Software Specification Requirements:
IdentiFisher
Group 7

McDonald, Christopher
1312456

Guo, Tian
1327833

Murray, Shandelle
1303109

Cheung, Ocean
1316057

Taylor, James
000000

February 5, 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 is any shareholders that are involved in this project. This could include, but is not limited to; the investors, developers, managers, marketers or human resource workers. Every person which is an entity in the aforementioned list should all take an interest in the details outlined hereafter to ensure 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 also will service novice 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 what fish are caught where. We hope 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-technology 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 Developement Kit
- GPS - Global Positioning System

Abbreviations:

1.4 References

1.5 Overview

Thus far we have given a very brief overview 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 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 being the last section of this document.

2 Overall Description

2.1 Product Perspective

The IdentiFisher application is similar to other 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 allow the user to input textual or pictorial data representing a specific fish, request an educated prediction of the type of fish described, and request statistical information about fish in a specific location. Secondly, the application must interface with an online mapping system in order to determine the geolocation of the user. The application must also be able to access a collection of data related to fish population statistics. Finally, the application must determine and display reasonable predictions about the type of fish the user has described.

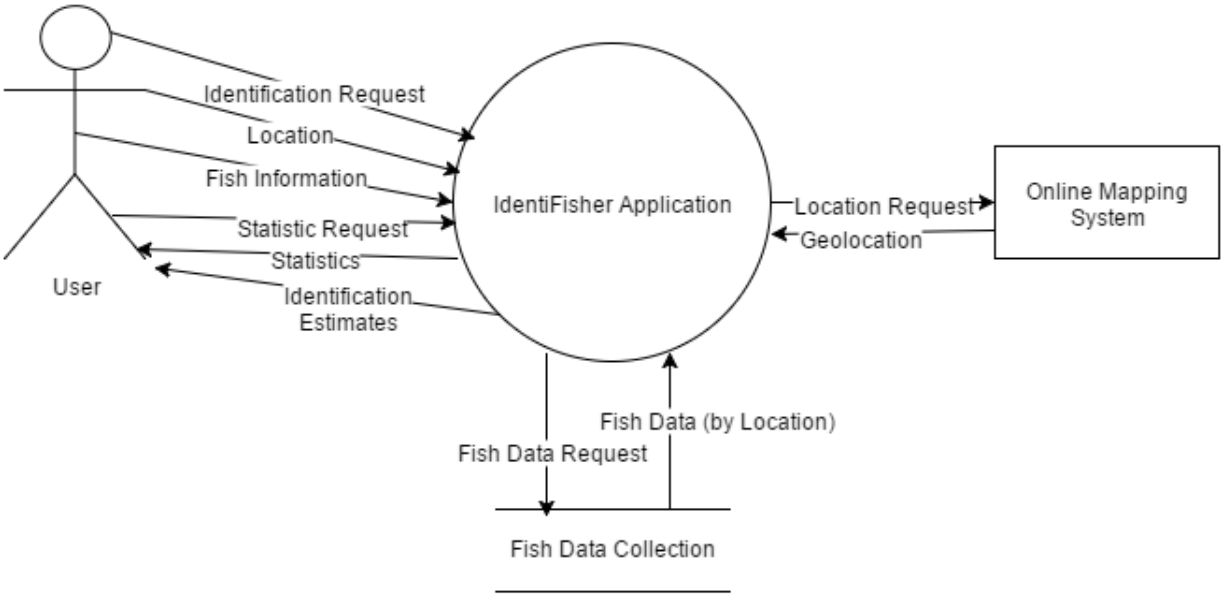

Figure 1: Context Diagram of IdentiFisher Application

2.3 User Characteristics

- The IdentiFisher application is intended to be used by novice to experienced fishers who wish to identify captured fish or access geographical fish population statistics.
- Technologically, an intended user of the application should have access to a 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 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 on which we are writing, which must be Android which is written in Java.
- Although the application does not have any notable budget constraints, there are time constraints to be considered. The Software Requirements Specification must be complete by February 8, 2016. The application must have been thoroughly tested and be fully functional by April 8th, 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 found in a specific lake.
- It is also assumed that the user has access to a GPS-enabled mobile device with an internet connection while they are at the location where they have located the fish they wish to identify.

