

**Course Name:**

**COMPUTER ARCHITECTURE AND ORGANIZAION (LAB)**

**Project Name:**

**EASYPAISA MIPS APPLICATION**

**Class and Section:**

**BSE 3-B FALL 2020**

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **STUDENTS NAME** | **ENROLMENT NO.** |
| **1** | **HAMZA ARSHAD** | **02-131192-070** |
| **2** | **MUHAMMAD SOHAIL ABBAS** | **02-131192-012** |
| **3** | **RAFI ZAMAN** | **02-131192-084** |
| **4** | **ABDUL HANNAN** | **02-131192-058** |

**Submitted To:**

**ENGR. MUHAMMAD REHAN BAIG**

1. **TABLE OF CONTENTS**
2. [**Abstract:** 3](#_Toc62178207)
3. [**Introduction:** 3](#_Toc62178208)
4. [**Description with Code:** 3](#_Toc62178209)
   1. [**.data:** 3](#_Toc62178210)
   2. [**.text:** 4](#_Toc62178211)
      1. [**Login:** 4](#_Toc62178212)
      2. [**Account:** 6](#_Toc62178213)
      3. [**Account Balance:** 7](#_Toc62178214)
      4. [**Balance Withdraw:** 8](#_Toc62178215)
      5. [**Balance Deposit:** 10](#_Toc62178216)
      6. [**Electric Bill:** 11](#_Toc62178217)
      7. [**Tickets:** 13](#_Toc62178218)
      8. [**Sui Gas Bills:** 15](#_Toc62178219)
      9. [**Internet Bills:** 17](#_Toc62178220)
      10. [**Ordering Foods:** 19](#_Toc62178221)
      11. [**Exit:** 22](#_Toc62178222)
5. [**Conclusion:** 23](#_Toc62178223)

# **Abstract:**

**Easy paisa MIPS Application** is such a user-friendly application. In which user can easily withdraw his/her cash, deposit cash, send or receive cash. If he/she has to pay bill, then there is no worry he/she can pay bill through this application. If he wants to get some energy, then he/she can order food by performing few steps. User wants some journey he/she can buy tickets of different classes.

# **Introduction:**

In this era, people want smart works for doing their tasks. Therefore, we make such application by which user can easily send, receive or withdraw cash. He/she can also order food, pay different bills etc.

**Operations:**

1. Account
2. Account Balance
3. Balance Withdraw
4. Balance Deposit
5. Electric Bill
6. Tickets
7. Sui Gas Bills
8. Internet Bills
9. Ordering Foods
10. Exit

# **Description with Code:**

## **.data:**

In .data section we have initialized the variables which we have used in our program.

.data

strinput1 : .space 16

strinput2 : .space 16

stringinput3 : .space 16

giveorder: .asciiz "please enter what u want to order:\n"

username : .asciiz "HAMZA"

password : .asciiz "BUKC123"

balance : .word 0

login\_prompt: .asciiz "\nProvide the username : \n"

paswd\_prompt: .asciiz "\nProvide the password :\n "

loginerror\_prompt: .asciiz "\nInvalid crendentials. Please retry."

login\_label:"LOGIN PAGE"

welcome\_prompt: .asciiz "\n\n WELCOME TO EASYPAISA MIPS APPLICATION\n "

action\_prompt: .asciiz "\n 1. Check your balance\n 2. Deposit Money\n 3. Withdraw Money\n 4. Electric Bill Payment\n 5. Ticket Payment\n 6. Sui Gas Payment\n 7. Internet Bill Payment\n 8. Order Food Payment\n 9. Exit\nSelect menu option : "

menu: .asciiz "\n 1. Chicken Roll RS-100\n 2. chicken burger RS-120\n 3. White karahi RS-800per-kg\n 4. California-pizza RS-1200\n 5. Fresh bread RS-100\n 6. California Fries RS-150/bucket\n"

tickets: .asciiz "\n 1. super service RS-1000\n 2. Super Deluxe Service RS-1500\n 3. Youtang Service RS-1800\n 4. Vip A/C Service RS-2000\n 5. Luxury Business Class Service RS-2200\n"

balance\_prompt: .asciiz "\nYour current balance is $ \n"

withdraw\_prompt: .asciiz "\nEnter the amount to withdraw :\n "

insufficient\_prompt: .asciiz "\nYou dont have suffucient balance.\n"

deposit\_prompt: .asciiz "\nEnter the amount to deposit :\n "

logout\_prompt: .asciiz "\nSuccessfully logged out."

action\_invalid: .asciiz "\nUnknown input provided. Valid inputs are 1-5."

pay: .asciiz "do you wanted to ELECTRIC BILL pay\n1.yes\n2.no\n"

