Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Hesham | Categories added, drafted up examples for objects | 18/04/2025 |
| 1.1 | Nickleirsch | Drafted sources of requirements | 19/04/2025 |
| 1.2 | Lim Xin Yee | Drafted development context objects | 21/04/2025 |
| 1.3 | Nickleirsch | Explanation for material context objects | 22/04/2025 |
| 1.4 | Nickleirsch | Explanation for immaterial context objects | 22/04/2025 |
| 1.5 | Danesh Veran | Drafted more immaterial and development context objects | 23/04/2025 |
| 1.6 | Danesh Veran | Explanation for development context objects | 23/04/2025 |
| 1.7 | Hesham | Added a source of requierement | 24/04/2025 |

Context objects are elements in and around the system that influence its design and behaviour. We can divide them into two main categories – material and immaterial.

**Material Context Objects:**

1. Students: expected to be the primary users of the system; access academic data, receive notifications, view attendance, etc.
2. Lecturers: educators who will enter academic records, view student data, communicate updates, etc.
3. Administrators: manage system access, maintain academic and financial records.
4. Parents: receive academic and financial updates via SMS and view performance summaries.
5. Campus: physical environment where the system is used — affects network access and system availability.
6. University Database Servers: store academic records, attendance, billing info, and user profiles.
7. Network Infrastructure: enables system access across campus (Wi-Fi, LAN, internet connectivity, etc).

**Immaterial Context Objects:**

1. User Roles and Permissions: which define what each user (student, parent, etc.) can see or do within the portal.
2. Academic Processes: workflows like course registration, grading, and attendance tracking.
3. Communication Protocols: rules that define how messages are sent and received (e.g., email, SMS format, response handling).
4. Administrative Processes: includes billing, scheduling, reporting, and system maintenance routines.
5. SMS/Notification Services: third-party or internal services used to deliver timely messages (alerts, reminders) to users.
6. API Integrations: Interfaces that connect the portal with the campus management system, SMS gateway, etc.
7. Authentication and Access Control Policies: define how users securely log in (e.g., via single sign-on or OTP), session timeouts and password rules.
8. Notification Prioritization and Escalation Logic: logic flow that determines the urgency and format of alerts. For example, critical alerts (e.g., payment deadlines, academic probation) are sent via SMS + portal, while general reminders (e.g., campus events) are sent via portal only.
9. Localization and Language Settings: optional configurations to support multi-language capabilities and regional time/date formats, especially for institutions with international students or multi-lingual local populations.

**Development Context Objects:**

1. Technical Environment: The hardware, software, and platforms used to develop, test, and deploy the portal.
2. IT/Development Team: responsible for designing, coding, testing, and maintaining the portal (developers, system admins, and project managers).
3. Testing Tools & Methods: verifying the portal’s functionality, performance, and security (e.g., unit testing, automated testing, UAT).
4. Budget & Timeline Constraints: cost limits and deadlines that influence scope, development speed, and resource allocation.
5. Version Control System: managing code changes, track development history, and support collaboration among developers.
6. Deployment Strategy: how and when the portal is rolled out to users, including testing phases, rollback plans, and live release timelines.
7. Maintenance and Support Plan: outlines post-launch support, including updates, bug fixes and scheduled maintenance.
8. Security Policies and Protocols: rules ensuring secure coding, user authentication and data encryption.
9. Integration Testing and Staging Environments: dedicated testing systems for integrations (e.g., APIs, SMS Gateway) and ensuring system stability.

**Sources of Requirements**

Stakeholders (students, parents, lecturers, admin) – understand what they need

University policies (rules)

System Project Guidelines

Current system documentation – look at what constraints the current system has