

Detailed Design:  
*IdentiFisher*  
An Android Application

McDonald, Christopher  
1312456

Guo, Tian  
1327833

Murray, Shandelle  
1303109

Cheung, Ocean  
1316057

Last Edited On: March 28, 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	System Description . . . . .	3
1.3	Overview . . . . .	3
<b>2</b>	<b>State Charts for Controller Classes</b>	<b>3</b>
<b>3</b>	<b>Sequence Diagrams</b>	<b>5</b>
<b>4</b>	<b>Detailed Class Diagram</b>	<b>9</b>
<b>A</b>	<b>Division of Labour</b>	<b>10</b>

Revision 0: This is the first draft written from the authors listed on the Title page.

# 1 Introduction

## 1.1 Purpose

This document is the third, and final documentation which outlines the details of the identification application *IdentiFisher*. This document specifically will provide ample detail into the components of the application so that a person with the appropriate expertise could implement it. These details will include the processes of the application, the life cycle of each component and what is expected or required from them.

This document is crucial for the project managers and programmers as there will be heavy detail on how the application will execute. However, it is also important for all stakeholders to read this document because of their involvement in the success of the application. These stakeholders should be making sure the application is aligning with the goals and aspirations it had always intended, and provide guidance when necessary.

## 1.2 System Description

This system is an Android Application, running on the most up to date Android operating system, *Marshmallow*. It will provide the service of identifying the type of fish a User is describing, when given a required set of criteria. Additional features included tracking the fish that users has caught with the geolocation they are at as well as providing warnings to users if the fish is potentially dangerous or endangered. This requires text input from the user, as well as GPS functionality from the smartphone. Given the information we are holding, we also provide the service to users to look up data for each geolocation they provide. This application will be very easy to use, as fishing can involve varying level of intensity and should be available in all of these times.

## 1.3 Overview

This documents contains varying levels of detailed design components the implementation must have. The first section to note is Section 2 which is the State Charts for Controller Classes. This will be a state machine to describe the action, outputs and states for each Controller class in the application. For reference, the Controller classes can be found in the High Level Architecture Design document. These will be useful to outline how the system acts when given an input during a particular time. Continuing from there in Section 3 is the Sequence Diagrams. This section is a major aspect of the life cycle of the application in its entirety and also the sub components of the application. This diagram will show when processes are started, when they end and what messages or actions are sent between them. This part is crucial to the synchronous aspects of the application. Lastly in Section 4 is the Detailed Class Diagram which will take each Class required for the implementation of the program and go into great detail about it. This will include the inputs, resources, expectations and outputs of each class. Moreover, the attributes and methods each class will need should be outlined as well.

# 2 State Charts for Controller Classes

In this section, a state chart for each control class will be displayed. The state chart will demonstrate how each classes behave as a result of respective inputs. Note that the Expert classes were combined into one chart as they behave similarly.

