NETAJI SUBASH UNVIERSITY OF TECHNOLOGY

SOFTWARE ENGINEERING

PRACTICAL FILE

SUBMITTED BY

Kaustubh Shankar 2021UCS1669

Vidhan Vyas 2021UCS1723

PROBLEM STATEMENT

IP ADDRESS TRACKER

The problem statement for an IP address tracker is to develop a web-based application that can track the location of a given IP address. The application should allow users to input an IP address and obtain information about its location, including the country, city, and geographic coordinates.

The system should be scalable and able to handle a high volume of requests without compromising its performance. Additionally, the application should be designed with security in mind and take appropriate measures to prevent malicious users from abusing the system.

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

SUBMITTED BY :-

KAUSTUBH SHANKAR – 2021UCS1669

VIDHAN VYAS – 2021UCS1723

INTRODUCTION

An IP address tracker is a tool or service that allows you to track the location of an IP address. An IP (Internet Protocol) address is a unique numerical identifier assigned to each device connected to the internet, such as a computer, smartphone, or server.

IP address trackers can provide information such as the geographic location of the IP address, the internet service provider (ISP), the organization that owns the IP address, and sometimes even the type of device or operating system being used.

FUNCTIONAL REQUIREMENT

1. **IP address lookup:**

The ability to input an IP address and retrieve information such as the geographic location, ISP, and organization that owns the IP address.

2. **Geolocation:**
   The ability to pinpoint the location of an IP address on a map or provide location coordinates.
3. **User interface:**
   A user-friendly interface that allows users to easily input IP addresses and retrieve information.
4. **Database:**
   A comprehensive database of IP addresses and associated information.
5. **Accuracy:**
   The ability to provide accurate and up-to-date information on IP addresses.
6. **Security:**
   Ensuring that the IP address tracker is secure and protects user data.
7. **Integration:**
   The ability to integrate with other tools or services, such as security software or website analytics tools.
8. **API:**
   An API (application programming interface) that allows other applications or services to access the information provided by the IP address tracker.

Non-Functional Requirements:

1. **Performance:**
   The IP address tracker should be fast and responsive, providing users with the required information quickly and without delay.
2. **Accuracy:**
   The information provided by the IP address tracker should be reliable and accurate.
3. **Security:**
   The IP address tracker should maintain the security and privacy of users' data and the system as a whole.
4. **Scalability:**
   The IP address tracker should be able to handle a large number of requests without any performance degradation or system downtime.
5. **Availability:**

The IP address tracker should be available to users at all times, without any significant periods of downtime.

6. **User Interface:**
   The user interface of the IP address tracker should be intuitive and user-friendly, allowing users to easily navigate and access the information they need.
7. **Compatibility:**
   The IP address tracker should be compatible with different browsers, devices, and platforms.

Technical Requirements

1. **Database management:**

   The IP address tracker should have a database management system that can efficiently store and retrieve large amounts of data.

2. **Data processing:**
   The IP address tracker should be able to process and analyze large amounts of data quickly and accurately.
3. **API development**:
   The IP address tracker should have an API that allows other applications or services to access the information provided by the tracker.
4. **Server infrastructure:**
   The IP address tracker should be hosted on a reliable and scalable server infrastructure to ensure high availability and performance.
5. **Geolocation technology:**
   The IP address tracker should use advanced geolocation technology to accurately identify the location of an IP address.
6. **Security protocols:**
   The IP address tracker should have security protocols in place to protect the system against unauthorized access, data breaches, and other security threats.

Testing Requirements:

1. **Functional testing:**

Testing the basic functionality of the IP address tracker, such as input validation, data processing, and data retrieval.

2. **Integration testing**:
   Testing the integration of the IP address tracker with other tools or services that it may interact with, such as security software or website analytics tools.
3. **Performance testing:**
   Testing the performance of the IP address tracker under different loads and usage scenarios, to ensure that it is fast and responsive.
4. **Security testing**:
   Testing the security of the IP address tracker to ensure that it is secure against hacking, data breaches, and other security threats.
5. **Compatibility testing:**
   Testing the compatibility of the IP address tracker with different browsers, devices, and platforms, to ensure that it works correctly on all devices.

