**Technical Specification Sheet**

***Notes:*** *technical documentation to help stakeholders and developers understand the software development.*

*May need to separate documentation and user manual*

*Include screenshots in some places where a diagram may be needed for demonstration purposes*

*If some parts are hard to write let the team know and we can adjust the layout if we are unable to write for everything listed here*

**Product Documentation**

*The programs are more prototype than product in their current state*

(Positive/Negative Preference System)

1. System Description
   1. This system is a web application that allows users to input and submit preferences for various projects. Users can submit both **positive** and **negative preferences**. Each team can submit a maximum of 4 positive preferences and 10 negative preferences. The system logs the most recent submission for each team.
   2. The **preference model** includes:

* **Positive Preferences:** Teams can express preferences for up to 4 projects.
* **Negative Preferences:** Teams can express preferences for up to 10 projects.

The system manages team numbers and student IDs to ensure that submissions are only accepted from valid, registered teams. Submitted data is saved into a Google Sheet.

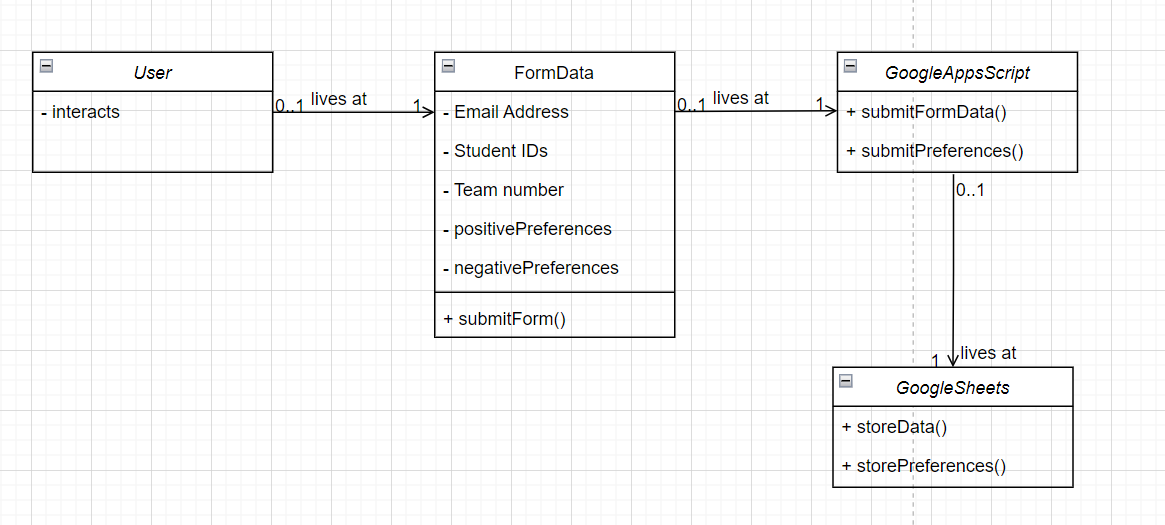
* 1. Requirements
  2. User stories

1. As a team, I want to submit project preferences so that my project assignments reflect my team’s needs.
2. As a user, I want to view instructions clearly before making a submission to ensure my preferences are submitted correctly.
3. As a developer, I want to manage submissions through Google Apps Script to simplify the backend logic.

You can see the user Story here: [User Story for P/N system](https://docs.google.com/spreadsheets/d/1XHC7Md2xk-OWQvvGMNbcslDmw10_Eo51/edit?gid=1732174599#gid=1732174599)

