To implement this functionality, you can follow these steps:

1. Create a new model called `CaseAssignment` to store the relationship between cases and advocates. The model should have the following fields:
- `case`: a foreign key to the `Clientcase` model
- `advocate`: a foreign key to the `User` model (advocate users)
- `status`: a field to indicate the status of the assignment (e.g., "pending", "accepted", "rejected")
2. Create a new view function called `advocate_list` to display a list of all advocates to the client. This view should retrieve all users whose `is_advocate` field is `True` and pass them to the template to display.
3. Create a new view function called `assign_case` to handle the assignment of cases to advocates. This view should retrieve the selected advocate and the case from the request, create a new `CaseAssignment` object with the "pending" status, and save it to the database. Finally, it should redirect the user to a "success" page.
4. Create a new view function called `advocate_assignments` to display all case assignments for a specific advocate. This view should retrieve all `CaseAssignment` objects for the selected advocate and pass them to the template to display.
5. Update the `Clientcase` model to include a field for the assigned advocate. This field should be nullable since the case may not be assigned yet.
6. Create a new view function called `accept_assignment` to handle advocate acceptance of case assignments. This view should retrieve the selected `CaseAssignment` object from the request, update its status to "accepted", and update the assigned advocate field in the corresponding `Clientcase` object. Finally, it should redirect the advocate to a "success" page.
7. Create a new view function called `reject_assignment` to handle advocate rejection of case assignments. This view should retrieve the selected `CaseAssignment` object from the request and update its status to "rejected". Finally, it should redirect the advocate to a "success" page.

Note: you will need to update your forms, templates, and URL configurations to support these new views and models.

With these changes, your workflow will look like this:

1. The client registers a case.
2. The client selects an advocate from the advocate list and submits an assignment request.
3. The advocate receives the request and can choose to accept or reject it.
4. If the advocate accepts the request, the case and client details are assigned to the advocate.
5. The client and advocate can view the status of the case assignment and update it accordingly