ParkSmart

CSCD350 Developer Documentation

Team 2 Boolean Bros

Submitted by:

Nicholas Burlakov	**nburlakov@ewu.edu**
Tyler Woody	**twoody1@ewu.edu**
Caleb Stewart	**cstewart15@ewu.edu**
Austin Harrison	**aharrison13@ewu.edu**
Dillon Pikulik	**dpikulik@ewu.edu**

Dr. Kaur Sanmeet

Computer Science and Electrical Engineering Department

Eastern Washington University
June 2024

Table Of Contents

1. Developer Documentation

1.1 How To Obtain Source Code

To obtain the source code, simply make a clone of the repository via:

git clone https://github.com/Sanmeet-EWU/github-teams-project-bid-boolean-bros.git

1.2 Layout Of Directory Structure

/Resources: All diagrams and papers for the project.

/src: Project files.

    /checkParking: Never implemented but checks to see if car is blocking QR code

    /static: Java Scripts, images, and CSS.

        /images: Images.

    /templates: HTML pages.

1.3 How To Build The Software

In PyCharm open the project github-teams-project-bid-boolean-bros/src

Change Run to Current File

Open app.py

Click Run

Go to URL specified in the running console.

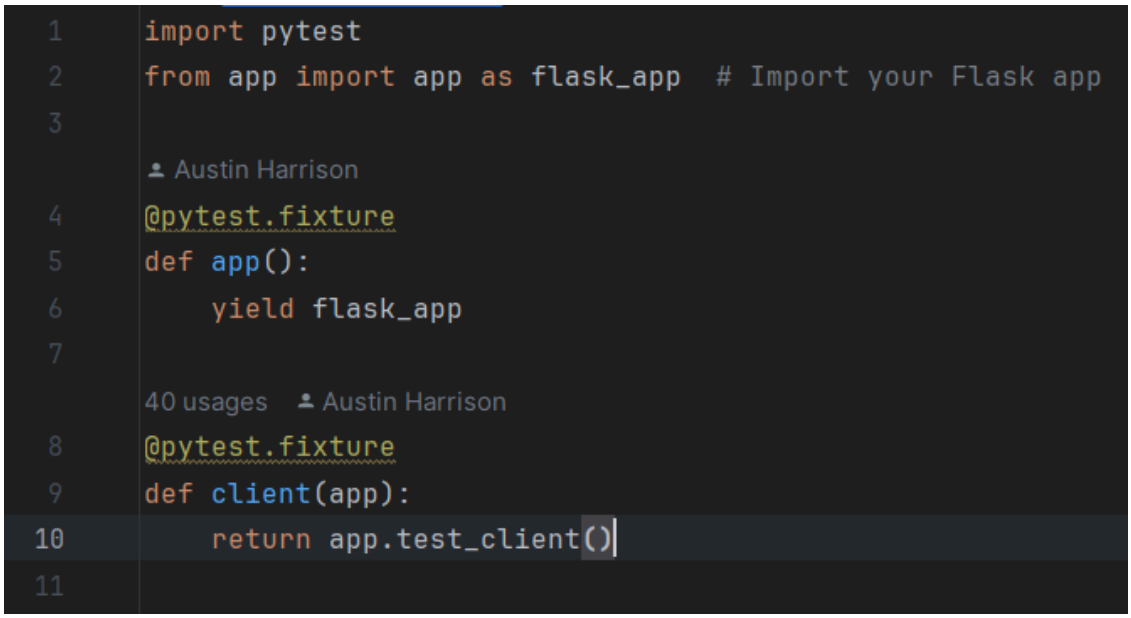
1.4 How To Test Software

Prerequisites:

- Ensure that Python and Flask are installed on the system.
- Ensure that sqlite3 and PyTest libraries are installed in the environment.

Test Configuration:

- Ensure that you have a configuration test file named `conftest.py` that looks like this-



Running The Tests:

- Ensure you have all of the files in an IDE, and all packages installed and imported.
- Open the `test_app.py` file in the directory.
- Right click on `test_app.py` and click run.
- All tests should execute and run from here.

1.5 How To Add New Tests

When adding new tests to the ParkSmart system, follow these guidelines to maintain consistency and clarity.

Naming Conventions:

- Test files should be named using the following pattern: `test_featurename.py`, where `featurename` describes the functionality being tested.
- Within each test file, individual test functions should be named using the pattern: `test_specificBehavior_context`, example - `test_login_success_validCredentials`.

Test Framework:

- Use the PyTest framework for writing and running tests. This choice is due to its powerful features and simplicity in writing scalable tests.
- Place new test files within the `flaskwDB` directory to ensure they are recognized by the PyTest runner.
- Utilize fixtures for setup and teardown processes to manage test preconditions and cleanup.

Writing Tests:

- Each test should ideally cover a single aspect of functionality.
- Assert statements should be clear, directly reflecting the expected outcomes.

1.6 How To Build A Release Of The Software

Building a release of ParkSmart involves several steps:

1. Update Version Number
    - Before building a new release, update the version number in the `README` file and any related documentation that references the software version.
2. Build the Project

- Ensure that all files are pulled from the repository.
- Ensure that all local changes are committed to the repository.

3. Sanity Checks

   After building the project, perform the following sanity checks:

   - Ensure that the generated packages are correct and include all necessary files (Example would be the database file since we are using a local database).
   - Install the package in a clean virtual environment to test that the installation process works correctly.
   - Run some of the critical test cases to confirm that the main functionalities are working as expected in the packaged version.

4. Documentation Updates
   - Review the `README` and any user guides or API documentation to ensure that all information is up to date and reflects the changes made in the new version.

5. Final Steps:
   - Label the commit corresponding to the release with the version number.
   - Push the changes to the repository.