Bye Felisha Spam Blocker

Release #2

Table of Contents:

Dynamic Design

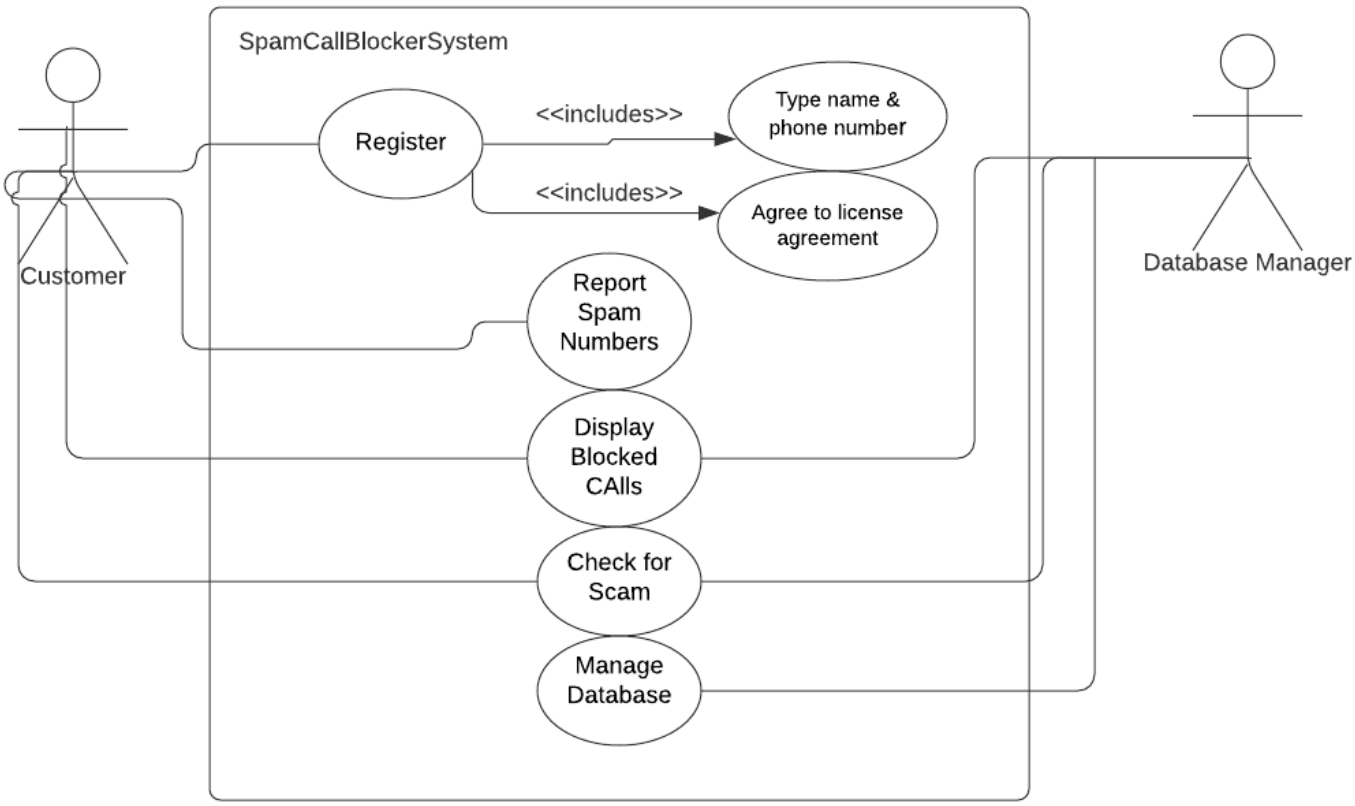
SEQUENCE DIAGRAMS::

STATE DIAGRAMS::

ACTIVITY DIAGRAM::

Bye, Felisha Use Case Diagram

| April 5, 2020



1. Use Case Specifications

Bye Felisha Spam Blocker – Use Case 1
Use Case Description: The Bye Felisha Spam Blocker program begins with a visual introduction of the name of the application, user registration, and a user agreement that needs to be accepted by the user. The user then enters "Y" in order to accept the policy and proceed using the application. Afterwards, the user is presented with a variety of options to choose from. If the user chooses option 4, the user is directed to the part of the program that checks for spam numbers. The program asks for the user to input a 10-digit number. The program reads through the system of spam numbers and notifies the user if the number they entered is a spam number or not.

Use Case Name: Check for Spam

Actors:

- Parking Customer (Person with a vehicle wanting to park in a designated space)
- Parking Machine (Any parking machine associating a cost for parking in a designated space)

Triggers:

- User chooses option 4 when then user menu is displayed.

Preconditions:

- The user opens the application, registers, and accepts the user agreement.
- The user chooses option 4 in the user menu to check if a number is a spam number.
- User enters a 10-digit number that the system can process.

Post Conditions:

- The system analyzes the user input to check that the entry is a valid input
- The program compares the valid user entry to the numbers in the spam numbers list.
- Depending on whether or not a match was found, the user is notified if the number entered is a spam number or not.
- The user is taken back to the user menu so that a different option can be selected depending on what the user wants to do.

Normal Flow:

1. The user opens the application Bye Felisha Spam Blocker application.
2. The user registers and accepts the user agreement by entering "Y" in order to keep using application.

3. The user is presented with a number of choices and chooses the one that best fits their needs.
4. If the user chooses option 4, the user is presented with the Spam Checker part of the application.
5. System asks the user to enter a 10-digit number. If the input is anything other than a 10-digit number, the program tells the user that the entry is an “Invalid Number”.
6. System reads the user input and compares the input to the list of scam numbers.
7. According to whether or not the number was found in the list, the user is notified if the number is a scam number or not.
8. The system returns to the user menu so that the user can chose a different option from the menu, depending on what they would like to do next.

Use Case Specification

Use Case Description: Allows any user to sign up and register for the ByeFelisha Spam Blocker application. This use case begins as the user starts providing their names and contact information into the application after being introduced and greeted. The user is prompted with the license and user agreement policies and can proceed accordingly after accepting the policies. The program stores the user's phone number and name which is available for verification purposes and prompts the user to procced forward into the program.

Use Case Name: Register

Actors:
- Any user choosing to use ByeFelisha Spam Blocker application.
- ByeFelisha application.

Triggers:
- The user wants to block all the irrelevant spam calls bothering them.
- The user wants to save time by not having to deal with unwanted calls.

Preconditions:
- The program is operational.
- The user has a valid phone number to register.

Postconditions:
- The user has successfully read and understood the privacy policies and license agreements.
- The program has successfully verified the provided phone number and name and stored it respectively.

Normal Flow:
1. The user opens up ByeFelisha Spam Blocker application.
2. The application provides a menu with description for the user to choose.
3. The user selects the appropriate option number.
4. The application greets the user after choosing option 1 from the menu.
5. The application prompts the user for their name and number.
6. The user enters their name and number.
7. The application prompts the user to press 'v' to view the license agreement.
8. The user presses 'v' to proceed.
9. The application displays all the user policies and license agreement associated with the application.
10. The application prompts the user to read carefully and press 'y' to accept.
11. The user presses 'y' to accept and proceed.
12. The application thanks the user for registering for the application.

Use Case: Report Scam Phone Number

Actors: App User

Description: The user enters a number they wish to report as a spam number. Then the user when prompted will enter the number of calls they have received from the potential scam number. If they have received 5 calls or more, then the number will be seriously considered as a potential scam number. The user will then be asked to their basic information: first and last name. The complete summary of the report will be returned to the screen as a confirmation their report was processed. The information will then be stored in a list of objects in which each object contains one report and all the entered information.

Stimulus: The user has received at least five calls of the same number which they believe to be a spam. The user selects option 3 in the main Bye Felisha user menu.

