

MODELING OF TECHNICAL MANAGEMENT INFORMATION SYSTEM WITH UML AND IMPLEMENTATION USING THE COMBINATION OF PHP AND MYSQL

Nik Maria Nik Mahamood

Jabatan Sains (Sains Komputer), Program Pengajian Diploma, Universiti Teknologi Malaysia,
Jalan Semarak, 54100 Kuala Lumpur.
maria@utmkl.utm.my

Abstract: This paper focuses on modeling of system using Unified Modeling Language notations and its implementation using Hypertext preprocessor (PHP) and MySQL. The Unified Modeling Language is one of the modeling languages for object-oriented analysis and design, which provide a set of concepts, and notations that can be used throughout the entire software development process. PHP is a scripting language, which embedded in Hyper Text Mark Language and interpreted by the server. It used to manage dynamic content databases, work with databases, handle session tracking and even build entire e-commerce sites. MySQL is the popular database management system, which offers a rich and very useful set of functions. This paper describes the process of modeling system with UML, and why the combination of Hypertext Preprocessor and MySQL selected for developing the system.

Keywords: Unified Modeling Language, Hypertext Preprocessor, Hyper Text Markup Language and Object-Oriented Analysis and Design.

Introduction

The Technical Management Information System (TMIS) developed to handle prepaid mobile services for technical division in CELCOM. The objective of this system is to allow the authorized users retrieve information of prepaid mobile subscribers from the server easily and faster via company intranet. This system was modeled based on the requirements given by users at technical division.

The system provides the prepaid subscribers information to the authorized users. Therefore, the authorized users can view the details of prepaid subscribers based on subscribers' categories as daily, fortnightly and monthly basis. The subscriber categories consist of subscriber trends, subscriber activities and subscriber account trends. In the subscriber trends, the authorized users can access information about numbers of cards activated, active with and without Air Time and inactive with and without Air Time. The second category of subscriber provides authorized users with the information about monitor activity in Electronic Serial Number (ESN) and Subscribers Identification Mobile (SIM), suspended activity, purged activity and reactivates. The authorized user also can view the information about daily activation, 010 registrations and first-use activity in the subscriber account trends. The system provides the authentication process before any of users log in to the system in order to make sure that the security of system is protected, which only permits authorized users.

The system basically, offers services that are easy to use and customizable in order to address the authorized users' need. Below are the following services that TMIS offered:

- Enables the authorized user to view and request the information of prepaid subscribers regarding to customer categories in the form of table list.
- Enable the authorized users to view the selected information as daily, fortnightly or monthly basis.
- Enable the authorized user to generate report in the graph and chart form.
- Enables the authorized users to export selected data to the text files.
- Enables the authorized users to print selected data in form of tables, text files, graph or chart.

The object-oriented analysis and design of this system was modeled using UML. The combination of PHP and MySQL used for coding in order to implement system modeled. The entire design of TMIS that will discuss in this paper includes the following UML diagrams: use case diagrams, package diagrams, sequence diagrams, collaboration diagrams and statechart diagrams.

What is UML?

The UML is a modeling language used to describe software systems. It stands for Unified Modeling Language. UML unifies several modeling tools that dominant by three methods, which are Object modeling technique (OMT) by Rumbaugh; Booch's Method by Booch and Object-Oriented Software Engineering (OOSE) by Jacobson. These unified methods are wrapped under the heading of the UML (Booch et al., 1999). According to OMG (Object Management Group, an industry standards body) the specification quote for UML is:

The UML is the standard language for specifying, visualizing, constructing, and documenting all the artifacts of a software system, as well as for business modeling and other non-software systems.

The efforts of Grady Booch, James Rumbaugh and Ivar Jacobson at Rational Software Corporation resulted in the release of the UML 0.9 and 0.91 documents in June and October of 1996. The UML adopted in 1997 as the standard for modeling object-oriented systems by the OMG. UML defines nine types of diagrams that grouped in two model views, which are static model view and dynamic model view. The static model view includes use case, class or package, object, component and deployment diagrams. The dynamic model view consists of sequence diagram, collaboration diagram, state chart diagram and activity diagram (Booch et. all., 1999).

The important point to note here is that UML is not a method, it is simply notation or we can say that UML is a standard diagramming notation. The UML used in the service of doing object-oriented analysis and design to define a software system, to detail the artifacts in the system, to document and construct. The UML may be used in a variety of ways to support a software development methodology such as the Rational Unified Process but in itself does not specify that methodology or process.

Next section, this paper will discuss on applying UML to illustrate analysis and design of system model. The diagrams that will be focused are use case, class or package that representing static model view while sequence, collaboration and state chart that representing dynamic model view.

System Overview

The overview of TMIS is shown in figure 1. It can be seen that the system interfaces with the *authorized user* and *server*. The *authorized user* is the primary actor of TMIS. The primary actor initiates activity for system. The actor is a person/user who will interact with the system. The main task of *Authorized User* is to view or request information from the system and can generate the appropriate report based on preferences. The second actor, *Server* provides the information to TMIS based on request from the primary actor.

Use Case Diagram

Use case diagram is a central to model the behavior of a system, subsystem or a class that shows the overall system overview. The emphasis is on what a system does rather than how. The use case diagram is a collection of actors, use cases, and communications association. The actors, use cases and communication associations are represented with stick figures symbol, ovals symbol and lines symbol consecutively. The following is a detail discussion of the use case diagram shown in figure 2.

An actor is who or what initiates the events involved in that task. Actors are simply roles that people or objects play. The use case diagram as shown above consists of one actor and five use cases. The actor is an *authorized user* who will initiate all activities with the system and will get some value in return. The use cases are made up of *to authenticate*, *customer trends*, *inactive customer activities*, *new customer account trends* and *utilities* use case.

