

P2: Back end implementation

Models

Entity	Attributes
User (Primary user)	ID (primary key, int) First name (charfield) Last name (charfield) Username (unique, charfield) Password (charfield, len>8)
Contact (secondary user)	ID (primary key, int) Owner (foreign key: User) First name (charfield) Last name (charfield) Email (emailfield) Key: (Owner, Email)
Calendar	ID (primary key, int) Owner (foreign key: User) Name (charfield) Description (charfield) Deadline (datetimefield) Invitees (foreign key: Contact) Finalized (boolean field)
Meeting	ID (primary key, int) Calendar (foreign key: Calendar) Contact (foreign key: Contact) Start time (datetimefield) End time (datetimefield)
Available slot	ID (primary key, int) Date (datefield) Start time (timefield) End time (timefield) Preference Level (int 1-3)
Owner Availability	ID (primary key, int) User (foreign key: User) Calendar (foreign key: Calendar) Slots (MultiKey: Available Slot)
Invitee Availability	ID (PK) Contact (FK: Contact) Calendar (FK: Calendar) Link (charfield) Status (booleanfield) slots (MultiKey: Available Slot)

List of all endpoints

1. Log in `/accounts/login/`
2. Sign up `/accounts/signup/`
3. Main page `/schedules/home/`
4. Create calendar `/schedules/calendars/create/`

5. Calendar details `/schedules/calendars/<calendar_id>/details/`
6. Remind contact `/schedules/calendars/remind/<calendar_id>/<contact_id>/`
7. Finalize calendar `/schedules/calendars/<calendar_id>/confirm/`
8. See contacts `/schedules/contacts/`
9. Add contact `/schedules/contacts/`
10. Delete contact `/schedules/contacts/`
11. Update contact `/schedules/contacts/`
12. Respond to invite `/schedules/calendars/<calendars_id>/respond/<token>`

For the User (Primary User)

Log in

Endpoint: `/accounts/login/`

Method: `POST`

Description: Log in as existing user

Body:

```
{
  "username": "username",
  "password": "password"
}
```

On success:

```
{
  "refresh": "token",
  "access": "token"
}
```

Validation errors:

- No active account found with the given credentials

Sign up

Endpoint: `/accounts/signup/`

Method: `POST`

Description: Create a new user

Body:

```
{
  "email": "hello@test.ca",
  "username": "username",
  "first_name": "John",
  "last_name": "Smith",
}
```

```
"password": "password",
"password2": "password"
}
```

On success:

```
{
  "id": 1,
  "username": "username",
  "email": "hello@test.ca",
  "first_name": "John",
  "last_name": "Smith"
}
```

Validation errors:

- Field Required
 - Password must be at least 8 characters long.
 - Passwords must match.
 - Username already taken.
-

Main page

Endpoint: `/schedules/home/`

Method: `GET`

Description: Get the data needed to generate an authenticated user's homepage of finalized and unfinalized meetings

Authorization: Bearer Token

Header:

```
{
  "Authorization": "Bearer <token>"
}
```

On Success:

```
{
  "finalized_meetings": [
    {
      "id": 1,
      "name": "name",
      "description": "description",
      "start_time": <datetime>,
      "end_time": <datetime>
    },
    {
      "id": 2,
      "name": "name",
      "description": "description",

```

```

        "start_time": <datetime>,
        "end_time": <datetime>
    }
],
"unfinalized_meetings": [
    {
        "id": 3,
        "name": "name",
        "description": "description",
        "deadline": <datetime>
    },
    {
        "id": 4,
        "name": "name",
        "description": "description",
        "deadline": <datetime>
    }
]
}

```

Errors:

- 401 Unauthorized: User is not authenticated

Create calendar

Endpoint: `/schedules/calendars/create/`

Method: `POST`

Description: Add a new calendar for the authenticated user with specified owner availabilities. All invitees will be sent an email containing a link directing them to `/schedules/calendars/<calendar_id>/respond/<token>` .

