

# Περιεχόμενα

<b>1 Υλοποίηση εφαρμογής</b>	<b>1</b>
1.1 Mockups και σχεδιαστική προσέγγιση . . . . .	1
1.2 Παρουσίαση της εφαρμογής . . . . .	3
1.3 Δομή της εφαρμογής . . . . .	11
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 . . . . .	22
1.3.2.2 Σελίδες του TaskManager . . . . .	23
1.3.2.3 Microflows του TaskManager . . . . .	33
1.3.2.4 Έγγραφα του TaskManager . . . . .	33

# Κεφάλαιο 1

## Υλοποίηση εφαρμογής

Σε αυτό το κεφάλαιο...

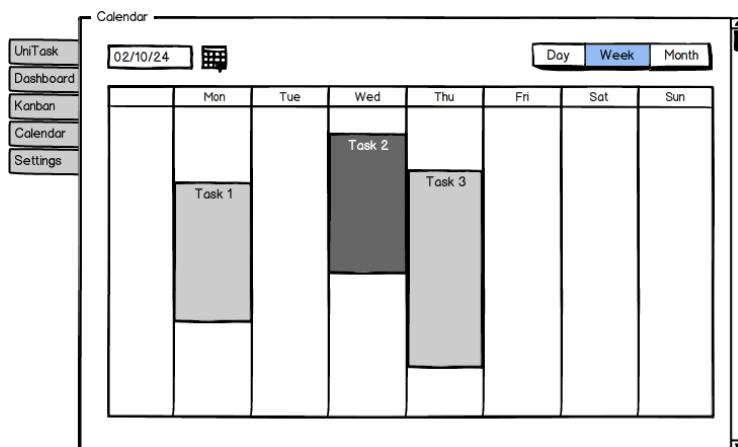
### 1.1 Mockups και σχεδιαστική προσέγγιση

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.

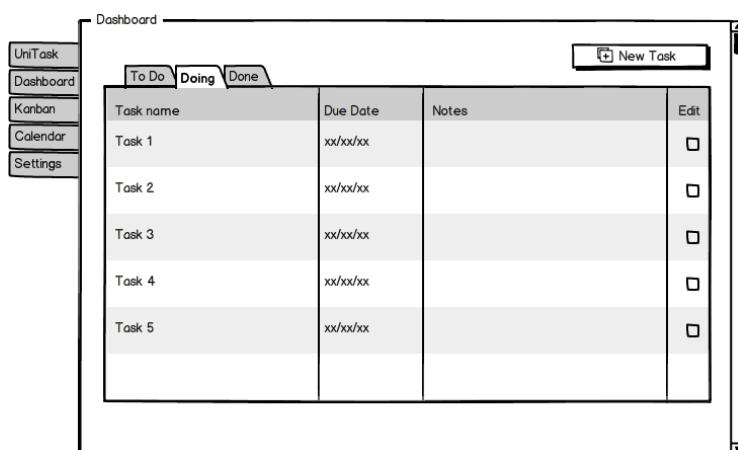
Αρχικά, ως μια εφαρμογή διαχείρισης εργασιών, θα περιλαμβάνει προφανώς ένα σύστημα δημιουργίας, τροποποίησης και διαγραφής εργασιών, καθορισμού του χρόνου έναρξης και λήξεώς τους, και η κατηγοριοποίηση των εργασιών σε κατηγορίες ανάλογα με το αν έχουν πραγματοποιηθεί, αν πραγματοποιούνται και αν έχουν 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.

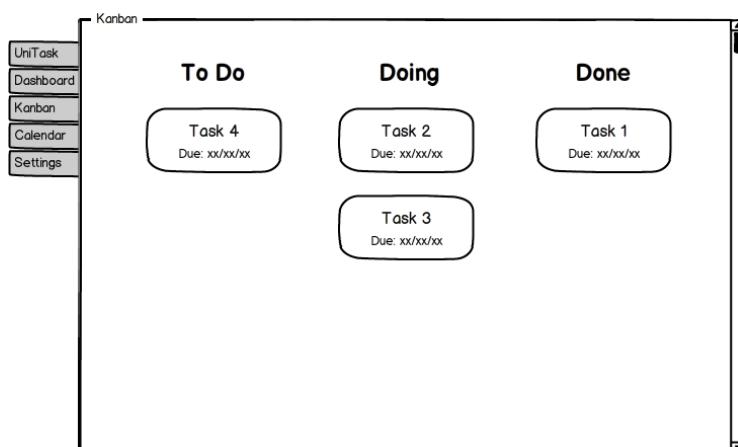
Στα σχήματα 1.1, 1.2 και 1.3 παρουσιάζονται κάποια αρχικά mockups που χρησιμοποιήθηκαν για τον σχεδιασμό της εφαρμογής.



Σχήμα 1.1: Mockup Calendar σελίδας



Σχήμα 1.2: Mockup Dashboard σελίδας

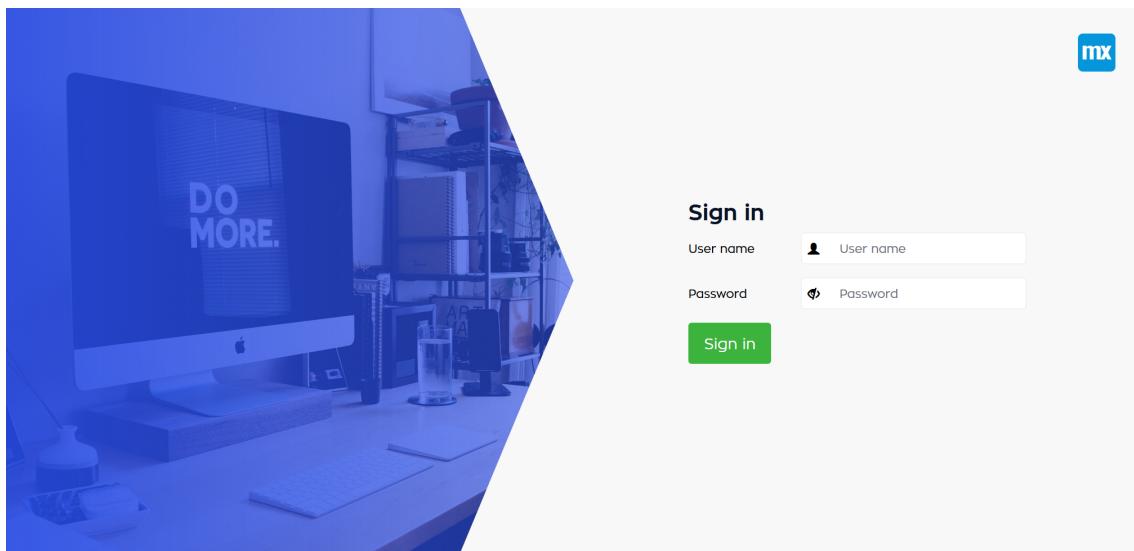


Σχήμα 1.3: Mockup Kanban σελίδας

## 1.2 Παρουσίαση της εφαρμογής

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.

Το σύστημα αναγνωρίζει δύο επίπεδα πρόσβασης, ανάλογα με τα στοιχεία σύνδεσης που εισάγονται: δικαιώματα διαχειριστή (Administrator) και δικαιώματα χρήστη (User). Ο ρόλος του χρήστη (User) αντιστοιχεί σε φοιτητές που κάνουν χρήση της εφαρμογής, ενώ ο ρόλος του διαχειριστή παρέχει επιπλέον λειτουργίες διαχείρισης.



Σχήμα 1.4: Σελίδα σύνδεσης

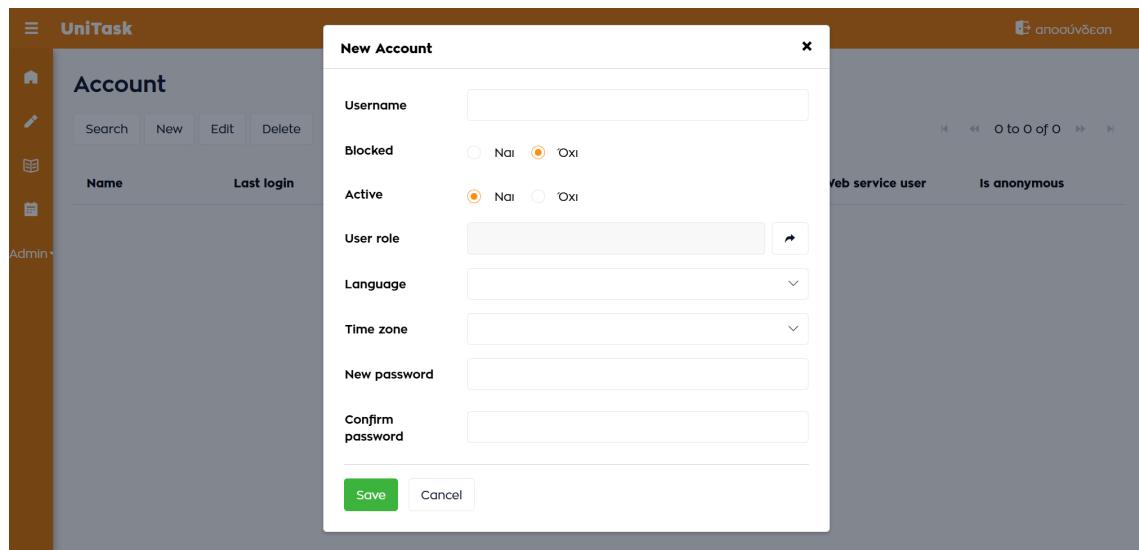
Αρχικά, προγματοποιείται σύνδεση με τον λογαριασμό διαχειριστή (Administrator) προκειμένου να παρουσιαστούν οι λειτουργίες διαχείρισης χρηστών, συμπεριλαμβανομένης της δυνατότητας προσθήκης νέου χρήστη. Μετά την επιτυχή εισαγωγή των διαπιστευτηρίων του διαχειριστή, τα οποία έχουν οριστεί προκαταβολικά 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 αναδυόμενη 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 επιλέγεται αν πρόκειται για προσθήκη διαχειριστή ή χρήστη, του κωδικού πρόσβασης



Σχήμα 1.5: Σελίδα διαχείρισης χρηστών

(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

Ένας νέος χρήστης δημιουργείται με το όνομα Foithths. Ο χρήστης προστίθεται στη λίστα χρηστών, όπως φαίνεται στο 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.

