

⚡ Attention Salesforce Certified Trailblazers! Maintain your credentials and link your [Trailhead](#) and Webassessor accounts by December 6th. [Learn more](#)

### 1. [Lightning Design System for Developers-](#)



### 2. [Lay Out a Record Home Page and Using Advanced Components-](#)

## Lay Out a Record Home Page and Using Advanced Components-

### Learning Objectives

After completing this unit, you'll be able to:

- Lay out a record home page using the Grid System.
- Employ two variants of the Card component to render record related lists.

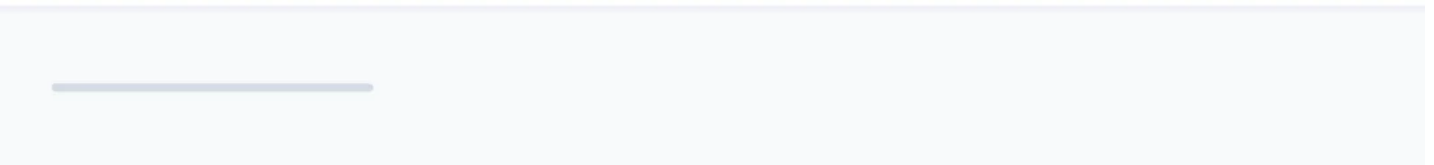
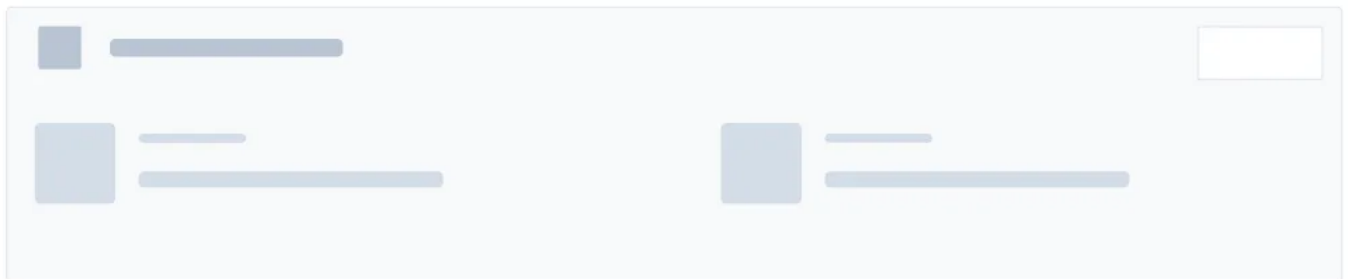
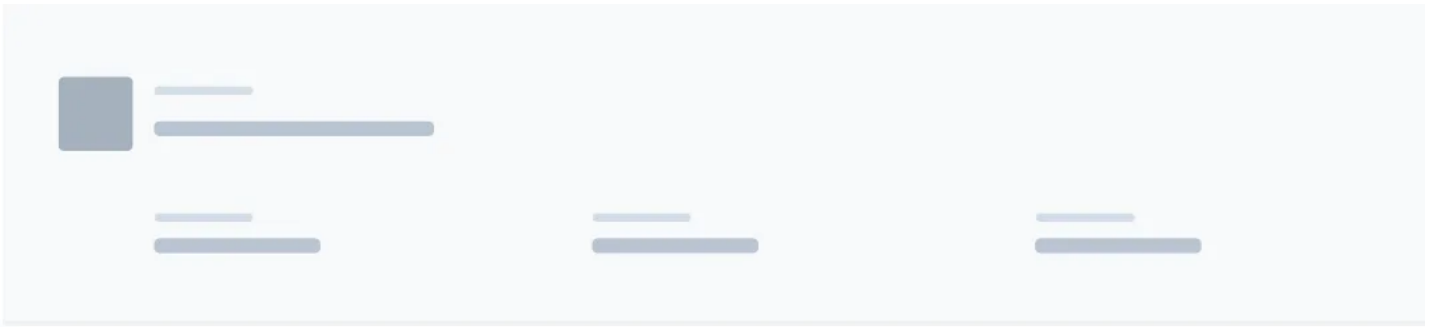
### The Grand Finale

Thanks for sticking with the module this far! In this final unit we're going to build another brand new page which will magically again conform to the Lightning UI without us writing a single line of CSS. We'll be using many of the components we've learned about in the previous modules including the **Grid System**, **Page Header**, and **Icons**. We're also going to introduce a couple of key new components, **Card** and **Tile**.