The connection between actor and use case is known as a communication association. There are five communication associations, which are:

- The *authorized user* communicates with the *To Authenticate Use Case*.
Control the access to the system.
- The *authorized user* communicates with the *Customer trends Use Case*.
Request to view information on customer trends.
- The *authorized user* communicates with the *Inactive Customer Activities Use Case*.
Request to view information on inactive customer activities.
- The *authorized user* communicates with the *New Customer Account Trends Use Case*.
Request to view information on new customer account trends.
- The *authorized user* communicates with the *Utilities Use Case*.
Request to view data in form of format, generate report in certain form such as graph, bar chart, pie chart or export data to text file and print data.

The Use case diagrams are helpful in three cases, which are to determine features or requirements of the system, to communicate with clients and to generate test cases.

Sequence diagrams

A sequence diagram is an interaction diagram that details how operations are carried out, what messages are sent and when. The sequence diagrams are organized according to time. The time progresses as we go down the page. The objects involved in the operation are listed from left to right according to when they take part in the message sequence. The objects are instances of classes while classes are the "blueprints" for objects. A class wraps attributes / data and behaviors / methods or functions into a single distinct entity.

Figure 3 shows a sequence diagram of an authentication process for *to authenticate* use case. There are five classes involved in this diagram which are *authorized user*, *clsLoginScreen*, *clsLoginManager*, *clsUserInfo* and *clsMainMenu*.

The object initiating the sequence of messages is an *authorized user*. The *authorized user* sends an *EnterInput ()* message to a *clsLoginScreen* Class. The *clsLoginScreen* then sends a *SubmitInfo()* message to a *clsLoginManager*. The *clsLoginManager* will send *GetInfo()* message to *clsUserInfo*. When information is received from *clsUserInfo*, *clsLoginManager* issues a self call to verify id's user and password's user are valid or not. Then *clsLoginManager* sends *openMainMenu()* message to a *clsMainMenu*. Then *clsMainMenu* sends *DisplayMainMenu()* message to *Authorized User*. So that, *Authorized User* will know whether the authentication is successful or not. If successful, the *Authorized User* has fully access to the system.

Each vertical dotted line is known as lifeline, which is representing the time that an object exists. Each arrow is a message call. An arrow goes from the sender to the top of the activation bar of the message on the receiver's lifeline. The activation bar represents the duration of execution of the message. The diagram has a clarifying note, which is text inside a dog-eared rectangle. The notes can be placed into any kind of UML diagram.

Collaboration diagrams

Collaboration diagrams are also known as interaction diagrams. They convey the same information as sequence diagrams, but they focus on object's roles instead of the times that messages are sent. The collaboration diagram can be generated from the existing sequence diagrams or in the other versa. These two diagrams are isomorphic which mean we can take one and transform it to the other. Figure 4 shows a collaboration diagram for authentication process.

In a collaboration diagram, object roles are the vertices and messages are the connecting links. Colon symbol precedes class's name. Each message in a collaboration diagram has a sequence number. The top-level message is numbered 1. Messages at the same level which sent during the same call have the same decimal prefix but suffixes of 1, 2, etc. according to when they occur.

Statechart diagram

The state chart diagram shows the possible states of the object and the transitions that cause a change in state. Every object has behaviors and state. The state of an object depends on its current activity or condition

Figure 5 shows a statechart diagram of system that consists of four states. It can be seen that in the first state, an authorized user wants to view and request the selected statistic in the form of table, chart or report on the web, which are based on the customer categories such as customer trends, customer activities and customer account trends from the system. The authorized user should press a button to select customer trends, inactive customer activity or new customer account trends.

After the category of customers selected, an authorized user reaches at the second state. Here the authorized user presses a button to select GSM, ETACS or TOTAL option. In the third state, the authorized user can view or request the information as daily, fortnightly and monthly basis. When it succeeds, the authorized user must enter the date to get selected information of prepaid subscribers.

In the finally state, the authorized user can generate report in form of table or bar chart, export data to text file and print the selected information. From each state comes a complete set of transitions that determine the subsequent state.

States are represented by rounded rectangle symbol. Transitions are arrows from one state to another. Events or conditions that trigger transitions are written beside the arrows. This diagram has one self-transition, on a *TableScreen* state.

The initial state represents, as a black circle is a dummy to start the action. Final states are also dummy states that will be terminated the action. The action that occurs as a result of an event or condition is expressed in the form or action. While in its *TableScreen* state, the object does not wait for an outside event to trigger a transition. Instead, it performs an activity. The result of that activity determines its subsequent state.

Packages and object diagrams

To simplify complex class diagrams, we can group classes into packages. A package is a collection of logically related UML elements. The diagram in figure 6 shows a business model in which all classes in the system can be grouped into five main packages and the relationship between packages. The criteria for grouping classes follow semantic roles. Each package defines a set of role, which results in a minimum of coupling and a maximum of cohesion within the package. The use cases are package based on their roles.

Let's take a look the diagram in Figure 6. Packages appear as rectangles with small tabs at the top. The package name is on the tab or inside the rectangle. The dotted arrows mean dependencies. One package depends on another if changes in the other could possibly force changes in the first.

The packages are briefly described as follow:

Interface

Interface package provides an interface between the user and system. It consists of *clsMainMenu* class, *clsCustTrendMenu* class, *clsInactiveCustActMenu* class, *clsNewCustAcctMenu* class, *clsUtilityScreen* class, *clsDisplayTypesScreen* class and *clsLoginScreen* class. The communication between this package with other packages is shown in Figure 7.

Display Data

The Displaydata package is responsible to display appropriate data to user. This package consists of *clsReportScreen* class, *clsTableScreen* class and *clsTextScreen* class. The communication between this package with other packages, as Figure 8 shows.

Management

Management package is responsible to manage and handle the activities for the TMIS. This package consists of *clsSysController* class, *clsDbMySQLMgr* class and *clsLoginManager* class. The communication between this package with other packages is shown in figure 9.

