



TRƯỜNG ĐẠI HỌC FPT

# FPT UNIVERSITY

## Capstone Project Document

---

### Build Extracurricular Activity Management System for University

<b>Group FA20SE02</b>	
<b>Group Members</b>	Phạm Đức Bình – SE130691 Trần Minh Chiến – SE130681 Phan Hoàng Minh Luân – SE63241
<b>Supervisor</b>	Kiều Trọng Khánh
<b>Project Code</b>	EAMS

- Ho Chi Minh City, 14 September, 2020 -

# Table of Contents

Table of Contents .....	2
List of Tables .....	6
List of Figures .....	10
Definition and Acronyms .....	13
I. Project Introduction .....	14
1. Overview .....	14
1.1 Project Information .....	14
1.2 Project Team .....	14
2. Product Background .....	14
3. Existing Systems .....	14
3.1 Manual Solution .....	14
3.2 SVOnline .....	15
4. Business Opportunity .....	15
5. Software Product Vision .....	15
6. Project Scope & Limitations .....	16
6.1 Major Features .....	16
6.2 Limitations & Exclusions .....	18
II. Project Management Plan .....	18
1. Overview .....	18
1.1 WBS & Estimation .....	18
1.2 Project Objectives .....	19
1.3 Project Risks .....	19
2. Management Approach .....	19
2.1 Project Process .....	19
2.2 Quality Management .....	20
2.3 Training Plan .....	21
3. Project Communication .....	21
3.1 Communication Plan .....	21
3.2 External Interface .....	21
4. Configuration Management .....	21
4.1 Tools & Infrastructures .....	21
4.2 Document Management .....	22

4.3	Source Code Management .....	22
III.	Software Requirement Specification .....	22
1.	Overall Description .....	22
1.1	User Requirement Overview .....	22
2.	User Requirements Specification.....	23
2.1	Use Case.....	23
2.2	Use Case Specifications .....	27
3.	Functional Requirements.....	163
3.1	System Functional Overview .....	163
4.	Non-Functional Requirements.....	169
4.1	External Interfaces.....	169
4.2	Quality Attributes .....	170
IV.	Software Design Description .....	171
1.	Overall Description .....	171
1.1	Assumptions .....	171
1.2	Design Constraints .....	172
2.	System Architecture Design.....	172
3.	Component Diagram.....	173
4.	Activity Diagram.....	174
4.1	Register event.....	174
4.2	Check-in event.....	175
4.1	Manage Event Proposals .....	177
5.	System Detailed Design .....	178
5.1	Class Diagram.....	178
5.2	Class Specification .....	180
5.3	State Machine Diagrams.....	191
5.4	Sequence Diagrams .....	193
6.	Data & Database Design .....	196
6.1	Database Design .....	197
6.2	Data File Design .....	203
7.	Algorithm .....	203
7.1	Calculate contribution grade .....	203
7.2	Student assessment process and ranking .....	205

7.3	Check-in processes	206
V.	Software Testing Documentation	207
1.	Overall Description	207
1.1	Test Model	207
1.2	Testing Levels	207
2.	Test Plan	207
2.1	Test Stages	207
2.2	Resources	207
2.3	Test Milestones	208
3.	Test Cases	208
4.	Test Reports	208
VI.	Release Package & User Guides	208
1.	Deliverable Package	208
1.1.	Source codes & documents	208
2.	Installation Guide	208
2.1.	System requirements at server	208
2.1.1.	Hardware requirement	209
2.1.2.	Software requirement	209
2.2.	Installation Instruction	209
2.2.1.	Prepare docker image at local	209
2.2.2.	Push docker image to registry	210
2.2.3.	Apply new version in production	210
2.2.3.1.	At server side	210
2.2.3.2.	At client side	211
3.	Installation guide without using docker and kubernetes	211
4.	User Manual	212
4.1.	Terms and definitions	212
4.2.	System requirement	212
4.2.1.	Hardware requirement	212
4.2.2.	Software requirement	212
4.3.	Application Usage	212
4.3.1.	Overview	212
4.3.2.	Features	213

4.4. Troubleshooting.....	237
4.4.1. Student login using their personal email not in university domain .....	237
4.4.2. Student assistant mobile phone cannot access camera to scan QR .....	238
VII. Appendix .....	238
1. OMG Unified Modeling Language™ (OMG UML) Superstructure .....	238
2. UML Diagram .....	238
3. HMAC-based One-time Password algorithm.....	238
4. Time-based One-time Password algorithm .....	239
5. ReactJS .....	239
6. NestJS.....	239
7. Material-UI.....	239
8. Firebase.....	239
9. Cloudmersive .....	239
10. Google PWA Training .....	239

## List of Tables

Table 1.	<Use Case> Get Public Achievement.....	29
Table 2.	<Use Case> Get Public Shareable Link.....	30
Table 3.	<Use Case> Export Public Achievement .....	32
Table 4.	<Use Case> Login .....	33
Table 5.	<Use Case> Get Student Achievement.....	36
Table 6.	<Use Case> Get Available Events .....	37
Table 7.	<Use Case> Get Event Information .....	39
Table 8.	<Use Case> Register Event .....	40
Table 9.	<Use Case> Get Event QR.....	42
Table 10.	<Use Case> Scan QR.....	44
Table 11.	<Use Case> Give Event Feedbacks Online .....	45
Table 12.	<Use Case> Give Event Feedbacks Offline .....	47
Table 13.	<Use Case> Get Grade Histories .....	49
Table 14.	<Use Case> Get Achievement List .....	50
Table 15.	<Use Case> Get Shareable Link.....	52
Table 16.	<Use Case> Export Achievement .....	53
Table 17.	<Use Case> Get Profile.....	54
Table 18.	<Use Case> Update Profile .....	55
Table 19.	<Use Case> Get Event Proposals List .....	58
Table 20.	<Use Case> Get Event Proposals Details .....	61
Table 21.	<Use Case> Create Event Proposals.....	64
Table 22.	<Use Case> Update Event Proposals .....	67
Table 23.	<Use Case> Delete Event Proposals.....	70
Table 24.	<Use Case> Submit Event Proposals .....	73
Table 25.	<Use Case> Retract Event Proposals .....	76
Table 26.	<Use Case> Add Planned Task .....	78
Table 27.	<Use Case> Get Planned Tasks List .....	80
Table 28.	<Use Case> Delete Planned Task .....	82
Table 29.	<Use Case> Update Planned Task.....	84
Table 30.	<Use Case> Get Announcements List .....	85
Table 31.	<Use Case> Get Announcement .....	86
Table 32.	<Use Case> Get Post .....	87
Table 33.	<Use Case> Get Tasks List .....	89
Table 34.	<Use Case> Log Out .....	90
Table 35.	<Use Case> Get Events List .....	94
Table 36.	<Use Case> Get Event .....	97
Table 37.	<Use Case> Create Event .....	99
Table 38.	<Use Case> Update Event .....	102
Table 39.	<Use Case> Delete Event .....	104
Table 40.	<Use Case> Get Feedback Questions List .....	105
Table 41.	<Use Case> Add Feedback Question .....	106
Table 42.	<Use Case> Update Feedback Question .....	108

Table 43.	<Use Case> Delete Feedback Question .....	109
Table 44.	<Use Case> Get Event Tasks List .....	111
Table 45.	<Use Case> Add Task .....	114
Table 46.	<Use Case> Delete Task .....	116
Table 47.	<Use Case> Delete Task .....	118
Table 48.	<Use Case> Approve Event Proposals .....	121
Table 49.	<Use Case> Reject Event Proposals .....	122
Table 50.	<Use Case> Finish Event .....	124
Table 51.	<Use Case> Evaluate Event Grades.....	127
Table 52.	<Use Case> Assign Event Permission .....	128
Table 53.	<Use Case> Revoke Event Permission .....	130
Table 54.	<Use Case> Import Students.....	131
Table 55.	<Use Case> Register Manually.....	132
Table 56.	<Use Case> Check-in Manually.....	134
Table 57.	<Use Case> Check-out Manually.....	135
Table 58.	<Use Case> Feedback Manually.....	136
Table 59.	<Use Case> Export Students .....	138
Table 60.	<Use Case> Get Participations List.....	139
Table 61.	<Use Case> Get Staffs List.....	139
Table 62.	<Use Case> Add Staff.....	141
Table 63.	<Use Case> Get Student Assessment .....	142
Table 64.	<Use Case> Export Feedback Data.....	143
Table 65.	<Use Case> Create Achievement.....	145
Table 66.	<Use Case> Update Achievement.....	147
Table 67.	<Use Case> Get Announcements List .....	148
Table 68.	<Use Case> Get Announcement .....	149
Table 69.	<Use Case> Send Announcement .....	150
Table 70.	<Use Case> Create Announcement .....	152
Table 71.	<Use Case> Update Announcement .....	153
Table 72.	<Use Case> Look Up Student .....	154
Table 73.	<Use Case> Get Posts List .....	155
Table 74.	<Use Case> Get Post .....	156
Table 75.	<Use Case> Create Post .....	158
Table 76.	<Use Case> Delete Post .....	159
Table 77.	<Use Case> Update Post.....	160
Table 78.	<Use Case> Get Semester Report.....	161
Table 79.	<Use Case> Calculate Student Ranking.....	163
Table 80.	<Class Diagram Attributes> PolicyDocument .....	180
Table 81.	<Class Diagram Methods> PolicyDocument .....	180
Table 82.	<Class Diagram Attributes> GradeCriteria.....	180
Table 83.	<Class Diagram Methods> GradeCriteria.....	180
Table 84.	<Class Diagram Attributes> GradeSubCriteria.....	180
Table 85.	<Class Diagram Methods> GradeSubCriteria .....	181
Table 86.	<Class Diagram Attributes> GradeHistory .....	181

Table 87.	<Class Diagram Methods> GradeHistory .....	181
Table 88.	<Class Diagram Attributes> AchievementTemplate .....	181
Table 89.	<Class Diagram Methods> AchievementTemplate .....	181
Table 90.	<Class Diagram Attributes> Achievement .....	182
Table 91.	<Class Diagram Methods> Achievement .....	182
Table 92.	<Enumeration> FetchFromEnum .....	182
Table 93.	<Class Diagram Attributes> AchievementStudentHistory .....	182
Table 94.	<Class Diagram Methods> AchievementStudentHistory .....	182
Table 95.	<Class Diagram Attributes> AchievementStudent .....	183
Table 96.	<Class Diagram Methods> AchievementStudent .....	183
Table 97.	<Class Diagram Attributes> Student .....	183
Table 98.	<Class Diagram Methods> Student .....	183
Table 99.	<Class Diagram Attributes> StudentAnnouncement .....	183
Table 100.	<Class Diagram Methods> StudentAnnouncement .....	183
Table 101.	<Class Diagram Attributes> Announcement .....	184
Table 102.	<Class Diagram Methods> Announcement .....	184
Table 103.	<Class Diagram Attributes> User .....	184
Table 104.	<Class Diagram Methods> User .....	184
Table 105.	<Enumeration> SystemRoleEnum .....	184
Table 106.	<Class Diagram Attributes> UserPermission .....	184
Table 107.	<Class Diagram Methods> UserPermission .....	184
Table 108.	<Class Diagram Attributes> Permission .....	185
Table 109.	<Class Diagram Methods> Permission .....	185
Table 110.	<Class Diagram Attributes> EventStudent .....	185
Table 111.	<Class Diagram Methods> EventStudent .....	185
Table 112.	<Class Diagram Attributes> Participation .....	185
Table 113.	<Class Diagram Methods> Participation .....	186
Table 114.	<Class Diagram Attributes> FeedbackAnswer .....	186
Table 115.	<Class Diagram Methods> FeedbackAnswer .....	186
Table 116.	<Class Diagram Attributes> FeedbackQuestion .....	186
Table 117.	<Class Diagram Methods> FeedbackQuestion .....	186
Table 118.	<Enumeration> FeedbackQuestionEnum .....	186
Table 119.	<Class Diagram Attributes> Event .....	188
Table 120.	<Class Diagram Methods> Event .....	188
Table 121.	<Enumeration> EventStatus .....	188
Table 122.	<Class Diagram Attributes> Semester .....	188
Table 123.	<Class Diagram Methods> Semester .....	188
Table 124.	<Class Diagram Attributes> Post .....	189
Table 125.	<Class Diagram Methods> Post .....	189
Table 126.	<Class Diagram Attributes> Task .....	189
Table 127.	<Class Diagram Methods> Task .....	189
Table 128.	<Class Diagram Attributes> SubTask .....	189
Table 129.	<Class Diagram Methods> SubTask .....	189
Table 130.	<Class Diagram Attributes> Assignee .....	189

Table 131.	<Class Diagram Methods> Assignee .....	190
Table 132.	<Physical Diagram> policy_document .....	197
Table 133.	<Physical Diagram> grade_criteria .....	197
Table 134.	<Physical Diagram> grade_sub_criteria .....	197
Table 135.	<Physical Diagram> grade_history.....	197
Table 136.	<Physical Diagram> student_assessment.....	198
Table 137.	<Physical Diagram> achievement_template .....	198
Table 138.	<Physical Diagram> achievement_student_history .....	198
Table 139.	<Physical Diagram> student.....	198
Table 140.	<Physical Diagram> announcement_template.....	198
Table 141.	<Physical Diagram> achievement .....	199
Table 142.	<Physical Diagram> achievement_student.....	199
Table 143.	<Physical Diagram> participation .....	200
Table 144.	<Physical Diagram> event_student .....	200
Table 145.	<Physical Diagram> announcement .....	200
Table 146.	<Physical Diagram> event_announcement .....	200
Table 147.	<Physical Diagram> user .....	200
Table 148.	<Physical Diagram> feedback_answer .....	201
Table 149.	<Physical Diagram> event.....	201
Table 150.	<Physical Diagram> event_permission .....	201
Table 151.	<Physical Diagram> post .....	202
Table 152.	<Physical Diagram> task .....	202
Table 153.	<Physical Diagram> sub_task.....	202
Table 154.	<Physical Diagram> assignee .....	202
Table 155.	<Physical Diagram> semester .....	203
Table 156.	<Physical Diagram> feedback_question .....	203
Table 157.	Data File Design.....	203

## List of Figures

Figure 1.	EAMS Major Features .....	17
Figure 2.	<Reference> Scrum Framework .....	20
Figure 3.	<Use Case Overview> EAMS Use Case Diagram .....	24
Figure 4.	<Use Case Overview> Unauthenticated User .....	28
Figure 5.	<Use Case> Get Public Achievement .....	28
Figure 6.	<Use Case> Get Public Shareable Link .....	29
Figure 7.	<Use Case> Export Public Achievement .....	31
Figure 8.	<Use Case> Login .....	32
Figure 9.	<Use Case Overview> Student Use Cases .....	34
Figure 10.	<Use Case> Get Student Achievement .....	35
Figure 11.	<Use Case> Get Available Events .....	36
Figure 12.	<Use Case> Get Event Information .....	38
Figure 13.	<Use Case> Register Event .....	39
Figure 14.	<Use Case> Get Event QR .....	41
Figure 15.	<Use Case> Scan QR .....	42
Figure 16.	<Use Case> Give Event Feedbacks Online .....	44
Figure 17.	<Use Case> Give Event Feedbacks Offline .....	46
Figure 18.	<Use Case> Get Grades Histories .....	47
Figure 19.	<Use Case> Get Achievements List .....	49
Figure 20.	<Use Case> Get Shareable Link .....	51
Figure 21.	<Use Case> Export Achievement .....	52
Figure 22.	<Use Case> Get Profile .....	53
Figure 23.	<Use Case> Update Profile .....	54
Figure 24.	<Use Case> Get Event Proposals List .....	55
Figure 25.	<Use Case> Get Event Proposals Details .....	58
Figure 26.	<Use Case> Create Event Proposals .....	61
Figure 27.	<Use Case> Update Event Proposals .....	64
Figure 28.	<Use Case> Delete Event Proposals .....	67
Figure 29.	<Use Case> Submit Event Proposals .....	70
Figure 30.	<Use Case> Retract Event Proposals .....	73
Figure 31.	<Use Case> Add Planned Tasks .....	76
Figure 32.	<Use Case> Get Planned Tasks List .....	79
Figure 33.	<Use Case> Delete Planned Task .....	80
Figure 34.	<Use Case> Update Planned Task .....	82
Figure 35.	<Use Case> Get Announcements List .....	84
Figure 36.	<Use Case> Get Announcement .....	85
Figure 37.	<Use Case> Get Post .....	87
Figure 38.	<Use Case> Get Tasks List .....	88
Figure 39.	<Use Case Overview> Authenticated User .....	89
Figure 40.	<Use Case> Log Out .....	89
Figure 41.	<Use Case Overview> Staff Use Cases .....	91
Figure 42.	<Use Case> Get Events List .....	92

Figure 43.	<Use Case> Get Event .....	95
Figure 44.	<Use Case> Create Event .....	97
Figure 45.	<Use Case> Update Event .....	100
Figure 46.	<Use Case> Delete Event .....	102
Figure 47.	<Use Case> Get Feedback Questions List .....	104
Figure 48.	<Use Case> Add Feedback Question .....	105
Figure 49.	<Use Case> Update Feedback Question .....	107
Figure 50.	<Use Case> Delete Feedback Question .....	108
Figure 51.	<Use Case> Get Event Tasks List .....	109
Figure 52.	<Use Case> Add Task .....	111
Figure 53.	<Use Case> Update Task .....	114
Figure 54.	<Use Case> Delete Task .....	116
Figure 55.	<Use Case> Approve Event Proposals .....	118
Figure 56.	<Use Case> Reject Event Proposals .....	121
Figure 57.	<Use Case> Finish Event .....	123
Figure 58.	<Use Case> Evaluate Event Grades .....	124
Figure 59.	<Use Case> Assign Event Permission .....	127
Figure 60.	<Use Case> Revoke Event Permission .....	129
Figure 61.	<Use Case> Import Students .....	130
Figure 62.	<Use Case> Register Manually .....	131
Figure 63.	<Use Case> Check-in Manually .....	132
Figure 64.	<Use Case> Check-out Manually .....	134
Figure 65.	<Use Case> Feedback Manually .....	135
Figure 66.	<Use Case> Export Students .....	137
Figure 67.	<Use Case> Get Participations List .....	138
Figure 68.	<Use Case> Get Staffs List .....	139
Figure 69.	<Use Case> Add Staff .....	140
Figure 70.	<Use Case> Get Student Assessment .....	141
Figure 71.	<Use Case> Export Feedback Data .....	142
Figure 72.	<Use Case> Create Achievement .....	144
Figure 73.	<Use Case> Update Achievement .....	145
Figure 74.	<Use Case> Get Announcements List .....	147
Figure 75.	<Use Case> Get Announcement .....	148
Figure 76.	<Use Case> Send Announcement .....	149
Figure 77.	<Use Case> Create Announcement .....	151
Figure 78.	<Use Case> Update Announcement .....	152
Figure 79.	<Use Case> Look Up Student .....	153
Figure 80.	<Use Case> Get Posts List .....	154
Figure 81.	<Use Case> Get Post .....	156
Figure 82.	<Use Case> Create Post .....	157
Figure 83.	<Use Case> Delete Post .....	158
Figure 84.	<Use Case> Update Post .....	159
Figure 85.	<Use Case> Get Semester Report .....	160
Figure 86.	<Use Case Overview> System Handler .....	161

Figure 87.	<Use Case> Calculate Student Ranking.....	162
Figure 88.	<Screen Flow> Student Application .....	164
Figure 89.	<Screen Flow> Staff Application .....	165
Figure 90.	Entity Relationship Diagram (without attributes) .....	167
Figure 91.	Entity Relationship Diagram .....	168
Figure 92.	Architecture Diagram.....	172
Figure 93.	Component Diagram.....	173
Figure 94.	<Activity Diagram> Register Event.....	174
Figure 95.	<Activity Diagram> Check-in Event.....	175
Figure 96.	<Activity Diagram> Manage Event Proposals .....	177
Figure 97.	Class Diagram.....	179
Figure 98.	<State Machine Diagram> Event Participation.....	191
Figure 99.	<State Machine Diagram> Event State .....	192
Figure 100.	<Sequence Diagram> Create and send announcement .....	194
Figure 101.	<Sequence Diagram> Event Grading .....	195
Figure 102.	Physical Diagram.....	196

## Definition and Acronyms

Acronym	Definition
BR	Business Rule
ERD	Entity Relationship Diagram
UC	Use Case
API	Application Program Interface
EAMS	Extracurricular Activities Management System

# I. Project Introduction

## 1. Overview

### 1.1 Project Information

- Project name: Build Extracurricular Activity Management System for University
- Project code: EAMS
- Group name: FA20SE02
- Software type: Web Application

### 1.2 Project Team

#### 1.2.1 Supervisor

Full Name	Email	Phone Number	Title
Kieu Trong Khanh	khanhkt@fe.edu.vn	0908 133 304	Lecturer

#### 1.2.2 Team Members

Full Name	Email	Mobile	Role
Pham Duc Binh	binhpdse130691@fpt.edu.vn	0943 620 820	Leader
Phan Hoang Minh Luan	luanphmse63241@fpt.edu.vn	0902 690 743	Member
Tran Minh Chien	chientmse130681@fpt.edu.vn	0969 982 324	Member

## 2. Product Background

Many universities in Vietnam currently evaluate students' progress based on their academic results as well as their performance in extracurricular activities. Whereas academic results can be measured by tests scores, in many universities, extracurricular policies can be ambiguous, and students are usually unaware of them. This makes students less enthusiastic in participating social events.

Currently, events are usually planned, organized, managed, evaluated separately using platforms such as Facebook to publish announcements, Google Form to manage registrations, Microsoft Excel to make reports. Moreover, events are arranged by both students and staffs of the university, which makes events' management and data such as participations and feedbacks decentralized.

As a result, performance evaluation related to students' extracurricular activities are usually done intuitively by a few university staffs. This leads to environment where students may feel unfair and that their efforts are not recognized, which negatively influences the students' motivation to take part in university events. Hence, student progress in universities is not assessed comprehensively and objectively.

## 3. Existing Systems

### 3.1 Manual Solution

Currently, in many universities where manual solutions is used, staffs have various obstacles when organising events. Firstly, they usually initiate an event plan using a prepared word template. Afterwards, for the students to know about the event, they usually publish posts on social media platforms such as Facebook or Twitter. Then, if students want to sign up for

the event, they have to submit their information on Google Form. This will later be managed by the staffs using Google Sheets.

From the staffs' perspective, this process is rather time-consuming and difficult to manage. Whereas for students, it is difficult to track the latest updates of events.

### 3.2 SVOnline

This is a SaaS that is being used in many public universities in Vietnam. It provides extracurricular activities management features as well as many others such as academic results management, rental bike and taxi services.

With this system, while students are graded for participating events, what they contribute to them and how they are graded are recorded manually. This is an approach that requires a lot of effort for staffs, which leaves rooms for human mistakes. Moreover, their personal information can be accessed by virtually anyone. Additionally, this application uses a static QR to take attendance for students' event participation, which from the university staffs' point of view, it can be difficult to prevent students from taking attendance for one another.

Reference for SVOnline: <https://svonline.vn/dang-nhap.html?rdr=dhdt>

Reference for SVOnline usage in ICTU: <https://sinhvien.ictu.edu.vn/hoat-dong/>

## 4. Business Opportunity

This presents an opportunity to build a system where staff can track the status, manage details of individual events as well as monitor numbers of registrants, participants, attendances and feedbacks. Furthermore, by only allowing students to access their grades if they have an account, this system can protect their personal information from being exposed to the public.

Moreover, the system provides a way for students to submit event plans which can include the event details such as budget, event organizers as well as their workloads. This means that students' efforts are recognized more comprehensively because their participation and contributions will be recorded in the system.

Furthermore, by taking advantage of the data related to students' participation and contribution, the system provides staffs with a more quantitative way to decide which students will receive certificates and rewards. These achievements can later be used to share with employers, which makes the students more competitive in the job market.

## 5. Software Product Vision

For event organizers in universities, Extracurricular Activity Management System which is a web application that will make the event management processes straightforward and centralized by providing features such as managing event details and announcements, keeping track of event registrations, participants, feedbacks and reports. This saves them time and efforts when compared to the manual solution where, for example, they usually get the students signature to check in, and then transfer the participants list to Google Sheets to make a report. Furthermore, the registration and participation processes will also be made easy and cheat-preventive using a smartphone-enabled web application with dynamically generated

QR for each student for individual event. Additionally, students' grades will be kept private by using an authentication layer, which prevents others from getting it without permission.

Such system will provide the following features:

- Semesters and events management
- Student's events participation management
- Achievements management
- Student's events planning management

## 6. Project Scope & Limitations

### 6.1 Major Features

FE-01: Allow staffs to manage event information such as details, announcements and posts.

FE-02: Allow staffs to view reports about event and semester.

FE-03: Allow staffs to scan QR and assign scan QR permission to students as event assistants

FE-04: Allow staffs to manage achievements

FE-05: Allow staffs to approve, reject and provide feedbacks to students' event proposals

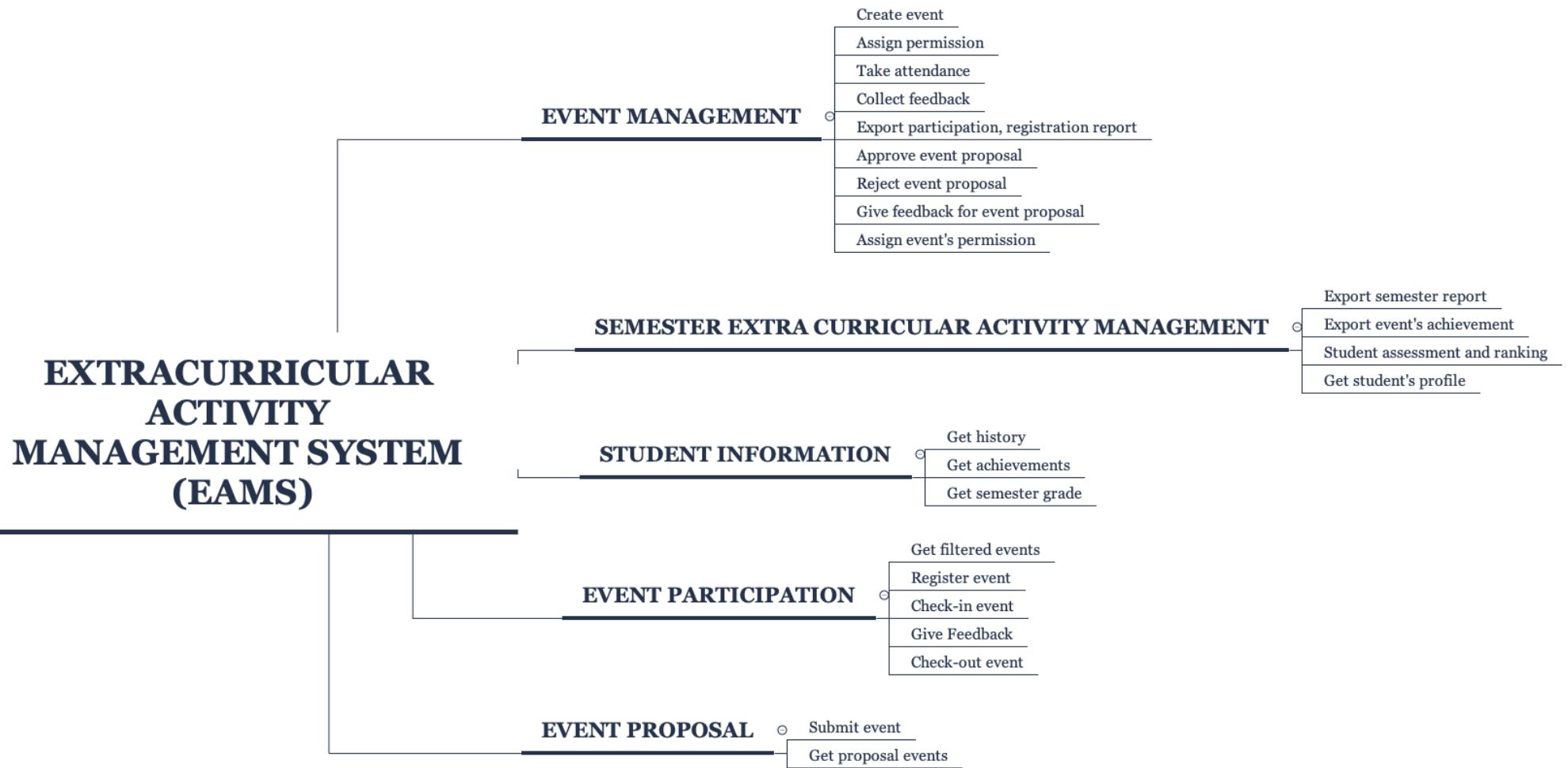
FE-06: Allow students to register an event, check-in, check-out and give feedback for that event.

FE-07: Allow students to view their extracurricular activities

FE-08: Allow students to view and filter events.

FE-09: Allow students and guest to view students' achievements.

FE-10: Allow students to submit events plans which staff can manage



*Figure 1. EAMS Major Features*

## 6.2 Limitations & Exclusions

- LI-1: EAMS only recognizes student information through QR Code, so it has a potential risk that students ask someone else to check-in for them.
- LI-2: EAMS can only send notification via Emails. It cannot send push notification
- LI-3: EAMS assumes that students have Internet connection
- LI-4: EAMS assumes that students have university email accounts
- LI-5: EAMS only manage student's participation or contribution to events that are outside the system
- LI-6: EAMS does not take account of internal club activities to evaluate students' extracurricular grades
- LI-7: EAMS does not support events that require check-in multiple times

- EX-1: EAMS will only evaluate students on their performance in extracurricular activities
- EX-2: EAMS does not manage students
- EX-3: EAMS does not generate achievement automatically
- EX-4: EAMS does not provide features such as sending and receiving text messages, comments between event organizers and students.

## II. Project Management Plan

### 1. Overview

#### 1.1 WBS & Estimation

#	WBS Item	Complexity	Est. Effort
1	<b><i>Student Application</i></b>		
1.1	View Upcoming Events	Simple	2
1.2	Sign up for an event	Simple	1
1.3	Get event announcement	Simple	2
1.4	View Events' posts	Simple	2
1.5	Check-in using QR Code	Complex	15
1.6	Scan student's QR code	Complex	15
1.7	Give feedbacks for participated events	Medium	5
1.8	View and share certificates and rewards	Simple	3
1.9	Submit events plans	Complex	20

1.10	View submitted events		
1.11	Assign and update tasks for other students		
2	<b>Staff Application</b>		
2.1	Manage semesters	Complex	15
2.2	Manage events	Complex	20
2.3	Import/Export participants list	Medium	5
2.4	Allow staff to track progress of events	Simple	1
2.5	Manage grading policies	Complex	20
2.6	Support auto announcement by email	Complex	10
2.7	Manage student certificates and rewards	Complex	20
2.8	Manage student event plans	Complex	20
2.9	Grade and provide feedbacks for students plans	Medium	10

**Total Estimated Effort (man-days)** **200**

## 1.2 Project Objectives

#	Quality Stage	No. of Defects	% of Defect	Notes
1	Reviewing	5	11.7%	
2	Unit Test	30	68.7%	
4	System Test	3	6.9%	
5	User Acceptance Test	5	11.7%	

**Total** **43** **100%**

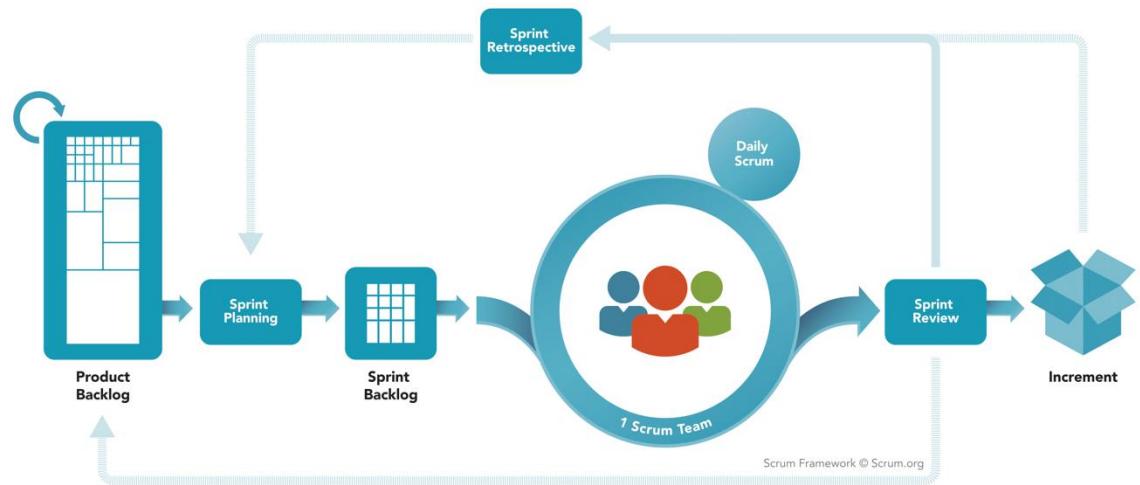
## 1.3 Project Risks

#	Risk Description	Impact	Possibility	Response Plans
1	Computer malfunctions	Medium	Medium	Use cloud services
2	Team member drops out	Critical	Medium	Reduce scope
3	Supervisor gets sick	Critical	Low	Request other lecturers for guidance

## 2. Management Approach

### 2.1 Project Process

# SCRUM FRAMEWORK



 Scrum.org

**Figure 2. <Reference> Scrum Framework**

***“Software in 30 Days” by Ken Schwaber and Jeff Sutherland***  
[\*\*\*https://www.scrum.org/resources/scrum-framework-poster\*\*\*](https://www.scrum.org/resources/scrum-framework-poster)

EAMS is developed using the Scrum model which is an agile framework for developing, delivering and maintaining products. Our team chooses this model for the following reasons:

- Because Scrum is suitable for small and medium-sized projects, it fits our team which consists of 3 members well.
- Each feature is assigned to only one member, which means that person has to complete all the steps: design, coding, implementing and testing. This helps us work independently yet can still be able to collaborate and help each other in every step of the process.
- Because our application is expected to be launched and used in a practical environment, the possibility of changing and updating user requirements is high. It is essential for us to have a flexible workflow and Scrum is suitable for this situation
- There are various techniques, technologies and framework for our team members to learn. Using this model can help us learn and develop simultaneously in quick manner

For this project, each sprint spans 5 days and the duration of our product backlog is 4 months

## 2.2 Quality Management

In order to maximize the project quality, our team will apply the following tactics to improve the project quality

- User acceptance tests as we develop more features
- Cross-review each other's code

### 2.3 Training Plan

Training Area	Participants	When, Duration	Waiver Criteria
NestJS	Everyone	14/09/2020. 1 week	Mandatory
React	Everyone	14/09/2020. 1 week	Mandatory
Docker, Jenkins	Everyone	14/09/2020. 1 week	Mandatory
Git, Gitsubmodule	Everyone	14/09/2020. 1 week	Mandatory

## 3. Project Communication

### 3.1 Communication Plan

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Computer	Team members	Internal Communication	Always	Slack
Phone	Supervisor	Upper Communication	Always	IMO
In Person	Team Members & Supervisor	Get everyone on the same page	Daily	

### 3.2 External Interface

Function	Contact Person (name, position)	Contact address (email, telephone)	Responsibility
Supervisor	Kieu Trong Khanh	khanhkt@fe.edu.vn 0908 133 304	<ul style="list-style-type: none"> <li>- Provide document template</li> <li>- Give instruction to project team</li> <li>- Review deliverables</li> <li>- Supervise project status</li> </ul>
End User	Nguyen Thi Thanh Thuy	<a href="mailto:thuyntt@fpt.edu.vn">thuyntt@fpt.edu.vn</a>	<ul style="list-style-type: none"> <li>- Provide end user's perspective</li> <li>- Provide requirements</li> </ul>

## 4. Configuration Management

### 4.1 Tools & Infrastructures

Programming languages	JavaScript/TypeScript
Framework/Library	NestJS, ReactJS
DBMS	MySQL
IDEs/Editors	Visual Studio Code

UML tools	Draw.io
Version Control	GitLab
Deployment server	Digital Ocean
Project management tool	Trello

#### 4.2 Document Management

The project documents will be managed using Microsoft OneDrive

#### 4.3 Source Code Management

The project source code will be managed using GitLab

### III. Software Requirement Specification

#### 1. Overall Description

##### 1.1 User Requirement Overview

###### 1.1.1 Guest (Unauthenticated User) Requirements

Guests are people who have not been identified by the system. They only have limited access to the system with the following features:

- Login
- View students' achievements

###### 1.1.2 Student Requirements

Students are authorized users, has a “student” role and have their own application to use the following features:

- View upcoming events.
- Sign up an event.
- Get event annotation.
- View event’s post.
- Check in by QR code that is generated from student’s mobile device.
- Feedback for their participated events.
- View grading history.
- View and share their certification/reward.
- Support for making and submit an event’s plan.
- View created or assigned events.
- Update and assign task for team members base on approved plan

###### 1.1.3 Staff Requirements

Staff are authorized users, has a “staff” role and have their own application to use the following features:

- Manage semesters.
- Manage events.
- Allow staff check in for student using mobile camera to scan QR code generating with camera.
- Import/export participants list.
- Allow manager to tracking progress for operating and upcoming events.

- Allow staff make reports for events, students and semester.
- Manage grading policy.
- Support auto announcement by email and notification.
- Manage student certification and reward.
- Manage student's plans.
- Approve or reject student's plan.
- Grading and feedback for students' plans.

## 2. User Requirements Specification

### 2.1 Use Case

#### 2.1.1 Use Case Diagram

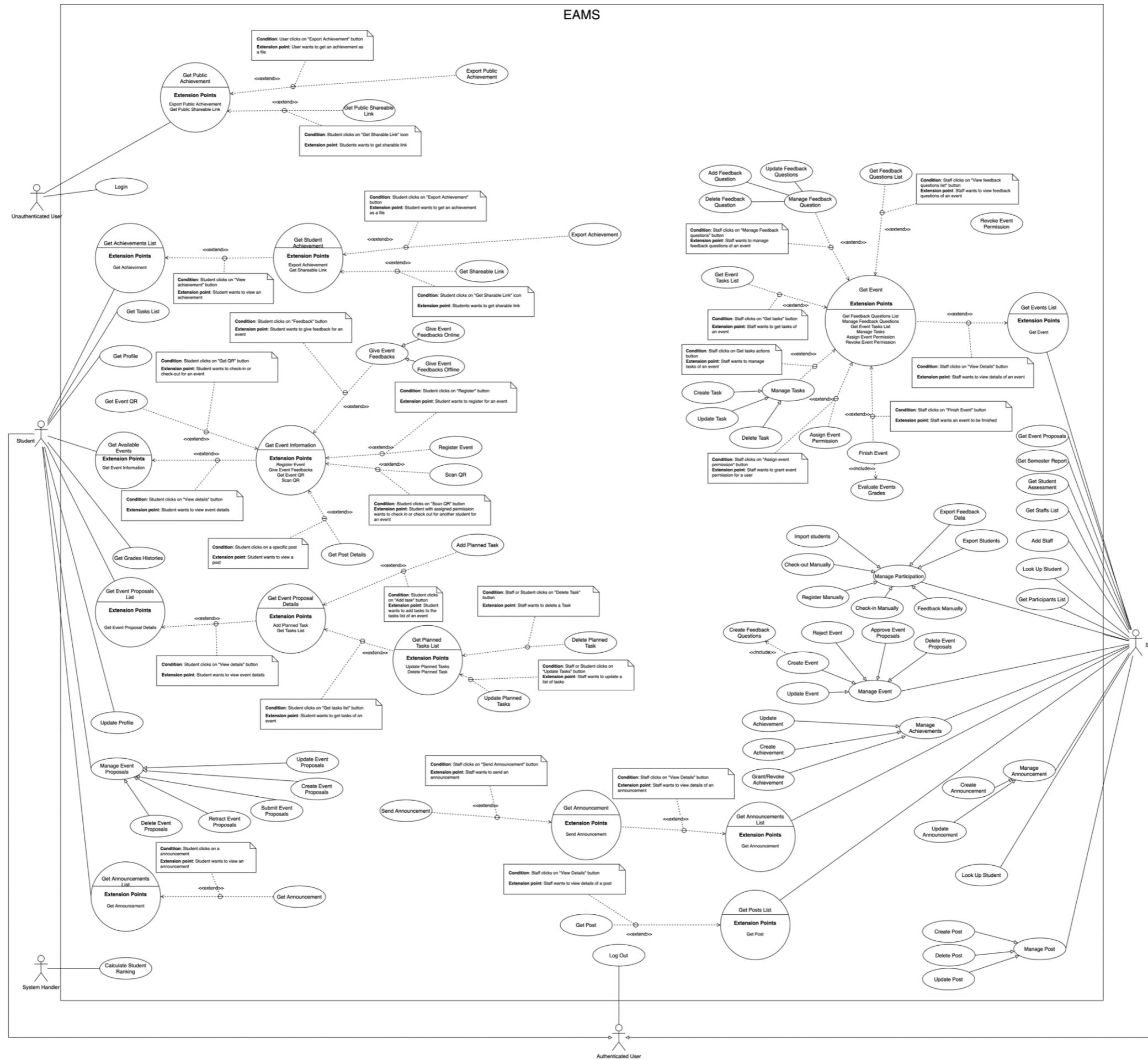


Figure 3. <Use Case Overview> EAMS Use Case Diagram

### 2.1.2 System Actors

#	Actor	Description
1	Unauthenticated User	Users have not logged in to EAMS
2	Student	Users have logged in to EAMS as student role
3	Staff	Users have logged in to EAMS as staff role
4	System Handler	EAMS system
5	Authenticated User	Users have logged in to EAMS

### 2.1.3 Use Cases List

ID	Use Case	Primary Actors	Secondary Actors
01	Get Public Achievement	Unauthenticated User	
02	Get Public Shareable Link	Unauthenticated User	
03	Export Public Achievement	Unauthenticated User	
04	Login	Unauthenticated User	
05	Get Student Achievement	Student	
06	Get Available Events	Student	
07	Get Event Information	Student	
08	Register Event	Student	
09	Get Event QR	Student	
10	Scan QR	Student	
11	Give Event Feedbacks Online	Student	
12	Give Event Feedbacks Offline	Student	
13	Get Grade Histories	Student	
14	Get Achievements List	Student	
15	Get Sharable Link	Student	
16	Export Achievement	Student	
17	Get Profile	Student	
18	Update Profile	Student	
19	Get Event Proposals List	Student	
20	Get Event Proposals Details	Student	
21	Create Event Proposals	Student	

22	Update Event Proposals	Student	
23	Delete Event Proposals	Student	
24	Submit Event Proposals	Student	
25	Retract Event Proposals	Student	
26	Add Planned Task	Student	
27	Get Planned Tasks List	Student	
28	Delete Planned Task	Student	
29	Update Planned Task	Student	
30	Get Announcements List	Student	
31	Get Announcement	Student	
32	Get Post	Student	
33	Get Tasks List	Student	
34	Log Out	Authenticated user	
35	Get Events List	Staff	
36	Get Event	Staff	
37	Create Event	Staff	
38	Update Event	Staff	
39	Delete Event	Staff	
40	Get Feedback Questions List	Staff	
41	Add Feedback Question	Staff	
42	Update Feedback Questions	Staff	
43	Delete Feedback Question	Staff	
44	Get Event Tasks List	Staff	
45	Add Task	Staff	
46	Update Task	Staff	
47	Delete Task	Staff	
48	Approve Event Proposals	Staff	
49	Reject Event Proposals	Staff	
50	Finish Event	Staff	
51	Evaluate Events Grades	Staff	

52	Assign Event Permission	Staff	
53	Revoke Event Permission	Staff	
54	Import Students	Staff	
55	Register Manually	Staff	
56	Check-in Manually	Staff	
57	Check-out Manually	Staff	
58	Feedback Manually	Staff	
59	Export Students	Staff	
60	Get Participants List	Staff	
61	Export Feedback Data	Staff	
62	Get Staffs Lists	Staff	
63	Add Staff	Staff	
64	Get Student Assessment	Staff	
65	Create Achievement	Staff	
66	Update Achievement	Staff	
67	Get Announcements List	Staff	
68	Get Announcement	Staff	
69	Send Announcement	Staff	
70	Create Announcement	Staff	
71	Update Announcement	Staff	
72	Look Up Student	Staff	
73	Get Posts List	Staff	
74	Get Post	Staff	
75	Create Post	Staff	
76	Delete Post	Staff	
77	Update Post	Staff	
78	Get Semester Report	Staff	
79	Calculate Student Ranking	System Handler	

## 2.2 Use Case Specifications

## 2.2.1 Unauthenticated User

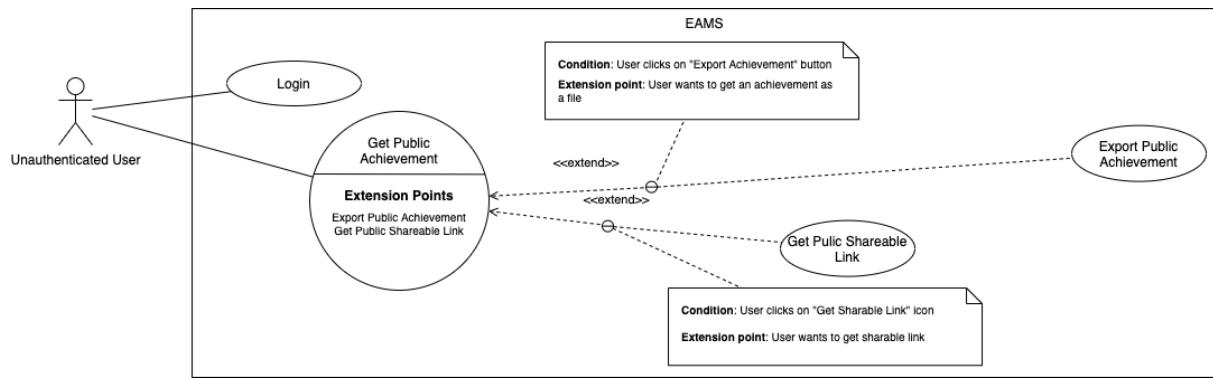


Figure 4. <Use Case Overview> Unauthenticated User

### 2.2.1.1 Get Public Achievement

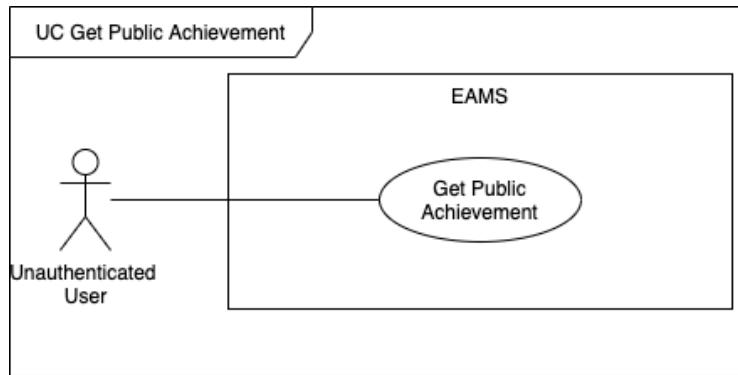


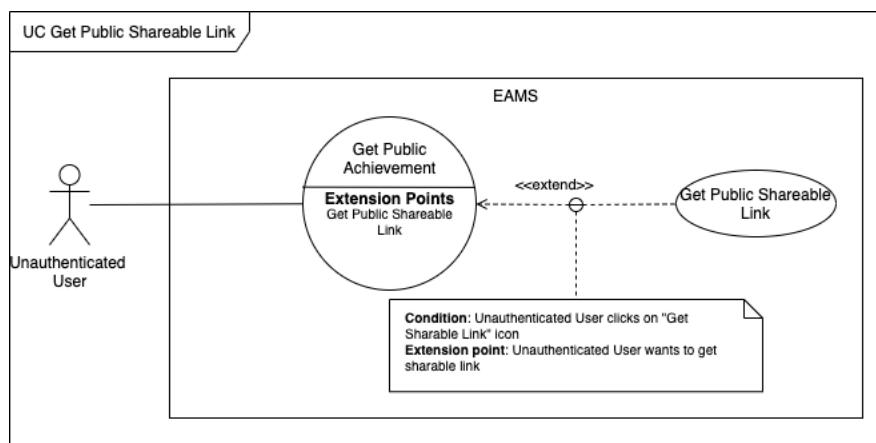
Figure 5. <Use Case> Get Public Achievement

ID and Name:	<b>UC-1. Get Public Achievement</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Unauthenticated User	Secondary Actors:	N/A
Description:	This feature allows users to share their extracurricular achievement to anyone, such as employers		
Trigger:	User send a request to get student achievement		
Preconditions:	N/A		
Post-conditions:	POST-1. User receives achievement details		
Normal Flow:	Step	Actor Action	System Response
	1	User requests an achievement detail	
			EAMS returns achievement detail [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response

		The achievement does not exist	EAMS returns that achievement does not exist
Priority:	Medium		
Frequency of Use:	Approximately 100 users, relies on the number of events that week.		
Business Rules:	<ul style="list-style-type: none"> <li>• In EAMS, after being graded for an event, participations list and contributions list will be handed over to award stage where staff will decide which one will receive achievement.</li> <li>• To enter this stage, student has checked-out if event requires check-out or checked-in if event doesn't. And event's grade of that student is larger than 0, whether that student is graded or not depends on staff in grading stage.</li> <li>• Achievements are granted for students for their performances in extracurricular activities. There are 2 types of achievements: <ul style="list-style-type: none"> <li>○ Achievements for participations and contributions: Students can earn this achievement by participating in or contributing to an event. Achievement of this type can only be granted for participating in or contributing to finished and graded events.</li> <li>○ Achievements for special reasons: Universities can grant students achievements for exceptional performances in extracurricular activities.</li> </ul> </li> <li>• Anyone can request student achievement details if they have URL of that achievement.</li> </ul>		
Other Information:	In case of internet connection failure, student cannot get achievement information		
Assumptions:	N/A		

**Table 1. <Use Case> Get Public Achievement**

#### 2.2.1.2 Get Public Shareable Link



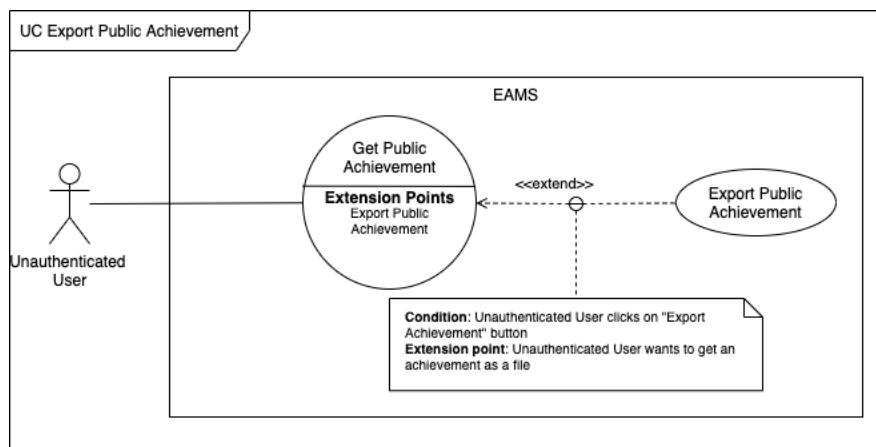
**Figure 6. <Use Case> Get Public Shareable Link**

ID and Name:	<b>UC-2. Get Public Shareable Link</b>
--------------	--

Created By:	LuanPHM		Date Created:	02/12/20		
Primary Actor:	Unauthenticated User		Secondary Actors:	N/A		
Description:	User requests to get a shareable link of an achievement					
Trigger:	Unauthenticated User requests to share an achievement					
Preconditions:	N/A					
Post-conditions:	POST-1. User receives a link.					
Normal Flow:	Step	Actor Action		System Response		
	1	User requests to get a shareable link				
				EAMS returns shareable link of that achievement		
Alternative Flows:	N/A					
Exceptions:	No	Cause		System Response		
	1	Achievement does not exist		EAMS returns that achievement does not exist		
Priority:	Medium					
Frequency of Use:	This feature is used most after an event has finished					
Business Rules:	<ul style="list-style-type: none"> <li>In EAMS, after being graded for an event, participations list and contributions list will be handed over to award stage where staff will decide which one will receive achievement.</li> <li>To enter this stage, student has checked-out if event requires check-out or checked-in if event doesn't. And event's grade of that student is larger than 0, whether that student is graded or not depends on staff in grading stage.</li> <li>Student can share their achievements to anyone</li> <li>Achievements are granted for students for their performances in extracurricular activities. There are 2 types of achievements: <ul style="list-style-type: none"> <li>Achievements for participations and contributions: Students can earn this achievement by participating in or contributing to an event. Achievement of this type can only be granted for participating in or contributing to finished and graded events.</li> <li>Achievements for special reasons: Universities can grant students achievements for exceptional performances in extracurricular activities.</li> </ul> </li> </ul>					
Other Information:	In case of internet connection failure, student cannot get shareable link					
Assumptions:	N/A					

**Table 2. <Use Case> Get Public Shareable Link**

### 2.2.1.3 Export Public Achievement



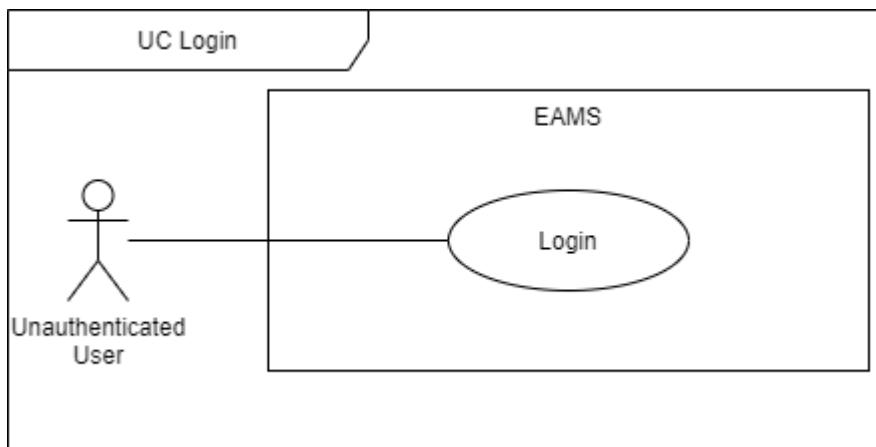
**Figure 7. <Use Case> Export Public Achievement**

ID and Name:	<b>UC-3. Export Public Achievement</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Unauthenticated User	Secondary Actors:	N/A
Description:	User requests to get achievement detail as PDF file.		
Trigger:	Unauthenticated User requests to download an achievement as a PDF file.		
Preconditions:	N/A		
Post-conditions:	POST-1. User receives an achievement as a PDF file.		
Normal Flow:	Step	Actor Action	System Response
	1	User sends request to download PDF file of an achievement	
			EAMS returns PDF file. [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	E1 Achievement does not exist	EAMS returns that achievement does not exist
Priority:	Medium		
Frequency of Use:	This feature is used most after an event has finished		
Business Rules:	<ul style="list-style-type: none"> <li>In EAMS, after being graded for an event, participations list and contributions list will be handed over to award stage where staff will decide which one will receive achievement.</li> <li>To enter this stage, student has checked-out if event requires check-out or checked-in if event doesn't. And event's grade of that</li> </ul>		

	<p>student is larger than 0, whether that student is graded or not depends on staff in grading stage.</p> <ul style="list-style-type: none"> <li>Achievement are public and can be shared through link or pdf file.</li> </ul>
Other Information:	<ul style="list-style-type: none"> <li>Every download request needs about 7-9s to accomplish.</li> <li>In case of internet connection failure, student cannot export achievement</li> </ul>
Assumptions:	User device is able to download and read pdf file and it has stable internet connection.

