flowsync

MILESTONE 3 REQUIREMENTS & ANALYSIS REPORT: SMART STORE OPERATIONS AND STAFF MANAGEMENT SYSTEM

COURSE: PRG800NAA
PROJECT: FLOWSYNC - SMART STORE OPERATIONS AND STAFF MANAGEMENT
TEAM: FLOWSYNC
SUBMISSION DATE: 10/02/2026

SUBMITTED TO
MILES MCDONALD

TEAM MEMBERS:
JAGAT PATEL: 118557248
ASINTHA JAYASEKARA: 170388235
ISURI KURUWITA: 120182241

flowsync

Table of Contents

List of Figures

List of Tables

1 Executive Summary

FlowSync is a web-based system made to help fast-food restaurants run daily work in an easier way. In many restaurants, staff still use paper checklists and give instructions by talking. Because of that, some tasks get missed, shift handoffs become messy, and sometimes items run out during busy hours. This can cause delays, mistakes, and stress for the team.

The main goal of FlowSync is to make store work more organized and clearer. It puts everything in one place, so managers and staff can follow the same process every shift. FlowSync helps with daily checklists, assigning roles and tasks, tracking progress, checking inventory, helping new staff learn faster, and sharing shift notes.

FlowSync main features are:

- Digital checklists for opening, mid-shift, and closing tasks
- Role and task assignment so managers can clearly give duties to staff
- Task completion tracking so staff can mark tasks done and managers can see progress
- Inventory tracking and low-stock alerts to avoid running out of important items
- Shift handoff notes so the next shift knows what happened and what is still pending
- Training materials so new staff can follow steps and learn faster
- Secure login and access control so users only see what they are allowed to see

By replacing paper and verbal updates, FlowSync helps the store finish work on time, reduce mistakes, and improve teamwork. It also helps managers control the shift better because they can see what is done, what is late, and what needs attention.

The first version of FlowSync will include the most important features like checklists, task assignment, task tracking, inventory updates, and shift handoff notes. Later versions can add reports, analytics, and support for multiple store locations.

2 Introduction

Fast-food restaurants have many tasks to do every day. For example, opening duties, cleaning, food prep, serving customers, and closing duties. These tasks must be done on time and in the correct way.

In many restaurants, staff still use paper checklists and give instructions by talking. Because of this, some tasks can be missed. Staff may not know who should do what. Shift handoff can also be unclear. Inventory problems can happen too, like running out of items during busy hours.

These problems can cause slow service, more mistakes, and more stress for staff. It also makes the store less efficient.

FlowSync addresses these challenges by:

- Replacing paper checklists with digital checklists (opening, mid-shift, closing) to make sure tasks are not missed.
- Helping managers assign roles and tasks clearly so every staff member knows what to do during the shift.
- Tracking task completion in real time so managers can see progress and follow up on overdue tasks.
- Supporting inventory tracking and low-stock alerts to help prevent running out of important items.
- Improving shift handoff communication by allowing supervisors to record notes for the next shift.
- Providing training and reference materials so new staff can learn steps quickly and reduce repeated mistakes.
- Using secure access and permissions so users only access what they are allowed to view or update.

This report explains the system requirements, stakeholders, use cases, system design parts, and non-functional requirements. These details will guide the development of FlowSync and help it match the needs of a fast-food restaurant.

3 Stakeholder Analysis

3.1 Stakeholders and Their Primary Interests

Stakeholder	Interest Roles	Actor in System
Store Manager	Assign roles/tasks, monitor completion, manage inventory, view shift status	Yes
Shift Supervisor	Run shift operations, ensure checklist completion, handoff communication	Yes
Store Staff (Crew)	View assigned tasks, update task status, access training steps	Yes
New Employee / Trainee	Access training guides and procedures quickly	Yes
Inventory / Stock Responsible Staff	Update stock counts, view low-stock alerts	Yes (can be same as staff)
Restaurant Owner (or Franchise Owner)	Better performance, fewer errors, consistent operations	Optional (view-only)
IT Admin (System Admin)	User accounts, roles/permissions, configuration	Yes
Project Team	Designs and develops the system	No
Instructor / Sponsor	Evaluate completeness/quality of analysis and deliverables	No

Table 1: Stakeholders and Their Primary Interests (FlowSync)

3.2 Power-Interest Chart & Communication Strategy

Stakeholder	Power	Interest	Communication Strategy
Store Manager	High	High	Weekly check-ins, short demos, confirm requirements, approve key decisions
Shift Supervisor	High	High	Weekly feedback sessions, test shift features, confirm workflow and checklist steps
Store Staff (Crew Members)	Medium	High	Simple training guide, quick feedback form, collect usability feedback during testing
New Staff / Trainees	Low	High	Provide step-by-step training pages, observe onboarding usage, gather clarity feedback
Stock / Inventory Responsible Staff	Medium	High	Training on inventory update flow, confirm low-stock alert rules, weekly review of issues
Owner	High	Medium	Monthly milestone summary, benefits overview, basic performance/report updates (read-only)
IT Administrator / System Admin	High	Medium	Technical documentation, security/roles review, periodic system setup meetings
Instructor / Sponsor	Medium	Medium	Milestone submissions, short presentations, progress updates based on rubric

Table 2: Power–Interest Chart and Communication Strategy

3.3 Context Diagram (Description) - FlowSync

The Context Diagram shows a high-level view of FlowSync as one complete system. It shows the system boundary and how FlowSync connects with outside users (external entities). It also shows what information goes into the system and what information comes out of the system.

