Iteration	Test	Function	Status

1.	Service s;
	Output = s.validateEmail(abc)
	
	Output==Error(“Email must contain @”)
		{
		
		}
			Fail

2.	Service s;
	Output = s.validateEmail(abc)
	
	Output==Error(“Email must contain @”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @)”
		else
		return true
		}
			Pass

3.	Service s;
	Output = s.validateEmail(abc@)
	
	Output==Error(“Invalid Email”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else
		return true
		}
			Fail

4.	Service s;
	Output = s.validateEmail(abc@)
	
	Output==Error(“Invalid Email”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(email contains '@' but no domain.com)
		return Error(“Invalid email”)
		
		else
		return true
		}
			Pass

5.	Service s;
	Output = s.validateEmail(abc@gmail.com)
	
	Output==true
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(email contains '@' but no domain.com)
		return Error(“Invalid email”)
		
		else
		return true
		}
			Pass

6.	Service s;
	Output = s.validateEmail(@gmail.com)
	
	Output==Error(“Invalid Email”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(email contains '@' but no domain.com)
		return Error(“Invalid email”)
		
		else
		return true
		}
			Fail

7.	Service s;
	Output = s.validateEmail(@gmail.com)
	
	Output==Error(“Invalid Email”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(email contains '@' but no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' AND domain.com but no username)
		return Error(“Invalid email”)
		
		else
		return true
		}
			Pass

8.	Service s;
	Output = s.validateEmail(abc@gmail)
	
	Output==Error(“Invalid Email”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(email contains '@' but no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' AND domain.com but no username)
		return Error(“Invalid email”)
		
		else
		return true
		}
			Pass

9.	Service s;
	Output = s.validateEmail(abc@.com)
	
	Output==Error(“Invalid Email”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(email contains '@' but no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' and domain.com but no username)
		return Error(“Invalid email”)
		
		else
		return true
		}
			Pass

10.	Service s;
	Output = s.validateEmail(abc@soulknit.org)
	
	Output==true
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(email contains '@' but no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' and domain.com but no username)
		return Error(“Invalid email”)
		
		else
		return true
		}
			Fail

11.	Service s;
	Output = s.validateEmail(abc@soulknit.org)
	
	Output==true
		validateEmail(email){
		if (email does not contain '@')
		return Error("Email must contain @")
		
		else if(email contains '@' but no domain.com)
		return Error("Invalid email")
		
		else if(email contains '@' and domain.com but no username)
		return Error("Invalid email")
		
		else if(email contains username and '@' and domain.org)
		return true
		else
		return true
		}
			Pass

12.	Service s;
	Output=s.validateEmail(.abc@gmail.com)
	
	Output==Error("Invalid Email")
		validateEmail(email){
		if (email does not contain '@')
		return Error("Email must contain @")
		
		else if(email contains '@' but no domain.com)
		return Error("Invalid email")
		
		else if(email contains '@' and domain.com but no username)
		return Error("Invalid email")
		
		else if(email contains username and '@' and domain.org)
		return true
		else
		return true
		}
			Fail

13.	Service s;
	Output=s.validateEmail(.abc@gmail.com)
	
	Output==Error("Invalid Email")
		validateEmail(email){
		if (email does not contain '@')
		return Error("Email must contain @")
		
		else if(email contains '@' but no domain.com)
		return Error("Invalid email")
		
		else if(email contains '@' and domain.com but no username)
		return Error("Invalid email")
		
		else if(username starts with '.')
		return Error("Invalid email")
		
		else if(email contains username and '@' and domain.org)
		return true
		else
		return true
		}
			Pass

14.	Service s;
	Output=s.validateEmail(abc.@gmail.com)
	
	Output==Error(“Invalid Email”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(email contains '@' but no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' and domain.com but no username)
		return Error(“Invalid email”)
		
		else if(username starts with '.')
		return Error(“Invalid email”)
		
		else if(email contains username and '@' and domain.org)
		return true
		else
		return true
		}
			Fail

