



Low Code

Mendix

Low Code Development

- Erst 2016 wurde sie unter anderem in einer Publikation des US-amerikanischen Marktforschungsunternehmens Forrester verwendet.
- Konzept: in den 90er Jahren gab es einige Ansätze, mit denen versucht wurde, das Programmieren von Apps durch den Verzicht auf umfassendes Code-Schreiben möglich zu machen. Diese Plattformen liefen unter dem Begriff „Rapid Application Development“, kurz RAD.
- Heutige Low-Code-Plattformen liefern meist einen visuellen Ansatz zur App-Entwicklung.
- Die Entwicklungsumgebung stellt die Elemente der späteren Anwendungen wie eine Kiste mit Bauklötzen zur Verfügung. Mittels Drag-and-Drop können Entwickler verschiedener Erfahrungsstufen die Benutzeroberfläche so aus vorgegebenen UI-Elementen (User-Interface-Elementen) zusammenbauen und diese später mit Funktionalität anreichern – ohne oder mit nur wenig selbst geschriebenem Code.
- Low-Code-Plattformen sind oft als „Application Platform as a Service“ (APaaS) konzipiert, also als Cloud- oder Web-Dienst, der sämtliche Schritte der Entwicklung, von der Präsentation bis zur Datenverarbeitung und Authentifizierung, abwickeln kann.
 - Nachteil: Nutzer muss sich an diesen Dienst binden – Exporte aus anderen Systemen sind meist nicht möglich.
- Durch den Wegfall von repetitiven Programmieraufgaben können Entwickler Fokus auf die Prozesse und Funktionalitäten statt auf das Coding legen



Was zeichnet eine gute Low-Code-Plattform aus?

Die folgenden 6 Kriterien sollten bei der Auswahl einer Plattform berücksichtigt werden.

1. **Sicherheit**

- Die Sicherheit der Daten muss gewährleistet sein.

2. **Skalierbarkeit**

- Fähigkeit der Anwendung, zu wachsen. Ist die Anwendungen den auf sie zukommenden komplexen Anwendungsszenarien gewachsen.

3. **Visual Development per Drag & Drop**

- Die Plattform sollte anwenderfreundlich sein, damit sie auch Laien problemlos verwenden können. Hier ist das oben beschriebene Drag-and-Drop-Prinzip von zentraler Bedeutung. Es ermöglicht es dem Entwickler, sich auf die Gestaltung und Funktion der App zu konzentrieren.

4. **Interoperabilität mit Drittanbietern**

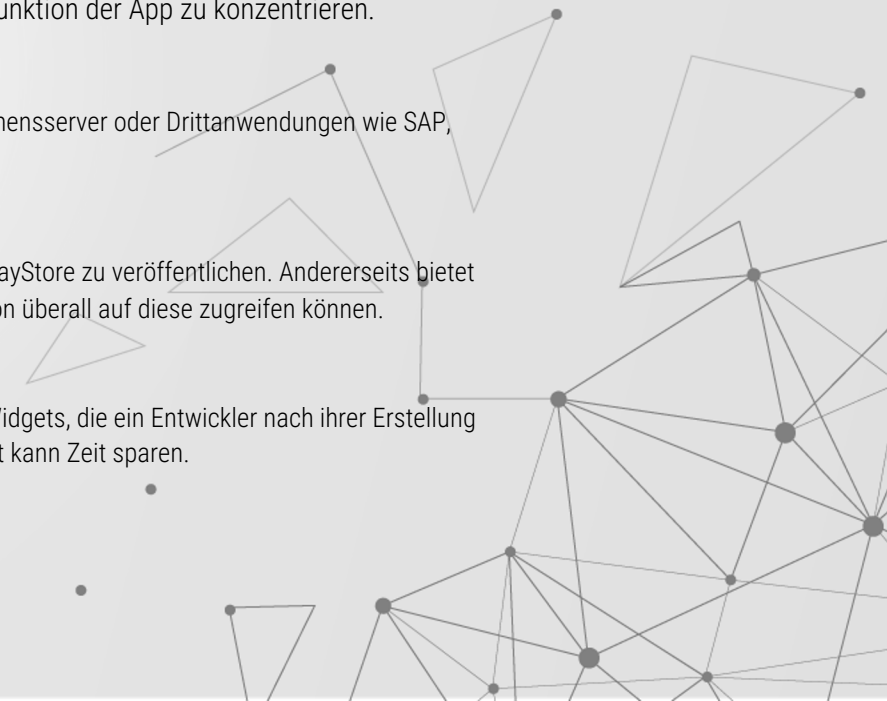
- Die Low-Code-Plattform sollte keine Insellösung sein. Sollte eine Vielzahl an Datenquellen wie eigene Unternehmensserver oder Drittanwendungen wie SAP, Oracle oder Outlook unterstützen, damit der Entwicklungsprozess möglichst reibungslos ablaufen kann.

5. **Unterstützung von mobilen Anwendungen**

- Eine Plattform muss in der Lage sein, Anwendungen auf gängigen App-Plattformen wie dem Apple- oder dem PlayStore zu veröffentlichen. Andererseits bietet auch die mobile Verwendung der Low-Code-Plattform selbst den Vorteil, dass Entwickler problemlos jederzeit von überall auf diese zugreifen können.

6. **Wiederverwendbarkeit**

- Wiederverwendbarkeit bestimmter Elemente im ganzen Unternehmen. Darunter fallen Vorlagen, Plug-Ins oder Widgets, die ein Entwickler nach ihrer Erstellung so speichern können sollte, dass seine Nachfolger ebenfalls darauf zugreifen können. Die Wiederverwendbarkeit kann Zeit sparen.





Was ist Mendix?

