



**ΕΠΑνεΚ 2014-2020**  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ  
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ  
ΚΑΙΝΟΤΟΜΙΑ



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

# ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ

**ΔΡΑΣΗ ΕΘΝΙΚΗΣ ΕΜΒΕΛΕΙΑΣ:  
«Ερευνώ-Δημιουργώ-Καινοτομώ»**

## ΒΕΛΟΣ

**Βελτιστοποίηση Μεταφορών με Αλγόριθμους Ελέγχου  
Οπισθοπίεσης**

## Π2.2

### Αρχιτεκτονική και σχεδιασμός του συστήματος

<b>Ημερομηνία</b>	Μήνας 15, 28/10/2022 Μήνας 18, 28/01/2023
<b>Ενότητα Εργασίας</b>	Ε.Ε.2
<b>Είδος</b>	Αναφορά
<b>Κατάσταση έγκρισης</b>	Υποβληθέν
<b>Έκδοση</b>	3
<b>Αριθμός σελίδων</b>	80
<b>Όνομα αρχείου</b>	VELOS_Paradoteo_2.2_v03_final.docx
<b>Σύνοψη</b>	Στο Παραδοτέο «Π2.2- Αρχιτεκτονική και σχεδιασμός του συστήματος» παρουσιάζεται το περίγραμμα της αρχιτεκτονικής και ο συνολικός σχεδιασμός του συστήματος του έργου «ΒΕΛΟΣ».



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Ταμείο  
Περιφερειακής Ανάπτυξης

**ΕΠΑνεΚ** 2014-2020  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ  
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ  
ΚΑΙΝΟΤΟΜΙΑ



ανάπτυξη - εργασία - αλληλεγγύη

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Αυτή η σελίδα αφέθηκε σκόπιμα κενή



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Ταμείο  
Περιφερειακής Ανάπτυξης

**ΕΠΑνεΚ** 2014-2020  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ  
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ  
ΚΑΙΝΟΤΟΜΙΑ



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

## Ιστορικό εκδόσεων

Έκδοση	Ημερομηνία	Περιγραφή	Αναθεώρηση
1	18/09/2021	Πρότυπο παραδοτέου	
2	28/10/2022	Ενδιάμεση έκδοση	
3	28/01/2023	Συνολική επιμέλεια κειμένου, τελικό παραδοτέο	

## Λίστα συγγραφέων

Οργανισμός	Όνοματεπώνυμο	Στοιχεία επικοινωνίας
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Φοίβος Μυλωνάς	Πλατεία Τσιριγώτη 7, Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87741
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Βασίλειος Καρυώτης	Πλατεία Τσιριγώτη 7, Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87757
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Παναγιώτης Κουρουθανάσης	Πλατεία Τσιριγώτη 7, Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87757
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Ευαγγελία Τσιλάκου	Πλατεία Τσιριγώτη 7, Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87757
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Κλειώ Στάμου	Πλατεία Τσιριγώτη 7, Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87757
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Γεώργιος Βούτος	Πλατεία Τσιριγώτη 7, Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87757
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Ιωάννης Καθενιώτης	Πλατεία Τσιριγώτη 7, Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87757
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Ειρήνη Μαθέ	Πλατεία Τσιριγώτη 7, Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87757
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Άννα Σωτηροπούλου	Πλατεία Τσιριγώτη 7, Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87757
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Έλενα Κουμεντάκου	Πλατεία Τσιριγώτη 7,



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Ταμείο  
Περιφερειακής Ανάπτυξης

**ΕΠΑνεΚ 2014-2020**  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ  
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ  
ΚΑΙΝΟΤΟΜΙΑ

**ΕΣΠΑ**  
2014-2020  
ανάπτυξη - εργασία - αλληλεγγύη

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

		Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87756
Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο	Ελένη Στάη	Πλατεία Τσιριγώπη 7, Τ.Κ. 49132 Κέρκυρα, Ελλάδα Τηλ.: 26610 87756
Ινστιτούτο Ανάπτυξης Επιχειρηματικότητας	Γεώργιος Γκόγκας	Βαλτετσίου και Τριπόλεως, Τ.Κ. 41336 Λάρισα, Ελλάδα Τηλ.:2410 626943
Ινστιτούτο Ανάπτυξης Επιχειρηματικότητας	Κατερίνα Παρίζα	Βαλτετσίου και Τριπόλεως, Τ.Κ. 41336 Λάρισα, Ελλάδα Τηλ.:2410 626943
RMR I.K.E.	Βασιλική Σουμπενιώτη	Ηρ. Πολυτεχνείου 95, Τ.Κ. 41222 Λάρισα, Ελλάδα Τηλ.:2410 669040
RMR I.K.E.	Απόστολος Παναγόπουλος	Ηρ. Πολυτεχνείου 95, Τ.Κ. 41222 Λάρισα, Ελλάδα Τηλ.:2410 669040
RMR I.K.E.	Άγγελος Παπαγεωργίου	Ηρ. Πολυτεχνείου 95, Τ.Κ. 41222 Λάρισα, Ελλάδα Τηλ.:2410 669040
RMR I.K.E.	Κωνσταντίνος Γρηγορίου	ΤΗρ. Πολυτεχνείου 95, Τ.Κ. 41222 Λάρισα, Ελλάδα Τηλ.:2410 669040



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Ταμείο  
Περιφερειακής Ανάπτυξης

**ΕΠΑνεΚ 2014-2020**  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ  
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ  
ΚΑΙΝΟΤΟΜΙΑ

**ΕΣΠΑ**  
2014-2020  
ανάπτυξη - εργασία - αλληλεγγύη

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

## Συντομογραφίες και ακρωνύμια

Συντομογραφία	Επεξήγηση
<b>BP</b>	Back-pressure (Οπισθοπίεση)
<b>IoT</b>	Internet of Things (Διαδίκτυο των Πραγμάτων)
<b>MWM</b>	Maximum Weight Matching (Ταίριασμα Μέγιστου Βάρους)
<b>QoS</b>	Quality-of-Service (Ποιότητα Υπηρεσίας)
<b>SCM</b>	Supply Chain Management (Διαχείριση Εφοδιαστικής Αλυσίδας)
<b>SME</b>	Small-Medium Enterprise (Μικρο-μεσαία Επιχείρηση)
<b>SOTA</b>	State-of-the-Art (Τεχνολογία Αιχμής)

## Περιεχόμενα

1. Εισαγωγή .....	11
Αρχιτεκτονικές ανάπτυξης λογισμικού .....	14
2.1 Monolithic architecture .....	14
2.2 Microservices / Cloud-Native architecture .....	15
2.3 Event-driven serverless architecture.....	16
2.4 Cloud-Based architecture .....	17
2.5 Η αρχιτεκτονική του συστήματος ΒΕΛΟΣ.....	18
3. Σχεδιασμός και ανάλυση της βάσης δεδομένων .....	21
3.1 Απαιτήσεις και προδιαγραφές της βάσης δεδομένων.....	21
3.2 Παρουσίαση του διαγράμματος οντοτήτων - συσχετίσεων(ER) του συστήματος ΒΕΛΟΣ .....	24
Πίνακας 1 – 3.2.1 Οντότητα: CUSTOMER.....	25
Πίνακας 2 – 3.2.2 Οντότητα: INDIVIDUAL .....	26
Πίνακας 3 – 3.2.3 Οντότητα: COMPANY.....	27
Πίνακας 1 – 3.2.4 Οντότητα: REQUEST .....	28
Πίνακας 1 – 3.2.5 Οντότητα: ITEM .....	29
Πίνακας 1 – 3.2.6 Οντότητα: SHIPMENT .....	31
Πίνακας 1 – 3.2.7 Οντότητα: HUB .....	32
Πίνακας 1 – 3.2.8 Οντότητα: EMPLOY .....	34
Πίνακας 1 – 3.2.9 Οντότητα: DRIVER .....	35
Πίνακας 1 – 3.2.10 Οντότητα: OFFICE_EMPLOY .....	36
3.3 Παρουσίαση του διαγράμματος της ολοκληρωμένης βάσης δεδομένων.....	37
Πίνακας 1 – 3.3.1 Πίνακας: CUSTOMER .....	39
Πίνακας 1 – 3.3.2 Πίνακας: INDIVIDUAL .....	40
Πίνακας 1 – 3.3.3 Πίνακας: COMPANY .....	41
Πίνακας 1 – 3.3.4 Πίνακας: REQUEST .....	42

Πίνακας 1 – 3.3.5 Πίνακας: ITEM.....	43
Πίνακας 1 – 3.3.6 Πίνακας: IS_PART_OF.....	44
Πίνακας 1 – 3.3.7 Πίνακας: SHIPMENT.....	45
Πίνακας 1 – 3.3.8 Πίνακας: EMPLOYEE .....	46
Πίνακας 1 – 3.3.9 Πίνακας: OFFICE_EMPLOYEE.....	47
Πίνακας 1 – 3.3.10 Πίνακας: DRIVER .....	48
Πίνακας 1 – 3.3.11 Πίνακας: DOOR_TO_DOOR.....	49
Πίνακας 1 – 3.3.12 Πίνακας: IS_DELIVERED_TO .....	50
Πίνακας 1 – 3.3.13 Πίνακας: HUB .....	51
Πίνακας 1 – 3.3.14 Πίνακας: IS_CONNECTED_TO.....	52
Πίνακας 1 – 3.3.15 Πίνακας: TRACKING.....	53
4. Παρουσίαση του User Interface του έργου ΒΕΛΟΣ.....	55
4.1 Εισαγωγή στην ανάλυση του User Interface του έργου ΒΕΛΟΣ.....	55
4.2 Ανάλυση και επεξήγηση των διαφορετικών οθονών του έργου ΒΕΛΟΣ .....	58
Αναφορές .....	79



## Κατάλογος Εικόνων

Εικόνα 1 – Αρχιτεκτονική του συστήματος ΒΕΛΟΣ .....	19
Εικόνα 2 – Διάγραμμα οντοτήτων συσχετίσεων του έργου ΒΕΛΟΣ .....	24
Εικόνα 3 – Το πλήρες διάγραμμα της ενδιάμεσης βάσης δεδομένων .....	38
Εικόνα 4 – Το περιβάλλον εργασίας για το έργο ΒΕΛΟΣ .....	57
Εικόνα 5 –: Αλληλεπιδράσεις και μονοπάτια πλοήγησης στις οθόνες του έργου ΒΕΛΟΣ .....	57
Εικόνα 6 – Οθόνη <i>log in</i> του συστήματος .....	58
Εικόνα 7 – Το κύριο μενού του συστήματος .....	59
Εικόνα 8 – Οι λεπτομέρειες του χρήστη .....	60
Εικόνα 9 – Οι κόμβοι του οδικού δικτύου .....	61
Εικόνα 10 – Η εμφάνιση λεπτομερειών του επιλεγμένου κόμβου .....	62
Εικόνα 11 – Η οθόνη αναζήτησης παραγγελίας. ....	63
Εικόνα 12 – Οθόνη θέασης και επεξεργασίας δεδομένων .....	64
Εικόνα 13 – Ο παραγόμενος πίνακας <i>IS_DELIVERED_TO</i> .....	65
Εικόνα 14 – Οθόνη στατιστικών .....	66
Εικόνα 15 – Εισαγωγή <i>HUB</i> στην βάση δεδομένων .....	67
Εικόνα 16 – Προσθήκη επιπλέον συνδέσεων στον κόμβο .....	68
Εικόνα 17 – Προσθήκη πελάτη .....	69
Εικόνα 18 – Αποτέλεσμα ήδη υπάρχον πελάτη .....	70
Εικόνα 19 – Εισαγωγή παραγγελίας .....	71
Εικόνα 20 – Μήνυμα επιτυχίας με την εισαγωγή των δεδομένων .....	72
Εικόνα 21 – Οθόνη εισαγωγής <i>.csv</i> αρχείου .....	73
Εικόνα 22 – Οθόνη εκκίνησης αλγορίθμου .....	74
Εικόνα 23 – Επιλογή διαθέσιμων μεταφορών .....	75
Εικόνα 24 – Οθόνη εκκίνησης αλγορίθμου με τις επιλεγμένες μεταφορές .....	76
Εικόνα 25 – Μήνυμα επιτυχίας από τον αλγόριθμο .....	77
Εικόνα 26 – Οθόνη με τις προηγούμενες επαναλήψεις .....	78

## Κατάλογος Πινάκων

Πίνακας 1 – 3.2.1 Οντότητα: CUSTOMER .....	25
Πίνακας 2 – 3.2.2 Οντότητα: INDIVIDUAL .....	26
Πίνακας 3 – 3.2.3 Οντότητα: COMPANY .....	27
Πίνακας 4 – 3.2.4 Οντότητα: REQUEST .....	28
Πίνακας 5 – 3.2.5 Οντότητα: ITEM.....	29
Πίνακας 6 – 3.2.6 Οντότητα: SHIPMENT .....	31
Πίνακας 7 – 3.2.7 Οντότητα: HUB.....	32
Πίνακας 8 – 3.2.8 Οντότητα: EMPLOY .....	34
Πίνακας 9 – 3.2.9 Οντότητα: DRIVER.....	35
Πίνακας 10 – 3.2.10 Οντότητα: OFFICE_EMPLOY .....	36
Πίνακας 11 – 3.3.1 Πίνακας: CUSTOMER .....	39
Πίνακας 12 – 3.3.2 Πίνακας: INDIVIDUAL.....	40
Πίνακας 13 – 3.3.3 Πίνακας: COMPANY .....	41
Πίνακας 14 – 3.3.4 Πίνακας: REQUEST.....	42
Πίνακας 15 – 3.3.5 Πίνακας: ITEM .....	43
Πίνακας 16 – 3.3.6 Πίνακας: IS_PART_OF .....	44
Πίνακας 17 – 3.3.7 Πίνακας: SHIPMENT .....	45
Πίνακας 18 – 3.3.8 Πίνακας: EMPLOYEE .....	46
Πίνακας 19 – 3.3.9 Πίνακας: OFFICE_EMPLOYEE .....	47
Πίνακας 20 – 3.3.10 Πίνακας: DRIVER .....	48
Πίνακας 21 – 3.3.11 Πίνακας: DOOR_TO_DOOR.....	49
Πίνακας 22 – 3.3.12 Πίνακας: IS_DELIVERED_TO.....	50
Πίνακας 23 – 3.3.13 Πίνακας: HUB .....	51
Πίνακας 24 – 3.3.14 Πίνακας: IS_CONNECTED_TO .....	52
Πίνακας 25 – 3.3.15 Πίνακας: TRACKING .....	53

## 1. Εισαγωγή

Το έργο ΒΕΛΟΣ εστιάζει στις δραστηριότητες των μικρο-μεσαίων μεταφορικών εταιρειών, και ειδικότερα στις ανάγκες της RMR, η οποία είναι μια μικρο-μεσαία επιχείρηση (Small-Medium Enterprise, SME) που δραστηριοποιείται στις μεταφορές αγαθών εξειδικευμένης διαχείρισης. Μεταξύ άλλων, ένα βασικό πρόβλημα της RMR, και αντίστοιχων μικρο-μεσαίων εταιρειών που δραστηριοποιούνται στον κλάδο των μεταφορών, είναι αυτό της βέλτιστης ανάθεσης πόρων (φορτωτικών, οχημάτων, χρονοδιαγραμμάτων, κλπ.) και χρονοπρογραμματισμού των αναγκαίων μεταφορών και/ή συνοδευτικών ενεργειών (π.χ. πακετάρισμα-packaging) σε ημερήσια/εβδομαδιαία/μηνιαία/ετήσια βάση μεταξύ των σημείων ενδιαφέροντος, ώστε να εξασφαλιστεί ότι το ανεκτέλεστο κάθε μέρας (εκκρεμείς παραδόσεις σε αναμονή-αποθήκη) και ο μέσος χρόνος παράδοσης παραμένουν οριοθετημένα (πιθανά και ελεγχόμενα), ενώ ελαχιστοποιούνται άλλα κριτήρια βέλτιστης λειτουργίας, π.χ. κόστος μεταφοράς, αποθήκευσης, συντήρησης στόλου, χρόνος παράδοσης, κλπ.

Τα παραπάνω ζητήματα είναι πολύ σημαντικά για την RMR, όπως και για κάθε αντίστοιχη μικρο-μεσαία επιχείρηση του κλάδου, καθώς μπορεί να αποβούν κρίσιμα για τη λειτουργία της και τις απαραίτητες οικονομίες κλίμακας που πρέπει να γίνουν σε μια παγκοσμιοποιημένη πλέον αγορά. Αυτή η ανάγκη, η οποία σε άλλες χώρες, π.χ. ΗΠΑ, Βρετανία, κλπ., έχει προκύψει εδώ και δεκαετίες, έχει πλέον παγιωθεί και στην Ελλάδα, τόσο λόγω της ανόδου τα τελευταία χρόνια του ηλεκτρονικού λιανεμπορίου, όσο και της αύξησης της ζήτησης που προέκυψε από την πανδημία COVID-19.

Η RMR θα κληθεί να καλύψει αυξημένη ζήτηση αιτούμενων μεταφορών με μεγαλύτερη ποικιλία σε σχέση με το παρελθόν, εν δυνάμει από/σε εγχώριες και

διεθνείς τοποθεσίες. Η ανάγκη αναμένεται να αυξηθεί περαιτέρω καθώς η χώρα μας έχει αποφασίσει στρατηγικά να γίνει διακομιστικό κέντρο προς την υπόλοιπη Ευρώπη. 15

Το πρόβλημα που ανακύπτει είναι αυτό ενός δικτύου διανομής με πηγές και προορισμούς, μέσα από το οποίο ζητείται να μεταφερθούν εμπορεύματα διαφορετικού τύπου. Σε κάθε ενδιάμεσο κόμβο υπάρχουν ουρές αγαθών, και ο στόχος είναι να ελαχιστοποιηθεί ο συνολικός μέσος χρόνος παράδοσης κάθε αγαθού, ενώ το δίκτυο διατηρείται σε σταθερή κατάσταση, δηλαδή ότι αθροιστικά όλες οι ουρές αναμονής σε όλους τους σταθμούς διατηρούνται πεπερασμένες. Επιπρόσθετοι στόχοι, όπως π.χ. μειωμένη κατανάλωσης ενέργειας, λειτουργικού κόστους, κλπ., μπορούν να τεθούν κατ' απαίτηση.

Για την αντιμετώπιση αντίστοιχων προβλημάτων έχουν προταθεί αρκετές προσεγγίσεις. Μερικές από τις πιο συχνά χρησιμοποιούμενες είναι ο δυναμικός χρονοπρογραμματισμός, βέλτιστος στοχαστικός έλεγχος, ο κυρτός και μικτός-διακριτός προγραμματισμός, και η μηχανική μάθηση. Όλες οι παραπάνω προσεγγίσεις εμφανίζουν δύο κοινές προκλήσεις: α) είναι υπολογιστικά απαιτητικές, και β) βασίζονται κατά βάση σε κεντροποιημένους αλγόριθμους.

Το έργο ΒΕΛΟΣ φιλοδοξεί να αντιμετωπίσει το παραπάνω πρόβλημα στο πλαίσιο της εφαρμογής εταιρειών μεταφορών με μια πρωτοποριακή προσέγγιση, βασισμένη στον αλγόριθμο δρομολόγησης-χρονοπρογραμματισμού οπισθοπίεσης (backpressure). Ο αλγόριθμος έχει χρησιμοποιηθεί εκτενώς σε δίκτυα επικοινωνιών και οδικά δίκτυα για τη μείωση της συμφόρησης κίνησης δεδομένων ή αυτοκινήτων. Στο ΒΕΛΟΣ, ο αλγόριθμος backpressure θα χρησιμοποιηθεί για την

επίλυση του προβλήματος κατανομής πόρων και διατήρησης σταθερής κατάστασης, καθώς και επιπρόσθετων στόχων βελτιστοποίησης κριτηρίων λειτουργίας/επίδοσης για μικρομεσαίες εταιρείες μεταφορών

Το παρόν παραδοτέο Π2 εστιάζει στην ανάλυση και τον σχεδιασμό του συστήματος που ως στόχο έχει να βοηθήσει την λειτουργία της RMR. Αυτό θα επιτευχθεί δημιουργώντας ένα πληροφοριακό σύστημα φιλικό προς τον χρήστη και κατανοητό στην λειτουργία του ενώ ταυτόχρονα, θα είναι σύγχρονο και φτιαγμένο στα μέτρα και τις ανάγκες της RMR. Αυτό το σύστημα το οποίο θα αναλυθεί εκτενώς παρακάτω θα αποτελείται από τρεις βασικές έννοιες. Πρώτον την αρχιτεκτονική του συστήματος, για την οποία παρακάτω θα αναλυθούν όλες οι υπάρχουσες αρχιτεκτονικές στην “αγορά” σήμερα, θα συγκριθούν μεταξύ τους και τέλος θα αναλυθεί η τελική αρχιτεκτονική για το έργο ΒΕΛΟΣ και ο λόγος που επιλέχθηκε. Δεύτερον τη βάση δεδομένων και την στρατηγική που ακολουθήθηκε για την σχεδίαση και την ανάλυση της. Θα αναφερθούν οι απαιτήσεις που ζητήθηκαν να καλυφθούν και τέλος θα παρουσιαστούν σχηματικά τα διαγράμματα που παρήχθησαν και η βάση με τον τρόπο που θα υπάρχει πραγματικά. Τρίτον το User Interface(UI) και η μακέτα που σχεδιάστηκε για το πως θα φαίνεται το σύστημα στον χρήστη. Θα επεξηγηθούν τι είναι αυτά που βλέπει ο χρήστης και η σκοπιμότητα τους ,η στρατηγική που ακολουθήθηκε και το User Experience (UX) στο οποίο αποσκοπούμε να έχει ο χρήστης.

Εν κατακλείδι, το όραμα μας για το έργο ΒΕΛΟΣ είναι να δώσουμε λύση σε προβλήματα που αντιμετωπίζουν μικρο-μεσαίες εταιρείες, όπως η RMR αλλά και μεγαλύτερες της, δημιουργώντας ένα πλήρες, λειτουργικό, εύκολο στην χρήση και καινοτόμο σύστημα που παρόλο την μικρή του “ηλικία” μπορεί να ανταγωνιστεί πολύ παλαιότερα συστήματα του είδους του.