External Entities

Store Manager / Shift Supervisor

Creates and manages shifts, assigns roles and tasks, updates checklists, posts announcements, updates inventory, and checks shift performance.

Store Staff / Trainee

Views assigned tasks, completes tasks, reports issues, reads announcements, and uses training materials to learn procedures.

IT Administrator

Creates user accounts, manages roles and permissions, and configures system settings to keep access secure.

Owner

Views summary reports and store performance information (read-only access).

Data Flows

Store Manager / Shift Supervisor → FlowSync

- Create shift
- Assign roles and tasks
- Create/update checklists
- Post announcements
- Update inventory
- Review performance

FlowSync → Store Manager / Shift Supervisor

- Dashboard status (what is done / pending)
- Task completion reports
- Low stock alerts
- Shift handoff summary

Store Staff / Trainee → FlowSync

- View tasks
- Mark tasks complete
- Submit issue notes
- Read training materials

FlowSync → Store Staff / Trainee

- Assigned task list
- Checklist steps
- Training guides
- Announcements and notes

IT Administrator → FlowSync

- Create users
- Manage roles and permissions
- Configure store settings

FlowSync → IT Administrator

- User list
- Audit logs
- System/configuration status

Owner (Optional) → FlowSync

- Request report view

FlowSync → Owner (Optional)

- Summary reports (read-only)

This context-level diagram shows FlowSync as the main system for store operations and staff management. It clearly shows how information moves between FlowSync and the external users (Store Manager/Shift Supervisor, Store Staff/Trainees, IT Administrator, and the optional Owner). Because the data paths are clear, everyone can understand what the system does and what each stakeholder must do. This helps both technical and non-technical people communicate in a clear way, reduces confusion about system scope and information sharing, and gives a strong base for deeper system analysis in the next steps.

