

Appian Composer Requirements Template

How to use this template

This template helps you define the scope of your application before you build it in Appian Composer. Work through each section and replace the sample content with details about your own project. Every section is required unless its heading is marked [Optional].

This template is optional. Composer accepts requirements in whatever form you already have them, so you don't need to force your project into this structure. Use this template only if you're not sure how to start, or if you don't have an existing requirements document to work from.

1. Overview

1.1 Executive Summary

What to think about: If you had to explain this project to your executive sponsor in 30 seconds, what would you say? What is the single biggest "win" this application provides?

Example: *The Case Management System will modernize ABC Insurance's claims processing workflow by providing a centralized platform for managing customer claims across all insurance products. This application will replace three legacy systems, reduce claim resolution time by an estimated 40%, improve customer satisfaction, and ensure regulatory compliance while providing management with enhanced visibility into workloads and performance metrics.*

1.2 Current State Description

What to think about: What is "broken" today? Where is your data currently "trapped" (e.g., in spreadsheets, emails, or old databases)? What specific manual tasks are wasting the most time?

Example: *Currently, ABC Insurance manages claims through three disconnected legacy systems based on product lines, supplemented by Excel spreadsheets and email. Pain points include duplicate data entry, lack of visibility across departments, inconsistent customer communications, manual handoffs between teams, delayed approvals due to paper-based processes, and the inability to generate comprehensive reports. Claims adjusters spend approximately 35% of their time on administrative tasks rather than customer service.*

1.3 Future State Vision

What to think about: If this application worked perfectly, what is the first thing a user would notice? How does the "day in the life" change for your team?

Example: The new Appian-based Case Management System will provide a unified platform where all claim types are processed through standardized workflows with automated routing and notifications. Adjusters will work from personalized dashboards showing their assigned claims, priorities, and SLAs. Management will access real-time analytics on team performance, bottlenecks, and compliance metrics. Key benefits include 360-degree claim visibility, automated document generation, integration with payment systems, improved compliance controls, and mobile access for field adjusters.

2. Personas

What to think about: Who are the distinct types of people using this app? Don't just list job titles; what are they trying to achieve? Who needs to see the data, and who is responsible for entering it?

Persona Name	Business Role / Description	Primary Goal in App
Example: Claims Adjuster	Investigates claims and determines coverage.	Efficiently access policy details to settle claims.
[Persona A]	[Role Description]	[Core Goal]
[Persona B]	[Role Description]	[Core Goal]

3. Functional Requirements

What to think about: This is the core of your application. Whether you prefer to write a narrative story or fill out a structured list, we need to understand the journey of your data.

- **The Milestones:** What are the major stages of this process?
- **The Persona:** Who is performing the action, and what is their "Success" at this stage?
- **The Logic:** What are the specific "must-have" rules the system needs to follow?
- **Exception Handling (The "What ifs"):** This is the most common missing piece. What happens when the standard journey is interrupted? If a manager rejects a request, where does it go? If a deadline is missed, who gets an alert?

Example: Process Narrative

Use this format if you prefer to describe your process in detail. Provide a description for each major phase or capability of the application.

Phase 1: [Phase Name, e.g., Claim Evaluation]

- **The Standard Journey:** Describe the "Happy Path" from start to finish. Who is involved and what are they trying to do?
 - Example: Once a claim is submitted, it arrives in the Supervisor's queue. They review the photos and notes to determine if the claim is covered. Their goal is to set a final payout amount.

- **The Fork in the Road (Exceptions):** What happens if the user makes a mistake or needs to send the work back?
 - Example: If the photos are blurry, the Supervisor must be able to click 'Request More Info.' This should send the claim back to the Adjuster with a mandatory comment. The process pauses until new photos are uploaded.
- **The System Rules:** What automated logic should happen here?
 - Example: The Supervisor cannot approve a payout that is higher than the customer's policy limit. The system should flag any payout over \$10,000 for additional VP approval.

Example: Structured Story Map

Use this format if you prefer a list of specific requirements. Ensure you include the "Why" (Value) and the "What If" (Exception) for every row.

Activity	User Story (Detailed)	Who does this?	Success Criteria & Exception Rules	Priority
Example: Intake	As an Adjuster , I want to enter a Policy Number , so that I can instantly verify if a customer is covered .	Adjuster	1. System must pull Policyholder Name and Effective Date from the database. 2. Exception: If the policy is expired, display a red 'Coverage Warning' and block further entry.	High
Example: Review	As a Supervisor , I want to review the evidence photos , so that I can verify the damage before approving payment .	Supervisor	1. Photos must be viewable in a gallery view. 2. Exception: If photos are blurry, Supervisor must be able to click 'Request New Photos,' which sends a notification back to the Adjuster.	High
Example: Decision	As the System , I want to auto-calculate the deductible , so that the Adjuster doesn't have to do manual math .	System	1. Deductible is pulled from the Policy Record. 2. Rule: Total Payment = (Total Loss - Deductible). 3. Exception: If Total Loss is less than the Deductible, payment amount is set to \$0.	Medium

[Enter Stage]	As a [Persona], I want to [Action], so that [Value].	[Persona/System]	Requirements: [Rule 1, Rule 2] Exceptions: [What if X goes wrong?]	[P1-P3]
----------------------	---	------------------	---	---------

4. Screens & User Experience

What to think about: What are the 3 most important screens a user will interact with? Do they need a "Command Center" (Dashboard) or just a simple data entry form? Will they be using this on a mobile phone in the field?