Development Requirements:

1. **Planning and design:**
   A clear plan and design document should be created to outline the goals, features, and requirements of the IP address tracker.
2. **Programming languages and frameworks:**
   The development team should have expertise in the programming languages and frameworks used to build the IP address tracker, such as Python, PHP, JavaScript, or Ruby on Rails.
3. **Database design:**
   The database used to store IP address and related data should be designed with efficiency, scalability, and security in mind.
4. **API development:**
   An API should be developed that allows other applications or services to access the information provided by the IP address tracker.
5. **Geolocation technology integration:**

The IP address tracker should integrate advanced geolocation technology that accurately identifies the location of an IP address.

Use Cases for IP Address Tracker:

1. **Website analytics:**
   Website owners can use an IP address tracker to analyze traffic to their website, including the geographic location of visitors.
2. **Cybersecurity:**
   Security teams can use an IP address tracker to identify the source of cyber attacks and take appropriate action to prevent further attacks.
3. **Fraud prevention:**
   Companies can use an IP address tracker to identify fraudulent activity, such as fake orders or attempts to access sensitive information.
4. **Network administration:**
   IT teams can use an IP address tracker to manage and monitor network activity, including identifying unauthorized devices or suspicious traffic.
5. **Digital marketing:**
   Marketers can use an IP address tracker to gather information about their target audience, including geographic location and browsing behavior.
6. **E-commerce:**
   Online retailers can use an IP address tracker to identify potential customers based on their location and tailor marketing efforts accordingly.
7. **Geolocation-based services:**
   Location-based services such as weather apps, ride-sharing services, and social media platforms can use an IP address tracker to provide personalized content and services to users.
8. **Compliance monitoring:**
   Organizations can use an IP address tracker to ensure compliance with regulations related to data privacy, such as the General Data Protection Regulation (GDPR).

Security Requirements:

1. **Data protection:**
   The IP address tracker should protect the data it collects and stores, including IP addresses and related information, from unauthorized access, modification, or deletion.
2. **Secure communication:**
   The communication between the IP address tracker and its users should be encrypted and secured with SSL/TLS protocols to prevent eavesdropping or man-in-the-middle attacks.
3. **Access control:**
   The IP address tracker should implement access controls that limit access to authorized users and prevent unauthorized access to the system or its data.
4. **Authentication and authorization:**
   The IP address tracker should implement user authentication and authorization protocols to ensure that only authorized users can access the system or its data.
5. **Secure storage:**
   The IP address tracker should store data in a secure and encrypted manner, protecting against data breaches, and implement secure backup and recovery procedures.
6. **Vulnerability management:**
   The IP address tracker should be regularly updated and patched to address any known vulnerabilities and to prevent exploitation by attackers.

Data Requirements:

1. **IP address data:**
   The IP address tracker should be able to collect and store IP addresses, which are unique identifiers assigned to each device connected to the internet.
2. **Geolocation data:**
   The IP address tracker should be able to identify the geographic location of an IP address, providing information such as the country, region, city, and postal code.

3. **ISP and organization data:**
   The IP address tracker should be able to identify the internet service provider (ISP) and organization associated with an IP address, which can provide valuable insights into the source of the traffic.
4. **Time and date data:**
   The IP address tracker should be able to record the time and date of each access or activity associated with an IP address.
5. **Device and browser data:**
   The IP address tracker should be able to identify the device and browser used to access the website or service associated with an IP address, providing information such as the device type, operating system, and browser version.
6. **Traffic data:**
   The IP address tracker should be able to collect and analyze traffic data, such as the number of visits, page views, and unique visitors associated with each IP address.

