

Περιεχόμενα

1	Mendix	1
1.1	Τι είναι το Mendix;	1
1.2	To Mendix Studio	2
1.2.1	Περιβάλλον ανάπτυξης	2
1.2.1.1	Επιλογές για deployment	3
1.3	Δομή εφαρμογών του Mendix	4
1.3.1	Modules	5
1.3.1.1	To module App	5
1.3.1.2	To module System	8
1.3.2	Έγγραφα	8
1.3.3	Σελίδες	8
1.3.3.1	Layout	9
1.3.3.2	Widgets	9
1.3.3.3	Εμφάνιση σελίδων	11
1.3.3.4	Παράδειγμα δομής σελίδας	12
1.3.4	Microflows	13
1.3.5	Εκφράσεις	15
1.3.6	Domain Model	16
1.3.6.1	Οντότητες	17

Κεφάλαιο 1

Mendix

Έχοντας πλέον μια καλή εικόνα για τον ορισμό του χαμηλού κώδικα και των Πλατφόρμων Ανάπτυξης Λογισμικού σε Low-Code, στο παρόν κεφάλαιο, θα επικεντρωθούμε σε μια από αυτές τις πλατφόρμες, το Mendix, η οποία αποτέλεσε το βασικό εργαλείο για την υλοποίηση της εφαρμογής που αναπτύχθηκε στο πλαίσιο της παρούσας διπλωματικής εργασίας. Σε αυτό το κεφάλαιο, θα...

1.1 Τι είναι το Mendix;

Το Mendix αποτελεί μία από τις πιο διαδεδομένες πλατφόρμες ανάπτυξης λογισμικού που βασίζεται σε χαμηλό κώδικα. Ιδρύθηκε το 2005 στο Ρότερνταμ της Ολλανδίας με στόχο να παρέχει στους επιχειρηματίες και τους οργανισμούς τη δυνατότητα να αναπτύσσουν, να προσαρμόζουν και να διαχειρίζονται εφαρμογές αποδοτικά με χαμηλό κόστος. Το Mendix περιλαμβάνει όλα τα οφέλη και τα χαρακτηριστικά των LCDP που έχουν περιγραφεί στην ενότητα ??, συμπεριλαμβάνοντας γραφικό περιβάλλον με WYSIWYG GUI σχεδιαστή, drag-and-drop εργαλεία και έτοιμες βιβλιοθήκες, τη χρήση domain models, το εύκολο deployment της εφαρμογής στο cloud, version control μέσω Git, συνεργασία χρησιμοποιώντας Agile μεθοδολογία και άλλα.

Το 2018, το Mendix εξαγοράστηκε από τη Siemens, τη μεγαλύτερη βιομηχανική κατασκευαστική εταιρεία στην Ευρώπη, γεγονός που επέφερε σημαντικές εξελίξεις στην πλατφόρμα. Η συγχώνευση αυτή επέτρεψε την ενσωμάτωση προηγμένων βιομηχανικών και IoT (Internet of Things) λύσεων, ενισχύοντας τη θέση του Mendix στην αγορά των λογισμικών σχεδιασμένων για επιχειρήσεις. Έτσι, το Mendix αποτελεί μία από τις πιο ισχυρές και ευέλικτες λύσεις στην αγορά του low-code προγραμματισμού, προσφέροντας αποτελεσματικότητα, ταχύτητα και καινοτομία στην ανάπτυξη λογισμικού, ενώ παράλληλα ενσωματώνει τις πιο σύγχρονες τεχνολογίες για να καλύψει τις ανάγκες επιχειρήσεων που επιθυμούν να παραμείνουν ανταγωνιστικές στην ψηφιακή εποχή. [2]

Η Gartner, μία από τις μεγαλύτερες εταιρείες έρευνας και συμβουλευτικής στον κλάδο της τεχνολογίας, χαρακτηρίζει το Mendix ως ηγέτη στην αγορά των πλατφορμών

ανάπτυξης λογισμικού για 8 συνεχόμενα χρόνια, (βλ. Σχήμα 1.1). Η κατάταξη αυτή αποδεικνύει την ικανότητα του Mendix να παρέχει λύσεις υψηλής ποιότητας και αξίας στους πελάτες του, καθώς και την ικανότητά του να προσαρμόζεται στις ανάγκες της αγοράς και να προσφέρει συνεχώς καινοτόμες λύσεις. [1] Για αυτούς τους λόγους έχει προτιμηθεί για την υλοποίηση της εφαρμογής που θα παρουσιαστεί στο επόμενο κεφάλαιο.



Σχήμα 1.1: Τεταρτημόριο της Gartner με πλατφόρμες ανάπτυξης λογισμικού [1]

1.2 Το Mendix Studio

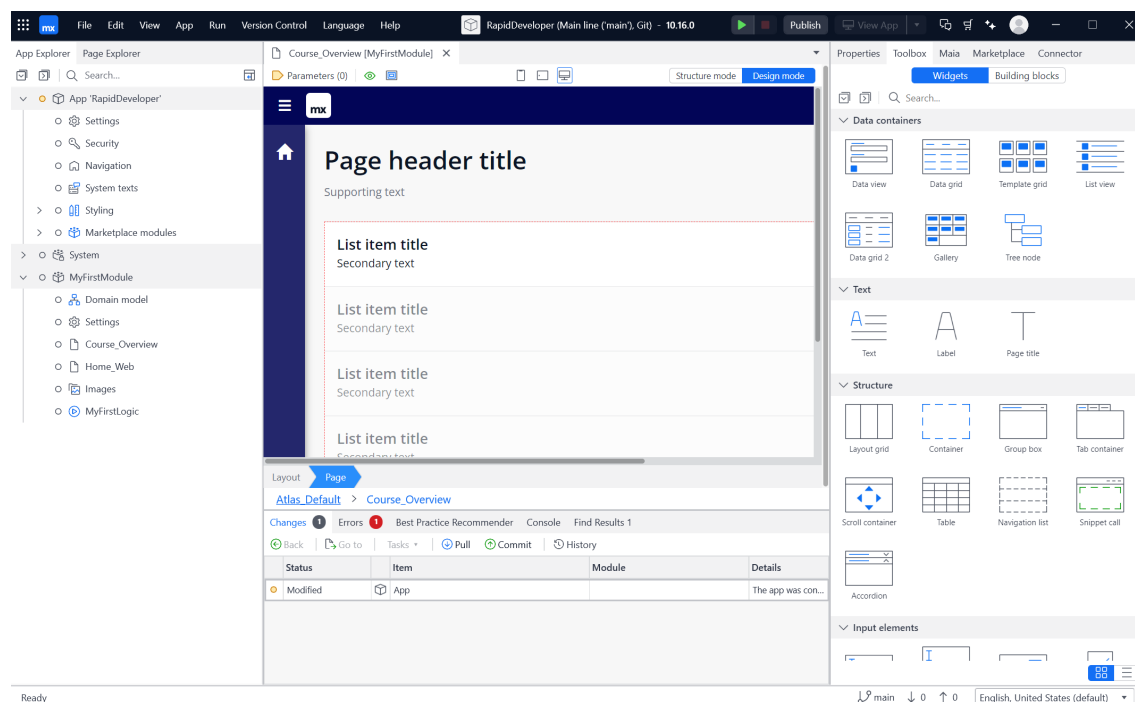
Το Mendix Studio Pro αποτελεί το βασικό εργαλείο ανάπτυξης εφαρμογών της πλατφόρμας Mendix, προσφέροντας στους χρήστες τη δυνατότητα να δημιουργήσουν, να προσαρμόσουν, να δοκιμάσουν και να αναπτύξουν εφαρμογές.

1.2.1 Περιβάλλον ανάπτυξης

Στο 1.2 παρουσιάζεται το γραφικό περιβάλλον του Mendix Studio Pro με ανοιχτή την εφαρμογή RapidDeveloper¹.

Μπορούμε να διαχωρίσουμε το γραφικό περιβάλλον σε τέσσερα μέρη. Η μαύρη μπάρα στο πάνω μέρος περιλαμβάνει το βασικό μενού της πλατφόρμας, το όνομα της εφαρμογής που αναπτύσσουμε, κουμπιά τα οποία επιτρέπουν είτε την τοπική εκτέλεση της εφαρμογής μέσω localhost ή τη διάθεσή της στο cloud (βλ. ενότητα

¹Η εφαρμογή RapidDeveloper δημιουργήθηκε ως αποτέλεσμα των μαθημάτων (crash courses) που προσφέρονται μέσω του Mendix Academy. Αυτά τα μαθήματα έχουν σχεδιαστεί για να παρέχουν στους χρήστες μια γρήγορη και πρακτική εισαγωγή στις βασικές δυνατότητες της πλατφόρμας Mendix, επιτρέποντάς τους να εξοικειωθούν με τη διαδικασία ανάπτυξης εφαρμογών σε περιβάλλον low-code.



