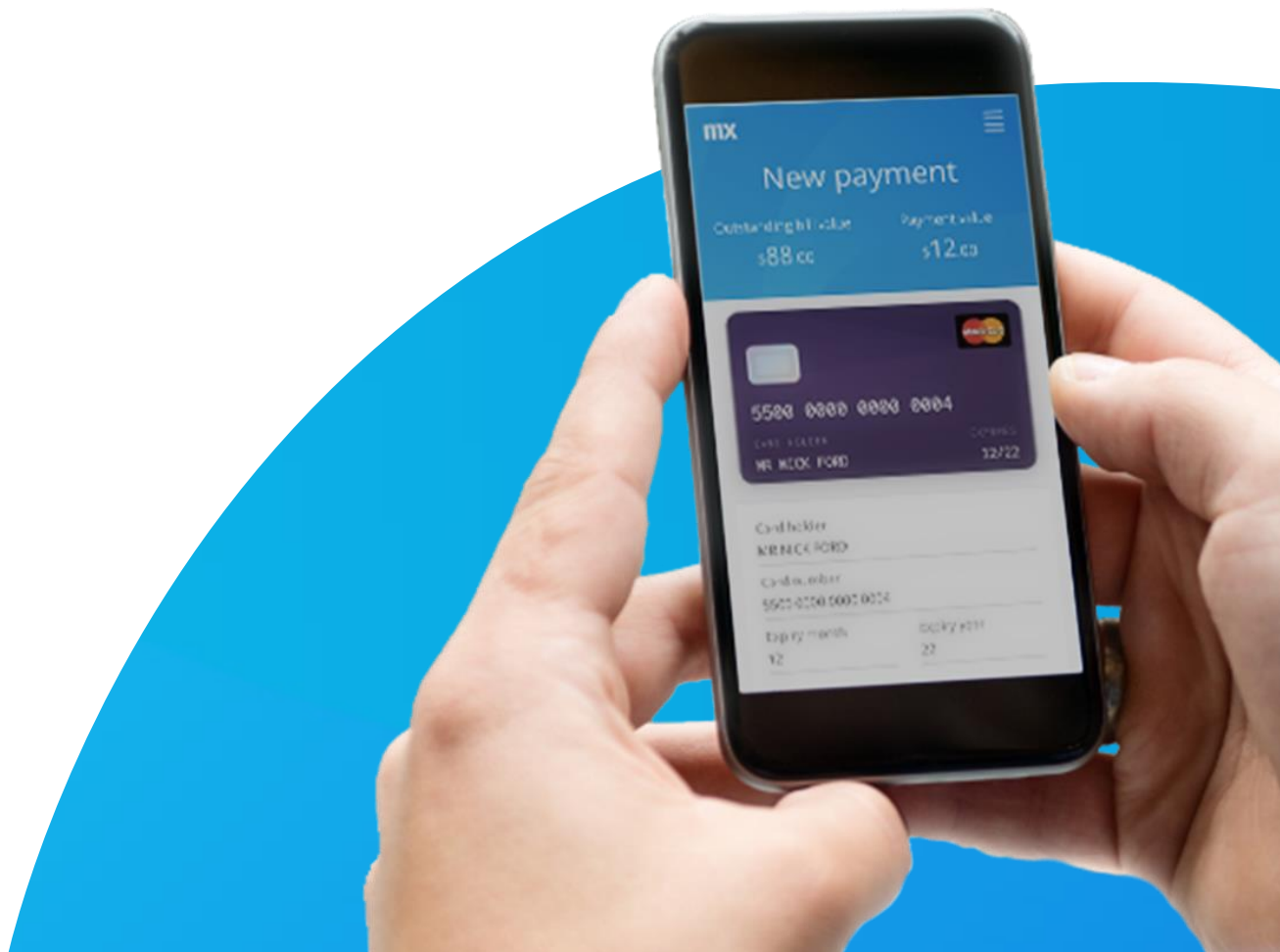


Become a Rapid Developer

Module 3

Start Building Your App



3. Start Building Your App

In this module, you will start building your app using the Desktop Modeler. You will learn how to create and edit pages and buttons. You will also dive into the Domain Model, which is your tool to define your data structure.

3.1 The Next Phase: Building Your App

Now that we have completed the **Capture** phase, by defining the User Stories and creating the Wireframe, it's time to actually start developing our App. The **Develop** phase.



3.1.1 Planning

The first phase of building any app is taking a look at the user stories and prioritize them to see which one would be most useful to start with. In general, the most important user story is the one that immediately generates value for the user, and which can serve as a starting point to further expand and enhance your app. As soon as you have defined which User Story to start with, you know the tasks you need to complete within that User Story, so you can actually start developing.

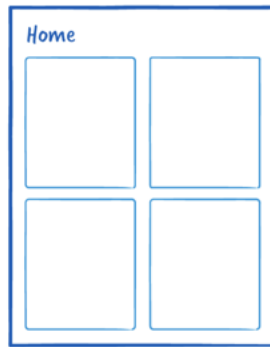
In Jimmy's app, the main user will be Jimmy. He's the administrator of the app. Therefore, the user story “**As an administrator, I want to be able to easily access and manage my data, so I can run my company more efficiently.**” looks like a good place to start, as Jimmy can then immediately start moving the most important data from his Excel sheets into the app. This means this story adds a lot of value for Jimmy! Completing this user story would involve the following tasks:

1. Have a homepage that gives access to pages where information can be seen.
2. Adjust the homepage layout to accommodate for the buttons to be created.
3. Add the buttons.
4. Add the pages to which the buttons will link.
5. Link the buttons to these pages that show the information in the system.

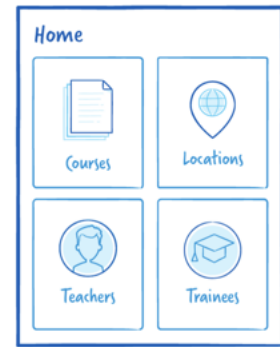
① Create Homepage



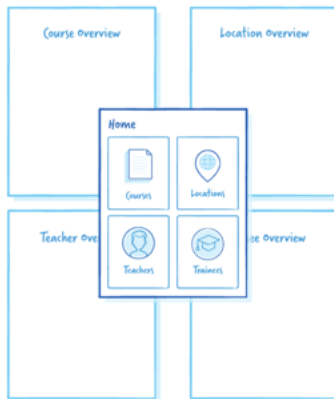
② Create page layout



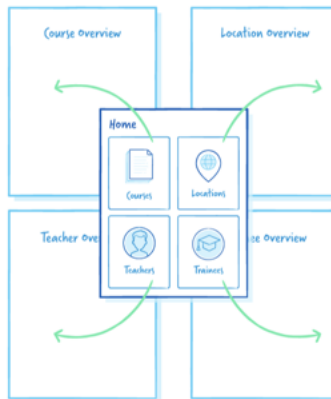
③ Add buttons



④ Add pages



⑤ Link buttons to pages



3.2 The Desktop Modeler

As we've seen in the previous paragraph, the first task is to create a nice landing page for the app: the homepage! From the homepage, Jimmy will be able to easily access the pages he wants.

You are going to use the Desktop Modeler to build Jimmy's app. With the Desktop Modeler you can easily create beautiful pages and set up your data structure. With the Desktop Modeler, you also have immediate access to all the awesome advanced features that the Mendix platform has to offer. You are going to learn more about these features in the upcoming modules!

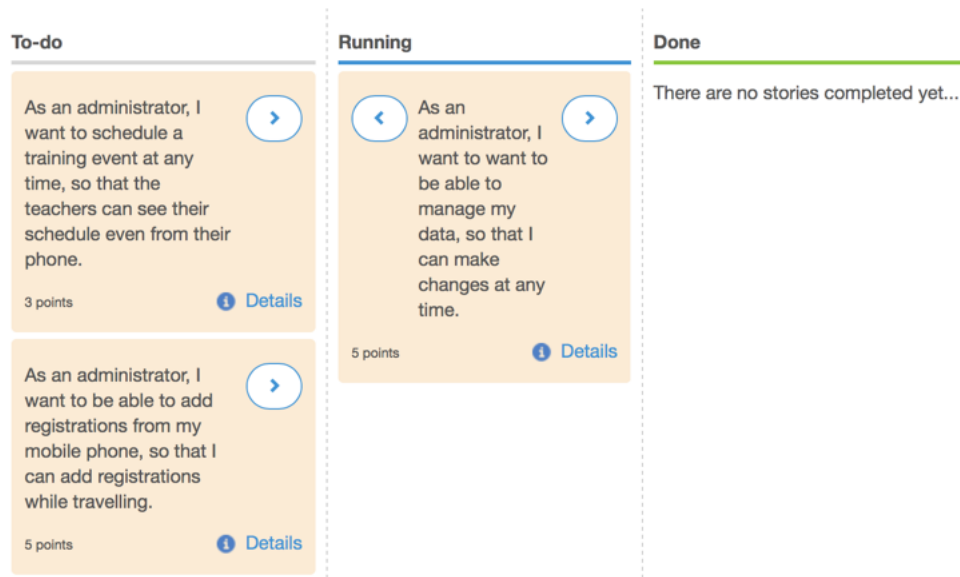
3.2.1 Open your App in the Desktop Modeler

- 1 First, to show the progress in your Scrum board, set the first story to **Running** by clicking the grey **To-do** button.

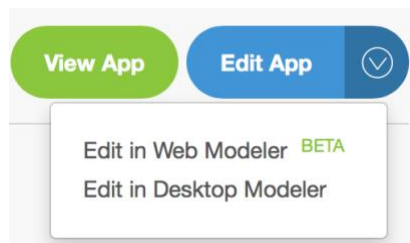
As an administrator, I want to be able to manage my data, so that I can make changes at any time. To-do

5 points • ID 1874807 • [Details](#) • [Comment](#) • [3 Tasks](#)

2 This is how it will look under the **Planning** tab:



To start building the homepage, open your app in the Desktop Modeler. Go to the Development Portal, click the arrow next to **Edit App**, and then select **Edit in Desktop Modeler**.

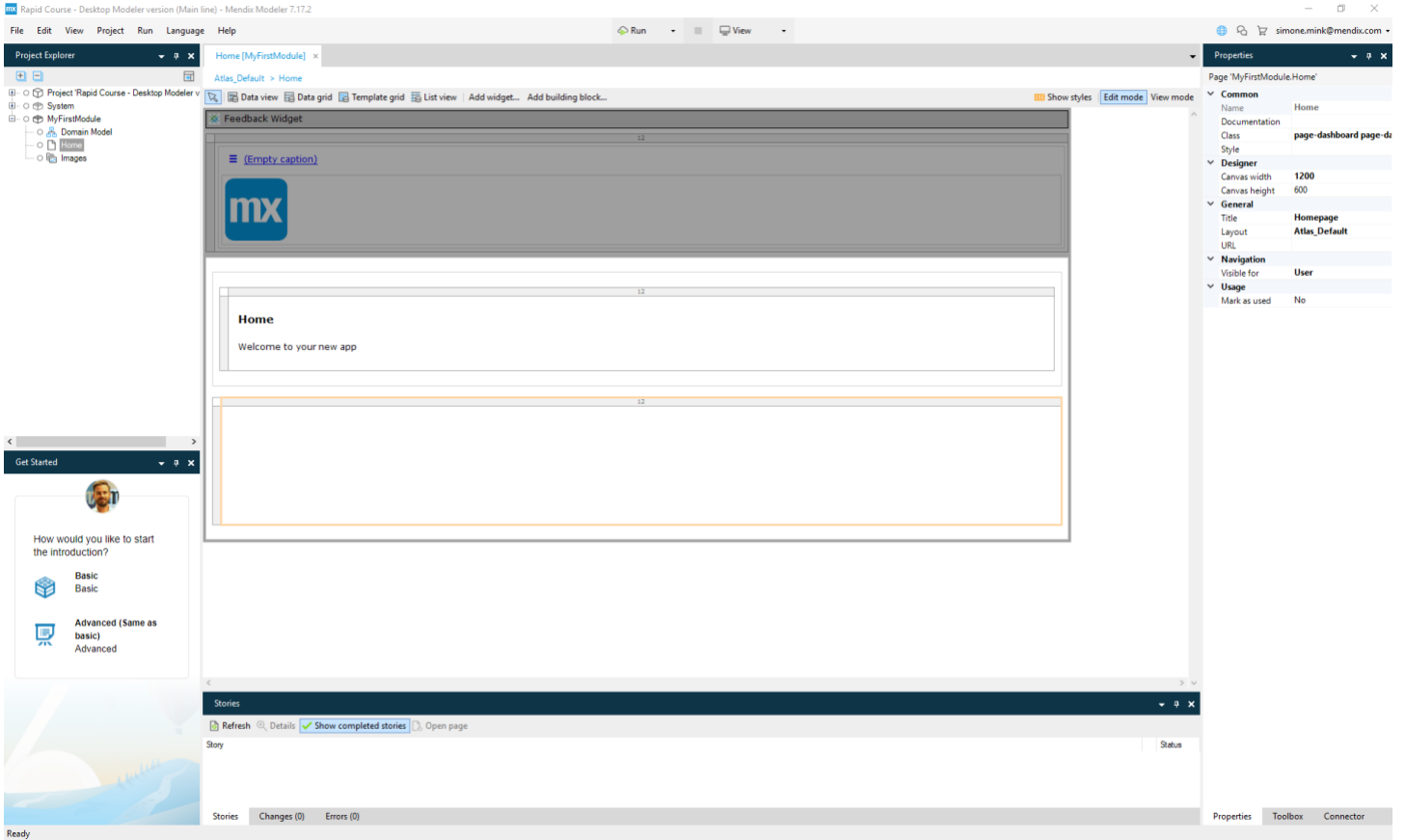


- In case you haven't installed the Desktop Modeler yet, a screen that will allow you to download the latest version directly will show first.
- In case you have multiple versions of the Desktop Modeler installed, you will get a pop-up asking you to switch to the **VersionSelector**. Click the blue confirmation button. This will check whether you already have the correct version of the Desktop Modeler installed. Select the latest version of the Desktop Modeler and click **OK**.

