**Library Management System**

**CS 157A - Team #22**

Michael Leonard

Nicholas Papano

Sarah Saber

# **Project Overview** Description

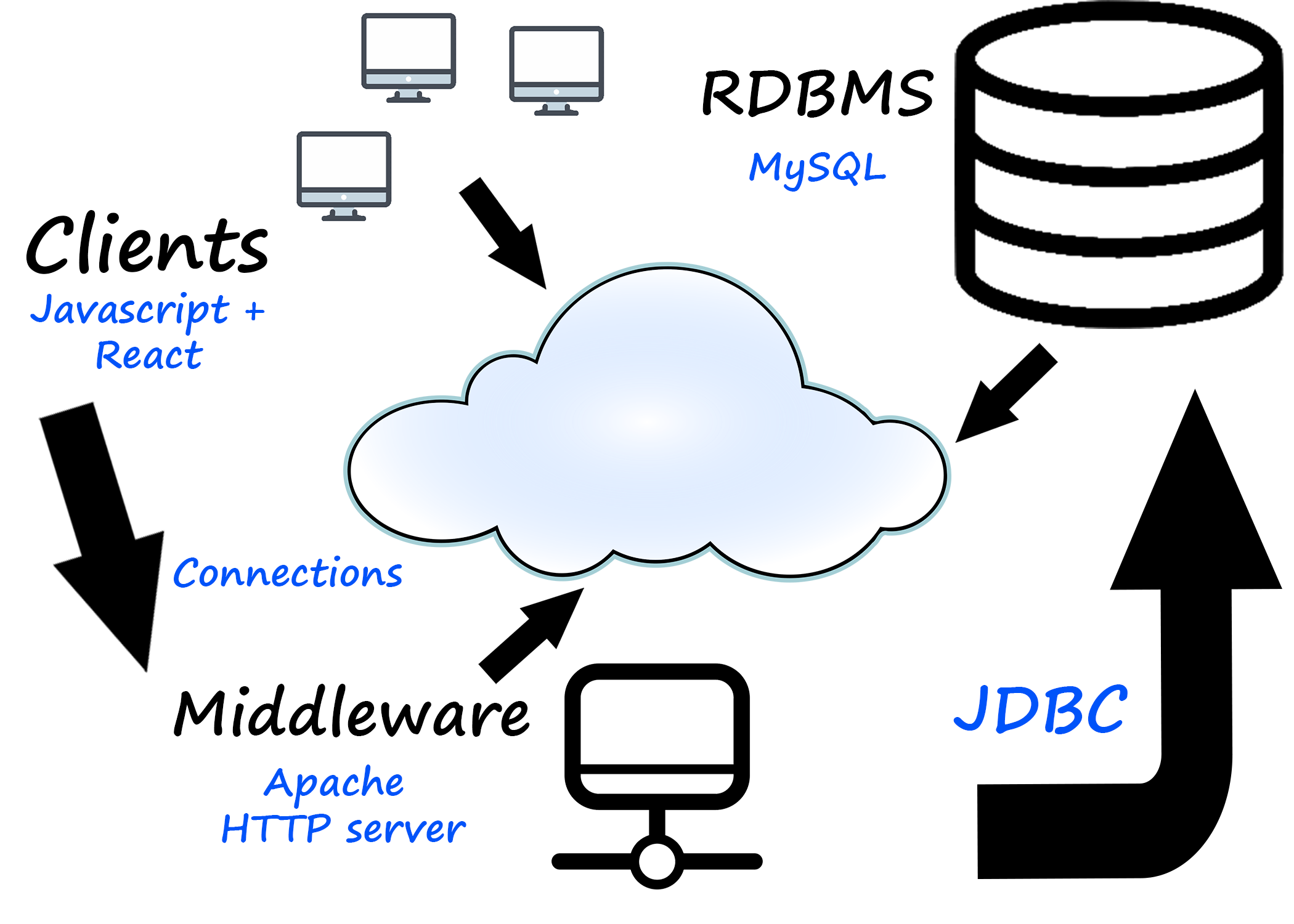
The three-tier database application we will be building is a library management system. The system will perform all fundamental features that a library will need in daily operation. These actions include operations such as issuing library cards, searching for books, checking books in and out, and placing holds. It will also include extra privileges for library staff, such as managing accounts, adding and removing books, and paying fines. The organization that oversees the library will be able to request general analytics and telemetry from the librarians. We as the developers will be designing and implementing the project, making sure all development goals are met. The system will be a web-accessible login-based application and maintain a user interface that is clean, clear, and informative so that all of the various users can navigate the application with ease.

Stakeholders

* Librarians
* Library Staff
* Library Users
* City Government
* Developers

**System Environment**

Structure of the system



Software used

● Apache HTTP Server

RDBMS

● MySQL

Application Languages

● Java, HTML, CSS, JavaScript, React.js. JDBC

# **Functional Requirements**

There are two groups of users for this system, the customers of the library and the staff at the library. Library customers will use the system to see specific information about the books they have borrowed, or to find new books to borrow. Library staff will use the system to manage library inventory and the accounts of the library customers. Both groups of users will access the system by logging in through a web page using their email address or library card number and a password.

Features of the system will be developed to support expected use cases for different user groups. Users will visit the home page and either log in using email address or library card number and a password. Based on the type of user logged in, different options will be made available on the following page.

