**User Guide for the Synagogue Cemetery Management System**

By Emily Crabb, Samuel Kearns, Brittany Regrut, Louis Seefeld, and Evan Alexander

Overview of the Program

The synagogue cemetery management system was created to provide cemetery managers with a user-friendly program to maintain and access the cemeteries’ databases of information about the graves. It was specifically created for the Beth Shalom Synagogue in Pittsburgh, Pennsylvania. This synagogue requested this software to replace their outdated cemetery management program that was not easy to use or to alter to meet new needs. The new program is Java-based, so it can be run on any of the most common types of platforms. This also makes it fairly easy for software developers to revise, as Java is a standard and well-known programming language. The program also uses an H2 database, which is an open source Java SQL database that is available for many types of platforms. As such, this new synagogue cemetery management system is easy to set up, use, and revise to meet any specific needs for a given cemetery. Installation instructions are provided in the next section, while details on the functionality are outlined in the subsequent section.

Installation and Execution Instructions

Installation of the Synagogue Cemetery Management System is relatively straightforward. However, we encourage first time users to allow the product to be installed by a developer or someone with knowledge of the project file structure.

The source code can be downloaded from our GitHub repository: https://github.com/skearns4/Synagogue-Cemetery.

The files necessary for installation are *Driver.java*, *MainWindow.java*, *DisplayPanel.java*, *EditEntry.java*, *EditWindow.java*, *Entry.java*, and *NewEntry.java* located in the *src* folder. The project directory should look generally like this:

.Synagogue-Cemetery

|

+-- h2

+-- src

| +-- Driver.java

| +-- MainWindow.java

| +-- DisplayPanel.java

| +-- EditEntry.java

| +-- EditWindow.java

| +-- Entry.java

| +-- NewEntry.java

After the files have been downloaded, the next step is to compile the Driver file. This can be done by navigating to the src folder and running the following command:

~$ javac Driver.java

This should result in the creation of several .class files in the src folder.

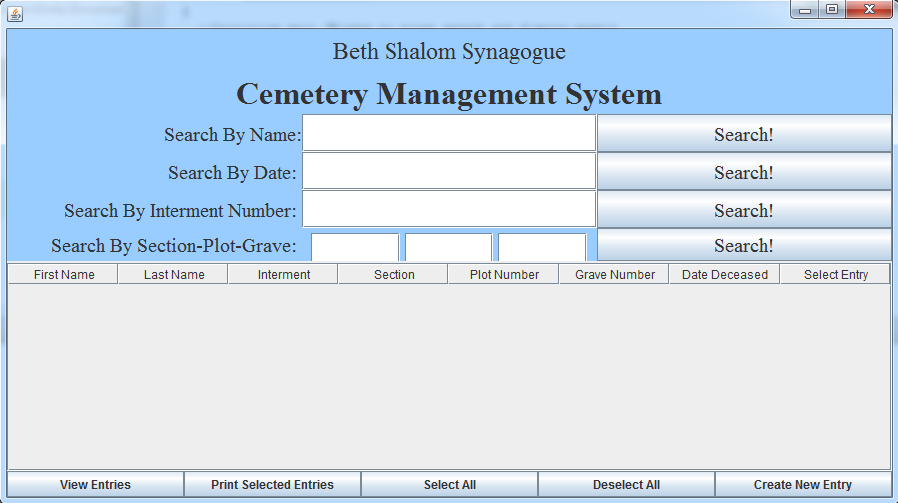
Now, all that is left to do is to move the h2 folder into the src folder so the program can locate the required database files.

To run the program, the following command should be entered in the terminal:

~$ java Driver

The program should now be running with the main window on screen.

