

“SW Engineering CSC648/848 Section 4”

DrillShare

Rent tools owned by anyone

Section 4 Team 1:

Tyler Heslop (Team Lead) - theslop@mail.sfsu.edu

Trevor Henrich (Front-End Lead)

Saksham Manandhar (Back-End Lead)

Kevin Karmacharya (Scrum Master)

Arsany Attalla (Github Master)

“Milestone 4”

Date of Submission: May 2, 2022

History Table

Revision ID	Revision Date	Revised By

1) Product (Drillshare) Summary

General contractors and homeowners alike often take one-off remodels that require expensive and specific tools that are outside of their budget. Brick and mortar tool rental shops typically lack the infrastructure for online orders, so customers often find that their local shop is out of stock only after they arrive. And with COVID-19 still in action, the demand for tools is high, however, the supply is limited. To make matters worse, what happens when you go to pick up your rental only to find that the table saw you need doesn't fit in your Toyota pickup? Looks like you're adding a truck to the cost of the rental. With so many add-ons and unknowns, the customer's time and money could have been used to effectively finish their project faster, but instead find themselves hunting for out of stock items. Imagine if there was a product that could save a customer's time and money, enabling them to get back to what they do best. That's where **Drillshare** comes in. **Drillshare** is an app that makes it easier not only for general contractors and homeowners, but anyone to locally source affordable tools for the duration of their project.

Drillshare aims to simplify the logistics and cost associated with tool rental, while being simple and easy to use. DrillShare allows users to rent tools from one another as well as put their tools up for rent at hourly or daily rates, all in one account. Users who put their tools up for rent have the option of offering delivery to the renter or an address of their choosing, such as a job site. Typically when a contractor owns large or bulky equipment, they have the vehicle necessary to transport it. Owners can charge renters a fee not only to borrow the tool but to deliver and pickup as well. Renters and owners will be able to check their order history to make sure that their transactions are in progress or completed. Unlike typical brick and mortar stores who have limited stock, Drillshare maximizes availability of tools by allowing customers to view stock of all tool owners near them. Unlike the Home Depot, a popular equipment store and online store, we want to give customers an easy experience to search and filter rental tools they want at the price they can afford, while giving them greater availability and an ease of mind when viewing transactions. DrillShare seeks to provide the infrastructure to do just that. DrillShare has a variety of choices like *Airbnb* with the convenience of *Amazon Prime*.

Major Committed Functions:

1. Login
2. Registration
3. List an item (Create an item for rent)
4. Rent an item
5. Return an item
6. Delete an item (Take the item off the site)
7. Searching for an item

URL: <https://drillshare.us/>

2) QA testing

Our focus for our user story will be on Chad Muska, a self made entrepreneur, who wants to rent out his tools to other people, and George Jetson, an average person, who wants to rent a tool for his backyard project. We are focusing on how users are able to create a tool, delete a tool when they no longer want to rent out their tools, rent an item, return an item and search for an item. We are using selenium to test our overall web application along with the frontend routes and GET requests.

a) Unit Test

- Creating an Item
 - Testing to see if functions in create can successful create an item, passing it into the backend, storing the created item into the database, and showing on homepage
 - Passed test
- Deleting an Item
 - Test if newly created item is deleted, with delete function, from the database and not showing on the front end
 - Passed test
- Renting an Item
 - Test if function allows user to rent out an item and updates the status to the backend saying item rented
 - Passed test
- Returning an Item
 - Test if function allows user to return the item they rented out and make the status available to rent again on the backend
 - Passed test
- Searching for an item
 - Test if search function gets the item from the backend and shows on front end
 - Passed test

b) Integration Test

To run selenium tests:
 pip3 install seleniumbase; cd /tests/
 pytest

Test Case ID	TC_CREATE_1	Test Case Description	Test the Create Functionality on Drillshare		
Created By	Kevin	Reviewed By	Tyler	Version	1.0

Tester's Name	Tyler	Date Tested	May-01-2022	Test Case (Pass/Fail/Not	Pass
---------------	-------	-------------	-------------	--------------------------	------

