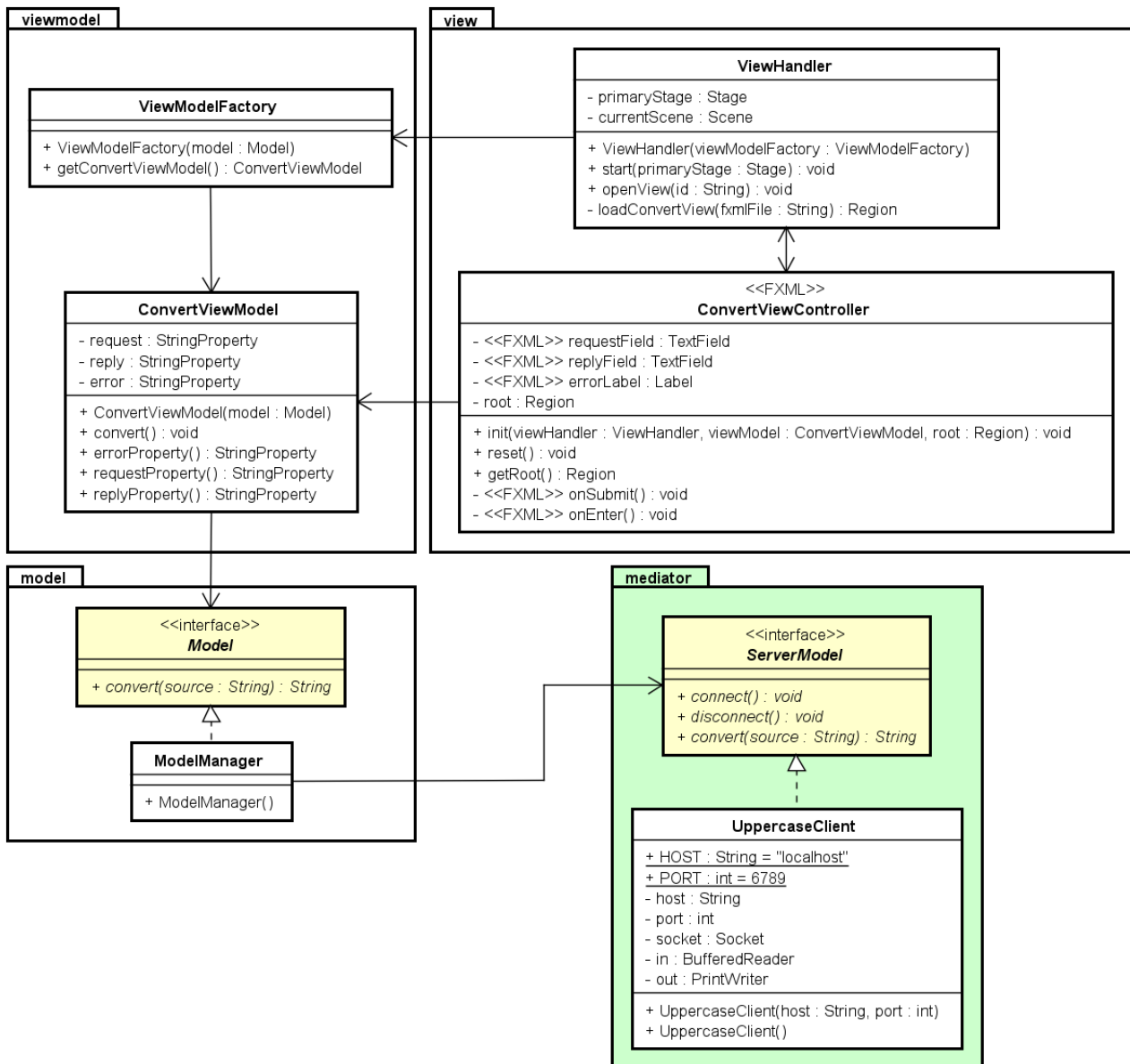


Exercise: Uppercase client (Request/reply only)

A simple console server is given in the file `Sockets-Uppercase-Server.zip`. The server is a multithreaded TCP socket server on port 6789, it accepts multiple incoming clients and repeatedly reads a `String` from a `BufferedReader` stream and reply with an uppercase `String` in a `PrintWriter` stream.

The purpose for this exercise is to implement an MVVM client application connecting to and getting and showing uppercase replies from the server. See the class diagram below and the steps on the next page.



Step 0: Create a new IntelliJ module (`Uppercase-SocketServer-Console`) including all server classes from the zip file.

Step 1: Create a new IntelliJ module (Uppercase-SocketClient-MVVM) and copy your solution to the (single user) one window version of the MVVM-Uppercase exercise (not the two window version).

Step 2: Define the following interface (in package mediator):

```
package mediator;

import java.io.IOException;

public interface ServerModel
{
    public void connect() throws IOException;
    public void disconnect() throws IOException;
    public String convert(String source) throws IOException;
}
```

Step 3: Implement class UppercaseClient (in package mediator)

- The class implements interface ServerModel
- Instance variables for host, port, socket, BufferedReader and PrintWriter
- Constants / final class variables (static final) for host and port, initialized to "localhost" and 6789, respectively
- Two-argument constructor setting host and port
- Zero-argument constructor setting host and port to the values of the two static final fields
- Method connect creating a socket with host and port and getting in- and out-put streams from the socket (BufferedReader and PrintWriter objects)
- Method disconnect calling close for both streams and the socket object.
- Method convert printing the string from the method parameter to the output stream (connected to server) and returning the string received calling readLine on the input stream

Step 4: Modify class ModelManager (in package model)

- Remove all connections to class Converter (the model class)
- Make an instance variable of type ServerModel
- In the constructor, initialize the instance variable to a new object of type UppercaseClient and call connect (on the instance variable object) – and define a throws clause.
- In the convert method, delegate to the instance variable calling its convert method. Also, define a throws clause.

Step 5: Delete class Converter (i.e. the client side has no local model)

Step 6: Modify method start in class MyApplication to include a try-catch block with all previous statements in the try-block

Step 7: Run the Server and then run the Client (or more clients) to test your solution