Πτυχιακή

Ανδρέας Ηροδότου - Α.Μ. 3698

Επιβλέπων: Δημήτρης Πλεξουσάκης

Συνεπιβλέπων: Παναγιώτης Παπαδάκος

ServiceLink: Μια διαδικτυακή πλατφόρμα για την εύρεση επαγγελματικών υπηρεσιών βάσει περιορισμών ορισμένων από χρήστες

ServiceLink: An online platform for finding professional services according to user-based constraints

### Εισαγωγή

Η τεχνολογία αναπτύσσεται και χρήσιμες λειτουργίες τείνουν να γίνονται ολοένα και περισσότερο από ένα υπολογιστή ή ένα smartphone. Το σύστημα που θέλουμε να δημιουργήσουμε έχει σαν σκοπό να κάνει πιο απλή και αξιόπιστη την παροχή υπηρεσιών, δίνοντάς στους users όσο το δυνατόν πιο ευχάριστη εμπειρία. Αυτό θα επιτυγχάνεται μέσα από διάφορους αυτοματοποιημένους μηχανισμούς. Με άλλα λόγια το σύστημα λειτουργεί σαν ένας μεσάζοντας μεταξύ των customers και των professionals, προσφέροντας μερικές χρήσιμες λειτουργίες και οφέλη προς τους users. Μερικά από αυτά είναι: μείωση του χρόνου που απαιτείται για την εύρεση του κατάλληλου ατόμου για μία δουλειά, αφού θα ενημερώνονται αυτόματα από το σύστημα όσοι professionals τηρούν τα κριτήρια και επομένως θα στέλνουν τις προσφορές τους. Η μείωση των εξόδων του user, αφού θα έχει πληθώρα επιλογών, άρα θα έχει την ευκαιρία να επιλέξει κάτι οικονομικό και αξιόπιστο. Η αύξηση εργασίας των professionals και πολλά άλλα…

### Παρεμφερή Συστήματα

Μερικές παρόμοιες υπηρεσίες παρέχει το “[www.worky.gr](http://www.worky.gr)”. Οι προσφορές γίνονται σαν δημοπρασία, ο κάθε επόμενος professional που θέλει να κάνει προσφορά πρέπει να έχει καλύτερη τιμή από τον προηγούμενο (αρνητικό). Ο κύριος σκοπός της ιστοσελίδας είναι να παρέχει στους customers όσο το δυνατό χαμηλότερες τιμές, χωρίς να δίνει τόσο έμφαση στην ποιότητα (δεν υπάρχουν feedbacks). Αφήνει τους customers να δημιουργούν κανονικά αγγελίες χωρίς χρέωση, ενώ αντίθετα για τους professionals υπάρχει χρέωση για κάθε προσφορά που κάνουν(πρέπει να αγοράζουν πακέτα συμμετοχών).

Ένα άλλο πολύ σχετικό site είναι το «[www.atyourservice.com.cy](http://www.atyourservice.com.cy)». Έχει μερικές προκαθορισμένες κατηγορίες και στηρίζει μόνο αγγελίες αυτών των κατηγοριών. Οι προσφορές που παίρνει ένας customer δεν φαίνονται σε κανένα άλλο. Οι professionals μπορούν να δουν απλά πόσες άλλες προσφορές έγιναν σε μια αγγελία(δεν μπορούν να δουν ποιοι professionals έκαναν τις προσφορές, ούτε τι χρεώνουν). Κατά την δημιουργία μιας αγγελίας ο customer συμπληρώνει μόνο δύο πεδία, την κατηγορία και την περιγραφή της υπηρεσίας που ζητά(ελλιπή δεδομένα). Οι αγγελίες δεν έχουν ημερομηνία λήξης... Οι βασικές υπηρεσίες του site είναι δωρεάν, γενικά υπάρχει και το επαγγελματικό πακέτο με μηνιαία συνδρομή που δίνει μερικά επιπλέον πλεονεκτήματα και βοηθητικές υπηρεσίες. Το επαγγελματικό πακέτο παρέχει χρήσιμες υπηρεσίες όπως αυτόματη ενημέρωση, online ημερολόγιο και μηνιαία ανάλυση δεδομένων. Επίσης υπάρχουν feedbacks από ένα customer προς ένα professional μόνο.

Γενικά το «[www.atyourservice.com.cy](http://www.atyourservice.com.cy)» καλύπτει πολλές από τις λειτουργίες που θα θέλαμε στο σύστημα μας. Παρόλα αυτά δεν είναι τόσο οργανωμένο και αυτοματοποιημένο λόγω ελλιπή δεδομένων, αυτό προκύπτει επειδή δεν είναι «απαιτητικό» ως προς τους users, οπόταν υπάρχει ελλείψει χρήσιμων δεδομένων που θα μπορούσαν να αξιοποιηθούν για να επιτευχθούν περισσότερες λειτουργίες και καλύτερη οργάνωση.

### Λειτουργικότητα Συστήματος

Γενικά θα υπάρχουν δύο ειδών accounts. Το ένα είδος account αφορά τον ***πελάτη (customer)*** και θα του δίνει τη δυνατότητα να ζητάει υπηρεσίες βάσει των προτιμήσεων και περιορισμών τους οποίους ο ίδιος θέτει και να λαμβάνει προσφορές από τους επαγγελματίες. Το δεύτερο είδος account αφορά τους ***επαγγελματίες (professionals)*** οι οποίοι μπορούν να κάνουν προσφορές στις αιτήσεις που κάνουν οι πελάτες, το σύστημα τους ετοιμάζει αυτόματα το πρόγραμμα με τις υπηρεσίες που έχουν αναλάβει, όπως επίσης ετοιμάζει και διάφορα στατιστικά στοιχεία για να μπορούν να παρακολουθούν την απόδοση τους σε διάφορα χρονικά διαστήματα. Οι προσφορές που κάνουν οι επαγγελματίες για την ίδια αίτηση είναι φανερές μόνο στον πελάτη, ο οποίος επίσης έχει πρόσβαση και σε ιστορικό με τα reviews/ratings των επαγγελματιών. Όταν ένας customer επιλέξει κάποια προσφορά από αυτές που του έχουν κάνει οι professionals, αυτόματα ανταλλάσσονται τα στοιχεία επικοινωνίας των δύο users (τηλέφωνα ή ενεργοποίηση κάποιου διαδικτυακού chat), το σύστημα κάνει track την όλη διαδικασία και ενημερώνει για το status της και τους δύο. Με το που τελειώσει η συγκεκριμένη εργασία οι δύο users έχουν την ευκαιρία αν θέλουν να αφήσουν ένα review (βαθμολογία και comments) ο ένας για τον άλλο (επώνυμα).

##### Πως ζητάει ένας πελάτης μία υπηρεσία

Ο customer πρέπει να καθορίσει ακριβώς ποιο είναι το πρόβλημα γράφοντας μια περιγραφή και παραθέτοντας μερικές φωτογραφίες. Επίσης πρέπει να καθορίσει τις διαθέσιμες ώρες, μέρες, τοποθεσία, σε ποιο είδος επαγγελματία απευθύνεται, τι ποσό διαθέτει, διάρκεια ζωής της αγγελίας κλπ.

##### Τι πεδία που πρέπει να συμπληρώσει ο επαγγελματίας

Ο professional πρέπει να συμπληρώσει πέραν των προσωπικών του στοιχείων, με τι ασχολείται, τις περιοχές που μπορεί να εξυπηρετήσει, τις διαθέσιμες ώρες εργασίας κλπ. Το σύστημα δημιουργεί δυναμικά το πρόγραμμα του professional έτσι ώστε να μην συμπέσουν δύο εργασίες την ίδια χρονική στιγμή.

##### Από τί αποτελείται μία προσφορά

Ο professional μπορεί να γράψει μερικές πληροφορίες για το πως σκέφτεται να επιλύσει το πρόβλημα, τι θα κοστίσει και πόσο χρόνο χρειάζεται για να το επιδιορθώσει.

##### Ενημέρωση πελάτη για τις διαθέσιμες προσφορές

Ένας customer ενημερώνετε άμεσα με τους τρόπους που αναφέραμε και παραπάνω όταν κάποιος professional του κάνει μία προσφορά. Επιπρόσθετα μπορεί να έχει απευθείας πρόσβαση στο profile του professional που έστειλε την προσφορά, για να δει το ιστορικό του καθώς και περισσότερες λεπτομέρειες για τις γνώσεις, την προϋπηρεσία που έχει, όπως επίσης και τα reviews (βαθμολογία και εντύπωσης από άλλα άτομα) του.

##### Πως γίνεται η ενημέρωση ενός professional ώστε να στείλει μία προσφορά

Η ενημέρωση ενός professional γίνεται ως εξής: το σύστημα αυτόματα εντοπίζει το σύνολο τον professionals που απευθύνεται η κάθε αγγελία βάση των πληροφοριών που παρέθεσε ο customer που την έφτιαξε, τα δεδομένα που υπάρχουν στο profile του κάθε professional και βάσει των χρονικών του περιορισμών. Οπότε με το που συνδεθεί κάποιος professional στο σύστημα του εμφανίζονται όσες αγγελίες τον αφορούν άμεσα. Επίσης όπως και ο customer, έτσι και ο professional μπορεί να δει προηγούμενα reviews του συγκεκριμένου customer και να αποφασίσει κατά πόσο είναι αξιόπιστος για να στείλει την προσφορά του.

**Απαιτήσεις Συστήματος (Requirements)**

**Λειτουργικές Απαιτήσεις (Functional** **Requirements)**

Όταν ένας user κάνει sign up δημιουργείται ένα νέο account (customer ή professional) με όλα τα στοιχεία του user. Tου παρέχεται δυνατότητα να κάνει edit ή delete τα στοιχεία του όποτε θέλει.

Όταν ένας customer δημιουργεί μια νέα αγγελία, η αγγελία δημοσιεύεται αυτόματα και είναι εμφανής σε όλους τους customers και professionals. Μετά τη δημοσίευση της αγγελίας υπάρχει η δυνατότητα να γίνουν edit τα πεδία περιγραφή, τίτλος και μέγιστο ποσό που διατίθεται. Το edit της αγγελίας μπορεί να γίνει μόνο από τον customer που την δημιούργησε.

Ο κάθε professional μπορεί να δώσει προσφορές σε όλες τις αγγελίες που τον αφορούν. Δεν μπορεί να κάνει περισσότερες από μια προσφορές σε μία αγγελία, μπορεί να κάνει edit όμως την προσφορά που έκανε ήδη.

Ένας customer μπορεί να έχει απευθείας πρόσβαση στο profile του professional που έστειλε την προσφορά και ένας professional στο profile του customer που πρόσθεσε μία αγγελία.

Όταν ο user επιλέξει έναν professional για την εργασία ανταλλάσσονται τα προσωπικά στοιχεία της επικοινωνίας τους.

Το πρόγραμμα του κάθε professional δημιουργείται δυναμικά, με βάσει τις προσφορές του που έχουν επιλεγεί από τους customers.

Μετά την εκτέλεση μιας εργασίας δίνεται η ευκαιρία στον customer και στον professional να βαθμολογήσουν και να γράψουν ένα review ο ένας για τον άλλον.

**Μη Λειτουργικές Απαιτήσεις (Non-Functional Requirements)**

Το σύστημα εγγυάται την ακεραιότητα των δεδομένων.

Στην βάση τα ευαίσθητα δεδομένα αποθηκεύονται encrypted.

Το response από τον server θα γίνεται σε χρόνο μικρότερο του ενός δευτερολέπτου.

Το σύστημα μπορεί να εξυπηρετήσει περισσότερα από 1000 request το δευτερόλεπτο.

