

WHERE 2NEXT

CS 3354.001



OBJECTIVE AND FEATURES

Where2Next is a game changing extension for foodies, athletes, social butterflies to find new and exciting outings and restaurants.

- “Where2Next,” is a Google Maps extension
- adjustable radius feature
- filters for user-nominated tags
 - “peaceful”
 - “vegan”
 - “adrenaline-inducing”
- match the intended atmosphere of the outing
 - “lively”
 - “active”
- records a chronological log of the past locations you’ve been to
- records past activities, food ordered, etc

COST ESTIMATION FUNCTION POINT

	Function Category	Cost	Complexity			Count x Complexity
			Simple	Average	Complex	
1	Number of user input	5	3	4	6	20
2	Number of user output	4	4	5	7	28
3	Number of user queries	3	3	4	6	9
4	Number of data files and relational tables	5	7	10	15	50
5	Number of external interfaces	2	5	7	10	20
					GFP	127

FUNCTION POINT CALCULATIONS

Q1: 4 Q2: 3 Q3: 2 Q4: 5 Q5: 5 Q6: 5 Q7: 2 Q8: 3 Q9: 3 Q10: 3 Q11: 3 Q12: 1 Q13: 3 Q14: 5

$$GFP = (5 * 4) + (4 * 7) + (3 * 3) + (5 * 10) + (2 * 10) = 127$$

$$PCA = 0.65 + 0.01 (4+ 3 + 2 + 5 + 5 + 5 + 2 + 3 + 3 + 3 + 3 + 1 + 3 + 5) = 1.12$$

$$FP = 127 \times PCA = 127 * 1.12 = 142.24 \text{ FP}$$

Productivity: 5 function points per person week because employees are not familiar with developing a project of this scope

$$E = FP / \text{Productivity}$$

$$E = 142.24/5 = 28.448 = 29 \text{ person-weeks}$$

Team size = 10 since our project is a start-up company.

$$\text{Project duration} = E / \text{team size} = 29/10 = \text{3 weeks}$$

COST ESTIMATION

HARDWARE, SOFTWARE, PERSONNEL

Estimated cost of hardware products: \$15,000

- Company laptops: $\$1500 \times 10 = \$15,000$

Estimated cost of software products over a year: \$63,599.95

- Jira is a project management software, which we believe is necessary to keep track of our project schedule and employees. Jira is essential when using agile, so we believe it is worth the price of \$1600.
- In order to cut costs on hardware, we plan to use Google cloud services for storing information. The cost is around \$5000 a month.
- Because our app utilizes Google Maps, we must purchase the Google Maps API license, specifically the license that allows us to use dynamic maps. Its cost is \$1500 a year.
- Although we are a start up, we take cybersecurity very seriously, which is why we subscribed to FalconPro's services, which cost \$499.95 a year for small businesses.

Estimated cost of personnel over a year:

- Employee engagement has a budget of \$500 a month.
- 10 employees
- Each employee will be paid \$65,000 a year.
- According to the Harvard Business Review, on average, employee retraining costs a company about \$24,800 per worker. This cost covers the time spent training as well.

PROJECT TIMELINE

Week 1: Initial Planning and First Design

- Day 1: Start the project and define its scope.
- Days 2-3: Set up infrastructure and begin design work.
- Days 4-5: Conduct risk analysis on the design.
- Days 6-7: Develop the first prototype.

Week 2: Development and User Feedback

- Days 8-9: Develop the first prototype.
- Days 10-12: Conduct user testing and gather feedback.
- Days 13-14: Develop the final prototype, incorporating user feedback.

PROJECT TIMELINE

Week 3: Prototype Refinement and Final Testing

- Days 15-16: Complete the final prototype and commence in-depth testing.
- Day 17: Begin beta testing.
- Days 18-19: Analyze beta test feedback and make final adjustments.
- Days 20-21: Complete project documentation and launch.

This timeline is based on our function point modeling technique with a team of 10 people. It is estimated to take 3 weeks to complete the project given to our team. The spiral method is also implemented so aspects of said model are in our timeline such as multiple prototypes and risk assessment. Weekends are counted as part of the schedule/work period. Normal working hours are 9 am - 5 pm as this is common company policy.

