



2020

PS1 Code Judgement Rubric

ONLY CODE JUDGES SHOULD BE USING THIS VERSION OF THE RUBRIC!

Judge (not shared with students): _____
Judged Team (name or number): _____
Date: _____

Instructions: for each requirement, mark the box that most closely aligns to your impression of the solution. Your judgments here should reflect how well the solution satisfies the directions in the problem statement and the team's presentation as well as your own review of the solution's source code. **Aesthetic appeal of a solution is not more important than functionality.** If a solution is not pretty but otherwise meets a requirement, that counts as reasonably satisfying that requirement. Keep your expectations high, teams had **a whole month** to work on this. **DO NOT FEEL OBLIGATED TO AWARD HIGH SCORES!** In the case that a solution attempts to "satisfy" a requirement by making it *look* like it works via the UI/demo code but the required feature is actually non-functional, a score of 0 is appropriate.

Solution Functionality Review [total weight: 100]

Requirement(s) to Reference	Score			
	0	1	2	3
	Missing or Completely Non-functional	Attempted But Does Not Meet My Expectations	Meets My Expectations	Exceeds My Expectations
Requirement 1	You do not have to login (i.e. you're a guest user) to view a list of, view info about, or book flights; view details of a ticket you've purchased as a guest; or create a customer account. [weight: 1]			
	Providing random login credentials <i>does not</i> allow you to log into the system successfully. [weight: 3]			
	Providing the correct login credentials <i>does</i> allow you to log into the system successfully. [weight: 1]			
	When you are already logged in, you should not be able to create a new account using the same URL/link used to create accounts when you're not logged in. [weight: 1]			
	The system supports 3 types of user (at minimum): guests, customers, and administrators. You can login as a customer or an administrator. [weight: 2]			

	Logging in as a customer allows you to do everything a guest can do except create a new account. Flights booked by this account are automatically associated with it. [weight: 2]		
	A customer account can delete itself. An administrator account cannot delete itself. [weight: 1]		
	A unique “root” user exists. (For code judges) The login system is based in the backend, backed by a database/datastore of some kind, and is not implemented solely in the frontend. [weight: 2]		
Requirement 2 Requirement 11	The flight view (the page that lists all flights in the system) is pleasant to look at, elegant, and ergonomic. It is pretty simple to figure out and use effectively. [weight: 1]		
	You can view, search through, and sort through a paginated list of all flights as long as they’re not flights with a “past” status. [weight: 5]		
	It is always obvious when a flight is a departure vs when it is an arrival. [weight: 1]		
	The flight view allows access to all of the required information for flights (depending on type). [weight: 5]		
	All flights are searchable and sortable by flight number, airline, airport, city, and arrival/departure time (should allow for a range of times); sorting can be done in ascending or descending order. [weight: 5]		
	You can book a flight from the flight view’s search results by clicking a link/button of some kind for that flight. [weight: 1]		

	Tickets can only be booked for flights scheduled to depart in more than 24-hours from the time of booking. [weight: 2]			
	Tickets can only be booked for departing flights scheduled to land at the airport the team represents before departing. Bookings for flights landing at other airports cannot be made. Note: the airport name and “short code” for this team is in the Reviewr submission form. [weight: 3]			
	Cancelled flights are differentiated from other flights in the UI. [weight: 1]			
	When looking at two flights where the first flight <i>does not</i> have the “past” or “cancelled” statuses and the second flight has the “departed” status, the information on your screen updates automatically after you use Tool 1 (open in another tab). Pay special attention to gate , arrival time , and status changes. The update should happen within a minute of making the change and <u>without pressing F5 or otherwise triggering a browser refresh manually</u> . If this solution is not using or is otherwise bypassing the API and is generating its own flight data, notify the chief judge immediately. [weight: 5]			
Requirement 3				
	The booking view has the appropriate components spelled out in the requirement text and provides the functionality spelled out at each bullet point. [weight: 4]			
	Inputting form data is pleasant and reasonable. The UI makes sense. [weight: 1]			
	Flight seats are the correct price. It was easy and intuitive for you to select and confirm your seat. You are able to change your booking details or go back into earlier stages of the form before you purchase. You were able to review your purchase before confirming. [weight: 2]			
	You cannot book a flight 24 hours (or less) before its departure time. [weight: 1]			
	You cannot book the same seat on the same flight twice. [weight: 1]			

Requirement 4 Requirement 11	You can get to the ticket view easily when logged in as a customer and you want to view the ticket for your previously purchased booking. Similarly, you can get to this view easily when you're only a guest by using a flight confirmation code you received at purchase or something similar. This view is accessible from each of the four ways listed in the problem statement text for requirement 4. [weight: 4]			
	The minimum required information (found in the problem statement text) is shown. [weight: 2]			
	When looking at the ticket for a flight you've just booked, the information on your screen updates automatically after you use Tool 2 (open in another tab) to change it. The update should happen within a minute of making the change and <u>without pressing F5 or otherwise triggering a browser refresh</u> manually. If this solution is not using or is otherwise bypassing the API and is generating its own flight data, notify the chief judge immediately. [weight: 5]			
Requirement 5	When viewing your customer dashboard, you are presented with your name, last login IP address, and the date and time of your last login. [weight: 1]			
	The dashboard shows upcoming and past flights as described in the problem statement text for requirement 5. [weight: 1]			
	As a customer, you can do each of the five items listed under "Customers can also:" found in the problem statement text for requirement 5. [weight: 5]			
Requirement 6	When logged in as an administrator, you have access to the administrator dashboard. It has all of the required functionality described in the problem statement text for requirement 6. [weight: 3]			
Requirement 7	You can create a new account as a guest (i.e. register). When you do so, you cannot use a weak password, must answer a simple CAPTCHA, and must fill out all of the items described in the problem statement text for requirement 7. [weight: 2]			

