Appendix I: Installation Guide

The "Wizard.jar" file, which consists of the research product of an object-oriented application framework for library systems domain is used to install it on a computer. On Windows file system, double click the Java executable jar file after the Java Development Kit 1.2.2 (JDK) or later has been installed. Alternatively, the Java Runtime Environment 1.2.2 (JRE) is also workable. On other file systems, such as UNIX, LINUX and related systems, it has to be invoked from the command prompt, such as

/home/roy/jdk1.2.2/bin/java -jar Wizard.jar

The file size of Wizard.jar is 640 KB. After installation of the application framework into a selected directory, there exist four files which consume approximately 673 KB.

1. a default application system, "JStartLib.jar",
2. the database data file, my_base
3. the database dictionary file, my_dict
4. the library system initialisation file, Status.ini

If the default library application system developed with POET 5.1 client database is to be reused, the database system has to be installed first. Although the relevant POET packages are included with the "Wizard.jar", there are some undocumented platform-dependent drivers and libraries necessary for utilising the database system.
If a user chooses to customise the framework, the total size of all relevant uncompiled Java source files and directory files representing the packages is 87 KB.
Appendix II : User Guide

To reuse the library systems domain application framework, a wizard tool exists to assist application developers in choosing and customising the components for developing the library application system. A step-by-step window interface will guide users in getting and setting the preferred requirements for the library application system.

![Application Wizard]

Figure 1: Step 1 of Wizard tool:

Choosing the Framework's Application Database.

Based on Figure 1, step 1 of the wizard tool in a library systems domain application framework is to assist users in choosing the database components for the library application system. If users want to have a complete library application system, then the default option will provide users with a database application implementation. Otherwise, the alternate option would leave out the database component and users have to adopt their own database component system.
Figure 2: Step 2 of Default Option in Step 1 Wizard Tool.

Based on Figure 2, step 2 of the wizard tool, the content of each package component of the library systems domain application framework is displayed. This window interface is displayed when users opted for the default option of database application. Step 2 will assist users in understanding the library systems domain package components for easier identification and reuse. The code of each class in a package component is displayed when selected individually. The white-box framework feature will assist users in reusing the code and customising it.
Based on Figure 5.5, when application developers opt to customise an application by using the framework components, a screen is provided for the selection of classes from each packaged component. Consequently, the wizard will act as an automated code generation tool and generate the relevant component and its selected classes. Only related classes and components can be enabled for selection based on association and message dependency.
Figure 4: Step 3 of Wizard tool:
Deployment of Package Components to Reuser's Preferred Directory.

Based on Figure 4, step 3 of the wizard tool is to assist users in deploying their selected components into packages at a chosen directory of their choice. Once this final step is done, the user will have to choose the "Finished" button at the bottom of the screen to complete the wizard guide.
available=available
borrowed=borrowed
onhold=onhold
allout=all borrowed
copyReturned=copy returned
reserved=reserved
MaxReserve=5
MaxLoan=5
dbName=poet://LOCAL/my_base
fineDayWeek=25
fineDayMonth=100
DueCondition1=7
DueCondition2=14
loanDays=14
holdDays=7
membership=staff, member, public
stafffee=1
memberfee=10
publicfee=20
itemType=book, magazine, cd, audio

Figure 5: Initialisation File for Configuration of Library Entity Status and Business Rules

Based on Figure 5, a user can customise the initialisation file for configuration of library entity status and business rules. This file is editable and deployed as part of the library systems domain application framework. For example, to change the value of the last key "itemType", the user can add or remove the type of items offered for loan by a library system. These variable features are meant to be flexible for changes to library requirements.
Based on Figure 6(a), the window dialog box is the initial user interface to access the library application system database. Figure 6(b) is a window dialog box to disconnect the database.
Figure 7: Insert a Title Dialog

Based on Figure 7, insertion of a Title object into the library system requires attributes of Title's name, author's name, ISBN no. and type of item. The item may be book, magazines, CD, video and others.

Figure 8: Insert a Borrower Dialog

Based on Figure 8, it is a dialog box for inserting a Borrower into the library system. The attributes expected of a Borrower object are name, address and type of
membership. The membership status available for selection is member, staff or public.

![Figure 9: Insert a Copy Dialog](image)

Based on Figure 9, insertion of a Copy to a Title can be made with the dialog box. Attributes expected are Title and item number which are unique.

![Figure 10: Insert a Loan Dialog](image)

Based on Figure 10, the attributes of a Loan object are the item's identification number and borrower's name.
Figure 11: Insert a Reservation Dialog

Based on Figure 11, the attributes expected of a Reservation object for association with other domain classes and inserting the object into a database are the title’s name and borrower’s name.

Figure 12(a): Delete a Borrower Dialog

Figure 12(b): Information Feedback of Deletion on Borrower
Based on Figure 12(a) and 12(b), if the Borrower has an outstanding Loan or Reservation, deletion is not permitted due to data relationship and referential integrity between the entity classes of Borrower and Loan or Reservation.

<table>
<thead>
<tr>
<th>Name: ANA TRUJILLO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address: MATADERSOS 2312</td>
</tr>
<tr>
<td>Membership: member</td>
</tr>
<tr>
<td>Loan: 0</td>
</tr>
<tr>
<td>Loan Details:</td>
</tr>
<tr>
<td>Reservation: 1</td>
</tr>
<tr>
<td>Reservation Details:</td>
</tr>
</tbody>
</table>

Figure 13: Retrieval of a Borrower. Information of Borrower as Displayed.

Based on Figure 13, the window reflects an information search on class Borrower. Based on a relevant search, we can retrieve one or more records that match.

Note: The Borrower details included are

1. name
2. address
3. membership
4. The number of loan & reservation by this Borrower.
5. The details of a reservation such as title, date and time of reservation (circled).

found 1 object.

Title: STRUCTURE AND INTERPRETATION OF COMPUTER PROGRAMS
Author: ABELSON, H.
ISBN: 0-262-01153-0
Type: book
State: available

Copy 1 available 0-262-01153-0-A

Figure 14: Retrieval of a Title Object.
Information of Title as Displayed.

Figure 14 shows the result of an information search on class Title. Based on a relevant search, we can retrieve one or more records that match.

Note: The Title details included are

1. title name,
2. author name
3. isbn no.
4. type of item
5. state of item
6. no. copy, status, item id. no. (circled)
Figure 15: Information on a Title Object with Reservation Status

Figure 15 shows the result of a search on a Title with its current status as reserved when all copies are on loaned and a Reservation is placed on it. The above window displays the information of a single copy which is borrowed on the particular date, time and borrower (circled).
Figure 16: Onhold status for Reservation.

Based on Figure 16, when we look up a Title and if a reservation is placed on a Title, the immediate return of the Title's copy will be rendered onhold to its status.

Note: The details of the above diagram include
1. title name,
2. author name
3. isbn no.
4. type of item
5. state of item
6. no. copy, status, item id. no. and borrower who placed the reservation(circled)
Figure 17: Fine on Borrower for Overdue Loan.

Based on Figure 17, upon the return of a Copy on Loan, the system will determine the last returning date. If it's overdue, a fine will be imposed on each day based on pre-configured library conditional parameters. For example, the system is set to a $0.25 fine for a day (25 for 1) or $0.50 fine for 2 days (50 for 2).