



SYSTEM USE-CASES

ACTOR

UML Use Case Description: Send Automated Alerts and Notifications

USE CASE #1:	SEND AUTOMATED ALERTS AND NOTIFICATIONS
Primary Actor:	System
Goal	The system automatically sends notifications to users (students, clinic staff, or SSD) based on events such as appointment reminders, survey prompts, or emergency updates.
Preconditions	<ul style="list-style-type: none"> User accounts and notification preferences are active.
Postconditions	<ul style="list-style-type: none"> Users receive appropriate and timely alerts.
Trigger	Detect event passed to System.
Main Flow	<ol style="list-style-type: none"> The System detects an event that requires a notification, such as: <ul style="list-style-type: none"> Upcoming appointment Appointment updates Health survey reminder SOS confirmation and status update The System generates the appropriate notification and sends it via: <ul style="list-style-type: none"> In-app alert University e-mail The user receives the notification and can act on it as needed.
Alternative Flows	<p>A1. Notification service unavailable: If the system cannot send notifications (in-app or email), it retries and logs the failure.</p> <p>A2. User opted out: If the user has disabled notifications, the alert is not sent but logged for tracking purposes.</p>

UML Use Case Description: Perform Data Backup and Recovery

USE CASE #1:	PERFORM DATA BACKUP AND RECOVERY
Primary Actor:	System
Goal	The system automatically performs periodic backups of critical health and appointment data to prevent data loss.
Preconditions	<ul style="list-style-type: none"> Backup schedule and storage configuration are active.
Postconditions	<ul style="list-style-type: none"> Data backups are successfully stored. Recovery option is available in case of system failure.
Trigger	Scheduled event every 8 pm from Monday to Saturday.
Main Flow	<ol style="list-style-type: none"> The System initiates a scheduled backup according to the configured interval. The System compresses the relevant data and stores it securely in the designated backup storage. The System generates a confirmation log indicating the success or failure of the backup.
Alternative Flows	<p>A1. Backup storage unavailable: If storage cannot be accessed, the system retries and alerts the administrator.</p> <p>A2. Data compression error: If compression fails, the system logs the error and continues with uncompressed backup if possible.</p>

UML Use Case Description: Auto-Award Gamification Stamps

USE CASE #1:	AUTO-AWARD GAMIFICATION STAMPS
Primary Actor:	System
Goal	The system automatically awards points or stamps to students who meet certain health participation milestones (daily surveys).
Preconditions	<ul style="list-style-type: none"> • Gamification rules are set by the administrator.
Postconditions	<ul style="list-style-type: none"> • Student profiles are updated with new rewards or progress points.
Trigger	Daily scheduled gamification checking per batch.
Main Flow	<ol style="list-style-type: none"> 1. The System checks the student's activity logs for daily health survey submissions 2. The System validates each action against the configured reward criteria <ul style="list-style-type: none"> • (HP (Health Points) per check-in • Milestone achievements 3. The System automatically awards the corresponding points or HP stamps and records them in the student's gamification profile. 4. The System updates the student's dashboard and reward progress in real-time.
Alternative Flows	<p>A1. Activity not logged: If the student activity is missing or corrupted, no points/stamps are awarded, and an error is logged.</p> <p>A2. Database save failure: If points/stamps cannot be saved, retry occurs; if still failing, alert is sent to admin.</p>

UML Use Case Description: Generate and Update Dashboard Analytics

USE CASE #1:	GENERATE AND UPDATE DASHBOARD ANALYTICS
Primary Actor:	System
Goal	To continuously aggregate operational, emergency, and health data to provide real-time updates to the dashboard widgets (KPIs, charts, and lists) for users.
Preconditions	<ul style="list-style-type: none"> Transactional data (Health Surveys, Appointments, Inventory logs, SOS alerts) exist in the database. External Weather API connection is active. User dashboards are active and capable of receiving updates.
Postconditions	<ul style="list-style-type: none"> Clinic/Doctor Dashboards: Displays updated symptom trends, weather advisories, supply recommendations, and appointment queues. SSD Dashboard: Displays real-time active SOS case maps and status. Admin Dashboard: Displays system activity logs and user statistics. Student Dashboard: Displays updated check-in progress.
Trigger	The System detects a new data submission (e.g., Student submits daily survey, Nurse updates inventory, SSD closes a case, or Weather API updates).
Main Flow	<ol style="list-style-type: none"> 1. The System aggregates relevant data from various records: <ul style="list-style-type: none"> Health Surveys: Anonymized symptom ratings (1-5), reported symptoms, and timestamps. Operations: Appointment statuses (Queued, Completed, Cancelled, No-Show) and check-ins. Inventory: Stock levels, expiration dates, and item categories. Emergencies: Real-time SOS alerts, GPS locations, and incident statuses. Gamification: Daily check-in streaks and point accumulations. Environmental: Real-time weather data fetched via the Weather API. 2. The System processes the data to recalculate specific dashboard elements: <ul style="list-style-type: none"> Health & Supply Advisories: Correlates weather data with symptom spikes to generate "Today's Health Advisory" and "Supply Advisory" (recommending specific medicines based on trends). Inventory KPIs: Updates counters for "Low Stock," "Out of Stock," and "Expiring Soon" items. Appointment Queue: Refreshes the "Today's Appointments" list and flags "Attention Needed" items. Emergency Status: Updates the map view and lists active cases for SSD. 3. The System refreshes the visual components on the respective user dashboards: <ul style="list-style-type: none"> Clinic/Doctor: Updates the "Health Survey Dashboard" graphs (Symptom prevalence, Comparisons) Student: Updates the "My Progress" stamp visualization Admin: Updates "Total User Count" and "Recent Activity Logs" SSD: Updates main Dashboard 4. The System stores the aggregated analytics to allow for historical reporting (e.g., "Past 30 Days" filter views).