Μετά την αποσύνδεση από τον λογαριασμό διαχειριστή και τη σύνδεση ως Foithths, εμφανίζεται η αρχική σελίδα της εφαρμογής (σχήμα 1.10). Η σελίδα περιλαμβάνει

Name	Last login	Blocked	Blocked since	Active	Web service user	Is anonymous
Foithths	Oxi	Nai		Oxi	Oxi	Oxi

Σχήμα 1.7: Λίστα χρηστών με τον χρήστη Foithths

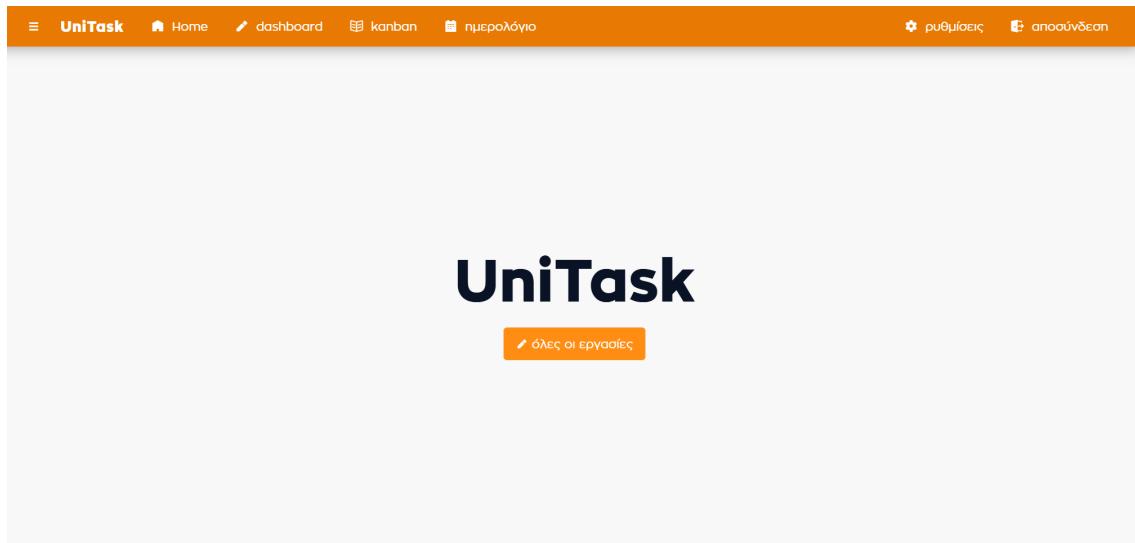
Σχήμα 1.8: Επεξεργασία στοιχείων χρήστη

Name	Last login	Blocked	Blocked since	Active	Web service user	Is anonymous
Foithths	Oxi	Nai		Oxi	Oxi	Oxi

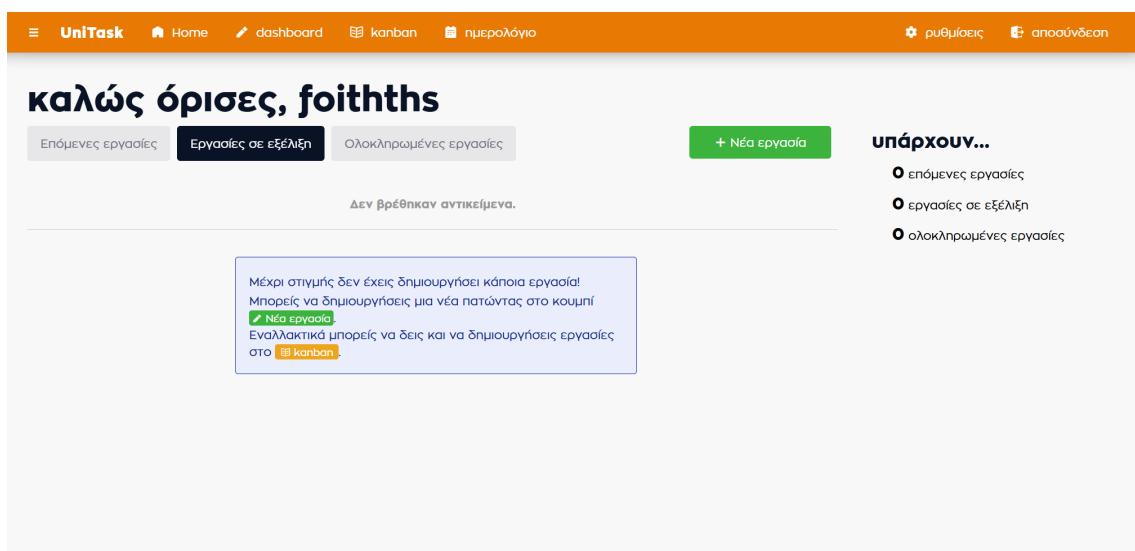
Σχήμα 1.9: Αναζήτηση χρήστη

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 σελίδα προβολής, δημιουργίας και επεξεργασίας των εργασιών του χρήστη. Περιλαμ-



Σχήμα 1.10: Αρχική σελίδα εφαρμογής



Σχήμα 1.11: Dashboard εργασιών

βάνονται 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).

The image consists of two vertically stacked screenshots of the UniTask application's interface, specifically the 'New Project' creation dialog.

**Screenshot 1 (Top):**

- Title:** Νέα εργασία
- Status:** Επόμενη (Upcoming)
- Priority:** Χαμηλή (Low)
- Color:** Blue
- Start Date:** 11/1/2025, 7:30 μμ.
- End Date:** 18/1/2025, 7:30 μμ.
- Note:** Η ημερομηνία έχει ήδη οριστεί σε μια εβδομάδα από τώρα!
- Buttons:** Αποθήκευση (Save) and Ακύρωση (Cancel)

**Screenshot 2 (Bottom):**

- Title:** Συγγραφή Διπλωματικής
- Status:** Μεσαία (Medium)
- Priority:** Μεσαία (Medium)
- Color:** Green
- Start Date:** 1/12/2024, 7:30 μμ.
- End Date:** 1/2/2025, 7:30 μμ.
- Buttons:** Αποθήκευση (Save) and Ακύρωση (Cancel)

**Side Panel (Both Screens):**

- Επόμενες εργασίες**: 0 εργασίες
- Εργασίες σε εξέλιξη**: 1 εργασία
- Ολοκληρωμένες εργασίες**: 0 εργασίες

Σχήμα 1.12: Δημιουργία νέας εργασίας

Πατώντας το, εμφανίζεται ένα αναδυόμενο 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 αναδυόμενο παράθυρο ορίζεται προεπιλεγμένα 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.

Σημειώνεται ότι οι επιλογές για την κατάσταση και την προτεραιότητα της εργασίας είναι χρωματικά κωδικοποιημένες (color-coded) και συνοδεύονται από γραφικά σύμβολα, 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 παρουσιάζει το UniTask dashboard με έναν πρόσφατα δημιουργημένη εργασία:

- Τίτλος:** καλώς όρισες, foithths
- Κατάσταση:** Εργασίες σε εξέλιξη (blue bar)
- Προτεραιότητα:** Μεσαία (yellow bar)
- Συγγραφή Διπλωματικής:** Μεσαία
- Σύντομη σημείωση:** Σύντομη σημείωση
- Επομένες εργασίες:** Εργασίες σε εξέλιξη, Ολοκληρωμένες εργασίες
- Άλλα:** + Νέα εργασία, Επόμενες εργασίες, Ρυθμίσεις, Αποσύνδεση
- Υπάρχουν...**

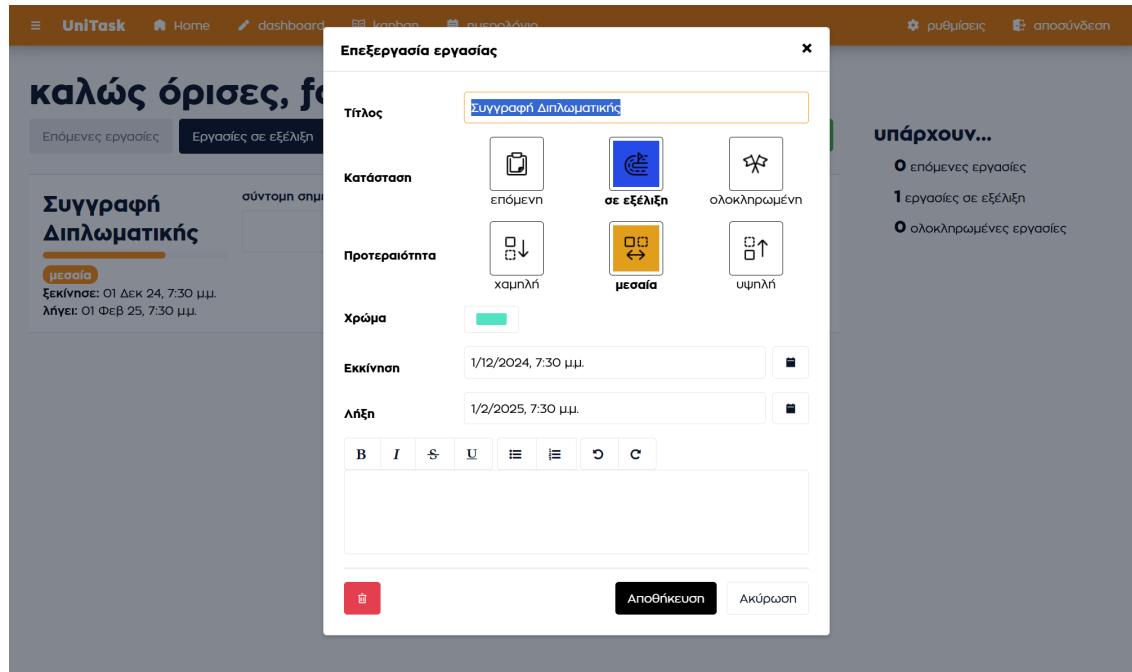
  - 0 επόμενες εργασίες
  - 1 εργασίες σε εξέλιξη
  - 0 ολοκληρωμένες εργασίες

Σχήμα 1.14: Dashboard με δημιουργημένη εργασία

Στη σελίδα πλέον είναι δημιουργημένη η κάρτα με την πρώτη μας εργασία (σχήμα 1.14). Το πλαίσιο σύντομη σημείωση επιτρέπει την εισαγωγή μιας συνοπτικής περιγραφής 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 δεξιό μέρος της κάρτας εμφανίζονται το εικονίδιο για την επεξεργασία της εργασίας.

