AnaTwitics

Tom Hood, Jared Edler, Noland Crane, Ryan Huffman

What is AnaTwitics?

- Anatwitics is a web application created using HTML, JavaScript, and Python aimed at users who want to track trends or popularity of a specified hashtag
- We imagine this application would be ideal for business, entrepreneurial, educational, and personal uses.

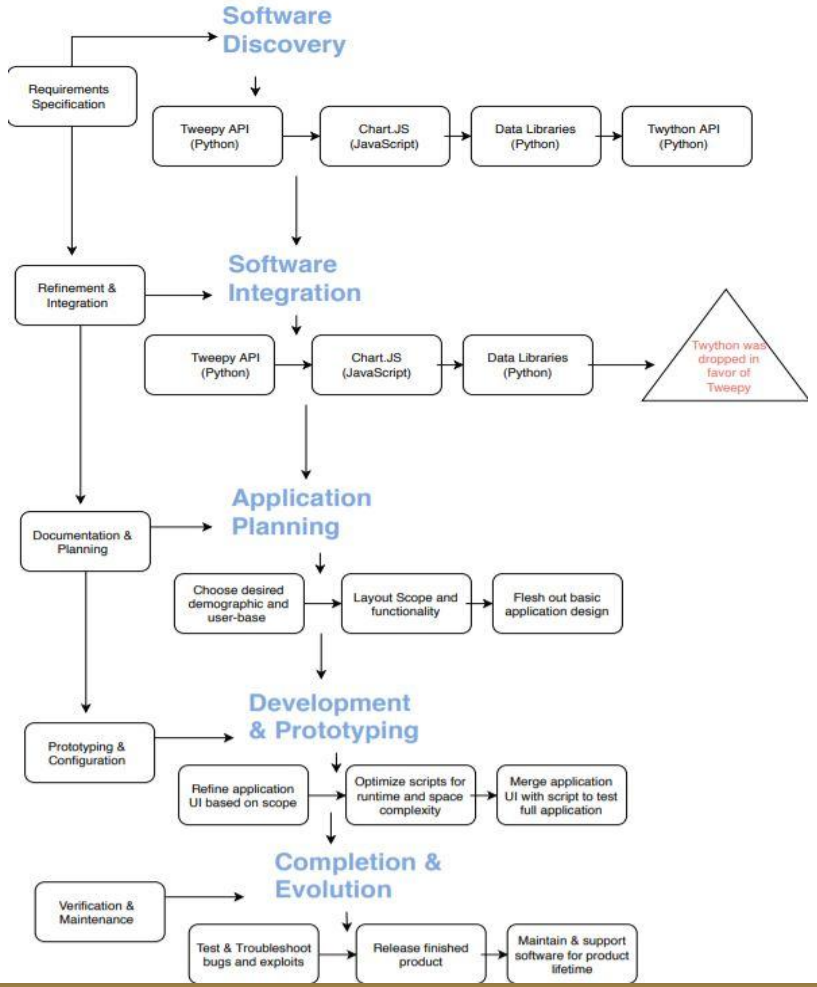
Focus Points

Before creating our application, we held a planning phase where we fleshed out important aspects that would be the backbone to our project scope. Some focus points we had for this project were:

- Easy, widespread accessibility
- Appealing and smooth UI design
- No handling of usernames, passwords, PII (personally identifiable information), or other account info
- Integrate pre-existing softwares and API’s to streamline software production phase

