**Software Requirements**

**Specification**

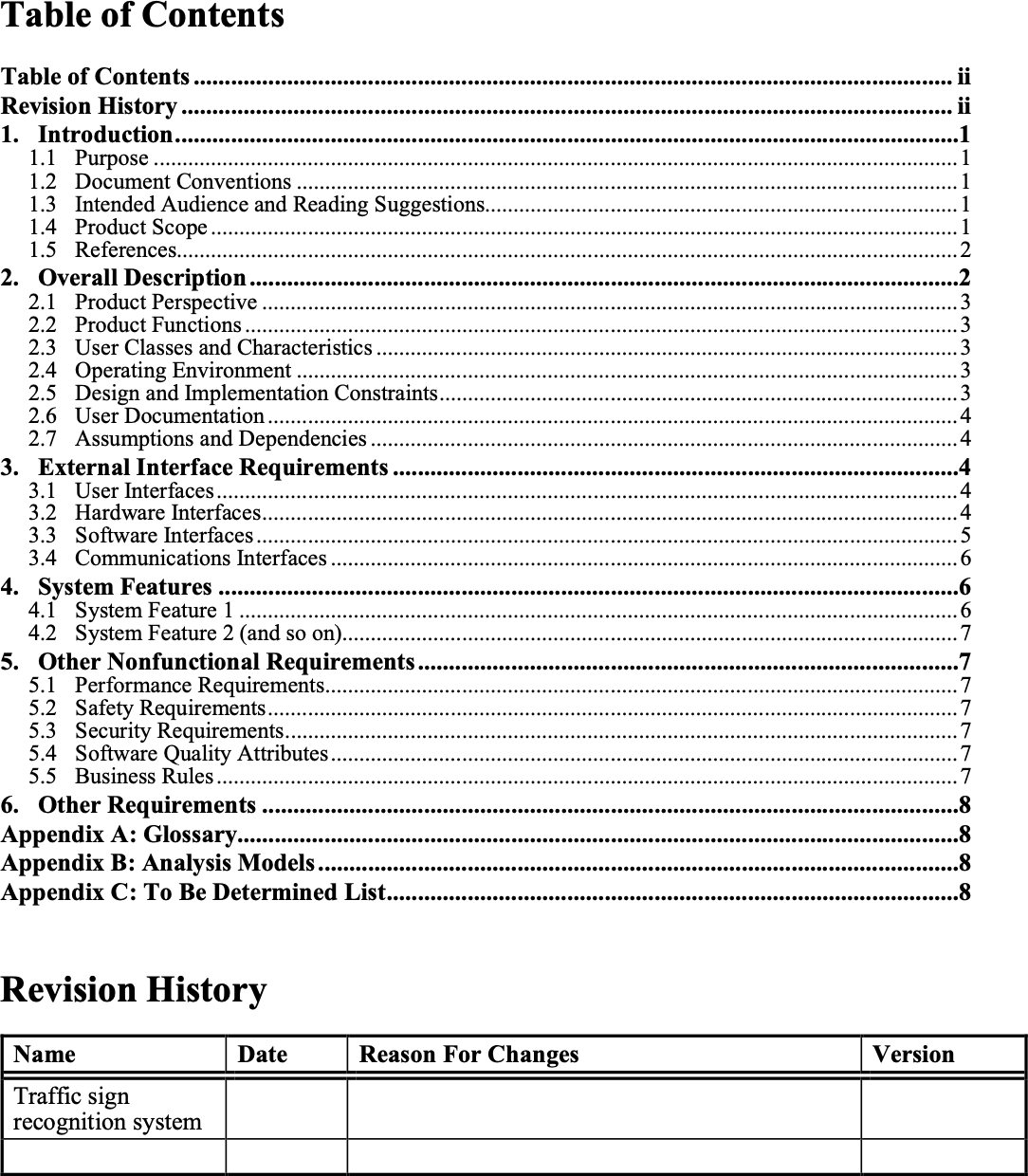
**for**

**Codeforces Analyser**

## Prepared by Sagar Srivastava, Priyanshi, Priyanshu Raj

**KIET group of Institutions, Ghaziabad**

## 21st April 2023



# Introduction

## Purpose

Codeforces Analyser is a tool designed to visualise the execution of code on test cases for problems hosted on the Codeforces platform. It allows users to better understand how their code is working and to identify potential errors or inefficiencies in their algorithm. Using Codeforces Analyser users can input their code and test cases for a Codeforces problem and see a visual representation of how their code performs on each test case. The tool displays the input and output for each test case, as well as the variables and data structures used by the code at each step of the execution.

## Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

## Intended Audience and Reading Suggestions

Typical users such as students who want to learn more about traffic sign recognition system and machine learning algorithms used in building this project.

## Product Scope

It is a system which is capable of recognising the traffic signals. This system can be installed in any car which is having ADAS (advanced driver assistant system) with a external camera mounted on it. It will then monitor the traffic signs and performs the desired task

## References

1. *A. Nikonorov, P. Yakimov, M. Petrov, Traffic sign detection on GPU using color shaperegular expressions, VISIGRAPP IMTA-4, Paper 8 (2013).*
2. *R. Belaroussi, P. Foucher, J.P. Tarel, B. Soheilian, P. Charbonnier, N. Paparoditis, Road Sign Detection in Images, A Case Study, 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 484-488*
3. *Aditya, A.M., & Moharir, S. (2016). Study of Traffic Sign Detection and RecognitionAlgorithms.*
4. *Mogelmose, M. M. Trivedi and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey", IEEE Trans. Intell. Transp. Syst., vol. 13, pp. 1484-1497, Dec. 2012.*
5. *P. Viola and M. J. Jones, "Robust real-time face detection", Int. J. Comput. Vis., vol. 57, no. 2, pp. 137-154, 2004.*
6. *L. Fletcher, N. Apostoloff, L. Petersson and A. Zelinsky, "Vision in and out of vehicles", IEEE Intell. Syst., vol. 18, no. 3, pp. 12-17, May/Jun. 2003*
7. *S. Lafuente-Arroyo, P. Gil-Jimnez, R. Maldonado-Bascn, F. Lpez-Ferreras and S. Maldonado- Bascn, "Traffic sign shape classification evaluation I: SVM using distance to borders", Proc. IEEE Intell. Veh. Symp., pp. 557-562, 2005-Jun.*
8. *V. Balali, A. A. Rad and M. Golparvar-Fard, "Detection classification and mapping of U.S. traffic signs using Google street view images for roadway inventory management", Vis. Eng., vol. 3, no. 1, pp. 15, 2015.*
9. *K. C. P. Wang, Z. Hou and W. Gong, "Automated road sign inventory system based on stereo vision and tracking", Computer.-Aided Civil Infrastructure. Eng., vol. 25, no. 6, pp. 468-477, 2010.*

# Overall Description

## Product Perspective

Codeforces Analyser recognition system is a system in which we have used machine learning and to be specific we have used CNN algorithm in it. Basically it is a system which is capable of recognizing the traffic signals. This system can be installed in any car which is having ADAS (advanced driver assistant system) with a external camera mounted on it. It will then monitor the traffic signals and performs the desired task.

## Product Functions

This system can be installed in any car which is having ADAS (advanced driver assistant system) with a external camera mounted on it. It will then monitor the traffic signals and performed the desired task.

Let’s see a scenario in which a car is running with a speed of 120km/hr. and if our system detects the sign board which says that the speed limit on this road is 100km/hr. system will automatically reduced to its speed limit as the car is laced with ADAS and all the things of car can be controlled electronically.

## User Classes and Characteristics

Typical users such as students who want to learn more about traffic sign recognition system and machine learning algorithms used in building this project.

## Operating Environment

* *Windows 11*
* *Windows 7*
* *Windows 8*
* *Windows 10*
* *Mac OS X*
* *Linux*

## Design and Implementation Constraints

The design is combination of hardware and software. That means failure of any module can leads to failure of whole project. The design is simple, external camera will provide the input to the algorithm and after that algorithm will process it .

