



CAPCOUNT

Group Project

CAPCOUNT

Isha Dighe, Rovin Jo, Shumei Lyu, Amy Parekh, Sabrina Wolday | System Analysis and
Project Management | August 1, 2020

Table of Contents

Executive Summary	1
Problem Statement	3
Business Process Model	4
Context Diagram	5
Use Case Diagram	6
Use Case 1	7
Use Case 2	8
Use Case 3	9
Use Case 4	10
Use Case 5	11
Use Case 6	12
Use Case 7	13
Class Diagram	14
Data Dictionary 1	15
Data Dictionary 2	16
Data Dictionary 3	17
Data Dictionary 4	18
Data Dictionary 5	19
Data Dictionary 6	20
Data Dictionary 7	21
Data Dictionary	22
Sequence Diagram	23
Functional Specification Document	24
Interface Design	25-29
Database Design	30
Database Constraints	31-33
Complete Class Diagram with Methods	34
Software Design	35
Contract #1	36
Contract #2	37
Contract #3	38
Contract #4	39
Contract #5	40

Executive Summary

Group number 5 has collectively chosen to pursue the development and analysis of an application that would allow users to track human capacity within restaurants, stores, etc. in live time. The current global COVID-19 pandemic has made consumers more conscious and wary of contact and exposure to other people. The CDC, the WHO, and local/federal governments have put in multiple mandates and recommendations in the past couple months to distance ourselves from people outside of our household to reduce the risk of exposure and spread of the virus.

Today, when we need to rush out to get groceries, fill up our gas tanks, grab a meal, see a doctor, etc. We are accepting the risks associated with being around people who have chosen to accomplish the same chores at the same time. However, many retail locations already have tracking/counting technology installed for sales/revenue purposes. If these technologies could report the same numbers to an open source and widely available application, then consumers could make a more informed decision about the risks they assume while leaving their homes.

For example, if it is now time to go back to the grocery store and fill up on food for the family, a user could log onto the app and see that there are currently 57 people currently inside their local Walmart, but only 22 people at a Kroger 3 blocks away. The user could then choose to make the statistically safer choice for them and their family to go to the store with less people, and therefore lower their risk of exposure to COVID-19.

We also believe this application to be very valuable, even not during times of a pandemic. Many immunocompromised people have to take precautions against exposure to others on a daily basis for their entire lives. Also, sometimes people are having a bad day or

prefer to not run into others. This application could prove very useful to these demographics as well.

Problem Statement

Business need:

- The increased risk of exposure to COVID-19 in high occupied businesses or areas.
 - It is currently recommended to maintain distance with others and refrain from gathering in large groups. Businesses may become a hub of COVID-19 cases without accurately accounting for each person present.
- The inability to communicate the level of occupancy to customers before arrival.
 - Businesses, especially large businesses, cannot communicate to each customer the amount of people present before their arrival. Customers only can have a visual representation once they have entered.
- The rise of COVID-19 has made people wary of fulfilling their errands.
 - Some businesses have lost customer traffic due to COVID-19 concerns. Customers may not want to expose themselves without knowing the amount of people there that may carry some risks.

Functionality (Objectives):

- Track the level of occupancy within the business.
- Communicate the level of occupancy with users in real time.
- Allow users to select business and see occupancy.
- Decrease stress associated with completing errands during this pandemic.

Expected Value:

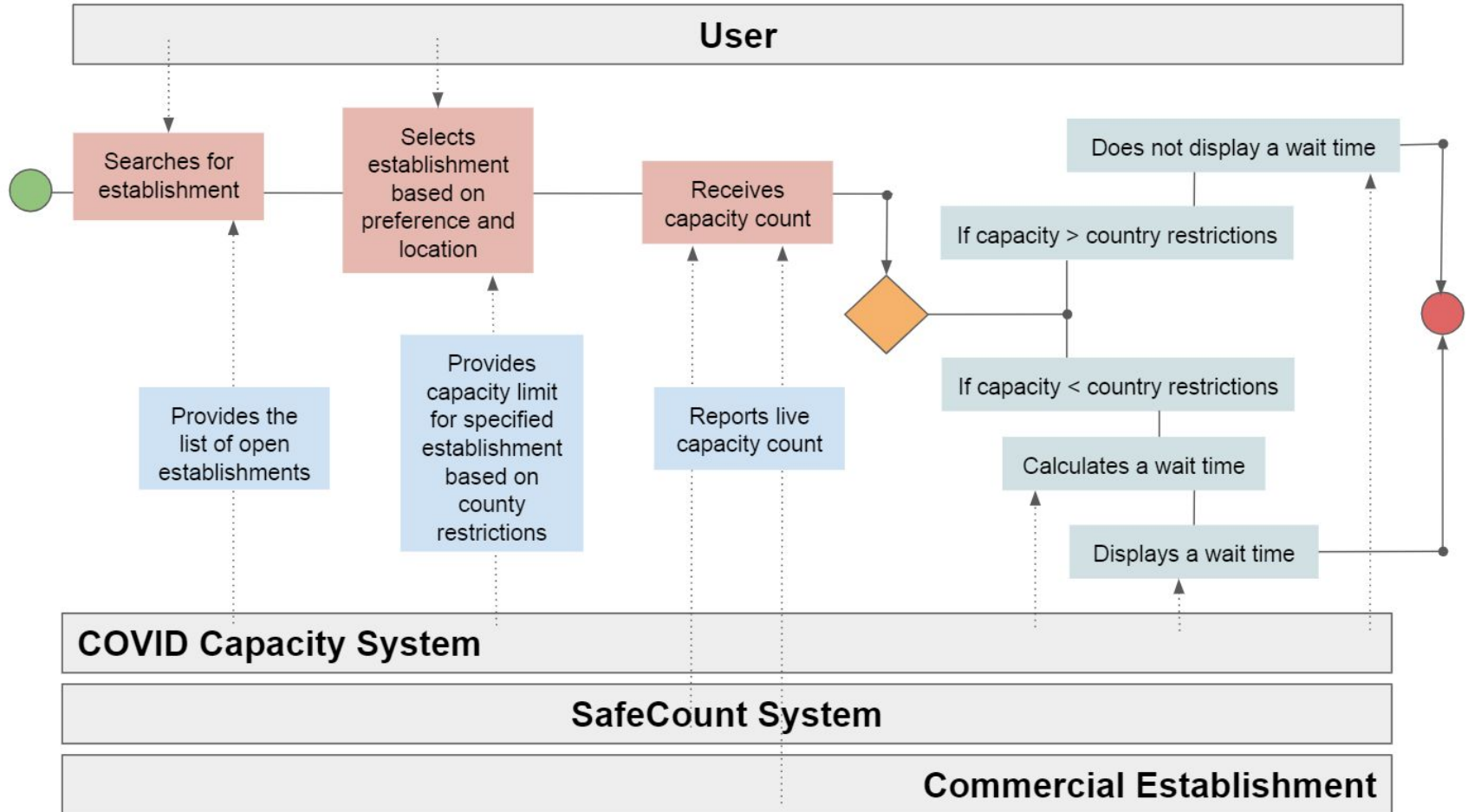
- Allow consumers to make informed decisions before entering commercial buildings in context of physical and mental health

