

# Escaleta-Teclado GC

# Especificación de realización de casos de uso

**Version <5.0>**

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| **15/01/2001** | **1.0** | **First Version** | **Sandra Lee** |
| **15/01/2001** | **2.0** | **Added Objects Diagrams** | **Sandra Lee** |
| **06/02/2001** | **3.0** | **Modified content to fit the new Use-Case Diagrams** | **Sandra Lee** |
| **30/03/2001** | **4.0** | **Adding Validate Timestamps Functionality** | **James Prevost** |
| **05/04/2001** | **5.0** | **Adding Classes: ShowWindow and CloseWindow.**  **Revised Objects Diagrams.** | **Sandra Lee** |

**Preface**

The following case study has been modified from its original content. The case study is meant to be used as a starting point to help you understand how to use the artifact. Thus, information has been shrunk to avoid navigating an enormous document (in size and pages).

You can also refer to the related template (in HTML format or WORD format) in the UPEDU Artifacts Templates Analysis & Design Section.

Regards,

Unified Process for Education Team

# Table of Contents

1. [Introduction 8](#_bookmark0)
   1. [Purpose 8](#_bookmark0)
   2. [Scope 8](#_bookmark0)
   3. [Definitions, Acronyms, and Abbreviations 8](#_bookmark0)
   4. [References 8](#_bookmark0)
   5. [Overview 8](#_bookmark0)
2. [USE CASE <Load MCM > 9](#_bookmark1)
   1. [Brief Description 9](#_bookmark1)
   2. [Flow Events 9](#_bookmark1)
   3. [Interaction Diagrams 9](#_bookmark1)
      1. [Sequence Diagrams 10](#_bookmark2)

[1.3.2 Collaboration Diagrams 10](#_bookmark2)

* 1. [Participating Objects 11](#_bookmark3)
  2. [Object Diagram 11](#_bookmark3)

1. [USE CASE <Load DCM > 12](#_bookmark4)
   1. [Brief Description 12](#_bookmark4)
   2. [Flow Events 12](#_bookmark4)
   3. [Interaction Diagrams 13](#_bookmark5)
      1. [Sequence Diagrams 13](#_bookmark5)
      2. [Collaboration Diagrams 13](#_bookmark5)
   4. [Participating Objects 13](#_bookmark5)
   5. [Object Diagram 14](#_bookmark6)
2. [USE CASE <Login> 14](#_bookmark6)
   1. [Brief Description 14](#_bookmark6)
   2. [Flow Events 15](#_bookmark7)
   3. [Interaction Diagrams 15](#_bookmark7)
      1. [Sequence Diagrams 15](#_bookmark7)
      2. [Collaboration Diagrams 17](#_bookmark9)
   4. [Participating Objects 19](#_bookmark11)
   5. [Object Diagram 19](#_bookmark11)
3. [USE CASE <Logout> 20](#_bookmark12)
   1. [Brief Description 20](#_bookmark12)
   2. [Flow Events 21](#_bookmark13)
   3. [Interaction Diagrams 21](#_bookmark13)
      1. [Sequence Diagrams 21](#_bookmark13)
      2. [Collaboration Diagrams 23](#_bookmark15)
   4. [Participating Objects 24](#_bookmark16)
   5. [Object Diagram 25](#_bookmark17)
4. [USE CASE <Show TMT Status > 26](#_bookmark18)
   1. [Brief Description 26](#_bookmark18)
   2. [Flow Events 27](#_bookmark19)
   3. [Interaction Diagrams 27](#_bookmark19)
      1. [Sequence Diagrams 27](#_bookmark19)
      2. [Collaboration Diagrams 28](#_bookmark20)
   4. [Participating Objects 28](#_bookmark20)
   5. [Object Diagram 28](#_bookmark20)
5. [USE CASE <Identify Errors > 29](#_bookmark21)
   1. [Brief Description 29](#_bookmark21)
   2. [Flow Events 30](#_bookmark22)
   3. [Interaction Diagrams 30](#_bookmark22)
      1. [Sequence Diagrams 30](#_bookmark22)
      2. [Collaboration Diagrams 31](#_bookmark23)
   4. [Participating Objects 31](#_bookmark23)
   5. [Object Diagram 31](#_bookmark23)
6. [USE CASE < Correct Errors > 32](#_bookmark24)
   1. [Brief Description 32](#_bookmark24)
   2. [Flow Events 33](#_bookmark25)
   3. [Interaction Diagrams 33](#_bookmark25)
      1. [Sequence Diagrams 33](#_bookmark25)
      2. [Collaboration Diagrams 34](#_bookmark26)
   4. [Participating Objects 34](#_bookmark26)
   5. [Object Diagram 34](#_bookmark26)
7. [USE CASE < Show TMT Window > 35](#_bookmark27)
   1. [Brief Description 35](#_bookmark27)
   2. [Flow Events 36](#_bookmark28)
   3. [Interaction Diagrams 36](#_bookmark28)
      1. [Sequence Diagrams 36](#_bookmark28)
      2. [Collaboration Diagrams 38](#_bookmark30)
   4. [Participating Objects 39](#_bookmark31)
   5. [Object Diagram 39](#_bookmark31)
8. [USE CASE < Close TMT Window > 40](#_bookmark32)
   1. [Brief Description 40](#_bookmark32)
   2. [Flow Events 41](#_bookmark33)
   3. [Interaction Diagrams 41](#_bookmark33)
      1. [Sequence Diagrams 41](#_bookmark33)
      2. [Collaboration Diagrams 42](#_bookmark34)
   4. [Participating Objects 43](#_bookmark35)
   5. [Object Diagram 43](#_bookmark35)

