

CSCI 5448
Object Oriented Analysis and Design
Semester Project - Final Report

Nikhil Jain(nija5462),
Harshit Hajela(haha4350),
Anusha Gupta(angu5645)

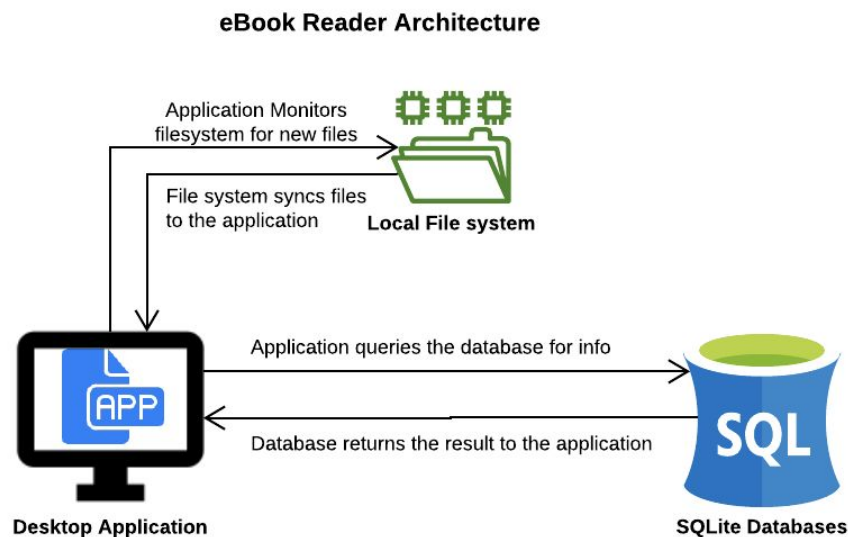
Table of Contents:

- Final State of System
 - Features implemented
- Initial class diagram of the system
- Final UML diagram with classes and key relationships
- Class diagram comparison with project 4
- Patterns used
- Third party code vs original code statement
- Statement on the OOAD process for overall semester project
 - Key design process elements or
 - Issues (positive or negative) experienced

Final state of the system:

Project Summary and Architecture:

We built an eBook Reader which will help users to read books digitally and to maintain a personal library. The eBook Reader has some cool functionalities like highlighting of text, Night mode for reading, finding word meanings on the fly and much more!

