

# KanBug Tracker

## Target Audience:

Anybody -> Students -> University Students -> NUS Students -> CS Students -> CS2103/T Students

## Value Proposition:

Providing a command line interface for software engineering students to manage and track the bug fixing process.

KanBug Tracker helps CS2103/T students manage bug details of their projects. It is optimised to work offline in the command line so that frequent tasks can be done remotely, securely and quickly by typing in commands.



A recipe for brainstorming user stories

Step 0: Clear your mind of preconceived product ideas/

Step 1: Define the target user as a persona:

Roger is a university student studying in an IT field who likes to build software in his spare time. As such, he usually works alone using a command line text editor (eg. Vim) and often with an unreliable internet connection at his favourite cafe. Unfortunately Roger is not a very organised person and as such writes down issues in his programme on scraps of paper that go missing.

Step 2: Define the problem scope

KanBug tracker helps Roger keep track of the bug fixing process in his code without the need for a stable internet connection. It does not help him find/solve bugs in his code.

Step 3: List scenarios to form a narrative:

**1. First Use**

- 1.1. Roger got to know about KanBug Tracker. He downloaded it and launched it to check out what it can do
- 1.2. After playing around with it for a bit, Roger appreciates that it is an offline command line tool that fits well into his workflow and thus wants to continue to use it

**2. Second Use**

- 2.1. Roger starts adding the bugs that he encounters while working on the project and realises that he no longer needs to worry about remembering the bugs he has encountered and is currently working on fixing.
- 2.2. He also starts categorising the bugs based on the progress he has made towards fixing them.

**3. 10th use**

- 3.1. Roger reaches the cafe and realises he is unsure of which bug he should fix first. He then opens up the application to refresh his memory on what the most important bugs are that he needs to work on for the day.
- 3.2. After working on the bug for the whole day, he has managed to solve the issue and as such has moved the bug over to the completed tab along with a written description of how he solved it so he can refer to it again in the event where he were to encounter it again.

**4. 100th use**

- 4.1. Roger's code has so many bugs that he is unable to find the specific bug that he is looking for. Being a power user of the application at this point, he makes use of the search functionality to find the bugs by category in order to locate the one he is looking for.
- 4.2. Roger realises that a bug he had identified in the past and marked as being low priority is now greatly hindering his progress and as a result he increases its priority to represent its urgency.

- 4.3. Roger has also finally completed work on his first project but before starting on his next project he decides to archive the bugs he had encountered while working on it so he can reference them in the future if he encounters the same bug again.

Step 4: List the user stories to supports the scenarios:

Roger

1. As a long time user I can check for unfixed bugs that were low in **priority**.
2. As a user with long term projects I can **change** the priority of bugs as the project progresses
3. As a user with limited time to work on the project, I can **edit and reduce** the size of the bug if I'm unable to fix it in one sitting
4. As a user with long term projects I can **delete/archive** to remove clutter from the UI
5. As a new user I can use the **help** function to **learn** more about the app

| Keep                                                                                                                                  | Toss (For v1.2)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"><li>1. As a long time user I can check for unfixed bugs that were low in <b>priority</b>.</li></ol> | <ol style="list-style-type: none"><li>1. As a user with long term projects I can <b>delete/archive</b> to remove clutter from the UI</li><li>2. As a user with limited time to work on the project, I can <b>edit and reduce</b> the size of the bug if I'm unable to fix it in one sitting</li><li>3. As a user with long term projects I can <b>change</b> the priority of bugs as the project progresses</li><li>4. As a new user I can use the <b>help</b> function to <b>learn</b> more about the app</li></ol> |

## Phong

1. As a forgetful user, I want to note down the **solution** for each bug I solved so that I won't make the same mistake again.
2. As an indecisive user, I want the app to **decide** for me **which bugs to fix** for today so that I don't have to waste much time choosing one.
3. As a user, I can perform most operations on multiple bugs as if performing on one bug (**mass operation**) so that I can update the state of multiple bugs at once.
4. As a user, I can perform **filters** based on different attributes of bugs (such as date created, priority) so that I don't be distracted by bugs that I am not interested in currently.
5. As a user, I can **sort** the list of bugs based on different attributes (such as date created, priority) so that I can see the bugs I am currently interested in first.
6. As a forgetful user, I can freely **customize different commands** (e.g. using "remove" instead of "rm" to remove a bug) so that I don't have to take time to remember the commands when I am still unfamiliar with the app.

| Keep                                                                                                                         | toss                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| As a forgetful user, I want to note down the <b>solution</b> for each bug I solved so that I won't make the same mistake aga | As a forgetful user, I can freely <b>customize different commands</b> (e.g. using "remove" instead of "rm" to remove a bug) so that I don't have to take time to remember the commands when I am still unfamiliar with the app. |
|                                                                                                                              | As a user, I can <b>sort</b> the list of bugs based on different attributes (such as date created, priority) so that I can see the bugs I am currently interested in first.                                                     |
|                                                                                                                              | As a user, I can perform <b>filters</b> based on different attributes of bugs (such as date created, priority) so that I don't be distracted by bugs that I am not interested in currently.                                     |
|                                                                                                                              | As an indecisive user, I want the app to <b>decide</b> for me <b>which bugs to fix</b> for today                                                                                                                                |

|  |                                                                                                                                                                             |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | so that I don't have to waste much time choosing one.                                                                                                                       |
|  | 7. As a user, I can perform most operations on multiple bugs as if performing on one bug ( <b>mass operation</b> ) so that I can update the state of multiple bugs at once. |

### Duy

1. **(Keep)** As a user who wants to phrase bug description in a better way or correct typos, I can **update information about the bug** so that I have a better understanding about this bug.
2. **(Keep)** As a user, I can **update/remove tags** (deadline, category, etc.) when I find them not appropriate anymore so that my list of bugs is more well-categorized.
3. **(Remove)** As a user, I can **detect duplicate tags within the same bug** in case I unintentionally add them many times so that my list of tags is well-defined.
4. **(Remove)** As a user, I can **detect duplicate bugs** which have the same type and description with others in case I unintentionally create them many times so that my list of bugs is well-defined.
5. **(Keep)** As a user, I can **search my bugs based on any tag keyword** so that I can easily find them without manual searching.
6. **(Remove)** As a user, I can search **my bugs based on any description keyword** so that I can easily find them when necessary.

### Rishabh :(

1. As a user, I want to be able to tag my bugs according to their **level of priority** so that I know the order in which to clear them.
2. As a user, I want a command that allows me to be able to **see all the bugs** I'm currently working on as well as those that I need to work on.
3. As a power user, I want to be able to **export** my app data to a .txt file which can be tracked by version control.
4. As a power user, I want to be able to track bugs from **multiple projects** with clear distinctions between the two projects.
5. As a user, I want the bugs to be **saved** so they can be re-loaded when the command line application is re-launched.
6. As a long time user, I want to have a **searchable list of bugs that I have encountered** and solved before so I can access information quickly if I were to encounter the same bug again.

| Keep                                                                              | Toss                                                                              |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| 1. As a user, I want to be able to tag my bugs according to their <b>level of</b> | 1. As a power user, I want to be able to track bugs from <b>multiple projects</b> |

|                                                                                                                                                              |                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>priority</b> so that I know the order in which to clear them.</p>                                                                                      | <p>with clear distinctions between the two projects.</p>                                                                                                                                                |
| <p>2. As a user, I want a command that allows me to be able to <b>see all the bugs</b> I'm currently working on as well as those that I need to work on.</p> | <p>2. As a long time user, I want to have a <b>searchable list of bugs that I have encountered</b> and solved before so I can access information quickly if I were to encounter the same bug again.</p> |
| <p>3. As a power user, I want to be able to <b>export</b> my app data to a .txt file which can be tracked by version control.</p>                            |                                                                                                                                                                                                         |
| <p>4. As a user, I want the bugs to be <b>saved</b> so they can be re-loaded when the command line application is re-launched.</p>                           |                                                                                                                                                                                                         |

#### Kishen (coding god) :)

- As a disorganised software engineer without regular internet access, I want to be able to record bugs in my code **offline** so that I can better keep track of my bugs.
- As a user, I want to be able to keep track of my **multiple bugs** so that I can fix all of them and improve my product.
- As a user, I want to be able to manage the **progress** of my bugs so that I can visualise how many bugs I still have to fix.
- As a busy person, I want to be able to see which issues are **most important** so I can dedicate my time to solving those first.
- As someone who has random short pockets of free time, I want to be able to see which bugs will take the **least amount of time** to fix so that I can work on those first.
- As a user, I want to be able to add **notes** to my bugs so that I can remember where I last left off.
- As a user with many deadlines to meet, I would like to be able to **tag bugs to deadlines** so that I can fix the product before my deadlines.
- As a user who prefers CLI tools, I want to be able to access the bug tracking application via **CLI** so that I can work more effectively.

- As a user with multiple projects, I would like to have logs of my **different projects** so that I can simultaneously keep track of each project without cluttering up a single board.
- As a forgetful user, I would like to add **reminders** for myself so that I can remember important tasks I set for myself
- As a long term user, I would like to be able to **archive** my old projects so that I can keep a copy of how I fixed my bugs so that I can use it in the future.
- As a user, I would like to keep track of the **date** the bug was added so that I can monitor how long it takes to solve it.
- As a user, I would like to add an estimate of how many **%** of a certain task I **have finished** so that I can better estimate how much more time I would need to solve an issue.
- As a long term user, I would like to be able to generate **bug reports** for completed bugs so that I can review the process I used to fix the issue.
- As a user, I would like to be able to tag issues with a number so that I can keep track of the **number of issues** that I have.
- As a user, I would like to be able to generate a **summary of all bugs** in my program so that I can get a big picture view of all the work that is to be done.
- As a user, I would like to be able to **link** my bugs to each other so that I understand interconnected problems in my product.
- As a user, I would like to be able to generate **summaries of inter-related bugs and issues** so that I can better assess the impact that some bugs have on my product.
- As a user, I would like to add tags to my bugs that detail which **component** of my product it **affects** so that I can easily track all problems related to specific components.

