

## Use Case #1: Query Events for University

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/24/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 1 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Client User/Guest: Someone looking for nearby events
<b>Secondary Actors</b>	Universities
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• Guest: Wants to find nearby events without having to log in. Might be visiting the university they are currently at, or someone living near a college campus.</li><li>• User: Wants to find nearby events by logging in. A user will be looking for events at his or her own college campus.</li><li>• University: Has the ability to monitor events, and possibly shut down malicious or inappropriate ones.</li></ul>
<b>Preconditions</b>	For guests: nothing For client users: already authenticated
<b>Success Post Condition</b>	All events at the current university are sent to the client
<b>Failed Post Condition</b>	An error status is sent to the client

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User opens the client app, or goes to the discover tab
2	Client sends the user's location and requests a list of nearby universities
3	Backend sends the client a list of nearby universities
4	Client requests all the events at the closest university
5	Backend sends events at the specific university to the client

### EXTENSIONS or Alternate Flows

Step	Branching Action
4a	User selects a specific university they want to see events for <ul style="list-style-type: none"><li>1. Client requests all the events at the user-specified university</li><li>2. Backend sends events at the specific university to the client</li></ul>
5a	Client requested events for a university that does not exist <ul style="list-style-type: none"><li>1. Backend sends an error status to the client</li></ul>

### ***SPECIAL REQUIREMENTS***

<b>Req Num</b>	<b>Requirement</b>
--------------------	--------------------

1	Backend should gracefully handle bad queries (invalid location, nonexistent university, etc)
---	--

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

<b>Var Num</b>	<b>Variation</b>
--------------------	------------------

0	N/A
---	-----

***FREQUENCY OF OCCURRENCE:*** Very often, since this is the main functionality of the service

### ***OTHER ISSUES***

<b>Issue Num</b>	<b>Issue</b>
----------------------	--------------

1	How many events to be returned; either by amount or date?
---	---

## Use Case #2: Search for University Events by Text

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 1 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Client User or Guest
<b>Secondary Actors</b>	Event Admin User
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• User/Guest: Want to search for nearby events by typing keywords specific to the event</li><li>• Event Admin User: Provides event details for user/guest to search by, and wants users to attend events</li></ul>
<b>Preconditions</b>	For guests: nothing For client users: already authenticated
<b>Success Post Condition</b>	Events containing the query text in its fields (such as name, location, etc) are sent to the client
<b>Failed Post Condition</b>	An error status is sent to the client

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User arrives at 'Search' page
2	User types in certain key phrases about the event (name, location, description)
3	Backend will send a list of events to the client based on the key phrases provided by the user

### EXTENSIONS or Alternate Flows

Step	Branching Action
3a	The backend could not find any results for the search queries <ul style="list-style-type: none"><li>1. Backend sends message to client that no events were found</li><li>2. Client displays message that search produced no results</li></ul>

### ***SPECIAL REQUIREMENTS***

<b>Req</b>	<b>Requirement</b>
------------	--------------------

- |   |  |
|---|--|
| 1 | Search by text must be efficient at searching through many events, so users don't have to wait for a long time |
|---|--|

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

<b>Var</b>	<b>Variation</b>
------------	------------------

- |   |   |
|---|---|
| 1 | Text search should cover many event details, such as name or location |
|---|---|

***FREQUENCY OF OCCURRENCE:*** This will be used quite frequently. We are assuming that users in general would want to search for a specific event quite frequently in order to fully use the app to its potential.

### ***OTHER ISSUES***

<b>Issue</b>	<b>Issue</b>
--------------	--------------

- |   |   |
|---|---|
| 1 | What other details can the user search the location by and would it make a significant difference in the app? |
| 2 | How accurate should the text search be? For instance, should it also search for synonyms and plurals?         |

## Use Case #3 - Filter University Events by Categories

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 1 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Client User/Guest
<b>Secondary Actors</b>	Event Admin User
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• User/Guest: Wants to search for nearby events by clicking on one or more categories</li><li>• Event Admin User: Provides event details for user/guest to search by, and wants users to attend events</li></ul>
<b>Preconditions</b>	Categories are established and events have "tags" representing the categories
<b>Success Post Condition</b>	Future events in the selected category (or categories) will be sent to the client
<b>Failed Post Condition</b>	An error status is sent to the client

