# Test Cases

## Project: Vox Realtime Groupchat App

### 1. Test Title: Create Account

#### Executed by:

#### SRS Section Id: 3.1

* Description: To login into the application, a user must be logged in. To do that, they must have an existing account by creating one.
* Preconditions: None
* Dependencies: If valid, account information gets saved in to the database.

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Click on register button | Any alphabets, 0-9, or . | Account registered |  |  |  |
| 2 | Click on register button | Register an account. Then, try to register the account again | An error message saying that user or password exists |  |  |  |

### 2. Test Title: User Login

#### Executed by:

#### SRS Section Id: 3.2

* Description: To access the application features, a must be log in.
* Preconditions: An account must exist
* Dependencies: Verify the log in info with the one in database. On success, user gets access to the application features. Otherwise, a user must create an account to use the application.

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Click on login button | Enter an account information that has been registered successfully | Login successful |  |  |  |
| 2 | Click on login button | Enter any account information that has not been registered already | Login unsuccessful |  |  |  |

### 3. Test Title: Create Vox Server

#### Executed by:

#### SRS Section Id: 3.3

* Description: To interact with another user, they both need to be in the same server. On way to do that is to create a vox server.
* Preconditions: User must be logged in.
* Dependencies: Database for holding server entries.

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Login in with a valid account |  |  |  |  |  |
| 2 | Click on create server | Enter a name for the server | Server created |  |  |  |

### 4. Test Title: Chat on Servers

#### Executed by:

#### SRS Section Id: 3.4

* Description: If you need to interact with other users, you can do that by messaging them in the chat area.
* Preconditions: Must be/have an server
* Dependencies: Database for holding server entries.

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Login in with a valid account |  |  |  |  |  |
| 2 | Click on an existing server |  |  |  |  |  |
| 3 | Click on the textbox | Enter any text | Message is sent and appears in the box above |  |  |  |
| 4 | Ask your friend in the same server to send you a message |  | Message appears on the box above |  |  |  |

### 5. Test Title: View/Edit Notebook

#### Executed by:

#### SRS Section Id: 3.5.1

* Description: A user may create a notebook for each server they are in. The contents should be able to be saved manually or when they close the notebook.
* Preconditions: Must have an account and be part of a server
* Dependencies: Database for holding notebook entries

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Open notebook in server | User credentials and DB | The user should recieve their notes from the db if any |  |  |  |
| 2 | Edit notebook and make changes if needed |  |  |  |  | This step is optional since they may not have changes to make |
| 3 | Close notebook | Notebook entry will be sent to DB |  |  |  | PUT (Update or Create) the notebook entry |

### 6. Test Title: View Shared Notebook

#### Executed by:

#### SRS Section Id: 3.5.1

* Description: A user of a server may share their notebook with other users of the same server. This would be viewable in a column that would show who all in the server shares their notes and the user would be able to choose any of them.
* Preconditions: A user must be part of a server and another user must have shared their notebook
* Dependencies: Database for holding notebook entries

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Open shared notebook in server | User credentials and DB | The user should receive a copy of another person’s notes |  |  | This may be outdated if the other user is updating the notes, this is intended |
| 2 | View notes |  |  |  |  | READ ONLY on other user’s notes |
| 3 | Close notes |  |  |  |  | Does not update entry |

### 7. Test Title: Share Notebook with Server

#### Executed by:

#### SRS Section Id: 3.5.2

* Description: A user of a server may share their notebook with other users of the same server.
* Preconditions: A user must be part of a server
* Dependencies: Database for holding notebook entries

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Open notebook in server | User credentials and DB | To be able to see their notebook |  |  |  |
| 2 | Enable sharing |  | User checks option to enable sharing on notebook |  |  |  |
| 3 | Close notes |  | Notebook is updated in DB to enable sharing |  |  |  |

### 8. Test Title: Create Event

#### Executed by:

#### SRS Section Id: 3.6.1

* Description: If you instructor gives you an important due date for a project. You would probably want to mark that in a calendar.
* Preconditions: Must have an existing account
* Dependencies: Database for holding calendar events

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Login in with a valid account |  |  |  |  |  |
| 2 | Hover over the day when your assignment is due |  |  |  |  |  |
| 3 | Click on add event button | Enter a description | Event gets saved in the calendar |  |  |  |

### 9. Test Title: Remove Event

#### Executed by:

#### SRS Section Id: 3.6.2

* Description: If your assignments due date has passed and you don’t want that event to exist in your calendar. It can be removed from the calendar.
* Preconditions: Must have an existing account
* Dependencies: Database for holding the calendar events

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Login in with a valid account |  |  |  |  |  |
| 2 | Hover over that event in your calendar |  |  |  |  |  |
| 3 | Click on delete button |  | Event is deleted from the calendar |  |  |  |

### 10. Test Title: Remove Users

#### Executed by:

#### SRS Section Id: 3.7

* Description: There should be the ability to remove a user from a server and possibly prevent them from being able to rejoin.
* Preconditions: Have users in a server other than the owner and be the owner of the server
* Dependencies: Database containing the server information

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Go to server | User credentials and DB | To see the users of a server |  |  |  |
| 2 | Select user and click remove |  | The user will get a confirmation on removing the user |  |  |  |
| 3 | Confirm removal |  | The user will no longer appear in the server |  |  |  |

### 11. Test Title: Delete Server

#### Executed by:

#### SRS Section Id: 3.8

* Description: If you do not want any of your own created to exist. The server can simple be deleted.
* Preconditions: Server must exist
* Dependencies: Database for holding the server information and it’s users information

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Login into your account |  |  |  |  |  |
| 2 | Click on a server that you created |  |  |  |  |  |
| 3 | Click on the delete button |  | Server is deleted |  |  |  |

### 12. Test Title: Invite User to server

#### Executed by:

#### SRS Section Id: 3.9

* Description: If you want to invite your friend to your gaming server, and talk about new games. This can simple be done by clicking on invite user in your server room.
* Preconditions: A user must have their own server created.
* Dependencies: Database for holding the server information and your friends information.

| Step # | Test Step | Data | Expected | Actual | Status | Notes |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Login into your account |  |  |  |  |  |
| 2 | Click on a server that you created |  |  |  |  |  |
| 3 | Click on invite button | Enter the username of the new user you want to invite | User is added to the server |  |  |  |

## Non functional Requirements

### 13. SSL Certificate

* This will be tested by opening the web page using HTTPS, if this doesn’t work then we know we do not have a SSL certificate for the web server.

### 14. Response Time of < 500ms for Chat Messages

* This will be tested by running the web app on two computers and checking the time it is sent to the time it is shown on the other computer. This is to check we are efficiently dealing with database inserts and that web sockets are functioning.