Video Surveillance System

Software Design Document

Version 1.0.0
Status: draft

Prepared by Jainam Shah(2017csb1107)

Dilip Sharma(2017csb1073)

Revision History

Name	Date	Reason For Changes	Version	Date of Approval
	29 Sep 2020	Initial Edit	1.0.0	--

Introduction

Purpose

The purpose of this document is to create a standalone desktop application for video analysis of security cameras with automated scene/object detection from video files. This will ease the process of reviewing the video for wrong activities manually by reducing inactive video frames. This document will explain the purpose and features of the software, the interface of the software, what the software will do, the constraints under which it must operate. It also identifies functional and non-functional requirements with a use case diagram. This document is intended for both the end-users and the developers of software.

Intended Audience and Reading Suggestions

The document is expected to be read by the end-users and the developers of the application. The document starts by briefing the purpose of the product being developed. Thereafter it goes on to describe the Functionalities and the Operating Environment. The developers are advised to read the whole document.

Product Scope

The Video Surveillance System will analyze the video footprint of security cameras. This product is based on computer vision which is a field of Artificial Intelligence. Using various deep learning models, this application will detect and identify objects given in a video frame. The product will be available for all major OS(Windows, macOS, Ubuntu). Above all, we hope to provide a comfortable user experience with the interface.

References

1. https://github.com/Breakthrough/PySceneDetect
2. https://medium.com/@abulka/electron-python-4e8c807bfa5e
3. https://github.com/argman/EAST

Overall Description

Product Perspective

Common techniques used for the above scenarios are tedious and consume a lot of time as well as manpower. Further, there are many software available and are also being developed but they are designed only for specific purposes. We plan to deliver software that can be adapted to various scenarios. This is a new self-contained product that provides the user an option to design complex workflows on the detection of desired object/movement.

Product Functions

We plan to deliver a thick client-based software. This product aims to provide the functionality to the user for extracting relevant information from a given photo or video which he/she can then use to proceed further with his/her defined workflow. The software will take input as an image or a video URL and then give output in a specific format which will contain the relevant information. This information will include the image id and other details about the persons/objects identified. After receiving this information, the user can proceed further with whatever workflow he has defined.