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User arrives at 'Discover' page
2	User selects categories to filter by (one or more, inclusive of sports, academic, 'Organizations I'm Following,' music, Greek life, etc.)
3	Backend queries future events relative to the categories selected
4	Client displays the returned events

### EXTENSIONS or Alternate Flows

Step	Branching Action
3a	There are no events/organizations related to the user-selected categories <ul style="list-style-type: none"><li>1. Backend indicates that there were no events found</li><li>2. Client shows a message for no events found</li><li>3. User can deselect categories in order to expand the search, or view all events</li></ul>

### SPECIAL REQUIREMENTS

Req Num	Requirement
1	Querying by filters must be efficient at searching through many events, so users don't have to wait for a long time

### **TECHNOLOGY AND DATA VARIATIONS LIST**

Var Num	Variation
0	N/A

**FREQUENCY OF OCCURRENCE:** Frequently because we assume that users will want to easily and efficiently look for events that are happening by a specific category.

### **OTHER ISSUES**

Issue Num	Issue
1	How should past events be handled?
2	What specific categories are needed?

## Use Case # 4: Query Universities by Location

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 2 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Client User/Guest: Operates automatically on behalf of the user
<b>Secondary Actors</b>	Database
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• Guest/Client User: Wants to see events at closest university, or if picking another one, then would want the universities to be sorted by distance</li><li>• University: Wants to be on the list so people have the option of using this service at that university</li></ul>
<b>Preconditions</b>	Client App has access to user's location, and universities in database have accurate location information
<b>Success Post Condition</b>	Universities near the provided location are sent to the client, sorted by distance
<b>Failed Post Condition</b>	Backend sends error status to client

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User opens client app, or wants to view events at a different university
2	Client app requests location of user
3	Client app sends user location to backend
4	Backend retrieves and sends universities closest to that location
5	Client app displays list of nearby universities, or selects closest one and requests events

### EXTENSIONS or Alternate Flows

Step	Branching Action
2a	User denies location request <ul style="list-style-type: none"><li>1. Client app will attempt to select last university used</li><li>2. If no university was used last (e.g. opening app for first time), client app requests alphabetical list of universities</li><li>3. Backend sends list of all universities, sorted alphabetically</li></ul>
3a	Client sends invalid location to backend <ul style="list-style-type: none"><li>1. Backend sends error message that location is invalid</li></ul>

### ***SPECIAL REQUIREMENTS***

<b>Req Num</b>	<b>Requirement</b>
--------------------	--------------------

1	List of universities should be retrieved fast, to avoid a long loading screen on the frontend
---	---

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

<b>Var Num</b>	<b>Variation</b>
--------------------	------------------

0	N/A
---	-----

***FREQUENCY OF OCCURRENCE:*** This will happen occasionally -- when the user first opens the app and when the user changes university

### ***OTHER ISSUES***

<b>Issue Num</b>	<b>Issue</b>
----------------------	--------------

1	How many universities should be returned, either by amount or by distance?
---	--



## Use Case #5: User Account Creation

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/2016
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 3 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Guest
<b>Secondary Actors</b>	Database
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• Guest: Wants to create account in order to follow events and organizations, and get updates for them</li><li>• Event Admin User: Wants users to attend events in order to keep track of attendees, as well as ensure everyone is kept up to date</li></ul>
<b>Preconditions</b>	Guest belongs to a university (has a university domain), university information exists in database
<b>Success Post Condition</b>	User account is successfully created, user information is written to the database, email address is verified
<b>Failed Post Condition</b>	An error message is sent the client indicating the failure (insecure password, email issue, etc)

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User arrives at 'Create Account' screen
2	User enters an email address for which an account does not already exist
3	User enters name
4	User enters a password that meets the security requirements
5	User re-enters a confirmation password that matches the previously entered password
6	User selects university from list of universities
7	User enters a valid email address
8	Password is hashed and sent to the server, along with the name, university, and email
9	Account details are stored in database
10	Verification email is sent to user email address
11	User clicks link in verification email
12	User account is verified on the backend and user is now allowed to log in