**Figures**

[Figure 1 : Load MCM 9](#_bookmark1)

[Figure 2 : Sequence Diagram : Load MCM 10](#_bookmark2)

[Figure 3 : Collaboration Diagram : Load MCM 10](#_bookmark2)

[Figure 4 : Object Diagrams : Load MCM 11](#_bookmark3)

[Figure 5 : Load DCM 12](#_bookmark4)

[Figure 6 : Sequence Diagram : Load DCM 13](#_bookmark5)

[Figure 7 : Collaboration Diagram : Load DCM 13](#_bookmark5)

[Figure 8 : Class Diagram : Load DCM 14](#_bookmark6)

[Figure 9 : Sequence Diagram : Login(Administrator) 15](#_bookmark7)

[Figure 10 : Sequence Diagram : Login(Developer) 16](#_bookmark8)

[Figure 11 : Collaboration Diagram : Login (Administrator) 17](#_bookmark9)

[Figure 12 : Collaboration Diagram : Login (Developer) 18](#_bookmark10)

[Figure 13 : Object Diagram : Login 19](#_bookmark11)

[Figure 14 : Logout 20](#_bookmark12)

[Figure 15 : Sequence Diagram : Logout (Administrator) 21](#_bookmark13)

[Figure 16 : Sequence Diagram : Logout (Developer) 22](#_bookmark14)

[Figure 17 : Collaboration Diagram : Logout (Administrator) 23](#_bookmark15)

[Figure 18 : Collaboration Diagram : Logout (Developer) 24](#_bookmark16)

[Figure 19 : Object Diagrams : Logout 25](#_bookmark17)

[Figure 20 : Show TMT Status 26](#_bookmark18)

[Figure 21 : Sequence Diagram : Show TMT Status 27](#_bookmark19)

[Figure 22 : Collaboration Diagram : Show TMT Status 28](#_bookmark20)

[Figure 23 : Object Diagrams : Show TMT Status 28](#_bookmark20)

[Figure 24 : Identify Errors 29](#_bookmark21)

[Figure 25 : Sequence Diagram : Identify Errors 30](#_bookmark22)

[Figure 26 : Collaboration Diagram : Identify Errors 31](#_bookmark23)

[Figure 27 : Object Diagrams : Identify Errors 31](#_bookmark23)

[Figure 28 : Correct Errors 32](#_bookmark24)

[Figure 29 : Sequence Diagram : Correct Errors 33](#_bookmark25)

[Figure 30 : Collaboration Diagram : Correct Errors 34](#_bookmark26)

[Figure 31 : Object Diagrams : Correct Errors 34](#_bookmark26)

[Figure 32 : Show TMT Window 35](#_bookmark27)

[Figure 33 : Sequence Diagram : Show TMT Window (Administrator) 36](#_bookmark28)

[Figure 34 : Sequence Diagram : Show TMT Window (Developer) 37](#_bookmark29)