Scope(Constraints):

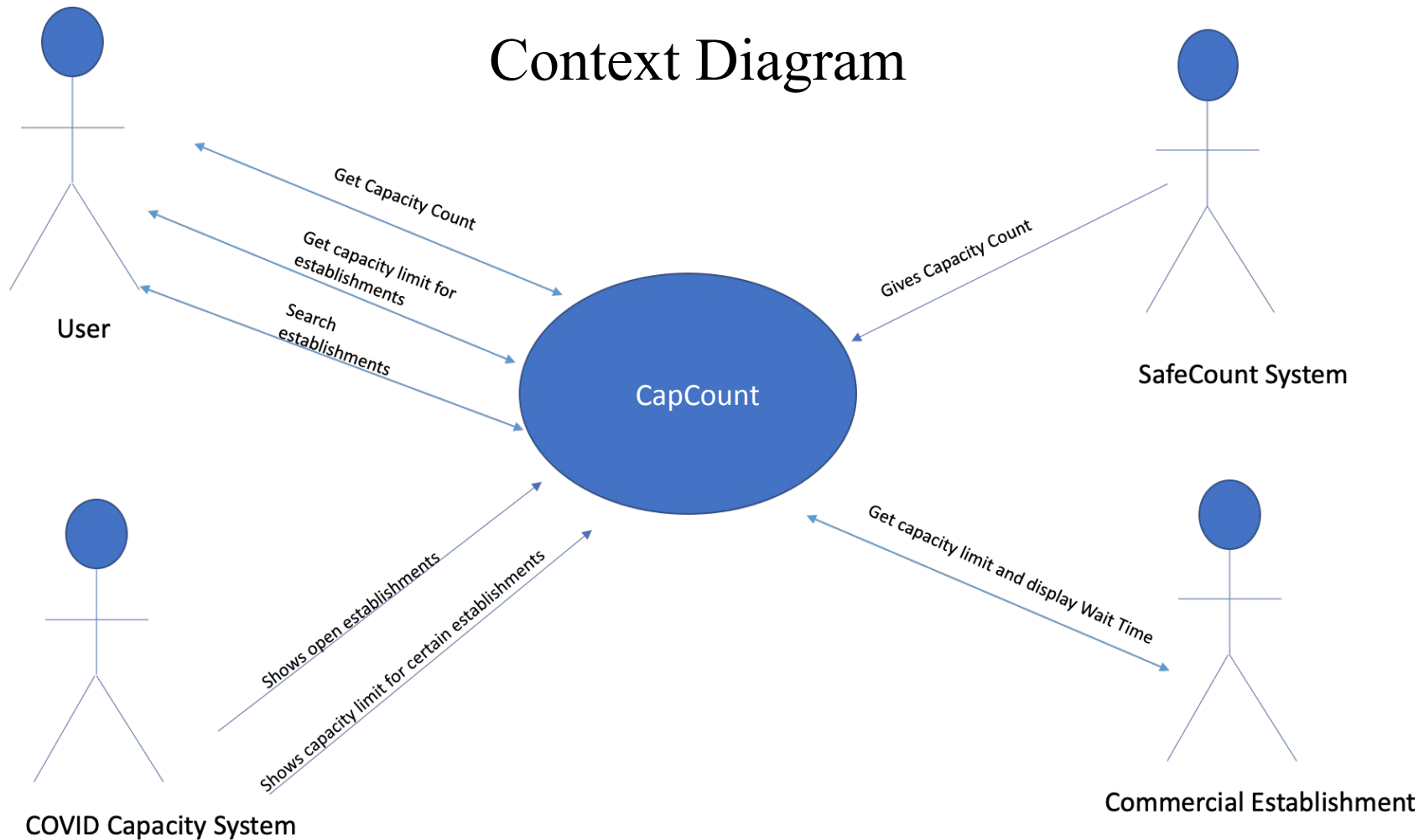
- Physically installing tracking sensors on entrances of businesses.
- Privacy concerns with consumer tracking if using GPS location
- Self-reporting can be deceiving