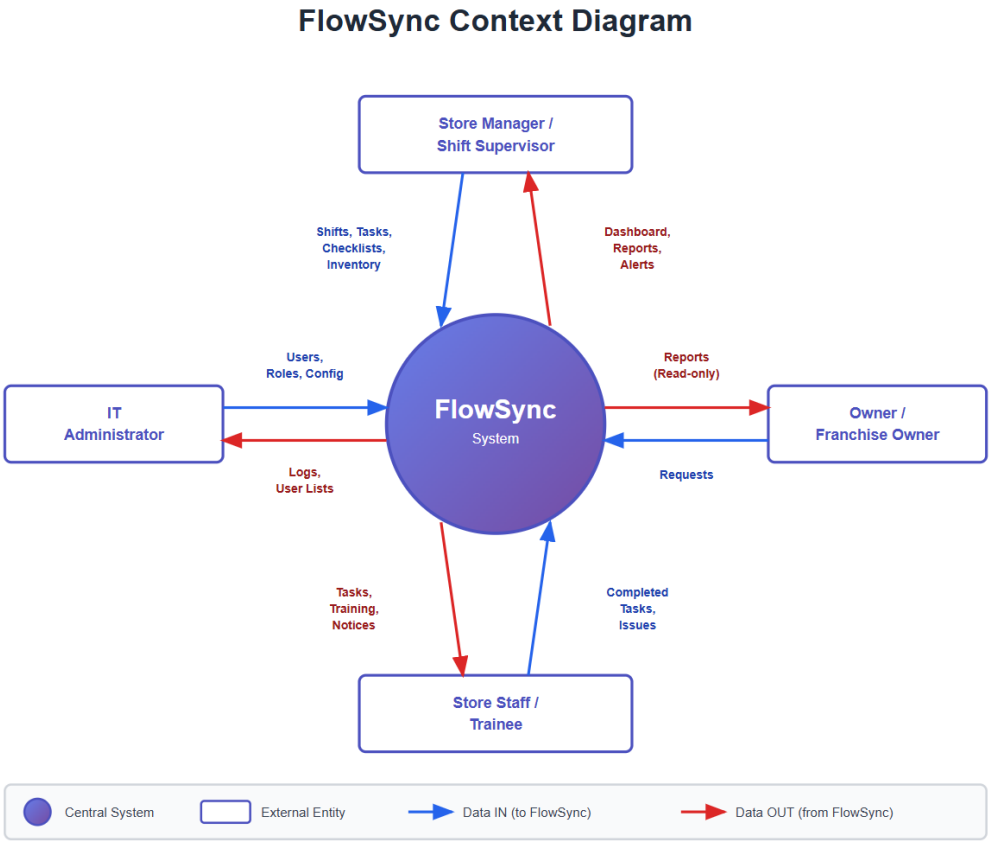

Figure 1: Context Diagram

4 Use Cases

4.1 Event Analysis Table

Event type	Event	Use case triggered	Result / Output
External	Store Manager/Shift Manager starts a new shift	Create Shifts	Shift created and became active
Temporal	Opening time is reached (start of day)	Generate Opening Checklist	Opening checklist is created/available for the shift
External	Store Manager assigns roles and tasks to staff	Assign Roles & Tasks	Roles and tasks are assigned and saved
External	Roles/tasks are assigned to staff	Notify Staff	Staff receive assigned task details/notification
External	Staff needs to check duties for the shift	View Task List	Staff can see their assigned tasks
External	Staff completes a task	Update Task Status	Task marked complete and progress updated
External	Staff finds a problem while doing tasks	Report Issue	Issue is recorded and visible to manager/supervisor
External	Staff wants to share comments/feedback	Submit Feedback	Feedback is saved for manager review
External	Manager wants to see shift progress	View Dashboard Status	Dashboard shows done/pending tasks and current status
External	Manager wants detailed task results	View Task Reports	Task report is displayed for performance review
External	Manager checks performance for the shift	Review Performance	Performance information is shown and reviewed

External	Supervisor writes a note for next shift	Record Shift Handoff Note	Handoff note saved for next shift
External	Inventory count is updated	Update Inventory	Inventory quantity is updated and saved
State	Inventory goes below reorder level	Check Low Stock	Low stock is detected and alert can be created
State	Low stock is detected	Low Stock Alert	Alert is created and shown to manager
Temporal	End of shift time is reached	Close Shift	Shift is closed and locked for updates
Temporal	Shift is closed	Generate Shift Summary	Shift summary report is generated
External	Owner wants to view reports (read-only)	Request/Generate Report	System prepares report for viewing
External	Owner opens summary reports	View Summary Reports (Read-Only)	Owner can view reports (no editing allowed)
External	IT Admin manages security settings	Security Administration	Security settings are updated
External	IT Admin manages access and permissions	Manage Roles and Permissions	Roles/permissions updated and enforced
External	IT Admin assigns a role to a user	Assign User Role	User role is assigned and saved
External	IT Admin wants to view logs	Access Audit Logs	Audit logs and usage reports are displayed
External	IT Admin views all users	View User Lists	User list is displayed
External	IT Admin adds/edits/removes users	Manage Users	User accounts updated and saved

Table 3: Event Analysis Table (Events, Triggers, and System Outputs)

4.2 Use Case Diagram

The Use Case Diagram shows how different users use the FlowSync Store Operations and Staff Management System. It explains the main things the system can do, and which user can do each action.

System Boundary

FlowSync Store Operations and Staff Management System - a secure web system used to manage shifts, checklists, staff tasks, inventory, training, reports, and logs.

Actors

Store Manager / Shift Manager: Manages the shift, gives tasks to staff, checks progress, updates inventory, writes handoff notes, and closes the shift.

Store Staff / Trainee: Views assigned tasks, marks tasks done, reports issues, reads training materials, views announcements, and submits feedback.

IT Administrator: Creates and manages user accounts, controls roles and permissions, and checks audit logs.

Owner: Can view reports and summaries but cannot edit anything.

Primary Use cases

Store Manager / Shift Manager can:

- Create Shifts
- Assign Roles & Tasks
- Update Inventory
- View Dashboard Status
- Review Performance
- Record Shift Handoff Note
- Close Shift
- View Low Stock Alert

Store Staff / Trainee can:

- View Task List
- Update Task Status
- Report Issue
- Submit Feedback
- View Training Materials

- View Announcements

IT Administrator can:

- Security Administration
- Manage Users
- View User Lists
- Manage Roles and Permissions
- Assign User Role
- Access Audit Logs

Owner can:

- Request/Generate Report
- View Summary Reports (Read-Only)

Relationships:

<<include>>

- Create Shifts includes Generate Opening Checklist
- Assign Roles & Tasks includes Notify Staff
- Update Inventory includes Check Low Stock
- Review Performance includes View Task Reports
- Close Shift includes Generate Shift Summary
- Security Administration includes Manage Roles and Permissions
- Security Administration includes Access Audit Logs
- Manage Users includes Assign User Role
- Request/Generate Report includes View Summary Reports (Read-Only)

<<extend>>

- Update Inventory extends Create Purchase Request (only when restocking is needed)

Key Point

This diagram shows that FlowSync helps the store run daily work in a clear way. Managers control shifts, tasks, and inventory. Staff can see what to do and update progress. IT Admin manages system security. The owner can view reports only. This reduces confusion and helps the store work smoothly.