The page in question is going to be a Record Home page. Why this page? We figure it's prototypical of many of the app pages you'll be building, in many ways it's the classic Salesforce page. This example will be a template on which you base your apps. Also it's complicated but not too complicated so it'll make good practice and you'll earn your Trailhead badge in the process! Shall we begin?

### Record Page Layout

Here's a wireframe of the page we're going for. At the top a page header, with two related lists below, a large one on the left, and a compact card to the right. Below that we have the usual footer. The good news is that we'll be able to reuse a bunch of the markup we've written previously.



Lets start with an outer page skeleton in a brand new Visualforce page, named `Trailhead_SLDS_RecordHome`.

```
<apex:page showHeader="false" standardStyleSheets="false" sidebar="false" applyHtmlTag="false" applyBodyTag="false" docType="html-5.0">
<html xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" lang="en">
<head>
<meta charset="utf-8" />
<meta http-equiv="x-ua-compatible" content="ie=edge" />
<title>Salesforce Lightning Design System Trailhead Module</title>
<meta name="viewport" content="width=device-width, initial-scale=1" />
<!-- Import the Design System style sheet -->
<apex:slds />
</head>
<body>
<!-- REQUIRED SLDS WRAPPER -->
<div class="slds-scope">
<!-- MASTHEAD -->
<p class="slds-text-heading_label slds-m-bottom_small">Salesforce Lightning Design System Trailhead Module</p>
<!-- / MASTHEAD -->
<!-- PAGE HEADER -->
<!-- / PAGE HEADER -->
<!-- PRIMARY CONTENT WRAPPER -->
<div class="myapp">
</div>
<!-- / PRIMARY CONTENT WRAPPER -->
<!-- FOOTER -->
<footer role="contentinfo" class="slds-p-around_large">
<!-- LAYOUT GRID -->
<div class="slds-grid slds-grid_align-spread">
<p class="slds-col">Salesforce Lightning Design System Example</p>
<p class="slds-col">&copy; Your Name Here</p>
</div>
<!-- / LAYOUT GRID -->
</footer>
<!-- / FOOTER -->
</div>
<!-- / REQUIRED SLDS WRAPPER -->
</body>
</html>
```

```
<!-- JAVASCRIPT -->
<!-- / JAVASCRIPT -->
</body>
</html>
</apex:page>
```

Copy

Copy

## Adding a Page Header

Next we'll add a Page Header component, this one a bit more involved than the one we used in the list view. We'll have two rows. The first one will be much the same as in the list view. The second row will show some key fields from the record.

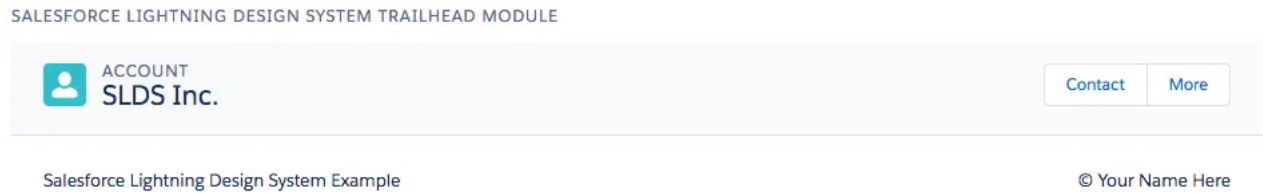
The code for the first row should be looking familiar by now. Place it in between the `<!-- PAGE HEADER -->` comments:

```
<!-- PAGE HEADER -->
<div class="slds-page-header">
  <!-- PAGE HEADER TOP ROW -->
  <div class="slds-grid">
    <!-- PAGE HEADER / ROW 1 / COLUMN 1 -->
    <div class="slds-col slds-has-flexi-truncate">
      <!-- HEADING AREA -->
      <!-- MEDIA OBJECT = FIGURE + BODY -->
      <div class="slds-media slds-no-space slds-grow">
        <div class="slds-media figure">
          <svg aria-hidden="true" class="slds-icon slds-icon-standard-user">
            <use xlink:href="{!URLFOR($Asset.SLDS, 'assets/icons/standard-sprite/svg/symbols.svg#user')}"/></use>
          </svg>
        </div>
        <div class="slds-media body">
          <p class="slds-text-title caps slds-line-height reset">Account</p>
          <h1 class="slds-page-header__title slds-m-right_small slds-align-middle slds-truncate" title="SLDS Inc.">SLDS Inc.</h1>
        </div>
      </div>
    </div>
    <!-- / PAGE HEADER / ROW 1 / COLUMN 1 -->
    <!-- PAGE HEADER / ROW 1 / COLUMN 2 -->
    <div class="slds-col slds-no-flex slds-grid slds-align-top">
      <div class="slds-button-group" role="group">
        <button class="slds-button slds-button_neutral">
          Contact
        </button>
        <button class="slds-button slds-button_neutral">
          More
        </button>
      </div>
    </div>
  </div>
  <!-- / PAGE HEADER TOP ROW -->
  <!-- PAGE HEADER DETAIL ROW -->
  <!-- / PAGE HEADER DETAIL ROW -->
</div>
<!-- / PAGE HEADER -->
```