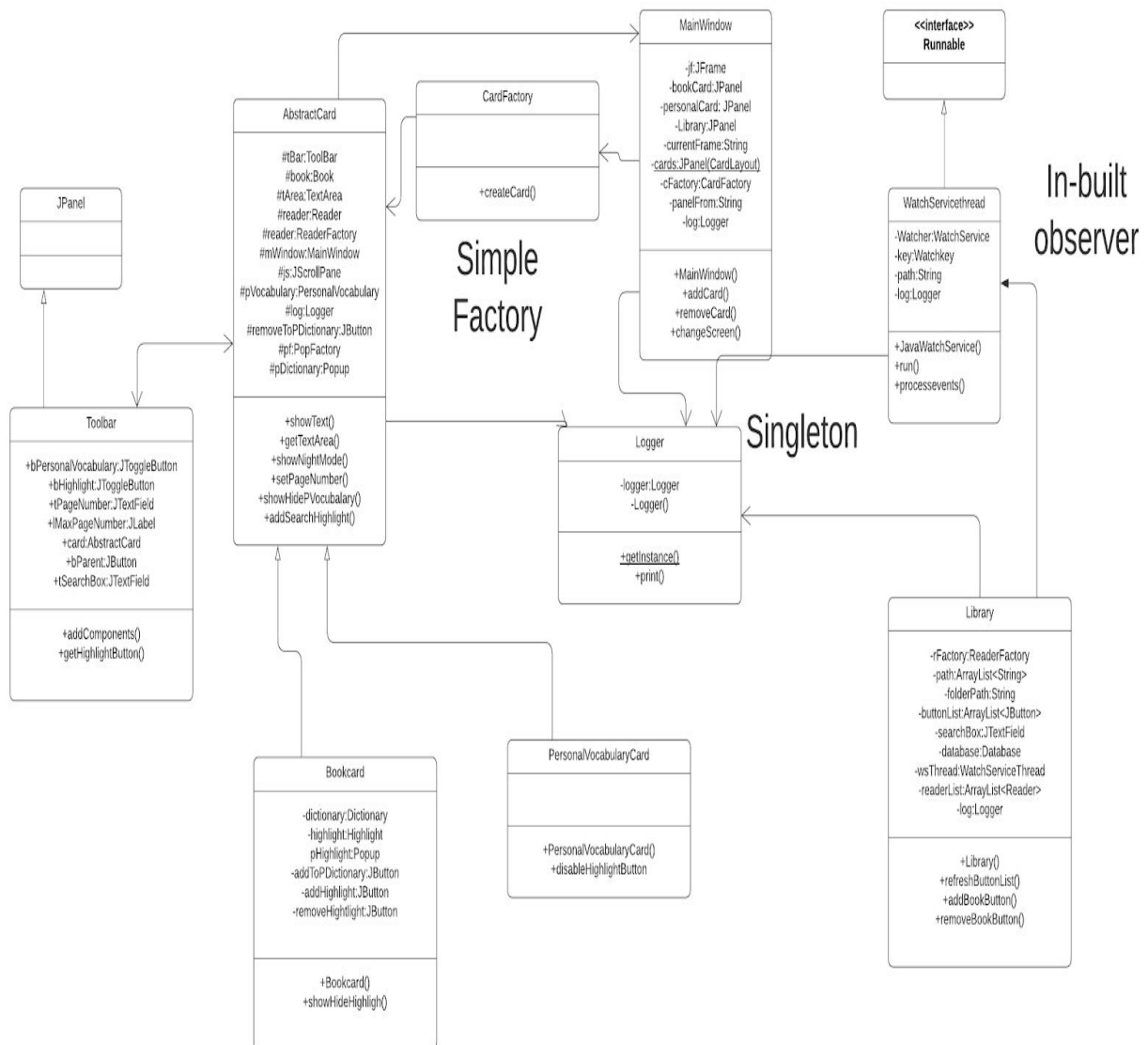


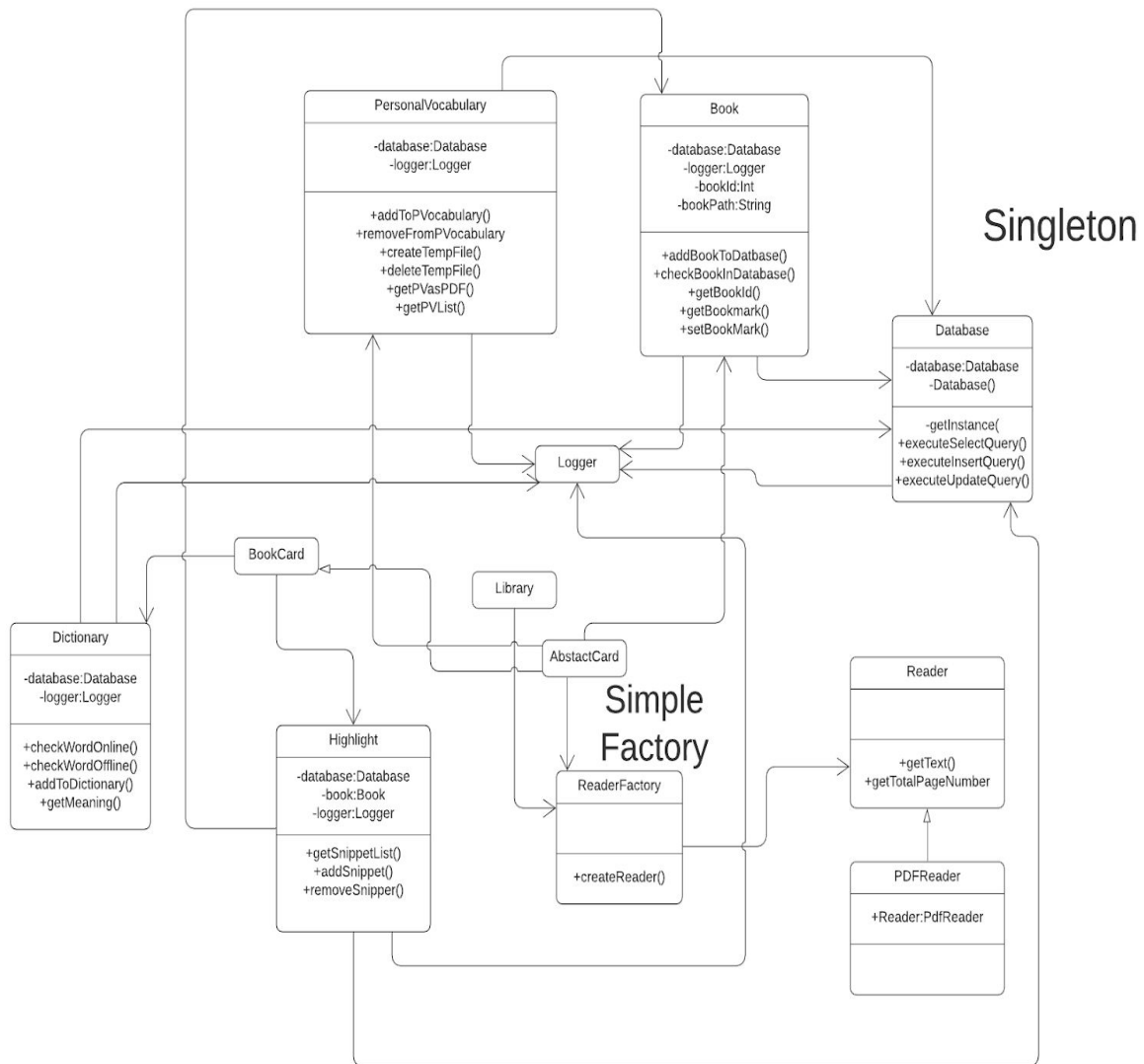
Features Implemented:

The eBook reader provides the functionality of:

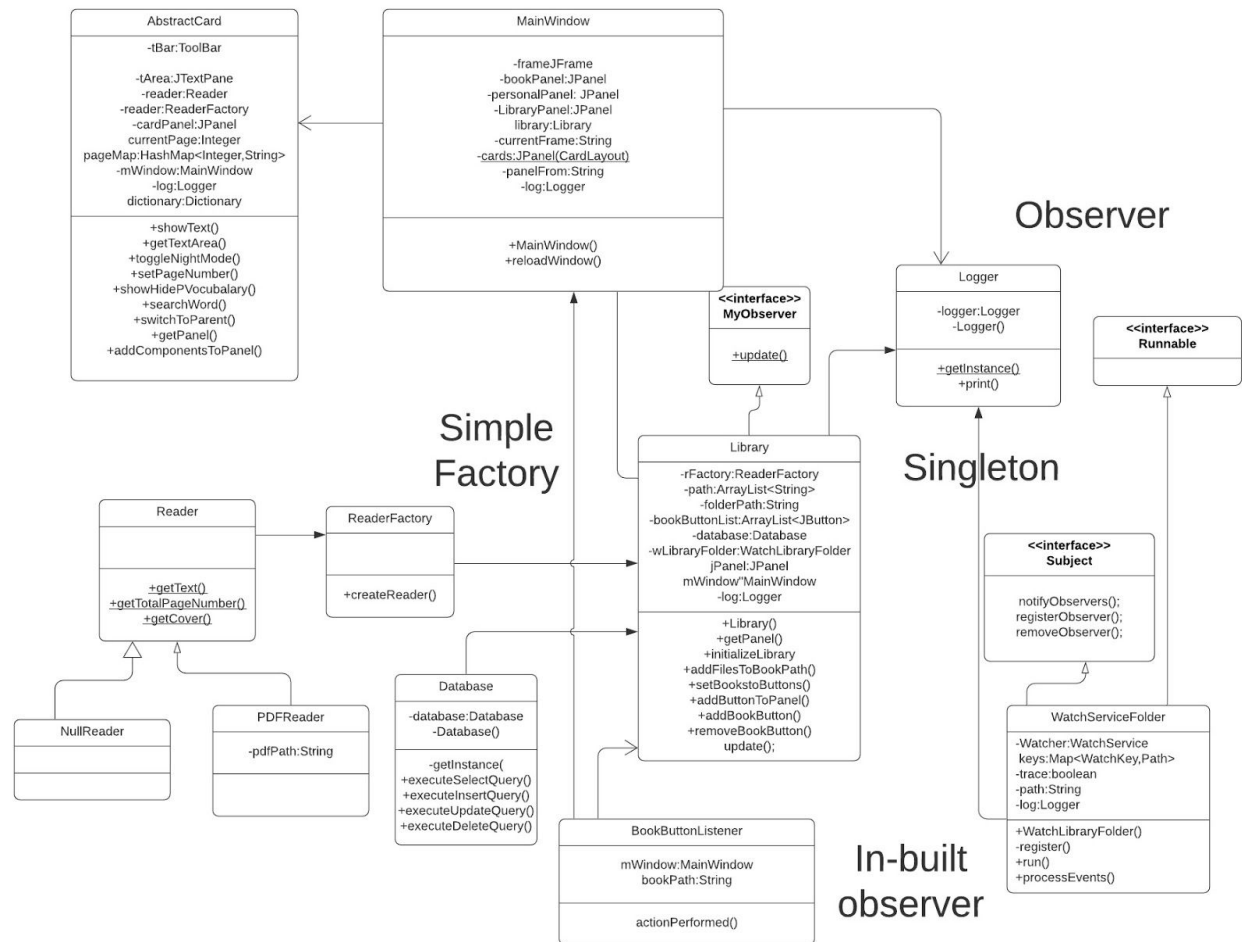
- Jumping to a page by providing the page number.
- Searching for a word.
- The user may switch to the Night mode.
- Bookmarking a page to continue where one left off.
- Look up word meanings on the fly, even in offline mode.
- Create a personal vocabulary archive of saved word meanings.
- Saving snippets of text from a book and viewing it later.

