Eyeconic - Glasses Shop Management System

Software Architecture Document

Version 1.0

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 14/Dec/19 | 1.0 | init | tungpd |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Architectural Representation 4

3. Architectural Goals and Constraints 5

4. Use-Case View 5

4.1 Use-Case Realizations 6

5. Logical View 18

5.1 Overview 18

5.2 Architecturally Significant Design Packages 19

6. Process View 19

7. Deployment View 20

8. Implementation View 20

9. Data View (optional) 21

10. Size and Performance 21

11. Quality 21

Software Architecture Document

# Introduction

## Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

## Scope

The document applies to Eyeconic - Glasses Shop Management System, for development phase and user-guide documentating phase.

## Definitions, Acronyms, and Abbreviations

* User - anyone accesses the system
* Administrator/admin - the user with manager priviledge, authenticated for setting up products and orders

## References

None

## Overview

In the following section, architectural design of the system shall be provided in details. Firstly, the document defines the primary software architecture. There are further discussions about the goals and constraints will be imposed upon the quality of the final product afterwards. In the precedence sections, the key views of the system are demonstrated to depict different aspects of the system. Lastly, criteria concerning with size, performance and quality of the system will be proposed.

# Architectural Representation

This documents presents the architectural as a series of mandatory views: Use-Case View, Logical View,

Deployment View and Data View. These views are presented as StarUML and use the Unified Modeling Language (UML).

**Use-Case View**

* Audience: all the stakeholders of the system, including the end-users
* Area: describes the set of scenarios and/or use cases that represent significant, central functionality to the system
* Related artifacts: Use-Case Model, Analysis Model, Use-Case-Realization documents

**Logical View**

* Audience: designers, programmers
* Area: functional requirements: describes the design’s object model
* Related artifacts: Design Model

**Deployment View**

* Audience: deployment managers, system administrators
* Area: topology: describes the mapping of the software onto the hardware and shows the systems’ distributed aspects
* Related artifacts: Deployment Model

**Data View**

* Audience: data specialists, database administrators.
* Area: persistence: describes the architecturally significant persistent elements in the data model.
* Related artifacts: Data Model

# Architectural Goals and Constraints

There are several key requirements and system constraints:

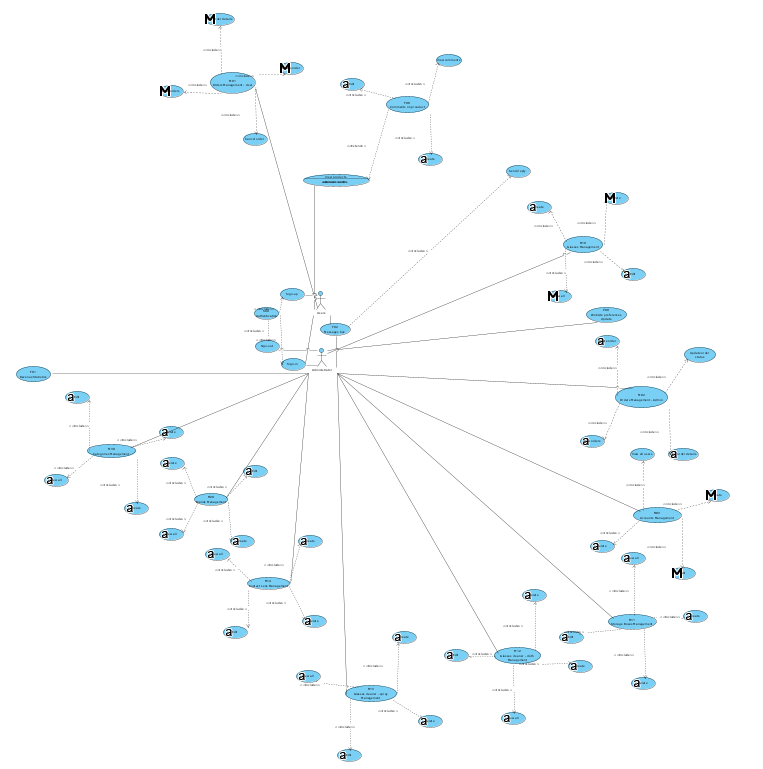
* Architecture design must fulfill all system requirements specified in 30\_Requirement-Definition
* The application must be scalable to satisfy the requirement changes
* System must be designed to take advantages of MVC model
* The prototype comes with this design must be ported to the full system with suitable interface and functions

# Use-Case View

The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage (that exercise many architectural elements) or that stress or illustrate a specific, delicate point of the architecture.

