

## # Sprint 2 Report

CPT\_S-322: Group 3

Video Link: <https://youtu.be/C35Tq2Z8l2M>

### ## What's New (User Facing)

- \* Feature 1 or Bug Fix 1: Upon logging out the bio and username associated with the user profile is saved in the user csv file including methods to prevent duplicate profiles.

- \* Feature 2 or Bug Fix 2: Users are able to now create new profiles from the login screen and have their information saved to the database.

- \* Feature 3: User can now upload a picture of themselves to their profile, profile pic is uploaded to a free to use database for storing.

### ## Work Summary (Developer Facing)

Provide a one paragraph synopsis of what your team accomplished this sprint. Don't repeat the "What's New" list of features. Instead, help the instructor understand how you went about the work described there, any barriers you overcame, and any significant learnings for your team.

The work that was accomplished for this sprint included a multitude of functional requirements necessary for the RoommateFinder application. From the user's perspective, they can now create a new user profile, and have their information saved to our database (that being the csv files). They are also now able to open their profile page and include a picture of themselves to upload.

### ## Unfinished Work

If applicable, explain the work you did not finish in this sprint. For issues/user stories in the current sprint that have not been closed, (a) any progress toward completion of the issues has been clearly tracked (by checking the checkboxes of acceptance criteria), (b) a comment has been added to the issue to explain why the issue could not be completed (e.g., "we ran out of time" or "we did not anticipate it would be so much work"), and (c) the issue is added to a subsequent sprint, so that it can be addressed later.

The work not completed this sprint is making the user interface more attractive to the end user. This will be a function that will be continuously improved throughout the implementation to reduce the number of changes we have to make as we add in more features and have to adapt the UI to reflect those changes.

## ## Completed Issues/User Stories

Here are links to the issues that we completed in this sprint:

- \* [https://github.com/mattboombox/CPT\\_S-322-Group-Project/issues/27](https://github.com/mattboombox/CPT_S-322-Group-Project/issues/27)
- \* [https://github.com/mattboombox/CPT\\_S-322-Group-Project/issues/15](https://github.com/mattboombox/CPT_S-322-Group-Project/issues/15)
- \* [https://github.com/mattboombox/CPT\\_S-322-Group-Project/issues/4](https://github.com/mattboombox/CPT_S-322-Group-Project/issues/4)
- \* [https://github.com/mattboombox/CPT\\_S-322-Group-Project/issues/16](https://github.com/mattboombox/CPT_S-322-Group-Project/issues/16)

## ## Incomplete Issues/User Stories

Here are links to issues we worked on but did not complete in this sprint:

- \* [https://github.com/mattboombox/CPT\\_S-322-Group-Project/issues/14](https://github.com/mattboombox/CPT_S-322-Group-Project/issues/14) <<Constantly being updated, ongoing issue for cleaning UI>>
- \* [https://github.com/mattboombox/CPT\\_S-322-Group-Project/issues/22](https://github.com/mattboombox/CPT_S-322-Group-Project/issues/22) <<Integration problem with other currently working functional requirements>>

## ## Code Files for Review

Please review the following code files, which were actively developed during this sprint, for quality:

- \* CloudinarySettings.cs ([https://github.com/mattboombox/CPT\\_S-322-Group-Project/blob/main/RoomateFinder/RoomateFinder/CloudinarySettings.cs](https://github.com/mattboombox/CPT_S-322-Group-Project/blob/main/RoomateFinder/RoomateFinder/CloudinarySettings.cs))
- \* Service/IPhotoService.cs ([https://github.com/mattboombox/CPT\\_S-322-Group-Project/blob/main/RoomateFinder/RoomateFinder/Service/IPhotoService.cs](https://github.com/mattboombox/CPT_S-322-Group-Project/blob/main/RoomateFinder/RoomateFinder/Service/IPhotoService.cs))
- \* Service/PhotoServices.cs ([https://github.com/mattboombox/CPT\\_S-322-Group-Project/blob/main/RoomateFinder/RoomateFinder/Service/PhotoService.cs](https://github.com/mattboombox/CPT_S-322-Group-Project/blob/main/RoomateFinder/RoomateFinder/Service/PhotoService.cs))

## ## Retrospective Summary

Here's what went well:

1: For me personally, (Heseltine) my additions to the project went smoothly. I encountered no major errors and had a good learning experience. This is despite working with the C# language in a way that I was not previously familiar with.

