

# **Time Monitoring Tool Software Architecture Document**

**Version <1.0>**

<b>Time Monitoring Tool</b>	<b>Version:</b> <1.0>
<b>Software Architecture Document</b>	<b>Date:</b> <26/01/01>
upedu ex sad	

## Revision History

Date	Version	Description	Author
26/01/01	<1.0>	First Version	Robert Latour

<b>Time Monitoring Tool</b>	<b>Version:</b> <1.0>
<b>Software Architecture Document</b>	<b>Date:</b> <26/01/01>
upedu_ex_sad	

## Table of Contents

<b>1. Introduction</b>	<b>4</b>
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms, and Abbreviations	4
1.4 References	4
<b>2. Architectural Representation</b>	<b>4</b>
<b>3. Architectural Goals and Constraints</b>	<b>4</b>
<b>4. Use-Case View</b>	<b>5</b>
4.1 Use-Case Realizations	5
<b>5. Logical View</b>	<b>5</b>
5.1 Overview	5
5.2 Architecturally Significant Design Packages	5
5.2.1 Design Model: Packages Diagrams	5
5.2.2 Packages Description	6
5.2.3 Design Model: Design Class Diagrams	8
5.2.4 Design Classes Description	9
<b>6. Interface Description</b>	<b>14</b>
<b>7. Size and Performance</b>	<b>14</b>
<b>8. Quality</b>	<b>14</b>

Time Monitoring Tool	Version: <1.0>
Software Architecture Document	Date: <26/01/01>
upedu_ex_sad	

# Software Architecture Document

## 1. Introduction

### 1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

### 1.2 Scope

This Software Architecture Document provides an architectural overview of the Time Monitoring Tool. The Time Monitoring Tool 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 an Administrator to derive analyses and produce reports based on the data entered in the system.

### 1.3 Definitions, Acronyms, and Abbreviations

See Glossary, document upedu\_ex\_gloss.pdf

### 1.4 References

1. TMT - Glossary
2. TMT - Use Case Specification
3. TMT - Supplementary Specification
4. TMT - Software Development Plan
5. TMT - Iteration Plan

## 2. Architectural Representation

This document presents the architectural as a series of views; use case view, process view, deployment view, and implementation view. These views are presented as Rational Rose Models and use the Unified Modeling Language (UML).

## 3. Architectural Goals and Constraints

The TMT system to be developed is a stand-alone tool that is integrated within the organization's Intranet. It consists in four major components: a Developer Client Module, a Server Module, a Database, and an Administrator Client Module

All components must execute on a WindowsNT personal computer.

The Developer and Administrator Client Modules must communicate with the server over a TCP/IP connection.

The Server and the Database components should be located on the same host.

Time Monitoring Tool	Version: <1.0>
Software Architecture Document	Date: <26/01/01>
upedu_ex_sad	

#### 4. Use-Case View

The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage (that exercise many architectural elements) or that stress or illustrate a specific, delicate point of the architecture.

Refer to *Use-Case Specifications document* for more information- upedu\_ex\_ucspec.pdf

##### 4.1 Use-Case Realizations

Refer *Use Case Realization document* – upedu\_ex\_ucrea.pdf

#### 5. Logical View

This section describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages. And for each significant package, its decomposition into classes and class utilities.

##### 5.1 Overview

This subsection describes visually the overall decomposition of the design model in terms of its package hierarchy and layers.

##### 5.2 Architecturally Significant Design Packages

###### 5.2.1 Design Model: Packages Diagrams

The design model represents explicitly the structure and organization of the TMT system. Packages and corresponding classes are presented with a brief description.

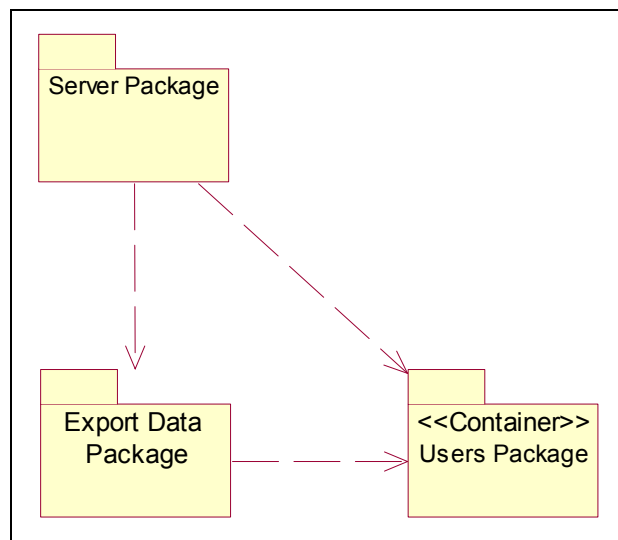


Figure 1: Design Model Packages Level 1

<b>Time Monitoring Tool</b>	<b>Version:</b> <1.0>
<b>Software Architecture Document</b>	<b>Date:</b> <26/01/01>
upedu ex sad	