Start Date: January 1st, 2023

Release Date: January 22nd, 2023

FUNCTIONAL REQUIREMENTS

1. The user shall be able to search for activities based on adjectives such as “lively” (filter) or specific key-words (vegan, epilepsy-friendly)
2. The user shall be able to set the distance radius of the results they receive from their current location.
3. The application shall keep a chronological log of past activities the user has attended.
4. The application shall keep a note for each past activity which includes information on what the user did while there.
 - For every past activity done, user should be able to append adjectives from application bank to describe the activity
5. The application shall have a secondary mode that enables users to add a public event/activity location.
6. The user shall be able to share activities from the application to external applications (e.g. social media)

NON-FUNCTIONAL PRODUCT REQUIREMENTS

PART 1

1. After completing the 1-minute application walk-through video, users should be able to identify the buttons for specific tasks
 - (e.g., searching a location, accessing activity logs) within 30 seconds.
 - This requirement is derived from the usability requirement leaf from figure 4.3
2. Each user transaction should take no more than 5 seconds
 - This requirement is derived from the performance requirements leaf from figure 4.3
3. Each user will be allocated 10 GB in data to store location history and additional notes
 - If 10GB is passed, then a paywall will appear.
 - This requirement is derived from the space requirement leaf from figure 4.3

NON-FUNCTIONAL PRODUCT REQUIREMENTS

PART 2

4. The application should not have more than three, 30 minute down-times within a 30-day period.

- Every downtime must be addressed and closed within 30 minutes.
- This requirement is derived from the dependability requirement leaf from figure 4.3

5. To protect against fake/spam accounts, the user will have to use some sort of third-party authentication to verify their identity.

- This requirement is derived from the security requirement leaf derived from the product requirement node from figure 4.3.

NON-FUNCTIONAL ORGANIZATIONAL REQUIREMENTS

1. The application must be run on a physical computer device that is between 32 to 95 degrees Fahrenheit in temperature for the specified product requirements to be met.
 - This requirement is derived from the environmental requirement leaf from figure 4.3.
2. The application must be supported on Android and IOS operating systems per Google Maps available usage.
 - This requirement is derived from the operational requirements leaf from figure 4.3.
3. Application must be written in a secure and reliable language.
 - Secure in the sense that the developers will prioritize language security over memory efficiency.
 - This requirement is derived from the development requirements derived from figure 4.3

NON-FUNCTIONAL EXTERNAL REQUIREMENTS

1. Updates on software should be done during the hours servers receive the least traffic and should take no more than 6 hours
 - This requirement is derived from the regulatory requirements leaf from figure 4.3.
2. The application will screen out activities that contain illegal actions and/or hateful/discriminatory language.
 - (e.g., drugs, prostitution, white-supremacist groups)
 - This requirement is derived from the ethical requirements leaf from figure 4.3.
3. Application profits must be reviewed every fiscal quarter and discrepancies in profits and operational costs are to be addressed within 72 hours.
 - This requirement is derived from the accounting requirement leaf from figure 4.3.
4. This app will not sell user data to third parties. Data will only be used to observe user trends and for the recommendation system.
 - This requirement is derived from the safety/security requirements leaf derived from the legislative requirement node from figure 4.3.

SOFTWARE PROCESS

SPIRAL



- “Where2Next” has multiple features and heavily relies on user interaction
- necessary to come up with multiple prototypes before slowly perfecting the product
- spiral model offers the perfect division
- spiral model > prototyping evolutionary model because our product deals with a lot of data
- with each prototype created, risk must be analyzed to ensure data is not vulnerable
- allows developers to move cyclically, repeating some steps to ensure we are able to meet requirements while developing a secure product that our customer is comfortable using
- determining objectives and constraints before evaluating and resolving risk
- develop product before planning the next phase offers an organic feel to the development process

TEST PLAN

To test our software the minimum unit required for our software to function is the user profile containing user information. Here, there are six test cases in this order: a successful creation of a new user (as an object), a test case for both valid and invalid ages when registering, a test case for both valid and invalid names when registering, and a failed test case for passwords when registering. A single user profile requires three things: a valid name, a valid password, and a valid age. An age is considered valid if the user's age is within 15 to 109 years of life ($14 < X < 110$, where X is the user age). A [user]name is valid if both input names are no greater than 20 characters in length (the underscore is added by the system and not by the users). Finally, a password is considered valid if the length of the password is either between or exactly 6 and 12 characters long, complexity is not required! ($6 \leq X \leq 12$, where X is the number of characters in the password). Should a user successfully meet all criteria they will become registered as a user and added to the database [implementation pending], but should a user fail at any stage of the criteria then they will be prompted to correct their error. This does not extend to user age verification as a failure for that component results in a critical failure to discourage underaged and joke accounts from becoming created.

```
21 //Below lines 24 - 55 represent what occurs inside makeNewUser() (Could not be tested on automatically because it
22 // requires user input and I could not figure out how to add pre-typed statements as proxy user input).
23 //Example of a test of Age that should pass
24 @Test
25 public void testMainVerifyAgeCorrect() {
26     String DOB = "12_2_2001";
27     WhereToNext.verifyAge(DOB);
28 }
29
30 //Example of a test of Age that should fail & therefore print a custom warning (Test should Pass)
31 @Test
32 public void testMainVerifyAgeIncorrect() {
33     String DOB = "12_2_2024";
34     WhereToNext.verifyAge(DOB);
35 }
36
37 //Example of a test that should pass
38 @Test
39 public void testMainVerifyNameCorrect(){
40     String name = "Lennon_Crow-bar";
41     WhereToNext.verifyName(name);
42 }
43
44 //Example of a test that should fail & therefore print a message to terminal (Test should pass)
45 @Test
46 public void testMainVerifyNameIncorrect(){
47     String name = "Ariel_PleaseReadTowerOfGod";
48     WhereToNext.verifyName(name);
49 }
50
51 //Example of a test that should fail & therefore print a message to terminal (Test should pass)
52 @Test
53 public void testMainVerifyPass(){
54     String pass = "Unacceptable!";
55     WhereToNext.verifyPass(pass);
56 }
```