UML Use Case Description: Generate and Update Dashboard Analytics

Alternative Flows	<p>A1. Weather Data Unavailable: If the Weather API fails, the System displays "Weather data unavailable" on the Health Survey Dashboard and generates advisories based on survey data only..</p> <p>A2. No Data Available: If a specific filter (e.g., "Past 5 Days") returns no records, the System displays "No data available for the selected period".</p> <p>A3. Processing Latency: If data aggregation takes longer than expected, the System displays a loading state before rendering the updated widgets.</p>
-------------------	---

UML Use Case Description: Detect Potential Outbreaks

USE CASE #1:	DETECT POTENTIAL OUTBREAKS
Primary Actor:	System
Goal	The system monitors aggregated health data to detect unusual symptom patterns that may indicate a potential outbreak, including those influenced or triggered by weather conditions.
Preconditions	<ul style="list-style-type: none"> • Active collection of daily health surveys and clinic records. • Weather API data is available and updated. • Thresholds for outbreak detection (symptom spikes, weather-linked triggers) are configured.
Postconditions	<ul style="list-style-type: none"> • An alert is sent to clinic and admin users if a potential outbreak is detected.
Trigger	<ul style="list-style-type: none"> • System detects symptom spikes or unusual health patterns. • System detects weather conditions associated with increased illness risk.
Main Flow	<ol style="list-style-type: none"> 1. The System retrieves updated entries from daily health surveys and clinic consultations trends. 2. The System pulls current and recent weather data and weather changes within the week. 3. The System correlates health data with weather conditions, analyzing patterns such as: <ul style="list-style-type: none"> ▪ increased cough/colds during rainy days ▪ fever clusters during heat waves 4. If thresholds are exceeded, the System: <ul style="list-style-type: none"> ▪ Flags a potential outbreak ▪ categorizes the type (weather-linked or general health spike) 5. The System sends alerts to clinic staff and doctor users with: <ul style="list-style-type: none"> ▪ detected symptoms involved ▪ potential weather connection ▪ suggested next steps (e.g., prepare necessary medical supplies) 6. The System logs the event for reporting and outbreak tracking.
Alternative Flows	<p>A1. Insufficient data. If survey data is too sparse for meaningful trend analysis, no alert is generated.</p> <p>A2. Weather feed unavailable. System continues symptom-based detection only, logs weather API failure.</p> <p>A3. Notification delivery failure: If alert cannot be sent to personnel, the system retries and logs the failure for admin review.</p>

UML Use Case Description: Log All Major User Activities

USE CASE #1:	LOG ALL MAJOR USER ACTIVITIES
Primary Actor:	System
Goal	The System automatically records significant actions such as logins, data updates, submissions, and role/permission changes for accountability and security purposes.
Preconditions	<ul style="list-style-type: none"> Logging module is enabled.
Postconditions	<ul style="list-style-type: none"> All activities are stored in the system audit log.
Trigger	Event-driven (Users interacts with system and logs new activities)
Main Flow	<ol style="list-style-type: none"> The System detects a major event, such as: <ul style="list-style-type: none"> User login or logout Submission of health surveys, appointments, or emergency reports Creation, modification, or deletion of clinic records Updates made by doctors (diagnosis, prescriptions, notes) Updates made by clinic staff (inventory edits, appointment adjustments) Administrative actions such as: <ul style="list-style-type: none"> Updating user roles Modifying system settings Changing access rules or audit configurations The System records the activity details, including: <ul style="list-style-type: none"> User ID and role Timestamp Type of action performed Affected module or record The System stores the activity in the secure audit log. The activity becomes accessible through the admin's audit and monitoring interface.
Alternative Flows	A1. Logging service unavailable: If the log service fails, the system temporarily queues events until logging is restored.

ADMIN USE-CASES

ACTOR

UML Use Case Description: Manage Use Accounts

USE CASE #1:	MANAGE USER ACCOUNTS
Primary Actor:	Administrator
Goal	To create, modify, or deactivate user accounts for students, clinic staff, SSD, and doctors.
Preconditions	<ul style="list-style-type: none"> Administrator is logged into the system. System access is granted to admin-level users. A user management module is available.
Postconditions	<ul style="list-style-type: none"> User accounts are successfully created, updated, or deactivated. Account changes are logged in the audit trail.
Trigger	Administrator selects “User Management” from the admin panel.
Main Flow	<ol style="list-style-type: none"> The system displays a list of all registered user accounts. The Administrator selects “Add,” “Edit,” or “Deactivate” for a specific user. The Administrator enters or updates the required user details including: <ul style="list-style-type: none"> First and Last Name Role University email Account status [Active Deactivated] The system validates the inputs and saves the changes. A confirmation message appears, and the update is recorded in the system’s audit trail for accountability.
Alternative Flows	A1. Invalid Input: Display “Missing or invalid user details.” A2. Action Failed: Display “Unable to update user account. Please retry.” A3. Unauthorized Access: Display “You do not have permission to modify these settings.”

UML Use Case Description: Configure System Settings (Security, Access Rules, Audit Trails)

USE CASE #2:	CONFIGURE SYSTEM SETTINGS (SECURITY, ACCESS RULES, AUDIT TRAILS)
Primary Actor:	Administrator
Goal	To configure system-level settings including access roles and security parameters.
Preconditions	<ul style="list-style-type: none"> Administrator is logged into the system. A configuration module is available. Admin has system privileges.
Postconditions	<ul style="list-style-type: none"> System configurations are updated and saved. New access and security settings take effect immediately.
Trigger	Administrator selects “System Configuration” from the admin panel.
Main Flow	<ol style="list-style-type: none"> The system displays configurable options such as security settings, access permissions, and audit parameters. The Administrator updates the necessary configurations: <ul style="list-style-type: none"> Role permissions Password policies The system validates all modified settings. The Administrator reviews and confirms the changes. The system applies the new configurations and logs the update in the audit trail.
Alternative Flows	A1. Unauthorized Access: Display “You do not have permission to modify these settings.” A2. Configuration Error: Display “Failed to apply settings. Please try again.”

UML Use Case Description: Set and Update Gamification Criteria

USE CASE #3:	SET AND UPDATE GAMIFICATION CRITERIA
Primary Actor:	Administrator
Goal	To set, modify, or update gamification rules, such as HP stamp points, event rewards, and participation thresholds.
Preconditions	<ul style="list-style-type: none"> Administrator is logged into the system. A Gamification module is accessible. Existing reward configurations are available.
Postconditions	<ul style="list-style-type: none"> Gamification rules are updated in the system. Changes are reflected in student app interfaces and dashboards.
Trigger	Administrator selects “Gamification Settings” from the admin panel.
Main Flow	<ol style="list-style-type: none"> The Administrator accesses “Gamification Settings.” The system displays the current HP stamp, badge, and reward configurations. The Administrator updates the desired criteria, such as: <ul style="list-style-type: none"> Points awarded per daily check-in Weekly event or milestone rewards The system validates all modified inputs and saves the updated settings. The students’ app interfaces automatically reflect the new gamification criteria.
Alternative Flows	A1. Invalid Configuration: Display “Invalid reward or point value.” A2. Save Failed: Display “Unable to update gamification settings.”

