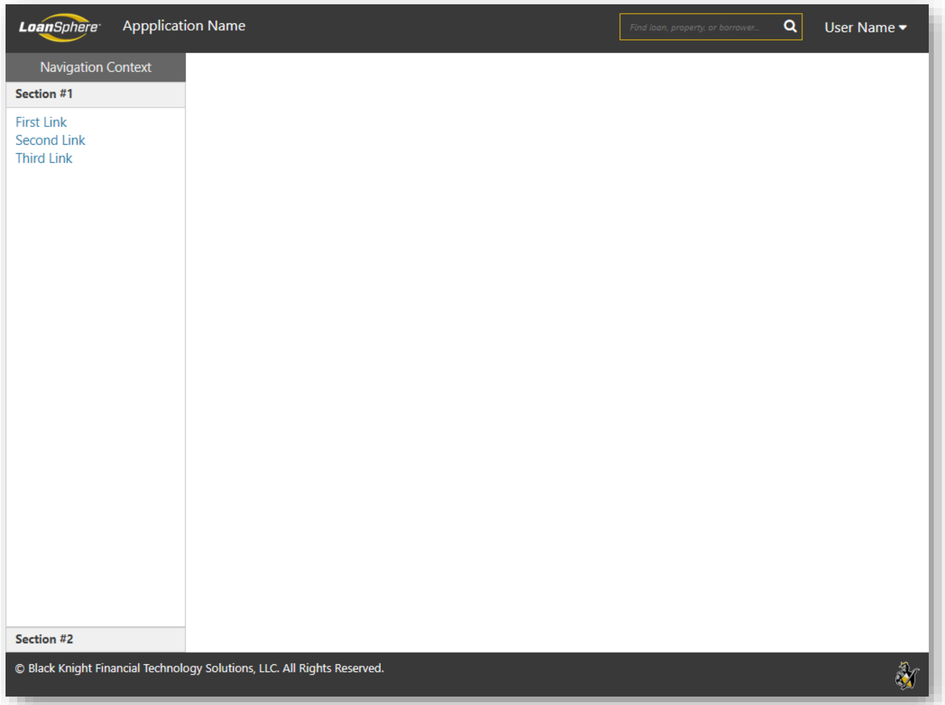
PROJECT INTRODUCTION

Design System

Leading Scalable, Accessible Design System Development Across Applications

Company: Black Knight, Inc.
Timeframe: 2016 – 2019+



This application frame template, built using Atomic Design principles, was applied across multiple apps to ensure consistency and scalability.

Project Summary

Initiated to modernize the UI for LoanSphere's Loss Mitigation functionality, the design system expanded to support multiple initiatives, ensuring consistency across a large developer base.

Results

- Adopted across 5+ projects, ensuring enterprise-wide consistency.
- Reduced development time by 30%, streamlining front-end workflows.
- Improved product quality with a unified, accessible UI system.

Relevance

This project highlights my ability to lead scalable design system development, collaborate across teams, and deliver high-quality, accessible solutions that enhance user experience.

