Post-Mortem Discussion (5/16):

1. **To what degree do each of us feel that we have achieved the outcome of "[following] systematic and logical design procedures to design a solution that meets requirements"? (Project phase report).**
    a. Marcus: 7
    b. Paula: 7
    c. Alyssa: 6
2. **Three major lessons learned**
    a. We feel that one major lesson learned was to have open communication in asking for help or setting up time frames in order to work on the App together.
    b. Another major lesson learned was to set aside time to dedicate to weekly meetings that contributed to making progress as a team.
    c. Lastly, the major lesson learned was to have that sense of thoroughness in order to really comprehend how time consuming and detailed the design process of it all was going to take. It was a lot easier said than done, and we regularly found new design decisions that needed to be made during the implementation phase.
3. **Usefulness and difficulty of git**
    a. Marcus: I would rate the utility as an 8 (extremely useful but not something that I couldn't do without) and the ease of use as a 4 (a little fiddly in some circumstances, but usually pretty easy).
    b. Paula: I would rate the usefulness of git as an 8 and the difficulty of it around a 6 as this is the first time I have worked with it before and I'm still learning how to use it.
    c. Alyssa: I would have to say the usefulness was around a 9 as it showed us where all the code was and it was easy to see what files were in the directory. The difficulty of git was around an 8, I personally have never used Git and it was a little challenging/confusing at times with pushing, pulling, and committing the lines of code.
4. **GitHub project management features**
    a. The kanban board helped us to remember the tasks we had scheduled, but most of the project management tools didn't seem essential to development and we either forgot to make use of them or just didn't want to be bogged down in learning more new things.
5. **Percent of original project completed**
    a. We estimate that roughly 75% of the functional requirements have been met. Some of them look a little different than we had originally planned (moving from one window to another, for example), but the functionality is mostly there.
6. **Anything we would change?**
    a. We agreed that the program design was good, but we would change our plans to allocate more time to work on the project (starting group work sessions earlier because even though we usually worked individually, knowing that we would all be there working at the same time helped stay focused) and make sure that we were all on the same page much earlier (had wxWidgets installed, VMs working, programming methods, etc.).

Pair Programming Session (5/16):

Git Commit: 1a3c6e0dc902fcaa7aee64feb16ccc298d28ffbb

Alyssa and Marcus got the calendar to open in the program.

Scrum Notes Week 9: Paula Rodriguez (5/11)

1. **What have I completed since the last meeting?**
    a. Marcus has finished working on the credits and configurations menu
    b. Alyssa is still working on fixing her panels.
    c. Paula was able to implement the database so a user can add a contact.
2. **What am I aiming to complete before the next meeting?**
    a. Marcus is going to work on taking unique notes for recurring meeting (multiple note files)
    b. Alyssa is going to work on restarting her VM and fixing her panels.
    c. Paula is going to work on making the contacts show up on the contact window and implement a function to add emails into the database.
3. **What is currently blocking my progress?**
    a. Considering this is the final week of classes and finals week is approaching, there is an issue of time. One of our group members is also experiencing technical difficulties with their VM.

Scrum Notes Week 8: Marcus Schmidt (5/4)

4. **What have I completed since the last meeting?**
    a. Marcus has added functionality to be able to manage meetings within the program (displays more information, can delete or edit meetings).
    b. Alyssa has made progress on fixing bugs in her code.
    c. Paula has not had time to work on the project.
5. **What am I aiming to complete before the next meeting?**
    a. Alyssa is aiming to finish fixing bugs and get the calendar view fully working.
    b. Paula's goal is the same as last week--incorporating the database into the contacts window.
    c. Marcus will work on adding the intro/credits and configuration windows.
6. **What is currently blocking my progress?**
    a. A combination of time restraints and just having lots of errors to work through.

