**Functional Specifications**

**For LabEMS**

**2025**

Table of Contents

[1. Introduction 3](#_Toc207215891)

[**1.** **Purpose** 3](#_Toc207215892)

[**2.** **Scope** 3](#_Toc207215893)

[2. System Overview 5](#_Toc207215894)

[**1.** **Role within the Organisation** 5](#_Toc207215895)

[**2.** **Key Improvements Over Current System:** 5](#_Toc207215896)

[**3.** **Target Users** 5](#_Toc207215897)

[**4.** **Environments & Technical Foundations** 6](#_Toc207215898)

[**5.** **General Capabilities** 6](#_Toc207215899)

[3. Functional Requirements 7](#_Toc207215900)

[**1.** **User functions** 7](#_Toc207215901)

[**2.** **Administrator functions** 7](#_Toc207215902)

[**3.** **System functions** 8](#_Toc207215903)

[4. Non-Functional Requirements 9](#_Toc207215904)

[5. Assumptions and Constraints 11](#_Toc207215905)

[**1.** **Assumptions:** 11](#_Toc207215906)

[**2.** **Constraints:** 11](#_Toc207215907)

[6. Acceptance Criteria 13](#_Toc207215908)

[**1.** **Functional Acceptance Criteria** 13](#_Toc207215909)

[**2.** **Non-Functional Acceptance Criteria** 13](#_Toc207215910)

[**3.** **Usability Acceptance Criteria** 13](#_Toc207215911)

[**4.** **Reliability & Maintainability Acceptance Criteria** 13](#_Toc207215912)

[**5.** **Compliance & Governance Acceptance Criteria** 13](#_Toc207215913)

# 1. Introduction

### **Purpose**

The Laboratory Equipment Booking and Maintenance System (LabEMS) is a system designed for the North-West University for bookings of laboratory equipment, digitising the current manual filing system implemented at the university.

This system will consist of a centralised, digital platform enabling users to manage bookings, maintenance and the utilisation of laboratory equipment. The main focus of the system is to eliminate inefficiencies caused by the current manual filing system implemented at the university. The system will also reduce the workload on administrative staff and uphold compliance regulations and requirements.

The system will enable students to reserve laboratory equipment in real time, preventing booking conflicts and ensuring fair access. Plans for equipment maintenance will be set up to include notifications.

LabEMS will create secure and auditable records of all activities that occur on the system relating to the laboratory equipment. This creation of records will allow the university to produce accurate, real-time reports for auditing and resource planning.

By streamlining workflows, data accuracy, general accessibility of information and the use of equipment will be optimised.

### **Scope**

The LabEMS system encompasses the design, development and deployment of a secure role-based web application focusing on the laboratories within the faculty of Natural Sciences at North-West University.

#### In-Scope Features

1. **Real-Time Booking Management**Centralised scheduling system with booking overlap detection, instant confirmation and real-time updates of schedules.
2. **Compliance Reporting**  
   All reporting will be in line with the university's compliance requirements.
3. **Equipment Maintenance Scheduling**  
   Setup plans for equipment maintenance with notifications.
4. **Role-Based Access Control**  
   Permissions are tailored to administrators and laboratory students.
5. **Mobile and Desktop Accessibility**A responsive design to be used on a range of devices.
6. **Data Security**Secure storage of all records, including encryption of records and authentication of users.
7. **Notification emails**  
   Emails will be sent to respective users with updates and reminder notifications with included calendar sync via a .ics file.

#### Out-of-Scope Features

1. Management of non-laboratory equipment
2. Offline or standalone application versions of the system.

#### Stakeholders

1. **Primary Users** - postgraduate students and administrators.
2. **Administrative Staff** - Faculty administrators, compliance staff and the IT support team.
3. **Development Team** - Current final-year students working on the system.
4. **Oversight** - The universities’ management and compliance bodies.

# 2. System Overview

### **Role within the Organisation**

LabEMS is a centralised digital platform that replaces NWU’s manual laboratory management processes (emails, spreadsheets) with an automated, compliance-ready solution. It directly addresses the Business Case’s pain points:

1. **Inefficient bookings** (double-booking risks, lack of real-time visibility).
2. **Manual maintenance tracking** (delayed calibrations, equipment downtime).
3. **Non-compliance** (missing audit trails, unreliable reporting).

### **Key Improvements Over Current System:**

1. **Structured Data Management**Like the Task Management System uses queue/stack system for tasks, LabEMS uses real-time scheduling algorithms to manage equipment bookings and maintenance queues**.**
2. **User-Centric Notifications**  
   Similar to the Task Management System’s deadline alerts, LabEMS sends automated reminders for bookings, maintenance, and compliance deadlines.

