Following is my take on the project-1 SRS document. It is not perfect, many things can be added. ***Please read it***, you will understand the outline of the SRS document. Everyone can write it differently; there is no correct answer. BUT an in depth analysis should be there.

Project: A simple web scraper.

SRS Document:

1. **Functional Requirements:**
   The main functionality of this product is to scrape or download text present in a webpage. Currently the product focuses on downloading news articles, The main aim is to help users download only the relevant news text from a webpage. In a news webpage there are a lot of contents, including advertisements, related articles, videos and many more. To analyze the news content of a webpage we need to extract only the text part of the news article discarding the other irrelevant information.

   One of the main functions is downloading multiple news articles. The product is capable of downloading multiple news articles and storing them in separate files for easy usage of the client. The client can upload a file containing all the links that the client wants the product to download as text.

   The software is also capable of handling errors. It is capable of identifying links that are not researchable so that the software does not get stuck when querying for an unreachable resource. Special check will be made to validate the syntax of the URL (Universal Resource Locator)

   Running the software: python3 news_scraper.py links.txt

   The links.txt contains the links that the client wants the product to download. One link should be one line. If multiple links are present in one line that is a invalid input and an error will be displayed. The output will be saved in the same folder as the folder in which the product's executable resides. For each link there will be one output text file. The name of the output text files will follow the format: link<link_number>.txt. To describe with an sample:

   link1.txt, link2.txt

   If there are two links in the links.txt input files. link1.txt will correspond to the first link inside of the links.txt file and so forth.

2. **External Interfaces:**
   The product is a CLI (Command Line Interface) based product. The product needs external packages to run. The packages are neatly assembled in a yaml/yml file which will help the user to automatically instantiate the environment. The yml file is a part of the product.
   One essential part of the product is the access to the internet. The packages will get installed from the internet. A decent speed is expected on the clients system for a

faster download of the packages. To scrap the news web pages the internet is expected otherwise an error will be displayed.
The output file format of the product is a text file. Any standard text editing package should be able to open the output of the product.

3. **Non functional requirements:**
   The product is available to any operating system that has a python interpreter already installed in it. This product will only download text and save text files. This product is not capable of downloading javascript or any other executable that may harm the clients system.

   A four month maintenance time will be provided with the product from the day of delivery. All the clients' issues will be recorded in this time frame and remedies will be provided in the form of updates to the product.

   The performance of the product depends on the bandwidth of the internet the client’s system is getting.

4. **Constraints:**
   The constraints are as follows:
   1. python version greater than 3.9.12
   2. The input file should contain one URL per line.