

Δίκτυα Υπολογιστών II

Ακαδημαϊκό Έτος 2024-2025

Υποχρεωτική Εργασία

(Επιπλέον Πληροφορίες)

Ανάπτυξη Εφαρμογής Chat and VoIP

1. Σύντομη Περιγραφή

Στα πλαίσια του μαθήματος οι φοιτητές θα κληθούν να πραγματοποιήσουν εργασία η οποία θα αφορά την ανάπτυξη μιας End-to-end Chat and VoIP εφαρμογής. Πιο συγκεκριμένα, η εφαρμογή, μέσω ενός απλού γραφικού περιβάλλοντος (GUI), θα επιτρέπει σε δύο διαφορετικούς χρήστες να ανταλλάσσουν μηνύματα κειμένου αλλά και να επικοινωνούν φωνητικά μέσω του διαδικτύου. Η μόνη γνώση που θα απαιτείται θα είναι οι IPs των συσκευών των χρηστών μέσω των οποίων θα επιχειρούν να συνομιλήσουν, αλλά και οι λογικές πόρτες (ports) που θα χρησιμοποιούν τοπικά η εφαρμογές τους. Σκοπός είναι να επιτευχθεί επικοινωνία παρόμοια με αυτή των σύγχρονων εφαρμογών όπως Skype, Viber κλπ., με την διαφορά πως δεν θα υπάρχει κάποιος κεντρικός διακομιστής (server) που θα είναι υπεύθυνος για τον συντονισμό της επικοινωνίας και μέσω του οποίου θα διέρχονται τα μηνύματα πριν φθάσουν στον τελικό τους προορισμό. Αντιθέτως η επικοινωνία θα γίνεται απευθείας μεταξύ των τερματικών (end-to-end επικοινωνία) χωρίς κεντρικό έλεγχο (δηλ. αποκεντροποιημένα) και θα βασίζεται αποκλειστικά στις δυνατότητες της ίδιας της υποδομής του Internet και της τρέχουσας κατάστασής του.

Η εφαρμογή θα υλοποιηθεί σε γλώσσα προγραμματισμού Java όπου θα χρησιμοποιηθούν τα βασικά πακέτα (βιβλιοθήκες) που παρέχει η γλώσσα τόσο για την υλοποίηση της δικτύωσης (networking) όσο και για την υλοποίηση των γραφικών διεπαφών (GUI). Η χρήση των βασικών πακέτων/βιβλιοθηκών αποσκοπεί στην απλούστευση της υλοποίησης, μέσω περιορισμού χρήσης εξωτερικών εξαρτήσεων (external dependencies), βοηθώντας έτσι την μεταφερισιμότητα της τελικής εφαρμογής μεταξύ διαφορετικών συσκευών και κατ' επέκταση τον τελικό έλεγχο και την βαθμολόγηση της εργασίας.

2. Γενικές Πληροφορίες

- **Τύπος Εργασίας:** Υποχρεωτική
- **Αριθμός Ατόμων:** 2
- **Βαθμολογία:** 3 μονάδες (30% του συνολικού βαθμού)
- **Ημερομηνία Υποβολής:** 16/12/2024

3. Αναλυτική Περιγραφή Απαιτήσεων και Ερωτημάτων

Η εργασία αποτελείται από **δύο τμήματα** τα οποία αφορούν την υλοποίηση των δύο βασικών λειτουργιών της εφαρμογής, οι οποίες είναι η **αποστολή μηνυμάτων (instant messaging - chat)** και η **φωνητική επικοινωνία σε πραγματικό χρόνο (real-time audio/voice communication)**. Τα τμήματα αυτά θα βαθμολογηθούν ισόποσα. Πιο αναλυτικά οι λειτουργίες που θα πρέπει (κατ' ελάχιστον) να υλοποιηθούν είναι οι εξής:

1. **Αποστολή Μηνυμάτων (Instant Messaging / Chat):** Οι χρήστες θα πρέπει να μπορούν μέσω της εφαρμογής να ανταλλάσσουν μηνύματα κειμένου μεταξύ τους. **(1,5 μονάδες)**
2. **Φωνητική Επικοινωνία (VoIP):** Οι χρήστες θα πρέπει μέσω της εφαρμογής να επιτύχουν αμφίδρομη φωνητική επικοινωνία σε πραγματικό χρόνο, ανταλλάσσοντας πακέτα φωνής μεταξύ τους. **(1,5 μονάδες)**

Παρατηρήσεις:

- Για την υλοποίηση και των δύο λειτουργιών θα γίνει χρήση του πρωτοκόλλου UDP για την επίτευξη της πλήρους peer-to-peer επικοινωνίας.
- Και στις δύο περιπτώσεις θα πρέπει η επικοινωνία να είναι **ασύγχρονη, αμφίδρομη** και **μη διακοπτόμενη**.

Επιπλέον Ενασχόληση: (Προαιρετικά - Θα εκτιμηθεί)

- Όσοι ενδιαφέρονται μπορούν μεταγενέστερα να υλοποιήσουν το πρώτο τμήμα (p2p chat) με χρήση του πρωτοκόλλου TCP.
- Στο πρώτο τμήμα θα είχε ενδιαφέρον η χρήση ενός αλγορίθμου κρυπτογράφησης για την επίτευξη πλήρους ανώνυμης και ιδιωτικής επικοινωνίας μεταξύ των δύο μερών.
- Στο δεύτερο τμήμα θα είχε ενδιαφέρον η μελέτη τεχνικών για την μείωση του latency.

4. Παραδοτέα Εργασίας

Τα παραδοτέα της εργασίας θα είναι (i) **ο κώδικας της εφαρμογής σε μορφή zip** και (ii) μια γραπτή **αναφορά**. Η αναφορά θα περιέχει μια σύντομη περιγραφή του κώδικα και **εικόνες** από το **Wireshark** οι οποίες θα δείχνουν: (i) παραδείγματα μηνυμάτων κειμένου που αποστάλθηκαν μέσω της εφαρμογής μεταξύ των δύο χρηστών (θα πρέπει να φαίνεται κατ' ελάχιστον το Network/Internet **header** και το **payload** τόσο σε εξαδική όσο και σε μορφή κειμένου)¹, (ii) το stream των πακέτων φωνής που ανταλλάσσονται (η λίστα όπως εμφανίζεται στο main window του Wireshark) και (iii) παραδείγματα πακέτων φωνής που ανταλλάχθηκαν μέσω της εφαρμογής (θα πρέπει να φαίνεται κατ' ελάχιστον το Network/Internet **header** και το **payload** τουλάχιστον σε δυαδική μορφή).

Τα παραδοτέα θα πρέπει να υποβληθούν πριν την ημερομηνία υποβολής μέσω του **e-learning**.

¹ Δοκιμάστε διαφορετικά μεγέθη πακέτων. Τι θα συμβεί αν το πακέτο είναι αρκετά μεγάλο (π.χ. μεγαλύτερο των 1480 bytes;)

5. Χρήσιμα για την εργασία

5.1 Αρχικός Κώδικας Εργασίας

Για ευκολία δεν θα κληθούν να γράψουν τον κώδικα της εφαρμογής από το μηδέν, αλλά αντιθέτως τους παρέχεται ένας **αρχικός βασικός κώδικας** που θα πρέπει να **συμπληρώσουν** ώστε να υλοποιήσουν τις **2 βασικές λειτουργίες της εφαρμογής**, δηλαδή του **chat** και του **VoIP**.