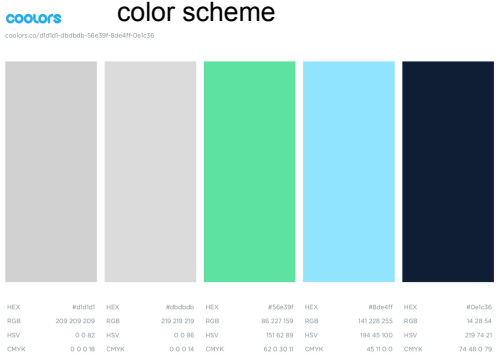
Documentation

Software Process

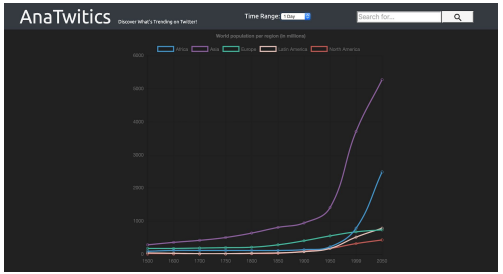

Software Discovery
Requirements Specification
Tweepy API (Python)
Chart.JS (JavaScript)
Data Libraries (Python)
Twython API (Python)
Refinement & Integration
Software Integration
Tweepy API (Python)
Chart.JS (JavaScript)
Data Libraries (Python)
Twython was dropped in favor of Tweepy
Documentation & Planning
Application Planning
Choose desired demographic and user-base
Layout Scope and functionality
Flesh out basic application design
Prototyping & Configuration
Development & Prototyping
Refine application UI based on scope
Optimize scripts for runtime and space complexity
Merge application UI with script to test full application
Verification & Maintenance
Completion & Evolution
Test & Troubleshoot bugs and exploits
Release finished product
Maintain & support software for product lifetime

Version Control - Web Application

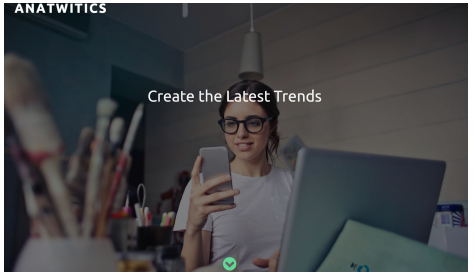
1. Drawing layout and picking color scheme