#### Features to be implemented:

- Record bugs **offline**.
- Add bugs
- Delete bugs
- Add **notes** to specific bugs.
- Access the bug tracking application via **CLI**.
- Keep track of the **number of issues**.
- Add tags about which **component** of my product the bug **affects**.
- Note down the **solution** for each bug
- Command to **see all the bugs**
- Export data to a txt file
- Save data after app is closed
- Remove tags



- Search bugs based on any tag keyword

### Feature Sets v1.2:

|                            |                                                                                                                                                                                                                                                 |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bug                        | <ul style="list-style-type: none"> <li>• Add bugs</li> <li>• Delete bugs</li> <li>• Add <b>notes (description, solution, etc.)</b> to specific bugs.</li> <li>• <i>Add tags about which component of my product the bug affects.</i></li> </ul> |
| Bug List                   | <ul style="list-style-type: none"> <li>• </li> <li>• Command to <b>see all the bugs</b></li> <li>• Export data to a txt file</li> <li>• Save data after app is closed</li> </ul>                                                                |
| Tag (Nice to have in v1.2) | <ul style="list-style-type: none"> <li>• Add tags</li> <li>• Remove tags</li> <li>• Search bugs based on any tag keyword</li> <li>• <i>Add tags about which component of my product the bug affects.</i></li> </ul>                             |

Allocation of Commands for UG :

Help: help (Phong)

<Add to bottom>

Exit: bye

Save:

List : list

List by state : list s/state (Duy)

Add bug: add n/bugName d/description s/state

Delete bug: delete <index>( Roger)

Edit bug: edit <index> n/newName d/newDescription (Kishen)

Move bug: move <index> s/state (Rishabh)

Shortlisted user stories:

- As a disorganised software engineer without regular internet access, I want to be able to record bugs in my code **offline** so that I can better keep track of my bugs.
- As a user, I want to be able to add **notes** to my bugs so that I can remember where I last left off.
- As a user who prefers CLI tools, I want to be able to access the bug tracking application via **CLI** so that I can work more effectively.
- As a user, I would like to be able to tag issues with a number so that I can keep track of the **number of issues** that I have.
- As a user, I would like to add tags to my bugs that detail which **component** of my product it **affects** so that I can easily track all problems related to specific components.
- As a forgetful user, I want to note down the **solution** for each bug I solved so that I won't make the same mistake again.
- As a forgetful user, I want to note down the **solution** for each bug I solved so that I won't make the same mistake again
- ~~● As a user, I want to be able to tag my bugs according to their **level of priority** so that I know the order in which to clear them.~~
- As a user, I want a command that allows me to be able to **see all the bugs** I'm currently working on as well as those that I need to work on.
- As a power user, I want to be able to **export** my app data to a .txt file which can be tracked by version control.
- As a user, I want the bugs to be **saved** so they can be re-loaded when the command line application is re-launched.
- ~~● As a user who wants to phrase bug description in a better way or correct typos, I can **update information about the bug** so that I have a better understanding about this bug.~~
- As a user, I can **update/remove tags** (deadline, category, etc.) when I find them not appropriate anymore so that my list of bugs is more well-categorized.
- As a user, I can **search my bugs based on any tag keyword** so that I can easily find them without manual searching.

## **User Stories for v1.2 Issues**

### **PRIORITY\_HIGH**

#### Help

- As a new/forgetful user, I want to see usage instructions so that I can refer to the instructions when I don't know/forget the commands

#### List

- As a user, I want to see all the bugs or bugs in a particular state in the tracker so that I can better keep track my bugs

#### Add

- As a user without regular internet access, I want to add new bug record in the tracker app offline

#### Delete

- As a user, I want to delete an existing bug record so that I can remove entries that I no longer need

#### Edit

- As a user, I want to edit an existing bug description so that I can update new relevant information to this bug

#### Move

- As a user, I want to move an existing bug in the tracker from one state to another so that I can update new state to this bug

#### Save

- As a user, I want the bugs to be saved so they can be re-loaded when the command line application is re-launched.

#### Exit

- As a user, I want to exit the application when it is not in use

### **PRIORITY\_MEDIUM**

#### Add tag

- As a user, I want to add new tags to an existing bug in the tracker so that my list of bugs is well-categorized

#### Remove tag

- As a user, I can remove tags to an existing bug in the tracker when I find them not appropriate anymore so that my list of bugs is more well-categorized.

### **PRIORITY\_LOW**

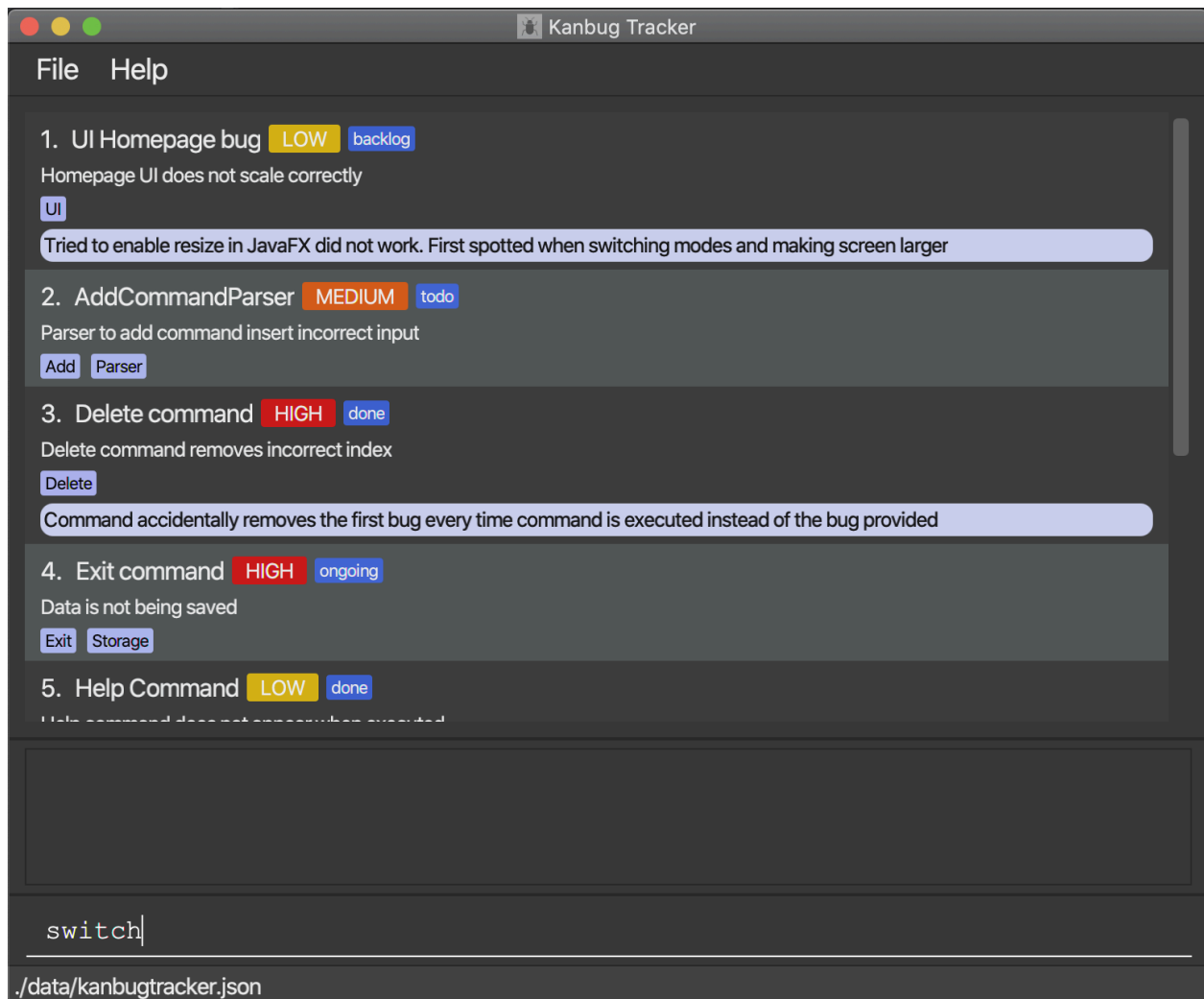
#### Search bugs based on tag keyword

- As a user, I can search my bugs based on any tag keyword so that I can easily find them without manual searching

