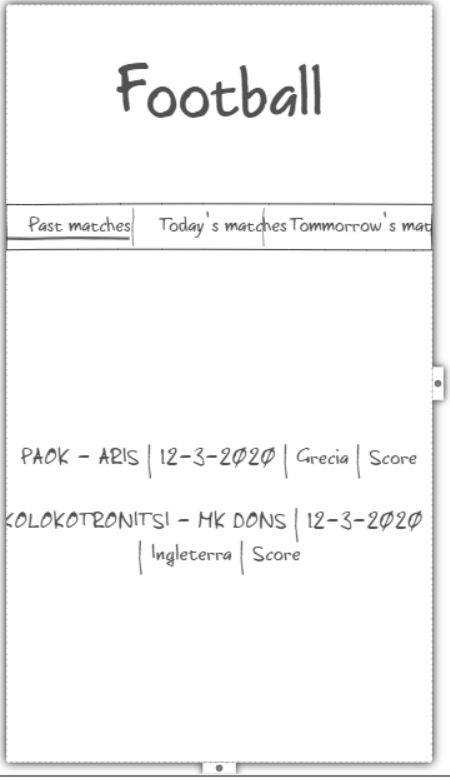
**- Εισαγωγή**

Η ιδέα της εφαρμογής κυμαίνεται γύρω από τα αθλήματα. Μάλιστα, διαλέξαμε πέντε τα οποία γνωρίζουμε και τα τρία μέλη της ομάδας σε έναν καλό βαθμό.

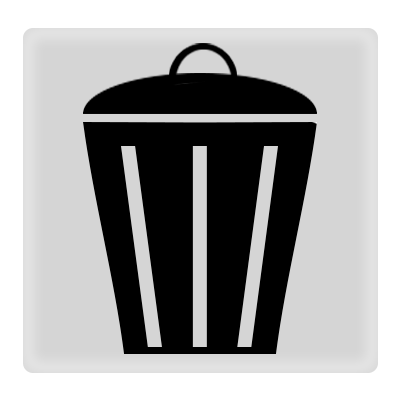
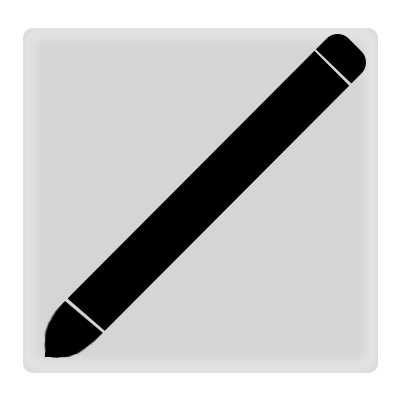
Τα αθλήματα μεταξύ τους έχουν χωριστεί σε ατομικά και ομαδικά, καθώς και σε αρρένων και θηλέων. Τα αθλήματα περιλαμβάνουν το Ποδόσφαιρο, το Μπάσκετ, Το Βόλεϊ, το Μποξ και την Πάλη. Να αναφερθεί ότι τα ονόματα των ομάδων, αλλά και των αθλητών, είναι δική μας δημιουργία, τα οποία περιέχουν **references** σε πραγματικά δεδομένα.

**- Η ιδέα και οι σημειώσεις για την δημιουργία του**

Μετά από μια μικρή έρευνα σχετικά με το εμφανισιακό κομμάτι παρόμοιων εφαρμογών, είχαμε στο μυαλό μας την πρώτη μορφή του **mock-up** πρωτοτύπων. Τα πρώτα σχεδιάστηκαν μέσω του **Ninjamock**, το οποίο έγινε συνεργατικά για αρχή, από το κομμάτι της τοποθέτησης των στοιχείων, μέχρι και το χρώμα. Η πρώτη γνωριμία με το πρόγραμμα έγινε από άλλη εργασία, αλλά η δοκιμή του στο να δημιουργηθούν τα **mock-ups** σε αυτό, σίγουρα μας ευνόησε σε μεγάλο βαθμό λόγω του **interface** του. Μάλιστα, μας επέτρεψε να μην έχουμε ένα σταθερό **template**, αλλά να επεκτείνετε ο καμβάς, ώστε να προσομοιωθεί το swipe προς τα πάνω και δεξιά, σαν κανονική εφαρμογή.