Η ταξινόμηση των εργασιών στο dashboard βασίζεται στην προτεραιότητα και

στον χρόνο λήξης τους. Έτσι μια εργασία με υψηλή προτεραιότητα θα εμφανίζεται φηλότερα από μια εργασία με χαμηλή προτεραιότητα που δε λήγει σύντομα.



Σχήμα 1.15: Επεξεργασία εργασίας

Με την επιλογή του εικονιδίου επεξεργασίας, εμφανίζεται το αναδυόμενο παράθυρο (σχήμα 1.15), το οποίο επιτρέπει την τροποποίηση των στοιχείων της εργασίας, συμπεριλαμβανομένου του πλαισίου σύντομη σημείωση. Το ίδιο παράθυρο περιλαμβάνει και το κουμπί διαγραφής της εργασίας.

Σχήμα 1.16: Σελίδα Kanban

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 απεικονίζονται ως κάρτες που περιλαμβάνουν τον τίτλο, την προτεραιότητα, καθώς και την ημερομηνία έναρξης και λήξης τους, ενώ πατώντας πάνω στην κάρτα μιας εργασίας, εμφανίζεται το αναδυόμενο 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 τη λίστα των εργασιών.

The screenshot shows the UniTask application's calendar interface. At the top, there is a navigation bar with links for 'UniTask', 'Home', 'dashboard', 'kanban', and 'ημερολόγιο'. On the far right of the bar are icons for 'ρυθμίσεις' (Settings) and 'αποσύνδεση' (Logout).

The main area is titled 'ημερολόγιο' (Calendar). It displays a monthly calendar for January 2025. The days of the week are labeled in Greek: Δευτέρα (Monday), Τρίτη (Tuesday), Τετάρτη (Wednesday), Πέμπτη (Thursday), Παρασκευή (Friday), Σαββατοκύρια (Saturday), and Κυριακή (Sunday). The dates are numbered from 30 to 05. Specific tasks are highlighted with colored bars: teal for 'Συγγραφή Διπλωματικής' (Writing of the diploma) on various days; light blue for another instance of the same task; and dark blue for a third instance. A sidebar on the right provides a summary of tasks from January 1st to January 5th, showing a total of 01 ΔΕΚ - 01 ΦΕΒ.

Σχήμα 1.17: Σελίδα Calendar

Η σελίδα **ρυθμίσεις** (σχήμα 1.18) παρέχει επιλογές για τη μαζική διαγραφή εργασιών ή την ενεργοποίηση της λειτουργίας γρήγορης διαγραφής. 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 αναδυόμενα μηνύματα προκειμένου να αποφευχθούν ακούσιες διαγραφές εργασιών (σχήμα 1.19).

Σχήμα 1.19: Σελίδα ρυθμίσεων

### 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

**Dashboard**

**Kanban**

**Πημερολόγιο**

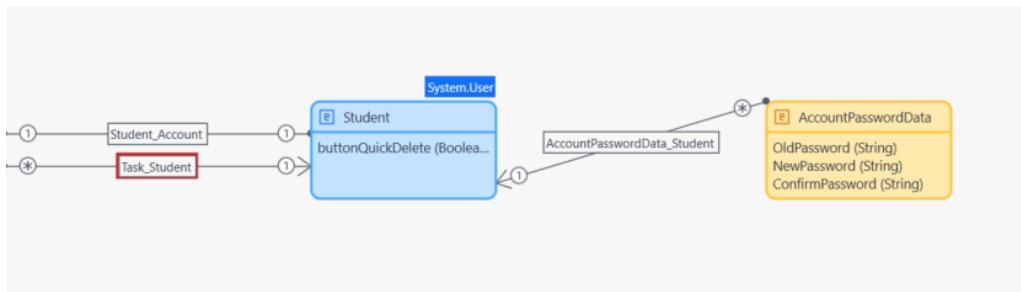
Σχήμα 1.20: Οι σελίδες Dashboard, Kanban και Calendar μετά την αρχικοποίηση των εργασιών. Στο Dashboard φαίνεται η λειτουργία γρήγορης διαγραφής εργασιών.

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**

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**

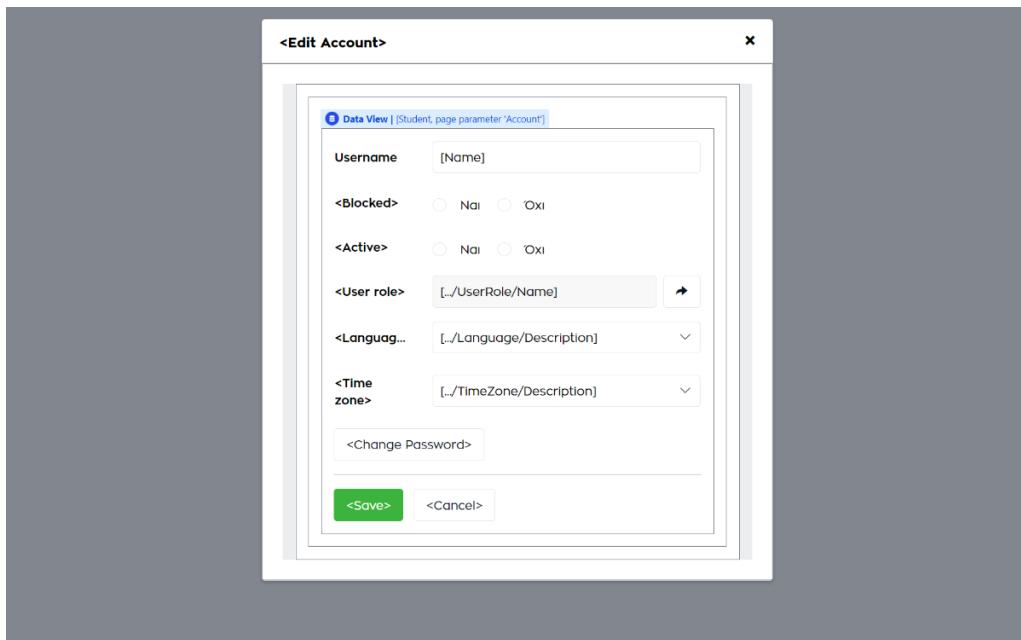
Η σελίδα χρησιμοποιείται για τη δημιουργία νέων χρηστών της εφαρμογής.

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 αποθηκεύομε 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**



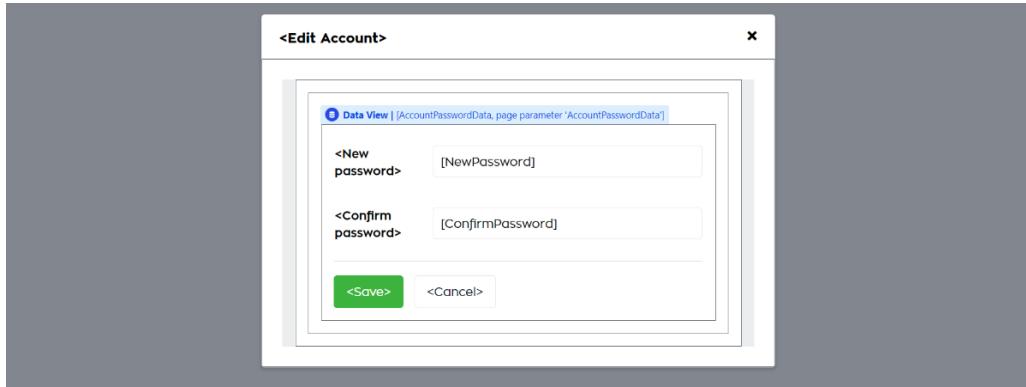
Η σελίδα χρησιμοποιείται για την επεξεργασία υπαρχόντων χρηστών της εφαρμογής.

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.

