Duck Tracker
Software Design Specification
Dennis Zipprich (dbz) - 4/26/2020 - v2.0

Table of Contents

1. SDS Revision History

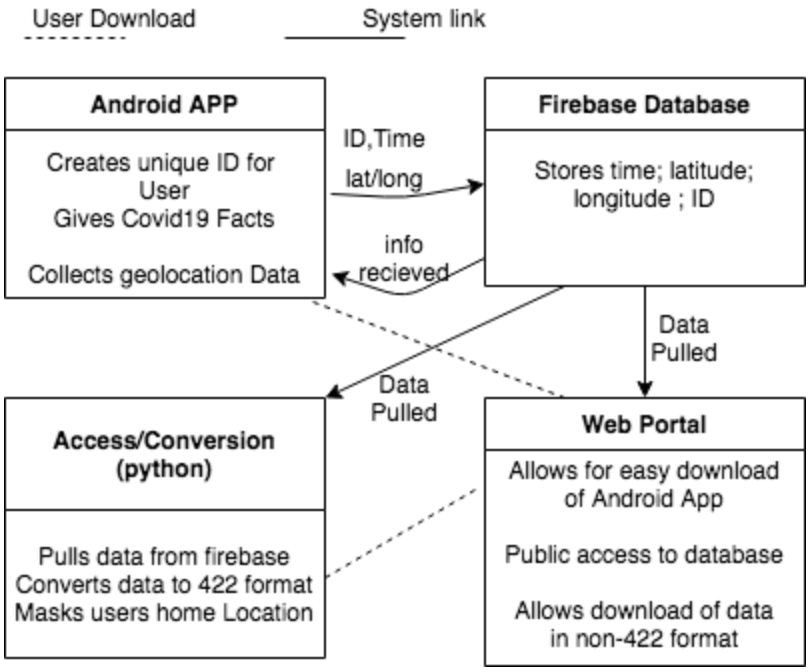
Date	Author	Description
4/10/2020	dbz	Created initial document.
4/15/2020	nbh	Added detail and made small changes to sections 3 and 4
4/23/2020	dbz	removed diagrams, formatted file for .txt format for git hub transfer
4/23/2020	dbz	edited section 1 for better summary
4/26/2020	dbz	edited multiple sections for final submission
4/26/2020	clw	proofread and added content
4/26/2020	nbh	proofread and added content to Python sections

2. System Overview and Software Architecture

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.

Duck Tracker employs 4 major component modules:



Mobile Phone Android Application
Uses built in infrastructure of the user's service provider to track geolocation data via trilateration (Wilson).

Additionally the application (depending on users phone settings) will send geolocation data to a firebase database every 5 minutes.

App includes a user interface that displays a map centered around current location, shows real time latitude and longitude, and provides facts about COVID-19,.

Firebase Database
Receives data from the Android app. Updates data in real time. Allows for Python module and web portal module to pull data in real time.

Python Data Access and Conversion Application
This module provides a Python script with a small GUI interface to pull the data from firebase into a text document (422 specified format)

Web Portal
Allows for easy access to the Phone application and Data Access and Conversion Application.

Allows for the database to be viewed publicly.

3. Software Modules

Duck Tracker is split into 4 different modules. Three of these modules have significant amounts of source code written by the members of “CIS 422 group 6”. The sections following describes the rationale for and functionally of the modules.

3.1 Android app

An Android OS based mobile application was chosen to fulfill the geolocation tracking requirements. The other major option would have been an IOS (iphone) application. Android was chosen because the barrier to entry into Android app development was perceived as lower. Additionally Android seems to have less comprehensive security features, providing less roadblocks to track users' location.

A major obstacle was maintaining consistent background processing across different devices and settings. (See README for more details on how a user can improve functionality through optimal use of the app).

A unique user ID is created when the app first comes online. This ID will only be changed if there is a factory reset done on a phone carrying the Duck Tracker app.

The Android app sends geolocation data to a Google Firebase database (maintained by “group 6”) every five (5) minutes autonomously to the database for storage and aggregation.
The data sent is:

- Latitude
- Longitude
- User ID
- Time and Date

3.2 Google Firebase Database

The data from the Android app is stored persistently in a Google Firebase database.

