SOFT 327 Software Quality Assurance
A4: Design Document

December 1st, 2020

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

Notes:

Github’s servers experienced errors on December 1st during the afternoon, which caused some GitHub Actions Unit Tests to fail.

Often in our code, we have functions that determine if an input is valid. These functions typically returned False when the input is valid and a String to describe the error otherwise. Then, a check within the calling function occurs similar to as described below:

```
error = function()
if not error:
    # No error, complete code
else:
   # Display error string to the user
```

Although it is permitted to return multiple different data types from a single function in Python, the team will consider alternative solutions for similar problems in the future, as some would argue that returning multiple data types is not a best practice. For example, instead, the team may write functions that return a single Tuple, where the first index is to represent if an error occurred, and the second index contains a string to describe the error; if an error did not occur, then the function would return (False, “”), and if an error did occur, then the Tuple may be (True, “The ticket does not exist.”). Then, this Tuple can be used within the calling function to determine if an error occurred using the first index, and then use the second index to print an error message if an error occurred.

Common

Table 1 shows the inputs and outputs for each function that is shared by multiple sections. Then, Table 2 gives a description for each shared function.

Number	Name	Inputs	Outputs
1	check_for_ticket_name_error	A ticket name as a string	Returns any error message as a string if the ticket name is not alphanumeric, if there is an invalid space in the name or the name is too long. If there are no errors it returns false.
2	check_for_ticket_quantity_error	A quantity as a string	Returns any error message as string if the ticket's quantity is not between 10 and 100, else if there are no errors it returns false.
3	check_for_ticket_price_error	A price as a string	Returns any error message as a string if the date is not in the format YYYYMMDD, else if there are no errors it returns false.
4	check_for_ticket_date_error	A data as a string	Checks the tickets name, quantity, price and date to ensure they are the correct format. If they are not in the correct format an error message is returned, else false is returned indicating that there are no errors.
5	flash [Flask]	A message as a string, and a category as a string	NA

Table 1 - Input/Output Listing of Common Functions

Number	Name	Explanation
1	check_for_ticket_name_error	A function that determines if a ticket’s name follows proper formatting guidelines.
2	check_for_ticket_quantity_error	A function that determines if a ticket’s quantity follows proper formatting guidelines.
3	check_for_ticket_price_error	A function that determines if a ticket’s price follows proper formatting guidelines.
4	check_for_ticket_date_error	A function that determines if a ticket’s date follows proper formatting guidelines.
5	flash [Flask]	A Flask function that adds to the list of flashed messages that will be displayed on the next page.

Table 2 - Descriptions of common functions

Sell

The sell functionality is contained within the home page and allows users to list a ticket on the market. It contains four fields: ticket name, quantity, price and expiry date. These fields must conform to strict guidelines, to ensure that the ticket has all of the necessary information in a presentable format. After the user inputs the fields, he or she may press the “sell” button which initiates the selling process. If successful, the added ticket will be shown in the user’s profile. Otherwise, an error message will be displayed. Table 3 shows the inputs and outputs for all of the custom functions used for the sell process, while Table 4 gives an explanation for each function. Finally, Figure 1 shows how all of these functions, interaction with each other and the common functions through a flow diagram.

Number	Name	Inputs	Outputs
1	sell_post	The user object corresponding with the currently logged in user	An HTML page that displays the profile page.
2	check_for_sell_ticket_format_error	A ticket’s: name, quantity, price and expiry date	If there is an error in one of the inputs an error message as a string is returned, else false is returned indicating that there are no errors.
3	sell_ticket	The user selling the ticket, and a ticket’s name, quantity, price and expiry date	Returns a string if an error occurred, else False.

Table 3: Inputs and outputs for sell functions

Number	Name	Explanation
1	sell_post	Called when a POST request is made to /sell. Follows R4’s requirements to complete the selling process.
2	check_for_sell_ticket_format_error	Called to determine if the specified inputs for the ticket conform to the formatting requirements specified in R4.1-4.5
3	sell_ticket	A backend function that updates the database using the ticket’s attributes. When called, it attempts to add the ticket to the database. This can fail if a ticket with the same name already exists, in which case an error will be reported. Otherwise it will return False to represent success.