DBManagement

DBManagement package consists of *clsServerMgr* class, which is responsible to receive appropriate data from the database, to construct and execute SQL command. The communication between DBManagement package with other packages, as Figure 10 shows:

Component

Component package provides information to the respective packages. It consists of *clsSCP* class, *clsPServer* class and *clsUserInfo* class. Figure 11 shows the communication between this package with other packages.

Result and Implementation

The model of system developed using UML notations is implemented using HTML, PHP Version 3.0 and MySQL, which run on UNIX operating system. Both PHP and MySQL are open source which means they are free to use. This section will discuss why the combination of them is used in the implementation of system.

HTML is the easiest web-altering tool and most browsers are varying widely in their recognition of HTML elements. HTML consists of a predefined set of tags that can be used to manipulate text and arrange data on a web page. The HTML code begins with the tag `<HTML>` and ends with `</HTML>`. It used to create the interfaces of TMIS.

The MySQL is an open source relational database management system. It used in developing of TMIS as the backend database on a UNIX environment. MySQL was chosen for a number of reasons. The first reason why the developers used MySQL that it is free to use and provides a set of facilities to enter, store and search tables of information. It also offers a rich and useful set of functions that very close cooperation with the users. The examples of MySQL functions and how it's used along with the PHP code can be seen in fragment code shown in figure 13.

Another reason is MySQL suitable to handle large databases, such as prepaid subscribers database because of it is fast, reliable and easy to use. Besides that, the stable of connectivity and speed make MySQL highly suited for accessing databases on the Internet. My SQL also helps protect the security of data while easily allowing users access to it from a browser. Next reason is MySQL is a client/server system that consists of a multi-threaded SQL server that supports different backend, several different client programs and libraries, administrative tools, and a programming interface (Luke et al., 2001). The last reason is MySQL is one of databases that supported by PHP.

Figure 12 shows the packages include classes and the programming language used for coding each class. (Nik Mahamood et al., 2000).

Lets take a look of *clsLoginManager* class as shown in figure 12. This class coded using the combination of PHP, MySQL and HTML. The *clsLoginManager* class is responsible to manage the authorized user that will allow accessing to the system based upon the user's privileges. Figure 13 shows the fragment code of authentication process for *clsLoginManager* class. We can see how this class constructed using the combination of HTML, PHP and MySQL.

In implementation of the system, PHP version 3.0 used for coding, as a front end to MySQL database. PHP stands for Hypertext Preprocessor and it was created by Rasmus Lerdof. PHP is a scripting language that embedded in HTML and interpreted by the server (Luke et al., 2001).

Let's take a look at how the PHP code can insert directly alongside HTML shown in the fragment code of authentication process above. The PHP code begins with the tag `<?php` and ends with `?>`. This tells the server that everything between PHP tags need to be parsed for PHP instructions and then need to be executed. This means the server processes the script and sends plain HTML to the clients. The clients would not see any PHP instructions unless the developers have made an error and the server spits them out as is instead of processing them first.

The strength of PHP is that it is cross-platform and runs everywhere. PHP is written in 100% C language therefore it can run on wide variety operating systems such as UNIX, Linux, Windows 95/98/NT/2000/XP, IBMOS/2 and SOLARIS. PHP also can run on many web-servers like Apache, Netscape/iPlanet and Microsoft IIS (Luke et al., 2001). Nowadays, PHP becomes the most well known for creating dynamic web-application, e-commerce, emails and general standalone applications. PHP can be compiled and optimized by using Zend Optimizer to make it run faster.

According to the benchmarking of four web-scripting languages done by Zdnet, it found that PHP was the fastest compared to three other languages. The benchmark results are PHP pumped out about 47 pages/second, Microsoft ASP pumped out about 43 pages/second, Allaire ColdFusion pumped out about 29 pages/second and Sun Java JSP pumped out about 13 pages/second. As we know, the important point that we should mind whenever we develop a website is the performance and speeding of web scripting engine. That is, how many pages per second the scripting engine can pump out to the browser clients. This reason tells why PHP selected for web development.

The finding indicated that combination of PHP and MySQL are one of the best combination for data-driven on website. They are suitable for creating dynamic web pages that retrieve information from large database like prepaid mobile subscribers database used in TMIS. The execution speed of PHP considered fast enough when compared with other programming languages such as ASP (Active Server Pages). This is because in PHP modules, everything runs in PHP's memory space, which means PHP code will run faster because there is no overhead of communicating with different COM objects in different processes. On the other hand, ASP built on a COM-based architecture, which means different processes use different COM objects and this may affect COM overhead adds up and will slow things down.

The finding also indicated that PHP syntax is quite easy to code, all the syntaxes are similar to the C language syntax. However, the finding noticed that the disadvantage of PHP is error handling. The PHP has a poor error handling and it made difficulty for the developers to trace errors quickly. On the other hand, Java has a proven method of handling errors within code. However, PHP applications are more stable and do not depend on browser unlike java script applications, which depend on browser.

One of the difficulties that the finding found is to map UML classes into PHP code, which the developers have to think how to map classes that involved access specifiers such as public, private and protected in UML that can be implemented in the PHP paradigm. As we remind that, every language has limitations and PHP is no exception. The mainly limitation of PHP is the PHP object model is incomplete when compared to other object-oriented language like C++, Java and Python. This affect the implementation of design system when it required the use of object-oriented features that PHP does not have, such as interface, extends, protected, private and public. Because of this limitation, the coding of TMIS did not apply 100% object-oriented programming.

Conclusion and Discussion

The UML recognized as a modeling language for visualizing, specifying, constructing and documenting the artifacts of a software-intensive system. It used for expressing underlying object-oriented analysis and design ideas. The UML diagrams help software developers explain how any system works and.