2. Basic prototype with limited functionality



3. Finished product (will still evolve with updates in future)

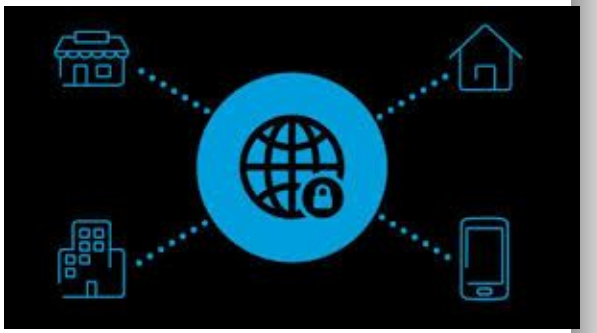


Version Control - Python Script

1. Basic visualization and data retrieval

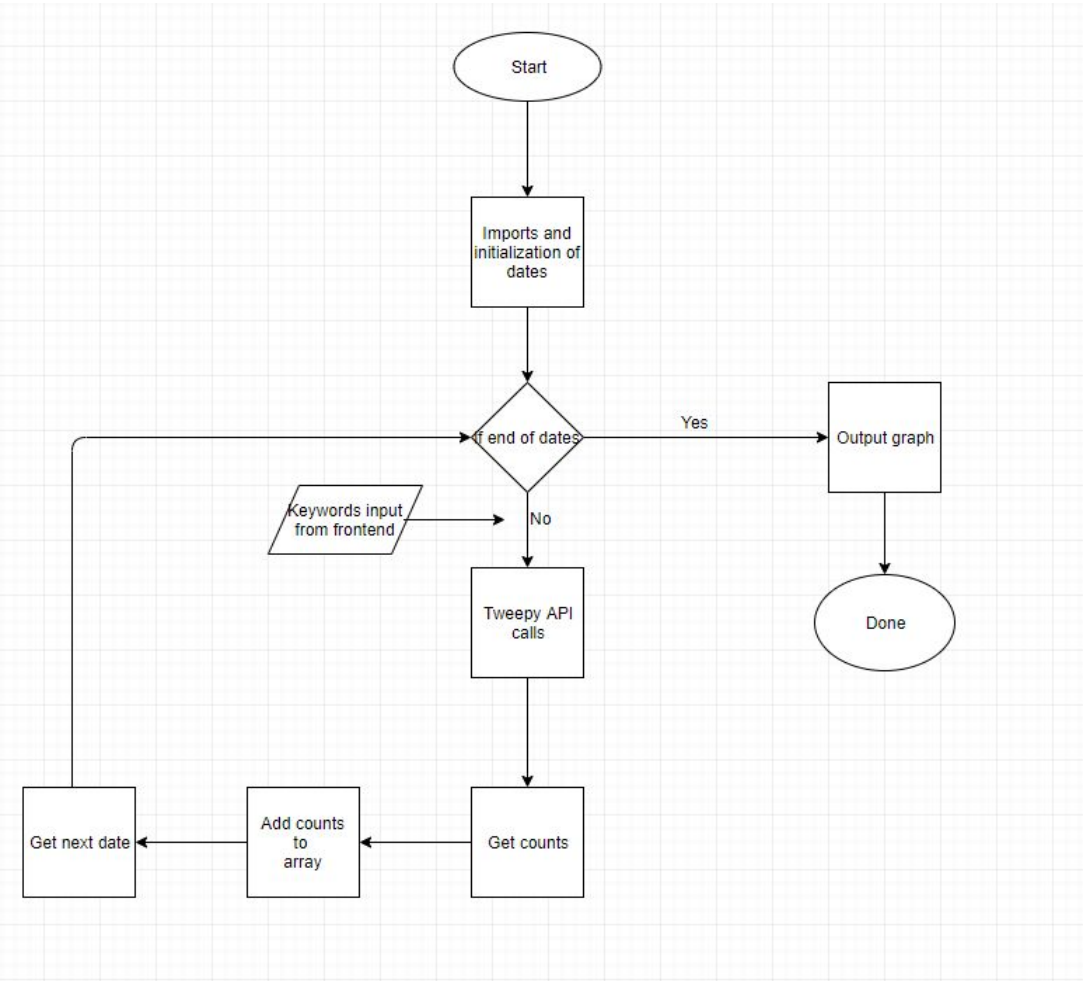
2. Integration of live data retrieval API and visualization into script with final API

3. Finished backend with API and data visualization and security features.



Demo - Python

Backend is not visually impressive, so here is a simple flowchart of how the latest version operates exclusive of the interface with front end!



Demo - Web Application

We will now present a brief demo of the web-app, demonstrating basic functionality.

M & E Report

Proposed Start Date	Actual Start Date	Activities
8/28	**8/28**	Project start date, select desired software functionality, assign roles in project
9/4	**8/29**	Draft project charter
9/11	**8/28**	Begin research on respective components of project. Finalize project charter.
9/13	**9/13**	Submit project charter for revision and approval.
9/14	**9/14**	Internal project status meeting
9/24	**9/26**	Internal project status meeting. Deliver first major version of project components.
9/25	**9/26**	External project status meeting
10/9	**10/12**	Internal project status meeting. Deliver second major version of project components.
10/10	**10/12**	External project status meeting.
10/23	**10/23**	Internal project status meeting. Deliver third major version of project components.
10/24	**10/24**	External project status meeting.
10/30	**11/2**	Being to merge front end and back end systems.
11/6	**11/10**	Troubleshoot any issues that directly impact functionality of merged system.
11/27	**11/29**	Test merged systems.
11/28	**11/28**	External project status meeting.
11/30	**12/3**	Close project.
12/4	**12/4**	Present project.

M & E Report

- 3 hours over budget accounts for an additional **$503.85** in labor.
- Still under budget.

Activity	Subactivities (with duration)	Total Duration	Estimated Duration	Difference
Python Script	Fetch data: 26 hours Address security: 8 hours Create graph: 4 hours	38	32	-6
Web Page	Design UI: 30 hours Address Security: 3 hours Display graph: 4 hours	37	42	5
Merge Systems	Verification: 16 hours	16	14	-2
				Total: -3

Course Learning Outcomes (CLO's)

CLO1	CLO2	CLO3	CLO4	CLO5
No collection of personal information Only data we handle and store is user-entered search phrases	Software process model Documentation pertaining to scope, schedule, cost, quality, communication, and stakeholder management Gantt chart	Software process model Prototyping	Project Charter and associated documentation	Team members worked collaboratively throughout the entirety of the project

Lessons Learned

- Standard Twitter API license issues
- Difficult to integrate front end and back end systems
- Balance between schoolwork and project
- Difficulty estimating a schedule and budget in advance