Software Specification Requirements:
IdentiFisher
Group 7

McDonald, Christopher
1312456

Guo, Tian
1327833

Murray, Shandelle
1303109

Cheung, Ocean
1316057

Taylor, James
000000

February 7, 2016

Contents

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

1 Introduction

1.1 Purpose

The purpose of the SRS is to provide a detailed account of all the expected functions and requirements of the software system. It will go into detail regarding the system as a whole, who we expect to use it, and any relevant information one would need to endorse or build the system. Lastly, we will outline both the functional and non-functional requirements of the project that are necessary for the system's success. The intended audience of this document includes any stakeholders that are involved in this project. This could include, but is not limited to, the investors, developers, managers, marketers and human resource workers. Every person who is an entity in the aforementioned list should take an interest in the details outlined hereafter to ensure that every person has a clear idea of what the software system should do.

1.2 Scope

The software system will be named hereafter as IdentiFisher, which is an Android application. This system will be a utility application for anyone who fishes, either recreationally or competivitely. It will service beginner to experienced fishers. Identifisher will allow the user to give information about a recently caught fish and help to identify what type of fish it is. From there, it can collect data and track which types of fish are caught where. The aim is to build a global logging system that will provide percentage catch rates by lake, educate young, novice fishers, and integrate technology into a relatively non-technological field.

1.3 Definitions, Acronyms, and Abbreviations

Definition:

- IdentiFisher - The application being referenced in this document.

Acronyms:

- API - Application Program Interface
- OS - Operation System
- SDK - Software Development Kit
- GPS - Global Positioning System

Abbreviations:

1.4 References

1.5 Overview

Thus far a very brief overview has been provided of the IdentiFisher application, its intended use, and what we expect a typical user would be. Going forward, we will go into deeper detail regarding those topics and more. The next section, the overall description, will give far more information regarding the application and some of the external matters regarding the system. After that, functional requirements will be listed, with non-functional requirements making up the last section of this document.

2 Overall Description

2.1 Product Perspective

The IdentiFisher application is similar to other Android applications that, by user request, analyze textual input or images in order to identify an entity. It is independent as it is not intended to be used as part of a larger system; however, it will interface with an online mapping system in order to perform geolocational functions.

2.2 Product Functions

The IdentiFisher application will:

- interface with an online mapping system in order to determine the geolocation of the user
- be able to access a collection of data related to catch rate statistics per body of water
- determine and display reasonable predictions about the type of fish the user has described
- allow the user to input textual or pictorial data representing a specific fish
- request an educated prediction of the type of fish described
- request statistical information such as catch rate about fish in a specific location

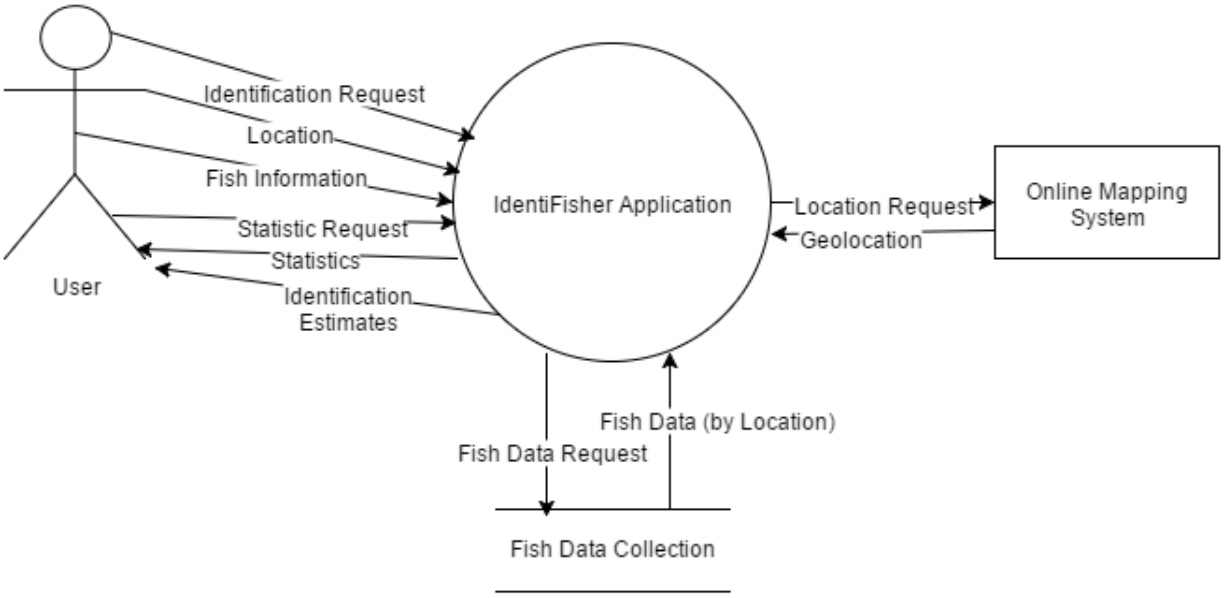

Figure 1: Context Diagram of IdentiFisher Application

2.3 User Characteristics

- The IdentiFisher application is intended to be used by beginner to experienced fishers who wish to identify the type of fish they have caught or access geographical catch rate statistics.
- Technologically, an intended user of the application should have access to a mobile device on which they are capable of installing and accessing an application, establishing an internet connection, as well as generating and inputting textual and pictorial data.

