

СОДЕРЖАНИЕ

1. Введение

1.1. Наименование Программы

1.2. Документ, на основе которого ведется разработка

2. Назначение и область применения

2.1. Назначение Программы

2.2. Краткая характеристика области применения

3. Технические характеристики

3.1. Постановка задачи на разработку программы

3.2. Описание алгоритма и функционирования программы

3.3. Внутреннее устройство

3.4. Организация входных и выходных данных

3.5. Выбор технических и программных средств

4. Ожидаемые технико-экономические показатели

5. Источники, использованные при разработке

Приложение 1. Таблицы с описанием классов и интерфейсов

Приложение 2. Таблицы с описанием методов

Лист регистрации изменений

1. ВВЕДЕНИЕ

1.1. Наименование Программы

Наименование программы: «Разработка сервиса высокоточного позиционирования для задач автономного управления».

1.2. Документ, на основе которого ведется разработка

Основанием для разработки является приказ от 15.01.2019. №2.3-02/1501-03 декана факультета компьютерных наук НИУ ВШЭ

Организация, утвердившая этот документ: Национальный исследовательский университет «Высшая школа экономики», Факультет компьютерных наук, образовательная программа «Прикладная математика и информатика».

Наименование темы разработки: «Разработка сервиса высокоточного позиционирования для задач автономного управления»

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Назначение Программы

Основным назначением программы является решение задачи вычисления высокоточного позиционирования на основе измерений предоставленных пользователем. Программа выполняет вычисление высокоточного позиционирования с помощью движка запускаемого на стороне бэкэнда сайта. Результат вычислений отправляется на почту пользователю в виде двух файлов с информацией о точных координатах статичного объекта.

2.2. Краткая характеристика области применения

В настоящее время для высокоточного определения позиции потребителя навигационной информации используется высокоточное оборудование, обладающее достаточными вычислительными мощностями для проведения расчетов в режиме реального времени. В то же время в задачах, менее требовательных ко времени отклика, использование подобного оборудования оказывается зачастую невозможным из-за его высокой стоимости. Для решения этой проблемы предлагается перенести расчета позиции потребителя в веб-сервис: в этом случае с оборудования потребителя полученные навигационные измерения будут отправляться на сервер, а в ответ от сервера потребитель будет получать рассчитанное местоположение.

Разрабатываемое программное средство является сайтом, предназначенным для выгрузки данных от потребителя на сторону сервиса, расчета высокоточной позиции с использованием полученных данных и предоставлению потребителю результата вычисления точного местоположения. Помимо этого сервис будет предоставлять пользователю статистику проведенного расчета.

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

Целью проекта «Разработка сервиса высокоточного позиционирования для задач автономного управления» (англ. «Development of the precise positioning service for auto-steering») является разработка и создание вебсайта с удобным интерфейсом для загрузки, обработки и выдачи результата вычислений высокоточной позиции статичного объекта. Представляется, что указанный подход при облачном процессинге данных сократит расходы на оборудование для определения местоположения.

3.2. Описание алгоритма и функционирования программы

На главной странице имеется форма логина с возможностью регистрации. После успешной авторизации показывается форма отправки файла с выбором некоторых опций движка. С отправкой файла в фоне создается процесс обработки файла. Внутри него запускается движок, который обрабатывает загруженный файл и выдает результат в виде двух файлов типа .gga и .rtk. Далее начинается обработка выходного .gga файла, из которого мы собираем статистику. Результат обработки кладется в два файла типа .csv и .pdf. Эти же файлы по завершению процесса отправляются на почту пользователю, а также кладутся во внутреннюю файловую систему сайта. Также, на главной странице поддерживается история всех посылок с возможностью скачивания исходного файла и результатов обработки.

3.3. Внутреннее устройство

Внутреннее устройство сайта представляет из себя одно Django приложение main и отдельный инстанс библиотеки для асинхронных задач huey. В приложении 1 данного документа представлена схема используемых баз данных, а также их взаимодействие: Request – таблица хранит информацию о всех посылках всех пользователей, Antennes – хранит список поддерживаемых антенн (загружается администратором через ручку /upload_antennes), Result – содержит информацию о всех результатах для каждой посылки, Status – таблица всех статусов обработки файла, User – хранит всех зарегистрированных пользователей. В приложении 2 описаны все реализованные методы обработки файла и отрисовки сайта.

3.4. Организация входных и выходных данных

На вход от пользователя ожидается файл в формате TPS или Rinex версии 2 или 3. На выход он получает .csv и .pdf файл.

3.5. Выбор технических и программных средств

Был использован свободный фреймворк для веб-приложений на языке Python Django за его гибкость и легкость в эксплуатации, а также огромный набор инструментов для создания сайтов. Также этот набор библиотек очень хорошо справляется с задачей разработки безопасных и устойчивых сервисов.

Использована SQLite в качестве системы управления базами данных, т.к. это стандартная СУБД используемая фреймворком и она заводится из коробки. Для асинхронных задач используется библиотека huey за легкую интеграцию в проект, а также за свою легковесность. По тем же причинам для верстки дизайна выбран CSS фреймворк W3.CSS.

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Предполагаемая потребность

Данный продукт позволяет уменьшить затраты на вычислительное оборудование, ограничившись только измерительными приборами.

4.2. Ориентировочная экономическая эффективность

В рамках данной работы расчет экономической эффективности не предусмотрен.

4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

На момент начала разработки на рынке не было выявлено аналогичных продуктов.

