

Περιεχόμενα

1	Υλοποίηση εφαρμογής	1
1.1	Σχεδίαση	1
1.2	Η εφαρμογή	3
1.3	Δομή της εφαρμογής	13
1.3.1	Module Administrator	13
1.3.1.1	Domain model του Administrator	13
1.3.1.2	Σελίδες του Administrator	13
1.3.1.3	Microflows του Administrator	17
1.3.2	Module TaskManager	21
1.3.2.1	Domain model του TaskManager	21
1.3.2.2	Σελίδες του TaskManager	22
1.3.2.3	Microflows του TaskManager	25
1.3.2.4	Έγγραφα του TaskManager	25

Κεφάλαιο 1

Υλοποίηση εφαρμογής

Σε αυτό το κεφάλαιο...

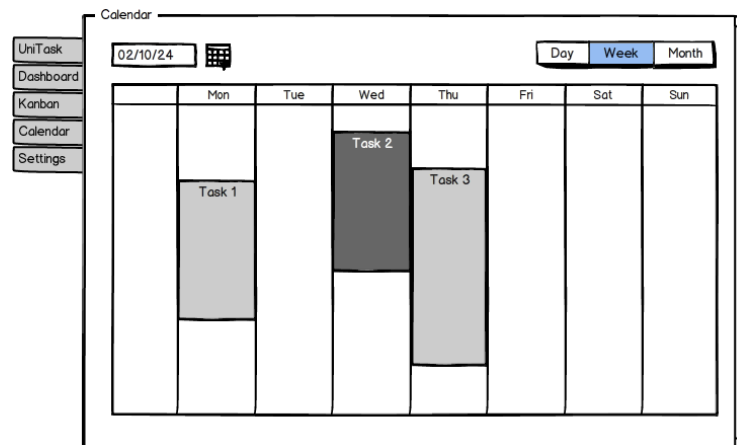
1.1 Σχεδίαση

Στην ενότητα ??, παρουσιάστηκε μια έρευνα με τα βασικά χαρακτηριστικά που θεωρήθηκαν απαραίτητα από τους φοιτητές για μια εφαρμογή τους. Λαμβάνοντας υπόψιν τις προτιμήσεις αυτές δόθηκε βάση στην υλοποίηση τους ώστε η εφαρμογή να ανταποκρίνεται στις ανάγκες της ακαδημαϊκής κοινότητας.

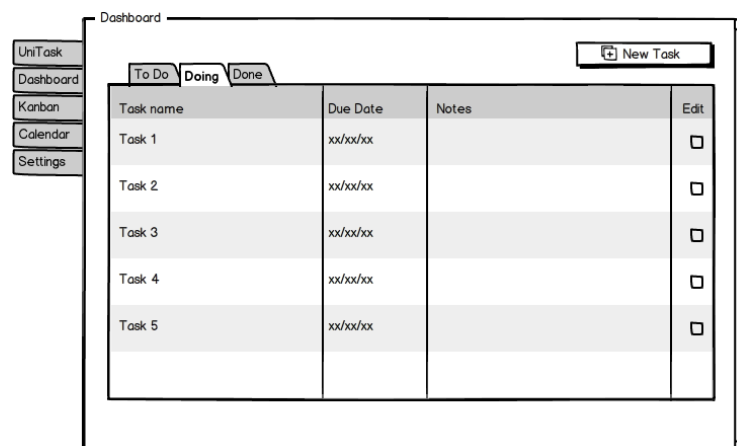
Αρχικά, ως μια εφαρμογή διαχείρισης εργασιών, θα περιλαμβάνει προφανώς ένα σύστημα δημιουργίας, τροποποίησης και διαγραφής εργασιών, καθορισμού του χρόνου έναρξης και λήξεώς τους, και η κατηγοροποίηση των εργασιών σε κατηγορίες ανάλογα με το αν έχουν πραγματοποιηθεί, αν πραγματοποιούνται και αν έχουν σκοπό να πραγματοποιηθούν μελλοντικά. Επιπλέον, με βάση την έρευνα, είναι σημαντική η ενσωμάτωση ενός ημερολογίου, η δυνατότητα χρωματικής ταξινόμησης (color-coding) και η υλοποίηση ενός συστήματος ανταμοιβής για την ενίσχυση της παρακίνησης των χρηστών. Επίσης, θα ήταν εξίσου σημαντική η δημιουργία ενός Kanban πίνακα για την άμεση οπτικοποίηση των εργασιών και την ευκολότερη διαχείρισή τους.

Σχεδιαστικά θεωρείται σημαντική η τήρηση σύγχρονων σχεδιαστικών κανόνων με ένα καθαρό interface και συνοχή στον σχεδιασμό για τη δημιουργία μιας λειτουργικής, αισθητικά ευχάριστης και ευκολόχρηστης εμπειρίας χρήστη ώστε να εξασφαλιστεί ότι η εφαρμογή μπορεί να ανταποκριθεί στις ανάγκες διαφορετικών τύπων χρηστών, αλλά και να παρουσιαστεί ως ένα προϊόν έτοιμο για χρήση σε πραγματικά περιβάλλοντα.

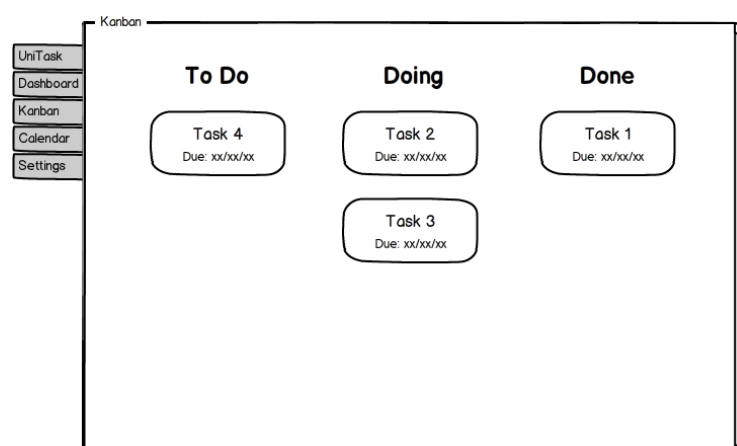
Στα σχήματα 1.1, 1.2 και 1.3 παρουσιάζονται κάποια αρχικά mockups που χρησιμοποιήθηκαν για τον σχεδιασμό της εφαρμογής.



Σχήμα 1.1: Mockup Calendar σελίδας



Σχήμα 1.2: Mockup Dashboard σελίδας

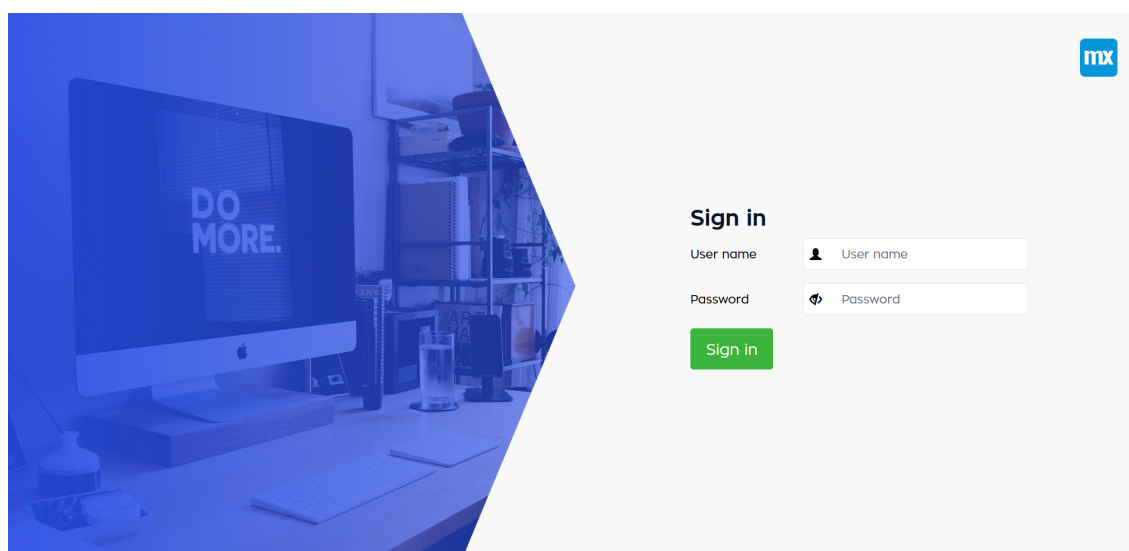


Σχήμα 1.3: Mockup Kanban σελίδας

1.2 Η εφαρμογή

Κατά την εκτέλεση της εφαρμογής, είτε τοπικά είτε μέσω της απομακρυσμένης πρόσβασης στο cloud, στη διεύθυνση <https://unitask-sandbox.mxapps.io/>, εμφανίζεται αρχικά η **σελίδα σύνδεσης**, όπως φαίνεται στο σχήμα 1.4. Στη συγκεκριμένη σελίδα, οι χρήστες καλούνται να εισάγουν τα στοιχεία σύνδεσής τους για να αποκτήσουν πρόσβαση στις λειτουργίες της εφαρμογής.

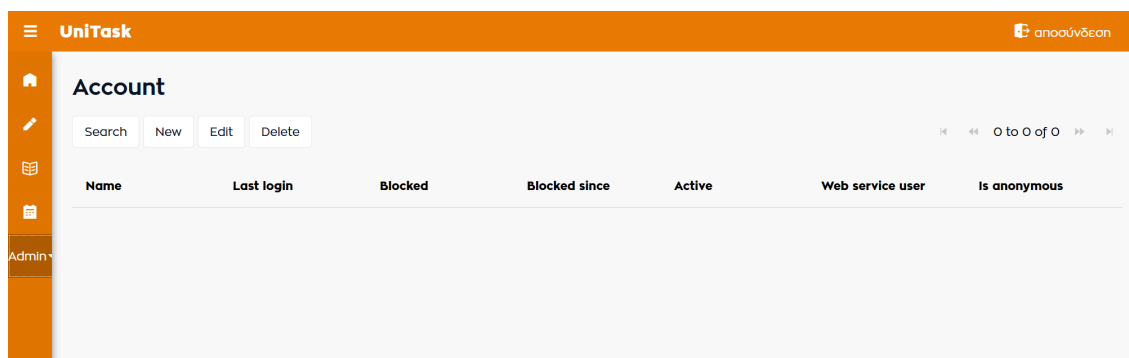
Ανάλογα με τα στοιχεία σύνδεσης που παρέχονται, το σύστημα διακρίνει δύο επίπεδα πρόσβασης: δικαιώματα διαχειριστή (Administrator) και δικαιώματα χρήστη (User), όπου εκπροσωπούν τους φοιτητές. Οι χρήστες με δικαιώματα Administrator έχουν πρόσβαση σε επιπλέον προηγμένες λειτουργίες διαχείρισης όπως η διαχείριση των χρηστών.



Σχήμα 1.4: Σελίδα σύνδεσης

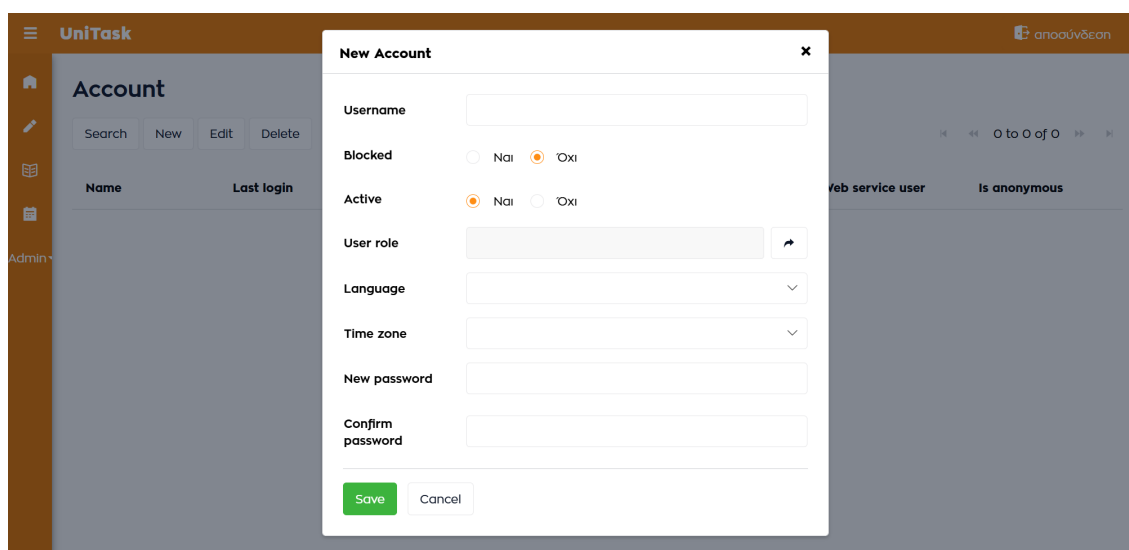
Αρχικά, πραγματοποιείται σύνδεση με τον λογαριασμό διαχειριστή (Administrator) προκειμένου να παρουσιαστούν οι λειτουργίες διαχείρισης χρηστών, συμπεριλαμβανομένης της δυνατότητας προσθήκης νέου χρήστη. Μετά την επιτυχή εισαγωγή των διαπιστευτηρίων του διαχειριστή, τα οποία έχουν οριστεί προκαταβολικά κατά την ανάπτυξη της εφαρμογής (βλ. ενότητα 1.3), εμφανίζεται η **σελίδα διαχείρισης χρηστών**, όπως απεικονίζεται στο σχήμα 1.5. Η σελίδα αυτή παρέχει στους διαχειριστές μια ολοκληρωμένη επισκόπηση της λίστας χρηστών της εφαρμογής, καθώς και εργαλεία για τη διαχείρισή τους. Το layout της σελίδας αποτελείται από μια κάθετη μπάρα μενού η οποία περιλαμβάνει τις ίδιες δυνατότητες με τους απλούς χρήστες η οποίες θα αναλυθούν στη συνέχεια.

