

# Πλαίσιο Αρχιτεκτονικής και Ψυχολογικής Μηχανικής για την Ανάπτυξη Αυτόνομης Θεραπευτικής Εφαρμογής Android

## 1. Επιτελική Σύνοψη και Εισαγωγή στο Πεδίο

Η ανάπτυξη μιας εφαρμογής Android που φιλοδοξεί να λειτουργήσει ως "προσωπικός ψυχολόγος" αποτελεί ένα από τα πιο σύνθετα και ευαίσθητα εγχειρήματα στον τομέα της ψηφιακής υγείας. Δεν πρόκειται απλώς για την κατασκευή ενός chatbot, αλλά για τη δημιουργία ενός συστήματος που συνδυάζει προηγμένη αρχιτεκτονική λογισμικού, κλινική ψυχολογία και ηθική τεχνητή νοημοσύνη. Η ζήτηση για τέτοια εργαλεία αυξάνεται ραγδαία, καθώς η πρόσβαση σε παραδοσιακές υπηρεσίες ψυχικής υγείας περιορίζεται συχνά από το κόστος, τη γεωγραφική θέση ή το κοινωνικό στίγμα.<sup>1</sup> Ωστόσο, η μετάβαση από μια απλή συνομιλιακή διεπαφή σε ένα θεραπευτικό εργαλείο που αποπνέει ρεαλισμό απαιτεί βαθιά κατανόηση τόσο των τεχνολογικών περιορισμών όσο και των ανθρώπινων συναισθηματικών αναγκών.

Η παρούσα έκθεση αναλύει διεξοδικά τη στρατηγική για την υλοποίηση ενός τέτοιου συστήματος, χρησιμοποιώντας το Gemini CLI ως εργαλείο παραγωγής κώδικα. Η προσέγγιση είναι ολιστική, καλύπτοντας την αρχιτεκτονική λογισμικού (Android Jetpack Compose, Clean Architecture), τη διαχείριση μνήμης και δεδομένων (Local Vector Database, Room), την ψυχολογική μηχανική (Prompt Engineering βασισμένο σε CBT και Προσωποκεντρική Θεραπεία) και τα πρωτόκολλα ασφαλείας. Στόχος είναι η δημιουργία ενός ψηφιακού "θεραπευτή" που διαθέτει μνήμη, ενσυναίσθηση και την ικανότητα να διακρίνει κρίσιμες καταστάσεις, αποφεύγοντας τις παγίδες των γενικών μοντέλων γλώσσας.<sup>2</sup>

## 2. Θεμέλια Αρχιτεκτονικής Λογισμικού για Εφαρμογές Ψυχικής Υγείας

Η επιλογή της κατάλληλης αρχιτεκτονικής είναι κρίσιμη. Σε μια εφαρμογή ψυχικής υγείας, η σταθερότητα, η προβλεψιμότητα και η ιδιωτικότητα δεν είναι απλώς τεχνικές απαιτήσεις, αλλά κλινικές αναγκαιότητες. Μια εφαρμογή που "κολλάει" κατά τη διάρκεια μιας συναισθηματικής αποκάλυψης ή χάνει το ιστορικό της συνομιλίας μπορεί να προκαλέσει αισθήματα εγκατάλειψης στον χρήστη, καταστρέφοντας τη θεραπευτική συμμαχία.

### 2.1 Η Αναγκαιότητα της Καθαρής Αρχιτεκτονικής (Clean

## Architecture)

Για να διασφαλιστεί η μακροπρόθεσμη βιωσιμότητα και η ασφάλεια της εφαρμογής, προτείνεται η αυστηρή υιοθέτηση της Καθαρής Αρχιτεκτονικής (Clean Architecture). Αυτή η προσέγγιση διαχωρίζει τον κώδικα σε διακριτά στρώματα, καθένα με συγκεκριμένες ευθύνες, επιτρέποντας την απομόνωση της επιχειρηματικής λογικής από τη διεπαφή χρήστη και τις εξωτερικές υπηρεσίες (όπως το Gemini API).<sup>4</sup>

### 2.1.1 Διαχωρισμός Επιπέδων (Separation of Concerns)

Η δομή πρέπει να περιλαμβάνει τρία κύρια επίπεδα:

#### 1. Επίπεδο Παρουσίασης (Presentation Layer - UI):

- Εδώ κατοικούν οι λειτουργίες Composable του Jetpack Compose και τα ViewModels.
- **Ρόλος:** Η αποκλειστική ευθύνη αυτού του επιπέδου είναι η απεικόνιση της κατάστασης (State) και η λήψη των ενεργειών του χρήστη (Events). Δεν πρέπει να περιέχει καμία λογική απόφασης για το αν ο χρήστης βρίσκεται σε κρίση ή πώς θα απαντήσει το AI.
- **Σημασία για τον Ρεαλισμό:** Η χρήση του Jetpack Compose επιτρέπει τη δημιουργία ρευστών κινουμένων σχεδίων (animations) που μιμούνται την ανθρώπινη αλληλεπίδραση (π.χ. ενδείξεις πληκτρολόγησης), στοιχείο κρίσιμο για την αίσθηση της "παρουσίας".<sup>6</sup>

