

# **Duck Tracker 2.0**

## **Software Design Specification**

**Group 6 -**

**Nick Henderson(nh), Wentian Lin(wl), Christopher Wycoff(cw), Hongtao Yang(hy),  
Dennis Zipprich(dz)**

**6/1/2020**

### **Table of Contents**

- 1. SDS Revision History**
- 2. System Overview**
- 3. Software Architecture**
  - 3.1 Software Components**
- 4. Software Modules**
  - 4.1 Linux Apache MySQL PHP (LAMP) Server**
  - 4.2 Android App**
- 5. Dynamic Models of Operational Scenarios (UseCases)**
  - 5.1 Alternative Design**
- 6. References**
- 7. Acknowledgements**

### **1. SDS Revision History**

This section will provide a change for this document. Changes made should be reviewed by all team members. If any change is made it will be documented here with the following information:

Date - This is the specific day that the change was made to this document.

Author - This is the person responsible for making that change in the form of initials.

Description - This is a brief overview of what changes are made in the entry.

Date	Author	Description
5/14/2020	dz,cw	Created initial document.
5/14/2020	dz,cw	Edit and revise all sections.
6/1/2020	dz,cw,nh	Edited and revised document.

## **2. System Overview**

Duck Tracker is a system for collecting human geolocation data. The data collected is intended to be used by future or existing systems that might benefit from geolocation data. Specifically the Duck Tracker system was inspired by the current COVID-19 pandemic.

The primary tool being utilized by society in the fight against COVID-19 is social distancing. Social distancing is difficult, especially in situations and places where people may aggregate physically.

Duck Tracker attempts to provide the data that may serve as a building block for applications that provide information about the popularity of locations. For example an application that suggests what time is best to visit a location.

The Duck Tracker 2.0 system seeks to not only provide a means to collect and store geolocation. Duck Tracker 2.0 makes installation and simplicity of maintenance a priority.

Additionally anonymity and security of data will be built into the system.

The following are some of the specific ways our software will implement the above goal.

1. The Android app will indicate to the user if they are connected
2. BASH scripts will be written to ease and automate some the installation process
3. Documentation and source code comments will guide a user through installation and setup
4. UUID technology used in version 1.0 will be used again for security and anonymity of user data
5. Data will be transmitted using secure protocols via SSH

## **3. Software Architecture**

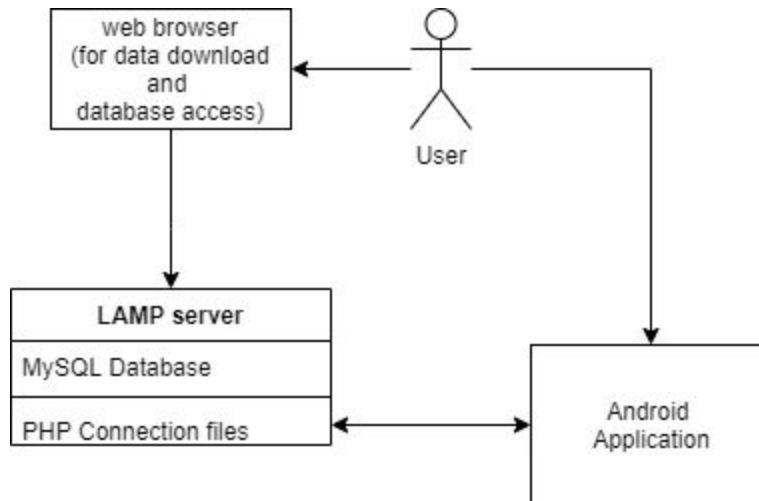
The following subsections details Duck Tracker 2.0 software architecture. This architecture is top level overview of how modules will interact with each other,

### **3.1 Software Components**

1. PHP device connection file
2. PHP MySQL connection/datachange file
3. MySQL Database
4. Android phone application

5. Installation BASH script for MySQL initialization
6. Additional BASH scripts for installation

The following static UML diagram visualizes how each component listed above interacts. Duck Tracker 2.0 was designed to be simpler, therefore our team implemented only two software modules as explained in section 4 of this document.



## 4. Software Modules

This section details the different modules we will use in Duck Tracker 2.0.

### 4.1 Linux Apache MySQL PHP (LAMP) Server

#### 4.1.1 MySQL database

- This will serve as persistent storage of the Duck Tracker geolocation data.
- Rationale: MySQL databases are widely used and proven technology that is transparent and easy to use. These databases are highly configurable to meet developers requirements.