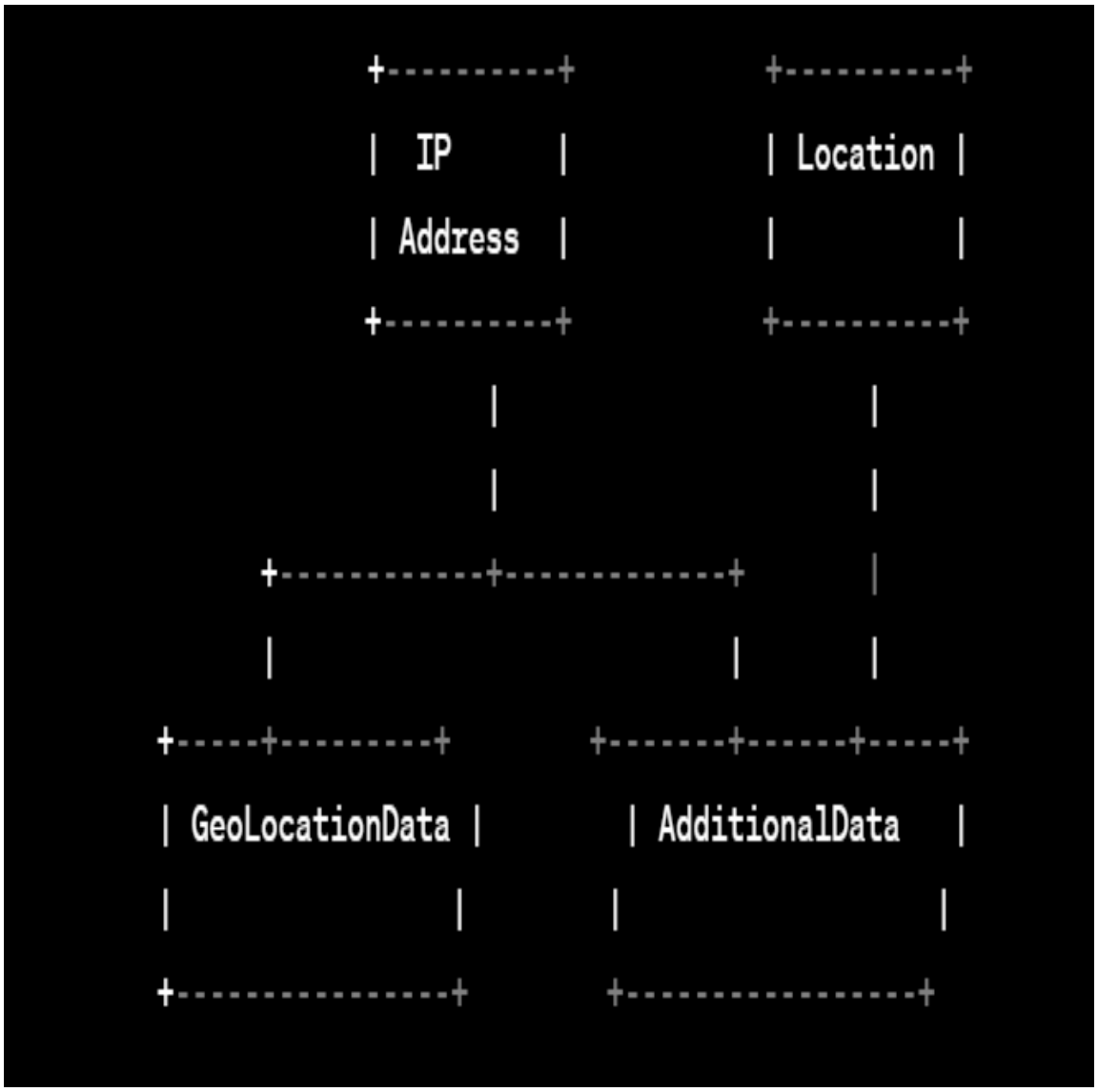
Integration Requirements:

1. **Compatibility with various platforms:**
   The IP address tracker should be able to integrate with various platforms, such as websites, mobile apps, and other software, to track IP addresses and related data.
2. **API integration**:
   The IP address tracker should provide an API that enables integration with other software or systems, allowing developers to access IP address data and related information.
3. **Third-party integration:**
   The IP address tracker should be able to integrate with third-party tools or services, such as analytics platforms or marketing automation software, to provide additional insights and functionality.
4. **Customization:**
   The IP address tracker should provide customization options, such as the ability to modify tracking parameters or implement custom tracking scripts, to meet the specific needs of each integration.
5. **Data export:**

The IP address tracker should allow users to export IP address data and related information in various formats, such as CSV or JSON, for further analysis or integration with other systems.

6. **Real-time tracking:**
   The IP address tracker should be able to track IP addresses and related data in real-time, providing up-to-date and accurate information to users.

ER DAIGRAM



DATA FLOW DAIGRAM

1. Input: User enters an IP address into the tracker.

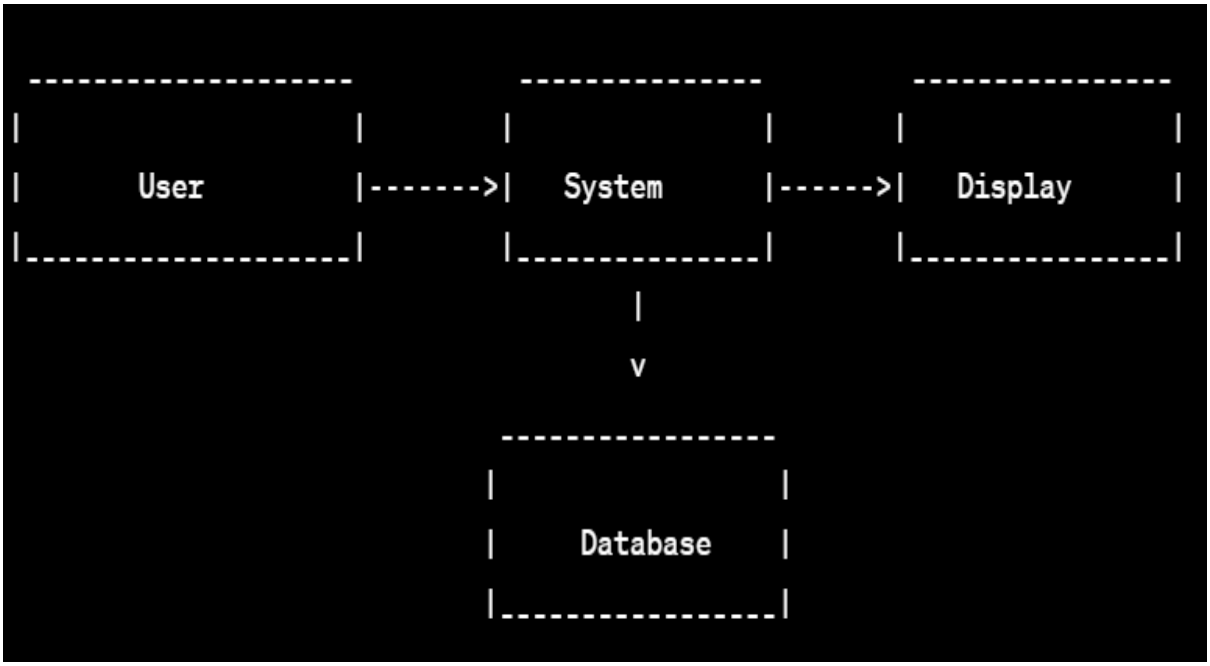
2. Processing: The system queries a database to retrieve information about the IP address, including its geographic location, ISP, and organization.

3. Output: The system displays the information retrieved from the database to the user, including a map showing the approximate location of the IP address.

4. Update: The system updates the database with information about the IP address and any associated activity, such as whether it has been used for malicious purposes.

