

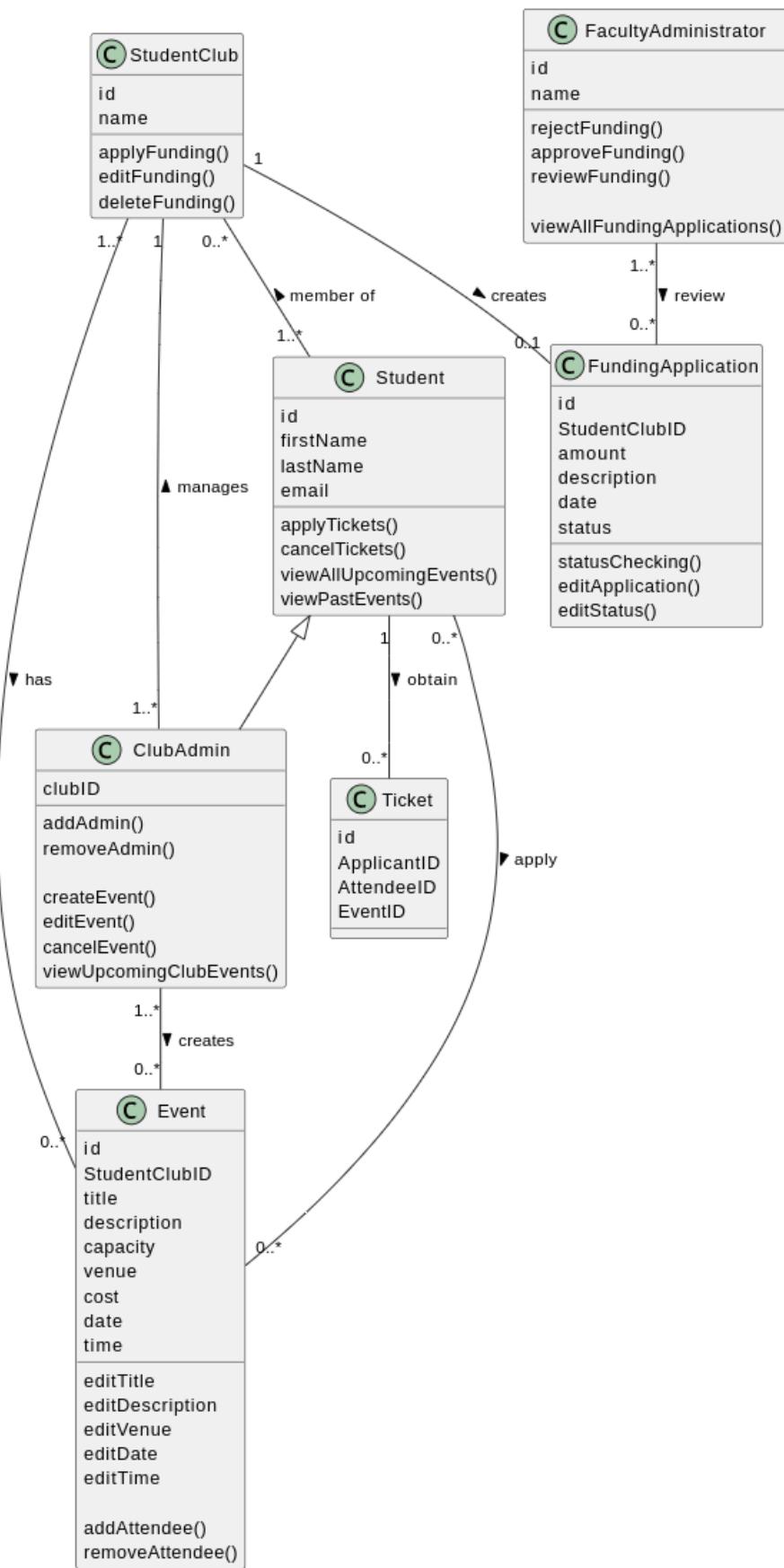
SWEN90007 Part 1A Report

Team Name: tv_addicts

Team Members:

Name	ID	Unimelb username	GitHub username	Email ID
Sameer Sikka	1169800	SSIKKA	SamSike	ssikka@student.unimelb.edu.au
Mingda Zheng	1382885	mingdaz1	MnnDa	mingdaz1@student.unimelb.edu.au
Haitian Wang	1467513	HAITIANW	Morty931	haitianw@student.unimelb.edu.au
Luyun Li	1586333	luyunli	lucy-lulu	luyunli@student.unimelb.edu.au

A.Domain Model Diagram:



B.Domain Model Explanation

This document explains the Domain Model we designed for the application intended to manage Student Clubs and Events at The University of Melbourne.

Entities and Relationships

According to the application description, we have selected these entities: *Student*, *StudentClub*, *ClubAdmin*, *Event*, *Ticket*, *FundingApplication* and *FacultyAdministrator*.

The following descriptions explain the relationship between these entities:

1. In the description:

Students with Admin privileges can create, amend or cancel events, and also add or remove admin users of the club.

We interpret it as:

ClubAdmin should inherit the Student class.

We represent this as:

```
Student <|-- Club_Admin
```

2. In the description:

Students can join various Student Clubs, and Student Clubs can have numerous members.

We interpret this as:

A Student Club can have multiple Students as members AND a Student can be a member of multiple Student Clubs, but it is also possible for a Student to not be a member of any Student Club.