Response: The summary is entered into the list of customer reports in which it is stored.

Comments: If user enters an invalid number, or has received less than 5 calls, they will not be allowed to report and will be returned to the main Bye Felisha menu.

USE CASE SPECIFICATION FOR USE CASE 4 (DATABASE MANAGEMENT)

Use Case Description: allows the application to store spam phone numbers in a database in order for other use cases to make use of it. The database manager will begin generating scam phone numbers by first generating an area code, then concatenating it with a 6-digit code to make a complete phone number, then placing that phone number into the database of the system, and looping through the process again multiple times to have a full list of phone numbers.

Use Case Name: Generate Scam Number Database

Actors:

- Database developer
- Application

Triggers:

- The application needs a database storage for other classes within it to access and make use of.

Preconditions:

- Other classes in the application that require access to the database have to be fully and correctly implemented
- The database developer possesses knowledge and experience of how to create and manage a database

Post conditions:

- The database has been successfully generated with no error
- The generated database has returned to the program to be accessed by other classes

Normal Flow:

1. The database developer creates a plan on how to generate a database consisting of all the scam phone numbers before proceeding
2. The database developer first develops a 3-digit area code
3. The database developer then develops a 6-digit phone code
4. The 3-digit area code will be concatenated with a 6-digit code
5. The database developer loops through this generation of phone number 500 times in order to have a sufficient list of scam phone numbers
6. This list will then be added to the database, in which the database is an array list holding each phone number as a string
7. The database will have been generated and ready for use in other classes

1) **Use Case Specification:**

Release #2: Bye, Felisha Spam Call Blocker

Use Case Description: Allows any spam call blocker (SCB) customer to request a list of blocked calls from their phone. This use case begins when a SCB customer presses the "Display Blocked Calls" button in the Bye, Felisha application user interface. The database manager receives the request and accesses the blocked calls database. The database manager prints out the blocked calls to the user.

Use Case Name: Display Blocked Calls

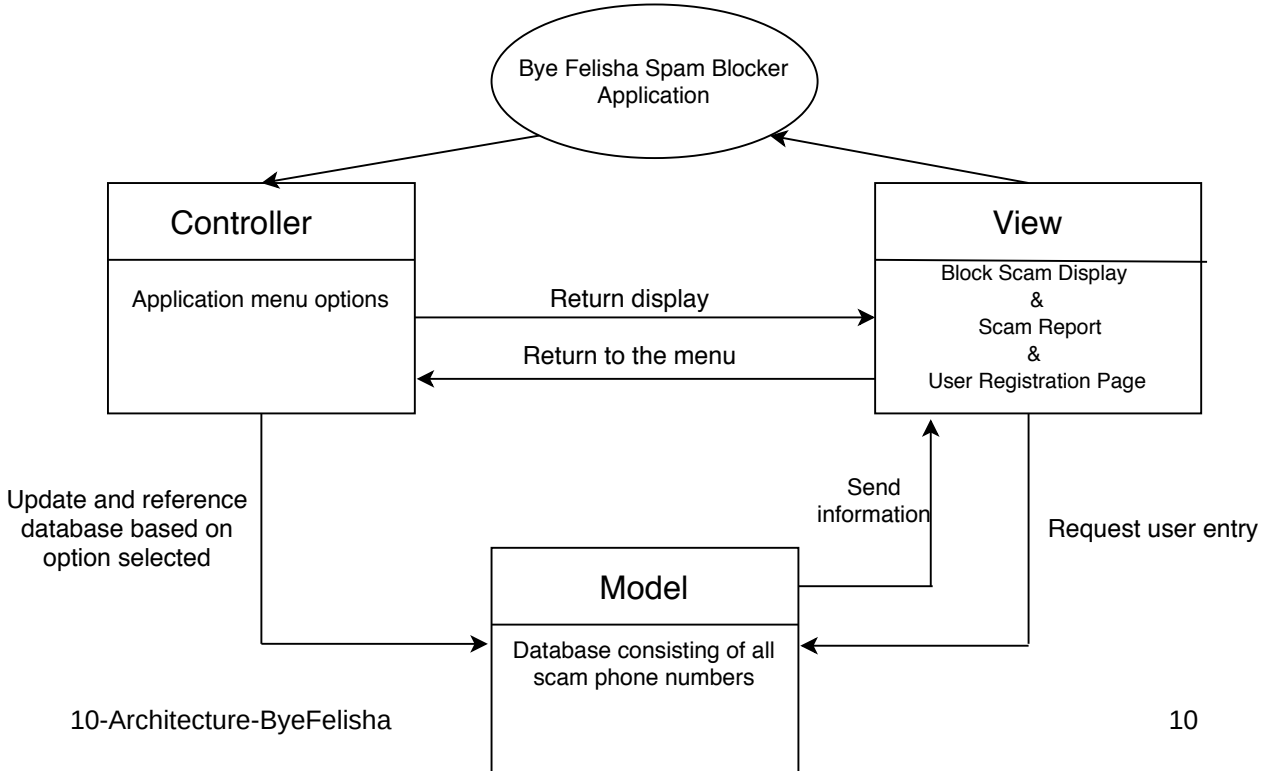
Actors:
- Any spam call blocker customer (someone who downloads the Bye, Felisha application)
- Database Manager

Triggers:
- The user wants to see what phone numbers have been blocked from their phone.

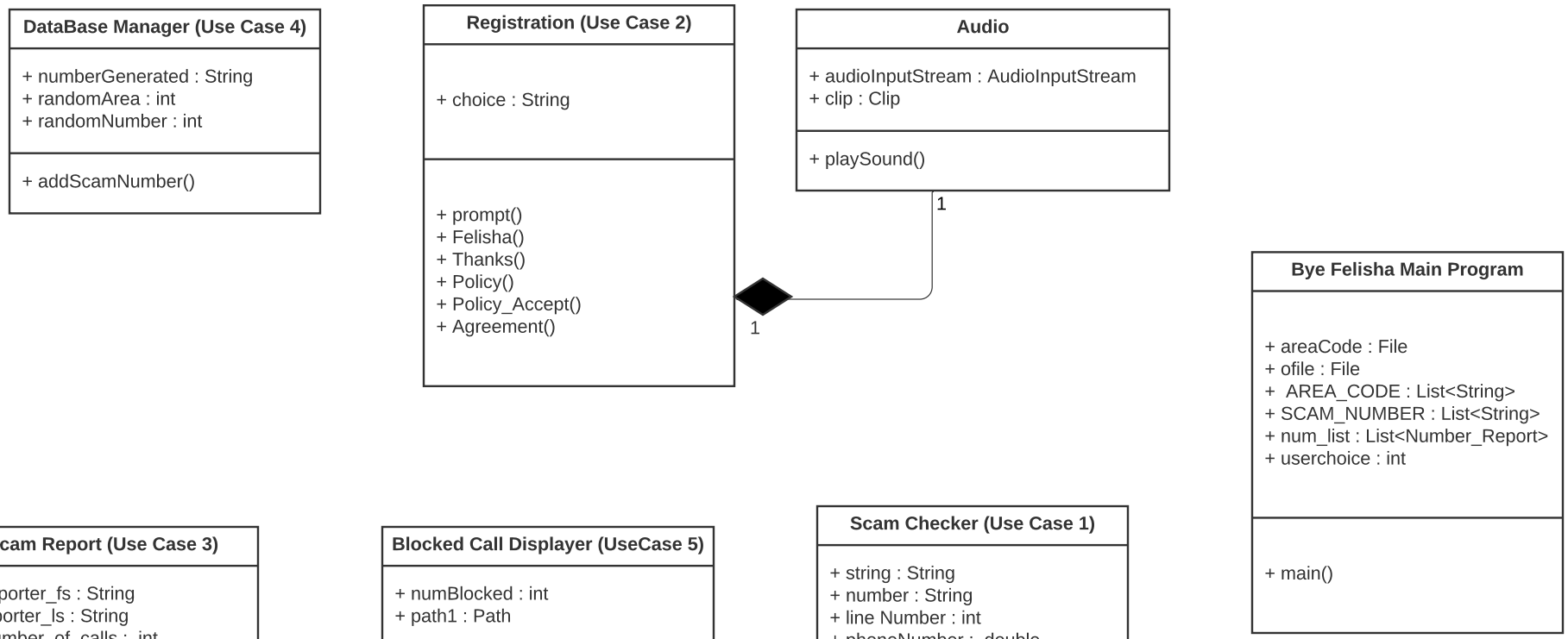
Preconditions:
- The database manager is operational.
- The Bye, Felisha application is operational.
- The user opens the Bye, Felisha application.