✓	WhereToNextTest	6 ms
✓	testMainVerifyPass	6 ms
✓	testMainWithName	0 ms
✓	testMainVerifyAgeIncorrect	0 ms
✓	testMainVerifyAgeCorrect	0 ms
✓	testMainVerifyNameCorrect	0 ms
✓	testMainVerifyNameIncorrect	0 ms

PASS

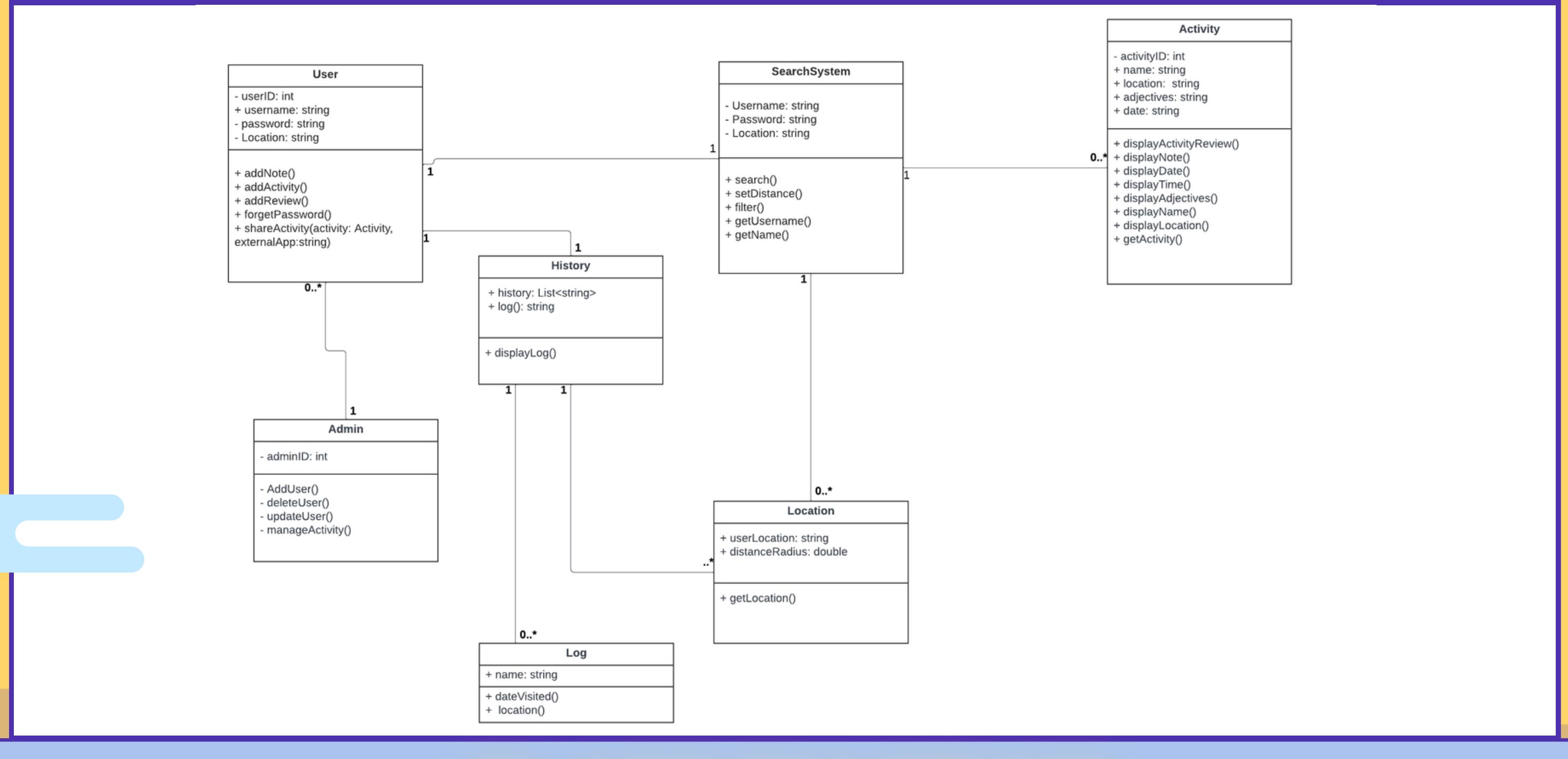
```
8  ↵ | public class WhereToNextTest extends TestCase {  
9  
10     //Should any one of the test below fail for whatever reason assume that the user cannot create/access their account  
11  
12     //A test of constructor to show that object can be created  
13     @Test  
14    ↵ |     public void testMainWithName() {  
15         String name = "Lennon_Crow";  
16         String pass = "Pr0gr@m";  
17         int age = 27;  
18         User test = new User(name, pass, age);  
19     }  
20 }
```