### EXTENSIONS or Alternate Flows

Step	Branching Action
2a	User enters an email address that exists in the database <ul style="list-style-type: none"><li>1. Backend sends error message that email address is taken</li></ul>

- 4a User enters a password that does not meet the security requirements
  - 1. Backend sends an error message that password does not meet security requirements
- 5a User enters a confirmation password that does not match the previously entered password
  - 1. Backend sends error message that passwords do not match
- 7a User email does not exist
  - 1. Backend tries to send verification email
  - 2. Verification email bounces back
  - 3. User information in database gets removed after some time
- 7b An error occurs while sending the verification email
  - 1. Backend sends an error message that the verification email failed
- 11a User receives verification email, but does not click verification link
  - 1. User information in database gets removed after some time

### ***SPECIAL REQUIREMENTS***

Req Num	Requirement
1	Backend should gracefully handle any invalid data
2	Backend should securely store sensitive user information

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

Var Num	Variation
1	Name field should allow special characters

***FREQUENCY OF OCCURRENCE:*** Quite frequently (at least at first) because we want most of our users to have an account with us, so that event organizers can keep track of the events.

### ***OTHER ISSUES***

Issue Num	Issue
1	How strongly should we enforce password security?

## Use Case # 6: User Login

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/2016
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 3 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Guest
<b>Secondary Actors</b>	Database, Client User
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>Client User: Wants access to profile and university specific organizations and events</li></ul>
<b>Preconditions</b>	User login information exists in database, and user's email is verified
<b>Success Post Condition</b>	User is logged into account and can view profile information, as well as followed organizations and events
<b>Failed Post Condition</b>	Backend sends error message for either email/password incorrect or email address unverified

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User arrives at 'Login' page
2	User enters email address for account that exists in database and is verified
3	User enters password matching selected account in database
4	Client sends user email and hashed password to backend
5	Backend checks user account has verified their email
6	Login is successful, user is redirected back to main screen, but logged in
7	Auth token saved to database

### EXTENSIONS or Alternate Flows

Step	Branching Action
2a	User enters email that does not exist in the database <ul style="list-style-type: none"><li>1. Backend sends error message that email or password is incorrect</li></ul>
3a	User enters password that does not match account in database <ul style="list-style-type: none"><li>1. Backend sends error message that email or password is incorrect</li></ul>
5a	User has not verified their account via email <ul style="list-style-type: none"><li>1. Backend sends error message that account has not been verified</li></ul>

### ***SPECIAL REQUIREMENTS***

<b>Req Num</b>	<b>Requirement</b>
1	For security purposes, the error message will not state specifically whether the user's email or password was incorrect

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

<b>Var Num</b>	<b>Variation</b>
0	N/A

***FREQUENCY OF OCCURRENCE:*** Often at the beginning because we want guests to create accounts in order to save their organization/event information; however, after a user logs in once, they will not have to login again until they log out

### ***OTHER ISSUES***

<b>Issue Num</b>	<b>Issue</b>
1	Should the user account be locked if too many failed login attempts?
2	Should there be a 'Forgot Password' option if the user doesn't remember their password?

## Use Case # 7: User Logout

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/2016
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 3 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Client User
<b>Secondary Actors</b>	Database
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• Client User: allow user to logout for whatever reason</li></ul>
<b>Preconditions</b>	User is already logged into their their own account
<b>Success Post Condition</b>	User is logged out of account, auth token is deleted
<b>Failed Post Condition</b>	auth token is not deleted and user is still logged in

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User arrives at 'Settings' page
2	User clicks logout
3	User auth token is deleted
4	User information is removed from local cache

### EXTENSIONS or Alternate Flows

Step	Branching Action
3a	User auth token is not deleted 1. User is still logged in instead of being logged out

### SPECIAL REQUIREMENTS

Req Num	Requirement
0	N/A

### TECHNOLOGY AND DATA VARIATIONS LIST

Var Num	Variation
0	N/A

**FREQUENCY OF OCCURRENCE:** Not often because users will stay logged in for ease of access to the app, rather than log out

