# CS2103 Project Meeting Notes

## Quick Reference (with links to relevant section)

1st Meeting: Week 3 (28/08/2020)

- Decided on evolving the current Addressbook Application
- Discussion on target audience & problem addressed
- Delegation of roles to respective team members

2nd Meeting: Week 4 (31/08/2020)

- Naming ideas:
- Target user profile:
- Value proposition

3rd Meeting: Week 5 (07/09/2020)

- Target Persona and Problem Scope
- Selecting relevant user stories for v1.2:
- Allocation of iP extension features:

4th Meeting: Week 6 (14/09/2020)

- Discussion of feature list CLI-nic v1.2
- Discussion of User Guide CLI-nic v1.2
- Allocation of feature each member will be working on

5th Meeting: Week 7 (21/09/2020)

- Do practice iterations for v1.1
- Update AboutUs page

6th Meeting: Week 7 (25/09/2020)

- Update team webpage
- Update User Guide
- Update Developer Guide
    - Product Scope
    - User Stories
    - Use Cases
    - Non-functional requirements
    - Glossary

7th Meeting: Week 8 (03/10/20)

- Draft out a simple UML diagram for Product, Warehouse and Supplier classes
- Updating user stories on GitHub
- Allocation of tasks to members

8th Meeting Week 8 (07/10/20)

- Add sample products for test cases
- Warehouse and Supplier classes has been created and merged with master

9th Meeting Week 9 (12/10/20)

Screenshots for v1.2 features demo
Iteration v1.3

10th Meeting Week 10 (19/10/20)
Post mortem of the previous iteration
Updates from members regarding features for iteration v1.3
Finalise user input structure for commands

11th Meeting Week 11 (25/10/20)
Merging in of codes for iteration v1.3
Testing of features for v1.3

12th Meeting Week 11 (26/10/20)
v1.3 Features Demo
Merge in all UG/DG for iteration v1.3
Smoke test among members and release jar file for v1.3 & closing of milestone v1.3
Other miscellaneous issues

**1st Meeting: Week 3 (28/08/2020)**

**Agenda:**

- Decide the direction whether to evolve or to morph the current addressbook
- Discuss target audience & problem addressed
- Delegation of roles to respective team members

**Discussion Notes:**

1. Decided on evolving the current Addressbook Application
   Cons: It would be harder to differentiate our projects with other teams apart from decrease in flexibility of our product's design.
   - **Potential ideas for differentiation our final product**
     - Using online APIs to retrieve information and store it in Addressbook (e.g. GitHub, LinkedIn, weblinks)
     - Implement other additional offline features (To reduce reliance on online APIs)

2. Discussion on target audience & problem addressed
   Given-constraints: **User is Tech Savvy, able to type fast and would prefer typing over other means**

   **Ideas discussed during the meeting:**

   1. **Consultation system for professors to use and keep track of scheduled consultation meetings with students**
      Problem that Addressbook will solve:
      Professors often have to manage many students due to large class size in addition to commitments from research/networking. This could be made worse if they are teaching different modules that have large class size. Therefore, it would be hard for them to keep track of which meetings they have scheduled for consultation with students (undergraduate or graduate research students). They would also require fast results to search for students' contact details using the address book in order to increase efficiency. Addressbook seeks to solve these issues by providing an easy and convenient way to keep track of these meetings in a central place.

      Features of Addressbook:
      Once the professor schedules a meeting with another user (e.g. student in the address book) and both parties confirm they are available for the meeting, a meeting will be scheduled for both parties.

Potential problem:

The physical addresses in the given Addressbook application will be deemed as redundant since users (professors) need not use them.

2. **System for businesses to manage their stocks in their warehouse (Chosen)**

   Problem that Addressbook will solve:

   A company selling vitamin products/masks may be sourcing their products from different suppliers. During a pandemic (e.g. COVID-19) they may find it hard to contact different suppliers at the same time (in order to increase their supplies rapidly and keep up with the demand) and check whether suppliers currently have stocks for certain items that they wish to import back. In addition, it will also be hard for them to know which items are selling out fast, which would require them to bring in more supplies in order to cope with the predicted increase in demand. Currently, the sales results may be managed using a different platform by the sales department. Addressbook aims to solve this issue by integrating the contact details of the suppliers so that managers in charge of stocks in the warehouse can quickly locate the relevant suppliers. In addition, suppliers will be able to update the stocks they currently have so that anyone who wishes to buy the items can place an order without calling/emailing just to confirm the availability of the products. This added efficiency will help companies to increase their profits especially in times where demand for certain items are soaring.

   A possible feature that can be added would be the option for companies to search for their competitors online through the use of Google search.

   Features of Addressbook:

   - Ability for companies to order from suppliers directly using the Addressbook.
   - They can place an order by emailing the supplier or personally visit the suppliers' warehouse to get the stocks they require. All these information are available on the supplier's profile in the application.
   - Possibility to track/outsource delivery of stocks to the warehouse to replenish the items or even direct delivery of products to clients.
   - Provide simple data analytics with the sales results (UI or calculations) so that companies can decide if they should increase their stocks for the next few weeks (if there is an increasing trend)

   Possible ways to further narrow down user profile:

   - Managing sales of vitamin products or masks
   - For pharmacy to manage sales or delivery of goods to clients

3. **For grab drivers/delivery men to track the deliveries their are allocated to**

   Problem that Addressbook will solve:

   Delivery men would need to know where the place is and when they would need to deliver the parcels by. They will also need to update their status of delivery for individual parcels. This becomes a greater problem for managers who manage a team of delivery men. For the managers, it is crucial for them to know where all the parcels currently as well as their status. For instance, will there be a delay in delivery due to unforeseen circumstances or has a particular parcel been delivered. Such updates by delivery men are currently done over messaging applications like WhatsApp, which may not be efficient when the team grows in size. Addressbook aims to solve this by allowing managers to track where the respective delivery men are at currently and where their final destinations for the parcels would be. This would be aided with the help of Google Map. Management of a team of delivery men will be much more efficient using a centralised system that Addressbook provides.

   Features of Addressbook:

   - Managers can track whether a certain delivery man has been doing their job efficiently/effectively by looking at the number of parcels delivered.
   - Furthermore, managers can also allocate the parcels to be delivered to the respective delivery man, who will see these jobs assigned in their Addressbook application. This is important when there is a parcel that needs to be urgently delivered, the manager can assign it to a delivery man who is efficient and reliable.
   - Statistics to show which areas are popular among users to get their parcel delivered. This allows the manager to bunch different orders together before allocating it to different delivery men so that their delivery speed would be much faster.
   - Additional features could be a customer loyalty program to see which users have been using their delivery services extensively.
   - Could have two tabs in the application, one to manage the deliveries and one to manage the drivers

   Possible ways to further narrow down user profile:

   - Medication delivery
   - For hospitals to manage delivery of medical supplies

**Final Decision for selection of target audience and problem addressed:** target audience listed in **Point 2**. **Reasons:** Flexible margin to play around with other additional features, allowing us to differentiate our final Addressbook product from other teams' submission.