Postconditions:
- The spam call blocker customer has viewed their blocked calls.

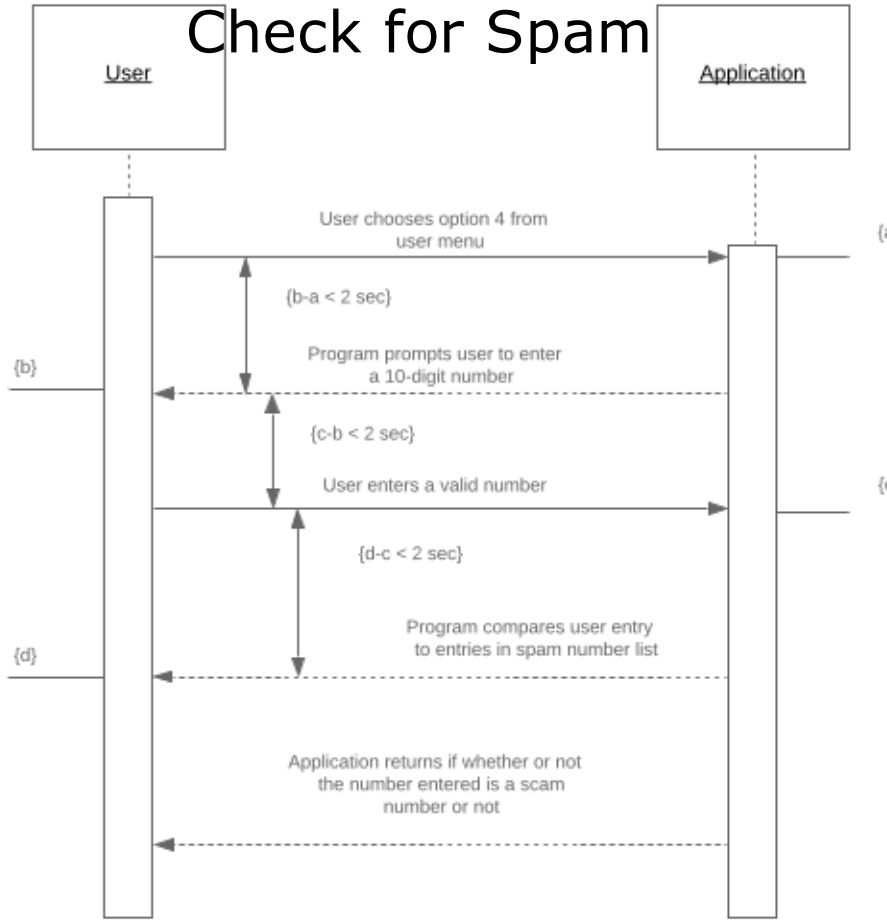
Normal Flow:
1. The customer presses the "Display Blocked Calls" button in the Bye, Felisha application.
2. The database manager verifies there exist blocked calls in the database.
3. The database manager prints the blocked calls to the user screen interface.
4. The customer views and reads the displayed blocked calls in the Bye, Felisha application.

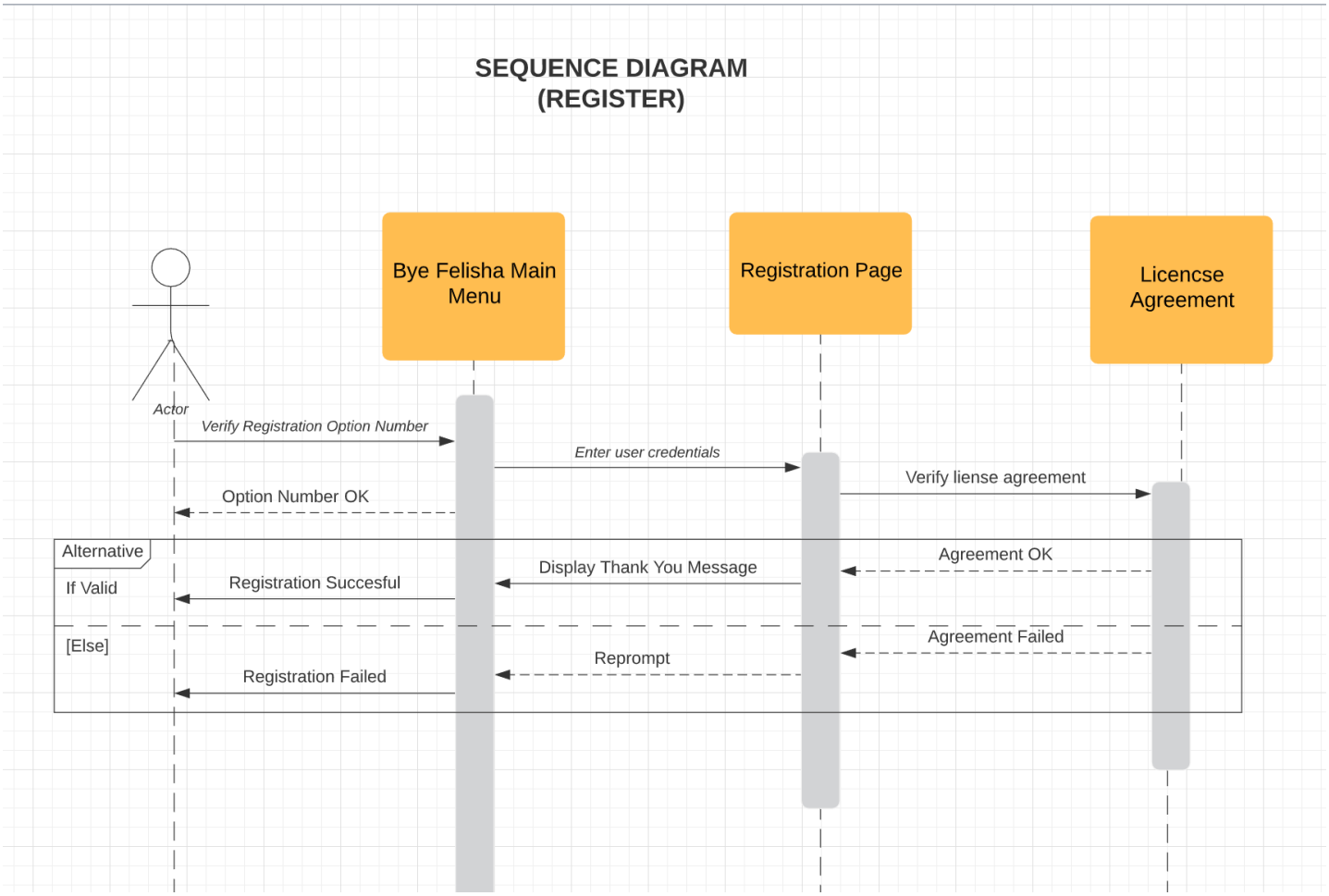

Bye Felisha Spam Blocker Application
Controller
Application menu options
Return display
Return to the menu
View
Block Scam Display
&
Scam Report
&
User Registration Page
Update and reference database based on option selected
Send information
Request user entry
Model
Database consisting of all scam phone numbers