The standard UML is intended primarily for software-intensive systems such domains as an Enterprise information system, Banking and financial services and distributed Web-based services (Booch et. all., 1999). However, the analysts, developers and implementers should remind that standard UML elements do not suit for modeling the web applications. According to Jim Conallen (Web Modeling Evangelist at Rational Software Corporation), some of web application's components do not fit nicely into a standard UML modeling when we trying to model a web application with UML. The best way to solve this problem is UML must be extended in order to fix with one modeling notations for the entire system. This extended called Web Application Extension (WAE) for the unified Modeling language that will provide a common way for designers and architects to express the entirety of their web application design with UML. (Conallen. J., 1999)

At the time of TMIS development, this system modeled using a standard UML. The finding only focused on how to model the interaction of elements such as objects/classes and their relationships inside the system rather than how to model the basic architecture of web application that involves web server, clients, browser and network. For future development, the modeling of TMIS can be refined with Web Application Extension for UML in order to explain the specific elements of the web application such as clients, web servers, browsers, web pages, forms, frames and session management.

As reminder, PHP is not suitable for coding that required fully object-oriented programming because it is not 100% object-oriented language. Its hoped that the object-oriented features like extends, public, private, protected will be supported in the future release of the PHP. Although PHP has a limitation in object-oriented approach, it is very helpful for those developers want to develop a web application with dynamic content and database interaction. As we noticed, every programming language has the strengths and limitations. So that the software developers should be able to choose the right programming language to meet with the user requirements and suite with the applications that are going to develop.

References

- [1] Bennett, McRobb and Farmer, *Object-Oriented Systems Analysis And Design using UML*, Second Edition, Mc Graw Hill, 2002.
- [2] Bennett, Skeleton and Lunn, *Schaum's OutLines Of UML*, Mc Graw Hill, 2002.
- [3] Booch, Rumbaugh and Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 1999.
- [4] Larman. C., *Applying UML And Patterns*, Second Edition, Prentice Hall PTR, 2002.
- [5] Luke, Welling. L. and Thomson, *PHP & MySQL Web Development*, Pearson Computer, 2001.
- [6] Atkinson, L., *Core PHP Programming*, Prentice Hall, 2000.
- [7] Conallen, J., *Modeling Web Application Architectures with UML*, in *Communication of ACM.*, 1999.p.63-70.
- [8] Nik Mahamood, N.M., Khamis, N., *Software Design Document for TMIS*, CELCOM, 2000.
- [9] MySQL, <http://www.mysql.com>
- [10] PHP, <http://www.php.net>
- [11] Benchmarks, <http://aldev0.virtualave.net/php-perl-benchmarks.html>

