

# StudentNote – Τελική Αναφορά Έργου

---

## Ομάδα Φοιτητών

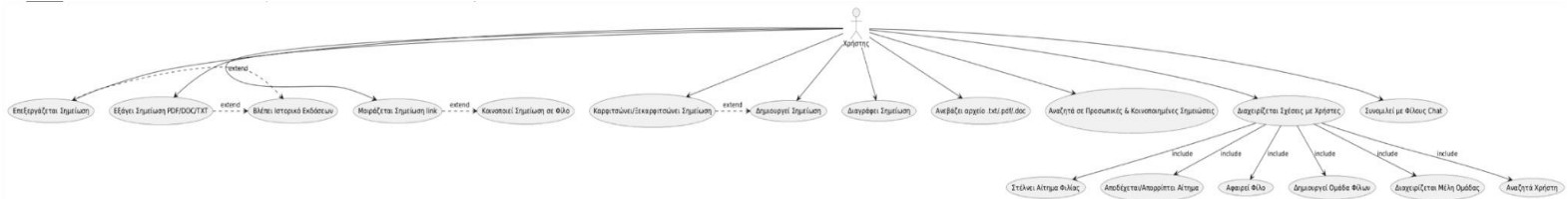
Ονοματεπώνυμο	Αριθμός Μητρώου (AM)
ΑΝΔΡΕΑΤΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ	AM:1084620
ΓΙΩΝΑΣ ΑΓΓΕΛΟΣ-ΑΣΤΕΡΙΟΣ	AM:1093344
ΜΠΟΧΩΤΗΣ ΝΙΚΟΛΑΟΣ	AM:1084525
ΚΟΛΟΙ ΝΙΚΟΛΑΟΣ	AM:1063429
ΓΚΟΥΜΑΣ ΓΙΩΡΓΟΣ	AM:1100522

## Ρόλοι Συμμετεχόντων

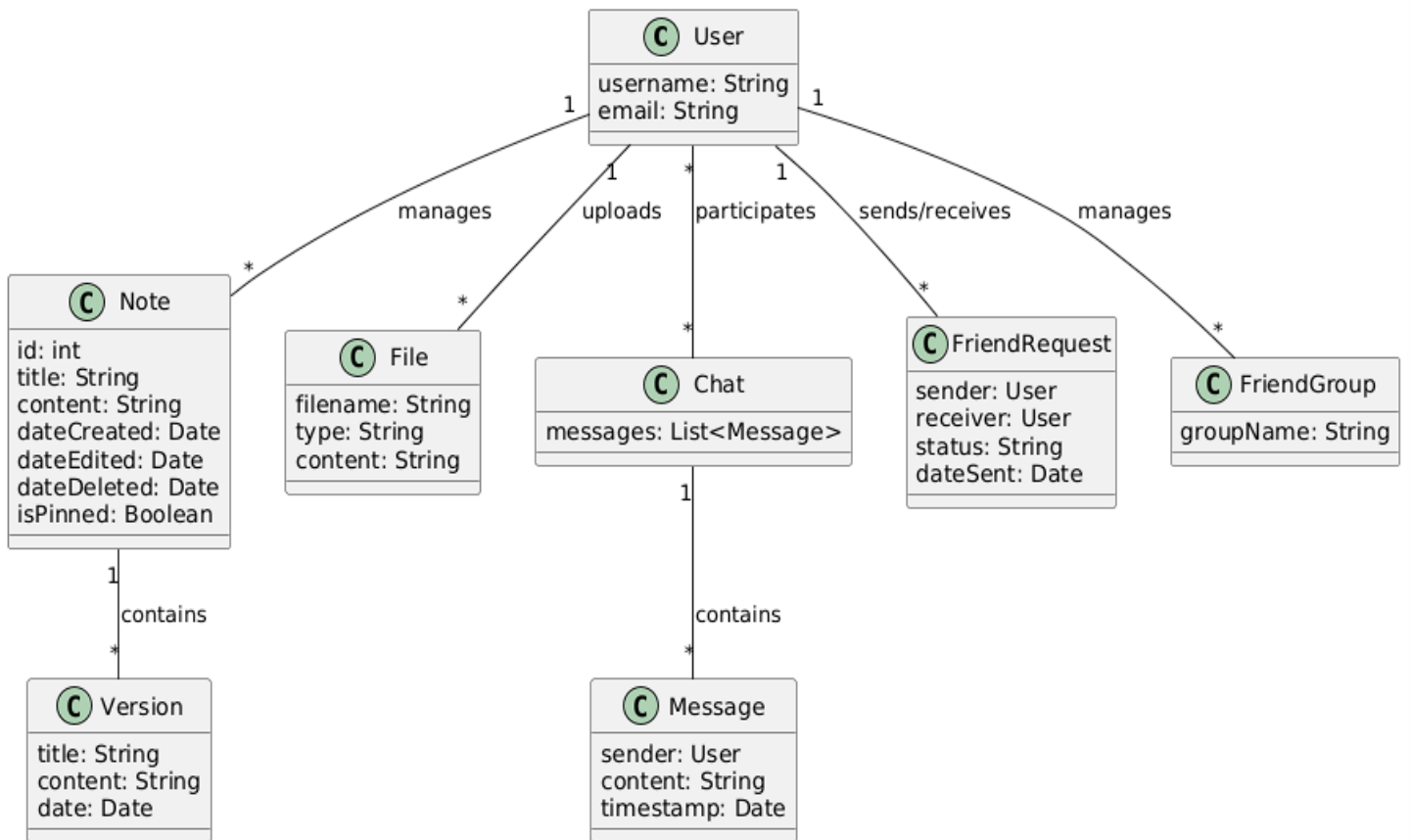
- ΑΝΔΡΕΑΤΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ: Τελικό Project Description, τελική επιμέλεια ολοκλήρου του κώδικα (HTML,CSS,JS) και υλοποίηση JavaScript για τα Use Cases 1-6 και 12.
- ΓΙΩΝΑΣ ΑΓΓΕΛΟΣ-ΑΣΤΕΡΙΟΣ: Επιμέλεια Τελικής Αναφοράς, Test Cases, συνεισφορά στην CSS του κώδικα
- ΜΠΟΧΩΤΗΣ ΝΙΚΟΛΑΟΣ: Κατασκευή UML Διαγραμμάτων (Class, Domain, Sequence), συνεισφορά στην HTML του κώδικα
- ΚΟΛΟΙ ΝΙΚΟΛΑΟΣ: Επιμέλεια του Use Case Diagram και του Domain Model
- ΓΚΟΥΜΑΣ ΓΙΩΡΓΟΣ: Επιμέλεια Τελικών Use Case Descriptions, Χρήση JavaScript για την υλοποίηση των Use Cases 7-11

## 1. Διαγράμματα

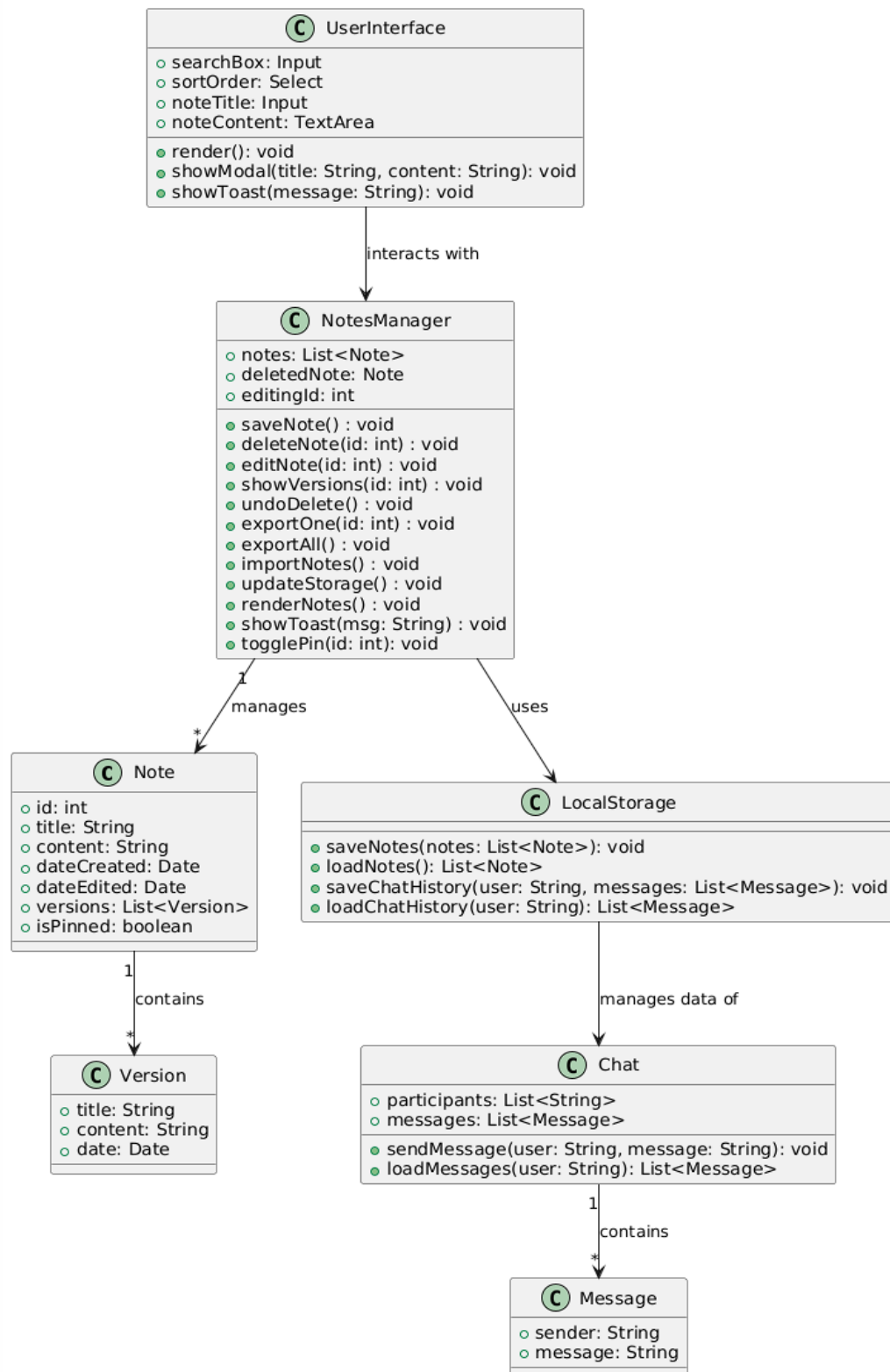
### Use Case Diagram



### Domain Model



## Class Diagram



## 2. Project Description (v1.0)

**To StudentNote** είναι ένα σύγχρονο web app online σημειώσεων, σχεδιασμένη για αποτελεσματική δημιουργία, διαχείριση, κοινοποίηση και οργάνωση προσωπικών σημειώσεων. Στόχος της είναι να διευκολύνει τους φοιτητές να μοιράζονται μεταξύ τους σημειώσεις που σχετίζονται με το ακαδημαϊκό τους πρόγραμμα, ενισχύοντας την ανταλλαγή γνώσης και τη συνεργασία.

Η εφαρμογή παρέχει **προηγμένες λειτουργίες**, όπως:

- **Δημιουργία, επεξεργασία και διαγραφή σημειώσεων** με δυνατότητα αναίρεσης διαγραφής.
- **Ιστορικό εκδόσεων** κάθε σημείωσης, με δυνατότητα επαναφοράς και εξαγωγής προηγούμενων εκδόσεων σε μορφές **PDF, DOC και TXT**.
- **Καρφίτσωμα (αγαπημένες) σημειώσεων**, ώστε να εμφανίζονται πρώτες στη λίστα.
- **Αναζήτηση και ταξινόμηση σημειώσεων** (αλφαβητικά ή κατά ημερομηνία).
- **Κοινοποίηση σημειώσεων** είτε μέσω **συνδέσμου** είτε **σε φίλο**, με ενσωμάτωση στο **ιστορικό συνομιλιών**.
- **Δυνατότητα συνομιλίας με φίλους** και αναζήτησης χρηστών μέσω email.
- **Υποστήριξη εισαγωγής αρχείων** (.txt, .doc, .pdf) ως σημειώσεις και εξαγωγής όλων των σημειώσεων ως **JSON**.
- **Dark mode** με αποθήκευση προτιμήσεων θέματος.

