Cemetery Management System

Samuel Kearns - Project Manager (skearns4)

Brittany Regrut - Quality Assurance (bnr12)

Louis Seefeld - Software Developer (lfs13)

Emily Crabb - Software Developer (ejc44)

Evan Alexander - Software Developer (ea914)

Due date: 10 NOVEMBER 2015

CS 1530 - SPRINT 4 DELIVERABLE

Sprint 4 Accomplishments

During this sprint, we completed one new user story relating to adding real data to the database. Lonnie, the cemetery manager, provided us with a CSV file with real data near the end of the last sprint, but we did not have time to add the data during the previous sprint. Thus, we added this data to our database during this sprint, which required adding new fields to our database and modifying our search functions, as is described in the sections of this report dealing with user stories. We also adapted our existing unit tests in the MainTest.java file to work with the new data. This was necessary because the tests use real names and plot numbers from the database and the previously selected names and numbers from our test database do not exist in the real database.

In addition to these functional changes, we also made some more cosmetic changes to our code. We renamed the .java files and function calls to conform to regular Java naming conventions and eliminated commented-out code. These changes increase the internal quality of our software. We also added more columns to our GUI display such as section to reflect the additional information included in our database as a result of implementing the user story about using real data. Finally, we added a view entry button that opens a pop-up window. As of now, this pop-up window just displays “Test,” but it will eventually display all the information about a given entry in the database and probably allow the information to be edited. As such, we have added an additional user story dealing with editing existing entries.

We did not meet with the customer during this sprint, but we did focus on incorporating the feedback we had received when we met with the customer near the end of the previous sprint. During this meeting, Lonnie expressed concern that it would be difficult to include all the required fields in our database. Therefore, the focus of this sprint was expanding the database to incorporate all the fields required for the real data.

We used the same methods of communication during this sprint as we had during our previous sprints. We mainly communicated via Slack, with a few short meetings after class to divide the work and discuss any problems. Unlike in previous sprints, we did not meet in person on the weekends. However, this was less important in this sprint because we figured out how to perform code reviews on GitHub. We had examined each other’s code in person in the past, so using GitHub reduced our need to meet. We also had no problems relating to configuring our IDE or database driver, which were other reasons we had to meet more frequently in person during some previous sprints.

User Stories Completed and Updated Backlog

Link to the repository:<https://github.com/skearns4/Synagogue-Cemetery>

In this sprint, we completed one new user story. As listed in the backlog included in our previous deliverable, this completed user story is:

|  |  |  |
| --- | --- | --- |
| Rank | User Story | Difficulty Estimate |
| 1 | As an administrator, I want to be able to access our existing burial plot information in this new database so that no information is lost in the transfer. | 4 |

Also, due to the inclusion of real data in our database, we have been forced to revisit our previous user stories relating to searching by name and plot number to account for differences between our sample data format and the format of the real data provided, as will be discussed in the following section.

We added one additional user story relating to editing existing entries in the database, so the updated backlog of user stories is:

|  |  |  |
| --- | --- | --- |
| Rank | User Story | Difficulty Estimate |
| 1 | As a user, I want to be able to print search results and individual user entries so that I can easily share the information with others. | 4 |
| 2 | As an administrator, I want to be able to edit existing entries so that I can correct any errors in the information stored in the database. | 4 |
| 3 | As a user, I want to be able to view a map of the cemetery, including number of open graves per plot so that I can more easily show clients the layout of the cemetery. | 8 |
| 4 | As a user, I want to be able to able to see the status of plots that are in close proximity to a current plot so that I can easily determine whether clients can purchase adjacent plots. | 8 |
| 5 | As a user, I want to be able to click on the map of the cemetery so that I can easily look up information for a specific plot. | 8 |
| 6 | As an administrator, I want to be able to manage payments for the plots so that it is easier to coordinate plot and payment information. | 16 |

Justification for Choosing These User Stories

We chose to implement the user story relating to adding real data to our database for several reasons. For one, we had already completed the basic GUI and database functionality needed for our project and tested this functionality with the existing database. The user could search the database by name and plot number and create new entries. This functionality is approximately equivalent to the functionality of the program the cemetery currently uses, so it is in line with our goal of making our software as useful as the existing software. Therefore, the next logical step towards this goal was to implement the user story relating to adding real data to the database.

