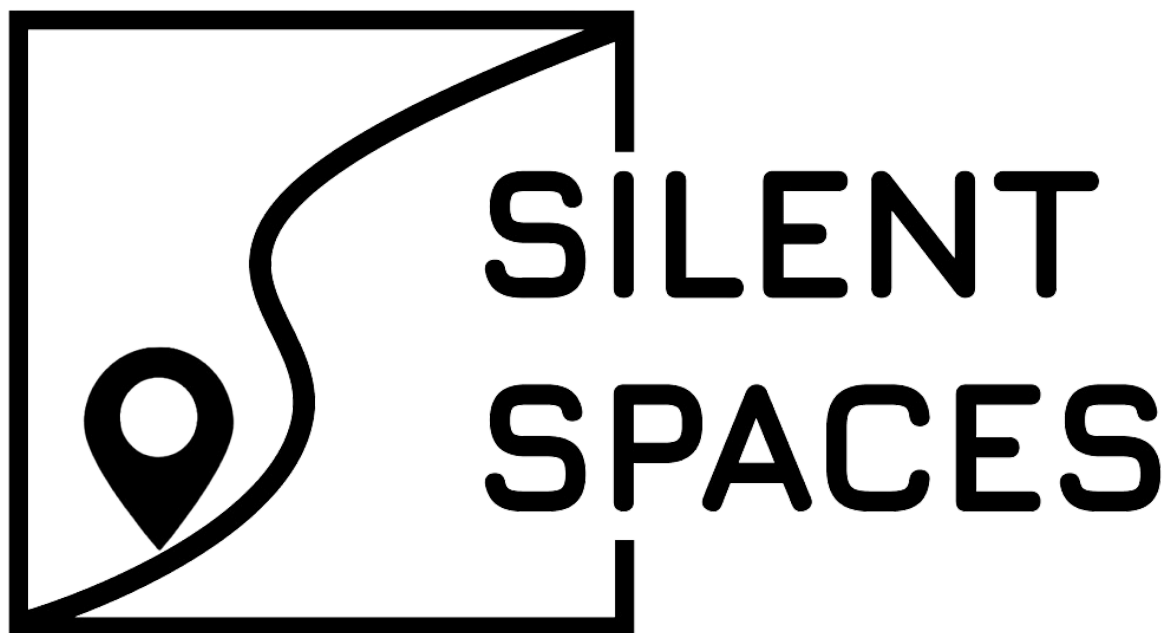


Coding Project Final Report



Silent Spaces Locator

Prepared by
Esat Duman, Jonathan Juarez, Kent Lizardo, Jose Tejada
for use in CS 440
at the
University of Illinois Chicago

Spring 2024

Table of Contents

	List of Figures	4
	List of Tables	5
I	Project Description.....	6
1	Project Overview.....	6
2	Project Domain.....	6
3	Relationship to Other Documents.....	6
4	Naming Conventions and Definitions.....	6
4a	Definitions of Key Terms	7
4b	UML and Other Notation Used in This Document	8
4c	Data Dictionary for Any Included Models	9
II	Project Deliverables.....	11
5	First Release.....	11
6	Second Release.....	13
7	Comparison with Original Project Design Document.....	16
III	Testing.....	16
8	Items to be Tested.....	16
9	Test Specifications.....	17
10	Test Results.....	26
11	Regression Testing.....	28
IV	Inspection.....	29
12	Items to be Inspected.....	29
13	Inspection Procedures.....	29
14	Inspection Results.....	29
V	Recommendations and Conclusions.....	30
VI	Project Issues.....	30
15	Open Issues.....	30

16	Waiting Room.....	31
17	Ideas for Solutions.....	32
18	Project Retrospective.....	32
VII	Glossary.....	33
VIII	References / Bibliography.....	34
IX	Index.....	34

List of Figures

Figure #1 : Sequence Diagram: Creating and Updating a Space	8
Figure #2 : Sequence Diagram: Opening Space View	9
Figure #3 - Initial Flowchart Diagram	13
Figure #4 : Original Home Page	13
Figure #5 : Original Dashboard Header	13
Figure #6 : Updated Home Screen	15
Figure #7 : Initial Google Maps Embed Implementation	16
Figure #8 : Space Creation / Editing Page	18
Figure #9 : Login / Account Creation Page	19
Figure #10 : Dashboard: SpaceList and Map Embed	21

List of Tables

Table #1 : Data Dictionary Table

10

I Project Description

1 Project Overview

Silent Spaces locator aims to aid students and every-day workers in finding the perfect location to work or study through a real-time network and user interface. The objective of Silent Spaces is to gather GPS, time, and self-surveyed user data from smartphones to create a real-time network flow of certain locations all around Chicago, specifically quiet work spaces. Based on the GPS and time information, we can gather data and create a Headcount Network of users in a certain location and provide an estimate to the user on how crowded a location is. Additionally, through user surveyed data we can declare a space's Ambience Level in a certain location so that the user can understand the noise level at a location and see if it suits their particular needs as a work space before arriving. Using all of this data gathered, we can construct a Space Ambience Rating for each location to display to users looking for a space to work or study.

2 Project Domain

The domain of this project revolves around public space management. In unknown locations or densely populated areas like university campuses, they are finding quiet places for studying, reading, participating in meetings, and for a place to relax. "Silent Spaces" addresses the growing need for individuals who want a quiet place but can't locate them or reserve spaces for themselves. "Silent Spaces" solves the problem for users who need a peaceful place to get things done and find locations efficiently and fast rather than searching for a place themselves, which depending on time of day can take an extensive amount of time. While places like libraries, common areas, study rooms, and reading spaces are available for most people, they are very densely populated during the day, making it hard to find a quiet place that meets users' needs, like good wifi locations and low sound levels. The project's value lies in the ability to be able to provide users with a seamless experience in a comprehensive directory of silent spaces.

3 Relationship to Other Documents

This document and project stems from the ideology of CS 440 Group 9 from the Fall 2022 semester. The associated report credit is below in the bibliography section.

All features besides the google maps integration were directly sourced from the original report document. The original UI mockups included in the report was the starting point for our applications layout and style as well.

4 Naming Conventions and Definitions

Silent Spaces is designed to enhance your experience in navigating and accessing various spaces across campus. To help you better understand the functionalities and features available, let's introduce some key terminology:

- **Space**
- **SpaceCards**
- **SpaceView**
- **SpaceList**
- **Organization**
- **Dashboard**
- **Find a Space**
- **Create a Space**
- **Check-in**
- **Check-out**
- **Favorites**
- **Location Data**

4a Definitions of Key Terms

Space: A physical area within a building or location designated for work or study purposes.

SpaceCards: Visual representations (cards) that provide summarized information about a particular space, located in the SpaceList and often used for browsing or selection purposes.

SpaceView: A detailed view or interface presenting comprehensive information about a specific space, including its location, amenities, and current status.

SpaceList: A catalog or list displaying available spaces, typically categorized and sortable by various criteria such as location, features, and ratings.

Organization: An entity such as a university, library, or company, within which spaces are managed and accessed by users. Organizations are essentially the owners of a registered space.

Dashboard: A centralized interface providing quick access to various features and functionalities of the app, typically customizable to suit user preferences. Home of the app's key functionality: Create/Find a Space, SpaceList, SpaceViews, SpaceList, etc.

Find a Space: A feature allowing users to search, browse, and select from available spaces based on location, and space amenities.

Create a Space: This functionality empowers you to establish a new area within your organization, whether it's a study room, conference area, or communal space, tailored to meet the needs of your users. Various tags are available when creating a space, which are displayed on the SpaceCard to signal what amenities are available in said space.