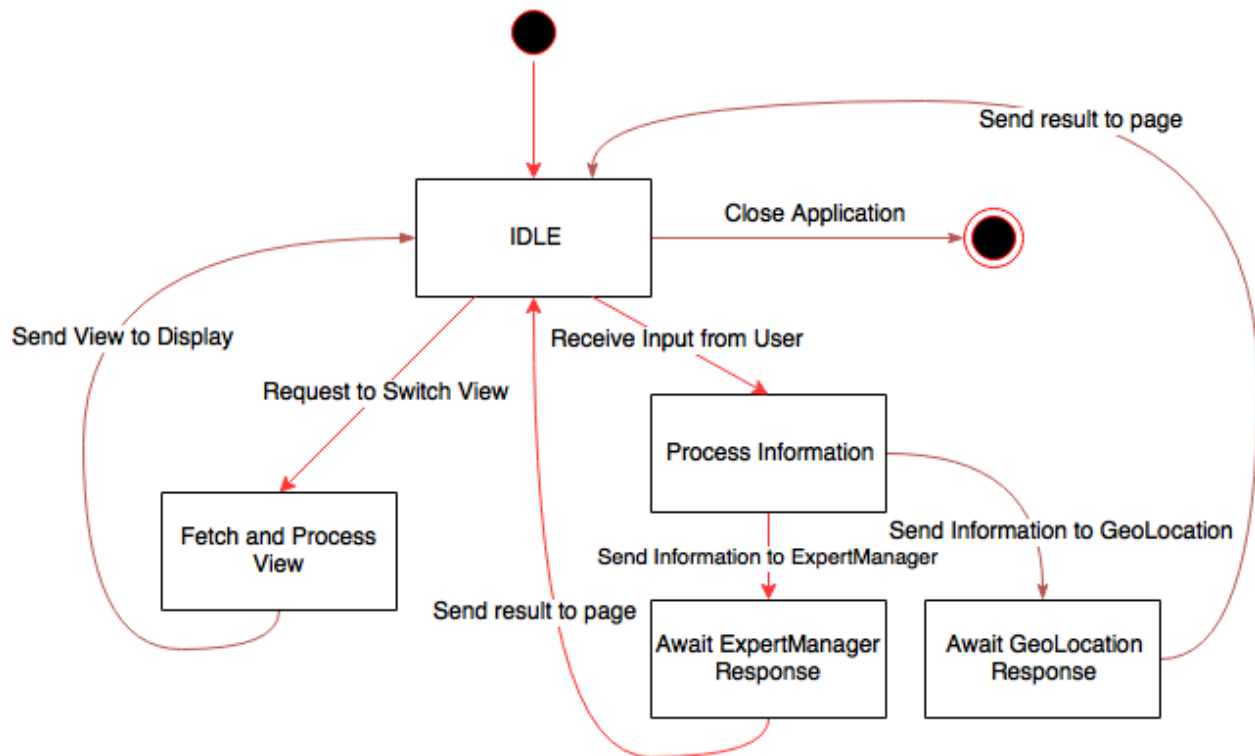


Figure 1: State Chart: ViewController

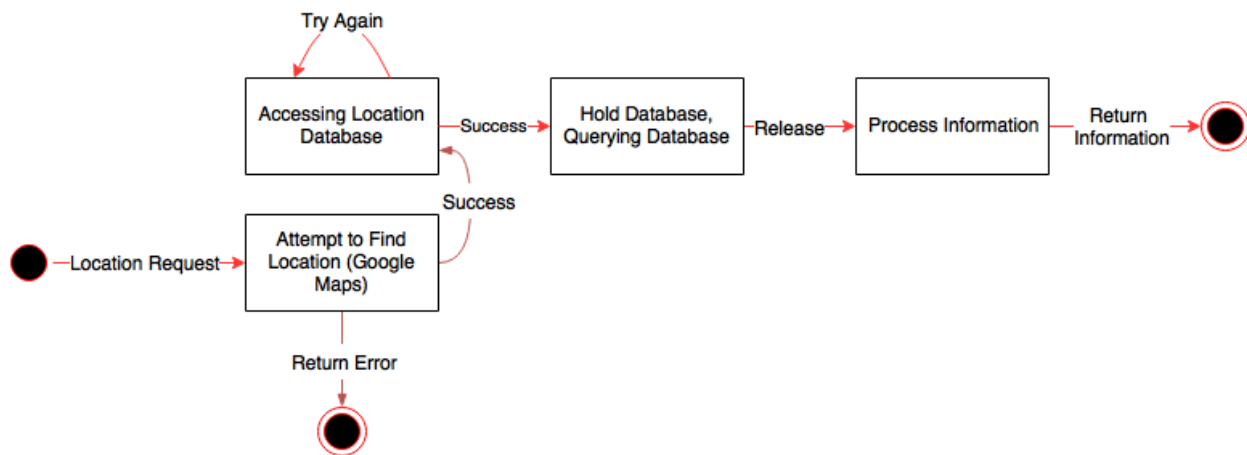


Figure 2: State Chart: GeoLocation

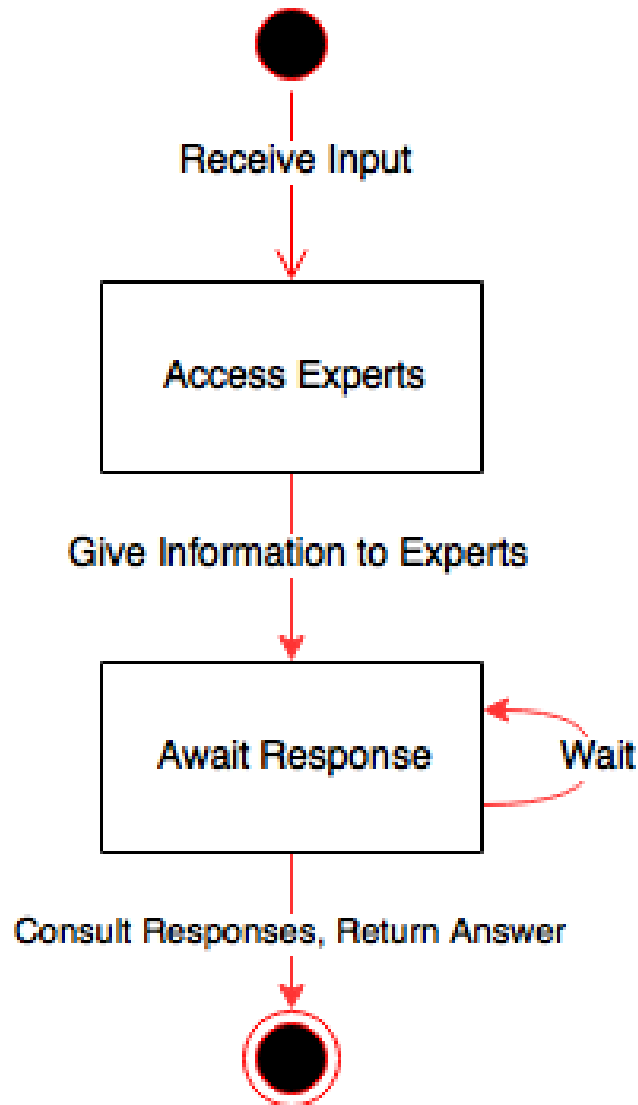


Figure 3: State Chart: Expert Manager

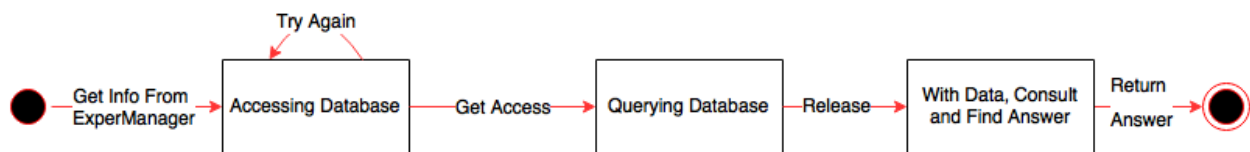


Figure 4: State Chart: Expert

### 3 Sequence Diagrams

The following sequence diagrams depict use cases of the IdentiFisher application. The first sequence diagram shows the process of a user requesting and the application providing an identification of a fish based on user input. The second sequence diagram shows the user obtaining fish population statistics of a specific lake or body of water. The third sequence diagram shows the user adding a self-identified fish to the database

which is tentative until the database administrator confirms the submission. The fourth sequence diagram shows the database administrator updating expert details, which may include adding, deleting, or modifying an expert.

