User Management Design

by

Zymr Inc.

Table of Contents

[1. Introduction 4](#__RefHeading__261_667133547)

[1.1 Purpose 4](#__RefHeading__212_667133547)

[1.2 Scope 4](#__RefHeading__214_667133547)

[1.3 Acronyms 4](#__RefHeading__216_667133547)

[2. Architecture 5](#__RefHeading__218_667133547)

[3. Use Cases 6](#__RefHeading__220_667133547)

[3.1 Use Case Diagrams 6](#__RefHeading__229_708971564)

[3.1.1 Admin User Login Logout Use Case Diagram : 6](#__RefHeading__231_708971564)

[3.1.2 Admin User Management Use Case Diagram 7](#__RefHeading__338_708971564)

[3.1.3 Generic User Login Logout Use Case Diagram 8](#__RefHeading__340_708971564)

[3.1.4 Generic User Use Case Diagrams 9](#__RefHeading__342_708971564)

[3.2 Use Cases 10](#__RefHeading__344_708971564)

[3.2.1 Login as Regular user 10](#__RefHeading__263_667133547)

[3.2.2 Login Admin user 10](#__RefHeading__267_667133547)

[3.2.3 User Logout 10](#__RefHeading__269_667133547)

[3.2.4 List Users 10](#__RefHeading__271_667133547)

[3.2.5 Create User 10](#__RefHeading__273_667133547)

[3.2.6 Delete User 11](#__RefHeading__275_667133547)

[3.2.7 Edit User 11](#__RefHeading__277_667133547)

[4. Software 12](#__RefHeading__222_667133547)

[4.1 Python 12](#__RefHeading__224_667133547)

[4.2 Python Flask 12](#__RefHeading__226_667133547)

[4.3 PeeWee 12](#__RefHeading__228_667133547)

[4.4 Backbone.js 12](#__RefHeading__230_667133547)

[5. References 13](#__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

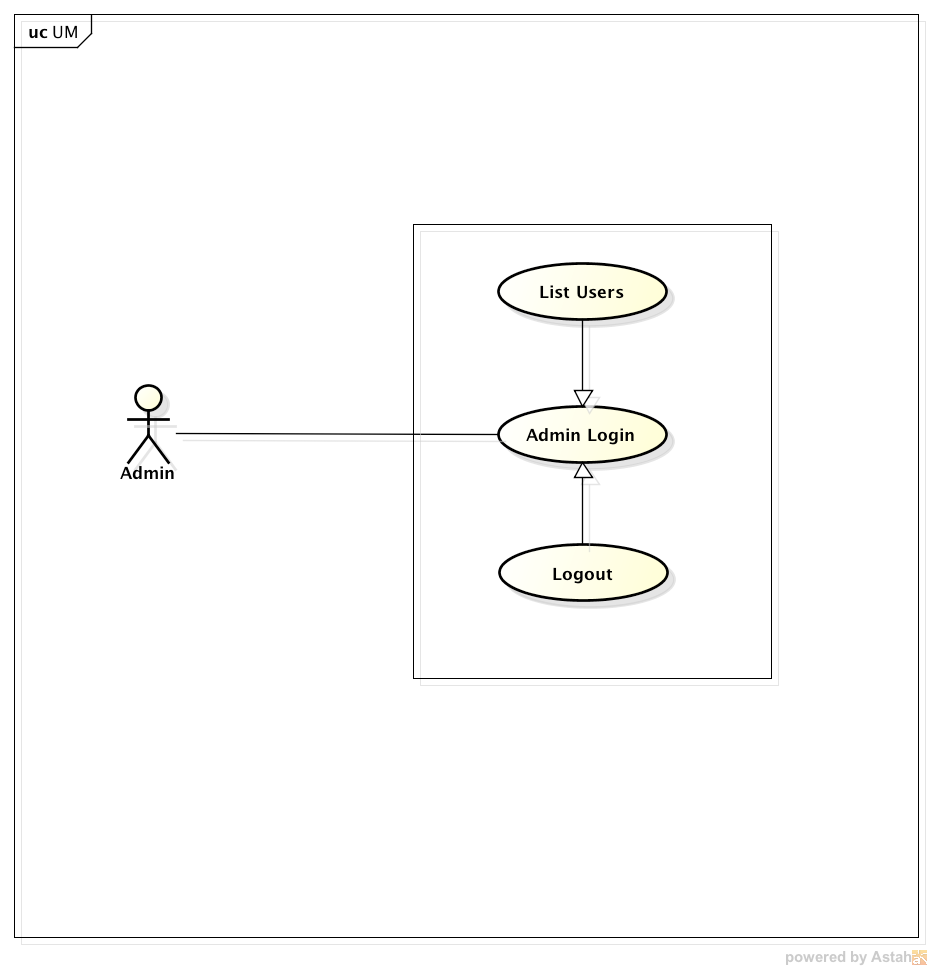
ORM

# 3. Use Cases

## 3.1 Use Case Diagrams

The Use Cases can broadly be divided as per the Use Case Diagrams below :

