HISTORY PUB

DAN FLOREA
30431

Contents

1. VISION

Introduction

This project intends to create an online environment simulating a pub. The virtual social space is populated with many interesting figures, bearing familiar, yet foreign names. Just like one would do at a social gathering in the real world, one may go to talk to one of the many colorful personalities found at that location. The twist would be, of course, that the people there are historical celebrities. “Walking” into that pub means that one will meet great people such as Cleopatra, Ibn-Battuta, or Mansa Musa.

Positioning

This project aims to introduce people of all ages to historical figures that they may learn from in a fun and engaging way. I strongly believe that one can learn a whole lot from the lives of the people that came before us, but not many are enthusiastic about reading scientific papers and Wikipedia articles about the lives of other people. This online platform aims to offer a quick and engaging way of learning a tiny bit about history, by offering small chunks of information presented in an easily digestible format.

Stakeholders

The target audience is made out of people who wish to have access to a quick read about significant people, all while engaging in some entertaining activities.

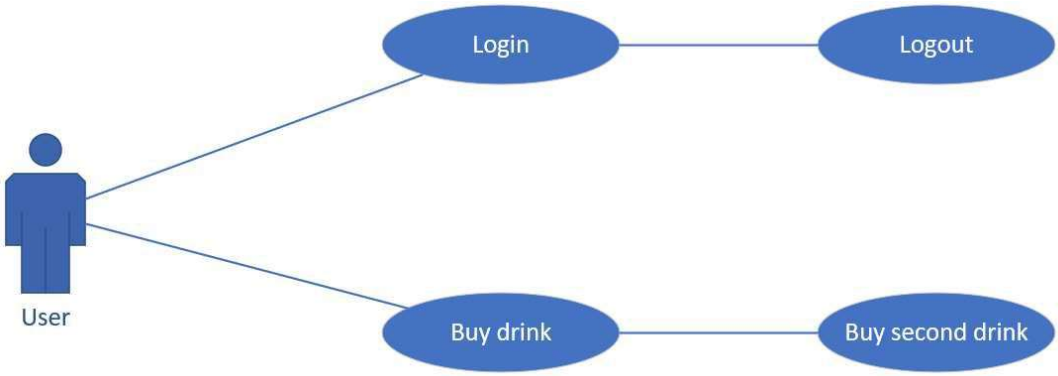
1.1. Use case model

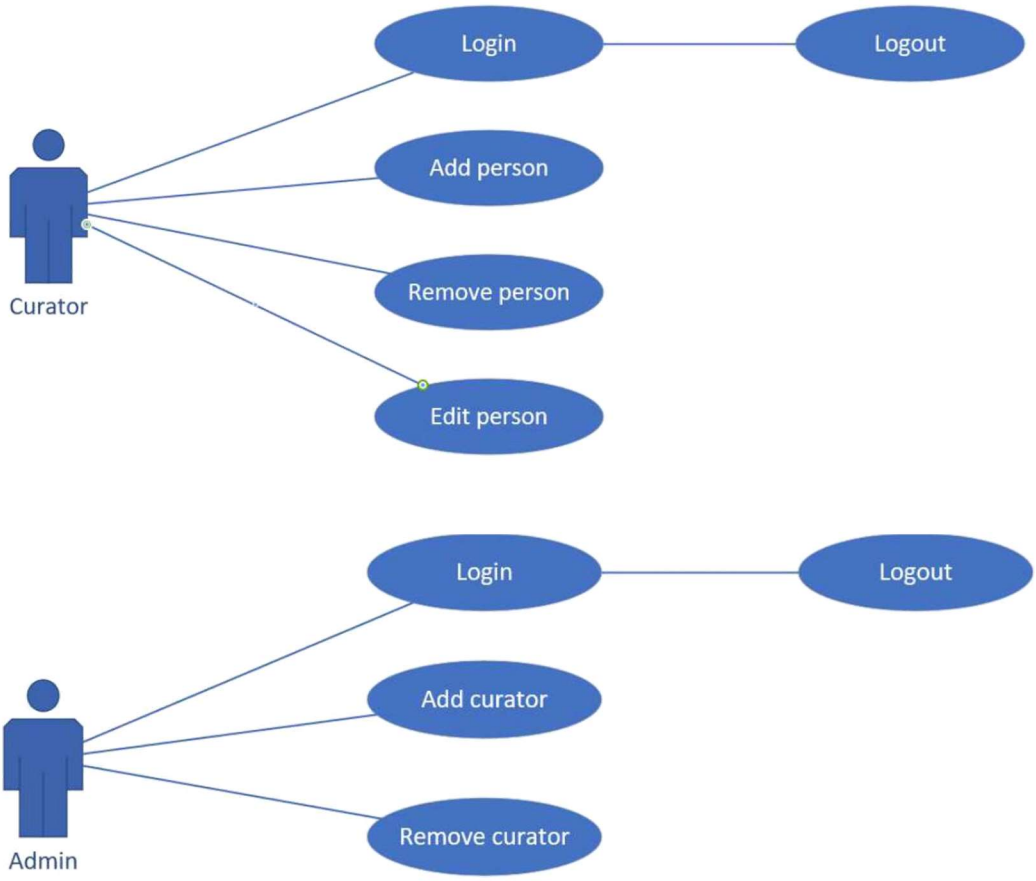
Goal	Buy a drink for one of the people presented to them
Actor	User
Success scenario	The logged in user selects a person that is presented to them on the screen. The user starts "talking" to the chosen person, finding out a small amount of new information about them.
Error scenario	The chosen person is not listed properly (or not found in the database), so the information is not displayed. Pressing the button will result in an error.