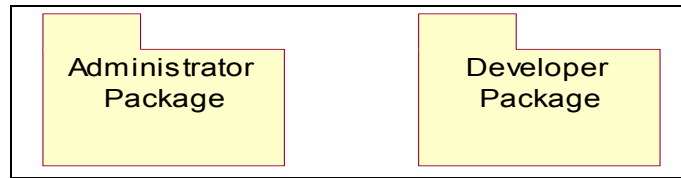


Figure 2: Design Model Packages Level 2 (Users Packages)

## 5.2.2 Packages Description

### 5.2.2.1 Level 1 Packages

<b>Server</b>	
<b>Description :</b>	<b>Main System Package. Although this package is dependant of other system packages, this package is the central point of all decisions made in the system. All client queries are managed by this package.</b>
<b>Corresponding classes* :</b>	<b>Observable Observer AdministratorObserver DeveloperObserver TMTServer</b>
<b>Relations :</b>	<b>Main TMT package. Dependant of: Developer, Administrator and Export Data packages</b>
<b>Sub packages :</b>	<b>Users, Export Data</b>

<b>Users</b>	
<b>Description :</b>	<b>Container Package for the Developer and Administrator Modules</b>
<b>Relations :</b>	<b>Is a sub package of the main package Server.</b>
<b>Sub packages :</b>	<b>None.</b>

<b>ExportData</b>	
<b>Description :</b>	<b>This package includes all necessary methods to import or export from/to Excel and MsProject. This package is dependant of the Developer Package: it gets its data from there.</b>
<b>Corresponding classes* :</b>	<b>MSExcel MSProject</b>
<b>Relations :</b>	<b>Is a sub package of the main package Server. Is dependant of the Developer Package.</b>
<b>Sub packages :</b>	<b>Users</b>

<b>Time Monitoring Tool</b>	<b>Version:</b> <1.0>
<b>Software Architecture Document</b>	<b>Date:</b> <26/01/01>
upedu_ex_sad	

