Deloitte.

NAVIENT REFINANCE

Visual Identity Document

Version 1.0
Published March 10, 2017
Created by Deloitte Digital

OVERVIEW

High level view of the application and the relevant form factors encompassing the design.

This chapter contains:

- Introduction
- Form factor considerations

Overview

1. Introduction

The purpose of this document is to provide a guide for the development of the Navient Refinance application, particularly as it pertains to the implementation of certain key areas of the site such as form controls, interactions and transitions and general site styles. This is not a replacement for the full set of Wireframes and Visual Designs/Specification. It is meant to serve as a tool to assist the development and QA teams during development and testing phases.

Images

Where applicable, the document will feature final visual design comps; however, not every screen has been represented via final visual design. As such, there is a section at the end of the document focused on visual design and the handling of areas such as layouts, styles, fonts and colors.

Notations

Notations are utilized in the document to provide further explanation and direction. Where appropriate, we have incorporated zoomed in areas, series, and other styles to document desired functionality and overall direction. We belief in most cases, the designs should be self explanatory; however, where there is possibility for multiple interpretations, we have provided specifc examples and guidance. ***Numbers used in the document are for illustrative purposes only.***

2. Form Factor Considerations

Desktop

Standard visual design widths utilized in this document are 1440 pixels wide for desktop. Tablet has not been addressed specifically, instead, the Deloitte Digitial team assumes that horizontal orientation will utilize desktop design patterns and vertical orientation will align with specified mobile design patterns as outlined in this document.

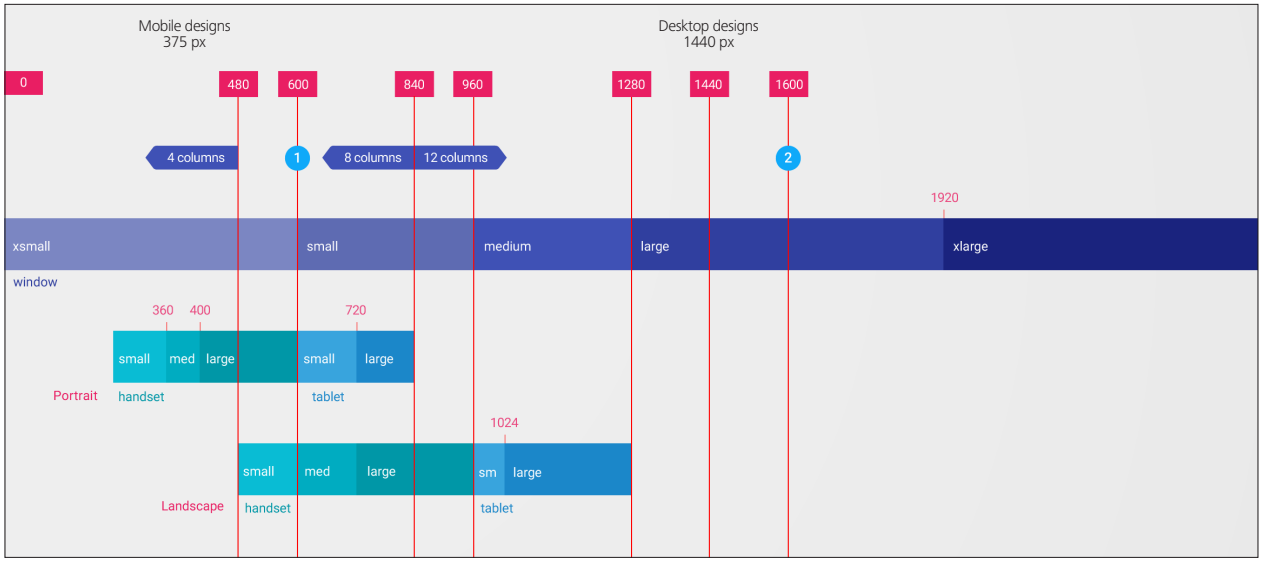
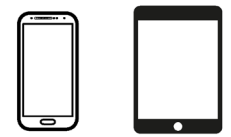


Figure 1: Form factor widths from Angular.js framework

Desktop design



Mobile design

Both form factors will allow for browser-specific controls such as the back button, however, the application will also provide means for the user to freely move backwards by way of onscreen navigational controls and breadcrumb. Due to the linear nature of the application, the user cannot "jump" forward in the process. They must follow the natural flow of the application which requires key information to be provided before proceeding to the next screen. These areas will be outlined in this document and in the wireframes.

Mobile

Mobile designs are created at 375 px width. As noted earlier, when viewed on tablet, the system display the mobile version when in portrait, while showing the desktop version in landscape. As with desktop, mobile will utilize browser controls as well as native phone controls where specified in this document. Several key areas will be optimized for mobile. As a result, while we sought consistency between the platforms, there are differences between mobile and desktop.

APPLICATION ARCHITECTURE

In-depth look at the form controls including functionality and system integration.

This chapter contains:

- Forms
- Application states
- Error handling & messaging
- Modals

Application Architecture

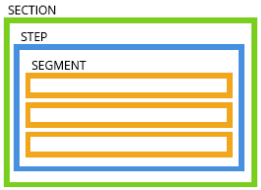
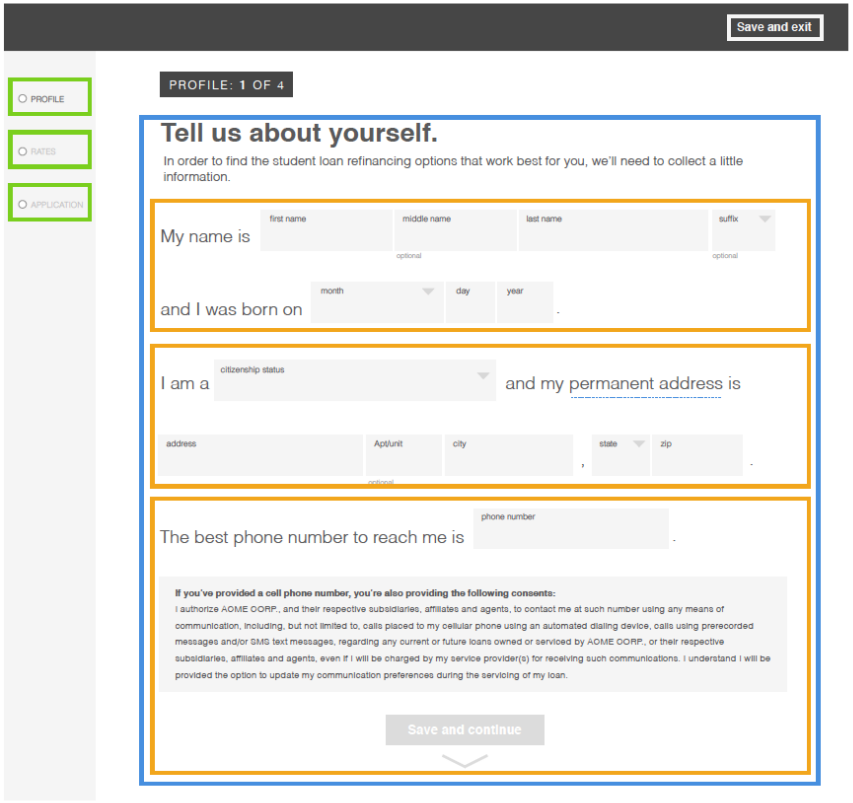
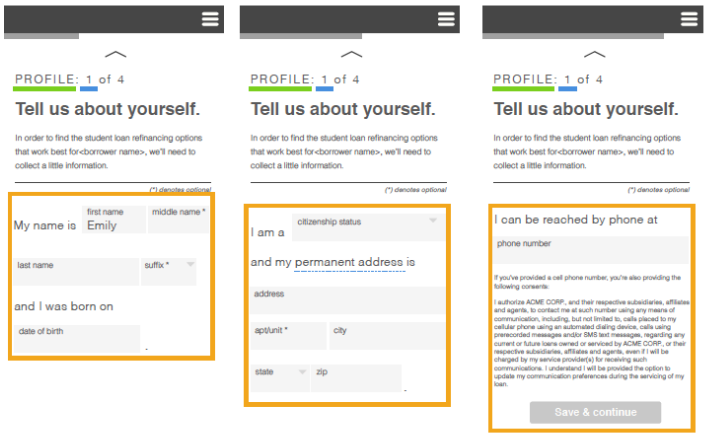