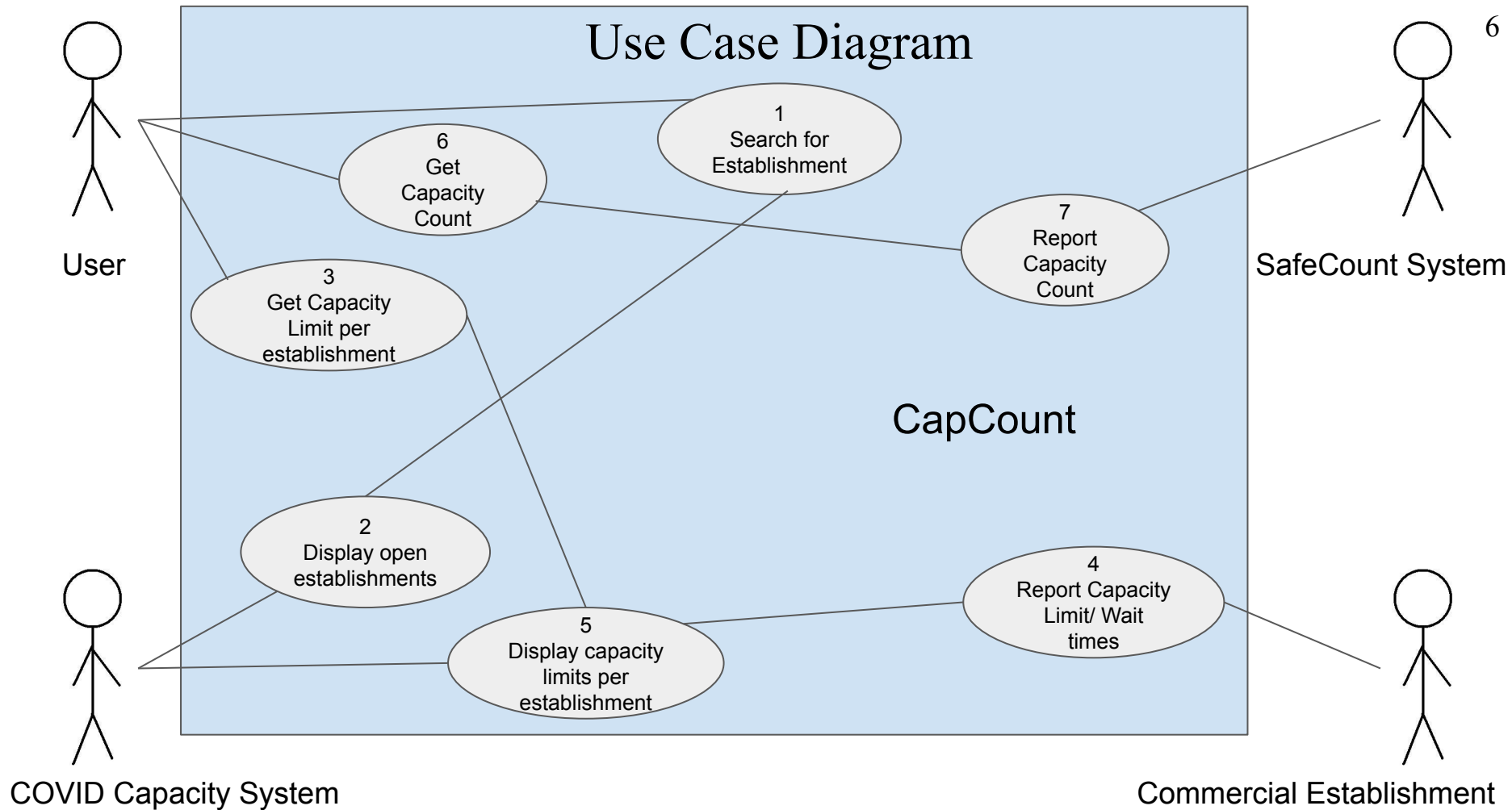
Business Process Model

4



Context Diagram





Use Case 1

Name: Search for Establishment

Description: Finds open establishments for preferred service (grocery store, gas station, auto repair, etc.)

Trigger: The user clicks on the search bar

Normal Flow:

1. Display home screen of CapCount application
2. Click search bar
3. Type in or select the desired service

Exception Flow:

The desired service is not available within the distance range selected

Use Case 2

Name: Display open establishments

Description: Show establishments providing desired service within the selected distance range

Trigger: A user selects the applicable service category

Normal Flow:

1. Give options for available service category
2. User selects the desired category
3. Asses which establishments fit the needs of service and are within distance range set
4. Determine which of these establishments is currently open
5. Return results

Exception Flow:

No results match the desired search

Use Case 3

Name: Get Capacity Limit per establishment

Description: User receives the capacity limit of their searched establishment

Trigger: User selects the establishment for their services and request capacity limit

Normal Flow:

1. An establishment is selected by the user
2. User inquires the capacity limit of certain establishment
3. Capacity limit per establishment is displayed on the CapCount application

Exception Flow:

Establishment has yet to report their capacity limit to the CapCount application.