Responsibilities

- Design System Creation
- Developer Support
- System Evangelism

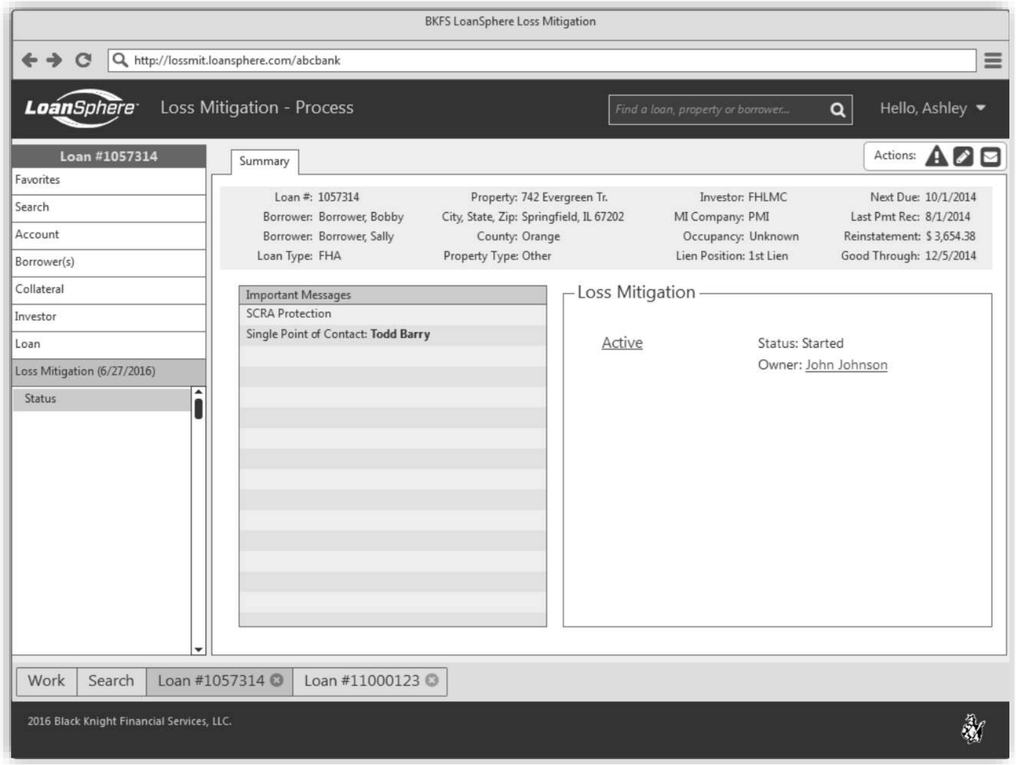
BACKGROUND

Creating a design system to modernize and streamline UI development.

This project modernized and streamlined UI development across Black Knight's applications, starting with Silverlight Remediation for Loss Mitigation and expanding to other projects like PCN. Leveraging lessons from my first design system, this initiative enhanced consistency, scalability, and usability across teams and applications.

Key Steps:

- **Migrating from Silverlight:** Transitioned from XAML to HTML, CSS, and JavaScript, enabling a more flexible and modern tech stack.
- **Implementing Atomic Design Principles:** Developed reusable UI components to standardize design and improve efficiency.
- **Ensuring a Unified Look and Feel:** Delivered consistent design across multiple applications, enhancing scalability and usability.
- **Evangelizing the Design System:** Partnered with architects and developers to ensure adoption and alignment with business goals.



An early prototype from the Silverlight Remediation project inspired the design system's foundation of consistency and scalability.

RESEARCH & DISCOVERY

I engaged in **strategic meetings** with the VP and development architects to address the deprecation of Silverlight, collaborating to set the foundation for a scalable and maintainable solution.

1. Goals

- Develop an HTML5-compliant SPA with stateless session management.
- Adopt industry-standard patterns to streamline hiring and onboarding.

2. Requirements

- Ensure separation of concerns across application layers.
- Migrate business rules to the presentation layer, creating a scalable solution supporting future growth.

3. Challenges

- Address the demand for a robust library of reusable controls.
- Simplify the UX and transition from outdated technologies while serving as the sole UX designer for Black Knight's Servicing Technologies division.

These efforts laid the groundwork for a scalable, maintainable system that improved developer efficiency and set a new standard for cross-application consistency.

MY ROLE

I independently **created and implemented** the design system. I collaborated closely with developers to ensure **consistent design**, enhanced **scalability**, and efficient development across all applications.

System Creation

Spearheaded the design system's development using HTML, CSS, and jQuery UI, defining principles and guidelines that ensured consistent, scalable UI components.

Developer Support

Provided code snippets and documentation, accelerating development timelines and ensuring consistent implementation across teams.

System Evangelism

Promoted the design system through workshops, virtual meetups, and direct communication, driving adoption and fostering cross-team alignment.