Σχήμα 1.2: Στιγμιότυπο του Mendix Studio Pro.

1.2.1.1) και σύνδεσμοι που οδηγούν στο Mendix προφίλ του χρήστη, στο Marketplace κ.α.

Στο κεντρικό τμήμα της οθόνης βρίσκεται το *Working Area*, ένας WYSIWYG σχεδιαστής όπου μπορούμε να προβάλλουμε και να επεξεργαστούμε τις σελίδες της εφαρμογής μας. Μια σελίδα μπορεί να εμφανιστεί στο *Working Area* με διάφορους τρόπους, οι οποίοι θα αναλυθούν στην ενότητα 1.3.3.3.

Στην αριστερή πλευρά, εντοπίζουμε τον *App Explorer*, ο οποίος περιλαμβάνει τη δομή φακέλων και αρχείων της εφαρμογής, καθώς και τον *Page Explorer*, που καταγράφει όλα τα στοιχεία που έχουν χρησιμοποιηθεί στη σελίδα της εφαρμογής που είναι ανοιχτή. Στη δεξιά πλευρά, βρίσκεται το *Properties Panel*, όπου εμφανίζονται όλες οι ρυθμίσεις και οι παράμετροι του στοιχείου ή της σελίδας που έχουμε επιλεγμένη, και το *Toolbox*, το οποίο περιέχει ένα σύνολο από προκατασκευασμένα στοιχεία που μπορούν να εισαχθούν στη σελίδα. Στο κάτω μέρος εμφανίζονται panels με τις αλλαγές που πραγματοποιούμε ανά commit, τα σφάλματα αν τυχόν υπάρχουν, logs, κονσόλα και άλλα. Μπορούμε σε οποιαδήποτε πεδίο να προσθέσουμε ή να αφαιρέσουμε Panels από το Μενού → View. [4]

1.2.1.1 Επιλογές για deployment


Το Mendix παρέχει διάφορες επιλογές για το deployment των εφαρμογών που αναπτύσσονται στην πλατφόρμα.

Με το Mendix Free μπορούμε να κάνουμε deploy την εφαρμογή στο διαδίκτυο


σε ένα URL της μορφής <όνομα εφαρμογής>-sandbox.mxapps.io, και είναι αυτό που έχει χρησιμοποιηθεί για την υλοποίηση της εφαρμογής μας στο κεφάλαιο ???. Το Mendix Free είναι κατάλληλο για την ανάπτυξη μικρών εφαρμογών και την εξοικείωση με την πλατφόρμα, αλλά δεν προσφέρει την απαιτούμενη ασφάλεια και ευελιξία για την ανάπτυξη μεγάλων επιχειρησιακών εφαρμογών καθώς οι εφαρμογές παύουν να λειτουργούν μετά από λίγες ώρες αδράνειας (sleep mode), δεν μπορούν να κλιμακωθούν, υπάρχει όριο στην υπολογιστική ισχύ και το μέγεθος της βάσης δεδομένων, δεν τρέχουν προγραμματιζόμενα γεγονότα, δεν υποστηρίζονται custom domains κ.α. Παρόλα αυτά είναι μια πολύ καλή επιλογή που εξυπηρετεί τις ανάγκες των αρχάριων χρηστών και μικρών εφαρμογών. Για χρήστες ή επιχειρήσεις με αυξανόμενες ανάγκες, το Mendix προσφέρει λύσεις επί πληρωμή που άρουν τους περιορισμούς που αναφέρθηκαν νωρίτερα. [3]

Το Mendix προφίλ κάθε χρήστη περιλαμβάνει dashboards για κάθε εφαρμογή που αναπτύσσει όπου περιλαμβάνονται ρυθμίσεις και πληροφορίες όσον αφορά το deployment.

Environments






 **Free Version.** You are using a [Free App](#) in Mendix Cloud. Try our Basic Package for more resources and control! Already have cloud resources? [Unlink your app.](#) [Learn More](#)

Deploy **Permissions**

 Your app "UniTask" is deployed in Mendix Cloud EU: AWS Ireland (Free).
Currently running version 1.0.0.811cf6c1

[View Live Log](#) [Show Latest Build Output](#) [Show Debugger Information](#)

Activity

Activity	Date ▼
 Deployed model version 'main-1.0.0.811cf6c1.mda' to your sandbox by Alexandros Xiarchos.	Mon, 06 Jan 2025 22:00:12 GMT+2
 Deployed model version 'main-1.0.0.a72a0c7b.mda' to your sandbox by Alexandros Xiarchos.	Sun, 29 Dec 2024 17:07:04 GMT+2
 Deployed model version 'main-1.0.0.4265c558.mda' to your sandbox by Alexandros Xiarchos.	Mon, 23 Dec 2024 16:48:04 GMT+2
 Deployed model version 'main-1.0.0.8abbe264.mda' to your sandbox by Alexandros Xiarchos.	Mon, 23 Dec 2024 15:44:55 GMT+2
 Deployed model version 'main-1.0.0.39856037.mda' to your sandbox by Alexandros Xiarchos.	Thu, 28 Nov 2024 20:17:00 GMT+2

Showing 1 to 5 of 56 Activities

< 1 2 3 4 5 ... 12 >

Σχήμα 1.3: Σελίδα Environments της εφαρμογής UniTask (κεφ. ??)

1.3 Δομή εφαρμογών του Mendix

Μια εφαρμογή στο Mendix απαρτίζεται από διαφορετικά έγγραφα και modules. Τα modules επιτρέπουν τον διαχωρισμό της εφαρμογής σε αυτόνομα λειτουργικά κομμάτια. Ο τρόπος με τον οποίο πραγματοποιείται ο διαχωρισμός εξαρτάται από την κρίση και τη σχεδιαστική προσέγγιση του μηχανικού λογισμικού.

1.3.1 Modules

Μια εφαρμογή αποτελείται από ένα App module, ένα System module, από modules που δημιουργούνται από τους χρήστες, από modules από το marketplace του Mendix ή από modules που καθορίζουν την εμφάνιση της εφαρμογής (UI resources modules). Τα modules του marketplace προσφέρουν έτοιμες λειτουργικότητες κατασκευασμένες από τρίτους ενώ τα UI resources modules εμφανίζονται με πράσινο χρώμα και περιλαμβάνουν πρότυπα σελίδων (page templates) και δομικά στοιχεία (building blocks).

Για παράδειγμα, στο App Explorer της εφαρμογής RapidDeveloper (σχήμα 1.2) παρατηρούμε πως εμφανίζονται τρία διαφορετικά modules: το module App, το module System και το module MyFirstModule. Τα δύο πρώτα είναι modules που δημιουργούνται αυτόματα κατά τη δημιουργία μιας εφαρμογής, ενώ το τρίτο είναι ένα module που δημιουργήθηκε από τον χρήστη.

Κάθε module περιλαμβάνει ένα domain model που ορίζει τη δομή των δεδομένων του. Θα αναφερθούμε αναλυτικά στο domain model στην ενότητα 1.3.6. Πέρα από το domain models, κάθε module περιλαμβάνει παράθυρα για τις Ρυθμίσεις (Settings) του module και την Ασφάλεια (Security).

Στις **Ρυθμίσεις** επιτρέπεται η εξαγωγή του Module ως *App Module* με όλο το πηγαίο κώδικα του ή ως *Add-on Module* με σκοπό να χρησιμοποιηθεί από άλλους χρήστες, και επίσης εκεί μπορεί να γίνει προσθήκη Java Dependancies στο module. Στην **Ασφάλεια** μπορεί να καθοριστεί η πρόσβαση όλων των ρόλων χρηστών για κάθε σελίδα, οντότητα ή microflow που υπάρχει στο συγκεκριμένο module.

