A1

CS230 Group 47

**Tawe-Lib Design Document**

**Contents**

1. Introduction
   1. Purpose
   2. Scope
   3. Design Overview
      1. Description of problem
      2. Architecture
2. Candidate Classes and Responsibilities
   1. Resources
   2. Book
   3. Laptop
   4. DVD
   5. Copies
   6. User
   7. Librarian
   8. Account
3. UML Class Diagrams
4. Database Design
5. UI Mock-ups

**Section 1 –** Introduction

* 1. **Purpose**

The purpose of this document is to describe the implementation of the Tawe-Lib Specification given to us in assignment one of CS230. The Tawe-Lib software is a library management system.

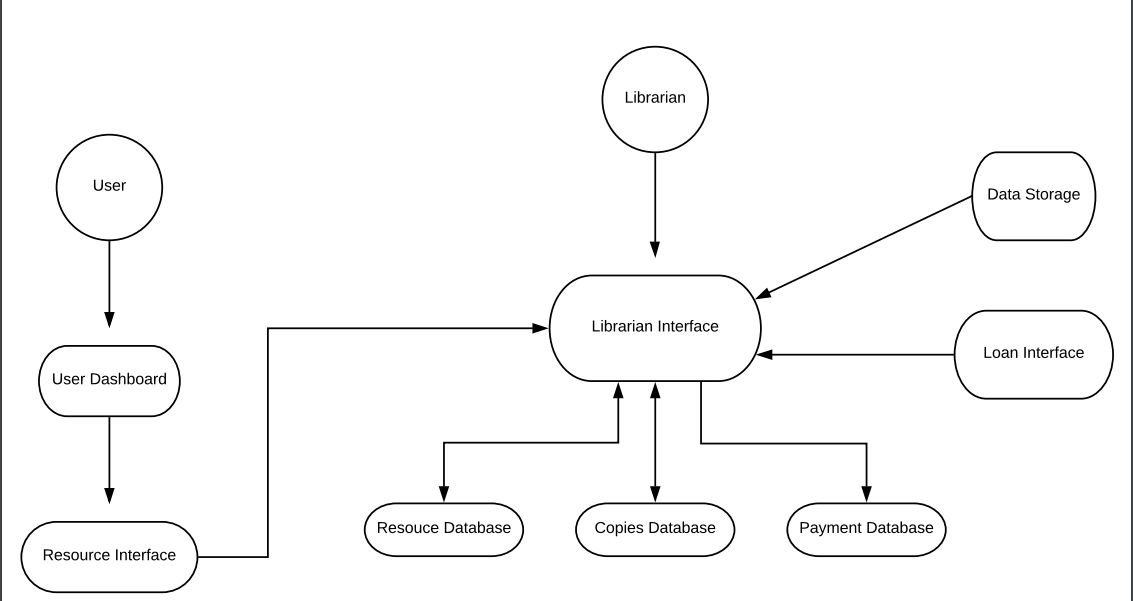
* 1. **Scope**

The Tawe-Lib software is a rudimental library system and is designed to keep track of all library resources, the check-in and check-out of these resources, fines for overdue loans, and multiple copies. All managed through a simple GUI with two distinct types of user.

* 1. **Design Overview**
     1. **Description of problem**

Users of the library must be able to browse all available library resources, get information on a resource and be able to check-out the resource if a copy of it is available. Or be Queued until one is available. Librarians must be able to add or remove resources or copies from the library and facilitate the payment of fines alongside the check-in and check-out of resources by users.

* + 1. **Architecture**

****

**Section 2** - Candidate Classes and Responsibilities

**2.1**

|  |  |
| --- | --- |
| **Resources** | |
| **Author:** Ryan Williams  **Super Class:** None  **Sub Classes:** DVD, Book, Laptop | |
| **Responsibilities**   * Provide a basic framework for the creation of other objects. * Unique resource ID. * Thumbnail image | **Collaborations**   * DVD * Book * Laptop * Copies |
| **Rough Description:** Abstract class. Contains the attributes held by all resources. | |

**2.2**

|  |  |
| --- | --- |
| **Book** | |
| **Author**: Ben Farrington  **Super Class:** Resources **Sub Classes:** None | |
| **Responsibilities**   * Get author string. * Get publisher string. * Get ISBN number. * Get genre string. * Get language string. | **Collaborations**   * Resource * Copies |
| **Rough Description**: A class to hold data on specifically the resource type of Book. A Book is a kind of resource. | |

**2.3**

|  |  |
| --- | --- |
| **Laptop** | |
| **Author:** Cristi Neacsu  **Super Class:** Resources  **Sub Classes:** None | |
| **Responsibilities**   * Get ID string. * Get Title/Name string. * Get Year number. * Get Manufacturer string. * Get Model string. * Get OS string. | **Collaborations**   * Resources * Copies |
| **Rough Description:** A child class of Resources, to store information about laptops available to rent in library. A Laptop is a kind of resource. | |