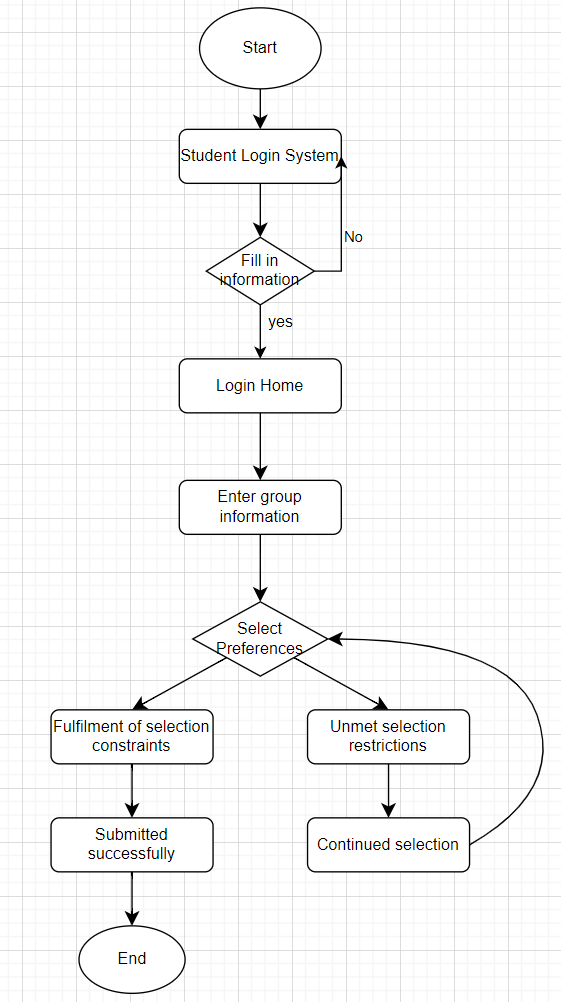
1. System Diagram

* **Class Diagram:** The key components involve an HTML form for input, Apps Script for backend processing, and Google Sheets for data storage.



* **Flow Diagram:**

1. User fills in the form (project preferences, team number, student IDs).
2. The data is submitted and processed by Apps Script.
3. Data is stored in Google Sheets, overwriting any previous submissions for the same team



1. System Details
   * **Framework/Software:** 
     + Google Apps Script for backend scripting
     + HTML, JavaScript, jQuery for frontend
     + Google Sheets as the database
   * **Versions Used:**
     + Google Apps Script platform
     + jQuery 3.0.0, Bootstrap 3.3.7
   * **File Structure:**
     + index.txt: Contains the main form page with inputs for positive and negative project preferences.
     + formPage.txt: Manages the form submissions for team details, including student IDs and team passwords.
     + Code.txt: Includes the backend logic for processing and storing the preferences and team data in Google Sheets.
2. System setup (Developers can set up the system by):
   1. Accessing the Google Apps Script editor.
   2. Deploying the web app to manage form submissions.
   3. Connecting the script to the Google Sheets for data storage.
3. System Testing
4. **Functionality Testing:**
   1. **Positive Preference Submission:** Testing ensures that the system accepts up to 4 positive preferences, verifies that no preferences are missing, and stores them correctly in the sheet.
   2. **Negative Preference Submission:** Testing ensures the negative preferences are capped at 10 and stored correctly.
5. **Error Testing:**
   1. **Form Validation:** Verifies that forms will not submit without required fields (email, team number, password, and student IDs).
   2. **Duplicate Submission:** Ensures that new submissions overwrite previous ones for the same team.
6. System troubleshooting

**Known Issues:**

1. If multiple teams submit preferences for the same project at the same preference level, data is concatenated.
2. Password hashing for teams is not fully implemented and needs further work to securely store team passwords.

