Table of Contents

[**Context Diagram** 2](#_Toc135051273)

[User Activity: 2](#_Toc135051274)

[Application Activity: 2](#_Toc135051275)

[**Solution Architecture** 3](#_Toc135051276)

[Functional requirements: 3](#_Toc135051277)

[**Design Decisions** 4](#_Toc135051278)

[**Non-functional requirements** 4](#_Toc135051279)

[Assumptions 4](#_Toc135051280)

[Future Enhancements 4](#_Toc135051281)

[URIs: 4](#_Toc135051282)

[Source for the implementation 4](#_Toc135051283)

# 

# **Context Diagram**

User through Web UI can upload the CSV file (containing the product related information), Search and Edit the product info as required. The Web Interface interacts with the Pricing Storing and Management application System to manage the product information.

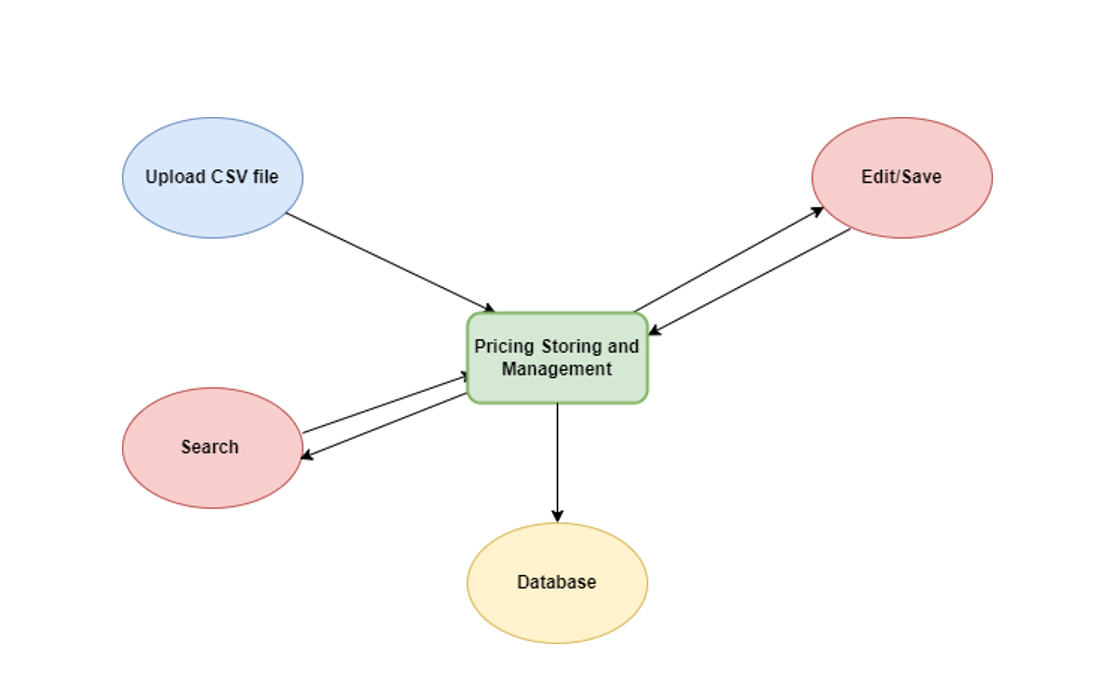


Fig1. Context Diagram

## User Activity:

1. Upload CSV: User can upload the CSV file having the product Info.
2. Search Product: User can search the product information using different search criteria.
3. Edit/Save Product: User can edit the product info

## Application Activity:

Process the request received from the Web UI and manages the product. Response will be returned with the required information.

# Solution Architecture

Below diagram represents the ground for projects, which includes the layered representation of the functional requirements and stages of implementation. It is comprised of many sub processes.