### Υποθετικό σενάριο

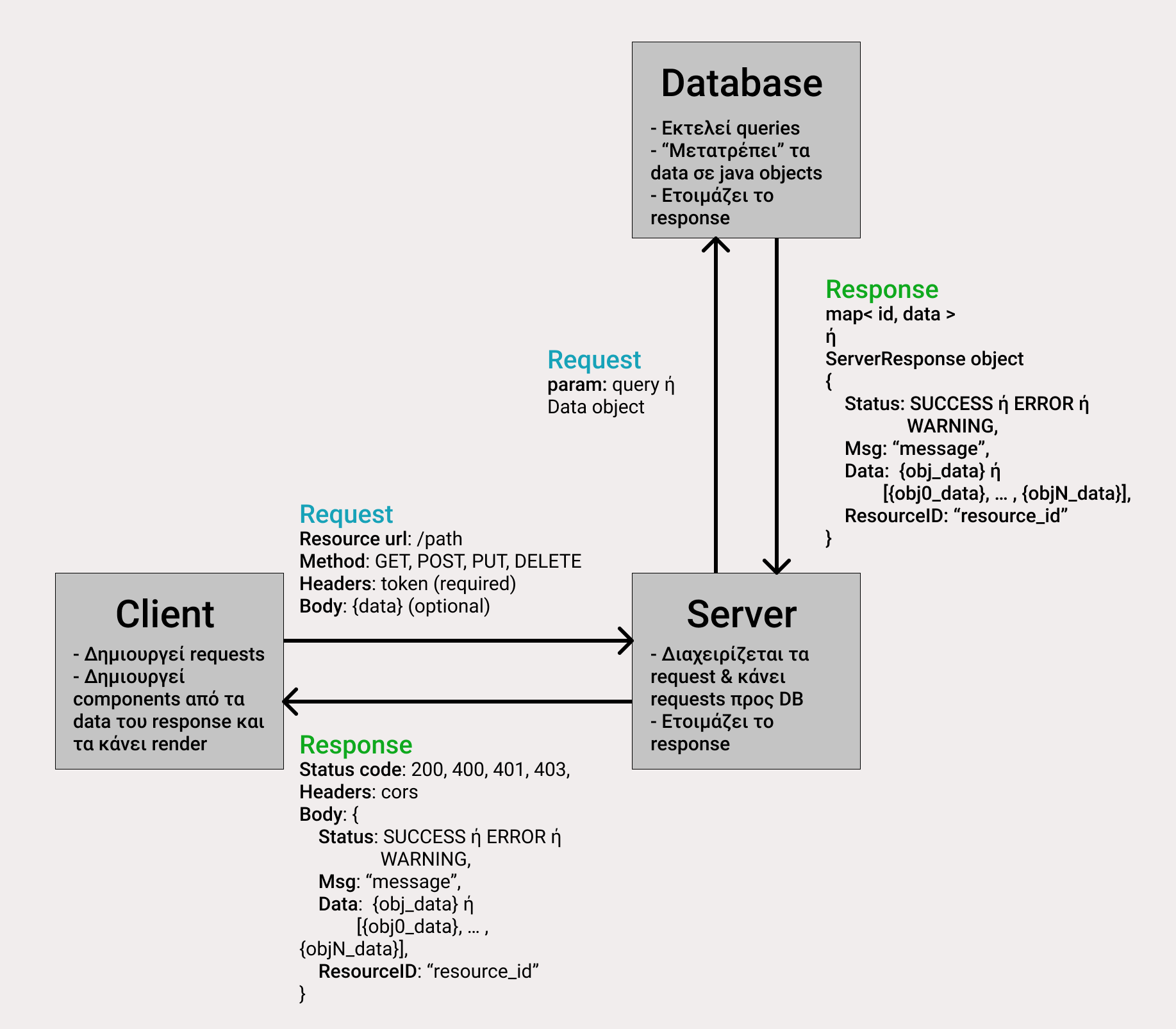
**Customer**

Ο Γιώργος ανακαλύπτει ότι η βρύση που υπάρχει στην κουζίνα του σπιτιού του έχει διαρροή. Ο Γιώργος δεν έχει κάποιο γνωστό του που να είναι υδραυλικός, οπότε αποφασίζει να δημιουργήσει μία αίτηση υπηρεσίας μέσω του ServiceLink για να βρει κάποιο ειδικό να του φτιάξει την διαρροή. Αφού κάνει εγγραφή στην σελίδα σαν customer, κάνει login και ψάχνει μέσω της αναζήτησης να βρει μερικά παρόμοια listings για να πάρει μία ιδέα για το πώς να φτιάξει το δικό του listing. Τώρα εμπνευσμένος και από άλλους χρήστες δημιουργεί ένα καινούριο listing στο οποίο παραθέτει μερικές φωτογραφίες, συμπληρώνει την τοποθεσία που βρίσκετε, σε ποιους απευθύνεται, το μέγιστο ποσό για προσφορές και τη διάρκεια ζωής της αγγελίας. Αφού γίνουν μερικές προσφορές από professionals, ο Γιώργος βλέπει τα ratings που έχουν και επισκέπτεται το profile τους για να διαβάσει περισσότερα για αυτούς και να δει τα comments που άφησαν άλλοι customers οι οποίοι συνεργάστηκαν μαζί τους. Μετά τον έλεγχο όλων των δεδομένων που έχει στην διάθεση του μέσω του συστήματος (ratings, comments, τιμή, προτεινόμενη επίλυση κλπ.) καταλήγει με ποιον professional θέλει να συνεργαστεί. O Γιώργος μετά την συνάντηση και χαρούμενος με την δουλειά που έχει γίνει από τον professional που είχε επιλέξει, κάνει ένα θετικό review για να τον ευχαριστήσει και για να βοηθήσει άλλα άτομα που αντιμετωπίζουν παρόμοιο πρόβλημα, να κάνουν σωστή επιλογή.

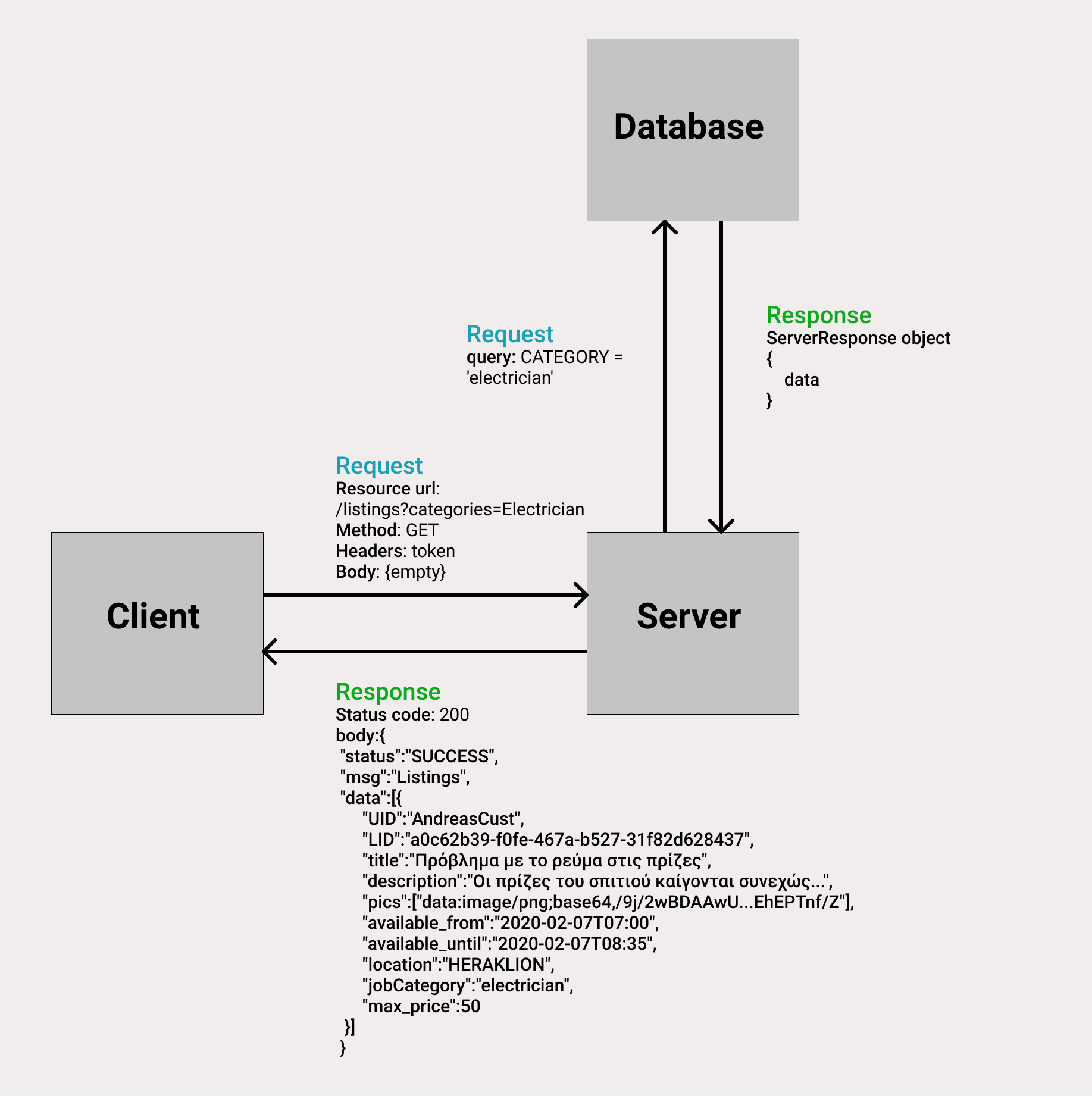
**Professional**

Ο Νίκος είναι υδραυλικός, παρόλο που είναι εξαιρετικός στην δουλειά του, λόγο των ελάχιστων χρόνων εμπειρίας και λίγων γνωστών που έχει, «πάσχει» από έλλειψη δουλείας. Προσπαθώντας να αυξήσει τους πελάτες του βρίσκει το ServiceLink σαν ένα κάλο μέσω για να τον γνωρίσει περισσότερος κόσμος. Αποφασισμένος για ένα καλύτερο μέλλον δημιουργεί ένα professional account και συμπληρώνει όλα τα απαραίτητα στοιχεία για να είναι πλήρες το profile του. Με το που συνδεθεί στο ServiceLink αυτόματα του εμφανίζονται όλα τα διαθέσιμα listings στην περιοχή και τον κλάδο του, βλέπει μερικά από αυτά και αποφασίζει να κάνει μία προσφορά για ένα που φαίνεται στα μέτρα του. Μετά από μερικά λεπτά συνειδητοποιεί χαρούμενος, ότι η προσφορά του έγινε δεκτή από τον customer. Επισκέπτεται το timetable του και βλέπει ότι αυτόματα το σύστημα δημιούργησε ένα event για την μέρα και ώρα που είναι η προγραμματισμένη συνάντηση του με τον πελάτη. Ο Νίκος μετά την επίσκεψη και επιδιόρθωση του προβλήματος που είχε ο πελάτης και ευτυχισμένος από την συνεργασία τους, αποφασίζει να κάνει ένα θετικό review, για να γνωρίζουν και οι υπόλοιποι professionals ότι είναι καλός πελάτης και να τον βοηθήσει έτσι ώστε να μπορούν να τον εμπιστεύονται περισσότεροι professionals και να παίρνει περισσότερες προσφορές σε ότι πρόβλημα του προκύψει μελλοντικά. Μετά από καιρό χρήσης του ServiceLink ο Νίκος αναρωτιέται πως τα πάει, αν βελτιώνετε σαν professional, αν παίρνει περισσότερα θετικά reviews, αν αυξήθηκαν με τον καιρό οι δουλείες και τα κέρδη του. Για να πάρει μια γεύση για όλ’ αυτά κατευθύνεται στα στατιστικά που του παρέχει το σύστημα και βλέπει σε διάφορα διαστήματα χρήσιμα στατιστικά που τον βοηθούν να βελτιωθεί ακόμη περισσότερο.

**Αρχιτεκτονική Συστήματος**

**Γενική Δομή Συστήματος**

**Παράδειγμα ενός request - response**



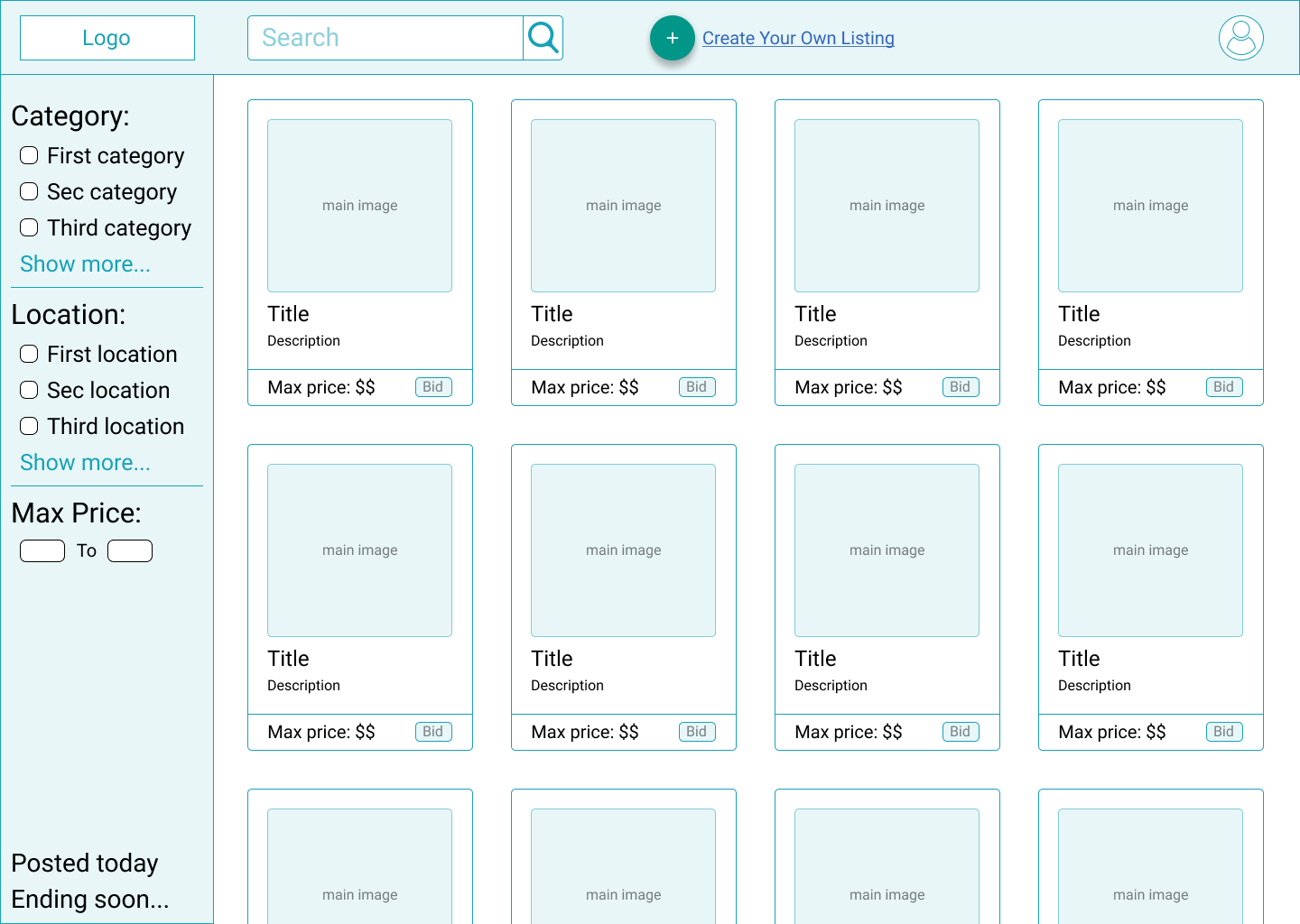
**Frontend**

Tο frontend αποτελείται από:

* Pages - containers
  + κάθε page έχει το δικό του url και το δικό του state, είναι ανεξάρτητο από το υπόλοιπα pages
* Components
  + Κάθε page αποτελείται από διάφορα components, μερικά components όπως φόρμες έχουνε δικό τους state
* Templates
  + Είναι υπεύθυνα για το structure σε κάθε σελίδα

Για παράδειγμα η αρχική σελίδα με path: «/listings» αποτελείτε από τα εξής components

* Το Navigation bar που βρίσκεται στην πάνω πλευρά της σελίδας
* Το Filters bar το οποίο περιέχει όλα τα φίλτρα που μπορεί ο χρήστης να χρησιμοποιήσει
* Τα listings που ανάλογα με το page που βρισκόμαστε και τα φίλτρα διαφέρουν
* Όλα αυτά για να έχουνε το σωστό structure χρησιμοποιούν το template το οποίο εμπεριέχει αυτά τα components (Μέσω των props περνάει πληροφορία για το τι θα περιέχει το κάθε component, έτσι ώστε να δημιουργείτε δυναμικά η σελίδα)



**Listing** component

**Navigation bar** component

**Filters bar** component

Κάθε component μπορεί να αποτελείτε από άλλα sub-components.