**Limitations:**

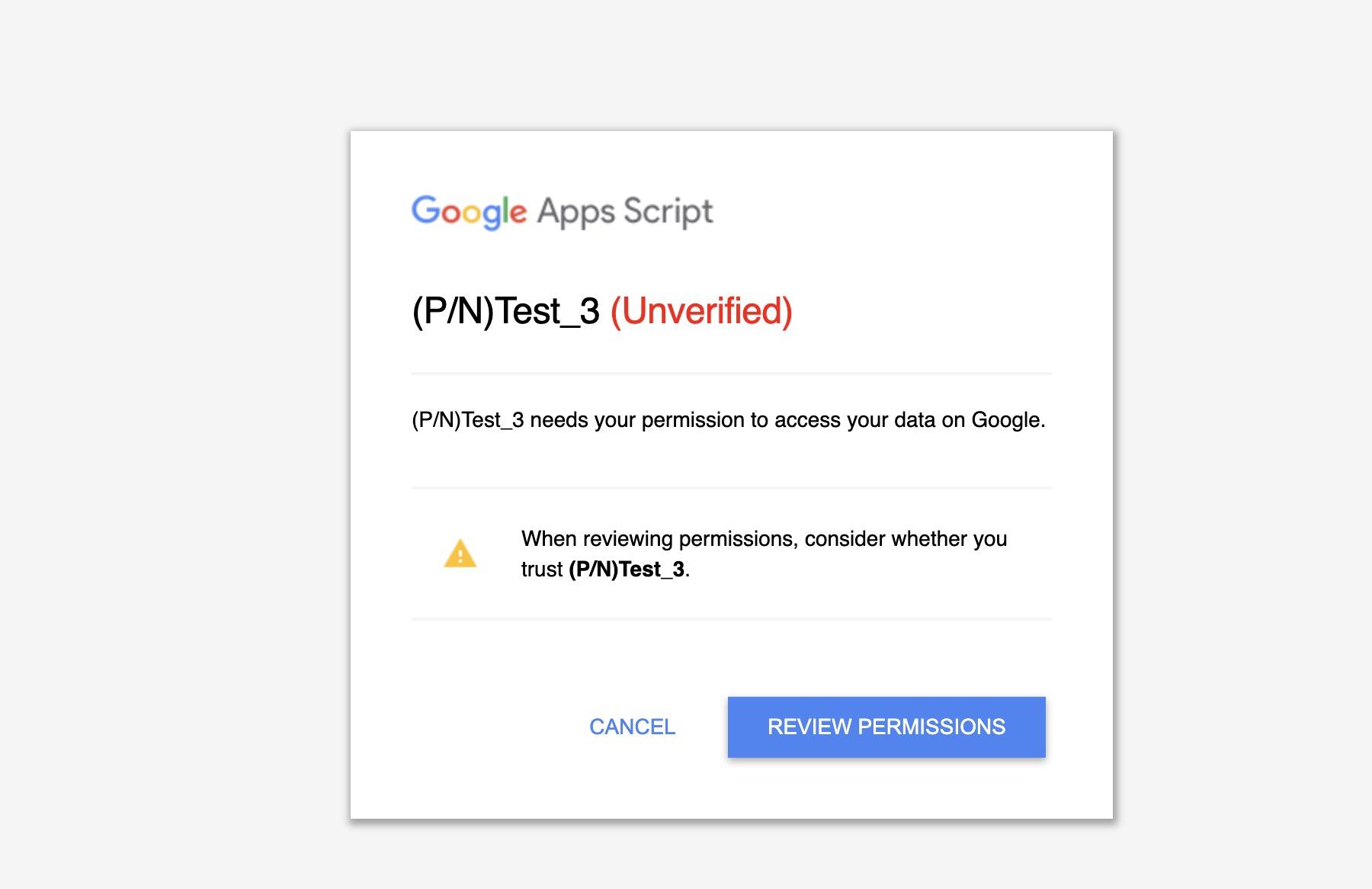
* Currently only supports up to 4 positive and 10 negative preferences per submission.
* Only the most recent submission is stored per team