Authorization: Bearer Token

Header:

```

{
    "Authorization": "Bearer <token>"
}

```

Body:

```

{
    "name" (string, required): name of the calendar,
    "description" (string, required): description of the calendar,
    "deadline" (string Datetime format:eg YYYY-MM-DDThh:mm:ss, required):
        date and time by which invitees need to respond by,
    "invitees" (list of contact_id's): invitees of the calendar,
    "availability" (list of "available_slot"s, required): user's available times
}

available_slot format:
{

```

```

    "date" (string date format: YYYY-MM-DD, required): date of availability,
    "start_time" (string datetime format, required): starting time of availability
    "end_time" (string datetime format, required): ending time of availability,
    "preference" (int 1-3, required): preference level for a meeting scheduled
                                   during the slot (3 being highest preference)
}

Example:
{
  "name": "CSC309 Phase 2",
  "description": "Interview session after P2 submission",
  "deadline": "2024-03-15",
  "invitees": [1, 3, 5],
  "available_slots": [{
    "date": "2024-03-12",
    "start_time": "13:00:00",
    "end_time": "15:00:00",
    "preference_level": 3
  },
  {
    "date": "2024-03-13",
    "start_time": "10:00:00",
    "end_time": "12:00:00",
    "preference_level": 1
  }
]
}

```

On Success (example response):

- 201 Created

```

{
  "id": 1,
  "name": "CSC309 Phase 2",
  "description": "Interview session after P2 submission",
  "deadline": "2024-03-15T00:00:00Z",
  "invitees": [
    1, 3, 5
  ],
  "finalized": false
}

```

Errors

- 401 Unauthorized: User is not authenticated
- 400 Bad request: Invalid input fields
- Validation errors
 - "Ensure this field has no more than 150 characters."

Deadline validation:

- "Datetime has wrong format."
- "Deadline cannot be in the past"

Invitees validation:

- "Invalid contact id" (contacts do not belong to the user)
- "Contact does not exist"

Available slots validation:

- "Invalid slot data. Must be in the form: 'date', 'start_time', 'end_time', 'preference_level' "
- "Date must be a string.", "Start and end times must be strings."
- "Date must be in a valid date format YYYY-MM-DD."
- "Available date cannot be in the past."
- "Start and end times must be in a valid date format HH:MM:SS."
- "End time must be after start time."
- "Preference level must be an integer."
- "Preference level must be either 1, 2, or 3 (3 being highest preference)."

See calendar details

Endpoint: `/schedules/calendars/<calendar_id>/details/`

Method: `GET`

Description: See the details of the calendar with `calendar_id` associated with the authenticated user.

Authorization: Bearer Token

Header:

```
{
  "Authorization": "Bearer <token>"
}
```

On Success:

- 200 OK

```
{
  "Calendar": {
    "id": 1,
    "name": "CSC309 Phase 2",
    "description": "Interview session after P2 submission",
    "deadline": "2024-03-15T00:00:00Z",
    "invitees": [3],
    "finalized": false
  },
  "Owner_availability": {
    "slots": [
      {
        "date": "2024-03-12",
        "start_time": "13:00:00",
```

```

        "end_time": "15:00:00",
        "preference_level": 3
      },
      {
        "date": "2024-03-15",
        "start_time": "10:00:00",
        "end_time": "12:00:00",
        "preference_level": 1
      }
    ]
  },
  "Invitee_availability": [
    {
      "contact": {
        "id": 3,
        "first_Name": "Dane",
        "last_Name": "Joe",
        "email": "janedoe@gmail.com"
      },
      "status": true,
      "slots": [
        {
          "date": "2024-03-12",
          "start_time": "13:00:00",
          "end_time": "15:00:00",
          "preference_level": 3
        }
      ]
    }
  ]
}

```

Errors:

- 404 Not found
 - Calendar not found
 - Owner availabilities not found

Update calendar details

Endpoint: `/schedules/calendars/<calendar_id>/details/`

Method: `PUT`

Description: Update the details of the calendar with `calendar_id` associated with the authenticated user. Any new added invitees will automatically receive an invitation email upon a successful request. Requests to remove invitees or availabilities that don't exist for the calendar are ignored.

