Product Design

**Team 6: Harshika Jain, Shivam Nayak, Sonu Guru, Subodh Sondkar**

# Design Overview

## Architectural design

### Modules

The project is divided mainly as:

1. 3-D Photo Point Cloud Retrieval
2. 3-D Video Point Cloud Retrieval
3. Variation Boundary Detection
4. Variation Measurement

The 3-D photo point cloud retrieval and 3-D video point cloud retrieval processes are further divided as follows:

1. Object Identification
2. Point Cloud Extraction
3. Noise Removal

### Functional Flow

The 3-D photo point cloud retrieval and 3-D video point cloud retrieval will have the same subprocesses but with different implementations (polymorphism).

In the standard flow, a 3-D video and an STL file of an object must be provided. The standard flow is as follows:

1. 3-D Video Point Cloud Retrieval
2. Variation Boundary Detection
3. Variation Measurement

## System interfaces

### User Interface

We are creating a web app so that the user can interact with the system. The home page will have the main options that the user can take. These are explained serially.

#### Pipeline

Firstly, the entire expected pipeline starting from video object identification, point cloud extraction, noise removal, variation boundary detection and variation measurement. Clicking this will get you to a page which will ask for the raw 3-D video and STL file of the object. On uploading and continuing, it will send the appropriate input to each process, and show the output of the processes serially. The expected output of each subsystem is described in the corresponding user interface section.

##### Stages of Pipeline:

1. Point Cloud Retrieval
2. Variation Boundary Detection
3. Variation Measurement

The preceding stages have been explained ahead.

#### 3-D Photo Point Cloud Retrieval

The next option is to extract the point cloud of an object from a 3-D photo. This will ask for a raw 3-D photo and an STL file. The process is divided into three subprocesses, and the output of each process will be displayed serially. The following subprocesses and their corresponding outputs are:

* Object identification: the accuracy of identification (the statistical quantity of similarity between the STL and the identified object).
* Point cloud extraction: the raw STL file of the extracted object (available for download).
* Noise removal: the cleaned STL file of the extracted object (available for download).

#### 3-D Video Point Cloud Retrieval

The third option is to extract the point cloud of an object from a 3-D video. This will ask for a raw 3-D video and an STL file. The process is divided into three subprocesses, and the output of each will be shown serially. The following subprocesses and their corresponding outputs are:

* Object identification: the average accuracy of identification over the video (the statistical quantity of similarity between the STL and the identified object).
* Point cloud extraction: the raw STL video file of the extracted object (available for download).
* Noise removal: the cleaned STL video file of the extracted object (available for download).

#### Variation Boundary Detection

The next option is to detect variation boundaries from an STL video file. On submitting the file, it will output the boundary detection model, which can be downloaded.

#### Variation Measurement

It will take in an STL video file or a boundary detection model, and calculate the required statistical quantities throughout the video, and store them in a file or dictionary, which can be downloaded.

### APIs

Our coding would be in python3 and cpp. All the function calls from the web would be to python3. To link python3 and the web, we would use **django** as the API. Any cpp modules we want to run would be through the python code only.

### Model

Draw a simple class diagram and describe the classes in the table in this section. This diagram should represent the classes and their relationships. It is only necessary to show methods that are publically accessible by other classes. Only show an instance variable of a class if it is publically accessible. The diagram and the table should be consistent with each other.

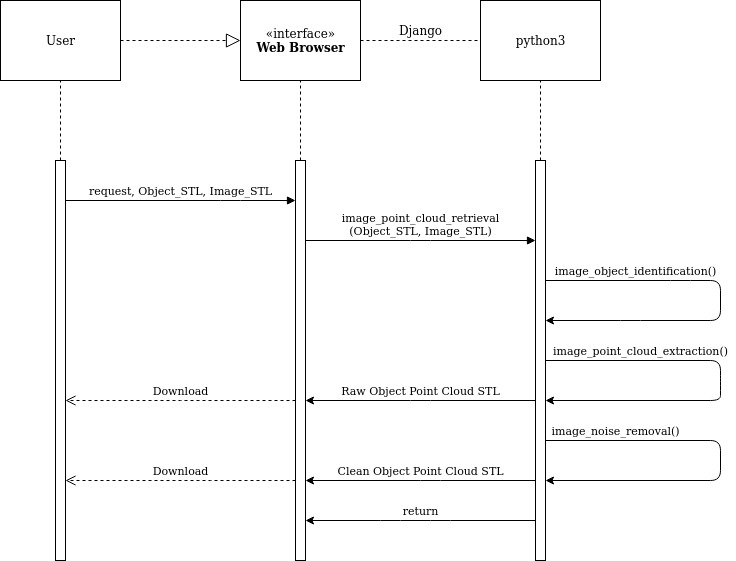
Identify the classes (logical groupings of software methods that provide a related set of services). Make sure the design conforms to good design principles.