Scrum Notes Week 7 : Alyssa Diaz (4/27)

1. **What have I completed since the last meeting:**
    a. Marcus has finished the database and has integrated it into the code.
    b. Paula has finished implementing a prompt to add into the contact
    c. Alyssa has finished up CalCntrl.h
2. **What am I aiming to complete before the next meeting**
    a. Marcus is aiming to work on MV_Head and building ways to manage the meetings within the program.
    b. Paula is working on utilizing the database and adding contacts to it

c. Alyssa is finishing up the calendar.cpp file.

3. **What is currently blocking my progress?**
    a. Progress is actually running smoothly due to weekly meetings we have set aside to run/test/create

Entry by: Marcus Schmidt (4/22)

Latest Git Commit: 850cadab1929a0e80e3d527ffe7301159bc233f7

This last commit brought the database code to a point of functionality that I thought was ready to be pushed to the GitHub repository. Below is a breakdown of the most important things to know about this code and how to use it. I'd be happy to go into more detail during a meeting if either of you are interested.

- All of the database code is inside of UserData.cpp. No specific SQLite knowledge is needed anywhere else in the program.
- Instead, the UserData.h header file can be included anywhere in the program so that one of the public functions can be called to retrieve data.
    - For example, to get a list of all meetings in the database, any file in the code with the "UserData.h" and <vector> headers can simply do the following: "std::vector<Meeting *> meetings = UserData::GetMeetings();"
        - This vector can now be searched for the desired meeting. For example, if it's called "Lecture" with a contact named "Software Engineering", you can iterate through the vector and compare with the results of the Meeting.h GetName() function, etc.
        - Alternatively, you can retrieve meetings for only one contact by passing a string with the contact's name into the UserData::GetMeetings() function. This helps to filter results out more efficiently and save memory.
        - IMPORTANT NOTE: Code that returns pointers like this allocates dynamic memory that has to be freed when you're done using it (as denoted by the *). This can be as simple as writing a for loop where 'i' goes from 0 to meetings.size() and executes: "delete(meetings[i]);"
    - Similarly, all contacts can be retrieved with: "std::vector<std::string> contacts = UserData::GetContacts();"
        - You will also need the <string> header file in addition to "UserData.h" and <vector> to use this function.
        - Because this function does not return pointers, there is no need to free any memory.

    - A list of all public database functions and their required parameters can be referenced in UserData.h. If one of the parameters is already assigned a default value (such as “sqlite3 *database = nullptr”) then it is optional and you can most likely ignore it--they’re almost always just used by the database to increase efficiency or for testing purposes.

- Once you pull this code into your local repository, you will have to install SQLite3 on your virtual machine before you will be able to compile and run the code. I’ve included directions on how to do this in the README.md file.

- To test that the database works, I’ve added several menu options to the main Meeting window to print the meetings and contacts in the database, as well as to reset it. You will have to do a reset to build the database for the first time before you will get any results. From there, you can add new meetings and contacts in the Create Meeting window and see them appear in the results until you reset the database again. You can also open a random meeting, type notes into the text entry field, and they will be saved for the next time that meeting is opened.

- As discussed in the Discord, there are no longer dedicated Contact objects. Instead, they are handled as strings.

UserData.cpp will almost certainly need to be expanded or do more filtering as the program needs it, such as only retrieving the meetings that occur on a particular day, only retrieving recurring meetings, etc. If something like this is required, let me know and we can talk about how to implement it.

Scrum Notes Week 6: Paula Rodriguez (4/20)

1. **What have I completed since the last meeting?**
    a. Marcus: Has built a foundation for implementing SQL code so it can read and write and information given by the user
    b. Alyssa: Has set up a frame for the calendar and needs to tie them up with the rest of her files.
    c. Paula: Working on the prompt in contacts so a user can input contact information and save it.
2. **What am I aiming to complete before the next meeting?**
    a. Marcus: Will continue to work on the database and will work on integrating it into the code.
    b. Alyssa: Is working on finishing up the calendar.
    c. Paula: Is working on finishing up the add new contact prompt.
3. **What is currently blocking my progress?**
    a. As a whole, the thing blocking the project is time management, we do not have a lot of time to put into the project because of other responsibilities we have to attend to.