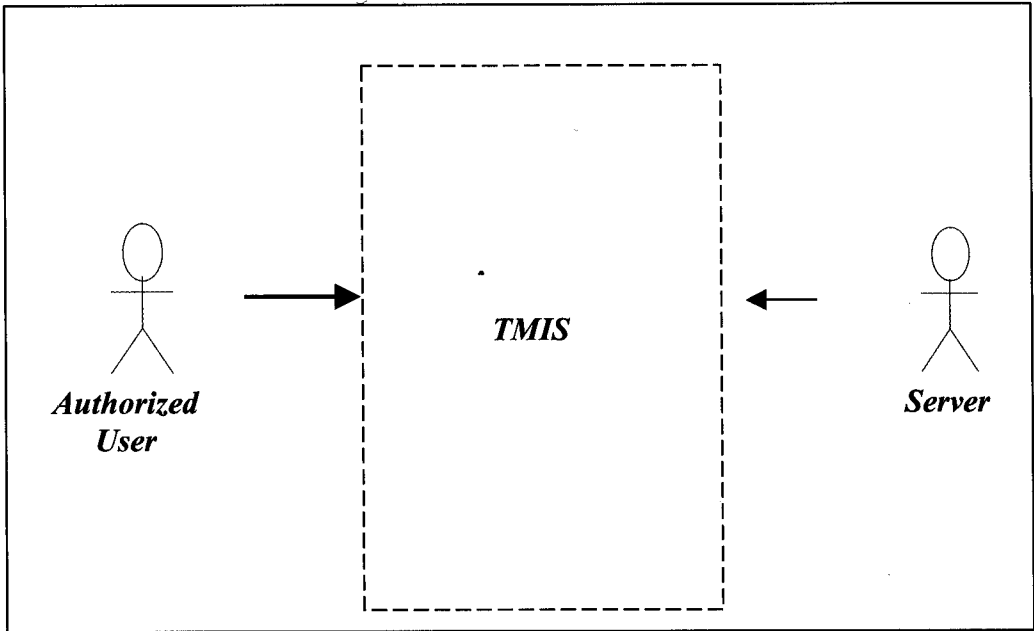


Figure 1 Context Diagram of System

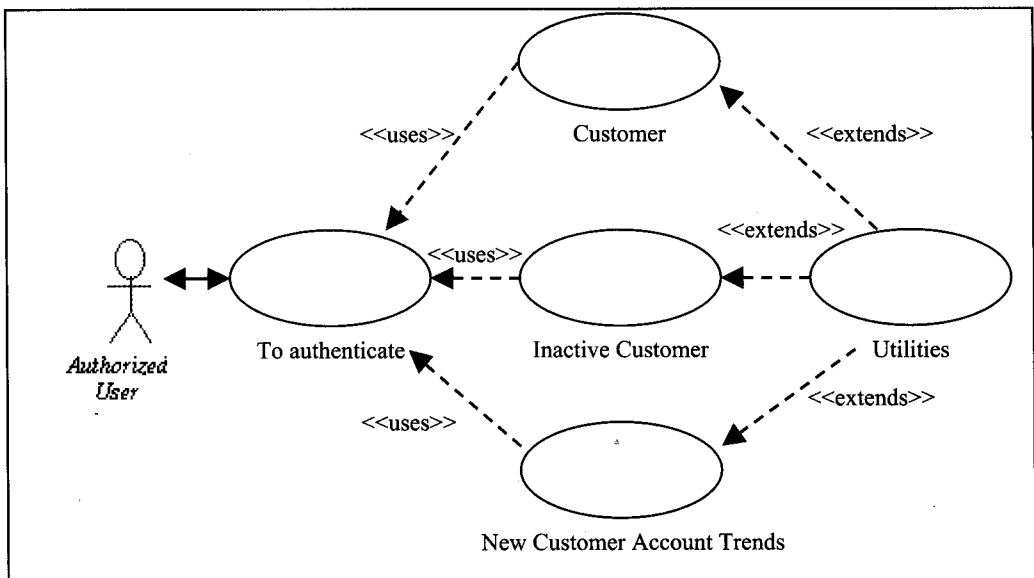


Figure 2 Use case Diagram of System

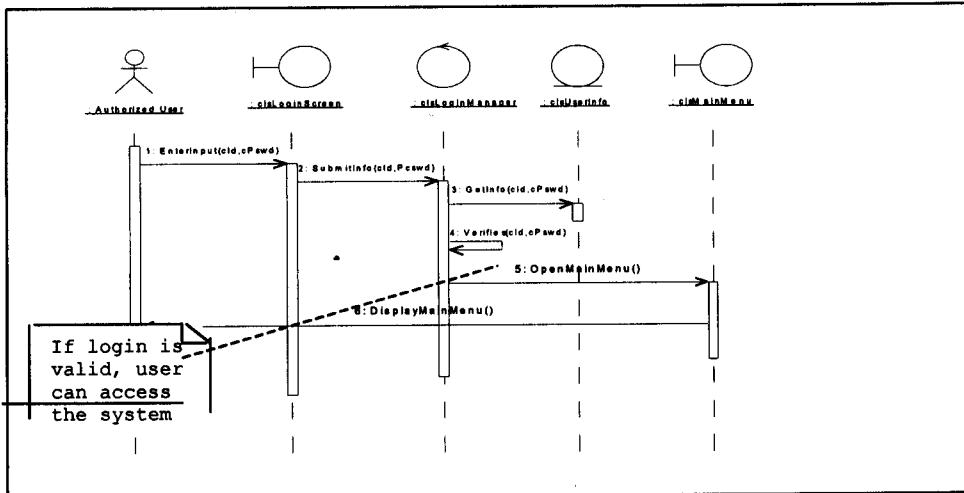


Figure 3 Sequence Diagram of System

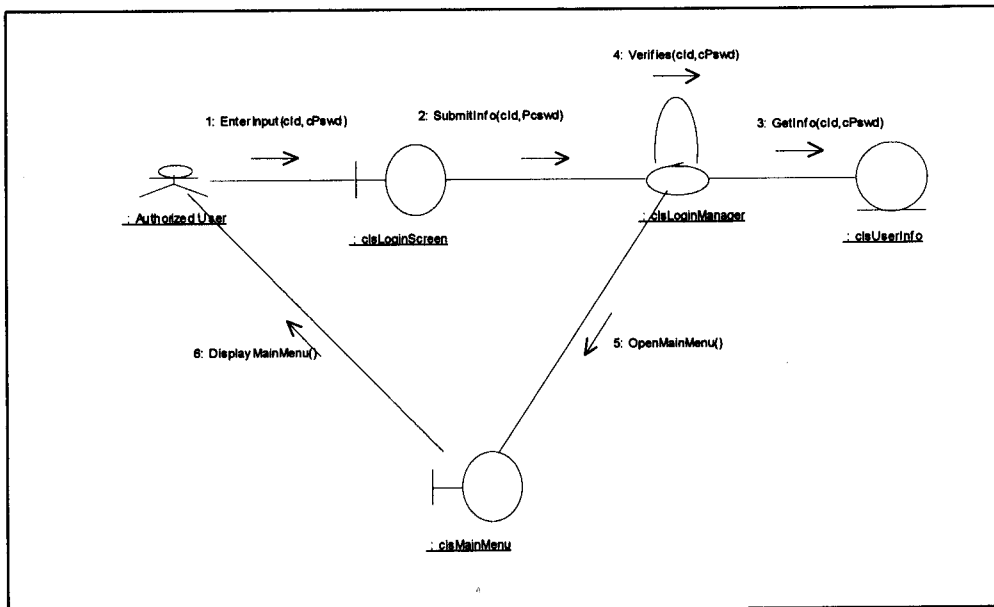


Figure 4: Collaboration Diagram of System

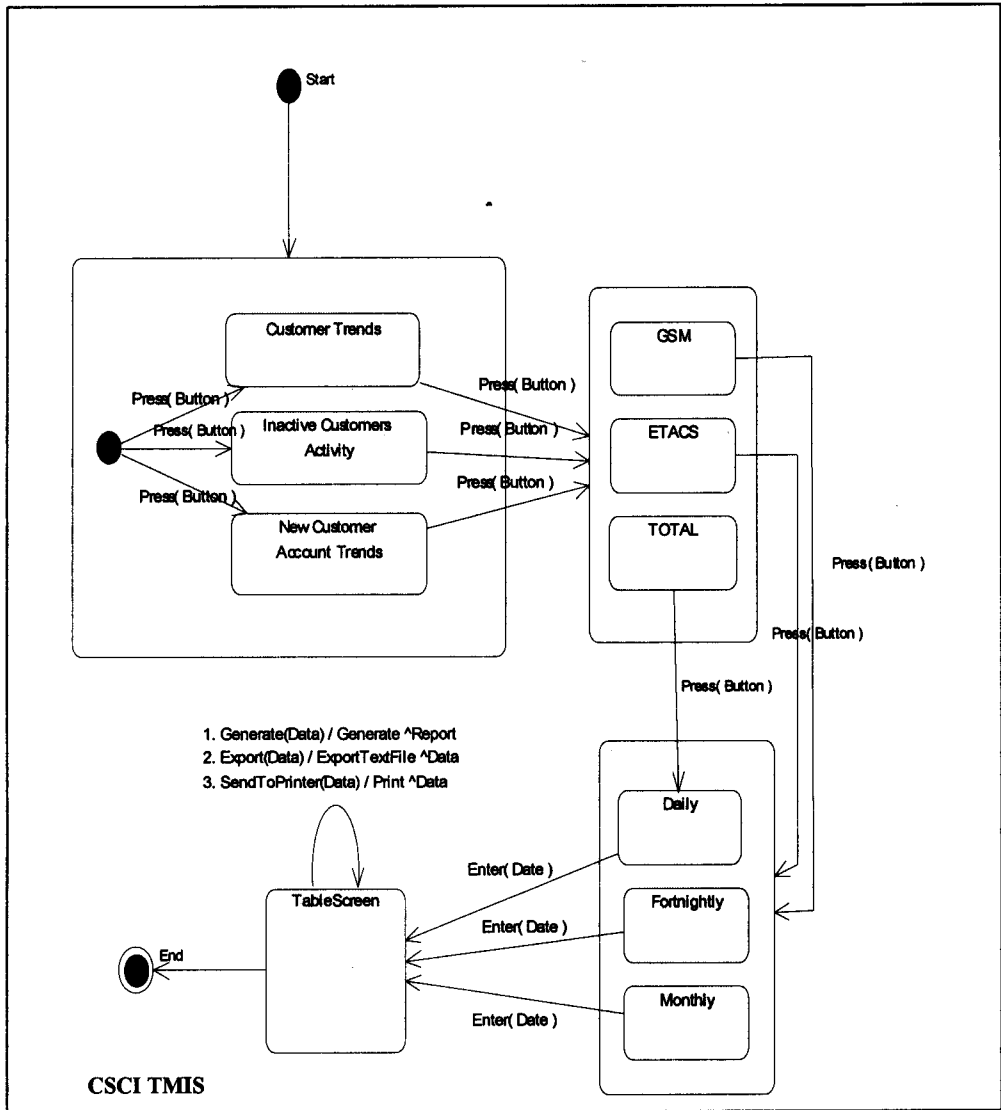


Figure 5 Statechart Diagram of System

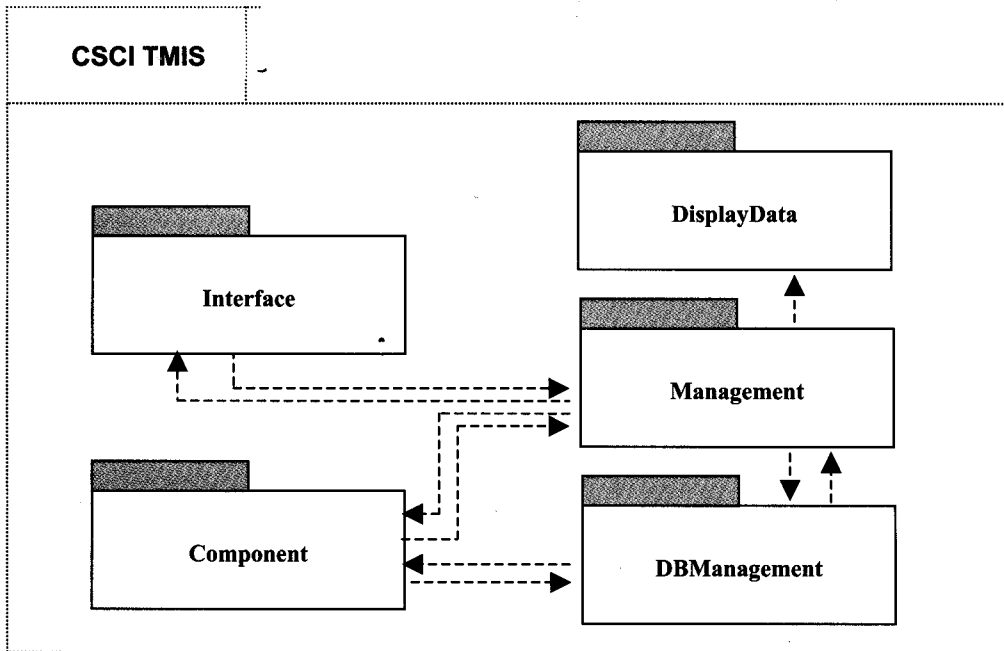


Figure 6 Relationship Between The Packages of System

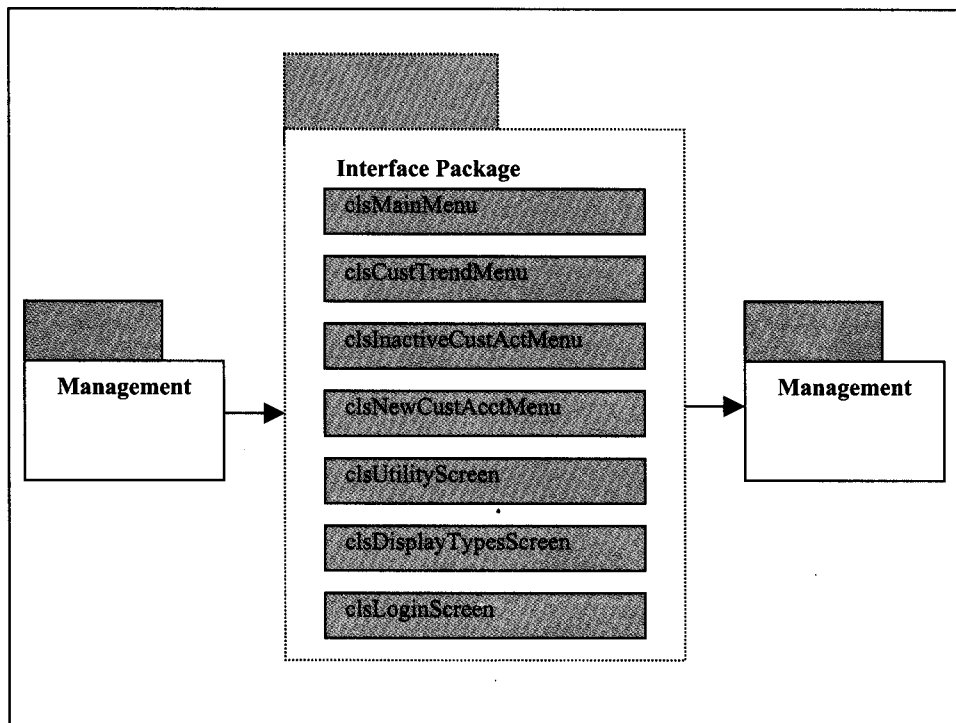


Figure 7 Visibility of interface Interface Package With Other Packages

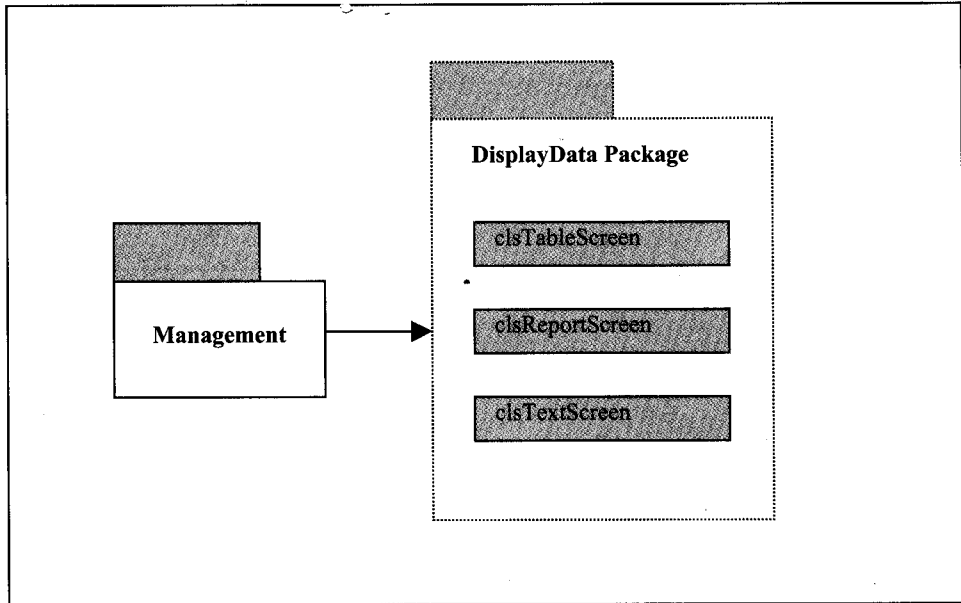


Figure 8 Visibility of DisplayData Package With Other Packages