- **Visual Goals:** What is the most important information a user needs to see at a glance?
- **Device Needs:** Does any part of this process happen away from a desk (e.g., taking photos of damage on a phone)?
- **Reporting:** What charts or lists does management need to see every morning?

Screen Name	Who is it for?	What is the primary goal?	Key Information / Charts to Include	Device Priority
<i>Example: Adjuster Dashboard</i>	<i>Claims Adjuster</i>	<i>To see what I need to work on today.</i>	<i>List of assigned claims, 'Days to SLA' countdown, and total pending payments.</i>	<i>Desktop & Tablet</i>
<i>Example: Claim 360 View</i>	<i>Adjuster & Supervisor</i>	<i>To see everything related to a single claim in one place.</i>	<i>Claimant details, incident timeline, photo gallery, and linked policy info.</i>	<i>Desktop Only</i>
<i>Example: Field Intake Form</i>	<i>Field Inspector</i>	<i>To capture damage details while at the customer's house.</i>	<i>Camera upload button, GPS location, and damage description notes.</i>	<i>Mobile / Smartphone</i>
[Enter Screen Name]	[Persona]	[Goal of this view]	[Top 3-5 data points or charts]	[Mobile/Desktop]

5. Business Rules

What to think about: These are the "if/then" statements that act as the guardrails for your application. Without these, the system cannot automate decisions or ensure data quality. Think about the specific logic, calculations, and validations that must happen automatically.

- **Validation:** What prevents a user from entering "bad" data (e.g., a phone number with too many digits or a future date for an accident that already happened)?
- **Thresholds:** Are there specific dollar amounts or percentages that trigger a change in the process (e.g., a claim over \$10,000 requiring a VP signature)?
- **Timing (SLAs):** How quickly must a task be finished before it is considered "late," and who should be notified?
- **Routing:** When should a record be automatically moved to a specialized team (e.g., "Special Investigations") based on specific risk factors?

5.1 Validation Rules

Focus on field-level format, mandatory requirements, and date range logic

Rule Name	Logic / Description	Enforcement
Example: Mandatory Fields	<i>Claim type, policy number, and incident date are required to save a record.</i>	<i>System Validation</i>
Example: Date Range	<i>The incident date cannot be more than 30 days in the future.</i>	<i>System Validation</i>
[Enter Rule Name]	[Describe the data format or requirement]	[System/Manual]

5.2 Process & Data Rules

Focus on calculations, authority limits, and Service Level Agreements (SLAs)

Rule Name	Logic / Description	Enforcement
Example: Authority Limits	<i>Adjusters can only approve payments up to their authorized limit (e.g., \$5K for Junior, \$25K for Standard).</i>	<i>System</i>
Example: SLA Calculation	<i>Response time is 24 hours for emergency claims and 48 hours for high priority.</i>	<i>System with Notifications</i>
Example: Reserve Logic	<i>Reserve amounts must be automatically calculated based on the claim type and severity factors.</i>	<i>System Calculation</i>
[Enter Rule Name]	[Describe the calculation or limit]	[System/Manual]

5.3 Business Logic Rules

Focus on complex decision-making and automatic assignment.

Rule Name	Logic / Description	Enforcement
Example: Fraud Detection	Claims with 3+ risk factors must be routed to the Special Investigation Unit (SIU) for review.	System Automation
Example: Threshold Approval	Any claim exceeding \$50,000 must be automatically routed to senior management for approval.	System Routing
[Enter Rule Name]	[Describe when a record should be moved or escalated]	[System/Manual]

6. Data Model

What to think about: Data is more than just a list of fields; it's the skeleton of your application. When defining your data, think about how different "nouns" (entities) relate to each other.

- **The "One-to-Many" Rule:** Does one Customer have many Claims? Does one Claim have many Documents? Defining these relationships helps the system understand how to display information.
- **Reporting Needs:** If you want to run a report on "Claims by State," you must ensure "State" is a specific data point, not just buried in a text paragraph.
- **Data Sources:** Is this information being created fresh in the app, or are we pulling it from an external "Source of Truth" like a Policy database?

6.1 Primary Transactional Entities

These are the core "nouns" of your business. These entities are created, edited, and moved through your workflow every day.

Entity Name	Business Description	Key Data Points (Fields)
Example: Claim	The central record representing a request for coverage.	Claim ID, Date of Incident, Description, Total Estimated Loss, Assigned Adjuster.
Example: Policy	The contract governing the coverage for a specific customer.	Policy Number, Effective Date, Expiration Date, Coverage Limits, Deductible Amount.

[Enter Entity]	[What does this represent?]	[List the 5-10 most important fields]
----------------	-----------------------------	---------------------------------------

6.2 Reference & Support Tables

These are the "Lookup" lists that populate your dropdown menus. They ensure that every user uses the same terminology (e.g., everyone selects "Closed" instead of some typing "Done" and others typing "Finished").

