SOFTWARE REQUIREMENTS SPECIFICATION

for

<Project>

Version 1.0 approved

Prepared by <author>

<Organization>

November 18, 2019

Contents

1 Introduction **5**
- 1.1 Purpose . . . 5
- 1.2 Document Conventions . . . 5
- 1.3 Intended Audience and Reading Suggestions . . . 5
- 1.4 Project Scope . . . 5
- 1.5 References . . . 6

2 Overall Description **7**
- 2.1 Product Perspective . . . 7
- 2.2 Product Functions . . . 7
- 2.3 User Classes and Characteristics . . . 7
- 2.4 Operating Environment . . . 7
- 2.5 Design and Implementation Constraints . . . 7
- 2.6 User Documentation . . . 8
- 2.7 Assumptions and Dependencies . . . 8

3 External Interface Requirements **9**
- 3.1 User Interfaces . . . 9
- 3.2 Hardware Interfaces . . . 9
- 3.3 Software Interfaces . . . 9
- 3.4 Communications Interfaces . . . 9

4 System Features **10**
- 4.1 System Feature 1 . . . 10
  - 4.1.1 Description and Priority . . . 10
  - 4.1.2 Stimulus/Response Sequences . . . 10
  - 4.1.3 Functional Requirements . . . 10
- 4.2 System Feature 2 (and so on) . . . 10

5 Other Nonfunctional Requirements **11**
- 5.1 Performance Requirements . . . 11
- 5.2 Safety Requirements . . . 11
- 5.3 Security Requirements . . . 11
- 5.4 Software Quality Attributes . . . 11
- 5.5 Business Rules . . . 12

6 Other Requirements **13**
- 6.1 Appendix A: Glossary . . . 13

6.2 Appendix B: Analysis Models 13
6.3 Appendix C: To Be Determined List 13

Revision History

Name	Date	Reason For Changes	Version
21	22	23	24
31	32	33	34

1 Introduction

1.1 Purpose

TyM is a software designed for generating and testing various Machine Learning models. It gives you a Machine Learning Model as an output which you can instantly use for prediction. This is the first version of this software, hence this document would specify the requirements of TyM 1.0. All the functionalities that have been implemented are covered in this SRS document.

1.2 Document Conventions

The font used while writing this document is Arial and the size is 12 points. For the functional requirements, the default priority is medium if not specified. No inheritance of any kind takes place among the high level requirements and the detailed requirements.

1.3 Intended Audience and Reading Suggestions

This document is primarily intended for developers and project managers who want to use Machine Learning models in their projects but don't have the time to implement them. This document is also for users who want to have a ready made Machine Learning model. The Organization of this document is as follows: Chapter 1 Introduction gives the introduction to this document; Chapter 2 Overall Description gives the high level explanation of the software, its functions and characteristics; Chapter 3 External Interface Requirements will give the information about various interfaces; Chapter 4 System Features will give detailed explanation of the features of the software; Chapter 5 Other Nonfunctional Requirements will give the overview of the Non functional requirements.

1.4 Project Scope

TyM software will enable its users to generate Machine Learning Models without implementing them. With the help of the GUI provided by our software, the user will be able to select the ML model he wants and give the data as input on which he wants to train the ML model on. Our software will give suggestions on whether the data overfit or underfit, will show graphical representations and also give the model which the user can download. TyM is for those users who do not have the time for implementing ML models but want to use one. TyM is also for developers who want to see how a ML model fits on the data without implementing them.

1.5 References

TODO <List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2 Overall Description

2.1 Product Perspective

The software TyM is self contained and has no relation to any existing software whatsoever. This is the first version of the software and hence has been developed from scratch.

2.2 Product Functions

TODO <Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 User Classes and Characteristics

Tym software will be used by two types of users: premium users and normal users. Premium users will have more ML models to choose from and will get more insight. Normal users will have the option to upgrade to premium. Normal users can try out the software and, if they want to, can become premium users.

2.4 Operating Environment

Since our software will be deployed over the web, there are no constraints on hardware and the OS. However, TyM software could be only run on Chrome (version 60 or higher) and Mozilla Firefox (version 56 or higher).

2.5 Design and Implementation Constraints

Tym Software will work only on Chrome (version 60 or higher) and Mozilla Firefox (version 56 or higher). This software was built using HTML5 and Bootstrap for Frontend and flask for backend. ML libraries like sklearn and tensorflow have been used for implementing Machine Learning models. SQL database has been used for storing the user data and their models. All the Backend part has been implemented in Python language.

2.6 User Documentation

A user manual will be provided with the TyM software. It would contain detailed descriptions of the functionalities of the software, how the user would interact with the software and what additional things are available to the premium users.

2.7 Assumptions and Dependencies

Tym software would be dependent on Machine Learning libraries sklearn and tensorflow provided by Python as we will be using these libraries to implement the ML models. Also, it would be dependent on Flask library for backend purposes.

3 External Interface Requirements

3.1 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.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 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.>

3.4 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.>

4 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.>

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 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).>

4.1.2 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.>

4.1.3 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:

4.2 System Feature 2 (and so on)

5 Other Nonfunctional Requirements

5.1 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.>

5.2 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.>

5.3 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.>

5.4 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.>

5.5 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.>

6 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.>

6.1 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.>

6.2 Appendix B: Analysis Models

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

6.3 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.>