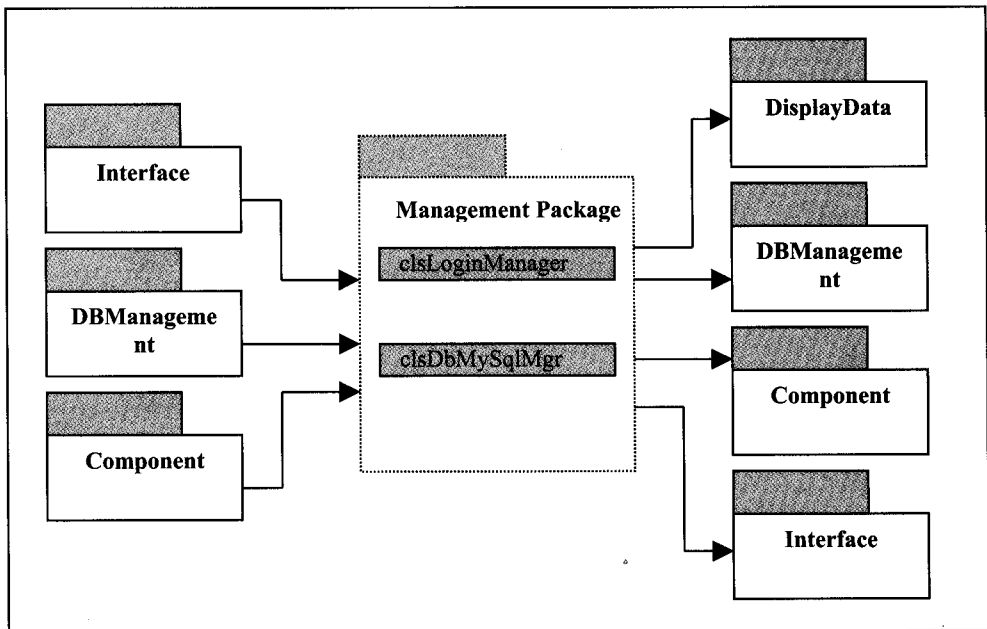


Figure 9 Visibility of Management Package With Other Packages

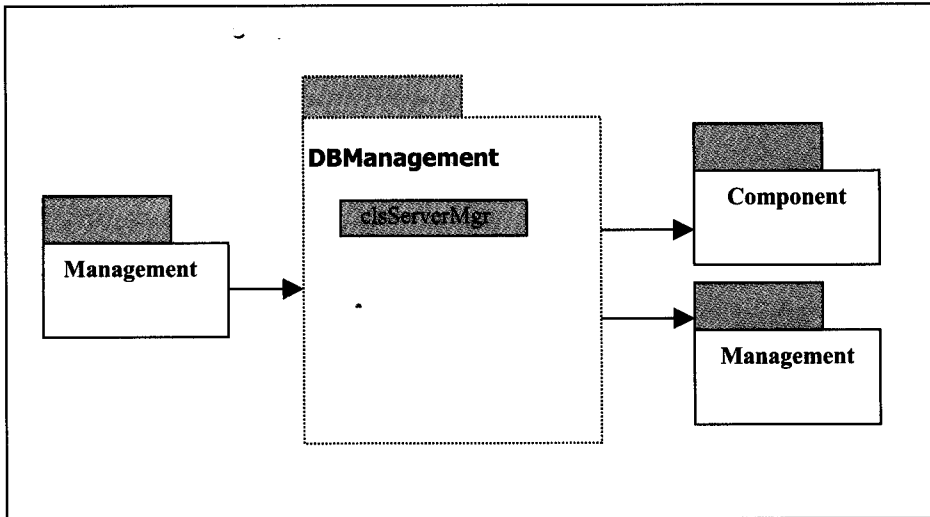


Figure 10 Visibility of DBManagement Package With Other Packages

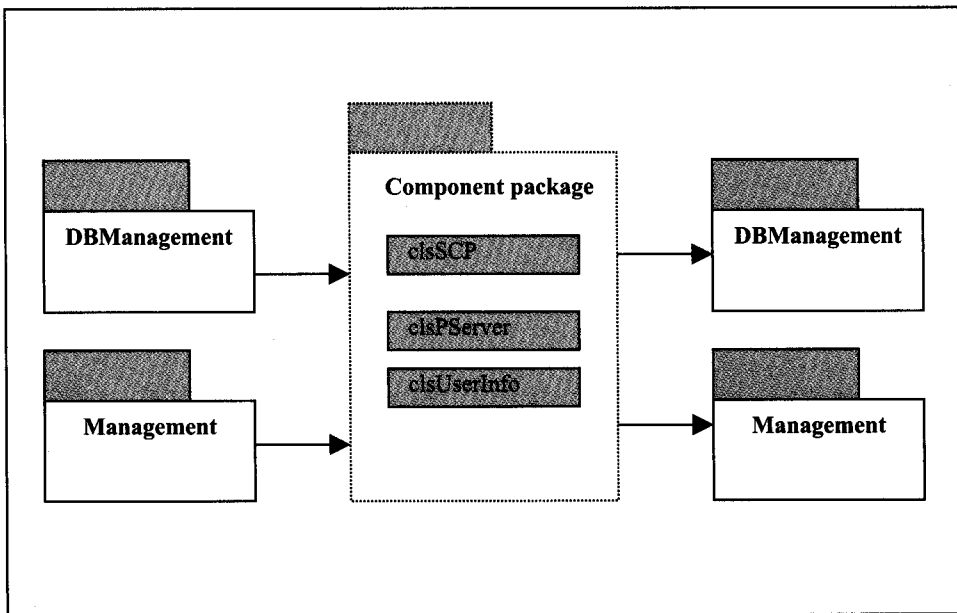


Figure 11 Visibility of Component Package With Other Packages

Description	Package	Class	Programming Language		
			HTML	MySQL	PHP
Interface		clsMainMenu	✓		
		clsCustTrendMenu	✓		
		clsInactiveCustActMenu	✓		
		clsNewCustAcctMenu	✓		
		clsUtilityScreen	✓		
		clsDisplayTypesScreen	✓		
		clsLoginScreen	✓		
DisplayData		clsTableScreen	✓		
		clsReportScreen	✓		
		clsTextScreen	✓		
Management		clsLoginManager	✓	✓	✓
		clsSysController	✓	✓	✓
		clsDbMySqlMgr	✓	✓	✓
DBManagement		clsServerMgr	✓	✓	✓