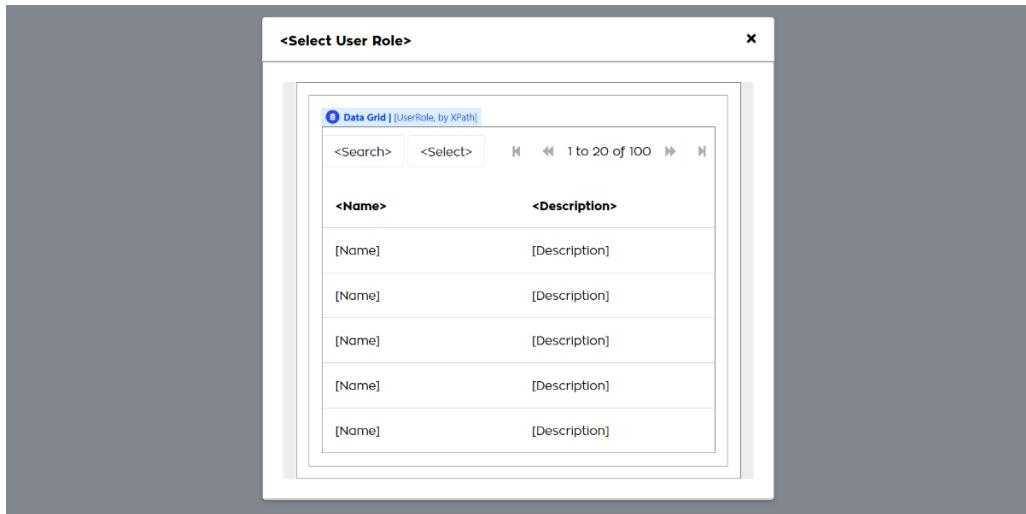
### **Change\_Password**



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-

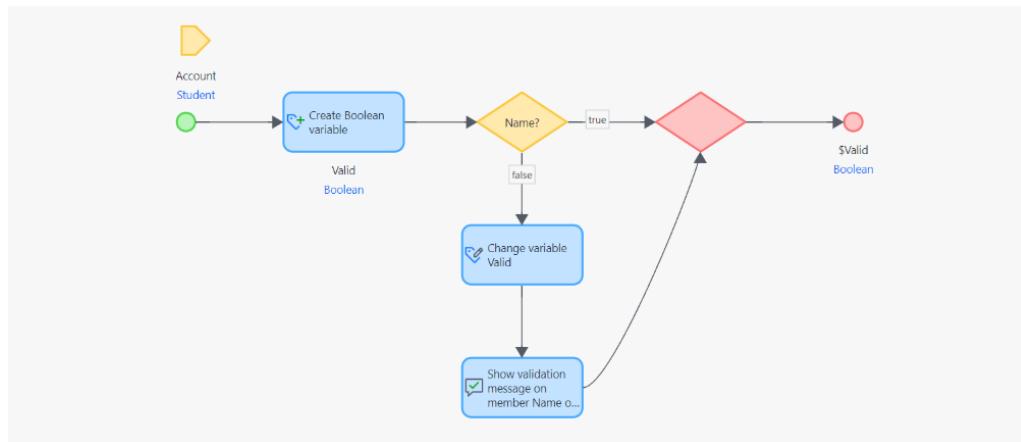
γία νέου χρήστη.

Χρησιμοποιείται το PopupLayout. Η σελίδα αποτελείται από ένα Data Grid με Data source το UserRole του System.<sup>1</sup>

### 1.3.1.3 Microflows του Administrator

Στο Administrator περιλαμβάνονται τα εξής microflows:

#### VAL\_Account

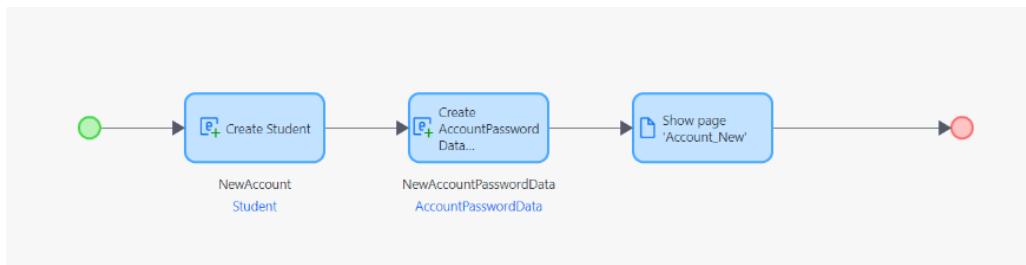


Το microflow καλείται από το microflow ACT\_Account\_Save για να επικυρώσει τον λογαριασμό του χρήστη πριν αποθηκευτούν οι τιμές του.<sup>2</sup>

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.

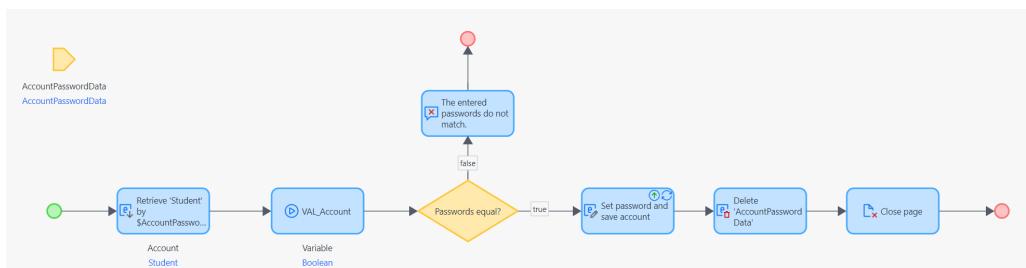
<sup>1</sup>Τεχνικά, λόγω των ρόλων χρηστών που έχουν αληθονομηθεί από το Mendix περιλαμβάνεται και ο ρόλος Guest, ο οποίος στην πράξη δε χρησιμοποιείται καθώς έχει πρόσβαση μόνο στη σελίδα σύνδεσης.

<sup>2</sup>Το πρόθεμα VAL χρησιμοποιείται στην ονομασία των microflows για να δηλώσει επικύρωση (validation).

**ACT\_Account\_New**

Το microflow καλείται από τη σελίδα Account\_Overview με σκοπό τη δημιουργία ενός νέου χρήστη.<sup>3</sup>

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**

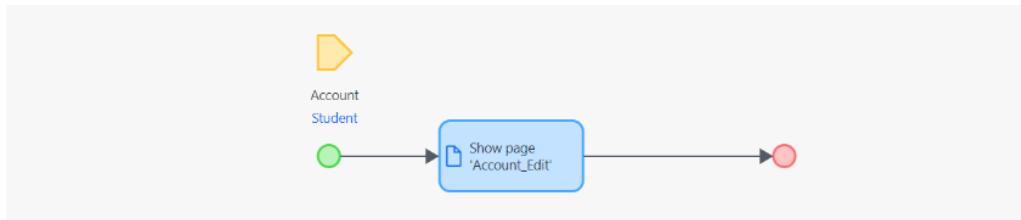
Το microflow καλείται από τη σελίδα Account\_New με σκοπό την αποθήκευση των τιμών του νέου χρήστη.

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 popout 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 popout μήνυμα και το microflow τερματίζεται. Αν η συνθήκη ισχύει, τότε το newPassword γίνεται commit στο Account τύπου Student

<sup>3</sup>Το πρόθεμα ACT χρησιμοποιείται στην ονομασία των microflows για να δηλώσει μια ενέργεια (action).

στο γνώρισμα Password το οποίο είναι Hashed string. Στη συνέχεια το αντικείμενο AccountPasswordData διαγράφεται και κλείνει η σελίδα.

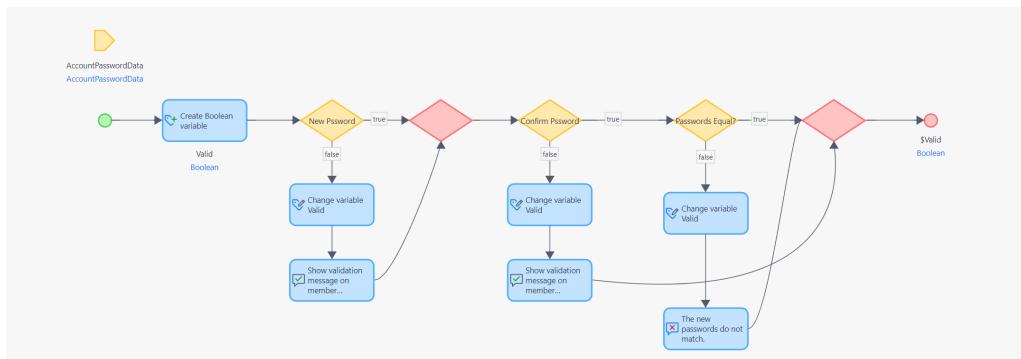
#### **ACT\_Account\_Edit**



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.

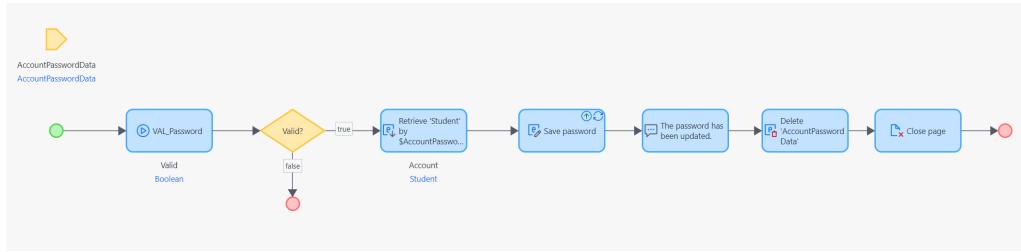
#### **VAL\_Password**



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 (`trim($AccountPasswordData/NewPassword) != ''`). 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 πορούτ μήνυμα. Αν όλες οι συνθήκες ισχύουν, τότε η μεταβλητή Valid παραμένει True και επιστρέφεται από το microflow.

