Problem Statement

Wikipedia search app using Flask Framework-Python.

**Software Requirement Specification (SRS)**

1. INTRODUCTION:

A Wikipedia search application that allows users to search and access information through the Python library Wikipedia. Implementing pytest as well as loggers to keep track of processes running in the background.

* 1. PURPOSE

**Educational purposes:** The purpose of this project is to use educate and apply knowledge gained throughout the training.

1.2 FUNCTIONALITIES OF THE SYSTEM:

1)    User:

Enters the landing page and enters data on the search bar

2)    System:

Receives user input, collects data from Wikipedia library and displays the user with the requested data search.

* 1. OPERATING ENVIRONMENT:
* Devices that have access to internet i.e. Desktop PC, Laptop, Tablet, Smartphone

1. SOFTWARE REQUIREMENTS:

*SR1*: Landing page

*SR2*: Search functionality

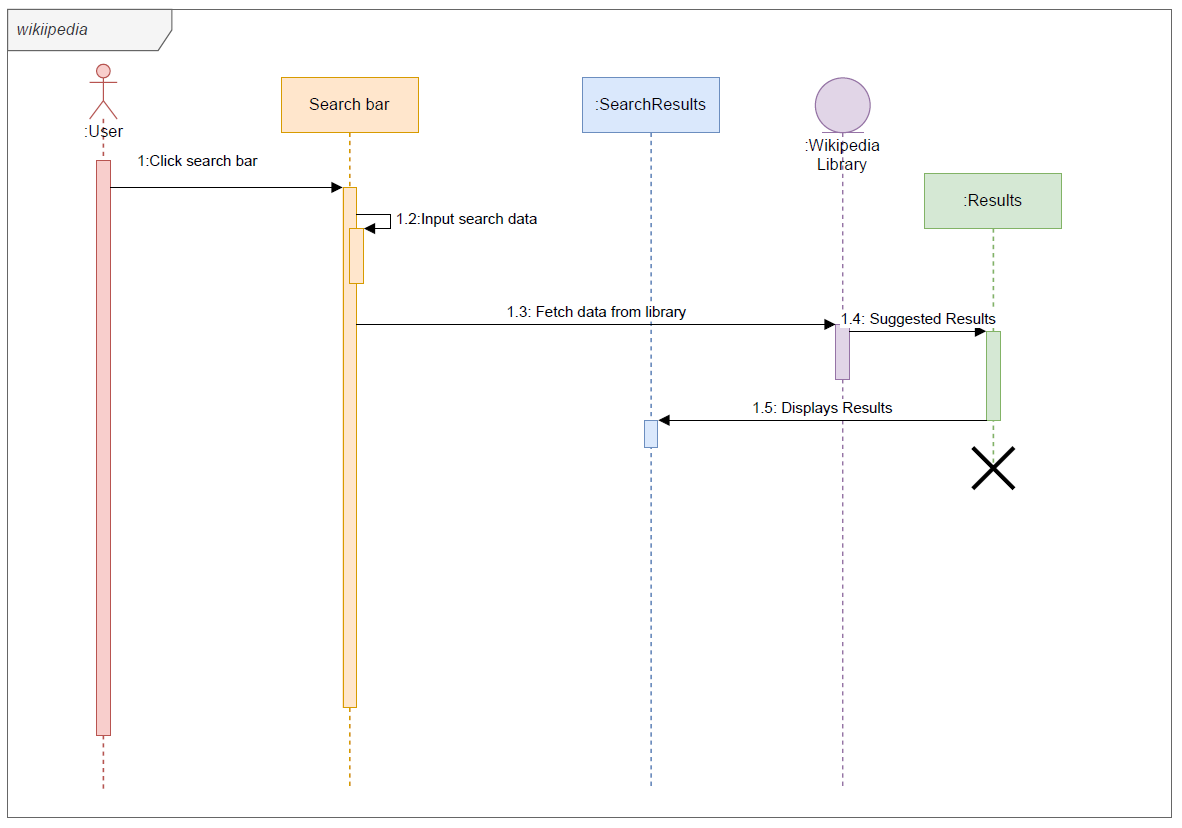
Search bar for user input

*SR3*: Display information searched.

*SR5*: Failure cases:

* Search input cannot be found or loaded to display.

**UML DIAGRAM**



**Pseudocode**

Given: Templates for wiki.html and results.html are styled, and all module prerequisites are met.

Map route for landing page

home():

if request method get does not receive a query:

load wiki landing page form

data = a query to get user input on search bar and format it

return all possible matches in links then display data upon clicked link

run application

if not request.args.get("q"):

            return render\_template('wiki.html')

        data = requests.get(URL, params={

            "action": "query",

            "format": "json",

            "list": "search",

            "srsearch":request.args.get("q")

            }

        )

        return render\_template('results.html',

                data=data.json(),

            q=request.args.get("q"))

if not request.args.get("q"):

return render\_template('wiki.html')

data = requests.get(URL, params={

"action": "query",

"format": "json",

"list": "search",

"srsearch":request.args.get("q")

}

)

return render\_template('results.html',

data=data.json(),

q=request.args.get("q"))