

This week, we mainly focused on writing Software Design Report. We almost finished designing all class descriptions, its instance variables and member functions.

Some modifications on Logger component has been considered with Apache Log4j, so Logger component has changed completely. Now, it conforms to singleton design pattern, which provides an only one Logger class instance for all components of the system.

```
import org.apache.log4j.*;
public class MyLogger {
    private static Logger logger;

    public static synchronized Logger getSingletonObject() {
        if(logger == null) {
            logger =Logger.getLogger("Logger Component");
        }
        return logger;
    }
}
```

We have added multi-threaded capability for our project. A user can do multiple Applet to JSF conversions simultaneously and watch their logs from different log windows.

We have arranged a meeting with Siemens and listened their thoughts on our project. We both agreed that building a database for a developer tool is not a good design. Overheads of a database in our project outweigh its advantages. So, there won't be any database table.

After Java ML's run in our project, we need to keep applet class info. Because of this need, we decided to use Java Reflection API to observe class's characteristics, methods, fields etc. It is still under research stage.

We deeply examined JavaML tags. Since, JavaML has been developed in 2000. it does not answer new Java syntax. We will overcome this problem with considering a new tool instead JavaML if necessary.