

Unit testing

- Check that getters and setters for items work properly
- Check that bucket getters and setters work properly

```
@Test
public void bucketGetSet() {
    BucketItem testItem = new BucketItem();
    testItem.setDay("4");
    testItem.setDescription("THIS IS A TEST");
    testItem.setIsCompleted(true);
    assertEquals(testItem.toString(), actual: "BucketItem{" +
        "itemName='" + "" + '\'' +
        ", month='" + "" + '\'' +
        ", day='" + "4" + '\'' +
        ", year='" + "" + '\'' +
        ", description='" + "THIS IS A TEST" + '\'' +
        ", isCompleted=" + true +
        ", item_id=" + "" +
        ", priority=" + "0" +
        '}');
}
```

✓ Tests passed: 1 of 1 test - 5ms

Build Analyzer results available
7:28:35 PM: Task execution finished 'app:testDebugUnitTest --tests "com.example.wombatworkshop_dreamcatcher.ExampleUnitTest.bucketGetSet"'

- Item completion checkbox is updating database properly
- Confirm that item delegation is removing proper item in the database
- Confirm that correct item information is displayed when item is clicked in bucket list
- Confirm that an email is sent to the correct user when resetting password
- Confirm that a new bucket is saved to database when creating a new bucket

- Checking that the username is filled in
- Checking that the email is filled in and valid
- Checking that password is long enough and does not contain the username

```

if(fName.isEmpty()){
    editFName.setError("First name is Required!");
    editFName.requestFocus();
    return;
}
if(lName.isEmpty()){
    editLName.setError("Last name is Required!");
    editLName.requestFocus();
    return;
}
if(email.isEmpty()){
    editEmail.setError("Email is required!");
    editEmail.requestFocus();
    return;
}
if(!Patterns.EMAIL_ADDRESS.matcher(email).matches()){
    editEmail.setError("A valid email is required!");
    editEmail.requestFocus();
    return;
}
if(password.isEmpty()){
    editPassword.setError("Password is required!");
    editPassword.requestFocus();
    return;
}
if(password.length() < 8){
    editPassword.setError("Password length must contain at least 8 characters!");
    editPassword.requestFocus();
    return;
}

if (password.toLowerCase().contains(fName.toLowerCase()) || password.toLowerCase().contains(lName.toLowerCase())) {
    editPassword.setError("Password can not contain User Name!");
    editPassword.requestFocus();
    return;
}

```

Use Case Testing

- Creating an account
- Creating a bucket
- Creating an item
- Deleting an item
- Marking an item as completed
- Setting a priority to an item