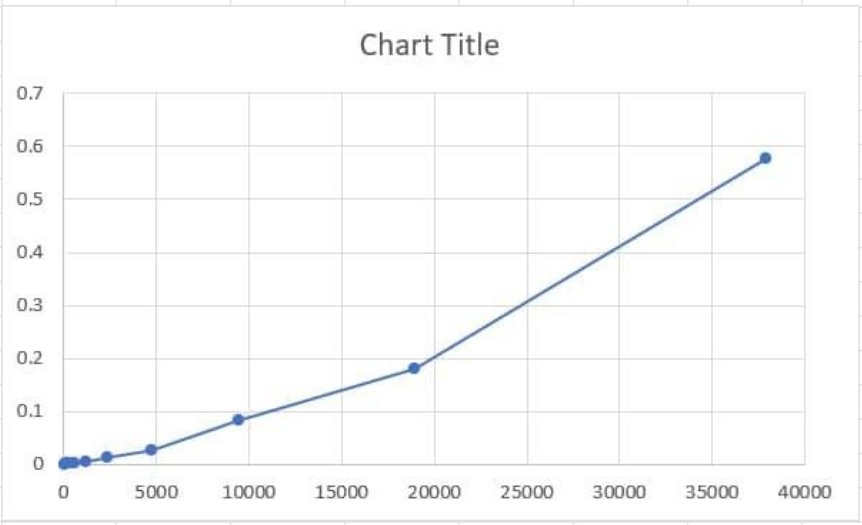
Лабораторна №2

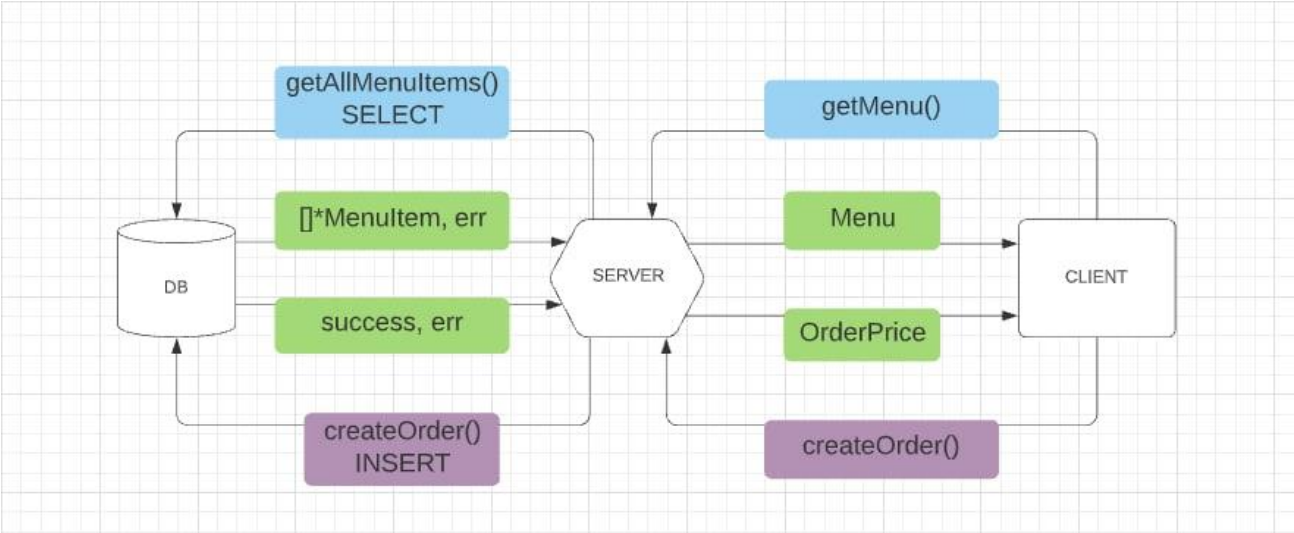
```go
func BenchmarkTransform(b *testing.B) {
  stringToTransform := "1 2 3 4 * 5 6 - / - 7 + 8 9 * 10 - / +"

  for k := 0; k < 10; k++ {
    stringToTransform = stringToTransform +
stringToTransform[1:len(stringToTransform)]

    b.Run(fmt.Sprintf("stringLen=%d", len(stringToTransform)), func(b *testing.B) {
      PostfixToPrefix(stringToTransform)
    })
  }
}
```

```
BenchmarkTransform/stringLen=75          1000000000          0.0005300 ns/op
BenchmarkTransform/stringLen=149         1000000000          0.001786 ns/op
BenchmarkTransform/stringLen=297         1000000000          0.001153 ns/op
BenchmarkTransform/stringLen=593         1000000000          0.003462 ns/op
BenchmarkTransform/stringLen=1185        1000000000          0.005997 ns/op
BenchmarkTransform/stringLen=2369        1000000000          0.01423 ns/op
BenchmarkTransform/stringLen=4737        1000000000          0.02710 ns/op
BenchmarkTransform/stringLen=9473        1000000000          0.08196 ns/op
BenchmarkTransform/stringLen=18945       1000000000          0.1792 ns/op
BenchmarkTransform/stringLen=37889       1000000000          0.5775 ns/op
PASS
```