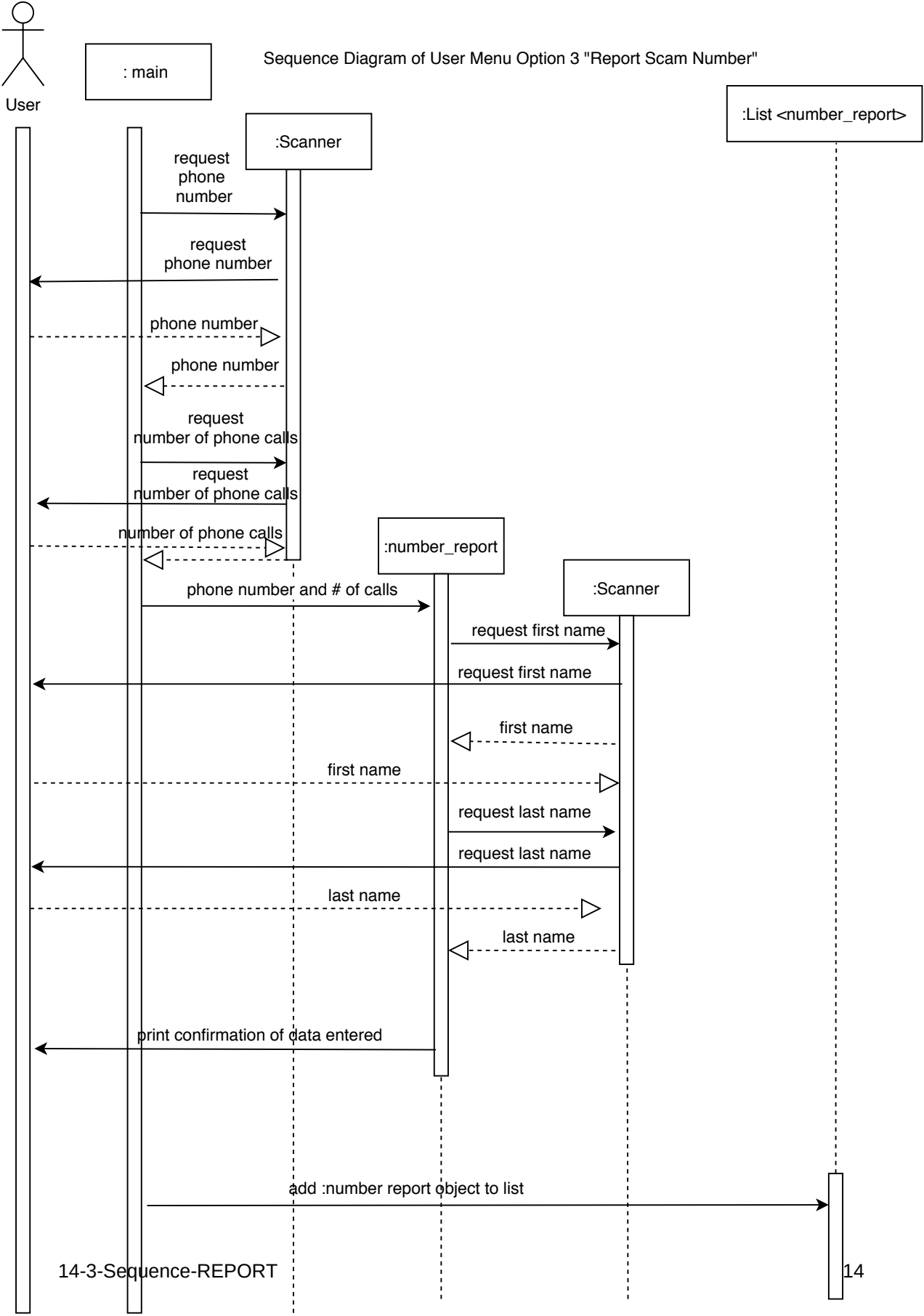
CLASS DIAGRAM



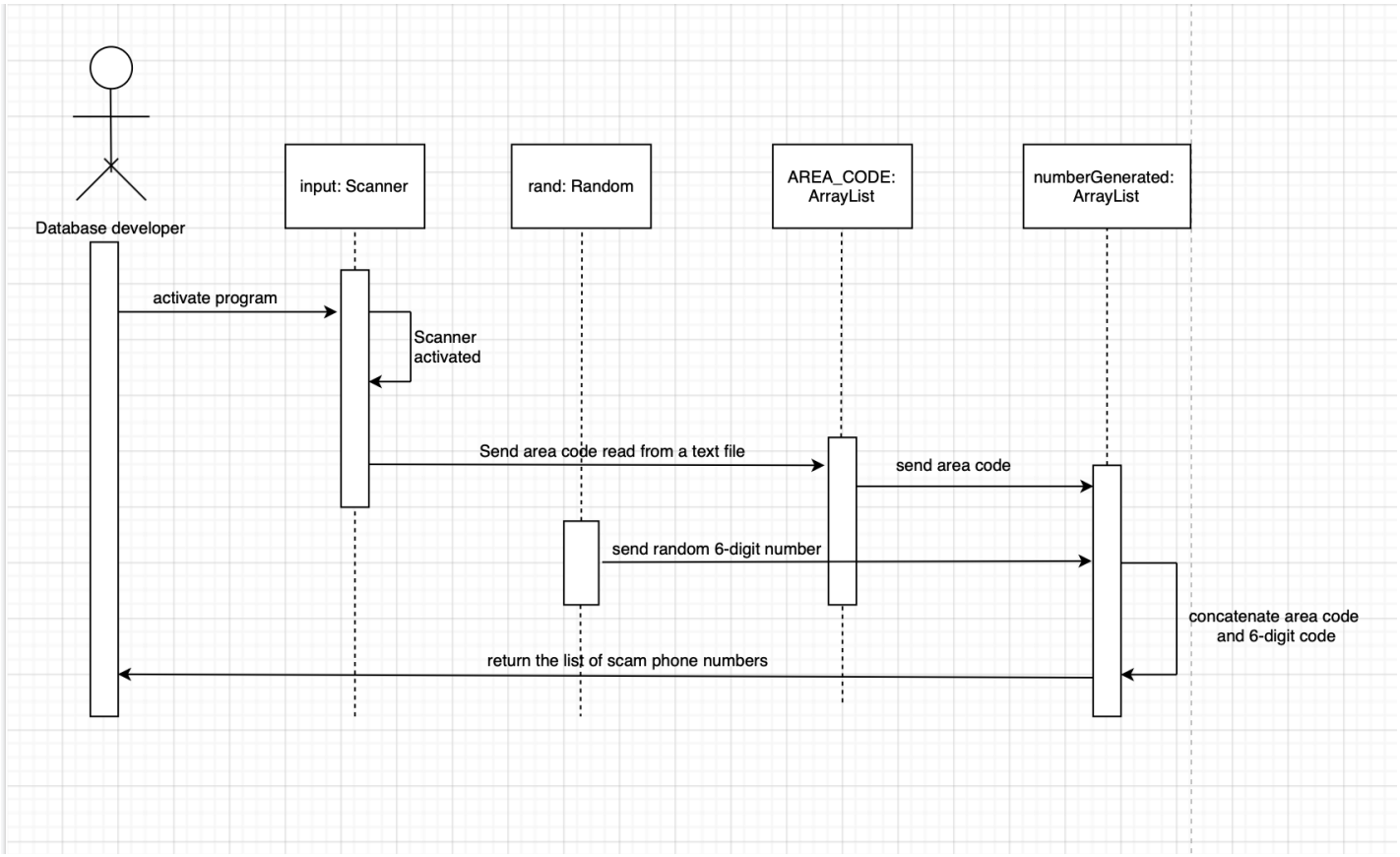
Sequence Diagram- Check for Spam

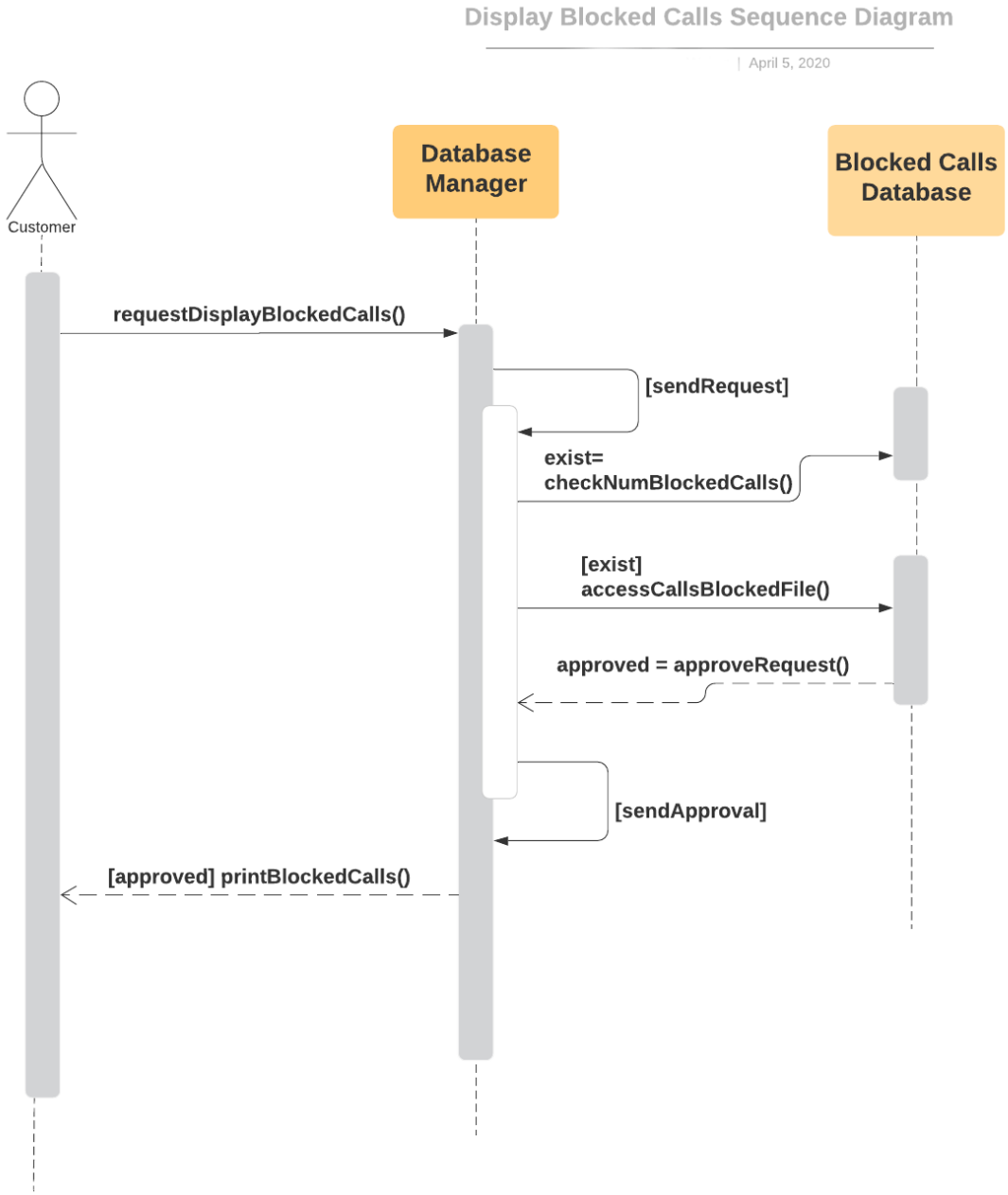


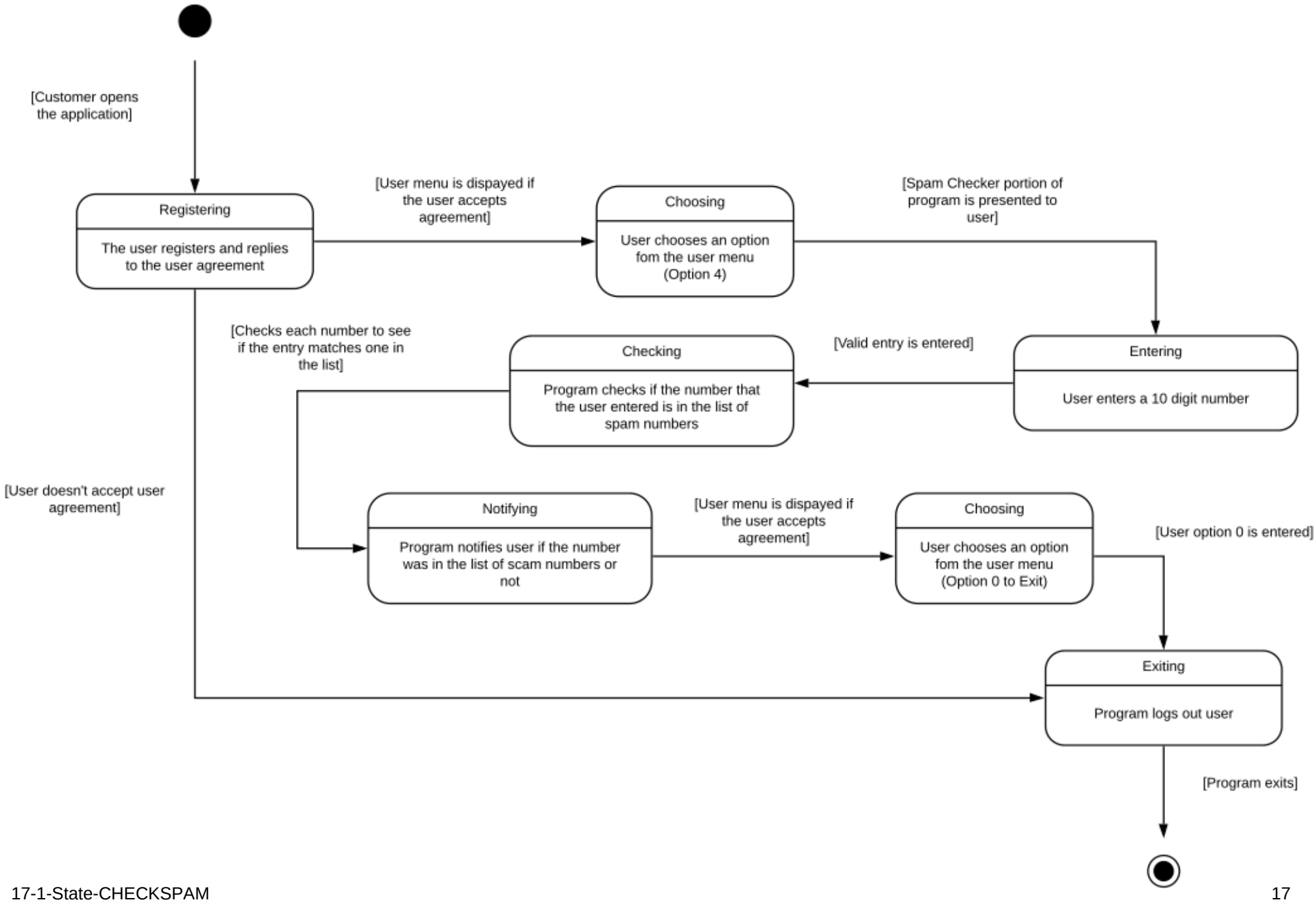

SEQUENCE DIAGRAM
(REGISTER)
Bye Felisha Main Menu
Registration Page
Licencse Agreement
Actor
Verify Registration Option Number
Enter user credentials
Verify liense agreement
Option Number OK
Alternative
If Valid
Registration Succesful
Display Thank You Message
Agreement OK
[Else]
Agreement Failed
Reprompt
Registration Failed