Goal	Add historical personality
Actor	Curator
Success scenario	This user presses the "add" button on their curator dashboards. The curator fills in the blanks with the person's details, such as birthday, and information them. This creates a new person to be added to the personality database.
Error scenario	Some fields may be left as blanks, which means the insertion to the database fails.

Goal	Add curator
Actor	Admin
Success scenario	The administrator chooses a user from a list. That user can be promoted to curator, which will update their status. This means that that specific user is given all the rights of a curator.
Error scenario	The user is not found in the user database, which will result in an error.

Diagrams





1.2. Supplementary specification

Design constraints

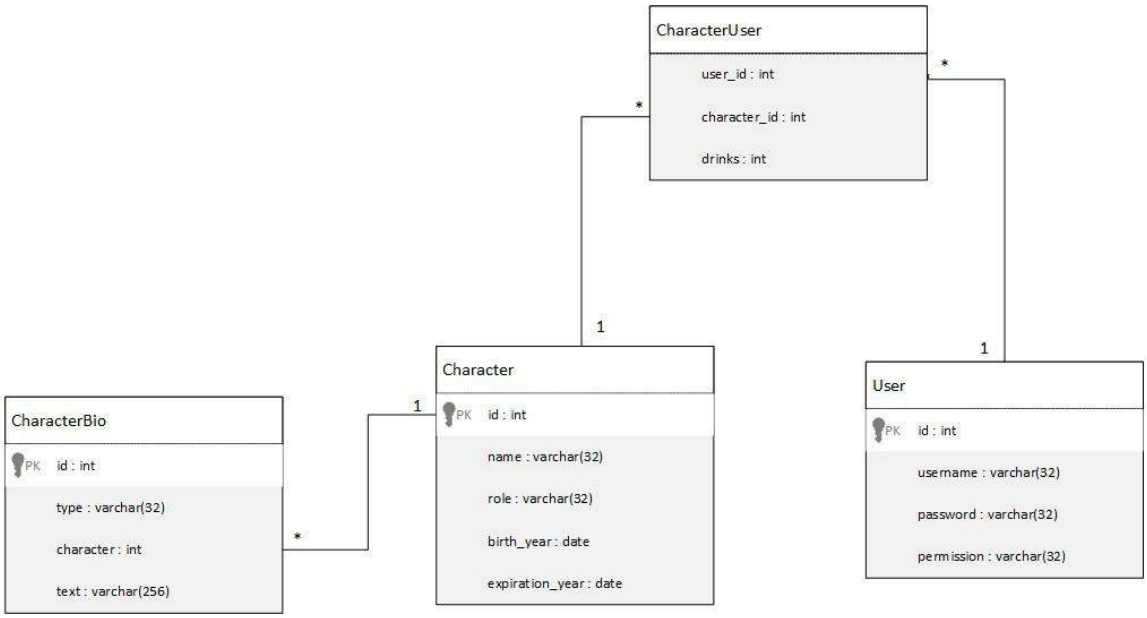
This project will be implemented using the Django framework for Python, Bootstrap, and JavaScript.

