

Team Name SAVS**Team Members**

Asmita Kumar	ak4581
Shikha Asrani	sa3864
Soamya Agrawal	sa3881
Vani Jain	vj2245

Programming Language: Python

Platform: Mac

GitHub repo: <https://github.com/SoamyaAgrawal17/SAVS>

Part 1: IA mentor meeting

State the date on which you met with your IA mentor. This should be in the past, not the future.

We met on Eurey(yn2377@columbia.edu) on October 21, 2021. In this proposal, we took into account his suggestions.

Part 2: Elaborated Service Overview

Reconsider your answers to the part 2 questions in the preliminary proposal. Elaborate in more detail the main features your service will provide.

- Our service enables clubs to host events. Club members or the general audience can register for these events.
- Club-head can manage club members, edit events, and set the visibility for these events.
- New clubs can also be registered by mentioning club name, club head, description, and category.
- Students can search for new upcoming events based on filters like category, club, date, fees.
- Students can see for their registered events (attended events, upcoming events, canceled events).
- Club members can propose an event, which can be approved or rejected by the club-head.

We will be creating the following entities, and this data (that persists across multiple executions) would be used to serve our previously mentioned use cases.

Corresponding to every entity, we have mentioned the attributes associated with it.

Event

- Event ID (auto-generated)
- Name
- Club_id (because club info can change, we can join to show the club info too)
- Visibility (Club member/ Students)
- Start Timestamp
- End Timestamp
- Location (Zoom/ Google meet link/ In-person location)
- The maximum number of people that can be accommodated.
- Description
- Fee (In \$)
- Status (Approved/Proposed/Rejected)
- Number of people registered

Student

- Name
- Student ID
- Email ID
- College
- Department

Student_Registered_Event

- Student id
- Event_id
- Status (Attended/ Upcoming/ Cancelled)

Club

- Club_id (auto-generated) (Primary Key)
- Club Name (Unique key)
- Club Head
- The category that the club belongs to. (Sports, Dance, etc.)
- Description

Roles

- Student_id
- Club_id
- Role (Club Member/ Club Head)

The service supports multiple clients. Every user is authenticated using a Google OAuth access token while making any request to the API. The clients are all distinguished by the credentials that are sent along with the request. The email IDs can be extracted and used by the API to uniquely identify the user.

Every user can only view the details for their own profile which is displayed to them using the credentials which they used to log in. This ensures that rogue users cannot access the details of other users. Additionally, users can only see details and perform actions for which they have authorization.

All users have roles assigned to them: Student(General), Club Head, Club Member. By default, all Columbia affiliates would be assigned the general (student) role. They can have Club Head, Club Member roles added to their profile with an identifier for which club they have the role for. For example, only a club head can add or remove members from their own club. Club heads do not have access to modify details of other clubs.

Use Case of our service for multiple different apps :

- Student canvas dashboard could include a list of upcoming events.
- Club websites could pull the recent events of the club to outsiders, allow club members to register events and club heads to edit events.
- On the Columbia main website. The webpage showing information on clubs could pull the recent list of clubs registered and their heads.

User Stories (as suggested by meeting with IA mentor)

Student

- As a student, I should be able to **sign up** for the application. My conditions of satisfaction are:
 - I need to be able to enter all my information clearly.
 - I should be informed if I am already registered.
 - My information should be validated with a descriptive error message response, in case it fails.
 - I should get confirmation upon successful registration.
- As a student, I should be able to **view all events**. My conditions of satisfaction are:
 - I need to be able to view all upcoming events.
 - I need to be able to filter events based on the following criteria:
 - Date (this week, this month, this year, custom date range)
 - Fees (no fees, minimum fees, maximum fees)

