

System and Unit Test Report

Talk to Me by TADAA

12/07/2020

System Test Scenarios

- **Sprint 1**

- **User Stories**

- As a user, I want to be able to be given a private chat room, so I could talk without anyone else seeing.
 - As a user, I want to be able to send and receive messages, so I can talk away my loneliness.

- **Scenarios**

- Private chat room:
 1. Open up a browser.
 2. Navigate to <https://talk-to-me.co/>
 3. Click the 'Talk' option.
 4. Wait until the "Finding room..." dialogue changes to "Connected to another person."
 5. User is now in a private chat room.
 - Sending/receiving messages:
 1. Do steps 1-4 from Private Chat Room scenario.
 2. Type a message into the chat box and hit the 'Send' button.
 3. Wait until you get a reply.
 4. User is able to send and receive messages.

- **Sprint 2**

- **User Stories**

- As a user, I want to be able to match with someone who can hear me vent, so I can vent.
 - As a site admin, I want to show a popup detailing the purpose of the site, so I don't get people trolling.
 - As a user, I want to be able to like messages, so I can show people I like what they're saying.

- **Scenarios**

- Venting:
 1. Open up a browser.
 2. Navigate to <https://talk-to-me.co/>
 3. Click the 'Be Heard' option.
 4. Wait until the "Finding room..." dialogue changes to "Connected to another person."
 5. User is now in a private room with someone who can hear them vent.

- Popup:
 1. Open up a browser with no stored cookies (incognito/private browsing).
 2. Navigate to <https://talk-to-me.co/>.
 3. User should see a popup detailing the purpose of the site.
- Liking messages:
 1. Open up a browser.
 2. Navigate to <https://talk-to-me.co/>
 3. Click any of the three options.
 4. Wait until the “Finding room...” dialogue changes to “Connected to another person.”
 5. Wait until the other person sends a message.
 6. Hover the mouse over the received message, at which point a translucent heart should pop up next to the message.
 7. User can click message to show that they liked it.

● Sprint 3

○ User Stories

- As a user, I want to be able to report people that send offensive messages, so I don't chat with them again
- As a site admin, I want to be able to view reported messages, so I can view reports.
- As a site admin, I want to be able to block IPs/users from chatting, so I can prevent trolls.
- As a user, I want the background of the website to change occasionally, so that it doesn't get stale and continues to look nice.
- As a user, I want to use this website on my phone/tablet/etc., so that I can use it wherever I want.

○ Scenarios

- Reporting people:
 1. Open up a browser.
 2. Navigate to <https://talk-to-me.co/>
 3. Click any of the three options.
 4. Wait until the “Finding room...” dialogue changes to “Connected to another person.”
 5. Click the ‘Report’ button in the top left corner.
 6. Enter the reason for the report into the text field and hit ‘Submit’.
 7. User has reported the person they were chatting with.
- Viewing reported messages:
 1. Open up a browser.
 2. Navigate to the database (cloud.mongodb.com).
 3. View the ‘Reports’ collection.
 4. Admin is able to see reports and reported messages.
- Blocking users:

1. Open up a browser.
 2. Navigate to the database (cloud.mongodb.com)
 3. View the 'banned_users' collection..
 4. Add a new object to the database with the property {'ip': <user's ip>} to the collection.
 5. Admin has banned said user.
 - Changing background:
 1. Open up a browser.
 2. Navigate to <https://talk-to-me.co/>
 3. Note the background.
 4. Refresh the page.
 5. User should see a different background.
 - Mobile view.
 1. Open up a browser on a mobile device.
 2. Navigate to <https://talk-to-me.co/>
 3. User should see a functional and aesthetically pleasing version of the site.
- **Sprint 4**
 - **User Stories**
 - As a site admin, I want to see the reports in a nicely formatted way, so that it is easy to read over the reports (auth admins, admin page).
 - **Scenarios**
 - Viewing reports:
 1. Open up a browser.
 2. Navigate to <https://talk-to-me.co/admin/>
 3. Enter the administrator password.
 4. Click the 'Reports' button.
 5. Admin should be able to see all reports.

Unit Tests

All unit tests were delegated to the DevOps position.

Each tested component is listed below, with the equivalence classes and requisite test cases nested below the component.

- User authorization.
 - Valid request.
 - GET HTTP request to <http://talk-to-me.co/auth>
 - Invalid request.
 - POST HTTP request to <http://talk-to-me.co/auth>
- User enqueue request.
 - Valid request to enqueue in 'vent' queue type.
 - POST HTTP request to <http://talk-to-me.co/queue> with appropriate body.
 - Valid request to enqueue in 'listen' queue type.

- POST HTTP request to <http://talk-to-me.co/queue> with appropriate body.
 - Valid request to enqueue in 'talk' queue type.
 - POST HTTP request to <http://talk-to-me.co/queue> with appropriate body.
 - Invalid request to enqueue.
 - POST HTTP request to <http://talk-to-me.co/queue> missing user secret.
 - POST HTTP request to <http://talk-to-me.co/queue> missing queue type.
 - GET HTTP request to <http://talk-to-me.co/queue>.
- User matchmaking.
 - Matching two users in talk.
 - Tests that the two users both complete queue.
 - Matching someone in vent with someone in listen.
 - Tests that the two users both complete queue.
- User messaging.
 - User can send messages.
 - POST HTTP request to <http://talk-to-me.co/messages>.
 - User can receive messages.
 - Socket.io 'send_message_to_client' event.