Lab 9 Timeline Review Discussion (4/13):

1. **Which features are currently complete/practically complete? Roughly how much time did each feature take to complete?**
    a. The functional requirements to create a meeting and open it to have access to the meeting link and taking notes is mostly complete. All of the components for this feature took roughly 25 hours to complete.
2. **Which features are in-progress? Roughly how much time has each taken so far, and how much do you think it will take to complete?**
    a. The calendar view window is in progress, it still needs to display a list of all meetings and times in a weekly calendar format. I think it should take about another week to complete. It's just a matter of setting aside some time to do so.
    b. The contact view window still needs to add an option to create a new contact, as well as have a way to add contact info. So far, these features have taken up approximately 2 hours of work. It may take up to an extra 3 hours to complete, give or take.
    c. The SQLite back-end is only partially integrated into the program, but it is in a solid working condition. It has taken roughly ten hours to get the foundation set up, and I anticipate it will need another five hours to incorporate the remaining features the app is currently capable of. As the program develops, this feature will need periodic revisiting.
3. **Time estimate from each group member.**
    a. Marcus: I can definitely commit to five hours a week at minimum, but my goal will be to sustain an average of eight hours a week, and I expect that there will be a week or two where I'll put in closer to ten or twelve hours. I estimate that I have somewhere between 30 to 45 hours left to put into this project.
    b. Paula: Has around 5-10 hours a week available to work on the project. Time is dependent on how much work she has for her other classes. Overall: Can commit anywhere from 25-50 hours into the project.
    c. Alyssa: Has this week available entirely to work on her portion of the DailyHub App, has the calendar view skeleton set up- just needs to fix some errors here and there. Will stick to a 50+ hour time estimate to wrap up her portion.
4. **Are any functional requirements at risk of being unfulfilled? Which ones might we need to cut, and why would we choose them?**
    a. All of the importing and exporting features will most likely be cut. They are largely stretch goals because they are not core functionality, and it's possible they could be more complicated than expected and become a time sink.
    b. It is possible that the configuration window will need to be cut because it is more complicated than the help/intro windows, but Paula might have time for it depending on how quickly the Contacts View comes together.

Scrum Notes Week 5: Marcus Schmidt (4/13):

4. **What have I completed since the last meeting?**
    a. Paula completed the split window for the contact view.

b. Alyssa pushed the base code for the calendar view to GitHub.
c. Marcus finished the window features to create a recurring meeting and created a basic text entry interface for taking notes during a meeting.

5. What am I aiming to complete before the next meeting?

a. Paula will add the ability to create a new contact from the Contact View.
b. Alyssa will incorporate the base Calendar View code into the app so that it can be opened from a menu, as well as resolving the last problem with compiling wxWidgets code.
c. Marcus will begin work on the SQLite back-end to read and write user data.

6. What is currently blocking my progress?

a. Alyssa is still grappling with strange errors trying to get wxWidgets to compile, which makes code development much harder.
b. Paula is still dealing with limited time to code.
c. The cooperative work session last week did help with productivity, so we will schedule another one for this upcoming week.

Scrum Notes Week 4: Alyssa Diaz (4/6)

4. What have I completed since the last meeting:

d. Marcus has been working on the meeting view creation. He is still trying to figure that out to be able to create recurring meetings.
e. Paula is still working on Contact View.
f. Alyssa is still working on Calendar View.

5. **What am I aiming to complete before the next meeting**

d. Marcus is working to finish up last week's goals and create text entries for note-taking in meetings.
e. Paula is working on finishing up the contact view with a split window.
f. Alyssa is finishing up the calendar view with proper display of windows.

