

TDD Assignment

```
class Inputter{  
    function signUp(string email, string password){  
        new Service =s.Service();  
        s.store(email,password);  
    }  
}
```

```
class Service{  
    function validateEmail(string email){  
    }  
    function validatePassword(string password){  
    }  
    function store(string email, string password){  
        validateEmail(email);  
        validatePassword(password);  
    }  
}
```

Acceptance Criteria:

User should sign up when:-

- 1)The email address is valid.
- 2)The email address is not missing.
- 3)The password is valid.
- 4)The email address is unique.
- 5)The password length is greater than 8.

ITERATION	TEST	FUNCTION BODY	STATUS
1)	Output = a.validateEmail(abc) Output == FALSE	validateEmail(email){ }	Fail
2)	Output = a.validateEmail(abc) Output == FALSE	validateEmail(email){ if(email contains @){return	Pass

		<pre> TRUE} else{return FALSE} } </pre>	
3)	Output = a.validateEmail(abc@) Output == FALSE	<pre> validateEmail(email){ if(email contains @){return TRUE} else{return FALSE} } </pre>	Fail
4)	Output = a.validateEmail(abc@) Output ==FALSE	<pre> validateEmail(email){ if(email is of the form string1@domain where string1 is a string){return TRUE} else{return FALSE} } </pre>	Pass
5)	Output = a.validateEmail(abc@xyz) Output ==FALSE	<pre> validateEmail(email){ if(email is of the form string1@domain where string1 && domain is a string){return TRUE} else{return FALSE} } </pre>	Fail
6)	Output = a.validateEmail(abc@xyz) Output ==FALSE	<pre> validateEmail(email){ if(email is of the form string1@domain.com where string1 && domain is a string){ return TRUE Out} else{return FALSE} } </pre>	Pass
7)	Output = a.validateEmail(abc@xyz.com) Output ==TRUE	<pre> validateEmail(email){ if(email is of the form string1@domain.com where string1 && domain is a string) { return TRUE } else{return FALSE} } </pre>	Pass
8)	Output = a.validateEmail(abc@xyz@xyz.c om) Output ==FALSE	<pre> validateEmail(email){ if(email is of the form string1@domain.com where string1 && domain is a string){return TRUE} else{return FALSE} } </pre>	Fail

		}	
9)	Output = a.validateEmail(abc@xyz@xyz.com) Output ==FALSE	validateEmail(email){ if(email is of the form string1@domain.com where string1 and domain is a string and string1 does not contain two @ symbols) { return TRUE } else{return FALSE} }	Pass
10)	Output = a.validateEmail(abc.cd@xyz.com) Output ==FALSE	validateEmail(email){ if(email is of the form string1@domain.com where string1 && domain is a string && string1 does not contain two @ symbols){ return TRUE } else{return FALSE} }	Fail
11)	Output = a.validateEmail(abc.cd@xyz.com) Output ==FALSE	validateEmail(email){ if(email is of the form string1@domain.com where string1 && domain is a string && string1 does not contain two @ symbols and multiple dots){ return TRUE } else{return FALSE} }	Pass
12)	Output = a.validateEmail(abc@xyz..com) Output ==FALSE	validateEmail(email){ if(email is of the form string1@domain.com where string1 && domain is a string && string1 does not contain two @ symbols && multiple dots){ return TRUE } }	Fail

		<pre> else{return FALSE} } </pre>	
13)	Output = a.validateEmail(abc@xyz..com) Output ==FALSE	<pre> validateEmail(email){ if(email is of the form string1@domain.com where string1 && domain is a string && string1 does not contain two @ symbols && multiple dots && domain doesn't contain dots){ return TRUE } else{return FALSE} } </pre>	Pass
14)	Output=a.validatePassword(abc) Output==FALSE	<pre> validatePassword(password){ } </pre>	Fail
15)	Output=a.validatePassword(abc) Output==FALSE	<pre> validatePassword(password){ if(password.length >= 8){ return TRUE; } else{return FALSE} } </pre>	Pass
16)	Output=a.validatePassword(abcd efgh) Output==FALSE	<pre> validatePassword(password){ if(password.length >= 8){ return TRUE; } else{return FALSE} } </pre>	Fail
17)	Output=a.validatePassword(abcd efgh) Output==FALSE	<pre> validatePassword(password){ if(password.length >= 8 && password should contain at least one alphabet, one digit,one special character){ return TRUE; } else{return FALSE} } </pre>	Pass
18)	Output=a.validatePassword(abcd e2@gh)	<pre> validatePassword(password){ if(password.length >= 8 && </pre>	Pass

	Output==TRUE	password should contain at least one alphabet, one digit, one special character){ return TRUE; } else{return FALSE} }	
19)	Output=a.validatePassword(xycdab@ah) Output==TRUE	validatePassword(password){ if(password.length >= 8 && password should contain at least one alphabet, one digit, one special character){ return TRUE; } else{return FALSE} }	Pass
20)	Output=a.store(abc@xyz.com,abc@12ww) Output==FALSE	store(email,password){ }	Fail
21)	Output=a.store(abc@xyz.com,abc@12ww) Output==Store in database	store(email,password){ if(validateEmail(email) && validatePassword(password)){ print("Store in databse"); } else{ return error("Store failed") } }	Pass
22)	Output=a.store(abc@xyz.com,abc@12ww) Output==Duplicate email address	store(email,password){ if(validateEmail(email) && validatePassword(password)){ print("Store in databse"); } else{ return error("Store failed") } }	Fail

		<pre> else if{ Service s; (s.store(email,password)){ print("Signed up"); } else{ print("Sign up failed"); } } </pre>	
28)	Output=signUp(abz@xyz.com,a bzer@3787) Output == Signed Up	<pre> signUp(email,password){ if(email == ""){print("Missing email")} else if{ Service s; (s.store(email,password)){ print("Signed up"); } else{ print("Sign up failed"); } } </pre>	Pass