Thinking about boundary use cases, we discover that there is another actor involved, the system administrator. The use cases for this actor are the following: manage PCMembers, which allows for CRUD operations over the PCMembers table in the database, and manage server, which allows the actor to start or stop the server-side application. On the client side there are no new use cases or actors.

Manage PCMembers:

entry: the sysadmin selects this option from the console and the server is off.
flow of events: the sysadmin is presented with the option of performing CRUD operations via a menu-based console interface. When performing a change the database will be updated immediately.
exit: the sysadmin chooses to exit option in the menu.

Manage server:

entry: the sysadmin selects this option from the console.
flow of events: the sysadmin can either start or shutdown the server from a menu-based console interface. If he chooses to start the server and it is already on he will get an error message. Otherwise the server starts up normally. If the actor chooses to shutdown the server and it is already off he gets notified of the situation. Otherwise the server shuts down normally. If the server did not shutdown normally the sysadmin is notified before being presented with the 2 options.
exit: the sysadmin chooses the exit option in the menu.

Create account:

entry: a new user clicks the create an account button on the client application
flow of events: the user enters their credentials, which are sent to the server-side DatabaseCommunicationSubsystem. The subsystem sends back to the client application a confirmation for the creation the new account and the user is notified on his side of the app that they need to confirm their account
exit condition: the client-side application returns to the login screen

Log in:

entry: the user selects the login function in the client-side application.
flow of events: the user enters the credentials, which are sent to the server and the DatabaseCommunicationSubsystem verifies them. If they are correct the server sends back a confirmation that the account is valid, otherwise it sends an error. In the second case the user is notified of the incorrect credentials and they can asked to try again.
exit condition: the client-side application gets a confirmation and advances.

Submit abstract:

entry: An author selects the “submit abstract” option in the GUI
flow of events: The client app verifies the deadline with the server. In case it has passed the client receives an error message which is displayed to the user. Otherwise the user fills up all the required fields and clicks the “submit” button. The data is sent to the server, where the DatabaseCommunicationSubsystem adds it to the user’s abstracts.
exit condition: The user selects another option from the GUI

Submit document:

entry: the author clicks the “upload document” button on the abstract screen
flow of events: the client app checks the deadline with the server. In case it has passed the client receives an error message which is displayed to the user. Otherwise a window for selecting the desired document opens and the user can choose it from there.
exit condition: the document is successfully uploaded, the user selects another option from the GUI or an error is received from the server

Bid on paper:

entry: the PCMember selects the “bid” option from the GUI.
flow of events: the client app requests all the submitted abstracts from the server and displays them in a list. The user can select from it a single one to view it in more detail. From there they can select the option that best describes their wilingnes to review the given paper and, once they do that, click on the “submit bid” button. They repeat this process for each paper.
exit condition: The PCMember selects an other option from the GUI.

View reviews:

entry: the reviewer selects the “view other reviews” option from the GUI.
flow of events: the client app requests from the server the papers ant the reviews for each paper. The reviewer can select from a list a paper, then from another list a review of the paper. The the review appears in a non-editable textbox.
exit: the reviewer selects an other option from the GUI.

Review paper:

entry: the reviewer selects the “review paper” option from the GUI.
flow of events: the client app requests from the server the list of abstracts that have been assigned to him. The user then selects an abstract from a list and fill the form with all the necessary data.

Clicking the “submit” button sends the data to the server, which saves it in the database. This happens for every paper assigned to the reviewer.
exit: the reviewer selects an other option from the GUI.

Change deadline:

entry: the chair selects the “change deadline” option from the GUI.
flow of events: the client app requests the deadlines from the server and displays them in textboxes. The chair then modifies which ones they desire and press the “submit” button. The new deadlines are then sent to the server and updated in the database.
exit: the chair selects an other option from the GUI.

Assign paper to reviewer:

entry: the chair selects the “assign paper to reviewer” option from the GUI or the chair chooses to assign new reviewers to a paper with a mixed review.
flow of events: The client app requests the list of abstracts and the list of all reviewers from the server and displays them in the lists. The chair first selects an abstract then, from a list of reviewers that bid positively for the respective paper, they select at least 2 and at most 4 reviewers for each paper and press the “assign” button.
exit: the chair selects an other option from the GUI.

Manage mixed reviews:

entry: the chair selects the “manage mixed reviews” option from the GUI.
flow of events: the client app request the abstracts for each mixed review and displays them in a list. The chair can then request a certain paper’s reviewers to reexamine it, can accept or reject he paper or assign new reviewers for that paper. All these options are represented through buttons and the decision is sent to the server along with the necessary data such that the database can be updated. If the chair decides to assign new reviewers, the use case for this option is opened
exit: the chair selects an other option from the GUI or decides to assign new reviewers.

Organize papers into sections:

entry: the chair selects the “organize papers into sections” option from the GUI.
flow of events: the client app requests all the accepted papers from the server and displays them in a list. The chair can create a section, give it a name and add papers to the section. When done, the chair click the “update” button and the changes are sent to the server and stored in the database.
exit: the chair selects an other option from the GUI.

Choose section to supervise:

entry: the chair selects the “choose section to supervise” option from the GUI.
flow of events: the client app requests the list of all sections from the server and displays it in a list. The chair can highlight an item in the list and click the “choose” button. This sends the data to the server, which returns an error message if the chair cannot supervise the selected section. Otherwise it does not send back anything and the database is updated
exit: the chair selects an other option from the GUI.

Improve paper:

entry: the speaker selects the “improve paper” option from the GUI.
flow of events: the client app requests the speaker’s papers and the reviews for each paper and displays them in two lists. The speaker first chooses the paper, then the review. After this the evaluation appears in a textbox and the speaker can read it. Then he can choose to update the paper if he so desires b uploading a new document for it.
exit: the chair selects an other option from the GUI.

Upload presentation:

entry: the speaker selects the “upload presentation” option from the GUI.
flow of events: the user can view their old presentation if it exists or press the “upload presentation” button. The second action opens a window for selecting the file which will be sent to the server.
exit: the chair selects an other option from the GUI.