Η εφαρμογή είναι πλήρως λειτουργική χωρίς την ανάγκη για backend, χάρη στη χρήση του **localStorage** για την αποθήκευση όλων των δεδομένων. Συγκεκριμένα, αποθηκεύονται οι σημειώσεις και οι εκδόσεις τους, το επιλεγμένο θέμα εμφάνισης (dark ή light mode), το ιστορικό συνομιλιών μεταξύ των χρηστών, καθώς και τα κοινοποιημένα στοιχεία, επιτρέποντας έτσι συνεχή και απλή χρήση από τον ίδιο τον χρήστη.

Ολόκληρο το σύστημα υλοποιήθηκε αποκλειστικά με **HTML, CSS και JavaScript**, χωρίς εξωτερικά frameworks, προσφέροντας **απλότητα, απόδοση και ευκολία μεταφοράς** του κώδικα για την αναβάθμιση και την επέκταση του σε ένα πλήρως λειτουργικό web app.

### 3. Use Case Descriptions (v1.0)

StudentNote - Use Case Descriptions (Updated)

#### UC1 – Δημιουργία Σημείωσης

Ο χρήστης ξεκινά τη διαδικασία δημιουργίας μιας νέας σημείωσης εισάγοντας τίτλο και περιεχόμενο, και στη συνέχεια πατά "Αποθήκευση". Ο τίτλος είναι υποχρεωτικός. Αν λείπει ο τίτλος, εμφανίζεται σχετικό μήνυμα σφάλματος.

#### UC2 – Επεξεργασία Σημείωσης

Ο χρήστης επιλέγει τη σημείωση που επιθυμεί να επεξεργαστεί και πατάει το κουμπί "Επεξεργασία". Μετά την ολοκλήρωση των αλλαγών, πατά "Αποθήκευση" και το σύστημα ενημερώνει τη σημείωση. Ακύρωση επαναφέρει την προηγούμενη έκδοση.

#### UC3 – Διαγραφή Σημείωσης

Ο χρήστης επιλέγει τη σημείωση προς διαγραφή, πατά "Διαγραφή" και επιβεβαιώνει τη διαγραφή. Μετά την επιβεβαίωση, η σημείωση διαγράφεται από τη λίστα σημειώσεων.

#### UC4 – Εξαγωγή Σημείωσης

Ο χρήστης επιλέγει μια σημείωση και πατά "Εξαγωγή". Επιλέγει τη μορφή εξαγωγής (PDF, DOC, TXT), και το σύστημα δημιουργεί το αντίστοιχο αρχείο και το κατεβάζει στον υπολογιστή του χρήστη.

#### UC5 – Μοιράζεται Σημείωση (link)

Ο χρήστης επιλέγει μια σημείωση και πατά "Κοινοποίηση με σύνδεσμο". Το σύστημα δημιουργεί ένα μοναδικό σύνδεσμο για προβολή της σημείωσης από άλλους.

#### UC6 – Ιστορικό Εκδόσεων

Ο χρήστης επιλέγει μια σημείωση και πατά "Ιστορικό Εκδόσεων". Εμφανίζεται λίστα παλαιότερων εκδόσεων και ο χρήστης μπορεί να επιλέξει μία έκδοση για να την επαναφέρει ως τρέχουσα.

#### UC7 – Ανεβάζει Αρχείο

Ο χρήστης ανεβάζει αρχείο (.pdf, .doc, .txt) από τον υπολογιστή του, το οποίο μετατρέπεται αυτόματα σε νέα σημείωση. Αν το αρχείο είναι άδειο ή μη υποστηριζόμενο, εμφανίζεται μήνυμα αποτυχίας.

#### UC8 – Αναζητά Σημείωση

Ο χρήστης πληκτρολογεί λέξη-κλειδί στο πεδίο αναζήτησης. Το σύστημα εμφανίζει άμεσα σχετικές σημειώσεις.

#### UC9 – Chat με Φίλους

Ο χρήστης επιλέγει φίλο από τη λίστα και ξεκινά συνομιλία.

### **UC10 – Κοινοποιεί Σημείωση σε Φίλο**

Ο χρήστης επιλέγει μια σημείωση, διαλέγει φίλο από τη λίστα και μπορεί να κοινοποιήσει μια σημείωση. Ο φίλος ενημερώνεται μέσω μηνύματος .

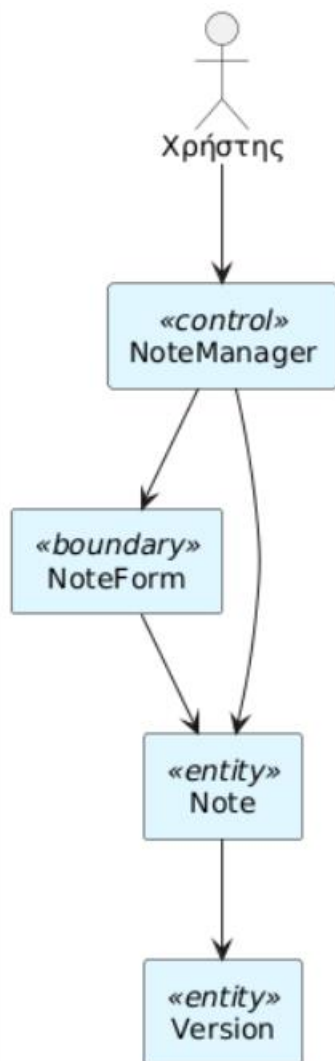
### **UC11.1 – Αναζήτηση Χρήστη**

Ο χρήστης αναζητά άλλους χρήστες μέσω ονόματος ή email. Τα αποτελέσματα εμφανίζονται και ο χρήστης μπορεί να στείλει αίτημα φιλίας.

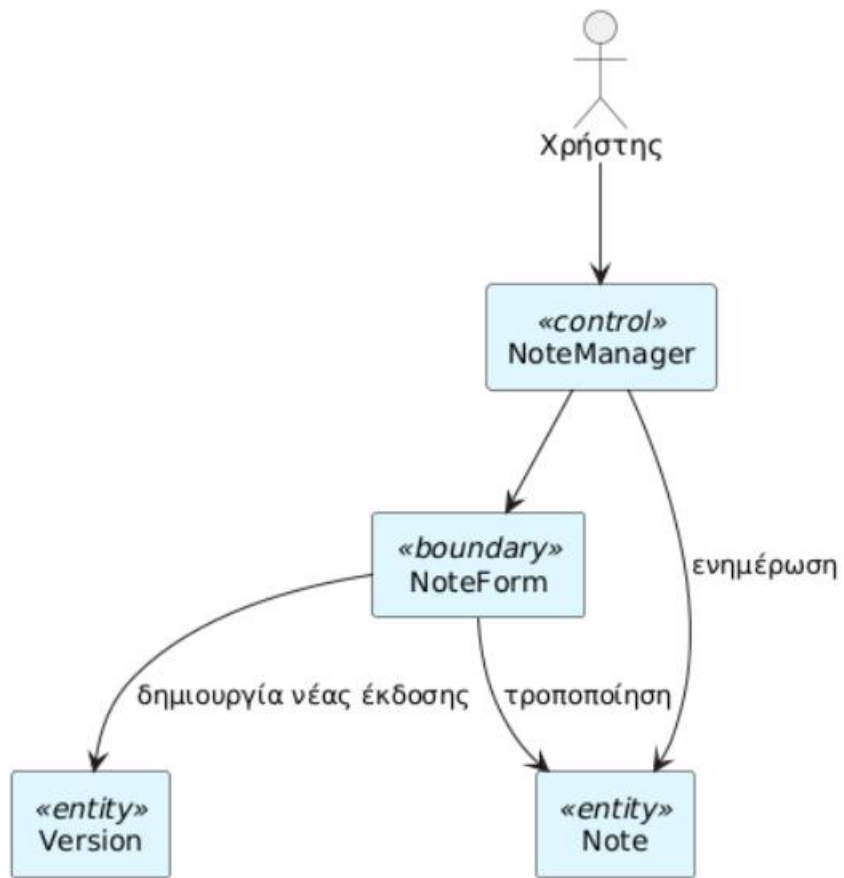
### **UC12 – Καρφίτσωμα Σημείωσης (νέο)**

Ο χρήστης μπορεί να καρφιτσώσει μια σημείωση ώστε αυτή να εμφανίζεται πάντα στην κορυφή της λίστας σημειώσεων. Επιλέγει τη σημείωση και πατά το κουμπί καρφιτσώματος (αστέρι). Αν η σημείωση είναι ήδη καρφιτσωμένη, το ίδιο κουμπί την ξεκαρφιτσώνει.