Check-in: The action taken by a user to indicate their presence or occupancy within a specific space, typically recorded for organizational or informational purposes.

Check-out: The action taken by a user to indicate their departure or no longer being present within a specific space.

Favorites: Spaces that a user has marked or saved for quick and easy access, often based on personal preferences or frequent usage.

Location Data: Information about the geographical position of a user, typically obtained from GPS or other positioning technologies.

4b UML and Other Notation Used in This Document

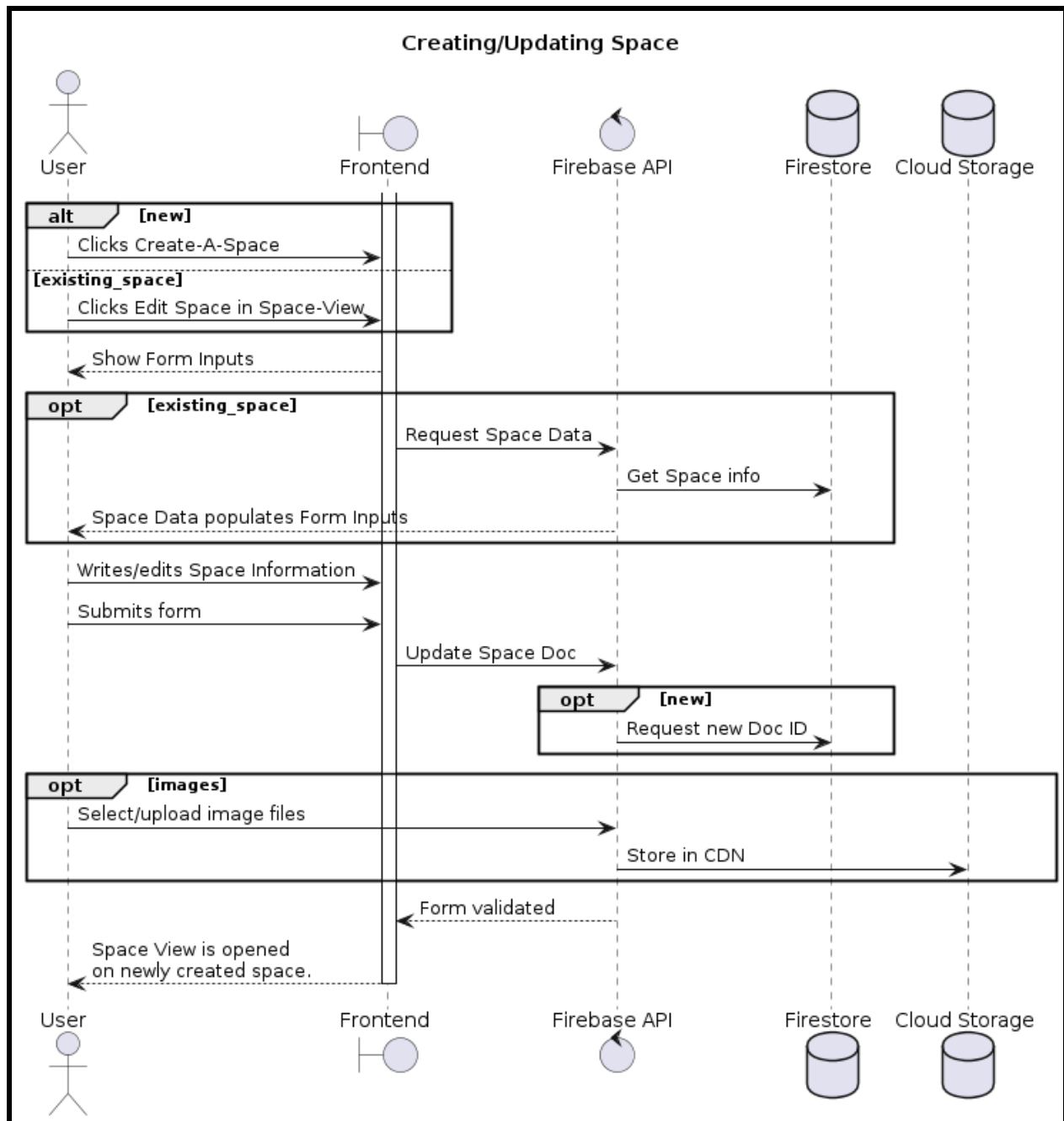


Figure #1 : Sequence Diagram: Creating and Updating a Space

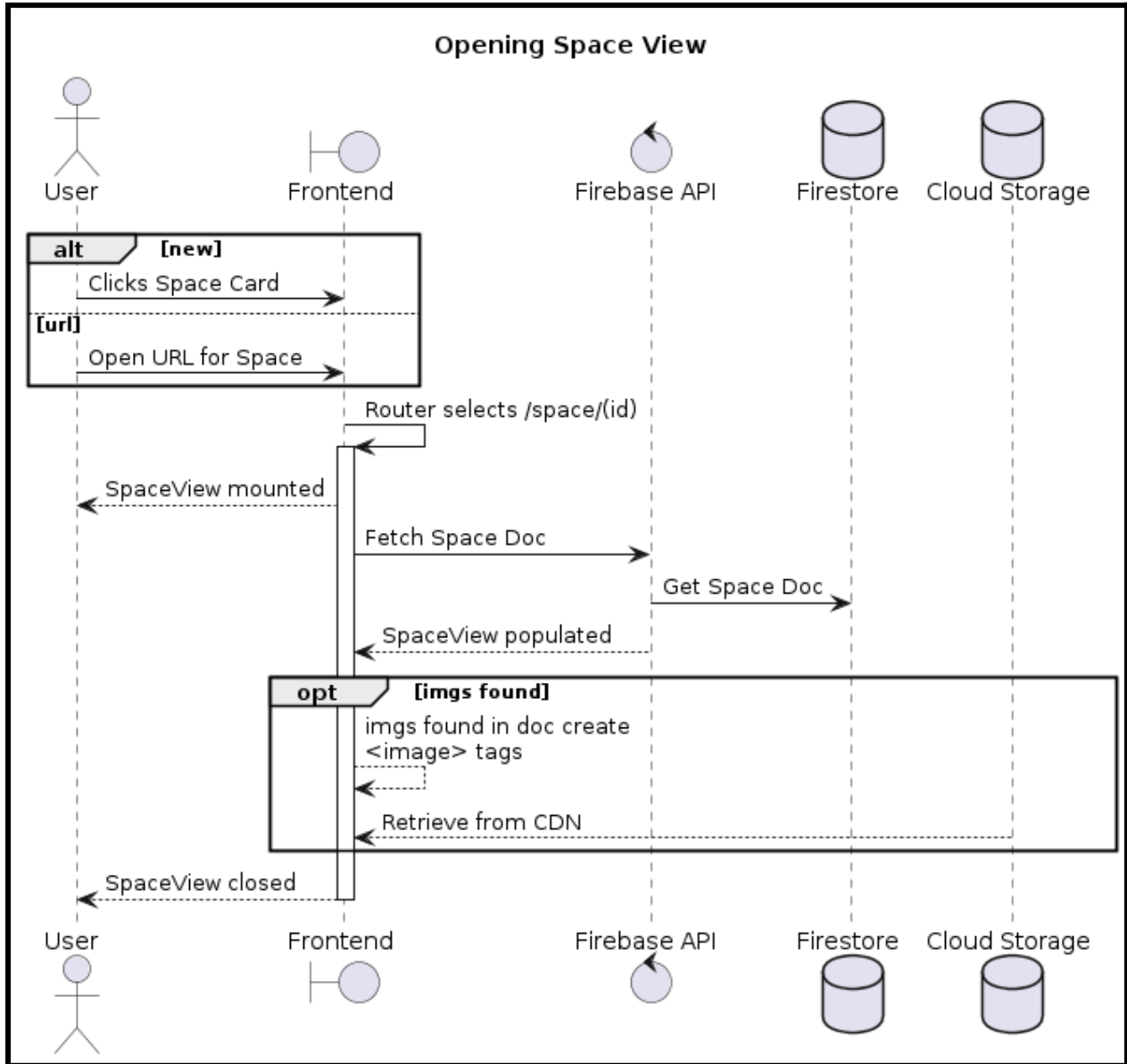


Figure #2 : Sequence Diagram: Opening Space View

4c Data Dictionary for Any Included Models

Space Document at db/spaces/{id}

Key	Type	Value Definition	Example
{id}	String	Document ID	“6DQkyeSSOWI1TrWlsKjv”
geohash	String	Geohash created when location is set; used for	“dp3wjtbbch”

