**Core Components**

**Security**

* User is identified/who is user is known.
* Verify identity of user
* User is identified correctly
* No way to tamper
* No way to have access to any info
* Quantifiable security mechanism to prevent brute force hacking or other types of security violations.
  + Metrics that you an measure and validate
  + How hard will it be to crack o ur software
* Everything has to be secured according to NIST Standards for security
  + Owast website
* All components should be secure according to these guidelines.

**Privacy**

* PII data is not sold or stored.
* PII data can be deleted by request of user.
* Ensure no one other than user has access to that data or component.

PII - Anything that can identify user, including IP address and student Id’s.

**User Management**

Functional

* Create a user
* Delete a user
* Modify a user
* Enable/disable a user

Non-functional

* Operations should take no longer than 5 seconds
* If an operation takes longer, give a progress bar.
* If it takes longer than 30, timeout and give error message
* Scenario 1
  + Success
    - System operations has to be 5 seconds or less
  + Fail
    - If Systems operation exceeds 5 seconds,the window will display a loading bar.
    - If it takes longer than 30 seconds, system must timeout and display an error message

**User Analysis Dashboard**

Functional

* Track every user
* Show how many users have logged in within the day-6 months-year
* Which component is most visited and most used.
* How long was average user section of a user
* User analytics not affected when user deletes account.

Nonfunctional

* Admin has access
* Should take no longer than 5 seconds to load information.
* If longer than 5, notify of progress
* If Longer than 30, time out and error message

**Logging**

Functional

* Log information on every transaction that leads to a success or fail, on the part of hte user.

Nonfunctional

* If successful, data is saved accurately
* Successful if it did not take longer than 5 seconds
* Longer than 6 seconds is a failure
  + Then abort log.

**Archiving**

* Process of backing up the data so whatever you are storing, that data does not get overwritten.
* No more than one month worth of data before archive.

**Error Handling**

* No action should cause the system to become unresponsive or crash.

**Registration**

Functional

* Providing info to determine how to identify the user
* Minimum necessary
* Ties into how you want to implement log in

Nonfunctional

* Shouldn’t take longer than 5 seconds (system response).
* If greater than 5, tell the user something went wrong with the login.
* GIve them another chance to log in

**User access control**

* What a user can do and what a user can’t do
  + Read or write access
* User management is governed by user access control.
* What else was said about this?

**Data store Access**

Functional

* Consistent storage for data.
* Must interact with it
  + Read
  + Update
  + Write
  + Delete

Nonfunctional

* Any database transaction should take no longer than 30 seconds.
* If longer, timeout.

**Network Communication**

Functional

* System has to have features that communicate over a network.

Nonfunctional

* Shouldn’t take longer than 400ms
* Anything longer than 30 seconds should be a timeout

**UI/UX**

Functional

* Ability to adjust font and text sizes
* Ability to change themes and colors.

Nonfunctional

* Intuitive
* Minimal reading
* Choice of colors is good.
* Font and text sized are clear to user
* Directed approach to ui

**Login**

NonFunctional

* System response should not take longer than 5 seconds.
* NOTE: Credentials and voice authentication

**Admin Log in**

Non Functional

* Users should not have access.
* Database manager will admin assign privileges

**Success**

User logs in within 5 seconds.

**Fail**

It takes longer than 5 seconds.

* Notify user something went wrong and time out.

**Tip**

Imagine a developer only gets a portion of the BRD, is that portion enough information to do thorough job?

Write links or references to other parts of application to restate things.