

© îs	ScheduleSetter		
m i	speakerSchedule(ActFactory, String, EventFactory)	void	
m 🖫	attendeeSchedule(ActFactory, String, EventFactory, String, Map < String >)	void	
m la	getAvailableEvent(RoomManager, String, String, Map <string, string="">, EventFactor</string,>	ory, ActFactory) void	
<u>,m ኈ</u>	getLikedEvents(String, String, Map <string, string="">, ActFactory, EventFactory)</string,>	void	
m 🖫	hasAvailableEvent(EventFactory, String, RoomManager, ActFactory)	boolean	
m ii	sortEvent(String, ArrayList <integer>, EventFactory)</integer>	ArrayList <integer></integer>	
m ii	availableEvent(RoomManager, String, EventFactory, ActFactory)	ArrayList <integer></integer>	
@ a	sortAndFilter(String, ArrayList <integer>, Map<string, string="">, EventFactory)</string,></integer>	ArrayList <integer></integer>	

C in	MessageDealer	
m 🚡	groupMessageTo(String, String, ActFactory, MessageManager, String)	void
m 🖫	eventMessage_Attendee(String, Integer, EventFactory, MessageManager, Strin	g) void
m 🚡	privateMessageTo(String, String, MessageManager, String)	void
m 🚡	getMessage(String, MessageManager, String, String)	void
m 🚡	markAsUnread(String, MessageManager, String, int, String)	void
m 🚡	deleteMessage(String, MessageManager, String, int, String)	void
m 🚡	archiveMessage(String, MessageManager, String, int)	void
m 🚡	hasMessage(String, MessageManager, String)	oolean

C i	EntityConstructors	
m ·	createRoom(int, ArrayList <string> , RoomManager)</string>	W
∭ 'n	createUser(String, String, String, ActFactory, MessageManager, String)	W
m la	createEvent(ArrayList <string>, String, String, int, int, String, ArrayList<string>, RoomManager, ActFactory, EventFactory, CalendarManager)</string></string>	book
m ·	validSpeaker(String, ActFactory)	book
m ·	validRoom(int, RoomManager)	book
m 's	hasSpeakers(ActFactory)	book
m 's	hasRoom(RoomManager)	book

© 'n EventDealer	
m 's signUp(int, ActFactory, EventFactory, String)	void
m '= cancelSpot(int, ActFactory, EventFactory, String)	void
🎮 🗽 cancelEvent(int, EventFactory, ActFactory, RoomManager, CalendarManager)	void
m 😘 likeEvent(int, ActFactory, String)	void
m ¹a hasEvent(EventFactory) bo	olear
m hasSignUp(String, ActFactory) bo	olear
m = qetAllEvents(EventFactory)	void

© is	Getter	
m i	speakerAvailable(ActFactory, String, CalendarManager)	void
@ °=	roomList(String, RoomManager, int, List <string>, Calenda</string>	arManager) void
m 🚡	hasAvailableRoom(RoomManager, int, List <string>)</string>	boolean
m in	getTopFiveEvents(EventFactory)	void
m in	getAppTraffic(EventFactory, RoomManager)	void
m 🚡	enrollmentStatistics(EventFactory, RoomManager)	void

C in	RequestDealer	
m 🖫	checkAllRequest(RequestManager)	void
m 🖫	checkMyRequest(String, RequestManager)	void
m 🖢	madeRequest(RequestManager, String, Str	ing) void
m 🖫	tagRequest(int, RequestManager)	void
m 🖢	hasRequest(RequestManager)	boolean
m 🖫	containRequest(RequestManager, int)	boolean

© 'n getSchedule	
m 1 getSchedule()	
m 🖫 getScheduleBySpeaker(String, ArrayList <integer>, EventFactory)</integer>	ArrayList <integer></integer>
m 🖫 getScheduleByDay(String, ArrayList <integer>, EventFactory)</integer>	ArrayList <integer></integer>
m = getScheduleByTime(String, ArrayList <integer>, EventFactory)</integer>	ArrayList <integer></integer>

C	9	UserPrinter	
m	'n	checkAudiences(Integer, EventFactory)	void
m	'n	checkContacts(MessageManager, String, String)	void
m	'n	checkUsers(ActFactory, String)	void
m	'n	changeUserAccess(String, ActFactory, String)	void