

Project Deliverables

Version: 5.0

Date: May 1, 2018

CometClubs

Object Oriented Analysis and Design

CS 6359— Instructor: R. Z. Wenkster

TA: Behnam Torabi

Your names:

Chase Davis

Pramod Hegde

Aswin Krishna Gunasekaran

Akash Kumar A M

Tej Patel

Table of Contents

PART I: USE CASE ANALYSIS	3
REVISION HISTORY	4
1. DOMAIN MODEL	5
2. USE CASE DIAGRAMS (UCD)	5
2.1. UCD Level 0	5
2.2. UCD Level 1	5
2.3. UCD Prioritization	5
3. USE CASE DESCRIPTIONS	5
3.1. UC xx 5	
3.1.1 UC xx Main Flow	5
3.1.2 UC xx «Includes»	5
3.1.3 UC xx «Extends»	5
3.1.4 UC xx Supplementary Specification	5
3.2. UC yy 5	
3.2.1 UC yy Main Flow	5
3.2.2 UC yy «Includes»	5
3.2.3 UC yy «Extends»	5
3.2.4 UC yy Supplementary Specification	5
4. SUPPLEMENTARY SPECIFICATION	5
5. SYSTEM SEQUENCE DIAGRAMS	6
6. OPERATION CONTRACTS	6
PART II: USE CASE DESIGN	7
REVISION HISTORY	8
1. SOFTWARE ARCHITECTURE	9
2. USE CASE REALIZATIONS	9
3. DESIGN CLASS DIAGRAMS	9
PART III: IMPLEMENTATION	10
REVISION HISTORY	11

Part I: Use Case Analysis

Version 5.0

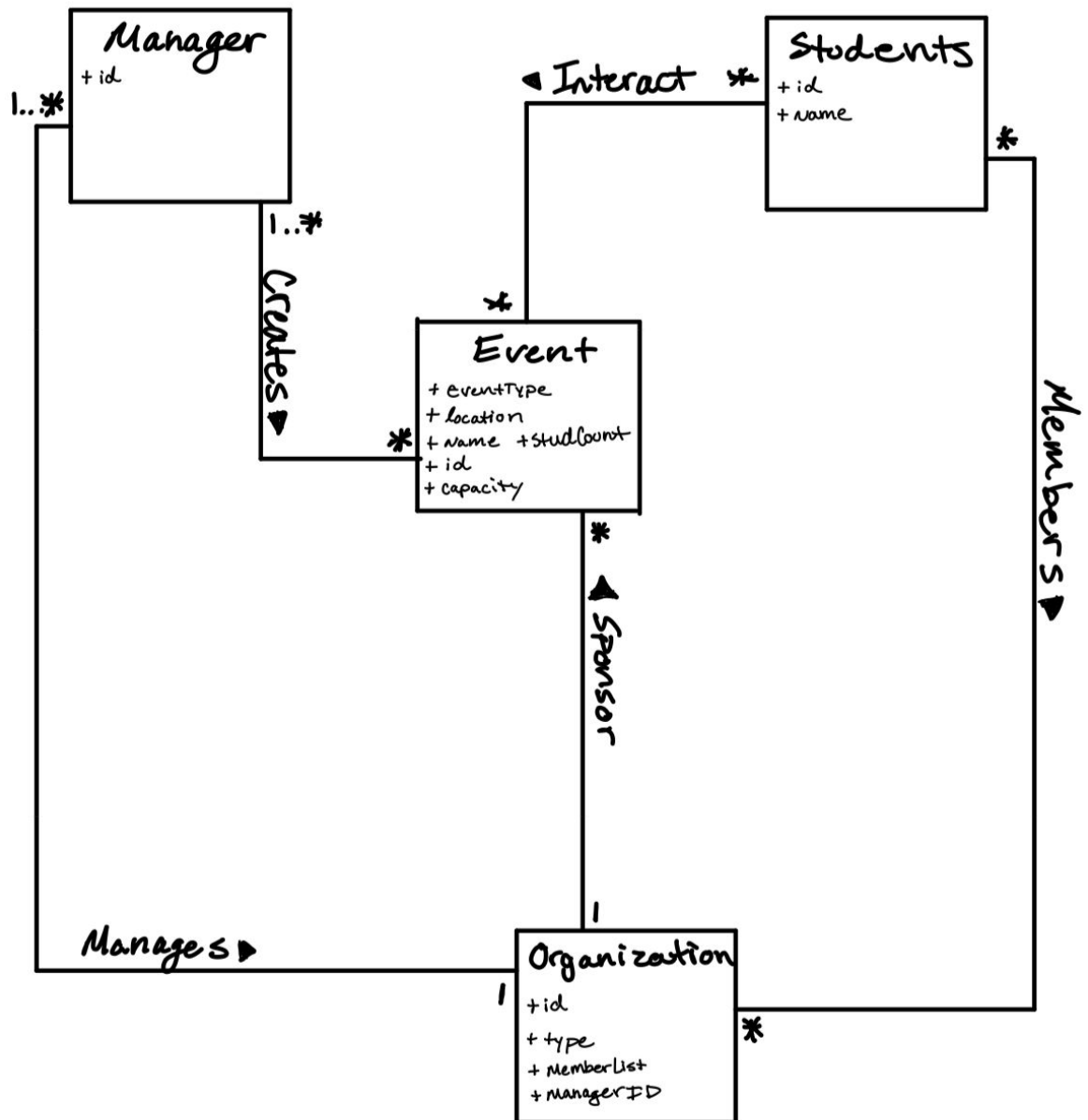
Date: May 1, 2018

Version - description	Date	Description	Author
vs 1.0 - Domain Model, UCD levels 0 and 1, Supplementary Specification	March 30, 2018	First draft. To be refined primarily during elaboration.	Team
vs 2.0 - Domain Model, UCD levels 0 and 1, Supplementary Specification	April 6, 2018	Updated domain model, updated UCD levels 0 and 1, and supplementary specifications	Team
vs 3.0 - Updated Deliverables Use Case Descriptions (Most important Use Case and <<includes>> in fully-dressed format, Other UCs in (casual and brief formats)), Black box sequence diagrams (for base use cases and <<includes>>), Operation contracts, Software Architecture	April 11, 2018	Expanding the UCD, creating the black box sequence diagram, operation contracts, and software architecture.	Team
vs 4.0 - Updated deliverables, Use-case realization, design class diagram	April 20, 2018	Updating the UCD to further include Join Organization correctly. Adding the new deliverables	Team
vs 5.0	Final Review	Reviewing all material for submission	Team

