User and Developer Manual

Form Buster: Web-Based Registration, Approval, and Tracking

Team Members

Daniel Acosta (dacosta2022@my.fit.edu)
Christopher Demuro (cdemuro2022@my.fit.edu)
Alex Merino (amerino2022@my.fit.edu)
Luka Miodrag Starcevic (lstarcevic2022@my.fit.edu)

Faculty Advisor

Phillip Bernhard (pbernhar@fit.edu)

Table of Contents

1. Introduction

The purpose of this document is to provide instructions and pertinent information for both users and future developers of the Form Buster system. Form Buster's goal is to be a centralized form filling, tracking, and editing system for Florida Institute of Technology. The current system requires emailing pdf files to advisors or registration staff and then not getting any feedback on the progress of the form until its completion. Form Buster aims to solve all the problems that are wrong with the current system and improve on the parts that already work with the system.

2. User Manual

This section of the document is dedicated to providing insight to users on how the web application works and how to use the application for many different tasks. There will be basic tasks that all users can accomplish, then user type specific tasks for students, advisors, and university staff.

2.1. Student/General User Functions

All general pages and functions can be used by any user type, but the student users are restricted to only these general functions. The Advisor and Administrator user types have access to all of these functions plus a few more which will be explained in more detail in future sections.

2.1.1. Authentication

When a user first uses the website, they will only be able to view the landing page, which describes the functionality and overview of the website. In the top right of the page there are two buttons; *"Sign in"* and *"Register"*. The purpose of the buttons are to send the user to the desired authentication page.

2.1.1.1. Sign In

The sign in page is the key to accessing the entire system for any user. There are two input fields which ask for users to input user email and password. If the email and password combination input is incorrect then the web application will show an alert and tell the user which input is affected. The *"Submit"* button allows for the submission of the email and password to authenticate. If the user does not have an account there is a link to the ***Registration*** page.

2.1.1.2. Registration

On the ***Registration*** page, when it first loads, there are three options for users to choose which user type fits them best. After selecting the user type they believe is correct, they are then asked to fill out the personal information required to complete an account. Some information differs depending on the user type, but most information is uniform across all types. If the wrong user type happens to be selected there is a *"Return to Previous Page"* button to choose a different user type.

2.1.2. Dashboard

After signing in, the ***Dashboard*** is the first page that is visible to users. The dashboard contains the simplified views of active forms. On each

form there are four main parts. Going from left to right, the form type (FERPA, Class Registration, etc.) and the date and time the form was created are on the left. Then there are a certain number of circles signifying the amount of signatures required for the completion of the form. The circles can be four colors; green (signed), red (rejected), light gray (unsigned), and dark gray (not applicable). When the mouse is hovered over a circle, the user that the signature belongs to, along with the date the signature occurred will be displayed. The next component, if visible, is the notifier which warns the user that they have not signed the form after 2 days. This component is only available if the two day limit is passed. The last component, displayed as three dots, allows users to view more information about the form or sign the form if they still need to.

2.1.3. Signature Graph

When viewing more information about a form from the dashboard, the user will get sent to a page which has more detailed information of the form, including the ***Signature Graph***. The ***Signature Graph*** is a novel feature to form tracking where the user can see every signature required in a flow chart like structure. The graph displays user type information on default and the current signature status.When a node (circle) in the graph is clicked, a popup to the right of the graph will display more information about the user that the signature belongs to. There are different line types which connect two signatures, with the explanations of each available on the left side of the page in a legend.

Above the ***Signature Graph*** is information about the form itself, (name and date created) along with two links to different parts of the website. The link titled *"View Form Here"* will send the user to the ***Form View*** page. The

2.1.4. Form View

The ***Form View*** page which displays the form in its entirety. A form display consists of the form template, the form data that was filled out, and an input for all signatures that are required for the form. A form template consists of field labels (name, ID, birthdate, etc.) and the corresponding input types (text, checkbox, date, etc.). Since the form has been initialized there is related data that has been filled out, so that data is placed into the matching inputs. Then below the form are signature boxes, which are filled if the signature has been signed or empty if the user has not signed their designated signature. All of these components combined create the ***Form View***.

2.1.5. Menus

On each page, after signing into the application, there is a gray bar on the top which has the user's name which can be clicked. When clicked, the user is sent to the ***Account*** page. This is the only current functionality of the top bar as there are no other general functions that can be placed there. The other menu that is available on each page is the ***Side Menu***. By default, the side menu is closed and is represented by the red box with

three lines (sometimes referred to as a hamburger menu). When clicked, the ***Side Menu*** will open and display page options. The page options for all users include the ***Dashboard, Inbox, Start a Form,*** and ***Settings***. When any of these are clicked in the menu, the application will take you to the corresponding page.