UML Use Case Description: Access Admin Dashboard

USE CASE #4:	ACCESS ADMIN DASHBOARD
Primary Actor:	Administrator
Goal	To view overall system analytics, including user activity logs, health statistics, and audit trails.
Preconditions	<ul style="list-style-type: none"> Administrator is logged into the system. Data sources (users, health surveys, activity logs) are available.
Postconditions	<ul style="list-style-type: none"> Dashboard metrics are displayed. Admin may export or review data insights.
Trigger	Administrator selects “Admin Dashboard” from the main menu.
Main Flow	<ol style="list-style-type: none"> The system loads the Admin Dashboard. The dashboard displays key data, including: <ul style="list-style-type: none"> Total user count System uptime Recent activity logs Active modules and components The Administrator filters or refreshes the displayed reports as needed. The system updates the dashboard visuals in real-time. The Administrator may export analytics or logs for record keeping or auditing.
Alternative Flows	A1. No Data Available: Display “No data to display at this time.” A2. Dashboard Error: Display “Failed to load dashboard data.”

STUDENT USE-CASES

ACTOR

UML Use Case Description: Book and View Clinic/Dental Appointments

USE CASE #1:	BOOK AND VIEW CLINIC/DENTAL APPOINTMENTS
Primary Actor:	USJ-R Tertiary Students
Goal	To schedule and view clinic or dental appointments conveniently through the JoseniCare app.
Preconditions	<ul style="list-style-type: none"> • Student is logged into the app. • Clinic system is online and has available appointment slots.
Postconditions	<ul style="list-style-type: none"> • Appointment is successfully booked and recorded in the system. • Confirmation and details are displayed to the student.
Trigger	Student selects the “Book” option in the application under the Schedule Appointment section.
Main Flow	<ol style="list-style-type: none"> 1.Student opens the JoseniCare app. 2.Student selects the option to “Book” an appointment. 3.System asks which campus the student is located [BASAK MAIN QUADRICENTENIAL]. 4.System asks if the student is booking a dental session or general check-up → [DENTAL CHECK-UP]. <ol style="list-style-type: none"> a.If Dental, System will ask follow up questions such as: for [CLEANING DENTAL FILLING (pasta)] b.If Check-up, System will ask: for [Event (Sportsfest, Intramurals) General] 5.After student answers the follow-up questions, System will show available and taken slots through visual calendar. 6.Student will select available slot; Specify [Date & time] and enter confirmation. 7.System confirms availability and saves the appointment if true. 8.Student views confirmation details showing date, time, and appointment purpose.
Alternative Flows	A1. No Available Slot: System will inform the student and suggest the nearest available schedule instead. A2. Connection Error: Alert “Unable to process booking. Please try again later.”

UML Use Case Description: Submit a Daily Health Survey

USE CASE #2:	SUBMIT A DAILY HEALTH SURVEY
Primary Actor:	USJ-R Tertiary Students
Goal	To submit daily self-health assessments for wellness tracking and potential outbreak detection.
Preconditions	<ul style="list-style-type: none"> • Student is logged into the app. • Clinic system is online and survey form is open for the current date.
Postconditions	<ul style="list-style-type: none"> • Survey data is recorded in the system. • Health status is updated in the Health Dashboard.
Trigger	Student selects the “Check-in” option in the application under the Daily Health Survey section.
Main Flow	<ol style="list-style-type: none"> 1. Student opens the JoseniCare app. 2. Student selects the option to “Check-in” their Daily Health Survey. 3. The system displays the Terms and Conditions before accessing the submission form. 4. The system prompts the student for a subjective physical assessment, asking them to rate their general health from 1 to 5 (Healthy → Sick). <ol style="list-style-type: none"> i. If the rating is between 3-5, System will ask for temperature, symptoms, severity and duration. ii. If the rating is between 1-2, the system provides a positive message acknowledging the student’s good health and asks them to confirm the absence of common symptoms. 5. After completing the questions, student confirms and submits their survey. 6. System will validate report and then save to database if true. 7. The student’s check-in progress is automatically updated and visually reflected (e.g., via progress stamp or indicator)
Alternative Flows	<p>A1. Failed Confirmation for Terms and Conditions: Student will be asked to agree to answer the survey form.</p> <p>A2. Submission Error: Display “Failed to submit form. Please try again.”</p> <p>A3. Report Mismatch: Display “It seems there’s a small inconsistency. You rated your health as good but also reported some symptoms. Please review your rating to ensure it reflects how you feel.”</p> <p>A4. Duplicate Submission: Display “You’ve already completed today’s survey.”</p>

UML Use Case Description: View Digital Health Record

USE CASE #3:	VIEW DIGITAL HEALTH RECORD
Primary Actor:	USJ-R Tertiary Students
Goal	To access personal medical and dental records digitally.
Preconditions	<ul style="list-style-type: none"> • Student is logged into the app. • Students' digital health record exists in the system. • Has set their biometric login detail. Either [Fingerprint Facial] recognition.
Postconditions	<ul style="list-style-type: none"> • Health data are displayed securely
Trigger	Student selects the “Health Record” option under the Personal Records section.
Main Flow	<ol style="list-style-type: none"> 1.The student opens the JoseniCare app and navigates to the “Personal Records” page. 2.The system prompts the student to complete biometric login for verification. 3. Once verification is conThe student selects the option to “Health Record”. 4.The system retrieves the student’s stored medical and dental records from the database. 5.The system displays the student’s patient information as recorded by the clinic, including: <ul style="list-style-type: none"> • Name • School ID • Department • Course & Year • Noted Allergies • History of previous illnesses or accidents 6.The system summarizes and visualizes health data sourced from the student’s submitted Health Surveys and clinic appointments, such as: <ul style="list-style-type: none"> • Frequency of specific symptoms (e.g., how many times cough, colds, fever occurred) • Health trends over time (e.g., improved/stable/worsening) • Prescribed medicines 7.The student can interact with the visual summary (e.g., view charts or detailed breakdowns for each health category).
Alternative Flows	<p>A1. Failed Biometric login: Prompt retry or alternative login.</p> <p>A2. No Health Record found: Display “Sorry, you have no digital health record yet. Please inquire the Clinic READS for assistance.”</p> <p>A3. Data retrieval error: System displays “Unable to load records. Please try again later.</p>

