

Study Guide: User Interface (14%)

Overview

The User Interface (UI) section of the Salesforce Platform App Builder Certification Exam, weighted at 14%, tests your ability to create and customize the front-end experience for Salesforce users. This is all about making the platform accessible, productive, and visually appealing using declarative tools—no coding required. The focus is on the Lightning Experience, Salesforce’s modern UI framework, which replaced the older Classic interface. Mastery here means understanding how to design pages, enhance usability with actions and buttons, and optimize for both desktop and mobile users.

Key tools include the Lightning App Builder, page layouts, custom buttons, links, actions, and components. This guide will dive deep into each area, offering practical insights and study aids to prepare you for the exam and real-world app-building scenarios.

Key Concepts

Here’s what you’ll need to master:

1. **Lightning App Builder:** Drag-and-drop tool for customizing pages.
 2. **Page Layouts:** Define field placement and related lists for objects.
 3. **Custom Buttons and Links:** Add functionality to records.
 4. **Quick Actions:** Streamline tasks with prebuilt or custom actions.
 5. **Lightning Components:** Reusable UI building blocks (standard, custom, or AppExchange).
 6. **Record Pages:** Tailor layouts for specific record types or apps.
 7. **Mobile Optimization:** Ensure usability in the Salesforce mobile app.
 8. **Dynamic Features:** Use visibility filters and dynamic forms.
-

Detailed Explanation

1. Lightning App Builder

The Lightning App Builder is the primary tool for designing custom pages in Salesforce Lightning. It's a drag-and-drop interface that lets you build Home pages, Record pages, and App pages without coding.

- **Page Types:**
 - **Home Page:** Dashboard-style layout for app entry points.
 - **Record Page:** Displays a single record (e.g., an Account).
 - **App Page:** Custom page for specific workflows.
- **Components:** Add prebuilt elements like Record Detail, Related Lists, Tabs, or Charts.
- **Example:** Build a custom Account Record page:
 - Drag “Tabs” component → Add “Details,” “Related,” and “Activity” tabs.
 - Add “Record Detail” component to the Details tab.
- **Real-World Scenario:** A sales team needs a Home page with key metrics:
 - Components: “Report Chart” (Open Opportunities), “Recent Items,” “Chatter Feed.”
- **Best Practices:**
 - Keep layouts clean—avoid cluttering with too many components.
 - Use templates (e.g., Header and Two Columns) for structure.
 - Test on desktop and mobile views.

2. Page Layouts

Page Layouts control the arrangement of fields, buttons, and related lists on an object's record detail page. They're configured in Object Manager.

- **Features:**
 - Add/remove fields.
 - Organize into sections (e.g., “Contact Info,” “Billing Details”).
 - Include related lists (e.g., Opportunities under Account).
- **Example:** Customize Contact layout:

- Section: “Personal Info” → Fields: FirstName, LastName, Email.
- Related List: “Open Activities.”
- **Real-World Scenario:** A support team needs a Case layout:
 - Section: “Case Details” → Case Number, Priority, Status.
 - Buttons: “Close Case,” “Escalate.”
- **Tips:**
 - Assign layouts by Profile or Record Type for tailored views.
 - Minimize scrolling—prioritize key fields at the top.
 - Use blank spaces for visual separation.

3. Custom Buttons and Links

Custom Buttons and Links add functionality to records, launching URLs, Visualforce pages, or JavaScript (though JS is less common in Lightning).

- **Types:**
 - **Detail Page Button:** Appears on record detail (e.g., “Generate Quote”).
 - **List Button:** Acts on multiple records (e.g., “Mass Update”).
 - **Custom Link:** Opens external URLs or resources.
- **Example:** Button on Opportunity to open a PDF quote:
 - Type: URL.
 - URL: /apex/QuotePDF?id={!Opportunity.Id}.
- **Real-World Scenario:** Link to a company’s support portal:
 - Link on Case: <https://support.company.com/case/{!Case.CaseNumber}>.
- **Best Practices:**
 - Use descriptive labels (e.g., “Create Invoice” not “Button1”).
 - Test URLs with sample records.
 - Limit use in Lightning—prefer Quick Actions.