1.3.1.1 To module App

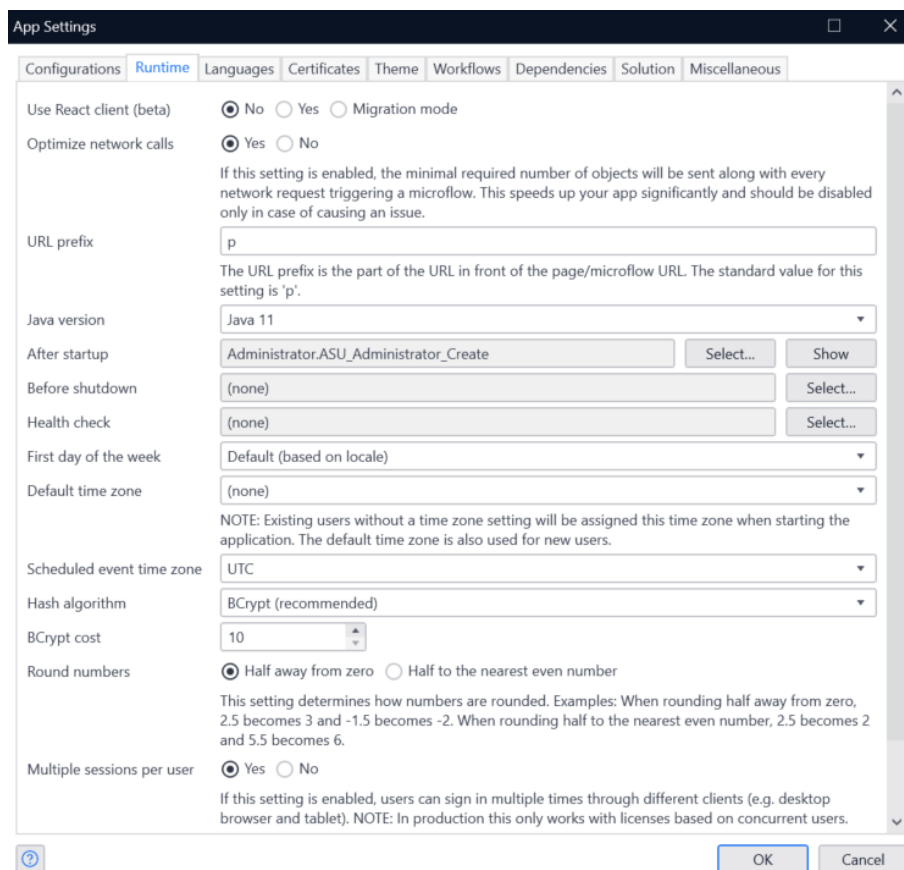
Κάτα τη δημιουργία μιας νέας εφαρμογής, το Mendix διαθέτει ένα σύνολο από προεγκατεστημένες σελίδες με έτοιμο σχεδιασμό και λειτουργικότητα. Τα αρχεία αυτών των σελίδων βρίσκονται στο module App. Το module App πέρα από τα παράθυρα των Ρυθμίσεων και Ασφάλειας (τα οποία έτσι και αλλιώς υπάρχουν σε όλα τα modules) επιπλέον περιλαμβάνει την Πλοήγηση (Navigation) και τα Κείμενα Συστήματος (System Texts)². Επιπλέον, περιλαμβάνει ένα φάκελο Styling με .js και .css αρχεία τα οποία χρησιμοποιούνται για το styling της εφαρμογής³ και έναν φάκελο Marketplace modules το οποίο περιλαμβάνει εξωτερικά modules που μπορούν να προστεθούν μέσω του Mendix Marketplace.

Το παράθυρο **Ρυθμίσεων** του App (βλ. σχήμα 1.4) διαφέρει σε σχέση με τα υπόλοι-

²Στο έγγραφο με τα Κείμενα Συστήματος μπορεί να γίνει μετάφραση των μηνυμάτων που παράγονται από τον διακομιστή κατά την εκτέλεση μιας εφαρμογής (για παράδειγμα “Password too short”).

³Στα συγκεκριμένα αρχεία μπορούν να επεξεργαστούν μεταβλητές που αφορούν τη σχεδίαση του UI resources module Atlas, το οποίο χρησιμοποιείται ως κεντρικό theme για τις προεγκατεστημένες σελίδες του Mendix. Το Atlas για παράδειγμα περιλαμβάνει έτοιμα color schemes (primary, success, warning, danger, info) τα οποία χρησιμοποιούνται σε όλα τα widgets των σελιδών, όπως επίσης γραμματοσειρές, spacings κτλ. Τα αρχεία λοιπόν έναν από τους τρόπους προσαρμογής της προεπιλεγμένης σχεδίασης του Atlas. Εναλλακτικός τρόπος είναι η δημιουργία ενός custom UI resources module.

πα modules, καθώς αυτό περιλαμβάνει παραμετροποιήσεις για το runtime περιβάλλον της εφαρμογής, το theme που χρησιμοποιείται, την επιλογή συγκεκριμένων ενεργειών πριν αρχικοποιηθεί η εφαρμογή κατά την εκκίνησή της, επιλογή συγκεκριμένου αλγόριθμου κρυπτογράφησης για το Hashed String τύπο δεδομένων, καθορισμός γλώσσας και άλλα.



Σχήμα 1.4: Παράθυρο Settings του App

Το παράθυρο **Ασφάλεια** του App (βλ. σχήμα 1.5) το οποίο και αυτό διαφέρει σε σχέση με τα αντίστοιχα παράθυρα των υπόλοιπων modules, χρησιμοποιείται για να τροποποιηθεί το *επίπεδο ασφάλειας* (Security level) της εφαρμογής. Συγκεκριμένα μπορεί να επιλεγεί ώστε οι χρήστες να μη χρειάζεται να συνδεθούν για να έχουν πρόσβαση στην εφαρμογή (Security level = Off), να χρειάζεται να συνδεθούν για να ώστε να μπορέσουν να χρησιμοποιήσουν φόρμες, microflows κτλ (Security level = Prototype/demo), ή να χρειάζεται να συνδεθούν ώστε να έχουν πρόσβαση στην εφαρμογή καθολικά (Security level = Production).

Επίσης, μπορούν να οριστούν *ρόλοι χρηστών* όπως για παράδειγμα Administrator, User ή Guest. Κατ' αυτόν τον τρόπο καθίσταται δυνατό να διαχωριστούν τα δικαιώματα πρόσβασης των modules ανάλογα με το ποιος είναι ο ρόλος του χρήστη. Για παράδειγμα, ένας Administrator χρήστης μπορεί να έχει πλήρη πρόσβαση στα modules

App Security

Security level

Security level ☐ Off ☐ Prototype / demo ☒ Production

Full security is applied. Configure administrator and anonymous access and define user roles and security for forms, microflows, entities, and reports.

Check security ☒ Yes ☐ No

If there are no other errors, Mendix Studio Pro checks per user role whether forms that are accessible for a certain role only refer to attributes and associations that are accessible for that same role. This assumes that each user role is independent and that users do not need two or more roles to access functionality in your application.

Strict page URL checking ☒ Yes ☐ No

Perform security checks against pages only accessible by page URLs.

App status ● Incomplete

Module status | User roles | Administrator | Demo users | Anonymous users | Password policy

Edit module security

Module	Page access	REST access	OData access	Microflow access	Nanoflow access	Entity access	Data set access
MyFirstModule	Incomplete	Complete	Complete	Complete	Complete	Complete	Complete
Atlas_Web_Conte...	Complete	Complete	Complete	Complete	Complete	Incomplete	Complete

OK Cancel

Σχήμα 1.5: Παράθυρο Security του App

και στις σελίδες της εφαρμογής, ενώ ένας Guest μπορεί να έχει περιορισμένη προβολή.

Επιπλέον, στην Ασφάλεια μπορούν να οριστούν τα διαπιστευτήρια του διαχειριστή (Administrator) της εφαρμογής (ώστε να μπορέσει να γίνει η πρώτη σύνδεση κατά το deployment), να οριστούν demo χρήστες (ώστε να δοκιμαστούν οι ρόλοι χρηστών) ή ανώνυμοι χρήστες (οι οποίοι έχουν πρόσβαση στην εφαρμογή χωρίς να συνδεθούν) και να ρυθμιστούν κανόνες για τους κωδικούς πρόσβασης.

Responsive web

General

Application title Edit...

Application icon Select... Show

Home pages

Default home page Select... Show

Role-based home pages Edit...

Fallback page Select...

Progressive Web App

☐ Publish as a Progressive Web App

☐ Allow 'Add to Home screen' prompt

☐ Preload static resources

To use all Progressive Web App features, the app needs to be published to the cloud.