FAIL

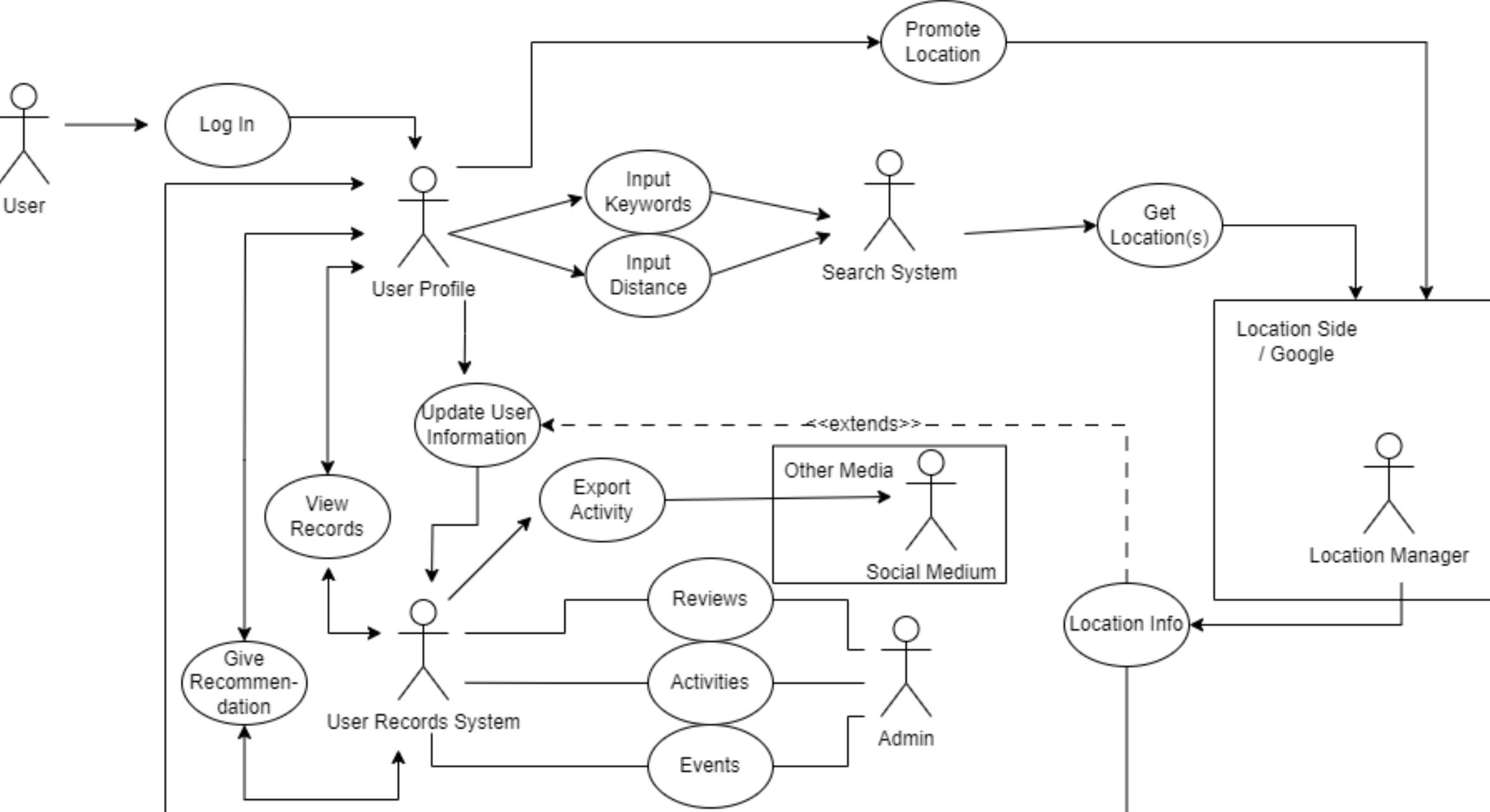
```
30     //Example of a test of Age that should fail & therefore print a custom warning (Test should Pass)  
31  
32     ↵ |     @Test  
33     ↵ |     public void testMainVerifyAgeIncorrect() {  
34         String DOB = "12_2_2024";  
35         WhereToNext.verifyAge(DOB);  
36     }
```