**Ανάλυση Συστήματος**

**Components**

Filters:

* FilterCategory.js
  + checkbox με όνομα και value
* Filters.js
  + όλη η μπάρα των φίλτρων

Listings:

* Listing.js
  + αποτελεί το κάθε «μεμονωμένο» listing
* ListingDetails.js
  + Όταν ο χρήστης ανοίξει το listing σε full page, αυτό το component εμφανίζει όλα τα στοιχεία του συγκεκριμένου listing
* NewListingField.js
  + Δημιουργεί δυναμικά το κάθε filed που πρέπει να περιέχετε στην σελίδα δημιουργίας νέου listing

Signup:

* SignupField.js
  + Δημιουργεί τα διάφορα πεδία που υπάρχουν στην σελίδα εγγραφής, εκτός πεδίων όπου ο χρήστης πρέπει να επιλέξει μέσω υπαρχόντων δεδομένων πχ dropdown menus
* SignupSelect.js
  + Δημιουργεί τα πεδία που είναι για την επιλογή δεδομένων από dropdown menus πχ. Customer ή professional account, Jobs, locations κλπ.

Timetable:

* Timetable.js
  + Δημιουργεί δυναμικά το calendar του professional

Statistics:

* AnaliticsChart.js
  + Δημιουργεί δυναμικά γραφικές παραστάσεις με διάφορα δεδομένα από τον professional πχ πόσα bids έκανε σε listings τον τελευταίο χρόνο

Bids:

* Bid.js
  + Δημιουργεί το κάθε πεδίο που υπάρχει στην φόρμα που συμπληρώνει ο professional
* BidForm.js
  + Δημιουργεί ολόκληρη την φόρμα
* BidsMade.js
  + Εμφανίζει στον customer τα bids που έγιναν στο listing του και του δίνει την δυνατότητα να επιλέξει κάποιο από αυτά

Navigation Bar:

* NavBar.js
  + Δημιουργεί το navigation bar

Reviews:

* Review.js
  + Εμφανίζει το review που έγινε από ένα user προς έναν άλλο
* ReviewForm.js
  + Δημιουργεί την φόρμα για να κάνει κάποιος user review μετά το τέλος ενός listing

Profile:

* UserDetails.js
  + Εμφανίζει τα στοιχεία του χρήστη

Γενικά components:

* Rating.js
  + Εμφανίζει το rating κάποιου χρήστη
* SuccessAlert.js
  + εμφανίζει ένα μήνυμα επιτυχίας πχ μετά από μια επιτυχής εγγραφή, μετά από την δημιουργία καινούριου listing, μετά από ένα bid ακόμα και μετά από edit κάποιου bid.

**Containers**

Pages:

* SigninPage.js
  + Σελίδα για signin
* SignupPage.js
  + Σελίδα για signup
* ListingsPage.js
  + Σελίδα με τα διαθέσιμα listings
* NewListingPage.js
  + Σελίδα για την δημιουργία καινούργιου listing
* FullListingPage.js
  + Παρουσίαση όλων τον πληροφοριών ενός listing
* ReviewsPage.js
  + Σελίδα για την προβολή των reviews που έγιναν και που απομένουν να γίνουν
* BidsPage.js
  + Σελίδα με τα bids τα οποία έγιναν και δυνατότητα επεξεργασίας τους
* UserProfilePage.js
  + Σελίδα με τα στοιχεία ενός χρήστη και τα reviews τα οποία έγιναν προς αυτόν
* TimetablePage.js
  + Σελίδα με το πρόγραμμα του χρήστη το οποίο δημιουργείτε δυναμικά με τα bids του που έχουν επιλεγεί από τους customers
* AnaliticsPage.js
  + Σελίδα με γραφικές παραστάσεις για στατιστικά του χρήστη, δυνατότητα επιλογής χρονικού διαστήματος

**Templates**

Templates:

* SimpleTemplatePage.js
  + Περιέχει το navigation bar, χρησιμοποιείτε για το NewListingPage, AnaliticsPage κλπ.
* TemplatePage.js
  + Περιέχει το navigation bar και τα φίλτρα, χρησιμοποιείτε για το Homepage, ReviewsPage, BidsPage κλπ.
* FullTemplatePage.js
  + Ίδιο με το TemplatePage με μόνη διαφορά στο margin μεταξύ του navigation bar και του περιεχομένου της σελίδας, χρησιμοποιείτε στο UserProfilePage

**State**

Global State – Context API:

* Auth-Context.js
  + Περιέχει σε ένα global state τα στοιχεία και το token του χρήστη που έκανε login (Αρχικοποιείται στο App.js)

Local State:

* Κάθε page και κάθε component που περιέχει κάποια φόρμα κρατάει ένα local state. Για παράδειγμα το ListingPage πρέπει να κράτα πληροφορία όπως:
  + Ένα πίνακα με τα listings που πρέπει να εμφανιστούν
  + Αν πατήθηκε κάποιο listing και ποιο είναι αυτό
  + Τα φίλτρα που πρέπει να περιέχει η σελίδα κλπ.

**Routing**

Page Routing:

* App.js
  + Προσφέρει το routing μεταξύ των σελίδων, προσδιορίζει για κάθε path πιο από τα pages που υπάρχουν πρέπει να γίνει render.

**BackEnd**

Το Backend υλοποιεί ένα REST API το οποίο υποστηρίζει το πιο κάτω methods:

* GET
  + για να πάρουμε ένα resource ή ένα collection από resources
* POST
  + για δημιουργήσουμε ένα resource
* PUT
  + για να κάνουμε update ένα resource
* DELETE
  + για να διαγράψουμε ένα resource

(Για περισσότερη ανάλυση του rest api υπάρχει το αρχείο rest api manual)

Αποτελείτε από τα εξής packages

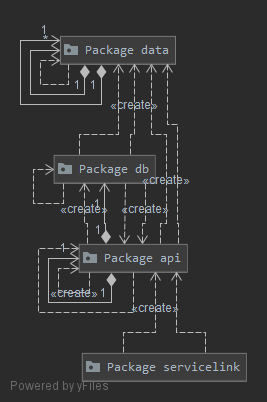
* ServiceLink (main)
  + Περιέχει όλα τα methods μαζί με τα routes που υποστηρίζονται από το REST API, Κάνει redirect το κάθε request στο αντίστοιχο Api. Όταν πάρει την απάντηση από το Api την επιστρέφει στον client
* Data
  + Περιέχει τις κλάσεις που χρησιμοποιούνται για να κρατάνε temporary δεδομένα από και προς την βάση δεδομένων
* Api
  + Λειτουργεί σαν controller μεταξύ του ServiceLink (main) και DB, παίρνει τα requests που γίνονται redirect από την main, ελέγχει αν είναι valid τα στοιχεία και το token και καλεί με την σειρά του μεθόδους από το package DB. Έτσι ετοιμάζει την απάντηση την οποία θα επιστρέψει στην main
* DB
  + Ετοιμάζει και κάνει τα queries προς την βάση δεδομένων, όταν πάρει την απάντηση με την χρήση του package data αποθηκεύει temporary δεδομένα σε maps ή σε απλά objects και τα επιστρέφει στο Api

Κάθε Package αποτελείτε από της εξής κλάσεις

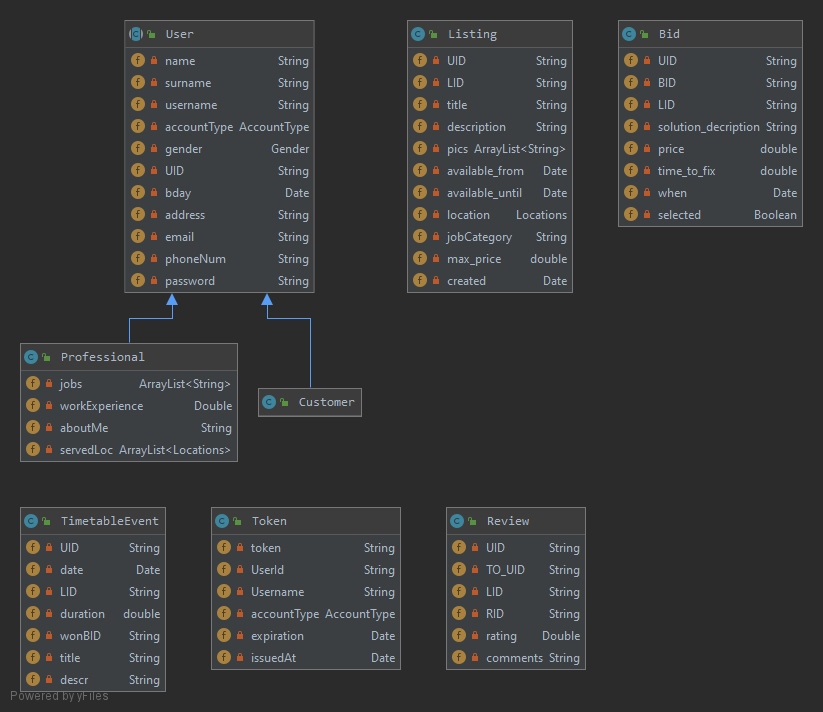
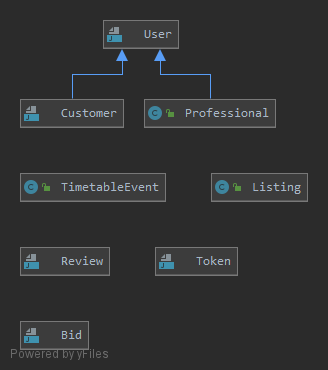
* Package ServiceLink (main)
  + ServiceLink.java
* Package Data
  + User.java (abstract)
  + Professional.java (extends User)
  + Customer.java (extends User)
  + Bid.java
  + Listing.java
  + Review.java
  + TimetableEvent.java
  + Token.java
* Package Api
  + DataApi.java (interface)
  + UserAPI.java (implements UserAPI)
  + BidAPI.java (implements DataApi)
  + ListingAPI.java (implements ListingAPI)
  + ReviewsAPI.java (implements ReviewsAPI)
  + AuthorizerApi.java
  + JwtAPI.java
  + ServerResponseAPI.java
  + StatisticsAPI.java
  + TimetableAPI.java
* Package DB
  + DB.java (interface)
  + UserDB.java (implements DB)
  + BidDB.java (implements DB)
  + ListingDB.java (implements DB)
  + ReviewDB.java (implements DB)
  + ConnectionDB.java
  + Statistics.java
  + Timetable.java