Revision History

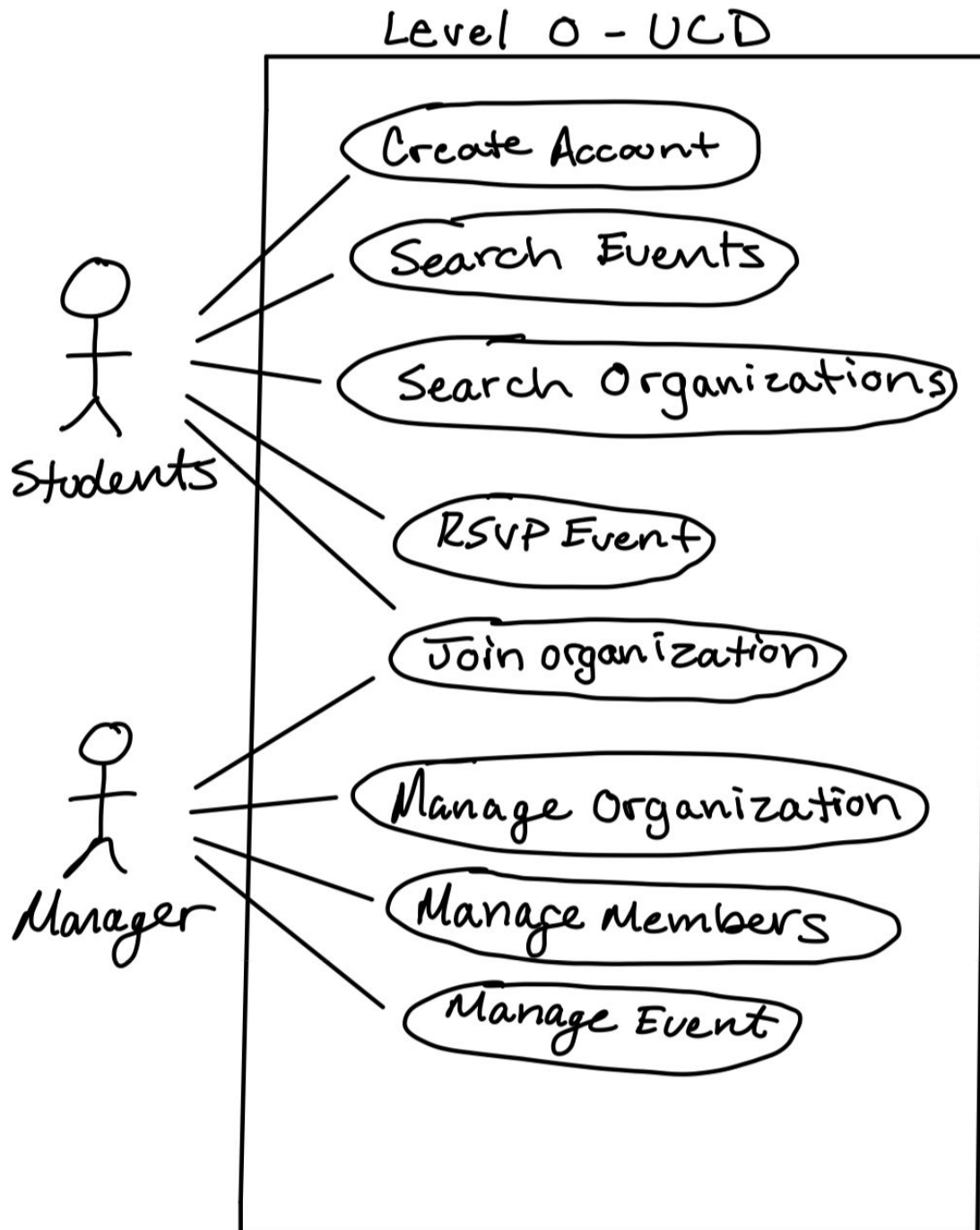
1. Domain Model

1.1. Diagram



2. Use Case Diagrams (UCD)

2.1. UCD Level 0



2.2. Use Case Descriptions in brief format

Create Account:

The student is able to open the mobile application and sign up for an account. Creating an account will involve the student filling out the form with all of their credentials (i.e name, date of birth, UTD ID, etc). Success of this use case means the student is able to login with the created account and use the application.

Manage Members:

The manager of an organization will be able to manage the current members. Managing the organization will allow the manager to add members to the organization. The manager will also able to remove members from the organization. The extensions point will not allow the manager to add more members than the organization capacity or remove all members.

Manage Organization:

The manager is able to manage the different aspects of an organization. Managing will allow the manager to create a new organization. The manager is also able to update the details of the organization (i.e. the name of the organization, description, events, etc). The manager will also have the ability to delete the organization completely. The extension point for this case will not allow a manager to create an organization that already exists.

2.3. Use Case Descriptions in causal format

Search Organizations:

The student is able to use the mobile application to search for a specific organization that they would like to see more information about. They would be able to enter an organization's name and see their organization's description and events. Search organizations will further breakdown into browse organizations and view organizations.

Browse organization will allow the student to scroll through a list of different organizations. This will allow students that do not have a desired organization a chance to see all of the organizations that are offered.

View organization will allow the students to see all of the details of the organization. Whether searching or browsing, when a student selects an organization they will be able to view the description of the organization and any related information.

Search Event:

The student is able to use the mobile application to search for a specific event that they would like to see more information about. They would be able to enter an event name and see the event description. Search events will further breakdown into browse events and view event.

Browse event will allow the student to scroll through a list of different events. This will allow students that do not have a desired event a chance to see all of the events that are currently offered.

View event will allow the students to see all of the details of the event. Whether searching or browsing, when a student selects an event they will be able to view the description of the event and any related information. View event will have an extend use case where the student may request to view an event that is not found. In this event the student will be notified that the event does not exist and they must refine their search.

Join Organization:

The student will be able to join an organization and become an active member. When joining an organization, the student will have to send a request to join. This will further break down join organization into send join request and request response.

Send join request will allow the student to send an active request to the organization manager for approval. Pending the response, the student will either be added to the organization or denied membership.

Request response will act as the action the manager will take when responding to a student's request to join the organization. Upon reviewing the request, the manager will have the option to accept the request or deny the request.