[Read more about Progressive Web Apps](#)

Menu

Caption	Action	User Roles
Home	Open page 'MyFirstModule.Home_Web'	

Σχήμα 1.6: Το έγγραφο Navigation.

Η **Πλοήγηση** του module App (βλ. σχήμα 1.6) εμφανίζει το σύνολο των σελίδων του κεντρικού μενού της εφαρμογής. Εκεί μπορεί να γίνει η επεξεργασία του μενού με την προσθήκη στοιχείων ή και υποστοιχείων σε στοιχεία. Επίσης, περιλαμβάνονται διαφορετικές προβολές ανάλογα με τον εκάστοτε ρόλο χρήστη, στις οποίες εμφανίζονται μόνο οι σελίδες που είναι προσβάσιμες σε κάθε ρόλο. Στην Πλοήγηση επίσης μπορεί να διαμορφωθεί το όνομα και το εικονίδιο της εφαρμογής όπως επίσης και να οριστεί και η αρχική σελίδα (home page) της, η οποία μάλιστα μπορεί να διαμορφωθεί ώστε να είναι διαφορετική για εκάστοτε ρόλο χρήστη. [4]

1.3.1.2 To module System

Το module System είναι ένα προεγκατεστημένο module που περιλαμβάνει τη βασική λειτουργικότητα η οποία χρησιμοποιείται στις ήδη κατασκευασμένες σελίδες και microflows του App. Το συγκεκριμένο module δεν μπορεί να επεξεργαστεί από τον χρήστη, αλλά μπορούν με βάση αυτό να προστεθούν συσχετίσεις (associations) ή γενικεύσεις (generalizations) τα modules τα οποία κατασκευάζονται από τους χρήστες. Για παράδειγμα, μπορεί να δημιουργηθεί μια οντότητα Πελάτης η οποία θα συσχετίζεται με την οντότητα User του module System και έτσι θα κληρονομεί τις ρυθμίσεις ασφαλείας του. [5]

1.3.2 Έγγραφα

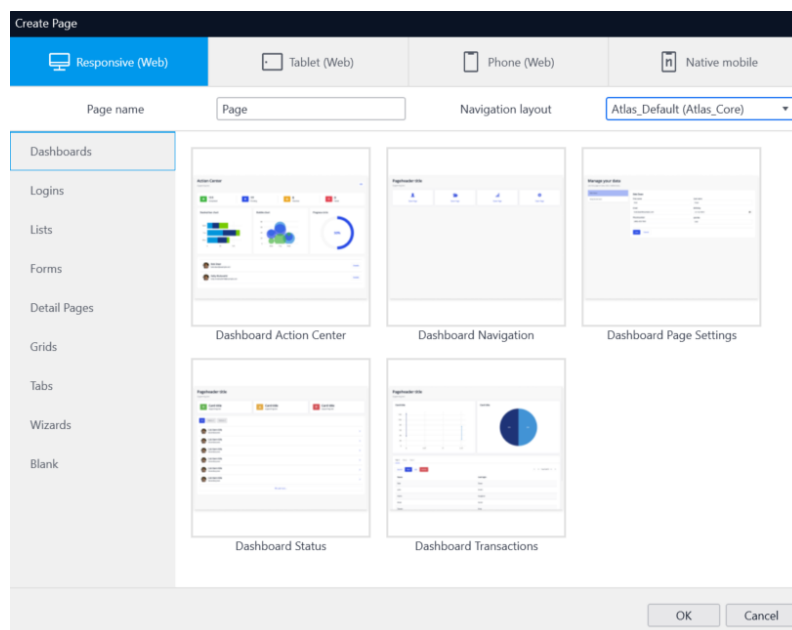
Η Πλοήγηση είναι ένα έγγραφο της εφαρμογής. Άλλα παραδείγματα εγγράφων είναι οι Σελίδες (Pages) (βλ. ενότητα 1.3.3), τα Microflows (βλ. ενότητα 1.3.4) και τα Enumerations⁴.

1.3.3 Σελίδες

Η σελίδα είναι ο κεντρικός τρόπος αλληλεπίδρασης του χρήστη με την εφαρμογή. Η δημιουργία μιας σελίδας μπορεί να βασιστεί σε πόρους που προέρχονται από έγγραφα, όπως οι Εικόνες (Images), τα Layouts τα οποία καθορίζουν τη διάταξη της σελίδας, τα Μενού (Menus) που διαμορφώνουν την πλοήγηση, ή τα Snippets, τα οποία αποτελούν επαναχρησιμοποιήσιμα τμήματα διεπαφής.⁵

⁴Τα enumerations (κατάλογος, απαρίθμηση) καθορίζουν μια λίστα από προκαθορισμένες τιμές. Για παράδειγμα, ένα enumeration μπορεί να χρησιμοποιηθεί για τον καθορισμό της κατάστασης μιας εργασίας ως To do, Done ή Doing.

⁵Το Mendix είναι μια Εφαρμογή Μίας Σελίδας (Single-page application – SPA) που σημαίνει ότι όλη η αλληλεπίδραση πραγματοποιείται σε μία μόνο καρτέλα ή παράθυρο του προγράμματος περιήγησης που φορτώνεται μία φορά και στη συνέχεια ενημερώνεται δυναμικά χωρίς να χρειάζεται να φορτωθεί ξανά. Ως αποτέλεσμα, δεν είναι δυνατή η ανοίγματος νέων σελίδων σε διαφορετική καρτέλα ή παράθυρο.



Σχήμα 1.7: Το παράθυρο δημιουργίας μιας νέας σελίδας.

1.3.3.1 Layout

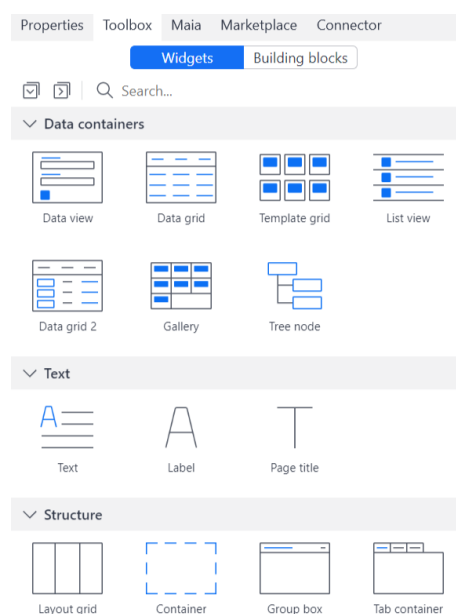
Κάθε σελίδα στο Mendix βασίζεται σε ένα προκαθορισμένο layout, το οποίο καθορίζει βασικές ιδιότητες της σελίδας, όπως το μήκος, το πλάτος ή για παράδειγμα αν πρόκειται για αναδυόμενη (popup) σελίδα. Επιπλέον, τα layouts επιτρέπουν τον ορισμό στατικών τμημάτων, όπως ένα header ή ένα μενού, που παραμένουν σταθερά σε όλες τις σελίδες που τα χρησιμοποιούν. Για παράδειγμα, θα μπορούσαν να δημιουργηθούν δύο layouts όπου το ένα θα έχει το μενού πλοήγησης ως μπάρα στο πάνω μέρος της οθόνης και το άλλο κάθετη στα αριστερά. Το Mendix παρέχει επίσης έτοιμα πρότυπα σελίδων (Page Templates), τα οποία διευκολύνουν τη γρήγορη και απλή δημιουργία σελίδων με προκαθορισμένη δομή και σχεδίαση.

Το σχήμα 1.7 απεικονίζει το παράθυρο δημιουργίας νέας σελίδας, όπου ο χρήστης μπορεί να επιλέξει είτε από τα έτοιμα πρότυπα είτε να δημιουργήσει μια κενή σελίδα. Εδώ δίνεται επίσης η δυνατότητα επιλογής του layout (Navigation layout) της σελίδας, το οποίο μπορεί να είναι είτε ένα από τα προεγκατεστημένα layouts του Mendix είτε ένα προσαρμοσμένο layout που έχει δημιουργηθεί από τον χρήστη.

