Reservoir

NYPL RESERVOIR DESIGN SYSTEM

Governance for Reservoir Integration

The NYPL Reservoir Design System (DS) should be utilized as the basis for UI/UX designs and functionality within web-based projects handled by the NYPL portfolio groups. This governance process defines how Product Managers and the Product Design team should approach using the DS.

Step 1: Look for Existing Solutions

Use the design system to help design and build new work.

Product teams should default to using existing DS components to help create new product work. DS components have been thoroughly explored for UX and accessibility best practices and standards, so the use of DS components in design and development will be more efficient and lead to consistent look and feel in NYPL's digital ecosystem.

When designing a new page template or adding new functionality on an existing template, a product designer should use the Reservoir Storybook to review the design and functionality of components that are available in the latest version of the DS and look for components that fulfill all needs and requirements.

Step 2: Request Help to Find a Solution

What happens when the design system’s components don’t exist or don’t fulfill requirements?

If an existing component doesn’t fulfill all the requirements for a specific need or if the DS doesn’t contain a component that directly correlates to what is needed – or if it's just not clear if the DS has what is needed – a product team should reach out to the Design System Team (DST) through the #design-system Slack channel to ask for assistance. The DST will work with the requesting product team (RPT) to find an existing solution that meets requirements or to determine if new work is needed.

Step 3: Determine Direction of New Work

If new work is needed, is it a "gooseneck" or a viable candidate for the design system?

If the RPT and the DST both agree that new work is necessary, the new work will be classified as a "gooseneck" or a "candidate." A gooseneck is a one-off component or variant that pertains to one specific product or use case, while a candidate is a component or variant that could serve multiple products.

If the new work is labeled a gooseneck, nothing new will be added to the DS. Instead, the work will be added to the RPT's backlog in accordance with that team's component guidelines. The RPT will own the new work and be responsible for its execution. The new work will run independently of the DS and the remaining steps in the DS governance process will no longer apply.

If the new work is labeled a candidate, the RPT and the DST will work together to determine the level of effort and prioritize the work. With that, a ticket will be added to the DS Jira backlog to handle the work to be done, a new work-in-progress (WIP) Specification Document (Spec Doc) will be created and the governance process will continue to move forward.

Step 4: Initial Concept and Spec Doc

Start to solidify the structure of a candidate and how it will work.

If no prior documentation or development has been completed for the new work (candidate), the RPT – lead by Product Design – will produce the initial concepts for the candidate in the form of a wireframe, hand-drawn sketch, lo-fi comp, in-browser prototype, or any other artifact that quickly and clearly articulates the use case and defines the work to be done. The *Introduction* and *Business Drivers* sections of the candidate's Spec Doc should be completed in conjunction with the initial concept prototype.

The DST and RPT will review the initial concept and Spec Doc and determine whether all necessary requirements have been met. If something is missing, the RPT will iterate over the initial concept and Spec Doc and both teams will continue to review until the requirements are met.

If prior work has been completed for a candidate (i.e. component designs exist in the DS Figma Main file), the DST and RPT will assess the level of detail provided by the existing work to determine if the prototype can be bypassed and only require the Spec Doc to be completed.

Step 5: UX Exploration and Designs

Formalize the functionality and visual treatment of the candidate.

Once the basic concept of a candidate has been established, a designer from the Product Design team – often also a member of the RPT – will use the Reservoir-focused UX Process as a guide to conduct UX exploration and design ideation to finalize visual designs for the candidate. The UX exploration and ideation process is also a time to start thinking about the accessibility requirements, so the designer should set aside time to meet with an accessibility specialist to work through any accessibility concerns.

When the visual designs for a candidate have been approved, the product designer will work with the DST to move the visual designs into the DS Figma Main file. When visual designs are moved into the Figma Main file, the "auto-layout" and "variants" functionality available in Figma should be utilized to create a flexible component that can be easily configured in consuming Figma files. Additionally, the component designs in Figma must adhere to best practices regarding layer management (i.e. adding color and text styles, naming layers properly, using coherent and minimal groups, etc.).

