Development of Staff Management System Using UML-Based Object-Oriented Approach

Rosziati Ibrahim, Abdulrahman Ahmed, Ida Aryanie Bahrudin,
Abubaker Ali Moftah and Elhadi Elfitory Algarai
Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia (UTHM)
Parit Raja, Batu Pahat, Johor, MALAYSIA
rosziati@uthm.edu.my

ABSTRACT

Management Information System (MIS) of an organization includes software systems as well as business processes and resources. The entire system of MIS is designed so that the organization will meet its specific aims. One of the most important systems that are very important to the organization is the system that can provide detail profiles of each staff. As a result, the staff management system has become one of the most important systems for improving the human resource management of any organization. In order to develop a system, there are many approaches that can be implemented. Studies had shown that by applying Unified Modeling Language (UML), many advantages can be gathered. Therefore, this paper describes the development of staff management system (SMS) using UML-based object-oriented approach. UML diagrams such as use-case diagram, class diagram and activity diagram are used prior to its implementation.

KEYWORDS

Unified Modeling Language (UML), use-case diagram, class diagram.

1 INTRODUCTION

Every organization, whether big or small, has human resource challenges to overcome. One of the most important ways to overcome the human resource challenge is through the implementation of staff management system (SMS). A staff management system which is usually called as human resources management system (HRMS) is the traditional foundation of a company's human resource technology. The role of HRMS is to centralize the repository of staff data. It is also refers to the systems and processes at the intersection between human resource management and information technology [1]. SMS usually include the details of staff profiles. An effective SMS helps by providing the technology to generate accurate and timely employee information to fulfill this objective [2].

In order to develop the system there are many system development methodologies that can be implemented. One of the most popular approaches is by applying UML in object-oriented environment. One of the reason of choosing UML because it will help communication between all participants in development process which will be contributed to the increasing probability of achieving successful developed system [3]. Therefore, this paper will describe the UML specification of SMS. Adopting the same role as HRMS [1], SMS is used for centralizing the details information regarding the staff including his/her salary and others.

The rest of the paper is organized as follows. Section 2 is the related works followed with UML specification in Section 3. Section 4 describes the experimentation done using mapping between UML specification into implementation. Finally Section 5 concludes this paper.

2 RELATED WORKS

Unified Modeling Language (UML) is accepted today as an important standard for developing software. It is also an object-oriented software design tool, providing all the features a software developer would need when designing an object oriented software system. UML includes 9 diagrams for describing system which are class diagram, object diagram, use-case diagram, sequence diagram, state diagram, collaboration

diagram, activity diagram, component diagram and deployment diagram [4].

UML specification is then mapped into the implementation by using the object programming language such as C++ language. C++ is an excellent language to start programming as there are a lot of applications that we use are usually written in C++. The C++ programming languages have been the most important higher programming languages for years, and they are still indispensable, when it comes into applications program which have to perform extremely efficient or that interoperate closely with the operating system. All these functions are possible to be implemented with C++ by direct access to memory manipulating functions.

There are several advantages of using the staff management system (SMS) such as:

- Providing a comprehensive information picture as a single, comprehensive database; this enables organizations to provide structural connectivity across units and activities and increase the speed of information transactions [5].
- Improving employee satisfaction by delivering human resource services more quickly and accurately to them [6].

There are many examples of staff management system that are available in the web. The followings are some of the review of existing staff management system.

2.1 OrangeHRM

OrangeHRM offers a flexible and easy to use HRMS solution for small and medium sized companies free of charge. By providing modules for personnel information management, employee self service, leave, time & attendance, benefits and recruitment companies are able to manage the crucial organization asset - people. The combination of these modules into one application assures the perfect platform for re-engineering and aligning your HR processes along with the organizational goals [7].

2.2 Employee Management Suite VI

This software allows user to edit employees, add new employees, transfer/promote/terminate employees and archive employees. Each employee in the database is associated with a position, which user can add/edit as well. If the users archive an employee, they can see what position he/she was associated with at the time of termination. User can transfer employee between positions easily without having to retype their information back into the database. The user can also check to see if there are duplicating position/employees and others [8].