Πατώντας στο κουμπί New, εμφανίζεται η popup σελίδα του σχήματος 1.6 με μια **φόρμα για την προσθήκη νέου χρήστη**. Η φόρμα περιλαμβάνει πεδία για την εισαγωγή του ονόματος χρήστη (Username), του ρόλου του χρήστη (User role) όπου



Σχήμα 1.5: Σελίδα διαχείρισης χρηστών

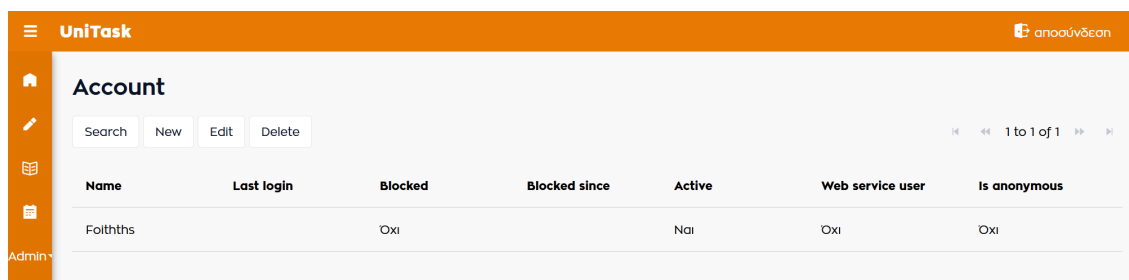
επιλέγεται αν πρόκειται για προσθήκη διαχειριστή ή χρήστη, του κωδικού πρόσβασης (New password και Confirm password). Λόγω του ότι ο χρήστης User έχει κληρονομήσει γνωρίσματα από την κλάση System.User του Mendix, έχουν προστεθεί πεδία όπως το Blocked, η οποία γίνεται αληθής μετά από κάποιες αποτυχημένες προσπάθειες σύνδεσης, το Active που γίνεται αληθές όταν ο χρήστης συνδεθεί, το Time zone όπου ορίζεται η ζώνη ώρας του χρήστη και το Language όπου ορίζεται η γλώσσα του χρήστη.



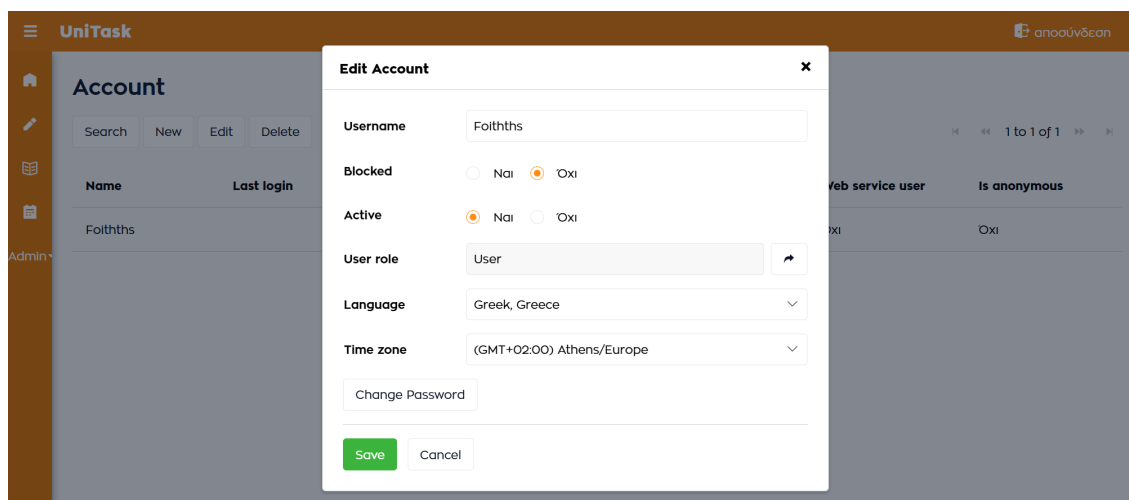
Σχήμα 1.6: Φόρμα προσθήκης νέου χρήστη

Δημιουργούμε έναν χρήστη με όνομα Foithths, ο οποίος εμφανίζεται στη λίστα των χρηστών (σχήμα 1.7). Πατώντας στο όνομά του, εμφανίζεται η σελίδα επεξεργασίας του χρήστη, όπως φαίνεται στο σχήμα 1.8. Στη λίστα των χρηστών υπάρχει η δυνατότητα αναζήτησης χρηστών βάσει όλων των στοιχείων τους (σχήμα 1.9), όπως επίσης και η δυνατότητα διαγραφής τους.

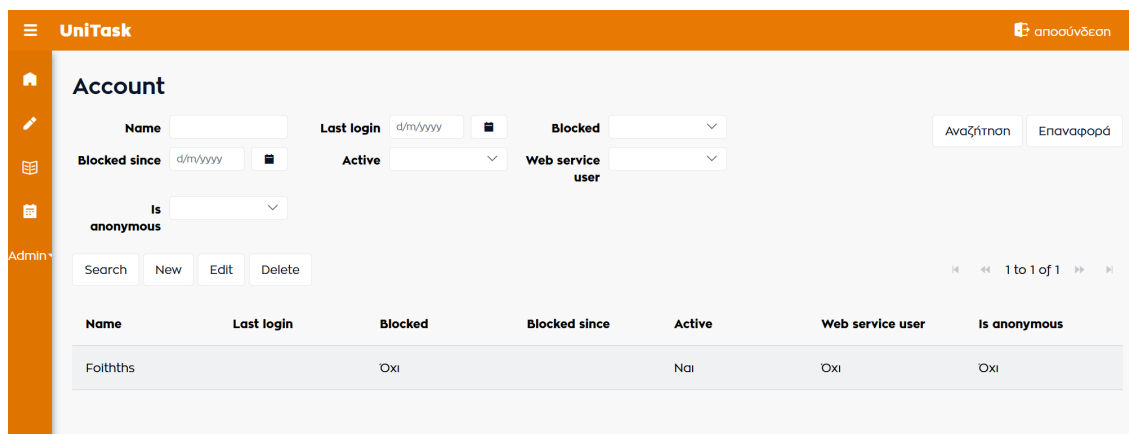
Αφού αποσυνδεθούμε και συνδεθούμε ως Foithths, μας υποδέχεται η αρχική



Σχήμα 1.7: Λίστα χρηστών με τον χρήστη Foithths



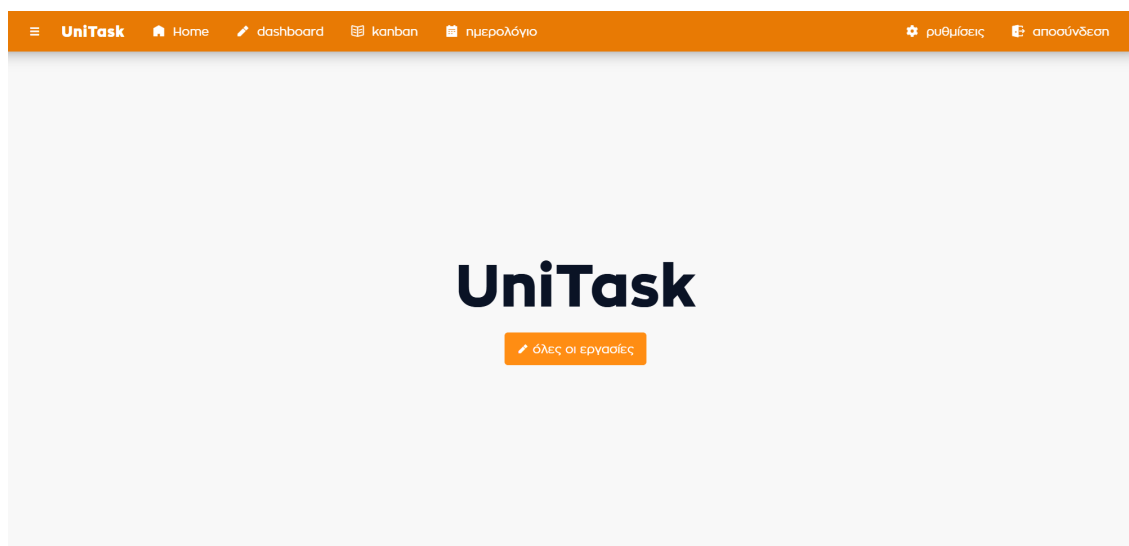
Σχήμα 1.8: Επεξεργασία στοιχείων χρήστη



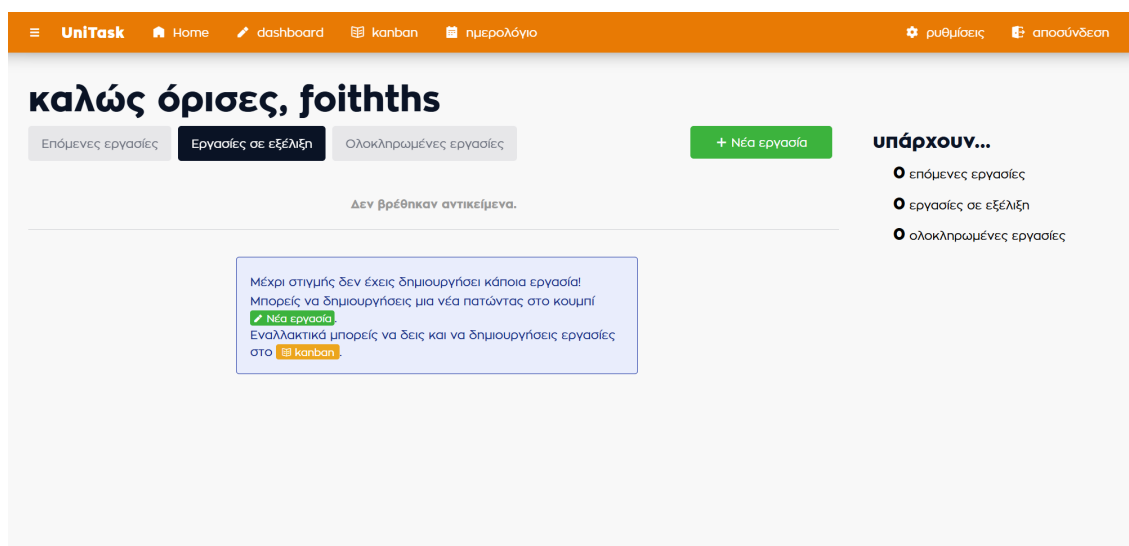
Σχήμα 1.9: Αναζήτηση χρήστη

σελίδα της εφαρμογής (σχήμα 1.10). Η σελίδα περιλαμβάνει ένα κεντρικό call to action κουμπί (όλες οι εργασίες) το οποίο οδηγεί στο dashboard.