### **Target Users**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Role** | **Pain Points (Business Case)** | **LabEMS Features** | **TMS Parallel** |
| **Students** | Uncertainty about equipment availability | Real-time booking dashboard, conflict detection. | TMS’s task visibility & prioritization |
| **Administrators** | Administrative overload (manual logs). | Audit-ready reports, role-based access logs. | TMS’s reporting module (completed vs. pending tasks) |

### **Environments & Technical Foundations**

1. **Platform:**  
   Web-based (responsive design), unlike TMS’s desktop-only scope.
2. **Data Structures**:
   * **Booking Queue**:   
     Prioritizes urgent maintenance (like TMS’s priority queue).
   * **Audit Stack**:   
     Logs historical actions (similar to TMS’s LIFO task history).
3. **Integration**:   
   NWU SSO (vs. TMS’s local authentication).
4. **Scalability**:   
   Cloud-hosted (Azure) supports 200+ users; TMS is limited to local storage.

### **General Capabilities**

Core Functions:

1. **Booking Management**:
   * Real-time conflict detection (like TMS’s task overlap prevention).
   * Extend/cancel bookings (mirrors TMS’s task update/delete).
2. **Maintenance Automation**:
   * Scheduled alerts (cf. TMS deadline notifications).
   * Priority-based scheduling (like TMS’s High/Medium/Low prioritization).

Advanced Features:

1. **Compliance Reporting**:
   * Exportable PDF/Excel reports (TMS’s reporting module, but compliance-focused).
2. **Role-Based Access**:
   * Fine-grained permissions (TMS lacks this; highlights LabEMS’s enterprise-grade needs).

Technical Resilience:

1. **90% Uptime**: Exceeds TMS’s basic reliability goal (local persistence).
2. **Security**: Encryption, POPIA/GDPR compliance (TMS has no security specs).

# 3. Functional Requirements

### **User functions**

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **Priority** |
| **Registration** | Allow users to register using their university information (student no, email, etc) | High |
| **Login /User Authentication** | Allow users to login into the system using unique credentials | High |
| **Browse Equipment** | Users must be able to view a list of all available laboratory equipment including details such as name, description, location | High |
| **View Equipment Availability** | Display real-time availability of the equipment (available, booked, under maintenance) | High |
| **Book Equipment** | Allows users to reserve equipment for a specific time slot | High |
| **Request Booking Extension** | Allow users to extend their current booking, if the equipment is available for the extension period | Medium |
| **Cancel booking** | Allow for the cancellation of booking. | Medium |
| **Manage account** | Allow users to update their profile information and review their past, current, and future bookings. | Medium |

|  |  |  |
| --- | --- | --- |
| **Administrator functions** | | |
| **Function** | **Description** | **Priority** |
| **Login /Authentication** | Securely login into the system to access the administrator dashboard. | High |
| **Manage User Profile** | Add a new user, update existing user information, and deactivate the user account. | High |
| **Manage Equipment** | Add new equipment, update existing equipment information, remove/delete obsolete equipment. | High |
| **Manage Booking** | Create new booking, update existing booking and cancel a booking. | High |
| **Schedule Equipment Maintenance** | Schedule equipment maintenance and update system once maintenance is complete | High |
| **Generate Reports** | Generate reports detailing equipment usage, bookings, and user activity. | Medium |
| **Export Reports** | Download generated reports for record-keeping or analysis. | Medium |

|  |  |  |
| --- | --- | --- |
| **System functions** | | |
| **Function** | **Description** | **Priority** |
| **Notify User - Booking Confirmation** | Send a notification to the user to confirm the details of a successful booking. | High |
| **Notify User - Booking Expiration** | Remind the user that their booking is about to expire. | High |
| **Notify Administrator - Equipment Booked** | Notify the administrator whenever a piece of equipment is successfully booked. | Medium |
| **Notify Administrator - Maintenance Due** | Notify the administrator when equipment is due for its scheduled maintenance. | High |

# 4. Non-Functional Requirements

1. **Portability and compatibility**

The system must be able to run on many common browser environments such as Google Chrome, Microsoft Edge, Firefox, Opera mini and Brave, without additional modifications. The host operating system environments include Android, MacOS and Windows. The interface must be consistent on both mobile and desktop-based use cases.

1. **Performance**

The system must be responsive under the normal load and sustain throughput to handle a usage volume of 200 active users.

1. **Reliability and availability requirements**

The system must have at least 90% availability to handle daily traffic volumes under normal conditions. The system must handle integrity errors and be able to rollback any uncommitted data in the event of abrupt external failure.

Downtimes must be planned for the hours and days where traffic is minimal such as after midnight on a weekend.

1. **Maintainability**

The system must have in-built logging to handle critical maintenance issues. The preliminary maintenance efficiency target is estimated at 85%. The accurate Mean Time to Restore the System (MTTRS) will be generated once the system is operational with real logging data.

The code base must be readable, well commented, documented and consistent with the industry best practices.

