

VENDING MACHINE APPLICATION

TEAM 6



May 15, 2018

North-West University

ITRW313

Contents

[**Planning documents** 2](#_Toc514192863)

[**Problem statement** 3](#_Toc514192864)

[**User manual with adjustments/additions** 3](#_Toc514192865)

[**The Finite State Machine for the vending machine** 3](#_Toc514192866)

[**Diagrams** 4](#_Toc514192867)

[**Code** 5](#_Toc514192868)

[**Team 6** 6](#_Toc514192869)

# **Planning documents**

Upon receiving the assignment on the 27th of March, we proceeded to make a Whatsapp group for all participants so communication could flow more easily and we could get the project done.

We then proceeded to split the tasks, namely Documentation (problem statement, diagrams, planning documents, end user documentation) and creating the CLIPS source code amongst group members. Nonhlanhla created branches in github for code to be backed up to; should there be an event where errors occur, it would not affect our main program.

On the 11th of April we had a group meeting whereby we discussed each person’s role: people that were good with documentation would do documentation and everyone would try help out with the code as we are all fairly new to CLIPS as a programming language.

Below is listed each document and a brief description of what it should entail:

1. Problem statement: setting out the goals which we are trying to accomplish with our finite state machine
2. Diagrams: compiled documentation of all diagrams used to explain the system and screenshots of the actual application
3. Planning documents: (current document) schedule of when each document should be made, what they entail and how they will be used to implement the system
4. End user documentation: should give the user instructions on how to download and use the program we are constructing.

All documents should be compiled and submitted on or before the 18-Apr-2018 23:55.

# **Problem statement**

The machine state space considers the purchase of a soft drink from the machine by adding coins in the machine. The machine accepts Quarters = 25 C and nickel = 5 C, a soft drink in total costs 25 C and no change is returned. The soft drink machine does not accept dimes as it causes complications, if someone insert a dime in the machine the response is undefined.

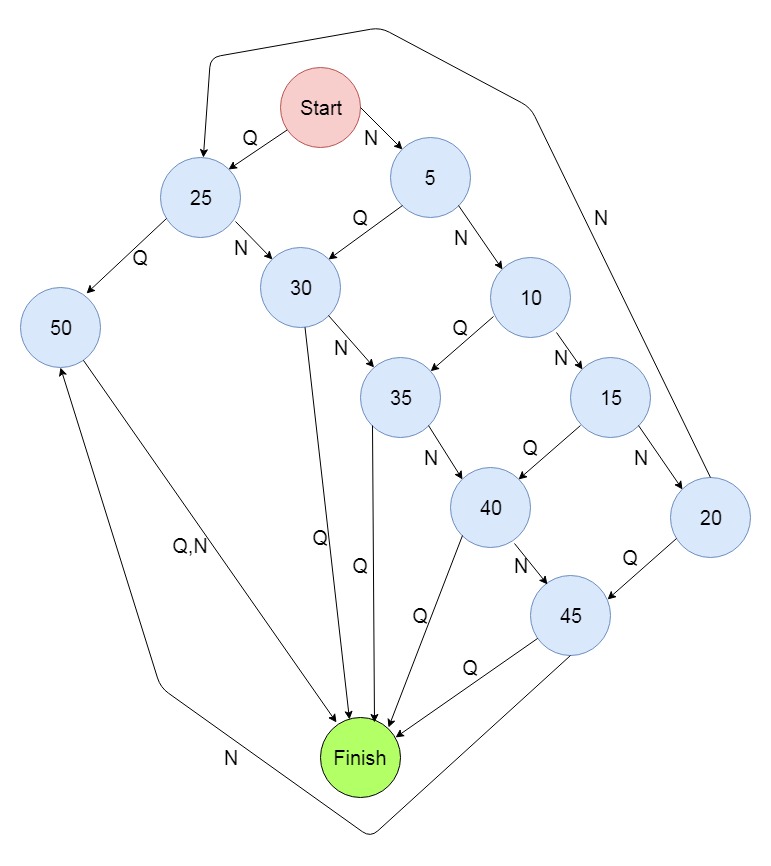
# **User manual with adjustments/additions**

* Download Clips 🡪 *clipsrules.sourceforge.net/Version63Beta.html*
* Follow the instruction prompts until clips is successfully installed onto your computer.
* Once it is installed you can load your .clp file into clips to check if it’s working.

