

**SOFT 327**  
**Software Quality Assurance**  
**A2: Design Document**

November 8th 2020

Eric Fillion: 20072951  
Cooper Harasyn: 20002241  
Jolene Lammers: 20029503  
Tingzhou Jia: 20130800

## Layout

The Design Document consists of six other sections: Common, Login, Register, Homepage, Logout and Page Not Found. The Common section describes functions and other code units that are used throughout the code base with two tables. The first table describes the inputs and outputs for each function, while the second table gives a short description for the function. The other five sections of the report contain similar tables, but for functions that are exclusive to their given category. These sections also contain a figure that shows all of the functions and code units that are involved for their category.

## Common

Table 1 shows the inputs and outputs for each function that is shared by multiple sections. For example, the `validate_login_format` function is used by both the login and register section. Then, Table 2 gives a description for each shared function.

Number	Name	Inputs	Outputs
1	<code>validate_login_format</code>	The email and password as strings	True if email and password are valid, otherwise False.
2	<code>__validate_email_format</code>	An email address in the form of a string	True if the email's format meets standards, otherwise False
3	<code>__validate_password_format</code>	An password in the form of a string	True if the password's format meets standards, otherwise False
4	<code>__contains_special_char</code>	An password in the form of a string	True if the password contains a special character, otherwise False
5	<code>authenticate</code> [function decorator]	Any function	The given function wrapped with an authentication check
6	<code>app.route</code> [function decorator] [Flask]	Any function, and a routing specification. Optionally, a list of supported HTTP methods.	The given function, set up in Flask to handle the given route
7	<code>render_template</code> [Jinja2]	Filename of template, and keyword arguments for filling template.	The rendered template, ready to be sent to the user

Table 1 - Input/Output Listing of Common Functions

Number	Name	Explanation
1	validate_login_format:	A function that determines if the email and password combination follows proper formatting guidelines.
2	__validate_email_format	Only called from validate_login_format to determine if an email address follows proper guidelines.
3	__validate_password_format	Only called from validate_login_format to determine if a password follows proper guidelines.
4	__contains_special_char	Only called from __validate_password_format to verify that the password contains a special character, such as "&".
5	authenticate [function decorator]	This function decorator wraps the decorated function in an authentication check. If the user has a valid log-in session, then the decorated function will execute; otherwise, a redirect to the login page will be returned.
6	app.route [function decorator] [Flask]	This function decorator marks a given function in Flask as handling an HTTP route, meaning that when there is a request for a given page, the function decorated with the corresponding app.route will be called to generate the output.
7	render_template [Jinja2]	This function uses the Jinja2 templating engine to fill in a given template. We use it to render information into our HTML pages that we serve to the user.

Table 2 - Descriptions of Common Functions

## Login

Like many e-commerce applications, this application contains a login page. The login page allows users to enter in an email address and a password. If the user enters a correct combination, then after pressing submit, they are directed to the homepage. Table 3 shows the inputs and outputs for all of the functions exclusive to the login section. Then, Table 4 gives a short description for each function. Finally, Figure 1 breaks the login system into a tree of functions, to demonstrate which functions are called by other functions.

Number	Name	Inputs	Outputs
1	login_get	NA	An HTML page that displays the login page
2	login_post	NA	If successful, goes to the homepage, otherwise returns the login page with a message.
3	login_user	An email and password	A user if the email and password are valid. Otherwise none
4	get_user	An email address as a string	A user that is identified by the email.

Table 3 - Input/Output Listing of Login Functions

Number	Name	Explanation
1	login_get	Called when a GET request is made to /login. Fetches an HTML page.
2	login_post	Called when a POST request is made to /login when the submit button is pressed within the login page. Determines if the user's input is valid, and if so goes to the homepage. Otherwise, rerenders the login page and displays an error message.
5	login_user	Called when a login page post request is made and if the email and password combination follow proper guidelines. Then, if the inputs combination passes authentication, a user is returned.
6	get_user	Called from login_user to fetch a user with a given email address.