Από το **Ninjamock**, τα στοιχεία μεταφέρθηκαν απευθείας στο **Android Studio**, μόλις η ιδέα εξακριβώθηκε με λεπτομέρεια στο εργαλείο των **mock-ups**. Το χρώμα αποφασίστηκε αφού έγινε ο σχεδιασμός τους. Βέβαια, δεν καταφέραμε να φτάσουμε στον τελικό βαθμό των **mock-ups**, για διάφορους λόγους. Το αποτέλεσμα, όμως, είναι αρκετά κοντά σε αυτό που σκεφτήκαμε, σε αρκετά ικανοποιητικό σημείο.



Τα εικονίδια που χρησιμοποιήθηκαν είναι τα μισά τα **default android buttons**, ενώ τα άλλα σχεδιασμένα από την αρχή. Ο σχεδιασμός τους αποτέλεσε έμπνευση από ήδη υπάρχοντα σχέδια, ελαφρώς τροποποιημένα σε βαθμό που να ταιριάζουν με τα ήδη υπάρχοντα. Μάλιστα, στα **fragments** που χρησιμεύουν στην αποθήκευση, είτε από δημιουργία, είτε από τροποποίηση δεδομένων, το **button** που υπάρχει αποτελεί ένα **reference**, το οποίο οι γνώστες του **Star Trek** σίγουρα θα εκτιμήσουν. Υπέροχη η πληροφορία ότι το κουμπί για να ανοίξει το **Drawer Menu** αποτελεί το γεγονός ότι λέγεται **Burger**.

Τέλος, στο κομμάτι του σχεδιασμού, επιλέξαμε αυτήν την μορφή, διότι θα είναι φιλική και οικεία προς τον χρήστη, αλλά και ξεκάθαρη η κάθε λειτουργία. Αυτό σίγουρα θα έχει ως αποτέλεσμα την ευκολία του για να την χρησιμοποιήσει, αλλά και δεν θα χρειαστεί να μαθαίνει από την αρχή νέες λειτουργίες.

**- Η βάση**

Υπάρχουν 3 πίνακες στην τοπική βάση. Ο πίνακας **Sport**, ο πίνακας **Team** και ο πίνακας **Athlete**, οι οποίες περιέχουν τα βασικά **components** της βάσης. Αναλυτικά, τα στοιχεία της κάθε μιας είναι:

-- Το **table Sport** περιέχει

--- το πεδίο **SID** τύπου **int**.

--- το πεδίο **NAME** τύπου **String**.

--- το πεδίο **TYPE** τύπου **String**.

--- το πεδίο **GENDER** τύπου **String**.

-- Το **table Athlete** περιέχει

--- το πεδίο **AID** τύπου **int**.

--- το πεδίο **F\_NAME** τύπου **String**.

--- το πεδίο **L\_NAME** τύπου **String**.

--- το πεδίο **CITY** τύπου **String**.

--- το πεδίο **COUNTRY** τύπου **String**.

--- το πεδίο **SID** τύπου **int**.

--- το πεδίο **B\_YEAR** τύπου **int**.

--- το πεδίο **HOMETOWN** τύπου **int**.

-- Το **table Team** περιέχει

--- το πεδίο **TID** τύπου **int**.

--- το πεδίο **NAME** τύπου **String**.

--- το πεδίο **S\_NAME** τύπου **String**.

--- το πεδίο **CITY** τύπου **String**.

--- το πεδίο **COUNTRY** τύπου **String**.

--- το πεδίο **SID** τύπου **int**.

--- το πεδίο **E\_YEAR** τύπου **int**.

--- το πεδίο **HOMETOWN** τύπου **int**.

Τα κύρια κλειδιά του εκάστοτε πίνακα αποτελούν τα **ID** τους, δηλαδή από το **Sport** το **SID**, από το **Athlete** το **AID**, και, τέλος, από το **Team** το **TID**.

Τα **Queries** που χρησιμοποιήθηκαν είναι τρεις **SELECT \* FROM** και τρεις **DELETE FROM** για τον κάθε πίνακα.

Στο κομμάτι της απομακρυσμένης βάσης, έχουμε τα εξής στοιχεία:

-- Για τα ατομικά αθλήματα:

--- το πεδίο **athlete\_a** τύπου **int**.

--- το πεδίο **athlete\_b** τύπου **int**.

--- το πεδίο **city** τύπου **String**.

--- το πεδίο **country** τύπου **String**.