Figure 2: Form architecture



Desktop architecture



Mobile architecture

1. Forms

Form Architecture

The system is a series of form fields, organized into sections that the user must complete before proceeeding to the next section. Each section will load after the previous section has sufficiently been completed by the user. *(See the* Transitions *section for details on the transition)*

Each of these 3 sections is a step which must be completed before proceeding to the next step. Each step includes segments. A segment is one or more sentences that are logically connected and like steps and sections, must be completed in the linear order in which they are presented. The key aspect of these segments is that on desktop and mobile, the segment sizes will vary in order to provide the best experience. Steps and sections will be identical across mobile and desktop. In order to understand how the components of each form come together, figure 2 shows both mobile and desktop modules and their relationship to each other.

Application Architecture (cont'd)

Form Tone
The Navient application will use a conversational flow in lieu of a more traditional form layout in the majority of the form segments. The form fields will be embedded into a sentence structure as shown in Figure 3 and these sentences will make up these form segments. In certain cases, mobile and desktop will differ in the number of sentences contained within a segment and those will be noted in the wireframes. The verbiage and structure of the sentences will remain unchanged between form factors.

Form Fields
The application utilizes standard form fields embedded into sentences for the following form elements:

- Text input
- Checkboxes / radio buttons
- Text areas
- Select dropdown
- Type ahead/smart selection
- File uploads (secure)
- Submit buttons

Form Labels
The presentation of the form fields will utilize 508-compliant labels in order to ensure the application is accessible. In order to maintain the conversational tone and design, the system should utilize infield top aligned form labels.

Form Label Layout
On Load: All form fields will show the label full sized within the field when the form block/page loads. The labels will be the same size as the user typed text. *(See* Design *section of this document for specifications)*.
On Click: Once the user has clicked into a field, the label will reposition itself to align to the top left corner of the form field. *(See* Transition *section for details on the transition between states.)* The user's entered text will then display where the original label was located, in a darker color, as shown in figure 4.

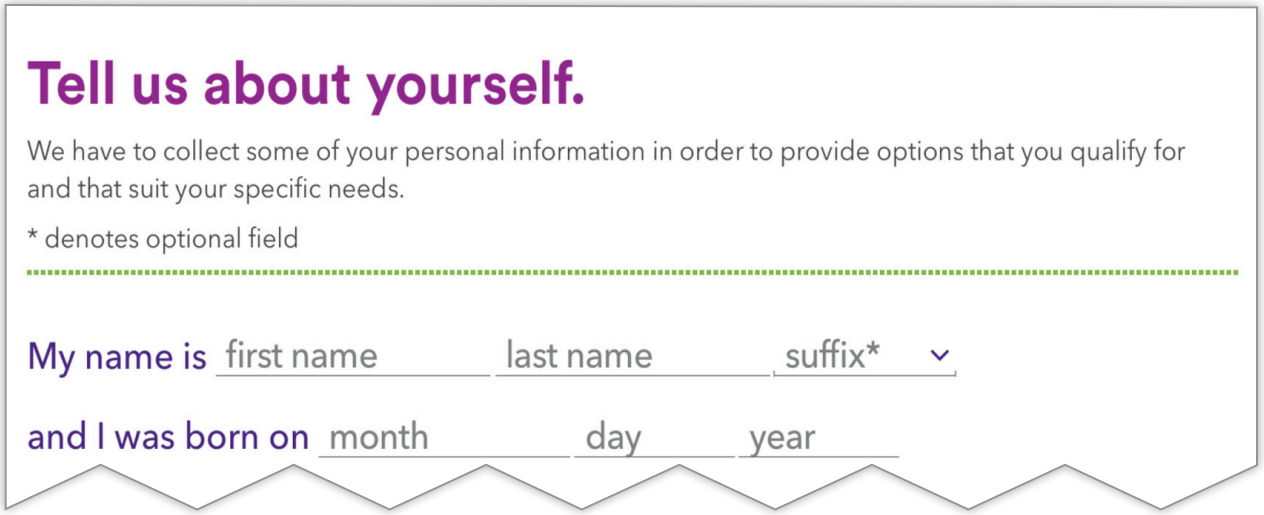


Figure 3: Conversational flow in the form

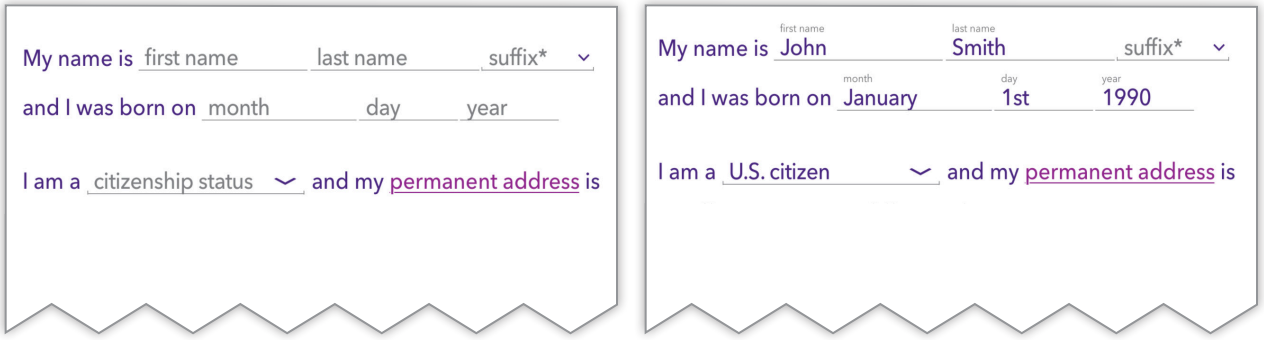


Figure 4: Form field labels OnLoad and OnClick

Application Architecture (cont'd)

Optional Input Notation
Generally all information is required in the form fields with the exception of fields that have been denoted with an asterisk in the label (*). The asterisk will remain in the label even when it has moved to the top left corner when and if the user has input a value for that optional field (*see figure 5*.)

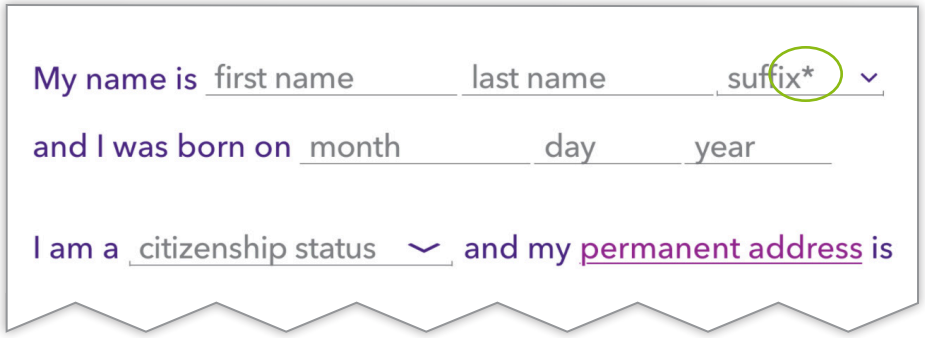


Figure 5: Suffix is optional and is denoted with an asterisk.

2. Application States

Overview
The system has several "modes" that are utilized:

- First time user, borrower
- First time user, cosigner
- Returning user, borrower
- Returning user, cosigner

The screens and forms presented to the user will be identical between the first time and return mode -- relative to the user type -- with the exception of providing pre-filled information for the latter that the user can update when they return to their incomplete application. The exceptions to this will be noted in the wireframes.

