EventManager

EventManager()	
createSpeakerEvent(String, ArrayList<String>, int, String, int, int, int, int, int, int)	void
createSpeakerlessEvent(String, int, String, int, int, int, int, int, int)	void
createVIPSpeakerEvent(String, ArrayList<String>, int, String, int, int, int, int, int, int)	void
createVIPEvent(String, int, String, int, int, int, int, int, int)	void
freeSpeakerCheck(ArrayList<LocalDateTime>, String)	boolean
checkSpeakerConflict(LocalDateTime, LocalDateTime, ArrayList<LocalDateTime>)	boolean
freeTitleCheck(String)	boolean
getTitle(Event)	String
nameToEvent(String)	Optional<Event>
eventToAttendees(Event)	ArrayList<String>
signUp(Event, String)	void
dropOut(Event, String)	void
getSpeakers(SpeakerEvent)	ArrayList<String>
changeSpeaker(SpeakerEvent, String, String)	boolean
addSpeaker(SpeakerEvent, String)	boolean
removeSpeaker(SpeakerEvent, String)	boolean
getEventRoom(Event)	int
cancelEvent(Event)	void
getEventCapacity(Event)	int
setEventCapacity(Event, int)	void
getToString(Event)	String
speakerlessEvents	HashMap<String, Event>
speakerEvents	HashMap<String, SpeakerEvent>
allEventsSortedByTime	ArrayList<Event>
allEvents	HashMap<String, Event>
eventInfoLists	ArrayList<LinkedHashMap<String, String>>

UserManager

masterList	ArrayList<User>
organizerList	ArrayList<OrganizeAble>
UserManager()	
createAttendee(String, String, UserType)	User
registeredSpeaker(String)	boolean
inSystem(String, String)	boolean
addEventToSpeakerList(TalkAble, String)	void
removeEventFromSpeakerList(TalkAble, String)	void
getItinerary(AttendAble)	ArrayList<String>
getSpeakingList(TalkAble)	ArrayList<String>
changeSpeaker(String, String, String)	void
checkIsOrganizer(Account)	Boolean
hasVIPAccess(Account)	Boolean
getUsername(Account)	String
signUp(AttendAble, String)	void
dropOut(AttendAble, String)	void
usernameToUserObject(String)	Optional<User>
usernameToAttendeeObject(String)	Optional<AttendAble>
usernameToOrganizer(String)	Optional<OrganizeAble>
usernameToSpeakerObject(String)	Optional<TalkAble>
checkValidEmail(String, String)	short
setUserEmail(String, String)	void
setAttendeePassword(String, String)	void
getUserEmail(String)	String
allAttendees	ArrayList<AttendAble>
allSpeakers	ArrayList<TalkAble>

MessageManager

messages	ArrayList<Message>
messageNumStorage	ArrayList<String>
MessageManager()	
createMessage(ArrayList<String>, String, String)	int
deleteMessage(Message)	void
getArchiveMessages(String)	ArrayList<String>
getSendBy(String)	ArrayList<String>
getReceivedBy(String)	ArrayList<String>
getAllReceivedBy(String)	ArrayList<Message>
getUnreadMessage(String)	ArrayList<String>
getNumOfUnreadMessage(String)	int
getArchivedObj(String)	ArrayList<Message>
getAllMessagesFrom(String, String)	ArrayList<String>
getSenderAndRecipients(Message)	ArrayList<String>
getContent(Message)	String
recipientsBuilder(ArrayList<String>)	String
messageNumGenerator(Message)	int
numToMessageObject(String)	Optional<Message>

RoomManager

rooms	ArrayList<Room>
RoomManager()	
addRoom(int, int)	boolean
idToRoom(int)	Room
getCapacity(Room)	int
checkRoomInSystem(int)	boolean
checkIfRoomAvailable(Room, LocalDateTime, LocalDateTime)	boolean
book(Room, String, LocalDateTime, LocalDateTime)	void
checkConflict(LocalDateTime, LocalDateTime, LocalDateTime[])	boolean
unbook(Room, String)	void

Package MessageObserver