

## SW Engineering CSC648/848

Application Name: DestiGo

Section 04, Team 06

Name	Role
Sahej Tuli	Team Lead
Ray Dela Cruz	Front-End Lead
Ryan Tong	Back-End Lead
Navjot Singh	Git Master
Faheemah Shaikh	Scrum Master
Jessica Christine Rosero	Full-Stack Developer

### Milestone 4

December 26, 2023

Revision Table			
Revision	Date	Author(s)	Description of Changes
1	09/27/2023	Team Members	Initial Draft
2	10/11/2023	Team Members	Milestone 2
3	10/29/2023	Team Members	<ul style="list-style-type: none"><li>Milestone 3</li><li>Team member(s) role changed.</li></ul> <p><b>Ryan Tong:</b> Back-end Lead <b>Jessica Christine Rosero:</b> Full-Stack Developer</p>
4	11/12/2023	Team Members	Milestone 4

## 1. Unit Tests

- Authentication

### P1 Features covered: Login & Register

These unit tests check if the registration and login are working.

Location: </backend/tests/unit/authController.test.js>

```
• Sahejs-MBP:backend sahejtuli$ npm test
```

```
> backend@1.0.0 test
```

```
> mocha tests/**/*.test.js
```

```
Auth Controller
```

```
POST /register
```

- ✓ Create new user and return a success message!
- ✓ should return a 400 status if the email already exists
- ✓ should return a 500 status if an error occurs

```
POST /login
```

- ✓ should login a user and return user details with a token
- ✓ should return a 401 status if the email or password is invalid
- ✓ should return a 500 status if an error occurs

```
6 passing (3ms)
```

- Blog

### P1 Features covered: Fetch Blogs, Featured Blogs, Create Blog, Update Blog, Like Blog, Delete Blog.

These unit tests check if Fetching Blog data, featuring blog posts, creating blog posts, updating (editing) blog posts, liking and deleting them are working perfectly.

Location: </backend/tests/unit/blogController.test.js>

```
Blog Controller
```

```
GET /blogs
```

- ✓ should return a list of blogs
- ✓ should handle errors and return a 500 status

```
GET /find/:id
```

- ✓ should return a specific blog by ID and update views
- ✓ should handle errors and return a 500 status

```
GET /featured
```

- ✓ should return a list of featured blogs
- ✓ should handle errors and return a 500 status

```
POST /
```

- ✓ should create a new blog
- ✓ should handle errors and return a 500 status

```
PUT /updateBlog/:id
```

- ✓ should update a specific blog by ID
- ✓ should handle errors and return a 500 status

```
PUT /likeBlog/:id
```

- ✓ should like or unlike a specific blog by ID
- ✓ should handle errors and return a 500 status

```
DELETE /deleteBlog/:id
```

- ✓ should delete a specific blog by ID
- ✓ should handle errors and return a 500 status

## 2. Integration Tests

### I. Chat:

Test Case ID	DG_001	Test Case Description:	Test the Chat Functionality Between Friends			
Created By	Ray	Reviewed By:	Ryan	Version		
QA Tester's Log						
Tester's Name	Ray	Date Tested	12/3/23	Test Case (Pass/Fail/Not)	Pass	
S#	Prerequisites:		S#	Test Data		
1	Access to Browser		1	FriendsUsername = ryan		
2	User is logged in		2	Message = "Hello"		
3	Know another user's username		3			
4			4			
5			5			
Test Scenario						
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended		
1	After login in navigate to home page: <a href="https://destigo.vercel.app">https://destigo.vercel.app</a>	Open the home page	As Expected	Pass		
2	Click you username on the top right corner of the page	A list of options will appear	As Expected	Pass		
3	Click Friends	A friends modal will appear	As Expected	Pass		
4	Use Searchbar Under Add Friend and search for FriendsUsername	Their name should appear below with a plus sign to the right of it	As Expected	Pass		
5	Click the plus sign to the right of your FriendsUsername	plus sign changes to a check mark and their name is on the top if the modal indicating you are now friends	As Expected	Pass		
6	Repeat step 1 and 2	Go back to list of options	As Expected	Pass		
7	Click Messages	A modal will appear with your FriendsUsername	As Expected	Pass		
8	Click on FriendsUsername	Open chat with selected Friend	As Expected	Pass		
9	Enter Message	Message can be entered	As Expected	Pass		
10	Click Send	Message is sent and displayed	As Expected	Pass		
11	Have Friend message you back with the same steps	Message on the right is you Message on the left is friend	As Expected	Pass		