4. Quick Actions

Quick Actions streamline tasks by letting users create records, update fields, or trigger custom logic from a button.

- **Types:**
 - **Object-Specific:** Tied to one object (e.g., “Log a Call” on Contact).
 - **Global:** Available across objects (e.g., “New Task”).
 - **Custom:** Built with Flow or Lightning Components.
- **Example:** “New Contact” action on Account:
 - Pre-fill: AccountId = {!Account.Id}.
 - Layout: Fields like FirstName, LastName, Phone.
- **Real-World Scenario:** “Escalate Case” action:
 - Action Type: Update Record.
 - Field Update: Priority = "High", Escalated__c = TRUE.
- **Tips:**
 - Add to page layouts or Lightning pages via the “Highlights Panel.”
 - Use sparingly—too many actions confuse users.
 - Test pre-fill values for accuracy.

5. Lightning Components

Lightning Components are reusable UI elements added to pages via Lightning App Builder. They can be standard (built by Salesforce), custom (coded), or from AppExchange.

- **Standard Components:**
 - Record Detail, Related List, Chatter, Report Chart.
- **Custom Components:**
 - Built with Aura or Lightning Web Components (LWC)—not required for App Builder but good to know.
- **AppExchange Components:**
 - Third-party add-ons (e.g., a calendar widget).
- **Example:** Add a “Recent Opportunities” component to an Account page:

- Component: Related List → Object: Opportunity.
- **Real-World Scenario:** A custom “Weather” component:
 - Displays live weather based on Account’s city (AppExchange or custom).
- **Best Practices:**
 - Use standard components for simplicity.
 - Evaluate AppExchange options before building custom.
 - Ensure mobile compatibility.

6. Record Pages

Record Pages in Lightning App Builder let you customize how records display, with dynamic features like visibility filters.

- **Customization:**
 - Tabs for different views (e.g., “Details,” “Notes”).
 - Dynamic visibility: Show components based on field values.
- **Example:** Opportunity Record page:
 - Tab: “Details” → Record Detail.
 - Tab: “Pipeline” → Report Chart (visible if StageName != "Closed").
- **Real-World Scenario:** A Case page for support reps:
 - Tab: “Case Info” → Fields.
 - Tab: “Escalation” → Visible if Priority = "High".
- **Tips:**
 - Use Dynamic Forms to make fields conditional.
 - Test with different record types.

7. Mobile Optimization

The Salesforce mobile app requires UI adjustments for smaller screens and touch navigation.

- **Features:**
 - Compact layouts: Prioritize key fields (e.g., Name, Phone on Contact).

- Navigation: Customize the mobile nav bar.
- Actions: Ensure Quick Actions are mobile-friendly.
- **Example:** Compact layout for Lead:
 - Fields: Name, Company, Status.
- **Real-World Scenario:** Mobile Home page for field reps:
 - Components: “Today’s Tasks,” “Nearby Accounts” (geolocation).
- **Best Practices:**
 - Test in the Salesforce mobile app simulator.
 - Avoid wide layouts—stack components vertically.
 - Limit scrolling on key pages.

8. Dynamic Features

Dynamic features like visibility filters and Dynamic Forms enhance UI flexibility.

- **Visibility Filters:**
 - Show/hide components based on conditions (e.g., Amount > 10000).
- **Dynamic Forms:**
 - Move fields from page layouts to Lightning pages.
 - Add field-level visibility (e.g., show Discount__c if Type = "Special").
- **Example:** Dynamic Account page:
 - Component: “Billing Info” → Visible if Industry = "Finance".
- **Real-World Scenario:** Sales dashboard:
 - Chart: “Closed Deals” → Visible if User Role = “Sales Manager.”
- **Tips:**
 - Use simple conditions to avoid performance lags.
 - Combine with Record Types for maximum flexibility.

