1. Визначення функціональних та нефункціональних вимог до програми, яка буде розроблятися

**Нефункціональні вимоги** (англ. Non-Functional Requirements) — це **вимоги** до програмного забезпечення, які задають критерії для оцінки якості його роботи.

**Функціональні вимоги** (Functional Requirements) — це **вимоги** до програмного забезпечення, які описують внутрішню роботу системи, її поведінку: калькулювання даних, маніпулювання даними, обробка даних та інші специфічні функції, які має виконувати система.

Функціональні вимоги:

1. Вимоги продукту:

Для використання веб-додатку потрібен буде доступ до інтернету. Підтримка зчитуваних файлів: HTML/CSS/Jquery, Python, Flask, Json

Потрібен среда для використання веб-додатку – веб-браузер. Найкраще підходить Google Chrome, тому що він підтримую остані оновлення бібліотек, та має свої вбудовані, які допомагаю пришвидшення роботи даного продукту

1. Вимоги процесу:

Система повина виконувати обробку інформації. Підтримка вище вказаних файлів, зчитування та запис. Особливих вимог не потребує.

1. Зовнішні вимоги:

Веб-додаток, а саме його реалізація повині працювати з базою даних Foursquare/Instagram. Зчитувати, обробляти та виводити її. Зчитування буде відбуватися через реалізацію Python, а записуватись буде в Json файл. Використання Folium для побудування карти з локаціями, які були зчитані з Json файлу. Також можливе буде використання бібліотеки GeoPy для отримання координат заданого місця. Потрібна повна підтримка всіх цих модулів та бібліотек

Нефункціональні вимоги:

1. Вимога до Веб-браузера, потрібна остання реалізація браузера,   
   найкраще Google Chrome
2. Від користувача потрібні точні дані, для точного виконання роботи
3. Дані повинні бути на англійській мові
4. Підтримка всіх модулів та бібліотек для коректності виконання запиту
5. Коректність роботи інтернету та HTTP запитів та дозвіл на використання Google maps

2. Описати можливості пакетів json, xml та html.

JSON

JSON (англ. JavaScript Object Notation, укр. запис об'єктів JavaScript, вимовляється джейсон) — це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією).

JSON знайшов своє головне призначення у написанні веб-програм.

(<https://uk.wikipedia.org/wiki/JSON>)

JSON - простий, заснований на використанні тексту, спосіб зберігати і передавати структуровані дані. За допомогою простого синтаксису ви можете легко зберігати все, що завгодно, починаючи від одного числа до рядків, масивів і об'єктів, в простому тексті. Також можна пов'язувати між собою масиви і об'єкти, створюючи складні структури даних

Після створення рядка JSON, її легко відправити іншому додатку або в інше місце мережі, так як вона являє собою простий текст.

JSON має такі переваги:

Він компактний.

Його пропозиції легко читаються і складаються як людиною, так і комп'ютером.

Його легко перетворити в структуру даних для більшості мов програмування (числа, рядки, логічні змінні, масиви і так далі)

Багато мови програмування мають функції і бібліотеки для читання і створення структур JSON.

Назва JSON означає JavaScript Object Notation (уявлення об'єктів JavaScript). Як і представляє ім'я, він заснований на способі визначення об'єктів (дуже схоже на створення асоціативних масивів в інших мовах) і масивів.

приклад використання:

Користувач натискає мініатюру продукту в онлайн магазині.

JavaScript, що виконується на браузері, генерує запит AJAX до скрипту PHP, запущеного на сервері, передаючи ID обраного продукту.

Скрипт PHP отримує назву продукту, опис, ціну та іншу інформацію з бази даних. Потім складає з даних рядок JSON і відсилає її браузеру.

JavaScript, що виконується на браузері, отримує рядок JSON, декодує її і виводить інформацію про продукт на сторінці для користувача.

Багато в чому ви можете розглядати JSON як альтернативу XML, по крайней мере, в сфері веб додатків. Концепція AJAX оригінально грунтувалася на використанні XML для передачі даних між сервером і браузером. Але в останні роки JSON стає все більш популярним для перенесення даних AJAX.

Хоча XML є перевіреною технологією, яка використовується в достатній кількості додатків, перевагами JSON є більш компактний і простий для розпізнавання формат даних

XML

XML означає Extensible Markup Language, з акцентом на markup (розмітка). Ви можете створювати текст і розмічати його за допомогою обрамляють тегів, перетворюючи кожне слово, пропозицію або фрагмент в ідентифіковану, що сортується інформацію.

Створювані вами файли, або екземпляри документа, складаються з елементів (тегів) і тексту, причому елементи допомагають правильно розуміти документ при читанні на папері або навіть обробляти його в електронному вигляді. Чим більше описових елементів, тим більше частин документа можна ідентифікувати. З перших днів існування розмітки одне з її переваг полягає в

тому, що в разі втрати комп'ютерної системи роздруковані дані все одно залишаються читабельними завдяки тегами.

XML відокремлює дані від HTML

Якщо вам в HTML документі необхідно відображати динамічні дані, то це буде займати занадто багато часу, якщо кожного разу, коли ці дані змінилися, редагувати сам HTML документ.

