

Ανάπτυξη Λογισμικού για Δίκτυα και Τηλεπικοινωνίες

(Δικτυακός Προγραμματισμός)

Ανδρουλακάκης Ιωάννης 1115201400012

Ματάνις Παναγιώτης 1115201400297

Σακελλαρίδης Ιωάννης 1115201400307

Τραχανάς Γεώργιος 1115201300178

BackHaulServer:

Σε σχέση με το προηγούμενο παραδοτέο(1η φάση), έχει προστεθεί και το αρχείο BackhaulServerDB.java, το οποίο περιλαμβάνει όλες τις υλοποιήσεις για τον Backhaul server, όπως απαιτούνται στη 2η φάση. Πιο συγκεκριμένα:

Έχει υλοποιηθεί η συνάρτηση **void sendToDatabase(String message)**. Λαμβάνει ως όρισμα ένα μήνυμα της μορφής, το σπάει στα επιμέρους κομμάτια και στη συνέχεια δημιουργεί το ερώτημα εισαγωγής INSERT. Τέλος, εκτελείται το ερώτημα.

```
static void sendToDatabase(String message){
    try {
        Connection con = getConnection();
        String[] token = message.split( regex: "§");
        int i ;
        System.out.println("Inserting data in the Database!");
        String query = "INSERT INTO Data(AndroidIdentifier,Timestamp,GpsSignal,CriticalityLevel)" +
            "VALUES (" +token[0]+"," +token[1]+"," +token[2]+"," +token[3]+")";
        PreparedStatement ps = con.prepareStatement(query);
        ps.executeUpdate();
    }catch(Exception ex){
        System.out.println("Something went wrong!");
        ex.printStackTrace();
    }
}
```

Στη main του συγκεκριμένου αρχείου, έχει υλοποιηθεί η επικοινωνία μέσω socket, με σκοπό την ανταλλαγή δεδομένων ανάμεσα στους 2 server. Σε αυτό το σημείο, γίνεται και η αποστολή του μηνύματος με τις αναγκαίες πληροφορίες για την αποστολή στη βάση.

```
public static void main(String [] args)
{
    try {
        System.out.println("\n \n");
        int port = 6060;

        //we can change the port when we face a problem with the connection
        ServerSocket serverSocket = new ServerSocket(port);
        System.out.println("Server Started and listening to the port " + port);
        //Server is running always.
        while(true)
        {
            //Reading the message from the client
            socket = serverSocket.accept();
            InputStream is = socket.getInputStream();
            InputStreamReader isr = new InputStreamReader(is);
            BufferedReader br = new BufferedReader(isr);
            String message = br.readLine();

            // System.out.println(message);
            //send the message to the database in Backhaul server to format a new query
            sendToDatabase(message);
        }
    }
}
```

BackhaulServer directory:

Σε σχέση με το προηγούμενο παραδοτέο(1η φάση), έχει προστεθεί και το αρχείο BackhaulServerDB.java, το οποίο περιλαμβάνει όλες τις υλοποιήσεις για τον Backhaul server, όπως απαιτούνται στη 2η φάση. Πιο συγκεκριμένα: Έχει υλοποιηθεί η συνάρτηση **void sendToDatabase(String message)**. Λαμβάνει ως όρισμα ένα μήνυμα της μορφής, το σπάει στα επιμέρους κομμάτια και στη συνέχεια δημιουργεί το ερώτημα εισαγωγής INSERT. Τέλος, εκτελείται το ερώτημα.

Στη main του συγκεκριμένου αρχείου, έχει υλοποιηθεί η επικοινωνία μέσω socket, με σκοπό την ανταλλαγή δεδομένων ανάμεσα στους 2 server. Σε αυτό το σημείο, γίνεται και η αποστολή του μηνύματος με τις αναγκαίες πληροφορίες για την αποστολή στη βάση.

Έχει δημιουργηθεί το αρχείο FeatureVector.java, που υλοποιεί μία αναπαράσταση για τα Feature vector, όπως ζητούνται. Περιλαμβάνει τη συνάρτηση **List<FeatureVector> readFile(File file)**, η οποία διαβάζει το αρχείο που προέκυψε από το test set και αποθηκεύει τα δεδομένα για κάθε γραμμή του σε ένα αντικείμενο **FeatureVector**. Στη συνέχεια τοποθετούνται μέσα σε μία ArrayList.

Έχει δημιουργηθεί το αρχείο KnnFileClassifier.java, όπου έχει δημιουργηθεί μέσα η ομώνυμη κλάση. Έχει υλοποιηθεί η συνάρτηση **void classify(String pathName)**. Η συνάρτηση αυτή υλοποιεί επί της ουσίας τον αλγόριθμο knn, όπως μας ζητείται στην εργασία. Πιο συγκεκριμένα, διαβάζει κάθε feature vector από το αρχείο της 1ης φάσης και τα αποθηκεύει σε μία λίστα. Στη συνέχεια, διαβάζει ένα αρχείο από το test set(όπως έχει σταλεί από τα κινητά) και εφαρμόζει τον knn.

Αποτελεσματικότητα υλοποίησης αλγορίθμου

Για τις διάφορες τιμές του k (3,5,7,11), η ευστοχία του αλγορίθμου κυμαίνεται ανάμεσα στις τιμές 70% και 90%.