**Διαγράμματα UML**

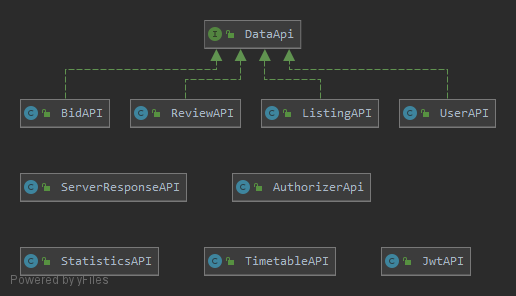
**Packages**

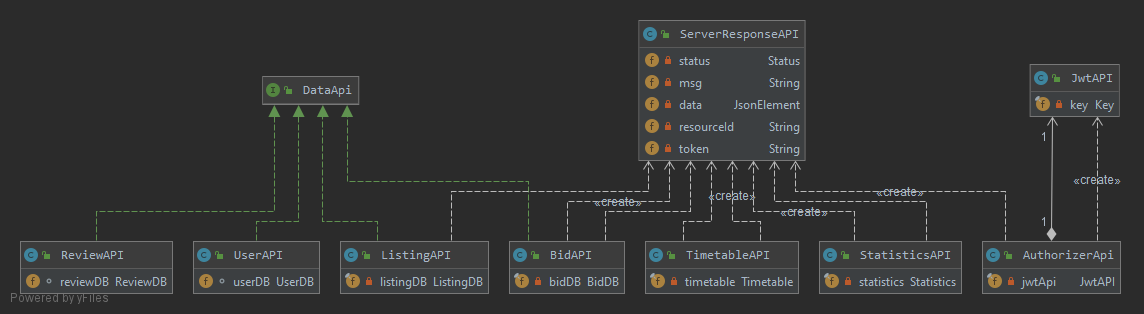


**Package Data**

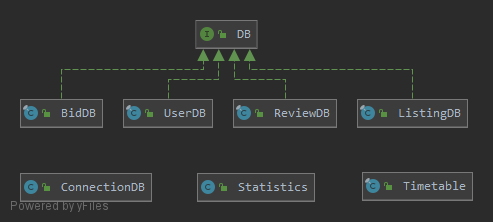


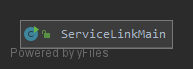
**Package API**

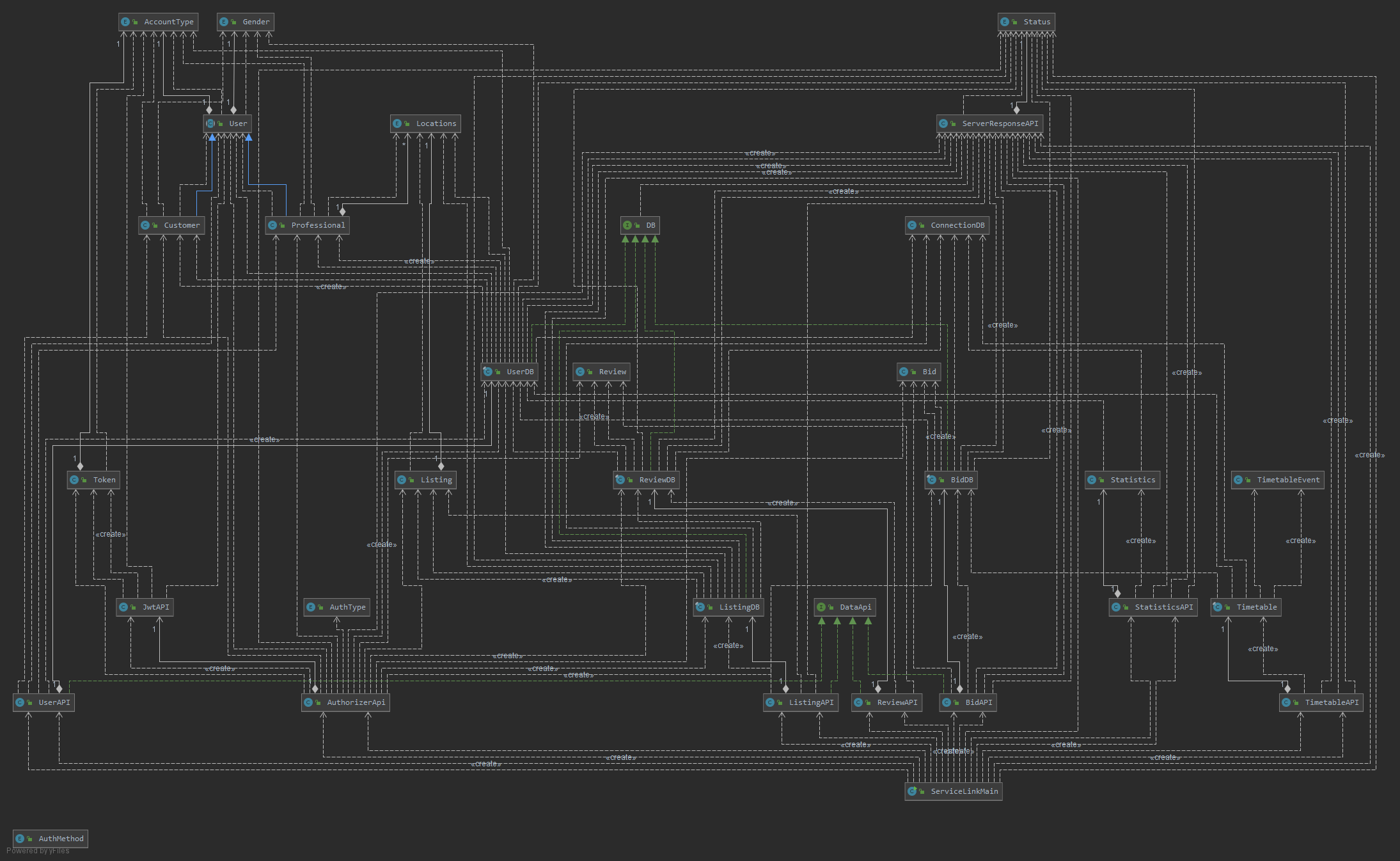




**Package DB**



**Package ServiceLink**

**Η Συνολική Εικόνα**

**Database**

Η Βάση δεδομένων αποτελείτε από 5 βασικές οντότητες

* Professional
* Customer
* Bid
* Listing
* Review

Παρότι η βάση δεδομένων αποτελείτε από μόνο 5 οντότητες, παρέχει επιπλέον δεδομένα τα οποία εξάγονται από την επεξεργασία των ήδη υπάρχοντών δεδομένων.

Για παράδειγμα το πρόγραμμα των professionals δημιουργείτε δυναμικά, αφού ξέρουμε ότι αποτελείτε από τα bids που έκανε κάποιος professional και επιλέγηκε από κάποιο customer ο οποίος δημιούργησε το listing. Στο κάθε bid περιέχονται πληροφορίες όπως μέρα και ώρα συνάντησης, κόστος επιδιόρθωσής και στο listing περιοχή, όνομα customer κλπ. Με το join των δύο πινάκων έχουμε αρκετή πληροφορία για να δημιουργήσουμε δυναμικά το πρόγραμμα του professional.

Παρομοίως εξάγονται και τα στατιστικά που αφορούνε κάθε professional.

**Οντότητες με τα πεδία τους**

User:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UID | NAME | SURNAME | USERNAME | ACCOUNT\_TYPE | GENDER | BDAY | ADDRESS | EMAIL | PHONE\_NUM |

Customer extends User:

|  |
| --- |
|  |

Professional extends User:

|  |  |  |  |
| --- | --- | --- | --- |
| UID | JOB | WORK\_EXPERIENCE | SLOCATIONS |

Review:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UID | TO\_UID | RID | RATING | COMMENTS |

Listing:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UID | LID | TITLE | DESCRIPTION | PICS | AVAILABLE\_FROM | AVAILABLE\_UNTIL | LOCATION | JOB\_CATEGORY | MAX\_PRICE |

Bid:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UID | BID | SOLUTION\_DESC | PRICE | WHEN | TIME\_TO\_FIX\_IT | SELECTED |

Timetable Event:

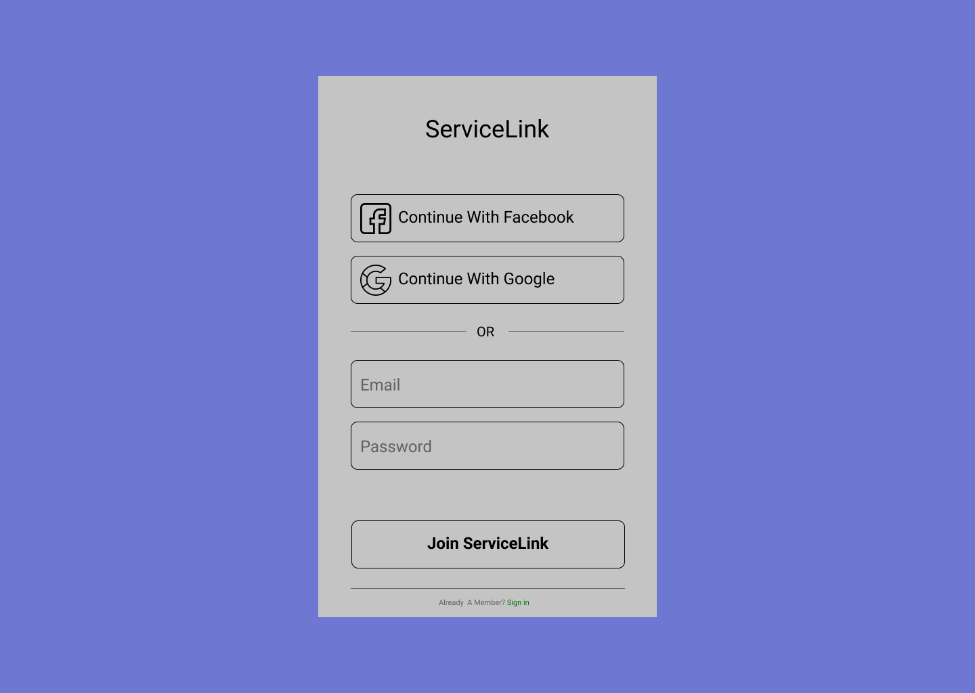
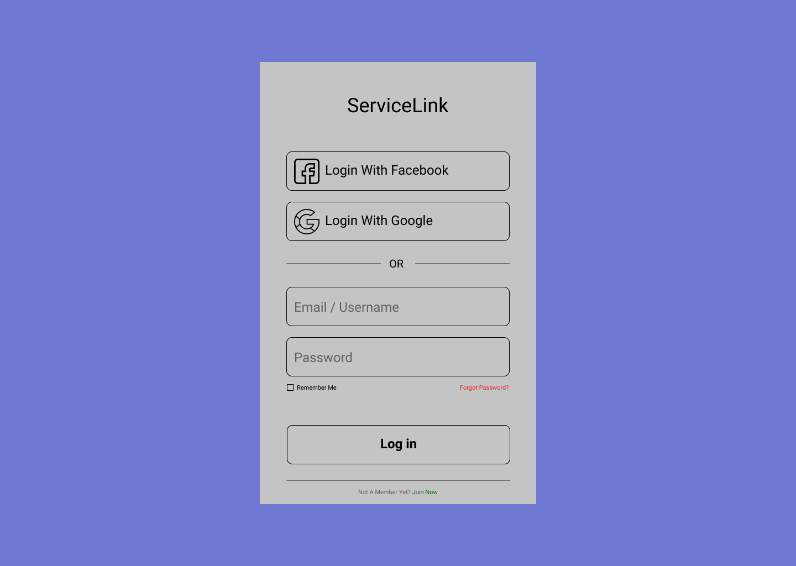
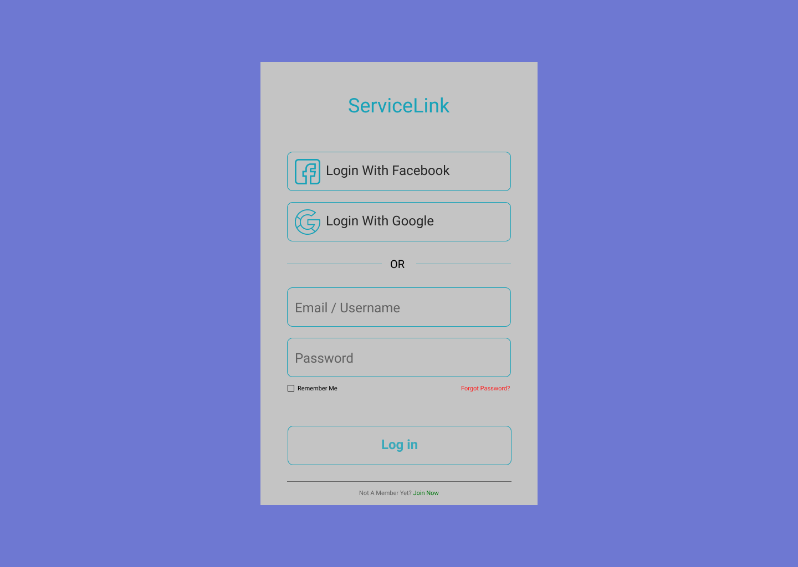
|  |  |  |  |
| --- | --- | --- | --- |
| LID | UID | DATE | DURATION |

**Σχεδίαση UI**

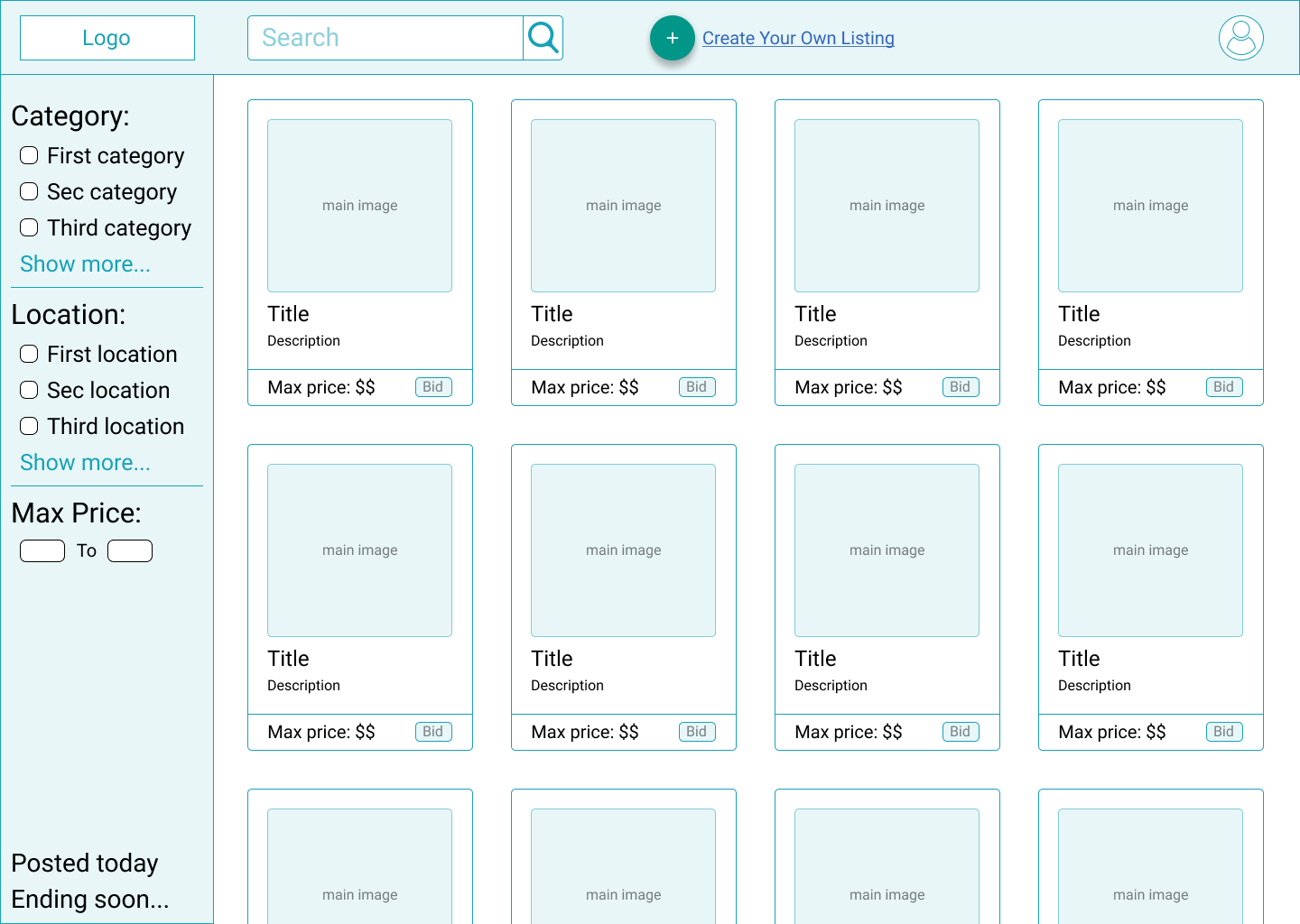
Η σχεδίαση των mockups έγινε στο figma πριν ξεκινήσει η υλοποίηση του frontend. Τα mockups δεν φτιάχτηκαν με αρκετή ακρίβεια και κατά την υλοποίηση έγιναν αρκετές αλλαγές, με επακόλουθο το τελικό αποτέλεσμα να διαφέρει αρκετά με τα αρχικά mockups.