3 UML SPECIFICATION

UML specification is used to specify the requirements of any system. In this paper, we will focus on 3 diagrams which is use-case diagram, class diagram and activity diagram for specifying the requirements of Staff Management System (SMS).

3.1 Use-Case Diagram

Figure 1 shows the use-case diagram for SMS. There are 2 actors involved in this system which are the administrator and also the staff. There are 5 use cases/functions in SMS which consists of Login, AddDetails, DisplayDetails, UpdateDetails and also DeleteDetails.

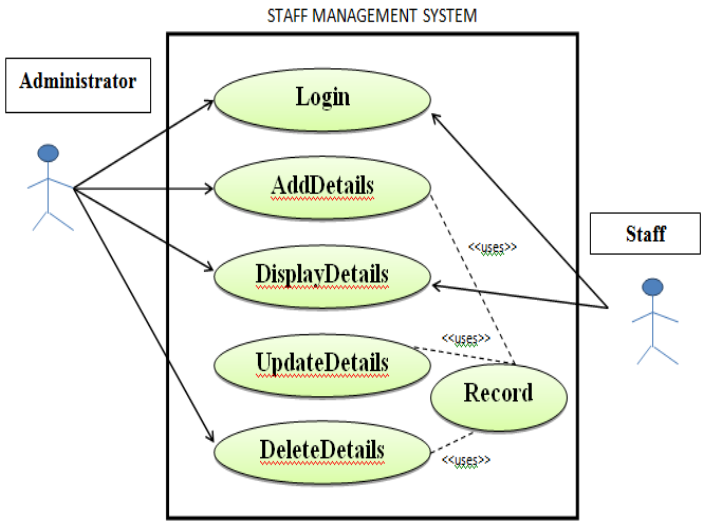

Figure 1: Use-case Diagram for SMS

3.2 Class Diagram

Based on the use-case diagram for the Staff Management System (SMS), there are 4 classes that had created. The classes are Staff, Activity, System and StaffInfo. The attributes of class Staff are Password, ID, Name, Salary, MaritalStatus and also JobTitle. For each attributes, Set and Get options will be the methods/services for this class.

The class Activity inherits attributes and functions from class Staff where all the functions and attributes are dependent to the class Staff. The functions of class Activity are AddDetails, DisplayDetails, UpdateDetails, DeleteDetails, GetActivityType and GetAnotherAct. Figure 2 shows the class Activity that inherits from class Staff. There are no attributes in class Activity as it inherits all the attributes from class Staff. There are also 2 more functions that added to class Activity which are GetActivityType and GetAnotherAct. The function GetActivityType is used for user to select the type of activity from the menu screen while the function GetAnotherAct is used to give an option to user whether to continue using the system or not.

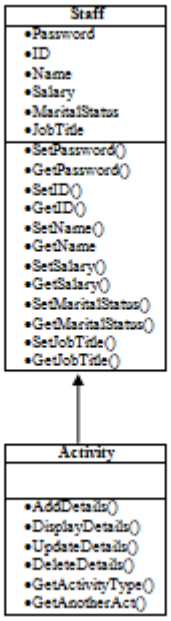

Figure 2: Class Activity using Inheritance Concept

Class System acts like a task that SMS supposed to perform. Therefore, class System can have functions such as ReadFile, Login and also Menu, with the Data as attribute of the class. Class StaffInfo perform all the functions to store the staff information entered by Administrator that will be viewed later by Staff. So, class StaffInfo will act as simple file processing for SMS. The functions that included in class StaffInfo is SetData and GetData. By putting all the classes together, the complete class diagram for SMS is shown in Figure 3 where class Staff is a superclass with class Activity is the subclass (Adapted from [9]). While there are 2 more classes which are class System and class StaffInfo.