S #	Prerequisites:
1	Access to Chrome Browser
2	
3	

S #	Test Data Requirement
1	Userid = kevink
2	Pass = drillshare777
3	Tool to create = Test Create Form

Test Conditions Upon successful logging with valid userid and password, the user can create an item

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to https://drillshare.us/	Site should open	as expected	Pass
2	Select Profile Icon	A dropdown appears	as expected	Pass
3	Select Login	Login form should open	as expected	Pass
4	Enter Userid & Password	Credential can be entered	as expected	Pass
5	Click Login	User is logged in, navigating back to	as expected	Pass
6	Select Profile Icon	A dropdown appears	as expected	Pass
7	Select Create	Create form should open	as expected	Pass
8	Enter item details	Item details can be entered	as expected	Pass
9	Click Create Listing	The listing is created	as expected	Pass

Test Case ID	TC_SEARCH_1	Test Case Description	Test the Search Functionality on Drillshare		
Created By	Kevin	Reviewed By	Tyler	Version	1.0

Tester's Name	Tyler	Date Tested	May-01-2022	Test Case (Pass/Fail/Not	Pass
---------------	-------	-------------	-------------	--------------------------	------

S #	Prerequisites:
1	Access to Chrome Browser

S #	Test Data Requirement
1	Tool to search = Test Create Form

Test Conditions Upon navigating to Drillshare the user can search for an item

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to https://drillshare.us/	Site should open	as expected	Pass
2	Search for an item	At least one item should show	as expected	Pass

Test Case ID	TC_RENT_1	Test Case Description	Test the Rent Functionality on Drillshare		
Created By	Kevin	Reviewed By	Tyler	Version	1.0

Tester's Name	Tyler	Date Tested	May-01-2022	Test Case (Pass/Fail/Not	Pass
---------------	-------	-------------	-------------	--------------------------	------

S #	Prerequisites:
1	Access to Chrome Browser
2	
3	

S #	Test Data Requirement
1	Userid = tyler
2	Pass = strongPassword
3	Tool to rent = Test Create Form

Test Conditions Upon successful logging with valid userid and password, the user can rent an item

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to https://drillshare.us/	Site should open	as expected	Pass
2	Select Profile Icon	A dropdown appears	as expected	Pass
3	Select Login	Login form should open	as expected	Pass
4	Enter Userid & Password	Credential can be entered	as expected	Pass
5	Click Login	User is logged in, navigating back to	as expected	Pass
6	Select an item	Item details should open	as expected	Pass

7	Select date and rate	Date and rate for rented item can be	as expected	Pass
8	Click Rent Now	Checkout Page should open	as expected	Pass
9	Click Checkout	Item will be rented.	as expected	Pass

Test Case ID	TC_RETURN_1	Test Case Description	Test the Return Functionality on Drillshare		
Created By	Kevin	Reviewed By	Tyler	Version	1.0

Tester's Name	Tyler	Date Tested	May-01-2022	Test Case (Pass/Fail/Not)	Pass
----------------------	-------	--------------------	-------------	----------------------------------	------

S #	Prerequisites:
1	Access to Chrome Browser
2	
3	

S #	Test Data Requirement
1	Userid = tyler
2	Pass = strongPassword
3	Tool to return = Test Create Form

Test Conditions Upon successful logging with valid userid and password, the user can return an item they have rented.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to https://drillshare.us/	Site should open	as expected	Pass
2	Select Profile Icon	A dropdown appears	as expected	Pass
3	Select Login	Login form should open	as expected	Pass
4	Enter Userid & Password	Credential can be entered	as expected	Pass
5	Click Login	User is logged in, navigating back to	as expected	Pass
6	Select Profile Icon	A dropdown appears	as expected	Pass
7	Select Profile	Profile page should open	as expected	Pass
8	Select Item to return under Rented	Item details will open	as expected	Pass
9	Click Return	A popout will show	as expected	Pass
10	Click Yes	Item will be returned	as expected	Pass

