



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ**  
**“ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ & ΥΠΗΡΕΣΙΕΣ”**

**Υλοποίηση εφαρμογής για κινητά τηλέφωνα για τη δημιουργία  
Ευρωπαϊκών Ενιαίων Εγγράφων Σύμβασης (ΕΕΕΣ)**

Από  
Αλέξανδρος Ζήσης  
Υποβάλλεται  
για την εκπλήρωση των προϋποθέσεων λήψης  
Μεταπτυχιακού Διπλώματος  
στην ειδίκευση «Προηγμένα Πληροφοριακά Συστήματα»  
του ΠΜΣ “Πληροφοριακά Συστήματα & Υπηρεσίες”  
στο  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
Φεβρουάριος 2024

Επιβλέπουσα: Πρέντζα Ανδριάννα

Ακαδημαϊκή Θέση: Καθηγήτρια

Πανεπιστήμιο Πειραιώς. Κάτοχος όλων των δικαιωμάτων  
University of Piraeus. All rights reserved.

Συγγραφέας / Author . . . . Αλέξανδρος Ζήσης

## ΣΕΛΙΔΑ ΕΓΚΥΡΟΤΗΤΑΣ

**Όνοματεπώνυμο Φοιτητή:** Αλέξανδρος Ζήσης

**Τίτλος Μεταπτυχιακής Διπλωματικής Εργασίας:** Υλοποίηση εφαρμογής για κινητά τηλέφωνα για τη δημιουργία Ευρωπαϊκών Ενιαίων Εγγράφων Σύμβασης (ΕΕΕΣ)

*Η παρούσα Μεταπτυχιακή Διπλωματική Εργασία υποβάλλεται ως μερική εκπλήρωση των απαιτήσεων του Προγράμματος Μεταπτυχιακών Σπουδών "Πληροφοριακά Συστήματα & Υπηρεσίες" του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς και εγκρίθηκε στις 29/2/2024 από τα μέλη της Εξεταστικής Επιτροπής.*

### Εξεταστική Επιτροπή

Επιβλέπουσα (Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς): Καθ. Ανδριάννα Πρέντζα

Μέλος Εξεταστικής Επιτροπής: Καθ. Δημοσθένης Κυριαζής

Μέλος Εξεταστικής Επιτροπής: Αναπλ. Καθ. Μαρία Χαλκίδη

### ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΑΥΘΕΝΤΙΚΟΤΗΤΑΣ

*Ο Αλέξανδρος Ζήσης, γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα ότι η παρούσα εργασία με τίτλο «Υλοποίηση εφαρμογής για κινητά τηλέφωνα για τη δημιουργία Ευρωπαϊκών Ενιαίων Εγγράφων Σύμβασης (ΕΕΕΣ)», αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές που έχω χρησιμοποιήσει, έχουν δηλωθεί κατάλληλα στις βιβλιογραφικές παραπομπές και αναφορές. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.*

*Επιπλέον δηλώνω υπεύθυνα ότι η συγκεκριμένη Μεταπτυχιακή Διπλωματική Εργασία έχει συγγραφεί από εμένα προσωπικά και δεν έχει υποβληθεί ούτε έχει αξιολογηθεί στο πλαίσιο κάποιου άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό.*

*Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου. Σε κάθε περίπτωση, αναληθούς ή ανακριβούς δηλώσεως, υπόκειμαι στις συνέπειες που προβλέπονται τις διατάξεις που προβλέπει η Ελληνική και Κοινοτική Νομοθεσία περί πνευματικής ιδιοκτησίας.*

### Ο/Η ΔΗΛΩΝ/ΟΥΣΑ

**Όνοματεπώνυμο:** Αλέξανδρος Ζήσης

**Αριθμός Μητρώου:** ΜΕ2135

**Υπογραφή:**



## Περίληψη

Η παρούσα διπλωματική εργασία στοχεύει στη δημιουργία μιας εφαρμογής για κινητά τηλέφωνα που θα εκμεταλλεύεται υπάρχον API για τη δημιουργία του Ευρωπαϊκού Ενιαίου Εγγράφου Σύμβασης (ΕΕΕΣ). Η ανάλυση ξεκινά με την αναφορά των αλλαγών και τροποποιήσεων του συστήματος που θα πρέπει να συμβούν για να υποστηρίξει τις πιο πρόσφατες προδιαγραφές όπως αυτές ορίζονται από την Ευρωπαϊκή Ένωση. Ακολούθως, εξετάζονται και καταγράφονται οι λειτουργίες και οι δυνατότητες του υπάρχοντος συστήματος. Ο γραφικός σχεδιασμός της εφαρμογής που περιλαμβάνει και την ανάλυση Εμπειρίας Χρήστη (UX) και προσδίδει στην εφαρμογή υψηλή ευχρηστία και αισθητική που ανταποκρίνονται στις σύγχρονες προσδοκίες των χρηστών αναλύεται στη συνέχεια. Ακολουθεί λεπτομερής περιγραφή της αρχιτεκτονικής, των τεχνολογιών και των κεντρικών λειτουργιών της εφαρμογής. Κλείνοντας, παρουσιάζονται προτάσεις για τα επόμενα βήματα, ολοκληρώνοντας έτσι μια εκτενή ανασκόπηση του κύκλου ζωής ανάπτυξης της εφαρμογής.

## Abstract

This thesis aims to develop a mobile application utilizing an existing API for the creation of European Single Procurement Document (ESPD) records. The analysis begins with enumerating the changes required to align the system with the latest specifications. Functionality is explored, and existing system capabilities are enumerated. The graphical design of the application follows, incorporating User Experience (UX) analysis to enhance usability and aesthetics, meeting contemporary user expectations. A detailed analysis of the architecture, technologies, and core functions of the application is provided. Finally, suggestions for future steps are discussed, concluding the comprehensive exploration of the application development lifecycle.

**Θεματική περιοχή:** Διαλειτουργικότητα στις Ηλεκτρονικές Προμήθειες

**Keywords:** ESPD, ΕΕΕΣ, Προμηθείς, Ηλεκτρονικές Προμήθειες, Ευρωπαϊκό Ενιαίο Έγγραφο Σύμβασης, European Single Procurement Document

## Ευχαριστίες

Στα πλαίσια της διπλωματικής μου εργασίας, θέλω να εκφράσω τις ειλικρινείς μου ευχαριστίες προς όλους όσους συνέβαλαν στην πραγματοποίησή της. Η επιτυχία αυτής της εργασίας οφείλεται σε μια σειρά από ατομικές και συλλογικές προσπάθειες που αξίζουν ειδικής αναγνώρισης. Καταρχάς, θέλω να ευχαριστήσω την επιβλέπουσά μου κυρία Πρέντζα Ανδριάνα, για την υποστήριξή της και τις ενδελεχείς οδηγίες που παρείχε καθ' όλη τη διάρκεια της έρευνας. Οι συμβουλές και η εμπειρία της αποδείχθηκαν κρίσιμες για την επίτευξη των στόχων της εργασίας. Επίσης, θα ήθελα να εκφράσω τις ευχαριστίες μου προς τους συναδέλφους και τους φίλους που μοιράστηκαν τις γνώσεις και τις ιδέες τους, προσφέροντας ένα ευνοϊκό περιβάλλον για την ανταλλαγή απόψεων. Τέλος, θέλω να ευχαριστήσω την εκπαιδευτική κοινότητα και όλους όσους προσέφεραν στη διαμόρφωση των γνώσεων μου μέσω διαφόρων πηγών πληροφόρησης. Η συνεισφορά και η υποστήριξή σας αποτελούν πολύτιμο μέρος αυτής της πορείας και σας είμαι ευγνώμων για κάθε στιγμή που αφιερώσατε για την επιτυχή ολοκλήρωσή της.

## Πίνακας περιεχομένων

<b>1</b>	<b>ΕΙΣΑΓΩΓΗ</b> .....	<b>1</b>
1.1	ΚΑΘΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ .....	1
1.2	ΤΙ ΕΙΝΑΙ ΤΟ ΕΥΡΩΠΑΪΚΟ ΕΝΙΑΙΟ ΈΓΓΡΑΦΟ ΣΥΜΒΑΣΗΣ .....	1
1.3	ΔΟΜΗ ΕΡΓΑΣΙΑΣ .....	3
<b>2</b>	<b>ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ</b> .....	<b>4</b>
2.1	ΔΙΑΦΟΡΕΣ ΥΛΟΠΟΙΗΣΕΙΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΕΕΣ .....	4
2.2	ΣΧΕΔΙΑΣΜΟΣ UI/UX ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ .....	5
2.3	ΣΥΓΓΡΑΦΗ ΚΩΔΙΚΑ ΕΦΑΡΜΟΓΗΣ.....	7
<b>3</b>	<b>ΚΑΘΟΡΙΣΜΟΣ ΑΠΑΙΤΗΣΕΩΝ</b> .....	<b>9</b>
3.1	ΧΡΗΣΤΕΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ .....	9
3.2	ΜΕΘΟΔΟΣ ΚΑΤΑΓΡΑΦΗΣ ΑΠΑΙΤΗΣΕΩΝ .....	9
3.3	ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ .....	9
3.4	ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ .....	10
<b>4</b>	<b>ΥΠΑΡΧΟΥΣΑ ΥΛΟΠΟΙΗΣΗ</b> .....	<b>11</b>
4.1	ΕΙΣΑΓΩΓΗ .....	11
4.2	ΑΡΧΙΚΗ ΕΚΚΙΝΗΣΗ ΕΦΑΡΜΟΓΗΣ .....	11
4.3	ΕΠΙΛΕΓΟΝΤΑΣ CONTRACTING AUTHORITY, NEW ESPD ΚΑΙ REGULATED.....	12
4.4	ΕΠΙΛΕΓΟΝΤΑΣ CONTRACTING AUTHORITY, NEW ESPD ΚΑΙ SELF-CONTAINED .....	13
4.5	ΕΠΙΛΕΓΟΝΤΑΣ CONTRACTING AUTHORITY, REUSE AN EXISTING ESPD .....	13
4.6	ΕΠΙΛΕΓΟΝΤΑΣ CONTRACTING AUTHORITY, REVIEW ESPD .....	14
4.7	ΕΠΙΛΕΓΟΝΤΑΣ CONTRACTING ENTITY, NEW ESPD ΚΑΙ REGULATED .....	14
4.8	ΕΠΙΛΕΓΟΝΤΑΣ CONTRACTING ENTITY, NEW ESPD ΚΑΙ SELF-CONTAINED .....	14
4.9	ΕΠΙΛΕΓΟΝΤΑΣ CONTRACTING ENTITY, REUSE AN EXISTING ESPD.....	15
4.10	ΕΠΙΛΕΓΟΝΤΑΣ CONTRACTING ENTITY, REVIEW ESPD .....	15
4.11	ΕΠΙΛΕΓΟΝΤΑΣ ECONOMIC OPERATOR, CREATE RESPONSE ΚΑΙ CREATE REGULATED .....	15
4.12	ΕΠΙΛΕΓΟΝΤΑΣ ECONOMIC OPERATOR, CREATE RESPONSE ΚΑΙ CREATE SELF-CONTAINED.....	16
4.13	ΣΥΜΠΕΡΑΣΜΑΤΑ .....	17
<b>5</b>	<b>ΣΧΕΔΙΑΣΜΟΣ ΔΙΕΠΑΦΗΣ ΧΡΗΣΤΗ - UI</b> .....	<b>18</b>
5.1	ΕΙΣΑΓΩΓΗ .....	18
5.2	ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΣΧΕΔΙΑΣΜΟΥ .....	18
5.2.1	<i>Sketch</i> .....	18
5.2.2	<i>Adobe XD</i> .....	18
5.2.3	<i>Figma</i> .....	18
5.2.4	<i>InVision Studio</i> .....	19
5.2.5	<i>Android Studio (XML)</i> .....	19
5.2.6	<i>Επιλογή προγράμματος σχεδιασμού</i> .....	19
5.3	ΧΡΩΜΑΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	19
5.4	ΟΘΟΝΗ ΣΥΝΔΕΣΗΣ/ΕΓΓΡΑΦΗΣ .....	22
5.5	ΑΡΧΙΚΗ ΟΘΟΝΗ (HOME) .....	24
5.6	ΙΣΤΟΡΙΚΟ (HISTORY) .....	26
5.7	ΡΥΘΜΙΣΕΙΣ (SETTINGS).....	28
5.8	ΔΗΜΙΟΥΡΓΙΑ ESPD ΕΓΓΡΑΦΟΥ.....	31
5.9	ΑΝΑΖΗΤΗΣΗ ΕΓΓΡΑΦΟΥ .....	36
5.10	ΑΝΑΣΚΟΠΗΣΗ ΣΧΕΔΙΑΣΗΣ ΟΘΟΝΩΝ .....	38
<b>6</b>	<b>ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ</b> .....	<b>40</b>

6.1	ΕΙΣΑΓΩΓΗ .....	40
6.2	ΑΡΧΙΤΕΚΤΟΝΙΚΗ .....	40
6.3	VERSION CONTROL.....	42
6.4	ΑΝΑΛΥΣΗ ΔΗΜΙΟΥΡΓΙΑΣ ΕΓΓΡΑΦΩΝ.....	43
6.5	ΥΠΟΣΤΗΡΙΞΗ ΠΟΛΛΑΠΛΩΝ ΓΛΩΣΣΩΝ ΕΜΦΑΝΙΣΗΣ .....	44
6.6	ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΧΡΗΣΤΩΝ.....	46
6.7	ΑΠΟΘΗΚΕΥΣΗ ΔΕΔΟΜΕΝΩΝ ΧΡΗΣΤΗ .....	48
6.8	ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΟΣ ΜΕ ΑΡΧΙΚΕΣ ΠΡΟΔΙΑΓΡΑΦΕΣ .....	50
6.9	ΔΗΜΟΣΙΕΥΣΗ ΕΦΑΡΜΟΓΗΣ.....	52
<b>7</b>	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ &amp; ΕΠΟΜΕΝΑ ΒΗΜΑΤΑ.....</b>	<b>53</b>
7.1	ΣΥΜΠΕΡΑΣΜΑΤΑ .....	53
7.2	ΠΡΟΕΤΟΙΜΑΣΙΑ ΓΙΑ ΜΕΛΛΟΝΤΙΚΕΣ ΑΛΛΑΓΕΣ.....	55
7.2.1	<i>Τρέχουσα κατάσταση του συστήματος.....</i>	<i>55</i>
<b>8</b>	<b>ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ .....</b>	<b>56</b>
<b>9</b>	<b>ΑΝΑΦΟΡΕΣ.....</b>	<b>57</b>



## Λίστα εικόνων

Εικόνα 1: Υλοποίηση συστήματος ΕΕΕΣ (Βέλγιο) .....	4
Εικόνα 2: Υλοποίηση συστήματος ΕΕΕΣ (Πολωνία).....	5
Εικόνα 3: Παλέτα για light mode .....	20
Εικόνα 4: Παλέτα για dark mode .....	20
Εικόνα 5: Υπολογισμένη παλέτα βάση του χρήστη – παράδειγμα 1 .....	21
Εικόνα 6: Υπολογισμένη παλέτα βάση του χρήστη – παράδειγμα 2 .....	22
Εικόνα 7: Οθόνη σύνδεσης .....	23
Εικόνα 8: Οθόνη εγγραφής.....	23
Εικόνα 9: Οθόνη σύνδεσης με λάθος .....	24
Εικόνα 10: Αρχική οθόνη .....	25
Εικόνα 11: Αρχική οθόνη - Κενή λίστα .....	26
Εικόνα 12: Αρχική οθόνη - Λίστα με scroll.....	26
Εικόνα 13: Αρχική οθόνη - Το info επιλεγμένο .....	26
Εικόνα 14: Οθόνη ιστορικού .....	27
Εικόνα 15: Επιλογή για προσθήκη στα αγαπημένα .....	28
Εικόνα 16: Επιλογή για κατέβασμα αρχείου .....	28
Εικόνα 17: Κενή λίστα.....	28
Εικόνα 18: Οθόνη ρυθμίσεων .....	29
Εικόνα 19: Αλλαγή γλώσσας .....	30
Εικόνα 20: Διαγραφή λογαριασμού .....	30
Εικόνα 21: Λογαριασμός μη επιβεβαιωμένος .....	31
Εικόνα 22: Λογαριασμός επιβεβαιωμένος .....	31
Εικόνα 23: Αρχή δημιουργίας εγγράφου .....	32
Εικόνα 24: Επιλογή υπάρχοντος αρχείου .....	33
Εικόνα 25: Δημιουργία εγγράφου.....	34
Εικόνα 26: Λανθασμένη εισαγωγή - συγκεκριμένο πεδίο.....	35
Εικόνα 27: Λανθασμένη εισαγωγή - Ολόκληρο παράρτημα .....	35
Εικόνα 28: Τελικό βήμα - επιλογή κατεβάσματος αρχείου .....	36
Εικόνα 29: Αναζήτηση εγγράφου .....	37
Εικόνα 30: Διαφορετικές εμφανίσεις για αποτελέσματα αναζήτησης .....	38
Εικόνα 31: Ενδεικτική οθόνη σε διαφορετική γλώσσα .....	45
Εικόνα 32: Σύνδεση με Google.....	47
Εικόνα 33: Μη επιβεβαιωμένος λογαριασμός .....	48
Εικόνα 34: Κανόνες βάσης δεδομένων .....	49
Εικόνα 35: Σύγκριση σχεδίων με τελικό αποτέλεσμα .....	51
Εικόνα 36: Δείγμα crash monitoring οθόνης .....	54

# 1 Εισαγωγή

## 1.1 Καθορισμός προβλήματος

Στην Ελλάδα όταν ένας χρήστης θέλει να δημιουργήσει ένα έγγραφο Electronic Single Procurement Document (ESPD) ή αλλιώς Ευρωπαϊκό Ενιαίο Έγγραφο Σύμβασης (ΕΕΕΣ), θα πρέπει να εισέλθει στην αντίστοιχη ιστοσελίδα<sup>1</sup> και να εκτελέσει τις ενέργειες που εμφανίζονται στην οθόνη του. Μετά το πέρας της διαδικασίας, του δίνεται η επιλογή να σώσει τοπικά στη συσκευή του αρχεία της επιλογής του. Τα αρχεία αυτά είναι της μορφής pdf, xml και html. Σε περίπτωση που θέλει να κάνει τροποποιήσεις, θα πρέπει να εισάγει στο σύστημα, αρχείο σε μορφή xml. Φυσικά όντας αυτά τα αρχεία xml, δεν είναι φιλικά προς τον χρήστη με την έννοια ότι δε θα μπορούσε να ανοίξει ένα τέτοιο έγγραφο και να καταλάβει με ευκολία όλες τις απαντήσεις που έχει δώσει. Φυσικά θα μπορούσε να εξάγει και αρχείο pdf ή html το οποίο μπορεί να είναι εύκολα κατανοητό. Το σύστημα υποστηρίζει μόνο έγγραφα xml. Έτσι αν ο χρήστης θέλει να τηρεί κατάλογο με όλα τα έγγραφα που έχει δημιουργήσει, θα πρέπει να έχει ίσως δύο αρχεία για κάθε έγγραφο, ένα αρχείο για εύκολη προεπισκόπηση των επιλογών του και το έγγραφο xml που χρειάζεται το σύστημα. Φυσικά κάτι τέτοιο δε φαίνεται να είναι ο βέλτιστος τρόπος λειτουργίας για τη διαδικασία αυτή.

Επιπροσθέτως, από την πλευρά ενός χρήστη που θέλει να δημιουργήσει μία απάντηση για έναν διαγωνισμό, τότε θα πρέπει να έχει στην κατοχή του το συγκεκριμένο έγγραφο xml της αναγγελίας του διαγωνισμού. Με αυτό το έγγραφο της αναγγελίας του διαγωνισμού και αφού το εισάγει στο σύστημα, τότε μπορεί να συμπληρώσει τα απαραίτητα στοιχεία που του ζητούνται. Σημειώνεται ότι ο χρήστης ως οικονομικός φορέας μπορεί να δημιουργήσει απάντηση σε διαγωνισμό είτε από το μηδέν συμπληρώνοντας όλα τα πεδία μόνος του, είτε συμπληρώνοντας μόνο τα πεδία που τον αφορούν με την εισαγωγή του xml αρχείου του διαγωνισμού στο σύστημα.

Κάτι ακόμα που φαίνεται να είναι περιορισμός του συστήματος είναι η έλλειψη μνήμης για κάθε χρήστη. Αν το σύστημα μπορούσε να θυμάται τις επιλογές κάθε χρήστη και όλα τα έγγραφα που έχει δημιουργήσει, τότε δε θα χρειαζόταν καν ο χρήστης να έχει στην κατοχή του έγγραφα. Θα μπορούσε να συνδεθεί, να κάνει προεπισκόπηση, και να κατεβάσει, να δημιουργήσει και να τροποποιήσει εύκολα τα έγγραφά του.

Η συγκεκριμένη διπλωματική εργασία προτίθεται να δώσει λύση στα παραπάνω υλοποιώντας εφαρμογή για κινητές συσκευές σε Android [1] περιβάλλον η οποία θα έχει λειτουργικότητες οι οποίες επιλύουν τους παραπάνω περιορισμούς. Η υλοποίηση θα πρέπει να ακολουθεί τις αρχές του clean code και της μεθοδολογίας που περιλαμβάνει τα παρακάτω: responsibility principle, open-closed principle, Liskov substitution principle, interface segregation principle, and dependency inversion principle γνωστή ως SOLID. Οι αρχές του Clean Code και η μεθοδολογία SOLID είναι σχεδιαστικές αρχές προγραμματισμού που στοχεύουν στη δημιουργία καλών, ευανάγνωστων, εύκολα συντηρήσιμων και ευέλικτο σε αλλαγές κώδικα [2]. Η μέθοδος που ακολουθήθηκε για την υλοποίηση της εφαρμογής θα αναλυθεί αναλυτικότερα σε επόμενο κεφάλαιο (Αρχιτεκτονική 6.2 παρακάτω).

## 1.2 Τι είναι το Ευρωπαϊκό Ενιαίο Έγγραφο Σύμβασης

Το ESPD είναι ένα ηλεκτρονικό έγγραφο που χρησιμοποιείται στις διαδικασίες δημόσιων συμβάσεων και δημόσιων προμηθειών. Σκοπός του ESPD είναι να διευκολύνει τις διαδικασίες υποβολής προσφορών για προμήθειες, εξοικονομώντας χρόνο και πόρους των εταιρειών που συμμετέχουν σε αυτές τις διαδικασίες. Το ESPD αντικαθιστά την ανάγκη για την υποβολή πολλαπλών εγγράφων, όπως

---

<sup>1</sup> Προμηθεύς - <https://espd.eprocurement.gov.gr/>

διάφορες διακηρύξεις και πιστοποιητικά, κατά την υποβολή προσφορών για δημόσιες συμβάσεις. Οι ενδιαφερόμενοι φορείς και επιχειρήσεις μπορούν να συμπληρώσουν το ηλεκτρονικό αυτό έγγραφο με τις αναγκαίες πληροφορίες σχετικά με την οικονομική τους κατάσταση, την τεχνική τους ικανότητα, και άλλα στοιχεία που αξιολογούνται κατά την αξιολόγηση των προσφορών. Μέσω του ESPD, οι ενδιαφερόμενοι προμηθευτές δύνανται να πιστοποιήσουν την καταλληλότητά τους χωρίς την ανάγκη υποβολής εκτενών πληροφοριών κατά τη φάση της προκήρυξης.

Γενικά τα οφέλη επιγραμματικά διακρίνονται στην απλότητα της υποβολής των προσφορών, τη μείωση της γραφειοκρατίας και την εναρμόνιση όλων των ευρωπαϊκών χωρών στον τομέα των προμηθειών. Οι εταιρείες μπορούν να υποβάλουν τις προσφορές τους ηλεκτρονικά με τη χρήση του ESPD, εξοικονομώντας τους από τον κόπο της φυσικής υποβολής εγγράφων. Επιτρέποντάς τους να συμπληρώνουν το έγγραφο online αφαιρούμε την πιθανή σύγχυση που μπορεί να υπάρξει για τον αριθμό και την υποβολή όλων των εγγράφων που χρειάζονται για τη συμμετοχή σε έναν διαγωνισμό. Έτσι, η χρήση του ESPD επιτρέπει τη μείωση της ποσότητας των απαιτούμενων εγγράφων και διαδικασιών, μειώνοντας τη γραφειοκρατία και καθιστώντας τις διαδικασίες πιο ευέλικτες. Οι προμηθευτές μπορούν να υποβάλουν μια ενοποιημένη συλλογή πληροφοριών σχετικά με την οικονομική και τεχνική τους ακαταλληλότητα, εξαιρίζοντας την ανάγκη για πολλαπλά έγγραφα. Το ESPD ενισχύει τη διαδικασία αξιολόγησης, εξασφαλίζοντας ισότιμες ευκαιρίες για συμμετοχή σε διαγωνισμούς, καθώς όλοι οι συμμετέχοντες χρησιμοποιούν την ίδια φόρμα αξιολόγησης. Η χρήση του ESPD προάγει τη συμμόρφωση με τους ευρωπαϊκούς κανονισμούς, επιτρέποντας στους οργανισμούς να ακολουθούν τα πρότυπα που καθορίζονται σε ευρωπαϊκό επίπεδο. Συνολικά, το ESPD, όπως αναφέρθηκε, βοηθά στην εξοικονόμηση χρόνου, τη μείωση της γραφειοκρατίας και την απλούστευση των διαδικασιών προμηθειών, προσφέροντας ταυτόχρονα διαφάνεια και ισότιμες ευκαιρίες για τους συμμετέχοντες.

Παρ' όλα αυτά, με τη μετατροπή της διαδικασίας προμηθειών σε ηλεκτρονική μορφή, ίσως παρουσιάσει κάποιες προκλήσεις. Η ύπαρξη επαρκούς τεχνικής υποδομής αποτελεί προϋπόθεση για την επιτυχημένη υλοποίηση του ESPD, ενώ η εκπαίδευση των χρηστών απαιτείται για τη σωστή χρήση του ηλεκτρονικού συστήματος. Τα διαδικαστικά προβλήματα, όπως προβλήματα σύνδεσης ή υποβολής ηλεκτρονικών εγγράφων, μπορεί να προκύψουν και πρέπει να αντιμετωπιστούν αποτελεσματικά από την εκάστοτε εφαρμογή που υλοποιεί τις διαδικασίες. Το επιπρόσθετο κόστος που ενδεχομένως απαιτείται για την εγκατάσταση και συντήρηση της τεχνικής υποδομής αποτελεί επίσης πρόκληση που πρέπει να ληφθεί υπόψη. Τέλος, η εξάρτηση από την τεχνολογία μπορεί να είναι προβληματική σε περιοχές με περιορισμένη πρόσβαση στο διαδίκτυο ή περιορισμένη χρήση ηλεκτρονικών συσκευών. Βέβαια, παρά τις προκλήσεις, πολλοί θεωρούν ότι τα οφέλη του ESPD υπερτερούν αυτών των προκλήσεων, συντελώντας στην αποτελεσματική και διαφανή διεξαγωγή διαγωνισμών προμηθειών.

Το ESPD είναι μια σημαντική εξέλιξη στον τομέα των δημόσιων συμβάσεων στην Ευρωπαϊκή Ένωση, στο πλαίσιο μιας συνεχούς προσπάθειας εναρμόνισης και βελτίωσης των διαδικασιών. Το ESPD έχει τις ρίζες του στην Οδηγία 2014/24/ΕΕ, που εκδόθηκε από το Ευρωπαϊκό Κοινοβούλιο και το Συμβούλιο το 2014 [3]. Η οδηγία αυτή αποτέλεσε μια σημαντική προσπάθεια για την εναρμόνιση των διαδικασιών δημόσιων συμβάσεων σε επίπεδο ΕΕ, με στόχο τη βελτίωση της αποτελεσματικότητας, της διαφάνειας, και της ισότιμης πρόσβασης σε αυτές. Η οδηγία επιβλήθηκε στα κράτη μέλη το 2016, και από τότε, παρακολουθούμε τη σταδιακή υλοποίηση και προσαρμογή του ESPD στα εθνικά νομικά πλαίσια. Σε αυτήν τη φάση, η αναζήτηση αποτελεσματικότερων, πιο διαφανών και λιγότερο γραφειοκρατικών διαδικασιών δημόσιων συμβάσεων ενισχύεται με την ευρύτερη υιοθέτηση του ESPD. Η εξέλιξη αυτή συνοδεύεται από τη συνεχή προσαρμογή στις νέες τεχνολογίες. Ο ψηφιακός τομέας παίζει καθοριστικό ρόλο, και ο στόχος είναι να δημιουργηθούν ψηφιακές λύσεις που θα

ενισχύσουν ακόμη περισσότερο την αποτελεσματικότητα και τη διαφάνεια στις διαδικασίες προμηθειών.

Οδηγία 2014/24/ΕΕ του ευρωπαϊκού κοινοβουλίου σχετικά με τις δημόσιες προμήθειες και την κατάργηση της οδηγίας 2004/18/ΕΚ ορίζει το έγγραφο με ονομασία European Single Procurement Document (ESPD) το οποίο χρησιμοποιείται κατά την προκαταρκτικό έγγραφο κατά τη διαδικασία δημόσιας προμήθειας αγαθών. Ο αγοραστής (ESPD requester), συνήθως μία κρατική υπηρεσία (Αναθέτουσα Αρχή / Contracting Authority – CA), δημοσιεύει το έγγραφο με τις προϋποθέσεις που θα πρέπει να πληροί ο προμηθευτής. Στη συνέχεια ο προμηθευτής (Οικονομικός Φορέας / Economic Operator – EO) ή κάποιο μέλος που δρα στη θέση του, συμπληρώνει την απάντηση (ESPD Response).