--- το πεδίο **date** τύπου **String**.

--- το πεδίο **num\_of\_athletes** τύπου **int**.

-- Για τα ομαδικά αθλήματα:

--- το πεδίο **team\_a** τύπου **int**.

--- το πεδίο **team\_b** τύπου **int**.

--- το πεδίο **city** τύπου **String**.

--- το πεδίο **country** τύπου **String**.

--- το πεδίο **score\_a** τύπου **int**.

--- το πεδίο **score\_b** τύπου **int**.

--- το πεδίο **sport** τύπου **String**.

**- Εφαρμογή**

Η εφαρμογή μας έχει γραφτεί σε Java. Αποτελείται από περίπου **30 κλάσεις** και **20 xml files**. Συγκεκριμένα, **τέσσερις κλάσεις** αφορούν την **απομακρυσμένη** βάση, **εννέα classes** για την τοπική βάση, χωρίς το **LocalDao**, το οποίο είναι **interface**, και οι υπόλοιπες αφορούν την λειτουργία των **Fragments** στο πρόγραμμα. Παρακάτω, θα δείτε με αυτόν ακριβώς τον τρόπο χωρισμένες τις κλάσεις:

-- **Remote Database**

Οι κλάσεις της απομακρυσμένης βάσης εκτελούν λειτουργίες εισαγωγής, διαγραφής, τροποποίησης και λήψης δεδομένων για αγώνες. Συγκεκριμένα, η κλάση **InsertData** περιέχει δύο μεθόδους για εισαγωγή αγώνων ατομικών ή ομαδικών αθλημάτων αντίστοιχα, οι μέθοδοι παίρνουν ως παραμέτρους όλα τα στοιχεία του αγώνα. Σε επιτυχή εισαγωγή δεδομένων στην βάση, αποστέλλεται στον χρήστη ειδοποίηση αλλιώς αποστέλλεται μήνυμα Toast.

Η κλάση **DeleteData** περιέχει μία μέθοδο για διαγραφή αγώνων η οποία παίρνει ως παραμέτρους τον κωδικό**(id)** του αγώνα καθώς και το άθλημα στο οποίο ανήκει ο αγώνας. Η μέθοδος επίσης ειδοποιεί τον χρήστη καταλλήλως, ανάλογα με το αν πέτυχε ή όχι η διαγραφή.

Η κλάση **UpdateData** περιέχει δύο μεθόδους, μία για να λαμβάνει όλους τους κωδικούς των αγώνων για ένα άθλημα με τις αντίστοιχες παραμέτρους και μία μέθοδο η οποία με παραμέτρους το άθλημα, τον κωδικό του αγώνα, το πεδίο του αγώνα και την τιμή που θα πάρει τροποποιεί ένα συγκεκριμένο πεδίο ενός αγώνα. Στην δεύτερη περίπτωση ενημερώνεται ο χρήστης με **Heads-Up** ειδοποίηση αν η ενημέρωση ήταν επιτυχής, αν απέτυχε ενημερώνεται με Toast.

Η κλάση **GetData** περιέχει μία μέθοδο για λήψη δεδομένων αγώνων που σχετίζονται με ένα μόνο άθλημα, τα δεδομένα αποθηκεύονται παραμετρικά σε ένα αντικείμενο τύπου Map.

-- **Local Database**

Η κλάση-πίνακας **AthleteDB** της τοπικής βάσης περιέχει μεταβλητές για όλα τα στοιχεία ενός αθλητή, μεθόδους getters-setters και toString, καθώς και τον πλήρη και τον κενό δομητή. Το πεδίο **AID**(Athlete ID) αποτελεί το κύριο κλειδί του πίνακα και το πεδίο **SID**(Sport ID) αποτελεί ξένο κλειδί προς τον πίνακα **Sport**.

Η κλάση-πίνακας **SportDB** περιέχει μεταβλητές για τα στοιχεία ενός αθλήματος, τον πλήρη και τον κενό δομητή, τις μεθόδους getters-setters και την toString. Το πεδίο **SportID** είναι το κύριο κλειδί του πίνακα.

Η κλάση-πίνακας **TeamDB** περιέχει πεδία για τα στοιχεία ενός αθλήματος, τις μεθόδους getters-setters και την toString καθώς και τον κενό και τον πλήρη δομητή. Το πεδίο **TID**(Team ID) αποτελεί το κύριο κλειδί και το πεδίο SID είναι ξένο κλειδί προς τον πίνακα **Sport**.

