

# Server Test Spec

	Requirement I	Test ID	Test Objective	Precondition	Steps	Test Data	Expected Result	Actual Result	Status	Bug ID
REGISTER										
	F.1.1	Register-1	User can register with valid credentials.	User is not registered.	1. Call register on ServerFacade with valid credentials.		New User is created and stored. Hashed password (not plaintext) and salt is stored. Salt is generated randomly. User is logged in to server.			
	F.1.1	Register-2	User cannot register if inputs are null.	User is not registered.	1. Call register on ServerFacade with null inputs.		Error thrown.			
	F.1.1	Register-3	User cannot register with duplicate username.	User is not registered.	1. Call register on ServerFacade with duplicate username.		Error thrown.			
LOGIN										
	F.1.2	Login-1	User can login with valid credentials.	User exists in server. User is not logged in.	1. Call login on ServerFacade with valid credentials.		User is logged in. User's correct ID is returned.			
	F.1.2	Login-2	User cannot login with invalid username.		1. Call login on ServerFacade with invalid username.		Error thrown.			
	F.1.2	Login-3	User cannot login with invalid password.	User exists in server.	1. Call login on ServerFacade with invalid password.		Error thrown.			
	F.1.2	Login-4	User cannot login with null inputs.		1. Call login on ServerFacade with null inputs.		Error thrown.			
	F.1.2	Login-5	User cannot login if already logged in.	User is logged in.	1. Successfully call login on ServerFacade. 2. Call login again.		Error thrown.			
LOGOUT										
	F.1.3	Logout-1	User can logout.	User is logged in.	1. Successfully login to server. 2. Call logout on ServerFacade.		User is logged out.			
	F.1.3	Logout-2	User cannot logout if not logged in.	User is not logged in.	1. Call logout on ServerFacade.		Error thrown.			
	F.1.3	Logout-3	User cannot logout with invalid User ID.	User is logged in.	1. Call logout on ServerFacade with an invalid User ID.		Error thrown.			
	F.1.3	Logout-4	User cannot logout another User.				Error thrown.			
CREATE GAME										
	F.2.1	Create-1	User can create game with valid credentials.	User is logged in.	1. Successfully login to server. 2. Call createGame on ServerFacade.		A game is added to the server with the provided information. The user is added to the newly created game with the specified color.			
	F.2.1	Create-2	User cannot create game with invalid input.	User is logged in.	1. Successfully login to server. 2. Call createGame on ServerFacade with invalid inputs.		Error thrown.			
	F.2.1	Create-3	User cannot create game when not logged in.	User is not logged in.	1. Call createGame on ServerFacade without logging in.		Error thrown.			
ADD PLAYER TO GAME										
	F.2.2	Add-1	User can add themselves to a game with valid credentials.		1. Login to server. 2. Call addPlayerToGame on ServerFacade with valid inputs.		User is added to the game with the given color.			
	F.2.2	Add-2	User cannot add themselves to a full game.	User is logged in.	1. Create game. 2. Add players to game until full. 3. Call addPlayerToGame on full game.		Error thrown.			
	F.2.2	Add-3	User cannot add themselves to a game that has already started.	User is logged in.	1. Create game. 2. Add player. 3. Start game. 4. Call addPlayerToGame on already started game.		Error thrown.			
	F.2.2	Add-4	User cannot add themselves to a game they already joined.	User is logged in.	1. Login to server. 2. Create game, adding self to new game. 3. Call addPlayerToGame on already joined game.		Error thrown.			
	F.2.2	Add-5	User cannot join game with duplicate color.		1. Login to server. 2. Create game. 3. On a separate client, call addPlayerToGame with same color as other player.		Error thrown.			
	F.2.2	Add-6	User cannot join game with null inputs.	User is logged in.	1. Call addPlayerToGame with null inputs.		Error thrown.			
	F.2.2	Add-7	User cannot join game with invalid inputs.	User is logged in.	1. Call addPlayerToGame with invalid inputs.		Error thrown.			