**Signin – Signup**

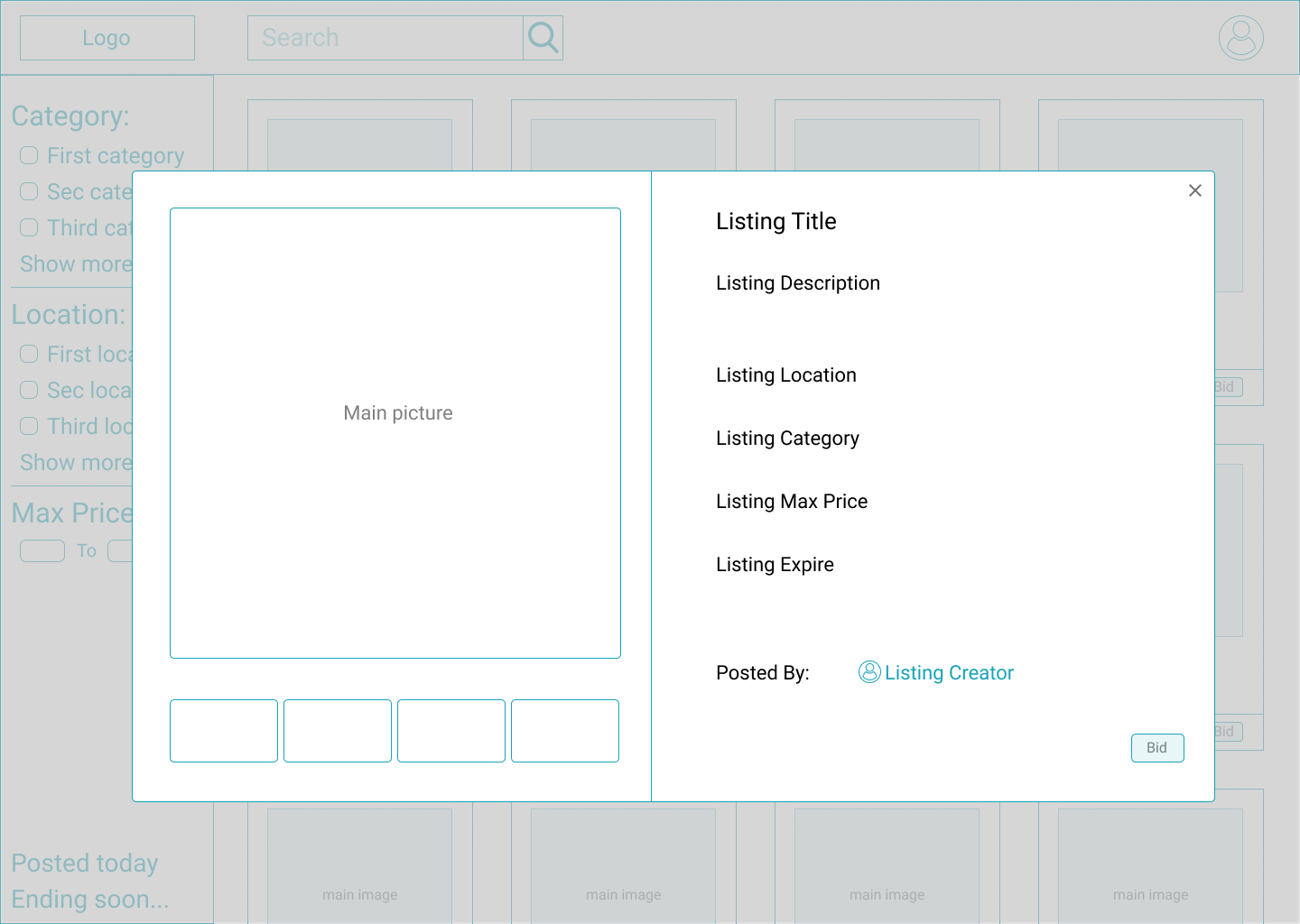
Αρχικά υπήρξε η ιδέα, σε περίπτωση που ο χρήστης δεν ήθελε να δημιουργήσει καινούριο account για το ServiceLink, να του δίνετε η δυνατότητα να κάνει login μέσω του ήδη υπάρχοντος account του στο facebook ή στο google με την χρήση της βιβλιοθήκης (OAuth 2.0), η οποία κάνει redirect τον χρήστη στην αντίστοιχή σελίδα και δημιουργεί ένα jwt token με το οποίο μπορούμε να έχουμε πρόσβαση στα στοιχεία του χρήστη.



