© ኈ ScheduleSystem													
f ≜ speakerManager	SpeakerManager												
● organizerManager	Organizer Manager					© ៕ UserSystem							
f ≜ eventManager	EventManager					f ≜ attendeeManager	AttendeeManager	© ኈ MessageSystem					
f ← roomManager	RoomManager					<ul><li>● organizerManager</li></ul>	Organizer Manager		AttendeeManager				
f ≜ attendeeManager	AttendeeManager					f ≜ speakerManager	SpeakerManager	f	SpeakerManager				
m   ■ ScheduleSystem(SpeakerManager, OrganizerManager, EventManager, RoomMan	nager, AttendeeManager)					f ≜ eventManager	EventManager	f ← messageManager	MessageManager				
m assignEvent(String, String, int, int, int, int, int, int, int)	String	C ← LoginSystem				f ≜ identification	Identification	f ← eventManager	EventManager	<pre>C ← SignUpSystem</pre>			
m assignVipEvent(String, String, int, int, int, int, int, int, int)	String	f ← attendeeManager	AttendeeManager			m 🖫 UserSystem(AttendeeManager, OrganizerM	lanager, SpeakerManager, EventManager)	f	UserRequestManager		A + +	© ኈ Registrar	
bookingNormalSpeaker(String, String)	String	f ← organizerManager	OrganizerManager			m = attendeeShowEventsRegistered(String)	String	m 🖢 MessageSystem(AttendeeManager, SpeakerManager,		f attendeeManager	AttendeeManager		A + +   > 4 >
m ₁ cancelEvent(String, String)	String	f	SpeakerManager	© ኈ VipSystem		m 🚡 attendeeEventsTotal(String)			void	f a eventManager	EventManager		AttendeeMana
m availableSpeakerId(int, int, int)	ArrayList < String >	f ← identification	Identification		A., 1 A4	m 🖫 isRegistered(String, String)			void	f a speakerManager	SpeakerManager		OrganizerMana
	String		anager, SpeakerManager)		er AttendeeManager	m	ArrayList < String >		void	f a organizerManager	OrganizerManager	● speakerManager	SpeakerManag
m ¹ getValidRoomId(int, int, int, int)	ArrayList < String >		boolean	m WipSystem(Atte		m = getUserFriendlist(String)			void		ger, EventManager, SpeakerManager, OrganizerManager)		MessageMana
m = printRoomIds(ArrayList < String >)		m ← checkPassword(String, String)	Boolean	, . =	(String) boolean	m 🚡 isFriend(String, String)		m	ArrayList < String >	m = signUp(String, String)		m 👚 Registrar(AttendeeManager, Organize	Manager, SpeakerManager, MessageManage
m ' checkEventOwner(String, String)		m ¹ logIn(String, String)	boolean	m modifyVip_statu	s(String, boolean) void	m 🖢 organizerEventHold(String)		m ≥ viewInbox(String)	String	m 🚡 cancel(String, String)		m 🔓 registerAttendee(String, String)	Str
m ¹ createRoom(int)		m = setSecurityQuestion(String, String, String)	void	1		m 🖢 organizerEventsTotal(String)		m wiewOutbox(String)	String	m 🚡 isValid(String, String)		m 🔓 registerOrganizer(String, String)	Str
m hasEvent(String)		m & checkSecurityQues(String, String, String)	boolean	į				m ← checkNumAndSet(String, String)	String	m wiewAvailableEvent(String)		m = registerSpeaker(String, String)	Str
m _ checkValidNewMax(int, String)		m ₁ resetPassword(String, String)	void			m = speakerEventsTotal(String)			String	m • checkDoubleSignUp(String, Stri	ring) boolean		1
m = resetEventMaxAttendee(String, int)	void	<b>1</b>	^			m = speakerEvents(String)			String		1	į	
m validYear(String)	String	1				m = getSchedule(String)		m ≥ sendUserRequest(String, String)	void			į	
m validMonth(String, String)	String					m  m getEventTotal(String)		m addressUserRequest(int)	void	į			
m validDay(String, String)	String						<u>^</u>	m 🕆 listAllUserRequests()	void				
m validHour(String, String, String)	String	į į	İ	İ		1		^	^				
m • checkDuration(String, String)	String	į į		į			į	1					
m hasSpeaker(String)	boolean						į						į
^ ^	<u>^</u>						[					į	
į	į				i	«cre at <sub>le»</sub> «cre at <sub>le»</sub> «cre at <sub>le»</sub> «cre at <sub>le»</sub>	'					į	
	į			L									
		<u> </u>			«create»			«create»					
					<u></u>				«сте	ate»		<sup> </sup>	
L		«create»				<u>-</u>	1 1						
						ControllerSystem							
						f o attendeeManager	AttendeeManager						
						f o organizerManager	OrganizerManager						
						f • speakerManager	SpeakerManager						
						f • messageManager	MessageManager						
						f o roomManager	RoomManager						
						f • eventManager	EventManager						
						f • userRequestManager Us	serRequestManager						
						f a messageSystem	MessageSystem						
						f a scheduleSystem	ScheduleSystem						
							SignUpSystem						
						<ul><li>f ≜ signUpSystem</li><li>f ≜ registrar</li></ul>	Registrar						
						6 A logistian	LoginCustom						

LoginSystem

UserSystem

SaverReader

VipSystem

LoginSystem

MessageSystem

ScheduleSystem

SignUpSystem

Registrar

UserSystem

VipSystem

void

f ≜ loginSystem

f ≜ vipSystem

m 🚡 loginSystem()

m 🚡 ControllerSystem()

m messageSystem()

m 🚡 scheduleSystem()

m 🚡 signUpSystem()

m 🔓 registrar()

m userSystem()

m 🚡 vipSystem()

m = exist()

Powered by yFiles