З XML дані можна зберігати в окремих файлах XML. При цьому ви зосереджуєтесь на використанні HTML / CSS для відображення і шаблонізаціі і можете бути впевнені, що надходять нові дані не зажадають яких-небудь змін в коді HTML документа.

XML спрощує розподіл даних

У реальному світі комп'ютерні системи і бази даних використовують дані в несумісних форматах.

XML дані зберігаються в простому текстовому форматі. Це забезпечує програмну і апаратну незалежність.

Це дозволяє легко створювати дані, які можуть використовуватися різними додатками.

XML спрощує передачу даних

Однією з найбільш час витратних проблем розробників завжди була і залишається досі проблема обміну даними між несумісними між собою системами.

Передача даних у вигляді XML значно знижує складність цієї проблеми, так як дані в цьому форматі можуть бути прочитані різними несумісними додатками.

XML спрощує модифікацію платформи

Перехід на нові системи (апаратні або програмні платформи) завжди займає багато часу. Безліч даних необхідно конвертувати в нові формати. При цьому часто несумісні дані губляться.

XML дані зберігаються в текстовому форматі. Це значно полегшує розширення або модернізацію операційних систем, перехід на нові програми або браузери без небезпеки втратити дані.

XML робить ваші дані більш доступними

Доступ до ваших даних можуть отримувати не тільки HTML документи, а й будь-які інші програми.

Завдяки XML ваші дані стають доступними для всіх видів "машин читання" (голосових машин, новинних каналів і т.д.), що дозволяє значно простіше звертатися до них людям з обмеженнями по зору та іншими фізичними проблемами.

HTML

HTML - це мова гіпертекстової розмітки, який отримав дуже широке поширення в мережі Інтернет. Мова HTML визначає структуру сторінок, яку ви бачите в браузері. Кожен сайт в мережі інтернет використовує для відображення інформації мову HTML.

Основним завданням мови HTML є структурування інформації на сторінці. Мова HTML дозволяє структурувати інформацію різними способами: створювати таблиці, різні види списків, параграфи, додавати в документ зображення, розбивати документ на частини за допомогою заголовків, створювати форми для взаємодії з користувачами, пов'язувати різні HTML документи списками.

Хоча основним завданням HTML є розмітка документа або створення структури, але у цієї мови є і елементи оформлення, для цього є спеціальні теги, які дозволяють виділити текст, зробивши його жирним або курсивним, зробити індекс або підіндекс і багато іншого

3. Описати дані які буде отримувати програма з Інтернету та з якими буде потрібно працювати програмі. Формат, та відповідні до формату всі елементи

Головний ресурс з яким буде працювати програма це соціальна мережа Foursquare/Instagram.

По-перше, мені потрібно отримати ключ-доступу для зчитування інформації та використання API Intagram та Foursquare API. Приклади використання цього API можливо знайти на відкритих репрозиторіях GITHUB’a. Даний API дозволяє зчитувати різного виду інформацію. Зазвичай зчитування інформації відбувається набором повідомлень запиту HTTP також визначається структура повідомлень-відповідей, зазвичай у розширенні мови розмітки XML або в форматі об'єктного запису JavaScript (JSON).

Опис функціоналу XML, JSON та HTML знаходиться в п.2.

Скоріше за все для реалізації я буду зчитувати інформацію в форматі JSON’а для подальшого зручного використання. JSON можливо використовувати як в JavaScript так і в Python. Ці мови підтримують дані такого типу та легко їх обробляють.

Для нашої реалізації вже з зчитаного файлу потрібно буде знайти потрібну нам  інформацію. Для цього нам потрібно буде обробити весь JSON файл та знайти там локацію та кількість користувачів на цій локації або кількість фотографій на локації.

Зазвичай після оброблена інформація виглядає дуже не зрозумілою для користувача та для комп’ютера. Для подальшої використання даної інформації потрібно ії перекодувати. Основна задача буде полягати в швидкому декодуванні інформації. Для цього потрібно буде використовувати додаткові пакети-бібліотеки.

Потрібна інформація – локація. За один запит можливо отримати одну локацію. Для побудування HTML карти потрібно використовувати додаткову бібліотеку Folium. Дана бібліотека з використанням Google Map дозволяє створювати мапу з різними можливостями. Мапа створюється та може зберігатися в любому файлі. Для використання HTML мапи та бібліотеки Folium потрібно використовувати бібліотеку Python – Flask. Дана бібліотека дозволяє використовувати localhost для виведення інформації.

За допомогою бібліотек буде зчитування інформації з файлу JSON, обробка її за дпомогою реалізації на Python, створення мапи за допомогою Folium та виведення всієї інформації в веб-браузер через localhost

Також можливо створити чи використовувати Host для того, щоб інші користувачі могли мати вільний доступ до даного веб-додатку. Можливо використати PythonAnyWhere, як хост та домен, але даний сервіс має багато мінусів та проблем, так що краще реалізувати окремі файли для роботи даного веб-додатку.