

Designing a Task Management System for a Banking System by Combining Relational Model with Use Case Diagram

Sohrab Hossain
Assistant Professor
University of Science and
Technology Chittagong
USTC, Foy's Lake, Chittagong-
4202. Bangladesh.

Razuan Karim
Associate Professor
University of Science and
Technology Chittagong
USTC, Foy's Lake, Chittagong-
4202. Bangladesh.

Dhiman Sarma
Lecturer
Feni University
Trunk Road, Feni 3900,
Bangladesh

ABSTRACT

This paper prompts to understand the analysis by using use case diagram and design by relational model of a task management system for a banking system. A soundly design task manager may play an important role to keep track of all assign tasks, pending tasks, completed tasks, due tasks, and impossible to complete tasks. In addition, it will help to get an instant list of all tasks, promote tasks efficiency and watch against missed deadlines, Balance personal workloads, Forecast work bottlenecks and promote consumer satisfaction with regular suggested deadline dates. Before designing a task manager, the requirement analysis should be accomplished by interviewing the users or reviewing the current system.

General Terms

Database Management System; Relational Database Management System

Keywords

Task management system; banking task manager; relational task management system, relational model, use case diagram;

1. INTRODUCTION

This report based on the task management system of Information Technology (IT) Division of Southeast Bank Limited. IT Division has five subdivisions: Help Desk, Business Critical Application Support Team, Core System Team, Network Management, and Software Development and each subdivision are responsible for different types of tasks. This paper mainly focuses on the analysis and design of the system. As an initial investigation, feasibility study is conducted to check whether a new system is organizationally, functionally, technically, and operationally feasible. To understand the both user and system requirement, a number of interviews are conducted with the system users and customers. However, the existing manual system is reviewed to understand the user requirements. For designing purpose, unified modeling language (UML), Entity Relationship Diagram, and Data Flow Diagram are used. Basically, use case, nonfunctional, class diagrams, and sequence diagram are used form UML. Finally we talk about data and system constraints and future update.

2. RELATED WORKS

2.1 Task management Software

A task management software prevents losing track of tasks, projects or work. It can easily manage and track all tasks, projects and jobs. Task Manager helps organizations to not

forget, giving them an instant snapshot of work they have outstanding. [7]

Task Manager is designed to help organizations get organized and stay organized with tasks. Whether they use it in a team/network environment or personally, Task Manager provides them with a very fast database environment, built-in task reports, and the ability to include and modify all of tasks according to their specific needs [7].

2.2 Interviews

Information-gathering tools are categorized by their degree of directness. If we want to know about something, we simply ask someone about it directly, but we might not get an answer. Almost all information gathering tools used in software analysis are relatively direct. The interview is a face-to-face interpersonal role circumstances in which a person or user called the interviewer asks a person being interviewed questions designed to gather data about proposed system. The interview is one of the oldest and most often used tools for gathering information systems work. It has qualities that behavioral and onsite observations do not possess. It is used for two main purposes: (1) as a tool to identify relations or verify information (2) to collect information as it [1].

2.3 Materials

Very few system design problems are unique. The increasing number of software system packages suggests that problem solutions are becoming standardized. Therefore, as a primary step, a search through certified references and manuals, books, company studies, government publications, or consultant studies may prove invaluable. The primary problem of this search is time. Often it is difficult to get expensive reports, publications, and the information may be outdated due to a time constraint in publication [1].

2.4 Unified Modeling Language

The unified modeling language (UML) is used for visualizing, specifying, constructing, and documenting the artifacts of a software system graphically. The UML is de facto standard to write a system's blueprint. It covers conceptual things, such as classes written in an object oriented programming language, relational database scheme, and reusable software components. UML was meant to be a unifying designing tool enabling IT professionals to model software systems. There are many types of UML diagram. We used only two of them: Use Case Diagram (for functional requirements) and Class Diagram (for structural modeling) [2].

2.5 Entity Relationship Diagram

The Entity-Relationship Diagram model is a conceptual data model developed by Chen (1976) to facilitate database design. A conceptual data model describes the structure of a database and the associated retrieval and update transactions on the database. The main purpose for developing a high-level data model is to support a user's perception of the data, and to conceal the more technical aspects associated with database design. Furthermore, a conceptual data model is independent of the particular Database Management system (DBMS) and hardware platform that is used to implement the database [6].

Entity

An object or concept or thing is identified by the enterprise as having an independent existence in the real world is called an entity type. Each uniquely identifiable instance of an entity type is known as an entity [4].