Test Case ID	TC_DELETE_1	Test Case Description	Test the Delete Functionality on Drillshare		
Created By	Kevin	Reviewed By	Tyler	Version	1.0

Tester's Name	Tyler	Date Tested	May-01-2022	Test Case (Pass/Fail/Not)	Pass
----------------------	-------	--------------------	-------------	----------------------------------	------

S #	Prerequisites:
1	Access to Chrome Browser
2	
3	

S #	Test Data Requirement
1	Userid = kevink
2	Pass = drillshare777
3	Tool to delete = Test Create Form

Test Conditions Upon successful logging with valid userid and password, the user can delete an item they have created.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to https://drillshare.us/	Site should open	as expected	Pass
2	Select Profile Icon	A dropdown appears	as expected	Pass
3	Select Login	Login form should open	as expected	Pass
4	Enter Userid & Password	Credential can be entered	as expected	Pass
5	Click Login	User is logged in, navigating back to	as expected	Pass
6	Select Profile Icon	A dropdown appears	as expected	Pass
7	Select Profile	Profile page should open	as expected	Pass
8	Select Item to delete under Owned	Item details will open	as expected	Pass
9	Click Delete Item	A popout will show	as expected	Pass
10	Click Yes, Confirm	Item will be deleted navigating back to	as expected	Pass

3) Code Review:

a) Coding Style

- i) The coding style our group has chosen is the [Google JavaScript Style Guide](#). We chose this coding style because we already had some parts of the Google JavaScript Style implemented, making the transition easier to the Google JavaScript Style Guide. In order to enforce the Google JavaScript Style, we are using the [ESLint](#) extension on VSCode such that everyone's code formatting is consistent with one another, making it easier to read, change, and test our code.

b) Submit ~10 Github pull requests

1. <https://github.com/CSC-648-SFSU/-csc648-spring22-01-Team01/pull/4>
2. <https://github.com/CSC-648-SFSU/-csc648-spring22-01-Team01/pull/6>
3. <https://github.com/CSC-648-SFSU/-csc648-spring22-01-Team01/pull/7>
4. <https://github.com/CSC-648-SFSU/-csc648-spring22-01-Team01/pull/8>
5. <https://github.com/CSC-648-SFSU/-csc648-spring22-01-Team01/pull/9>
6. <https://github.com/CSC-648-SFSU/-csc648-spring22-01-Team01/pull/11>
7. <https://github.com/CSC-648-SFSU/-csc648-spring22-01-Team01/pull/12>
8. <https://github.com/CSC-648-SFSU/-csc648-spring22-01-Team01/pull/16>
9. <https://github.com/CSC-648-SFSU/-csc648-spring22-01-Team01/pull/18>
10. <https://github.com/CSC-648-SFSU/-csc648-spring22-01-Team01/pull/21>

4) Self-check: Adherence to original Non-functional specs

- a) **ON TRACK | Platform Compatibility (Priority 1)** - Our application delivery for Drillshare will target desktop users so that it can work on any browser, including Google Chrome, Safari, Mozilla Firefox and more with the exception of Internet Explorer; however, Drillshare will also be responsive such that Drillshare works on mobile devices as well.
 - i) **ON TRACK | Internet explorer**
 - ii) **DONE | Safari**
 - iii) **ON TRACK | Firefox**
 - iv) **DONE | Chrome**
- b) **ON TRACK | Availability (Priority 1)** - The code on the master branch on Github shall be maintained and guaranteed to be working at time.
- c) **DONE | Storage (Priority 1)** - Data will be stored using MongoDB on our AWS EC2 server.
- d) **DONE | Security (Priority 1)**- User credit card data must be stored in a secure, salted database.
- e) **DONE | Security (Priority 1)**- User transactions shall be executed using a secure internet connection (HTTPS).
- f) **DONE | Usability (Priority 1)**- Application will be simple and efficient to use by all users, regardless of experience, by taking feedback from users' input so that tasks such as searching / renting / returning / posting an item will be in 3 - 4 steps .