



الجامعة الإسلامية العالمية ماليزيا  
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA  
يُونِيسَيْتِي إِسْلَامِيَّةٌ أَنْتَارَاغُشَا مَلِيسِيَا

*Garden of Knowledge and Virtue*

---

**KULLIYAH OF INFORMATION & COMMUNICATION TECHNOLOGY**

---

**CSCI 3301 COMPUTER ARCHITECTURE AND ASSEMBLY LANGUAGE**

**SEMESTER 1, 2021/2022**

**SECTION 3**

**GROUP PROJECT**

Parcel Delivery Alert App

**PREPARED BY:**

NAME	MATRIC NO.
MD RAKIBUL HASSAN	1720465
MOHAMED MOUBARAK MOHAMED MISBAHOU MKOUBOI	1820705

**LECTURER**

ASST. PROF. DR. HAFIZAH MANSOR

Table of Contents:

<b>1. Abstract</b>	<b>3</b>
<b>2. Introduction</b>	<b>3</b>
<b>2.1 Work Flow:</b>	<b>3</b>
<b>3. Pseudocode</b>	<b>4</b>
<b>4. Flow-chart</b>	<b>5</b>
<b>5. Obstacles</b>	<b>6</b>
<b>5.1 Summary</b>	<b>6</b>
<b>5.2 References</b>	<b>6</b>
<b>6. Source Code</b>	<b>7</b>

## 1. Abstract

As our lives grow increasingly dependent on electronic gadgets such as smartphones and laptops, we can't imagine life without them, and these products have shown to be very helpful and easy to use so far. Electronic transactions are a highly prevalent method of purchasing products online from a variety of companies located all over the globe. It's now a question of minutes. Additionally, we are more likely to spend time on social media. Individuals are even sharing their struggles over social media rather than in person. In a word, we have entered the virtual age. While the virtual environment offers several benefits, one major concern is the security of sharing all of one's personal information in this environment. That is why clients depend on SMS notifications when their deliveries arrive, enabling them to manage their purchases without leaving their homes. We wanted to design a Parcel Delivery Alert App that would be available only to the confirmed user and the delivery person due to the security issue.

## 2. Introduction

Our goal is to develop a package delivery alert system that is accessible exclusively to verified users. We selected assembly language since this course is about computer architecture and assembly language. We accomplish our intended task using assembly language (MIPS). The MIPS assembly language is a non-interlocked pipeline assembly language based on an instruction set and computer architecture for microprocessors. It is a low-level programming language created by John Hennessey. It has a distinct syntax. Unlike other programming languages such as C++, Java, and Python, it is not object-oriented. Because of its flexibility in application, it is referred to as a low-level language. We attempt our system to make use of this language and to create a user-friendly application.

### 2.1 Work Flow:

Our system is essentially a parcel delivery system in which delivery personnel must enter the recipient's login and address to get access to the system.

In the initial stage, there will be a built-in delivery app available to consumers and delivery drivers who wish to utilize our system. To get access to the system, they must input the login and password. But our major concern is not with validating the username and password, but with checking the entire name of both the delivery person and the recipient and then checking their respective addresses.

In the following stage, they will discover some other possibilities for interacting with the system. Checking the address is one of the alternatives. So, if a user is dissatisfied with the

address or believes that the address is incorrect, he or she can modify it. Choose an option and request that the delivery person re-enters his address accurately.