Flow
As mentioned earlier in the document, the application presents itself in a linear fashion. A user may go backwards at anytime to return to an area previously completed; however, they cannot skip forward. In order to go backwards, the user may use either browser controls OR the breadcrumb (*see Figures 6a and 6b*.)

When a user returns to an incomplete application, they should be returned to the screen in which they exited the application. Again, they may go backwards to review previously submitted information and make updates; however, they must still follow the application flow in order to move forward.

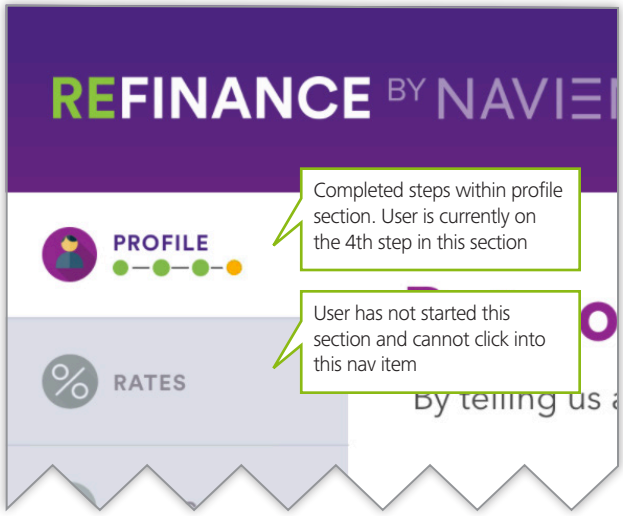


Figure 6a: In-app breadcrumb/application controls, desktop

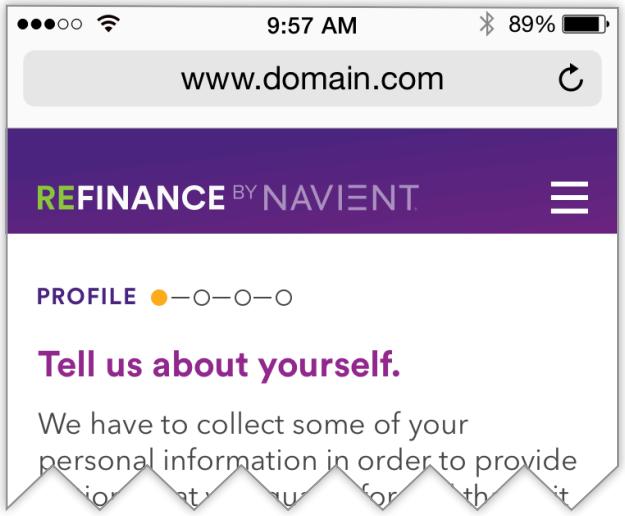


Figure 6b: In-app breadcrumb/application controls, mobile

The application is essentially a one-time system. That is, once a borrower has completed their application, their engagement with the system is finished. They will not come back to the application to check on status or take any further action. Any further updates or continued interaction with Navient will come via email or phone. Cosigners are the same: once a cosigner has completed their portion of a borrower's potential loan consolidation, they will no longer utilize the system for the loan they cosigned.

Application Architecture (cont'd)

3. Error Handling and Messaging

Error handling

During the course of application use, the system will either be in the default state (in use) or the application will be in an error state. Error states should be handled as follows:

1. User is alerted that there is an error
2. User is shown the error in line
3. User cannot proceed forward until the error is fixed

When the system isn't in an error state, by default it's in use. Errors should always be handled in the moment and provide the user enough information to correct the error. The system should gracefully handle errors and show the errors in a user friendly format. *(See the wireframes for error messaging and detailed layout.)*

Errors will occur in one of two ways:

- On screen prior to the user submitting
- Upon submission

If the system determines that the information isn't in a valid format, the system should display the error on the screen before submission. Examples of this would be text input on numerical fields, incorrect string lengths, or missing information on a required field. For more complex errors that require data validation on the back end, the user should remain on the same screen and the error messaging at the top of the screen (item #1 shown in Figure 7) should explain why the data was invalid on submission.

Messaging

The system will include success inline messaging at several key points in the application which are dependent on the user entering information correctly. Figure 8 shows the instances where this messaging is used to either alert the user to a successful data entry or a data entry that is in process or pending.

Exact areas where inline messaging occurs are denoted in the full set of wireframes.

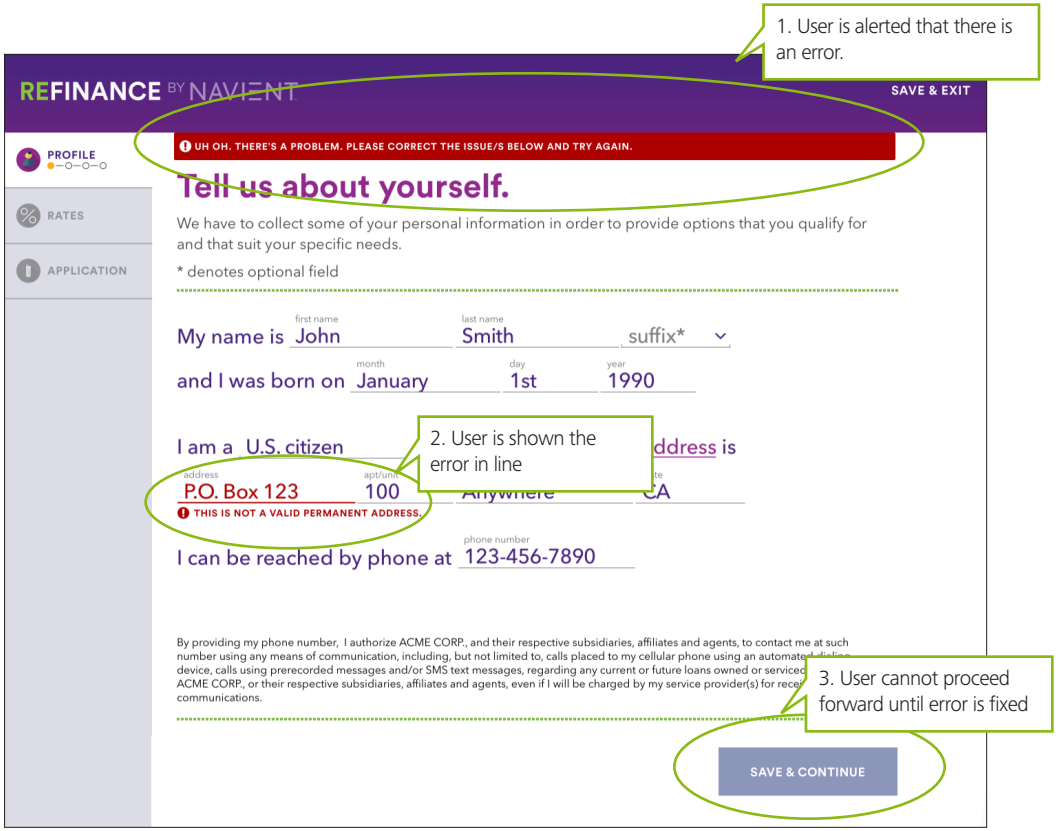


Figure 7: Error state

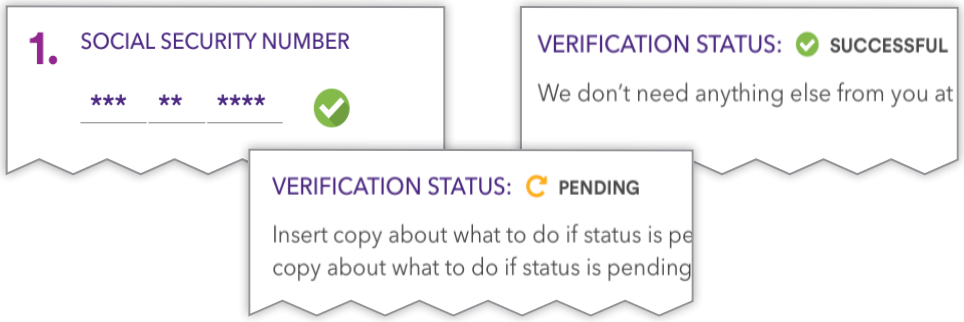


Figure 8: Various inline messaging states

Application Architecture (cont'd)

4. Modals

Use

Modals are used for viewing required disclosures and documents in the application. By their nature, modals are disruptive and force the user to stop their workflow. As such, they should only used for required activities involving legal documentation.