	When creating accounts, administrators must provide a secure password. [weight: 1]			
	When an account created by an administrator logs in for the first time, you are prompted to change your password and fill in any missing fields. [weight: 1]			
	Only the "root" account can create or delete other administrator accounts or modify another administrator account's information. Only customers can delete their own accounts. [weight: 1]			
Requirement 8				
	When logging in, you had to give your first name, last name, and a password at the very least. [weight: 1]			
	The login system seems reasonable, easy, and does not have any security flaws to the best of your knowledge. [weight: 1]			
	You cannot login for an hour after 3 failed login attempts. [weight: 1]			
	There is "remember me" functionality when logging in. You are automatically logged out after 15 minutes of no activity unless you are using "remember me" functionality. [weight: 1]			
Requirement 9				
	You cannot book a flight for a passenger with the following information: Name: Ronald McDonald (no middle name), Sex: male , Birthdate: June 14, 1963 This passenger should not be able to get a ticket even if their name is spelled in all caps or in lowercase, too! [weight: 2]			
Requirement 10				
	You can use your security questions to reset your password and gain entry into your account. This should be available through a "forgot password" link of some kind when visiting the login view. [weight: 2]			

Requirement 11	When you view flight/ticket information, you notice the data on your screen change (where appropriate) after around a minute after data changes occur in the API and all without you having to press F5 or otherwise manually refresh the page. See the text of the problem statement for requirement 11 for more details on this. [weight: 1]			
Requirement 12	All results and lists of items displayed in the frontend UI are paginated where appropriate. See the text of the problem statement for requirement 13 for more details on this. [weight: 1]			
Requirement 13	You encountered no XSS, SQL injection, or related security vulnerabilities, to the best of your knowledge. See the text of the problem statement for requirement 13 for more details on this. [weight: 2]			
	Passwords seem to be transacted and stored securely. (For code judges) Passwords are hashed/encrypted when stored in the database and are actually transacted securely. [weight: 1]			
Requirement 14	Using Tool 3 , ensure the application fails gracefully when bad response codes are returned by the API such as 401, 404, 500, 503, etc. [weight: 2]			
Requirement 15	This app looks good when I view it on a small smartphone-sized screen. To simulate this, use your mouse to drag the right edge of the browser window left until it cannot shrink anymore. This shrunken column represents the size of a phone screen. Is the app still usable in this state? Is it visually pleasant? [weight: 2]			
	This app looks good when I view it on a medium tablet-sized screen. To simulate this, make your browser window half the size of your screen (try windows key + left arrow). This simulates the size of a tablet screen. Is the app still usable? Is it visually pleasant? [weight: 1]			
	This app looks good when I view it on a normal desktop/laptop screen. Ensure the browser window is maximized. [weight: 2]			

	Overall, I feel this solution is easy to use. [weight: 1]			
	Overall, I feel this solution is beautiful to look at. [weight: 1]			
	Overall, I feel this solution functioned well. App functionality that worked fine when the browser window was maximized did not suddenly break when you shrunk the browser to simulate a smartphone or a tablet. [weight: 1]			

Instructions: for each of the five criteria below, mark the box that most closely aligns to your impression of the solution's source code after your review. Does their source code meet your expectations of quality? **If you notice source code that seems strangely out of place, overly complex, extremely nonsensical, suspiciously unformatted or very badly organized syntactically, or you otherwise infer that students copy-pasted/plagiarized part of their solution's source code from the internet or some illegal source, inform the chief judge immediately.**

Solution Source Code Review [total weight: 100]

Criterion	Score			
	0	1	2	3
	Does Not Meet Expectations	Partially Meets	Meets Expectations	Exceeds Expectations
Naming Conventions	Variables should be well-named (e.g. accountSum is good while acsu is bad). There should be few "magic numbers" floating around. The chosen variable naming scheme should be consistent and self-documenting. [weight: 5]			
Organization	The code should be well-organized, with a consistent logical file/directory layout, encapsulation, and proper separation of concerns. Similar units of code should be grouped together in a logical way. If object-oriented language features are used, they should be used properly (i.e. <u>SOLID</u>). [weight: 30]			
Maintainability and Extensibility	The solution should be implemented such that a future group of programmers would not have to rewrite large portions of the source code to add new functionality. Source code should be <u>DRY</u> where reasonable. Pay special care to inflexible and hardcoded values. [weight: 25]			
Readability	How easily readable is the source code? Did you have a hard time going through it? Solution code should be as intuitive and self-documenting as possible. Confusing sections should be documented with comments. There should be few " <u>WTF?!</u> " moments. [weight: 25]			
Efficiency	Any implementations should not be too inefficient. Does some piece of code waste cycles looping or otherwise doing something unnecessary? Is the application wasteful with system resources? Does it make more network requests than necessary? [weight: 15]			



--	--	--	--	--

Positive Comments

Please share any positive comments you have for this team.

--

Constructive Criticism

Please share any unanswered concerns or comments you have for this team.

--

Two Questions For This Team

Please share the two most pressing questions you'd like answered about this team's solution.

--

Shared Thoughts (visible to all judges)

Please add any thoughts or concerns you believe other judges should be made aware of.

--