Table 4: A description for each sell function

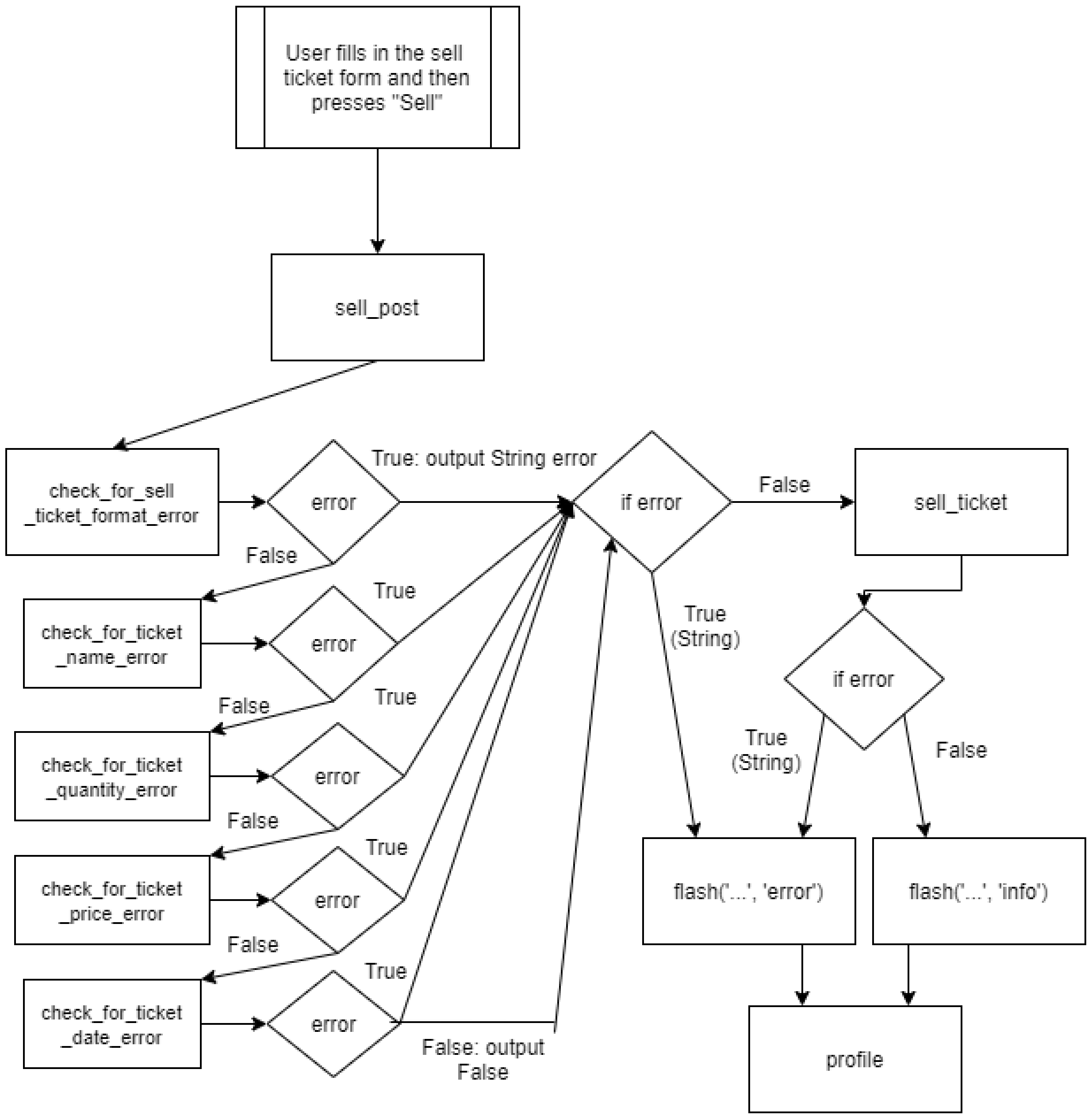
User fills in the sell ticket form and then presses "Sell"
sell_post
check_for_sell _ticket_format_error
error
True: output String error
False
check_for_ticket _name_error
error
True
False
check_for_ticket _quantity_error
error
True
False
check_for_ticket _price_error
error
True
False
check_for_ticket _date_error
error
True
False: output False
if error
False
True (String)
sell_ticket
if error
True (String)
False
flash('...', 'error')
flash('...', 'info')
profile