- Interests (choose one of the club categories)
 - Clubs (look for events hosted by a club)
 - Event eligibility (to view events that I am eligible for based on clubs, department, etc)
- I need to be able to view event information on selecting an event.
- As a student, I should be able to **register for an event**. My conditions of satisfaction are:
 - I should not be able to register for an event again if I've already registered for it.
 - I need to be able to register for an event if I have not registered for it already.
 - I should not be able to register for an event that I do not qualify for. For example, college/department/club eligibility.
 - I should not be able to register if the event is at maximum capacity.
 - The number of participants in the event should increase after registration.
- As a student, I should be able to **view my registered events**. My conditions of satisfaction are:
 - I need to be able to view my past events.
 - I need to be able to view my upcoming events.
 - I need to be able to view my cancelled events.
- As a student, I should be able to **withdraw from a registered event**. My conditions of satisfaction are:
 - I need to be able to withdraw from any registered event.
 - I should not be able to withdraw from a past event.
 - I should not be able to withdraw from an event that I am not registered for.
 - I should be able to view the event I withdrew from in the cancelled events list.
 - The number of participants in the event should decrease after I withdraw from it.
- As a student, I should be able to **create a new club**. My conditions of satisfaction are:
 - I need to be able to give all the required information to create a new club.
 - I should not be able to create a new club if the information is missing/invalid. I should get a descriptive error message response.
 - I should not be able to create a club with an already existing name.
 - I should be assigned the role of Club Head for the newly created club.

- As a student, I should be able to **see all the clubs and my role in that club**. My conditions of satisfaction are:
 - I need to be able to view all clubs.
 - I should be able to see my correct role associated with the club: Club Member or Club Head.

Club Member

- As a club member, I should be able to **perform all student tasks**.
- As a club member, I should be able to **propose an event**. My conditions of satisfaction are:
 - I need to be able to enter information to create a new event.
 - I should be informed if the information entered is wrong with descriptive error messages.
 - I should get confirmation upon successful event creation.
 - I should be able to see the event status as "Proposed".
 - I should not be able to create an event for a club I am not a part of.
- As a club member, I should be able to **edit events**. My conditions of satisfaction are:
 - I need to be able to edit all events proposed by me.
 - I should **be allowed to edit** the following fields for proposed but not yet approved events:
 - Name
 - Visibility
 - Start and end time
 - Location
 - Maximum number of participants
 - Description
 - Fees
 - I should **not be allowed to edit** the following fields for proposed but not yet approved events:
 - Club Name
 - I should **be allowed to edit** the following fields for the approved event:
 - Name
 - Start and end time
 - Location
 - Description

- I should **not be allowed to edit** the following fields for the approved event:
 - Visibility
 - Maximum number of participants
 - Fees
 - I should be informed if the information entered is wrong with descriptive error messages.
 - I should not be able to edit events not proposed by me.
 - I should not be able to edit past events.
- As a club member, I should be able to **see all the events I proposed**. My conditions of satisfaction are:
- I should be able to see all the events I proposed.
 - I should be able to see the status of each event:
Approved/Proposed/Rejected
 - I should not be able to see events I did not propose.

Club Head:

- As a club head, I should be able to **edit the details of my club**. My conditions of satisfaction are:
 - I should be able to edit the following fields in the club table:
 - Club Name
 - Club Head
 - The category that the club belongs to. (Sports, Dance, etc.)
 - Description
 - I should be informed if the information entered is wrong with descriptive error messages.
 - I should not be able to edit club details for clubs that I am not head of.
- As a club head, I should be able to **delete my club**. My conditions of satisfaction are:
 - All events of the club should be deleted.
 - All club members associated with my club should be removed from the club members list.
 - The club should be removed from the list of clubs.

- As a club head, I should be able to **replace myself with my successor**. My conditions of satisfaction are:
 - I should be able to edit the head field.
 - I should be informed if the information entered is wrong with descriptive error messages.
 - I should no longer be associated with the club. (If I am becoming a member, the new head should add me).
 - The new head should now have the role of head.

- As a club head, I should be able to **add a member to the club**. My conditions of satisfaction are:
 - New members added should now have all rights as club members.
 - The new member is still not a part of other clubs that he/she is not registered to.

- As a club head, I should be able to **remove a member from the club**. My conditions of satisfaction are:
 - The member removed from the club no longer has the rights as a club member.
 - Is not removed from any other club that they were part of.

- As a club head, I should be able to **approve/reject a pending event**. My conditions of satisfaction are:
 - The event if approved will appear on the list of events to students.
 - The event if rejected will not appear on the list of events to students.
 - Any other event is not affected.
 - Students can now register for approved events.

