TimeRange

TimeRange(LocalDateTime, LocalDateTime)

hasOverlap(TimeRange) boolean CSC207 group_0011 Calendar
uuidToTimeRange Map<UUID, TimeRange> getUUIDToTimeRange() Map<UUID, TimeRange> UML Diagram Phase 2 getUUIDtoTimeRanges()

hasConflict(TimeRange)

addTimeBlock(UUID, TimeRange)

removeTimeBlock(UUID)

getBooking(UUID)

TimeRange Diagram 1 Expanded Backend Diagram Event(String, TimeRange, UUID, Set<UUID>)

addSpeaker(UUID) void
removeSpeaker(UUID) void
isSpeaker(UUID) boolean
addAttendee(UUID) void
removeAttendee(UUID) void Conference
Conference(String, TimeRange, UUID)
isOrganizer(UUID) boolean
addOrganizer(UUID) boolean
isSpeaker(UUID) boolean
addSpeaker(UUID) boolean
addSpeaker(UUID) void
removeSpeaker(UUID) boolean
isAttendee(UUID) boolean
addAttendee(UUID) boolean
addAttendee(UUID) boolean
addAttendee(UUID) boolean
porganizerUUIDs Set<UUID>
attendeeUUIDs Set<UUID>
conferenceName String
pevents Map<UUID, Event>
timeRange
prooms Map<UUID, Room>
puuID
uuid
speakerUUIDs Set<UUID>
speakerUUIDs Set<UUID> EventManager

events Map<UUID, Event>

EventManager(Map<UUID, Event>)

eventExists(UUID) boolean

validateEventTitle(String) boolean

getEvent(UUID) Event

createEvent(String, TimeRange, UUID, Set<UUID>) UUID

deleteEvent(UUID) void

getEventSpeakers(UUID, UUID) void

removeEventSpeaker(UUID, UUID) void

setEventTitle(UUID, String) void

setEventTitle(UUID, UUID) void

setEventTimeRange(UUID, TimeRange) void

getEventTitle(UUID) String

getEventTitle(UUID) UUID

getEventTitle(UUID) String

getEventTitle(UUID) String

getEventTitle(UUID) String

getEventTitle(UUID) String

getEventTimeRange(UUID, TimeRange) void

getEventTimeRange(UUID) TimeRange

getEventConversationUUID(UUID) UUID

setEventConversationUUID(UUID) void

getEventAttendees(UUID) Set<UUID>

registerAttendees(UUID) void UserManager

getUserFullName(UUID)

setUserFirstName(UUID, String)

getUserIstName(UUID, String)

getUserLastName(UUID)

setUserLastName(UUID)

setUserUsername(UUID)

getUserUsername(UUID)

getUserIsGod(UUID)

setUserPassword(UUID)

getUserIsGod(UUID)

volume

getUser(UUID)

getUser(UUID)

user

getUser(UUID)

user

getUser(UUID)

user

registerUser(String, String, String, boolean, boolean)

UUID RoomManager

RoomManager(Map<UUID, Room>)

RoomManager(Map<UUID, Room>)

validateRoomLocation(String) boolean
validateRoomCapacity(int) boolean
getCalendarManager(UUID) CalendarManager
roomExists(UUID) boolean
getRoom(UUID) Room
createRoom(String, int) UUID
setRoomLocation(UUID, String) void
setRoomCapacity(UUID, int) void
deleteRoom(UUID) String
getRoomLocation(UUID) string
getRoomCapacity(UUID) int
prooms
Set<UUID> content String
timestamp LocalDateTime
senderUUID UUID
Message(UUID, String)
getSenderUUID() UUID
getContent() String
getTimestamp() LocalDateTime
toString() String registerUser(String, String, String)
UUID loadGodUsers(List<String[]>) login(String, String)
clearCurrentUser() registerAttendee(UUID, UUID)

unregisterAttendee(UUID, UUID)

void

void

void

void

set<UUID> Conversation(String, Set<UUID>, Set<UUID>, List<Message>)

grantWriteAccess(UUID) void

grantReadAccess(UUID) void

removeUserFromWrite(UUID) void getRoomManager(UUID) RoomManager createConference(String, TimeRange, UUID) setTimeRange(UUID, TimeRange)
void setConferenceName(UUID, String)
void isAffiliated(UUID, UUID, UserManager) boolean isOrganizer(UUID, UUID, UserManager) boolean addAttendee(UUID, UUID) setSpeakers(UUID, Set<UUID>) void
addSpeaker(UUID, UUID) void
removeSpeaker(UUID, UUID) void