		query optimization	
lat, lng	float/double	Latitude and longitude coordinates referring to space	41.87204804661626, -87.64749748087038
displayName	string	Human readable name	“Student Center East”
name	String	Post identifier	“sce”
desc	String	A lengthy description of the space that can be seen in Space View.	“Lounge located on the second floor of Student Center East.”
owner	String	UID of the user who created this space	“SFVVuAmtGjMxvjrTxjUziBt fp7Z2”
imgs	Array[String]	Array of image sources that will be converted into <image> tags. Typically a Firebase Cloud Storage link	[URL removed for brevity]
tags	Array[String]	A list of tags that the space refers to	[“quiet”, “reserved”, “wheelchair_acc”]

Username Link at db/usernames/{username}

Key	Type	Value Definition	Example
{username}	String	Document ID, in this case the unique username	dev
uid	String	User UID used for double relation between user data.	“SFVVuAmtGjMxvjrTxjUziBt fp7Z2”

User data at db/users/{uid}

Key	Type	Value Definition	Example
{uid}	String	Document ID, in this case the user’s uid.	dev
spaces	Array[Reference]	List of spaces this user has created, referring to the document id	/spaces/crmZSiJZoH98aCbRS VFQ

Table #1 : Data Dictionary Table

II Project Deliverables

The course of our project production spanned 12 weeks throughout the semester. The work was split into three individual releases, each one spanning four weeks, and each week acting as its own individual sprint. The first two releases were accompanied by a scenario that we as a team drafted up to set a benchmark for what the project state should be by time of that release. At the end of the first two releases the project was demoed privately for the course's teaching assistant, in which we were to showcase the functionality of our project at that point in time, receiving feedback and seeing how it matches up to our proposed scenario. The third and final release was the cumulative work throughout the entire semester which was demonstrated to the entire class on April 22nd, 2024.

1 First Release

The date of Silent Spaces first project release was February 23rd, 2024, and the expected functionality by time of first release was as described in our first scenario:

Scenario "First Time Use"

The user upon launching the app was met with a login portal prompting users to either login to an existing account, or create an account. In order to create an account, the user must be tied to an organization, which can be done with an invitation link/code provided by the organization. Once logged in, the user will be asked to provide their location data to populate a list of local registered spaces. The user can be involved in multiple organizations, but requires creating a profile for each one. If the user is registered with an organization (i.e University, Library, etc.), then the relevant spaces are then automatically displayed.

Once the user has properly logged on they are met with a "Dashboard" interface. This "Dashboard" page links to the following pages. "*Find a Space*", "*Profile*", "*Settings*", or "*Support*".

The "*Find a Space*" page lets you choose from 3 options. Either into "*Spaces Near You*", "*Favorite Spaces*", or "*All Spaces*."

Spaces Near You (*Not fully implemented by first release*)

Uses the location data to find out which spaces are closest to the user that they have access to. It will display it in a linear order, sorted by an algorithm taking into account distance, and the space's rating. The user can then enter a Space Overview for a certain space.

Favorite Spaces (*Not fully implemented by first release*)

A list of the spaces the user has labeled one of their "Favorite Spaces." They enter the Space Overview from here.

All Spaces *(Implemented by first release)*

Provides a catalog of all spaces owned by that organization. It also has the option to filter by their current location in addition to listing in that specific order just like “Spaces Near You”. This page also has a basic search, and an advanced search with filters. The user can also enter a *Space Overview* using this page.

Profile *(Chose to not to implement)*

A profile/account menu will also be available for the user to edit their personal information that will be visible to other users in the organization. They can change their public display name, username/email, and various other details. Any changes to this will send a request to the organization’s hosting server.

Settings *(Chose to not to implement)*

A settings menu will also be available for editing the app preferences. This includes things such as lightmode/darkmode. This will only be stored on the user’s device.

Space Overview *(Not fully implemented by first release)*

Once a user has selected their preferred space, they are met with general information pertaining to that space such as the specific location (building, floor, etc.), a noise level indicator, and current power outlet availability depending on what features the organization’s server hosts. seat or outlet availability etc.

The user can “check in” to a space, notifying the server that the user is currently “residing” in it. They can also choose not to, if needed.

If the user decides to check out a space they are given the option to review the space based on their experience, and also give an update on the noise levels or occupant status. If the user enjoyed their experience, they are given the option to add the space to a favorites list for easier accessibility. The users are also given the option of reporting or adding their own data to the space, such as “This space has more people than suggested!”, or “This space has faulty outlets.”

Support *(Chose to not to implement)*

The user can use the support page to request help, and send reports detailing any contradictory or erroneous information regarding a page, or the system. Once the user presses the support button they will be able to submit a form which will be seen on an google excel sheet for the developers.