Manage Event:

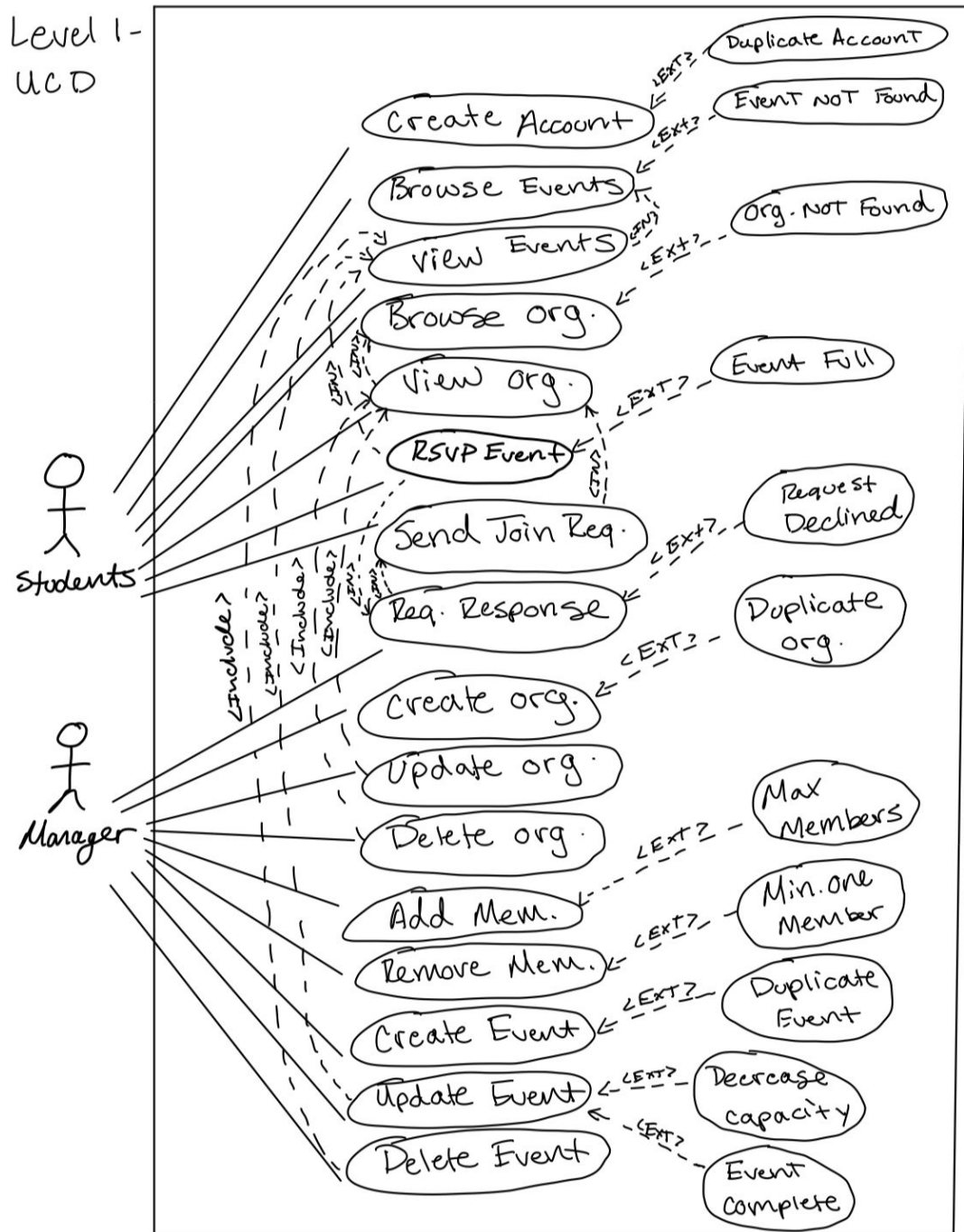
Manage event will allow the organization's manager to make changes to the event. These changes will include actions such as creating events, updating events, or deleting events.

Creating events will allow the organization's manager to post new events to the system. They will be able to provide the event description, the event capacity, as well as decide if the event will be open to the public or just to the organization's members. The extension point of this case will not allow the manager to create a duplicate event.

Update events will allow the organization's manager to make changes to an event that has already been posted. These changes may include, but are not limited to, changing the event time, capacity, or location. The extension point for updating an event is that a manager is not allowed to decrease the event capacity below the total of students that have already RSVP'd for the event. Additionally, the manager may not update an event after the event has already completed.

Delete events will allow the organization's manager to remove an event all together. There are many different occasions in which a manger may decided to do this. The event will be removed from the system and all student RSVPs will be removed.

2.4. UCD Level 1



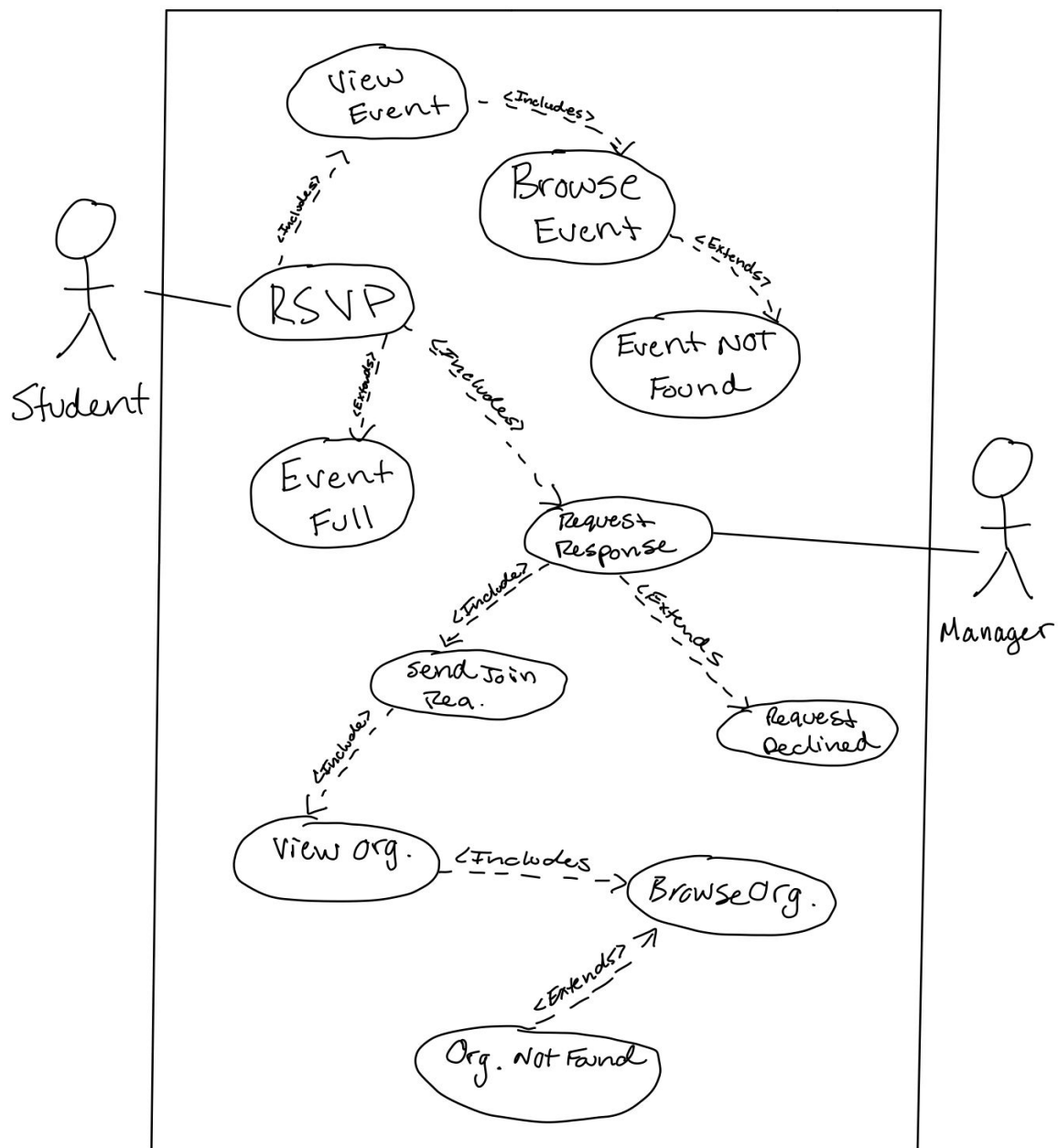
2.5. UCD Prioritization (at level 0)