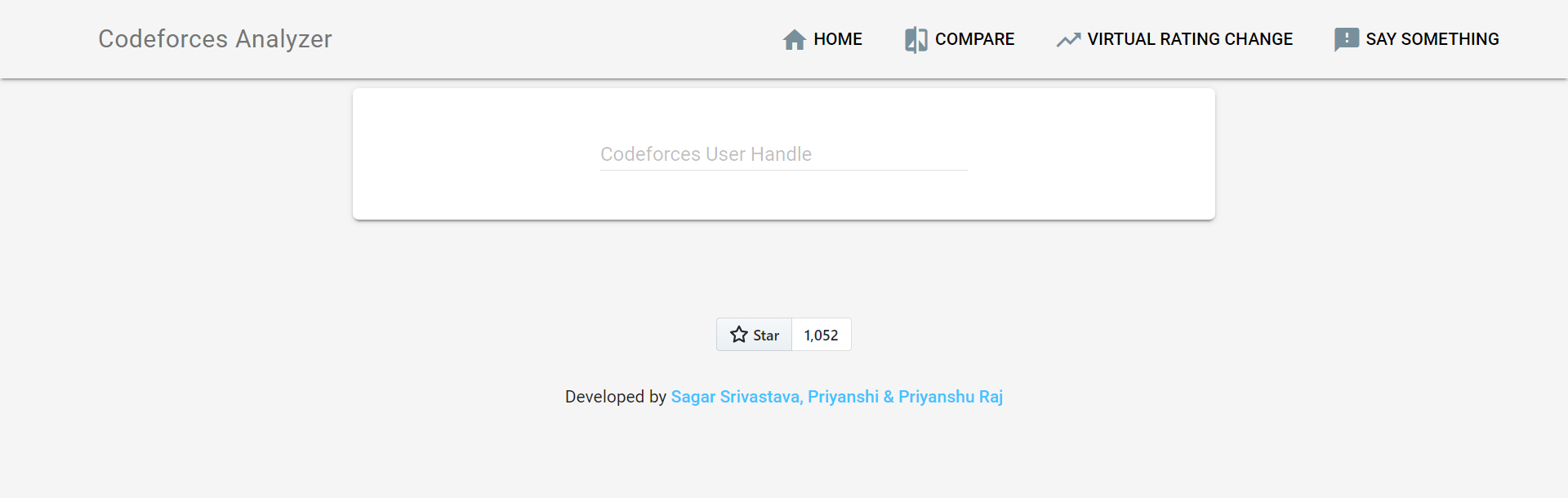
Anything which takes input will limit the function of the system. Hardware like camera, wires , couplers and mount should have to perform better and every time in order to continuously provide the input.

Algorithm failure and slow system can lead to failure of project. As the algorithm have to respond as soon as it gets the input.

## User Documentation

There is a quick guide available on the link :- https://cfviz.netlify.app

Below is some image to understand the user manual easily.



## Assumptions and Dependencies

The assumptions for this project is that roads should have proper traffic signs and camera should be placed at the proper angle to capture all the traffic signs and CPU of car should be fast enough to process the data.

