UML Model Entities:

RegularUser

- isBanned : boolean
- + RegularUser(username : String, password : String)
- + RegularUser()
- + setIsBanned(banned: boolean): void
- + getIsBanned() : boolean + isAdmin() : boolean

Calendar

- calendarID : String- name : String
- description : String
- + Calendar(name : String, description : String)
- + setName(name : String) : void
- + setDescription(description String): void
- + getName() : String
 + getDescription() : String
- + getCalendarID () : String
- + setCalendarID(CalendarID : String) : void
- + toString(): String

<< abstract>> User

- username : String
- password : String
- ownedCalendars: List<String>viewableCalendars: List<String>editableCalendars: List<String>
- + User(username : String, password : String)
- + getUsername() : String + getPassword() : String
- + setUsername(username : String) : void + setPassword(password : String) : void
- + authenticate(username : String , password : String) : Boolean
- + toString() : String + isAdmin(): boolean
- + ownsCalendar (calendarID : String) : boolean
- + getOwnedCalendarsID() : List<String>
- + getEditableCalendarsID() : List<String>
- + getViewableCalendarsID() : List<String>
- + canEditCalendar(calendarID : String) : boolean
- + canViewCalendar(String calendarID) : boolean
- + clearSharedCalendars(): void
- + editSharingPermission(calendarID : String, editAccess : boolean) : void
- + addOwnedCalendar(calendarID : String) : void
- + removeCalendar(calendarID : String) : void

AdminUser

- + AdminUser(username : String,
- password : String)
 + AdminUser()
- + isAdmin() : boolean

<<abstract>>

NameDuplicatable

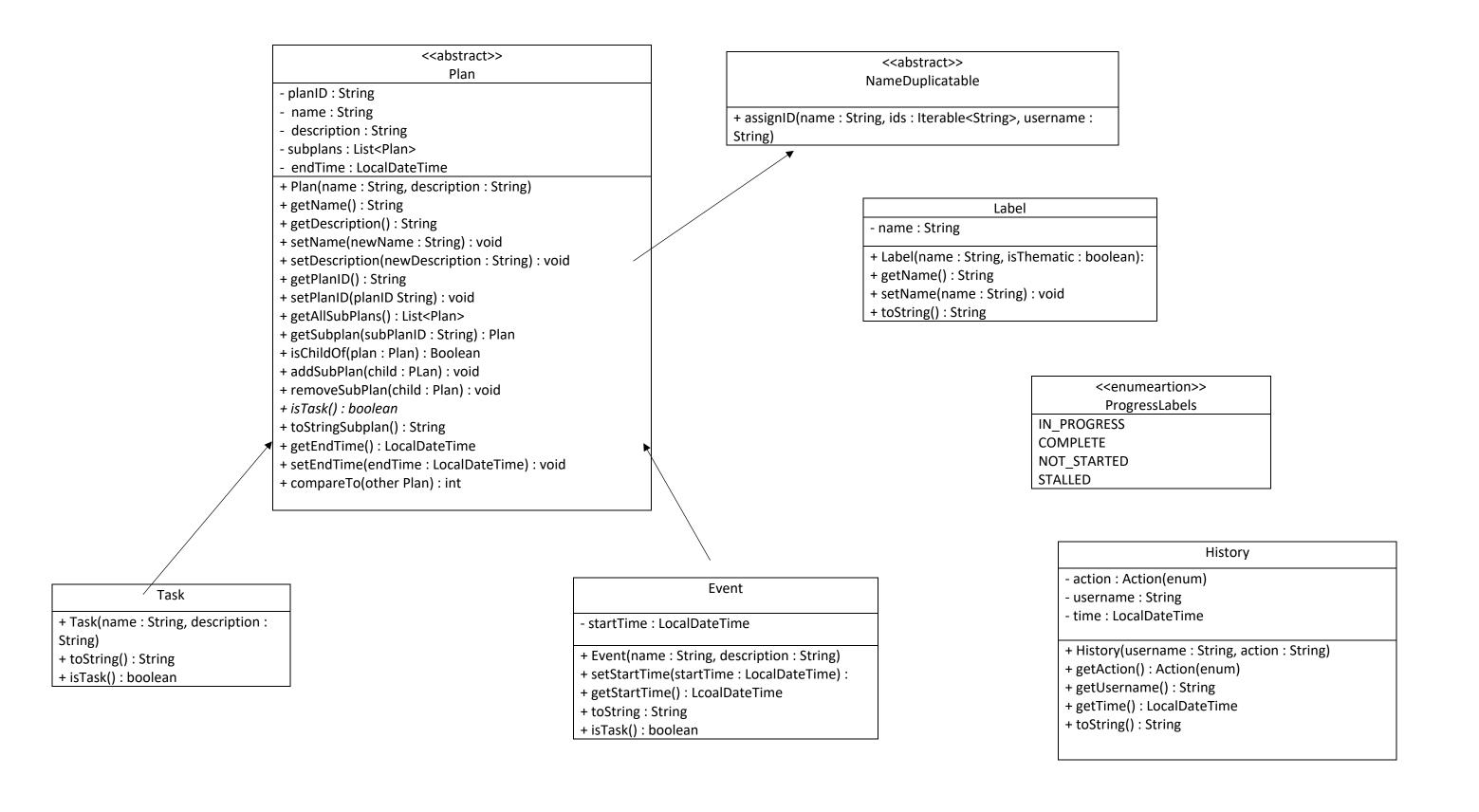
+ assignID(name : String, ids : Iterable<String>, username : String)

