

Use Case 1: Add Inventory Item(s)

Goal: User wants to add item to inventory

Primary Actor: Finders Keepers user

Scope: App

Level: User

Precondition: App is open to inventory page

Success end: Item is added to inventory

Failure end: Item is not added to inventory

Trigger: User pushes add to inventory button

Success Scenario:

1. User pushes add to inventory button
2. Add to inventory screen opens
3. User chooses a main image for their item
4. User adds at least one tag to their item
5. User adds text description to their item
6. User selects 'free' or 'for trade' toggle appropriately
7. User presses 'add to inventory'
8. Item is added to user's inventory, and appears on their inventory screen

Extensions:

- 7a. Not all fields are filled properly
 - i. 'Add to inventory' button is greyed out
- 8a. Device has no connection to the remote server
 - i. Device displays 'could not connect to server' error
 - ii. App returns to add to inventory screen

Alternative Scenarios:

- 3a. User adds additional images to their item
- 7a. User presses 'cancel' button
 - i. User is returned to inventory screen

Use Case 2: Add Wishlist Item(s)

Goal: User wants to add item to wishlist

Primary Actor: Finders Keepers user

Scope: App

Level: User

Precondition: App is open to wishlist page

Success end: Item is added to wishlist

Failure end: Item is not added to wishlist

Trigger: User pushes add to wishlist button

Success Scenario:

1. User pushes add to wishlist button
2. Add to wishlist screen opens
3. User chooses a number of tags
4. User presses 'add to wishlist' button
5. Tags are added to user's wishlist

Extensions:

- 5a. App cannot contact server
 - i. App displays 'cannot connect to server' error
 - ii. App returns to wishlist screen

Alternative Scenarios:

- 4a. User instead presses 'cancel' button
 - i. User returns to wishlist screen without adding items

Use Case 3: Browse by tag

Goal: User wants to browse items with given tag(s)

Primary Actor: Finders Keepers user

Scope: App

Level: User

Precondition: App is open to the home screen

Success end: Items corresponding to tag are displayed

Failure end: No items are displayed

Trigger: User presses 'browse' button

Success Scenario:

1. User presses 'browse' button
2. Browse screen opens, user presses 'add tags'
3. User adds a number of tags
4. User presses 'search'
5. App contacts remote server, receives items matching selected tags
6. Items are displayed to user

Extensions:

- 5a. No items with matching tags found in the user's area
 - i. App displays 'No items found matching selected tags'
 - ii. App returns to browse screen.

Alternative Scenarios:

None.

Use Case 4: Browse matched users' inventories

Goal: User wants to browse the profile of a matched user

Primary Actor: Finders Keepers user

Scope: App

Level: User

Precondition: App is open to the home screen

Success end: The inventory of a matched user is displayed to the end user.

Failure end: No inventory displayed.

Trigger: User presses 'matches' button.

Success Scenario:

1. User presses 'matches' button
2. App communicates with server, displays matched users
3. Matching profiles displayed to user
4. User presses on a profile
5. Matched user's inventory is displayed to end user

Extensions:

- 2a. No matches found
 - i. App displays 'No matches yet - Try again later!'

Alternative Scenarios:

- 5a. User presses 'view wishlist' button on matched user's inventory
 - i. Matched user's wishlist is displayed

Use Case 5: Propose a trade from a user's inventory

Goal: User wants to trade with another user

Primary Actor: Finders Keepers user

Scope: App

Level: User

Precondition: App is open to another user's inventory

Success end: A new trade offer is created

Failure end: No new trade offer is created

Trigger: User presses 'offer trade' button

Success Scenario:

1. User presses 'offer trade' button
2. The app presents a screen with both user's inventories visible
3. User adds items from each inventory to trade
4. User presses 'offer trade' button
5. Trade offer is sent to user 2

Extensions:

- 3a. One user has no valid items to add to trade
 - i. The 'offer trade' button stays greyed out until at least one item has been added from each side (unless all requested or offered items are free)

Alternative Scenarios:

None.

Use Case 6: Trade with another user from the 'browse' screen

Goal: User wants to trade with a user from the 'browse' screen

Primary Actor: Finders Keepers user

Scope: App

Level: User

Precondition: App is open to the browse screen with items displayed

Success end: User initiates trade with other user

Failure end: Trade is not initiated

Trigger: User presses on an item on the 'browse' screen

Success Scenario:

1. User presses on item
2. Item description window pops up
3. User presses 'view inventory'
4. Inventory screen of other user is displayed
5. User presses 'offer trade' button
6. The app presents a screen with both user's inventories visible
7. User adds items from each inventory to trade
8. User presses 'offer trade' button
9. Trade offer is sent to user 2

Extensions:

- 7a. One user has no valid items to add to trade
 - i. The 'offer trade' button stays greyed out until at least one item has been added from each side (unless all requested or offered items are free)

Alternative Scenarios:

- 4a. User desires to view other user's wishlist before trading
 - i. User presses 'wishlist' button on other user's inventory screen
 - ii. Other user's wishlist is displayed
 - iii. User closes wishlist, returns to step 4

Use Case 7: Manage an existing trade offer

Goal: User wants to accept an existing trade offer

Primary Actor: Finders Keepers user

Scope: App

Level: User

Precondition: App is open to the home screen

Success end: User accepts a trade offer

Failure end: Trade offer cannot be accepted

Trigger: User presses 'offers' on home screen

Success Scenario:

1. User presses 'offers'
2. A list of pending outgoing and incoming offers is displayed
3. User presses on incoming offer
4. Offer details are displayed to user
5. User reviews details
6. User presses accept
7. App provides contact details of other user, and sends push notification to other user that their offer has been accepted.

Extensions:

- 6a. Offer has been canceled
 - i. App displays 'offer canceled'
 - ii. App returns to offers screen

Alternative Scenarios:

- 3a. User instead wishes to modify an outgoing offer
 - i. User selects outgoing offer instead
 - ii. User may now review and/or cancel outgoing offer
- 5a. User wants to review specific item descriptions
 - i. User presses on item in offer detail
 - ii. Item description is displayed on screen
- 6a. User dislikes incoming offer
 - i. User presses 'decline' instead of accept
 - ii. Other user sent push notification that their offer was declined
 - iii. User given option to reply with a counter-offer