1.3.3.2 Widgets

Τα Widgets είναι προκατασκευασμένα στοιχεία, έτοιμες λειτουργικές μονάδες που μπορούν να προστεθούν απευθείας σε κάθε σελίδα της εφαρμογής. Πρόκειται για εργαλεία που ενσωματώνονται εύκολα μέσω του Toolbox, όπως παρουσιάστηκε στο σχήμα 1.2. Ενδεικτικά παραδείγματα περιλαμβάνουν:

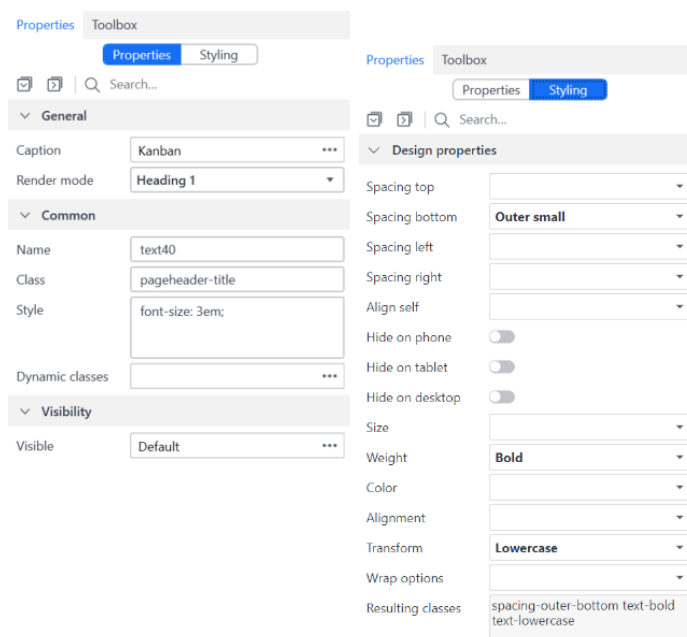
- Data containers – δομές δεδομένων που περιέχουν δεδομένα από τη βάση δεδομένων. Παραδείγματα είναι Data view, Data grid, List view κ.α.
- Text widgets – περιέχουν κείμενο. Παραδείγματα είναι Text, Label, Page Title κ.α.
- Structure widgets – δομικά στοιχεία που χρησιμοποιούνται για την οργάνωση των widgets στη σελίδα. Παραδείγματα είναι Layout grid, Container, Tab container, Snippet call, Table κ.α.
- Input widgets – πεδία εισόδου δεδομένων. Παραδείγματα είναι Text box, Text area, Check box, Radio button, Drop-down, Date picker, File uploader κ.α.
- Images, Videos ή Files – widgets που περιέχουν πολυμέσα.
- Buttons – widgets που εκτελούν ενέργειες. Το Mendix παρέχει αρκετά buttons με προκαθορισμένες ενέργειες για την εκτέλεση ενεργειών όπως Save, Cancel, Delete, New, Edit, Crate, Call microflow κ.α.



Σχήμα 1.8: Τα Widgets περιλαμβάνονται στο Toolbox panel.

Το Mendix προσφέρει προκαθορισμένα σύνολα από widgets, γνωστά ως *Building blocks*, τα οποία δημιουργούν στοιχεία όπως επικεφαλίδες (headers), φόρμες και ειδοποιήσεις (notifications). Αυτά τα Building blocks διευκολύνουν και επιταχύνουν τη διαδικασία ανάπτυξης, παρέχοντας στους χρήστες έτοιμες λύσεις που μπορούν να ενσωματωθούν απευθείας στις εφαρμογές.

Τέλος, για κάθε widget (όπως και για κάθε σελίδα) μπορούν να επεξεργαστούν οι ιδιότητές του. Το panel Properties (σχήμα) είναι ένα δυναμικό panel στο οποίο αλλάζει το περιεχόμενό του ανάλογα με το στοιχείο που έχουμε επιλεγμένο.



Σχήμα 1.9: To panel Properties.

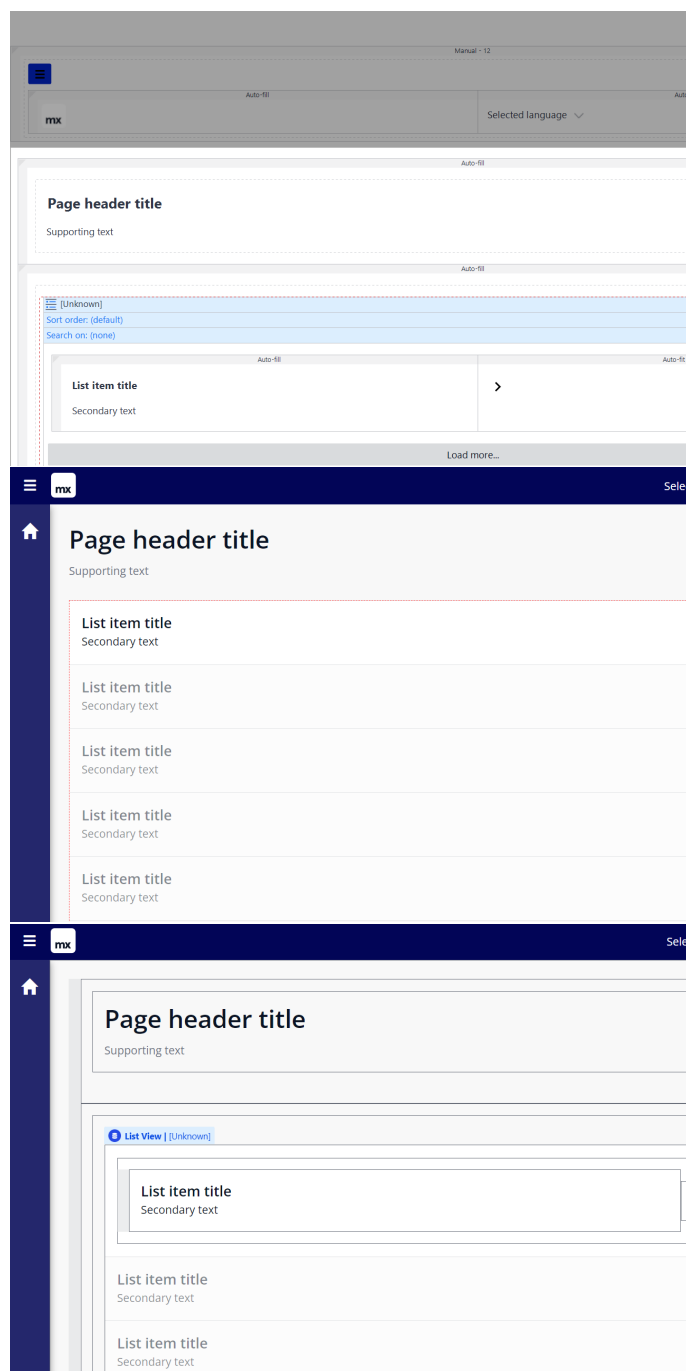
1.3.3.3 Εμφάνιση σελίδων

Στο Mendix Studio Pro, μια σελίδα μπορεί να προβληθεί είτε σε Structure Mode είτε σε Design Mode, προσφέροντας στους χρήστες διαφορετικές οπτικές για τη διαχείριση και τον σχεδιασμό των σελίδων.

Στο Structure Mode, παρουσιάζονται με σαφήνεια όλα τα δομικά στοιχεία που συνθέτουν τη σελίδα, δίνοντας έμφαση στη δομή της και επιτρέποντας την εύκολη προσαρμογή και οργάνωση των περιεχομένων. Αυτή η λειτουργία είναι ιδανική για την ανάλυση της λογικής της σελίδας, τη διαχείριση των στοιχείων της και τη διόρθωση πιθανών προβλημάτων διάταξης. Είναι επίσης ο μοναδικός τρόπος προβολής των εφαρμογών κινητών συσκευών (Native mobile).

Αντίθετα, στο Design Mode, η σελίδα απεικονίζεται όπως ακριβώς θα εμφανίζεται στον τελικό χρήστη. Αυτό προσφέρει μια πιο ρεαλιστική απεικόνιση της εμπειρίας χρήστη. Επιπλέον, το Design Mode περιλαμβάνει τη λειτουργία X-Ray Mode, η οποία συνδυάζει στοιχεία από το Structure Mode και το Design Mode. Με το X-Ray Mode, οι χρήστες μπορούν να δουν τόσο την αισθητική εμφάνιση όσο και τη δομή της σελίδας ταυτόχρονα. Αυτή η λειτουργία επιτρέπει την ακριβή τοποθέτηση και διαχείριση στοιχείων, ενώ ταυτόχρονα προσφέρει τη δυνατότητα άμεσων προσαρμογών σε επίπεδο σχεδιασμού και λογικής.