#### 4. Robustness Diagrams (v1.0) Use-Cases 1-12

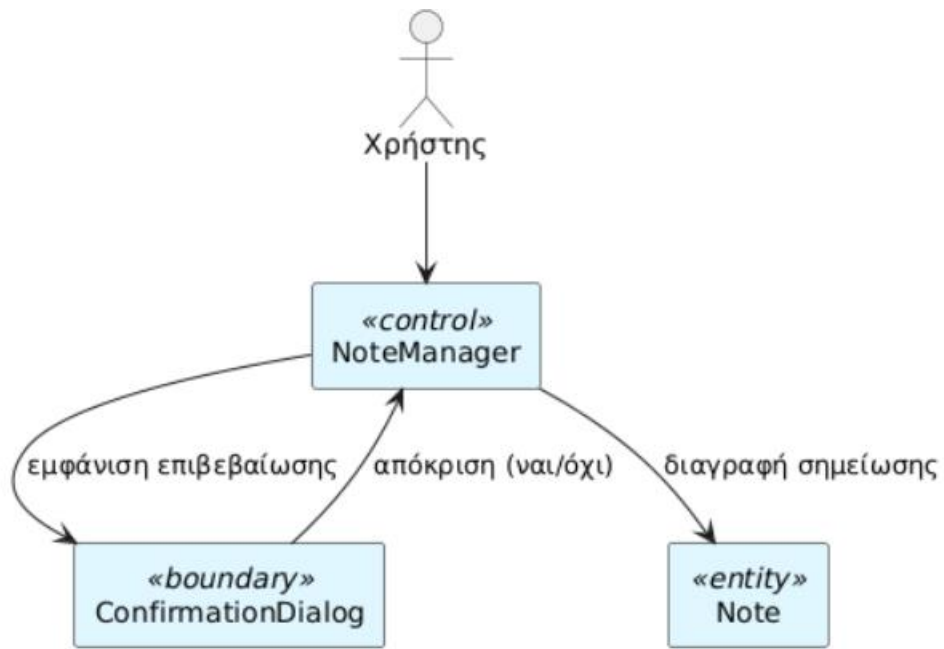


1 Usecase1

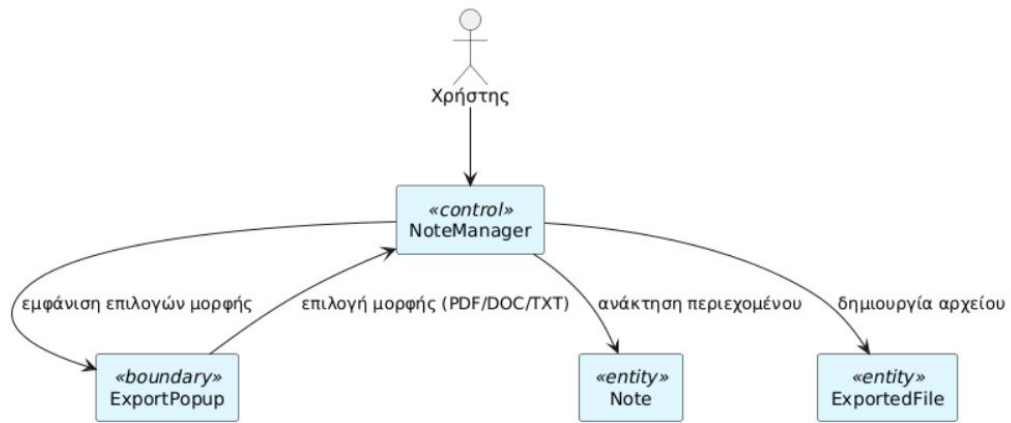


2Usecase

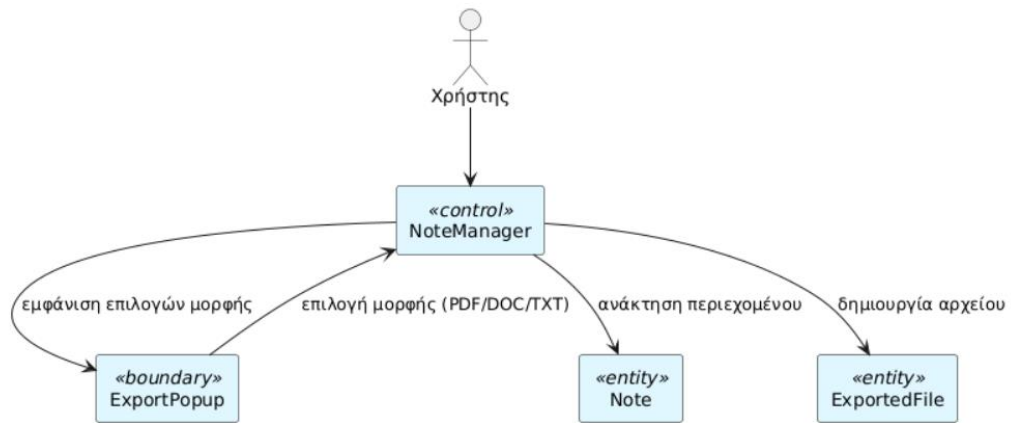




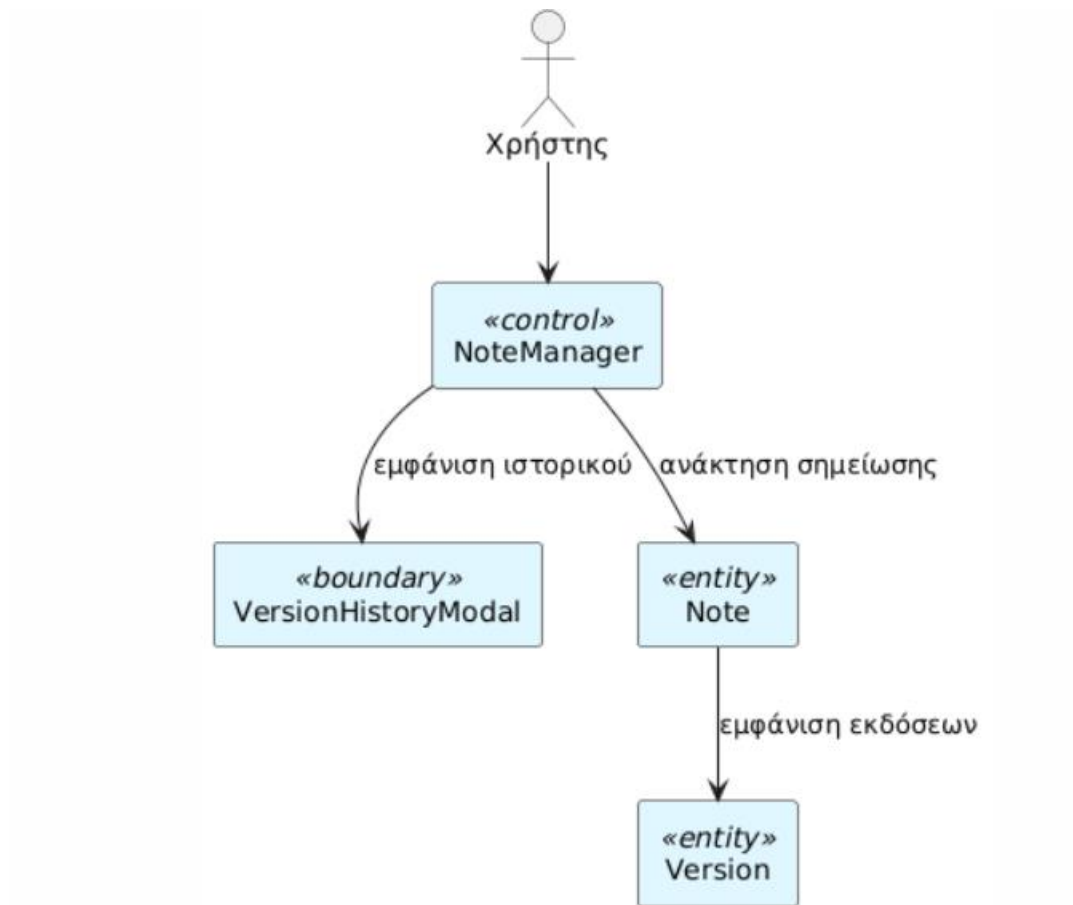
3Usecse



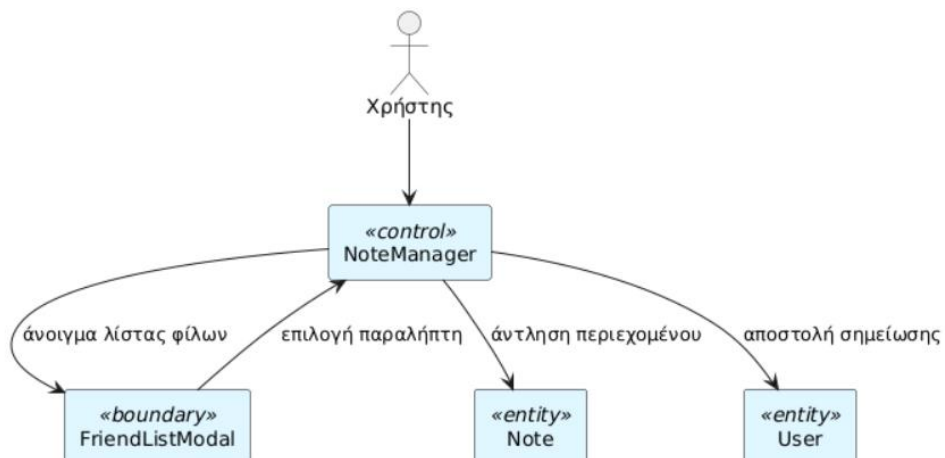
#### 4Usecase



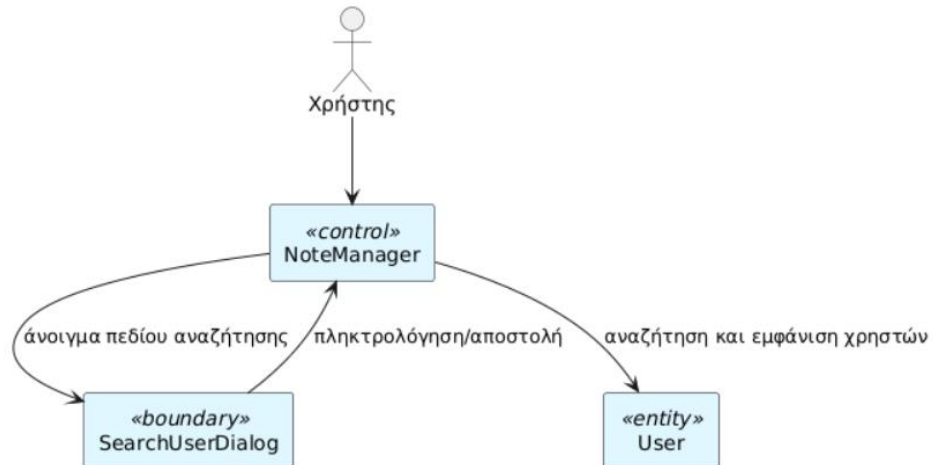
#### 5Usecase



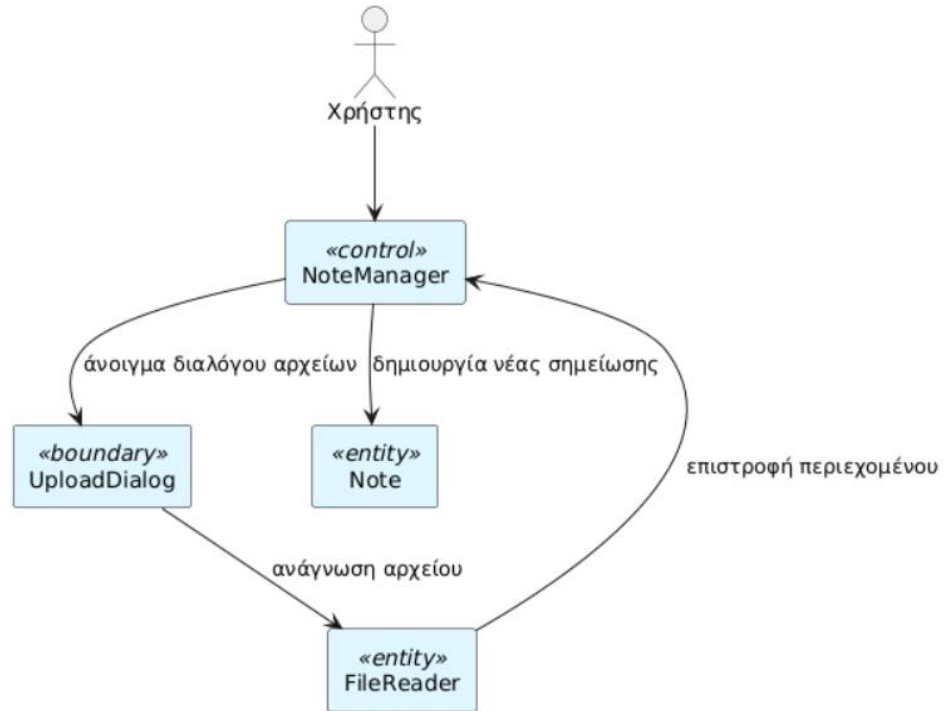
