User Management Design

by

Zymr Inc.

Table of Contents

[1. Introduction 3](#__RefHeading__261_667133547)

[1.1 Purpose 3](#__RefHeading__212_667133547)

[1.2 Scope 3](#__RefHeading__214_667133547)

[1.3 Acronyms 3](#__RefHeading__216_667133547)

[2. Architecture 4](#__RefHeading__218_667133547)

[3. Use Cases 5](#__RefHeading__220_667133547)

[3.1 Login as Regular user 5](#__RefHeading__263_667133547)

[3.2 Login Admin user 5](#__RefHeading__267_667133547)

[3.3 User Logout 5](#__RefHeading__269_667133547)

[3.4 List Users 5](#__RefHeading__271_667133547)

[3.5 Create User 5](#__RefHeading__273_667133547)

[3.6 Delete User 6](#__RefHeading__275_667133547)

[3.7 Edit User 6](#__RefHeading__277_667133547)

[4. Software 7](#__RefHeading__222_667133547)

[4.1 Python 7](#__RefHeading__224_667133547)

[4.2 Python Flask 7](#__RefHeading__226_667133547)

[4.3 PeeWee 7](#__RefHeading__228_667133547)

[4.4 Backbone.js 7](#__RefHeading__230_667133547)

[5. References 8](#__RefHeading__232_667133547)

# **1. Introduction**

### 1.1 Purpose

This document contains the Design and Architecture for the UserManagement module. It also contains the softwares used, their justifications and the Use Case scenarios for the User Management module.

The UserManagement module described in this document is so far an internal project with no real customer deployments. However, this framework and learnings from this project can be used when deploying actual solutions.

### 1.2 Scope

This document provides an Architectural overview for the UserManagement module created for Zymr Inc.

The main idea behind developing this project was to learn the various aspects of developing software in the real world.

### 1.3 Acronyms

# 2. Architecture



HTML and Backbone.js

Python/Flask Web Service

Date Access Object (DAO) Layer

ORM

# 3. Use Cases

The following are the Use Case scenarios for the UserManagement module :

### 3.1 Login as Regular user

Pre-Condition : Atleast one regular exists on the system

Basic Flow :

A Regular user enters his credentials (username and password) and Logs in from the Login Page

Alternate Flow :

The user enters invalid credentials.

In this case, he will not be logged in and will not be able to browse any page

### 3.2 Login Admin user

Pre-Condition:

Basic Flow:

Alternate Flow:

### 3.3 User Logout

Pre-Condition:

Basic Flow:

Alternate Flow:

### 3.4 List Users

Pre-Condition:

Basic Flow:

Alternate Flow:

### 3.5 Create User

Pre-Condition:

Basic Flow:

Alternate Flow:

### 3.6 Delete User

Pre-Condition:

Basic Flow:

Alternate Flow:

### 3.7 Edit User

Pre-Condition:

Basic Flow:

Alternate Flow:

# 4. Software

### 4.1 Python

### 4.2 Python Flask

### 4.3 PeeWee

### 4.4 Backbone.js

# 5. References

* Python-Flask : <http://flask.pocoo.org/>
* PeeWee : <http://peewee.readthedocs.org/en/latest/>
* Backbon.js : <http://backbonejs.org/>