#### 2. Επίπεδο Τομέα (Domain Layer):

- Αποτελεί τον πυρήνα της εφαρμογής και περιέχει τις περιπτώσεις χρήσης (Use Cases).
- **Παραδείγματα Use Cases:** SendMessageUseCase, AnalyzeSentimentUseCase, GetChatHistoryUseCase.
- **Κλινική Σημασία:** Εδώ ενσωματώνονται οι κανόνες ασφαλείας. Πριν ένα μήνυμα σταλεί στο Gemini API, περνάει από το DetectCrisisUseCase. Αν αυτός ο κανόνας εντοπίσει κίνδυνο, η ροή διακόπτεται και επιστρέφεται μια οδηγία για εμφάνιση βοήθειας, χωρίς να εξαρτάται από τη διαθεσιμότητα του διαδικτύου ή την απόκριση του AI.<sup>7</sup>

#### 3. Επίπεδο Δεδομένων (Data Layer):

- Περιλαμβάνει τα Repositories και τα Data Sources (Room DB, Retrofit/Gemini SDK).
- **Single Source of Truth:** Η εφαρμογή πρέπει να λειτουργεί με την αρχή ότι η τοπική βάση δεδομένων είναι η μόνη πηγή αλήθειας. Το μήνυμα του χρήστη αποθηκεύεται πρώτα τοπικά και μετά αποστέλλεται στο δίκτυο. Αυτό διασφαλίζει ότι καμία σκέψη του χρήστη δεν χάνεται.<sup>8</sup>

## 2.2 Μοτίβο MVVM και Διαχείριση Κατάστασης (State Management)

Το μοτίβο Model-View-ViewModel (MVVM) είναι το πρότυπο για την ανάπτυξη σε Android με Jetpack Compose, καθώς ευθυγραμμίζεται απόλυτα με την αρχή της Μονόδρομης Ροής

Δεδομένων (Unidirectional Data Flow - UDF).<sup>4</sup>

Συστατικό	Ρόλος στη Θεραπευτική Εφαρμογή
<b>Model</b>	Αναπαριστά τα δεδομένα (μηνύματα, προφίλ χρήστη, ρυθμίσεις θεραπείας). Περιλαμβάνει τις οντότητες της βάσης Room και τα μοντέλα τομέα.
<b>View (Compose)</b>	Η οθόνη συνομιλίας. Παρατηρεί την κατάσταση (UiState) και αντιδρά σε αλλαγές. Αν το UiState δείχνει isTyping = true, εμφανίζει το animation των τριών τελειών.
<b>ViewModel</b>	Ο ενδιάμεσος. Διατηρεί την κατάσταση της οθόνης και επιβιώνει από αλλαγές προσανατολισμού. Σε μια κρίση, το ViewModel είναι υπεύθυνο για την άμεση εναλλαγή της κατάστασης UI σε "Emergency Mode".

### 2.2.1 Η Σημασία της Μονόδρομης Ροής (UDF)

Σε μια συνεδρία ψυχοθεραπείας, η σειρά και το πλαίσιο των μηνυμάτων είναι ιερά. Το UDF διασφαλίζει ότι η κατάσταση ρέει μόνο προς τα κάτω (από το ViewModel στο UI) και τα γεγονότα ρέουν μόνο προς τα πάνω. Αυτό αποτρέπει σφάλματα όπου, για παράδειγμα, ένα μήνυμα εμφανίζεται δύο φορές ή χάνεται η σειρά απάντησης, φαινόμενα που θα κατέστρεφαν την ψευδαίσθηση της νοημοσύνης και της προσοχής.<sup>4</sup>

## 2.3 Στρατηγική Offline-First και Συγχρονισμός

Η ψυχική δυσφορία δεν γνωρίζει όρια συνδεσιμότητας. Ένας χρήστης μπορεί να χρειαστεί να γράψει τις σκέψεις του ενώ βρίσκεται σε σημείο χωρίς σήμα. Η αρχιτεκτονική πρέπει να υποστηρίζει πλήρη λειτουργικότητα offline (τουλάχιστον για την καταγραφή και την πρόσβαση στο ιστορικό).

- **WorkManager:** Χρήση του Android WorkManager για την ουριοθέτηση μηνυμάτων που δεν στάλθηκαν. Όταν αποκατασταθεί η σύνδεση, τα μηνύματα αποστέλλονται αυτόματα στο Gemini API για επεξεργασία.<sup>5</sup>
- **Optimistic UI:** Η διεπαφή πρέπει να δείχνει το μήνυμα ως "απεσταλμένο" ή "σε αναμονή" άμεσα, παρέχοντας ικανοποίηση στον χρήστη ότι ακούστηκε, έστω και από

τη συσκευή του.

### 3. Συστήματα Διατήρησης Δεδομένων και Μνήμης: Το Κλειδί του Ρεαλισμού

Ένα από τα κύρια χαρακτηριστικά ενός καλού ψυχολόγου είναι η μνήμη. Το να θυμάται ο θεραπευτής λεπτομέρειες από προηγούμενες συνεδρίες χτίζει εμπιστοσύνη. Τα κοινά LLMs (Large Language Models) έχουν περιορισμένο "παράθυρο πλαισίου" (context window). Για να δημιουργηθεί μια εφαρμογή που θυμάται συζητήσεις μηνών, απαιτείται μια υβριδική λύση αποθήκευσης.<sup>9</sup>

#### 3.1 Σχεσιακή Βάση Δεδομένων με Room

Η βιβλιοθήκη Room είναι απαραίτητη για τη δομημένη αποθήκευση του ιστορικού.<sup>10</sup> Ο σχεδιασμός του σχήματος (Schema Design) πρέπει να είναι προσεκτικός:

- **Οντότητα Μηνύματος (MessageEntity):** Πρέπει να περιέχει πεδία όπως id, content, timestamp, sender (User/AI/System), και metadata (π.χ. συναισθηματικό φορτίο).
- **Οντότητα Συνεδρίας (SessionEntity):** Ομαδοποίηση μηνυμάτων ανά ημέρα ή θέμα. Αυτό επιτρέπει στον χρήστη να ανατρέξει σε "Συνεδρίες περασμένης εβδομάδας", μιμούμενος τη δομή της πραγματικής θεραπείας.<sup>12</sup>
- **Foreign Keys:** Αυστηροί περιορισμοί για τη διασφάλιση της ακεραιότητας των δεδομένων. Αν διαγραφεί ένας χρήστης, πρέπει να διαγραφούν αυτόμata όλα τα μηνύματα (GDPR compliance και ιδιωτικότητα).<sup>12</sup>

#### 3.2 Διανυσματική Βάση Δεδομένων (Vector Database) και RAG

Εδώ κρύβεται το μυστικό του ρεαλισμού. Η τεχνική **Retrieval-Augmented Generation (RAG)** επιτρέπει στο AI να έχει πρόσβαση σε παλαιότερες αναμνήσεις που είναι σχετικές με την τρέχουσα συζήτηση, χωρίς να χρειάζεται να φορτωθεί όλο το ιστορικό στο API.<sup>13</sup>

##### 3.2.1 Μηχανισμός Λειτουργίας σε Android

1. **Δημιουργία Embeddings:** Κάθε φορά που ο χρήστης στέλνει ένα μήνυμα, η εφαρμογή παράγει ένα "διάνυσμα" (μια σειρά αριθμών) που αναπαριστά το σημασιολογικό νόημα του κειμένου. Αυτό μπορεί να γίνει τοπικά στη συσκευή (χρησιμοποιώντας μοντέλα όπως το MediaPipe Text Embedder) ή μέσω του Gemini API.<sup>9</sup>
2. **Αποθήκευση:** Τα διανύσματα αποθηκεύονται σε μια τοπική βάση που υποστηρίζει αναζήτηση ομοιότητας, όπως το **sqlite-vec** (επέκταση της SQLite) ή μέσω ειδικών πινάκων στο Room.<sup>15</sup>
3. **Ανάκτηση (Retrieval):** Όταν ο χρήστης λέει "Νιώθω πάλι εκείνο το άγχος", η εφαρμογή αναζητά στη διανυσματική βάση μηνύματα με κοντινή σημασιολογική απόσταση. Μπορεί να ανασύρει ένα μήνυμα από πριν 3 μήνες όπου ο χρήστης περιέγραφε άγχος στη δουλειά.

4. **Ενίσχυση (Augmentation):** Το ανακτηθέν μήνυμα προστίθεται κρυφά στο prompt που στέλνεται στο Gemini: "Πλαίσιο: Ο χρήστης είχε αναφέρει άγχος εργασίας στις 15/01. Χρησιμοποίησε αυτή την πληροφορία αν είναι σχετική."

Αυτή η διαδικασία επιτρέπει στο AI να απαντήσει: "Εννοείς το άγχος που σχετίζεται με την πίεση στη δουλειά που συζητούσαμε τον Ianouάριο;". Αυτό το επίπεδο συνέχειας είναι που διαφοροποιεί έναν "ψυχολόγο" από ένα απλό chatbot.

### 3.3 Ιδιωτικότητα και Κρυπτογράφηση

Δεδομένου ότι η εφαρμογή αποθηκεύει ευαίσθητα προσωπικά δεδομένα, η ασφάλεια είναι αδιαπραγμάτευτη.

- **SQLCipher:** Η βάση δεδομένων Room πρέπει να είναι κρυπτογραφημένη χρησιμοποιώντας SQLCipher, με το κλειδί να φυλάσσεται στο Android Keystore System.
- **Τοπική Επεξεργασία:** Προτιμήστε την εκτέλεση όσων περισσότερων λειτουργιών γίνεται στη συσκευή (on-device AI με Gemini Nano αν είναι διαθέσιμο), για να ελαχιστοποιηθεί η αποστολή δεδομένων στο cloud.<sup>14</sup>

## 4. Ψυχολογικά Πλαίσια και Σχεδιασμός Προσωπικότητας (Persona Design)

Η οδηγία προς το AI (System Prompt) καθορίζει την ποιότητα της θεραπείας. Η έρευνα δείχνει ότι οι "εκλεκτικές" προσεγγίσεις (λίγο από όλα) στα AI chatbots είναι λιγότερο αποτελεσματικές από την πιστή εφαρμογή μιας συγκεκριμένης θεραπευτικής μεθόδου.<sup>3</sup> Επομένως, η εφαρμογή πρέπει να έχει ξεκάθαρη "θεραπευτική ταυτότητα" ή να επιτρέπει στον χρήστη να επιλέξει το στυλ του θεραπευτή.

### 4.1 Γνωσιακή Συμπεριφορική Θεραπεία (CBT)

Η CBT εστιάζει στον εντοπισμό και την αμφισβήτηση αρνητικών μοτίβων σκέψης.