**Table 3.      <Use Case> Export Public Achievement**

#### 2.2.1.4 Login



**Figure 8.    <Use Case> Login**

ID and Name:	<b>UC-4. Login</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Unauthenticated User	Secondary Actors:	N/A
Description:	Determines user's identity, role and permissions, give them access to more features in the system		
Trigger:	Unauthorized User sends request to login		
Preconditions:	PRE-1. User is not authorized		
Post-conditions:	POST-1. User is identified and authorized, has access to the features that correspond to his or her role		
Normal Flow:	Step	Actor Action	System Response
	1	User sends a request to login	
			EAMS redirects user to the Authentication page where they can login using their university email
	2	User logs in using a university email	

		[Exception 1]	
			Authentication page redirects user back to EAMS
			EAMS determines user's identity, role and permission then redirects user to home page [Exception 2]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	User's email does not belong to the system's university	Authentication page shows user error 403 – Forbidden
	2	User's email is not in the system	EAMS returns to user that their account is not found
Priority:	High		
Frequency of Use:	Approximately 2000 users, average of one usage per 7 days. Peak usage load for this use case happens when an event is about to begin		
Business Rules:	<ul style="list-style-type: none"> <li>User logs in using an email account provided by the university in order to use system functions. If the email is in the system and has the appropriate role, EAMS returns a token for users and redirect them to their corresponding home page: <ul style="list-style-type: none"> <li>User logs in as "staff" on the admin application</li> <li>User logs in as "Student" on the student application</li> </ul> </li> <li>User's login session lasts for 7 days by default</li> </ul>		
Other Information:	<ul style="list-style-type: none"> <li>In order for a university account to be in the system, users accounts have to be imported into the database beforehand</li> <li>In case of Internet connection failure, students will not be able to login</li> </ul>		
Assumptions:	All students are given a university account		

**Table 4. <Use Case> Login**

## 2.2.2 Student

# EAMS

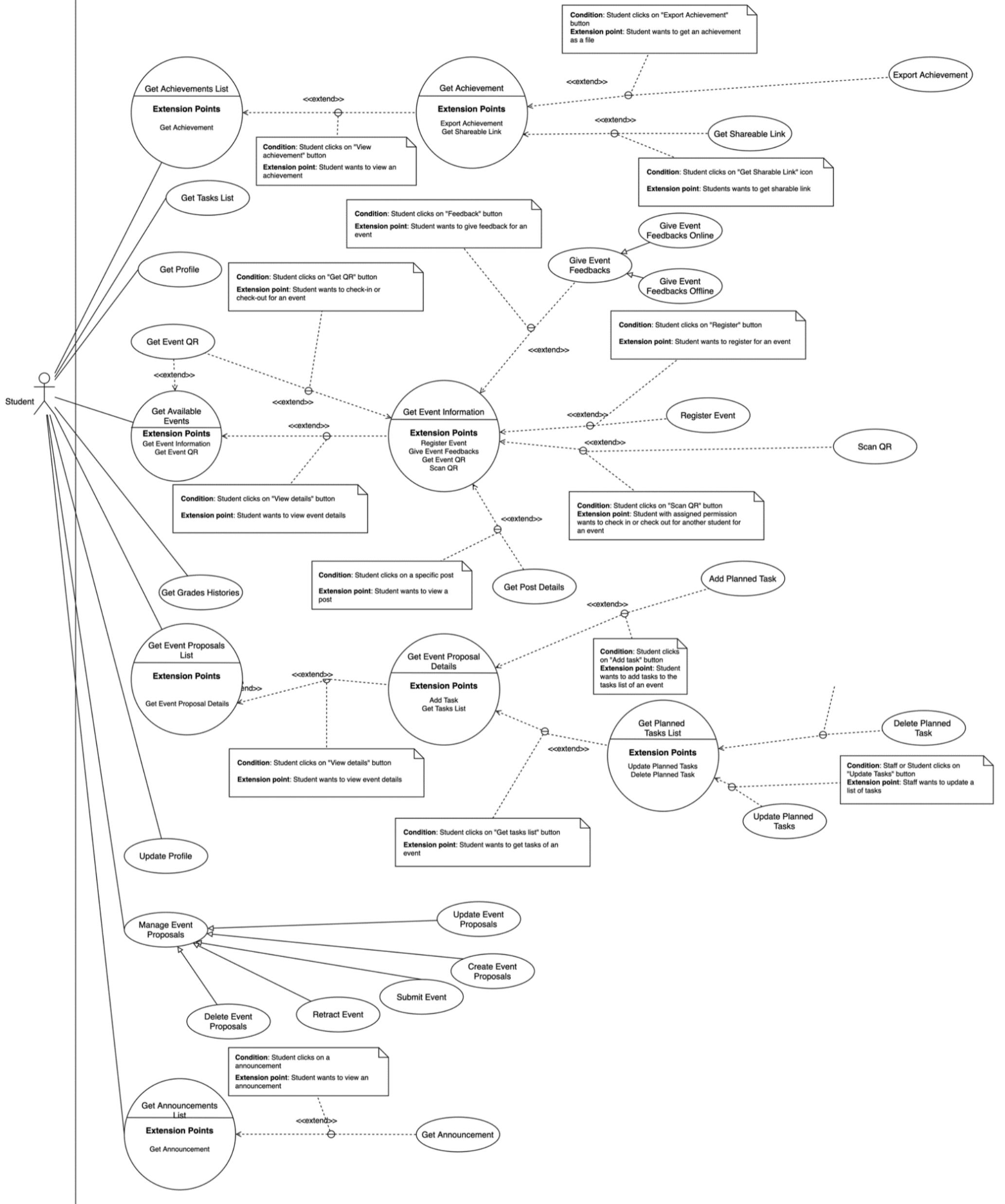
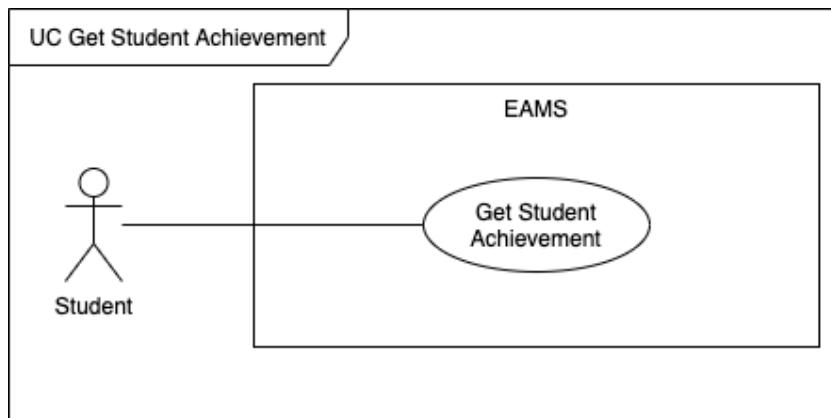


Figure 9. <Use Case Overview> Student Use Cases

### 2.2.2.1 Get Student Achievement



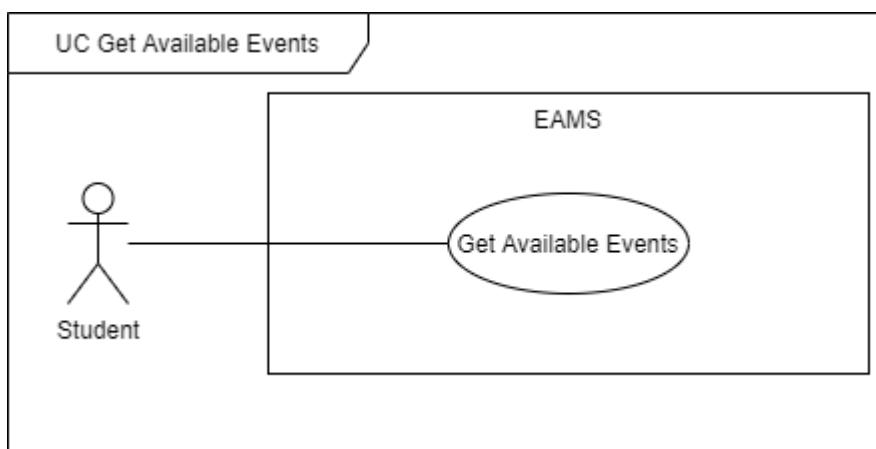
**Figure 10. <Use Case> Get Student Achievement**

ID and Name:	<b>UC-5. Get Student Achievement</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	This feature allows students to share their extracurricular achievement to anyone, such as employers		
Trigger:	User send a request to get student achievement		
Preconditions:	N/A		
Post-conditions:	POST-1. User receives achievement details		
Normal Flow:	Step	Actor Action	System Response
	1	User requests an achievement detail	
			EAMS returns achievement detail [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
		The achievement does not exist	EAMS returns that achievement does not exist
Priority:	Medium		
Frequency of Use:	Approximately 100 users, relies on the number of events that week.		
Business Rules:	<ul style="list-style-type: none"> <li>In EAMS, after being graded for an event, participations list and contributions list will be handed over to award stage where staff will decide which one will receive achievement.</li> <li>To enter this stage, student has checked-out if event requires check-out or checked-in if event doesn't. And event's grade of that</li> </ul>		

	<p>student is larger than 0, whether that student is graded or not depends on staff in grading stage.</p> <ul style="list-style-type: none"> <li>Achievements are granted for students for their performances in extracurricular activities. There are 2 types of achievements:           <ul style="list-style-type: none"> <li>Achievements for participations and contributions: Students can earn this achievement by participating in or contributing to an event. Achievement of this type can only be granted for participating in or contributing to finished and graded events.</li> <li>Achievements for special reasons: Universities can grant students achievements for exceptional performances in extracurricular activities.</li> </ul> </li> <li>Anyone can request student achievement details if they have URL of that achievement.</li> </ul>
Other Information:	In case of internet connection failure, student cannot get achievement information
Assumptions:	N/A

**Table 5. <Use Case> Get Student Achievement**

#### 2.2.2.2 Get Available Events



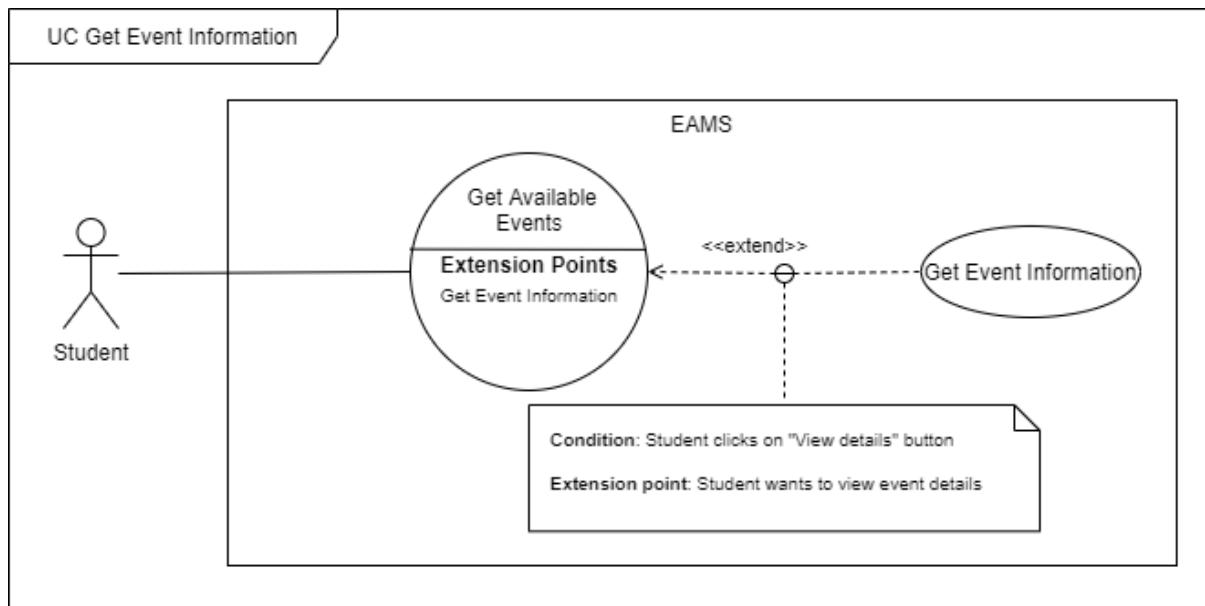
**Figure 11. <Use Case> Get Available Events**

ID and Name:	<b>UC-6. Get Available Events</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Get extracurricular events in university		
Trigger:	Student sends a request to get events list based on their status and title		
Preconditions:	PRE-1. User has logged in as Student role		
Post-conditions:	POST-1. A list of events is returned to student		
Normal Flow:	Step	Actor Action	System Response

	1	Student requests to get available events [Exception 1]	
		EAMS returns available events	
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	There is no available event for this student	EAMS returns to student that there is no available event for this student
Priority:	High		
Frequency of Use:	Approximately 2000 users per week. Peak usage load for this use case is when an event is about to begin		
Business Rules:	<ul style="list-style-type: none"> <li>In university, extracurricular activities are known as events. Students participate these events to be recorded as extracurricular activity history</li> <li>An available event is an event that was, is being or is going to be organized in the university. This also includes cancelled event. Student can only see published, finished or cancelled events when they request to get available events.</li> <li>It is possible for an event to restrict registrants, which means only students that are in the whitelist can participate. If event restricts register, student must be in registration whitelist to see the event. Registrants whitelist can be determined by staffs</li> <li>There are certain events that the organizer wants to restrict who can or cannot register. Restriction whitelists are used for this purpose. Restriction whitelists are determined by staffs. <ul style="list-style-type: none"> <li>Registration whitelist: Only students in this list can register to participate for registration restricted events. If a student is not in the registration whitelist of an event, he or she can not get the event information when getting available events</li> </ul> </li> </ul>		
Other Information:	In case of internet connection failure, student cannot finish this use case		
Assumptions:	N/A		

**Table 6. <Use Case> Get Available Events**

### 2.2.2.3 Get Event Information



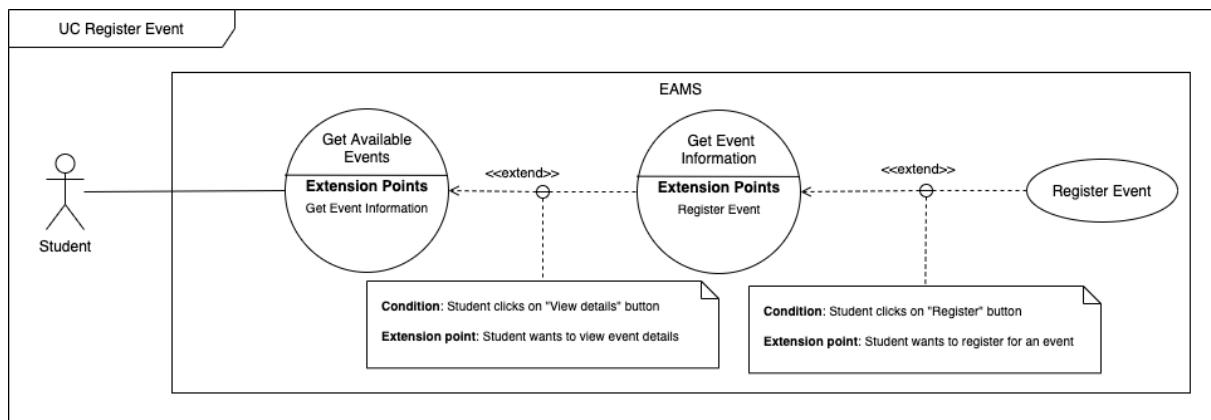
**Figure 12. <Use Case> Get Event Information**

ID and Name:	<b>UC-7. Get Event Information</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Get detailed event information and posts list of event		
Trigger:	Student sends a request to get detailed of an event		
Preconditions:	PRE-1. User has logged in as Student role		
Post-conditions:	POST-1. Event information is returned to student		
Normal Flow:	Step	Actor Action	System Response
	1	Student requests detail of an events [Exception 1]	
			EAMS returns details of an event
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	This event is not available for this student	EAMS returns to student that this event is not available
Priority:	High		
Frequency of Use:	Approximately 2000 users per week. Peak usage load for this use case is when an event is about to begin		
Business Rules:	<ul style="list-style-type: none"> <li>Student can only see details of published, finished or cancelled event.</li> <li>It is possible for an event to restrict registrants, which means only students that are in the whitelist can participate. If event restricts</li> </ul>		

	<p>register, student must be in registration whitelist to see the event. Registrants whitelist can be determined by staffs</p> <ul style="list-style-type: none"> <li>There are certain events that the organizer wants to restrict who can or cannot register. Restriction whitelists are used for this purpose. Restriction whitelists are determined by staffs.           <ul style="list-style-type: none"> <li>Registration whitelist: Only students in this list can register to participate for registration restricted events.</li> </ul> </li> </ul>
Other Information:	In case of internet connection failure, student cannot finish this use case
Assumptions:	N/A

**Table 7. <Use Case> Get Event Information**

#### 2.2.2.4 Register Event



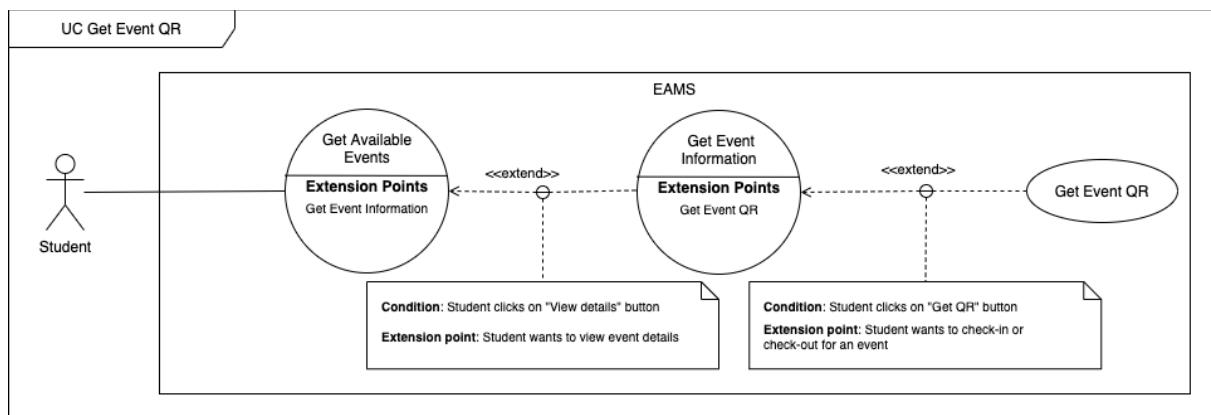
**Figure 13. <Use Case> Register Event**

ID and Name:	<b>UC-8. Register event</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	This feature allows student to register an event		
Trigger:	Student requests to register an event		
Preconditions:	PRE-1. User has logged in into EAMS as Student role PRE-2. Current date time is between start and end register date time of that event PRE-3. Event is published		
Post-conditions:	POST-1. EAMS saves student's register date time		
Normal Flow:	Step	Actor Action	System Response
	1	Student sends a request to register an event	
			EAMS return a message that student has registered successfully [Exception 1]

Alternative Flows:	N/A		
	Step	Actor Action	System Response
Exceptions:	No	Cause	System Response
	1	Student fail to register	EAMS returns a message that student is not eligible to register this event
Priority:	High		
Frequency of Use:	Approximately 40 users per event		
Business Rules:	<ul style="list-style-type: none"> <li>To be able to register an event, some requirements need to be fulfilled: <ul style="list-style-type: none"> <li>Student has not registered before</li> <li>If event restricts register, student must be in registrant whitelist</li> <li>Student must register between start register date time and end register date time</li> <li>Event must be published</li> </ul> </li> <li>There are certain events that the organizer wants to restrict who can and cannot register or check-in. Restriction whitelists are used for this purpose. Restriction whitelists are determined by staffs. There are 2 types of restriction whitelist: <ul style="list-style-type: none"> <li>Registration whitelist: Only students in this list can register to participate for registration restricted events.</li> <li>Check-in whitelist: Only students in this list can check-in for check-in restricted events</li> </ul> </li> <li>Registration stage of an event will be closed when end register date time is over</li> </ul>		
Other Information:	In case of internet connection failure, student cannot register event		
Assumptions:	N/A		

**Table 8. <Use Case> Register Event**

#### 2.2.2.5 Get Event QR



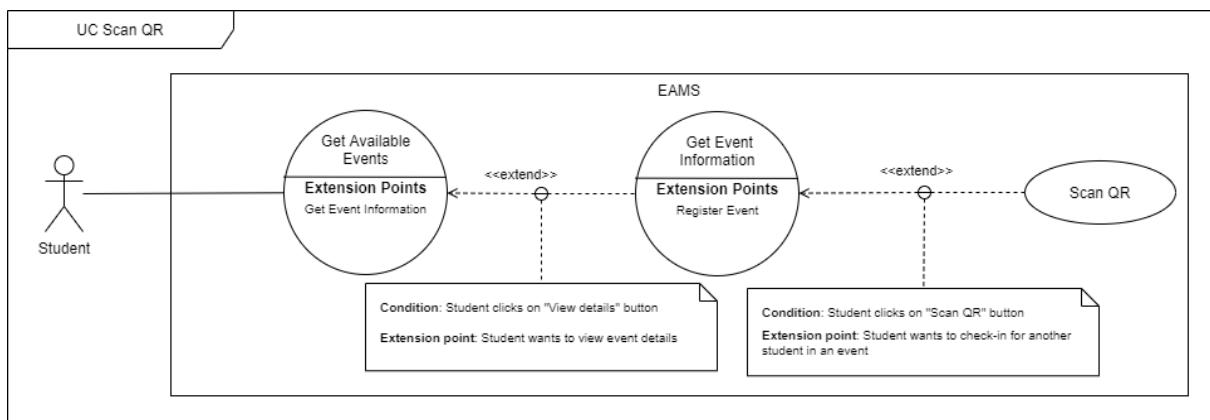
**Figure 14. <Use Case> Get Event QR**

ID and Name:	<b>UC-9. Get Event QR</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	This feature allows student to get a QR code that related to an event in order to check-in or check-out for it		
Trigger:	Students request to show their QR for checking-in or checking-out purpose		
Preconditions:	PRE-1. User has logged in into EAMS as Student role. PRE-2. Student has registered that event before.		
Post-conditions:	POST-1. EAMS returns QR code		
Normal Flow:	Step	Actor Action	System Response
	1	Student requests an event QR code	
			EAMS generates QR based on information of an event and returns to student
Alternative Flows:	N/A		
	Step	Actor Action	System Response
Exceptions:	No	Cause	System Response
Priority:	High		
Frequency of Use:	Approximately 1000 users will use this feature per event.		
Business Rules:	<ul style="list-style-type: none"> <li>To be able to check-in, some requirements need to be fulfilled: <ul style="list-style-type: none"> <li>Student has registered for this event</li> <li>Student has not checked-in before</li> <li>If event is restricted check-in, student must be allowed to check-in that event</li> <li>Student must check-in between start check-in date time and end check-in date time</li> </ul> </li> <li>To be able to check-out, some requirements need to be fulfilled: <ul style="list-style-type: none"> <li>Student must register before</li> <li>Student must check-in before</li> <li>Student must not check-out before</li> <li>Event must require check-out</li> <li>Student must check-out between start check-out date time and end check-out date time</li> <li>If event requires feedback before check-out, student must give feedback before.</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>Students show their QR code and other students with event scan QR permission (event's assistants) scan that QR in order to check-in or check-out for them. Time needed for a student, depends on internet connection, will be less than 3 seconds</li> <li>There are certain events that the organizer wants to restrict who can and cannot register or check-in. Restriction whitelists are used for this purpose. Restriction whitelists are determined by staffs. There are 2 types of restriction whitelist: <ul style="list-style-type: none"> <li>Registration whitelist: Only students in this list can register to participate for registration restricted events.</li> <li>Check-in whitelist: Only students in this list can check-in for check-in restricted events</li> </ul> </li> </ul>
Other Information:	Student can retrieve QR regardless of internet connection if student has registered that event on the same device before.
Assumptions:	Assume that 80% of registrants will use this feature.

**Table 9. <Use Case> Get Event QR**

#### 2.2.2.6 Scan QR



**Figure 15. <Use Case> Scan QR**

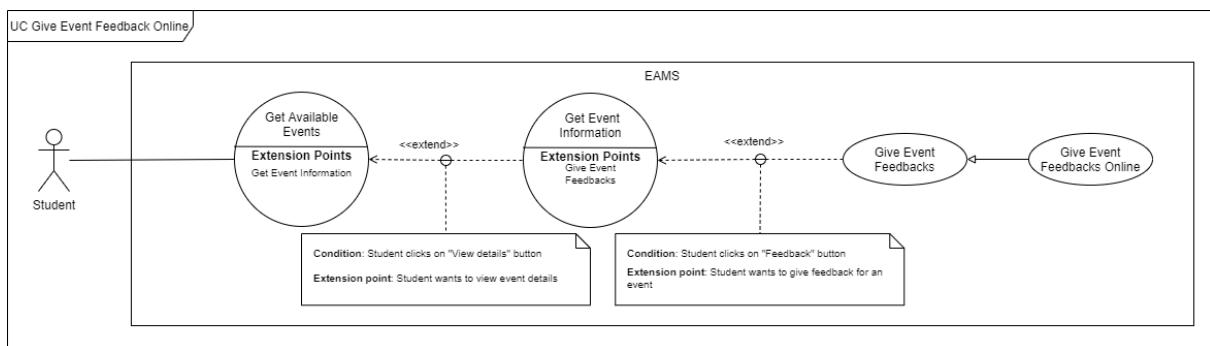
ID and Name:	<b>UC-10. Scan QR</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Event's assistants scan QR of other students to check-in or check-out for them		
Trigger:	Event's assistants request to scan QR		
Preconditions:	PRE-1. User has logged in into EAMS as Student role. PRE-2. User has internet connection PRE-3. Student has permission to scan QR for this event		
Post-conditions:	POST-1. EAMS stores check-in or check-out information of student		
Normal Flow:	Step	Actor Action	System Response

	1	Students who want to check-in or check-out show their QR code	
	2	Event's assistant scans that QR code and sends QR information to EAMS	
		EAMS verifies QR information then returns to assistant check-in or check-out successfully message [Exception 1] [Exception 2]	
Alternative Flows:	N/A		
	Step	Actor Action	System Response
Exceptions:	No	Cause	System Response
	1	Assistant scans check-in QR code failed	EAMS returns a failure message to assistant that check-in has failed
	2	Assistant scans check-out QR code failed	EAMS returns a failure message to assistant that check-out has failed
Priority:	High		
Frequency of Use:	Approximately 60 to 80 usages per event		
Business Rules:	<ul style="list-style-type: none"> <li>Event assistants are user with permission to scan QR to check-in for an event. Every event needs at least one user as an assistant to check-in or check-out for students. Assistants scan QR to check-in or check-out for students. Staff can assign Scan-QR permission to any users for them to become assistants. This is recommended to be done before event's start check-in date time</li> <li>Participants show their QR code and students with event scan QR permission (event's assistants) scan that QR in order to check-in or check-out for them. Time needed for a student, depends on internet connection, will be less than 3 seconds</li> <li>To be able to check-in, some requirements need to be fulfilled: <ul style="list-style-type: none"> <li>Student has registered for this event</li> <li>Student has not checked-in before</li> <li>If event is restricted check-in, student must be allowed to check-in that event</li> <li>Student must check-in between start check-in date time and end check-in date time</li> </ul> </li> <li>To be able to check-out, some requirements need to be fulfilled: <ul style="list-style-type: none"> <li>Student must register before</li> <li>Student must check-in before</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ Student must not check-out before</li> <li>○ Event must require check-out</li> <li>○ Student must check-out between start check-out date time and end check-out date time</li> <li>○ If event requires feedback before check-out, student must give feedback before.</li> <li>● There are certain events that the organizer wants to restrict who can and cannot register or check-in. Restriction whitelists are used for this purpose. Restriction whitelists are determined by staffs. There are 2 types of restriction whitelist: <ul style="list-style-type: none"> <li>○ Registration whitelist: Only students in this list can register to participate for registration restricted events.</li> <li>○ Check-in whitelist: Only students in this list can check-in for check-in restricted events</li> </ul> </li> <li>● After being checked-in, or checked-out, student's checked-in or checked-out date time will be saved in database</li> </ul>
Other Information:	In case of internet connection failure, assistant cannot scan QR
Assumptions:	Assistant's phone must have camera that is able to scan QR.

**Table 10. <Use Case> Scan QR**

#### 2.2.2.7 Give Event Feedbacks Online



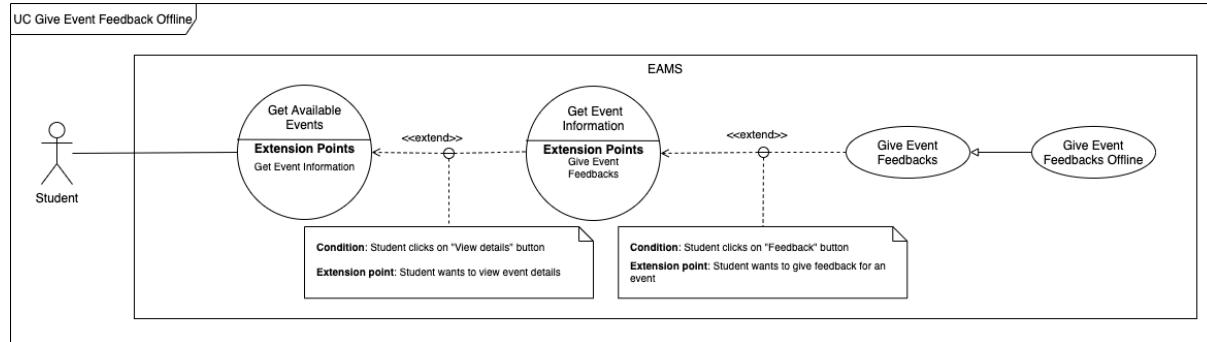
**Figure 16. <Use Case> Give Event Feedbacks Online**

ID and Name:	<b>UC-11. Give Event Feedbacks Online</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Allow student to give feedbacks for an event when they are online so that the event organizers can improve on future events		
Trigger:	Student requests to feedback for an event		
Preconditions:	PRE-1. User has logged in the system as Student PRE-2. Student has internet connection PRE-3. Student has checked-in for the event they want to feedback PRE-4. The current date time is between event's start feedback date time and 12 hours after event's end date time		

Post-conditions:	POST-1. EAMS stores student's feedback answers and the date time that they feedback		
Normal Flow:	Step	Actor Action	System Response
	1	Student sends request to get feedback questions	
			EAMS returns feedback questions of the event
	2	Student enters the feedback answers	
	3	Student sends request to feedback for the event [Exception 1]	
			EAMS returns a message that student has given feedbacks successfully
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Student does not answer all the questions	EAMS informs student that they have to fill in the missing answers
Priority:	Medium		
Frequency of Use:	Average of 30 usages per event. Peak usage load for this use case happens when multiple events with high number of participants are about to end		
Business Rules:	<ul style="list-style-type: none"> <li>The university staffs have to create feedback questions when they create the event. They also have to decide when the students can start giving feedback by specifying start feedback date time.</li> <li>Students can only give feedbacks for events that they have checked in. Feedback is optional unless the organizer wants students to feedback before checking out</li> <li>When giving feedback, users have to fill answer for all feedback questions. There are 2 types of feedback question: <ul style="list-style-type: none"> <li>Text: user can freely type, length between 1 - 255</li> <li>Rating: user have to input an integer number from 1 to 5</li> </ul> </li> <li>An event must have at least one and no more than 10 feedback questions</li> <li>After students' feedback, the system stores student's feedback answers and the date time when they give the feedback</li> </ul>		
Other Information:	Student cannot send more than one feedback request		
Assumptions:	Most events will have more than 70% participants that finish giving feedbacks		

**Table 11. <Use Case> Give Event Feedbacks Online**

### 2.2.2.8 Give Event Feedbacks Offline



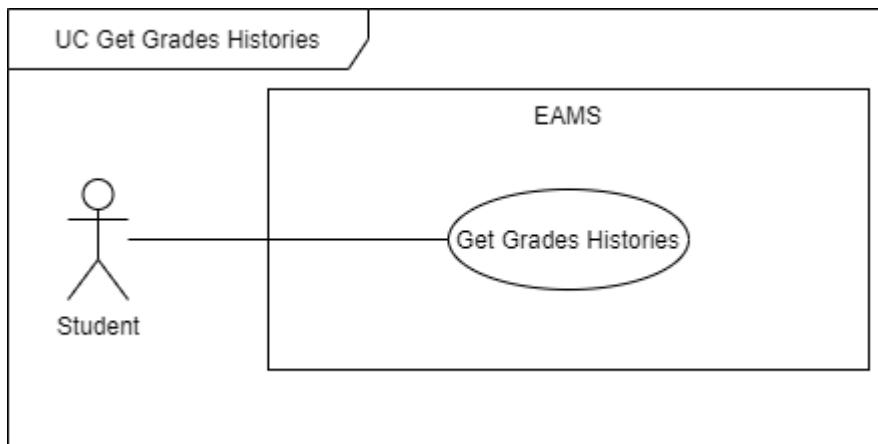
**Figure 17. <Use Case> Give Event Feedbacks Offline**

ID and Name:	UC-12. Give Event Feedbacks Offline		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	When internet connection is unavailable, student still be able to give their feedback. Their feedback answers will be sent to EAMS when event's assistants scan check-out QR.		
Trigger:	Student requests to give their feedback answers to an event.		
Preconditions:	PRE-1. User has logged in into EAMS as Student role. PRE-2. Event requires students to check-out PRE-3: Student uses the device that they used to register.		
Post-conditions:	POST-1. Feedback answers are stored on student's device which will later be sent to EAMS		
Normal Flow:	Step	Actor Action	System Response
	1	Student requests to give feedback	
			EAMS loads feedback questions from student's device
	2	Student enters their feedback answers	
	3	Student submits feedback answers	
Alternative Flows:	Student's device stores feedback answers		
	Step	Actor Action	System Response
Exceptions:	No	Cause	System Response

	1	Student does not answer all answers	EAMS returns a failure message to student about missing answers
Priority:	Medium		
Frequency of Use:	Average of 30 usages per event. Peak usage load for this use case happens when multiple events with high number of participants are about to end		
Business Rules:	<ul style="list-style-type: none"> <li>EAMS provides some features that are more convenient for both staffs and students: <ul style="list-style-type: none"> <li>Allow staff to define start and end feedback date time which means feedback form will close after a defined time</li> <li>Staff can export feedback report about that event</li> <li>Students can conveniently give feedback on their smartphone</li> </ul> </li> <li>To be able to feedback offline, some requirements need to be fulfilled before: <ul style="list-style-type: none"> <li>Student must register before</li> <li>Student must check-in before</li> <li>Event must require check-out</li> </ul> </li> <li>Every event needs at least one user as an assistant to check-in or check-out for students. Assistants scan QR to check-in or check-out for students. Staff can assign Scan-QR permission for any users for them to become assistants. This should be done before event's start check-in date time</li> </ul>		
Other Information:	N/A		
Assumptions:	Most events will have more than 70% participants that finish giving feedbacks. Internet connection is disable.		

**Table 12. <Use Case> Give Event Feedbacks Offline**

#### 2.2.2.9 Get Grade Histories



**Figure 18. <Use Case> Get Grades Histories**

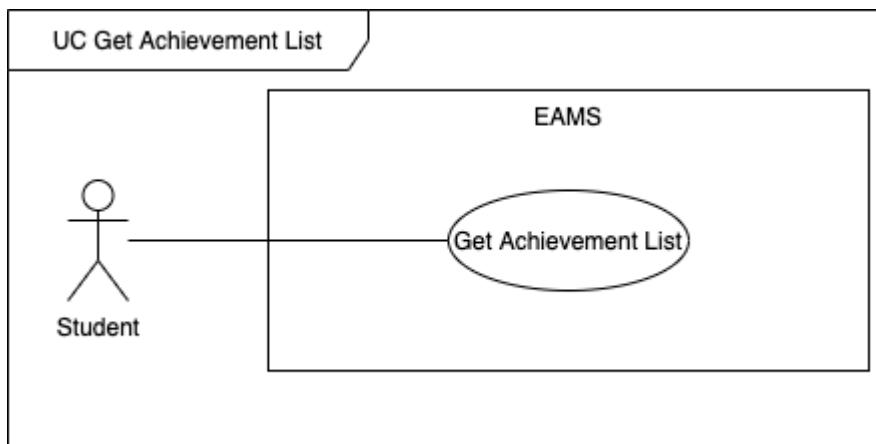
ID and Name:	<b>UC-13. Get Grade Histories</b>
--------------	-----------------------------------

Created By:	BinhPD		Date Created:	02/12/20		
Primary Actor:	Student		Secondary Actors:	N/A		
Description:	Allow students to keep track of their performances in extracurricular activities					
Trigger:	Student requests to view grade histories					
Preconditions:	PRE-1. User has logged in the system as Student					
Post-conditions:	POST-1. EAMS returns information about student's grade histories					
Normal Flow:	Step	Actor Action	System Response			
	1	Student sends request to get grade histories including semester info				
	2		EAMS shows students their grading histories along with their total grades by criteria and the reasons why they are qualified or disqualified for grades for their participations			
Alternative Flows:	N/A					
Exceptions:	N/A					
Priority:	Medium					
Frequency of Use:	Average of one usage per user per week.					
Business Rules:	<ul style="list-style-type: none"> <li>After an event is graded by a staff, its participants and contributors are able to view their grade histories of that event.</li> <li>For each event, student can both participate and contribute. This is shown as 2 distinct histories in the system.</li> <li>For participation histories: <ul style="list-style-type: none"> <li>Staff decides the grades that will later be stored as students' histories by picking students from the "Qualified participants" list.</li> <li>In order to be in "Qualified Participants" list of an event, student has to register and check-in. If the event requires checking out, that also student has to check-out.</li> <li>When grading the event, staff decides which students are not qualified as a participant of that event by removing them from the "Qualified Participants" list. When doing that, staff has to give a reason why that student is disqualified.</li> </ul> </li> <li>For contribution histories: <ul style="list-style-type: none"> <li>As long as a student contributes to at least one sub-task of a task of an event. That student is considered as a contributor to</li> </ul> </li> </ul>					

	<p>that event, no matter how much points that student receives for doing that task.</p> <ul style="list-style-type: none"> <li>After finishing an event, meaning staffs has finalized the participation histories and contribution histories of students, this information is saved to the system and can no longer be changed afterwards.</li> <li>From then on, student can request to get their histories and see their participation or contribution histories of the graded event that they are in.</li> </ul>
Other Information:	In case of internet connection failure, student may not be able to see the latest histories
Assumptions:	Most students will participate events and qualify for grades histories

**Table 13. <Use Case> Get Grade Histories**

#### 2.2.2.10 Get Achievements List



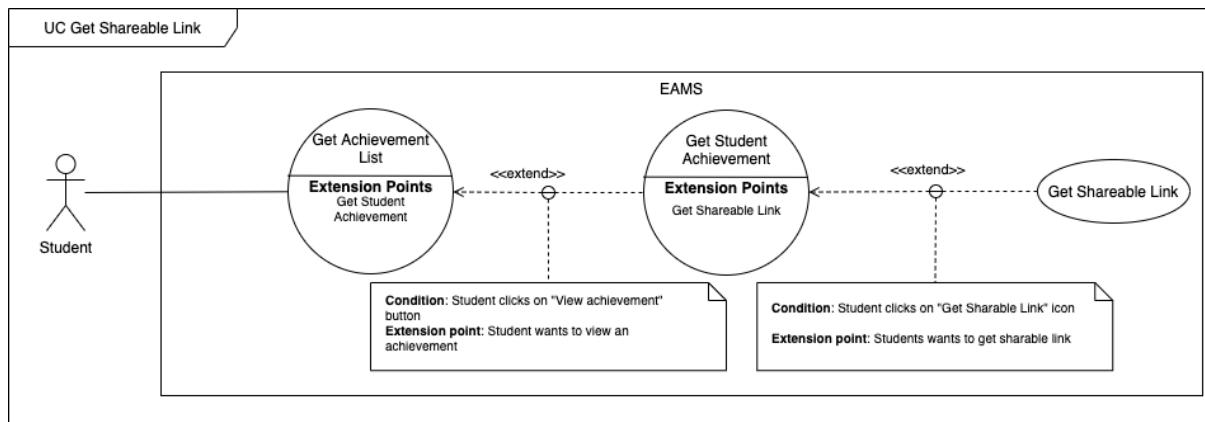
**Figure 19. <Use Case> Get Achievements List**

ID and Name:	<b>UC-14. Get Achievements List</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Student requests a list of their achievements which are filtered by semester.		
Trigger:	Student requests their achievements.		
Preconditions:	PRE-1. User has logged in into EAMS as Student role.		
Post-conditions:	N/A		
Normal Flow:	Step	Actor Action	System Response
	1	Students request a list of their achievements by semester	

			EAMS returns their achievements filtered by semester
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
Priority:	Medium		
Frequency of Use:	Approximately 20 users a week.		
Business Rules:	<ul style="list-style-type: none"> <li>In EAMS, after being graded for an event, participations list and contributions list will be handed over to award stage where staff will decide which one will receive achievement.</li> <li>To enter this stage, student has checked-out if event requires check-out or checked-in if event doesn't. And event's grade of that student is larger than 0, whether that student is graded or not depends on staff in grading stage.</li> <li>Achievements are granted for students for their performances in extracurricular activities. There are 2 types of achievements: <ul style="list-style-type: none"> <li>Achievements for participations and contributions: Students can earn this achievement by participating in or contributing to an event. Achievement of this type can only be granted for participating in or contributing to finished and graded events.</li> <li>Achievements for special reasons: Universities can grant students achievements for exceptional performances in extracurricular activities.</li> </ul> </li> <li>Achievements for participations can only be granted for qualified participants. These are the conditions for a participant to be qualified: <ul style="list-style-type: none"> <li>A student must register, check-in to be considered as qualified</li> <li>If the event requires check-out, student must check-out to be considered qualified</li> </ul> </li> <li>Staff can choose which students to grant achievements <ul style="list-style-type: none"> <li>For achievements for participation and contribution: Staff can only grant achievements for qualified students. However, they can choose to disqualify students if they provide a reason for it.</li> <li>For achievements for special reasons: Staff can grant achievements of this type to any student if they provide a reason for it.</li> </ul> </li> </ul>		
Other Information:	In case of internet connection failure, student may not be able to get the latest achievements		
Assumptions:	N/A		

**Table 14. <Use Case> Get Achievement List**

### 2.2.2.11 Get Shareable Link



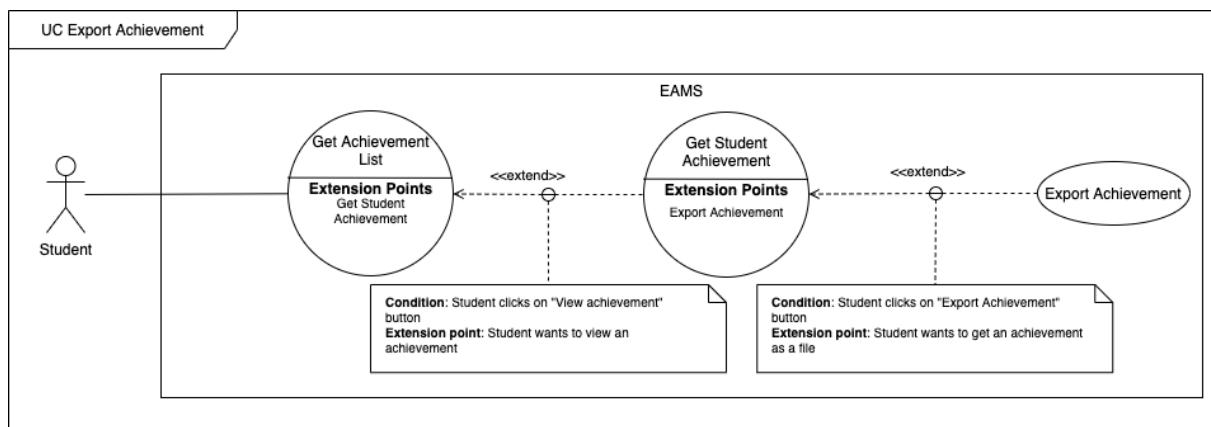
**Figure 20. <Use Case> Get Shareable Link**

ID and Name:	<b>UC-15. Get Shareable Link</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Student get a shareable link of an achievement they receive in the university in order to show it off to others such as employers		
Trigger:	Student requests to get a shareable link of an achievement		
Preconditions:	N/A		
Post-conditions:	POST-1. User receives a link.		
Normal Flow:	Step	Actor Action	System Response
	1	User requests to get a shareable link	
			EAMS returns shareable link of that achievement
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Achievement does not exist	EAMS returns that achievement does not exist
Priority:	Medium		
Frequency of Use:	This feature is used most after an event has finished		
Business Rules:	<ul style="list-style-type: none"> <li>After being graded for an event, participations list and contributions list will be handed over to award stage where staff will decide which one will receive achievement.</li> <li>Student can share their achievements to anyone</li> <li>Achievements are granted for students for their performances in extracurricular activities. There are 2 types of achievements:</li> </ul>		

	<ul style="list-style-type: none"> <li>○ Achievements for participations and contributions: Students can earn this achievement by participating in or contributing to an event. Achievement of this type can only be granted for participating in or contributing to finished and graded events.</li> <li>○ Achievements for special reasons: Universities can grant students achievements for exceptional performances in extracurricular activities.</li> </ul>
Other Information:	<p>In case of internet connection failure, student cannot get shareable link.</p> <p>All achievements in the system are public regardless of whether or not it has been shared by a student before</p>
Assumptions:	N/A

**Table 15. <Use Case> Get Shareable Link**

#### 2.2.2.12 Export Achievement



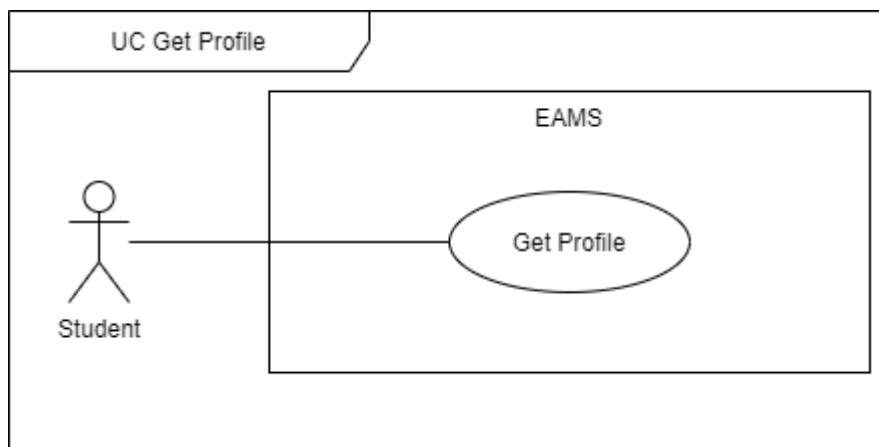
**Figure 21. <Use Case> Export Achievement**

ID and Name:	<b>UC-16. Export Achievement</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Student requests to get achievement detail as PDF file.		
Trigger:	Student requests to download an achievement as a PDF file.		
Preconditions:	N/A		
Post-conditions:	POST-1. User receives an achievement as a PDF file.		
Normal Flow:	Step	Actor Action	System Response
	1	User sends request to download PDF file of an achievement	
			EAMS returns PDF file. [Exception 1]

Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	E1 Achievement does not exist	EAMS returns that achievement does not exist
Priority:	Medium		
Frequency of Use:	This feature is used most after an event has finished		
Business Rules:	<ul style="list-style-type: none"> <li>In EAMS, after being graded for an event, participations list and contributions list will be handed over to award stage where staff will decide which one will receive achievement.</li> <li>To enter this stage, student has checked-out if event requires check-out or checked-in if event doesn't. And event's grade of that student is larger than 0, whether that student is graded or not depends on staff in grading stage.</li> <li>Achievement are public and can be shared through link or pdf file.</li> </ul>		
Other Information:	<ul style="list-style-type: none"> <li>Every download request needs about 7-9s to accomplish.</li> <li>In case of internet connection failure, student cannot export achievement</li> </ul>		
Assumptions:	User device is able to download and read pdf file and it has stable internet connection.		

**Table 16. <Use Case> Export Achievement**

#### 2.2.2.13 Get Profile



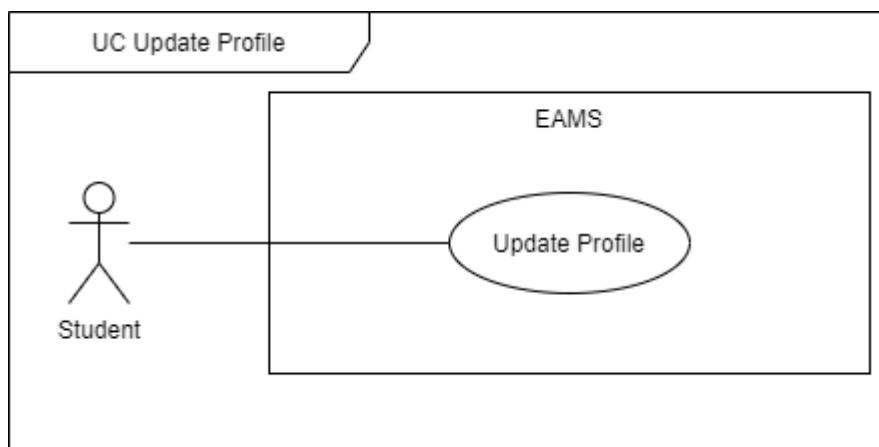
**Figure 22. <Use Case> Get Profile**

ID and Name:	<b>UC-17. Get Profile</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Student views their profile in current semester: basic info, grade, achievements...		
Trigger:	Student sends a request to get their profile		

Preconditions:	PRE-1. User has logged in as Student role		
Post-conditions:	POST-1. Student profile is returned to them		
Normal Flow:	Step	Actor Action	System Response
	1	Student requests to get profile	
			EAMS returns student profile
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Medium		
Frequency of Use:	Approximately 2000 users, average of one usage per user per week		
Business Rules:	<ul style="list-style-type: none"> <li>Student can get their current semester grade, not all semester grade</li> <li>Student's profile is imported as the start of a semester, student cannot manually change name, email...</li> </ul>		
Other Information:	In case of internet connection failure, student cannot finish this use case		
Assumptions:	N/A		

**Table 17. <Use Case> Get Profile**

#### 2.2.2.14 Update Profile



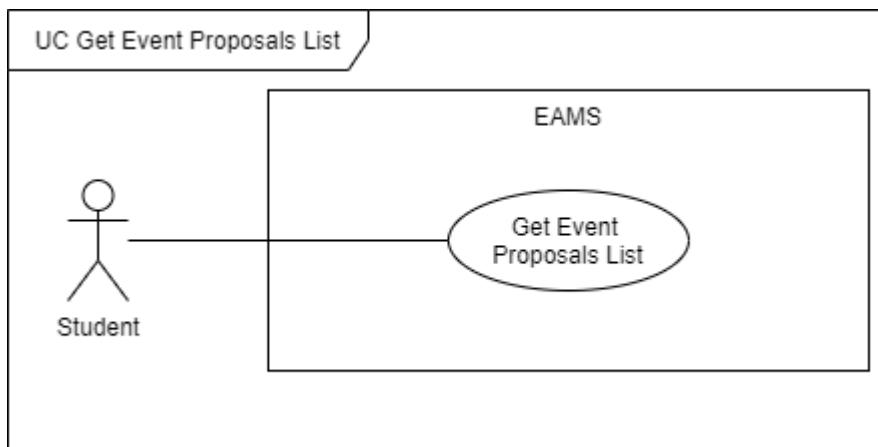
**Figure 23. <Use Case> Update Profile**

ID and Name:	<b>UC-18. Update Profile</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Allow students to personalize their profile by choosing an image to represent them		
Trigger:	Student requests to update their profile		
Preconditions:	PRE-1. User has logged in the system as Student		

Post-conditions:	POST-1. EAMS stores student's profile		
Normal Flow:	Step	Actor Action	System Response
	1	Student sends request to update profile	
			EAMS shows dialog for user to input
	2	Student update fields	
			EAMS stores student's profile
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Average of one usage per user		
Business Rules:	<ul style="list-style-type: none"> <li>The university only allow student to update their avatar in their profile</li> </ul>		
Other Information:	<ul style="list-style-type: none"> <li>Student cannot view previous avatars</li> <li>In case of internet connection failure, students cannot update their profile</li> </ul>		
Assumptions:	The university does not allow students to update their names and emails		

**Table 18. <Use Case> Update Profile**

#### 2.2.2.15 Get Event Proposals List



**Figure 24. <Use Case> Get Event Proposals List**

ID and Name:	<b>UC-19. Get Event Proposals List</b>		
Created By:	BinhPD	Date Created:	02/12/20

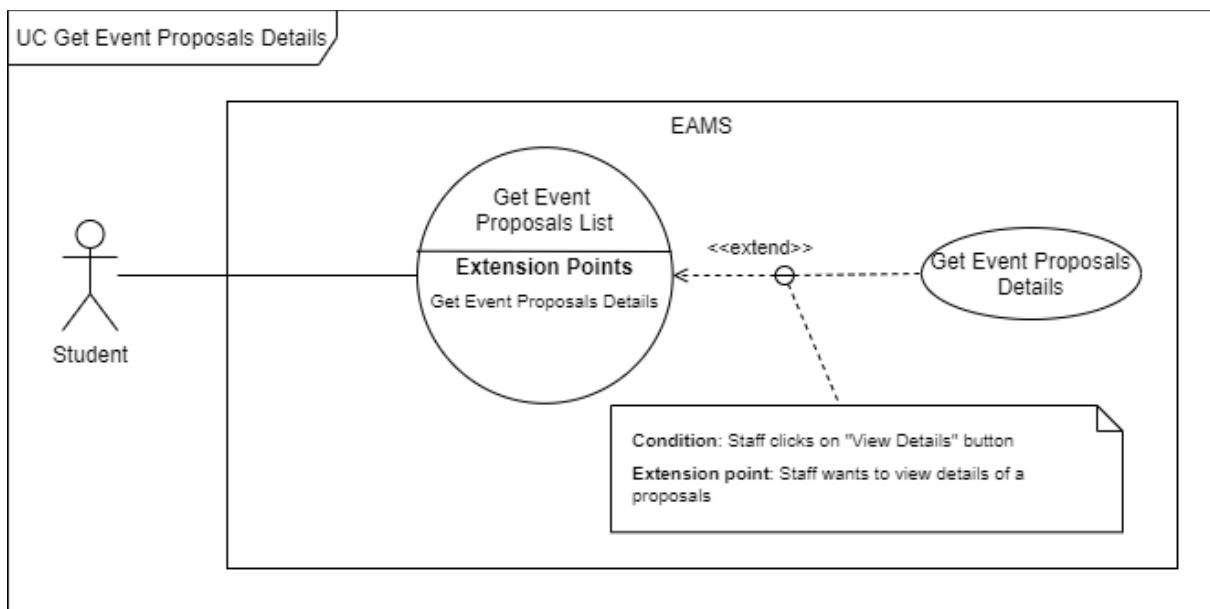
Primary Actor:	Unauthorized User	Secondary Actors:	N/A
Description:	Provide students with the ability to view the events that they propose to staff		
Trigger:	Student sends request to see the events that they have proposed		
Preconditions:	PRE-1. User has logged in the system as Student		
Post-conditions:	POST-1. EAMS shows users their proposed events		
Normal Flow:	Step	Actor Action	System Response
	1	Student sends request to get the events that they have created	
			EAMS returns the events that student has created
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Medium		
Frequency of Use:	Most students rarely uses this feature		
Business Rules:	<ul style="list-style-type: none"> <li>Students have the ability to plan, manage, submit proposals to organize events. Students can only view the events that they created, not other students' events.</li> <li>Students can only manage basic info, criteria, total contribution grade, description, posts, and tasks of each of the events that they propose</li> <li>Students can only make changes to basic info, criteria, total contribution grade, description <u>before</u> submitting the event to staff. Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Student cannot make changes to their events during the time they wait for them to be approved</li> <li>An event requires the following fields: <ul style="list-style-type: none"> <li>Title: between 1 – 63 characters</li> <li>Location: between 1 – 63 characters</li> <li>Semester: current semester or the semester that will begin in the next 30 days</li> <li>Start and end date time: End date time comes after start date time</li> <li>Start register date time: Less than end date time</li> <li>End register date time: between start register date time - end date time</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ Start check-in date time: between start register date time - end date time</li> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that's less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● When an event proposal gets created, EAMS stores it as an event with a “Draft” status. Student can then submit event proposal to staff, which turns it to “Pending” status. Staff can then reject the proposal, changing it to “Rejected”, give it some feedback for them to submit it again until it gets approved. If a staff decides to approve the event, its status changes to “Unpublished”, they will take over and manage it as if they created that event.</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are</li> </ul> </li> </ul>
--	---

	<p>recommended to assign assistants to help scan QR to check-in or check-out for the event</p> <ul style="list-style-type: none"> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> </ul>
Other Information:	<ul style="list-style-type: none"> <li>● Students are recommended to use a computer with a large screen to use this feature</li> <li>● In case of internet connection failure, student may not be able to see the latest event details</li> </ul>
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools' event organization departments use this feature frequently

**Table 19. <Use Case> Get Event Proposals List**

#### 2.2.2.16 Get Event Proposals Details



**Figure 25. <Use Case> Get Event Proposals Details**

ID and Name:	<b>UC-20. Get Event Proposals Details</b>		
Created By:	BinhPD	Date Created:	02/12/20