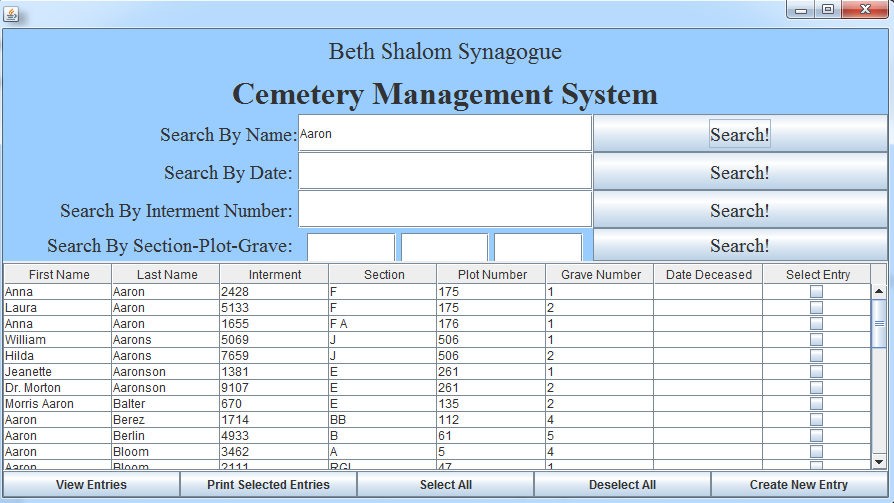
Program Functionality

When the program is first launched, a window pops up with an interactive user interface. This window is shown in the figure below. 

Once the program is launched, it has four major categories of functionality: searching for entries in the database, editing existing entries, adding new entries, and printing information to a text file. Each will be discussed in some detail below.

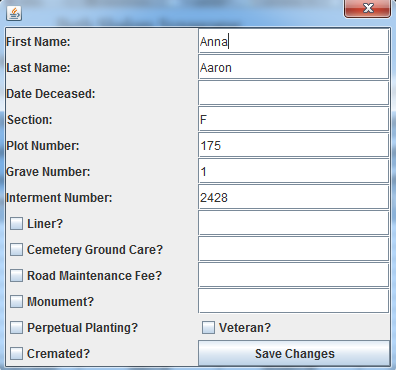
*Searching for entries in the database*

Entries can be searched for by name, date of death, interment number, section number, plot number, or grave number by inputting the appropriate information in the appropriate text box and clicking the search button. When searching by name, a first name, a last name, or a full name can be used. Any name that contains the entered text will be returned, and the search is not case sensitive. The matching results are then displayed in the table in the main program window. If any information is not known for a given entry, the corresponding spot in the table is left blank. If there are multiple entries matching the given inputs, all of them are displayed. If there are no matching entries, the display table empty (as it was when the program first began). An example of searching for the name “Aaron” is shown below. Results with Aaron as either a first or last name or as part of the first or last name (ex: Aarons) are displayed in the table. As there are more entries than fit on the screen, there is a scroll bar to access the other entries.



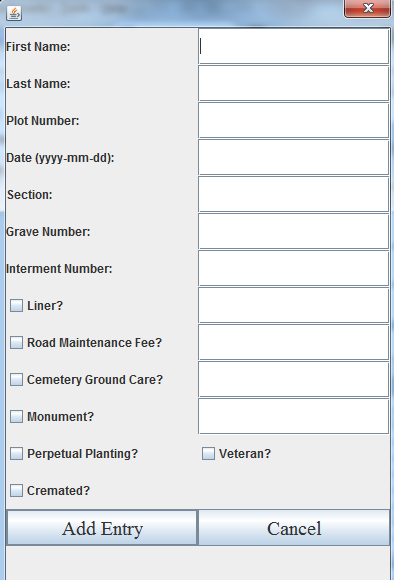
*Editing existing entries in the database*

One can edit entries by first selecting the entry by clicking on the box in the select entry column and then clicking on view entries. This opens a new pop-up window where the existing information on an entry is displayed and new information can be inputted and saved. An example of the view entries pop-up window is shown below.



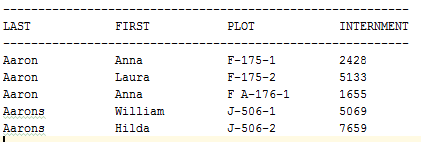
*Adding new entries to the database*

One can add new entries to the database by clicking on the create new entry button in the main program window and then filling in the appropriate information for the new entry. Information can be entered for all the fields existing in the database. The create new entry pop-up window’s format is shown in the figure below. It is very similar to the view entries window format, but it is initially blank as there is no existing information for the new entry to display.



*Printing information to a text file*

Information from the database can be printed to a text file by first selecting any desired entries in the main program window and then clicking the print selected entries button. The first and last name, section-plot-grave information, and interment number for each of the selected entries is printed to a text file. The text file is named based on the time it is created with the format YYYY-MM-DD\_hh\_mm.txt (year-month-day\_hour\_minute.txt). For example, printing the first five entries from the first search for “Aaron” at 8:01 pm on November 22, 2015 would result in a text file named 2015-11-22\_20\_01.txt that reads:



These four actions (searching for entries, editing existing entries, creating new entries, and printing selected entries) form the functionality of the program.