

TDI deliverable 3

Members:

Kevin Tran

Bohdan Ivchenko

Kemeria Mustfa

Tewodros Sisay

Zachary Moriarty

Table of contents

Introduction	3
Requirements	3
Implemented user stories	3
Not implemented user stories	4
Instructions	5
Contributions	7
Testing	8
Source Catalog	11
Emergency Contacts	12

Introduction:

Team TDI worked to make a simple and easy to use catalog for notes and items around the house we decided to call “Household Handbook”. We ended up making a simple system of items that the user can generate, sort, and search through to organize their notes from around the house. While there are aspects we weren't able to get into the current iteration of the project, what we do have works well and helps keep track of things.

The application runs smoothly, starting from a login page where the user puts in their credentials, or registers if they do not have an account. From there it brings up a list of their items, that they can sort by the name of the items, where they are located, or what type of item it is. This makes organizing items around the house simple, and while there are features missing, these could be added in through future iterations.

Requirements Addressed

User Stories

User Story #1: I would like to register a new user.

Business Rules:

- Users are able to register a new account (✓)
- Password confirmation(✓)
- User can edit their account (X)
- Password encryption (X)

User Story #2: I bought a new washing machine and would like to store the description on how to use it.

Business Rules:

- User is able to copy information to the description (✓)
- User is able to label newly created objects with a type (✓)
- User can store the room location of the object (✓)
- User can create new room locations (✓)
- User can create new typings(✓)

User Story #3: I recently got rid of my washing machine and would like to delete it from the list of objects.

Business Rules:

- User is able to edit the description of the object(✓)
- User is able to delete the object (✓)
- User can edit the name of the object (✓)
- User can edit the room (X)
- User can edit the type (X)

User Story#4: As a user I want to upload images of my ongoing project.

Business Rules:

- User is able to add files associated with the object (X)

User Story #5: As a user I want to be able to search for my Wifi password within the table.

Business Rules:

- User is able to search for specific items within the list (✓)
- Entries must include the full name of the object

User Story #6: As a user I want to sort the list of objects by the room it is in and by the type.

Business Rules:

- User is able to sort the list alphabetically by rooms (✓)
- User is able to sort the list alphabetically by type (✓)

Incomplete User Stories:

User Story #7: As a user I want to upload images of my ongoing project.

Business Rules:

- User is able to add files associated with the object (X)

User Story #8: As a user I want to be able to select levels of administration privileges

Features:

- User are able to select admin privilege that allows them to access all features of the Household Handbook(add, delete, search, edit) (✓)
- User are able to select guest privilege that limits the features they are allowed to use in the Household Handbook(search, open object) (X)

How to run the application

Household Handbook

Our project is called home

#1: How to register a new user.

- 1) Click on Register button
- 2) Enter name, username, email, password.
- 3) Hit confirm
- 4) If I want to check where my information is, click the about button.

#2 How to add a new Object to the table.

- 1) Click the add button on the mainPage
- 2) Fill in the name of the object
- 3) Select a room and a type
- 4) Fill in the instructions for the washing machine
- 5) Click confirm.

#3: How to delete an Object from the table.

- 1) Find the object in the table and click on it and open
- 2) The information for the washing machine should show
- 3) Click the delete button

#4: How to search for an object

- 1) Click on the search bar
- 2) Type the name of the object on the search bar
- 3) Don't type only some part of the word but type all the words
- 4) Click search to get your result

#5: How to sort by alphabetical order

- 1) Go to sort by
- 2) Click on it
- 3) Choose how you want to sort it
- 4) Choose between "room", "name", "type"
- 5) Click on it

#6: How to Login on the application.

- 1) Type in your name in the name text field
- 2) Type in your password in the password text field
- 3) Click the Login button.
- 4) If the name and password are correct it'll successfully logged in

#7: How to edit an existing name

- 1) Select the name or room we want to edit
- 2) Click the "Get selected item" button
- 3) Click on edit button
- 4) Change the existing name
- 5) Click save