It then proceeds to open the Desktop Modeler.

3 The Desktop Modeler opens. Sign in using your Mendix credentials.

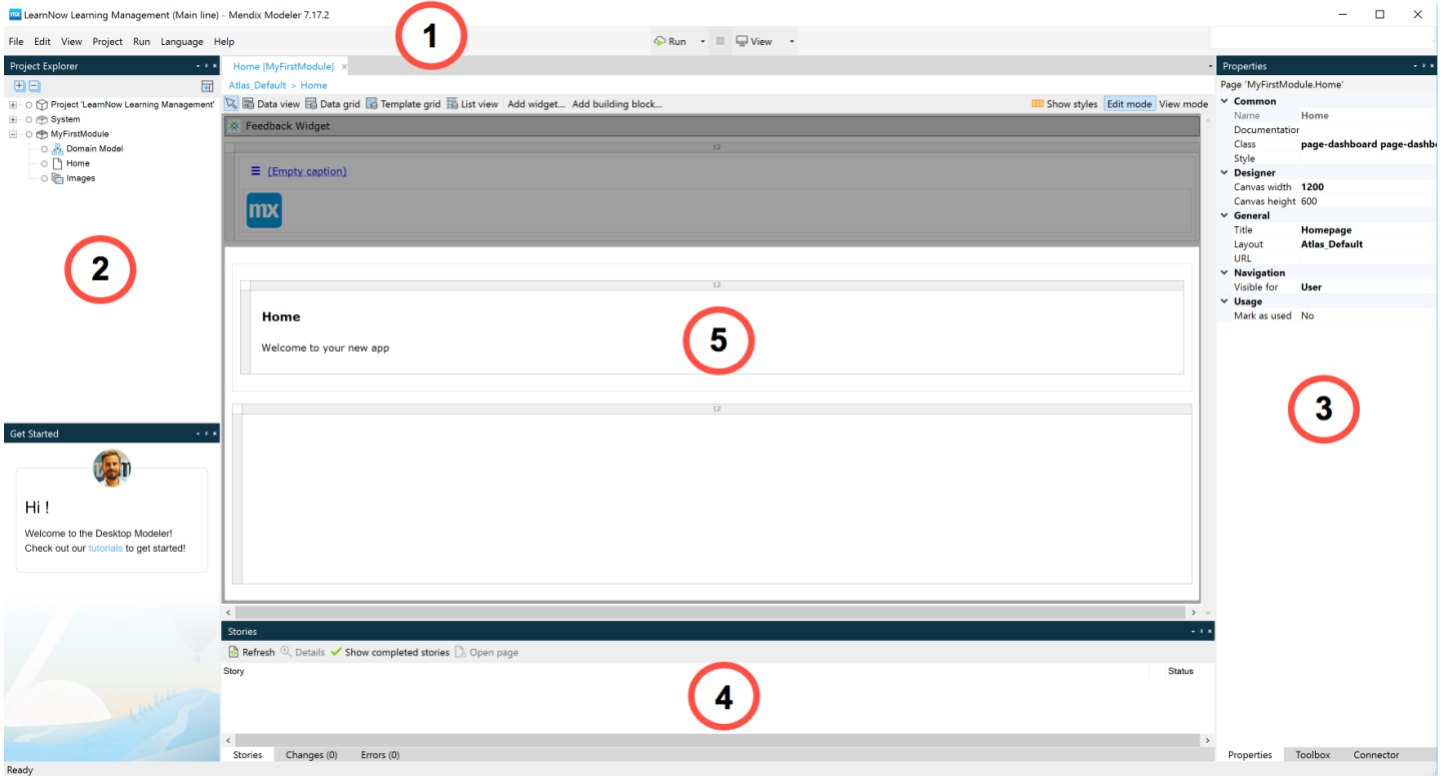
4 The LearnNow Training Management app will automatically begin to download from the Team Server. You will learn more about the Team Server in Module 6. Once it's done the Desktop Modeler will open the project and you can get started!



Continue to the next lecture to learn more about all the different tabs and areas in the Desktop Modeler.

3.3 Exploring the Desktop Modeler

To make sure you have the same Desktop Modeler layout as the one used in this course, click **View** and then **Reset layout**.



1. First of all, at the top of the Desktop Modeler is the control bar. These are the menus and buttons, from left to right:
 - **File** – open and close documents and projects
 - **Edit** – all the options related to editing files, finding things in your project, and setting your preferences
 - **View** – from here, you can open all available tool windows
 - **Project** – all the functions related to working with Team Server (more about this later on)
 - **Run** – run and preview your app
 - **Language** – manage the language settings and access to language operations
 - **Help** – quick access to help environments
 - **Run** – deploy your app in the cloud or locally
 - **View** – preview your app
 - A button for opening your app in the Web Modeler
 - A button for opening the app **Project Dashboard** in the Developer Portal
 - A button that will take you to the Mendix App Store
 - The last dropdown menu provides access to all your apps, the Developer Portal, your profile, and the sign out button

2. At the top-left of the Desktop Modeler, you see the **Project Explorer** (underneath the control bar). From here, you can access all the resources of the app (pages, microflows, and navigation, for

example).

Underneath the Project Explorer you will find the **Get Started** pane. You will not need this pane throughout this course. Go ahead and close it by clicking **x** in the blue topbar of the pane.

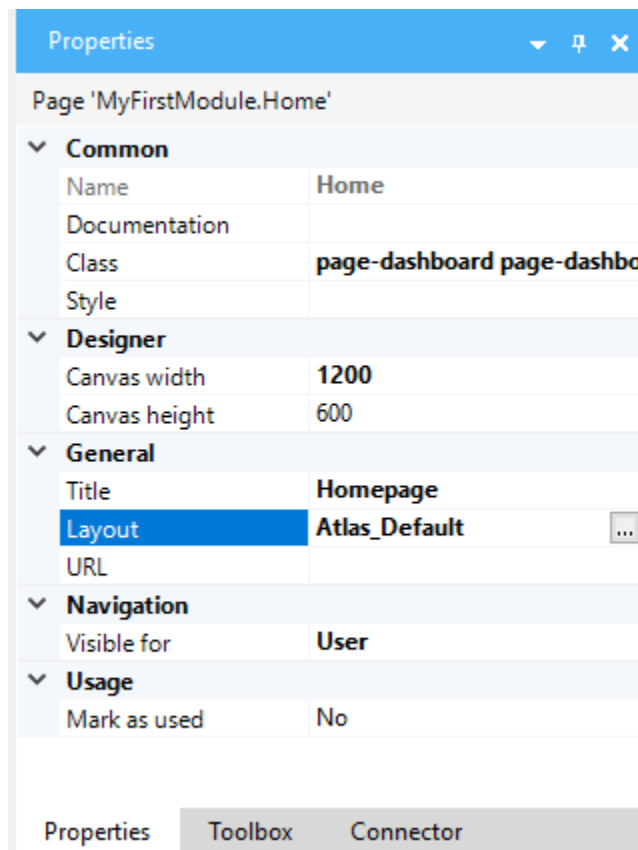
3. On the right is the **Toolbox**, **Properties**, and **Connector** panes.
 - You can use the **Toolbox** to place widgets on a page. With the building blocks, you can add predesigned and completed templates to your page, so you can build apps even faster!
 - On the **Properties** pane, you change the properties of the page or element that is currently selected.
 - You can use the **Connector** to quickly connect data to pages.
4. At the bottom, you will find the **Stories**, **Changes**, and **Errors** tab. Here you can see the stories in your current sprint (and change their status). You can also see all the resources you have added or changed to the project and whether you have errors (and where they are).
5. The center of the Desktop Modeler is the editor. This is where you will build all your resources. At the top right corner of the editor you will find the **Edit mode** and **View mode** buttons. You can use these buttons to quickly switch between building and previewing your app. In **View mode** you can also select on which device type you want to preview the app: desktop, tablet or mobile.

3.3.1 Create the Home Page of your App

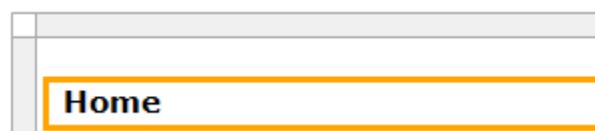
The Desktop Modeler opened with the Home page already opened, ready for you to get started! Isn't that convenient? You can now build a beautiful homepage for Jimmy!

1. The layout defines how a user will experience the app and can navigate through it in general. Under **Properties**, when you have no element selected, you can change the **Layout** of each page:
 - The **Atlas_Default** layout shows the navigation menu at the sidebar
 - The **Atlas_TopBar** layout shows the navigation menu at the top of the page
 - The **PopupLayout** layout shows the page as a pop-up

For now, let's stick to the **Atlas_Default** layout. It is important to stay consistent with the layout you select throughout your app. If you have different layouts on each page, your users will be confused.

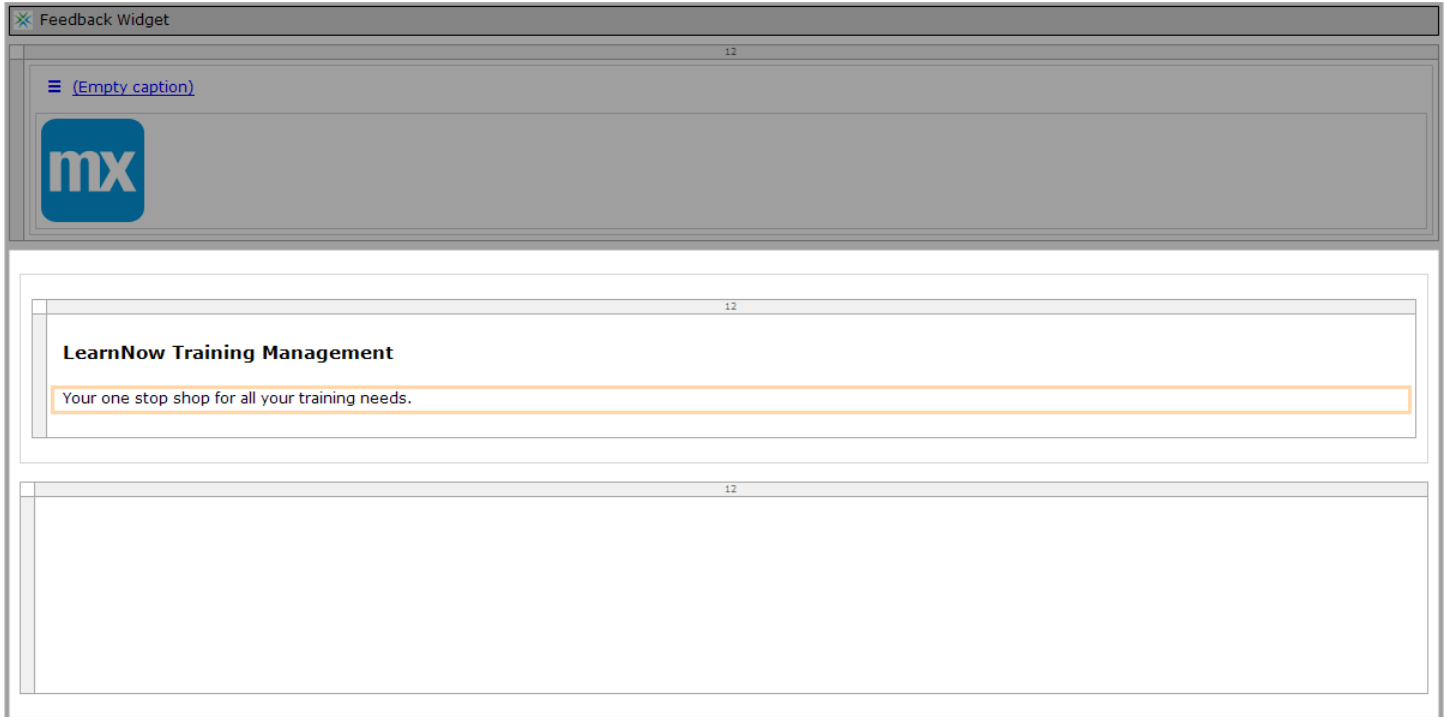


2. To change the text appearing in a page, click on the text field and start typing!



Change the big title on the page that says **Home** to **LearnNow Training Management** and think of a nice subtitle to really welcome the users of your app! How about 'Your one stop shop for all your training needs.'?