Mendix ist nach Forrester und Gartner einer der Marktführer im Bereich Low-Code-Development und bietet komplette Makro- und Mikroentwicklung an.

Nutzt die Anbindung an offene Standards wie BPMN (Business Process Model and Notation), UML (Unified Modeling Language) oder OData (Open Data), und ist somit keine abgekapselte Insellösung.

Mendix stellt Anwendern eine Entwicklungsumgebung zur Verfügung, in der sie Apps entwickeln, bereitstellen, testen und verwalten können, die dann sowohl auf iOS- als auch auf Android-Geräten laufen. Komponenten des Domain-Models, der Logik und der Benutzeroberfläche können in Mendix-Anwendungen abgelegt und später von anderen Entwicklern wiederverwendet werden.

Unterstützt das Konzept des Multi-User-Development. Das heißt es können zeitgleich mehrere Entwickler transparent in einer App arbeiten und sich abstimmen.

Möglichkeit der Versionierung (Git, Mercurial)



Mendix Studio vs Mendix Studio Pro

Mendix Studio

- No-Code Plattform für sog. Citizen-Developer (Personen ohne bes. Kenntnisse in der Entwicklung)
- Entwicklungsumgebung ist zielgruppenspezifisch, intuitiv und visuell ausgerichtet
- Verständnis von Geschäftsfaktoren, -zielen und –prozessen
- Apps sind responsive
- Mendix Assist, ein mit künstlicher Intelligenz ausgestatteter Agent, unterstützt Anwender bei der Konfiguration von Microflows

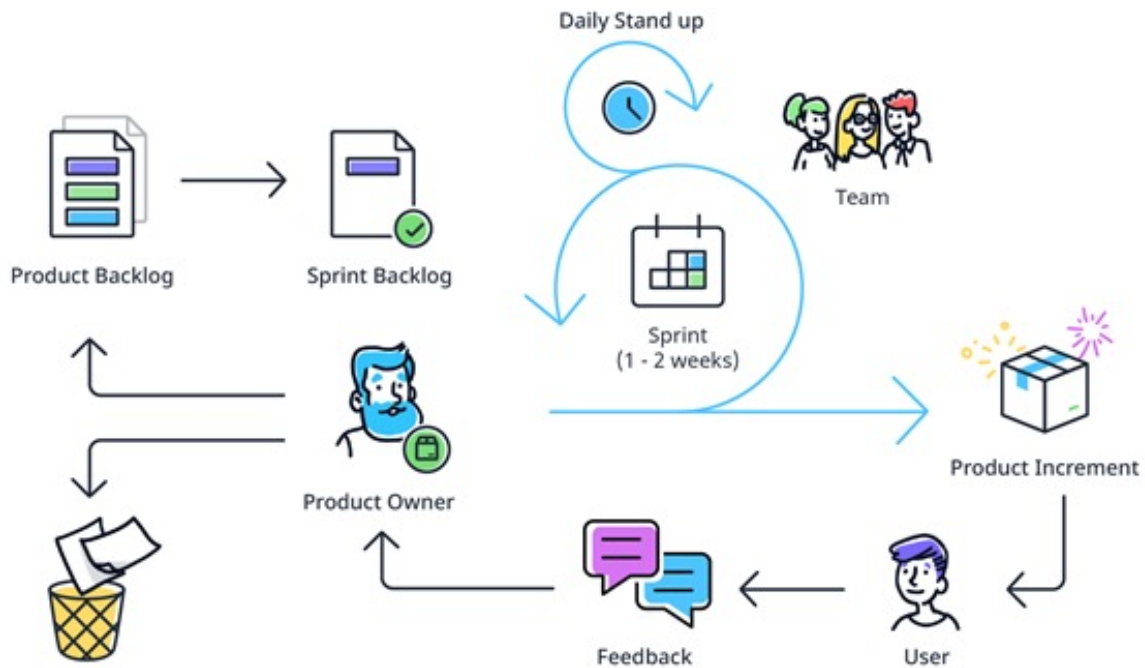
Mendix Studio Pro

- richtet sich als leistungsstarke, visuelle und modellbasierte Low-Code-Umgebung an professionelle Entwickler
- möglich, Apps mit individuellem Code oder wiederverwendbaren Komponenten zu erweitern und an individuelle Anforderungen anzupassen
- Mendix Assist fungiert in Studio Pro als Co-Programmierer, der die nächsten Schritte voraussagt, Logikabläufe überprüft und Fehler aufdeckt