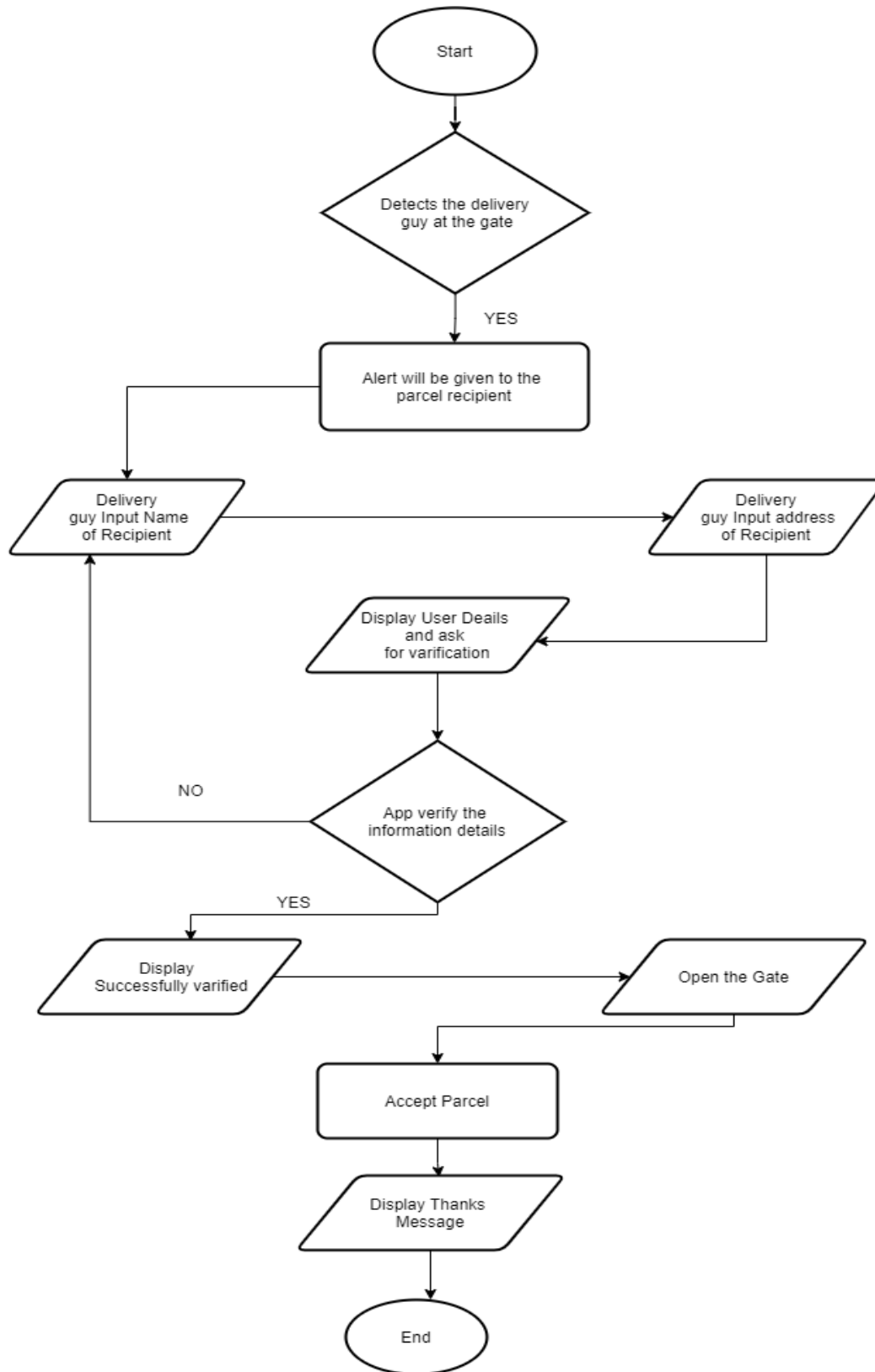
If an unauthorized person enters the system and attempts to modify the address or name. As a result, if a user changes his or her name or address, he or she must enter his or her current address first, followed by his or her entire name. Only if it matches may he/she advance.

The final step is to quit the system, after which all data will be stored with the new address and name.

### 3. Pseudocode

- 1) Show a welcoming message
- 2) Show a detective message of the delivery guy
- 3) Prompt message to show that the user has received an alert message
- 4) Prompt message to tell the delivery guy to put the name of the recipient user
- 5) Prompt message to tell the delivery guy to put the address of the recipient user
- 6) Show a message of the recipient user details
- 7) Prompt the user to select options either 'Y/y' or 'N/n' which means yes and no for verification of the identity
- 8) If the user-selected option (1) then, display step 4 again.
- 9) If the user-selected option (2) then displays a successful message.
  - a) Prompt a message telling the delivery guy to open the gate.
  - b) Prompt a message telling the delivery guy and the recipient user that the parcel has been accepted.
  - c) Display THANKS FOR YOUR SERVICE!! message

## 4. Flow-chart



## 5. Obstacles

One of the most significant issues we encountered was in the code section. We must put in a lot of effort to master it because it is a low-level language. After that, no matter how many times we coded, we couldn't get the required outcome. For us, it was a very early start. We are gradually coming to terms with it. Because this language is not commonly used nowadays, there aren't many references or resources to help us fix our mistakes.

### 5.1 Summary

The primary goal of this course is to familiarise ourselves with assembly language. And the project's goal is to study, practice, and understand how to use assembly code to solve a real-world problem. Working on this project was a very beneficial experience for all of us. We discussed assembly language and gained a practical understanding of it, which will aid us in our future work in this field.