5. Error handling: If the user enters an invalid or nonexistent IP address, the system will display an error message.

6. Security: The system should ensure the security of the user's data and prevent unauthorized access to the system.



STATE FLOW DIAGRAM

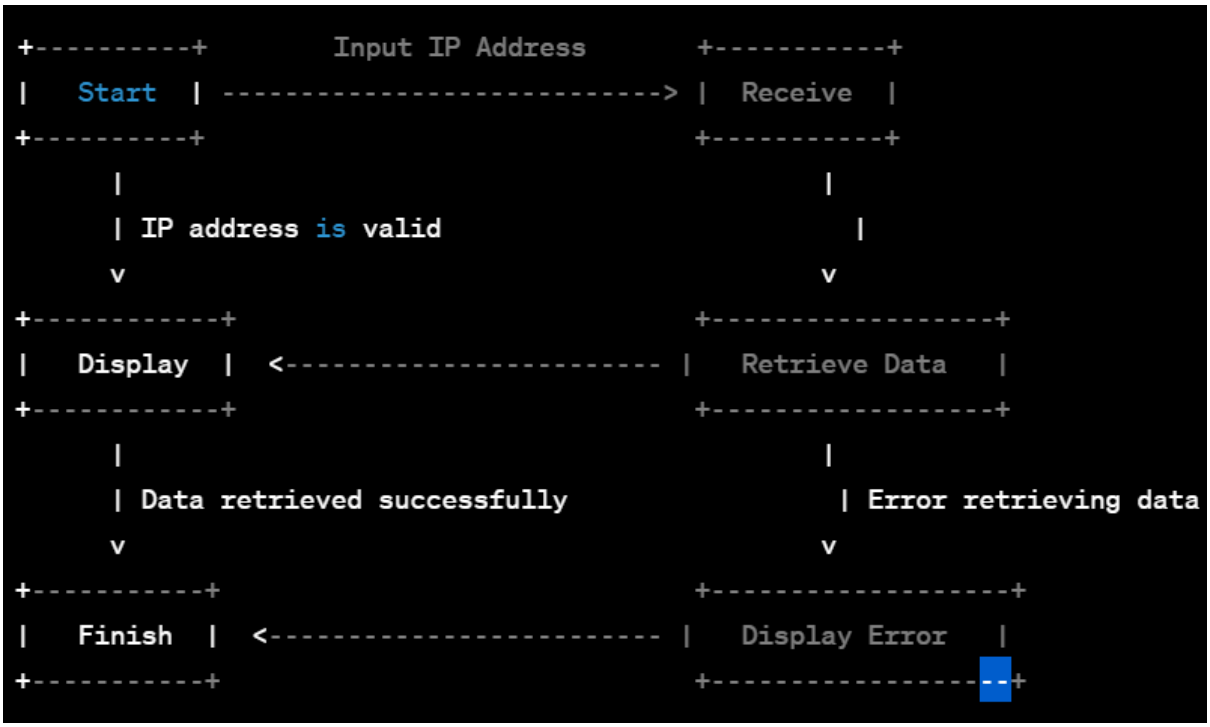
1. The process starts in the Start state, where the system is waiting for input from the user.

2. When the user enters an IP address, the system transitions to the Receive state, where it checks whether the IP address is valid.

