**Event Object**

* **eventID**: A unique identifier for the event, generated randomly by Firestore. Used to access or re-access the event.
* **eventDate**: The date and time of the event, stored as a DateTime object.
* **eventTotalCost**: The total cost of the event, calculated by summing up all groupTotalPaidCost from all groups.
* **eventAvgCost**: The average cost per person, calculated as eventTotalCost / totalParticipants.
* **groupObjects**: A collection of unique group objects participating in the event.
* **totalParticipants**: The total number of participants in the event, calculated by looping through all groupSize values.

**Group Data Object**

* **id**: A unique identifier for the group, generated in ascending order or randomly by Firestore.
* **groupName**: The name of the group, with a letter limit of 25 characters and no symbols allowed.
* **groupSize**: The number of people in the group, controlled by +/- buttons.
* **groupPaidItem**: A list of items paid for by the group, with an extra layer of JSON for detailed information.
* **groupTotalAmount**: The total amount the group is responsible for, calculated as eventAvgCost \* groupSize.
* **groupNetPaidAmount**: The net amount the group has paid, calculated as groupTotalAmount - groupTotalPaidCost.

**PaidItem Object**

* **itemID**: A unique identifier for the paid item, preferably in ascending order.
* **itemDescription**: A description of the item, with a letter limit and allowing symbols.
* **itemCost**: The cost of the item.
* **Remarks**: Additional remarks about the item, with a letter limit of 100 characters. Includes a "read more" button and an option to allow photos (not stored in Firestore).

**Calculations and Flow**

1. **eventTotalCost** is calculated by summing up groupTotalPaidCost from all groups.
2. **eventAvgCost** is calculated as eventTotalCost / totalParticipants.
3. **groupTotalAmount** is calculated as eventAvgCost \* groupSize.
4. **groupNetPaidAmount** is calculated as groupTotalAmount - groupTotalPaidCost.
5. **totalParticipants** is calculated by looping through all groupSize values.

**Summary**

This design ensures that each event can manage multiple groups, where each group can consist of one or more participants. Each group has a list of items they paid for, and the application calculates the total and net amounts for each group and the entire event. This structure allows for a flexible and clear organization of data, making it easy for programmers to understand and implement.

**Data Flow**

* The Event object contains high-level information about the event.
* The Group Data Object contains details about each group participating in the event.
* The PaidItem Object contains details about each item paid for by the groups.
* Calculations ensure accurate distribution and understanding of costs among participants.

This text version provides a clear and organized overview of the model, helping you or any other programmer understand the relationships and logic within the SnapSettle application.