Spay: .asciiz "do you wanted to SUI GAS pay\n1.yes\n2.no\n"

Ipay: .asciiz "do you wanted to SUI GAS pay\n1.yes\n2.no\n"

BUYTICKETS: .asciiz "do you wanted to buy tickets\n1.yes\n2.no\n"

ORDERFOOD: .asciiz "do you wanted to order food\n1.yes\n2.no\n"

en: .asciiz "enter amount: "

Sn: .asciiz "enter amount: "

In: .asciiz "enter amount: "

OF: .asciiz "enter amount OF food which u want to buy: "

BT: .asciiz "enter tickket package amount you want to avail: "

no: .asciiz "You have zero balance"

p: .asciiz "\nBill paid successfully"

O: .asciiz "\n ORDER FOOD successfully"

T: .asciiz "\n TICKET PURCHASED successfully"

more: .asciiz "do you want to withdraw again:\n1.yes\n2.no\n "

## **.text:**

In .text this we code our program.

### **Login:**

Coming towards our **Login Page**, user has to put his **username** and **password** in order to use our **Easypaisa Application**. If he gives correct username and password then he moves forward and in case of wrong detail application shows a message of **Invalid Credentials Please Retry !!.**

login:

la $a0, login\_label # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

la $a0, login\_prompt # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

la $a0, strinput1 # load address of input buffer for syscall

li $a1, 16 # Maximum number of the length of read string service

li $v0, 8 # specify read string service

syscall # Read the string.

la $a0, paswd\_prompt # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

la $a0, strinput2 # load address of input buffer for syscall

li $a1, 16 # Maximum number of the length of read string service

li $v0, 8 # specify read string service

syscall # Read the string.

la $a0, username # load address of the username

la $a1, strinput1 # load address of input buffer for username

jal StringCompare

bne $v0, 0 , login\_incorrect

la $a0, password # load address of the password

la $a1, strinput2 # load address of input buffer for password

jal StringCompare

bne $v0, 0 , login\_incorrect

j account

login\_incorrect:

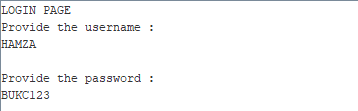
la $a0, loginerror\_prompt # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

j login

#### **Output:**

****

### **Account:**

At account section, first our application welcomes our user by showing the message of **WELCOME TO EASYPAISA MIPS APPLICATION** and show his/her username. Then shows list of options which user wants to perform which includes the given below options.

account:

la $a0, welcome\_prompt # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

la $a0, username # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

la $a0, action\_prompt # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

li $v0, 5 # specify Read Integer service

syscall # Read the number. After this instruction, the number read is in $v0.

beq $v0, 1 , account\_balance

beq $v0, 2 , account\_deposit

beq $v0, 3 , account\_withdraw

beq $v0, 4 , EBILL

beq $v0, 5 , Buy

beq $v0, 6 , SBILL

beq $v0, 7 , IBILL

beq $v0, 8 , order

beq $v0, 9 , exit

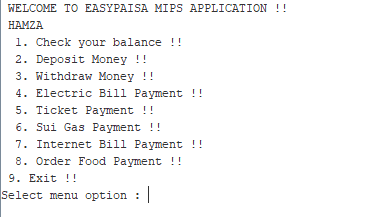
la $a0, action\_invalid # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

j account

#### **Output:**



### **Account Balance:**

Moving towards **Account Balance** if user has some balance then it shows balance to user. Incase of no balance it shows a message of **Your current balance is $0.**

account\_balance:

la $a0, balance\_prompt # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

la $t0, balance

lw $a0, ($t0) # load the integer to be printed

li $v0, 1 # specify Print Integer service

syscall # print the balance number

j account

#### **Output:**



### **Balance Withdraw:**

If user wants to withdraw some cash then he selects **Withdraw Money** option and withdraw the required amount which he/she wants. Application asks from the user to withdraw money again until he selects **No** option. If he/she has no balance then it shows the message that **You don’t have sufficient balance.**

account\_withdraw:

la $a0, withdraw\_prompt # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

li $v0, 5 # specify Read Integer service

syscall # Read the number. After this instruction, the number read is in $v0.

la $t0, balance

lw $t1, ($t0)

sub $t1, $t1, $v0

blt $t1, 0, withdraw\_insufficient

sw $t1, ($t0)

la $a0,more

li $v0,4

syscall

li $v0,5

syscall

beq $v0,1,account\_withdraw

j account\_balance

withdraw\_insufficient:

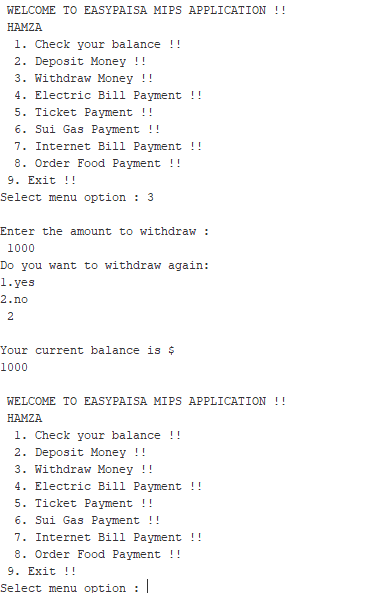
la $a0, insufficient\_prompt # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

j account\_balance

#### **Output:**



### **Balance Deposit:**

Incase of user has no balance then he selects this option **Deposit Money** and easily deposit cash in his/her account. After this application shows his/her balance in his/her account.

account\_deposit:

la $a0, deposit\_prompt # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

li $v0, 5 # specify Read Integer service

syscall # Read the number. After this instruction, the number read is in $v0.

la $t0, balance

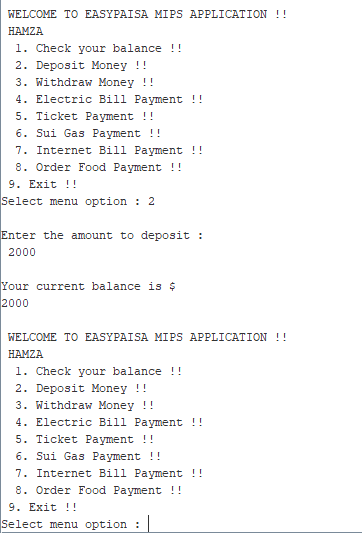
lw $t1, ($t0)

add $t1, $t1, $v0

sw $t1, ($t0)

j account\_balance

#### **Output:**



### **Electric Bill:**

As our application provides maximum ease to our user, therefor it has feature of pay bills. User can easily pay bill here. After this application displays the message of **Bill paid successfully.**

EBILL:

la $a0, pay # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

li $v0,5

syscall

beq $v0,1,yesh

beq $v0,2,nooo

yesh:

la $a0,en

li $v0,4

syscall

li $v0,5

syscall

la $t0, balance

lw $t1, ($t0)

sub $t1, $t1, $v0

blt $t1, 0, withdra\_insufficient

sw $t1, ($t0)

la $a0,p

li $v0,4

syscall

j account\_balance

withdra\_insufficient:

la $a0, no # load address of prompt for syscall

li $v0, 4 # specify Print String service

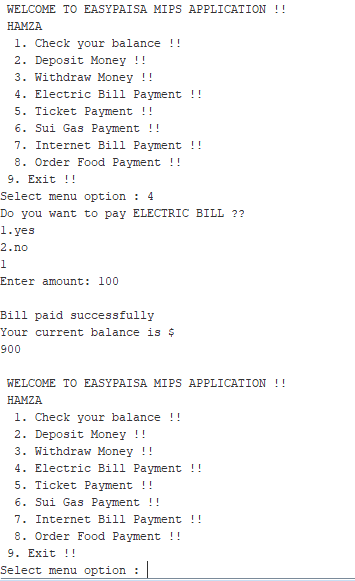
syscall # print the prompt string

j account

nooo:

j account # print the prompt string

#### **Output:**



### **Tickets:**

Application also provides feature of buying ticket online. User can buy ticket easily and after buying tickets application shows a message of **Ticket purchased successfully.**

Buy:

la $a0, BUYTICKETS # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

li $v0,5

syscall

la $a0, tickets # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

beq $v0,1,yes1

beq $v0,2,no1

yes1:

la $a0,en

li $v0,4

syscall

li $v0,5

syscall

la $t0, balance

lw $t1, ($t0)

sub $t1, $t1, $v0

blt $t1, 0, withdraw1\_insufficient

sw $t1, ($t0)

la $a0,T

li $v0,4

syscall

j account\_balance

withdraw1\_insufficient:

la $a0, no # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

j account

no1:

j account # print the prompt string

#### **Output:**



### **Sui Gas Bills:**

**Sui Gas Bills** are also paid in this application. After bill paid it shows a message **Bill paid successfully.**

SBILL:

la $a0, Spay # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

li $v0,5

syscall

beq $v0,1,yes2

beq $v0,2,no2

yes2:

la $a0,Sn

li $v0,4

syscall

li $v0,5

syscall

la $t0, balance

lw $t1, ($t0)

sub $t1, $t1, $v0

blt $t1, 0, withdraw2\_insufficient

sw $t1, ($t0)

la $a0,p

li $v0,4

syscall

j account\_balance

withdraw2\_insufficient:

la $a0, no # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

j account

no2:

j account # print the prompt string

#### **Output:**



### **Internet Bills:**

User also pays bills of **Internet**.After bill paid application shows a message of **Bill paid successfully.**