Since this system was built for a online glasses store, then the most significant use-cases included in this design at the time being are:

* M01: User orders mangament
* M02: Admin orders management
* M03: Users (accounts) management
* M10: Glasses management
* M11: Storage boxes management
* M12: Glasses cleaner - cloth management
* M13: Glasses cleaner - spray management
* M14: Contact lens management
* M20: Brands (manufacturers) management
* M30: Categories (menu) management
* F00: Reviews/comments on product (sub-system)
* F01: Revenue statistics
* F02: Messages box (customer care) (sub-system)
* S00: Authentication (sign-in/sign-up/sign-out)
* P00: Website preferences

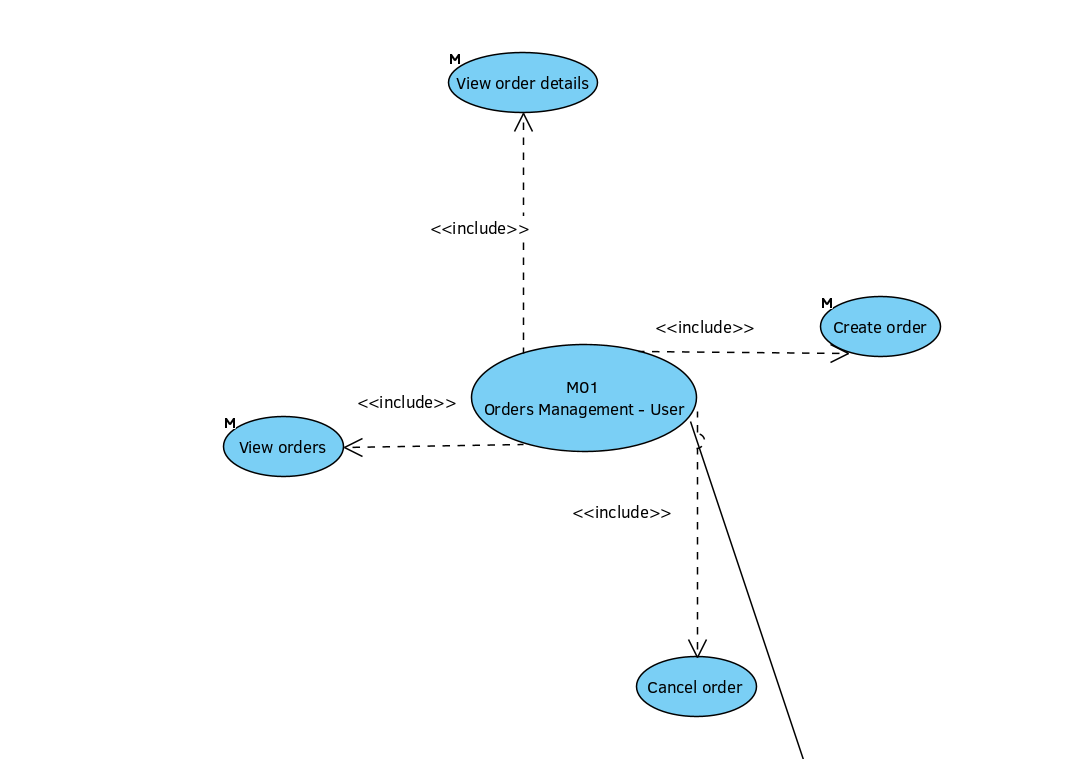


Please refer to **~/20\_Analysis-Design/Use-Case-Diagram.vpp** to take an overview of use-case diagram, the file can be opened with Visual Paradigm.