Figure 1: A diagram to show the functions used for the sell ticket feature

Update

This page contains an update form allowing a user to update a ticket. This form has four fields which are the ticket name, quantity, price per ticket and expiry date. The user enters the name of the ticket they want to update and then they input the quantity, price and date to be updated. Table 5 shows the inputs and outputs for all of the functions exclusive to the update section. Then, Table 6 gives a short description for each function. Finally, Figure 2 breaks the update system into a tree of functions, to demonstrate which functions are called by other functions.

Number	Name	Inputs	Outputs
1	update_post	NA	An HTML page that displays the profile page.
2	check_for_update_ticket_format_error	A ticket name, quantity, price and expiry date	If there is an error in one of the inputs an error message as a string is returned, else false is returned indicating that there are no errors.
3	update_ticket	A ticket name, quantity, price and expiry date	Returns a string if an error occurred, else False.

Table 5 - Input/output listing of update functions

Number	Name	Explanation
1	update_post	Called when a POST request is made to /update. If the given inputs match the R5 requirements, then the quantity, price and expiry date of the ticket of the given name are updated, else an error message is shown.
2	check_for_update_ticket_format_error	Called to check the tickets name, quantity, price and date to ensure they are the correct format. If they are not in the correct format an error message is returned, otherwise false is returned indicating that there are no errors.
3	update_ticket	Called when an update POST request is made and if the inputs follow the proper guidelines. This function updates the quantity, price and expiry data of a ticket of the given name database. In the event that the ticket cannot be updated, it returns a string representing the error, otherwise it will return False to represent success.