Ο CA θα πρέπει να μπορεί να επιλέξει είτε Regulated ESPDs, είτε Self-contained ESPDs. Στο Regulated ESPDs όλα τα έγγραφα έχουν τη δομή που ορίζεται ρητά από Annex 2 of the Commission Implementing Regulation (EU) 2016/7: Standard form for the European Single Procurement document (ESPD). Συνήθως αναφέρεται σε εξωτερικά έγγραφα που είναι ορισμένα στο eCertis των οποίων η δομή δε μεταβάλλεται φυσικά. Από την άλλη πλευρά στο Self-contained ESPD όλη η πληροφορία εμπεριέχεται σε αυτό. Έτσι μία εφαρμογή θα πρέπει να αναγνωρίζει αν πρόκειται για Regulated ή Self-contained ESPDs ώστε να εμφανίζει τα κατάλληλα πεδία προς συμπλήρωση. Παράδειγμα αυτού είναι η δήλωση απαιτήσεων για την επιλογή του προμηθευτή από την CA.

Γενικά, η οδηγία θέτει ως στόχο την εναρμόνιση των κανόνων και των διαδικασιών που διέπουν τις δημόσιες συμβάσεις σε όλα τα κράτη μέλη. Το πεδίο εφαρμογής καλύπτει τις συμβάσεις για την προμήθεια, τις υπηρεσίες και τις έρευνες και ανάπτυξη, με έμφαση στη διασφάλιση του ανοιχτού ανταγωνισμού. Ενισχύει τις αρχές της διαφάνειας, του ανοιχτού ανταγωνισμού και της ισότιμης μεταχείρισης. Καθιερώνει τις αρχές της ισότιμης μεταχείρισης, της μη διάκρισης και της αναλογικότητας. Ενισχύει τη χρήση ηλεκτρονικών μέσων για την υποβολή προσφορών, την επικοινωνία και τη διαχείριση των συμβάσεων, προκειμένου να διευκολυνθεί η συμμετοχή των οικονομικών φορέων από όλη την ΕΕ. Προβλέπει μέτρα για την ενίσχυση της συμμετοχής των μικρομεσαίων επιχειρήσεων στους διαγωνισμούς, όπως η διευκόλυνση της πρόσβασής τους σε αυτούς και η ελάφρυνση των γραφειοκρατικών φραγμάτων που συνήθως συναντούν. Η Οδηγία 2014/24/ΕΕ αποτελεί ένα σημαντικό βήμα προς τη δημιουργία ενός εναρμονισμένου νομικού πλαισίου για τις δημόσιες συμβάσεις στην ΕΕ, στηρίζοντας τις αρχές της ελεύθερης κυκλοφορίας των προϊόντων και των υπηρεσιών εντός της ενιαίας αγοράς.

### 1.3 Δομή εργασίας

Η παρούσα διπλωματική εργασία θα περιλαμβάνει μία βιβλιογραφική ανασκόπηση του τομέα της δημιουργίας των ESPD εγγράφων. Αυτό περιλαμβάνει τη σύγκριση διαφορετικών υλοποιήσεων του συστήματος σε διαφορετικές χώρες της Ευρωπαϊκής Ένωσης αλλά και τις κατευθυντήριες γραμμές ανάπτυξης του κώδικα που προέκυψαν από σχετική αναζήτηση σωστών πρακτικών υλοποίησης εφαρμογών από πλευράς κώδικα. Στη συνέχεια θα γίνει ο καθορισμός των απαιτήσεων που θα πρέπει να πληροί η υλοποίηση της εφαρμογής. Πριν την εκκίνηση της υλοποίησης θα πρέπει επίσης να προηγηθεί ανάλυση του υπάρχοντος συστήματος και καταγραφή του τρόπου λειτουργίας του. Στη συνέχεια θα πραγματοποιηθεί ο σχεδιασμός της διεπαφής του χρήστη (UI), και ακολούθως θα ξεκινήσει η συγγραφή του κώδικα. Η εργασία θα κλείσει με την παρουσίαση συμπερασμάτων και αναφορές για πιθανές μελλοντικές βελτιώσεις που μπορούν να υλοποιηθούν.

## 2 Βιβλιογραφική ανασκόπηση

### 2.1 Διάφορες υλοποιήσεις του συστήματος ΕΕΕΣ

Όπως αναφέρθηκε, το ΕΕΕΣ είναι το ελληνικό σύστημα για την παραγωγή ηλεκτρονικών εγγράφων προμηθειών όπως ορίζεται από την ΕΕ. Σε άλλες χώρες το σύστημα είναι υλοποιημένο από διάφορους φορείς, συνήθως κρατικούς. Στις παρακάτω εικόνες - Εικόνα 1: Υλοποίηση συστήματος ΕΕΕΣ (Βέλγιο) και Εικόνα 2: Υλοποίηση συστήματος ΕΕΕΣ (Πολωνία), βλέπουμε δύο διαφορετικές υλοποιήσεις από δύο διαφορετικές χώρες. Είναι εμφανές ότι και οι δύο υλοποιήσεις έχουν τα ίδια βήματα ακόμα και με ίδιες ονομασίες για την εκτέλεση της διαδικασίας δημιουργίας των ΕΕΕΣ εγγράφων.

Start Procedure Exclusion Selection Finish

Welcome to the ESPD service

**⚠** Due to a technical issue, the link with MyTelemarc is temporarily unavailable.

**i** The European Single Procurement Document (ESPD) is a self-declaration of the businesses' financial status, abilities and suitability for a public procurement procedure. It is available in all EU languages and used as a preliminary evidence of fulfilment of the conditions required in public procurement procedures across the EU.

This application has been developed using the ESPD-service's source code of the European Commission.

Contracting authorities can use it

- to create and edit ESPD-requests, and export them as PDF and/or XML;
- to view ESPD-responses and export them as PDF and/or XML.

Companies can use it

- to create or edit an ESPD-response (starting from an existing ESPD-request or ESPD-response, if available) and export it as PDF and/or XML.

For more information, please refer to our manual for contracting authorities or for economic operators.

**Important:** The ESPD is only used for tenders with an **estimated value equal to or higher than the thresholds for European publication**. However, some exceptions are foreseen for contracts awarded by negotiated procedure without prior publication (see Article 38, §1, first subparagraph, of the Royal Decree of 18 April 2017 on the award of public contracts in the traditional sectors) and for contracts awarded by negotiated procedure without prior call for competition (see Article 46, §3, of the Royal Decree of 18 June 2017 on public procurement in the water, energy, transport and postal services sectors).

As of 18 April 2018, the ESPD is to be provided in **electronic form exclusively**.

Who are you? **i**

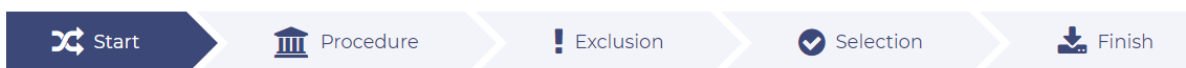
I am a contracting authority **i**

I am a contracting entity **i**

I am an economic operator **i**

Previous Start over Next

Εικόνα 1: Υλοποίηση συστήματος ΕΕΕΣ (Βέλγιο)



## Welcome to the ESPD service

**i** European Single Procurement Document (ESPD) is a self-declaration of the businesses' financial status, abilities and suitability for a public procurement procedure. It is available in all EU languages and used as a preliminary evidence of fulfilment of the conditions required in public procurement procedures across the EU. Thanks to the ESPD, the tenderers no longer have to provide full documentary evidence and different forms previously used in the EU procurement, which means a significant simplification of access to cross-border tendering opportunities. From October 2018 onwards the ESPD shall be provided exclusively in an electronic form.

The European Commission provides a free web service for the buyers, bidders and other parties interested in filling in the ESPD electronically. The online form can be filled in, printed and then sent to the buyer together with the rest of the bid. If the procedure is run electronically, the ESPD can be exported, stored and submitted electronically. The ESPD provided in a previous public procurement procedure can be reused as long as the information remains correct. Bidders may be excluded from the procedure or be subject to prosecution if the information in the ESPD is seriously misrepresented, withheld or cannot be complemented with supporting documents.

### Who are you? **i**

- I am a contracting authority **i**
- I am a contracting entity **i**
- I am an economic operator **i**

Previous

Cancel

Next

Εικόνα 2: Υλοποίηση συστήματος ΕΕΕΣ (Πολωνία)

Με περεταίρω ανάλυση στις δύο υλοποιήσεις φαίνεται να παρέχουν ακριβώς τις ίδιες δυνατότητες και λειτουργίες. Σε αντιπαράθεση με το ελληνικό σύστημα με ονομασία Προμηθεύς, παρατηρούμε επίσης ότι παρέχονται οι ίδιες δυνατότητες και λειτουργίες.

Έτσι, ομοίως με τα υπόλοιπα συστήματα της ΕΕ για την παραγωγή εγγράφων, οι περιορισμοί που έχουν αναφερθεί παραμένουν και στις άλλες υλοποιήσεις πέρα από την ελληνική. Αυτό σημαίνει ότι δεν υπάρχει η έννοια του profile χρήστη με αποτέλεσμα να μην υπάρχει ιστορικό και διάφορες διευκολύνσεις που θα ήταν άκρως χρήσιμες για έναν χρήστη του συστήματος.

Οι λύσεις που προτείνονται σε αυτή τη διπλωματική εργασία αν και απέχουν από λύσεις που μπορούν να λειτουργήσουν σε παραγωγικό περιβάλλον, αποτελούν ένα καλό παράδειγμα για το πώς θα μπορούσε να λειτουργεί ένα τέτοιο σύστημα σε κινητές συσκευές Android, και ακόμα πώς θα μπορούσε να τροποποιηθεί το υπάρχον web σύστημα ώστε να υποστηρίζει επιπλέον χρήσιμες λειτουργίες και δυνατότητες. Η υλοποίηση που παρουσιάζεται σε αυτή τη διπλωματική εργασία λόγω του ότι πραγματοποιείται ανεξάρτητα από το backend και λαμβάνοντας ως δεδομένη την τρέχουσα κατάστασή του, δημιουργούνται διάφοροι περιορισμοί. Για παράδειγμα, σε mobile εφαρμογές δεν είναι σύνηθες ο χρήστης να ανεβάζει αρχεία, πόσο μάλλον σε μορφή xml. Το συγκεκριμένο θα αναλυθεί και σε επόμενο κεφάλαιο αναλυτικότερα (6.4 Ανάλυση δημιουργίας εγγράφων). Κάτι ακόμα είναι το scalability του συστήματος. Η υλοποίηση που προτείνεται σε αυτή τη διπλωματική εργασία μπορεί να υποστηρίξει περιορισμένο αριθμό χρηστών λόγω των δωρεάν υπηρεσιών που θα χρησιμοποιηθούν.

## 2.2 Σχεδιασμός UI/UX για κινητές συσκευές

Η σχεδίαση UI/UX για mobile συσκευές είναι ένας σημαντικός παράγοντας για την επιτυχία μιας εφαρμογής. Αυτό οφείλεται στο γεγονός ότι η εμπειρία του χρήστη είναι στενά συνδεδεμένη με τον σχεδιασμό της διεπαφής χρήστη (UI), καθώς και με την ευχρηστία της εφαρμογής (UX). Στην

παρακάτω βιβλιογραφική ανασκόπηση θα εξετάσουμε τις βασικές αρχές και τεχνικές για το σχεδιασμό UI/UX για mobile συσκευές, και θα αναφέρουμε κάποια παραδείγματα επιτυχημένων εφαρμογών. Οι τάσεις στον σχεδιασμό UI/UX στοχεύουν στην εξέλιξη της εμπειρίας του χρήστη με βάση τις νέες τεχνολογίες. Η χρήση γραφικών και κινούμενων στοιχείων, όπως το micro-interactions, συνεισφέρει στην προσφορά ευχάριστης και αποδοτικής εμπειρίας. Επίσης, ο σχεδιασμός με βάση την αρχή του responsive design εξασφαλίζει την προσαρμοστικότητα της διεπαφής σε διάφορα μεγέθη οθονών.

Συγκεκριμένα, για τον σχεδιασμό μιας UI πρέπει να λαμβάνονται υπόψη οι βασικές αρχές της σχεδίασης, όπως η αισθητική, η σαφήνεια, η δομή και η συνέπεια του σχεδιασμού. Συγκεκριμένα, ο σχεδιασμός πρέπει να είναι πολύ απλός και ευανάγνωστος, και να παρέχει ευκολία στο χρήστη για να μπορεί να πλοηγηθεί στην εφαρμογή με άνεση και ομαλότητα. Επιπλέον, η δομή του σχεδιασμού πρέπει να είναι καλά οργανωμένη, ώστε ο χρήστης να βρίσκει εύκολα τις λειτουργίες της εφαρμογής και να μην αισθάνεται χαμένος στην πλοήγησή του. Για τον σχεδιασμό μιας UX, η βασική αρχή που πρέπει να λαμβάνεται υπόψη είναι η χρηστικότητα της εφαρμογής. Δηλαδή, πρέπει να είναι εύκολη στη χρήση, να ανταποκρίνεται στα αιτήματα του χρήστη και να παρέχει αποτελεσματικά και έγκυρα αποτελέσματα βάσει των προσδοκιών του χρήστη. Επιπλέον, ο σχεδιασμός UX πρέπει να είναι ευχάριστος και ενδιαφέρον για το χρήστη, και να προσφέρει καλή απόδοση και ταχύτητα στην εφαρμογή. Δεδομένου ότι η κινητή τεχνολογία είναι σε συνεχή ανάπτυξη, υπάρχουν πολλές εφαρμογές που έχουν σχεδιαστεί με επιτυχία. Παραδείγματα επιτυχημένων εφαρμογών είναι οι Spotify, Uber και Instagram, οι οποίες έχουν σχεδιαστεί με τέτοιο τρόπο ώστε να προσφέρουν μια πολύ χρηστική και ευχάριστη εμπειρία για τους χρήστες τους.

Το Material Design [4], είναι ένας πολύτιμος πόρος για οποιονδήποτε επιθυμεί να βελτιώσει τον σχεδιασμό του με βάση τις αρχές που περιγράφονται. Το Material Design είναι μια συλλογή από οδηγίες (guidelines) σχεδίασης που αναπτύχθηκε από τη Google και παρέχει μια ενιαία προσέγγιση για τη δημιουργία διεπαφών χρήστη σε διάφορες πλατφόρμες και συσκευές. Η ενσωμάτωση των αρχών και σχεδιαστικών τμημάτων που προσφέρονται από τη συλλογή, μπορεί να βελτιώσει σημαντικά την οπτική έλξη και τη χρηστικότητα οποιασδήποτε εφαρμογής.

- **Χρωματικές Παλέτες**

Το Material Design προσφέρει μια συνεκτική σειρά κατευθυντήριων γραμμών για την επιλογή χρωμάτων και παλετών που μπορούν να χρησιμοποιηθούν για τη δημιουργία οπτικά εντυπωσιακών και ενιαίων σχεδίων. Ακολουθώντας αυτές τις κατευθυντήριες γραμμές, μπορούμε να εξασφαλίσουμε ότι το κείμενό μας θα περιλαμβάνει έναν ισορροπημένο και συνεκτικό χρωματικό συνδυασμό. Οι χρωματικές παλέτες που παρέχονται από το Material Design έχουν επιλεγεί με προσοχή, λαμβάνοντας υπόψη παράγοντες όπως η προσβασιμότητα και η ιεραρχία των χρωμάτων. Χρησιμοποιώντας μια καλά σχεδιασμένη χρωματική παλέτα, μπορούμε να ενισχύσει την αναγνωσιμότητα και την ευκολότερη ανάγνωση και αναγνώριση των διαφορετικών αντικειμένων που παρουσιάζεται στην οθόνη.

- **Τυπογραφία (typography)**

Το Material Design προσφέρει κατευθυντήριες γραμμές για την επιλογή κατάλληλων γραμματοσειρών και στυλ τυπογραφίας προκειμένου να δημιουργήσει ένα οπτικά ελκυστικό και προσβάσιμο κείμενο. Εφαρμόζοντας αυτές τις κατευθυντήριες γραμμές για την τυπογραφία, μπορούμε να βεβαιωθούμε ότι το κείμενο μας είναι εύκολο στην ανάγνωση και αισθητικά ευχάριστο. Μια συνεπής και καλά δομημένη τυπογραφία μπορεί να βελτιώσει σημαντικά την συνολική εμπειρία του χρήστη με το κείμενο, καθιστώντας το πιο ελκυστικό και οπτικά ευχάριστο.

- **Κίνηση (animation)**

Το Material Design προσφέρει κατευθυντήριες γραμμές για τη χρήση της κίνησης προκειμένου να παρέχει μια ομαλή και ευαίσθητη εμπειρία χρήστη. Η κίνηση και τα γραφικά είναι κρίσιμα στοιχεία της εμπειρίας των χρηστών (UI/UX) σε μια εφαρμογή κινητού. Η κίνηση, για παράδειγμα, μπορεί να βοηθήσει τον χρήστη να καταλάβει τη σειρά εργασιών που πρέπει να ακολουθήσει, ενώ τα γραφικά μπορούν να προσθέσουν έναν παραπάνω επίπεδο ενδιαφέροντος στο χρήστη, ενθαρρύνοντας τον να εξερευνήσει την εφαρμογή και να μην την αποκλείσει από την αρχή. Επιπλέον, η κίνηση και τα γραφικά μπορούν να βελτιώσουν την αποτελεσματικότητα της εφαρμογής καθώς μπορούν να κάνουν την πλοήγηση πιο γρήγορη και ευέλικτη. Τα γραφικά μπορούν επίσης να βελτιώσουν την απόδοση της εφαρμογής με τη μείωση της φόρτωσης στον επεξεργαστή του κινητού. Τέλος, τα καλά σχεδιασμένα γραφικά και η κίνηση σε μια εφαρμογή καλύπτουν τις ανάγκες αισθητικής του χρήστη και μπορούν να αυξήσουν την ικανοποίησή του, καθιστώντας την εφαρμογή πιο ελκυστική και ευχάριστη στη χρήση.

Σημαντικό επίσης να αναφερθεί, είναι ότι η βιβλιοθήκη του Material Design περιέχει διάφορα αντικείμενα (components) για UI έτοιμα προς χρήση. Φυσικά τέτοια έτοιμα αντικείμενα είναι δύσκολα στη χρήση όταν η υλοποίηση που πρέπει να επιτευχθεί είναι εμφανισιακά και χρηστικά διαφορετική από αυτή που προσφέρει η βιβλιοθήκη. Έτσι ενώ φυσικά βοηθάει να υπάρχουν τέτοια components, προσθέτουν και έξτρα χρόνο υλοποίησης για την προσαρμογή τους στα εκάστοτε ζητούμενα.

### 2.3 Συγγραφή κώδικα εφαρμογής

Οι μεθοδολογίες που θα ακολουθηθούν για την ανάπτυξη του κώδικα της εφαρμογής θα βασιστεί στο βιβλίο του Robert C. Martin Clean Code: A Handbook of Agile Software Craftsmanship [2]. Το βιβλίο αυτό αποτελεί έναν δημοφιλή οδηγό για προγραμματιστές και επιδιώκει να προωθήσει τις βέλτιστες πρακτικές για τη συγγραφή καθαρού και συντηρήσιμου κώδικα. Ο Robert C. Martin, επίσης γνωστός ως "Uncle Bob" προωθεί αρχές όπως η απλότητα, η αναγνωσιμότητα και η σωστή δόμηση τα οποία συνεισφέρουν στη δημιουργία υψηλής ποιότητας κώδικα εφαρμογών.

Κάποια βασικά θέματα που καλύπτονται στο βιβλίο περιλαμβάνουν:

1. **Ονομασία (Naming):** Ο συγγραφέας επιμένει στη σημασία της καλής ονομασίας για μεταβλητές, συναρτήσεις, κλάσεις και άλλα στοιχεία κώδικα για την καλύτερη κατανόηση του κώδικα από άλλους προγραμματιστές.
2. **SOLID principles:** Εξηγεί τις αρχές των SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) και πώς η συμμόρφωση με αυτές μπορεί να οδηγήσει σε κώδικα που είναι ευέλικτος και εύκολος στη συντήρηση.
3. **Σχόλια (Comments):** Συμβουλές σχετικά με τη χρήση σχολίων και πώς να γράφονται κατανοητά και χρήσιμα.
4. **Διαχείριση Σφαλμάτων (Error Handling):** Προτάσεις για το πώς να διαχειρίζεστε τις εξαιρέσεις και τα σφάλματα.
5. **Τεστ (Testing):** Προτάσεις για τον τρόπο εφαρμογής και διατήρηση των unit tests και τη σημασία τους για την ποιότητα του κώδικα.

Στο σύνολο, το βιβλίο έχει ως στόχο την καθοδήγηση των προγραμματιστών προς τη δημιουργία κώδικα που όχι μόνο λειτουργεί σωστά αλλά και είναι καλοσχεδιασμένο, κατανοητό και



ευκολοδιάβαστο. Μέσα από παραδείγματα και πρακτικές, το βιβλίο αναλύει πώς οι προγραμματιστές μπορούν να εφαρμόσουν αυτές τις αρχές για τη δημιουργία κώδικα που είναι όχι μόνο λειτουργικός αλλά και καλαίσθητος, ευανάγνωστος και ευκολοδιάβαστος. Η συνολική φιλοσοφία του βιβλίου είναι ότι η προσεκτική συγγραφή του κώδικα είναι καίρια για τη δημιουργία υψηλής ποιότητας λογισμικού.

Η εφαρμογή των αρχών που προτείνει το "Clean Code: A Handbook of Agile Software Craftsmanship" παρέχει πολλαπλά οφέλη και είναι κρίσιμη για τη συγγραφή υψηλής ποιότητας κώδικα. Αυτές οι αρχές επιτρέπουν:

1. **Κατανοητό Κώδικα:** Ο κώδικας γίνεται πιο διαφανής και ευανάγνωστος. Με καλή ονομασία και δομή κώδικα, οι άλλοι προγραμματιστές (και ο ίδιος ο συγγραφέας) μπορούν να κατανοήσουν τη λειτουργία του κώδικα ευκολότερα.
2. **Συντηρησιμότητα:** Η εφαρμογή αρχών όπως οι αρχές SOLID και ο διαχωρισμός ευθυνών συνεισφέρει στην ευκολία συντήρησης. Το να είναι ο κώδικας ευανάγνωστος και οργανωμένος σημαίνει ότι είναι πιο εύκολο να προστεθούν νέα χαρακτηριστικά ή να διορθωθούν σφάλματα χωρίς να δημιουργούνται νέα προβλήματα.
3. **Ευκολία Συνεργασίας:** Ο συγκροτημένος κώδικας διευκολύνει τη συνεργασία μεταξύ προγραμματιστών. Όταν ο κώδικας είναι καλά οργανωμένος και κατανοητός, η ομάδα μπορεί να συνεργαστεί αποτελεσματικότερα χωρίς περιττές δυσκολίες.
4. **Ποιότητα Λογισμικού:** Η εφαρμογή των SOLID αρχών και άλλων αρχών σχεδίασης συνεισφέρει στη δημιουργία υψηλής ποιότητας λογισμικού. Αυτό σημαίνει λιγότερα σφάλματα, μικρότερο χρόνο ανάπτυξης, και μειωμένο κόστος συντήρησης.
5. **Βελτιστοποίηση Απόδοσης:** Ο βελτιστοποιημένος κώδικας είναι συνήθως πιο αποδοτικός. Η αποφυγή περίπλοκου, ανακύκλισης και μη απαραίτητου κώδικα οδηγεί σε καλύτερη απόδοση.

Συνολικά, η συμμόρφωση προς αυτές τις αρχές επηρεάζει θετικά την ποιότητα, τη συντηρησιμότητα και την αποδοτικότητα του λογισμικού, προσφέροντας μακροπρόθεσμα οφέλη στην ανάπτυξη και συντήρησή του.

Τα παραπάνω είναι οι κοινώς αποδεκτές κατευθυντήριες γραμμές για τη σωστή συγγραφή κώδικα οποιασδήποτε εφαρμογής από την κοινότητα. Έτσι, θα ακολουθηθούν και στον κώδικα της παρούσας διπλωματικής εργασίας για την υλοποίηση της εφαρμογής κινητών συσκευών με σκοπό την παραγωγή του καλύτερου δυνατού κώδικα που μπορεί να υποστηρίξει ακόμα και παραγωγικά περιβάλλοντα. Στόχος θα είναι ο κώδικας να μπορέσει να στηρίξει τις οποιοσδήποτε αλλαγές μπορεί να προκύψουν, με τη λιγότερη δυνατή ενέργεια των προγραμματιστών που θα ασχοληθούν.

## 3 Καθορισμός απαιτήσεων

### 3.1 Χρήστες της εφαρμογής

Η εφαρμογή που θα υλοποιηθεί θα έχει κύριο σκοπό τη δημιουργία ESPD εγγράφων. Ορισμένοι χρήστες θα τη χρησιμοποιούν για τη δημιουργία ESPD requests και άλλοι για ESPD responses. Στα πλαίσια της εργασίας δε θα υπάρξει περιορισμός των ενεργειών βάση του τύπου του χρήστη ώστε ο καθένας να είναι ελεύθερος να δημιουργεί ότι έγγραφο θέλει. Σε παραγωγικό περιβάλλον τα requests θα τα φτιάχνουν οι αναθέτουσες αρχές (contracting entity/authority) και τα responses οι οικονομικοί φορείς (economic operator). Οι αναθέτουσες αρχές μπορεί να είναι υπηρεσίες του δημοσίου ή εταιρείες. Οικονομικοί φορείς είναι εταιρίες ή ελεύθεροι επαγγελματίες που θέλουν να συμμετέχουν σε κάποιον διαγωνισμό.

### 3.2 Μέθοδος καταγραφής απαιτήσεων

Η συλλογή των απαιτήσεων για μια εφαρμογή κινητών συσκευών είναι ένα σημαντικό βήμα στη διαδικασία ανάπτυξης λογισμικού. Η καταγραφή θα συμβεί συνδυαστικά και για τις λειτουργικές και τις μη λειτουργικές απαιτήσεις. Μία μέθοδος που χρησιμοποιήθηκε ήταν οι συνεντεύξεις με τους χρήστες της υπάρχουσας εφαρμογής. Για την παραγωγή των ερωτήσεων είχε προηγηθεί ανάλυση με σκοπό τη δημιουργία μιας λίστας με πιθανά ζητούμενα. Οι χρήστες φυσικά είχαν και τη δυνατότητα να προσθέσουν και δικές τους προτάσεις. Στο τέλος της διαδικασίας πραγματοποιήθηκε φιλτράρισμα με γνώμονα τον διαθέσιμο χρόνο υλοποίησης αλλά και τη σημαντικότητα κάθε ερωτήματος/ζητούμενου. Έτσι με αυτή τη μέθοδο μπορούν να παραχθούν και λειτουργικές και μη λειτουργικές απαιτήσεις.

Μία άλλη μέθοδος ήταν ανάλυση των υπάρχοντων υλοποιήσεων. Όπως αναφέρθηκε και στο 2.1, όλες οι υλοποιήσεις του συστήματος πάσχουν από τα ίδια προβλήματα. Έτσι ήταν εύκολο να γίνει καταγραφή των λειτουργιών που χρειάζονται βελτίωση και των λειτουργιών που θα πρέπει να υλοποιηθούν από το μηδέν. Από αυτές τις βελτιώσεις παράγονται εύκολα και οι λειτουργικές απαιτήσεις.

Άλλες απαιτήσεις είναι κοινώς αποδεκτές. Για παράδειγμα όταν ένας χρήστης ανοίγει μία εφαρμογή περιμένει και να είναι σε θέση να εκτελέσει τις ενέργειες που θέλει άμεσα και χωρίς καθυστερήσεις.

### 3.3 Λειτουργικές απαιτήσεις

Οι χρήστες θα πρέπει να μπορούν να κάνουν εγγραφή χρησιμοποιώντας το e-mail τους.

Το e-mail με το οποίο έκαναν εγγραφή οι χρήστες θα πρέπει να επιβεβαιώνεται ώστε να αποτραπεί χρήση ψεύτικων διευθύνσεων.

Ο χρήστης θα πρέπει να μπορεί να δημιουργήσει έγγραφα είτε είναι συνδεδεμένος είτε όχι.

Η εφαρμογή θα πρέπει να υποστηρίζει ιστορικό είτε ο χρήστης είναι συνδεδεμένος είτε όχι.

Ένας χρήστης θα πρέπει να μπορεί να δει τα έγγραφα που έχει εκδώσει.

Ένας χρήστης θα πρέπει να μπορεί να δει τις λεπτομέρειες για καθένα από τα έγγραφα που έχει εκδώσει.

Κάθε έγγραφο που δημιουργεί ο χρήστης θα πρέπει να αποθηκεύεται τοπικά ώστε σε να μπορεί να παρουσιαστεί ξανά σε δεύτερο χρόνο. Αν ο χρήστης είναι συνδεδεμένος τότε θα πρέπει να αποθηκεύεται και online.

Ένας χρήστης θα πρέπει να μπορεί να προσθέτει και να αφαιρεί έγγραφα από τη λίστα με τα αγαπημένα του.

### 3.4 Μη λειτουργικές απαιτήσεις

Η εφαρμογή θα πρέπει να παρέχει μέτρα ασφαλείας για την προστασία των δεδομένων των χρηστών.

Η εφαρμογή θα πρέπει να είναι αποδοτική και να ανταποκρίνεται γρήγορα στις ενέργειες των χρηστών.

Τα network calls που ξέρουμε ότι μπορεί να καθυστερήσουν λόγω του μεγέθους της απάντησης, θα πρέπει να αποθηκεύονται και τοπικά (χρήση cache).

Η εφαρμογή θα πρέπει να είναι φιλική προς τον χρήστη και να παρέχει ευανάγνωστη και εύκολη στη χρήση, διεπαφή.