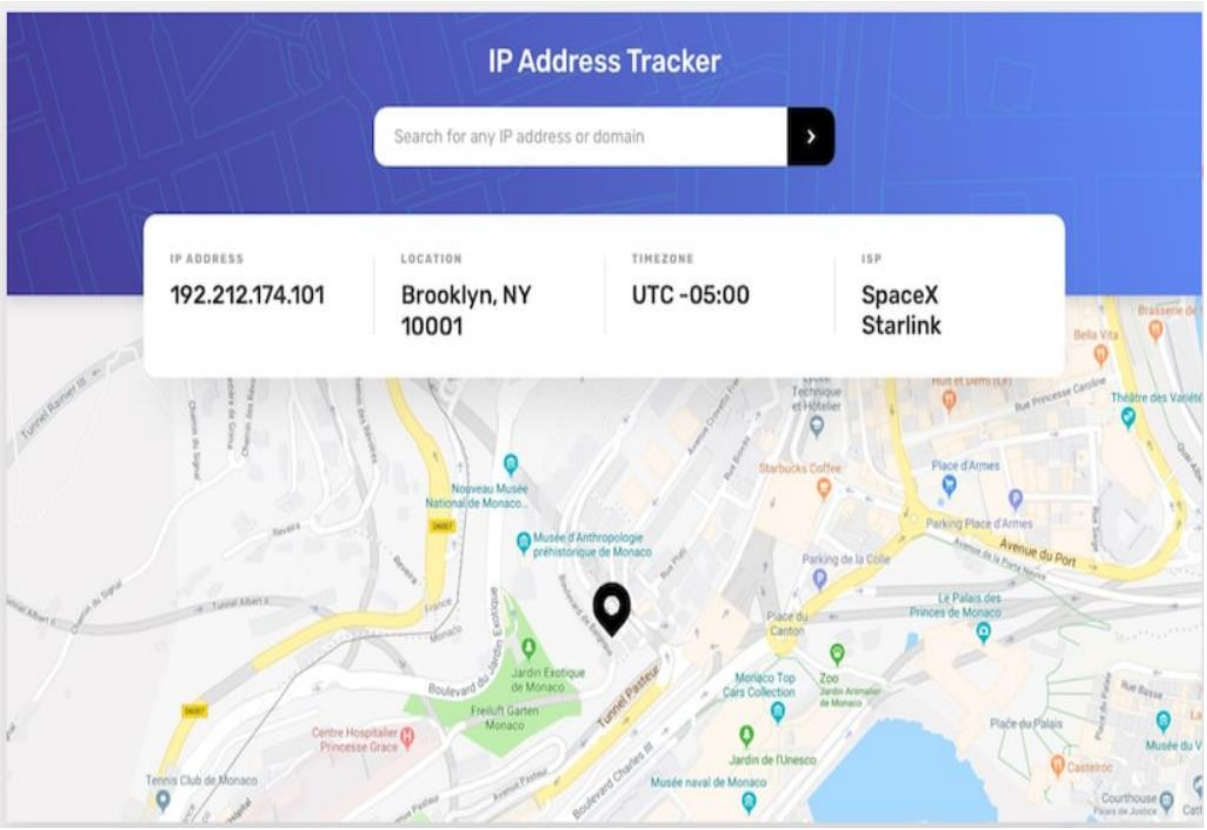
3. If the IP address is valid, the system transitions to the Display state, where it retrieves data about the IP address and displays it to the user.

4. If the data is retrieved successfully, the system transitions to the Finish state, where it displays the information to the user and the process is complete.

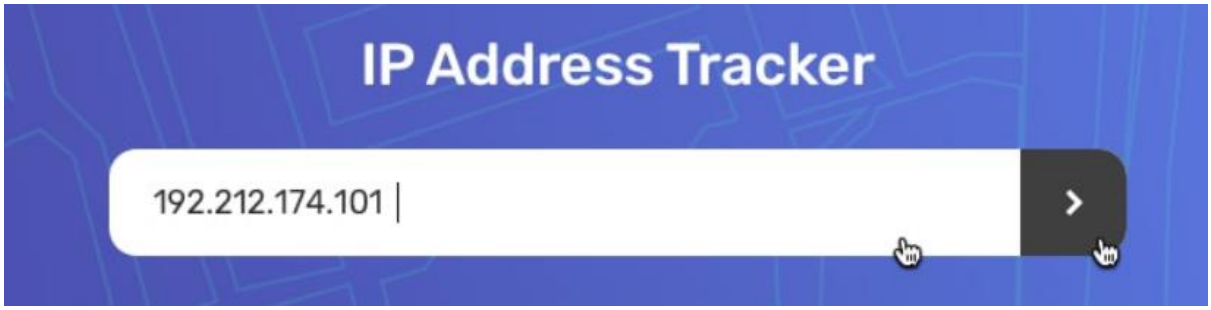
5. If there is an error retrieving the data, the system transitions to the Display Error state, where it displays an error message to the user and the process is complete.