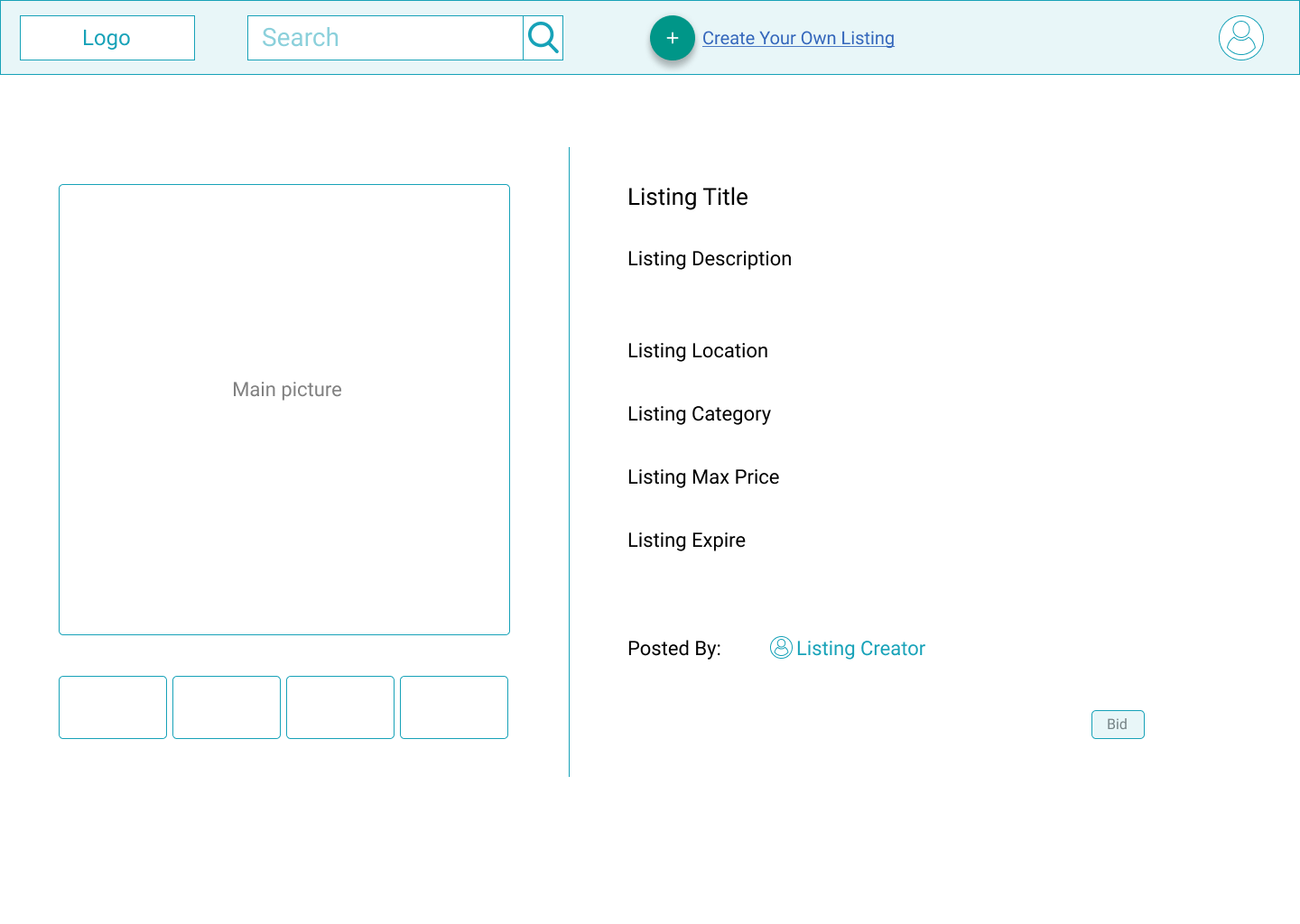
**Homepage**



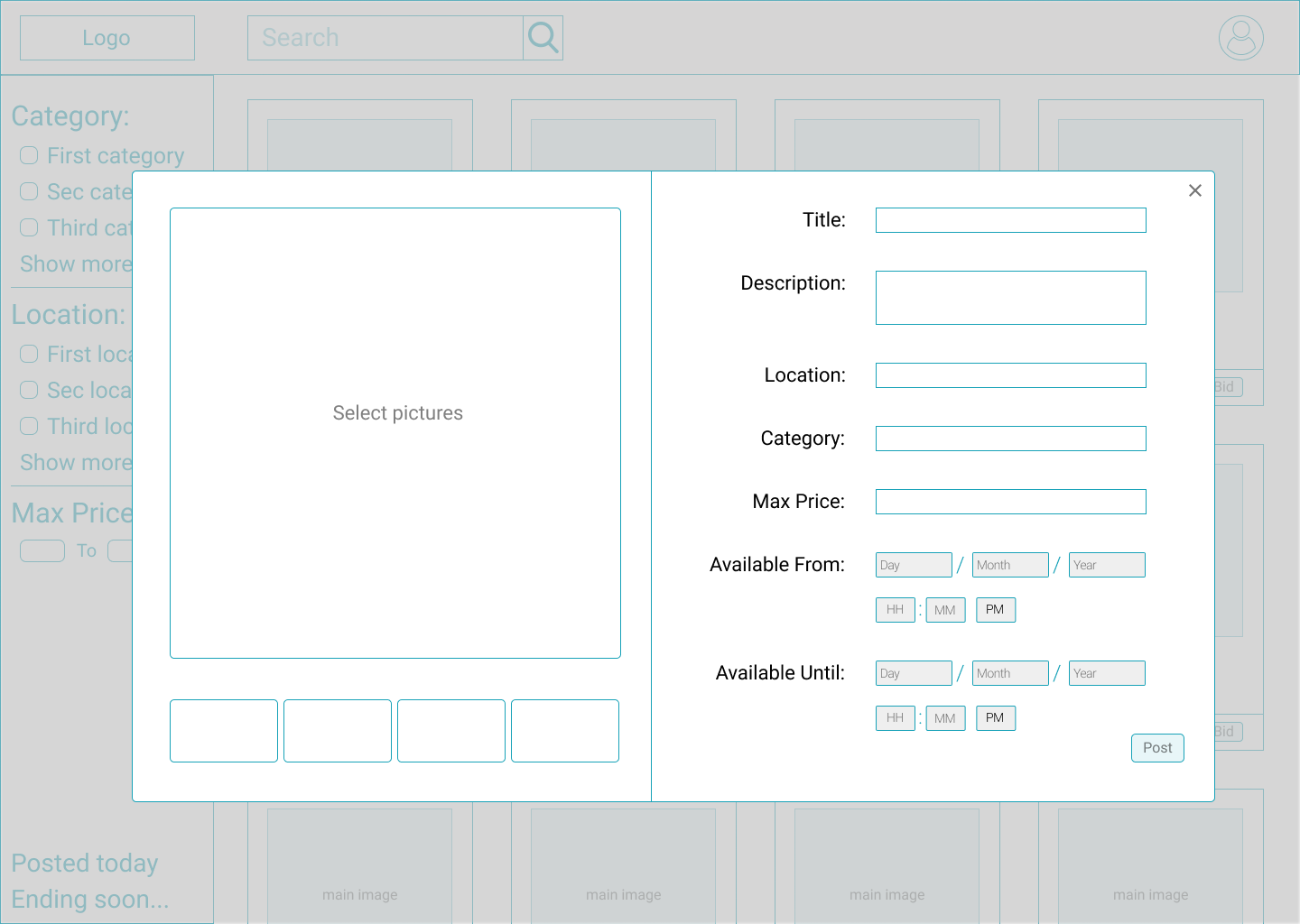
**Full Listing (v1)**



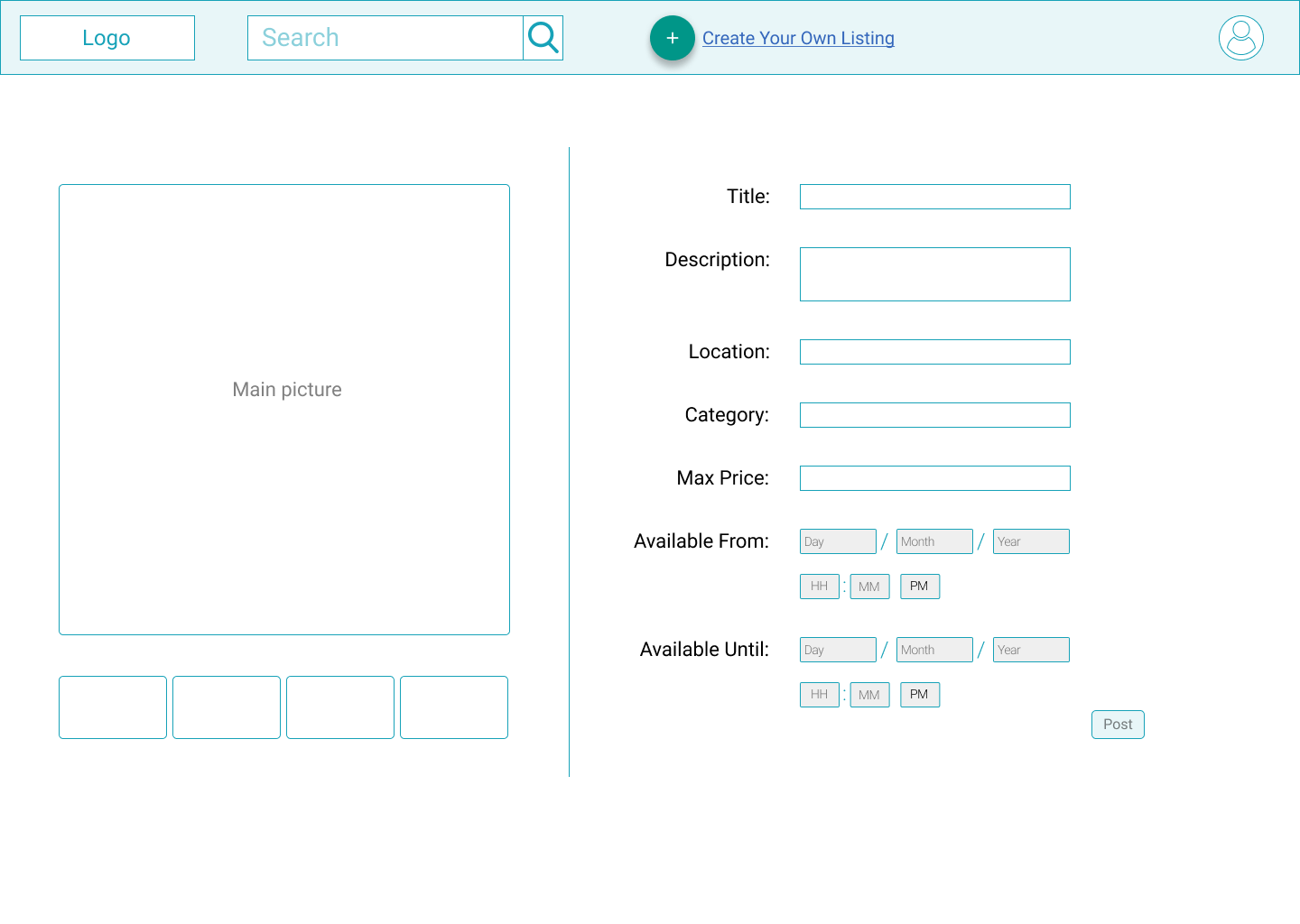
**Full Listing (v2)**



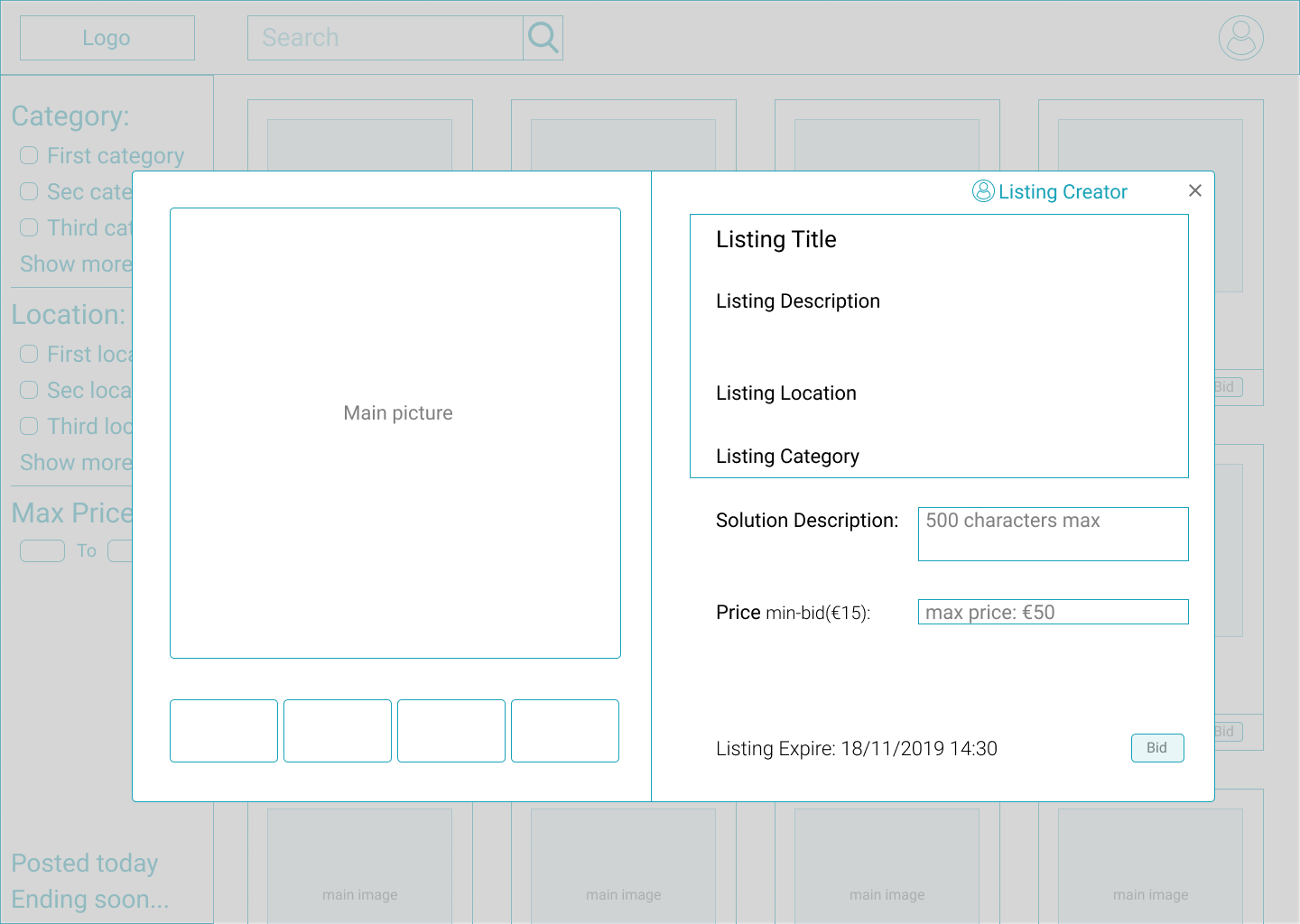
**Create Listing Form (v1)**



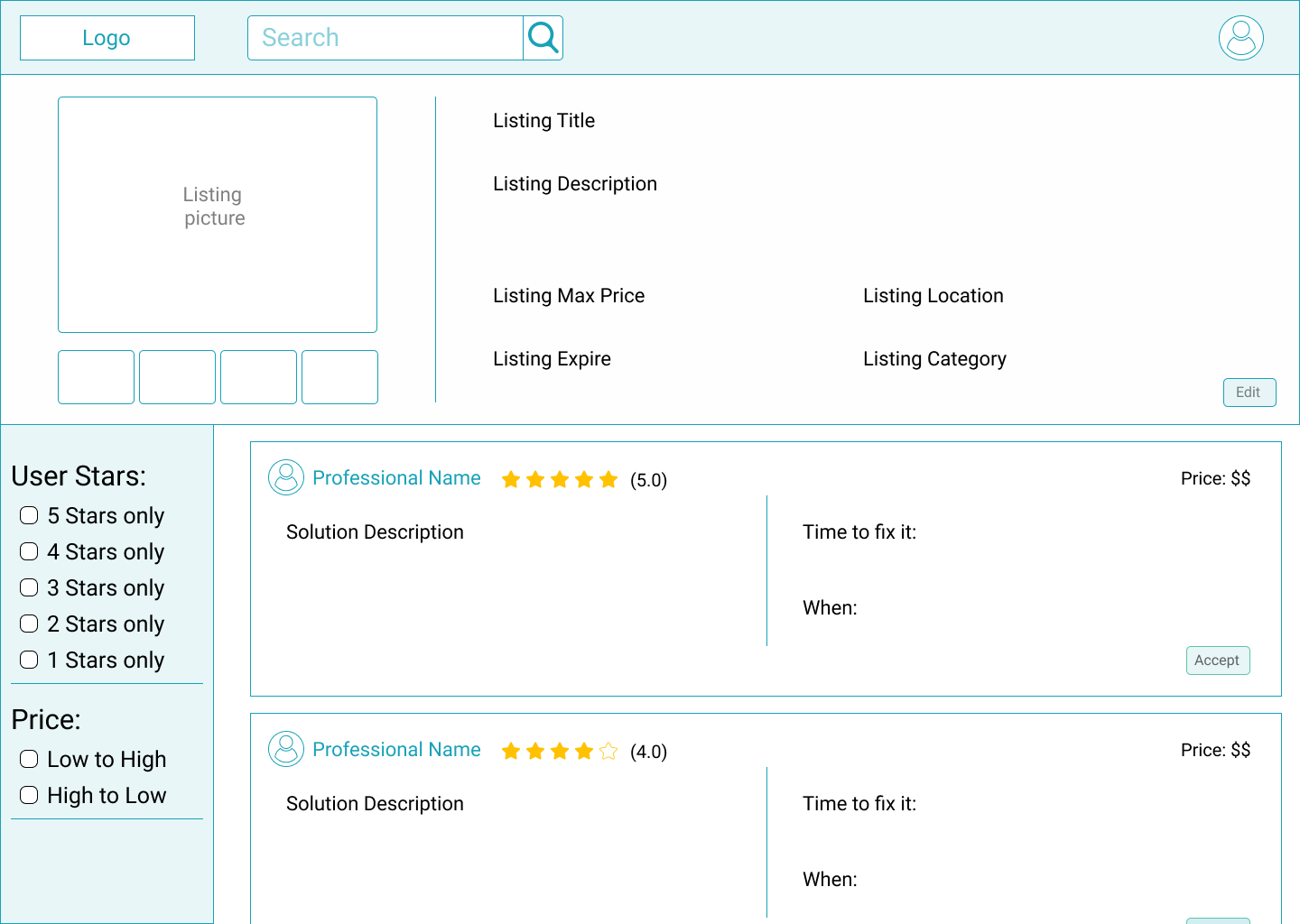
**Create Listing Form (v2)**



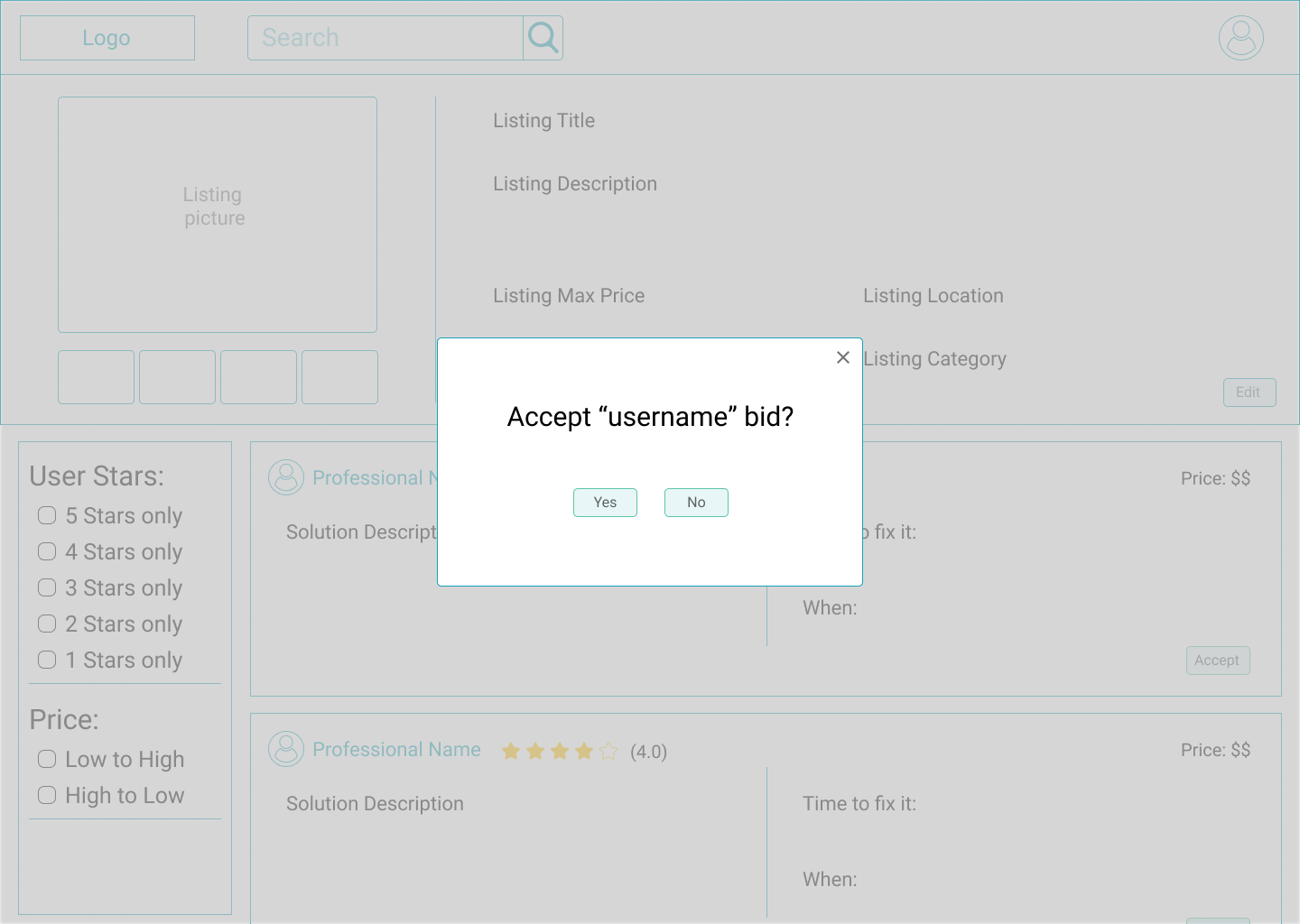
**Bid Form**



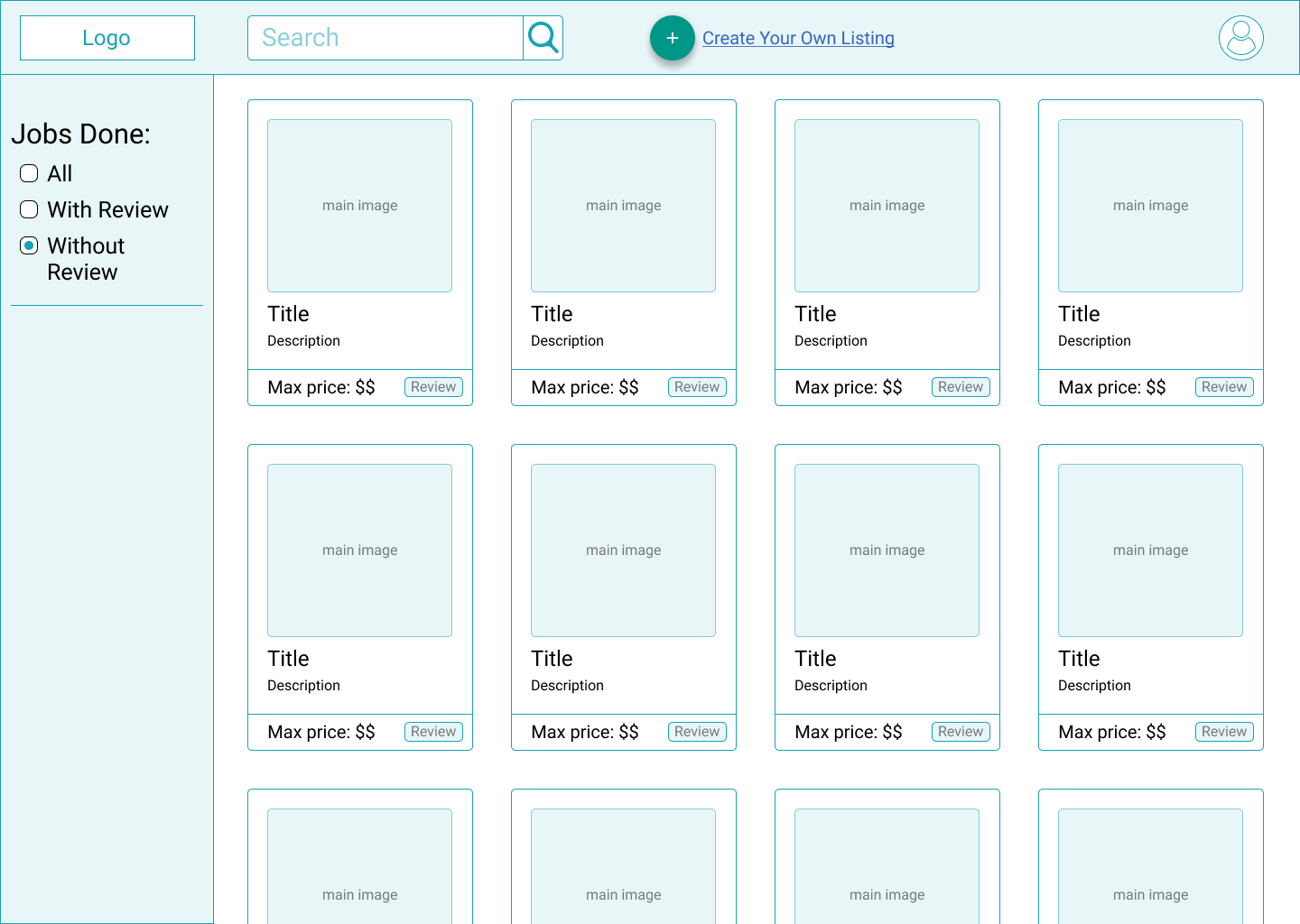
**Select Bid**



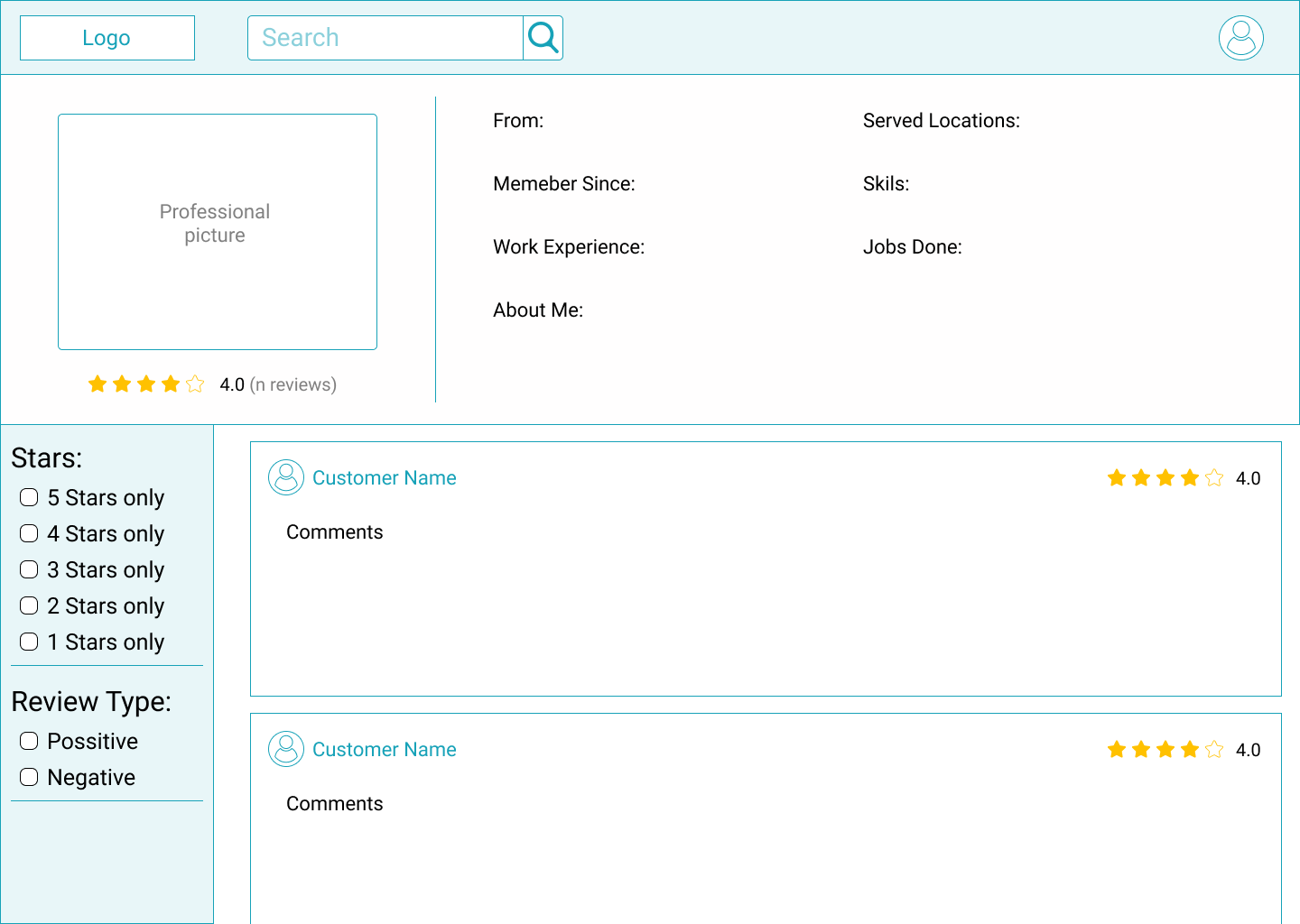
**Selected Bid Confirmation**



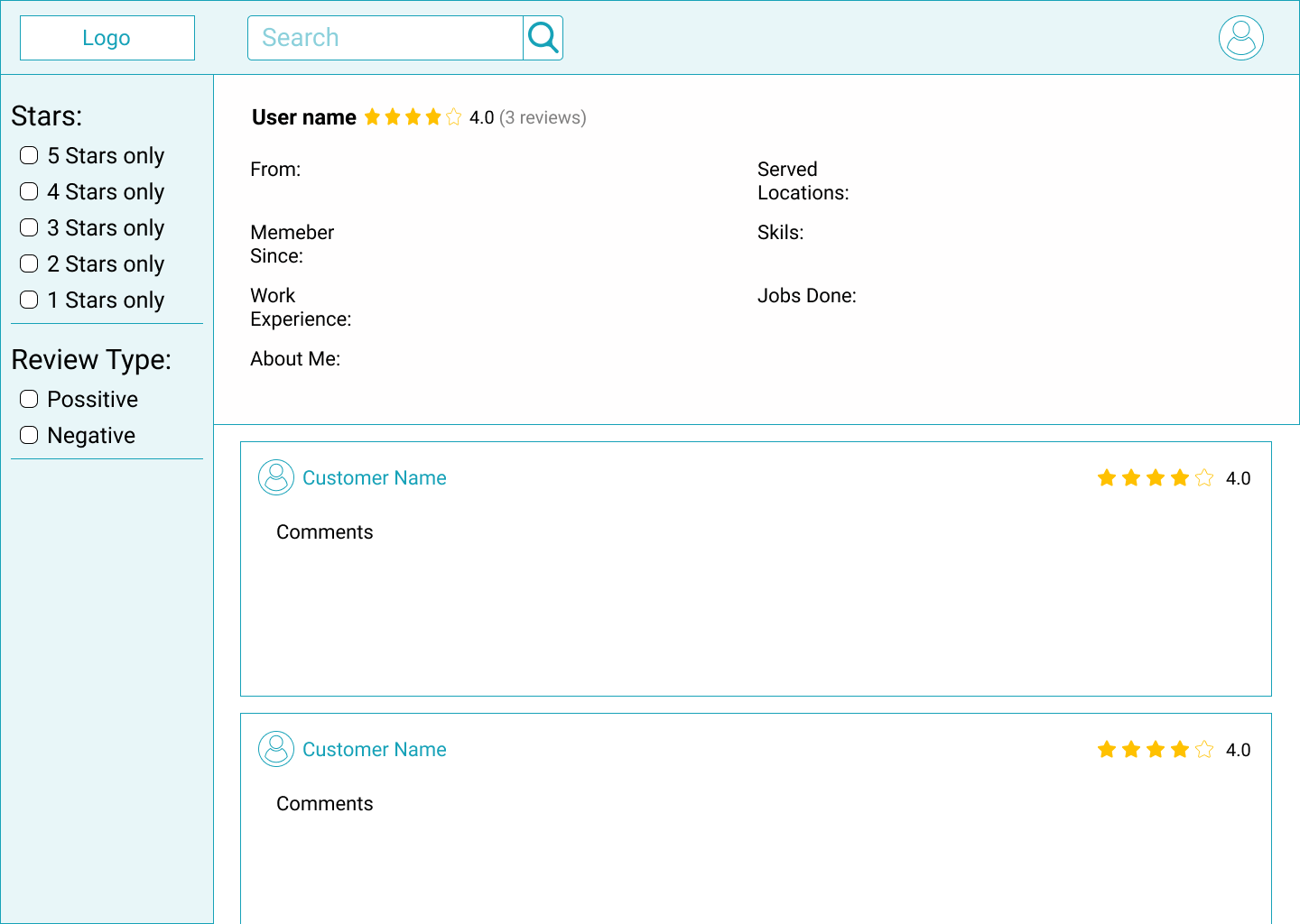
**Make Review**



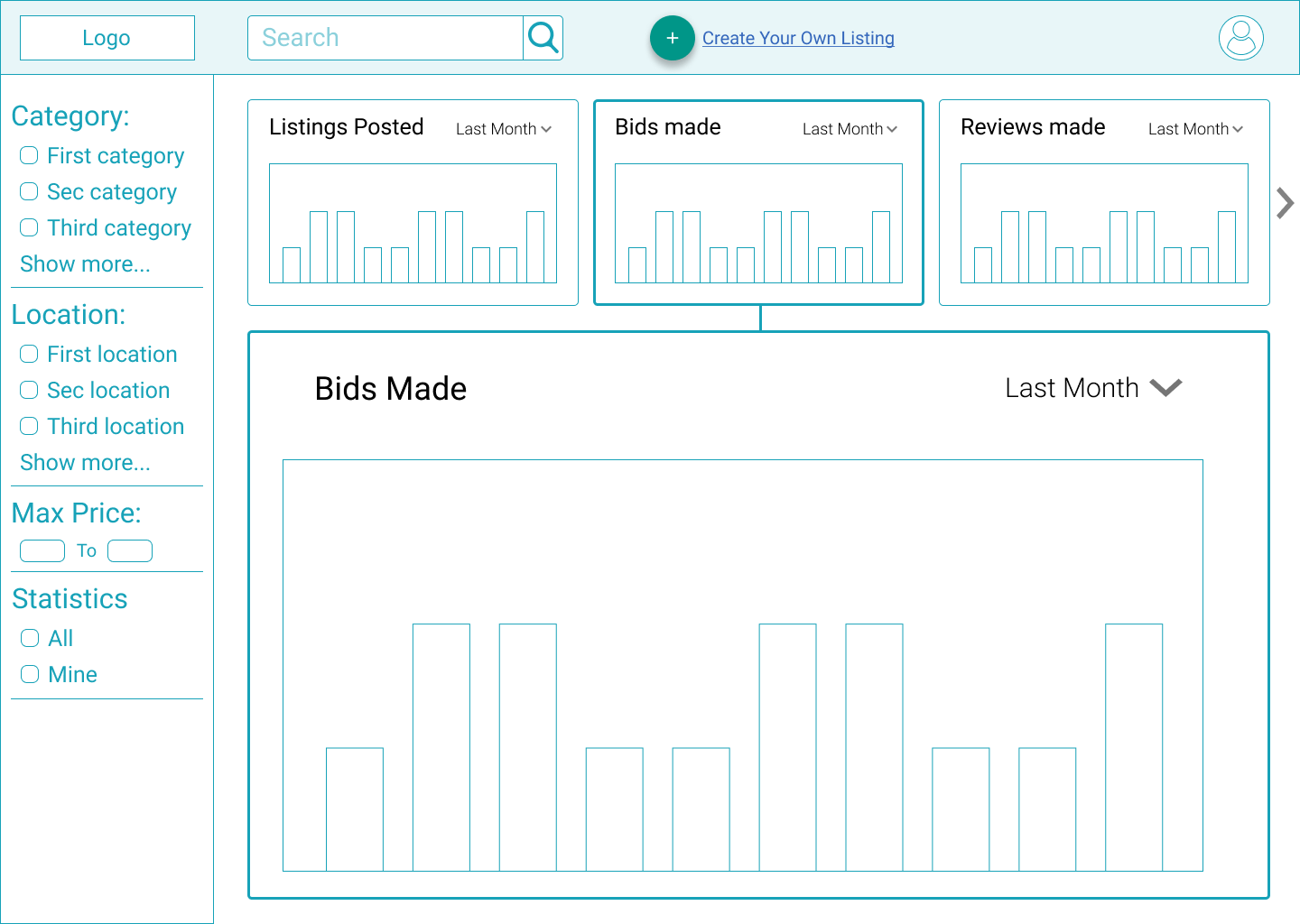
**Professional Profile (v1)**



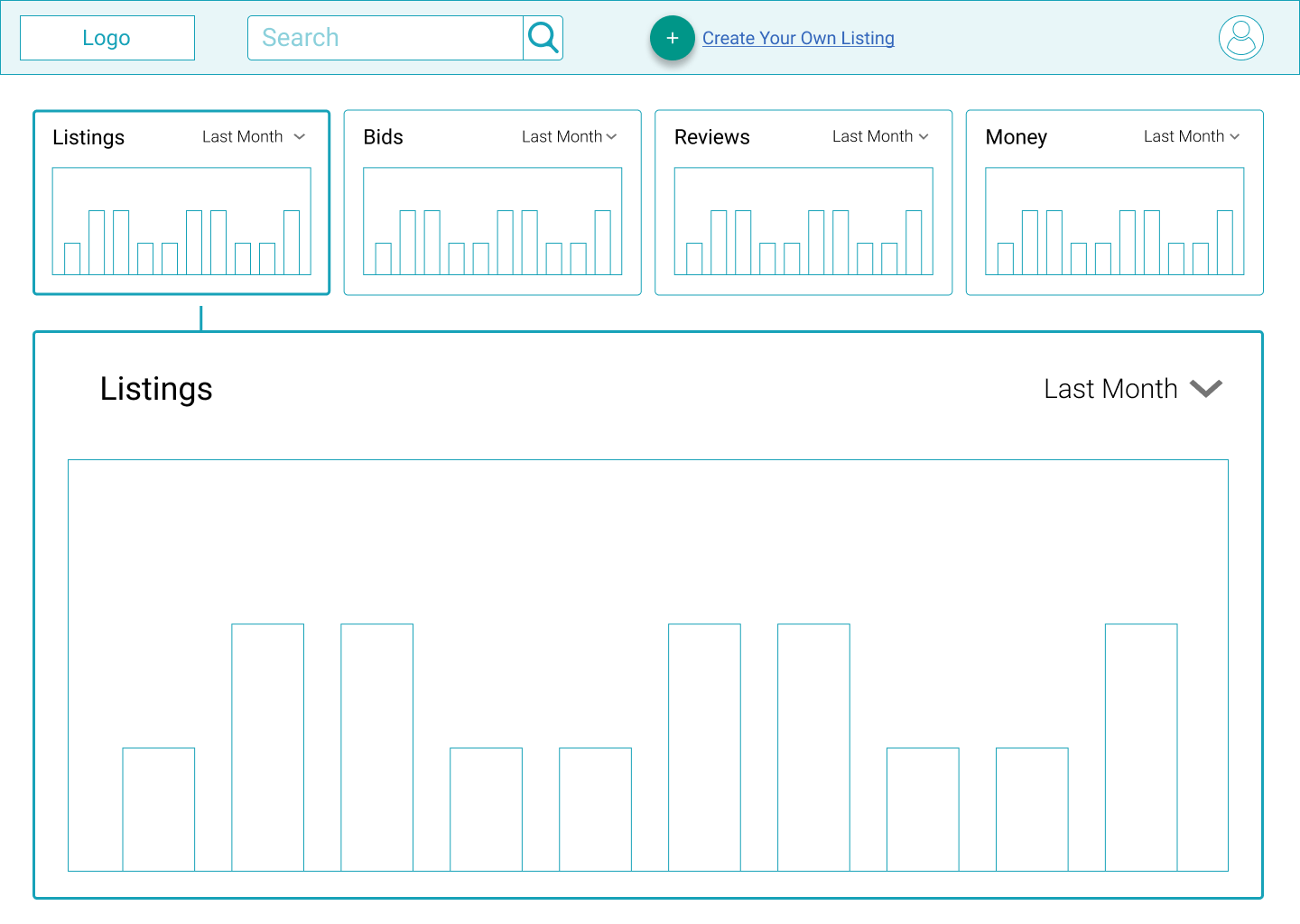
**Professional Profile (v2)**



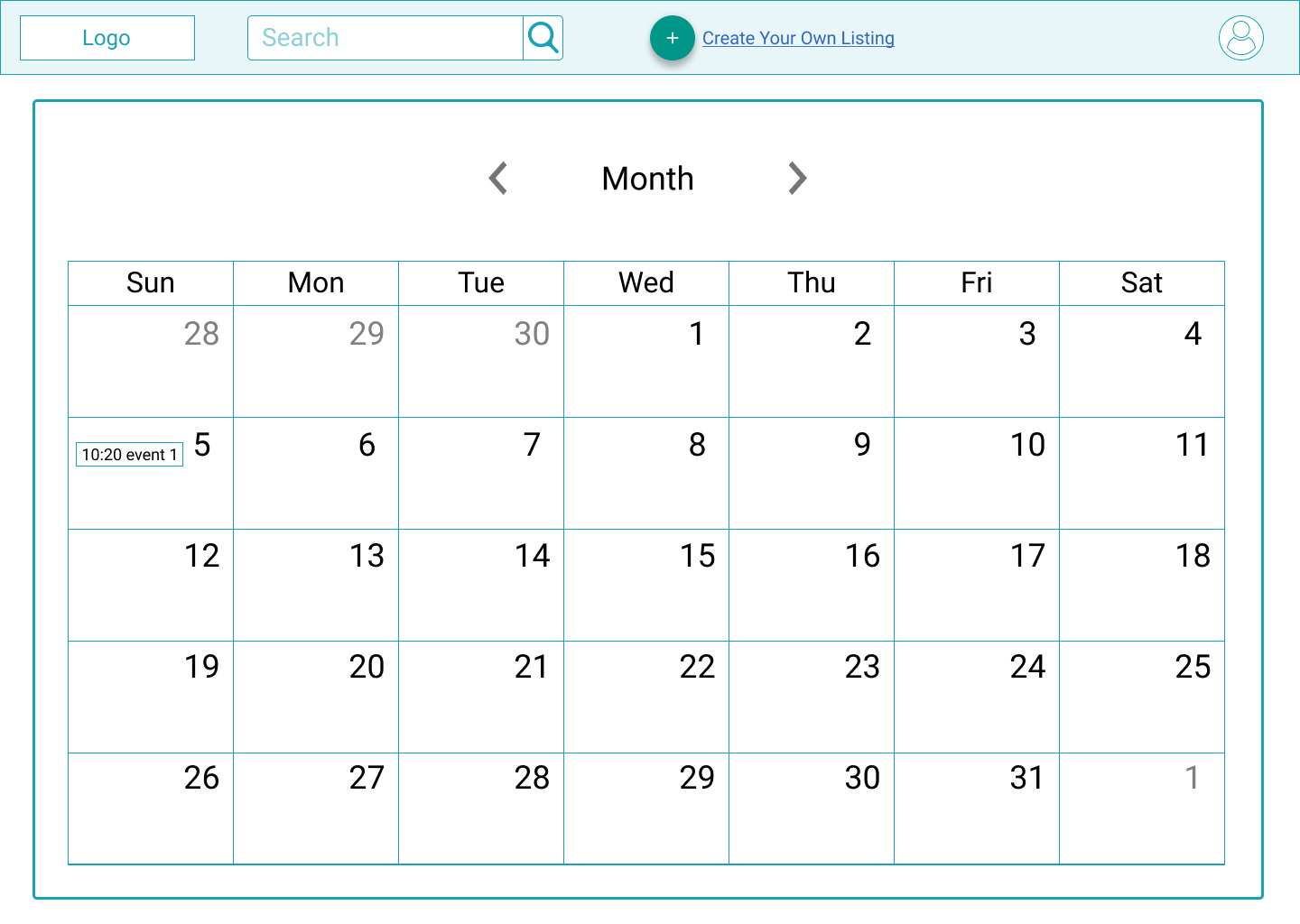
**Statistics (v1)**



**Statistics (v2)**



**Timetable**



**Υλοποίηση**

Γενικά το σύστημα όπως φαίνεται και πιο πάνω, χωρίζεται σε τρία μέρη, το frontend που είναι υπεύθυνο για το UI του συστήματος, το backend που είναι stateless και αναλαμβάνει όλη την επεξεργασία των request που φτάνουν στον server και τη βάση δεδομένων η οποία αποτελείτε από 5 οντότητες και περιέχει όλα τα δεδομένα του συστήματος.

Για την υλοποίηση του συστήματος χρησιμοποιήθηκαν διάφορα εργαλεία. Για το frontend χρησιμοποιήθηκε η βιβλιοθήκη ReactJS, για το REST API η βιβλιοθήκη Spark Java και για την βάση δεδομένων η PostgreSQL. Επιπρόσθετα στο frontend έγινε χρήση μερικών επιπλέων βιβλιοθηκών για τα στατιστικά και για το δυναμικό πρόγραμμα των professionals.

**Frontend**

Το frontend χωρίζεται σε 4 κομμάτια, το πρώτο αφορά τα components του συστήματος, το δεύτερο τα pages και templates του συστήματος, τα pages αποτελούνται από διάφορα components και τα templates έχουν σκοπό να στοιχίζουν σωστά τα components σε κάθε page, το τρίτο κομμάτι αφορά τo global state (χρήση context api) το οποίο περιέχει δεδομένα του χρήστη που είναι απαραίτητα σε όλα τα pages (όπως το token, το username του κλπ.) και το τέταρτο κομμάτι αφορά το routing μεταξύ των pages(συσχετίζει τα URLs με τα pages που πρέπει να γίνονται render)