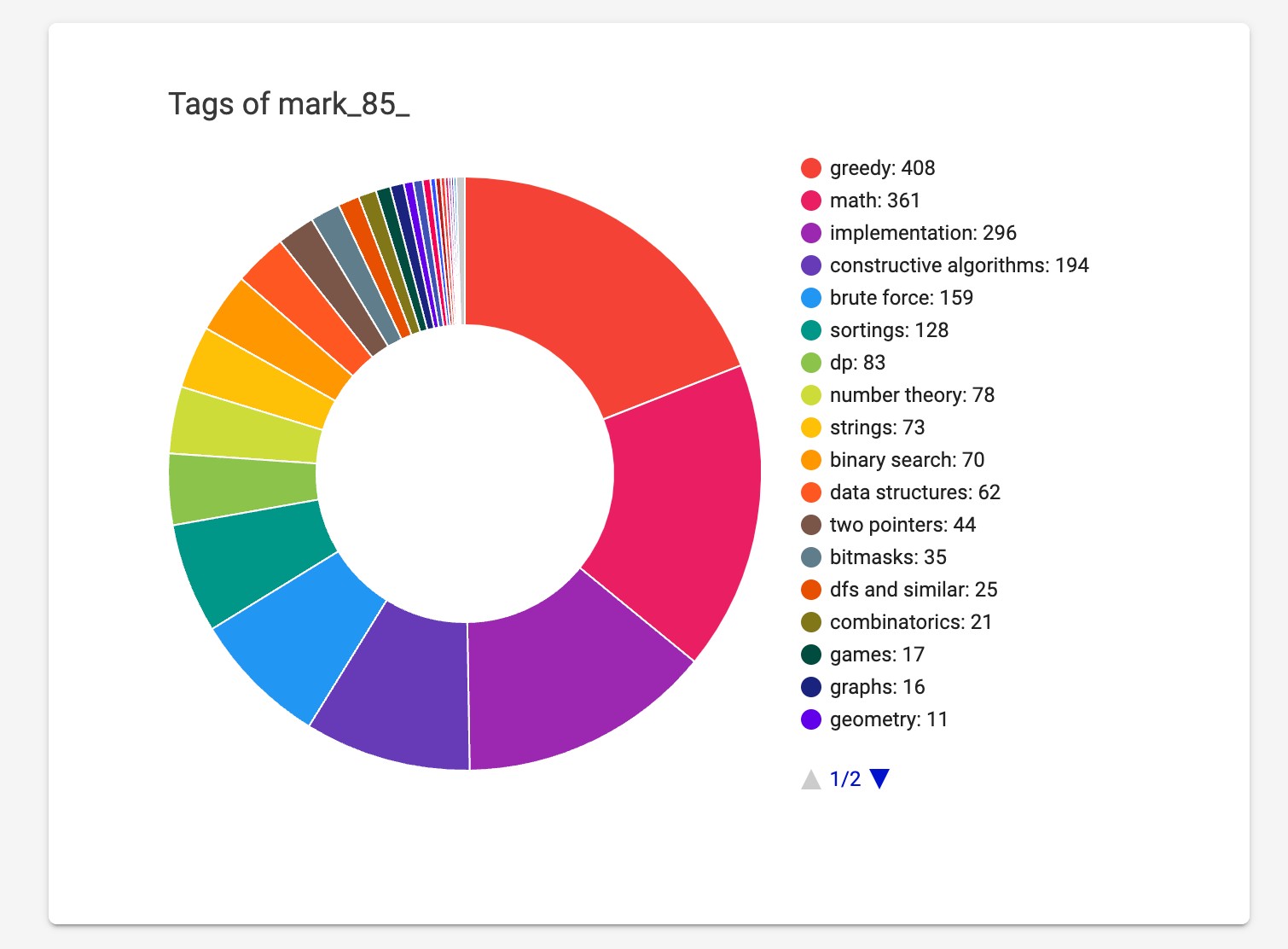
# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

The first things is the coupler to connect the camera to the CPU of the car.



Second thing we required is a front camera in a car.

Third thing required is the infotainment system or CPU in a car.

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

### Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

### Stimulus/Response Sequences

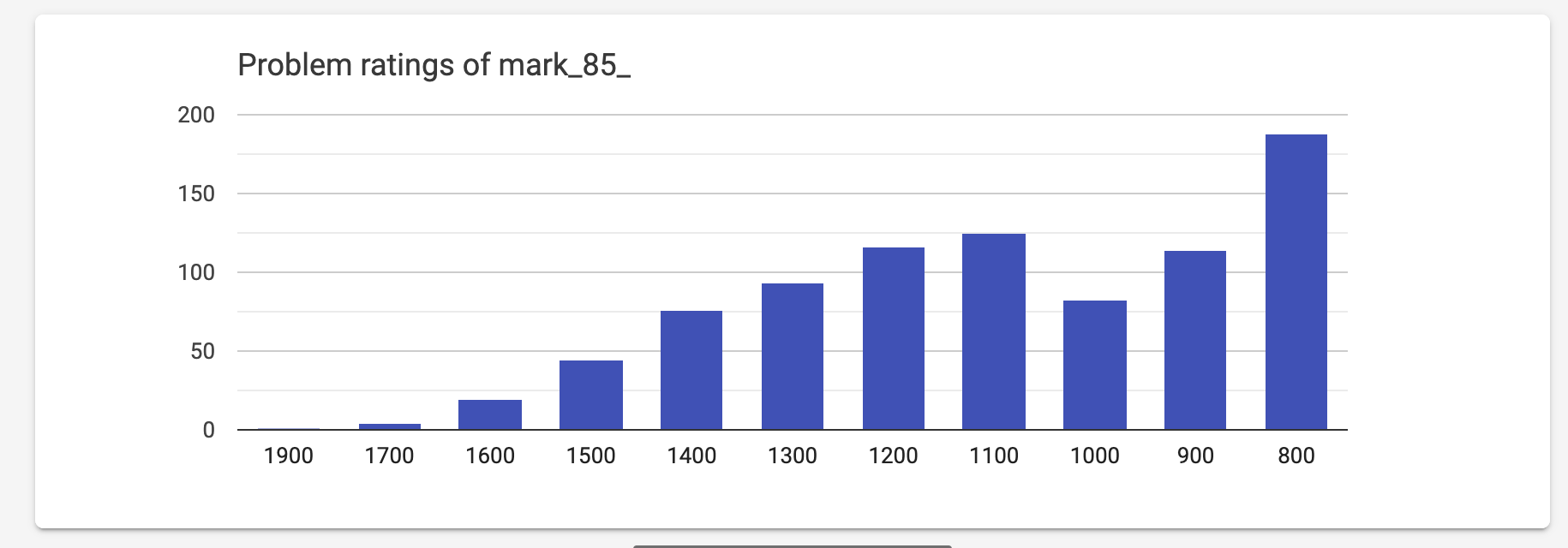
<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

