

Design Document 2

Requirements, Constraints, and Standards

2.1. REQUIREMENTS & CONSTRAINTS

This section describes the requirements and constraints that our team has identified and categorized as main prerequisites for successful implementation of our project. The requirements and constraints listed will be used as guidelines throughout the development process, ensuring that we meet the necessary criteria to provide a meaningful solution to our users.

Functional Requirements

Requirements that fall under functional requirements pertain to the application's capabilities.

- Application shall allow for users to create and login to a personal account.
- The user shall be able to load datasets from a variety of supported formats from their device to a database.
- The user shall be able to select an active dataset from the datasets they've uploaded to the database.
- The user shall be able to select from supported whereabouts algorithms based on their current user type.
- Based on the currently selected whereabouts algorithm the user should be able to create a query from a selection of query types.
- Query creation should allow the user to select subsets of data from the active dataset if desired.
- The interface should allow for query creation/configuration through interaction with the visualization space/window.
- The application interface should attempt to constrain queries that would otherwise be uncomputable within the listed constraint time as seen in the constraint section below.
- Visualization will take several forms depending on the current user type and whereabouts algorithm (i.e. map overlay, 2D, and 3D white-space backgrounds).
-

Experiential Requirements

Requirements that fall under experiential requirements pertain to increasing application usability.

- User's preferences should be stored and recalled to create a seamless experience.
- Based on the current visualization tool, inclusion of a set of navigation features (Pan, Zoom, Rotate, ect.)
- User notifications and updates to inform the user of current application state or status.

Aesthetic Requirements

Requirements that fall under aesthetic requirements pertain to the application's visual appearance.

- Application's user interface will take on a utilitarian aesthetic.
- The interface shall be designed in such a way that promotes intuitive interaction and navigability.
- The visualization tools provided by the application to the user should cater to the current user type.
- Visualization tool - color customization for points and probabilities - where applicable.

Non-Functional Requirements

Requirements that fall under non-functional requirements pertain to application's construction.

- The application should take the form of a web application that can be accessed from a domain outside the hosts domain.
- Utilize exclusively well-tested open-source libraries.
- Develop with modular coding practices in mind.
- Maintain adequate code documentation.

Constraints:

The constraints outlined determine the level to which functionality is supported.

- Time from query request to visualization rendered should take no longer than thirty-seconds.
- Input datasets must adhere to a supported format.

2.2. ENGINEERING STANDARDS

IEEE Standards

Standard	Rationale
23026-2023 - ISO/IEC/IEEE: Engineering and Management of Websites for Systems, Software, and Services Information	These will be the standards we follow for our development of the web application.
15288-2023 - ISO/IEC/IEEE International Standard - Systems and software engineering--System life cycle processes	This standard will help us to follow a correct lifecycle of software from conception to retirement
1012-2016 - IEEE Standard for System, Software, and Hardware Verification and Validation	This standard we will use to help with requirements to best create the software and risks to make sure they are mitigated correctly. Along with testing and other areas.

Other Standards/Conventions

Standard	Rationale
Oracle Java Coding Conventions	We will be using a Spring Boot mysql server for the backend. Following these conventions will keep all code similar and easy to read.
Vue.js Style Guide	Using the Style guide from Vue.js will keep all code easy to read. This will also keep certain issues to a minimum following the same style.
Scrum Style	We need a style to develop our project. We will have Scrum Sprints that last 2-3 weeks.

2.3. REFERENCES

- “Vue.js.” *Introduction* | *Vue.js*, vuejs.org/guide/introduction.html. Accessed 19 Mar. 2024.
- “Code Conventions for the Java Programming Language.” *Code Conventions for the Java Programming Language: 1. Introduction*, www.oracle.com/java/technologies/javase/codeconventions-introduction.html. Accessed 19 Mar. 2024.
- "ISO/IEC/IEEE International Standard - Systems and Software Engineering -- Engineering and Management of Websites for Systems, Software, and Services Information," in ISO/IEC/IEEE 23026:2023(E) , vol., no., pp.1-70, 18 July 2023, doi: 10.1109/IEEESTD.2023.10186263. keywords: {IEEE Standards;IEC Standards;ISO Standards;Software engineering;Systems engineering and theory;Engineering management;Web sites},
- "ISO/IEC/IEEE International Standard - Systems and software engineering--System life cycle processes," in ISO/IEC/IEEE 15288:2023(E) , vol., no., pp.1-128, 16 May 2023, doi: 10.1109/IEEESTD.2023.10123367. keywords: {IEEE Standards;ISO Standards;IEC Standards;Software engineering;Systems engineering and theory;Product lifecycle management},
- "IEEE Standard for System, Software, and Hardware Verification and Validation," in IEEE Std 1012-2016 (Revision of IEEE Std 1012-2012/ Incorporates IEEE Std 1012-2016/Cor1-2017) , vol., no., pp.1-260, 29 Sept. 2017, doi: 10.1109/IEEESTD.2017.8055462. keywords: {IEEE Standards;Software testing;Performance evaluation;Product life cycle management;Environmental factors;Hazards;acceptance testing;architecture evaluation;component testing;concept documentation evaluation;criticality;criticality analysis;design evaluation;disposal plan evaluation;environmental verification and validation (V&V) factors;hardware life cycle;hardware V&V;hardware verification and validation;hazard analysis;IEEE 1012;implementation evaluation;independent verification and validation (IV&V);integration testing;integrity level;interface analysis;IV&V;minimum V&V tasks;nth of a kind;objective evidence;operating procedure evaluation;qualification testing;quality assurance;regression analysis;regression testing;requirements allocation analysis;requirements evaluation;reuse software;risk analysis;security analysis;software life cycle;software quality assurance (SQA);software V&V;software verification and validation;source code documentation evaluation;source code evaluation;SQA;stakeholder needs and requirements evaluation;system element interaction analysis;system life cycle;system maintenance strategy assessment;system of interest;system requirements evaluation;system V&V;system verification and validation;testing;traceability analysis;V&V;V&V measures;validation;verification},