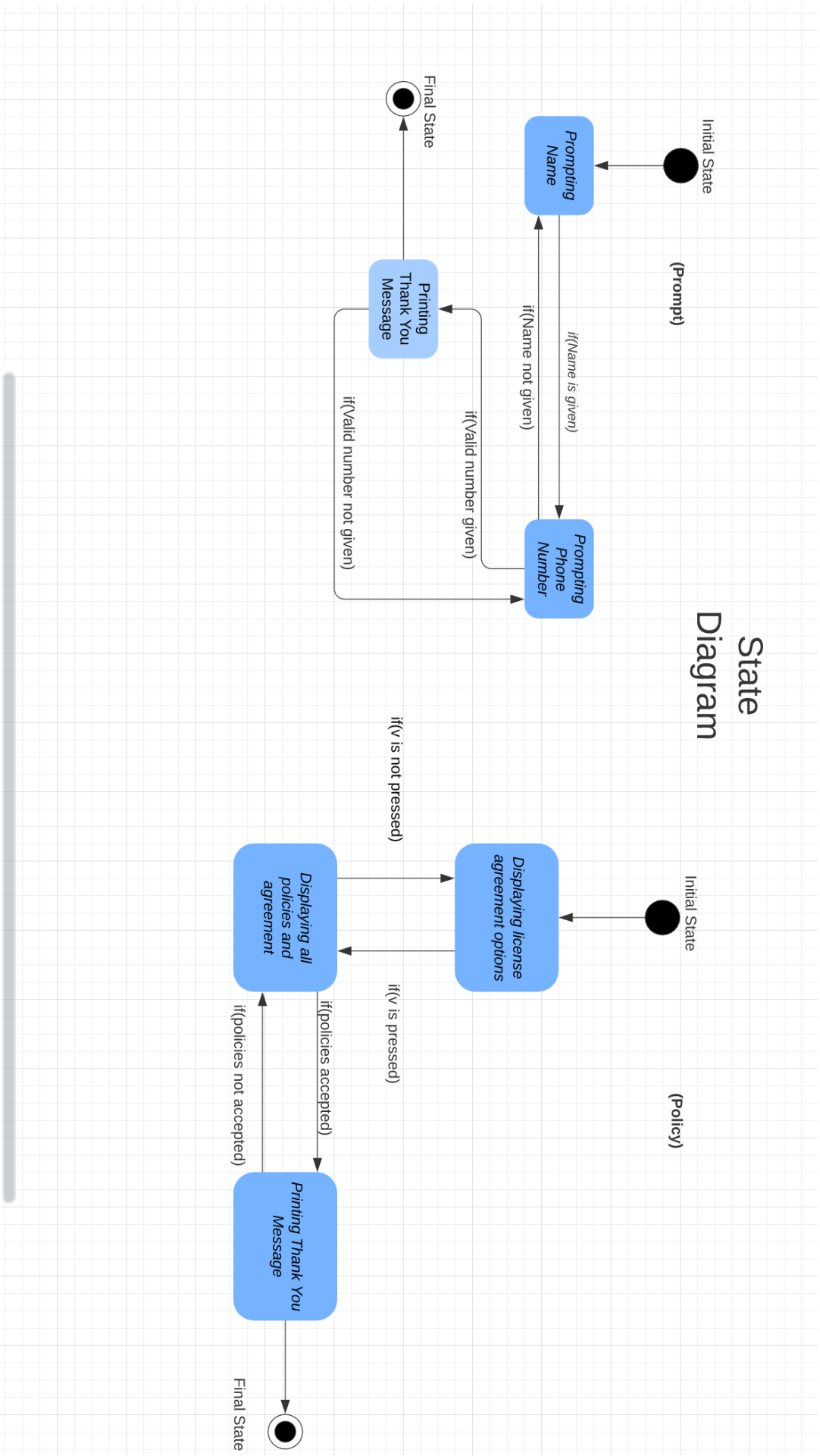

Sequence Diagram of User Menu Option 3 "Report Scam Number"
User
: main
:Scanner
:List <number_report>
request phone number
request phone number
phone number
phone number
request number of phone calls
request number of phone calls
number of phone calls
:number_report
:Scanner
phone number and # of calls
request first name
request first name
first name
first name
request last name
request last name
last name
last name
print confirmation of data entered
add :number report object to list

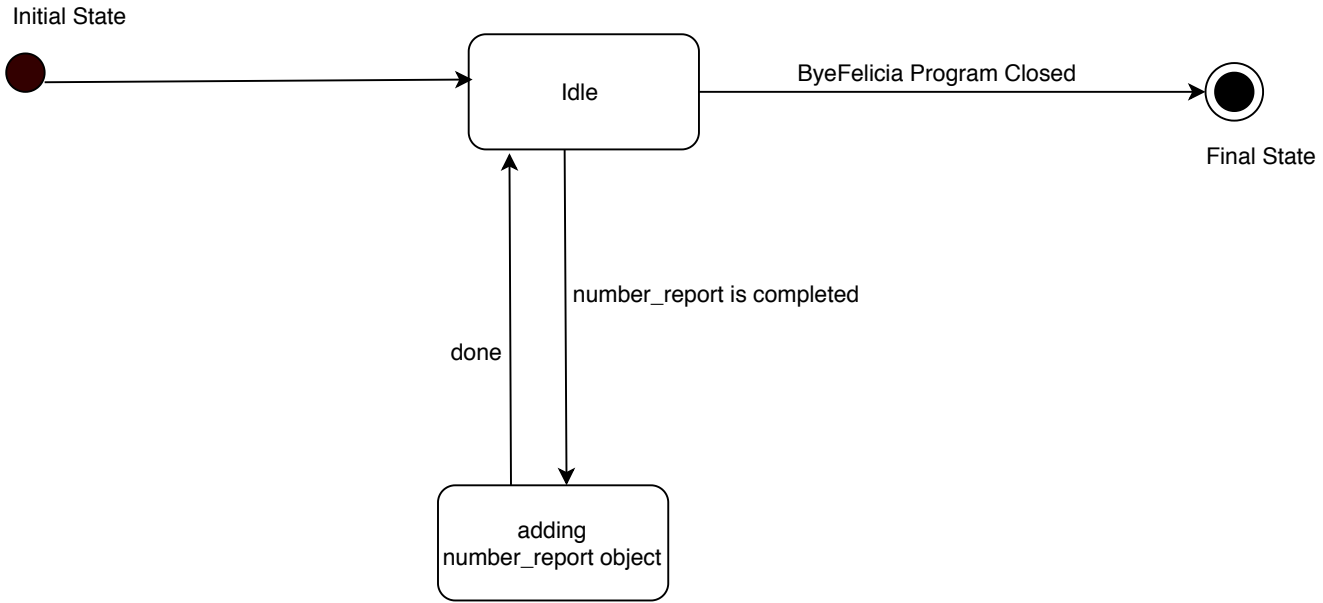
SEQUENCE DIAGRAM FOR CREATING A DATABASE