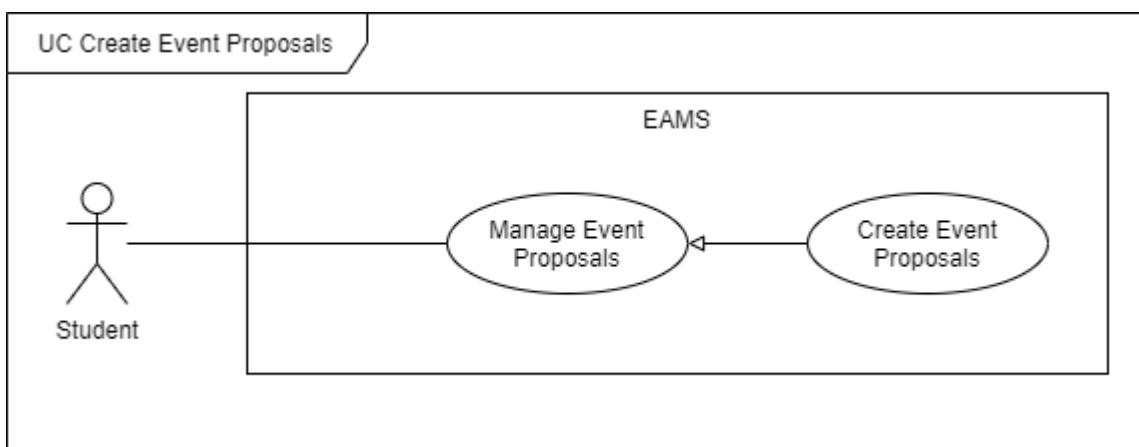
Primary Actor:	Unauthorized User	Secondary Actors:	N/A
Description:	Provide students with the ability to view the details of an event that they created to propose to staff		
Trigger:	Student sends request to see the event that they created		
Preconditions:	PRE-1. User has logged in the system as Student PRE-2. User is the creator of the event being requested		
Post-conditions:	POST-1. EAMS returns student their event details		
Normal Flow:	Step	Actor Action	System Response
	1	Student sends request to get the details of the event that they have created	
			EAMS returns the details of the event being requested
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Medium		
Frequency of Use:	Most students rarely uses this feature		
Business Rules:	<ul style="list-style-type: none"> <li>Students have the ability to plan, manage, submit proposals to organize events. Students can only view the events that they created, not other students' events.</li> <li>Students can only manage basic info, criteria, total contribution grade, description, posts, and tasks of each of the events that they propose</li> <li>Students can only make changes to basic info, criteria, total contribution grade, description <u>before</u> submitting the event to staff. Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Student cannot make changes to their events during the time they wait for them to be approved</li> <li>An event requires the following fields: <ul style="list-style-type: none"> <li>Title: between 1 – 63 characters</li> <li>Location: between 1 – 63 characters</li> <li>Semester: current semester or the semester that will begin in the next 30 days</li> <li>Start and end date time: End date time comes after start date time</li> <li>Start register date time: Less than end date time</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ End register date time: between start register date time - end date time</li> <li>○ Start check-in date time: between start register date time - end date time</li> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that's less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● When an event proposal gets created, EAMS stores it as an event with a “Draft” status. Student can then submit event proposal to staff, which turns it to “Pending” status. Staff can then reject the proposal, changing it to “Rejected”, give it some feedback for them to submit it again until it gets approved. If a staff decides to approve the event, its status changes to “Unpublished”, they will take over and manage it as if they created that event.</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> </ul> </li> </ul>
--	--

	<ul style="list-style-type: none"> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> </ul>
Other Information:	<ul style="list-style-type: none"> <li>● Students are recommended to use a computer with a large screen to use this feature</li> <li>● In case of internet connection failure, student may not be able to see the latest event details</li> </ul>
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools' event organization departments use this feature frequently

**Table 20. <Use Case> Get Event Proposals Details**

#### 2.2.2.17 Create Event Proposals



**Figure 26. <Use Case> Create Event Proposals**

ID and Name:	<b>UC-21. Create Event Proposals</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A

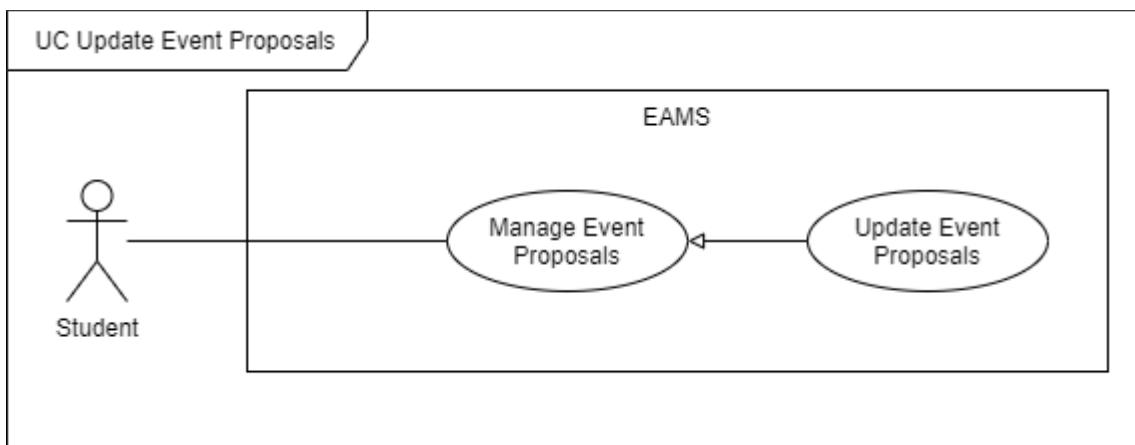
Description:	Provide students with the ability to propose an event to university staffs		
Trigger:	Student sends request to create an event		
Preconditions:	PRE-1. User has logged in the system as Student		
Post-conditions:	POST-1. EAMS stores the event information		
Normal Flow:	Step	Actor Action	System Response
	1	Students fills in the event information and sends requests to create the event [Exception 1]	
			EAMS validates the event information and stores them
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Student fails to fill in all the required fields	EAMS returns to student to fill in the missing fields
Priority:	Medium		
Frequency of Use:	Most students rarely use this feature		
Business Rules:	<ul style="list-style-type: none"> <li>When an event proposal gets created, EAMS stores it as an event with a “Draft” status. Student can then submit event proposal to staff, which turns it to “Pending” status. Staff can then reject the proposal, changing it to “Rejected”, give it some feedback for them to submit it again until it gets approved. If a staff decides to approve the event, its status changes to “Unpublished”, they will take over and manage it as if they created that event.</li> <li>Students have the ability to plan, manage, submit proposals to organize events. Students can only view the events that they created, not other students’ events.</li> <li>Students can only manage basic info, criteria, total contribution grade, description, posts, and tasks of each of the events that they propose</li> <li>Students can only make changes to basic info, criteria, total contribution grade, description <u>before</u> submitting the event to staff. Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Student cannot make changes to their events during the time they wait for them to be approved</li> <li>An event requires the following fields: <ul style="list-style-type: none"> <li>○ Title: between 1 – 63 characters</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ Location: between 1 – 63 characters</li> <li>○ Semester: current semester or the semester that will begin in the next 30 days</li> <li>○ Start and end date time: End date time comes after start date time</li> <li>○ Start register date time: Less than end date time</li> <li>○ End register date time: between start register date time - end date time</li> <li>○ Start check-in date time: between start register date time - end date time</li> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that's less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the</li> </ul> </li> </ul>
--	--

	<p>event status becomes “Published”. This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</p> <ul style="list-style-type: none"> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> </ul>
Other Information:	<ul style="list-style-type: none"> <li>● Students are recommended to use a computer with a large screen to use this feature</li> <li>● In case of internet connection failure, student cannot create event</li> </ul>
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools’ event organization departments use this feature frequently

**Table 21. <Use Case> Create Event Proposals**

#### 2.2.2.18 Update Event Proposals



**Figure 27. <Use Case> Update Event Proposals**

ID and Name:	<b>UC-22. Update Event Proposals</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Provide students with the ability to update an event that they want to propose for university staffs to organize		
Trigger:	Student sends request to update an event		

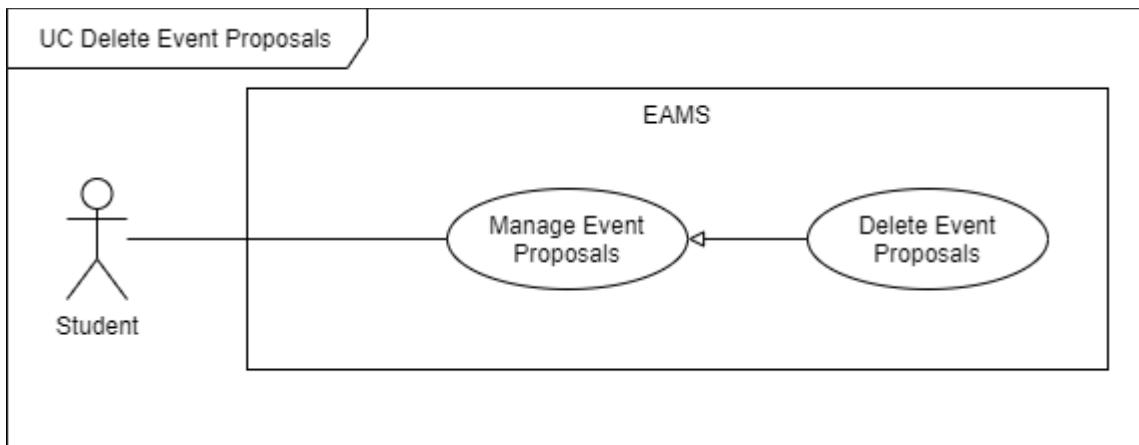
Preconditions:	PRE-1. User has logged in the system as Student PRE-2. User is the creator of the event		
Post-conditions:	POST-1. EAMS stores the new event information		
Normal Flow:	Step	Actor Action	System Response
	1	Students fills in the event information and sends requests to update the event [Exception 1]	
			EAMS validates the event information and stores the event information
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Student fails to fill in all the required fields	EAMS returns to student to fill in the missing fields
Priority:	Medium		
Frequency of Use:	Most students rarely use this feature		
Business Rules:	<ul style="list-style-type: none"> <li>Students have the ability to plan, manage, submit proposals to organize events. Students can only view the events that they created, not other students' events.</li> <li>Students can only manage basic info, criteria, total contribution grade, description, posts, and tasks of each of the events that they propose</li> <li>Students can only make changes to basic info, criteria, total contribution grade, description <u>before</u> submitting the event to staff. Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Student cannot make changes to their events during the time they wait for them to be approved</li> <li>An event requires the following fields: <ul style="list-style-type: none"> <li>Title: between 1 – 63 characters</li> <li>Location: between 1 – 63 characters</li> <li>Semester: current semester or the semester that will begin in the next 30 days</li> <li>Start and end date time: End date time comes after start date time</li> <li>Start register date time: Less than end date time</li> <li>End register date time: between start register date time - end date time</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ Start check-in date time: between start register date time - end date time</li> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that's less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● When an event proposal gets created, EAMS stores it as an event with a “Draft” status. Student can then submit event proposal to staff, which turns it to “Pending” status. Staff can then reject the proposal, changing it to “Rejected”, give it some feedback for them to submit it again until it gets approved. If a staff decides to approve the event, its status changes to “Unpublished”, they will take over and manage it as if they created that event.</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are</li> </ul> </li> </ul>
--	---

	<p>recommended to assign assistants to help scan QR to check-in or check-out for the event</p> <ul style="list-style-type: none"> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> </ul>
Other Information:	<ul style="list-style-type: none"> <li>● Students are recommended to use a computer with a large screen to use this feature</li> <li>● In case of internet connection failure, student cannot update event</li> </ul>
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools' event organization departments use this feature frequently

**Table 22. <Use Case> Update Event Proposals**

#### 2.2.2.19 Delete Event Proposals



**Figure 28. <Use Case> Delete Event Proposals**

ID and Name:	<b>UC-23. Delete Event Proposals</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Provide students with the ability to delete an event proposal		
Trigger:	Student sends request to delete an event		
Preconditions:	PRE-1. User has logged in the system as Student		

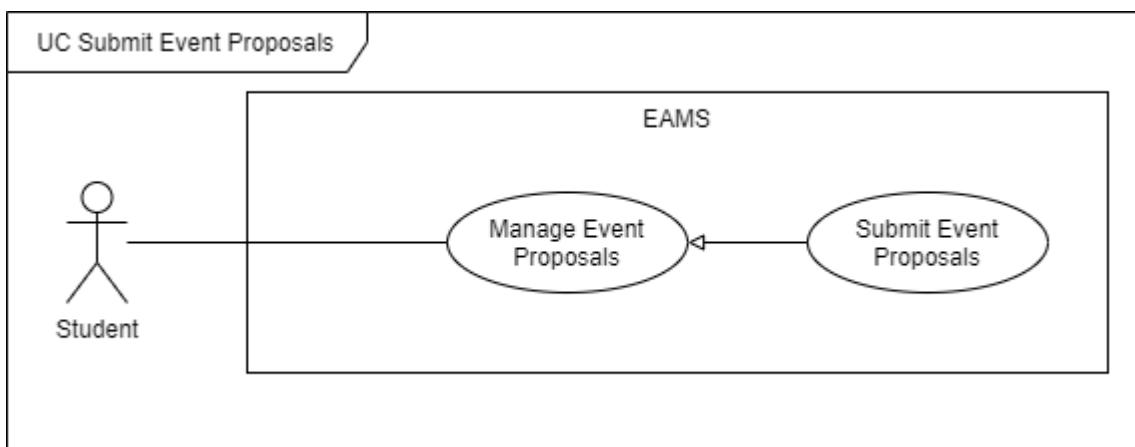
	PRE-2. User is the creator of the event		
Post-conditions:	POST-1. EAMS deletes the event		
Normal Flow:	Step	Actor Action	System Response
	1	Student sends request to delete event [Exception 1]	
			EAMS Returns to user that event has been deleted
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Event not a draft event	EAMS returns to student that they can only delete draft events
Priority:	Medium		
Frequency of Use:	Most students rarely use this feature		
Business Rules:	<ul style="list-style-type: none"> <li>Students have the ability to plan, manage, submit proposals to organize events. Students can only view the events that they created, not other students' events.</li> <li>Students can only manage basic info, criteria, total contribution grade, description, posts, and tasks of each of the events that they propose</li> <li>Students can only make changes to basic info, criteria, total contribution grade, description <u>before</u> submitting the event to staff. Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Student cannot make changes to their events during the time they wait for them to be approved</li> <li>An event requires the following fields: <ul style="list-style-type: none"> <li>Title: between 1 – 63 characters</li> <li>Location: between 1 – 63 characters</li> <li>Semester: current semester or the semester that will begin in the next 30 days</li> <li>Start and end date time: End date time comes after start date time</li> <li>Start register date time: Less than end date time</li> <li>End register date time: between start register date time - end date time</li> <li>Start check-in date time: between start register date time - end date time</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that's less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● When an event proposal gets created, EAMS stores it as an event with a “Draft” status. Student can then submit event proposal to staff, which turns it to “Pending” status. Staff can then reject the proposal, changing it to “Rejected”, give it some feedback for them to submit it again until it gets approved. If a staff decides to approve the event, its status changes to “Unpublished”, they will take over and manage it as if they created that event.</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> </ul> </li> </ul>
--	---

	<ul style="list-style-type: none"> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> <li>● Students can only delete events that they created with a “Draft” status. After an event gets deleted, EAMS no longer stores it.</li> </ul>
Other Information:	In case of internet connection failure, student cannot delete event
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools' event organization departments use this feature frequently

**Table 23. <Use Case> Delete Event Proposals**

#### 2.2.2.20 Submit Event Proposals



**Figure 29. <Use Case> Submit Event Proposals**

ID and Name:	<b>UC-24. Submit Event Proposals</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Provide students with the ability to submit an event proposal to university staffs		
Trigger:	Student sends request to retract an event		
Preconditions:	PRE-1. User has logged in the system as Student PRE-2. User is the creator of the event		

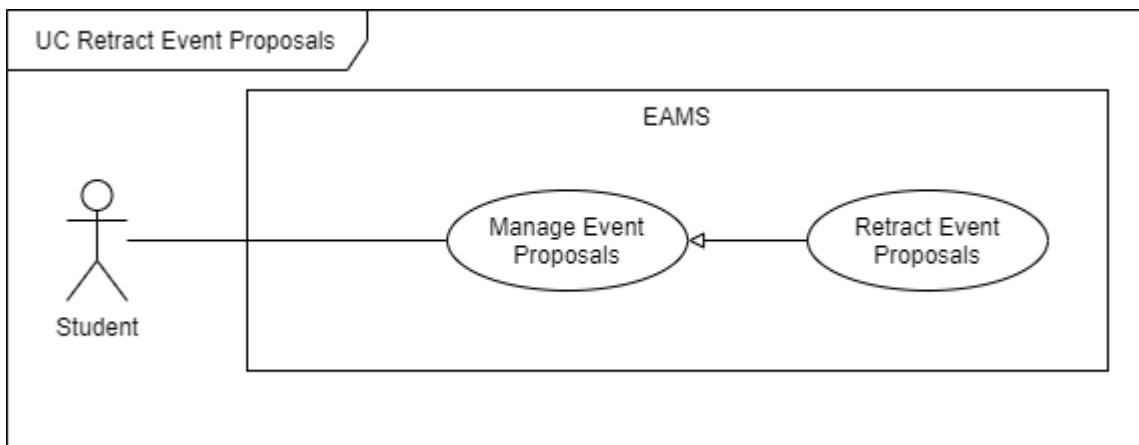
	PRE-3. The event status is “Rejected” or “Draft”		
Post-conditions:	POST-1. EAMS changes event status to “Pending”		
Normal Flow:	Step	Actor Action	System Response
	1	Student sends request to submit event [Exception 1]	
Alternative Flows:	N/A		
	No	Cause	System Response
Exceptions:	1	Event can no longer be submitted	EAMS returns to student that they can only submit draft and rejected events
Priority:	Medium		
Frequency of Use:	Most students rarely use this feature		
Business Rules:	<ul style="list-style-type: none"> <li>Students have the ability to plan, manage, submit proposals to organize events. Students can only view the events that they created, not other students’ events.</li> <li>Students can only manage basic info, criteria, total contribution grade, description, posts, and tasks of each of the events that they propose</li> <li>Students can only make changes to basic info, criteria, total contribution grade, description <u>before</u> submitting the event to staff. Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Student cannot make changes to their events during the time they wait for them to be approved</li> <li>An event requires the following fields: <ul style="list-style-type: none"> <li>Title: between 1 – 63 characters</li> <li>Location: between 1 – 63 characters</li> <li>Semester: current semester or the semester that will begin in the next 30 days</li> <li>Start and end date time: End date time comes after start date time</li> <li>Start register date time: Less than end date time</li> <li>End register date time: between start register date time - end date time</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ Start check-in date time: between start register date time - end date time</li> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that's less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● When an event proposal gets created, EAMS stores it as an event with a “Draft” status. Student can then submit event proposal to staff, which turns it to “Pending” status. Staff can then reject the proposal, changing it to “Rejected”, give it some feedback for them to submit it again until it gets approved. If a staff decides to approve the event, its status changes to “Unpublished”, they will take over and manage it as if they created that event.</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are</li> </ul> </li> </ul>
--	---

	<p>recommended to assign assistants to help scan QR to check-in or check-out for the event</p> <ul style="list-style-type: none"> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> </ul>
Other Information:	In case of internet connection failure, student cannot submit event
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools' event organization departments use this feature frequently

**Table 24. <Use Case> Submit Event Proposals**

#### 2.2.2.21 Retract Event Proposals



**Figure 30. <Use Case> Retract Event Proposals**

ID and Name:	<b>UC-25. Retract Event Proposals</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Provide students with the ability to retract an event proposal in case they want to make changes to it		
Trigger:	Student sends request to retract an event		
Preconditions:	PRE-1. User has logged in the system as Student PRE-2. User is the creator of the event		

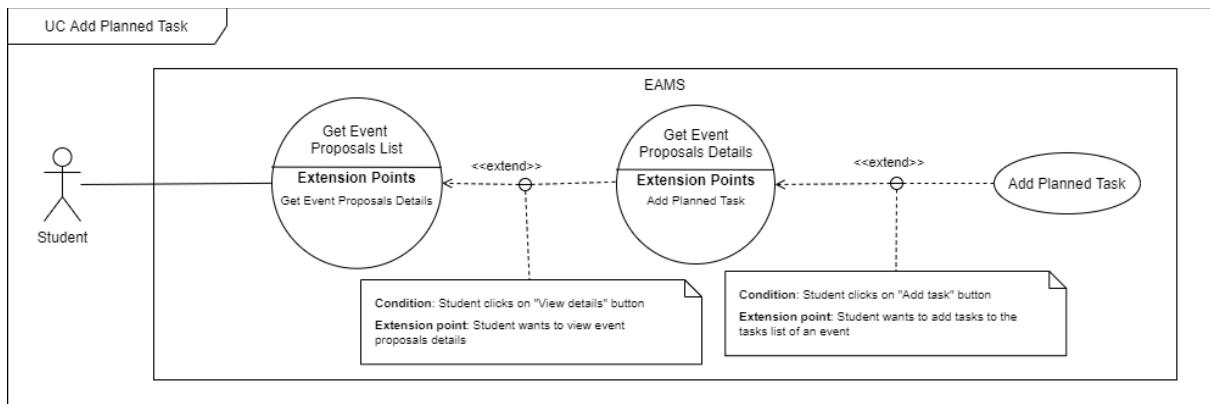
	PRE-3. The event status is “Pending”		
Post-conditions:	POST-1. EAMS changes event status to “Draft”		
Normal Flow:	Step	Actor Action	System Response
	1	Student sends request to retract event [Exception 1]	
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Event can no longer be retracted	EAMS returns to student that they can only retract pending events
Priority:	Medium		
Frequency of Use:	Most students rarely use this feature		
Business Rules:	<ul style="list-style-type: none"> <li>Students have the ability to plan, manage, submit proposals to organize events. Students can only view the events that they created, not other students’ events.</li> <li>Students can only manage basic info, criteria, total contribution grade, description, posts, and tasks of each of the events that they propose</li> <li>Students can only make changes to basic info, criteria, total contribution grade, description <u>before</u> submitting the event to staff. Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Student cannot make changes to their events during the time they wait for them to be approved</li> <li>An event requires the following fields: <ul style="list-style-type: none"> <li>Title: between 1 – 63 characters</li> <li>Location: between 1 – 63 characters</li> <li>Semester: current semester or the semester that will begin in the next 30 days</li> <li>Start and end date time: End date time comes after start date time</li> <li>Start register date time: Less than end date time</li> <li>End register date time: between start register date time - end date time</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ Start check-in date time: between start register date time - end date time</li> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that's less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● When an event proposal gets created, EAMS stores it as an event with a “Draft” status. Student can then submit event proposal to staff, which turns it to “Pending” status. Staff can then reject the proposal, changing it to “Rejected”, give it some feedback for them to submit it again until it gets approved. If a staff decides to approve the event, its status changes to “Unpublished”, they will take over and manage it as if they created that event.</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are</li> </ul> </li> </ul>
--	---

	<p>recommended to assign assistants to help scan QR to check-in or check-out for the event</p> <ul style="list-style-type: none"> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>● After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> <li>● Student can only retract events that are in “Pending” state. When an event gets retracted, its status becomes “Draft”</li> </ul>
Other Information:	In case of internet connection failure, student cannot submit event
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools' event organization departments use this feature frequently

**Table 25. <Use Case> Retract Event Proposals**

#### 2.2.2.22 Add Planned Tasks



**Figure 31. <Use Case> Add Planned Tasks**

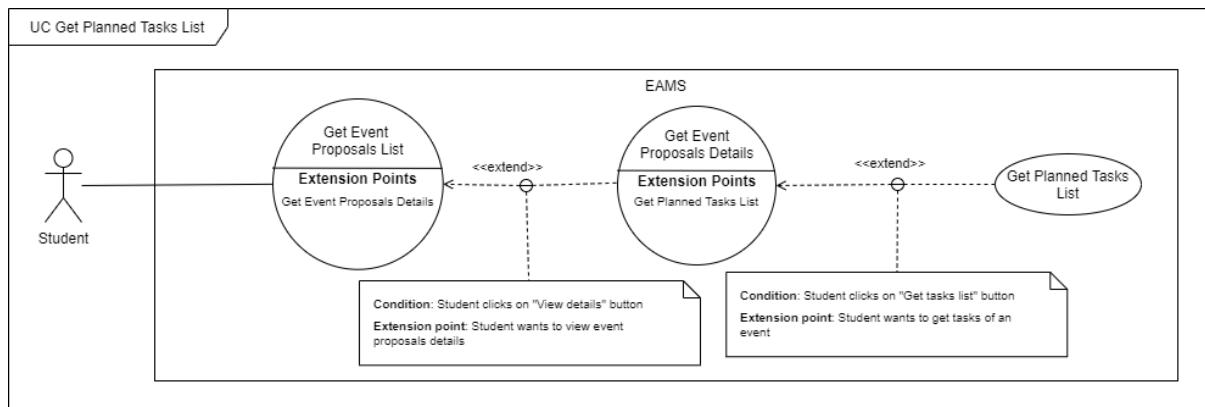
ID and Name:	<b>UC-26. Add Planned Tasks</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	In order for students to receive contribution grades, events can include tasks. This provides students the ability to add a task to an event		
Trigger:	Student send a request to add a task to an event		

Preconditions:	PRE-1. User has logged in the system as Student PRE-2. User is the creator of the event PRE-3. The event status is “Unpublished” or “Draft”		
Post-conditions:	POST-1. EAMS stores a new task for the event		
Normal Flow:	Step	Actor Action	System Response
		Students requests to add task by entering task name and contribution rate to the system [Exception 1] [Exception 2]	
		Student requests to add sub-task by entering sub-task name and contribution rate to the system [Exception 3] [Exception 4]	
		Student send request to add planned task	
			EAMS Returns to user that the task has been added successfully
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
		Task is missing name or contribution rate	EAMS returns to student that they must enter name and contribution rate for tasks
		The sum of tasks contribution rates of the event being updated does not equal to 100	EAMS returns to student that the sum of contribution rates does not equal to 100
		Sub-task is missing name or contribution rate	EAMS returns to student that they must enter name and contribution rate for sub-tasks
		The sum of tasks contribution rates of the task being added does not equal to 100	EAMS returns to student that the sum of contribution rates of sub-tasks of a task does not equal to 100
Priority:	Medium		
Frequency of Use:	Most students rarely use this feature		

Business Rules:	<ul style="list-style-type: none"> <li>Students contribute to events by completing sub-tasks of event tasks. Event tasks, sub-tasks, assignees are optional to an event</li> <li>Each task belongs to an event, must include “name” and “contribution rate”</li> <li>Each sub-task belongs to a task, must include “name” and “contribution rate”</li> <li>Each sub-task can have multiple students as assignees</li> <li>Event tasks, sub-tasks, assignees are used to record and grade students’ contribution to an event</li> <li>Event tasks must have a sum of contribution rates equals to 100%</li> <li>Sub-tasks of a task must have a sum of contribution rates equals to 100%</li> <li>Each event is created with a total contribution grade which is used to give to contributors of it. Event total contribution grade is divided to tasks based on its “contribution rate”, and then later be divided to sub-tasks and assignees.</li> <li>A task’s grade is divided to sub-tasks based on its “contribution rate”</li> <li>A sub-task’s grade is equally divided among assignees, which becomes assignee grade</li> <li>Staff evaluate the student’s performance in each sub-task by using completion rate. Each sub-task of each assignee has one completion rate. The contribution grade for one student for one sub-task is equals to its assignee grade multiplied by its completion rate</li> <li>The total contribution grade, contribution rates and completion rate of an event are decided by staffs when they give grades for an event. Student’s proposed grade is only for helping staffs to make a judgement when approving the event</li> </ul>
Other Information:	In case of internet connection failure, student cannot add task
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools’ event organization departments use this feature frequently

**Table 26.      <Use Case> Add Planned Task**

### 2.2.2.23 Get Planned Tasks List



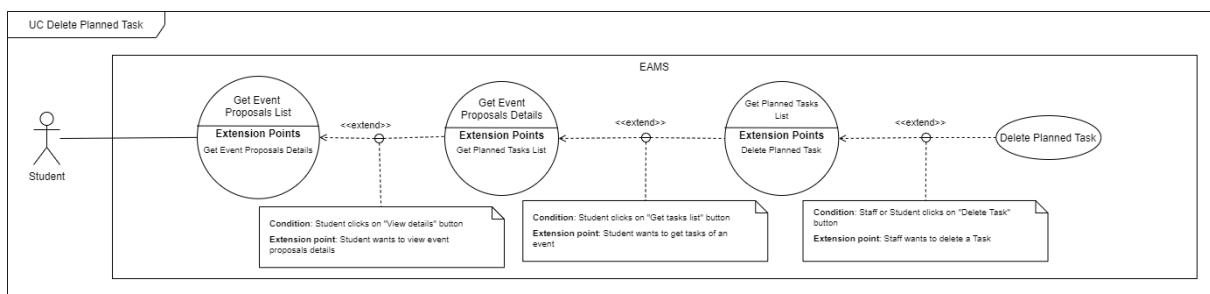
**Figure 32. <Use Case> Get Planned Tasks List**

ID and Name:	<b>UC-27. Get Planned Tasks List</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	In order for students to receive contribution grades, events can include tasks. This provides students the ability to get the tasks of an event		
Trigger:	Student requests to get the list of tasks of an event		
Preconditions:	PRE-1. User has logged in the system as Student PRE-2. User is the creator of the event		
Post-conditions:	POST-1. EAMS returns a list of tasks		
Normal Flow:	Step	Actor Action	System Response
	1	Student sends request to get tasks list of an event	
			EAMS returns the list of the tasks of that event along with their sub-tasks and assignees
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Medium		
Frequency of Use:	Most students rarely use this feature		
Business Rules:	<ul style="list-style-type: none"> <li>Students contribute to events by completing sub-tasks of event tasks. Event tasks, sub-tasks, assignees are optional to an event</li> <li>Each task belongs to an event, must include "name" and "contribution rate"</li> <li>Each sub-task belongs to a task, must include "name" and "contribution rate"</li> <li>Each sub-task can have multiple students as assignees</li> </ul>		

	<ul style="list-style-type: none"> <li>Event tasks, sub-tasks, assignees are used to record and grade students' contribution to an event</li> <li>Event tasks must have a sum of contribution rates equals to 100%</li> <li>Sub-tasks of a task must have a sum of contribution rates equals to 100%</li> <li>Each event is created with a total contribution grade which is used to give to contributors of it. Event total contribution grade is divided to tasks based on its "contribution rate", and then later be divided to sub-tasks and assignees.</li> <li>A task's grade is divided to sub-tasks based on its "contribution rate"</li> <li>A sub-task's grade is equally divided among assignees, which becomes assignee grade</li> <li>Staff evaluate the student's performance in each sub-task by using completion rate. Each sub-task of each assignee has one completion rate. The contribution grade for one student for one sub-task is equals to its assignee grade multiplied by its completion rate</li> <li>The total contribution grade, contribution rates and completion rate of an event are decided by staffs when they give grades for an event. Student's proposed grade is only for helping staffs to make a judgement when approving the event</li> </ul>
Other Information:	In case of internet connection failure, may not see the latest tasks
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools' event organization departments use this feature frequently

**Table 27. <Use Case> Get Planned Tasks List**

#### 2.2.2.24 Delete Planned Task



**Figure 33. <Use Case> Delete Planned Task**

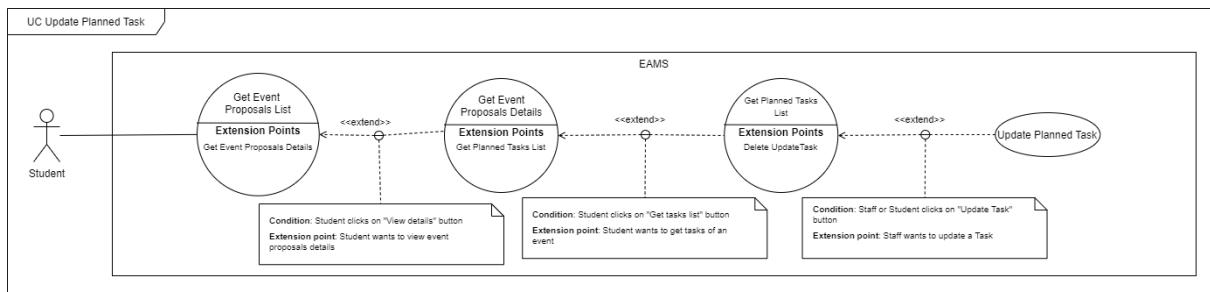
<b>ID and Name:</b>	<b>UC-28. Delete Planned Task</b>		
<b>Created By:</b>	BinhPD	<b>Date Created:</b>	02/12/20
<b>Primary Actor:</b>	Student	<b>Secondary Actors:</b>	N/A
<b>Description:</b>	Provides students the ability to delete a task of an event		

Trigger:	Student send a request to delete a task from an event		
Preconditions:	PRE-1. User has logged in the system as Student PRE-2. User is the creator of the event		
Post-conditions:	POST-1. EAMS delete a task of an event		
Normal Flow:	Step	Actor Action	System Response
	1	Student sends request to delete a task of an event	
			EAMS delete the tasks from that event along with their sub-tasks and assignees
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Medium		
Frequency of Use:	Most students rarely use this feature		
Business Rules:	<ul style="list-style-type: none"> <li>Students contribute to events by completing sub-tasks of event tasks. Event tasks, sub-tasks, assignees are optional to an event</li> <li>Each task belongs to an event, must include “name” and “contribution rate”</li> <li>Each sub-task belongs to a task, must include “name” and “contribution rate”</li> <li>Each sub-task can have multiple students as assignees</li> <li>Event tasks, sub-tasks, assignees are used to record and grade students’ contribution to an event</li> <li>Event tasks must have a sum of contribution rates equals to 100%</li> <li>Sub-tasks of a task must have a sum of contribution rates equals to 100%</li> <li>Each event is created with a total contribution grade which is used to give to contributors of it. Event total contribution grade is divided to tasks based on its “contribution rate”, and then later be divided to sub-tasks and assignees.</li> <li>A task’s grade is divided to sub-tasks based on its “contribution rate”</li> <li>A sub-task’s grade is equally divided among assignees, which becomes assignee grade</li> <li>Staff evaluate the student’s performance in each sub-task by using completion rate. Each sub-task of each assignee has one completion rate. The contribution grade for one student for one sub-task is equals to its assignee grade multiplied by its completion rate</li> <li>The total contribution grade, contribution rates and completion rate of an event are decided by staffs when they give grades for</li> </ul>		

	an event. Student's proposed grade is only for helping staffs to make a judgement when approving the event
Other Information:	In case of internet connection failure, cannot delete tasks
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools' event organization departments use this feature frequently

**Table 28. <Use Case> Delete Planned Task**

#### 2.2.2.25 Update Planned Task



**Figure 34. <Use Case> Update Planned Task**

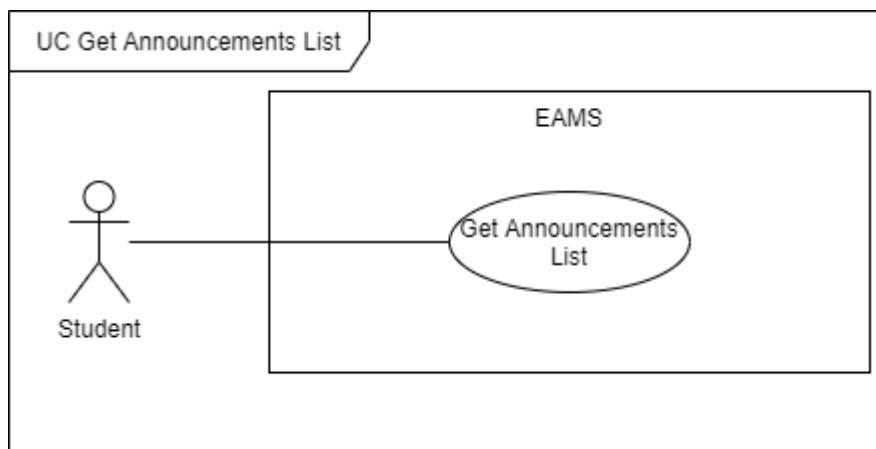
ID and Name:	UC-29. Update Planned Task		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	In order for students to receive contribution grades, events can include tasks. This provides students the ability to update a task of an event		
Trigger:	Student send a request to add a task to an event		
Preconditions:	PRE-1. User has logged in the system as Student PRE-2. User is the creator of the event PRE-3. The event status is "Unpublished" or "Draft"		
Post-conditions:	POST-1. EAMS update a task of an event		
Normal Flow:	Step	Actor Action	System Response
	1	Student update task name and contribution rate [Exception 1] [Exception 2]	
	2	Student update sub-task name and contribution rate [Exception 3] [Exception 4]	
	3	Student add or remove assignee by entering and removing assignees' emails	

	4	Student send request to update planned task	
			EAMS returns to user that the task has been updated successfully
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Task is missing name or contribution rate	EAMS returns to student that they must enter name and contribution rate for tasks
	2	The sum of tasks contribution rates of the event being updated does not equal to 100	EAMS returns to student that the sum of contribution rates does not equal to 100
	3	Sub-task is missing name or contribution rate	EAMS returns to student that they must enter name and contribution rate for sub-tasks
	4	The sum of tasks contribution rates of the task being added does not equal to 100	EAMS returns to student that the sum of contribution rates of sub-tasks of a task does not equal to 100
Priority:	Medium		
Frequency of Use:	Most students rarely use this feature		
Business Rules:	<ul style="list-style-type: none"> <li>Students contribute to events by completing sub-tasks of event tasks. Event tasks, sub-tasks, assignees are optional to an event</li> <li>Each task belongs to an event, must include “name” and “contribution rate”</li> <li>Each sub-task belongs to a task, must include “name” and “contribution rate”</li> <li>Each sub-task can have multiple students as assignees</li> <li>Event tasks, sub-tasks, assignees are used to record and grade students’ contribution to an event</li> <li>Event tasks must have a sum of contribution rates equals to 100%</li> <li>Sub-tasks of a task must have a sum of contribution rates equals to 100%</li> <li>Each event is created with a total contribution grade which is used to give to contributors of it. Event total contribution grade is divided to tasks based on its “contribution rate”, and then later be divided to sub-tasks and assignees.</li> <li>A task’s grade is divided to sub-tasks based on its “contribution rate”</li> </ul>		

	<ul style="list-style-type: none"> <li>• A sub-task's grade is equally divided among assignees, which becomes assignee grade</li> <li>• Staff evaluate the student's performance in each sub-task by using completion rate. Each sub-task of each assignee has one completion rate. The contribution grade for one student for one sub-task is equals to its assignee grade multiplied by its completion rate</li> <li>• The total contribution grade, contribution rates and completion rate of an event are decided by staffs when they give grades for an event. Student's proposed grade is only for helping staffs to make a judgement when approving the event</li> </ul>
Other Information:	In case of internet connection failure, student cannot update task
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools' event organization departments use this feature frequently

**Table 29. <Use Case> Update Planned Task**

#### 2.2.2.26 Get Announcements List



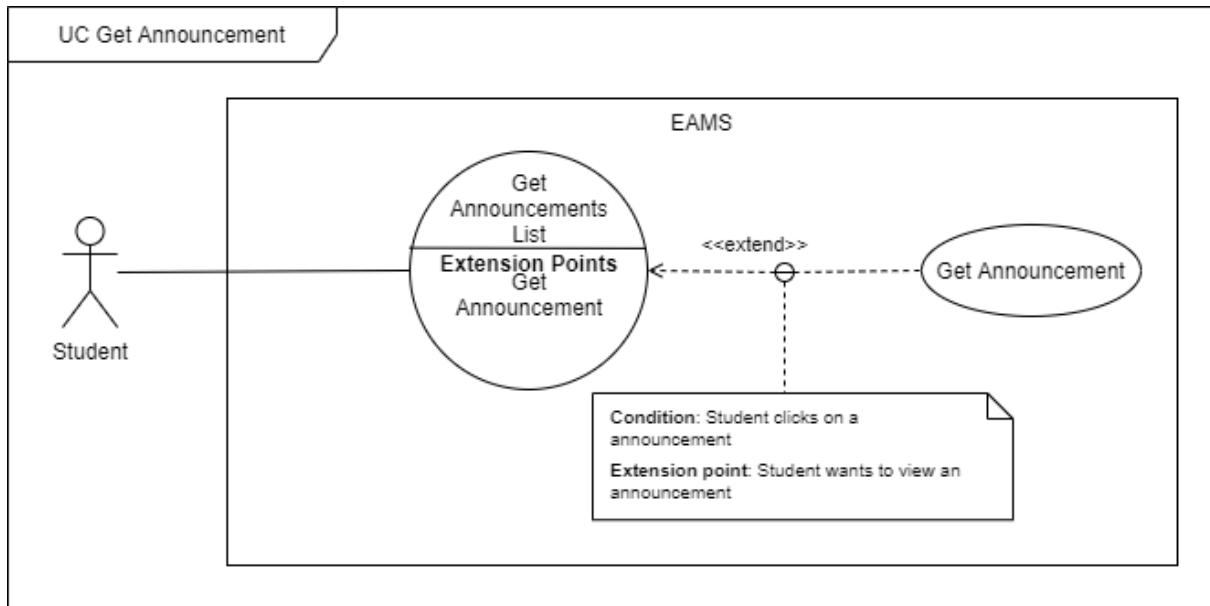
**Figure 35. <Use Case> Get Announcements List**

ID and Name:	<b>UC-30. Get Announcements List</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Get notifications list about events that involve this student		
Trigger:	Student requests to get notifications		
Preconditions:	PRE-1. User has logged in as Student role		
Post-conditions:	POST-1. Notifications list and unread number are returned to student		
Normal Flow:	Step	Actor Action	System Response
	1	Student requests list of their notifications	

		[Exception 1]	
			EAMS returns notifications list
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	This student has no notifications	EAMS returns to student that there is no notification yet
Priority:	Low		
Frequency of Use:	Approximately 2000 users, average of one usage per week		
Business Rules:	<ul style="list-style-type: none"> <li>Announcement is a mean for staff to communicate with students more effectively using emails about information such as new achievements and urgent messages.</li> <li>Announcements list also let student know about unread messages</li> </ul>		
Other Information:	In case of internet connection failure, student cannot finish this use case		
Assumptions:	N/A		

**Table 30. <Use Case> Get Announcements List**

#### 2.2.2.27 Get Announcement



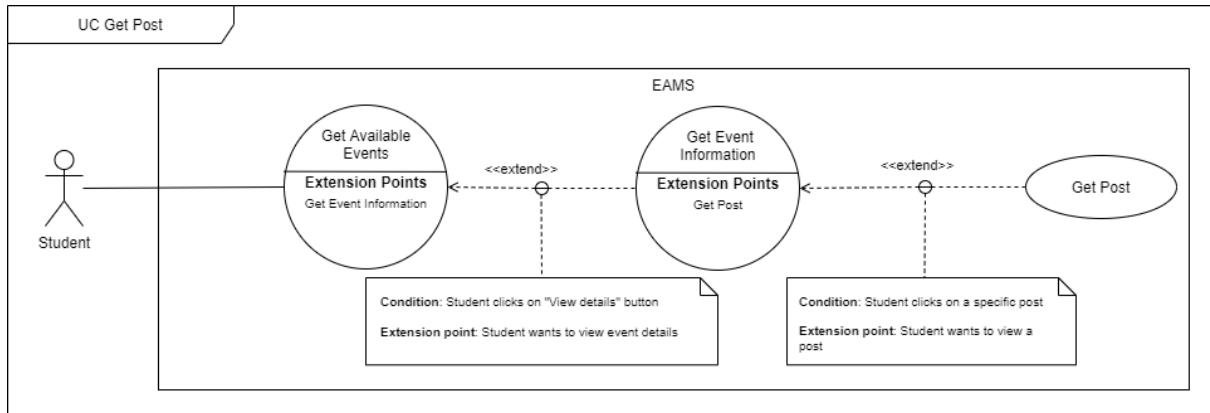
**Figure 36. <Use Case> Get Announcement**

ID and Name:	<b>UC-31. Get Announcement</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Get notification details, which matches to received email		
Trigger:	Student requests to get a specific notification's detailed information		
Preconditions:	PRE-1. User has logged in as Student role		

Post-conditions:	POST-1. Notification's details are displayed POST-2. Notification's status changes from unread to read		
Normal Flow:	Step	Actor Action	System Response
	1	Student requests detailed information of a specific notification [Exception 1]	
			EAMS returns notification detail
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	This notification is no longer available	EAMS returns to student that this notification is not available
Priority:	Low		
Frequency of Use:	Approximately 2000 users, average of one usage per week		
Business Rules:	<ul style="list-style-type: none"> <li>Announcement is a mean for staff to communicate with students more effectively using emails about information such as new achievements and urgent messages.</li> <li>An announcement relates to a specific event</li> <li>Announcement content is similar to email they received from system</li> <li>When student gets an announcement, its status change from unread to read</li> </ul>		
Other Information:	<ul style="list-style-type: none"> <li>Notification's content is similar to email they received from system</li> <li>In case of internet connection failure, student cannot get notification details</li> </ul>		
Assumptions:	N/A		

**Table 31. <Use Case> Get Announcement**

#### 2.2.2.28 Get Post

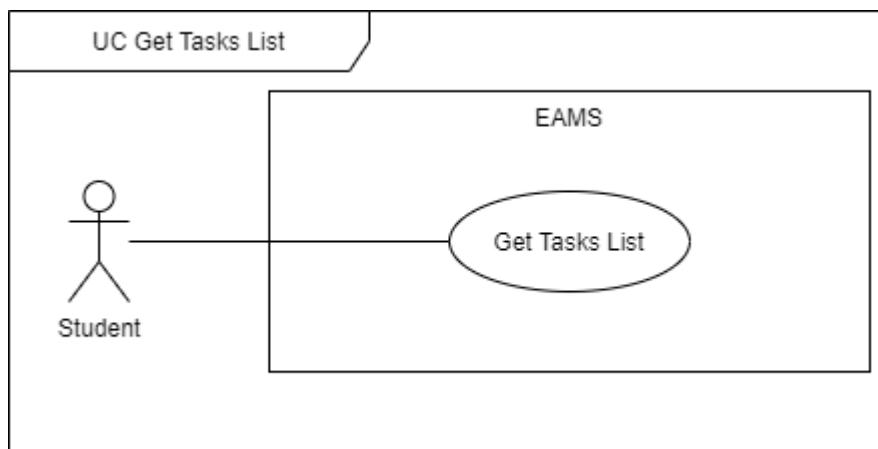


**Figure 37. <Use Case> Get Post**

ID and Name:	<b>UC-32. Get Post</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	Get event's post, which is additional information for event's description		
Trigger:	Student requests to get a specific event's post details		
Preconditions:	PRE-1. User has logged in as Student role		
Post-conditions:	POST-1. Post's details are returned		
Normal Flow:	Step	Actor Action	System Response
	1	Student requests detailed information of a post [Exception 1]	
			EAMS returns post content
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	This post is no longer available	EAMS returns to student that this post is not available
Priority:	Low		
Frequency of Use:	Approximately 2000 users, average of one usage per week		
Business Rules:	<ul style="list-style-type: none"> <li>Some events need to provide more detail information which cannot be expressed clearly in event's description or short description. Post can tackle this problem as a mean of updating new information for students.</li> <li>A post belongs to a specific event</li> </ul>		
Other Information:	In case of internet connection failure, student cannot get event post		
Assumptions:	N/A		

**Table 32. <Use Case> Get Post**

### 2.2.2.29 Get Tasks List



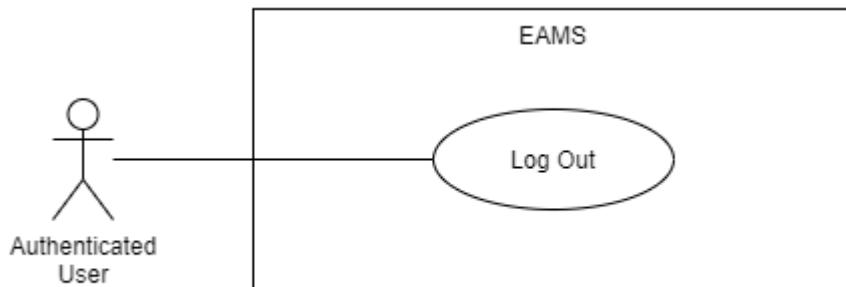
**Figure 38. <Use Case> Get Tasks List**

ID and Name:	UC-33. Get Tasks List		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Student	Secondary Actors:	N/A
Description:	A student gets list of event tasks that he or she has been assigned to in order to make contributions to them so that they may later receive contribution grades		
Trigger:	Student requests to get their list of assigned tasks		
Preconditions:	PRE-1. User has logged in as Student role		
Post-conditions:	POST-1. Return list of event tasks of this student		
Normal Flow:	Step	Actor Action	System Response
	1	Student requests to get event tasks list [Exception 1]	
			EAMS returns assigned tasks of this student
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	There is no assigned tasks	EAMS return that there is no assigned task of them
Priority:	Medium		
Frequency of Use:	Rarely		
Business Rules:	<ul style="list-style-type: none"> <li>Student can only see their tasks list of published, finished or cancelled events</li> <li>Each task belongs to an event. A student can have multiple tasks for one event</li> <li>A student can not get tasks of other students.</li> </ul>		

Other Information:	In case of internet connection failure, student cannot get tasks list
Assumptions:	N/A

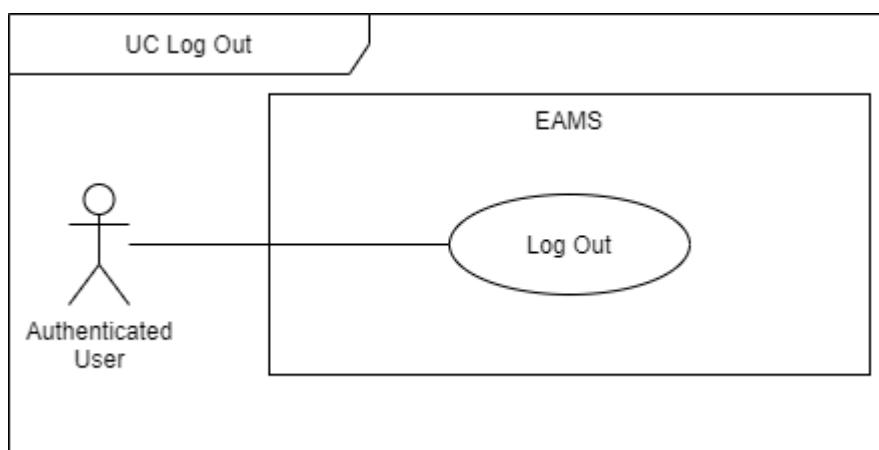
**Table 33. <Use Case> Get Tasks List**

### 2.2.3 Authenticated User



**Figure 39. <Use Case Overview> Authenticated User**

#### 2.2.3.1 Logout



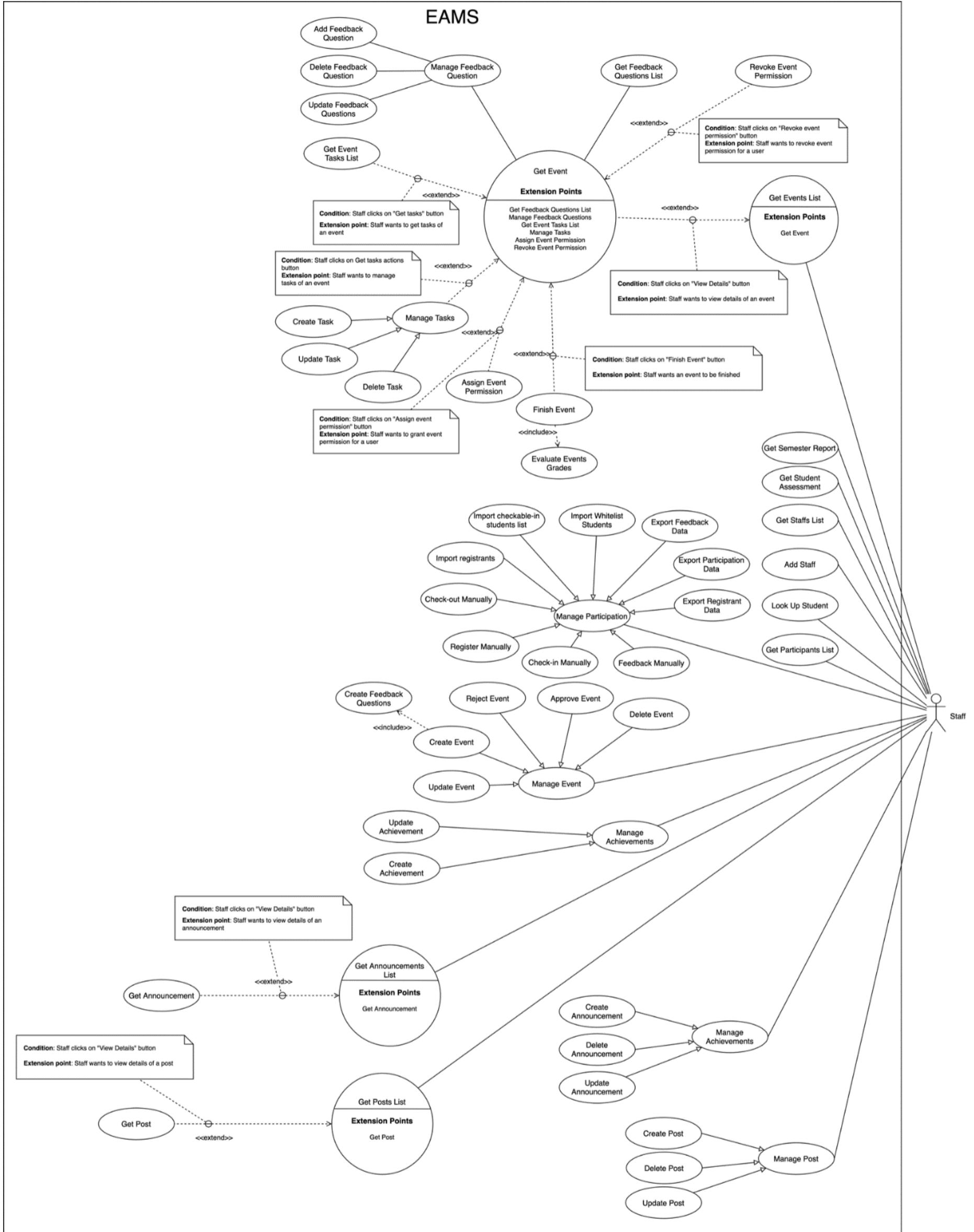
**Figure 40. <Use Case> Log Out**

ID and Name:	<b>UC-34. Log Out</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Authenticated User	Secondary Actors:	N/A
Description:	Authenticated user logs out EAMS		
Trigger:	Authenticated user requests to logout system		
Preconditions:	PRE-1. User has logged in EAMS		
Post-conditions:	POST-1. EAMS delete their login session POST-2. User is navigated to login page		
Normal Flow:	Step	Actor Action	System Response
	1	Authenticated user requests to logout system	
			EAMS clears their login token Navigates user to login page

Alternative Flows:	N/A
Exceptions:	N/A
Priority:	Medium
Frequency of Use:	Rarely
Business Rules:	<ul style="list-style-type: none"> <li>• If an authenticated user logs out, EAMS clears their login session, their register information in case of offline check-in</li> </ul>
Other Information:	N/A
Assumptions:	N/A