TimeParser

- + parseTime(time : String) : LocalDateTime
- YYYYMMDDHHMM(time : String) : LocalDateTime
- DDHHMM(time : String) : LocalDateTime
- MMDDHHMM(time : String) : LocalDateTime
- YYYYMMDDHH(time : String) : LocalDateTime
- YYYYMMDD(time : String) : LocalDateTime
- MMDD(time : String) : LocalDateTime
- DD(time : String) : LocalDateTime
- intSlice(text : String, i : int, j : int) : int
- tomorrow() : LocalDateTime
- nextDat(time : String) : LocalDateTime

Comment

- text : String
- username : String
- id : String
- posted : LocalDateTime
- updated : LocalDateTime
- replies : List<Comment>
- + Comment(text : String, username : String)
- + Comment()
- + getText() : String
- + setText(text : String) : void
- + toString(): String
- toNestedString(): String
- + getUsername() : String
- + setUsername(username : String) : void
- + addReply(c : Comment) : void
- + removeReply(c : Comment) : void
- + getReplies() : List<Comment>
- + getId() : String
- + setId(id : String) : void
- + containsReplyId(id : String) : boolean



Use Case Class: LoggingManager - currUser : User + getCurrUsername() : String + getCUrrPassword() : String + loggedOut(): boolean + userLoggedIn(): boolean + adminUserLoggedIn(): boolean + logIn(user : User) : String + logOut() : String HistoryManager - histories : Map<LocalDateTime, History> + HistoryManager() + addHistory(username : String, action : String): void + getHistories() : String + getHistoriesByUser(username : String) : String + clearHistoriesByUser(username): void

newUsername: String): void

property: String): List<String>

getHistoriesByUserAsList(username :

+ clearComments(planID : String) : void

+ handleNewPlan(planID : String) : void

+ handleDeletePlan(planID : String) : void

+ toString(planID : String) : String

+ containsPlanID(planID: String)