UML Use Case Description: View Appointment History

USE CASE #4:	VIEW APPOINTMENT HISTORY
Primary Actor:	USJ-R Tertiary Students
Goal	To view the list and details of past clinic and dental appointments.
Preconditions	<ul style="list-style-type: none"> • Student is logged into the app. • Students' appointment history record exists in the system.
Postconditions	<ul style="list-style-type: none"> • Appointment history is displayed securely
Trigger	Student selects the “Appointment History” option under the Personal Records section.
Main Flow	<ol style="list-style-type: none"> 1.The student opens the JoseniCare app and navigates to the “Personal Records” page. 2.The system prompts the student to complete biometric login for verification. 3.The student selects the button option “Appointment History”. 4.The system retrieves the student’s stored medical and dental appointment history information from the database, including: <ul style="list-style-type: none"> • Date & time • Purpose for appointment • Doctor in Charge • Campus Clinic selected 5.The system displays the list of past appointments in chronological order (most recent first). 6.The student may select a specific appointment to view consultation notes or prescriptions issued by the doctor.
Alternative Flows	<p>A1. Failed Biometric login: Prompt retry or alternative login.</p> <p>A2. No Appointment History found: Display “You have no recorded clinic appointments yet.”</p> <p>A3. Data retrieval error: System displays “Unable to load appointment history. Please try again later.”</p>

UML Use Case Description: Report Accidents through SSD Emergency Channel

USE CASE #5:	REPORT ACCIDENTS THROUGH SSD EMERGENCY CHANNEL
Primary Actor:	USJ-R Tertiary Students
Goal	To report accidents or emergencies directly to the Safety and Security Department (SSD).
Preconditions	<ul style="list-style-type: none"> • Student is logged into the app. • Network connection is active.
Postconditions	<ul style="list-style-type: none"> • Emergency report is sent to the SSD channel. • SSD receives the report and can respond.
Trigger	The student enters Emergency Mode and holds the “REPORT” button for 3 seconds.
Main Flow	<ol style="list-style-type: none"> 1.The student swipes from the bottom-right corner of the screen. 2.The SOS button appears with the prompt: “Hold for 3 seconds to send alert” and a Cancel option. 3.The student presses and holds the SOS button continuously for 3 seconds. 4.The System detects the sustained press and prompts the student to select their current floor level (e.g., Ground Floor, 2F, 3F). 5.The System automatically: <ul style="list-style-type: none"> • Captures the student’s current GPS location, • Records the selected floor level, • Generates an emergency report, and • Sends an instant alert to the SSD dashboard. 6.The SSD receives the alert containing the student’s name, floor level, location, and timestamp. 7.The System confirms to the student that the SOS has been successfully sent.
Alternative Flows	<p>A1. Premature Release: If the student releases the SOS button before 3 seconds, the alert is canceled and no report is sent.</p> <p>A2. Location Service Disabled: System prompts: “Enable location services to send emergency alert.”</p> <p>A3. Connection Error: System retries sending the alert or queues it until a stable connection is available.</p> <p>A4. False Trigger: If the student taps “Cancel” before the 3-second hold is completed, the alert is aborted and the screen returns to normal.”</p>

UML Use Case Description: Interact with “Jose” Chatbot

USE CASE #6:	INTERACT WITH “JOSE” CHATBOT
Primary Actor:	USJ-R Tertiary Students
Goal	To get quick responses to health-related inquiries, clinic doctor availability status, and app guidance through the AI chatbot “Jose.”
Preconditions	<ul style="list-style-type: none"> • Student is logged into the app. • Chatbot system is active.
Postconditions	<ul style="list-style-type: none"> • Student receives helpful information or is directed to proper in-app services.
Trigger	Student opens and messages the JoseniCare chatbot “Jose”
Main Flow	<ol style="list-style-type: none"> 1. The student clicks the JoseniCare chatbot “Jose” icon in the home page. 2. The student types a question or request in the text box (e.g., “How do I book an appointment?” or “What are today’s clinic hours?”). 3. The system’s Chatbot processes the input using Natural Language Processing (NLP) to determine intent and context. 4. The Chatbot displays a relevant response or provides links and resources, such as: <ul style="list-style-type: none"> • Health tips or self-care reminders • Appointment details or scheduling assistance • Doctor availability and clinic hours • Guidance on how to use JoseniCare features 5. The conversation continues or ends depending on the student’s follow-up questions or satisfaction with the response.
Alternative Flows	<p>A1. Unrecognized Input: Chatbot displays “I didn’t catch that, please rephrase for me.”</p> <p>A2. System Unavailable: Display “Jose is taking a break. Please come back later :D”</p> <p>A3. Message limit reached: Display “Limit reached. Please start new chat.”</p> <p>A4. Chatbot limitation: If Chatbot can’t resolve inquiry, display “Please proceed to the nearest Clinic for further assistance.”</p>

UML Use Case Description: Track Check-In Progress

USE CASE #7:	TRACK CHECK-IN PROGRESS
Primary Actor:	USJ-R Tertiary Students
Goal	To monitor earned HP stamps, weekly achievements, and event participation rewards.
Preconditions	<ul style="list-style-type: none"> • Student is logged into the app. • Gamification system is active and updated with recent user activity.
Postconditions	<ul style="list-style-type: none"> • Gamification data through visual check-in stamp is displayed accurately.
Trigger	Student selects “My Progress” in the Health & Progress section.
Main Flow	<ol style="list-style-type: none"> 1.The student opens the My Progress dashboard within the JoseniCare app. 2.The system retrieves the student’s Health Points (HP) data, including stamps, badges, and reward summaries. 3.The system displays both weekly and cumulative points, visualized through stamp collections and coupon-style icons. 4.The student views detailed progress and may check reward eligibility or milestone achievements (consecutive daily check-ins → reapable benefits).
Alternative Flows	<p>A1. No Data Available: Displays “No progress yet. Complete your daily health check-ins to start earning stamps!”</p> <p>A2. Data Retrieval Error: Displays “Unable to load your progress. Please try again later.”</p>