2.6 Apportioning of Requirements

3 Functional Requirements

Requirement #: 1
Description: 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.
Rationale: The application will not be able to give an accurate estimate of what the breed of fish may be without accepting any inputs.

Requirement #: 2
Description: The application must be able to access the user's geolocation.
Rationale: The geolocation data could help the user identify what breed of fish the user might have because not all fish exist in every lake.

Requirement #: 3
Description: The application must display the fishing statistics of different bodies of water by the geolocation specified from the user.
Rationale: The user should be able to access the information about the statistics regarding the odds of catching certain breeds of fish in the specified body of water.

Requirement #: 4
Description: The application must allow the user to submit their catch results from fishing.
Rationale: The statistics regarding the odds of catching certain breeds of fish in the specified body of water should be accurate and will require constant updates because fish populations could fluctuate over time.

4 Non-Functional Requirements

4.1 Look and Feel Requirements

The IdentiFisher must have an attractive design, with an 70% satisfaction rate with Users.

4.1.1 Apperance Requirements

The IdentiFisher must look professional with a 70% satisfaction rate with Users.

4.1.2 Style Requirements

The Identifisher must have a simple design to promote usability. This is ensure a fast learning curve for the User.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

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

4.2.2 Personalization and Internationalization Requirements

The IdentiFisher must have English language support.

4.2.3 Learning Requirements

The IdentiFisher must be easy to learn by people of ages 10 and above. The User should take no longer than 5 minutes.

4.2.4 Understandability and Politeness Requirements

The IdentiFisher must contain no offensive language.

4.2.5 Accessibility Requirements

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

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

The IdentiFisher must respond to a User input in less than 2 seconds.

4.3.2 Safety-Critical Requirements

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

4.3.3 Percision or Accuracy Requirements

The IdentiFisher must produce correct identifications 75% of attempts.

4.3.4 Reliability and Availability Requirements

The IdentiFisher must be operational and responsive 100% of the time.

4.3.5 Robustness or Fault-Tolerance Requirements

The IdentiFisher must produce correct identifications 50% 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 IndentiFisher must be able to be used for 1 image or query at a time, for more than 95% Users.

4.3.7 Scalability or Extensiblity Requirements

The IndentFisher must be designed modurally such that multiple types of inputs can be added with ease.

4.3.8 Longevity Requirements

The IdentiFisher must remain operational for at least 3 years.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

The IdentiFisher will exclusively be running on the Android Operating System which is on a smartphone.

4.4.2 Requirments with Interfacing with Adjacent Systems

The IdentiFisher must recieve over a 95% apporval rate from the Google Maps API.

4.4.3 Productization Requirements

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

4.5 Maintainability and Support Requirements

4.5.1 Maintainence Requirements

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

4.5.2 Supportability Requirements

The IdentiFisher must have technical support for all of the Users.

4.5.3 Adaptiblity Requirements

The IdentiFisher must adapt to changes and updates in the Android operating system and other dependancies.

4.6 Security Requirements

4.6.1 Access Requirements

The IndentiFisher will be available to all Users which satisy the Android Play Store requirements to download applications.

4.6.2 Integrity Requirements

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

4.6.3 Privacy Requirements

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

4.6.4 Audit Requirements

The IndentiFisher will not have any Audit Requirements.

4.6.5 Immunity Requirements

The IndentiFisher must store a back-up of information regarding which Fish were caught in which lake in the event it is corrupted.

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

The IdentiFisher must not be offensive towards any culture.

4.7.2 Political Requirements

The IdentiFisher must not have any political significance or influence.

4.8 Legal Requirements

4.8.1 Complicance Requirements

The IdentiFisher must not go against and laws or regulations in the Country it operates in or APIs being used.

4.8.2 Standards Requirements

The IndentiFisher will not have Standards Requirements.

List of Figures