

User Methods:

```
createUser(name, email, password, userType)
    new User ← initialize all fields, empty lists
    return new User

displayUser()
    print name, email, userType

updateUser(newEmail, newProfilePic, newPreference)
    if newEmail set email
    if newProfilePic set profilePic
    if newPreference set preference

deleteUser()
    for each course in courseList course.removeMember(this)
    clear courseList and assignmentList

login(emailIn, passwordIn)
    if emailIn=email and passwordIn=password onlineStatus←true; return true
    else return false

logout()
    onlineStatus ← false
```

Course Methods:

```
createCourse(name, courseID, sectionNumber, professor, startDate)
    new Course ← set all fields, empty memberList
    return new Course

displayCourse()
    print name, courseID, sectionNumber, professor.name, startDate

editCourse(newName, newSection, newProfessor)
    if newName name←newName
    if newSection sectionNumber←newSection
    if newProfessor professor←newProfessor

deleteCourse()
    remove this from System.courses

addMember(user)
    if user not in memberList
        add user; user.courseList.add(this)

displayMembers()
    for each u in memberList print u.name
```

```
updateMember(user, newRole)
    if user in memberList user.preference["role"]←newRole
```

```
removeMember(user)
    if user in memberList remove user; user.courseList.remove(this)
```

Resource Methods:

```
createResource(name, course, isFromProfessor, text, website, attachment)
    new Resource ← set all fields; course.resources.add(this)
    return new Resource
```

```
displayResource()
    print name; if text...; if website...; if attachment...
```

```
editResource(newText, newWebsite, newAttachment)
    if newText text←newText; likewise website, attachment
```

```
deleteResource()
    course.resources.remove(this)
```

```
uploadContent(textContent)
    text ← textContent
```

```
removeContent(contentID)
    if attachment.id=contentID attachment←null
```

Message Methods:

```
createMessage(text, user, forum)
    new Message ← set text, user, forum, timestamp=now()
    return new Message
```

```
sendMessage()
    if forum forum.messageList.add(this)
```

```
displayMessage()
    print "[", timestamp, "]", user.name, ": ", text
```

```
editMessage(newText)
    if user is author text←newText
```

```
deleteMessage()
    if forum and this in forum.messageList forum.messageList.remove(this)
```

```
addReaction(type, by)
    if type in react react[type].addUnique(by) else react[type]=[by]
```

```
displayReactions()
```

```

    for each (type, users) in react print type, users.names

removeReaction(type, by)
    if type in react and by in react[type]
        react[type].remove(by)
        if react[type] empty remove key type

displayReply()
    for each r in reply r.displayMessage()

```

Forum Methods:

```

createForum(course)
    new Forum ← set course, empty memberList & messageList
    return new Forum

displayForum()
    print "Forum for ", course.name

editForum(newCourse)
    if newCourse course←newCourse

deleteForum()
    // remove references as needed

addMember(user)
    if user not in memberList memberList.add(user)

displayMembers()
    for each u in memberList print u.name

removeMember(user)
    memberList.remove(user)

displayMessageList()
    for each m in messageList m.displayMessage()

```

Calendar Methods:

```

createCalendar(courses)
    new Calendar ← set courseList, currentDate=today(), empty assignmentList
    return new Calendar

displayCalendar()
    print "Calendar (" , currentDate, ")"
    for each c in courseList print c.name

updateCalendar(newDate, newCourses)
    currentDate←newDate

```

```

    if newCourses courseList←newCourses

deleteCalendar()
    clear courseList and assignmentList

getCourse(courseID)
    return first c in courseList where c.courseID=courseID

getAssignment(name)
    return first a in assignmentList where a.name=name

displayAssignment(name)
    a←getAssignment(name); if a a.displayAssignment()

changeView(viewType)
    currentView←viewType

```

Assignment Methods:

```

createAssignment(course, name, description, dueDate, assigner)
    new Assignment ← set fields, completionStatus=false, empty feedback
    course.assignmentList.add(this)
    return new Assignment

displayAssignment()
    print name, dueDate, (completionStatus ? "Complete" : "Incomplete")

editAssignment(newDescription, newDueDate)
    if newDescription description←newDescription
    if newDueDate dueDate←newDueDate

deleteAssignment()
    course.assignmentList.remove(this)

assign(to)
    submitter←to; to.assignmentList.add(this)

unassign(from)
    if submitter=from submitter←null; from.assignmentList.remove(this)

uploadText(content)
    description←content

removeText()
    description←""

uploadFile(file)
    attachment←file

```

```
removeFile(fileID)
  if attachment.id=fileID attachment←null

submitWork(user, content)
  if user=submitter description←content; completionStatus=false

unsubmitWork()
  if not completionStatus description←""

submitGrade(gradeValue)
  grade←gradeValue

addFeedback(comment)
  feedback.add(comment)

displayFeedback()
  for each c in feedback print c

editFeedback(i, newComment)
  if i in range feedback[i]←newComment

removeFeedback(i)
  if i in range feedback.removeAt(i)

markCompleted()
  completionStatus←true
```