## Αρχιτεκτονικές ανάπτυξης λογισμικού

Σε αυτό το κεφάλαιο θα αναλύσουμε τις διαφορετικές αρχιτεκτονικές που υπάρχουν για την ανάπτυξη, την επέκταση και την συντήρηση πληροφοριακών συστημάτων. Συγκεκριμένα, θα αναφέρουμε και θα συγκρίνουμε την χρησιμότητα, τα πλεονεκτήματα και μειονεκτήματα των αρχιτεκτονικών αυτών, οι οποίες είναι: η Monolithic Architecture, η Microservices/Cloud-native architecture, η Event-Driven serverless architecture και η Cloud-based architecture. Τέλος, θα αναφέρουμε την αρχιτεκτονική που επιλέχθηκε για την ανάπτυξη του έργου ΒΕΛΟΣ καθώς και τους λόγους επιλογής της, επιπλέον θα δείξουμε σχηματικά τα επιμέρους κομμάτια του συστήματος.

### 2.1 Monolithic architecture

Η Monolithic architecture(MA) είναι ο παραδοσιακός τρόπος με τον οποίο οι εταιρείες στο παρελθόν δημιουργούσαν πληροφοριακά συστήματα. [1] Στην MA όλη η λειτουργικότητα του συστήματος βρίσκεται σε μία μεγάλη εφαρμογή, εξωτερικά μπορεί να φαίνεται ότι διαφορετικά κομμάτια του συστήματος βρίσκονται και εκτελούνται ξεχωριστά αλλά στην MA αυτό δεν υφίσταται. Όλο το σύστημα, από όσα κομμάτια και αν αποτελείται, είναι ένα μονάχα αρχείο, για παράδειγμα ένα αρχείο .exe. Αυτό σημαίνει ότι οι λειτουργικότητες του συστήματος εκφράζονται σε συναρτήσεις και κλάσεις.

Η MA παρόλο που είναι η παλαιότερη από τις αρχιτεκτονικές που αναφέρθηκαν έχει αρκετά πλεονεκτήματα σε σχέση με αυτές. Αρχικά, η MA επειδή είναι ένα μεγάλο αυτόνομο σύστημα επιτρέπει την καλύτερη κατανόηση και την ταχύτερη ανάπτυξη, ενώ παράλληλα διευκολύνει πολύ την διαδικασία εύρεσης σφαλμάτων(ή αλλιώς debugging).[2] Επίσης, η MA είναι ιδανική για μικρές ομάδες ανάπτυξης διότι η πολυπλοκότητα του συστήματος είναι αισθητά μικρότερη από τις άλλες αρχιτεκτονικές. Ακόμα, εάν το σύστημα έχει αυξημένο φόρτο μία δεδομένη χρονική στιγμή, μπορεί κανείς απλά να τρέξει μία δεύτερη φορά ένα στιγμιότυπο του συστήματος μοιράζοντας τον φόρτο στα δύο[1].

Ωστόσο, όσο μεγαλύτερο γίνεται το σύστημα τόσο περισσότερα προβλήματα παρουσιάζονται με αυτήν την αρχιτεκτονική. Αρχικά, με το μέγεθος του συστήματος να αυξάνεται η πολυπλοκότητα του έχει την ίδια τροπή, οι συνδέσεις μεταξύ των τμημάτων

γίνονται περισσότερες και η δυνατότητα κατανόησης της λειτουργίας του συστήματος όλο και δυσκολότερη.[1] Επίσης, οι αλλαγές που γίνονται στο σύστημα, μικρές ή μεγάλες, χρειάζονται λεπτομερή ανάλυση και σχεδιασμό, διότι για κάθε αλλαγή είναι απαραίτητο να ξαναφτιαχτεί το εκτελέσιμο αρχείο του συστήματος[2]. Σε κάθε περίπτωση, η MA χρειάζεται ίσως τον μεγαλύτερο χρονικά σχεδιασμό από οποιοδήποτε άλλη αρχιτεκτονική επειδή όλο το σύστημα μπορεί να χρησιμοποιεί μονάχα μία γλώσσα προγραμματισμού. Επακόλουθο είναι, να μην μπορεί να αξιοποιηθεί κάποια νέα γλώσσα που ίσως αναπτυχθεί με τα χρόνια και να υπάρχουν συνεχώς προβλήματα με τους περιορισμούς της επιλεγμένης[1][2].

## 2.2 Microservices / Cloud-Native architecture

Η Microservices architecture (MSA) είναι η εξέλιξη της παράδοσης που έθεσε η Monolithic architecture[3]. Σε αντίθεση με την δεύτερη, η MSA χωρίζει τα λειτουργικά κομμάτια ενός συστήματος σε μικρότερα αλληλεξαρτώμενα υποσυστήματα[1]. Επομένως, πλέον δεν υπάρχει ένα εκτελέσιμο αρχείο αλλά πολλά μικρά αρχεία που εκτελούνται και υπάρχουν στο υπολογιστικό νέφος(cloud). Ωστόσο η επικοινωνία μεταξύ τους είναι απαραίτητη καθώς πολλές φορές ένα σύστημα στηρίζεται στα δεδομένα που θα παραχθούν από ένα άλλο[4]. Για παράδειγμα, εάν είχαμε ένα σύστημα για ένα e-shop θα χρειαζόμασταν μία λειτουργία καλαθιού και μία λειτουργία πληρωμών όπου προφανώς τα δύο θα έπρεπε να “επικοινωνούν” μεταξύ τους για να υπάρχει σωστή λειτουργία του συστήματος.

Η MSA έχει κάποια σημαντικά πλεονεκτήματα όπως το γεγονός ότι υπάρχει διαμοιρασμός συστημάτων σε μικρότερα[2]. Αυτό δίνει την δυνατότητα για ανθεκτικότητα στα σφάλματα, διότι εάν κάποια από τα υποσυστήματα σταματήσει να λειτουργεί μπορεί κανείς απλά να αντικαταστήσει το υποσύστημα με ένα καινούργιο στιγμιότυπο χωρίς να επηρεάσει καθόλου το σύνολο του συστήματος. Έπειτα, υπάρχει η δυνατότητα για κάθε υποσύστημα και κάθε ομάδα που το αναπτύσσει και συντηρεί να επιλέξει μία ξεχωριστή γλώσσα προγραμματισμού στα μέτρα και τις απαιτήσεις του υποσυστήματος[1]. Ως συνέπεια, ο κώδικας των υποσυστημάτων να είναι αναγνωρίσιμος και κατανοητός, κάτι που βοηθάει στην συντήρηση του[1][2][3][4]. Επίσης σε ώρες αιχμής που μπορεί κάποιο υποσύστημα να δέχεται μεγάλο φόρτο κανείς μπορεί να εκτελέσει δεύτερη φορά το ίδιο υποσύστημα μοιράζοντας τον φόρτο και εξοικονομώντας υπολογιστικούς πόρους στο νέφος σε αντίθεση με την προσέγγιση της MA[1][3].

Από την άλλη πλευρά, η MSA είναι απαγορευτικά αργή και κοστοβόρα για μικρές εταιρείες με λίγο εργατικό δυναμικό. Η επικοινωνία μεταξύ των υπηρεσιών είναι δύσκολη και συχνά χρειάζεται εξειδικευμένο προσωπικό για να διευθύνει την λειτουργία των υποσυστημάτων[1].

Επίσης, τις περισσότερες φορές τα υποσυστήματα πρέπει να λειτουργούν σε διαφορετικά μηχανήματα με αποτέλεσμα να υπάρχουν μεγαλύτερα κόστη συντήρησης στην χρήση της υποδομής για την επικοινωνία μεταξύ τους[4]. Τέλος, η MSA χρειάζεται περισσότερη σχεδίαση όσον αφορά την ασφάλεια, την αποδοτικότητα του συνολικού συστήματος καθώς και τους αυτόματους μηχανισμούς που είναι απαραίτητοι στα υποσυστήματα για την συνεχόμενη λειτουργία και συντήρηση τους[4][3]

### 2.3 Event-driven serverless architecture

Η Event-driven serverless architecture (Serverless computing architecture ή SCA[5]) είναι ίσως η πιο νέα αρχιτεκτονική από τις τέσσερις. Αρχικά, ο όρος Serverless δεν είναι κυριολεκτικός, στην πραγματικότητα υπάρχει κάποιος σέρβερ αλλά αυτός δεν λειτουργεί όπως ένας παραδοσιακός, όπου είναι συνέχεια ενεργός και στέλνει ή λαμβάνει δεδομένα από και προς κάποιο χρήστη. Ο σέρβερ της SCA είναι ανενεργός έως ότου κάποιο συμβάν(event) εκτελεστεί από τις κινήσεις του χρήστη την δεδομένη χρονική στιγμή[6]. Όταν το συμβάν αυτό εμφανιστεί εκτελείται μία συνάρτηση στον σέρβερ όπως η AWS Lambda[5][6].

Ένα μεγάλο πλεονέκτημα της SCA είναι ότι τα λειτουργικά κόστη των σέρβερ είναι πολύ χαμηλά διότι χρειάζεται υπολογιστική ισχύ μόνο όταν ένα event συμβαίνει[6]. Η SCA χρειάζεται λίγη έως καθόλου προετοιμασία εκτέλεσης και προγραμματισμό για την χρήση της, ενώ οι σέρβερ στους οποίους υπάρχουν οι συναρτήσεις ανήκουν σε κάποιο εξωτερικό συνεργάτη[5]. Αυτό προσφέρει μεγαλύτερη ασφάλεια και λιγότερη συντήρηση κατά την πάροδο του χρόνου[5][6]. Ακόμα, η επεκτασιμότητα του συστήματος είναι σχεδόν αυτόματη με μεγάλο αριθμό εταιρειών να επιλέγουν την SCA για την ανθεκτικότητα της στην αντιμετώπιση μεγάλου φόρτου αιτημάτων[6][7].

Ωστόσο, η SCA έχει αρκετά μειονεκτήματα. Αρχικά, τα τεστ και η εξακρίβωση σφαλμάτων(debugging) που πρέπει να γίνουν κατά την διάρκεια της ανάπτυξης απαιτούν



πολύ χρόνο και εγείρουν δυσκολίες γιατί χρειάζεται να προσομοιωθεί το περιβάλλον του σέρβερ εντός της εταιρείας[6]. Ένα ακόμα αρνητικό της SCA είναι η φύση των συναρτήσεων, δηλαδή εάν μία συνάρτηση χρειάζεται να είναι ενεργή για μεγάλο χρονικό διάστημα ο σέρβερ θα την τερματίσει μόνος του γιατί το επιτρεπόμενο χρονικό όριο έχει περάσει[8]. Ακόμα, οι συναρτήσεις πρέπει να είναι εναρμονισμένες με τον πάροχο που κατέχει τους σέρβερ για να δουλεύουν σωστά, με αποτέλεσμα στην περίπτωση αλλαγής παρόχου οι συναρτήσεις να χρειάζονται μετατροπή για να ταιριάζουν στο νέο API με το οποίο θα συνδεθούν[6][7]. Το μεγαλύτερο μειονέκτημα όμως της SCA είναι η λογική της Event-driven αρχιτεκτονικής[6]. Σε ένα μικρό σύστημα τα “βήματα” που πρέπει να ακολουθήσει ένας χρήστης για να προβεί στο κατάλληλο συμβάν μπορεί να είναι λίγα και προβλέψιμα. Αντίθετα, σε ένα μεγάλο σύστημα που ο χρήστης μπορεί να κάνει πολλά και απρόβλεπτα βήματα, για τα οποία οι προγραμματιστές μπορεί να μην έχουν προνοήσει, έχει ως αποτέλεσμα την συνεχόμενη εκτέλεση πολλών αλγορίθμων για την επίλυση του προβλήματος, ανεβάζοντας δραματικά το κόστος λειτουργίας.[6][8]

## 2.4 Cloud-Based architecture

Η Cloud-Based architecture χωρίζεται σε τρεις βασικές υπο-ενότητες. Αυτές είναι: Infrastructure as a service(IaaS), Platform as a service(PaaS) και Software as a service(SaaS)[9]. Το IaaS είναι οι σέρβερ, ο αποθηκευτικός χώρος και γενικά η υπαρκτή υποδομή. Το PaaS χτίζει πάνω στην έννοια του IaaS και παρέχει ένα χώρο για δοκιμές με χαμηλό κόστος, στο οποίο ο χρήστης δεν χρειάζεται να ρυθμίσει το περιβάλλον εργασίας παρά μόνο να εγκαταστήσει τα απαραίτητα εργαλεία.[10] Το SaaS δίνει την δυνατότητα σε χρήστες να χρησιμοποιούν λογισμικό που δεν τους ανήκει και ούτε έχουν εγκατεστημένο στον υπολογιστή τους. Ενώ, για την χρήση του χρειάζεται μονάχα ένας φυλλομετρητής και μία σύνδεση στο διαδίκτυο[10].

Τα πλεονεκτήματα μιας τέτοιας αρχιτεκτονικής είναι εμφανή. Ο χρήστης χρειάζεται ελάχιστο εξοπλισμό για να χρησιμοποιήσει τα παραπάνω, ενώ η εταιρεία που το αναπτύσσει δεν έχει πολλά λειτουργικά κόστη για την συντήρηση των σερβερ γιατί η πλειοψηφία χρησιμοποιεί παρόχους για αυτόν τον σκοπό[7][10]. Τέλος, ένας πάροχος διατηρεί σχεδόν πάντα υψηλά στάνταρ ασφαλείας σε δεδομένα, με αποτέλεσμα ο κίνδυνος για την κλοπή ή διαγραφή δεδομένων να είναι μηδαμινός[7].

Ωστόσο, αυτή η προσέγγιση έχει και κάποια μειονεκτήματα, με το πιο προφανές να είναι η σύνδεση στο διαδίκτυο. Σε περίπτωση που χαθεί η σύνδεση στο διαδίκτυο ή δεν είναι αρκετά γρήγορη, θα ήταν αδύνατο για έναν χρήστη να χρησιμοποιήσει το σύστημα[9]. Όπως και με την προηγούμενη αρχιτεκτονική υπάρχει κίνδυνος στην αλλαγή παρόχου, διότι τα στάνταρ λειτουργίας και APIs αλλάζουν από πάροχο σε πάροχο, με αποτέλεσμα να χρειάζεται ολική αλλαγή της βάσης δεδομένων ή διαφόρων συναρτήσεων που μπορεί να εκτελεί ο σέρβερ[10].

## 2.5 Η αρχιτεκτονική του συστήματος ΒΕΛΟΣ

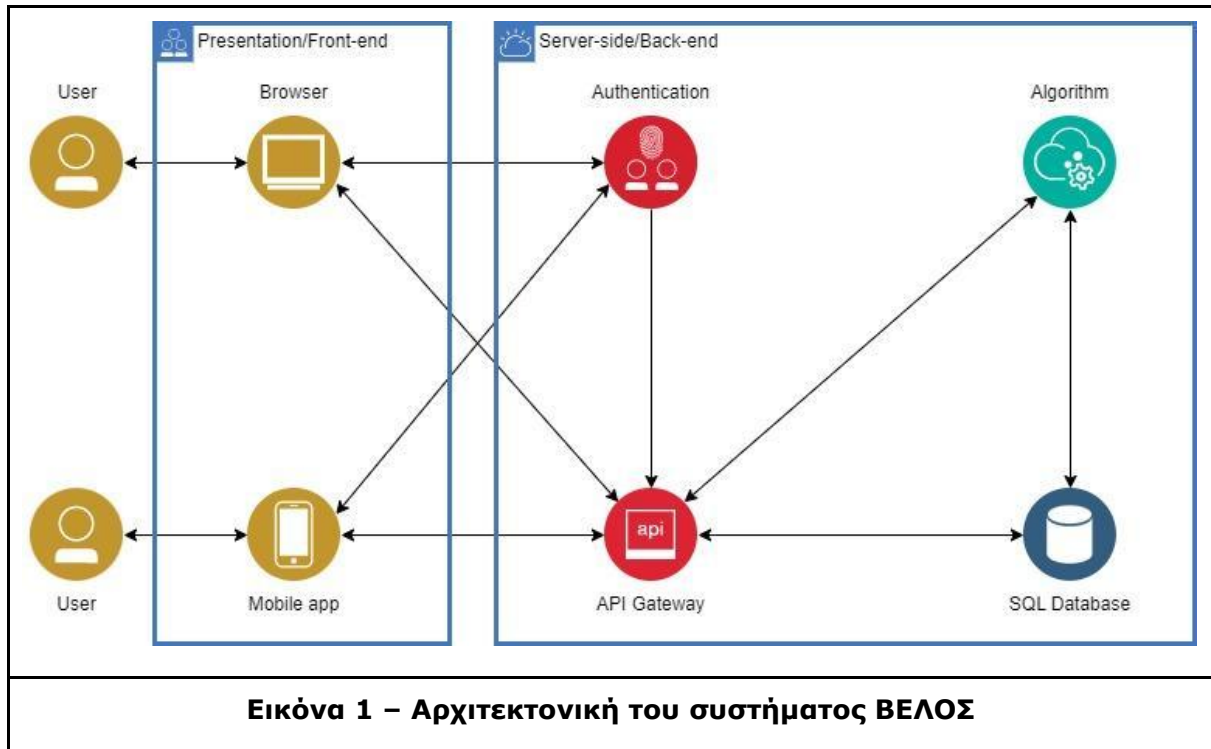
Η αρχιτεκτονική του συστήματος βασίστηκε πάνω στην φιλοσοφία της διαμόρφωσης αντίστοιχων συστημάτων που επιτελούν τον ίδιο σκοπό με το ΒΕΛΟΣ. Το έργο ΒΕΛΟΣ λοιπόν, πρέπει να είναι ένα εταιρικό σύστημα στο οποίο ένας εξουσιοδοτημένος υπάλληλος να μπορεί να εισέλθει και να εισάγει δεδομένα. Έπειτα ο ίδιος να μπορεί να εκτελεί τον αλγόριθμο και να εξάγει κάποια συμπεράσματα με τα ευρήματα του. Τέλος, το ΒΕΛΟΣ θα πρέπει να δημιουργεί στατιστικά για την λειτουργία και την αποδοτικότητα της εφοδιαστικής αλυσίδας που “επιβλέπει”.

Έχοντας τα παραπάνω κατά νου συμπεραίνουμε τα εξής για το σύστημα:

- Πρέπει να είναι φιλικό προς τον χρήστη για να μην χρειάζεται κοστοβόρα και χρονοβόρα εκπαίδευση προσωπικού πάνω σε αυτό.
- Χρειάζεται κάποιο τρόπο αυθεντικοποίησης του χρήστη που επιθυμεί να εισέλθει.
- Πρέπει να υποστηρίζει εισαγωγή, επεξεργασία, δημιουργία και εξαγωγή δεδομένων.
- Πρέπει να εκτελεί τον αλγόριθμο που θα βελτιστοποιεί την εφοδιαστική αλυσίδα.

Η αρχιτεκτονική του συστήματος αποφασίστηκε έχοντας υπ όψιν την βιβλιογραφία και τα παραπάνω. Το σύστημα ΒΕΛΟΣ θα έχει μία υβριδική αρχιτεκτονική. Δηλαδή, θα είναι ένα Monolithic architecture με στοιχεία Cloud-based architecture και συγκεκριμένα των software-as-a-service κομματιών του. Συγκεκριμένα, το γραφικό περιβάλλον και το κομμάτι του συστήματος που θα διαχειρίζεται ο χρήστης θα είναι monolithic architecture. Δηλαδή το υποσύστημα εισαγωγής/εξαγωγής δεδομένων, η αποθήκευση τους, η αυθεντικοποίηση και η εντολή λειτουργίας του αλγορίθμου. Από την άλλη, ο αλγόριθμος θα είναι software-as-a-

service Cloud-based architecture, καθώς ο αλγόριθμος θα βρίσκεται και θα εκτελείται στον σέρβερ του συστήματος και θα συνδέεται μονάχα με την βάση δεδομένων. Επομένως, ο χρήστης θα μπορεί να δει μονάχα τα αποτελέσματα αυτού και όχι τον πηγαίο κώδικα του ή την επεξεργασία δεδομένων που χρειάζεται για την εκτέλεση του. Ακολουθεί το σχήμα λειτουργίας του συστήματος ΒΕΛΟΣ και η επεξήγηση των επιμέρων στοιχείων του



Ξεκινώντας με το presentation κομμάτι του συστήματος, θα δημιουργηθούν δύο εφαρμογές για την πρόσβαση στο σύστημα ΒΕΛΟΣ. Αφενός ένα Web-app που έχει ως σκοπό την πλήρη χρήση του συστήματος και των δυνατοτητων του. Μέσω αυτού θα γίνεται η εισαγωγή, επεξεργασία, δημιουργία και εξαγωγή δεδομένων, ενώ ταυτόχρονα από το ίδιο θα γίνεται η διαχείριση του αλγορίθμου. Αφετέρου ένα Mobile-app το οποίο προορίζεται για τους οδηγούς των φορητών που θα κάνουν τα πραγματικά δρομολόγια. Σε αυτό ο οδηγός θα μπορεί να εισέλθει με τα πιστήρια του και θα του εμφανίζονται οι λεπτομέρειες του δρομολογίου του.

Συνεχίζοντας με το Server-side κομμάτι του συστήματος, θα δημιουργούν 4 διαφορετικά τμήματα για την εξυπηρέτηση των αναγκών του presentation κομματιού. Αρχικά, μία βάση δεδομένων η οποία θα αποθηκεύει όλα τα εξερχόμενα και εισερχόμενα δεδομένα (

εξερχόμενα είναι τα αποτελέσματα του αλγορίθμου). Μετά, ο αλγόριθμος που θα υπάρχει ξεχωριστά της βάσης και θα εκτελείται αν και όταν ζητηθεί. Έπειτα, το σύστημα θα έχει ένα API για να καθιστά εφικτή τόσο την επικοινωνία μεταξύ των presentation και server-side κομματιών όσο και την επικοινωνία μεταξύ των τμημάτων εντός του server-side κομματιού. Ο αλγόριθμος και η βάση θα πρέπει να μπορούν να επικοινωνήσουν τόσο για την ορθή λειτουργία του αλγορίθμου όσο και για την αποθήκευση των αποτελεσμάτων του. Τέλος το σύστημα θα έχει ένα ξεχωριστό τμήμα για την ταυτοποίηση των πιστηρίων των χρηστών και την διαχείριση των cookies.