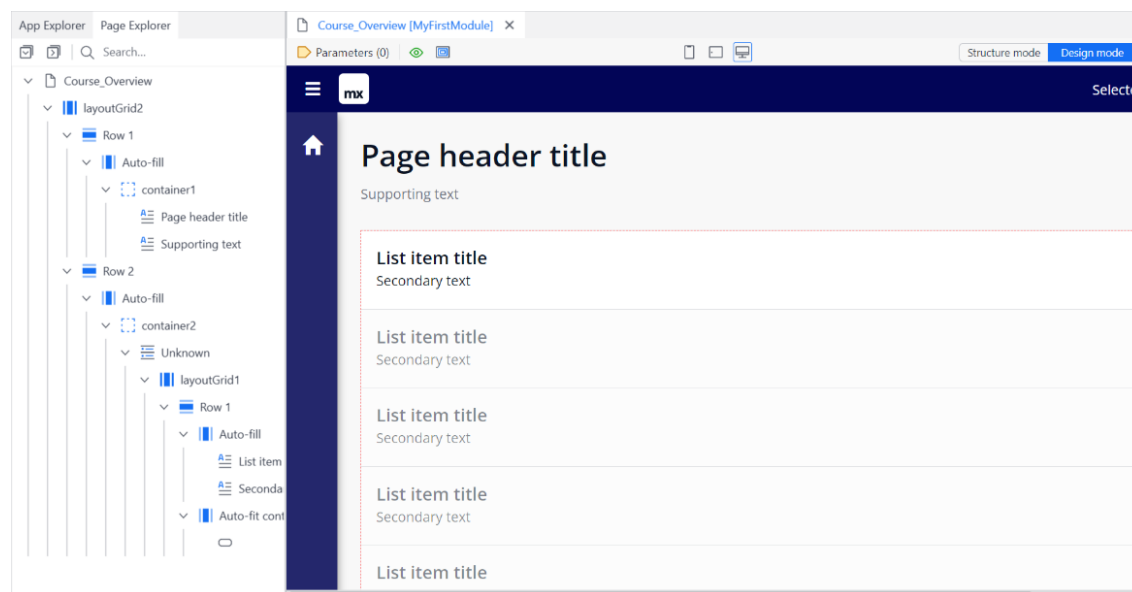
Οι διαφορετικές προβολές μπορούν να επιλεγούν από το πάνω μέρος του Working Area, το οποίο επιπλέον περιλαμβάνει και δυνατότητα αλλαγών στο μέγεθος του καμβά της σελίδας ώστε να ελεγχθεί το πως εμφανίζεται η εφαρμογή σε κινητά και τάμπλετ.



Σχήμα 1.10: Διαφορετικές εμφανίσεις της ίδιας σελίδας από αριστερά προς τα δεξιά: Structure Mode, Design Mode, X-Ray Mode.

1.3.3.4 Παράδειγμα δομής σελίδας

Στο σχήμα 1.11 εμφανίζεται ο Page Explorer και το Working area της σελίδας *Course_Overview* της εφαρμογής RapidDeveloper. Παρατηρούμε πως όλη η σελίδα είναι δομημένη γύρω από ένα Layout Grid, το οποίο αποτελείται από δύο Rows



Σχήμα 1.11: Page Explorer και Working area της σελίδας Course_Overview.

(γραμμές). Το Row 1 δημιουργεί το Header της σελίδας μαζί με το Supporting text τα οποία είναι τοποθετημένα σε ένα container⁶, και το Row 2 δημιουργεί ένα List View το οποίο επαναλαμβάνεται. Το List View αποτελείται και αυτό από ένα Layout Grid το οποίο δημιουργεί τα δύο texts τα οποία βλέπουμε στη λίστα όπως επίσης και ένα κουμπί (το οποίο εμφανίζεται εκτός οθόνης). Οι μπλε οριζόντια μπάρα και η μπλε κάθετη μπάρα στα αριστερά αποτελούν κομμάτι του Layout της σελίδας, το οποίο είναι το Atlas_Default.

1.3.4 Microflows

Τα Microflows (όπως και τα Nanoflows και τα Workflows)⁷ αποτελούν τον εγχεί-
φαλο των εφαρμογών του Mendix καθώς αναπαριστούν της λογικής λειτουργίας της

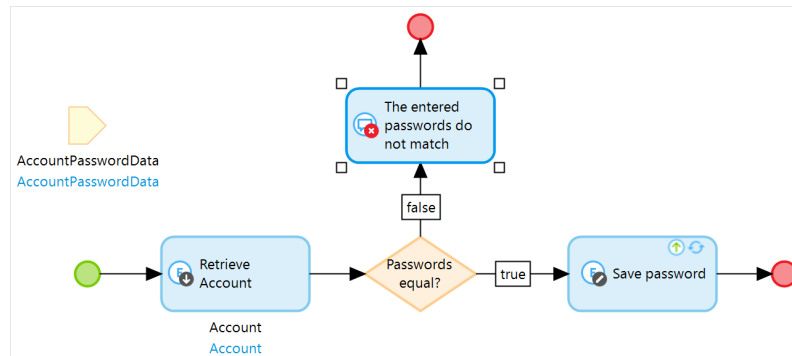
⁶Τα containers χρησιμοποιούνται καθώς είναι ένας βολικός τρόπος ομαδοποίησης κοινών στοιχείων της σελίδας. Έτσι μπορούν να δωθούν εύκολα κανόνες για αυτά τα στοιχεία όπως για παράδειγμα το visibility, συγκεκριμένα events ή και custom CSS κλάσεις.

⁷Η βασική διαφορά μεταξύ Microflows και Nanoflows έγκειται στη λειτουργικότητα και τον τρόπο εκτέλεσής τους. Τα Microflows βασίζονται σε βιβλιοθήκες της Java, εκτελούνται στον διακομιστή (runtime server) και, ως εκ τούτου, δεν είναι διαθέσιμα για εφαρμογές που λειτουργούν εκτός σύνδεσης. Από την άλλη πλευρά, τα Nanoflows χρησιμοποιούν βιβλιοθήκες της JavaScript, εκτελούνται στην πλευρά του client, γεγονός που τα καθιστά εν δυνάμει γρηγορότερα.

Τα Microflows είναι ιδανικά για την προσκόμιση και την επεξεργασία δεδομένων από τη βάση δεδομένων ή από εξωτερικές πηγές, εξασφαλίζοντας υψηλή αξιοπιστία και συνέπεια. Αντίθετα, τα Nanoflows χρησιμοποιούνται κυρίως για ενέργειες που σχετίζονται με την εμπειρία του χρήστη, όπως η εμφάνιση αναδυόμενων (pop-up) μηνυμάτων, η προβολή progress bars ή η ανταλλαγή cookies.

Τέλος, τα Workflows ενδείκνυνται για τη διαχείριση σταθερών και επαναλαμβανόμενων διαδικασιών, επιτρέποντας την αυτοματοποίηση και την απλοποίηση της εκτέλεσής τους.

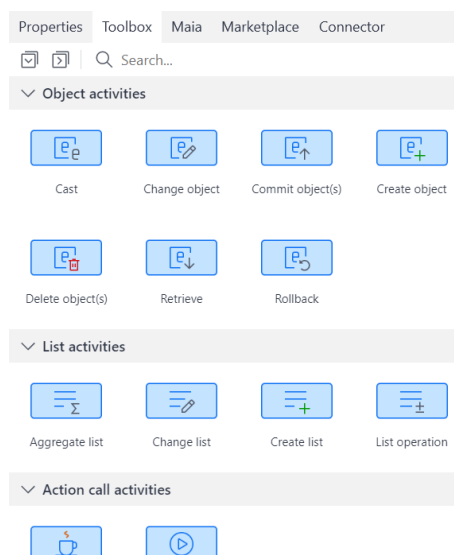
εφαρμογής με έναν οπτικό τρόπο χαμηλού κώδικα. Αυτά τα διαγράμματα ροής λειτουργούν ως οδικοί χάρτες για την εκτέλεση διάφορων εντολών και αλληλουχιών ενεργειών, επιτρέποντας την κατασκευή σύνθετης λειτουργικότητας χωρίς την ανάγκη γραφής παραδοσιακού κώδικα. Χρησιμοποιούνται ευρέως για ενέργειες όπως η δημιουργία, η ενημέρωση και η διαγραφή δεδομένων, η εμφάνιση σελίδων, το φιλτράρισμα δεδομένων, η εκτέλεση ελέγχων, η είσοδος δεδομένων από εξωτερικές πηγές και άλλα



Σχήμα 1.12: Παράδειγμα Microflow. Τα πράσινα και κόκκινα κυκλάκια αναπαριστούν τα events, τα μπλε ορθογώνια τα activities και ο πορτοκαλί ρόμβος το decision. Ως είσοδο έχουμε το parameter AccountPasswordData. [4].