Given the project’s constraints, using an already-established system like Google's Firebase development platform was the most feasible option for storing the data obtained via the Android

module.

Google Firebase appears to be a robust system with a large and established company behind it. Given our relatively small data storage needs, “group 6” is able to use firebase at the free tier. If this system were scaled up, cost might become a factor and another database option might have been chosen

This module is where all data is stored for safekeeping. All of the other software modules will interact with this database.

3.3 Web App

The Web application has access to all data in the database for viewing and conversion. The web application is written in html and javascript. This interaction is a read only interaction and simply displays user identification, time, date, latitude, longitude and time at location.

Our web application interacts with multiple different modules and serves as a portal for users and stakeholders to view data. This web application can be most conveniently accessed on most modern day web browsers, such as Google Chrome, Safari, Brave, Firefox etc.

Our team chose to go with a simple html and javascript web application as it is a widely used language, simple to maintain, and works in most modern web browsers.

3.4 Python Script

To accommodate for the project requirements of a tab-delimited file for database delivery while also maintaining use of the JSON format for the database, we provide a Python 3 script to download the database in the format specified in the system requirements. Python 3 was chosen due to the team's expertise with the language and because text parsing is well-supported in the Python Standard Library.

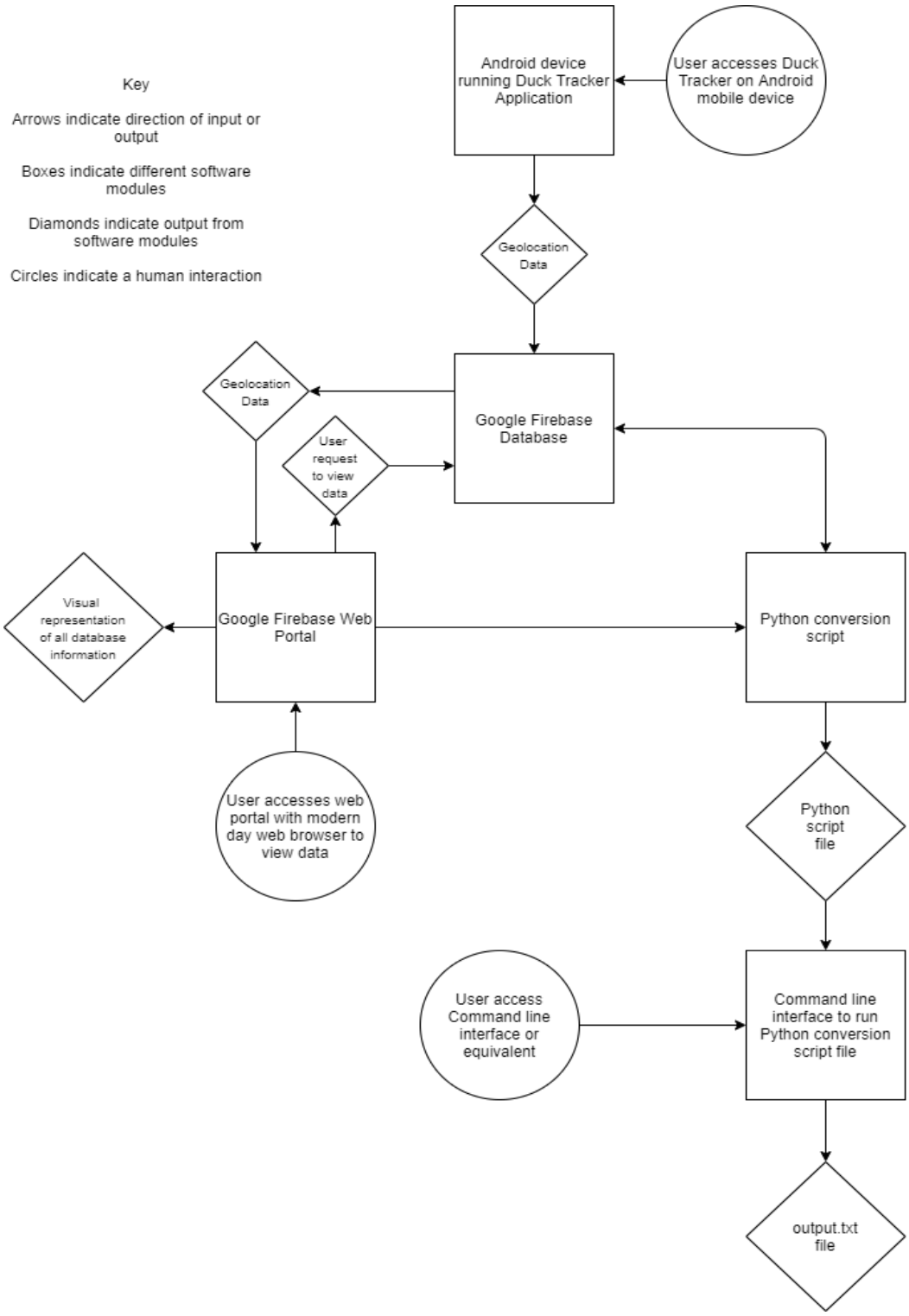
This module is delivered to the user as a .py file. If the user does not have Python 3 installed on their device they must install it first before use. The user will need to use a command line interface or an equivalent to execute the file.