1. **Security and compliance needs**

The system must be able to integrate standard industry security measures to guard against security threats such as SQL injections, Man in the middle attacks, viral etc. Unauthorized users will not be able to edit, delete, update or view the data. A comprehensive authorization and authentication scheme should be in place to prevent illegal access.

Compliance measures to keep consistent with are Protection of Personal Information Act (POPIA), and General Data Protection Regulation (GDPR). The POPIA asserts the user that their data will be protected and private to even the developers. For example, passwords will be hashed before storage. GDPR is a contract stipulating to the end-users what their information will be used for in the application.

1. **Usability expectations**

The system must provide a clean minimalistic laboratory-themed user interface with consistent typography and imagery. Users will be able to find desired equipment with a few clicks and search function. Multiple users will be able to use the system at different places and at the same time.

Usability tests will be conducted to uphold the standard based on the user inputs. The KPIs identified in the functional requirements that must be achievable by all the authorized users.

Convenient downloadable user guides and frequently asked questions will be enabled to improve the usability and help new users adopt seamlessly.

1. **Scalability**

The system must be able to handle current and future traffic workloads which is estimated at 200 users. The current scaling technique is vertical. Which can be used in the initial beta phase. Higher traffic workloads prompt for horizontal scaling with the added advantage of improved system resilience.

# 5. Assumptions and Constraints

### **Assumptions:**

1. **Infrastructure availability**

North-West University will provide the necessary infrastructure for the system to be fully operational. No additional equipment or software will need to be acquired in the initial release.

1. **Authentication and Identity Management**

All users will already have valid NWU credentials (student number, password and university email account) which can be used for secure login. It is also assumed that NWU’s existing Single Sign-On services are operational and can be used for role-based access to the system.

1. **Compliance and Policies**

LabEMS will only be compliant with current North-West University policies. All data security requirements are already aligned with NWU’s existing governance policies. If any changes are made in the future the system will have to be re-evaluated and adjusted to ensure ongoing compliance.

1. **Stakeholder availability for testing**

Laboratory staff, administrative personnel and selected student representatives will be available to provide feedback and recommendations throughout the project lifecycle.

1. **Device accessibility**

All users will have access to at least one compatible device (PC, smartphone, lab computer) that can run a modern browser.

1. **User Readiness**

All users are willing to transition from the current manual booking system to LabEMS and minimal training, user guides and FAQs will be sufficient for adoption.

### **Constraints:**

1. **Budgetary constraint**

LabEMS will only have a budget of R1600 that has been allocated to web hosting. Paid third-party solutions and commercial services cannot be used and free options need to be implemented instead.

1. **Timeline constraint**

The system needs to be fully operational and ready for implementation by the 24th of October 2025. Strict adherence to project timelines is crucial as no further extensions will be permitted.

1. **Technical constraints**

LabEMS must operate within NWU’s IT infrastructure and must be compatible with commonly used browsers (Chrome, Edge, Firefox, Opera, Brave). Due to budget and time limitations a desktop version cannot be implemented.

1. **No Offline Functionality**

The system will only be functional with an internet connection and offline functionality is excluded from the project scope.

1. **Limited scalability in initial release**

LabEMS will support up to 200 users in the first release. Time and budget constraints limit the scale initially, but the solution is designed to be scalable to adjust for future needs. Any future expansions will be at an additional cost and are therefore not included in the system at this time.

1. **Human Resource Constraint**

The development will be carried out by a small group of final-year students with limited manpower and expertise. This will restrict the complexity and size of features that can be realistically implemented.

# 6. Acceptance Criteria

### **Functional Acceptance Criteria**

* Users must be able to log in with their NWU credentials and access only their role-based features (tested via login demo).
* Equipment cannot be double-booked (tested with conflict scenarios).
* The system must send booking confirmation emails within 1 minute of booking (tested live).
* Equipment availability updates in real-time when a booking or maintenance is logged (tested in demo).

### **Non-Functional Acceptance Criteria**

* The system responds to booking/search requests in ≤ 2 seconds for 200 users (performance test).
* 90%+ uptime during the semester (monitored via logs or PRTG, etc).
* Mobile and desktop UI display consistently (browser & platform test).
* Security audit confirms compliance with POPIA, GDPR, ISO standards (audit report).

### **Usability Acceptance Criteria**

* 50%+ of test users complete a booking in ≤5 clicks.
* At least 70% of surveyed users rate the system “easy to use” after pilot rollout.
* User guides and FAQs are accessible and downloadable from the system (demonstration).

### **Reliability & Maintainability Acceptance Criteria**

* Automatic rollback occurs when a booking fails mid-transaction (tested simulation).
* Logs are generated for all admin actions (review logs).
* MTTR (Mean Time to Restore) ≤ 4 hours in case of downtime (post-maintenance review).

### **Compliance & Governance Acceptance Criteria**

* Reports generated by the system must match compliance body formats (tested by mock audit).
* Data retention and deletion policies align with POPIA & GDPR (compliance audit check).