Function

Both mobile and desktop will utilize modals in the same manner.

1. The modal will be opened via button telling the user what is required when they click.
2. Upon clicking, the user will see the disclosure in a scrollable pane within the modal.
3. The user must scroll to the bottom of this modal pane to activate the acceptance button.
4. The button become active and the user can now progress forward in the main application.

This ensures the user actually attempts to read the content in the disclosure statement(s) and is a more legally binding method for reqesting online user review.

Mobile considerations

Mobile should function in the same manner as desktop. The only diference will be that mobile will utilize controls already present in the mobile browser for the **print** and **download** functions. That is, the two icons / links for these functions as shown in Figure 9, steps 2 and 3, will not be displayed. *See wireframes and visual comps for how these screens will look on mobile.*

Note: All disclosures that are more than 100 words in length should be placed into a modal.

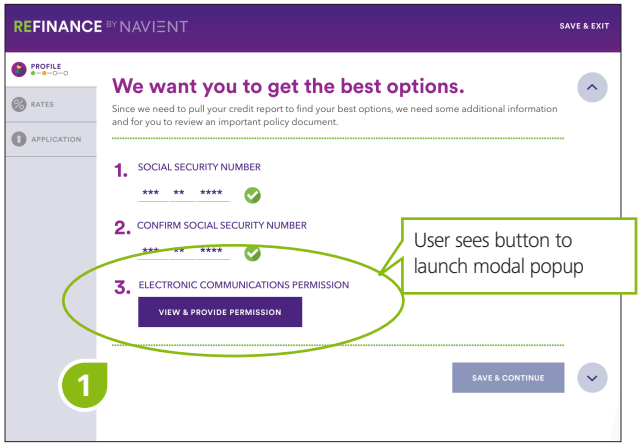


Figure 9a: Modal flow, step 1

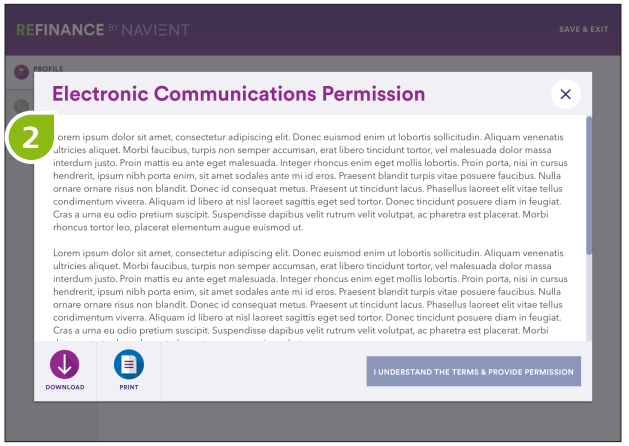


Figure 9b: Modal flow, step 2

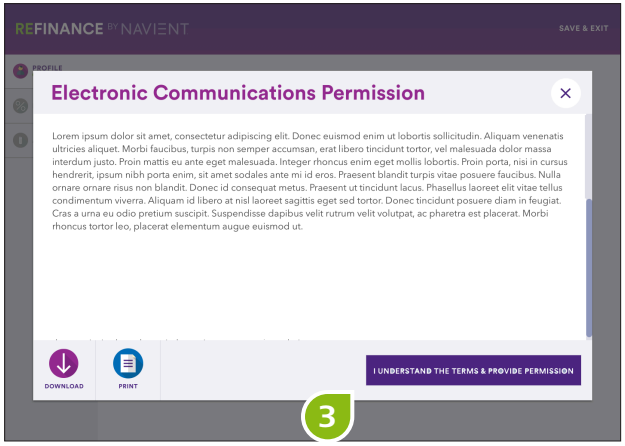


Figure 9c: Modal flow, step 3

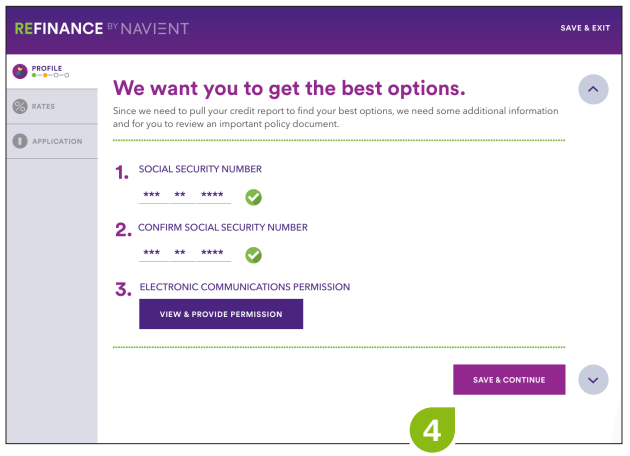


Figure 9d: Modal flow, step 4

TRANSITIONS

Analysis of how the system behaves between and within screens, including user movement through the application.

This chapter contains:

- Overview
- Transitions between form sections
- Transitions between form steps
- Transitions between form segments

Transitions

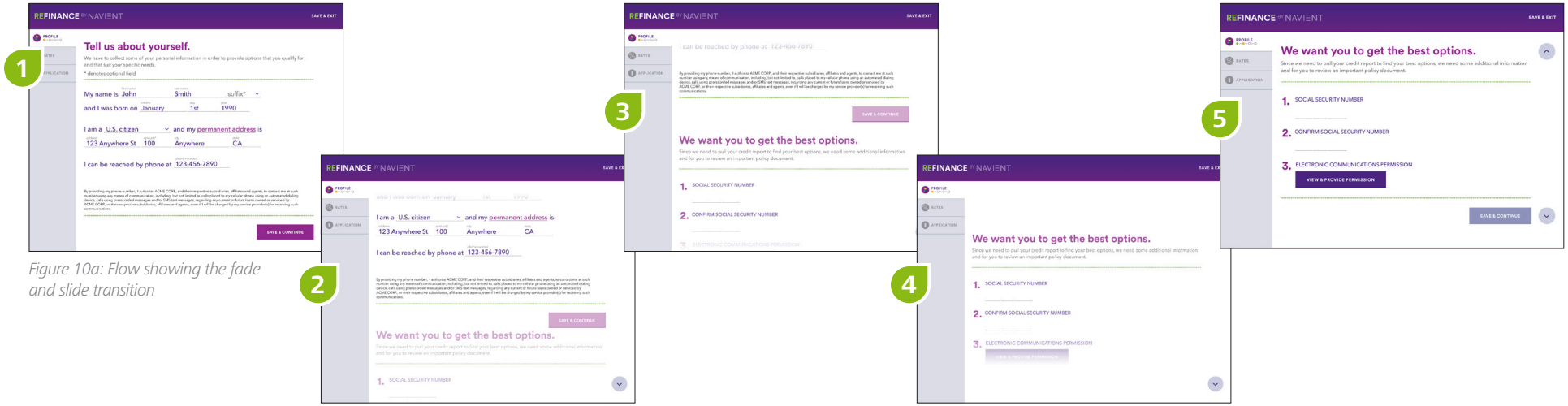


Figure 10a: Flow showing the fade and slide transition

1. Overview

The system will utilize a set of transitions across form factors that are tied to the component type. These components are outlined in the second section of this document, Form Architecture under Application Architecture. A description of transitions will be provided in this section. Developers can also view the videos to see animated examples.