Table 6 - Descriptions of update functions

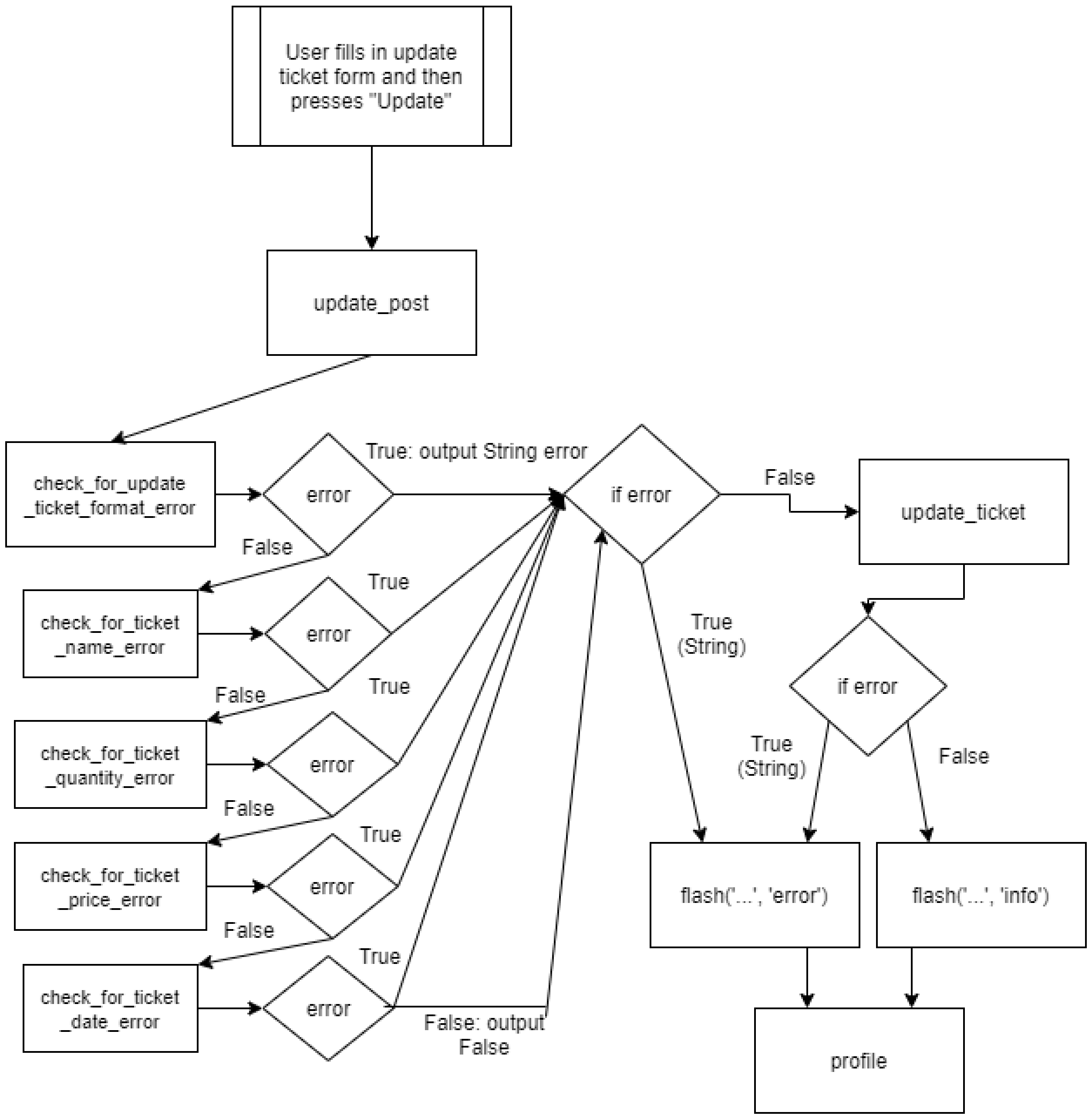
User fills in update ticket form and then presses "Update"
update_post
check_for_update _ticket_format_error
error
True: output String error
False
check_for_ticket _name_error
error
True
False
check_for_ticket _quantity_error
error
True
False
check_for_ticket _price_error
error
True
False
check_for_ticket _date_error
error
True
False: output False
if error
False
update_ticket
True (String)
if error
True (String)
False
flash('...', 'error')
flash('...', 'info')
profile

Figure 2: Function call diagram for update system

Buy

This page contains a buy form allowing a user to buy a ticket. This form has two fields which are the ticket name and quantity.The user enters the name of the ticket they want to buy and then they input the quantity they want to buy. Table 7 shows the inputs and outputs for all of the functions exclusive to the buy section. Then, Table 8 gives a short description for each function. Finally, Figure 3 breaks the buy system into a tree of functions, to demonstrate which functions are called by other functions.

Number	Name	Inputs	Outputs
1	buy	NA	An HTML page that displays the profile page.
2	get_ticket	name	Return a list of ticket
3	get_user	email	A User object
4	buy_ticket	email, price,name,quantity	NA
5	calculate_price_ticket	Price of ticket, quantity of ticket to buy	The numerical price of the ticket

Table 7 - Input/Output Listing of Buy Functions

Number	Name	Explanation
1	buy	Called when a POST request is made to /buy to perform buying actions.
2	get_all_tickets	Retrieves all tickets from the database.
3	get_user	Gets a user object for a specified email address.
4	buy_ticket	Retrieves the current user and then deducts the price of the ticket from the user’s account. Retrieves the current ticket and deducts the quantity of current ticket
5	calculate_price_ticket	Calculates total price of ticket by factoring in a 35% service few and a 5% tax.