	F.2.2	Add-8	User cannot join game when not logged in.	User is not logged in.	1. Call addPlayerToGame with valid inputs.		Error thrown.			
START GAME										
	F.2.5	Start-1	User can start a game with valid credentials.	User is logged in.	1. Call startGame on ServerFacade with valid credentials.		The gameStarted flag is set to true. It is the user's turn. Game's initial card selection begins.			
	F.2.5	Start-2	User cannot start a game they did not create.	User is logged in.	1. Call startGame on game not created by the user.		Error thrown.			
	F.2.5	Start-3	User cannot start a game with only 1 player.	User is logged in.	1. Create game. 2. Call startGame on newly created game. (There should only be one player)		Error thrown.			
	F.2.5	Start-4	User cannot start game with invalid inputs.	User is logged in.	1. Call startGame with invalid inputs.		Error thrown.			
	F.2.5	Start-5	User cannot start game when not logged in.	User is not logged in.	1. Call startGame when not logged in.		Error thrown.			
LEAVE GAME										
	F.3.8	Leave-1	User can leave a game with valid credentials.	User is logged in.	1. Join a previously created game. 2. Call leaveGame on ServerFacade with valid credentials.		The User is removed from the game.			
	F.3.8	Leave-2	User cannot leave a game that has already started.	User is logged in.	1. Join a game. 2. Start the game. 3. Call leaveGame on ServerFacade.		Error thrown.			
	F.3.8	Leave-3	User cannot leave game they are not currently in.	User is logged in.	1. Call leaveGame with a game User is not currently in.		Error thrown.			
	F.3.8	Leave-4	User cannot leave game with invalid inputs.	User is logged in.	1. Call leaveGame with invalid inputs.		Error thrown.			
	F.3.8	Leave-5	User cannot leave game when not logged in.	User is not logged in.	1. Call leaveGame with valid inputs.		Error thrown.			
BUY ROUTE										
	F.3.5	Buy-1	User can buy a route with valid credentials.	User is logged in. Game has started. It is User's turn.	1. Call buyRoute on ServerFacade with valid inputs.		The route is bought and assigned to the user. The User's train cars decrease appropriately. The User's train cards decrease appropriately. Turn advances to next player.			
	F.3.5	Buy-2	User cannot buy route with insufficient train cars.	User is logged in. Game has started. It is User's turn.	1. Call buyRoute with insufficient train cars.		Error thrown.			
	F.3.5	Buy-3	User cannot buy route when not his/her turn.	User is logged in. Game has started. It is not User's turn.	1. Call buyRoute on ServerFacade with valid inputs.		Error thrown.			
	F.3.5	Buy-4	User cannot buy route when not logged in.	User is not logged in.	1. Call buyRoute with valid inputs.		Error thrown.			
	F.3.5	Buy-5	User cannot buy route with insufficient train cards.	User is logged in. Game has started. It is User's turn.	1. Call buyRoute with insufficient train cards.		Error thrown.			
	F.3.5	Buy-6	Longest route is updated correctly.	User is logged in. Game has started. It is User's turn. User does not have longest route.	1. Call buyRoute such that User now has longest route.		The longest route flag is set for the current user. The longest route flag is unset for every other user. The route is bought and assigned to the user. The User's train cards decrease appropriately. The User's train cards decrease appropriately. Turn advances to next player.			
	F.3.5	Buy-7	Points are updated correctly.	User is logged in. Game has started. It is User's turn.	1. Call buyRoute with valid inputs.		User's points increase correctly.			

# Server Test Spec

[illegible]

Server Test Spec

SERVER SPEED TEST										
		Speed-1	The server can handle between 40-50 requests per second.		1. Create multiple client thread that continually generate requests for 1 minute.		No exceptions are thrown, all requests have a response within 1000ms			
SERVER SYNCHRONIZATION TEST										
COMMAND GENERATING										
		Command-1	The Command Factory can correctly create command objects of each type.		1. Provide Command Factory with input for each command type.		Each command generated by the factory is not null			