Use Case 4

Name: Report Capacity Limit/ Wait times

Description: Commercial establishments report capacity limit and wait times to user

Trigger: User selects commercial establishment for services and receives information from businesses

Normal Flow:

1. User requests capacity limit and wait time of the establishment
2. Commercial establishments retrieves capacity limit set by government regulations
3. Commercial establishment calculates wait time in real time
4. Commercial establishment reports capacity limit and wait time to CapCount application

Exception Flow:

The establishment has yet to report government regulation of capacity limit (usually set by the fire department) or the establishment is not reporting live wait time estimations.

Use Case 5

Name: Display capacity limits per establishment

Description: COVID-19 capacity system receives capacity limits per establishments

Trigger: Establishments relays capacity limit regulations to the capacity system

Normal Flow:

1. Establishment receives capacity limit per government regulations
2. Establishment logs information of capacity limit into CapCount system
3. CapCount system displays capacity limit of establishments for user

Exception Flow:

Establishment has yet to report their government mandated capacity limit.

Use Case 6

Name: Get Capacity Count

Description: User receives capacity count of establishment

Trigger: User search for establishment for desired service

Normal Flow:

1. User searches for preferred establishment
2. User request the capacity count of establishment
3. SafeCount system relays capacity count of establishment to CapCount application
4. CapCount application displays the retrieved information

Exception Flow:

SafeCount system has faltered or failed and is no longer accurately reporting the number of people in an establishment.

Use Case 7

Name: Report Capacity Count

Description: CapCount app reports capacity count to user

Trigger: SafeCount receives user's request for capacity count of a certain business

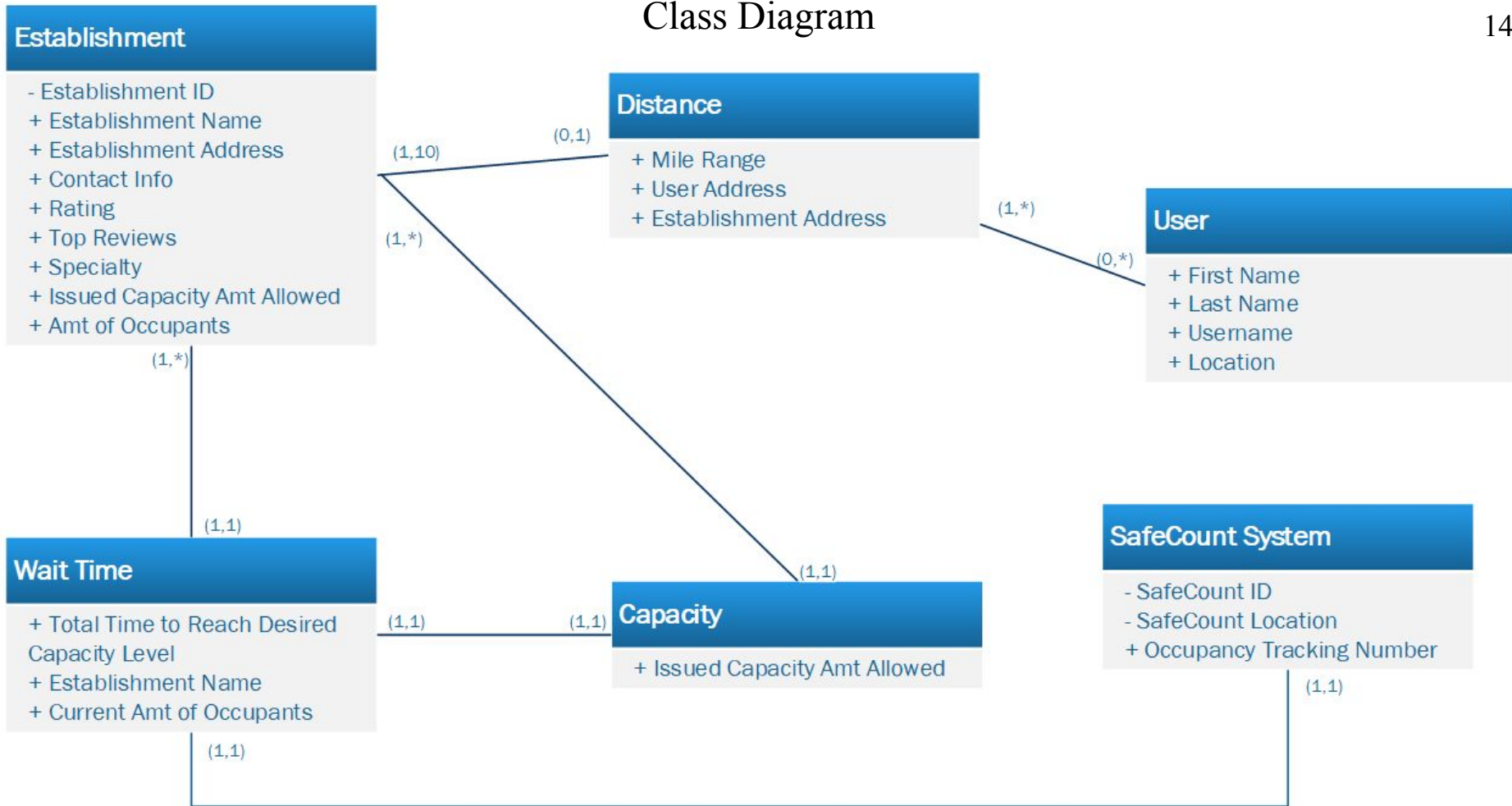
Normal Flow:

1. SafeCount receives user's request for establishment's capacity count
2. SafeCount retrieves capacity count from tracking system installed on entrances of the business
3. SafeCount reports capacity count to CapCount application
4. CapCount application displays the capacity count to the user

Exception Flow: SafeCount system has faltered or failed and is no longer accurately reporting the number of people in an establishment.

Class Diagram

14



Data Dictionary 1

User — CapCount

Use Case: Search for Establishment

Search Criteria = [Name | Establishment Type]

Distance Range = [Address | Neighborhood | City | State | Zip]

Home Screen = Recommended Establishments + Recent Activity

Data Dictionary 2

COVID Capacity System — CapCount

Use Case: Display Open Establishments

Search Criteria = [Name | Establishment Type]

Distance Range = [Address | Neighborhood | City | State | Zip]

Filter list = {Service Category} + Open Now

Establishment = 1{EstablishmentID + EstablishmentName + Address + ContactInfo + Rating + TopReviews + Speciality}10

Data Dictionary 3

User — CapCount

Use Case: Get Capacity Limit per establishment

Establishment = 1{EstablishmentID + EstablishmentName + Address + ContactInfo + Rating + TopReviews + Speciality}10

Capacity_Limit = Issued Capacity Amount Allowed

Estabalishment_Capacity = EstablishmentName + Capacity_Limit

Data Dictionary 4

CommercialEstablishment——CapCount

Use Case: Report Capacity Limit and Display Waittime

Establishment = 1{EstablishmentID + EstablishmentName + Address + ContactInfo + Rating + TopReviews + Speciality}10

Capacity_Limit = Issued Capacity Amount Allowed

Waittime = EstablishmentID + Occupants_CurrentAmount + {RealTime + TimetoReachCapacity}

Report_Capacity = 1{EstablishmentName + Capacity_Limit + Waittime}10

Data Dictionary 5

COVID Capacity System — CapCount

Use Case: Display Capacity limits per establishment

Establishment = 1{EstablishmentID + EstablishmentName + Address + ContactInfo + Rating + TopReviews + Speciality}10

Capacity_Limit = Issued Capacity Amount Allowed

Display_CapacityLimit = 1{EstablishmentName + Capacity_Limit}10

Data Dictionary 6

User — CapCount

Use Case: Get Capacity Count

Establishment = 1{EstablishmentID + EstablishmentName + Address + ContactInfo + Rating + TopReviews + Speciality}10

Capacity_Limit = Issued Capacity Amount Allowed

Capacity_Count = SafeCountID + SafeCount Location + {Establishment + Occupancy Tracking Number}

Waittime = EstablishmentID + Occupants_CurrentAmount + {RealTime + TimetoReachCapacity}

Data Dictionary 7

SafetyCount——CapCount

Use Case: Report Capacity Count

Capacity_Count = SafeCountID + SafeCount Location {Establishment + Occupancy Tracking Number}