6. What is currently blocking my progress?

b. Progress is being blocked by lack of time management . We have other classes going on, and do not actively use the time we have to sit down and figure it out,which is how we fall behind.
c. We can set up meetings to help use that time wisely. Even if we are not working on the same things, we can just block off time frames to actually be productive.

Scrum Notes Week 3: Paula Rodriguez (3/23)

7. What have I completed since the last meeting?

a. Marcus has been working on a simple text entry, and finished his goals for the week.
b. Alyssa is finishing up the Calendar view meeting and is changing the frame size.
c. Paula has been working on the contact view meeting and is close to implementing it to the main frame.

8. **What am I aiming to complete before the next meeting**

a. Marcus' goal is to create recurring meetings.

b. Alyssa's goal is to finish up the calendar view and change the frame size.
c. Paula's goal is to finish sprint 3 goals (add windows to contact view).

9. What is currently blocking my progress?

a. We are experiencing issues with time.
b. There is some confusion on the code base but we are scheduling a meeting to discuss it and remove any confusion.

Scrum Notes Week 2: Marcus Schmidt (3/16):

10. What have I completed since the last meeting?

a. Alyssa successfully set up her VM and is halfway done with the code for her calendar view.
b. Paula created the window for contacts and is part way through tying it into the code base.
c. Marcus set up the meeting view window and established the code base.

11. What am I aiming to complete before the next meeting?

a. Alyssa will work on getting the calendar to draw in her window.
b. Paula will work on creating the contact view template within the window.
c. Marcus will work on being able to create a simple version of meetings.

12. What is currently blocking my progress?

a. Resolved difficulties getting VM set up, so that block is no longer a problem.
b. Still contending with midterms and just having the time to work on this project.

Entry by: Marcus Schmidt (3/14)

Git Commit: 70d6de85920ae08cae8d021e9521018c3284339f

I completed my task to set up a basic Meeting View window.

In order to do this, I wrote a tentative code base for the window to exist inside of. This is also one of our unassigned Sprint 2 goals (to tie all of our windows together). It did get more complicated than I expected, so we'll need to discuss whether or not we all like the direction it's moving in, but I pushed it to the GitHub repository for review and/or testing. I'd be very interested to hear if it compiles properly for you two. *fingers crossed*

A basic breakdown of the code is as follows:

- The class DailyHub (a wxApp subclass) is the central process that manages the windows (or, more precisely, wxFrames). It opens and closes frames as it is told to, and keeps track of them in a private vector.
- The class TempHomeFrame is just a temporary, empty window that is opened at the start of the program and serves as an example of a bare minimum window within this code base. It inherits from HubFrame (which inherits from wxFrame) so that the DailyHub class can interact with it in the ways it needs to.

- The class MV_Head is the beginning of the Meeting View. It's basically a copy of TempHomeFrame right now, but there are allowed to be duplicates of it and it's a little bit smaller.

I also wrote a tentative Makefile for the project, because even just compiling a few files by hand every time gets really annoying really fast. The Makefile is in the base directory, but running "make" grabs everything in the ./src directory, dumps the object files in a new ./obj directory, and creates the executable back in the base directory. "make clean" removes the object files but leaves the executable alone. "make clean_all" removes both the object files and the executable. Side note, GNU Make is actually really super cool.

SCRUM NOTES PT1: (3/9): Alyssa Diaz

- Talked about the installation of wxWidgets- need to decide what version to use.

13. **What have I completed since the last meeting:**
    Marcus built the wxWidgets library and pushed the repository structure into GitHub.
    Paula set up wxWidgets library.
    Alyssa is in the process of setting up wxWidgets as well.
14. **What am I aiming to complete before the next meeting**
    Marcus is working on to set up the meeting view.
    Paula is setting up a contact view.
    Alyssa is setting up a calendar view.
15. **What is currently blocking my progress?**
    Well we are in midterm szn, classes in general. Alyssa had a family emergency but she is going to get back on track.

    We can set up meetings to help finish the installation process with Alyssa on wxWidgets, and share the procedures on how it got done.