User Classes and Characteristics

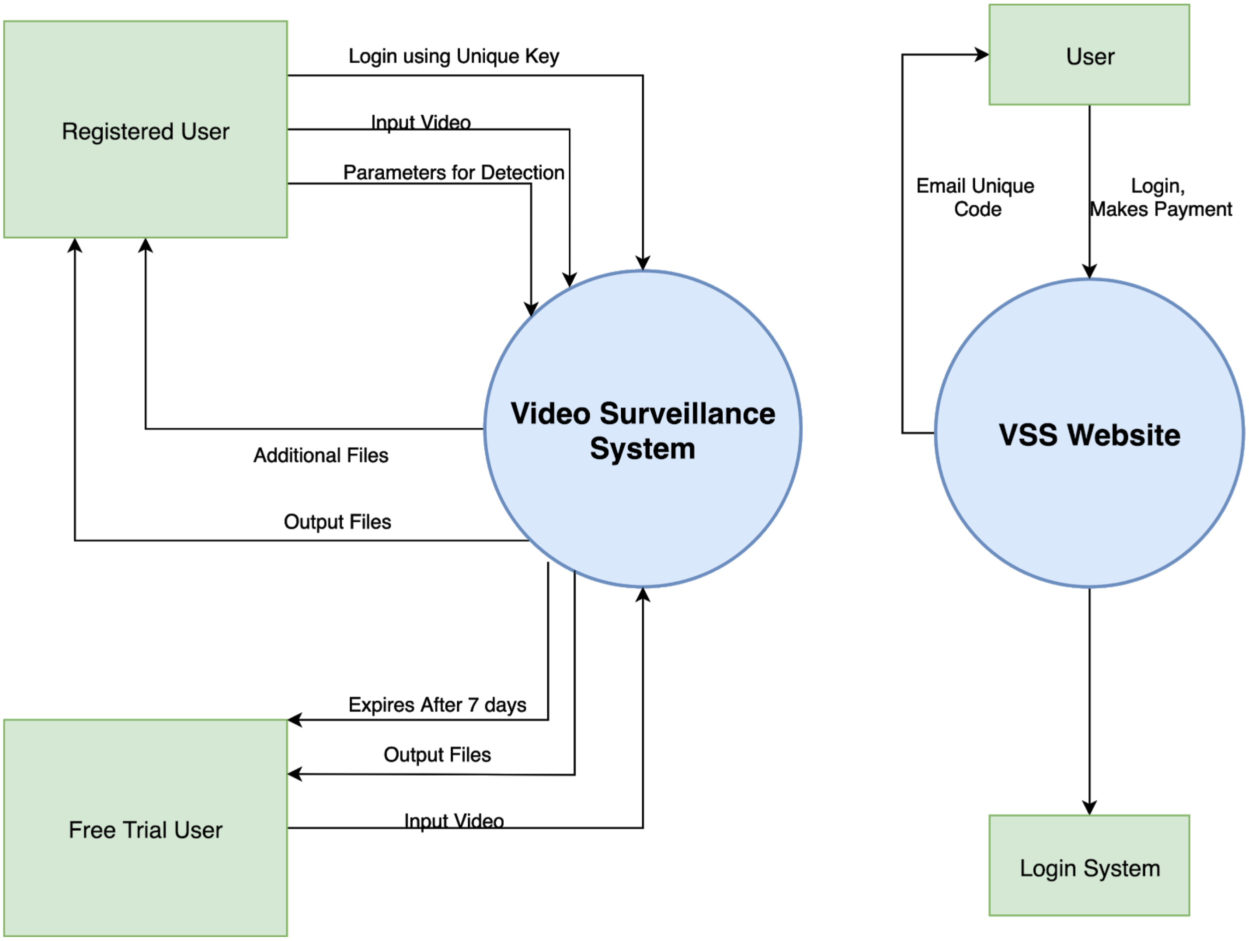
ID	**User classes**	**Description**
U-1	Registered user	A user can directly access the software to perform the functions offered.
