**Meeting Minutes**

**Date:** March 14, 2014

**Start Time:** 6:00pm

**End Time:** 9:00pm

**Members Present:** Drew Aaron, Michael Beaver, Clay Boren,

Chad Farley, Andrew Hamilton, and Travis Hunt

**Members Absent:** N/A

**Topics** **Discussed**

* Frontend Custom Component Development
* Backend Detailed Design Pseudocode

**Decisions and Actions Taken**

Drew developed a custom Source Code Editor component for the Frontend user interface. He implemented line numbers that automatically update when the user inserts lines, and he added static column numbers. Unfortunately, the line number implementation is defective and inefficient, so a new solution is necessary. Currently Drew has two labels on a status bar that display the line number and column number at the cursor’s current position. The team will negotiate solutions and options with the client at the next client meeting.

Chad, Andrew, Michael, and Travis developed the machine op table pseudocode. The machine op table will essentially be a special type of hash table with five columns: Opcode; OpType; Operand1Type; Operand2Type; and, Operand3Type. The contents of the machine op table will be initialized by the constructor. The machine op table will also be its own class rather than inherit from the abstract Table class because its needs and functionality are too dissimilar to those of the SymbolTable and the LiteralTable classes. See the attached design particulars.

The next team meeting will be Saturday, March 15, 2014 at 2:00pm in the Christian Student Center.

**Supplementary Information**

**Machine Op Table Data Members**

const int NUM\_INSTRUCTIONS = #;

const int NUM\_OP\_TABLE\_COLS = 5;

const int OPCODE\_COL = 0;

const int OPTYPE\_COL = 1;

const int OPERAND1\_COL = 2;

const int OPERAND2\_COL = 3;

const int OPERAND3\_COL = 4;

string opTable[NUM\_INSTRUCTIONS][NUM\_OP\_TABLE\_COLS];

**Machine Op Table Methods Pseudocode**

public GetOperand1Type(int index)

Return opTable[index][OPERAND1\_COL];

public GetOperand2Type(int index)

Return opTable[index][OPERAND2\_COL];

public GetOperand3Type(int index)

Return opTable[index][OPERAND3\_COL];

public GetOpType(int index)

Return opTable[index][OPTYPE\_COL];

public IsOpcode(string opcode)

Get index using Hash(opcode)

If opTable[index][OPCODE\_COL] is opcode, return index

Return -1

private Hash(string opcode)

Assign index in opTable based on mathematical hash of opcode

Return index

**Machine Op Table Opcode Types**

See the software requirements specification document.

**Machine Op Table Operand Types**

R = Register

D = Doubleword

F = Fullword

H = Halfword

C = Character

X = Hexadecimal

Z = Zoned

P = Packed

B = Binary

D(X) = Memory addressing

D(X,B) = Memory addressing

NULL = None