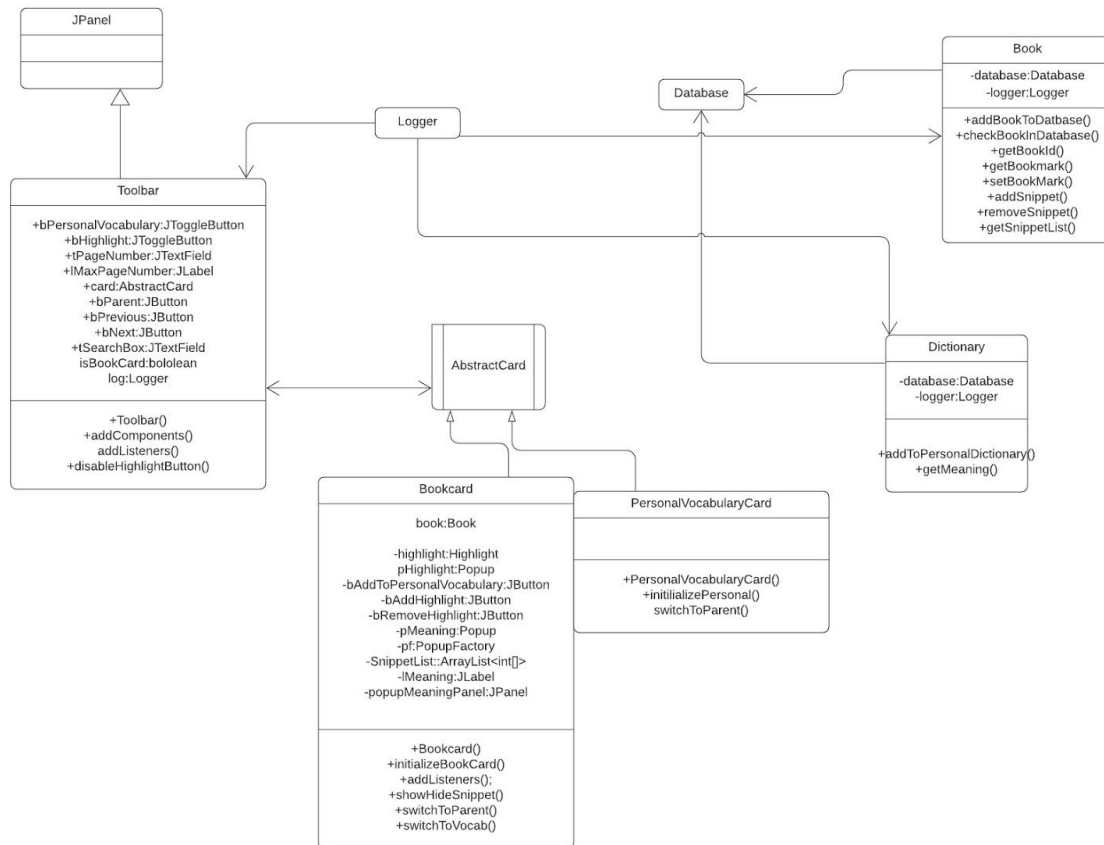
UML Class Diagram (Project 4):