SSD USE-CASES

ACTOR

UML Use Case Description: Receive Emergency Reports

USE CASE #1:	RECEIVE EMERGENCY REPORTS
Primary Actor:	Security & Safety Department Officer
Goal	To receive emergency reports submitted by students through the SSD emergency channel.
Preconditions	<ul style="list-style-type: none"> • System is online and connected to the central database. • Student has submitted a valid emergency report.
Postconditions	<ul style="list-style-type: none"> • Emergency report is logged in the system. • SSD is notified and can take appropriate action.
Trigger	A student submits an emergency report through the emergency channel connected to the SSD.
Main Flow	<ol style="list-style-type: none"> 1.The SSD dashboard receives a real-time notification of an SOS alert. 2.The System displays the alert containing: <ul style="list-style-type: none"> • The student's name and ID, • Their GPS location and floor level (shown on a map), and • The timestamp of the alert.
Alternative Flows	N/A

UML Use Case Description: Access Allergy Information

USE CASE #2:	ACCESS ALLERGY INFORMATION
Primary Actor:	Security & Safety Department Officer
Goal	To access essential allergy or health restriction data of a student during an emergency.
Preconditions	<ul style="list-style-type: none"> • Student's record exists in the system.
Postconditions	<ul style="list-style-type: none"> • Allergy/health restriction information is displayed temporarily. • Access is logged for audit.
Trigger	In the 'Active Cases' tab under the 'Emergency Reports' section, the SSD Officer clicks the 'View Allergy Info' button on the specified case.
Main Flow	<ol style="list-style-type: none"> 1.The SSD officer requests to view a student's health information during an emergency by clicking the 'View Allergy Info' button on the selected case. 2.The System retrieves and displays limited health data — specifically noted allergies, conditions, and emergency contact. 3.The SSD officer views and records the essential information for response purposes. 4.For privacy protection, the System automatically closes the record after viewing and logs the access in the audit trail.
Alternative Flows	<p>A1. Record Not Found: System notifies that student data is unavailable.</p> <p>A2. Manual Lookup: The SSD Officer searches for the specific student in the Student Health Information section to retrieve allergy data.</p>

UML Use Case Description: Manage Emergency Reports

USE CASE #2:	MANAGE EMERGENCY REPORTS
Primary Actor:	Security & Safety Department Officer
Goal	To monitor, update, and close emergency cases, ensuring accurate incident tracking, communication, and documentation.
Preconditions	<ul style="list-style-type: none"> SSD officer is logged into the system. An emergency report exists in the system.
Postconditions	<ul style="list-style-type: none"> Emergency report status is updated and saved. Notifications are sent to relevant users (student reporter, clinic staff, admin). Closed incidents are archived with complete closure details.
Trigger	SSD officer selects an emergency report from the system for update or closure.
Main Flow	<ol style="list-style-type: none"> The SSD officer opens the list of active emergency reports. The System displays all current reports along with their details and status. The SSD officer selects a specific report to manage. The SSD assigns the case to a responder. The System updates the alert status from “Pending” to “Responding”. The SSD monitors the response progress in real time. Once the situation is resolved, the SSD closes the alert by clicking the “Resolve” button. The System updated the alert status from “Responding” to “Resolved”. The SSD records the incident closure details (Incident Summary, Action Taken, and Final Condition/Outcome) by clicking the “Log” button. The SSD saves the entry by clicking 'Submit Log' once the form is complete. The System validates and saves the closure report. The System marks the incident as Closed and archives it.
Alternative Flows	<p>A1. Missing Required Fields (during closure): System prompts SSD officer to complete all mandatory fields.</p> <p>A2. Connection Failure: System temporarily saves updates locally and syncs once the connection is restored.</p> <p>A3. Submission Error: System retries saving up to 3 times before displaying an error message.</p>

CLINIC STAFF USE-CASES

ACTOR

UML Use Case Description: Manage Appointment Scheduling

USE CASE #1:	MANAGE APPOINTMENT SCHEDULING
Primary Actor:	Clinic Staff (Clinic READS and In-Charge Nurse)
Goal	To monitor, review, or adjust clinic and dental consultation schedules as needed.
Preconditions	<ul style="list-style-type: none"> Nurse and Clinic READS is logged into the web. Appointment scheduling module is accessible. Student appointments are already being booked automatically by the system.
Postconditions	<ul style="list-style-type: none"> Appointment records are updated if manually adjusted. Notifications are sent to affected students or doctors (if rescheduled or cancelled)
Trigger	The Clinic Staff accesses the 'Appointments' page, either via the navigation menu or by clicking 'View Details' on the 'Today's Appointments' dashboard widget.
Main Flow	<ol style="list-style-type: none"> The Clinic Staff navigates to the “Appointments” section. The system automatically renders the “Today” tab, displaying a tabular list of the current day's schedule. The nurse reviews the appointment queue, including details such as: <ul style="list-style-type: none"> Time Student name Purpose/type of appointment Assigned doctor Current status [Cancelled, Attention needed, Queued, Completed] Action Button (Manage or View) The Clinic Staff reviews the queue, paying special attention to items marked “Attention Needed” due to: <ul style="list-style-type: none"> Doctor unavailability Student reschedules Student No-Show For statuses requiring intervention, the Clinic Staff clicks “Manage” on that specific appointment row. The system opens a context-specific pop-up indicating: <ul style="list-style-type: none"> What issue occurred (Notice section) Appointment details Available actions appropriate to the scenario: <ul style="list-style-type: none"> Reschedule (if slots are still available) Reassign Doctor (if other doctors are still available) Cancel Appointment

UML Use Case Description: Manage Appointment Scheduling

Main Flow	<p>7. The Clinic Staff performs the necessary action:</p> <ul style="list-style-type: none"> ◦ Reschedule → selects a new slot from available times ◦ Reassign Doctor → system shows available doctor(s) ◦ Cancel → confirms cancellation if necessary <p>8. Upon confirmation, the system immediately updates:</p> <ul style="list-style-type: none"> ◦ Appointment record ◦ Status indicator ◦ Doctor and student notifications ◦ Queue ordering (cancels and completions removed from order) <ul style="list-style-type: none"> ▪ Removes the appointment from Today if canceled and transfers it to the “Cancelled” tab <p>9. When the doctor finishes a consultation and sets the appointment to Completed, the system automatically:</p> <ul style="list-style-type: none"> ◦ Updates the status to “Completed” ◦ Attaches the doctor’s consultation notes ◦ Enables a “View Details” button on the clinic staff interface <p>10. The Clinic Staff may now click “View details” to review:</p> <ul style="list-style-type: none"> ◦ Student information ◦ Consultation information ◦ Medication prescribed ◦ Doctor’s additional notes <p>11. All changes are saved to the Appointment record and Patient Record history</p>
Alternative Flows	<p>A1. No Appointments Found: Displays: “No appointments booked for this date.”</p> <p>A2. No Available Slots. Displays: “There are no available appointment slots remaining for this semester. This appointment can only be cancelled.”</p> <p>A3. Network or Update Error: Displays: “Failed to update schedule. Please try again later.”</p>