3. This is how it should look like at the end:



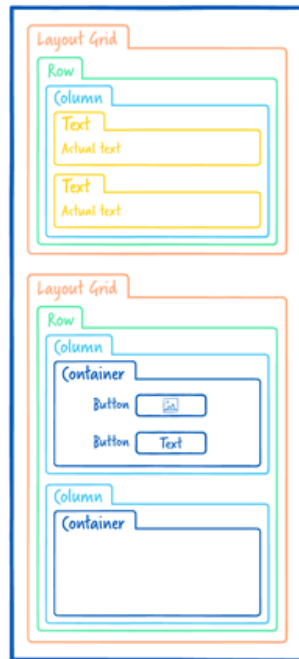
Excellent! You've set up the basics for a nice homepage. Go to the next lecture to see how you can add some extra columns to this page!

3.4 Structure the Layout

You have now created the homepage for the LearnNow app! Before you can add buttons, you need to structure the homepage in a way to accommodate these buttons. For that, you will use the layout grid, containers, and widgets.

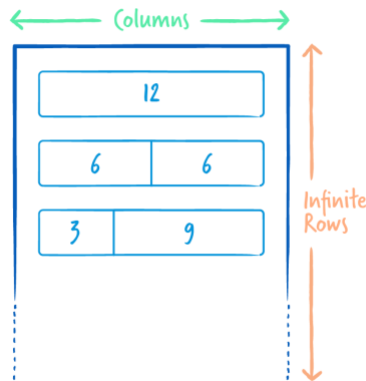
You will learn more about these components in this lecture and you will also use them in the upcoming assignment.

Navigation Layout



Layout Grid

The layout grid is a widget that gives structure to your pages. It contains one or more rows, and each row contains 1 to 12 columns. Each column has a weight (a number from 1 to 12), and the total of the weights of the columns in a row must always add up to 12 (in the example below: one column with weight 12; two columns with weight 6; one of 3 and one of 9).



You can use the layout grid to create different layouts for mobile, tablet, and desktop. For example, you can show four buttons underneath each other on a mobile device (so they are full-width and easy to read and use) but show them all on one row on a desktop device in order to make better use of the available space on the screen. The power of the layout grid is that you can achieve this without having to create additional (device-specific) pages!

Note:

- The **layout** determines what kind of page it is (device-specific, responsive, or a pop-up) and what the position of the menu is.
- The **layout grid** is a widget you use to build the structure or layout of the page. Is the page content divided into columns or not? If so, how are these columns shown on each device type?

Containers and Widgets

Once you have created the page structure (the layout), you start building the content of the page by adding different kinds of widgets. These can be, for example, text widgets to display text or button widgets.

You can place these widgets inside containers. There are a few reasons to do this:

- You can group the widgets. This will allow you to move them all at once (if needed), or put a border around them to group them visually for your end-user. You can also group them to hide all of them at once.
- You can add multiple widgets in a container to give them consistent styling.
- You can use the container to create spacing and alignment in pages (for example, to align a group of buttons to the right side of the page, or to create spacing around a button).

A container is also considered a widget.

Take a look at the information that the LearnNow app will need to contain:

- All the courses LearnNow provides
- The locations where training events can be hosted
- The teachers that work for Jimmy
- The trainees that have signed up and/or will sign up for the training events

For each of these items, Jimmy would like to have a page where he can manage the information (for example, to create a new course or change a trainee's email address). He would like these pages to be accessible directly from the homepage. So, let's start with that! Take a look at the wireframe, so you know what the design needs to be.



Responsive pages

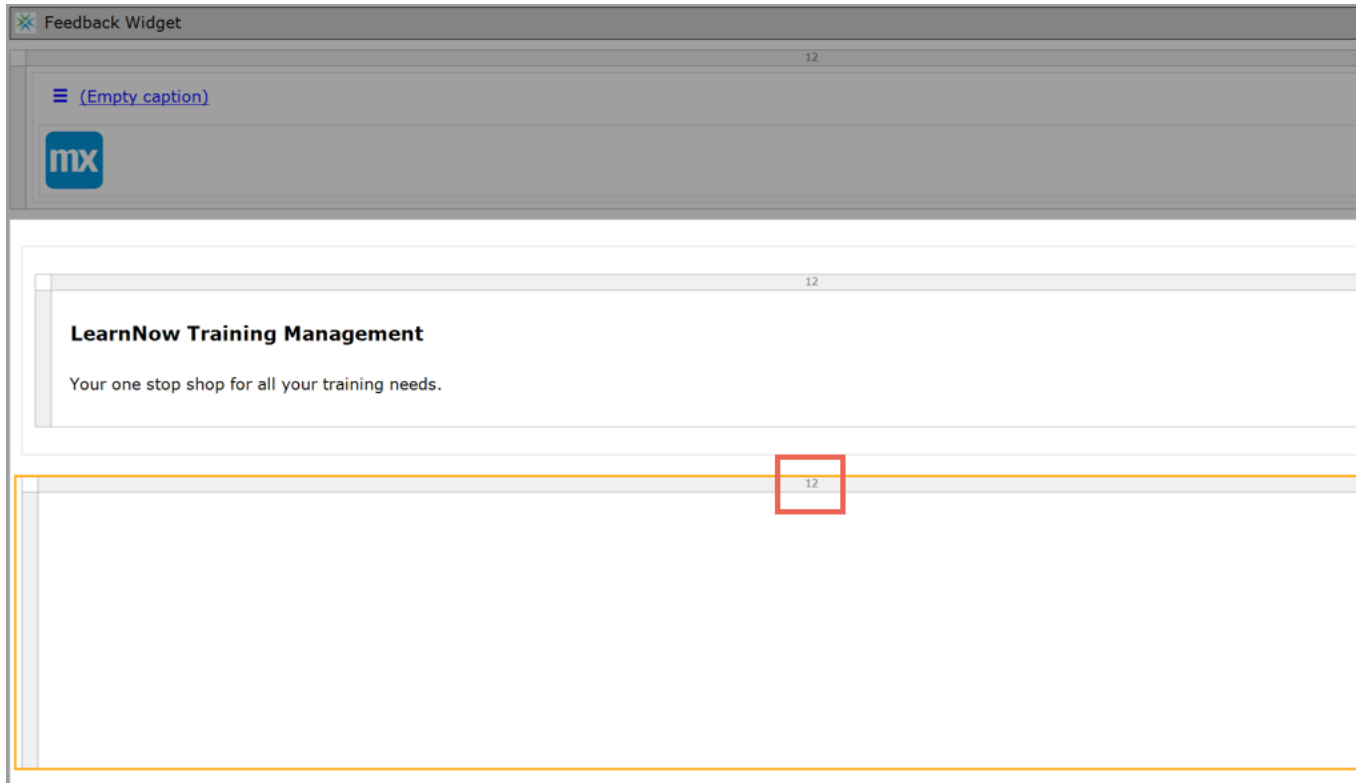
As you can see, the homepage needs four buttons. But the page looks a little bit different on different devices. They will be shown underneath each other on a phone, two by-two on a tablet, and all four on one row on a desktop (or laptop). This is what is called a responsive page, when the same page looks different on different devices. Responsive pages allow you to create a nice user experience on every device type, without having to build different pages for each device.

However, your app currently only has one column, not four. Let's fix this!

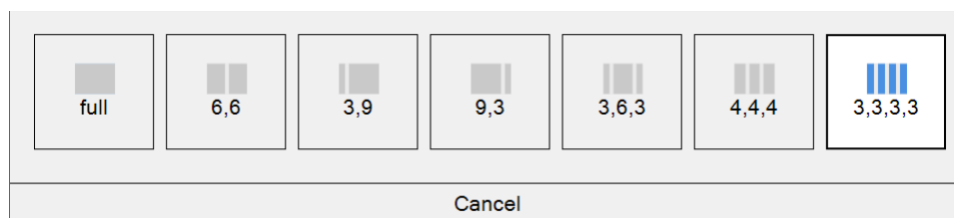
3.4.1 Add Columns to Your App

In this assignment you will adjust the layout of your homepage, so it matches your wireframes for each device!

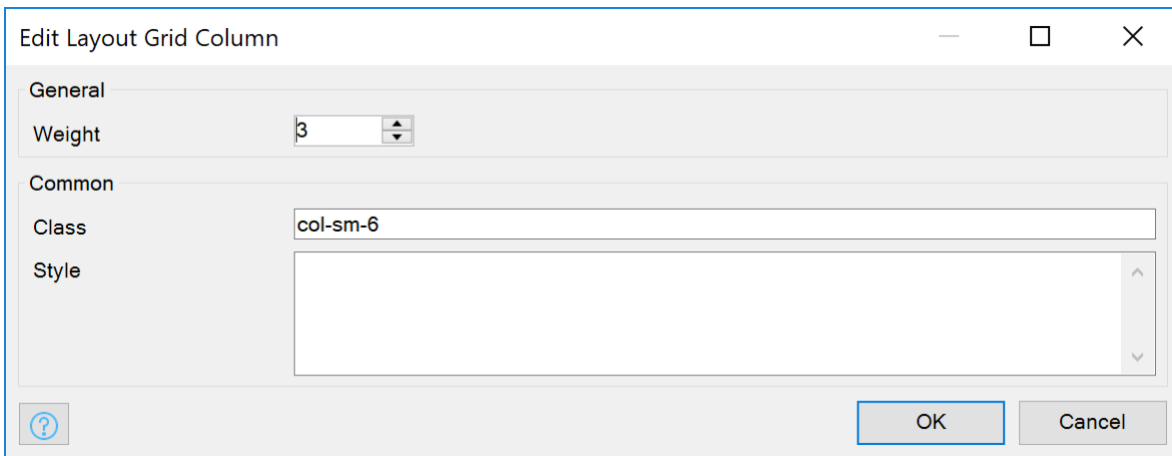
1. In the Desktop Modeler, with the homepage opened, take a look at the bottom layout grid. It says **12**, which means the layout grid consists of one full width column.