Tests passed: 1 of 1 test - 0 ms

C:\Users\John\.jdks\openjdk-21\bin\java.exe -ea -Didea.test.cyclic.buffer.size=1048576 "-javaagent:C:\Program Files\
We regret to inform you that you are too young to be serviced, please inform your parent / legal guardian

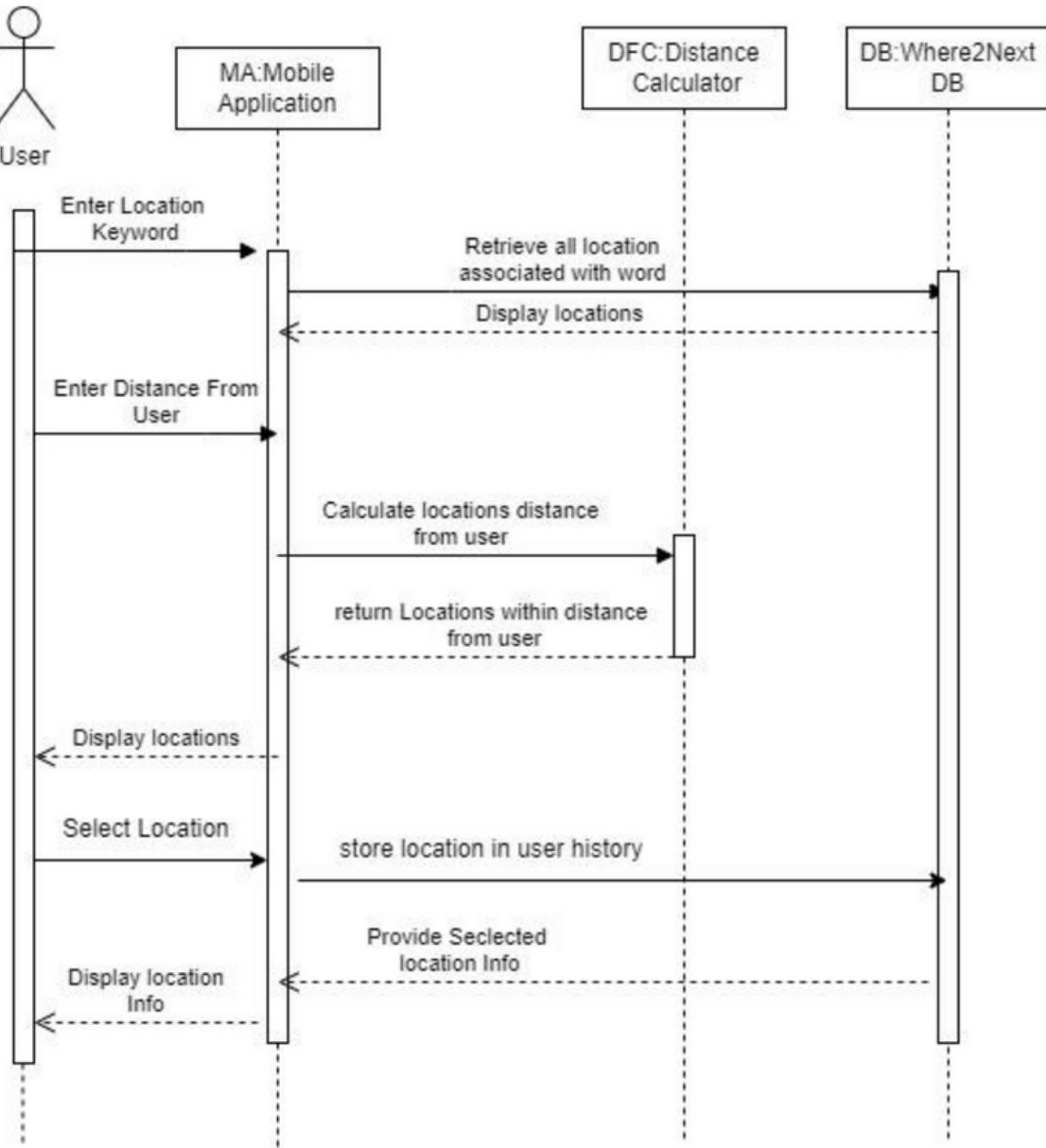


CLASS DIAGRAM



CASE DIAGRAM

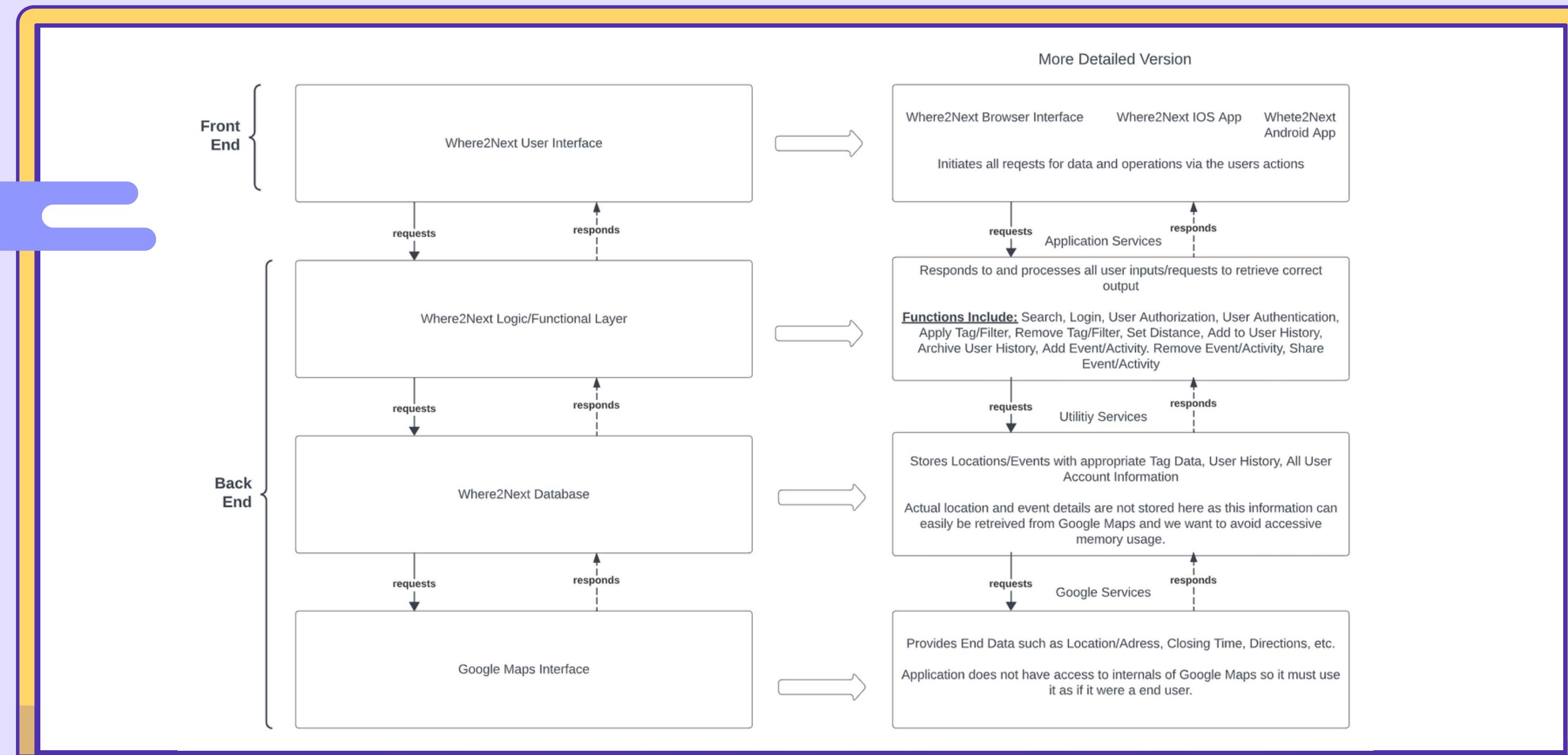
Sequence Diagram for searching for a location using keywords within a distance



SEQUENCE DIAGRAM

WHY LAYERED ARCHITECTURE?

- It is well suited when building new facilities on top of existing systems
 - You can easily add new layers on top of existing systems to introduce new functionalities and operations
- Supports the incremental development of subsystems in new layers
 - Allows replacement/removal/addition of entire layers so long as the interface between adjacent layers is maintained
 - This couples well with the Spiral Approach
- Redundancy can be introduced between layers to increase software reliability
- It is often a naive and impractical approach to very complex systems
 - Difficult to provide a clean separation between layer functionality
 - Performance of system may be hindered as layers can only communicate with adjacent layers



LAYERED ARCHITECTURE

LAYERED ARCHITECTURE PATTERN

Where2Next User Interface:

- Provides the graphical interface to user (Renders data and interface to user).
- Initiates all requests for data and operations via the users actions.