The primary function of this module is to provide a stakeholder with a .txt file of formatted data. This will read all data from the Google Firebase database in a similar fashion as the web application, however it will perform several functions before writing to a .txt file. In addition to being formatted, the script also anonymizes the user’s home location by replacing it with an offset location and calculates the time at location value required by the system requirements.

The stakeholder is able to choose where to save the file and access it at their convenience or use the script as a backup method.

4. Dynamic Models of Operational Scenarios

This UML diagram below outlines how Duck Tracker behaves while operating.



5. References

Vliet, Hans. (2008). Software Engineering: Principles and Practice, 3nd edition, John Wiley & Sons.

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

Tracy V. Wilson "How GPS Phones Work" 24 October 2005. HowStuffWorks.com. https://electronics.howstuffworks.com/gps-phone.htm 14 April 2020

6. Acknowledgments

Android App
https://www.youtube.com/watch?v=M0bYvXlhgSI&list=PLgCYzUzKIBE-vInwQhGSdnbyJ62nixHCt&index=3&ab_channel=CodingWithMitch

https://codelabs.developers.google.com/codelabs/realtime-asset-tracking/index.html?index=..%2F..index#0

https://www.youtube.com/watch?v=hyi4dLyPtpI&ab_channel=ProgrammerWorld

https://programmerworld.co/android/create-location-tracking-android-app-by-exchanging-location-information-over-firebase-database/

https://www.uuidgenerator.net/

Python Script
URLLib documentation
https://docs.python.org/3/library/urllib.html

TKinter documentation
https://docs.python.org/3/library/tkinter.html

TKinter Dialog documentation
https://docs.python.org/3.9/library/dialog.html

Web App

pdfmake v0.1.36, @license MIT, @link http://pdfmake.org */

SZip - A Javascript class for generating and reading zip files
<http://stuartk.com/jszip>

(c) 2009-2014 Stuart Knightley <stuart [at] stuartk.com>
Dual licenced under the MIT license or GPLv3. See
https://raw.github.com/Stuk/jszip/master/LICENSE.markdown.

JSZip uses the library pako released under the MIT license :
https://github.com/nodeca/pako/blob/master/LICENSE

jQuery v1.10.2 | (c) 2005, 2013 jQuery Foundation, Inc. | jquery.org/license
//@ sourceMappingURL=jquery.min.map

DataTables 1.10.16
©2008-2017 SpryMedia Ltd - datatables.net/license

Buttons for DataTables 1.5.1
©2016-2017 SpryMedia Ltd - datatables.net/license

http://purl.eligrey.com/github/FileSaver.js/blob/master/FileSaver.js

jQuery v3.5.0 | (c) JS Foundation and other contributors | jquery.org/license