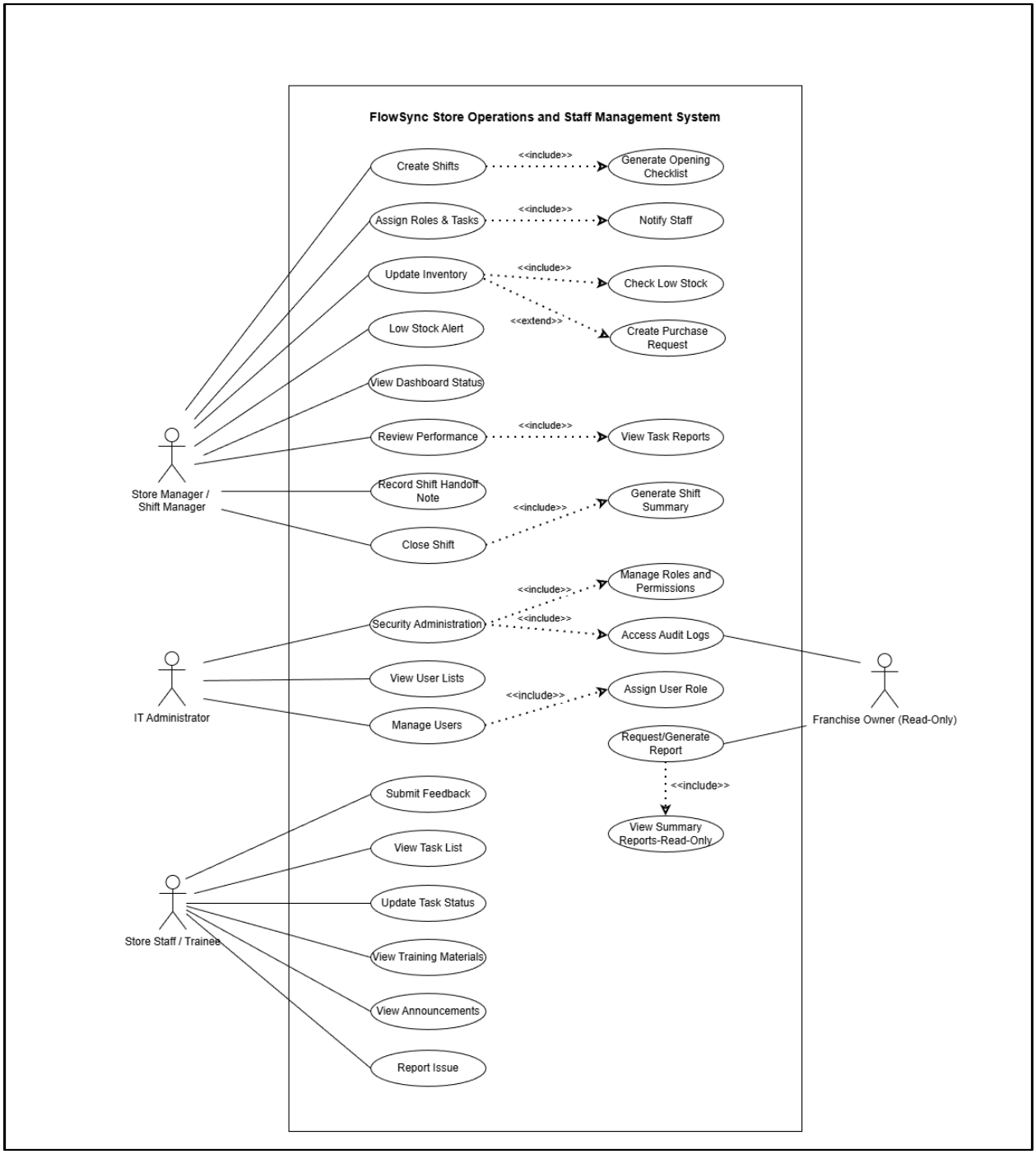

Figure 2: Use case Diagram

4.3 Use Case Narratives

UC1 - Create Shifts

Description:

This use case allows the Store Manager/Shift Manager to create a new shift in FlowSync by entering the shift date and time. The system stores the shift information as a shift record and makes it available for checklist generation, task assignment, and progress tracking during that shift.

Trigger:

The Store Manager/Shift Manager selects Create Shifts.

Pre-conditions:

- The Store Manager/Shift Manager is authenticated (logged in).
- The Store Manager/Shift Manager has permission to create shifts.

Post-conditions:

- A new shift record is created and saved in the system database.
- The shift becomes active and available for assigning roles and tasks.
- The system is ready to run checklist generation for the shift.

Stakeholders and Actors:

- Primary Actor: Store Manager / Shift Manager
- Supporting Actor: FlowSync System
- Stakeholders: Store management team, store staff, franchise owner

Dialog (Actor-System interaction):

1. Actor: Opens the shift management screen.
2. System: Displays the list of shifts and the option to create a new shift.
3. Actor: Clicks Create Shifts.
4. System: Prompts for shift details (date, start time, end time).
5. Actor: Enters the shift details and confirms.
6. System: Validates the inputs (format, time logic, conflicts).
7. System: Creates and saves the shift record and marks it active.
8. System: Confirms that the shift has been created successfully.

Exceptions:

- **E1:** If shift details are missing or invalid, the system displays an error and requests correct input.
- **E2:** If the new shift overlaps with an existing shift, the system blocks saving and asks the actor to adjust the schedule.
- **E3:** If the actor lacks permission, the system denies access to shift creation.

UC2 - Generate Opening Checklist

Description:

This use case generates the opening checklist for an active shift using checklist templates stored in the system. The system creates checklist task records and links them to the shift so they can be assigned, completed, and tracked during operations.

Trigger:

Opening time is reached, or checklist generation is executed as an included step after Create Shifts.

Preconditions:

- A shift exists and is active in FlowSync.
- An opening checklist template exists in the system.

Postconditions:

- The opening checklist tasks are generated and saved as task records.
- The tasks are linked to the correct shift and become available for assignment and tracking.