Glossary

- Buy a drink (to a person) – the user chooses the person they want to talk to, in order to get to know them better; find out more about that chosen person
- "talking" to a person – just as in a real life at a social gathering, when you start mingling with the other guests, they may talk about themselves; when a user buys a historical person a drink, that person "talks about themselves" (i.e. a small amount of information about that person is displayed)

2. ANALYSIS AND DESIGN

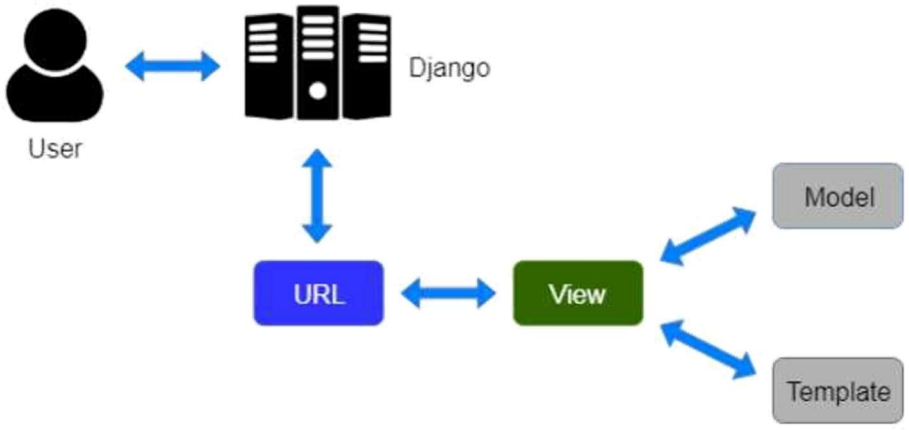
2.1. Domain model



2.2. Conceptual architecture

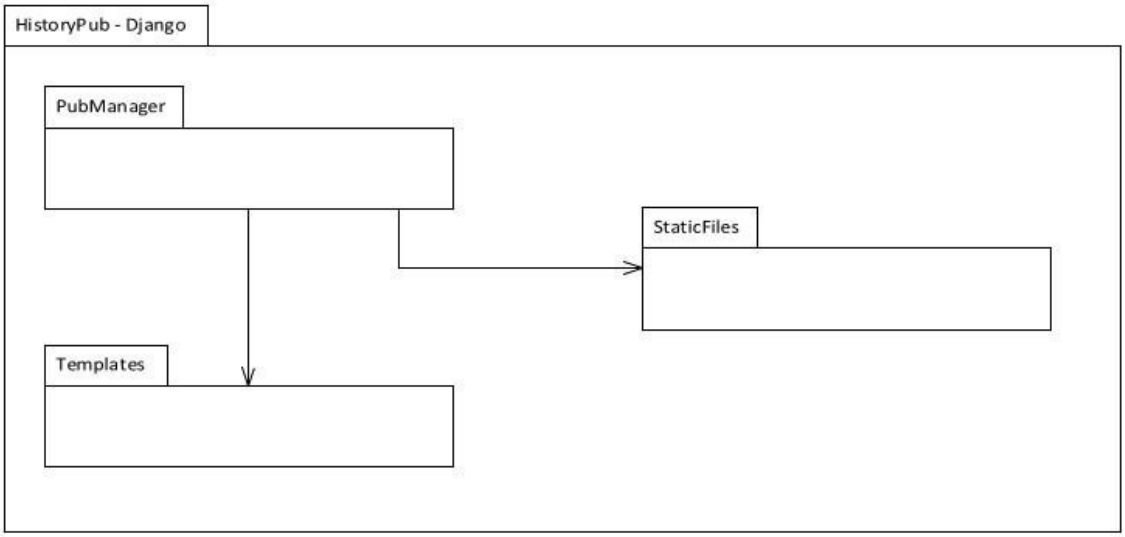
Django is a Python-based free and open-source web framework, which follows the model-view-template (MVT) architectural pattern.