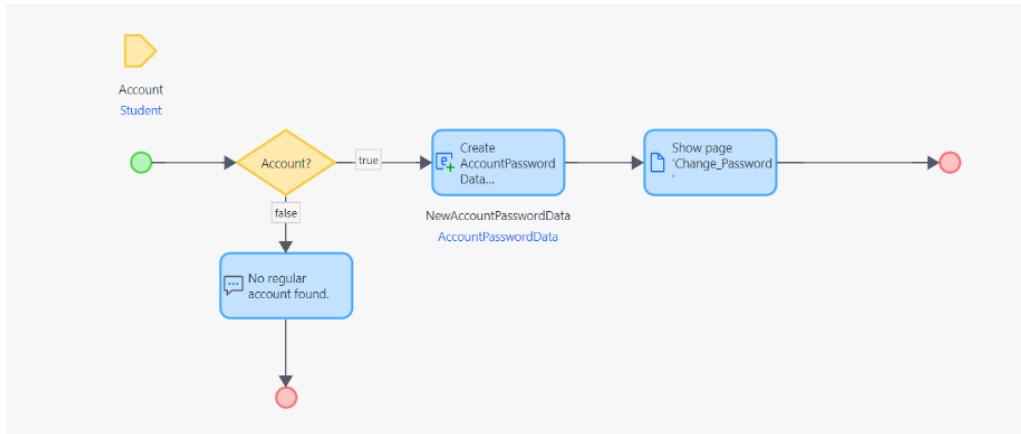
### ChangePassword



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

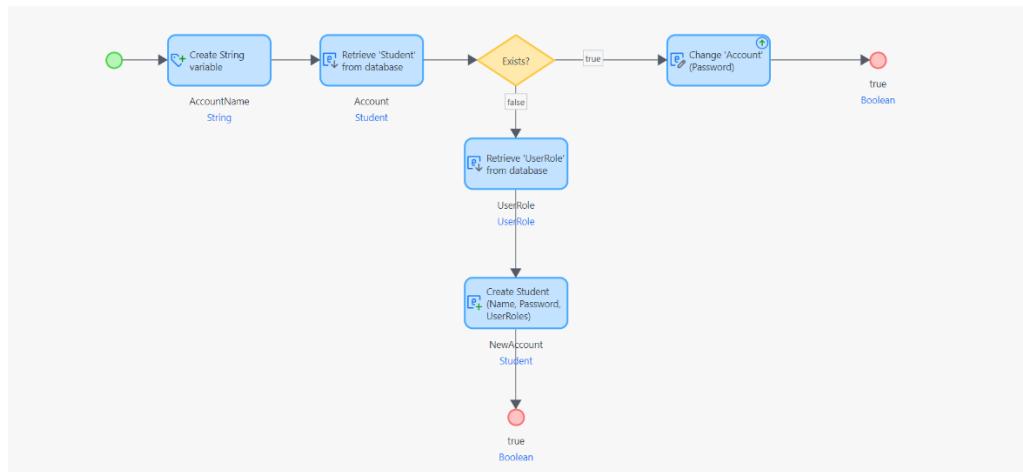
To microflow έχει ως Parameter το `AccountPasswordData`. Στην αρχή καλείται το microflow `VAL_Password` για τον έλεγχο των συνθηκών. Αν η μεταβλητή `Valid` που επιστρέφεται από το microflow είναι `False`, τότε το microflow τερματίζεται. Αν είναι `True`, τότε γίνεται `retrieve` και το αντικείμενο `Student` ως συσχέτιση, γίνεται `commit` το `NewPassword` ως `Hashed string Password` στο `Student`, εμφανίζεται κατάλληλο πορούτ 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 πορούτ 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.<sup>4</sup>

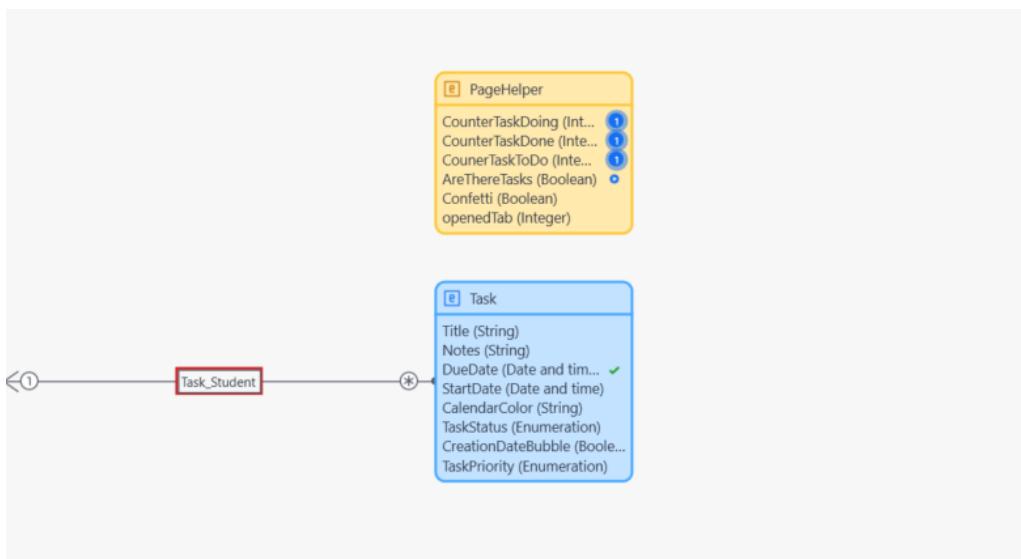
Αρχικά δημιουργείται το String AccountName με τιμή 'admin' με το username του διαχειριστή. Στη συνέχεια γίνεται retrieve από τη βάση δεδομένων το Student που έχει ως Name το AccountName. Αν δεν υπάρχει τέτοιος λογαριασμός, τότε γίνονται retrieve τα System.UserRoles και δημιουργείται ένα νέο αντικείμενο 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 o Guest.

<sup>4</sup>Το πρόθεμα ASU (After Startup) χρησιμοποιείται στην ονομασία των microflows για να δηλώσει ότι καλείται αμέσως μετά την εκκίνηση της εφαρμογής.

### 1.3.2.1 Domain model του TaskManager



To domain model του TaskManager περιλαμβάνει την οντότητα Task και τη μηδιατηρήσιμη 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 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.

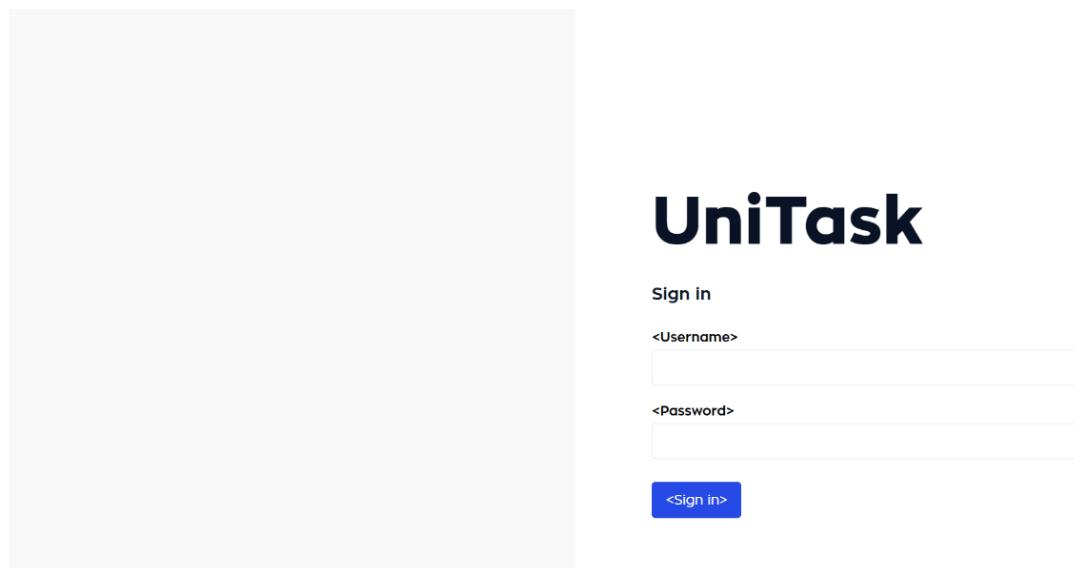
Η οντότητα PageHelper περιλαμβάνει τις ιδιότητες CounterTaskDoing, CounterTaskDone και CounterTaskToDo τύπου Integer, των οποίων οι τιμές καθορίζο-

νται 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

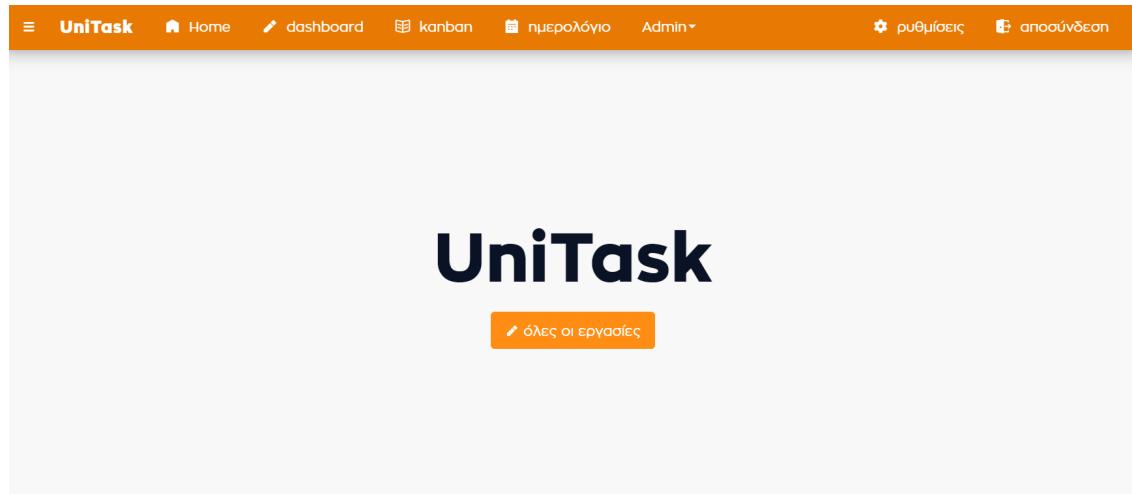
Στο TaskManager περιλαμβάνονται οι εξής σελίδες:

Custom\_LogIn\_Overview



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 UniTask\_TopBar 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 άριστες, {1}. Τα {X} αποτελούν placeholders για μεταβλητές στη συγκεκριμένη περίπτωση το {1} αντιστοιχεί στο Name του Student.