[Figure 35 : Collaboration Diagram : Show TMT Window (Administrator) 38](#_bookmark30)

[Figure 36 : Collaboration Diagram : Show TMT Window (Developer) 38](#_bookmark30)

[Figure 37 : Object Diagrams : Show TMT Window 39](#_bookmark31)

[Figure 38 : Close TMT Window 40](#_bookmark32)

[Figure 39 : Sequence Diagram : Close TMT Window (Administrator) 41](#_bookmark33)

[Figure 40 : Sequence Diagram : Close TMT Window (Developer) 41](#_bookmark33)

[Figure 41 : Collaboration Diagram : Close TMT Window (Administrator) 42](#_bookmark34)

[Figure 42 : Collaboration Diagram : Close TMT Window (Developer) 42](#_bookmark34)

[Figure 43 : Object Diagrams : Close TMT Window 43](#_bookmark35)

# Use-Case-Realization Specification

### Introduction

* 1. **Purpose**

This document provides a comprehensive overview of the system, using a number of different diagrams for representing the system functions.

* 1. **Scope**

The Time Monitoring Tool system allows developers working within a defined software development process to record the time spent on the various activities, in a database. The TMT will also allow a manager to derive analyses and produce reports based on the data entered in the system. This Use Case Realization document provides an overview of the use cases developed in Time Monitoring Tool.

* 1. **Definitions, Acronyms, and Abbreviations**

See Glossary, document upedu\_ex\_gloss.doc

* 1. **References**
  2. **Overview**

1. **TMT - Glossary**
2. **TMT - Use Case Specification**
3. **TMT - Supplementary Specification**
4. **TMT - Iteration Plan**

The sections of the Use-Case Realization document describes use cases in terms of their flow of events, participant objects and corresponding diagrams.

### USE CASE <Load MCM >

* 1. **Brief Description**

This Use-Case defines how the MCM is loaded allowing the Administrator to manage the application through the Internet Browser.

Administrator

(f rom Mngmt - Client/Serv er)

Load MCM

**Figure 1 : Load MCM**

* 1. **Flow Events**

Upon logging in the system, the Administrator implicitly makes a Load MCM query which is received by the Browser, then executed.

* 1. **Interaction Diagrams**
     + **The Administrator launches the Load MCM query.**
     + **The Internet Browser receives the query and loads the module.**

*1.3.1. Sequence Diagrams*

This Sequence Diagram shows Actors and Objects messages exchange in the Use-Case < Load MCM >.

: Administrator

Launches Load Query

Loads MCM

Netscape Browser

**Figure 2 : Sequence Diagram : Load MCM**

*1.3.2 Collaboration Diagrams*

This Collaboration Diagram shows the static structure of the Use-Case < Load MCM >.

2: Loads MCM

1: Load Query

Netscape Browser

: Administrator

**Figure 3 : Collaboration Diagram : Load MCM**

* 1. **Participating Objects**

The following objects collaborate and define the Use-Case < Load MCM > Behavior:

|  |  |
| --- | --- |
| **Netscape Browser** | **This object represents the visible part of the application and allows**  **the Administrator to load his module.** |

* 1. **Object Diagram**

The following Object Diagram shows the relations and constraints between Classes and Objects involved in the Use-Case.



Observable

(f rom Serv er Package)

<<Interface>>

*Observer*

*(from Server Package)*

<<implements>>

<<Singleton>>

TMTServer

(f rom Serv er Package)

AdministratorObserv er

(f rom Serv er Package)

**Figure 4 : Object Diagrams : Load MCM**

### USE CASE <Load DCM >

* 1. **Brief Description**

This Use-Case defines how the DCM is loaded allowing the Developer to timestamp his work hours through the Internet Browser.

Developer

(f rom Mngmt - Client/Serv er)

Load DCM

**Figure 5 : Load DCM**

* 1. **Flow Events**

Upon logging in the system, the Developer implicitly makes a Load DCM query which is received by the Browser, then executed.

* 1. **Interaction Diagrams**
     + **The Developer launches the Load DCM query.**
     + **The Internet Browser receives the query and loads the module.**
     1. *Sequence Diagrams*

This Sequence Diagram shows Actors and Objects messages exchange in the Use-Case < Load DCM >.