Authorization: Bearer Token

Header:

```
{
    "Authorization": "Bearer <token>"
}
```

Body:

```
{
    "name" (string, optional): name of the calendar,
    "description" (string, optional): description of the calendar,
    "deadline" (string Datetime YYYY-MM-DDThh:mm:ss, optional):
        deadline for responses,
    "add_invitees" (list of contact_id's, optional): invitees to add
    "remove_invitees" (list of contact_id's, optional): invitees to remove
    "add_availabilities" (list of "available_slot"s, optional):
        available_slots to add
    "remove_availabilities" (list of "available_slot" ids, optional):
        available_slots to remove
}
```

Example:

```
{
    "name": "New Calendar Name",
    "description": "New Calendar Description",
    "deadline": "2025-10-15",
    "add_invitees": [5],
    "remove_invitees": [3],
    "add_availabilities": [{
        "date": "2024-03-15",
        "start_time": "13:00:00",
        "end_time": "15:00:00",
        "preference_level": 3
    }],
    "remove_availabilities": [1]
}
```

On Success (Example response):

- 200 OK

```
{
    "Calendar": {
        "id": 3,
        "name": "New Calendar Name",
        "description": "New Calendar Description",
        "deadline": "2025-10-15T00:00:00Z",
        "invitees": [
            5
        ],
        "finalized": false
    }
}
```



```

    },
    "Owner_availability": {
      "slots": [
        {
          "date": "2024-03-15",
          "start_time": "10:00:00",
          "end_time": "12:00:00",
          "preference_level": 1
        },
        {
          "date": "2024-03-15",
          "start_time": "13:00:00",
          "end_time": "15:00:00",
          "preference_level": 3
        }
      ]
    },
    "Invitee_availability": [
      {
        "contact": {
          "id": 5,
          "first_Name": "John",
          "last_Name": "Smith",
          "email": "johnsmith@gmail.com"
        },
        "status": false,
        "slots": [
        ]
      }
    ]
  }
}

```

Errors:

- 401 Unauthorized: User is not authenticated
- 404 Not found: Calendar, or owner availabilities not found
- 400 Bad request: Invalid input fields
- Validation errors

- "This field is required."
- "non_field_errors": no or empty body
- "Ensure this field has no more than 150 characters."

Deadline validation:

- "Datetime has wrong format."
- "Deadline cannot be in the past"

Add_invitees validation:

- "You do not have permission to add this contact." (contacts do not belong to the user)

- "Contact does not exist"
- "This contact is already invited."

Add_available slots validation:

- "Invalid slot data. Must be in the form: 'date','start_time', 'end_time', 'preference_level' "
- "Date must be a string.", "Start and end times must be strings."
- "Date must be in a valid date format YYYY-MM-DD."
- "Available date cannot be in the past."
- "Start and end times must be in a valid date format HH:MM:SS."
- "End time must be after start time."
- "Preference level must be an integer."
- "Preference level must be either 1, 2, or 3 (3 being highest preference)."

Remind Contact

Endpoint: `/schedules/calendars/remind/<calendar_id>/<contact_id>`

Method: GET

Description: Sends a reminder email to a contact with `contact_id` if their availability for a calendar with `calendar_id` has not yet been filled.

Authorization: Bearer Token

Header:

```
{
  "Authorization": "Bearer <token>"
}
```

On Success:

```
{
  "message": "Reminder email successfully sent to <Contact Name>"
}
```

Errors:

- 401 Unauthorized: user not logged in
- 403 Forbidden: calendar not owned by user
- 404 Not Found: calendar doesn't exist, contact does not exist, contact is not an invitee

Finalize calendar

Endpoint: `/schedules/calendars/<calendar_id>/confirm/`

Method: GET

Description: Get the data needed to finalize a calendar

Authorization: Bearer Token

Header:

```
{
    "Authorization": "Bearer <token>"
}
```

On Success:

- View suggested schedules & custom schedule options

```
{
  "id": 1,
  "name": "name",
  "description": "description",
  "deadline": <date>,
  "invitees": [
    {
      "id": 1,
      "first_name": "John",
      "last_name": "Smith"
    },
    {
      "id": 2,
      "first_name": "James",
      "last_name": "Smith"
    }
  ],
  "suggested_schedules": [
    [
      {
        "id": 1,
        "first_name": "John",
        "last_name": "Smith",
        "start_time": <datetime>,
        "end_time": <datetime>
      },
      {
        "id": 2,
        "first_name": "James",
        "last_name": "Smith",
        "start_time": <datetime>,
        "end_time": <datetime>
      }
    ],
    [
      {
        "id": 1,
        "first_name": "John",
        "last_name": "Smith",
        "start_time": <datetime>,