	A	B	C	D	E	F	G	H	I	J	K
1	75	0.00053									
2	149	0.001786									
3	297	0.001153									
4	593	0.003462									
5	1185	0.005997									
6	2369	0.01423									
7	4737	0.0271									
8	9473	0.08196									
9	18945	0.1792									
10	37889	0.5775									
11											
12											
13											
14											
15											
16											
17											
18											



Лабораторна №3



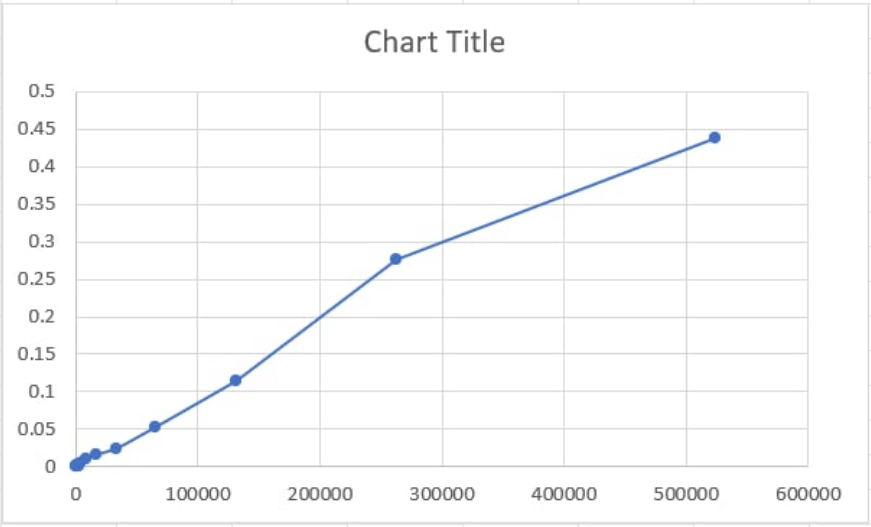
Програма складається з трьох основних частин: база даних, сервер та клієнт. Робота починається з клієнта, що посилає запит на серевер (getMenu() - отримати меню чи createOrder() – створити замовлення). Сервер реагує на запит клієнта і посилає запит у базу даних (getAllMenuItems() (SELECT 0) - для отримання даних або createOrder() (INSERT) – для створення замовлення). База даних відповідає серверу із даними ([]*MenuItem) або результатом про успішне виконання (success) або err – помилка. Після отримання даних з бази, сервер відповідає клієнту – Menu (меню), OrderPrice (розрахована ціна створеного замовлення).

