

5. NON-FUNCTIONAL REQUIREMENTS

Quality Requirements:

Usability:

- The system should provide a **clear and structured UI**, allowing users to **navigate easily** (i.e., complete key tasks in under 3 interactions) between functionalities such as “Workload,” “Notifications,” and “Settings.”
- Key actions** (logging workload, requesting leave, checking assignments) should be **accessible within 3 clicks** from the main dashboard, ensuring a smooth user experience.
- The **menu layout should be responsive**, ensuring that options like “Proctoring” and “Course Assisting” are easily clickable across **different screen sizes**, including:
 - Desktop (1280x720+)
 - Tablet (600x800+)
 - Mobile (320x480+)

Reliability:

- The system should store TA workload, assignments, and notifications **reliably**, by:
 - Performing **automatic daily backups**
 - Using **transactional database operations**
 - Implementing **redundancy** in data storage
 - Ensuring recovery is possible within **30 minutes** in case of failure
- Session management**: Users should be automatically logged out after **15 minutes** of inactivity for security purposes.
- A **graceful error handling mechanism** should notify users of any issues (e.g., failed workload entry, network errors) with **clear, actionable messages**, such as “Check your internet connection and try again.”

Performance:

- The backend system should efficiently handle **100+ concurrent requests** from users including TAs, instructors, and administrative staff, ensuring responsiveness even during peak activity periods such as course assignment deadlines or exam weeks.
- The system should support **scalable performance**, making it usable not only for a small group but for **all teaching assistants and academic personnel across the entire engineering faculty**, regardless of department or course load.
- Actions such as logging a workload entry, submitting a leave request, or updating assignment status should receive confirmation within **under 2 seconds** under normal operating conditions.
- The infrastructure should be designed to **scale horizontally**, so that future expansions (e.g., integrating other faculties or departments) can be achieved without a major system overhaul.
- This level of performance ensures that **every engineer in our school**—from first-year course assistants to graduate-level TAs—can actively track, manage, and optimize their workload, helping both academic efficiency and personal time management.

Constraints or Pseudo Requirements:

Implementation:

•The system must be **compatible with modern web browsers**, including:

- Chrome (117.0+)
- Firefox (122+)
- Safari (15+)
- Microsoft Edge (latest versions)

•The **design should be fully responsive**, adapting to:

- Desktop (1280x720+)
- Tablets (600x800+)
- Mobile (320x480+)

•**Frontend:** React.js for a dynamic UI.

•**Backend:** Django (Python) for handling authentication, workload processing, and notifications.

•**Database:** MySQL for storing TA workload, course assignments, and logs.

•**Web Server:** Apache2 on Linux (Ubuntu).

Security & Compliance:

•**Role-Based Access Control (RBAC):** Ensure that only authorized users (TAs, instructors, admins) can access specific functionalities.

•**Data Privacy & GDPR Compliance:** Users must be able to:

- View a **cookie consent banner** before non-essential cookies are stored.

•**Data Encryption:** Store sensitive information (TA contact details, login credentials) using **industry-standard encryption**.

Enhancements Based on Wireframe:

- Left Sidebar:** Ensure that the sidebar (profile details, settings, logout) is **accessible on all screens** without excessive scrolling.
- Top Navigation Buttons:** Ensure buttons like “**Home**,” “**Workload**,” and “**Notifications**” are **prominently visible and clickable**.