- As a club head, I should be able to **delete an event**. My conditions of satisfaction are:
 - I should be able to delete an event
 - Once an event is deleted no matter if it was pending/approved/rejected, it should no longer be viewable to anyone.
 - Student registrations for the event, if approved, should also be deleted.

Part 3: Elaborated Testing

Acceptance User Stories:

Student

- **Signing up as a new user**

Acceptance Test for pass cases:

- Input would be all required and valid user information and the result expected would be that the student is able to sign up.

Acceptance Test for fail cases:

- Input would be invalid form field(s) such as invalid email address, missing name, etc and the result would be a descriptive error message informing users of the expected changes. The user would also not be able to register in this case.
- Input would be valid form fields but already registered student ID or email address and the expected result would be that the user will get an error message saying that they're already registered.

- **View all events**

Acceptance Test for pass cases:

- Input would be the Student ID and the expected result is that the user will receive a list of all upcoming events.
- Input would be a valid filter such as department, club, category, etc and the result would be a list of corresponding events.

Acceptance Test for fail cases:

- Input would be an invalid filter field such as invalid category, invalid date range, etc. and the result would be an error message informing the user about the invalid field.

- **Register for an event**

Acceptance Test for pass cases:

- Input would be the Event ID and Student ID and the result would be that the user gets registered for the event and the number of participants registered should get incremented.

Acceptance Test for fail cases:

- Input would be the Student ID and the Event ID for an already registered event and the result would be that the user is informed that they're already registered.

- Input would be the Student ID and the Event ID for which the user is not eligible and the result would be an error message informing the user about the eligibility criteria (college/department/club/maximum capacity).
- **View my registered events**
Acceptance Test for pass cases:
 - Input would be the Student ID and the result would be the list of all past, upcoming, and cancelled registered events for this user.
 - Input would be the Student ID and filter criteria (past, upcoming, cancelled) and the result would be the list of events satisfying that criteria for this user.
 Acceptance Test for fail cases:
 - Input would be an invalid Student ID and the result would be an error message informing the user about the invalid Student ID.
 - Input would be an invalid filter criterion and the result would be an error message informing the user about the invalid filter.
- **Withdraw from a registered event**
Acceptance Test for pass cases:
 - Input would be the Student ID and the Event ID and the result would be that the student is no longer registered for the event. The number of participants would be decremented.
 Acceptance Test for fail cases:
 - Input would be an invalid Event ID (unregistered event, past event, non-existent event) and the result would be an error message informing the user about the invalid Event ID.
- **Create a new club**
Acceptance Test for pass cases:
 - Input would be all required and valid club information and the result expected would be that the student is able to create a new club. The student should be assigned as the Club Head for this club.
 Acceptance Test for fail cases:
 - Input would be invalid club information (missing information, invalid field value, duplicate club name) and the result would be an error message informing the user about the invalid information.
- **View all clubs and my roles**
Acceptance Test for pass cases:

- Input would be Student ID and the result expected would be a list of all clubs and their corresponding role in that club.

Acceptance Test for fail cases:

- Input would be an invalid Student ID and the result would be an error message informing the user about the invalid Student ID.

Club Member

All test cases for students also apply for a club member.

- Propose an Event

Acceptance Test for pass cases:

- Input would be all required and valid event information and the result expected would be that the club member is able to create a new event. The event should have status as "Proposed".

Acceptance Test for fail cases:

- Input would be invalid event information (missing information, invalid field value, invalid club name, club of which the user is not a member) and the result would be an error message informing the user about the invalid information.

- Edit event

Acceptance Test for pass cases:

- Input would be Event ID and allowed editable field values, and the result expected would be the event gets updated with new values.

- Acceptance Test for fail cases:

- Input would be invalid field values (invalid field value/non-editable field value), and the result would be an error message informing the user about the invalid information.
- Input would be an invalid Event ID (event not proposed by the user, past event, non-existent event), and the result would be an error message informing the user of the reason for the invalid Event ID.

- See all the events I proposed

Acceptance Test for pass cases:

- Input would be the Student ID and the result expected would be a list of events that the user proposed and their corresponding status.