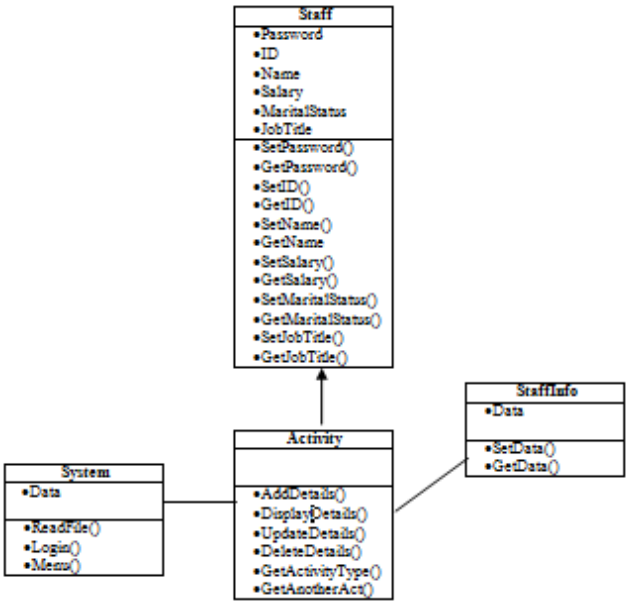

Figure 3: Class Diagram of SMS

3.3 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In UML, activity diagrams are intended to model both computational and organizational processes. Based from use-case diagram in Figure 1, two actors have been identified for SMS. These two actors represent two users that are going to use the system. They are staff and administrator. Therefore, the activity diagram of SMS consists of these two activities. Figure 4 shows the activity diagram of staff in SMS and Figure 5 shows the activity diagram of administrator in SMS. Based

from Figure 4, every staff needs to login to the system before he/she can perform any function in the system. SMS will verify the staff ID and the password. If the password entered is incorrect, system will display an error message where staff has to re-login or exit the system. There is only one menu for staff which is display staff details.

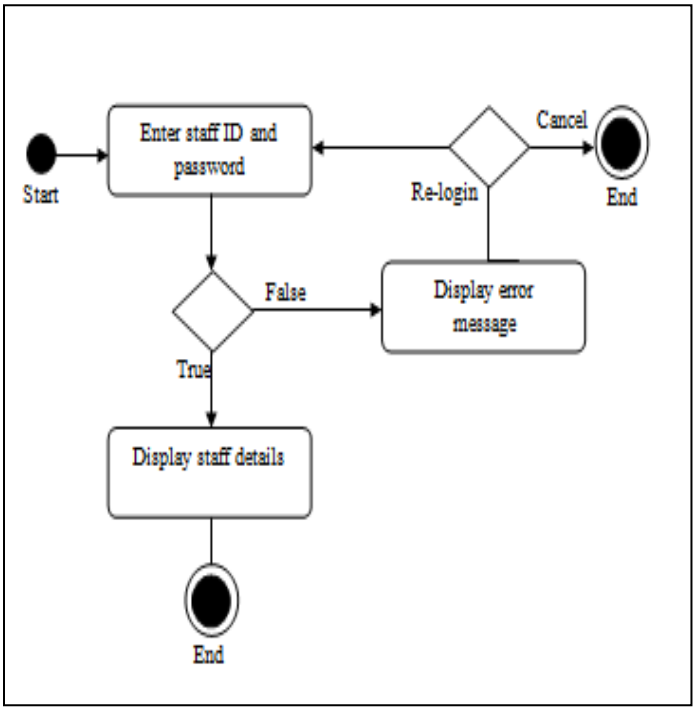

Figure 4: Activity Diagram for Staff

Figure 5 shows the activity diagram for SMS administrator. Similar to the staff activity, administrator also needs to login to the system using staff ID and password. Once the login process successes, there are 4 options of functions that can be chosen by the administrator. The functions are Add staff details, Update staff details, Display staff details and also Delete Staff Details.