**OTHER ISSUES**

Issue Num	Issue
0	N/A

## Use Case #8: Query Information for Profile

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User Level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 4 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Client User
<b>Secondary Actors</b>	Database
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• Client User: Wants to see their own personal information for their own account</li></ul>
<b>Preconditions</b>	The client user has to be successfully logged into the application
<b>Success Post Condition</b>	The client will be sent their profile information
<b>Failed Post Condition</b>	Backend sends an error status to the client

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User successfully logs in
2	Client requests user profile information from the backend
3	Backend finds the user and retrieves the specified user's information
4	User can view their user information on profile screen

### EXTENSIONS or Alternate Flows

Step	Branching Action
3a	Backend could not find the specified user in the database <ul style="list-style-type: none"><li>1. Backend sends message to client that user does not exist</li></ul>

### SPECIAL REQUIREMENTS

Req Num	Requirement
0	N/A

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

<b>Var Num</b>	<b>Variation</b>
--------------------	------------------

- |   |   |
|---|---|
| 1 | Should only be queried once and stored into the permanent storage of the device, to prevent multiple queries for the same data. |
|---|---|

***FREQUENCY OF OCCURRENCE:*** Getting the information from the database will be once for each user and then stored into the permanent storage of the device.

### ***OTHER ISSUES***

<b>Issue Num</b>	<b>Issue</b>
----------------------	--------------

- |   |   |
|---|---|
| 1 | Should users be able to change their profile information, even though the only thing might really be changed is their name? |
|---|---|



## Use Case #9 - User Follows Organization

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 5 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Client User
<b>Secondary Actors</b>	Event Admin User
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• User: Wants to be notified of events posted by organization</li><li>• Event Admin User: Wants a list of users following organization, as well as to know how many users are interested in the organization</li></ul>
<b>Preconditions</b>	Organization exists and is verified by the university, user is logged in, verified, and part of the university the organization belongs to
<b>Success Post Condition</b>	User follows organization and receives notifications of organization's posts and events
<b>Failed Post Condition</b>	Backend sends error status

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User arrives at organization's page
2	User clicks 'Follow' button
3	Database checks that user is part of organization's university
4	Organization appears under user's "Following" list
5	User will receive notifications when the organization has a new event

### EXTENSIONS or Alternate Flows

Step	Branching Action
3a	User is not part of organization's university <ul style="list-style-type: none"><li>1. Backend sends message that user is not part of organization's university</li></ul>
4a	User is already following organization <ul style="list-style-type: none"><li>1. Button to follow organization is removed, and is replaced by 'Unfollow' organization</li><li>2. When backend receives the request, user will no longer be following organization</li></ul>

### ***SPECIAL REQUIREMENTS***

Req Num	Requirement
0	N/A

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

Var Num	Variation
1	Should support at least Android, iPhone push notifications; other platforms are also possible to explore

***FREQUENCY OF OCCURRENCE:*** Frequently, because we assume users will want to follow organizations they are a part of in order to get event notifications and see what organizations they are a part of

### ***OTHER ISSUES***

Issue Num	Issue
1	Should users be able to turn on/off notifications from an organization?

## Use Case #10 - User Follows Event

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 6 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Client User
<b>Secondary Actors</b>	Event Admin User
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• User: wants to be notified of event reminders and changes in event details (location, time, cancellation, etc.)</li><li>• Event Admin User: wants a list of users following/attending event, as well as know how many users are interested in the event</li></ul>
<b>Preconditions</b>	Organization exists and is verified by the university, user is logged in, verified, and is a part of the event's university
<b>Success Post Condition</b>	User follows event and receives notifications of changes in event details and event reminders
<b>Failed Post Condition</b>	Backend sends error status

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User arrives at event page
2	User clicks 'Follow' button
3	Backend checks that user is part of event's university
4	Event appears under user's "Following" list
5	User will receive notifications when the event has updates

### EXTENSIONS or Alternate Flows

Step	Branching Action
3a	User is not part of event's university <ul style="list-style-type: none"><li>1. Backend sends error message that user is not part of event's university</li></ul>
4a	User is already following event <ul style="list-style-type: none"><li>1. Button to follow event is removed, and is replaced by 'Unfollow' event</li><li>2. When backend receives the request, user will no longer be following event</li></ul>

### ***SPECIAL REQUIREMENTS***

<b>Req Num</b>	<b>Requirement</b>
0	N/A

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

<b>Var Num</b>	<b>Variation</b>
1	Should support at least Android, iPhone push notifications; other platforms are also possible to explore

***FREQUENCY OF OCCURRENCE:*** Frequently, because we assume users will want to follow events they are interested in attending in order to get event notifications/reminders

### ***OTHER ISSUES***

<b>Issue Num</b>	<b>Issue</b>
1	How often should a user be reminded that an event is coming up? 1. Should a user be able to control how often they are notified?
2	Should users be able to turn on/off reminders and notifications from an event?

