Use Case Modelling of Bingham University Library Management System

[1]Faki A.S, [2]Ogedebe P.M.

[1,2] Department of Computer Science, Faculty of Science and Technology,
Bingham University, Karu - Nasarawa State, Nigeria
faki.silas@binghamuni.edu.ng. peter@binghamuni.edu.ng

Abstract

*With the advent of object oriented design, Unified Modelling Language (UML) has become prominent in software industry. Software is better modelled with the use of UML diagrams like **use cases** which provide a better flow of logic and comprehensive summary of the whole software system in a single illustration. In this paper, we are able to model a University library which shows clear flow of logic and makes for easy for coding of software.*

Keywords: *Actor, Use Case, UML, Login*

Introduction

The principal medium for capturing requirements in the Unified Process is the ***use-case*** diagram. Use-case diagrams illustrate the relationship between a software system and its users. They also describe how each type of user will interface with the system. In this paper, we illustrated ***use case*** by modelling a University Library Information System. The application of ***use case*** made the coding aspect of the software easier and also shows a clear flow of logic in the system. The rest of the paper is presented as follows: II Review of related literature, III Shows the Library Information System design using ***use case***, IV Present entity diagram from the ***use case*** and V The summary of work.

Review of Related Work

As object-oriented software engineering gains prominence in software practices, **use cases** have enjoyed a seemingly explosive growth to become ubiquitous in both development methods and development practice. Part of this ubiquity can be attributed to their utility; use cases have proved to be versatile conceptual tools for many facets of design and software development.

A use case defines a sequence of actions a system performs that yields an observable result of value to a particular actor. In other words, a **use case** describes primarily functional and also non-functional requirements from the perspective of an actor achieving particular goals. The goal of object oriented use case analysis and design according to [1] is to *transform the 'black box' use case specification into a 'white box' use case realization.*

As an effective bridge between usability engineering and user interface design on the one hand and software design and development on the other, part of the promise that **use cases** offer is due precisely to their chameleon-like adaptability [2]. For requirements engineering, use cases provide a concise medium for modelling user requirements; in the hands of user interface designers, use cases can become a powerful task model for understanding user needs and guiding user interface design; for software engineers, use cases guide the design of

communicating objects to satisfy functional requirements.

As observed by [2], most commonly used, use cases describe the actual interaction between external users (or system actors) and a system through a particular interface. Because they are expressed in concrete terms, such use cases are best referred to as concrete use cases. Also an essential use case can be looked upon as a single, discrete, complete, meaningful, and well-defined task of interest to an external user in some specific role or roles in relationship to a system, comprising the user intentions and system responsibilities in the course of accomplishing that task, described in abstract, technology-free, implementation independent terms using the language of the application domain and of external users in role.

[3] differentiates use case and use case scenarios by defining the first with three keys things, (i) the actor or actors involved (ii) the system being used (iii) the functional goal that the actor achieve using the system. Use case scenario is seen as different outcomes based on circumstances.

[4] describes use case as a technique for capturing and communicating functional requirements for software development. Use cases are written from the perspective of the user as a flow of events. The user is called an "actor" and the narrative of the flow of events between this actor and the system is called the "use case." Use cases are literally the specific "cases" for which the actor wants to "use" the system. Use cases differ from declarative statements in two primary ways. They describe functional requirements from the user perspective rather than the system perspective, and they provide a coherent goal focused flow of events rather than a set of discrete declarative statements. A fully described use case will have a main or basic flow as well as alternate flows [5]. The alternate flows describe regular variants to the main flow as well as error handling or unusual situations.

Library Management Design Using Use Case

To model software using object oriented design, a case study of Bingham University Library Management System is used. Software is developed to automate the Bingham University manual Library. The system will be stand alone and will be designed with the following functionalities:

Issues of Books

A student of any course should be able to get a book issued.

Books from general section are issued to all but departmental books are strictly issued to students in the departments.

A limitation is imposed on the number of books a student can borrow.

A maximum of 2 books from departmental and 3 from general section can be borrowed by a student for 15 days.

Software takes in the current system date as the date of issue and calculates the date of return. A bar code detector is used to save the student as well as the book information.

The due date for return of the book is stamped on the book.

Returning of Books borrowed by Students.

Any person can return a borrowed book. The student information is displayed using the bar code detector. The system displays the student's details on whose name the book(s) were issued as well as the issued and return date. The system operator verifies the duration for the borrowed book. The information is saved and the database updated.

Query Process

The system is able to provide the following information.

- Availability of a particular book
- Availability of a book of any particular author
- Number of copies available of a particular book.

- Generate report of available books in the library

Use Cases

Use case diagrams is a simplest representation of a user's interaction with the system and also depict the different types of users of a system and the various ways that they interact with the system. From the problem definitions, we can identify four majors' actors (Librarian, Operator, Bar Code Reader (BCR) and User) within the system. Their use case is given in Figure 1.

