

Cemetery Management System

Emily Crabb - Software Developer

Louis Seefeld - Software Developer

Evan Alexander - Software Developer

Samuel Kearns - Project Manager

Brittany Regrut - Quality Assurance

Due date: 29 SEPTEMBER 2015

CS 1530 - SPRINT 1 DELIVERABLE

Ranked Backlog of User Stories

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
3	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
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
6	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
7	As a user, I want to be 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
8	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
9	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

Walking Skeleton

<https://github.com/skearn4/Synagogue-Cemetery/blob/master/src/walkingSkeleton.java>

Repo: <https://github.com/skearn4/Synagogue-Cemetery>

Our walking skeleton compiles and displays a blue GUI window with a search box that currently has no functionality. It is written in Java and uses a Swing GUI. There are currently functioning database files in the repository, however, they don't directly interact with the code in place. In subsequent sprints, we will add database functionality, search capabilities, and an interactive map of the cemetery, as described in the other sections.

Test Outline

There are many aspects of our program we will eventually need to test. For one, we will need to test linking a back-end database with a front-end search which can issue queries handled by JDBC, Java's Database Connectivity. Our GUI will be written in Java as well. We are researching databases to host our data, and we are specifically considering using H2 database. There are two other options that we are deciding between, but we would like to give them all a fair chance before making a final decision. Additionally, testing a smaller scale version of the database will help us achieve the functionality we need, which then can be ported over to a larger working version of the database.

A search functionality will have to be tested as the database will be a scalable object dependent on the amount of data entered. The search functionality should perform within seconds regardless of whether the database has 10 entries or 10,000 entries. Testing will ensure that the search functionality does not decrease as we enter more data into the working software. In addition, the queries must return accurate data.

To test reliability, it will also be important to test the program across various machines running different operating systems, processors, and instances of the program. Java should mitigate compatibility issues, but testing across machines as the program develops will make sure that any extraneous errors do not impact further progress.

We will also need to test the accuracy of the portion of our program that reads in the currently existing data. The data for existing burial plots is currently stored in a database, and we will need to have our program read this data into our database. We will then need to check the accuracy of this transfer process by comparing search results using our program to search results using the existing program.

In summary, our test outline will become more detailed as we make more design decisions regarding our database and user interface.

Description of Decisions

Not much initial background research was done on this topic by our team. We discussed some possible user stories for our software, but most of the information in our backlog of user stories came from direct conversation with the customer. Before talking to the customer, we established a system for communication, creating a Slack page for easier whole-group communications. We agreed on the roles that each of us would take on. We decided that our software will be best executed with a back-end database and a user-friendly GUI front-end that will be created with Swing.

The most significant problem we encountered during our first sprint was in interacting with the customer. The customer was slow to respond to our emails, and our project manager could not meet with the customer until the Friday before the sprint was due. Additionally, we have interacted with two people at the cemetery with slightly different goals. Rob is the executive director of the Beth Shalom Synagogue, while Lonnie is the cemetery director of the Beth Shalom Cemetery. Lonnie is the sole user of the existing software that manages the cemetery plot information and is very familiar with it, so he is not particularly interested in a new program with added functionality or greater ease of use. Lonnie would prefer to continue using the existing program – the same one he’s been using for the past 15 years. Rob, in contrast, is the one who initiated our project and is the primary stakeholder. Rob wants a new, improved program that anyone could use. Going forward, we will have to balance their competing goals by creating a program that implements the functionality Rob, our customer, desires but is still acceptable to Lonnie, the primary user.

Another major aspect of our decision making process was prioritizing the user stories. Through our interactions with the customer, we learned that they want a program that manages

cemetery plots. They want to be able search for existing plots, check whether plots are available, and add new entries when people purchase new plots. The cemetery currently has some outdated software that performs this functionality, though it does not make searching very user-friendly. We thus prioritized implementing the functionality of their existing software in our new software so that they do not lose functionality. We also prioritized searching by name and interment number because those are the actions for which they most commonly use their current software.

The cemetery managers also would like a map of the cemetery that they can click on to view information about specific plots. They currently have no easy way to look up a plot by its location, rather than its owner's name or interment number. Thus, after getting a searchable database working, our next priority is implementing this interactive map.

Additionally, the cemetery currently uses two separate programs to manage payments and to manage plot information. Thus, we included a lower priority user story involves adding payment management functionality to our database, effectively eliminating the need for their second program.

When it came to deciding what kind of database and user interface framework we would use, some research had to be done. It was initially proposed that we use the newer, JavaFX framework to develop our GUI. However, after some deliberation between our group members, it was decided a better option would be to use something more of us were familiar with. We decided on the Swing framework because our UI is fairly straight forward, it doesn't require any additional software installation, it's very heavily documented, and several members of our group have used it in the past. Which database we will use is actually still being discussed. It seems we've narrowed our options down to three: Apache Derby, HSQLDB, and H2, which we are currently leaning towards. All of these provide options for embedded databases and don't require

the commissioning of any external servers. They are also extremely lightweight when compared to many other database options and considering there are only a finite number of cemetery plots, these seemed most appropriate.

For this sprint, our communication with the customer was very ineffective for the majority of the two weeks. For the next sprint, we plan to be more proactive in repeatedly contacting the customer if we do not receive a response. We also established different means of contacting our customer, so this should not be an issue in the future. Ideally, we will establish a weekly schedule for communicating with the customer to receive information and share status updates.