Table 4 - Descriptions of Login Functions

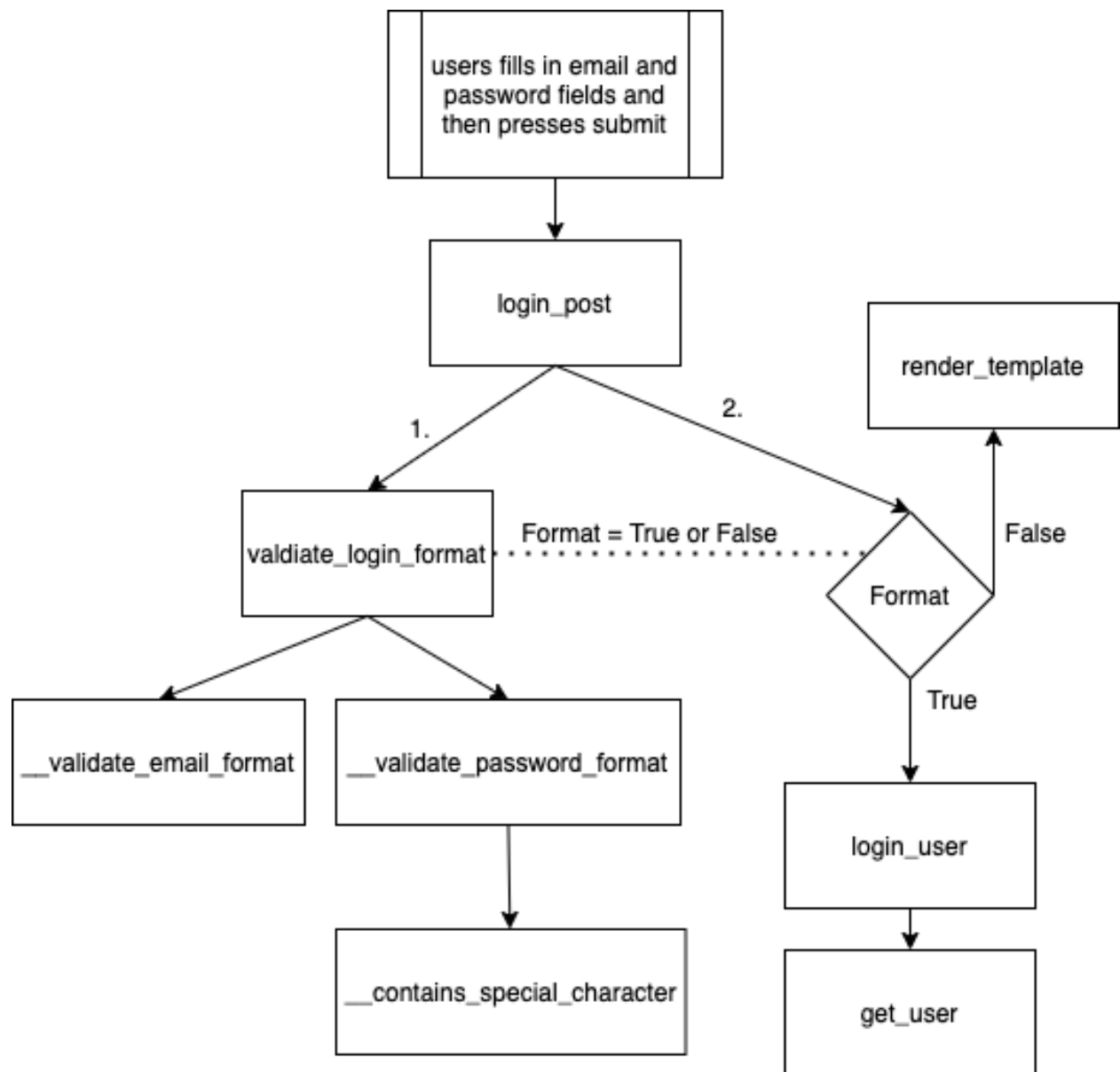


Figure 1 - Function Call Diagram for Login System

## Register

The register page allows users to create an account on our website by using email and password. If the user enters a valid email, name, password and password2, then after pressing submit, a new user will be created in the website's backend database. Table 5 shows the inputs and outputs for all of the functions used exclusively in the registration section. Then, Table 5 gives a short description for each function. Finally, Figure 2 breaks the register system into a tree of functions, to demonstrate which functions are called by other functions.