3. Delegation of roles to respective team members

   **Minute manager/assistant draft manager:** record minutes during meeting, manage collaborative doc, work closely with draft manager, update user/developer docs not specific to a feature(**Yu Ting**)

   **Git/code manager**: In charge of Github/code structure and file management: e.g draft README, AboutUs; Check on LoC contribution/Git frequency, milestone management; (**Zheng Wei**)

   **Contact/info manager**: In charge of communication issues (Setup channel/deal with emergency situation), talk with every member before meeting about their progress and promptly update any lag of progress/course info to the team (**Zhenlin**)

   **Draft manager**: Responsible for user stories, user guide, UML diagram, developer guide management, preparation for demo video (**Jeffrey**)

   **Review/analysis manager**: In charge of final checkup of everything (most accountable person per se.), e.g: (final code/report/model/release review and submission, ) code and documentation/user guides, in charge of release management (**Qin Liang**)

**2nd Meeting: Week 4 (31/08/2020)**

**Agenda:**

- Finalise Product Name, Target User Profile and Value Proposition for submission on TEAMMATES

**Discussion Notes:**

1. Naming ideas:
    - WhatsHap
    - SalesPower
    - maSales
    - Medi-livery
    - **CLI-nic (Chosen)**

2. Target user profile:
   Tech savvy manager of a company that focuses on selling medical supplies (e.g. masks/vitamins/medicine). The company would source for medical supplies from various suppliers (locally and overseas).
    - Tech savvy
    - Require fast access of medical supply condition
    - Dislikes having to contact multiple suppliers separately; prefers a centralised place
3. Value proposition
    - Solves the **problem** of having to call/email **different suppliers** to enquire about the **current stocks** they have in the **supplier's warehouse**. CLI-nic allows users to see **all available stocks** and **locations** for a **particular medical supply** from **various suppliers** using **one** command in CLI-nic.
    - In addition, users are able to **place an order** using CLI-nic as well — with **one** supplier or with **multiple** suppliers. In the event that a particular supplier does not have enough stocks to fulfil the user's demand, CLI-nic will inform the user and **provide a list of suggestions** to make up the shortfall. If the manager is in charge of **multiple branches**, the manager will be able to check which particular branch needs a **certain supply** of product as well.
    - During a pandemic, medical supplies could run out of stock **really quickly**. CLI-nic allows the manager to analyse which products are likely to be **in-demand** and make suitable planning for **future pre-orders**.

**Feedback from Tutorial:**

- May need a database to store the data, API data calls needed, may not work without internet access
- There is a single user constraint (this can be a bit blurry)

**3rd Meeting: Week 5 (07/09/2020)**

**Agenda:**

- Discuss Target Persona and Problem Scope
- Discuss/select relevant user stories
- Allocation of iP Extension feature for each member

**Discussion Notes:**

1. Target Persona and Problem Scope

   **Target Persona**

   John is the manager of a medical supplies company for 3 warehouses across the country. He is a tech-savvy manager that prefers typing compared to clicking since he is always in a rush to source for supplies. He has to retrieve suppliers contacts so that he can contact them to restock medical supplies when they run low at various stores. These contacts are stored in an excel sheet, separated by different countries. He also has to keep track of how much supplies each store has by calling every store to check at the end of the week and note down on separate excel sheets the sales of each store. He wants things to be efficient by reducing the number of applications (referring to email, excel sheets, reports etc.) he needs to use and automating as many components as possible (reminder for low stocks, auto placing of order if feasible because of login details and all). He wants to access the most popular items and their suppliers' contact quickly. This helps him get promoted faster by ensuring a higher sales for each product.

   **Problem Scope**

   CLI-nic will keep track of details for each product as well as the product suppliers/stores contact details. In addition, CLI-nic helps John to retrieve and identify key information quickly. It also offers insights on product stock variations such as sales data. However, CLI-nic does not allow John to contact the suppliers and he has to call or email them individually. There will also not be any real-time updates for the number of stocks held by suppliers and John has to update them manually. Stocks at the warehouses will not be updated real-time as well. Instead, employees will have to update John at the end of every week. Lastly, CLI-nic does not allow users to immediately place an order to suppliers or schedule a delivery to the stores.

2. Selecting relevant user stories for v1.2:
   Link to Google Sheets documenting user stories sorted according to **priorities**:
   User Stories for v1.2

3. Allocation of iP extension features:
   Zhen Lin (Criss): **Reminders**
   Yu Ting: **Tagging**
   Qin Liang: **FriendlierSyntax**
   Zheng Wei: **DetectDuplicates**
   Jeffrey: **Help**

**4th Meeting: Week 6 (14/09/2020)**

**Agenda:**

- Discuss and finalise feature list for v1.2
- Discuss and finalise User Guide for v1.2 submission
- Allocation of feature each member will be working on

**Discussion Notes:**

1. Discussion of feature list CLI-nic v1.2

   **Feature List for v1.2**

   - Access command list/user guide
   - Add a new supplier/product/stores contacts entry
   - Create purchase order
   - Delete a supplier/product/stores contacts entry
   - Find a supplier/product by name
   - View suppliers/product/stores details (remarks)

   **Good to have for v1.2**

   - Find supplier/product by their ID
   - Find a suppliers of a certain product
   - Find out the delivery fees of the different suppliers

2. Discussion of User Guide CLI-nic v1.2

# User Guide

CLI-nic is a desktop application used to help medical product sales managers keep track of medical products and storage. The user interacts with it using a CLI, and it has a GUI created with JavaFX. It is optimized for these managers to update product conditions and access critical product information quickly via fast typing. It is written in Java, and has about __ kLoC.

CLI-nic is **a desktop application to help medical product sales managers keep track of medical products and storage**. It is optimized for these managers to **update product conditions and access critical product information quickly via fast typing**. It is written in Java, and has about 10 kLoC.

CLI-nic is **a desktop application to help medical product sales managers keep track of medical products and storage**. It is optimized for these managers to **update product conditions and access critical product information quickly via fast typing**. It is written in Java, and has about 10 kLoC.

Features

- Viewing help : help

# Features

Notes about the command format:

1. Words in UPPER_CASE are the parameters to be supplied by the user.

2. Items in square brackets [] are optional.

3. Items with … after them can be used multiple times including zero times.

### Accessing the command list/user guide : help

Displays a list of available commands and their utility description.

Narrows down to a specific command and its actual input format and samples if specified. **Format**: **help [COMMAND]**

**Examples**:

help : Displays the entire list of commands and their description

help add : Displays the detailed description, input format and an input example of add command

### Adding a warehouse : add

Adds warehouse to the CLI-nic application.