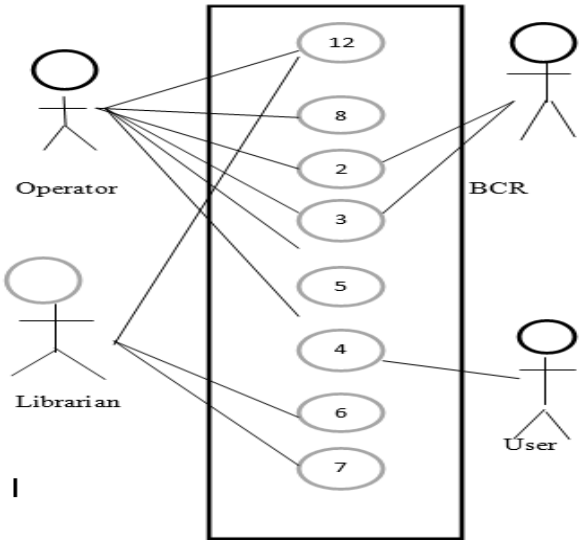

Figure 1: Use case diagram for Bingham University Library Management System.

Objects name in the use case are, 1: Login, 2: Issue Book, 3: Return Book, 4: Query Book, 5: Maintain catalogue 6: Generate report 7: Maintain Login 8: Maintain Student details.

Use Case Description

The description of the use cases are given as follows:

Login

These use case introduces the procedure for logging in to the University Library management System based on user privilege.

Here, the operator takes care of the following: issue books, return books, Query books, maintain catalogue, and maintain student details.

Also the Librarian maintain login and generate report.

Actors

Operator and Librarian

Pre-Condition

None

Post-Condition

If use case is successful, the user logged into the system, otherwise the system state is unchanged.

Logic Flow

Basic Flow

This use case starts when the use wishes to log into the University Library Management System. The following action can be trigger.

- The system request the user to enter his/her user_id and password.
- The actor enter user_id and password
- The system validates the user_id and password and check for his/her privileges.
- If the user is an operator, the system present operators menu.
- Otherwise, if the user is Librarian, the system present Librarian menu.
- The use case end.

Alternate Flow

Invalid name/Password

If the system received an invalid user_id or password, an error massage is displayed and the use case ends.

Special Requirement

None

Related Use Case

None

2. Issue Book

This use case documents the procedure of issuing a book for following accounts.

- General (15 days)
- Departmental Book (for the semester)

Actors

Operator, Barcode Reader

Pre-Condition

Operator must be logged in to the system

Post-Condition

If use case is successful, the book is issued to the student in his/her general or departmental account, otherwise the system state is unchanged.

Logic Flow

The use case starts when a student wants to get a book issued.

- The system reads and validates the student's information using the Bar Code reader.
- The System reads the book information using the Bar Code Reader.
- The return date of the book is calculated as per the account in which the student wishes to get the book issued.
- The book and student information is saved in the database.
- The issued details are sent to the printer to generate the receipt.
- The use case ends.

Alternate Flow

Unauthorized student

If the system does not validate the student, an error message is flagged and the use case ends.

Book Accounts if Full

If the student has already borrowed the required number of books, the request for issue is denied and the use case ends.

Course Mismatch

If a student requests book from a departmental library that is not his/her department, the request is denied unless under special cases and the use case ends.

Special Requirement

None

Related Use Case

Generate Barcode

Return Book

This use case documents the procedure of returning a book and calculating the fine amount if the student has returned the book after the specified return date.

Actors

Pre-Condition

The operator must be logged into the system.

Post-condition

If use case is successful, the book is returned to the library and if needed, a fine is calculated, otherwise the system state is unchanged.

Logic Flow

Basic Flow

This use case starts when a student wants to return a book.

- The system reads the book's information using Bar Code Reader.
- The book is return to the library
- The student and book database is updated
- The returned details are sent to the printer to generate receipt
- The use case ends.

Alternate Flow
Late return of book
If the book is returned after due date, fine is calculated and database is updated accordingly. The use case ends.

Special Requirement
None

Related Use Case
Generate Barcode

Query Book
This use case documents the procedure for searching a book based on the specified criteria, which are:

- Search by Author Name
- Search by Title Name

Actors
Operator, user

Precondition
Operator user must be logged into the system.

Post-Condition
If use case is successful, the book details are displayed.

Logic Flow
Basic Flow
This use case starts when a student wants to search for a particular book.

- The system displays the various search criteria to the user.
- The user selects the search criteria
- The result is displayed to the user
- The use case ends.

Special Requirement
None

Related Use Case
None

Maintain Catalog

This use case documents the procedure for updating the catalog of the library.

Actors
Operator

Pre-Condition
Operator must log into the system

Post-Condition
If use case is successful, the book should be updated, otherwise the system state is unchanged.

Logic flow
Basic Flow
This use case starts when the operator wishes to add, delete or modify some details in the library.

- The corresponding changes are saved in the database
- The use case ends

Alternate Flow
None

Related Use case
None

General Reports
This use case documents the procedures for generating reports as desired by the Liberian

Actors
Librarian

Pre-Condition
Librarian must be logged into the system

Post-Condition
If use case is successful, the various reports, regarding the details of books available in the library at any given time are generated.

Logic Flow
Basic Flow

This use case starts when a Librarian wants to generate reports of the books available in the library.