**User Manual**

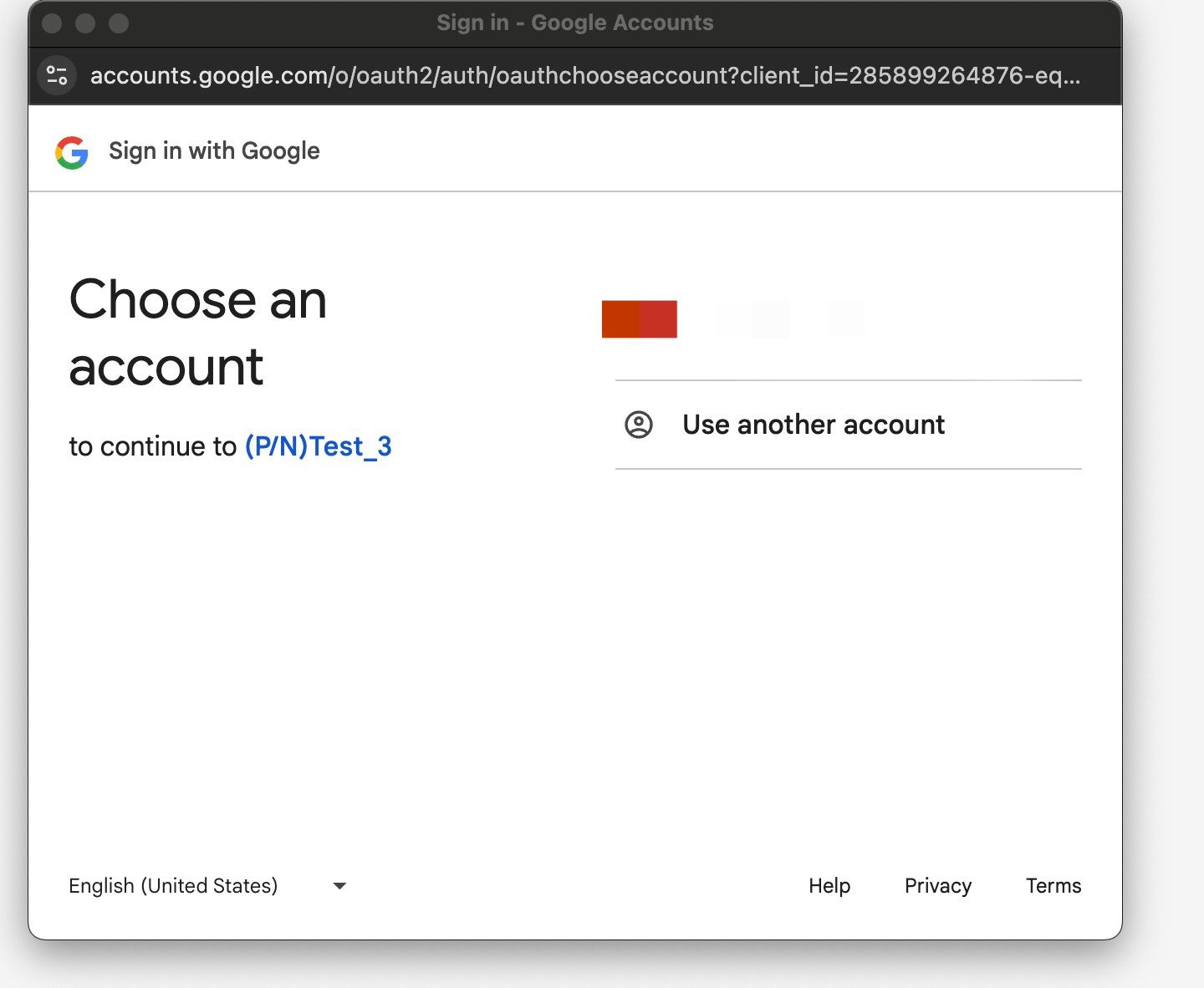
1. Software information

Google Apps Script is a cloud-based scripting language developed by Google, based on JavaScript, that allows users to write automation scripts for applications such as Google Workspace (e.g. Google Sheets, Google Docs, Google Forms, Google Drive). When used in conjunction with Google Forms, Apps Script can handle form submissions, automate data storage, trigger email notifications, and more.