Στο σχήμα 1.11 εμφανίζεται η σελίδα dashboard. Πρόκειται για την κεντρική



Σχήμα 1.10: Αρχική σελίδα εφαρμογής

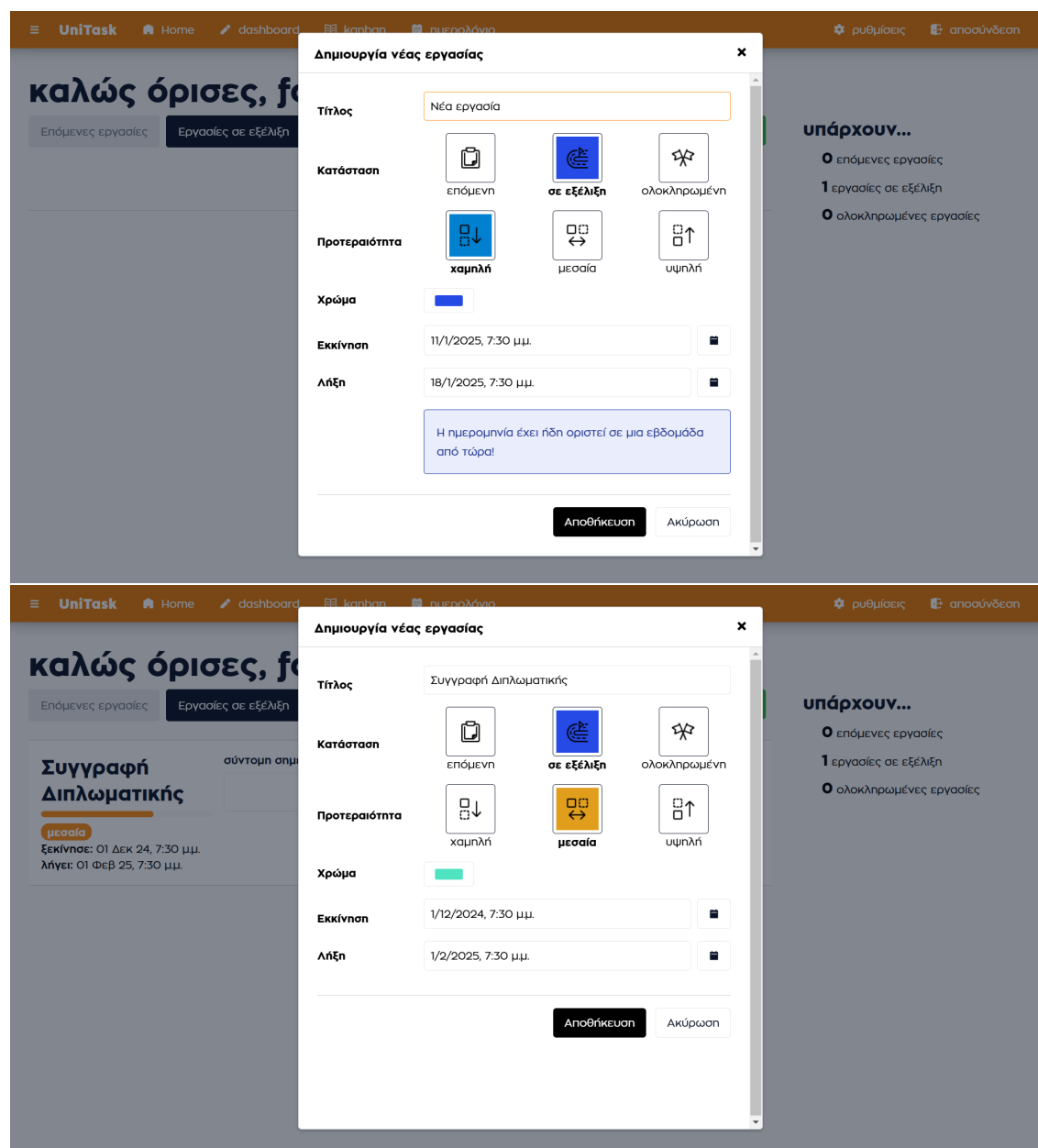


Σχήμα 1.11: Dashboard εργασιών

σελίδα προβολής, δημιουργίας και επεξεργασίας των εργασιών του χρήστη. Περιλαμβάνονται τρεις καρτέλες (Επόμενες εργασίες, Εργασίες σε εξέλιξη, Ολοκληρωμένες εργασίες). Οι επόμενες εργασίες αφορούν εργασίες που έχουν σκοπό να πραγματοποιηθούν στο άμεσο μέλλον αλλά όχι τη δεδομένη χρονική στιγμή, οι εργασίες σε εξέλιξη αφορούν εργασίες που βρίσκονται σε εξέλιξη και οι ολοκληρωμένες εργασίες αφορούν εργασίες που έχουν ολοκληρωθεί.

Στη σελίδα επίσης περιλαμβάνεται ένα επεξηγηματικό παράθυρο που εμφανίζεται μόνο όταν ο χρήστης δεν έχει δημιουργήσει κάποια εργασία και του εξηγεί το τρόπο λειτουργίας της εφαρμογής. Στο dashboard επίσης περιλαμβάνονται μετρητές για το

σύνολο των εργασιών που υπάρχουν ανά κατηγορία, όπως επίσης και ένα κεντρικό κουμπί δημιουργίας εργασιών (Νέα εργασία).



Σχήμα 1.12: Δημιουργία νέας εργασίας

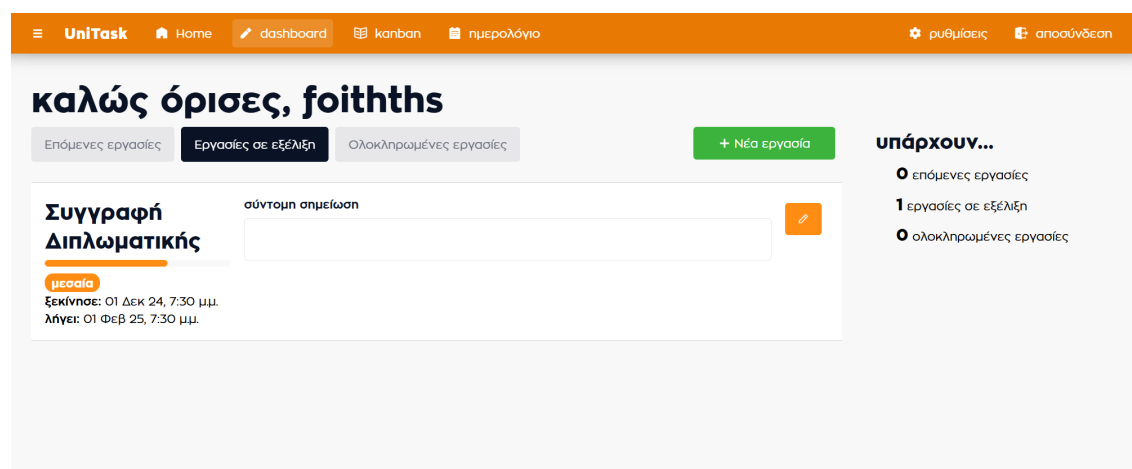
Πατώντας το, εμφανίζεται ένα popup παράθυρο (σχήμα 1.12.1) με μια φόρμα για τη δημιουργία μιας νέας εργασίας. Η φόρμα αυτή περιλαμβάνει πεδία για την εισαγωγή του τίτλου της εργασίας (Τίτλος), της ημερομηνίας έναρξης (Εκκίνηση) και λήξης (Λήξη), της κατάστασης της εργασίας (Κατάσταση), της προτεραιότητας της εργασίας (Προτεραιότητα) όπως επίσης και του χρώματος της εργασίας (Χρώμα), όπως θα αποτυπωθεί μετέπειτα στο ημερολόγιο. Στο popup παράθυρο ορίζεται προεπιλεγμέ-

να ως ημερομηνία λήξης της εργασίας μια εβδομάδα μετέπειτα από την ημερομηνία δημιουργίας της, ενώ η κατάσταση της εργασίας έχει προκαθοριστεί (γίνεται να τροποποιηθεί φυσικά) ανάλογα με το ποια καρτέλα ήταν ανοιχτή στο dashboard. Αφού επεξεργαστούμε το παράθυρο όπως επιθυμούμε (σχήμα 1.12.2), πατάμε αποθήκευση για την αποθήκευση της εργασίας.

Να σημειωθεί πως οι επιλογές που αφορούν την κατάσταση και την προτεραιότητα της εργασίας είναι χρωματικά ταξινομημένες (color-coded) με το δικό τους γραφικό σύμβολο, όπως φαίνεται στο σχήμα 1.13, με σκοπό να διευκολύνει τον χρήστη για την άμεση αναγνώριση τους.



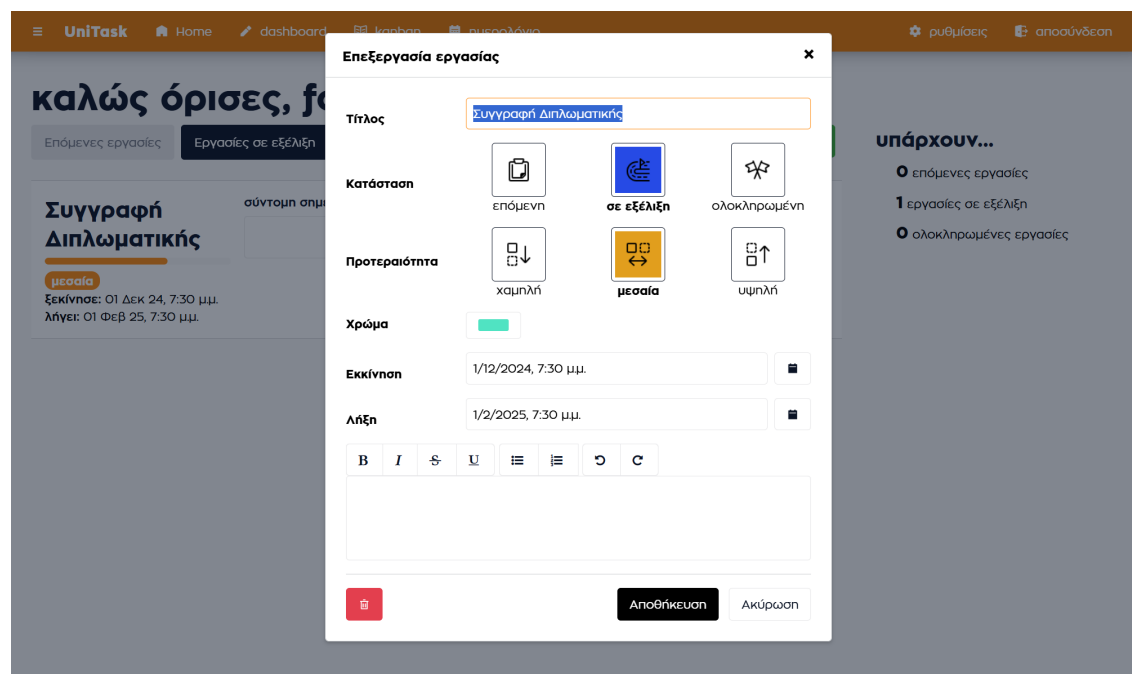
Σχήμα 1.13: Επιλογές κατάστασης και προτεραιότητας εργασίας



Σχήμα 1.14: Dashboard με δημιουργημένη εργασία

Στη σελίδα πλέον είναι δημιουργημένη η κάρτα με την πρώτη μας εργασία (σχήμα 1.14). Στο πλαίσιο σύντομη σημείωση μπορούμε να προσθέσουμε μια σύντομη περιγραφή της εργασίας, στα αριστερά εμφανίζεται ο τίτλος της εργασίας, μια μπάρα προόδου (progress bar) η οποία δυναμικά αυξάνεται όσο πλησιάζουμε στη λήξη της εργασίας, όπως επίσης η προτεραιότητα της εργασίας και οι ημερομηνίες και ώρες εκκίνησης και λήξης της εργασίας, ενώ στο δεξί μέρος της κάρτας εμφανίζονται το εικονίδιο για την επεξεργασία της εργασίας.

Ο τρόπος εμφάνισης των εργασιών στο dashboard λαμβάνει υπόψη του την προτεραιότητα τους και τον χρόνο λήξης τους. Έτσι μια εργασία με υψηλή προτεραιότητα θα εμφανίζεται ψηλότερα από μια εργασία με χαμηλή προτεραιότητα που δε λήγει σύντομα.



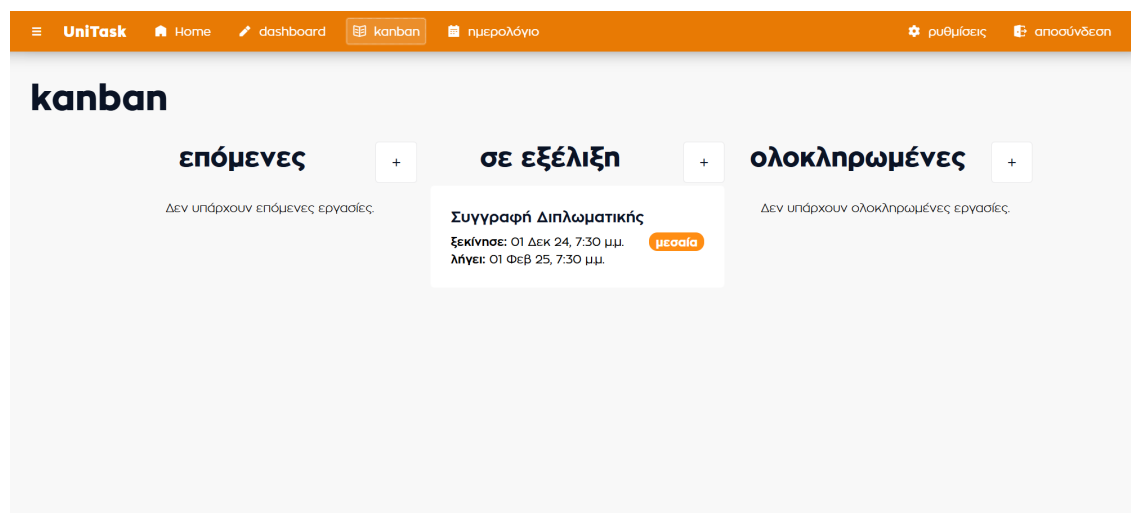
Σχήμα 1.15: Επεξεργασία εργασίας

Πατώντας στο εικονίδιο επεξεργασίας, εμφανίζεται το popup παράθυρο του σχήματος 1.15 όπου μπορούν να επεξεργαστούν τα στοιχεία της εργασίας μαζί με το πλαίσιο σύντομης σημείωσης. Το ίδιο παράθυρο περιλαμβάνει και το κουμπί διαγραφής της εργασίας.

