La Caza del Tesoro

Grupo K:
Tegshigzugder Otgonbayar
Tim Reiprich
Torres Ruiz Daniel Rafael
Gómez Baco José
Yepez Chavez Leticia Denise

January 23, 2020

Contents

1 Introduction

In the project of the course Cloud Application Development we were to develop a web application for a Treasure Hunt Game. By the given requirements we were to use different technologies of ...

2 General description

* deploy over a PaaS cloud * https://github.com/DRTorresRuiz/LaCazaDelTesoro

2.1 Programming languages and frameworks

- project created using the Django framwork because Python every group member knows this language and the framework is supposed to be beginner friendly
- python version and django version
- for certain tasks like the interaction with Google Maps JavaScript had to be used as there is no explicit Python API
- application deployed using Heroku (easy to handle, provides possibly useful add-ons)

2.2 Treasure Hunt

The Treasure Hunt is a competition game in which a group of participants. They must find a series of treasures scattered throughout a play area. The location of each treasure will come indicated by a clue that suggests where it is hidden. The first winner will win the game. So the main goal of the game is to locate all the treasures.

3 Components

The Django project is split into multiple apps. Each app handles one component of the whole project. This is done to get a clear structure of the project and to simplify parallel work. Each of the following sections will explain the functionality and implementation of one specific app.

3.1 Registration

- handles user login, storing of user data and logout
- functionality provided by django-social-auth
- only logout is done using the django standard logout as earlier mentioned module uses the same models

- uses OAuth 2.0 to enable interaction with social network APIs → we used Google+ API
- module redirects new user to Google+ Login homepage, stores known users in database and provides our homepage with needed information about the user
- API can be configured using the Google developer console
- login_required (requires user to log in to access a certain view) and user.is_authenticated (possibility to check if user is logged in) are used in different apps of the project

3.1.1 Homepage

- contains initial view of our website
- shows overview of current and finished games etc.
- provides access to detail views of games and the possibility to log out
- can only be accessed if user is logged in to a Google account, otherwise user is redirected to registration app (login_required before view)

3.2 Chat

- contains functionality of interaction between users in the same game
- add descrption how chat works and why this implementation was chosen
- backend uses Redis to transmit the messages
- locally a docker container providing a Redis server was used but couldn't be launched on Heroku
- Heroku provied add-on "Heroku Redis" which was used to launch a Redis instance in the cloud to enable transmitting messages without requiring a local Redis server

3.3 Game

- provides all functionality in relation to a single game: contains details, join, leave views as well as the views to create games and treasures
- ...

4 User manual

- required setup (?)
- explanation of user interface

5 Limitations

- limitations of our project
- difficulties encountered
- possible improvements in the future

6 Summary

Through this project we were able to
It is very interesting to study a