AL Local Library Management

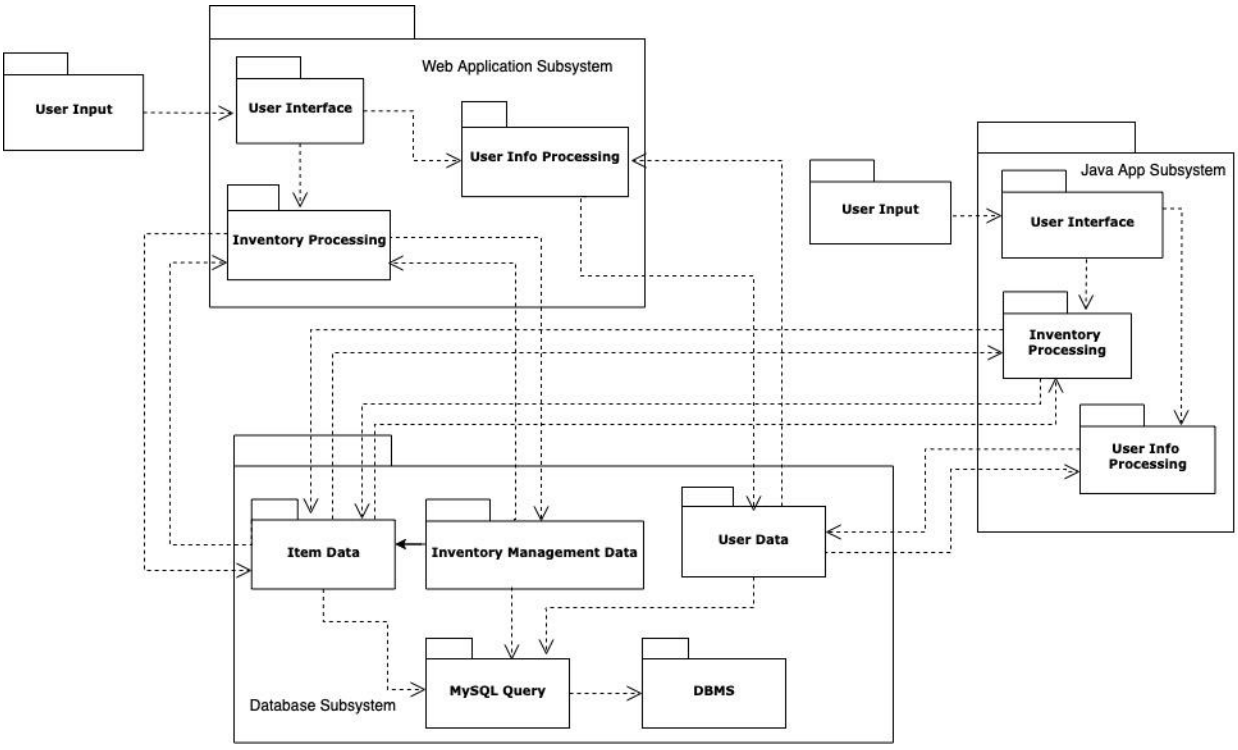
System Architecture and System Design

Architectural Styles

This system is client-server, where for both the Java application and the web application, the client is responsible for user input and interface, but the server-side programming accesses the data and processes the results. The system is two-tiered, involving both the client and server without additional middleware. My system uses a XAMPP server to process data from php programming and using MySQL to manipulate the database, and the results are returned to either the Java application or the web application on the client-side. Because I am using a local XAMPP server, the system is tested and operational on a single machine, but the intent is for the system to be able to run on multiple machines by instead using a non-local server address as this capability would be more practical for business purposes.

Identifying Subsystems

In the package diagram below, the three subsystems are shown, the web application, the Java application, and the relational database.



Mapping Subsystems to Machines

These systems can all be run on the same machine, and are for testing purposes, but generally the web application would be hosted on the Internet for access from basically any device, while the database and Java application would be run from computers (either a single machine or multiple) at the library location.

Persistent Data Storage

The system does require data storage to outlive a single execution of either application. Persistent data storage will be managed by the relational database, the database subsystem in the diagram above. Any edits made in either the web application or the Java application to database records will be reflected in the database instantaneously.

Network Protocol

The web application uses HTTP network protocols, while the Java application uses JDBC protocols to communicate with the server and access the database.

Global Control Flow

Execution orders: The system is event-driven, as both the web application and the Java application react to user clicks and input to generate options in various different orders. From the web application, the user can click through and perform various tasks and access different webpages by clicking on different navigation buttons, and in the Java application, the user selects options from the menu to begin tasks but can branch off to different tasks from certain windows as well, so the experience is more dependent on user events.

Time dependency: The system processes data in real time. This is important to make sure waitlist positions and availability is accurate when the information is requested, returns are processed immediately so subsequent checkouts can occur immediately, and so inventory items are easy to find without waiting for periodic updates. This is especially important when the system uses multiple machines.

Concurrency: The system does not use multiple threads.

Hardware Requirements

The system depends on screen display and communication with the server, which requires hard disk storage. The system can run either on a single machine or on multiple, in which case the server may not be local, but in my testing of the system, the XAMPP software is stored locally.

The most basic requirements to run this system would be a color display with a minimum 640 x 480 pixels, a desktop for the web application and either a desktop or mobile device

with Internet access for the web application, and at least 1 GB of RAM and 1GB of hard drive space, though these requirements may increase if the size of the database increases. The system uses roughly 30 kbps of bandwidth per user so this would be the minimum bandwidth required for the smallest scale of the system (single machine), but this requirement would need to be scaled up as the system expands.