**Requirements Analysis Document**

**Web-Scraping Project**

CITS3200

2021

University of Western Australia

Crawley, WA 6009

--------------------------------------------------------------------------------------------------------------------------

**Preface:**

This document addresses the requirements of the Web-Scraping system. The intended audience for this document are the designers and the clients of the project.

**Target Audience:**

* Client
* Developers

**Web-Scraping Project Members:**

* Shaun Samuel (22717075)
* Maxwell John Dix-Matthews (22701987)
* Juveria Jamal Khan (22650684)
* Joe Win Tan (22702179)
* Xuan Ru Loh (22880502)

**Instructions:**

Care Opinion Australia ( [https://www.careopinion.org.au](https://www.careopinion.org.au/)) is an independent site where anyone can share their stories about their experience of care in the Australian health-care system.

Tags describing the user’s experience can be added when they post their story. For example what they were there for, their sentiment (e.g. what was good, how they felt), etc. Furthermore, health-care providers can post replies to each story. All this information is displayed in the page dedicated to each story and it is divided into different sections of the page layout.

Our client is currently studying how Australians express their emotions when talking about health ( <https://www.uwa.edu.au/research/health-humanities>). They would like to use the data from Care Opinion Australia to analyse how people talk about their health-care experiences online and how they self-report their emotions.

As there are more than 10,500 stories on Opinion Cares website, manual download of the stories is not feasible. Our project would be to build a web-scraping program that will be able to download each story from the website and it would need to save different relevant sections of each page layout as separate files (e.g. the section with the text of the story should be saved as a separate file from the section with the sentiment tags).

Note. The information published on Care Opinion Australia's website is made available under a Creative Commons Attribution 3.0 Australian License ( <https://www.careopinion.org.au/info/terms>), which enables users to copy, share and adapt the material published on the website (<https://creativecommons.org/licenses/by/3.0/au/deed.en>).

**Milestones**

* 29/7 First team meeting and selection of project preferences
* 30/7 Web-Scraping Project allocated
* 3/8 First meeting with the client

--------------------------------------------------------------------------------------------------------------------------

**1.0 General Goals**

* Use data collected from the Care Opinion Australia website to analyse how people talk about their health care experiences online and how they self-report their emotions
* Build a web-scraping program that downloads each story and all the tags from this website into an easily accessible database
* User is able to search and retrieve the data they require from a database

**2.0 Current System**

The system is currently in its planning phase

**3.0 Proposed System**

* Web-scraping program
* Database

**3.1 Overview**

Building a web-scraping program in python, using the beautiful soup package, that is able to collect and download specific data from the Care Opinion Australia website into separate files and then upload these downloaded files into a easy to use database where the downloaded files will be accessible using a unique ID that has been assigned to each file.

**3.2 Functional Requirements**

* Web-scraping program needs to save different relevant sections of each page layout as separate files
* Download meta-data into separate files
* Create a database that stores the downloaded information
* Assign a unique ID for each of the downloaded files
* Link all the files that have been downloaded from the same page
* Make the database searchable by tags/keywords
* Build a DIY web-scraping function in the user interface

**3.3 Non-functional Requirements**

* Maintainability of the database and the web-scraping program
* User friendly interface
* Security and usability of the web-scraping program

**3.3.1 User Interface and Human Factors**

* The web-scraping program is being built for use by the client alone, this program has one sole purpose and will be used on this single occasion
* The client has proficient knowledge in python and should have no issues navigating through the database
* It would be difficult for the user to cause errors when accessing files from the database

**3.3.2 Documentation**

* For developer
  + Risk Register
  + Skills and resource audit
* For client
  + ReadMe file for the program
  + Project acceptance

**3.3.3 Hardware Consideration**

* Overloading the database with too many files
* Search queries may be slow

**3.3.4 Performance Characteristics**

* Not having enough storage to store the files
* Ensure there are no duplication of meta-data

**3.3.5 Error Handling and Extreme Conditions**

* User friendly exceptions in place for search errors in the database
* System to end process and alert user for any input errors

**3.3.6 System Interfacing**

* An input will be the URL that the data is going to be extracted out from
* Input into the database will be the data files collected from the web-scraping program
* Output will be the data files accessed from the database

**3.3.7 Quality Issues**

* System has one sole purpose and will not be used again

**3.3.8 System Modifications**

* Improvements could be sorting reviews by the date of being published
* Include more meta-data that is downloaded into separate files

**3.3.9 Physical Environment**

* The sub-system will only be operated at one location at a time
* The surrounding environment will not affect or be affected by the sub-system

**3.3.10 Security Issues**

* Physical security is not an issue
* All data collected in the database is creative commons

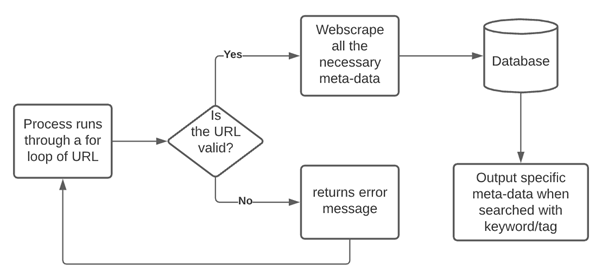
**3.3.11 Resource Issues**

* System will be managed by the client
* All the tools needed to run this program are in the computer

**3.4 Constraints**

* Use of the beautiful soup library
* Primary language will be python
* Proficient at database management

**3.5 System Model**



**3.5.1 Scenarios**

* The user would like to have all the 10,00 stories and the tags related to it from the Care Opinion Australia website for research purposes
* User wants to only see the stories that have the keyword “comfortable” under the “How did you feel?” tag
* A new story has been uploaded onto the website and the user wants to store it into the database

**3.5.2 Use Case Models**

**3.5.2.1 Actors**

* Client

**3.5.2.2 Use Cases**

* Client will run the program for the first time and will get the meta-data in a database
* Client can search for keywords or tags and get the relevant stories linked to it
* Client can update the database by entering the URL of the story

**3.5.3 Object Models**

**3.5.3.1 Data Dictionary**

| **Column** | **Datatype** | **Description** |
| --- | --- | --- |
| Story\_ID | int | Primary key of the table |
| Story | Text | People’s experiences at a healthcare in Australia |
| Title | Text | Title of the story |
| About | Text | The location of where the story is about |
| Good\_Tag | Text | Which part of the experience was good |
| Improved\_Tag | Text | Which part of the experience needed improvement |
| Feel\_Tag | Text | How did the experience made them feel |
| Activity | Text | How many staff members has read this story |
| Response | Text | Response from a staff |
| Response\_Header | Text | Who the response is from |
| Progress | Text | The progress of the story i.e. This story has had a response. |
| Date | String | Date of when the story was posted |