2: For me (Brickner), my addition was pretty hard to implement. This required me understanding databases and how to operate with C# in a way that I was pretty

uncomfortable and intimidated by, requiring me to do a lot of research. But all in all, it was a rewarding experience

3. For me (Nelson), my additions to the project went smoothly for the most part. After enough debugging, my code was implemented without any errors or problems. I would have liked to contribute more during this sprint, but I ended up dealing with a bad illness for the second half. I am looking forward to contributing maximum time and effort in our next sprint to make up for it.

Here's what we'd like to improve:

### **1: Preventing overlapping**

One thing I'd like to improve is ensuring that we avoid overlapping in the issues we are working on. The roommate finder application we've chosen to develop involves storing a lot of information, such as usernames, passwords, results, and more. This has led to multiple classes accessing the same files, resulting in some crossover on certain issues. Consequently, we may be wasting productivity or potentially creating problems for the future.

I suspect that some of the code we've written so far may need to be modified or even entirely redone due to a lack of foresight. This could set us back and increase the time required to complete the project.

To address this, we could meet more frequently and communicate more specifically about what each team member is working on. Ideally, we would work side by side on issues that are highly interconnected, similar to an office environment. However, due to other commitments, it's more convenient for the four of us to work independently in our own time.

Another way to prevent overlapping is to ensure that issues that are highly related are assigned to the same individual. This approach would allow the programmer to fully understand the code they are working on and create a cohesive and consistent solution.

### **2: Enhancing team communication**

Building on the previous issue, I'd like to see our group communicate and meet more frequently. As mentioned earlier, working in an office-like environment would allow for quick, verbal communication, eliminating the delays caused by sending emails or waiting for class or other predetermined meeting times.

To replicate this environment, we could create a schedule for collective work sessions. For example, we could meet somewhere on campus with our laptops to work on our respective issues together. However, as we've discovered, our combined weekly schedules don't leave many opportunities for all of us to meet, given the variety of classes we're taking at different times.

To address this, we could compromise by conducting smaller meetings with two or three team members. While not ideal, this would still be an improvement over our current method of communication. That said, we're not currently suffering from a lack of communication, but a bit more coordination certainly wouldn't hurt.

### **3: Better identification of critical issues**

Lastly, I believe it would benefit our group to spend more time prioritizing the issues that are most critical to completing the project. This is a lesson learned in hindsight and is something we can aim to improve in future projects, as we are now nearing the final phases of this one.

In the previous sprint, time was spent working on issues that required other issues to be completed first, which caused some delays. In the future, it would be better to prioritize issues that have the greatest impact on enabling progress on other tasks. This wasn't as significant a problem this time around, as the number of remaining issues has dwindled, but it's still a valuable insight for future work.

This issue might have been avoided with a deeper understanding of how all the project components are interconnected. It may also have been influenced by the conflicting diagrams we created early on, as we produced several different types immediately after learning about them in class. If we were to restart the project, the experience we've gained would likely help us avoid this issue altogether.

This may have been caused by the creation of conflicting diagrams since a multitude of them of different varieties were created immediately after learning about them in class. If we were to restart the project what we've learned so far would likely prevent this issue.

Here are changes we plan to implement in the next sprint:

#### **1: Meet for overview of project**

As we near the final stages of the project, it's crucial to clearly identify the remaining tasks. Achieving this will require us to meet more frequently and for extended periods as well as

properly distribute the fewer task we have left.

## **2: Begin significant testing**

Although the project is not yet complete, it will be essential for us in the next sprint to review previous additions to the code to ensure they function properly. This will allow us to move forward confidently without needing to revisit earlier sections.