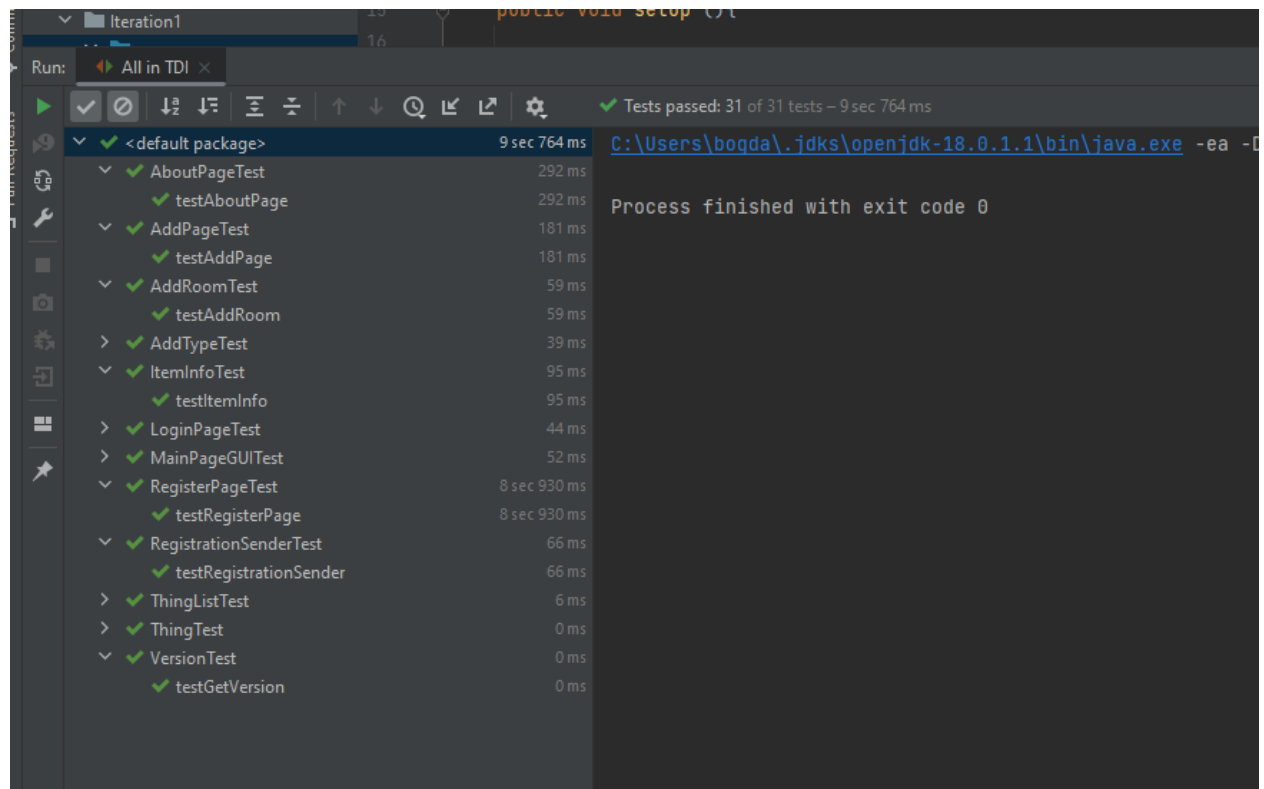
Contributions

Name	Class-Method	Activity
Group	Version class	Author
Group	aboutPage class	Author
Group	LoginPage class	Author
Zachary Moriarty	Thing class	Author
Zachary Moriarty	ThingList class	Author
Zachary Moriarty	MainPageGui class	Author
Zachary Moriarty	RegistrationSender class	Edited
Kevin Tran	addRoom class	Author
Kevin Tran	addPage class	Author
Kevin Tran	addType class	Author

Bohdan Ivchenko	ItemInfo class	Author
Bohdan Ivchenko	Test classes	Author
Kemeria Mustfa	RegisterPage class	Author
Kemeria Mustfa	RegistrationSender class	Author

Testing:

We wrote tests for each class with junit:

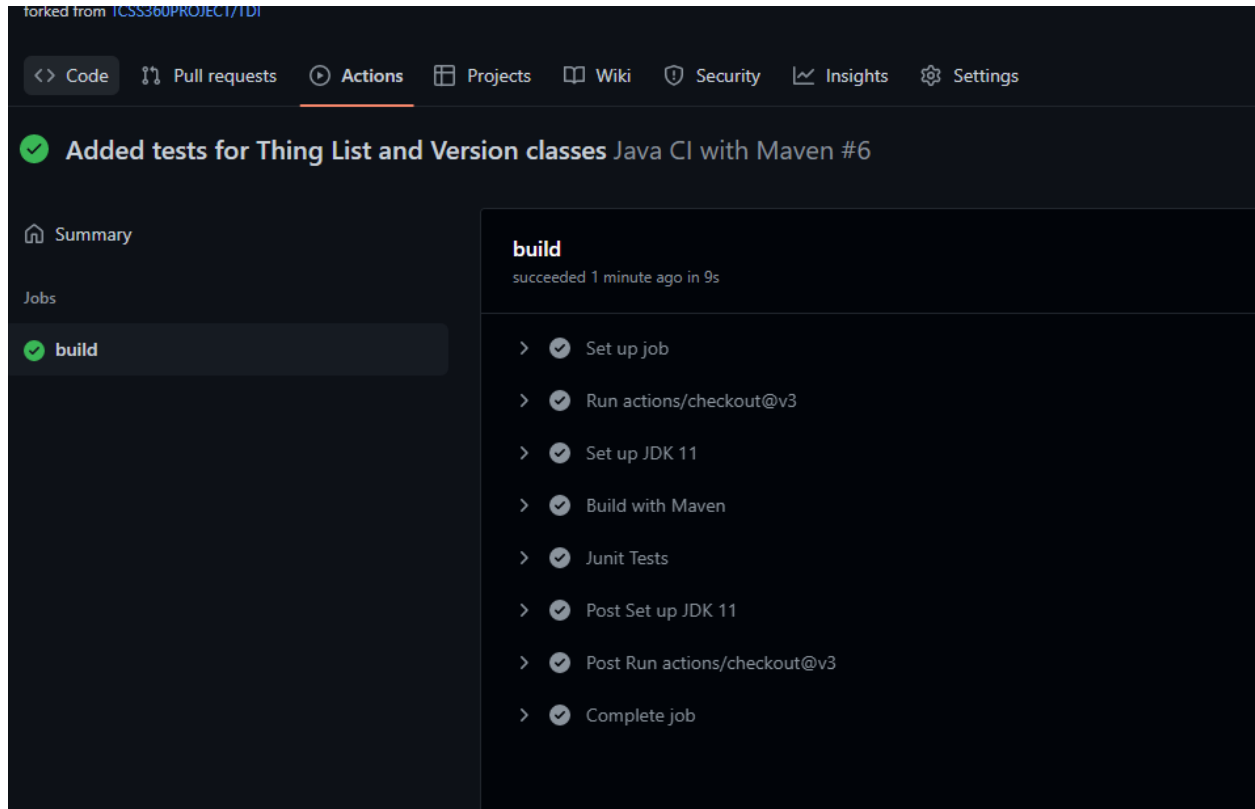


This is what our test coverage looked like:

Coverage: All in IDE			
Element ▲	Class, %	Method, %	Line, %
▼ all	93% (108/116)	73% (206/282)	72% (762/1056)
AboutPage	100% (1/1)	100% (1/1)	100% (9/9)
AboutPageTest	100% (1/1)	100% (1/1)	100% (3/3)
AddPage	71% (5/7)	38% (5/13)	48% (23/47)
AddPageTest	100% (1/1)	100% (2/2)	100% (15/15)
addRoom	100% (4/4)	40% (4/10)	70% (14/20)
AddRoomTest	100% (1/1)	100% (1/1)	100% (3/3)
addType	100% (4/4)	40% (4/10)	70% (14/20)
AddTypeGUI	0% (0/1)	100% (0/0)	100% (0/0)
AddTypeTest	100% (1/1)	100% (1/1)	100% (3/3)
ItemInfo	100% (5/5)	55% (5/9)	44% (21/47)
ItemInfoTest	100% (1/1)	100% (2/2)	100% (15/15)
LoginPage	100% (4/4)	57% (4/7)	37% (12/32)
LoginPageTest	100% (1/1)	100% (1/1)	100% (3/3)
Main	0% (0/1)	0% (0/1)	0% (0/1)
MainPageGUI	100% (7/7)	62% (10/16)	72% (63/87)
MainPageGUITest	100% (1/1)	100% (1/1)	100% (3/3)
RegisterPage	100% (3/3)	83% (5/6)	86% (37/43)
RegisterPageTest	100% (1/1)	100% (1/1)	100% (4/4)
RegistrationSender	100% (3/3)	60% (3/5)	69% (9/13)
RegistrationSenderTest	100% (1/1)	100% (1/1)	100% (4/4)
SortByName	100% (1/1)	100% (1/1)	100% (2/2)
SortByRoom	100% (1/1)	100% (1/1)	100% (2/2)
SortByType	100% (1/1)	100% (1/1)	100% (2/2)
Thing	100% (1/1)	100% (10/10)	100% (14/14)
ThingList	100% (1/1)	92% (13/14)	46% (26/56)
ThingListTest	100% (1/1)	100% (13/13)	100% (59/59)
ThingTest	100% (1/1)	100% (10/10)	100% (18/18)
Version	100% (1/1)	100% (1/1)	100% (1/1)
VersionTest	100% (1/1)	100% (1/1)	100% (2/2)

It is sometimes hard to test ui functions with Junit.

We have also used Github actions to automate the process every time we pushed. I did this on my account:



Source Catalog

Json-simple.jar

Src

MainPageGUI.java

AboutPage.java

Version.java

Thing.java

addPage.java

AddRoom.java

ThingList.java

AddType.java

ItemInfo.java
LoginPage.java
RegisterPage.java
RegisterSender.java

Test
contains all JUnit tests.

Emergency Contacts:

Name	Email
Kevin Tran	Ktran253@uw.edu
Zachary Moriarty	cevis906@uw.edu
Bohdan Ivchenko	ibohdan@uw.edu
Kemeria Mustfa	kemerm@uw.edu