TBP 101: REQUIREMENTS

Lab 03 - CS 411 @ Boston University

Solution Requirements

○ **Describe** specific characteristics that a product must have to meet the needs of the stakeholders and the business itself. They fall into two large groups.

- **Functional** requirements define what a product must do, what its features and functions are.

- **Nonfunctional** requirements describe the general properties of a system. They are also known as *quality attributes*.

Let's get into some…

HIGH-LEVEL REQUIREMENTS

Utilize a Database:

- **MySQL**
  - *Nothing wrong with keeping it simple*
- **PostgreSQL**
  - *MySQL+*
- **MongoDB**
  - *We like that*
- **Oracle Database**
  - *Ok boomer*
- **Microsoft SQL Server**
  - *What are you a business student?*

Two Publicly Available APIs:

- **Twitter API**
  - *Elon is that you?*
- **Google Maps API**
  - *In all seriousness, this one is actually useful*
- **OpenWeatherMap API**
  - *"Hey Siri, what's the high today?"*
- **OpenAI API**
  - *Maybe Skynet isn't so far off*
- **Spotify Web API**
  - *Apple Music? No thanks. Pandora?! Who are you?*

Third-Party Authentication

- **Oauth (Facebook, Google, Twitter)**
  - *You will actually… no you SHOULD use this*
- **Microsoft Account Authentication (Microsoft Services)**
  - *Someone say Outlook?*
- **Apple ID Authentication (Apple Services)_**
  - *Green bubbles statistically get ghosted*
- **GitHub Authentication (GitHub Services)**
  - *Hey… We've seen this before…*
- **Okta (Identity Management Platform for Third-Party Auth)**
  - *Honestly, I just went down a rabbit hole at this point*

A Decoupled Architecture

- A headless content management system (CMS) that separates content creation from content delivery

- A microservices architecture, where each service is responsible for a specific function and communicates with other services via APIs

- A serverless architecture, where the application's code is split into small, independent functions that are executed in response to specific events or triggers

- **A client-server architecture, where the client and server are loosely coupled and communicate via APIs or other standardized protocols**
  - *This is what you all will be putting together*

RESOURCES...

git add lab03.txt
git commit -m "complete"
git push origin master

Lab 03 - CS 411 @ Boston University

(dcmag@bu.edu)