## Use-Case Realizations

**the symbol (\*) means requires admin priviledge**

* M01 - User orders management



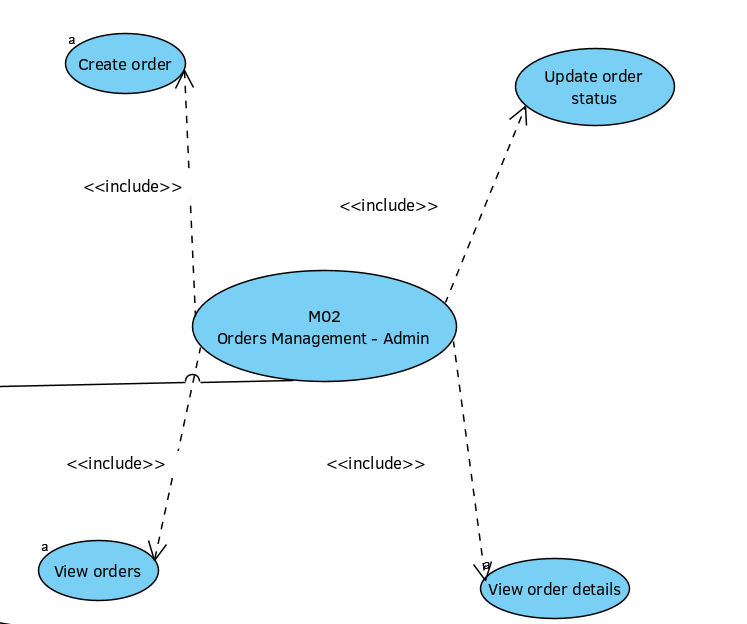
Create: user choose product and add to cart, then fill in payment/shipment information to create an order

Cancel: user checkout order detail and click a button to cancel the order

View orders: view all created orders

View order details: see orders’ information, including shipment detail, items list, quantity, price, etc..

* M02 - Admin orders management (\*)



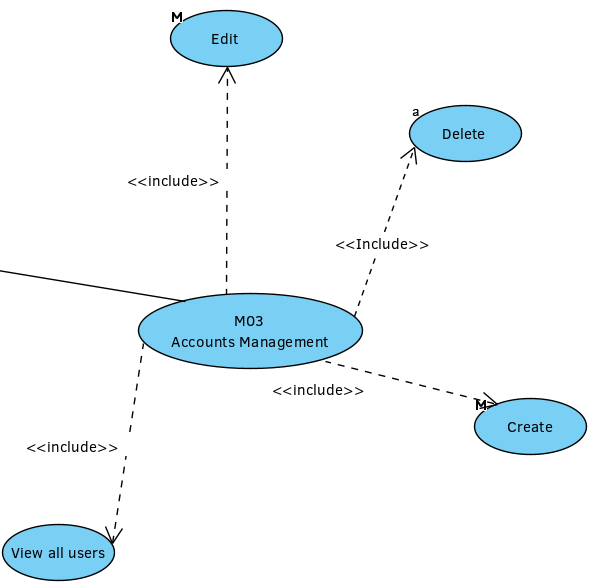
Create: admin can create new order

Update status: update status based on real shipment state (pending, shipping, completed, failed, rescinded)

View orders: view all created orders

View order details: see orders’ information, including shipment detail, items list, quantity, price, etc..

* M03 - Accounts management (\*)



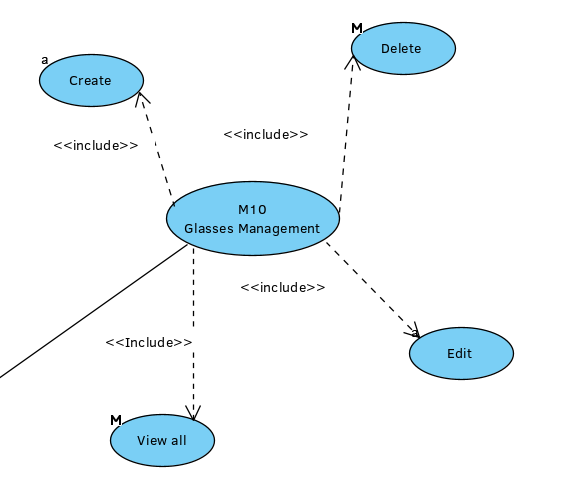
View all: show all accounts

Create: create new account

Edit: edit any users’ information, except username, email and password

Delete: remove user from system

* M10 - Glasses management



View all: show all glasses

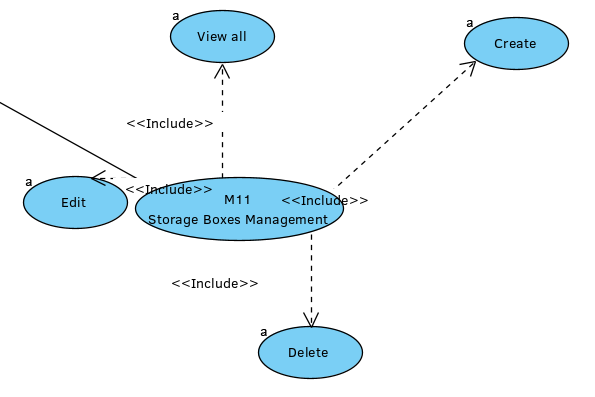
View detail: show product detail

Create: create new glass (\*)

