Suggested sequence of events:

1. The admin sets a deadline by which points 2 & 3 have to be completed by the supervisors.
2. A supervisor specifies a list of topics that they are willing to supervise. They can choose from a list of available topics or suggest new ones.
3. A supervisor specifies how many groups they are willing to supervise. This will be useful later in the matching as if only one supervisor can supervise topics A and B, and if that supervisor can only supervise one group, then it is not good to make one group on topic A and another group on topic B.
4. Once the above is done, the admin has a view of all available supervisors and topics they are willing to supervise.
5. The admin sets a deadline by which point 6 has to be completed by the students.
6. A student submits their top 4 choice of preferences from the available list of topics before the given deadline.
7. Once the above is done, the admin has a view of all students who have registered and submitted their preferences.
8. Looking at the total number of students who have registered and submitted preferences + the available supervisors and topics, the admin indicates an upper and lower limit as to how big he wants each group to be. For example, if the number of students are low, he may say that each group should have 3-5 members + supervisor, but if there are a large number of students, he may say that the group should be between 5-7 + supervisor.
9. The admin then runs the matching/allocation algorithm and is able to view results in an editable table. He can publish once he's happy.

Additional notes:

- The problem is not simply the matching/allocation of students to groups/topics (with the added complexity of adding supervisors), but to measure the quality of an allocation.
- Each student is only matched to one topic/group. This is a one-sided preference project. Only students indicate preferences over topics. Topics and supervisors do not have preferences.
- Not all topics need to be allocated.
- The more students who are allocated to their first preference, the better. Among the students who are not allocated their first preference, as many as possible should be allocated their second preference, and so on.
- Sometimes different groups are allocated to the same topic. It is good if as few topics as possible are used for more than one group. The topics that get allocated must be such that the supervisors can be matched to those topics. This is where point 3 from above becomes relevant.
- In terms of the algorithm to use, a flow-based method can be good in trying to allocate as many students as possible, but it doesn't automatically ensure that each topic that is used has at least the minimum number of students required for a group. So it may be good to do a preprocessing or enumeration that fixes the topics to be used and then applies the flow-based method to determine the allocation for an instance where all topics are used. One can also assign weights to potential matching edges (e.g. smaller cost for first preference, larger for second preference, etc.) and compute a minimum-cost flow.