### II. Create Post:

Test Case ID	DG_002	Test Case Description:	Create Post in Communities			
Created By	Sahej	Reviewed By:	Ray	Version		
QA Tester's Log						
Tester's Name	Sahej	Date Tested	12/3	Test Case (Pass/Fail/Not)	Pass	
S#	Prerequisites:		S#	Test Data		
1	Access to browser		1	userid: ray		
2	User is authenticated		2	title: Canada		
3			2	desc: Lake		
4			3	photo: <a href="http://res.cloudinary.com/destigo/image/upload/v1701658555/fnhcbarmtzvsu2m7biyg.jpg">http://res.cloudinary.com/destigo/image/upload/v1701658555/fnhcbarmtzvsu2m7biyg.jpg</a>		
			4	category: Events		
Test Scenario User should be logged in and authenticated						
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended		
1	Navigate to: <a href="https://destigo.vercel.app/create">https://destigo.vercel.app/create</a>	Create post page	As Expected	Pass		
2	Enter title, desc	Information can be entered	As Expected	Pass		
3	Click image and choose image	Image to post can be entered	As Expected	Pass		
4	Click Submit	Post is added to Events	As Expected	Pass		

### III. Like Post:

<b>Test Case ID</b>	DG_003	<b>Test Case Description:</b>	Like and Dislike Blog Post			
<b>Created By</b>	Sahej	<b>Reviewed By:</b>	Ray	<b>Version</b>		
<b>QA Tester's Log</b>						
<b>Tester's Name</b>	Sahej	<b>Date Tested</b>	12/3	<b>Test Case (Pass/Fail/Not)</b>	Pass	
<b>S#</b>	<b>Prerequisites:</b>		<b>S#</b>	<b>Test Data</b>		
1	Access to browser		1	blogId: 656d429d727c1efd88cd3f00		
2	User is authenticated					
2	Post exists					
3						
4						
<b>Test Scenario</b> User should be logged in and authenticated and on a post.						
<b>Step #</b>	<b>Step Details</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail/Not Executed/Suspended</b>		
1	Navigate to: <a href="https://destigo.vercel.app/blogDetails/656d429d727c1efd88cd3f00">https://destigo.vercel.app/blogDetails/656d429d727c1efd88cd3f00</a>	Navigate to Blog Post	As Expected	Pass		
2	Click like button	Likes post	As Expected	Pass		
3	Click like button	Unlikes post	As Expected	Pass		

#### IV. Delete Post:

<b>Test Case ID</b>	DG_004	<b>Test Case Description:</b>	Delete Blog Post			
<b>Created By</b>	Sahej	<b>Reviewed By:</b>	Ray	<b>Version</b>		
<b>QA Tester's Log</b>						
<b>Tester's Name</b>	Sahej	<b>Date Tested</b>	12/3/23	<b>Test Case (Pass/Fail/Not)</b>	Pass	
<b>S#</b>	<b>Prerequisites:</b>		<b>S#</b>	<b>Test Data</b>		
1	Access to browser		1	userId: ray		
2	User is authenticated		2	blogId: 656d429d727c1efd88cd3f00		
2	User must be author of blog post		2			
3			3			
4			4			
<b>Test Scenario</b> User should be logged in and authenticated. User is viewing their own blog post.						
<b>Step #</b>	<b>Step Details</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail/Not Executed/Suspended</b>		
1	Navigate to: <a href="https://destigo.vercel.app/blogDetails/656d429d727c1efd88cd3f00">https://destigo.vercel.app/blogDetails/656d429d727c1efd88cd3f00</a>	Navigate to own blog post	As Expected	Pass		
2	Click delete button	Post is deleted	As Expected	Pass		