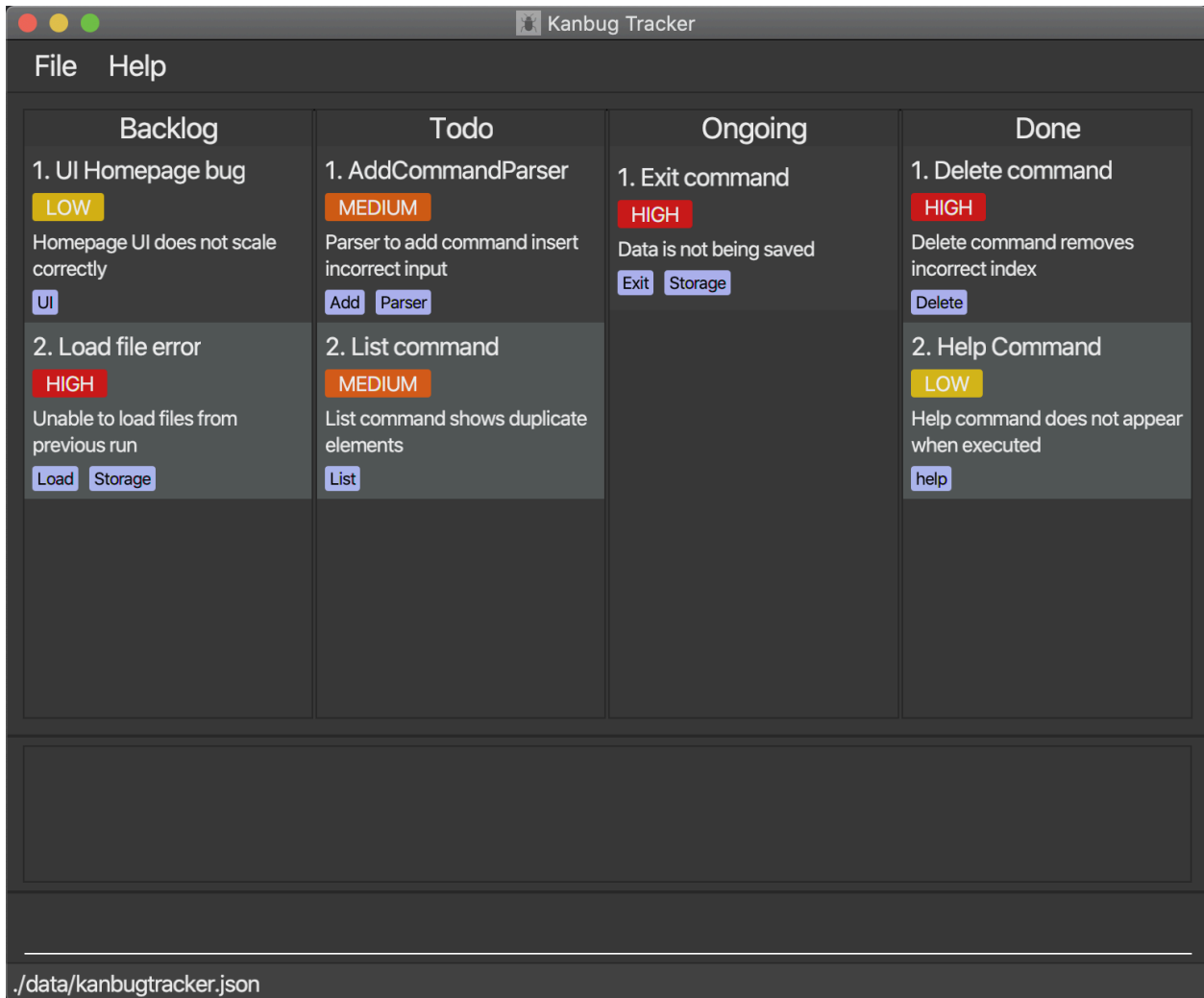
# v1.3 features demo

## 1. Switch command

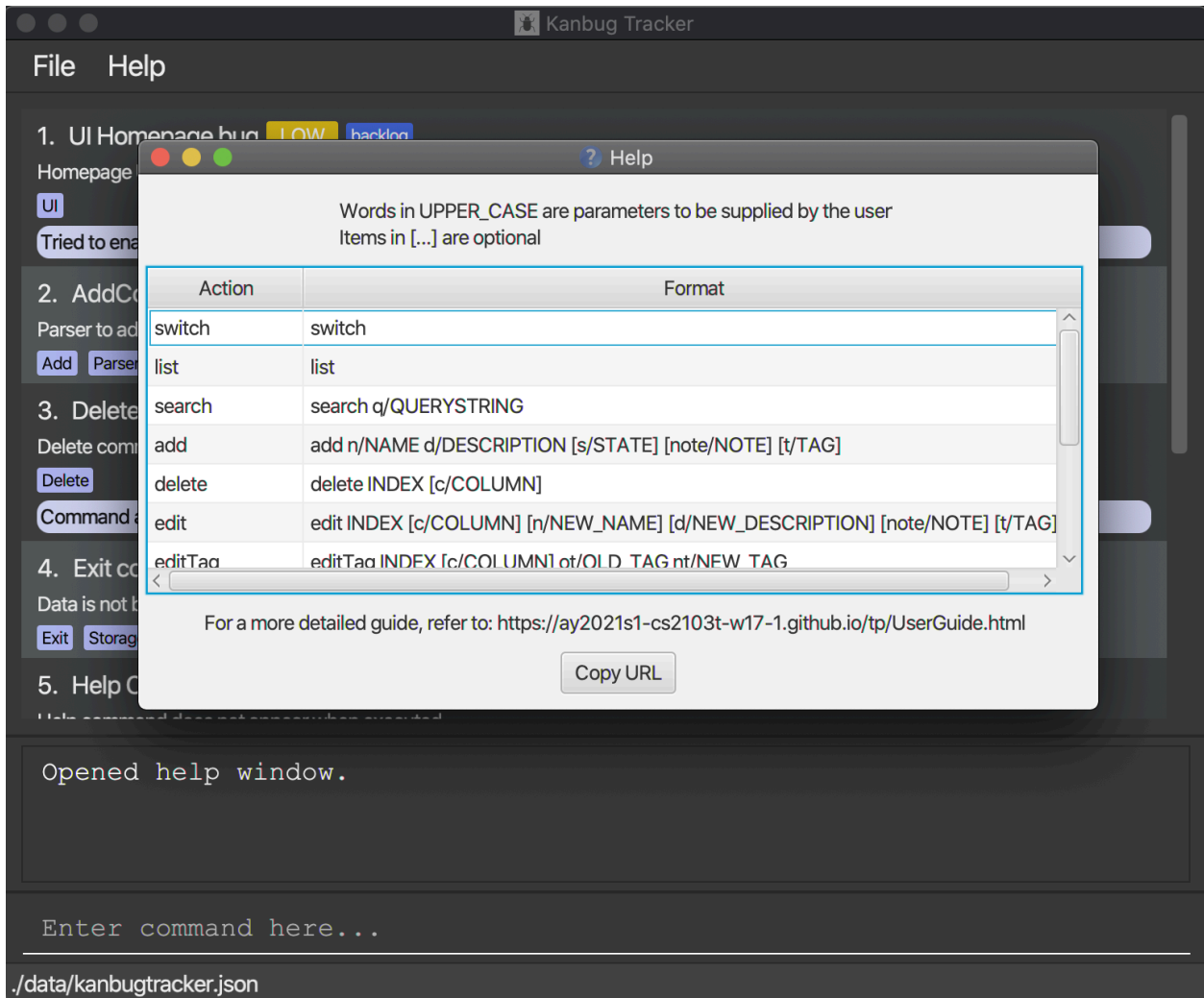
### a. Typing in **switch** command



### b. View changed to KanBan view



## 2. Help command



File Help

1. UI Homepage bug LOW backlog

Homepage

UI

Tried to ena

2. AddCo

Parser to ad

Add Parser

3. Delete

Delete com

Delete

Command a

4. Exit co

Data is not b

Exit Storage

5. Help C

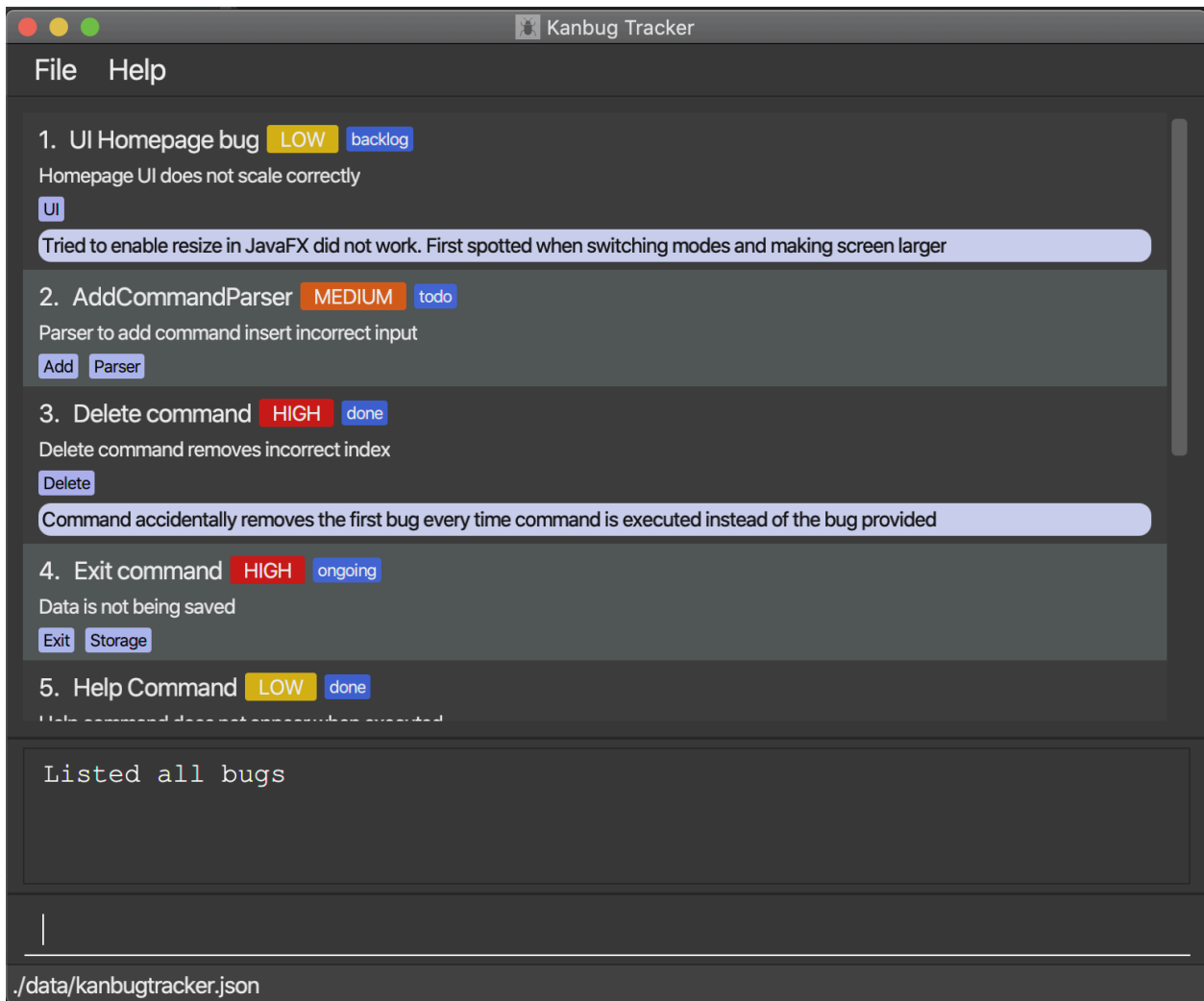
Help command description provided

Opened help window.

Enter command here...