**Format**:

**1. add w/WAREHOUSE_NAME p/CONTACT_NUMBER addr/ADDRESS [wn/WAREHOUSE_NOTE]**

**Examples**:

add w/warehouseA p/00000000 addr/John street, block 123, #01-01 wn/First warehouse : Adds a warehouse with the name warehouseA located at John street, block 123, #01-01. The warehouse is noted to be the “First warehouse”.

## Adding a supplier : add

Adds a supplier to the CLI-nic application.

**Format**:

**1. add s/SUPPLIER_NAME p/CONTACT_NUMBER [e/EMAIL_ADDRESS] [sn/SUPPLIER_NOTE]**

**Examples**:

add s/Philips Pharmaceutical p/00000000 e/philipsPharm@gmail.com sn/largest contractor : Adds a supplier named Philips Pharmaceutical. His contact number is 00000000 and his email is philipsPharm@gmail.com. The supplier is noted to be the “largest contractor”.

## Adding a product to a supplier : add

Adds product information to a supplier; associates a particular product with the supplier.

**Format**:

**add s/SUPPLIER_NAME p/PRODUCT_NAME [t/TAG...]**

**Examples**:

add s/SupplierA p/PANADOL SUSP t/FEVER: Adds the product PANADOL SUSP to list of products from supplierA. This indicates that supplierA is selling this product. PANADOL SUSP also has a tag of FEVER.

## Update the stock for a product: update

If the product does not exist for that store, it will associate the new product with the store and the input quantity. Otherwise, it will update the stock of the existing product with the new quantity.

**Format:**

**update w/WAREHOUSE_NAME p/PRODUCT_NAME q/QUANTITY**

**Example:**
update w/WarehouseA p/Panadol q/10 : Updates the quantity of Panadol in WarehouseA to 10. The quantity of Panadol in WarehouseA can be more than 10 or lesser than 10 before the update is done.

~~**Creating a purchase order : create**~~

~~Create a purchase order to track the purchase of medical products from a supplier to a store.~~

~~**Format**: **create sid/SUPPLIER_ID s/STORE_ID id/PRODUCT_ID… qty/PRODUCT_QUANTITY… date/EXPECTED_DELIVERED_DATE**~~

- ~~The number specified for PRODUCT_ID **cannot exceed the total number of products**. All the IDs must be identifiable~~
- ~~The number of arguments specified for PRODUCT_QUANTITY and PRODUCT_ID **must match**~~
- ~~EXPECTED_DELIVERED_DATE **must be after current time and of the form *YYYY-MM-DD***~~

~~**Examples**:~~
~~create sid/01 s/123 id/1 2 4 8 qty/100 200 400 800 date/2020-12-12 : Creates a purchase order for the delivery of 100, 200, 400 and 800 of products with ID 1, 2, 4 and 8 respectively from supplier ID of 01 to store ID of 123 by December 12, 2020.~~

**Deletes a particular warehouse or supplier : delete**

Delete entries of warehouses or suppliers that are not needed anymore.

**Format**: **delete TYPE INDEX**

- The TYPE specified should be one of these values: **warehouse / supplier**
- The INDEX **must be a positive integer**, not exceeding the total number of items

**Examples**:

delete warehouse 1 : removes the warehouse at index 1.

delete supplier 12 : Removes supplier at index 12 from the list of suppliers.

**Finding key information : find**

Finds all suppliers or warehouses managed by the manager that sells the specified medical product.

**Format**: **find PRODUCT TYPE**

- PRODUCT and KEYWORD specified is case-insensitive.
- The TYPE specified should be one of these values: **warehouse / supplier**

**Examples**:

find PANADOL warehouse: displays all the warehouses managed by the manager that has a product named PANADOL.

find masks supplier: displays all the suppliers that have stock for the input product.

**View a specific supplier / warehouse: view**

Shows a particular supplier/warehouse with their relevant information e.g. products associated with the supplier/warehouse, address etc.

**Format: view TYPE NAME**

- The TYPE specified should be one of these values: **supplier or warehouse**
- NAME specified is case-insensitive

**Examples:**

view supplier supplierA: displays all the information associated with supplierA e.g. address, contact, email, products sold by the supplier etc.

view warehouse warehouseB: displays all the information associated with warehouseB e.g. address, all the products stored in the warehouse etc.

**Saving the data**

CLI-nic data are saved in the hard disk every time the lists are updated.

3. Allocation of feature each member will be working on

**Features:**

1. Help + List (**Yu Ting**) & Creation of 2 new classes
2. Add (**Jeffrey**)
3. Create (**Qin Liang**)
4. Delete (**Zhen Lin**)
5. Find (**Zheng Wei**)

**Possible new classes required:**

1. Supplier
2. Warehouse
3. Products
4. ~~Order~~

**5th Meeting: Week 7 (21/09/2020)**

**Agenda:**

- Do practice iterations for v1.1
- Update AboutUs page

**Discussion Notes:**

1. Do practice iterations for v1.1
   Practice workflow for an iteration, where each member created a pull request individually from their own fork. Afterwards, PR review was done and the pull request was then merged into the master branch on Team's repo. Lastly, an update was done on individual's fork and local repo by following the instructions stated in the website.

2. Update AboutUs page
   AboutUs page was updated with our individual photo and portfolio on the project's website.

**Things to be completed before next meeting (25/09/2020, 2pm)**

1. Update of User Guide for features each member is in charge of as allocated previously. (All members)
2. Update of Developer's Guide to include target user profile, value proposition, and user stories. (Yu Ting)
3. Update site-wide settings and README page (excluding UI image) (Zhenlin)
4. Brainstorm for Use Cases, Non-functional requirements, glossary

**6th Meeting: Week 7 (25/09/2020)**

**Agenda:**

- Update team webpage
- Update User Guide
- Update Developer Guide (Product scope, User Stories, Use cases, Non-functional requirements and Glossary)

**Discussion Notes:**

1. Update team webpage
   Replaced AddressBook UI with UI of CLI-nic, ensuring badges (Java CI, codecov) are working. Updated user guide and developer guide as detailed below.

2. Update User Guide
   Merge pull requests from all members who contributed to a part of the user guide before this meeting. Practiced the workflow required to resolve merge conflicts.

3. Update Developer Guide
   Product Scope
   **Target user profile**:

   - manager of a medical supplies company that manages warehouses across the country
   - tech-savvy manager who prefers typing to clicking
   - keeps track of supplies in each warehouse
   - needs to quickly contact suppliers to restock medical supplies when the stock runs low at various warehouses

   **Value proposition**:

   - manages suppliers/warehouses information conveniently
   - helps to retrieve and identify key suppliers/warehouses information quickly

   User Stories

   Updated user stories in Developers Guide, adapted from excel sheet tabulated in previous meetings.

   Use Cases
   (For all use cases below, the **System** is the CLI-nic and the **Actor** is the user, unless specified otherwise)

****Use case: UC01 Add a warehouse****

****MSS****

1. User keys in command to add a warehouse.
2. CLI-nic adds the warehouse into the list and shows a success message.

   Use case ends.