### **3. Σχεδιασμός και ανάλυση της βάσης δεδομένων**

Ο σχεδιασμός και η ανάλυση της βάσης δεδομένων για το σύστημα ΒΕΛΟΣ ήταν πιθανώς το δυσκολότερο κομμάτι του έργου. Προσπαθήσαμε να δημιουργήσουμε μία βάση που θα εκπληρώνει όλες τις απαιτήσεις που μας δόθηκαν από την RMR ενώ ταυτόχρονα αποφασίστηκε να δοθεί έμφαση στην, όσο το δυνατόν περισσότερο, απλούστευση των μελλοντικών ερωτημάτων που θα δέχεται η βάση για την επιστροφή δεδομένων. Έτσι επιτυγχάνουμε δύο βήματα που θα χρησιμεύσουν την ανάπτυξη. Αφενός η αποδοτικότητα του συστήματος θα είναι πολύ μεγαλύτερη καθώς δεν θα επηρεάζεται από τον όγκο δεδομένων εντός της βάσεις. Αφετέρου οι τυχόν υπολογισμοί για τμήματα όπως τα στατιστικά θα γίνονται στο μηχανήμα του χρήστη και όχι στον σέρβερ. Κάνοντας τον σέρβερ ανθεκτικότερο σε μεγάλο πλήθος ταυτόχρονων χρηστών.

Ακόμα φάνηκε μέσω της ανάλυσης μας για το σύστημα ότι η βάση θα πρέπει να καλύψει τις ανάγκες του αλγορίθμου και τις ανάγκες του mobile-app. Ο αλγόριθμος για να λειτουργήσει χρειάζεται το οδικό δίκτυο της εφοδιαστικής αλυσίδας. Για αυτόν τον λόγο κάθε κέντρο διανομής στην πορεία του οδικού δικτύου μέσα στην βάση αναπαρίσταται σαν ένας κόμβος ενώ η απόσταση ενός κέντρου διανομής αναπαρίσταται σαν μία ακμή και το βάρος της σαν κόστος. Επομένως για να δημιουργηθεί ο αναγκαίος, για τον αλγόριθμο, πίνακας γειννίας έπρεπε φαίνονται στην βάση δεδομένων όλες οι πιθανές διαδρομές που ένας κόμβος θα είχε. Τέλος το mobile-app όπως προαναφέρθηκε θα χρησιμοποιείται από οδηγούς με οδηγίες για το δρομολόγιο που έχουν να εκτελέσουν. Ως εκ τούτου, εκτός από τα πιστήρια του υπαλλήλου, οι οδηγοί χρειάζεται να είναι μία ξεχωριστή οντότητα και να υπάρχει ξεκάθαρη σύνδεση μεταξύ αυτών και των δρομολογίων που θα εκτελούν.

#### **3.1 Απαιτήσεις και προδιαγραφές της βάσης δεδομένων**

Οι απαιτήσεις για την βάση δεδομένων αποκτήθηκαν και αποσαφηνίστηκαν μετά από μία σειρά ερωτήσεων προς τους αρμόδιους υπαλλήλους της RMR. Από την άλλη οι προδιαγραφές της βάσης δεδομένων εξακριβώθηκαν από την ανάλυση που διεξήχθη για τις ανάγκες των επιμέρων τμημάτων του έργου ΒΕΛΟΣ και πρότυπα εταιρειών που δραστηριοποιούνται στο χώρο.

Καταρχάς οι απαιτήσεις της βάσης δεδομένων που δόθηκαν και συμφώνησαν εκατέρωθεν ήταν για τα αναγκαία δεδομένα που θα έπρεπε η βάση να αποθηκεύει για τους πελάτες της εταιρείας και τις παραγγελίες που θα έκαναν. Για παράδειγμα: ο διαχωρισμός μεταξύ ιδιώτη και εταιρείας, προσωπικά στοιχεία για τα προηγούμενα όπως ΑΦΜ για τις εταιρείες και αριθμό ταυτότητας για τους ιδιώτες, στοιχεία σχετικά με την παράδοση των παραγγελιών π.χ διεύθυνση και ταχυδρομικός κώδικας. Ακόμα, στοιχεία για την παραγγελία και το ή τα δέματα που θα περιήχε για παράδειγμα: Αν η συσκευασία είναι μία παλέτα ή ένα δέμα, για τις διαστάσεις και το βάρος του, για τυχόν ειδικές οδηγίες που αφορούν την παραγγελία όπως για το αν τα δέματα της παραγγελίας είναι εύθραυστα, ευπαθής ή επικίνδυνα.

Με τα παραπάνω υπόψην χρειάστηκε να διευρύνουμε την ανάλυση μας για την βάση δεδομένων του έργου ΒΕΛΟΣ έτσι ώστε να καλύπτει τις απαιτήσεις αυτές ενώ ταυτόχρονα να συνάδει με τις ανάγκες των υπόλοιπων τμημάτων του. Συγκεκριμένα τα τμήματα αυτά είναι: Η τροφοδοσία του αλγορίθμου με δεδομένα και η αποθήκευση των ευρημάτων του, τα στατιστικά δεδομένα που πρέπει να εμφανίζει και το σύστημα ταυτοποίησης των πιστηρίων των εργαζομένων

Ξεκινώντας με την τροφοδοσία του αλγορίθμου, ο κρίσιμος ρόλος του αλγορίθμου για το έργο ΒΕΛΟΣ μας έκανε να ξεκινήσουμε την ανάλυση μας με αυτόν. Η βασική προϋπόθεση της λειτουργίας του αλγορίθμου είναι ένας πίνακας γειτνίασης με το γράφο που θα αναπαριστά το οδικό δίκτυο της εφοδιαστικής αλυσίδας. Επομένως η βάση δεδομένων χρειάζεται να αποθηκεύει τους κόμβους αλλά και τις συνδέσεις τους με άλλους κόμβους, κάνοντας δυνατή την δημιουργία ενός πίνακα γειτνίασης. Ακόμα, μετά την εκτέλεση του αλγορίθμου τα παραγόμενα δεδομένα πρέπει να αποθηκεύονται στην βάση δεδομένων και να μπορεί κανείς να ξεχωρίσει τις διαφορετικές εκτελέσεις του αλγορίθμου.

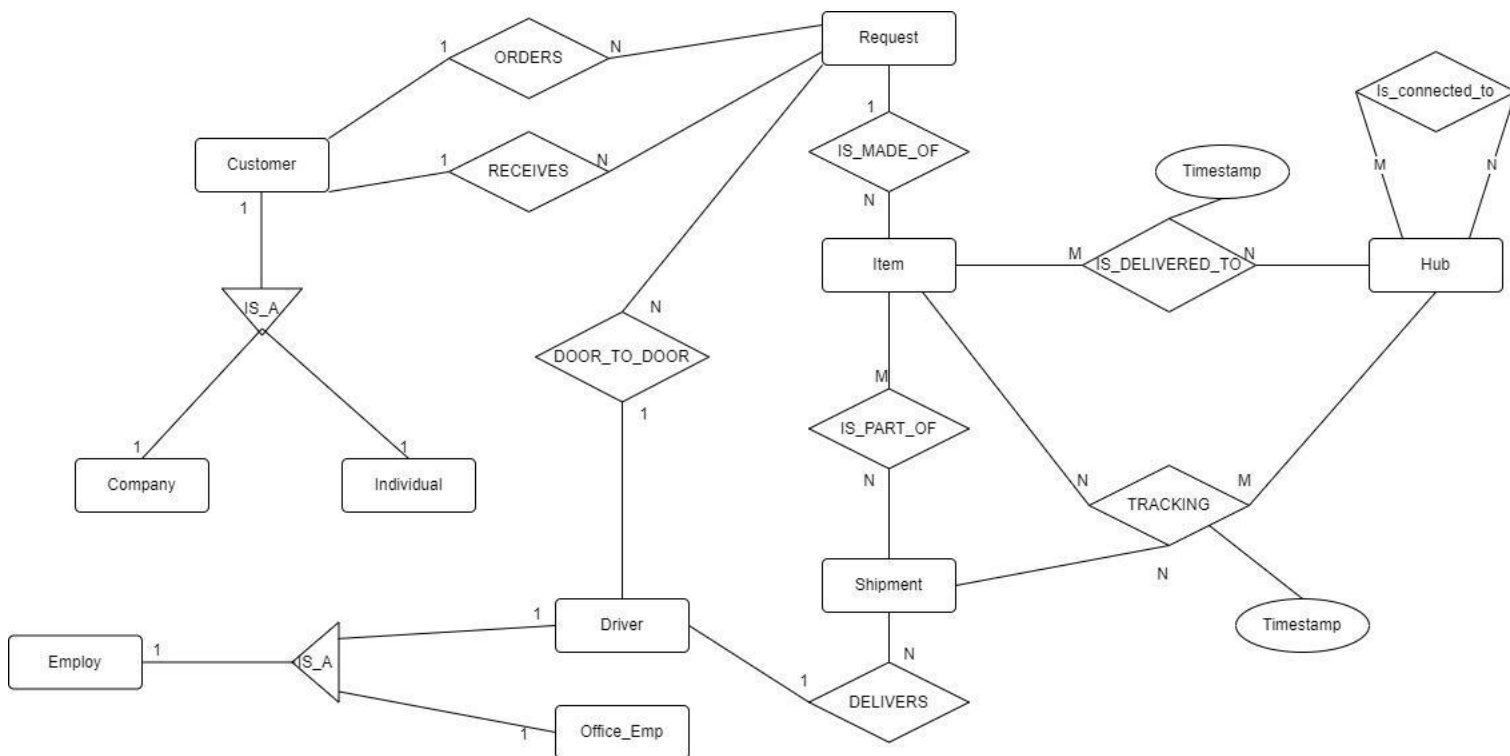
Συνεχίζοντας με τα στατιστικά δεδομένα που το σύστημα πρέπει να παράγει, η ανάλυση μας ακολούθησε μία προσέγγιση απλότητας. Δηλαδή θέλαμε η βάση δεδομένων να έχει ίσως περισσότερα από τα απαραίτητα δεδομένα για κάθε οντότητα. Ενώ, οι συναρτήσεις που θα παράγουν, τα γραφήματα και τα ποσοστά, θα υπάρχουν με μεγαλύτερη πολυπλοκότητα στον κώδικα του συστήματος και θα τρέχουν στο μηχάνημα του χρήστη.

Επιλέξαμε αυτήν την προσέγγιση επειδή τα στατιστικά αυτά δεν θα αποθηκεύονται στην βάση δεδομένων και θα παράγονται κατά παραγγελία του χρήστη. Παράλληλα με αυτήν την προσέγγιση κάνουμε τους σέρβερ του συστήματος πιο ανθεκτικούς σε περιπτώσεις μεγάλου αριθμού ταυτόχρονων χρηστών επειδή δεν υπάρχει υπολογιστικό κόστος για την παραγωγή στατιστικών δεδομένων.

Τέλος η βάση δεδομένων πρέπει να αποθηκεύει τα πιστήρια και την ιδιότητα των υπαλλήλων της RMR. Συγκεκριμένα, δύο θα είναι οι ιδιότητες των υπαλλήλων που θα έχουν πρόσβαση στο σύστημα. Αρχικά, οι γενικοί υπάλληλοι που θα διαχειρίζονται το σύνολο του συστήματος, δηλαδή την εκτέλεση του αλγορίθμου, την εξαγωγή στατιστικών στοιχείων και την εισαγωγή νέων παραγγελιών. Από την άλλη είναι οι οδηγοί της εταιρείας που θα έχουν πρόσβαση στο mobile app του συστήματος και θα μπορούν να δουν μονάχα το δρομολόγιο που τους ανατέθηκε να κάνουν, τα δέματα του δρομολογίου και τη διαδρομή. Η βάση έχει σχεδιαστεί για να μην επιτρέπει στους δύο να έχουν πρόσβαση στο υποσύστημα του άλλου και να διαχωρίζει ρητά τις ιδιότητες των δύο.

## 3.2 Παρουσίαση του διαγράμματος οντοτήτων - συσχετίσεων (ER) του συστήματος ΒΕΛΟΣ

Στην ενότητα αυτή θα παρουσιάσουμε διεξοδικά το διάγραμμα οντοτήτων συσχετίσεων που δημιουργήθηκε στα πλαίσια της ανάλυσης και του σχεδιασμού για την υλοποίηση του συστήματος. Θα εξηγήσουμε για τις οντότητες και τις συσχετίσεις αυτών με άλλες καθώς και την αναγκαιότητα τους στην βάση δεδομένων. Σε επόμενο κεφάλαιο θα αναλύσουμε τους πίνακες που θα υπάρχουν πραγματικά στην βάση δεδομένων, την λειτουργία τους και τον τύπο δεδομένων που θα αποθηκεύονται σε αυτούς. Ακολουθεί το διάγραμμα οντοτήτων συσχετίσεων



Εικόνα 2 – Διάγραμμα οντοτήτων συσχετίσεων του έργου ΒΕΛΟΣ



### **Πίνακας 1 – 3.2.1 Οντότητα: CUSTOMER**

Γενική περιγραφή της οντότητας CUSTOMER				
Η οντότητα CUSTOMER αναφέρεται σε οποιονδήποτε πελάτη εξυπηρετεί η εταιρεία RMR. Η οντότητα δε διαχωρίζεται σε ιδιώτη και εταιρεία.				
<b>Πληθυντικός</b>		CUSTOMERS		
<b>Διαφορετική ονομασία</b>				
<b>Συσχετίσεις</b>		<ol style="list-style-type: none"> <li>1. Κάθε πελάτης μπορεί να είναι είτε εταιρεία είτε ιδιώτης</li> <li>2. Κάθε πελάτης μπορεί να κάνει μία ή παραπάνω παραγγελίες</li> <li>3. Κάθε πελάτης μπορεί να παραλάβει μία η παραπάνω παραγγελίες</li> </ol>		
<b>Υλοποίηση πίνακα</b>		CUSTOMER		
<b>Επιμέρους Χαρακτηριστικά</b>				
Name	Datatype	Description	Domain	Default
Customer_ID	int(10)			
		Ο μοναδικός αριθμός αναγνώρισης για κάθε πελάτη.		
Postal_Code	varchar2(60)			
		Ο ταχυδρομικός κώδικας της περιοχής του πελάτη		
Contant_Number	varchar2(10)			
		Αριθμός τηλεφωνικής επικοινωνίας του πελάτη		
Delivery_Address	varchar2(100)			
		Η διεύθυνση παραλαβής των προϊόντων του πελάτη		

Type_of_Customer	Enum( <b>Company OR Individual</b> )		
	Ο τύπος του πελάτη. Μπορεί να είναι ιδιώτης ή επιχείρηση		
Email	varchar2(55)		
	Email επικοινωνίας με τον πελάτη		

### **Πίνακας 2 – 3.2.2 Οντότητα: INDIVIDUAL**

Γενική περιγραφή της οντότητας INDIVIDUAL			
Η οντότητα INDIVIDUAL είναι μία από τις δύο περιπτώσεις που μπορεί να είναι ένας πελάτης και εκφράζει αυτή του ιδιώτη			
Πληθυντικός	INDIVIDUALS		
Διαφορετική ονομασία	-		
Συσχετίσεις	1. Κάθε ιδιώτης μπορεί να είναι ένας και μόνο ένας πελάτης		
Υλοποίηση πίνακα	INDIVIDUAL		
Επιμέρους χαρακτηριστικά			
Name	Datatype Description	Domain	Default
Customer_ID	int(10)		
	Ο μοναδικός αριθμός αναγνώρισης για κάθε πελάτη		
First_Name	varchar2(60)		
	Το πρώτο όνομα του ιδιώτη		

Last_Name	varchar2(60)		
	Το επώνυμο του ιδιώτη		
Identity_Number	varchar2(15)		
	Ο αριθμός ταυτότητας του ιδιώτη		

### **Πίνακας 3 – 3.2.3 Οντότητα: COMPANY**

Γενική περιγραφή της οντότητας COMPANY			
Η οντότητα COMPANY είναι επίσης μία από τις δύο περιπτώσεις που μπορεί να είναι ένας πελάτης και εκφράζει την εταιρεία			
Πληθυντικός	COMPANIES		
Διαφορετική ονομασία	-		
Συσχετίσεις	1. Κάθε εταιρεία μπορεί να είναι ένας και μόνο ένας πελάτης		
Υλοποίηση πίνακα	COMPANY		
Επιμέρους χαρακτηριστικά			
Name	Datatype Description	Domain	Default
Customer_ID	int(10)		
	Ο μοναδικός αριθμός αναγνώρισης για κάθε πελάτη		
Company_Name	varchar2(60)		
	Η δηλωμένη επωνυμία της εταιρείας		
TIN	int(9)		
	Το ΑΦΜ της εταιρείας		



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Ταμείο  
Περιφερειακής Ανάπτυξης

**ΕΠΑνεΚ 2014-2020**  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ  
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ  
ΚΑΙΝΟΤΟΜΙΑ

**ΕΣΠΑ**  
2014-2020  
ανάπτυξη - εργασία - αλληλεγγύη

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

PFD	varchar2(100)		
	Η ΔΟΥ στην οποία είναι δηλωμένη η εταιρεία		

#### **Πίνακας 4 – 3.2.4 Οντότητα: REQUEST**

Γενική περιγραφή της οντότητας REQUEST			
Η οντότητα REQUEST περιγράφει μία αίτηση για μία κίνηση. Δηλαδή περιγράφει την αποστολή ή παράδοση ενός δέματος.			
<b>Πληθυντικός</b>	REQUESTS		
<b>Διαφορετική ονομασία</b>	-		
<b>Συσχετίσεις</b>	<ol style="list-style-type: none"> <li>1. Κάθε αίτηση μπορεί να αποτελείται από ένα ή περισσότερα τεμάχια(items)</li> <li>2. Κάθε παραγγελία έχει έναν παραλήπτη</li> <li>3. Κάθε παραγγελία έχει έναν αποστολέα</li> <li>4. Μία ή πολλές παραγγελίες διανέμονται από έναν διανομέα</li> </ol>		
<b>Υλοποίηση πίνακα</b>	REQUEST		
Επιμέρους χαρακτηριστικά			
Name	Datatype Description	Domain	Default
Request_ID	int(10)		
	Ο μοναδικός αριθμός αναγνώρισης κάθε αίτησης		
Orderer_ID	int(10)		
	Ο μοναδικός αριθμός αναγνώρισης κάθε πελάτη που κάνει την παραγγελία		
Receiver_ID	int(10)		



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Ταμείο  
Περιφερειακής Ανάπτυξης

**ΕΠΑνεΚ 2014-2020**  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ  
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ  
ΚΑΙΝΟΤΟΜΙΑ

**ΕΣΠΑ**  
2014-2020  
ανάπτυξη - εργασία - αλληλεγγύη

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

	Ο μοναδικός αριθμός αναγνώρισης κάθε πελάτη που λαμβάνει την παραγγελία		
Request_status	Enum (In_transit OR Last_mile OR Delivered)		
	Παίρνει την αντίστοιχη τιμή TRUE ή FALSE όταν η αίτηση αναφέρεται σε εύθραυστο προϊόν		
Priority	Enum (Low OR Normal OR High)		
	Η προτεραιότητα της αίτησης. Ορίζεται από τον πελάτη		
Total_volumetric_weight	float(8)		
	Ορίζει το συνολικό ογκομετρικό βάρος της παραγγελίας		

### **Πίνακας 5 – 3.2.5 Οντότητα: ITEM**

Γενική περιγραφή της οντότητας ITEM	
Η οντότητα ITEM περιγράφει τα περιεχόμενα μιας παραγγελίας,όσα και να είναι αυτά	
Πληθυντικός	ITEMS
Διαφορετική ονομασία	-
Συσχετίσεις	<ol style="list-style-type: none"> <li>1. Κάθε τεμάχιο αποστέλλεται σε ένα ή περισσότερα κέντρα διανομής</li> <li>2. Κάθε τεμάχιο παρακολουθείται μία ή περισσότερες φορές</li> <li>3. Κάθε τεμάχιο είναι μέρος μιας ή περισσότερων μεταφορών</li> <li>4. Ένα ή πολλά τεμάχια αποτελούν μία παραγγελία</li> </ol>
Υλοποίηση πίνακα	ITEM



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Ταμείο  
Περιφερειακής Ανάπτυξης

**ΕΠΑνεΚ 2014-2020**  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ  
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ  
ΚΑΙΝΟΤΟΜΙΑ

**ΕΣΠΑ**  
2014-2020  
ανάπτυξη - εργασία - αλληλεγγύη

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Επιμέρους χαρακτηριστικά			
Name	Datatype Description	Domain	Default
Item_ID	int(10)		
	Ο μοναδικός αριθμός αναγνώρισης κάθε αίτησης		
Request_ID	int(10)		
	Ο μοναδικός αριθμός αναγνώρισης κάθε πελάτη που κάνει την παραγγελία		
Item_type	Enum( <b>Parcel OR Pallete</b> )		
	Εκφράζει την μορφή του τεμαχίου και διαφοροποιεί τα δέματα από τις παλέτες		
Height	int(4)		
	Εκφράζει το ύψος του τεμαχίου		
Length	int(4)		
	Εκφράζει το μήκος του τεμαχίου		
Width	int(4)		
	Εκφράζει το πλάτος του τεμαχίου		
Weight	int(4)		
	Εκφράζει το βάρος του τεμαχίου		
Vulnerable	boolean		
	Εκφράζει αν το τεμάχιο είναι ευάλωτο		
Dangerous	boolean		
	Εκφράζει αν το τεμάχιο είναι επικίνδυνο		

