Software Requirements Specification

for

Teaching Tasks

**Version 1.0 approved**

**Prepared by Chad Berry, Benjamin Caras, Jeremy Hutchinson, Abdullah Karim, Ethan Levy, Mark Sullivan, James West**

**Blackbear Consultants**

**02/15/20**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

Children with Autism Spectrum Disorder generally learn in non-conventional ways compared to children in a normal school setting. Because of this, new techniques and methodologies about how to effectively teach children with Autism Spectrum Disorder have been created. One such technique is Discrete Trial instruction. The purpose of our mobile application is to capture the technique of Discrete Trial Instruction and make it into a fun and intuitive game that children can use to learn material set forth by the Maine Early Childhood Learning Guidelines. Another purpose of the application, Teaching Tasks, is to help parents cut the cost of educating their children, as instructors and clinicians in the field charge high fees to help educate children with Autism Spectrum Disorder. This is the first release of the third revision of the SRS document, and its aim is to describe the application as a whole.

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

## Document Conventions

We have followed the standard conventions for the SRS using Times New Roman as the font with a size 14 for headings and size 11 for standard body text. For this document there has been a text color change to display the priority levels of tasks. Red reflects a high level task, Yellow for a medium level task, and Green for a low level task. In regards to determining our priority levels each task will have its own priority level assigned.

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## Intended Audience and Reading Suggestions

This document is intended for developers, project managers, uers, testers, and documentation writers. For the developers, this document will describe the expected functional and non-functional requirements for this project. For project managers this document will detail the project overview along with scope and goal for the project. This document will assist users and testers by listing different features of the project for better understanding what the application is aimed for. In the case of documentation writers this document will serve to assist with understanding the application and scope of the project.

The rest of this document will describe the scope, references, description, requirements, functions, and features of the application.

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

## Product Scope

The software that Teaching Tasks is creating is an educational game for children with Autism Spectrum Disorder, this educational game is being created so that professionals that were trained to teach these kids can have another tool to help teach these children. Children with Autism Spectrum Disorder need one on one help, someone needs to be there making sure they know that they are learning and doing it correctly. So the goal of this application is to be able to aid these teachers, so that the kids can play this game and learn what they need to learn at the same time. We want these teachers to be able to give the game to the kid and they will be able to start learning, this could also be very useful for parents they could give this to their kid at home and they could continue to learn at home.

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>*

## References

<https://www.apa.org/>

This link contains the standards and guidelines that are

Boi

*<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.>*

# Overall Description

## Product Perspective

This project is a new self-contained product. It comes with all new systems and subsystems designed to aid in the functionality of the application.

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

## Product Functions

* User Login
* Tasks that following DTI guidelines
* Statistics based on user performance

*<Summarize the major functions the product must perform, just a bullet list is needed here. Details will be provided in Section 3.* ***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.>*

## User Classes and Characteristics

User Class: General Users - Adults

Adults using the application will have access to the whole application, but will not be permitted to play games. Gameplay is strictly for children as it is used to educate them and track their progress. Adults will be assumed to have access to the same technology that the children do, with little expertise on how the functionality is but a good understanding of the statistics tracking the children. Generally, adults shouldn’t be using the product too much, as the main reason for the application to be open is for the children to play the game.

User Class: General Users - Children

Children are the primary users of the application. With little to no technical expertise, they will be accessing the productly solely to play the game and educate themselves on material from their grade category. They will also have the second most experience using the application over time as they play the game and are accustomed to the user interface.

User Class: Power Users

These users are the users with a vast amount of resources and access to development tools. Their primary use is establishing gameplay and functionality for the other users. They will initially be the primary users of the application and will over time use the application less and less as it becomes developed. There will be some cyclicality in the frequency of use as new features are brought up and implemented. They will have the highest security clearance on the application and the most experience using it.