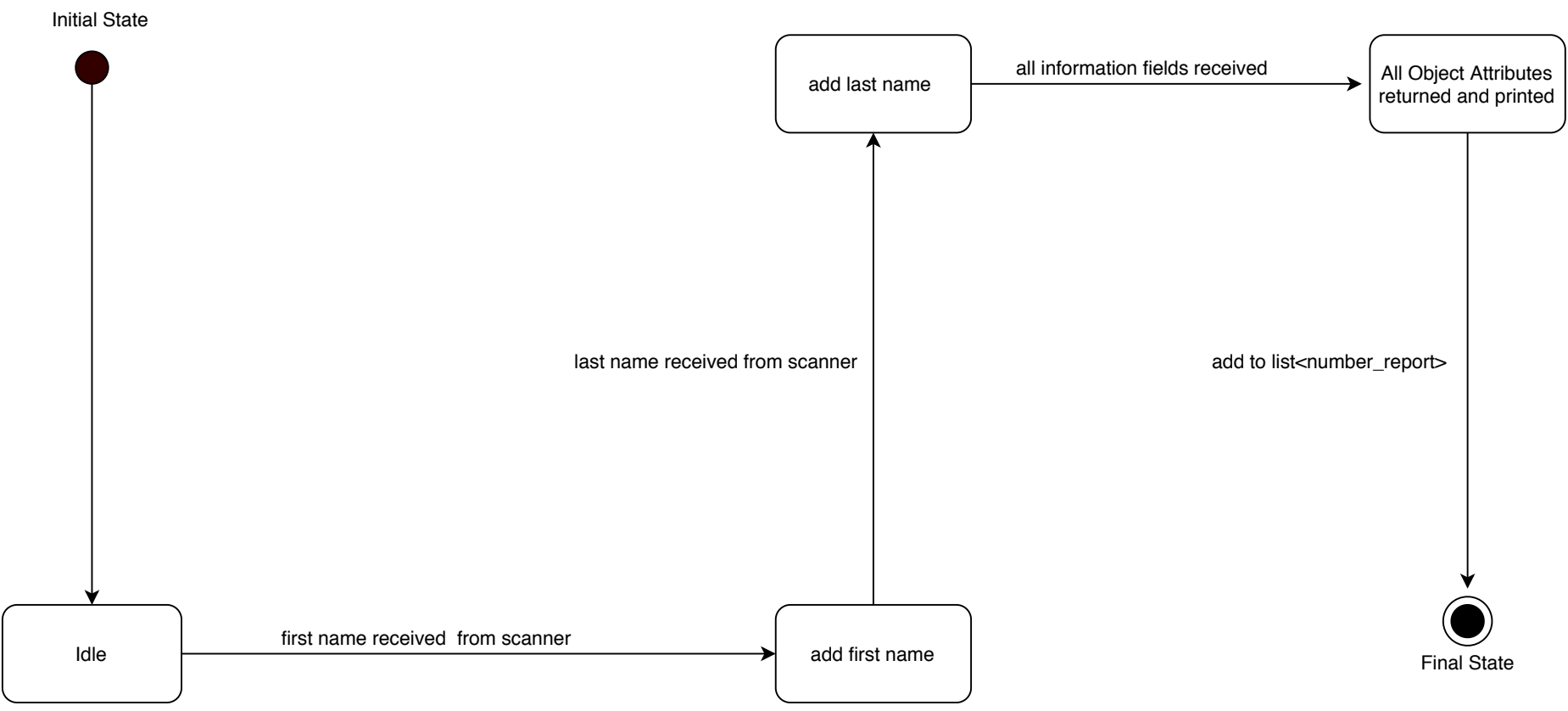

Display Blocked Calls Sequence Diagram
April 5, 2020
Customer
Database Manager
Blocked Calls Database
requestDisplayBlockedCalls()
[sendRequest]
exist= checkNumBlockedCalls()
[exist] accessCallsBlockedFile()
approved = approveRequest()
[sendApproval]
[approved] printBlockedCalls()


[Customer opens the application]
Registering
The user registers and replies to the user agreement
[User menu is dispayed if the user accepts agreement]
Choosing
User chooses an option fom the user menu (Option 4)
[Spam Checker portion of program is presented to user]
Entering
User enters a 10 digit number
[Valid entry is entered]
Checking
Program checks if the number that the user entered is in the list of spam numbers
[Checks each number to see if the entry matches one in the list]
[User doesn't accept user agreement]
Notifying
Program notifies user if the number was in the list of scam numbers or not
[User menu is dispayed if the user accepts agreement]
Choosing
User chooses an option fom the user menu (Option 0 to Exit)
[User option 0 is entered]
Exiting
Program logs out user
[Program exits]


State Diagram
Initial State
(Prompt)
Prompting Name
if(Name is given)
if(Name not given)
Prompting Phone Number
if(Valid number given)
if(Valid number not given)
Printing Thank You Message
Final State
Initial State
(Policy)
Displaying license agreement options
if(v is pressed)
if(v is not pressed)
Displaying all policies and agreement
if(policies accepted)
if(policies not accepted)
Printing Thank You Message
Final State