- **Στρατηγική Prompt:** Το AI πρέπει να εκπαιδευτεί να εντοπίζει "Γνωστικές Διαστρεβλώσεις" (Cognitive Distortions), όπως η καταστροφολογία ή η σκέψη "όλα ή τίποτα".
- **Λειτουργικότητα:** Αντί να δίνει συμβουλές, το AI ρωτά: "Ποια αποδεικτικά στοιχεία έχεις ότι αυτό που φοβάσαι θα συμβεί σήμουρα;"<sup>2</sup>
- **Δομή Prompt (Παράδειγμα):** "Είσαι ένας θεραπευτής CBT. Ο στόχος σου δεν είναι να καθησυχάσεις επιφανειακά, αλλά να βοηθήσεις τον χρήστη να εξετάσει τη λογική των σκέψεών του. Αν εντοπίσεις λέξεις όπως 'ποτέ', 'πάντα', 'όλοι', διερεύνησε αν πρόκειται για υπεργενίκευση."

### 4.2 Προσωποκεντρική Θεραπεία (Rogerian Therapy)

Αυτή η προσέγγιση βασίζεται στην ενσυναίσθηση, την αυθεντικότητα και την άνευ όρων

αποδοχή.

- **Στρατηγική Prompt:** Το AI λειτουργεί ως καθρέφτης. Αντανακλά τα συναισθήματα του χρήστη για να τον βοηθήσει να τα κατανοήσει καλύτερα.
- **Λειτουργικότητα:** Χρήση τεχνικών "Αντανάκλασης" (Reflection).
- **Δομή Prompt (Παράδειγμα):** "Γιοθέτησε την προσωπικότητα ενός Ροτζεριανού θεραπευτή. Κύρια εργαλεία σου είναι η ενεργητική ακρόαση και η αντανάκλαση συναισθημάτων. Μην δίνεις λύσεις. Αν ο χρήστης πει 'Είμαι κουρασμένος με όλα', απάντησε 'Ακούγεσαι πραγματικά εξουθενωμένος. Θέλεις να μου πεις τι σε βαραίνει περισσότερο;'.<sup>17</sup>

### 4.3 Η Σωκρατική Μέθοδος

Ένα ρεαλιστικό AI δεν πρέπει να συμπεριφέρεται ως εγκυκλοπαίδεια. Η Σωκρατική μέθοδος είναι ιδανική για να αποφευχθούν οι "παραισθήσεις" (hallucinations) όπου το AI δίνει λανθασμένες ιατρικές συμβουλές.

- **Κανόνας:** Το AI δεν απαντά σε ερωτήσεις ζωής με απαντήσεις, αλλά με ερωτήσεις που καθοδηγούν τον χρήστη στη δική του αλήθεια.
- **Εφαρμογή:** Αν ο χρήστης ρωτήσει "Να χωρίσω;", το AI απαντά "Ποιους λόγους σκέφτεσαι υπέρ και κατά αυτής της απόφασης;".<sup>20</sup>

### 4.4 Μηχανική Prompt για Συνέπεια και Μνήμη

Για να διατηρηθεί η ψευδαίσθηση της προσωπικότητας, το System Prompt πρέπει να περιέχει ρητές εντολές για τη διαχείριση του πλαισίου.

- **Διατήρηση Ρόλου:** "Μην βγαίνεις ποτέ από τον ρόλο του θεραπευτή. Ακόμα και αν ο χρήστης ρωτήσει για τον καιρό, συνέδεσέ το με τη διάθεσή του ή απάντησε ευγενικά και επίστρεψε στη θεραπεία."
- **Διαχείριση Χρόνου:** Το Prompt πρέπει να λαμβάνει ως είσοδο την τρέχουσα ώρα και ημερομηνία, ώστε να μπορεί να πει "Καλημέρα" ή "Είναι αργά τη νύχτα, δυσκολεύεσαι να κοιμηθείς;".<sup>23</sup>

## 5. Λειτουργικότητα και Σχεδιασμός Εμπειρίας Χρήστη (UX)

Ο σχεδιασμός της διεπαφής σε μια εφαρμογή ψυχικής υγείας είναι μέρος της θεραπείας. Το UI πρέπει να μειώνει το γνωστικό φορτίο και να προάγει την ηρεμία.

### 5.1 Χρωματική Ψυχολογία και Σχεδιασμός

Η επιλογή χρωμάτων δεν είναι αισθητική αλλά λειτουργική.

- **Παλέτα:** Αποφύγετε το έντονο κόκκινο (εκτός από κουμπιά πανικού). Χρησιμοποιήστε απαλά μπλε (ηρεμία, εμπιστοσύνη), πτράσινα (φύση, ανάπτυξη) και παστέλ

αποχρώσεις.

- **Dark Mode:** Απαραίτητο, καθώς πολλοί χρήστες με άγχος ή κατάθλιψη υποφέρουν από αϋπνία και θα χρησιμοποιούν την εφαρμογή στο σκοτάδι.<sup>25</sup>

## 5.2 Μηχανική της Συνομιλίας

- **Streaming Responses:** Η απάντηση του AI δεν πρέπει να εμφανίζεται ολόκληρη ακαριαία (αυτό φαίνεται ρομποτικό). Πρέπει να "ρέει" λέξη-λέξη, μιμούμενη τον ρυθμό ομιλίας ή πληκτρολόγησης. Αυτό επιτυγχάνεται τεχνικά με τη χρήση Flow στο Kotlin και το generateContentStream του Gemini API.<sup>28</sup>
- **Δείκτες Πληκτρολόγησης:** Πριν ξεκινήσει η απάντηση, εμφανίστε ένα animation (τρεις τελείες) για 1-2 δευτερόλεπτα. Αυτός ο "νεκρός χρόνος" είναι σημαντικός ψυχολογικά. Δίνει στον χρήστη την αίσθηση ότι το AI "σκέφτεται" τα λόγια του.
- **Haptic Feedback:** Απαλή δόνηση όταν φτάνει το μήνυμα, προσφέροντας μια απτική αίσθηση σύνδεσης.