Η εφαρμογή θα πρέπει να μπορεί να εγκατασταθεί και να εκτελεστεί σε συσκευές που τρέχουν σε Android 8 (API 26) και πάνω.

Η εφαρμογή θα πρέπει να παρέχει λειτουργικότητα για καταγραφή σφαλμάτων.

Για την παραγωγή καινούργιων εγγράφων θα πρέπει να υπάρχει σύνδεση στο διαδίκτυο.

Η εφαρμογή θα πρέπει να υποστηρίζει ελληνικά και αγγλικά για τη γλώσσα παρουσίασης της διεπαφής.

Η εφαρμογή θα πρέπει να υλοποιηθεί βάση των αρχών του Clean Code όπως ορίζονται στο Clean Code: A Handbook of Agile Software Craftsmanship [2].

Η εφαρμογή θα πρέπει να υλοποιηθεί στη γλώσσα προγραμματισμού Kotlin με χρήση Jetpack Compose για τη διεπαφή (UI).

## 4 Υπάρχουσα υλοποίηση

### 4.1 Εισαγωγή

Η υπάρχουσα υλοποίηση αποτελείται από μια web εφαρμογή και έναν Rest API server. Κάθε χρήστης όταν θέλει να παράξει ένα έγγραφο ESPD, πρέπει να εισέλθει στην ιστοσελίδα [5] και να εκτελέσει τις ενέργειες που του ζητούνται. Αυτές οι ενέργειες στο πρώτο βήμα περιλαμβάνουν επιλογές ώστε να γνωστοποιηθεί στο σύστημα το είδος του χρήστη (ως τι τύπος να αντιμετωπιστεί) και σε επόμενα βήματα πληροφορίες που είναι απαραίτητες για την επιτυχή δημιουργία ενός ESPD εγγράφου.

Έπειτα από τη συμπλήρωση των υποχρεωτικών πεδίων στα τέσσερα βήματα (έναρξη/start, διαδικασία/procedure, αποκλεισμός/exclusion, επιλογή/selection) όπως φαίνονται στην διαδικτυακή εφαρμογή, στο πέμπτο (λήξη/finish) μπορεί να εκτελεστεί η δημιουργία του εγγράφου και να ληφθεί τοπικά στη συσκευή του. Ο χρήστης έτσι έχει στην κατοχή του πλέον ένα αρχείο της μορφής που έχει επιλέξει και ολοκληρώνεται η διαδικασία.

Η εφαρμογή που θα δημιουργηθεί στα πλαίσια αυτής της διπλωματικής, θα χρησιμοποιεί το υπάρχον Rest API που χρησιμοποιεί και η διαδικτυακή (web) εφαρμογή. Είναι αναγκαίο να αναλυθεί η υπάρχουσα εφαρμογή με σκοπό την καταγραφή της ροής της πληροφορίας. Για αυτό το λόγο είναι ιδιαίτερα σημαντικό να καταγραφούν όλες οι κλήσεις δικτύου που γίνονται προς το API [6]. Συγκεκριμένα θα χρειαστεί να καταγραφούν οι κλήσεις δικτύου που πραγματοποιούνται βάση των επιλογών του χρήστη στο βήμα έναρξη/start της διαδικτυακής εφαρμογής.

### 4.2 Αρχική εκκίνηση εφαρμογής

Κατά την είσοδο στην αρχική οθόνη δεν εκτελείται κάποια ενέργεια. Με την επιλογή γλώσσας ο χρήστης μεταφέρεται στο πρώτο βήμα της ροής. Σε αυτό το σημείο εκτελούνται διαδοχικές κλήσεις με σκοπό την αρχικοποίηση τιμών στην εφαρμογή. Οι κλήσεις αυτές αναλυτικότερα είναι οι παρακάτω:

1. `api/v2/codelists/Currency/lang/en`
2. `api/v2/codelists/CountryIdentification/lang/en`
3. `api/v2/codelists/EOIDType/lang/en`
4. `api/v2/codelists/BidType/lang/en`
5. `api/v2/codelists/FinancialRatioType/lang/en`
6. `api/v2/codelists/ProcedureType/lang/en`
7. `api/v2/codelists/ProjectType/lang/en`
8. `api/v2/codelists/EORoleType/lang/en`
9. `api/v2/codelists/WeightingType/lang/en`
10. `api/v2/codelists/LanguageCodeEU/lang/en`

Παρατηρώντας το API, είναι φανερό ότι οι κλήσεις αυτές είναι πάνω στο ίδιο endpoint με διαφορετικές μεταβλητές. Συγκεκριμένα υπάρχουν δύο μεταβλητά κομμάτια του endpoint, το `codelist` και το `language`. Η δομή δηλαδή του endpoint είναι η εξής: `/v2/codelists/{codelist}/lang/{language}`

Επιλέγοντας αλλαγή γλώσσας, εκτελείται κλήση δικτύου σε τοπικά αρχεία της web εφαρμογής με σκοπό την ανάκτηση όλων των λεκτικών που θα χρειαστούν στο UI. Η συγκεκριμένη κλήση στη mobile εφαρμογή δε θα ληφθεί υπόψη και θα δημιουργηθεί διαφορετικός μηχανισμός για τη διαχείριση των λεκτικών ο οποίος θα αναλυθεί σε επόμενο κεφάλαιο. Το συγκεκριμένο προσθέτει αρκετό παραπάνω χρόνο ανάπτυξης για την εφαρμογή δεδομένου ότι τα λεκτικά αυτά είναι δυναμικά. Παρόλα αυτά μεταφράσεις που δεν είναι δυναμικές θα συμπεριληφθούν. Περισσότερα θα αναφερθούν στο 6.5 Υποστήριξη πολλαπλών γλωσσών εμφάνισης.

Στη συνέχεια θα καταγραφούν οι κλήσεις δικτύου που πραγματοποιούνται, βάση των επιλογών του χρήστη στο πρώτο βήμα της υπάρχουσας web εφαρμογής. Στόχος είναι να γίνει ξεκάθαρο, τι πληροφορία χρειάζεται να έχει η εφαρμογή android για να εκτελέσει τις ίδιες ενέργειες που εκτελεί και η web εφαρμογή και πως τις αποκτά. Στο πως τις αποκτά απάντηση θα δοθεί από το API. Άρα αρκεί να καταγραφούν όλες οι κλήσεις δικτύου.

### 4.3 Επιλέγοντας contracting authority, new ESPD και Regulated

Σε αυτό το σημείο ο χρήστης έχει εκκινήσει την web εφαρμογή, έχει επιλέξει γλώσσα και προχωράει στο πρώτο βήμα του flow όπου πρέπει να επιλέξει το είδος του λογαριασμού του (αναθέτουσα αρχή – contracting authority, αναθέτων φορέας – contracting entity, και οικονομικός φορέας – economic operator), την ενέργεια που θέλει να εκτελέσει (δημιουργία νέου ΕΕΕΣ – create response, επαναχρησιμοποίηση υφιστάμενου ΕΕΕΣ – import ESPD, ανάγνωση ΕΕΕΣ – Review ESPD), τον τύπο του εγγράφου (Βάσει κανονισμού - regulated ή Αυτοτελή τύπο - self-contained) και τη χώρα που εδρεύει.

Επιλέγοντας Αναθέτουσα αρχή, Δημιουργία νέου ΕΕΕΣ/ΤΕΥΔ και βάση κανονισμού ΕΕ (regulated) και πατώντας συνέχεια παρατηρούνται οι εξής κλήσεις δικτύου:

1. `api/v2/regulated/criteria/reduction`
2. `api/v2/regulated/criteria/eo_related`
3. `api/v2/regulated/criteria/exclusion_a?contractingOperator=CONTRACTING_AUTHORITY`
4. `api/v2/regulated/criteria/exclusion_b?contractingOperator=CONTRACTING_AUTHORITY`
5. `api/v2/regulated/criteria/exclusion_c?contractingOperator=CONTRACTING_AUTHORITY`
6. `api/v2/regulated/criteria/exclusion_d?contractingOperator=CONTRACTING_AUTHORITY`
7. `api/v2/regulated/criteria/selection?contractingOperator=CONTRACTING_AUTHORITY`
8. `api/v2/regulated/criteria/selection_a?contractingOperator=CONTRACTING_AUTHORITY`
9. `api/v2/regulated/criteria/selection_b?contractingOperator=CONTRACTING_AUTHORITY`
10. `api/v2/regulated/criteria/selection_c?contractingOperator=CONTRACTING_AUTHORITY`
11. `api/v2/regulated/criteria/selection_d?contractingOperator=CONTRACTING_AUTHORITY`

Όπως φαίνεται, οι κλήσεις αυτές σχετίζονται με τα κριτήρια. Η δομή του endpoint είναι η εξής: `/v2/{qualificationApplicationType}/criteria/{filter}?contractingOperator={contractingOperator}`. Έτσι για τις επιλογές τους προηγούμενου βήματος έχουμε `qualificationApplicationType = regulated`. Άλλη πιθανή τιμή για το συγκεκριμένο είναι η `self-contained`. Η `contractingOperator` βάση της επιλογής είναι η `CONTRACTING_AUTHORITY` με άλλες πιθανές επιλογές το `CONTRACTING_ENTITY`. Ως φίλτρα στο πεδίο `filter` μπορούν να τοποθετηθούν οι εξής τιμές `EXCLUSION`, `EXCLUSION_A`, `EXCLUSION_B`, `EXCLUSION_C`, `EXCLUSION_D`, `ALL_SATISFIED`, `SELECTION`, `SELECTION_NO_ALPHA`, `SELECTION_A`, `SELECTION_B`, `SELECTION_C`, `SELECTION_D`, `EO_RELATED`, `REDUCTION`, `EO_RELATED_A`, `EO_RELATED_B`, `EO_RELATED_C`, `EO_LOTS`, `OTHER_CA`.

Σε περίπτωση που ο χρήστης επιλέξει να δει παραπάνω πληροφορίες για ένα κριτήριο, τότε γίνεται κλήση δικτύου στο τερματικό `/v2/{qualificationApplicationType}/criteria/eCertisData/{criterionUUID}/country/{country}/lang/{language}`. Η παράμετρος `qualificationApplicationType` μπορεί να λάβει δύο τιμές `regulated` και `self-contained`. Η παράμετρος `criterionUUID` είναι το `criterionUUID` για το οποίο θα ληφθούν πληροφορίες από το eCertis. Η παράμετρος `country` είναι η χώρα που θα χρησιμοποιηθεί κατά την αναζήτηση στο eCertis. Η παράμετρος `language` είναι η γλώσσα στην οποία θα ληφθούν τα λεκτικά της απάντησης.

Με αυτές τις κλήσεις δικτύου η εφαρμογή μπορεί να συλλέξει τις πληροφορίες που χρειάζεται για να δημιουργήσει τη διεπαφή βάση των αρχικών επιλογών του χρήστη. Στο τελευταίο βήμα, επιλέγοντας PDF, εκτελούνται δύο τελευταίες κλήσεις, οι ακόλουθες:

1. `api/v2/espd/request/pdf?language=en`
2. `api/v2/espd/request/pdf?language=en`

Φαίνεται να καλείται το ίδιο endpoint αλλά παρατηρώντας αναλυτικά τις κλήσεις δικτύου είναι φανερό ότι αυτές οι δύο κλήσεις εκτελούν εντελώς διαφορετικές ενέργειες.

Η πρώτη κλήση εκτελείται με OPTIONS request method και όρισμα τη γλώσσα, ενώ η δεύτερη κλήση είναι POST και στο payload περιέχονται όλες οι επιλογές του χρήστη σε JSON μορφή.

Ακολουθως το endpoint βάση της επιλογής του χρήστη επιστρέφει το έγγραφο στη μορφή που έχει ζητηθεί. Το συγκεκριμένο endpoint έχει την εξής μορφή: `/{version}/espd/{artefactType}/{exportType}?language={language}`.

Πιθανές τιμές για το version είναι οι v1 και v2 όπου στη συγκεκριμένη υλοποίηση θα χρησιμοποιηθεί το v2 χωρίς να υπάρχει δυνατότητα αλλαγής από ενέργεια του χρήστη. Πιθανές τιμές για το artefactType είναι οι request και response, για το exportType είναι οι xml, html και pdf και για το language είναι οι DA, DE, ET, EN, ES, FR, GA, HR, IT, LV, LT, HU, MT, NL, PL, PT, RO, SK, SL, FI, SV, CS, EL, BG όπως και για τα endpoints που έχουν ήδη αναφερθεί.

#### 4.4 Επιλέγοντας contracting authority, new ESPD και self-contained

Σε αυτό το σημείο ο χρήστης έχει εκκινήσει την web εφαρμογή, έχει επιλέξει γλώσσα και προχωράει στο πρώτο βήμα του flow όπου πρέπει να επιλέξει το είδος του λογαριασμού του (αναθέτουσα αρχή – contracting authority, αναθέτων φορέας – contracting entity, και οικονομικός φορέας – economic operator), την ενέργεια που θέλει να εκτελέσει (δημιουργία νέου ΕΕΕΣ – create response, επαναχρησιμοποίηση υφιστάμενου ΕΕΕΣ – import ESPD, ανάγνωση ΕΕΕΣ – Review ESPD), τον τύπο του εγγράφου (Βάσει κανονισμού - regulated ή Αυτοτελή τύπο - self-contained) και τη χώρα που εδρεύει.

Ο χρήστης επιλέγει contracting authority, new ESPD και self-contained. Εκτελούνται οι παρακάτω κλήσεις δικτύου κατά την εκκίνηση της διαδικασίας δημιουργίας ESPD εγγράφου.

1. `api/v2/regulated/criteria/reduction`
2. `api/v2/regulated/criteria/eo_related`
3. `api/v2/regulated/criteria/other_ca`
4. `api/v2/regulated/criteria/eo_lots`
5. `api/v2/regulated/criteria/exclusion_a?contractingOperator=CONTRACTING_AUTHORITY`
6. `api/v2/regulated/criteria/exclusion_b?contractingOperator=CONTRACTING_AUTHORITY`
7. `api/v2/regulated/criteria/exclusion_c?contractingOperator=CONTRACTING_AUTHORITY`
8. `api/v2/regulated/criteria/exclusion_d?contractingOperator=CONTRACTING_AUTHORITY`
9. `api/v2/regulated/criteria/selection?contractingOperator=CONTRACTING_AUTHORITY`
10. `api/v2/regulated/criteria/selection_a?contractingOperator=CONTRACTING_AUTHORITY`
11. `api/v2/regulated/criteria/selection_b?contractingOperator=CONTRACTING_AUTHORITY`
12. `api/v2/regulated/criteria/selection_c?contractingOperator=CONTRACTING_AUTHORITY`
13. `api/v2/regulated/criteria/selection_d?contractingOperator=CONTRACTING_AUTHORITY`

#### 4.5 Επιλέγοντας contracting authority, reuse an existing ESPD

Σε αυτό το σημείο ο χρήστης έχει εκκινήσει την web εφαρμογή, έχει επιλέξει γλώσσα και προχωράει στο πρώτο βήμα του flow όπου πρέπει να επιλέξει το είδος του λογαριασμού του (αναθέτουσα αρχή – contracting authority, αναθέτων φορέας – contracting entity, και οικονομικός φορέας – economic operator), την ενέργεια που θέλει να εκτελέσει (δημιουργία νέου ΕΕΕΣ – create response, επαναχρησιμοποίηση υφιστάμενου ΕΕΕΣ – import ESPD, ανάγνωση ΕΕΕΣ – Review ESPD), τον τύπο του εγγράφου (Βάσει κανονισμού - regulated ή Αυτοτελή τύπο - self-contained) και τη χώρα που εδρεύει.

Με την επιλογή του χρήστη, contracting authority και reuse an existing ESPD, η εφαρμογή εκτελεί μία κλήση δικτύου στο `api/importESPD/request` και ως απάντηση λαμβάνει όλο το αρχείο έτοιμο προς προβολή.

#### 4.6 Επιλέγοντας contracting authority, review ESPD

Σε αυτό το σημείο ο χρήστης έχει εκκινήσει την web εφαρμογή, έχει επιλέξει γλώσσα και προχωράει στο πρώτο βήμα του flow όπου πρέπει να επιλέξει το είδος του λογαριασμού του (αναθέτουσα αρχή – contracting authority, αναθέτων φορέας – contracting entity, και οικονομικός φορέας – economic operator), την ενέργεια που θέλει να εκτελέσει (δημιουργία νέου ΕΕΕΣ – create response, επαναχρησιμοποίηση υφιστάμενου ΕΕΕΣ – import ESPD, ανάγνωση ΕΕΕΣ – Review ESPD), τον τύπο του εγγράφου (Βάσει κανονισμού - regulated ή Αυτοτελή τύπο - self-contained) και τη χώρα που εδρεύει.

Με την επιλογή του χρήστη, contracting authority και review ESPD, η εφαρμογή εκτελεί μία κλήση στο `api/importESPD/response` και ως απάντηση λαμβάνει όλο το αρχείο έτοιμο προς προβολή.

#### 4.7 Επιλέγοντας contracting entity, new ESPD και regulated

Σε αυτό το σημείο ο χρήστης έχει εκκινήσει την web εφαρμογή, έχει επιλέξει γλώσσα και προχωράει στο πρώτο βήμα του flow όπου πρέπει να επιλέξει το είδος του λογαριασμού του (αναθέτουσα αρχή – contracting authority, αναθέτων φορέας – contracting entity, και οικονομικός φορέας – economic operator), την ενέργεια που θέλει να εκτελέσει (δημιουργία νέου ΕΕΕΣ – create response, επαναχρησιμοποίηση υφιστάμενου ΕΕΕΣ – import ESPD, ανάγνωση ΕΕΕΣ – Review ESPD), τον τύπο του εγγράφου (Βάσει κανονισμού - regulated ή Αυτοτελή τύπο - self-contained) και τη χώρα που εδρεύει.

Οι κλήσεις δικτύου είναι όμοιες για την επιλογή ‘contracting authority’ που αναφέρθηκε νωρίτερα. Η διαφορά για αυτές τις επιλογές είναι ότι για τη μεταβλητή `contractingOperator` δίνουμε ως είσοδο το λεκτικό `CONTRACTING_ENTITY`

1. `api/v2/regulated/criteria/reduction`
2. `api/v2/regulated/criteria/eo_related`
3. `api/v2/regulated/criteria/exclusion_a?contractingOperator=CONTRACTING_ENTITY`
4. `api/v2/regulated/criteria/exclusion_b?contractingOperator=CONTRACTING_ENTITY`
5. `api/v2/regulated/criteria/exclusion_c?contractingOperator=CONTRACTING_ENTITY`
6. `api/v2/regulated/criteria/exclusion_d?contractingOperator=CONTRACTING_ENTITY`
7. `api/v2/regulated/criteria/selection?contractingOperator=CONTRACTING_ENTITY`
8. `api/v2/regulated/criteria/selection_a?contractingOperator=CONTRACTING_ENTITY`
9. `api/v2/regulated/criteria/selection_b?contractingOperator=CONTRACTING_ENTITY`
10. `api/v2/regulated/criteria/selection_c?contractingOperator=CONTRACTING_ENTITY`
11. `api/v2/regulated/criteria/selection_d?contractingOperator=CONTRACTING_ENTITY`

Η υπόλοιπη διαδικασία δημιουργίας του ESPD εγγράφου είναι όμοια με τη δημιουργία για την επιλογή “contracting authority”.

#### 4.8 Επιλέγοντας contracting entity, new ESPD και self-contained

Σε αυτό το σημείο ο χρήστης έχει εκκινήσει την web εφαρμογή, έχει επιλέξει γλώσσα και προχωράει στο πρώτο βήμα του flow όπου πρέπει να επιλέξει το είδος του λογαριασμού του (αναθέτουσα αρχή – contracting authority, αναθέτων φορέας – contracting entity, και οικονομικός φορέας – economic operator), την ενέργεια που θέλει να εκτελέσει (δημιουργία νέου ΕΕΕΣ – create response, επαναχρησιμοποίηση υφιστάμενου ΕΕΕΣ – import ESPD, ανάγνωση ΕΕΕΣ – Review ESPD), τον τύπο

του εγγράφου (Βάσει κανονισμού - regulated ή Αυτοτελή τύπο - self-contained) και τη χώρα που εδρεύει.

Με την επιλογή του χρήστη, contracting entity, new ESPD και self-contained, έχουμε τις παρακάτω κλήσεις δικτύου κατά την εκκίνηση της διαδικασίας δημιουργίας ESPD εγγράφου.

1. `api/v2/regulated/criteria/reduction`
2. `api/v2/regulated/criteria/eo_related`
3. `api/v2/regulated/criteria/other_ca`
4. `api/v2/regulated/criteria/eo_lots`
5. `api/v2/regulated/criteria/exclusion_a?contractingOperator=CONTRACTING_ENTITY`
6. `api/v2/regulated/criteria/exclusion_b?contractingOperator=CONTRACTING_ENTITY`
7. `api/v2/regulated/criteria/exclusion_c?contractingOperator=CONTRACTING_ENTITY`
8. `api/v2/regulated/criteria/exclusion_d?contractingOperator=CONTRACTING_ENTITY`
9. `api/v2/regulated/criteria/selection?contractingOperator=CONTRACTING_ENTITY`
10. `api/v2/regulated/criteria/selection_a?contractingOperator=CONTRACTING_ENTITY`
11. `api/v2/regulated/criteria/selection_b?contractingOperator=CONTRACTING_ENTITY`
12. `api/v2/regulated/criteria/selection_c?contractingOperator=CONTRACTING_ENTITY`
13. `api/v2/regulated/criteria/selection_d?contractingOperator=CONTRACTING_ENTITY`

#### 4.9 Επιλέγοντας contracting entity, reuse an existing ESPD

Σε αυτό το σημείο ο χρήστης έχει εκκινήσει την web εφαρμογή, έχει επιλέξει γλώσσα και προχωράει στο πρώτο βήμα του flow όπου πρέπει να επιλέξει το είδος του λογαριασμού του (αναθέτουσα αρχή – contracting authority, αναθέτων φορέας – contracting entity, και οικονομικός φορέας – economic operator), την ενέργεια που θέλει να εκτελέσει (δημιουργία νέου ΕΕΕΣ – create response, επαναχρησιμοποίηση υφιστάμενου ΕΕΕΣ – import ESPD, ανάγνωση ΕΕΕΣ – Review ESPD), τον τύπο του εγγράφου (Βάσει κανονισμού - regulated ή Αυτοτελή τύπο - self-contained) και τη χώρα που εδρεύει.

Επιλέγοντας contracting entity, reuse an existing ESPD, η εφαρμογή εκτελεί μία κλήση στο `api/importESPD/request` και ως απάντηση λαμβάνει όλο το αρχείο έτοιμο προς προβολή.

#### 4.10 Επιλέγοντας contracting entity, review ESPD

Σε αυτό το σημείο ο χρήστης έχει εκκινήσει την web εφαρμογή, έχει επιλέξει γλώσσα και προχωράει στο πρώτο βήμα του flow όπου πρέπει να επιλέξει το είδος του λογαριασμού του (αναθέτουσα αρχή – contracting authority, αναθέτων φορέας – contracting entity, και οικονομικός φορέας – economic operator), την ενέργεια που θέλει να εκτελέσει (δημιουργία νέου ΕΕΕΣ – create response, επαναχρησιμοποίηση υφιστάμενου ΕΕΕΣ – import ESPD, ανάγνωση ΕΕΕΣ – Review ESPD), τον τύπο του εγγράφου (Βάσει κανονισμού - regulated ή Αυτοτελή τύπο - self-contained) και τη χώρα που εδρεύει.

Επιλέγοντας contracting entity, review ESPD, η εφαρμογή εκτελεί μία κλήση στο `api/importESPD/repsonse` και ως απάντηση λαμβάνει όλο το αρχείο έτοιμο προς προβολή.

#### 4.11 Επιλέγοντας economic operator, create response και create regulated

Σε αυτό το σημείο ο χρήστης έχει εκκινήσει την web εφαρμογή, έχει επιλέξει γλώσσα και προχωράει στο πρώτο βήμα του flow όπου πρέπει να επιλέξει το είδος του λογαριασμού του (αναθέτουσα αρχή – contracting authority, αναθέτων φορέας – contracting entity, και οικονομικός φορέας – economic operator), την ενέργεια που θέλει να εκτελέσει (δημιουργία νέου ΕΕΕΣ – create response, επαναχρησιμοποίηση υφιστάμενου ΕΕΕΣ – import ESPD, ανάγνωση ΕΕΕΣ – Review ESPD), τον τύπο



του εγγράφου (Βάσει κανονισμού - regulated ή Αυτοτελή τύπο - self-contained) και τη χώρα που εδρεύει.

Με την επιλογή του χρήστη economic operator, response και regulated, εκτελούνται οι παρακάτω κλήσεις δικτύου.

1. `api/v2/regulated/criteria/eo_related_A`
2. `api/v2/regulated/criteria/eo_related_B`
3. `api/v2/regulated/criteria/eo_related_C`
4. `api/v2/regulated/criteria/eo_lots`
5. `api/v2/regulated/criteria/exclusion_a?contractingOperator=ECONOMIC_OPERATOR`
6. `api/v2/regulated/criteria/exclusion_b?contractingOperator=ECONOMIC_OPERATOR`
7. `api/v2/regulated/criteria/exclusion_c?contractingOperator=ECONOMIC_OPERATOR`
8. `api/v2/regulated/criteria/exclusion_d?contractingOperator=ECONOMIC_OPERATOR`
9. `api/v2/regulated/criteria/selection?contractingOperator=ECONOMIC_OPERATOR`
10. `api/v2/regulated/criteria/selection_a?contractingOperator=ECONOMIC_OPERATOR`
11. `api/v2/regulated/criteria/selection_b?contractingOperator=ECONOMIC_OPERATOR`
12. `api/v2/regulated/criteria/selection_c?contractingOperator=ECONOMIC_OPERATOR`
13. `api/v2/regulated/criteria/selection_d?contractingOperator=ECONOMIC_OPERATOR`
14. `api/v2/regulated/criteria/reduction`

Όπως φαίνεται από την ονομασία του endpoint, φαίνεται να πραγματοποιούνται απανωτές κλήσεις με σκοπό την αρχικοποίηση τιμών στην εφαρμογή σχετικών με τα κριτήρια. Τα δεδομένα αυτά των κριτηρίων, θα χρησιμοποιηθούν στη συνέχεια με σκοπό τη δημιουργία των πεδίων που θα περιέχει το ESPD έγγραφο και θα πρέπει να συμπληρώσει ο χρήστης.

#### 4.12 Επιλέγοντας economic operator, create response και create self-contained

Σε αυτό το σημείο ο χρήστης έχει εκκινήσει την web εφαρμογή, έχει επιλέξει γλώσσα και προχωράει στο πρώτο βήμα του flow όπου πρέπει να επιλέξει το είδος του λογαριασμού του (αναθέτουσα αρχή – contracting authority, αναθέτων φορέας – contracting entity, και οικονομικός φορέας – economic operator), την ενέργεια που θέλει να εκτελέσει (δημιουργία νέου ΕΕΕΣ – create response, επαναχρησιμοποίηση υφιστάμενου ΕΕΕΣ – import ESPD, ανάγνωση ΕΕΕΣ – Review ESPD), τον τύπο του εγγράφου (Βάσει κανονισμού - regulated ή Αυτοτελή τύπο - self-contained) και τη χώρα που εδρεύει.

Με την επιλογή του χρήστη economic operator, response και regulated, εκτελούνται οι παρακάτω κλήσεις δικτύου.

1. `api/v2/selfcontained/criteria/eo_related_A`
2. `api/v2/selfcontained/criteria/eo_related_B`
3. `api/v2/selfcontained/criteria/eo_related_C`
4. `api/v2/selfcontained/criteria/eo_lots`
5. `api/v2/selfcontained/criteria/other_ca`
6. `api/v2/selfcontained/criteria/exclusion_a?contractingOperator=ECONOMIC_OPERATOR`
7. `api/v2/selfcontained/criteria/exclusion_b?contractingOperator=ECONOMIC_OPERATOR`
8. `api/v2/selfcontained/criteria/exclusion_c?contractingOperator=ECONOMIC_OPERATOR`
9. `api/v2/selfcontained/criteria/exclusion_d?contractingOperator=ECONOMIC_OPERATOR`
10. `api/v2/selfcontained/criteria/selection?contractingOperator=ECONOMIC_OPERATOR`
11. `api/v2/selfcontained/criteria/selection_a?contractingOperator=ECONOMIC_OPERATOR`
12. `api/v2/selfcontained/criteria/selection_b?contractingOperator=ECONOMIC_OPERATOR`

13. `api/v2/selfcontained/criteria/selection_c?contractingOperator=ECONOMIC_OPERATOR`
14. `api/v2/selfcontained/criteria/selection_d?contractingOperator=ECONOMIC_OPERATOR`
15. `api/v2/selfcontained/criteria/reduction`

#### 4.13 Συμπεράσματα

Φαίνεται ότι οι κλήσεις δικτύου είναι όμοιες σε διάφορα σενάρια και αλλάζουν συγκεκριμένες παράμετροι βάση των επιλογών του χρήστη. Αυτό είναι κάτι χρήσιμο γιατί αφαιρεί αρκετή πολυπλοκότητα κατά την υλοποίηση των διαφορετικών σεναρίων. Παρόλα αυτά θα πρέπει να γίνει σωστός σχεδιασμός ώστε να εκτελούνται μόνο οι κλήσεις δικτύου που χρειάζονται. Τα δεδομένα που συλλέγονται από αυτές τις κλήσεις δικτύου θα πρέπει να τροποποιηθούν κατάλληλα από την εφαρμογή ώστε να μπορούν να χρησιμοποιηθούν. Η διαδικασία αυτή είναι γνωστή ως parsing δεδομένων. Στην περίπτωση μας αυτό είναι ιδιαίτερα δύσκολο γιατί τα έγγραφα ESPD είναι μεγάλα σε μέγεθος.