The MVT is a software design pattern. It is a collection of three important components Model View and Template. The Model helps to handle database. It is a data access layer which handles the data. The Template is a presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template. Although Django follows MVC pattern but maintains its own conventions. So, control is handled by the framework itself.

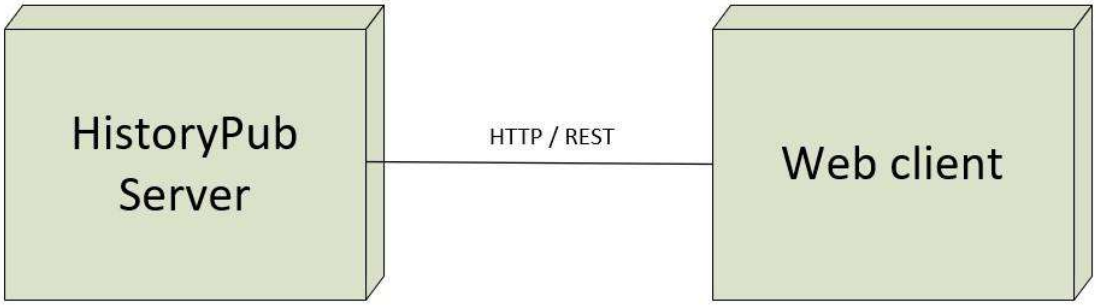


2.3. Package diagram

The fundamental unit of a Django web application is a Django project. A Django project is made up one or more Django apps.