U-2	Trial user	A user who is given the service free of cost. He has access to limited features.

Operating Environment

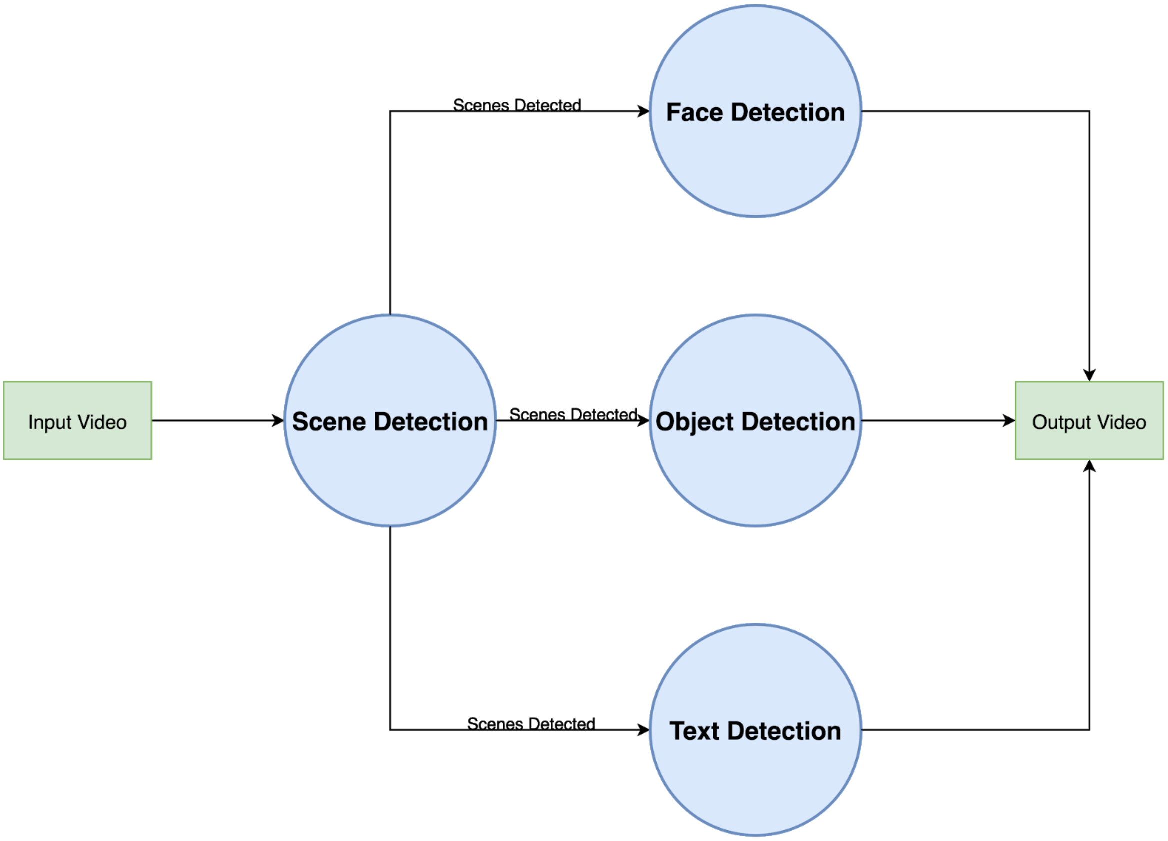
- **Client**
    - Operating System - Windows/Ubuntu/macOS