Η σελίδα **Kanban** του σχήματος 1.16 εμφανίζει μια διαφορετική παρουσίαση στις εργασίες με τον τρόπο που έχει εξηγηθεί στην ενότητα ???. Στον πίνακα Kanban οι εργασίες χωρίζονται σε τρεις κατηγορίες: τις εργασίες που έχουν ολοκληρωθεί, τις εργασίες που βρίσκονται σε εξέλιξη και τις εργασίες που έχουν σκοπό να πραγματοποιηθούν στο μέλλον. Κάθε στήλη περιλαμβάνει ένα κουμπί δημιουργίας νέας εργασίας που αυτόματα καθορίζει και την κατηγορία της.

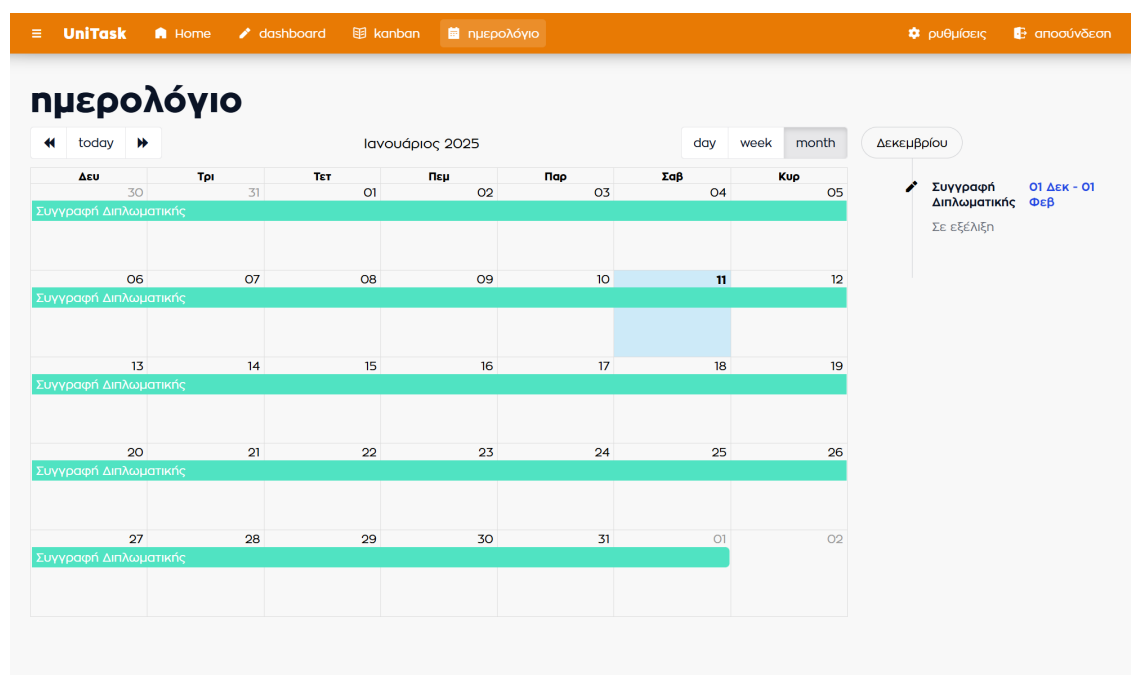
Οι εργασίες εμφανίζονται σε κάρτες με τον τίτλο της εργασίας, την προτεραιότητα της, την ημερομηνία έναρξης και λήξης της, ενώ πατώντας πάνω στην κάρτα μιας εργασίας, εμφανίζεται το popup παράθυρο επεξεργασίας της εργασίας του σχήματος 1.15.

Στη σελίδα **ημερολόγιο** (σχήμα 1.17) εμφανίζεται ένα ημερολόγιο με τις εργασίες του χρήστη. Οι εργασίες εμφανίζονται στο ημερολόγιο με το χρώμα που έχει οριστεί στην επιλογή του χρήστη κατά τη δημιουργία της εργασίας, υπάρχουν προβολές ανά ημέρα, εβδομάδα ή μήνα ενώ εμφανίζεται και ένα σύντομο παράρτημα στα δεξιά με



Σχήμα 1.16: Σελίδα Kanban

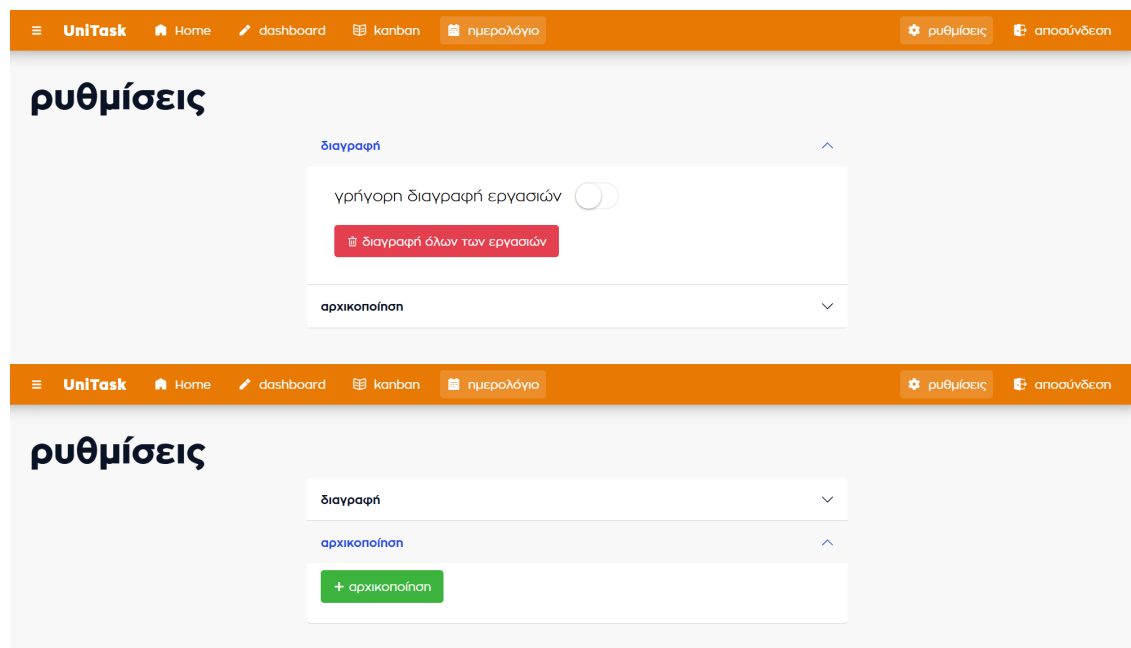
την λίστα των εργασιών.



Σχήμα 1.17: Σελίδα Calendar

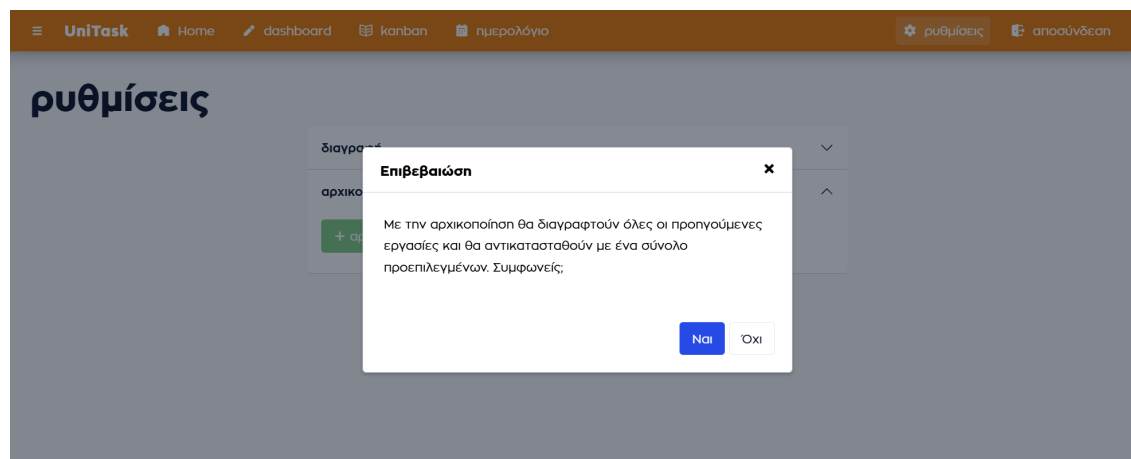
Τέλος, στη σελίδα **ρυθμίσεις** (σχήμα 1.18) μπορούμε με ένα κουμπί να διαγράψουμε όλες τις εργασίες μας, ή να επιλέξουμε τη λειτουργία γρήγορης διαγραφής εργασιών. Αυτή η λειτουργία εισάγει ένα κουμπί διαγραφής στις κάρτες εργασιών στο dashboard, επιτρέποντας την άμεση διαγραφή των εργασιών που θέλουμε χωρίς να χρειάζεται να μεταβούμε στη σελίδα επεξεργασίας κάθε εργασίας (σχήμα 1.20.1).

Τέλος, παρέχεται η δυνατότητα αρχικοποίησης των εργασιών. Στην αρχικοποίηση διαγράφονται οι υπάρχουσες εργασίες του χρήστη και δημιουργούνται κάποιες προκαθορισμένες οι οποίες λειτουργούν ως ένα demo (σχήμα 1.20).



Σχήμα 1.18: Σελίδα ρυθμίσεων

Σε κάθε ρύθμιση εμφανίζονται επιβεβαιωτικά popout μηνύματα προκειμένου να αποφευχθούν ακούσιες διαγραφές εργασιών (σχήμα 1.19).



Σχήμα 1.19: Σελίδα ρυθμίσεων

The image displays three screenshots of the UniTask application interface, showing the Dashboard, Kanban, and Calendar views.

Dashboard View: The top navigation bar includes UniTask, Home, dashboard, kanban, ημερολόγιο, ρυθμίσεις, and αποσύνδεση. The main heading is "καλώς όρισες, foithths". Below it are tabs for "Επόμενες εργασίες", "Εργασίες σε εξέλιξη", and "Ολοκληρωμένες εργασίες", along with a "+ Νέα εργασία" button. The content area shows two project cards: "Πρότζεκτ Λειτουργικά Συστήματα" and "Πρότζεκτ Ανάκτηση", each with a "μεσαία" status and start/end dates. A sidebar on the right lists "υπάρχουν..." tasks: 1 επόμενες εργασίες, 1 εργασίες σε εξέλιξη, and 2 ολοκληρωμένες εργασίες.

Kanban View: The top navigation bar is the same. The main heading is "kanban". Below it are three columns: "επόμενες", "σε εξέλιξη", and "ολοκληρωμένες", each with a "+" button. The "επόμενες" column contains "Εργασία 2 Μικροεπεξεργαστών" (status: χαμηλή). The "σε εξέλιξη" column contains "Δημιουργία Gantt Chart" (status: υψηλή). The "ολοκληρωμένες" column contains two project cards: "Πρότζεκτ Λειτουργικά Συστήματα" and "Πρότζεκτ Ανάκτηση", both with "μεσαία" status.

Calendar View: The top navigation bar is the same. The main heading is "ημερολόγιο". Below it are tabs for "today", "Δεκέμβριος 2024", "day", "week", and "month". The calendar grid shows tasks for December 2024. The right sidebar shows a list of tasks for October and December, including "Πρότζεκτ Λειτουργικά Συστήματα", "Πρότζεκτ Ανάκτηση", "Δημιουργία Gantt Chart", and "Εργασία 2 Μικροεπεξεργαστών", with their respective dates and statuses.

Σχήμα 1.20: Οι σελίδες Dashboard, Kanban και Calendar μετά την αρχικοποίηση των εργασιών. Στο Dashboard φαίνεται η λειτουργία γρήγορης διαγραφής εργασιών.

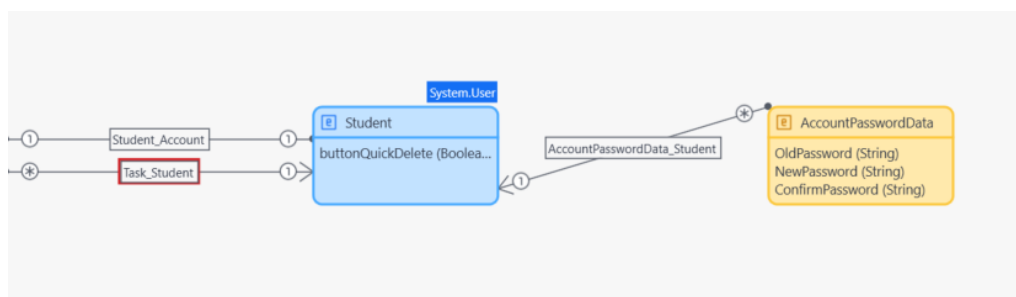
1.3 Δομή της εφαρμογής

Πέρα από τα προκατασκευασμένα modules του Mendix, η λειτουργικότητα της εφαρμογής έχει οργανωθεί σε τρία modules, το Administrator, το TaskManager και το UniTask.