****Extensions****

* 1a. CLI-nic detects an error in the entered warehouse info.

  * 1a1. CLI-nic shows an error message.
  * 1a2. CLI-nic requests for the correct format of warehouse info.
  * 1a3. User enters new data.

  Steps 1a1-1a3 are repeated until the data entered are correct. <br>

  Use case resumes from step 2.

****Use case: UC02 Add a supplier****

****MSS****

1. User keys in command to add a supplier.
2. CLI-nic adds the supplier into the list and shows a success message.

   Use case ends.

****Extensions****

* 1a. CLI-nic detects an error in the entered supplier info.

  * 1a1. CLI-nic shows an error message .
  * 1a2. CLI-nic requests for the correct data.
  * 1a3. User enters new data.

  Steps 1a1-1a3 are repeated until the data entered are correct.

  Use case resumes from step 2.

****Use case: UC03 Add a new product to supplier****

****MSS****

1. User keys in command to add a product.
2. CLI-nic adds the product into the list and shows a success message.

   Use case ends.

****Extensions****

* 1a. CLI-nic detects an overlap in the entered product and the existing products in the supplier's supply list.
  * 1a1. CLI-nic shows an error message.
  * 1a2. CLI-nic requests for a new product name.
  * 1a3. User enters a product name.

  Steps 1a1-1a3 are repeated until the name entered is correct. <br>
  Use case resumes from step 2.

* 1b. CLI-nic cannot find the supplier from the supplier list.
  * 1b1. CLI-nic shows an error message.
  * 1b2. CLI-nic requests for a new supplier name.
  * 1b3. User enters a new supplier name.

  Steps 1b1-1b3 are repeated until the supplier is found. <br>
  Use case resumes from step 2.

****Use case: UC04 clear all supplier and warehouse entries****

****MSS****

1. User requests to clear the data in the application.
2. CLI-nic clears all supplier and warehouse entries, shows empty lists of suppliers and warehouses and shows a success message.
   Use case ends.

****Use case: UC05 Delete a supplier****

****MSS****

1. User requests to view a specific supplier by keyword.

2. CLI-nic shows the specific supplier and its index.
3. User deletes the supplier via its index in the list.
4. CLI-nic deletes the supplier in the list and shows a success message.

   Use case ends.

**Extensions**

* 2a. The supplier list is empty.
  * 2a1. CLI-nic informs the user there is no supplier in the list currently.

Use case ends.

* 3a. The given index is invalid.
  * 3a1. CLI-nic shows an error message and gives command suggestions.
  * 3a2. User enters the new supplier index.

  Steps 3a1-3a2 are repeated until the data entered are correct.
  Use case resumes at step 4.

**Use case: UC06 Delete a warehouse**

**MSS**

1. User requests to view a specific warehouse by keyword.
2. CLI-nic shows the specific warehouse and its index.
3. User deletes the warehouse via its index in the list.
4. CLI-nic deletes the warehouse in the list and shows a success message.

   Use case ends.

**Extensions**

* 2a. The warehouse list is empty.
  *2a1. CLI-nic informs the user there is no warehouse in the list currently.

  Use case ends.

* 3a. The given index is invalid.
  * 3a1. CLI-nic shows an error message and gives command suggestions.
  * 3a2. User enters the new warehouse index.

Steps 3a1-3a2 are repeated until the data entered are correct.
Use case resumes at step 4.

****Use case: UC07 Find Suppliers of a product****

****MSS****

1. User enters the command to find the suppliers of a specific product.
2. CLI-nic displays all suppliers that sell the product if any.
3. User scrolls through all the relevant results and looks for the information that they desire.

   Use case ends.

****Extensions****

* 1a.
  * 1a1. User enters invalid command for finding.
  * 1a2. CLI-nic requests for the correct command.

  Steps 1a1 and 1a2 are repeated until a valid find command is entered.

  Use case resumes from step 2.

****Use case: UC08 Find Warehouses containing a product****

****MSS****

1. User enters the command to view a particular product stored in all warehouses.
2. CLI-nic displays all the relevant products that are stored in the warehouses if any.
3. User scrolls through all the relevant results and looks for the information that they desire.
   Use case ends.

****Extensions****

* 1a. User enters invalid command for finding.
  * 1a1. CLI-nic requests for the correct command.
  * 1a2. User enter a new command for finding.

  Steps 1a1 and 1a2 are repeated until a valid find command is entered.

Use case resumes from step 2.

**Use case: UC09 View Help**

**MSS**

1. User enters the `help` command.
2. CLI-nic displays information about all the commands and contains sample commands that the user can try out.
3. User try out commands listed under help to familiarise themselves with CLI-nic.

   Use case ends.

**Extensions**

* 1a. User asks for a specific command via the command keyword.
  * 1a1. CLI-nic displays instructions on how the command they ask can be used. Sample command call will also be given.
  * 1a2. User follows the instruction to try out command they asked to familiarise themselves with this command.

  Use case ends.

* 1b.User enters invalid help command.
  * 1b1. CLI-nic shows an error message.
  * 1b2. CLI-nic requests for the correct command.
  * 1b3. User enters a new help command.

  Steps 1b1-1b3 are repeated until a valid help command is entered.

  Use case resumes from step 2.

**Use case: UC10 Update quantity of a Product in a Warehouse**

**MSS**

1. User keys in command to update a product's quantity with a specific product, warehouse and quantity.
2. If the product exists, CLI-nic overwrites the product's quantity. Else if the product does not exist, CLI-nic adds the product and its quantity to the warehouse. CLI-nic shows a success message.

   Use case ends.

**Extensions**

* 1a. CLI-nic cannot find the warehouse.

* 1a1. CLI-nic shows an error message.
* 1a2. CLI-nic requests for a new warehouse name.
* 1a3. User enters a new name.

Steps 1a1-1a3 are repeated until the data entered are correct.

Use case resumes from step 2.

****Use case: UC11 View Supplier****

****MSS****

1. User enters command to view supplierA information.
2. CLI-nic displays all of supplierA information, including contact number and any notes such as “fast supplier”.
3. User looks through the information of supplierA displayed and might call up the supplier if necessary.

Use case ends.

****Extensions****

* 1a User enters invalid supplier name.
  * 1a1. CLI-nic requests for new supplier name.
  * 1a2. User enters a new supplier name.

  Steps 1a1 and 1a2 are repeated until a valid supplier name is entered.

Use case resumes from step 2.

****Use case: UC12 View Warehouse****

****MSS****

1. User enters command to view WarehouseA information.
2. CLI-nic displays all of WarehouseA information, including contact number and any notes such as “first warehouse”.
3. User looks through the information of WarehouseA displayed and might call up the warehouse if necessary.

Use case ends.

****Extensions****

* 1a User enters invalid warehouse name.
  * 1a1. CLI-nic requests for new warehouse name.
  * 1a2. User enters a new warehouse name.

Steps 1a1 and 1a2 are repeated until a valid warehouse name is entered.