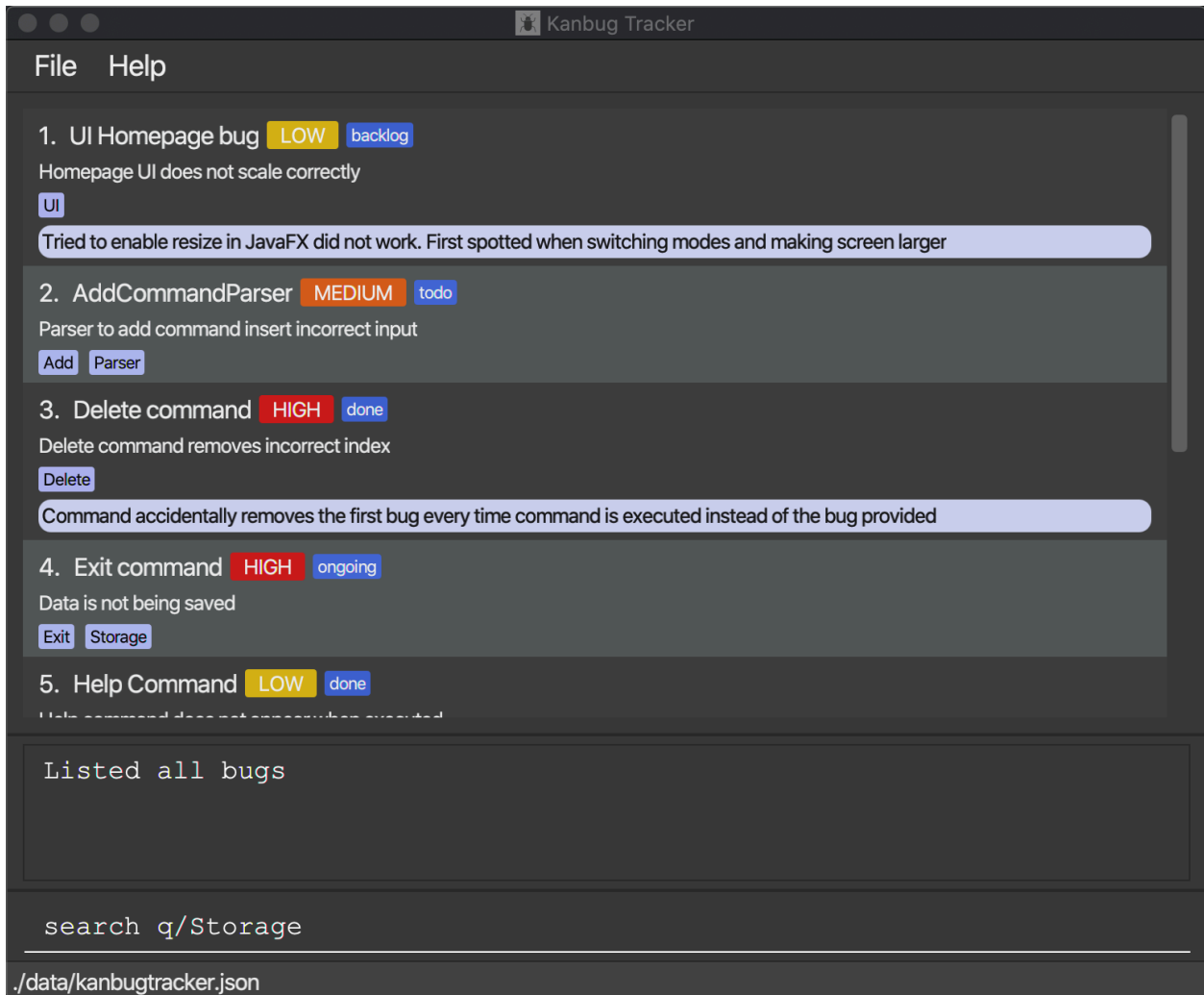
./data/kanbugtracker.json

### 3. List all bugs



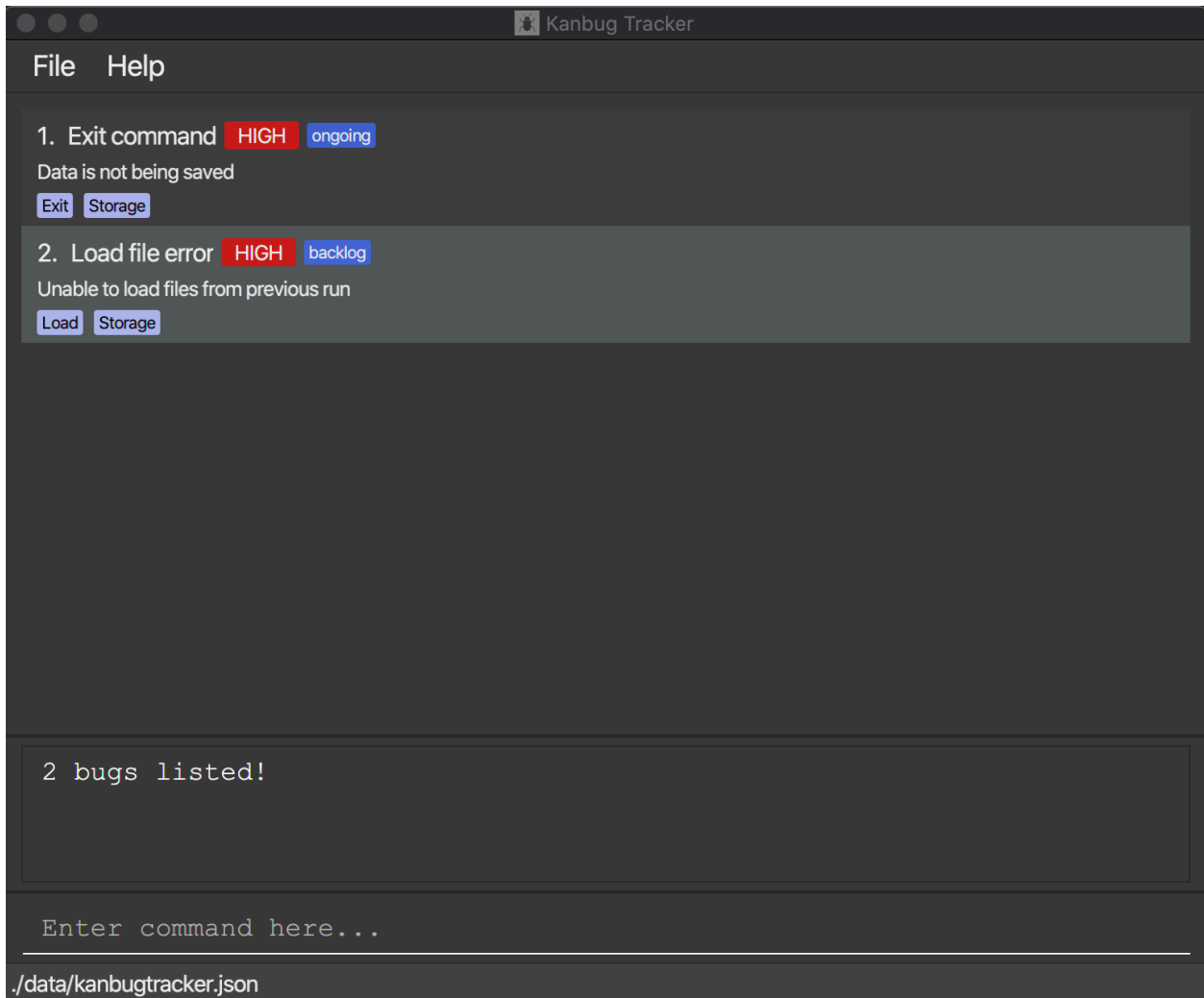
### 4. Search for a bug

- a. Search for a bug with the query string "Storage"



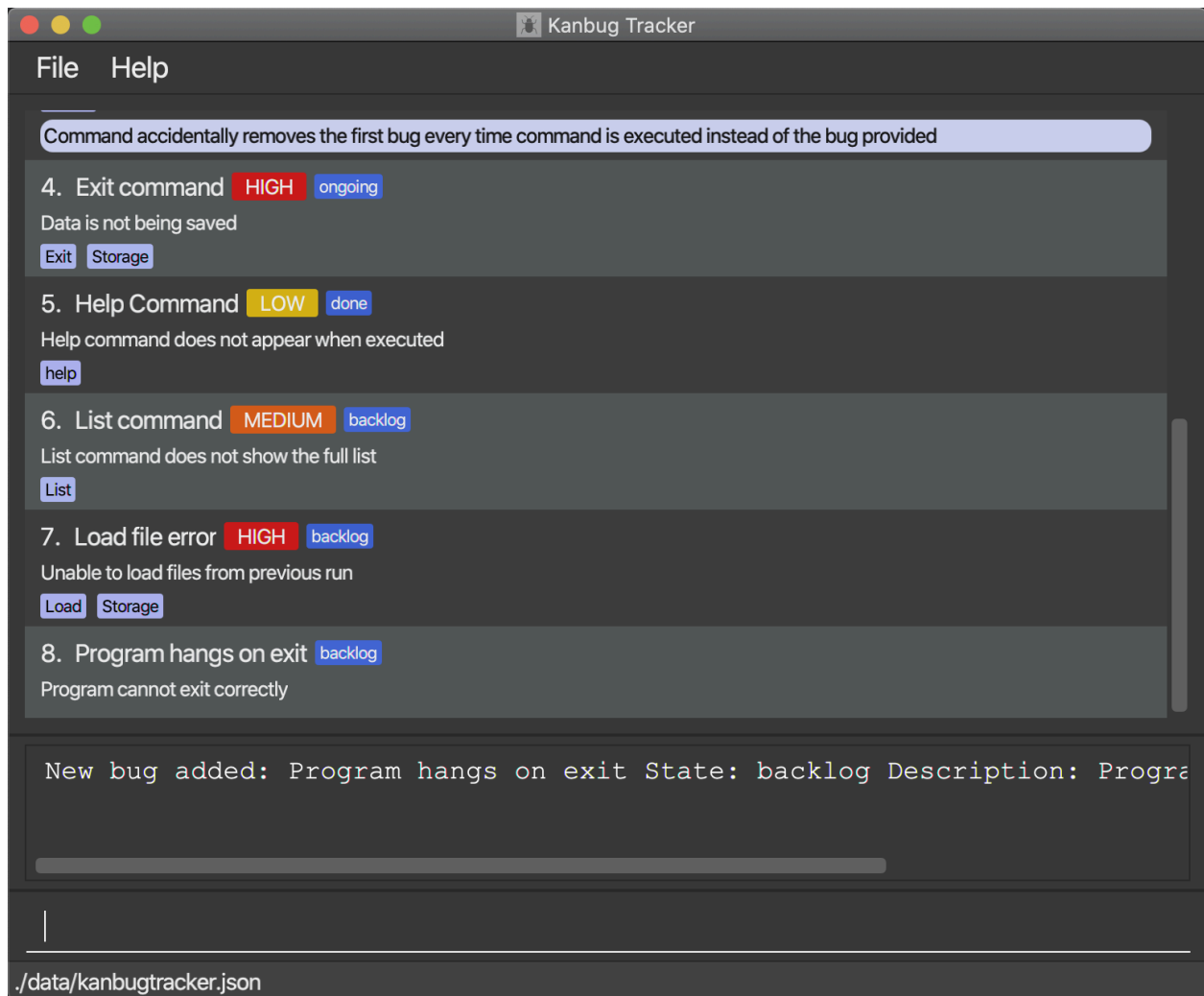
b. Search result shown





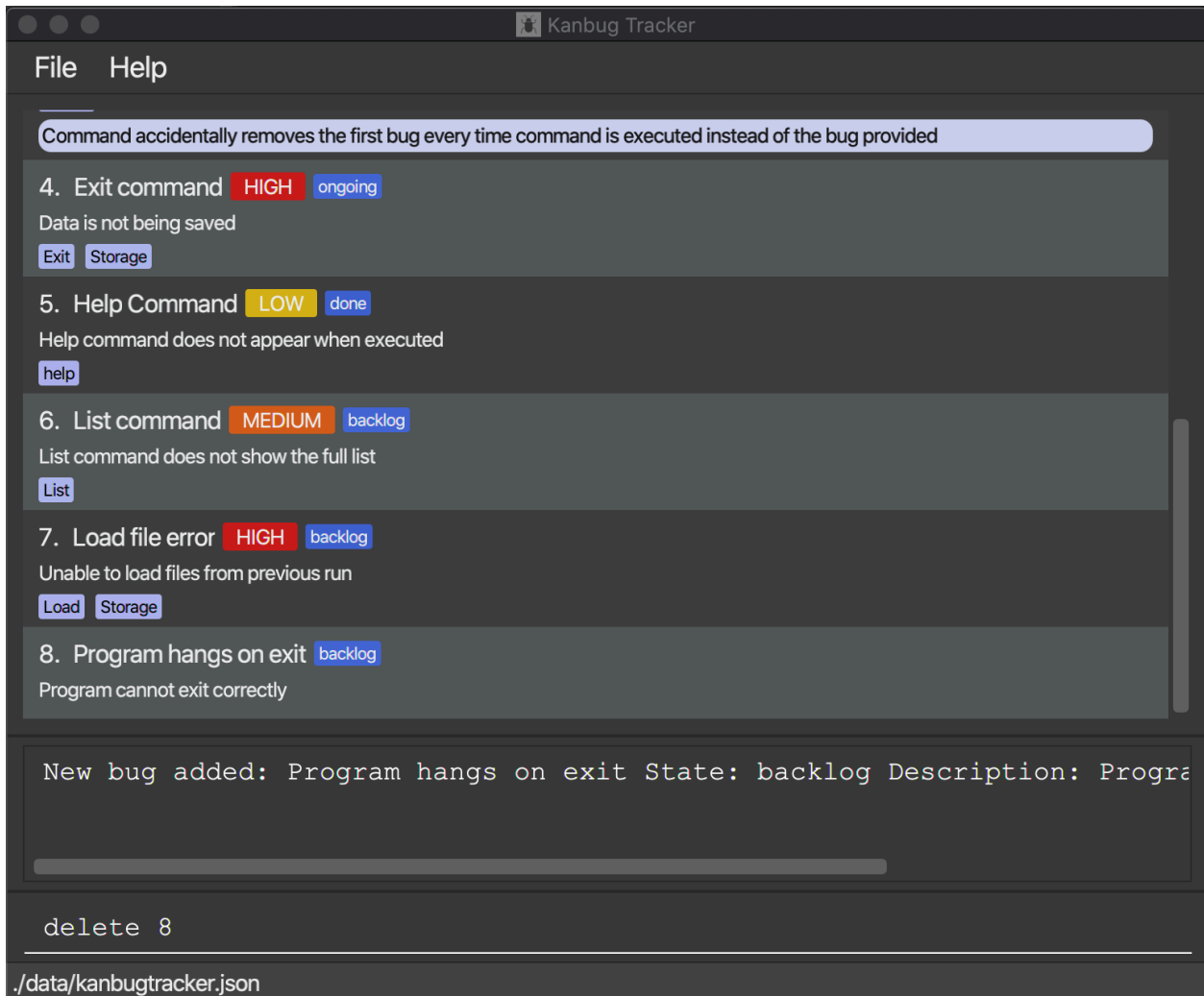
## 5. Adding a bug

- a. Adding a bug with name "Program hangs on exit", description "Program cannot exit correctly" and state "backlog"
- b. Bug added to KanBug Tracker