1.3.1 Module Administrator

Το Administrator περιλαμβάνει τη λειτουργικότητα που αφορά τη διαχείριση των χρηστών της εφαρμογής. Όλες οι οντότητες του domain model, οι σελίδες και τα microflows του module έχουν δικαιώματα ανάγνωσης και εγγραφής από τον Administrator ρόλο, όπως ορίζεται στο Security της εφαρμογής.

1.3.1.1 Domain model του Administrator



Το domain model του Administrator περιλαμβάνει την οντότητα Student που κληρονομεί την οντότητα System.User του Mendix. Η οντότητα περιγράφει τον κάθε χρήστη της εφαρμογής και περιλαμβάνει την Boolean ιδιότητα buttonQuickDelete αρχικοποιημένη σε False η οποία χρησιμοποιείται για την ενεργοποίηση της λειτουργίας γρήγορης διαγραφής εργασιών. Η οντότητα Student συσχετίζεται με την οντότητα Account του System.User με σχέση 1-προς-1, η οντότητα Task του TaskManager με σχέση ένα-προς-πολλά (ένα Student συσχετίζεται με πολλά Tasks) και την οντότητα AccountPasswordData με σχέση 1-προς-πολλά (ένα Student συσχετίζεται με πολλά AccountPasswordData).

Η οντότητα AccountPasswordData είναι μη-διατηρήσιμη οντότητα (δεν αποθηκεύεται στη βάση δεδομένων αλλά μόνο στη μνήμη) και περιλαμβάνει τις ιδιότητες OldPassword, NewPassword και ConfirmPassword και χρησιμοποιείται για την αλλαγή του κωδικού πρόσβασης του χρήστη.

1.3.1.2 Σελίδες του Administrator

Στο Administrator περιλαμβάνονται οι εξής σελίδες:

Account_Overview

Name	<Last login>	<Blocked>	<Blocked since>	<Active>	<Web service user>	<Is anonymous>
[Name]	[LastLogin]	[Blocked]	[BlockedSince]	[Active]	[WebServiceUser]	[IsAnonymous]
[Name]	[LastLogin]	[Blocked]	[BlockedSince]	[Active]	[WebServiceUser]	[IsAnonymous]
[Name]	[LastLogin]	[Blocked]	[BlockedSince]	[Active]	[WebServiceUser]	[IsAnonymous]
[Name]	[LastLogin]	[Blocked]	[BlockedSince]	[Active]	[WebServiceUser]	[IsAnonymous]
[Name]	[LastLogin]	[Blocked]	[BlockedSince]	[Active]	[WebServiceUser]	[IsAnonymous]

Η σελίδα χρησιμοποιείται για τη διαχείριση των χρηστών της εφαρμογής από την πλευρά των διαχειριστών.

Χρησιμοποιείται το UniTask_SideBar layout του UniTaskDesignSystem module. Το κύριο μέρος της σελίδας αποτελείται από ένα Data Grid με Data source την οντότητα Student και με στήλες τις ιδιότητες Name, Last login, Blocked, Blocked since, Active, Web service user και Is anonymous.

Account_New

<New Account>

Data View | [AccountPasswordData, page parameter 'AccountPasswordData']

Data View | [Student, over association 'AccountPasswordData_Student']

Username [Name]

<Blocked> ☐ Ναι ☐ Όχι

<Active> ☐ Ναι ☐ Όχι

<User role> [../UserRole/Name] →

<Language> [../Language/Description] v

<Time zone> [../TimeZone/Description] v

<New password> [NewPassword (AccountPasswordData)]

<Confirm password> [ConfirmPassword (AccountPasswordData)]

<Save> <Cancel>

Η σελίδα χρησιμοποιείται για τη δημιουργία νέων χρηστών της εφαρμογής.

Χρησιμοποιείται το PopupLayout layout του Atlas_Core module. Η σελίδα περιλαμβάνει δύο Parameters, το Student και AccountPasswordData του module Administrator. Η σελίδα αποτελείται από δύο εμφωλευμένα Data Views, το εξωτερικό έχει ως Data source το AccountPasswordData, ενώ το εσωτερικό έχει ως Data source τη συσχέτιση του AccountPasswordData με το Student. Η χρήση του AccountPasswordData είναι απαραίτητη καθώς η δημιουργία ενός νέου χρήστη χρειάζεται την αποθήκευση του κωδικού πρόσβασής του.

Στο εσωτερικό Data View περιλαμβάνει Text Boxes, Radio Buttons και Input Reference Set Selectors όπου εισάγονται τιμές για τα Username, Blocked, Active, User role, Language, Time zone, New password και Confirm password. Έχει σημασία να σημειωθεί πως οι ιδιότητες (γνωρίσματα) που αποθηκεύουμε στην πραγματικότητα δεν είναι ιδιότητες του Student αλλά του System.User του οποίου αποτελεί παιδί. Το Input Reference Set Selector χρησιμοποιείται για την επιλογή του UserRole, που αποτελεί διαφορετική σελίδα που θα αναλυθεί στη συνέχεια.

Τέλος, περιλαμβάνεται κουμπί για την αποθήκευση, το οποίο καλεί το microflow ACT_Account_Save του Administrator για την αποθήκευση των τιμών, και κουμπί για την ακύρωση της διαδικασίας.

Account_Edit

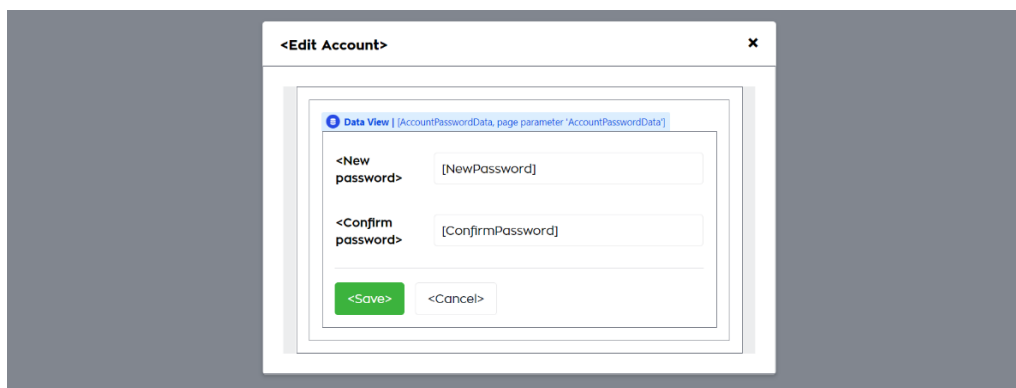
Η σελίδα χρησιμοποιείται για την επεξεργασία υπαρχόντων χρηστών της εφαρμογής.

Χρησιμοποιείται το PopupLayout. Η σελίδα περιλαμβάνει το Parameter Student. Η σελίδα αποτελείται από ένα Data View με Data source το Student με παρόμοια

Text Boxes και Radio Buttons όπως και το Account_New. Επίσης, περιλαμβάνεται το κουμπί που καλεί το microflow ACT_Password_Change για την αλλαγή κωδικού.

Τέλος, περιλαμβάνεται κουμπί για την αποθήκευση και κουμπί για την ακύρωση της διαδικασίας. Τα κουμπιά καλούν προεπιλεγμένες ενέργειες του Mendix.

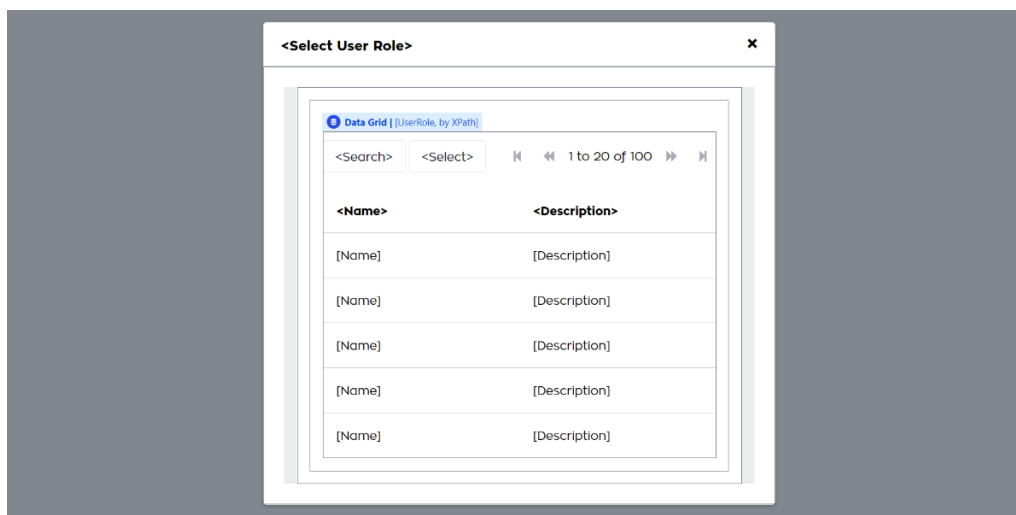
Change_Password



Η σελίδα χρησιμοποιείται για την αλλαγή του κωδικού πρόσβασης υπάρχοντος χρήστη.

Χρησιμοποιείται το PopupLayout. Η σελίδα περιλαμβάνει το Parameter AccountPasswordData, ένα Data View με Data source το Student με τα απαραίτητα Text Boxes για την αλλαγή των τιμών. Τέλος, περιλαμβάνεται κουμπί για την αποθήκευση που καλεί το microflow ChangePassword του Administrator και κουμπί για την ακύρωση της διαδικασίας.

UserRole_Select



Η σελίδα χρησιμοποιείται για την επιλογή του ρόλου του χρήστη κατά τη δημιουρ-

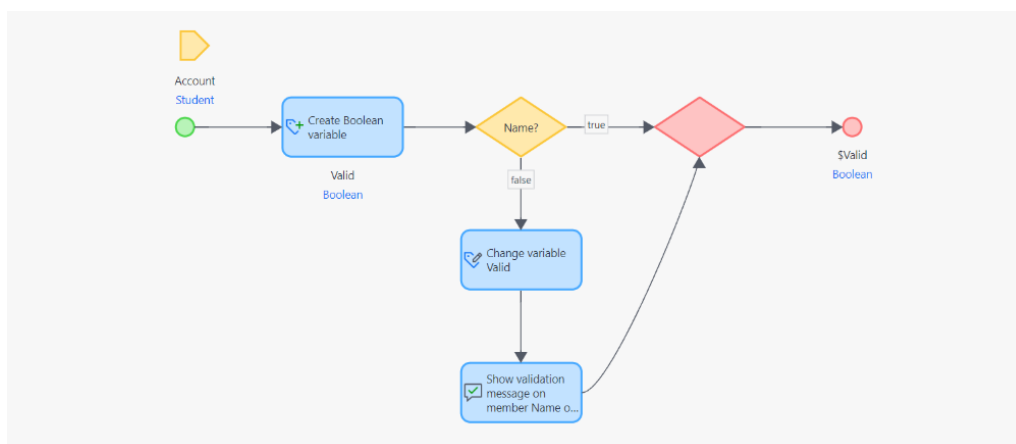
γία νέου χρήστη.

Χρησιμοποιείται το `PopupLayout`. Η σελίδα αποτελείται από ένα Data Grid με Data source το `UserRole` του System.

1.3.1.3 Microflows του Administrator

Στο Administrator περιλαμβάνονται τα εξής microflows:

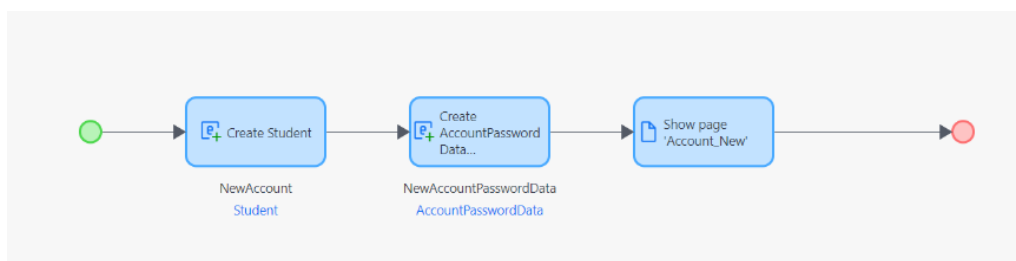
VAL_Account



Το microflow καλείται από το microflow `ACT_Account_Save` για να επικυρώσει το λογαριασμό του χρήστη πριν αποθηκευτούν οι τιμές του.¹

Αρχικά δημιουργείται μια boolean μεταβλητή `Valid` με αρχική τιμή `True` η οποία θα επιστραφεί από το microflow. Στη συνέχεια ελέγχεται αν για το αντικείμενο `Account` τύπου `Student` ισχύει η συνθήκη `(trim($Account/Name) != '')`. Η έκφραση στη συνθήκη αφού καθαρίσει τα κενά (whitespaces) από το `Name` του `Account`, ελέγχει αν είναι διαφορετικό από το κενό string. Αν η συνθήκη δεν ισχύει, τότε η μεταβλητή `Valid` γίνεται `False`, η οποία επιστρέφεται μαζί με ένα popup μήνυμα. Αν η συνθήκη ισχύει, δηλαδή αν υπάρχει όνομα, τότε επιστρέφεται `True`.