+ save() : void

+ load() :void

String) : List<History>

CommentManager

String): void

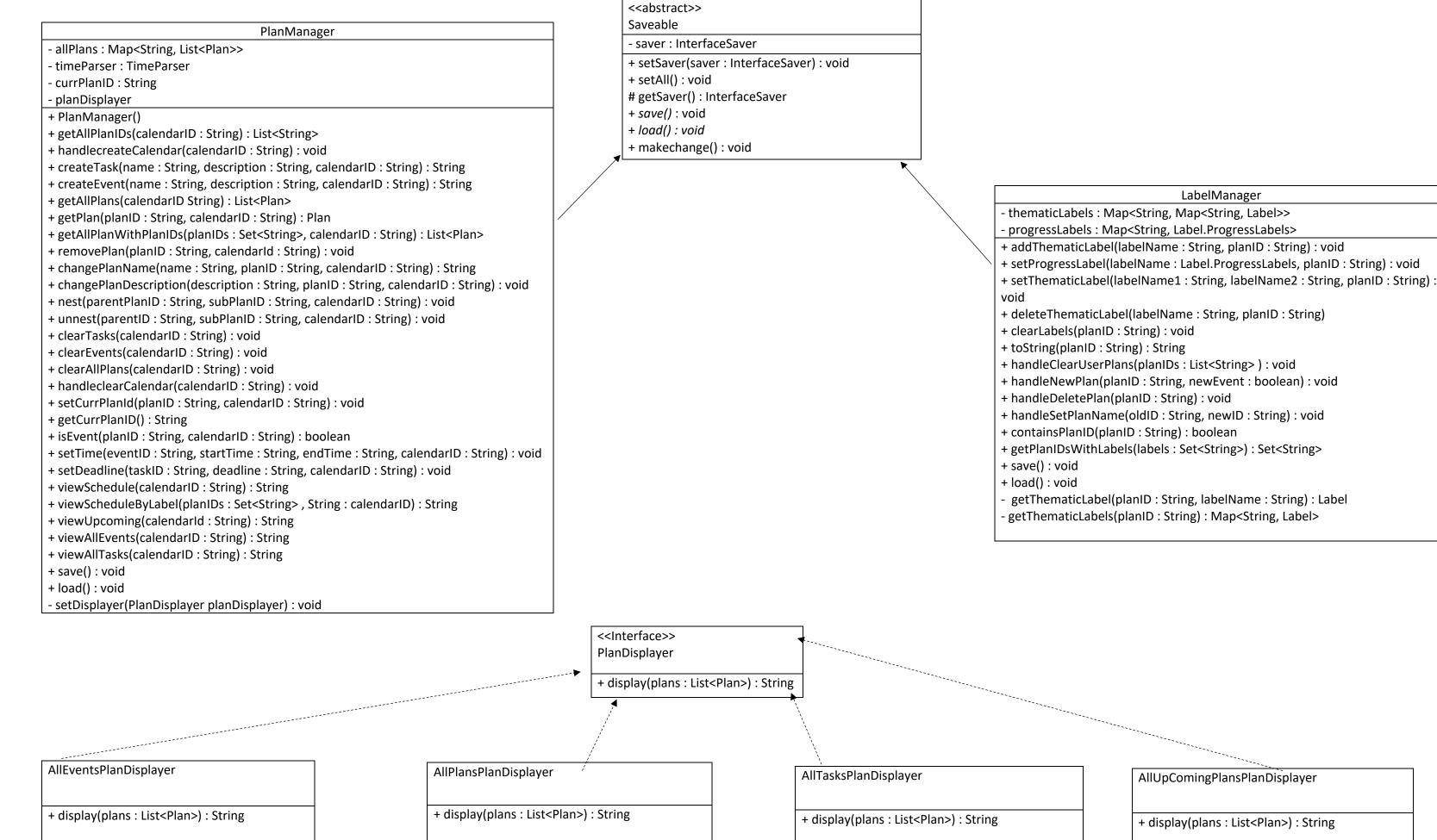
+ save(): void + load(): void

void

<<abstract>>

Saveable calendars : Map<String, Calendar> - saver : InterfaceSaver currCalendarID : String + CalendarManager() + setSaver(saver : InterfaceSaver) : void + getCalendar(calendarID : String) : Calendar + setAll() : void # getSaver(): InterfaceSaver + createCalendar(name: String, description: String, username: String): String + deleteCalendar(calendarID : String) : String + save(): void + changeCalendarName(newName : String, calendarID : String) : void + load(): void + changeCalendarDescription(description: String, calendarID: String): void + makechange(): void + setCurrCalendarID(currCalendarID : String) : boolean + viewAllOwnedCalendars(ownedCalendarIDs : List<String>) : String + viewSharedCalendars(editableCalednarsIDs: List<String>, viewabelCalendarIDs: List<String>): String + containCalendar(calendarID : String) : boolean + deleteCalendars(calendarIDs : List<String>) : void + getCurrCalendarID(): String + getCalendarName(calendarID : String) : String + getCalendarDescription(calendarID : String) : Strin + save() : void + load(): void - viewEditableCalendars(editableCalendarIDs : List<String>) : String - viewViewableCalendars(viewableCalendars : List<String>) : String - addCalendar(c : Calendar) : void + handleChangeUsername(username : String, UserManager users : Map<String, User> + getHistoryPropertyAsList(username : String, + getUser(username : String) : User + setPassword(username : String, newPassword : String) : void + setUsernam(username : String, newUsername : String) : void + getAllRegularUsernames(): List<String> + getAllUsernames() : List<String> + authenticateUser(username : String, password : String) : User + isBanned(username: String): Boolean + createUser(username : String, password : String) : void + createAdminUser(username : String, password : String) : void - comments : Map<String, Map<String, Comment>> + deleteUser(username : String, deleteSelf : boolean) : void + addReply(planID : String, commentID : String, username : String, text : + banUser(username : String) : void + unbanUser(username : String) : void + addComment(planID : String, username : String, text : String) : String + addCalendar(calendarID : String, username : String) : void + editComment(planID : String, commentID : String, text : editComment) : + deleteCalendar(calendarID : String, username : String) : void + clearCalendars(username : String) : void + deleteComment(planID : String, commentID : String) : void + shareCalendar(calendarID: String, targetUser: String, calOwner: String, access: boolean) + deleteOrRemoveAccess(calendarID : String, username : String) : void + getComment(planID : String, commentID : String) : Comment + getOwnedCalendars(username : String) : List<String> + getComments(planID : String) : Map<String, Comment> + getEditableCalendars(username : String) : List<String> + getViewableCalendars(username : String) : List<String> + handleClearUserPlans(planIDs : List<String>) : void + validateCalendarForEditAccess(username : String, calendarID : String) : boolean + save(): void + load(): void + handleSetPlanName(oldID : String, newID : String) : void - addUser(newUser : User) : void getAllReplyIDs(planID : String, commentID : String) : List<String>

CalendarManager



Controller:

ShellController

cals: CalenarManager plans : PlanManager - labs : LabelManager

comms: CommentManager

hists: HistoryManager users: UserManager logs: LoggingManager

- s : InterfaceSaver p: ShellPresenter

menus : Map<MenuState, ShellMenu>

+ ShellController() + run(): void - load() : void

MenuState <<abstract>> ShellMenu **USER MENU**

<<enumeration>>

ADMIN USER MENU **EDIT CALENDAR**

EDIT LABEL EVENT

EDIT LABEL TASK

EDIT COMMENT

LOG OUT

+ commands: Map<String, Runnable>

menu State : MenuState

setMenuState(menuState : MenuState) : void

getMenuState() : MenuState

handelInput(data: ShellData, input: String): void

+ run(presenter: ShellPresenter): void

ShellMenuLoggedOut

+ ShellMenuLoggedOut(p:

ShellPresenter, users: UserManager,

logs: LoggingManager, hists:

HistoryManager)

+ run(p : ShellPresenter) : void

logIn(p : ShellPresenter, users :

UserManager, logs: LoggingManager,

hists: HistoryManager): void

newUser(p : ShellPresenter, users :

UserManager, hists: HistoryManager):

void

ShellMenuEditPlan

+ ShellMenuEditPlan(p : ShellPresenter, cals : CalendarManager, plans :

PlanManager, labs: LabelManager, comms: CommentManager)

+ run(p : ShellPresenter) : void

- back() : void

- addEvent(p : ShellPresenter, cals : CalendarManager, plans : PlanManager, labs :

LabelManager, comms: CommentManager): void

- addTask(p : ShellPresenter, cals : CalendarManager, plans : PlanManager, labs :

LabelManager, comms: CommentManager): void

- addSubevent(p : ShellPresenter, cals : CalendarManager, plans : PlanManager, labs : LabelManager, comms: CommentManager): void

- addSubtask(p : ShellPresenter, cals : CalendarManager, plans : PlanManager, labs : LabelManager, comms: CommentManager): void

- editname(p : ShellPresenter, cals : CalendarManager, plans : PlanManager, labs :

