# **Test Plan**

# Overview

Testing for our application will be straightforward. Test cases will attempt to span all functionality over the database, backend, and frontend sections. For the database, we will be testing create, read, update, and delete functions, to assure that tables and relations store and update information correctly. For the backend, along with testing the same features as the database to assure that the application can connect with the database, we will also be testing the algorithms to process data. Lastly, for the frontend, we will be testing components to confirm that visuals, bindings, and functions are being called correctly from the user interface. In essence, we will perform end-to-end testing.

# **Test Cases**

#### DB1

- Purpose: Create database entry
- Description
  - Adds an item that isn't currently in the database
- Inputs: New item name and Quantity
- Outputs: Updated database entry
- Blackbox
- Functional
- Unit

#### DB2

- Purpose: Read database entry(ies)
- Description
  - Returns the current list of items and quantities from the database
- Inputs: Item(s) to be read
- Outputs: Current state of the database
- Blackbox
- Functional
- Unit

## DB3

- Purpose: Update database entry
- Description
  - Update the quantity of an item that is currently in the database
- Inputs: Item and quantity

- Outputs: Updated database entry
- Blackbox
- Functional
- Unit

#### DB4

- Purpose: Delete database entry
- Description
  - Removes item from database, regardless of quantity (stop tracking item)
- Inputs: Item to be removed
- Outputs: Updated database with missing entry
- Blackbox
- Functional
- Unit

#### BE1

- Purpose: Run API call to create database entry
- Description
  - API call that creates a new database entry
- Inputs: New item name and quantity
- Outputs: Updated database entry
- Blackbox
- Functional
- Integration

# BE2

- Purpose: Run API call to read database entry
- Description
  - API call that returns the current list of items and quantities from the database
- Inputs: Item(s) to be read
- Outputs: Current state of the database
- Blackbox
- Functional
- Integration

#### BE3

- Purpose: Run API call to update database entry
- Description:

- Unit testing to make calls from the API to the database to update entry(ies)
- Inputs: Get test table in database, update a certain item(s)
- Outputs: Table is updated correctly.
- Blackbox
- Functional
- Integration

### BE4

- Purpose: Run API call to delete database entry
- Description:
  - Unit testing to make calls from the API to the database to delete entry(ies) in the database
- Inputs: Get test table in database, make delete call for certain item(s)
- Outputs: Database table is updated correctly
- Blackbox
- Functional
- Integration

#### FE1

- Purpose: Confirm landing page renders
- Description:
  - Check that:
    - All parts and components of the page render
    - Interactive and input components react to user
    - Users cannot access any information before signing in
- Inputs: Start rendering of page
- Outputs: Page renders, user can interact with page, and no info is shown if logged out
- Blackbox
- Functional
- Unit

# FE2

- Purpose: Confirm login page renders
- Description:
  - Check that:
    - All parts and components render correctly
    - Interactive user inputs can take in data

- User can log in (i.e. get redirected to home page when success logging in)
- If user is cached as logged in, that the page redirects to the home page when coming from landing page
- Inputs: Start rendering of page
- Outputs:
- Blackbox
- Functional
- Unit

#### FE3

- Purpose: Confirm home page renders
- Description:
  - Check that:
    - All parts and components render correctly
    - User can create rooms/inventories
    - User can add/create items to add to inventories
    - User can read items in inventories
    - User can update items in inventories
    - User can delete items in inventories
- Inputs: User home page renders and automated actions are performed
- Outputs: Page(s) render correctly and "user" can interact with inventories and items correctly
- Blackbox
- Functional
- Integration

#### FE4

- Purpose: Confirm add button component calls API call to create
- Description:
  - When a user clicks the add button, they should be prompted with a dialogue to input item information
  - Inputs in dialogue should be interactable
  - When clicking submit, inventory(ies) should be updated
- Inputs: Dialogue renders and automated user attempts to fill out the form
- Outputs: Dialogue renders, fields are filled out, and inventory is updated with new item
- Blackbox
- Functional
- Integration

# FE5

- Purpose: Confirm add button component calls API call to read
- Description:
  - When home page is loaded, correct inventory(ies) with corresponding items are rendered correctly
- Inputs: API read call is made
- Outputs: Page renders, correct inventory(ies) and items are rendered
- Blackbox
- Functional
- Integration

#### FE6

- Purpose: Confirm add button component calls API call to update
- Description:
  - When home page is rendered, user attempts to edit an existing item
- Inputs: Page renders, automated user attempts to edit and fill out the form, then submits
- Outputs: Item(s) are updated correctly
- Blackbox
- Functional
- Integration

#### FE7

- Purpose: Confirm add button component calls API call to delete
- Description
- Inputs: Page renders, user attempts to delete an existing item
- Outputs: Inventory(ies) are updated correctly
- Blackbox
- Functional
- Integration

# Test Matrix

Test Case ID	Normal/Abnormal	Blackbox/Whitebox	Functional/Performance	Unit/ Integration
DB1	Normal	Blackbox	Functional	Unit
DB2	Normal	Blackbox	Functional	Unit
DB3	Normal	Blackbox	Functional	Unit
DB4	Normal	Blackbox	Functional	Unit
BE1	Normal	Blackbox	Functional	Integration
BE2	Normal	Blackbox	Functional	Integration
BE3	Normal	Blackbox	Functional	Integration
BE4	Normal	Blackbox	Functional	Integration
BE5	Normal	Whitebox	Functional	Unit
BE6	Normal	Whitebox	Functional	Unit
BE7	Normal	Whitebox	Functional	Unit
FE1	Normal	Blockbox	Functional	Unit
FE2	Normal	Blockbox	Functional	Unit
FE3	Normal	Blockbox	Functional	Integration
FE4	Normal	Blockbox	Functional	Integration
FE5	Normal	Blockbox	Functional	Integration
FE6	Normal	Blockbox	Functional	Integration
FE7	Normal	Blockbox	Functional	Integration