: Developer

Launches Load Query

Loads DCM

Netscape Browser

**Figure 6 : Sequence Diagram : Load DCM**

* + 1. *Collaboration Diagrams*

This Collaboration Diagram shows the static structure of the Use-Case -<Load DCM>.

2: Loads DCM

1: Load Query

Netscape Browser

: Developer

**Figure 7 : Collaboration Diagram : Load DCM**

* 1. **Participating Objects**

The following objects collaborate and define the Use-Case < Load DCM > Behavior:

|  |  |
| --- | --- |
| **Netscape**  **Browser** | **This object represents the visible part of the application and allows the**  **Developer to load his module.** |

* 1. **Object Diagram**

The following Object Diagram shows the relations and constraints between Classes and Objects involved in the Use-Case.



Observable

(f rom Serv er Package)

<<Interface>>

*Observer*

*(from Server Package)*

<<implements>>

<<Singleton>>

TMTServer

(f rom Serv er Package)

Dev eloperObserv er

(f rom Serv er Package)

**Figure 8 : Class Diagram : Load DCM**

### USE CASE <Login>

* 1. **Brief Description**

This Use-Case defines how users are logged into the system and get access to their respective functionalities.

Administrator

(f rom Mngmt - Client/Serv er)

Login

Developer

(f rom Mngmt - Client/Serv er)

Figure 8: Login

* 1. **Flow Events**

The user provides his username and password and submits the form. Data is validated and login process is activated.

* 1. **Interaction Diagrams**
     + **The user enters his username and password and submits the data**
     + **Query is received by the MCM or DCM and transmitted to the SM**
     + **The SM validates, executes and accesses the database for login confirmation.**
     + **The DBMS returns confirmation to the SM which transmits it to the MCM.**
     1. *Sequence Diagrams*

These Sequence Diagrams show Actors and Objects messages exchange in the Use-Case < Login>.

: Administrator

: DBMS

Launches Query

Transmits Query

Validate Query

Execute Query

Access Database

Confirmation

Return Results

SM

MCM

**Figure 9 : Sequence Diagram : Login(Administrator)**



: Developer

: DBMS

Launches Query

Transmits Query

Validate Query

Execute Query

Access Database

Confirmation

Return Results

SM

DCM

**Figure 10 : Sequence Diagram : Login(Developer)**

* + 1. *Collaboration Diagrams*

This Collaboration Diagram shows the static structure of the Use-Case <login>.

: Administrator

1: Give PW and Name

4: Execute Query

3: Validate Query

2: Transmit Query

MCM

SM

7: Return Results

5: Access Database

6: Confirmation

: DBMS

**Figure 11 : Collaboration Diagram : Login (Administrator)**



: Developer

: DBMS

Launches Query

Transmits Query

Validate Query

Execute Query

Access Database

Confirmation

Return Results

SM

DCM

**Figure 12 : Collaboration Diagram : Login (Developer)**

* 1. **Participating Objects**

The following objects collaborate and define the Use-Case <login> :

|  |  |
| --- | --- |
| **MCM** | **This object represents the visible part of the application and allows the**  **Administrator to login to the system.** |
| **DCM** | **This object represents the visible part of the application and allows the**  **Developer to login to the system.** |
| **SM** | **This object executes and validates the Login query by communicating**  **with the Database.** |

* 1. **Object Diagram**

The following Object Diagram shows the relations and constraints between Classes and Objects involved in the Use-Case.



Observable

(f rom Serv er Package)

<<Interface>>

*Observer*

*(from Server Package)*

<<implements>>

<<implements>>

<<Singleton>>

TMTServer

(f rom Serv er Package)

Dev eloperObserv er

(f rom Serv er Package)

AdministratorObserv er

(f rom Serv er Package)

**Figure 13 : Object Diagram : Login**

### USE CASE <Logout>

* 1. **Brief Description**

This Use-Case defines how users are logged out from system and exit the application normally.

Administrator

(f rom Mngmt - Client/Serv er)

Developer

(f rom Mngmt - Client/Serv er)

Logout

**Figure 14 : Logout**

* 1. **Flow Events**

The user exits the application by using the appropriate End Session button. Query is validated and logout process is activated.