LabelManager, comms: CommentManager): void

- editDescription(p : ShellPresenter, cals : CalendarManager, plans : PlanManager) :

editLabels(p : ShellPresenter, cals : CalendarManager, plans : PlanManager) : void

- editTime(p : ShellPresenter, cals : CalendarManager, plans : PlanManager) : void

- nestPlan(p : ShellPresenter, cals : CalendarManager, plans : PlanManager) : void

- unnestPlan(p : ShellPresenter, cals : CalendarManager, plans : PlanManager) : void

- deletePlan(p : ShellPresenter, cals : CalendarManager, plans : PlanManager, labs :

LabelManager, comms: CommentManager): void

- clearTasjs(p : ShellPresenter, cals : CalendarManager, plans : PlanManager) : void

- clearEvents(p : ShellPresenter, cals : CalendarManager, plans : PlanManager) : void

- viewEvents(p : ShellPresenter, cals : CalendarManager, plans : PlanManager) : void

- viewTasks(p : ShellPresenter, cals : CalendarManager, plans : PlanManager) : void

- editComments(p : ShellPresenter, cals : CalendarManager, plans : PlanManager)

ShellMenuEditCalendar

+ShellMenuEditCalendar(p: ShellPresenter, plans: PlanManager, cals: CalendarManager, users: UserManager, logs: LoggingManager, labs: LabelManager, hists: HistoryManager,

comms: CommentManager) + run(p : ShellPresenter) : void

back(logs : LoggingManager) : void

- viewSchedule(p : ShellPresenter, plans : PlanManager) : void

- viewScheduleByLabel(p: ShellPresenter, plans: PlanManager, labs: LabelManager): void

- viewUpcoming(p : ShellPresenter, plans : PlanManager) : void

· createCalendar(p : ShellPresenter, cals : CalendarManager, users : UserManager, logs :

LoggingManager, plans: PlanManager, hists: HistoryManager): void

 - deleteCalendar(p : ShellPresenter, cals : CalendarManager, users : UserManager, plans : PlanManager, labs: LabelManager, logs: LoggingManager, hists: HistoryManager, comms:

CommentManager): void

- clearCalendars(p : ShellPresenter, cals : CalendarManager, users : UserManager, plans : PlanManager, logs: LoggingManager, labs: LabelManager, hists: HistoryManager, comms: CommentManager): void

-editDescription(p : ShellPresenter, users : UserManager, cals : CalendarManager, hists :

HistoryManager. Logs: LoggingManager): void - editName(p : ShellPresenter, users : UserManager, cals : CalendarManager, hists :

HistoryManager, logs: LoggingManager): void

editSharePermission(p : ShellPresenter, users : UserManager, logs : LoggingManager) : void

- viewAllCalednars(p : ShellPresenter, cals : CalendarManager, users : UserManager, logs : LoggingManager): void

- viewSharedCalendars(p: ShellPresenter, cals: CalendarManager, users: UserManager, logs: LoggingManager): void

- editPlans(p : ShellPresenter, cals : CalendarManager, users : UserManager, logs :

loggingManager): void

ShellMenuUser

+ ShellMenuUser(p : ShellPresenter, s : InterfaceSaver, plans : PlanManager, cals : CalednarManager, logs : LoggingManager, labs :LabelManager, hists : HistoryManager)

+ run(p : ShellPresenter)

viewHistory(p : ShellPresenter, logs : LoggingManager, hists : HistoryManager)

logOut(p : ShellPresenter, s : InterfaceSaver, logs : LoggingManager, hists :

HistoryManager): void

deleteOwnAccount(p : ShellPresenter, logs : LoggingManager, hists : HistoryManager, cals : CalednarManager, plans : PlanManager, labs : LabelManager, users : UserManager,

comms : CommentManager) : void

editCalendars(): void

changeUsername(p : ShellPresenter, users : UserManager, logs : LoggingManager,

hists: HistoryManager): void

changePassword(p : ShellPresenter, users : UserManager, logs : LoggingManager,

hists: HistoryManager): void

save(p : ShellPresenter, cals : CalednarManager, plans : PlanManager, labs :