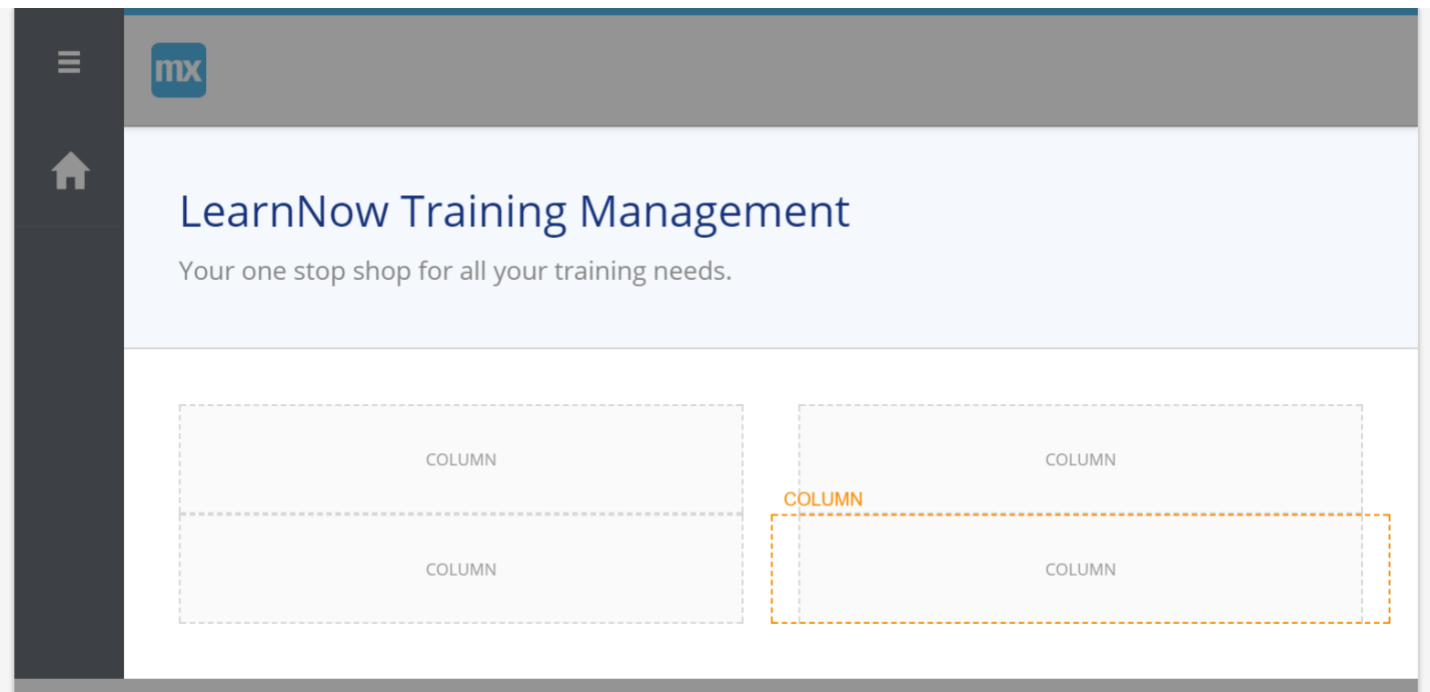
2. To get the desired layout for the homepage, you need to change this one column to four columns. The fastest way to achieve this, is to delete the existing layout grid and place a new one. Select the bottom layout grid and delete it. Click the white square in the top left corner of the layout grid and press **Delete** on your keyboard.
3. Go to the **Toolbox** pane. In the **Widgets** section, search for the **Layout grid** widget.
4. Drag and drop it below the LearnNow Training Management container (where the layout grid you just deleted used to be).
5. You will be prompted to select your preferred layout. Select the **3,3,3,3** option. This will create a layout grid with four columns. All four columns will have a weight of **3** and the total weight of the row will add up to **12**.



6. Click **View mode** on the top right of the editor to preview the homepage. Click on the phone, tablet, and desktop icons in the middle, to see how your app will appear on different devices. Is that correct according to your wireframes? The layout for desktop and mobile is, but the one for tablet still needs some work. Currently the four columns are displayed underneath each other, but that doesn't match your design.
7. Click **Edit mode** to go back to the editor.
8. Double-click the left most column of the new layout grid to open its properties. Set the **Class** to **col-sm-6**. This will make the column display as a column of weight 6 on a tablet device.



9. Repeat this for the other three columns. Once you are done, click **View mode** again to check the result.



Congratulations! You just created columns that match your wireframes for the different devices. Continue to the next lecture to learn how you can add buttons.

3.5 Add Buttons

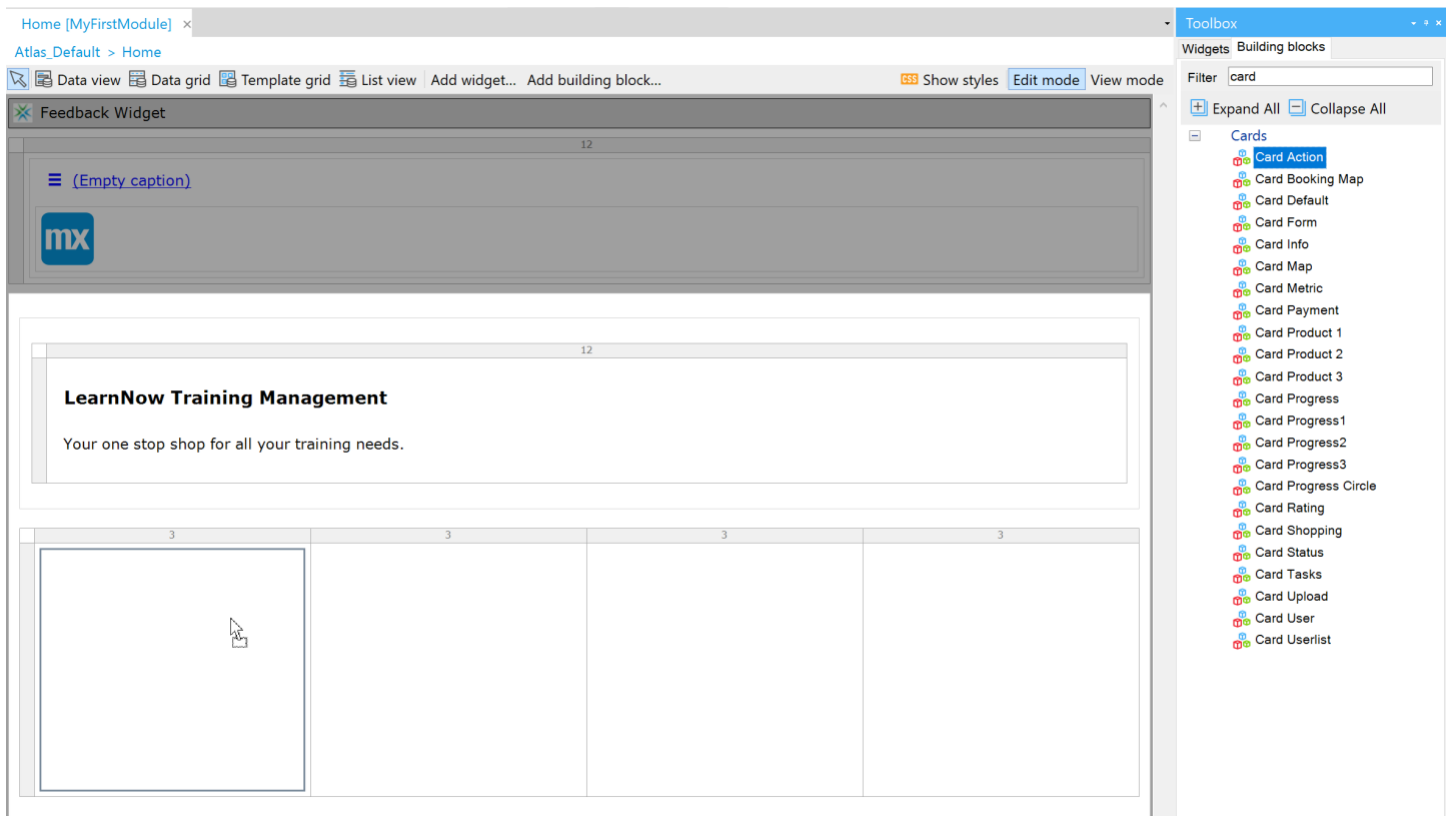
After creating a layout for the home page, it is time to add buttons. These buttons will allow Jimmy to navigate to the other pages from the home page. It doesn't matter that these pages aren't there yet; you will create them after you have created the buttons.

To add buttons, you can use building blocks to quickly add predefined (button) templates to your page. The Mendix building blocks have been designed by the Mendix UX Team, so they ensure that you are adding a user-friendly section to your page. You can look at it as though it's paint by numbers, as the outline is already created for you. All you have to do is drag and drop the building blocks into your page and then (if needed) connect them to your data or pages.

You can always change a building block to your preferences once you've placed it.

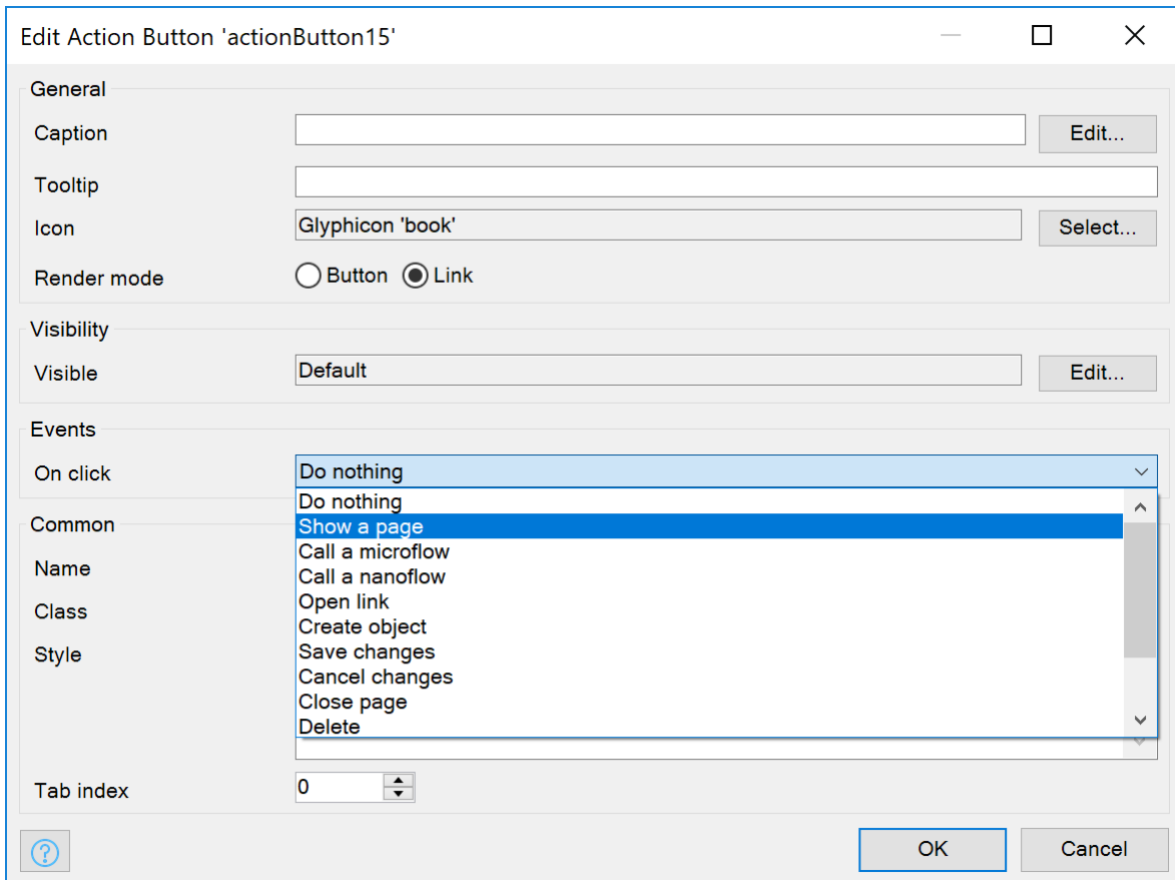
3.5.1 Add Buttons to Your Home Page

1. Open the **Toolbox** pane and search for the **Card Action** building block. Drag this building block onto the first column.