Πρακτικά θα πρέπει κάθε αντικείμενο μέσα στην απάντηση η οποία είναι σε Json μορφή, να μεταφραστεί σε μία κλάση. Η δυσκολία έγκειται στο γεγονός ότι ένα αντικείμενο μπορεί να περιέχει πολλαπλά άλλα αντικείμενα ίδιου τύπου. Για τη μετάφρασή του σε UI components θα πρέπει να είναι αναγκαστικά μία αναδρομική διαδικασία. Αυτή η διαδικασία θα πρέπει να μπορεί να παράξει τα πεδία που θα μπορεί να συμπληρώσει ο χρήστης. Τα δεδομένα αυτά πιο συγκεκριμένα είναι αυτά που συλλέγονται από το endpoint /criteria. Τα κριτήρια μπορούν να περιέχουν λίστες με άλλα κριτήρια. Το γεγονός αυτό προσθέτει αρκετή πολυπλοκότητα στον κώδικα και στη λογική. Παρόλα αυτά εάν ακολουθηθούν τα βήματα και οι προτάσεις του Clean Code [2], θα μπορέσει να παραχθεί κώδικας που να είναι σωστά γραμμένος και εύκολα συντηρήσιμος.

Συνοψίζοντας, θα χρειαστεί να υλοποιηθεί η επικοινωνία API – εφαρμογής για τα τέσσερα παρακάτω τερματικά.

Τερματικό (endpoint)	Χρήση
<code>v2/codelists/{codelist}/lang/{language}</code>	Χρησιμοποιείται για το γέμισμα διάφορων drop-downs ώστε ο χρήστης να μπορεί να επιλέγει μία επιλογή από μία προκαθορισμένη λίστα.
<code>v2/{qualificationApplicationType}/criteria/{filter}</code>	Χρησιμοποιείται για το κατέβασμα των κριτηρίων. Τα ESPD έγγραφα αποτελούνται από κριτήρια τα οποία μπορεί να περιέχουν και αυτά άλλα κριτήρια.
<code>v2/esp/{artefactType}/{exportType}</code>	Χρησιμοποιείται για τη δημιουργία ESPD εγγράφων. Το αποτέλεσμα είναι ένα αρχείο μορφής XML, HTML ή PDF.
<code>importESPD/{artefactType}</code>	Χρησιμοποιείται για τη μετατροπή ενός αρχείου XML, HTML ή PDF σε json μορφή η οποία μπορεί να μεταφραστεί στη συνέχεια σε UI components.

## 5 Σχεδιασμός διεπαφής χρήστη- UI

### 5.1 Εισαγωγή

Πριν από την εκκίνηση της διαδικασίας ανάπτυξης της εφαρμογής, είναι αναγκαίο να υλοποιηθεί η γραφική απεικόνιση όλων των οθονών. Κάθε οθόνη μπορεί να έχει διάφορες καταστάσεις (*states*), οι οποίες θα πρέπει επίσης να καταγραφούν και να σχεδιαστούν. Παράδειγμα τέτοιων καταστάσεων είναι σημεία όπου μπορεί να υπάρξουν σφάλματα, παραδείγματος χάρη, από κλήσεις δικτύου ή από λανθασμένες ενέργειες του χρήστη.

### 5.2 Ανάλυση προγραμμάτων σχεδιασμού

Υπάρχουν διάφορα προγράμματα που μπορεί να χρησιμοποιήσει κάποιος για εύκολη δημιουργία οθονών για κινητές συσκευές. Η ανάλυση θα επικεντρωθεί στις δυνατότητες αυτών των εργαλείων να απεικονίζουν οθόνες κινητών συσκευών. Βασικό ζητούμενο είναι να μπορούν να επαναχρησιμοποιηθούν ίδια κομμάτια σε διαφορετικές οθόνες με ευκολία. Η λογική που θα ακολουθηθεί δηλαδή, είναι η σχεδίαση να ξεκινήσει από το μικρότερο κομμάτι και να συνεχιστεί με μεγαλύτερα και τέλος ολόκληρες τις οθόνες. Αρχικά θα σχεδιαστούν τα *components* όπως κουμπιά, κείμενα, πεδία που μπορεί ο χρήστης να γράψει, αντικείμενα που θα χρησιμοποιηθούν σε λίστες, το *top bar*, το *bottom bar*, χρώματα, γραμματοσειρές και διάφορα άλλα. Αφότου αυτά είναι έτοιμα, τότε θα ακολουθήσει ο σχεδιασμός των οθονών με χρήση όλων των δομικών στοιχείων που παράχθηκαν από το προηγούμενο βήμα. Με γνώμονα αυτό το ζητούμενο, στη συνέχεια θα αναλυθούν τα πιο γνωστά και διαδεδομένα εργαλεία UI/UX.

#### 5.2.1 Sketch

Το Sketch είναι γνωστό για τη φιλική διεπαφή που έχει προς τον χρήστη. Μπορεί ακόμα και κάποιος που δεν έχει γνώσεις πάνω στο αντικείμενο να ξεκινήσει να το χρησιμοποιεί και εύκολα να μάθει ότι χρειάζεται ώστε σε σύντομο χρονικό διάστημα να είναι σε σημείο όπου να μπορεί να σχεδιάζει ολόκληρες οθόνες. Ειδικά θετικό είναι ότι το συγκεκριμένο εργαλείο είναι σχεδιασμένο για τη δημιουργία διεπαφών χρήστη, προσφέροντας δυνατότητες που ευνοούν τον σχεδιασμό του UI/UX. Σημαντικό επίσης είναι η υποστήριξη *plugins* τα οποία επιτρέπουν στους σχεδιαστές να επεκτείνουν τις δυνατότητες του Sketch, προσαρμόζοντάς το στις ανάγκες τους. Μειονέκτημα του Sketch είναι ότι είναι διαθέσιμο μόνο για χρήστες macOS. Αυτό περιορίζει τη χρήση του από σχεδιαστές που χρησιμοποιούν άλλα λειτουργικά συστήματα. Λόγω του περιορισμού αυτού, το Sketch, δε θα μπορέσει να χρησιμοποιηθεί για τις ανάγκες αυτής της διπλωματικής εργασίας.

Επίπεδο δυσκολίας: Εύκολο

#### 5.2.2 Adobe XD

Με χρήση του Adobe XD μπορεί ο χρήστης να επωφεληθεί από τη διαλειτουργικότητα μεταξύ όλων των εφαρμογών της Adobe. Το Photoshop και το Illustrator είναι μερικές άλλες εφαρμογές της εταιρείας και είναι πιθανό να φανούν χρήσιμες. Το XD παρέχει εργαλεία για *prototyping* και σχεδιασμό εξοικονομώντας χρόνο και απλοποιώντας τη διαδικασία σχεδίασης. Μειονέκτημα του εργαλείου είναι το γεγονός ότι δεν υποστηρίζει επεκτάσεις από τρίτους, έτσι ο χρήστης είναι περιορισμένος στις δυνατότητες που προσφέρει η εταιρεία.

Επίπεδο δυσκολίας: Μέτριο-Υψηλό

#### 5.2.3 Figma

Το Figma είναι μία εφαρμογή ιστού όπου μπορεί ο χρήστης να δουλεύει συνεργατικά με άλλα άτομα της ομάδας του για τη σχεδίαση γραφικών. Η web-based φύση της εφαρμογής δίνει την ευχέρεια στο χρήστη να μπορεί να δουλέψει από οποιαδήποτε συσκευή θέλει. Αυτό βέβαια για κάποιους χρήστες

ίσως είναι κάτι αρνητικό γιατί προτιμούν τις εφαρμογές που εκτελούνται τοπικά στον υπολογιστή τους.

Επίπεδο δυσκολίας: Εύκολο

#### 5.2.4 InVision Studio

Το InVision Studio ξεχωρίζει για τα ισχυρά του εργαλεία για τη δημιουργία prototypes και την προσθήκη κινήσεων σε αυτά. Κυρίως χρησιμοποιείται για γραφικά που απαιτούν κίνηση. Το πρόγραμμα επιτρέπει τη συνεργασία μεταξύ των σχεδιαστών ώστε να επιτυγχάνεται μέγιστη αποτελεσματικότητα. Κάτι αρνητικό που έχουν εκφράσει οι χρήστες του συστήματος είναι ότι πάσχει από έλλειψη ευελιξίας.

Επίπεδο δυσκολίας: Μέτριο-Υψηλό

#### 5.2.5 Android Studio (XML)

Το Android Studio είναι το λογισμικό στο οποίο οι προγραμματιστές αναπτύσσουν τις εφαρμογές για το Android λογισμικό. Το εργαλείο παρέχει xml έγγραφα στα οποία με κώδικα, μπορεί κάποιος να σχεδιάσει οποιαδήποτε οθόνη θέλει. Το εργαλείο πέρα από τον κώδικα είναι εφικτό να δημιουργήσει οθόνες και χωρίς την συγγραφή κώδικα αλλά με έτοιμα γραφικά δομικά στοιχεία που μπορεί ο χρήστης να τοποθετήσει με το ποντίκι του στη θέση που τα θέλει. Η μέθοδος αυτή προορίζεται για την υλοποίηση των οθονών και λιγότερο για τη σχεδίαση.

Επίπεδο δυσκολίας: Μέτριο-Υψηλό

#### 5.2.6 Επιλογή προγράμματος σχεδιασμού

Ένα σημαντικό κριτήριο το οποίο έχει αποφασιστικό ρόλο είναι η άδειας χρήσης των παραπάνω εργαλείων. Από αυτά μόνο το Figma είναι δωρεάν. Τα υπόλοιπα προγράμματα (Sketch, Adobe XD, InVision Studio, Android Studio) διαθέτουν περιορισμένες δωρεάν εκδόσεις ή δυνατότητες δοκιμής, αλλά για πλήρη χρηστικότητα και επιπλέον χαρακτηριστικά, απαιτείται συνήθως συνδρομή ή αγορά της άδειας χρήσης.

Επιπροσθέτως, για τις ανάγκες της εργασίας ήταν αναγκαίο να βρεθεί και να χρησιμοποιηθεί ένα εργαλείο που είναι εύκολο στη χρήση, να υπάρχει αρκετή γνώση online (plugins, tutorials κτλ) ώστε να επιτευχθεί μέγιστη παραγωγικότητα κατά αυτό το στάδιο της σχεδίασης των οθονών.

Κάτι επίσης θετικό θα ήταν να είναι δυνατή από οποιαδήποτε συσκευή, η πρόσβαση στα αρχεία, με όσο το δυνατόν λιγότερη τριβή. Σε εφαρμογές οι οποίες τρέχουν τοπικά στον υπολογιστή είναι δύσκολο να παρέχουν τέτοια δυνατότητα. Αν και πολλά από τα εργαλεία που αναφέρθηκαν διαθέτουν δυνατότητες συγχρονισμού, το Figma δουλεύει εγγενώς με αυτό τον τρόπο λόγω του ότι είναι μια εφαρμογή ιστού (*web app*).

Έτσι, βάση των παραπάνω, το Figma είναι το εργαλείο που επιλέχθηκε για τον σχεδιασμό της διεπαφής και των δομικών μερών της εφαρμογής.

### 5.3 Χρώματα της εφαρμογής

Η επιλογή χρωμάτων είναι πολύ σημαντική για ένα ηλεκτρονικό προϊόν. Είναι σημάδι κατατεθέν συνήθως και βοηθάει τον χρήστη να αναγνωρίσει γρήγορα την ταυτότητα του προϊόντος. Υπάρχει μεγάλη θεωρία πίσω από την επιλογή της βασικής παλέτας που θα χρησιμοποιήσει κάποιος για το ηλεκτρονικό προϊόν του. Η επιλογή των χρωμάτων για ένα brand δεν είναι απλώς αισθητικής φύσεως αλλά συνδέεται στενά με το μήνυμα και την ταυτότητα που θέλει να προβάλλει. Ένα καλά επιλεγμένο χρώμα μπορεί να ενισχύσει την αναγνωρισιμότητα και την αποτελεσματικότητα του brand στην αγορά.

Συγκεκριμένα για την παρούσα εφαρμογή για κινητές συσκευές θα χρησιμοποιηθούν δύο παλέτες. Μία για light mode και μία για dark mode Εικόνα 3: Παλέτα για light mode, Εικόνα 4: Παλέτα για dark mode.

Primary P-40	Secondary S-40	Tertiary T-40	Error E-40		
On Primary P-100	On Secondary S-100	On Tertiary T-100	On Error E-100		
Primary Container P-90	Secondary Container S-90	Tertiary Container T-90	Error Container E-90		
On Primary Container P-10	On Secondary Container S-10	On Tertiary Container T-10	On Error Container E-10		
Primary Fixed P-90	Primary Fixed Dim P-80	Secondary Fixed S-90	Secondary Fixed Dim S-80	Tertiary Fixed T-90	Tertiary Fixed Dim T-80
On Primary Fixed P-10	On Secondary Fixed S-10	On Tertiary Fixed T-10	On Primary Fixed Variant S-30	On Secondary Fixed Variant S-30	On Tertiary Fixed Variant T-30
Surface Dim N-87	Surface N-98	Surface Bright N-98	Inverse Surface N-20		
Surf. Container Lowest N-100	Surf. Container Low N-96	Surf. Container N-94	Surf. Container High N-92	Surf. Container Highest N-90	Inverse On Surface N-95
On Surface N-10	On Surface Var. NV-30	Outline NV-50	Outline Variant NV-80	Scrim N-0	Shadow N-0

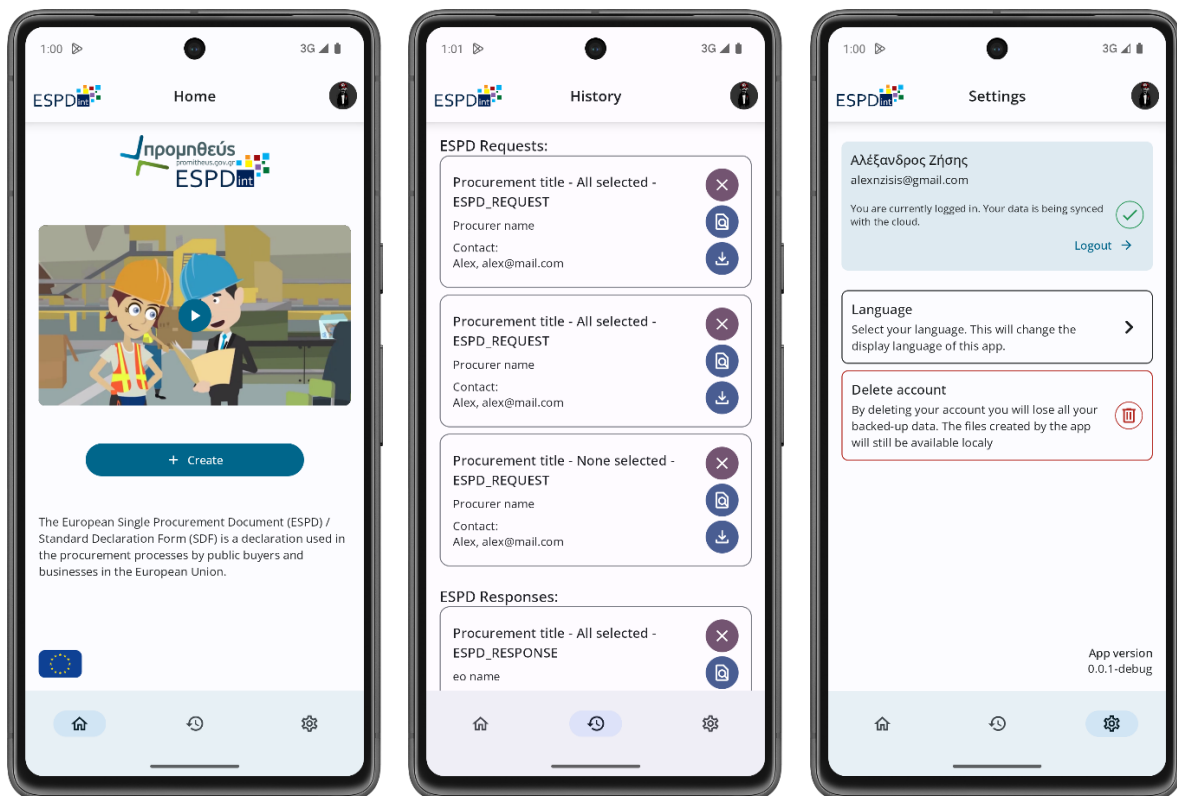
Εικόνα 3: Παλέτα για light mode

Primary P-80	Secondary S-80	Tertiary T-80	Error E-80		
On Primary P-20	On Secondary S-20	On Tertiary T-20	On Error E-80		
Primary Container P-30	Secondary Container S-30	Tertiary Container T-30	Error Container E-30		
On Primary Container P-90	On Secondary Container S-90	On Tertiary Container T-90	On Error Container E-90		
Primary Fixed P-90	Primary Fixed Dim P-80	Secondary Fixed S-90	Secondary Fixed Dim S-80	Tertiary Fixed T-90	Tertiary Fixed Dim T-80
On Primary Fixed P-10	On Secondary Fixed S-10	On Tertiary Fixed T-10	On Primary Fixed Variant S-30	On Secondary Fixed Variant S-30	On Tertiary Fixed Variant T-30
Surface Dim N-6	Surface N-6	Surface Bright N-24	Inverse Surface N-90		
Surf. Container Lowest N-4	Surf. Container Low N-10	Surf. Container N-12	Surf. Container High N-17	Surf. Container Highest N-22	Inverse On Surface N-20
On Surface N-90	On Surface Var. NV-80	Outline NV-60	Outline Variant NV-30	Scrim N-0	Shadow N-0

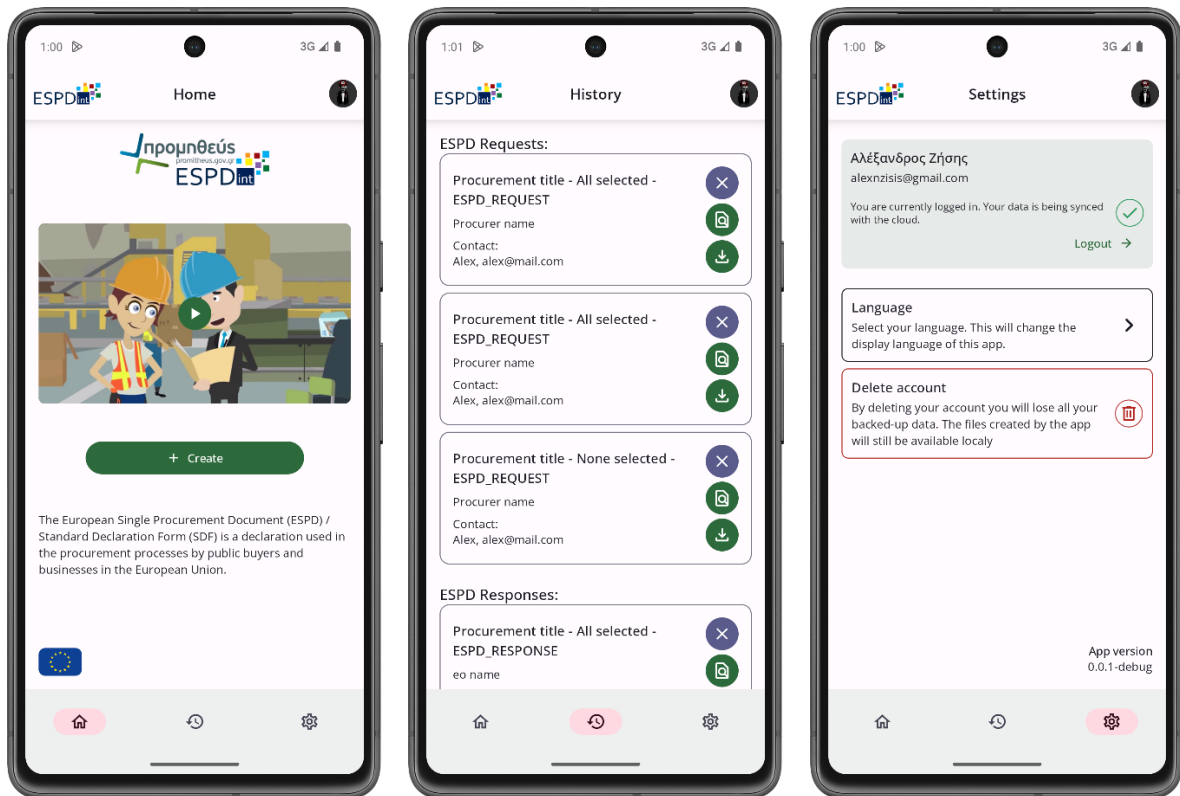
Εικόνα 4: Παλέτα για dark mode

Με την κωδική ονομασία *Surface* ορίζεται το χρώμα της ουδέτερης επιφάνειας. Έτσι μπορούμε να καταλάβουμε ότι στο dark mode οι επιφάνειες έχουν σκούρο χρώμα, ενώ σε light mode φωτεινό. Παρόμοια λογική ακολουθούν και τα υπόλοιπα χρώματα.

Αυτά τα χρώματα θα χρησιμοποιηθούν για τη γραφική απεικόνιση των οθονών και βάση αυτών θα γίνει και η ανάπτυξη του κώδικα. Σημειώνεται ότι στην τελευταία έκδοση του Android είναι δυνατό να παράγονται αυτόματα νέα χρώματα και αποχρώσεις βάση του φόντου της ταπετσαρίας που έχει επιλέξει ο χρήστης. Αυτό φυσικά για να λειτουργήσει πρέπει να έχει ληφθεί υπόψη. Επιλέχθηκε να γίνει ενεργοποίηση αυτής τη δυνατότητας αφού δεν προσθέτει σημαντικό χρόνο στην ανάπτυξη του κώδικα. Παρακάτω φαίνονται δύο εκδοχές δυναμικού υπολογισμού χρωμάτων βάση του φόντου που έχει επιλέξει ο χρήστης Εικόνα 5, Εικόνα 6.



Εικόνα 5: Υπολογισμένη παλέτα βάση του χρήστη – παράδειγμα 1

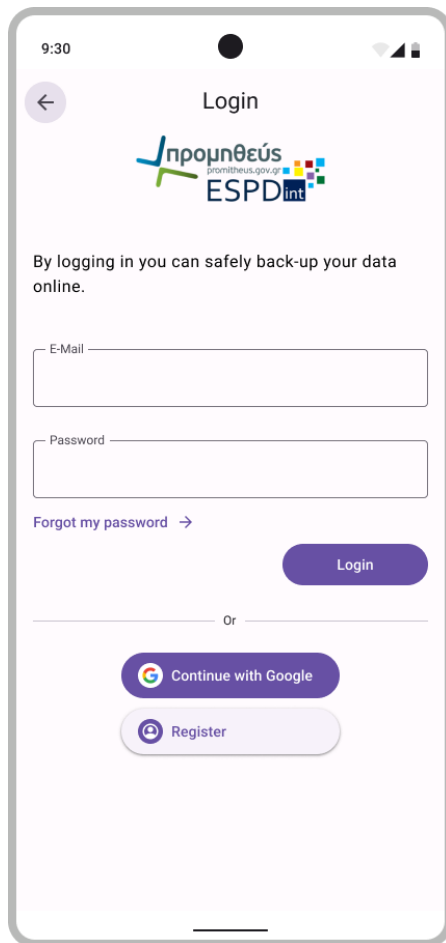


Εικόνα 6: Υπολογισμένη παλέτα βάση του χρήστη – παράδειγμα 2

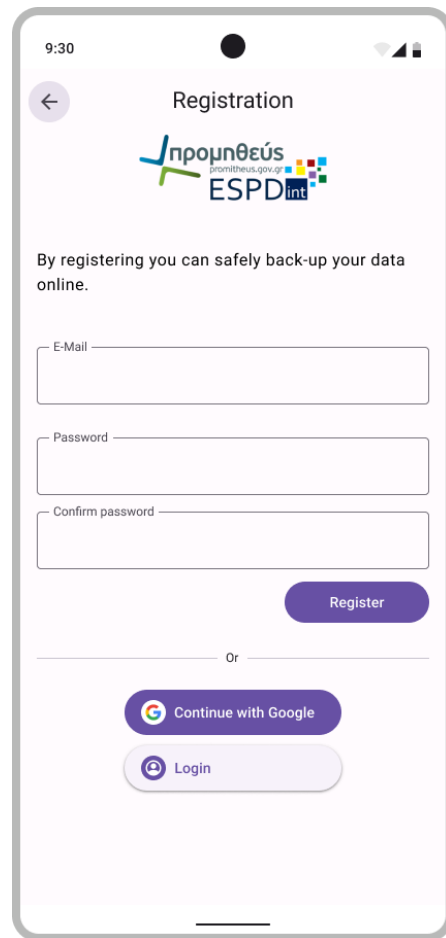
\*Σημειώνεται ότι οι Εικόνα 5 και Εικόνα 6 παρουσιάζουν οθόνες από το τελικό παραδοτέο της εφαρμογής. Αυτό σημαίνει ότι υπάρχουν αλλαγές και τροποποιήσεις σε σχέση με το αρχικό σχέδιο.

#### 5.4 Οθόνη σύνδεσης/εγγραφής

Όπως έχει αναφερθεί, η εφαρμογή θα πρέπει να υποστηρίζει αυθεντικοποίηση. Κάθε χρήστης θα πρέπει να μπορεί να συνδεθεί στον λογαριασμό του από την οθόνη σύνδεσης ή να κάνει εγγραφή από την οθόνη εγγραφής. Παρακάτω φαίνονται οι οθόνες για τη σύνδεση και την εγγραφή.



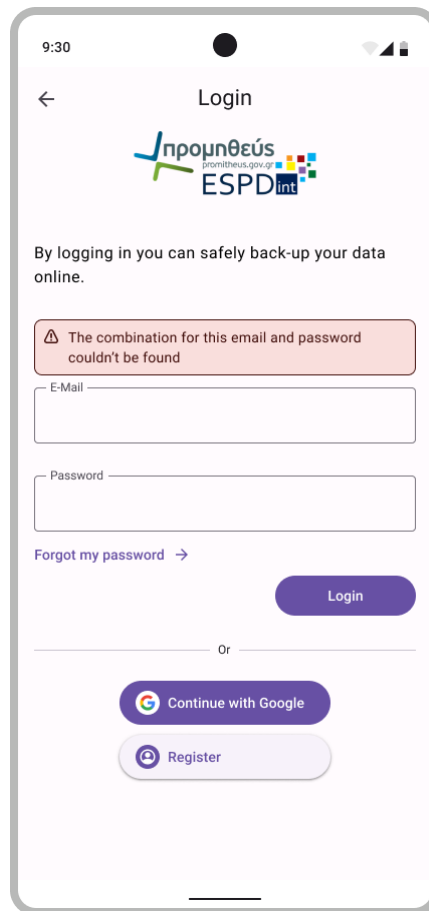
Εικόνα 7: Οθόνη σύνδεσης



Εικόνα 8: Οθόνη εγγραφής

Μηνύματα λάθους είναι σύνηθες να παρουσιάζονται σε τέτοιες οθόνες. Έτσι και στις δύο οθόνες, σε περίπτωση σφάλματος, εμφανίζεται ένα πλαίσιο με έντονο χρώμα το οποίο να ενημερώνει τον χρήστη για το γεγονός που πήγε στραβά. Το κείμενο σε αυτό το πλαίσιο είναι δυναμικό και από άποψη γλώσσας εμφάνισης αλλά και από άποψη περιεχομένου, που σημαίνει ότι το κείμενο δεν είναι στατικό και το ίδιο για κάθε σφάλμα Εικόνα 7, Εικόνα 8, Εικόνα 9.

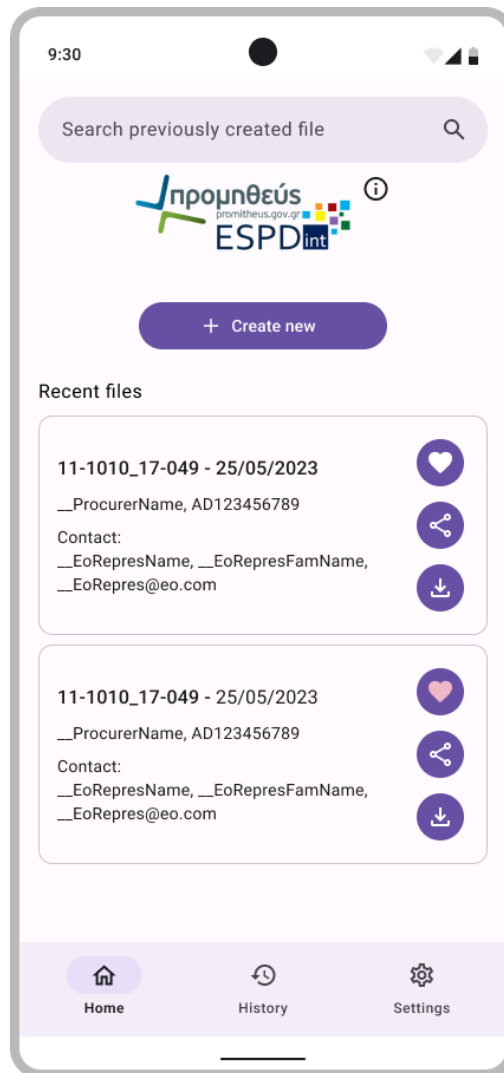




Εικόνα 9: Οθόνη σύνδεσης με λάθος

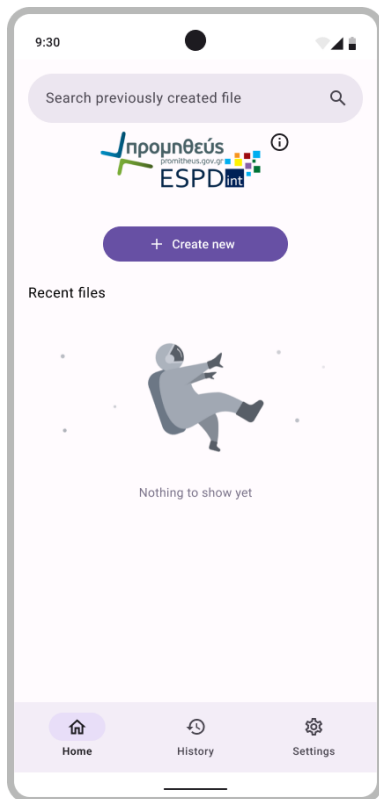
## 5.5 Αρχική οθόνη (Home)

Η αρχική οθόνη θα αποτελείται από ένα bottom bar, ένα top bar και το περιεχόμενο που θα βρίσκεται ανάμεσα. Στο top bar θα υπάρχει πεδίο το οποίο όταν πατιέται, ο χρήστης θα μεταφέρεται με κάποιο animation σε οθόνη όπου θα μπορεί να γίνει αναζήτηση ανάμεσα σε όλα τα αρχεία που έχει δημιουργήσει. Το bottom bar θα περιέχει τρεις επιλογές για μετάβαση ανάμεσα στις οθόνες Αρχική, Ιστορικό και Ρυθμίσεις. Η εναλλαγή μεταξύ αυτών των οθονών θα πρέπει να γίνει και αυτή με animation. Σημειώνεται ότι το top bar όπως και το bottom bar θα πρέπει να είναι σταθερά στην οθόνη, να μην κινούνται και να είναι εμφανή και από τις τρεις αυτές οθόνες. Μόνο το ενδιάμεσο περιεχόμενο θα πρέπει να εναλλάσσεται. Στο κεντρικό περιεχόμενο της οθόνης θα υπάρχει το λογότυπο της εφαρμογής σε μεγάλο μέγεθος. Συνεχίζοντας κάθετα προς τα κάτω, θα υπάρχει κουμπί που θα οδηγεί στη δημιουργία εγγράφου ESPD (response/request). Κάτω από το κουμπί θα υπάρχει λίστα με τα τελευταία δημιουργημένα αρχεία.

