

# Analysis Report

Wednesday, April 27, 2022 10:21 PM

When eliciting requirements from the stakeholders, they may not be able to provide optimal phrasing or description for what they want from the software. The stakeholders often have a general idea, but the more important information needs to be established from requirements analysis. This process serves to resolve conflicts between requirements, determine what the software is expected to do, and transform the original set into high-level requirements. The requirements need to be analyzed to ensure they are complete, consistent, feasible, comprehensible, and able to be validated. Requirements are complete when they, alone, define the system to be developed, they are consistent when they are all unique with no overlapping information amongst requirements, feasible if they can be met within the project constraints, comprehensible if they are easily understood, and they must be able to be validated in the sense that each requirement directly leads to the satisfaction of system expectations. Properly analyzed requirements allow the project to have accurate estimates for time and cost while ensuring the project is built as specified in the elicitation phase.

Requirements are analyzed after elicitation and the general process is to take a look at the set of requirements to determine if they are complete, consistent, feasible, comprehensible, and able to be validated. This is done by going through and determining if each individual requirement in the set satisfies these characteristics of a set of requirements, and then either rephrasing or splitting the requirements into better individual requirements as needed to make the set complete and consistent.

Requirements classification, conceptual modeling, negotiation, and formal analysis are all ways to analyze a set of requirements.

- **Requirements classification** is the process of determining if each requirement falls into a certain category, like if it is functional or non-functional, what its priority is, or what the scope of the requirement is.
- **Requirements conceptual modeling** is the process of taking the real-world issue that the software is aiming to solve, and graphically demonstrating it in an easy-to-understand fashion. This can be done with models such as use-case diagrams or dataflow models and it shows how the requirements for the solution will be solving the issue in the whole system. These models not only show how the solution works, but they help with finding requirements that may have been missed or refining the current set of requirements to make them ready for development.

For my project with slalom, I first went through the process of elicitation and gathered user stories for each of the stakeholders. These user stories allowed me to write use cases for the software and come up with my set of requirements for the software. To analyze this set of requirements I performed requirements classification by labeling my requirements' priority, whether they are functional or non-functional, and determining what kind of requirement they were (interface, usability, performance, or limitation). This allowed me to separate the requirements and made further analysis easier. I then went back through my requirements and checked whether each was complete, consistent, feasible, comprehensible, and able to be validated. This led to me changing some of the requirements or even splitting some up into multiple, more specific requirements. Lastly, I went through the list of requirements and used the MITRE system-level requirements checklist as a guide to determine if my set of requirements are complete.

## Prioritized requirements:

## Set of Requirements

Key	Summary	T	Linked Issues	P *	Labels
SLA-29	When a user downloads a reading resource from the BARD, the system shall ask the user for a download destination.	✓	SLA-9, SLA-6, SLA-8, SLA-7	✓	Functional, Interface, Usability
SLA-25	The filtered list shall display resources on a page with no more than 25 results at a time.	✓		✓	Functional, Interface, Usability
SLA-30	When a user visits the BARD home page, the system shall display books that have been recently added within the last month.	✓	SLA-9, SLA-6, SLA-8, SLA-7	✓	Functional, Interface, Usability
SLA-28	When a user downloads an audiobook from the BARD, the user's talking-book device shall receive the download in within 1 minute.	✓		✓	Database, Functional, Performance
SLA-27	When a user searches the BARD for braille or audiobooks, the BARD shall display them in a list with default filters.	✓	SLA-9, SLA-6	✓	Functional, Interface, Usability
SLA-24	When the database receives multiple requests for downloads at the same time, it shall respond to these requests within 1 minute of receiving them.	✓		✓	
SLA-21	When a user downloads a resource from the BARD, the system shall retrieve the resource from the database within 1 minute.	✓	SLA-9, SLA-6, SLA-8, SLA-7	✓	Database, Functional, Performance
SLA-38	The BARD system does not save patron personal information in accordance with PII and PHI government regulations.	✓		✓	
SLA-37	The BARD upgrade shall be completed within 1 year of starting.	✓		✓	
SLA-26	The transition to the new framework for the BARD system shall have a downtime of less than 1 hour.	✓		✓	Nonfunctional, Usability
SLA-23	When a user logs in from an IP outside of the U.S, the authentication will grant them access.	✓		✓	Functional, Performance
SLA-20	When a user attempts to log in for the first time, the authentication system shall request the user to enter a new password.	✓		✓	Functional, Usability
SLA-19	When a user attempts to log in, the authentication system shall authenticate the user's credentials within 30 seconds.	✓		✓	Functional, Performance

## Characteristics of this set of software/system requirements

### Complete

A set of requirements is considered complete when they, alone, define the capabilities, characteristics, constraints, or quality factors without any more information. This set of requirements for the BARD system is complete because the BARD is getting upgrades to its backend framework and database system, and this set of requirements fully describes what the BARD should be able to do, constraints on its abilities, and characteristics that the users can expect of the BARD.