The product designer will fill out the *UX Considerations & Best Practices* section of the Spec Doc. Additionally, with the assistance of the DST, the product designer will complete the *Functional Specifications* section of the Spec Doc including a compiled list of UI specifications and functional requirements.

Step 6: Complete Spec Doc

Add technical specifications and prepare to handoff for development.

The remaining sections in a candidate's Spec Doc – *Accessibility*, *React Component Props*, *Code Examples* and *References* – will be completed by the DST. The DST will also include any ad hoc information that engineers may require to complete the development for the component.

The DST will start to compile details for the *Accessibility* section, but the final language should be finalized and approved by an Accessibility Specialist to ensure that the accessibility specifications meet WCAG 2.1 AA standards.

After all sections of the Spec Doc have been filled in, the document will be reviewed by the DST and the product designer to check for technical viability and consistency with DS best practices and patterns. The DST will continue to iterate over the technical aspects of the document until all questions or concerns are resolved. At this point, the WIP status of the Spec Doc will be removed.

For more information about the Reservoir Spec Docs, refer to Specification Document Definition.

Step 7: Development and Testing

Build the candidate in code and bang on it to make sure it works!

When the Spec Doc has been completed and the candidate is ready for development, the RPT and the DST will determine if the development should be completed by an engineer from the DST or by an engineer from the RPT.

The assigned engineer will use a candidate's Spec Doc and designs in the Figma Main file to execute the development of a candidate as part of the Reservoir Design System React Component Library. The development will follow the DS code guidelines, with an aim toward reuse, flexibility, composability, accessibility, performance and other best practices. The final functional code must include the component file with relevant unit tests and a Storybook documentation page.

For review and basic QA testing, the candidate will be included in the Reservoir Storybook. The story pages within the Reservoir Storybook will include code examples and information detailing how to implement the candidate in a consuming app.

In addition to the Reservoir Storybook, the candidate component as part of the Reservoir Design System React Component Library will be incorporated into Turbine, a consuming app dedicated to testing the Reservoir Design System. The Turbine Next.JS web application will have a dedicated page for the candidate and the implementation will mirror any production-ready application using the candidate component.

QA testing of the candidate will include the following areas: functionality, accessibility, variations in content form, cross-browser support, cross-platform support, mobile/responsive support and visual representation. For specific details about the QA process, please refer to the Design System’s QA Process.

Step 8: Final Review and Sign-off

Get the blessing of the requesting product team.

The RPT and the DST will meet and use branch previews to perform a final review of the candidate in the Reservoir Storybook and the Turbine test app. If the RPT does not approve the candidate as it has been implemented, the deficiencies will be noted and the DST will make the appropriate code changes and, if necessary, update documentation in the Spec Doc and the component story accordingly. This cycle will repeat until the RPT gives a final sign-off on the candidate.

Step 9: New Design System Release

The candidate is ready for prime time.

The DS follows an incremental release schedule that aligns with the DST's 2-week sprint cycle. After the final sign-off for the candidate, the candidate's feature branch will be merged into the development branch and will be included, along with other new work that was completed within the current sprint, in the next scheduled release. The new release will follow the semantic versioning guidelines used by the DS.

After the new release has been successfully publish, the DST will announce the new release on the #design-system Slack channel.

Step 10: Adoption and QA by Product Team

Get going and send us a postcard.

Once the new version of the DS is available, the RPT will pull the new version of the DS into their consuming app and test the candidate.

If questions come up or bugs are identified, the RPT should submit a new ticket to the DS Jira backlog. If a bug is being reported, the ticket should include steps to reproduce the bug and, if possible, a link to the github branch where the candidate is being implemented.

If no bugs are found, the candidate is ready to fly on its own!