UML Use Case Description: View Health Survey Dashboard

USE CASE #2:	VIEW HEALTH SURVEY DASHBOARD
Primary Actor:	Clinic Staff (Clinic READS and In-Charge Nurse)
Goal	To view a consolidated dashboard containing anonymized health survey results, symptom trends, department-level insights, weather-based health advisories, and supply recommendations.
Preconditions	<ul style="list-style-type: none"> The system has collected anonymized daily health check-in data from students. The system has access to the latest weather data.
Postconditions	<ul style="list-style-type: none"> The system successfully displays the updated anonymized health data, weather insights, and advisory dashboard for clinic staff review.
Trigger	The Clinic Staff selects the “Health Survey” section from the navigation menu.
Main Flow	<ol style="list-style-type: none"> The Clinic Staff opens the Health Survey section. The system loads the default view: Today’s Summary. The system displays the Daily Check-In Summary: <ul style="list-style-type: none"> Total check-ins Students feeling well Reported symptoms Comparison to previous day The Clinic Staff selects a time filter (Today, Past 5 Days, Past 30 Days). The system updates the dashboard contents accordingly: <ul style="list-style-type: none"> Symptom prevalence Observation notes Trend insights (by Department / Program / Year Level) Summary (Today/Past 5 Days/ Past 30 Days) The system fetches and displays current weather data, including: <ul style="list-style-type: none"> Temperature Heat index Humidity UV Index The system generates and displays Today’s Health Advisory based on weather + symptom data. The system generates and displays Today’s Supply Advisory, recommending supplies needed and the reason for each recommendation. Clinic Staff reviews insights to guide resource allocation and health response.
Alternative Flows	<p>A1. No Health Check-In Data Available. Displays: “No health survey data available for the selected period.”</p> <p>A2. No Weather Data Available. Displays: “Weather data unavailable. Unable to generate weather-based advisories.” Health advisory and supply advisory sections adjust to show survey-only-based recommendations.</p> <p>A3. Network / Loading Error. Displays: “Dashboard failed to load. Please refresh or try again later.”</p>

UML Use Case Description: View Doctor's Prescriptions or Notes

USE CASE #3:	VIEW DOCTOR'S PRESCRIPTIONS OR NOTES
Primary Actor:	Clinic Staff (Clinic READS and In-Charge Nurse)
Goal	To access and review doctors' treatment notes or prescriptions for proper documentation and follow-up.
Preconditions	<ul style="list-style-type: none"> Clinic Staff is logged into the web. Doctor's notes or prescriptions exist in the patient's record.
Postconditions	<ul style="list-style-type: none"> Doctor's notes are displayed for viewing. Optional download function is available.
Trigger	Clinic Staff selects a patient record and opens the consultation details.
Main Flow	<ol style="list-style-type: none"> The nurse searches for or selects a patient from the clinic database. The system retrieves the corresponding consultation data, including: <ul style="list-style-type: none"> Student Information Consultation Information Medication Prescribed Additional Notes The nurse views the doctor's notes, prescribed medications, and any related instructions.
Alternative Flows	A1. Patient Record Not Found: Displays: "No consultation records found for this patient." A2. Download Error: Displays: "Failed to export file. Please try again later."



UML Use Case Description: Manage Resource Inventory Dashboard

USE CASE #4:	MANAGE RESOURCE INVENTORY DASHBOARD
Primary Actor:	Clinic Staff (Clinic READS and In-Charge Nurse)
Goal	To monitor and update the inventory of clinic resources such as medicines and medical supplies.
Preconditions	<ul style="list-style-type: none">Clinic Staff is logged into the web.Inventory module is functional.
Postconditions	<ul style="list-style-type: none">Resource quantities are updated (added/edited/deleted) and saved.Alerts are triggered for low-stock, out-of-stock, near-expiry, and health trend recommended items.Inventory audit trail logs the action.
Trigger	Clinic Staff accesses “Inventory” section
Main Flow	<ol style="list-style-type: none">The System displays the Inventory Dashboard, presenting:<ul style="list-style-type: none">Total Items, Low Stock, Out of Stock, Expiring SoonA searchable table of current medical supplies including Item Name, Category, Stock Quantity, Unit, Status, and Expiration Date.A dynamic "Inventory Alerts" side panel highlighting critical items (e.g., Low Stock, Health Trend Demand).The Clinic Staff performs one of the following actions:<ul style="list-style-type: none">To Add New: Clicks the "Add Item" button.To View/Edit: Clicks on a specific item row in the inventory list.Execution of Action:<ul style="list-style-type: none">If Adding New Item:<ul style="list-style-type: none">The System opens the "Add New Inventory Item" modal.The Nurse populates the Item Information (Name, Category, Unit, Description) and Initial Stock Details (Quantity, Expiration Date, or checks "This item does not expire").The System provides a Status Preview (auto-calculated) showing the resulting status (e.g., "IN STOCK") and expiration warning.The Nurse clicks "Add Item" to save.If Viewing/Updating Existing Item:<ul style="list-style-type: none">The System opens the "Item Details" modal (read-only view) showing current stats and specific alerts (e.g., "Quantity below threshold").The Nurse clicks the "Update Item" button.The System opens the "Update Item" modal.The Nurse may edit item details or perform Stock Adjustments by entering values in the "Add Stock" or "Reduce Stock" fields, or updating the Expiration Date.The Nurse clicks "Save Changes".The System validates all inputs (ensuring required fields are filled and quantities are valid numeric values).