6Usecase



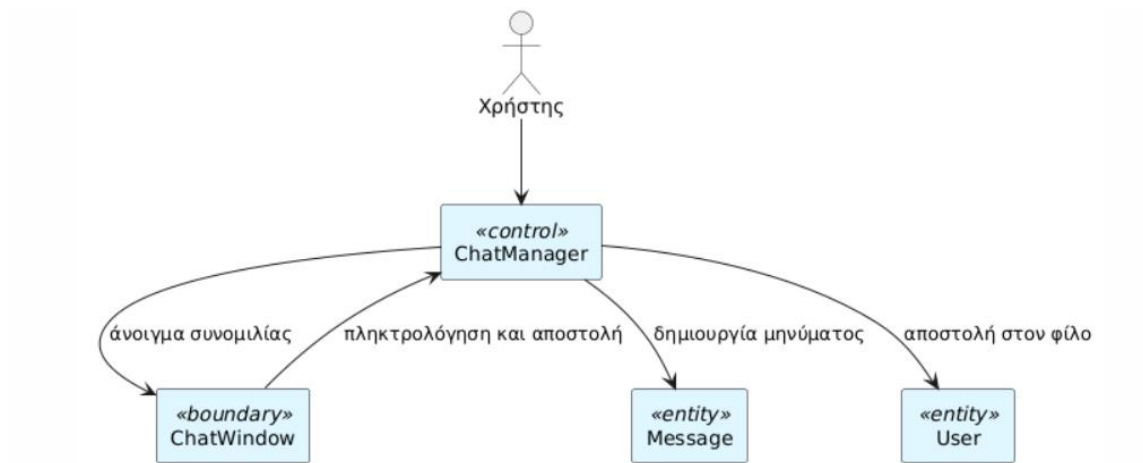
7Usecase



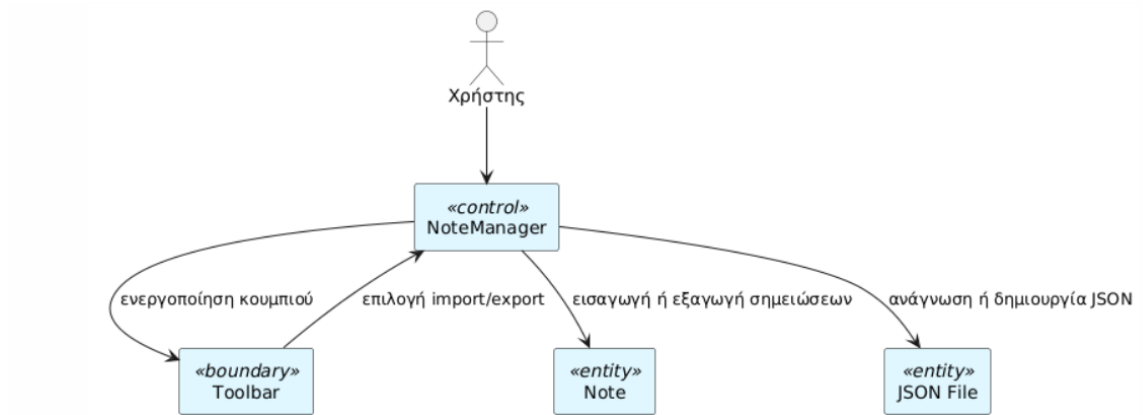
8Usecase



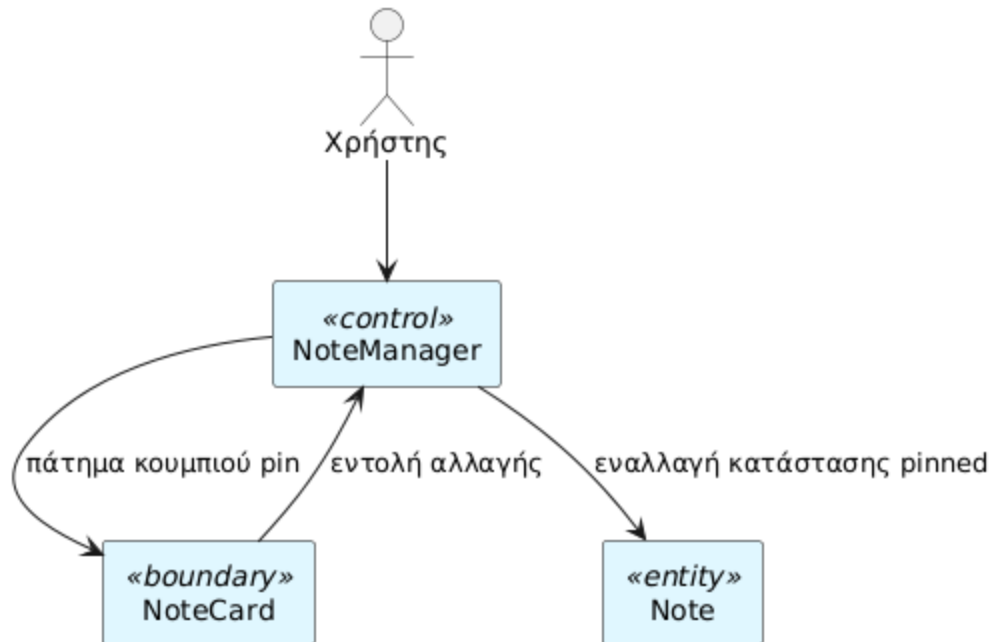
9Usecase



10Usecase

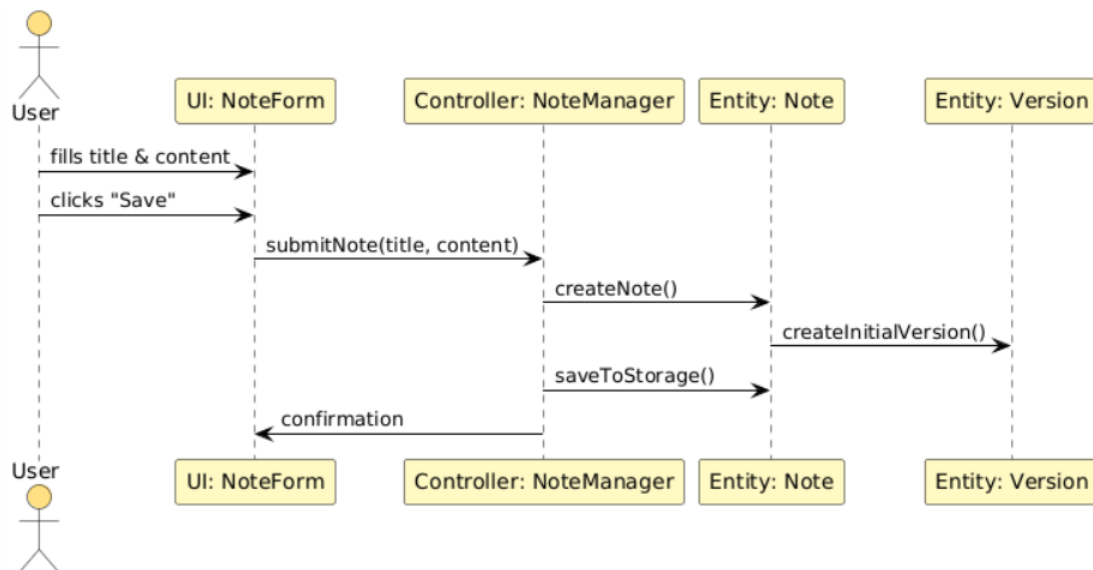


11Usecase

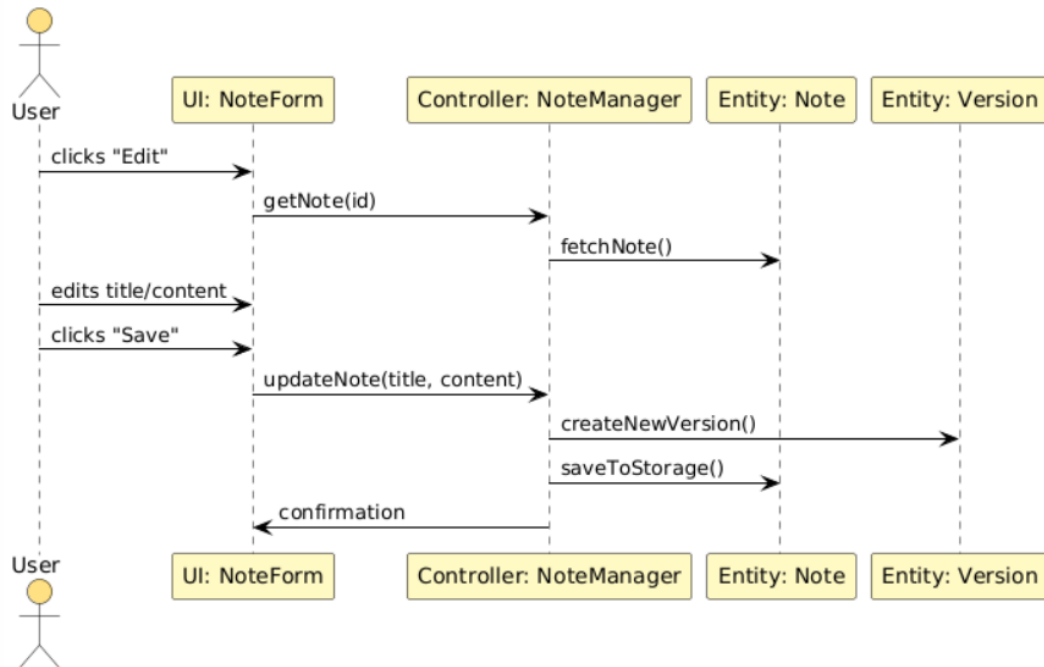


12Usecase

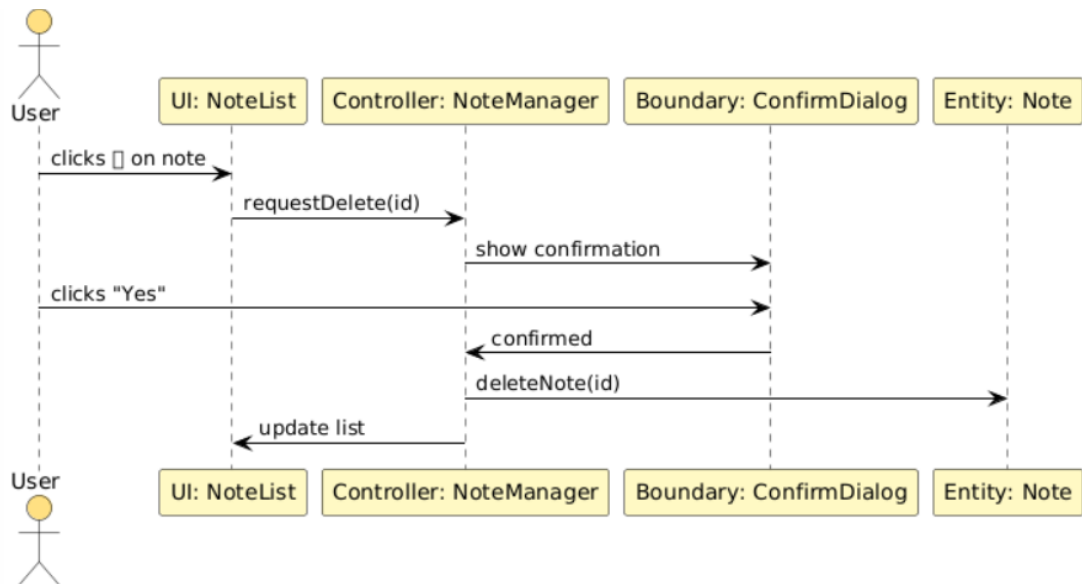
## 5. Sequence Diagrams (v1.0) Use-Cases 1-12