1. Search Event
2. RSVP Event
3. Search Organization
4. Join Organization
5. Create Account
6. Manage Members
7. Manage Event
8. Manage Application

3. Use Case Specifications in Fully Dressed Format

3.1. UC RSVP

3.1.1 Level 1 UCD for UC RSVP



3.1.2 Base Use Cases

Use Case RSVP	
ID: 04	Type: Base
Brief Description: Student will successfully register to attend the event	
Primary Actors: Student	
Secondary Actors: None	
Preconditions: Student has an account and is successfully logged in	
Main Flow: <ol style="list-style-type: none">1. Includes(Req Response)2. Includes (View Event)3. The student reads the event description.4. The student selects to RSVP for the event.5. The system checks if the event is at capacity. Extension point: capacity Full, Extension: Event Full6. The system adds the student to the event.7. The system updates the capacity of the event.8. The system sends a confirmation notification to the student.	
Postconditions: Student has successfully registered and the event capacity is updated	
Alternative Flows: None	
Non-Functional Requirements: <ul style="list-style-type: none">- Student should receive a confirmation within five seconds.- Student data and identification should be confidential.	
Technology and Data Variation List: <ul style="list-style-type: none">- Student RSVPs for the event through the mobile application on their mobile device.	
Open Issues: <ul style="list-style-type: none">- If more than one student RSVPs for the same spot in an event, at the same time, who receives the RSVP?- Is the student able to cancel the RSVP?	

3.1.3 «Includes» Use Cases

Use Case Req Response	
ID: 05	Type: Include
Brief Description: The manager approves or rejects students join request to an organization	
Primary Actors: Manager	
Secondary Actors: None	
Preconditions: Manager has an account and is successfully logged in	
Main Flow: <ol style="list-style-type: none">1. Includes (Send Join Request)2. The manager approves or rejects a student's request to join the organization3. The system checks if the request is rejected	

Extension point: request rejected, Extension: Request Declined 4. The system adds the student to the organization.
Postconditions: A student's request to join the organization is approved or denied by the manager.
Alternative Flows: None
Non-Functional Requirements: - The managers response to the join request should be processed within five seconds.
Technology and Data Variation List: - Manager views the join requests through the web application.
Open Issues: None

3.1.4 «Includes» Use Cases

Use Case Send Join Request	
ID: 06	Type: Include
Brief Description: The student views an organization's details and sends a join request to the manager	
Primary Actors: Student	
Secondary Actors: None	
Preconditions: Student has an account and is successfully logged in	
Main Flow: <ol style="list-style-type: none"> 1. Includes (View Organization) 2. The student sends a join organization request to the system 3. The system forwards the student's join organization request to the manager for approval 	
Postconditions: A student successfully placed an request to join the organization and waits for manager's approval	
Alternative Flows: None	
Non-Functional Requirements: <ul style="list-style-type: none"> - The join request should be processed within five seconds. 	
Technology and Data Variation List: <ul style="list-style-type: none"> - Student views the event through the mobile application on their mobile device. 	
Open Issues: None	

3.1.5 «Includes» Use Cases

Use Case View Organization	
ID: 07	Type: Include
Brief Description: The student is able to view the description of the organization	
Primary Actors: Student	
Secondary Actors: None	
Preconditions: Student has an account and is successfully logged in	

Main Flow:
<ol style="list-style-type: none"> 1. Includes (Browse Organizations) 2. The student selects an organization from the organization list 3. The system displays the full description of the organization to the student
Postconditions: The student is able to read the description of an individual organization.
Alternative Flows: None
Non-Functional Requirements:
<ul style="list-style-type: none"> - The organization description should be displayed within five seconds.
Technology and Data Variation List:
<ul style="list-style-type: none"> - Student views the organization through the mobile application on their mobile device.
Open Issues: None

3.1.6 «Includes» Use Cases

Use Case Browse Organization	
ID: 08	Type: Include
Brief Description: The student is able to scroll through a list of organizations	
Primary Actors: Student	
Secondary Actors: None	
Preconditions: Student has an account and is successfully logged in	
Main Flow: <ol style="list-style-type: none"> 1. The student opens the application 2. The student types in the name of the organization to search 3. The student searches for an organization Extension point: orgNotFound, Extension: Organization Not Found 4. The system displays a list of the organizations found 	
Postconditions: The student is able to see the list of organizations	
Alternative Flows: None	
Non-Functional Requirements:	
<ul style="list-style-type: none"> - The system should display the returned results within five seconds 	
Technology and Data Variation List:	
<ul style="list-style-type: none"> - Student browses for organizations through the mobile application on their mobile device. 	
Open Issues: None	

a. «Includes» Use Cases

Use Case View Event	
ID: 09	Type: Include
Brief Description: The student is able to view the description of the event	
Primary Actors: Student	
Secondary Actors: None	

Preconditions: Student has an account and is successfully logged in
Main Flow: <ol style="list-style-type: none"> 1. Includes (Browse Event) 2. The student selects an event from the event list 3. The system displays the full description of the event to the student
Postconditions: The student is able to read the description of an individual event.
Alternative Flows: None
Non-Functional Requirements: <ul style="list-style-type: none"> - The event description should be displayed within five seconds.
Technology and Data Variation List: <ul style="list-style-type: none"> - Student views the event through the mobile application on their mobile device.
Open Issues: None