UML Use Case Description: Manage Resource Inventory Dashboard

Main Flow	5. The System recalculates the item's total quantity and status based on the inputs. 6. The System updates the central database and refreshes the Dashboard view in real-time. 7. The System logs the transaction in the audit trail. 8. The System closes the modal and displays a success notification to the Clinic Staff.
Alternative Flows	<p>A1. Invalid Data Entry. If the Clinic Staff leaves required fields blank or enters invalid logic (e.g., negative numbers in "Initial Quantity"), the System disables the submission button or highlights the erroneous fields and prompts for correction.</p> <p>A2. Duplicate Item Detected (Add action): If the Clinic Staff attempts to add an item name that exactly matches an existing record, the System prevents the creation and displays an error: "Item already exists. Please update the existing record instead."</p> <p>A3. Delete Item. In the "Update Item" modal, the Clinic Staff clicks the "Delete Item" button. The System displays a confirmation dialog: "Are you sure you want to delete this item? This action cannot be undone." Upon confirmation, the System removes the record from the active inventory, updates the dashboard, and logs the deletion.</p> <p>A4. Update/Save Failed: If the system cannot save changes due to a database error or network issue, it displays a message: "Inventory not updated. Please retry." and retains the input data in the modal so the Clinic Staff does not need to re-type it.</p>

UML Use Case Description: Export Patient Consultation Records

USE CASE #6:	EXPORT PATIENT CONSULTATION RECORDS
Primary Actor:	Clinic Staff (Clinic READS and In-Charge Nurse)
Goal	To export patient medical data, ranging from specific consultation details to the entire medical history, for documentation or referral purposes.
Preconditions	<ul style="list-style-type: none"> Clinic Staff is logged into the web. The Patient Record modal is currently open for a specific student.
Postconditions	<ul style="list-style-type: none"> The requested data is generated into a file (e.g., PDF) and downloaded to the user's local device.
Trigger	The Clinic Staff clicks a "Download" button (Full Record, Section, or Individual Row).
Main Flow	<ol style="list-style-type: none"> The Clinic Staff views the Patient Record modal, which displays sections for Student Information, Alerts and Important Notes, Latest Visits, Dental Records, and Medical Certificates. The Staff determines the scope of the export and performs one of the following actions: <ul style="list-style-type: none"> Full History: Clicks the “Download Full Record” button at the top right. Category History: Clicks the “Download entire section” button (e.g., beside “Latest Visits”). Single Visit: Clicks the specific “Download” button next to a single table row (e.g., a specific check-up on Mar 14). System Processing: The system retrieves the relevant data context based on the button clicked. The system compiles the data into a standardized document format (e.g., PDF). The system automatically triggers the download of the file to the staff's device.
Alternative Flows	<p>A1. Empty Section Export: If the staff attempts to "Download entire section" for a category with no records, the system displays a notification: "No records available to export for this section."ompt nurse to select a record first.</p> <p>A2. Generation Failure: If the system cannot generate the file (network/server error), it displays: "Export failed. Please try again."</p>

DOCTOR USE-CASES

ACTOR

UML Use Case Description: View and Update Student Appointments

USE CASE #1:	VIEW AND UPDATE STUDENT APPOINTMENTS
Primary Actor:	Doctor
Goal	To review the daily consultation schedule and update appointment statuses (start consultation, cancel due to unavailability, or mark as no-show).
Preconditions	<ul style="list-style-type: none"> • Doctor is logged into the application. • Appointments have been scheduled and confirmed by the Clinic Staff/Students.
Postconditions	<ul style="list-style-type: none"> • Appointment status is updated in the system (e.g., Completed, Cancelled, No-Show) and the relevant parties are notified.
Trigger	Doctor clicks the "Appointments" icon in the sidebar navigation.
Main Flow	<ol style="list-style-type: none"> 1. The System displays the Appointments Dashboard with the "Today" tab active by default. The list includes: <ul style="list-style-type: none"> ◦ Time, Student Name, Purpose/Type, and Status (e.g., Queued, Attention Needed, In Progress). 2. The Doctor reviews the queue. For upcoming slots, the status is typically "Queued" or "In Progress." 3. The Doctor clicks the "View Details" button for a specific student. 4. The System opens the "Consultation Details" modal, displaying: <ul style="list-style-type: none"> ◦ Appointment Info (Name, Purpose, Time). ◦ Primary Action: "Start Patient Record". ◦ Other Options: "Mark as No-Show" or "Cancel Appointment". 5. The Doctor clicks "Start Patient Record." 6. The System redirects the Doctor to the "Patient Records" module for that specific student, allowing them to begin documenting the medical case.
Alternative Flows	<p>A1. Mark as No-Show: In the "Consultation Details" modal, the Doctor clicks "Mark as No-Show". The System displays a warning modal: "You are about to mark this appointment as a NO-SHOW." The Doctor clicks "Confirm No-Show." The System updates the status and notifies the clinic staff.</p> <p>A2. Doctor-Initiated Cancellation: In the modal, the Doctor clicks "Cancel Appointment." The System opens the "Cancel Appointment" modal with a notice about "Doctor-Side Unavailability." The Doctor must select a reason from the checklist (Medical Emergency, Off-Campus Duty, Health Issue, Schedule Conflict, or Other). The Doctor clicks "Confirm Cancellation." The System updates the status to "Cancelled," logs the reason, and triggers a notification to the Clinic Staff to assist the student in rescheduling.</p> <p>A3. Record Not Found: System displays "No health record available."</p> <p>A4. Save/Update Failure: System notifies the doctor of a failed save and prompts a retry.</p> <p>A5. Missing Required Information: System prompts the doctor to complete mandatory fields</p>