Relationship

An association among entity types is called relationship types. Each uniquely identifiable occurrence of a relationship type is known as a relationship [4].

Attributes

The properties of entities are known as attributes. For example, a Branch entity may be described by the branch number (Branch_ID), address (Address), phone number (Tel_no), and fax number (Fax_no) [4].

2.6 Relational Model

The relational model has a collection of tables. The tables represent both data and the relationships among those data. Each table name is unique to a database schema. Each table contains multiple columns, and each column has a unique name in a table. Relational model uses three keys to maintain relationship among data. They are primary key, composite key, and foreign key [6]. A primary is an attribute that can uniquely identifies each row in a relational table. A composite is a primary key that needs of two or more attributes to uniquely identify each row. Foreign key is used to represent one-to-many relationship in relational model, is an attributes of one table that can serve as a primary key of another relational table in the same database [5].

3. DESIGNING THE TASK MANAGER

Designing the database consists of three stages: information gathering, requirement analysis and relational model design.

3.1 Information Gathering

In this stage, information about the proposed system has been collected through interviews and reviewing existing documents. In addition, we look for reusable software resources through the Internet.

3.1.1 Interviews

We mainly conduct interviews from five officers from different subdivisions of IT division: Mr. Muhibur Rahman (Help Desk), Mr. Zahid Morshed (Network Management), Mr. Mahabubur Rahman (Core Systems Team), Mr. Shakhawat Hossain (Business critical application support team), Mr. A.K.M. Monirul Islam (Software Development Team). A brief summary of these interviews are given below:

a) Mr. Muhibur Rahman (Help Desk)

All branches can directly access help desk by phone if they face problem. Help Desk try to solve the problem over phone. If job is very tedious, they just pass it to appropriate Sub-division. The major job of this team is to satisfy any technical queries over phone. Some minor tasks are assigning

responsibility to relevant team & observing progress and update. Other responsibilities are creating user account, maintaining user account. However, they administer user accounts of active directory network, email, system logon account. In addition to that they are responsible for paper works regarding purchase, payments, requiring, network account request, general ledger account opening, stock maintenance and others. For example, a network problem occurs in Principle Branch. Principal Branch calls the IT Division's Help Desk. Help Desk try to solve the problem over phone. If they failed to solve the problem, they pass it to network management team. In addition they set duration for that work. If network management team failed to solve the problem on time, the problem goes to delay queue.

b) Mr. Zahid Morshed (Network Management)

Help Desk assign tasks to Network Management. They design and deploy Active Directory Service architecture for establishing a Centralized Domain Controller as per organizational Diagram of the Bank. In addition, they configure a file server for preserving branches data and the divisional Data through LAN and WAN with a security logon as well as backup regularly. For example, a network problem occurs in Principle Branch. Principal Branch calls the IT Division's Help Desk. Help Desk try to solve the problem over phone. If they failed to solve the problem, they pass it to network management team. In addition they set duration for that work. If network management team failed to solve the problem on time, the problem goes to delay queue. Or requisitions may come from branches and this team work according to the demand of that requisition. For example, a requisition from Principle Branch comes requesting for the Internet connection a new computer.

c) Mr. Mahabubur Rahman (Core Systems Team)

Help Desk assign tasks to Core System Team. This team mainly deals with installation, monitoring, and performance optimizing Windows 2000/XP, Antivirus, Office Application etc on Servers and PCs of Head Office as well as Branches. They also configure and troubleshoot of printers and networks of servers and PCs, email, client configuration of banking software, centralized file access, printers, backup and restore of servers and PCs. They also play role as 2nd line support maintenance of server, PCs & negotiate with 3rd parity vendors in case any critical issue arises.

d) Mr. Shakhawat Hossain (Business critical application support team):

Help Desk assign tasks to Business critical application support team. This team mainly deals with installation, monitoring, performance optimizing and operation of banking software (PcBank2000), Security auditing. However, They are responsible for critical business activities, advising users on complex procedures and take necessary steps to eliminate critical system related issue with vendors.

e) Mr. A.K.M. Islam (Software Development Team)

This team spends most of their time for developing in-house software system. However, they test newly acquired software and evaluation of new software system, after sale support and cost etc and provide 3rd line support to systems developed in-house. For example, a new version of PcBank2003 is acquired. Now the job of this teams start. They make test data for black box testing. Try to get any fault of that software. If IT Division get green signal for this team, they purchase the software. This team has no direct contact with help desk. Sometimes they develop software for domestic use. For example, Mr. A.K.M. Monirul Islam develops Inventory Management Software.