Stakeholders and Actors:

- Primary Actor: FlowSync System
- Supporting Actors: Store Manager/Shift Manager, Store Staff/Trainee
- Stakeholders: Store management, staff, owner

Dialog (Actor-System interaction):

1. System: Detects an active shift and the opening checklist requirement.
2. System: Loads the stored opening checklist template.
3. System: Creates checklist tasks for the shift based on the template.
4. System: Saves the tasks and links them to the active shift.
5. System: Makes the checklist available in the shift workspace for staff tracking.

Exceptions:

- E1: If no opening checklist template is found, the system shows an error and notifies the manager.
- E2: If no active shift exists, the system does not generate tasks and logs the event for review.

UC3 - Assign Roles & Tasks

Description:

This use case allows the Store Manager/Shift Manager to assign roles and tasks to staff for an active shift. FlowSync stores the role assignments and task assignments, then updates staff task lists and the manager dashboard using the saved shift data.

Trigger:

The Store Manager/Shift Manager selects Assign Roles & Tasks for an active shift.

Preconditions:

- The Store Manager/Shift Manager is logged in and authorized.
- An active shift exists.
- Staff user accounts exist in the system roster.

Postconditions:

- Roles and tasks are saved and linked to both the staff member and the shift.
- Assigned tasks appear in the staff member's task list.
- The system executes Notify Staff to send task details.

Stakeholders and Actors:

- Primary Actor: Store Manager / Shift Manager
- Supporting Actors: Store Staff/Trainee, FlowSync System
- Stakeholders: Managers, staff, owner

Dialog (Actor-System interaction):

1. Actor: Opens an active shift and selects Assign Roles & Tasks.
2. System: Displays the roster and available roles/task templates.
3. Actor: Selects a staff member and assigns a role (e.g., cashier, grill, cleaning).
4. System: Displays checklist-based tasks and optional custom tasks.
5. Actor: Selects tasks and confirms assignment.
6. System: Validates the assignment (e.g., checks for missing critical roles).

7. System: Saves the role/task records linked to the shift.
8. System: Updates staff task lists and the dashboard status.
9. System: Includes Notify Staff and sends the task details to the assigned staff.

Exceptions:

- **E1:** If the staff member is unavailable, the system blocks the assignment and requests a different selection.
- **E2:** If a critical role is missing, the system warns the manager before saving.
- **E3:** If the actor lacks permission, the system denies access.

UC4 - Update Task Status

Description:

This use case allows Store Staff/Trainee to update the completion status of assigned tasks during a shift. FlowSync records the updated status, the user who completed the task, and the completion time, and it updates checklist progress and dashboard information based on stored task data.

Trigger:

Store Staff/Trainee selects a task and performs Update Task Status.

Preconditions:

- The staff member is logged in.
- The staff member has assigned tasks for the active shift.

Postconditions:

- The task status is updated and saved with user ID and timestamp.
- Checklist progress and dashboard status reflect the new completion update.

Stakeholders and Actors:

- Primary Actor: Store Staff / Trainee
- Supporting Actors: FlowSync System, Store Manager/Shift Manager
- Stakeholders: Store staff, management, owner

Dialog (Actor-System interaction):

1. Actor: Opens View Task List.
2. System: Displays tasks assigned to the actor for the shift.
3. Actor: Selects a task and clicks Update Task Status (e.g., Complete).

4. System: Verifies that the task is assigned to the actor and is not locked.
5. System: Saves the new task status with completion time and user ID.
6. System: Updates checklist progress and dashboard metrics.
7. System: Confirms the update to the actor.
8. System: If the actor indicates a problem, the process may extend to Report Issue.

Exceptions:

- **E1:** If the task is already completed, the system shows "Already completed" and prevents duplicate updates.
- **E2:** If the actor is not assigned to the task, the system denies the update.
- **E3:** If the system cannot save the update, it shows an error and logs the failure.

UC5 - Update Inventory

Description:

This use case allows a Store Manager/Shift Manager or authorized user to update inventory item quantities in FlowSync. The system stores inventory updates with timestamps and user details, checks quantities against reorder levels, and generates low-stock alerts when required.

Trigger:

The user submits a new quantity through Update Inventory.

Preconditions:

- The user is logged in.
- The user has permission to update inventory.
- Inventory items exist in the system.

Postconditions:

- Inventory quantity updates are saved with user and timestamp.
- Check Low Stock is executed, and Low Stock Alert may be created if thresholds are crossed.

Stakeholders and Actors:

- Primary Actor: Store Manager / Shift Manager
- Supporting Actor: FlowSync System
- Stakeholders: Store management, staff, owner

Dialog (Actor-System interaction):

1. Actor: Opens the inventory page and selects Update Inventory.
2. System: Displays inventory items with current quantity and reorder level.
3. Actor: Selects an item and enters the updated quantity.
4. System: Validates the input value (numeric, non-negative).
5. System: Saves the inventory update with timestamp and user details.
6. System: Includes Check Low Stock and compares quantity to reorder level.
7. System: If stock is low, it generates a Low Stock Alert for management.
8. System: Confirms that the inventory update has been saved successfully.
9. System: If restocking is needed, the flow may extend to Create Purchase Request.

Exceptions:

- **E1:** If the quantity is invalid (negative or non-numeric), the system rejects it and asks for correction.
- **E2:** If the item does not exist, the system shows an error and returns to the inventory list.
- **E3:** If the actor is unauthorized, the system denies access.