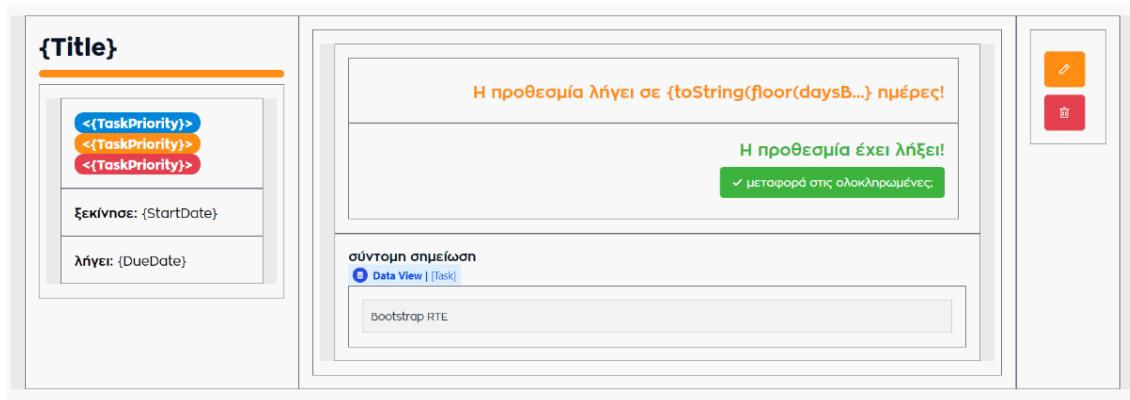
Το δεύτερο row του Layout Grid περιλαμβάνει ένα εσωτερικό Layout Grid που δημιουργεί δύο στήλες (9 και 3)<sup>5</sup>. Στην αριστερή στήλη περιλαμβάνονται οι κάρτες με τις εργασίες. Για να επιτευχθεί αυτό έχουν δημιουργηθεί δύο εμφωλευμένα Data Views με Data sources τα Student και PageHelper. Μέσα στο 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\_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.

<sup>5</sup>Χρησιμοποιώντας παρόμοια λογική όπως το Bootstrap, το Mendix χρησιμοποιεί ένα σύστημα 12 στηλών ώστε να καθορίσουμε το πλάτος των στηλών.

**TaskCard\_Overview\_Doing**

Πρόκειται για ένα Snippet που χρησιμοποιείται για την εμφάνιση των καρτών με τις εργασίες που βρίσκονται στην κατάσταση Doing. Έχει ως Parameters τα Task, PageHelper και Student.

Αποτελείται από ένα Layout Grid με τρεις στήλες. Η πρώτη στήλη περιλαμβάνει τον τίτλο, το Progress Bar widget και ένα εσωτερικό Layout Grid με την προτεραιότητα της εργασίας και τις ημερομηνίες έναρξης και λήξης.

Το Progress Bar χρησιμοποιείται για την εμφάνιση της προόδου της εργασίας και ολοκληρώνεται όσο πλησιάζει η ημερομηνία λήξης. Αυτό επιτυγχάνεται με τον καθορισμό τριών τιμών: Current, Minimum και Maximum value. Στις τιμές Minimum και Maximum value καθορίζονται οι εκφράσεις `dateTimeToEpoch($Task/StartTime)` και `dateTimeToEpoch($Task/DueDate)` αντίστοιχα, στις οποίες ουσιαστικά μετατρέπεται η ημερομηνία και ώρα σε ακέραιο αριθμό. Αυτό βοηθάει στο να είναι καθορισμένο ένα άνω και κάτω αριθμητικό όριο στο οποίο θα κινείται το Current value. Το Current value καθορίζεται από την έκφραση<sup>6</sup>:

```
1 if [%CurrentDateTime%] > $Task/DueDate then dateTimeToEpoch($Task/DueDate)
2 else if [%CurrentDateTime%] < $Task/StartTime then dateTimeToEpoch($Task/StartTime)
3 else dateTimeToEpoch[%CurrentDateTime%]
```

η οποία καταφέρνει τη συγκράτηση της τιμής μέσα σε αυτά τα όρια.

Η προτεραιότητα εμφανίζεται ως ένα Badge widget. Στην κάρτα βρίσκονται το ποιο είναι τα τρία, και ανάλογα με το ποιο είναι το TaskPriority εμφανίζεται και εξαφανίζονται τα αντίστοιχα Badges. Τέλος, η ημερομηνία έναρξης και λήξης έχει επιλεχθεί να εμφανίζεται με τη μορφή dd MMM yy, h:mm a, που αντιστοιχεί σε “23 Απρ 18, 1:37 μ.μ.” για παράδειγμα.

<sup>6</sup>Η έκφραση επιστρέφει την τρέχουσα ημερομηνία σε Epoch μορφή, ή –σε περίπτωση που βρισκόμαστε πριν την ημερομηνία έναρξης ή μετά την ημερομηνία λήξης– αυτές τις ημερομηνίες πάλι σε Epoch μορφή. Η αναπαράσταση μιας χρονικής στιγμής σε Epoch βοηθάει στη σύγχριση μεταξύ ακεραίων για το Progress Bar. Να σημειωθεί πως λέγονται Epoch μορφή εννοούμε τα δευτερόλεπτα που έχουν περάσει από τη 1η Ιανουαρίου 1970 μέχρι τη χρονική στιγμή που καθορίζουμε.

Η δεύτερη στήλη περιλαμβάνει ένα Layout Grid με διαφορετικά rows, το πρώτο αφορά δύο containers που εμφανίζονται υπό συνθήκη και περιλαμβάνουν ενημερώσεις για το αν λήγει μια εργασία σε λιγότερο από μια εβδομάδα ή αν έχει ήδη λήξει. Το πρώτο container εμφανίζεται από την έκφραση<sup>7</sup>:

```
1 if daysBetween($Task/DueDate, [%CurrentDateTime%]) < 7 and daysBetween($Task/DueDate,
2 [%CurrentDateTime%]) > 0 and $Task/DueDate > [%CurrentDateTime%]
then true else false
```

και εμφανίζει “Η προθεσμία λήγει σε {1} ημέρες!” όπου το {1} αντιστοιχεί σε:

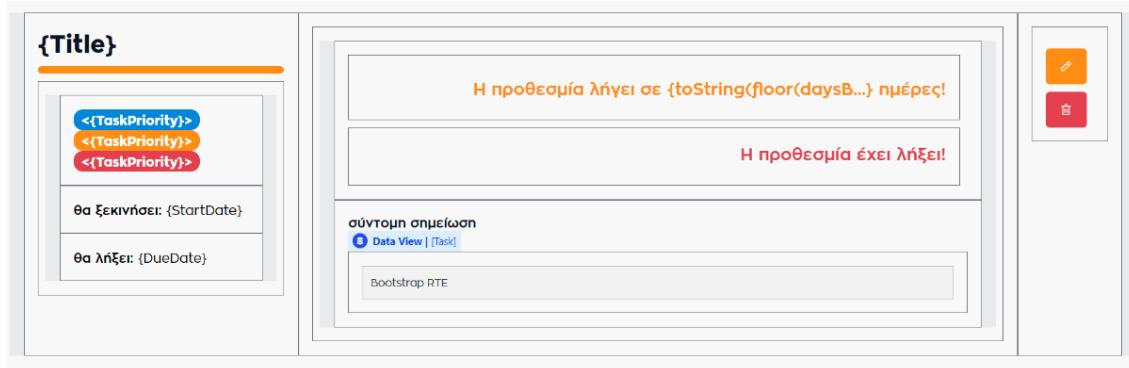
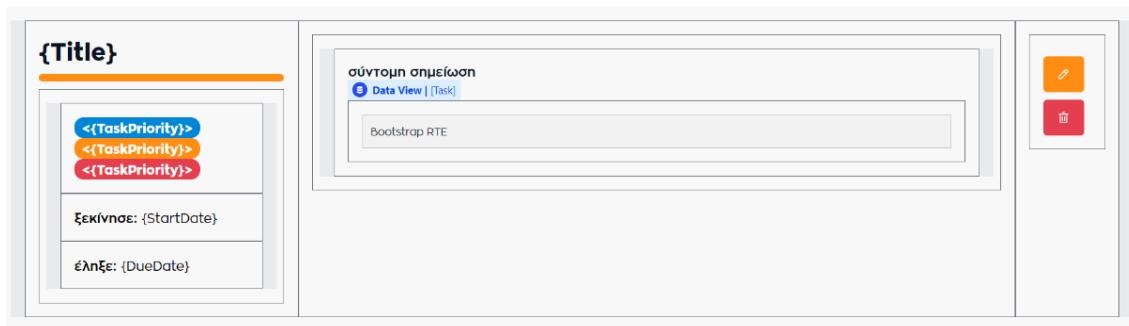
```
1 toString(floor(daysBetween($Task/DueDate, [%CurrentDateTime%])))
```

Το δεύτερο container εμφανίζεται το [%CurrentDateTime] είναι μικρότερο ή ίσο από το DueDate και το κουμπί του καλεί το microflow ChangeTaskStatus\_TaskDone.

Κάτω από το container υπάρχει ένα Data View με Data source το Task και το Bootstrap RTE widget. Το Rich Text Editor παρέχει στους χρήστες τη δυνατότητα δημιουργίας εμπλουτισμένου περιεχομένου, όπως έντονο (bold) και πλάγιο (italic) κείμενο. Το widget αποθηκεύει το κείμενο σε String μορφή.