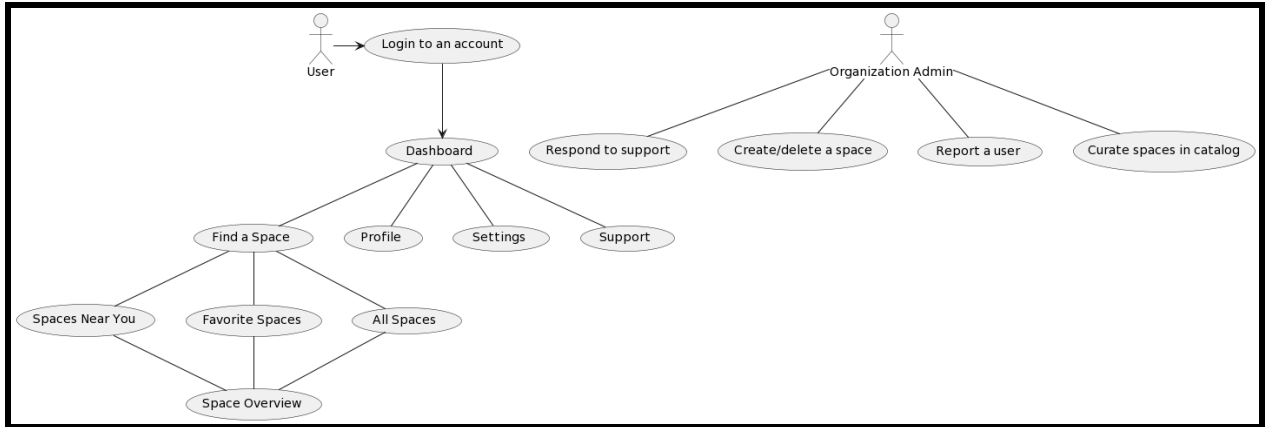


Figure #3 - Initial Flowchart Diagram

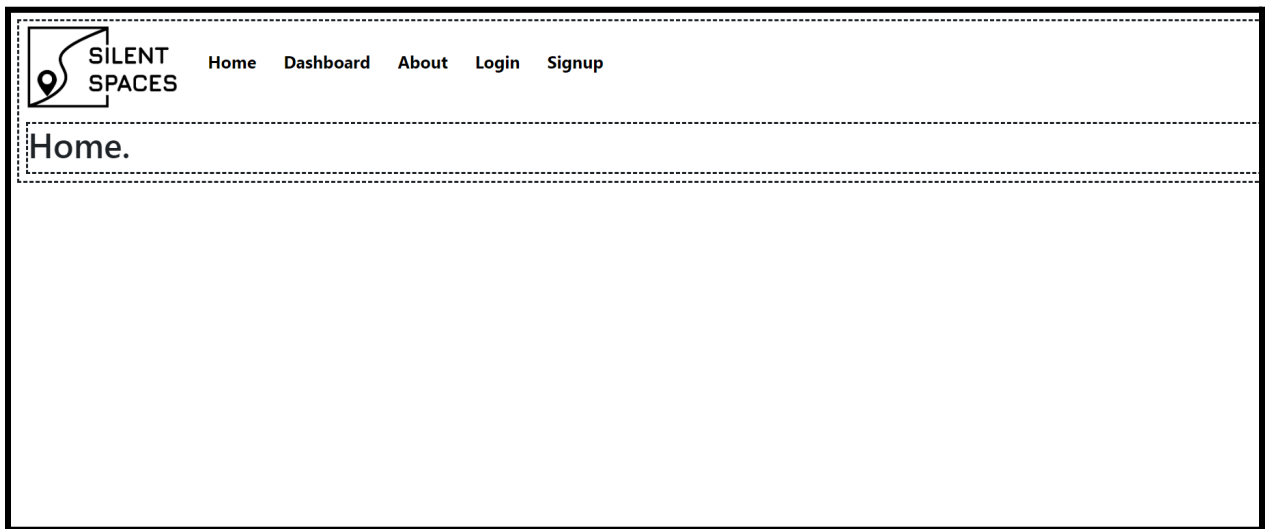


Figure #4 : Original Home Page

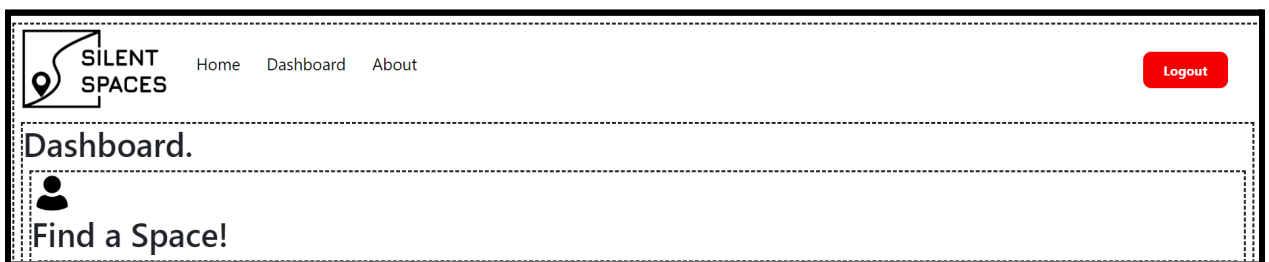


Figure #5 : Original Dashboard Header

2 Second Release

The date of Silent Spaces second project release was March 29th, 2024, and the expected functionality by time of first release was as described in our second scenario:

Scenario “Enhanced Functionality: Silent Spaces 2.0”

We described the initial user experience of the **Silent Spaces Locator** in our previous scenario, which detailed the user account creation process, finding a space, among other things. For this release we expect to expand upon our work and make the app more streamlined and feature rich.

Starting off, the login process has been adjusted a bit, where the user is met with an individual login/account creation screen upon entering the site for the first time, also requiring a user to be logged into to create/access the spaces.

Updated UI and Styling (*Implemented by Second Release*)

Our first scenario mainly focused on the functionality of the app, getting the backend setup, making sure the included features were working, etc. For this second release users will notice a cleaner and sleek aesthetic as they browse the pages of Silent Spaces. Updating the UI will not only help aesthetically, but will make navigating the app much more intuitive and enjoyable.

Users will quickly notice the new and improved *Space Cards* (more information below), which will include images to provide users with a glimpse of the atmosphere and amenities offered at a space before clicking to view more details. Furthermore, we are introducing additional information details within each Space card such as the space's, **overall rating, power outlet availability, and feedback on how crowded a space is currently**, all in efforts to enable the users to assess the suitability of a space more efficiently.

To enhance user convenience, user's will also now be able to search for a specific space by name or keyword, eliminating the need to scroll through the entire list of nearby spaces. Overall in effort to save a user time and effort when trying to locate a space.

Additionally, during the *Create a Space* process users will also have access to a Google Maps embed directly in the app (more information below). This integration will allow users to explore nearby spaces themselves, allowing them to make more informed decisions about where to work given their commute, location, etc.

New Features: Space Cards and Views (*Implemented by second release*)

Upon logging in, users are presented with a range of options to navigate through such as finding a space, creating a space, and adding a space to their favorites. As previously mentioned, spaces will now have “*space cards*” that quickly showcase relevant information to that space, that when clicked on, will enter the space-view.

Now, alongside the space's name and location, users can quickly assess a space's suitability by checking the space's overall rating, power outlet and printer availability. Additionally, Users will have the ability to check-in to the space directly from the Space View which facilitates real time updates to a space's occupancy levels.

Overall, these updates offer users enhanced visibility into each space at a quick glance, significantly improving the process of finding their ideal workspace.

Google Maps Integration (*Implemented by Second Release*)

During the *Create a Space* process the app will retrieve a user's location using location services upon obtaining permission. A user can also manually locate themselves using the Google Maps embed. When this is done, the geopoint is added to the Space document's location field. If a Space is missing a location, or there is incorrect data, some users might be able to recommend certain locations that can be used in lieu of data placed by the owner.

Find a Space can then create a query on the *spaces* collection table in our database, using a geopoint distance query. This query might also have different options depending on the search filter used.

Integrating Google Maps as a service allows users to visually explore nearby locations, providing more context and physical orientation. This allows users to make more informed decisions about their workspace based on many factors such as proximity to other locations or transportation hubs. Ultimately, this adds another layer of engagement for the user experience, along with the app's updated functionality.