* 1. **Interaction Diagrams**
     + **The user clicks the End Session Button**
     + **Query is received by the MCM or DCM and transmitted to the SM**
     + **The SM validates, executes and accesses the database for logout confirmation.**
     + **The DBMS returns confirmation to the SM which transmits it to the MCM.**
     1. *Sequence Diagrams*

These Sequence Diagrams show Actors and Objects messages exchange in the Use-Case < Logout>.

: Administrator

: DBMS

Logout Query

Transmits Query

Validate Query

Execute Query

Access Database

Confirmation

Return Results

SM

MCM

**Figure 15 : Sequence Diagram : Logout (Administrator)**

 

**Figure 16 : Sequence Diagram : Logout (Developer)**

: Developer

: DBMS

Logout Query

Transmits Query

Validate Query

Execute Query

Access Database

Confirmation

Return Results

SM

DCM

* + 1. *Collaboration Diagrams*

This Collaboration Diagram shows the static structure of the Use-Case <Logout>

: Administrator

1: Logout Query

6: Execute Query

4: Validate Query

2: Transmit Query

MCM

SM

7: Return Results

3: Access Database

5: confirmation

: DBMS

**Figure 17 : Collaboration Diagram : Logout (Administrator)**

: Developer

1: Logout Query

6: Execute Query

4: Validate Query

2: Transmit Query

DCM

SM

7: Return Results

3: Access Database

5: Confirmation

: DBMS

**Figure 18 : Collaboration Diagram : Logout (Developer)**

* 1. **Participating Objects**

The following objects collaborate and define the Use-Case <Logout> :

|  |  |
| --- | --- |
| **MCM** | **This object represents the visible part of the application and allows the**  **Administrator to logout from the system.** |
| **DCM** | **This object represents the visible part of the application and allows the**  **Developer to logout from the system.** |
| **SM** | **This object executes and validates the Logout query by communicating**  **with the Database.** |

* 1. **Object Diagram**

The following Object Diagram shows the relations and constraints between Classes and Objects involved in the Use-Case.



Observable

(f rom Serv er Package)

<<Interface>>

*Observer*

*(from Server Package)*

<<implements>>

<<implements>>

<<Singleton>>

TMTServer

(f rom Serv er Package)

Dev eloperObserv er

(f rom Serv er Package)

AdministratorObserv er

(f rom Serv er Package)

**Figure 19 : Object Diagrams : Logout**

### USE CASE <Show TMT Status >

* 1. **Brief Description**

This Use-Case defines how the TMT Status is displayed to the user.

DBMS

(f rom Mngmt - Client/Serv er)

Access Database

(f rom Mngmt - Client/Serv er)

Show TMT Status

**Figure 20 : Show TMT Status**

* 1. **Flow Events**

The TMT status is constantly verified by the Server Module. If the status is abnormal, error details and comments will be displayed, else the normal TMT Window content is displayed.

* 1. **Interaction Diagrams**
* **The SM verifies the status**
* **The SM shows the status**
  + 1. *Sequence Diagrams*

These Sequence Diagrams show Actors and Objects messages exchange in the Use-Case < Show TMT Status>.

Verify Status

Show Status

If Error: Show Error + Comments

SM

**Figure 21 : Sequence Diagram : Show TMT Status**

* + 1. *Collaboration Diagrams*

This Collaboration Diagram shows the static structure of the Use-Case <Show TMT Status>.

2: Show Status

1: Verify Status

SM

**Figure 22 : Collaboration Diagram : Show TMT Status**

* 1. **Participating Objects**

The following objects collaborate and define the Use-Case < Show TMT Status > :

|  |  |
| --- | --- |
| **SM** | **This object interacts with the database and the Internet Browser in**  **order to determine the current status of the TMT.** |

* 1. **Object Diagram**

The following Object Diagram shows the relations and constraints between Classes and Objects involved in the Use-Case.



Observ able

*Observer*

<<Singleton>>

TMTServer

**Figure 23 : Object Diagrams : Show TMT Status**

### USE CASE <Identify Errors >

* 1. **Brief Description**

This Use-Case defines the TMT status error identification process

DBMS

(f rom Mngmt - Client/Serv er)

Access Database

(f rom Mngmt - Client/Serv er)

Show TMT Status

<<include>>

<<extend>>

Identify Errors

Correct Errors

**Figure 24 : Identify Errors**

* 1. **Flow Events**

Upon each TMT status verification, the SM must detect any error presence (Execution, Communication, Validation, others).