Copy

Copy

Preview your page and you will see the top row of the page header as expected. If any of the code is unclear, please refer to the previous units or the respective component documentation.



Next let's implement the second detail row using another **Grid System** component. It will contain four columns. Include the following code inside the `<!-- PAGE HEADER DETAIL ROW -->` placeholder comments:

```
<!-- PAGE HEADER DETAIL ROW -->
<ul class="slds-grid slds-page-header_detail-row">
  <!-- PAGE HEADER / ROW 2 / COLUMN 1 -->
  <li class="slds-page-header_detail-block">
    <p class="slds-text-title slds-truncate slds-m-bottom xx-small" title="Field 1">Field 1</p>
    <p class="slds-text-body_regular slds-truncate" title="Description that demonstrates truncation with a long text field">Description that demonstrates truncation
  </li>
  <!-- PAGE HEADER / ROW 2 / COLUMN 2 -->
  <li class="slds-page-header_detail-block">
    <p class="slds-text-title slds-truncate slds-m-bottom xx-small" title="Field2 (3)">Field 2 (3)
    <button class="slds-button slds-button icon" aria-haspopup="true" title="More Actions">
      <svg class="slds-button icon slds-button icon small" aria-hidden="true">
        <use xlink:href="/assets/icons/utility-sprite/svg/symbols.svg#down"/></use>
      </svg>
    <span class="slds-assistive-text">More Actions</span>
    </button>
  </li>
  <p class="slds-text-body--regular">Multiple Values</p>
```

```

</li>
<!-- PAGE HEADER / ROW 2 / COLUMN 3 -->
<li class="slds-page-header detail-block">
<p class="slds-text-title slds-truncate slds-m-bottom_xx-small" title="Field 3">Field 3</p><a href="javascript:void(0);">Hyperlink</a></li>
<!-- PAGE HEADER / ROW 2 / COLUMN 4 -->
<li class="slds-page-header detail-block">
<p class="slds-text-title slds-truncate slds-m-bottom_xx-small" title="Field 4">Field 4</p>
<p>
<span title="Description (2-line truncation-must use JS to truncate).">Description (2-line truncat...</span>
</p>
</li>
</ul>
<!-- / PAGE HEADER DETAIL ROW -->


```

Copy

Copy

Preview the page and you'll see the complete header with a second row for populating with key fields from the record:

#### SALESFORCE LIGHTNING DESIGN SYSTEM TRAILHEAD MODULE

<div>  <div>ACCOUNT SLDS Inc.</div> </div> <div>ContactMore</div>			
Field 1 Description that demonstrate...	Field 2 (3) Multiple Values	Field 3 Hyperlink	Field 4 Description (2-line truncat...

Salesforce Lightning Design System Example © Your Name Here

Yet again, the Design System has made our life really easy. This code lays out a complex component conforming to the Lightning UI spec without typing a character of CSS. Lets step through the code for the detail row. Again, we recommend having the code open in your favorite editor besides this window so you can cross-reference.

The outer `<ul>` instantiates our grid and adds the class `slds-page-header_detail-row` for correct alignment.

The four fields are laid out using a series of `<li>` with a `slds-page-header_detail-block` class.

Inside each detail block, we have one or multiple `<p>` element(s) carrying three classes: `slds-text-title`, `slds-m-bottom_xx-small` to add space below them, and `slds-truncate` to truncate the field value.

