Design Class Diagram (DCD) EventManagement UserManagement KitchenManagement SummarySheet UserManager enum enum Variation **0..**n UserRole VariationType compiles -order: TaskListOrder event sender methods propose currentVariation -voice: string -editable: boolean | cancella manager ` 0..1 -type: string modifica cook operations methods +create(order: boolean, editable: boolean) MenuManager -state: string chef aggiungi enum SummarySheet +setCurrentUser(u: User -menuFeatures: String[] waiter TaskListOrder +getCurrentUser(): User -setOrder(order: TaskListOrder) event sender methods +orderTaskList(order: TaskListOrder) ByImportance ` 0..l +addReceiver(er: MenuEventReceiver) 0..1 currentUser +setEditable(flag: boolean) byDifficulty +removeReceiver(er: MenuEventReceiver) -addTaskToSheet(t: Task) MenuEventReceiver -notifyMenuCreated(m: Menu) User +getTaskList(): ArrayList<Task> {interface} -notifyMenuDeleted(m: Menu) +setTaskList() Task -name: string -notifyVariationCreated(v: Variation) +move(index: int, t: Task) -surname: string -name: string -notifyVariationModified(v: Variation) +updateMenuCreated(m: Menu) -role: string -ingredients: string[] -notifyVariationDeleted(v: Variation) +updateMenuDeleted(m: Menu) +isChef(): boolean -staffInstructions: string[] +updateVariationCreated(v: Variation) operations methods -alrteadyDoneNote: string +isCook(): boolean +updateVariationDeleted(v: Variation) +createMenu(title: String): Menu -recipe: Recipe +isWaiter(): boolean +updateVariationModify(v: Variation) approves +deleteMenu(title:string) -start: Date +isManager(): boolean +createVariation(voice: string): Variation contains -end: Date +modifyVariation(voice: string): Variation -cook: User +deleteVariation(voice: string) 0..n menuReceivers + getRecipe(): Recipe disciples +getVariationQueue(): ArrayList<Variation organizes + setRecipe(): Recipe works + getStart(): Task + getEnd(): Date Shift composed by applie - getCook(): User currentMenu + setCook(cook: User) -start: string setAlreadyDoneNote(note: string) -end: string -type: string Menu -limit: int Recipe Kitchen -title: string +create(start: string, end: string, produces -published: boolean -name: string type: string, limit: int):Shift -hotDish: string -steps: Preparation +setStart(start: string) -requiresKitchen: boolean +setEnd(end: string) buffet: string +setStaff(staff: string[]) -fingerFood: string composes +setType(type: string) EventEventReceiver +create(title: string, published: boolean +setLimit(limit: int) currentKitchen {interface} hotDish: string[], requiresKitchen: boolean, \ +getShift(): Shift Preparation buffet: string, fingerFood: string) +isAvailableInTimeframe(u: User, start: Date, end: Start) +updateEventCreated(e: Event) creates +setTitle(title: string) +makeAvailableInTimeframe(u: User, start: Date, end: Start) -name: string +updateEventDeleted(e: Event) kitchenReceivers +setPublished(published: boolean) +updateEventModified(e: Event) -description: string +setHotDish(piattiCaldi: string[]) +updateRecurrencyCreated(r: Recurren KitchenManager +setRequiresKitchen(requiresKitchen: boole +updateRecurrencyDeleted(r: Recurren currentShift +setBuffet(buffet: string) -notifySummarySheetCreated(s: SummarySheet) +updateRecurrencyModified(r: Recurre +setFingerFood(fingerFood: string) -notifySummarySheetDeleted(s: SummarySheet) +updateServiceCreated(serv: Service) -notifyTaskCreated(t: Task) +getRecipe(): ArrayList<Recipe> +updateServiceDeleted(serv: Service) ShiftManager enum -notifyTaskModified(t: Task) +updateServiceModified(serv: Service) ShiftType event sender methods -notifyTaskRemoved(t: Task) -notifyShiftRemoved(sh: Shift) -notifyRecipeCreated(re: Recipe) preparatory KitchenEventReceiver -notifyShiftCreated(sh: Shift) -notifyRecipeModified(re: Recipé) service eventReceivers -notifyShiftModified(sh: Shift) updateSummarySheetCreated(sh: SummaryShee -notifyRecipeRemoved(re: Recipe) updateSummarySheetDeleted(sh: SummaryShee -notifyPreparationCreated(p: Preparation) operations methods Service EventManager updateTaskCreated(t: Task) -notifyPreparationRemoved(p: Preparation) +createShift(start: string, end: string, -title:string updateTaskDeleted(t: Task) -notifyPreparationModified(p: Preparation) type: string, limit: int): Shift -currentEvent: Event -offset: string updateTaskModify(t: Task) -notifyAssignTask(t: Task, s: SummarySheet) +modifyShift(start: string, end: string, event sender methods -confimed: boolean updateRecipeCreated(re: Recipe) type: string, limit: int): Shift +createSummarySheet(Service e): SummaryShee -notifyEventCreated(e: Event) -menu: Menu updateRecipeDeleted(re: Recipe) +deleteShift(start: string, end: string, +deleteSummarySheet(Service e) -notifyEventModified(e: Event) -sheet: SummarySheet updateRecipeModify(re: Recipe) type: string, limit: int) -notifyEventDeleted(e: Event) +createTask(name: string): Task +create(title: string, offset: string, updatePreparationCreated(p: Preparation) +askStaffAvailability() +deleteTask(name: string) -notifyRecurrencyCreated(r: Recurrency) confirmed: boolean): Service updatePreparationDeleted(p: Preparation) +getShiftList(startDate:Date, endDate:Date): List<Shift> +modifyTask(name: string): Task -notifyRecurrencyModified(r: Recurrency) +create(offset: string) updatePreparationModify(p: Preparation) +makeUserAvailable(u: User, s: shift) +createRecipe(name: string): Recipe -notifyRecurrencyDeleted(r: Recurrency) +getSummarySheet() +isShiftFullInTimeframe(s: shift, start:Date, end:Date) +deleteRecipe(name: string) -notifyServiceCreated(serv: Service) +setSummarySheet(sheet: SummarySheet) +modifyRecipe(name: string): Recipe -notifyServiceDeleted(serv: Service) 0 n shiftReceivers +getMenu(): Menu Penal +createPreparation(name: string): Preparation -notifyApproveMenu(e: Event) +setMenu() +deletePreparation(name: string) ShiftEventReceiver description: string operations methods +setTitle(title: string) +modifyPreparation(name: string): Preparation {interface} sum: int +setCurrentEvent(e: Event) +setOffset(offset: string) +getRecipeInstructions(): Task +setCurrentRecurrency(r: Recurrency) +setConfirmed(confirmed: boolean) +assignTask(u: User, t: Task): Task +createEvent(title: string): Event +changeTaskAssignment(t: Task, cook: User) +updateShiftCreated(sh: Shift) +deleteEvent(title: string) +assignCookToService(s: Service, c: User) +updateShiftDeleted(sh: Shift) +modifyEvent(title: string):Event +updateShiftModify(sh: Shift) +createRecurrency(startRec: string, endRec: string, countRec: int, typeRec: st occurrency: int): Recurrency +modifyRecurrency(startRec: string, has endRec: string, countRec: int, typeRec: str occurrency: int, offset[]): Recurrency +deleteRecurrency(startRec: string, Event endRec: string, countRec: int, typeRec: st -title:string └ 0..n SortUtils occurrency: int) General module -start: string +createService(title: string): Service defines -end: string +deleteService(title: string) -location: string TaskSorte Exception ShiftException UseCaseLogicException +modifyService(title: string): Service currentEvent docs: string +approveMenu() -notes: string[] +getMenu(): Menu -isElegant: boolean +getServices(): ArrayList<Service> -isPrivate:boolean +getSummarySheet(): SummarySheet -guest: int EventException UserException +getCurrentService(): Service -type: string +getCurrentTime(): LocalDateTime -isOngoing: boolean +create(title: string, start: string, end: s location: string, Recurrency isElegant: boolean, isPrivate: boolean, ServiceException KitchenException startRec: string guest: int, type: string, onGoing: boolear endRec: string services: ArrayList<Service>): Event countRec: int +create(start: string, end: string) Section typeRec:string +setTitle(title: String) -description: stri occurrency: int +setStart(start: string) SummarySheetException MenuException +create(startRec: string, endRec: string, +setEnd(end: string) countRec: int, typeRec: string, occurrency: int) +setLocation(location: string) occurs +getEvents(): ArrayList <Event> +setDocs(docs: string) made by +setStartRec(startRec: string) +setNotes(notes: string[]) RecurrencyException +setEndRec(endRec: string) +setElegant(isElegant: boolean) VariationException Menultem +setPrivate(isPrivate: boolean) +setCountRec(countRec: int) +setTypeRec(typeRec: string) title: string +setGuest(guest: int) +setOccurrency(occurrency: int) +setType(type: string) +setOngoing(isOngoing: boolean)