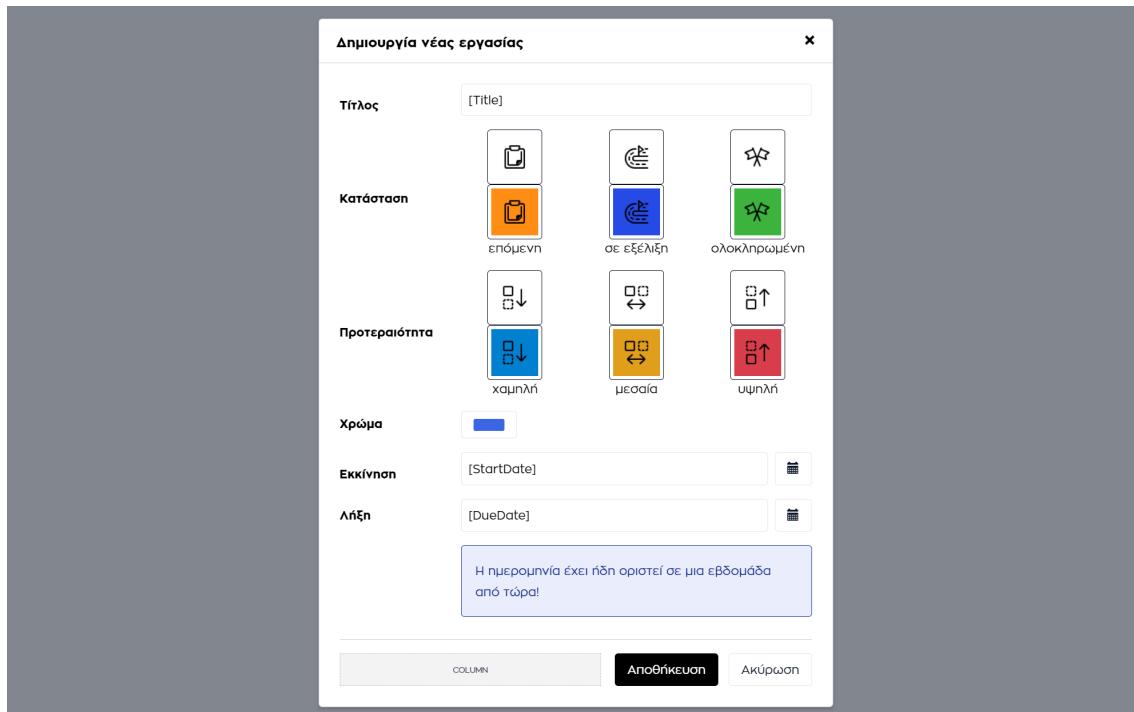
Στη δεξιά στήλη υπάρχουν τα κουμπιά επεξεργασίας και διαγραφής που καλούν τα microflows EditTask και DeleteTask (μετά από επιβεβαίωση) αντίστοιχα. Να σημειωθεί πως το κουμπί διαγραφής εμφανίζεται ανάλογα με την τιμή της Boolean ιδιότητας buttonQuickDelete του Student.

<sup>7</sup>Η έκφραση συγκρίνει τις ημέρες μεταξύ στην τρέχουσα μέρα και της ημερομηνίας λήξης της εργασίας, και επιστέψει True όταν η διαφορά είναι μικρότερη από μια εβδομάδα και η ημερομηνία λήξης ακόμη είναι στο μέλλον.

**TaskCard\_Overview\_ToDo****TaskCard\_Overview\_Done**

Τα Snippets TaskCard\_Overview\_ToDo και TaskCard\_Overview\_Done είναι παρόμοια με το TaskCard\_Overview\_Doing με τις απαραίτητες διαφοροποιήσεις.

Να σημειωθεί πως στο TaskCard\_Overview\_Done το Progress Bar πάντα είναι ολοκληρωμένο, ενώ το TaskCard\_Overview\_ToDo πάντα κενό, και επίσης πως στο δεύτερο δεν υπάρχει call-to-action κουμπί για την αλλαγή κατάστασης της εργασίας μετά την ημερομηνία λήξης, καθώς σε ένα σενάριο που ο χρήστης έχει προσθέσει μια εργασία ως To-Do και έχει τελειώσει η προθεσμία της πριν περάσει στην Doing κατάσταση, η προβλεπόμενη κίνηση θα είναι να τη διαγράψει.

**POPOUT\_Task\_NewEdit**

Η σελίδα καλείται στα microflows CreateNewTask\_ToDo, CreateNewTask\_Doing και CreateNewTask\_Done και χρησιμοποιείται για τη δημιουργία νέων εργασιών. Χρησιμοποιεί το Popout\_Layout layout του Atlas\_Core και έχει ως Parameters το Task και το PageHelper.

Περιλαμβάνει ένα Data View με ένα Layout Grid με τα απαραίτητα Text Boxes και Date Pickers για την εισαγωγή του τίτλου και των ημερομηνιών έναρξης και λήξης. Για την επιλογή του χρώματος χρησιμοποιείται το widget ColorPicker που αποθηκεύει το επιλεγμένο χρώμα στη μεταβλητή CalendarColor.

Για την επιλογή της κατάστασης και της προτεραιότητας έχουν δημιουργηθεί ένα σύνολο από ζεύγη κουμπιών, το ένα άχρωμο και το άλλο έγχρωμο. Ανάλογα με το ποιο είναι το TaskStatus και το TaskPriority εμφανίζεται και εξαφανίζεται το αντίστοιχο σύνολο κουμπιών. Ταυτόχρονα, οι λεζάντες κάτω από τα κουμπιά περιέχουν τη δυναμική κλάση που καθορίζεται από την έκφραση:

```
1 if $Task/TaskPriority = TaskManager.TaskPriority.Low then 'labelSelected'
2 else ''
```

Αντίστοιχα στο αρχείο Styling/web/custom-variables.scss έχει δημιουργηθεί η κλάση:

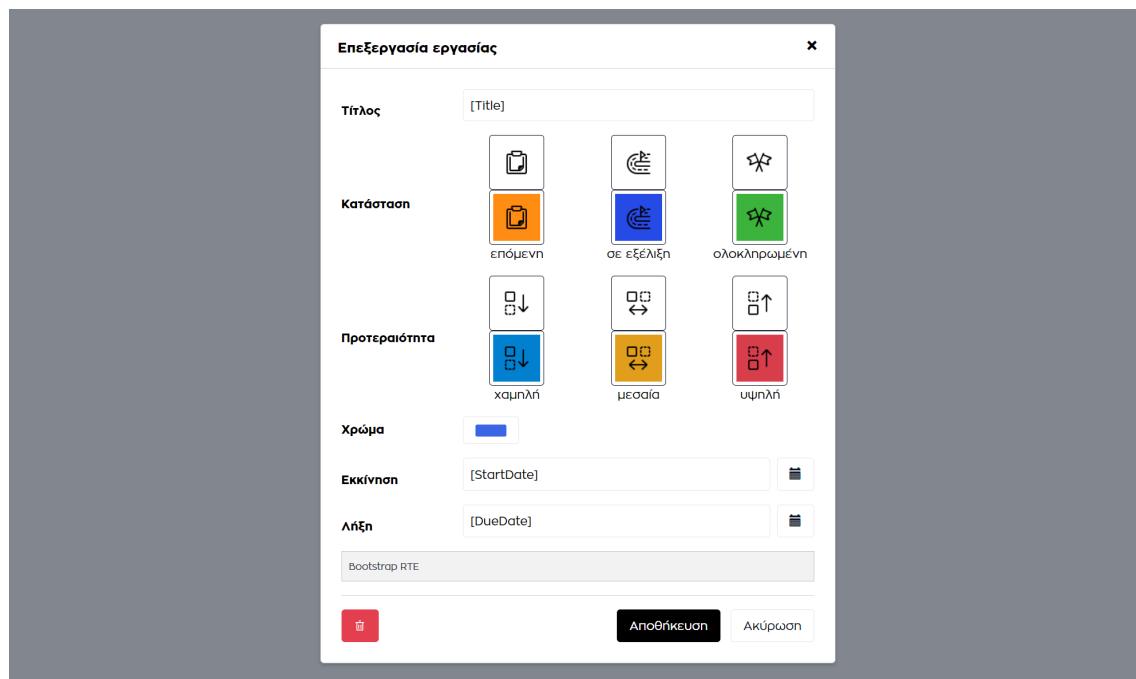
```
1 .labelSelected {
2     font-weight: bold;
3 }
```

Με αυτόν τον τρόπο επιτυγχάνεται η εμφάνιση της επιλεγμένης κατάστασης και προτεραιότητας με έντονη γραφή. Επιπλέον, το κάθε σύνολο από τα ζεύγη κουμπιών βρίσκεται τοποθετημένο σε ένα container, το οποίο αν πατηθεί καλεί τα microflow ChangeTaskStatus\_TaskToDo, ChangeTaskStatus\_TaskDoing, ChangeTaskStatus\_TaskDone και ChangeTaskPriority\_Low, ChangeTaskPriority\_Medium και ChangeTaskPriority\_High αντίστοιχα.

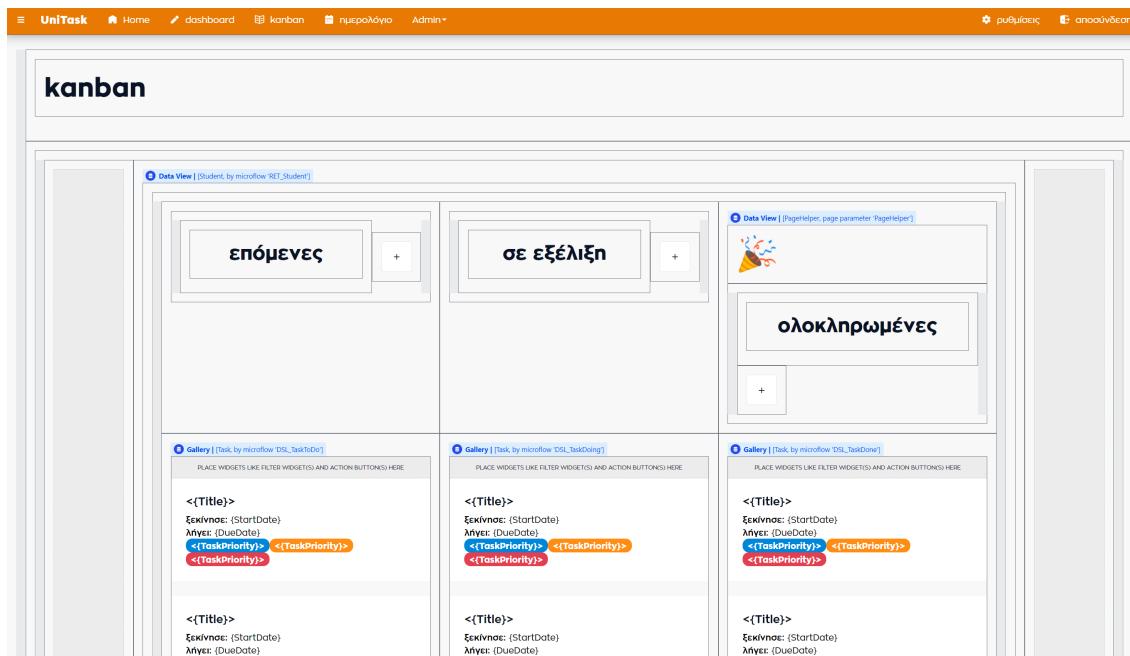
Τέλος, η εμφάνιση του μπλε παραθύρου που ενημερώνει ότι η ημερομηνία λήξης έχει προκαθοριστεί αυτόματα βασίζεται στην Boolean μεταβλητή CreationDateBubble του Task.