Where2Next Logic/Functional Layer:

- Receives inputs from the User Interface and maps user interaction to corresponding logic operations/functions.
- Carries out operations and requests necessary data from Where2Next Database



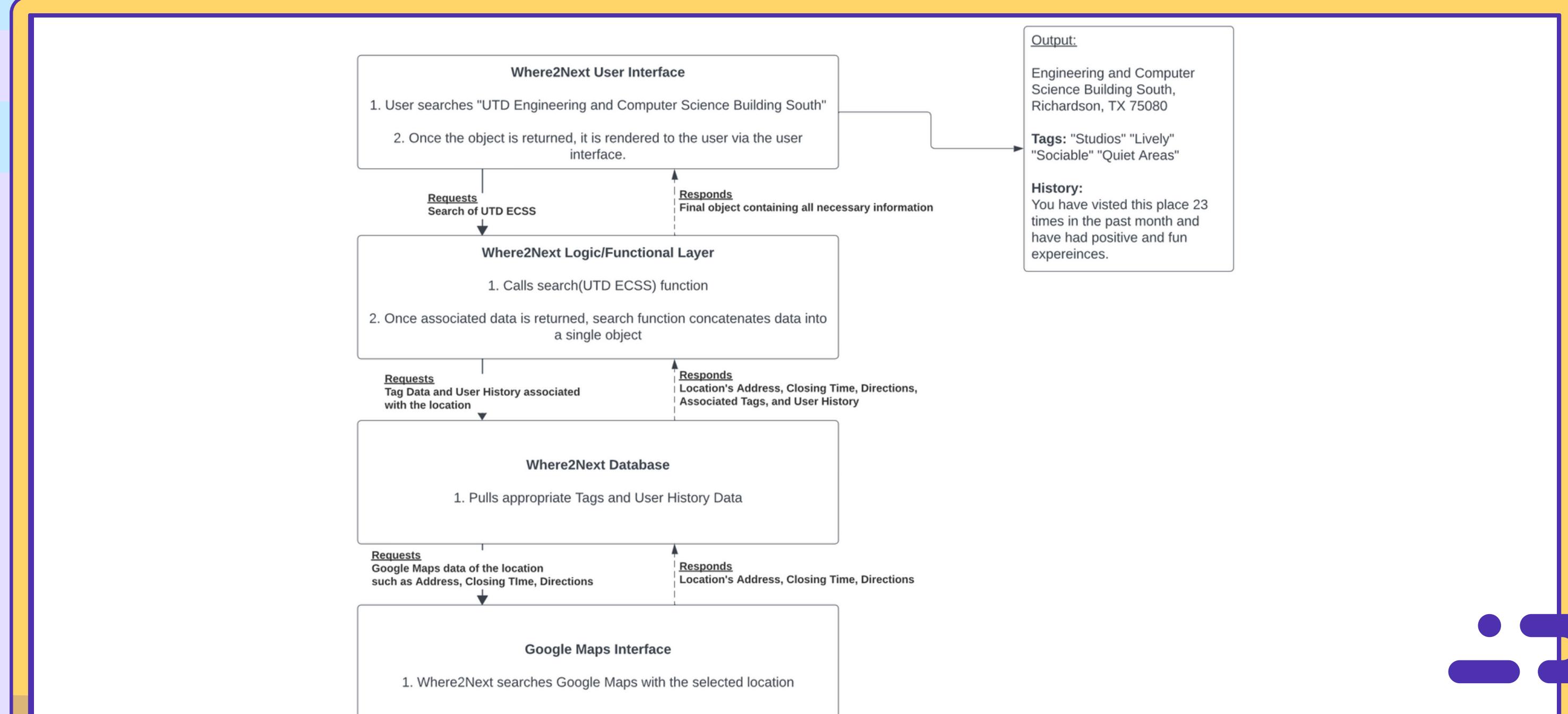
LAYERED ARCHITECTURE PATTERN

Where2Next Database:

- Houses all data specific to Where2Next such as tags and user data (Account Information, History, Reviews, Activities, Events).
- Actual location and event details are not stored here as this information can easily be retrieved from Google Maps and we want this to avoid excessive and redundant memory usage.
- The goal here is restrict stored data only to that which is necessary and unique to Where2Next.