Display_CapacityCount = EstablishmentName + Capacity_Count

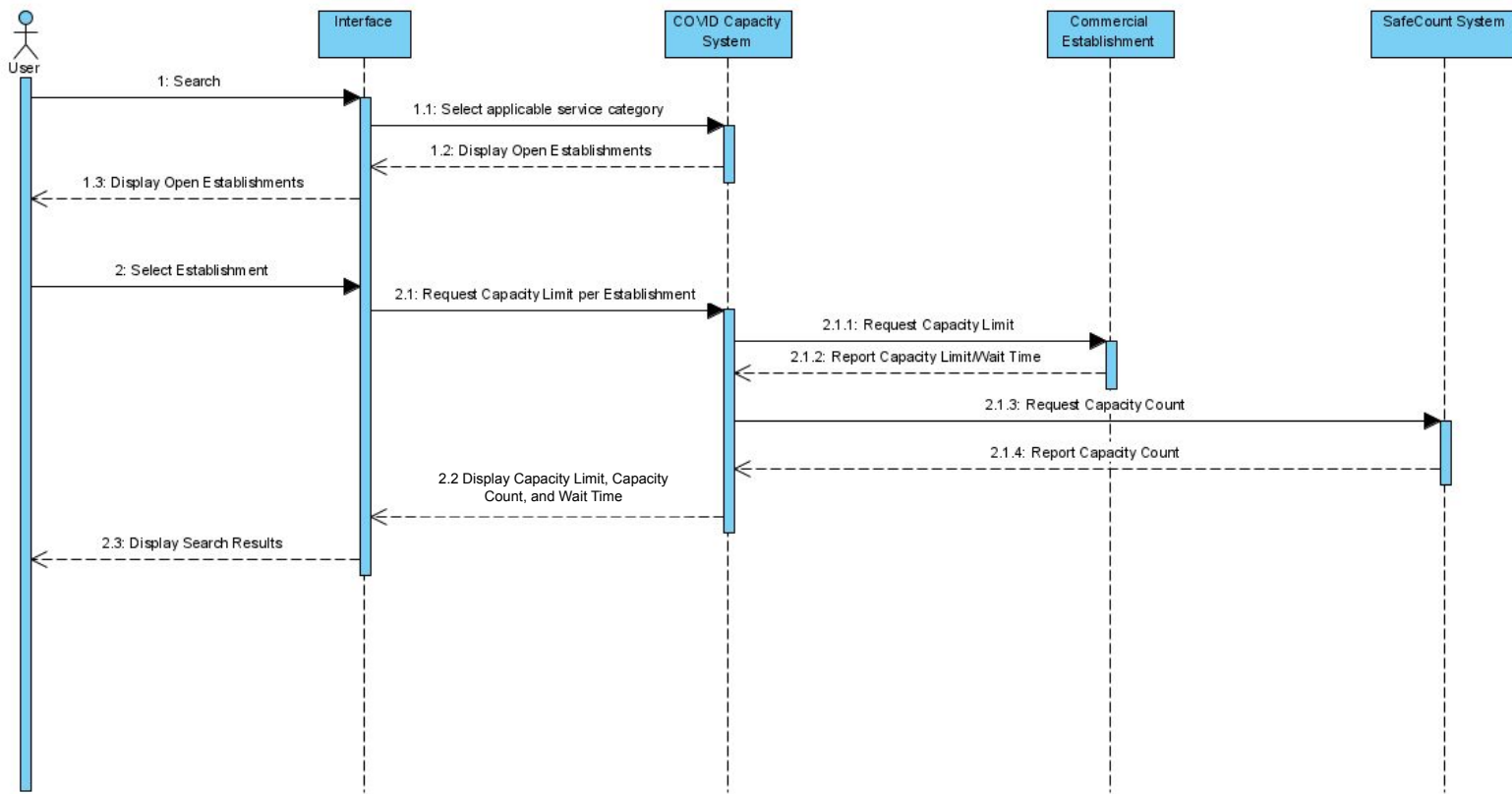
Data Dictionary

Restaurant list = 1{RestaurantID + RestaurantName + Address + ContactInfo + Rating + TopReviews + Speciality}10

Restaurant information = Overview + GetMenu Overview = RestaurantName + Rating + (Reviews) + Address + cuisine type + neighborhood + (more details) + (Direction) + restaurant description + wait time + open hours + Price range + Health inspection rating + business info + recommended review

Wait time = Data Element

Sequence Diagram



Functional Specification Document for the Proposed System

- CapCount will allow users to search for their preferred service according to live capacity updates per establishment.
 - Can filter according to service required
 - Can filter according to distance from user (GPS location must be activated)
- CapCount will display the establishments in order of distance from the user. The current capacity of the establishment, the total capacity of the establishment, and the estimated wait time will all also be displayed.
- CapCount will allow individual establishments to pay a small monthly fee to be listed before other results as a sponsored location. These establishments will only be displayed as a sponsored location if they adhere to the distance standards set by the user.
- CapCount will be free for all consumer users and only optional fees apply for establishments/service providers.
- CapCount will set an optional alert for individual users, should one of their “favorited” establishments be completely empty, or below a set capacity limit. Ex. Alert me when my preferred 7/11 convenience store has 5 or fewer customers inside.

Interface Design



CAPCOUNT



Welcome!



Username



Password

Remember me ☐

[Forgot your password?](#)

LOGIN



SEARCH FOR AN ESTABLISHMENT



RECENT SEARCHES





ENTER LOCATION



CAPCOUNT



24

BASED ON YOUR
DISTRICT,
THIS ESTABLISHMENT
HAS A
LIMIT OF 15 PEOPLE AT
ONE TIME.