3.1.2 Materials

We collect data from Annual Report 2003, incoming, outgoing letters and Human Resource Department.

f) Annual Report 2003

We look at Annual Report 2003 for formal information. We do not get any information that is directly related to our project. But we got some ideas about the interfaces and color combination they use in the report. However, we got the format and style of the report they are habituated. All the reports are very simple and user friendly. There is no excess of color.

g) On-line Branch Code & IP Address

In addition, we also go through On-line Branch Code & IP Address to collect the details information about Branches. To know the maximum and minimum length of branch name, code, address, IP Address, Phone Number.

h) Employee's PABX number Index

To collect details information about employees, we go through the Employee's PABX number Index In this form, employee's name, designation, phone number are collected. However, the maximum and minimum lengths of these fields are found here.

3.1.3 Reusable Software Resources

We search the Internet and the organization for reusable software resources. Software engineering emphasizes reusability-that is, the creation and reuse of software building blocks. Such building blocks, often called components, must be categories for easy reference, standardized for easy application, and validated for easy integration. We considered four software recourse categories: off-the-shelf components, full experience components, partial-experience components, and new components.

a) Off-the-shelf components

Existing software that can be acquired from a third party or that has been developed internally for a past project. COTS (commercial off-the-shelf) components are purchased from a third party, are ready for use in the current project, and have been fully validated. We found an off-the-shelf component at <http://www.orbisoft.com>. Task manager 2012 is a readymade software that can be easily use by the bank. We get limited version of Task manager 2005 and user manual of that software. The full version of that software is very costly (\$500). Our bank is not ready to spend that much of money to that software

b) Full-experience components

Existing specifications, designs, code, or test data developed for past projects that are similar to the software to be built for the current project. Members of the current software team have had full experience in the application area represented by these components. Therefore, modifications required for full-experience components will be relatively low cost. Unfortunately, we do not get such software components [3].

c) Partial-experience components

Existing specifications, designs, code, or test data developed for past projects that are related to the software to be built for the current project but will require substantial modification. Members of the current software team have only limited experience in the application area represented by these components. Unfortunately, we do not get such software components [3].

3.2 Requirement Analysis

After collecting data about the proposed system, a use case diagram and a class diagram are used to represent the user requirement graphically.

3.2.1 Use case Diagram

The task management system use case allows a user or an administrator to assign, update, search, insert, monitor and delete tasks.

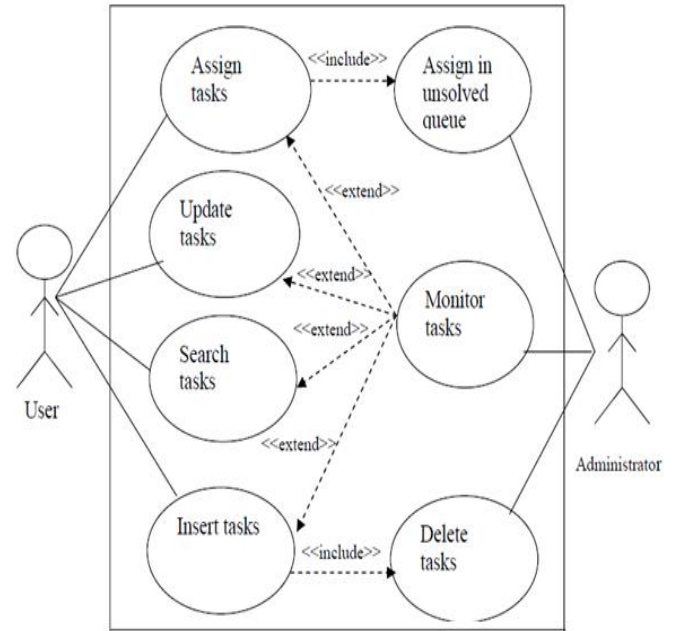


Figure 1. Use Case Diagram for the Task manager

Primary Flow:

1. The use case begin when a user open the Task Manger.
2. The form presents the following six options
 - ✓ Assign tasks
 - ✓ Update tasks
 - ✓ Search tasks
 - ✓ Insert tasks
 - ✓ Monitor tasks
 - ✓ Delete tasks
3. The user enters required information and selects Assign tasks.
4. The system assigns tasks to relevant subdivision.
5. If the user enters any irrelevant and insufficient data, Alternative Flow A1 is performed.
6. The user enters required information and selects Update.
7. Information is updated in database.
8. If the user enters any irrelevant and insufficient data, Alternative Flow A1 is performed.
9. The user enters required information and selects Search tasks.
10. The system will search the required information and display on the screen.

