# User Guide

CSE 700 – Independent Study | Sowmiya Murugiah | 50485124

Table of Contents

1. Introduction
2. Tools Used
3. Guide to Install
   1. Postgres
   2. Docker Desktop
   3. Visual Studio Code
4. Explanation of the code structure
5. **INTRODUCTION**

The phone localization project aims to provide a robust solution for determining the geographical location of mobile devices using Docker and PostgreSQL. This technology enables businesses, developers, and users to track and analyze the real-time or historical location data of phones efficiently. The significance lies in its potential applications, including location-based services, asset tracking, and data analytics. By leveraging Docker for containerization and PostgreSQL for data management, the project offers a scalable and versatile solution for organizations seeking reliable phone localization capabilities.

1. **TOOLS USED**

*Docker:*

Docker is a containerization platform that allows the project to encapsulate the application and its dependencies into isolated containers. This ensures consistency in deployment across various environments, simplifies scalability, and enhances overall portability.

*PostgreSQL:*

PostgreSQL serves as the project's relational database management system (RDBMS). It efficiently stores and manages location data, providing a reliable and scalable foundation for storing and retrieving information related to phone localization.

*Visual Studio Code:*

Visual Studio Code is the integrated development environment (IDE) used for coding, debugging, and version control. Its versatility and extensive plugin support make it an ideal choice for developing and maintaining the project codebase.

1. **Guide to Install:**

**a. Postgres:**

1. **Download PostgreSQL:**

Visit the official PostgreSQL website (https://www.postgresql.org/download/) and download the appropriate version for your operating system.

1. **Installation:**

Follow the on-screen instructions to install PostgreSQL on your system.

1. **Configuration:**

During installation, set up a strong password for the default PostgreSQL user (usually **'postgres'**), password used in this code is **123456**.

Take note of the port number used by PostgreSQL (default is **5432**).

1. **Verification:**

Ensure PostgreSQL is running by accessing the command line or using a graphical tool like pgAdmin.

Connect to the PostgreSQL server using the configured credentials.

**b. Docker Desktop:**

1. **Download Docker Desktop:**
   * Visit the official Docker website (https://www.docker.com/products/docker-desktop) and download the Docker Desktop application.
2. **Installation:**
   * Run the installer and follow the prompts to complete the installation.
3. **Configuration:**
   * Adjust Docker settings according to the project requirements.
   * Ensure that Docker is set to start on system boot.
4. **Verification:**
   * Open a terminal and run **docker --version** to confirm a successful installation.
   * Test Docker by running a sample container, e.g., **docker run hello-world**.

**c. Visual Studio Code:**

1. **Download Visual Studio Code:**
   * Visit the official Visual Studio Code website (https://code.visualstudio.com/download) and download the installer for your operating system.
2. **Installation:**
   * Run the installer and follow the on-screen instructions.
3. **Extensions:**
   * Install relevant extensions for your programming languages and Docker support. Search for "Docker" and "Language Support for Java/Python" in the Extensions view.
4. **Configuration:**
   * Customize settings according to the preferences.
   * Set up version control if necessary, using built-in features or relevant extensions.
5. **Verification:**
   * Open Visual Studio Code, create a new file, and ensure syntax highlighting works.
   * Test Docker integration by checking Docker-related commands in the integrated terminal.

**4. Explanation of the Code Structure:**

**1. Data Collection:**

* This component is responsible for gathering raw phone details, including location data and other relevant information. It interfaces with external APIs or sources to ensure a continuous stream of data.

**2. Preprocessing Node:**

* The Preprocessing Node takes the raw data collected and performs cleaning and processing tasks. This includes handling missing or inconsistent data, formatting, and preparing the data for further analysis. [YOU CAN ADD MORE PREPROCESSING STEPS]

**3. Inference Node:**

* Utilizing machine learning models, the Inference Node extracts meaningful insights from the preprocessed data. This could involve predicting or inferring additional information about phone locations or patterns based on historical data. [YOU CAN ADD MORE ML ALGORITHMS HERE]

**4. Local Database:**

* This component manages the storage of data in various stages—raw data, processed data, and inferred information. It uses PostgreSQL as the local database to ensure data integrity and easy retrieval.

**5. Deployment using Docker:**

* Docker is employed for containerization, encapsulating the entire application along with its dependencies. This ensures consistency across different environments, simplifies deployment, and facilitates scalability.