UC6 - Close Shift

Description:

This use case allows the Store Manager/Shift Manager to close an active shift in FlowSync. The system finalizes shift data by locking further updates and generates a shift summary report based on task completion, issues, and recorded updates.

Trigger:

The Store Manager/Shift Manager selects Close Shift, or the shift end time is reached.

Preconditions:

- A shift exists and is active.
- The manager is logged in and authorized to close shifts.

Postconditions:

- The shift status becomes closed and updates are restricted.
- The system includes Generate Shift Summary and stores the summary report.

Stakeholders and Actors:

- Primary Actor: Store Manager / Shift Manager
- Supporting Actors: FlowSync System, Store Staff/Trainee
- Stakeholders: Store management, staff, owner

Dialog (Actor-System interaction):

1. Actor: Opens the active shift and selects Close Shift.
2. System: Displays a confirmation message and warnings for incomplete critical tasks (if applicable).
3. Actor: Confirms shift closure.
4. System: Marks the shift status as closed and locks editing.
5. System: Includes Generate Shift Summary and creates the summary report.
6. System: Stores the report for later review and reporting.
7. System: Confirms that the shift has been closed successfully.

Exceptions:

- **E1:** If no active shift exists, the system prompts the actor to create or select a shift.
- **E2:** If the user is unauthorized, the system denies the action.
- **E3:** If report generation fails, the system logs the error and notifies the manager.

5 Domain Class Diagram

The Domain Class Diagram shows the main objects (classes) in FlowSync and how they connect. It helps explain how FlowSync stores data for users, shifts, tasks, inventory, training, alerts, reports, and logs. This makes store operations more organized and easier to track.

Core Classes

User Attributes:

Attributes: user_id, name, email_address, password_hash, status
The User class represents anyone who can log in to FlowSync. It stores the user's identity and login details. The password_hash is used for security, so the password is not stored as plain text. The status field helps control whether the account is active or disabled.

StoreManager Attributes:

Attributes: manager_level
StoreManager is a type of User who manages store operations. A manager can create shifts, assign roles and tasks, update inventory, view dashboard status, review performance, write handoff notes, and close shifts. The manager_level shows the manager's authority level.

StoreStaff Attributes:

Attributes: role_type
StoreStaff is a type of User who performs shift work. Staff members view their task list, update task status, report issues, view announcements, access training materials, and submit feedback. The role_type shows the staff category or job type (example: cashier, kitchen, cleaning).

ITAdministrator Attributes:

Attributes: admin_id, accessLevel, last_Login_At
ITAdministrator is a type of User who manages system security. This user manages users, roles, and permissions, and checks audit logs. The accessLevel shows what admin actions are allowed, and last_Login_At helps track admin activity.

Owner Attributes:

Attributes: ownerType, viewScope
Owner is a type of User who mainly views reports and summaries. This role is usually read-only. The viewScope limits what the owner can view (example: one store or many stores).

Shift Attributes:

Attributes: shiftId, date, startTime, endTime, status
A Shift represents one working time period in the store. It is the main object that links daily operations such as tasks, checklists, role assignments, and handoff notes. The status shows if the shift is active or closed.

Checklist Attributes:

Attributes: checklistId, type (open/mid/close), created_At
A Checklist represents a list of standard tasks for a shift type. For example, opening checklist, mid-shift checklist, and closing checklist. It helps the store follow the same steps every day.

Task Attributes:

Attributes: taskId, title, description, taskType (open/mid/close/custom), status, dueTime, completedAt
A Task represents a single work item that must be done during a shift. It stores task details and completion tracking. The system uses status, dueTime, and completedAt to measure progress and performance.

RoleAssignment Attributes:

Attributes: assignmentId, roleName, assignedAt
A RoleAssignment records which role is given to a staff member during a shift. Examples of roles are cashier, grill, cleaning, or stock handling. The assignedAt field stores when the assignment was created.

Schedule Attributes:

Attributes: scheduleId, weekStartDate
A Schedule represents weekly planning. It can group many shifts within a week and helps organize staffing and store planning.

InventoryItem Attributes:

Attributes: item_id, name, unit, currentQty, reorderLevel, lastUpdatedAt
An InventoryItem represents a stock item used in the store (example: buns, cups, sauces). It stores the current quantity and the reorder level. This supports low-stock checking.

InventoryUpdate Attributes:

Attributes: updateId, qty, updatedAt
An InventoryUpdate stores each inventory change. It records the quantity update and the time of update. This creates a history of inventory actions.

Alert Attributes:

Attributes: alertId, alertType (lowStock/overdue), message, createdAt, resolved
An Alert represents system warnings such as low stock or overdue tasks. It helps managers react quickly. The resolved field shows if the alert is still active or already handled.

TrainingMaterial Attributes:

Attributes: trainingId, title, content, roleTarget, lastUpdatedAt
TrainingMaterial represents training and reference content for staff. It supports onboarding and helps staff follow correct procedures. The roleTarget makes training easier to filter by role.

HandOffNote Attributes:

Attributes: note_id, content, createdAt
A HandOffNote stores important notes at the end of a shift. It helps the next shift understand issues, shortages, or unfinished tasks.

