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: 13 OCTOBER 2015

CS 1530 - SPRINT 2 DELIVERABLE

Sprint 2 Accomplishments

During this sprint, we generally focused on developing a database to hold the plot information and a basic GUI to allow the user to search the database and create new entries. We decided to use the H2 database engine, which supports the Java SQL database standard. We chose this database management system because several members of our group have experience using SQL and because it interfaces nicely with IntelliJ, the IDE our group has decided to use for developing this project. However, we did encounter some difficulties with using this database because of versioning issues. Older versions of H2 (before Version 1.4) use a different file extension than newer versions (Version 1.4 and up) do. We did not initially realize this or that IntelliJ has a specific H2 driver that also depends on the version, so initially some group members could not use database files generated by other group members. Once we discovered this was a problem, we decided to use the newest version and update the IntelliJ driver accordingly. We also updated our original GUI from the walking skeleton to include buttons and text boxes to create a new entry, search by name, and search by plot number. The outputted information is also displayed in the GUI. The details of the functionality added during this sprint are included in the later sections on the user stories and defects.

We met with the customer twice during the course of this sprint. In our first meeting, we specifically met with Lonnie, the cemetery manager, rather than Rob, the director of the synagogue who initiated the project. During this meeting, we learned that Lonnie often needs to print entries in the database. As such, we added a user story to our backlog relating to this new information (discussed in the section on user stories). In our second meeting, we met with both Lonnie and Rob. They expressed interest in integrating our program with their existing program but agreed that this was a low priority.

Additionally, our group had some problems with communication during this sprint. Due to different schedules and commitments, it was difficult for us to find time to meet in person. We continued using Slack for most of our communication, but we have found this method of communication can be less than ideal for dividing work and helping other team members. Specifically, the small number of in-person meetings delayed the process of each group member getting the code working, as problems with IntelliJ and the H2 database were slow to resolve remotely. We had incorrectly assumed that our walking skeleton, which everyone got working with minimal problems, was representative of our project. In actuality, we had not all used the IDE to test the walking skeleton, and the walking skeleton really did not have any database functionality. As such, the problems with properly configuring IntelliJ and the H2 database only became fully apparent during this second sprint. For the next sprint, we intend to meet more often in person, though we hope it will be less of an issue because everyone has now sorted out the IDE and database problems.

User Stories Completed and Updated Backlog

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

In this sprint, we completed several user stories, though perhaps not as smoothly as we originally intended. As they were listed in the backlog included in our previous deliverable, these completed user stories are:

|  |  |  |
| --- | --- | --- |
| Rank | User Story | Difficulty Estimate |
| 1 | As a user, I want to view information for existing burial plots so that I can see if plots are available and look up information for existing clients. | 4 |
| 2 | As an administrator, I want to create a new entry so that I can keep track of newly sold plots. | 8 |
| 4 | As a user, I want to be able to search the burial plot entries by name so that I can look up information on a specific client. | 2 |
| 5 | As a user, I want to be able to search the burial plot entries by interment number so that I can look up information on a specific client. | 2 |

The justification for focusing on these user stories is given in the next section of this report, while some defects and limits on their functionality are given in the final section. Additionally, through conversations with the customer, we have added a new user story relating to printing the search results. Thus, the current, updated backlog of user stories 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 |
| 2 | As a user, I want to be able to search the burial plot entries by last name or full name (rather than just first name) so that I can look up information on a specific client. | 2 |
| 3 | 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 |
| 4 | 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 |
| 5 | 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 |
| 6 | 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 |
| 7 | 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 |

Other than the second user story relating to printing and the addition of a user story relating to searching by last or full name (discussed in the defects section), the remaining user stories are unchanged since the last deliverable.

Justification for Choosing These User Stories

We chose to implement the four user stories related to viewing information on existing burial plots, adding new burial plot entries to the database, and searching for data plots by name and internment number for several reasons. For one, all of these user stories were very important. We initially ranked these user stories as 1, 2, 4, and 5 in our backlog because the cemetery manager and synagogue director, our users, value this functionality highly. The existing program for managing the cemetery plots can display information, add new entries, and search the entries by name and internment number, so adding this functionality to our program is necessary to even make it as useful as the existing program. The other user stories, such as being able to select plots from a map, are desired add-ons but are not as crucial to making the software useful.

Additionally, all four of these user stories were closely intertwined with the task of creating a back-end database for our program. Our focus of this sprint was really creating this database, so we prioritized implementing user stories dependent only on basic database functionality. Searching, adding entries, and displaying information are all critical tasks for a working database, so these user stories were very logical to implement.

Finally, these user stories did not require a very complex GUI. The user stories relating to selecting entries from a map will require significant alterations to our GUI, as we will need to add some sort of button or link to each plot on a map. Our walking skeleton just had one search bar with an accompanying button, and it was easier to extend this to include several buttons and search boxes than to implement an entirely new map interface.

In sum, these user stories were relatively simple to implement as we developed our back-end database, requiring only minor changes to our front-end GUI. They also are important to our users, as our program will be almost useless if it cannot store entries, add information, or search for desired information. As we go forward, we plan to read in the user’s existing database information before we expand to include other user stories. This will hopefully allow our program to match and then surpass the functionality of the user’s current program.

Defects and Limits on Functionality

Though we have implemented four user stories, they are not as thoroughly or smoothly implemented as we originally intended and will have to be polished in subsequent sprints. For one, the GUI text fields are currently very finicky. The search by name only accepts a first name and then returns the information for all of the entries with the specified first name to the command line interface. If an incorrect name is given, it is ignored and no error message is displayed either in the GUI or the command line interface. We did not view this as a major defect for this sprint because we are only using a very small test database, and we instead prioritized also including add entry functionality. However, we will need to return to this problem before we add the real data from the existing database, as we expect the user will at minimum also want to be able to search by last name and have the information be displayed in the GUI. For this reason, we have added an additional user story to our backlog relating to searching by last name or full name.

Additionally, there are several limits and defects with the create a new entry functionality. One of these is the fact that the add entry pop-up window does not disappear or get reset once the entry is added to the database. The user must manually exit it, which might be confusing. However, we again focused on adding functionality over polishing the results, so we will probably clean this up later.

A genuine defect that we discovered during casual testing is that the date format displayed in the add entry pop-up window (mm/dd/yyyy) did not match the date format needed by the program (yyyy-mm-dd). Instead of adding an entry to the database, an error message was displayed in the command line interface that did not really explain the problem to the user. We therefore corrected this error by changing the date format displayed in the pop-up window.

We have not really developed much in the way of formal unit or system tests yet due to our focus on creating the structure of the database and accompanying GUI. As we focus more on user functionality in subsequent sprints, we will add more formal tests.

An additional defect is that Evan does not yet have a computer. As such, he does not have a running version of the code.