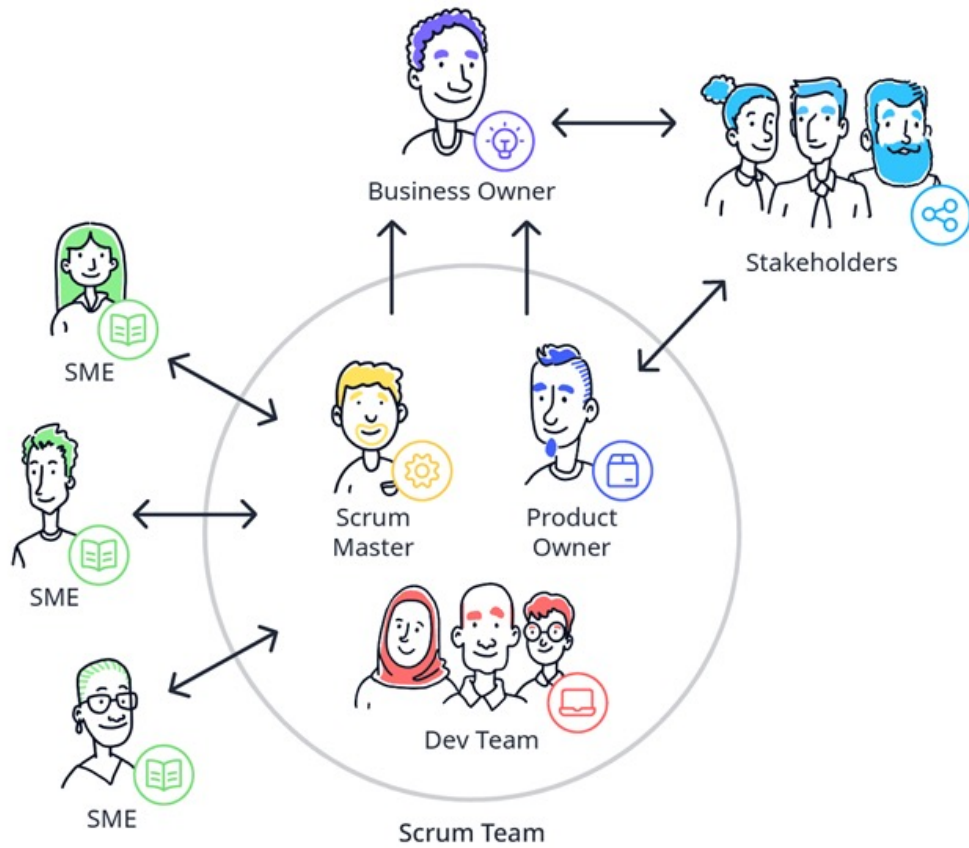
Agile Projektmethodik

Agile Process



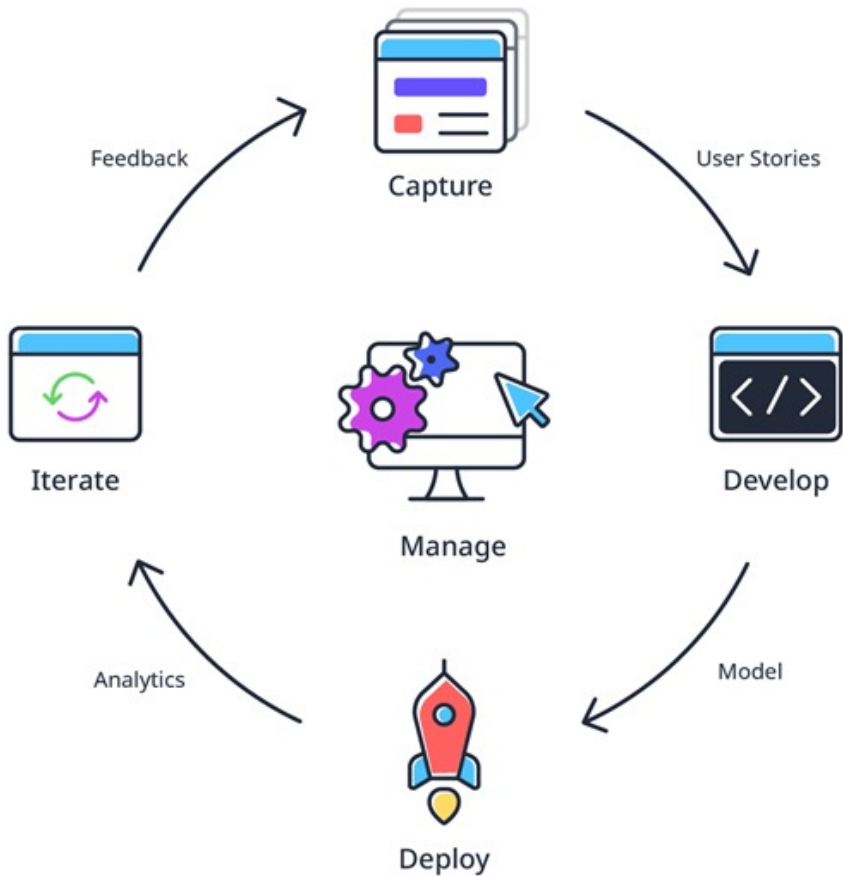
1. Kleiner Teil der App wird definiert, entwickelt und getestet, was ein Feedback und eine Anpassung der Anforderungen viel früher im Projekt ermöglicht
2. Mendix ermöglicht eine agile Entwicklung in allen Teilen des Produktlebenszyklus
3. Schnelle Entwicklung transformativer Anwendungen erfordert schnelles Experimentieren, häufige Iterationen und eine enge Zusammenarbeit
4. Ermöglicht schnelles Prototyping, bietet höheren Mehrwert bei Time-to-Market und in der Gesamtqualität

Scrum



1. Der agile Mendix-Ansatz basiert auf Scrum
2. Kleine Teams von bis zu 7 Personen erreichen:
 - a. eine größere Konzentration
 - b. höhere Produktivität
 - c. effektivere Kommunikation
 - d. einfachere Dokumentation
3. Product Owner: verfolgt Erwartungen der Projektbeteiligten (Stakeholder) und definiert Anforderungen für das Projekt. Kommuniziert die Vision an das Team, um Prios zu setzen
4. Scrum Master: konzentriert sich auf das Optimieren der Prozesse innerhalb des Teams. Beseitigt Hindernisse die den Entwicklungsprozess behindern könnten