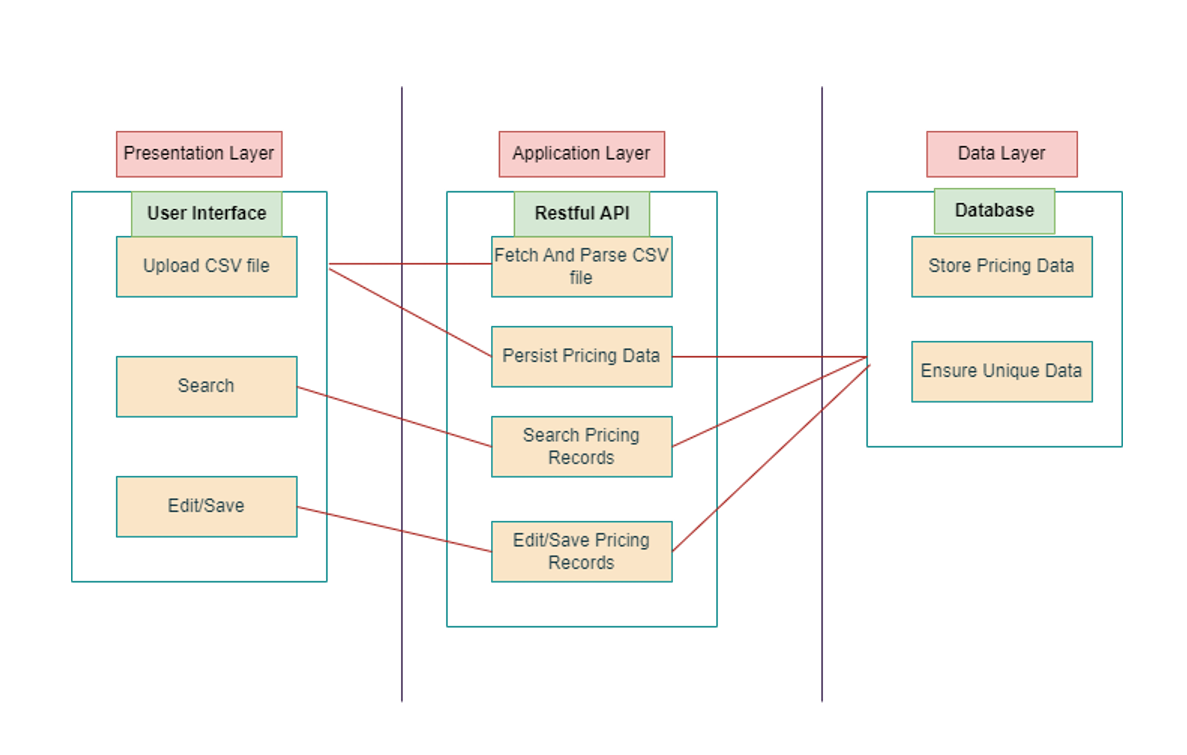


Fig2. Solution Architecture Diagram

## Functional requirements:

* Upload and persist the store and product data.
* Search the product info using different criteria.
* Edit the product info.

To achieve this requirement, multiple layered structure has been introduced.

There are three layers:

* Presentation Layer(User Interface): It consists of multiple options such as:
* Upload: To upload the CSV file.
* Search: To search the product info.
* Edit/Save: To modify the product info.
* Application Layer (RestFul API): It consists of the business logics which has been implemented using the RestFul APIs.
* Data Layer (Database): It consists of the database related activities like table structure, data etc.

# **Design Decisions**

Based on the functional requirement the web application is going to mainly user friendly, hence user focused design approach is used for the application development.

Since it is a web application and will be used by users for uploading the CSV file for pricing feed and searching the pricing records based on some criteria, we will use User-Focused Design

# **Non-functional requirements**

The below non-functional requirements has been considered:

* No Functional impairment
* No Performance issue
* No Data inconsistency
* Highly compatible with the user need
* No user impacted

# Assumptions

The below assumptions has been made

* Correct file type extension is going to be used to load the data.
* User will be aware of the products in his store, miss lead of data should not happen.
* Valid data will be uploaded
* Required information will be present for the product, to increase the readability of the application.

# Future Enhancements

* Security: User authentication
* Product Id and Country Name can be added in CSV file to make it more descriptive.

# URIs:

1. Post Method -- [/api/csv/upload](http://localhost:8080/api/csv/upload)
2. Get All Products -- /api/csv/prices
3. Get by SKU -- /api/csv/sku/{Id}
4. Update Price -- /api/csv/prices/{Id}

**Note:** **I haven’t worked much on UI part. It can be done using Html and CSS by creating three buttons: Upload, Search and Update and we can call Rest API accordingly by giving methods. Due to time constraint , I couldn’t implement it here.**

# Source for the implementation :

Uploaded on Github: https://github.com/Nehagarg4195/CaseStudy\_Tiger