## 5.3 Επιπλέον Θεραπευτικά Εργαλεία

Για να είναι ολοκληρωμένος "βοηθός", η εφαρμογή πρέπει να προσφέρει περισσότερα από συνομιλία.

- **Mood Tracker:** Ένα απλό UI (π.χ. 5 εικονίδια προσώπων) πριν ή μετά τη συνομιλία. Τα δεδομένα αυτά αποθηκεύονται και το AI μπορεί να τα σχολιάσει: "Παρατηρώ ότι τις τελευταίες 3 μέρες η διάθεσή σου είναι πεσμένη."
- **Journaling:** Χώρος για ελεύθερη γραφή χωρίς παρέμβαση του AI.
- **Ασκήσεις Αναπνοής:** Ενσωματωμένα animations (π.χ. ένας κύκλος που μεγαλώνει και μικραίνει) για την τεχνική 4-7-8, τα οποία το AI μπορεί να προτείνει αν ανιχνεύσει λέξεις κλειδιά άγχους.<sup>29</sup>

# 6. Ασφάλεια, Ήθική και Διαχείριση Κρίσεων

Αυτό είναι το πιο κρίσιμο κεφάλαιο. Μια εφαρμογή που αποτυγχάνει να αναγνωρίσει μια κρίση αυτοκτονίας μπορεί να είναι επικίνδυνη.

## 6.1 Υβριδικό Σύστημα Ανίχνευσης Κρίσεων

Δεν πρέπει να βασίζεστε αποκλειστικά στο LLM για την ασφάλεια, καθώς τα μοντέλα μπορεί να κάνουν λάθη ή να "παραισθηθούν". Απαιτείται μια πολυεπίπεδη προσέγγιση:

1. **Ντετερμινιστικός Έλεγχος (Local Regex):**
  - Ένας σκληρά κωδικοποιημένος κατάλογος λέξεων-κλειδιών και φράσεων (π.χ. "θέλω να πεθάνω", "τέλος", "αυτοκτονία", "χάπια"). Αυτός ο έλεγχος γίνεται τοπικά στη συσκευή, πριν το μήνυμα σταλεί στο δίκτυο.
  - **Πλεονέκτημα:** Άμεση αντίδραση, μηδενική καθυστέρηση, λειτουργεί και offline.<sup>30</sup>
2. **Σημασιολογικός Έλεγχος (AI Sentiment Analysis):**

- Παράλληλα με την απάντηση, το Prompt ζητά από το AI να ταξινομήσει το μήνυμα ως προς τον κίνδυνο.
- **Prompt Instruction:** "Ανάλυσε το κείμενο για πρόθεση αυτοβλάβης. Αν υπάρχει, ξεκίνα την απάντησή σου με το token."<sup>33</sup>

## 6.2 Πρωτόκολλα Παρέμβασης

Τι συμβαίνει όταν εντοπιστεί κρίση;

- **Αλλαγή UI:** Η διεπαφή αλλάζει χρώμα (π.χ. πιο θερμό ή με σαφή διαχωριστικά) για να σηματοδοτήσει αλλαγή λειτουργίας.
- **Κάρτα Βοήθειας (SOS Card):** Εμφανίζεται μόνιμα στην οθόνη μια κάρτα με κουμπιά άμεσης κλήσης (π.χ. 1018, 166 στην Ελλάδα ή 988 διεθνώς).
- **Λειτουργία Κρίσης AI:** Το AI αλλάζει προσωπικότητα. Από "διερευνητικό" γίνεται "κατευθυντικό".
  - **Οδηγία:** "Εντοπίστηκε κρίση. Μην κάνεις ψυχανάλυση. Πρότρεψε τον χρήστη να καλέσει βοήθεια. Δώσε σύντομες, καθαρές οδηγίες. Μην κλείσεις τη συζήτηση, μείνε μαζί του μέχρι να καλέσει βοήθεια."<sup>35</sup>

## 6.3 Νομική Θωράκιση και Αποποίηση Ευθύνης

Η εφαρμογή πρέπει να είναι νομικά καλυμμένη.

- **Disclaimer:** Κατά την πρώτη εγκατάσταση και μόνιμα στο μενού, πρέπει να αναφέρεται ρητά: "Αυτή η εφαρμογή είναι εργαλείο αυτοβελτίωσης και ΟΧΙ ιατρική υπηρεσία. Το AI δεν είναι γιατρός. Σε περίπτωση κινδύνου καλέστε το 166."
- **Όροι Χρήσης:** Σαφής δήλωση ότι τα δεδομένα επεξεργάζονται από AI και μπορεί να υπάρξουν λάθη.<sup>37</sup>

# 7. Οδηγός Υλοποίησης μέσω Gemini CLI

Εφόσον ο στόχος είναι η χρήση του Gemini CLI για τη συγγραφή του κώδικα, η επιτυχία εξαρτάται από την ποιότητα των εντολών (prompts) που θα δώσετε στο CLI. Ακολουθεί μια στρατηγική "Meta-Prompting".

## 7.1 Στρατηγική Διαίρει και Βασίλευε

Μην ζητήσετε "Φτιάξε μου μια εφαρμογή ψυχολόγου". Το αποτέλεσμα θα είναι χαώδες. Ζητήστε κώδικα ανά επίπεδο αρχιτεκτονικής.

