<<interface>> INotification
+ sendNotification(): void
+ receiveNotification(): void

implements

<<interface>> IJobHandler
+ allocateJob(): void
+ getJobDetails(): void

implements
implements
implements

<<enumeration>> JobType
WEEKEND_NIGHT
WEEKDAY_NIGHT
WEEKEND_DAY
WEEKDAY_DAY

<<enumeration>> NotifStatus
READ
UNREAD
RESOLVED
UNRESOLVED

<<enumeration>> JobStatus
PENDING
ASSIGNED
COMPLETED
REJECTED

<<boundary>> JobUI
+ displayJobDetails(): staff[]
+ rejectJobPrompt(): void

uses

<<entity>> Notification
- notificationID: int
- message: String
- recipientID: int
- status: String
+ getNotificationID(): int
+ setNotificationID(notifID: int): void
+ getRecipientID(): int
+ setRecipientID(recipientID: int): void
+ getMessage(): String
+ setMessage(message: String): void
+ getNotifStatus(): String
+ setNotifStatus(notifStatus: String): void
+ getNotifStatus(): String
+ getRejectionDate(): date
+ getRejectionReason(): String
+ setRejectionDate(date): void
+ setRejectionReason(String): void
+ sendNotification(recipientID: String, message: String): void

uses

<<control>> JobAssignmentController
+ allocateJob(job: Job, staffID: staff): void
+ updateJobStatus(job: Job): void
+ rejectJob(job: Job, staffID: staff): void
+ getJobDetails(jobID: String): void
+ sendNotification(): void

uses

<<control>> RejectionController
+ updateRejectionDate(): date
+ updateRejectionReasons(): String

<<boundary>> ManagerUI
+ displayJobAssignments(): void
+ displayNotifications(): void
+ displayAllocateJobs(): void
+ displayCreateJob(): void
+ displayOver40HourStaff(): void
+ displayLowestWorkloadStaff(): void
+ displayExportWorkload(): void
+ displayModifyStaffWorkload(): void

uses

<<entity>> Job
- jobID: String
- jobTitle: String
- description: String
- duration: int
- assignedTo: Staff
- jobType: String
- jobStatus: String
+ getJobID(): String
+ setJobID(jobID: String): void
+ getJobTitle(): String
+ setJobTitle(jobTitle: String): void
+ getDescription(): String
+ setDescription(description: String): void
+ getDuration(): int
+ setDuration(duration: int): void
+ getAssignedTo(): int
+ setAssignedTo(staffID: int): void
+ getJobType(): String
+ setJobType(jobType: String): void
+ getJobRoute(): String
+ setJobRoute(jobRoute : String): void
+ getJobStatus(): void
+ setJobStatus(status: JobStatus): void

<<boundary>> NotificationUI
+ displayNotificationDetails(): string

uses

<<boundary>> StaffUI
+ displayJobAssignments(): void
+ displayIndicatePreferenceAvailability(List<String>): void
+ displayRejectionPage(): void
+ displayNotifications(): void

uses

<<control>> ManagerController
+ addJobs(): void
+ retrieveJobAssignments(staffID: String): void
+ calculateOver40HourStaff(): staff[]
+ calculateLowestWorkloadStaff(): staff[]
+ exportStaffWorkload(): void
+ allocateJob(staffID: String): void
+ modifyStaffWorkload(job: Job, staffID: String): void
+ getJobDetails(staffID: String): void

uses

<<control>> StaffController
+ updatePreferenceAvailability(staffID: int): void
+ retrieveJobAssignments(): void

uses

<<boundary>> AdminUI
+ displayCreateUser() : void
+ displayEditUser(): void
+ displayDeleteUser(): void

uses

<<entity>> Staff
- staffID: int
- preferences: List<String>
+ getStaffID(): int
+ setStaffID(staffID: int): void
+ getPreferenceAvailability(): List<String>
+ setPreferenceAvailability(List<String>): void

assigns jobs to 1..* 1

<<entity>> Manager
- managerID: int
+ getManagerID() : int
+ setManagerID(managerID: int): void

<<entity>> User
- name: String
- role: String
- userID: int
- email: String
- password: String
- isLoggedIn: Boolean
+ getName(): String
+ setName(name: String): void
+ getRole(): String
+ setRole(): void
+ getUserID: int
+ setUserID(userID: int): void
+ getEmail(): String
+ setEmail(email: String): void
+ getIsLoggedIn(): Boolean

<<control>> AdminController
- staffList: List<Staff>
+ createUser(staffID: int): void
+ updateUser(staffID: int): void
+ deleteUser(staffID: int): void

<<entity>> Admin
- adminID: int
+ getAdminID(): int
+ setAdminID(adminID: int) : void

uses

<<entity>> PreferenceAvailability
- preferenceID: int
- unavailableSlots: List<DateTime>
- jobTypePreferences: List<String>
+ getPreferenceID(): int
+ setPreferenceID(preferenceID: int): void
+ getPreferredJobs(): List<String>
+ setPreferredJobs(jobs: List<String>): void
+ getUnavailableSlots(): List<DateTime>
+ setUnavailableSlots(dates: List<DateTime>): void

1..*