## Use Case #11 - Create an Organization

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 7 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Client User
<b>Secondary Actors</b>	University, Database
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• Client User: creates the specific organization; client user will be an Event Admin User once the organization is created</li><li>• University: wants to know who is the admin of the organization if they choose to remove a post or remove an organization for having malicious content/intent</li></ul>
<b>Preconditions</b>	User has successfully logged in, and client user is not already the admin of another organization
<b>Success Post Condition</b>	Client user is an Event Admin and can update information about organization/events, post events, and etc.
<b>Failed Post Condition</b>	Backend will not allow the user to make an organization, and will send an error message to the client

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User who is not already admin for an organization requests the backend to make an organization
2	User provides an email address specifically for the organization to use
2	Backend will ensure user is not already an event admin
3	Backend creates new organization
4	Client displays new organization to user
5	The user, now an event admin user, has permission to create or update events for that organization

### EXTENSIONS or Alternate Flows

Step	Branching Action
2a	User does not specify an organization-specific email address <ul style="list-style-type: none"><li>1. Defaults to user's personal email address, as listed in their profile</li><li>2. Process continues as normal</li></ul>

- |    |  |
|----|--|
| 3a | User is already an admin of another organization   |
| 1. | Backend will send the user an error message and not allow him to create another organization |

### ***SPECIAL REQUIREMENTS***

Req Num	Requirement
0	N/A

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

Var Num	Variation
0	N/A

***FREQUENCY OF OCCURRENCE:*** This will be used in moderation. Whenever a group needs to be created this will be used.

### ***OTHER ISSUES***

Issue Num	Issue
1	Student organizations should be verified to actually be associated with the university <ul style="list-style-type: none"><li>• Right now, the one organization per user limit hopefully will prevent spam</li></ul>

## Use Case #12: Add User as Organization Admin

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 7 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Client User
<b>Secondary Actors</b>	Event Admin User, Database
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• Client User: Wants to become an admin for the organization to post events and maintain the organization page</li><li>• Event Admin User: Needs another admin for the organization to post events and maintain the organization page</li></ul>
<b>Preconditions</b>	Client user is a part of the organization and maintains an officer-related position in the organization The user who is being added can't be part an Event Admin User of another organization
<b>Success Post Condition</b>	User is added as an admin of the organization in the database
<b>Failed Post Condition</b>	Backend will send an error status to the client, user is not made an admin

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	Existing admin arrives at organization page
2	Existing admin clicks 'Add Admin' button
3	Existing admin enters user email (username, not inclusive of @<domain>.edu)
4	Database checks if entered email exists
5	Database checks that user of entered email is not already an admin of another organization
6	Confirmation email is sent to user email address
7	User clicks link in confirmation email
8	User account is confirmed on the backend and user is now allowed edit organization information and add events
9	Newly added admin is listed under admin in database

### ***EXTENSIONS or Alternate Flows***

<b>Step</b>	<b>Branching Action</b>
-------------	-------------------------

- |    |  |
|----|--|
| 4a | Email does not exist in database <ol style="list-style-type: none"><li>1. User does not exist through the entered email</li><li>2. Backend sends error message that user does not exist</li></ol>  |
| 5a | User is already an admin of another organization <ol style="list-style-type: none"><li>1. Backend will send the client an error message</li><li>2. Client will display that the user is already an admin of another organization</li></ol> |
| 6a | An error occurs while sending the verification email <ol style="list-style-type: none"><li>1. Backend sends an error message that the verification email failed</li></ol>  |
| 7a | User does not click confirmation link <ol style="list-style-type: none"><li>1. User is not allowed to edit to organization's page, and user's information in organization's table in the database gets removed after some time</li></ol>   |

### ***SPECIAL REQUIREMENTS***

<b>Req</b>	<b>Requirement</b>
------------	--------------------

**Num**

0	N/A
---	-----

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

<b>Var</b>	<b>Variation</b>
------------	------------------

**Num**

0	N/A
---	-----

***FREQUENCY OF OCCURRENCE:*** Not often, admin should only need to give a few users admin privileges for an organization.

