DEPARTMENT OF COMPUTER SCIENCE
COLLEGE OF ARTS AND SCIENCES
CSCI 4961/4962 Capstone Documentation

Title of Project:	Library Book Finder
Client:	Lee Cummings
Supervisor:	Dr. Michael Goldwasser
Student(s):	Lorna Xiao, Sara Jain

Description:
The Library Book Finder project allows users to locate a bookshelf containing their desired book through the Saint Louis University library catalogue site. It focuses on enabling user-friendly accessibility to that location in the library by sending the user a link to Google Maps that is available on all platforms.

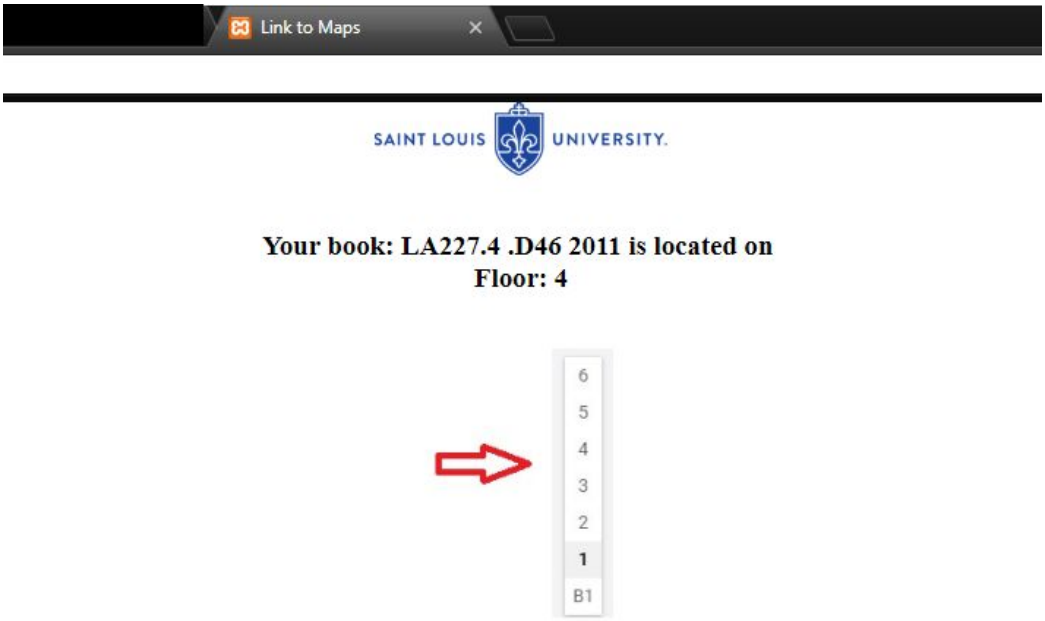
Purpose:
The library previously had a similar project that no longer is used due to obsolete code and software. Currently, the library has floor plans with call number ranges that cover a general area of each floor. Our project, however, assigns a call number range for each bookshelf which allows the project to narrow down a book's location to a shelf. This makes it much more efficient and faster for a user to find a book, as they only need to search one shelf.

How It Works:
A user browsing the library catalogue site will pick a book they would like to find in the library. When they click on the book's title, the browser will direct them to another page that contains a button: "Find Book on Map" which opens a pop-up prompting them to directly view the map (Google Maps) or they can receive a text message with a link that directs them to the map. Google Maps will contain a marker that pinpoints a bookshelf that contains the book the user picked. If the user is on a mobile device, they will be able to turn on their location settings and navigate towards the marker. For a more detailed explanation on each component, see tbe **Components** section.

Features:
- ➢ Google Maps has already provided the indoor floor plans of the library. Even though a few of the floor plans are missing shelves, we have included pinpoints to the locations of where the physical bookshelf is located.
- ➢ Each bookshelf corresponds to a range of call numbers. When a user finds a book on the library catalogue site, they have the option to be directly taken to Google Maps or have a text message containing a link sent to their phone. The former is for desktop users and the latter for mobile users.
- ➢ Mobile users will be directed to a page instructing them how to correctly display their book's location. This is necessary because the mobile version of Google Maps does not automatically display the correct floor plan based on the URL; the user must select it themselves.

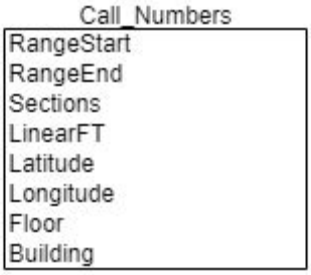
Components:

- ➢ Library Button(s)
  - The library currently has a button that says "Send a Text Message". When a user clicks on it, it opens a pop-up that enables the user to receive a text message with their book's call number.
  - The library IT staff have the authority to change the button and pop-up. Instead of the above bullet point, the button will say something similar to "Find Book on Map", with two buttons on the pop-up. One that directly takes you to Google Maps, and another that sends the text message. The text message will contain the same link as the other button that takes the user to Google Maps.