* 1. **Interaction Diagrams**
* **The SM verifies the TMT status**
* **The SM identifies errors, if any**
  + 1. *Sequence Diagrams*

These Sequence Diagrams show Actors and Objects messages exchange in the Use-Case < Identify Errors>.

Verify TMT Status

Execution Communication

Identify Errors Validation

Others...

SM

**Figure 25 : Sequence Diagram : Identify Errors**

* + 1. *Collaboration Diagrams*

This Collaboration Diagram shows the static structure of the Use-Case <Identify Errors>.

2: Identify Errors

1: Verify TMT Status

Execution Communication Validation Other..

SM

**Figure 26 : Collaboration Diagram : Identify Errors**

* 1. **Participating Objects**

The following objects collaborate and define the Use-Case < Identify Errors> :

|  |  |
| --- | --- |
| **SM** | **This object interacts with the database and the Internet Browser in**  **order to determine and detect error presence.** |

* 1. **Object Diagram**

The following Object Diagram shows the relations and constraints between Classes and Objects involved in the Use-Case.



Observ able

*Observer*

<<Singleton>>

TMTServer

**Figure 27 : Object Diagrams : Identify Errors**

1. **USE CASE < Correct Errors >**
   1. **Brief Description**

After Identifying errors, the SM must try to automatically correct the error, if possible. The error correction must remain transparent to the user and the application must follow its course of actions normally.

DBMS

(f rom Mngmt - Client/Serv er)

Access Database

(f rom Mngmt - Client/Serv er)

Show TMT Status

<<include>>

<<extend>>

Identify Errors

Correct Errors

**Figure 28 : Correct Errors**

* 1. **Flow Events**

Upon each TMT status error identification, the SM must try to correct the error.

* 1. **Interaction Diagrams**
* **The SM gets the error information from its identification**
* **The SM tries to correct the error**
* **If the error is corrected, the SM restores the TMT last good status**
* **If the SM cannot correct the error, a special TMT status error message is displayed with appropriate comments**
  + 1. *Sequence Diagrams*

These Sequence Diagrams show Actors and Objects messages exchange in the Use-Case < Correct Errors>.

Identify Errors

Correct Errors

Restore Last Good Status

SM

**Figure 29 : Sequence Diagram : Correct Errors**

* + 1. *Collaboration Diagrams*

This Collaboration Diagram shows the static structure of the Use-Case <Correct Errors>.

3: Restore Last Good Status

2: Correct Errors

1: Identify Errors

SM

**Figure 30 : Collaboration Diagram : Correct Errors**

* 1. **Participating Objects**

The following objects collaborate and define the Use-Case < Correct Errors> :

|  |  |
| --- | --- |
| **SM** | **This object interacts with the database and the Internet Browser in**  **order to correct the errors.** |

* 1. **Object Diagram**

The following Object Diagram shows the relations and constraints between Classes and Objects involved in the Use-Case.



Observ able

*Observer*

<<Singleton>>

TMTServer

**Figure 31 : Object Diagrams : Correct Errors**

## USE CASE < Show TMT Window >

* 1. **Brief Description**

This Use-Case defines how the appropriate TMT Window (Manager Client Window or Developer Client Window) is displayed on screen, using the data provided on the Login Screen (username and password).

DBMS

(f rom Mngmt - Client/Serv er)

Access Database

(f rom Mngmt - Client/Serv er)

Show TMT Window

**Figure 32 : Show TMT Window**

* 1. **Flow Events**

After the username and password have been verified and validated, the SM provides the Manager Client Window to the MCM or provides the Developer Client Window to the DCM. Windows are displayed to the user (Administrator or Developer)

* 1. **Interaction Diagrams**
* **The SM confirms the Login**
* **The SM provides the appropriate Client Window to the Client Module**
* **The Client Module displays the window to the user.**
  + 1. *Sequence Diagrams*

These Sequence Diagrams show Actors and Objects messages exchange in the Use-Case < Show TMT Window>.