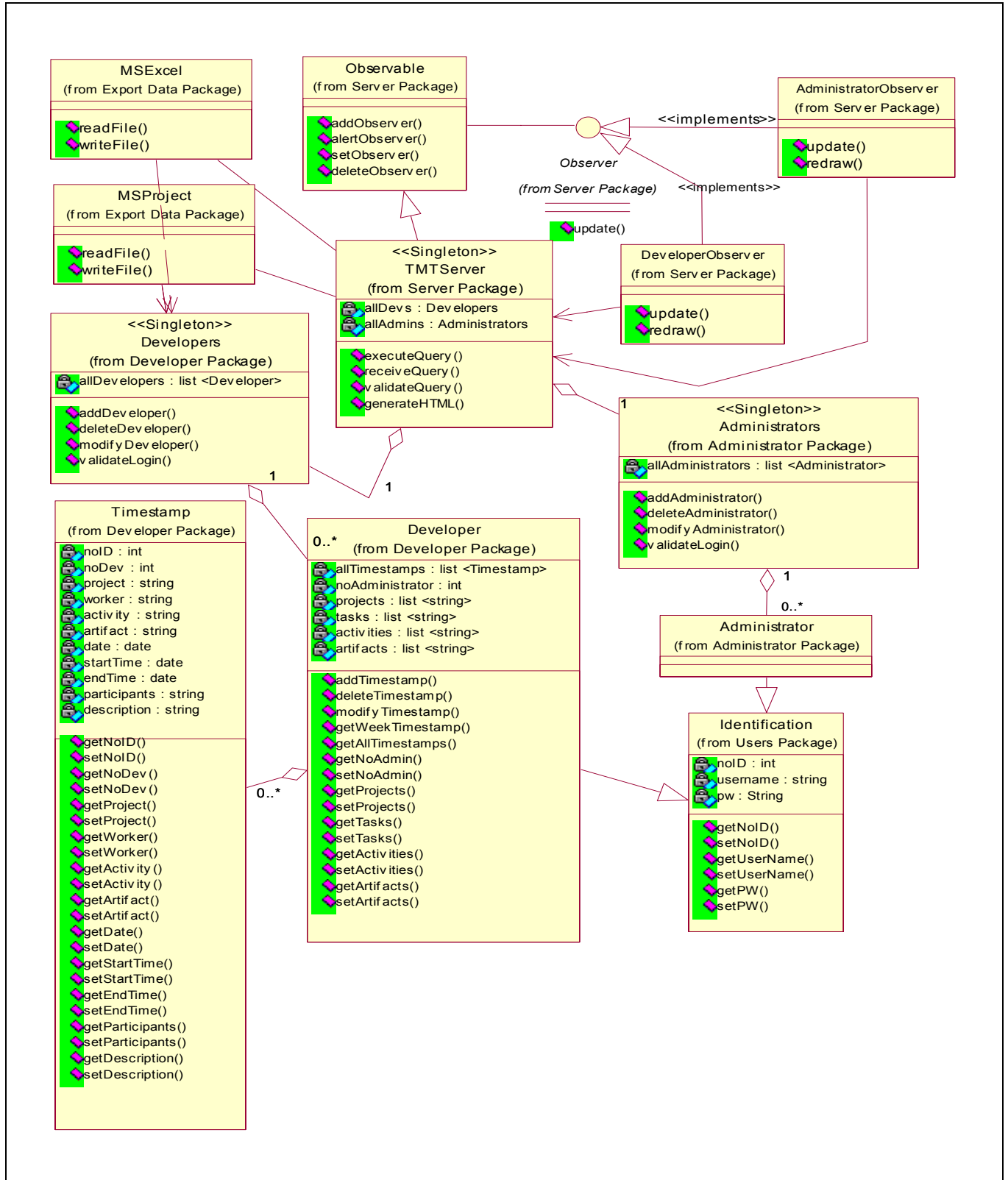
#### 5.2.2.2 Level 2 Packages

<b>Developer</b>	
<b>Description :</b>	<b>This package corresponds to the Developer Module. All information and methods regarding Developers are contained within this package.</b>
<b>Corresponding classes* :</b>	<b>Developers Timestamp Developer Identification</b>
<b>Relations :</b>	<b>Is a sub package of Users.</b>
<b>Sub packages :</b>	<b>None.</b>

<b>Administrator</b>	
<b>Description :</b>	<b>This package corresponds to the Administrator (Manager) Module. All information and methods regarding Administrators are contained within this package</b>
<b>Corresponding classes* :</b>	<b>Administrators Administrator Identification</b>
<b>Relations :</b>	<b>Is a sub package of Users.</b>
<b>Sub packages :</b>	<b>None.</b>

<b>Time Monitoring Tool</b>	<b>Version:</b> <1.0>
<b>Software Architecture Document</b>	<b>Date:</b> <26/01/01>
<b>upedu ex sad</b>	

### 5.2.3 Design Model: Design Class Diagrams





Time Monitoring Tool	Version: <1.0>
Software Architecture Document	Date: <26/01/01>
upedu ex sad	

#### 5.2.4 Design Classes Description

Property	Description
Name	Observable
Description	Abstract class representing the subject to be observed (Template Observer).
Responsibilities	None that need detailed information.
Relations	Abstract class, is a TMTServer Class Generalization.
Methods	<u>AddObserver()</u> : Add a new observer into subject Observers' list. <u>DeleteObserver()</u> : Delete an observer from the subject Observers' list. <u>SetChanged()</u> : Enable subject change. <u>AlertObserver()</u> : Alert engaged observers from subject change.
Attributes	None
Special Requirements	None

Property	Description
Name	Observer
Description	Abstract class representing the Subject's Observer (Template Observer).
Responsibilities	None that need detailed information.
Relations	Abstract class, is a DeveloperObserver & AdministratorObserver Generalization.
Methods	<u>Update()</u> : Indicates to the observer that the Subject has changed and to update his observations.
Attributes	None
Special Requirements	None

Property	Description
Name	DeveloperObserver
Description	Observer (Developer) implementation.
Responsibilities	Observes the TMTServer and alert client when the TMTServer changes.
Relations	Association with TMTServer & is an Observer Implementation.
Methods	<u>Update()</u> : Indicates to the DCM that the TMTServer has changed. <u>Redraw()</u> : Indicates to the DCM that its window must be refreshed.
Attributes	None
Special Requirements	None