### ***OTHER ISSUES***

<b>Issue</b>	<b>Issue</b>
--------------	--------------

**Num**

1	Should there be a limit on the number of admins for an organization?
---	--



## Use Case #13: Revoke Access for Organization Admin

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 7 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Event Admin User
<b>Secondary Actors</b>	Event Admin User, Database
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• Event Admin User: Wants to be removed (or needs to be removed) from an organization's admin list</li></ul>
<b>Preconditions</b>	Must be an admin to remove another admin, admin to be removed is an admin of the organization, admin performing the remove is an admin of the organization
<b>Success Post Condition</b>	User is removed as an admin of the organization in the database
<b>Failed Post Condition</b>	Backend will send an error status to the client, admin is not removed from organization

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	Existing admin arrives at organization page
2	Existing admin clicks 'Remove Admin' button
3	Database queries list of admins for the organization
4	Existing admin clicks on admin to remove
5	Database removes the admin selected

### EXTENSIONS or Alternate Flows

Step	Branching Action
2a	Admin is removing a different admin
2b	Admin is removing themselves <ul style="list-style-type: none"><li>1. If the admin is the only admin for the organization, they should not be allowed to remove themselves</li></ul>
5a	An error occurs while sending the verification email <ul style="list-style-type: none"><li>1. Backend sends error message that the admin was not removed</li></ul>

### ***SPECIAL REQUIREMENTS***

<b>Req Num</b>	<b>Requirement</b>
0	N/A

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

<b>Var Num</b>	<b>Variation</b>
0	N/A

***FREQUENCY OF OCCURRENCE:*** Not often; an admin should only need to revoke a user's admin privileges for an organization once or twice a year

### ***OTHER ISSUES***

<b>Issue Num</b>	<b>Issue</b>
1	Should there be a verification prompt that questions the user whatever or not they should remove the user?

## Use Case #14: Create an Event

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 8 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Event Admin User
<b>Secondary Actors</b>	Guest, Client User, University
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• Guest: Would want to see events at a university without having to log in</li><li>• Client User: Wants to see events at their university, as well as get a notification if they were following the organization that is creating the event</li><li>• University: Might want to keep track of events that are going on</li></ul>
<b>Preconditions</b>	Organization for the event exists, and event admin user is an admin of that organization
<b>Success Post Condition</b>	Event is created for the organization, and users following the organization are sent a notification for the event
<b>Failed Post Condition</b>	Backend sends an error message to the client, and no event is created

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	User navigates to an organization they are admin for
2	Event Admin User starts the process to create a new event
3	Event Admin User enters the event details, name and date at least are required
4	Client sends admin user info along with event details to backend
5	Use case #15 takes care of updating the event details as provided by the user
6	Backend creates the event for the organization
7	Backend sends notifications with event info to all users following the organization

### EXTENSIONS or Alternate Flows

Step	Branching Action
2a	User isn't an admin for the organization <ul style="list-style-type: none"><li>1. Backend sends an error message that user doesn't have permission to create an event for the organization</li></ul>

- 3a Not enough details provided for event (eg. name not provided)
  - 1. Backend sends an error message that event needs more detail, and event is not created
- 7a Sending push notifications to user might fail
  - 1. Backend logs an error and does nothing else

### ***SPECIAL REQUIREMENTS***

Req Num	Requirement
------------	-------------

- |   |   |
|---|---|
| 1 | Backend should take advantage of native push notification frameworks for best reliability |
|---|---|

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

Var Num	Variation
------------	-----------

- |   |  |
|---|--|
| 1 | Should support at least Android, iPhone push notifications; other platforms are also possible to explore |
|---|--|