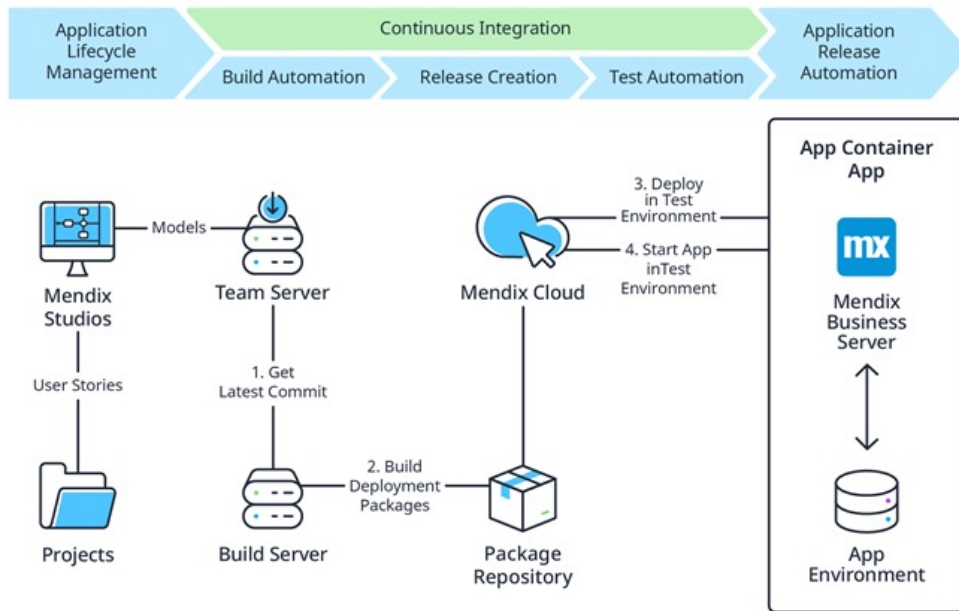
User Stories



1. Arbeit wird mit User Stories definiert
2. Erste Schritt des Produktlebenszyklus
3. Ist eine Möglichkeit, Bedürfnisse des Benutzers in Elemente u übersetzen an denen man arbeiten kann
4. Product Owner definiert Priorität der User Stories im Product Backlog
5. Team wählt aus, welche im Sprint abgearbeitet werden (ca. 2 Wochen)
6. Tägliche Standup-Meetings (Dailys) geben Einblick in die Arbeit der einzelnen Teammitglieder



Application Lifecyclemanagement



1. User Stories können in das Projekt importiert werden
2. Wird mit dem Team Server verbunden, wo Mendix die Versionsverwaltung übernimmt
3. Team Server ist in den Build-Server integriert, der Packages der Anwendung erstellt, die in das Package Repository gestellt werden können
4. Von dort kann das Projekt in einer Cloud Umgebung bereit gestellt werden für Testphase

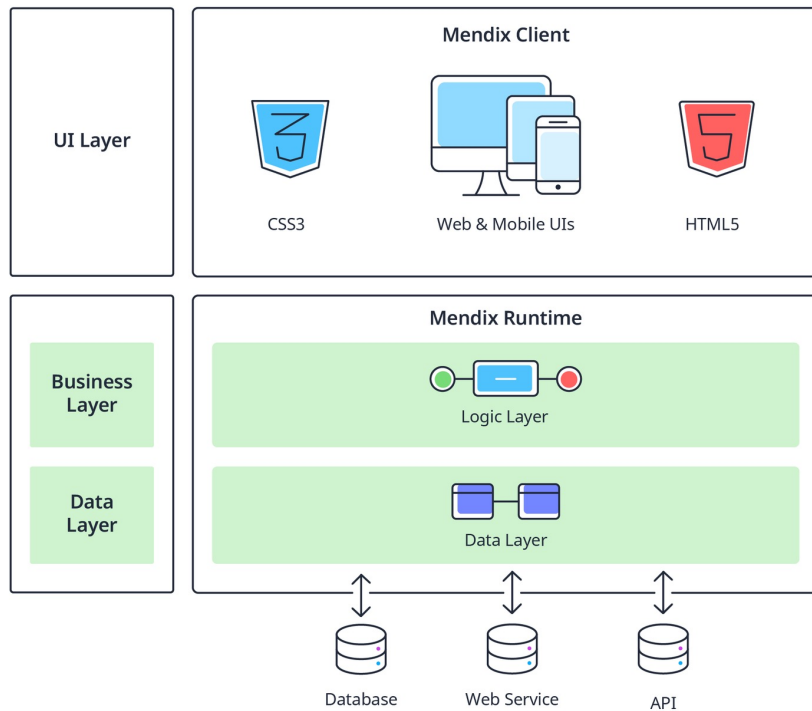
Integration with Existing CI/CD Pipelines



1. Mit einem Build-Server wie Jenkins kann an mehreren Stellen in die Mendix-Build-Pipeline eingegriffen werden.
2. Es können die neueste Version vom Team-Server abgerufen werden, automatisierte Tests mit ATS ausgeführt werden, Integrationstests und Benutzerakzeptanztests durchgeführt werden und einiges mehr



Mendix Architektur



1. Der Mendix-Client ist im Wesentlichen nur eine Implementierung des Model View Controller-Pattern in einer Single-Page-Anwendung.
2. Diese Komponente kommuniziert über HTTP mit der Mendix Runtime, einer Implementierung eines objektrelationalen Mappings.
3. Zusammen bieten diese beiden Komponenten alles, was zum Erstellen eines Microservices benötigt wird.