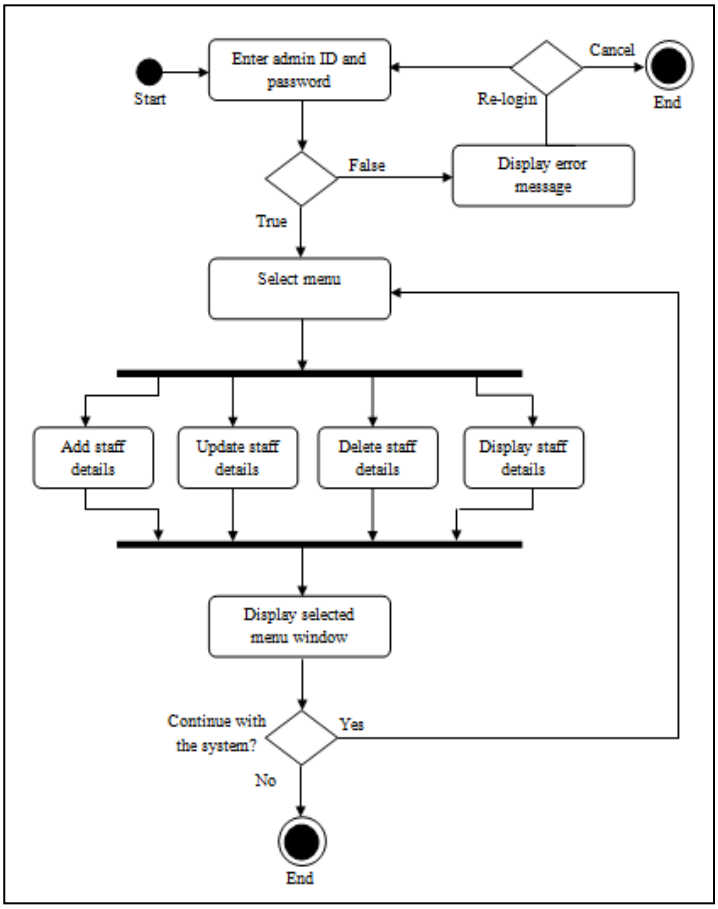

Figure 5: Activity Diagram for Administrator

4 MAPPING FROM UML SPECIFICATION INTO SOURCE CODES

After all the UML diagrams had been created, the UML specification will be mapped into the implementation using C++. Mapping from UML specification into implementation involves using UML diagrams and converting them into program source codes. The mapping is shown in Diagram 1 until Diagram 4. The idea of mapping from UML specification into source codes is adopted from [10].

Diagram 1: Mapping of Class Staff into Implementation

Analysis	*Implementation*
Staff	**class** Staff {
• Password • ID • Name • Salary • MaritalStatus • Job	**private:** **char** Password[7]; **char** ID[10]; **char** Name[20]; **float** Salary; **char** MaritalStatus[10]; **char** Job[12];

• SetPassword
• GetPassword
• SetID
• GetID
• SetName
• GetName
• SetSalary
• GetSalary
• SetMaritalStatus
• GetMaritalStatus
• SetJob
• GetJob
	public:
	Staff();//constructor
	void SetPassword(**char** ResPassword[7]);
	char *GetPassword();
	void SetID (**char** ResID[10]);
	char *GetID();
	void SetName (**char** ResName[20]);
	char *GetName();
	void SetSalary (**float** Sal);
	float GetSalary();
	void SetMaritalStatus (**char** ResMaritalStatus[10]);
	char *GetMaritalStatus();
	void SetJob (**char** ResJob[12]);
	char *GetJob();
	}; //class Staff

Diagram 1 shows the mapping of class staff into source codes. Based from class diagram in Figure 3, class staff consists of attributes and methods for staff. The attributes are translated into data and methods are translated into functions. In implementing class using object-oriented approach, all data are declared private and all methods are declared public. The class visibility is discussed in [9].

Based from Figure 4, class Activity inherits from class Staff. The inheritance concept for class is then translated into implementation of the source codes. Diagram 2 shows the mapping of class Activity into the source codes.

Diagram 2: Mapping of Class Activity Using Inheritance into Implementation

Analysis	*Implementation*
Activity	**class** Activity: public Staff{ //inheritance

• AddDetails
• DisplayDetails
• UpdateDetails
• DeleteDetails
• GetActivityType
• GetAnotherAct
	public:
	Activity();//constructor
	void AddDetails();
	void DisplayDetails();
	void UpdateDetails();
	void DeleteDetails();
	int GetActivityType();
	char GetAnotherAct();
	}; //class Activity

Based from Diagram 2, note that because of inheritance mechanism, subclass Activity inherits all attributes and methods from its superclass Staff. Therefore, all methods from superclass Staff are visible in class Activity.

The other two classes from class diagram in Figure 3 can also be mapped into source codes as illustrated in Diagram 3 and Diagram 4. Diagram 3 shows the mapping of class System into its source codes and Diagram 4 shows the mapping of class StaffInfo into its source codes.

