TOME Application Model

The TOME Team

October 4, 2006

Contents

List of Figures

1 Introduction

1.1 Purpose of Document

The purpose of this document is to specify the TOME system and the application model it uses.

1.2 Background

In December of 2003, several students on Dorm 41 started a system called TOME. The basic idea is that at the end of the semester, instead of everyone selling their books back to the bookstore, they all donate them to central repository. Anyone on the floor can then check out whatever books they need free of charge for a semester.

The advantage of having a computer-based system to keep track of all those books is easy to see, and one has been under development ever since the start of TOME. Since its humble beginnings as a quick solution over Christmas break, the system has grown to well over 3,000 lines of Perl code as well as HTML templates, a well-planned database schema, and significant documentation. The system not only has comprehensive facilities for tracking books and patrons, but also keeps tabs on what books are used for what classes and other alternatives to purchasing new books.

1.3 References

All project data will be stored in a combination Subversion repository and Trac environment. All of this will be made viewable at the following URL:

`http://enosh.letnet.net/trac/tome`

2 Application Overview

2.1 Scope

The TOME system is responsible for managing:

- Books
- Book information

- Class information
- Class-to-book mappings
- Book reservations
- Book checkouts

The TOME system is **not** responsible for managing:

- Book acquisition
- Book disposal
- Selling old books
- Any financial activity

2.2 Context

TOME exists as an independent system with no ties to other databases. Information from other databases, such as class listings, book information, and book-to-class associations will be used, but not in an automated fashion.

2.3 Technical Environment

TOME is a web-based Perl application. It is intended to be run under the Apache webserver, but any webserver that supports CGI should be capable of running the system. Template::Toolkit is used to process HTML templates. PostgreSQL is used as the database backend. CGI::Application is used as the framework for the system as a whole.

3 Actors

3.1 Actor Diagram

Figure 1 shows an overview of the actors involved in the TOME system and how they interact with each other. Patrons talk to TOMEkeepers who use the TOME web interface which is based on the TOME database.

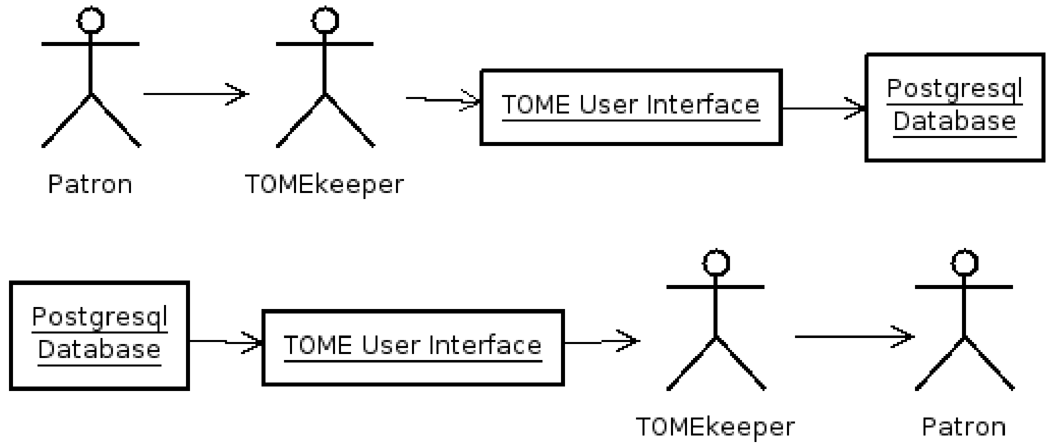

Figure 1: Actor Diagram

3.2 Actor Definitions

3.2.1 Patron

Description	The Patron is any student who wishes to use the TOME system. Only students that reside on floors with an active TOME system can become Patrons. Patrons never interact directly with TOME, only through TOMEkeepers.
Aliases	Student.
Inherits	None.
Actor Type	Passive - Person.

3.2.2 TOMEkeeper

Description	The TOMEkeeper is the primary user of the TOME system. They are responsible for all interactions with Patrons, all system administration, and all TOME activity.
Aliases	None.
Inherits	None.
Actor Type	Active - Person.

