

Class	
OpenDocument	Details of an open document. This will be provided to you.
Server	<p>Main Server class. Responsible for the interaction with the EMU clients. This will be provided to you, let me know other methods that may be needed.</p> <p>Methods that can be called by clients:</p> <p>User login(String c, String logname, String password) void logout(String c, int userId) Boolean addDocument(String client, int userId, String docName, String docText) Boolean removeDocument(String client, int userId, String docName) OpenDocument openDoc(String client, int userId, String docPath) void closeDoc(String client, int userId, int docId) Boolean lockRequest(String client, int userId, int docId, AtomId ald) void lockRelease(String client, int userId, int docId, AtomId ald) void updateDoc(String client, int userId, int docId, CommandC[] cmds)</p> <p>Methods the Server calls the Client</p> <p>EMUDoc - void lockAtom(int serverId, int docId, AtomId ald) EMUDoc - void unlockAtom(int serverId, int docId, AtomId ald) EMUDoc - void addClientToDoc(int serverId, int docId, int userId) EMUDoc - void removeClientFromDoc(int serverId, int docId, int userId) EMU - void updateDoc(int serverId, int docId, CommandC[][] mCmds) EMU - void updateServerFreq(int serverId, int serverFreq)</p>
ServerMonitor	Shows the details of the Server operations. This will be provided to you.
ServerSetup	Allows the configuration of the Server. This will be provided to you.
ServerW	Server main window. Allows the opening of Setup and Monitor. This will be provided to you.

Class	
EMU	This will be provided to you. Needs interaction with the Server, management of the threads that update the open docs after receiving of updateDoc call from the server.
Atom	Defines the Atom abstraction. This will be provided to you. Complete the code of some methods.
AtomId	This will be provided to you. Should be complete, modify (and report) as needed.
Command	This will be provided to you. Should be complete, modify (and report) as needed.
CommandC	This will be provided to you. Should be complete, modify (and report) as needed.
EMUDoc	This will be provided to you. Needs: editing commands while updating the history, needs to track user actions on the viewing document together with the EMUW class.
History	This will be provided to you. Needs the remaining compacting rules, undo and redo.
Part	This will be provided to you.
Pos	This will be provided to you.
PosC	This will be provided to you.

Class	
Selection	This will be provided to you.
User	This will be provided to you.
EMUW	This will be provided to you.
FileChooserW	Allows the opening and saving of Documents. Not yet complete. This will be provided to you.
FindW	Handles search and replace operations. The window and all related variables are done. You will have to implement the basic search and replace. No need to implement the options.
HelpW	Shows the EMU Help Window. This will be provided to you.
LoginW	Allow the client to log in the EMU Server. Not yet complete. This will be provided to you.
NewFileW	Creates a new (local/remote) document. Not yet complete. This will be provided to you.
PreferencesW	Specifies the EMU Preferences. You have to use these specifications in your implementation. This will be provided to you.
PrintFileW	Prints the current doc. Not yet complete. This will be provided to you.
ShortCutW	Shows the EMU client shortcuts in effect. This will be provided to you.
UsersW	Show the current list of users editing a given document. Selecting one of the users highlights all atoms he currently holds a lock. You must implement this class.