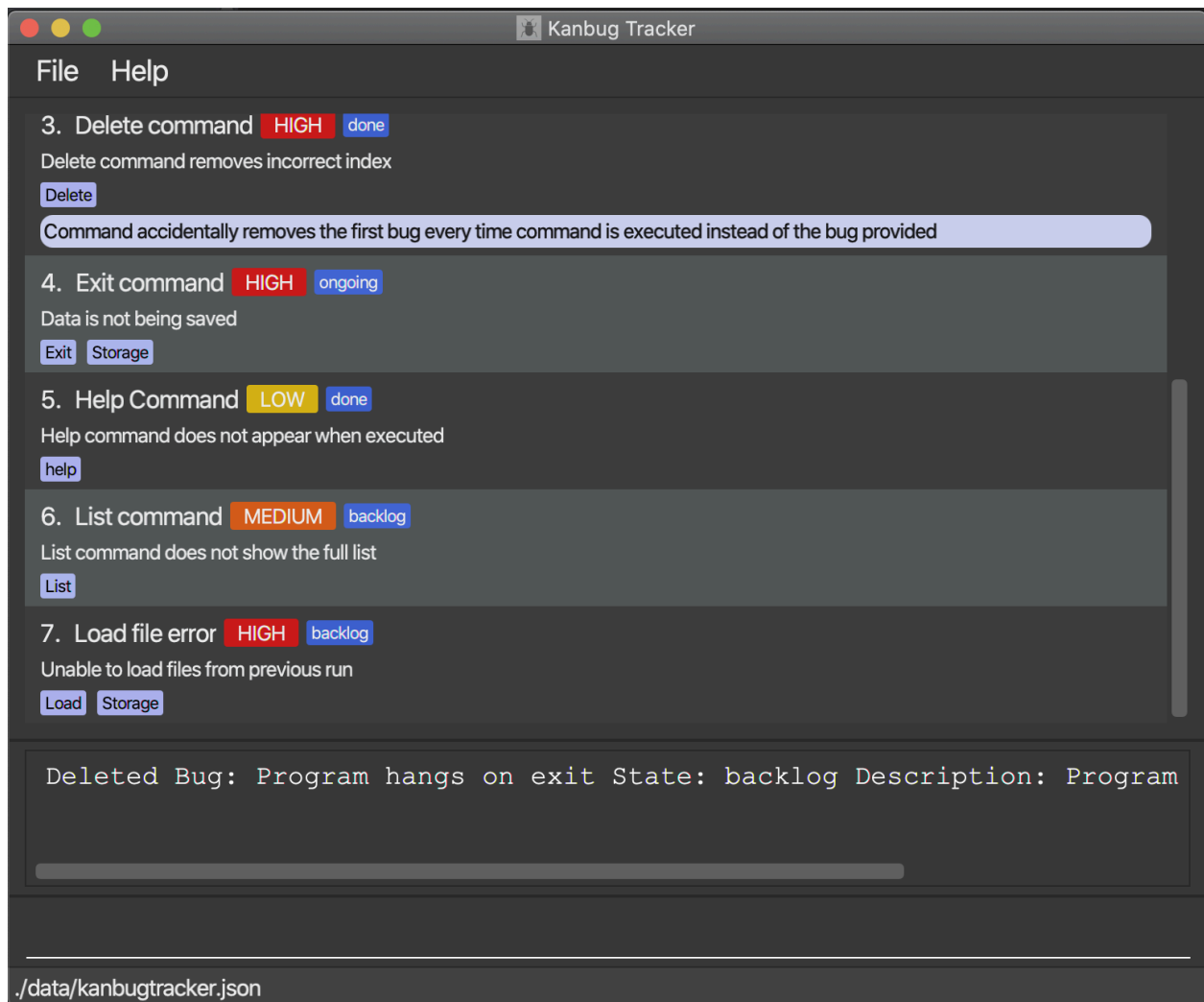


## 6. Deleting a bug

- a. Deleting bug at index 8

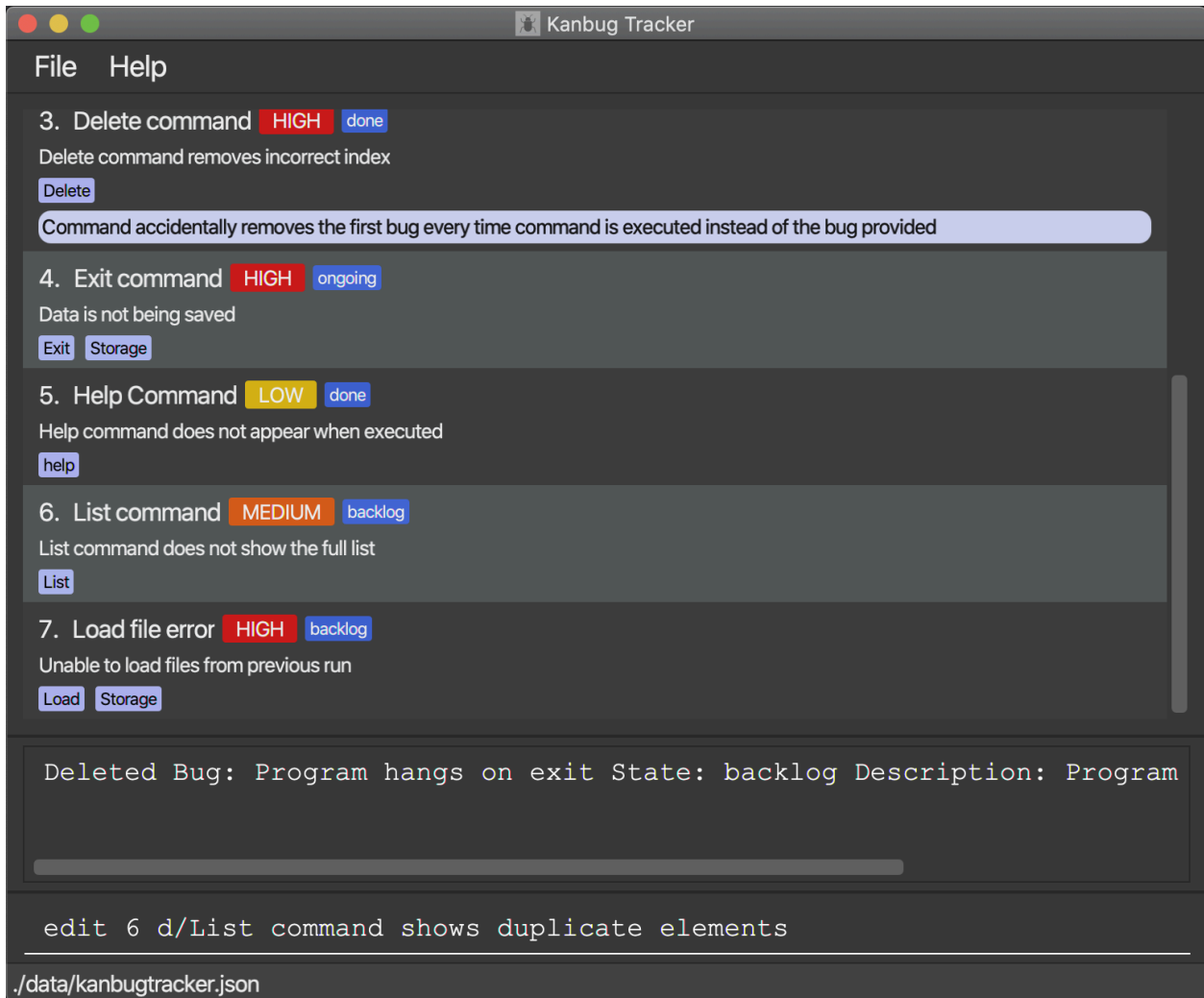


b. Bug deleted

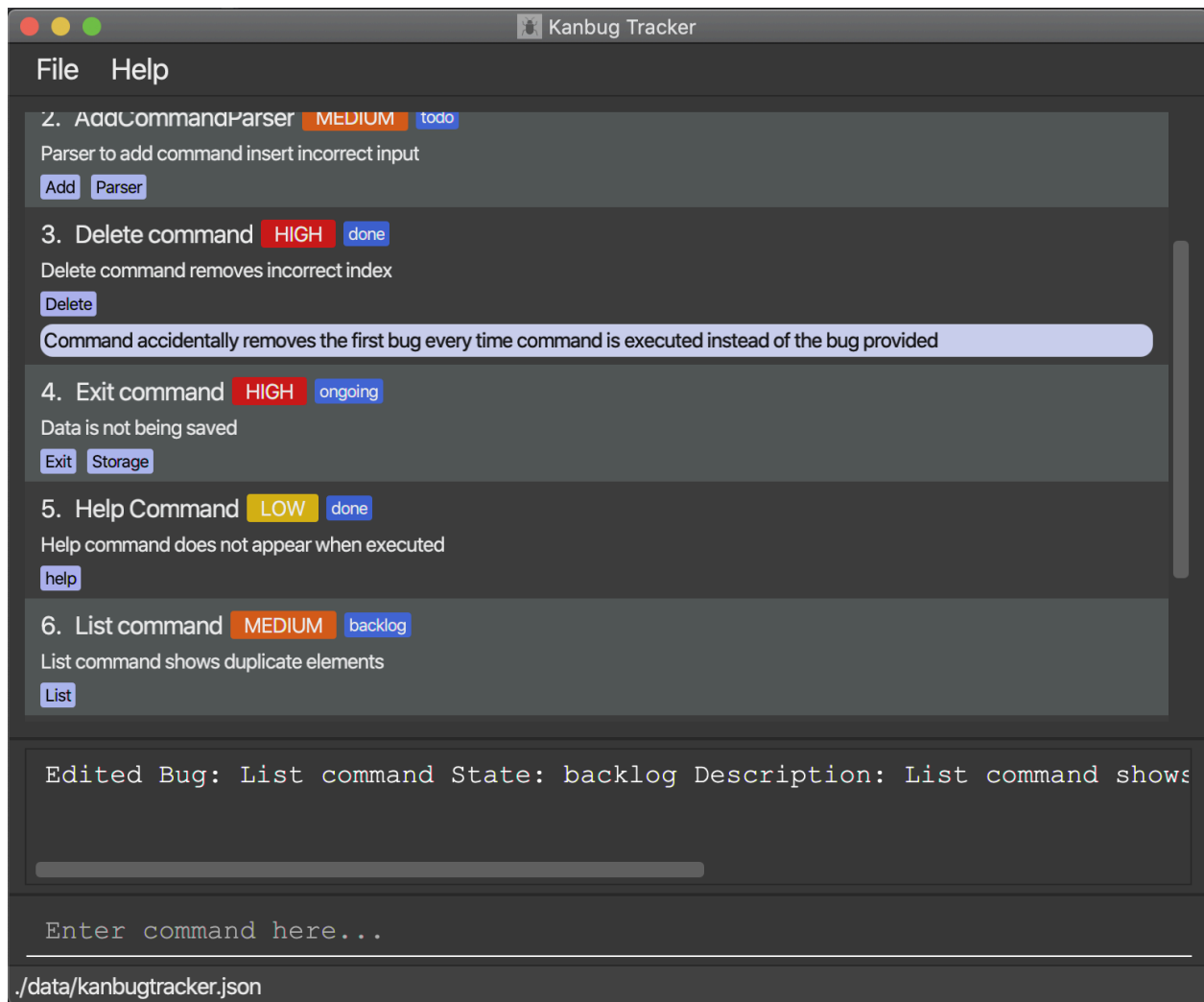


## 7. Editing a bug

- Editing bug at index 6, changing its description from "List command does not show the full list" to "List command shows duplicate elements"

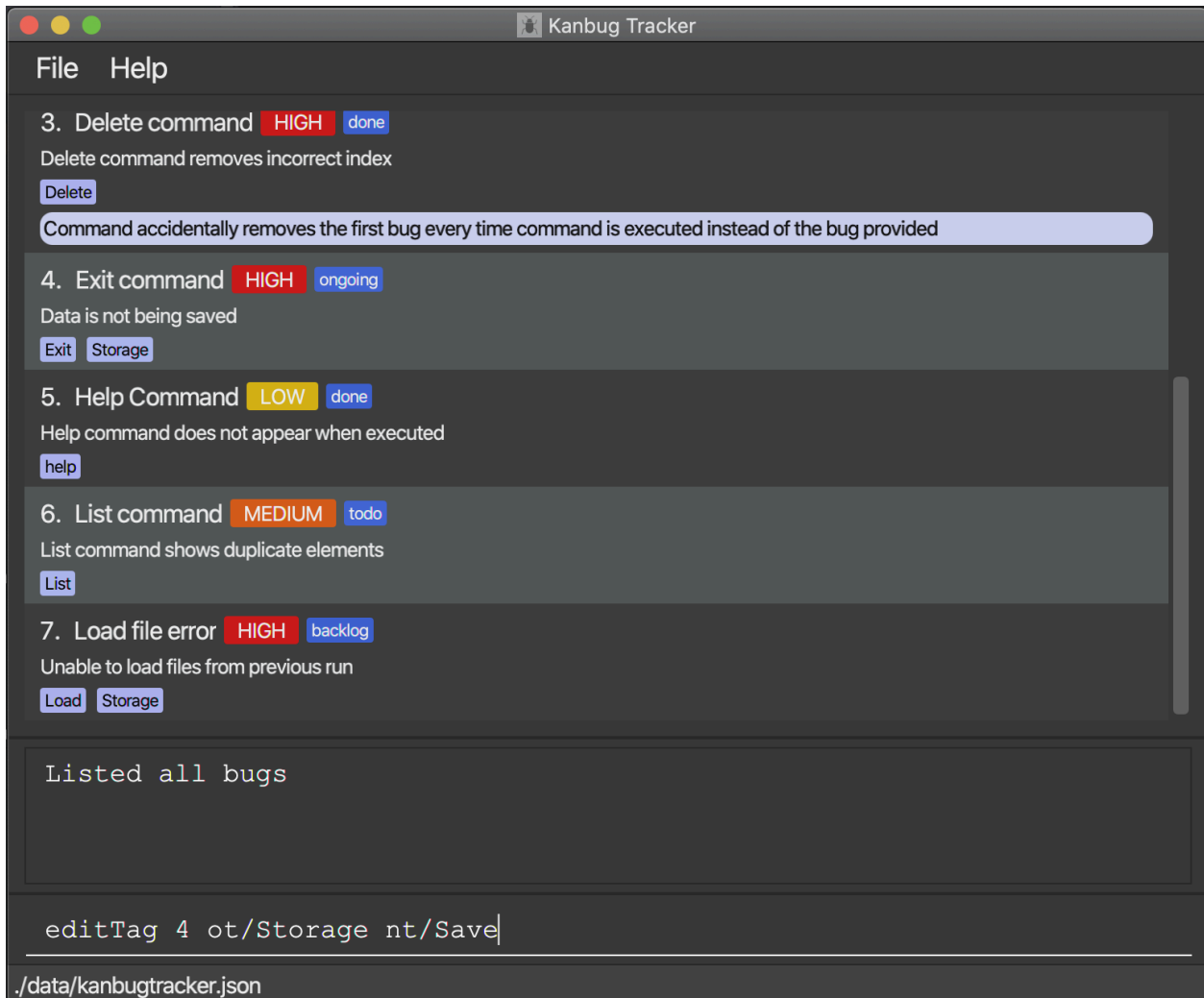


b. Bug edited

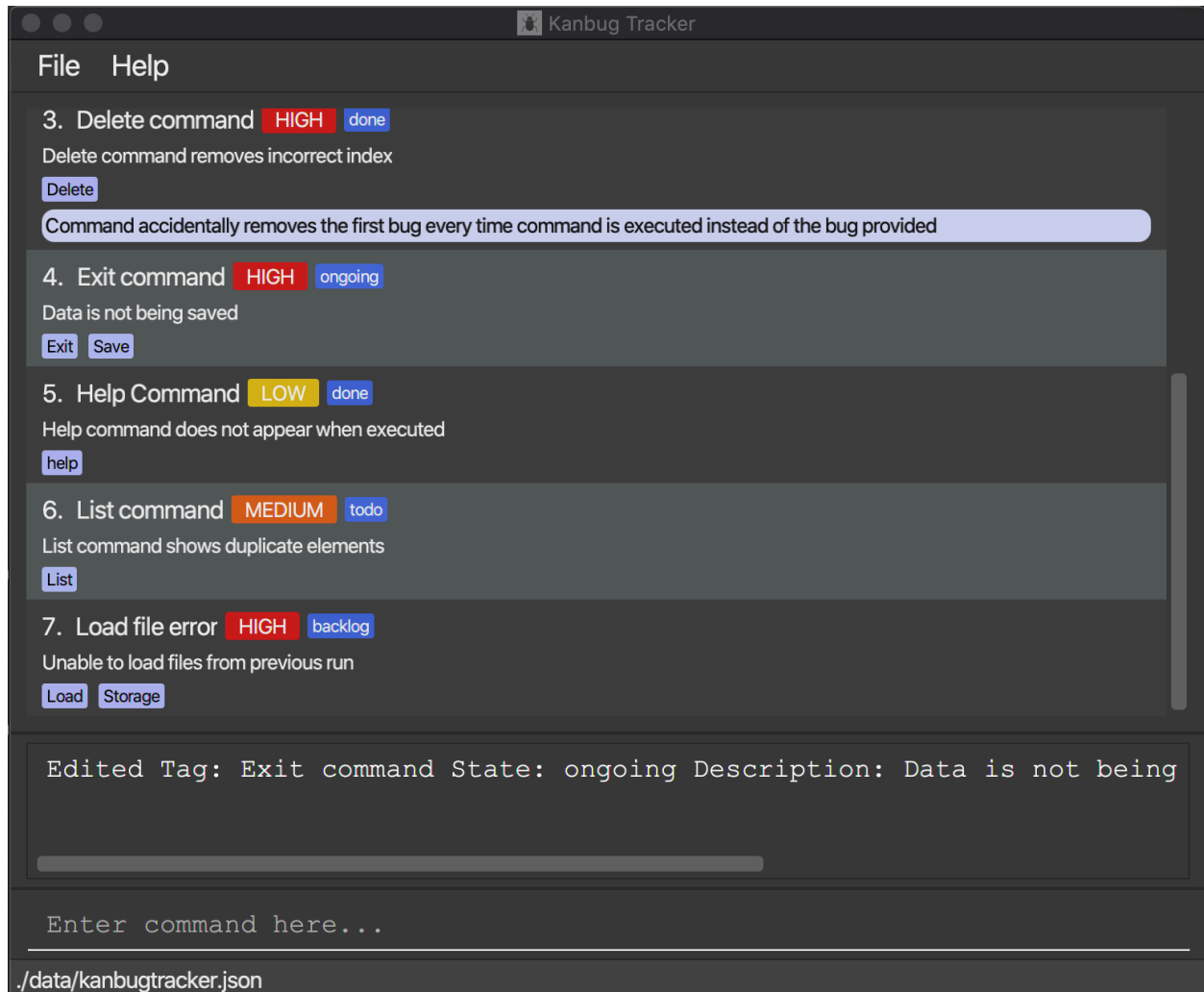


## 8. Edit a tag of a bug

- Editing tag of bug at index 4 from "Storage" to "Save"



