

Python API for visualisation of sets

Outline Document

School of Computer Science & Information Technology
University College Cork

Student Name	Nathan Crowley
Student Number	118429092
Supervisor Name	Rosane Minghim

November 2021

1) Analysis

The project involves creating a Python API/ library to be used by students or academic professionals with the aim of visualising sets, using InteractiVenn's approach to Set Visualisation, to allow a greater understanding of given sets, and their unions and intersections. The Python API developed should allow users to input up to six sets of data to be visualised as an interactive Venn Diagram with functionality that:

- 1) Allow users to visualise the union and intersections of the given complex input sets.
- 2) Allow users to hover over set unions and present the elements of a given set.

The developed Python API should take into account that the users may have little or no technical skills, this should direct the design towards being as easy as possible to use for the target audience. The documentation of the Python API must have clear and concise information on its functionality, along with concrete examples displaying to the user the operations that can be performed.

The project includes running and evaluating the developed Python API by itself as well as against an established API, *UpSet – Visualizing Intersecting Sets*, to highlight the advantages and disadvantages of each API given a number of examples.

2) Broad plan of steps

The project should commence with research into which technologies should be used and what API's are best for outputting interactive models to the user through Python. The design of the Python API should have consideration for users coming from any Operating System, with minimal importing of dependencies needed, allowing for users of any technical skill level to operate.

Technologies that need to be decided on:

- Which technology for displaying interactive outputs through Python.
- How the documentation of the Python API will be presented to the user.
- Which file types the user will be inputting and relevant error handling measures.

Once the technologies have been chosen, the implementation may begin. This will involve the development of the Python API using the chosen technologies from the initial definition phase of the project. The implementation involves:

- 1) Creating a directory to store the Python library.
- 2) Creating a virtual environment for the directory.
- 3) Creating the content of the Python library.
- 4) Building the Python library.

Development of the documentation, to allow the users to understand the functionality of the API, may commence. Once the implementation and documentation have been completed to a satisfactory level, the final stage of the project may begin.

The final stage of the project will involve testing the Python API extensively, to discard unforeseen errors or missing functionality. The Python API will be evaluated by itself to assure it satisfies the requirements of the project. The Python API will also be evaluated against the existing *UpSet Visualisation* API to compare the two projects. The documentation will be finalised and completed. The project closure will involve public and peer testing of the developed Python API by the target audience.