Edit: edit glass information (\*)

Delete: remove glass from system (\*)

* M11 - Storage boxes management



View all: show all storage boxes

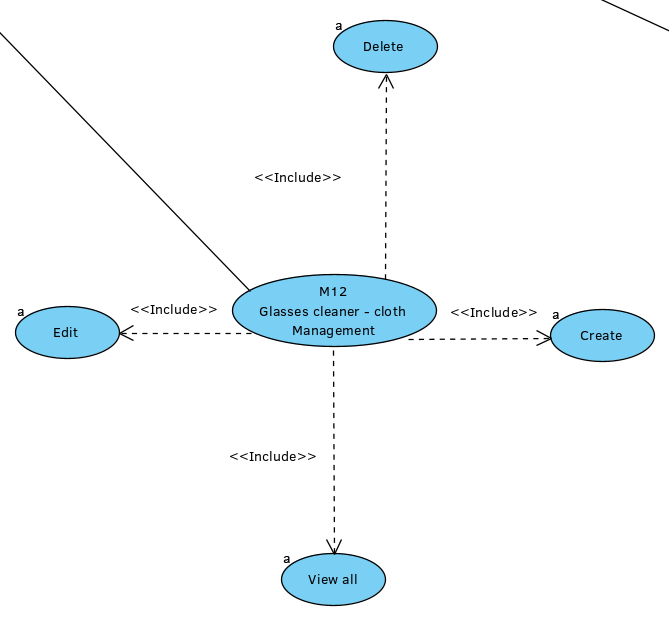
View detail: show product detail

Create: create new storage box (\*)

Edit: edit storage box information (\*)

Delete: remove storage box from system (\*)

* M12 - Cloth glasses cleaner management



View all: show all cloth cleaners

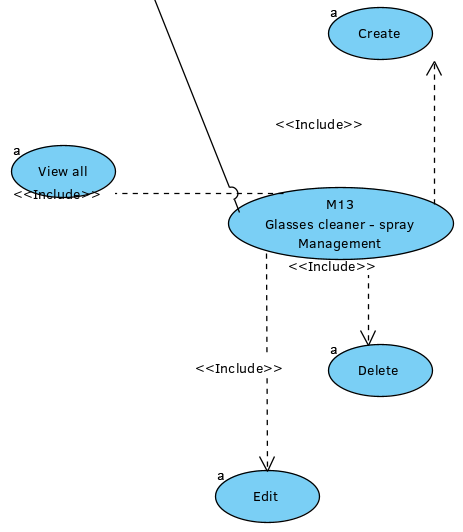
View detail: show product detail

Create: create new cloth cleaner (\*)

Edit: edit cloth cleaner information (\*)

Delete: remove cloth cleaner from system (\*)

* M13 - Spray glasses cleaner management



View all: show all spray cleaners

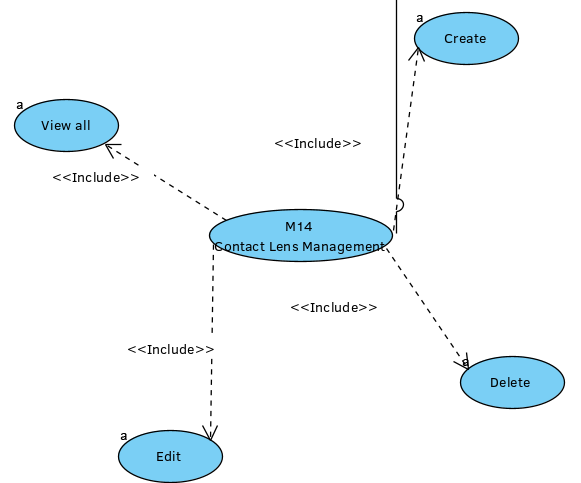
View detail: show product detail

Create: create new spray cleaner (\*)

Edit: edit spray cleaner information (\*)

Delete: remove spray cleaner from system (\*)

* M14 - Contact lens management



View all: show all contact lens

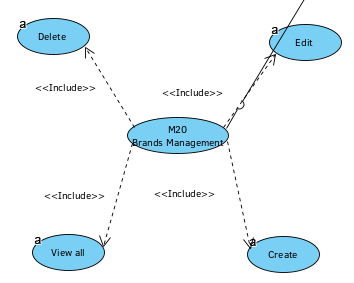
View detail: show product detail

Create: create new contact lens (\*)