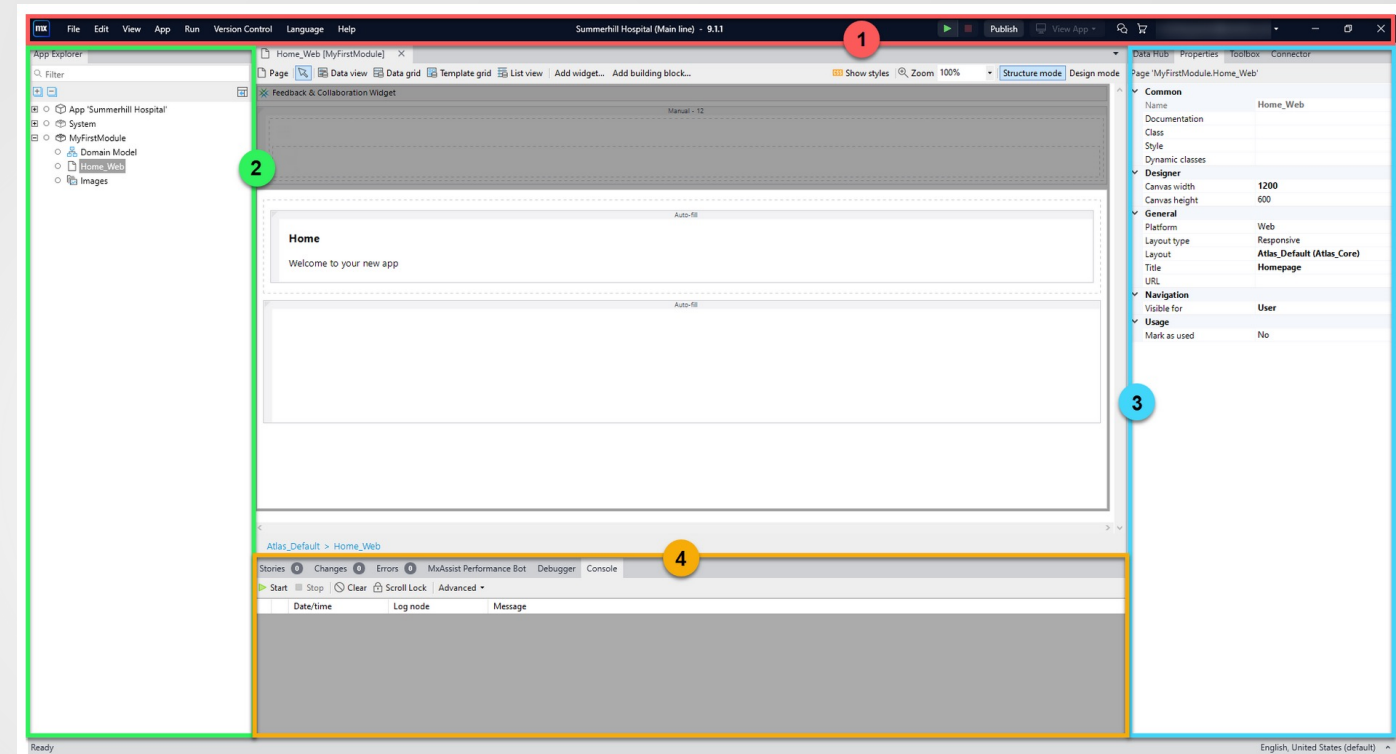


The background is a light gray gradient. On the left side, there is a complex network of thin gray lines connecting various black dots of different sizes, creating a web-like structure. Scattered across the entire background are numerous thin, light gray triangles of various sizes and orientations. Some of these triangles are solid, while others are just outlines. In the upper right corner, there are small, faint clusters of dots, some of which are open circles.

