

1. Layout:

- Start with a clear information architecture: Organize your data and objects logically, using Salesforce's built-in objects like Accounts, Contacts, and custom objects for properties, leases, and tenants. Plan how data will flow through your application.
- Use custom objects and fields judiciously: Avoid cluttering your database with unnecessary fields or objects. Keep your data model simple and easy to understand.
- Create user-friendly page layouts: Design user interfaces with user experience (UX) in mind. Use Salesforce's Lightning App Builder to create clean, intuitive layouts for data entry and viewing.
- Implement record types: Use record types to display different page layouts for various property types, such as residential and commercial, to provide context-specific data entry forms.

2. Readability:

- Maintain consistent naming conventions: Use clear and consistent naming for objects, fields, and labels to make it easier for developers and administrators to understand and manage the application.
- Leverage field descriptions and help text: Provide descriptive help text for fields and objects to guide users and reduce the need for external documentation.
- Use validation rules and field-level security: Implement validation rules and field-level security to ensure data quality and protect sensitive information. Document these rules for clarity.

3. Reusability:

- Develop reusable components: Utilize Salesforce Lightning components and Apex code to encapsulate and reuse common functionality like property search, tenant management, or maintenance requests. This reduces code duplication and accelerates development.
- Create managed packages: Consider packaging your custom components, objects, and functionality into managed packages that can be distributed to other Salesforce users. This promotes reusability across different Salesforce instances.

- Document and comment your code: Ensure that your code is well-documented with clear comments explaining its purpose and functionality. This helps other developers understand and extend your work.

Additional best practices:

- Conduct regular code and data reviews with your development team to maintain code quality and data consistency.
- Implement robust testing, including unit tests, integration tests, and user acceptance testing to ensure the application works as expected and is bug-free.
- Keep an eye on Salesforce platform updates and best practices, as they may influence your application's architecture and design choices.

By focusing on these aspects during the project development phase, you'll enhance the quality and maintainability of your property management application in Salesforce. It will also make it easier for your team to understand, maintain, and extend the application in the future.