Use case resumes from step 2.

========Coming Soon=======

****Use case: UC01 Create a purchase order****

****MSS****

1. User requests to create a purchase order with a specific supplier, store, and a list of products and their quantities
2. CLI-nic creates the purchase order

Use case ends.

****Extensions****

* 1a. Any of the given names are invalid.
  * 1a1. CLI-nic shows an error message.
  * 1a2. User enters new names for the command.

  Steps 1a1 and 1a2 are repeated until all the given names are valid.

  Use case resumes from step 2.

* 1b. The number of products names supplied and the number of
  product quantities
  given do not match.
  * 1b1. CLI-nic shows an error message.
  * 1b2. User enters the command again with the new products and quantities.

  Steps 1b1 and 1b2 are repeated until the products and their quantities match.

  Use case resumes from step 2.

* 1c. Any one of the arguments is not supplied.
* 1c1. CLI-nic shows an error message.
* 1c2. User enters the command again.

Steps 1c1 and 1c2 are repeated until all required arguments are supplied.

Use case resumes from step 2.

Non-functional requirements

1. Should work on any *mainstream OS* as long as it has Java 11 or above installed.
2. A user with above average typing speed for regular English text (i.e. not code, not system admin commands) should be able to accomplish most of the tasks faster using commands than using the mouse.
3. The files used to store information about supplier/warehouse/product should be independent from each other and follow a similar format.
4. The system should recognize common French/German letters as they may appear in the name of medical products.
5. The system should be able to track > 1k suppliers, > 1k warehouses, > 1k medical supplies without a noticeable sluggishness in performance for typical usage.
6. The size of the application excluding the data files should be minimal (< 30MB).
7. The system should work off-line.
8. The system stores data locally.

Glossary

- **Mainstream OS**: Windows, Linux, Unix, OS-X
- **Private contact detail**: A contact detail that is not meant to be shared with others
- **Medical products/supplies**: The items / tools / medicine consumed by patients
- **Supplier**: The companies / entities providing the sources of medical products
- **Warehouse**: The places where the medical supplies are channeled to and kept. The storage condition of these warehouses are managed by the manager, which is our app user

**7th Meeting: Week 8 (03/10/20)**

**Agenda:**

- Draft out a simple UML diagram for Product, Warehouse and Supplier classes
- Updating user stories on GitHub
- Allocation of tasks to members

**Discussion Notes:**

1. Draft out a simple UML diagram for Product, Warehouse and Supplier classes
   Link to UML diagrams - UML diagrams

2. Updating user stories on GitHub
   Mark any outdated user stories on GitHub as deprecated and include new user stories added to the developers guide from last meeting.

3. Allocation of tasks to members
   Qin Liang: Implement Supplier class, Update command
   Jeffrey: Add command (add supplier and warehouse information)
   Zheng Wei: Find command
   Yu Ting: Implement Product class, View and Help command
   Zhenlin: Implement Warehouse class, Add product to supplier, Delete command

   **Deadline for v1.2: 12 October 2020**

**8th Meeting Week 8 (07/10/20)**

**Agenda:**

- Discuss current progress for v1.2
- Merge in Warehouse and Supplier class

**Discussion Notes:**

1. Add sample products for test cases
   Craft test cases for Product class
   Consider how to fix getProductSet in test cases, where quantity and tags are required as input for warehouses and suppliers respectively.
2. Warehouse and Supplier classes has been created and merged with master
3. Individual members to work on their own features

