Group 2 Software Development Team

Thomas Behner

Cameron Haught

Jacob Forte

**Use Case Specifications**

**Use case name:** Create Account.

**Use case purpose:** For businesses and guests to be able to create/register conferences, the user must first create an account.

**Actor:** User

**Optimistic flow:**

1. Actor wants to make an account.
2. Actor chooses to create an account as a business or a guest
3. Actor enters required credentials
   1. For a business:
      1. Business Name
      2. Owner
      3. Payment info
   2. For a guest:
      1. Name
      2. Payment info
4. System checks credentials.
5. Actor leaves system

**Pragmatic flow:**

**Conditions triggering alternate flow:**

**Condition 1**: The business credentials entered are invalid

D1. Alert actor regarding invalid credentials

D2. Return to step Ca

**Condition 2:** The guest credential entered are invalid

D1. Alert actor regarding invalid credentials

D2. Return to step Cb

**Use case name:** Login.

**Use case purpose:** Used to access the appropriate interface for the user

**Actor:** User

**Optimistic flow:**

1. Actor wants to access an account.
2. Actor enters username and password
3. System Checks username and password
4. Actor gets forwarded to the appropriate interface

**Pragmatic flow:**

**Conditions triggering alternate flow:**

**Condition 1**: Username or password is incorrect

C1. System informs Actor that username or password is incorrect

C2. Redirect to B

**Use case name:** Register a conference

**Use case purpose:** This use case provides the information that makes up a conference to be registered into the UCCS. This use case allows the actor to request rooms and equipment needed.

**Actor:** Conference Organizer (or a representative)

**Optimistic flow:**

1. Actor wants to hold a conference at the UCC, accesses UCCS. # should we make Conference

# host accounts?

1. Actor enters desired date range of conference
   1. System checks availability of dates desired #Schedule
      1. System returns available dates desired
2. Actor enters info regarding needs of conference # times for sessions is required here too
   1. System checks availability of resources requested #Rooms, Equipment
      1. System returns available resources for dates desired # resources and

# time slots

1. Actor registers conference with desired available settings in date range
   1. System reserves selected resources for date range
      1. System calculates estimated cost based on Actor criteria, request payment
         * 1. System creates a conference in schedule

Sessions are created inside the conference

System grants Actor temporary **Conference Host** credentials to modify conference settings

1. Actor leaves system

**Pragmatic flow:**

**Conditions triggering alternate flow:**

**Condition 1**: The dates selected by the Actor are currently unavailable

B1. Alert actor regarding unavailable dates to meet request

Show dates available

B2. Prompt Actor to choose new date range or to try again later

B3. If new dates are selected, perform step B

B4. If Actor decides to try again later, perform step E

**Condition 2:** The resources selected by the Actor are currently unavailable

C1. Alert actor regarding unavailable resources to meet request

Show resources available

C2. Prompt Actor to choose new resource selection or to try again later

C3. If new resources are selected, perform step C

C4. If Actor decides to try again later, perform step E

**Use case name:** Register a guest into a conference

**Use case purpose:** This use case provides the guests of a hosted conference to register attendance.

**Actor:** Conference attendee (or a representative)

**Optimistic flow:**

1. Actor wants to attend a conference at the UCC, accesses UCCS. # should we make user # accounts?
2. Actor enters desired conference to attend # should we consider conference # invitations/permissions?
   1. System checks availability/capacity of conference desired
3. Actor registers conference attendance
   1. Actor selects sessions to attend
      1. System records attendance data
         1. System calculates estimated cost based on Actor criteria, request payment
         2. Systems grants user access to sessions.
4. Actor leaves system

**Pragmatic flow:**

**Conditions triggering alternate flow:**

**Condition 1**: The conference selected by the Actor are currently unavailable

B1. Alert actor regarding unavailable conference to meet request

Show conferences available

B2. Prompt Actor to choose new conference or to try again later

B3. If new conference is selected, perform step B

B4. If Actor decides to try again later, perform step D

**Condition 2:** The sessions(s) selected by the Actor is/are currently unavailable

C1. Alert actor regarding unavailable sessions to meet request

Show sessions available

C2. Prompt Actor to choose new session selection or to try again later

C3. If new sessions are selected, perform step C

C4. If Actor decides to try again later, perform step D

**Condition 3:** The Actor chooses to register for a special session

**Cai1a the actor makes payment, then goes to Cai2**

**Cai1b if no payment, system informs Actor they must pay to register for special sessions**

**Cai1c Actor goes to step Cai1**

**Use case name:** Manually manipulate resource availability

**Use case purpose:** This use case provides the UCC to override resource availability to conferences (if a piece of equipment needs repaired or discarded from the list, or a room needs upkeep, etc.. Conversely, if the actor wants to add new resources to the resource pool)