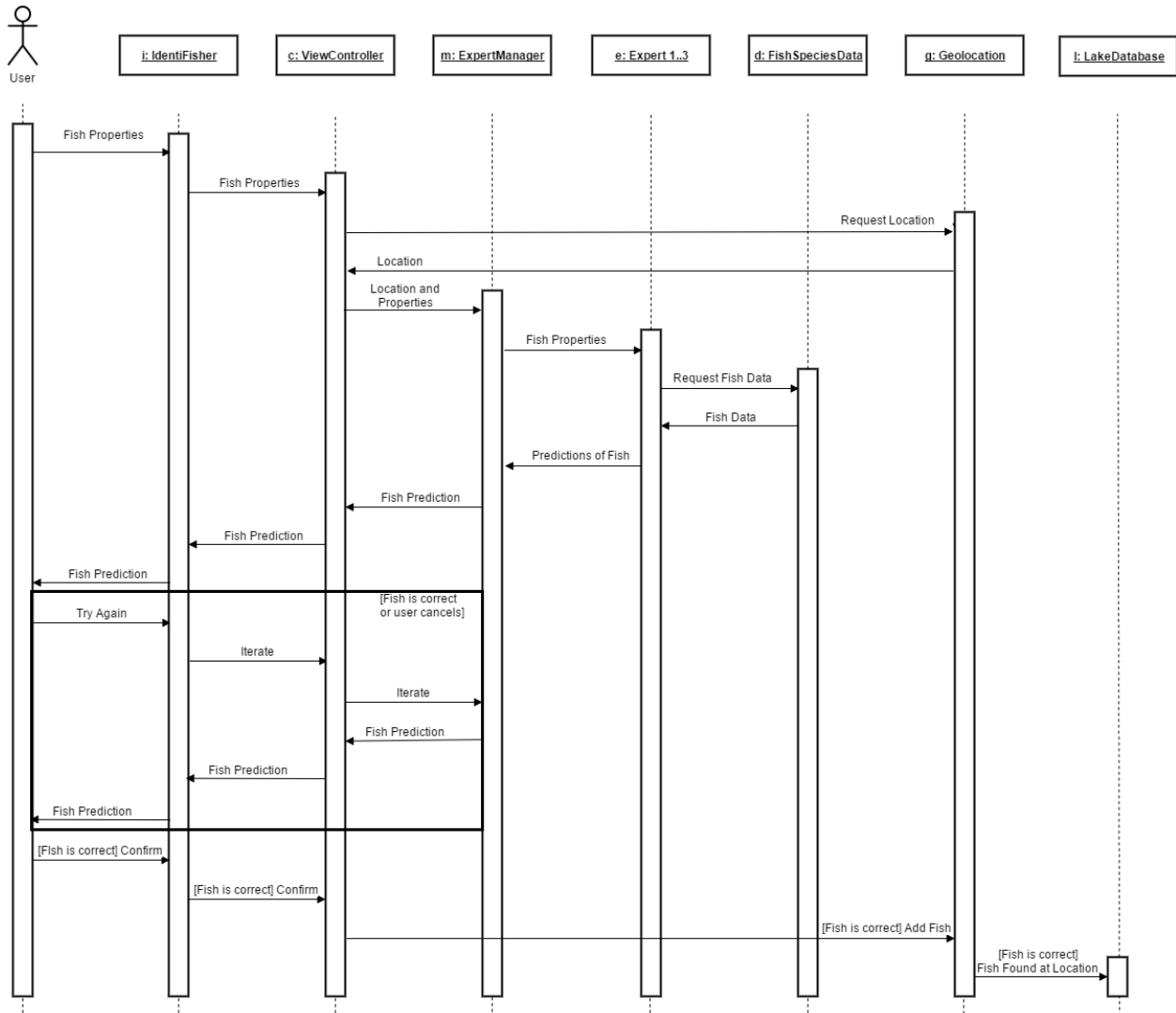


Figure 5: Sequence Diagram: Fish Identification

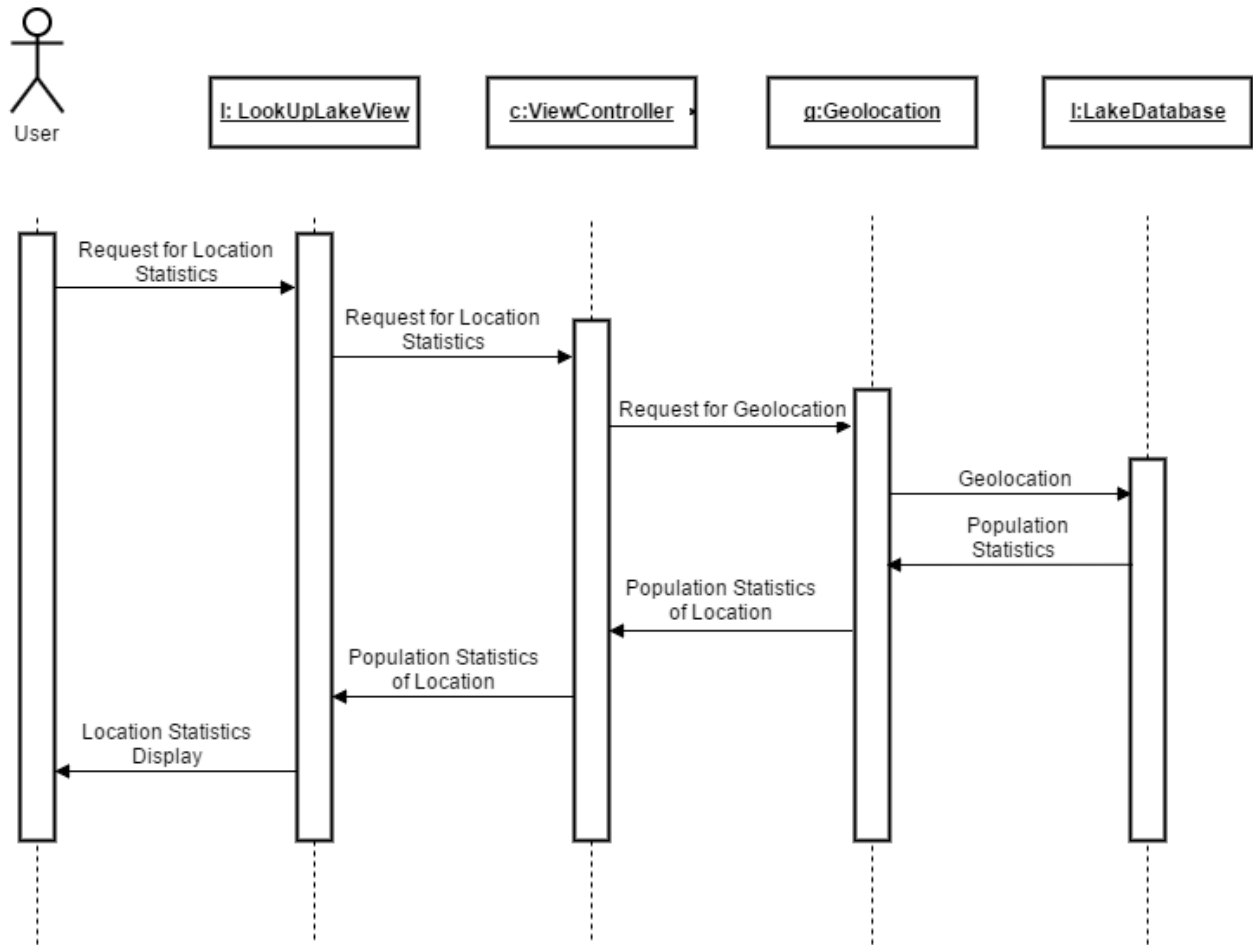


Figure 6: Sequence Diagram: Look up Body of Water Statistics

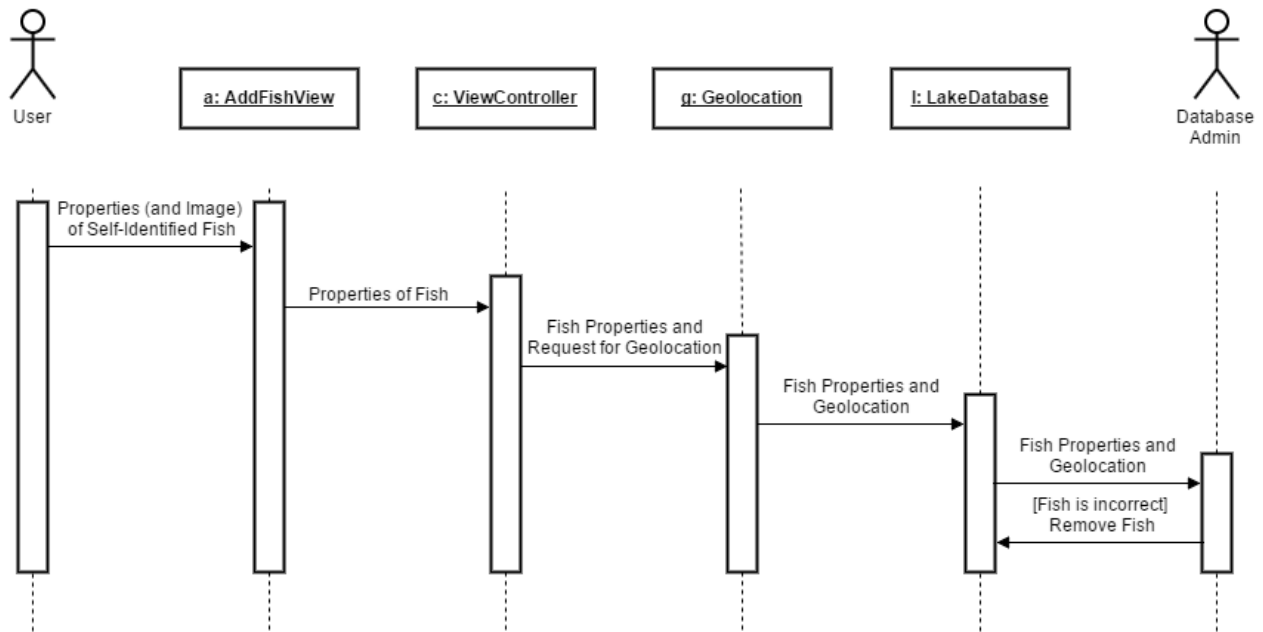


Figure 7: Sequence Diagram: Add Self-Identified Fish to Database