Figure #6 : Updated Home Screen

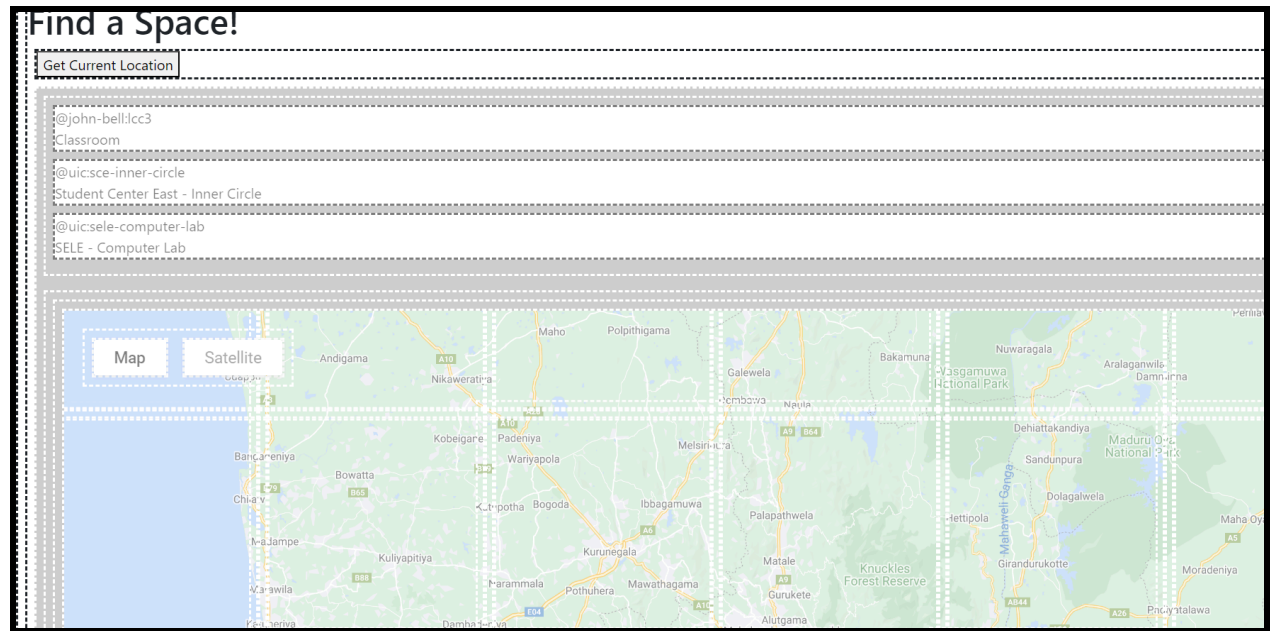


Figure #7 : Initial Google Maps Embed Implementation

3 Comparison with Original Project Design Document

While our implementation draws heavily from the original concept developed by Group 9, we made the strategic decision to omit certain features that were either technically challenging or deemed less critical for our prototype. For example, the real-time microphone ambience level detection feature, while innovative, presented significant technical hurdles and privacy considerations that we opted not to tackle in our initial iteration of Silent Spaces Locator. Instead, we prioritized features that would provide immediate value to users, such as space creation, image uploading, and location-based search functionalities.

We extensively drew upon the original report's key components, particularly the overall description, project requirements, and UI mockups provided by Group 9. While we didn't directly adopt their content, we utilized it as a guiding framework for our development process across the three releases. This approach ensured that our implementation stayed aligned with the original vision while allowing us to tailor the solution to our specific needs and constraints.

III Testing

1 Items to be Tested

- **Creating a Space**
- **Location Services**
- **Login / Account Creation**
- **Image Upload while Creating Space**

- **Google Maps Embed Pins**
- **SpaceList / Nearby Spaces**
- **Editing an Existing Space**

2 Test Specifications

ID# CS-001 - Create a Space

Description: This test verifies the ability of the user to successfully create a new space within the app.

Items covered by this test:

- Creation of a new space
- Setting a Space's name, location, amenities, and availability

Requirements addressed by this test:

- **#03 - Load Space Information** (Group 9: Development Project Final Report)

Environmental needs:

- Access to the app with appropriate permissions to create spaces.

Inter-case Dependencies: NA

Test Procedures:

1. Launch the app and navigate to the "Create a Space" section.
2. Enter the necessary details for the new space, including name, location, amenities, and availability.
3. Submit the form to create the space.
4. Verify that the space is successfully created and visible within the app.

Input Specification:

- **Space ID:** daleylibrary
- **Display Name:** "Idea Commons"
- **Description:** "Located on the first floor of UIC's Richard J. Daley Library"
- **Location:** First Floor Richard J. Daley Library
 - Latitude: 41.87216221012142
 - Longitude: -87.65073117364635
- **Tags:** Charging, Scheduled, Printers

Output Specifications:

- **Confirmation message:** "Space created successfully"

- Newly created space displayed in the list of available spaces within the app.

Pass/Fail Criteria:

- **Pass:** The space is successfully created and displayed within the app.
- **Fail:** The space creation process fails or the newly created space is not visible within the app.

Editing a Space

Space ID:

Display Name:

Description:

Location: Latitude: 41.87216221012142, Longitude: -87.65073117364635

Images:

Figure #8 : Space Creation / Editing Page

ID# CS-002 - Login / Account Creation

Description: This test verifies the ability of the user to successfully login within the app.

Items covered by this test:

- **User login**

Requirements addressed by this test:

- #01 - Provide Login Information

Environmental needs:

- Access to the app with user login already created.

Intercase Dependencies: N/A

Test Procedures:

1. Launch the app and navigate to login screen
2. Provide user email and password
3. Press login button
4. Verify users are logged in by going to the “Dashboard” section and seeing a list of spaces viewable.

Input Specification:

For Testing Purposes:

- **Email:** testing@google.com
- **Password:** test123

Output Specifications:

- User is redirected to Dashboard page from login page

Pass/Fail Criteria:

- **Pass:** The user is redirected to the Dashboard page.
- **Fail:** Email/password provided is incorrect or not within the database and the user remains on the login screen.

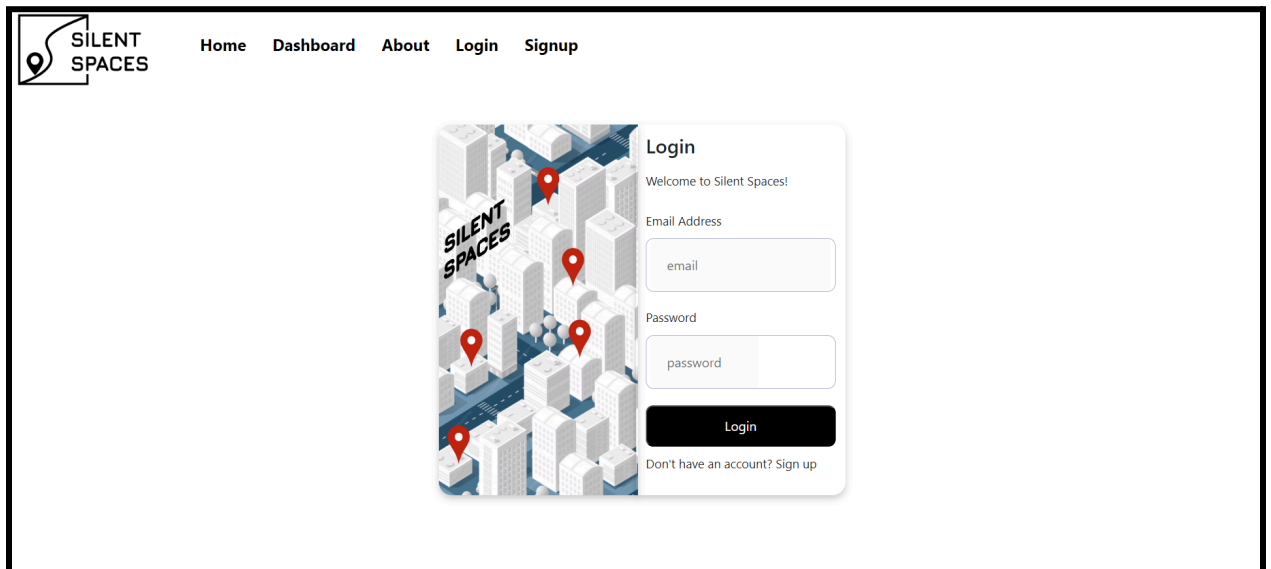


Figure #9 : Login / Account Creation Page

ID# CS-003 - Nearby Spaces

Description: This test verifies the ability of the user to view nearby spaces

Items covered by this test:

- **Nearby Spaces viewability in Dashboard**

Requirements addressed by this test:

- #02 - Load a nearby spaces list

Environmental needs:

- Access to the app with the user already logged in.