Table 8 - Descriptions of buy functions

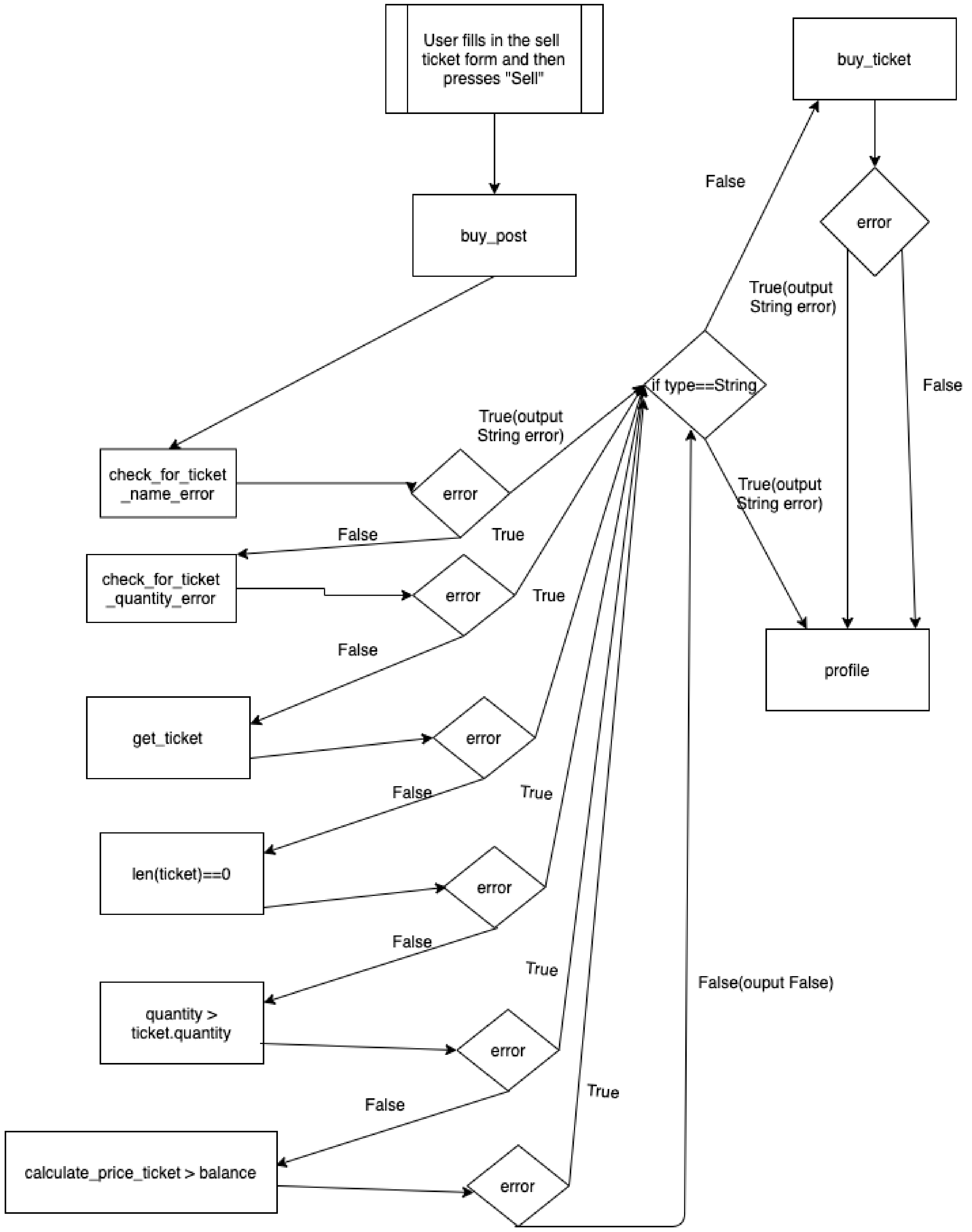
User fills in the sell ticket form and then presses "Sell"
buy_post
check_for_ticket _name_error
error
True(output String error)
False
True
check_for_ticket _quantity_error
error
True
False
get_ticket
error
False
True
len(ticket)==0
error
False
True
quantity > ticket.quantity
error
False
True
calculate_price_ticket > balance
error
False(ouput False)
if type==String
False
True(output String error)
buy_ticket
error
True(output String error)
False
profile

Figure 3: Function diagram for buy