2. Transitions between form sections

The system contains 3 form sections, in the following order:

1. Profile
2. Rates
3. Application

Borrowers will need to complete all 3, cosigners will only need to complete sections 1 and 3. ***Note: Section 2 (Rates) will be hidden from cosigner's navigation and will be inaccessible to those users.***

Navigation transition

When a user has completed the last segment of the last step within a section:

1. The sidenav updates by activating the next section (going from a grayed out to a full color view)
2. The completed section stays full color but the progress indicator within that section disappears
3. The section name slides back to its original position

See Figure 10a for a step-by-step guide as well as the complete visual designs.

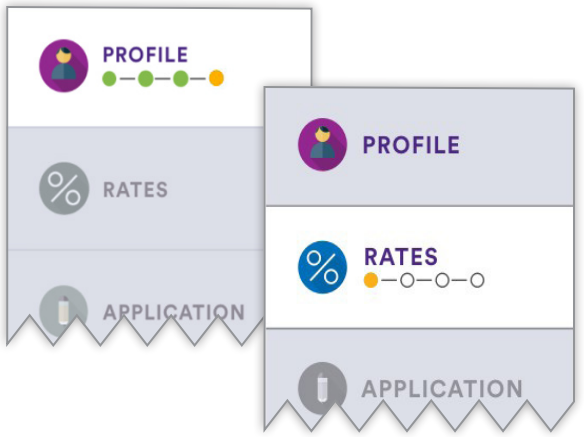


Figure 10b: Navigation transition between step 4 in Profile and step 1 in Rates

Main content transition

The main content area will transition in the same fashion as a step transition which is covered in depth in the next section since a new section is a new step.

Transitions (cont'd)

3. Transitions between form steps

Mobile

1. The completed step and its title and intro text slides upwards, off the screen, fading to 0% opacity as it slides offscreen
2. At the same time, the new step title, intro blurb and first segment slides up from the bottom while simultaneously fading in, from 0% to 100% opacity, and replaces the completed step
3. The onscreen indicator extends to show progress

Desktop

1. The completed step and its title and intro text slides upwards, off the screen, fading to 0% opacity as it slides offscreen
2. At the same time, the new step title, intro blurb and first segment slides up from the bottom while simultaneously fading in, from 0% to 100% opacity, and replaces the completed step

By default, the active step should show as yellow on the side nav progress indicator within the active section. Once a user completes the current step within the application, the yellow step indicator in the open section nav should then turn to green and the next step should show as yellow.

Once the user reaches the last step, the section will transition as laid out in **2. Transitions between form sections.** (shown in figure 10)

4. Transitions between form segments

When a user completes a segment and the next one loads the following should happen:

Mobile

1. The completed segment slides upwards, off the screen
2. The new segment slides up from the bottom at the same time and replaces the completed segment

Desktop

1. The completed segment remains on-screen
2. The new segment slides up just below the completed segment
3. The last segment in a step includes the "Save and Continue" button which is disabled until the user has completed all required information in the segment
4. Upon clicking the "Save and Continue" button, the next step loads as laid out in **3. Transitions between form steps**

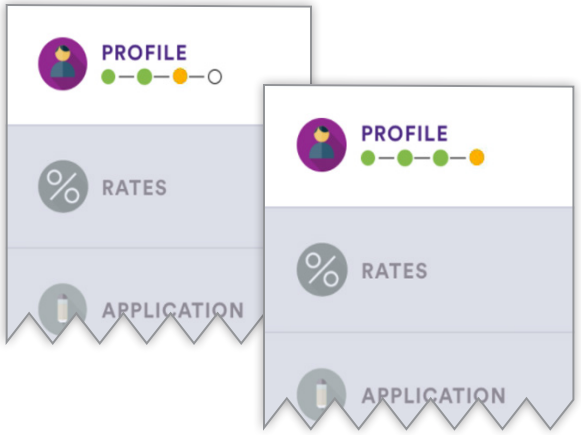


Figure 11a: Navigation transition between steps 3 & 4 in Profile section

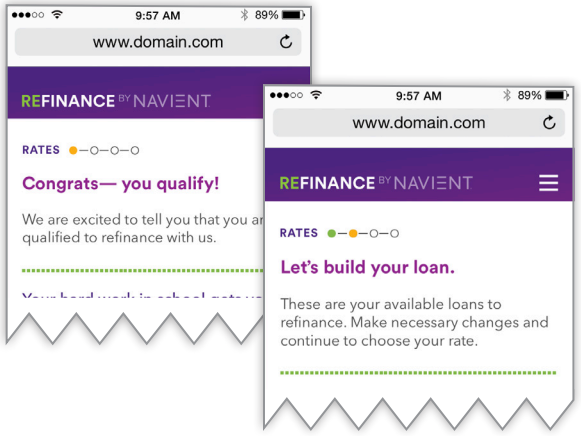


Figure 11b: Navigation transition between steps 1 & 2 in mobile Rates section

INTERACTIONS

Analysis of how the system behaves when the user interacts with specific elements onscreen.

This chapter contains:

- Overview
- General keyboard interactions
- Mobile specific-interactions

Interactions

1. Overview

The system will utilize a set of interactions that allow a user to add or input information for moving forward in the process.

The system uses five main interaction types for adding information.

1. Selecting and entering information into form fields (Interactions vary across desktop and mobile)
2. Adding additional income
3. Adding a cosigner
4. Adding and removing loan amounts
5. Enrolling in autopay

When adding information for items 2 - 4, the interaction is consistently denoted by a plus sign (+) icon. The removal or cancellation of this information is denoted by a minus icon (--) with a related text label.

When adding information into fields or submitting information, any inline or submission error notifications would appear as outlined in section 3.

Adding information on Desktop

On desktop, a field becomes active when it is selected.

Fields may be selected by:

1. User selects field with cursor
2. User <tabs> to advance from one field to the next
3. After a field has been completed, a user can select <ENTER> to advance to the next field

The field type determines how a user may enter information into a selected field.

1. **Free form** allows a user to type in information. *Shown in Figure 12a*.
2. **Type ahead drop down** allows user to type and select from dropdown of options that correspond to what has been entered (includes [graduation] month, year, state, school fields). *Shown in Figure 12b*.
3. **Drop down** allows user to select option from a drop down menu (Includes suffix, citizenship status, degree obtained, GPA range, GPA scale, and income source). *Shown in Figure 12c.*

Adding information on Mobile

Similar to desktop, on mobile, a field becomes active when it is selected. When a field is selected, the cursor appears and inline help text slides to the top of the this field to indicate it is active.

The native keyboard pops up when a free form or type ahead drop down field is selected.

When fields that require only numbers are selected, the numeric keyboard opens. These fields include the following:

- SSN
- Phone number
- Account number
- Editing loan amount fields

Figure 12a

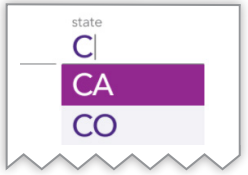


Figure 12b

Figure 12c

Interactions (cont'd)

2. Adding additional income

Use
Borrowers and cosigners will be presented with the opportunity to report additional annual income if necessary. By default, this optional step is displayed in a closed drawer (shown in Figure 11a).

Function
When a user selects to add another income (shown in figure 11b) :

1. Add income drawer slides down to reveal income amount and income source dropdown fields
2. Add income button gets pushed down
3. Option to remove income appears to allow users to remove additional fields and any data entered.
4. If one of the fields is left blank, a submission error would get triggered and prompt the user to complete the fields

When a user selects to remove income:

1. The two additional fields, the data entered, and the remove income button disappear
2. The add income drawer slides up
3. The add income button slides back to original position above

Mobile Considerations
The interaction for adding income should remain consistent across desktop and mobile platforms.