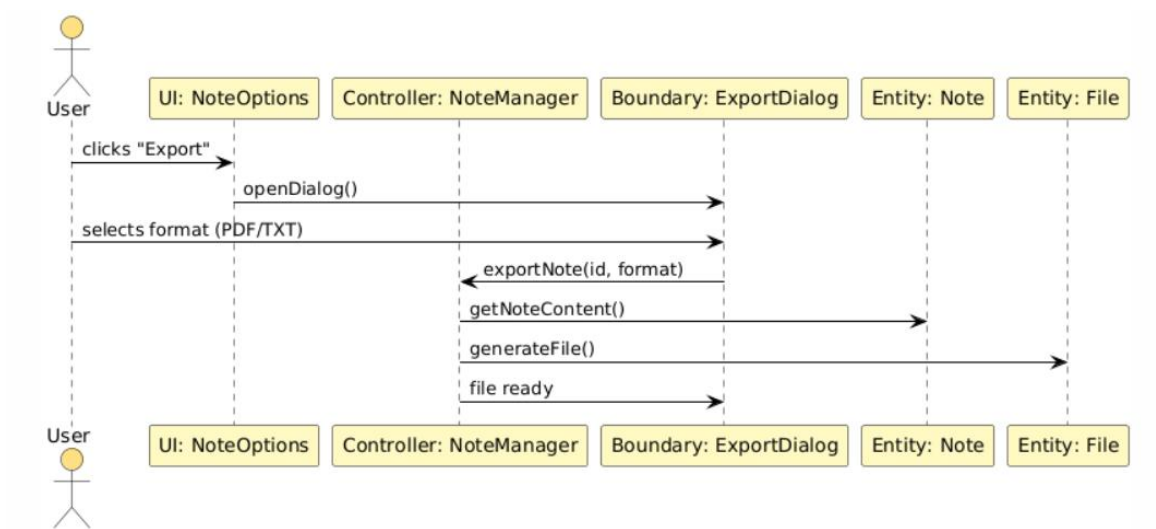
Usecase 1



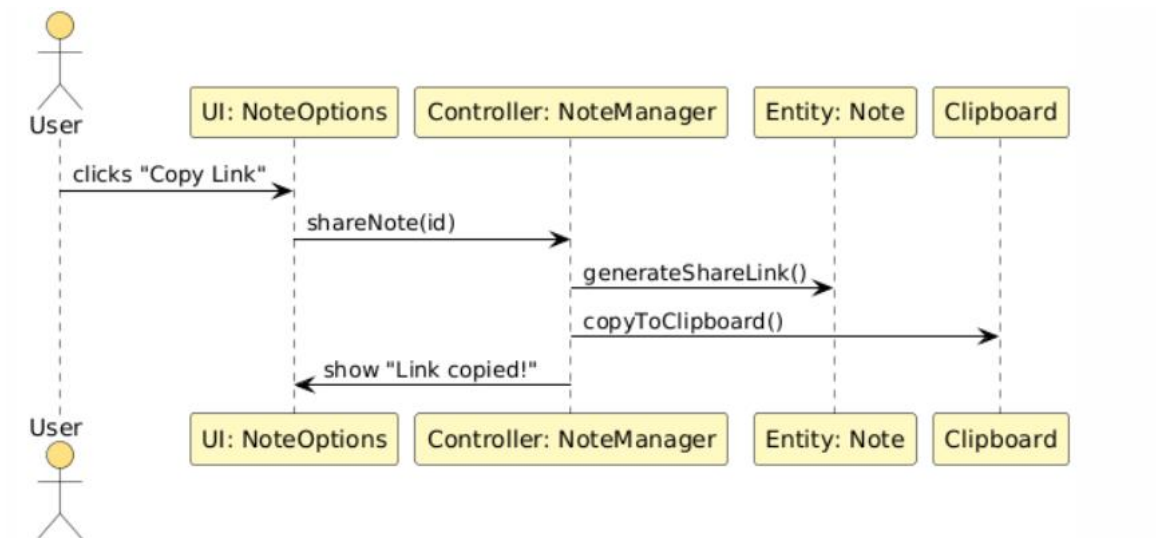
Usecase 2



Usecase 3

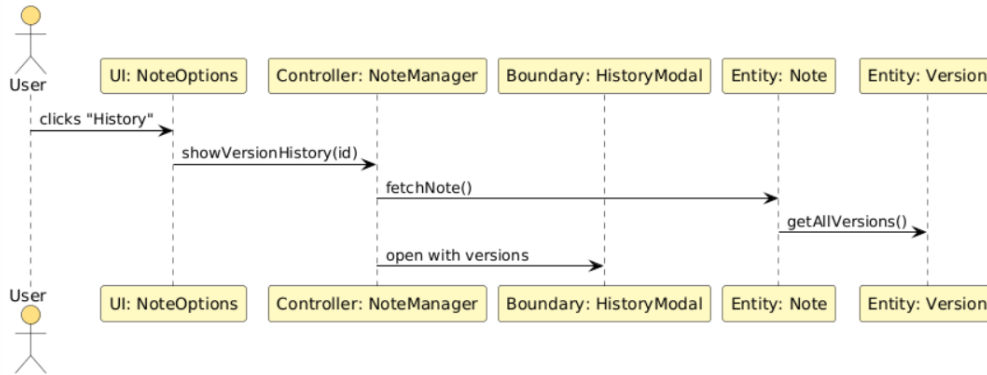


Usecase 4

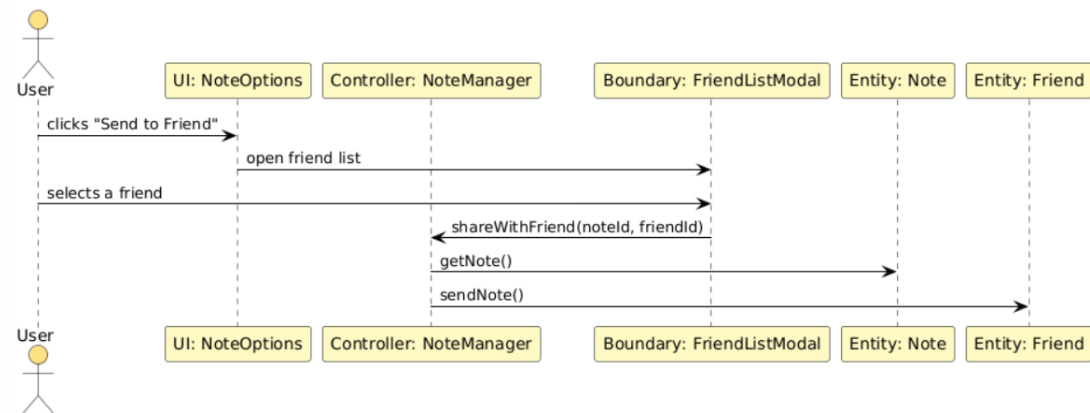


Usecase 5

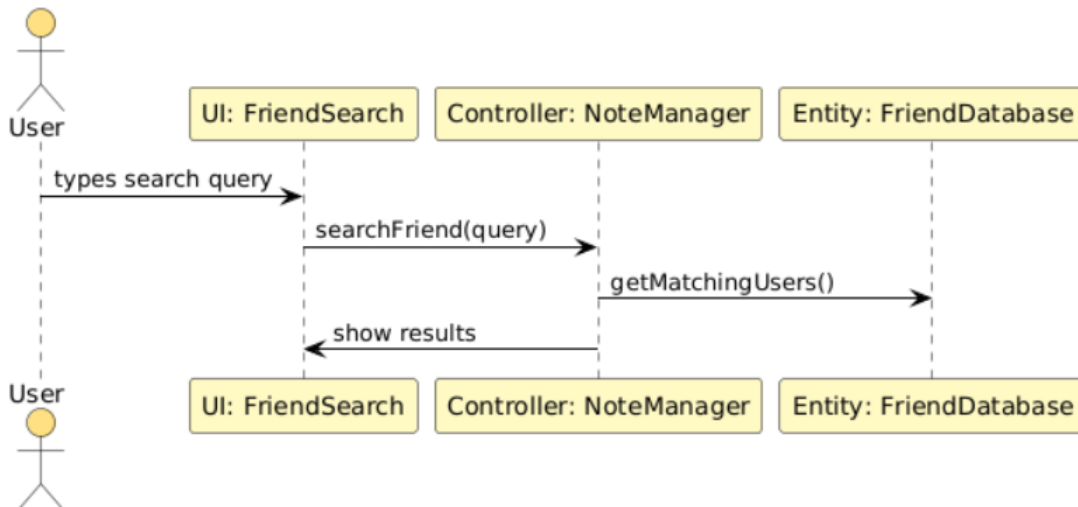




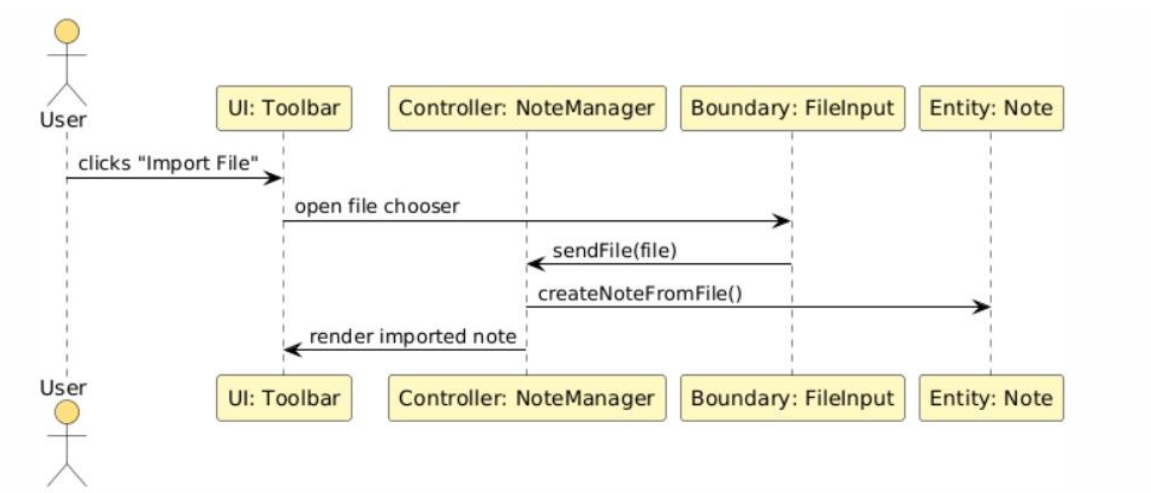
## Usecase 6



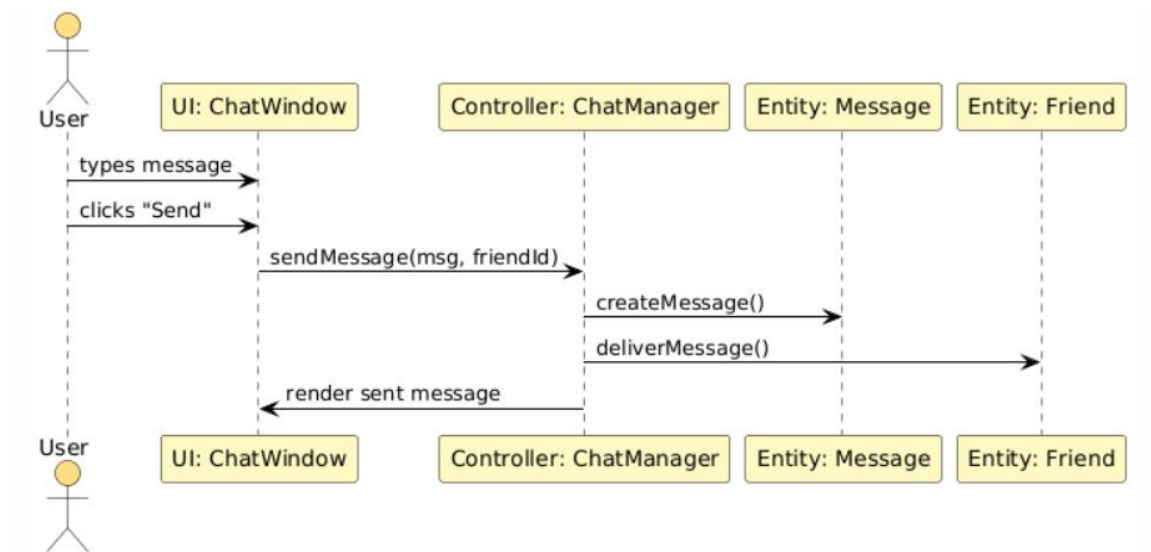
## Usecase 7



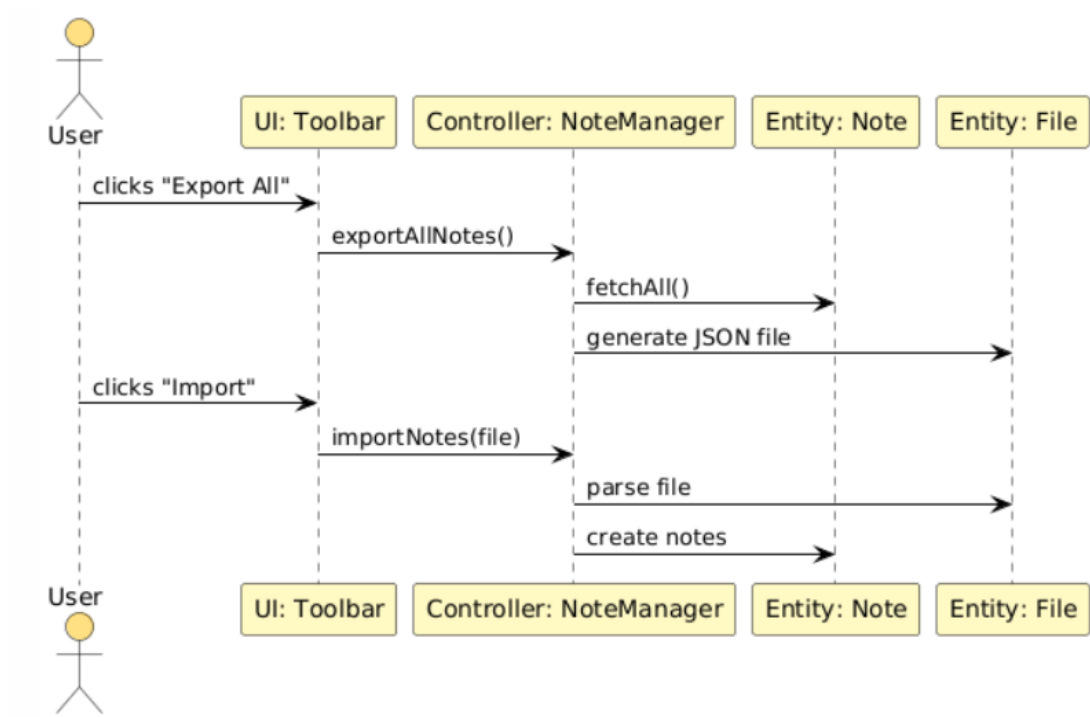
## Usecase 8



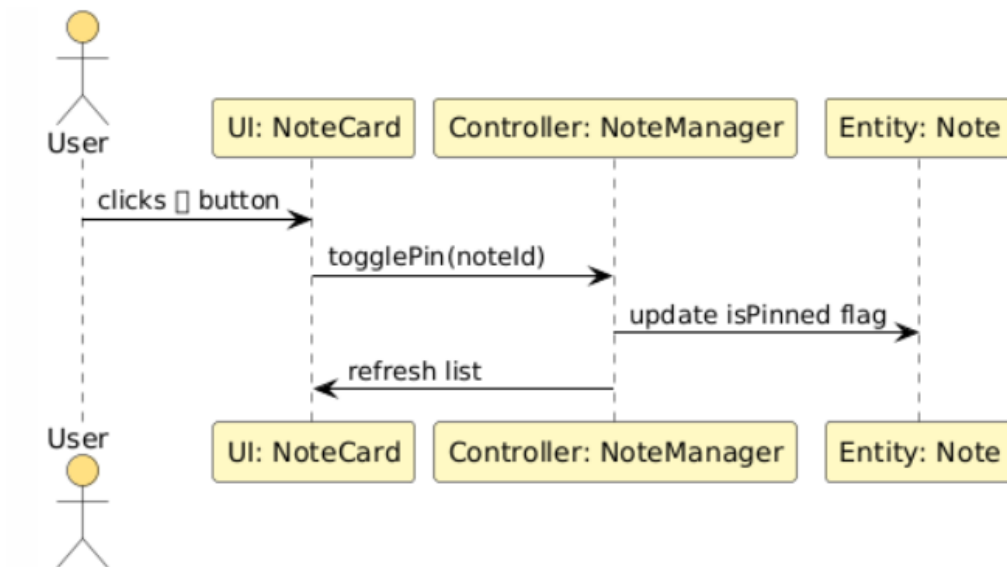
Usecase 9



Usecase 10














Usecase 11



Usecase 12

## 6. Test Cases (v1.0)

	A	B	C	D	E	F
1	Test Case ID	Use Case	Σκοπός	Είσοδος	Αναμενόμενο Αποτέλεσμα	Βήματα
2	TC01	UC1 – Δημιουργία Σημείωσης	Να δημιουργηθεί νέα σημείωση	Τίτλος: "To Do", Περιεχόμενο: "Buy milk"	Εμφάνιση σημείωσης στη λίστα	1. Γράψε τίτλο & περιεχόμενο 2. Πάτα "Αποθήκευση"
3	TC02	UC2 – Επεξεργασία Σημείωσης	Τροποποίηση υπάρχουσας σημείωσης	Αλλαγή τίτλου σε "Updated"	Η σημείωση ενημερώνεται	1. Πάτα  2. Τροποποίησε και αποθήκευσε
4	TC03	UC3 – Διαγραφή Σημείωσης	Διαγραφή υπάρχουσας σημείωσης	Πάτημα κουμπιού 	Η σημείωση αφαιρείται	1. Εντόπισε τη σημείωση 2. Πάτα 
5	TC04	UC4 – Εξαγωγή Σημείωσης	Εξαγωγή σε αρχείο	Επιλογή μορφής (PDF)	Λήψη αρχείου στον υπολογιστή	1. Πάτα  2. Διάλεξε "PDF"