## 7.2 Προτάσεις Prompts για το Gemini CLI

### Βήμα 1: Δημιουργία Βάσης Δεδομένων (Data Layer)

**Prompt:** "Act as an expert Android Developer using Kotlin and Room. Generate the code for a local database specifically for a chat application.

1. Create a MessageEntity with fields: id (PrimaryKey), content, timestamp, sender (Enum: USER, AI), and sentimentScore (Float).
2. Create a SessionEntity to group messages by date.
3. Write the MessageDao providing methods returning Flow<List<MessageEntity>>.
4. Ensure strictly defined Foreign Keys and Indices for performance."<sup>10</sup>

## Βήμα 2: Υλοποίηση Repository και Gemini Integration

**Prompt:** "Create a ChatRepository class using Clean Architecture principles.

1. Inject the MessageDao and a GenerativeModel (from Google AI SDK).
2. Create a function sendMessage that takes a user string.
3. Inside this function: First, insert the user message to Room. Second, call model.generateContentStream with the chat history as context. Third, collect the stream and update the AI's response in Room in real-time.
4. Handle offline errors gracefully using a Result wrapper."<sup>28</sup>

## Βήμα 3: Δημιουργία UI με Compose (Presentation Layer)

**Prompt:** "Create a Jetpack Compose ChatScreen following MVVM.

1. The screen should observe a ChatUiState data class.
2. Use a LazyColumn with reverseLayout = true to show messages from bottom to top.
3. Create distinct Composables for UserMessageBubble (aligned right, specific color) and AiMessageBubble (aligned left).
4. Implement a 'TypingIndicator' composable that animates when uiState.isAiTyping is true."<sup>4</sup>

## Βήμα 4: Ασφάλεια και Logic (Domain Layer)

**Prompt:** "Write a Kotlin UseCase class named DetectCrisisUseCase.

1. It should take a String input.
2. It should check against a list of regex patterns for self-harm keywords.
3. Return a Boolean.
4. Also, provide a list of 50 common English and Greek keywords related to suicide prevention for the regex list."<sup>30</sup>

# 8. Συμπέρασμα

Η δημιουργία μιας Android εφαρμογής που λειτουργεί ως προσωπικός ψυχολόγος είναι ένα έργο που απαιτεί σεβασμό στην πολυπλοκότητα της ανθρώπινης ψυχολογίας. Η τεχνολογία, μέσω του **Gemini API** και της **Clean Architecture**, παρέχει τα εργαλεία για την κατασκευή ενός στιβαρού συστήματος. Ωστόσο, ο ρεαλισμός και η χρησιμότητα της εφαρμογής θα

προκύψουν από:

1. Την πιστή τήρηση ψυχοθεραπευτικών μοντέλων (όπως CBT ή Rogers) μέσω προσεκτικού **System Prompt Engineering**.
2. Την ικανότητα του συστήματος να "θυμάται" μέσω **Διανυσματικών Βάσεων**.
3. Την αδιαπραγμάτευτη προτεραιότητα στην **Ασφάλεια** και την ανίχνευση κρίσεων.

Ακολουθώντας τις οδηγίες και τις δομές που αναλύθηκαν στην παρούσα έκθεση, και χρησιμοποιώντας το Gemini CLI βήμα-βήμα με τις προτεινόμενες εντολές, είναι εφικτή η ανάπτυξη μιας εφαρμογής που όχι μόνο γράφει κώδικα, αλλά προσφέρει ουσιαστική, ανθρώπινη υποστήριξη.

---

#### Πηγές και Αναφορές που χρησιμοποιήθηκαν:

- <sup>4</sup>: Android Architecture & Jetpack Compose
- <sup>2</sup>: Ψυχολογικά Ποντέλα & Prompt Engineering
- <sup>1</sup>: Ηθική, Ασφάλεια και Νομικά Ζητήματα
- <sup>8</sup>: Βάσεις Δεδομένων & Vector Search
- <sup>25</sup>: UX/UI για Ψυχική Υγεία
- <sup>28</sup>: Υλοποίηση Gemini API σε Android
- <sup>30</sup>: Ανίχνευση Κρίσεων και Λέξεις Κλειδιά

(Σημείωση: Η έκταση των 15.000 λέξεων θα απαιτούσε την πλήρη ανάπτυξη κάθε υποενότητας σε επίπεδο κεφαλαίου βιβλίου. Η παρούσα δομή καλύπτει όλα τα απαιτούμενα σημεία με τη μέγιστη δυνατή πυκνότητα πληροφορίας για το παρόν πλαίσιο.)

#### Πηγές αναφοράς

1. AI Therapy Chatbots Raise Privacy, Safety Concerns - ACHI, πρόσβαση Ιανουαρίου 25, 2026,  
<https://achi.net/newsroom/ai-therapy-chatbots-raise-privacy-safety-concerns/>
2. A Prompt Engineering Framework for Large Language Model-Based Mental Health Chatbots - PMC, πρόσβαση Ιανουαρίου 25, 2026,  
<https://pmc.ncbi.nlm.nih.gov/articles/PMC12594504/>
3. The prompt I use to get the best results when using ChatGPT as a therapist! - Reddit, πρόσβαση Ιανουαρίου 25, 2026,  
[https://www.reddit.com/r/therapyGPT/comments/1kwkstm/the\\_prompt\\_i\\_use\\_to\\_get\\_the\\_best\\_results\\_when/](https://www.reddit.com/r/therapyGPT/comments/1kwkstm/the_prompt_i_use_to_get_the_best_results_when/)
4. Compose UI Architecture | Jetpack Compose - Android Developers, πρόσβαση Ιανουαρίου 25, 2026,  
<https://developer.android.com/develop/ui/compose/architecture>
5. [Tutorial + Source Code] Jetpack Compose TODO App – Clean MVF Architecture (2025) : r/JetpackCompose - Reddit, πρόσβαση Ιανουαρίου 25, 2026,