- The system displayed the various reports generating criteria to the user, which can be the books issued to students at a particular time, books available in the library etc.
- The Librarian selects the criteria and enters the various parameters based on the criteria selected.
- The system generates the report and sends it to printer for printing.
- The use case ends.

Alternate Flow
Printer out of Paper or low ink
If the printer goes out of paper or low ink, then the printing operation is aborted and the necessary action needs to be taken, which can be feeding paper to the printer or replacing think cartridge. The use case ends.

Special Requirement
None

Related Use Case
None

Maintain Login
This use case documents the procedure for maintaining log in details

Actors
Librarian

Pre-Condition
Librarian must be logged into the system.

Post-Condition
If use case is successful, the login details should be updated, otherwise the system state is unchanged
.
Logic Flow
Basic Flow

.

This use case starts when the Liberian wishes to add, delete or modify some details of login.

- The corresponding changes will be done
- The use case ends.

Alternate Flow
None

Special Requirement
None

Maintain Student Details
This use case documents the procedure for maintaining student's details.

Actors
Operator
Pre-Condition
Operator must log in the system
Post-Condition
If use case is successful, the student details should be updated, otherwise the system state is unchanged.

Logic Flow
Basic Flow
This use case starts when the operator wishes to add, delete or modify some details of students.

- The corresponding changes will be done
- The use case ends.

Alternate Flow
None
Special Requirement
None
Related Use Case
None
Class Diagram of the Entity Class
From the use case description, we can have a class diagram as the one given in figure 2 .These entity classes generally becomes tables in the database.

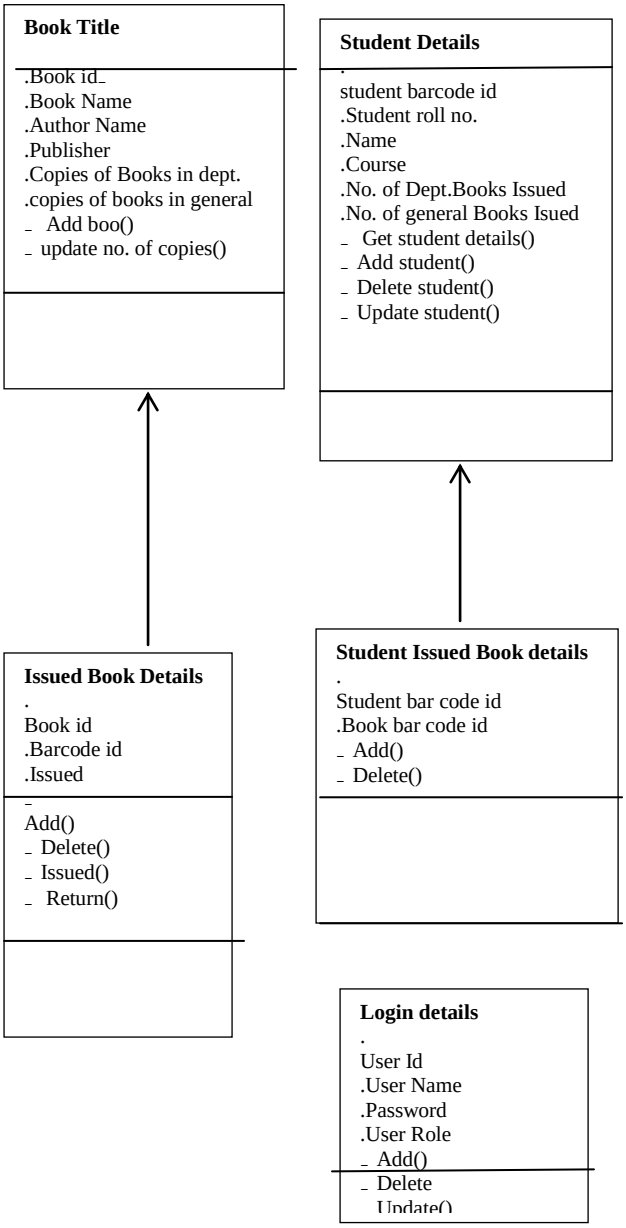

Figure 2: Class diagram for Bingham University Library Information System.

Summary

In summary, a well-prepared use case is worth its effort. Once the client has agreed to the use cases, the project managers can plan their schedule, and the developers can then have a clear understanding of what they must develop. From personal experience, we make bold to state that, while a use case is very beneficial, they are rarely perfect, but they always serve a useful purpose.

References

[1] Peter H (2004), Use case-based Software development. Retrieved from, http://haumer.net [Accessed on 18/04/20012].

[2] Larry L.C, Lucy A.D. (2002), Structure and Styles for Use Cases for User Interface Design. Constantine and Lockwood Ltd.

[3] Jarson G. (2006), Retrieved from, www.parlezuml.com [Accessed 18/04/2014].

[4] Dennise. E. (2005), Use Case: Background, Best Practices and Benefits. MKS White Paper in Application Engineering.

[5] Aggarwal.K.K, Yogesh S. (2008), Software Engeneering. Third Edition. New Age International (P) Limited Publoishers, New Delhi.