- The set of requirements demonstrates completeness by including all relevant systems for the BARD.
- The set of requirements includes both before and after requirements for this upgrade.
- The set of requirements takes into consideration input from all stakeholders of the BARD system.

### Consistent

A set of requirements is consistent when all requirements are unique from each other. This set of requirements for the BARD contains no redundancy or conflict amongst the requirements, demonstrating consistency.

### **Feasible**

A set of requirements is feasible if it can be materialized within the constraints of the project. With the time constraint of 1 year for completion, these requirements are well within the capabilities of the Slalom staff to complete.

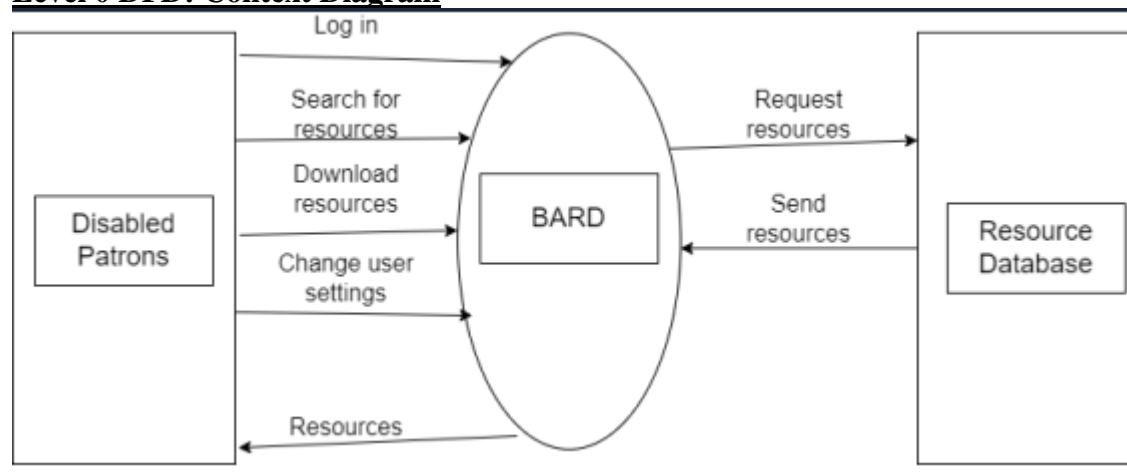
### **Comprehensible**

A set of requirements is comprehensible if the system capabilities and relationship with external systems are outlined. These requirements for the BARD explicitly state the capabilities of the BARD that can be expected with the upgrade, along with the BARD relationship qualities with the cloud database.