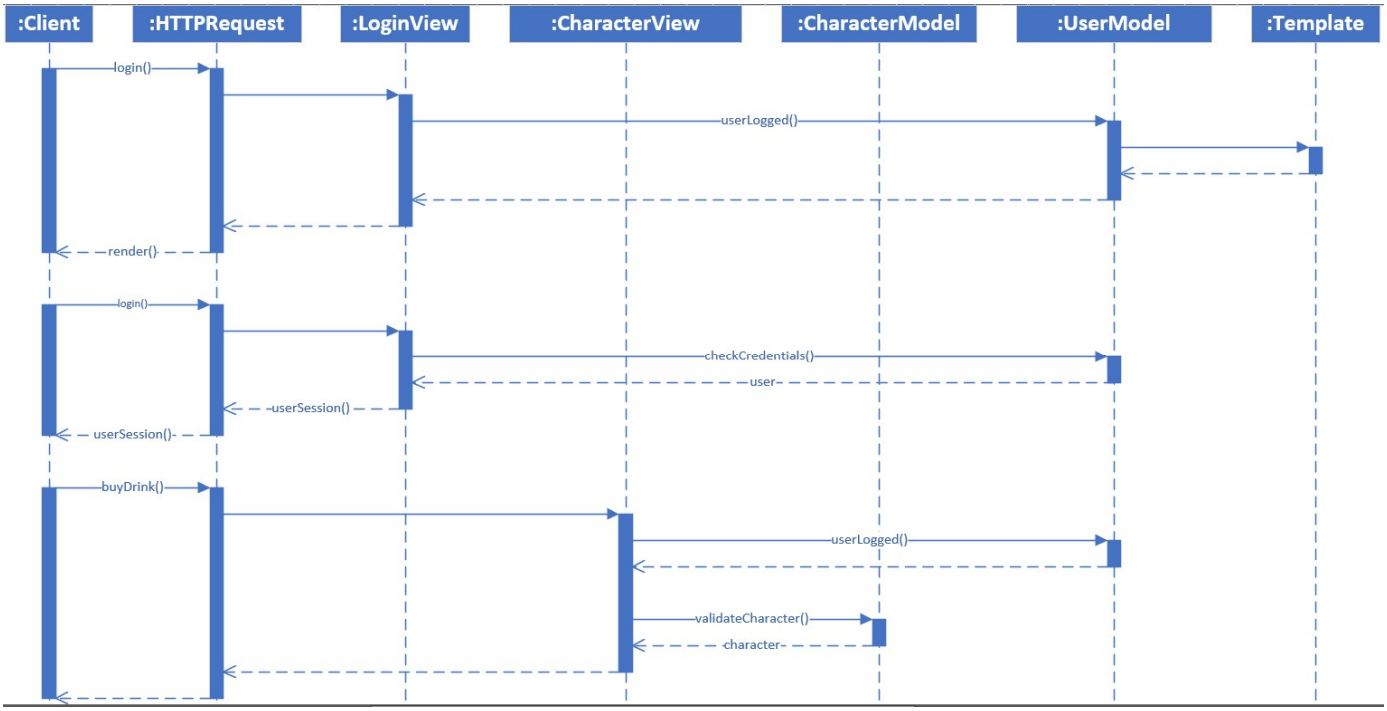


2.4. Components and deployment

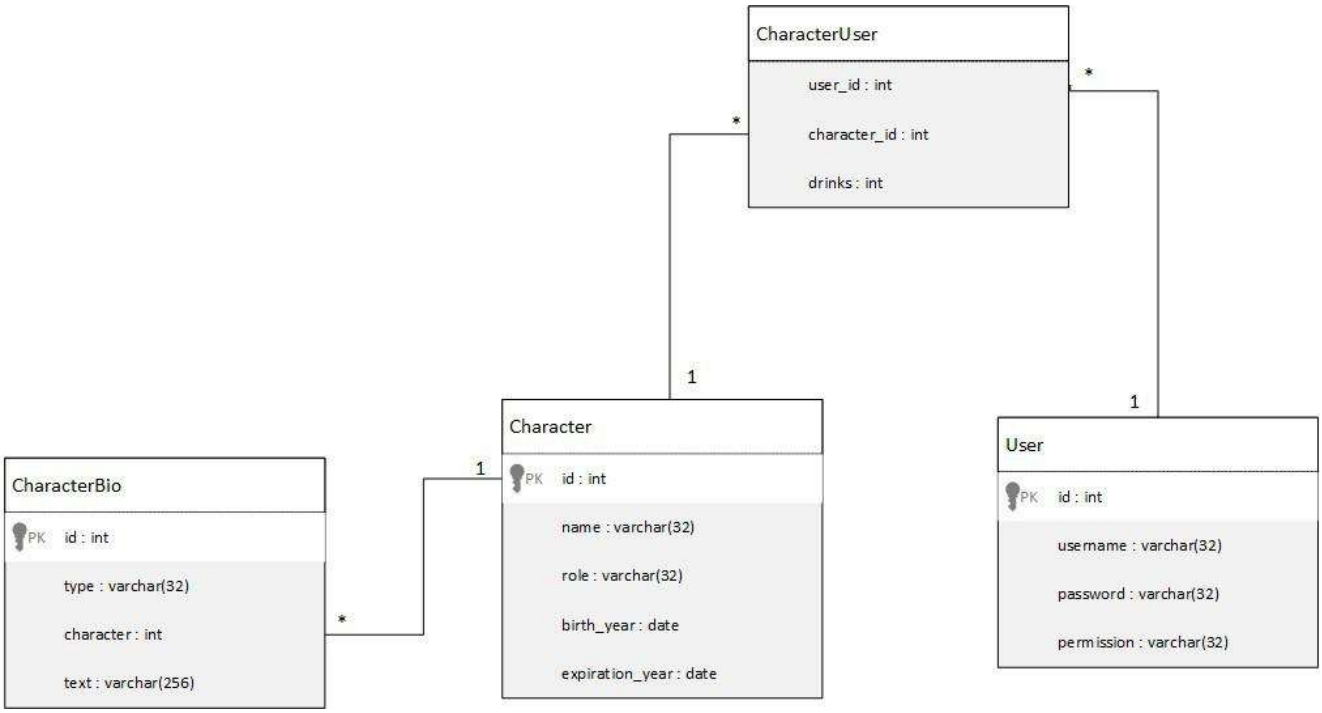


3. DYNAMIC BEHAVIOR

3.1. Interaction diagram

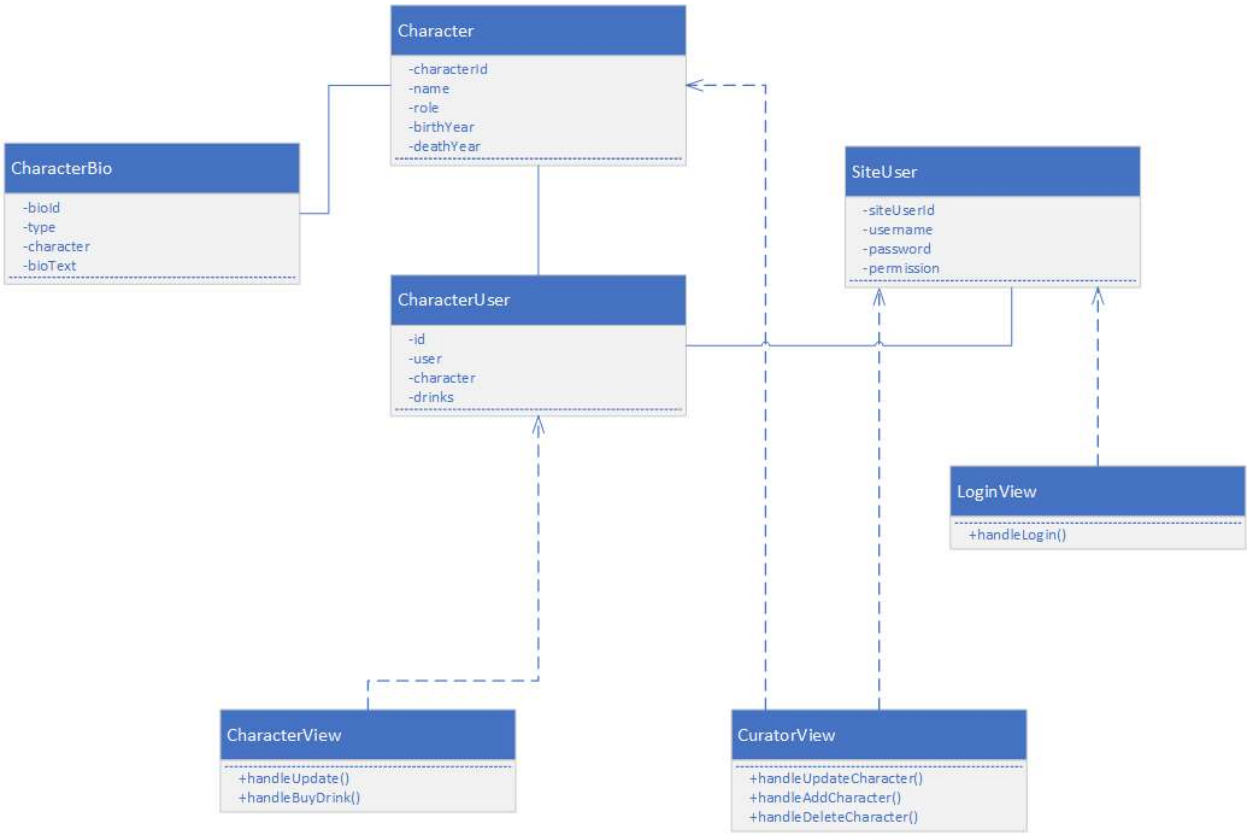


3.2. Data model



3.3. Class diagram

The Django framework integrates the database tables in the class models.



3.4. Test strategy

The project will use some unit tests in order to validate the correctness of the project's basic use cases and functionality.