6	TC05	UC5 – Μοιράζεται Σημείωση (link)	Δημιουργία κοινοποιήσιμου συνδέσμου	Πάτημα κουμπιού 	Το URL αντιγράφεται στο πρόχειρο	1. Πάτα  2. Επαλήθευσε toast
7	TC06	UC6 – Ιστορικό Εκδόσεων	Να εμφανιστεί ιστορικό και επαναφορά	Πατημένο  σε σημείωση με εκδόσεις	Εμφανίζεται modal με εκδόσεις	1. Πάτα  2. Επαλήθευσε preview & restore
8	TC07	UC7 – Ανεβάζει Αρχείο	Να μετατραπεί αρχείο σε σημείωση (pdf/doc/txt)	Επιλογή αρχείου σε μορφή .pdf / .doc / .txt	Δημιουργία νέας σημείωσης από το αρχείο (εκτός PDF)	1. Πάτα "Εισαγωγή Αρχείου" 2. Διάλεξε αρχείο
9	TC08	UC8 – Αναζητά Σημείωση	Να φιλτραριστούν οι σημειώσεις	Αναζήτηση: "milk"	Προβάλλονται μόνο σχετικές σημειώσεις	1. Πληκτρολόγησε "milk" στο search 2. Διάλεξε φίλο
10	TC09	UC9 – Chat με Φίλους	Αποστολή μηνύματος και προβολή ιστορικού	Μήνυμα: "Γεια σου!"	Το μήνυμα αποθηκεύεται και εμφανίζεται	3. Στείλε μήνυμα 1. Πάτα  2. Διάλεξε φίλο
11	TC10	UC10 – Κοινοποιεί Σημείωση σε Φίλο	Να σταλεί σημείωση σε φίλο	Επιλογή χρήστη + "Αποστολή"	Εμφανίζεται μήνυμα στο ιστορικό του	3. Πάτα "Αποστολή" 1. Πάτα "Αναζήτηση Χρήστη" 2. Πληκτρολόγησε "nikos"
12	TC11	UC11.1 – Αναζήτηση Χρήστη	Να βρεθεί χρήστης βάσει email	Πληκτρολόγηση "nikos"	Εμφάνιση σχετικών αποτελεσμάτων	3. Επαλήθευσε
13	TC12	UC11.2 – Καρφίτσωμα Σημείωσης	Επισήμανση σημαντικών σημειώσεων	Πάτημα κουμπιού 	Η σημείωση εμφανίζεται καρφιτσωμένη	1. Πάτα το  δίπλα από τίτλο