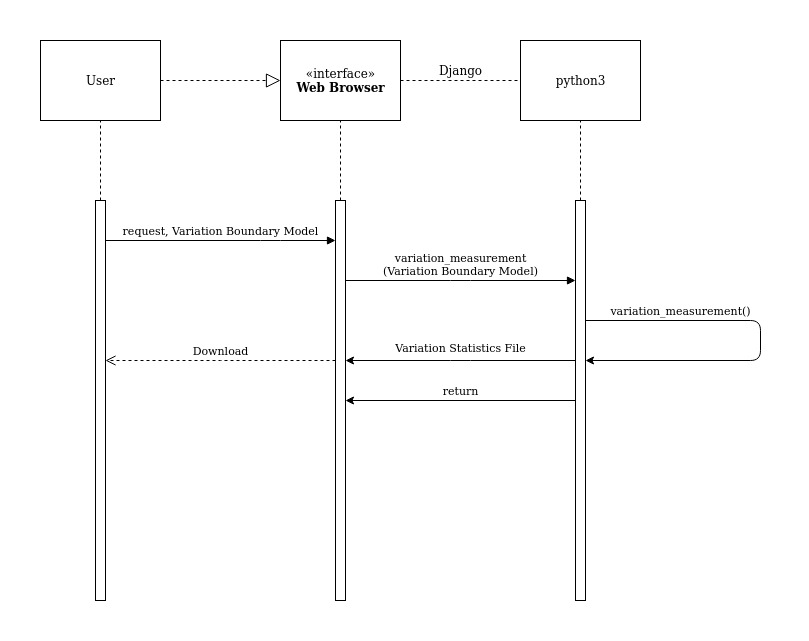
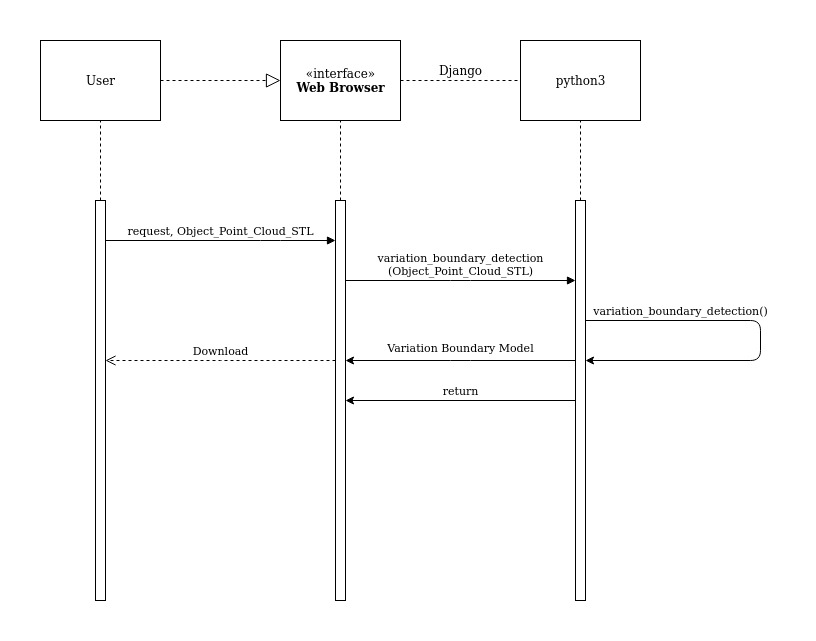
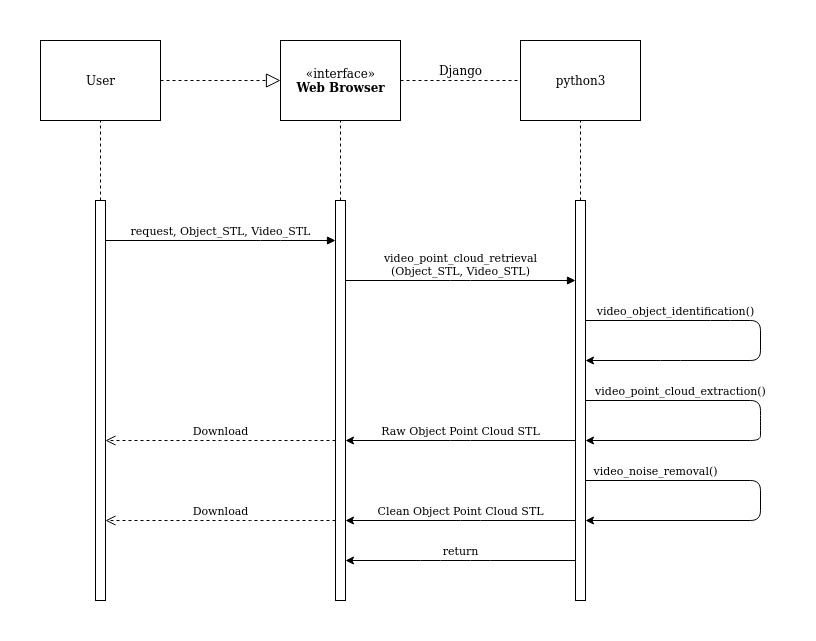
For each class, specify the information it maintains and the functionality it provides. Provide sufficient detail so that the purpose of each class in the design is clear.