Για να αποθηκευτεί η νέα εργασία καλείται το microflow SaveTask.

#### POPOUT\_Task\_NewEdit\_Edit



Η σελίδα καλείται από το microflow Edit\_Task. Χρησιμοποιείται αντίστοιχη λογική με την POPOUT\_Task\_NewEdit με τη διαφορά ότι δεν υπάρχει επεξηγηματικό παράθυρο για την ημερομηνία λήξης, υπάρχει το widget Bootstrap RTE για την επεξεργασία της ιδιότητας Notes και επιπλέον υπάρχει το κουμπί διαγραφής της εργασίας που καλεί το microflow DeleteTask μετά από επιβεβαίωση.

**PAGE\_Tasks\_Kanban**

Η σελίδα χρησιμοποιείται ως ένας εναλλακτικός τρόπος εμφάνισης των εργασιών του χρήστη σε έναν Kanban πίνακα. Χρησιμοποιεί το UniTask\_TopBar layout του UniTaskDesignSystem module και έχει ως Parameter το PageHelper.

Η σελίδα αποτελείται από ένα Layout Grid με 3 στήλες, με τις δύο ακραίες να αποτελούν negative space. Η κεντρική στήλη περιέχει ένα Layout Grid με 3 στήλες. Το πρώτο row του περιλαμβάνει τις επικεφαλίδες του πίνακα μαζί με κουμπιά για την προσθήκη νέας εργασίας. Κάθε κουμπί είναι προσαρμοσμένο να δημιουργεί μια εργασία που αντιστοιχεί στην κατάσταση της επικεφαλίδας, καλώντας τα microflows CreateNewTask\_ToDo, CreateNewTask\_Doing και CreateNewTask\_Done αντίστοιχα. Το δεύτερο row περιλαμβάνει Galleries με Data sources τα DSL\_TaskToDo, DSL\_TaskDoing και DSL\_TaskDone που δημιουργούν τις κάρτες, που αν πατηθούν καλούνται το microflow EditTask. Τέλος, περιλαμβάνεται το Confetti widget, όπως και στην Dashboard σελίδα.

**TaskCard\_Kanban**

```
<{Title}>
Σεκίνησε: {StartDate}
Λήγει: {DueDate} <{TaskPriority}> <{TaskPriority}> <{TaskPriority}>
```

Πρόκειται για ένα Snippet που χρησιμοποιείται για την εμφάνιση των καρτών με τις εργασίες στον πίνακα Kanban. Έχει ως Parameters τα Task και Student.

Αποτελείται από ένα Layout Grid με τον τίτλο, τις ημερομηνίες έναρξης και λή-

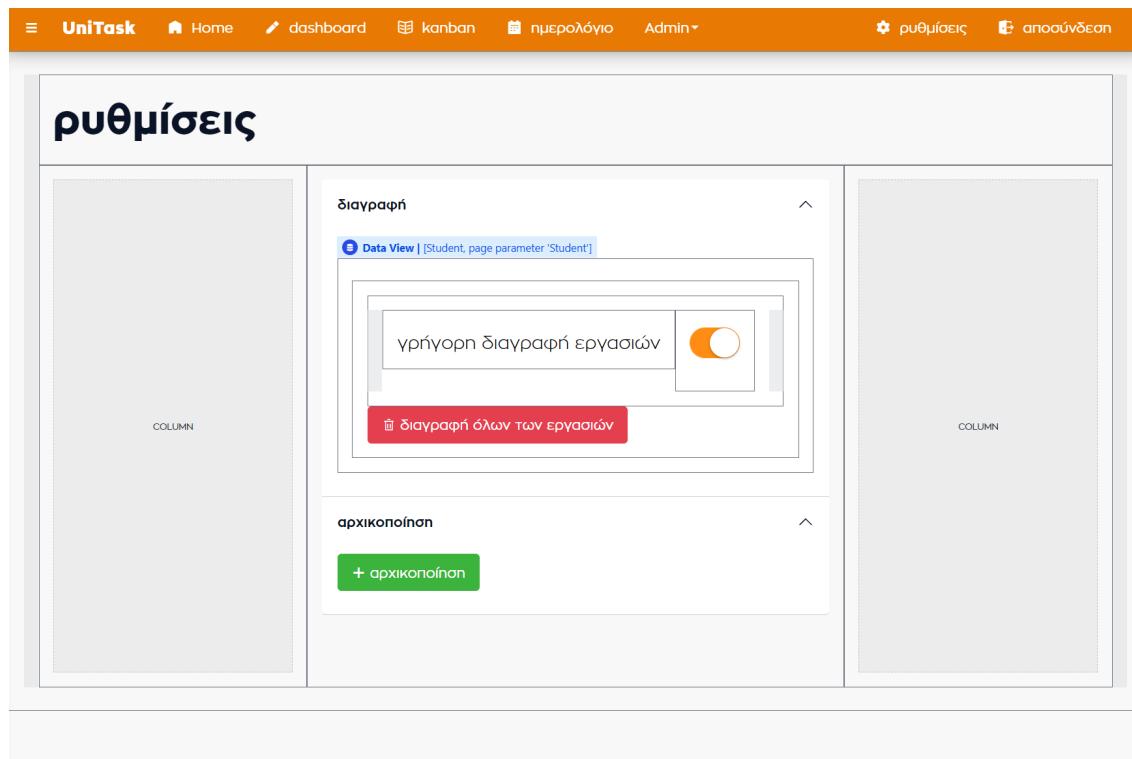
Έντονος και Badges για την προτεραιότητα της εργασίας, με παρόμοιο τρόπο όπως στα Snippets του Dashboard.

### PAGE\_Calendar

The screenshot displays the UniTask application interface. At the top, there's a navigation bar with links for Home, dashboard, kanban, πινακίδα, Admin, and user status. Below the navigation is a title 'ημερολόγιο'. The main area features a calendar for January 2025. The calendar grid shows various events: several blue bars representing 'Leave' days, a red bar for 'BD' (Birthday) on the 16th, and grey bars for 'Bank Holiday' days. To the right of the calendar, a 'Timeline' section provides a detailed view of specific tasks. Each task entry consists of two columns: 'Title' and 'TaskStatus' on the left, and 'StartDate - DueDate' on the right. The tasks listed are identical to those shown in the calendar grid.

Η σελίδα χρησιμοποιείται για την εμφάνιση των εργασιών του χρήστη σε έναν ημερολόγιο. Χρησιμοποιεί το UniTask\_TopBar layout του UniTaskDesignSystem module και έχει ως Parameter το Task και PageHelper.

To Calendar widget είναι τοποθετημένο μέσα σε ένα Data View με Data source το Task. Έτσι καθορίζεται πως το Event entity, δηλαδή ο τύπος αντικειμένου που θα εμφανίζεται στο ημερολόγιο, είναι το Task. Το ημερολόγιο γίνεται populate μέσω του microflow ShowTasks\_Calendar και στα Properties του Widget επιλέγονται οι ιδιότητες Title, StartDate, DueDate και CalendarColor του Task για να καθοριστεί η εμφάνισή του στο ημερολόγιο. Όσον αφορά το Timeline στα δεξιά του, έχει επιλεχθεί να εμφανίζονται οι εργασίες ομαδοποιημένες βάσει της ημερομηνίας έναρξής τους και μάλιστα να μπορεί να γίνει επεξεργασία τους αν γίνει κλικ πάνω τους.

**PAGE\_Settings**

Η σελίδα χρησιμοποιείται για την εμφάνιση των ρυθμίσεων του χρήστη. Χρησιμοποιεί το UniTask\_TopBar layout του UniTaskDesignSystem module και έχει ως Parameters το Student και το PageHelper.

Οι ρυθμίσεις βρίσκονται τοποθετημένες σε ένα Accordion widget με δύο ομάδες: διαγραφή και αρχικοποίηση. Στη διαγραφή περιλαμβάνεται ένα Data View με Data source το Student. Εσωτερικά του υπάρχει ένα Switch widget που κάνει toggle την Boolean ιδιότητα buttonQuickDelete του Student και ένα κουμπί (διαγραφή όλων των εργασιών) που καλεί το microflow DeleteAllTasks μετά από επιβεβαίωση. Στην αρχικοποίηση περιλαμβάνεται το κουμπί (αρχικοποίηση) που καλεί το microflow InitializeTasks μετά από επιβεβαίωση.

#### 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](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](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, Μαρ. 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, Construction, 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](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](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](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://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). [https://en.wikipedia.org/wiki/Rapid\\_application\\_development](https://en.wikipedia.org/wiki/Rapid_application_development). [Accessed 29-12-2024].
- [34] Thomas Q Reefe. *Lukasa: A Luba Memory Device*. doi: [doi:10.2307/3335144](https://doi.org/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](https://doi.org/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](https://doi.org/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). <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](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.