### Functional Requirements

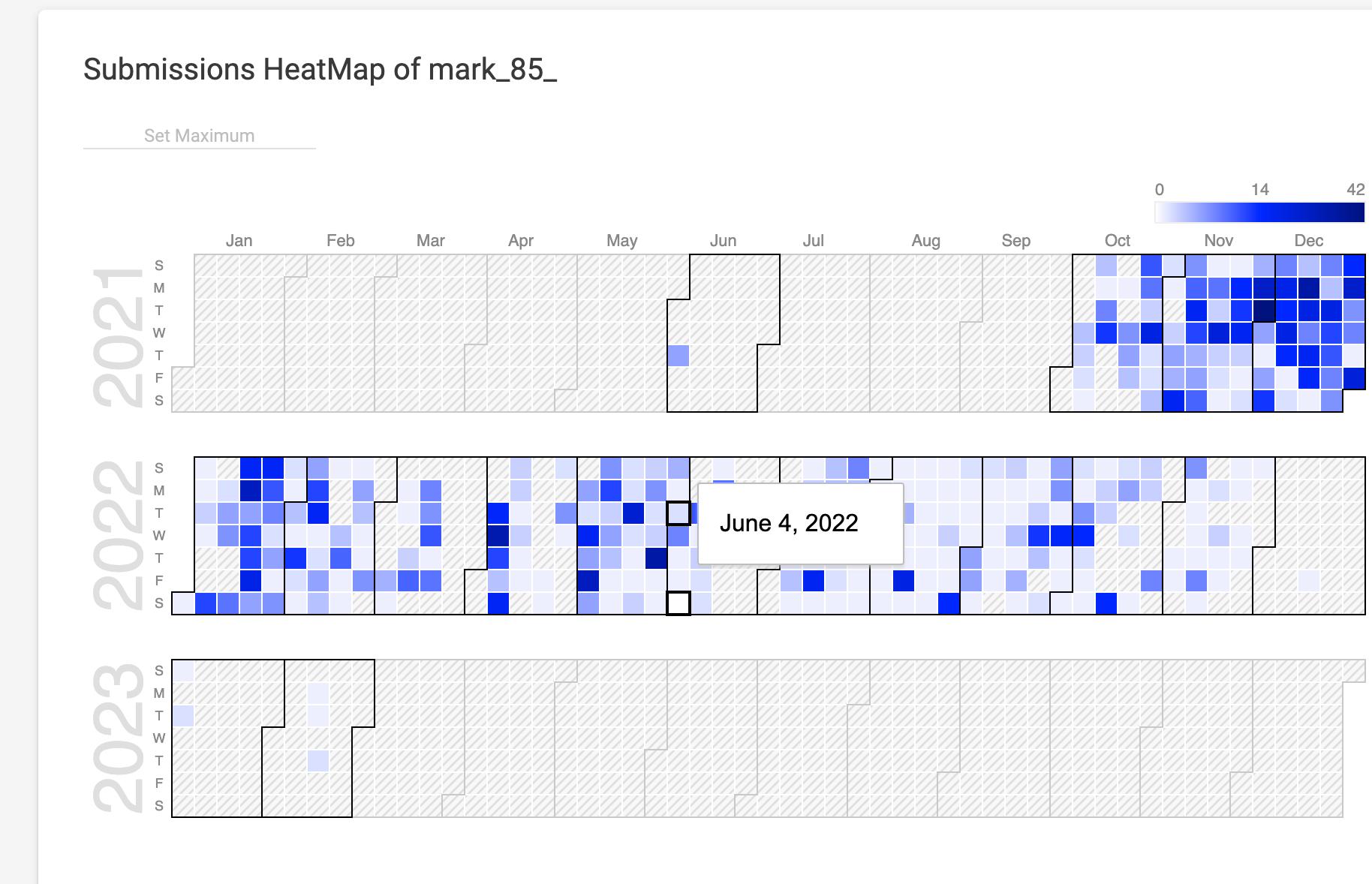
<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