ACT_Account_New

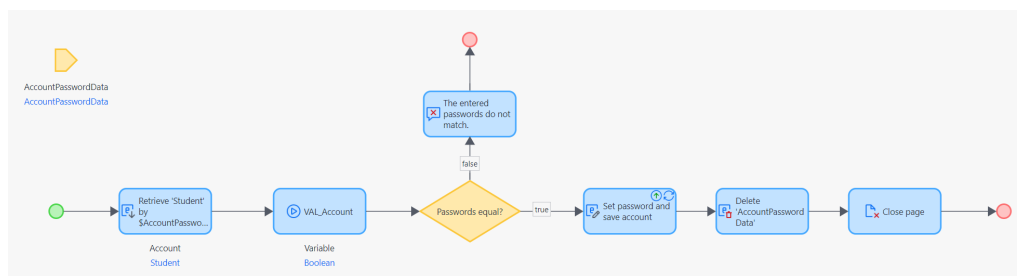


¹Το πρόθεμα `VAL` χρησιμοποιείται στην ονομασία των microflows για να δηλώσει επικύρωση (validation).

Το microflow καλείται από τη σελίδα Account_Overview με σκοπό τη δημιουργία ενός νέου χρήστη.²

Αρχικά δημιουργούνται δύο στιγμιότυπα τύπου Student και AccountPasswordData με ονόματα NewAccount και NewAccountPasswordData αντίστοιχα. Να σημειωθεί πως το NewAccountPasswordData συσχετίζεται με το Student. Τα αντικείμενα δε γίνονται commit ακόμα στη βάση, καθώς είναι κενά. Στη συνέχεια εμφανίζεται η σελίδα Account_New με τα αντικείμενα NewAccount και NewAccountPasswordData ως Parameters.

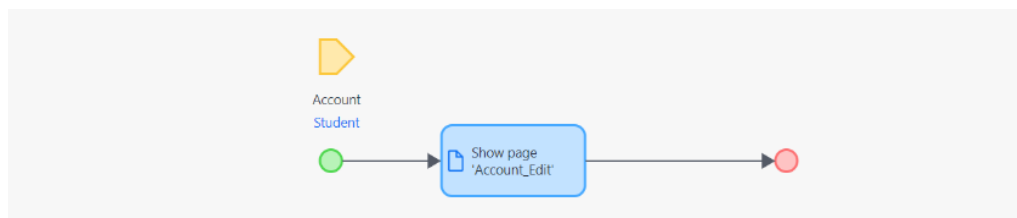
ACT_Account_Save



Το microflow καλείται από τη σελίδα Account_New με σκοπό την αποθήκευση των τιμών του νέου χρήστη.

Το microflow έχει ως Parameter το AccountPasswordData. Μαζί με αυτό, ανακτάται το Student αφού συσχετίζονται, και καλείται το microflow VAL_Account το οποίο ελέγχει αν το Name του Student είναι κενό. Αν το Name είναι κενό, τότε εμφανίζεται ένα popup μήνυμα και το microflow τερματίζεται. Αν το Name δεν είναι κενό, τότε ελέγχεται αν το NewPassword του AccountPasswordData είναι ίσο με το ConfirmPassword, όπως έχουν δοθεί στη φόρμα Account_New. Αν η συνθήκη δεν ισχύει, τότε εμφανίζεται ένα popup μήνυμα και το microflow τερματίζεται. Αν η συνθήκη ισχύει, τότε το NewPassword γίνεται commit στο Account τύπου Student στο γνώρισμα Password το οποίο είναι Hashed string. Στη συνέχεια το αντικείμενο AccountPasswordData διαγράφεται και κλείνει η σελίδα.

ACT_Account_Edit

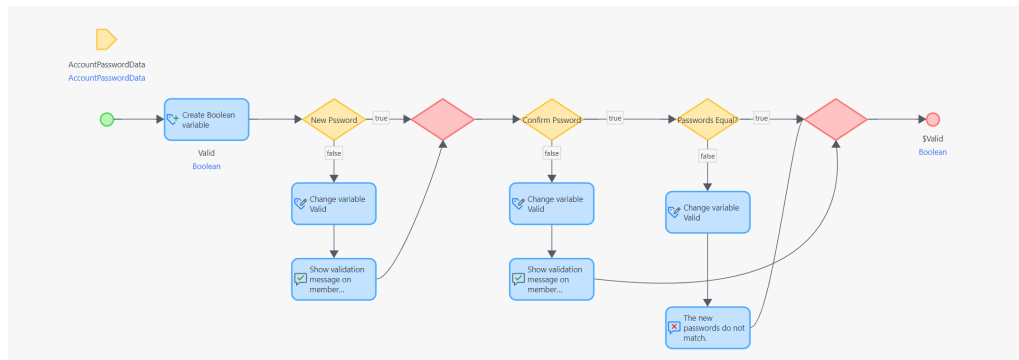


²Το πρόθεμα ACT χρησιμοποιείται στην ονομασία των microflows για να δηλώσει μια ενέργεια (action).

Το microflow καλείται από τη σελίδα Account_Overview με σκοπό την επεξεργασία ενός υπάρχοντος χρήστη.

Το microflow εμφανίζει τη σελίδα Account_Edit με το Account τύπου Student ως Parameter.

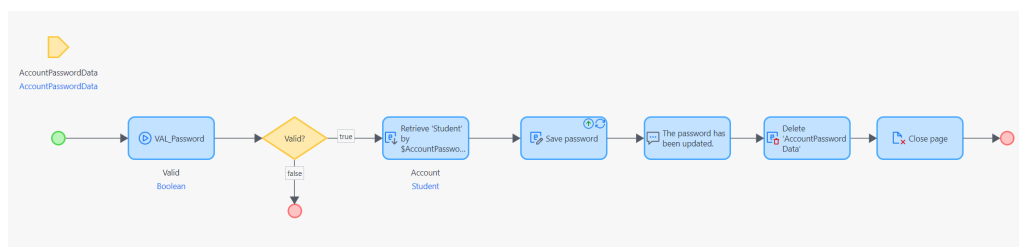
VAL_Password



Το microflow καλείται από το microflow ChangePassword με σκοπό τον έλεγχο των συνθηκών για την αλλαγή του κωδικού πρόσβασης.

Αρχικά δημιουργείται μια boolean μεταβλητή Valid με αρχική τιμή True. Στη συνέχεια ελέγχεται αν για το NewPassword του αντικειμένου AccountPasswordData ισχύει η συνθήκη $(\text{trim}(\$AccountPasswordData/NewPassword) \neq '')$. Η έκφραση στη συνθήκη ελέγχει αν έχει δοθεί όντως νέος κωδικός στη φόρμα της σελίδας Change_Password. Αν η συνθήκη ισχύει, ελέγχεται με παρόμοιο τρόπο και το ConfirmPassword όπως επίσης και το αν είναι ίσο με το NewPassword. Αν κάποια συνθήκη από τις προαναφερθείσες δεν ισχύει, η μεταβλητή Valid γίνεται False και εμφανίζεται κατάλληλο popup μήνυμα. Αν όλες οι συνθήκες ισχύουν, τότε η μεταβλητή Valid παραμένει True και επιστρέφεται από το microflow.

ChangePassword

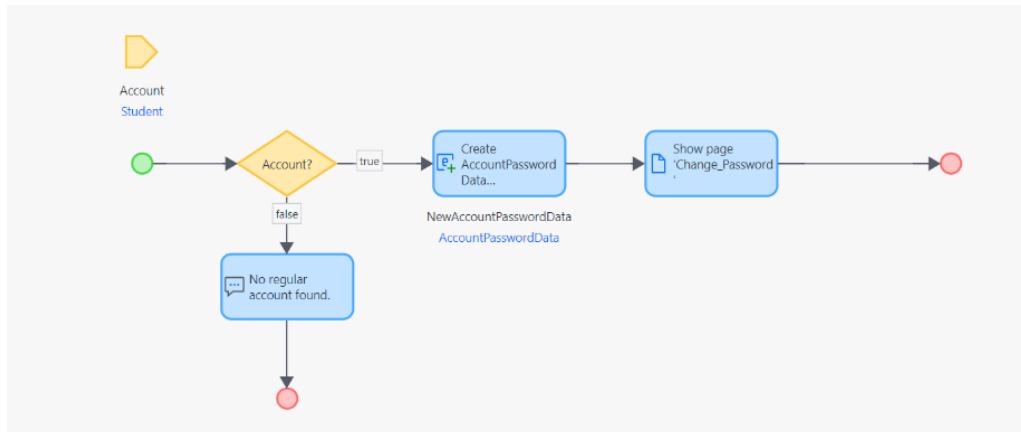


Το microflow καλείται από τη σελίδα Change_Password με σκοπό την αποθήκευση του νέου κωδικού πρόσβασης για έναν υπάρχων χρήστη.

Το microflow έχει ως Parameter το AccountPasswordData. Στην αρχή καλείται το microflow VAL_Password για τον έλεγχο των συνθηκών. Αν η μεταβλητή Valid που

επιστρέφεται από το microflow είναι False, τότε το microflow τερματίζεται. Αν είναι True, τότε γίνεται retrieve και το αντικείμενο Student ως συσχέτιση, γίνεται commit το NewPassword ως Hashed string Password στο Student, εμφανίζεται κατάλληλο popup μήνυμα, διαγράφεται το AccountPasswordData και κλείνει η σελίδα.

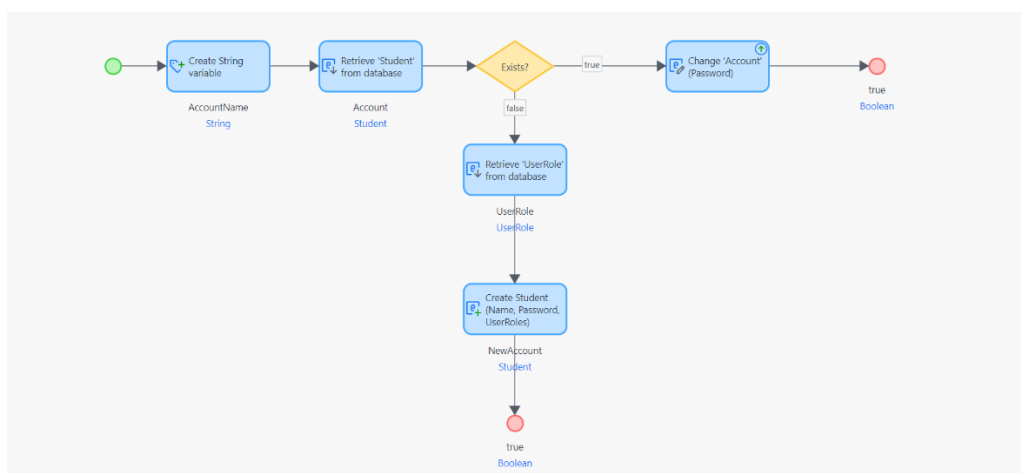
ACT_Password_Change



Το microflow καλείται από τη σελίδα Account_Edit με σκοπό την αλλαγή του κωδικού πρόσβασης ενός υπάρχοντος χρήστη.

Με Parameter το Account τύπου Student, αρχικά ελέγχεται αν υπάρχει όντως κάποιο υπαρκτό Account. Αν δεν υπάρχει, εμφανίζεται popup μήνυμα και το microflow τερματίζεται. Αν υπάρχει, τότε δημιουργείται ένα νέο αντικείμενο AccountPasswordData συσχετισμένο με το Account και εμφανίζεται η σελίδα Change_Password με το AccountPasswordData και Account ως Parameters.

