Software Requirements Specification

for

WorkBit

Version 1.0 approved

Prepared by

Mayo Ramos Angel David

Alvarez Galindo Aldo Yamil

Gomez Miramontes Daniel

Muñoz Reynoso Oscar Gael

03/06/2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

This document defines the architecture, modules, and technical specifications of the system Workbit. It is intended for developers, project supervisors, academic IT personnel, and facility managers responsible for its deployment and maintenance.

1.1 Document Conventions

The Workbit system will be deployed in designated study rooms, reading areas, and small classrooms. Its functionality supports environmental monitoring, attendance tracking, room readiness automation, and analytics dashboards for resource optimization.

1.2 Intended Audience and Reading Suggestions

It is targeted for developers, project supervisors, academic IT personnel, and facility managers responsible for its deployment and maintenance.

1.3 Product Scope

This project consists of developing an integrated smart management system for work cubicle spaces. Combining IoT capabilities, role-based digital access, and multi-platform monitoring, the system optimizes environmental conditions, automates room readiness, and enhances usage logging and operational efficiency within educational institutions, companies and different organizations.
The system detects authorized entries via RFID, monitors room current status, and activates necessary environmental systems. All users—including users, staff, and administrators—can access tailored tools and real-time insights through a centralized platform available on web and mobile.

1.4 References

Krüger, G., & Lane, C. (2023, January 17). *How to Write an SRS (Software Requirements Specification Document)*. Perforce Software. Retrieved June 7, 2024, from https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document

2. Overall Description

2.1 Product Perspective

WorkBit is a modular and scalable smart management system designed for educational institutions, libraries, coworking spaces, and other collaborative environments. Each designated area such as rooms, study booths and cubicles is equipped with IoT environmental monitoring and

RFID-based access control, integrated into a centralized platform for data visualization and administrative control.

- **Environmental Monitoring**
  Continuously measures temperature, humidity, and CO_2 levels in real time.
  All data is logged and analyzed to ensure air quality compliance and trend detection.

- **Access Logging**
  Tracks the entry of students and staff through RFID card scanning.
  Each access event is timestamped and stored to analyze usage patterns and occupancy time.

- **Smart Room Preparation**
  Automatically activates ventilation and lighting systems upon authorized entry.
  Air quality is visually indicated through an LED traffic light system, promoting a safe and responsive environment.

2.2 Product Functions

The system provides the following key functionalities:

- **Space Reservation**
  Users can reserve rooms or study booths through the mobile app, depending on the area and availability. Each reservation is linked to the user's credentials, ensuring secure access.

- **Environmental Conditions Overview**
  Users can check real-time room conditions—such as temperature, humidity, and CO_2 levels—before entering, allowing them to choose a comfortable and safe environment.

- **Access and Usage Records**
  Administrators have access to detailed logs of all room usage, including entry timestamps and occupancy levels. This enables effective monitoring of space utilization and adherence to safety protocols.

2.3 User Classes and Characteristics

User Type	Abilities	Activities
User (Client/Student/Employee)	View, Book	View room availability, air quality, and environmental data. Book and cancel reservations.
Operator (Staff)	View, Control	Open cubicles, trigger ventilation/lights, monitor sensor data in real time.
Facility Admin (Local Admin)	Manage bookings, sensors, and staff	Assign cubicles, edit device thresholds, view usage statistics, handle reports.
System Admin	Full Access	Manage institutions, users, cubicles, devices, database, security policies.

2.4 Operating Environment

The system is being developed across multiple environments to support both web and mobile platforms. The web application is built using C# with ASP.NET Core within Visual Studio, while the mobile application is developed using React Native and Expo within Visual Studio Code. Although the development process is conducted on Windows-based systems, the final product is designed to be cross-platform, ensuring compatibility with all major operating systems, including Windows, macOS, Linux, Android, and iOS.

2.5 Design and Implementation Constraints

The main limitation is the use of certain technologies imposed in the development of the project, limiting us in terms of the use of better tools and/or technologies.

2.6 User Documentation

- Manual: An instruction manual on the installation and operation process of the program.
- Contact card: With the contacts and necessary information in case the program fails or the user needs any information.

2.7 Assumptions and Dependencies

We'll use Visual Studio and Visual Studio Code dependencies as a programming system in which we will create our software, just as we will use the SQL Server system and MongoDB to create our database.

3. External Interface Requirements

3.1 User Interfaces

The user interfaces will be a mobile app and a web app, which users will have access to according to their roles.

Mobile App

- Space Reservation

The user will be able to reserve a workspace by choosing the vacancy, time, reason, and a person responsible for the space.