Figure 13a: default state for add income drawer

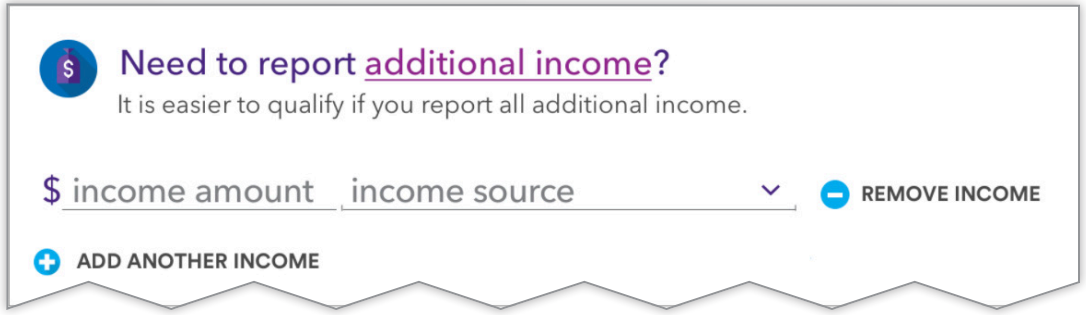


Figure 13b: steps 1-3

Interactions (cont'd)

3. Add a cosigner

Borrowers can invite a cosigner to join their application. For borrowers who are not required to add a cosigner to their application, the user can still activate the drawer which contains fields for inviting a cosigner by clicking the add a cosigner button (show in Figure 14a).

If a borrower's credit decision result requires a cosigner to join the application, the fields to invite a cosigner will be displayed on screen and treated as a separate step in the rates section of the borrower's flow.

When a user selects to add a cosigner (step is optional) as shown in Figure 14b:

1. The drawer slides open to reveal fields that allow borrowers to copy the reference code to their clipboard or enter an email to which the code will get sent (please note: different interaction on mobile device)
2. Add a cosigner button disappears

Notifications
If a borrower enters an email and opts to email the code, a notification with a rotating arrow icon animates while the system generates and sends the invitation email.

Once the system has sent the invitation email, a "sent notification" will appear before fading out after several seconds.

Status
A status will display in place of the instructional text to signify whether a cosigner has started the application. Once a cosigner has successfully completed an application, the instructions and fields for inviting a cosigner will disappear and a success status will appear in their place.

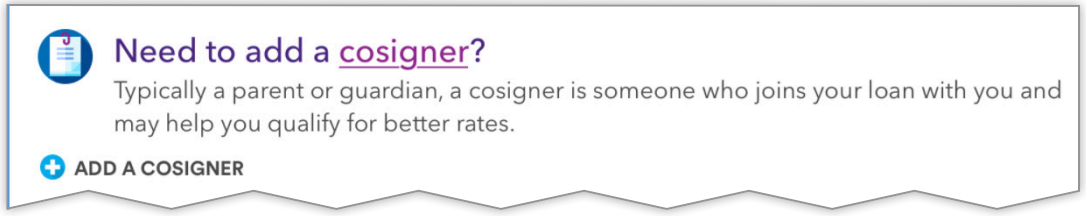


Figure 14a: Step 1

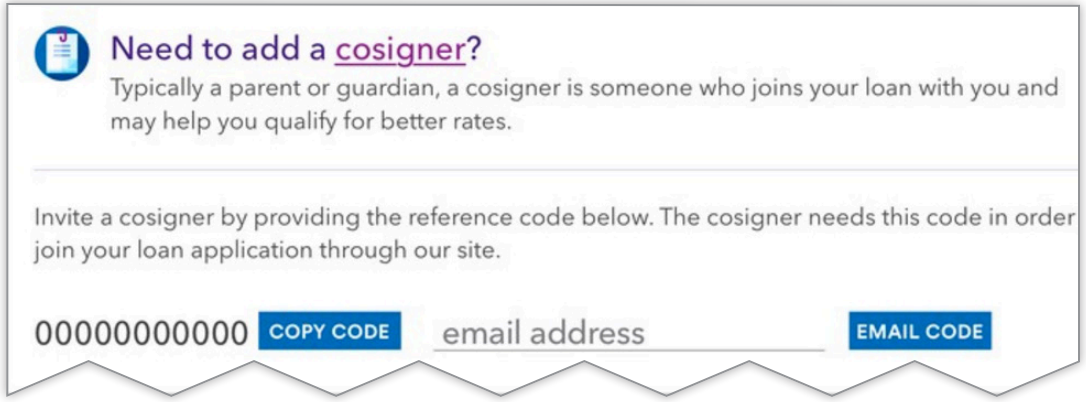


Figure 14b: Step 2

Mobile Considerations
When a user selects to add a cosigner on mobile, the user can select to email the code, which launches the native application for email with the reference code prepopulated or the user select to text the code, which opens the native application for text messaging with the code prepopulated (shown in Figure 14c).

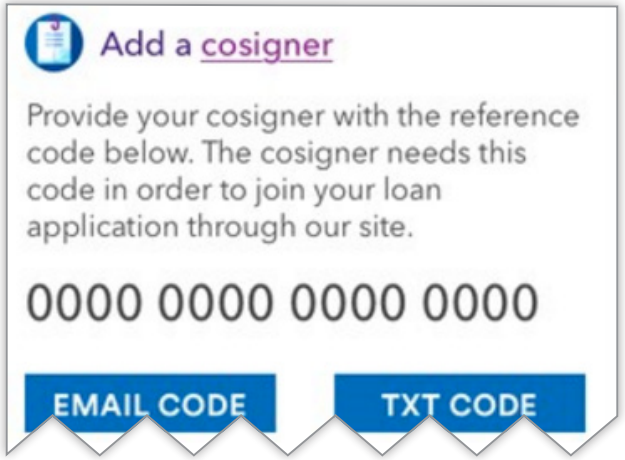


Figure 14c: Mobile "Add Cosigner"

Interactions (cont'd)

Status messagess
When a user selects to add a cosigner on mobile, the user can select to email the code, which launches the native application for email with the reference code prepopulated or the user select to text the code, which opens the native application for text messaging with the code prepopulated (shown in Figure 14c).

Figure 14d-f: Cosigner status messages

4. Add and removing loan amounts

Use
When a borrower is approved and the system pulls loan information for a specific borrower, the borrower can edit the loan amounts to manage the refinance total.

Function
The system will allow the borrower to edit loan amounts in the following ways:

1. Reduce or remove automatically pulled loan amounts
2. Add loan amount manually

Editing Automatically Pulled Loans
When a user selects remove automatically pulled loan:

1. The entire row gets removed from the table
2. The refinance total updates automatically

When a user wants to change the amount of the loan to be Refinanced:

1. The user can select the field with the amount
2. The user enters the new amount
3. If user enters an amount greater than the amount pulled, the field automatically populates with original amount and an inline error appears notifying the user that amount can not exceed original