ASU_Administrator_Create



Microflow που καλείται κατά την αρχικοποίηση της εφαρμογής για τη δημιουργία

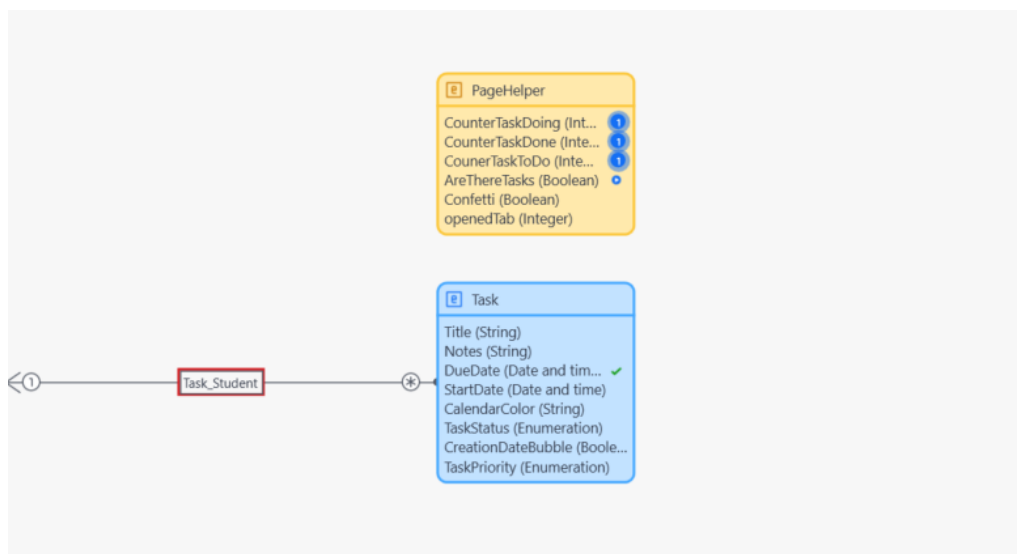
του διαχειριστή της εφαρμογής.³

Αρχικά δημιουργείται το String AccountName με τιμή 'admin' με το username του διαχειριστή. Στη συνέχεια γίνεται retrieve από τη βάση δεδομένων το Student που έχει ως Name το AccountName. Αν δεν υπάρχει τέτοιος λογαριασμός, τότε γίνονται retrieve τα System.UserRoles και δημιουργείται ένα νέο αντικείμενο Student με Name το AccountName, Password η τιμή 'admin' και UserRole το UserRole. Αν υπάρχει ήδη λογαριασμός με το AccountName, τότε αλλάζει ο κωδικός σε 'admin'. Είναι προφανές ότι τα στοιχεία σύνδεσης του διαχειριστή έχουν οριστεί ως 'admin' και 'admin'.

1.3.2 Module TaskManager

Το TaskManager περιλαμβάνει τη λειτουργικότητα που αφορά τη διαχείριση των εργασιών της εφαρμογής. Όλες οι οντότητες του domain model, οι σελίδες και τα microflows του module έχουν δικαιώματα ανάγνωσης και εγγραφής από τον User ρόλο, όπως ορίζεται στο Security της εφαρμογής, με εξαίρεση το Custom_LogIn_Page που έχει δικαίωμα ο Guest.

1.3.2.1 Domain model του TaskManager



Το domain model του TaskManager περιλαμβάνει την οντότητα Task και τη μη-διατηρήσιμη οντότητα PageHelper.

Η οντότητα Task αναπαριστά την εκάστοτε εργασία του χρήστη της εφαρμογής. Περιλαμβάνει τις ιδιότητες Title τύπου String ως 200 χαρακτήρες με το όνομα της εργασίας, Notes τύπου String με απεριόριστους χαρακτήρες όπου μπορούν να προστεθούν σημειώσεις για αυτή και DueDate τύπου Date and time με την ημερομηνία λήξης. Επίσης, περιλαμβάνει τη StartDate τύπου Date and time με την ημερομηνία

³Το πρόθεμα ASU (After Startup) χρησιμοποιείται στην ονομασία των microflows για να δηλώσει ότι καλείται αμέσως μετά την εκκίνηση της εφαρμογής.

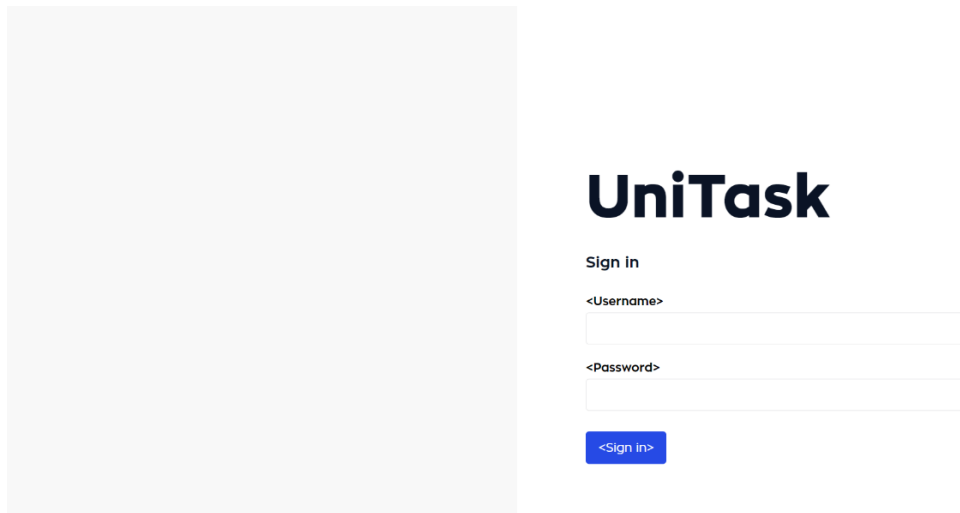
έναρξης, η οποία αρχικοποιείται με την τιμή '%CurrentDateTime%' (Token που επιστρέφει την τρέχουσα ημερομηνία και ώρα) και τη CalendarColor τύπου String που αποθηκεύει το χρώμα της εργασίας στο ημερολόγιο. Λόγω της φύσης του widget του ημερολογίου, το String θα έχει πάντα τη μορφή 'rgb(<0-255>, <0-255>, <0-255>)' και στην οντότητα αρχικοποιείται με την τιμή 'rgb(38,74,229)' που αντιστοιχεί στο μπλε χρώμα.

Επιπλέον, η οντότητα περιλαμβάνει τις ιδιότητες TaskStatus και TaskPriority όπου είναι Enumeration ιδιότητες των Enumeration εγγράφων TaskStatus και TaskPriority, αρχικοποιημένες με To_Do και Low αντίστοιχα. Το TaskStatus καθορίζει την κατάσταση της εργασίας με τις δυνατές καταστάσεις να είναι 'To_Do', 'Doing' και 'Done' ενώ το TaskPriority καθορίζει αν η προτεραιότητα της εργασίας είναι χαμηλή, μεσαία ή υψηλή. Επίσης, περιλαμβάνεται η ιδιότητα CreationDateBubble τύπου Boolean που αρχικοποιείται με True. Η ιδιότητα αυτή χρησιμοποιείται για την εμφάνιση ενός επεξηγηματικού μηνύματος στον χρήστη όταν δημιουργεί μια νέα εργασία. Να σημειωθεί επίσης πως το DateDue περιλαμβάνει ένα Validation rule που ελέγχει αν η ημερομηνία λήξης είναι μετά την ημερομηνία έναρξης StartDate, και αν δεν ισχύει, τότε εμφανίζεται κατάλληλο μήνυμα.

Η οντότητα PageHelper περιλαμβάνει τις ιδιότητες CounterTaskDoing, CounterTaskDone και CounterTaskToDo τύπου Integer, των οποίων οι τιμές καθορίζονται από τα microflows CounterTaskDoing, CounterTaskDone και CounterTaskToDo αντίστοιχα. Οι ιδιότητες αυτές χρησιμοποιούνται για την εμφάνιση των τριών μετρητών. Επίσης, περιλαμβάνει την Boolean ιδιότητα AreThereTasks που καθορίζεται από το microflow AreThereTasks και χρησιμεύει για την εμφάνιση του επεξηγηματικού παραθύρου στο Dashboard, την Boolean ιδιότητα Confetti που αρχικοποιείται με False και χρησιμοποιείται για την εμφάνιση του κομφετί (σύστημα επιβράβευσης), και τέλος την ιδιότητα openedTab τύπου Integer αρχικοποιημένη με μηδέν που χρησιμεύει για την αποθήκευση της τρέχουσας καρτέλας του Dashboard που έχει ανοίξει ο χρήστης.

1.3.2.2 Σελίδες του TaskManager

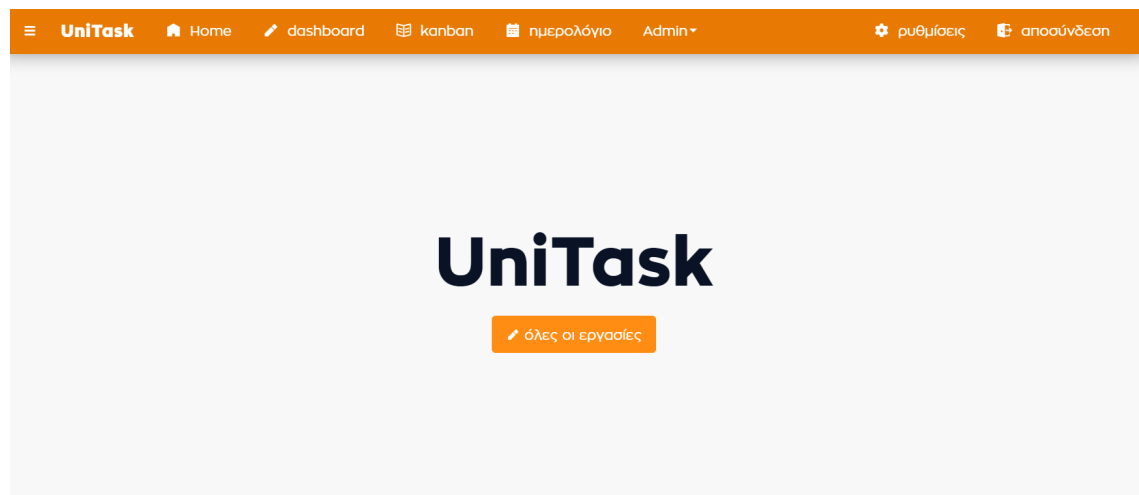
Στο TaskManager περιλαμβάνονται οι εξής σελίδες:

Custom_LogIn_Overview

The screenshot shows a sign-in form titled "UniTask". Below the title, there is a "Sign in" label. The form consists of two input fields: one for "<Username>" and one for "<Password>". Below these fields is a blue button labeled "<Sign in>".

Η σελίδα χρησιμοποιείται για την είσοδο του χρήστη στην εφαρμογή. Πρόσβαση στη σελίδα έχουν μόνο οι Guest χρήστες.

Χρησιμοποιείται το `Layout_LogIn` layout του `UniTaskDesignSystem` module, με μια φόρμα που περιλαμβάνει τα Text Boxes για το Username και Password του χρήστη και το κουμπί για την είσοδο. Τα κουμπιά καλούν προεπιλεγμένες ενέργειες του Mendix.

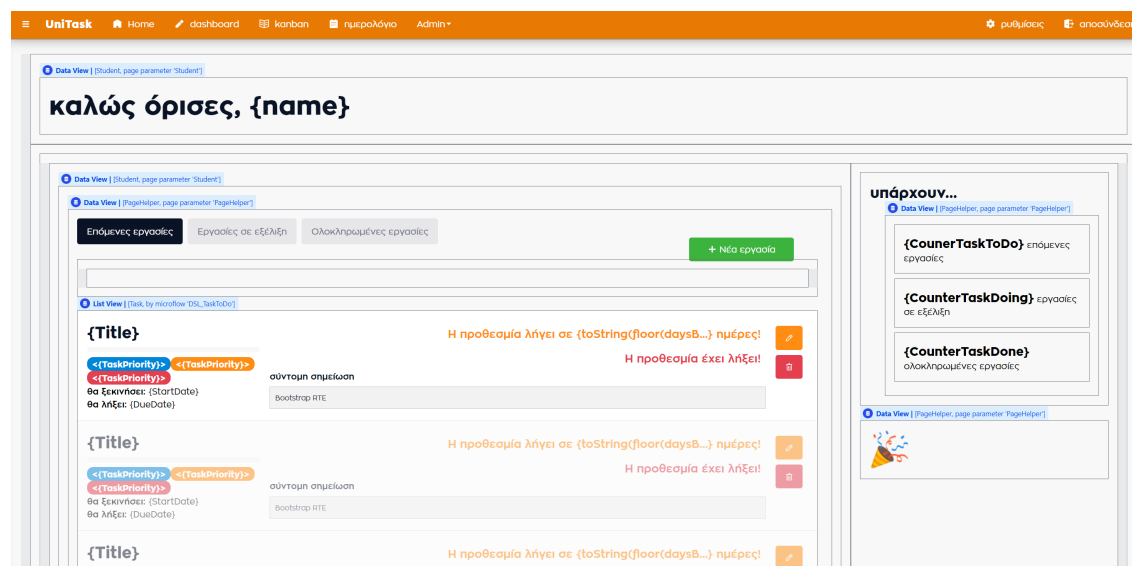
PAGE_Home_Page

Η αρχική σελίδα της εφαρμογής που εμφανίζεται μετά την είσοδο του χρήστη.

Η σελίδα χρησιμοποιεί το `UniTask_TopBar` layout του `UniTaskDesignSystem` module, το οποίο εμφανίζει το κεντρικό μενού της εφαρμογής στο πάνω μέρος της σελίδας. Περιλαμβάνει το τίτλο **UniTask** με ένα call to action κουμπί που καλεί το microflow

ShowPage_TasksOverview.

PAGE_Tasks_Overview



Η σελίδα χρησιμοποιείται για την εμφάνιση και διαχείριση των εργασιών του χρήστη. Χρησιμοποιεί το `Layout_TasksOverview` layout του `UniTaskDesignSystem` module και έχει ως `Parameters` το `Student` και το `PageHelper`.