Feedback Attributes:

Attributes: feedbackId, message, dateSubmitted
Feedback stores staff feedback and suggestions. It supports continuous improvement and communication.

Report Attributes:

Attributes: report_id, reportType, generateDate
A Report stores summary outputs for owners and managers. It supports oversight and performance tracking.

AuditLog Attributes:

Attributes: logId, timestamp, action
An AuditLog stores important system actions for security and accountability. It supports monitoring of activities like role changes and admin actions.

Relationships

- User → StoreManager / StoreStaff / ITAdministrator / Owner (Inheritance)

  User is the parent class. These roles inherit from User.
- Shift → Checklist (1 → 0.*)

  One shift can have many checklists.
- Checklist → Task (1 → 0.*)

  One checklist can have many tasks.
- Shift → RoleAssignment (1 → 1.*)

  One shift has role assignments.
- Shift → HandOffNote (1 → 0.*)

  One shift can have many handoff notes.
- Schedule → Shift (1 → 0.*)

  One schedule can include many shifts.
- InventoryItem → InventoryUpdate (1 → 0.*)

  One inventory item can have many updates.
- InventoryItem → Alert (1 → 0.*)

  One inventory item can generate many alerts.
- StoreStaff → Feedback (1 → 0.*)

  One staff member can submit many feedback entries.
- StoreStaff ↔ TrainingMaterial (0. ↔ 0.)**

  Many staff can access many training materials.
- Owner → Report (1 → 0.*)

  One owner can view many reports.
- ITAdministrator → AuditLog (1 → 0.*)

  One admin can create or view many audit logs.

Aggregation and Composition

Composition (strong "part-of")

- Checklist - Task: tasks belong to a checklist.
- InventoryItem - InventoryUpdate: updates belong to an inventory item.
- Shift - RoleAssignment: role assignments belong to a shift.
- Shift - HandOffNote: handoff notes belong to a shift.

Aggregation (weak "has-a")

- Schedule - Shift: schedule groups shifts, but shifts can still exist as records.
- StoreStaff - TrainingMaterial: training material is shared and exists independently.

Diagram Overview

- The diagram shows Shift as the center of daily operations.
- Checklist and Task support daily routines and task tracking.
- InventoryItem, InventoryUpdate, and Alert support inventory control and low stock warnings.
- TrainingMaterial, HandOffNote, and Feedback support learning and communication.
- Report supports management and owner visibility.
- AuditLog supports security and accountability.
- Inheritance from User supports role-based access and responsibilities.