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

## Operating Environment

Boi

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

## Design and Implementation Constraints

Boi

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## User Documentation

N/A

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

## Assumptions and Dependencies

Boi

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# External Interface Requirements

## User Interfaces

Boi

*<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.>*

Initially after opening the app, Teaching Tasks will allow the user to select from a list of accounts to login. The user selects an account by tapping on it, which prompts the user with a password requirement. Once logged into the app, the user is brought to the Task interface. At the bottom of the screen are the buttons for the Task interface, Statistics interface, and Settings interface in that order.



On the Tasks interface, the user will be brought to the last task they had been working on. At the top is the account user’s name, followed by the category the current task is in. For the image above, the current category is Matching and the current Task is matching the Square. The user can see the current progress this account has made on the task at hand which is being represented by the stars. A reference to the task is shown using an icon, in this example, it is the red square. The user will click the START button, at which point the task will begin. While in *Task Mode* the user will be required to enter the account password when trying to leave the Task interface. During the Task Mode, the user will be given tasks to complete with new tasks being generated after each completion of a task.



## 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.>*

## 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.>*

## Account Registration

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

4.1.1 Description and Priority

Priority: High

Allows a user to register a new account within the system, including a name, password, and optional information such as color blindness to enhance their experience.

*<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

The user taps a register button on the login page, which navigates to a registration page with text inputs for name and password, and checkboxes for color blindness and other information. After the “Create Account” button is tapped, the information is saved in a database and the account name is added to the account list in the login page. The user is returned to the login page.

*<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: The system shall perform a secure hash function on the given password before it is saved 100% of the time.

REQ-2: The system shall communicate with a local database to save account information within 3 seconds 95% of the time.

## Account Login

4.2.1 Description and Priority

Priority: High

Allows a user to access a registered account and use the application with that account.

*<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).>*

4.2.2 Stimulus/Response Sequences

The user taps a registered account name on the login screen. A popup asks for the account password. After tapping the login button, if the password is incorrect the user is told and given 3 more chances to login before the account is locked for half an hour. If the password is correct the user is shown the game selection page.

*<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.>*

4.2.3 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: The system shall communicate with a local database to compare the hash of a given password to the stored hash within 3 seconds 95% of the time.

REQ-2:

## Game Selection

4.3.1 Description and Priority

Priority: High

Allows a user to choose a game type and lock the application to the game screen.

*<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).>*

4.3.2 Stimulus/Response Sequences

The user chooses a game type, or simply taps play to proceed with their current track. The system moves to the game screen and if the option is enabled pops up a message about locking the app.

*<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.>*

4.3.3 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: The system shall load and display the game selection screen within 3 seconds of an account being logged in 95% of the time.

REQ-2: The system shall load the game screen after selection within 2 seconds 90% of the time.

## Shape Identification

4.4.1 Description and Priority

Priority: High

A game in which the user taps the shape they are prompted to, with trials following the Discrete Trial Instruction(DTI) model.

*<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).>*

4.4.2 Stimulus/Response Sequences

The user is shown one or more shapes and told to tap one of them. If the user succeeds the system responds with a congratulatory screen paired with sound. If the user does not succeed, the system will follow DTI instructions, which may result in replaying the same game or a similar game.

*<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.>*

4.4.3 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: The system shall load and display the game within 3 seconds 95% of the time.

REQ-2:

## Letter Identification

4.5.1 Description and Priority

Priority: Medium

A game in which the user taps the letter they are prompted to, with trials following the Discrete Trial Instruction(DTI) model.

*<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).>*

4.5.2 Stimulus/Response Sequences

The user is shown one or more letters and told to tap one of them. If the user succeeds the system responds with a congratulatory screen paired with sound. If the user does not succeed, the system will follow DTI instructions, which may result in replaying the same game or a similar game.

*<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.>*

4.5.3 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: The system shall load and display the game within 3 seconds 95% of the time.

REQ-2:

## Number Identification

4.6.1 Description and Priority

Priority: Medium

A game in which the user taps the number they are prompted to, with trials following the Discrete Trial Instruction(DTI) model.