Number	Name	Inputs	Outputs
1	validate_name_format	The name as strings	True If names are valid, otherwise False.
2	register_get	NA	An HTML page that displays the register page
3	register_post	NA	If successful, goes to the login page, otherwise returns the login page with a message.
4	register_user	An email,name,password, and password2	A user if the email ,name, and password are valid. Otherwise none

Table 5 - Input/Output Listing of Register Functions

Number	Name	Explanation
1	validate_name_format	This function validates the username of registration is valid or not
2	register_get	Called when a GET required is made to /register. Fetches an HTML page.
3	register_post	Called when a POST required is made to /register when the submit button is pressed within the register page. Determines if the user's input is valid, and if so goes to the login page. Otherwise, rerenders the login page and displays an error message.
4	register_user	Called when a register page post request is made and if the email, name, password, and password2 combination follow proper guidelines. Then, if the inputs combination passes authentication, a user is created.

Table 6 - Descriptions of Register Functions

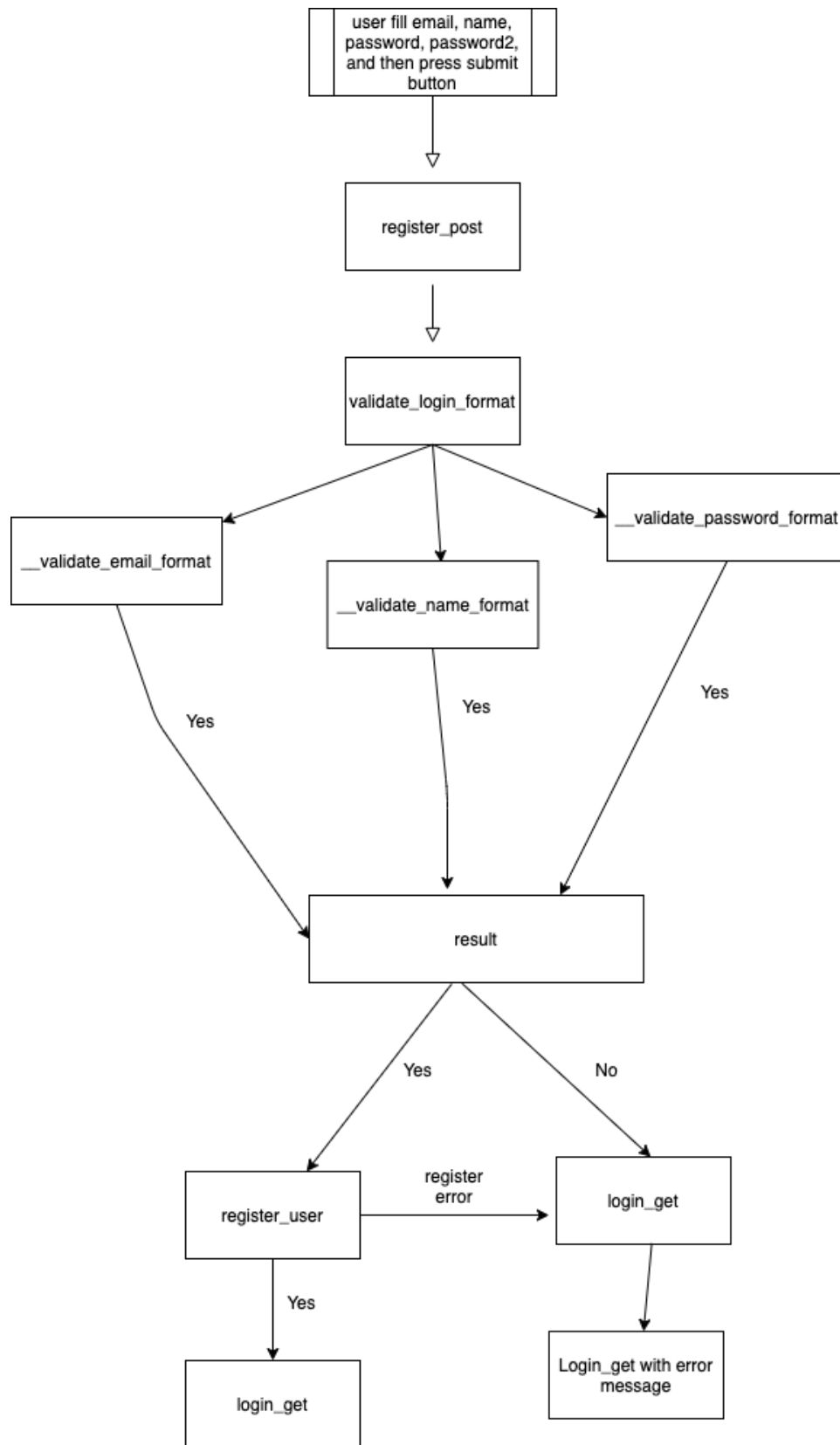


Figure 2 - Function Call Diagram for Register System

## Homepage

Once a user has logged in, they will be brought to the homepage, where they can view their balance, the ticket list, and functionality to buy/sell/update tickets. Table 7 shows the inputs and outputs of the functions used for the homepage section, with Table 8 giving a short description of the functions. Figure 3 shows the tree of function calls when the user requests the homepage, to demonstrate the flow of the system.