## **The Finite State Machine for the vending machine**

* The machine allows the user to make a selection of various soft drinks, sweets, chocolates, and snacks they would like to purchase.
* The machine allows R5, R2, R1, 50c, 20c and 10c coins.
* The machine accepts all notes.
* Once the user has inserted the right amount of notes and coins to make their purchases, the vending machine will release the item(s) selected and give out change if necessary.

# **Diagrams**



# **Code**

|  |
| --- |
| ;.\*; |
|  |

|  |
| --- |
| ; class VendingMachineAssignement { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| (deftemplate item "items in the machine" |
|  |

|  |
| --- |
| (slot itemNumber) (slot itemName) (slot amount) ) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| (deftemplate coin "different coins" |
|  |

|  |
| --- |
| (slot coinAmount)) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| (deffacts initial-facts "some initial facts" |
|  |

|  |
| --- |
| (item (item1) (cola) ((8.50)(d?n2))) |
|  |

|  |
| --- |
| (item (item2) (sweet) ((12.50) (d?n2))) |
|  |

|  |
| --- |
| (item (item3) (orange) ((10.00)(double))) |
|  |

|  |
| --- |
| (item (item4) (chocolate) (15.00)) |
|  |

|  |
| --- |
| ) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| (deffacts Coint-facts |
|  |

|  |
| --- |
| (coin (R 5)) |
|  |

|  |
| --- |
| (coin (R 2)) |
|  |

|  |
| --- |
| (coin (R 1)) |
|  |

|  |
| --- |
| (coin (50 c)) |
|  |

|  |
| --- |
| (coin (20 c)) |
|  |

|  |
| --- |
| (coin (10 c)) |
|  |

|  |
| --- |
| ) |
|  |

|  |
| --- |
| (defrule seletectItem "" |
|  |

|  |
| --- |
| => |
|  |

|  |
| --- |
| (printout t "Enter an item : ") |
|  |

|  |
| --- |
| ) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| (defrule AmountPaid |
|  |

|  |
| --- |
| ) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| (defrule print-solution |
|  |

|  |
| --- |
| item1 <- (solution AmountPaid ?n1 1) |
|  |

|  |
| --- |
| item2 <- (solution item ?c1 1) |
|  |

|  |
| --- |
| item3 <- (solution AmountPaid ?n2 2) |
|  |

|  |
| --- |
| item4 <- (solution item ?c2 2) |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| => |
|  |

|  |
| --- |
| (retract item1 ?item2 ?item3 ?itme4 ) |
|  |

|  |
| --- |
| (format t "# |%-11s|%12s%n" AmountPaid + seletectItem ) |
|  |

|  |
| --- |
| (format t "===========================================%n") |
|  |

|  |
| --- |
| (format t "1 |%-11s|%-12s%n" ?n1 ?c1) |
|  |

|  |
| --- |
| (format t "2 |%-11s|%-12s%n" ?n2 ?c2) |
|  |

|  |
| --- |
| (format t "3 |%-11s|%-12s%n" ?n3 ?c3) |
|  |

|  |
| --- |
| (format t "4 |%-11s|%-12s%n" ?n4 ?c4) |
|  |

(format t "%n"))

# **Team 6**

27299066 LR Mogomotsi

26720175 NA Mkhize

26915804 M MATEBESI

23999969 M MOEKETSI

**2647397 MM Mofokeng**

**25618717 KE Moeti**