***FREQUENCY OF OCCURRENCE:*** Fairly often, as this is one of the main functions of the service

### ***OTHER ISSUES***

Issue Num	Issue
--------------	-------

- |   |  |
|---|--|
| 1 | Should followers of the organization be automatically marked as following the event? |
|---|--|

## Use Case #15 - Update Event Details

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User-Level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 8 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Event Admin User
<b>Secondary Actors</b>	Database
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>Event Admin User: Wants to change the details of the event if something wants to be changed about the event. The things that could be updated about an event are event name, location, date, time, description, or image.</li></ul>
<b>Preconditions</b>	Event is already created and posted, Event admin is updating the event details
<b>Success Post Condition</b>	Database will have the new and updated details about the event
<b>Failed Post Condition</b>	Backend sends error message to client, event is not updated

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	Event admin makes a request to change the details about the event (name, location, date, time, description, or the image)
2	Backend receives the request and searches for the event
3	Database will update the specified attributes of the event

### EXTENSIONS or Alternate Flows

Step	Branching Action
3a	Backend can't find the event to update in the database <ul style="list-style-type: none"><li>1. Backend will send an error message to the client app</li></ul>
3b	Backend can't update the specified attributes of the event <ul style="list-style-type: none"><li>1. Backend will send an error message to the client app</li></ul>

### ***SPECIAL REQUIREMENTS***

<b>Req Num</b>	<b>Requirement</b>
--------------------	--------------------

- |   |   |
|---|---|
| 1 | Ensure name and date aren't removed from an event   |
| 2 | Backend should take advantage of native push notification frameworks for best reliability |

### ***TECHNOLOGY AND DATA VARIATIONS LIST***

<b>Var Num</b>	<b>Variation</b>
--------------------	------------------

- |   |  |
|---|--|
| 1 | Should support at least Android, iPhone push notifications; other platforms are also possible to explore |
|---|--|

***FREQUENCY OF OCCURRENCE:*** Fairly often, as event details can often change at any time

### ***OTHER ISSUES***

<b>Issue Num</b>	<b>Issue</b>
----------------------	--------------

- |   |  |
|---|--|
| 1 | What should be the default values for event details that aren't provided (especially the event image, placeholder or university logo or something else)? |
| 2 | Should there be an upper limit on how long a description can be?   |
| 3 | Should there be any sort of filtering be used for the event detail text to remove inappropriate words?   |

## Use Case #16 - Delete Event

### GENERAL CHARACTERISTICS

<b>Author</b>	Anirudh Bagde, Rupin Bhalla, Catherine Ta
<b>Last Update:</b>	9/25/16
<b>Scope</b>	Backend of HypeU event discovery service
<b>Level</b>	User Level
<b>Status</b>	Under review
<b>UML Diagram</b>	See page 8 of 'UML Diagrams.pdf'
<b>Primary Actor</b>	Event Admin User, University
<b>Secondary Actors</b>	Database, Client User, Guest
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"><li>• Event Admin User: Should have the power to delete an event if the event gets cancelled, when the event is deleted the backend remove it from the database</li><li>• University: Has the power to delete an event in case the event post/description contains malicious content.</li></ul>
<b>Preconditions</b>	Event is already created
<b>Success Post Condition</b>	Event doesn't appear anymore in the database
<b>Failed Post Condition</b>	Backend sends error status to client, event is not deleted

### MAIN SUCCESS SCENARIO (or basic flow)

Step	Action
1	Event Admin User navigates to event
2	Client sends request to delete event
3	Backend finds the event and removes it from the database
4	Client displays a message for successful deletion

### EXTENSIONS or Alternate Flows

Step	Branching Action
3a	Backend cannot find the event to be deleted <ul style="list-style-type: none"><li>1. Backend sends an error message to the client app</li></ul>

### SPECIAL REQUIREMENTS

Req Num	Requirement
0	N/A

### **TECHNOLOGY AND DATA VARIATIONS LIST**

Var Num	Variation
0	N/A

**FREQUENCY OF OCCURRENCE:** This will happen occasionally. It will depend on how many times the Event Admin Users want to delete an event

### **OTHER ISSUES**

Issue Num	Issue
1	Should event followers be sent a notification of event deletion?