3.1.7 «Includes» Use Cases

Use Case Browse Event	
ID: 10	Type: Include
Brief Description: The student is able to scroll through a list of events	
Primary Actors: Student	
Secondary Actors: None	
Preconditions: Student has an account and is successfully logged in	
Main Flow: <ol style="list-style-type: none"> 1. The student opens the application 2. The student types in the name of the event to search 3. The student searches for an event Extension point: eventNotFound, Extension: Event Not Found 4. The system displays a list of the events found 	
Postconditions: The student is able to see the list of events	
Alternative Flows: None	
Non-Functional Requirements: <ul style="list-style-type: none"> - The system should display the returned results within five seconds 	
Technology and Data Variation List: <ul style="list-style-type: none"> - Student browses for events through the mobile application on their mobile device. 	
Open Issues: None	

3.1.8 «Extends» Use Cases

Use Case Request Declined	
ID: 11	Type: Extends

Brief Description: The system is unable to add the student to the organization because the request is denied by the manager.
Primary Actors: Manager
Secondary Actors: None
Segment 1
Preconditions: Manager denied student's request to join the organization.
Main Flow:
<ol style="list-style-type: none"> 1. The system detects that a student's join request is denied by the manager. 2. The student is notified that the join request to an organization is denied.
Postconditions: The student is not added to the organization
Non-Functional Requirements:
<ul style="list-style-type: none"> - The student should be notified that the manager declined the join request to an organization within five seconds.
Technology and Data Variation List:
<ul style="list-style-type: none"> - The system sends an alert to the students mobile device that the join request to an organization was rejected.
Open Issues: None

3.1.9 «Extends» Use Cases

Use Case Organization Not Found	
ID: 12	Type: Extends
Brief Description: The organization that the student searches for is not found.	
Primary Actors: Student	
Secondary Actors: None	
Segment 1	
Preconditions: The student has entered and search for a specific organization	
Main Flow:	
<ol style="list-style-type: none"> 1. The system searches for an organization 2. The system is unable to find the organization 	
Postconditions: The system notifies the student that the search was not found.	
Non-Functional Requirements:	
<ul style="list-style-type: none"> - The system should notify the student that the organization was not found within five seconds. 	
Technology and Data Variation List:	
<ul style="list-style-type: none"> - The system sends an alert to the students mobile device that the organization was was not found. 	
Open Issues: None	

3.1.10 «Extends» Use Cases

Use Case Event Full	
ID: 13	Type: Extends

Brief Description: The system is unable to add the student to the event because the event is at maximum capacity.
Primary Actors: Student
Secondary Actors: None
Segment 1
Preconditions: The event capacity has reached maximum capacity.
Main Flow:
<ol style="list-style-type: none"> 1. The system detects that the event has reached maximum capacity. 2. The student is notified that the event was at maximum capacity.
Postconditions: The student is not added to the event
Non-Functional Requirements:
<ul style="list-style-type: none"> - The student should be notified that the event is at maximum capacity within five seconds.
Technology and Data Variation List:
<ul style="list-style-type: none"> - The system sends an alert to the students mobile device that the event was at maximum capacity and they were not added to the event.
Open Issues: None

3.1.11 «Extends» Use Cases

Use Case Event Not Found	
ID: 14	Type: Extends
Brief Description: The event that the student searches for is not found	
Primary Actors: Student	
Secondary Actors: None	
Segment 1	
Preconditions: The student has entered and search for a specific event	
Main Flow:	
<ol style="list-style-type: none"> 1. The system searches for the event 2. The system is unable to find the event 	
Postconditions: The system notifies the student that the search was not found.	
Non-Functional Requirements:	
<ul style="list-style-type: none"> - The system should notify the student that the event was not found within five seconds. 	
Technology and Data Variation List:	
<ul style="list-style-type: none"> - The system sends an alert to the students mobile device that the event was was not found. 	
Open Issues: None	

4. System Supplementary Specification

4.1. Quality attributes

Reliability - Proper handling of transactions

Security - User Authentication and authorization

Performance - Event creation is immediately available, search should return within less than two seconds , RSVP takes less than fifteen seconds

Usability - Easy navigation, audio playback for the blind

Availability - Amazon EC2 manages resources for high availability

Manageability - System administrator has rights to manage groups

Maintainability - Agile development and version control

4.2. Hardware and software constraints

Mobile application supports Android OS version 5.0 & above

GPS support

Touch screen (mobile phones, tablets)