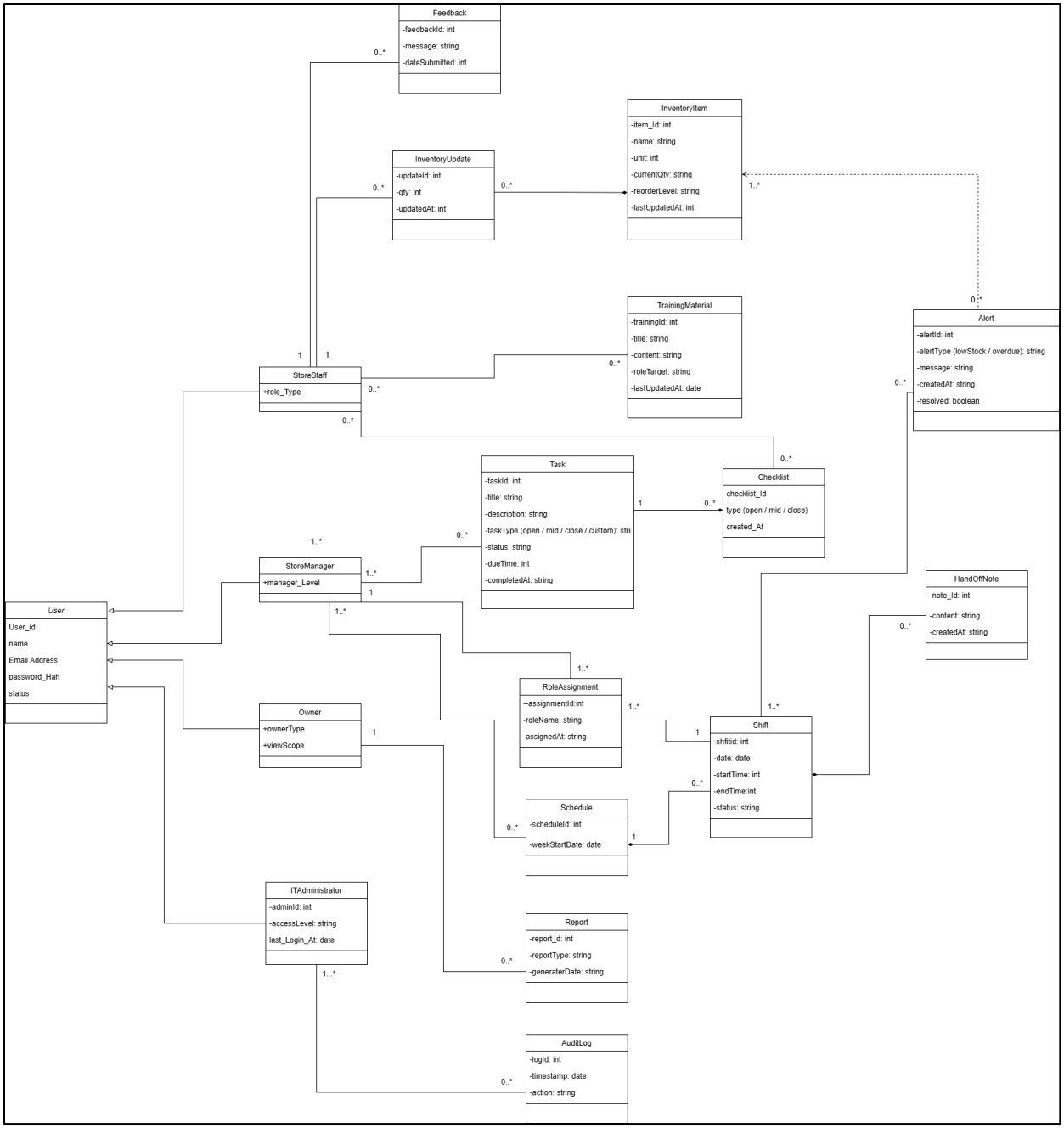

Figure 3: UML Class diagram

6 Nonfunctional Requirements

6.1 Constraints

- FlowSync is a web application. It must be finished within the course timeline and meet all milestone deadlines.
- It is built by a small team, so the system must stay within the project scope and focus on main features first.
- The web application is for internal store use, not for public users. So it must support simple store workflows.
- The system must work well during busy hours. So the pages must be simple and fast to use.
- The system must support different user roles (manager, staff, IT admin, owner). Each role must only access what they are allowed to access.
- The system may run on basic devices (PC/tablet) and with slow internet. So it must not need high-end hardware.
- The system must save all data correctly (tasks, inventory updates, handoff notes) so important shift records are not lost.

6.2 Business Rules

- Only the Store Manager / Shift Manager can create shifts and close shifts.
- Only the Store Manager / Shift Manager can assign roles and tasks to staff.
- Store staff can update task status only for tasks assigned to them (unless a manager allows override).
- A shift cannot be closed without manager confirmation. If important tasks are not finished, the system must show a warning.
- Inventory quantity cannot be negative. The system must reject invalid numbers.
- Low stock must be detected when currentQty < reorderLevel. Then the system must create a Low Stock Alert.
- Every inventory update must save who updated it and when it was updated (timestamp).
- Handoff notes must be linked to the correct shift and must be visible to the next shift team.
- Owners can view summary reports in read-only mode and cannot change store data.
- IT Administrators can manage users, roles, permissions, and audit logs, but they should not do daily store operations unless needed.

6.3 Additional System Requirements

Security

- FlowSync must require secure login for all users.
- Passwords must be stored as hashed values (not plain text).
- The system must use Role-Based Access Control (RBAC), so users can only access what their role allows.
- Important actions (role changes, security changes, admin actions) must be recorded in audit logs.

Reliability

- The system must save data correctly for tasks, inventory updates, and handoff notes.
- The system should prevent wrong or duplicate updates (example: completing the same task twice).
- If an error happens (crash or network issue), the system must show an error message and must not lose saved data.

Availability

- FlowSync should be available during store business hours.
- Staff and managers should access tasks and checklists without long delays.

Performance

- Main pages like View Task List, Dashboard, and Inventory should load fast.
- Task updates and inventory updates should save in real time or near real time.

Usability

- The interface must be simple, clear, and easy to learn with little training.
- Staff should complete tasks with minimum clicks and clear instructions.

Integrity

- Data must stay accurate and consistent (example: inventory level must match the latest update).
- Normal users must not be able to change audit logs.
- Reports must show correct shift and task data.

Serviceability / Maintainability

- IT Admin should add/remove users and assign roles easily.
- Managers should update checklist templates, tasks, and inventory items without code changes (if the system supports it).

Scalability

- The system should support many shifts, many staff users, and many tasks per day without slowing down.
- In the future, the system should support multiple store locations (optional improvement).

7 Summary and Conclusion

This Milestone 3 document explains the requirements for the FlowSync Smart Store Operations & Staff Management System. FlowSync is made for a fast-food restaurant where staff often use paper checklists and verbal instructions. This can cause missed tasks, poor shift handoffs, and inventory shortages. FlowSync helps by using digital checklists, clear role and task assignments, task status tracking, inventory updates with low-stock alerts, training materials for staff, and shift handoff notes for better communication. This milestone also lists the main stakeholders, shows the system boundary using a context diagram, connects events to use cases using an event analysis table, and includes fully defined use case narratives for the most important functions. It also includes a domain class diagram to show the main system objects and their relationships. Finally, it lists non-functional requirements and business rules to guide security and system quality.

Overall, FlowSync is a useful solution for a busy restaurant. It helps the store work in a more organized way and reduces mistakes. It improves accountability because managers can see what is done and what is not done. It also helps prevent low stock issues and improves shift-to-shift communication. The requirements in this milestone give a clear base for the next steps, like system design and implementation. FlowSync can be developed to meet real store needs and still stay secure, easy to use, and ready for future improvements.