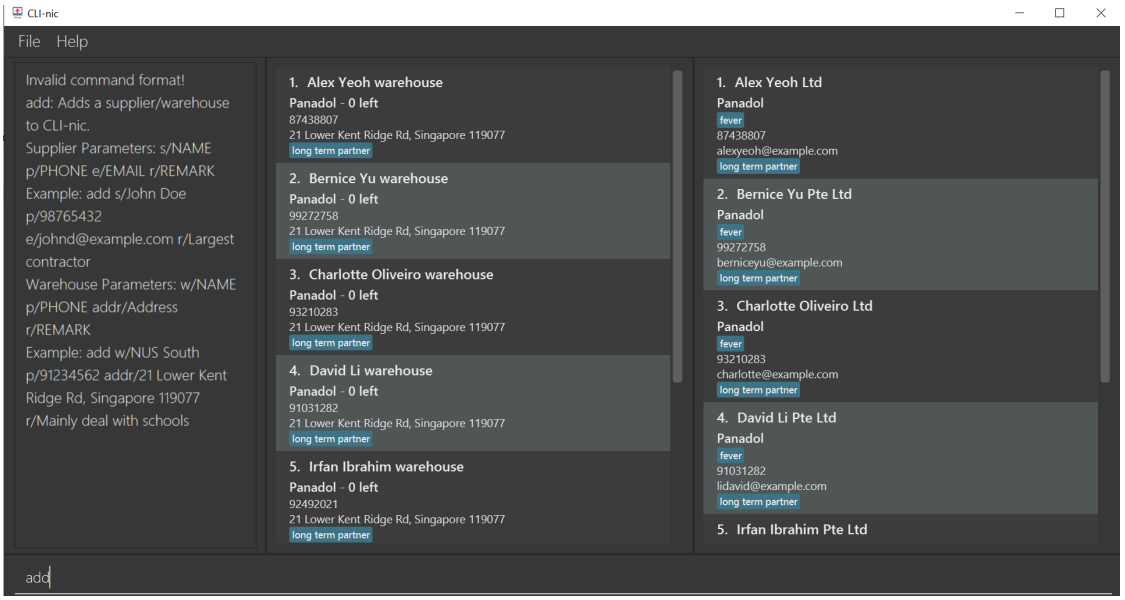
**9th Meeting Week 9 (12/10/20)**

**Agenda:**

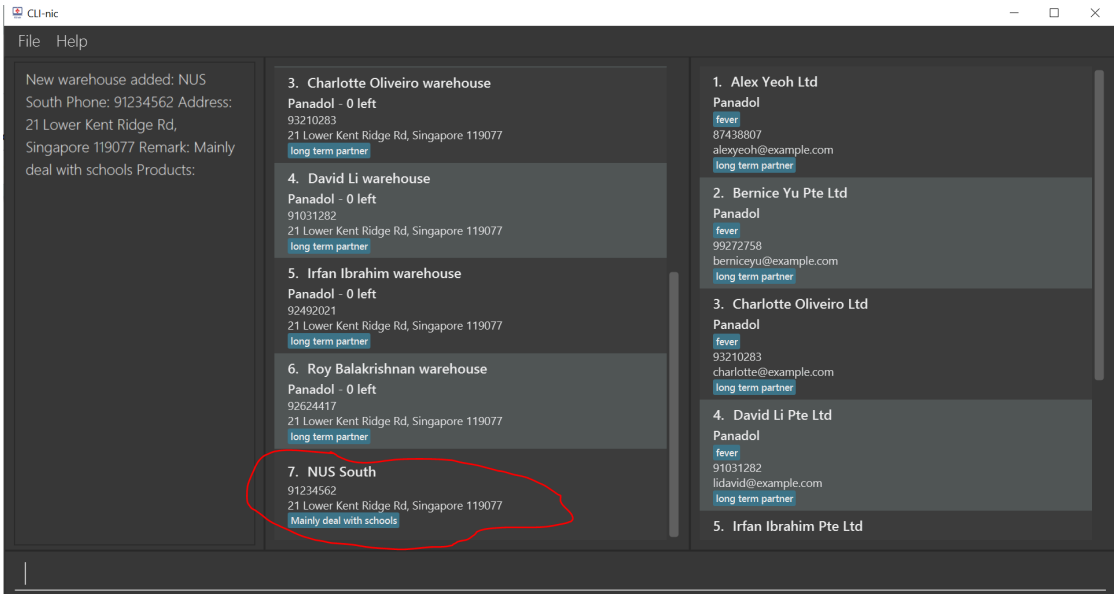
- Wrap up v1.2 (Merge in all pull requests to master branch)
- Add screenshots for tutorial Demo
- Discuss and finalise features to be implemented for v1.3. Allocation of these features to individual members

**Discussion Notes:**

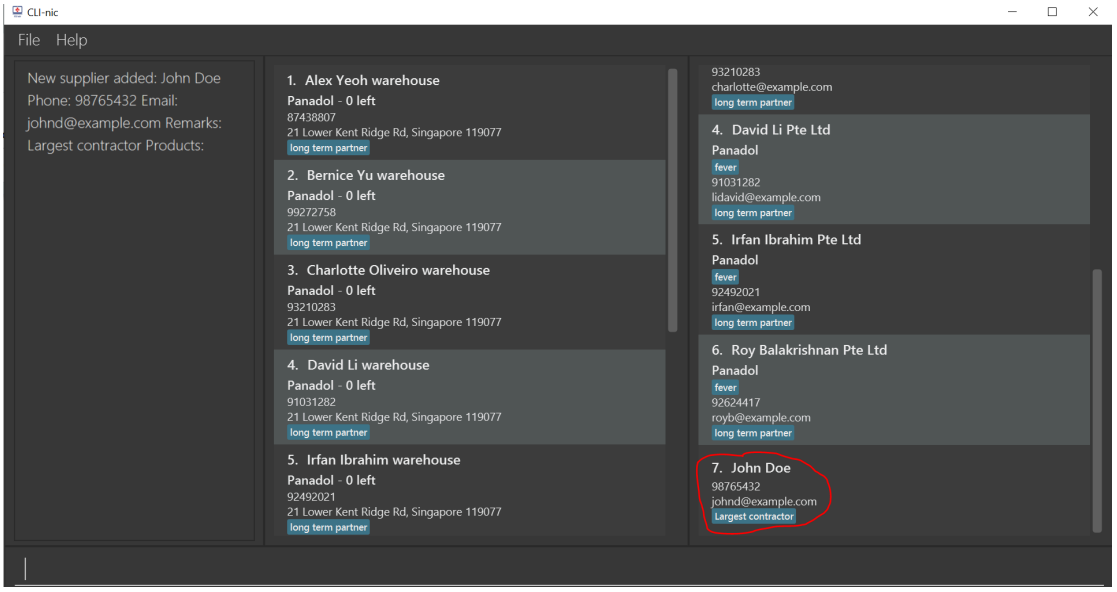
1. Screenshots for v1.2 features demo
   Add command demo



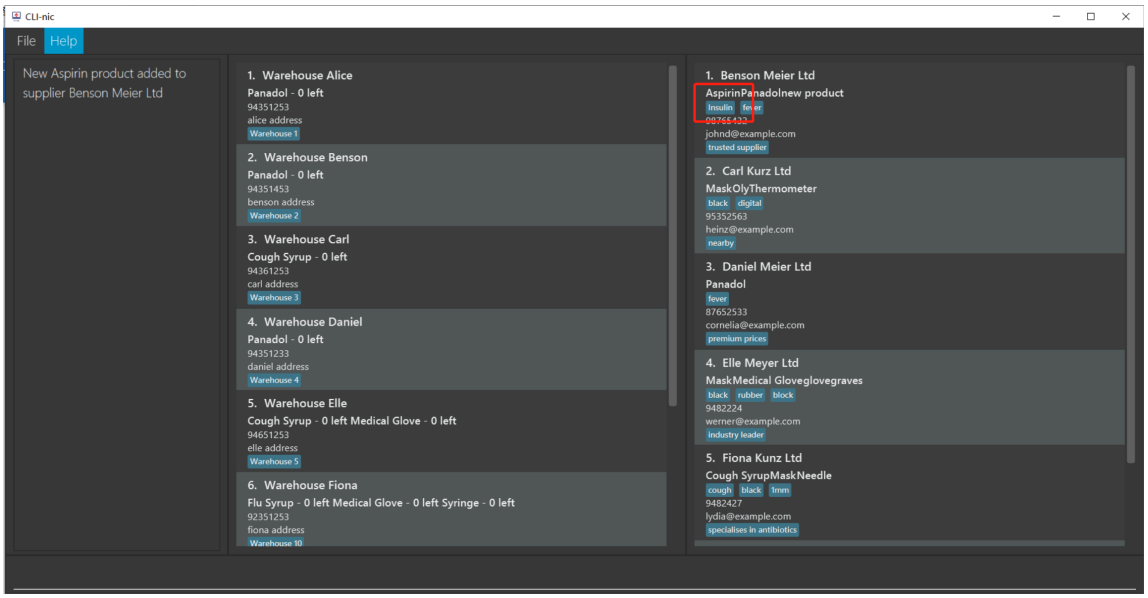
# (Add warehouse)

