Функціональні вимоги	
Орієнтовані на процес	● Система повинна перевіряти валідність імейлів, щоб реєстрація чи пізніше логування пройшло успішно ● Система повинна присвоювати їм унікальний ключ, за яким вони зможуть доступитися до інформації про їхні витрати за весь час користування програмою. ● Система повинна дозволянти користувачу зробити фотографію чеку ● Система повинна дозволяти користувачу переглянути статистику по їхніх витратах за час користування програмою
Орієнтовані на інформацію	● Система повинна зберігати всю історію введення даних користувачем за весь час користування програмою ● Система повинна аналізувати місце покупки (витрат) та зберігати його разом із сумою витрат ● Система повинна зберігати інформацію про час покупки ● Система повинна дозволяти користувачу виправити дані про витрати, якщо певну інформацію було введено неправильно та зберігати нові дані.

Нефункціональні вимоги	
Операційні	● Система повинна працювати на мобільних телефонах ● Система повинна підлаштовуватись під будь-який екран ● Система повинна працювати як на операційній системі Android так і на IOS
Продуктивність	● Обмін інформацією з сервером повинен буди надзвичайно швидким (очікуємо менше 5 сек.) ● Система повинна працювати 24/7, 365 днів на рік.
Безпека	● Тільки зареєстровані користувачі можуть користуватись додатком ● Тільки ввійшовши в акаунт можна переглянути історію, статистику, внести зміни, тощо. ● Тільки розробники можуть вносити зміни в функціонування системи ● Система захищена від DOS-атаки (не більше одного запита в 30 секунд)
Культурні на політичні	● Інформація про клієнтів захищена актом про “Згоду на обробку персональних даних” ● Система оригінально працювати в україно-мовному середовищі, проте в майбутньому буде доданий фнукціонал інших мов

	(англійська, німецька тощо)

JSON

Відповідь на запит може повертається у форматі json. Щоб переконвертувати json формат у більш принятний для python, спершу потрібно імпортувати бібліотеку json. Щоб конвертувати інформацію, використовуйте метод json.loads(fp, cls=None, object_hook=None, parse_float=None, parse_int=None, parse_constant=None, object_pairs_hook=None, **kw), першим аргументом якого є саме json інформація. Схожий функціонал має json.loads(s, encoding=None, cls=None, object_hook=None, parse_float=None, parse_int=None, parse_constant=None, object_pairs_hook=None, **kw), проте у цьому випадку не звертається увага на кодування. Після цього, щоб доступитися до елементу json об’єкту, варто просто використовувати квадратні дужки з ключовим словом:

```
f = {
‘example’: ‘this is example’
}
>>> json.loads(f)[‘example’]
this is example
```

XML

Файл, який розглянено далі, можна знайти у репозиторії з назвою data.xml
Для робти з XML файлом варто використовувати xml.etree.ElementTree. Для початку потрібно імпортувати модуль xml.etree.ElementTree as ET. Щоб завантажити файл потрібно використати метод ET.parse('filename.xml').

```
>>>tree = ET.parse('filename.xml').
```

Метод tree.getroot() повертає корінь файлу, з якого вже можна буде доступитись до більше атрибутів.

```
>>> root = tree.getroot()
```

Щоб перейтись по елементах root можна використати натсупну конструкцію:

```
for child in root:
    print(child.tag, child.attrib)
```

Також доступ до тексту елементів можна здійснювати за допомогою індексів

```
>>>root[1][0].text
```

Цікавим у використанні в циклах є root.iter(parameter), що дозволяє пройтись по всіх елементах xml файлу, які мають такий parameter.

```
>>> for neighbor in root.iter('neighbor'):
```

```
    print(neighbor.attrib)
```

Також важливим є root.findall(parameter).

Використання є таким:

```
>>>for country in root.findall('country'):
        rank = country.find('rank'').text
        name = country.get('name')
        print(name, rank)
```

Liechtenstein 1

Singapore 4

Panama 68

HTML

Парсити сторінку мжна використовуючи клас HTMLParser.

- HTMLParser.feed(data)

data - стрінг, який містить текст у форматі html. Він обробляється, оскільки складається з багатьох тегів всередині тегів, доки тег не буде закрито(приклад використання див. html_parsing.py)

```
>>>parser.feed('<img src="python-logo.png" alt="The Python logo">')
Start tag: img
    attr: ('src', 'python-logo.png')
    attr: ('alt', 'The Python logo')

>>> parser.feed('<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" '
...             '"http://www.w3.org/TR/html4/strict.dtd">')
Decl     : DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd"

>>> parser.feed('<h1>Python</h1>')
Start tag: h1
Data     : Python
End tag  : h1

>>> parser.feed('<!-- a comment -->'
...             '<!--[if IE 9]>IE-specific content<![endif]-->')
Comment  :  a comment
```

Comment : [if IE 9]>IE-specific content<![endif]

З мережі Інтернет програма буде отримувати дані пов*язані з особистим акаунтом Google, використовуючи Google authentification API. Відповідь буде повернена у форматі json:

{'id': '100631245810788397029', 'email': 'darynareshetukha@gmail.com', 'verified_email': True, 'name': 'Daryna Reshetukha', 'given_name': 'Daryna', 'family_name': 'Reshetukha', 'link': 'https://plus.google.com/100631245810788397029', 'picture': 'https://lh3.googleusercontent.com/-Ip3-bfHya4E/AAAAAAAAAAI/AAAAAAAAABU/WphzKlhmMys/photo.jpg', 'gender': 'female'}

Тут ми бачимо такі ключі: id, email, verified_email, name, given_name, family_name, link, picture, gender. З цих ключів необхідним для нашої системи буде email, name, given_name, last_name, gender. Доступ до цих значень можна отримати перетворивши відповідь у json об’єкт та доступившись до значень за ключами.

```
>>>json_obj = json.loads(json_response)
>>> json_obj[‘email’]
darynareshetukha@gmail.com
>>> json_obj[‘name’]
Daryna Reshetukha
>>> json_obj[‘given_name’]
Daryna
>>> json_obj[‘family_name’]
Reshetukha
>>>  json_obj[‘gender’’]
female
```

Також у майбутньому буде реалізовано використання Google API для визначення типу місця покупки, для подальшого сортування витрат за категоріями(наприклад, Трапезна УКУ - їжа). Для цього спочатку потрібно буде визначити координати місця(зісканувавши адресу із чеку), далі дізнатись ID місця і тоді тип закладу.