Fragile	boolean		
	Εκφράζει αν το τεμάχιο είναι ευθραυστο		
Volumetric weight	float(8)		
	Το συνολικό ογκομετρικό βάρος του τεμαχίου με βάση τα παραπάνω χαρακτηριστικά		

### **Πίνακας 6 – 3.2.6 Οντότητα: SHIPMENT**

Γενική περιγραφή της οντότητας SHIPMENT			
Η οντότητα SHIPMENT εκφράζει τη μεταφορά δεμάτων και παλετών που ανήκουν σε μία αίτηση από ένα κέντρο διανομής σε ένα άλλο.			
<b>Πληθυντικός</b>	SHIPMENTS		
<b>Διαφορετική ονομασία</b>	-		
<b>Συσχετίσεις</b>	<ol style="list-style-type: none"> <li>1. Κάθε μεταφορά περιέχει ένα ή παραπάνω τεμάχια(Items)</li> <li>2. Μία η παραπάνω μεταφορές εκτελούνται από έναν οδηγό</li> <li>3. Κάθε μεταφορά παρακολουθείται μία ή παραπάνω φορές</li> </ol>		
<b>Υλοποίηση Πίνακα</b>	SHIPMENT		
Επιμέρους χαρακτηριστικά			
Name	Datatype Description	Domain	Default
Shipment_ID	int(10)		
	Ο μοναδικός αριθμός αναγνώρισης κάθε μεταφοράς		
User_ID	int(10)		

	Ο μοναδικός αριθμός αναγνώρισης κάθε εργαζόμενου( συγκεκριμένα οδηγού)		
Location_Start_ID	int(10)		
	Ο μοναδικός αριθμός αναγνώρισης του κέντρου έναρξης της μεταφοράς		
Location_End_ID	int(10)		
	Ο μοναδικός αριθμός αναγνώρισης του κέντρου αφίξης της μεταφοράς		
Quantity	float(8)		
	Το ποσό που μεταφέρεται σε μονάδα μέτρησης ογκομετρικού βάρους		
Cost	float(8)		
	Το κόστος μεταφοράς		
Data_made	timestamp - HH/MM/SS DD/MM/YYYY		
	Η ημερομηνία δημιουργίας της μεταφοράς		

### **Πίνακας 7 – 3.2.7 Οντότητα: HUB**

<b>Γενική περιγραφή της οντότητας HUB</b>	
<p>Η οντότητα HUB περιγράφει αποθήκες και κέντρα διανομής στο οδικό δίκτυο της εταιρείας. Τα αιτήματα μεταφέρονται από και προς αυτά καθώς εκφράζει τις στάσεις που ένα αίτημα θα κάνει προς το σημείο παράδοσης του στον τελικό αποδέκτη</p>	
<b>Πληθυντικός</b>	HUBS
<b>Διαφορετική ονομασία</b>	-
<b>Συσχετίσεις</b>	<ol style="list-style-type: none"> <li>1. Κάθε κέντρο διανομής παραλαμβάνει ένα ή περισσότερα αιτήματα</li> <li>2. Κάθε κέντρο διανομής</li> </ol>



	παρακολουθείται μία ή περισσότερες φορές		
<b>Υλοποίηση Πίνακα</b>	HUB		
<b>Επιμέρους χαρακτηριστικά</b>			
<b>Name</b>	<b>Datatype Description</b>	<b>Domain</b>	<b>Default</b>
Hub_ID	int(10)		
	Ο μοναδικός αριθμός αναγνώρισης ενός κέντρου διανομής		
Hub_name	varchar2(60)		
	Η ονομασία του κέντρου διανομής(π.χ RMR υποκατάστημα βόλου)		
Company_name	varchar2(60)		
	Το όνομα της εταιρείας στο οποίο ανήκει το κέντρο(για το πάνω παράδειγμα η τιμή αυτού του πεδίου θα ήταν RMR)		
Address	varchar2(60)		
	Η διεύθυνση του κέντρου διανομής		
City	varchar2(60)		
	Η πόλη στην οποία βρίσκεται το κέντρο		
Max_capacity	float(8)		
	Η μέγιστη χωρητικότητα του κέντρου διανομής και μετριέται σε μονάδες ογκομετρικού βάρους		
Current_capacity	float(8)		
	Η τρέχον χωρητικότητα του κέντρου διανομής, μετριέται επίσης σε μονάδες ογκομετρικού βάρους		
Postal_code	varchar2(10)		
	Ο ταχυδρομικός κώδικας του κέντρου		
Longitude	float(9)		

	Το γεωμετρικό μήκος της τοποθεσίας του κέντρου		
Latitude	float(9)		
	Το γεωμετρικό πλάτος της τοποθεσίας του κέντρου		

### **Πίνακας 8 – 3.2.8 Οντότητα: EMPLOY**

Γενική περιγραφή της οντότητας EMPLOY			
Η οντότητα EMPLOY εκφράζει τους υπαλλήλους της εταιρείας που διαθέτουν πρόσβαση στο σύστημα ΒΕΛΟΣ			
<b>Πληθυντικός</b>	EMPLOYEES		
<b>Διαφορετική ονομασία</b>	-		
<b>Συσχετίσεις</b>	1. Ένας υπάλληλος μπορεί να είναι οδηγός ή υπάλληλος γραφείου		
<b>Υλοποίηση πίνακα</b>	EMPLOY		
Επιμέρους χαρακτηριστικά			
Name	Datatype Description	Domain	Default
User_ID	int(10)		
	Το μοναδικό αναγνωριστικό για κάθε χρήστη του συστήματος		
Username	varchar2(18)		
	Το όνομα χρήστη του υπαλλήλου		
Password	varchar2(18)		
	Ο κωδικός του υπαλλήλου		

Type_of_Employ	Enum(Driver OR Office_employ)		
	Εκφράζει την ιδιότητα του υπαλλήλου σε οδηγό ή υπάλληλο γραφείου		
Company_Assigned	varchar2(18)		
	Η εταιρεία στην οποία εργάζεται ο υπάλληλος		

### **Πίνακας 9 – 3.2.9 Οντότητα: DRIVER**

Γενική περιγραφή της οντότητας DRIVER			
Η οντότητα DRIVER εκφράζει τους οδηγούς της εταιρείας που διαθέτουν πρόσβαση στο mobile σύστημα ΒΕΛΟΣ			
Πληθυντικός	DRIVERS		
Διαφορετική ονομασία	-		
Συσχετίσεις	<ol style="list-style-type: none"> <li>Ένας οδηγός μπορεί να είναι ένας και μόνος ένας υπάλληλος</li> <li>Ένας οδηγός μπορεί να μεταφέρει ένα ή περισσότερα φορτία προς μεταφορά</li> <li>Ένας οδηγός μπορεί να διανέμει πόρτα, πόρτα μία ή παραπάνω παραγγελίες</li> </ol>		
Υλοποίηση πίνακα	DRIVER		
Επιμέρους χαρακτηριστικά			
Name	Datatype Description	Domain	Default
User_ID	int(10)		
	Το μοναδικό αναγνωριστικό κάθε χρήστη του συστήματος		
Name	varchar2(18)		

	Το όνομα του οδηγού		
Surname	varchar2(18)		
	Το επώνυμο του οδηγού		
Type_of_Driver	Enum (Truck_Driver OR Courier)		
	Εκφράζει τον τύπο του οδηγού σε οδηγό φορτηγού και courier		

### **Πίνακας 10 – 3.2.10 Οντότητα: OFFICE EMPLOY**

Γενική περιγραφή της οντότητας OFFICE_EMPLOY			
Η οντότητα OFFICE_EMPLOY εκφράζει τους υπαλλήλους της εταιρείας που διαθέτουν πρόσβαση στο web σύστημα ΒΕΛΟΣ			
Πληθυντικός	Office_employees		
Διαφορετική ονομασία	-		
Συσχετίσεις	1. Ένας υπάλληλος γραφείου μπορεί να είναι ένας και μόνος ένας υπάλληλος		
Υλοποίηση πίνακα	OFFICE_EMPLOY		
Επιμέρους χαρακτηριστικά			
Name	Datatype Description	Domain	Default
User_ID	int(10)		
	Το μοναδικό αναγνωριστικό κάθε χρήστη του συστήματος		
Name	varchar2(18)		
	Το όνομα του υπαλλήλου γραφείου		



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Ταμείο  
Περιφερειακής Ανάπτυξης

ΕΠΑνεΚ 2014-2020  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ  
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ  
ΚΑΙΝΟΤΟΜΙΑ



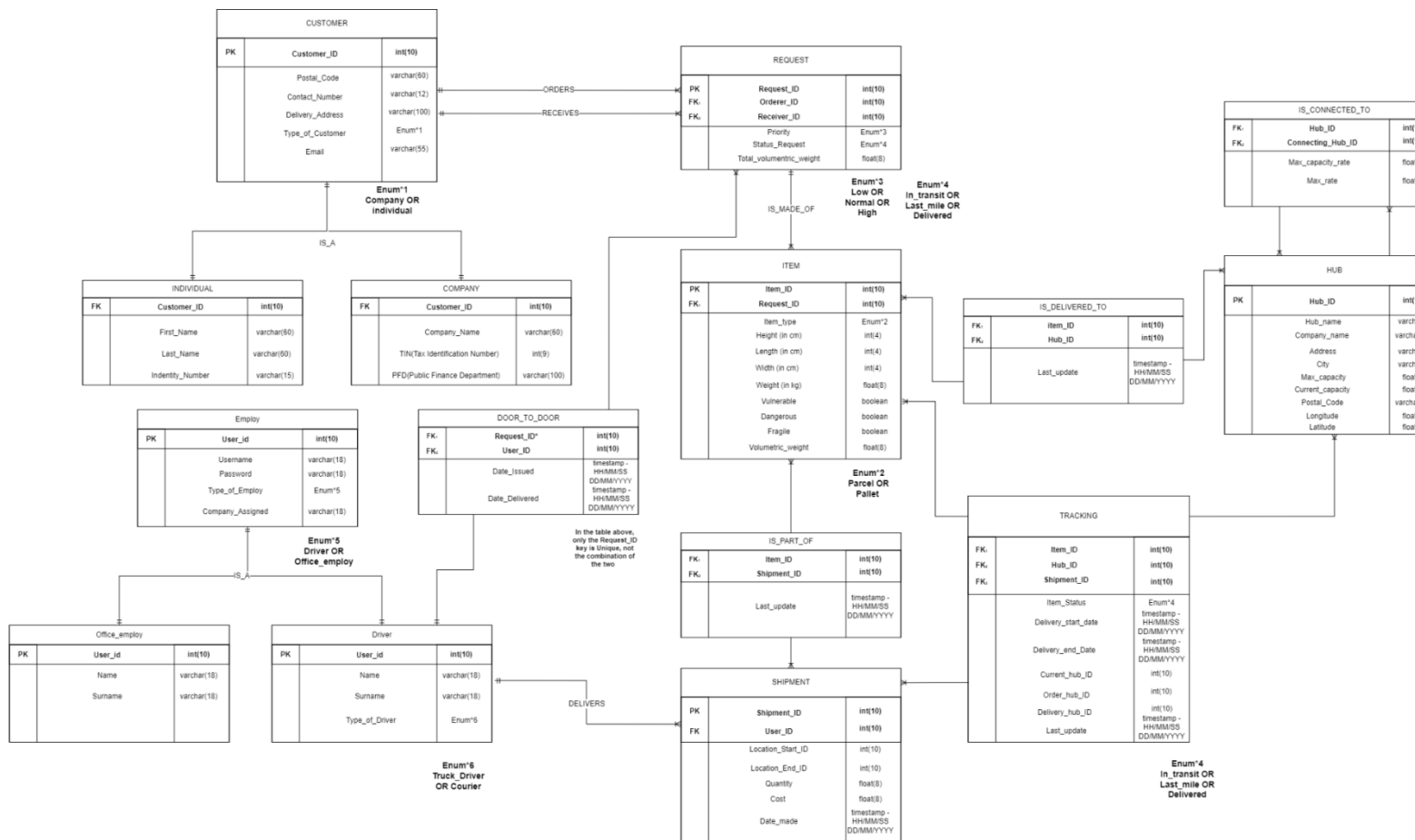
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Surname	varchar2(18)		
	Το επώνυμο του υπαλλήλου γραφείου		

### 3.3 Παρουσίαση του διαγράμματος της ολοκληρωμένης βάσης δεδομένων

Στην ενότητα αυτή θα γίνει η παρουσίαση των πινάκων της εφαρμογής οι οποίοι εξυπηρετούν τις προς υλοποίηση λειτουργίες. Οι πίνακες προέκυψαν από τα διαγράμματα οντοτήτων - συσχετίσεων που παρουσιάστηκαν στην προηγούμενη ενότητα και περιγράφουν τους πραγματικούς πίνακες τις βάσεις. Αυτοί αποτελούνται από τις οντότητες που έχουν ήδη αναφέρει αλλά και από τους παραγωγόμενους πίνακες που προκύπτουν από τις συσχετίσεις των οντοτήτων αυτών. Παράλληλα θα αναφερθεί η χρήση κάθε πίνακα στα προς υλοποίηση τμήματα του λογισμικού μαζί με την αναγκαιότητα του καθ ενός πίνακα για την λειτουργία αυτών. Ακόμα θα αναφερθούν περισσότερες λεπτομέρειες όσον αφορά τα πεδία του κάθε πίνακα, καθώς στην προηγούμενη ενότητα αναφέραμε τον τύπο δεδομένων που θα έχουν τα πεδία των πινάκων αλλά όχι τους περιορισμούς που θα έχουν. Σημειώνεται τέλος, ότι τα σχόλια για τους πίνακες και τα πεδία αυτών θα είναι αγγλικά, όπως ακριβώς έχουν αποθηκευτεί στο σχήμα της Βάσης δεδομένων.

Στην επόμενη σελίδα παρουσιάζεται το διάγραμμα της βάσης δεδομένων (Data Schema)



Εικόνα 3 – Το πλήρες διάγραμμα της ενδιάμεσης βάσης δεδομένων

### Πίνακας 11 – 3.3.1 Πίνακας: CUSTOMER

Γενική Περιγραφή Πίνακα CUSTOMER	
Alias	-
Rows	
Primary Key	Customer_ID
Index(s)	No

#### Επιμέρους χαρακτηριστικά

Column	Data Type / Comments	Max	Dec	Null	Up?
Customer_ID	int	10	1	No	N
	The Customer's ID				
Postal_Code	varchar2	60		Yes	N
	The postal code				
Contact_Number	varchar2	12	12	No	N
	The contact number				
Delivery_Address	varchar2	100	5	No	N
	The address to be delivered				
Type_of_Customer	Enum( <b>Company OR Individual</b> )			No	N
	The type of customer				
Email	varchar2	55		Yes	N
	The email address				

## Πίνακας 12 – 3.3.2 Πίνακας: **INDIVIDUAL**

Γενική Περιγραφή Πίνακα <b>INDIVIDUAL</b>	
<b>Alias</b>	-
<b>Rows</b>	
<b>Primary Key</b>	Customer_ID
<b>Index(s)</b>	Customer_FK1

### Επιμέρους χαρακτηριστικά

Column	Data Type / Comments	Max	Dec	Null	Up?
Customer_ID	int	10	1	No	N
	The Customer's ID				
First_Name	varchar2	60	12	No	N
	The Individual's first name				
Last_Name	varchar2	60	12	No	N
	The Individual's last name				
Identity_Number	varchar2	15	15	No	N
	The Individual's ID number				



### **Πίνακας 13 – 3.3.3 Πίνακας: COMPANY**

Γενική Περιγραφή Πίνακα COMPANY	
Alias	-
Rows	
Primary Key	Customer_ID
Index(s)	Customer_FK1

#### **Επιμέρους χαρακτηριστικά**

Column	Data Type / Comments	Max	Dec	Null	Up?
Customer_ID	int	10	1	No	N
	The Customer's ID				
Company_name	varchar2	60	12	No	N
	The Company's registered name				
TIN	int	60	12	No	N
	Tax Identification Number				
PFD	varchar2	15	15	No	N
	Public Finance Department				

### **Πίνακας 14 – 3.3.4 Πίνακας: REQUEST**

Γενική Περιγραφή Πίνακα <b>REQUEST</b>	
<b>Alias</b>	-
<b>Rows</b>	
<b>Primary Key</b>	Request_ID
<b>Index(s)</b>	Customer_FK1 Customer_FK2

#### **Επιμέρους χαρακτηριστικά**

<b>Column</b>	<b>Data Type / Comments</b>	<b>Max</b>	<b>Dec</b>	<b>Null</b>	<b>Up?</b>
Request_ID	int	10	1	No	N
	The Request's ID				
Orderer_ID	int	10	1	No	N
	The Orderer's ID				
Receiver_ID	int	10	1	No	N
	The Receiver's ID				
Priority	Enum( <b>Low OR Normal OR High</b> )			No	N
	The Priority of the Request				
Status_Request	Enum( <b>In_transit OR Last_mile OR Delivered</b> )			Yes	N
	The status of the Request				
Total_volumetric_weight	float	8		Yes	N
	The total volumetric weight of the order				

### **Πίνακας 15 – 3.3.5 Πίνακας: ITEM**

Γενική Περιγραφή Πίνακα <b>ITEM</b>	
<b>Alias</b>	-
<b>Rows</b>	
<b>Primary Key</b>	Item_ID
<b>Index(s)</b>	Request_FK1

#### **Επιμέρους χαρακτηριστικά**

<b>Column</b>	<b>Data Type / Comments</b>	<b>Max</b>	<b>Dec</b>	<b>Null</b>	<b>Up?</b>
Item_ID	int	10	1	No	N
	The Item's ID				
Request_ID	int	10	1	No	N
	The Request's ID				
Item_type	Enum( <b>Parcel OR Pallette</b> )			No	N
	The type of the item				
Height	int	4		Yes	N
	The height of the item				
Length	int	4		Yes	N
	The length of the item				
Width	int	4		Yes	N
	The Width of the item				
Weight	int	4		Yes	N
	The type of the item				
Vulnerable	boolean			No	N

	The type of the item				
Dangerous	boolean			No	N
	The type of the item				
Fragile	boolean			No	N
	The type of the item				
Volumetric_Weight	float	8		Yes	N
	The type of the item				

### **Πίνακας 16 – 3.3.6 Πίνακας: IS\_PART\_OF**

Γενική Περιγραφή Πίνακα <b>IS_PART_OF</b>	
<b>Alias</b>	-
<b>Rows</b>	
<b>Primary Key</b>	Item_ID Shipment_ID
<b>Index(s)</b>	Item_FK1 Shipment_FK2

### **Επιμέρους χαρακτηριστικά**

Column	Data Type / Comments	Max	Dec	Null	Up?
Item_ID	int	10	1	No	N
	The Item's ID				
Shipment_ID	int	10	1	No	N
	The shipment's ID				
Last_update	Timestamp			No	N
	The last time it was updated				

### **Πίνακας 17 – 3.3.7 Πίνακας: SHIPMENT**

Γενική Περιγραφή Πίνακα <b>SHIPMENT</b>	
Alias	-
Rows	
Primary Key	SHIPMENT_ID
Index(s)	USER_FK1

#### Επιμέρους χαρακτηριστικά

Column	Data Type / Comments	Max	Dec	Null	Up?
Shipment_ID	int	10	1	No	N
	The Shipment's ID				
User_ID	int	10	1	No	N
	The User's ID				
Location_Start_ID	int	10	1	No	N
	The starting location ID				
Location_End_ID	int	10	1	No	N
	The ending location ID				
Quantity	float	8		Yes	N
	The quantity to be shipped				
Cost	float	8		Yes	N
	The cost of the shipment				
Date_made	timestamp			No	N
	The date that the shipment				

	was made				
--	----------	--	--	--	--

### Πίνακας 18 – 3.3.8 Πίνακας: EMPLOYEE

Γενική Περιγραφή Πίνακα EMPLOYEE	
Alias	-
Rows	
Primary Key	User_ID
Index(s)	No

#### Επιμέρους χαρακτηριστικά

Column	Data Type / Comments	Max	Dec	Null	Up?
User_ID	int	10	1	No	N
	The User's ID				
Username	varchar2	18	9	No	N
	The user's username				
Password	varchar2	18	9	No	N
	The user's password				
Type_of_employee	Enum( <b>Driver OR Office_employee</b> )			No	N
	The type of the employee				
Company_assigned	varchar(18)	18	9	No	N
	The company that the employee is working for				

### Πίνακας 19 – 3.3.9 Πίνακας: OFFICE\_EMPLOYEE

Γενική Περιγραφή Πίνακα OFFICE_EMPLOYEE	
Alias	-
Rows	
Primary Key	User_ID
Index(s)	User_FK1

#### Επιμέρους χαρακτηριστικά

Column	Data Type / Comments	Max	Dec	Null	Up?
User_ID	int	10	1	No	N
	The User's ID				
Name	varchar2	18	9	No	N
	The office employee's name				
Surname	varchar2	18	9	No	N
	The office employee's surname				

## **Πίνακας 20 – 3.3.10 Πίνακας: DRIVER**

Γενική Περιγραφή Πίνακα <b>DRIVER</b>	
<b>Alias</b>	-
<b>Rows</b>	
<b>Primary Key</b>	User_ID
<b>Index(s)</b>	User_FK1

### **Επιμέρους χαρακτηριστικά**

<b>Column</b>	<b>Data Type / Comments</b>	<b>Max</b>	<b>Dec</b>	<b>Null</b>	<b>Up?</b>
User_ID	int	10	1	No	N
	The User's ID				
Name	varchar2	18	9	No	N
	The driver's name				
Surname	varchar2	18	9	No	N
	The driver's surname				
Type_of_Driver	Enum( <b>Truck_Driver OR Courier</b> )			No	N
	The type of the driver				



## Πίνακας 21 – 3.3.11 Πίνακας: DOOR TO DOOR

Γενική Περιγραφή Πίνακα <b>DRIVER</b>	
<b>Alias</b>	Last_mile
<b>Rows</b>	
<b>Primary Key</b>	Request_ID
<b>Index(s)</b>	Request_FK1 User_FK2