#### V. Log Out:

<b>Test Case ID</b>	DG_005	<b>Test Case Description:</b>	Log Out			
<b>Created By</b>	Sahej	<b>Reviewed By:</b>	Faheemah	<b>Version</b>	Latest	
<b>QA Tester's Log</b>						
<b>Tester's Name</b>	Faheemah	<b>Date Tested</b>	12/3/23	<b>Test Case (Pass/Fail/Not)</b>	Pass	
<b>S#</b>	<b>Prerequisites:</b>		<b>S#</b>	<b>Test Data</b>		
1	Access to Browser		1	UserId = faheemahashaikh@gmail.com		
2			2	Pass = destigoisgr8		
2			2			
3			3			
4			4			
<b>Test Scenario</b> When a user is logged in but wants to log out						
<b>Step #</b>	<b>Step Details</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass/Fail/Not Executed/Suspended</b>		
1	Navigate to <a href="https://destigo.vercel.app/login">https://destigo.vercel.app/login</a>	Site should open	As expected	Pass		
2	Click on your name on the top right	Panel should open	As expected	Pass		
3	Click log out	User should be logged out	As expected	Pass		

## VI. Edit Post:

Test Case ID	DG_006	Test Case Description:	Edit Blog Post				
Created By	Sahej	Reviewed By:	Ray		Version		
QA Tester's Log							
Tester's Name	Sahej	Date Tested	12/3/23		Test Case (Pass/Fail/Not)	Pass	
S#	Prerequisites:		S#	Test Data			
1	Access to browser		1	userId: ray			
2	User is authenticated		2	title: Canada			
2	User must be author of blog post		2	desc: Lake			
3			3	category: Events			
4			4				
Test Scenario User should be logged in and authenticated. User is viewing their own blog post.							
Step #	Step Details	Expected Results	Actual Results		Pass/Fail/Not Executed/Suspended		
1	Navigate to: <a href="https://destigo.vercel.app/blogDetails/656d429d727c1efd88cd3f00">https://destigo.vercel.app/blogDetails/656d429d727c1efd88cd3f00</a>	Navigate to own blog post	As Expected		Pass		
2	Click edit button	Goes to modify post page	As Expected		Pass		
3	Make modifications	New values entered	As Expected		Pass		
4	Click modify	New values posted on blog	As Expected		Pass		

## VII. Display Events:

Test Case ID	DG_007	Test Case Description:	Test functionality of displaying events on the events page				
Created By	Jessica	Reviewed By:	Jessica		Version	1	
QA Tester's Log							
Tester's Name	Jessica	Date Tested	3-Dec-2023		Test Case (Pass/Fail/Not)	Pass	
S#	Prerequisites:		S#	Test Data			
1	Access to website through browser		1				
2			2				
2			2				
3			3				
4			4				
Test Scenario							
Step #	Step Details	Expected Results	Actual Results		Pass/Fail/Not Executed/Suspended		
1	Navigate to <a href="https://destigo.vercel.app/">https://destigo.vercel.app/</a>	Site should open on Home Page	As expected		pass		
2	Click on Events tab	Should open events page and display all events	As expected		pass		

## VIII. Search Events:

Test Case ID	DG_008	Test Case Description:	Test functionality of search bar in events tab				
Created By	Jessica	Reviewed By:	Jessica		Version	1	
QA Tester's Log							
Tester's Name	Jessica	Date Tested	3-Dec-2023		Test Case (Pass/Fail/Not)	Pass	
S#	Prerequisites:		S#	Test Data			
1	Access to website through browser		1	(Search by location) San Francisco			
2			2	(Search by name) Fair			
2			2				
3			3				
4			4				
Test Scenario							
Step #	Step Details	Expected Results	Actual Results		Pass/Fail/Not Executed/Suspended		
1	Navigate to <a href="https://destigo.vercel.app/">https://destigo.vercel.app/</a>	Site should open on Home Page	As expected		pass		
2	Click on Events tab	Should open events page	As expected		pass		
3	Click on Search bar	should be able to enter text	As expected		pass		
4	Click left side of search bar to toggle to "Search by location"	left side should say "Search by location"	As expected		pass		
5	Enter text for a location	text can be entered, only events at the location are shown	As expected		pass		
6	Click left side of search bar to toggle to "Search by name"	left side should say "Search by name"	As expected		pass		
7	Enter text for the name of an event or keyword	text can be entered, only events with keyword in name are shown	As expected		pass		