conferences Set<UUID> eateConversation(String, Set<UUID>, Set<UUID>, UUID, String) UUID getConversation(UUID)
addConversationToUserList(UUID, UUID)
removeConversationFromUserList(UUID, UUID)
deleteConversation(UUID) SELF_OR_ORGANIZER
 PermissionManager(ConferenceManager, UserManager)
 generateExecutorAccessDeniedError(UUID, UUID, String)
 generateTargetAccessDeniedError(UUID, UUID, UUID, String) getConversationList(UUID) testlsOrganizer(UUID, UUID) sendMessage(UUID, String, UUID, boolean) testIsSpeaker(UUID, UUID) testIsAttendee(UUID, UUID) getMessages(UUID, UUID, boolean) List<Map<String, String>> testTargetIsAttendee(UUID, UUID, UUID) void testTargetsAreAttendee(UUID, UUID, Set<UUID>) testIsAttendeeSelfOrAdmin(UUID, UUID, UUID) void getDayEvents(UUID, LocalDate) registerForEvent(UUID, UUID, UUID, UUID) doUnregisterForEvent(UUID, UUID, UUID) GhostAcceptDeniedException
GhostAcceptDeniedException
GhostAcceptDeniedException(UUID, UUID)

GhostDeleteException
GhostDeleteException(UUID, UUID)
GhostDeleteException
GhostDeleteException(UUID, UUID)
GetContactManager()
GetCont unregisterForEvent(UUID, UUID, UUID, UUID) dpdateSpeakers(OOID)
 speakerTimeRangeOccupied(UUID, UUID, TimeRange)
 testSpeakersTimeRangeOccupied(UUID, Set<UUID>, TimeRange)
 isEventDuringConference(UUID, TimeRange)
 createEvent(UUID, UUID, String, TimeRange, UUID, Set<UUID>)
 UUID TableTools

table List<List<String>>
TableTools(List<List<String>>)

stringifyTable(String) String

TableTools(ListString) String addEventSpeaker(UUID, UUID, UUID, UUID) oveEventSpeaker(UUID, UUID, UUID, UUID) deleteEvent(UUID, UUID, UUID) setEventTitle(UUID, UUID, UUID, String) setEventRoom(UUID, UUID, UUID, UUID) setEventTimeRange(UUID, UUID, UUID, TimeRange) getEventRoom(UUID, UUID, UUID) etEventConversationUUID(UUID, UUID, UUID) isRegistered(UUID, UUID, UUID) getEventTitle(UUID, UUID, UUID) getEventSpeakers(UUID, UUID, UUID) etEventAttendees(UUID, UUID, UUID) getNumRegistered(UUID, UUID, UUID)

updateEventConversationMembers(UUID, UUID)

compileEventConversationMembers(UUID, UUID)

createEventConversation(UUID, UUID, UUID)

UUID Logger
UserManager
ConversationManager
EventController
ConferenceManager
PermissionManager
(ConversationManager, EventController, ConferenceManager, UserManager)
UUID)
Set<UUID>
Ces(UUID)
Ces(UUID)
D)
Doolean
UUID)
String
Conge(UUID)
String
ConversationManager, EventController, ConferenceManager, UserManager)
UUID
ConversationManager, EventController, ConferenceManager, UserManager)
UUID
ConversationManager
ConversationManager
ConversationManager
ConversationManager
ConversationManager
ConversationManager
ConversationManager
ConversationManager
ConversationManager
ConferenceManager
ConversationManager
ConferenceManager
ConversationManager
ConferenceManager
ConversationManager
ConferenceManager
ConversationManager
ConferenceManager
ConversationManager
Conve ConversationController

conversationManager

contactManager

contactManager

UserManager

ConversationController(ContactManager, ConversationManager, UserManager)

checkAccess(UUID, UUID)

sendMessage(UUID, String, UUID)

initiateConversation(String, UUID, Set<UUID>, String)

getConversationName(UUID)

set<UUID>

addUserToConversation(UUID, UUID)

addUserToConversation(UUID, UUID)

conversationManager

ConversationManager

UserManager

UserManager

UserManager

UserManager

UserManager

UserManager

List<Manager

Void getUserConferences(UUID)

getNotUserConferences(UUID)

conferenceExists(UUID)

getConferenceName(UUID)

getConferenceTimeRange(UUID)

createConference(String, TimeRange, UUID)

setConferenceTimeRange(UUID, UUID, TimeRange)

setConferenceName(UUID, UUID, String)

deleteConference(UUID, UUID)

addAttendee(UUID, UUID) ContactController

contactManager ContactManager

ContactController(ContactManager)

sendRequest(UUID, UUID) void

deleteRequest(UUID, UUID) void

acceptRequest(UUID, UUID) void

rejectRequest(UUID, UUID) void

deleteContacts(UUID, UUID) void

deleteContacts(UUID, UUID) void title

documentPrinter

userManager

conferenceManager

eventController

ScheduleController(IDocumentPrinter, UserManager, ConferenceManager, EventController)

compileSchedule(List<Pair<UUID, UUID>>)

aetReaisteredEventPairs(UUID)

List<Pair<UUID. UUID>> ■ setRoomLocation(UUID, UUID, UUID, String) void deleteRoom(UUID. UUID. UUID)