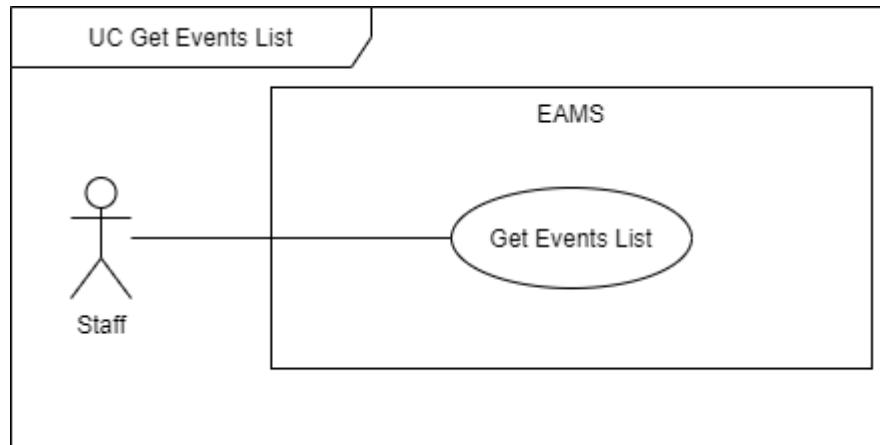
***Table 34. <Use Case> Log Out***

## 2.2.4 Staff



**Figure 41. <Use Case Overview> Staff Use Cases**

#### 2.2.4.1 Get Events List



**Figure 42. <Use Case> Get Events List**

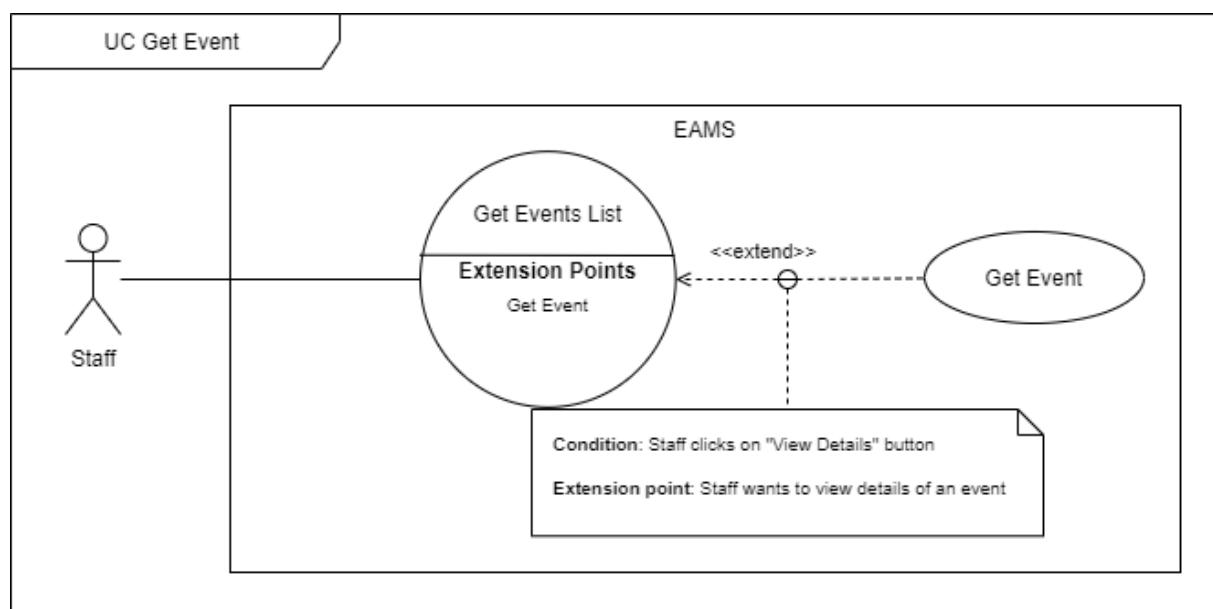
ID and Name:	<b>UC-35. Get Events List</b>		
Created By:	BinhhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allow staff to get all the events that they manage		
Trigger:	Staff sends request to get events list		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS return events list		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request to get the events that they manage	
			EAMS returns the events that staff manages
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Staff has no event to manage	EAMS returns "there's no record to display" message
Priority:	High		
Frequency of Use:	Approximately 10 usages per day. Peak usage load for this use case happens when a semester is about to end		
Business Rules:	<ul style="list-style-type: none"> <li>Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Staff manages all events in the university, excluding student's draft events</li> <li>An event requires the following fields:</li> </ul>		

	<ul style="list-style-type: none"> <li>○ Title: between 1 – 63 characters</li> <li>○ Location: between 1 – 63 characters</li> <li>○ Semester: current semester or the semester that will begin in the next 30 days</li> <li>○ Start and end date time: End date time comes after start date time</li> <li>○ Start register date time: Less than end date time</li> <li>○ End register date time: between start register date time - end date time</li> <li>○ Start check-in date time: between start register date time - end date time</li> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that's less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> </ul> </li> </ul>
--	---

	<ul style="list-style-type: none"> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> <li>● Staff can make updates to an event as long as its status is “Unpublished” or “Published”</li> <li>● After an event finishes, staff can start looking up student information, grant achievements for students that participate in or contributes too it</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university frequently organize events that have an average participations of 40 students

**Table 35.      <Use Case> Get Events List**

#### 2.2.4.2 Get Event



**Figure 43. <Use Case> Get Event**

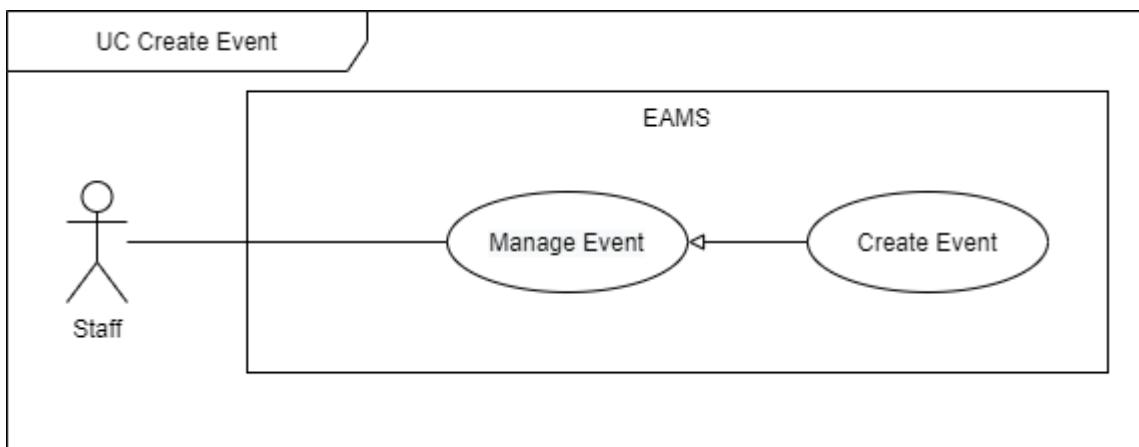
ID and Name:	<b>UC-36. Get Event</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allow staff to get details of an event that they manage		
Trigger:	Staff sends request to get event details		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS returns event details		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request to get the details of an event that they manage	
			EAMS returns the requested event details
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Staff has no event to manage	EAMS returns “there’s no record to display” message
Priority:	High		
Frequency of Use:	Approximately 20 usages per day. Peak usage load for this use case happens when multiple events is happening		
Business Rules:	<ul style="list-style-type: none"> <li>Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Staff manages all events in the university, excluding student’s draft events</li> <li>An event requires the following fields: <ul style="list-style-type: none"> <li>Title: between 1 – 63 characters</li> <li>Location: between 1 – 63 characters</li> <li>Semester: current semester or the semester that will begin in the next 30 days</li> <li>Start and end date time: End date time comes after start date time</li> <li>Start register date time: Less than end date time</li> <li>End register date time: between start register date time - end date time</li> <li>Start check-in date time: between start register date time - end date time</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that's less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> </ul> </li> </ul>
--	---

	<ul style="list-style-type: none"> <li>When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> <li>Staff can make updates to an event as long as its status is “Unpublished” or “Published”</li> <li>After an event finishes, staff can start looking up student information, grant achievements for students that participate in or contributes to it</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university frequently organize events that have an average participations of 40 students

**Table 36. <Use Case> Get Event**

#### 2.2.4.3 Create Event



**Figure 44. <Use Case> Create Event**

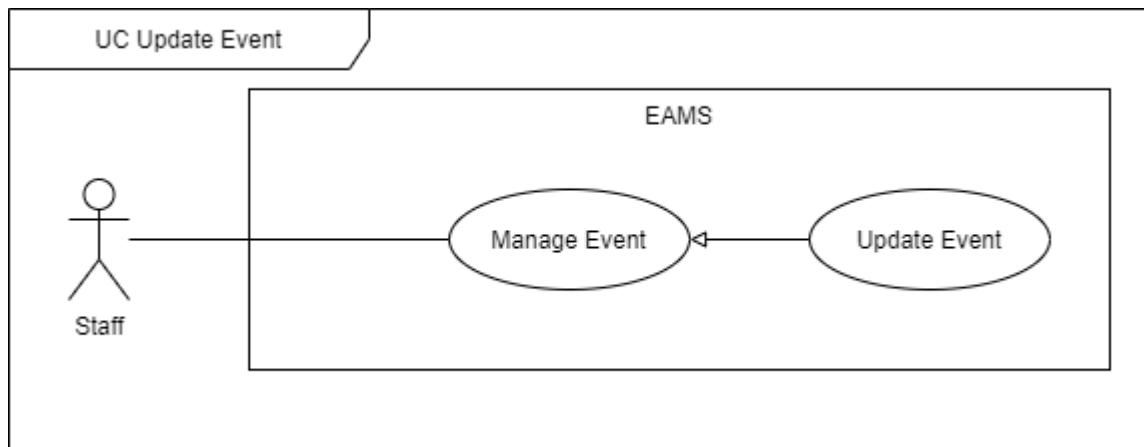
ID and Name:	<b>UC-37. Create Event</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allows staff to create an event		
Trigger:	Staff sends request to create event		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS stores event details		
Normal Flow:	Step	Actor Action	System Response
	1	Staff fills in the event information and sends requests to create the event [Exception 1]	

			EAMS validates the event information and stores them
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Staff fails to fill in all the required fields	EAMS returns to staff to fill in the missing fields
Priority:	High		
Frequency of Use:	Approximately 2 usages per week.		
Business Rules:	<ul style="list-style-type: none"> <li>Staff manages all events in the university, excluding student's draft events</li> <li>An event requires the following fields: <ul style="list-style-type: none"> <li>Title: between 1 – 63 characters</li> <li>Location: between 1 – 63 characters</li> <li>Semester: current semester or the semester that will begin in the next 30 days</li> <li>Start and end date time: End date time comes after start date time</li> <li>Start register date time: Less than end date time</li> <li>End register date time: between start register date time - end date time</li> <li>Start check-in date time: between start register date time - end date time</li> <li>End check-in date time: between start check-in date time – end date time</li> <li>Whether or not event has registration or check-in restrictions</li> <li>Short description: between 1 – 127 characters</li> <li>Description: between 1 – 4095 characters</li> <li>Image: URL that's less than 1023 characters</li> <li>At least one and maximum of 10 feedback questions</li> <li>Start feedback date time: between start date time – end date time</li> <li>One grade criteria of a policy document that is being applied in the chosen semester</li> <li>One grade sub-criteria that belongs to the chosen criteria</li> <li>Total contribution grade: positive, less than 32767</li> </ul> </li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>Note: Additional info for students</li> <li>Checkout requirement: determines if event requires students to checkout</li> <li>If event requires checking out, start and end checkout date time is required</li> </ul>		

	<ul style="list-style-type: none"> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> </ul> </li> <li>● Staff can make updates to an event as long as its status is “Unpublished” or “Published”</li> <li>● After an event finishes, staff can start looking up student information, grant achievements for students that participate in or contributes to it</li> <li>● Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university frequently organize events that have an average participations of 40 students

**Table 37.      <Use Case> Create Event**

#### 2.2.4.4 Update Event



**Figure 45. <Use Case> Update Event**

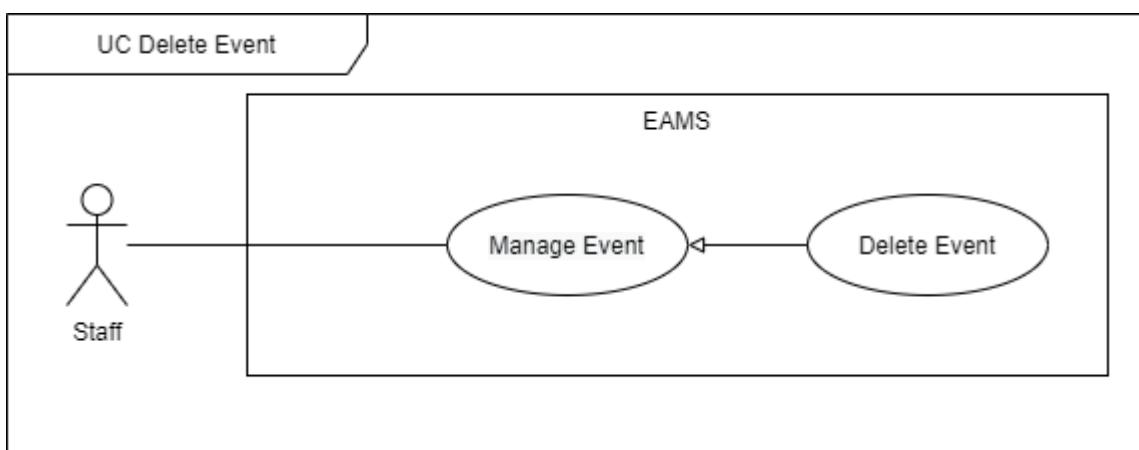
ID and Name:	<b>UC-38. Update Event</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allow staff to update an event		
Trigger:	Staff sends request to update event		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS stores the updated event details		
Normal Flow:	Step	Actor Action	System Response
	1	Staff fills in the event information and sends requests to update the event [Exception 1]	
			EAMS validates the updated event information and stores them
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Staff fails to fill in all the required fields	EAMS returns to staff to fill in the missing fields
Priority:	High		
Frequency of Use:	Approximately 10 usages per event.		
Business Rules:	<ul style="list-style-type: none"> <li>Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Staff manages all events in the university, excluding student's draft events</li> </ul>		

	<ul style="list-style-type: none"> <li>• An event requires the following fields: <ul style="list-style-type: none"> <li>○ Title: between 1 – 63 characters</li> <li>○ Location: between 1 – 63 characters</li> <li>○ Semester: current semester or the semester that will begin in the next 30 days</li> <li>○ Start and end date time: End date time comes after start date time</li> <li>○ Start register date time: Less than end date time</li> <li>○ End register date time: between start register date time - end date time</li> <li>○ Start check-in date time: between start register date time - end date time</li> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that's less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> </li> </ul> <p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>• After creation, the event semester cannot be changed</li> <li>• An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> </ul> </li> </ul>
--	--

	<ul style="list-style-type: none"> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> <li>● Staff can make updates to an event as long as its status is “Unpublished” or “Published”</li> <li>● After an event finishes, staff can start looking up student information, grant achievements for students that participate in or contributes too it</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university frequently organize events that have an average participations of 40 students

**Table 38. <Use Case> Update Event**

#### 2.2.4.5 Delete Event



**Figure 46. <Use Case> Delete Event**

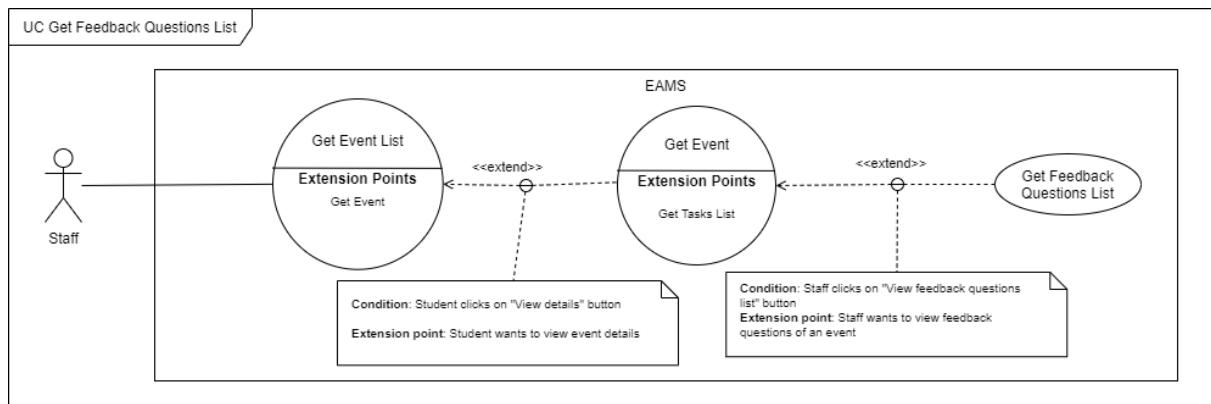
ID and Name:	<b>UC-39. Delete Event</b>		
Created By:	BinhPD	Date Created:	02/12/20

Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allow staff to delete an event		
Trigger:	Staff sends request to delete an event		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS removes the event		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request to delete event [Exception 1]	
			EAMS Returns to user that event has been deleted
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Staff can only delete unpublished events	EAMS informs that staff can only delete events that are unpublished and have no registrations yet
Priority:	Medium		
Frequency of Use:	Events are rarely delete unless staff accidentally does it		
Business Rules:	<ul style="list-style-type: none"> <li>Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Staff manages all events in the university, excluding student's draft events</li> <li>An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>When created by a staff or submitted by a student and approved, its default status is "Unpublished". This is when staff review event information before publishing the event for students.</li> <li>When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes "Published". This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> <li>When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>When event starts allowing students to feedback, staff monitors feedback results</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> <li>● Staff can make updates to an event as long as its status is “Unpublished” or “Published”</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university frequently organize events that have an average participations of 40 students

**Table 39. <Use Case> Delete Event**

#### 2.2.4.6 Get Feedback Questions List



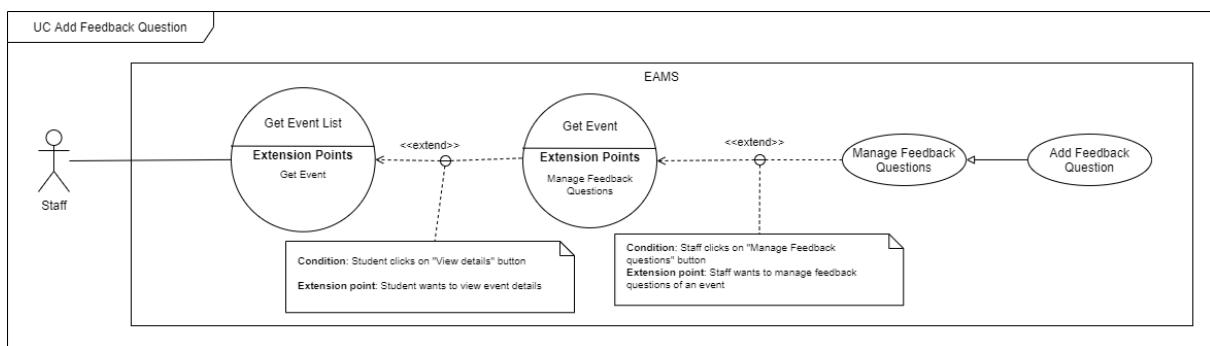
**Figure 47. <Use Case> Get Feedback Questions List**

ID and Name:	<b>UC-40. Get Feedback Questions List</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allow staff to get all the feedback questions of an event that they manage		
Trigger:	Staff sends request to get feedback questions list		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS return event's feedback questions list		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request to get the feedback questions of the events that they manage [Exception 1]	
			EAMS returns the event's feedback questions

Alternative Flows:	N/A
Exceptions:	N/A
Priority:	High
Frequency of Use:	Approximately 3 usages per event. Peak usage load for this use case happens when a semester is about to start feedback
Business Rules:	<ul style="list-style-type: none"> <li>Students can only give feedbacks for events that they have checked-in. Feedback is optional unless the organizer wants students to feedback before checking out</li> <li>When giving feedback, users have to fill answer for all feedback questions. There are 2 types of feedback question: <ul style="list-style-type: none"> <li>Text: user can freely type, length between 1 - 255</li> <li>Rating: user have to input from 1 to 5</li> </ul> </li> <li>An event must have at least one and no more than 10 feedback questions</li> <li>The university staffs have to create feedback questions when they create the event. They also have to decide when the students can start giving feedback by specifying start feedback date time.</li> <li>Staff cannot make changes to the feedback questions list after start register date time</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university frequently organize events that have an average participations number of 40 students. Each event usually has 10 feedback questions. Approximately 70 of participants gives feedback

**Table 40. <Use Case> Get Feedback Questions List**

#### 2.2.4.7 Add Feedback Question



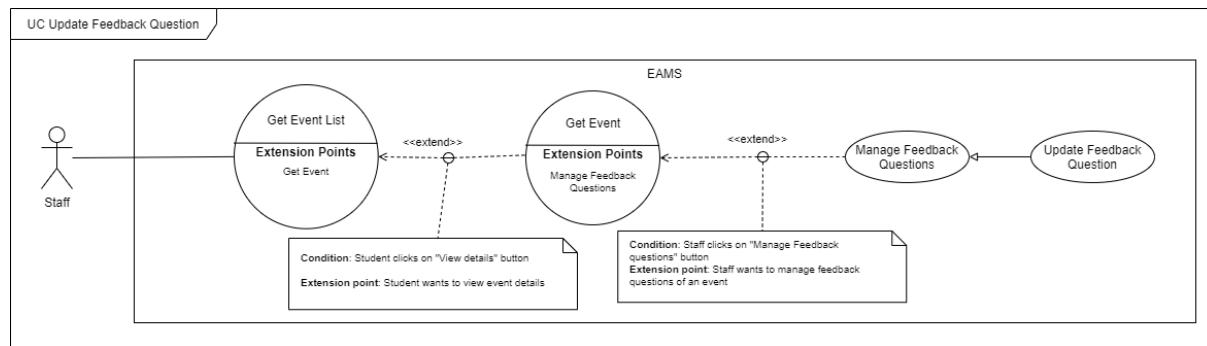
**Figure 48. <Use Case> Add Feedback Question**

ID and Name:	<b>UC-41. Add Feedback Question</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allow staff to add a feedback question to an event		
Trigger:	Staff sends request to add a feedback question to an event		

Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS stores the feedback question as part of an event		
Normal Flow:	Step	Actor Action	System Response
	1	Staff fills in the feedback question details and sends requests [Exception 1]	
Alternative Flows:	N/A		
	No	Cause	System Response
Exceptions:	1	Staff fails to fill in all the required fields	EAMS returns to staff to fill in the missing fields
	Priority: High		
Frequency of Use:	About 7-10 usages per event		
Business Rules:	<ul style="list-style-type: none"> <li>Students can only give feedbacks for events that they have checked-in. Feedback is optional unless the organizer wants students to feedback before checking out</li> <li>When giving feedback, users have to fill answer for all feedback questions. There are 2 types of feedback question: <ul style="list-style-type: none"> <li>Text: user can freely type, length between 1 - 255</li> <li>Rating: user have to input from 1 to 5</li> </ul> </li> <li>An event must have at least one and no more than 10 feedback questions</li> <li>The university staffs have to create feedback questions when they create the event. They also have to decide when the students can start giving feedback by specifying start feedback date time.</li> <li>Staff cannot make changes to the feedback questions list after start register date time</li> </ul>		
Other Information:	In case of internet connection failure, staffs cannot use this feature		
Assumptions:	The university frequently organize events that have an average participations number of 40 students. Each event usually has 10 feedback questions. Approximately 70 of participants gives feedback		

**Table 41.      <Use Case> Add Feedback Question**

#### 2.2.4.8 Update Feedback Question



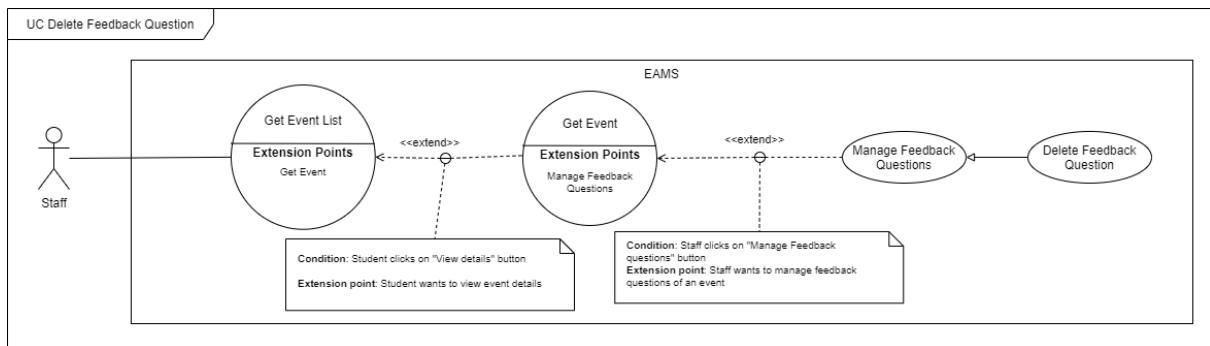
**Figure 49. <Use Case> Update Feedback Question**

ID and Name:	<b>UC-42. Update Feedback Question</b>		
Created By:	BinhdPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allows staff to update a feedback question to an event		
Trigger:	Staff sends request to update a feedback question to an event		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS updates the feedback question		
Normal Flow:	Step	Actor Action	System Response
	1	Staff fills in the feedback question details and sends requests [Exception 1]	
			EAMS validates the question information and stores them
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Staff fails to fill in all the required fields	EAMS returns to staff to fill in the missing fields
Priority:	High		
Frequency of Use:	About 7-10 usages per event		
Business Rules:	<ul style="list-style-type: none"> <li>Students can only give feedbacks for events that they have checked-in. Feedback is optional unless the organizer wants students to feedback before checking out</li> <li>When giving feedback, users have to fill an answer for all feedback questions. There are 2 types of feedback question: <ul style="list-style-type: none"> <li>Text: user can freely type, length between 1 - 255</li> <li>Rating: user have to input from 1 to 5</li> </ul> </li> <li>An event must have at least one and no more than 10 feedback questions</li> </ul>		

	<ul style="list-style-type: none"> <li>The university staffs have to create feedback questions when they create the event. They also have to decide when the students can start giving feedback by specifying start feedback date time.</li> <li>Staff cannot make changes to the feedback questions list after start register date time</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university frequently organize events that have an average participations number of 40 students. Each event usually has 10 feedback questions. Approximately 70 of participants gives feedback

**Table 42. <Use Case> Update Feedback Question**

#### 2.2.4.9 Delete Feedback Question



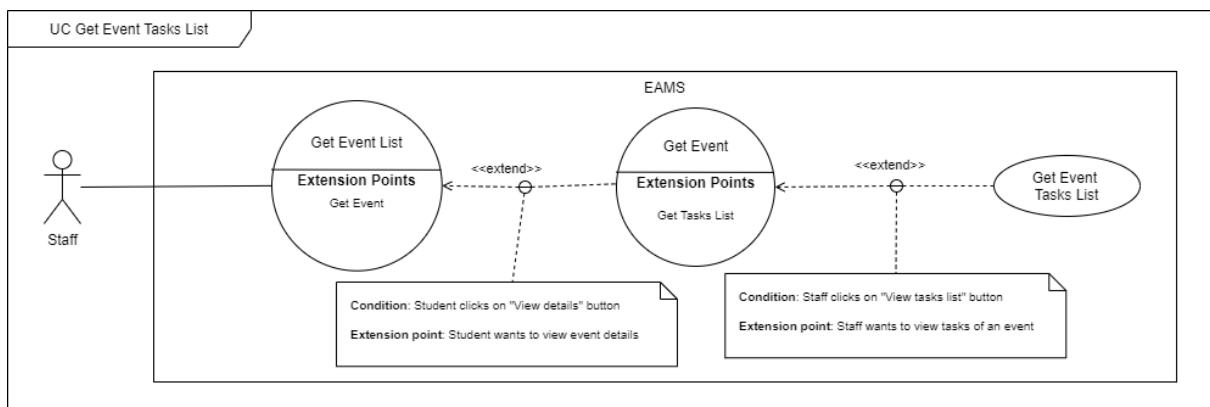
**Figure 50. <Use Case> Delete Feedback Question**

ID and Name:	<b>UC-43. Delete Feedback Question</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allow staff to delete a feedback question from an event		
Trigger:	Staff sends request to delete a feedback question from an event		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS delete the feedback question		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends a request to delete the feedback question [Exception 1]	
			EAMS deletes the feedback question from the event
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Staff deletes the last question of the event	EAMS inform user that at least one feedback questions is required

Priority:	High
Frequency of Use:	About 3 usages per event
Business Rules:	<ul style="list-style-type: none"> <li>Students can only give feedbacks for events that they have checked-in. Feedback is optional unless the organizer wants students to feedback before checking out</li> <li>When giving feedback, users have to fill answer for all feedback questions. There are 2 types of feedback question: <ul style="list-style-type: none"> <li>Text: user can freely type, length between 1 - 255</li> <li>Rating: user have to input from 1 to 5</li> </ul> </li> <li>An event must have at least one and no more than 10 feedback questions</li> <li>The university staffs have to create feedback questions when they create the event. They also have to decide when the students can start giving feedback by specifying start feedback date time.</li> <li>Staff cannot make changes to the feedback questions list after start register date time</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university frequently organize events that have an average participations number of 40 students. Each event usually has 10 feedback questions. Approximately 70 of participants gives feedback

**Table 43. <Use Case> Delete Feedback Question**

#### 2.2.4.10 Get Event Tasks List



**Figure 51. <Use Case> Get Event Tasks List**

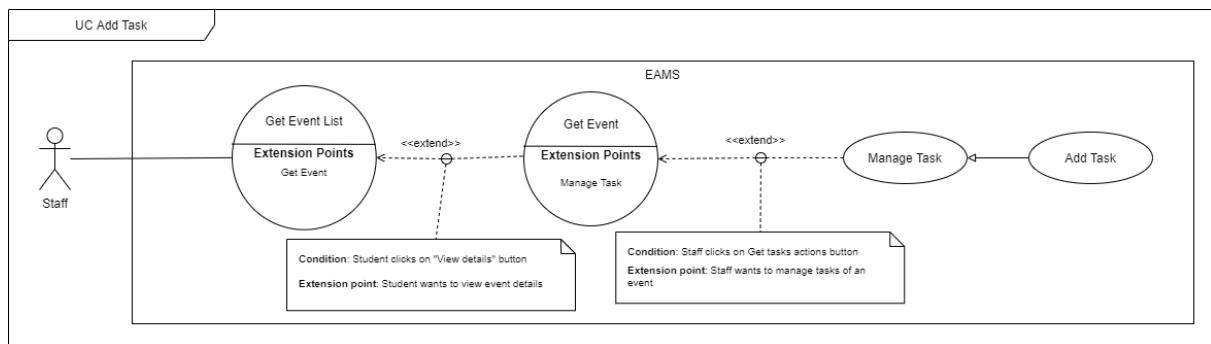
ID and Name:	<b>UC-44. Get Event Tasks List</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allow staff to get all the tasks of an event that they manage		
Trigger:	Staff sends request to get tasks list		

Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS return event's tasks list		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request to get the tasks of the events that they manage [Exception 1]	
Alternative Flows:	N/A		
	No	Cause	System Response
Exceptions:	1	The tasks list is empty	EAMS informs user that there is no tasks
Priority:	High		
Frequency of Use:	The more assignees, the more likely this feature is used frequently		
Business Rules:	<ul style="list-style-type: none"> <li>Students contribute to events by completing sub-tasks of event tasks. Event tasks, sub-tasks, assignees are optional to an event</li> <li>Each task belongs to an event, must include "name" and "contribution rate"</li> <li>Each sub-task belongs to a task, must include "name" and "contribution rate"</li> <li>Each sub-task can have multiple students as assignees</li> <li>Event tasks, sub-tasks, assignees are used to record and grade students' contribution to an event</li> <li>Event tasks must have a sum of contribution rates equals to 100%</li> <li>Sub-tasks of a task must have a sum of contribution rates equals to 100%</li> <li>Each event is created with a total contribution grade which is used to give to contributors of it. Event total contribution grade is divided to tasks based on its "contribution rate", and then later be divided to sub-tasks and assignees.</li> <li>A task's grade is divided to sub-tasks based on its "contribution rate"</li> <li>A sub-task's grade is equally divided among assignees, which becomes assignee grade</li> <li>Staff evaluate the student's performance in each sub-task by using completion rate. Each sub-task of each assignee has one completion rate. The contribution grade for one student for one sub-task is equals to its assignee grade multiplied by its completion rate</li> <li>The total contribution grade, contribution rates and completion rate of an event are decided by staffs when they give grades for</li> </ul>		

	an event. Student's proposed grade is only for helping staffs to make a judgement when approving the event
Other Information:	<ul style="list-style-type: none"> <li>In case of internet connection failure, staffs cannot use this feature</li> </ul>
Assumptions:	The university has incentives for students to contribute in events. Events tasks are assigned fairly to students

**Table 44. <Use Case> Get Event Tasks List**

#### 2.2.4.11 Add Task



**Figure 52. <Use Case> Add Task**

ID and Name:	UC-45. Add Task		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allow staff to add a task to an event for students to contribute to		
Trigger:	Staff sends request to add a task to an event		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS stores the task as part of an event		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to add task by entering task name and contribution rate to the system [Exception 1] [Exception 2]	
	2	Staff requests to add sub-task by entering sub-task name and contribution rate to the system [Exception 3] [Exception 4]	
	3	Staff requests to add assignee by entering	

		assignees' emails to the system	
	4	Staff send request to add planned task	
		EAMS returns to user that the task has been added successfully	
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Task is missing name or contribution rate	EAMS returns to student that they must enter name and contribution rate for tasks
	2	The sum of tasks contribution rates of the event being updated does not equal to 100	EAMS returns to student that the sum of contribution rates does not equal to 100
	3	Sub-task is missing name or contribution rate	EAMS returns to student that they must enter name and contribution rate for sub-tasks
	4	The sum of tasks contribution rates of the task being added does not equal to 100	EAMS returns to student that the sum of contribution rates of sub-tasks of a task does not equal to 100
Priority:	High		
Frequency of Use:	The more assignees, the more likely this feature is used frequently		
Business Rules:	<ul style="list-style-type: none"> <li>Students contribute to events by completing sub-tasks of event tasks. Event tasks, sub-tasks, assignees are optional to an event</li> <li>Each task belongs to an event, must include "name" and "contribution rate"</li> <li>Each sub-task belongs to a task, must include "name" and "contribution rate"</li> <li>Each sub-task can have multiple students as assignees</li> <li>Event tasks, sub-tasks, assignees are used to record and grade students' contribution to an event</li> <li>Event tasks must have a sum of contribution rates equals to 100%</li> <li>Sub-tasks of a task must have a sum of contribution rates equals to 100%</li> <li>Each event is created with a total contribution grade which is used to give to contributors of it. Event total contribution grade is divided to tasks based on its "contribution rate", and then later be divided to sub-tasks and assignees.</li> </ul>		

- A task's grade is divided to sub-tasks based on its "contribution rate"
- A sub-task's grade is equally divided among assignees, which becomes assignee grade
- Staff evaluate the student's performance in each sub-task by using completion rate. Each sub-task of each assignee has one completion rate. The contribution grade for one student for one sub-task is equals to its assignee grade multiplied by its completion rate
- The total contribution grade, contribution rates and completion rate of an event are decided by staffs when they give grades for an event. Student's proposed grade is only for helping staffs to make a judgement when approving the event

The student contribution grade calculation process can be illustrated in the following example

- An event has total contribution grade of 10.
- It has 2 tasks, each have its own sub-tasks, each sub-task has its own assignees:
  - Logistic Division: 40% contribution rate – 4 points
    - Transport equipment: 50% contribution rate – 2 points
      - 2 assignees: A and B – Each receives 1 point
    - Build the event's stage: 50% contribution rate – 2 points
      - 2 assignees: B and C – Each receives 1 point
  - Public Relations Division: 60% contribution rate – 6 points
    - Write Facebook posts: 100% contribution rate – 6 points
      - 1 assignee: A – receives 6 points

When the event finishes, each assignee receive grades for sub-tasks

- A
  - 1 point for "Transport equipment"
  - 6 points for "Write Facebook posts"
- B
  - 1 point for "Transport equipment"
  - 1 point for "Build the event's stage"
- C
  - 1 point for "Build the event's stage"

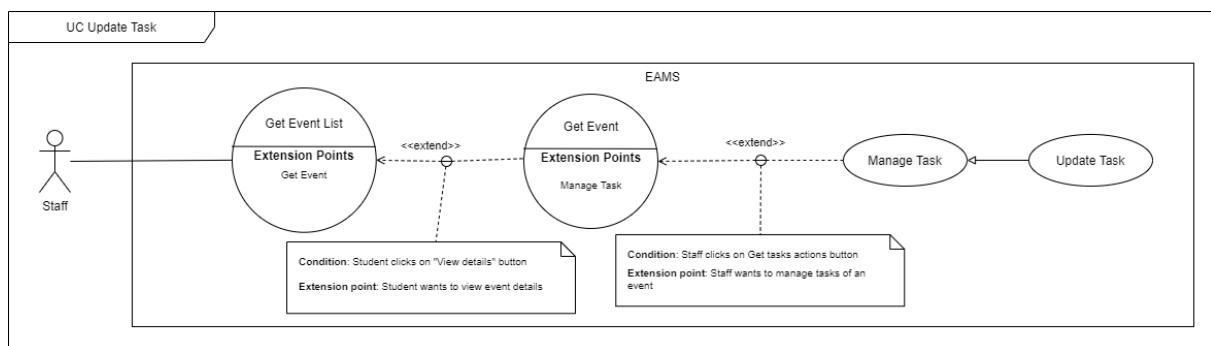
However, staff monitor each person's performance and give completion rate for each of the grade from the sub-task that they get:

- A
  - 1 point for "Transport equipment": Completion rate: 100%
  - 6 points for "Write Facebook posts": Completion rate: 50%
- B
  - 1 point for "Transport equipment": Completion rate: 20%
  - 1 point for "Build the event's stage": Completion rate: 30%

	<ul style="list-style-type: none"> <li>• C <ul style="list-style-type: none"> <li>○ 1 point for “Build the event’s stage”: Completion rate: 100% Finally, each student receives:</li> </ul> </li> <li>• A – Total contribution grade: 4 points <ul style="list-style-type: none"> <li>○ 1 point for “Transport equipment”: <math>(1 * 100\%)</math></li> <li>○ 3 points for “Write Facebook posts”: <math>(6 * 50\%)</math></li> </ul> </li> <li>• B – Total contribution grade: 0.5 point <ul style="list-style-type: none"> <li>○ 0.2 point for “Transport equipment”: <math>(1 * 20\%)</math></li> <li>○ 0.3 point for “Build the event’s stage”: <math>(1 * 30\%)</math></li> </ul> </li> <li>• C – Total contribution grade: 1 point <ul style="list-style-type: none"> <li>○ 1 point for “Build the event’s stage”: <math>(1 * 100\%)</math></li> </ul> </li> </ul> <p>Staff finishes event and everyone’s total contribution grade are saved to their histories.</p>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university has incentives for students to contribute in events. Events tasks are assigned fairly to students

**Table 45. <Use Case> Add Task**

#### 2.2.4.12 Update Task



**Figure 53. <Use Case> Update Task**

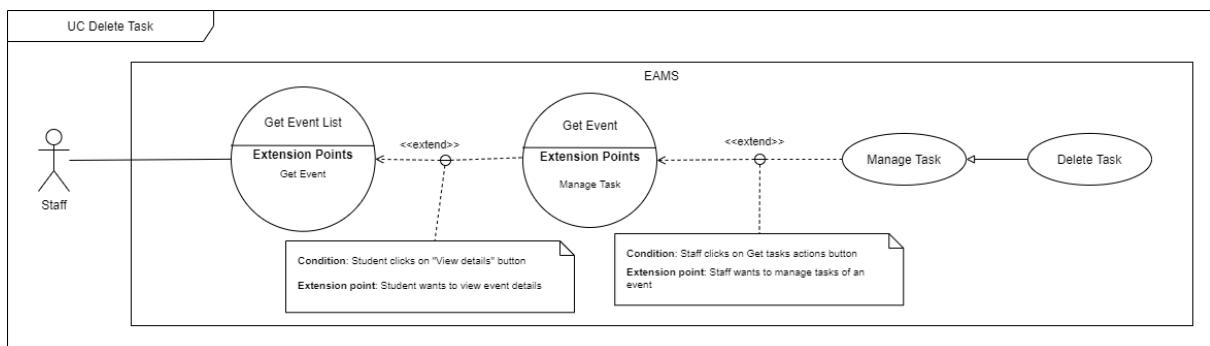
ID and Name:	<b>UC-46. Update Task</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This use case allow staff to update a task of an event		
Trigger:	Staff sends request to update a task of an event		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS update the task		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to update task by entering task name and contribution rate to the system	

		[Exception 1] [Exception 2]	
	2	Staff requests to update sub-task by entering sub-task name and contribution rate to the system [Exception 3] [Exception 4]	
	3	Staff requests to update assignee by entering assignees' emails to the system	
	4	Staff send request to add planned task	
			EAMS returns to user that the task has been added successfully
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Task is missing name or contribution rate	EAMS returns to student that they must enter name and contribution rate for tasks
	2	The sum of tasks contribution rates of the event being updated does not equal to 100	EAMS returns to student that the sum of contribution rates does not equal to 100
	3	Sub-task is missing name or contribution rate	EAMS returns to student that they must enter name and contribution rate for sub-tasks
	4	The sum of tasks contribution rates of the task being added does not equal to 100	EAMS returns to student that the sum of contribution rates of sub-tasks of a task does not equal to 100
Priority:	High		
Frequency of Use:	The more assignees, the more likely this feature is used frequently		
Business Rules:	<ul style="list-style-type: none"> <li>Students contribute to events by completing sub-tasks of event tasks. Event tasks, sub-tasks, assignees are optional to an event</li> <li>Each task belongs to an event, must include "name" and "contribution rate"</li> <li>Each sub-task belongs to a task, must include "name" and "contribution rate"</li> <li>Each sub-task can have multiple students as assignees</li> </ul>		

	<ul style="list-style-type: none"> <li>Event tasks, sub-tasks, assignees are used to record and grade students' contribution to an event</li> <li>Event tasks must have a sum of contribution rates equals to 100%</li> <li>Sub-tasks of a task must have a sum of contribution rates equals to 100%</li> <li>Each event is created with a total contribution grade which is used to give to contributors of it. Event total contribution grade is divided to tasks based on its "contribution rate", and then later be divided to sub-tasks and assignees.</li> <li>A task's grade is divided to sub-tasks based on its "contribution rate"</li> <li>A sub-task's grade is equally divided among assignees, which becomes assignee grade</li> <li>Staff evaluate the student's performance in each sub-task by using completion rate. Each sub-task of each assignee has one completion rate. The contribution grade for one student for one sub-task is equals to its assignee grade multiplied by its completion rate</li> <li>The total contribution grade, contribution rates and completion rate of an event are decided by staffs when they give grades for an event. Student's proposed grade is only for helping staffs to make a judgement when approving the event</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university has incentives for students to contribute in events. Events tasks are assigned fairly to students

**Table 46. <Use Case> Delete Task**

#### 2.2.4.13 Delete Task



**Figure 54. <Use Case> Delete Task**

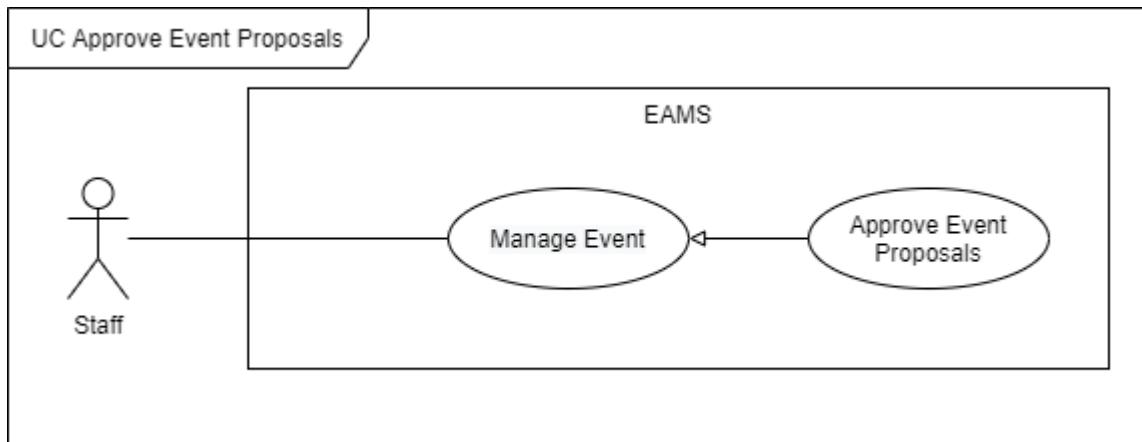
ID and Name:	<b>UC-47. Delete Task</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A

Description:	This use case allow staff to delete a task from an event		
Trigger:	Staff sends request to delete a from an event		
Preconditions:	PRE-1. User has logged in the system as Staff		
Post-conditions:	POST-1. EAMS delete the task along with its sub-task and assignees		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends a request to delete the task [Exception 1] [Exception 2]	
			EAMS deletes the task from the event
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	High		
Frequency of Use:	The more assignees, the more likely this feature is used frequently		
Business Rules:	<ul style="list-style-type: none"> <li>Students contribute to events by completing sub-tasks of event tasks. Event tasks, sub-tasks, assignees are optional to an event</li> <li>Each task belongs to an event, must include “name” and “contribution rate”</li> <li>Each sub-task belongs to a task, must include “name” and “contribution rate”</li> <li>Each sub-task can have multiple students as assignees</li> <li>Event tasks, sub-tasks, assignees are used to record and grade students’ contribution to an event</li> <li>Event tasks must have a sum of contribution rates equals to 100%</li> <li>Sub-tasks of a task must have a sum of contribution rates equals to 100%</li> <li>Each event is created with a total contribution grade which is used to give to contributors of it. Event total contribution grade is divided to tasks based on its “contribution rate”, and then later be divided to sub-tasks and assignees.</li> <li>A task’s grade is divided to sub-tasks based on its “contribution rate”</li> <li>A sub-task’s grade is equally divided among assignees, which becomes assignee grade</li> <li>Staff evaluate the student’s performance in each sub-task by using completion rate. Each sub-task of each assignee has one completion rate. The contribution grade for one student for one sub-task is equals to its assignee grade multiplied by its completion rate</li> <li>The total contribution grade, contribution rates and completion rate of an event are decided by staffs when they give grades for</li> </ul>		

	an event. Student's proposed grade is only for helping staffs to make a judgement when approving the event
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	The university has incentives for students to contribute in events. Events tasks are assigned fairly to students

**Table 47. <Use Case> Delete Task**

#### 2.2.4.14 Approve Event Proposals



**Figure 55. <Use Case> Approve Event Proposals**

ID and Name:	<b>UC-48. Approve Event Proposals</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Allow staff to approve students' event proposals		
Trigger:	Staff send request to approve student's event		
Preconditions:	PRE-1. User has logged in the system as Staff PRE-2. The event status is "Pending"		
Post-conditions:	POST-1. EAMS changes event status to "Approved"		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request to approve event [Exception 1]	
			EAMS Returns to user that event has been approved and change the event status to "Unpublished"
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response

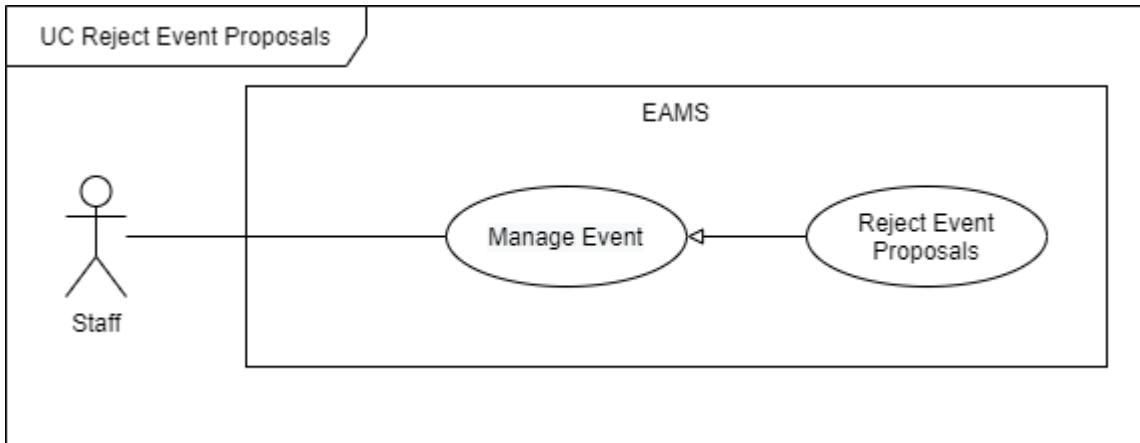
	1	Event can no longer be approved	EAMS returns to staff that they can only approve pending events
Priority:	Medium		
Frequency of Use:	The more active students are, the more likely this feature is used		
Business Rules:	<ul style="list-style-type: none"> <li>When an event proposal gets created, EAMS stores it as an event with a “Draft” status. Student can then submit event proposal to staff, which turns it to “Pending” status. Staff can then reject the proposal, changing it to “Rejected”, give it some feedback for them to submit it again until it gets approved. If a staff decides to approve the event, its status changes to “Unpublished”, they will take over and manage it as if they created that event.</li> <li>Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>Staff manages all events in the university, excluding student’s draft events</li> <li>An event requires the following fields: <ul style="list-style-type: none"> <li>○ Title: between 1 – 63 characters</li> <li>○ Location: between 1 – 63 characters</li> <li>○ Semester: current semester or the semester that will begin in the next 30 days</li> <li>○ Start and end date time: End date time comes after start date time</li> <li>○ Start register date time: Less than end date time</li> <li>○ End register date time: between start register date time - end date time</li> <li>○ Start check-in date time: between start register date time - end date time</li> <li>○ End check-in date time: between start check-in date time – end date time</li> <li>○ Whether or not event has registration or check-in restrictions</li> <li>○ Short description: between 1 – 127 characters</li> <li>○ Description: between 1 – 4095 characters</li> <li>○ Image: URL that’s less than 1023 characters</li> <li>○ At least one and maximum of 10 feedback questions</li> <li>○ Start feedback date time: between start date time – end date time</li> <li>○ One grade criteria of a policy document that is being applied in the chosen semester</li> <li>○ One grade sub-criteria that belongs to the chosen criteria</li> <li>○ Total contribution grade: positive, less than 32767</li> </ul> </li> </ul>		

	<p>An event also has the following optional fields:</p> <ul style="list-style-type: none"> <li>○ Note: Additional info for students</li> <li>○ Checkout requirement: determines if event requires students to checkout</li> <li>○ If event requires checking out, start and end checkout date time is required</li> <li>○ Start checkout date time: between start check-in date time – end date time</li> <li>○ End checkout date time: comes after start checkout date time</li> <li>○ If event requires checking out, user can choose to require students to feedback before checking out</li> <li>○ Event tasks, subtasks and assignees for subtasks: for students to contribute to events</li> <li>● After creation, the event semester cannot be changed</li> <li>● An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>○ When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> </ul> </li> <li>● Staff can make updates to an event as long as its status is “Unpublished” or “Published”</li> <li>● After an event finishes, staff can start looking up student information, grant achievements for students that participate in or contributes to it</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature

Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools' event organization departments use this feature frequently
--------------	--

**Table 48. <Use Case> Approve Event Proposals**

2.2.4.15 Reject Event Proposals



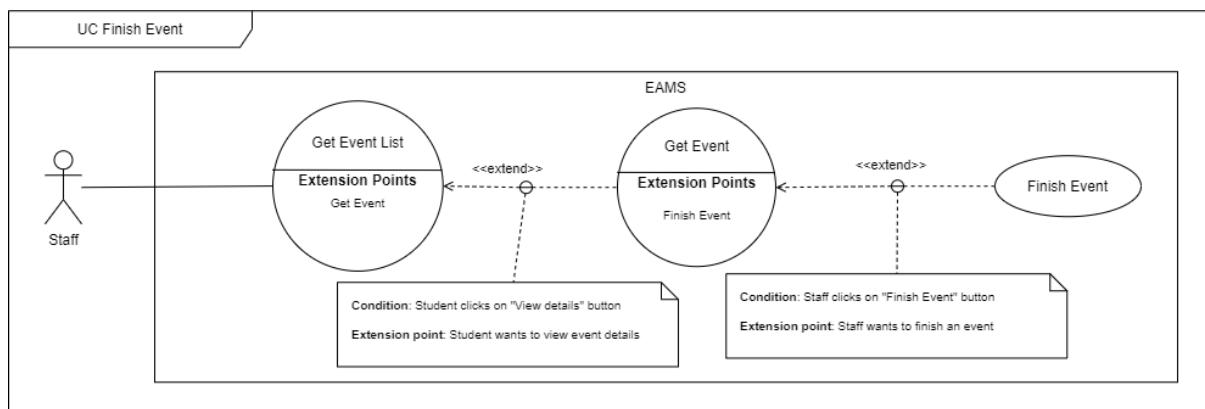
**Figure 56. <Use Case> Reject Event Proposals**

ID and Name:	<b>UC-49. Reject Event Proposals</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Allow staff to reject students' event proposals		
Trigger:	Staff send request to reject student's event		
Preconditions:	PRE-1. User has logged in the system as Staff PRE-2. The event status is "Pending"		
Post-conditions:	POST-1. EAMS changes event status to "Rejected"		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request to reject event [Exception 1]	
			EAMS Returns to user that event has been rejected and change the event status to "Rejected"
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Event can no longer be rejected	EAMS returns to staff that they can only reject pending events
Priority:	Medium		

Frequency of Use:	The more active students are, the more likely this feature is used
Business Rules:	<ul style="list-style-type: none"> <li>When an event proposal gets created, EAMS stores it as an event with a “Draft” status. Student can then submit event proposal to staff, which turns it to “Pending” status. Staff can then reject the proposal, changing it to “Rejected”, give it some feedback for them to submit it again until it gets approved. If a staff decides to approve the event, its status changes to “Unpublished”, they will take over and manage it as if they created that event.</li> <li>Staff manages all events in the university, excluding student’s draft events</li> <li>An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>When created by a staff or submitted by a student and approved, its default status is “Unpublished”. This is when staff review event information before publishing the event for students.</li> <li>When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> <li>When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>When event starts allowing students to feedback, staff monitors feedback results</li> <li>Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> </ul> </li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	Most students rarely propose events. Only university club presidents or students that take part in the schools’ event organization departments use this feature frequently

**Table 49.      <Use Case> Reject Event Proposals**

#### 2.2.4.16 Finish Event



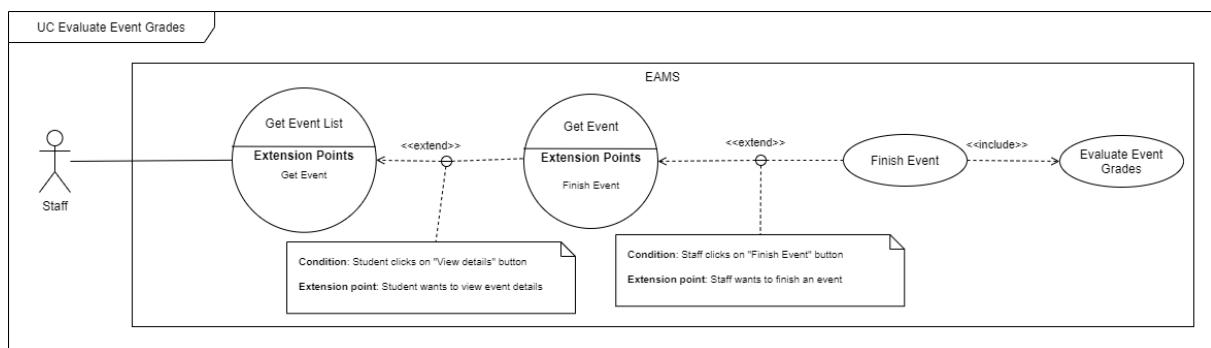
**Figure 57. <Use Case> Finish Event**

ID and Name:	<b>UC-50. Finish Event</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	When an event is over, staff has to finish event in order to give students grades histories		
Trigger:	Staff send request to finish an event		
Preconditions:	PRE-1. User has logged in the system as Staff PRE-2. The event has ended PRE-3. The event status is "Published"		
Post-conditions:	POST-1. EAMS changes event status to "Finished" POST-2. EAMS stores students grade histories		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request to finish event	
			EAMS Returns to user that event has been rejected and change the event status to "Rejected"
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	High		
Frequency of Use:	The more active students are, the more likely this feature is used		
Business Rules:	<ul style="list-style-type: none"> <li>An event managed by staff goes through the following process:           <ul style="list-style-type: none"> <li>When created by a staff or submitted by a student and approved, its default status is "Unpublished". This is when staff review event information before publishing the event for students.</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes “Published”. This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> <li>○ When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>○ When event starts allowing students to feedback, staff monitors feedback results</li> <li>○ Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>○ When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> <li>● When a staff finishes an event, its status changes to “Finished”. All the qualified participants and contributors get graded based on the criteria, sub-criteria and the tasks details of that event.</li> <li>● After an event finishes, staff can start looking up student information, grant achievements for students that participate in or contributes to it</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	N/A