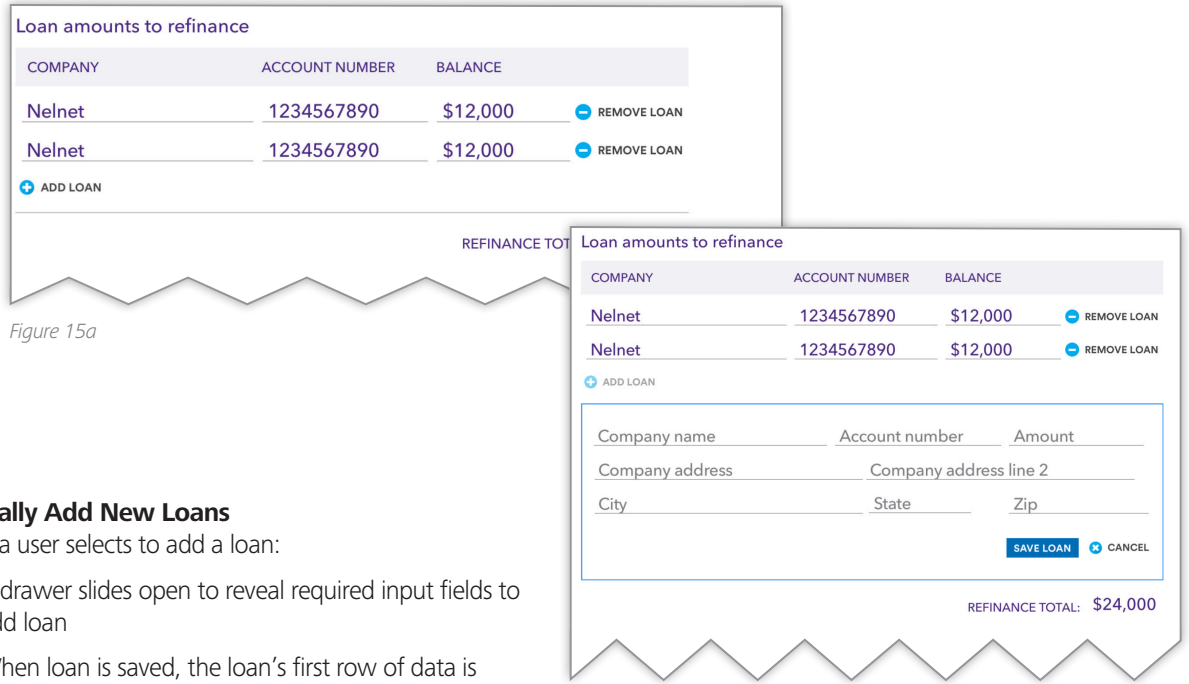


Figure 15a

Figure 15b

Manually Add New Loans
When a user selects to add a loan:

1. A drawer slides open to reveal required input fields to add loan
2. When loan is saved, the loan's first row of data is added to the existing table of loans

Interactions (cont'd)

Notifications
When saving a loan to the system, the saving arrow icon rotates to signify that the system is validating the information. When the system successfully validates the information, the new loan displays.

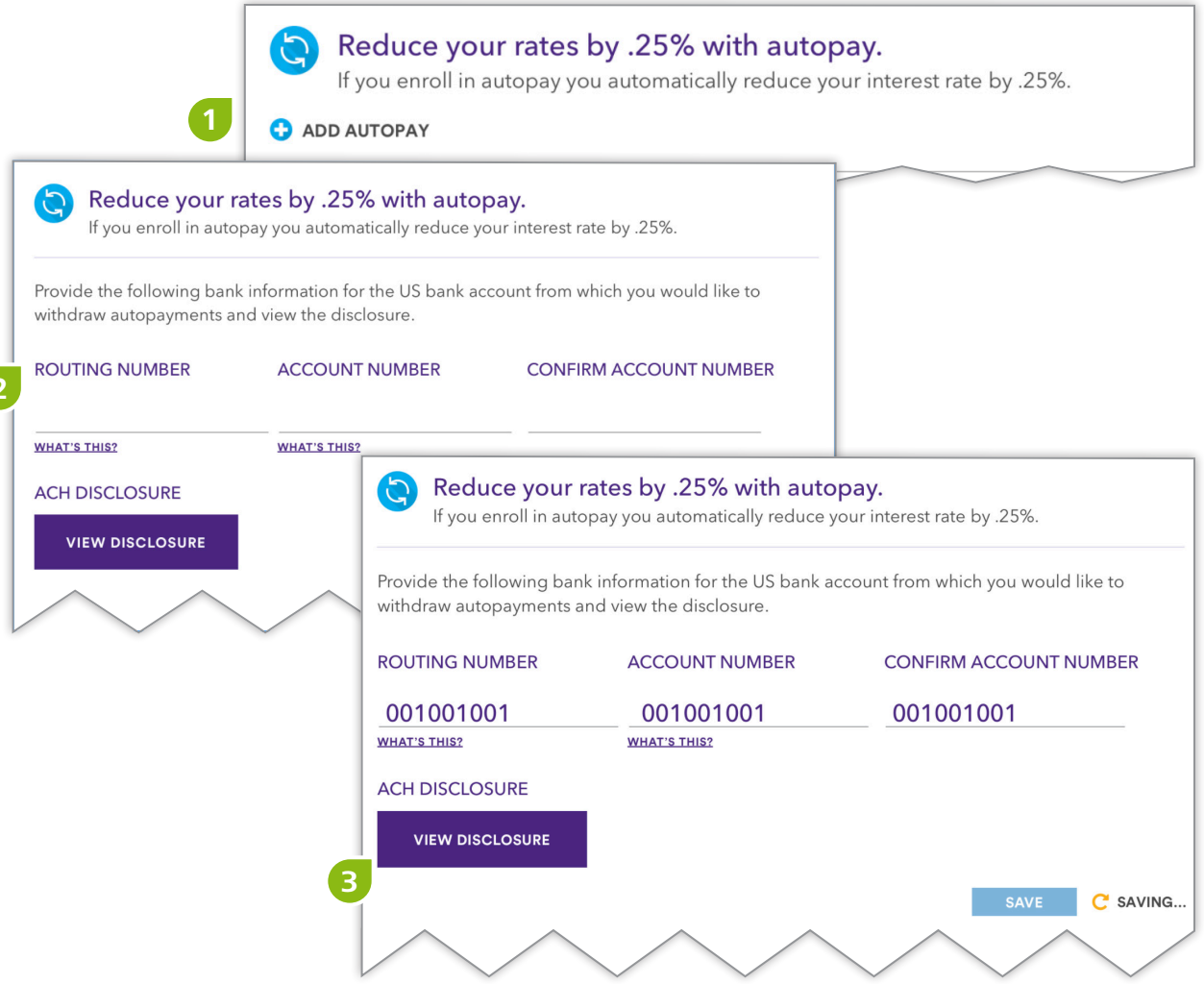
If the system does not successfully validate the information, a submission error should appear, and all invalid fields should be highlighted.

5. Enrolling in Autopay

Use
Once a borrower has signed and electronically submitted their loan application, they are presented with the option to enroll in autopay.

When a user selects to add autopay:

- The user selects the add autopay button and drawer expands to reveal additional fields
- Required fields must be complete and ACH disclosure viewed before the save button becomes active
- Once a user saves the information, the rotating arrow icon labeled "saving" indicates that system is capturing and validating the information, and a saved status displays.