Η σελίδα αποτελείται από ένα `Layout Grid` με διαφορετικά `rows`. Το πρώτο `row` χρησιμοποιείται για το καλωσόρισμα του χρήστη. Για την εμφάνιση του ονόματός του έχει χρησιμοποιηθεί ένα `Data View` με `Data source` το `Student`. Πίσω από το `text widget` που εμφανίζει το μήνυμα καλωσορίσματος υπάρχει στην πραγματικότητα η έκφραση `καλώς όρισε, {1}`. Τα `{X}` αποτελούν `placeholders` για μεταβλητές: στη συγκεκριμένη περίπτωση το `{1}` αντιστοιχεί στο `Name` του `Student`.

Το δεύτερο `row` του `Layout Grid` περιλαμβάνει ένα εσωτερικό `Layout Grid` που δημιουργεί δύο στήλες (9 και 3)⁴. Στην αριστερή στήλη περιλαμβάνονται οι κάρτες με τις εργασίες. Για να επιτευχθεί αυτό έχουν δημιουργηθεί δύο εμφωλευμένα `Data Views` με `Data sources` τα `Student` και `PageHelper`. Μέσα στο `Data View` περιλαμβάνεται ένα `Tab Container` με τρία `tabs` που αντιστοιχούν στις τρεις καταστάσεις των εργασιών. Σε κάθε `tab` αντιστοιχίζεται ένα ξεχωριστό πράσινο κουμπί (Νέα εργασία) που, ανάλογα με το `tab` που είναι ενεργό, καλεί το `microflow CreateNewTask_ToDo`, `CreateNewTask_Doing` ή `CreateNewTask_Done`. Το περιεχόμενο της καρτέλας είναι ένα `List View` με τις κάρτες εργασιών. Το περιεχόμενο των καρτών γίνεται `populate` (έχει δηλαδή `Data source`) από το `microflow DSL_TaskToDo`, `DSL_TaskDoing` ή `DSL_TaskDone` αντίστοιχα. Για την καλύτερη οργάνωση, η κάρτα ουσιαστικά αποτελεί ένα `Snippet` που επαναχρησιμοποιείται. Έχουν δημιουργηθεί τα `Snippets TaskCard_`

⁴Χρησιμοποιώντας παρόμοια λογική όπως το `Bootstrap`, το `Mendix` χρησιμοποιεί ένα σύστημα 12 στηλών ώστε να καθορίσουμε το πλάτος των στηλών.

Overview_Doing, TaskCard_Overview_ToDo και TaskCard_Overview_Done που θα αναλυθούν στη συνέχεια.

Η δεξιά στήλη με τους μετρητές περιλαμβάνει ένα Data View με Data source το PageHelper. Εσωτερικά περιλαμβάνονται containers με text widgets με τις μεταβλητές CounterTaskToDo, CounterTaskDoing και CounterTaskDone.

Όσον αφορά για το σύστημα επιβράβευσης, το κομφετί αποτελεί ένα add-on widget από το Marketplace του Mendix. Το widget αυτό βρίσκεται τοποθετημένο μέσα σε ένα Data View με Data source το PageHelper και εμφανίζεται μόνο όταν η μεταβλητή Confetti γίνεται True.

Στο κάτω μέρος του Tab container υπάρχει ένα επεξηγηματικό μπλε παράθυρο (βλέπε σχήμα 1.11) που εμφανίζεται μόνο όταν δεν έχουν δημιουργηθεί εργασίες. Αυτό επιτυγχάνεται με την έκφραση not (\$PageHelper/AreThereTasks) στην συνθήκη Visibility του container που αντιπροσωπεύει το παράθυρο. Αξίζει επίσης να σημειωθεί πως τα call-to-action κουμπιά για τη νέα εργασία και τη σελίδα Kanban είναι και αυτά clickable και έχουν χρησιμοποιηθεί custom CSS κλάσεις για την εμφάνισή τους.

POPOUT_Task_NewEdit

1.3.2.3 Microflows του TaskManager

1.3.2.4 Έγγραφα του TaskManager

Βιβλιογραφία

- [1] David J. Anderson. *Kanban: Successful evolutionary change for your technology business*. Blue Hole Press, 2010.
- [2] Asana. *Manage your team's work, projects, & tasks online* • Asana • Asana — *asana.com*. <https://asana.com/>. [Accessed 28-12-2024].
- [3] Atlassian. *Jira | Issue & Project Tracking Software* | Atlassian — *atlassian.com*. <https://www.atlassian.com/software/jira>. [Accessed 28-12-2024].
- [4] Alexander C. Bock και Ulrich Frank. «Low-Code Platform». Στο: *Business and Information Systems Engineering* 63 (6 Δεκ. 2021), σσ. 733–740. issn: 18670202. doi: 10.1007/s12599-021-00726-8.
- [5] Alessio Bucaioni, Antonio Cicchetti και Federico Ciccozzi. «Modelling in low-code development: a multi-vocal systematic review». Στο: *Software and Systems Modeling* 21 (5 Οκτ. 2022), σσ. 1959–1981. issn: 16191374. doi: 10.1007/s10270-021-00964-0.
- [6] *BUS402: History of Project Management* | Saylor Academy — *learn.saylor.org*. <https://learn.saylor.org/mod/page/view.php?id=65663>. [Accessed 26-10-2024].
- [7] Albert E Case. *Computer-aided software engineering (case): technology for improving software development productivity*. 1985.
- [8] E. J. Chikofsky. *Software Development — Computer-Aided Software Engineering (CASE)*.
- [9] *End-user development* - Wikipedia — *en.wikipedia.org*. https://en.wikipedia.org/wiki/End-user_development. [Accessed 29-12-2024].
- [10] *Fourth-generation programming language* - Wikipedia — *en.wikipedia.org*. https://en.wikipedia.org/wiki/Fourth-generation_programming_language. [Accessed 29-12-2024].
- [11] Ryoko Fukuzawa, Hideo Joho και Tetsuya Maeshiro. «Practice and experience of task management of university students: Case of University of Tsukuba, Japan». Στο: *Education for Information* 31 (3 Ιούλ. 2015), σσ. 109–124. issn: 01678329. doi: 10.3233/EFI-150953.
- [12] *Gartner Magic Quadrant for Mobile App Development Platforms* — *gartner.com*. <https://www.gartner.com/en/documents/3882864>. [Accessed 31-12-2024].

- [13] Gartner® Magic Quadrant™ for Enterprise Low-Code Application Platforms — mendix.com. <https://www.mendix.com/resources/gartner-magic-quadrant-for-low-code-application-platforms/>. [Accessed 26-11-2024].
- [14] Dmitry Golovin. *OutSystems as a Rapid Application Development Platform for Mobile and Web Applications*. 2017.
- [15] Jack Goody. «Memory in Oral Tradition». Στο: Cambridge University Press, Μάρ. 2013, σσ. 73–94. doi: 10.1017/cbo9781139171137.005.
- [16] *Guide to the Project Management Body of Knowledge*. Project Management Institute, 2021. ISBN: 1628256648.
- [17] *Hoover Dam – the Greatest Project in Times of the Great Depression. What Can Be Done to Achieve Success? - Strefa PMI* — strefapmi.pl. <https://strefapmi.pl/strefa-studenta/hoover-dam-the-greatest-project-in-times-of-the-great-depression/>. [Accessed 30-10-2024].
- [18] *Hoover Dam | Description, Location, Constructino, Facts, History, & Pictures | Britannica* — britannica.com. <https://www.britannica.com/topic/Hoover-Dam>. [Accessed 25-12-2024].
- [19] Bryan Kasam, Imran McMullen και Micah Kenneweg. *Building Low-Code Applications with Mendix enterprise web and mobile app development made... easy with mendix and the power of no-code development*. Packt Publishing Limited, 2021. ISBN: 9781800201422.
- [20] D. L. Kuhn. *Selecting and Effectively Using a Computer-Aided Software Engineering Tool*. 1989.
- [21] Tim Leung. «Introducing Power Apps». Στο: *Beginning Power Apps: The Non-Developer's Guide to Building Business Applications*. Berkeley, CA: Apress, 2021, σσ. 3–19. ISBN: 978-1-4842-6683-0. doi: 10.1007/978-1-4842-6683-0_1. URL: https://doi.org/10.1007/978-1-4842-6683-0_1.
- [22] Benne Lientz και Kathryn Rea. *Project Management for the 21st Century*. 2007.
- [23] *Manage Your Team's Projects From Anywhere | Trello* — trello.com. <https://trello.com/>. [Accessed 16-10-2024].
- [24] *MDA FAQ | Object Management Group* — omg.org. https://www.omg.org/mda/faq_mda.htm. [Accessed 08-11-2024].
- [25] *Mendix Cloud* — docs.mendix.com. <https://docs.mendix.com/developerportal/deploy/mendix-cloud-deploy/>. [Accessed 04-01-2025].
- [26] *Mendix Documentation* — docs.mendix.com. <https://docs.mendix.com/>. [Accessed 04-01-2025].
- [27] *Mendix Forum - System Module* — community.mendix.com. <https://community.mendix.com/link/space/studio-pro/questions/88842>. [Accessed 05-01-2025].

- [28] OutSystems Platform Architecture | Evaluation Guide | OutSystems — [outsystems.com](https://www.outsystems.com/evaluation-guide/architecture/). URL: <https://www.outsystems.com/evaluation-guide/architecture/>.
- [29] G. Premkumar και Michael Potter. *Adoption of Computer Aided Software Engineering (CASE) Technology: An Innovation Adoption Perspective*.
- [30] QuickBase. *The State Of Citizen Development Report – September 2015*. https://cdn2.hubspot.net/hubfs/172645/QuickBase_Citizen_Developer_Report.pdf. [Accessed 25-11-2024].
- [31] Quickbase. *Gartner® Report: Future of Work Trends* — [quickbase.com](https://www.quickbase.com/gartner-future-of-work). <https://www.quickbase.com/gartner-future-of-work>. [Accessed 25-11-2024].
- [32] *Quipu* - Wikipedia — [en.wikipedia.org](https://en.wikipedia.org/wiki/Quipu). <https://en.wikipedia.org/wiki/Quipu>. [Accessed 22-10-2024].
- [33] *Rapid application development* - Wikipedia — [en.wikipedia.org](https://en.wikipedia.org/wiki/Rapid_application_development). https://en.wikipedia.org/wiki/Rapid_application_development. [Accessed 29-12-2024].
- [34] Thomas Q Reefer. *Lukasa: A Luba Memory Device*. doi: doi:10.2307/3335144.
- [35] E. G. Richards. *Mapping time: The calendar and its history*. Oxford University Press, 2000.
- [36] Eric Rosenbaum. *Next frontier in Microsoft, Google, Amazon cloud battle is over a world without code* — [cnbc.com](https://www.cnbc.com/2020/04/01/new-microsoft-google-amazon-cloud-battle-over-world-without-code.html). <https://www.cnbc.com/2020/04/01/new-microsoft-google-amazon-cloud-battle-over-world-without-code.html>. [Accessed 25-11-2024].
- [37] Davide Di Ruscio κ.ά. «Low-code development and model-driven engineering: Two sides of the same coin?». Στο: *Software and Systems Modeling* 21 (2 Απρ. 2022), σσ. 437–446. issn: 16191374. doi: 10.1007/s10270-021-00970-2.
- [38] Apurvanand Sahay κ.ά. «Supporting the understanding and comparison of low-code development platforms». Στο: *Proceedings - 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020*. Institute of Electrical και Electronics Engineers Inc., Αύγ. 2020, σσ. 171–178. isbn: 9781728195322. doi: 10.1109/SEAA51224.2020.00036.
- [39] Matthias Book Sami Beydeda και Volker Gruhn. *Model-Driven Software Development*.
- [40] Phil Simon. *Low-Code/No-Code: Citizen Developers and the Surprising Future of Business Applications*. 2022.
- [41] Andrew Stellman. *Learning agile: Understanding scrum, XP, lean, and Kanban*. Findaway World, 2023.
- [42] O'Reilly Editorial Team. «Low-Code and the Democratization of Programming». Στο: *O'Reilly Media* (2021).
- [43] *Todoist | A To-Do List to Organize Your Work & Life* — todoist.com. <https://todoist.com/>. [Accessed 16-10-2024].

-
- [44] TrackVia. *The next generation worker: The Citizen Developer – Insights on the behaviors and characteristics of an emerging class of technology users within the enterprise*. https://lumenmarketing.com/wp-content/uploads/2017/11/TV_Citizen_Dev.pdf. [Accessed 25-11-2024]. 2014.
- [45] Julia Castillo Trujillo. *Designing A Time Management App For And With Informatics Students*. 2020.
- [46] *What Is Low-Code? | IBM — ibm.com*. <https://www.ibm.com/topics/low-code>. [Accessed 11-10-2024].
- [47] *WinWorld: Welcome — winworldpc.com*. <https://winworldpc.com/home>. [Accessed 31-10-2024].
- [48] *Your connected workspace for wiki, docs & projects | Notion — notion.so*. <https://www.notion.so/>. [Accessed 16-10-2024].
- [49] Μιχαήλ Ξένος. *Ποιότητα Λογισμικού*. GOTSIS, 2021. ISBN: 9786185560102.