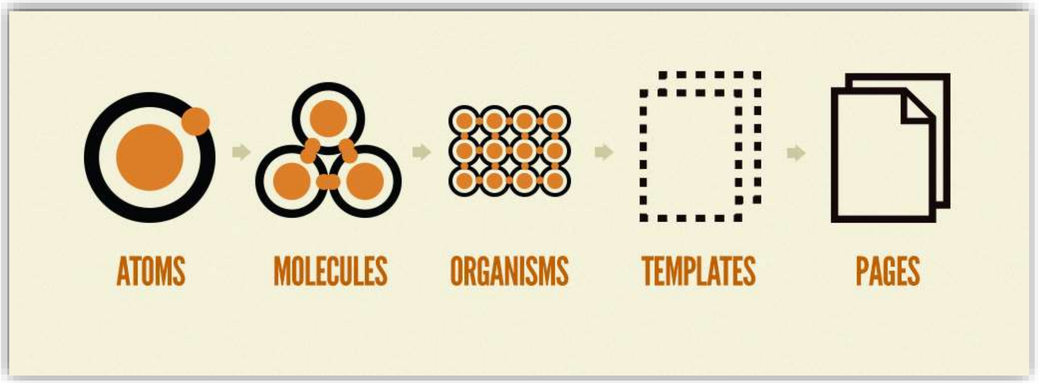
MY ROLE

Design System Creation

Foundation Establishment

- **Inspiration from Brad Frost's Atomic Design**
  Attending Brad Frost's 2016 talk inspired the hierarchical and reusable component structure of this system, applying Atomic Design principles to achieve scalability and consistency.
- **Technology-Agnostic Approach**
  Designed a flexible, browser-rendered system to ensure adaptability across different technology stacks and future-proof development.
- **Evolution from Previous Experience**
  Leveraged lessons from my first design system, applying insights from defining XAML code to create a scalable and robust framework.

Guiding framework for component hierarchy.



Meeting Brad Frost, whose Atomic Design principles directly influenced the creation of this scalable system.

MY ROLE

Design System Creation

Component Modernization

I spearheaded the transition from Rincon's XAML-based application to a modern, HTML/CSS framework. This modernization effort aligned with Atomic Design principles, enabling consistent and scalable UI development across multiple applications.

Key Contributions:

- **Transition to Modern Framework**
  Replaced XAML components with HTML, CSS, and JavaScript implementations to align with current industry standards.
- **Comprehensive Component Review**
  Conducted a detailed inventory of all UI components, categorizing them as atoms, molecules, or organisms.
- **Documentation and Scalability**
  Provided thorough documentation to ensure reusable and scalable UI solutions across projects.

This inventory guided the migration from XAML to HTML/CSS, ensuring a scalable, maintainable UI framework that adhered to Atomic Design principles.

Name	Count	Atomic	RT Comments
AutoCompleteBox	4	Atom	Lookup functionality
MasterDetailsGrid	486	Organism	Complex control
BorderGroupExpander	481	Molecule	HTML DIV with a button that changes visibility of contents
Button	42	Atom	HTML <Button>
CheckBox	115	Atom	HTML <Input type="checkbox" />
ComboBox	9	Atom	HTML DropDownList + Search/Filter ability
ComboBoxItem	16	Atom	HTML <Option>
CurrencyBox	88	Atom	HTML <Input type="text" placeholder="$0.00" /> + JS input mask
DataGrid	49	Molecule	Complex control
DateBox	18	Atom	HTML <Input type="text" placeholder=" / / " /> + JS input mask + calendar pop-up In the strictist definition, a DateBox is just input with a mask but I want our date fields to also have a date picker pop-up
DatePicker	33	Atom	Replaced by DateBox definition
DateTimePicker	2	Atom	DateBox definition with time added to display and pop-up
DropDownList	82	Atom	HTML <Select> without the "Size" property
DropDownListItem	14	Atom	HTML <Option>
Image	246	Atom	HTML <IMG />
IntegerBox	3	Atom	HTML <Input type="number" />
Timeline	24	Organism	
Label	418	Atom	HTML <Label />
ListBox	22	Atom	HTML <Select Size="[number of items to show]" />
ListBoxItem	5	Atom	HTML <Option>
MaskedEditBox	6	Atom	generalized version of the CurrencyBox, DateBox, etc.
MultiLineTextBox	30	Atom	HTML <TextArea>
NumericBox	3	Atom	HTML <Input type="number" />
PasswordBox	4	Atom	HTML <Input type="password" />
PercentRateBox	10	Atom	HTML <input type="number" /> + JS input mask
Popup	17	Molecule	In the strictist definition, a pop-up is more like a tooltip.
PopupWindow	24	Molecule	For a modal window, we'd need some HTML and JS with buttons to save/close. For a non-modal window, JS Window.Open