Showcast

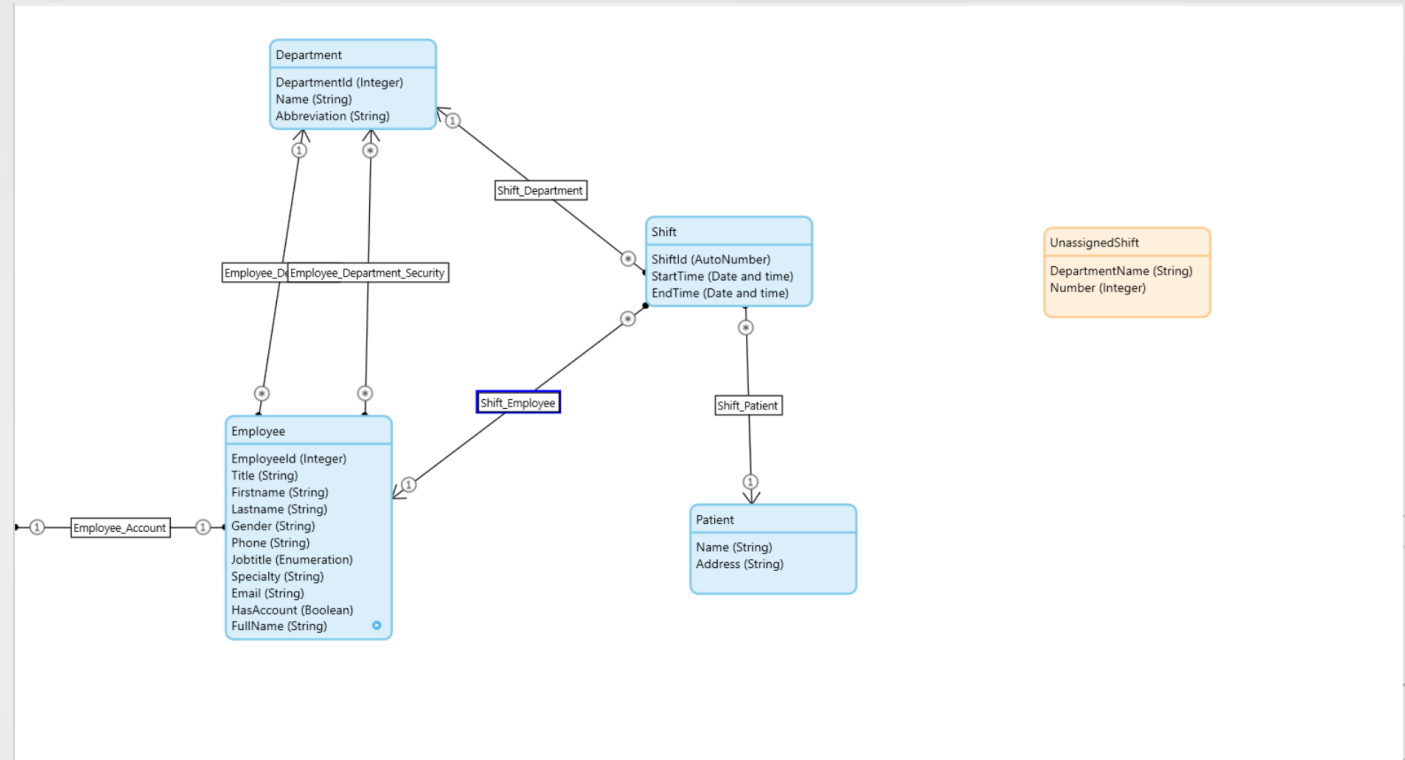
Mendix Studio Pro UI

1. Control bar
2. App Explorer
 - a. Properties des Projekts (Navigation, Security, ...)
3. Properties, Toolbox Tabs
4. User Story Management, Debugging, ...



Domain Model View

1. Um Daten zu speichern wird das Domain Model benötigt, ein Object Relational Mapping (ORM)
2. Unterstützt verschiedene Datenbanken
 - a. Interne Datenbank ist gut für kleine Apps
 - b. IBM DB2
 - c. Microsoft SQL Server
 - d. MySQL/MariaDB
 - e. Oracle Database
 - f. PostgreSQL
3. Einmal Konfiguriert übernimmt Mendix die Kommunikation zwischen Datenbank und App (creating & migrating des Datenbank Schemas inkl. Versionierung)



Domain Model

1. Mendix speichert Daten in Entitäten ab
2. Entitäten besitzen Attribute (sind Merkmale, die eine Entität beschreiben und/oder identifizieren)
 - a. Bsp: Kundenentität hat Attribut für Namen, Mail, usw.
Rechnungseinheit hat eine ID, Lieferadresse usw.
3. Entitäten können Beziehungen zu anderen Entitäten haben => Associations
 - a. Bsp: Bestellung die einem Kunden gehört
4. Vererbung mithilfe von Klassentabellenvererbung
 - a. Jede Entität erhält eigene Tabelle in der Datenbank
 - b. Untergeordnete Tabellen sind mit übergeordneter Tabelle verknüpft
 - c. Beim Abrufen von Daten aus einer untergeordneten Klasse durchläuft das System die Hierarchie zu übergeordneter Tabelle um Attribute der übergeordneten Tabelle abzurufen

Properties of Entity 'UserManager.Employee'

General

Name:

Generalization:

Image:

Persistable: ☒ Yes ☐ No
Objects of this entity can only be stored in the database if it is persistable.

System members

☐ Store 'createdDate'

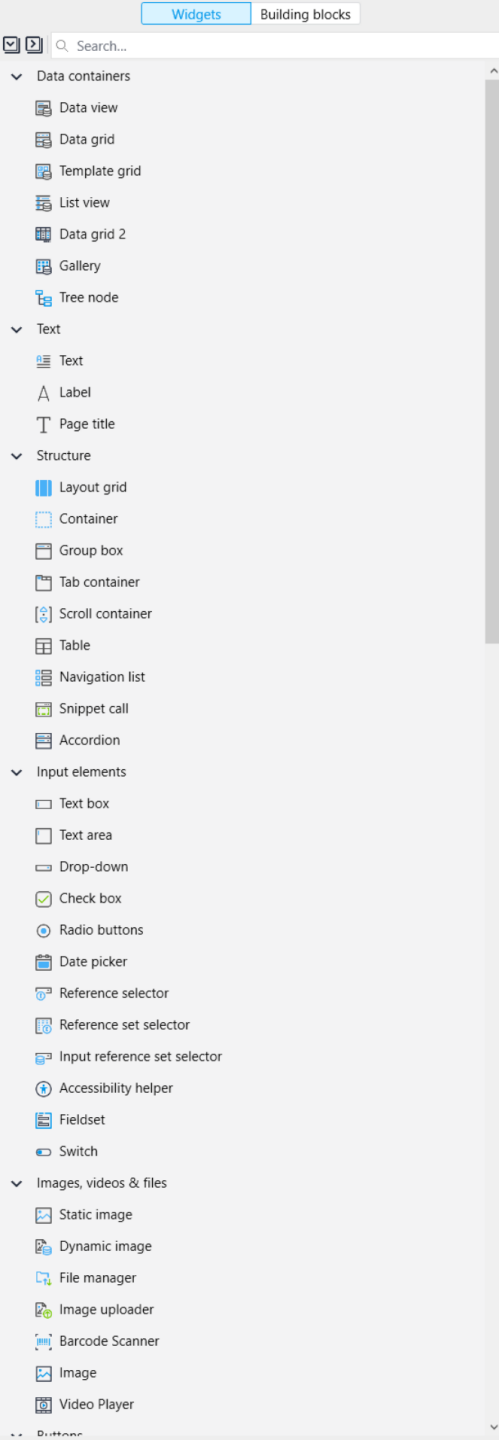
☐ Store 'changedDate'

☐ Store 'owner'

☐ Store 'changedBy'

Attributes Associations Validation rules Event handlers Indexes Access rules Documentation

Name	Type	Default value	Microflow	Limitations
Employeeid	Integer	0		
Title	String (200)			
Firstname	String (200)			
Lastname	String (200)			
Gender	String (200)			
Phone	String (200)			
Jobtitle	Enumeration 'Enum_Jobtitle'	Nurse		
Specialty	String (200)			
Email	String (200)			
HasAccount	Boolean	false		
FullName	String (200)		UserManager.CAL_Employee_FullName	Non-filterable, non-sortable



