
























































































































 UserManager	
  UserManager()	
  getAttendee(String)	Attendee
  addAttendee(String, String)	void
  addAttendee(Attendee)	void
  getOrganizer(String)	Organizer
  addOrganizer(String, String)	void
  addOrganizer(Organizer)	void
  getSpeaker(String)	Speaker
  addSpeaker(String, String)	void
  addSpeaker(Speaker)	void
  canAddPerson(String)	boolean
  getUser(String)	User
  canAddSpeaker(LocalDateTime, List<Event>)	boolean
  findPosOfEvent(int, int, LocalDateTime, List<Event>)	int
  signUp(String, Event, List<Event>)	void
  canSignUp(LocalDateTime, List<Event>)	boolean
  cancelMyEvent(String, String)	boolean
  cancelWholeEvent(List<String>, String, String)	void
  login(String, String)	boolean
  changePassword(String, String)	void
 organizers	Hashtable<String, Organizer>
 attendees	Hashtable<String, Attendee>
 users	Hashtable<String, User>
 speakers	Hashtable<String, Speaker>

  EventManager	
  events	Hashtable<String, Event>
  EventManager()	
  EventManager(ArrayList<Event>)	
  isAtCapacity(String, int)	boolean
  doesOverlap(Event)	boolean
  getEvent(String)	Event
  containsEvent(String)	boolean
  getEventsByUsername(User)	ArrayList<Event>
  getEventsBySpeaker(String)	ArrayList<Event>
  addPersonToEvent(String, String)	void
  canAddPerson(String)	boolean
  removeAttendee(String, String)	boolean
  scheduleEvent(LocalDateTime, int, String, String, String)	void
  canScheduleEvent(LocalDateTime, int, String, String, String)	boolean
  removeEvent(String)	void
  canRemoveEvent(String)	boolean
  setSpeaker(String, String)	void
  withinHours(Event)	boolean
  getEventToString(String)	String
 events	ArrayList<Event>

  RoomManager	
  allRooms	Hashtable<String, Room>
  RoomManager()	
  getRoom(String)	Room
  addRoom(String, int)	boolean
  isRoomTaken(String, LocalDateTime)	boolean
  addEvent(String, String, LocalDateTime)	void
  removeRoomEvent(String, String)	void

  MessageManager	
  messageList	ArrayList<Message>
  sendMessage(User, User, String)	void
  getMessage(int)	Message
  getMyMessages(List<Integer>)	ArrayList<Message>

  ReadAndWrite	
  helperSerialize(String, Hashtable<String, User>)	void
  helperDeserialize(String)	Hashtable<String, User>
  save(UserManager)	void
  load(UserManager)	void