Ο κώδικας αυτός βρίσκεται στο GitHub στον παρακάτω σύνδεσμο:

https://github.com/siavvasm/CN2_AUTH_ChatAndVoIP.git

5.1.1 Βήματα Εγκατάστασης, Μεταγλώττισης και Εκτέλεσης του κώδικα

Αρχικά, θα πρέπει να γίνει λήψη του κώδικα τοπικά. Αυτό μπορεί να επιτευχθεί είτε μέσω λήψης του κώδικα (επιλογής Download που παρέχεται από το GitHub), είτε μέσω κλωνοποίησης (εντολής clone) του κώδικα μέσω του τερματικού (terminal). Για την εξοικείωση των φοιτητών με την τεχνολογία του Git (η οποία είναι πολύ χρήσιμη στην πράξη) συνίσταται ο δεύτερος τρόπος.

Πιο συγκεκριμένα, για να κλωνοποιήσουμε τον κώδικα τοπικά στον υπολογιστή μας ανοίγουμε το τερματικό και μεταβαίνουμε στον φάκελο που επιθυμούμε να κατεβάσουμε τον κώδικα και εκτελούμε την ακόλουθη εντολή:

```
git clone https://github.com/siavvasm/CN2_AUTH_ChatAndVoIP.git
```

Μόλις ολοκληρωθεί η διαδικασία, θα υπάρχει ο φάκελος με όνομα "CN2_AUTH_ChatAndVoIP" εντός του φακέλου που επιλέξαμε.

Στην συνέχεια, για να επεξεργαστούμε τον κώδικα θα πρέπει να χρησιμοποιήσουμε είτε έναν απλό text editor (π.χ. Notepad, Notepad++ ή Vim) είτε ένα Integrated Development Environment (IDE). Συνίσταται η χρήση IDE και συγκεκριμένα του *Eclipse IDE for Enterprise Java and Web Developers* η τελευταία έκδοση του οποίου μπορεί να βρεθεί στον παρακάτω σύνδεσμο:

<https://www.eclipse.org/downloads/packages/>

Αφού εγκατασταθεί το Eclipse IDE και το εκκινήσουμε θα πρέπει να εισάγουμε τον κώδικά μας στο IDE. Για να γίνει αυτό ακολουθούμε τα παρακάτω βήματα:

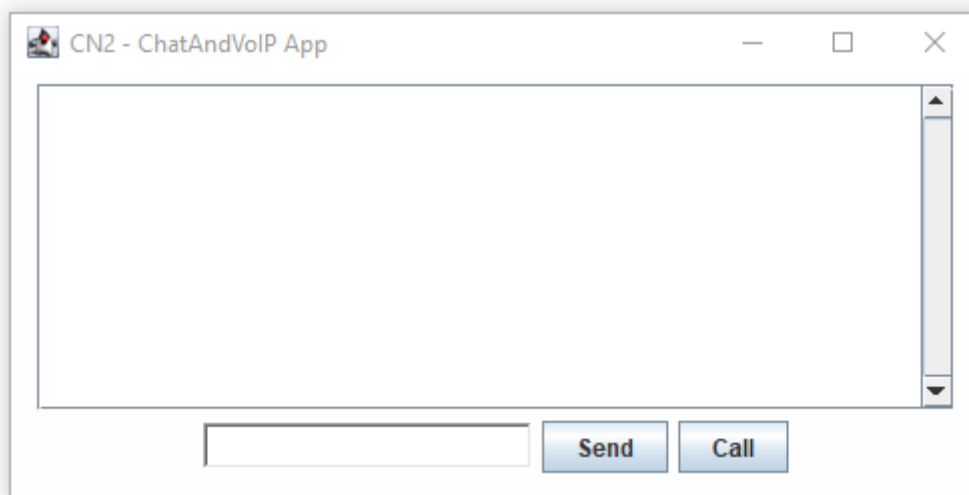
- Στο Eclipse IDE, επιλέγουμε **File -> Import**
- Στο παράθυρο που εμφανίζεται, επιλέγουμε **"Existing Maven Projects"** και κάνουμε κλικ στο **Next**
- Στη συνέχεια, κάνουμε κλικ στο κουμπί **Browse** και πλοηγούμαστε μέσα στον φάκελο **CN2_AUTH_ChatAndVoIP**, έπειτα κάνουμε κλικ στο κουμπί **"Select Folder"**
- Επιλέγουμε το checkbox με το αρχείο **"pom.xml"** και κάνουμε κλικ στο κουμπί **Finish**