**Backend**

Το backend χωρίζεται και πάλι σε 4 κομμάτια, το πρώτο κομμάτι περιέχει όλα τα routes, λαμβάνει το request και απλά το κάνει redirect στο σωστό api(δεύτερο κομμάτι)για να εξυπηρετηθεί. Το api είναι υπεύθυνο για τον έλεγχο, πχ αν ο user είναι authenticated, αν το token που έχει σταλεί είναι valid, αν έχει πρόσβαση στο resource που ζητά, αν είναι όλα εντάξει κάνει τα requests που χρειάζονται προς την βάση δεδομένων(τρίτο κομμάτι) και περιμένει το response της βάσης με τα δεδομένα τα οποία μετατρέπει σε json και τα επιστρέφει στον χρήστη. Η βάση δεδομένων εκτελεί τα διάφορα queries, μετατρέπει τα δεδομένα σε java objects με βάση της κλάσεις που υπάρχουν στο package data(τέταρτο κομμάτι) και τα επιστρέφει στο api. Τα data είναι οι κλάσεις που χρησιμοποιούνται προσωρινά για αναπαράσταση των δεδομένων που υπάρχουν στην βάση.

**Database**

H βάση δεδομένων αποτελείτε από 5 οντότητες. Είναι υπεύθυνη για την σωστή εκτέλεση των queries και για την μετατροπή των δεδομένων της βάσης σε java objects ή collections από objects. Ανάλογα με το request επιστρέφει ένα map με τα δεδομένα ή απευθείας ένα ServerResponse object το οποίο περιέχει όλη την πληροφορία που χρειάζεται.

**Προβλήματα - Δυσκολίες**

Κατά την διάρκεια της πτυχιακής εργασίας προέκυψαν αρκετά προβλήματα και δυσκολίες. Ενώ στο κομμάτι τις ανάλυσης απαιτήσεων δεν υπήρξε κάποια ιδιαίτερη δυσκολία, οι πρώτες δυσκολίες προέκυψαν κυρίως στο χωρισμό των λειτουργιών (τι λειτουργίες θα γίνονταν στο frontend, ποιες στο backend, τι οντότητες και πεδία θα υπάρχουν στην βάση δεδομένων κλπ.). Λόγο της μικρής εμπειρίας μου στην σχεδίαση μεγαλύτερων συστημάτων η δομή του συστήματος άλλαξε αρκετές φορές μέχρι να φτάσει σε μία σχετικά οργανωμένη μορφή, η οποία να είναι πιο απλή και να υποστηρίζει πιο εύκολα προσθήκες και αλλαγές. Στο frontend η αρχική δυσκολία ήταν η εξοικείωση με την βιβλιοθήκη react, μου πήρε αρκετό χρόνο μέχρι να μάθω την βιβλιοθήκη και να συνηθίσω το «νέο» τρόπο σκέψης. Ο διαχωρισμός των components, το global και local state που έπρεπε να περιέχουν μερικά components και η αλληλεπίδραση μεταξύ των components ήταν επίσης κάτι αρκετά χρονοβόρο, αφού κατά την διάρκεια της υλοποίησης χρειάστηκε να γίνουν αρκετές αλλαγές για να φτάσει σε ένα ικανοποιητικό σημείο.