Template Engine

1. Mendix bietet einen webbasierten Client, der auf modernen Webstandards wie JavaScript, CSS und HTML5 basiert. Die Client-Implementierung verwendet Technologien wie React und SCSS
2. pagebuilder in Studio Pro vorhanden
3. 12 Column Grid System
4. CSS basierend auf Bootstrap
5. Page stored as
6. Eine Seite wird als XML-Umschlag gespeichert, der HTML und einen Link zu einem möglichen Layoutdokument enthält.
7. Die Layoutkomponenten selbst sind ebenfalls XML-Dateien und enthalten HTML-Elemente. Der Client erstellt eine Seite, indem er die entsprechende XML-Datei herunterlädt. Diese Datei wird von der Vorlagen-Engine analysiert. Alle zusätzlichen Layoutdateien werden heruntergeladen und in die Seite eingefügt. Dieser Vorgang wiederholt sich, bis alle Layouts heruntergeladen sind und die finale Seite im Browser sichtbar ist.



Navigation

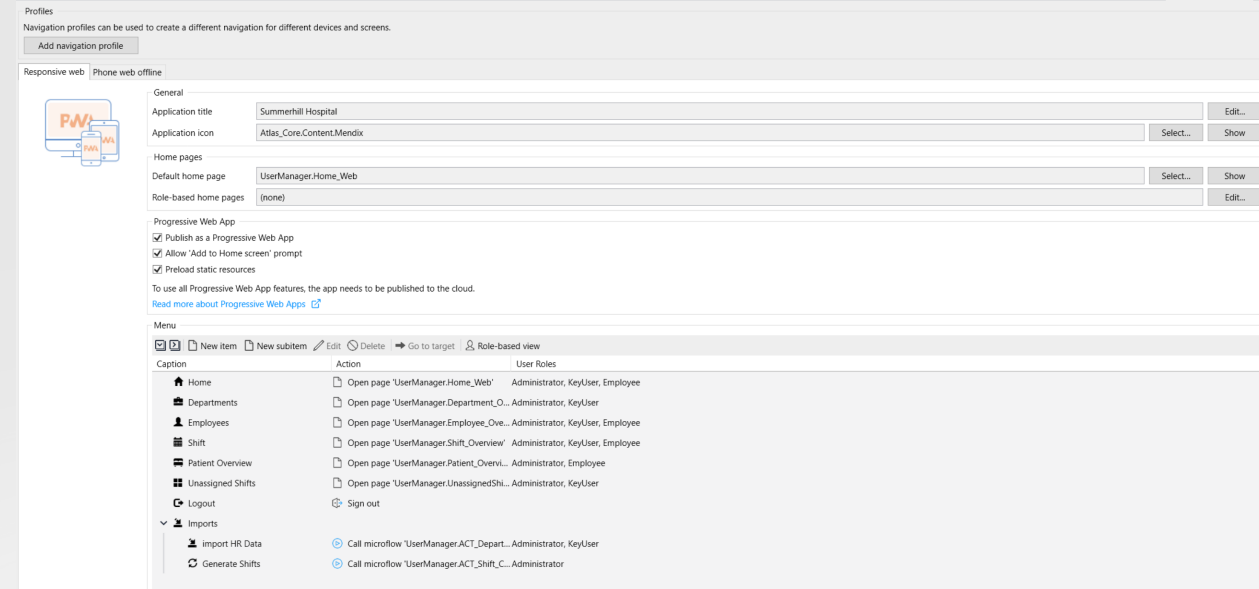
Es gibt sieben Navigationsprofile, die Mendix anbietet:

- Responsive web
- Responsive web offline
- Tablet web
- Tablet web offline
- Phone web
- Phone web offline
- Native mobile (tablet & phone)

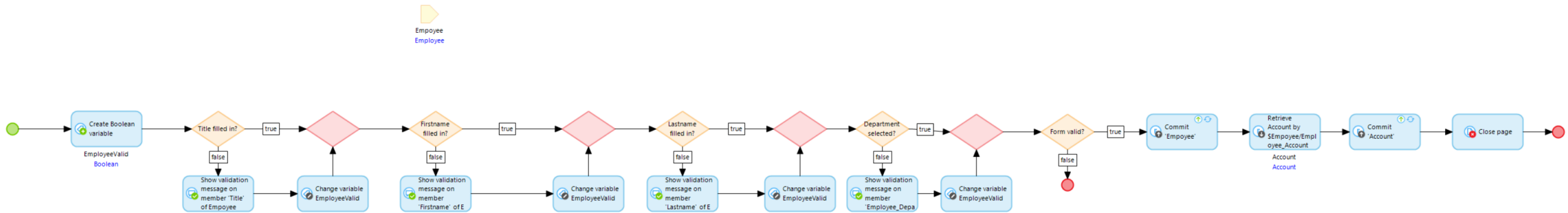
Responsive-Profil = Standardeinstellung (kann nicht gelöscht werden)

Mit einem Offline-Profil, können die Funktionalitäten von Power Web Apps (PWA) genutzt werden.

Mit Native Mobile können Seiten entwickelt werden, die nativ auf Android oder iOS ausgeführt werden.



Microflows



Mendix Assist, ein mit künstlicher Intelligenz ausgestatteter Agent, unterstützt Anwender bei der Konfiguration von Microflows. Microflows sind eine visuelle Ausdrucksform dessen, was normalerweise in Code wiedergegeben werden würde. So kann ein Microflow beispielsweise Objekte erstellen oder verändern, Seiten anzeigen und Entscheidungen treffen. In Studio liefert Mendix Assist dem Citizen-Developer außerdem Empfehlungen und Vorschläge für die nächsten Schritte und sichert die Qualität der Anwendungen.