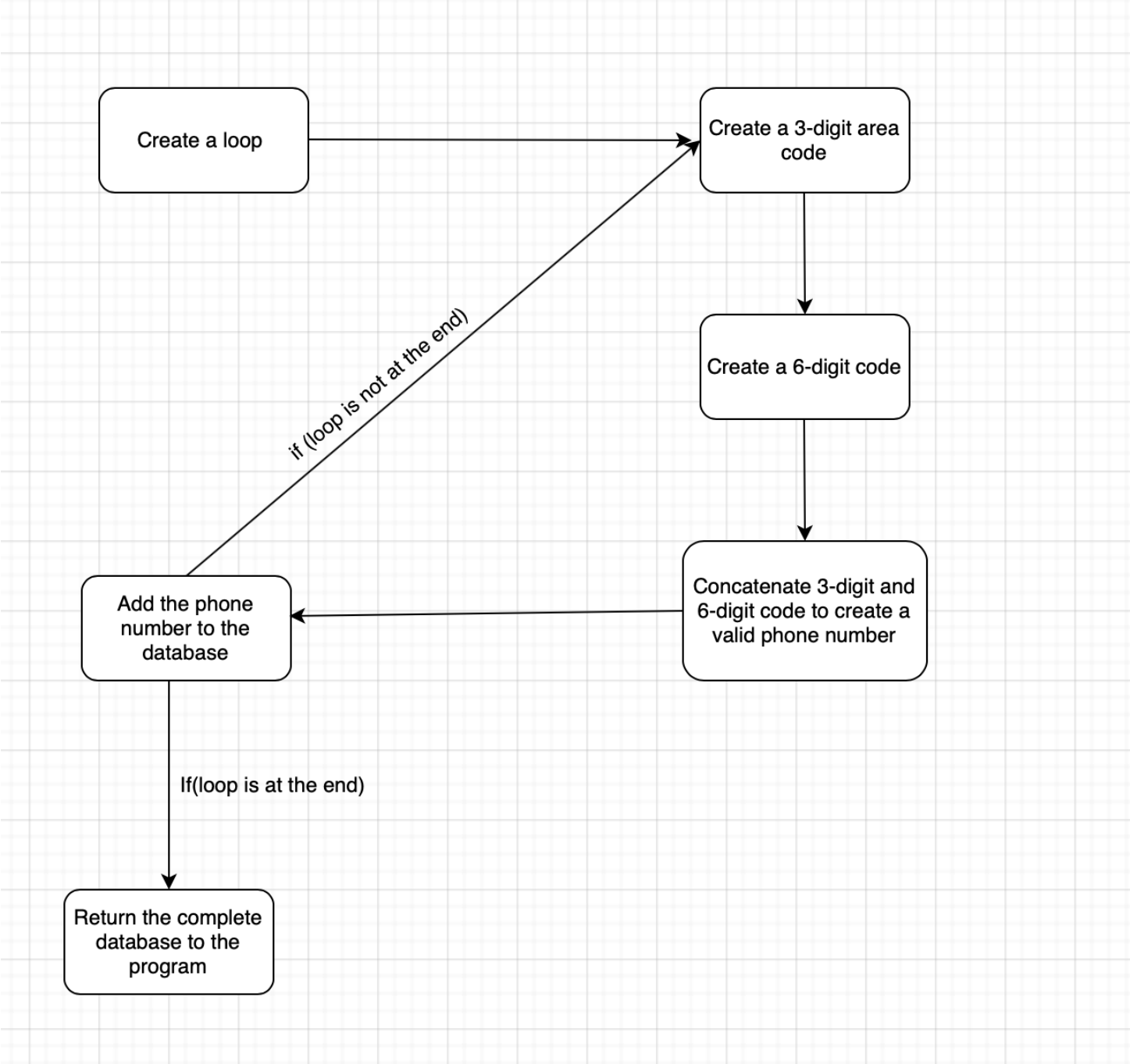


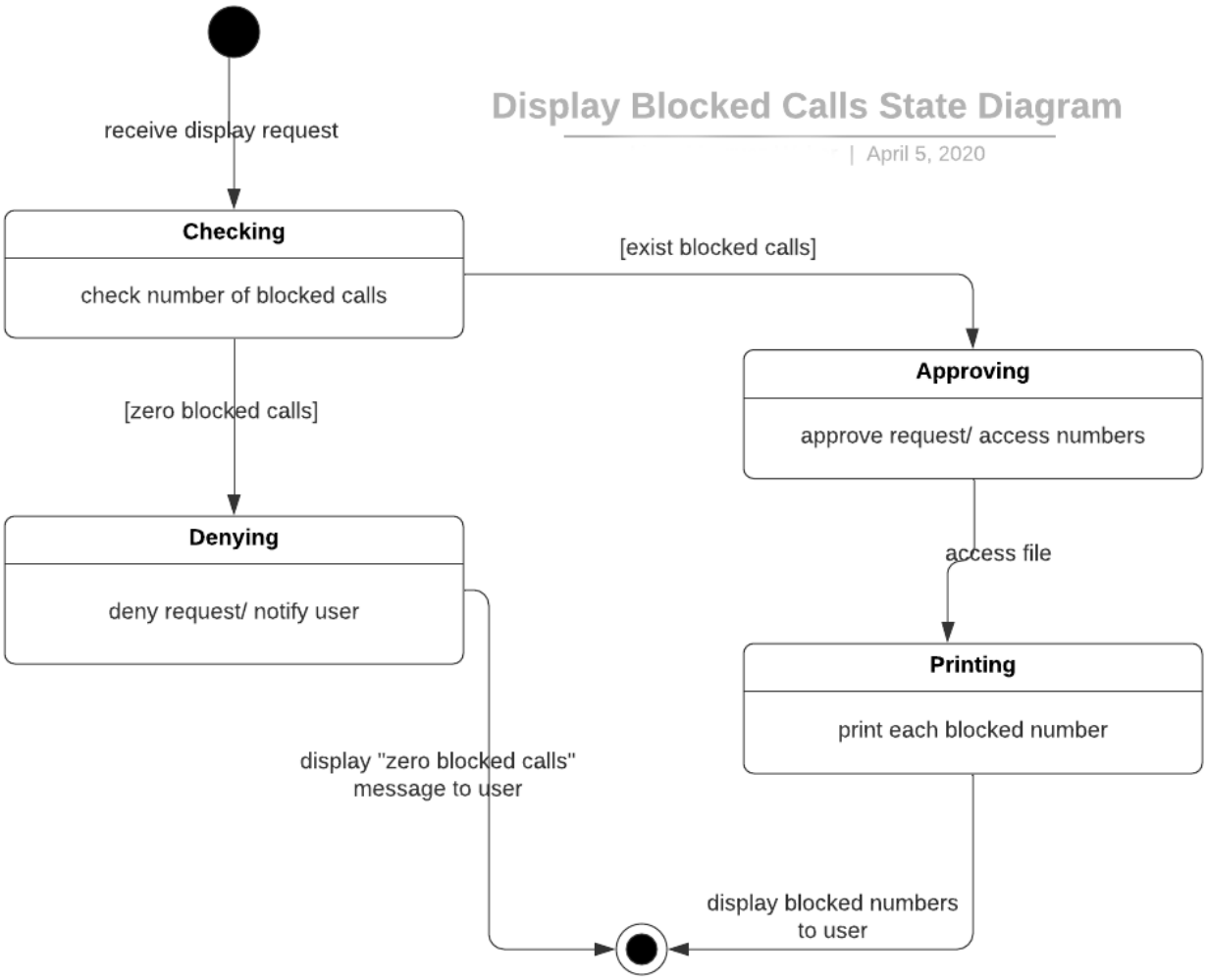
State Diagram: Use Case 3 "Report Scam Phone Number" - Object ArrayList<>



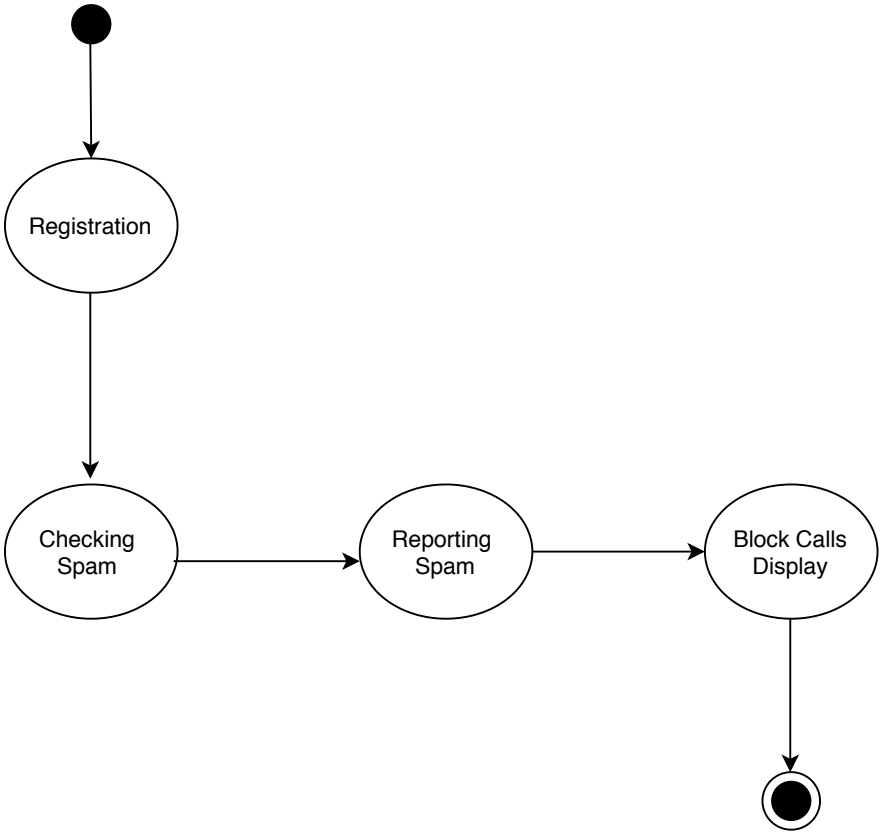
State Diagram: Use Case 3 "Report Scam Phone Number" - Object number_report

STATE DIAGRAM FOR CREATING A DATABASE




Display Blocked Calls State Diagram
| April 5, 2020
receive display request
Checking
check number of blocked calls
[exist blocked calls]
[zero blocked calls]
Approving
approve request/ access numbers
Denying
deny request/ notify user
access file
Printing
print each blocked number
display "zero blocked calls"
message to user
display blocked numbers
to user

Bye Felicia Spam Blocker - Activity Diagram



THANKS!