Intercase Dependencies: N/A**Test Procedures:**

1. Launch the app and navigate to login from navigation menu
2. Login with user's email and password
3. Navigate to the Dashboard section from navigation menu
4. Verify that the Nearby Spaces list is populated with spaces

Input Specification:

For Testing Purposes:

- **Email:** testing@google.com
- **Password:** test123

Output Specifications:

- User is able to view list of spaces from Nearby Spaces list in Dashboard

Pass/Fail Criteria:

- **Pass:** The user is able to view a list of spaces in the Dashboard.
- **Fail:** The user is not able to view the list of spaces in the Dashboard.

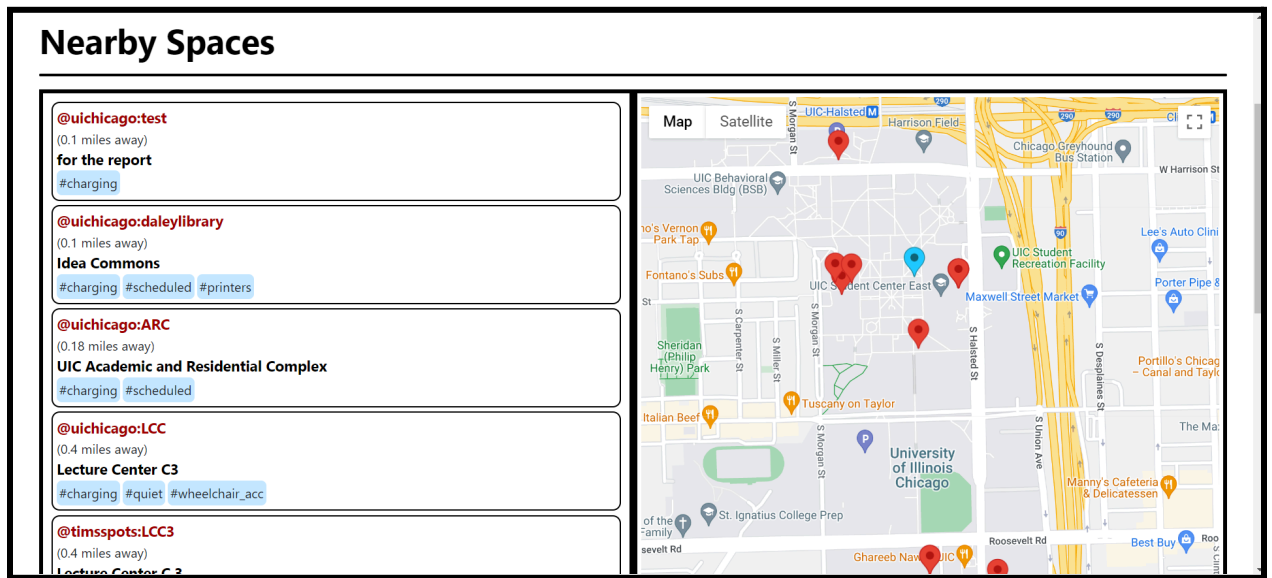


Figure #10 : Dashboard: SpaceList and Map Embed

ID# CS-004 - Space Descriptions

Description: This test verifies the ability of the user to view the selected space's information.

Items covered by this test:

- Selected space information viewable by the user

Requirements addressed by this test:

- #18 - Spaces Information

Environmental needs:

- Access to the app with the user already logged in.

Intercase Dependencies: N/A

Test Procedures:

1. Launch the app and navigate to login from navigation menu
2. Login with user's email and password
3. Navigate to the Dashboard section from navigation menu
4. Click on space with space id of @uichicago:daleylibrary
5. Verify that the user is redirected to @uichicago:daleylibrary's information page with the space's tags, picture, location, and description.

ID# CS-005 - Location Services

Description: This test verifies the functionality of the app's location services, ensuring accurate retrieval and utilization of user location data.

Items covered by this test:

- Retrieval of user location data (long/lat)
- Display of spaces based on user's current location
- Accuracy of distance calculations for nearby spaces

Requirements addressed by this test:

- #1 - Location Services

Environmental needs:

- Access to a device with GPS capabilities
- Permission granted for the app to access device location

Intercase Dependencies: N/A

Test Procedures:

1. Ensure that location services are enabled on the device.
2. Launch the app and navigate to the dashboard.
3. Allow the app to access the device's location data if prompted.
4. Verify that the app accurately retrieves the user's current location.
5. Check that the list of spaces displayed is relevant to the user's current location.
6. Select a nearby space and verify that its distance from the user's current location is accurately calculated.

Input Specification:

- **Device location:** Simulated location coordinates within a known area.
- **Nearby spaces:** A set of predefined spaces within the vicinity of the simulated location.

Output Specifications:

- User's current location displayed accurately on the app.
- List of nearby spaces displayed based on the user's current location.
- Distance calculation for nearby spaces matches the actual distance from the user's location.

Pass/Fail Criteria:

- **Pass:** The app accurately retrieves user location, displays nearby spaces, and calculates distances within an acceptable margin of error.
- **Fail:** The app fails to retrieve user location, displays irrelevant spaces, or calculates distances inaccurately.

ID# CS-006 - Google Map Pins Accuracy

Description: This test ensures that the Google Map pins displayed on the embedded map within the app accurately represent the locations of the corresponding spaces.

Items covered by this test:

- Placement of Google Map pins on the embedded map
- Accuracy of pin locations compared to the actual coordinates of the spaces

Requirements addressed by this test:

- #2 - Accurate Location Detection

Environmental needs:

- Access to the app with the embedded Google Map feature enabled
- Internet connectivity for map data retrieval

Intercase Dependencies: N/A

Test Procedures:

1. Login to the app and navigate to the dashboard.
2. Compare the location of the Google Map pin with the known coordinates of the space.
3. Zoom in and out of the map to ensure that the pin remains accurately positioned.
4. Repeat the process for multiple spaces..

Input Specification:

- Known coordinates of the spaces with embedded maps.
- Actual visual inspection of the pin's location on the map.

Output Specifications:

- Google Map pin accurately positioned at the known coordinates of the space.
- Consistent placement of pins across different spaces with embedded maps.

Pass/Fail Criteria:

- **Pass:** The Google Map pins are accurately positioned on the embedded map and remain consistent across different spaces.
- **Fail:** The pins are inaccurately placed or inconsistent across spaces, deviating significantly from the known coordinates of the spaces.

ID# CS-007 - Space Image Uploading

Description: This test verifies the ability of users to upload images for a space within the app.

Items covered by this test:

- Uploading images for a space
- Ensuring images are correctly associated with the designated space

Requirements addressed by this test:

- #10 - Updating a Space

Environmental needs:

- Access to the app with appropriate permissions to upload images.
- A device with a camera or access to image files for upload.

Intercase Dependencies: N/A

Test Procedures:

1. Navigate to the SpaceCard of a space within the app and interact with it.
2. Enter the edit menu and locate the option to upload images for the space.
3. Select the upload option and choose an image from the device's gallery or capture a new photo using the device's camera.
4. Confirm the upload of the selected image.
5. Verify that the uploaded image is displayed correctly within the space's details page.
6. Repeat the process in other spaces if needed.

Input Specification:

- **Image files:** Sample images selected for upload, including JPEG, PNG, or other supported formats.
- **Device camera:** Capture new images to test the upload functionality.

Output Specifications:

- Uploaded images are correctly associated with the designated space.
- Images are displayed prominently and clearly within the space's details page.

Pass/Fail Criteria:

- **Pass:** The Google Map pins are accurately positioned on the embedded map and remain consistent across different spaces.
- **Fail:** The pins are inaccurately placed or inconsistent across spaces, deviating significantly from the known coordinates of the spaces.

ID# CS-008 - Editing a Space

Description: This test verifies the functionality of editing existing space details within the app.