### Επιμέρους χαρακτηριστικά

Column	Data Type / Comments	Max	Dec	Null	Up?
Request_ID	int	10	1	No	N
	The Request's ID				
User_ID	int	10	1	No	N
	The driver's ID				
Date_issued	timestamp			No	N
	The date that the courier got the request for delivery				
Date_delivered	timestamp			No	N
	The date that the courier delivered the request				

## **Πίνακας 22 – 3.3.12 Πίνακας: IS\_DELIVERED\_TO**

Γενική Περιγραφή Πίνακα <b>IS_DELIVERED_TO</b>	
<b>Alias</b>	
<b>Rows</b>	
<b>Primary Key</b>	Item_ID Hub_ID
<b>Index(s)</b>	Hub_FK1 Item_FK2

### **Επιμέρους χαρακτηριστικά**

<b>Column</b>	<b>Data Type / Comments</b>	<b>Max</b>	<b>Dec</b>	<b>Null</b>	<b>Up?</b>
Item_ID	int	10	1	No	N
	The Item's ID				
Hub_ID	int	10	1	No	N
	The Hub's ID				
Last_update	timestamp			No	N
	The date that the courier got the request for delivery				

### **Πίνακας 23 – 3.3.13 Πίνακας: HUB**

Γενική Περιγραφή Πίνακα <b>HUB</b>	
Alias	-
Rows	
Primary Key	Hub_ID
Index(s)	No

#### Επιμέρους χαρακτηριστικά

Column	Data Type / Comments	Max	Dec	Null	Up?
Hub_ID	int	10	1	No	N
	The Hub's ID				
Hub_name	varchar2	60	15	No	N
	The Hub's name				
Company_name	varchar2	60	15	No	N
	The company that owns the Hub				
Address	varchar2	60	15	No	N
	The address of the Hub				
City	varchar2	60	15	No	N
	The city that the Hub is located				
Max_capacity	float	8	1	No	N
	The max capacity of the Hub				
Current_capacity	float	8	1	No	N
	The current capacity of the				

	Hub				
Postal_code	varchar2	60	15	No	N
	The postal code of the hub				
Longitude	float	9	5	No	N
	The Longitude of the Hub				
Latitude	float	9	5	No	N
	The Latitude of the Hub				

#### **Πίνακας 24 – 3.3.14 Πίνακας: IS\_CONNECTED\_TO**

Γενική Περιγραφή Πίνακα IS_CONNECTED_TO	
<b>Alias</b>	
<b>Rows</b>	
<b>Primary Key</b>	Hub_ID Connecting_Hub_ID
<b>Index(s)</b>	Hub_FK1 Hub_FK2

#### **Επιμέρους χαρακτηριστικά**

Column	Data Type / Comments	Max	Dec	Null	Up?
Hub_ID	int	10	1	No	N
	The Item's ID				
Connecting_Hub_ID	int	10	1	No	N

	The Hub's ID that is connected to				
Max_capacity_rate	float	8		No	N
	The max capacity that can be transfer in the stream that connects the two nodes				
Max_rate	The max rate of shipments that the stream of the two nodes supports at the same time				

### **Πίνακας 25 – 3.3.15 Πίνακας: TRACKING**

Γενική Περιγραφή Πίνακα <b>TRACKING</b>	
<b>Alias</b>	-
<b>Rows</b>	
<b>Primary Key</b>	Item_ID Hub_ID Shipment_ID
<b>Index(s)</b>	Item_FK1 Hub_FK2 Shipme_FK3

#### **Επιμέρους χαρακτηριστικά**

Column	Data Type / Comments	Max	Dec	Null	Up?
Item_ID	int	10	1	No	N
	The Item's ID				
Hubt_ID	int	10	1	No	N
	The Hub's ID				

Shipment_ID	int	10	1	No	N
	The Shipment's ID				
Item_status	Enum(In_transit OR At_last_mile OR Delivered)			No	N
	The status of the item				
Delivery_start_date	timestamp			No	N
	The date that the delivery was ordered				
Delivery_end_date	timestamp			No	N
	The date that the delivery got delivered				
Current_Hub_ID	int	10	1	No	N
	The Hub's ID of the current hub				
Order_Hub_ID	int	10	1	No	N
	The Hub's ID of the starting Hub				
Delivery_Hub_ID	int	10	1	No	N
	The Hub's ID of the last Hub				
Last_update	timestamp			No	N
	The date of the last update				

## 4. Παρουσίαση του User Interface του έργου

### ΒΕΛΟΣ

Σε αυτό το κεφάλαιο θα παρουσιάσουμε την γραφική διεπαφή του έργου ΒΕΛΟΣ. Θα αναφέρουμε το λογισμικό μοντελοποίησης το οποίο μας επέτρεψε να σχεδιάσουμε το πρότυπο της γραφικής διεπαφής του έργου. Επίσης θα αναλύσουμε την φιλοσοφία με την οποία σχεδιάσαμε την διεπαφή που θα βλέπει ο χρήστης αλλά και για την εμπειρία που θέλουμε να έχει. Τέλος, θα εξηγήσουμε τις επιλογές που έχουμε κάνει όσον αφορά την διάταξη των οθονών ενώ παράλληλα θα μιλήσουμε για τους τρόπους πλοήγησης στην εφαρμογή.

### 4.1 Εισαγωγή στην ανάλυση του User Interface του έργου

#### ΒΕΛΟΣ

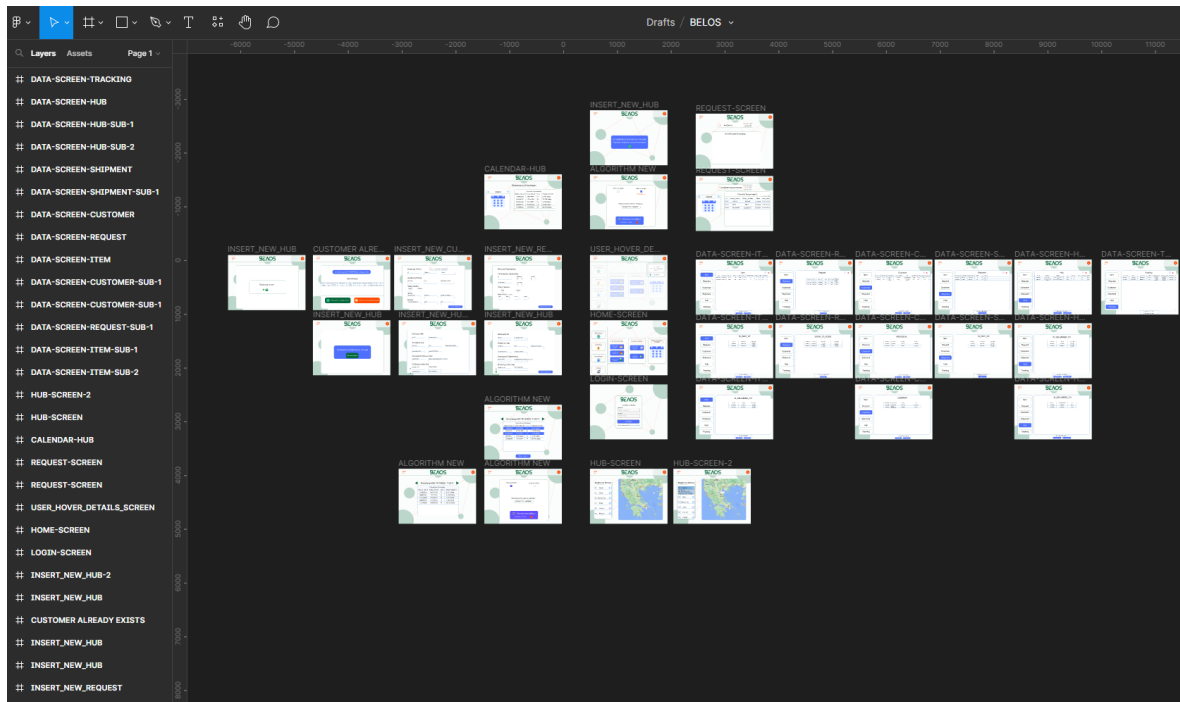
Για τις ανάγκες του έργου και της καλύτερης οργάνωσης της ομάδας που εργάζεται σε αυτό χρειάστηκε να φτιαχτεί ένα πρότυπο για την γραφική διεπαφή του συστήματος. Αυτό θα μας έδωσε την δυνατότητα να κάνουν τα εξής: Πρώτον να αναλύσουμε τις ανάγκες που θα είχε μία τέτοια διεπαφή και τους τρόπους υλοποίησης της. Για αυτόν τον λόγο, κάναμε έρευνα στις διάφορες γλώσσες προγραμματισμού που υπάρχουν για την υλοποίηση της διεπαφής αλλά και για τις λειτουργίες του συστήματος. Δηλαδή φτιάχνοντας το πρότυπο έπρεπε να διαχωρίσουμε τι υπολογισμοί και αλγόριθμοι θα εκτελούνται από την πλευρά του χρήστη(client-side) και τι από την πλευρά του σέρβερ(server side). Σαφώς η εμπειρία που έχουμε πάνω σε παρόμοια έργα, σαν το ΒΕΛΟΣ, βοήθησε στην εξακρίβωση των παραπάνω αλλά η οπτικοποίηση των αναγκών ήταν το καθοριστικό κομμάτι στην διευκρίνιση τους.

Δεύτερον μας έδωσε την δυνατότητα να έχουμε ένα κοινό σημείο αναφοράς καθ' όλη την διάρκεια του κύκλου ανάπτυξης του έργου. Ωστόσο το πρότυπο που θα παρουσιάσουμε στην επόμενη ενότητα είναι το καλύτερο δυνατό που μπορούμε να κάνουμε στην περίπτωση του έργου. Είμαστε αρκετά βέβαιοι ότι τα επιμέρους τμήματα που έχουμε

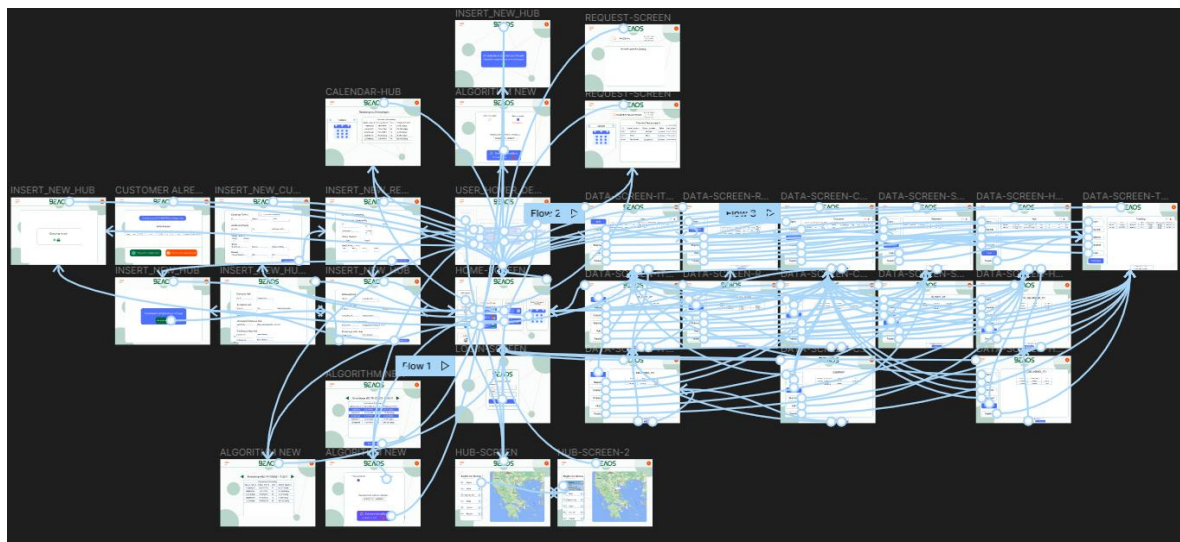
προδιαγράψει είναι εφικτά, ωστόσο το σύστημα που θα παραχθεί θα είναι ένα με πολλά κινούμενα κομμάτια και πάντα υπάρχει περίπτωση κάποιο από αυτά να μην εξελιχθεί βάση σχεδιασμού.

Το πρότυπο του συστήματος ΒΕΛΟΣ έχει δημιουργηθεί στο λογισμικό Figma. Το Figma είναι ένα εργαλείο wireframing που βασίζεται στο cloud για σχεδιαστές περιβάλλοντος εργασίας χρήστη και σχεδιασμού της εμπειρίας χρήσης (UX - User Experience). Η πρώτη του κυκλοφορία έγινε το 2016 και τα τελευταία χρόνια έχει γίνει ένα από τα πιο γνωστά και ευρέως χρησιμοποιούμενα εργαλεία σχεδιασμού παγκοσμίως. Τα πλεονεκτήματα χρήσης του Figma, που θα αναφέρουμε παρακάτω, είναι οι λόγοι που μας οδήγησαν στην επιλογή του έναντι άλλων παρόμοιων προγραμμάτων. Το Figma λειτουργεί σε οποιοδήποτε λειτουργικό σύστημα που εκτελεί προγράμματα περιήγησης ιστού. Mac, υπολογιστές Windows και συγκεκριμένα tablet. Παράλληλα, οι κινητές συσκευές έχουν επίσης την δυνατότητα να το χρησιμοποιήσουν, με λιγότερες λειτουργίες. Το Figma είναι εξ' ολοκλήρου cloud-native, αυτό το κάνει κατάλληλο για απομακρυσμένες ομάδες ανάπτυξης που χρειάζεται λογισμικά για συνεργατική εργασία. Τέλος έχει την δυνατότητα ενσωμάτωσης καναλιών επικοινωνίας με τρίτα λογισμικά όπως το slack, μέσω αυτού μπορεί κανείς να γράψει και να στείλει σχόλια σε συνεργάτες. Οι παραπάνω λόγοι μας έκαναν να επιλέξουμε το λογισμικό Figma





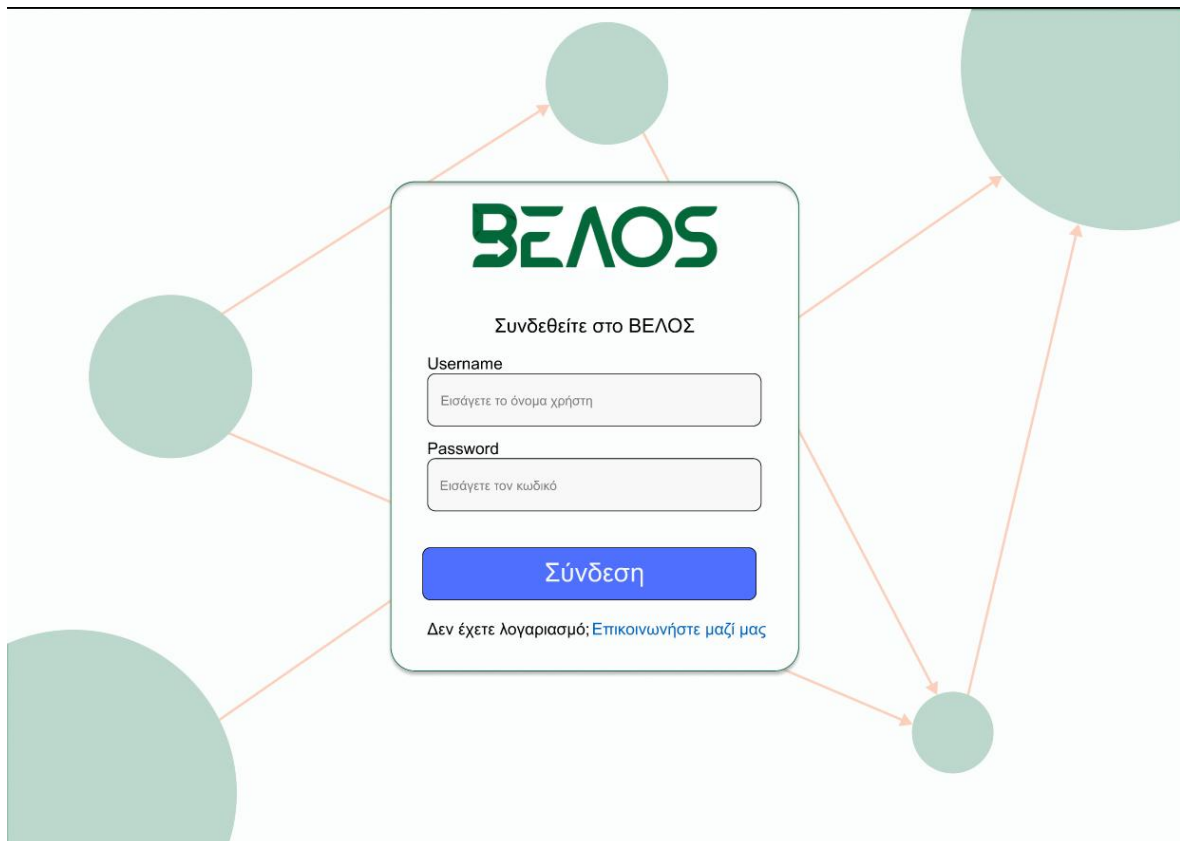
Εικόνα 4 – Το περιβάλλον εργασίας για το έργο ΒΕΛΟΣ



Εικόνα 5 –: Αλληλεπιδράσεις και μονοπάτια πλοήγησης στις οθόνες του έργου ΒΕΛΟΣ

## 4.2 Ανάλυση και επεξήγηση των διαφορετικών οθονών του έργου ΒΕΛΟΣ

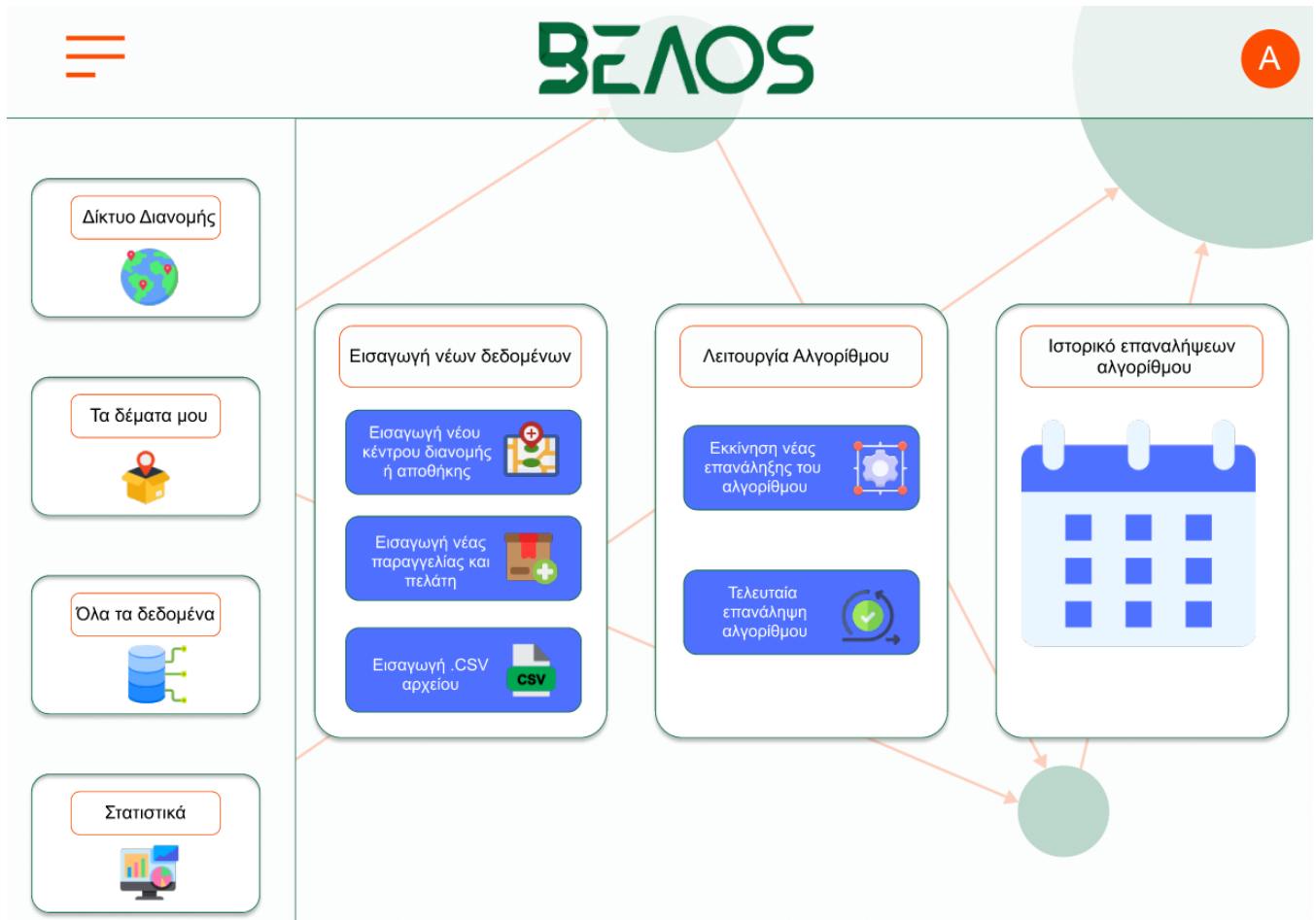
Σε αυτήν την ενότητα θα εξηγήσουμε τις διαφορετικές οθόνες που δημιουργήθηκαν για το έργο. Έχοντας υπ όψιν την εταιρική φύση του συστήματος η πρώτη οθόνη που δημιουργήθηκε ήταν αυτή της εισαγωγής ενός χρήστη στο σύστημα.



Εικόνα 6 – Οθόνη log in του συστήματος

Στην οθόνη όπως θα παρατηρήσετε δεν υπάρχει η επιλογή εγγραφής. Η δυνατότητα εγγραφής θα γίνεται άμεσα από τον διαχειριστή του συστήματος, όπως δηλαδή, συμβαίνει στις περισσότερες εταιρικές εφαρμογές. Όπως προαναφέραμε στην ενότητα για την βάση δεδομένων οι μόνοι χρήστες που θα έχουν πρόσβαση θα είναι οι υπάλληλοι γραφείου

διότι, οι οδηγοί θα έχουν εξειδικευμένο app για τις κινητές συσκευές. Μετά την διαδικασία της σύνδεσης οι χρήστες θα μπαίνουν στο κύριο μενού



Εικόνα 7 – Το κύριο μενού του συστήματος.