Library customers will be presented with features allowing them to perform tasks such as searching for books/items in the library inventory, viewing their history of borrowed items, place a hold on a book, renew a borrowed item, check due dates for borrowed items, check late fees associated with their account and create a list of books they would like to borrow. Most of these features involve simple queries from the database. To place a hold on a book or add a book to a list to borrow in the future, information in the database will be modified.

Library staff will have additional features that are not accessible to customers. These features will support use cases such as the ability to create a new account for a customer, checkout or return books, add new books to the inventory, change the check-out duration of an item if it is in demand, remove holds and late fees from an account upon receipt of payment, and generate analytic reports of items that are high in demand or of how many books are late. Most of these additional features allow library staff to modify or add to information in the database.

# **Describe each individual function, functional process and I/O:**

**Functions:**

Check in item

* Library staff shall be able to change the status of an item in the database from “checked-out” to “available” after searching for it and clicking the “edit’ button next to the item on the results page.
* The system shall reflect those changes for all users to see.

Check out item

* Library users shall be able to check out an item from the database by clicking the “check-out” button next to an item.
* The system shall verify that the item is not already checked out or placed on hold. It shall also update the status of the item in the database and add the item to the checked-out list of the user with the date and time it was checked out.

Browse inventory

* Library staff/users shall be able to browse all items in inventory whether on hold, checked in, or checked out. They will be able to click on any item to view further details or edit information (only if they were staff members).
* The system shall display a list of all items in the database. It will also allow users to refine results by specifying certain parameters such as author, genre, or “available” for browsing.

Search for item

* Library staff/users shall be able to search the inventory for items by providing information about the item such as name, genre, or author in a search bar.
* The system shall display an item or a list of items that satisfy the search criteria retrieved from the database. If no item matches the search parameters, a message will be displayed to the user to inform them that no items were found. Functionality on items on the results page will differ depending on the type of account logged in (Staff vs User).

View item information

* Library staff/users shall be able to view item information by clicking on a button.
* The system shall display the item information in a new page by retrieving data from the database. The system will also allow users to add the item to their wish list. Functionality on the item information page will differ depending on the type of account logged in (Staff vs User)

Add/Remove item

* Library staff shall be able to add an item to the library inventory by adding its information (such as name, barcode, call number, checkout duration, …) to the database. Items can come from purchases, donations, or transfers from another library.
* Library staff shall be able to remove an item from the library inventory by clicking on the delete button after searching for that item and selecting it.
* The system shall allow staff members to add items by filling out a form. The system shall also update the database with information entered and reflect it on the web application.

Edit check-out duration of item

* Library staff shall be able to edit the check-out duration of an item by clicking the “edit” button next to the item on the search results page.
* The system shall update the database with those changes and reflect them on the application.

Add item to wishlist

* Library users shall be to click an “Add to wish list” button next to an item being viewed.
* The system shall add that item to an ordered list of items the user would like to check out in the future.

Renew an item

* Users of the system shall be able to click a “Renew” button when viewing item(s) that they currently have checked out.
* The system will update the due date for the item.

Place hold on item

* When viewing an item, if the item status is not “available” users shall be able to click a button to place a hold on the item.
* The system will add the user to an ordered list of users waiting to check out the item.
* The system will prevent users from checking out the item unless they are the next user on the hold list, or the list is empty.

Create account

* Library staff shall be able to create a new account for a user by entering the user information such as name, address, phone number and email address.
* The system shall assign a user ID number and a temporary password.
* The system shall update the database with the information for the new user.

View reading history

* When viewing their account, users shall be able to click a button to view reading history.
* The system shall display a list of all items the user has previously checked out.

View late fees

* While looking at account details, library users can clearly see any outstanding fines they need to pay
* System will display current fines and when clicked, will provide additional information regarding what incurred the fine

View due date of items

* While viewing their account, users can click to see their checkout history displayed as a list
* Currently checked out books will show their check-out date

Generate analytic report

* Library staff can generate a detailed report to see analytic data the system collects about the library
* The report will show for a given month or calendar year:
  + The number of check-ins and check-outs for books
  + The current amount of each book in stock and checked out, and missing
  + Total fines owed to the library

Login/Logout

* The user will be able to log in and out from any computer
* Users can then view any account information from there
* Users can reset their passwords using the provided email address

# **Non-Functional Requirements**

Performance

* Application response time for each use case should be efficient.
  + No transactions or queries with the exception of generating the analytic report should take no longer than 0.5 seconds. The analytic report feature is expected to be used infrequently and it is acceptable for the report generation to take a few seconds to complete.

GUI

* React.js will be used along with other UI frameworks to develop the GUI for the application.
  + After the initial login page, the user interface will include a sidebar with buttons to access different features depending on the type of user account logged in. The main portion of the user interface will be for displaying information or text entry fields. The information displayed could be the result of a search for a book, a particular user’s reading history or wish list, a list display of all books due on a particular day, or the results of one of the analytic reports. Text entry fields would be displayed to support features for library staff such as adding new items to inventory or adding new user accounts.

Security

* Handle login/logout for different types of users.
  + Users will login to the system using a combination of their password and either their email address or ID number. Users that are classified as library staff will have access to the full set of features in the application while users that are classified as library customers will have access only to a limited feature set.
* Securely store passwords in database.
  + Passwords will not be stored in plain text but will first be hashed and salted.
* Allow users to reset passwords securely if they forget their credentials
  + Users can request a password reset using their email address. A temporary password will be emailed to the user and they will be prompted to change their password once they have logged in to the system.