  - 1
  - This is an example of what the page will look like.
  - The buttons are programmed using PHP
- ➢ Google Maps API
  - This component creates a connection to the library database (see below).
  - It queries the database for a latitude, longitude, and floor number from the database and uses that information to construct a Google Maps URL that pinpoints where the bookshelf is located (with a marker).
  -  This is what a marker generally looks like.
  - For desktop users, the link will directly take the user to Google Maps.
  - For mobile users, the link will take the user to a separate page that displays instructions on selecting the correct floor level and contains the link that takes the user to Google Maps.
  - The API is programmed using PHP, HTML, and SQL.
- ➢ Call Number Database

- The database contains a record of all call number ranges that correspond to a bookshelf.
- Each bookshelf record also contains its coordinates (latitude and longitude), floor number, and building.
- Sections and linear feet were part of the Excel sheet the library staff provided us for this project.
- There is no primary or foreign key as this database only contains one table. The schema is as follows:



Using the Project:

- Download all of the files included with this project. A list and description of them are as follows:
  a. dbconnection.php
     - Contains the database information and creates a connection to the database. See **Using the Database** on modifying this file.
  b. detectDeviceType.php
     - Detects what type of device the user is accessing this project from
       - Desktop or mobile, with a distinction on mobile, iOS devices as the URL link differs
  c. floorPickerImage.png
     - Displays Google Map's floor picker as an image on the mobilePage.html page
  d. index.php
     - The library button accesses this file
     - Receives a book's call number from the library site, which is queried for data used to generate a Maps URL
     - Determines how to redirect the user depending on the user's device type
  e. markerurl.php
     - Takes the information retrieved from the database and generates a URL
  f. mobilePage.html
     - Displays instructions for the user on mobile devices.
  g. sluLogo.png
     - Displays the SLU logo on the mobilePage.html page
  h. getTextFromHtml.php
     - Used for testing purposes on a local machine
     - Contains a hardcoded URL that contains a call number to a book
- Move all of the files into a single folder with an appropriate name.
- Download a web-server program or open the library's web-server program.

- ➢ Upload the folder to the web-server program. If necessary, adjust the settings so that the index.php file is executed when the library button is clicked.
- ➢ On how to import and update the database, see the **Using the Database** section below.
- ➢ **On the library's web-server program,** the project should run properly when a user clicks on the button to Google Maps.
- ➢ **On a local machine,** open index.php and under 'include dbconnection.php', add a line below saying 'include getTextFromHtml.php'
    a. Comment out the line containing $callNum with // in front of it
    b. Save the file and close out.
    c. Open a browser and enter localhost/[folder-name]/index.php
    d. To test various books, change the end of the URL inside getTextFromHtml.php to a different call numbers. Replace spaces within the call number with %20

Using the Database:

- ➢ Importing the database:
    - ○ Our project includes a SQL script which can be imported into any database software. This contains the call numbers table with columns according to the schema above under the Call Number Database section of **Components**.
    - ○ It is important to create a username and password for the database, and name the imported database, preferably a name along the lines of "bookfinder_database" or "call_number_database".
    - ○ Modify the file dbconnection.php with the created username, password, and database name.
    - ○ The servername is the library's server's IP address.
- ➢ Updating the database:
    - ○ Open the database software and click on the book finder's database.
    - ○ The user can directly open the table within the database and manually add a record to the table. The user can use the database's forms to add a record, which is safer as it prevents possible deletion of data. Thirdly, the user can write SQL statements to insert/update the data in the database, all within the database software.
    - ○ Each column's type is specified, and it is important to follow those types when updating the database. For example, the call number ranges are strings (surrounded by quotation marks i.e. "Q1 .100 2018") and the coordinates are integers (i.e. 93.10534).

Constraints:

- ➢ Desktop:
    - ○ The user cannot use the Google Maps generated directions when they are redirected to Maps through the library button.
        - ■ This will reset the marker that pinpoints the bookshelf to the library entrance, as well as the floor number to level one.
        - ■ The user must right click on the marker and click on "Directions to here" and input a starting point in order to get the most accurate path and directions from the starting point to the book shelf..

- Mobile:
    - If the user has installed Apple Maps or Google Maps on their device, the link will automatically open the app. If not, the link will direct them to an app store to install one.
    - If the user would like to use their phone's browsers, they must copy and paste the link from the page to their browser. However, this results in a very slow experience for the user, as Google Maps does not appear compatible with a mobile device's browser.
    - The user must turn on their location settings to manually navigate the map to the marker that pinpoints the bookshelf
    - Using the Google Maps generated directions may reset the marker's location

Assumptions:

- Users accessing the project through the library catalogue site must:
    - Know how to browse the catalogue site
    - Be exposed to or able to easily see the button that uses this project
    - Understand the basics of using Google Maps
    - These apply both on desktop and mobile devices
- Library IT/staff deploying the project must:
    - Be knowledgeable in the library's current web server
    - Understand the implications of modifying a database
    - Know how to read and understand basic server side code and client side code: PHP, SQL, HTML/CSS
    - Understand how to use a database software's resources