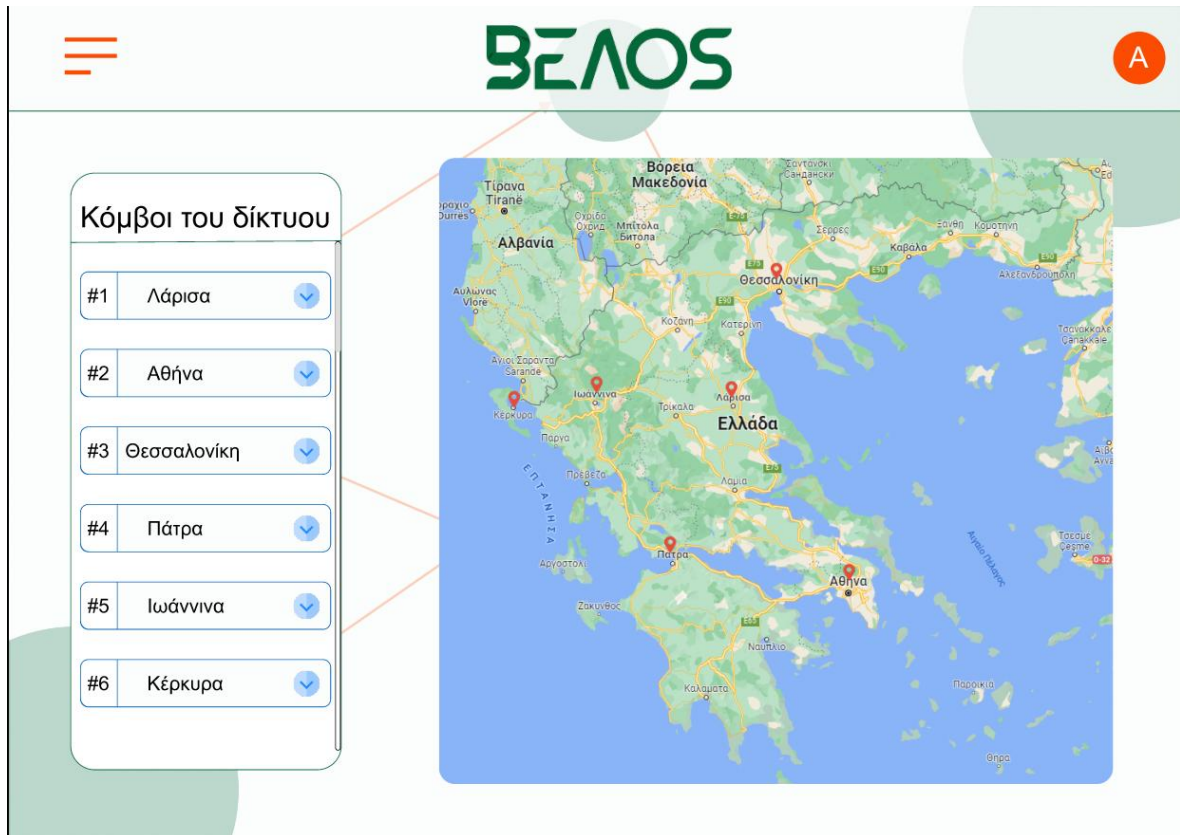
Στο κύριο μενού του συστήματος έχουμε δύο διακριτά τμήματα που εξυπηρετούν την πλοήγηση. Πρώτον έχουμε τις τρεις ομάδες επιλογών στο κέντρο της οθόνης, με τις τρεις βασικές λειτουργίες του συστήματος. Η πρώτη ομάδα επιλογών είναι η εισαγωγή δεδομένων στην βάση, εκεί έχουμε τα εξής: α) Εισαγωγή ενός κέντρου διανομής β) Εισαγωγή μιας νέας παραγγελίας και πελάτη. γ) Εισαγωγή .csv αρχείου που θα περιέχει όλα τα δεδομένα μαζί σε συγκεκριμένη μορφή. Η δεύτερη ομάδα επιλογών συμπεριλαμβάνει την λειτουργία του αλγορίθμου και την επισκόπηση των προηγούμενων

επαναλήψεων. Ενώ, η τρίτη ομάδα δίνει την επιλογή στον χρήστη να αναζητήσει όλες τις προηγούμενες εκτελέσεις του αλγορίθμου βάσει ημερομηνίας. Έπειτα, έχουμε το δευτερεύον μενού στο πλάι της οθόνης που έχει τις υπόλοιπες επιλογές. Στο κομμάτι αυτό έχουμε γενικής φύσεως επιλογές όπως: α) Η εμφάνιση του οδικού δικτύου που έχει δημιουργηθεί από τον χρήστη. β) Η οθόνη αναζήτησης και εποπτείας δεμάτων γ) Η οθόνη που μπορεί κανείς να βρει όλα τα δεδομένα συγκεντρωμένα δ) Η οθόνη που περιέχει τα στατιστικά δεδομένα που παράγονται από την βάση. Τέλος στο πάνω μέρος έχουμε το προφίλ του χρήστη που αν κανείς πατήσει πάνω έχουμε το εξής αποτέλεσμα:



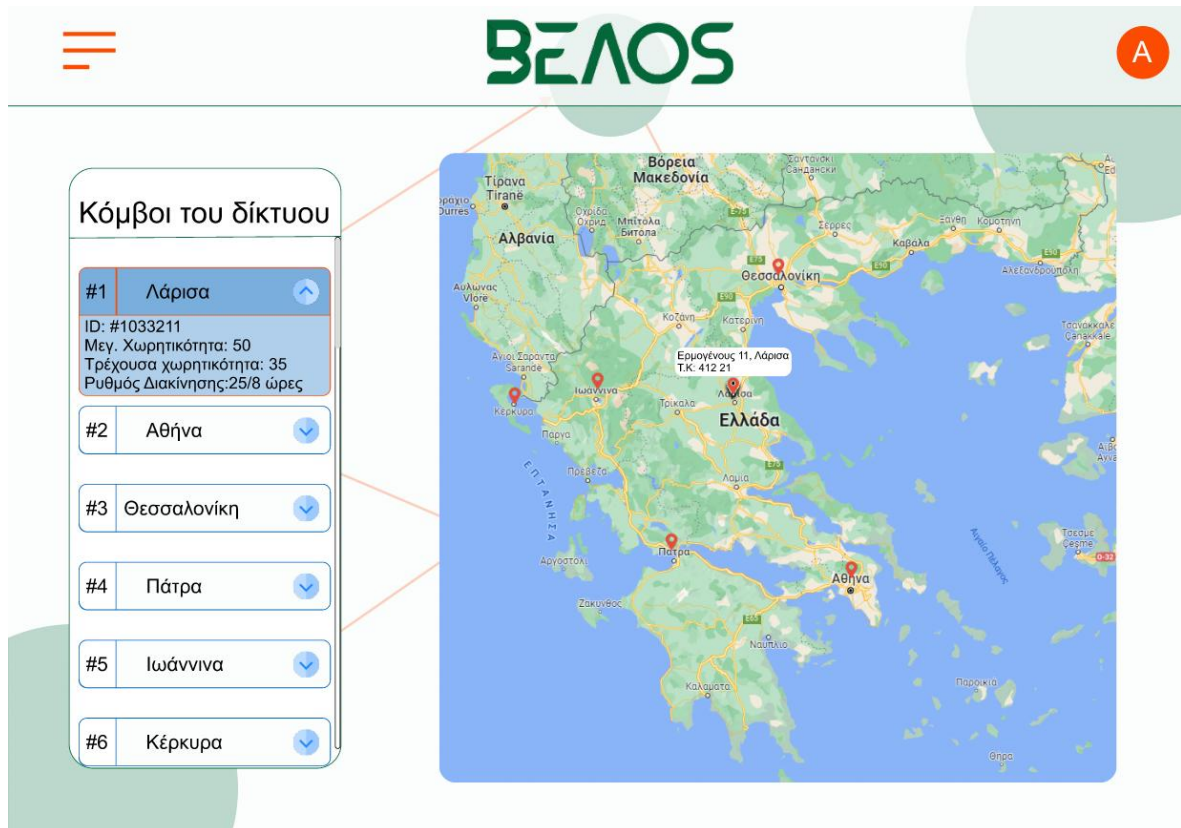
Εικόνα 8 – Οι λεπτομέρειες του χρήστη

Ξεκινώντας με το πλαϊνό μενού, πατάμε την πρώτη επιλογή και πηγαίνουμε στην παρακάτω οθόνη:



Εικόνα 9 – Οι κόμβοι του οδικού δικτύου

Στην παραπάνω οθόνη βλέπουμε όλους τους κόμβους του οδικού δικτύου. Δεξιά βλέπουμε το πλήθος αυτών ενώ, αριστερά βλέπουμε την διάταξη τους στον χάρτη με τις πορτοκαλί πινέζες. Επιλέγοντας έναν από τους κόμβους έχουμε το εξής:

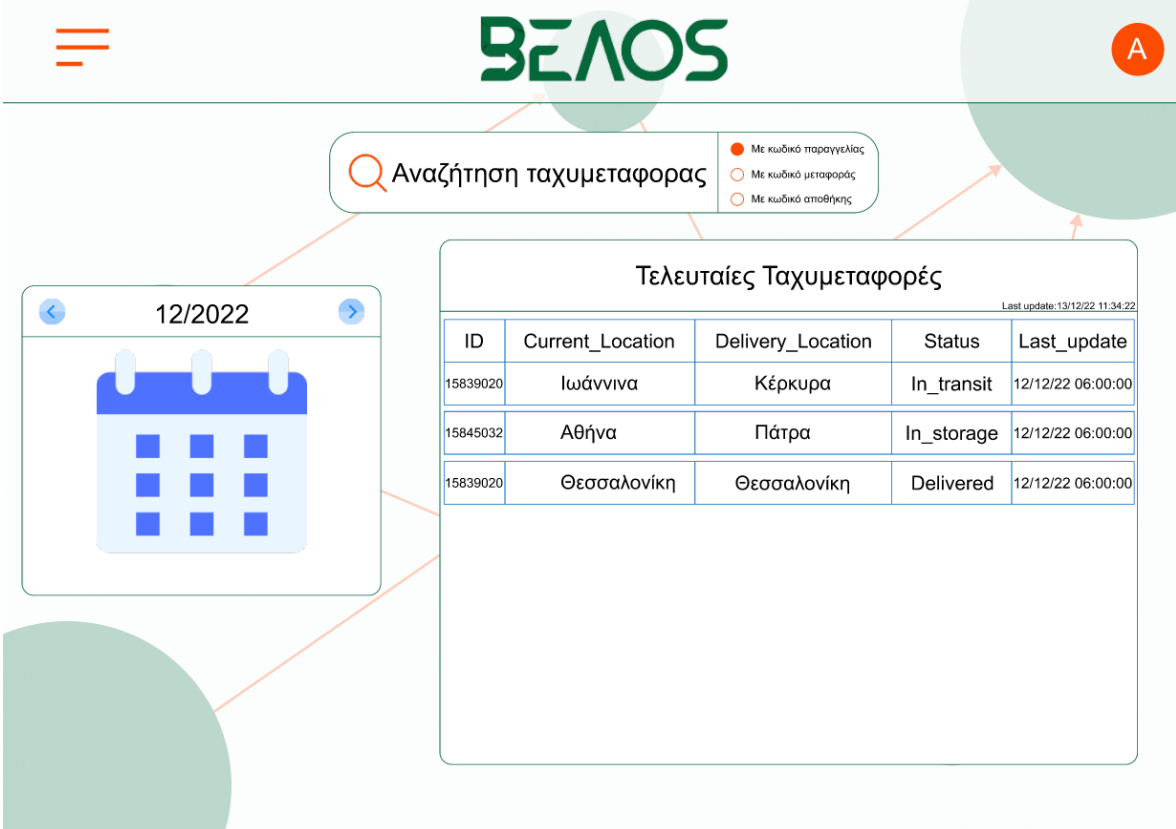


The screenshot shows the BELOS application interface. On the left, there is a sidebar titled "Κόμβοι του δικτύου" (Network Nodes) with a list of nodes: #1 Λάρισα (Larissa), #2 Αθήνα (Athens), #3 Θεσσαλονίκη (Thessaloniki), #4 Πάτρα (Patras), #5 Ιωάννινα (Ioannina), and #6 Κέρκυρα (Corfu). The selected node, #1 Λάρισα, is expanded to show details: ID: #1033211, Μέγ. Χωρητικότητα: 50 (Max Capacity: 50), Τρέχουσα χωρητικότητα: 35 (Current Capacity: 35), and Ρυθμός Διακίνησης: 25/8 ώρες (Throughput Rate: 25/8 hours). On the right, a map of Greece is displayed with a red pin indicating the location of Larissa. The map also shows other major cities and regions in Greece and neighboring countries like Albania and North Macedonia.

Εικόνα 10 – Η εμφάνιση λεπτομερειών του επιλεγμένου κόμβου

Όπως φαίνεται παραπάνω επιλέγοντας έναν κόμβο έχουμε περισσότερες πληροφορίες για αυτόν. Συγκεκριμένα έχουμε την μέγιστη χωρητικότητα του, την τρέχουσα χωρητικότητα και τον ρυθμό με τον οποίο επεξεργάζονται παραγγελίες. Ενώ στον χάρτη φαίνεται η διεύθυνση και ο ταχυδρομικός κώδικας αυτού.

Επιστρέφοντας στην αρχική οθόνη, συνεχίζουμε την δεύτερη επιλογή του πλαϊνού μενού, την αναζήτηση παραγγελίας.



ΒΕΛΟΣ

Αναζήτηση ταχυμεταφορας

- Με κωδικό παραγγελίας
- Με κωδικό μεταφοράς
- Με κωδικό αποθήκης

12/2022

Τελευταίες Ταχυμεταφορές

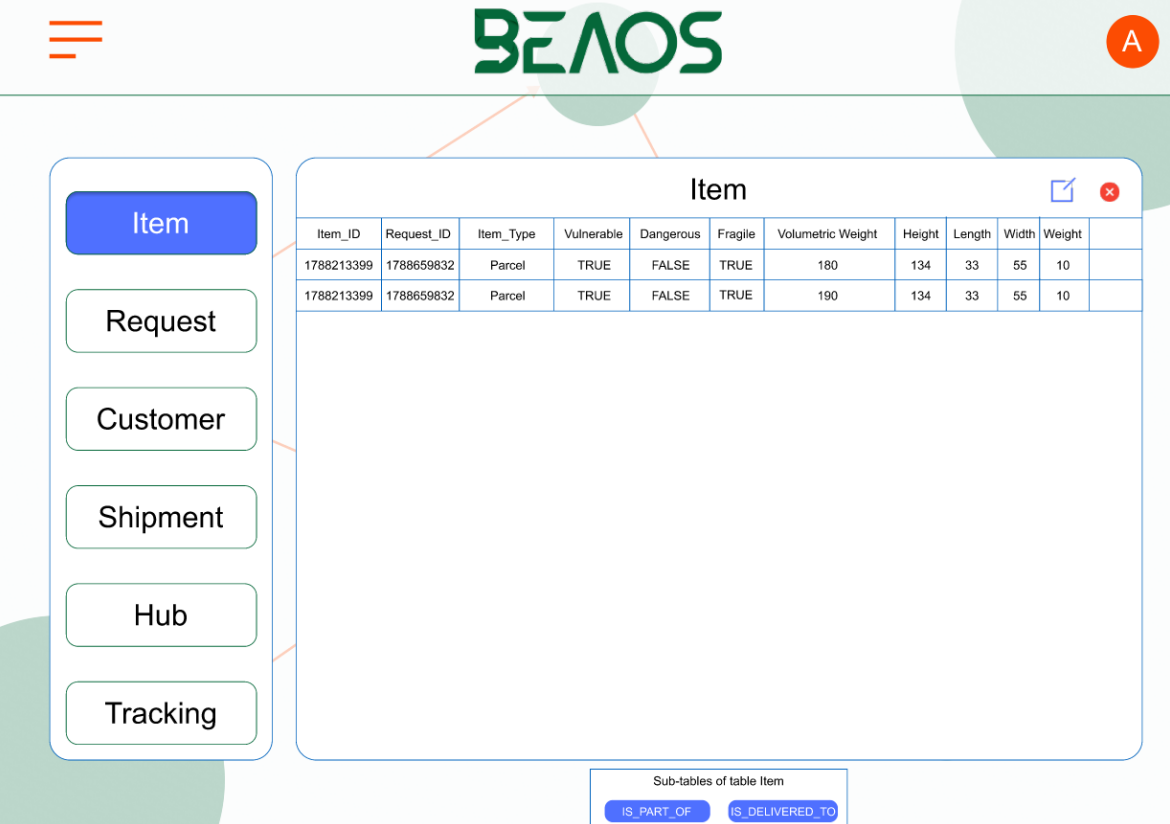
ID	Current_Location	Delivery_Location	Status	Last_update
15839020	Ιωάννινα	Κέρκυρα	In_transit	12/12/22 06:00:00
15845032	Αθήνα	Πάτρα	In_storage	12/12/22 06:00:00
15839020	Θεσσαλονίκη	Θεσσαλονίκη	Delivered	12/12/22 06:00:00

Last update: 13/12/22 11:34:22

Εικόνα 11 – Η οθόνη αναζήτησης παραγγελίας.

Η παραπάνω οθόνη μας δίνει όλες τις δυνατές πληροφορίες που μπορεί μία παραγγελία να έχει. Δηλαδή με βάση την ημερομηνία η οποία επιλεγεί από τον χρήστη μπορούμε να δούμε όλες τις παραγγελίες, ενεργές και μη, την συγκεκριμένη χρονική στιγμή. Ακόμα μπορούμε να αναζητήσουμε την μεταφορά με την οποία μετακινήθηκε η παραγγελία ή με την αποθήκη στην οποία σταμάτησε.

Έπειτα, πηγαίνοντας στην τρίτη επιλογή έχουμε τα δεδομένα της βάσης συγκεντρωμένα όπως φαίνεται παρακάτω.



**ΒΕΛΟΣ**

Item

Item_ID	Request_ID	Item_Type	Vulnerable	Dangerous	Fragile	Volumetric Weight	Height	Length	Width	Weight
1788213399	1788659832	Parcel	TRUE	FALSE	TRUE	180	134	33	55	10
1788213399	1788659832	Parcel	TRUE	FALSE	TRUE	190	134	33	55	10