Web application is supported by IE 9+, Chrome 9+,

4.3. Development constraints

Android Studio, Eclipse, and SQL Database

4.4. Design and implementation constraints

HTML5, CSS, Javascript, Java Technologies and Oracle 11g

4.5. Licensing and other legal concerns

Using open source components

4.6. Standards (technical, safety, quality)

Quality - Accurate location of the events

Safety - Authentication and authorization of users

Technical - Robustness

4.7. Physical environment concerns (heat, vibration)

Overheating

4.8. Operational concerns (how are errors handled?)

Contact systems administrator

4.9. Application specific domain rules

For UTD students only

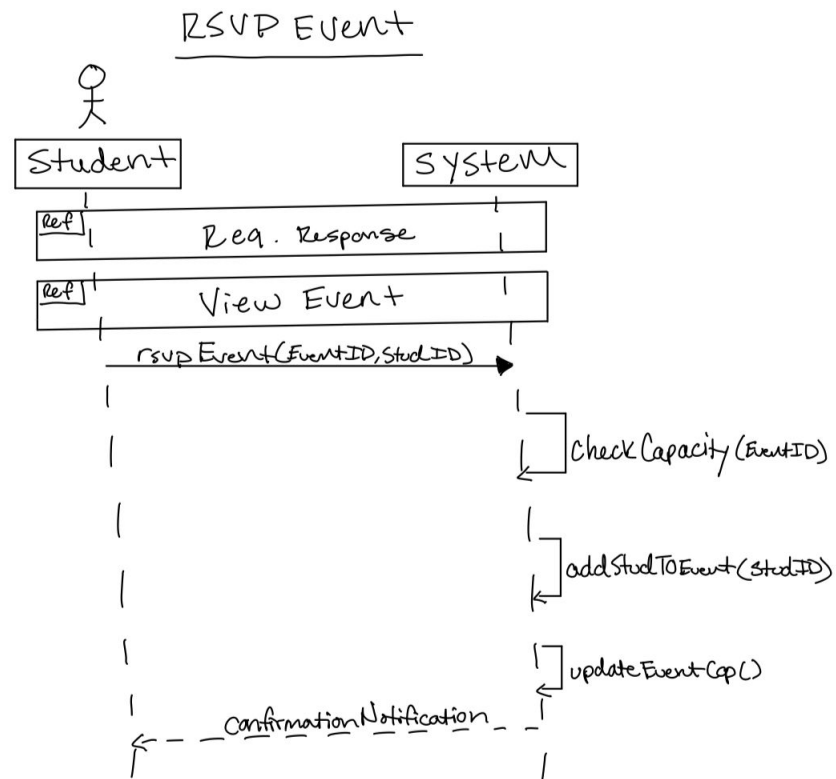
Events within UTD

4.10. Information in domains of interest (what is the cycle of credit payment handling?)

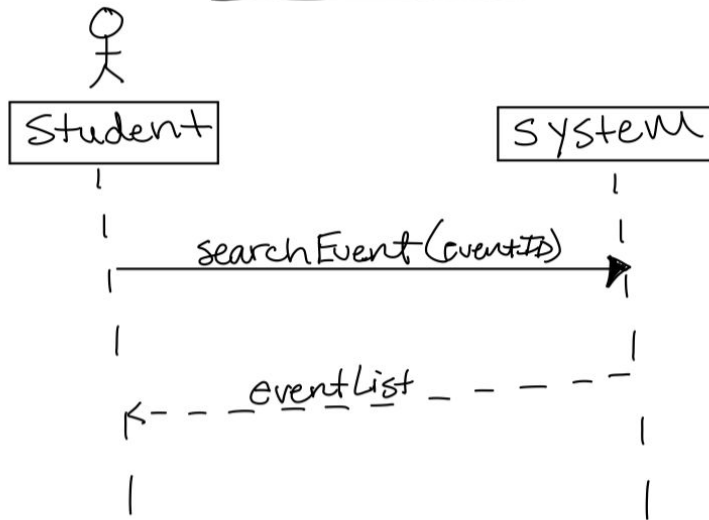
Distance of events from UTD

5. System Sequence Diagrams

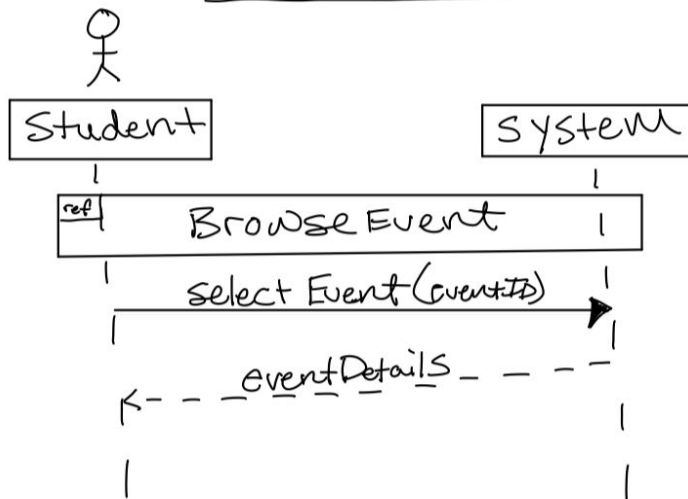
UC RSVP - Black Box

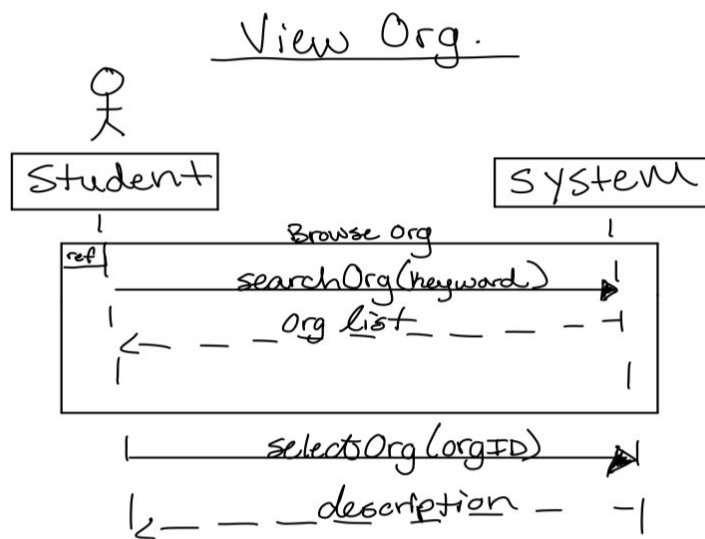
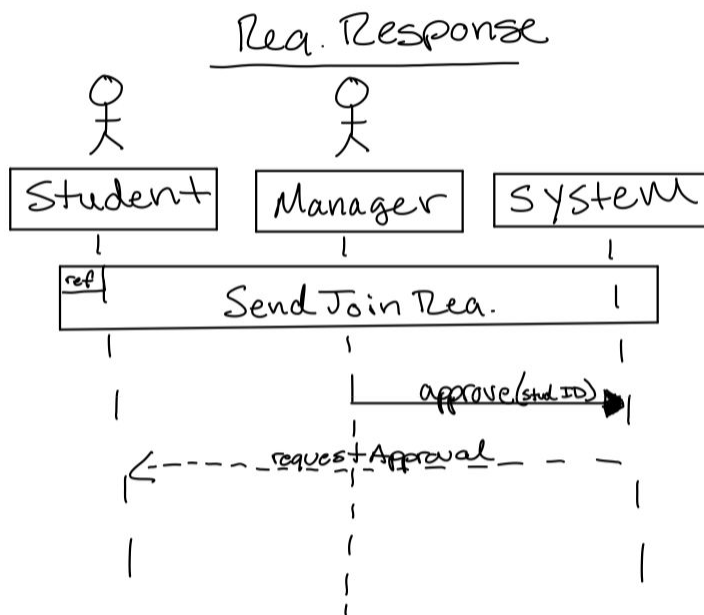


Browse Event



View Event





6. Operation Contracts

6.1 Operation contract for view event use case

Contract C01 : selectEvent

Operation: selectEvent(EventID)

Cross References : View Event

Precondition : An object e of Event and object s of Student was created

Postcondition : s is associated with e based on ID attribute of Event

6.2 Operation contract for the Browse Event use case

Contract C02 : searchEvent

Operation: searchEvent(keyword)

Cross References : Browse Event

Precondition : An object s of student was created

Postcondition : s is associated with event based on name attribute

6.3 Operation contract for the RSVP use case

Contract C03 : RSVPEvent

Operation: RSVPEvent (EventID,studentID)

Cross References : RSVP Event

Precondition : An object e of Event and object s of Student was created

Postcondition : s is associated with e of event on studentID attribute.

studCount attribute of object e is updated by 1 based on EventID .