Number	Name	Inputs	Outputs
1	profile	The user who is requesting to see their homepage.	The rendered homepage HTML.
2	get_all_tickets	None	All tickets currently in the database.

Table 7 - Input/Output Listing of Homepage Functions

Number	Name	Explanation
1	profile	This function is called to render the homepage for the user. It displays the user's balance, the current valid tickets, and allows for the user to buy/sell/update tickets. It also has a logout button. This function is decorated with <code>@authenticate</code> , which will redirect to the login page if the user is not logged in.
2	get_all_tickets	This backend function returns all tickets currently in the database.

Table 8 - Descriptions of Homepage Functions



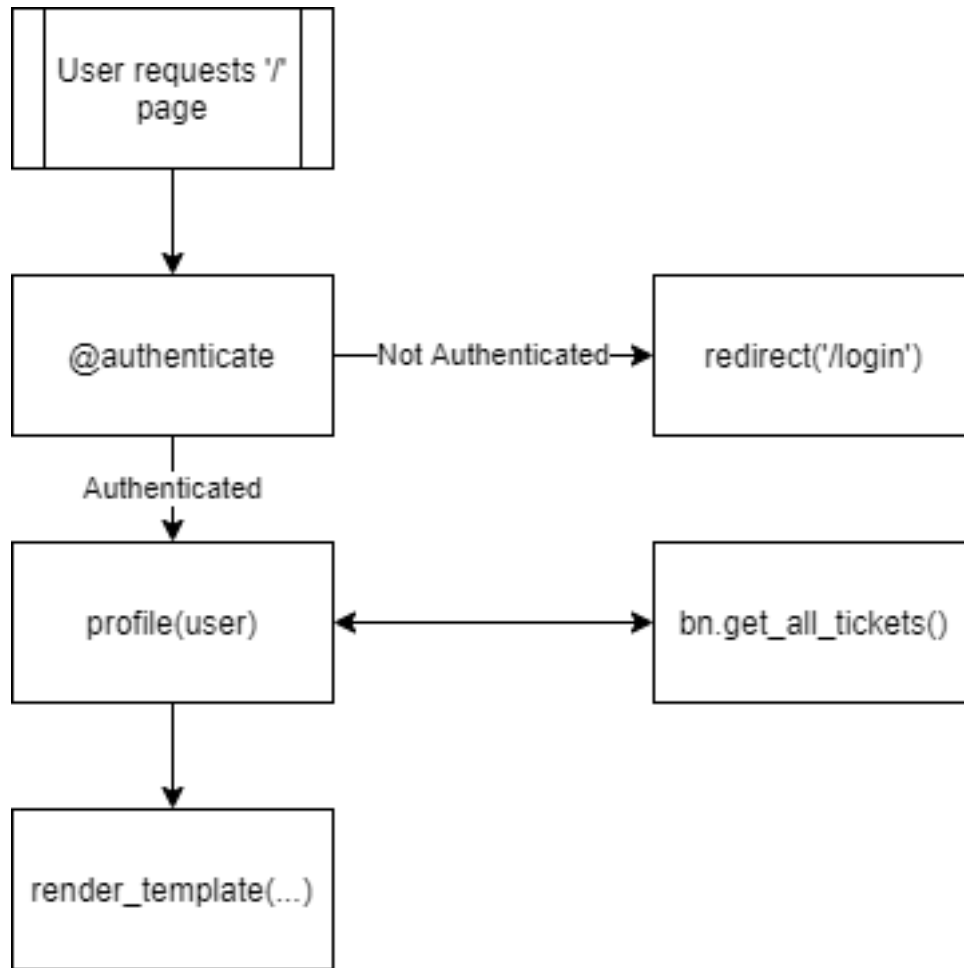


Figure 3 - Function Call Diagram for Homepage System

## Logout

When a user is logged in they can logout to invalidate the current session and redirect to the login page. After logout, the user should not be able to access restricted pages. Table 9 shows the inputs and outputs of the function used for the logout section. Then, Table 10 gives a short description for the function. Finally, Figure 4 breaks the logout system into a tree of functions, to demonstrate the flow of the system.