IX. Create Account:

Test Case ID	DG_009	Test Case Description:	Test functionality of registering an account		
Created By	Jessica	Reviewed By:	Jessica	Version	1
QA Tester's Log					
Tester's Name	Jessica	Date Tested	3-Dec-2023	Test Case (Pass/Fail/Not)	Pass
S#	Prerequisites:		S#	Test Data	
1	Access to website through browser		1	username icebear	
2			2	email: icebeargos@gmail.com	
2			2	pass: icebear!	
3			3		
4			4		
Test Scenario					
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	
1	Navigate to <a href="https://destigo.vercel.app/">https://destigo.vercel.app/</a>	Site should open on Home Page	As expected	pass	
2	Click Login/Sign Up in top right corner	Should have text boxes to enter credentials	As expected	pass	
3	Click Create An Account	Should reroute to register page	As expected	pass	
4	Fill in required fields	all text boxes can be used to enter text	As expected	pass	
5	Choose file for profile picture	able to choose image and file name appears	As expected	pass	
6	Click Register	all text boxes can be used to enter text	As expected	pass	

X. Log In:

Test Case ID	DQ_010	Test Case Description:	Test functionality of login feature			
Created By	Jessica	Reviewed By:	Jessica			
Version		1				
QA Tester's Log						
Tester's Name		Jessica	Date Tested	3-Dec-2023	Test Case (Pass/Fail/Not)	Pass
S#	Prerequisites:		S#	Test Data		
1	Access to website through browser		1	username: jessica.rosero@gmail.com		
2	Be a registered user		2	pass: jess1234		
2	Be logged into a user account		2			
3			3			
4			4			
Test Scenario						
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended		
1	Navigate to <a href="https://destigo.vercel.app/">https://destigo.vercel.app/</a>	Site should open on Home Page	As expected	pass		
2	Click Login in top right corner	Should have text boxes to enter credentials	As expected	pass		
3	Enter username and password	Credentials can be entered	As expected	pass		
4	Click login	User logged in and rerouted to home page	As expected	pass		

## Analysis of Testing Coverage

In our integration tests, 10 P1 features were tested, and all features passed.

### 3. Coding Practices

- a. Please describe what coding style you chose and how to enforce it by the tool or plug-in.

In our project, we have adhered to the Google JavaScript Style Guide, which provides guidelines for writing clean and maintainable JavaScript code. This coding style emphasizes practices such as consistent indentation (2 spaces), meaningful variables, and function names so it is easier to understand what that piece of code is doing. To enforce the Google JavaScript Style Guide in your project, we have utilized ESLint as an extension to ensure that the coding format stays intact for the project

and allows everyone to follow the same format as they are making the updates to the project. ESLint is a powerful tool that can be configured to enforce coding style rules automatically. In our project, ESLint has been set up to check your JavaScript code against the rules outlined in the Google JavaScript Style Guide for proper readability and logic.

- b. Please list source files related to 5 P1 features that demonstrate your coding style.

*Feature 1: User Registration*

Register.jsx: This file contains the code for the user registration feature. It adheres to the Google JavaScript Style Guide, ensuring clean and readable JavaScript code.

*Feature 2: Search Functionality*

Search.jsx: The JavaScript code for the search functionality follows the Google JavaScript Style Guide's recommendations, emphasizing code readability and maintainability.

*Feature 3: Chat*

Chat.jsx: The JavaScript code for the chat section conforms to the Google JavaScript Style Guide's principles, including clear indentation and naming conventions for improved code quality throughout the entire project.

*Feature 4: Updating the blog*

UpdateBlog.jsx: This JavaScript file for the user profile page aligns with the Google JavaScript Style Guide, maintaining coding standards for modularity and clarity.

*Feature 5: Community Page*

Community.jsx: The JavaScript code for the community page strictly follows the Google JavaScript Style Guide, ensuring consistency and readability across your project.

**Team lead ensures that all members have read and understood this document.**

Yes, 12/03/2023  
- Sahej Tuli