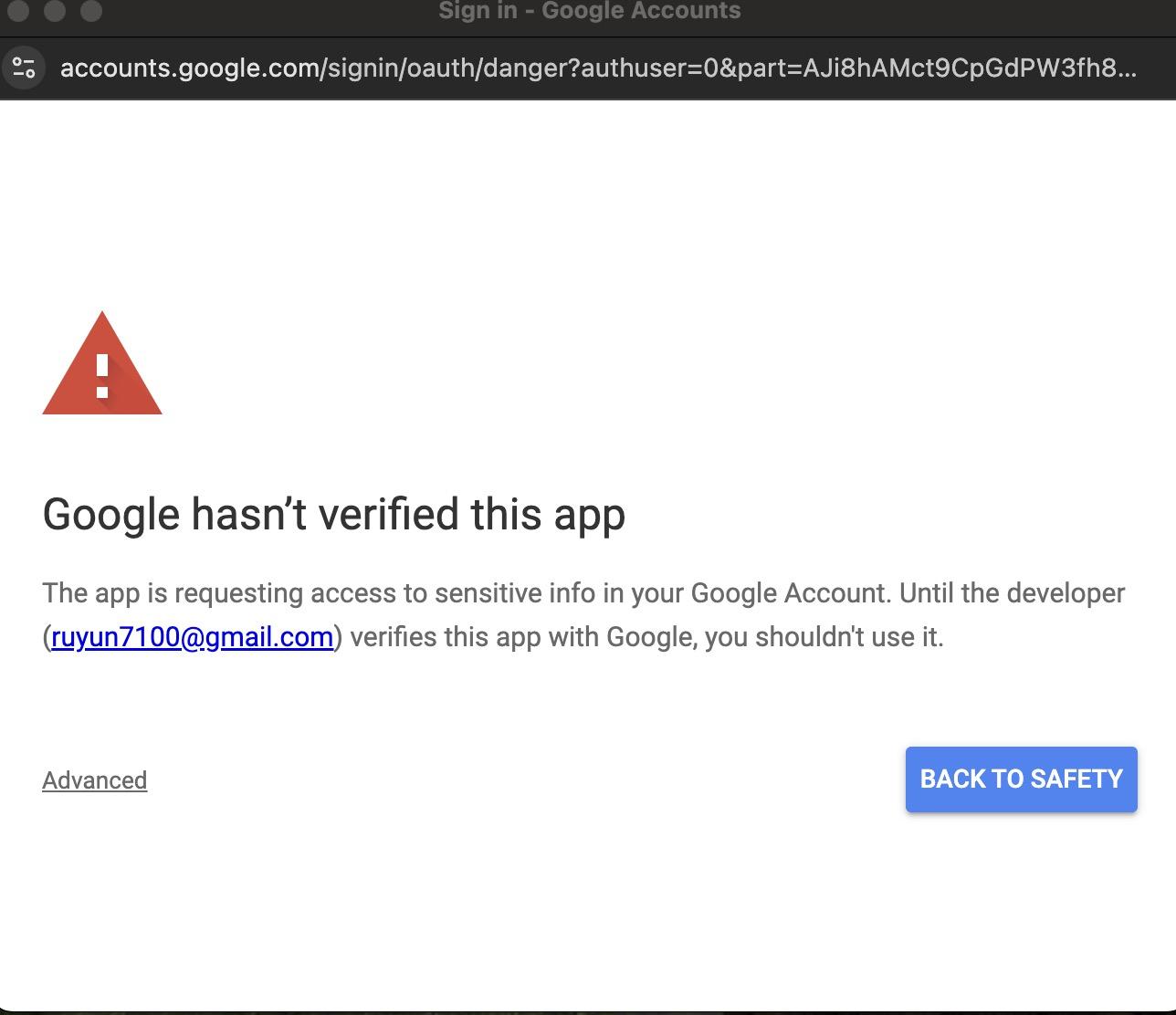
1. Quick start/guide for users
   * Authorise a Google account to log into the system
     + Open the web app link provided.
     + Click on **REVIEW PERMISSION**.