We represent this as:

```
StudentClub "0..*" -- "1..*" Student : member of <
```

3. In the description:

Students can RSVP to multiple Events, but each Ticket is personal.

We interpret it as:

A Student can obtain zero or more Tickets for Events AND each Ticket is associated with exactly one Student.

We represent this as:

```
Student "1" -- "0..*" Ticket : obtain >
```

4. In the description:

Each Student Club can only submit one Funding Application per semester. But since this application will be used for more than one semester, each Student Club can have multiple Funding Applications over time.

We interpret it as:

A Student Club can create multiple Funding Applications AND each Funding Application is linked to exactly one Student Club.

We represent this as:

```
StudentClub "1" -- "0..*" FundingApplication : creates >
```

5. In the description:

Student Clubs organize various Events, and each Event needs one organizing Student Club.

We interpret it as:

A Student Club can create zero or more Events AND each Event is associated with exactly one Student Club.

We represent this as:

```
StudentClub "1..*" -- "0..*" Event : creates >
```

6. In the description:

Students can choose to attend many Events, and Events are designed for multiple attendees.

We interpret it as:

A Student can RSVP to zero or more Events AND each Event can have multiple students attending.

We represent this as:

```
Student "0..*" -- "0..*" Event : apply >
```

7. In the description:

Faculty Administrators are responsible for evaluating Funding Applications, and each Funding Application must be assessed by the Faculty Administrator.

We interpret it as:

A Faculty Administrator can review zero or more Funding Applications AND each Funding Application is reviewed by at least one faculty administrator.

We represent this as:

```
FacultyAdministrator "1..*" -- "0..*" FundingApplication : review >
```

Entities details

1. Student

- Attributes:
 - id: Unique identifier for each Student.
 - firstName: First name of the Student.
 - lastName: Last name of the Student.
 - email: Contact email for the Student.
- Methods:
 - applyTickets(): Allows a Student to RSVP to an Event, and on behalf of other Students.
 - cancelTickets(): Allows a Student to cancel Tickets that they previously applied for.
 - viewAllUpcomingEvents(): Enables a Student to view all upcoming Events.
 - viewPastEvents(): Enables a Student to view all past Events.

2. StudentClub

- Attributes:
 - id: Unique identifier for each Student Club.
 - name: Name of the Student Club.
- Methods:
 - applyFunding(): Submits a Funding Application for the Student Club, limited to 1 per semester.
 - editFunding(): Modifies an existing Funding Application.
 - deleteFunding(): Cancels a Funding Application, unless it has been Approved or Rejected.

3. ClubAdmin

- Attributes:
 - clubID: Identifier for the Student Club that this Club Admin manages.
- Methods:
 - addAdmin(): Club Admin can invite other Students to become Club Admins for the Student Club.
 - removeAdmin(): Club Admin can remove Club Admins from the Student Club.
 - createEvent(): Club Admin can create an Event for the Student Club.
 - editEvent(): Club Admin can edit an Event for the Student Club.
 - cancelEvent(): Club Admin can cancel an Event for the Student Club.

- `viewUpcomingClubEvents()`: Allow Club Admin to view all upcoming Events of the Student Club.

4. Event

- Attributes:
 - `id`: Unique identifier for each Event.
 - `StudentClubID`: The Student Club organizing the Event.
 - `title`: Title of the Event.
 - `description`: Detailed description of the Event.
 - `capacity`: Maximum number of attendees (optional).
 - `venue`: Location of the Event, either online or in-person.
 - `cost`: Expenditure for the Event, limited to total funds received using Funding Application.
 - `date`: Date of the Event.
 - `time`: Time of the Event.
- Methods
 - `editTitle`: Edits title of the Event.
 - `editDescription`: Edits description of the Event.
 - `editVenue`: Edits venue of the Event. Updates capacity of the Event as per the venue. Cancels Event if attendees are more than Venue capacity.
 - `editDate`: Edits date of the Event.
 - `editTime`: Edits time of the Event.
 - `addAttendee`: Adds an attendee to the Event. Limited to Venue capacity.
 - `removeAttendee`: Removes an attendee from the Event, and frees up Venue capacity for additional attendees.

5. Ticket

- Attributes:
 - `id`: Unique identifier for each Ticket.
 - `ApplicantID`: Unique identifier for the Student who applied for the Ticket.
 - `AttendeeID`: Unique identifier for the Student attending the Event using this Ticket.
 - `EventID`: Unique identifier for the Event of this Ticket.

6. FundingApplication

- Attributes:
 - `id`: Unique identifier for each Funding Application.
 - `StudentClubID`: The Student Club applying for funding.
 - `amount`: Amount of money requested.

- description: Description of intended use of funds by Student Club to hold Events.
- date: Date of submission of the Funding Application.
- status: Current status of the Funding Application (In Draft, Submitted, In Review, Approved, Rejected).
- Methods:
 - statusChecking(): Checks the current status of the Funding Application.
 - editApplication(): Edits the amount of funding and description for the Funding Application.
 - editStatus(): Edits the status of the Funding Application.

7. FacultyAdministrator

- Attributes:
 - id: Unique identifier for each Faculty Administrator.
 - name: Name of the Faculty Administrator.
- Methods:
 - rejectFunding(): Rejects a Funding Application.
 - approveFunding(): Approves a Funding Application.
 - reviewFunding(): Reviews a Funding Applications.
 - viewAllFundingApplications(): Administrators can view all Funding Applications.