2.1.6. Inbox

The ***Inbox*** page provides a central location for all notifications that are sent to a user. One might receive a notification if:

- All signatures for a form have been signed
- A user rejects a form
- A form is paused/put on hold
- The user can now sign a form

There will be a specific message attached to each type of notification in the ***Inbox*** along with a small simplified status in the top right of each notification. With the ***Inbox*** a user can do three actions with notifications; *Mark All Read*, *Refresh*, and *Delete All*. When any of the buttons are clicked, the described action will occur to the notifications.

2.1.7. Start a Form

Starting a form consists of two parts; the ***Form List*** and the ***Form Initializor***. When clicking on *"Start a Form"* from the ***Side Menu*** the website goes to the ***Form List*** which displays all possible forms that can be filled out. When selecting any form, the user will be taken to the ***Form Initializor*** which contains the current form template selected.
Another novel feature to form registration is the technology of autofill. The information that the autofill uses comes solely from the data submitted during account creation. The website does not gather or track any other information than that given explicitly by the user. Using that data, the input fields that require the account information are automatically filled in, saving time for users. The date is also automatically filled in and cannot be changed. A user then must fill out all the other required information and type their signature themselves before clicking *"Submit"* which submits the form and then sends the user back to the ***Dashboard***.

2.1.8. Form Signature

The ***Form Signature*** page can be accessed from many different pages on the website, but it is also our most secure page of the website. Users only see a link to the ***Form Signature*** if they are able to currently sign the form. If a user has already signed a form they do not have access to ***Form Signature***.
On the ***Form Signature*** page the ***Form View*** is presented as described in ***2.1.4***. The only difference is that instead of a list of signatures, there are two buttons to click at the bottom of the page; *"Sign"* and *"Decline"*. When clicking on *"Sign"* a signature input will appear at the bottom of the page, allowing for the user to sign. When the *"Decline"* button is clicked a signature input will still be displayed but a text box will also appear requiring the user to write a comment on why they are declining the form.

If the inputs are not filled out, the signature will not submit, but once they are filled out a user can hit the *"Submit"* button and submit their signature.

2.1.9. Account

The ***Account*** page displays account information, which is given by the user when creating their account. There is an option to show their password, which will allow the user to see their password. On the left side of the account information there is a small sidebar which allows the user to switch between the ***Account*** and ***Settings*** pages.

2.1.10. Settings

The ***Settings*** page displays the different settings that can be modified by users. At the current implementation of the website the only modifiable settings is whether or not the user would like to receive Email Notifications. Email notifications are actually not currently implemented so the button does nothing substantial when clicked.

2.2. Advisor User Functions

2.2.1.

2.3. Administrator User Functions

2.3.1. Dashboard

2.3.1.1. Deleting a Form

2.3.1.2. Pausing a Form

2.3.2. Form Editor

3. Frontend Developer Manual

4. Backend Developer Manual

4.1. User Authentication

4.1.1. Component Location

The components that handle user authentication are located in the ***/backend/controllers/userFunctions.jsx*** file. This file contains two main functions. The ***register*** function creates a new user account and stores it in the database. The ***logIn*** function locates an existing user account (if present) in the database and authenticates the session.

4.1.2. User Registration

When a new user attempts to register, the ***register*** function is called. It first checks to see if a user with the same ID number has already been registered. If so, the new registration is rejected. If not, the new user account is created and saved in the database.

4.1.3. User Authentication

When a user attempts to log in to their account, the ***logIn*** function is called. It first attempts to locate an account associated with the given email address in the database. If this lookup fails, it returns an error

message stating “Email is not registered.” Otherwise, it attempts to compare the given password hash to the one stored in the database for the given user account. If they do not match, an error message is returned stating “Incorrect Password.” Otherwise, the user is then signed in to their account.

4.1.4. Developer Recommendation

It is the recommendation of the developers that the user authentication and registration components be replaced with the institutional Single Sign-On authentication system in any production deployment of this system. Not only will this increase the security of user accounts, but it will make for a smoother experience for student and faculty users alike, and will reduce the possibility of fraudulent user accounts being registered by unauthorized users.

4.2. Database Schema

4.2.1. Form Template Format

Form templates are stored in the database as strings of HTML that define the appearance of the form. Input fields must be located in a div class entitled “custom” in order to be picked up by the saving function. The Template Editor takes care of this automatically, but this rule must be followed when creating templates by hand or using external software. Form templates each have their own unique identifiers given in the “id” field. The schema definition is located in ***/backend/schemas/FormTemplate.js***.

4.2.2. Active Form Format

Active forms, like form templates, are identified by a unique identifier given in the “id” field. Active forms store the identifier of the form template they are associated with in the “formType” field along with the form status, creation date, comments and feedback, and signatory information. The schema definition is located in ***/backend/schemas/CurrentForm.js***.

4.3. Form Tracking

The signature graph is generated from the signatory information stored with the active form. Inbox notifications are generated when signatures are added to forms by signatory parties. There are methods to query the database for forms that require attention from a given user. These methods can be found in ***/backend/controllers/CurrentFormFunctions.js*** and ***/backend/controllers/inboxFunctions.js***.