Property	Description
Name	AdministratorObserver
Description	Observer (Administrator) implementation.
Responsibilities	Observes the TMTServer and alert client when the TMTServer changes.
Relations	Association with TMTServer & is an Observer Implementation.
Methods	<u>Update()</u> : Indicates to the MCM that the TMTServer has changed. <u>Redraw()</u> : Indicates to the MCM that its window must be refreshed.
Attributes	None
Special Requirements	None

Time Monitoring Tool	Version: <1.0>
Software Architecture Document	Date: <26/01/01>
upedu ex sad	

Property	Description
Name	TMTServer
Description	Singleton and heart of the system. This class controls all other classes.
Responsibilities	Ensures that all queries from the clients are received and processed.
Relations	Is an Observable Implementation. Can be observed by Administrators and Developers. Association with Export and Import modules. Aggregation with Developers and Administrators.
Methods	<u>ExecuteQuery()</u> : Execute the query <u>ReceiveQuery()</u> : Listens to coming queries <u>ValidateQuery()</u> : Validates the query <u>GenerateHTML()</u> : Generates the query results
Attributes	<u>AllDevs</u> : Object: Developers' list. <u>AllAdmins</u> : Object: Administrators' list.
Special Requirements	None

Property	Description
Name	MSProject
Description	Transforms data coming from MSProject or data destined to be exported to MSProject.
Responsibilities	Ensures data importation from MSProject. Ensures data exportation to MSProject
Relations	Association with TMTServer. Dependant of Developers.
Methods	<u>ReadFichier()</u> : Read an MSProject file and add data to the database. <u>WriteFile()</u> : Write an MSProject file with data from the database.
Attributes	None
Special Requirements	None

Property	Description
Name	MSExcel
Description	Transforms data coming from MSExcel or data destined to be exported to MSExcel.
Responsibilities	Ensures data importation from MSExcel. Ensures data exportation to MSExcel
Relations	Association with TMTServer. Dependant of Developers.
Methods	<u>ReadFichier()</u> : Read an MSExcel file and add data to the database. <u>WriteFile()</u> : Write an MSExcel file with data from the database.
Attributes	None
Special Requirements	None

Time Monitoring Tool	Version: <1.0>
Software Architecture Document	Date: <26/01/01>
upedu ex sad	

Property	Description
Name	Administrators
Description	Singleton containing Administrators' list.
Responsibilities	Maintain an updated list of all Administrators.
Relations	Aggregation to TMTServer. Aggregation to Administrators.
Methods	<u>AddAdministrator()</u> : Add an Administrator to the system <u>DeleteAdministrator()</u> : Delete an Administrator from the system. <u>ModifyAdministrator()</u> : Modify an Administrator in the system. <u>ValidateLogin()</u> : Validates username and password.
Attributes	<u>AllAdministrators</u> : List of all system's Administrators.
Special Requirements	None

Property	Description
Name	Developers
Description	Singleton containing Developers' list.
Responsibilities	Maintain an updated list of all Developers.
Relations	Aggregation to TMTServer. Aggregation to Developers.
Methods	<u>AddDeveloper()</u> : Add an Developer to the system <u>DeleteDeveloper ()</u> : Delete a Developer from the system. <u>ModifyDeveloper ()</u> : Modify a Developer in the system. <u>ValidateLogin()</u> : Validates username and password.
Attributes	<u>AllAdministrators</u> : List of all system's Developers.
Special Requirements	None

Property	Description
Name	Identification
Description	Parent class representing a system user (Administrator or Developer).
Responsibilities	None
Relations	Is an Administrator or a Developer Generalization.
Methods	<u>GetNoID()</u> : Returns user's ID number. <u>SetNoID()</u> : Modifies user's ID number.  <u>GetName()</u> : Returns username. <u>SetName()</u> : Modifies username. <u>GetPW()</u> : Returns user's password. <u>SetPW()</u> : Modify user's password.
Attributes	<u>NoID</u> : User's ID number. <u>Name</u> : Username. <u>PW</u> : User's Password.
Special Requirements	None

Property	Description
Name	Administrator
Description	Represents the Administrator entity.
Responsibilities	Models the Administrator entity.
Relations	Aggregation to Administrators and derived from Identification. An Administrator is in the Administrators' list and is based upon a user's Identification.