- Space Monitoring

The user will be able to monitor the reserved space using sensors installed in each room.

Web App

- Condition Monitoring

Monitor all workspaces using all the sensors in each room, as well as monitor the number of people remaining in each room.

3.2 Hardware Interfaces

Temperature and humidity measurement using DHT11 sensor

The system must measure temperature and humidity in each conference room using the DHT11 sensor. These values will be sent periodically to the server for real-time access via the web and mobile interfaces.

CO_2 monitoring using MH-Z19 or CCS811 sensor

Each room must include a CO_2 sensor to monitor air quality, ensuring healthy environments for attendees and signaling when ventilation is necessary.

RFID card reading using MFRC522

Attendees and staff must scan their RFID cards using the MFRC522 reader when entering a conference room. The ESP32 will log this event (user ID, room, date, time) and send it to the server as an entry record.

Data transmission from ESP32 to server via Wi-Fi

All data from sensors (temperature, humidity, CO_2, RFID scans) must be sent wirelessly via the ESP32's Wi-Fi to the server using HTTP requests.

Environmental status LED indicator (traffic light)

A LED-based visual indicator must reflect the air quality in the room: green for good, yellow for moderate, and red for poor conditions based on CO_2 levels. This aids in quick, visual evaluation of indoor air health.

3.3 Software Interfaces

- **Backend**: ASP.NET Core with RESTful API
- **Database**: SQL Server and MongoDB
- **Frontend**: React for Web, React Native for Mobile

3.4 Communications Interfaces

The system will rely on Wi-Fi communication for all ESP32-connected IoT components.
Each ESP32 microcontroller will transmit sensor data and RFID scans over the local network or internet.
Data will be sent via HTTP POST requests to the ASP.NET Core backend.

This setup:
- Eliminates the need for physical wiring.
- Enables centralized monitoring.
- Supports scalability, allowing more rooms and devices to be added easily.
- Ensures compatibility with existing network infrastructures in modern event venues.

4. System Features

4.1 System Feature 1

4.1.1 Description and Priority

Space reservation in the mobile app (high priority)
Workspace monitoring in the mobile app (medium priority)
Workspace monitoring in the web app (high priority)
Viewing backup data with different search criteria (high priority)
Workspace monitoring using sensors (high priority)
Access via card (high priority)

4.1.2 Stimulus/Response Sequences

- **Use Case: Room Reservation (Mobile App)**
  Stimulus: User opens the mobile app and selects a date and available room to reserve.

  Response:

  System queries availability.

  System displays a list of available rooms.

  Upon user confirmation, the system creates a reservation.

  Confirmation and reservation details are shown to the user.

- **Use Case: Room Entry via RFID**
  Stimulus: User swipes RFID card at the room's entry reader.

  Response:

  ESP32 reads the RFID and sends user ID, timestamp, and room ID to the backend.

  Backend validates if the user has a valid reservation.

  If authorized:

  Entry is logged.

  Access is granted (door opens or LED indicator shows green).

  If unauthorized:

  Entry is denied.

  User receives a denial notification in the app.

- **Use Case: Environmental Monitoring**
  Stimulus: ESP32 periodically reads sensor values (temperature, humidity, CO_2).

  Response:

  Values are transmitted to the server via Wi-Fi.

  The backend stores and processes the data.

  Admin portal and mobile app update in real-time.

  If thresholds are breached, a notification is triggered and the LED traffic light updates status.

- **Use Case: Admin Monitoring Dashboard**
  Stimulus: Admin logs into the web portal and selects a specific room.

  Response:

  System loads and displays:

  Current sensor data.

  Historical trends.

  Recent access logs.

  Upcoming and past reservations.

  Admin may generate or export reports as needed.

- **Use Case: Alert Notification**
  Stimulus: Sensor detects high CO_2 level (above configured threshold).

  Response:

  The LED traffic light changes to red.

  Backend triggers a push notification.

  Users with active reservations in the affected room receive alerts on their mobile devices.

  Admin receives an alert on the web dashboard.

4.1.3 Functional Requirements

1. IoT Module:

- Environmental Monitoring
  - The system must monitor real-time temperature, humidity, and CO_2 levels in each study or work area.
  - The ESP32 microcontroller must collect sensor data and transmit it to the backend server over Wi-Fi.

- An LED traffic light indicator must visually display air quality levels:
    - Green = Optimal
    - Yellow = Moderate
    - Red = Poor – ventilation recommended

- RFID-Based Access Control
    - The system must automatically log the entry of users through RFID cards.
    - Each room or cubicle must be equipped with an MFRC522 RFID reader connected to an ESP32.
    - The ESP32 must send access logs containing user ID, timestamp, and space ID to the server for validation and storage.