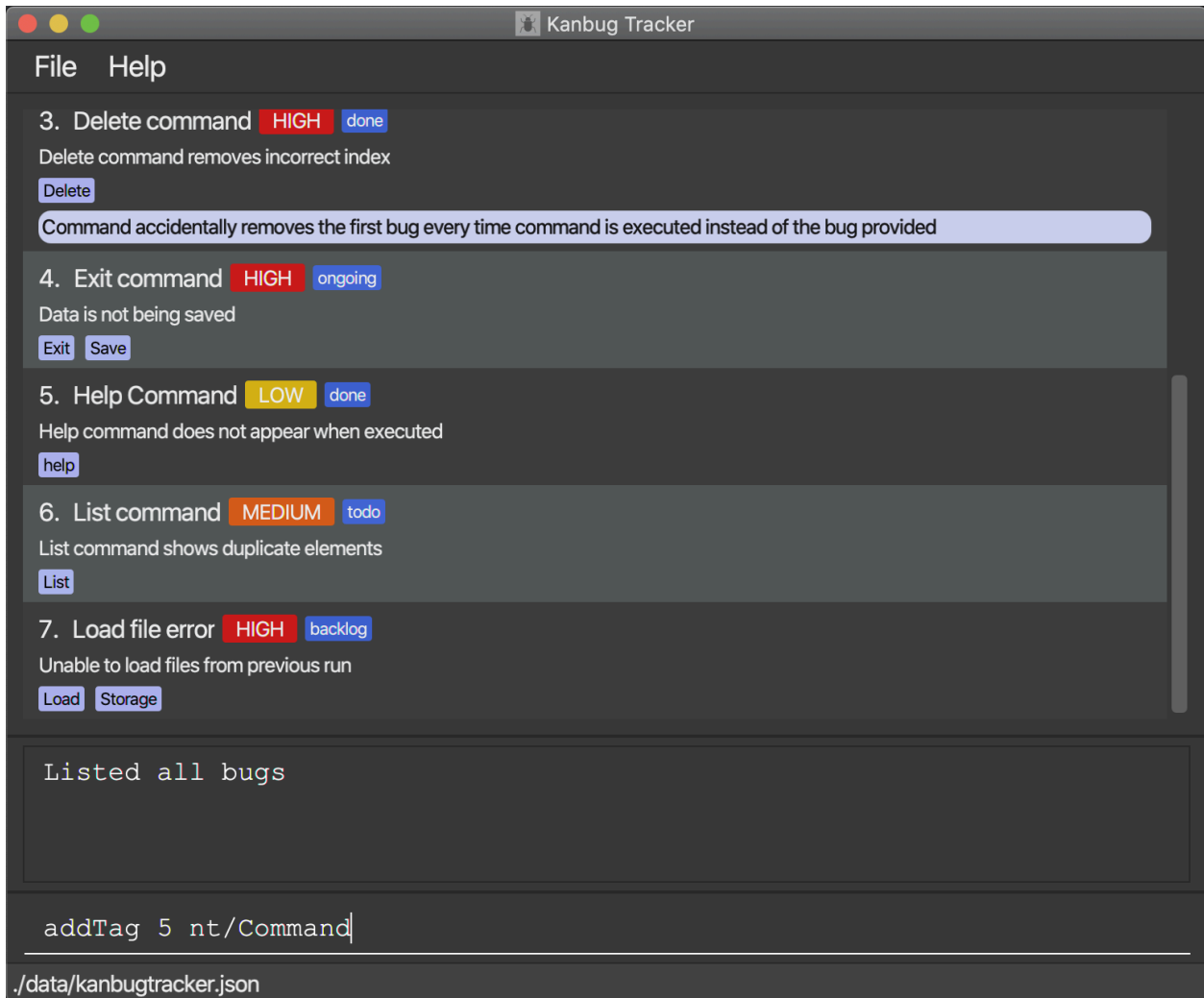
b. Tag of bug changed to "Save"