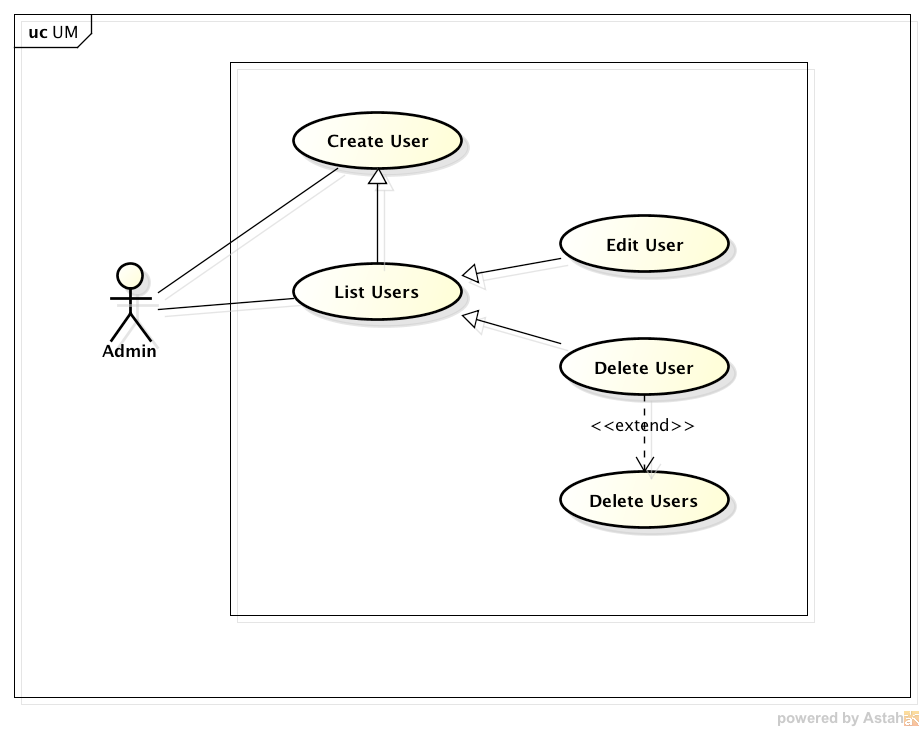
### 3.1.1 Admin User Login Logout Use Case Diagram :

Figure 1: Admin User Login Logout Use Case Diagram

This Use Case Diagram describes the use cases of the Admin user Logging in and out of the system.

As soon as the Admin user logs in, he is presented with the list of users in the system.

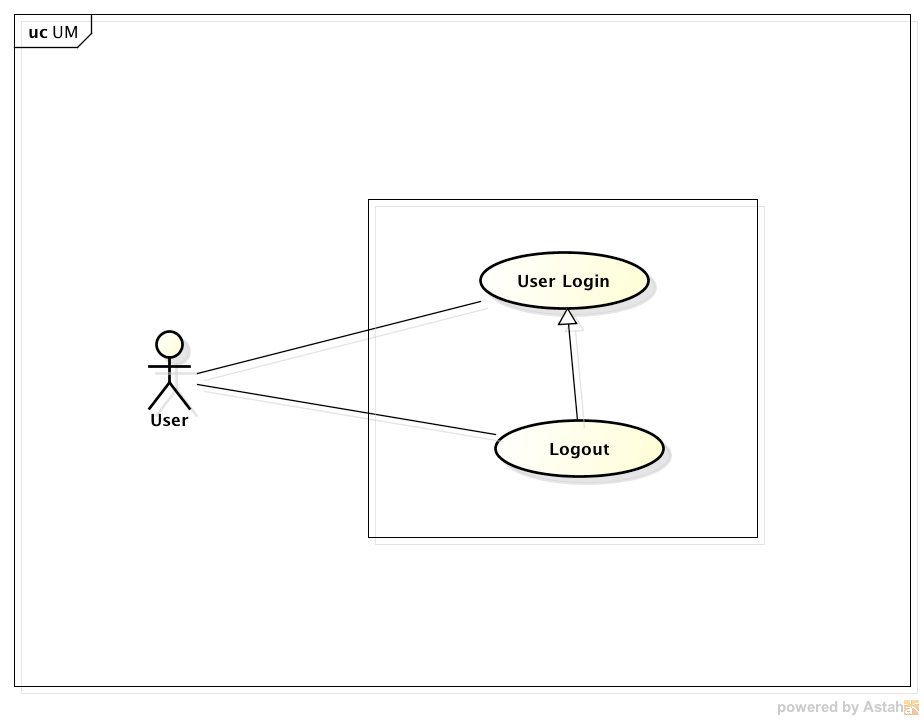
### 3.1.2 Admin User Management Use Case Diagram