Edit: edit contact lens information (\*)

Delete: remove contact lens from system (\*)

* M20 - Brands management



View all: show all brands

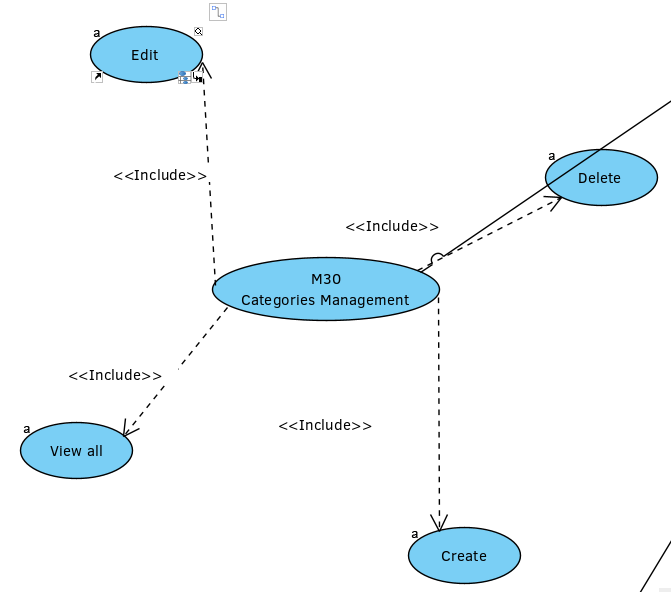
View detail: show product detail

Create: create new brand (\*)

Edit: edit brand information (\*)

Delete: remove brand from system (\*)

* M30 - Categories management



View all: show all categories

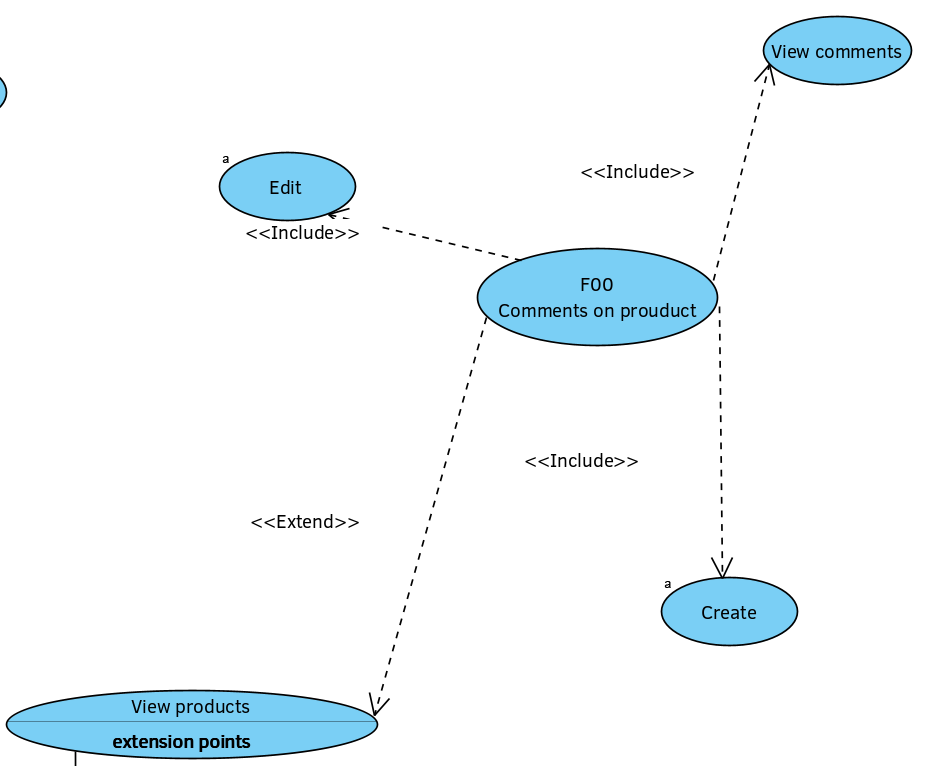
View detail: show product detail

Create: create new category (\*)

Edit: edit category information (\*)

Delete: remove category from system (\*)