Τα Microflows αποτελείται από τα παρακάτω στοιχεία:

- **Events** – λειτουργούν ως σημεία εκκίνησης και τερματισμού του Microflow. Χρησιμοποιώντας το τελικό event μπορούμε να ορίσουμε την τιμή και το τύπο δεδομένων που επιστρέφει το Microflow, με παρόμοιο τρόπο όπως στις συναρτήσεις ή μεθόδους του υψηλού κώδικα.
- **Decisions** – επιτρέπουν την εισαγωγή λογικών συνθηκών. Για παράδειγμα, ένα decision μπορεί να ελέγξει αν μια μεταβλητή έχει τιμή και, ανάλογα με την απάντηση, να κατευθύνει τη ροή σε διαφορετικές ενέργειες. Οι συνθήκες ορίζονται με τη χρήση εκφράσεων (βλ. ενότητα 1.3.5).
- **Activities** – αποτελούν τις κύριες ενέργειες που εκτελούνται στη ροή. Παραδείγματα τέτοιων ενεργειών είναι η δημιουργία (ή ενημέρωση ή διαγραφή) αντικειμένων μέσω του activity Create (ή Change ή Delete) object, η εμφάνιση μιας σελίδας στον χρήστη μέσω του Show page, η κλήση ενός άλλου Microflow μέσω του Microflow call, το φιλτράρισμα λιστών (List operation), η σύνδεση με εξωτερικές υπηρεσίες μέσω REST APIs κ.α.
- **Loops** – επιτρέπουν την εκτέλεση επαναλαμβανόμενων ενεργειών.
- **Parameter** – πρόκειται για τα δεδομένα εισόδου του Microflow, με αντίστοιχη λογική όπως οι συναρτήσεις υψηλού κώδικα.



Σχήμα 1.13: Όταν επεξεργαζόμαστε ένα Microflow, το Toolbox panel περιλαμβάνει τα διαθέσιμα activities.

Ένα Microflow μπορεί να εκτελεστεί από διαφορετικά μέρη όπως το Navigation menu, ένα κουμπί, ένα link ή ακόμα και να καλεστεί από ένα άλλο Microflow. Επιπλέον, τα Microflows μπορούν να εκτελεστούν αυτόματα μετά από μια συγκεκριμένη ενέργεια (Event Handlers), όπως η αποθήκευση ενός αντικειμένου ή η ενημέρωση ενός πεδίου.

Τέλος, κάθε Microflow αποθηκεύεται ως ένα Java αρχείο στον πηγαίο κώδικα της εφαρμογής. Για εξειδικευμένες λειτουργίες, το Mendix παρέχει τη δυνατότητα ενσωμάτωσης προσαρμοσμένου κώδικα Java μέσω του App → Deploy to Eclipse.

1.3.5 Εκφράσεις

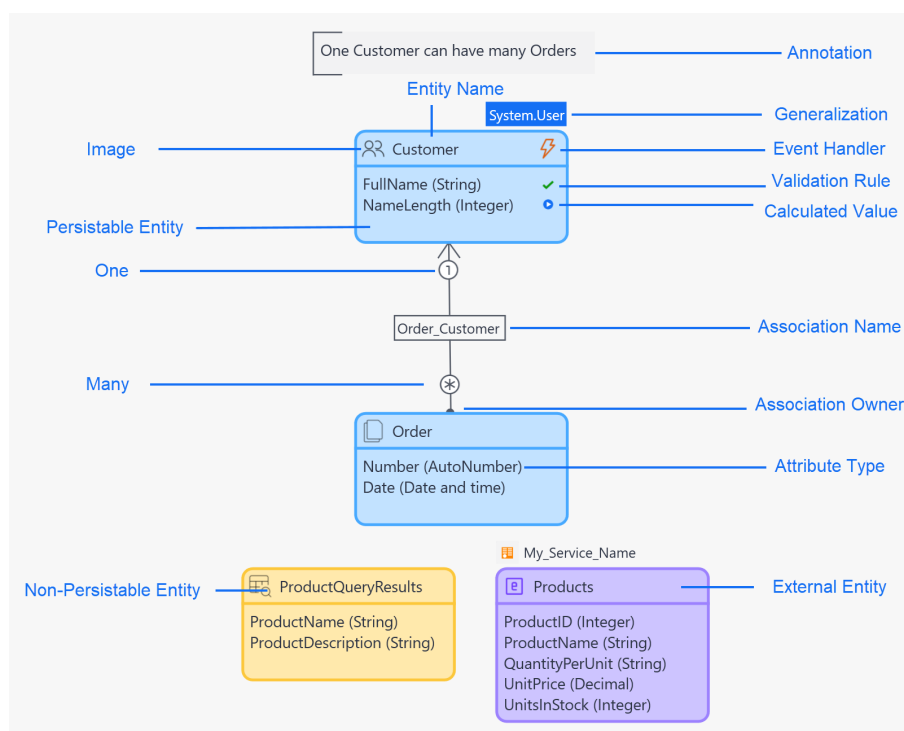
Οι εκφράσεις (expressions) του Mendix είναι ένας τρόπος ενσωμάτωσης λειτουργικότητας στην εφαρμογή μας. Οι εκφράσεις μπορούν να περιλαμβάνουν σταθερές τιμές, μεταβλητές, συναρτήσεις, λογικές πράξεις, συγκρίσεις, επιλογές κ.α. Για παράδειγμα, μπορεί να οριστεί η εμφάνιση ενός συγκεκριμένου widget μόνο αν ισχύει μια συγκεκριμένη συνθήκη. Οι εκφράσεις μπορούν να χρησιμοποιηθούν σε πολλά σημεία της εφαρμογής, όπως στα Microflows, στις ιδιότητες των widgets κ.α.

Για παραδειγμα, η έκφραση `if $package/weight < 1.00 then 0.00 else 5.00` ελέγχει το γνώρισμα `weight` της οντότητας `package` και επιστρέφει `0.00` αν το βάρος είναι μικρότερο από `1.00`, αλλιώς επιστρέφει `5.00`. Θα δούμε περισσότερες τέτοιες εκφράσεις στην πράξη στο κεφάλαιο ??.

1.3.6 Domain Model

Το domain model αναπαριστά τη δομή των δεδομένων κάποιου module στην πλατφόρμα Mendix. Τα δεδομένα που περιγράφονται από το domain model αποθηκεύονται στη συνέχεια σε ένα σχεσιακό σύστημα βάσεων δεδομένων του Mendix.

Το domain model αποτελεί κεντρικό πυλώνα της αρχιτεκτονικής κάθε εφαρμογής. Κάθε module έχει το δικό του domain model, και όλα τα modules μπορούν να χρησιμοποιούν δεδομένα από όλα υπόλοιπα domain modules μέσω συσχετίσεων.



Σχήμα 1.14: Παράδειγμα από domain model. [4]

Το σχήμα 1.14 είναι ένα παράδειγμα ενός domain model που αναπαριστά πελάτες και παραγγελίες. Οι πελάτες και οι παραγγελίες αποτελούν οντότητες (entities) του domain model. Οι οντότητες συσχετίζονται μεταξύ τους με μια συσχέτιση (association) πολλών-προς-ένα. Κάθε παραγγελία ανήκει σε έναν μόνο πελάτη, ενώ ένας πελάτης μπορεί να έχει πολλές παραγγελίες. Φυσικά, το Mendix περιλαμβάνει και άλλες πληθικότητες, όπως συσχετίσεις ένα-προς-ένα όπως επίσης και πολλά-προς-πολλά. Επιπλέον, αν διαγραφτεί κάποια οντότητα μπορεί να ρυθμιστεί τι θα συμβεί με τις συσχετίσεις της (π.χ. να διαγραφούν και αυτές).