Items covered by this test:

- Editing various attributes of a space, such as name, location, amenities, and availability.
- Ensuring that changes made to a space are accurately reflected and updated within the app.

Requirements addressed by this test:

- #10 - Updating a Space

Environmental needs:

- Access to the app with appropriate permissions to edit spaces.
- A space with existing details that need to be modified for testing purposes.

Intercase Dependencies: N/A**Test Procedures:**

1. Navigate to the dashboard within the app, select a space that needs to be edited.
2. Click on the SpaceCard and locate the option to edit the space details.
3. Modify various attributes of the space, such as name, location, amenities, or availability, as needed.
4. Confirm the changes made to the space.
5. Verify that the edited space details are accurately reflected and updated within the app.
6. Optionally, revert the changes made to the space to its original state for further testing.

Input Specification:

- **Modified space details:** New values or updates for attributes such as space name, location, amenities, or availability.
- **Device interface:** Interaction with app elements for editing space details.

Output Specifications:

- Edited space details are accurately reflected and updated within the app.
- Confirmation message or indication of successful space edit operation.

Pass/Fail Criteria:

- **Pass:** Changes made to the space details are accurately reflected and updated within the app, with confirmation of successful edit operation.
- **Fail:** Edited space details fail to update or are inaccurately reflected within the app, leading to inconsistencies or errors in space information.

3 Test Results

ID# - CS001-Test-Create a Space

Date(s) of Execution: 4/26/24

Staff conducting tests: Kent Lizardo

Expected Results: Upon space card creation, the user is shown a “Space Created Successfully” confirmation message. User is shown Created space is reflected within the space list in Dashboard

Actual Results: Identical to expected results

Test Status: Pass

ID# - CS002-Test-Login

Date(s) of Execution: 4/26/24

Staff conducting tests: Jose Tejeda

Expected Results: User is redirected from login page to Dashboard page. Spaces are now viewable to the user with login features available. User should be able to see options in navigation menu “Find a space”, “Create a Space”, “Logout”

Actual Results: Identical to expected results

Test Status: Pass

ID# - CS003-Test-Nearby Spaces

Date(s) of Execution: 4/26/24

Staff conducting tests: Esat Duman

Expected Results: Dashboard page should show nearby spaces available to the user with the space list. Spaces should appear within their respective location inside the google maps api.

Actual Results: Identical to expected results

Test Status: Pass

ID# - CS004-Test-Space Descriptions

Date(s) of Execution: 4/26/24

Staff conducting tests: Jonathan Juarez

Expected Results: Selected space should redirect user to the respective space's information page. Users should be able to look at the space's display name, id, image, location and tags.

Actual Results: Identical to expected results

Test Status: Pass

ID# CS-005 - Location Services

Date(s) of Execution: 4/27/24

Staff conducting tests: Jonathan Juarez

Expected Results: Location should be accurately retrieved when creating a space after selecting the get location option. Once a space is created with a specific location a pin should appear on the map, along with showing how far a space is from the user (ex: 0.23 miles).

Actual Results: Identical to expected results.

Test Status: Pass, everything performed as expected

ID# CS-006 - Google Map Pins Accuracy

Date(s) of Execution: 4/27/24

Staff conducting tests: Kent Lizardo

Expected Results: Google Map pins accurately positioned on the embedded map, matching the known coordinates of the corresponding spaces.

Actual Results: Identical to expected results.

Test Status: Pass, pins work as expected.

ID# CS-007 - Space Image Uploading

Date(s) of Execution: 4/27/24

Staff conducting tests: Esat Duman

Expected Results: Uploaded images should be correctly associated with the designated space and displayed prominently within the SpaceView.

Actual Results: Identical to expected results.

Test Status: Pass, Images uploaded successfully.

ID# CS-008 - Editing a Space

Date(s) of Execution: 4/27/24

Staff conducting tests: Jose Tejeda

Expected Results: Changes made to the space details should be accurately reflected and updated within the app, with confirmation of successful edit operation.

Actual Results: Identical to expected results.

Test Status: Pass, Spaces fields are updated successfully after editing space.

4 Regression Testing

- **ID#-CS001 - Create a Space:**
 - Potentially adding new fields or validation checks, do not affect the ability to create spaces successfully.
- **ID#-CS002 - Login**
 - Potentially add multi-factor authentication or integrating with third-party authentication providers, do not disrupt the login functionality.
- **ID#-CS003 - Nearby Spaces**
 - Potentially optimize distance calculations or adding new filtering options, do not affect the accuracy of the nearby spaces list.
- **ID#-CS004 - Space Description**
 - Potentially allow rich text formatting or supporting multimedia content, do not affect the readability or presentation of space descriptions.
- **ID# CS-005 - Location Services**
 - Potentially improve accuracy, do not degrade the overall functionality of the location-aware features.

- **ID# CS-006 - Google Map Pins Accuracy**
 - Potentially adding new map layers or customizing marker styles, do not affect the accuracy of the Google Map pins.
- **ID# CS-007 - Space Image Uploading**
 - Potentially increasing the number of images associated with a space such as increasing file size limits or supporting additional image formats, do not introduce any new issues.
- **ID# CS-008 - Editing a Space**
 - Potentially adding more fields or tags during space creation, or adding new validation rules or improving error handling, do not disrupt the ability to edit spaces accurately.

IV Inspection

1 Items to be Inspected

Security rules: Located at *firestore.rules* and *storage.rules* must be updated for post-development. They are currently in development mode, in which all changes to databases are permitted.

Naming Scheme for useGeoLocation: evaluate the clarity of the naming for Geolocation used in the dashboard.

Placement of the logout button: The logout button for the user was placed incorrectly on the header nav bar. Instead it should be moved to the dashboard nav bar.

2 Inspection Procedures

Check Code: : utilizing githubs commit tracking to see the most recent changes upon the codebase. Making what's not right or something that can lead to bigger problems down the road and addressing it.

Holding Meetings: Discuss the issues and find solutions to the problem as a team so everyone is on the same page.

Communication: Discussing inspection results in-person or virtually (Zoom) making sure the team was informed of the inspection.

3 Inspection Results

Inspection 1:

- **Inspector:** [Kent Lizardo]
- **Date & Time:** [3/6/2024, 8:32pm]
- **Findings:** Naming scheme for useGeoLocation is confusing. The hooks named are the following. **useGeoLocation**: a module used to utilize the browser's location support. **useLocation()**: a function used to retrieve the location. **setLocation()**: a React state modifier. **location**: React state that returns the current location. I would most likely rename the retrieving function to retrieveLocation() or promptLocation() for clarity. Since **getLocation()** and **setLocation()** sound like they are closely related as if they were a setter/getter, but they are actually not in direct relation.

Inspection 2:

- **Inspector:** [Esat Duman]
- **Date & Time:** [4/5/2024, 7:28pm]
- **Findings:** The logout button was put into the header component in our react codebase. The functionality of the logout button needs to be present in the dashboard component. The function that allows this event to happen is the **handleLogout()** it's an async function. The solution that was thought of is to move this function into the dashboard component allowing the button to be present in the dashboard's navbar when signed in.

Inspection 3:

- **Inspector:** [Jon Juarez]
- **Date & Time:** [4/25/2024, 4:21pm]
- **Findings:** Images cannot be deleted or updated accessibly. A user must upload a new set of images depending on what they prefer. They cannot add/remove without resetting all of the image links. I tried adding a single image but it deletes the rest of the images. The rest of the images were also still remaining in the database. Conflicting file-names can also influence the Cloud Storage image uploads.

V Recommendations and Conclusions

Security rules: This was not fully implemented in order to meet our deadlines for our project demo. Must be implemented in a way that is parallel to the database relationships and transactions. If taking this further, using atomic serverless transactions/functions would eliminate the issues with security rules.