```

```

        "end_time": <datetime>
    },
    {
        "id": 2,
        "first_name": "James",
        "last_name": "Smith",
        "start_time": <datetime>,
        "end_time": <datetime>
    }
]
],
"invitee_availability": [
    {
        "contact": {
            "id": 1,
            "first_name": "John",
            "last_name": "Smith",
        },
        "availability": [
            {
                "date": <date>,
                "start_time": <time>,
                "end_time": <time>,
                "preference": <int 1-3>
            },
            {
                "date": <date>,
                "start_time": <time>,
                "end_time": <time>,
                "preference": <int 1-3>
            }
        ],
    },
    {
        "contact": {
            "id": 2,
            "first_name": "James",
            "last_name": "Smith"
        },
        "availability": [
            {
                "date": <date>,
                "start_time": <time>,
                "end_time": <time>,
                "preference": <int 1-3>
            },
            {
                "date": <date>,
                "start_time": <time>,
                "end_time": <time>,
                "preference": <int 1-3>
            }
        ],
    }
]

```

```

    },
  ],
}

```

Errors:

- 401 Unauthorized: user not logged in
- 403 Forbidden: calendar not owned by user
- 404 Not Found: calendar does not exist
- 404 Not Found: calendar has already been finalized

Edge Case:

- If 1 or more invitee have not responded no suggested schedules will be generated

Method: **POST**

Description: Finalize a schedule

Authorization: Bearer Token

Header:

```

{
  "Authorization": "Bearer <token>"
}

```

Body:

- Chosen schedule

```

{
  "finalized_schedule": [
    {
      "contact_id" (int, required): id of the contact to assign,
      "start_time" (string, required): "YYYY/MM/DD HH:MM:SS",
      "end_time" (string, required): "YYYY/MM/DD HH:MM:SS"
    },
    {
      "contact_id" (int, required): id of the contact to assign,
      "start_time" (string, required): "YYYY/MM/DD HH:MM:SS",
      "end_time" (string, required): "YYYY/MM/DD HH:MM:SS"
    }
  ]
}

```

On Success:

- Send email to all invitees of finalized schedule

```

{
  "meetings": [

```

```

    {
        "id": 1,
        "first_name": "John",
        "last_name": "Smith",
        "email": "hello@test.ca",
        "start_time": <datetime>,
        "end_time": <datetime>
    },
    {
        "id": 2,
        "first_name": "James",
        "last_name": "Smith",
        "email": "hello@test.ca",
        "start_time": <datetime>,
        "end_time": <datetime>
    }
],
}

```

Errors:

- 401 Unauthorized: user not logged in
- 403 Forbidden: calendar not owned by user
- 404 Not Found: calendar does not exist
- 404 Not Found: calendar has already been finalized
- Validation errors (on custom schedule)
 - Unresolved availability conflict
 - Availability not selected for a user
 - Missing Fields
 - invalid contact id
 - Contact not invited to calendar
 - Meeting times overlap
 - Invalid meeting time for contact:

Edge cases:

- It is not possible to schedule meetings with everyone
 - Owner can choose to only schedule meetings with some of the invitees with finalizing
- Someone has not submitted availabilities
 - Meeting will be assigned manually by owner

See all contacts

Endpoint: `/schedules/contacts/`

Description: Retrieves all contacts associated with the authenticated (logged in) user.

Method: `GET`

Authorization: Bearer Token

Header:

```
{
  "Authorization": "Bearer <token>"
}
```

On success: (example response)

```
[
  {
    "first_Name": "John",
    "last_Name": "Smith",
    "email": "johnsmith@aol.com"
  },
  {
    "first_Name": "Josh",
    "last_Name": "Cano",
    "email": "joshcano@gmail.com"
  }
]
```

Errors:

- 401 Unauthorized: User is not authenticated

Add Contacts

Endpoint: `/schedules/contacts/`

Description: Adds a new contact associated with the authenticated user.

Method: `POST`

Authorization: Bearer Token

Header:

```
{
  "Authorization": "Bearer <token>"
}
```

Body:

```
{
  "first_Name" (string, required): first name of contact,
  "last_Name" (string, required): last name of contact,
  "email" (string, required): email address of contact
}
```

On success: (example response)

- 201 CREATED

```
{
  "first_Name": "John",
  "last_Name": "Smith",
  "email": "johnsmith@aol.com"
}
```

Errors:

- 401 Unauthorized: User is not authenticated
 - 400 Bad request: Invalid input fields
 - "This field is required"
 - "A contact with this email already exists."
 - "Enter a valid email address."
-

Delete contact

Endpoint: `/schedules/contacts/`

Description: Delete a contact associated with the authenticated user.

