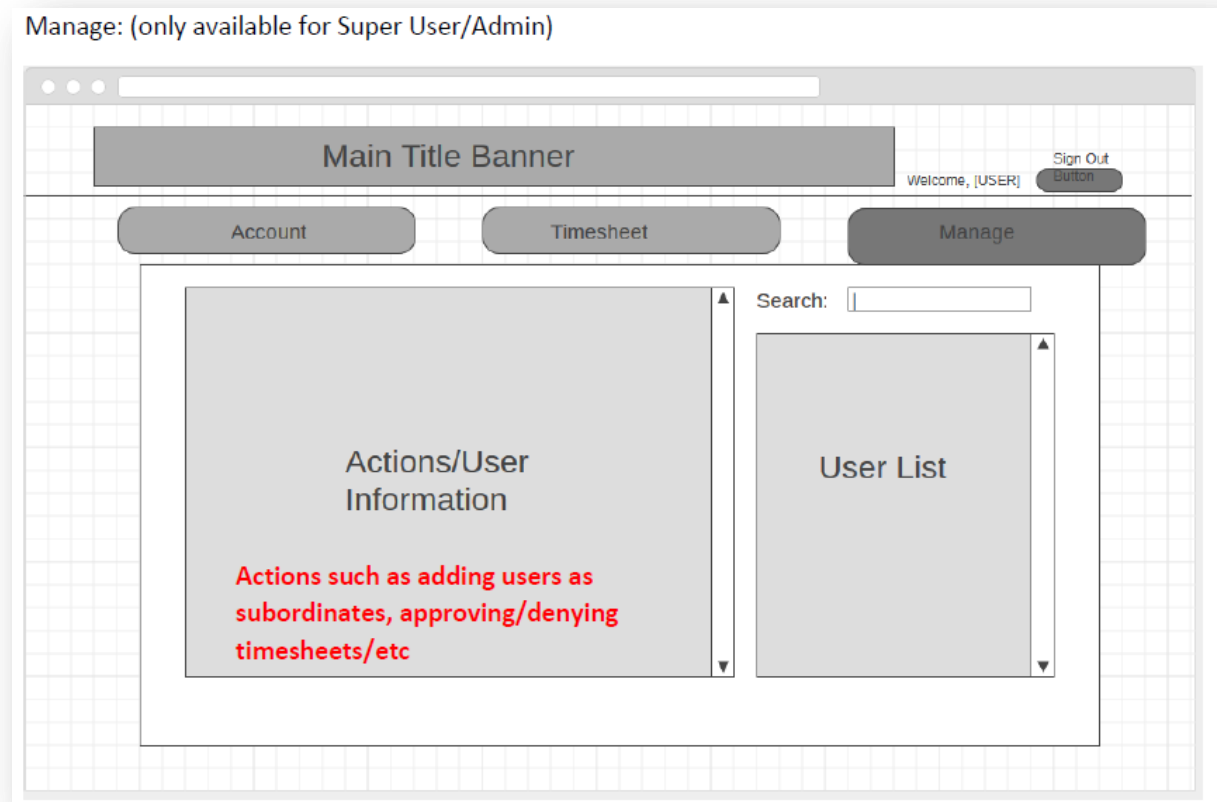


FIGURE 11 ETMS - USER PRIVILEGES