: Administrator

Manager Client Window

Confirm Login

Provide MCW

Show:

* User name
* Project Management Icons
* Developers Management Icons

Show MCW

MCM

SM

**Figure 33 : Sequence Diagram : Show TMT Window (Administrator)**



: Developer

Confirm Login

Developper Client Window

Provide DCW

Show:

* User name
* Previous Week Timestamp
* Identifier for this week

Show DCW

DCM

SM

**Figure 34 : Sequence Diagram : Show TMT Window (Developer)**

* + 1. *Collaboration Diagrams*

This Collaboration Diagram shows the static structure of the Use-Case <Show TMT Window>.

|  |  |  |
| --- | --- | --- |
| SM |  | MCM |
| 2: Provide MCW |

**Figure 35 : Collaboration Diagram : Show TMT Window (Administrator)**

Manager Client Window

3: Show MCW

: Administrator

Show:

* User name
* Project Management Icons
* Developers Management Icons

1: Confirm Login

|  |  |  |
| --- | --- | --- |
| SM |  | DCM |
|  |

**Figure 36 : Collaboration Diagram : Show TMT Window (Developer)**

2: Provide DCW

Developper Client Window

3: Show DCW

: Developer

Show:

* User name
* Previous Week Timestamp
* Identifier for this week

1: Confirm Login

* 1. **Participating Objects**

The following objects collaborate and define the Use-Case < Show TMT Window> :

|  |  |
| --- | --- |
| **SM** | **This object interacts with the Database and determines if the user is an**  **Administrator or a Developer** |
| **MCM** | **This object interacts with the SM in order to display the TMT client**  **window to the user (Administrator)** |
| **DCM** | **This object interacts with the SM in order to display the TMT client**  **window to the user (Developer)** |

* 1. **Object Diagram**

The following Object Diagram shows the relations and constraints between Classes and Objects involved in the Use-Case.



Observ able

*Observer*

<<implements>>

<<implements>>

<<Singleton>>

TMTServer

Dev eloperObserv er

AdministratorObserv er

**Figure 37 : Object Diagrams : Show TMT Window**

## USE CASE < Close TMT Window >

* 1. **Brief Description**

This Use-Case defines how the TMT Window is closed upon logging out.



<<communicate>>

SGBD

(from Gestion - Client/Serveur)

Acceder a la BD

(from Gestion - Client/Serveur)

Si requete est logout

<<include>>

Fermer fenetre TMT

**Figure 38 : Close TMT Window**

* 1. **Flow Events**

After a logout query has been accepted and validated by the SM, the Client Modules (DCM or MCM) close the TMT client windows.

* 1. **Interaction Diagrams**
* **The SM confirms the Logout query**
* **The MCM or DCM closes the MCW or DCW (Manager or Developer Client Window)**
  + 1. **Sequence Diagrams**

These Sequence Diagrams show Actors and Objects messages exchange in the Use-Case <Close TMT Window>.

Confirm Logout

Close MCW

MCM

SM

**Figure 39 : Sequence Diagram : Close TMT Window (Administrator)**

Confirm Logout

Close DCW

DCM

SM

**Figure 40 : Sequence Diagram : Close TMT Window (Developer)**

* + 1. **Collaboration Diagrams**

This Collaboration Diagram shows the static structure of the Use-Case <Close TMT Window>.

|  |  |  |
| --- | --- | --- |
| SM |  | MCM |
|  |

**Figure 41 : Collaboration Diagram : Close TMT Window (Administrator)**

2: Close MCW

1: Confirm Logout

|  |  |  |
| --- | --- | --- |
| SM |  | DCM |
|  |

**Figure 42 : Collaboration Diagram : Close TMT Window (Developer)**

2: Close DCW

1: Confirm Logout

* 1. **Participating Objects**

The following objects collaborate and define the Use-Case < Close TMT Window> :

|  |  |
| --- | --- |
| **SM** | **This object interacts with the Database and determines if the user is an**  **Administrator or a Developer** |
| **MCM** | **This object interacts with the SM in order to close the TMT client**  **window (Administrator)** |
| **DCM** | **This object interacts with the SM in order to close the TMT client**  **window (Developer)** |

* 1. **Object Diagram**

The following Object Diagram shows the relations and constraints between Classes and Objects involved in the Use-Case.

Observ able

*Observer*

<<implements>>

<<implements>>

<<Singleton>>

TMTServer

Dev eloperObserv er

AdministratorObserv er

**Figure 43 : Object Diagrams : Close TMT Window**