Diagram 3: Mapping of Class System into Implementation

Analysis	*Implementation*
System	Class System {

• Data	private:
	struct Data {
	//Data for information details for user
	char password[7];
	char ID[10]; } dat;

• ReadFile • Login • Menu	**public**: **void** ReadFile(); **void** Login(); **void** Menu(); }; // class System

Diagram 4: Mapping of Class StaffInfo into Implementation

Analysis	*Implementation*
StaffInfo	Class StaffInfo {
• Data	**private**: **struct** StaffInfoData { **char** Name[25]; **char** StaffDetail[25]; } stdata;
• SetData • GetData	**public**: **void** SetData(); **void** GetData(); }; // class StaffInfo

Once the mapping is done and the source codes for all the classes have been implemented, the object can be created from the class. In C++, an object is instantiated from a class. Figure 6 shows an example of object instantiation and options that can be performed based on the two users: administrator and staff.

```
int main() {
StaffInfo employee1; //object instantiation
System  admin1;
 int ch;
  cout<<"\n--------------------------";
  cout<<"\n press [1] if you admin ";
  cout<<"\n press [2] if you staff";
  cout<<"\n press [3] if you want exit";
  cout<<"\n--------------------------";
  cout<<"\n enter your choose ---------->";
  cin>>ch;
   switch(ch) {
     case 1:  admin1.login();
              break;
     case 2 : employee1.login();
              break;
     case 3 : exit(0);
              break;
     default:
               system("cls");
               cout<<"\n ----------------------";
               cout<<"\n Enter Valid choice";
               cout<<"\n ----------------------";
              cout<<endl;
    }
```

Figure 6: Example of Source Codes for Main Program

5 CONCLUSIONS

Mapping from design into implementation is an important process in software development life cycle. This study had shown that UML specification is an important specification to ensure the completeness of an information system as well as mapping its design into source codes. C++ is one of the object-oriented programming languages that are best suited to use for developing simple staff management system. This SMS shows an example of effective way in organizing staff details especially for small and medium company. The combination of all modules into one application will ensure the perfect platform for reorganizing human resource processes along with the organizational goals.

6 ACKNOWLEDGEMENTS

We would like to thanks Universiti Tun Hussein Onn Malaysia (UTHM) for supporting this research study.

7 REFERENCES

[1] A. S. Syed Navaz, A. S. Syed Fiaz, C. Prabhadevi, V. Sangeetha, and S. Gopalakrishnan, "Human Resource Management System," IOSR Journal of Computer Engineering, Vol. 8, Issue 4, pp. 62-71, Jan 2013.

[2] M. Thite and M. J. Kavanagh, Human Resource Information Systems: Basic, Application and Future Directions. Sage Pub. Inc, 2012.

[3] S. Dessic, D. Gvosdanovic, M. Kusek and D. Huljenic. "Advantages of UML-based Object-oriented System Development Functions," Retrieved from http://bib.irb.hr/datoteka/71601.CTE33.pdf (Search Date: December 12, 2013).

[4] T. Weilkiens and B. Oestereich. UML 2 Certification Guide: Fundamental and Intermediate Exams. The MK/OMG Press, 2010.

[5] C. A. Lengnick-Hall and M. L. Lengnick-Hall. HR, ERP, and knowledge for competitive. 2006.

[6] A. M. Beckers, and M. Z. Bsat, A DSS Classification Model for Research in Human Resource. Information Systems Management, Vol. 19 No. 3, pp. 41-50. 2002.
A Complete HRIS –OrangeHRM Product Features. http://www.orangehrm.com/open-source-product-features-pim.shtml (Search Date: December 17, 2013).

[7] Employee Management Suite V1. http://empsuite.codeplex.com/. (Search Date: December 17, 2013).

[8] R. Ibrahim. An Introduction to Object-Oriented Programming with UML Using Borland C++. Malaysia: UTHM Publications, Batu Pahat, 2008.

[9] R. Ibrahim "From UML Specification into Implementation using Object Mapping," Computer and Information Science, Vol 3, No 3, August 2010, pp.149-157.