Acceptance Test for fail cases:

- Input would be an invalid Student ID, and the result would be an error message informing the user about the invalid Student ID.

Club Head

All test cases for club members and students also apply for a club head.

- Edit club details

- Acceptance Test for pass cases:
 - Input would be any permutation of valid club information and the result expected would be that the club details are changed.
- Acceptance Test for fail cases:
 - Input would be invalid club information (missing information for the head, invalid club name) and the result would be an error message informing the club head about the invalid information.
 - Any club member trying to edit details of the club

- Delete club

- Acceptance Test for pass cases:
 - Input would be the club name and the expected result is that the club is deleted along with any members of the club.
- Acceptance Test for fail cases:
 - Trying to delete a club that doesn't exist.
 - Any club member trying to delete the club.

- Replace me with the next head

- Acceptance Test for pass cases:
 - Input would be the club name and name of the new head(who is a student), the expected result is that the clubhead is changed. The old head loses the rights of the club head, and the new head has the rights of the club head.
- Acceptance Test for fail cases:
 - Input would be invalid club information (invalid club name) or invalid head information(not a member of the club) and the result would be an error message informing the club head about the invalid information.
 - Trying to assign the head of a club that the user is not head of.
 - Any club member trying to change the head of the club.

- Add member

- Acceptance Test for pass cases:
 - Input would be valid club information (club id or student id), the expected result is that the member has the rights of a club member.
- Acceptance Test for fail cases:

- Input would be invalid information (invalid club id or invalid student id) and the result would be an error message informing the club head about the invalid information.
- Trying to add a member who is not a student.
- Add the club member to a club that doesn't exist.

- Remove member

- Acceptance Test for pass cases:
 - Input would be valid club information (club id or student id), the expected result is that the member no more has the rights of a club member.
- Acceptance Test for fail cases:
 - Input would be invalid information (invalid club id or invalid student id) , and the result would be an error message informing the club head about the invalid information.
 - Trying to delete a member who is not a member of the club.
 - Trying to delete a member from another club.

- Approve /Reject events

- Acceptance Test for pass cases:
 - Input would be valid event_id and status approve /reject, the expected result is that the event if approved appears in the list of events to students, and if rejected does not appear in the list.
- Acceptance Test for fail cases:
 - Input would be invalid information (invalid event id or invalid status), and the result would be an error message informing the club head about the invalid information.
 - Trying to edit an event from another club.

- Delete event

- Acceptance Test for pass cases:
 - Input would be valid event_id, the expected result is that the event does not appear in the list anymore and any registrations associated with the event are deleted too.
- Acceptance Test for fail cases:
 - Input would be invalid information (invalid event id), and the result would be an error message informing the club head about the invalid information.
 - Trying to delete an event from another club.

Testing for authentication and authorization of users

Users can be mocked for the purpose of testing.

- Acceptance Test for pass cases:
 - Authenticated users should be able to get a success 200 OK message while hitting the login or any of the secure endpoints
 - Authorized users should be able to perform actions for which they have permission: Club heads should be able to add new members to the club
- Acceptance Test for fail cases:-
 - rogue users should get 401 Unauthorized messages while hitting the login or any of the secure endpoints
 - Unauthorized users should get errors if they try to perform actions for which they do not have permission: Club heads/members should not be allowed to edit details of clubs that they are not a part of.

The test plan demonstrates an understanding of API testing, without relying on an interactive user interface.

- Unit tests in python for all methods like register for an event/create club/ edit event, including both happy and error cases.
- Writing integration tests to test if the service is working as a whole in addition to unit testing of individual components.
- Write a test suite in Postman(or any other tool like SoapUI, Rest Assured) for functionality testing like students are able to register for an event, club heads are able to create an event without approval. And all other scenarios - happy and sad as mentioned in Part1.

Part 4: Tools

Libraries, Function, and Tools that will be used

Backend runtime: Python 3

IDE: VSCode

Framework: Flask

Style Checker: [flake8](#), [pylint](#)

Static Analysis Bug Finder: [pylint](#), [mypy](#)

Test Runner: [pytest](#), [unittest](#)

Coverage Tracker: [coverage](#)