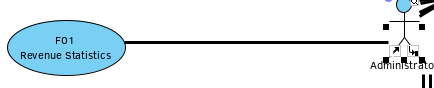
* F00 - Comments on product (user only)



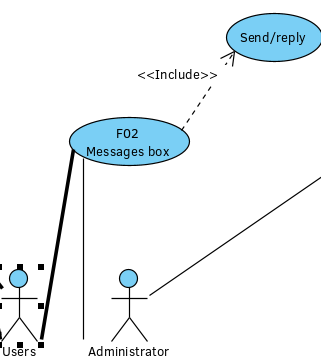
View all: show all comments of a specific product

Create: user create comment

* F01 - Revenue statistics (\*): admins can track their stores’ revenue, by years, by months.



* F02 - Messages box



For user: user can send and read messages of the conversation between him and administrator

For admin: he can send and read messages from all users sent to him

* S00 - Authentication

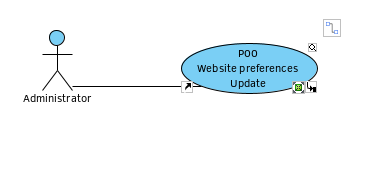


Sign-in: login and take access to system

Sign-up (user only): register a new account

Sign-out: destroy session

* P00 - Website preferences: updates some settings



# Logical View

## Overview

A description of the logical view of the architecture. Describes the overall decomposition of the design

model in terms of package hierarchy and layers.

While taking advantages of MVC design pattern, the architecture of Eyeconic is comprised of 3 significant layers:

* **Model**: classes directly manages data
* **View**: the layer with attractive markup engaging users to interact with system
* **Controller**: processes all logic and requests, manipulates data through Model layer and prepares it for the View



## Architecturally Significant Design Packages

**Model layer:**

since the diagram is too large, please refer to **~/20\_Analysis-Design/Model-Layer.vpp**

**View layer:**

since the diagram is too large, please refer to **~/20\_Analysis-Design/View-Layer.vpp**

**Controller layer:**

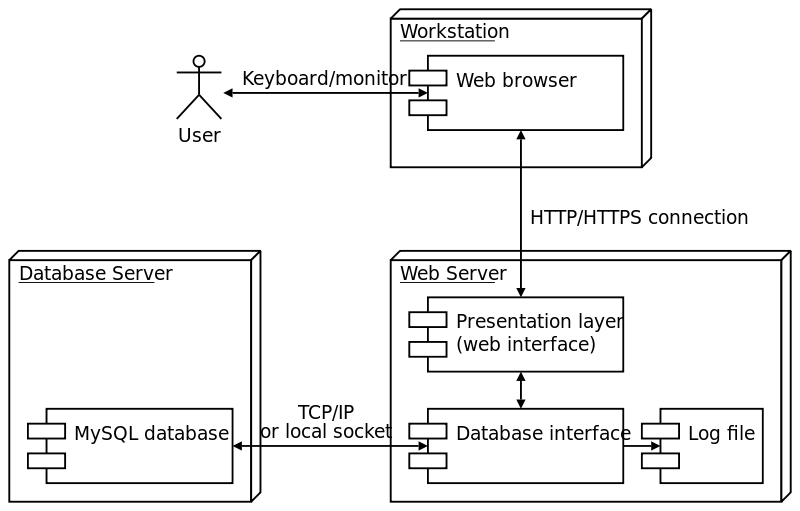
since the diagram is too large, please refer to **~/20\_Analysis-Design/Controller-L ayer.vpp**

# Process View



1. The index.php serves as the front controller, initializing the base resources needed to run CodeIgniter.
2. The Router examines the HTTP request to determine what should be done with it.
3. If a cache file exists, it is sent directly to the browser, bypassing the normal system execution.
4. Security. Before the application controller is loaded, the HTTP request and any user submitted data is filtered for security.
5. The Controller loads the model, core libraries, helpers, and any other resources needed to process the specific request.
6. The finalized View is rendered then sent to the web browser to be seen. If caching is enabled, the view is cached first so that on subsequent requests it can be served.