Time Monitoring Tool	Version: <1.0>
Software Architecture Document	Date: <26/01/01>
upedu ex sad	

Methods	None
Attributes	None – Inherited attributes from Identification.
Special Requirements	None

Property	Description
Name	Developer
Description	Represents the Developer entity.
Responsibilities	Models the Developer entity.
Relations	Aggregation to Developers and derived from Identification. A Developer is in the Developers' list and is based upon a user's Identification.
Methods	<u>AddTimestamp()</u> : Add a timestamp. <u>DeleteTimestamp()</u> : Delete a timestamp. <u>ModifyTimestamp()</u> : Modify a timestamp. <u>GetWeekTimestamps()</u> : Return current week's timestamps. <u>GetAllTimestamps()</u> : Return all timestamps. <u>GetNoAdmin()</u> : Return Developer's Administrator ID number. <u>SetNoAdmin()</u> : Modify Developer's Administrator ID number. <u>GetProjects()</u> : Return Projects' list. <u>SetProjects()</u> : Modify Projects' list. <u>GetTasks()</u> : Return Tasks' list. <u>SetTasks()</u> : Modify Tasks' list. <u>GetActivities()</u> : Return Activities' list. <u>SetActivities()</u> : Modify Activities' list. <u>GetArtifacts()</u> : Return Artifacts' list. <u>SetArtifacts()</u> : Modify Artifacts' list.
Attributes	<u>allTimestamps</u> : Timestamps' list. <u>NoAdmin</u> : Developer's Administrator ID number. <u>Projects</u> : Projects' list. <u>Tasks</u> : Tasks' list. <u>Activities</u> : Activities' list. <u>Artifacts</u> : Artifacts' list.
Special Requirements	None

Time Monitoring Tool	Version: <1.0>
Software Architecture Document	Date: <26/01/01>
upedu ex sad	

Property	Description
Name	Timestamp
Description	Represents the Timestamp entity.
Responsibilities	Models the Timestamp entity.
Relations	Aggregation to Developer. A Developer has 0 or more timestamps.
Methods	<u>GetNoID()</u> : Return Timestamp's ID <u>SetNoID()</u> : Modify Timestamp's ID. <u>GetNoDev()</u> : Return Timestamp's associated Developer ID. <u>SetNoDev()</u> : Modify Timestamp's associated Developer ID. <u>GetProject()</u> : Return Timestamp's associated Project. <u>SetProject()</u> : Modify Timestamp's associated Project. <u>GetWorker()</u> : Return Timestamp's associated Role (worker). <u>SetWorker()</u> : Modify Timestamp's associated Role (worker).. <u>GetActivity()</u> : Return Timestamp's associated Activity. <u>SetActivity()</u> : Modify Timestamp's associated Activity. <u>GetArtifact()</u> : Return Timestamp's associated Artifact. <u>SetArtifact()</u> : Modify Timestamp's associated Artifact. <u>GetDate()</u> : Return Timestamp's Date. <u>SetDate()</u> : Modify Timestamp's Date. <u>GetStartTime()</u> : Return Timestamp's Starting Hour. <u>SetStartTime()</u> : Modify Timestamp's Starting Hour. <u>GetEndTime()</u> : Return Timestamp's Ending Hour. <u>SetEndTime()</u> : Modify Timestamp's Ending Hour. <u>GetParticipants()</u> : Return Timestamp's Participants. <u>SetParticipants()</u> : Modify Timestamp's Participants. <u>GetDescription()</u> : Return Timestamp's Description. <u>SetDescription()</u> : Modify Timestamp's Description.
Attributes	<u>NoID</u> : Timestamp's ID. <u>NoDev</u> : Timestamp's associated Developer ID <u>Project</u> : Timestamp's associated Project. <u>Worker</u> : Timestamp's associated Role (worker).. <u>Activity</u> : Timestamp's associated Activity. <u>Artifact</u> : Timestamp's associated Artifact. <u>Date</u> : Timestamp's Date. <u>StartTime</u> : Timestamp's Starting Hour. <u>EndTime</u> : Timestamp's Ending Hour. <u>Participants</u> : Timestamp's Participants. <u>Description</u> : Timestamp's Description.
Special Requirements	None

<b>Time Monitoring Tool</b>	<b>Version:</b> <1.0>
<b>Software Architecture Document</b>	<b>Date:</b> <26/01/01>
<b>upedu_ex_sad</b>	

## **6. Interface Description**

See User Interface documents - upedu\_ex\_uiprt.pdf

## **7. Size and Performance**

The selected architecture supports the sizing and timing requirements through the implementation of a client-server architecture. The client portion is implemented on local campus PCs or remote dial up PCs. The components have been designed to ensure that minimal disk and memory requirements are needed on the PC client portion.

## **8. Quality**

The software architecture supports the quality requirements, as stipulated in the Software Requirements Specification and Supplementary Specification