Γενικά κατά την υλοποίηση παρουσιάστηκαν προβλήματα και δυσκολίες που δεν είχαν υπολογιστεί από την φάση της ανάλυσης. Επίσης μικρά λάθη, απροσεξίες, αλλαγές που επηρέαζαν έμμεσα άλλα σημεία στον κώδικα και ελλιπής testing, ήταν αιτίες για να χαθούνε αρκετές ώρες στο debugging.

**Μελλοντικές Προεκτάσεις**

Σαν σύστημα μπορεί να υποστηρίξει μελλοντικά αρκετές νέες λειτουργίες και να βελτιώσει ήδη υπάρχουσες. Μερικές νέες λειτουργίες θα μπορούσε να είναι

Αναδημοσίευση listings από άλλους customers, δεδομένου ότι το listing έχει λήξει και έχει γίνει η απαραίτητη επεξεργασία πριν την αναδημοσίευση.

Μηνιαία challenges μεταξύ των professionals, πχ όποιος έχει τα περισσότερα bids που επιλέγηκαν ή τα καλύτερα reviews ή το μεγαλύτερο κέρδος.

Οι ειδοποιήσεις να μπορούν αποστέλλονται μέσω email, για να μπορεί ο χρήστης να είναι ενήμερος χωρίς να χρειάζεται να συνδεθεί στο ServiceLink.

Τα customer accounts να μπορούν να γίνουν upgrade σε professional accounts, ή να υπάρχουν accounts που να συνδυάζουν τις λειτουργίες των 2 accounts για να μπορεί κάποιος user και να ανεβάζει νέα listings αλλά και να κάνει bids σε listings από άλλους users.

Η πιο σημαντική προέκταση για το σύστημα, θα είναι να φτιαχτεί ένα app για το κινητό (κάτι το οποίο είχα στα σχέδια της πτυχιακής, αλλά δεν πρόλαβα να υλοποιήσω). Από το κινητό θα είναι ακόμη πιο εύκολη η χρήση, αφού θα μπορεί άμεσα ο χρήστης να βγάλει φωτογραφίες μέσω του app, επίσης θα μπορεί να ενημερώνετε άμεσα για την κατάσταση των listings που έφτιαξε, τα bids ή τα reviews που έγιναν τώρα, επιπρόσθετα για τους professionals μπορούν να υπάρχουν ειδοποιήσεις ηχητικές και γραπτές για την επόμενη συνάντηση που υπάρχει στο πρόγραμμα τους, όπως επίσης και ενημέρωση κάθε τέλος της βδομάδας με στατιστικά στοιχεία για το πώς πήγε η βδομάδα σε σχέση με τις προηγούμενες.

Μερικά mockups που ξεκίνησα να σχεδιάζω στο figma για το αντίστοιχο app για smartphone