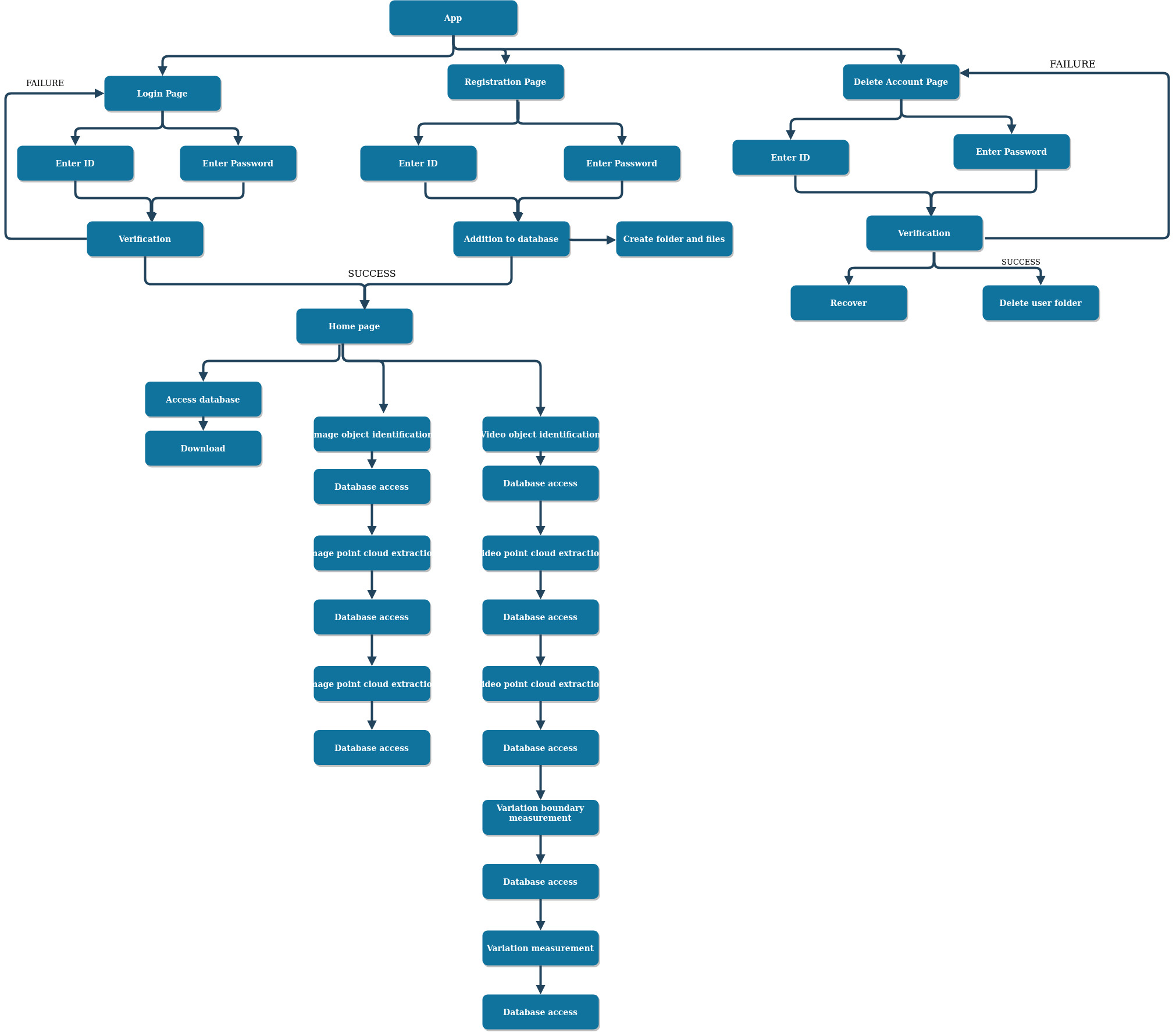
|  |  |
| --- | --- |
| 3-D Photo | Class State   * Point Cloud Data   Class Methods   * Object Identification * Point Cloud Extraction * Noise Removal |
| 3-D Video | Class State   * Point Cloud Data   Class Methods   * Object Identification * Point Cloud Extraction * Noise removal * Variation Boundary Detection * Variation Measurement |

.

# Sequence Diagram(s)







# Design Rationale

This is a running list of issues that arise as your design process proceeds. This is an important section of the design document as it captures the **thought process** of the product's designers. It includes why or why not (rejected solutions) a design decision was made and supports future changes to the product. It should be updated whenever a design change occurs.

*It is RARELY the case that the first design you consider is the best one that you can come up with that meets the requirements and that can be implemented, tested, and delivered on schedule. Your instructor will be looking for signs that you considered at least a few approaches, and that you had a coherent rationale for preferring the design your team eventually adopts.*

*This is the place to record such thoughts – what alternatives did you consider? What are the strengths (and deficiencies) of the final design compared to the other alternatives considered? Why did you select the approach you finally chose? This last question should be answered with an eye to the tradeoffs inevitably involved in creating an appropriate design.*