**Actor:** UCC technician or administrator

**Optimistic flow:**

1. Actor needs to add/remove resource(s) from available pool, accesses UCCS Admin Mode
   1. Actor enters **Admin** credentials
      1. Logs Actor into UCCS Admin Mode

# Assuming there WILL be some sort of administration access

1. Actor enters desired resource to add # Actor can only add equipment resources, NOT new # rooms is assumed
   1. System prompts actor to name new resource
      1. System checks name availability
   2. System prompts actor with required fields for resource attributes
      1. System adds resource to available resource pool
2. Actor enters desired resource to remove
   1. System checks resource availability
   2. Actor sets resource availability to “Unavailable”
      1. Actor selects condition to make resource available again

# Could be a certain time-frame, or other type of flag that re-enters it into the

# system

* + 1. System records condition data
       1. Resource becomes available when conditions are met

1. Actor leaves system

**Pragmatic flow:**

**Conditions triggering alternate flow:**

**Condition 1**: The Actor enters invalid **Admin** credentials

A1. Prompt actor to re-enter **Admin** credentials or abort

A2. If Actor enters valid credentials, perform step A.a.i.

A3. If Actor chooses to abort, perform step D

**Condition 2**: The new resource name chosen by the Actor already exists

B1. Prompt Actor to choose new resource name or to abort

B2. If actor chooses an unused resource name, perform step B.b.

B3. If actor chooses to abort, perform step D

**Condition 3**: The resource selected by the Actor are already currently unavailable

C1. Alert actor regarding unavailable resource to meet remove request

Show resources available to remove

C2. Prompt Actor to choose new resource or to try again later

C3. If new resource is selected, perform step C

C4. If Actor decides to try again later, perform step D

**Use case name:** Conference Organizer - Modify/withdraw from a conference

**Use case purpose:** This use case provides a conference organizer to modify resources or withdraw from a hosted conference

**Actor:** Conference Organizer (or a representative)

**Optimistic flow:**

1. Actor wants to modify or withdraw from a conference at the UCC, accesses UCCS.
   1. Actor enters **Conference Host** credentials
      1. Logs Actor into UCCS Conference Host Mode
2. Actor chooses to add/modify conference settings (resources)
   1. System checks availability of resources
      1. System modifies pool of available resources
         1. If new resource cost greater than original cost, modification invoice issued, else prorated refund issued.
   2. Actor chooses to cancel hosted conference
      1. Conference resources are released back into the system resource availability pool.
      2. Actor receives refund (prorated as conference date approaches)
      3. Actor **Conference Host** credentials are revoked
3. Actor leaves system

**Pragmatic flow:**

**Conditions triggering alternate flow:**

**Condition 1**: Actor enters invalid credentials

B1. Prompt actor to re-enter **Conference Host** credentials or abort

B2. If Actor enters valid credentials, perform step B.

B3. If Actor chooses to abort, perform step C.

**Use case name:** Conference Attendee - Modify/withdraw from a conference

**Use case purpose:** This use case provides a conference attendee to modify session attendance or withdraw from a session(s)

**Actor:** Conference Attendee

**Optimistic flow:**

1. Actor wants to modify or withdraw from a session(s) at the UCC, accesses UCCS.
2. Actor enters **User** credentials for conference
3. Actor chooses to modify attendance settings
   1. If user wants to add sessions, see:

**Use Case Name**: “Register a guest into a conference"

* 1. Actor chooses to withdraw attendance from specific session
     1. System checks if Actor is attending selected session
        1. System removes attendee from session attendance roster
           1. System modifies session attendance total
           2. Actor receives refund (prorated as conference date approaches)
           3. Actor’s **User** credentials are revoked for specific session

1. Actor leaves system

**Pragmatic flow:**

**Conditions triggering alternate flow:**

**Condition 1**: Actor enters invalid credentials

B1. Prompt actor to re-enter **User** credentials or abort

B2. If Actor enters valid credentials, perform step C.

B3. If Actor chooses to abort, perform step D

# Potential other cases, based on development (or if any of you guys have spare time):

# **Payment**, which could be performed online through the system, or be sent an invoice to

# pay in person