11. If the user enters any irrelevant and insufficient data, Alternative Flow A1 is performed
12. The user enters required information and selects Insert tasks
13. Information is stored in database.
14. If the user enters any irrelevant and insufficient data, Alternative Flow A1 is performed.
15. The user enters required information and selects Monitor tasks.
16. The software prompts to enter administrative password.
17. The user enters the password.
18. The software conforms that the password is valid. If the password is not valid, Alternative Flow A2 is performed
19. Proper information about the task is collected and displayed to the Administrator.
20. If the user enters any irrelevant and insufficient data, Alternative Flow A1 is performed.
21. The user enters required information and selects Delete.
22. The software prompts to enter administrative password.
23. The user enters the password.
24. The software conforms that the password is valid. If the password is not valid, Alternative Flow A2 is performed
25. If the user enters any irrelevant and insufficient data, Alternative Flow A1 is performed.
26. The software delete information form database.
27. The use case ends.

Alternative Flow

Exceptional Flow of events (A1):

1. The software notifies the user that the information is not correct or insufficient.
2. The use case restarts.

Exceptional Flow of events (A2):

1. If the user enters invalid passwords two times in a row, the software notifies the user that password is not correct.
2. The use case restarts.
3. If the user enters invalid passwords three times in a row, the software notifies

the user to contract with administrator.

4. The use case ends.

Exceptional Flow of Events (A3):

1. The user can reset the form any time by pressing the reset button.
2. The use case restarts.

Exceptional Flow of Events (A4):

1. The user can exit at any time by pressing Control and Alter button at the same time.
2. The use case restarts.

Exceptional Flow of Events (A5):

1. The user can exit at any time by pressing the exit button.
2. The use case ends.

Exception Flow of Events (A6)

1. The user skips the field and press assign, update, search, insert, and monitor and deletes button.
2. The software ignores such operation.
3. The use case restarts.

3.2.2 Class Diagram

IT division of the bank has five classes named IT division, subdivision, employee, problem, and branch. Each class has their private and public data. The private data can be accessed only by the other friend classes by using the public methods of the particular class.

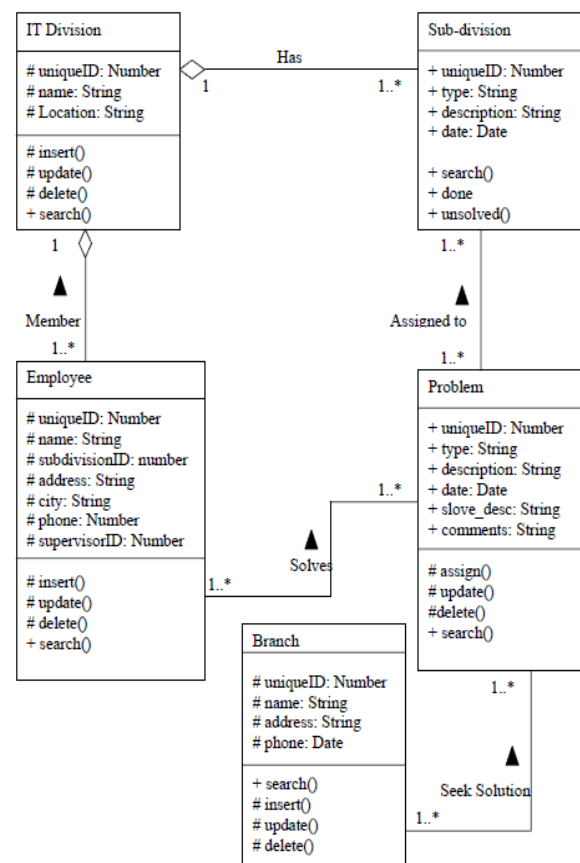


Figure 2. Class Diagram for the Task Manager

3.3 Relational Model

Two designing tools have been considered to represent the task manager graphically. They are entity relationship diagram (ERD) and relational model.

3.3.1 Entity Relationship Model

In this phase of the design, underlying structure of data is considered. Firstly, the entities are determined from the use case and class diagram. They are problem, branch, Help_Desk, Business_Critical, Application_Support_Team, Core_System_Team, Network_Management, and Software_Development, The relationship among the entities are shown in figure 3. All the relationship are binary relationships.

3.3.2 Relational Model

In this phase of the design, tabular form of entity is considered. In addition, the relationships among the entities are shown in the form of primary key and foreign key concepts in figure 4.