Η κλάση **AthleteManageData** περιέχει μεθόδους διαχείρισης των δεδομένων των αθλητών, συγκεκριμένα μπορεί να προσθέσει ή να διαγράψει έναν αθλητή από την βάση ή να τροποποιήσει τα στοιχεία ενός υπάρχων αθλητή.

Η κλάση **SportManageData** περιέχει μεθόδους προσθήκης, τροποποίησης ή διαγραφής ενός αθλήματος από την βάση.

Η κλάση **TeamManageData** περιέχει μεθόδους τροποποίησης, διαγραφής ή προσθήκης μίας ομάδας.

Οι μέθοδοι διαχείρισης δεδομένων των πινάκων βασίζουν την λειτουργία τους στις μεθόδους της διεπαφής **LocalDAO** και όταν εκτελούν μία επιτυχή λειτουργία, ενημερώνουν τον χρήστη με **Heads-Up** ειδοποίηση ενώ αν αποτύχουν λόγω κάποιου σφάλματος ενημερώνουν τον χρήστη με μήνυμα **Toast**.

Η κλάση **SendNotification** με μία μέθοδο δημιουργεί ειδοποιήσεις, το περιεχόμενο και ο τίτλος των οποίων προσαρμόζεται με παραμέτρους της μεθόδου.

Η κλάση **FillDB** δοκιμάζει σε κάθε εκκίνηση της εφαρμογή να αποθηκεύσει δεδομένα αθλημάτων, ομάδων και αθλητών στην τοπική βάση, αν αυτά δεν υπάρχουν ήδη.

Η κλάση **LocalDatabase** περιέχει μία μέθοδο για διαγραφή όλων των δεδομένων της βάσης και δηλώνει ένα αντικείμενο τύπου **LocalDAO** με το οποίο έχουν πρόσβαση στην βάση άλλα κομμάτια της εφαρμογής. Επίσης δηλώνει και ποιες κλάσεις θα αποτελέσουν τους πίνακες της βάσης.

Η διεπαφή **LocalDAO** είναι το μέσο επικοινωνίας της εφαρμογής με την τοπική βάση και περιέχει τις βασικές μεθόδους λήψης, τροποποίησης και διαγραφής των δεδομένων των πινάκων της βάσης.

-- **Base Program**

Το βασικό πρόγραμμα αποτελείται από **δεκαπέντε** κλάσεις. Αυτές συνδέονται άμεσα με τα **fragments**. Κάθε μία έχει και μία ξεχωριστή λειτουργία για την εφαρμογή. Υπάρχουν **κλάσεις** που κάνουν **τροποποίηση** των στοιχείων είτε από ομάδες(**EditTeam**), είτε από αθλητές(**EditAthlete**), είτε από τα **match(EditTeamMatch** και **EditIndividualMatch)**. Επίσης, υπάρχουν κλάσεις για να κάνεις προσθήκη στοιχείων είτε ομάδων, είτε αθλητών, είτε αγώνων και κλάσεις που κάνουν διαγραφή των ομάδων, αθλητών ή αγώνων**.** Έχει δημιουργηθεί, επίσης, κλάση για το **Homescreen**, το οποίο περιέχει και το **logo** της εφαρμογής, μαζί με ένα βασικό κείμενο, τα μέλη της ομάδας και δύο **Buttons** για την εισαγωγή **Αθλητών** και **Ομάδων** στη βάση αντίστοιχα. Προφανώς, υπάρχει και η κλάση **MainActivity**, όπως πρέπει σε κάθε **Android App**, η οποία αρχικοποιεί διάφορα κρίσιμα κομμάτια της εφαρμογής όπως την απομακρυσμένη και τοπική βάση, το κανάλι ειδοποιήσεων και το αντικείμενο ειδοποιήσεων, τα αντικείμενα στα οποία αποθηκεύονται τα δεδομένα την απομακρυσμένης βάσης, καθώς και το **toolbar** και το **Drawer Layout**. Τέλος, υπάρχουν οι κλάσεις οι οποίες ειδικεύονται στους πίνακες της τοπικής βάσης, δηλαδή για τις ομάδες και τους αθλητές και οι κλάσεις οι οποίες ειδικεύονται στις συλλογές της απομακρυσμένης βάσης, δηλαδή για τους αγώνες. Αυτές οι κλάσεις αφορούν είτε το να εμφανίζουν τις πληροφορίες των αθλητών ή των ομάδων ή των αγώνων, να κάνουν **τροποποίηση**, όπως προαναφέρθηκε, τα στοιχεία τους, αλλά και να προσθέτουν νέα **components**. Το navigation της εφαρμογής έγινε με υλοποίηση στοίβας των fragments χρησιμοποιώντας την κλάση **FragmentTransaction** και τις λειτουργίες που αυτή προσφέρει.