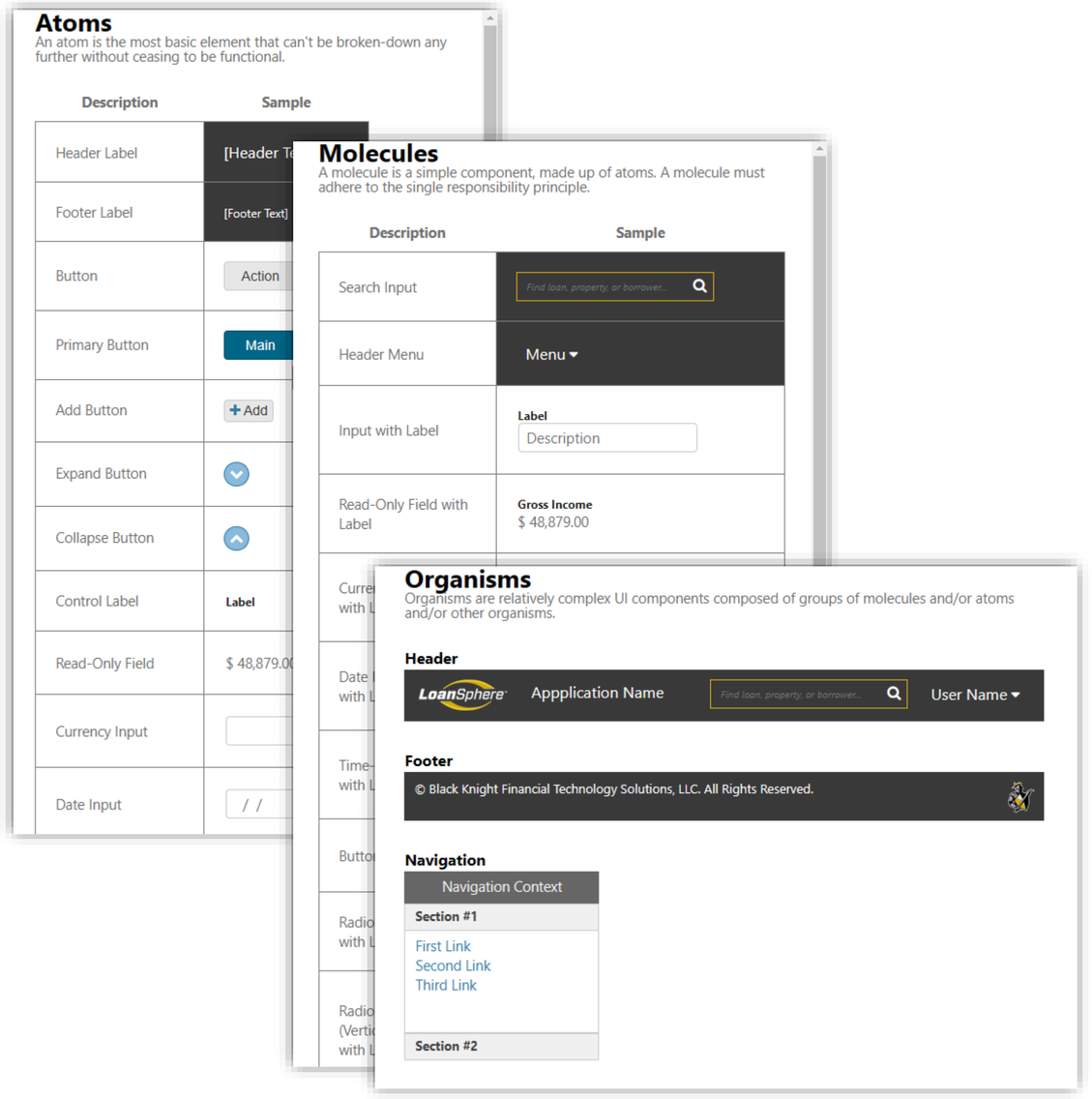
MY ROLE

Design System Creation

Construction

- **Website Creation**
  Developed a centralized design system website, providing developers with access to design principles, reusable UI components, and best practices for consistent application development.
- **Component Design**
  Designed accessible UI components aligned with Black Knight's branding, incorporating stakeholder feedback and WCAG AA compliance to ensure usability and inclusivity.
- **Deployment**
  Collaborated with release engineers to deploy the design system on the company intranet, ensuring seamless integration through robust version control and deployment practices.

These screenshots showcase atomic components built for scalability and efficiency, enabling developers to create consistent and accessible applications.



Developer Support

A key objective of the design system was to ensure consistent adoption and application by developers. To bridge the gap between design intent and implementation, I provided resources and guidance that empowered developers to apply components accurately and efficiently.

Code Snippets

Developed and shared HTML code snippets to visually demonstrate component usage and rendering, reducing misinterpretation and ensuring consistency across teams and frameworks.

Screenshot of an HTML code snippet describing a "Tile" component

```
<div class="tile">
   <header>Borrowers</header>
   <summary>
      <i class="fa fa-users fa-2x"></i>
      <ul>
         <li>Behrensmeyer, Anna <i class="fa fa-star muted"></i></li>
         <li>Feynman, Richard</li>
      </ul>
   </summary>
</div>
```

Documentation

Authored comprehensive documentation outlining design principles such as progressive disclosure and color guidelines, enabling developers to effectively adopt and implement the design system:

- Foster a clear understanding and streamlined adoption of the design system across teams.
- Enhance implementation accuracy by reducing guesswork and providing actionable guidance.

Screenshot of the rendered "Tile" component

MY ROLE

System Evangelism

Beyond implementing the design system for the Silverlight Remediation project, I focused on driving division-wide adoption. Through collaboration, communication, and continuous updates, I ensured the system became integral to UI development across teams.