#### 4.1.2 PHP Connection files

- These files will serve as the bridge between the Android app and the MySQL database.
- Rationale: The LAMP server will provide a portable environment for installation and maintenance. The ix-dev server specifically will be used for this function, as it is already configured and likely to be a familiar environment to all stakeholders.

#### 4.1.3 Installation Scripts

- A Shell install script will facilitate the installation of the MySQL database.
- Additional install scripts and documentation will exist to support common errors, this will be based off of group 6's installation testing.
- Rationale: A central goal of the Duck Tracker 2.0 system is to provide a smoother installation experience. These installation scripts will serve as a tool for easier installation.

#### 4.2 Android App

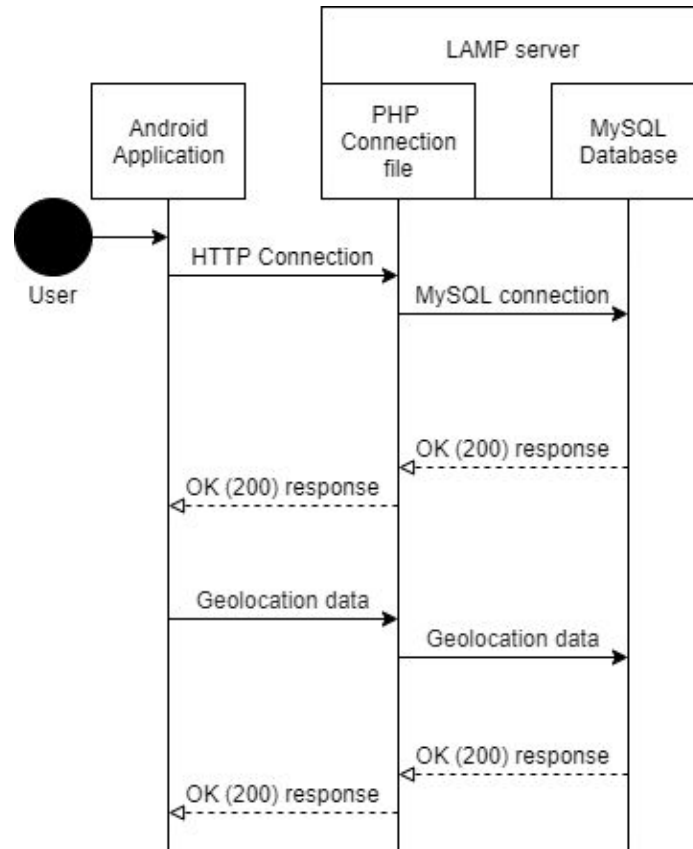
- Allows a user to provide geolocation data to the Duck Tracker system. The app uses the built-in GPS technology that comes with the user's mobile device and service provider.
- Rationale: The Android OS is open source and free to develop on. The Android studio environment allows for a source folder to be imported and auto-configuration options make source code relatively portable.

### 5. Dynamic Models of Operational Scenarios (Use Cases)

The primary use case of Duck Tracker is geolocation data tracking the secondary is data aggregation. The UML sequence diagram below will outline how the primary function is met.

The secondary use case of Duck Tracker allows the user to view all data stored in the database. Users will access their preferred web browser and navigate to Duck Tracker's public site where a download will start instantly.

A UML sequence diagram of Duck Tracker 2.0 collecting a user's geolocation data. "User" indicates the phone App requesting to send collected geolocation data.



## 5.1 Alternative Design (Duck Tracker 1.0)

Below are the original technologies and methods used Duck Tracker 1.0 and the Duck Tracker 2.0 technologies that group 6 seeks replace them.

- + Google Services API --- ( Replaced By)---> Built in phone geolocation technology.  
Why? Limit dependencies and ease installation.
- + Google Firebase DB --- ( Replaced By)---> MySQL and PHP  
Why? Limit dependencies and ease installation. Increase transparency and limit black box software components.
- + Original README and Installation instructions --- ( Replaced By)---> Comprehensive Instructions with referenced inline comments. ALSO installation BASH scripts.  
Why? Increase chance of successful installation without direct human intervention from a group 6 member.

## **6. References**

Uses of Class java.util.UUID. (n.d.). Retrieved from  
<https://docs.oracle.com/javase/8/docs/api/java/util/class-use/UUID.html>

Best practices for working with Android identifiers. (2020, March 19) Retrieved from  
<https://developer.android.com/training/articles/user-data-ids#best-practices-android-identifiers>

## **7. Acknowledgements**

SDS template provided by A. Hornof (ajh) - 5-6-2020 - v0.91, University of Oregon, CIS422,  
<https://classes.cs.uoregon.edu/20S/cis422/Handouts/Template-SDS.pdf>