Der große Unterschied zwischen Code und Microflows besteht darin, dass Microflows aktivitätsbasiert sind. (blauen Rechtecke) Die anderen Kernkomponenten sind die Teilung (mit ihrem Gegenstück die Zusammenführung), Ereignisse, Parameter, Schleifen und das Aufrufen anderer Microflows. Es gibt auch Aktivitäten, die Daten verarbeiten, die Runtime anweisen, eine Aktion auszuführen (z. B. das Abrufen von Daten von einem externen Dienst), oder den Client anweisen, eine Aktion auszuführen (z. B. das Öffnen einer Seite, das Schließen einer Seite oder das Herunterladen einer Datei). .) Werden von der Laufzeit ausgeführt. Bei einem Fehler, wird der gesamte Microflow rückgängig gemacht.

Toolbox

Filter

Expand All

Collapse All

Object activities

Cast object

Change object

Commit object(s)

Create object

Delete object(s)

Retrieve

Rollback object

List activities

Aggregate list

Change list

Create list

List operation

Action call activities

Java action call

Microflow call

Variable activities

Change variable

Create variable

Client activities

Close page

Download file

Show home page

Show message

Show page

Validation feedback

Integration activities

Call REST service

Call web service

Export with mapping

Import with mapping

Logging activities

Log message

Document generation

Generate document

Resources Interaction

Get json from resources

Decisions

Decision

Object type decision

Merge

Other

Annotation

Parameter

Loop

Events

Start event

End event

Error event

Continue event

Break event

Microflows

Microflows können auf viele Arten ausgelöst werden:

- Durch die On-Click-Aktion für eine Schaltfläche
- Durch das on-enter/on-change/on-leave-Ereignis auf UI-Elementen
- Durch das On-Load-Ereignis auf dynamischen UI-Elementen
- Von den Ereignishandlern einer Entität
- Aus dem Projektnavigationsmenü
- Aus einem benutzerdefinierten Navigationsmenü
- Aus einer Importzuordnung
- Aus einer Exportdienstdefinition
- Nach einem Zeitplan
- Nach dem Start Ihrer Anwendung
- Vor dem Herunterfahren Ihrer Anwendung
- Während einer Systemdiagnose (dieser Trigger wird regelmäßig ausgeführt und kann verwendet werden, um sicherzustellen, dass Ihre App normal funktioniert)

Element	Type	Graphic
Start Event	Event	
End Event	Event	
Sequence Flow	Flow	
Activity	Activity	
Decision	Decision	
Merge	Decision	
Parameter	Artifact	
Annotation	Artifact	
Annotation Flow	Flow	



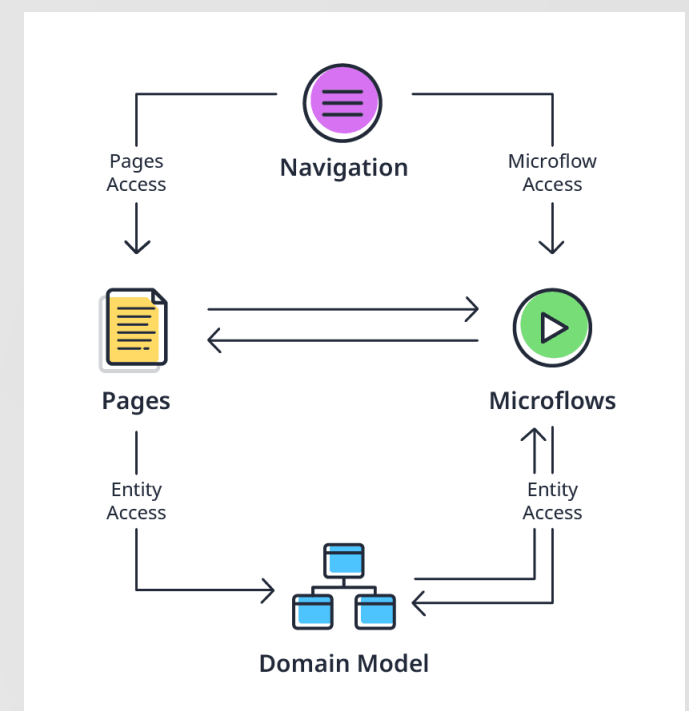
xPath

1. Abfragesprache, die zum Auswählen von Daten in XML-Dateien verwendet wird
2. Beim Schreiben von Abfragen kann XPath im folgenden Kontext verwendet werden:
 - a. Abrufen von Aktivitäten in Microflows.
 - b. XPath-Datenquelle in Daten-Widgets.
 - c. Registerkarte XPath-Einschränkungen in Sicherheit.



Security

1. In der Sicherheit gibt es zwei wichtige Konzepte: Authentifizierung und Autorisierung. Die Authentifizierung ist die Bestimmung der Identität des Benutzers, während die Autorisierung bestimmt, worauf der Benutzer Zugriff hat.
2. Authentifizierung
 - a. Mendix bietet standardmäßig eine grundlegende Kennwortauthentifizierung und bietet die Integration mit Authentifizierungsdiensten von Drittanbietern über Standardprotokolle wie OpenID und SAML.
3. Genehmigung
 - a. Das Mendix-Berechtigungsmodell ist in die Plattform integriert.
zwei Teile aufgeteilt: Projektsicherheit und Modulsicherheit.
 - i. Auf Projektebene konfigurieren Sie allgemeine Sicherheitseinstellungen wie die Passwortrichtlinie.
 - ii. Auf Modulebene konfigurieren Sie den Zugriff auf Seiten, Microflows, Entitäten usw.



Ende