# Deployment View

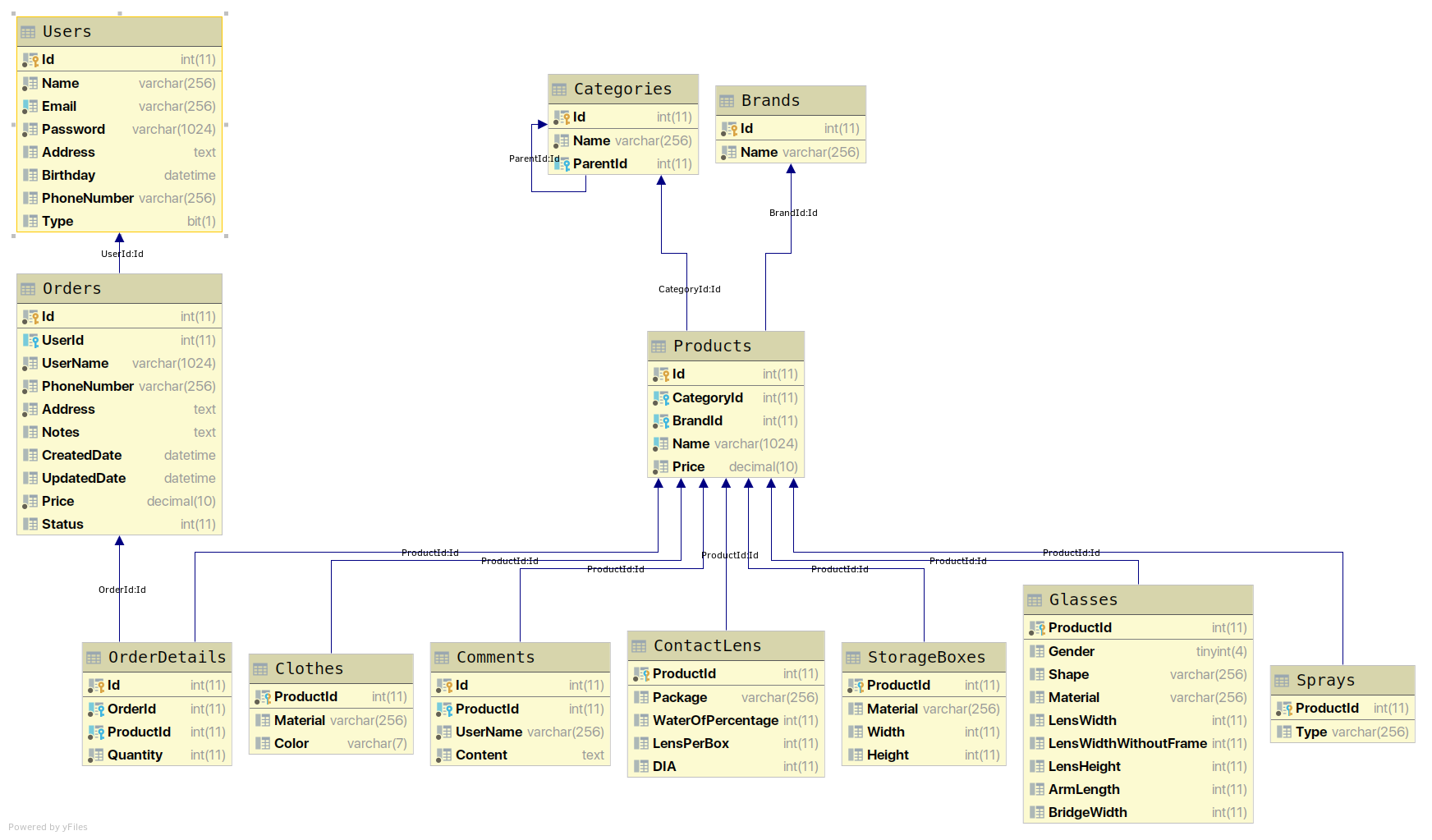


# Implementation View

The implementation of the system is strictly driven from the design; therefore, the implementation view

will not be considered in this document.

# Data View (optional)



# Size and Performance

The major dimensioning characteristics of the software that impact the architecture and performance

constraints:

* The system shall support up to 1000 concurrent users against the primary database at any given
* time, and up to 500 concurrent users against the local servers at any one time
* The system must perform all functions with minimal time delays
* The system must also accurately save all information transactions

# Quality

The system architecture supports the quality requirements:

* In order to maintain the highest degree of system integrity, the system is capable of ensuring that all information transitions are saved
* Databases will be backed up on a daily basis in concern with safety implications
* The system website is capable of display correctly on different devices web browser of any screen size (i.e. responsive design)
* All system website functions are available through popular web browsers; for instance, Google Chrome, Mozilla Firefox, Opera, Safari, Microsoft Edge, Internet Explorer