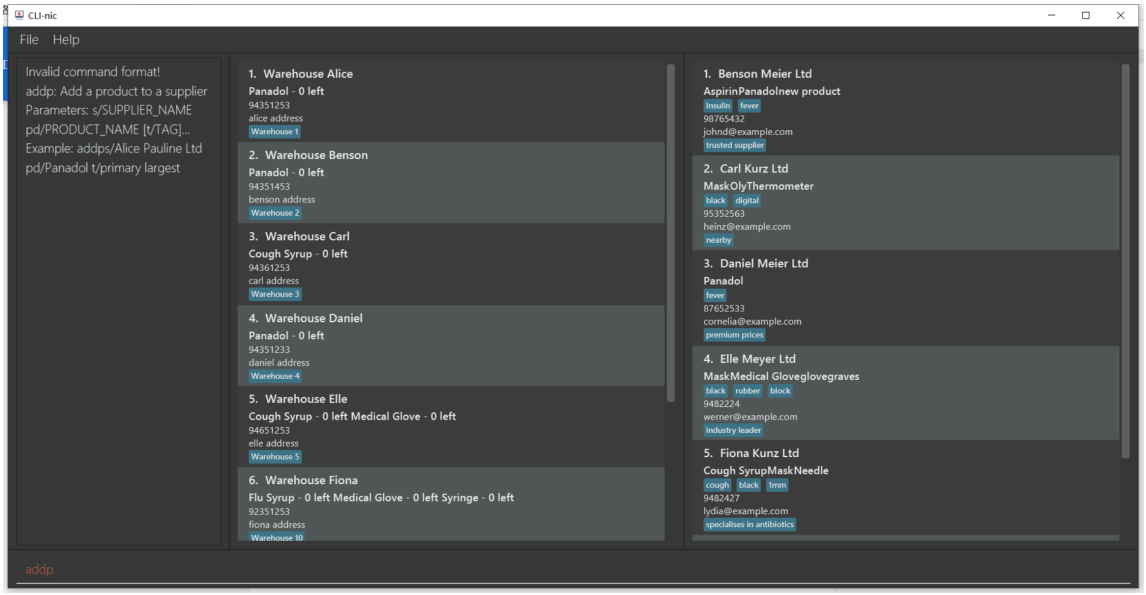



# (Add Supplier)

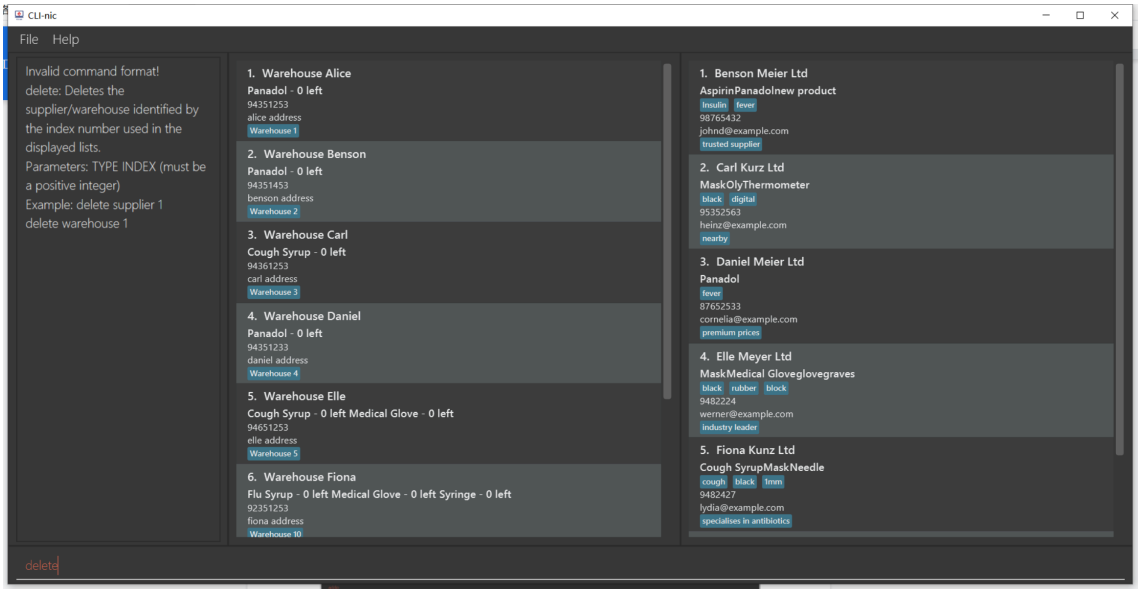


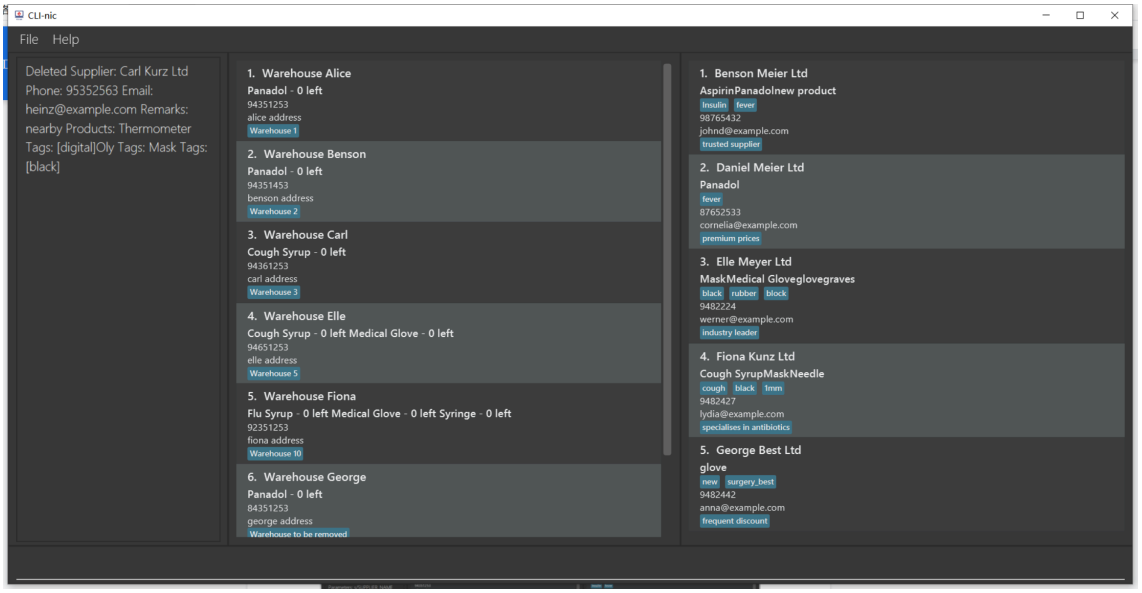
(Add Product)


CLI-nic
File
Help
New Aspirin product added to supplier Benson Meier Ltd
1. Warehouse Alice
Panadol - 0 left
94351253
alice address
Warehouse 1
2. Warehouse Benson
Panadol - 0 left
94351453
benson address
Warehouse 2
3. Warehouse Carl
Cough Syrup - 0 left
94361253
carl address
Warehouse 3
4. Warehouse Daniel
Panadol - 0 left
94351233
daniel address
Warehouse 4
5. Warehouse Elle
Cough Syrup - 0 left Medical Glove - 0 left
94651253
elle address
Warehouse 5
6. Warehouse Fiona
Flu Syrup - 0 left Medical Glove - 0 left Syringe - 0 left
92351253
fiona address
Warehouse 10
1. Benson Meier Ltd
AspirinPanadolnew product
Insulin
fever
johnd@example.com
trusted supplier
2. Carl Kurz Ltd
MaskOlyThermometer
black
digital
95352563
heinz@example.com
nearby
3. Daniel Meier Ltd
Panadol
fever
87652533
cornelia@example.com
premium prices
4. Elle Meyer Ltd
MaskMedical Gloveglovegraves
black
rubber
block
9482224
werner@example.com
industry leader
5. Fiona Kunz Ltd
Cough SyrupMaskNeedle
cough
black
1mm
9482427
lydia@example.com
specialises in antibiotics

