Event (Entity)

Responsibilities:

* Store time, title, and speaker for the event, (number of attendees?)

Collaborators:

* EventManager

LoginManager (Use Case)

Responsibilities:

* Determines if a given username and password corresponds with an account

Collaborators:

* ConferenceSystem
* Connection to some gateway to account credential storage?

MessageSender (Use Case)

Responsibilities:

* Sends messages, and their sender to AttendeeUsers
* Checks if a sent message is valid (possible filters, recipient is valid)

Collaborators:

* ConferenceSystem
* AttendeeUser
* OrganizerUser

EventManager (Use Case)

Responsibilities:

* Store list of events
* Ability to add events to the list, also checks if events can be added (no time conflicts)
* Create and add events to user’s signup list
* Return a list of all events

Collaborators:

* Event
* ConferenceSystem

AttendeeUser (Use Case)

Responsibilities:

* Store events signed up for in a “signup list”
* Store user’s account credentials
* Store user’s network (other Attendees)
* Request to sign up for and cancel their spots in events
* Request to see all events
* Store other users (attendees) they can message
* Stores messages in a list received by other users
* Indicator of whether or not its an organizer

Collaborators:

* EventManager
* Superclass of OrganizerUser
* ConferenceSystem

OrganizerUser (Use Case)

Responsibilities:

* Request to add events to the list of events
* Request to send messages to all Attendees

Collaborators:

* EventManager
* Subclass of AttendeeUser

ConferenceSystem (main program, Controller)

Responsibilities:

* Stores list of AttendeeUsers (and subclass), Event Manager, Message Sender, Login Manager
* Instantizes EventManager and reads in all events from the text file
* Takes user input to login, send messages, add/remove events, sign up for events etc.