**Technical Solution Approach**

Contents

[1 Introduction 2](#_Toc127885736)

[1.1 About this document 2](#_Toc127885737)

[1.1.1 Purpose & Scope of the document 2](#_Toc127885738)

[2 Component Design 2](#_Toc127885739)

[2.1 Component Design Diagram 2](#_Toc127885740)

[2.1.1 Overall Workflow 2](#_Toc127885741)

[2.1.2 Low level Design 2](#_Toc127885742)

[3 Technology & Frameworks to be used 3](#_Toc127885743)

[4 Solution Approach 3](#_Toc127885744)

# Introduction

## About this document

### Purpose & Scope of the document

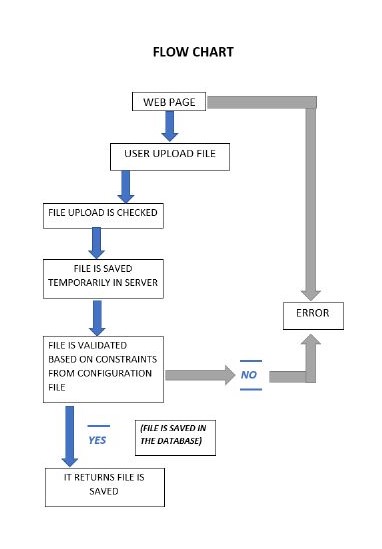
The purpose of the component specification is to systematically record the understanding of component design and the expected usage. This document also include’s the design elements and the overall workflow.

# Component Design

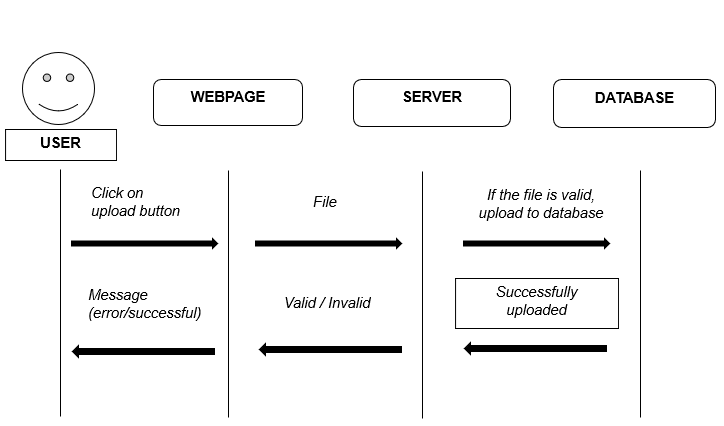
## 2.1 Component Design Diagram

This module explains the details about how the component design is and why the component is designed in the specific way.

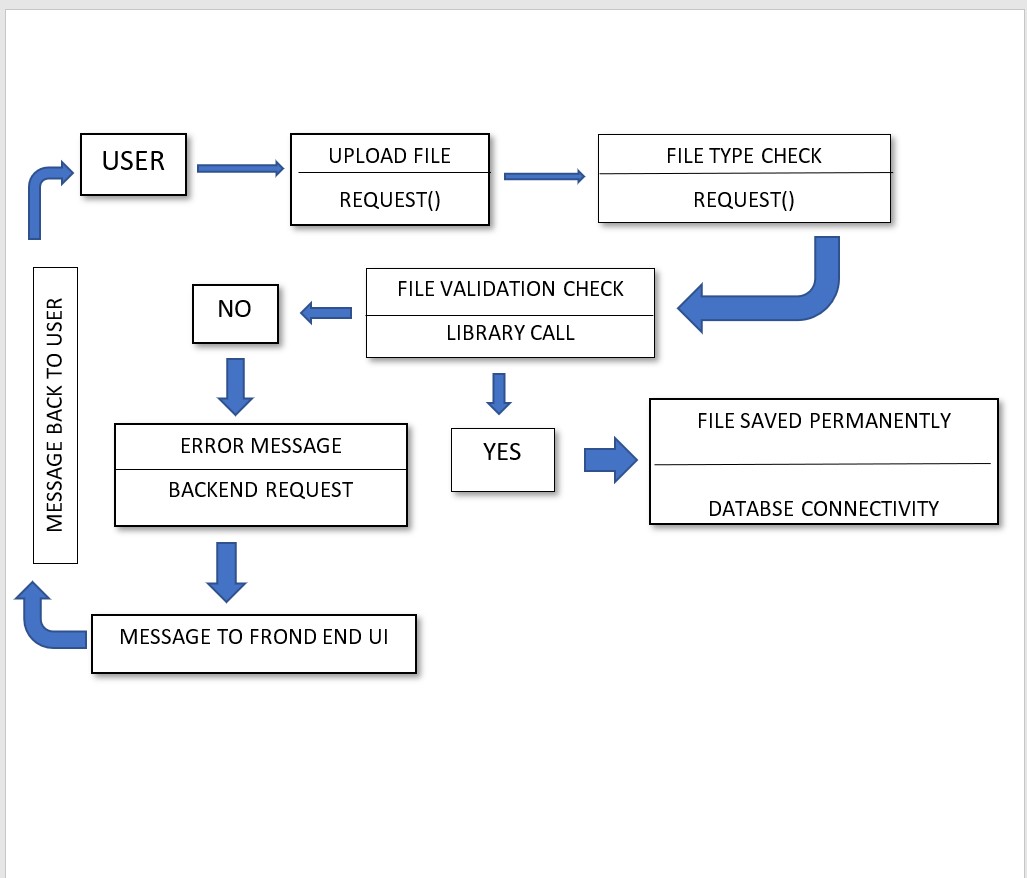
### Overall Workflow



E.g. Sample diagram below is for the File Validation.



### Low level Design

This module explains end to end workflow with the method level details. Below is the sample diagram 

# 

# Technology & Frameworks to be used

This module list’s down the details on technology and framework used for development as well as for testing :

1. Django
2. Flask
3. FastAPI
4. Python

# Solution Approach

This module explains how the functionality of the model and the responsibilities assigned to the subcomponents. The specific pattern used is highlighted here

1. As per the above design diagram, we need to create a web page for user to upload the file.
2. Adaptor: structural design pattern which acts as a wrapper between two objects.
3. The upload of the file type is checked (based on the file extension)
4. The library is used for the validation:
   * The file is temporarily saved in the server (path specified by the developer)
   * The file is validated based on the constraints specified in the config file
   * If the file is valid, the file is saved permanently in the model (table in the database) specified
   * If invalid it raises an exception or error which can be used by the developer to ping the user what’s wrong with file
5. This library is a self-independent module (Implemented for Django, Flask and)
6. We will create a sample test application and use the library created for file validation
7. The newly created library will be added to the test application with the set of configurations below:
   * File size
   * File format
   * File column length
   * Mandatory columns
   * Unique key column
   * Column datatype
   * Datatype format (date, time)
8. Below are the test scenarios for the file upload validation:

* File’ s Column length differs (greater than) from length specified in config file
* File size exceeds 10 MB
* Unique column specified in the config file is not unique
* File with duplicate rows
* File’ s Column’s datatype differs from datatype specified in config file
* File is missing mandatory column specified in config file
* File’s format is different from XLXS,XLSM,CSV,XML
* File’s date columns’ format is different from format specified in config file.