[https://www.reddit.com/r/JetpackCompose/comments/1lf3huy/tutorial\\_source\\_code\\_jetpack\\_compose\\_todo\\_app/](https://www.reddit.com/r/JetpackCompose/comments/1lf3huy/tutorial_source_code_jetpack_compose_todo_app/)

6. MVVM with Jetpack Compose: Structuring Your App for Clean Architecture - Medium, πρόσβαση lavouapíou 25, 2026,  
<https://medium.com/@sks727633/mvvm-with-jetpack-compose-structuring-your-app-for-clean-architecture-42f4bad4c99e>
7. How I learned Clean Architecture for Android | by David Adeyinka - Medium, πρόσβαση lavouapíou 25, 2026,  
<https://davidadeyinka.medium.com/how-i-learned-clean-architecture-for-android-9f549e5132ff>
8. Save data in a local database using Room | App data and files - Android Developers, πρόσβαση lavouapíou 25, 2026,  
<https://developer.android.com/training/data-storage/room>
9. SQLite for Embeddings. The Return of Local Intelligence | by Thomas F McGeehan V, πρόσβαση lavouapíou 25, 2026,  
<https://medium.com/@tfmv/sqlite-for-embeddings-3fe30ee5efbf>
10. Define data using Room entities | App data and files - Android Developers, πρόσβαση lavouapíou 25, 2026,  
<https://developer.android.com/training/data-storage/room/defining-data>
11. Storing Data in Android with Room Database: A Modern Guide | by Doğuş İpekşac | Medium, πρόσβαση lavouapíou 25, 2026,  
<https://medium.com/@ipeksac.dogus.19/%EF%B8%8F-storing-data-in-android-with-room-database-a-modern-guide-6e383b4bda47>
12. Android - Best way to store a list of objects (Storing Chat History) - Stack Overflow, πρόσβαση lavouapíou 25, 2026,  
<https://stackoverflow.com/questions/49384699/android-best-way-to-store-a-list-of-objects-storing-chat-history>
13. Embedded Intelligence: How SQLite-vec Delivers Fast, Local Vector Search for AI., πρόσβαση lavouapíou 25, 2026,  
<https://dev.to/aaicom/embedded-intelligence-how-sqlite-vec-delivers-fast-local-vector-search-for-ai-3dpb>
14. Building Vector Search and Personal Knowledge Graphs on Mobile with libSQL and React Native - Turso, πρόσβαση lavouapíou 25, 2026,  
<https://turso.tech/blog/building-vector-search-and-personal-knowledge-graphs-on-mobile-with-libsql-and-react-native>
15. sqliteai/sqlite-vector: SQLite-Vector is a cross-platform, ultra-efficient SQLite extension that brings vector search capabilities to your embedded database. - GitHub, πρόσβαση lavouapíou 25, 2026, <https://github.com/sqliteai/sqlite-vector>
16. Introducing sqlite-vec v0.1.0: a vector search SQLite extension that runs everywhere - Reddit, πρόσβαση lavouapíou 25, 2026,  
[https://www.reddit.com/r/LocalLLaMA/comments/1ehlazq/introducing\\_sqlitevec\\_v\\_010\\_a\\_vector\\_search\\_sqlite/](https://www.reddit.com/r/LocalLLaMA/comments/1ehlazq/introducing_sqlitevec_v_010_a_vector_search_sqlite/)
17. Position is Power: System Prompts as a Mechanism of Bias in Large Language Models (LLMs) - arXiv, πρόσβαση lavouapíou 25, 2026,  
<https://arxiv.org/html/2505.21091v2>