4 Business Use Cases

4.1 Use Case Listing

ID	Use Case Name	Comments
UC01	Book Added	testing 1 2 3

4.2 Graphical Use Case Diagram

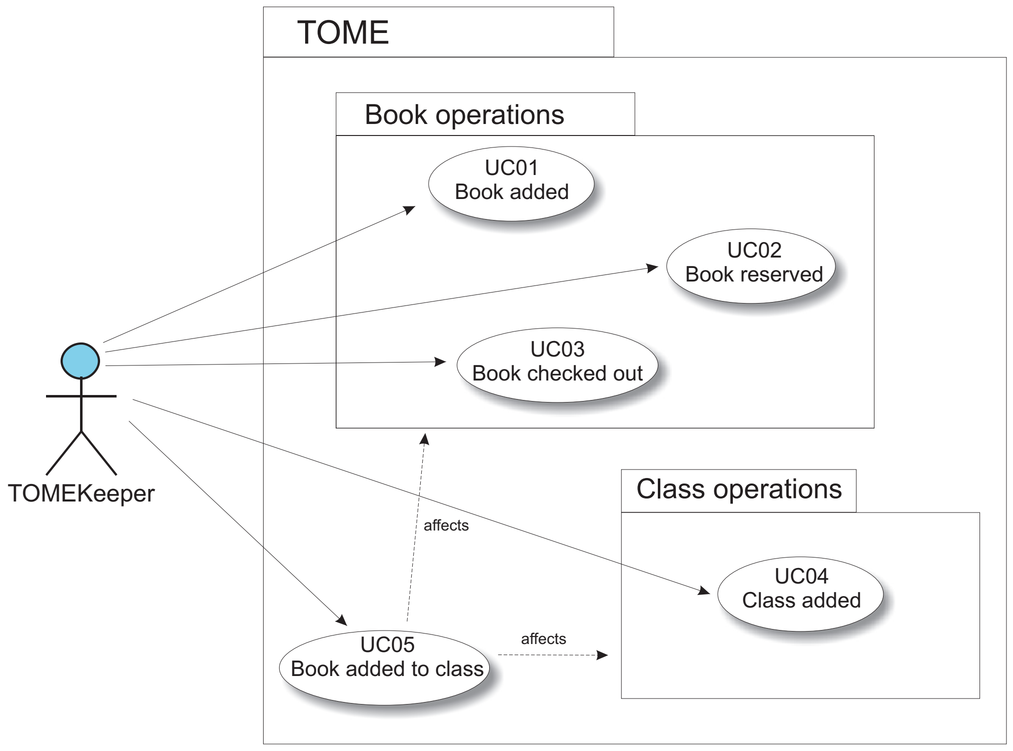

Figure 2: Use Case Diagram

4.3 Business Use Cases

This section documents the complete business scenarios within the scope of this project.

4.3.1 UC01: Book Added

Description test

5 Database Structure

The overall structure of the database can be seen in Figure 3. In addition to the structure, there are also triggers that ensure the consistency of the database at all times.

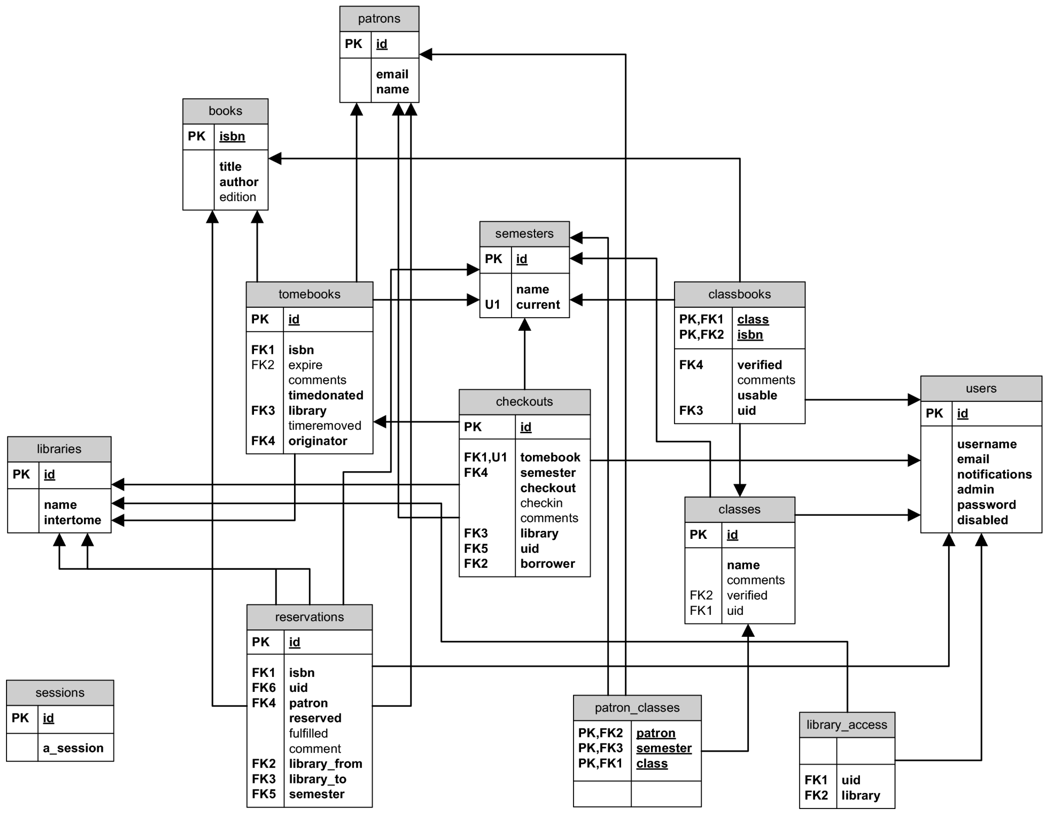

Figure 3: Database Diagram