You can see a complete record home page header example on the [Page Header](#) component page.

## Adding the Primary Related List

Now let's move onto the record home detail area where we want to have two related lists side by side.

Include the following code inside the `<!-- PRIMARY CONTENT WRAPPER -->` comments:

```

<!-- PRIMARY CONTENT WRAPPER -->
<div class="myapp">
<!-- RELATED LIST CARDS-->
<div class="slds-grid slds-m-top_large">
<!-- MAIN CARD -->
<div class="slds-col slds-col_rule-right slds-p-right_large slds-size_8-of-12">
left hand column related list
</div>
<!-- / MAIN CARD -->
<!-- NARROW CARD -->
<div class="slds-col slds-p-left_large slds-size_4-of-12">
right hand column related list
</div>
<!-- / NARROW CARD -->
</div>
<!-- / RELATED LIST CARDS -->
</div>
<!-- / PRIMARY CONTENT WRAPPER -->

```

Copy

Copy

We use another **Grid System** component to lay our two columns sized manually using the [size helper classes](#). The first column will contain a large related list card. The narrower right-hand column will contain a compact card. Each of these is a variant of the [Card](#) component.

Here is the code for the left hand Card. Place it inside the `<!-- MAIN CARD -->` comments:

```

<!-- MAIN CARD -->
<div class="slds-col slds-col_rule-right slds-p-right_large slds-size_8-of-12">
<article class="slds-card">
<div class="slds-card header slds-grid">
<header class="slds-media slds-media--center slds-has-flexi-truncate">
<div class="slds-media figure">
<svg aria-hidden="true" class="slds-icon slds-icon-standard-contact slds-icon_small">
<use xlink:href="{!URLFOR($Asset.SLDS, 'assets/icons/standard-sprite/svg/symbols.svg#contact')}"/></use>
</svg>
</div>
<div class="slds-media body slds-truncate">
<a href="javascript:void(0);" class="slds-text-link--reset">
<span class="slds-text-heading_small">Contacts</span>
</a>

```


```
</div>
</header>
</div>
<!-- CARD BODY = TABLE -->
<div class="slds-card_body">
<table class="slds-table slds-table_bordered slds-no-row-hover slds-table_cell-buffer">
<thead>
<tr class="slds-text-heading--label">
<th class="slds-size 1-of-4" scope="col">Name</th>
<th class="slds-size 1-of-4" scope="col">Company</th>
<th class="slds-size 1-of-4" scope="col">Title</th>
<th class="slds-size 1-of-4" scope="col">Email</th>
<th scope="col"></th>
</tr>
</thead>
<tbody>
<tr class="slds-hint-parent">
<th class="slds-size 1-of-4" scope="row"><a href="javascript:void(0);">Adam Choi</a></th>
<td class="slds-size 1-of-4">Company One</td>
<td class="slds-size 1-of-4">Director of Operations</td>
<td class="slds-size 1-of-4">adam@company.com</td>
<td class="slds-cell-shrink">
<button class="slds-button slds-button icon-border-filled slds-button icon-x-small">
<svg aria-hidden="true" class="slds-button icon slds-button icon hint slds-button icon small">
<use xlink:href="{!URLFOR($Asset.SLDS, 'assets/icons/utility-sprite/svg/symbols.svg#down')}"></use>
</svg>
<span class="slds-assistive-text">Show More</span>
</button>
</td>
</tr>
</tbody>
</table>
</div>
<!-- / CARD BODY = SECTION + TABLE -->
<div class="slds-card_footer">
<a href="javascript:void(0);">View All <span class="slds-assistive-text">contacts</span></a>
</div>
</article>
</div>
<!-- / MAIN CARD -->
```

Copy

Copy

Preview the page to see the first related list in place:

SALESFORCE LIGHTNING DESIGN SYSTEM TRAILHEAD MODULE

 ACCOUNT  
SLDS Inc.

[Contact](#) [More](#)

Field 1	Field 2 (3)	Field 3	Field 4
Description that demonstrate...	Multiple Values	<a href="#">Hyperlink</a>	Description (2-line truncat...

Contacts

NAME	COMPANY	TITLE	EMAIL
Adam Choi	Company One	Director of Operations	adam@company.com

View All

right hand column related list

Salesforce Lightning Design System Example © Your Name Here

Now this demo just has one example contact so don't show this to your sales people! In real life you would have loads more people to sell stuff to!