Building Relationships & Trust

1. Earned trust as the primary UI development advisor, resolving complex design challenges for engineers.
2. Gained stakeholder confidence by delivering polished, reliable solutions that highlighted design excellence.
3. Served as the division's sole designer, providing cross-functional support and promoting system-wide consistency

Leveraging Developer Networks

1. Collaborated with developers who later championed knowledge-sharing, scaling adoption via virtual meetups.
2. Presented at virtual meetups, demonstrating use cases and best practices to streamline adoption.
3. Fostered cross-team collaboration, driving adoption and ensuring alignment with organizational goals.

Communicating Openly

1. Established open channels for developer feedback, fostering an iterative approach to system improvement.
2. Maintained consistent communication through emails, posts, and meetings, keeping stakeholders informed.
3. Facilitated continuous dialogue, ensuring developers received timely guidance and support.

Continuously Updating

1. Adapted the design system to evolving project needs, ensuring relevance and scalability.
2. Refined documentation and components based on developer insights, promoting ongoing usability.

Successfully Drove Consistency & Efficiency with a Comprehensive Design System

Key Outcomes & Results

- **Enhanced Consistency**
  Achieved a unified look and feel across multiple projects, significantly enhancing user experience and interface consistency.
- **Increased Efficiency**
  Enabled faster and more accurate implementation of UI components, reducing development cycles and boosting productivity.
- **Broad Adoption**
  Adopted across multiple division projects, the design system maintained its utility and scalability beyond my tenure.

What I Learned

- **Collaboration is Key**
  Strong developer and stakeholder relationships were crucial for smooth implementation and adoption.
- **Adaptability Drives Success**
  Flexibility and openness to innovative ideas allowed the design system to evolve and scale efficiently.
- **Resource Limitations Require Strategic Solutions**
  With limited design resources, some developers implemented UI independently, requiring **ongoing documentation & training** to ensure consistency.