## 7. Κώδικας (v1.0)

```
<!DOCTYPE html>

<html lang="el">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>StudentNote</title>

  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;700&display=swap"
rel="stylesheet">

  <style>

:root {

  --bg-color: #f9fafb;

  --text-color: #333;
```

```
--primary-color: #2a2f45;
--button-hover: #1d2133;
}
* { box-sizing: border-box; }
body {
  font-family: 'Inter', sans-serif;
  max-width: 900px;
  margin: auto;
  padding: 2rem;
  background-color: var(--bg-color);
  color: var(--text-color);
}
.note-button {
  background: var(--button-bg);
  color: white;
  border: none;
  border-radius: 12px;
  font-weight: bold;
  cursor: pointer;
  transition: background 0.3s;
}
.note-button:hover {
  background: var(--button-hover);
}
.toolbar {
  display: flex;
```

```
flex-wrap: wrap;

gap: 0.5rem;

margin-top: 2rem;
}

.note {

background: var(--note-bg);

color: var(--note-text);

border-radius: 12px;

padding: 1rem;

box-shadow: 0 2px 8px rgba(0,0,0,0.08);

margin-top: 1rem;

transition: transform 0.2s;
}

.note:hover {

transform: scale(1.01);
}

input[type="text"], textarea, select {

width: 100%;

padding: 0.75rem;

margin-top: 1rem;

border-radius: 8px;

border: 1px solid #ccc;

font-family: inherit;
}

.note-header {

display: flex;
```

```
    justify-content: space-between;

    align-items: center;

    margin-bottom: 0.5rem;
}

.note-title {

    font-size: 1.2rem;

    font-weight: bold;
}

.note-meta {

    font-size: 0.8rem;

    color: var(--text-color);

    opacity: 0.75;

    margin-top: 0.5rem;
}

.toast {

    position: fixed;

    bottom: 1rem;

    right: 1rem;

    background: #323232;

    color: white;

    padding: 1rem;

    border-radius: 8px;

    opacity: 0;

    animation: fadeInOut 3s forwards;

    z-index: 2000;
}
```

```
body:not(.dark) .toast {  
  background: #2a2f45;  
  color: white;  
}  
  
.modal {  
  position: fixed;  
  top: 0; left: 0; width: 100%; height: 100%;  
  background-color: rgba(0,0,0,0.5);  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  z-index: 1000;  
  backdrop-filter: blur(8px);  
}  
  
.modal-box {  
  background: var(--note-bg);  
  color: var(--text-color);  
  padding: 1.5rem;  
  border-radius: 16px;  
  box-shadow: 0 8px 32px rgba(0, 0, 0, 0.25);  
  max-height: 80%;  
  overflow-y: auto;  
  width: 360px;  
  transition: all 0.3s ease;  
}  
  
body.dark .modal-box {
```



```
background: rgba(32, 32, 48, 0.8);
border: 1px solid rgba(255, 255, 255, 0.1);
}

.modal-box button {
background: var(--button-bg);
color: white;
border: none;
border-radius: 999px;
padding: 0.5rem 1.2rem;
font-weight: bold;
cursor: pointer;
margin: 0.25rem 0.2rem;
transition: background-color 0.3s;
}

.modal-box button:hover {
background: var(--button-hover);
}

.modal-box h3 {
font-size: 1.3rem;
margin-bottom: 1rem;
text-align: center;
}

@keyframes fadeInOut {
0% { opacity: 0; }
10% { opacity: 1; }
90% { opacity: 1; }
```

```
100% { opacity: 0; }  
}  
  
:root {  
  --bg-color: #f9fafb;  
  --text-color: #333;  
  --note-bg: #ffffff;  
  --note-text: #000000;  
  --button-bg: #2a2f45;  
  --button-hover: #1d2133;  
}  
  
body.dark {  
  --bg-color: #1c1e26;  
  --text-color: #f0f0f0;  
  --note-bg: #2a2f3c;  
  --note-text: #ffffff;  
  --button-bg: #8bc34a; /* Λαχανί χρώμα */  
  --button-hover: #76a931;  
}  
  
#darkModeToggle {  
  position: absolute;  
  top: 1rem;  
  right: 1rem;  
  padding: 0.5rem 1rem;  
  border-radius: 8px;  
  background-color: var(--button-bg);  
  color: white;
```

```
font-weight: bold;

border: none;

cursor: pointer;

transition: background-color 0.3s;
}

#darkModeToggle:hover {

background-color: var(--button-hover);
}

.note-button {

display: block;

width: 60%;

margin: 2rem auto 1rem auto;

font-size: 1.1rem;

background: var(--button-bg);

color: white;

border: none;

border-radius: 12px;

font-weight: bold;

cursor: pointer;

transition: background 0.3s;

padding: 1rem;
}

.note-button:hover {

background: var(--button-hover);
}

.action-grid {
```

```
display: flex;

justify-content: space-between;

gap: 1rem;

margin-top: 2rem;
}

.action-card {

flex: 1;

background: var(--note-bg);

color: var(--note-text);

padding: 1.2rem;

text-align: center;

border-radius: 12px;

box-shadow: 0 2px 10px rgba(0,0,0,0.1);

cursor: pointer;

font-size: 1.1rem;

transition: transform 0.2s, background 0.2s;
}

.action-card:hover {

background: var(--button-bg);

color: white;

transform: scale(1.03);
}

.note-header button {

background-color: transparent;

border: none;

font-size: 1.2rem;
```

```
    cursor: pointer;

    padding: 0.3rem 0.5rem;

    border-radius: 8px;

    transition: background-color 0.2s ease;
}
```

```
.note-header button:hover {

    background-color: rgba(0, 0, 0, 0.1);
}
```

```
body.dark .note-header button:hover {

    background-color: rgba(255, 255, 255, 0.1);
}
```

```
.toolbar button {

    background: var(--button-bg);

    color: white;

    border: none;

    padding: 0.6rem 1rem;

    border-radius: 8px;

    font-weight: bold;

    cursor: pointer;

    transition: background 0.3s;
}
```

```
.toolbar button:hover {

    background: var(--button-hover);
}
```

```
.tooltip {
```

```
    position: relative;
    display: inline-block;
}
.tooltip .tooltiptext {
    visibility: hidden;
    width: auto;
    background-color: #333;
    color: #fff;
    text-align: center;
    border-radius: 6px;
    padding: 0.5rem;
    position: absolute;
    z-index: 1001;
    bottom: 125%;
    left: 50%;
    transform: translateX(-50%);
    opacity: 0;
    transition: opacity 0.3s;
    white-space: nowrap;
}
.tooltip:hover .tooltiptext {
    visibility: visible;
    opacity: 1;
}
</style>
</head>
```

```
<body>

  <h1>StudentNote</h1>

  <div class="action-grid">

    <div class="action-card" onclick="uploadNoteFile()">📄 Εισαγωγή Αρχείου</div>

    <div class="action-card" onclick="openChatMock()">💬 Συνομιλία</div>

    <div class="action-card" onclick="searchUserMock()">🔍 Αναζήτηση Χρήστη</div>

  </div>


  <div class="toolbar">

    <input type="text" id="searchBox" placeholder="Αναζήτηση σημειώσεων..."
    oninput="renderNotes()">

    <select id="sortOrder" onchange="renderNotes()">

      <option value="title">Αλφαβητικά</option>

      <option value="date">Κατά Ημερομηνία</option>

    </select>

    <button onclick="undoDelete()">↶ Αναίρεση Διαγραφής</button>

    <button onclick="exportAll()">📁 Εξαγωγή Όλων</button>

    <button onclick="importNotes()">📁 Εισαγωγή</button>

  </div>


  <h2>Νέα / Επεξεργασία Σημείωσης</h2>

  <input type="text" id="noteTitle" placeholder="Τίτλος">

  <textarea id="noteContent" rows="4" placeholder="Περιεχόμενο..."></textarea>

  <button class="note-button" onclick="saveNote()">Αποθήκευση</button>

  <button id="darkModeToggle" onclick="toggleDarkMode()">🌙 Dark Mode</button>
```

```
<h2>Οι Σημειώσεις μου</h2>
```

```
<div id="notesContainer"></div>
```

```
<script>
```

```
let notes = [], deletedNote = null, editingId = null;
```

```
window.onload = () => {
```

```
  const saved = localStorage.getItem("notesJSON");
```

```
  if (saved) notes = JSON.parse(saved);
```

```
  renderNotes();
```

```
};
```

```
function saveNote() {
```

```
  const title = document.getElementById("noteTitle").value.trim();
```

```
  const content = document.getElementById("noteContent").value.trim();
```

```
  if (!title) return showToast("Ο τίτλος είναι υποχρεωτικός");
```

```
  const timestamp = new Date().toISOString();
```

```
  if (editingId !== null) {
```

```
    const note = notes.find(n => n.id === editingId);
```

```
    if (note) {
```

```
      note.versions.push({ title: note.title, content: note.content, date: note.dateEdited });
```

```
      note.title = title;
```

```
      note.content = content;
```

```
      note.dateEdited = timestamp;
```

```
    }
```

```
    editingId = null;
```

```
  } else {
```



```
    notes.push({ id: Date.now(), title, content, dateCreated: timestamp, dateEdited:
timestamp, versions: [] });
}

document.getElementById("noteTitle").value = "";
document.getElementById("noteContent").value = "";
updateStorage();
renderNotes();
}
```

```
function deleteNote(id) {
    const index = notes.findIndex(n => n.id === id);
    if (index !== -1) {
        deletedNote = notes[index];
        notes.splice(index, 1);
        updateStorage();
        renderNotes();
        showToast("Διαγράφηκε. Πατήστε αναίρεση αν έγινε κατά λάθος.");
    }
}
```

```
function editNote(id) {
    const note = notes.find(n => n.id === id);
    if (!note) return;
    editingId = id;
    document.getElementById("noteTitle").value = note.title;
    document.getElementById("noteContent").value = note.content;
```

```
}
```

```
function exportVersion(version, title, format) {  
  const filename = `${title.replace(/\s+/g, '_')}_version.${format}`;  
  if (format === "pdf") {  
    const doc = new window.jspdf.jsPDF();  
    doc.setFont("Times", "normal");  
    doc.setFontSize(12);  
    const lines = doc.splitTextToSize(`${version.title}\n\n${version.content}`, 180);  
    doc.text(lines, 10, 10);  
    doc.save(filename);  
  } else if (format === "doc") {  
    const blob = new Blob([`${version.title}\n\n${version.content}`], { type:  
'application/msword' });  
    const link = document.createElement("a");  
    link.href = URL.createObjectURL(blob);  
    link.download = filename;  
    link.click();  
  } else {  
    const blob = new Blob([`${version.title}\n\n${version.content}`], { type: 'text/plain' });  
    const link = document.createElement("a");  
    link.href = URL.createObjectURL(blob);  
    link.download = filename;  
    link.click();  
  }  
}
```

```
function showVersions(id) {

  document.querySelectorAll('.modal').forEach(m => m.remove());

  const note = notes.find(n => n.id === id);

  if (!note || !note.versions.length) return showToast("Δεν υπάρχουν εκδόσεις.");


  const modal = document.createElement("div");

  modal.className = "modal";

  const box = document.createElement("div");

  box.className = "modal-box";

  box.className = "modal-box";

  box.setAttribute("role", "dialog");

  box.setAttribute("aria-label", "Ιστορικό Εκδόσεων");

  const title = document.createElement("h3");

  title.innerText = "Ιστορικό Εκδόσεων";

  box.appendChild(title);


  note.versions.slice().reverse().forEach((v, i) => {

    const preview = document.createElement("p");

    preview.className = "note-meta";

    preview.innerText = `Τίτλος: ${v.title}\nΠροεπισκόπηση: ${v.content.substring(0, 60)}...`;


    const restoreBtn = document.createElement("button");

    restoreBtn.innerText = `Επαναφορά (${new Date(v.date).toLocaleString()}`;

    restoreBtn.style.marginBottom = "0.5rem";

    restoreBtn.onclick = () => {

      note.title = v.title;
```

```
note.content = v.content;

note.dateEdited = new Date().toISOString();

updateStorage();

modal.remove();

renderNotes();

};
```

```
const exportMenuBtn = document.createElement("button");

exportMenuBtn.innerText = "📎 Εξαγωγή";

exportMenuBtn.onclick = () => showExportPopup(v, note.title);
```

```
const btns = document.createElement("div");

btns.style.marginBottom = "0.5rem";

btns.appendChild(restoreBtn);

btns.appendChild(exportMenuBtn);
```

```
box.appendChild(preview);

box.appendChild(btns);

});
```

```
const cancel = document.createElement("button");

cancel.innerText = "Κλείσιμο";

cancel.onclick = () => modal.remove();

box.appendChild(cancel);
```

```
modal.appendChild(box);
```

```
document.body.appendChild(modal);  
}
```

```
function showExportPopup(version, title) {  
  document.querySelectorAll('.modal').forEach(m => m.remove());  
  const popup = document.createElement("div");  
  popup.className = "modal";  
  const box = document.createElement("div");  
  box.className = "modal-box";
```

```
  const header = document.createElement("h3");  
  header.innerText = "Επιλέξτε μορφή εξαγωγής";  
  box.appendChild(header);
```

```
  ["txt", "pdf", "doc"].forEach(format => {  
    const btn = document.createElement("button");  
    btn.innerText = format.toUpperCase();  
    btn.style.margin = "0.25rem";  
    btn.onclick = () => {  
      exportVersion(version, title, format);  
      popup.remove();  
    };  
    box.appendChild(btn);  
  });
```

```
  const cancel = document.createElement("button");
```

```
cancel.innerText = "Ακυρο";  
cancel.onclick = () => popup.remove();  
cancel.style.marginTop = "1rem";  
box.appendChild(cancel);
```

```
popup.appendChild(box);  
document.body.appendChild(popup);  
}
```

```
function copyShareLink(id) {  
  const url = `${window.location.origin}${window.location.pathname}#noteId=${id}`;  
  navigator.clipboard.writeText(url)  
    .then(() => showToast("Ο σύνδεσμος αντιγράφηκε στο πρόχειρο!"))  
    .catch(() => showToast("Αποτυχία αντιγραφής συνδέσμου."));  
}
```

```
function loadSharedNote() {  
  const hash = window.location.hash;  
  if (hash.startsWith("#noteId=")) {  
    const id = parseInt(hash.split("=")[1]);  
    const note = notes.find(n => n.id === id);  
    if (!note) return;
```

```
document.body.innerHTML = `  
  <div style="max-width: 800px; margin: 3rem auto; font-family: Inter, sans-serif;">  
    <h2>🔗 Προβολή Κοινοποιημένης Σημείωσης</h2>
```

```

    <div class="note">

      <div class="note-header"><span class="note-title">${note.title}</span></div>

      <div>${note.content}</div>

      <div class="note-meta">Ημερομηνία: ${new
Date(note.dateEdited).toLocaleString()}</div>

    </div>

  </div>

  `;
}
}

```

```

window.onload = () => {
  const saved = localStorage.getItem("notesJSON");
  if (saved) notes = JSON.parse(saved);
  if (window.location.hash.startsWith("#noteId=")) {
    loadSharedNote();
  } else {
    renderNotes();
  }
};

```

```

function exportVersion(version, title, format) {
  const cleanContent = `${version.title}\n\n${version.content}`.replace(/\n/g, '\r\n');
  const filename = `${title.replace(/\s+/g, '_')}_version.${format}`;
  if (format === "pdf") {
    const doc = new window.jspdf.jsPDF();