UML Use Case Description: Access and Update Individual Digital Health Records

USE CASE #1:	ACCESS AND UPDATE INDIVIDUAL DIGITAL HEALTH RECORDS
Primary Actor:	Doctor
Goal	To retrieve a student's medical history, view critical alerts, and document new consultation data (physical assessments, diagnoses, and prescriptions).
Preconditions	<ul style="list-style-type: none"> The Doctor is logged into the JoseniCare web application. Student's health record exists in the system.
Postconditions	<ul style="list-style-type: none"> Student's health record is displayed and may be updated. Any added notes or prescriptions are saved and timestamped. Access and modifications are logged for audit purposes.
Trigger	Doctor selects a student's record from the appointment list or search results from the "Patient Records" section.
Main Flow	<ol style="list-style-type: none"> The System displays the Patient Records list. The Doctor locates the specific student by: <ul style="list-style-type: none"> Entering the Name or ID in the Search Bar. Using Filters (e.g., Department, Year Level) or Sort By functions. The Doctor clicks on the student's row in the results table. The System opens the Patient Record modal overlay. The System displays the student's profile, explicitly highlighting: <ul style="list-style-type: none"> Student Info: Name, ID, Course/Year. Alerts & Important Notes: Critical Allergies (e.g., Peanuts) and Conditions (e.g., Asthma). History: Previous Consultation Notes, Latest Visits, Dental Records, and Medical Certificates Issued. The Doctor clicks the "Add New Assessment" button under the Quick Actions section. The System opens the Assessment Details modal. The Doctor populates: <ul style="list-style-type: none"> Reason for Visit. Physical Examination Findings (Temperature, Blood Pressure, Pulse Rate, Respiratory Rate). Diagnosis. Clinical Notes. The Doctor confirms/saves the assessment. If Doctor determines medication is necessary after step 5: <ul style="list-style-type: none"> From the Patient Record modal, the Doctor clicks "Add New Prescription" under Quick Actions. The System opens the Prescription Details modal. The Doctor enters: <ul style="list-style-type: none"> Medication Name (e.g., Paracetamol). Dosage Instructions (e.g., 1 tablet every 6-8 hours). Duration and For (Indication).

UML Use Case Description: Access and Update Individual Digital Health Records

Main Flow	<ul style="list-style-type: none"> ◦ The Doctor clicks the inner “Add Medication” button. The system validates the entry and displays it in the summary list (e.g., “1.) Paracetamol 500mg...”). ◦ The Doctor adds any general notes and closes the modal to save the prescription record. <p>6. The System saves the data, adds it to the “Consultation Notes” history table, and updates the “Latest Visit” timestamp.</p>
Alternative Flows	<p>A1. Viewing Historical Data: Instead of adding new data, the Doctor clicks the “View Details” button on a specific row in the Consultation Notes or Dental Records tables to read about a past procedure or check-up.</p> <p>A2. Modifying a Prescribed Item: During the prescription sub-flow, if the Doctor makes a mistake before saving, they click the “Remove” button (pink badge) next to the listed medication to delete it and re-enter the correct details.</p> <p>A3. Save/Update Failure: System notifies the doctor of a failed save and prompts a retry.</p> <p>A4. Missing Required Information: System prompts the doctor to complete mandatory fields</p>

UML Use Case Description: View Health Survey Dashboard

USE CASE #3:	VIEW HEALTH SURVEY DASHBOARD
Primary Actor:	Doctor
Goal	To view a consolidated dashboard containing anonymized health survey results, symptom trends, department-level insights, weather-based health advisories, and supply recommendations.
Preconditions	<ul style="list-style-type: none"> The system has collected anonymized daily health check-in data from students. The system has access to the latest weather data.
Postconditions	<ul style="list-style-type: none"> The system successfully displays the updated anonymized health data, weather insights, and advisory dashboard for clinic staff review.
Trigger	The Doctor selects the “Health Survey” section from the navigation menu.
Main Flow	<ol style="list-style-type: none"> The Doctor opens the Health Survey section. The system loads the default view: Today’s Summary. The system displays the Daily Check-In Summary: <ul style="list-style-type: none"> Total check-ins Students feeling well Reported symptoms Comparison to previous day The Doctor selects a time filter (Today, Past 5 Days, Past 30 Days). The system updates the dashboard contents accordingly: <ul style="list-style-type: none"> Symptom prevalence Observation notes Trend insights (by Department / Program / Year Level) Summary (Today/Past 5 Days/ Past 30 Days) The system fetches and displays current weather data, including: <ul style="list-style-type: none"> Temperature Heat index Humidity UV Index The system generates and displays Today’s Health Advisory based on weather + symptom data. The system generates and displays Today’s Supply Advisory, recommending supplies needed and the reason for each recommendation. Doctor reviews insights to guide resource allocation and health response.
Alternative Flows	<p>A1. No Health Check-In Data Available. Displays: “No health survey data available for the selected period.”</p> <p>A2. No Weather Data Available. Displays: “Weather data unavailable. Unable to generate weather-based advisories.” Health advisory and supply advisory sections adjust to show survey-only-based recommendations.</p> <p>A3. Network / Loading Error. Displays: “Dashboard failed to load. Please refresh or try again later.”</p>

UML Use Case Description: View Medicine Resource Utilization Dashboard

USE CASE #3:	VIEW MEDICINE & RESOURCE UTILIZATION DASHBOARD
Primary Actor:	Doctor
Goal	To view current stock levels, check expiration dates, and identify shortages of medical supplies to ensure availability for patient treatment.
Preconditions	<ul style="list-style-type: none"> • Doctor is logged into the web application. • Clinic inventory data is up to date and synchronized.
Postconditions	<ul style="list-style-type: none"> • The Doctor has obtained the necessary stock information (no data is modified).
Trigger	Doctor clicks the “Inventory” icon in the sidebar navigation.
Main Flow	<ol style="list-style-type: none"> 1. The System displays the Inventory Dashboard, presenting: <ul style="list-style-type: none"> ◦ Total Items, Low Stock, Out of Stock, Expiring Soon ◦ A searchable table of current medical supplies including Item Name, Category, Stock Quantity, Unit, Status, and Expiration Date. ◦ A dynamic "Inventory Alerts" side panel highlighting critical items (e.g., Low Stock, Health Trend Demand). 2. The Doctor locates a specific medical supply by either: <ul style="list-style-type: none"> ◦ Typing the item name (e.g., "Paracetamol") in the Search Bar. ◦ Using the "Sort By" or "Filters" dropdowns to narrow the list. 3. The Doctor clicks on a specific row in the inventory list to view more details. 4. The System opens the "Item Details" modal (Read-Only view), displaying: <ul style="list-style-type: none"> ◦ Item Information: Name, Category, Unit, and Description/Usage. ◦ Stock Details: Current Quantity, calculated Status (e.g., "LOW STOCK"), and Expiration status. ◦ Alerts & Important Notes: Specific warnings such as "Quantity below threshold" or "Expiring Soon." 5. The Doctor reviews the information and clicks the "Close" button to return to the dashboard.
Alternative Flows	<p>A1. No Results Found: If the Doctor searches for an item that is not in the database, the system displays “No items found” in the list area.</p> <p>A2. Connection Error: System retries connection or prompts to reload page.</p>