

Architecturally Significant Requirements (ASRs) for CAMDAIS:

Quality Attribute	Attribute Refinement	ASR Scenario
Performance	Fast Response Time	To enable accurate identification of mathematical shortcomings and worries for students, the system needs to process and evaluate a large amount of data in real-time. Students obtain rapid feedback and suggestions because of the quick response time.
	Throughput	At peak demand, the system should be able to deal with a large number of test requests and efficiently execute them. With high throughput, several students can access and use the system at the same time without experiencing any delays.
	Efficient Data Storage and Retrieval	Huge amounts of student data, including test results and historical information, must be stored and retrieved efficiently by the system. Data storage and retrieval procedures that are efficient ensure that the system can handle an immense amount of student data without compromising performance.
	Optimization of Network Latency	When students have poor connectivity to the network, the system should allow quick accessibility to test materials and results. Regardless of the user's location, lowering network latency ensures that data is transferred and received quickly and reliably.
Usability	Efficiency	This system minimizes the time to enter data and retrieve information about features like real time updates. The system successfully completes the process without any input mistakes.

	Learnability	This system provides clear instructions and the process of using the system for the first time .With just one week of training, a new employee with at least two years in the industry may learn to use any of the system's fundamental features in under five seconds.
Configurability	Customizability	The system allows customization of the settings and preferences based on unique needs. The configuration team implements the modification in 1 and tests it.
Maintainability	Modularity	This system is modularly constructed, with each module carrying out a distinct function. In the future, this will make system upkeep and updates simpler.
	Routine Changes	Within three person-days of work, a maintainer who experiences response time issues will fix the bug and share the bug remedy.
	Adding New Feature	Within two person-months, a trading component is developed and effectively implemented.
	Documentation	Each module's operation and interrelationships are explained in detail in the system documentation. Future developers will find it simpler to comprehend the system and make adjustments as needed as a result.
	Testing	To make sure that the system is operating appropriately, it will be thoroughly tested. Both the system as a whole and each module individually must be tested. This procedure can be made more effective by using automated testing.
	Error handling	The system is built to gracefully manage errors. This involves dealing with erroneous network

		connections, unexpected input, and other potential problems. It should be simple to understand and rectify errors if the error message is clear and helpful.
Security	Resisting attacks	In less than 90 seconds, the technology thwarts an unauthorized intrusion attempt and alerts the appropriate authorities.
	Authentication and Authorization	To make sure that only authorized users, such as system users, can access the system, we employed security attributes.
	Data Confidentiality	We make sure that the information gathered from users and preserved in the system is kept private and inaccessible to unauthorized parties.
	Secure Data Storage	We implemented secure data storage mechanisms to ensure that the data collected and stored in the system is not vulnerable to attacks such as data theft or data tampering.
	Data Integrity	In order to guarantee the accuracy and dependability of the data gathered and stored in the system, we implemented security attributes. Implementing strategies like data backup, version control, and checksums can help with this.
Availability	No down time	The system is operational and available 24/7/365 to the students and the other users, with minimal downtime for maintenance and upgrades. We focused on reducing system downtime through measures such as redundant servers, proactive maintenance and system monitoring.
	Scalability	The system is able to handle an increasing number of users and

		data without degradation in performance. We focused on improving server infrastructure, database performance and system architecture to ensure our system can handle more users and requests.
	Load balancing	The system is designed with load balancing in mind to ensure that the system remains available and responsive even during peak usage times.
	Redundancy	The system has redundant components to ensure that there is no single point of failure that could bring down the entire system.