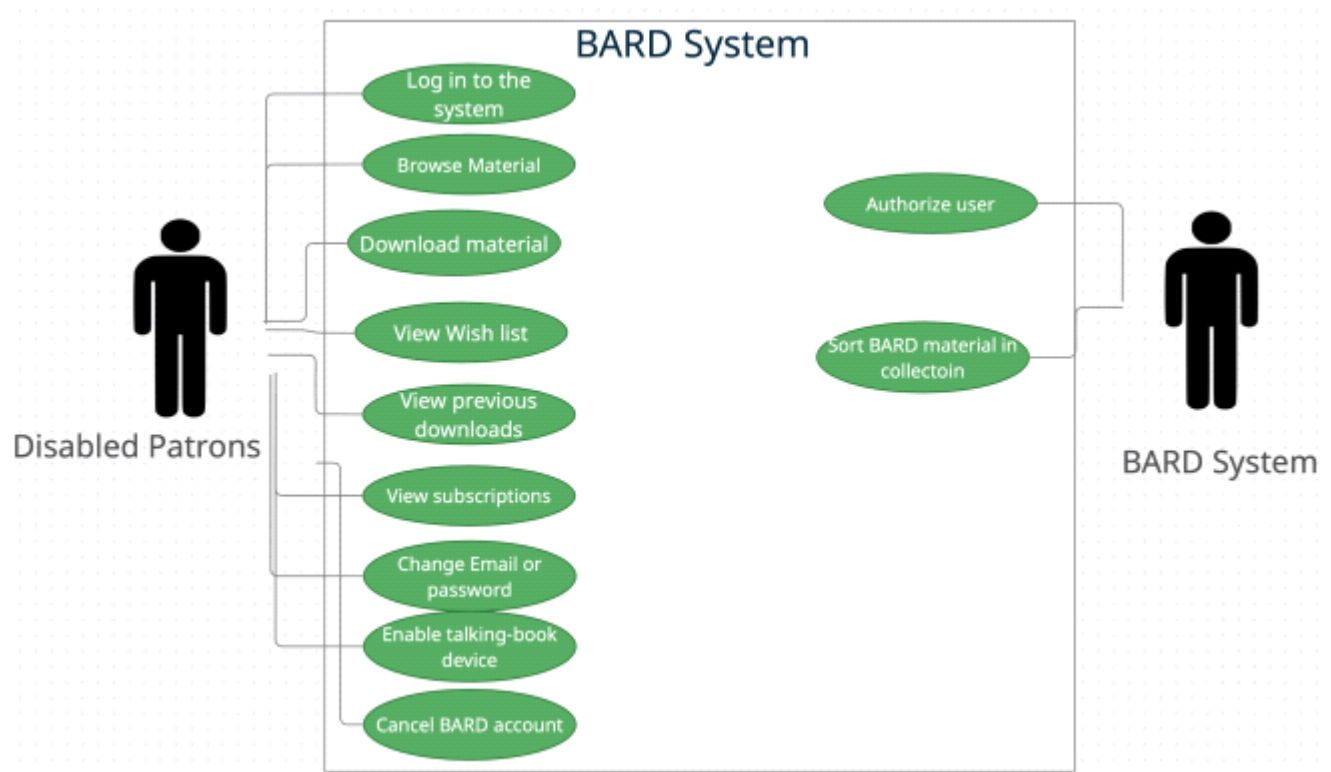
### **Able to be validated**

A set of requirements is able to be validated when the satisfaction of the requirements directly leads to the expected achievement of the system. This set of requirements for the BARD describes clear expectations for the BARD post-upgrade by explaining the expected functionality for users as well as the expected functionality of the system performance.

### **Level 0 DFD: Context Diagram**



### **Use case diagram:**



The elder paul critical thinking model describes important aspects for how to think critically when evaluating information both online and in the world. The model has the following elements of thought:

- Point of view of the information.
  - o The point of view can be used when determining how to turn user stories into requirements because the point of view is the user themselves, and this can often lead to a high level of abstraction when they describe what they want vs how a requirements should actually be written.
- Purpose of the information.
  - o With the purpose of the information kept in mind, I was able to take the requirements that I developed and think about whether they satisfy the complete, consistent, comprehensible, feasible, and validate-able characteristics. If the requirements' purpose didn't fit the project, they would be reworted, scrapped, or broken into smaller more concise requirements.
- Question at issue.
  - o The question at issue was what is expected of the BARD software. This question lead the thought process for requirements because they are meant to describe completely what the system should do, meaning with every requirement I was thinking about whether or not it is answering the question of "Does this describe the expectations of the BARD?".
- What information is present.
  - o The information present is the information I was given from the clients, the Library of Congress, the information I gathered for the user stories, and the answer to the questions I asked in my questionnaire to the stakeholders. I used this information to decide what information should compose the requirements.
- How the information can be interpreted or inferred.
  - o Requirements need to be complete and comprehensive. This means they need to fully describe expectations but also need to not be difficult to understand. This ties into the interpretation of the information because if the requirement is difficult to interpret

- then it can be hard to implement for developers.
- What concepts or theories are being applied to the information.
    - o The concept of requirements engineering is that requirements need to be elicited from stakeholders and then further refined into properly phrased and structured requirements. This means that I needed to take this concept, and apply it to every requirement by deciding if it is phrased or structured correctly, and changing it if not.
  - What assumptions can be made about the information.
    - o It can be assumed that the stakeholders don't know much about the requirements process or even what the system does behind its GUI, so taking this into consideration I am able to take the stakeholders' words and take necessary elements to form the correct requirements for the project.
  - What implications or consequences does the information have.
    - o The implications or consequences of bad requirements is a project with a poor foundation, most likely wrong time and budget estimates, and overall a bad end-product in general. Because of this, I ensure that the requirements satisfy the SWEBOK characteristics of a set of requirements to make sure the project can begin confidently development.

Engineering inherently comes with the personal responsibility to be ethical and professional. This is even more so important when you are creating something that could potentially harm someone in any way. The project I was working on with Slalom wouldn't harm anybody if done incorrectly, but if I were to lack professionalism and not do my duties in the most ethical way possible, then any information I provide or create could be pointless and a waste of time. This shows that it's important to always be professional and ethical when working with any engineering situation, and determining what is best for an optimal outcome comes with the job. Disregarding these aspects of engineering puts money, time, future opportunities/business, and potentially lives on the line.