StatusManager

- statuses: string array
- indicator: unsigned char
- -+ StatusManager()
- + increment()
- + setStatus(char)
- + getStatus(): string

DeadlineManager

- day: unsigned char
- month: unsigned char
- year: unsigned short
- + Deadline Manager (date: string)
- + ~Deadline Manager()
- + setDeadline(string)
- + getDeadline(): string

MandatoryProject

- addParticipant(&client: User)
- + MandatoryProject(&initiator: User, name: string, objective: string, tasks: string, subjectField: string, deadline: string, prerequisites: string)
- + findParticipants()
- + ~MandatoryProject()

OptionalProject

- addParticipant(&client: User)
- + Optional Project (&initiator: User, name: string, objective: string, tasks: string, subjectField: string, deadline: string, prerequisites: string)
- + findParticipants()
- + ~OptionalProject()

<abstract>>

Project

- # name: string
- # objective: string # tasks: string
- # status: StatusIndicator* # deadline: DeadlineManager*
- # subjectField: string
- # Prerequisites: string vector
- # initiator: User*
- # client: User* # manager: User*
- # participants: map<string, User*>
- # setStatus(status: string)
- # setClient(client: string)
- # setManager(&client: User)
- # setDeadline(deadline: string)
- # setName(name: string)
- # setObjective(objective: string)
- # setTasks(tasks: string)
- # setSubjectField(subjectField: string)
- # addPrerequisites(prerequisites:
- string)
- + findParticipants() = 0
- + changeInfo()
- + getInitiator(): User*
- + getClient(): string
- + getManager(): User*
- + getParticipants(): map<string,
- User*>
- + getSubjectField() : string
- + getName() : string
- + getObjective() : string
- + getTasks(): string
- + getDeadline() : string
- + getPrerequisites() : string vector
- + getStatus(): string
- + operator« (&out: ostream,
- &project: const Project Abstract):

ostream&

<<struct>> Notification

+ text : string + type : string + sender: string

+ receiver : string

<<static>>

RatingManager

-имя участника

project : string)

Database

Project*>

vector <User*>>

vector <User*>

vector < Project *>

+AddUser()

vector < Project *>>

+ static giveRating (user: string,

-Users : map <string, User*>

-Project : map <string, Project* >

-UsersPrerequisites map <string,

-ProjectPrerequisites map <string,

+FindUser(name : string) : User*

+FindUsers(Prerequisites: string):

+FindProject(name : string) : Project*

+FindProjects(Prerequisites: string):

+AddProject(&project : Project)

+DeleteProject(project : string)

+FinishProject(project : string)

+DeleteUser(user : string)

-FinishedProjects: map <string,

<<static>>

-имя участника

AuthorizationManager

+ <static> registerUser()

+ <static> authUser()

Notification Manager

- new_notifications : vector
- <Notification>
- +addNotification(notification:
- Notification)
- +displayNotifications()
- +reactNotification(notification:
- Notification)

User

- name : string
- password : string
- freetime : int
- currentProjects : map <Project*>
- finishedProjects: map <Project*>
- notifications : NotificationManager
- rating: vector <float>
- prerequisites : vector <string>
- StudyFields: vector <string>
- + User(name : string, password :
- string, free_time : int, prerequisites:
- string, StudyFields: string)
- setName(name : string)
- changeFreetime(time : int)
- addCurrentProjects(&project:
- Project):
- addFinishedProjects(&project :
- Proiect):
- addPrerequisites(prerequisite :
- string) - addStudyFields(StudyField: string)
- + addNewNotification(notification
- Notification) + getName() : string
- + getFreetime(): int
- + getRating() : float
- + getCurrentProjects():
- map<Project*>
- + getFinishedProjects():
- map<Project*>
- + getPrerequisites() : vector<string>
- + getStudyFields() : vector<string>
- + checkPassword(pass : string)
- + changeInfo()
- + checkNotifications()
- + checkNewProjects()
- + checkCurrentProjects()
- + displayFinishedProjects()
- + findNewProject()
- + applyToNewProjects(newProject :
- *Project)
- + workOnProject(activeProject:
- *Project)
- + collectRating(newRating : float)
- + createProject()
- + ~User()
- + operator<< (&out: ostream, &user:
- const User): ostream&