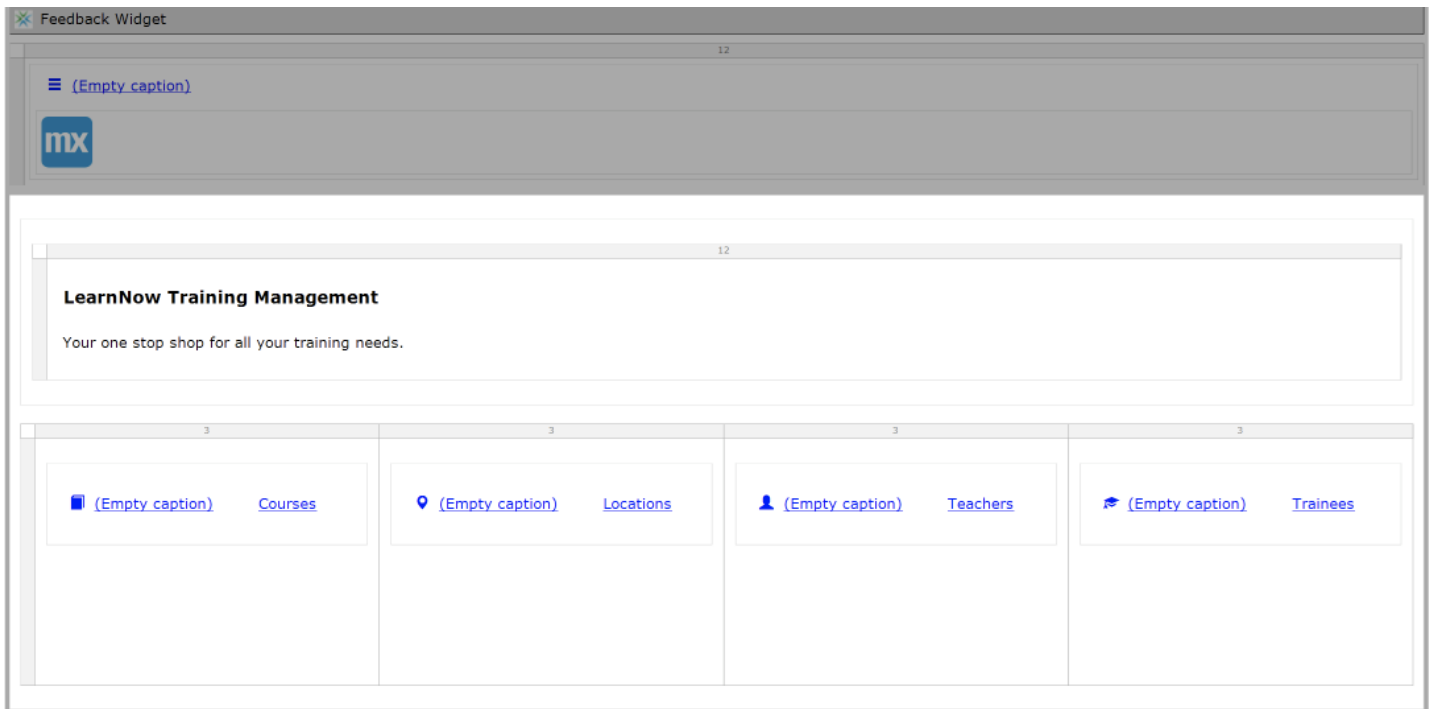
2. Also, change the **Open Page** caption (just like you changed the title in the header) to something that clearly describes which bit of information (data) this button will take you to. Remember: you are creating buttons for the courses, locations, teachers, and trainees overview pages. So these would be good captions for these buttons: Courses, Locations, Teachers, and Trainees.

3. Lastly, you can change the icons to your liking. To do that double-click the icon to open its properties. Click **Select** next to where it says **Glyphicon 'signal'** and search for the icon you want. In this example the **Book**, **Map marker**, **User**, and **Education** icons were used.

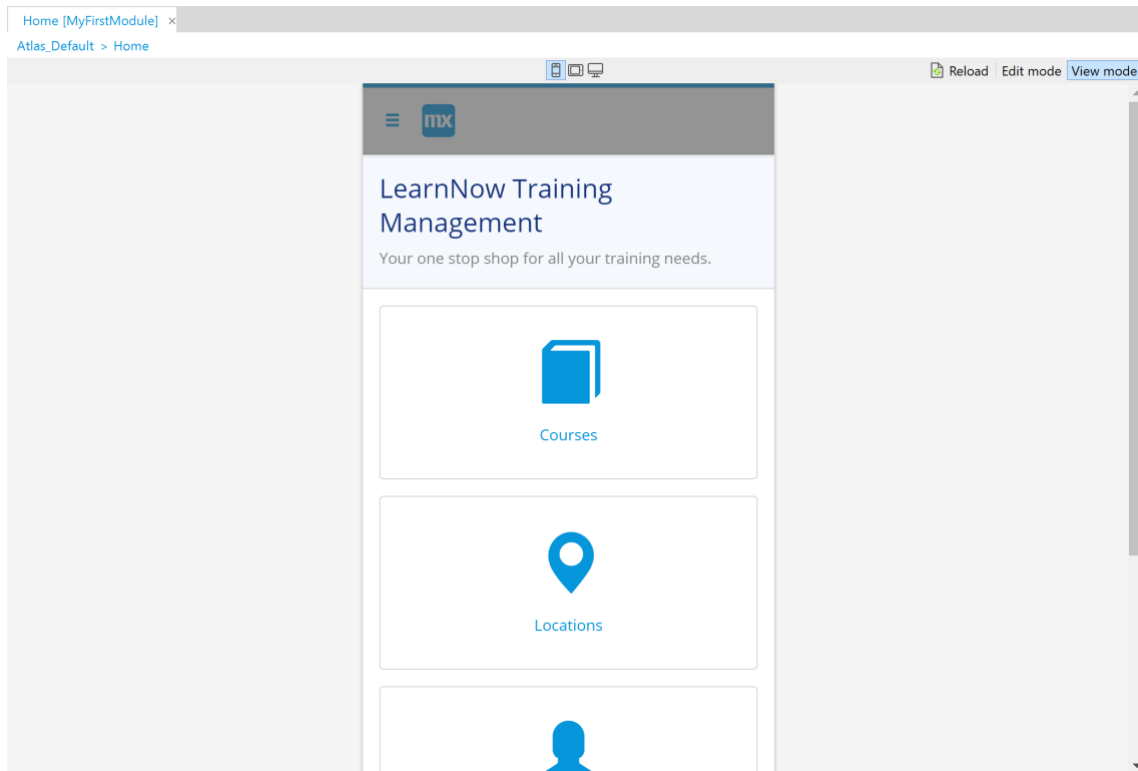


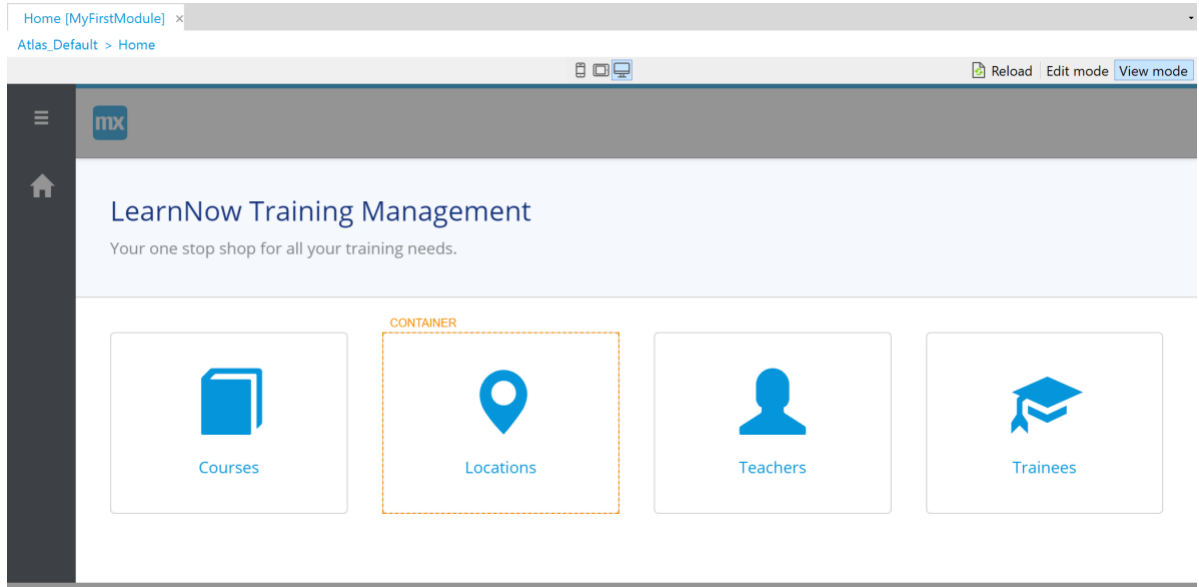
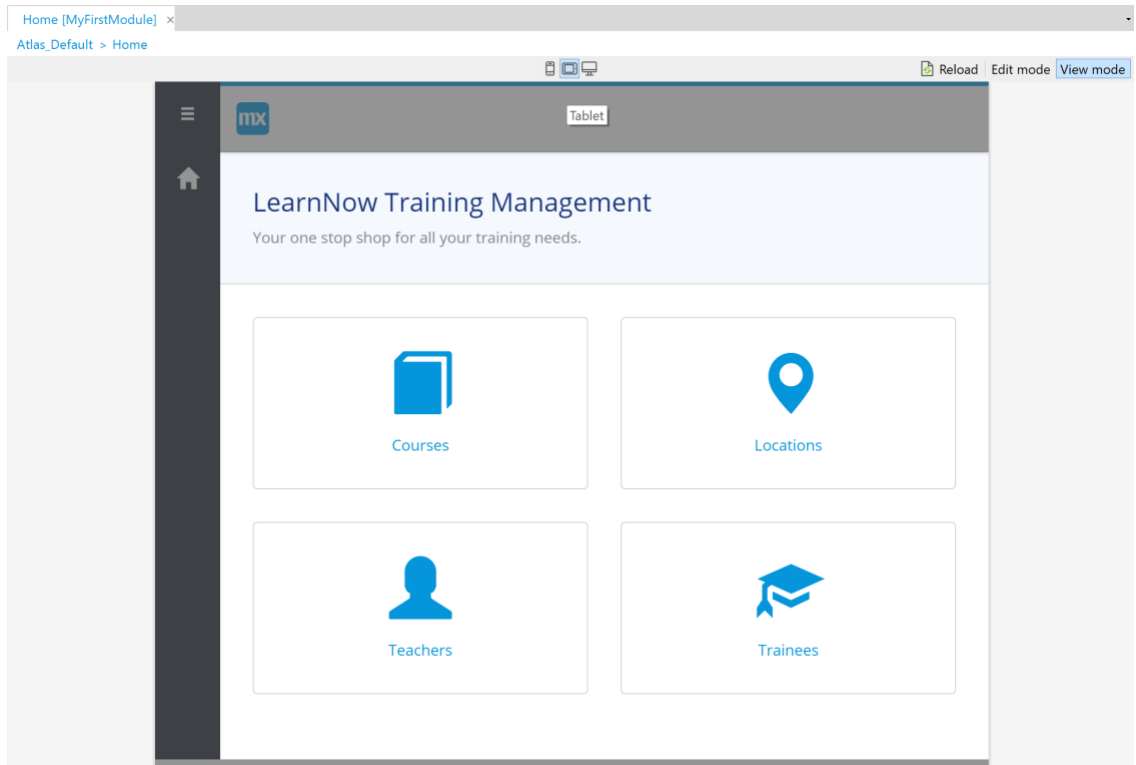
Don't worry about the buttons not going anywhere yet, that is the next task in this user story. This task is all about adding the buttons to the homepage.