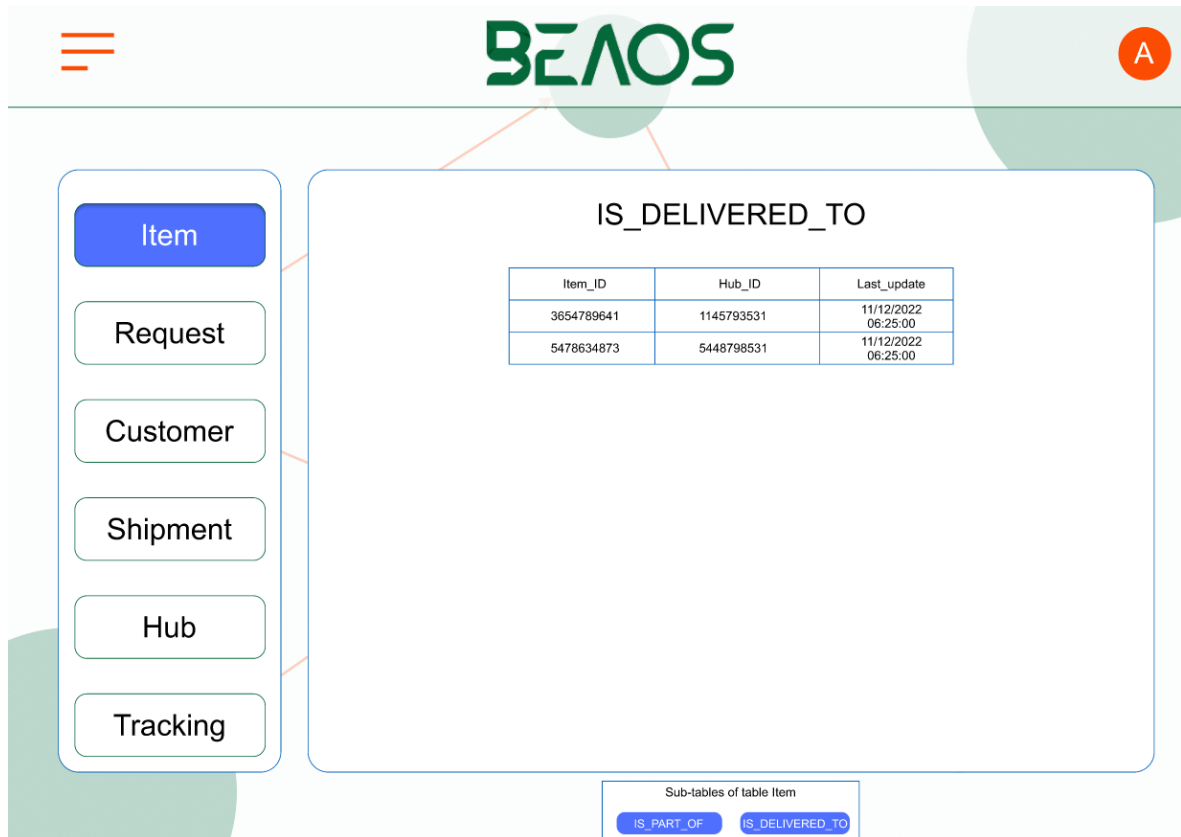
Sub-tables of table Item

IS\_PART\_OF IS\_DELIVERED\_TO

Εικόνα 12 – Οθόνη θέασης και επεξεργασίας δεδομένων

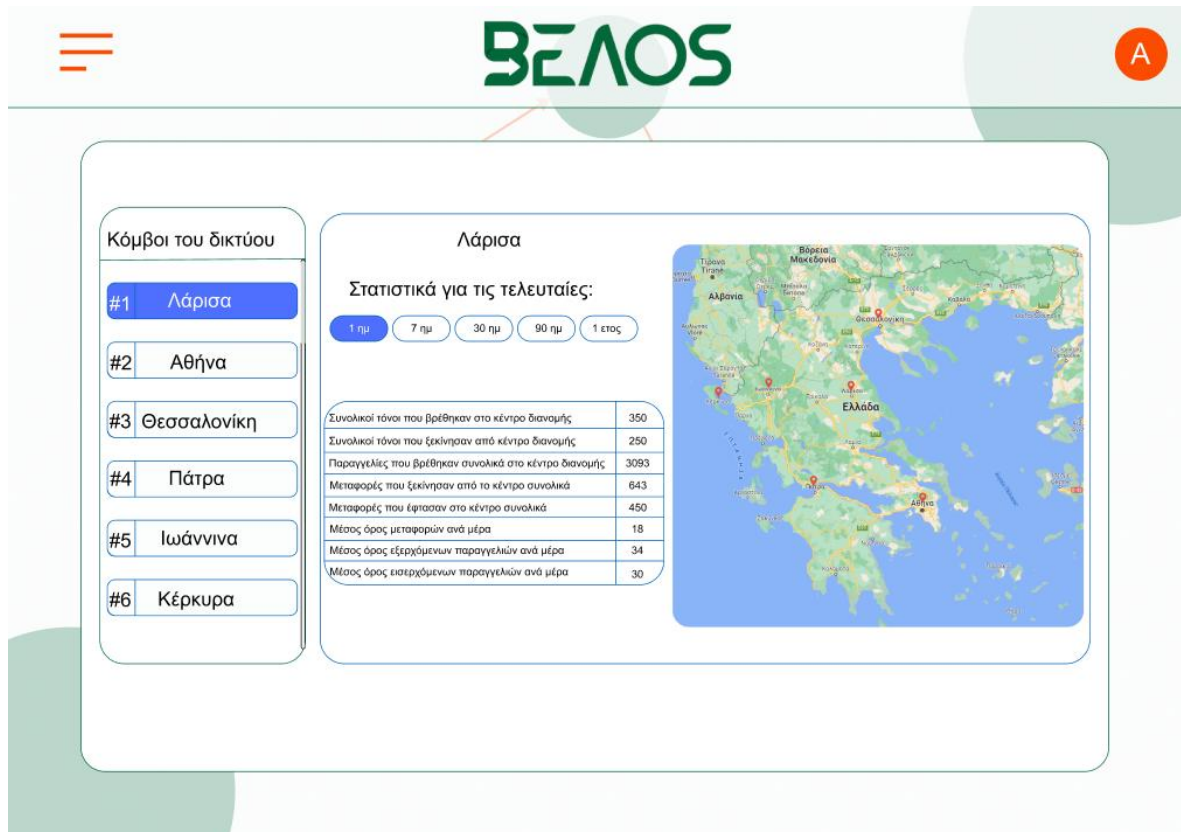
Στην παραπάνω οθόνη βλέπουμε τον πίνακα Item(πανω) και τους παραγόμενους πίνακες του(κάτω). Ενώ δίπλα από το Item έχουμε δύο επιλογές, την επεξεργασία ενός επιλεγμένου κελιού ή την διαγραφή του. Ωστόσο, πατώντας σε έναν από τους παραγόμενους πίνακες έχουμε το εξής αποτέλεσμα





Εικόνα 13 – Ο παραγόμενος πίνακας IS\_DELIVERED\_TO

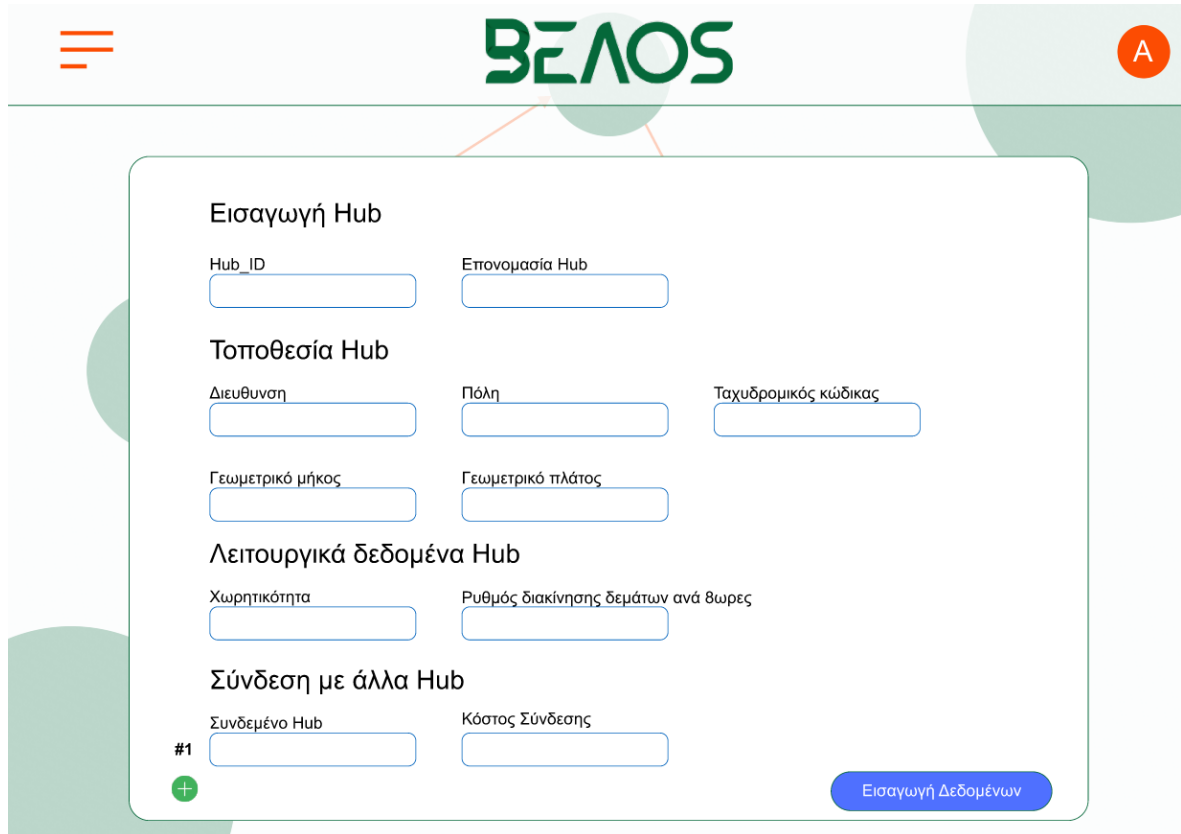
Όπως κανείς θα παρατηρήσει, ο παραγόμενος πίνακας δεν έχει την δυνατότητα επεξεργασίας, αυτό συμβαίνει διότι οι παραγόμενοι πίνακες συνήθως δημιουργούνται από εσωτερικές λειτουργίες του συστήματος, άγνωστες προς τον χρήστη. Συνάμα, εάν γίνει επεξεργασία του μοναδικού αριθμού αναγνώρισης ενός Item, για παράδειγμα, η βάση πρέπει να ανανεωθεί αυτόματα, αλλιώς θα υπάρχει ασυμφωνία μεταξύ των κελιών της βάσης. Τέλος έχουμε τα στατιστικά, επιλέξαμε να εστιάσουμε στους κόμβους του συστήματος για τα στατιστικά.



Εικόνα 14 – Οθόνη στατιστικών

Έχοντας κατάλληλα στατιστικά δεδομένα για έναν κόμβο μία επιχείρηση θα μπορούσε να πάρει διοικητικές αποφάσεις για την λειτουργία του. Δηλαδή να αυξήσει το προσωπικό για να έχει μεγαλύτερη ρυθμαπόδοση ή να επεκτείνει τον διαθέσιμο αποθηκευτικό χώρο διότι υπάρχει μεγάλη εισροή παραγγελιών. Αυτές τις πληροφορίες μπορεί κανείς να τις δει για κάθε έναν κόμβο του συστήματος σε βάθος χρόνου, από μία ημέρα έως ένα έτος

Έχοντας εξαντλήσει τις επιλογές στο πλαινό μενού, πηγαίνουμε στο κεντρικό, εκεί έχουμε τις τρεις ομάδες όπως προαναφέραμε. Ξεκινώντας με την εισαγωγή δεδομένων, έχουμε την οθόνη για την εισαγωγή ενός κέντρου διανομής όπως βλέπουμε παρακάτω



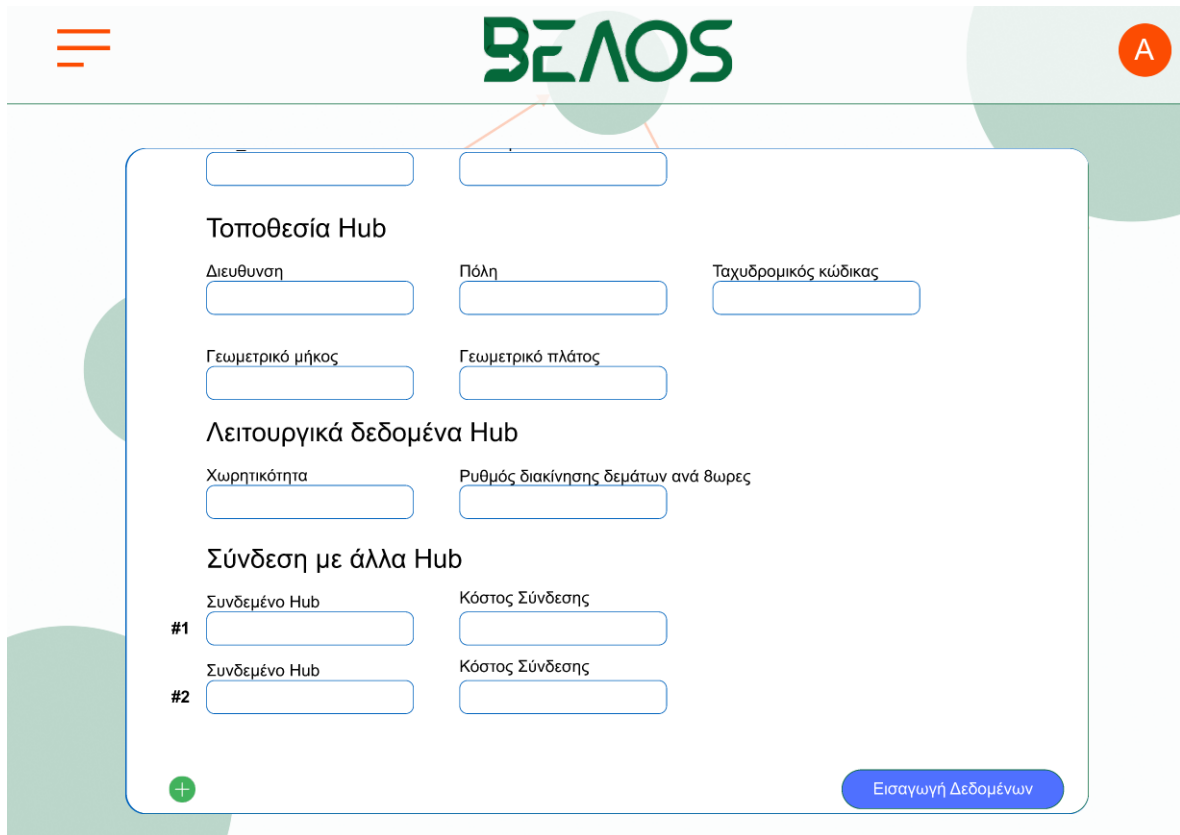
The screenshot shows the 'Εισαγωγή Hub' (Hub Entry) form in the BELOS system. The form is organized into several sections:

- Εισαγωγή Hub**: Contains two input fields: 'Hub\_ID' and 'Επωνομασία Hub'.
- Τοποθεσία Hub**: Contains three input fields: 'Διευθυνση', 'Πόλη', and 'Ταχυδρομικός κώδικας'.
- Γεωμετρικά μέγρος**: Contains two input fields: 'Γεωμετρικό μήκος' and 'Γεωμετρικό πλάτος'.
- Λειτουργικά δεδομένα Hub**: Contains two input fields: 'Χωρητικότητα' and 'Ρυθμός διακίνησης δεμάτων ανά θωρες'.
- Σύνδεση με άλλα Hub**: Contains two input fields: 'Συνδεμένο Hub' and 'Κόστος Σύνδεσης'. Below the 'Συνδεμένο Hub' field is a green plus sign icon and the label '#1'.

At the bottom right of the form is a blue button labeled 'Εισαγωγή Δεδομένων'.

Εικόνα 15 – Εισαγωγή HUB στην βάση δεδομένων

Στην παραπάνω οθόνη βλέπουμε μία απλή φόρμα εισαγωγής δεδομένων. Στο κάτω μέρος της φόρμας υπάρχει η δυνατότητα να συνδέσεις έναν κόμβο με έναν άλλο ή άλλους πατώντας το πράσινο “συν” στο τέλος. Με τα παρακάτω αποτελέσματα



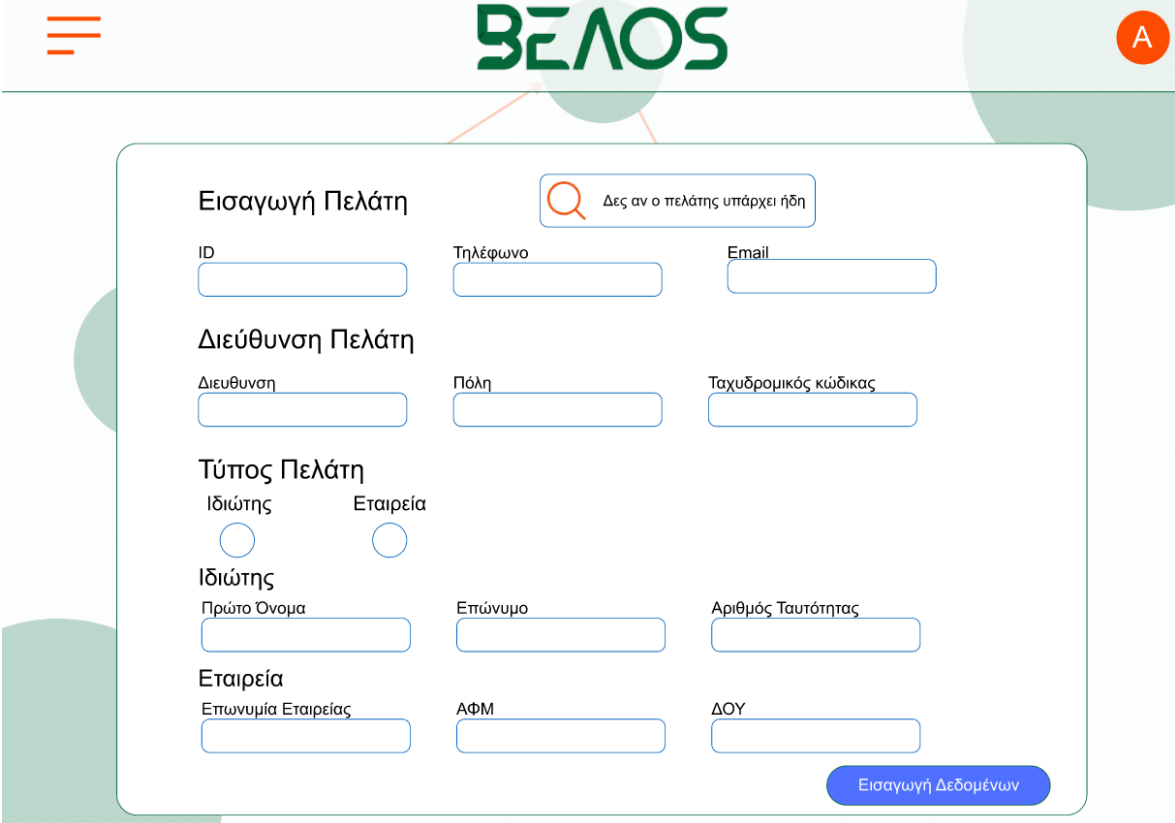
The screenshot shows the BELOS web application interface. At the top, there is a navigation menu (hamburger icon) and the BELOS logo. A red circle with the letter 'A' is in the top right corner. The main content area is a form titled 'Προσθήκη επιπλέον συνδέσεων στον κόμβο' (Add additional connections to the node). The form is divided into several sections:

- Τοποθεσία Hub** (Hub Location):
  - Διεύθυνση (Address):
  - Πόλη (City):
  - Ταχυδρομικός κώδικας (Postal code):
  - Γεωμετρικό μήκος (Geometric length):
  - Γεωμετρικό πλάτος (Geometric width):
- Λειτουργικά δεδομένα Hub** (Operational data of the Hub):
  - Χωρητικότητα (Capacity):
  - Ρυθμός διακίνησης δεμάτων ανά 8ωρες (Parcel handling rate per 8 hours):
- Σύνδεση με άλλα Hub** (Connection with other Hubs):
  - #1 Συνδεδεμένο Hub (Connected Hub #1):
  - #1 Κόστος Σύνδεσης (Connection cost #1):
  - #2 Συνδεδεμένο Hub (Connected Hub #2):
  - #2 Κόστος Σύνδεσης (Connection cost #2):

At the bottom left of the form is a green plus sign icon, and at the bottom right is a blue button labeled 'Εισαγωγή Δεδομένων' (Enter data).

Εικόνα 16 – Προσθήκη επιπλέον συνδέσεων στον κόμβο

Συνεχίζονται στην ίδια ομάδα επιλογών έχουμε την εισαγωγή μιας παραγγελίας και πελάτη.



The screenshot shows the 'ΒΕΛΟΣ' website interface for adding a customer. The form is titled 'Εισαγωγή Πελάτη' and includes a search bar with the text 'Δες αν ο πελάτης υπάρχει ήδη'. The form is divided into sections for 'Ιδιώτης' and 'Εταιρεία'.

**Εισαγωγή Πελάτη**

Δες αν ο πελάτης υπάρχει ήδη

ID  Τηλέφωνο  Email

**Διεύθυνση Πελάτη**

Διευθυνση  Πόλη  Ταχυδρομικός κώδικας

**Τύπος Πελάτη**

Ιδιώτης  Εταιρεία

**Ιδιώτης**

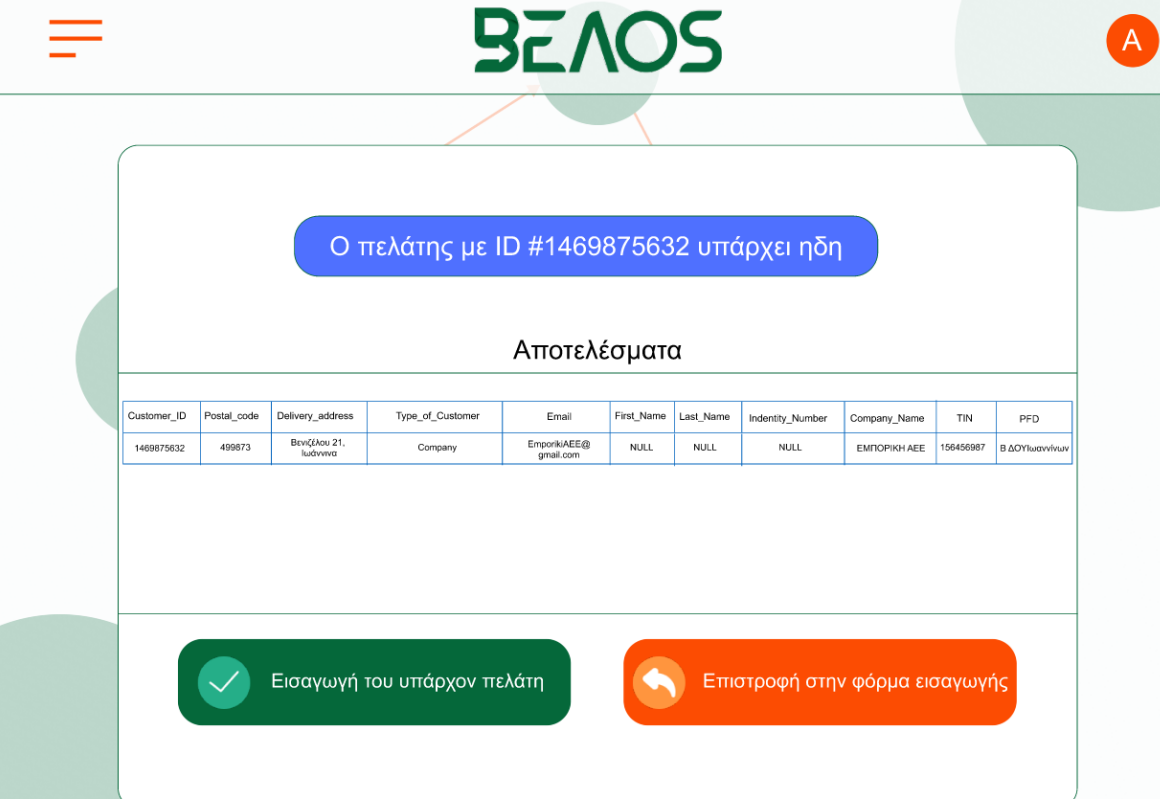
Πρώτο Όνομα  Επώνυμο  Αριθμός Ταυτότητας

**Εταιρεία**

Επωνυμία Εταιρείας  ΑΦΜ  ΔΟΥ

Εικόνα 17 – Προσθήκη πελάτη

Για την εισαγωγή μιας παραγγελίας πρέπει να προϋπάρχει ο πελάτης που την κάνει και την δέχεται. Παραπάνω βλέπουμε την φόρμα εισαγωγής ενός πελάτη, ωστόσο στο πάνω μέρος έχουμε την δυνατότητα αναζήτησης ενός πελάτη για την αποφυγή σφαλμάτων στην βάση καθώς δεν γίνεται να υπάρχει ένας πελάτης δύο φορές. Με την αναζήτηση του μοναδικού αναγνωριστικού του (ID) έχουμε το εξής αποτέλεσμα.