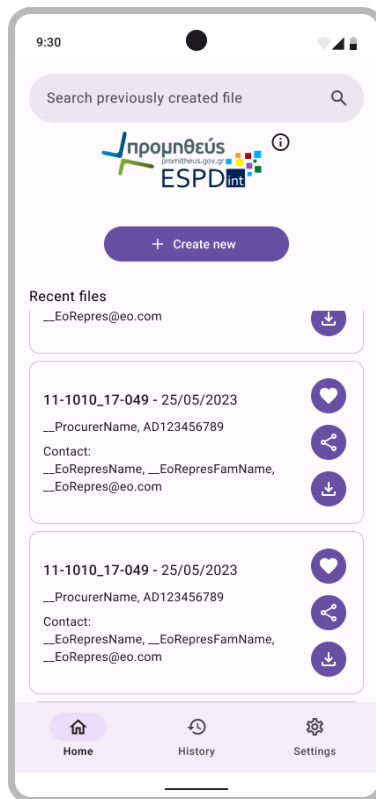


Εικόνα 10: Αρχική οθόνη

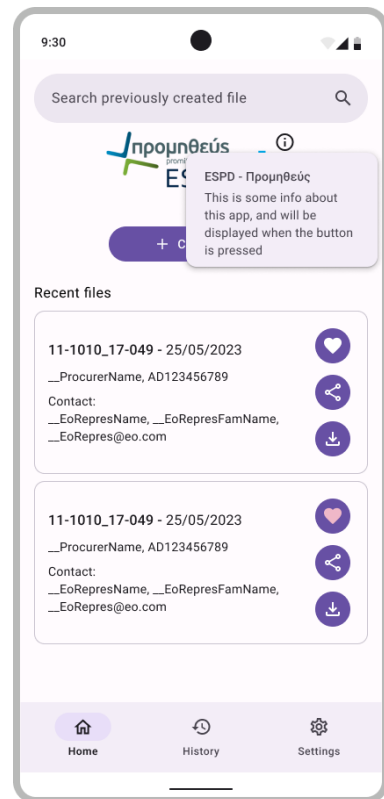
Στο σχέδιο, Εικόνα 10, είναι φανερό ότι η καρδιά μπορεί να επιλεγθεί. Σε αυτή την περίπτωση αλλάζει και χρώμα ώστε να ενημερώσει τον χρήστη για την αλλαγή. Για αυτή την οθόνη, χρειάζεται να γνωρίζουμε τι συμβαίνει όταν η λίστα είναι αρκετά μεγάλη και ο χρήστης κάνει scroll αλλά και άλλες ενέργειες του χρήστη που μπορεί να επηρεάσουν το UI τόσο ώστε να χρειάζεται επιπλέον σχέδιο. Τέτοιες καταστάσεις είναι οι παρακάτω Εικόνα 11, Εικόνα 12, Εικόνα 13.



Εικόνα 11: Αρχική οθόνη - Κενή λίστα



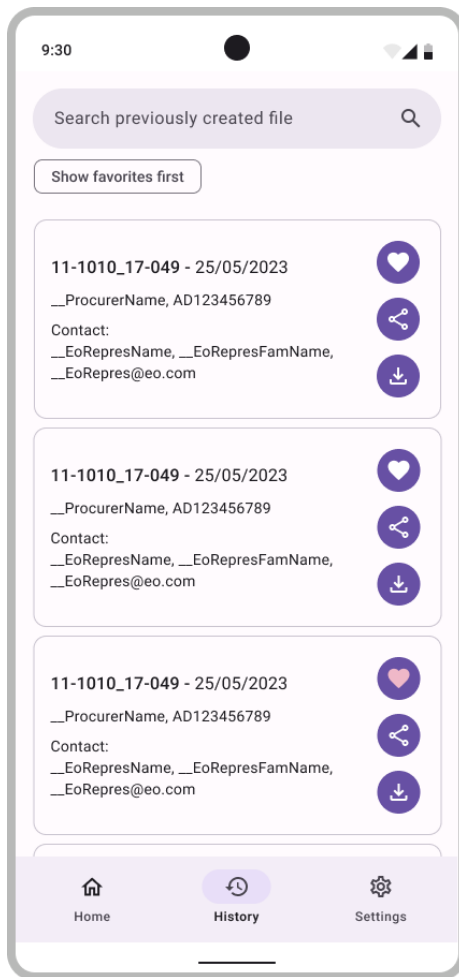
Εικόνα 12: Αρχική οθόνη - Λίστα με scroll



Εικόνα 13: Αρχική οθόνη - Το info επιλεγμένο

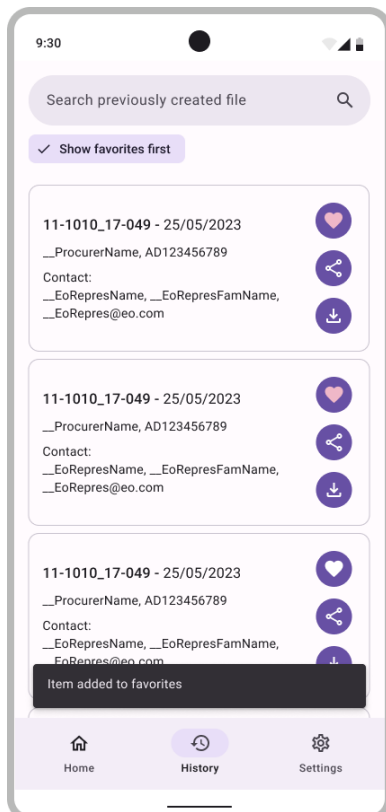
## 5.6 Ιστορικό (History)

Η οθόνη ιστορικού υπάρχει για να δίνει τη δυνατότητα επισκόπησης όλων των εγγράφων που έχει δημιουργήσει ο χρήστης. Σε αυτή την οθόνη θα μπορεί να επιλέξει να εμφανίζονται πρώτα τα αγαπημένα του. Κάθε επισκόπηση (preview) εγγράφου στη λίστα θα έχει τρεις επιλογές. Για κάθε έγγραφο θα πρέπει να μπορεί να γίνει κατέβασμα του αρχείου τοπικά στη συσκευή, διαμοιρασμός του εγγράφου και προσθήκη στα αγαπημένα. Στην παρακάτω εικόνα φαίνεται η εν λόγω οθόνη Εικόνα 14.

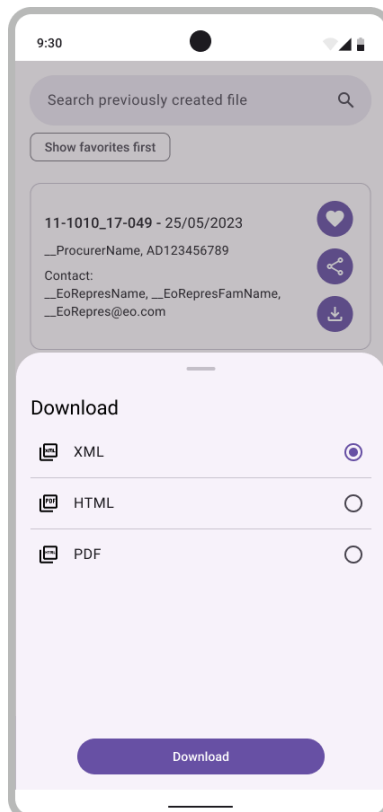


Εικόνα 14: Οθόνη ιστορικού

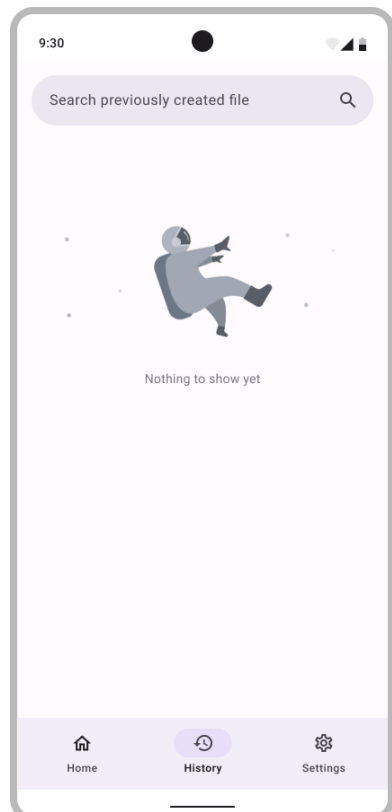
Κάποιες έξτρα καταστάσεις της οθόνης που πρέπει να σχεδιαστούν είναι το πώς θα φαίνεται όταν το κουμπί *'Show favorites first'* είναι επιλεγμένο, πώς θα φαίνεται όταν έχει γίνει scroll και πώς θα εκτελούνται οι ενέργειες των κουμπιών των αντικειμένων της λίστας. Για το τελευταίο υποθέτουμε ότι η προσθήκη στα αγαπημένα θα είναι απλά με το πάτημα του κουμπιού και θα πρέπει να γίνεται toggle η κατάσταση. Μόλις γίνεται αυτό θα πρέπει ο χρήστης να ενημερώνεται για την ενέργεια και με ένα *Toast* μήνυμα. Το κουμπί για τον διαμοιρασμό θα ανοίγει το παράθυρο του λειτουργικού συστήματος για διαμοιρασμό του αρχείου και τέλος το κουμπί για το κατέβασμα θα πρέπει να ανοίγει ένα *bottom sheet* στο το οποίο ο χρήστης να μπορεί να επιλέξει τι τύπο αρχείου θέλει να κατεβάσει. Οι καταστάσεις αυτές φαίνονται παρακάτω Εικόνα 15, Εικόνα 16, Εικόνα 17.



Εικόνα 15: Επιλογή για προσθήκη στα αγαπημένα



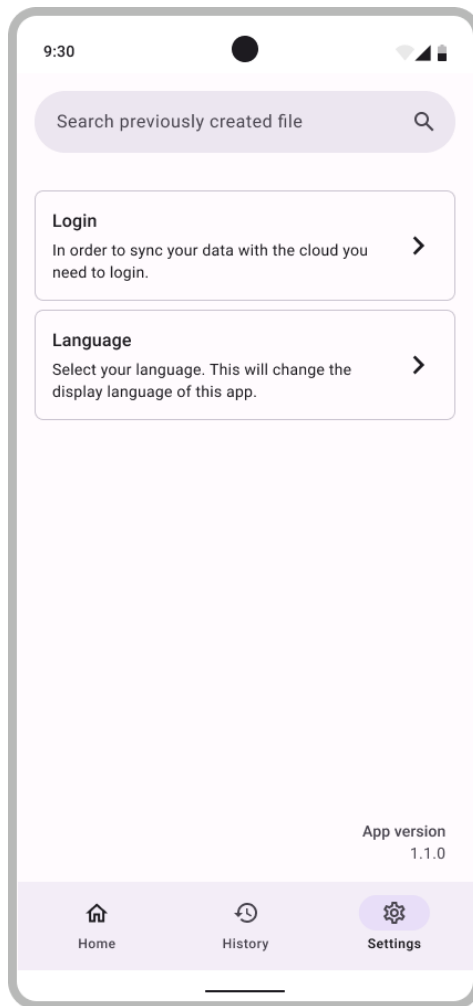
Εικόνα 16: Επιλογή για κατέβασμα αρχείου



Εικόνα 17: Κενή λίστα

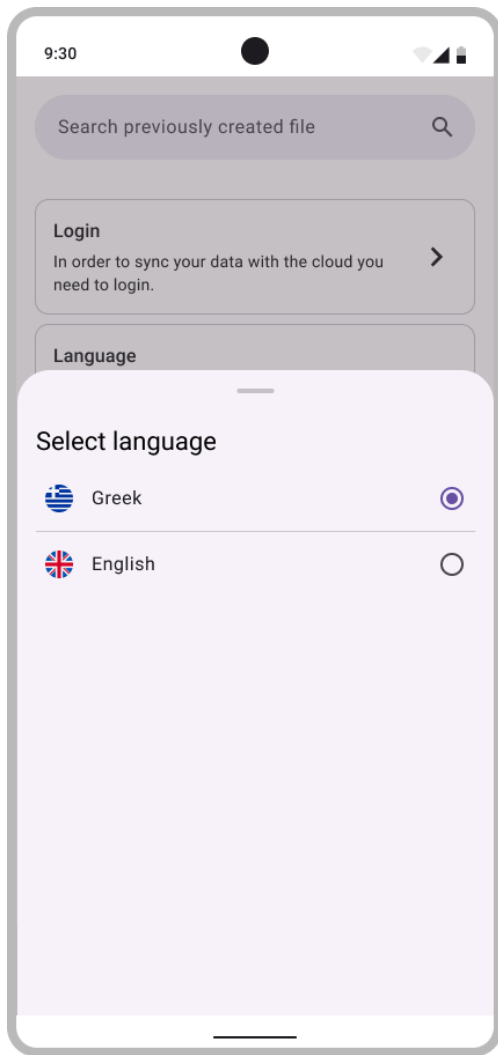
## 5.7 Ρυθμίσεις (Settings)

Στις ρυθμίσεις θα πρέπει να φαίνεται η κατάσταση του λογαριασμού του χρήστη, όπως και κουμπιά από τα οποία θα μπορεί να κάνει αποσύνδεση και να αλλάξει τη γλώσσα της εφαρμογής (Εικόνα 19). Σε περίπτωση που έχει γίνει σύνδεση, τότε θα πρέπει να υπάρχει και κουμπί για διαγραφή λογαριασμού (Εικόνα 20). Θα πρέπει να φαίνεται σε διακριτικό σημείο η έκδοση της εφαρμογής. Παρακάτω φαίνεται η εν λόγω οθόνη Εικόνα 18.

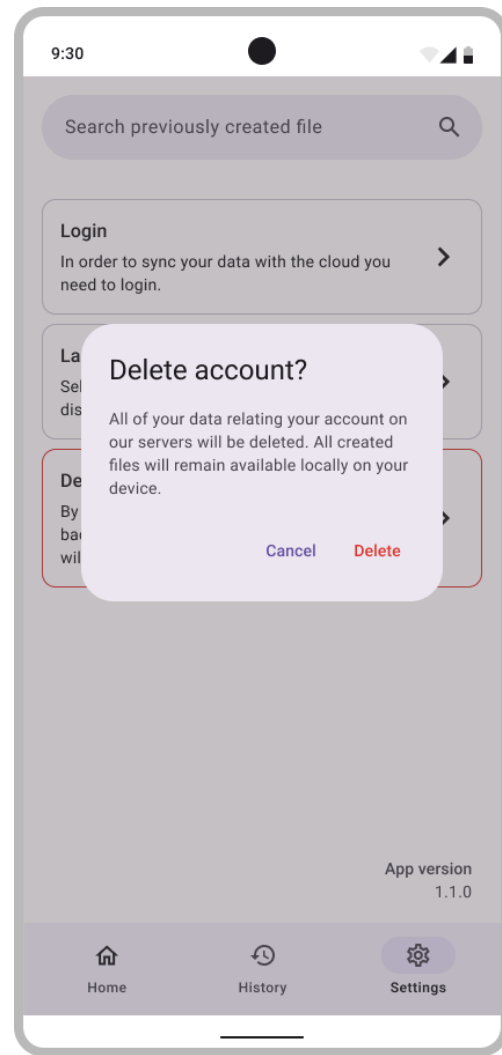


Εικόνα 18: Οθόνη ρυθμίσεων

Επιπλέον θα πρέπει να απεικονίζεται η κατάσταση του λογαριασμού. Πιθανές καταστάσεις είναι ένας λογαριασμός να είναι επιβεβαιωμένος ή όχι. Η επιβεβαίωση θα πρέπει να γίνεται βάση του e-mail του χρήστη.

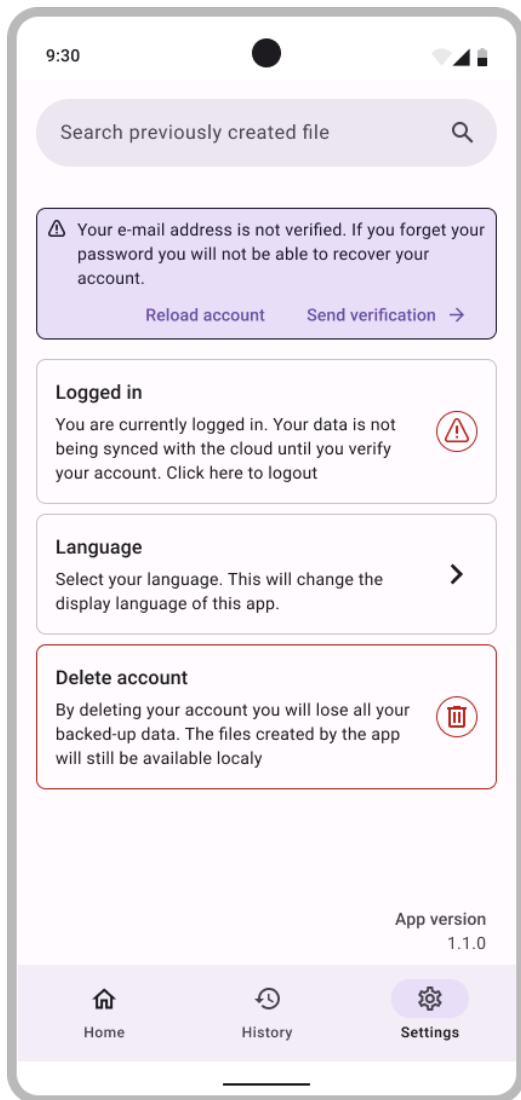


Εικόνα 19: Αλλαγή γλώσσας

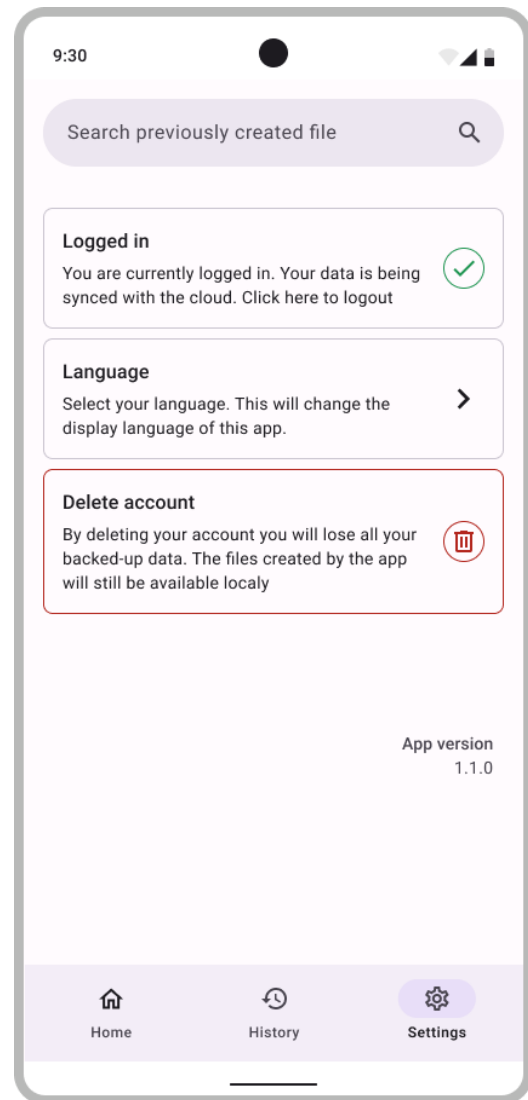


Εικόνα 20: Διαγραφή λογαριασμού

Και οι διάφορες καταστάσεις του λογαριασμού του χρήστη παρακάτω Εικόνα 21, Εικόνα 22.



Εικόνα 21: Λογαριασμός μη επιβεβαιωμένος

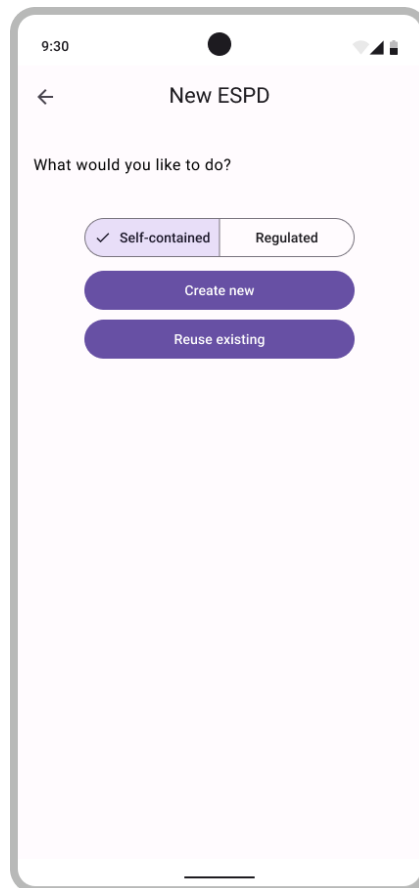


Εικόνα 22: Λογαριασμός επιβεβαιωμένος

## 5.8 Δημιουργία ESPD εγγράφου

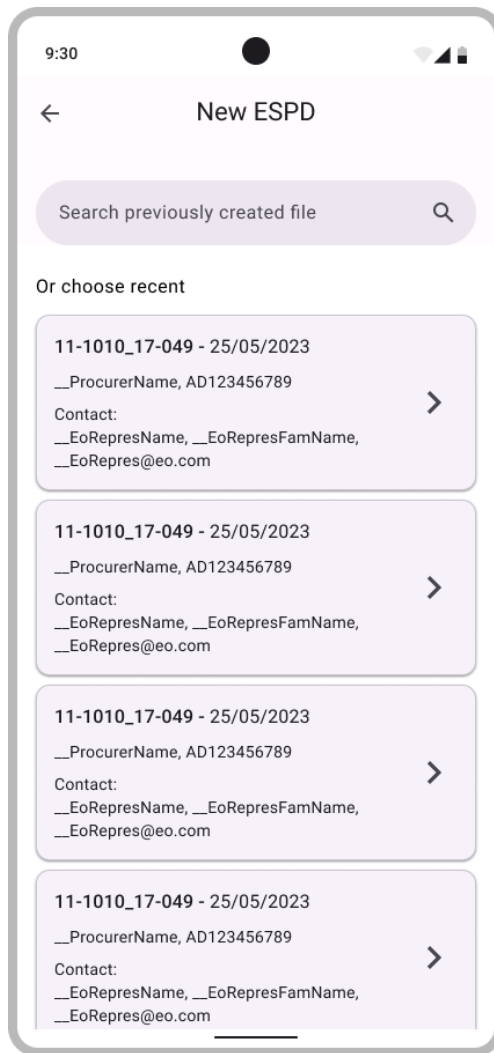
Η δημιουργία εγγράφου αποτελείται από δύο ή τρεις οθόνες, ανάλογα τις επιλογές του χρήστη. Με την επιλογή δημιουργίας ο χρήστης μεταφέρεται σε καινούρια οθόνη όπου μπορεί να επιλέξει τη μέθοδο δημιουργίας του εγγράφου, δηλαδή *self-contained* ή *regulated* και μετά για να προχωρήσει σε επόμενο βήμα πρέπει να επιλέξει είτε να δημιουργήσει έγγραφο από το μηδέν, είτε να επαναχρησιμοποιήσει ένα ήδη υπάρχον. Σημειώνεται, ότι σε κανένα σενάριο δε ζητείται από τον χρήστη να εισάγει ολόκληρο αρχείο από τη μνήμη της συσκευής ή από άλλο σημείο (Εικόνα 23).





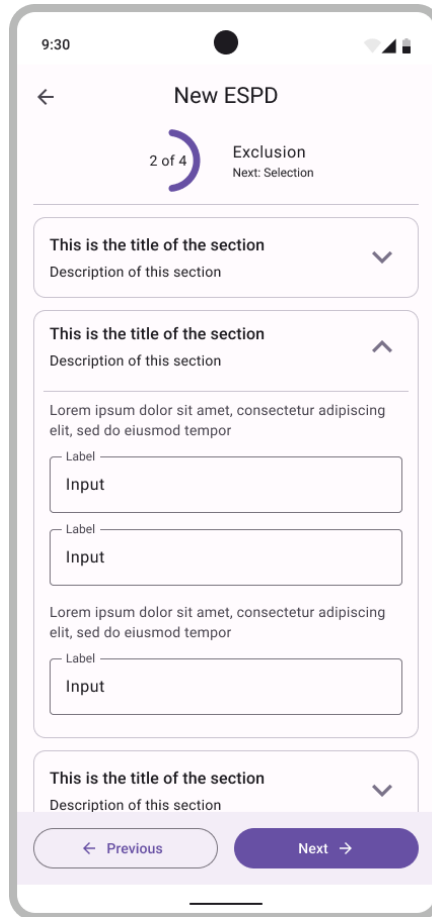
*Εικόνα 23: Αρχή δημιουργίας εγγράφου*

Σε περίπτωση που ο χρήστης επιλέξει να επαναχρησιμοποιήσει ήδη υπάρχον έγγραφο τότε μεταφέρεται στην αντίστοιχη οθόνη για να το κάνει (Εικόνα 24).



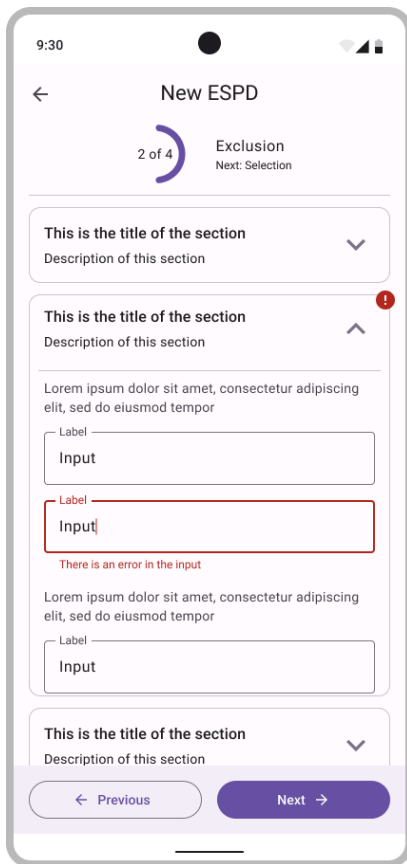
Εικόνα 24: Επιλογή υπάρχοντος αρχείου

Σε διαφορετική περίπτωση ο χρήστης μεταφέρεται στην τελική οθόνη, όπου και θα πρέπει να εισάγει όλες τις πληροφορίες που θέλει να περιέχει το έγγραφο (Εικόνα 25).

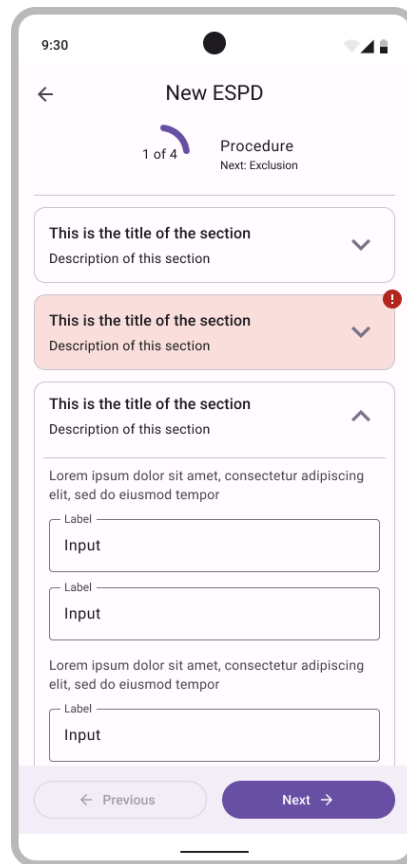


Εικόνα 25: Δημιουργία εγγράφου

Για πιθανές λανθασμένες εισόδους του χρήστη θα πρέπει να ενημερωθεί κατάλληλα (Εικόνα 26, Εικόνα 27).