2.4 Constraints

- Since the statistics about a lake are generated by actual users of the application, the method by which data is collected must be subject to strict integrity constraints. The application must employ a method of data verification before adding it to the statistics available to other users of the application.
- The application's functionality is constrained by the OS/Software framework through which the application is being written, which is Android/SDK/Java.
- Although the application does not have any notable budget constraints, there are time constraints to be considered. This document, the Software Requirements Specification, must be complete by February 8, 2016. The

application must have been thoroughly tested and be fully functional by April 3th, 2016.

2.5 Assumptions & Dependencies

- An important assumption is that a user of the application is inputting data which represents a real fish and not data that is fake or replicated from another source. This includes inputting images from the internet or simply inputting details about a fish that the user has not actually caught.
- It is assumed that the user has access to a GPS-enabled mobile device with an internet connection through which they may access this application while they are at the location where they have caught the fish they wish to identify.

2.6 Apportioning of Requirements

3 Functional Requirements

Business Event #1:
Description: The User begins to search for a particular fish.
Persons Involved: The User, Google Maps
Requirements:

- The application must be able to access the user's geolocation.
- The application must accept inputs from the user about the physical specifications of a fish and return an output of what breed of fish the user might have.
- The application must be able to return the most probable fish given the inputs of the User.
- The application must allow the user to submit their catch results from fishing.

Business Event #2:
Description: The User requests lake statistics.
Persons Involved: The User, Google Maps
Requirements:

- The application must recieve up to date information regarding each specified lake.
- The application must be able to access the user's geolocation.
- The application must show the User the statistics for each lake specified.

Business Event #3:
Description: The User requests to add a caught fish to the system.
Persons Involved: The User, Google Maps
Requirements:

- The application must be able to access the user's geolocation.
- The application must be able to show accepted fish per lake.
- The application must recieve choice of fish from the User.
- The application must allow the user to submit their catch results from fishing.

4 Non-Functional Requirements

4.1 Look and Feel Requirements

4.1.1 Apperance Requirements

The Identifisher application must have an attractive design and look professional with a 70% satisfaction rate among users concerning the appearance of the application.

4.1.2 Style Requirements

The Identifisher application must have a simple design to promote usability. This is to ensure a small learning curve for users.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

The IdentiFisher application must be easy to use by people of ages 10 and above.

4.2.2 Personalization and Internationalization Requirements

The IdentiFisher application must have English language support.

4.2.3 Learning Requirements

The IdentiFisher application must be easy to learn by people of ages 10 and above. Users should take no longer than five minutes in order to familiarize themselves with the application.

4.2.4 Understandability and Politeness Requirements

The IdentiFisher application must contain no offensive language.

4.2.5 Accessibility Requirements

The IdentiFisher application must be usable by at least 95% of people with acceptable vision.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

The IdentiFisher application must respond to basic user input within less than 5 seconds.

4.3.2 Safety-Critical Requirements

The IdentiFisher application must be safe to use with a 0% mortality rate.

4.3.3 Percision or Accuracy Requirements

The IdentiFisher application must produce correct identifications 75% of the time provided the input is reasonably dependable.

4.3.4 Reliability and Availability Requirements

The IdentiFisher application must be operational and responsive 100% of the time, except during maintenance or update procedures.

4.3.5 Robustness or Fault-Tolerance Requirements

The IdentiFisher application must produce correct identifications 60% of the time in high-noise environments. This could include ambigous attributes, low light pictures, and small resolution images.

4.3.6 Capacity Requirements

The IdentiFisher application must be able to be used for 1 image or query at a time, for more than 95% of users.

4.3.7 Scalability or Extensiblity Requirements

The IdentFisher application must be designed such that multiple types of inputs can be added with ease.

4.3.8 Longevity Requirements

The IdentiFisher application must be available to acquire and remain operational for at least three years.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

The IdentiFisher application will exclusively be running on the Android OS which is on a smartphone.

4.4.2 Requirments with Interfacing with Adjacent Systems

The IdentiFisher application must recieve over a 95% approval rate from the online mapping system API.

4.4.3 Productization Requirements

The IdentFisher application must be able to be uploaded to the Android Play Store. It must also be able to be installed from it with no user interaction after the initial prompt.

4.4.4 Release Requirements

The IdentiFisher application must be ready to be released by April 3, 2016.

4.5 Maintainability and Support Requirements

4.5.1 Maintainence Requirements

The IdentiFisher application must be easily maintained for the time it stays operational.

4.5.2 Supportability Requirements

The IdentiFisher application must have technical support for all of the users.

4.5.3 Adaptiblity Requirements

The IdentiFisher application must adapt to changes and updates in the Android OS and other dependencies.

4.6 Security Requirements

4.6.1 Access Requirements

The IdentiFisher application will be available to all users which satisfy the Android Play Store requirements to download applications.

4.6.2 Integrity Requirements

The IdentiFisher application must encrypt data transferred from the modules of the design to ensure integrity.

4.6.3 Privacy Requirements

The IdentiFisher application must adhere to the principle of least privilege and not access information regarding the user that it does not need.

4.6.4 Audit Requirements

The IdentiFisher application does not have any Audit Requirements.

4.6.5 Immunity Requirements

The IdentiFisher application must store a back-up copy of information regarding which fish were caught in which lake in the event that it is corrupted.

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

The IdentiFisher application must not be offensive towards any culture.

4.7.2 Political Requirements

The IdentiFisher application must not have any political affiliation or influence.

4.8 Legal Requirements

4.8.1 Complicance Requirements

The IdentiFisher application or APIs being used must not go against andy laws or regulations in the country it operates in.

4.8.2 Standards Requirements

The IdentiFisher application does not have Standards Requirements.

List of Figures