Ο πελάτης με ID #1469875632 υπάρχει ήδη

Αποτελέσματα

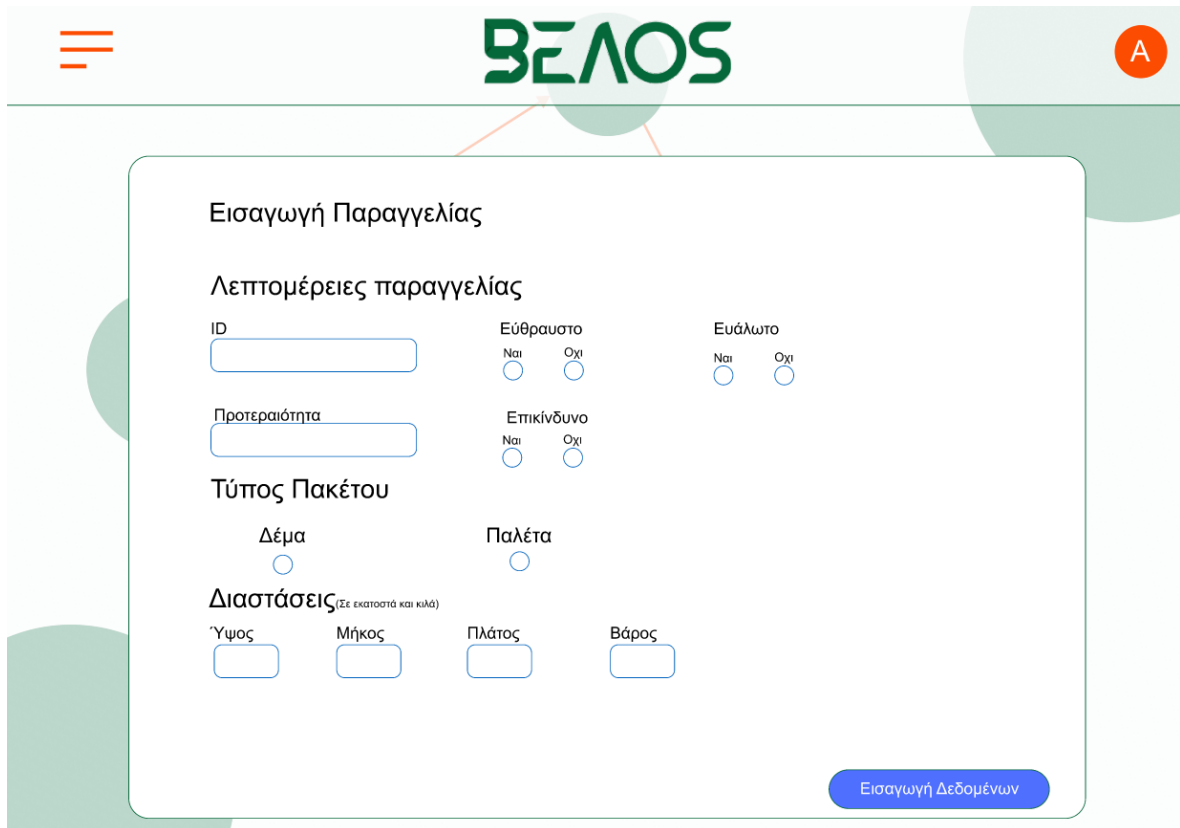
Customer_ID	Postal_code	Delivery_address	Type_of_Customer	Email	First_Name	Last_Name	Identity_Number	Company_Name	TIN	PFD
1469875632	499873	Βενιζέλου 21, Ιωάννινα	Company	EmponikiAEE@gmail.com	NULL	NULL	NULL	ΕΜΠΟΡΙΚΗ ΑΕΕ	156456987	Β.ΔΟΥλιανώνων

Εισαγωγή του υπάρχον πελάτη

Επιστροφή στην φόρμα εισαγωγής

Εικόνα 18 – Αποτέλεσμα ήδη υπάρχον πελάτη

Έχοντας βρει τον πελάτη μπορούμε να δούμε τα δεδομένα που έχει αποθηκευμένα στην βάση δεδομένων. Έπειτα έχουμε την επιλογή να συνεχίσουμε με τον υπάρχον πελάτη ή να επιστρέψουμε στην φόρμα και να εισάγουμε νέο πελάτη. Συνεχίζοντας με την πρώτη επιλογή πηγαίνουμε τελικά, στην εισαγωγή της νέας παραγγελίας όπως φαίνεται παρακάτω.



**Εισαγωγή Παραγγελίας**

**Λεπτομέρειες παραγγελίας**

ID

Εύθραστο  Ναι  Όχι

Ευάλωτο  Ναι  Όχι

Προτεραιότητα

Επικίνδυνο  Ναι  Όχι

**Τύπος Πακέτου**

Δέμα

Παλέτα

**Διαστάσεις** (Σε εκατοστά και κιλό)

Ύψος

Μήκος

Πλάτος

Βάρος

[Εισαγωγή Δεδομένων](#)

Εικόνα 19 – Εισαγωγή παραγγελίας

Τελειώνοντας με την εισαγωγή μπορούμε να πατήσουμε το κουμπί “Εισαγωγή δεδομένων”. Το σύστημα ελέγχει για την μορφή των δεδομένων και για το αν ταιριάζει με αυτή της βάσης. Έχοντας βάλει τα σωστά δεδομένα έχουμε το παρακάτω αποτέλεσμα

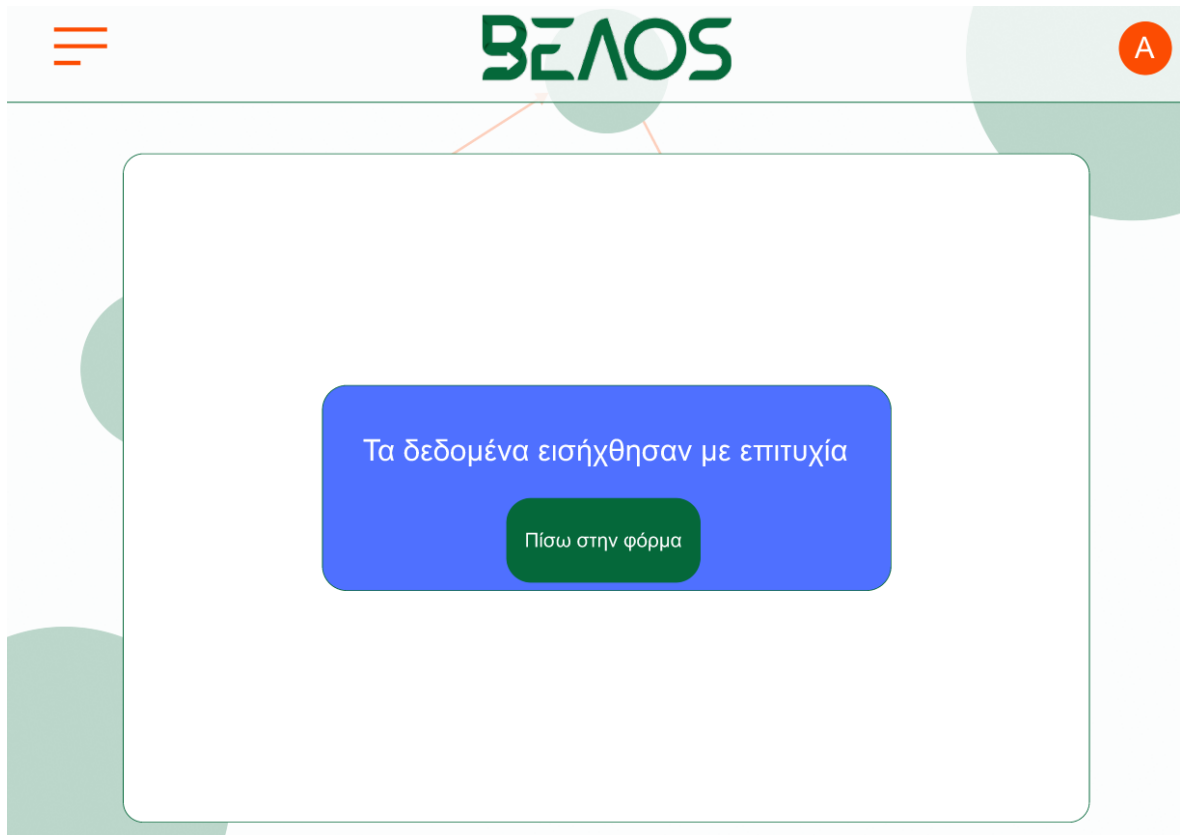


Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Ταμείο  
Περιφερειακής Ανάπτυξης

ΕΠΑνεΚ 2014-2020  
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ  
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ  
ΚΑΙΝΟΤΟΜΙΑ

ΕΣΠΑ  
2014-2020  
ανάπτυξη - εργασία - αλληλεγγύη

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Εικόνα 20 – Μήνυμα επιτυχίας με την εισαγωγή των δεδομένων



Η τελευταία επιλογή της πρώτης ομάδας του κύριου μενού είναι η εισαγωγή .csv αρχείου.  
Πατώντας το αντίστοιχο κουμπί έχουμε την επόμενη οθόνη



Εικόνα 21 – Οθόνη εισαγωγής .csv αρχείου

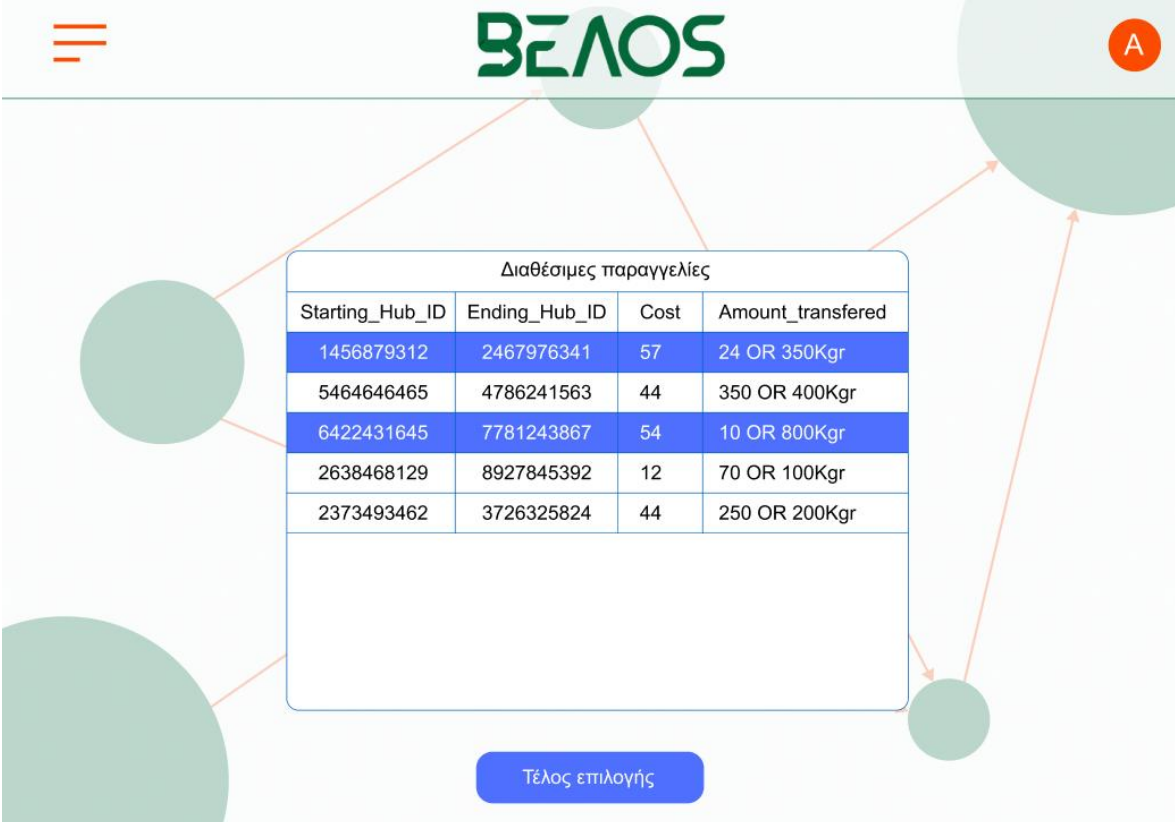
Συνεχίζοντας στην δεύτερη ομάδα έχουμε την λειτουργία του αλγορίθμου. Αρχικά έχουμε την επιλογή της εκτέλεσης του αλγορίθμου.



Εικόνα 22 – Οθόνη εκκίνησης αλγορίθμου

Στην παραπάνω οθόνη βλέπουμε τρεις επιλογές. Πρώτον, την επιλογή μεταφορών για την εκτέλεση του αλγορίθμου. Δεύτερον τον προγραμματισμό της επόμενης εκτέλεσης του αλγορίθμου και τέλος την εκκίνηση του ίδιου του αλγορίθμου.

Όπως φαίνεται παραπάνω η προεπιλογή για κάθε εκτέλεση του αλγορίθμου είναι όλες οι μεταφορές να λάβουν μέρος, όμως, αν θέλει κάποιος να επιλέξει συγκεκριμένες για την εκτέλεση μπορεί. Πατώντας το κουμπί της επιλογής έχουμε το εξής:



Διαθέσιμες παραγγελίες			
Starting_Hub_ID	Ending_Hub_ID	Cost	Amount_transferred
1456879312	2467976341	57	24 OR 350Kgr
5464646465	4786241563	44	350 OR 400Kgr
6422431645	7781243867	54	10 OR 800Kgr
2638468129	8927845392	12	70 OR 100Kgr
2373493462	3726325824	44	250 OR 200Kgr

Τέλος επιλογής

Εικόνα 23 – Επιλογή διαθέσιμων μεταφορών

Στην οθόνη αυτή θα βγαίνουν οι παραγγελίες που είναι διαθέσιμες, δηλαδή που δεν έχουν ήδη παραδοθεί και μπορούν να συμμετάσχουν στην εκτέλεση του αλγορίθμου. Με την επιλογή των επιθυμητών παραγγελιών μπορούμε να πατήσουμε του κάτω κουμπί το οποίο θα μας στείλει στην προηγούμενη οθόνη αλλά θα μας λέει πόσες παραγγελίες έχουμε επιλέξει όπως φαίνεται παρακάτω



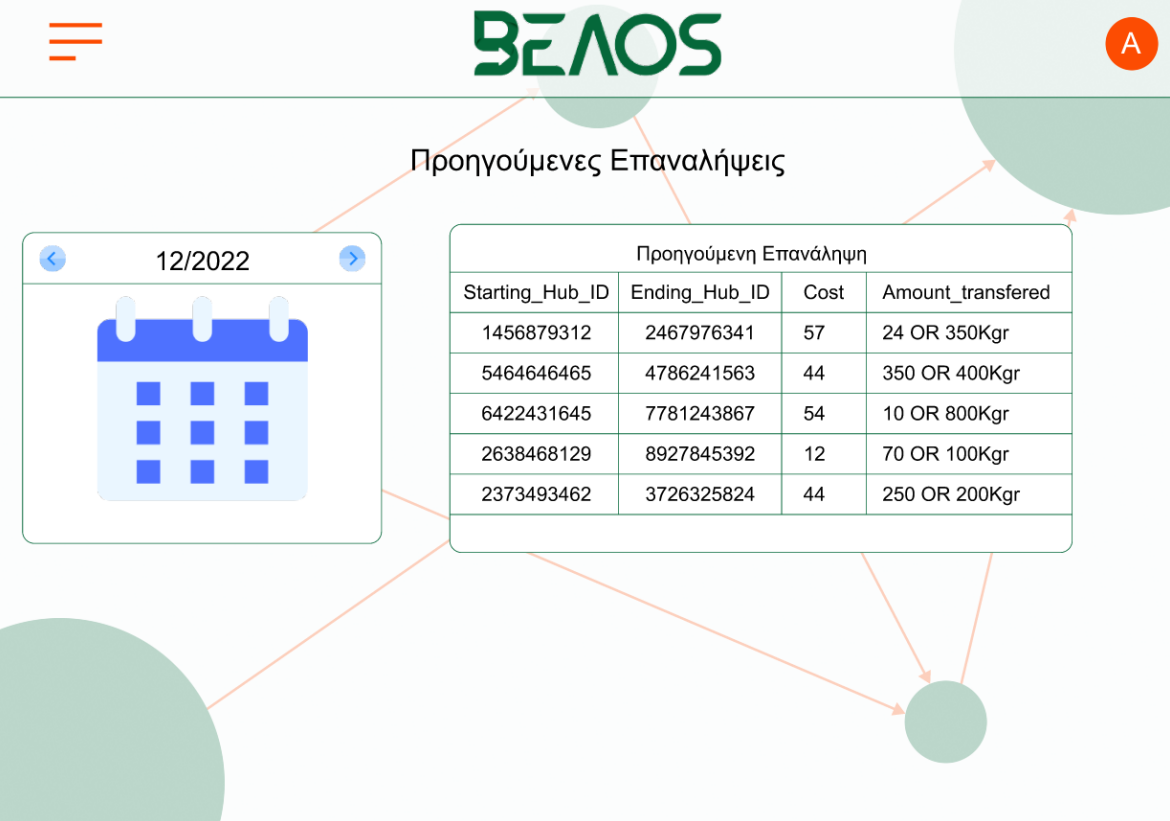
Εικόνα 24 – Οθόνη εκκίνησης αλγορίθμου με τις επιλεγμένες μεταφορές

Έχοντας επιλέξει τις μεταφορές που θέλουμε μπορούμε πλέον να πατήσουμε το κουμπί για την εκκίνηση του αλγορίθμου. Εάν ο αλγόριθμος κατάφερε να διαβάσει όλες τις παραγγελίες που του δώσαμε για να ξεκινήσει την λειτουργία του τότε έχουμε το παρακάτω μήνυμα από το σύστημα



Εικόνα 25 – Μήνυμα επιτυχίας από τον αλγόριθμο

Για να δούμε τα αποτελέσματα του αλγορίθμου μαζί με τα αποτελέσματα προηγούμενων επαναλήψεων του μπορούμε να πάμε στην τρίτη και τελευταία ομάδα επιλογών του κεντρικού μενού όπως φαίνεται παρακάτω.




**ΒΕΛΟΣ**

A

Προηγούμενες Επαναλήψεις

12/2022



Προηγούμενη Επανάληψη			
Starting_Hub_ID	Ending_Hub_ID	Cost	Amount_transferred
1456879312	2467976341	57	24 OR 350Kgr
5464646465	4786241563	44	350 OR 400Kgr
6422431645	7781243867	54	10 OR 800Kgr
2638468129	8927845392	12	70 OR 100Kgr
2373493462	3726325824	44	250 OR 200Kgr

Εικόνα 26 – Οθόνη με τις προηγούμενες επαναλήψεις

Στην παραπάνω οθόνη μπορεί κανείς να δει τα αποτελέσματα της προηγούμενης επανάληψης αλλά και όλων των προηγούμενων επαναλήψεων ανά ημερομηνία. Στην δεξιά πλευρά θα υπάρχει ένα ημερολόγιο μέσα από το οποίο θα μπορεί κανείς να δει μία συγκεκριμένη μέρα τα αποτελέσματα της εκτέλεσης του αλγορίθμου

## Αναφορές

- [1] L. De Lauretis, "From Monolithic Architecture to Microservices Architecture," *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Berlin, Germany, 2019, pp. 93-96, doi: 10.1109/ISSREW.2019.00050.
- [2] F. Ponce, G. Márquez and H. Astudillo, "Migrating from monolithic architecture to microservices: A Rapid Review," *2019 38th International Conference of the Chilean Computer Science Society (SCCC)*, Concepcion, Chile, 2019, pp. 1-7, doi: 10.1109/SCCC49216.2019.8966423.
- [3] N. Alshuqayran, N. Ali and R. Evans, "A Systematic Mapping Study in Microservice Architecture," *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*, Macau, China, 2016, pp. 44-51, doi: 10.1109/SOCA.2016.15.
- [4] M. Amaral, J. Polo, D. Carrera, I. Mohomed, M. Unuvar and M. Steinder, "Performance Evaluation of Microservices Architectures Using Containers," *2015 IEEE 14th International Symposium on Network Computing and Applications*, Cambridge, MA, USA, 2015, pp. 27-34, doi: 10.1109/NCA.2015.49.
- [5] Fatima Samea, Farooque Azam, Muhammad Waseem Anwar, Mehreen Khan, and Muhammad Rashid. 2019. A UML Profile for Multi-Cloud Service Configuration (UMLPMSC) in Event-driven Serverless Applications. In *Proceedings of the 2019 8th International Conference on Software and Computer Applications (ICSCA '19)*. Association for Computing Machinery, New York, NY, USA, 431-435. <https://doi.org/10.1145/3316615.3316636>
- [6] D. Taibi, J. Spillner and K. Wawruch, "Serverless Computing-Where Are We Now, and Where Are We Heading?," in *IEEE Software*, vol. 38, no. 1, pp. 25-31, Jan.-Feb. 2021, doi: 10.1109/MS.2020.3028708.
- [7] W. He and L. D. Xu, "Integration of Distributed Enterprise Applications: A Survey," in *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 35-42, Feb. 2014, doi: 10.1109/TII.2012.2189221.
- [8] D. Jackson and G. Clynch, "An Investigation of the Impact of Language Runtime on the Performance and Cost of Serverless Functions," *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, Zurich,

Switzerland, 2018, pp. 154-160, doi: 10.1109/UCC-Companion.2018.00050.

- [9] M. Mahjoub, A. Mdhaffar, R. B. Halima and M. Jmaiel, "A Comparative Study of the Current Cloud Computing Technologies and Offers," 2011 First International Symposium on Network Cloud Computing and Applications, Toulouse, France, 2011, pp. 131-134, doi: 10.1109/NCCA.2011.28.
- [10] J. Gibson, R. Rondeau, D. Eveleigh and Q. Tan, "Benefits and challenges of three cloud computing service models," 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN), Sao Carlos, Brazil, 2012, pp. 198-205, doi: 10.1109/CASoN.2012.6412402.