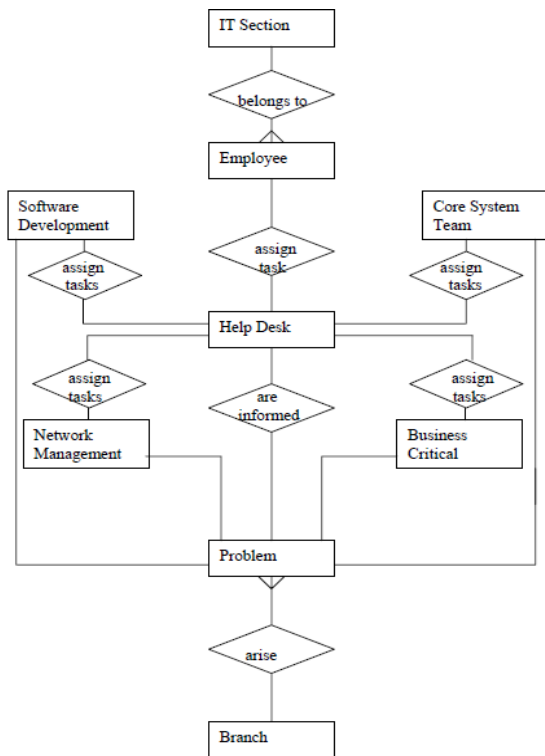


Figure 3. Entity Relationship Diagram for the Task manager

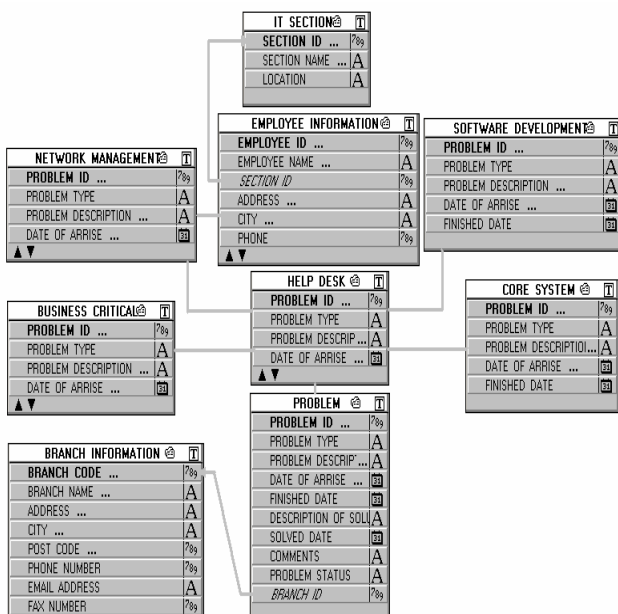


Figure 4. Relational Model for the Task manager

4. CONCLUSION

The major drawback of the designed software is that it can only keep track of tasks in a banking system. In addition, it stores same types of information for all subdivisions. But the subdivisions are very in their working areas. However, only use case diagram, class diagram, entity relationship diagram, and relational model are used to represent data and their relationship. The design does not show data flow diagram between source and storage. It tells nothing about the sequence of the tasks or state of the tasks.

5. ACKNOWLEDGMENT

Many individuals have contributed to the preparation of this report. For their invaluable support and suggestions, firstly, we wish to thank Ms. Samira Alam, Lecturer, School of Engineering and Computer Science, Independent University, Bangladesh. Secondly, we would like to express thanks to Mr. Ahmed Tabsir Choudhury, Senior Vice President and Head of Information Technology Division, Southeast Bank Limited (SEBL), without his support it would not be possible for us to learn so many things and work in a short time in an enterprise system. Finally, we thank Mr. Muhibur Rahman, Saiful Islam, Mr. Shakhawat Hossain, Mr. Moniruzzam ; employees of Southeast Bank Limited.

6. REFERENCES

- [1] E. M. Awad, Systems Analysis and Design, New Delhi, India: Galgotia Publications Pvt. Ltd, 2002.
- [2] G. Booch, J. Rumbaugh, I Jacobson , The Unified Modeling Language User Guide, New Jersey: Person Education, 2001.
- [3] R. S. Pressman, Software Engineering: A Practitioner’s Approach, New York: McGraw- Hill, 2001
- [4] H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems The complete Book, Upper Saddle River, New Jersey: Prentice-Hall, 2009.
- [5] P. Rob, C. Coronel, Database Systems Design, Implementation, and Management, United Kingdom: Thomson Course Technology, 2007
- [6] A. Silberschatz, F. Korth, S. Sudarshan, Database System Concepts, New York: McGraw-Hill, 2004.
- [7] (2013, February, 01) Task Management Software [online] 2012. Available: <http://www.orbisoft.com/index.htm>