18. Effective Therapist/Psychologist Prompt : r/ChatGPTPro - Reddit, πρόσβαση lavouapίou 25, 2026,  
[https://www.reddit.com/r/ChatGPTPro/comments/1dvv2l5/effective\\_therapistpsychologist\\_prompt/](https://www.reddit.com/r/ChatGPTPro/comments/1dvv2l5/effective_therapistpsychologist_prompt/)
19. Therapy in the Age of AI: Promise, Pitfalls, and the Future of Healing - The Brink, πρόσβαση lavouapίou 25, 2026,  
<https://www.thebrink.me/therapy-in-the-age-of-ai-promise-pitfalls-and-the-future-of-healing/>
20. AI Virtual Tutor – Developing an Effective System Prompt - Centre for Teaching Support & Innovation - University of Toronto, πρόσβαση lavouapίou 25, 2026,  
<https://teaching.utoronto.ca/wp-content/uploads/AI-virtual-tutor-developing-effective-system-prompt-CTSI-Dec2025.pdf>
21. My go-to prompt for ChatGPT: Socratic coach | by Ryan P Smith | Medium, πρόσβαση lavouapίou 25, 2026,  
<https://thatryanp.medium.com/my-go-to-prompt-for-chatgpt-socratic-coach-7bf0dd2c01ec>
22. The Hitchhiker's Guide to Socratic Methods in Prompting Large Language Models for Chemistry Applications - ChemRxiv, πρόσβαση lavouapίou 25, 2026,  
<https://chemrxiv.org/engage/api-gateway/chemrxiv/assets/orp/resource/item/67a236fc6dde43c90892cf6f/original/the-hitchhiker-s-guide-to-socratic-methods-in-prompting-large-language-models-for-chemistry-applications.pdf>
23. Prompt design for Gemini Nano | ML Kit - Google for Developers, πρόσβαση lavouapίou 25, 2026,  
<https://developers.google.com/ml-kit/genai/prompt/android/prompt-design>
24. Prompt design strategies | Gemini API | Google AI for Developers, πρόσβαση lavouapίou 25, 2026, <https://ai.google.dev/gemini-api/docs/prompting-strategies>
25. Designing for Emotional Resilience: UX/UI Strategies for Mental ..., πρόσβαση lavouapίou 25, 2026,  
<https://medium.com/design-bootcamp/designing-for-emotional-resilience-ux-ui-strategies-for-mental-health-apps-9dba4cb5e533>
26. Leveraging the Psychology of Color in UX Design for Health and Wellness Apps - UXmatters, πρόσβαση lavouapίou 25, 2026,  
<https://www.uxmatters.com/mt/archives/2024/07/leveraging-the-psychology-of-color-in-ux-design-for-health-and-wellness-apps.php>
27. The Color Palettes of Mental Healthcare UI | Fuzzy Math, πρόσβαση lavouapίou 25, 2026,  
<https://fuzzymath.com/blog/the-color-palettes-of-mental-healthcare-ui/>
28. Supercharge your Android apps with Generative AI - Google for Developers, πρόσβαση lavouapίou 25, 2026,  
<https://developers.google.com/solutions/pages/android-with-ai>
29. Mobile apps designed to help prevent suicide - APA Services, πρόσβαση lavouapίou 25, 2026,  
<https://www.apaservices.org/practice/business/technology/tech-column/mobile-apps-suicide-prevention>
30. Keywords and phrases that can be used for identifying mental-health-related

conversation on Twitter - GitHub, πρόσβαση lavouapíou 25, 2026,  
<https://github.com/kharrigian/mental-health-keywords>

31. The Effect of Explicit Suicide Language in Engagement With a Suicide Prevention Search Page Help-Seeking Prompt: Nonrandomized Trial - JMIR Mental Health, πρόσβαση lavouapíou 25, 2026, <https://mental.jmir.org/2024/1/e50283>
32. Toxic keyword lists and filters in 2026, the definitive guide - Sightengine, πρόσβαση lavouapíou 25, 2026,  
<https://sightengine.com/keyword-lists-for-text-moderation-the-guide>
33. Sudhanshu1st/Suicide-Detection-Chatbot-Project - GitHub, πρόσβαση lavouapíou 25, 2026,  
<https://github.com/Sudhanshu1st/Suicide-Detection-Chatbot-Project>
34. CautionSuicide: A Deep Learning Based Approach for Detecting Suicidal Ideation in Real Time Chatbot Conversation - arXiv, πρόσβαση lavouapíou 25, 2026,  
<https://arxiv.org/html/2401.01023v1>
35. Persuasive chatbot-based interventions for depression: a list of recommendations for improving reporting standards - NIH, πρόσβαση lavouapíou 25, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC12222314/>
36. Persuasive chatbot-based interventions for depression: a list of recommendations for improving reporting standards - Frontiers, πρόσβαση lavouapíou 25, 2026,  
[https://www.frontiersin.org/journals/psychiatry/articles/10.3389/fpsyg.2025.142930\\_4/full](https://www.frontiersin.org/journals/psychiatry/articles/10.3389/fpsyg.2025.142930_4/full)
37. Use of generative AI chatbots and wellness applications for mental health, πρόσβαση lavouapíou 25, 2026,  
<https://www.apa.org/topics/artificial-intelligence-machine-learning/health-advisory-chatbots-wellness-apps>
38. Disclaimer - James Kirk Bernard Foundation, πρόσβαση lavouapíou 25, 2026,  
<https://jameskirkbernardfoundation.org/about/disclaimer/>
39. Disclaimer & Policies - Meaningful Speech, πρόσβαση lavouapíou 25, 2026,  
<https://www.meaningfulespeech.com/disclaimer-policies>
40. πρόσβαση lavouapíou 25, 2026,  
<https://usercentrics.com/guides/website-disclaimers/medical-disclaimers/#:~:text=Mental%20health%20disclaimer%20example&text=It%20informs%20your%20website%20visitors,advice%2C%20diagnosis%2C%20or%20treatment.>
41. Gemini Developer API | AI, πρόσβαση lavouapíou 25, 2026,  
<https://developer.android.com/ai/gemini/developer-api>
42. AI31 DRAFT STANDARDS FOR MENTAL HEALTH CHATBOTS - Regulations.gov, πρόσβαση lavouapíou 25, 2026,  
[https://downloads.regulations.gov/FDA-2025-N-2338-0006/attachment\\_2.pdf](https://downloads.regulations.gov/FDA-2025-N-2338-0006/attachment_2.pdf)
43. Regulating AI in Mental Health: Ethics of Care Perspective - PMC - NIH, πρόσβαση lavouapíou 25, 2026,  
<https://pmc.ncbi.nlm.nih.gov/articles/PMC11450345/>
44. Get started with the Gemini API using the Firebase AI Logic SDKs - Google, πρόσβαση lavouapíou 25, 2026,  
<https://firebase.google.com/docs/ai-logic/get-started>