- Η εφαρμογή **CN2_AUTH_ChatAndVoIP** θα πρέπει να είναι ορατή στο **Navigation Pane** του **Eclipse** **IDE**.

Μόλις ολοκληρωθεί η διαδικασία θα δείτε τον φάκελο με τον κώδικα της εφαρμογής στην στήλη στα αριστερά. Αν ανοίξετε τον φάκελο θα δείτε πως υπάρχει μια μόνο κλάση με όνομα **App.java**. Αυτή είναι η βασική κλάση της εφαρμογής στην οποία βρίσκεται και η `main()` μέθοδος.

- **ΠΡΟΣΟΧΗ:** Θα πρέπει να σιγουρευτούμε πως η σωστή έκδοση της Java (μεγαλύτερη της 1.8) χρησιμοποιείται από το Eclipse IDE. Για να το κάνουμε αυτό κάνουμε δεξί κλικ στον φάκελο του **CN2_AUTH_ChatAndVoIP** όπως εμφανίζεται στην αριστερή στήλη του Eclipse IDE και έπειτα στην επιλογή **Properties**. Στο παράθυρο που εμφανίζεται επιλέγουμε **Java Compilers** και σιγουρευόμαστε ότι είναι επιλεγμένη έκδοση από 1.8 και έπειτα.

Για να μεταγλωττίσετε και να εκτελέσετε τον κώδικα θα πρέπει να μεταβείτε στον κώδικα της **App.java** εντός του Eclipse IDE και να επιλέξετε την επιλογή **"Run As"** και έπειτα την επιλογή **"Run as Java Application"**. Αν η μεταγλώττιση πραγματοποιηθεί επιτυχώς στην οθόνη σας θα προβληθεί η γραφική διεπαφή της εφαρμογής, η οποία φαίνεται στην Εικόνα 1.



Εικόνα 1 - Η γραφική διεπαφή της εφαρμογής Chat and VoIP

Όπως φαίνεται στην Εικόνα 1, η εφαρμογή περιλαμβάνει ένα πεδίο κειμένου (**Text Field**) που χρησιμοποιείται για τη σύνταξη μηνυμάτων που πρέπει να σταλούν στον άλλο συνομιλητή, μια περιοχή κειμένου (**Text Area**) για την εμφάνιση των μηνυμάτων που έχουν σταλεί ή ληφθεί, και δύο κουμπιά. Το κουμπί **"Send"** είναι υπεύθυνο για την αποστολή των μηνυμάτων που πληκτρολογούνται στο πεδίο κειμένου. Το κουμπί **"Call"** είναι υπεύθυνο για την έναρξη μιας κλήσης με τον απομακρυσμένο συνομιλητή (δηλαδή, για την έναρξη της ακρόασης πακέτων φωνής και την αποστολή πακέτων φωνής που καταγράφονται από το μικρόφωνο του υπολογιστή).

Στα πλαίσια της εργασίας, οι φοιτητές θα πρέπει να υλοποιήσουν τις λειτουργικότητες των κουμπιών **"Send"** και **"Call"** (καθώς προς το παρόν δεν εκτελούν καμία λειτουργία). Αυτές οι λειτουργικότητες αντιστοιχούν στα 2 τμήματα της παρούσας εργασίας που περιγράφονται στην Υποενότητα 3.