4. This is what your homepage should look like now:



5. This is what your homepage is going to look like on a phone, tablet, or laptop:



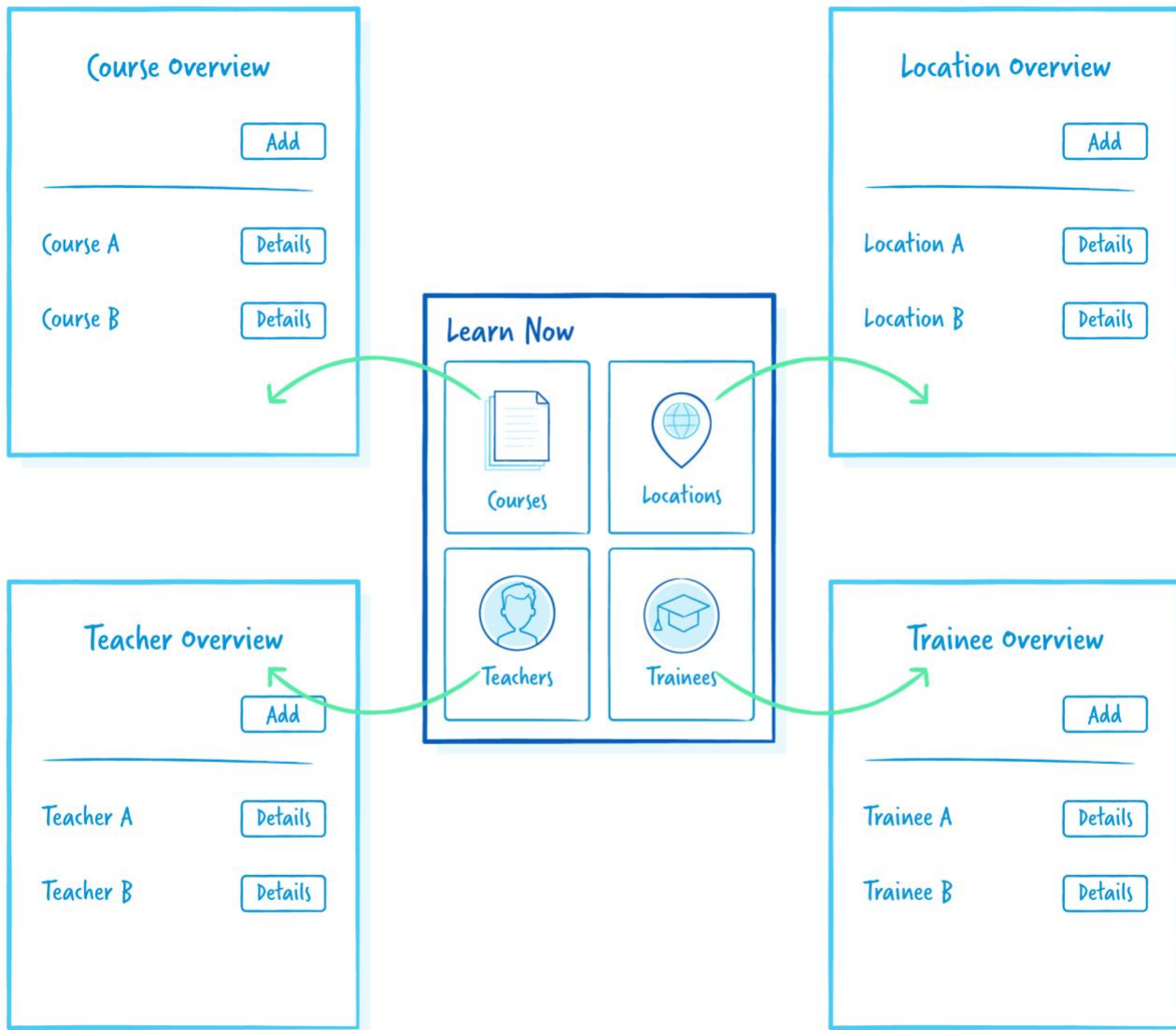


Awesome! You have added all the buttons to the homepage and gave them nice icons and captions! Now it's time to make sure that these buttons actually go somewhere: to the correct overview pages!

3.6 Create Pages

When a page has the appropriate layout and buttons, the next step is to have these buttons actually lead somewhere.

In our use case, the next task in the user story is to make the buttons on the homepage go to the pages where Jimmy can see and manage all the information that is needed to start planning training events. So, in this step, you are going to create the pages for the “root data” (courses, locations, teachers, and trainees) and link the buttons on the homepage to those pages.



You can see that now you have only one page, the homepage. You need to create the pages where the buttons will redirect you to.

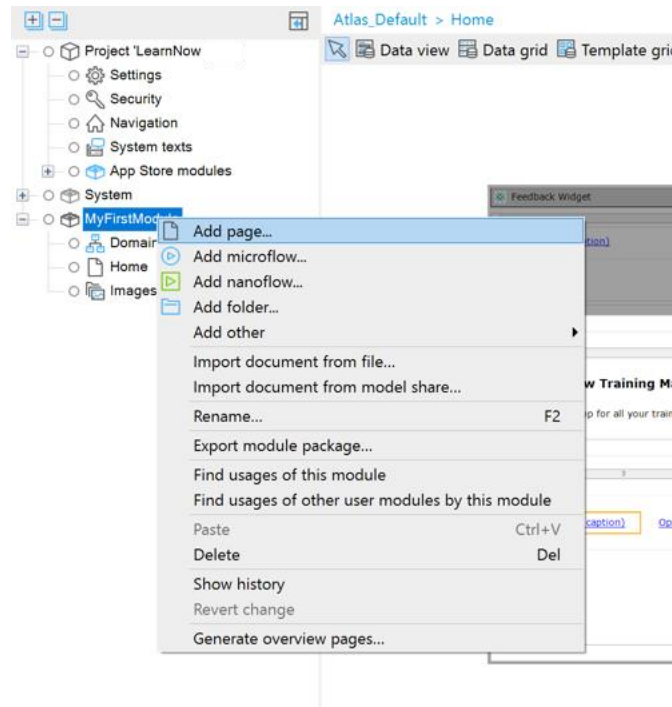
When creating pages:

- Give the page a clear **Page name**. The name should describe the information the page will show and what kind of page it is. For example, a good name for the page that will contain a list of all the existing courses is *Course_Overview*.
- You also have to select your **Layout**, which determines whether it's a full screen page or a pop-up page. For full screen pages, it also determines where the navigation menu is shown: at the top of the page or in a sidebar. It is best to be consistent when it comes to using layouts, especially regarding your main navigation. So, within one app, stick to the layout option that you selected for your first page. In this case, that will be the **Atlas_Default** layout.
- Last but not least, you have to select a page template. The Mendix Platform comes packed with all different types of page templates, from dashboards to wizards. These templates have been created by our UX Team and are an easy way to quickly create beautiful and user-friendly pages. In this case, you are going to use a template from the **Lists** section, because the page is an overview page that will show a list of items. The **List Default** template looks a lot like your wireframe, so let's go with that one.

3.6.1 Create Additional Pages

In the previous assignment you created buttons. These buttons need to lead to pages. Time to create these pages!

1. To create a new page, right-click **MyFirstModule** in the **Project Explorer**. Then click **Add page...** to start creating the page.



2. In the **Create Page** screen, use the following settings:
 - Page Name: `Course_Overview`
 - Layout: **Atlas_Default**
 - Page template: **List_Default**, from the **Lists** section.

Create Page

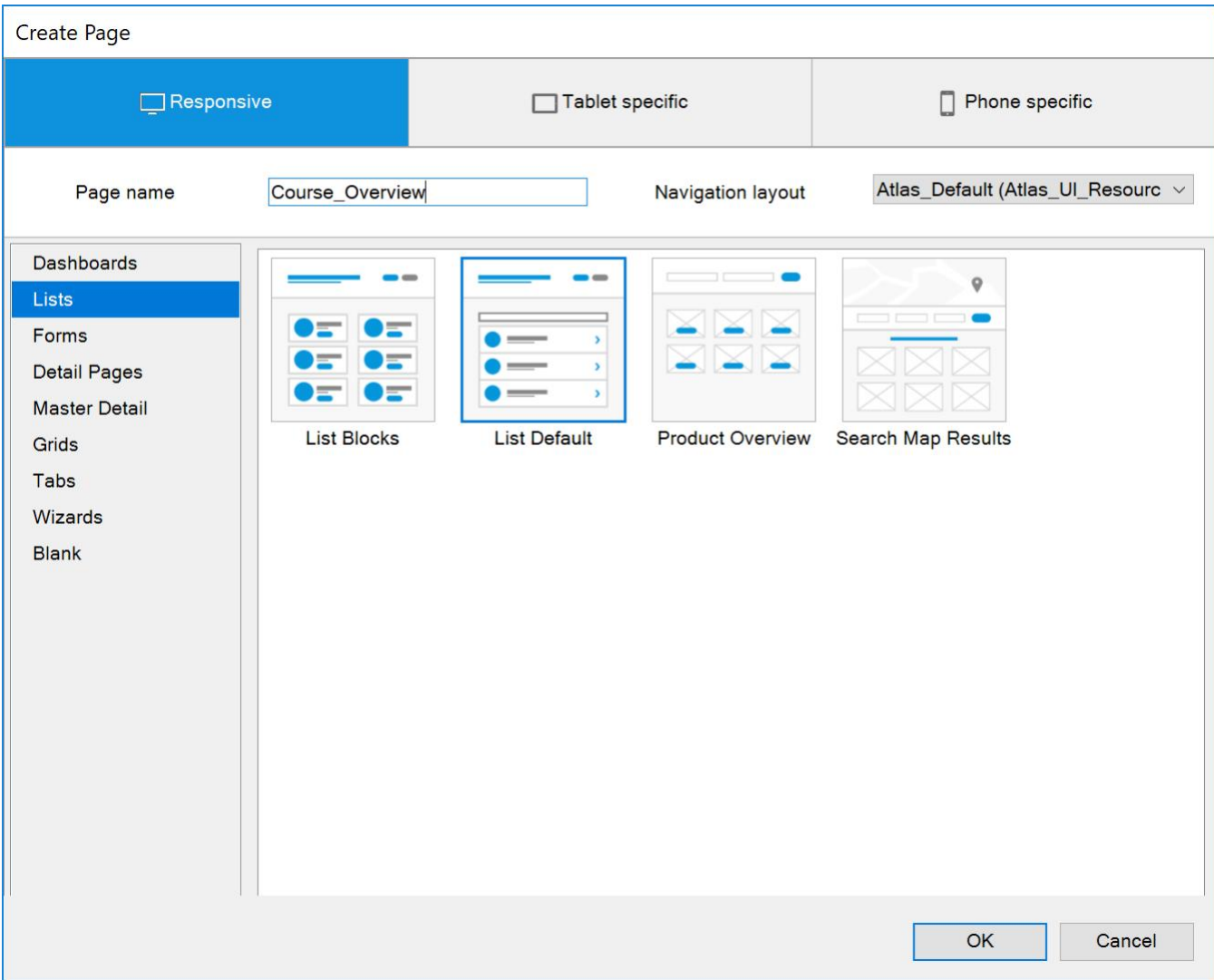
Responsive Tablet specific Phone specific

Page name Navigation layout ▾

Dashboards
Lists
Forms
Detail Pages
Master Detail
Grids
Tabs
Wizards
Blank

List Blocks List Default Product Overview Search Map Results

OK Cancel

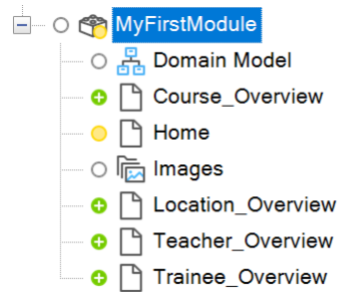


- When you are finished, click **OK**.
- Using these steps above, create each of the pages Jimmy needs right now:
 - Page name:** Location_Overview
Layout: Atlas_Default
Page template: List Default, from the Lists section
 - Page name:** Teacher_Overview
Layout: Atlas_Default
Page template: List Default, from the Lists section
 - Page name:** Trainee_Overview
Layout: Atlas_Default
Page template: List Default, from the Lists section



If you want to know more about the different page templates that Mendix offers, check out the [Atlas UI](#) site.

5. This should result in the following pages being available at the end of this assignment:



Let's make sure to connect your newly created pages to the buttons on the home page!

