Amanda Chan
Sreerenjini Namboothiri
Briana Lonappan

CSC 365 Group Project: Chores Manager API Specification and Example Flows

Project Summary

Our project is a backend API for managing chores in shared living spaces such as dorms, apartments, or houses. The goal is to allow users (college students) to form roommate groups and assign, view, and complete chores collaboratively.

Supabase pw: Xkb90vcsD9Hx1qn6

Connection string: postgresql+psycopg://postgres.dviouzpbucnzdqltalaw:Xkb90vcsD9Hx1qn6@aws-0-us-west-1.pooler.supabase.com:5432/postgres

API key: 6e218fff10d4c18817bc5ee89cbfd51f

API Specification

The API is designed using RESTful principles. Below are 8 endpoints with descriptions, input/output examples, and methods.

1. POST /groups

Create a new roommate group.

Input:

```
{
  "group_name": "The Cottage Core",
  "admin_id": 1
}
```

Output:

```
{
  "group_id": 42,
  "message": "Group created successfully."
}
```

2. POST /groups/join

Join an existing group using an invite code.

Input:

```
{
  "user_id": 3,
  "invite_code": "WELCOME123"
}
```

Output:

```
{
  "message": "User added to group."
}
```

3. POST /chores

Create a new chore.

Input:

```
{
  "group_id": 42,
  "chore_name": "Take out trash",
  "due_date": "2025-04-25",
  "assignees": [2, 3],
  "recurring": "weekly"
}
```

Output:

```
{
  "chore_id": 101,
  "message": "Chore created."
}
```

4. PUT /chores/{chore_id}

Edit an existing chore.

Input:

```
{
  "chore_name": "Take out recycling",
  "due_date": "2025-04-26"
```

```
}
```

Output:

```
{
  "message": "Chore updated."
}
```

5. PATCH /chores/{chore_id}/complete

Mark a chore as completed.

Input: No input necessary.

Output:

```
{
  "message": "Chore marked as complete."
}
```

6. GET /users/{user_id}/chores

View all chores assigned to a user.

Input: No input necessary.

Output:

```
[
  {
    "chore_id": 101,
    "chore_name": "Dishes",
    "due_date": "2025-04-22"
  }
]
```

7. GET /groups/{group_id}/chores

View all chores for a group.

Input: No input necessary.

Output:

```
[
  {
    "chore_id": 101,
    "chore_name": "Dishes",
    "completed": false
  }
]
```

8. GET /groups/{group_id}/stats

View completion stats for each group member.

Input: No input necessary.

Output:

```
{
  "Sreerenjini": 4,
  "Amanda": 2,
  "Briana": 5
}
```

Example Flows

Flow 1: Creating a Group and Adding Members

Summary: Amanda wants to create a group and invite roommates.

1. **POST /groups**

```
{ "group_name": "Beachside Roomies", "admin_id": 1 }
```

2. **POST /groups/join**

```
{ "user_id": 2, "invite_code": "BEACH123" }
```

3. **GET /groups/101/chores**

```
[ ]
```

Flow 2: Creating and Completing a Chore

Summary: Sreerenjini creates a recurring chore and Briana completes it.

1. **POST /chores**

```
{
  "group_id": 101,
  "chore_name": "Water Plants",
  "due_date": "2025-04-23",
  "assignees": [3],
  "recurring": "weekly"
}
```

2. **PATCH /chores/202/complete**

```
{ "message": "Chore marked as complete." }
```

3. **GET /groups/101/stats**

```
{ "Sreerenjini": 0, "Amanda": 0, "Briana": 1 }
```

Flow 3: Viewing Weekly Chores

Summary: Briana checks her chores for the week.

1. **GET /users/3/chores**

```
[
  { "chore_name": "Clean Kitchen", "due_date": "2025-04-22" },
  { "chore_name": "Vacuum", "due_date": "2025-04-24" }
]
```

2. **GET /groups/101/chores**

```
[
  { "chore_name": "Clean Kitchen", "completed": false },
  { "chore_name": "Vacuum", "completed": false }
]
```

3. **GET /groups/101/stats**

```
{ "Amanda": 2, "Briana": 4, "Sreerenjini": 1 }
```

Complex Endpoints

1) POST /chores/assign-balanced

Description:
Automatically assigns a chore to the least busy members of a group by analyzing current active (incomplete) chore loads.

Input:

```
{
  "group_id": 42,
  "chore_name": "Clean bathroom",
  "description": "Scrub, mop, disinfect",
  "due_date": "2025-06-01T12:00:00",
  "assignees": [1, 2, 3],
  "recurring": "weekly"
}
```

Output:

```
{
  "chore_id": 105,
  "assigned_to": [2, 3],
  "message": "Chore assigned fairly."
}
```

2) POST /chores/reminders/send

Description:
Sends reminders to users who have chores due within the next specified number of hours.

Input:

```
{
  "group_id": 42,
  "timeframe_hours": 48
}
```

Output:

```
{
  "reminders_sent": [
```

```
    {
      "user_id": 3,
      "chore_id": 108,
      "message": "Reminder: 'Take out trash' is due by
2025-05-28T18:00:00"
    }
  ]
}
```