Using assembly language, we successfully constructed our Parcel Delivery Alert system (MIPS). The system can notify the user if his package has arrived, which was our primary goal. Though it is merely a delivery portion interface, it may be used in any major system to safeguard products and prevent robbers from stealing someone else's shipment. Adding some more options will help this method be more adaptable.

### 5.2 References

- ❖ Websites: stackoverflow.com, tutorialspoint.com, and geeksforgeeks.com  
Link: <https://stackoverflow.com/how-to-store-user-input-string>
- ❖ People: ASST. PROF. DR. HAFIZAH MANSOR, Br Parves Sheik( He dropped course middle of the semester)
- ❖ Slides and videos are provided in the class.

## Source Code

```
#MOHAMED MOUBARAK MOHAMED MISBAHOU MKOUBOI 1820705
#MD RAKIBUL HASSAN 1720465
```

```
#GROUP PROJECT on Parcel Delivery Alert App
```

```
.data
```

```
welcomemsg: .ascii "Welcome to our Parcel Delivery Alert App "
detectmsg:  .ascii "\nWe have detected a delivery guy at the gate "
alertmsg:   .ascii "\nThe recipient has already received an alert "
successfulmsg: .ascii "\nSuccessfully verified .... "
opengatemsg: .ascii "\nPlease open the gate ..... "
acceptmsg:   .ascii "\nParcel accepted "
thanksmg:    .ascii "\nTHANKS FOR YOUR SERVICE!! "
getinput1:   .ascii "\nPlease to enter the name of the recipient of the parcel: "
getinput2:   .ascii "Please to enter the mahallah of the recipient of the parcel: "
verifymsg:   .ascii "\nIs this your address? (Y/N)/(y/n) "
addressmsg:  .ascii "\nIUM Campus, Gombak, Kuala Lumpur "
buffer:      .space 20
```

```
.text
```

```
#    Print out a message string with a welcome message
```

```
    li $v0,4
    la $a0, welcomemsg
    syscall
```

```
#    Print out a message string with a detect message
```

```
    li $v0,4
    la $a0, detectmsg
    syscall
```

```
#    Print out a message string with an alert message
```

```
    li $v0,4
    la $a0, alertmsg
    syscall
```

```
#    Print out a message string with the message "Please to enter the name of the recipient
of the parcel "
```

option1:

```
li $v0,4
la $a0, getinput1
syscall
```

# Get input from the delivery guy to enter the name of the recipient of the parcel

```
li $v0,8 #take in input
la $a0, buffer #load byte space into address
li $a1, 20 # allot the byte space for string
move $t0,$a0 #save string to t0
syscall
```

# Print out a message string with the message "Please to enter the mahallah of the recipient of the parcel "

```
li $v0,4
la $a0, getinput2
syscall
```

# Get an input from the user to enter the address of the recipient of the parcel

```
li $v0,8 #take in input
la $a0, buffer #load byte space into address
li $a1, 20 # allot the byte space for string
move $t0,$a0 #save string to t0
syscall
```

# Print out a message string with the recipient user details

```
la $a0,addressmsg #load and print "addressmsg" string
li $v0,4
syscall
```

```
la $a0, buffer #reload byte space to primary address
move $a0,$t0 # primary address = t0 address (load pointer)
li $v0,4 # print string
syscall
```

# Print out another message string to ask the recipient to select options either 'Y' or 'N'  
# which means yes and no for verification of the identity

```
li $v0,4
la $a0, verifymsg
syscall
```



```
#      If the user selects N, the program loops to step 2. Otherwise, go to the next line.  
#compare
```

```
    li $v0,12  
    syscall  
    li $t0, 'N'  
    beq $t0,$v0,option1 #if user enters N, then go to option 1
```

```
    li $v0,12  
    syscall  
    li $t0, 'Y'  
    beq $t0,$v0,option2 #if user enters Y, then go to option 2
```

```
option2:
```

```
#      Print out a message string with a successful message
```

```
    li $v0,4  
    la $a0, successfulmsg  
    syscall
```

```
#      Print out a message string with an open gate t message
```

```
    li $v0,4  
    la $a0, opengatemsg  
    syscall
```

```
#      Print out a message string with an accept parcel message
```

```
    li $v0,4  
    la $a0, acceptmsg  
    syscall
```

```
#      Print out a message string with a thanks message
```

```
    li $v0,4  
    la $a0, thanksmg  
    syscall
```

```
#      End the code
```

```
end:  
    li $v0,10  
    syscall
```