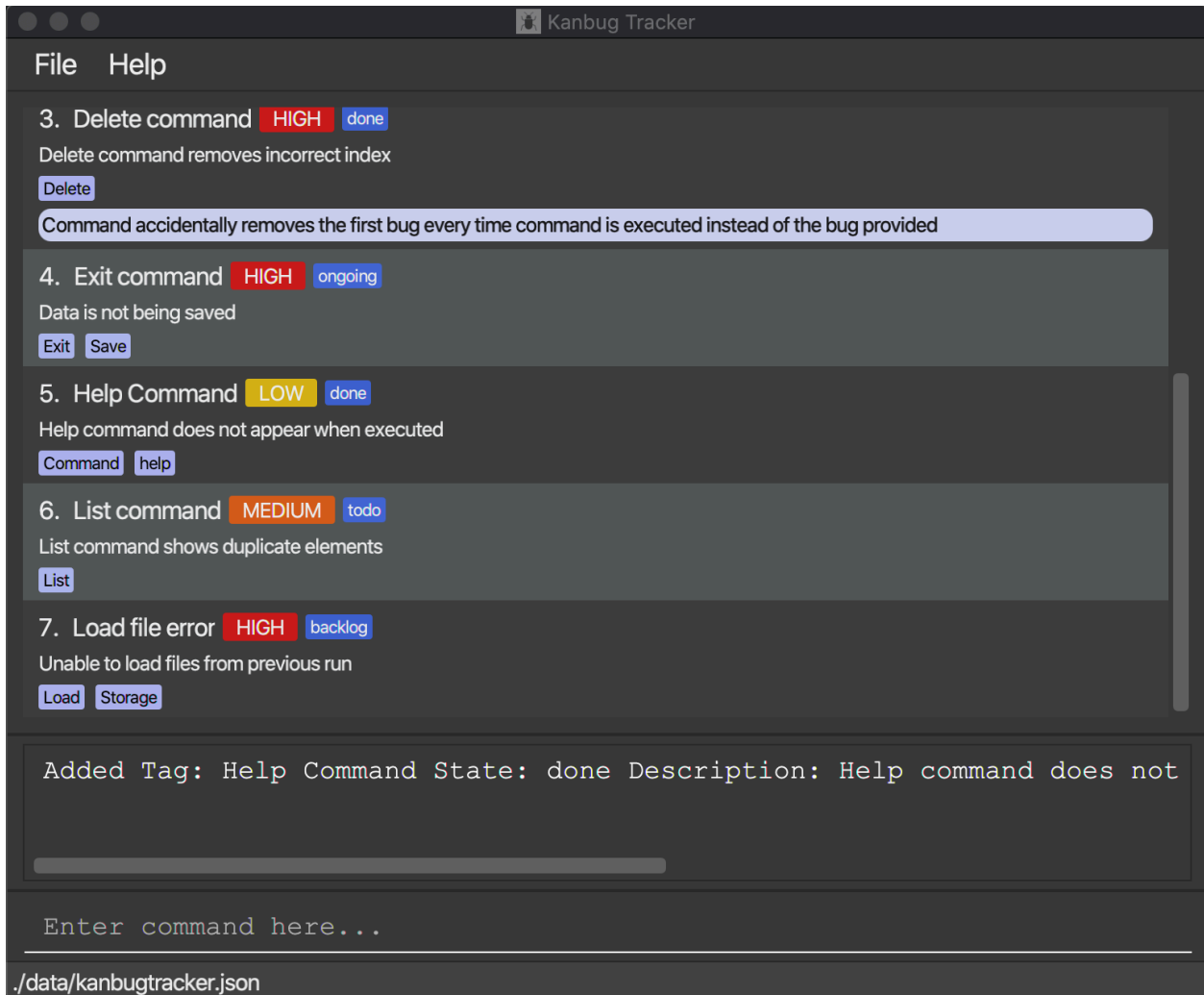
## 9. Adding a Tag to an existing bug

- Adding a "Command" tag to bug at index 5



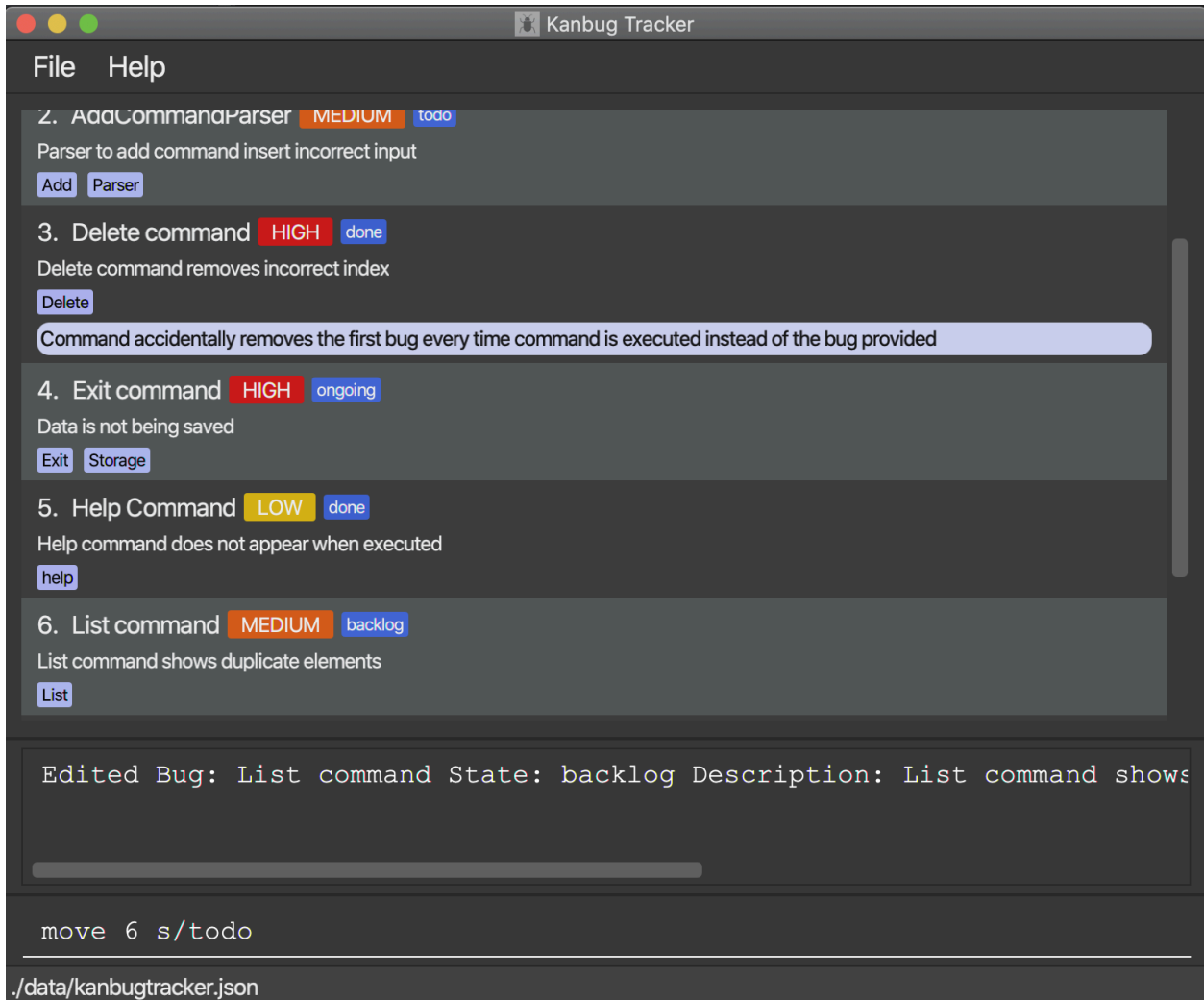


b. "Command" tag added

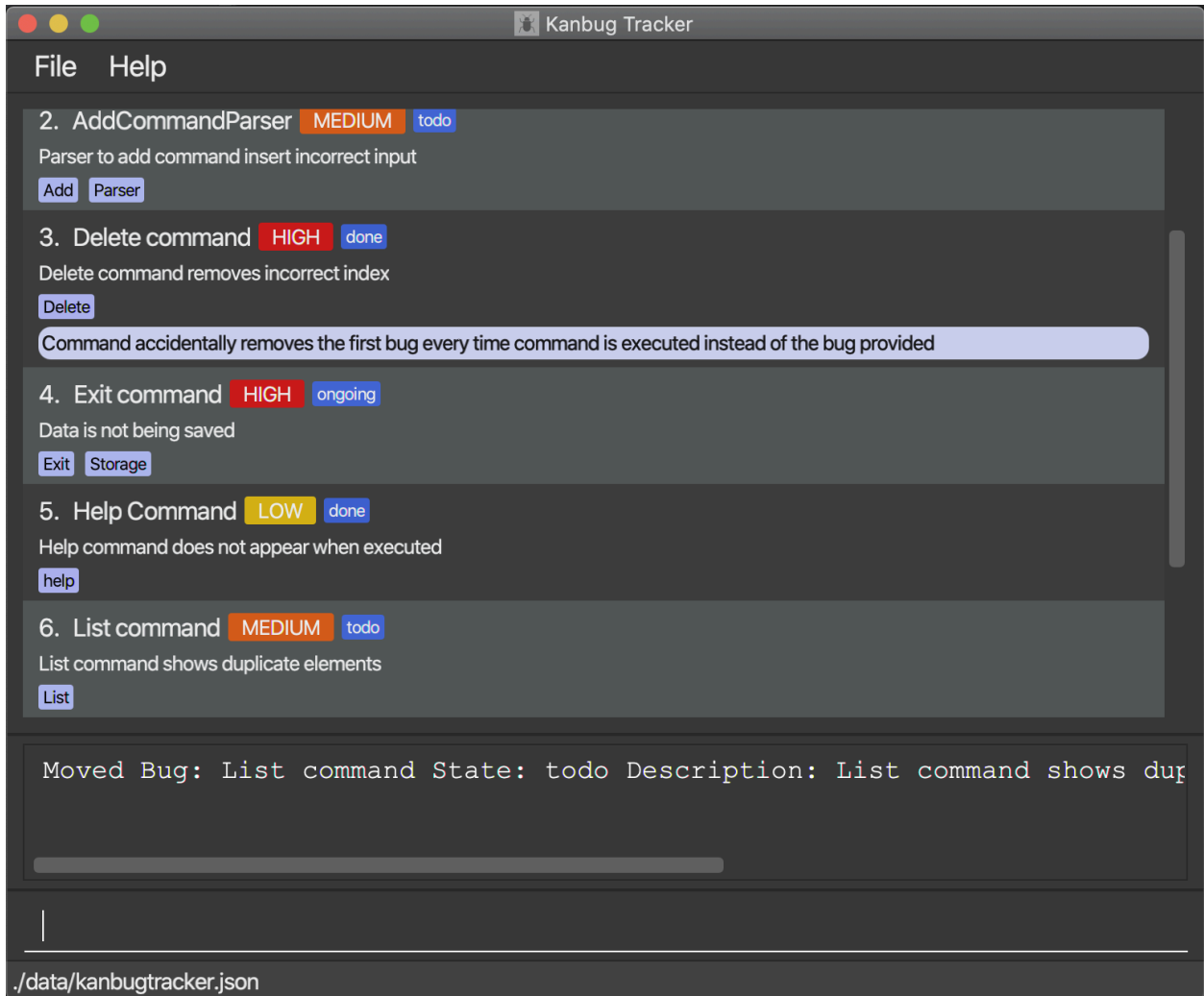


## 10. Moving a bug

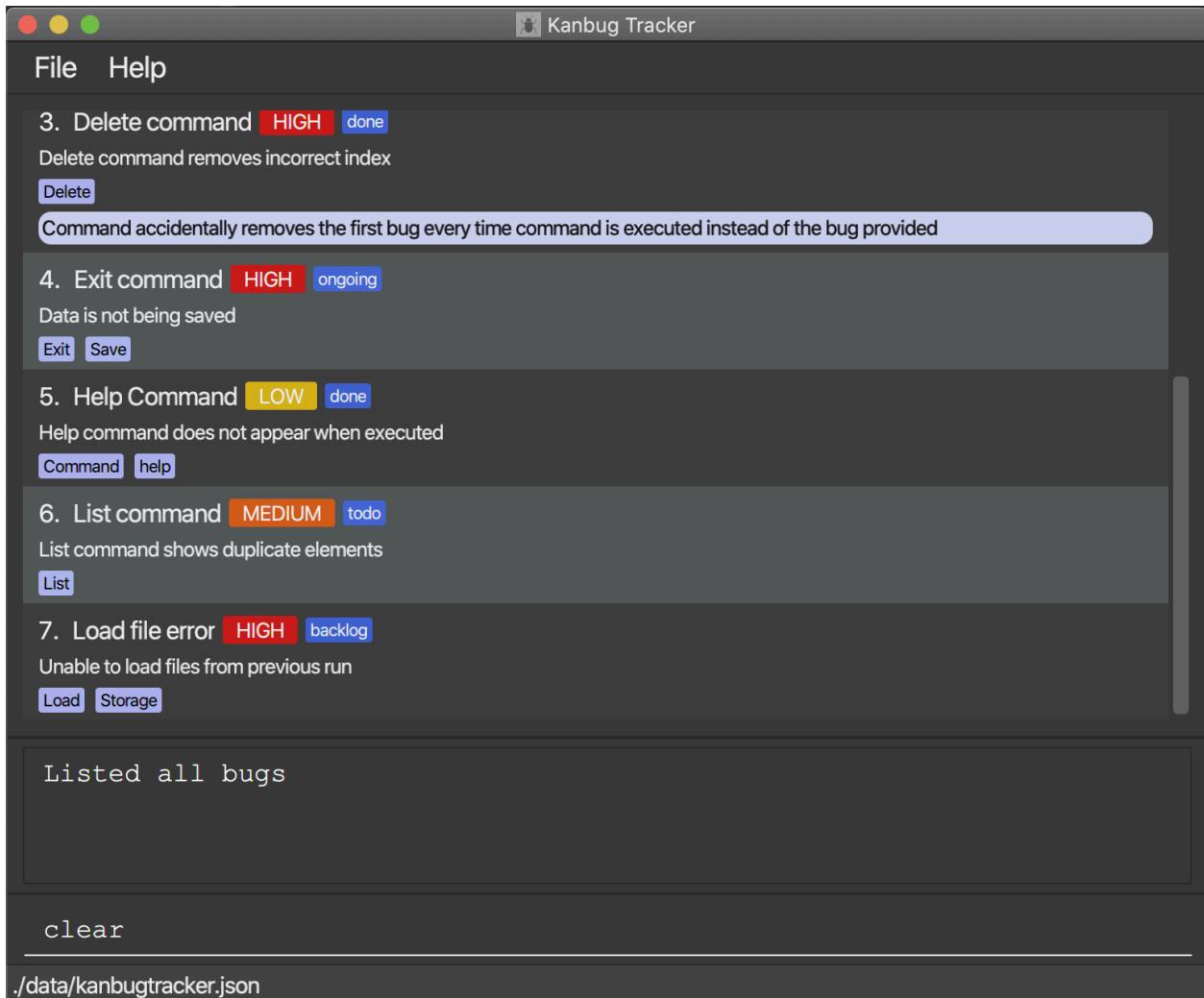
- Moving bug at index 6 from "backlog" state to "todo" state



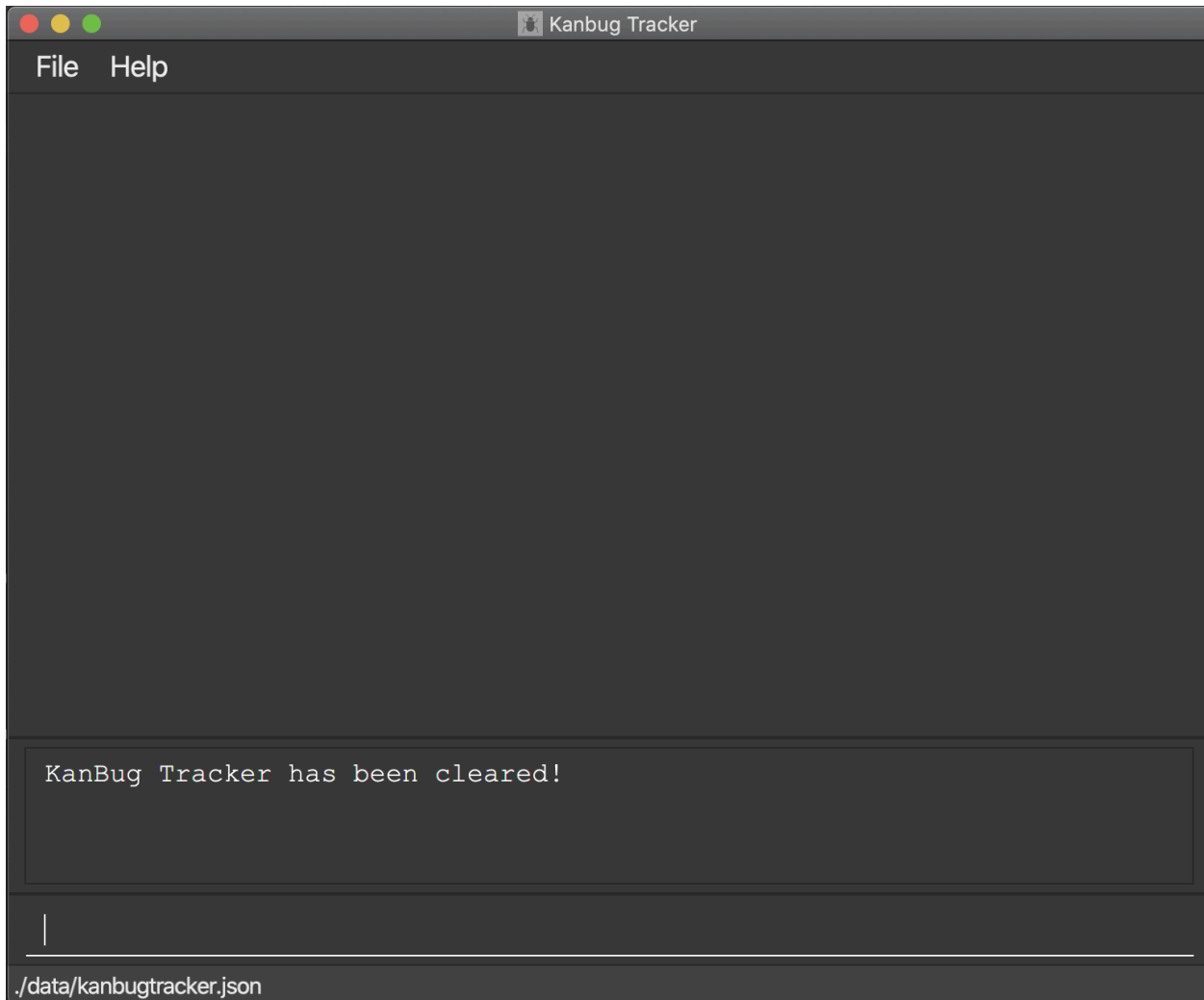
b. Bug moved



11. Clearing all bugs from the tracker
  - a. Type in `clear` command



b. All bugs are removed from tracker



## Do a postmortem of the previous iteration

1. We did too much of the work sequentially and so we worked at a pace that was much slower than what we could have. We should make better use of branching so that we can work on our parts in parallel.
2. The size of each person's allocated task was too small. The benefit of this is that everyone managed to finish their parts on time. However because each part was so small, it is somewhat linked to the first point because the tasks became too dependent on one another which caused us to follow a sequential workflow. We will be assigning each person larger and more independent tasks in the future iterations.
3. We need to improve the organisation of the branch structure, at one point we were all very confused about which branch we should be pushing our code to.
4. Haphazard testing with no clear people to work on them resulting in insufficient code coverage. We need to set aside clear times and roles to work on the testing.
5. No proper integration phase, we need to schedule in specific time to integrate