15.	Service s;
	Output=s.validateEmail(abc.@gmail.com)
	
	Output==Error(“Invalid Email”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(email contains '@' and no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' and domain.com but no username)
		return Error(“Invalid email”)
		
		else if(username starts or ends with '.')
		return Error(“Invalid email”)
		
		else if(email contains username and '@' and domain.org)
		return true
		else
		return true
		}
			Pass

16.	Service s;
	Output=s.validateEmail(abc@gmail
	@gmail.com)
	
	Output==Error(“Invalid Email”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(email contains '@' and no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' and domain.com but no username)
		return Error(“Invalid email”)
		
		else if(username starts or ends with '.')
		return Error(“Invalid email”)
		
		else if(email contains username and '@' and domain.org)
		return true
		else
		return true
		}
			Fail

17.	Service s;
	Output=s.validateEmail(abc@gmail
	@gmail.com)
	
	Output==Error(“Invalid Email”)
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(number of '@' > 1)
		return Error(“Invalid email”)
		
		else if(email contains '@' and no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' and domain.com but no username)
		return Error(“Invalid email”)
		
		else if(username starts or ends with '.')
		return Error(“Invalid email”)
		
		else if(email contains username and '@' and domain.org)
		return true
		else
		return true
		}
			Pass

18.	Service s;
	Output=s.validateEmail(xyz@outlook.com)
	
	Output==true
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(number of '@' > 1)
		return Error(“Invalid email”)
		
		else if(email contains '@' and no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' and domain.com but no username)
		return Error(“Invalid email”)
		
		else if(username starts or ends with '.')
		return Error(“Invalid email”)
		
		else if(email contains username and '@' and domain.org)
		return true
		else
		return true
		}
			Pass

19.	Service s;
	Output=s.validateEmail(jkl@yahoo.com)
	
	Output==true
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(number of '@' > 1)
		return Error(“Invalid email”)
		
		else if(email contains '@' and no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' and domain.com but no username)
		return Error(“Invalid email”)
		
		else if(username starts or ends with '.')
		return Error(“Invalid email”)
		
		else if(email contains username and '@' and domain.org)
		return true
		else
		return true
		}
			Pass

20.	Service s;
	Output=
	s.validateEmail(j123kl@yahoo.com)
	
	Output==true
		validateEmail(email){
		if (email does not contain '@')
		return Error(“Email must contain @”)
		
		else if(number of '@' > 1)
		return Error(“Invalid email”)
		
		else if(email contains '@' and no domain.com)
		return Error(“Invalid email”)
		
		else if(email contains '@' and domain.com but no username)
		return Error(“Invalid email”)
		
		else if(username starts or ends with '.')
		return Error(“Invalid email”)
		
		else if(email contains username and '@' and domain.org)
		return true
		else
		return true
		}
			Pass

21.	Service s; Output=s.validatePassword(“123”) Output==Error(“Password must have atleast 8 characters.”)	{ }	Fail

22.	Service s;
	Output=s.validatePassword(“123”)
	
	Output==Error(“Password must have atleast 8 characters.”)
		validatePassword(password){
		if (password.length < 8)
		return Error(“Password must have atleast 8 characters”)
		else
		return true
		}
			Pass

23.	Service s;
	Output=s.validatePassword(“abcdefghijk”)
	
	Output==true
		validatePassword(password){
		if (password.length < 8)
		return Error(“Password must have atleast 8 characters”)
		else
		return true
		}
			Pass

24.	Service s;
	Output=s.validatePassword(“12345678”)
	
	Output==true
		validatePassword(password){
		if (password.length < 8)
		return Error(“Password must have atleast 8 characters”)
		else
		return true
		}
			Pass

25.	Service s;
	Output=s.validatePassword(“abcd1234”)
	
	Output==true
		validatePassword(password){
		if (password.length < 8)
		return Error(“Password must have atleast 8 characters”)
		else
		return true
		}
			Pass

26.	Service s;
	Output=s.validatePassword(“abcd1234abcd1234b”)
	
	Output==Error(“Maximum password length is 16 characters”)
		validatePassword(password){
		if (password.length < 8)
		return Error(“Password must have atleast 8 characters”)
		else
		return true
		}
			Fail

