

Веб-приложение для идентификации динамических систем

Основной функционал:

- Идентифицировать динамическую техническую систему (объект) по её переходной характеристике (идентифицировать – означает отобразить передаточную функцию системы (объекта)) одним из предлагаемых пользователю методов идентификации.

Дополнительный функционал:

- Отображение частотных характеристик
- Отображение функционала качества
- Выгрузка параметров передаточной функции
- Регистрация пользователей
- Сохранение истории использования приложения

1. Веб-приложение

Веб-приложения разделяются по своей архитектуре или системе организации на следующие виды:

SPA (single page application – одностраничное приложение). Для разработки таких приложений используют HTML¹ и JavaScript². По сути, это разработка лендинга, только интерактивного. Но SPA могут быть достаточно сложными. Суть одностраничных приложений в том, что на сервере хранится одна HTML-страница, контент на которой обновляется по мере прокрутки или переходов по ссылкам. Например, когда пользователь нажимает на кнопку, он не переходит на новую страницу, а новые элементы (или данные) добавляются к уже загруженной. Например, по этому принципу работает Gmail.

¹ (HyperText Markup Language) стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

² Мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Обычно работает на стороне клиента и добавляет интерактивности в контексте отображения элементов на веб-сайте. Скрипты на JavaScript также могут выполнять функцию валидации данных и не только.

MPA (multi page application – многостраничное приложение). Веб-приложение состоит из отдельных страниц, каждая из которых связана с другими гиперссылками. Загрузка данных в таком виде приложений происходит целыми страницами, а не частями, как в SPA. Например, если пользователь после идентификации захочет получить частотные характеристики объекта, нажав на соответствующую кнопку, он перейдет на другую страницу. Если бы то же самое происходило в SPA, то пользователю отобразилось всплывающее окно с частотными характеристиками.

Предпочтение отдаётся **SPA**, так как разработка этого вида веб-приложения экономит время, а также скорость работы приложения будет выше, чем у MPA.

Основной функционал приложения (логика реализации методов) будет реализован на языке программирования **Python**. Так как на основе этого языка уже написаны библиотеки для работы с динамическими системами на основе преобразования Лапласа (в Python реализованы s-модели).

1.1 Клиент-серверное взаимодействие

Взаимодействие клиента с сервером будет происходить по протоколу HTTPS.

Далее будет выбран хост-провайдер для хранения данных на сервере и выдача доменного имени.

1.2 Backend (серверная часть)

Далее будут рассмотрены три фреймворка языка программирования Python, на основе которых разрабатывается серверная часть приложения, то есть backend³. Это Django, Flask и Pyramid.

В фреймворке **Django** встроено множество компонентов для поддержания инфраструктуры веб-приложения, а именно встроены механизмы шаблонизации, формы сбора данных от пользователя, маршрутизация URL-адресов, аутентификация, базовое администрирование баз данных и многое другое. В тоже время, Django

³ Инкапсуляция логики работы веб-приложения на стороне сервера. Эта части веб-приложения реализует передачу данных от базы данных до клиента, реализует обработку данных. В контексте ВКР на стороне серверной части будут реализованы методы идентификации.

предназначен для сложных и больших веб-приложений. Встроенные функции могут быть излишние. Используется в сайтах Instagram, Pintrest, Youtube.

Flask – это минималистичный каркас для создания веб-приложений, предоставляет лишь самые базовые возможности. В него не встроено администрирование баз данных и аутентификация, однако всё это можно восполнить подключаемыми модулями (библиотеками). В этот фреймворк встроены шаблонизатор и диспетчер URL-адресов. Используется в сайтах LinkedIn, Netflix, Uber.

Pyramid – что-то среднее между Flask и Django. По сравнению с Flask, Pyramid предназначен для более сложных приложений.

Предпочтение отдаётся фреймворку **Flask**, так как данный инструмент популярен среди русскоязычного сегмента его пользователей и он нацелен для создания простых веб-приложений. Не исключено, что со временем приложение может быть переписано под Django.

1.2.1 База данных

В приложении будет использована одна из SQL-подобных реляционных баз данных.

1.3 Frontend (клиентская часть)

Frontend – часть отображения пользовательского интерфейса клиенту нашего приложения. Элементы (кнопки, графики, поля заполнения) можно создавать вручную с помощью HTML, CSS и JavaScript, а можно использовать фреймворки. В качестве предполагаемых frontend-фреймворков будут рассмотрены Vue и React – основаны на языке JavaScript. Возможно, в процессе поиска новой информации, автор будет использовать другой каркас для пользовательского интерфейса.

2. Идентификация динамических систем

Для начала планирую реализовать три метода идентификации в своём веб-приложении: идентификация методом наименьших квадратов, идентификация вещественным интерполяционным методом и идентификация градиентным методом.

Данные методы были выбраны так как они все обладают простотой реализации, возможностью численного решения задачи и высокой точностью.

В процессе разработки приложения скорее всего будут добавляться новые методы идентификации.