On the start of a commit, there are several hashmaps that are initialized. There are three hashmaps that are used to record the number of commit votes, abort votes, and acknowledgements sent by the user nodes. These are important because they allow the server to know whether to commit a collage or abort it based on the values that are mapped. In the case that the number of commit votes matches the number of user nodes, the server will know to commit a collage; in the case that there is at least one abort vote, the server will know to abort. The server will continuously loop until all acknowledgements are received by all user nodes and the acknowledgements are recorded using the hashmap.

After the initialization of these hashmaps, the server must log necessary metadata before sending out a message to user nodes indicating that preparation has started. The server also sends user nodes the collage image array and an array of each node's respective source images. After sending this message out, the server waits for votes from each user node using the deliverMessage() function. Depending on the vote, deliverMessage() will update the global hashmaps commitMap or abortMap. The server loops until either all user nodes send a commit vote or at least one user node sends an abort. Another case is the server does not receive all votes before a timeout value of 6 seconds, the server will abort.

When a user node receives a message from the server indicating the "prepare" phase, it makes two checks first: all of the sources exist on the user node and those sources are not locked. If the user node fails these checks, then an abort vote is sent back to the server. Otherwise, the user locks the sources and calls askUser() to ask the user whether to commit or abort. If the vote is to commit, the user node logs information about what phase it is in and the decision that has been made by the user. This decision is then serialized and sent to the server.

Once a decision has been made from the server, the server logs metadata and sends the decision to all user nodes using broadcastDecision(). After sending out the decision, the server must wait for acknowledgement from each user node. If acknowledgements are not received by the server within 6 seconds, the server resends its decision to every user node again until all acknowledgements are received. The acknowledgements from each user node are recorded in a hashmap, ackMap, which maps each user node to an ACK. This is to ensure that when the decision is sent out again, acks from the same user node are not double counted.

When a user node receives the server decision, it either deletes the source files and unlocks the sources for a commit decision or simply unlocks source files. In both cases, acknowledgement is sent back to the server.

When handling server crashes, a recovery routine is called in main() which is implemented by the recoverState() function. To do this, the log file is read using a file input stream and an object input stream. The server retrieves the state class which contains all of the necessary information to resume from where the server crashed. If the decision retrieved from the state is to commit, then the server commits the collage by calling commitCollage(). In either case, the decision is broadcasted until all acknowledgements are received.

Similarly, on the user node side, information about the phase is retrieved from the log file. If the phase indicates the "Prepare" phase, the vote that the user made before crashing is sent to the server. Again, in the case that the phase indicates either abort or commit, the same routine (delete source files and unlock sources or just unlock sources) is followed and then acknowledgement is sent to the server.

For logging information, a class called state is used to log important information. State includes two constructors: ose for the server and one for the user node. The server state constructor logs the following information: collage file name, last decision made, hashmap mapping each user node to the sources it contributes, hashmap of user node to acknowledgement, hashmap mapping collage filename to all of its sources, and image array. The user node state constructor logs the following information: vote, String array of each user node's sources, and the phase. This information is sufficient to recover the state and resume the process. In order to log, a new random access file is created called "userState.log" or "serverState.log" and a file output stream is created using the random access file. Finally an object output stream is created using the file output stream and the state is written into the object output stream. The random access file and object output stream are closed.