Μέσα στον κώδικα της εφαρμογής, υπάρχουν συγκεκριμένα σημεία όπου θα χρειαστεί να συμπληρωθούν με τον κατάλληλο κώδικα τα οποία επισημαίνονται με τα εξής σχόλια:

```
// TODO: Your code goes here...
```

```
// TODO: Please define and initialize your variables here...
```

Επομένως, οι φοιτητές ΔΕΝ θα χρειαστεί να υλοποιήσουν κάποιο κομμάτι της γραφικής διεπαφής. Θα πρέπει να υλοποιήσετε ΜΟΝΟ την λειτουργικότητα της εφαρμογής και συγκεκριμένα την λειτουργία της **main()** αλλά και την λειτουργία των κουμπιών “Send” και “Call” (η οποία πραγματοποιείται από την μέθοδο **actionPerformed()**).

Προαιρετικά (Για όσους ενδιαφέρονται):

- Για απλότητα η γραφική διεπαφή υλοποιήθηκε χρησιμοποιώντας την `java.swing` η οποία είναι native βιβλιοθήκη της Java (αλλά όχι τόσο elegant). Εναλλακτικά θα μπορούσε να χρησιμοποιηθεί κάποιο πιο advanced framework όπως το JavaFX. Όσοι το επιθυμείτε μπορείτε να το δοκιμάσετε. Αν θέλετε υλικό ή σχετικές πληροφορίες στείλτε μου email. Απλά τονίζεται πως η υλοποίηση της γραφικής διεπαφής ΔΕΝ αποτελεί μέρος της εργασίας.

5.2 Οδηγίες και συμβουλές για την υλοποίηση

5.2.1 Διευκρίνιση για χρήση βιβλιοθηκών (libraries) και πλαισίων (frameworks)

Η εφαρμογή θα υλοποιηθεί αποκλειστικά σε γλώσσα προγραμματισμού **Java**. Η υλοποίηση θα βασιστεί μόνο στις **native** βιβλιοθήκες της Java. Δεν επιτρέπεται η χρήση εξωτερικών εξαρτήσεων (dependencies). Αυτό θα γίνει αφενός για λόγους απλότητας και μεταφερσιμότητας του κώδικα, αλλά και αφετέρου για αποφυγή χρήσης βιβλιοθηκών που υλοποιούν έτοιμα τις ζητούμενες λειτουργίες. Πιο συγκεκριμένα θα πρέπει να βασιστείτε στις παρακάτω βιβλιοθήκες και πλαίσια (frameworks):

- `java.net`
- `java.io`
- `javax.sound`

5.2.2 Χρήση της `java.net`

Η επικοινωνία θα βασιστεί στο πρωτόκολλο UDP. Επομένως, οι δύο κλάσεις που θα χρειαστείτε είναι οι `DatagramSocket` για την δημιουργία sockets και `DatagramPacket` για την δημιουργία datagram πακέτων που θα αποστέλλονται μέσω των sockets. Για την αποστολή πακέτων προτείνεται να χρησιμοποιήσετε buffers (πίνακες) των 1024 bytes.

5.2.3 Χρήση της `javax.sound`

Σε αυτή την εργασία, θα χρησιμοποιήσετε τη βιβλιοθήκη `javax.sound` για να **καταγράψετε** και να **αναπαράγετε δεδομένα ήχου**. Οι κλάσεις που θα χρειαστείτε κατά βάση θα είναι η `TargetDataLine` και η `SourceDataLine` για την επικοινωνία με το μικρόφωνο και το ηχείο του υπολογιστή. Επίσης, θα χρειαστείτε την `AudioFormat` για να ορίσετε την κατάλληλη μορφή των δεδομένων ήχου. Μπορείτε να χρησιμοποιήσετε ότι audio format επιθυμείτε, αλλά για λόγους ευκολίας της εκπόνησης της εργασίας προτείνεται να υιοθετήσετε την **μονοφωνική παλμοκωδική διαμόρφωση ήχου PCM (pulse code**

modulation) με τα εξής χαρακτηριστικά: (i) συχνότητα δειγματοληψίας 8000 samples/sec, (ii) μέγεθος δείγματος 8 bits (1 byte), (iii) μονοφωνικό κανάλι (1 κανάλι) και (iv) signed δείγματα.