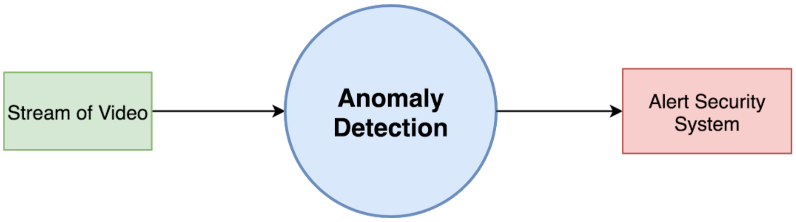
System Design Diagrams

Context Diagram

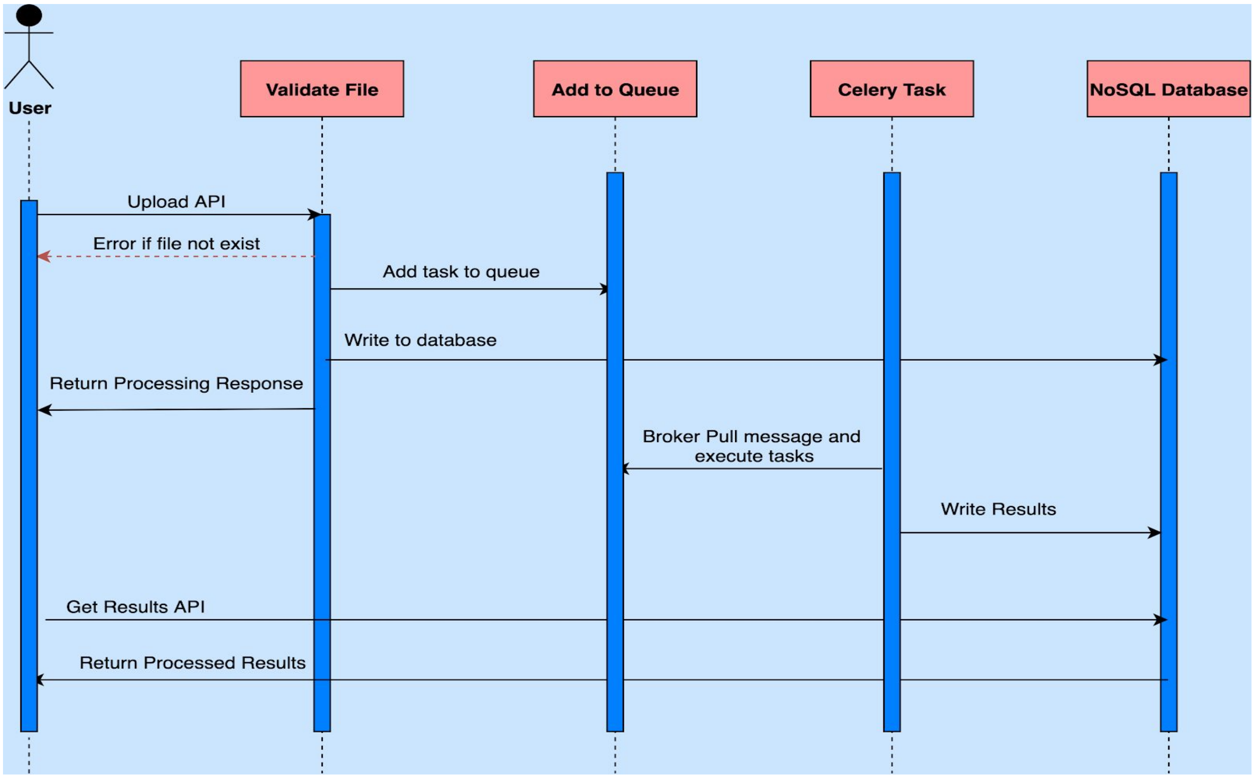


Video Processing Pipeline

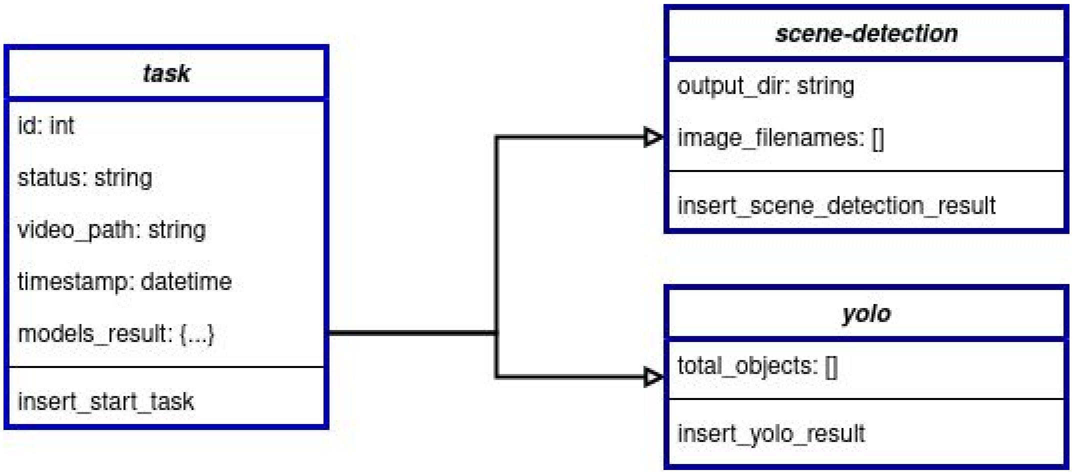




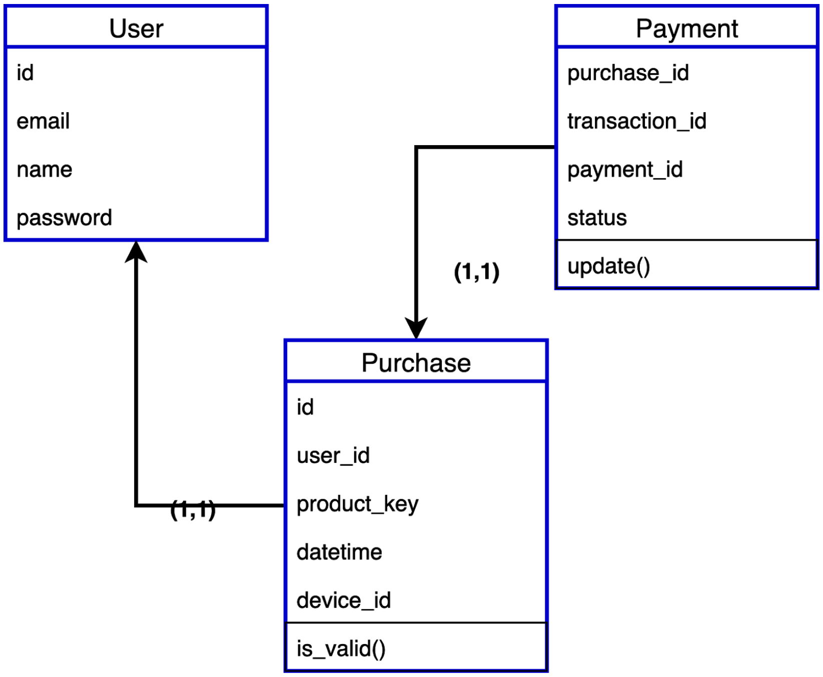
Sequence Diagram



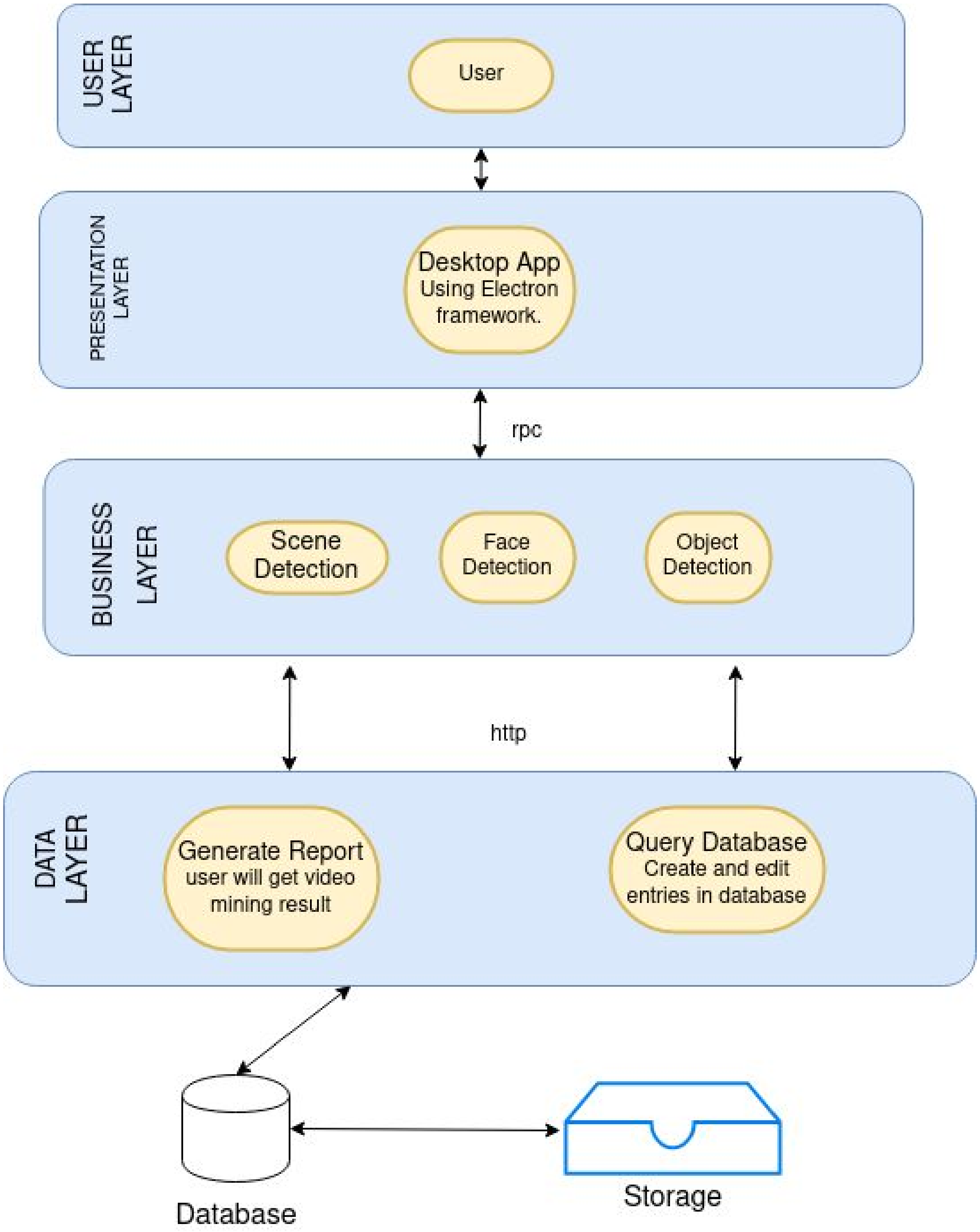
Domain Model