ESTABLISHMENT
CAPACITY

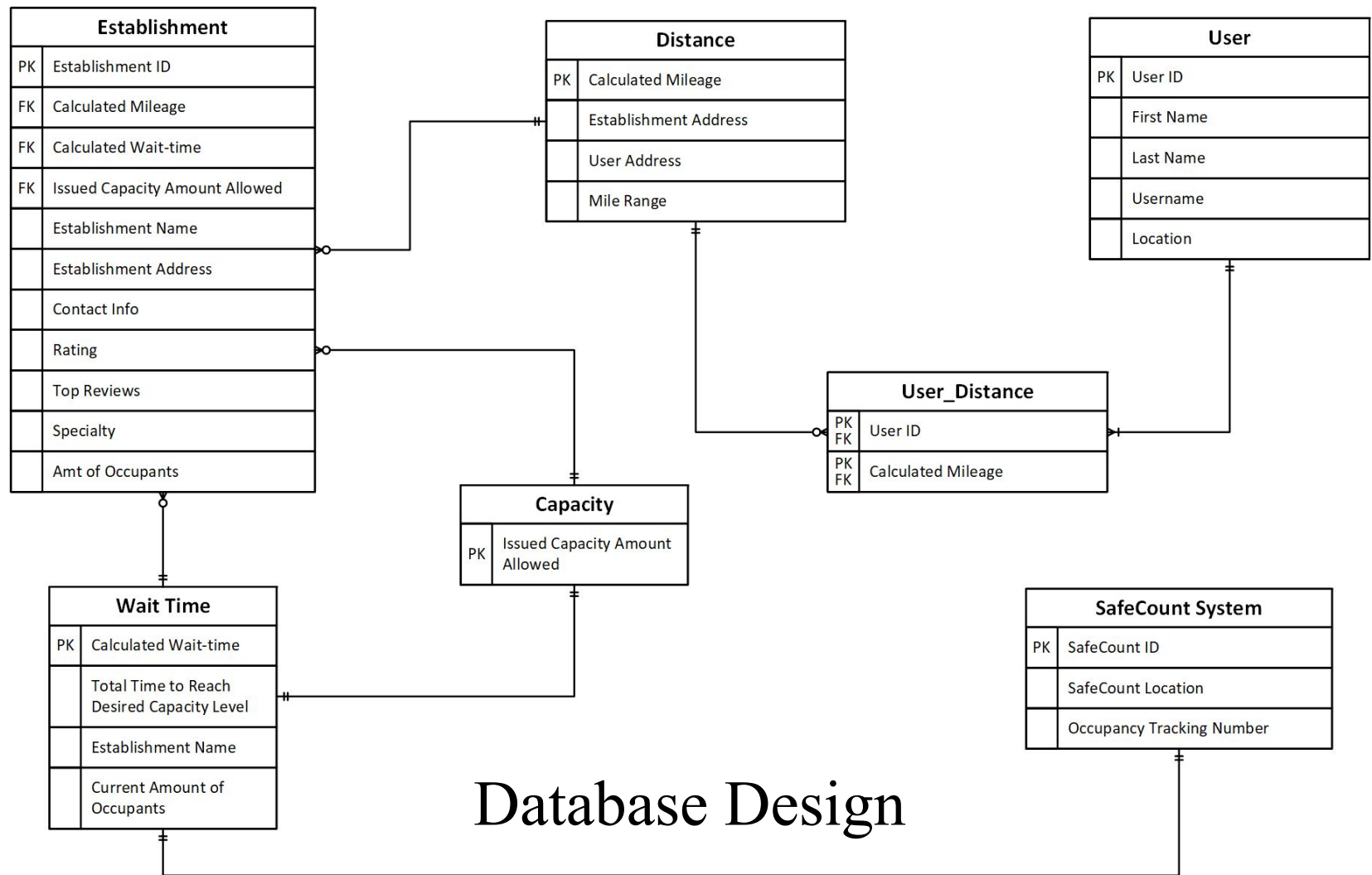
CHECK BACK LATER
FOR UPDATED WAIT TIMES!

THANK YOU FOR
KEEPING THE
COMMUNITY SAFE
THROUGH SOCIAL
DISTANCING.



42

WAIT TIME
-MINUTES-



Database Design

Database Constraints

Establishment:

Primary Key 'Establishment ID' should not be NULL.

Primary Key 'Establishment ID' should be unique.

Foreign key 'Calculated Wait-time' should exist as the primary key in Wait Time Table.

Foreign key 'Calculated Mileage' should exist as the primary key in Distance Table.

Foreign key 'Issued Capacity Amt Allowed' should exist as the primary key in Capacity Table.

Database Constraints

Wait Time:

Primary Key 'Calculated Wait-time' should not be NULL.

Primary Key 'Calculated Wait-time' should be unique.

Distance:

Primary Key 'Calculated Mileage' should not be NULL.

Primary Key 'Calculated Mileage' should be unique.

Database Constraints

User:

Primary Key 'User ID' should not be NULL.

Primary Key 'User ID' and attribute 'User ID' should be unique.

Capacity:

Primary Key 'Issued Capacity Amt Allowed' should not be NULL.

Primary Key 'Issued Capacity Amt Allowed' should be unique.