Number	Name	Inputs	Outputs
1	logout	NA	Redirects to the login page.

Table 9 - Input/Output Listing for Logout Functions

Number	Name	Explanation
1	logout	Called when a request is made to /logout, this happens when the logout button is pressed in the user's homepage. This invalidates the current session and redirects to the users profile which in turn redirects to the login page as the user's session is no longer valid.

Table 10 - Descriptions for Logout Functions

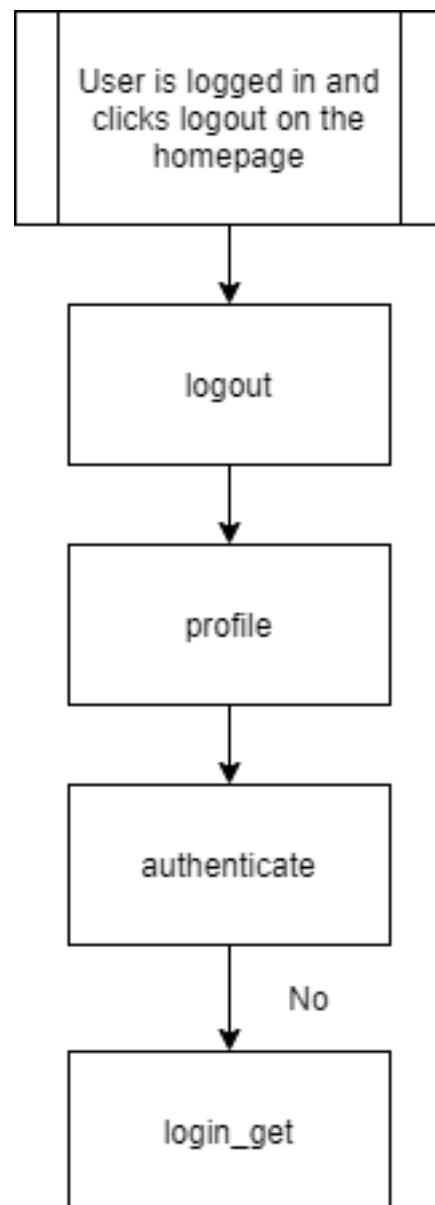


Figure 4 - Function Call Diagram for Logout System

## Page Not Found

For any other requests except the ones listed above, the system should return a 404 error. If the user types in a url where there is no page, a custom 404 not found page will be displayed. Table 11 shows the inputs and outputs for the function used for the 404 page. Then, Table 12 gives a short description for the function. Finally, Figure 5 breaks the 404 page feature into a tree to demonstrate the flow of the feature.

Number	Name	Inputs	Outputs
1	page_not_found	An error number as an integer	An Custom HTML page that displays the 404 page not found page

Table 11 - Input/Output for Page Not Found functions

Number	Name	Explanation
1	page_not_found	If a request is made to a non-existent page then the page is redirected to a 404 page not found page

Table 12 - Description of Page Not Found functions

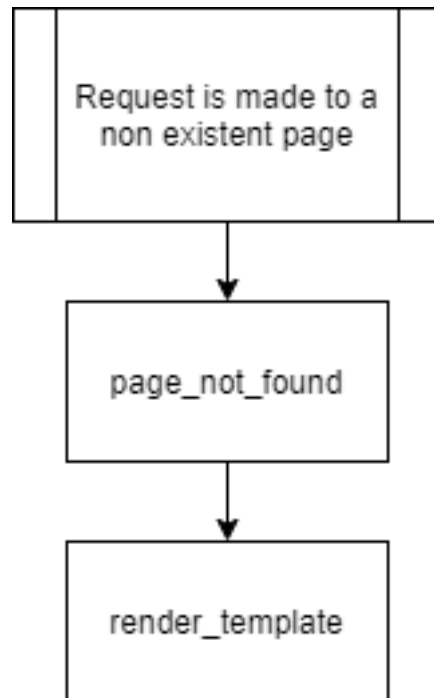


Figure 5 - Function Call Diagram for Page Not Found