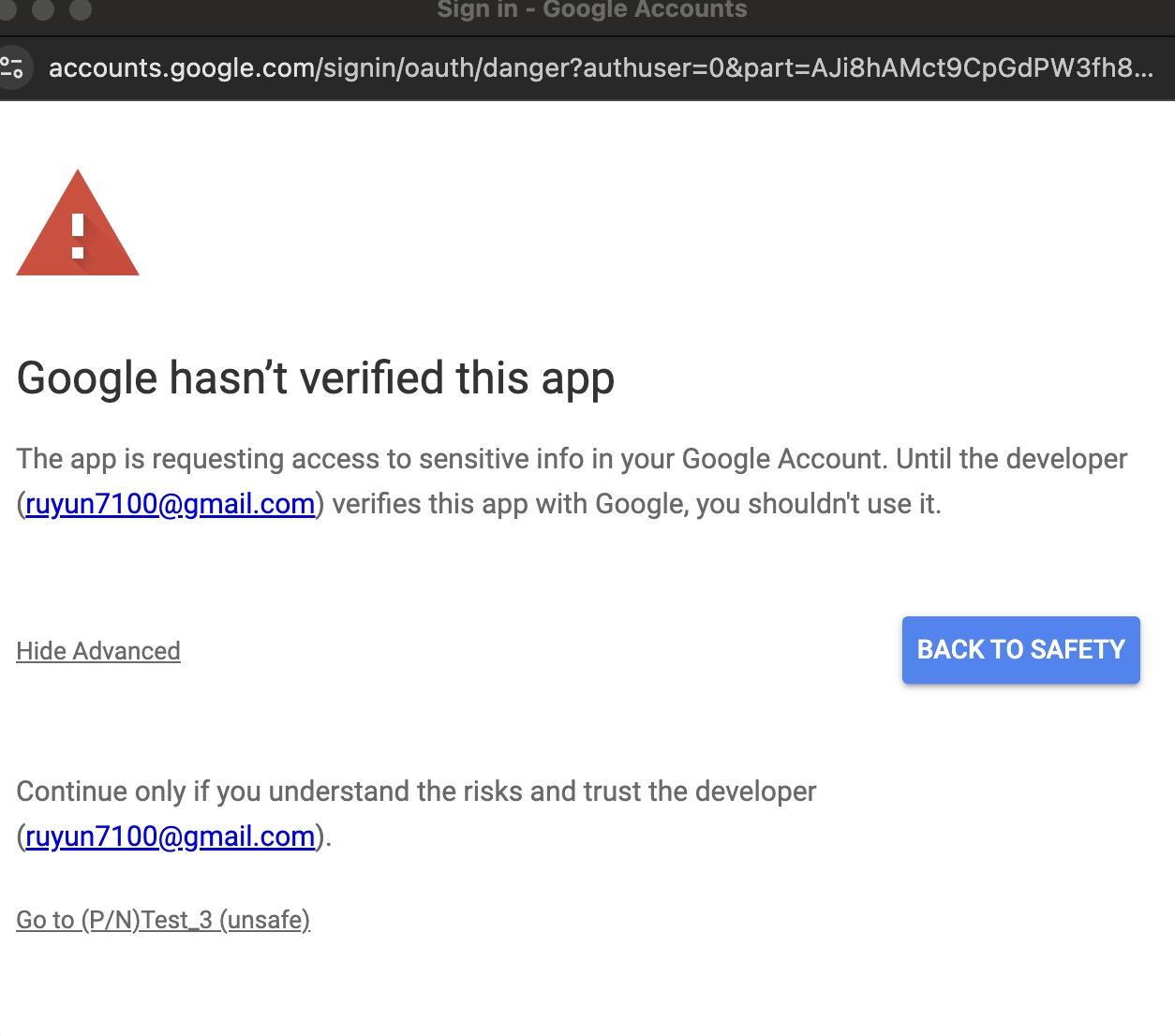
* + - Choose a suitable Google account.