Table Name	Description	Example Values / Options
<i>Example: Claim Status</i>	<i>The standard stages a claim moves through.</i>	<i>New, Under Review, Pending Information, Approved, Denied, Closed.</i>
<i>Example: Loss Type</i>	<i>The category of damage being reported.</i>	<i>Collision, Comprehensive, Liability, Medical, Property Damage.</i>
<i>Example: State/Province</i>	<i>Geographic regions for tax and legal routing.</i>	<i>Alabama, Alaska, Arizona...</i>
[Enter Table]	[What list does this provide?]	[List a few example options]

7. Integrations

What to think about: Applications rarely live in a vacuum. They need to "talk" to other systems to pull in existing data or send updates out. When defining an integration, think about the **Conversation**:

- **The Source of Truth:** Which system "owns" the data? (e.g., The app might *show* policy details, but the Policy System *owns* them).
- **Direction:** Is our app just listening (Inbound), or are we telling another system to do something (Outbound)?
- **The "What If":** What should the user see if the external system is down or slow? Should the app stop working, or can they continue and "sync" later?
- **Trigger:** Does the data move the moment a button is clicked (Real-time), or does it happen once a day in a big group (Batch)?

7.1 Integration Inventory

Use this table to list every system your application needs to connect with. Focus on the business purpose of the link.

System Name	Business Purpose	Direction	When does it happen? (Trigger)	What happens if it fails?
<i>Example: Policy Admin System</i>	<i>To retrieve current coverage and deductible details for a claimant.</i>	<i>Inbound (Pull)</i>	<i>When an Adjuster enters a Policy Number.</i>	<i>Show an error: "System Unavailable. Enter details manually."</i>
<i>Example: Payment Gateway</i>	<i>To issue the final settlement payment to the customer's bank account.</i>	<i>Outbound (Push)</i>	<i>When a Supervisor clicks 'Approve Final Payment'.</i>	<i>Flag the claim as 'Pending Payment' and retry every hour.</i>
<i>Example: Document Cloud (S3)</i>	<i>To store and retrieve large accident photos and legal PDFs.</i>	<i>Bi-Directional</i>	<i>Whenever a file is uploaded or viewed.</i>	<i>Block the upload and notify the user to try again.</i>
[Enter System Name]	[Why do we need this?]	[In/Out/Bot h]	[The Action that starts it]	[Retry / Error Message / Manual]

7.2 Data Migration

What to think about: If you are replacing an old system, do you need to move the "History" into the new app?

- The Cut-off: Are we moving all historical claims from the last 10 years, or just the "Open" ones?
- Data Cleanup: Is the old data "dirty"? (e.g., misspelled names or missing dates). Who is responsible for cleaning it before it enters the new system?

Example: "We must migrate all 'Open' and 'In-Progress' claims from the Legacy SQL Server. Closed claims older than 2 years will remain in the old system for read-only archive purposes."

8. Security and Access Matrix [Optional]

What to think about: Now that we know your **Personas** and your **Data**, we need to define the "Wall" between them. Not everyone should see everything.

- **Sensitive Data:** Are there fields (like Social Security Numbers, Bank Details, or Salaries) that should be masked or hidden for certain roles?
- Who is allowed to *Delete* a record? Who can *Re-open* a claim after it's been closed?

- **Visibility:** Does an Adjuster only see their own claims, or can they see their whole team's work?

Data Entity / Action	Persona: Adjuster	Persona: Supervisor	Persona: Executive
<i>Example: Claim Record</i>	<i>View & Edit (Assigned Only)</i>	<i>View & Edit (All)</i>	<i>View Only (All)</i>
<i>Example: Payment Info</i>	<i>View Only</i>	<i>View & Edit</i>	<i>Hidden</i>
<i>Example: Delete Claim</i>	<i>No Access</i>	<i>No Access</i>	<i>Full Access</i>
[Enter Entity/Action]	[Access Level]	[Access Level]	[Access Level]

9. Glossary [Optional]

What to think about: Every industry has its own "language." Don't assume the developers or new stakeholders know what your acronyms mean. Defining these here prevents costly misunderstandings during the build phase.

Term / Acronym	Definition
Example: FNOL	First Notice of Loss: The initial report made to an insurance provider following a loss, theft, or damage of an insured asset.
Example: SLA	Service Level Agreement: The contractually obligated timeframe in which a task must be completed (e.g., "Initial contact within 24 hours").
Example: Deductible	The amount of money the policyholder must pay out-of-pocket before the insurance company covers the remaining costs.
Example: Subrogation	The process by which an insurance company pursues a third party that caused an insurance loss to the insured.
Example: CAT Claim	Catastrophe Claim: A claim resulting from a large-scale event (e.g., hurricane, wildfire) that requires a different priority workflow.
[Term]	[Enter definition here]

10. Appendix [Optional]

What to think about: Use this section to house supplemental documents that are too large to fit in the main body. If it's a visual aid or a detailed spreadsheet, link it or embed it here. Can include screenshots, mockups, process diagrams, and other supporting materials.