**2.4**

|  |  |
| --- | --- |
| **DVD** | |
| **Author:** Kieran Hughes  **Super Class:** Resources  **Sub Classes:** None | |
| **Responsibilities**   * Get Director. * Get Runtime. * Get Language. * Get Subtitle Language. | **Collaborations**   * Resources * Copies |
| **Rough Description:** Child class of Resources, stores attributes of DVDs available to rent from the library. A DVD is a kind of resource. | |

**2.5**

|  |  |
| --- | --- |
| **Copies** | |
| **Author:** Alex Moras  **Super Class:** None  **Sub Classes:** None | |
| **Responsibilities**   * Contains:   + Unique copy-ID (PK)   + Resource ID (FK)   + Issued to (User FK)   + Issue date   + Issued by (Librarian FK)   + Loan duration   + Expected return date   + Actual return date * Return date isn’t set until someone else requests the item and one isn’t available. * Each copy is a unique instance referencing a resource FK. * Each copy references the user (FK) borrowing it. | **Collaborations**   * User * Book * Laptop * DVD |
| **Rough Description:** This class is standalone and doesn’t have any super/sub classes.  Instead, it references other objects using a foreign key. A librarian can issue a copy to a User which requires them to specify which resource (i.e. type of book/DVD/laptop) through a foreign key for that object. On issue, return date isn’t set. If someone requests an unavailable item, the oldest “on loan” item is marked with a due date that is no-shorter than the loan-duration. | |

**2.6**

|  |  |
| --- | --- |
| **Users** | |
| **Author:** Christian Onishile  **Super Class:** Account  **Sub Classes:** None | |
| **Responsibilities**   * Username to identify instances of account * First name of instance of account * Last name of instance of account * Phone number on instance of account * Address of instance of account * Profile image of instance of account | **Collaborations**   * Accounts * Copy |
| **Rough Description:** This class models the characteristics of a valid user’s account within the Tawe-Lib. A user is a kind of account. | |

**2.7**

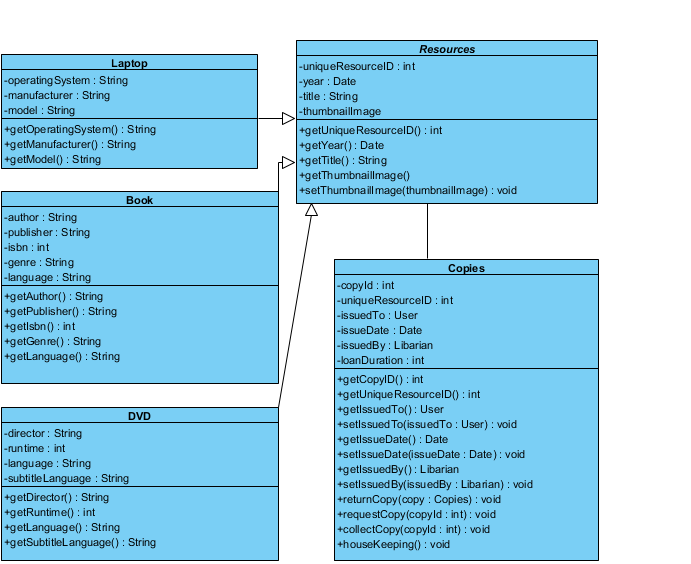
|  |  |
| --- | --- |
| **Librarian** | |
| **Author:** Christian Onishile  **Super Class:** Account  **Sub Classes:** None | |
| **Responsibilities**   * Username to identify instances of account * First name of instance of account * Last name of instance of account * Phone number on instance of account * Address of instance of account * Profile image of instance of account * Employment date of instance of account * Staff number of instance of account | **Collaborations**   * Accounts * Copy |
| **Rough Description:** This class models the characteristics of a valid librarian’s account within the Tawe-Lib. A librarian is a kind of account. | |

**2.8**

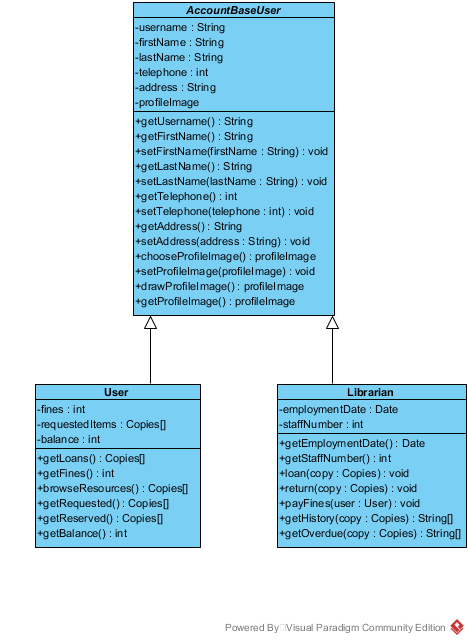
|  |  |
| --- | --- |
| **Account** | |
| **Author:** Christian Onishile  **Super Class:** None  **Sub Classes:** Librarian, User | |
| **Responsibilities**   * Username to identify instances of account * First name of instance of account * Last name of instance of account * Phone number on instance of account * Address of instance of account * Profile image of instance of account | **Collaborations** |
| **Rough Description:** An abstract class. With two sub-types of User, a standard User and a Librarian. | |

**Section 3** – UML Class Diagrams

**3.1**



**3.2**

****