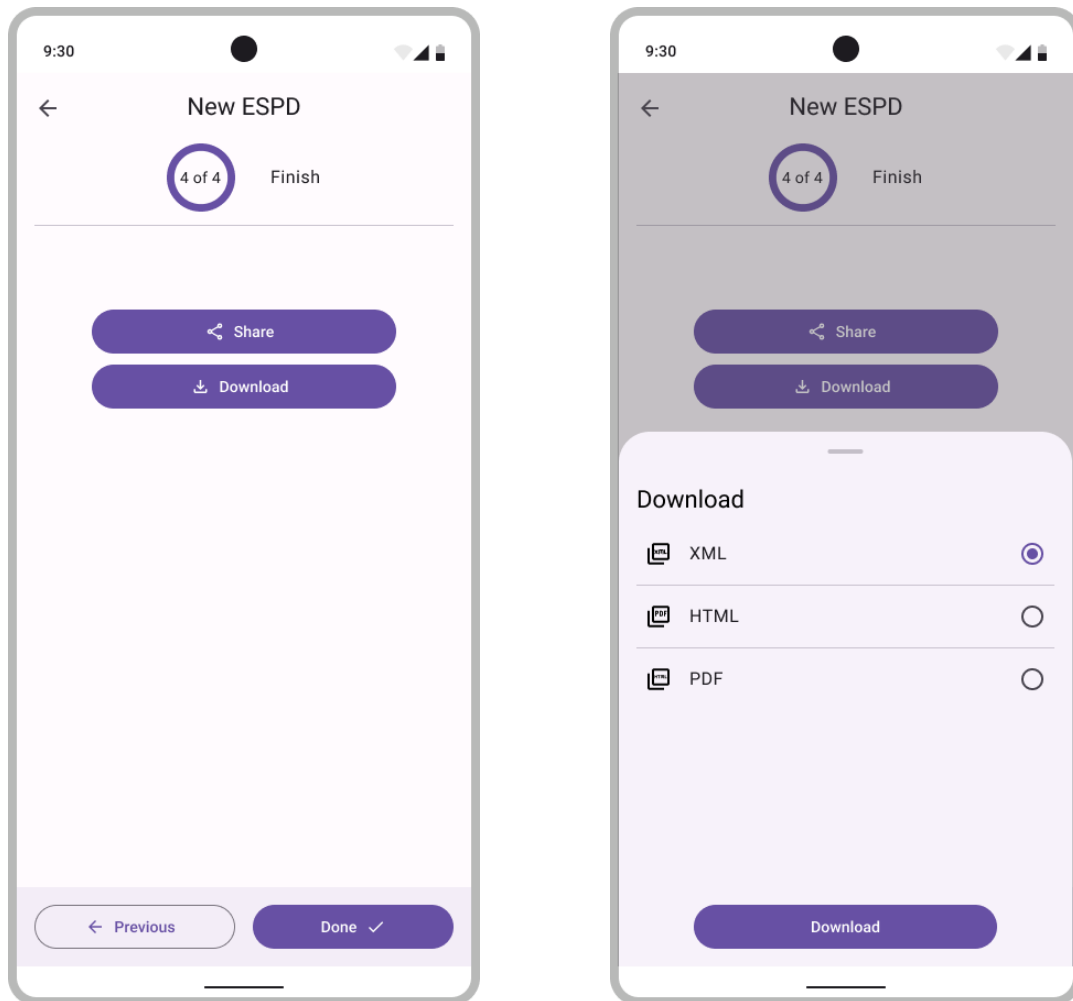


Εικόνα 26: Λανθασμένη εισαγωγή - συγκεκριμένο πεδίο



Εικόνα 27: Λανθασμένη εισαγωγή - Ολόκληρο παράρτημα

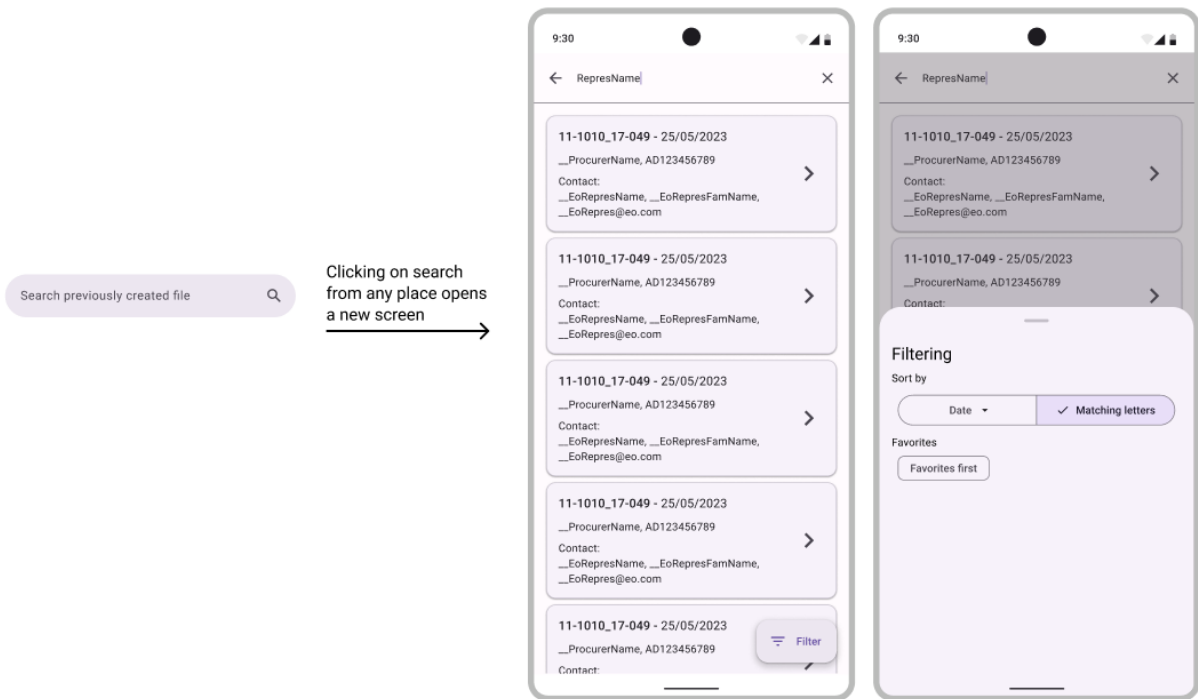
Στο τελικό βήμα ο χρήστης μπορεί να κατεβάσει το έγγραφο στη συσκευή του ή να το διαμοιραστεί (Εικόνα 28).



Εικόνα 28: Τελικό θήμα - επιλογή κατεβάσματος αρχείου

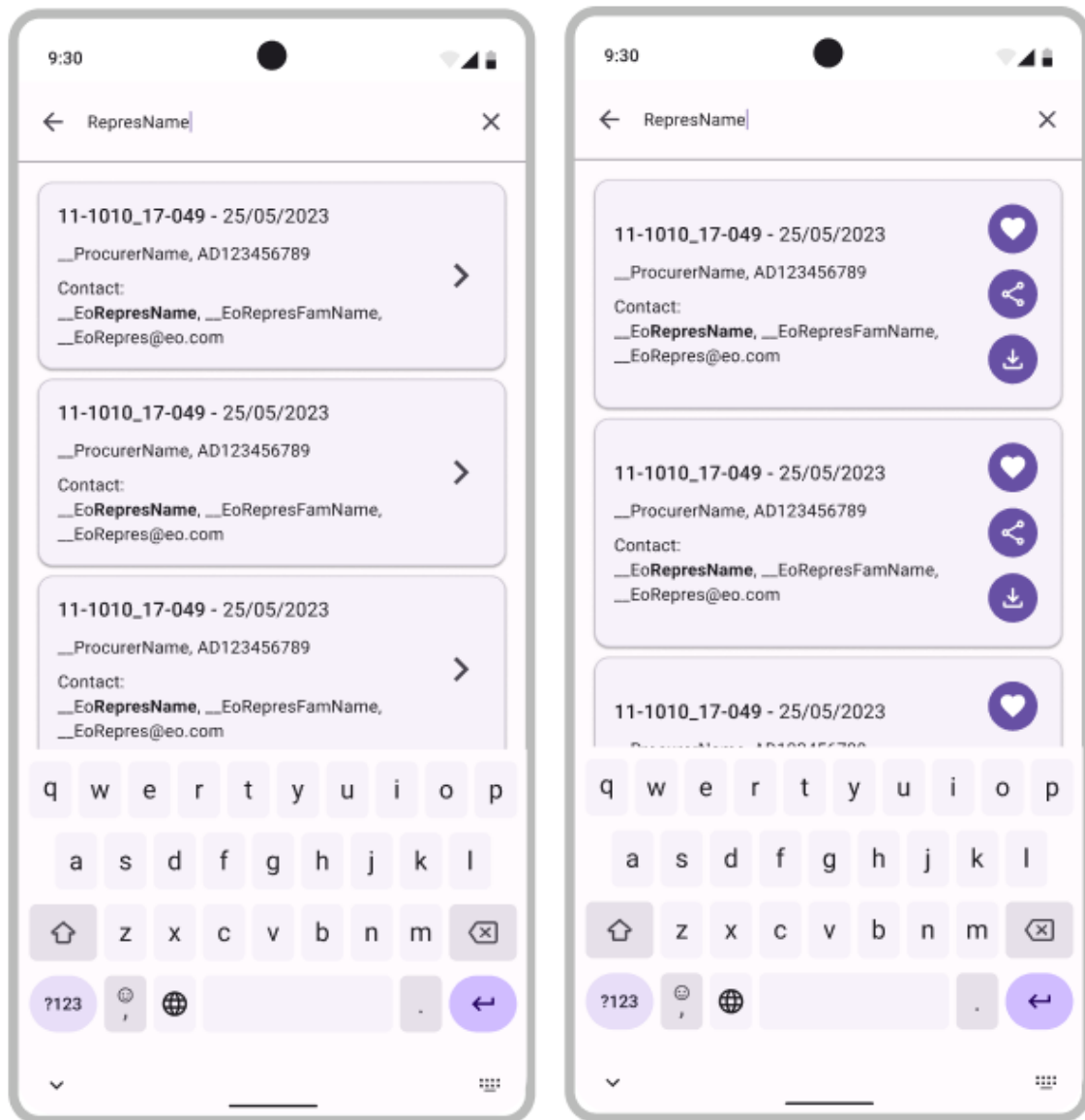
## 5.9 Αναζήτηση εγγράφου

Για την πραγματοποίηση της αναζήτησης ο χρήστης πατώντας το αντίστοιχο κουμπί στην οθόνη θα μεταφέρεται σε διαφορετική οθόνη όπου θα έχει τη δυνατότητα να αναζητήσει βάση κειμένου ή ημερομηνίας (Εικόνα 29, Εικόνα 30).



Εικόνα 29: Αναζήτηση εγγράφου

Επίσης έχει σημασία από ποιο σημείο ο χρήστης έφτασε στην οθόνη αναζήτησης. Αυτό γιατί τα previews των εγγράφων έχουν διαφορετική εμφάνιση ανάλογα την οθόνη. Στην οθόνη του ιστορικού κάνε έγγραφο έχει τρεις ενέργειες που προσφέρει, ενώ στην οθόνη επιλογής για επαναχρησιμοποίηση εγγράφου δεν υπάρχει καμία ενέργεια πέρα από την επιλογή του εγγράφου.



Εικόνα 30: Διαφορετικές εμφανίσεις για αποτελέσματα αναζήτησης

### 5.10 Ανασκόπηση σχεδίασης οθονών

Η σχεδίαση των οθονών πέρα από το UI περιέχει και το UX, την εμπειρία του χρήστη. Σημαντικό κομμάτι της διαδικασίας ήταν να αναγνωριστούν οι ενέργειες που πρέπει να μπορεί να εκτελέσει ο χρήστης και να αυτοματοποιηθούν όσες το δυνατόν περισσότερες από αυτές. Το καθαρό και ευανάγνωστο UI επίσης ήταν κάτι στο οποίο έπρεπε να δοθεί προσοχή. Σημαντικές πτυχές της σχεδίασης της διεπαφής είναι ο απλός σχεδιασμός, η αφηγηματική χρήση του χώρου και το mobile-friendly design. Λόγω του περιορισμένου χώρου, η γραφική διεπαφή πρέπει να είναι απλή και ευανάγνωστη. Οι εικόνες, τα κείμενα και τα κουμπιά πρέπει να τοποθετούνται στρατηγικά ώστε να δομείται μία λογικά σωστή δομή. Ο σχεδιασμός πρέπει να είναι αφηγηματικός. Ο χρήστης πρέπει να μπορεί να κατανοήσει την ιστορία και να αλληλεπιδρά με τα αντικείμενα (components) χωρίς να νιώθει σύγχυση. Η σχεδίαση πρέπει να είναι ευέλικτη και προσαρμοσμένη στο μέγεθος των οθονών των κινητών συσκευών. Έτσι με σωστές αρχές για τη δόμηση του UI έχουμε το βέλτιστο αποτέλεσμα. Από την άλλη η εμπειρία χρήστη (UX) δίνει έμφαση στη γρήγορη απόκριση, το mobile navigation, το mobile design και τη χρήση αφήχ. Ο χρήστης περιμένει αμεσότητα στις αλληλεπιδράσεις του, έτσι η εφαρμογή πρέπει να ανταποκρίνεται γρήγορα στις ενέργειές του. Η πλοήγηση πρέπει να είναι απλή

και ευέλικτη, χρησιμοποιώντας συχνά εικονίδια και μενού τα οποία να είναι ευανάγνωστα και εύκολα στη χρήση. Η κίνηση μπορεί να χρησιμοποιηθεί για να κατευθύνει το μάτι του χρήστη και να τον βοηθήσει να κατανοήσει τις αλληλεπιδράσεις. Ωστόσο, πρέπει να χρησιμοποιείται με διακριτικό τρόπο ώστε να μην προκαλεί σύγχυση. Η αφή είναι το κύριο μέσο αλληλεπίδρασης στα κινητά. Ο σχεδιασμός πρέπει να λαμβάνει υπόψη την ακρίβεια και την ευαισθησία της αφής για να παρέχει μια άνετη εμπειρία. Αντικείμενα θα πρέπει να απέχουν σωστή απόσταση μεταξύ τους ειδικά όταν αυτά έχουν τη δυνατότητα να πατηθούν για να γίνει κάποια ενέργεια.

Θα μπορούσε κάποιος να αναλώσει πολλές ώρες στην ανάλυση UI/UX για οποιοδήποτε έργο. Αυτό δεν ήταν κάτι δυνατό λόγω του χρονικού πλαισίου στο οποίο η παρούσα διπλωματική εργασία θα έπρεπε να έχει ολοκληρωθεί. Είναι γνωστό ότι για την ανάπτυξη εφαρμογών η Google<sup>2</sup> έχει δημιουργήσει ένα σύνολο από οδηγίες ώστε να βοηθήσει τους προγραμματιστές να σχεδιάζουν ευκολότερα οθόνες. Οι οδηγίες αυτές είναι γνωστές ως Material 3<sup>3</sup>. Έτσι οι παραπάνω οθόνες έχουν βασιστεί σε αυτό και στα δομικά μέρη που προτείνει. Κάθε κουμπί, κείμενο, πεδίο εγγραφής, μενού, τίτλοι κτλ είναι σχεδιασμένα βάση κοινά αποδεκτών κανόνων UI/UX. Στην ιστοσελίδα του Material 3 μπορεί κάποιος να βρει επεκτάσεις για το Figma οι οποίες βοηθούν στο σχεδιασμό των οθονών. Αυτός ήταν και ένα λόγος της επιλογής του συγκεκριμένου εργαλείου που δεν αναφέρθηκε στο αντίστοιχο κεφάλαιο.

---

<sup>2</sup> Google - <https://www.google.gr/>

<sup>3</sup> Material 3 - <https://m3.material.io/>



## 6 Υλοποίηση εφαρμογής

### 6.1 Εισαγωγή

Για την υλοποίηση της εφαρμογής θα πρέπει αρχικά να αποφασιστεί η πλατφόρμα πάνω στην οποία θα γραφτεί ο κώδικας. Η πλατφόρμα αυτή συνήθως ορίζει και τη γλώσσα προγραμματισμού. Υπάρχουν διάφορες πλατφόρμες για τη δημιουργία εφαρμογών κινητών συσκευών. Κάποιες υποστηρίζουν τη δυνατότητα με ένα *codebase* να παράγουν εφαρμογές και για τα δύο πιο διαδεδομένα λογισμικά, Android και iOS. Στην παρούσα διπλωματική εργασία ο σκοπός είναι να υλοποιηθεί εφαρμογή σε Android έτσι δε θα δοθεί έμφαση σε εργαλεία που υποστηρίζουν τέτοιες δυνατότητες.

Με μία αναζήτηση στα έγγραφα της Google σχετικά με την ανάπτυξη εφαρμογών σε Android, είναι φανερό ότι το προτεινόμενο εργαλείο είναι το Android Studio<sup>4</sup> και η προτεινόμενη γλώσσα προγραμματισμού η Kotlin<sup>5</sup>.

### 6.2 Αρχιτεκτονική

Όταν αναφερόμαστε στον όρο αρχιτεκτονική στον προγραμματισμό για Android [1], εννοούμε τον τρόπο με τον οποίο οργανώνεται ο κώδικας και οι δομές του. Η επιλογή της κατάλληλης αρχιτεκτονικής είναι σημαντική για τη συντήρηση και τη διαχείριση του κώδικα αλλά και την επεκτασιμότητα και την αποδοτικότητα της εφαρμογής.

Για τη συγγραφή του κώδικα θα ακολουθηθούν οι αρχές που αναφέρονται στο σύγγραμμα του R. C Martin [2], γνωστές ως SOLID. Οι αρχές SOLID αποτελούνται από 5 αρχές που αναφέρονται στη συνέχεια.

1. Single Responsibility Principle (SRP). Βάση αυτής της αρχής, κάθε κλάση θα πρέπει να έχει μία ευθύνη. Άρα και έναν λόγο ύπαρξης. Έτσι σε περίπτωση που χρειαστεί αλλαγή να είναι γνωστή και εύκολα κατανοήσιμη η λειτουργία της. Αυτό βοηθάει στη διατήρηση του κώδικα και τη συνεργασία μεταξύ προγραμματιστών.
2. Open-Closed Principle (OCP). Σύμφωνα με αυτήν την αρχή, μια κλάση πρέπει να είναι ανοικτή για επέκταση αλλά κλειστή για τροποποίηση. Αυτό σημαίνει ότι νέες λειτουργίες πρέπει να προστίθενται χωρίς να αλλάζει ο υπάρχον κώδικας.
3. Liskov Substitution Principle (LSP). Η αρχή αυτή εξηγεί πως πρέπει να είναι δυνατή η χρήση παιδιών κλάσεων (*derived class*), χωρίς να αλλοιώνεται η σωστή λειτουργία και η ορθότητα του προγράμματος.
4. Interface Segregation Principle (ISP). Αυτή η αρχή δηλώνει ότι μια κλάση δεν πρέπει να είναι υποχρεωμένη να υλοποιεί μεθόδους που δεν χρειάζεται. ένα *interface* πρέπει να είναι συνεκτικό και κάθε κλάση πρέπει να υλοποιεί μόνο αυτά που την ενδιαφέρουν.
5. Dependency Inversion Principle (DIP). Αυτή η αρχή αναφέρει ότι οι υψηλού επιπέδου μονάδες δεν πρέπει να εξαρτώνται απευθείας από χαμηλού επιπέδου μονάδες. Και οι δύο πρέπει να εξαρτώνται από αφαιρετικά στρώματα. Συνήθως με την έννοια αφαιρετικά στρώματα αναφερόμαστε σε *interfaces*. Χαμηλού επιπέδου μονάδες μπορεί να είναι κλάσεις που έχουν να κάνουν με τη σύνδεση στο δίκτυο, ενώ υψηλού επιπέδου θεωρούνται κλάσεις που σχετίζονται με το UI.

---

<sup>4</sup> Android Studio - <https://developer.android.com/studio>

<sup>5</sup> Kotlin - <https://kotlinlang.org/>

Υπάρχουν διάφορες αρχιτεκτονικές που μπορεί κάποιος να χρησιμοποιήσει όταν γράφει κώδικα, αλλά δύο από τις πιο διαδεδομένες είναι η Model-View-Controller (MVC) και η **Model-View-ViewModel (MVVM)**.

1. Model-View-Controller (MVC):
  - **Model:** Αναφέρεται στην επιχειρησιακή λογική και τα δεδομένα της εφαρμογής.
  - **View:** Είναι υπεύθυνο για την παρουσίαση των δεδομένων και τον χειρισμό των γραφικών στοιχείων της εφαρμογής.
  - **Controller:** Είναι υπεύθυνος για τον έλεγχο των αλληλεπιδράσεων με τον χρήστη και τη διαχείριση των αιτημάτων.
2. Model-View-ViewModel (MVVM):
  - **Model:** Παρέχει τα δεδομένα και την επιχειρησιακή λογική.
  - **View:** Είναι υπεύθυνη για την παρουσίαση των δεδομένων και τη διεπαφή χρήστη.
  - **ViewModel:** Συνδέει το Model με τη View και χειρίζεται την επικοινωνία μεταξύ τους. Επίσης, παρέχει μεθόδους για την ενημέρωση των δεδομένων.

Οι παραπάνω αρχιτεκτονικές συνδυάζονται συχνά με άλλα πρότυπα σχεδίασης όπως το Dependency Injection για τη διαχείριση των εξαρτήσεων και Clean Code για να παραχθεί εύκολα συντηρήσιμος και κατανοητός κώδικας. Θα χρησιμοποιηθεί **MVVM με Clean Code και Dependency Injection**. Οι τρεις αυτές τεχνικές θα πρέπει να χρησιμοποιηθούν παράλληλα ώστε να πραχθεί το βέλτιστο αποτέλεσμα. Ο συνδυασμός καλής πρακτικής στη σχεδίαση και κατασκευή του κώδικα με μια καλή αρχιτεκτονική μπορεί να οδηγήσει σε εφαρμογές που είναι ευανάγνωστες, συντηρήσιμες και ευέλικτες.

Το Dependency Injection (DI) είναι μια τεχνική στον προγραμματισμό λογισμικού που στοχεύει στη μείωση των συνδέσεων (dependencies) μεταξύ των κλάσεων. Συγκεκριμένα, η τεχνική Dependency Injection επικεντρώνεται στην εισαγωγή (injection) των εξαρτήσεων (dependencies) μιας κλάσης από έναν εξωτερικό προμηθευτή, αντί να τις δημιουργεί η ίδια η κλάση.

Οι βασικοί στόχοι της **Dependency Injection** είναι η αποφυγή σκληρών συνδέσεων (hard-coded connections) μεταξύ κλάσεων και η προώθηση της επαναχρησιμοποίησης κώδικα. Επιπλέον, αυξάνει το testability και τη συντηρησιμότητα του κώδικα. Το Dependency Injection συνήθως συνδυάζεται με τη χρήση διαχειριστών εξαρτήσεων (Dependency Injection Containers ή IoC Containers) που αναλαμβάνουν τη διαχείριση και παροχή των εξαρτήσεων. Ένα δημοφιλές παράδειγμα είναι το Spring Framework στον κόσμο της Java και το Hilt<sup>6</sup> και το Koin<sup>7</sup> για την Kotlin.

1. Constructor Injection:
  - Στην Constructor Injection, οι εξαρτήσεις περνούν μέσω του constructor της κλάσης. Αυτό συνήθως γίνεται κατά τη δημιουργία του αντικειμένου.
2. Setter Injection:
  - Στην Setter Injection, οι εξαρτήσεις περνούν μέσω setter μεθόδων της κλάσης. Αυτό επιτρέπει την αλλαγή των εξαρτήσεων μετά τη δημιουργία του αντικειμένου.
3. Method Injection:
  - Στην Method Injection, οι εξαρτήσεις περνούν μέσω μεθόδων κατά την κλήση της μεθόδου.

---

<sup>6</sup> Hilt - <https://dagger.dev/hilt/>

<sup>7</sup> Koin - <https://insert-koin.io/>

Το **Clean code** αναφέρεται σε έναν τρόπο προγραμματισμού που έχει στόχο να παράγει κώδικα που είναι ευανάγνωστος, κατανοητός, συντηρήσιμος και εύκολα δοκιμάσιμος. Η έννοια αυτή έχει προταθεί και αναλυθεί από τον Martin [2]. Ορισμένες βασικές αρχές του Clean Code περιλαμβάνουν:

1. Κατανοητή Ονομασία:
  - Τα ονόματα μεταβλητών, συναρτήσεων και κλάσεων πρέπει να είναι περιγραφικά και να αντικατοπτρίζουν τη λειτουργικότητα τους.
  - Η καλή ονομασία βοηθάει τον αναγνώστη να κατανοήσει τον κώδικα χωρίς να χρειάζεται να διαβάσει τον ίδιο τον κώδικα.
2. Μικρές Συναρτήσεις:
  - Κάθε συνάρτηση πρέπει να εκτελεί μια μόνο λειτουργία και να είναι σύντομη.
  - Μικρές συναρτήσεις είναι πιο εύκολες στην κατανόηση, στην αποσφαλμάτωση και στη συντήρηση.
3. Καμία Διπλή Σημασία (No Duplication):
  - Η επαναχρησιμοποίηση του κώδικα είναι σημαντική. Εάν έχετε δύο τμήματα κώδικα που κάνουν το ίδιο πράγμα, πρέπει να συγχωνευτούν.
4. Μείωση Πολυπλοκότητας:
  - Η πολυπλοκότητα του κώδικα πρέπει να μειώνεται όσο το δυνατόν περισσότερο.
  - Χρήση καλών δομών δεδομένων και αλγορίθμων που είναι ευανάγνωστοι και αποτελεσματικοί.
5. Σχόλια και Τεκμηρίωση:
  - Χρησιμοποιείτε σχόλια όταν είναι απαραίτητα για να εξηγήσετε τις πιο περίπλοκες ή σημαντικές πτυχές του κώδικα.
  - Η καλή τεκμηρίωση βοηθάει στην κατανόηση του κώδικα, ειδικά από άλλους προγραμματιστές που μπορεί να εργαστούν πάνω σε αυτόν.
6. Δοκιμές:
  - Ενθαρρύνεται η συχνή δοκιμή του κώδικα (Unit Testing) για να εξασφαλιστεί η σωστή λειτουργία και για τη διατήρηση της ασφάλειας κατά την εξέλιξη του κώδικα.

Η συμμόρφωση με αυτές τις αρχές διευκολύνει τη συντήρηση, την επεκτασιμότητα και την κατανόηση του κώδικα, καθιστώντας τον πιο αποτελεσματικό και εύκολο στη διαχείριση.

### 6.3 Version control

Η χρήση ενός συστήματος ελέγχου εκδόσεων (version control) σε μια εφαρμογή είναι εργαλείο ζωτικής σημασίας για τη διαχείριση του κώδικα. Τα πλεονεκτήματα του version control αναδεικνύονται σε πολλά επίπεδα: Πρώτον, παρέχει μια πλήρη εικόνα του ιστορικού αλλαγών στον κώδικα. Κάθε μεταβολή, κάθε προσθήκη ή διαγραφή αρχείου και κάθε άλλη τροποποίηση καταγράφεται με σαφήνεια, επιτρέποντας στην ομάδα ανάπτυξης να παρακολουθεί την εξέλιξη του κώδικα με τον χρόνο. Δεύτερον, εξασφαλίζει μια οργανωμένη διαχείριση των αλλαγών. Οι προγραμματιστές μπορούν να καταγράψουν, να περιγράψουν και να προσδιορίσουν τον σκοπό των αλλαγών που πραγματοποιούν, δημιουργώντας ένα δομημένο σύστημα διαχείρισης αλλαγών. Τρίτον, διευκολύνει τη συνεργασία μεταξύ πολλαπλών προγραμματιστών. Πολλοί προγραμματιστές μπορούν να εργαστούν ταυτόχρονα στον κώδικα, και το σύστημα αναλαμβάνει τον συγχρονισμό των αλλαγών τους, επιτρέποντας την αποτελεσματική συνεργασία. Επίσης, παρέχει τη δυνατότητα εύκολης αναστροφής σε προηγούμενες εκδόσεις του κώδικα, προσφέροντας τη δυνατότητα αποκατάστασης σε περιπτώσεις προβλημάτων ή σφαλμάτων. Τέλος, οι ετικέτες (tags) επιτρέπουν την σηματοδότηση σημαντικών σημείων στην ανάπτυξη, όπως σταθερές εκδόσεις της εφαρμογής, διευκολύνοντας την παρακολούθηση και τη διαχείριση των εκδόσεων. Έτσι, το σύστημα ελέγχου

εκδόσεων αποτελεί θεμέλιο πυλώνα για την αποτελεσματική ανάπτυξη λογισμικού, προσφέροντας διαφάνεια, οργάνωση, συνεργασία και ασφάλεια.

Αντίθετα, η απόφαση να αγνοηθεί η χρήση συστήματος ελέγχου εκδόσεων σε ένα έργο ανάπτυξης λογισμικού έχει σημαντικές επιπτώσεις σε διάφορους τομείς. Η έλλειψη δομημένης και τακτοποιημένης καταγραφής των αλλαγών στον κώδικα δημιουργεί προβλήματα στην ανακατασκευή του ιστορικού ενώ ταυτόχρονα καθιστά περίπλοκη την αποτελεσματική αντιμετώπιση σφαλμάτων. Οι συγκρούσεις στον κώδικα είναι συχνές όταν πολλοί προγραμματιστές εργάζονται ανεξάρτητα χωρίς συντονισμό, δημιουργώντας δυσκολίες στη διαχείριση και ενσωμάτωση των αλλαγών. Η ανεύρεση και επίλυση σφαλμάτων γίνεται περίπλοκη λόγω της έλλειψης σαφούς ιστορικού αλλαγών. Επιπλέον, η απουσία συστήματος ελέγχου εκδόσεων δυσχεραίνει τη συνεργασία σε μεγάλες ομάδες προγραμματιστών, μειώνοντας την αποτελεσματικότητα. Το ζήτημα της ασφάλειας εντείνεται, καθώς η δυσκολία επαναφοράς σε προηγούμενες εκδόσεις μπορεί να αποδειχθεί επικίνδυνη σε περιπτώσεις κακόβουλων επιθέσεων ή προβλημάτων ασφαλείας. Συνολικά, η εφαρμογή ενός συστήματος ελέγχου εκδόσεων αποτελεί αναγκαίο βήμα για τη διασφάλιση οργάνωσης, συνεργασίας και ασφάλειας στη διαδικασία ανάπτυξης λογισμικού.