Additionally, the user gave us the real data at the end of the last sprint. We thus did not have time to implement this user story during the last sprint but had plenty of time to implement it during this sprint. When we met with the customer during the last sprint, he also expressed some concern over the differences between our sample database and the real data. As a result, we decided adding the real data as soon as possible and making any necessary changes to our database structure was the best way to allay the user’s fears.

After we added the real data to the database, we realized we needed to revisit the previously completed user stories related to searching by name and plot number because the name and plot formats were different than we anticipated. These problems are discussed in more depth in the section on defects, but some examples include plot numbers containing letters and names containing titles. As a result, we altered some of our parsing functions and data types.

We did not implement any of the user stories related to the map primarily because the user still has not provided us with a high enough resolution map to be able to use in a user interface. For this reason and because the map user stories all have high difficulty estimates, we do not anticipate implementing these user stories.

We added one additional user story relating to editing existing entries in the database because we anticipate that an administrator might need to correct entries. It is reasonable to expect there could be data entry errors, and there is currently no way to correct such errors in the program. We have thus set the foundation for completing this user story by adding a view entry button that currently only displays “Test” but will eventually display all the information for a given database entry and allow the user to edit this information.

Customer Feedback

During this sprint, we did not meet with the customer, as we met with the customer near the end of the previous sprint. Our previous meeting was with Lonnie, the cemetery manager. He gave us some real cemetery data in CSV format and expressed concerns over how difficult it would be to include all the needed fields in our database. At that point, our database only contained first names, last names, plot numbers, and dates of death. It also only had about ten sample entries.

Our work during this sprint is a direct result of this meeting with Lonnie. We focused on the user story relating to creating a database with real data, as this was the first time Lonnie provided us with real data. In order to complete this user story, we had to add additional categories or fields to the database, such as section, veteran status, and grave number (which is different than plot number). This expansion of our database addressed Lonnie’s concerns that it might be difficult to include all the necessary fields. We also showed that it is easy to add additional fields to the database should the user ever feel the need.

Completing this user story also required converting the CSV file with the existing data into our database format. We discovered that this conversion process is fairly straightforward and did not require a great deal of time or effort, so we will be able to add any additional data quickly and efficiently. This is important because we believe Lonnie has only provided us with a small fraction of the overall data and might want the rest of the data added on short notice.

Finally, our prior meeting with Lonnie also shaped the user stories we did not address, primarily those stories focused on the map interface. As our user has not provided us with a sufficiently high-resolution map to even attempt a GUI, we have not done any work on those user stories.

Defects and Limits on Functionality

One defect we discovered was that our changes to the database and GUI broke our search by plot number functionality. We discovered this defect when the testPlotNumber175 unit test failed. We then ran the regular Driver.java file and tried to search by plot number with the GUI. The expected behavior was that searching by a valid plot number would result in the corresponding database entries being displayed on the GUI. Instead, a data conversion error message was printed and no results were displayed. The defect was caused by the fact that the plot number was previously stored as and searched for as an integer, while the real data required it to be stored as a string because some of the plot numbers contained letters. We thus corrected the defect by searching for the plot number as a string.

We discovered a similar defect with the create a new entry functionality, which was also broken by changes to our database and GUI. In this case, we caught the error running the Driver.java file and trying to add a new entry with the GUI. The expected behavior was that a new entry would be created, which could then be found by searching by the appropriate name or plot number. Instead, a column count does not match error message was printed and no new entry was added to the database. This defect was caused by the fact that the NewEntry.java file had not been altered to adapt to the expanded database that has more fields. We thus corrected the defect by adding more fields to the add entry pop-up window with supporting code in the NewEntry.java file.