Final UML Class diagram:





Changes in UML, from Project 4 - Final stage

The major change in the project was in the class pdfReader. Before, we were extracting the text from all the pages of pdf and storing it in a single string. Later, we realize this may not be a good idea for the text formatting. In the final submission, we used HashMap which maps page number to the text so that we can maintain the page order.

Another change was in book class, which was representing the book table as well as book tuple. While implementing, we realize that this is not a good design. So, we changed the class design to represent the book table as whole.

There are some additional components which we have added into Panel which was not part of Project 4 UML.

Patterns used:

Simple Factory Pattern - This pattern is being used in the application to create a reader object. The ReaderFactory creates the reader object.

Singleton Pattern- This pattern is being used for connection with database and logging. We are using a singleton pattern here since multiple instances of these classes could cause issues and are not required.

Observer Pattern- JavaWatchService is Java's in-built observer pattern to monitor the update on the File System. It watches a directory along with all sub-directories and files inside it. This API enables us to register a directory (or directories) with the watch service. When registering, we tell the service which types of events we are interested in: file creation, file deletion, or file modification. When the service detects an event of interest, it is forwarded to the registered process. Also, we have used observer pattern as discussed in Design Change.

Third-Party code vs. Original code Statement

We used the following third party code to build our project. Apart from these all code has been written by us :-

1. Java AWT - Java Abstract Windows Toolkit which provides a set of classes to build a basic GUI application with support for windows, graphics etc. we use this to implement events that we can attach handlers to.
2. Java Swing - Swing is a GUI widget toolkit that provides a more sophisticated set of GUI components and is designed to be platform independent. We use Java Swing for most of our GUI elements and controls.
3. iTextPdf - iTextPdf is a pdf engine that allows text extraction from and conversion to PDFs with multiple languages supported. We use it for extraction of all pdf text content
4. pdfbox - pdfbox is a java library that allows extraction of text from and conversion to pdf documents. We use it to extract the cover page of the pdf to be displayed in the library pane of our application.
5. Java nio library for keeping track of changes to monitored folders for addition of new books so we can update our library.
6. All the dependent binaries are included in the github repo

Project URL: <https://github.com/nikhiljain217/eBookReader>

Statement on the OOAD process for overall Semester Project:

Key design process elements :

- Understanding how the listener works in Swing and how we can arrange the class interaction so that a component can do its work and keep it as decoupled from other classes as possible. Sharing minimum information between the classes helped us to keep the code more flexible for the change
- Designing the Mockup GUI gave a better picture of the requirement.
- Using the requirement to define responsibility.
- Writing as minimum code as possible to understand the framework while designing .

Issues

1. Pdf Formatting- We realized very late that we could use icepdf instead of itextpdf for extracting text. Icepdf maintains the formatting of the text better than the itextpdf library. Doing a thorough research of third-party packages or libraries during the design phase is necessary.
2. Word Meaning - Initial idea was to maintain the database for the words and their meanings. Later, we realize having online sources for new words would be a good addition. Revaluating the requirement during the design as well as during the implementation does improve the project.