3.6.2 Link Your Pages to the Buttons

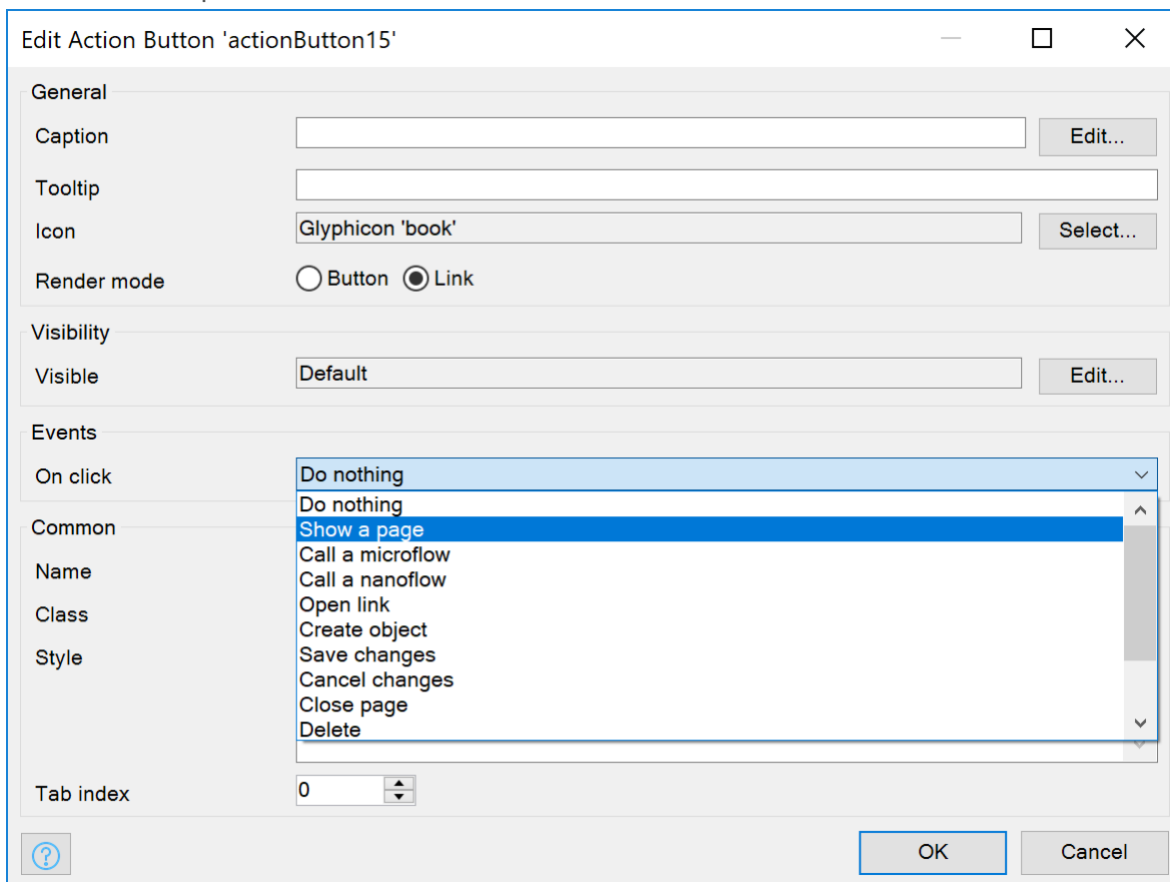
You have now created four overview pages for the courses, locations, teachers, and trainees. It's time to connect those separate pages to your homepage!

- 1 To connect a page to a button, go to your homepage and double-click the button that will open the Course_Overview page.

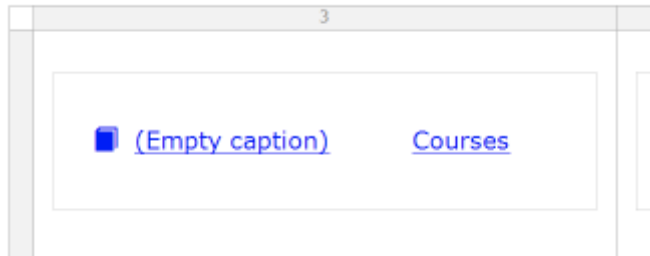
In the **Properties** screen, under **Events** you can see the different options for **On Click Action**. That means that when you click that button, a certain action will be triggered. The most important options, that you will work with during this course, are:

- **Do nothing:** This is the default option. It means that no action will be taken, and you will stay on the same page.
- **Show a page:** You will be redirected to a certain page.
- **Call a microflow:** A microflow will be triggered. You will learn more about microflows in module 5.

You will become more familiar with the other **On Click Action** options when you progress to the Advanced and Expert courses.



- 2 In this case, select **Show a page** as your **On Click Action**, as you want a new page to open once you click the button. In the **Select Page** screen, select the **Course_Overview** page.
- 3 Click **Select** and then **OK** to complete the configuration of the button.
- 4 Make both the icons and the captions underneath them go to the pages.



- 5 Link all the pages that you created to the overview page. Make sure that both the icons and the captions underneath them go to the pages.



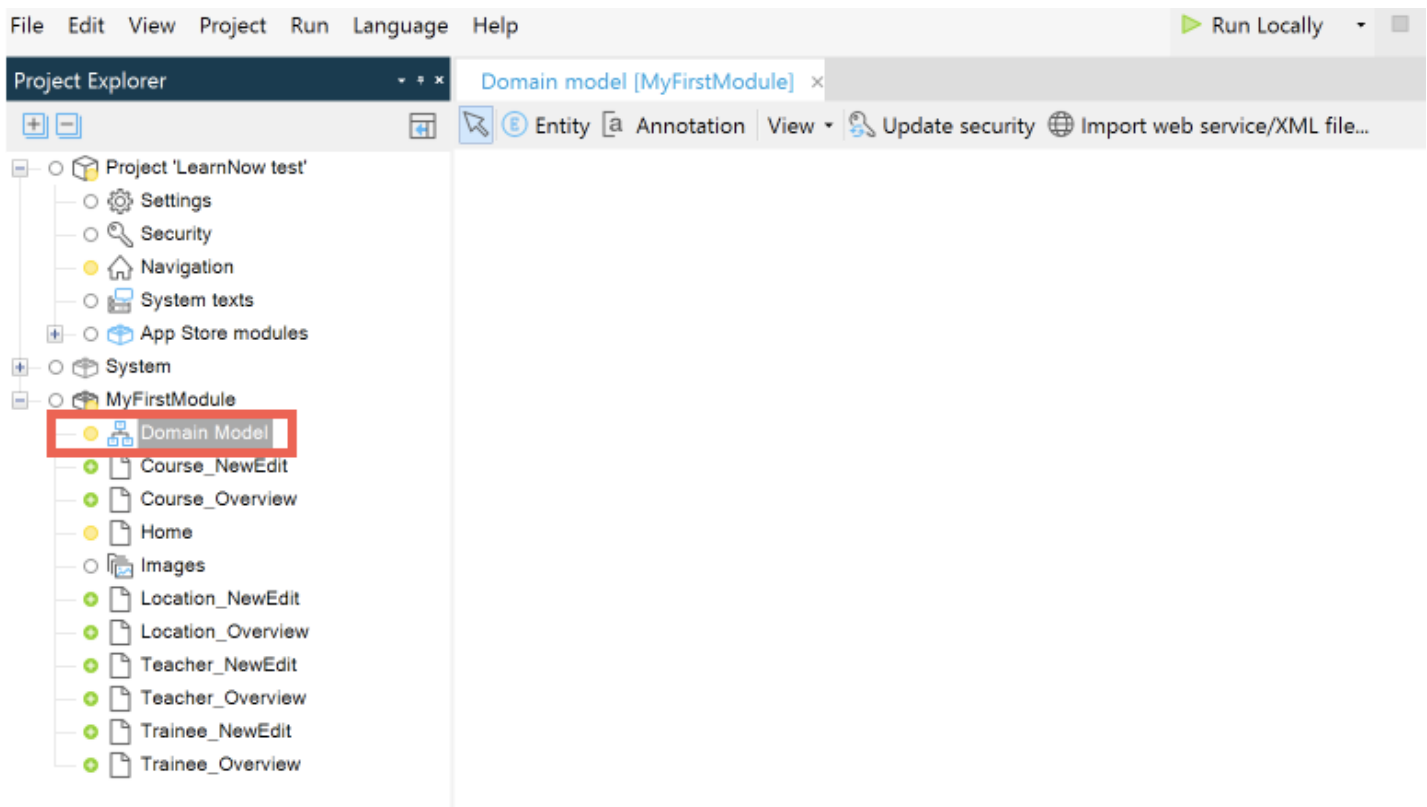
You can also create new pages directly from the button. To do that, click **New page** when selecting the page you want the button to open. This will allow you to create pages even faster. Try both methods and see which one you prefer!

Congratulations, you linked all your buttons to the correct overview pages. In the next lecture you will learn how to make your app dynamic, by creating the data structure.

3.7 Add Data to Your App

After creating pages and linking to them using buttons, the next step in the process is to fill the pages with actual data, as the new overview pages aren't showing any information yet. On each of these pages, Jimmy would like to see a list of the corresponding information. For example, under **Course_Overview**, he would like to see a list of all the existing courses. As those courses are often changing, he wants to be able to add them himself, instead of having them as static text in the app.

You are going to define your data structure (all the data that can be viewed and managed in the app) in the Domain Model. You can find the Domain Model in the Project Explorer at the top of the MyFirstModule module folder. Go ahead and open it!



Right now the Domain Model is still empty. If you want to have dynamic data in the app, the data structure needs to be present in the Domain Model in the shape of **Entities** and **Attributes**. Let's take a look at what those are in the next lecture.

3.8 Entities and Attributes

There are two concepts that are important when discussing adding data to an app:

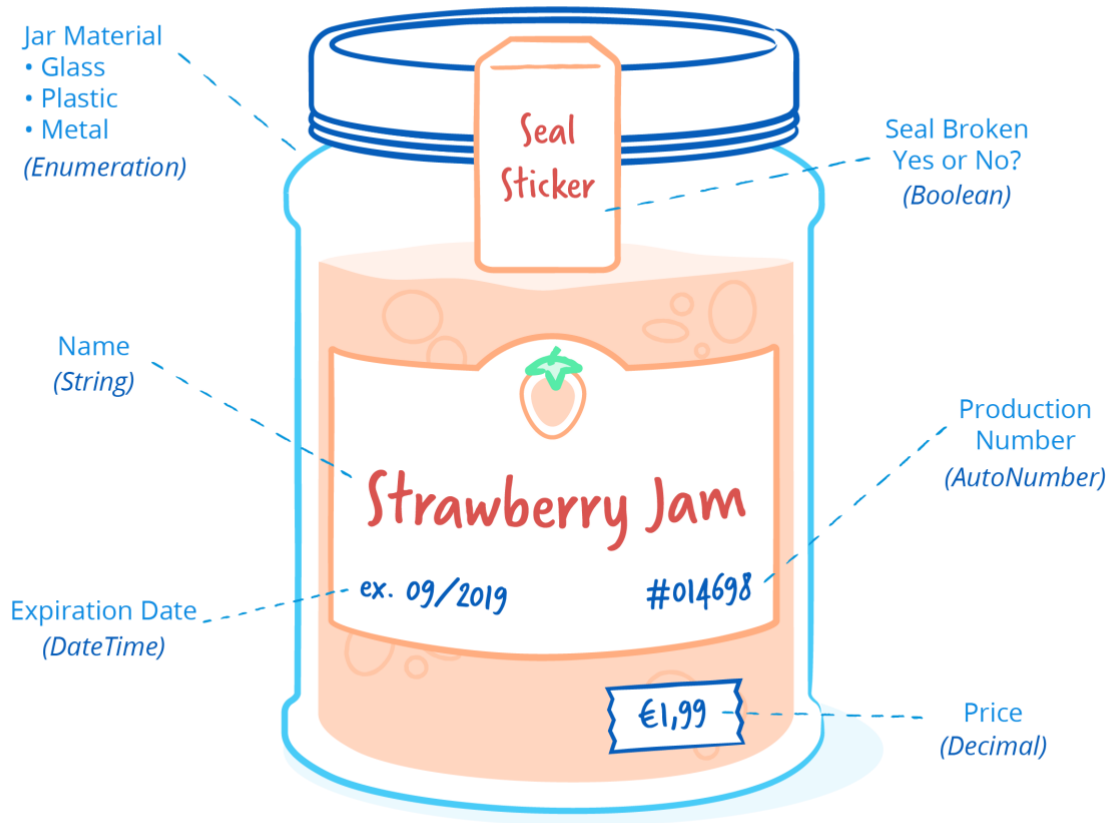
- **Entities** - To show data in your app, you need to connect the list view to an entity. An example of an entity for the LearnNow app would be **Course**. The entity will be filled with attributes.
- **Attributes** - Attributes determine what kind of information regarding the courses you can see and add (for example, a title and description). Each attribute has a data type. This property defines the type of data that can be stored in the attribute, just like you format the cells of an Excel sheet.

The following Attribute types are available, each with their possible values:

Type	Possible values
AutoNumber	Automatically generated positive or negative whole number. The default value of the attribute determines the first number that will be generated. Each object created of the same type will have an AutoNumber that is one greater than the previous one.
Binary	Binary data (for example, an entire file).
Boolean	True or false.
Decimal	A positive or negative number that can have digits after the decimal point.
DateTime	A point in time that consists of a date and time component, accurate up to the millisecond.
Enumeration	A list of predefined values.
HashString	The string will be stored after it is hashed (encrypted). This type can be used to store a password, for example. Hash values are generated using the hash algorithm that is chosen in the Project Settings.
Integer	A (small) whole number that can be positive.
Long	A (big) whole number that can be positive.
String	A text containing letters, spaces, numbers, and other characters.