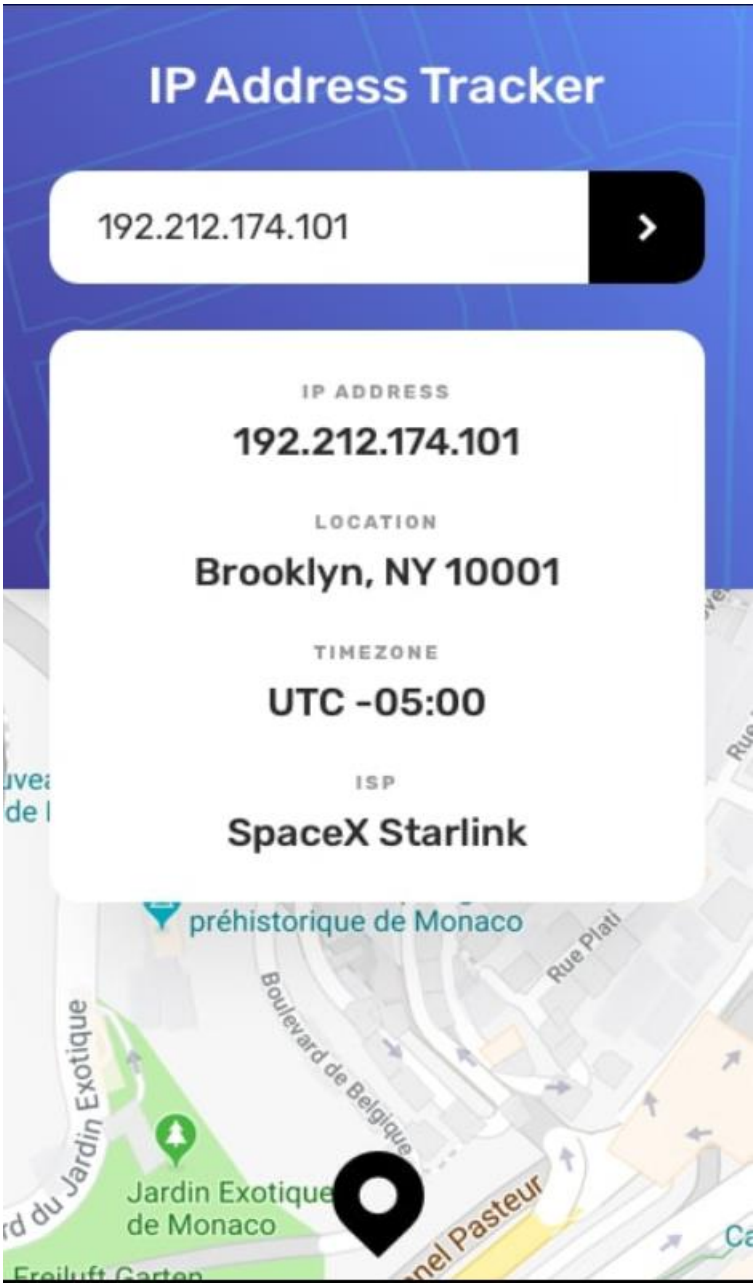


SCREENSHOTS



DESKTOP VIEW




IP Address Tracker
192.212.174.101
IP ADDRESS
192.212.174.101
LOCATION
Brooklyn, NY 10001
TIMEZONE
UTC -05:00
ISP
SpaceX Starlink
préhistorique de Monaco
Rue Plati
Boulevard de Belgique
Jardin Exotique
de Monaco

MOBLIE VIEW