**Table 50. <Use Case> Finish Event**

#### 2.2.4.17 Evaluate Event Grades



**Figure 58. <Use Case> Evaluate Event Grades**

ID and Name:	<b>UC-51. Evaluate Event Grades</b>		
Created By:	BinhPD	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A

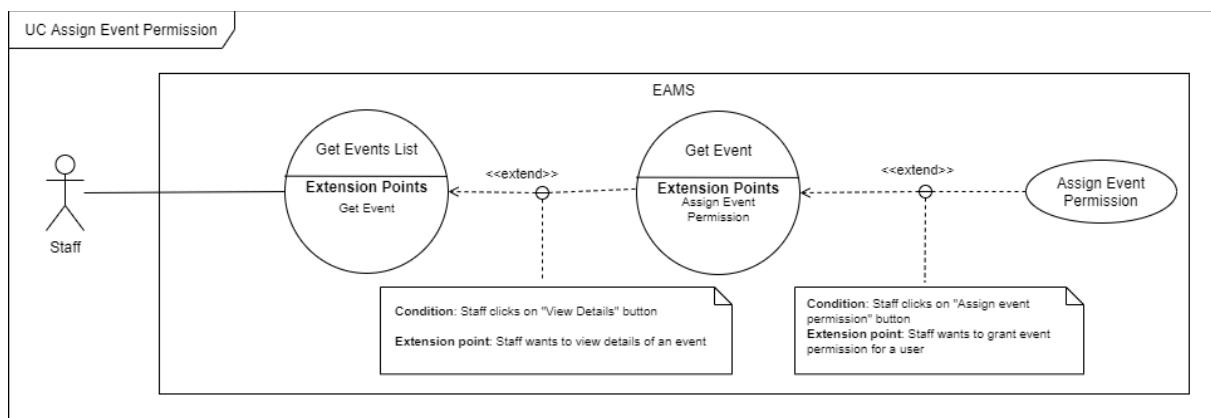
Description:	When an event is over, staff has to finish event in order to give students grades histories. During this process, staff has to evaluate students' participations and contributions		
Trigger:	Staff send request to evaluate participants and contributors		
Preconditions:	PRE-1. User has logged in the system as Staff PRE-2. The event has ended PRE-3. The event status is "Published"		
Post-conditions:	POST-1. EAMS stores disqualified participants POST-2. EAMS stores assignees' completion rates		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request to evaluate participants and contributors	
			EAMS stores participants and contributors info
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	High		
Frequency of Use:	The more active students are, the more likely this feature is used		
Business Rules:	<ul style="list-style-type: none"> <li>Events can also be created by students and then later be approved by a staff. After an event gets approved, students can only update event posts and tasks. Staff takes over and can manage everything about that event.</li> <li>An event managed by staff goes through the following process: <ul style="list-style-type: none"> <li>When created by a staff or submitted by a student and approved, its default status is "Unpublished". This is when staff review event information before publishing the event for students.</li> <li>When staff finish finalizing the event information, they publish the event for student to view and register to participate, the event status becomes "Published". This is when staffs are recommended to assign assistants to help scan QR to check-in or check-out for the event</li> <li>When the start date of the event comes and the event starts to allow checking in, staff monitor participations, check-in or check-out manually if necessary</li> <li>When event starts allowing students to feedback, staff monitors feedback results</li> <li>Afterward, if the event requires checking out, staff monitor participations and checkout manually if necessary</li> <li>When the event ends, staff can start finishing stage, giving grades to participants and contributors.</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ After an event finishes, its status changes to “Finished” and staff can no longer make any updates to the event. They can only get reports of it.</li> <li>● After an event finishes, staff can start looking up student information, grant achievements for students that participate in or contributes to it</li> <li>● The total contribution grade, contribution rates and completion rate of an event are decided by staffs when they give grades for an event. Student’s proposed grade is only for helping staffs to make a judgement when approving the event</li> <li>● When grading events, staff can disqualify any participants as long as they provide a reason for it</li> <li>● Student can contribute to multiple sub-tasks in multiple tasks in one event. Each sub-task gives student one contribution grade. This means that student can have multiple contribution grades in one event. However, for every event, only one final contribution grade is saved in the student’s grade history. This grade is equal to the sum of all the contribution grade that they have for that event</li> </ul> <p>The grades evaluation process can be illustrated in the following example:</p> <ul style="list-style-type: none"> <li>● An event has total contribution grade of 10.</li> <li>● It has 2 tasks, each have its own sub-tasks, each sub-task has its own assignees:       <ul style="list-style-type: none"> <li>○ Logistic Division: 40% contribution rate – 4 points           <ul style="list-style-type: none"> <li>■ Transport equipment: 50% contribution rate – 2 points               <ul style="list-style-type: none"> <li>● 2 assignees: A and B – Each receives 1 point</li> </ul> </li> <li>■ Build the event’s stage: 50% contribution rate – 2 points               <ul style="list-style-type: none"> <li>● 2 assignees: B and C – Each receives 1 point</li> </ul> </li> </ul> </li> <li>○ Public Relations Division: 60% contribution rate – 6 points           <ul style="list-style-type: none"> <li>■ Write Facebook posts: 100% contribution rate – 6 points               <ul style="list-style-type: none"> <li>● 1 assignee: A – receives 6 points</li> </ul> </li> </ul> </li> </ul> </li> <li>● When the event finishes, each assignee receive grades for sub-tasks       <ul style="list-style-type: none"> <li>○ A           <ul style="list-style-type: none"> <li>■ 1 point for “Transport equipment”</li> <li>■ 6 points for “Write Facebook posts”</li> </ul> </li> <li>○ B           <ul style="list-style-type: none"> <li>■ 1 point for “Transport equipment”</li> <li>■ 1 point for “Build the event’s stage”</li> </ul> </li> <li>○ C           <ul style="list-style-type: none"> <li>■ 1 point for “Build the event’s stage”</li> </ul> </li> </ul> </li> <li>● However, staff monitor each person’s performance and give completion rate for each of the grade from the sub-task that they get:</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>○ A <ul style="list-style-type: none"> <li>▪ 1 point for “Transport equipment”: Completion rate: 100%</li> <li>▪ 6 points for “Write Facebook posts”: Completion rate: 50%</li> </ul> </li> <li>○ B <ul style="list-style-type: none"> <li>▪ 1 point for “Transport equipment”: Completion rate: 20%</li> <li>▪ 1 point for “Build the event’s stage”: Completion rate: 30%</li> </ul> </li> <li>○ C <ul style="list-style-type: none"> <li>▪ 1 point for “Build the event’s stage”: Completion rate: 100%</li> </ul> </li> <li>• Finally, each student receives: <ul style="list-style-type: none"> <li>○ A – Total contribution grade: 4 points <ul style="list-style-type: none"> <li>▪ 1 point for “Transport equipment”: <math>(1 * 100\%)</math></li> <li>▪ 3 points for “Write Facebook posts”: <math>(6 * 50\%)</math></li> </ul> </li> <li>○ B – Total contribution grade: 0.5 point <ul style="list-style-type: none"> <li>▪ 0.2 point for “Transport equipment”: <math>(1 * 20\%)</math></li> <li>▪ 0.3 point for “Build the event’s stage”: <math>(1 * 30\%)</math></li> </ul> </li> <li>○ C – Total contribution grade: 1 point <ul style="list-style-type: none"> <li>▪ 1 point for “Build the event’s stage”: <math>(1 * 100\%)</math></li> </ul> </li> </ul> </li> <li>• Staff finishes event and everyone’s total contribution grade are saved to their histories.</li> </ul>
Other Information:	In case of internet connection failure, staffs cannot use this feature
Assumptions:	Staff evaluate events fairly

**Table 51. <Use Case> Evaluate Event Grades**

#### 2.2.4.18 Assign Event Permission



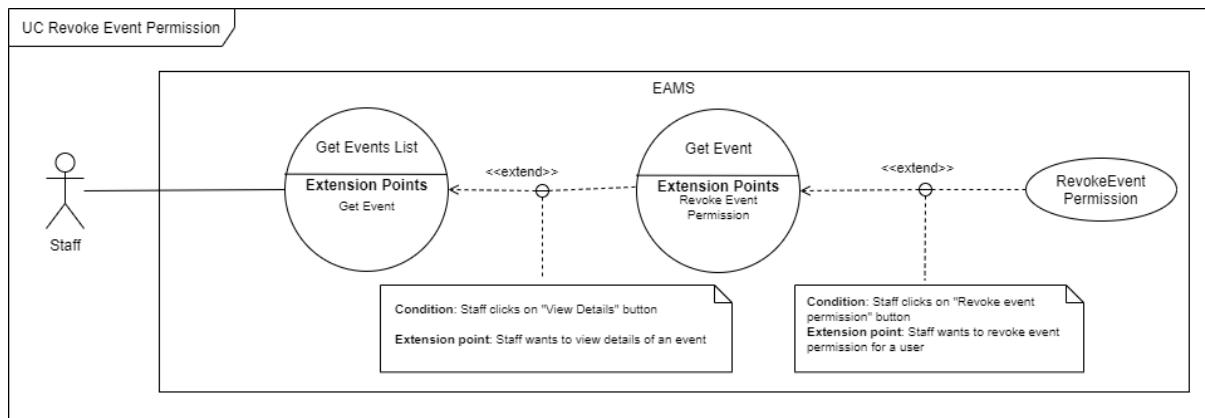
**Figure 59. <Use Case> Assign Event Permission**

ID and Name:	UC-52. Assign Event Permission		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A

Description:	Staff assigns permission to an event for student to let student be event assistant		
Trigger:	Staff sends request to assign permission to a student		
Preconditions:	PRE-1. User has logged in as Staff role		
Post-conditions:	POST-1. EAMS stores event permission for this student		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to assign permission for a student [Exception 1]	
			EAMS returns that event permission has been assigned successfully
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Student has been assigned permission already	EAMS returns that this student had permission, cannot reassign old permission
Priority:	High		
Frequency of Use:	Every time an event is organized, 5 usage per event		
Business Rules:	<ul style="list-style-type: none"> <li>Event permission must be defined in EAMS. Currently, there is one permission “check-in event” in the system. This permission allows any user who has it to scan QR to check-in or check-out for an event. However, permissions are resource based, meaning that a user has permission to check-in for event A may not be able to check-in for event B</li> <li>Every event needs at least one user as an assistant to check-in or check-out for students. Assistants scan QR to check-in or check-out for students.</li> <li>Staff can assign Scan-QR permission for any users for them to become assistants. This should be done before event’s start check-in date time</li> <li>After assigning event permission, assistant has to logout and login again to apply new permission</li> </ul>		
Other Information:	<ul style="list-style-type: none"> <li>In case of network failures, staff cannot assign event permission for student</li> </ul>		
Assumptions:	Every event has at least one student as assistant		

**Table 52. <Use Case> Assign Event Permission**

#### 2.2.4.19 Revoke Event Permission



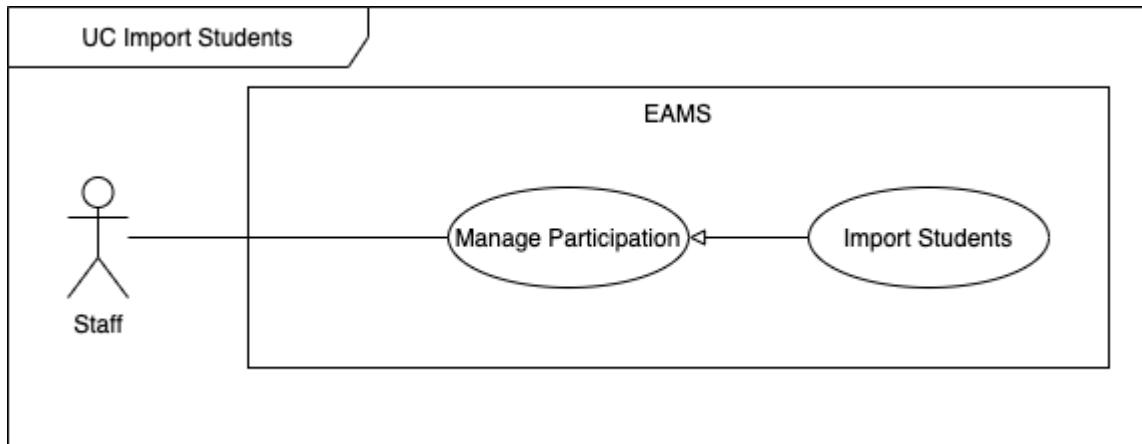
**Figure 60. <Use Case> Revoke Event Permission**

ID and Name:	<b>UC-53. Revoke Event Permission</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff revokes permission to an event of student to let student not to be event assistant anymore		
Trigger:	Staff sends request to delete permission of a student		
Preconditions:	PRE-1. User has logged in as Staff role PRE-2. Student had permission before		
Post-conditions:	POST-1. EAMS deletes event permission of this student		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to revokes permission of a student [Exception 1]	
Alternative Flows:			EAMS returns that event permission has been revoked successfully
Exceptions:	No	Cause	System Response
	1	Student does not have this permission	EAMS returns that this student does not have this permission
Priority:	Medium		
Frequency of Use:	Every time an event is organized		
Business Rules:	<ul style="list-style-type: none"> <li>Every event needs at least one user as an assistant to check-in or check-out for students. Assistants scan QR to check-in or check-out for students. Staff can assign Scan-QR permission for any users for them to become assistants. This should be done before event's start check-in date time</li> </ul>		

	<ul style="list-style-type: none"> <li>Revoking event permission makes student cannot scan that event's QR for others anymore</li> </ul>
Other Information:	In case of network failures, staff cannot revoke event permission of student
Assumptions:	Every event has at least one student as assistant

**Table 53. <Use Case> Revoke Event Permission**

#### 2.2.4.20 Import Students



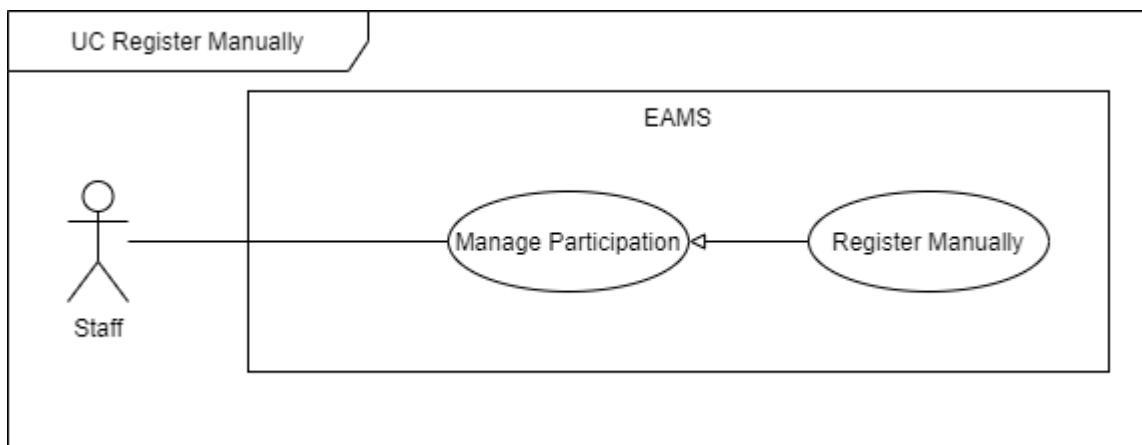
**Figure 61. <Use Case> Import Students**

ID and Name:	<b>UC-54. Import Students</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This feature allows staff to import list of students		
Trigger:	Staff sends a request to import students list		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role PRE-2. Event is published		
Post-conditions:	POST-1. EAMS saves these students		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends a request along with an excel files that contains students.	
			EAMS returns a message that students have been imported successfully. [Exception 1] [Exception 2]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response

	1	There are some students that does not exist in EAMS	EAMS returns an excel file that contains list of students with fail reason
	2	Imported file uses wrong file	EAMS informs that you have used wrong template
Priority:	Medium		
Frequency of Use:	Approximately 1 usage per event		
Business Rules:	<ul style="list-style-type: none"> <li>There are some cases that we need to add a list of students into system for some specific purposes such as import registrants, import whitelist registrants and import whitelist participants of an event.</li> <li>Staffs have to use provided template to import students into system.</li> </ul>		
Other Information:	In case of internet connection failure, staff cannot import students.		
Assumptions:	Staffs have Microsoft Office on their computer		

**Table 54. <Use Case> Import Students**

#### 2.2.4.21 Register Manually



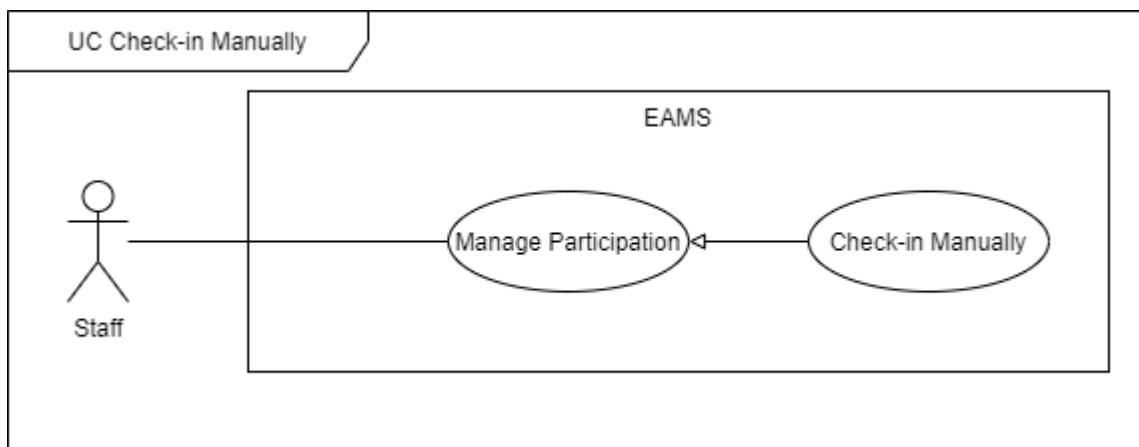
**Figure 62. <Use Case> Register Manually**

ID and Name:	<b>UC-55. Register Manually</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff registers event for student in case student cannot register by themselves		
Trigger:	Staff sends request to register event for student		
Preconditions:	PRE-1. User has logged in as Staff role PRE-2. Event must be published		
Post-conditions:	POST-1. EAMS stores current time as student register date time		
Normal Flow:	Step	Actor Action	System Response

	1	Staff requests to register event for student [Exception 1]	
		EAMS returns that register successfully for student	
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Current date time is not between start and end register time	EAMS returns that registration is not right time
Priority:	High		
Frequency of Use:	Every time an event happens, 10 times per event		
Business Rules:	<ul style="list-style-type: none"> <li>Staff can register for a student if and only if current time is between start and end register date time</li> <li>When staffs register manually or check-in manually for student: <ul style="list-style-type: none"> <li>Check event restricts register or not, if not, record current time as student's register date time</li> <li>If event restricts register, firstly add student to event's registrant whitelist then record current time as student's register date time</li> </ul> </li> <li>Register manually requires reason also, which indicates why staff has to register manually for student, for example: forget mobile phone, forget account password, out of battery</li> </ul>		
Other Information:	In case of network failures, staff cannot finish this use case properly		
Assumptions:	N/A		

**Table 55. <Use Case> Register Manually**

#### 2.2.4.22 Check-in Manually



**Figure 63. <Use Case> Check-in Manually**

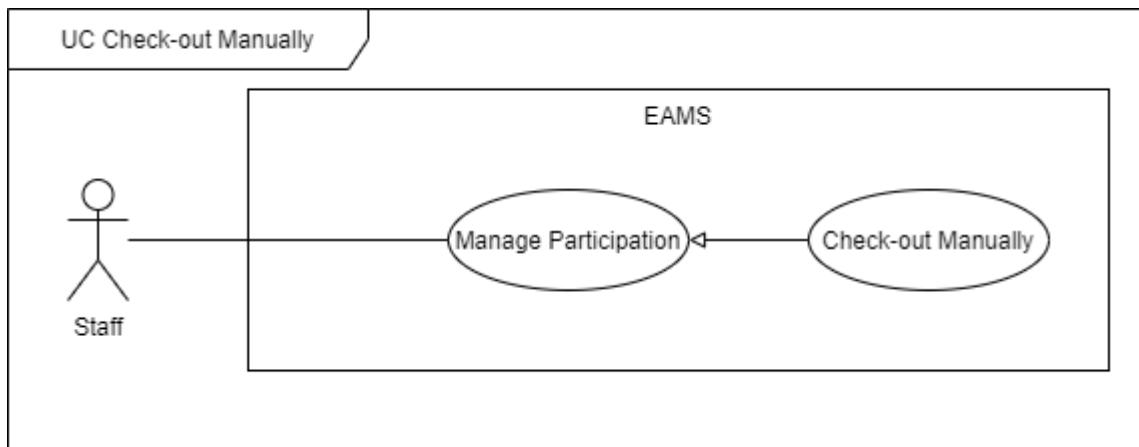
ID and Name:	UC-56.	Check-in Manually
--------------	--------	-------------------

Created By:	ChienTM		Date Created:	02/12/20		
Primary Actor:	Staff		Secondary Actors:	N/A		
Description:	Staff check-in event for student in case student cannot check-in by themselves					
Trigger:	Staff sends request to check-in event for student					
Preconditions:	PRE-1. User has logged in as Staff role PRE-2. Event must be published					
Post-conditions:	POST-1. EAMS stores current time as student check-in date time					
Normal Flow:	Step	Actor Action	System Response			
	1	Staff requests to check-in event for student [Exception 1]				
			EAMS returns that check-in successfully for student [Alternative 1]			
Alternative Flows:	Alternative 1					
	Step	Actor Action	System Response			
	1		EAMS check registration then return this student has not registered			
	2	Staff requests to register event for student then check-in [Exception 2]				
			EAMS returns that register and check-in successfully for student			
Exceptions:	No	Cause	System Response			
	1	Current date time is not between start and end check-in time	EAMS returns that checking-in is not right time			
	2	Current date time is not between start and end register time	EAMS returns that registration is not right time			
Priority:	High					
Frequency of Use:	Every time an event happen, 10 times per event					
Business Rules:	<ul style="list-style-type: none"> <li>Staff can check-in for a student if and only if current time is between start and end check-in date time</li> <li>When staff check-in manually for student: <ul style="list-style-type: none"> <li>If student has not registered, register for them</li> <li>Check event restricts check-in or not, if not, record current time as student's check-in date time</li> </ul> </li> </ul>					

	<ul style="list-style-type: none"> <li>○ If event restrict check-in, firstly add student to check-in whitelist then record current time as student's check-in date time</li> <li>● Check-in manually requires reason also, which indicates why staff has to check-in manually for student, for example: forget mobile phone, forget account password, out of battery</li> </ul>
Other Information:	In case of network failures, staff cannot finish this use case properly
Assumptions:	N/A

**Table 56. <Use Case> Check-in Manually**

#### 2.2.4.23 Check-out Manually



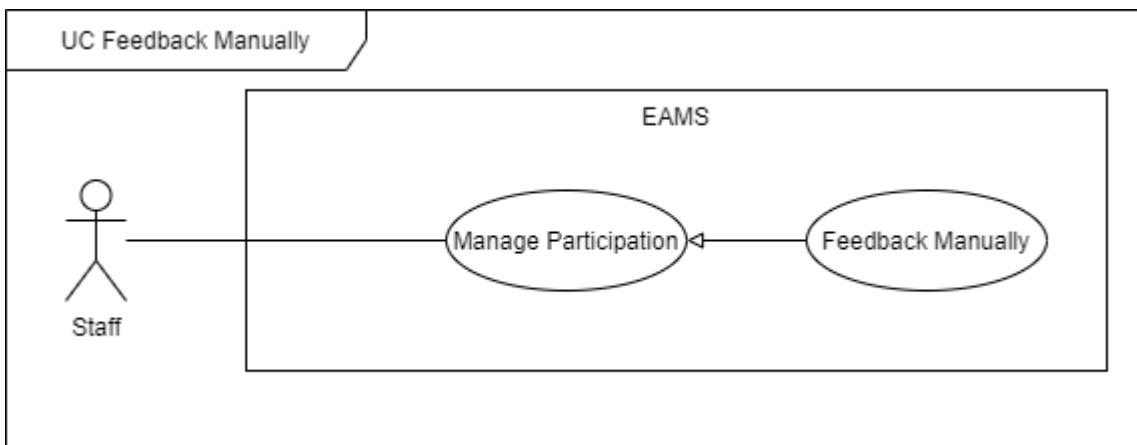
**Figure 64. <Use Case> Check-out Manually**

ID and Name:	<b>UC-57. Check-out Manually</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff check-out event for student in case student cannot check-out by themselves		
Trigger:	Staff sends request to check-out event for student		
Preconditions:	PRE-1. User has logged in as Staff role PRE-2. Event must be published		
Post-conditions:	POST-1. EAMS stores current time as student check-out date time		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to check-out event for student [Exception 1]	
			EAMS returns that check-out successfully for student
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response

	1	Current date time is not between start and end check-out time	EAMS returns that checking-out is not right time
Priority:	High		
Frequency of Use:	Every time an event happen, 10 times per event		
Business Rules:	<ul style="list-style-type: none"> <li>To be able to check-out, some requirements need to be fulfilled:           <ul style="list-style-type: none"> <li>Student must register before</li> <li>Student must check-in before</li> <li>Event must require check-out</li> <li>Current time is between start check-out date time and end check-out date time</li> <li>If event requires feedback before check-out, student must give feedback before.</li> <li>Check-out manually requires reason also, which indicates why staff has to check-out manually for student, for example: out of battery</li> </ul> </li> </ul>		
Other Information:	In case of network failures, staff cannot finish this use case properly		
Assumptions:	N/A		

**Table 57. <Use Case> Check-out Manually**

#### 2.2.4.24 Feedback Manually



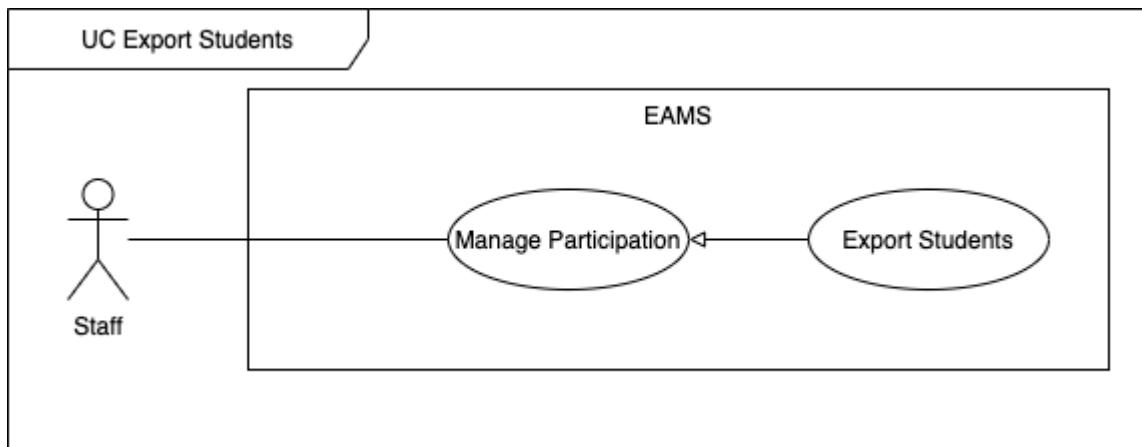
**Figure 65. <Use Case> Feedback Manually**

ID and Name:	<b>UC-58. Feedback Manually</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff record student's feedback for event from a hard copy to EAMS		
Trigger:	Staff sends request to record student feedback		
Preconditions:	PRE-1. User has logged in as Staff role PRE-2. Event must be published		

	PRE-3. Student has registered event before PRE-4. Student has not done feedback before		
Post-conditions:	POST-1. EAMS stores current time as student feedback date time POST-2. EAMS stores student feedback answers		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to feedback event for student	
			EAMS requires staff fills answer for all feedback questions [Alternative 1]
	2	Staff fills all feedback questions then sends request to finish	EAMS return that staff has done feedback manually successfully [Exception 1]
Alternative Flows:	Alternative 1		
	Step	Actor Action	System Response
	1		EAMS require staff updates answers for existing feedback question's answers
	2	Staff updates old feedback answers then send request to finish [Return to step 2 in normal flow]	
Exceptions:	No	Cause	System Response
	1	Answers is invalidated	EAMS returns that feedback answers are not valid
Priority:	Medium		
Frequency of Use:	Every time an event happen, 10 times per event		
Business Rules:	<ul style="list-style-type: none"> <li>Staff can give feedback manually for student who has not given feedback before</li> <li>Staff can view and update feedback for students that finished feedback manually</li> <li>When giving feedback, users have to fill answer for all feedback questions. There are 2 types of feedback question: <ul style="list-style-type: none"> <li>Text: user can freely type, length between 1 - 255</li> <li>Rating: user have to input from 1 to 5</li> </ul> </li> </ul>		
Other Information:	In case of network failures, staff cannot finish this use case properly		
Assumptions:	N/A		

**Table 58. <Use Case> Feedback Manually**

#### 2.2.4.25 Export Students

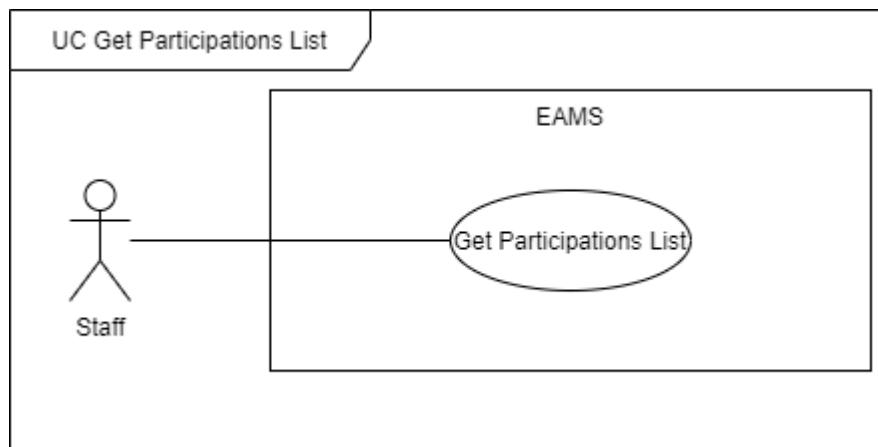


**Figure 66. <Use Case> Export Students**

ID and Name:	<b>UC-59. Export Students</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This feature allows staffs export a list of students of that event		
Trigger:	Staff sends requests to download students list		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role		
Post-conditions:	POST-1. EAMS returns list of registrants list		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request with a specific event to get students list of that events [Exception 1]	
			EAMS returns an excel file that contains list of students
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Event does not exist	EAMS informs that event does not exist
Priority:	Medium		
Frequency of Use:	Approximately 1 usage per event		
Business Rules:	<ul style="list-style-type: none"> <li>Staffs are provided some download reports feature such as registrants report, participation reports and feedback reports to support staff getting an overview of the current event.</li> </ul>		
Other Information:	In case of internet connection failure, staff cannot download students list		
Assumptions:	Staffs have Microsoft Office on their computer		

**Table 59. <Use Case> Export Students**

2.2.4.26 Get Participations List



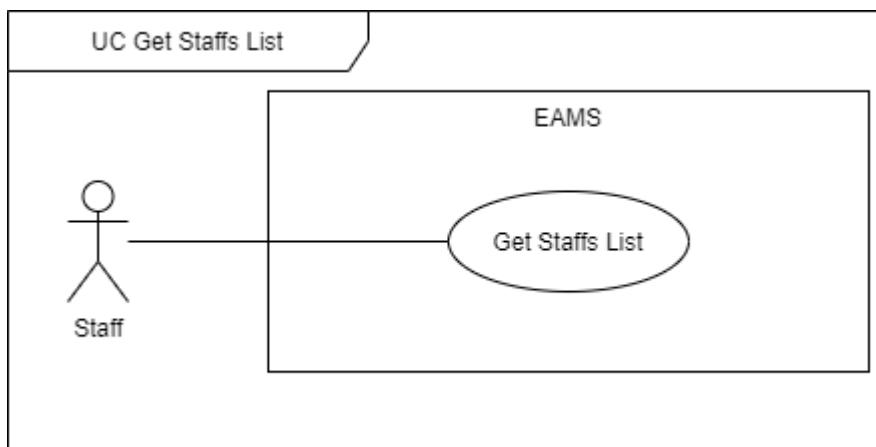
**Figure 67. <Use Case> Get Participations List**

ID and Name:	<b>UC-60. Get Participations List</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff gets a list of event's participants and relevant information		
Trigger:	Staff sends request to get event participations list		
Preconditions:	PRE-1. User has logged in as Staff role PRE-2. Event status is "Published"		
Post-conditions:	POST-1. EAMS returns a list of event participations		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to get event participations list [Exception 1]	
			EAMS returns list of event's participations
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Event does not have any participation	EAMS returns that there is no participation
Priority:	High		
Frequency of Use:	Every time an event happen		
Business Rules:	<ul style="list-style-type: none"> <li>Event Participations List: monitor any student who is in whitelist, or has registered event</li> </ul>		
Other Information:	<ul style="list-style-type: none"> <li>In case of network failures, staff cannot finish this use case properly</li> </ul>		

	<ul style="list-style-type: none"> <li>This list include all student in whitelist or student has registered event</li> </ul>
Assumptions:	N/A

**Table 60.** *<Use Case> Get Participations List*

#### 2.2.4.27 Get Staffs List

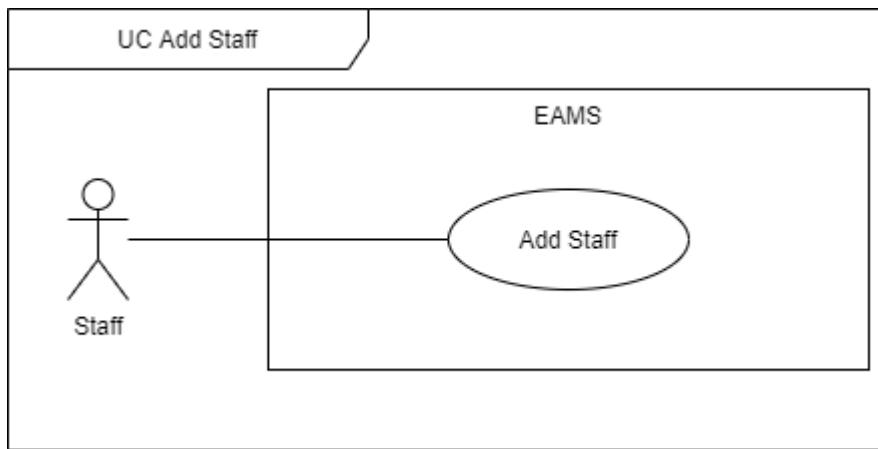


**Figure 68.** *<Use Case> Get Staffs List*

ID and Name:	<b>UC-61. Get Staffs List</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff gets a list of staffs in the system		
Trigger:	Staff sends request to get staffs list		
Preconditions:	PRE-1. User has logged in as Staff role		
Post-conditions:	POST-1. EAMS returns a list of system's staffs user		
Normal Flow:	Step	Actor Action	System Response
	1	Staff request to get staffs list	EAMS returns list of staffs
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ul style="list-style-type: none"> <li>Staff list is list of users who have authorization to use staff application</li> </ul>		
Other Information:	<ul style="list-style-type: none"> <li>In case of network failures, staff cannot finish this use case properly</li> </ul>		
Assumptions:	N/A		

**Table 61.** *<Use Case> Get Staffs List*

#### 2.2.4.28 Add Staff



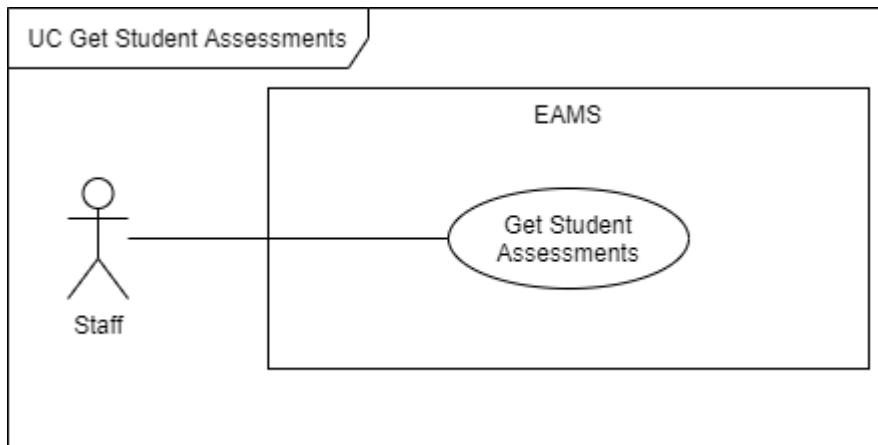
**Figure 69. <Use Case> Add Staff**

ID and Name:	<b>UC-62. Add Staff</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff creates new staff in system by using organization email address		
Trigger:	Staff sends request to create new staff		
Preconditions:	PRE-1. User has logged in as Staff role		
Post-conditions:	POST-1. EAMS returns that create successfully		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to create new staff	
			EAMS requires staff to fill new staff email address and name
	2	Staff fills required fields and finish creating [Exception 1]	
			EAMS returns that create new staff successfully
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Data is invalidated	EAMS returns that create failed
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ul style="list-style-type: none"> <li>Currently, staff can add a new staff to EAMS</li> <li>Validation rule of creating new user:           <ul style="list-style-type: none"> <li>Email is a valid email, under 63 characters</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ Name is required, under 127 characters</li> </ul>
Other Information:	In case of network failures, staff cannot finish this use case properly
Assumptions:	N/A

**Table 62. <Use Case> Add Staff**

#### 2.2.4.29 Get Student Assessment



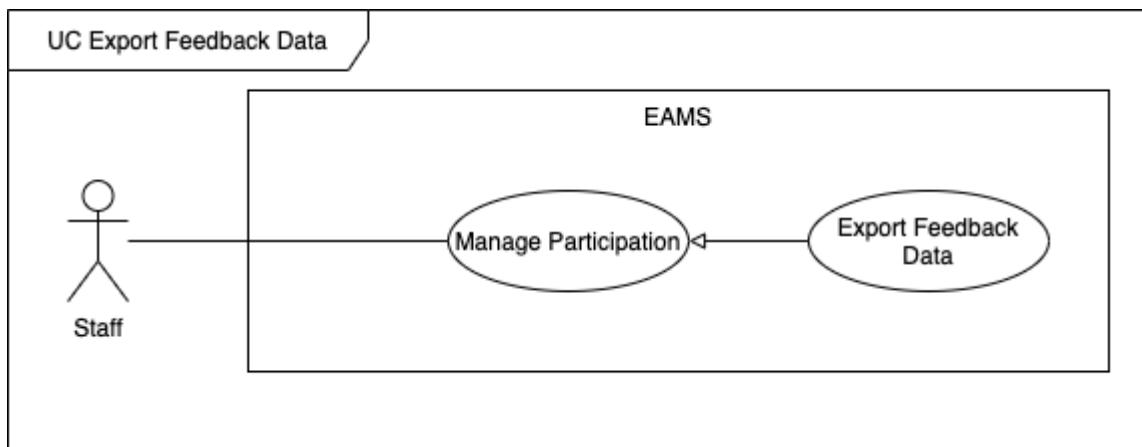
**Figure 70. <Use Case> Get Student Assessment**

ID and Name:	<b>UC-63. Get Student Assessment</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Get list of students by semester and their extracurricular grade		
Trigger:	Staff sends request to get student assessments list		
Preconditions:	PRE-1. User has logged in as Staff role		
Post-conditions:	POST-1. EAMS returns list of student assessments		
Normal Flow:	Step	Actor Action	System Response
	1	Staff request to get student assessments list	
			EAMS return a list of student assessment
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	High		
Frequency of Use:	Approximately 20 staffs use this feature		
Business Rules:	<ul style="list-style-type: none"> <li>• By the start of a semester, students list has to be imported to the system in order to determine if a student is still active. For example, if a student drops out, staff may decide to deactivate that student's account</li> <li>• Process of assessment:</li> </ul>		

	<ul style="list-style-type: none"> <li>○ Search in current semester, get all active students</li> <li>○ Check activity histories of processed student based on current semester's events</li> <li>○ Get total grade by every criterion of processed student</li> <li>○ Compare total grade by criterion and criterion's max grade to produce capped grade</li> </ul>																		
	<table border="1"> <thead> <tr> <th>Criterion name</th> <th>Grade per event</th> <th>Max grade</th> </tr> </thead> <tbody> <tr> <td>Sports &amp; Arts</td> <td>5</td> <td>25</td> </tr> <tr> <td>Seminar, Workshop &amp; Talkshows</td> <td>10</td> <td>50</td> </tr> <tr> <td>Community, Experiences and Cultural</td> <td>5</td> <td>25</td> </tr> <tr> <td>Others</td> <td>5</td> <td>25</td> </tr> <tr> <td>Contribution</td> <td>Depends on students' contribution</td> <td>70</td> </tr> </tbody> </table>	Criterion name	Grade per event	Max grade	Sports & Arts	5	25	Seminar, Workshop & Talkshows	10	50	Community, Experiences and Cultural	5	25	Others	5	25	Contribution	Depends on students' contribution	70
Criterion name	Grade per event	Max grade																	
Sports & Arts	5	25																	
Seminar, Workshop & Talkshows	10	50																	
Community, Experiences and Cultural	5	25																	
Others	5	25																	
Contribution	Depends on students' contribution	70																	
	<ul style="list-style-type: none"> <li>○ Capped grade range to assess a student <ul style="list-style-type: none"> <li>▪ 0 – 9: need improvement</li> <li>▪ 10 – 54: average</li> <li>▪ 55 – 99: good</li> <li>▪ &gt; 100: excellent</li> </ul> </li> </ul>																		
Other Information:	In case of network failures, staff cannot finish this use case properly																		
Assumptions:	N/A																		

**Table 63. <Use Case> Get Student Assessment**

#### 2.2.4.30 Export Feedback Data



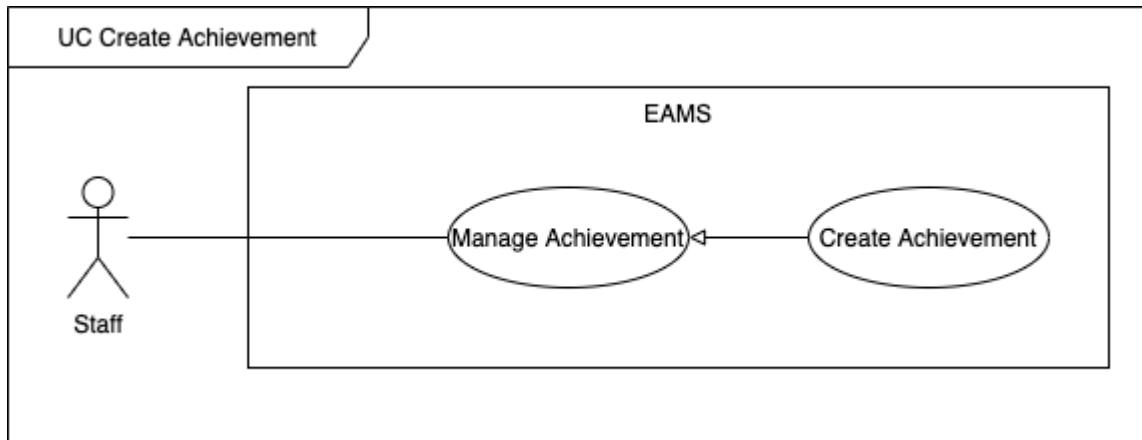
**Figure 71. <Use Case> Export Feedback Data**

ID and Name:	<b>UC-64. Export Feedback Data</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A

Description:	This feature is to export a list of feedback answers of that event		
Trigger:	Staff sends requests to download feedback answers list		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role		
Post-conditions:	POST-1. EAMS returns list of feedback answers		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends request with a specific event to get feedback answers list of that events	
			EAMS returns an excel file that contains feedback answers [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
		Event does not exist	EAMS informs that event does not exist
Priority:	Medium		
Frequency of Use:	Approximately 1 usage per event		
Business Rules:	<ul style="list-style-type: none"> <li>Staffs are provided some download reports feature such as registrants report, participation reports and feedback reports to support staff getting an overview of the current event.</li> <li>Staffs do not know the identity of student who gives feedbacks</li> </ul>		
Other Information:	In case of internet connection failure, staff cannot download feedback answers data		
Assumptions:	Staffs have Microsoft Office on their computer		

**Table 64. <Use Case> Export Feedback Data**

#### 2.2.4.31 Create Achievement



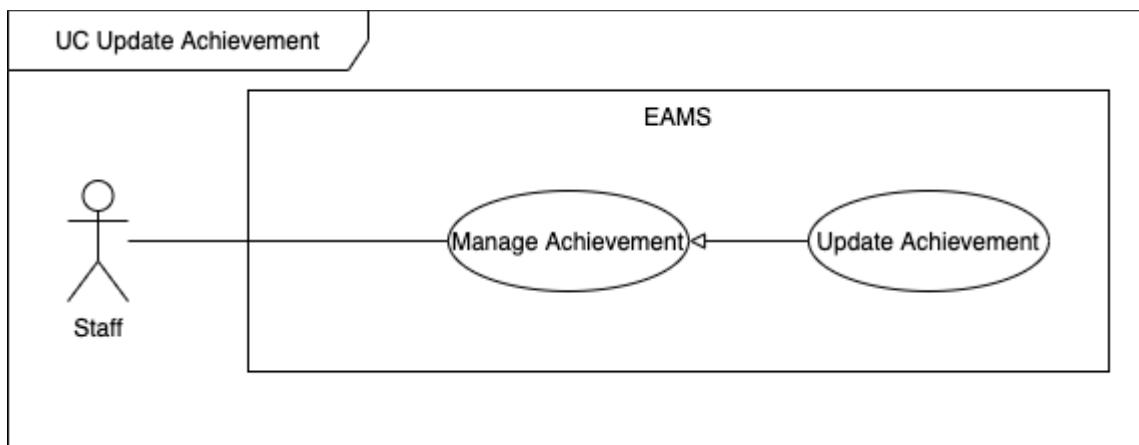
**Figure 72. <Use Case> Create Achievement**

ID and Name:	<b>UC-65. Create Achievement</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This feature allows staff to create achievement for students		
Trigger:	Staff sends a request to create an achievement		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role		
Post-conditions:	POST-1. EAMS return a message that achievement has been created successfully		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends a request with a list of students to create an achievement	
			EAMS validates all fields and returns success message. [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Fields are invalidated	EAMS informs that fields are invalidated
Priority:	Medium		
Frequency of Use:	Approximately 2 usage per event		
Business Rules:	<ul style="list-style-type: none"> <li>Achievements are granted for students for their performances in extracurricular activities. There are 2 types of achievements: <ul style="list-style-type: none"> <li>Achievements for participations and contributions: Students can earn this achievement by participating in or contributing to an event. Achievement of this type can only be granted for participating in or contributing to finished and graded events.</li> <li>Achievements for special reasons: Universities can grant students achievements for exceptional performances in extracurricular activities.</li> </ul> </li> <li>An achievement requires following fields: <ul style="list-style-type: none"> <li>Title: between 1 - 63 characters</li> <li>Description: between 1 - 127 characters</li> <li>Semester: The semester an achievement is associated to</li> <li>Issue At: The date when an achievement is issued</li> <li>Reason: Reason for an achievement that is recorded in the achievement file. Ranges between 1 - 63 characters</li> <li>Fetch from: Determine which students are qualified for the achievement</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ Achievement template: The template used to generate achievement</li> <li>○ Event: The event that an achievement is associated to</li> <li>○ List of students who receive this achievement</li> <li>● An achievement also has optional field: <ul style="list-style-type: none"> <li>○ Alternative event title: between 1 - 63 characters</li> </ul> </li> <li>● Participants and contributors are granted achievements by staff after an event finishes. <ul style="list-style-type: none"> <li>○ Participants can only be granted achievement if they register and check-in for an event. If an event requires check-out, that student has to check-out in order to qualify for participation recognition and achievement. A participant can be disqualified by staffs if they have a reason for it. Qualification and disqualification histories are saved in the system with the date time and reasons for them</li> <li>○ Contributors may not be qualify for an achievement if their event contribution grade does not meet a certain threshold determined by staff.</li> </ul> </li> </ul>
Other Information:	In case of internet connection failure, staff cannot create achievement
Assumptions:	N/A

**Table 65. <Use Case> Create Achievement**

#### 2.2.4.32 Update Achievement



**Figure 73. <Use Case> Update Achievement**

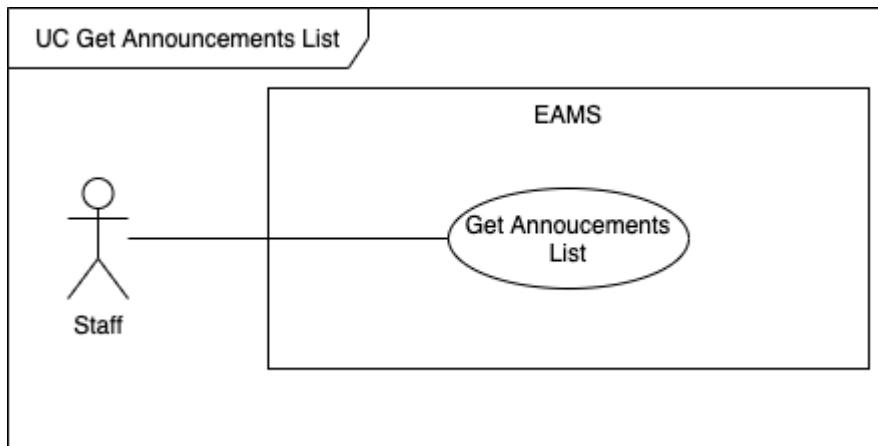
ID and Name:	<b>UC-66. Update Achievement</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This feature allows staff to update achievement		
Trigger:	Staff sends requests to update an achievement		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role		

Post-conditions:	POST-1. EAMS returns a message that achievement has been updated successfully		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends a request with a list of students to update an achievement	
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	List of students is empty	EAMS informs that list of students is empty
Priority:	Medium		
Frequency of Use:	Approximately 1 usage per event		
Business Rules:	<ul style="list-style-type: none"> <li>In EAMS, after being graded for an event, participations list and contributions list will be handed over to award stage where staff will decide which one will receive achievement.</li> <li>To enter this stage, student has checked-out if event requires check-out or checked-in if event doesn't. And event's grade of that student is larger than 0, whether that student is graded or not depends on staff in grading stage.</li> <li>Achievements are granted for students for their performances in extracurricular activities. There are 2 types of achievements: <ul style="list-style-type: none"> <li>Achievements for participations and contributions: Students can earn this achievement by participating in or contributing to an event. Achievement of this type can only be granted for participating in or contributing to finished and graded events.</li> <li>Achievements for special reasons: Universities can grant students achievements for exceptional performances in extracurricular activities.</li> </ul> </li> <li>An achievement requires following fields: <ul style="list-style-type: none"> <li>Title: between 1 - 63 characters</li> <li>Description: between 1 - 127 characters</li> <li>Semester: The semester an achievement is associated to</li> <li>Issue At: The date when an achievement is issued</li> <li>Reason: Reason for an achievement that is recorded in the achievement file. Ranges between 1 - 63 characters</li> <li>Fetch from: Determine which students are qualified for the achievement</li> <li>Achievement template: The template used to generate achievement</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>○ Event: The event that an achievement is associated to</li> <li>○ List of students who receive this achievement</li> </ul> <p>An achievement also has optional field:</p> <ul style="list-style-type: none"> <li>○ Alternative event title: between 1 - 63 characters</li> <li>● Staff cannot update achievement detail after creating, only update list of students of achievements</li> </ul>
Other Information:	In case of internet connection failure, staff cannot update achievement
Assumptions:	N/A

**Table 66. <Use Case> Update Achievement**

#### 2.2.4.33 Get Announcements List



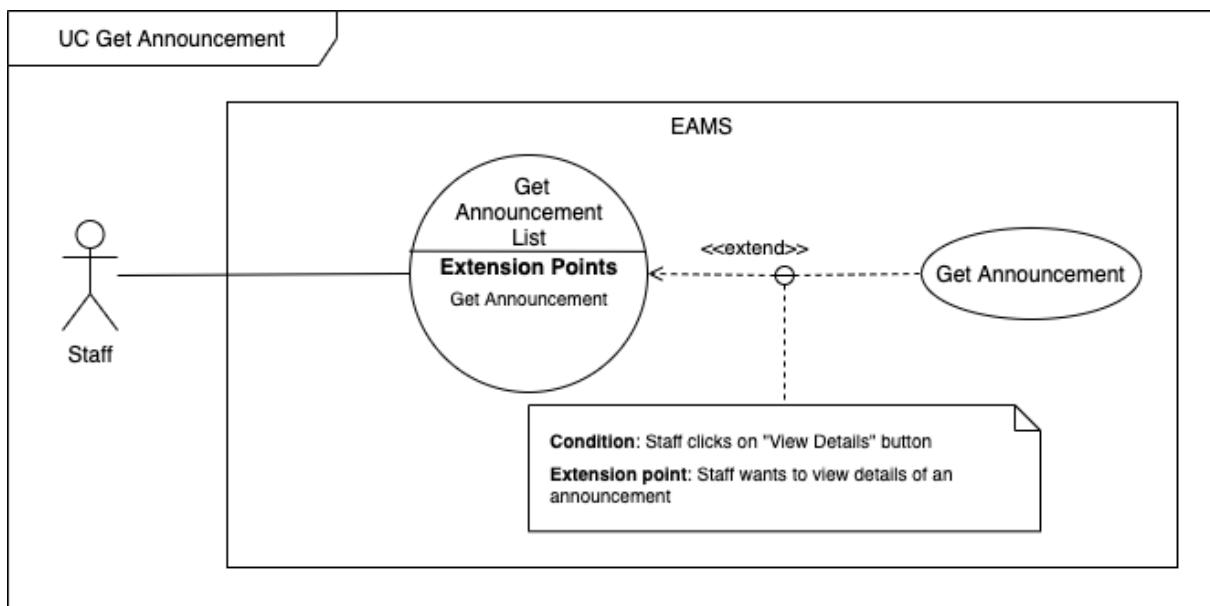
**Figure 74. <Use Case> Get Announcements List**

ID and Name:	<b>UC-67. Get Announcements List</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This feature allows staff to get list of current announcements by event		
Trigger:	Staff sends a request to retrieve list of announcements		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role		
Post-conditions:	POST-1. EAMS returns announcements list		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends a request to get a list of announcements by event	
			EAMS returns announcements list
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response

Priority:	Medium
Frequency of Use:	Approximately 3 usages per event
Business Rules:	<ul style="list-style-type: none"> <li>EAMS provides some features that support sending email more convenient <ul style="list-style-type: none"> <li>Linking announcement to specific event which is easy for staff to manage announcement about an event</li> <li>Fetch students list (receivers) from registrants list, participants list of an event.</li> </ul> </li> <li>Announcement is a mean for staff to communicate with students more effectively using emails about information such as new achievements and urgent messages.</li> </ul>
Other Information:	In case of internet connection failure, staff cannot get announcements list
Assumptions:	N/A

**Table 67. <Use Case> Get Announcements List**

#### 2.2.4.34 Get Announcement



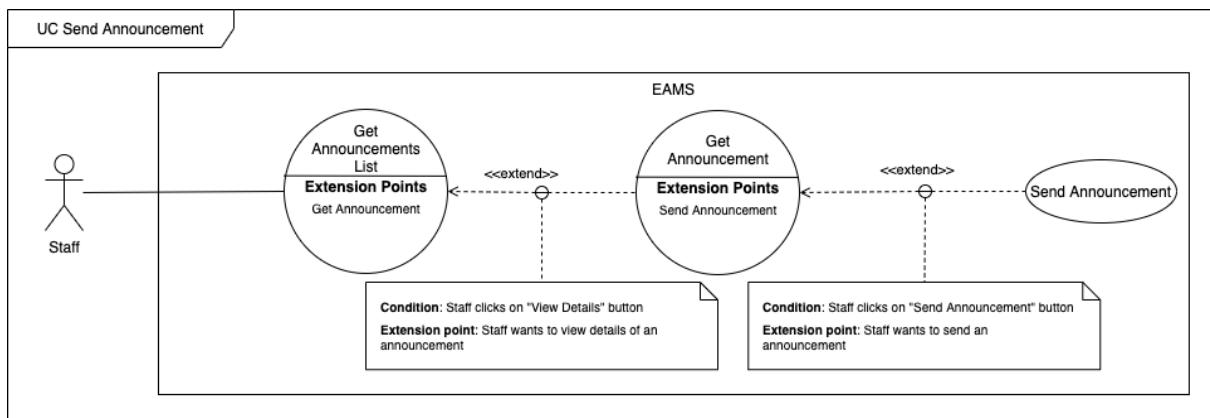
**Figure 75. <Use Case> Get Announcement**