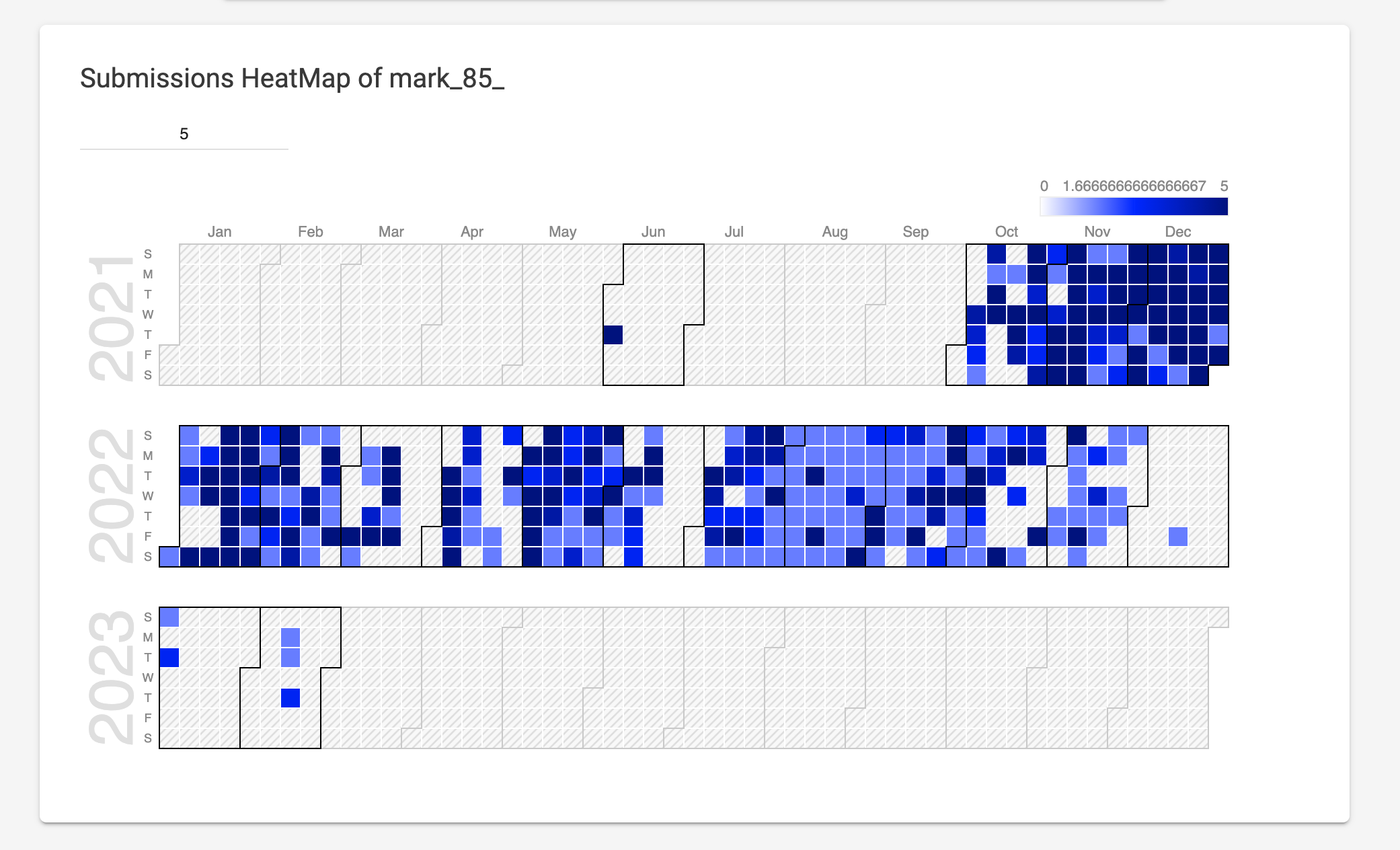
<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

### REQ-1:



REQ-2:





* 1. **System Feature 2 (and so on)**

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

# Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

# Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

# Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>