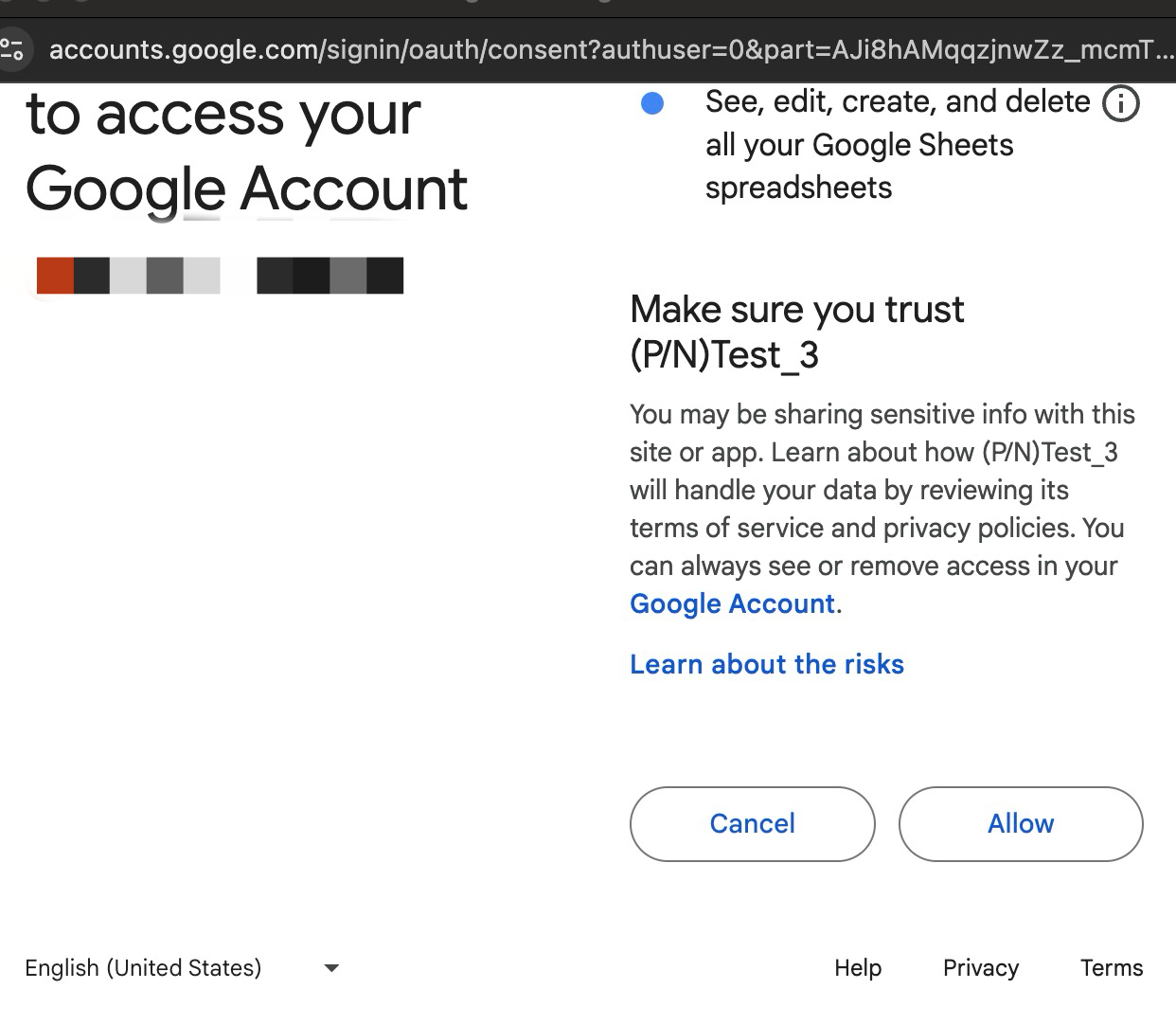
* + - Click **Advanced** to give permission to log in to the app.



* + - Click **Go to (P/N)Test\_3(unsafe)**.



* + - Click **Allow**.



* + Fill in the required fields:
    - **Email Address**
    - **Team Number**
    - **Student IDs (4 required)**
  + Enter your project preferences under the **Positive** and **Negative** sections.
  + Submit the form.

1. System Installation

For the web app:

* + Ensure access to Google Apps and the necessary permissions for interacting with Google Sheets.

1. Using the Software
   1. Positive Preferences: Enter up to 4 project preferences.
   2. Negative Preferences: Enter up to 10 project preferences.
   3. Each submission is saved into a Google Sheet, with the most recent submission overwriting older data for the same team.
2. User Troubleshooting

* Form Errors: Ensure all fields are completed before submitting. Missing information will trigger an error message.
* Submission Issues: If submissions fail, ensure your internet connection is active and you have the correct permissions for accessing Google Sheets.