5.2.4 Επίτευξη ασύγχρονης λειτουργίας και επικοινωνίας

Για να επιτευχθεί ασύγχρονη λειτουργία θα πρέπει να χρησιμοποιηθεί πολυνημάτωση. Πιο συγκεκριμένα θα πρέπει να χρησιμοποιήσετε την κλάση Thread της Java και ιδίως το interface Runnable που επιτρέπει να ορίσετε συγκεκριμένες λειτουργίες να εκτελούνται παράλληλα.

5.2.5 Χρήση του GUI

Όπως αναφέραμε, στα πλαίσια της εργασίας δεν θα χρειαστεί να δημιουργήσετε ή να τροποποιήσετε την γραφική διεπαφή της εφαρμογής. Το μόνο που θα χρειαστεί είναι να διαβάσετε τα μηνύματα που πρέπει να αποσταλούν μέσω της εφαρμογής σας από το παρεχόμενο πεδίο κειμένου (Text Field) και να εμφανίζετε τα μηνύματα που λαμβάνει η εφαρμογή σας στην περιοχή κειμένου (Text Area) που παρέχεται από την γραφική διεπαφή. Για να γίνει αυτό θα πρέπει να χρησιμοποιήσετε τις παρακάτω εντολές:

- **inputTextField.getText():** Διαβάζει το κείμενο που είναι γραμμένο εκείνη την στιγμή στο πεδίο κειμένου του GUI, το οποίο αναπαρίσταται ως αντικείμενο τύπου JTextField με όνομα inputTextField. Η μέθοδος επιστρέφει τα δεδομένα σε μορφή String.
- **textArea.append("...your text goes here..."):** Η μέθοδος αυτή προσθέτει το κείμενο που λαμβάνει ως όρισμα στην περιοχή κειμένου του GUI, η οποία αναπαρίσταται ως αντικείμενο της κλάσης JTextArea με όνομα textArea. **Συμβουλή:** Καθώς στην περιοχή κειμένου αυτή θα "τυπώνονται" τόσο τα μηνύματα που στέλνετε εσείς όσο και τα μηνύματα που λαμβάνετε, καλό θα είναι να βάζετε μια ετικέτα στην αρχή για να ξεχωρίζετε σε ποιόν ανήκει το εκάστοτε μήνυμα (π.χ. "local", "remote", κλπ.).

6. Επιπλέον παρατηρήσεις και σχόλια

- Προσπαθήστε να συνδεθείτε από διαφορετικούς υπολογιστές και να επικοινωνήσετε μεταξύ σας. Η επικοινωνία σε επίπεδο τοπικού δικτύου (LAN) αρκεί.
- **Firewall:** Το τείχος προστασίας σας ενδέχεται να εμποδίζει την επικοινωνία. Βεβαιωθείτε ότι το τείχος προστασίας σας επιτρέπει την διέλευση UDP πακέτων από τις θύρες που χρησιμοποιείτε. Ίσως χρειαστεί να ορίσετε συγκεκριμένους κανόνες στο λειτουργικό σας σύστημα για να επιτρέψετε την διέλευση πακέτων από συγκεκριμένες θύρες μέσω του τείχους προστασίας.
- Αν επιθυμείτε να επικοινωνήσετε μέσω διαδικτύου (με χρήση των δημοσίων διευθύνσεων IP σας) τότε θα πρέπει να βρείτε τρόπο να αντιμετωπίσετε το NAT (πιο απλός τρόπος είναι το PORT Forwarding μέσω των ρυθμίσεων του router σας, ενώ υπάρχουν και άλλοι ενδιαφέροντες τρόποι όπως το UDP Punching όπως αναφέραμε στο μάθημα).