Για την εν λόγω υλοποίηση θα μπορούσε να παραλειφθεί η χρήση version control (git), αφού θα ήμουν ο μόνος προγραμματιστής που θα δούλευε στο έργο. Έτσι δε θα υπήρχαν συγκρούσεις (conflicts) σε σημεία του κώδικα που έχουν τροποποιηθεί από πολλούς προγραμματιστές. Βέβαια τα οφέλη από την παροχή ιστορικού και εύκολης αναστροφής σε παλαιές εκδόσεις παραμένουν. Έτσι αποφασίστηκε να χρησιμοποιηθεί ένα τέτοιο σύστημα. Αυτό που επιλέχθηκε είναι το BitBucket<sup>8</sup>. Η ανάπτυξη του κώδικα επίσης θα πραγματοποιούνταν ανά feature. Δηλαδή login/register, δημιουργία εγγράφων, home page. Κάθε feature υλοποιούνταν σε δικό του branch. Μόλις τελείωνε γινόταν merge στο develop. Μόλις τελείωσαν όλα τα βασικά feature και η εφαρμογή ήταν έτοιμη για δημοσίευση τότε τα έκανα merge και στο master.

#### 6.4 Ανάλυση δημιουργίας εγγράφων

Όπως αναφέρθηκε στο κεφάλαιο Υπάρχουσα υλοποίηση, το υπάρχον σύστημα λειτουργεί βάση αρχείων. Άρα ένα τυπικό flow είναι το εξής. Η αναθέτουσα αρχή μπαίνει στην ιστοσελίδα, επιλέγει χρήση υπάρχοντος αρχείου και ανεβάζει το αντίστοιχο αρχείο, ή επιλέγει δημιουργία και μεταφέρεται σε καινούρια οθόνη με όλα τα πεδία έτοιμα για συμπλήρωση. Στο τέλος της διαδικασίας το αρχείο κατεβαίνει στη συσκευή του χρήστη. Ακολούθως ο οικονομικός φορέας, όταν θέλει να δημιουργήσει μία απάντηση για κάποιο διαγωνισμό θα πρέπει να ανεβάσει το αρχείο που δημιούργησε η αναθέτουσα αρχή και να συμπληρώσει όλα τα απαιτούμενα πεδία. Στην web εφαρμογή υπάρχει και η δυνατότητα ένας οικονομικός φορέας να δημιουργήσει απάντηση από το μηδέν, χωρίς να ανεβάσει αρχείο και φαινομενικά χωρίς την ύπαρξη κάποιου διαγωνισμού. Αυτό δε θεωρείται αποδεκτό σενάριο στα πλαίσια της υλοποίησης, από τη στιγμή που το σύστημα μπορεί να συμπληρώσει αυτόματα αυτά τα πεδία και η απάντηση σίγουρα θα πρέπει να αφορά κάποιο διαγωνισμό.

Το flow αυτό θα υλοποιηθεί με μερικές αλλαγές, ώστε να καλύπτει τα κριτήρια που χρειάζεται. Σωστό UI/UX με την έννοια ότι ο χρήστης θα πρέπει με ταχύτητα και ευκολία να εκτελεί τις ενέργειες που θέλει. Θα πρέπει η πληροφορία να είναι γρήγορα και εύκολα διαθέσιμη. Πεδία που μπορούν να είναι προσυμπληρωμένα πρέπει να είναι, ώστε επιτυγχάνεται βέλτιστη απόδοση από πλευράς ταχύτητας ολοκλήρωσης της ενέργειας του θέλει να εκτελέσει ο χρήστης και όσο το δυνατό λιγότερη τριβή του χρήστη με συμπλήρωση πεδίων τα οποία ήδη έχει συμπληρώσει. Ο κώδικας θα πρέπει να είναι

---

<sup>8</sup> BitBucket - <http://bitbucket.org/>

γραμμένος με τρόπο ώστε να μπορεί να αλλαχθεί εύκολα, για παράδειγμα θα πρέπει να μπορούν να προστεθούν και να αφαιρεθούν πεδία δυναμικά. Το UI δε θα πρέπει να μην εξαρτάται από τα δεδομένα αυτά και να λειτουργεί σωστά με οτιδήποτε είσοδο του έρθει.

Το παραπάνω flow, θα πρέπει λοιπόν να επανασχεδιαστεί ώστε να μη χρειάζεται ο χρήστης να μεταχειρίζεται raw αρχεία. Θα πρέπει όλα να μπορούν να γίνουν μέσω της εφαρμογής. Δυστυχώς η μέθοδός μου ίσως καταναλώνει περισσότερους πόρους στον server που διαχειρίζεται το API. Παρόλα αυτά, αυτό θα συμβεί μόνο σε ένα σενάριο και θεωρείται είναι κάτι αναγκαίο βάση των αρχών του σωστού UX. Πληροφορία που μπορεί να σωθεί τοπικά και remotely (περισσότερα στο κεφάλαιο Αποθήκευση δεδομένων χρήστη) είναι αντικείμενα σε μορφή json. Τέτοια αντικείμενα ανταλλάσσονται μεταξύ του API και της εφαρμογής. Έτσι εάν αποθηκευτεί το json που έστειλε η αναθέτουσα αρχή στο API για τη δημιουργία του εγγράφου (request), τότε αυτόματα αυτό σημαίνει ότι η εφαρμογή θα έχει και ότι χρειάζεται για να παράξει αυτό το έγγραφο όταν το ζητήσει ο χρήστης (ως raw αρχείο) ή όταν χρειαστεί κατά τη δημιουργία μίας απάντησης από κάποιον άλλο χρήστη που δρα ως οικονομικός φορέας. Έτσι δε χρειάζεται να αποθηκεύονται τοπικά τα raw έγγραφα όπως χρειάζεται για την web εφαρμογή.

Σε περίπτωση που ένας οικονομικός φορέας θέλει να δημιουργήσει μία απάντηση, όπως ήδη αναφέρθηκε, υφίσταται ο περιορισμός να πρέπει να διαλέξει έναν διαγωνισμό που ήδη υπάρχει, ή να επαναχρησιμοποιήσει μία απάντηση που έχει ήδη δημιουργήσει. Άρα και στα δύο σενάρια, ο οικονομικός φορέας θα πρέπει να δει κάποια πεδία (κυρίως τις πληροφορίες της αναθέτουσας αρχής) να είναι προσυμπληρωμένα. Για να γίνει αυτό, στο σενάριο όπου ο οικονομικός φορέας θέλει να δημιουργήσει νέα απάντηση για κάποιο διαγωνισμό, τότε βάση του περιορισμού στο backend, πρέπει να χρησιμοποιηθεί η αποθηκευμένη πληροφορία του request που έφτιαξε η αναθέτουσα αρχή και έχει αποθηκευτεί, ώστε να ληφθεί το xml αρχείο, το οποίο θα πρέπει στη συνέχεια να σταλεί αμέσως πίσω με σκοπό να ληφθεί το μοντέλο (json) της απάντησης, το οποίο και θα πρέπει να συμπληρώσει ο οικονομικός φορέας. Στη συνέχεια όταν ο ίδιος έχει συμπληρώσει όλα τα αναγκαία πεδία, μετά από ενέργεια του χρήστη, αποθηκεύεται το συμπληρωμένο μοντέλο της απάντησης και η διαδικασία δημιουργίας του response θεωρείται ολοκληρωμένη.

Σε περίπτωση βέβαια που ο οικονομικός φορέας επιλέξει να χρησιμοποιήσει ήδη υπάρχουσα απάντηση, δεν υφίσταται κανένας περιορισμός λόγω του ότι η αναγκαία πληροφορία μπορεί να βρεθεί στη βάση δεδομένων που χρησιμοποιεί η εφαρμογή.

Συνοψίζοντας, το μόνο σενάριο κατά οποίο θα υπάρχει μεγαλύτερη καθυστέρηση σε σχέση με την υπάρχουσα υλοποίηση είναι η περίπτωση όπου ένας οικονομικός φορέας θέλει να δημιουργήσει απάντηση από το μηδέν για κάποιο διαγωνισμό. Σε κάθε άλλη περίπτωση ο χρήστης μπορεί να δει και να τροποποιήσει έγγραφα με μηδενική καθυστέρηση λόγω του ότι όλα τα μοντέλα είναι αποθηκευμένα σε ανάλογη βάση δεδομένων και προσπελάσιμα σε δευτερόλεπτα. Έτσι με τις παραπάνω αποφάσεις και υλοποίηση η mobile εφαρμογή είναι σε θέση να βελτιώσει όλες τις ενέργειες που μπορεί να εκτελέσει η τρέχουσα web εφαρμογή.

## 6.5 Υποστήριξη πολλαπλών γλωσσών εμφάνισης

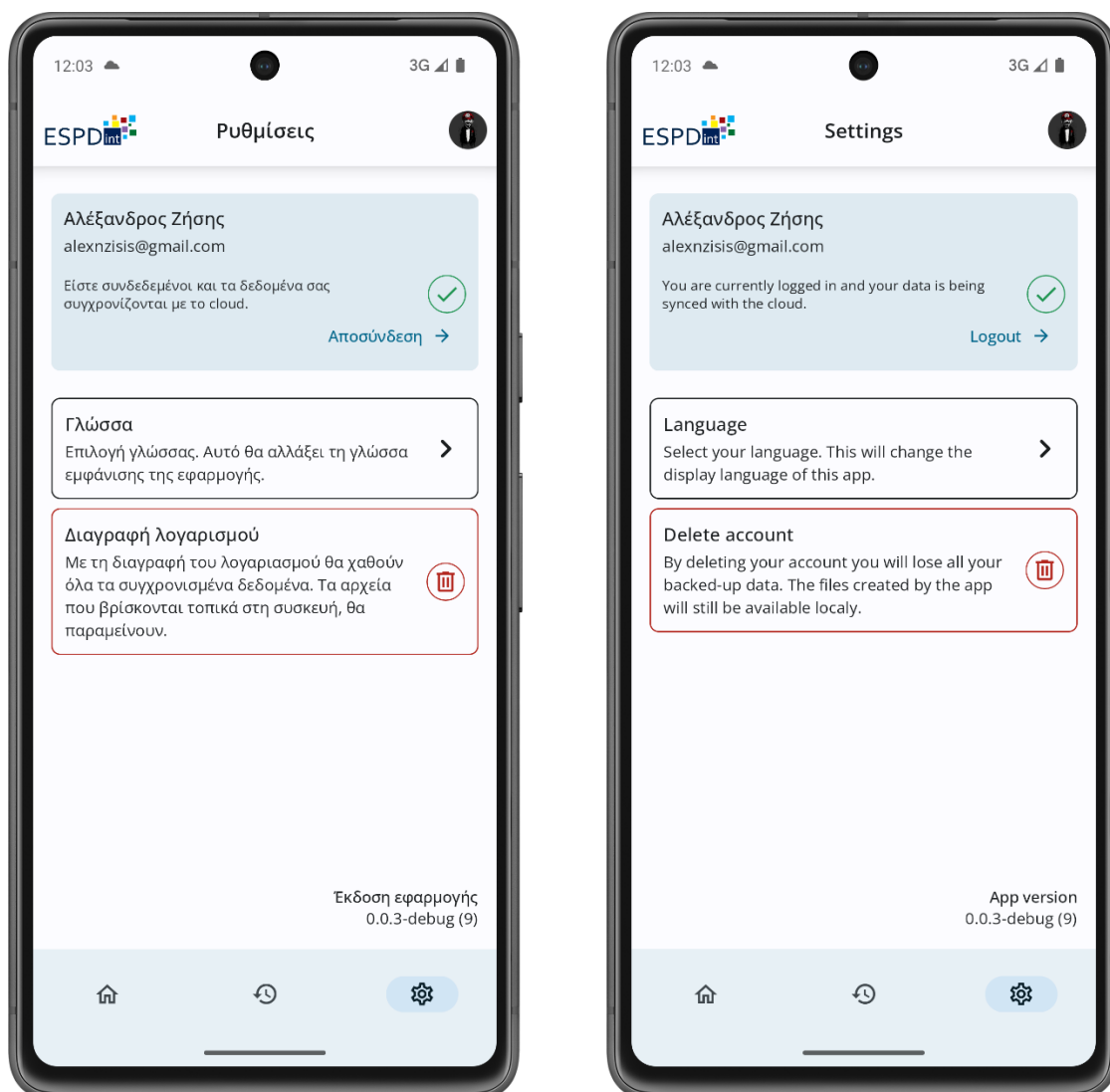
Όπως έχει αναφερθεί, η εφαρμογή θα πρέπει να υποστηρίζει Ελληνικά και Αγγλικά ως γλώσσες εμφάνισης (Εικόνα 31). Η υποστήριξη πολλαπλών γλωσσών σε μια εφαρμογή αντιπροσωπεύει έναν σημαντικό παράγοντα που συμβάλλει στην επιτυχία της, καθώς προσφέρει αρκετά πλεονεκτήματα τα οποία απευθύνονται σε ένα ευρύ κοινό χρηστών.

Πρώτον και κύριον, η υποστήριξη πολλαπλών γλωσσών επεκτείνει το κοινό στο οποίο απευθύνεται η εφαρμογή και αυξάνει την προσβασιμότητα. Επιτρέπει σε χρήστες από διάφορες περιοχές του

κόσμου να αλληλεπιδρούν με την εφαρμογή στη δική τους γλώσσα, προσφέροντάς τους μια πιο εξατομικευμένη και φιλική εμπειρία. Αυτό όχι μόνο επιτυγχάνει ποικιλομορφία στον χρήστη, αλλά επίσης συντελεί στην προσέλκυση νέων χρηστών από διαφορετικές γλωσσικές κοινότητες.

Δεύτερον, η υποστήριξη πολλαπλών γλωσσών διαδραματίζει ένα σημαντικό ρόλο στην ενίσχυση της παγκόσμιας εικόνας και φήμης της εφαρμογής. Αναδεικνύει την προορατικότητα και τη διεθνή προσανατολισμένη στρατηγική της εφαρμογής, καθιστώντας την πιο ελκυστική για διεθνείς χρήστες και πιθανούς συνεργάτες. Στην προκειμένη περίπτωση η εφαρμογή προορίζεται για χρήστης σε όλη την Ευρωπαϊκή Ένωση, οπότε η υποστήριξη γλωσσών από τις χώρες που ανήκουν σε αυτή, είναι κάτι αναγκαίο.

Η υποστήριξη πολλαπλών γλωσσών σε μια εφαρμογή προσφέρει ένα ευρύ φάσμα οφελών, από την επέκταση της αγοράς και τη βελτίωση της προσβασιμότητας μέχρι την ενίσχυση της παγκόσμιας φήμης και την ενίσχυση της διαλειτουργικότητας. Αυτή η προσέγγιση δεν αφορά μόνο την επιτυχία της εφαρμογής σε τοπικό επίπεδο στην Ελλάδα, αλλά επίσης δίνει τη δυνατότητα σε χρήστες εκτός της χώρας, να μπορούν να συμμετέχουν σε διαγωνισμούς. Η ελληνική γλώσσα θα ήταν ένα εμπόδιο το οποίο δε θα μπορούσαν να παρακάμψουν εάν δεν υπήρχε η μετάφραση στην αγγλική γλώσσα.



Εικόνα 31: Ενδεικτική οθόνη σε διαφορετική γλώσσα

Στην πράξη έχει ορισθεί ως κύρια και προεπιλεγμένη γλώσσα εμφάνισης η αγγλική. Η επιλογή αλλαγής γλώσσας υπάρχει στην οθόνη των ρυθμίσεων όπου ο χρήστης μπορεί να διαλέξει ανάμεσα σε ελληνικά και αγγλικά. Τα ονόματα των γλωσσών δεν μεταφράζονται ώστε να είναι εύκολα κατανοητά από τους χρήστες που ψάχνουν τη γλώσσα που κατανοούν.

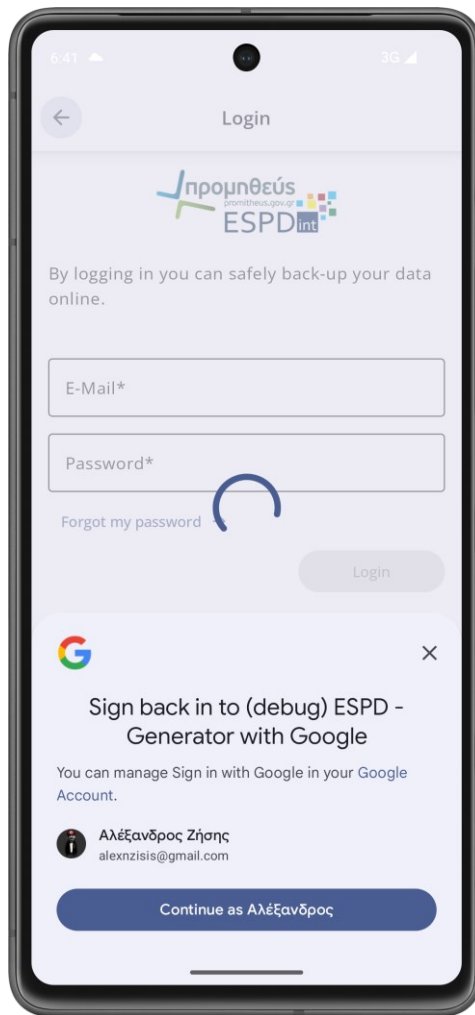
Κάτι που πρέπει να σημειωθεί είναι ότι κατά τη δημιουργία των ESPD request/response, γίνεται λήψη δεδομένων από το API. Τα δεδομένα αυτά περιέχουν την πληροφορία που θα πρέπει να μεταφραστεί σε UI components. Αυτή η πληροφορία είναι πάντα στην αγγλική γλώσσα. Η τρέχουσα υλοποίηση για την web εφαρμογή, μεταφράζει τα κείμενα βάση εκτελώντας μία διαφορετική κλήση δικτύου και αντιστοιχίζοντας τα IDs που περιέχονται στις δύο αυτές κλήσεις δικτύου. Το γεγονός αυτό, προσθέτει μεγάλη πολυπλοκότητα και χρόνο ανάπτυξης, έτσι αποφασίστηκε να μην υλοποιηθεί. Σε μία παραγωγική υλοποίηση κάτι τέτοιο δε θα έπρεπε να υφίσταται, δηλαδή το API θα έπρεπε να δέχεται παράμετρο για τη γλώσσα και να επιστρέφει σωστά την πληροφορία. Το να βασίζεται η εφαρμογή σε δύο κλήσεις δικτύου για την ίδια πληροφορία, είναι σχεδιαστικό λάθος και μελλοντικά θα πρέπει να διορθωθεί από την πλευρά του API.

## 6.6 Αυθεντικοποίηση χρηστών

Για την αυθεντικοποίηση χρηστών θα χρησιμοποιηθεί το Firebase Authentication<sup>9</sup> [7]. Σε περίπτωση που ο χρήστης έχει Google λογαριασμό τότε η διαδικασία σύνδεσης είναι απλή. Ανοίγει ένα bottom sheet στο οποίο ο χρήστης επιλέγει και αποδέχεται τη σύνδεση με google (Εικόνα 32) και τον ανακατευθύνεται στην αρχική οθόνη.

---

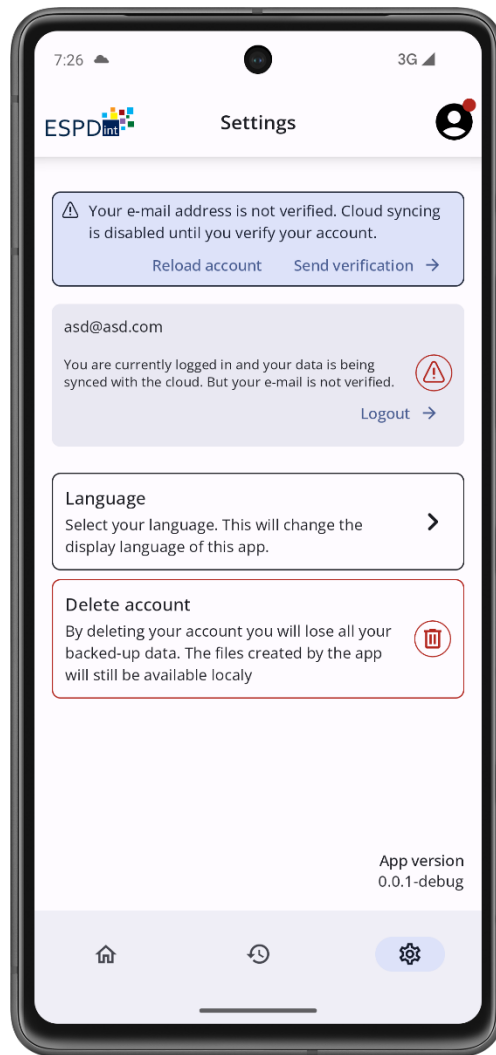
<sup>9</sup> Firebase Authentication - <https://firebase.google.com/docs/auth>



Εικόνα 32: Σύνδεση με Google

Σε αυτή την περίπτωση φυσικά δεν υπάρχει ανάγκη για όνομα χρήστη και κωδικό, το οποίο διευκολύνει την είσοδο του χρήστη στην εφαρμογή. Στον χρήστη δίδεται και η επιλογή να δημιουργήσει νέο account ή/και να συνδεθεί με όνομα χρήστη και κωδικό, σε περίπτωση που δεν θέλει να χρησιμοποιήσει τον λογαριασμό Google ή δεν έχει. Σε κάθε περίπτωση, τα δεδομένα που χρησιμοποιούνται μετά τη σύνδεση είναι η εικόνα του λογαριασμού, το ονοματεπώνυμο, η κατάσταση επιβεβαίωσης του λογαριασμού και το μοναδικό αναγνωριστικό. Ο λογαριασμός κάθε χρήστη μετά από εγγραφή με e-mail και κωδικό, δημιουργείται άμεσα, αλλά η κατάστασή του είναι μη επιβεβαιωμένη. Πρακτικά θα πρέπει να επιβεβαιωθεί ότι το e-mail ανήκει όντως στον χρήστη. Εάν διαπιστωθεί ότι ο λογαριασμός του χρήστη δεν είναι επιβεβαιωμένος (Εικόνα 33), τότε δεν υπάρχει κάποια αρνητική συνέπεια.





Εικόνα 33: Μη επιβεβαιωμένος λογαριασμός

Όπως φαίνεται στην Εικόνα 33, ο χρήστης ενημερώνεται για την κατάσταση του λογαριασμού και του δίνεται η δυνατότητα να επιβεβαιώσει τον λογαριασμό του πατώντας το κουμπί 'Send verification'. Με την επιλογή αυτού του κουμπιού, αποστέλλεται μήνυμα στο e-mail του, το οποίο περιέχει έναν υπερσύνδεσμο που όταν επιλεχτεί, ουσιαστικά γίνεται η επιβεβαίωση του λογαριασμού. Επιστρέφοντας στην εφαρμογή, σε περίπτωση που δε γίνει αυτόματη ανανέωση ο χρήστης μπορεί χειροκίνητα με την επιλογή 'Reload account' να φορτώσει εκ νέου τα δεδομένα του λογαριασμού του. Τέλος, με την επιλογή διαγραφής ο χρήστης μπορεί να διαγράψει τον λογαριασμό του. Αυτή η ενέργεια δεν διαγράφει τα αρχεία του από τη βάση. Συνοψίζοντας, όταν ο χρήστης έχει μη επιβεβαιωμένο λογαριασμό, τότε ενημερώνεται με κατάλληλο μήνυμα. Αυτό είναι γενικά κακή πρακτική γιατί θα μπορούσε κάποιος κακόβουλος να δημιουργεί λογαριασμούς με ψευδή e-mails. Παρόλα αυτά δε θα το λάβω υπόψη αυτό το σενάριο σε αυτό το σημείο.

## 6.7 Αποθήκευση δεδομένων χρήστη

Πέρα από τα στοιχεία με τα οποία συνδέεται ο χρήστης στην εφαρμογή, υπάρχουν και τα υπόλοιπα δεδομένα τα οποία παράγει χρησιμοποιώντας την εφαρμογή. Τέτοια δεδομένα είναι τα ESPD αρχεία που παράγει. Όλα αυτά αποθηκεύονται στην Firebase Realtime Database<sup>10</sup> [7]. Η συγκεκριμένη υλοποίηση για βάση δεδομένων παρέχει κάποιες δυνατότητες που είναι βασικές για τη λειτουργία

<sup>10</sup> Firebase Realtime Database - <https://firebase.google.com/docs/database>

της εφαρμογής. Βασική λειτουργία αυτής της βάσης δεδομένων είναι ότι υπάρχει συγχρονισμός των δεδομένων του χρήστη με το cloud. Έτσι όταν κάποιος δημιουργήσει ένα έγγραφο αυτό αυτόματα ανεβαίνει και στη βάση δεδομένων που διατηρεί η Google για το συγκεκριμένο project στο Firebase. Μία δυνατότητα που χρειάζεται απολύτως, είναι η ασύγχρονη ανανέωση δεδομένων ανάμεσα σε όλους τους clients. Αυτό σημαίνει ότι αν για παράδειγμα ο χρήστης 1 βρίσκεται σε οθόνη η οποία εμφανίζει δεδομένα από άλλους χρήστες και σε αυτούς είναι και ο χρήστης 2, και ο χρήστης 2 προσθέσει ένα νέο έγγραφο ή αλλάξει τα δεδομένα ενός υπάρχοντος, τότε αυτή η αλλαγή θα φανεί σχεδόν αμέσως και στην οθόνη του χρήστη 1. Συγκεκριμένο τέτοιο σενάριο είναι ένας χρήστης να βρίσκεται στην οθόνη επιλογής διαγωνισμού για τη δημιουργία απάντησης. Εκεί είναι αναγκαίο να λαμβάνονται αυτόματα ανανεώσεις ώστε να προβάλλεται πάντα η τελευταία έκδοση της λίστας αυτής. Κάτι ακόμα χρήσιμο που παρέχει αυτή η βάση δεδομένων είναι οι δυνατότητα να δουλεύει και εκτός σύνδεσης. Έτσι μπορεί κάποιος χρήστης να κάνει αλλαγές και να προσθαφαιρεί έγγραφα άσχετα από τη σύνδεση στο διαδίκτυο. Όταν σε δεύτερο χρόνο η εφαρμογή ανακτήσει τη σύνδεση με το διαδίκτυο, τότε όλες οι αλλαγές του χρήστη συγχρονίζονται με το cloud. Φυσικά οι αλλαγές του χρήστη θα φανούν στους υπόλοιπους μόνο όταν τελειώσει με επιτυχία ο συγχρονισμός των δεδομένων αυτών.