-- **Fragments**

Υπάρχουν, περίπου, είκοσι **xml** fragments στην εφαρμογή μας. Αυτά ειδικεύονται σε κάθε χρήση που θέλουμε να υπάρχει σε αυτήν. Υπάρχουν ξεχωριστά fragments για την προσθήκη ή την τροποποίηση παραμέτρων, ή fragments για την εμφάνιση των προγραμματισμένων αγώνων ξεχωριστά για κάθε είδος αθλήματος, είτε ομαδικό, είτε ατομικό. Με αυτόν τον τρόπο χωρίζεται και η επιλογή των **Sports**, μιας και μέσω του **Drawer Menu** μπορείς να επιλέξεις αν θες να δεις αγώνες για ατομικά ή για ομαδικά αθλήματα. Τα fragments χρησιμοποίησαν, κατά έναν μεγάλο βαθμό, **Constraint Layouts**, διότι ήταν ο πιο χρήσιμος, βολικός, και εύκολος τρόπος τοποθέτησης components στην εφαρμογή. Μέσα σε αυτά έχουν χρησιμοποιηθεί **TextView**, **EditText**, **ImageView**, **Map** και **Buttons**, διότι αυτά εκτελούσαν τις λειτουργίες με τον τρόπο που επιθυμούσαμε.

Μιας και αναφέρθηκε πριν, το **Drawer**, μέσω αυτού μπορείς να περιηγηθείς στην εφαρμογή, ώστε να διαλέξεις αν θες να δεις τους αγώνες που προαναφέρθηκε, αλλά και να δεις την λίστα των αθλητών και των ομάδων που περιέχονται μέσα στην βάση. Αυτές οι επιλογές γίνονται ανοίγοντάς το.

Τέλος, υπάρχει ένα **Toolbar**, το οποίο αναγράφει σε ποιο **fragment** βρίσκεσαι. Μάλιστα, ο σκοπός ήταν, τουλάχιστον στο κομμάτι των αθλημάτων, να αλλάζει θεματικά σε διαφορετικό χρώμα, το οποίο θεωρήσαμε ότι είναι ταιριαστό για το κάθε άθλημα. Δείγματα πάρθηκαν από το χρώμα του γηπέδου του ποδοσφαίρου για παράδειγμα, ή το χρώμα γαντιών του μποξ.

**- Συμπεράσματα**

Η συνεργασία στην ομάδα ήταν εξαιρετική. Σίγουρα, το γεγονός ότι γνωριζόμασταν βοήθησε σε μεγάλο βαθμό, μιας και η επικοινωνία μας ήταν πιο άμεση και πιο οικεία. Ο χαβαλές υπήρχε στον βαθμό που έπρεπε, και δεν μας απέσπασε από τον τελικό στόχο, δηλαδή, την ολοκλήρωση της εφαρμογής.

Το κομμάτι των δουλειών, όλα μοιράστηκαν σε ομοιόμορφο βαθμό. Ο καθένας ασχολήθηκε, από την μία αποκλειστικά με το κομμάτι που είναι ειδικός, από την άλλη υπήρχαν βοήθειας από όλους προς όλους. Παρόλο το χρόνο που είχαμε, καταφέραμε να βγάλουμε ένα επιθυμητό αποτέλεσμα.

Αν μας δοθεί ξανά η ευκαιρία για συνεργασία, είτε σε αυτή, είτε σε παρόμοια εργασία, σίγουρα μπορούμε να μάθουμε από τα οποιαδήποτε λάθη που έγιναν στην οργάνωση, αλλά, εν τέλη, καταφέραμε μία δυνατή εφαρμογή, από την οποία είμαστε ευχαριστημένοι.

Τα μέλη της ομάδας,

Κόνσουλας Ηρακλής

Μηδέλιας Αθανάσιος

Ψαρράς Αθανάσιος