Method: `DELETE`

Authorization: Bearer Token

Header:

```
{
  "Authorization": "Bearer <token>"
}
```

Body:

```
{
  "contact_id": (integer, required): ID of the contact to be deleted
}
```

On success (example response):

- 204 No content

```
[
  "Contact deleted successfully."
]
```

Errors:

- 401 Unauthorized: User is not authenticated
- 403 Forbidden: Contact does not belong to the user
 - "Invalid contact"
- 404 Not found: Contact does not exist in database
 - "detail": "Not found."

Update contact

Endpoint: `/schedules/contacts/`

Description: Update a contact associated with the authenticated user.

Method: `PATCH`

Authorization: Bearer Token

Header:

```
{
    "Authorization": "Bearer <token>"
}
```

Body:

```
{
    "contact_id" (int, required): ID of the contact to be updated,
    "first_Name" (string, optional): updated first name of contact,
    "last_Name" (string, optional): updated last name of contact,
    "email" (string, optional): updated email address of contact
}
```

Example:

```
{
    "contact_id": 1,
    "first_Name": "John",
    "last_Name": "Smith",
    "email": "johnsmith@aol.com"
}
```

On success (example response):

```
{
    "contact_id": 1
    "first_Name": "John",
    "last_Name": "Smith",
    "email": "johnsmith@aol.com"
}
```

Errors:

- 401 Unauthorized: User is not authenticated
- 403 Forbidden: Contact does not belong to the user
 - "Invalid contact"
- 404 Not found: Contact does not exist in database
 - "detail": "Not found."
- 400 Bad request: Invalid request body
 - "email": ["Enter a valid email address"]

- "non_field_errors": ["No input."]
 - <"first_Name", "last_Name">: ["Ensure this field has no more than 150 characters."]
-

For the Invitee (Secondary User)

Respond to invite

Endpoint: `/schedules/calendars/<calendar_id>/respond/<token>/`

Method: `GET`

Description: Get the data needed to generate a page for a contact (as identified by `token`) to fill out their availability for a calendar with `calendar_id`

On Success:

```
{
  "calendar_id": 1,
  "name": "name",
  "description": "description",
  "deadline": <datetime>,
  "owner": {
    "id": 1,
    "first_name": "Jaden",
    "last_name": "Smith"
  },
  "invitees": [
    {
      "id": 1,
      "first_name": "John",
      "last_name": "Smith"
    },
    {
      "id": 2,
      "first_name": "James",
      "last_name": "Smith"
    }
  ],
  "owner_availability": [
    {
      "date": <date>,
      "start_time": <time>,
      "end_time": <time>,
      "preference": <int 1-3>
    },
    {
      "date": <date>,
      "start_time": <datetime>,
      "end_time": <datetime>,
      "preference": <int 1-3>
    }
  ]
}
```

```
    ],  
  }  
}
```

Errors:

- 404 Not Found: calendar does not exist

Method: POST

Description: Submit availability for a contact (as identified by `token`) on a calendar with `calendar_id`

Body:

```
{  
  "availability": [  
    {  
      "start_time" (string, required): "YYYY/MM/DD HH:MM:SS",  
      "end_time" (string, required): "YYYY/MM/DD HH:MM:SS",  
      "preference_level" (int, required): <int 1-3>  
    },  
    {  
      "start_time" (string, required): "YYYY/MM/DD HH:MM:SS",  
      "end_time" (string, required): "YYYY/MM/DD HH:MM:SS",  
      "preference_level" (int, required): <int 1-3>  
    }  
  ]  
}
```

On Success:

- 200 OK

```
{  
  "success_message": "Availability submitted successfully."  
}
```

Errors:

- 401 Unauthorized: invalid token
- 404 Not Found: calendar does not exist
- 404 Not Found: invitee availability does not exist
- 400 Bad Request: calendar is no longer accepting submissions
 - "error": "Deadline has passed"
- 400 Bad Request: contact already responded
 - "error": "Already responded"