Και η αυθεντικοποίηση και η βάση δεδομένων παρέχονται από το Firebase. Αυτό το γεγονός δίνει τη δυνατότητα σε αυτές τις δύο οντότητες να συνεργάζονται. Ένας κοινός κόμβος των δύο οντοτήτων είναι ότι στη βάση δεδομένων δεν επιτρέπεται να γράφει μη αυθεντικοποιημένος χρήστης. Αυτό μπορεί να δηλωθεί στους κανόνες της βάσης δεδομένων με λίγες γραμμές κώδικα.

```
{
  "rules": {
    "responseDocuments": {
      "$uid": {
        ".read": "auth.uid == $uid",
        ".write": "auth.uid == $uid"
      },
    },
    "requestDocuments": {
      ".read": "auth != null ",
      "$uid": {
        ".write": "auth.uid == $uid"
      }
    }
  }
}
```

Εικόνα 34: Κανόνες βάσης δεδομένων

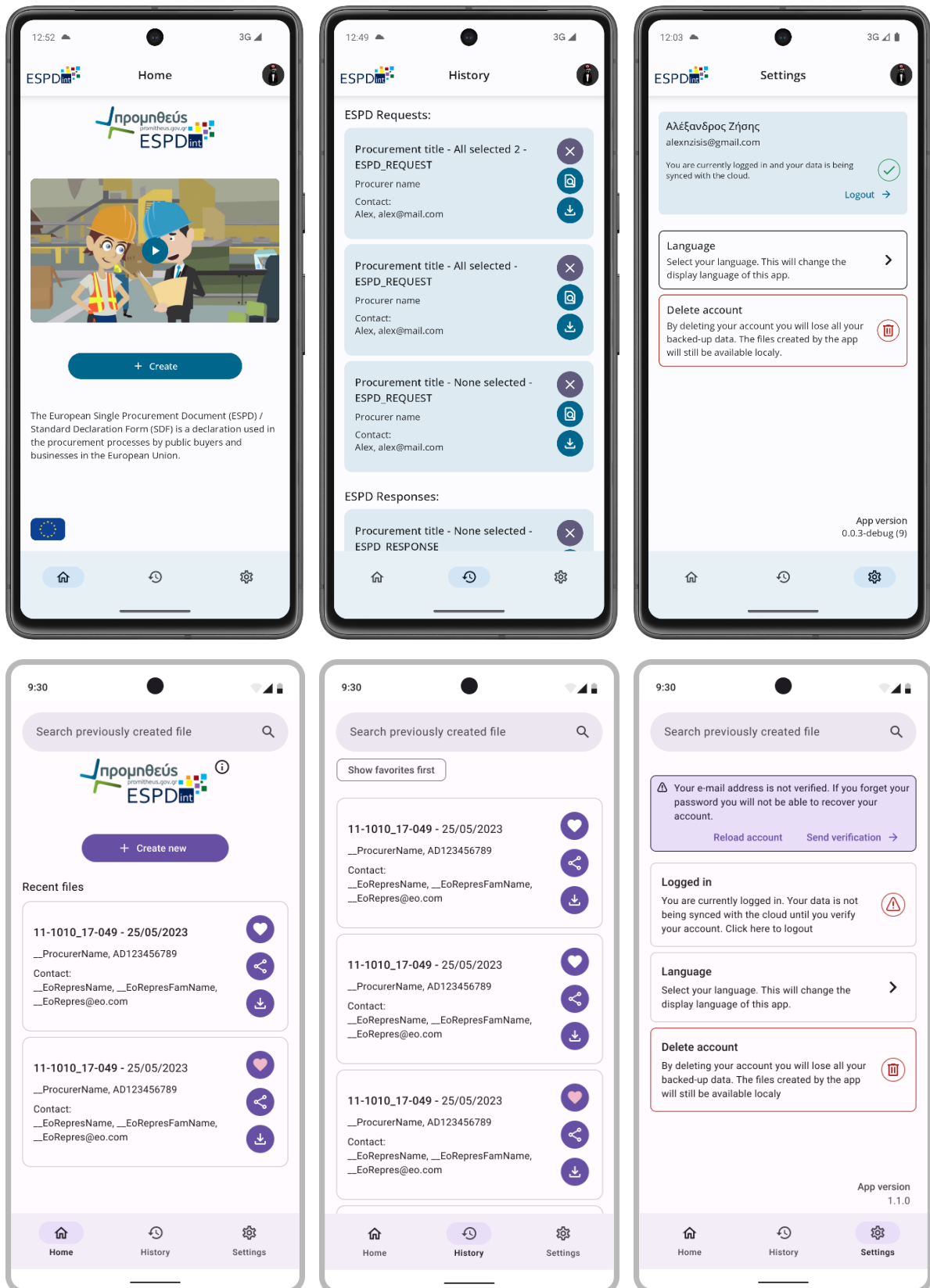
Στην Εικόνα 34, φαίνονται οι κανόνες της βάσης δεδομένων. Βλέπουμε ότι οι κανόνες αντικατοπτρίζουν και τη δομή της βάσης. Έτσι φαίνεται ότι υπάρχουν δύο βασικές διαδρομές, ή 'responseDocuments' και η 'requestDocuments'. Μέσα στην κάθε διαδρομή μπορεί να ανεβάσει κάποιος χρήστης τα έγγραφά του. Ανάλογα με το είδος του εγγράφου, θα σωθεί και σε διαφορετική διαδρομή. Αυτό δε θα ήταν αρκετό για να ξέρουμε όμως σε ποιον χρήστη ανήκει κάθε έγγραφο. Κάθε διαδρομή, προκαθορισμένα είναι private και δεν επιτρέπει ούτε διάβασμα ούτε γράψιμο. Έτσι στη διαδρομή 'responseDocuments' ενώ δεν επιτρέπουμε καμία ενέργεια, φαίνεται ότι στην υποδιαδρομή '\$uid' επιτρέπουμε και τις δύο ενέργειες αλλά μόνο όταν το αναγνωριστικό του χρήστη

ταιριάζει με αυτό που προσπαθεί εκείνη τη στιγμή να δράσει. Άρα μέσα στον φάκελο 'responseDocuments' κάθε χρήστης έχει τον δικό του χώρο στον οποίο αποθηκεύει τις απαντήσεις σε διαγωνισμούς. Αντίθετα ο φάκελος 'requestDocuments' βλέπουμε ότι μπορεί να αναγνωστεί από όλους τους χρήστες, αρκεί δηλαδή το authentication τους να μην είναι κενό. Το γράψιμο όμως από την άλλη, επιτρέπεται μόνο στους χρήστες όταν προσπαθούν να γράψουν στον δικό τους χώρο. Έτσι όταν κάποιος επιλέγει να δημιουργήσει μία απάντηση για έναν διαγωνισμό, μπορεί να δει όλη τη λίστα αυτών. Αντίθετα οι απαντήσεις είναι προσωπικές για κάθε χρήστη και μπορεί να τις δει και να τις τροποποιήσει μόνο ο ίδιος.

## 6.8 Σύγκριση αποτελέσματος με αρχικές προδιαγραφές

Για κάθε ηλεκτρονική εφαρμογή για όποια πλατφόρμα και να προορίζεται, σίγουρα το τελικό προϊόν θα απέχει από αυτό που είχε προσχεδιαστεί στα αρχικά πλάνα. Η απόκλιση από τα αρχικά σχέδια για την ανάπτυξη μιας τέτοιας εφαρμογής, μπορεί να συμβεί για διάφορους λόγους. Τέτοιοι λόγοι μπορεί να είναι τεχνικοί, λειτουργικοί, ή σχετικοί με περιορισμούς πόρων. Στο τεχνικό επίπεδο, η έλλειψη κατάλληλων εργαλείων ή η δυσκολία της υλοποίησης συγκεκριμένων λειτουργιών μπορεί να απαιτήσει προσαρμογές στον αρχικό σχεδιασμό. Επίσης, περιορισμοί στις δυνατότητες της εκάστοτε πλατφόρμας, μπορούν να επηρεάσουν τις αρχικές προδιαγραφές. Η τροποποίηση των απαιτήσεων του πελάτη αποτελεί επίσης σημαντικό παράγοντα τροποποίησης των αρχικών προδιαγραφών. Νέες απαιτήσεις ή αλλαγές στις προτεραιότητες μπορεί να απαιτήσουν προσθήκη ή τροποποίηση λειτουργιών, ενώ αλλαγές στον σχεδιασμό ή στη συνολική προσέγγιση μπορεί να οδηγήσουν σε διαφορετικές αποφάσεις εξολοκλήρου. Οι περιορισμοί πόρων, όπως η περιορισμένη διαθεσιμότητα χρόνου ή πόρων, επίσης επηρεάζουν την εξέλιξη του έργου. Η πίεση για την ολοκλήρωση του έργου εντός συγκεκριμένων χρονικών πλαισίων μπορεί να απαιτήσει συμβιβασμούς και προσαρμογές. Όταν οι προθεσμίες είναι αυστηρές, οι ομάδες ανάπτυξης ενδέχεται να υιοθετήσουν στρατηγικές προτεραιοποίησης, εστιάζοντας στα βασικά χαρακτηριστικά ή αναβάλλοντας ορισμένες λειτουργίες για μετέπειτα εκδόσεις. Αυτό μπορεί να οδηγήσει σε διαφορές από τον αρχικό σχεδιασμό, καθώς πρέπει να ληφθούν αποφάσεις για το ποια χαρακτηριστικά έχουν ύψιστη προτεραιότητα. Η πίεση για γρήγορη ανάπτυξη επίσης, μπορεί να οδηγήσει σε συμβιβασμούς ή στην αποδοχή λύσεων που δεν ανταποκρίνονται πλήρως στον αρχικό σχεδιασμό. Επιπλέον, η έλλειψη εξειδικευμένου προσωπικού ή περιορισμένη πρόσβαση σε εργαλεία και πόρους μπορεί να περιορίσει τη δυνατότητα υλοποίησης ορισμένων λειτουργιών. Συνολικά, η ευελιξία και η προσαρμογή είναι κάτι αναγκαίο και αναπόφευκτο κατά τη διάρκεια της ανάπτυξης μίας τέτοιας εφαρμογής, καθώς οι αλλαγές είναι αναμενόμενες και η επικοινωνία με τους ενδιαφερόμενους είναι καίρια για τη διατήρηση της συνοχής μεταξύ των αρχικών σχεδίων και του τελικού προϊόντος. Στην περίπτωσή μου η περιορισμένη διαθεσιμότητα χρόνου και οι αυστηροί χρονικοί περιορισμοί, ήταν κάτι που αποτέλεσε βασικό κριτήριο για την παράληψη ή τροποποίηση κάποιων λειτουργιών.

Τα σχέδια έχουν αναλυθεί στο κεφάλαιο Σχεδιασμός διεπαφής χρήστη - UI. Όπως αναφέρθηκε είναι λογικό να έχουμε αποκλίσεις από την αρχική σχεδίαση κατά τη διάρκεια της ανάπτυξης της εφαρμογής. Παρόλα αυτά οι περισσότερες αλλαγές πραγματοποιήθηκαν βάση των κριτηρίων που είχαν τεθεί στο αντίστοιχο κεφάλαιο για το UI. Καθαρά και ευκόλως κατανοητά components ήταν ο στόχος. Όλες οι οθόνες του 'bottom navigation bar' έχουν αλλαγές σε σχέση με τα σχέδια.



Εικόνα 35: Σύγκριση σχεδίων με τελικό αποτέλεσμα

Στην Εικόνα 35 φαίνονται αναλυτικά οι αλλαγές σε κάθε οθόνη στο Home screen. Φαίνεται ότι πλέον στην αρχική οθόνη δε γίνεται προβολή των τελευταίων εγγράφων που δημιούργησε ο χρήστης, αλλά αντίθετα εμφανίζονται γενικές πληροφορίες για την εφαρμογή και ένα βίντεο που μπορεί να

παρακολουθήσει ο χρήστης ώστε να μάθει περισσότερα. Στην οθόνη του ιστορικού βλέπουμε ότι δεν υπάρχει πια φίλτρο για να προβληθούν τα αγαπημένα πρώτα στη λίστα. Αυτό είναι προφανές από τη στιγμή που αφαιρέθηκε και η επιλογή να προσθέσει κάποιος χρήστης έγγραφα σε λίστα με αγαπημένα. Η αφαίρεση αυτού έγινε λόγω του ότι αποφάσισα ότι δεν ήταν μια χρήσιμη λειτουργία. Όπως φαίνεται έχω αντικαταστήσει τις επιλογές για προσθήκη σε αγαπημένα και διαμοιρασμό από κάθε έγγραφο με τις επιλογές για διαγραφή και προβολή. Ο διαμοιρασμός του εγγράφου θα ήταν κάτι χρήσιμο να μπορεί να πραγματοποιηθεί αλλά αυτό θα ήταν κάτι αρκετά χρονοβόρο. Σχετικά με την οθόνη των ρυθμίσεων οι αλλαγές που πραγματοποιήθηκαν ήταν για να γίνει πιο ξεκάθαρο το UI και οι ενέργειες που μπορεί να εκτελέσει ο χρήστης. Η λειτουργία της μπάρας αναζήτησης ήταν κάτι εξαιρετικά πολύπλοκο και άρα και χρονοβόρο να υλοποιηθεί, οπότε αποφάσισα να το αφήσω εκτός της υλοποίησης. Σημειώνεται ότι τα χρώματα είναι δυναμικά οπότε δεν αποτελούν αλλαγή.

Οι υπόλοιπες οθόνες έχουν μικρές αλλαγές και δεν έχουν νόημα να αναλυθούν περειαίρω. Κάτι σημαντικό που πρέπει να αναφερθεί είναι ότι ενώ κάποιες λειτουργίες παραλήφθηκαν, είναι εύκολο να προστεθούν στη συνέχεια και εφόσον η εφαρμογή είναι λειτουργική. Πολλές εταιρείες προτιμούν να διαθέτουν στο κοινό καινούργιες εφαρμογές και περιορισμένες λειτουργικότητες και στη συνέχεια να τις προσθέτουν σε επόμενα updates. Αυτό είναι κάτι που έχει προβλεφθεί στην υλοποίηση του κώδικα. Κάθε δομικό μέρος της διεπαφής είναι εύκολα παραμετροποιήσιμο και επεκτάσιμο. Πέρα από τη διεπαφή, η επεξεργασία των δεδομένων και οι επιχειρησιακοί κανόνες είναι επίσης εύκολα επεκτάσιμοι. Σε αυτή τη λογική είναι που φαίνεται και η χρησιμότητα της αρχιτεκτονικής που χρησιμοποιήθηκε. Έχουν ακολουθηθεί όλες οι σωστές πρακτικές ώστε το έργο να είναι ευέλικτο σε αλλαγές και άκρως παραμετροποιήσιμο. Είναι εύκολο να προστεθούν και να αφαιρεθούν λειτουργίες αλλά και να τροποποιηθούν οι υπάρχουσες.

## 6.9 Δημοσίευση εφαρμογής

Η εφαρμογή με την ολοκλήρωσή της είναι έτοιμη για παραγωγικό περιβάλλον. Διαθέτει όλες τις σωστές πρακτικές και ρυθμίσεις. Αυτές οι ρυθμίσεις είναι σχετικές με obfuscation, shrinking (κώδικα και resources) και optimization. Το obfuscation μεταλλάσσει τα ονόματα των κλάσεων, μεταβλητών και μεθόδων με σκοπό να μην είναι κατανοητός ο ρόλος τους από τρίτους. Αυτή η διαδικασία παράγει ένα αρχείο mappings.txt το οποίο χρειάζεται σε περιπτώσεις που χρειάζεται να γνωρίζουμε τον κώδικα για κάποιο συμβάν όταν η εφαρμογή είναι σε παραγωγικό περιβάλλον. Ένα τέτοιο σενάριο είναι η αναφορά σφαλμάτων. Το αυτοματοποιημένο σύστημα που τρέχει στην εφαρμογή όταν παρουσιάζεται κάποιο σφάλμα, αναφέρει το προβληματικό σημείο του κώδικα και το μήνυμα της εξαίρεσης. Αυτά όμως όταν η εφαρμογή είναι obfuscated θα είναι μη κατανοητά στους προγραμματιστές. Έτσι με το mappings.txt μπορεί να γίνει η μετάφραση των λεκτικών αυτών. Το code/resource shrinking αφαιρεί κώδικα και εικόνες/assets κτλ, από το παραδοτέο. Αυτό μπορεί να αυξήσει σημαντικά την ταχύτητα της εφαρμογής κατά την εκτέλεση. Πέρα από αυτό μειώνει και το μέγεθος του τελικού παραδοτέου. Αυτό θεωρείται σημαντικό γιατί ο χρήστης θέλει να μπορέσει να εγκαταστήσει και να εκτελέσει την εφαρμογή σε σύντομο χρονικό διάστημα χωρίς να αναμένει το κατέβασμα πολλών δεδομένων. Τέλος το optimization αναλύει τον κώδικα με σκοπό να αφαιρέσει κομμάτια που δεν εκτελούνται ποτέ. Το όφελος της διαδικασίας αυτής είναι όμοιο με το shrinking.

Έχοντας τις σωστές ρυθμίσεις μπορεί η εφαρμογή να ανέβει στο Google Play όπου δημοσιεύονται οι επίσημες android εφαρμογές. Υπάρχουν αρκετά tracks όπου μπορούν να δημοσιευτούν εκδόσεις της εφαρμογής ανάλογα με την κατάσταση. Για παράδειγμα στο track internal testing, γίνεται η δημοσίευση σε ένα εσωτερικό κανάλι όπου έχουν πρόσβαση μόνο οι δηλωμένοι testers της εφαρμογής. Υπάρχει επίσης και το open testing όπου η έκδοση δημοσιεύεται δημόσια αλλά φαίνεται ότι βρίσκεται σε early access. Η εφαρμογή θα δημοσιευτεί στο open testing όπου θα είναι δημόσια διαθέσιμη σε όλους για testing.

## 7 Συμπεράσματα & επόμενα βήματα

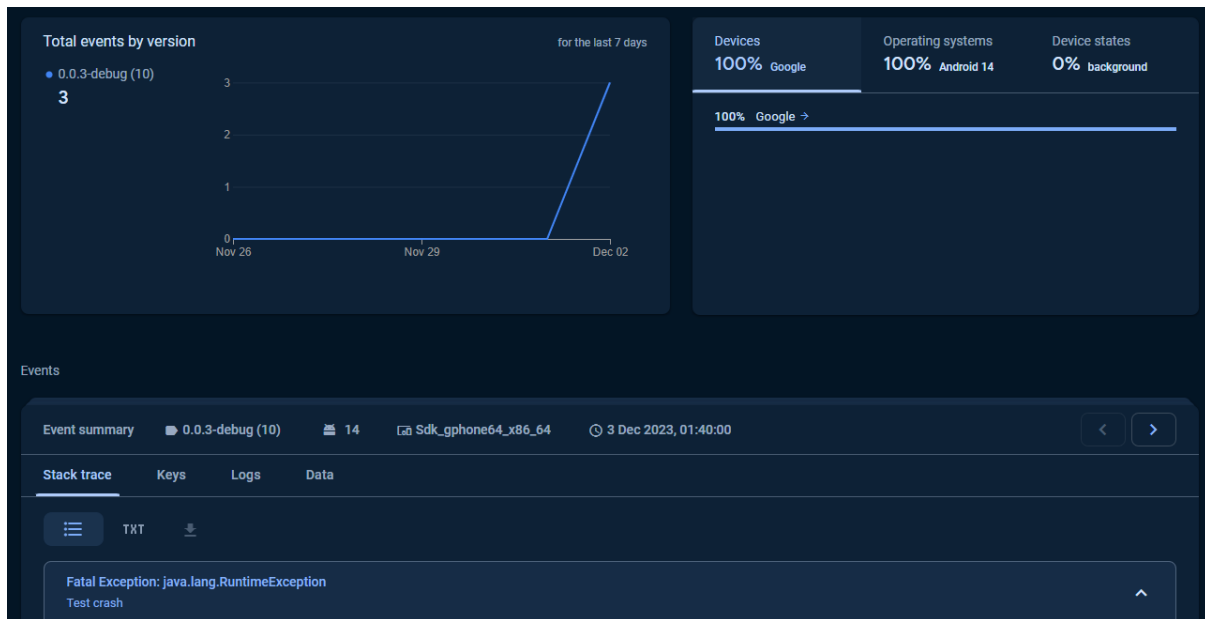
### 7.1 Συμπεράσματα

Όπως αναφέρθηκε, υπάρχουν λειτουργικότητες που αφέθηκαν εκτός αυτής της υλοποίησης λόγω χρονικών περιορισμών και πόρων. Μία από τις πρώτες λειτουργίες που θα πρέπει να υλοποιηθεί σε επόμενα στάδια είναι η υποστήριξη και αυτοτελών τύπων εγγράφων EESS (self-contained ESPD) πέρα από τα έγγραφα βάση κανονισμού της ΕΕ (regulated). Με το πέρας της υλοποίησης υποστηρίζεται η regulated μορφή με μερικούς ειδικούς περιορισμούς όπως ότι δε μπορούν να προστεθούν δυναμικά παραπάνω πεδία με ενέργεια του χρήστη σε κάποια κριτήρια. Πέρα από αυτό η τρέχουσα έκδοση των εγγράφων είναι η 2.1.0. Θα πρέπει να υποστηριχθεί η έκδοση 2.1.1 και 3.0.1. Στα κεφάλαια **Error! Reference source not found.** και **Error! Reference source not found.** έχουν αναφερθεί όλες οι αλλαγές που θα πρέπει να υποστηριχθούν. Αυτές οι αλλαγές και τροποποιήσεις θα πρέπει να συμβούν τόσο στο frontend όσο και στο backend. Αυτό σημαίνει ότι το API θα πρέπει να αρχίσει να υποστηρίζει αυτές τις εκδόσεις πριν γίνει οποιαδήποτε αλλαγή στις εφαρμογές για το web και για το mobile. Οι αλλαγές στο frontend έχουν να κάνουν με την προσθαφαίρεση πεδίων κατά τη δημιουργία εγγράφων. Η δημιουργία αυτή καθ' αυτή θα πρέπει να παραμείνει αρμοδιότητα του API.

Κάθε εφαρμογή τέτοιου είδους σίγουρα θα έχει bugs. Καταστάσεις που δεν έχουν υπολογιστεί κατά την ανάπτυξη της εφαρμογής μπορεί να οδηγήσει σε λανθάνουσες καταστάσεις. Γενικά, τα σφάλματα σε εφαρμογές αποτελούν ουσιαστικό κομμάτι της διαδικασίας ανάπτυξης λογισμικού, αναδεικνύοντας την αναπόφευκτη φύση της τεχνολογικής προόδου. Παρά τις προσπάθειες για την επίτευξη της τελειότητας, η αναπόφευκτη πολυπλοκότητα του λογισμικού και οι διάφοροι παράγοντες που επηρεάζουν την ανάπτυξή του δημιουργούν τις συνθήκες για την εμφάνιση σφαλμάτων (bugs). Η διαδικασία επίλυσης σφαλμάτων αρχίζει με την ανίχνευση του προβλήματος. Οι προγραμματιστές χρησιμοποιούν εξειδικευμένα εργαλεία για τον εντοπισμό της πηγής του σφάλματος και την καταγραφή σχετικών πληροφοριών. Η αναφορά του σφάλματος σε ένα σύστημα παρακολούθησης βοηθάει στην αποτελεσματική διαχείριση του προβλήματος και στην καταγραφή ιστορικού σφαλμάτων για μελλοντικές αναφορές. Κατόπιν, τα σφάλματα κατατάσσονται με βάση τη σοβαρότητα και τον αντίκτυπο τους, προκειμένου να προτεραιοποιηθούν. Οι προγραμματιστές αναλαμβάνουν την επίλυση των σφαλμάτων, προσθέτοντας και δοκιμάζοντας νέο κώδικα που διορθώνει τα προβλήματα. Το στάδιο των δοκιμών είναι ζωτικό για να εξασφαλιστεί ότι οι αλλαγές δεν προκαλούν νέα προβλήματα και ότι η εφαρμογή λειτουργεί ομαλά. Συνολικά, η διαδικασία επίλυσης σφαλμάτων είναι συνεχής και δρα παράλληλα με την εξέλιξη της εφαρμογής. Η συνεχής βελτίωση και αντιμετώπιση των σφαλμάτων συμβάλλει στη σταθερότητα και την αξιοπιστία της εφαρμογής, ενισχύοντας την εμπειρία των χρηστών και διασφαλίζοντας την επιτυχημένη λειτουργία της. Ένα από τα πιο διαδεδομένα εργαλεία για διαχείριση σφαλμάτων σε πλατφόρμες Android είναι το Firebase Crashlytics. Παρέχει λεπτομερείς αναφορές σφαλμάτων εστιάζοντας στον εντοπισμό και την ανάλυση τους. Η δυνατότητά του να παρακολουθεί τα σφάλματα σε πραγματικό χρόνο και να παρέχει προσαρμοσμένες ειδοποιήσεις επιτρέπει στην ομάδα ανάπτυξης να αντιμετωπίζει άμεσα προβλήματα. Η αναφορά και κατηγοριοποίηση των σφαλμάτων είναι κάτι αναγκαίο για τους προγραμματιστές όταν πρέπει να αποφασίσουν τη σειρά με την οποία θα επιλύσουν τα σφάλματα αυτά. Επιλέγοντας σφάλματα που εμφανίζονται σε πολλές διαφορετικές συσκευές και συχνά είναι σημαντικό να επιλυθούν το γρηγορότερο.

Η επίλυση σφαλμάτων είναι σημαντική γιατί το αντίθετο οδηγεί σε μειωμένη αξιοπιστία της εφαρμογής. Οι χρήστες επικοινωνούν με τις εφαρμογές με την προσδοκία ότι θα λειτουργούν απροβλημάτιστα. Τα σφάλματα μπορούν να δημιουργήσουν απογοήτευση και απώλεια εμπιστοσύνης. Επιπλέον, η μη επίλυση σφαλμάτων επηρεάζει αρνητικά την εμπειρία των χρηστών. Αναμενόμενο είναι να παρέχονται εφαρμογές που προσφέρουν εύκολη και αποτελεσματική χρήση,

και τα σφάλματα αποτελούν εμπόδιο για αυτόν τον σκοπό. Η μη επίλυση σφαλμάτων δημιουργεί επιπλοκές στη διαχείριση έργων. Σφάλματα που παραμένουν ανεπίλυτα μπορεί να επιστρέφουν στις μετέπειτα φάσεις ανάπτυξης, καθιστώντας τη διαδικασία πιο περίπλοκη και χρονοβόρα. Το θέμα της ασφάλειας αποτελεί επίσης σημαντική αρνητική επίπτωση. Σφάλματα που δεν επιλύονται μπορούν να αφήσουν ευπαθείς περιοχές στον κώδικα, αποτελώντας πηγή προβλημάτων ασφαλείας. Τέλος, η μη επίλυση των σφαλμάτων μπορεί να επηρεάσει αρνητικά τη φήμη του αναπτυσσόμενου προϊόντος ή του ίδιου του προγραμματιστή. Στον κόσμο της τεχνολογίας, η φήμη αποτελεί σημαντικό παράγοντα για την αποδοχή και επιτυχία ενός προϊόντος. Συνεπώς, η επίλυση των σφαλμάτων αποτελεί βασική απαίτηση για τη δημιουργία αξιόπιστων και αποτελεσματικών εφαρμογών, προσφέροντας στους χρήστες την ασφάλεια και την εμπειρία που αξίζουν.



Εικόνα 36: Δείγμα crash monitoring οθόνης

Στην Εικόνα 36, είναι ορατά 3 σφάλματα ομαδοποιημένα, τύπου 'RuntimeException' με κείμενο σφάλματος 'Test crash'. Είναι ορατές διάφορες επιπλέον πληροφορίες όπως το API level της συσκευής και η μάρκα, αλλά και αναλυτικότερο διάγραμμα της διαδρομής του χρήστη, τις οθόνες που προσέλασε και πόση ώρα ήταν σε καθμία μέχρι το συμβάν που μας ενδιαφέρει. Είναι σημαντικό οι προγραμματιστές να έχουν όσες περισσότερες πληροφορίες γίνεται ώστε να επιτύχουν αυξημένη ταχύτητα στη βελτίωση των σφαλμάτων της εφαρμογής.

Κάτι άλλο που μελλοντικά σε δεύτερο χρόνο θα πρέπει να υλοποιηθεί είναι η μετάβαση στην τελευταία έκδοση του μοντέλου εγγράφων. Όπως αναφέρθηκε στο 4 - Υπάρχουσα υλοποίηση, η τρέχουσα έκδοση του μοντέλου είναι η 3.3.0 (July 2023). Το API και η εφαρμογή αυτή τη στιγμή χρησιμοποιούν τα δεδομένα και τα μοντέλα της έκδοσης 2.1.0. Έτσι, βάση της ανάλυσης που έχει γίνει στο εν λόγω κεφάλαιο, θα πρέπει να γίνουν αλλαγές στο backend (API) ώστε να υποστηρίζει όλα τα πεδία που οφείλει αλλά και στο frontend (εφαρμογή) όπου θα πρέπει να προστεθούν, αφαιρεθούν και τροποποιηθούν τα ανάλογα πεδία ώστε να παρέχουν στο backend την πληροφορία που χρειάζεται.

## 7.2 Προετοιμασία για μελλοντικές αλλαγές

### 7.2.1 Τρέχουσα κατάσταση του συστήματος

Το σύστημα αυτή τη στιγμή τρέχει βάση της έκδοσης 2.1.0<sup>11</sup>. Βάση αυτής της έκδοσης έχει φτιαχτεί το τρέχων backend και frontend. Η λύση που προτείνεται σε αυτή την εργασία θα εκτελεστεί και αυτή βάση της εν λόγω έκδοσης. Παρόλα αυτά θα πρέπει να γίνει ανάλυση περί των αλλαγών που θα προκύψουν από τις επόμενες εκδόσεις όπως τις ορίζει η ΕΕ [8]. Θα πρέπει να παραχθεί ένα προϊόν που να μπορεί να επεκταθεί εύκολα και να υποστηρίξει μελλοντικές αλλαγές. Ο κώδικα της εφαρμογής που εξαρτάται από το API έχει συνταχθεί με τρόπο τέτοιο ώστε να μπορέσει να υποστηρίξει εύκολα μελλοντικές αλλαγές στη μορφή του xml και των απαντήσεων σε μορφή json.

---

<sup>11</sup> ΕΕΕΣ εκδόσεις - <https://github.com/OP-TED/ESPD-EDM/releases>



## 8 Συντομογραφίες

ΕΕΕΣ	Ευρωπαϊκό Ενιαίο Έγγραφο Σύμβασης
ΤΕΥΔ	Τυποποιημένο Έντυπο Υπεύθυνης Δήλωσης
ΕΕ	Ευρωπαϊκή Ένωση
ESPD	European Single Procurement Document
UBL	Universal Business Language
CA	Contracting Authority
CE	Contracting Entity
EO	Economic Operator
MVVM	Model View ViewModel
UX	User Experience
UI	User Interface

## 9 Αναφορές

- [1] Google, «Android Open Source Project,» [Ηλεκτρονικό]. Available: <https://developer.android.com/>. [Πρόσβαση 02 11 2023].
- [2] R. C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship., 2008.
- [3] «Οδηγία 2014/24/ΕΕ,» [Ηλεκτρονικό]. Available: <https://eur-lex.europa.eu/legal-content/EL/TXT/?uri=celex%3A32014L0024>. [Πρόσβαση 20 02 2024].
- [4] «Material Design,» [Ηλεκτρονικό]. Available: <https://m3.material.io/>. [Πρόσβαση 14 02 2024].
- [5] Προμηθεύς, «Προμηθεύς ΕΕΕΣ,» [Ηλεκτρονικό]. Available: <https://espd.eprocurement.gov.gr/>. [Πρόσβαση 24 11 2023].
- [6] Π. Πειραιά, «ESPD app API,» [Ηλεκτρονικό]. Available: <https://app.swaggerhub.com/apis/ESPDint/espd-designer/1.0.1>. [Πρόσβαση 09 11 2023].
- [7] Firebase, «Firebase docs,» [Ηλεκτρονικό]. Available: <https://firebase.google.com/docs>. [Πρόσβαση 10 11 2023].
- [8] «ESPD Technical notes,» [Ηλεκτρονικό]. Available: <https://github.com/OP-TED/ESPD-EDM>. [Πρόσβαση 06 11 2023].