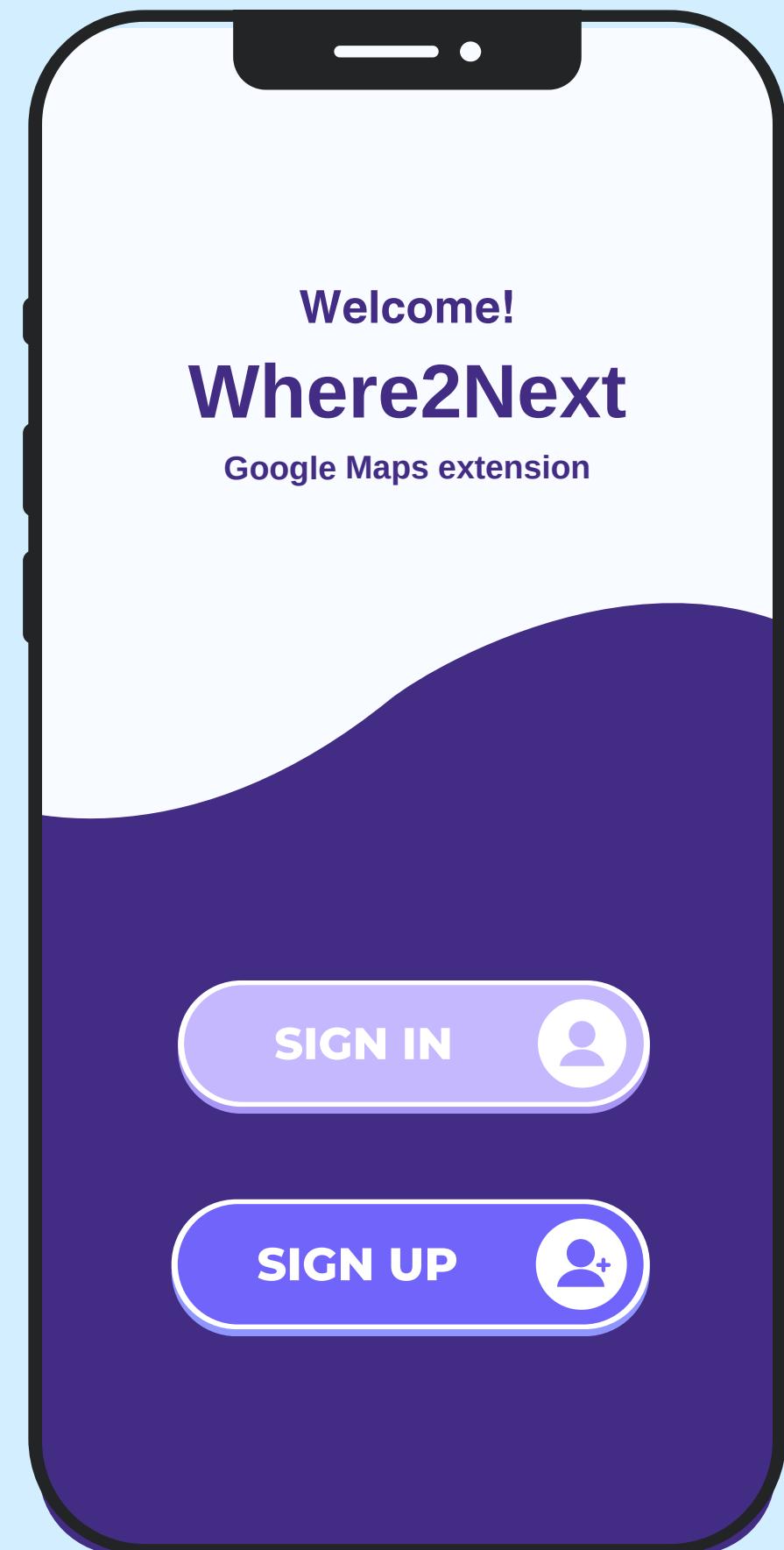
Google Maps Interface:

- We do not have access to the internals to Google Maps, so Where2Next must use it as if it were a ordinary end user.
- Google Maps will provide end details such as address, closing time, and directions.
- The data from Google Map can be processed however necessary and presented via the Where2Next Interface rather than the Google Maps Interface.



EXAMPLE SEARCH

USER INTERFACE





Temoc
School mascot

[Edit Profile](#)

Previous Activity [Favored Tags](#)

Activity

 **Burning Rice** 10/27/2023
Ordered: Rice Bowl with Chicken
Rating: 4/5

 **Orange Theory Fitness** 10/06/2023
Attended: 60 minute HIIT Class
Rating: 3/5

 **Torchy's Tacos** 9/27/2023
Ordered: Taco of the Month
Rating: 5/5



Looking for somewhere to go? Click a tag to see recommendations.

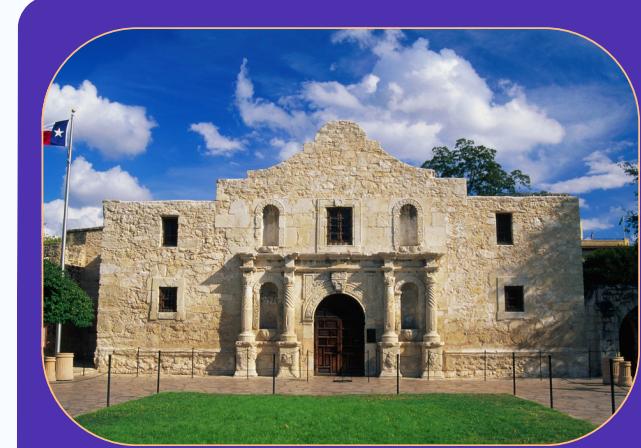
[Lively](#) [Athletic](#)
[Coffee Shop](#) [Quiet](#)
[Gluten - Free](#) [Romantic](#)
[Chill](#) [Wheelchair Friendly](#)

...





"Historic"
200 Results



The Alamo 
Rating: 4/5 • Previous Visits: 0
300 Alamo Plaza
San Antonio, TX 78205





THANK
YOU!