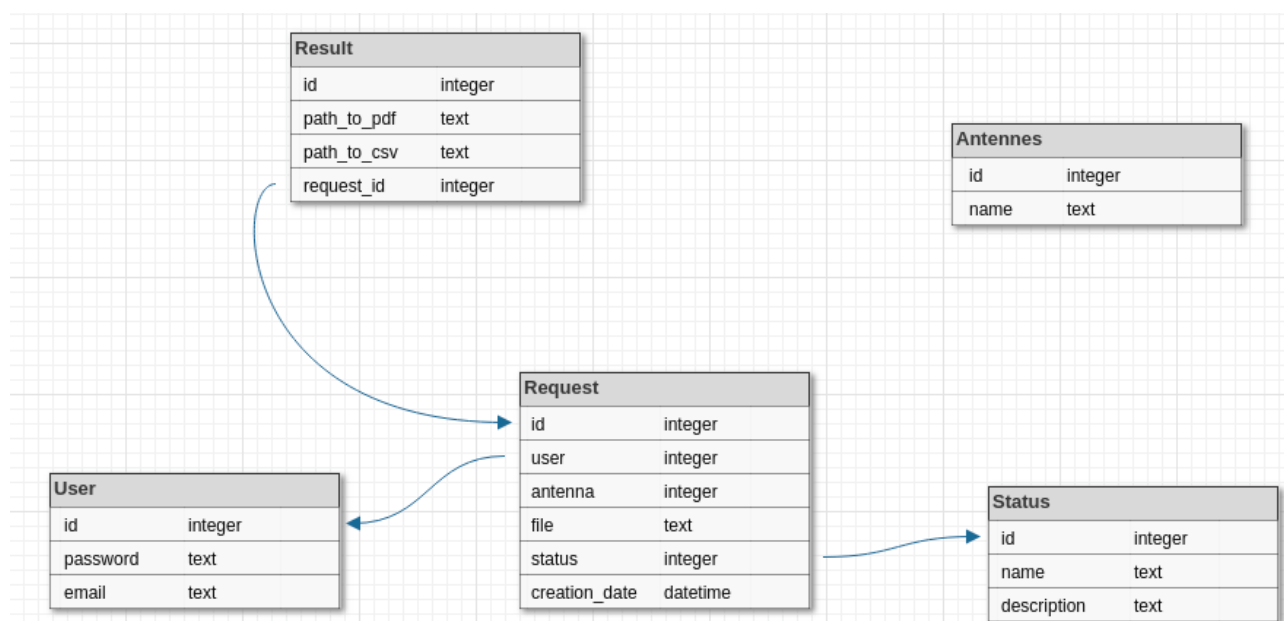
5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКИ

1. <https://www.djangoproject.com>
2. <https://www.w3schools.com>
3. <http://htmlbook.ru>
4. <https://huey.readthedocs.io>

ПРИЛОЖЕНИЕ 1

Таблицы с описанием классов и интерфейсов

Таблица 1. Диаграмма таблиц базы данных



ПРИЛОЖЕНИЕ 2

Таблицы с описанием методов

Метод	Расположение	Входные данные	Выходные данные	Что делает
home	main/views.py	request	страница html	Показывает главную страницу с формой отправки, либо форму авторизации
login	main/views.py	request	страница html	Пытается авторизовать пользователя и вызывает функцию home
register	main/views.py	request	страница html	Показывает форму регистрации и пытается зарегистрировать нового пользователя. При успехе вызывает функцию home
logout	main/views.py	request	страница html	Пытается разлогинить пользователя и вызывает функцию home
upload	main/views.py	request	страница html	Обрабатывает загрузку файла и вызывает функцию home. При успехе загрузки файла

				также вызывает функцию process
result	main/views.py	request	страница html	Показывает страницу результатов конкретной посылки
upload_antennes	main/views.py	request	страница html	Показывает скрытую страницу загрузки списка антенн для администратора, а также обрабатывает загрузку файла с антеннами
process	main/tasks.py	request_id		Выполняет фоновую обработку файла с последующей отправкой результата на почту
process_gga	main/tasks.py	req (Request) – посылка, элементы таблицы Request		Парсер ответа движка
parse_coord	main/tasks.py	coord (string) – строка координаты, is_lat (bool) – флаг, что координата долготы	координата (float)	Парсер координат из .gga файла
parse_time	main/tasks.py	tm (string) – строка времени	секунды (int)	Парсер времени из .gga файла
convert_to_xyz	main/tasks.py	lon, lat, hgt широта в радианах, долгота	tuple с координатами x, y, z	Конвертирует географические координаты в

		в радианах, высота в метрах		декартовы
create_tex_file	main/tasks.py	req (Request) – посылка, элементы таблицы Request x, y, z – координаты x_cov, y_cov, z_cov – ковариации координат x_agg, y_agg, z_agg – массивы координат во все моменты времени sputniks – кол-во спутников в каждый момент времени time_in_sec – список времен в секундах		Создает pdf файл с результатом обработки
create_csv_file	main/tasks.py	req (Request) – посылка, элементы таблицы Request x, y, z – координаты x_cov, y_cov, z_cov – ковариации координат x_agg, y_agg		Создает csv файл с результатом обработки

create_ini_file	main/tasks.py	req (Request) – посылка, элементы таблицы Request		Создает ini файл с настройками движка
main	engine/engine.cpp	в параметрах ожидает путь до ini файла		Заглушка, имитирующая работу движка