High Level Architecture Design:
IndentiFisher
An Android Application

McDonald, Christopher
1312456

Guo, Tian
1327833

Murray, Shandelle
1303109

Cheung, Ocean
1316057

Last Edited On: March 4, 2016

Contents

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

1 Introduction

This section will give a detailed description of what to expect from the entire document. This will clear any assumptions, prior knowledge and any other improtant information that the reader should now prior to reading this document.

1.1 Purpose

a) The purpose of this document is to present the high-level architectural design of the IdentiFisher Android application introduced in the Software Specification Requirements document. This will be accomplished using various diagrams and textual descriptions.

b) The intended audience for this document includes any stakeholders involved in the project or interested in the application. This document is especially intended for any person on the project development team who will have a role in the design and implementation of the application and may also include investors, managers, or future users of the application who wish to see the high-level design of the application.

1.2 System Description

The system described in this document is called IdentiFisher, an Android application intended to be used by beginner to experienced fishers. IdentiFisher accepts user input about a fish that they have caught and attempts to identify the type based on specific details about the physical appearance and geolocation of the fish. The application will interface with an online mapping system in order to obtain geolocational information and will maintain a data collection of fish caught in specific locations in order to generate catch-rate and other statistics that will be available to users of the application upon request.

1.3 Overview

The beginning of this document has introduced the purpose of the document and has given a brief outline of IdentiFisher, the system being designed. The subsequent sections will go into detail about the uses and high-level design of the application. First of all, a Use Case Diagram will be presented with a description of the uses of the application. Next, an Analysis Class Diagram will be included in order to show the general organization of the application's classes. The fourth section will provide an overview of the architectural design of the application, including a Structural Architecture Diagram as well as a description of any subsystems. The fifth section will be comprised of Class Responsibility Collaboration (CRC) Cards that will outline the responsibilities and collaborators for each class identified. Finally, a Division of Labour section is included in this document in order to identify each author along with the portion so the document that they have completed.

2 Use Case Diagram

This section should provide a use case diagram for your application.

a) Each use case appearing in the diagram should be accompanied by a text description.

3 Analysis Class Diagram

This section should provide an analysis class diagram for your application.

4 Architectural Design

This section will outline a detailed description of the whole architecture system. This will incude, but is not limited to, the name of the design, benefits of the design as well as a justification of the choice and enough detail to implement the design.

4.1 System Architecture

This system will mainly consist of front-end modules, processors and entities. The front-end modules will consist of every view of the Android application. They will handle all the input from the User and output from the application. After accepting valid input from the User, the application will send that data to the processors. The processors will consist of the Experts and ExpertManager. They will process this information while querying the entities for information. The entities will consist of the Database used to store all information regarding Users and Fish. After the processors are finished with the Entities, the ExpertManager will return the processed information to the front-end modules to be displayed. The front-end modules will then display the information to the User and wait for more input. Due to this process, we will be using a Black Board Architecture Style. The reasoning behind the choice is that all the Experts are contributing to a single and complex problem. With a Black Board style, all Experts can contribute as much as possible so the ExpertManager can make the best decision. Within the system, the ExpertManager will play the role of both the Blackboard and Controller and the Experts will play the role of the knowledge source.

ADD DRAW.IO DRAWING

4.2 Subsystems

This section serves the purpose of breaking each subsystem down and outlining the actions, responsiblites and expectations of each subsystem. Ordered with respect to level of abstraction, the following sections represent broad groupings each class will fall into. The following groups are as follows.

4.2.1 Front End Modules

These modules are majorly responsible for the I/O of the application. Without these being functional, it would be impossible for the User to interact with our application. It consists of all views in the Android Manifest, and every element each view contains. It will preform minor data verification, provide ease of communication and display an attractive, functional interface. The whole system depends on these modules to provide valuable, correct information so that it can process the information accordingly.

4.2.2 Processors & Controllers

These modules are majorly responsible for the relay, delivery and preocessing of all the information going through the application. Without these being functional, the front-end modules would have no way of communicating to the data sources. It will preform the majority of the data verification, querying the database and formatting the data returned.

4.2.3 Data Source

These modules are majorly repsonsible for holding all of the information in the application. This is necessary in order to use any information given by the User. It will preform organization, maintaining and intrgrity of the information.

5 Class Responsibility Collaboration (CRC) Cards

Class Name: Home	
Responsibility:	**Collaborators:**
This Class is part of the Front End Modules and carries all the responsibilities associated with that. These include handling all of the information coming from the User and displaying information to the User. This class in particular will be a hub to direct the User to the other Front End Modules. This class should be welcoming, as it is the first thing the User will interact with.	It will collaborate with all the other Front End Modules. This is because it will provide the means for getting to the other modules. These include Settings, Look Up Lake and IdentiFisher.

Class Name: Settings	
Responsibility:	**Collaborators:**
This Class is part of the Front End Modules and carries all the responsibilities associated with that. These include handling all of the information coming from the User and displaying information to the User. This class in particular will be a hub to change any settings the User has that they have permissions for.	It will collaborate with the other Front End Modules and the Database. It needs to provide a means for getting to the other modules and change any settings that exist in the Database.

Class Name: LookUpLake
Responsibility:
This Class is part of the Front End Modules and carries all the responsibilities associated with that. These include handli

---

Class Name: IdentiFisher	
Responsibility:	**Collaborators:**

Class Name: ExpertManager	
Responsibility:	**Collaborators:**

Class Name: Forum	
Responsibility:	**Collaborators:**

Class Name: Expert	
Responsibility:	**Collaborators:**

Class Name: Database	
Responsibility:	**Collaborators:**

A Division of Labour

Name	Labour	Signature
Shani	Introduction I	
Chris	Class Responsibility Charts & Architecture Design	
Ocean	Analysis Class Diagram	
Tian	Use Case Diagram	