2. Web Module:

- User and Classroom Management:
    - The system must allow administrators to manage:
    - Users (students, clients, technicians, administrators)
    - Study/work areas
    - It must enforce a role-based access model with permissions such as:
    - Administrator: full access
    - User: limited to viewing and booking
    - Role and permission enforcement must be handled by the backend logic.

- Data Visualization and Management:
    - Administrators must be able to:
    - View reservation history and entry logs
    - Monitor environmental metrics for each space
    - Technicians must be able to:
    - Monitor real-time sensor data
    - Review access logs for their assigned areas
    - The system must support report generation by space, user, or time period (daily, weekly, monthly).

- Room Automation
    - When a designated technician accesses a room, the system must automatically activate connected systems such as:
    - Lights
    - Fans or HVAC units (if available)

3. Mobile Application Module:

- Information Visualization:

  Users must be able to:

    - View their bookings, access level, and assigned room schedule.
    - Check environmental conditions of currently booked rooms.
    - Technicians must have access to:
    - Sensor readings
    - Room assignments and any malfunction reports
    - All users must receive push notifications for:
    - Air quality warnings
    - Access status changes

- Alerts:
    - The system must issue real-time alerts when:
    - Air quality exceeds safe thresholds
    - User access is denied due to insufficient permissions

- Control and Communication:
    - Users must be able to:
    - Manage their profile
    - Receive system messages and updates from the admin
    - Technicians must be able to:
    - Report issues or malfunctions directly from the app interface

4. Database:
The system database must support real-time and historical operations and include:

- A Users table with roles such as: Administrator, Technician, Standard User, (Student/Employee)
- A Spaces table for each study or work area.
- A Sensors table linked to each space (e.g., DHT11, MH-Z19).
- An Access Logs table for RFID-based entries.
- An Environmental Readings table storing:
- Room ID
- Timestamp
- Temperature, Humidity, CO_2 values
- An Automation Configurations table for predefined environmental thresholds and device behaviors.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Environmental monitoring must be in real time.
- Data visualization must be in real time (web and mobile).
- The system must alert about non-optimal environmental conditions in real time.

5.2 Safety Requirements

- Data encryption to protect transmission.

- User authentication using secure protocols.

- Implementation of periodic database backups.

- Real-time logging and monitoring of critical events.

5.3 Security Requirements

- All sensitive data (usernames, passwords) must be encrypted.
- Access to information must be based on user roles and enforced via backend.

5.4 Software Quality Attributes

Maintainability
Maintenance must be performed by:

- Backend/frontend developers for app or website errors.
- IoT technicians for hardware issues (ESP32, sensors, RFID).
- DB administrators for SQL and NO SQL Server performance/security.
- Trained internal support staff for basic resets and diagnostics.

Portability

- Programming Languages: JavaScript (React), C# (ASP.NET Core), C++ (ESP32 firmware).
- OS Independent: Runs on Windows, Linux servers; Android/iOS mobile devices.
- Not bound to any proprietary server technology.

5.5 Business Rules

BR1 — Role-Based Access Control
Only users with valid credentials and assigned roles may access the system.

Roles include: Administrator, User(e.g., student or employee) ad Operator Staff.

Permissions are strictly defined and enforced by the backend logic.

BR2 — Reservation Policy
Standard Users can only make reservations during available time slots defined by the system.

Users may only access rooms they have successfully booked in advance.

Reservations must include a start time, end time, and room ID.

BR3 — Access Validation
Entry to rooms is granted only if:

The user has an active reservation for the corresponding room and time.

The RFID card matches a registered user in the system.

Unauthorized users are denied access and the event is logged.

BR4 — Environmental Alerts

Alerts triggered when CO_2/temperature exceeds thresholds.

LED status changes and push notification sent.

BR5 — Administrator Rights

Admins manage users, spaces, sensors, and logs.

Full access to configuration and reporting.

BR6 — Data Handling

Logs retained for at least 6 months.

All interactions are auditable.

6. Other Requirements

Scalability:

- The system must be modular and scalable to accommodate more rooms, sensors, and users.
- Backend and database must allow horizontal scaling.

Ease of Implementation/Integration:

- Any conference venue must be able to implement the system without high technical complexity or cost.
- Must be compatible with standard Wi-Fi networks and common servers (Windows/Linux).

Usability/User Experience:

- The system must offer a clean, modern, and intuitive interface.
- Web and mobile UIs should support light/dark themes.
- The system must improve communication between organizers, technicians, and attendees.