IBILL:

la $a0, Ipay # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

li $v0,5

syscall

beq $v0,1,yes3

beq $v0,2,no3

yes3:

la $a0,In

li $v0,4

syscall

li $v0,5

syscall

la $t0, balance

lw $t1, ($t0)

sub $t1, $t1, $v0

blt $t1, 0, withdraw3\_insufficient

sw $t1, ($t0)

la $a0,p

li $v0,4

syscall

j account\_balance

withdraw3\_insufficient:

la $a0, no # load address of prompt for syscall

li $v0, 4 # specify Print String service

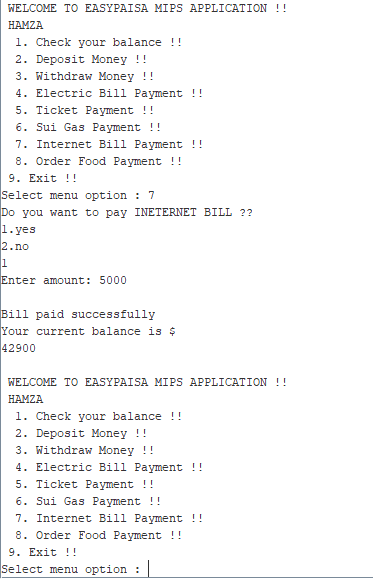
syscall # print the prompt string

j account

no3:

j account # print the prompt string

#### **Output:**



**Ordering Food :**

**Foods** are also ordered through this application. User has options which he wants to ordered. After order application shows a message **Food ordered successfully**

order:

la $a0, ORDERFOOD # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

li $v0,5

syscall

la $a0, foodmenu # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

li $v0, 5 # specify Read Integer service

syscall # Read the number. After this instruction, the number read is in $v0.

beq $v0, 1 , Chickenroll

beq $v0, 2 , Chickenburger

beq $v0, 3 , Whitekarahi

beq $v0, 4 , Californiapizza

beq $v0, 5 , Freshbread

beq $v0, 6 , exit

beq $v0,1,yes4

beq $v0,2,no4

Chickenroll:

la $a0, chickroll # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

Chickenburger:

la $a0, chickburger # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

Whitekarahi:

la $a0, whitekarahi # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

Californiapizza:

la $a0, calpizza # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

Freshbread:

la $a0, fb # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall

yes4:

la $a0,OF

li $v0,4

syscall

li $v0,5

syscall

la $t0, balance

lw $t1, ($t0)

sub $t1, $t1, $v0

blt $t1, 0, withdraw4\_insufficient

sw $t1, ($t0)

la $a0,O

li $v0,4

syscall

j account\_balance

withdraw4\_insufficient:

la $a0, no # load address of prompt for syscall

li $v0, 4 # specify Print String service

syscall # print the prompt string

j account

la $a0, action\_invalid # load address of prompt for syscall

li $v0, 4 # specify Print String service

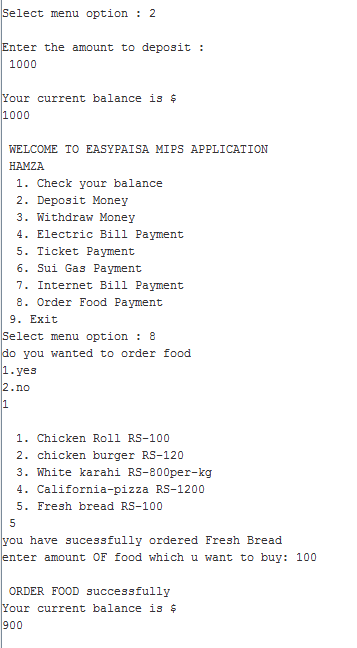
syscall # print the prompt string

j account

no4:

j account # print the prompt string

#### **Output:**



### **Exit:**

If user wants to exit from the application then he selects **Exit** option to logged out from application.

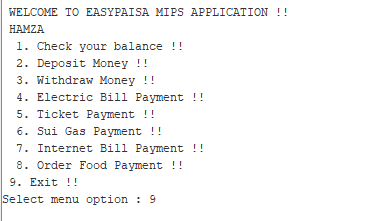
exit:

# The program is finished. Exit.

li $v0, 10 # system call for exit

syscall # Exit!

#### **Output:**



# **Concept used:**

1. Branches Concept
2. Load and store Technique
3. Functions or procedures
4. Loops

# **Conclusion:**

**Easy paisa MIPS Application** is a user-friendly application by which user can easily perform different task. It has all those features which users want nowadays.