```

```
doc.setFont("Times", "normal");

doc.setFontSize(12);

const lines = doc.splitTextToSize(cleanContent, 180);

doc.text(lines, 10, 10);

doc.save(filename);

} else if (format === "doc") {

  const blob = new Blob([cleanContent], { type: 'application/msword' });

  const link = document.createElement("a");

  link.href = URL.createObjectURL(blob);

  link.download = filename;

  link.click();

} else {

  const blob = new Blob([cleanContent], { type: 'text/plain' });

  const link = document.createElement("a");

  link.href = URL.createObjectURL(blob);

  link.download = filename;

  link.click();

}

}
```

```
function exportMain(id) {

  const note = notes.find(n => n.id === id);

  if (!note) return;

  showExportPopup(note, note.title);

}
```



```

function renderNotes(showOnlyPinned = false) {

  const container = document.getElementById("notesContainer");

  container.innerHTML = "";

  const search = document.getElementById("searchBox").value.toLowerCase();

  const sort = document.getElementById("sortOrder").value;

  let displayNotes = [...notes].filter(n =>

    (n.title.toLowerCase().includes(search) || n.content.toLowerCase().includes(search)) &&

    (!showOnlyPinned || n.isPinned)

  );

  displayNotes.sort((a, b) => {

    if (b.isPinned !== a.isPinned) return b.isPinned - a.isPinned;

    if (sort === "title") return a.title.localeCompare(b.title);

    return new Date(b.dateEdited) - new Date(a.dateEdited);

  });

  displayNotes.forEach(note => {

    const div = document.createElement("div");

    div.className = "note";

    div.innerHTML = `

      <div class="note-header">

        <span class="note-title">${note.title} <button onclick="togglePin(${note.id})"

title="${note.isPinned ? 'Ξεκαρφίτσωμα' : 'Καρφίτσωμα'}" style="background:none;

border:none; cursor:pointer; font-size:1.1rem">${note.isPinned ? '☆' : '☆'}

'</button></span>

      <div>

        <button onclick="editNote(${note.id})" title="Επεξεργασία">  </button>

```

```

        <button onclick="deleteNote(${note.id})" title="Διαγραφή">🗑️</button>

        <button onclick="showVersions(${note.id})" title="Ιστορικό
Εκδόσεων">📄</button>

        <button onclick="exportMain(${note.id})" title="Εξαγωγή">📁</button>

        <button onclick="copyShareLink(${note.id})" title="Κοινοποίηση με
Σύνδεσμο">🔗</button>

        <button onclick="shareNoteToFriend(${note.id})" title="Κοινοποίηση σε
Φίλο">👤</button>

    </div>

</div>

<div>${note.content}</div>

<div class="note-meta">Τελευταία ενημέρωση: ${new
Date(note.dateEdited).toLocaleString()}</div>

`;

container.appendChild(div);

});

}

function exportOne(id) {

    const note = notes.find(n => n.id === id);

    if (!note) return;

    const blob = new Blob([note.title + "\n\n" + note.content], { type: 'text/plain' });

    const link = document.createElement("a");

    link.href = URL.createObjectURL(blob);

    link.download = `${note.title.replace(/\s+/g, '_')}.txt`;

    link.click();

}

```

```
function exportAll() {  
    const blob = new Blob([JSON.stringify(notes, null, 2)], { type: 'application/json' });  
    const link = document.createElement("a");  
    link.href = URL.createObjectURL(blob);  
    link.download = "StudentNotes.json";  
    link.click();  
}
```

```
function importNotes() {  
    const input = document.createElement("input");  
    input.type = "file";  
    input.accept = "application/json";  
    input.onchange = e => {  
        const file = e.target.files[0];  
        if (!file) return;  
        const reader = new FileReader();  
        reader.onload = () => {  
            try {  
                notes = JSON.parse(reader.result);  
                updateStorage();  
                renderNotes();  
                showToast("Επιτυχής εισαγωγή σημειώσεων.");  
            } catch {  
                showToast("Σφάλμα κατά την εισαγωγή.");  
            }  
        };  
    };  
};
```

```
        reader.readAsText(file);

    };

    input.click();
}

function updateStorage() {

    localStorage.setItem("notesJSON", JSON.stringify(notes));
}

function showToast(msg) {

    const toast = document.createElement("div");

    toast.className = "toast";

    toast.innerText = msg;

    document.body.appendChild(toast);

    setTimeout(() => toast.remove(), 3000);
}

function toggleDarkMode() {

    document.body.classList.toggle('dark');

    const isDark = document.body.classList.contains('dark');

    localStorage.setItem("theme", isDark ? "dark" : "light");
}

window.addEventListener("DOMContentLoaded", () => {

    const theme = localStorage.getItem("theme");

    if (theme === "dark") document.body.classList.add("dark");

});

function undoDelete() {
```

```
if (deletedNote) {  
  notes.push(deletedNote);  
  deletedNote = null;  
  updateStorage();  
  renderNotes();  
  showToast("Ανακτήθηκε η σημείωση.");  
} else {  
  showToast("Δεν υπάρχει διαγραφμένη σημείωση.");  
}  
}  
  
function searchUserMock() {  
  const modal = document.createElement("div");  
  modal.className = "modal";  
  
  const box = document.createElement("div");  
  box.className = "modal-box";  
  
  const title = document.createElement("h3");  
  title.innerText = "🔍 Αναζήτηση Χρήστη";  
  box.appendChild(title);  
  
  const input = document.createElement("input");  
  input.type = "text";  
  input.placeholder = "Πληκτρολόγησε όνομα ή email...";  
  
  input.style = "margin-bottom: 1rem; width: 100%; padding: 0.5rem; border-radius: 8px;  
border: 1px solid #ccc;";
```

```
box.appendChild(input);
```

```
const resultsBox = document.createElement("div");
```

```
resultsBox.style = "margin-bottom: 1rem; font-size: 0.9rem;";
```

```
box.appendChild(resultsBox);
```

```
const searchBtn = document.createElement("button");
```

```
searchBtn.innerText = "🔍 Αναζήτηση";
```

```
searchBtn.onclick = () => {
```

```
  const val = input.value.toLowerCase();
```

```
  if (!val.trim()) {
```

```
    resultsBox.innerHTML = "<i>Πληκτρολόγησε κάτι πρώτα.</i>";
```

```
    return;
```

```
  }
```

```
  const matches = ["nikos@example.com", "kostas@example.com",  
"maria@example.com"].filter(user =>
```

```
    user.toLowerCase().includes(val)
```

```
  );
```

```
  resultsBox.innerHTML = "";
```

```
  if (matches.length === 0) {
```

```
    resultsBox.innerHTML = "<i>Δεν βρέθηκαν χρήστες.</i>";
```

```
    return;
```

```
  }
```

```

matches.forEach(user => {

  const div = document.createElement("div");

  div.innerText = user;

  const addBtn = document.createElement("button");

  addBtn.innerText = "✚ Αίτημα Φιλίας";

  addBtn.onclick = () => {

    showToast(`Εστάλη αίτημα φιλίας προς ${user}`);

    modal.remove();

  };

  div.appendChild(addBtn);

  resultsBox.appendChild(div);

});

};

box.appendChild(searchBtn);


const cancelBtn = document.createElement("button");

cancelBtn.innerText = "Κλείσιμο";

cancelBtn.onclick = () => modal.remove();

cancelBtn.style.marginTop = "1rem";

box.appendChild(cancelBtn);


modal.appendChild(box);

document.body.appendChild(modal);

}

function uploadNoteFile() {

```

```
const input = document.createElement("input");

input.type = "file";

input.accept = ".txt,.doc,.docx,.pdf";

input.onChange = e => {

  const file = e.target.files[0];

  if (!file) return;


  const reader = new FileReader();

  reader.onload = () => {

    const content = reader.result;

    if (!content || content.trim() === '') {

      showToast("Το αρχείο είναι κενό ή μη υποστηριζόμενο.");

      return;

    }


    const note = {

      id: Date.now(),

      title: file.name.replace(/\.([^.]*)$/, ""),

      content: content,

      dateCreated: new Date().toISOString(),

      dateEdited: new Date().toISOString(),

      versions: []

    };


    notes.push(note);

    updateStorage();
```



```
renderNotes();

showToast("Το αρχείο μετατράπηκε σε σημείωση.");

};

if (file.type === "application/pdf") {

  showToast("Δεν υποστηρίζεται ακόμα η ανάγνωση PDF.");

} else {

  reader.readAsText(file);

}

};

input.click();

}

function shareNoteToFriend(id) {

  const note = notes.find(n => n.id === id);

  if (!note) return;

  const modal = document.createElement("div");

  modal.className = "modal";

  const box = document.createElement("div");

  box.className = "modal-box";

  const title = document.createElement("h3");

  title.innerText = "👤 Κοινοποίηση Σημείωσης";

  box.appendChild(title);
```

```
const select = document.createElement("select");

["nikos@example.com", "kostas@example.com", "maria@example.com"].forEach(user => {

  const option = document.createElement("option");

  option.value = user;

  option.innerText = user;

  select.appendChild(option);

});

box.appendChild(select);
```

```
const shareBtn = document.createElement("button");

shareBtn.innerText = "✉ Αποστολή";

shareBtn.onclick = () => {

  const historyKey = "chatHistory_" + select.value;

  const history = JSON.parse(localStorage.getItem(historyKey)) || [];

  history.push({

    sender: "Εσύ",

    message: `📩 Κοινοποίησες τη σημείωση: "${note.title}"`

  });

  localStorage.setItem(historyKey, JSON.stringify(history));

  showToast(`Η σημείωση κοινοποιήθηκε στον ${select.value}!`);

  modal.remove();

};

box.appendChild(shareBtn);
```

```
const cancelBtn = document.createElement("button");

cancelBtn.innerText = "Άκυρο";
```

```
cancelBtn.onclick = () => modal.remove();

box.appendChild(cancelBtn);


modal.appendChild(box);

document.body.appendChild(modal);
}

function openChatMock() {

  const modal = document.createElement("div");

  modal.className = "modal";


  const box = document.createElement("div");

  box.className = "modal-box";


  const title = document.createElement("h3");

  title.innerText = "📬 Συνομιλία με Φίλο";

  box.appendChild(title);


  const select = document.createElement("select");

  ["nikos@example.com", "kostas@example.com", "maria@example.com",
  "aggelos@example.com", "giwrgos@example.com" ].forEach(name => {

    const opt = document.createElement("option");

    opt.value = name;

    opt.textContent = name;

    select.appendChild(opt);

  });

  box.appendChild(select);
```

```
const chatBox = document.createElement("div");

chatBox.style = "margin-top:1rem; height:150px; background:#f0f0f0; overflow-y:auto;
padding:0.5rem; border-radius:8px; font-size:0.9rem";

box.appendChild(chatBox)


function loadHistory(user) {

  const history = JSON.parse(localStorage.getItem("chatHistory_" + user)) || [];

  if (history.length === 0) {

    chatBox.innerHTML = "<i>Δεν υπάρχει ιστορικό συνομιλίας.</i>";

  } else {

    chatBox.innerHTML = history.map(m => `<div><b>${m.sender}</b>
    ${m.message}</div>`).join("");

  }

}

select.onchange = () => loadHistory(select.value);

loadHistory(select.value);


const input = document.createElement("input");

input.type = "text";

input.placeholder = "Μήνυμα...";

input.style = "margin-top:1rem; width:100%; padding:0.5rem; border-radius:8px;
border:1px solid #ccc";

box.appendChild(input);


const sendBtn = document.createElement("button");

sendBtn.innerText = "✉ Αποστολή";
```

```
sendBtn.onclick = () => {  
  const user = select.value;  
  const msg = input.value.trim();  
  if (!msg) return;  
  
  const historyKey = "chatHistory_" + user;  
  const history = JSON.parse(localStorage.getItem(historyKey)) || [];  
  history.push({ sender: "Εσύ", message: msg });  
  localStorage.setItem(historyKey, JSON.stringify(history));  
  
  loadHistory(user);  
  input.value = "";  
  chatBox.scrollTop = chatBox.scrollHeight;  
};  
box.appendChild(sendBtn);  
  
const cancel = document.createElement("button");  
cancel.innerText = "Κλείσιμο";  
cancel.style.marginTop = "1rem";  
cancel.onclick = () => modal.remove();  
box.appendChild(cancel);  
  
modal.appendChild(box);  
document.body.appendChild(modal);  
}  
  
function togglePin(id) {
```

```
const note = notes.find(n => n.id === id);  
if (!note) return;  
note.isPinned = !note.isPinned;  
updateStorage();  
renderNotes();  
}  
</script>  
</body>  
</html>
```

## 8. GitHub Repository Link

<https://github.com/OnisannCong/TL>