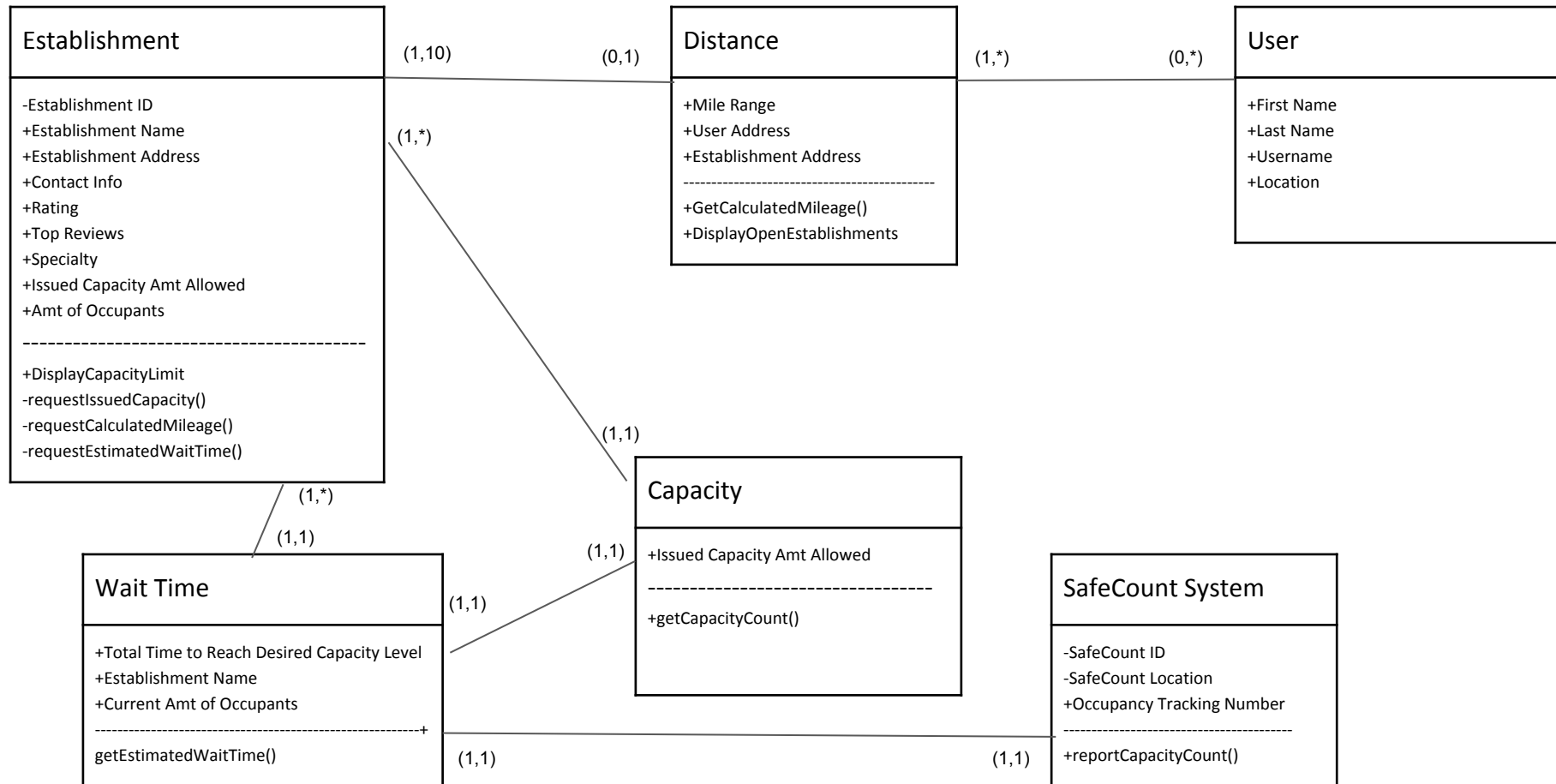
SafeCount System:

Primary Key 'SafeCount ID' should not be NULL.

Primary Key 'SafeCount ID' should be unique.

Complete Class Diagram

34



Software Design

Methods

Distance:

- +GetCalculatedMileage()
- +DisplayOpenEstablishments

Establishment:

- +DisplayCapacityLimit
- requestIssuedCapacity()
- requestCalculatedMileage()
- requestEstimatedWaitTime()

Wait Time:

- getEstimatedWaitTime()

Capacity:

- +getCapacityCount()

SafeCount System:

- +reportCapacityCount()

Contract #1: GetCalculatedMileage()

Method Name: GetCalculatedMileage()

Class Name: Distance

ID: #21

Clients(Consumers): User

Associated Use Cases: Use Case #1 - Search for Establishment

Description of Responsibilities: This method calculates the mileage between the user address and their preferred establishment choice

Arguments Received: Mile Range, User Address, Establishment Address

Type of Value Received: Varchar (max)

Pre-Conditions: Positive values, All arguments must be given, no optional parameters

Post-Conditions: Positive Numerical Value output

Contract #2 - DisplayOpenEstablishments

Method Name: DisplayOpenEstablishments

Class Name: Distance

ID: #22

Clients(Consumers): User

Associated Use Cases: Use Case #2 - Display Open Establishments

Description of Responsibilities: Display the open establishment based on user's preference.

Arguments Received: Results from GetCalculatedMileage(), based on the numerical mileage the method shows establishments providing desired establishment within the selected distance range

Type of Value Received: Integer or Float

Pre-Conditions: Positive values, All arguments must be given, no optional parameters

Post-Conditions: Will return list of open establishments (varchar output)

Contract #3 - requestIssueCapacity()

Method Name: requestIssuedCapacity()

Class Name: Establishment

ID: #32

Clients(Consumers): User

Associated Use Cases: Use Case #3

Description of Responsibilities: This method allows the user to request the establishment's capacity limit set by government regulation

Arguments Received: Capacity limit of establishments

Type of Value Received: Integer

Pre-Conditions: Positive values, All arguments must be given, no optional parameters

Post-Conditions: Will show the output of issued capacity limit of requested establishments

Contract #4 - getCapacityCount()

Method Name: getCapacityCount() Class Name: Capacity ID: #51

Clients(Consumers): User

Associated Use Cases: Use Case #6

Description of Responsibilities: This method will retrieve the Capacity Count from the establishment

Arguments Received: Capacity Count from establishment

Type of Value Received: Integer

Pre-Conditions: Positive values, All arguments must be given, no optional parameters

Post-Conditions: Positive integers that will report the capacity count from establishment

Contract #5 - reportCapacityCount()

Method Name: reportCapacityCount() Class Name: Capacity ID: #61

Clients(Consumers): User

Associated Use Cases: Use Case #7

Description of Responsibilities: This method will retrieve the reported Capacity Count from the tracking system of SafeCount.

Arguments Received: SafeCountID, SafeCount Location, Establishment and Occupancy Tracking number.

Type of Value Received: Integer

Pre-Conditions: Positive values, All arguments must be given, no optional parameters

Post-Conditions: Positive Integers from SafeCount will be reported to the User in CapCount.