Naming Scheme for useGeoLocation: This issue did not impact the entirety of the project as only 1 individual had to account for it when implementing usage of it. This

was easily accounted for but in the case of further development or onboarding with a larger development team, will require suitable changes and documentation.

VI Project Issues

1 Open Issues

- 1a Account creation is still not fully completed. Due to a dependency on Firebase Authentication for OAuth providers, our app requires manual deletion of accounts in the database. The Firestore and Cloud Storage data linked to each account also does not delete when an account is deleted. In order to solve this issue, we would have to spend a certain amount of money on Firebase's serverless functionality, such as Cloud functions to run transactional data operations on actions such as Account Creation/Deletion, and more.
- 1b Some database relations are weakly linked and do not have a way to use and enforce completely atomic operations. An example would be changing the latitude and longitude coordinate of a space does not change its geohash, and vice versa. Another example would be changing a guide/organization username. This can be fixed by instead converting direct database access to custom made Firebase Cloud Functions, and instead accessing that interface/API.
- 1c Choosing location for a space is inaccessible and restricted by which locations the user can access physically, as well as their device. An issue that could come up would be the device could not have accurate locational data in a certain area, blocking them from placing the space in a spot that represents the location accurately. Another issue is that a user would have to travel to the spot in order to change it's location.

2 Waiting Room

Favorites List

Users of the application may frequent certain spaces for their productivity needs. Having a system like a favorites list allows them to add/remove their favorite spaces which they can easily access instead of having to search for them among other spaces. This allows for efficient use of the application which in turn will bring less work and frustration for the user.

Account Follow

Users should also be able to follow guide/organization accounts and prioritize them over other users. This allows for them to focus only on official accounts, trustworthy guides or close friends. This could also notify them of new spots that a person or organization wants to notify them about, whether it be a release of a new official study space, or one that not many people knew about before. This can be used as both public information for the University's amenities, as well as a good way to spread information about unused spaces or lounges on campus.

Space Ratings

Determining which space to use can be a very daunting task. The space rating feature helps alleviate these problems as this allows the user to be able to rate any of the spaces they have visited. Space ratings will have numerous ways the user can review the space. Options like rating the amenities, atmosphere, crowd density, overall star rating, and a description section where the user can write about their visit there. The space rating is public to other users and it can be viewed when clicking on space to see more information.

Check In / Check Out

When picking a space, the users may want to make sure that their location is not already full or crowded with people. The check in/out feature attempts to solve this problem. Users of the app can have an option to be able to check into their space and check out when they leave. If location is allowed, the application will automatically detect if the user is within the space region and do the checking in/out process automatically. The result of this feature can be displayed directly on the space card itself as a meter signaling other users on the app how full that space is.

Search Filtering

As the list of spaces for a certain area is populated with more and more information, it will become more difficult to find spaces. A method to filter spaces by eliminating or filtering certain tags, or sorting by distance/popularity/rating order would be a great addition for accessibility.

3 Ideas for Solutions

For checking in and checking out, a simple assumption that can be made to simplify implementation is that a user can only be checked into one space at a time. Meaning the starting relation or data representing this could be a space property under the db/users table. The rest of the relational data structures and transactions can be modeled from this.

A way to choose locations based not only on device location data could be done by having a “map picker” window when creating a space. This can be implemented by having a Pin (using the vis.gl Google Maps and React library) and drag/drop functionality.

Some solutions to implement search filtering without querying unnecessarily would be using client-side filtering and sorting. This would prevent unneeded requests and traffic of the Google Maps API and database as locational queries are realistically only required on large distance changes. This can also be implemented using the search params in a URL if needed. Some method of storing the user’s history or current page would be required to not reset the app’s data.

4 Project Retrospective

As we approach the completion of the Silent Space's Locator's documentation, it's important to reflect on the journey we've taken to efficiently and coherently fill out each section of this report. Throughout the project, certain approaches proved effective, others not so much. Below we dive into what worked well and areas where improvements could be made.

What Worked Well: The usage of a dedicated discord channel, along with weekly team meetings helped facilitate clear and timely communication among team members. Most of us would check discord fairly regularly so if any questions or concerns/thoughts were raised there would be an easy way to decide the best path moving forward. This helped in sharing progress updates, and dividing work based on what each team member could do given their workload for that week. The discord channel was also used to jump on calls, simulating an all hands meeting when it came to discussing changes being made to the site or to showcase functionality.

What Didn't Work Well: We would typically underestimate the time a task or section of the report would actually take, which led to stressful last minute pushes to make sure deliverables were submitted on time. Additionally, there was sometimes a lack of consideration for the fluctuating schedules of team members due to personal or work commitments. We could have been a bit more proactive in adjusting our plans to accommodate these changes throughout the semester. We also originally set the bar a bit too high for what we were hoping to accomplish by each scenario, and believe that given a bit more time we could have implemented everything but were proud of our final product nonetheless.

Recommendations for Future Improvement: Include better/more detailed diagrams, since diagrams are a great way to facilitate a reader's understanding. Keeping expectations manageable as well would help, seeing as we would come up with big ideas that would sometimes veer us off course of what an app like Silent Spaces is intended to be, especially with a team of only 4 novice developers. Lastly, maintaining flexibility in accommodating conflicting schedules and work sessions will greatly benefit a team in the long run, much like the agile principles we've been following throughout the semester, in adherence to the Scrum methodologies. This ensures smoother collaboration and helps streamline the process of drafting large project reports.

VII Glossary

Space: A physical area within a building or location designated for work or study purposes.

SpaceCards: Visual representations (cards) that provide summarized information about a particular space, located in the SpaceList and often used for browsing or selection purposes.

SpaceView: A detailed view or interface presenting comprehensive information about a specific space, including its location, amenities, and current status.

SpaceList: A catalog or list displaying available spaces, typically categorized and sortable by various criteria such as location, features, and ratings.

Organization: An entity such as a university, library, or company, within which spaces are managed and accessed by users. Organizations are essentially the owners of a registered space.

Dashboard: A centralized interface providing quick access to various features and functionalities of the app, typically customizable to suit user preferences. Home of the app's key functionality: Create/Find a Space, SpaceList, SpaceViews, SpaceList, etc.

Find a Space: A feature allowing users to search, browse, and select from available spaces based on location, and space amenities.

Create a Space: This functionality empowers you to establish a new area within your organization, whether it's a study room, conference area, or communal space, tailored to meet the needs of your users. Various tags are available when creating a space, which are displayed on the SpaceCard to signal what amenities are available in said space.

Check-in: The action taken by a user to indicate their presence or occupancy within a specific space, typically recorded for organizational or informational purposes.

Check-out: The action taken by a user to indicate their departure or no longer being present within a specific space.

Favorites: Spaces that a user has marked or saved for quick and easy access, often based on personal preferences or frequent usage.

Location Data: Information about the geographical position of a user, typically obtained from GPS or other positioning technologies.

VIII References / Bibliography

- [1] Robertson and Robertson, **Mastering the Requirements Process.**
- [2] A. Silberschatz, P. B. Galvin and G. Gagne, **Operating System Concepts, Ninth ed., Wiley, 2013.**
- [3] J. Bell, **"Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.**
- [4] M. Fowler, **UML Distilled, Third Edition, Boston: Pearson Education, 2004.**

[5] Group 9: I. Banal, M. Dodiya, F. Mantilla, N. Nerella, “Development Project Final Report”, University of Illinois Chicago, 2022

IX Index

Project Description	(6 - 10)	Project Deliverables (11 - 16)
Testing	(16-29)	Rec. & Conclusions (30)
Project Issues	(31)	Glossary (34)
References	(34)	Index (34)