APPENDICES

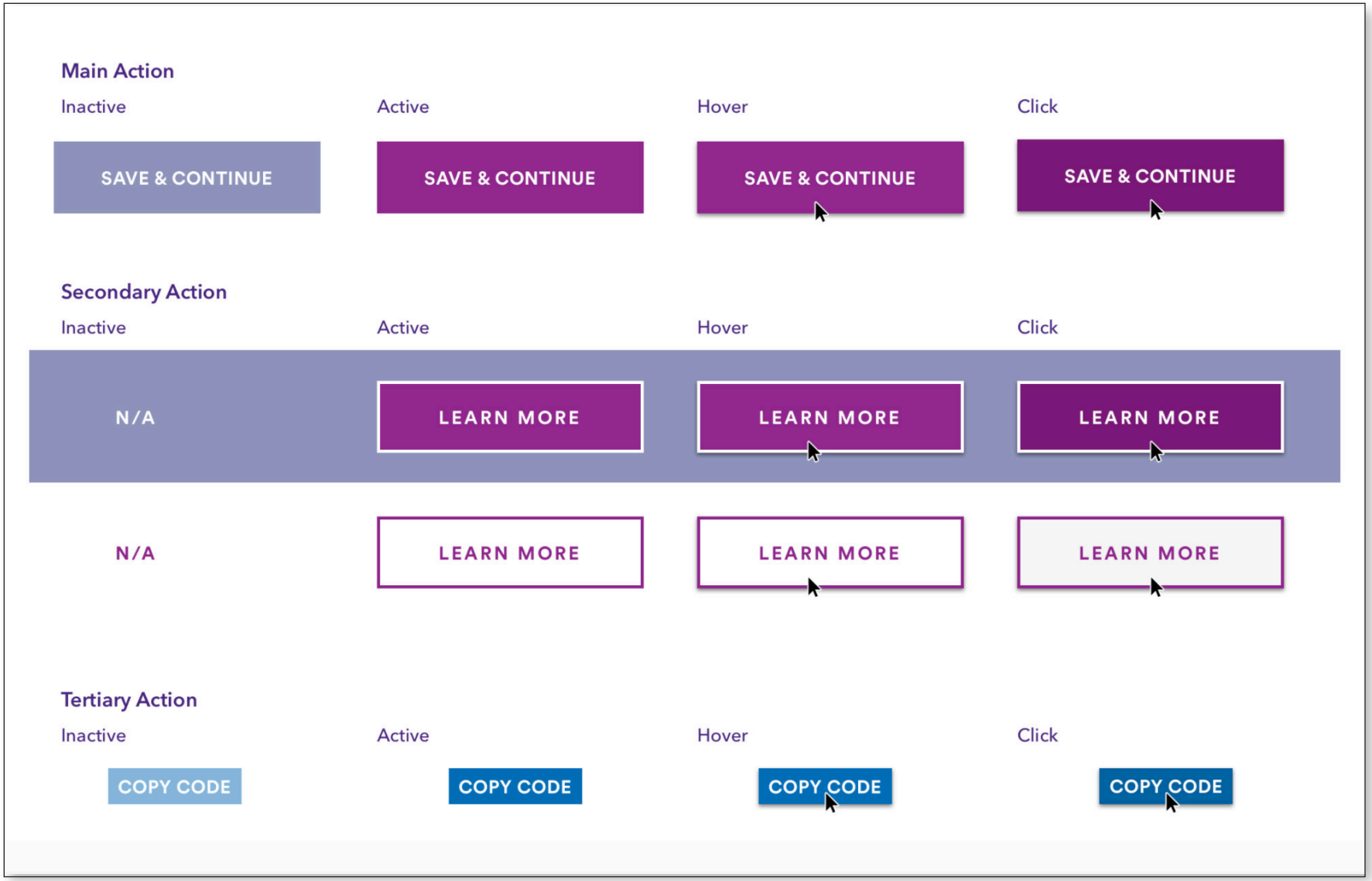
Supporting files and documents for application implementation.

This chapter contains:

- Final visual styles
- Final comps link

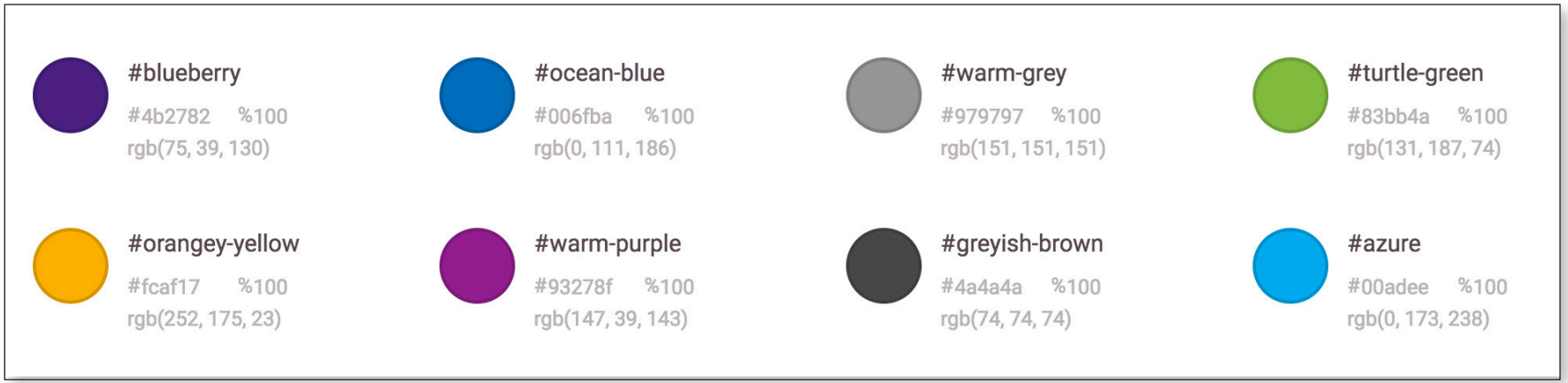
Final visual styles, Desktop

Buttons



Final visual styles, Desktop & Mobile

Colors



Final visual styles, Desktop & Mobile (cont'd)

Text Styles, Desktop

H1	CircularStd, Bold, 48px #warm-purple
H2	CircularStd, Bold, 32px #ffffff Center, Letter spacing: 1px
P madlib	AvenirNext, Medium, 30px #blueberry
Input text empty	AvenirNext, Medium, 30px #steel-grey Letter spacing: -0.5px
Input text filled	AvenirNext, Medium, 30px #blueberry Letter spacing: -0.5px
H3	AvenirNext, Medium, 22px #blueberry
P body	AvenirNext, Regular, 22px #greyish-brown
Button	CircularStd, Bold, 18px #warm-purple Center, Letter spacing: 0.4px
P footer	CircularStd, Bold, 18px #ffffff Center, Letter spacing: 2px
Button	CircularStd, Bold, 18px #39393a
Button	CircularStd, Bold, 18px #ffffff Center, Letter spacing: 0.4px
Button	CircularStd, Bold, 18px #ffffff Center, Letter spacing: 2.3px
H4	CircularStd, Bold, 16px #greyish-brown Letter spacing: 0.9px
H4	CircularStd, Bold, 16px #blueberry Letter spacing: 0.9px
P small	AvenirNext, Regular, 16px #000000 Line height: 20px
Error message, top	CircularStd, Bold, 14px #ffffff Letter spacing: 0.8px
Error message, in line	CircularStd, Bold, 14px #af0404 Center, Letter spacing: 0.8px
Tooltip heading	CircularStd, Bold, 14px #blueberry Center, Letter spacing: 0.8px
Label	AvenirNext, Regular, 14px #717376

Text Styles, Mobile

H2	CircularStd, Bold, 22px #ffffff Center, Letter spacing: 0.7px
H1	CircularStd, Bold, 22px #warm-purple Center
H1	CircularStd, Bold, 22px #warm-purple
Input text	AvenirNext, Medium, 20px #808285 Letter spacing: -0.3px
input text filled	AvenirNext, Medium, 20px #blueberry Letter spacing: -0.3px
P madlib	AvenirNext, Medium, 20px #blueberry
Button	CircularStd, Bold, 18px #39393a
H3	AvenirNext, Medium, 18px #blueberry
P body	AvenirNext, Regular, 18px #greyish-brown Line height: 22px
P body	AvenirNext, Regular, 18px #greyish-brown Center, Line height: 22px
Button	CircularStd, Bold, 16px #ffffff Center, Letter spacing: 2px
Button	CircularStd, Bold, 16px #ffffff Center, Letter spacing: 0.4px
H4	CircularStd, Bold, 16px #blueberry Letter spacing: 0.9px
P Small	AvenirNext, Regular, 16px #000000 Line height: 20px
H5	CircularStd, Bold, 14px #blueberry Letter spacing: 0.8px
H5	CircularStd, Bold, 14px #blueberry Center, Letter spacing: 0.8px
P small	AvenirNext, Regular, 12px #greyish-brown Line height: 16px

Final visual styles

Final visual designs can be found in the following locations:

Desktop
https://app.zeplin.io/project.html#pid=58c1bb373e94bbd5c95e6b6e

Mobile
https://app.zeplin.io/project.html#pid=58c1b032b55bb2a0afa5876d