Μέσα στα ορθογώνια που αναπαριστούν τις οντότητες βρίσκονται τα γνωρίσματα, οι ιδιότητες (attributes) των οντοτήτων. Στην παρένθεση κάθε γνωρίσματος καταγράφεται ο τύπος δεδομένων του. Παρατηρούμε πως υπάρχουν ορθογώνια με διαφορετικά χρώματα, κάτι που αντιστοιχεί σε διαφορετικού είδους οντότητες. Τα μπλε ορθογώνια αναπαριστούν οντότητες που αποθηκεύονται στη βάση δεδομένων, με κίτρινο

μη-διατηρήσιμες οντότητες (non-persistent entities), δηλαδή οντότητες που δεν αποθηκεύονται στη βάση δεδομένων αλλά αποθηκεύονται προσωρινά στη μνήμη, και τέλος με μωβ οντότητες από εξωτερικές πηγές δεδομένων. Η μπλε ετικέτα `System.User` που συνοδεύει την οντότητα `Customer` δηλώνει πως η οντότητα αυτή βασίζεται σε μια άλλη οντότητα, την οντότητα `User` του module `System` (Γενίκευση – Generalization)⁸. Τέλος, ανάλογα αν στην οντότητα έχει οριστεί κάποιος Event handler ή αν σε κάποιο γνώρισμα υπάρχει κάποιο validation rule, το Mendix το αναγνωρίζει και το αναπαριστά με το αντίστοιχο σύμβολο.

1.3.6.1 Οντότητες

Name	Type	Default value	Microflow	Limitations
Title	String (200)			
Notes	String (unlimited)			
DueDate	Date and time			
StartDate	Date and time	[%CurrentDateTi...		
CalendarColor	String (200)	rgb(38,74,229)		
TaskStatus	Enumeration 'TaskStatus'	To_Do		
CreationDateBubble	Boolean	true		
TaskPriority	Enumeration 'TaskPriority'	Low		

Σχήμα 1.15: Ιδιότητες μιας οντότητας Task (βλ. κεφ. ??)

⁸Η έννοια της γενίκευσης στο Mendix βασίζεται σε μια λογική που θυμίζει την κληρονομικότητα (inheritance) στις αντικειμενοστραφείς γλώσσες προγραμματισμού, δηλαδή επιτρέπει σε μία οντότητα (entity) να κληρονομεί τις ιδιότητες και τις συσχετίσεις μιας άλλης υπερωντότητας, χρησιμοποιώντας τα χαρακτηριστικά και τις συσχετίσεις της, ενώ παράλληλα μπορεί να ορίσει πρόσθετες ιδιότητες ή συσχετίσεις που είναι μοναδικές για την ίδια. Η γενίκευση είναι ιδιαίτερα χρήσιμη καθώς συμβάλλει στη διατήρηση της ακεραιότητας και της ασφάλειας των δεδομένων. Για παράδειγμα, οι “Customers” που κληρονομούν τα γνώρισμα της οντότητας `System.User`, ταυτόχρονα κληρονομούν και όλα τα χαρακτηριστικά ασφάλειας του `User` που το Mendix έχει προκατασκευάσει.

Στο σχήμα 1.15 παρουσιάζεται το παράθυρο που εμφανίζεται όταν δημιουργούμε ή επεξεργαζόμαστε μια οντότητα. Στο πάνω μέρος ορίζεται αν η οντότητα έχει κάποια γενίκευση και το αν είναι διατηρήσιμη στη βάση δεδομένων.

Στην καρτέλα Attributes (Γνωρίσματα) καθορίζονται όλα τα γνωρίσματα της οντότητας. Εκεί επιλέγεται ο τύπος δεδομένων του γνωρίσματος. Οι τύποι δεδομένων που υποστηρίζονται από το Mendix είναι οι εξής: AutoNumber (αυτόματα παραγόμενοι αριθμοί, π.χ. IDs), Binary, Boolean, Date and time, Decimal, Enumeration (επιλέγεται κάποιο Enumeration έγγραφο), Hashed string, Integer, Long, String. Ο τύπος ενός γνωρίσματος είναι πιθανό να καθοριστεί αυτόματα από το Mendix βάσει του όνοματος που επιλέγεται κατά τον ορισμό του. Τέλος, μπορεί να οριστεί μια προεπιλεγμένη τιμή του κάθε ορίσματος ή να οριστεί η τιμή να καθορίζεται από κάποιο microflow.

Σχήμα 1.16: Προσθήκη καινούριου γνωρίσματος σε μια οντότητα

Η καρτέλα Associations είναι ένας εναλλακτικός τρόπος που επιτρέπει να επεξεργαστούν οι συσχετίσεις μιας οντότητας. Ένας εναλλακτικός τρόπος είναι απευθείας από το domain model μέσω των συνδέσεων μεταξύ των οντοτήτων.

Η καρτέλα Validation rules (Κανόνες Επικύρωσης) δημιουργεί συνθήκες που πρέπει να ικανοποιούνται για να είναι έγκυρα τα γνωρίσματα κάθε οντότητας. Οι κανόνες επικύρωσης μπορούν να είναι απλές συνθήκες, όπως π.χ. ένα πεδίο να μην είναι κενό, ή πιο σύνθετες, όπως π.χ. η τιμή ενός πεδίου να καθορίζεται από ένα εύρος τιμών. Αν η συνθήκη δεν πληρούται, εμφανίζεται αυτόματα μήνυμα σφάλματος κάτω από το πεδίο.

Η καρτέλα Event handlers (Χειριστές Συμβάντων) είναι ένας τρόπος για να εκτελεστούν συγκεκριμένες ενέργειες (microflows) όταν συμβεί κάποιο συγκεκριμένο συμβάν στην οντότητα. Τα συμβάντα μπορεί να είναι η δημιουργία (create), η ενη-

Validation Rule

General

Attribute: DueDate

Error message: Δεν μπορεί η ημερομηνία λήξης να είναι πριν την ημερομηνία εκκίνησης!

Rule

Type: Range

☒ Greater than or equal to (>=)

☐ Value ☒ Attribute

StartDate

☐ Smaller than or equal to (<=)

☒ Value ☐ Attribute

OK Cancel

Σχήμα 1.17: Προσθήκη κανόνα επικύρωσης (βλ. κεφ. ??)

Edit Access Rule of Entity 'TaskManager.Task'

Documentation

Rule applies to the following module roles

☐ Guest ☒ User

☒ Select / deselect all

Access rights

Create and delete rights

☒ Allow creating new objects ☒ Allow deleting existing objects

Member read and write rights

Default rights for new members ☐ None ☐ Read ☒ Read, Write

Member	Access rights
Title (String (200))	Read, Write
Notes (String (unlimited))	Read, Write
DueDate (Date and time)	Read, Write
StartDate (Date and time)	Read, Write
CalendarColor (String (200))	Read, Write
TaskStatus (Enumeration 'TaskStatus')	Read, Write

Set all to:

XPath constraint (used to constrain read, write and delete rights)

XPath constraint:

OK Cancel

Σχήμα 1.18: Επεξεργασία κανόνων πρόσβασης της οντότητας Task (βλ. κεφ. ??)

μέρωση (commit ή save), η διαγραφή (delete) ή η ακύρωση (rollback ή cancel) μιας οντότητας. Μπορεί να επιλεγεί η εκτέλεση του microflow πριν ή μετά την εκτέλεση

των παρακάτω συμβάντων.

Η καρτέλα Access Rules (Κανόνες Πρόσβασης) ορίζει τα δικαιώματα χρήστης από κάθε ρόλο χρήστη όσον αφορά την αλληλεπίδραση με τα γνωρίσματα της οντότητας. Μπορεί να επιλεγεί ποιος ρόλος χρήστη μπορεί να δημιουργήσει ή να διαγράψει στιγμιότυπα από οντότητες και να διαβάσει ή να τροποποιήσει τα γνωρίσματά τους.

Τέλος, μπορούν να προστεθούν breakpoints στα Microflows ώστε να γίνουν debug πιθανά σφάλματα κατά την εκτέλεση τους.

Βιβλιογραφία

- [1] *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].
- [2] 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.
- [3] *Mendix Cloud* — *docs.mendix.com*. <https://docs.mendix.com/developerportal/deploy/mendix-cloud-deploy/>. [Accessed 04-01-2025].
- [4] *Mendix Documentation* — *docs.mendix.com*. <https://docs.mendix.com/>. [Accessed 04-01-2025].
- [5] *Mendix Forum - System Module* — *community.mendix.com*. <https://community.mendix.com/link/space/studio-pro/questions/88842>. [Accessed 05-01-2025].