ID and Name:	<b>UC-68. Get Announcement</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This feature allows staff to get announcement detail		
Trigger:	Staff sends a request to retrieve announcement detail		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role		
Post-conditions:	POST-1. Staff receives announcement detail		
Normal Flow:	Step	Actor Action	System Response

	1	Staff sends a request to obtain announcement detail	
		EAMS returns announcement detail	
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
Priority:	Medium		
Frequency of Use:	Approximately 3 usages per event		
Business Rules:	<ul style="list-style-type: none"> <li>EAMS provides some features that support sending email more convenient: <ul style="list-style-type: none"> <li>Linking announcement to specific event which is easy for staff to manage announcement about an event</li> <li>Fetch students list (receivers) from registrants list, participants list of an event.</li> </ul> </li> <li>Announcement is a mean for staff to communicate with students more effectively using emails about information such as new achievements and urgent messages.</li> </ul>		
Other Information:	In case of internet connection failure, staff cannot get announcement detail		
Assumptions:	N/A		

**Table 68. <Use Case> Get Announcement**

#### 2.2.4.35 Send Announcement



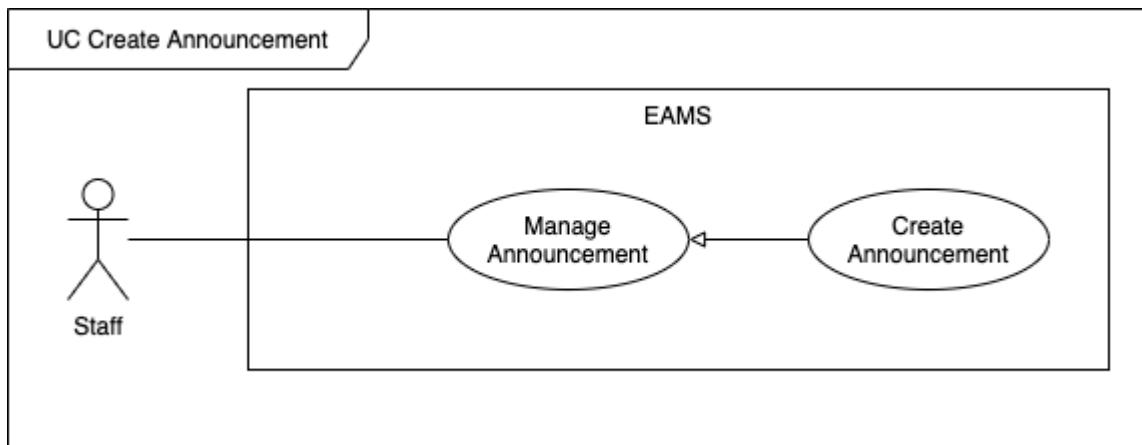
**Figure 76. <Use Case> Send Announcement**

ID and Name:	<b>UC-69. Send Announcement</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This feature allows staff to send announcement to multiple students		

Trigger:	Staff sends a request to send an announcement		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role		
Post-conditions:	POST-1. Students receives announcement that has been sent to staff		
Normal Flow:	Step	Actor Action	System Response
	1	1. Staff sends a request to send announcement to multiple students	
			EAMS update status of announcement then returns success message to staff.
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
Priority:	Medium		
Frequency of Use:	Approximately 3 usages per event		
Business Rules:	<ul style="list-style-type: none"> <li>EAMS provides some features that support sending email more convenient: <ul style="list-style-type: none"> <li>Linking announcement to specific event which is easy for staff to manage announcement about an event</li> <li>Fetch students list (receivers) from registrants list, participants list of an event.</li> </ul> </li> <li>An announcement can be sent once</li> <li>Announcement is a mean for staff to communicate with students more effectively using emails about information such as new achievements and urgent messages.</li> </ul>		
Other Information:	In case of internet connection failure, staff cannot send announcement		
Assumptions:	N/A		

**Table 69. <Use Case> Send Announcement**

#### 2.2.4.36 Create Announcement



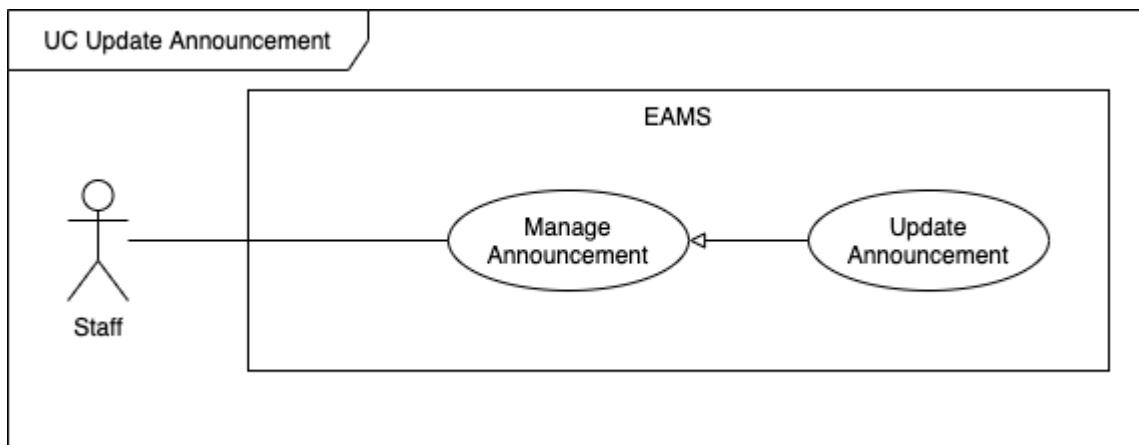
**Figure 77. <Use Case> Create Announcement**

ID and Name:	<b>UC-70. Create Announcement</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This feature allows staff to create an announcement		
Trigger:	Staff sends a request to create an announcement		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role		
Post-conditions:	POST-1. EAMS stores announcement		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends a request with a list of students who will receive this announcement	
			EAMS validates all fields then return success message [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
		Student is not available in system	EAMS returns a message that student does not exist
Priority:	Medium		
Frequency of Use:	Approximately 3 usages per event		
Business Rules:	<ul style="list-style-type: none"> <li>EAMS provides some features that support sending email more convenient: <ul style="list-style-type: none"> <li>Linking announcement to specific event which is easy for staff to manage announcement about an event</li> <li>Fetch students list (receivers) from registrants list, participants list of an event.</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>Announcement is a mean for staff to communicate with students more effectively using emails about information such as new achievements and urgent messages.</li> <li>An announcement acquires following fields: <ul style="list-style-type: none"> <li>Mail content</li> <li>Mail subject</li> <li>Announcement brief</li> </ul> </li> </ul>
Other Information:	In case of internet connection failure, staff cannot create announcement
Assumptions:	N/A

**Table 70. <Use Case> Create Announcement**

#### 2.2.4.37 Update Announcement



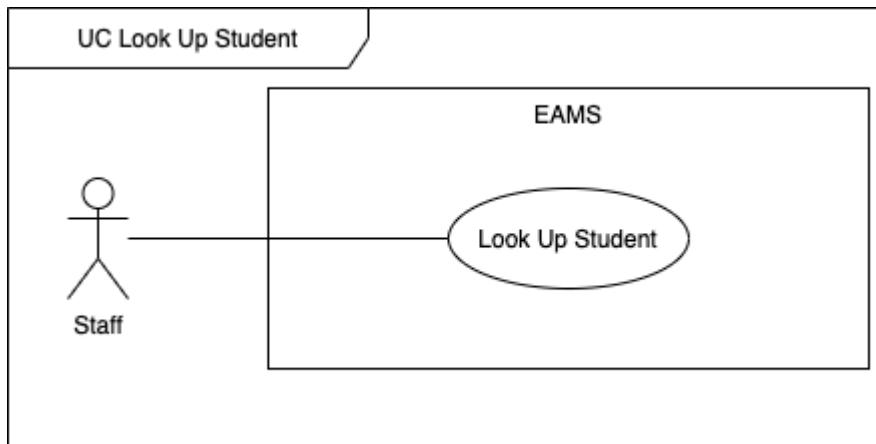
**Figure 78. <Use Case> Update Announcement**

ID and Name:	<b>UC-71. Update Announcement</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This feature allows staff to update an announcement		
Trigger:	Staff sends a request to update an announcement		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role		
Post-conditions:	POST-1. EAMS saves announcement		
Normal Flow:	Step	Actor Action	System Response
	1	Staff sends a request with a list of students who will receive this announcement for updating purpose	
			EAMS validates all fields then return success message [Exception 1]

Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
		Student is not available in system	EAMS returns a message that student does not exist
Priority:	Medium		
Frequency of Use:	Approximately 3 usages per event		
Business Rules:	<ul style="list-style-type: none"> <li>Announcement is a mean for staff to communicate with students more effectively using emails about information such as new achievements and urgent messages.</li> <li>An announcement acquires following fields: <ul style="list-style-type: none"> <li>Mail content</li> <li>Mail subject</li> <li>Announcement brief</li> </ul> </li> <li>Staff cannot update an achievement if it has been sent before</li> <li>An announcement can be sent once</li> </ul>		
Other Information:	In case of internet connection failure, staff cannot update achievement		
Assumptions:	N/A		

**Table 71. <Use Case> Update Announcement**

#### 2.2.4.38 Look Up Student



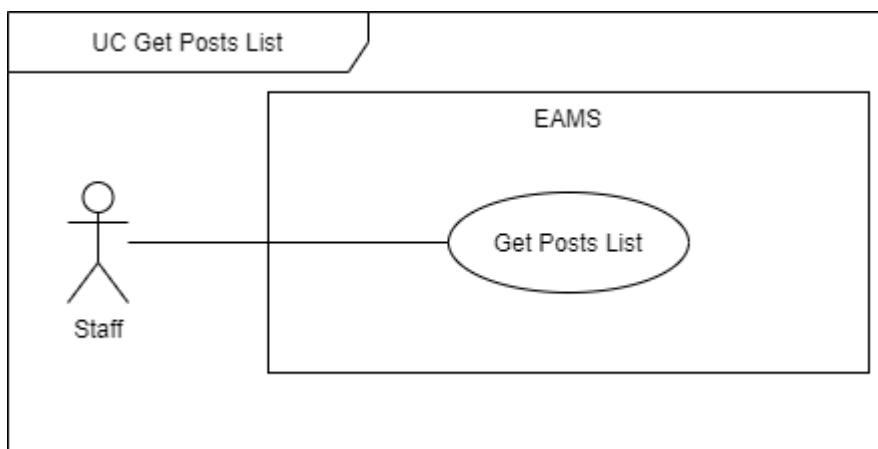
**Figure 79. <Use Case> Look Up Student**

ID and Name:	<b>UC-72. Look Up Student</b>		
Created By:	LuanPHM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	This feature allows staff to search student profile		
Trigger:	Staff sends a request to find out about a student		
Preconditions:	PRE-1. User has logged in into EAMS as Staff role		
Post-conditions:	POST-1. EAMS returns student profile		

Normal Flow:	Step	Actor Action	System Response
	1	Staff sends a request that contains student's email and semester	
			EAMS returns student profile [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
		Student is not available in system	EAMS returns a message that student does not exist
Priority:	Medium		
Frequency of Use:	Usually use		
Business Rules:	<ul style="list-style-type: none"> <li>Search Student Profile is provided for staff to help staffs have an overview of student activities. In profile page, staff can know about: <ul style="list-style-type: none"> <li>Total contribution and participation grade of student</li> <li>Total events that student has participated in or contributed to, in terms of unfinished and finished events</li> <li>Participation and contribution details of each events</li> <li>Achievements</li> </ul> </li> <li>Every information is filtered by semester</li> </ul>		
Other Information:	In case of internet connection failure, staff cannot find student profile		
Assumptions:	N/A		

**Table 72. <Use Case> Look Up Student**

#### 2.2.4.39 Get Posts List



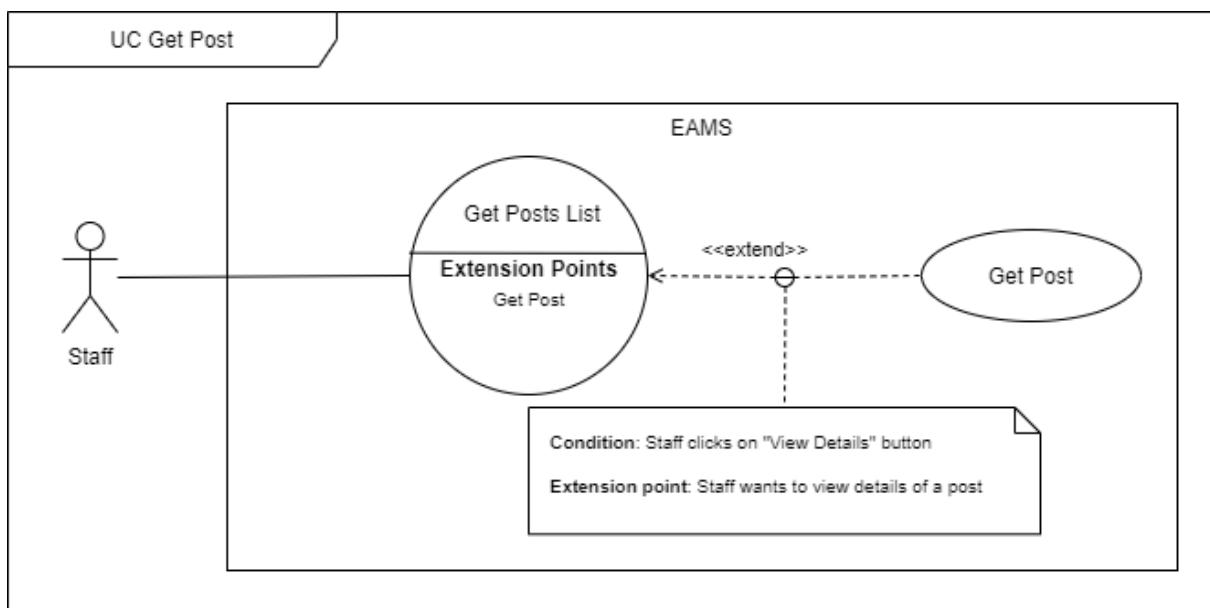
**Figure 80. <Use Case> Get Posts List**

ID and Name:	<b>UC-73. Get Posts List</b>		
Created By:	ChienTM	Date Created:	02/12/20

Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff gets a list of event's posts		
Trigger:	Staff sends request to get event's posts list		
Preconditions:	PRE-1. User has logged in as Staff role		
Post-conditions:	POST-1. EAMS returns a list of posts		
Normal Flow:	Step	Actor Action	System Response
	1	Staff request to get posts list	
			EAMS returns list of posts
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	There is no post of this event	EAMS returns that there is no post of this event
Priority:	Low		
Frequency of Use:	Every time an event is organized		
Business Rules:	<ul style="list-style-type: none"> <li>Some events need to provide more detail information which cannot be expressed clearly in event's description or short description. Post can tackle this problem as a mean of updating new information for students.</li> <li>A post belongs to a specific event</li> </ul>		
Other Information:	In case of network failures, staff cannot finish this use case properly		
Assumptions:	N/A		

**Table 73. <Use Case> Get Posts List**

#### 2.2.4.40 Get Post

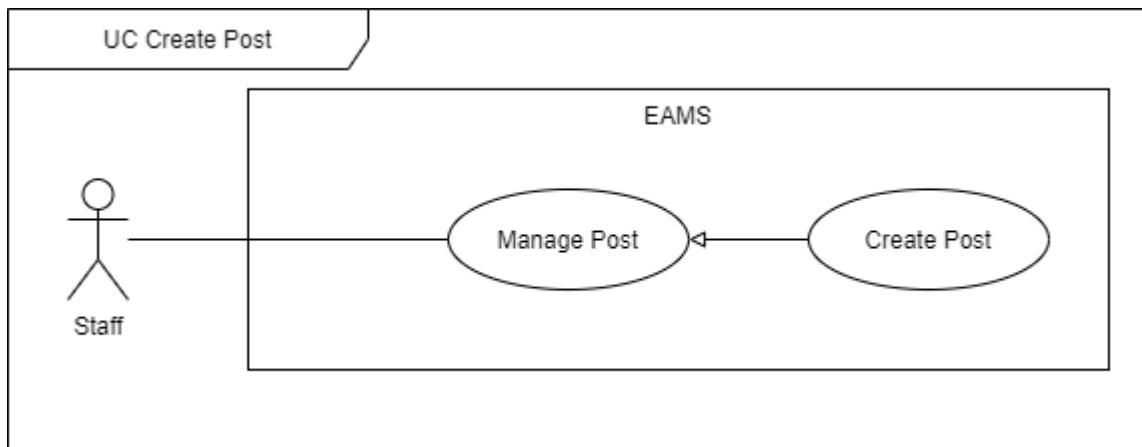


**Figure 81. <Use Case> Get Post**

ID and Name:	<b>UC-74. Get Post</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff gets event's post details		
Trigger:	Staff sends request to get post details		
Preconditions:	PRE-1. User has logged in as Staff role		
Post-conditions:	POST-1. EAMS returns that details of a post		
Normal Flow:	Step	Actor Action	System Response
	1	Staff request to get post details [Exception 1]	
			EAMS returns details of a post
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	The requested post does not exist	EAMS returns that the requested post is not found
Priority:	Low		
Frequency of Use:	Every time an event is organized		
Business Rules:	<ul style="list-style-type: none"> <li>Some events need to provide more detail information which cannot be expressed clearly in event's description or short description. Post can tackle this problem as a mean of updating new information for students.</li> <li>A post belongs to a specific event</li> </ul>		
Other Information:	In case of network failures, staff cannot finish this use case properly		
Assumptions:	N/A		

**Table 74. <Use Case> Get Post**

#### 2.2.4.41 Create Post



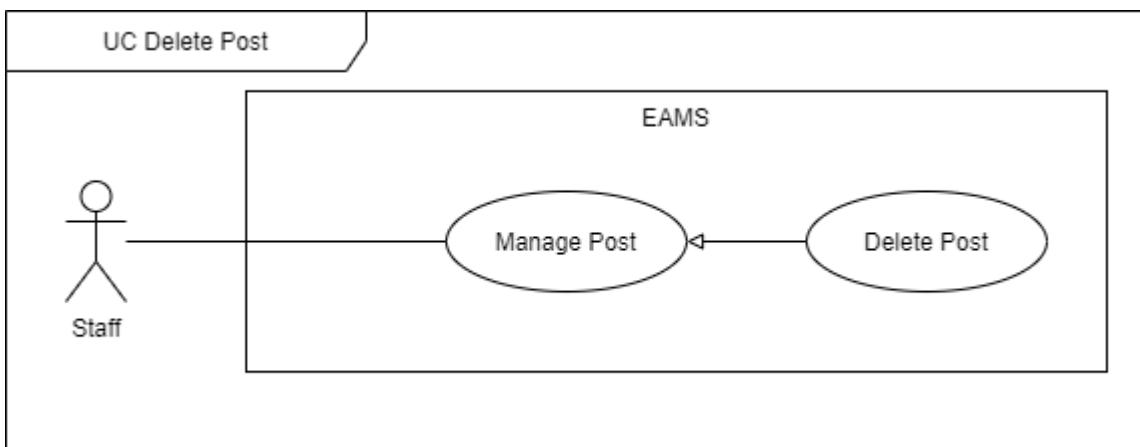
**Figure 82. <Use Case> Create Post**

ID and Name:	<b>UC-75. Create Post</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff creates new post of an event to give student more information		
Trigger:	Staff sends request to create new post		
Preconditions:	PRE-1. User has logged in as Staff role		
Post-conditions:	POST-1. EAMS returns that create successfully		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to create new post	
			EAMS requires staff to fill post's details
	2	Staff fills required fields and finish creating [Exception 1]	
			EAMS returns that create new post successfully
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Data is invalidated	EAMS returns that create failed
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ul style="list-style-type: none"> <li>Some events need to provide more detail information which cannot be expressed clearly in event's description or short description. Post can tackle this problem as a mean of updating new information for students.</li> </ul>		

	<ul style="list-style-type: none"> <li>• A post belongs to a specific event</li> <li>• Validation rule of creating new post: <ul style="list-style-type: none"> <li>◦ Title: required, under 127 characters</li> <li>◦ Short description: required, under 127 characters</li> <li>◦ Content: required, under 4095 characters</li> </ul> </li> </ul>
Other Information:	In case of network failures, staff cannot finish this use case properly
Assumptions:	N/A

**Table 75. <Use Case> Create Post**

#### 2.2.4.42 Delete Post



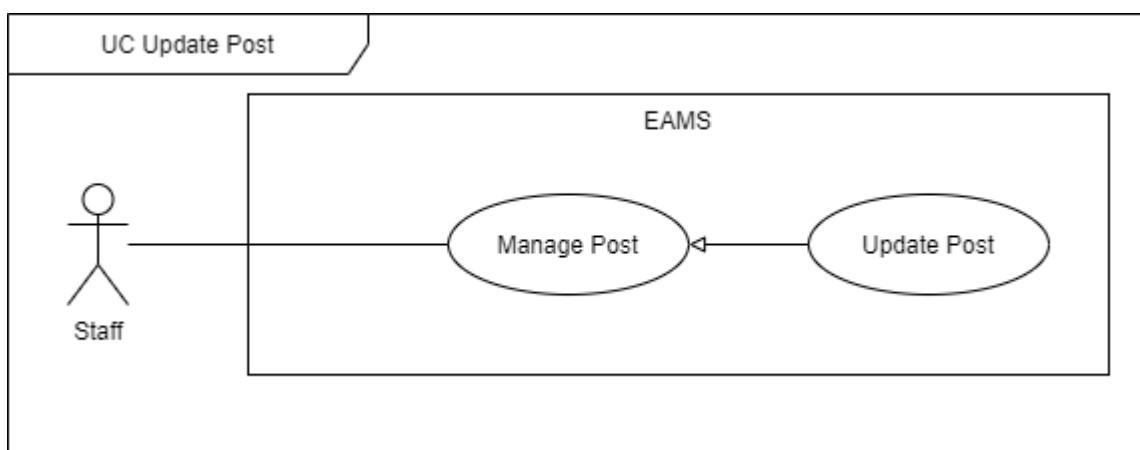
**Figure 83. <Use Case> Delete Post**

ID and Name:	<b>UC-76. Delete Post</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff deletes a post of an event		
Trigger:	Staff sends request to delete post		
Preconditions:	PRE-1. User has logged in as Staff role		
Post-conditions:	POST-1. EAMS returns that delete successfully		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to delete a specific post	
			EAMS returns that delete post successfully
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Low		

Frequency of Use:	Rarely
Business Rules:	<ul style="list-style-type: none"> <li>Some events need to provide more detail information which cannot be expressed clearly in event's description or short description. Post can tackle this problem as a mean of updating new information for students.</li> <li>A post belongs to a specific event</li> </ul>
Other Information:	In case of network failures, staff cannot finish this use case properly
Assumptions:	N/A

**Table 76. <Use Case> Delete Post**

#### 2.2.4.43 Update Post



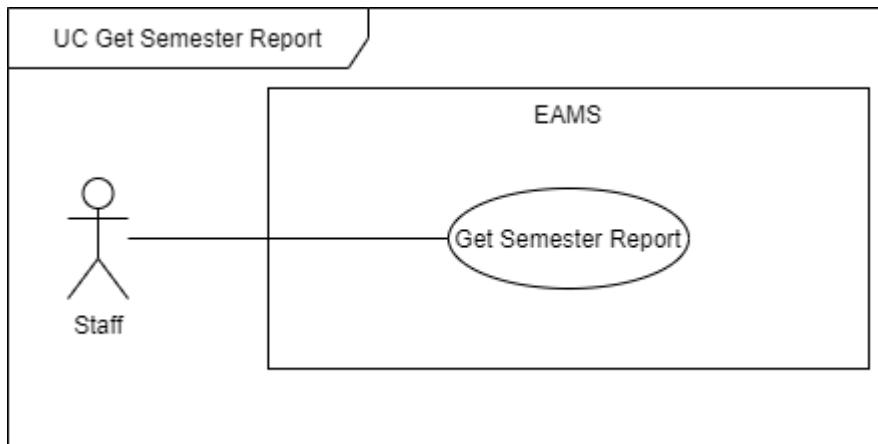
**Figure 84. <Use Case> Update Post**

ID and Name:	<b>UC-77. Update Post</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Staff updates existed event post		
Trigger:	Staff sends request to update a post		
Preconditions:	PRE-1. User has logged in as Staff role		
Post-conditions:	POST-1. EAMS returns that update successfully		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to update a specific post	
			EAMS requires staff to update post's details
	2	Staff fills required fields and finish updating [Exception 1]	
			EAMS returns that update post successfully

Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Data is invalidated	EAMS returns that update failed
Priority:	Low		
Frequency of Use:	Rarely		
Business Rules:	<ul style="list-style-type: none"> <li>Some events need to provide more detail information which cannot be expressed clearly in event's description or short description. Post can tackle this problem as a mean of updating new information for students.</li> <li>A post belongs to a specific event</li> <li>Validation rule of updating existing post: <ul style="list-style-type: none"> <li>Title: required, under 127 characters</li> <li>Short description: required, under 127 characters</li> <li>Content: required, under 4095 characters</li> </ul> </li> </ul>		
Other Information:	In case of network failures, staff cannot finish this use case properly		
Assumptions:	N/A		

**Table 77. <Use Case> Update Post**

#### 2.2.4.44 Get Semester Report



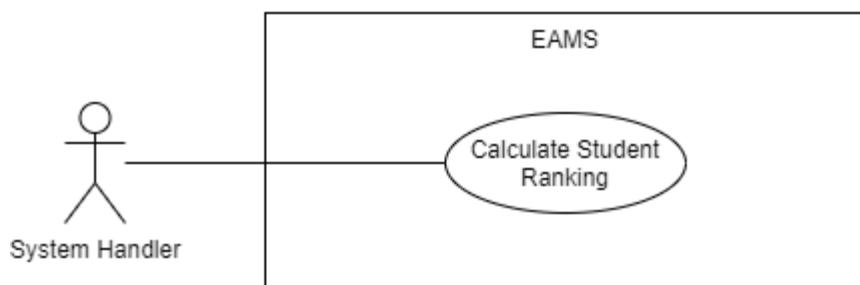
**Figure 85. <Use Case> Get Semester Report**

ID and Name:	<b>UC-78. Get Semester Report</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	Staff	Secondary Actors:	N/A
Description:	Download excel report about student assessment of a specific semester		
Trigger:	Staff requests to get semester report		
Preconditions:	PRE-1. User has logged in as Staff role		

Post-conditions:	POST-1. An excel file is downloaded		
Normal Flow:	Step	Actor Action	System Response
	1	Staff requests to get semester report	EAMS return an downloadable excel file
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Medium		
Frequency of Use:	At the end of semester		
Business Rules:	<ul style="list-style-type: none"> <li>Semester report is provided for rewarding purpose: top students shall be rewarded</li> <li>Semester report includes all students of a semester, contains information about their ranking and assessment level</li> </ul>		
Other Information:	In case of network failures, staff cannot finish this use case properly		
Assumptions:	N/A		

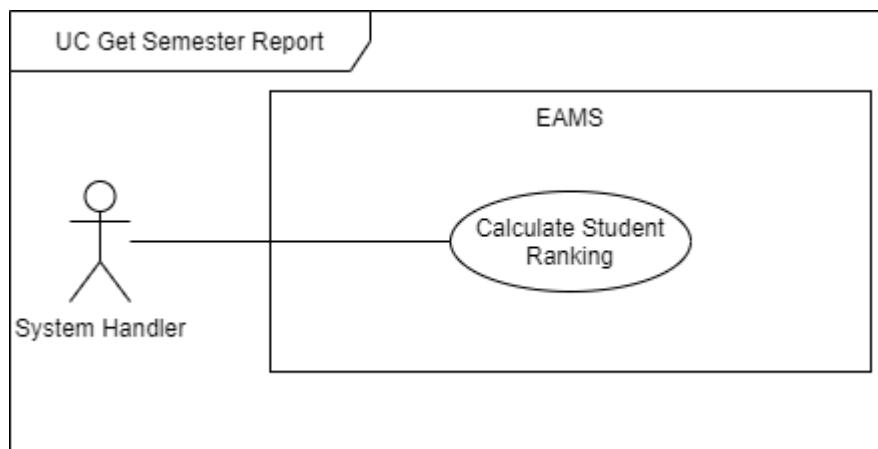
**Table 78. <Use Case> Get Semester Report**

## 2.2.5 System Handler



**Figure 86. <Use Case Overview> System Handler**

### 2.2.5.1 Calculate Student Ranking



**Figure 87. <Use Case> Calculate Student Ranking**

ID and Name:	<b>UC-79. Calculate Student Ranking</b>		
Created By:	ChienTM	Date Created:	02/12/20
Primary Actor:	System Handler	Secondary Actors:	N/A
Description:	EAMS calculates student extracurricular activity ranking once per day		
Trigger:	When the system time is 12:00:00 A.M		
Preconditions:	N/A		
Post-conditions:	POST-1. EAMS stores new ranking of students		
Normal Flow:	Step	Actor Action	System Response
	1	EAMS calculate and update new ranking	
			EAMS returns cron job has finished
Alternative Flows:	N/A		
Exceptions:	N/A		
Priority:	Medium		
Frequency of Use:	Once per day		
Business Rules:	<ul style="list-style-type: none"> <li>Ranking is provided for rewarding purpose: top students shall be rewarded</li> <li>EAMS defines 4 grade ranges correspond 4 assessment level: <ul style="list-style-type: none"> <li>Under 10 points: need improvement</li> <li>From 10 to under 55 points: average</li> <li>From 55 to under 100 points: good</li> <li>Above 100 point: excellent</li> </ul> </li> <li>Besides that, to encourage students to participate multiple types of extracurricular activities, EAMS has a term named grade criteria. All events are classified to a criterion. Every criterion has a max grade student can get.</li> </ul>		

	Criterion name	Grade per event	Max grade
	Sports & Arts	5	25
	Seminar, Workshop & Talkshows	10	50
	Community, Experiences and Cultural	5	25
	Others	5	25
	Contribution	About 10	70
	<ul style="list-style-type: none"> <li>• Student ranking in the system serves as a reference material for staff to evaluate semester rewards, EAMS defines some rules</li> </ul> <p>Student ranking is measured by 4 fields, with the corresponding priority:</p> <ul style="list-style-type: none"> <li>- capped grade (by max grade by every criterion)</li> <li>- number of criteria that a student satisfies</li> <li>- total grade (without capped)</li> <li>- number of participated/contributed events</li> </ul> <ul style="list-style-type: none"> <li>• Because of the complexity of ranking calculation, EAMS has an automated task which runs every midnight to re-calculate ranking for all students in the system so users can retrieve student's ranking immediately.</li> </ul>		
Other Information:	N/A		
Assumptions:	N/A		

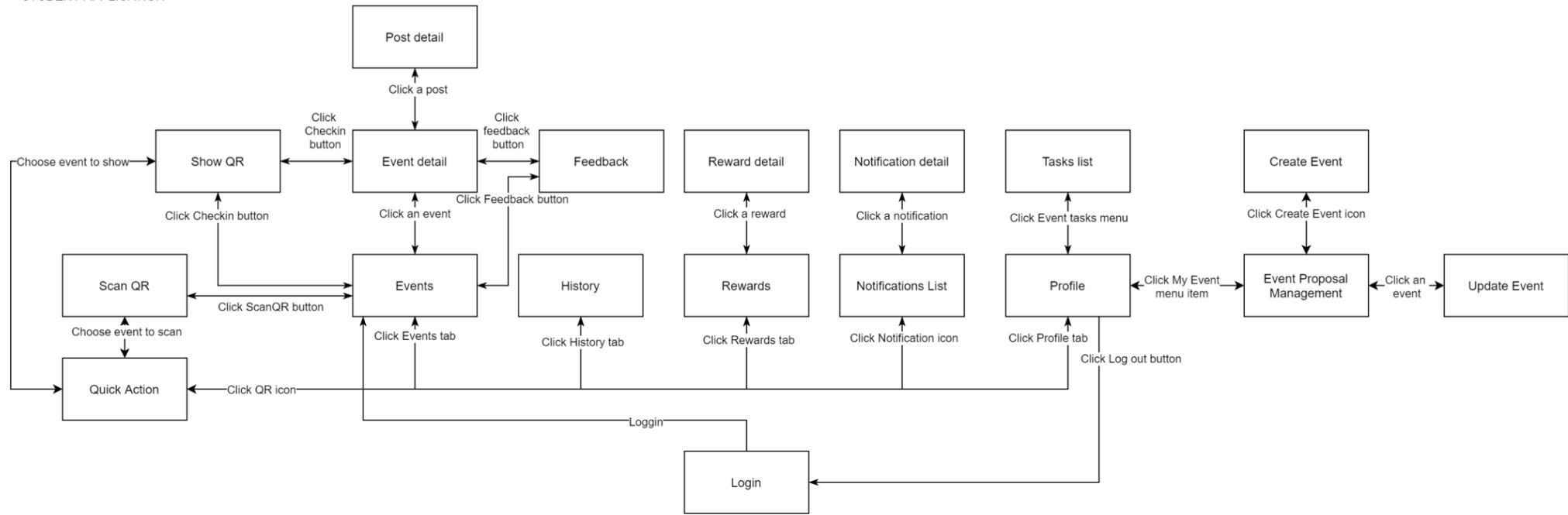
**Table 79. <Use Case> Calculate Student Ranking**

### 3. Functional Requirements

#### 3.1 System Functional Overview

##### 3.1.1 Screen flow

STUDENT APPLICATION



**Figure 88. <Screen Flow> Student Application**

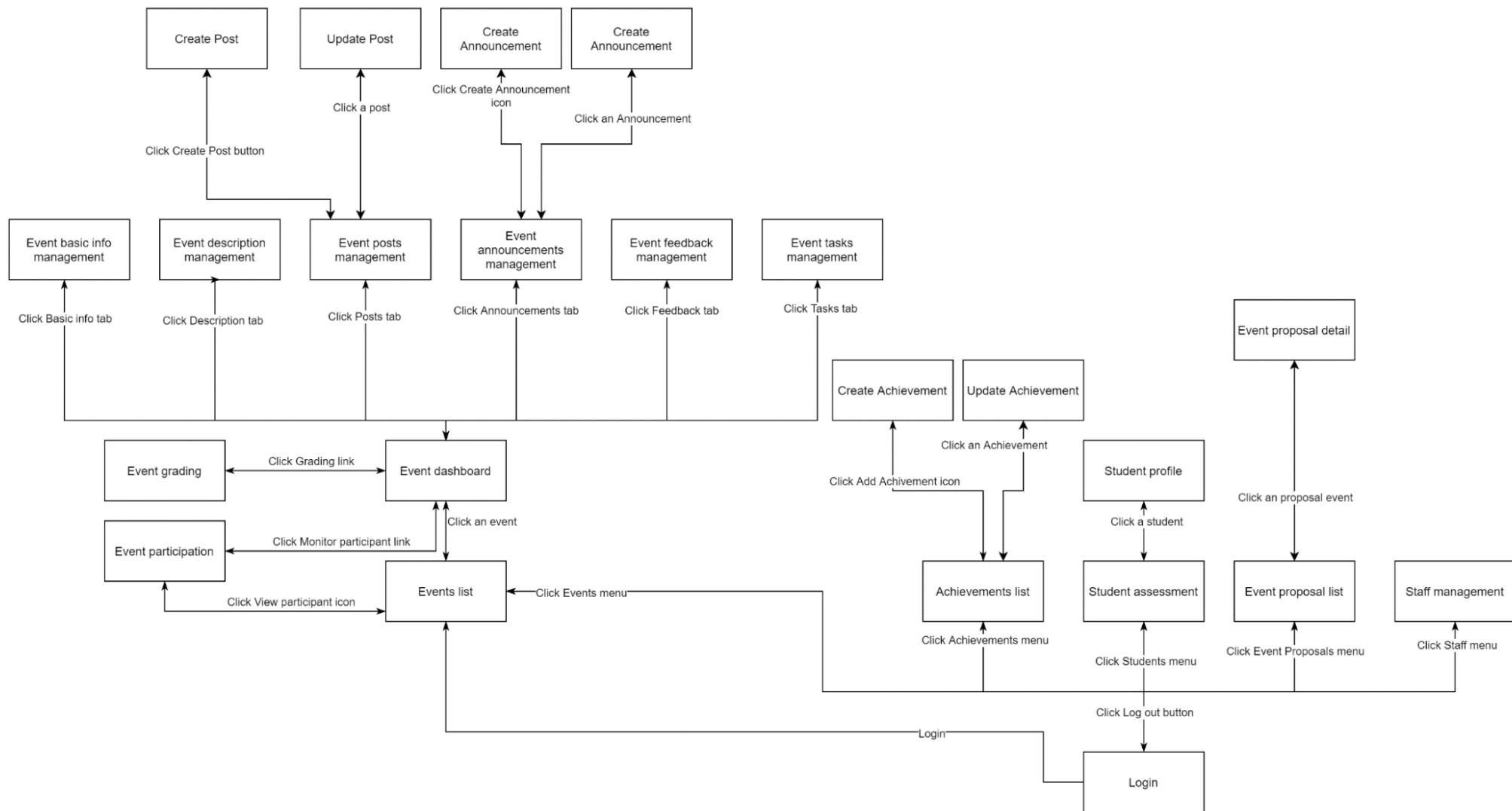


Figure 89. &lt;Screen Flow&gt; Staff Application

### 3.1.2 Non-screen Functions

No	System Function	Description
1	Calculate student ranking	This is a cron job runs every day at 12A.M to calculate student grade ranking

### 3.1.3 Entity Relationship Diagram

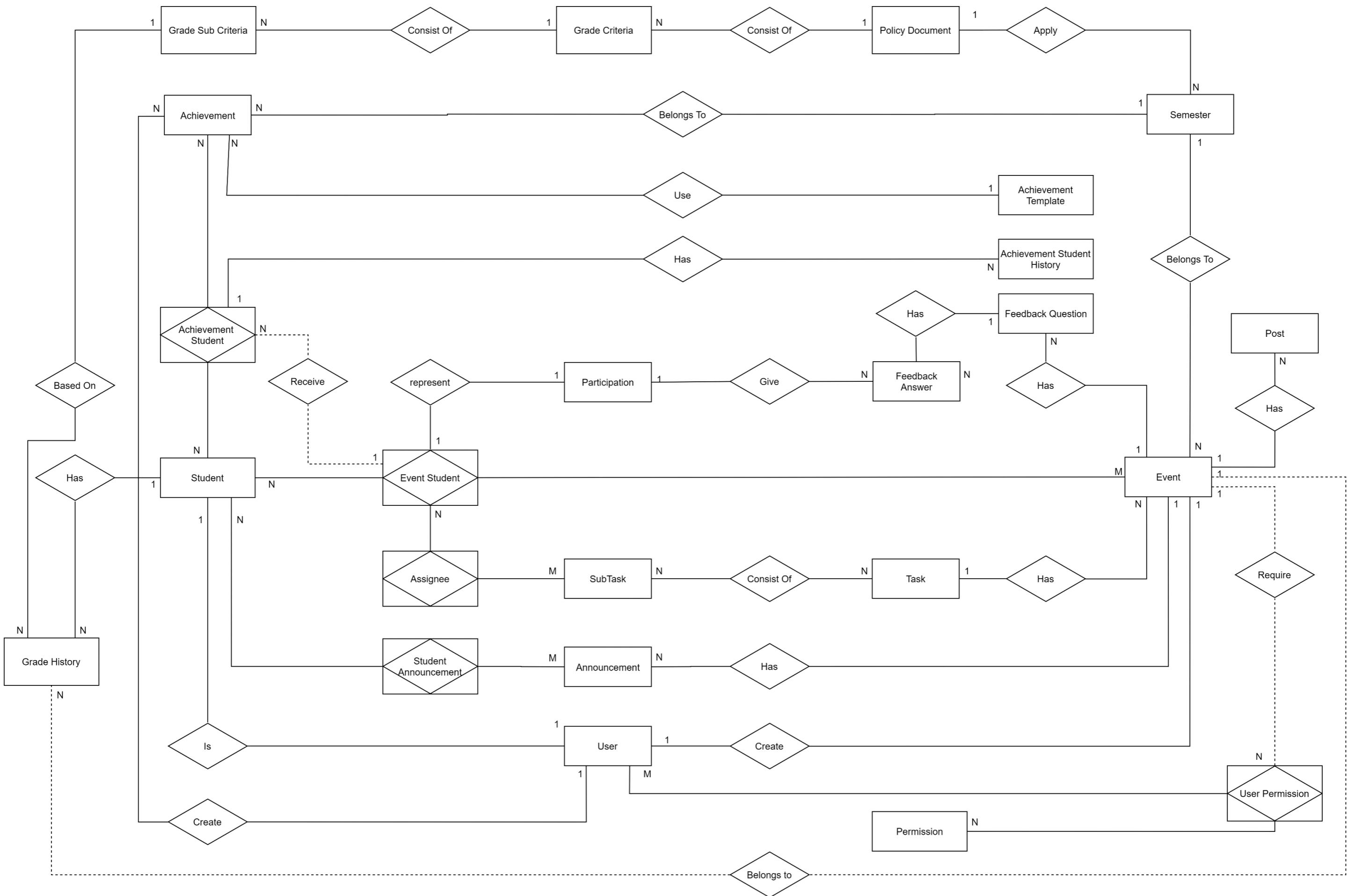
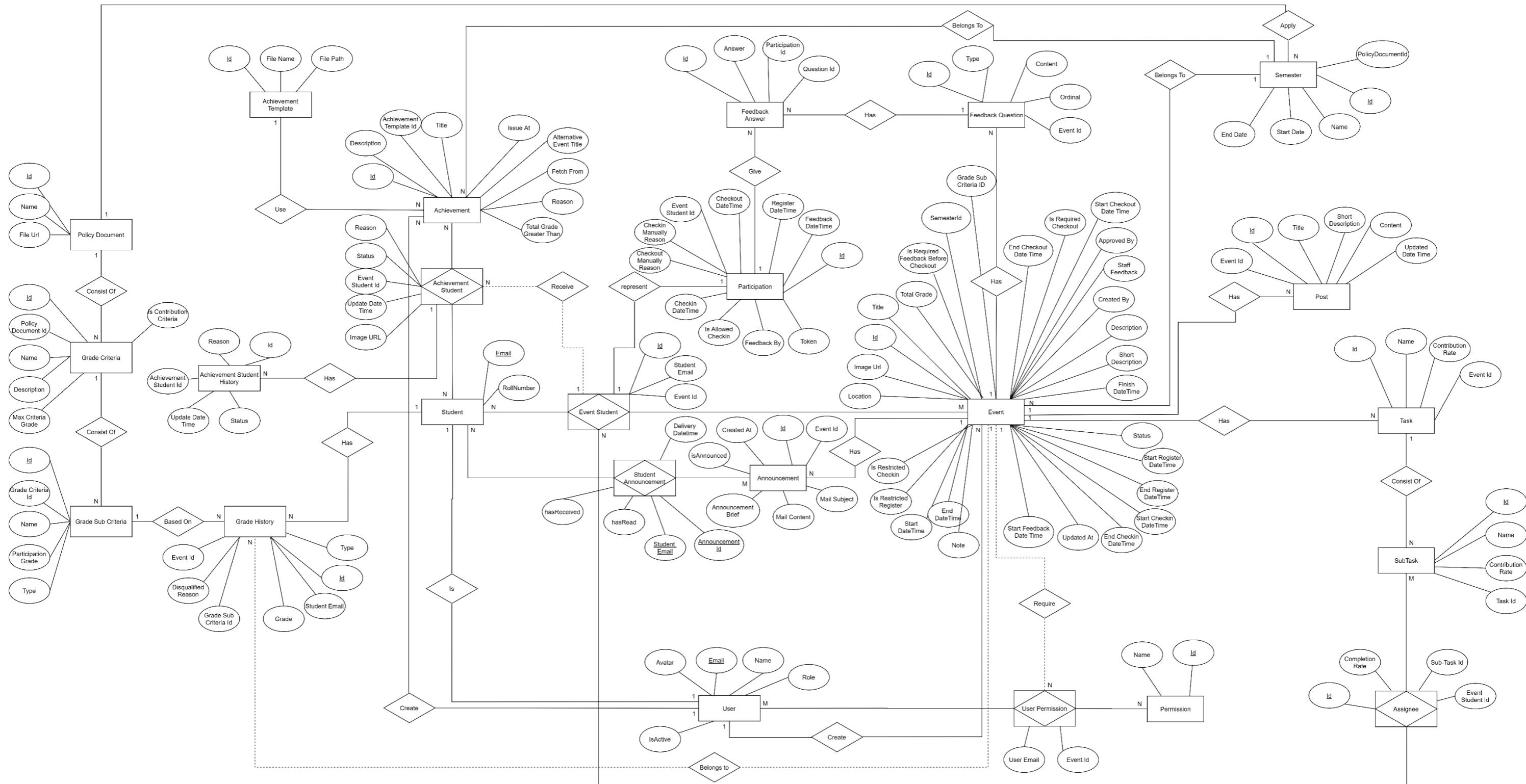


Figure 90. Entity Relationship Diagram (without attributes)



**Figure 91. Entity Relationship Diagram**

#	Entity	Description
1	Policy Document	Represents the document which the university uses to assess students' extracurricular activities grades
2	Grade Criteria	The criteria that the university takes into account when assess students based on types of extracurricular activities
3	Grade Sub-criteria	The sub-criteria that the university uses to assign what types of activities get how many points
4	Grade History	The histories of students that reflects their extracurricular activities performances
5	Achievement Student History	Represent a change in students' histories of an achievement
6	Achievement Template	Templates that are used to create students' achievement files
7	Achievement	The achievements that students can earn during their time in the university
8	Achievement Student	Represent a student's being qualified or disqualified for an achievement
9	Student	University students' information
10	Event Student	Represent a student's participation or contribution in an event
11	Student Announcement	Represent an announcement sent to a student
12	User	The users in the system including staffs and students
13	Feedback Answer	The feedback answers of students to events' feedback questions
14	Participation	Represent students' participation in an event
15	Announcement	The announcements that staffs send to students via emails
16	Feedback Question	The questions that are used for students to give feedbacks for events
17	Event	An extracurricular event in a university
18	User Permission	The permissions that users have over a certain resource
19	Permission	The permissions to access certain resources
20	Semester	The semesters of a university
21	Post	The posts of an event that are used to give updates about that event
22	Task	The tasks that an event has for students to contribute to
23	Sub-task	Sub-tasks belong to tasks. Students contribute to sub-tasks to earn contribution grades
24	Assignee	Each assignee represents a student's contribution to a sub-task

#### 4. Non-Functional Requirements

##### 4.1 External Interfaces

#### 4.1.1 User Interfaces

UI-1: The whole system shall follow Material-UI design

UI-2: The student application shall be mobile friendly

UI-3: System shall inform user anytime an error is fired by a popup

#### 4.1.2 Software Interfaces

SI-1: Authentication Service

SI-1.1: The EAMS can integrate with OAuth2 (Authentication Service) to authenticate user

SI-1.2: EAMS use an Email API to send email

SI-2: File Converter Service

SI-2.1: EAMS can integrate with a File Converter Service system to use its API

SI-2.2: EAMS provide docx file to File Converter Service convert from docx to png and pdf

SI-3: File Storage Service

SI-3.1: EAMS can integrate with a File Storage Service and use its upload file API

#### 4.1.3 Hardware Interfaces

Smart phone with camera is required for student application

### 4.2 Quality Attributes

#### 4.2.1 Usability

- Student can use Student application without network connection to view event information, check-in and check-out
- Users directly test the system to give feedback at development cycle. Consequently, the UI/UX design will be modified as users' desires
- UI/UX follows material design to give users the feelings of using a native app
- EAMS will need less than two days of training for staff to use
- Student can instantly use EAMS with less than 15 minutes of guidance
- User do not have to update their application every time a new version is released

#### 4.2.2 Reliability

- Prevent student from cheating
- The chance of application failure is less than 3%
- EAMS provides manual operations for checking-in procedure to cover unexpected situations: students cannot login, do not have phone..
- Number of critical failures is less than 3 per month
- In case of application errors, hotfix version can be deployed in less than 3 hours

#### 4.2.3 Availability

- EAMS is available 100% of the time
- EAMS can be deployed with no down time

#### 4.2.4 Performance

- Scan QR: EAMS needs under 3 seconds to detect QR and process participant information
- EAMS can check in, check out an event which has up to 1000 participants
- File Converter Service response time is about 10 seconds. This is limitation of file converting process.
- Other common response time is under 500ms

Note: response time is estimated based on recommended hardware specifications

#### 4.2.5 Security

- EAMS is divided into 2 roles: student and staff. Each role can only access its own group of functions
- EAMS always checks authorization before executing a function
- EAMS uses cloud services, which minimizes threats or unexpected problems (hardware, network failure..)