*<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).>*

4.6.2 Stimulus/Response Sequences

The user is shown one or more numbers and told to tap one of them. If the user succeeds the system responds with a congratulatory screen paired with sound. If the user does not succeed, the system will follow DTI instructions, which may result in replaying the same game or a similar game.

*<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.>*

4.6.3 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: The system shall load and display the game within 3 seconds 95% of the time.

REQ-2:

## Shape Matching

4.7.1 Description and Priority

Priority: Low

A game in which the user taps the shape they are shown, with trials following the Discrete Trial Instruction(DTI) model.

*<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).>*

4.7.2 Stimulus/Response Sequences

The user is shown a shape, paired with one or more shapes. They are then told to tap the matching shape from the second set. If the user succeeds the system responds with a congratulatory screen paired with sound. If the user does not succeed, the system will follow DTI instructions, which may result in replaying the same game or a similar game.

*<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.>*

4.7.3 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: The system shall load and display the game within 3 seconds 95% of the time.

REQ-2:

## Letter Matching

4.8.1 Description and Priority

Priority: Low

A game in which the user taps the letter they are shown, with trials following the Discrete Trial Instruction(DTI) model.

*<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).>*

4.8.2 Stimulus/Response Sequences

The user is shown a letter, paired with one or more letters. They are then told to tap the matching letter from the second set. If the user succeeds the system responds with a congratulatory screen paired with sound. If the user does not succeed, the system will follow DTI instructions, which may result in replaying the same game or a similar game.

*<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.>*

4.8.3 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: The system shall load and display the game within 3 seconds 95% of the time.

REQ-2:

## Number Matching

4.9.1 Description and Priority

Priority: Low

A game in which the user taps the number they are shown, with trials following the Discrete Trial Instruction(DTI) model.

*<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).>*

4.9.2 Stimulus/Response Sequences

The user is shown a number, paired with one or more numbers. They are then told to tap the matching number from the second set. If the user succeeds the system responds with a congratulatory screen paired with sound. If the user does not succeed, the system will follow DTI instructions, which may result in replaying the same game or a similar game.

*<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.>*

4.9.3 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: The system shall load and display the game within 3 seconds 95% of the time.

REQ-2:

## Settings

4.10.1 Description and Priority

Priority: Low

The settings screen holds various global settings such as whether to show a popup reminding the user to pin the application before giving it to the child.

*<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).>*

4.10.2 Stimulus/Response Sequences

The settings screen is a list of items that can be toggled on or off or edited with a dropdown menu. Each setting may cause changes throughout the application.

*<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.>*

4.10.3 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: The system shall load and display the settings window within 3 seconds 95% of the time.

REQ-2: The system shall connect and update a settings database within 5 seconds of settings being updated 97% of the time.

## Statistics

4.11.1 Description and Priority

Priority: Medium

The statistics screen holds various account-based statistics such as trial progress and accuracy.

*<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).>*

4.11.2 Stimulus/Response Sequences

The statistics screen is a list of information pulled from a database for each account.

*<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.>*

4.11.3 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: The system shall load and display the statistics window within 3 seconds 95% of the time.

REQ-2: The system shall connect and read a database within 2 seconds 96% of the time.

# Other Nonfunctional Requirements

## Performance Requirements

Boi

*<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

Boi

*<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

The system shall not communicate any information externally.

The system shall secure passwords never saving them plaintext.

The system shall comply with all HIPAA mandates.

*<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

Boi

*<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

Boi

*<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

Boi

*<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**

ASD: Autism Spectrum Disorder - A broad range of conditions characterized by challenges with social skills, repetitive behaviors, speech and nonverbal communication.

DTI: Discrete Trial Instruction - A highly structured instruction method developed specifically for children with ASD.

HIPAA: Health Insurance Portability and Accountability Act - A law that protects the privacy of medical patients.

*<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**

Boi

*<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**

Boi

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