Figure 2: Admin User Management UCD

This Use Case Diagram describes the use cases of the Admin user's User Management functions.

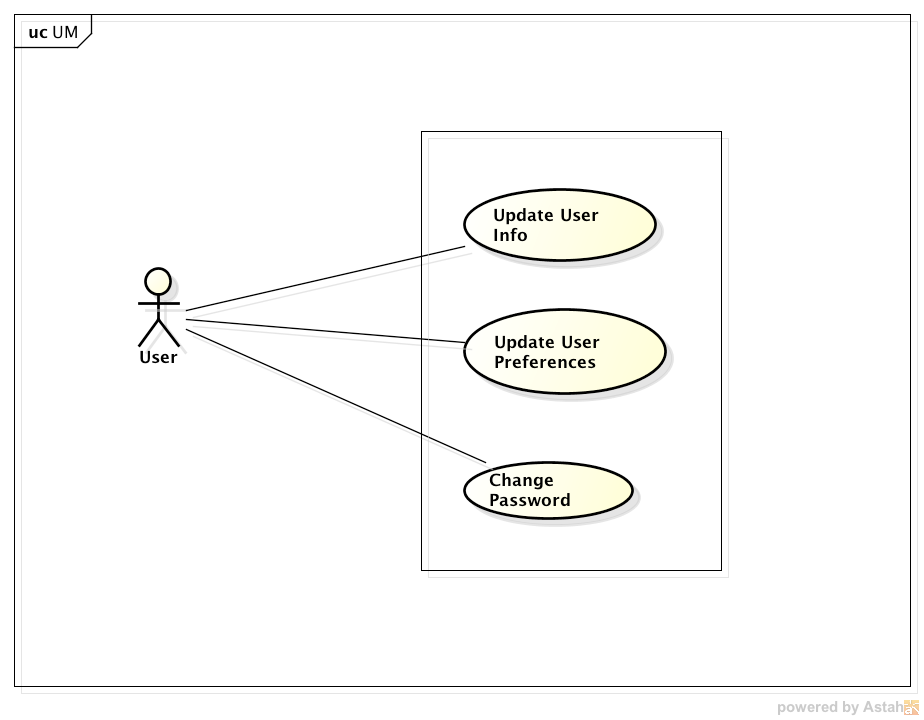
The Admin user can, once logged in, list all the users in the system, create new users, delete one or more existing users, edit the information and preferences of a created user.

### 3.1.3 Generic User Login Logout Use Case Diagram



This Use Case Diagram describes the use cases of any user Loggin in and Logging out.

### 3.1.4 Generic User Use Case Diagrams



This Use Case Diagram describes the use cases of the capabilities of any user(regular or admin).

Every user can update his own User information (like Address, Emails, Phone, etc.) and his own Preferences (like Color, Time, Date, Currency, etc.)

## 3.2 Use Cases

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

### 3.2.1 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

### 3.2.2 Login Admin user

Pre-Condition:

Basic Flow:

Alternate Flow:

### 3.2.3 User Logout

Pre-Condition:

Basic Flow:

Alternate Flow:

### 3.2.4 List Users

Pre-Condition:

Basic Flow:

Alternate Flow:

### 3.2.5 Create User

Pre-Condition:

Basic Flow:

Alternate Flow:

### 3.2.6 Delete User

Pre-Condition:

Basic Flow:

Alternate Flow:

### 3.2.7 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/>