#### 4.2.6 Maintainability

- EAMS is divided into components therefore every component can be modified, maintained without affect the entire system, based on the necessary of incremental release
- EAMS applies repository pattern, dependency injection which are popular in modern web frameworks
- Clear code conventions help source code easy to read and maintain

#### 4.2.7 Portability

- Admin and Student applications are web applications therefore it can take advantage of cross-platform capabilities
- The entire EAMS system is developed to deploy using container mechanism. Therefore it can be deployed effortlessly regardless of server OS

#### 4.2.8 Scalability

- EAMS can quickly scale to adapt to unexpected demands in less than an hour

#### 4.2.9 Design Constraints

- Client modules is developed using ReactJS library
- Web service is developed using Node.js, use its event-driven architecture advantages
- Entire system uses TypeScripts to increase maintainability

#### 4.2.10 Purchased Components

- DigitalOceanKubernetes and Droplets: 20\$ per month (scale to demands)
- Firebase Storage: limit 5GB storage total, 50000 requests or 1GB bandwidth, per day
- Cloudmersive: limit 800 requests per month

## IV. Software Design Description

### 1. Overall Description

#### 1.1 Assumptions

This system is designed basing on these following assumptions:

- Windows 10, MacOS Catalina

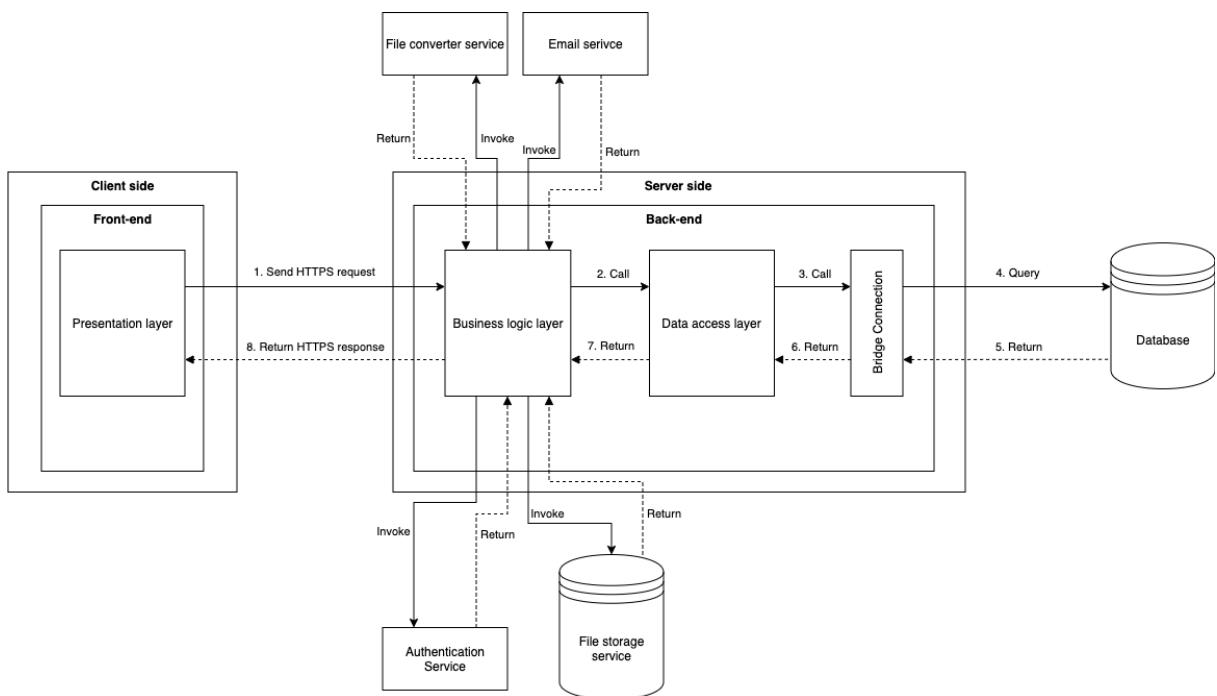
- Android 8.0, iOS 10
- Chrome 70
- MySQL 5.7
- End-user: university's student affairs department and university's students
- User requirements may change unexpectedly
- Detailed design is flexible to quickly adapt user's requirement

## 1.2 Design Constraints

This system should be compiled with following items:

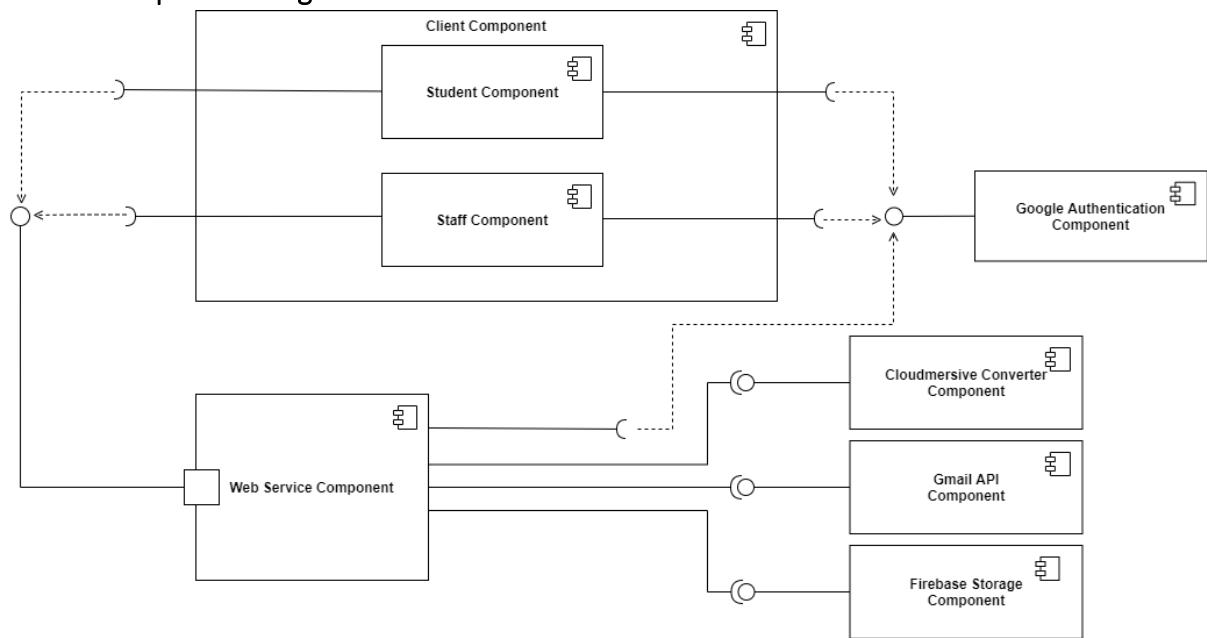
- End-user environment: any OS with Chrome or Safari browser
- Every process in this system must be less than 10 seconds
- Security requirement: prevent user from cheating as much as possible
- Network communication: this system depends on network connection

## 2. System Architecture Design



**Figure 92. Architecture Diagram**

### 3. Component Diagram



**Figure 93. Component Diagram**

## 4. Activity Diagram

### 4.1 Register event

Summary: This diagram illustrates the process of a student registering to participate an event

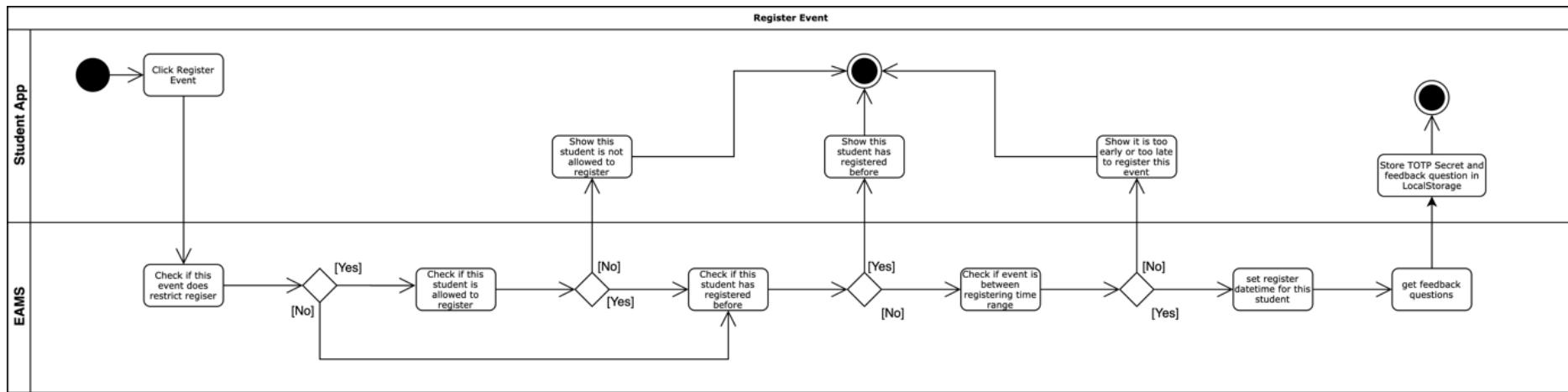


Figure 94. <Activity Diagram> Register Event

## 4.2 Check-in event

Summary: This diagram illustrates the process of a student checking in for an event

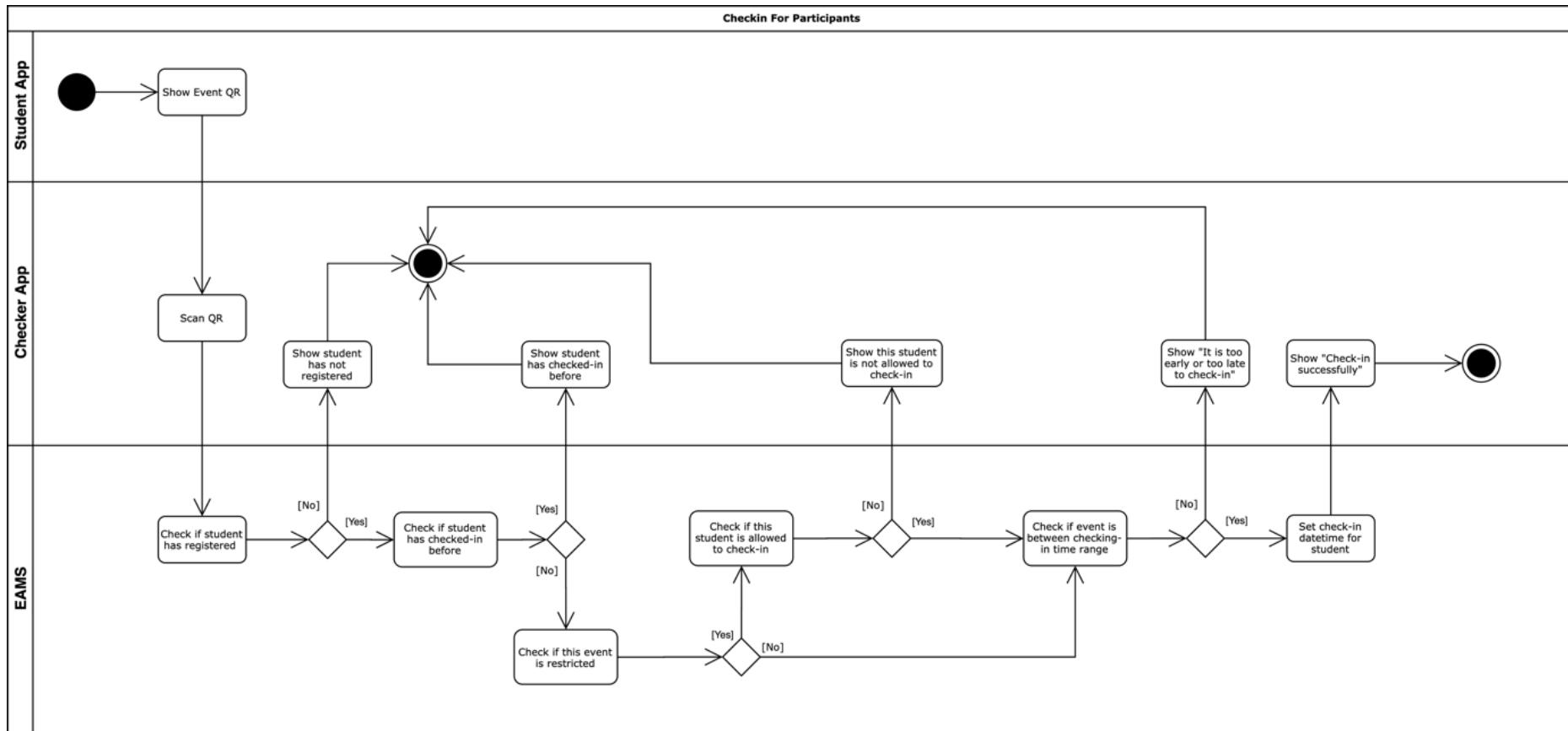


Figure 95. <Activity Diagram> Check-in Event



#### 4.1 Manage Event Proposals

Summary: This diagram shows the process of an event proposal from when they are created by a student to when it's graded by a staff

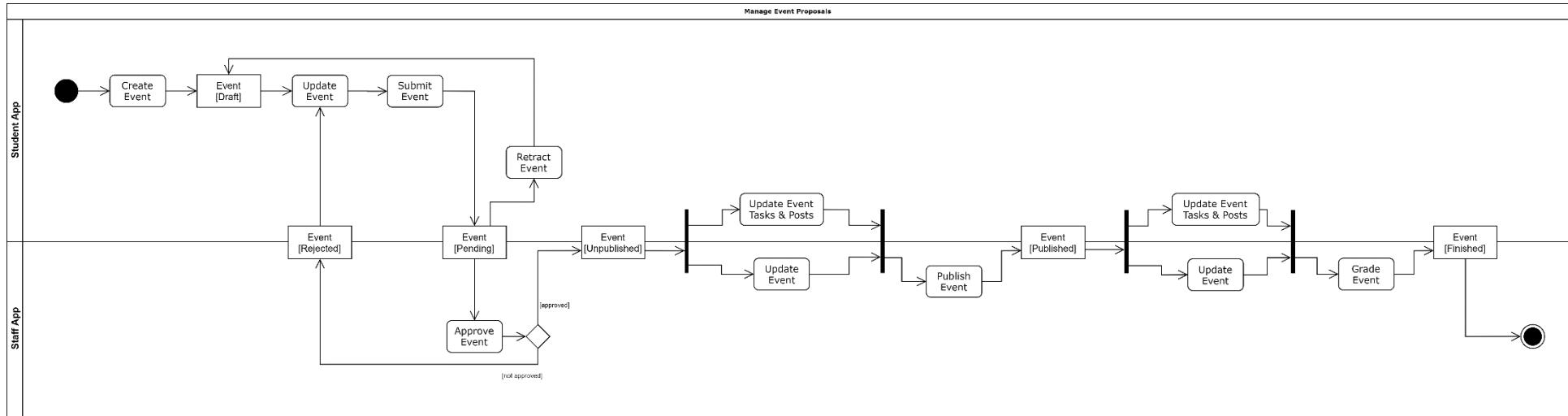


Figure 96. <Activity Diagram> Manage Event Proposals

## 5. System Detailed Design

### 5.1 Class Diagram

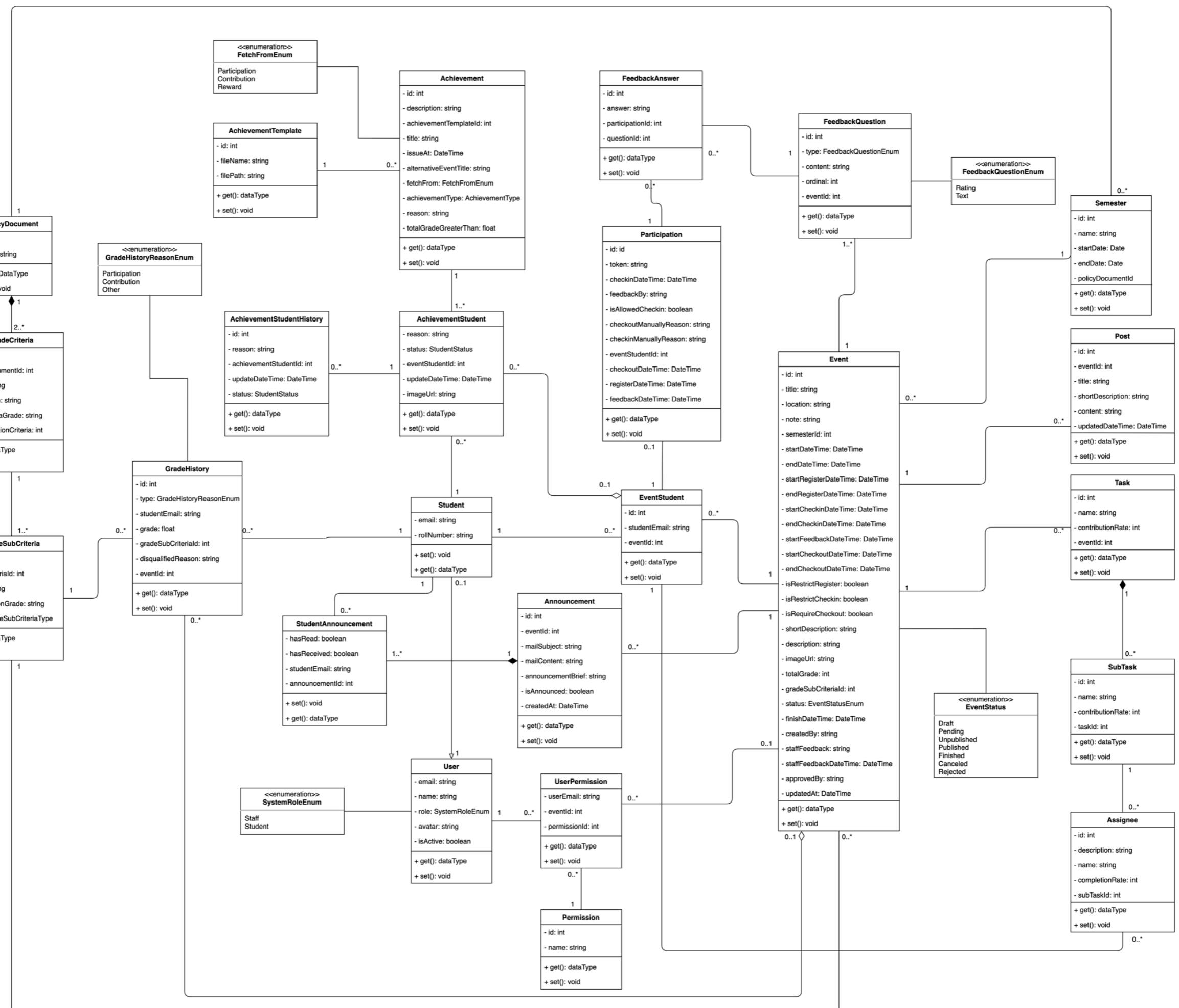


Figure 97. Class Diagram

## 5.2 Class Specification

### 5.2.1 PolicyDocument Class

Field name	Type	Description
id	int	Unique identifier of a policy document
name	string	The given name of a document

**Table 80.** *<Class Diagram Attributes> PolicyDocument*

Method	Return type	Description
get	Data type	Get attribute value
set	void	Set value of attribute

**Table 81.** *<Class Diagram Methods> PolicyDocument*

### 5.2.2 GradeCriteria Class

Field name	Type	Description
id	int	Unique identifier of a grade criteria
policyDocumentId	int	The id of the document that a criteria belongs to
name	string	Name of the criteria
description	string	Description of the criteria
maxCriteriaGrade	string	The grade that is used to calculate the capped grade of students in order to assess them
isContributionCriteria	int	Check if a criteria is a contribution criteria

**Table 82.** *<Class Diagram Attributes> GradeCriteria*

Method	Return type	Description
get	Data type	Get attribute value
set	void	Set value of attribute

**Table 83.** *<Class Diagram Methods> GradeCriteria*

### 5.2.3 GradeSubCriteria Class

Field name	Type	Description
Id	int	Unique identifier of a sub-criteria
gradeCriteriaId	int	The id of the criteria that this sub-criteria belongs to
Name	string	The name of the sub-criteria
participationGrade	string	The maximum grade that student can receive while participating an event of this criteria
type	GradeSubCriteriaType	Identifying type of grade sub criteria

**Table 84.** *<Class Diagram Attributes> GradeSubCriteria*

Method	Return type	Description
get	Data type	Get attribute value
set	void	Set value of attribute

**Table 85.** *<Class Diagram Methods> GradeSubCriteria*

5.2.4 GradeHistory Class

Field name	Type	Description
id	Int	Unique identifier of a
type	GradeHistoryReasonEnum	Type of history record
studentEmail	String	Email of student that is graded
Grade	Float	The grade student receives for the history
gradeSubCriteriaId	Int	The identifier of the sub-criteria that is used to give grade
disqualifiedReason	String	The reason why student is disqualified for the history
eventId	Int	The event in which student participated to have a this history record

**Table 86.** *<Class Diagram Attributes> GradeHistory*

Method	Return type	Description
get	Data type	Get attribute value
set	Void	Set value of attribute

**Table 87.** *<Class Diagram Methods> GradeHistory*

5.2.5 AchievementTemplate Class

Field name	Type	Description
id	int	Unique identifier of an achievement template
fileName	string	File name of the template
filePath	string	File path of the template

**Table 88.** *<Class Diagram Attributes> AchievementTemplate*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 89.** *<Class Diagram Methods> AchievementTemplate*

5.2.6 Achievement Class

Field name	Type	Description
id	int	Unique identifier of an achievement
description	string	Description of an achievement
achievementTemplateId	int	The identifier of the template that the achievement uses
title	string	The title of the achievement
issueAt	DateTime	The date time when the achievement is issued
alternativeEventTitle	string	Title that is shown on the achievement file
fetchFrom	FetchFromEnum	Indicates where the students list was fetched to get the receivers of the achievement
reason	string	Reason that is shown on the achievement file

totalGradeGreaterThan	float	The filter that is used to fetch students list to grant achievement
-----------------------	-------	---

**Table 90.** *<Class Diagram Attributes> Achievement*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 91.** *<Class Diagram Methods> Achievement*

#### 5.2.7 FetchFromNum Class

Field name	Type	Description
Participation	string	Indicates the staff fetch participants of an event to grant achievements
Contribution	string	Indicates the staff fetch contributors of an event to grant achievements
Reward	string	Indicates the staff manually add students to grant achievements

**Table 92.** *<Enumeration> FetchFromEnum*

#### 5.2.8 AchievementStudentHistory Class

Field name	Type	Description
id	int	Unique identifier of a history of an achievement of a student
reason	string	The reason why students are disqualified or qualified for an achievement
achievementStudentId	int	The original record of an achievement of a student
updateDateTime	DateTime	The date time when this record gets inserted into the original record
status	StudentStatus	The status of an achievement of a student

**Table 93.** *<Class Diagram Attributes> AchievementStudentHistory*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 94.** *<Class Diagram Methods> AchievementStudentHistory*

#### 5.2.9 AchievementStudent Class

Field name	Type	Description
reason	string	The reason why a student is qualified or disqualified for an achievement
status	studentStatus	Current status of an achievement of a student
eventStudentId	int	Save the information of the event that that student participated or contributed to in order to receive an achievement
updateDateTime	DateTime	The date time when this record get inserted

imageUrl	string	The image url of the merged achievement
----------	--------	---

**Table 95.** *<Class Diagram Attributes> AchievementStudent*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 96.** *<Class Diagram Methods> AchievementStudent*

#### 5.2.10 Student Class

Field name	Type	Description
email	int	Email of a student
rollNumber	int	Roll number of a student

**Table 97.** *<Class Diagram Attributes> Student*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 98.** *<Class Diagram Methods> Student*

#### 5.2.11 StudentAnnouncement Class

Field name	Type	Description
hasRead	int	Indicates whether or not an announcement is read by a student
hasReceived	int	Indicates whether or not an announcement is received by a student
studentEmail	string	The student whom an announcement is sent to
announcementId	string	The announcement identifier which the announcement is sent

**Table 99.** *<Class Diagram Attributes> StudentAnnouncement*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 100.** *<Class Diagram Methods> StudentAnnouncement*

#### 5.2.12 Announcement Class

Field name	Type	Description
id	int	Unique identifier of an announcement
eventId	int	The event which the announcement is about
mailSubject	string	Subject of mail that is sent to group of people in the event.
mailContent	string	Mail's content that is sent to group of people in the event.
announcementBrief	string	The brief of announcement that send to people in the event

isAnnounced	boolean	Indicate announcement is sent or not.
createdAt	DateTime	The date time when an announcement is created.

**Table 101.** *<Class Diagram Attributes> Announcement*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 102.** *<Class Diagram Methods> Announcement*

#### 5.2.13 User Class

Field name	Type	Description
email	String	Email of a user
name	string	Fullname of a user
role	SystemRoleEnum	Role of a user in the system
avatar	string	Image that represents a user
isActive	boolean	The active status of a user

**Table 103.** *<Class Diagram Attributes> User*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 104.** *<Class Diagram Methods> User*

#### 5.2.14 SystemRoleEnum Class

Field name	Type	Description
Staff	string	Indicates that user is a staff
Student	string	Indicates that user is a student

**Table 105.** *<Enumeration> SystemRoleEnum*

#### 5.2.15 UserPermission Class

Field name	Type	Description
userEmail	int	The email of the user that a permission belongs to
eventId	int	The event which the permission is related to
permissionId	string	The

**Table 106.** *<Class Diagram Attributes> UserPermission*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 107.** *<Class Diagram Methods> UserPermission*

#### 5.2.16 Permission Class

Field name	Type	Description
Id	int	Unique identifier of a permission

name	string	Name of a permission
------	--------	----------------------

**Table 108. <Class Diagram Attributes> Permission**

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 109. <Class Diagram Methods> Permission**

#### 5.2.17 EventStudent Class

Field name	Type	Description
Id	int	Unique identifier of student's participation or contribution in an event
studentEmail	string	Student's email
eventId	int	Subject of mail that send to group of people in the event.

**Table 110. <Class Diagram Attributes> EventStudent**

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 111. <Class Diagram Methods> EventStudent**

#### 5.2.18 Participation Class

Field name	Type	Description
Id	int	Unique identifier of student's participation in an event
token	string	
checkinDateTime	DateTime	The date time when student check-in
feedbackBy	string	If students are not able to give feedbacks using the app (internet connection loss, out of battery), they will feedback on papers and staff would help them manually feedback. This is where staff's who does the manual feedback is saved
isAllowedCheckin	boolean	Determines if a student is allowed to check-in on restricted events
checkoutManuallyReason	string	Reason for student's checking out manually
checkinManuallyReason	string	Reason for student's checking in manually
eventStudentId	int	Unique identifier of student's participation in an event
checkoutDateTime	DateTime	The date and time when a student check-out
registerDateTime	DateTime	The date and time when a student register
feedbackDateTime	DateTime	The date and time when a student give feedbacks

**Table 112. <Class Diagram Attributes> Participation**

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 113.** *<Class Diagram Methods> Participation*

#### 5.2.19 FeedbackAnswer Class

Field name	Type	Description
Id	int	Unique identifier of a student's feedback answer
answer	string	The student's answer for the feedback question
participationId	int	The unique identifier for student's participations
questionId	int	Determine which question a student is answering for feedback

**Table 114.** *<Class Diagram Attributes> FeedbackAnswer*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 115.** *<Class Diagram Methods> FeedbackAnswer*

#### 5.2.20 FeedbackQuestion Class

Field name	Type	Description
id	int	Unique identifier of a feedback question
type	FeedbackQuestionEnum	Type of a feedback question
content	string	Content of a question
ordinal	string	The order of a question
eventId	string	Identifier of the event that this feedback question belongs to

**Table 116.** *<Class Diagram Attributes> FeedbackQuestion*

Method	Return type	Description
get	datatype	Get attribute value
set	void	Set value of attribute

**Table 117.** *<Class Diagram Methods> FeedbackQuestion*

#### 5.2.21 FeedbackQuestionEnum Class

Field name	Type	Description
Rating	string	The question type which student answers by giving a rate from 1 to 5
Text	string	The question type which student answers by entering a textual answer

**Table 118.** *<Enumeration> FeedbackQuestionEnum*

### 5.2.22 Event Class

Field name	Type	Description
id	int	Unique identifier of an event
title	string	The title of an event
location	string	The location where an event is held
note	string	The additional notes that students should know when coming to an event
semesterId	int	The identifier of the semester that this event belongs to
startTime	DateTime	The date time when an event starts
endTime	DateTime	The date time when an event ends
startRegisterDateTime	DateTime	The date time when an event starts allowing registration
endRegisterDateTime	DateTime	The date time when an event ends allowing registration
startCheckinDateTime	DateTime	The date time when an event starts allowing students to check in
endCheckinDateTime	DateTime	The date time when an event ends allowing students to check in
startFeedbackDateTime	DateTime	The date time when an event starts allowing students to give feedback
endFeedbackDateTime	DateTime	The date time when an event ends allowing students to give feedback
startCheckoutDateTime	DateTime	The date time when an event starts allowing students to check out
endCheckoutDateTime	DateTime	The date time when an event stops allowing students to check in
isRestrictRegister	boolean	Determines if an event restricts registration
isRestrictCheckin	boolean	Determines if an event restricts checking in
isRequiredCheckout	boolean	Determines if an event requires checking out
shortDescription	string	A brief description of an event
description	string	A full description of an event
imageUrl	string	A URL for the image represents an event
totalGrade	int	Total contribution grade of an event
gradeSubCriteriaId	int	The identifier of the sub-criteria that applies to an event
status	EventStatusEnum	The current state of an event
finishDatetime	DateTime	The datetime when an event is graded finished
createdBy	string	The email of the creator of an event
staffFeedback	string	The feedback of a staff in case a student's plan gets approved or rejected
staffFeedbackDateTime	DateTime	The datetime when a staff feedback
approvedBy	string	Email of the staff that approves an event
updatedAt	DateTime	The date time when an event gets updated

**Table 119. <Class Diagram Attributes> Event**

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 120. <Class Diagram Methods> Event**

#### 5.2.23 EventStatus Class

Field name	Type	Description
Draft	string	An event created by a student but not submitted yet. Only the owner can see this type of events
Pending	string	An event created by a student that's been submitted but not approved yet.
Unpublished	string	An event that has just been created by a staff or created by a student and approved by a staff.
Published	string	An event that is published for every student in the system
Finished	string	An event that is graded by a staff
Cancelled	string	An event that was published but then cancelled before start register date time
Rejected	string	An event created by a student but rejected by a staff

**Table 121. <Enumeration> EventStatus**

#### 5.2.24 Semester Class

Field name	Type	Description
id	int	Unique identifier of a semester
name	int	Name of a semester
startDate	string	The start date of a semester
endDate	string	The end date of a semester
policyDocumentId	string	The policy document identifier a semester applies

**Table 122. <Class Diagram Attributes> Semester**

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 123. <Class Diagram Methods> Semester**

#### 5.2.25 Post Class

Field name	Type	Description
id	int	Unique identifier of a post
eventId	int	The event identifier that a post belongs to
title	string	The title of a post
shortDescription	string	The short description of a post
content	string	Content of a post
updatedDateTime	boolean	The date time when a post gets updated

**Table 124.** *<Class Diagram Attributes> Post*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 125.** *<Class Diagram Methods> Post*

#### 5.2.26 Task Class

Field name	Type	Description
id	int	Unique identifier of a task
name	int	The name of a task
contributionRate	string	The contribution rate of a task to an event
eventId	string	The identifier of the event that a task contributes to

**Table 126.** *<Class Diagram Attributes> Task*

Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 127.** *<Class Diagram Methods> Task*

#### 5.2.27 SubTask Class

Field name	Type	Description
id	int	Unique identifier of a sub-task
name	string	Name of a sub-task
contributionRate	int	The contribution rate of a subtask that contributes to a task
taskId	int	The task that a sub-task contributes to

**Table 128.** *<Class Diagram Attributes> SubTask*

Method	Return type	Description
get	dataType	Get attribute value
set	Void	Set value of attribute

**Table 129.** *<Class Diagram Methods> SubTask*

#### 5.2.28 Assignee Class

Field name	Type	Description
id	int	Unique identifier of an assignee
completionRate	string	The completion rate of an assignee to a task
subTaskId	string	The task that an assignee contributes to

**Table 130.** *<Class Diagram Attributes> Assignee*

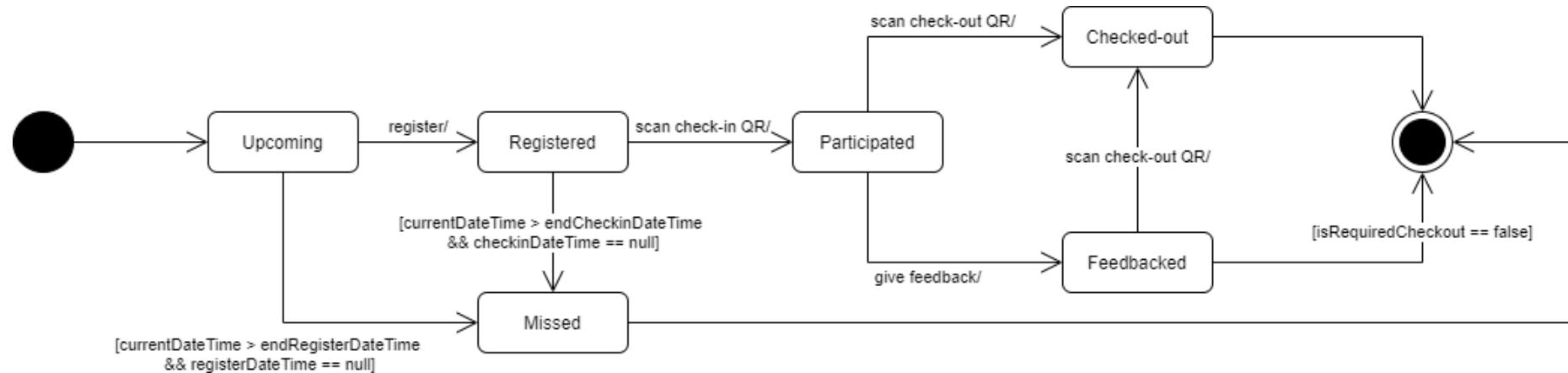
Method	Return type	Description
get	dataType	Get attribute value
set	void	Set value of attribute

**Table 131. <Class Diagram Methods> Assignee**

## 5.3 State Machine Diagrams

### 5.3.1 Event Participation

Summary: This diagram illustrates the state of a student when he or she participates an event



**Figure 98. <State Machine Diagram> Event Participation**

### 5.3.2 Event State Machine

Summary: This diagram illustrates the status of an event in the system

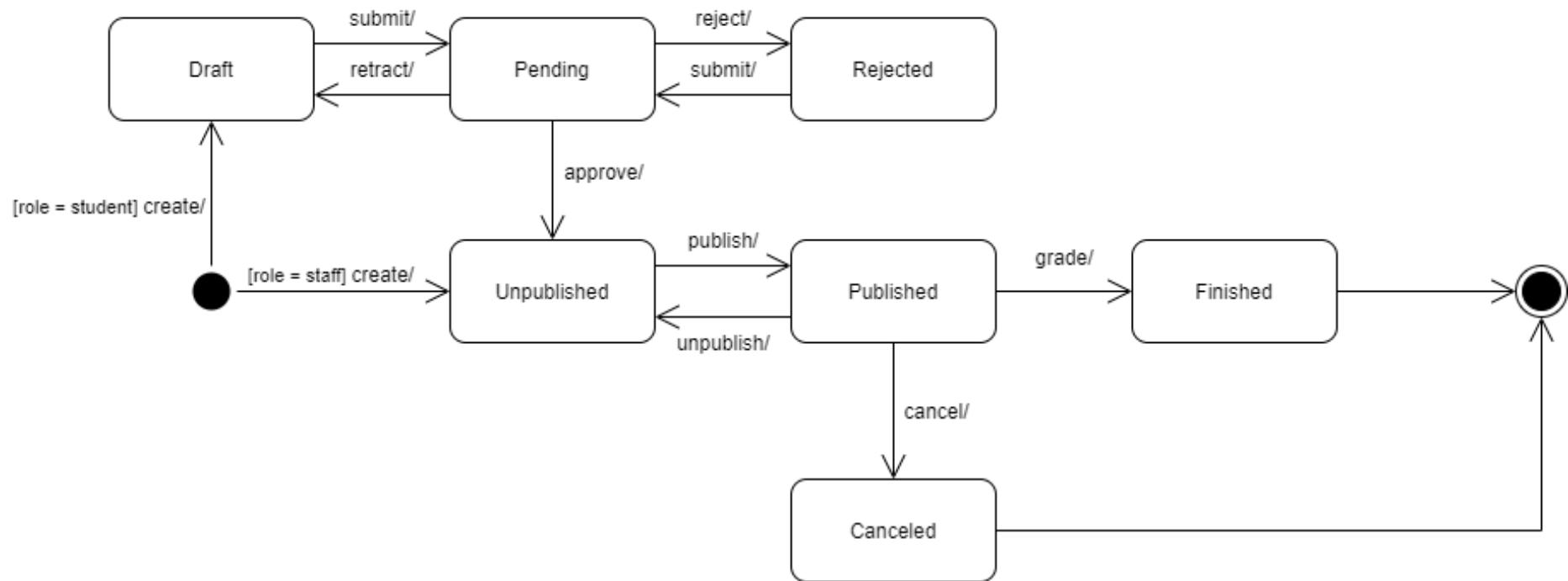
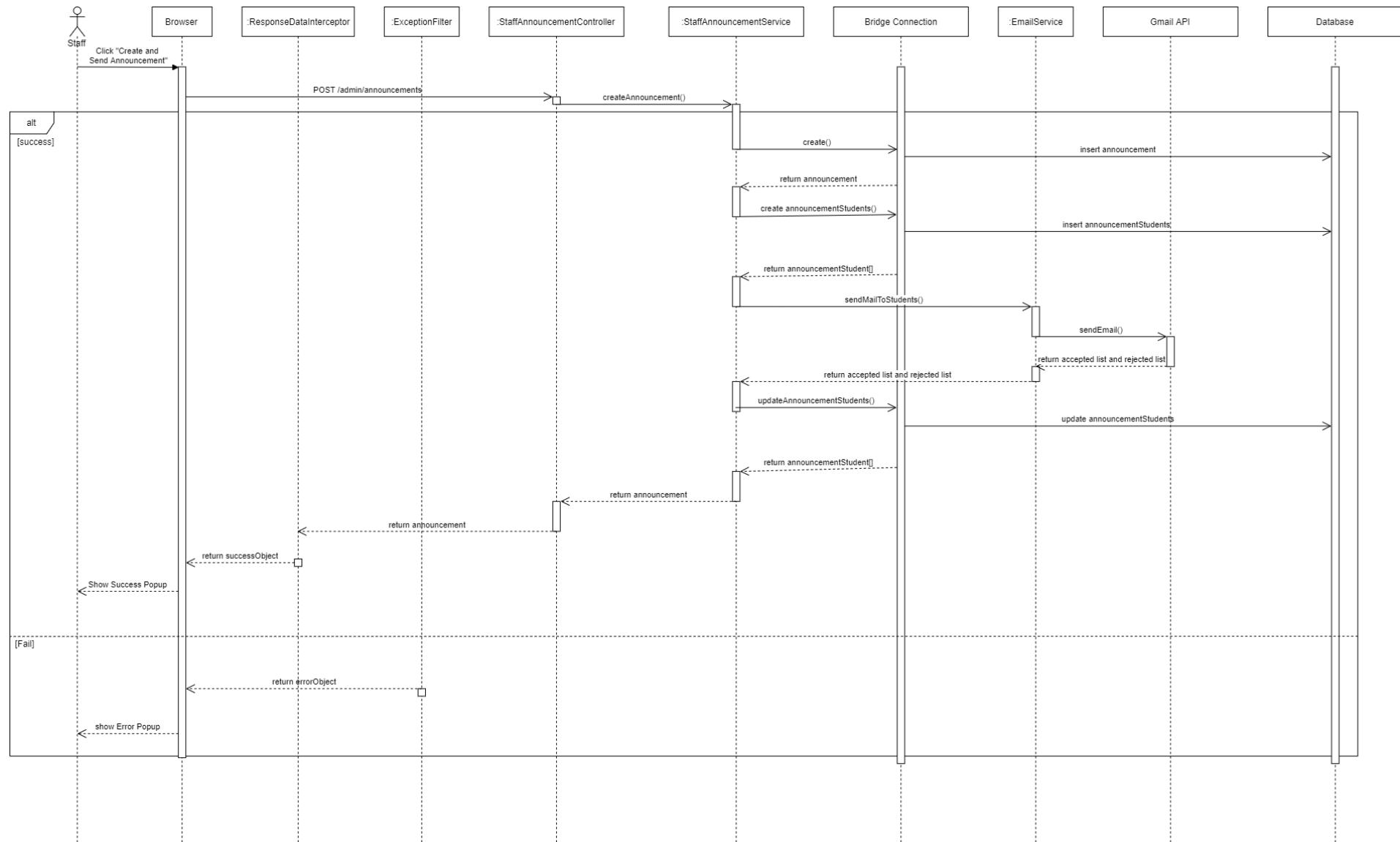


Figure 99. <State Machine Diagram> Event State

## 5.4 Sequence Diagrams

#### 5.4.1 Create and Send Announcement

Summary: This diagram illustrates the interactions that occur in the system when a Staff triggers Create and Send Announcement Action



**Figure 100. <Sequence Diagram> Create and send announcement**

#### 5.4.2 Event Grading

Summary: This diagram illustrates the interactions that occur in the system when a Staff triggers finish event

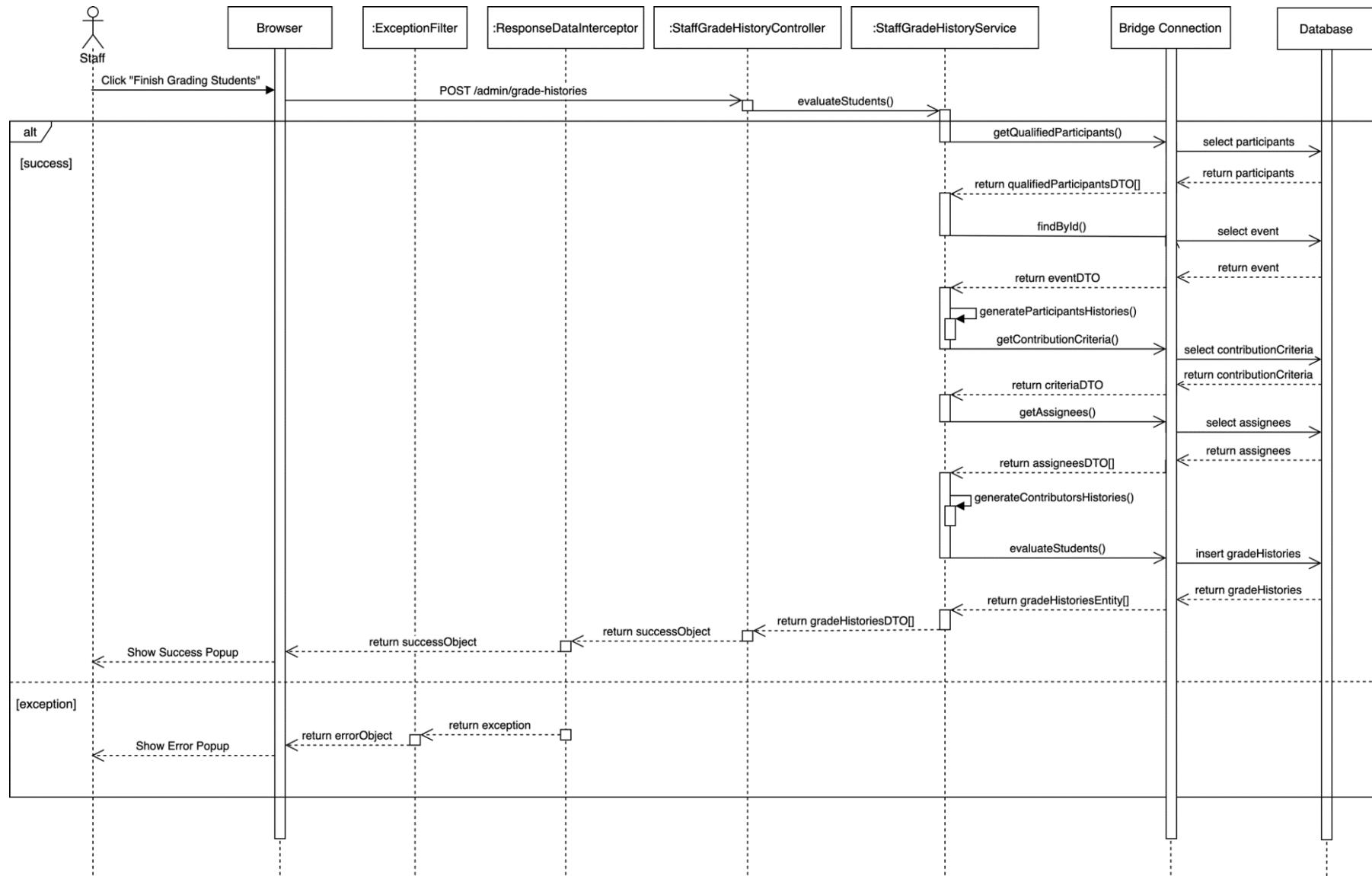


Figure 101. <Sequence Diagram> Event Grading

## 6. Data & Database Design

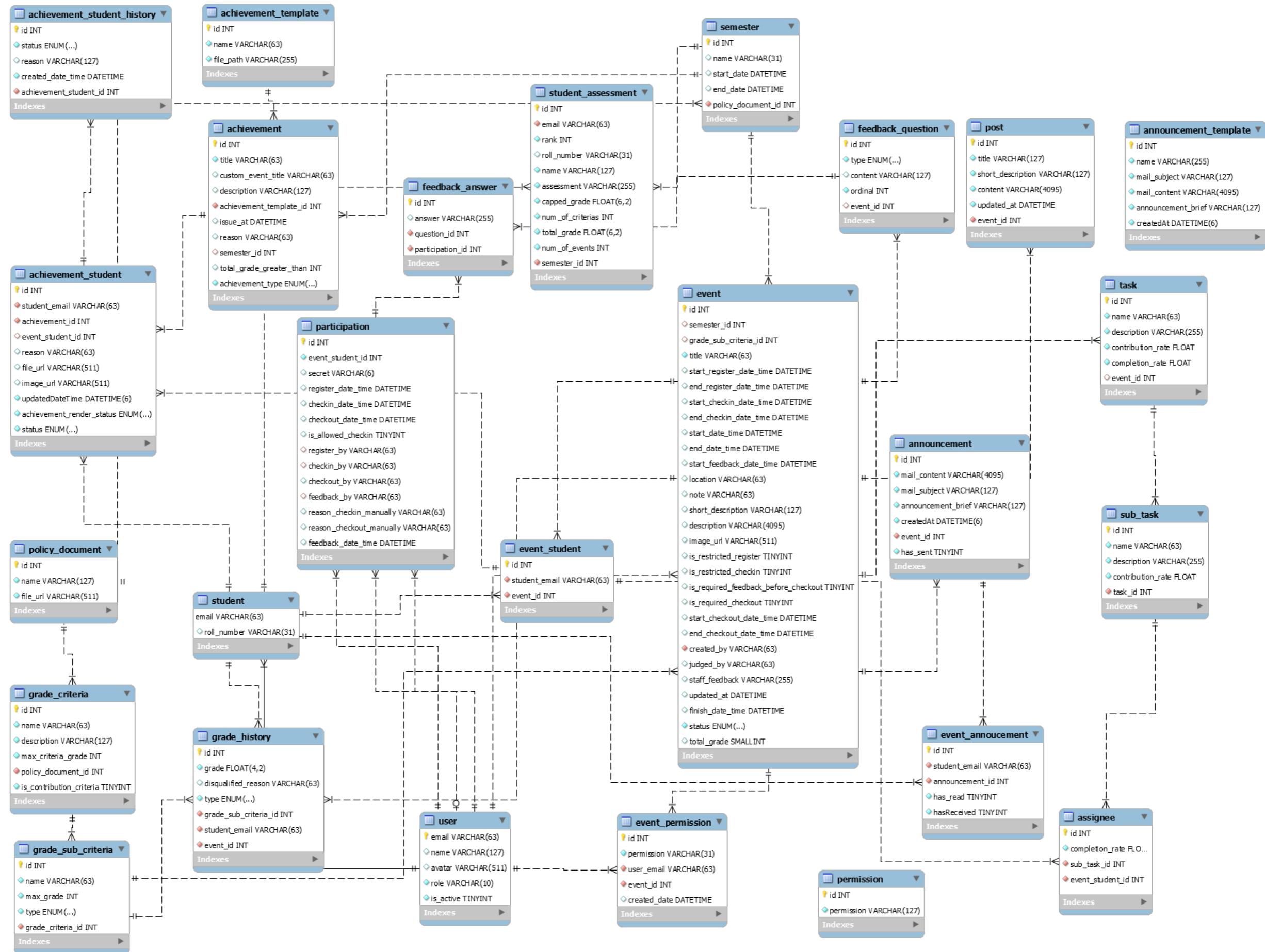


Figure 102. Physical Diagram

## 6.1 Database Design

### 6.1.1 policy\_document

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
name	VARCHAR(127)	127*4	No	No	
file_url	VARCHAR(511)	511*4	No	No	

*Table 132. <Physical Diagram> policy\_document*

### 6.1.2 grade\_criteria

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
name	VARCHAR(63)	63*4	No	No	
description	VARCHAR(127)	127*4	No	No	
max_criteria_grade	INT	4	No	No	
policy_document_id	INT	4	No	No	FK
is_contribution_criteria	TINYINT	1	No	No	

*Table 133. <Physical Diagram> grade\_criteria*

### 6.1.3 grade\_sub\_criteria

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
name	VARCHAR(63)	63*4	No	No	
max_grade	INT	4	No	No	
type	ENUM	2	No	No	
grade-criteria_id	INT	4	No	No	FK

*Table 134. <Physical Diagram> grade\_sub\_criteria*

### 6.1.4 grade\_history

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
grade	FLOAT(4,,2)	4	No	No	
disqualified_reason	VARCHAR(63)	63*4	No	Yes	
type	ENUM	2	No	No	
grade_sub_criteria_id	INT	4	No	No	FK
student_email	VARCHAR(63)	63*4	No	No	FK
event_id	INT	4	No	No	FK

*Table 135. <Physical Diagram> grade\_history*

### 6.1.5 student\_assessment

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
email	VARCHAR(63)	63*4	No	No	FK
rank	INT	4	No	No	
roll_number	VARCHAR(31)	31*4	Yes	No	

name	VARCHAR(127)	127*4	No	No	
assessment	VARCHAR(255)	255*4	No	No	
capped_grade	FLOAT(6,2)	4	No	No	
num_of_criterias	INT	4	No	No	
total_grade	FLOAT(6,2)	4	No	No	
num_of_events	INT	4	No	No	
semester_id	INT	4	No	No	FK

**Table 136.** *<Physical Diagram> student\_assessment*

#### 6.1.6 achievement\_template

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
name	VARCHAR(63)	63*4	No	No	
file_path	VARCHAR(255)	255*4	No	No	

**Table 137.** *<Physical Diagram> achievement\_template*

#### 6.1.7 achievement\_student\_history

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
status	ENUM	2	No	No	
reason	VARCHAR(127)	127*4	No	Yes	
created_date_time	DATETIME	8	No	No	
achievement_student_id	INT	4	No	No	FK

**Table 138.** *<Physical Diagram> achievement\_student\_history*

#### 6.1.8 student

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
email	VARCHAR(63)	63*4	Yes	No	PK
roll_number	VARCHAR(31)	31*4	Yes	No	

**Table 139.** *<Physical Diagram> student*

#### 6.1.9 announcement\_template

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
name	VARCHAR(255)	255*4	No	No	
mail_subject	VARCHAR(127)	127*4	No	No	
mail_content	VARCHAR(4095)	4095*4	No	No	
announcement_brief	VARCHAR(127)	127*4	No	No	
createdAt	DATETIME(6)	8	No	No	

**Table 140.** *<Physical Diagram> announcement\_template*

#### 6.1.10 achievement

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK

id	INT	4	Yes	No	PK
title	VARCHAR(63)	63*4	No	No	
custom_event_title	VARCHAR(63)	63*4	No	Yes	
description	VARCHAR(127)	127*4	No	No	
achievement_template_id	INT	4	No	No	FK
issue_at	DATETIME	8	No	No	
reason	VARCHAR(63)	63*4	No	No	
semester_id	INT	4	No	No	FK
total_grade_greater_than	INT	4	No	No	
achievement_type	ENUM	2	No	No	

**Table 141. <Physical Diagram> achievement**

#### 6.1.11 achievement\_student

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
student_email	VARCHAR(63)	63*4	No	No	FK
achievement_id	INT	4	No	No	FK
event_student_id	INT	4	No	Yes	FK
reason	VARCHAR(63)	63*4	No	No	
file_url	VARCHAR(511)	511*4	No	Yes	
image_url	VARCHAR(511)	511*4	No	Yes	
updatedDateTime	DATETIME	8	No	No	
achievement_render_status	ENUM	2	No	No	
status	ENUM	2	No	No	

**Table 142. <Physical Diagram> achievement\_student**

#### 6.1.12 participation

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
event_student_id	INT	4	Yes	No	
secret	VARCHAR(6)	6*4	No	Yes	
register_date_time	DATETIME	8	No	Yes	
checkin_date_time	DATETIME	8	No	Yes	
checkout_date_time	DATETIME	8	No	Yes	
is_allowed_checkin	TINYINT	1	No	Yes	
register_by	VARCHAR(63)	63*4	No	Yes	FK
checkin_by	VARCHAR(63)	63*4	No	Yes	FK
checkout_by	VARCHAR(63)	63*4	No	Yes	FK
feedback_by	VARCHAR(63)	63*4	No	Yes	FK
reason_checkin_manually	VARCHAR(63)	63*4	No	Yes	
reason_checkout_manually	VARCHAR(63)	63*4	No	Yes	
feedback_date_time	DATETIME	8	No	Yes	

**Table 143.** *<Physical Diagram> participation*

6.1.13 event\_student

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
student_email	VARCHAR(63)	63*4	No	No	FK
event_id	INT	4	No	No	FK

**Table 144.** *<Physical Diagram> event\_student*

6.1.14 announcement

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
mail_context	VARCHAR(4095)	4095*4	No	No	
mail_subject	VARCHAR(127)	127*4	No	No	
announcement_brief	VARCHAR(127)	127*4	No	No	
createdAt	DATETIME(6)	8	No	No	
event_id	INT	4	No	No	FK
has_sent	TINYINT	1	No	No	

**Table 145.** *<Physical Diagram> announcement*

6.1.15 event\_announcement

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
student_email	VARCHAR(63)	63*4	No	No	FK
announcement_id	INT	4	No	No	FK
has_read	TINYINT	1	No	No	
hasReceived	TINYINT	1	No	No	

**Table 146.** *<Physical Diagram> event\_announcement*

6.1.16 user

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
email	VARCHAR(63)	63*4	Yes	No	PK
name	VARCHAR(127)	127*4	No	No	
avatar	VARCHAR(511)	511*4	No	Yes	
role	VARCHAR(10)	10*4	No	No	
is_active	TINYINT	1	No	No	

**Table 147.** *<Physical Diagram> user*

6.1.17 feedback\_answer

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
answer	VARCHAR(255)	255*4	No	No	
question_id	INT	4	No	No	FK
participation_id	INT	4	No	No	FK

**Table 148. <Physical Diagram> feedback\_answer**

6.1.18 event

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
semester_id	INT	4	No	No	FK
grade_sub_criteria_id	INT	4	No	No	FK
title	VARCHAR(63)	63*4	No	No	
start_register_date_time	DATETIME	8	No	No	
end_register_date_time	DATETIME	8	No	No	
start_checkin_date_time	DATETIME	8	No	No	
end_checkin_date_time	DATETIME	8	No	No	
start_date_time	DATETIME	8	No	No	
end_date_time	DATETIME	8	No	No	
start_feedback_date_time	DATETIME	8	No	No	
location	VARCHAR(63)	63*4	No	No	
note	VARCHAR(63)	63*4	No	Yes	
short_description	VARCHAR(127)	127*4	No	No	
description	VARCHAR(4095)	4095*4	No	No	
image_url	VARCHAR(1023)	1023*4	No	No	
is_restricted_register	TINYINT	1	No	No	
is_restricted_checkin	TINYINT	1	No	No	
is_required_feedback_before_checkout	TINYINT	1	No	No	
is_required_checkout	TINYINT	1	No	No	
start_checkout_date_time	DATETIME	8	No	Yes	
end_checkout_date_time	DATETIME	8	No	Yes	
created_by	VARCHAR(63)	63*4	No	No	FK
judged_by	VARCHAR(63)	63*4	No	Yes	
staff_feedback	VARCHAR(255)	255*4	No	Yes	
updated_at	DATETIME	8	No	No	
finish_date_time	DATETIME	8	No	Yes	
status	ENUM	2	No	No	
total_grade	SMALLINT	2	No	No	

**Table 149. <Physical Diagram> event**

6.1.19 event\_permission

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
permission	VARCHAR(127)	127*4	No	No	
user_email	VARCHAR(63)	63*4	No	No	FK
event_id	INT	4	No	No	FK
created_date	DATETIME	8	No	No	

**Table 150. <Physical Diagram> event\_permission**

#### 6.1.20 post

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	8	Yes	No	PK
title	VARCHAR(127)	127*4	No	No	
short_description	VARCHAR(127)	127*4	No	No	
context	VARCHAR(4095)	4095*4	No	No	
updated_at	DATETIME	8	No	No	
event_id	INT	4	No	No	FK

**Table 151.** *<Physical Diagram> post*

#### 6.1.21 task

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
name	VARCHAR(63)	63*4	No	No	
description	VARCHAR(255)	255*4	No	No	
contribution_rate	FLOAT	4	No	No	
completion_rate	FLOAT	4	No	Yes	
event_id	INT	4	No	No	FK

**Table 152.** *<Physical Diagram> task*

#### 6.1.22 sub\_task

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
name	VARCHAR(63)	63*4	No	No	
description	VARCHAR(255)	255*4	No	Yes	
contribution_rate	FLOAT	4	No	No	
task_id	INT	4	No	No	FK

**Table 153.** *<Physical Diagram> sub\_task*

#### 6.1.23 assignee

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
completion_rate	FLOAT	4	No	No	
sub_task_id	INT	4	No	No	FK
event_student_id	INT	4	No	No	FK

**Table 154.** *<Physical Diagram> assignee*

#### 6.1.24 semester

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
name	VARCHAR(31)	31*4	No	No	
start_date	DATETIME	8	No	No	FK
end_date	DATETIME	8	No	No	

policy_document_id	INT	4	No	No	FK
--------------------	-----	---	----	----	----

**Table 155. <Physical Diagram> semester**

#### 6.1.25 feedback\_question

Field name	Type	Size (bytes)	Unique	Nullable	PK/FK
id	INT	4	Yes	No	PK
type	ENUM	2	No	No	
content	VARCHAR(127)	127*4	No	No	FK
ordinal	INT	4	No	No	
event_id	INT	4	No	No	FK

**Table 156. <Physical Diagram> feedback\_question**

### 6.2 Data File Design

#	File Name	Type	Notes
1	assessment-grade.json	json	Define range of grade to assess student
2	feedback-report.xlsx	xlsx	Template to export feedback report
3	firebase-admin-upload-service.json	json	Firebase storage configuration
4	Participations-report.xlsx	Xlsx	Template to export participations report
5	Registrants-report.xlsx	Xlsx	Template to export registrants report
6	Semester-report.xlsx	Xlsx	Template to export semester report
7	Upload-template.xlsx	Xlsx	Template to upload event's whitelist student
8	Upload-template-create-feedback-questions.xlsx	Xlsx	Template to upload event's feedback questions

**Table 157. Data File Design**

## 7. Algorithm

### 7.1 Calculate contribution grade

#### 7.1.1 Definition

Universities need to save information about students' contribution to events. Each event can include many tasks, which can be divided to subtasks in order to distribute to assignees. Students contribute in these sub-tasks to earn contribution grades.

#### 7.1.2 Define Problem

In order to record students' contribution grades, it is necessary to define a process for staffs to give contribution grades to student that is fair for everyone. This process needs to give staffs full control the grades that each student receives after one event, based on the tasks that they contribute to and how important that task is for an event

#### 7.1.3 Solution

EAMS offers a way to assign a contribution grade to an event, which represent the effort it takes to organize it. This contribution grade is then distributed to multiple tasks. This represent a part of an event that takes a lot of effort to complete. Each of the tasks are evaluated using "contribution rate". For example, an event can have 2 tasks: such as: "Public

Relations Division" and "Logistic Division", each contributes 60% and 40% to an event respectively.

Each task can be divided to sub-tasks, which represent the tasks that students needs to complete in order to achieve the tasks' goals. For example, sub-tasks of a "Public Relations" task can be "Write Facebook posts to attract participants" or "Give Pamphlets". Each of these sub-tasks also has contribution rate for finer control over grades.

Sub-tasks' grades are equally divided to all the students that are assigned to complete these sub-tasks. When the event finishes, staff determines how much of the sub-task a student completes in order to calculate the final grade that they receive

This can be illustrated in the following example

- An event has total contribution grade of 10.
- It has 2 tasks, each have its own sub-tasks, each sub-task has its own assignees:
  - Logistic Division: 40% contribution rate – 4 points
    - Transport equipment: 50% contribution rate – 2 points
      - 2 assignees: A and B – Each receives 1 point
    - Build the event's stage: 50% contribution rate – 2 points
      - 2 assignees: B and C – Each receives 1 point
  - Public Relations Division: 60% contribution rate – 6 points
    - Write Facebook posts: 100% contribution rate – 6 points
      - 1 assignee: A – receives 6 points
- When the event finishes, each assignee receive grades for sub-tasks
  - A
    - 1 point for "Transport equipment"
    - 6 points for "Write Facebook posts"
  - B
    - 1 point for "Transport equipment"
    - 1 point for "Build the event's stage"
  - C
    - 1 point for "Build the event's stage"
- However, staff monitor each person's performance and give completion rate for each of the grade from the sub-task that they get:
  - A
    - 1 point for "Transport equipment": Completion rate: 100%
    - 6 points for "Write Facebook posts": Completion rate: 50%
  - B
    - 1 point for "Transport equipment": Completion rate: 20%
    - 1 point for "Build the event's stage": Completion rate: 30%
  - C
    - 1 point for "Build the event's stage": Completion rate: 100%
- Finally, each student receives:
  - A – Total contribution grade: 4 points
    - 1 point for "Transport equipment":  $(1 * 100\%)$
    - 3 points for "Write Facebook posts":  $(6 * 50\%)$

- B – Total contribution grade: 0.5 point
  - 0.2 point for “Transport equipment”:  $(1 * 20\%)$
  - 0.3 point for “Build the event’s stage”:  $(1 * 30\%)$
- C – Total contribution grade: 1 point
  - 1 point for “Build the event’s stage”:  $(1 * 100\%)$
- Staff finishes event and everyone’s total contribution grade are saved to their histories.

The complexity of this method is  $O(n^3)$

## 7.2 Student assessment process and ranking

### 7.2.1 Definition

EAMS can calculate student extracurricular activity grade to output their assessment. There are 4 levels of assessment: excellent, good, average and need improvements

### 7.2.2 Define problem

Firstly, assessing student’s extracurricular by grade improve encourage them to be more active in university to get higher assessment.

Secondly, students usually have their strength and attention in a particular field. They prefer take parts in their preference. This makes student’s personal development does not comprehensive.

### 7.2.3 Solution

#### 7.2.3.1 Assessment

EAMS defines 4 grade ranges correspond 4 assessment level:

- Under 10 points: need improvement
- From 10 to under 55 points: average
- From 55 to under 100 points: good
- Above 100 point: excellent

Besides that, to prevent student from only participating one type of extracurricular activity, EAMS has a term named grade criteria. All events are classified to a criterion. Every criterion has a max grade student can get.