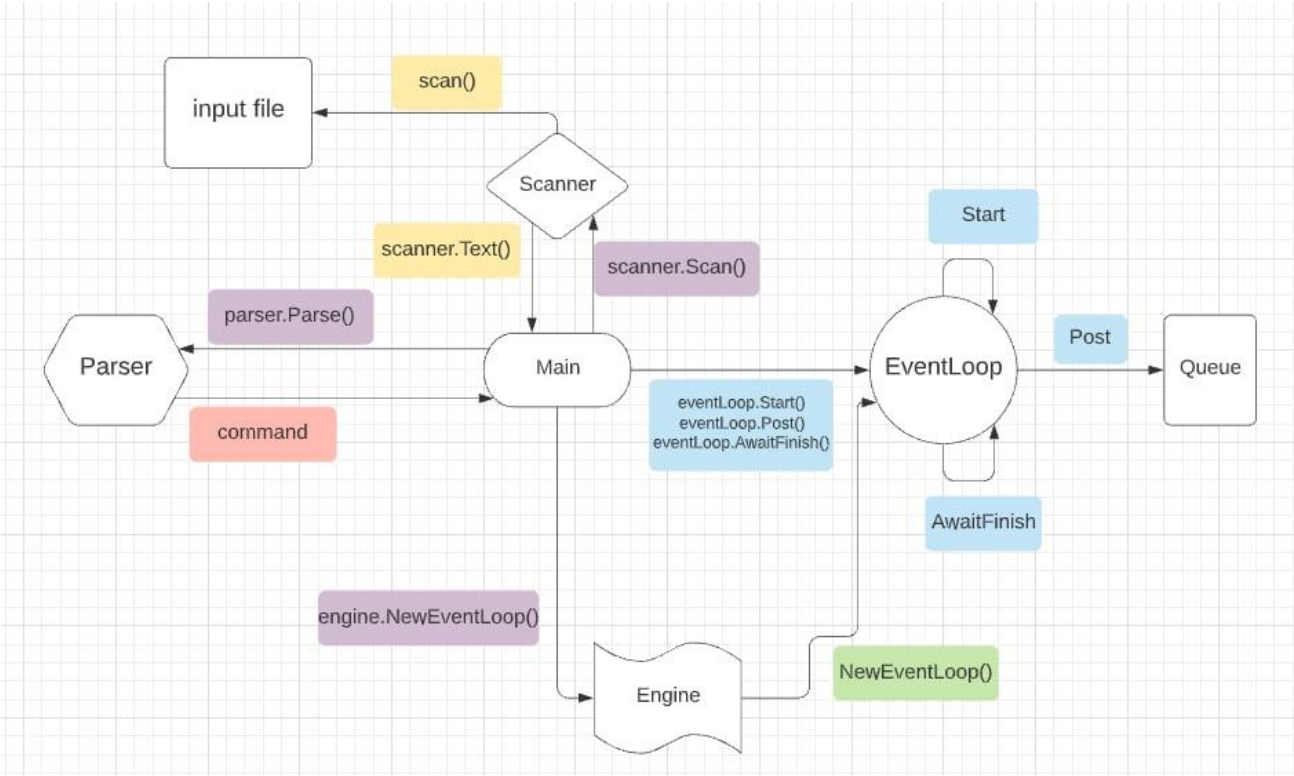
Лабораторна №4

```go
func BenchmarkParser(b *testing.B) {

  file, err := os.Create("./test.txt")
    if err != nil {
      log.Fatal(err)
    }

  linesToWrite := []string{"print Hello!", "delete hello 1"}

  for k := 0; k < 18; k++ {
    linesToWrite = append(linesToWrite,linesToWrite...)

    for _, line := range linesToWrite {
      file.WriteString(line + "\n")
    }

    b.Run(fmt.Sprintf("Len=%d", len(linesToWrite)), func(b *testing.B) {
      b.StopTimer()
        if input, err := os.Open("./test.txt"); err == nil {
        defer input.Close()
        scanner := bufio.NewScanner(input)
        for scanner.Scan() {
            commandLine := scanner.Text()
            b.StartTimer()
            parser.Parse(commandLine)
          }
        }
    })

  }
}
```

```
BenchmarkParser/Len=4           1000000000
BenchmarkParser/Len=8           1000000000
BenchmarkParser/Len=16          1000000000
BenchmarkParser/Len=32          1000000000
BenchmarkParser/Len=64          1000000000               0.0005465 ns/op
BenchmarkParser/Len=128         1000000000               0.001001 ns/op
BenchmarkParser/Len=256         1000000000               0.0009982 ns/op
BenchmarkParser/Len=512         1000000000               0.0009996 ns/op
BenchmarkParser/Len=1024        1000000000               0.001702 ns/op
BenchmarkParser/Len=2048        1000000000               0.001608 ns/op
BenchmarkParser/Len=4096        1000000000               0.004519 ns/op
BenchmarkParser/Len=8192        1000000000               0.009623 ns/op
BenchmarkParser/Len=16384       1000000000               0.01604 ns/op
BenchmarkParser/Len=32768       1000000000               0.02316 ns/op
BenchmarkParser/Len=65536       1000000000               0.05238 ns/op
BenchmarkParser/Len=131072      1000000000               0.1139 ns/op
BenchmarkParser/Len=262144      1000000000               0.2754 ns/op
BenchmarkParser/Len=524288      1000000000               0.4389 ns/op
PASS
```

256	0.0009982
512	0.0009996
1024	0.001702
2048	0.001608
4096	0.004519
8192	0.009623
16384	0.01604
32768	0.02316
65536	0.05238
131072	0.1139
262144	0.2754
524288	0.4389





Програма складається з 7 основних частин: файл із вхідними даними, сканер, парсер, головний файл виконання(main), engine, цикл подій та черга виконання. В головному файлі за допомогою виклику методу engine – newEventLoop() створюється цикл подій, що має 3 власних методи: start – запуск, awaitFinish – завершення, post – додавання команди до черги подій. Також в головному файлі створюється сканер і викликається його метод scan(), що зчитує інформацію з вхідного файлу і повертає його зміст. Після сканування в головному файлі викликається метод парсеру parse(), що інтерпретує команди і повертає перелік, що передається методу post() циклу подій. Результат виконання команд виводиться в командному рядку.