LabelManager, users: UserManager, hists: HistoryManager, comms:

CommentManager): void

ShellMenuUserAdmin

+ Shell Menu Admin User (p: Shell Presenter, s: Interface Saver, plans: Plan Manager, cals:

CalendarManager, logs: LoggingManager, labs: LabelManager, users: UserManager, hists:

HistoryManager, comms: CommentManager)

- + run(p : ShellPresenter) : void
- newAdminUser(p : ShellPresenter, users : UserManager, plans : PlanManager, cals : CalendarManager, logs : LoggingManager) : void
- banUser(p : ShellPresenter, users : UserManager, logs : LoggingManager, hists : HistoryManager) : void
- unbanUser(p : ShellPresenter, users : UserManager, logs : LoggingManager, hists : HistoryManager) : void
- deleteNonAdminAccount(p : ShellPresenter, users : UserManager, plans : PlanManager, cals :

CalendarManager, labs: LabelManager, hists: HistoryManager, logs: LoggingManagercomms:

CommentManager): void

<<abstract>> ShellMenu + commands : Map<String, Runnable> ShellMenuEditLabelEvent menu State : MenuState # setMenuState(menuState : MenuState) : void + ShellMenuEditLabel(p : ShellPresenter, s : # getMenuState(): MenuState InterfaceSaver, cals: CalendarManager, plans: # handelInput(data : ShellData, input : String) : void PlanManager, labs: LabelManager, logs: + run(presenter : ShellPresenter) : void LoggingManager, hists: HistoryManager) + run(p: ShellPresenter, s: InterfaceSaver, hists: HistoryManager, logs: LoggingManager, cals: CalendarManager): void back(p : ShellPresenter) : void editLabel(p : ShellPresenter, plans : PlanManager, labs: LabelManager): void - deleteLabel(p : ShellPresenter, plans : PlanManager, labs : LabelManager) : void clearLabels(p : ShellPresenter, plans : PlanManager, labs : LabelManager) : void - addThematicalLabel(p : ShellPresenter, plans : PlanManager, labs : LabelManager) : void - viewLabels(p : ShellPresenter, plans : PlanManager, labs : LabelManager) : void ShellMenuEditLabelTask + ShellMenuEditLabelTask(p : ShellPresenter, s : InterfaceSaver, cals : CalendarManager, plans: PlanManager, labs: LabelManager, logs: LoggingManager, hists: HistoryManager) + run(p : ShellPresenter) - setProgressLabel(p : ShellPresenter, plans : PlanManager, labs :

ShellMenuEditComment

+ ShellMenuEditComment(p : ShellPresenter, plans : PlanManager, comms :

LabelManager)

CommentManager, logs: LoggingManager)

- + run(p : ShellPresenter) : void
- back() : void
- viewComments(p: ShellPresenter, plans: PlanManager, comms: CommentManager): void
- addComment(p : ShellPresenter, logs : LoggingManager, plans : PlanManager, comms :

CommentManager) : void

- editComment(p : ShellPresenter, plans : PlanManager, comms : CommentManager) : void
- deleteComment(p : ShellPresenter, plans : PlanManager, comms : CommentManager) :
- reply(p : ShellPresenter, plans : PlanManager, comms : CommentManager, logs : LoggingManager) : void
- clearComments(p : ShellPresenter, plans : PlanManager, comms : CommentManager) : void

Presenter

ShellPresenter

+ getInput() : String

+ displayResult(result : String) : void

+ getArguments(allowEmpty : String, args : String...) : String[]

+ getArguments(allowEmpty : boolean, args : String...) : String[]

+ heyArguments(allowed : List<String>, args : String...)

+ prompter(firstLine : String, args : String...) : String

GateWay

<<Interface>> InterfaceSaver

+ changesMade(): boolean

+ makeChange() : void

+ saveAll(): void

+save(saveable : Serizable, fileName : String) :

void

+ load(item : T, fileName : String) : T



Saver

- changesMade : boolean

- path : String

+ save(saveable : Serializable, fileName : String) :

void

+ saveAll(): void

+ changesMade() : boolean + makeChange() : void

+ load(item : T, fileName : String) : T