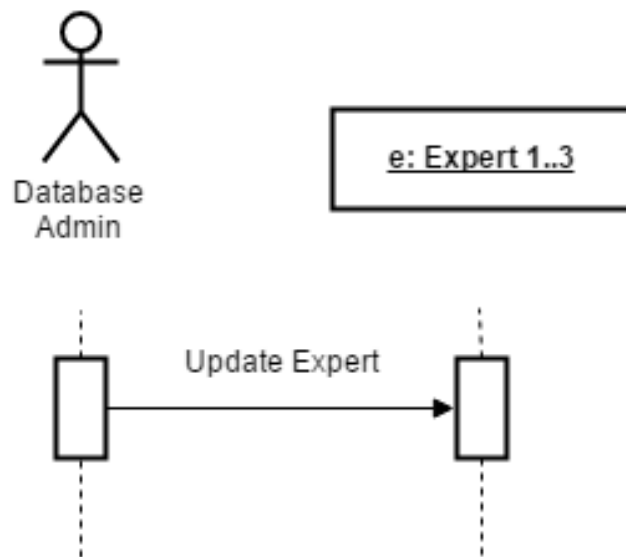


Figure 8: Sequence Diagram: Update Experts



## 4 Detailed Class Diagram

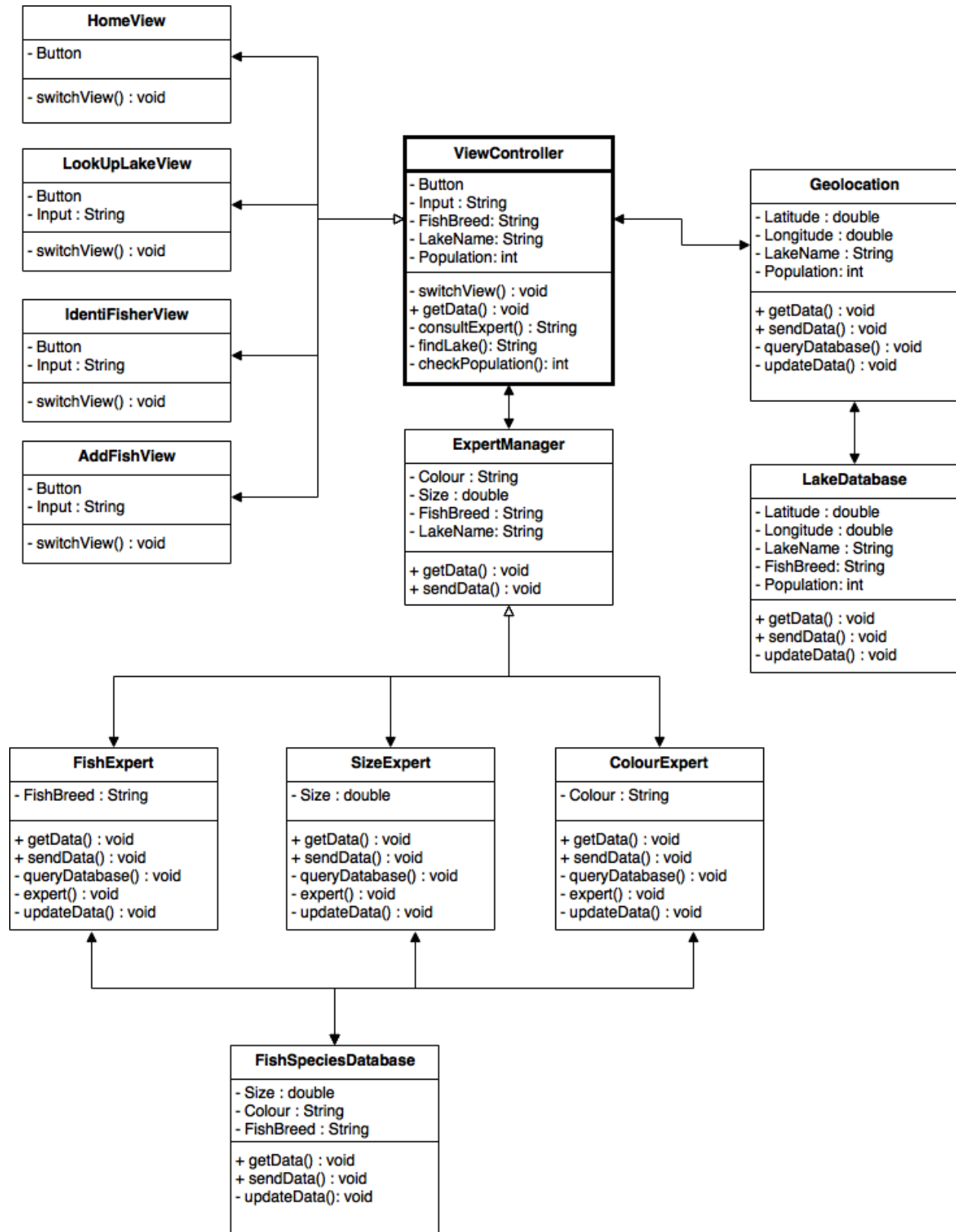


Figure 9: Detailed Class Diagram

## List of Figures

1	State Chart: ViewController . . . . .	4
2	State Chart: GeoLocation . . . . .	4
3	State Chart: Expert Manager . . . . .	5
4	State Chart: Expert . . . . .	5
5	Sequence Diagram: Fish Identification . . . . .	6
6	Sequence Diagram: Look up Body of Water Statistics . . . . .	7
7	Sequence Diagram: Add Self-Identified Fish to Database . . . . .	8
8	Sequence Diagram: Update Experts . . . . .	8
9	Detailed Class Diagram . . . . .	9

## A Division of Labour

Name	Labour	Signature
Shani	Sequence Diagrams	
Chris	Introduction	
Ocean	Detailed Class Diagram	
Tian	State Charts for Controller Classes	