Criterion name	Grade per event	Max grade
Sports & Arts	5	25
Seminar, Workshop & Talkshows	10	50
Community, Experiences and Cultural	5	25
Others	5	25
Contribution	About 10	70

This means if a student only participates one type of event, they cannot get higher grade than its criterion max grade. To get excellent assessment, a student has to really active at least 2 or 3 different types of event.

### 7.2.3.2 Ranking

Student extracurricular activity ranking is quite more complex. As the source of semester rewards, EAMS defines some rules

Student ranking is measured by 4 fields, with the corresponding priority:

- capped grade (by max grade by every criterion)
- number of criteria
- total grade (without capped)
- number of participated/contributed events

Because of the complexity of ranking calculation, EAMS has a cron job which runs every midnight to re-calculate ranking for all students in the system so users can retrieve student's ranking immediately.

Response time of ranking about 3000 students is 15 seconds, with 1 Gbps network bandwidth

## 7.3 Check-in processes

### 7.3.1 Definition

Our check-in, check-out process applies QR code to store information about a student who participate an event. When students attend an event, they are required to show their QR code and event's assistants will scan that QR code to check-in or check-out for these students.

### 7.3.2 Define Problem

However, if QR code only contains basic information such as email and event id, it will be easily captured by students and they send it to the others to take attendance for them. We need a solution that QR will be generated dynamically and only available in short period of time.

### 7.3.3 Solution

To tackle this such problem, we use TOTP (Time-based One Time Password), an extension of HOTP (Hash-based One Time Password), as a mean of generating dynamic, short-living QR code.

Before going to detail explanation of check-in process, we will describe HOTP first, then TOTP and the final is about check-in process

#### 7.3.3.1 Hash-based One Time Password (HOTP)

This feature implementation uses a library that utilizes the Hash-based One Time Password algorithm that can be found at:

[https://en.wikipedia.org/wiki/HMAC-based\\_One-time\\_Password\\_algorithm](https://en.wikipedia.org/wiki/HMAC-based_One-time_Password_algorithm)

#### 7.3.3.2 Time-based One Time Password (TOTP)

This feature implementation uses a library that utilizes the Time-based One Time Password algorithm that can be found at:

[https://en.wikipedia.org/wiki/Time-based\\_One-time\\_Password\\_algorithm](https://en.wikipedia.org/wiki/Time-based_One-time_Password_algorithm)

### 7.3.3.1 Check-in Process

Our current check-in process:

- Student registers an event
- EAMS returns a secret key that will be stored in client-side
- In client-side, the TOTP method will use that secret key to generate a token which will be stored in QR code combines several basic information
- Event's assistance scans that QR code and send it to server for further processing
- In server side, we use the same secret key that has been stored in student's device before to verify token.
- If that token is verified successfully, it means that student has been checked-in successfully

## V. Software Testing Documentation

### 1. Overall Description

#### 1.1 Test Model

In EAMS project, we apply Agile testing for many purposes:

- It establishes an environment that developers and customers can interact with each other in a rapid and flexible way.
- It can quickly adapt with rapid pace requirement changing

#### 1.2 Testing Levels

There are some testing levels that is applied in EAMS:

- Unit testing: only use for participation flow such as register, check-in, feedback, check-out
- Acceptance testing: we work thoroughly with Student Affairs Department of FPT University to determine whether the system reaches stable level for release.

### 2. Test Plan

#### 2.1 Test Stages

Type of Test	Stage of Test			
	Unit	Integration	System	Acceptance
Functional Testing	X			
User Interface Testing				X

#### 2.2 Resources

##### 2.2.1 Human Resources

Worker/Doer	Role	Specific Responsibilities/Comments
Phạm Đức Bình	Leader	Planning, verifying test deliverables
Trần Minh Chiến	Member	Do unit testing as planned

Phan Hoàng Minh Luân	Member	Do unit testing as planned
-------------------------	--------	----------------------------

### 2.3 Test Milestones

Milestone Task	Effort (md)	Start Date	End Date
Manage event (registration, participation, feedback, check-out) Event's permission	2	10/10/2020	12/10/2020
Report for feedback Student report Event announcement Student grade history	2	01/11/2020	03/11/2020
Semester report Student assessment Manage achievement	3	15/11/2020	18/11/2020
Manage event's post Optimization	1	01/12/2020	02/12/2020

### 3. Test Cases

[Report5 Test Case Document.xlsx](#)

### 4. Test Reports

[Report5 Unit Test Case.xlsx](#)

## VI. Release Package & User Guides

### 1. Deliverable Package

#### 1.1. Source codes & documents

No.	Items	Sub-Items	Type	Version
<b>Code Package</b>				
1	Sags.backend	Web service	Modify	1.9
2	Sags.frontend	Student web application	Modify	1.9
3	Sags.admin	Staff web application	Modify	1.9
<b>Database</b>				
1	Script	db-scripts.sql	Modify	N/A

### 2. Installation Guide

#### 2.1. System requirements at server

### 2.1.1. Hardware requirement

Server	Recommended Specs
Internet connection	Cable, Wi-Fi (1 Gbps)
Computer processor	Intel® Xeon 1.4GHz
Computer Memory	2GB RAM
Storage Space	5GB

### 2.1.2. Software requirement

Area name	Name and version	Description
Operating System	Ubuntu 18.04 LTS or later Windows Server 2012 or later	Operating system for production
Environment	Kubernetes 1.19.x (web service) Docker 19.03 * Without docker: Node 12	Containerization environment
DBMS	MySQL 5.7.29	Used to create and manage database for system
Web server	Nginx	Proxy server for production

## 2.2. Installation Instruction

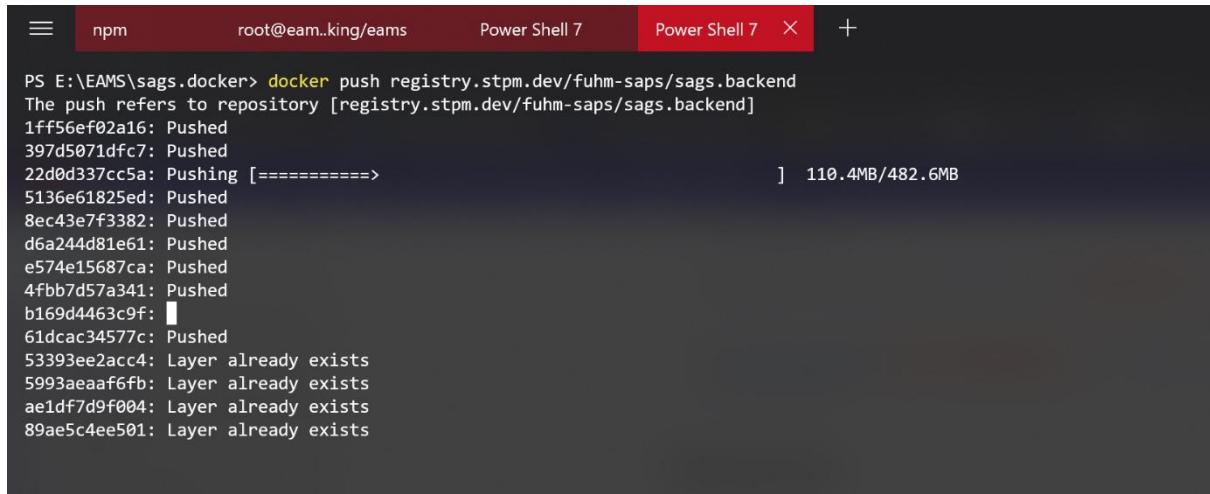
We recommend an installation procedure using docker and kubernetes to make installation and deployment faster

### 2.2.1. Prepare docker image at local

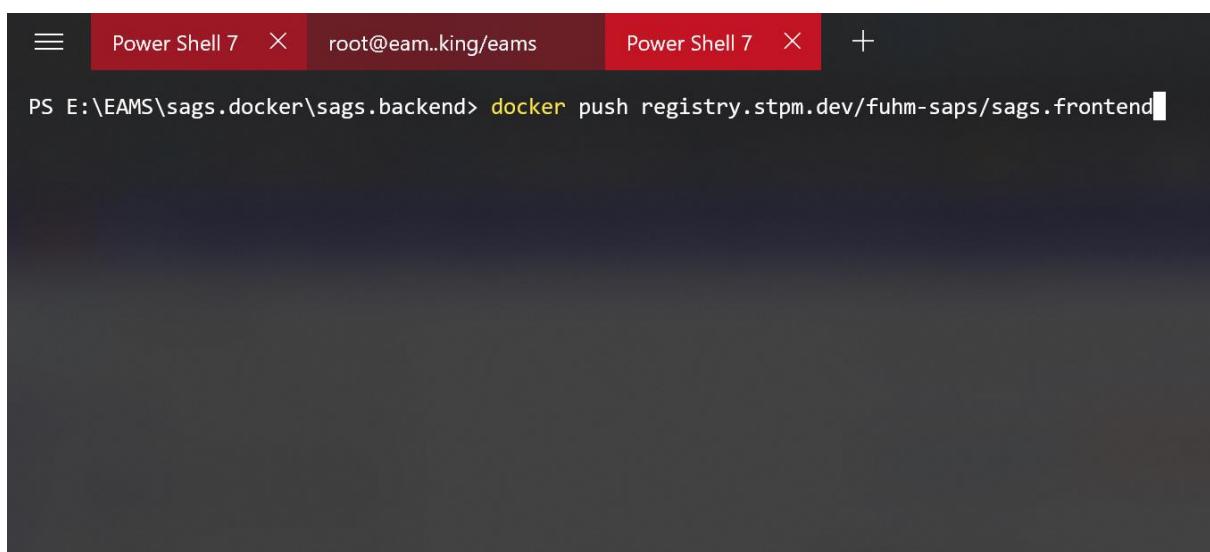
```
☰ npm          root@eam..king/eams          Power Shell 7          Power Shell 7  X  +  
PS E:\EAMS\sags.docker> docker build -t registry.stpm.dev/fuhm-saps/sags.backend:latest -f ./docker/prod.backend.Dockerfile .  
Sending build context to Docker daemon 1.274GB  
Step 1/16 : FROM node:12.18.0-alpine3.9 as build  
--> b3c8f06bfd5a  
Step 2/16 : WORKDIR /tmp  
--> Using cache  
--> be3050b51662  
Step 3/16 : COPY sags.backend/package.json ./  
--> 5c4ce3a82238  
Step 4/16 : RUN npm install  
--> Running in 70d1b3109de4
```

```
☰ Power Shell 7          root@eam..king/eams          Power Shell 7  X  +  
PowerShell 7.0.2  
Copyright (c) Microsoft Corporation. All rights reserved.  
  
https://aka.ms/powershell  
Type 'help' to get help.  
  
A new PowerShell stable release is available: v7.1.0  
Upgrade now, or check out the release page at:  
https://aka.ms/PowerShell-Release?tag=v7.1.0  
PS E:\EAMS\sags.docker\sags.backend> docker build -t registry.stpm.dev/fuhm-saps/sags.frontend:latest -f ./docker/prod.nginx.Dockerfile .
```

## 2.2.2. Push docker image to registry



```
PS E:\EAMS\sags.docker> docker push registry.stpm.dev/fuhm-saps/sags.backend
The push refers to repository [registry.stpm.dev/fuhm-saps/sags.backend]
1ff56ef02a16: Pushed
397d5071dfc7: Pushed
22d0d337cc5a: Pushing [=====] 110.4MB/482.6MB
5136e61825ed: Pushed
8ec43e7f3382: Pushed
d6a244d81e61: Pushed
e574e15687ca: Pushed
4fbb7d57a341: Pushed
b169d4463c9f: Pushed
61dcac34577c: Pushed
53393ee2acc4: Layer already exists
5993aeaf6fb: Layer already exists
ae1df7d9f004: Layer already exists
89ae5c4ee501: Layer already exists
```

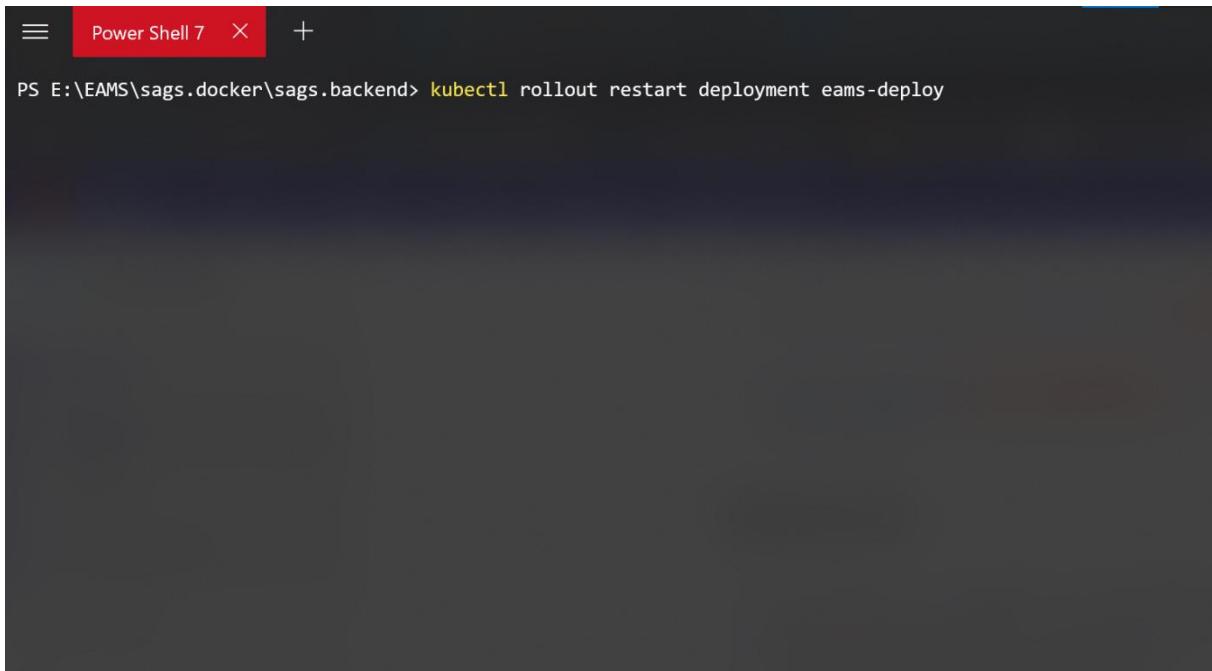


```
PS E:\EAMS\sags.docker\sags.backend> docker push registry.stpm.dev/fuhm-saps/sags.frontend
```

## 2.2.3. Apply new version in production

### 2.2.3.1. At server side

Restart Kubernetes deployment



```
PS E:\EAMS\sags.docker\sags.backend> kubectl rollout restart deployment eams-deploy
```

### 2.2.3.2. At client side

Pull docker images

```
*** System restart required ***
Last login: Tue Dec  1 02:46:54 2020 from 118.69.233.167
root@eams-main-vm:~# docker-compose -f prod2.docker-compose.yml pull
```

Run docker images

```
root@eams-main-vm:~# cd /home/working/eams
root@eams-main-vm:/home/working/eams# docker-compose -f prod2.docker-compose.yml --env-file prod.env up -d
sags-nginx is up-to-date
sags-mysql is up-to-date
root@eams-main-vm:/home/working/eams#
```

## 3. Installation guide without using docker and kubernetes

In case not using docker and kubernetes for installation and deployment, following software and environments are necessary to installation:

- OS: Ubuntu 18.04
- Environment: node 12.18
- DBMS: MySQL 5.7

- Server: nginx

Run this command to start server

```
*** System restart required ***
Last login: Tue Dec  1 02:56:48 2020 from 118.69.233.167
root@eams-main-vm:~# npm run start
```

#### 4. User Manual

##### 4.1. Terms and definitions

- Docker: a set of PaaS to deliver software in packages called containers
- Kubernetes: an open source container-orchestration system for automating deployment, scaling and management

##### 4.2. System requirement

###### 4.2.1. Hardware requirement

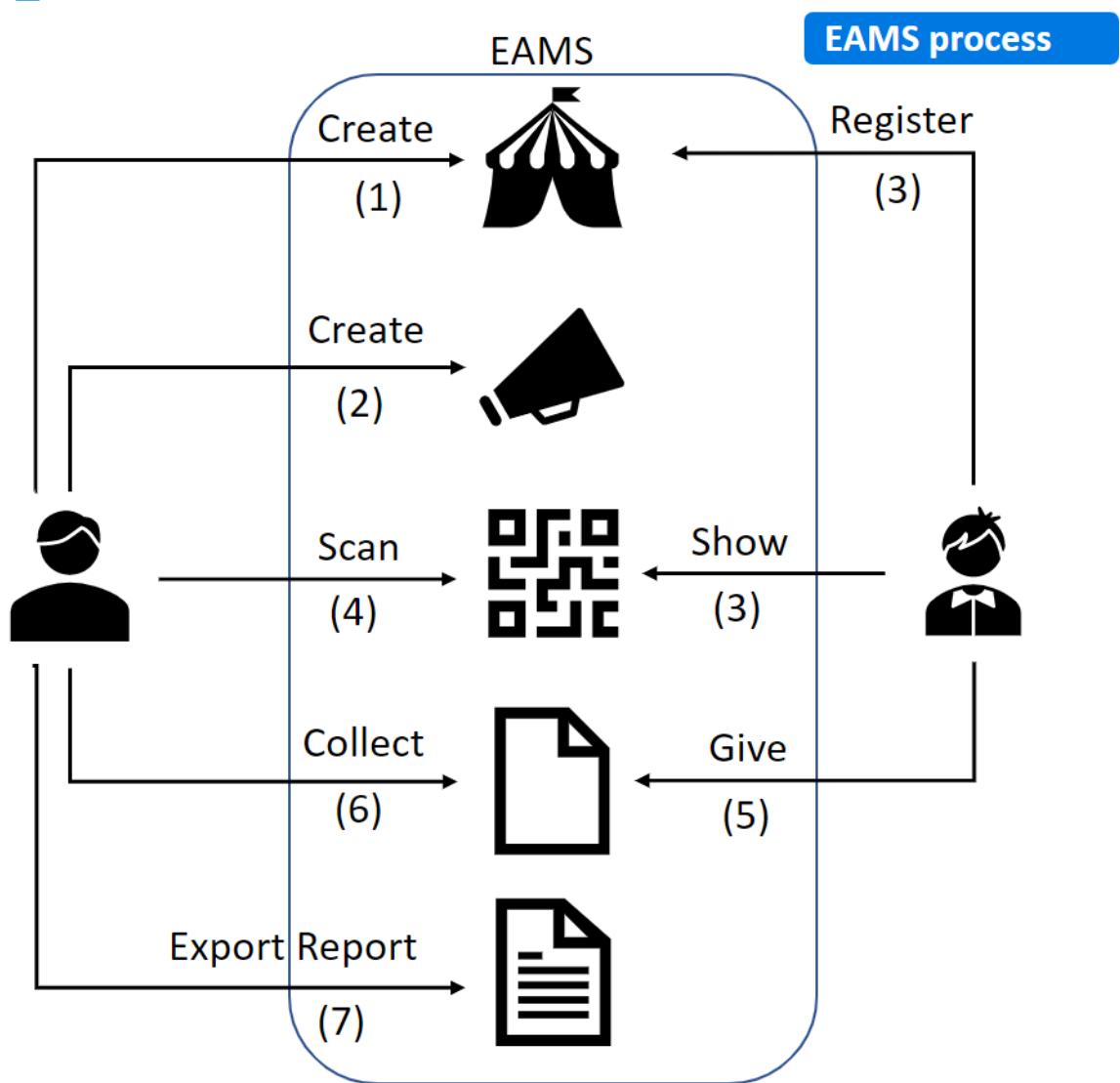
Area name	Recommended Requirements
Internet connection	Cable, Wi-Fi (8 Mbps), Mobile data (3G)
Computer Memory	1GB RAM
Camera	Required, 4MP (student application)

###### 4.2.2. Software requirement

Area name	Name and version	Description
Operating System	Windows 10, MacOS 10.11.5 (admin application)  Android 6.0 or later iOS 11.3 or later (student application)	Operating system for production
Web browser	Chrome v73 or later Safari 11.1 or later	Run web application and support PWA

##### 4.3. Application Usage

###### 4.3.1. Overview



#### 4.3.2. Features

##### 4.3.2.1. Staff creates an event

Purpose: create an event and relevant information

Step	Description
1	Click “Add event” icon to open the create event page

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
8	Event Title	The title of an event	No	Yes	Text Box	String
9	Semester	The semester this event belongs to	No	Yes	Select	String

10	Location	Location where event is organized	No	Yes	Text Box	String
11	Note	Brief text that give some more information	No	No	Text Box	String
12	Start time	The specific time event is started	No	Yes	Text Box	DateTime
13	End time	The specific time event is ended	No	Yes	Text Box	DateTime
14	Start register	The specific time that allow student to register this event	No	Yes	Text Box	DateTime
15	End register	The specific time that event closes registration	No	Yes	Text Box	DateTime
16	Restrict register	Only allows students in whitelist can register	No	No	Check Box	Boolean
17	Start checkin	The specific time that allow student to checkin this event	No	Yes	Text Box	DateTime
18	End checkin	The specific time that close checking-in this event	No	Yes	Text Box	DateTime
19	Restrict checkin	Only allows students in whitelist can check in	No	No	Check Box	Boolean
20	Start checkout	The specific time that allow student to check out this event	No	Yes	Text Box	DateTime
21	End checkout	The specific time that	No	Yes	Text Box	DateTime

		close checking-out this event				
22	Require Checkout	This event requires checking-out to recorded as participated	No	No	Check Box	Boolean

No	Function	Description	Validation	Outcome
1	Basic Information	Basic information of an event	No	
2	Description	More details about event's content	No	
3	Feedbacks	Event's feedback questions	No	
4	Tasks	Event's tasks, which can be divided into smaller sub tasks	No	
5	Sub Tasks	Event's tasks, which can be assigned to students	No	
6	Assignee	Event's contributors, who are students take account to organize this event	No	
7	Grade Criterias	Criterion to grade this event, which is placed on an official document	No	
23	BACK	Go to previous function (tab)	No	
24	NEXT	Confirm inputted fields and go to next function (tab)	Yes	

SAGS - Social Activity Grading System

CREATE A NEW EVENT

Event description

Short description \*

Short description of event

0 WORDS POWERED BY TINY

UPLOAD IMAGE

BACK NEXT

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Short description	Short description of content of this event	No	Yes	Text Box	String
2	Short description of event	Full description of content of this event	No	Yes	Text Box	String

No	Function	Description	Validation	Outcome
3	UPLOAD IMAGE	Select an image for event's cover image	Yes	A downloadable url of image
4	BACK	Go to previous function (tab)	No	
5	NEXT	Confirm inputted fields and go to next function (tab)	Yes	

SAGS - Social Activity Grading System

CREATE A NEW EVENT

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
3	Question Type	There are 2 types of question: Text and Rating	No	Yes	Select	Enum
4	Content of the question	Question content	No	Yes	Text Box	String
7	Start feedback time	The specific time allows students to give feedback for this event	No	Yes	Text Box	DateTime
8	Require Feedback Before Checkout	This event requires feedback then checkout to recorded as participated	No	No	Check Box	Boolean

No	Function	Description	Validation	Outcome
1	GET UPLOAD TEMPLATE	Download feedback question template as an excel file	No	
2	IMPORT QUESTIONS	Upload feedback questions using excel file	Yes	
5	REMOVE	Delete a question	No	

6	ADD FEEDBACK QUESTION	Add new question	No	
9	BACK	Go to previous function (tab)	No	
10	NEXT	Confirm inputted fields and go to next function (tab)	Yes	

☰ SAGS - Social Activity Grading System

CREATE A NEW EVENT

Basic Information Description Feedbacks Tasks Sub Tasks Assignee Grade Criterias

Event Tasks

Task name \*  Contribution Rate \*  REMOVE

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Task name	Name of event's task	No	Yes	Text Box	String
2	Contribution Rate	The percentage of this task of 100% event total	No	Yes	Text Box	Number

No	Function	Description	Validation	Outcome
3	REMOVE	Remove a task	No	
4	ADD TASK	Add new task	No	
5	BACK	Go to previous function (tab)	No	
6	NEXT	Confirm inputted fields and go to next function (tab)	Yes	

SAGS - Social Activity Grading System

CREATE A NEW EVENT

Event Sub Tasks

Review presentations Contribution Rate: 100%

Sub Task name \*  Contribution Rate \*  REMOVE

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Sub task name	Name of event's task	No	Yes	Text Box	String
2	Contribution Rate	The percentage of this sub task of 100% task	No	Yes	Text Box	Number

No	Function	Description	Validation	Outcome
3	REMOVE	Remove a sub task	No	
4	ADD SUB TASK	Add new sub task	No	
5	BACK	Go to previous function (tab)	No	
6	NEXT	Confirm inputted fields and go to next function (tab)	Yes	

SAGS - Social Activity Grading System

CREATE A NEW EVENT

Event Assignee

Review presentations

Contribution Rate: 100%

Book rooms

Contribution Rate: 100%

Student roll number

1

2 BACK    NEXT 3

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Student roll number	Roll number of a student who is assigned to this sub task	No	Yes	Text Box	String

No	Function	Description	Validation	Outcome
2	BACK	Go to previous function (tab)	No	
3	NEXT	Confirm inputted fields and go to next function (tab)	Yes	

SAGS - Social Activity Grading System

CREATE A NEW EVENT

Event Criterias

Grade Criteria: Academic (1) Basic Academic Events (2)

Participation Grade: 5

Please enter total contribution grade

Total Contribution Grade (3)

BACK (4) FINISH (5)

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Grade Criteria	Criteria to grade event's point	No	Yes	Select	String
2	Grade Sub Criteria	Detail criteria to grade event's point	No	Yes	Select	String
3	Total Contribution Grade	Total grade of contributions, which is divided into assignees	No	Yes	Text Box	Number

No	Function	Description	Validation	Outcome
4	BACK	Go to previous function (tab)	No	
5	NEXT	Confirm inputted fields and go to next function (tab)	Yes	

SAGS - Social Activity Grading System

Event Brief

Fcode K16 recruitment Unpublished

COPY EVENT URL 1

DASHBOARD 2 BASIC INFO 3 DESCRIPTION 4 POSTS 5 ANNOUNCEMENTS 6 FEEDBACK 7 TASKS 8

Review event information 9

Assign event helpers 10

Monitor Participation 11

Feedbacks 12

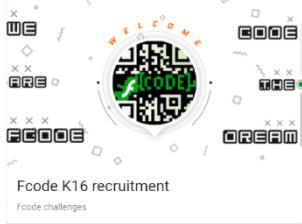
Grading 13

Finished Event 14

PUBLISH EVENT 15

**Review event information**

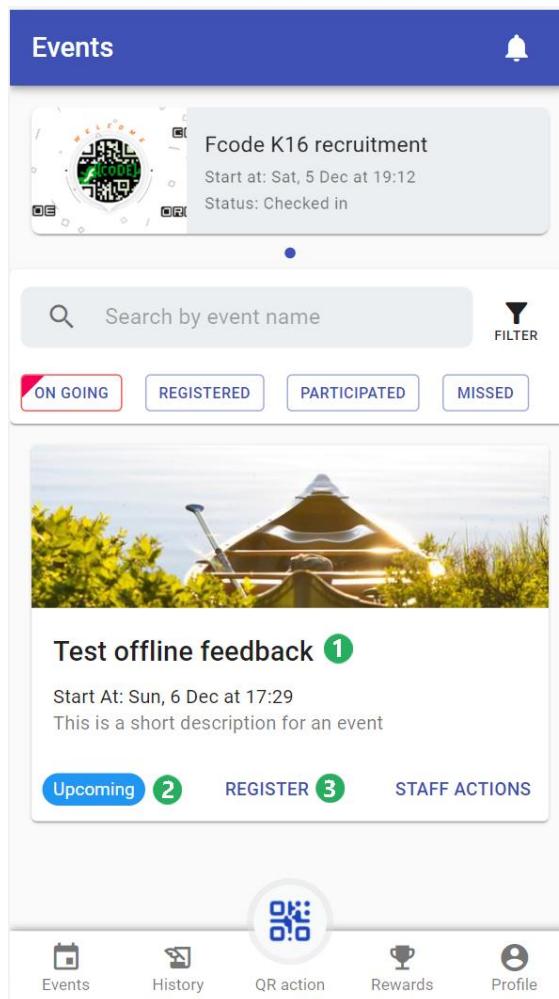
**Title:** Fcode K16 recruitment  
**Semester:** Fall 2020  
**Short Description:** Fcode challenges  
**Description:**  
 K16 will have to face 3 challenges to decide whether they are suitable for F-code club.  
**Duration:** 05/12/2020 19:12 - 05/12/2020 19:12  
**Start Register At:** 05/12/2020 19:12  
**Start Check-in At:** 05/12/2020 19:12  
**Number of feedback questions:** 1  
**Number of tasks:** 1

 Fcode K16 recruitment  
 Fcode challenges

No	Function	Description	Validation	Outcome
1	COPY EVENT URL	Get url to navigate to event's details page in student application for announcement purpose	No	
2	DASHBOARD	Sections about operating this event	No	
3	BASIC INFO	Sections about basic info: title, date time...	No	
4	DESCRIPTION	Sections about detailed desc	No	
5	POSTS	Sections about event's posts	No	
6	ANNOUNCEMENTS	Sections about event's announcements	No	
7	FEEDBACK	Sections about event's announcements	No	
8	TASKS	Sections about event's tasks	No	
9	Review event information	Displays event basic info, published or unpublished event	No	
10	Assign event helpers	Assign event permission for students	No	
11	Monitor Participation	View participation list	No	

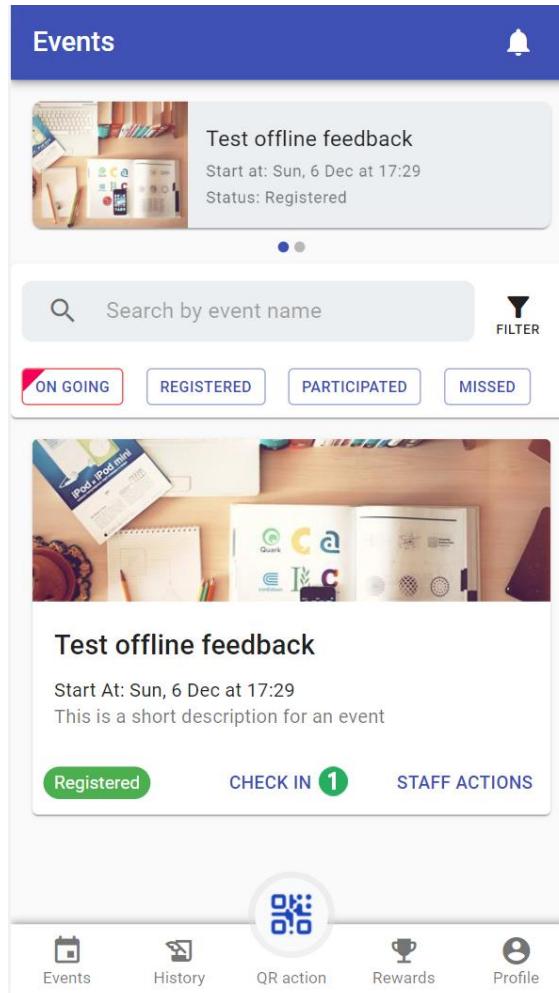
12	Feedbacks	Brief statistics of feedback result	No	
13	Grading	Grading for event participants and contributors	No	
14	Finished Event	Overview all information of event	No	
15	PUBLISH EVENT	Make event observable to students	No	

#### 4.3.2.2. Student participates an event

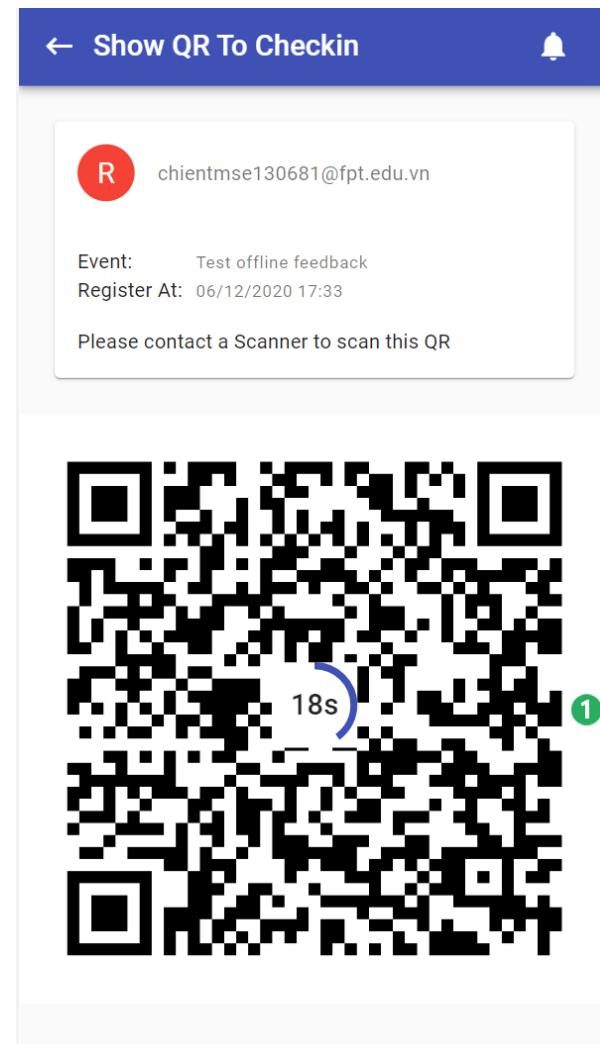


No	Function	Description	Validation	Outcome
1	Event title	Title of event	No	
2	Status	Event student status, it can be Upcoming, Registered, Participated, Feedbacked, Checked out, Missed	No	

3	REGISTER	Student sends request to register this event	Yes	
---	----------	--	-----	--



No	Function	Description	Validation	Outcome
1	CHECK IN	Student sends request to check in this event	Yes	



No	Function	Description	Validation	Outcome
1	QR code	Student's QR code to check in event	No	

← Events Feedback Form 



Test offline feedback

Feedback Form

**This is a test question type rating**

1 for unsatisfied , 5 for very satisfied

1    2    3    4    5

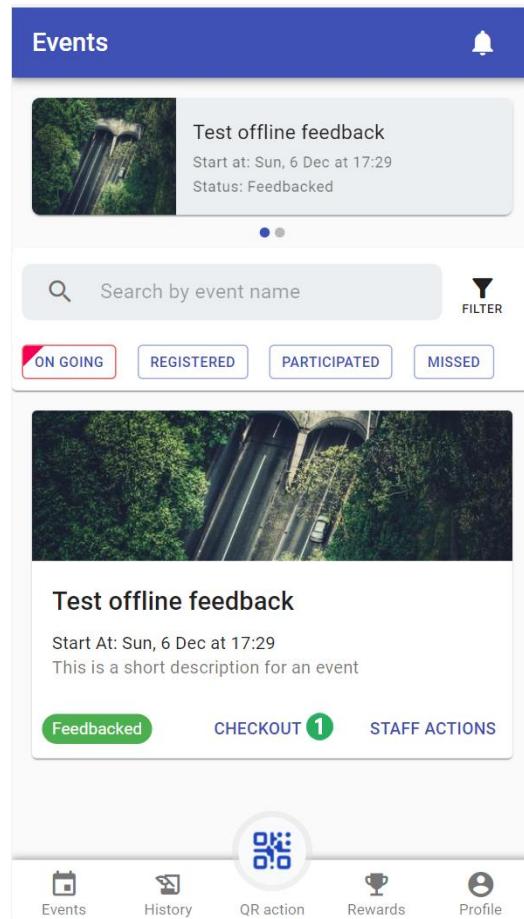
**This is a test question type text**

Please answer this question honestly

Required \*

 FEEDBACK 1

No	Function	Description	Validation	Outcome
1	FEEDBACK	Student sends their feedback answer	Yes	



No	Function	Description	Validation	Outcome
1	CHECKOUT	Student sends request to check out this event	Yes	

← Show QR To Checkout Bell

R chientmse130681@fpt.edu.vn

Event: Test offline feedback  
Register At: 06/12/2020 17:33

Please contact a Scanner to scan this QR



No	Function	Description	Validation	Outcome
1	QR code	Student's QR code to check out this event	No	

### 4.3.2.3. Staff gives grade for students

The screenshot shows the SAGS interface. At the top, the title 'Event Brief' is displayed, followed by the event name 'Fcode K16 recruitment' with a 'Published' status and a 'COPY EVENT URL' button. Below this is a navigation bar with tabs: DASHBOARD, BASIC INFO, DESCRIPTION, POSTS, ANNOUNCEMENTS, FEEDBACK, and TASKS. The DASHBOARD tab is selected. On the left, a vertical sidebar lists event milestones with checkboxes: 'Review event information' (checked, Start), 'Assign event helpers' (checked, 05/12/2020 19:12), 'Monitor Participation' (checked, 05/12/2020 19:12), 'Feedbacks' (checked, 05/12/2020 19:12), 'Grading' (unchecked, 06/12/2020 17:15), and 'Finished Event' (unchecked, End). The main content area is titled 'Grading' with the sub-instruction 'Start grading and finish this event' and a green circled number '1'.

No	Function	Description	Validation	Outcome
1	Start grading and finish this event	Open new tab to grading for participations and contributions	No	

The screenshot shows the 'Event Criterias' section. At the top, there are dropdown menus for 'Grade Criteria' (1) set to 'Academic' and 'Grade Sub Criteria' (2) set to 'Basic Academic Events'. Below this, the 'Participation Grade: 5' is displayed. A text input field 'Please enter total contribution grade' contains the value '3 20' (3). At the bottom right is a blue button 'UPDATE GRADE CRITERIAS & CONTRIBUTION GRADE' (4). A vertical sidebar on the left lists steps: 'Review event criterias & Total contribution grade' (1), 'Give participation grades' (2), 'Review tasks' (3), and 'Give contribution grades' (4).

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
----	------------	-------------	-----------	-----------	--------------	-----------

1	Grade Criteria	Criteria of this event	No	Yes	Select	String
2	Grade Sub Criteria	Sub criteria of this event	No	Yes	Select	String
3	Total Contribution Grade	Reassign total grade for contributions of this event	No	Yes	Text Box	Number

No	Function	Description	Validation	Outcome
4	UPDATE GRADE CRITERIAS & CONTRIBUTION GRADE	Update criteria and grade of this event	Yes	

SAGS - Social Activity Grading System

Review event criterias & Total contribution grade

Give participation grades

Search student ①

Name Checkin Date Checkout Date Actions

Tran Trung Kien (K15 HCM) kientntse151340@fpt.edu.vn 06/12/2020 17:15 → ②

Tran Minh Chien chientntse130681@fpt.edu.vn 06/12/2020 17:15 →

Pham Duc Binh binhphose130691@fpt.edu.vn 06/12/2020 17:14 →

5 rows < < 1-3 of 3 > >

③ Review tasks ④ Give contribution grades

GRADE 3 PARTICIPANTS ④

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Search student	Search student to add to qualified students list	No	No	Text Box	String

No	Function	Description	Validation	Outcome
2	-> icon	Disqualify a student from qualified students list	No	

3	BACK	Go to previous function	No	
4	GRADE 3 PARTICIPANTS	Confirm qualified students list and go to next function	No	

The screenshot shows the SAGS interface with the following sections and numbered callouts:

- Event Tasks:**
  - Task name: Review presentations (1)
  - Contribution Rate: 100% (2)
  - Buttons: REMOVE (3), ADD TASK (4), UPDATE TASKS (5)
- Event Sub Tasks:**
  - Sub Task name: Book rooms (6)
  - Contribution Rate: 100% (7)
  - Buttons: REMOVE (8), ADD SUB-TASK (9), UPDATE SUB TASKS (10)
- Event Assignee:**
  - Book rooms
  - Student roll number: kienttse151340@fpt.edu.vn, bichphoe130691@fpt.edu.vn (11)
  - Contribution Rate: 100%
  - Buttons: UPDATE ASSIGNEES (12), BACK (13), NEXT (14)

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Task name	Name of event's task	No	Yes	Text Box	String
2	Contribution Rate	The percentage of this task of 100% total	No	Yes	Text Box	Number
6	Sub Task name	Name of event's subtask	No	Yes	Text Box	String
7	Contribution Rate	The percentage of this sub task of 100% total of a task	No	Yes	Text Box	Number

No	Function	Description	Validation	Outcome
3	REMOVE	Remove a task from tasks list	No	
4	ADD TASK	Add new task to tasks list	No	

5	UPDATE TASKS	Update event's tasks	No	
8	REMOVE	Remove a sub task	No	
9	ADD SUB-TASK	Add new sub task to a task	No	
10	UPDATE SUB TASKS	Update sub task of a task	No	
11	x icon	Remove a student from assigned students list for a sub task	No	
12	UPDATE ASSIGNEES	Update assignees list of sub tasks list	No	
13	BACK	Back to previous function	No	
14	NEXT	Confirm and go to next function	Yes	

SAGS - Social Activity Grading System

Review event criterias & Total contribution grade

Give participation grades

Review tasks

Give contribution grades

binhpdsse130691@fpt.edu.vn

Review presentations

Book rooms

Completion Rate 3 100 % 10

kienttse151340@fpt.edu.vn

Review presentations

Book rooms

Completion Rate 100 % 10

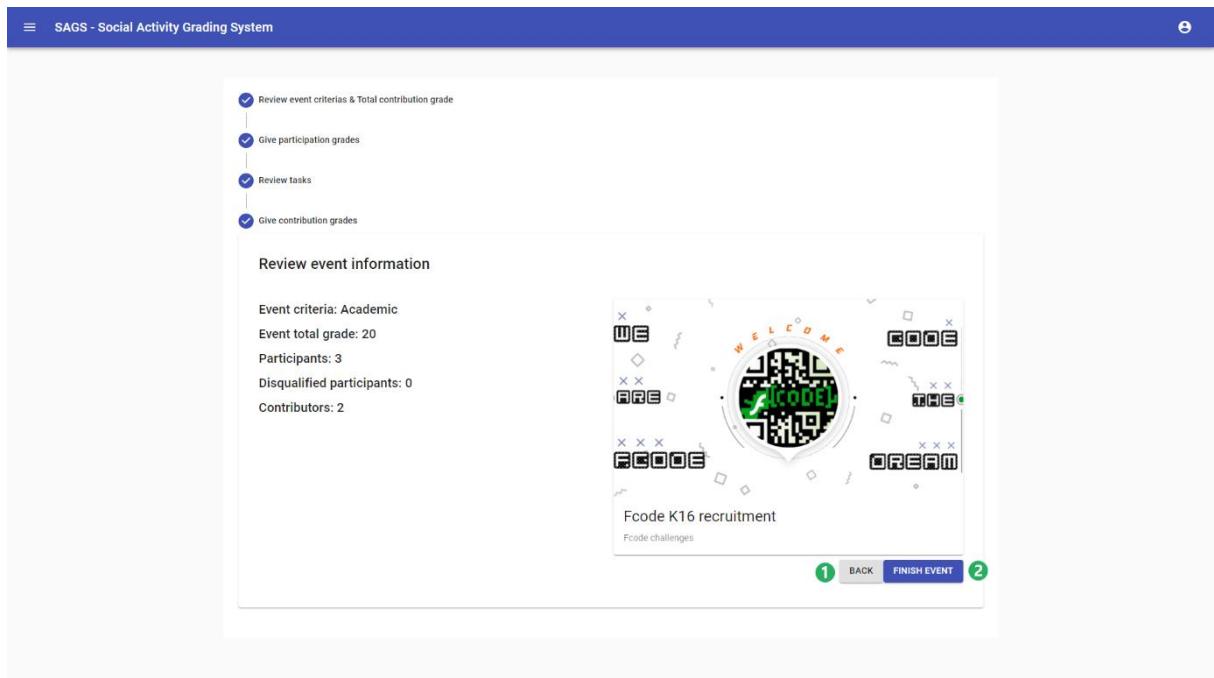
UPDATE GRADES FOR 2 CONTRIBUTORS 4

5 BACK 6

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
3	Completion Rate	The percentage of completion of a task or sub task	No	Yes	Text Box	Number

No	Function	Description	Validation	Outcome
1	COLLAPSE ALL	Collapse list of sub tasks of a contributor	No	
2	EXPAND ALL	Expand list of sub tasks of a contributor	No	

4	UPDATE GRADES FOR 2 CONTRIBUTORS	Update grade for contributors list	Yes	
5	BACK	Back to previous function	No	
6	NEXT	Confirm and go to next function	Yes	



No	Function	Description	Validation	Outcome
1	BACK	Go to previous function	No	
2	FINISH EVENT	Confirm grading then finish this event	No	

#### 4.3.2.4. Staff grants achievement for students

No	Function	Description	Validation	Outcome
1	Search	Search achievement by name	No	
2	“+” Add achievement icon	Create new achievement	No	

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Title	Achievement's title to managed	No	Yes	Text Box	String
2	Description	More details about this achievement	No	Yes	Text Box	String
5	Event	The destination event which this achievement is granted for	No	No	Select	String
6	Alternative Event Title	The text displayed on achievement pdf file	No	No	Text Box	String
7	Reason	The reason of granting	No	Yes	Text Box	String
8	Issue At	The date this achievement is issued	No	Yes	Text Box	DateTime

No	Function	Description	Validation	Outcome
3	CHOOSE A TEMPLATE	Choose achievement display template	No	
4	Reload button	Apply new achievement information to template	No	
9	BACK	Go to previous function	No	
10	NEXT	Confirm inputted fields then go to next function	Yes	

SAGS - Social Activity Grading System

Fill achievement information

Decide which students will be able to achieve this

Fetch students from \* Contribution and Participation Grade greater than \*

**1** Contribution and Participation **2**

**3** Qualified Students (3) **4** Disqualified Students (0)

**5** Search **6** BACK **7** FINISH

**8**

**9**

**10**

**11**

**12**

**13**

**14**

**15**

**16**

**17**

**18**

**19**

**20**

**21**

**22**

**23**

**24**

**25**

**26**

**27**

**28**

**29**

**30**

**31**

**32**

**33**

**34**

**35**

**36**

**37**

**38**

**39**

**40**

**41**

**42**

**43**

**44**

**45**

**46**

**47**

**48**

**49**

**50**

**51**

**52**

**53**

**54**

**55**

**56**

**57**

**58**

**59**

**60**

**61**

**62**

**63**

**64**

**65**

**66**

**67**

**68**

**69**

**70**

**71**

**72**

**73**

**74**

**75**

**76**

**77**

**78**

**79**

**80**

**81**

**82**

**83**

**84**

**85**

**86**

**87**

**88**

**89**

**90**

**91**

**92**

**93**

**94**

**95**

**96**

**97**

**98**

**99**

**100**

**101**

**102**

**103**

**104**

**105**

**106**

**107**

**108**

**109**

**110**

**111**

**112**

**113**

**114**

**115**

**116**

**117**

**118**

**119**

**120**

**121**

**122**

**123**

**124**

**125**

**126**

**127**

**128**

**129**

**130**

**131**

**132**

**133**

**134**

**135**

**136**

**137**

**138**

**139**

**140**

**141**

**142**

**143**

**144**

**145**

**146**

**147**

**148**

**149**

**150**

**151**

**152**

**153**

**154**

**155**

**156**

**157**

**158**

**159**

**160**

**161**

**162**

**163**

**164**

**165**

**166**

**167**

**168**

**169**

**170**

**171**

**172**

**173**

**174**

**175**

**176**

**177**

**178**

**179**

**180**

**181**

**182**

**183**

**184**

**185**

**186**

**187**

**188**

**189**

**190**

**191**

**192**

**193**

**194**

**195**

**196**

**197**

**198**

**199**

**200**

**201**

**202**

**203**

**204**

**205**

**206**

**207**

**208**

**209**

**210**

**211**

**212**

**213**

**214**

**215**

**216**

**217**

**218**

**219**

**220**

**221**

**222**

**223**

**224**

**225**

**226**

**227**

**228**

**229**

**230**

**231**

**232**

**233**

**234**

**235**

**236**

**237**

**238**

**239**

**240**

**241**

**242**

**243**

**244**

**245**

**246**

**247**

**248**

**249**

**250**

**251**

**252**

**253**

**254**

**255**

**256**

**257**

**258**

**259**

**260**

**261**

**262**

**263**

**264**

**265**

**266**

**267**

**268**

**269**

**270**

**271**

**272**

**273**

**274**

**275**

**276**

**277**

**278**

**279**

**280**

**281**

**282**

**283**

**284**

**285**

**286**

**287**

**288**

**289**

**290**

**291**

**292**

**293**

**294**

**295**

**296**

**297**

**298**

**299**

**300**

**301**

**302**

**303**

**304**

**305**

**306**

**307**

**308**

**309**

**310**

**311**

**312**

**313**

**314**

**315**

**316**

**317**

**318**

**319**

**320**

**321**

**322**

**323**

**324**

**325**

**326**

**327**

**328**

**329**

**330**

**331**

**332**

**333**

**334**

**335**

**336**

**337**

**338**

**339**

**340**

**341**

**342**

**343**

**344**

**345**

**346**

**347**

**348**

**349**

**350**

**351**

**352**

**353**

**354**

**355**

**356**

**357**

**358**

**359**

**360**

**361**

**362**

**363**

**364**

**365**

**366**

**367**

**368**

**369**

**370**

**371**

**372**

**373**

**374**

**375**

**376**

**377**

**378**

**379**

**380**

**381**

**382**

**383**

**384**

**385**

**386**

**387**

**388**

**389**

**390**

**391**

**392**

**393**

**394**

**395**

**396**

**397**

**398**

**399**

**400**

**401**

**402**

**403**

**404**

**405**

**406**

**407**

**408**

**409**

**410**

**411**

**412**

**413**

**414**

**415**

**416**

**417**

**418**

**419**

**420**

**421**

**422**

**423**

**424**

**425**

**426**

**427**

**428**

**429**

**430**

**431**

**432**

**433**

**434**

**435**

**436**

**437**

**438**

**439**

**440**

**441**

**442**

**443**

**444**

**445**

**446**

**447**

**448**

**449**

**450**

**451**

**452**

**453**

**454**

**455**

**456**

**457**

**458**

**459**

**460**

**461**

**462**

**463**

**464**

**465**

**466**

**467**

**468**

**469**

**470**

**471**

**472**

**473**

**474**

**475**

**476**

**477**

**478**

**479**

**480**

**481**

**482**

**483**

**484**

**485**

**486**

**487**

**488**

**489**

**490**

**491**

**492**

**493**

**494**

**495**

**496**

**497**

**498**

**499**

**500**

**501**

**502**

**503**

**504**

**505**

**506**

**507**

**508**

**509**

**510**

**511**

**512**

**513**

**514**

**515**

**516**

**517**

**518**

**519**

**520**

**521**

**522**

**523**

**524**

**525**

**526**

**527**

**528**

**529**

**530**

**531**

**532**

**533**

**534**

**535**

**536**

**537**

**538**

**539**

**540**

**541**

**542**

**543**

**544**

**545**

**546**

**547**

**548**

**549**

**550**

**551**

**552**

**553**

**554**

**555**

**556**

**557**

**558**

**559**

**560**

**561**

**562**

**563**

**564**

**565**

**566**

**567**

**568**

**569**

**570**

**571**

**572**

**573**

**574**

**575**

**576**

**577**

**578**

**579**

**580**

**581**

**582**

**583**

**584**

**585**

**586**

**587**

**588**

**589**

**590**

**591**

**592**

**593**

**594**

**595**

**596**

**597**

**598**

**599**

**600**

**601**

**602**

**603**

**604**

**605**

**606**

**607**

**608**

**609**

**610**

**611**

**612**

**613**

**614**

**615**

**616**

**617**

**618**

**619**

**620**

**621**

**622**

**623**

**624**

**625**

**626**

**627**

**628**

**629**

**630**

**631**

**632**

**633**

**634**

**635**

**636**

**637**

**638**

**639**

**640**

**641**

**642**

**643**

**644**

**645**

**646**

**647**

**648**

**649**

**650**

**651**

**652**

**653**

**654**

**655**

**656**

**657**

**658**

**659**

**660**

**661**

**662**

**663**

**664**

**665**

**666**

**667**

**668**

**669**

**670**

**671**

**672**

**673**

**674**

**675**

**676**

**677**

**678**

**679**

**680**

**681**

**682**

**683**

**684**

**685**

**686**

**687**

**688**

**689**

**690**

**691**

**692**

**693**

**694**

**695**

**696**

**697**

**698**

**699**

**700**

**701**

**702**

**703**

**704**

**705**

**706**

**707**

**708**

**709**

**710**

**711**

**712**

**713**

**714**

**715**

**716**

**717**

**718**

**719**

**720**

**721**

**722**

**723**

**724**

**725**

**726**

**727**

**728**

**729**

**730**

**731**

**732**

**733**

**734**

**735**

**736**

**737**

**738**

**739**

**740**

**741**

**742**

**743**

**744**

**745**

**746**

**747**

**748**

**749**

**750**

**751**

**752**

**753**

**754**

**755**

**756**

**757**

**758**

**759**

**760**

**761**

**762**

**763**

**764**

**765**

**766**

**767**

**768**

**769**

**770**

**771**

**772**

**773**

**774**

**775**

**776**

**777**

**778**

**779**

**780**

**781**

**782**

**783**

**784**

**785**

**786**

**787**

**788**

**789**

**790**

**791**

**792**

**793**

**794**

**795**

**796**

**797**

**798**

**799**

**800**

**801**

**802**

**803**

**804**

**805**

**806**

**807**

**808**

**809**

**810**

**811**

**812**

**813**

**814**

**815**

**816**

**817**

**818**

**819**

**820**

**821**

**822**

**823**

**824**

**825**

**826**

**827**

**828**

**829**

**830**

**831**

**832**

**833**

**834**

**835**

**836**

**837**

**838**

**839**

**840**

**841**

**842**

**843**

**844**

**845**

**846**

**847**

**848**

**849**

**850**

**851**

**852**

**853**

**854**

**855**

**856**

**857**

**858**

**859**

**860**

**861**

**862**

**863**

**864**

**865**

**866**

**867**

**868**

**869**

**870**

**871**

**872**

**873**

**874**

**875**

**876**

**877**

**878**

**879**

**880**

**881**

**882**

**883**

**884**

**885**

**886**

**887**

**888**

**889**

**890**

**891**

**892**

**893**

**894**

**895**

**896**

**897**

**898**

**899**

**900**

**901**

**902**

**903**

**904**

**905**

**906**

**907**

**908**

**909**

**910**

**911**

**912**

**913**

**914**

**915**

**916**

**917**

**918**

**919**

**920**

**921**

**922**

**923**

**924**

**925**

**926**

**927**

**928**

**929**

**930**

**931**

**932**

**933**

**934**</p

Google

## Lỗi ủy quyền

Lỗi 403: org\_internal

This client is restricted to users within its organization.

Chi tiết yêu cầu

- access\_type=offline
- response\_type=code
- redirect\_uri=https://sags.stpm-hcm.com/login
- state=Lw==
- prompt=consent
- client\_id=221974845316-0p7kb9olko3knd3ar60np35h08gphnjo.apps.googleusercontent.com
- suppress\_webview\_warning=true
- scope=https://www.googleapis.com/auth/userinfo.profile https://www.googleapis.com/auth/userinfo.email

Tiếng Việt ▾ Trợ giúp Bảo mật Điều khoản

Solution: login using university email

4.4.2. Student assistant mobile phone cannot access camera to scan QR  
Solution:

- Using Google Chrome or Safari browser
- Allow camera access permission for EAMS website

## VII. Appendix

1. OMG Unified Modeling Language™ (OMG UML) Superstructure  
<https://www.omg.org/spec/UML/2.4.1/Superstructure/PDF>

2. UML Diagram  
<https://www.uml-diagrams.org/>

3. HMAC-based One-time Password algorithm  
<https://www.ietf.org/rfc/rfc4226.txt>  
[https://en.wikipedia.org/wiki/HMAC-based\\_One-time\\_Password\\_algorithm](https://en.wikipedia.org/wiki/HMAC-based_One-time_Password_algorithm)

4. Time-based One-time Password algorithm

[https://en.wikipedia.org/wiki/Time-based\\_One-time\\_Password\\_algorithm](https://en.wikipedia.org/wiki/Time-based_One-time_Password_algorithm)

5. ReactJS

<https://reactjs.org/>

6. NestJS

<https://docs.nestjs.com/>

7. Material-UI

<https://material-ui.com/>

8. Firebase

<https://material-ui.com/>

9. Cloudmersive

<https://api.cloudmersive.com/>

10. Google PWA Training

<https://developers.google.com/web/ilt/pwa>