**Group Journal –** last update 2/27/2018 dls

**Meeting Date : 01-16-2018**

Members Present: Bhavani Putrevu and Doug Skayer

**GitHub link:** <https://github.com/bputrevu/ICS499_Project_Library/tree/master/DOCS>

**Meeting date #1:** 1/9/2018

**Members:** Doug & Bhavani

**Major discussions**: Basic design of Library system, some requirements, and some db tables.

**Major decisions**:

* The 3-tier development project we decided to do is a library checkout system.
* Bhavani and I outlined and submitted on 1/9/2018 to Ryan the entities we thought were relevant to our implementation.
* **Book** (title, author, ISBN, type, description), **location** (name, address, hours), **user** (name, address, library id), **book\_status** (hold, location, check-in/out).
* Other functions/entities/objects/attributes include: book\_search, due\_date, user\_login, account\_management, cancel\_hold, book\_inventory.
* We decided to exclude the library-employee aspect of the project, focusing more on book management.
* The platform suggested by Ryan was Amazon AWS. I’m not familiar with it, but if that is likely what others would use, I will look into that. It sounds like use of the CS server at metro may also be an option.
* While somewhat familiar with Java, I’m not sure how the 3-tier framework would work. The only practical way I can think of deploying this project is by using html/javascript, php and MySQL. We hadn’t talked about it, but perhaps Bhavani has better ideas for the implementation.

**Meeting date #2:** 1/16/2018

**Members:** Doug & Bhavani

**Major discussions**: How to divide work requirements for next week’s class.

**Major decisions**:

* I would setup GitHub repository and add Bhavani as a contributor.
* We would split the use cases.
* We would split the DB tables.
* We would split the class diagrams with methods.
* We would both contribute requirements.
* I would do a high-level over view of what the project is in the first 5 minutes of the presentation.

**Meeting date #3:** 1/23/2018 **– missed due to snow storm.**

**Meeting date #4:** 1/30/2018

**Members:** Doug & Bhavani

**Major discussions**: How to divide work requirements for next week’s class.

**Major decisions**:

Doug will:

* Create DOCS folder on github
* Build platform to work on: Java-Swing for the web tier, Java for the application and MySQL for the database, Linux for the OS.

Bhavani will:

* Work on application design

**Meeting date #5:** 2/13/2018

**Members:** Doug & Bhavani

**Major discussions**: How to divide work requirements for next week’s class.

**Major decisions**:

Doug will:

* Go to Java tutor to determine how to read data from MariaDB using a Java-Swing class.
* Will update the user guide
* Will update the status report
* Will update this journal.

Bhavani will:

* Requirements Document
* Design Document
* Test Data

**Meeting date #6:** 2/21/2018

**Members:** Doug & Bhavani

**Major discussions**: How to divide work requirements for next week’s class.

**Major decisions**:

Doug will:

* Continue working on MariaDB using a Java-Swing class.
* Will update the user guide
* Will update the status report
* Will update this journal.

Bhavani will:

* Requirements Document
* Design Document
* Test Data

**Meeting date #7:** 2/27/2018

**Members:** Doug & Bhavani

**Major discussions**: How to divide work requirements for next week’s class.

**Major decisions**: It turns out we both were independently working on Java Swing and different database implantations. We agreed to meet to determine how we would split the functions within the menu and work independently to try to get more accomplished.

As of this week Bhavani:

(1) Created GUI menus using Java SWING.

            This has two sub menus

                        (A) Admin Functions

                                    - Add User

                                    - Add Book

                        (B) Member Functions

                                    - Login

                                    - Search for a book

                                    - Check out

                                    - Return

                                    -Hold

(2) Created Java model objects for

            - Book

            - User

            - Hold

            - Library

(3) Created a PostgreSQL DB on my mac

            - Created DB

            - Created Tables

            - Uploaded the table scripts in GitHub

(4) Started coding on Java template to connect to DB and make an update to DB

**Meeting date #8:** 4/03/2018

**Members:** Doug & Bhavani

**Major discussions**:

**Major decisions**:

As of this week Bhavani:

(1) Created PostgreSQL DB scripts

(2) Added common DAO component for CRUD operations

(3) Added AddUser Menu. Connection between UI and backend is working.

- On submit button click, new user is added to DB

- On cancel button click, the current screen is closed

(4) Added AddBook Menu. Connection between UI and backend is working.

- On submit button click, new book is added to DB

- On cancel button click, the current screen is closed

(5) Added/Modified several java components.

(6) Updated Journal doc.

(7) Created PostgreSQL DB.

As of this week I:

* Updated the Group Journal doc
* Updated the Library Application Status doc
* Updated the Library System User Guide doc
* Create MariaDB tables and populated with data.
* Continued to work at getting a JavaSwing button to display a database table. I was partially successful, but there is still one bug with this test. I added POC code to <https://github.com/bputrevu/ICS499_Project_Library/tree/master/POC>

**Meeting date #8:** 5/01/2018

**Members:** Doug & Bhavani

**Major discussions**:

**Major decisions**: Share what was done during the project.

As of this date, Bhavani added the following:

1. Adding a new user to the User List in the Library
   1. <https://github.com/bputrevu/ICS499_Project_Library/blob/master/src/UI/AddUserScreen.java>
   2. Updated with outstanding penalty fees
2. Adding a new book to the Book List in the Library
   1. https://github.com/bputrevu/ICS499\_Project\_Library/blob/master/src/UI/AddBookScreen.java
3. User checks-out a book from the Library
   1. <https://github.com/bputrevu/ICS499_Project_Library/blob/master/src/UI/LoanBooks.java>
   2. Creates transaction
4. User returns a book to the Library
   1. <https://github.com/bputrevu/ICS499_Project_Library/blob/master/src/UI/ReturnBook.java>
   2. Creates transaction
   3. Calculates penalty, if any
5. User holds a book
   1. <https://github.com/bputrevu/ICS499_Project_Library/blob/master/src/UI/HoldBook.ja>
   2. Creates transaction
6. Removing a hold on a book
   1. <https://github.com/bputrevu/ICS499_Project_Library/blob/master/src/UI/RemoveHold.java>
   2. Creates a transaction
7. Renewing a book
   1. <https://github.com/bputrevu/ICS499_Project_Library/blob/master/src/UI/RenewBook.java>
   2. Creates transaction
   3. Calculates penalty, if any
8. Printing user’s transactions
   1. <https://github.com/bputrevu/ICS499_Project_Library/blob/master/src/UI/CreateTransaction.java>
   2. Transaction creation done when book is loaned, renewed or returned.

Modified:

Several models: <https://github.com/bputrevu/ICS499_Project_Library/tree/master/src/Models>

<https://github.com/bputrevu/ICS499_Project_Library/blob/master/src/DBScripts/PostgresDBScripts>

<https://github.com/bputrevu/ICS499_Project_Library/blob/master/src/DAO/PostgresDao.java>