Component		clsSCP		✓	
		clsPServer		✓	
		clsUserInfo		✓	

Figure 12 Relationships Classes With The Programming Languages Used For

```

<HTML>
<BODY>
<? PHP
.....
.....

if (!isset($PHP_AUTH_USER)) {
    echo "Authenticate: Basic realm="THIS IS PRIVATE STUFF ";
    echo "'HTTP/1.0 401 Unauthorized";
    <A HREF=http://2.84.1.210/~tmis/LoginMenu.php3</A>

else if (isset($PHP_AUTH_USER))
    $db=@mysql_connect("localhost", "root");
    $sql = "SELECT * FROM UserInfo
           WHERE cId='$PHP_AUTH_USER' and
           cPasswd='$PHP_AUTH_PW'";
    $result = mysql_query($sql);
    $num = mysql_numrows($result);

    if ($num != "0") {
        echo " You're authorized! ";
        <A HREF=http://2.84.1.210/~tmis/MainMenu.php3</A>

    else
        echo "WWW-Authenticate: Basic realm=\"AUTH
        REQUIRED\" ";
        echo "HTTP/1.0 401 Unauthorized";
        <A HREF=http://2.84.1.210/~tmis/LoginMenu.php3</A>

?>
</BODY>
</HTML>
    
```

Figure 13 Fragment Code Of Authentication Process

Abbreviation

ASP	:	Active Server Pages
COM	:	Microsoft Component Object Model
OOAD	:	Object-Oriented Analysis and Design
ESN	:	Electronic Serial Number
ETACS	:	Extended Total Access Communications System
GSM	:	Global System for Mobile Communications
HTML	:	Hyper Text Markup Language
TMIS	:	Technical Management Information System
OMG	:	Object Management Group
PHP	:	Hypertext Preprocessor
SIM	:	Subscribers Identification Mobile
UML	:	Unified Modeling Language