Example

Take a look at the following image of a jar of strawberry jam. The jar would be the entity in this example, but can you think of a couple of attributes and their types for this jar?



Perhaps you came up with something like this:

- The name of the product in the jar (Strawberry Jam). This could be the Name attribute, type String, because it's text.
- The price of the jar (€ 1,99). This would be a Price attribute, type Decimal, because it's a number with digits after the decimal point.
- The expiration date (09/2019). This would be a DateAndTime attribute called Expiration Date, because it's a date value.
- Whether the seal is broken or not would be a Boolean attribute type, because it's a question you can answer with True (Yes) or False (No).
- The material of the jar could be either glass, plastic, or metal . This would be translated into the Material attribute, which would become an Enumeration, because it's a fixed list of options.
- The production number (#014698). That's a bit of a tricky one. It could be a regular number, which means it would be an Integer. But take into consideration that the production number would have to go up by one for every newly created jar. This can be easily achieved by making it an AutoNumber attribute.

3.8.1 Define attribute types

Jimmy has provided you with a list of what information he would like to be able to see and add to the system. Can you come up with an example for each attribute? What type would each attribute be?

Entity: Course		
Attributes	Example	Attribute type
Title	<i>Rapid Developer</i>	<i>String</i>
Description		
Duration (in days)		
Price		

Entity: Location		
Attributes	Example	Attribute type
Name		
Address		

Entity: Teacher		
Attributes	Example	Attribute type
Name		
Email address		

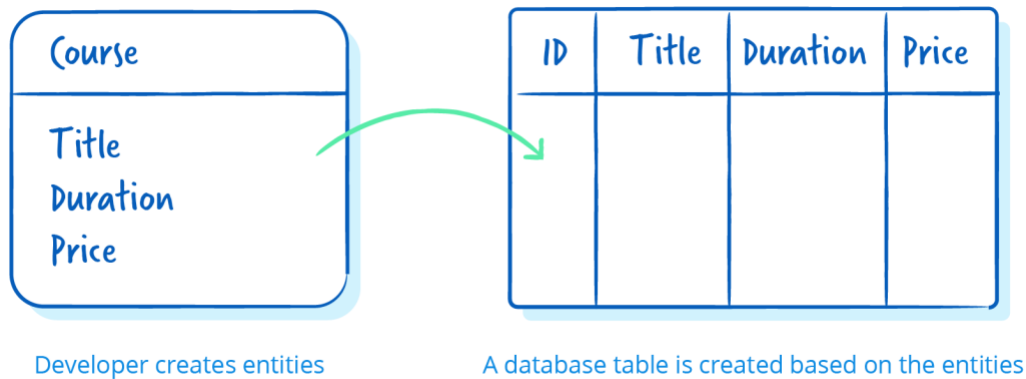
Entity: Trainee		
Attributes	Example	Attribute type
Name		
Address		
Email address		

3.9 The Domain Model

The domain model is a visual representation of your data structure. All the entities and attributes that you defined in the previous assignment will need to be added to the domain model. The entities will be represented by blue rectangles.

The structure of the domain model determines what the database will look like. Each entity will become a database table, and each attribute will become a column in the database. Every object you store in the database (for example, a course) becomes a row in the corresponding database table. This also means that if something is not in the domain model, you also can't add that information to the database.

This is what the database table for the **Course** entity looks like:



You see that ID column? That isn't an attribute. So how did it end up in the database table? This is something Mendix does automatically for you, because every object always needs a unique identifier (an ID) so that the system knows which object it is.

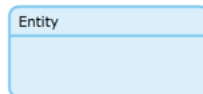
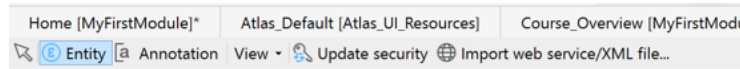
You can change the domain model as often as you like, and the database will synchronize when you publish the app again. Also, when you change the name of an entity or attribute, your app will still work, because the Modeler will automatically apply these changes everywhere.

In the next assignment you will create the Domain Model for the LearnNow app.

3.9.1 Create the Entities and Attributes

You will now create the Domain Model for the LearnNow app.

1. Open the **Domain Model** of the **MyFirstModule** module.
2. Create a new entity by clicking **Entity** in the domain model toolbar.
3. Click somewhere in the empty domain model to place the newly created entity.



Naming entities and attributes

When you create entities or attributes that have more than one word in their name (for example, email address), use CamelCase. This means you use no white-spaces or underscores (“_”) and start each new word with an upper-case letter. Mendix will understand this, and everywhere the name of the entity or attribute needs to be shown, Mendix will output it as separate words and make those upper-case letters lower-case again. For example, “Email address” written in CamelCase is “EmailAddress.”

4. Double-click on the entity to start filling out its details. Let’s start with the **Course** entity. Entity names should always be self-explanatory and singular. It is not a list of courses, but a visual representation of the information a course will contain. A good name for the entity therefor is simply **Course**.
5. Time to create the attributes! In the **Attributes** tab (this tab is shown by default), click **New**.
 - Name the attribute **Title**.
 - The default attribute type is **String**, so you don’t have to change that.
 - Click **OK**.
6. Click **New** again to create the next attribute. Once you have created all attributes, click **OK**.

Properties of Entity 'MyFirstModule.Course'

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'

Documentation

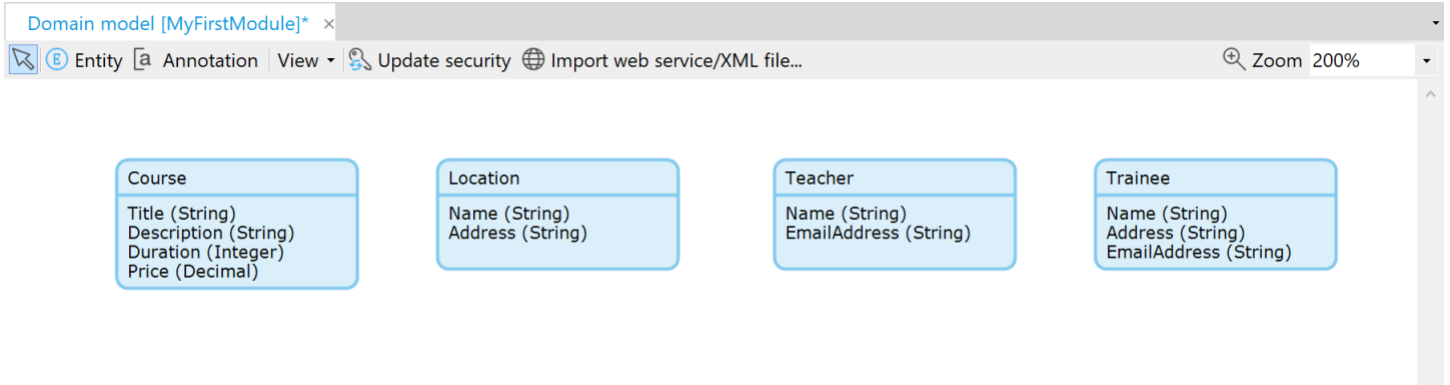
Attributes Associations Validation rules Event handlers Indexes

Name	Type	Default value	Microflow
Title	String (200)		
Description	String (200)		
Duration	Integer	0	
Price	Decimal	0	

?
OK
Cancel

- Keep doing this until you have created all necessary entities and attributes. Be careful to always select the correct attribute type! To refresh your memory, take a look at the entities and attributes you defined in 3.7.1.

Your domain model should now look like this:



Awesome! You have created the homepage and defined your data structure, you are doing great! Continue to the next module to learn all about creating beautiful overview and detail pages. Once you have built those, Jimmy will be able to enter his data in the app!

3.10 Get the Latest Version of the LearnNow App

In the next module, you are going to learn how to display data in your app. If everything is working well for you so far, you can mark this activity as **Completed** and continue to the next module.

Otherwise, under the **Resources**, you can find a small gift from us: the latest version of the LearnNow App! You can either start from scratch with a new project or keep your own project and update your model. Assignments for both options are presented below.

Option 1: I am now starting with the learning path

If you're starting from scratch, follow the instructions below to prepare the training project and content.

Import the Project Package

We've provided a project package that you can use to create a new project. It contains all the required development content up to this point.

1. Download the **Project package** from the **Resources** tab.
2. In the Desktop Modeler, select **File > Import Project Package**.
3. Select the downloaded package.
4. Select a **Location** on your drive to store the project files.
5. On the **Import Project Package** dialog box, select **Team Server** and upload to a **New repository**.
6. Enter *LearnNow Management* for the **App name**, and click **OK**.

Import the User Stories

1. Download the **User stories** from the **Resources** tab in the training environment.
2. In the Desktop Modeler, click **Project Dashboard** to go to your project's dashboard in the Developer Portal.
3. Select **Collaborate > Stories** in the sidebar menu.
4. Click **More**.
5. Select **Import/Export**.
6. Select **Update stories from Excel** and then click **Browse**.
7. Select the **User stories** Excel file and click **Import**.
8. Click **Next** to import the sprint and user stories:
9. Click **Develop > Planning** in the sidebar menu.
10. Mark the **Get started** sprint as completed.

Option 2: I already have a project and I want to Update My Module

If you want to make sure your model is up to date for the assignments in this module, you can update your model with a specific module we've provided. Follow the instructions below to update your existing model.

Import the Module Package

1. Download the **Module package** from the **Resources** tab of module 4 in the training environment.
2. Open your **LearnNow Management** project in the Desktop Modeler.
3. Go to the **Project Explorer**, right-click on the project name and select **Import module package**.
4. Select the downloaded module.
5. Select **Replace existing module** for the **Action**, and then select **LearnNow Management** for the **Module to replace** (or **MyFirstModule** if you haven't renamed it).
6. Click **Import**.

Set Up Your Navigation

1. Open the project **Navigation**.
2. Add the following navigation items:

Caption	Target Page
Courses	Course_Overview
Locations	Location_Overview
Teachers	Teacher_Overview
Trainees	Trainee_Overview

That's it! You're all set to do the assignments. Go ahead and continue with the next module.

3.11 Summary

In this module you learned:

- The basic elements of the Desktop Modeler
- What widgets are and how to use them to build pages
- How to use building blocks to quickly build your pages
- How to adjust the layout of a page for desktop, tablet, and mobile
- How to create buttons
- How to create new pages
- How to link pages to buttons
- What the domain model is
- What entities and attributes are
- How to create your domain model

In the next Module you are going to learn how to connect the entities in your Domain Model to your pages. This will allow Jimmy to have dynamic data on his pages.

3.12 Knowledge Check

1. In the Desktop Modeler, you can access the Domain Model of your app from:
 - a) The Project Explorer
 - b) The Project Dashboard
 - c) The Connector
 - d) The Domain Model tab

2. The value 1.23 could be stored in an attribute of the type:
 - a) AutoNumber
 - b) Integer
 - c) Decimal
 - d) Long

3. The Attribute type Enumeration is used to store:
 - a) Automatically generated positive or negative whole number
 - b) A list of predefined values
 - c) A whole number that can only be positive
 - d) A positive or negative number

4. A page is responsive when:
 - a) It works and looks good on all devices
 - b) It includes user interaction
 - c) It rotates as you rotate your mobile
 - d) It allows dynamic data to be displayed

5. When using the Layout Grid, the weight of all columns must add up to:
 - a) 4
 - b) 6
 - c) 12
 - d) 16

6. It is possible to add multiple widgets in a container
 - a) True
 - b) False

7. You can have different row layout for mobile, tablet, and phone
 - a) True
 - b) False

Answers:

1-a, 2-c, 3-b, 4-a, 5-c, 6-a, 7-a

Mendix is the fastest and easiest platform to build and continuously improve mobile and web applications at scale. Recognized as a market leader by leading analysts, we help our customers digitally transform their organizations and industries by building, managing, and improving apps at unprecedented speed and scale. More than 4,000 forward-thinking enterprises, including Kuehne + Nagel, KLM, Philips and North Carolina State University use our platform to build business applications to delight their customers and improve operational efficiency.

Learn more at [Mendix.com](https://mendix.com)