Table 1: Lightning App Builder Components

Component	Purpose	Example Use Case
Record Detail	Show record fields	Display Account info
Related List	List related records	Show Opportunities
Tabs	Organize content	Separate Details and Notes
Report Chart	Visualize data	Pipeline by Stage

Table 2: Quick Actions vs. Custom Buttons

Feature	Quick Action	Custom Button
Setup	Declarative	URL or code
Scope	Object/Global	Detail/List
Example	“Log a Call”	“Open PDF”
Mobile Support	Strong	Limited in Lightning

Table 3: Mobile Optimization Tips

Element	Recommendation	Example
Compact Layout	3-5 key fields	Name, Phone, Email
Navigation	Prioritize frequent tasks	“Tasks,” “Accounts”
Components	Stack vertically	List above Chart

Practical Examples

1. **Lightning App Builder:** Sales Rep Home page:
 - Components: “Open Tasks,” “Top Opportunities” (Chart), “Chatter.”
2. **Page Layout:** Case for support:
 - Section: “Details” → Case Number, Status.
 - Related List: “Case Comments.”

3. **Quick Action:** “New Lead” on Account:
 - Pre-fill: AccountId.
 - Fields: FirstName, LastName.
 4. **Dynamic Form:** Opportunity page:
 - Field: Discount__c → Visible if StageName = "Negotiation".
-

Tips for Success

- **Hands-On Practice:** In a Developer org:
 - Build a custom Record page for Contact with 3 tabs.
 - Add a Quick Action to Opportunity.
 - Create a mobile-optimized Home page.
 - **Focus on Lightning:** Classic UI is outdated—stick to Lightning tools.
 - **Trailhead Modules:**
 - “Lightning App Builder”
 - “Salesforce Mobile App Customization”
 - **Scenarios:** Practice questions like:
 - “Design a page for X user role.”
 - “How would you add Y functionality?”
 - **Test Mobile:** Use the mobile simulator in Setup.
 - **Dynamic Features:** Experiment with visibility filters—key for modern UIs.
 - **User-Centric Design:** Think about usability—less is often more.
-

Bullet Point Summary

- **Lightning App Builder:**
 - Drag-and-drop customization.
 - Build Home, Record, App pages.

- Use templates wisely.
 - **Page Layouts:**
 - Arrange fields and lists.
 - Assign by Profile/Record Type.
 - Keep intuitive.
 - **Custom Buttons/Links:**
 - Add specific actions.
 - Prefer actions in Lightning.
 - Test thoroughly.
 - **Quick Actions:**
 - Streamline tasks.
 - Pre-fill for efficiency.
 - Optimize for mobile.
 - **Lightning Components:**
 - Standard for speed.
 - AppExchange for extras.
 - Ensure compatibility.
 - **Record Pages:**
 - Tailor to use cases.
 - Use dynamic visibility.
 - Test with data.
 - **Mobile Optimization:**
 - Prioritize key info.
 - Simplify navigation.
 - Test on app.
-

Study Plan

1. **Day 1:** Study Lightning App Builder (build 2 pages).
 2. **Day 2:** Master Page Layouts and Buttons (customize Account).
 3. **Day 3:** Practice Quick Actions (create 3 actions).
 4. **Day 4:** Explore Components and Record Pages (add dynamic features).
 5. **Day 5:** Focus on Mobile UI (optimize a page).
 6. **Day 6:** Review all concepts, test a full scenario.
 7. **Day 7:** Take a practice quiz on UI.
-

Exam Scenarios to Practice

1. **Scenario:** “A sales team needs a custom Opportunity page with pipeline data.”
 - Solution: Record Page → Tabs: “Details,” “Pipeline” (Chart).
2. **Scenario:** “Add a button to create a Task from a Lead.”
 - Solution: Quick Action → “New Task” → Pre-fill Related To.
3. **Scenario:** “Show a field only for managers on Case.”
 - Solution: Dynamic Form → Escalation_Notes__c visible if Role = “Manager.”