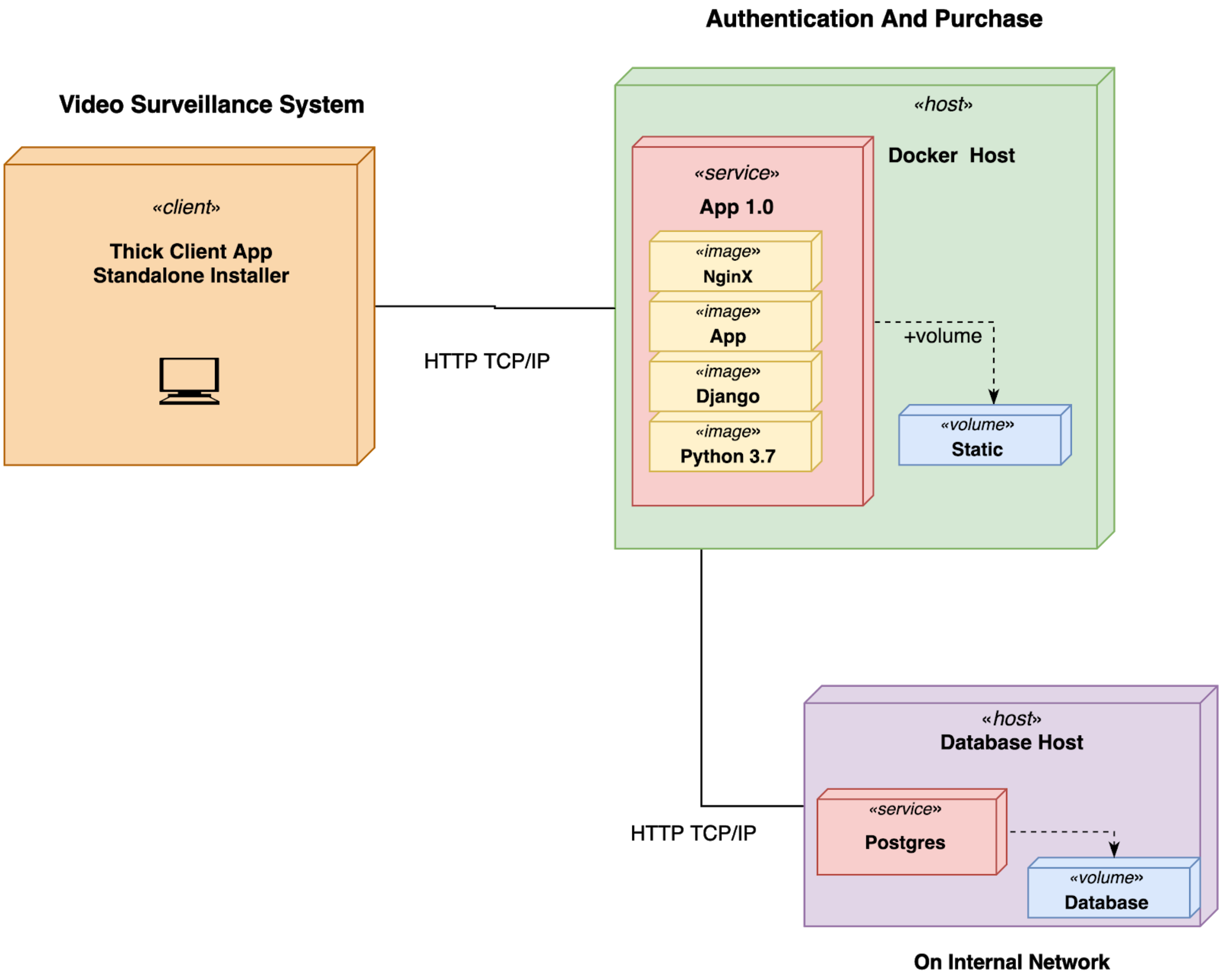
Class Diagram



Logical Structure Diagram



Deployment Diagram



Design Decisions

- **Use of Electron JS for UI:** helps us develop cross-platform applications by using existing web technologies
- **Use of React JS:** very a simple and lightweight library that only deals with the view layer
- **FastAPI:** It is a modern Python web framework designed to: provide lightweight microframework with an intuitive, Flask-like routing system.
- **Celery:** Celery is a task queue implementation for Python web applications used to asynchronously execute work outside the HTTP request-response cycle.