6.4 Operation contract for Request Response

Contract C04 : Approve

Operation: Approve(studentID)

Cross References : Request Response

Precondition : An object m of Manager and o of Organization was created

Postcondition :m is associated o of Organization based on managerID

An object s of Student was created with attribute studentID and associated with o object

memberList attribute of o is updated

6.5 Operation contract for Send Join Request

Contract C05 : sendJoinRequest

Operation: sendJoinRequest (studentID,OrgID)

Cross References :Send Join Request.

Precondition : An object s of student gets created

Postcondition :s with attribute studentID is associated with Organization based on OrgID

6.6 Operation contract for view Org use case

Contract C06 : selectOrg

Operation: selectOrg(orgId)

Cross References : View Org

Precondition : An object o of Organization and object s of Student was created

Postcondition: s is associated with o based on ID attribute of Organization

6.7 Operation contract for the Browse Org use case

Contract C07 : search Org

Operation: searchOrg(Keyword)

Cross References : Browse Org

Precondition : An object s of student was created

Postcondition: s is associated with Organization based on name(keyword searched is the name) attribute

Part II: Use Case Design

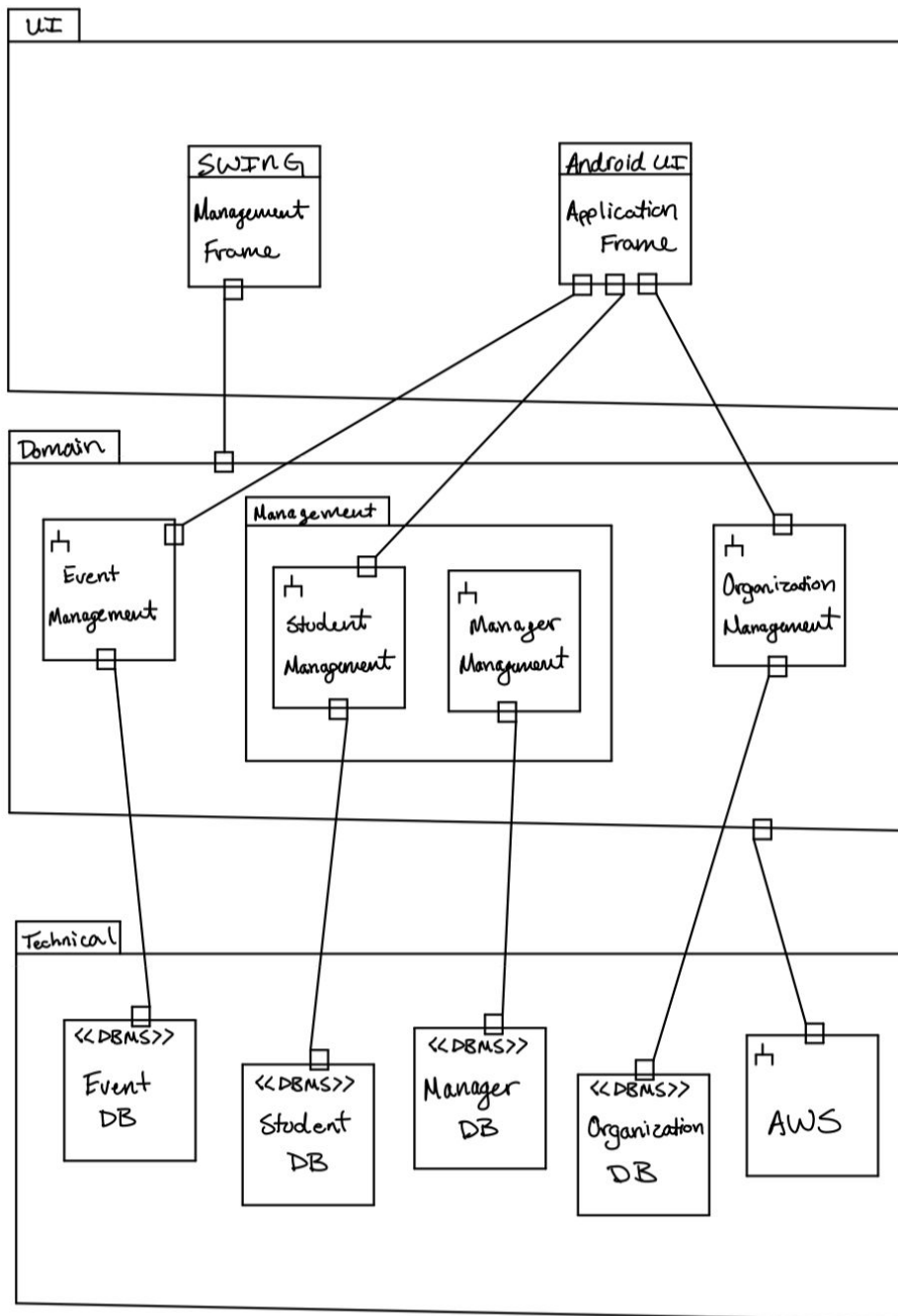
Version 3.0

May 1, 2018

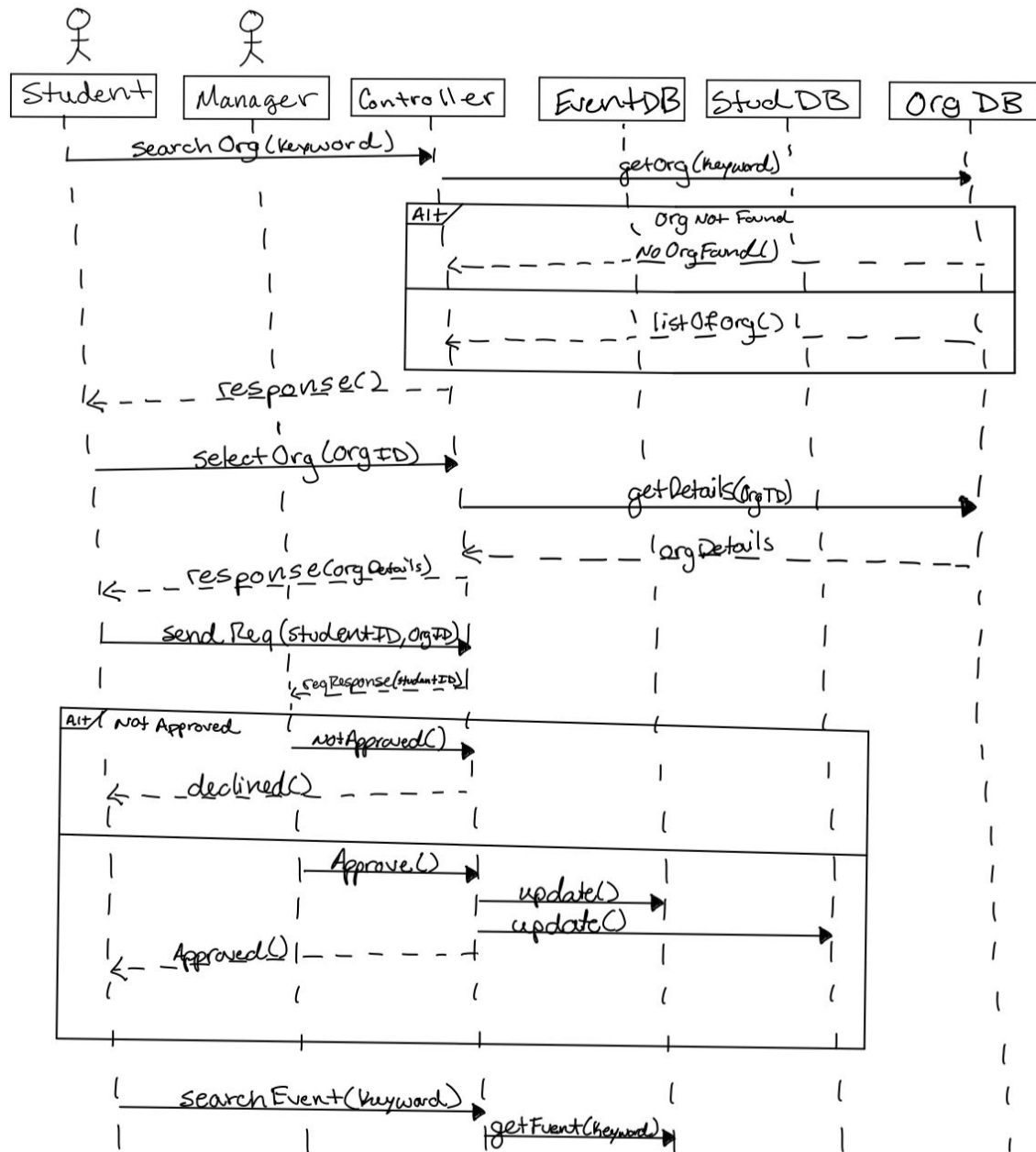
Revision History

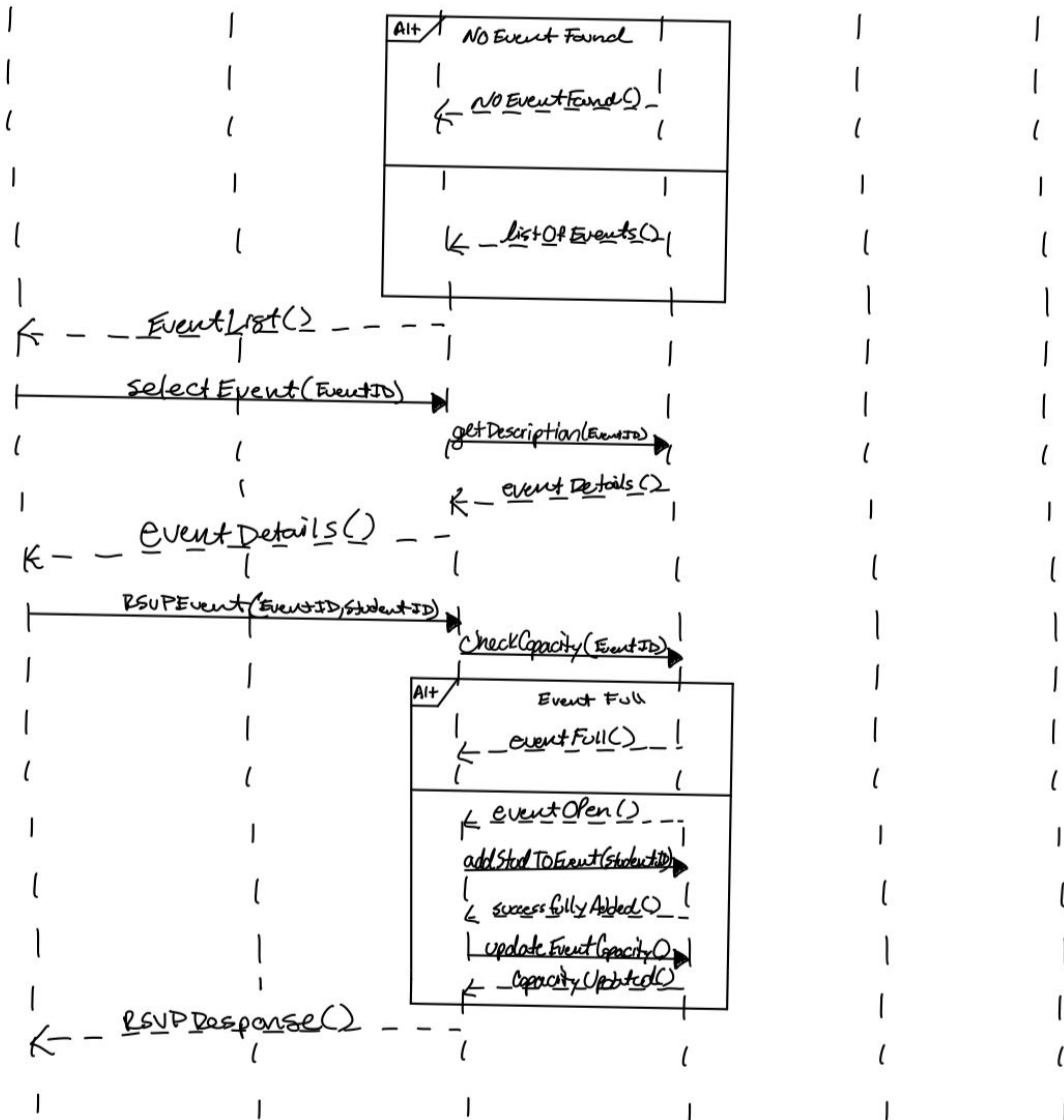
Version	Date	Description	Author
1.0	April 13, 2018	First draft of the software architecture diagram	Team
2.0	April 20, 2018	First draft of the use-class realization and the design class diagram	Team
3.0	May 1, 2018	Final review for submission	Team

1. Software Architecture



2. Use Case Realizations





3. Design Class Diagrams