Let's step through the markup.

The [Card](#) component is wrapped in a `<article class="slds-card">` element.

The card header is specified by a `<div>` element with the `slds-card_header` and `slds-grid` classes. The two columns contain a **Media Object** and an action button respectively. The second column could easily be extended using a **Button Group** component as in the list view. As always you can find more documentation and examples of all these classes on the [Design System](#) website.

The Card body is a `<div>` element with the class `slds-card_body`. Inside we have a Data Table component.

The Card is completed with a "View all" link inside `<div class="slds-card_footer">`.

### Adding the Final card

The final step of our record home is adding the right-hand, narrower card. For this we simply use the [Card](#) component again. Because of the grid system, this card will automatically fill the narrower space. Here is the markup to go within the `<!-- NARROW CARD -->` comments:

```
<!-- NARROW CARD -->
<div class="slds-col slds-p-left_large slds-size_4-of-12">
<article class="slds-card">
<div class="slds-card_header slds-grid">
<header class="slds-media slds-media_center slds-has-flexi-truncate">
```

```

<div class="slds-media figure">
<svg class="slds-icon slds-icon-standard-lead slds-icon small" aria-hidden="true">
<use xlink:href="{!URLFOR($Asset.SLDS, 'assets/icons/standard-sprite/svg/symbols.svg#lead')}"/></use>
</svg>
</div>
<div class="slds-media body slds-truncate">
<h2 class="slds-text-heading_small">Team</h2>
</div>
</header>
</div>
<div class="slds-card body">
<div class="slds-card body_inner">
<div class="slds-tile">
<h3 class="slds-truncate" title="Anne Choi"><a href="javascript:void(0);">Anne Choi</a></h3>
<div class="slds-tile detail slds-text-body_small">
<dl class="slds-list--horizontal slds-wrap">
<dt class="slds-item_label slds-text-color_weak slds-truncate" title="Email:">
Email:
</dt>
<dd class="slds-item detail slds-truncate" title="achoi@burlington.com">
achoi@burlington.com
</dd>
</dl>
</div>
</div>
</div>
</div>
<div class="slds-card footer">
<a href="javascript:void(0);">View All <span class="slds-assistive-text">team members</span></a>
</div>
</article>
</div>
<!-- / NARROW CARD -->

```

Copy


Copy

As before, we provide a card header and a card body. The card body is a series of [Tile](#) components. A Tile is a grouping of related information associated with a record. There are several variants of this component, each with different groupings of information. Pay close attention to the markup in the examples provided in the documentation, as each tile layout is constructed differently. Here we use the simplest base variant.

Each tile has the `slds-tile` class. Inside that we provide a title for the Tile and some contents. As before, we wrap up the card with the same `<div class="slds-card__footer">` markup we used in the other Card.

You should now be able to preview the complete record home page:


## SALESFORCE LIGHTNING DESIGN SYSTEM TRAILHEAD MODULE



ACCOUNT  
SLDS Inc.

Contact
More


Field 1	Field 2 (3)	Field 3	Field 4
Description that demonstrate...	Multiple Values	Hyperlink	Description (2-line truncat...



Contacts

NAME	COMPANY	TITLE	EMAIL
Adam Choi	Company One	Director of Operations	adam@company.com

View All



Team

Anne Choi  
Email: achoi@burlington.com

View All

Salesforce Lightning Design System Example
© Your Name Here

This complete example of the markup for a fairly complex record home brings this Trailhead module to a close. Hopefully you now have a solid foundation in the Design System and are eager to use it in your own applications.

Obviously these sample pages are just the beginning. When building your own apps, your next step would be to hook in real data, link pages together, and build out your business logic. We hope the Design System will free you up from styling UI with CSS so you can focus on building out this real functionality. Have fun and please do [send us feedback on how you get on](#).

## Resources

- [Detailed Card documentation](#)
- [Detailed Tile documentation](#)

1 Which of these is NOT a Design System component?

- ☐ A. Activity Timeline.  
☐ B. Thumbnail.  
☐ C. Datepicker.  
☐ D. Pill.

2 Which of the following is true about the Tile component?

- ☐ A. It is a grouping of related information associated with a record.  
☐ B. It may contain a wide range of content such as images, icons, action overflows, and description lists

- ☐ C.It is based upon the Media Object component.
- ☐ D.All of the above.