27.	Service s;
	Output=s.validatePassword(“abcd1234abcd1234b”)
	
	Output==Error(“Maximum password length is 16 characters”)
		validatePassword(password){
		if (password.length < 8)
		return Error(“Password must have atleast 8 characters”)
		
		else if(password.length > 16)
		return Error(“Maximum password length is 16 characters”)
		else
		return true
		}
			Pass

28.	Service s;
	Output=s.store(“abc@gmail.com”,”abcdefgh”)
	
	output==true
		{
		
		}
			Fail

29.	Service s;
	Output=s.store(“abc@gmail.com”,”abcdefgh”)
	
	output==true
		store(email,password) {
		bool1=validateEmail(email)
		bool2=validatePassword(password)
		if(bool1==true and bool2==true) {
		if(email already exists in database)
		return Error(“Email already exists”)
		else
		Add to database and return true
		
		}
		else
		return false
		
		}
			Pass

30.	Service s;
	Output=s.store(“abc@gmail.com”,”abcdefgh”)
	
	output== Error(“ Email already exists”)
		store(email,password) {
		bool1=validateEmail(email)
		bool2=validatePassword(password)
		if(bool1==true and bool2==true) {
		if(email already exists in database)
		return Error(“Email already exists”)
		else
		Add to database and return true
		
		}
		else
		return false
		
		}
			Pass

31.	Service s;
	Output=s.store("@gmail.com","abcdefgh")
	
	output== false
		store(email,password) {
		bool1=validateEmail(email)
		bool2=validatePassword(password)
		if(bool1==true and bool2==true) {
		if(email already exists in database)
		return Error("Email already exists")
		else
		Add to database and return true
		
		}
		else
		return false
		
		}
			Pass

32.	Service s;
	Output=s.store("xyz@gmail.com","abcde")
	
	output== false
		store(email,password) {
		bool1=validateEmail(email)
		bool2=validatePassword(password)
		if(bool1==true and bool2==true) {
		if(email already exists in database)
		return Error("Email already exists")
		else
		Add to database and return true
		
		}
		else
		return false
		
		}
			Pass

33.	Inputter obj;
	output=obj.signUp("","")
	
	output==Error("Missing fields. Please enter username and password.")
		{
		
		}
			Fail

34.	Inputter obj;
	output=obj.signUp("","")
	
	output==Error("Missing fields")
		signUp(username,password){
		if (username ="" or password ="")
		return Error("Missing fields")
		}
			Pass

35.	Inputter obj;
	output=obj.signUp("xyz@yahoo.com","12345678")
	
	output==true
		signUp(username,password){
		if (username ="" or password ="")
		return Error("Missing fields")
		}
			Fail

36.	Inputter obj;
	output=obj.signUp("xyz@yahoo.com","12345678")
	
	output==true
		signUp(username,password){
		if (username ="" or password ="")
		return Error("Missing fields")
		else{
		Service s;
		bool1=s.store(username,password)
		if(bool1==true)
		return true;
		else
		return false.
		}
		}
			Pass

37.	Inputter obj;
	output=obj.signUp("xyz@yahoo.com","abcdefgh")
	//Trying to sign up xyz@yahoo.com again (duplication)
	
	output==false
		signUp(username,password){
		if (username ="" or password ="")
		return Error("Missing fields")
		else{
		Service s;
		bool1=s.store(username,password)
		if(bool1==true)
		return true;
		else
		return false.
		}
		}
			Pass

38.	Inputter obj;
	output=obj.signUp("ikl@yahoo.com","")
	
	output==Error("Missing fields")
		signUp(username,password){
		if (username ="" or password ="")
		return Error("Missing fields")
		else{
		Service s;
		bool1=s.store(username,password)
		if(bool1==true)
		return true;
		else
		return false.
		}
		}
			Pass

General Structure:

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

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

Acceptance Criteria:
System should not sign up user if:
-Email is invalid or missing.
-Password is invalid or missing.
-Email already exists in database,

System should sign up user if:
-Email and password both are valid.
-User data is successfully stored into database.