```cpp
// Copyright 2023 Thomas O’Connor
#include "LogParser.hpp"

const string LogParser::generateRPT() const {
    // get all lines of refrence text
    string line, outputString, startDate, endDate;
    unsigned int lineNum = 0;
    bool booting = 0;
    std::regex rgx1("\\(log\\.c\\.166\\) server started"),
               rgx2("oejs\\.AbstractConnector:Started SelectChannelConnector@0\\.0\\.0\\.0:9080");
    while (std::getline(_referenceLog, line)) {
        lineNum++;
        // if "server started" is found, boot began at current line
        if (std::regex_search(line, rgx1)) {
            // early termination of previous boot sequence. End and restart log
            if (booting) {
                outputString.append("**** Incomplete boot ****\n\n");
            }
            booting = true;
            outputString.append("=== Device boot ===\n");
            outputString.append(std::to_string(lineNum));
            outputString.append("(" + _path + "): ");
            startDate = line.substr(0, 19);
            outputString.append(startDate);
            outputString.append(" Boot Start\n");
        // boot process terminated at current line
        } else if (std::regex_search(line, rgx2)) {
            booting = false;
            outputString.append(std::to_string(lineNum));
            outputString.append("(" + _path + "): ");
            endDate = line.substr(0, 19);
            outputString.append(endDate);
            outputString.append(" Boot Completed\n");
            outputString.append("\tBoot Time: ");
            // calculate difference in time in ms
            // construct the times from the strings directly
            ptime t1(time_from_string(startDate)), t2(time_from_string(endDate));
            boost::posix_time::time_duration td = t2 - t1;
            outputString.append(std::to_string(td.total_milliseconds()) + "ms\n\n");
        }
    }
    // outside of the loop: reached end of file
    if (booting) {
        outputString.append("**** Incomplete boot ****\n\n");
    }
    return outputString;
}
```