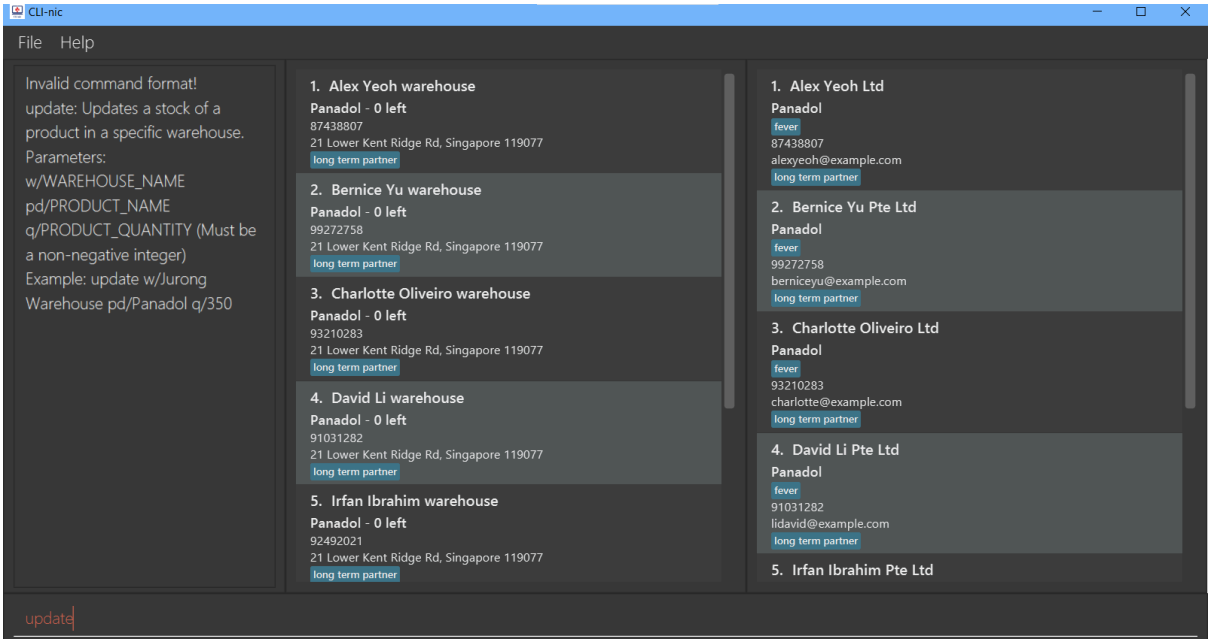

CLI-nic
File
Help
Invalid command format!
addp: Add a product to a supplier
Parameters: s/SUPPLIER_NAME pd/PRODUCT_NAME [t/TAG]...
Example: addps/Alice Pauline Ltd pd/Panadol t/primary largest
1. Warehouse Alice
Panadol - 0 left
94351253
alice address
Warehouse 1
2. Warehouse Benson
Panadol - 0 left
94351453
benson address
Warehouse 2
3. Warehouse Carl
Cough Syrup - 0 left
94361253
carl address
Warehouse 3
4. Warehouse Daniel
Panadol - 0 left
94351233
daniel address
Warehouse 4
5. Warehouse Elle
Cough Syrup - 0 left Medical Glove - 0 left
94651253
elle address
Warehouse 5
6. Warehouse Fiona
Flu Syrup - 0 left Medical Glove - 0 left Syringe - 0 left
92351253
fiona address
Warehouse 10
1. Benson Meier Ltd
AspirinPanadolnew product
Insulin
fever
98765432
johnd@example.com
trusted supplier
2. Carl Kurz Ltd
MaskOlyThermometer
black
digital
95352563
heinz@example.com
nearby
3. Daniel Meier Ltd
Panadol
fever
87652533
cornelia@example.com
premium prices
4. Elle Meyer Ltd
MaskMedical Gloveglovegraves
black
rubber
block
9482224
werner@example.com
industry leader
5. Fiona Kunz Ltd
Cough SyrupMaskNeedle
cough
black
1mm
9482427
lydia@example.com
specialises in antibiotics
addp

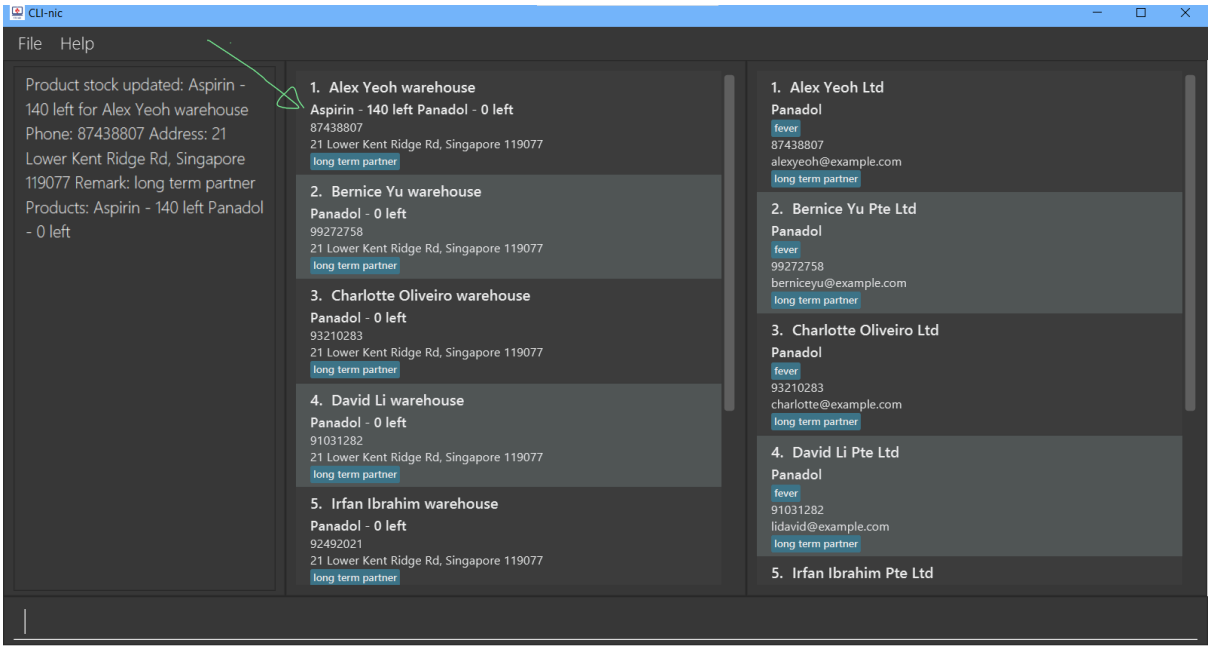
# Delete command demo



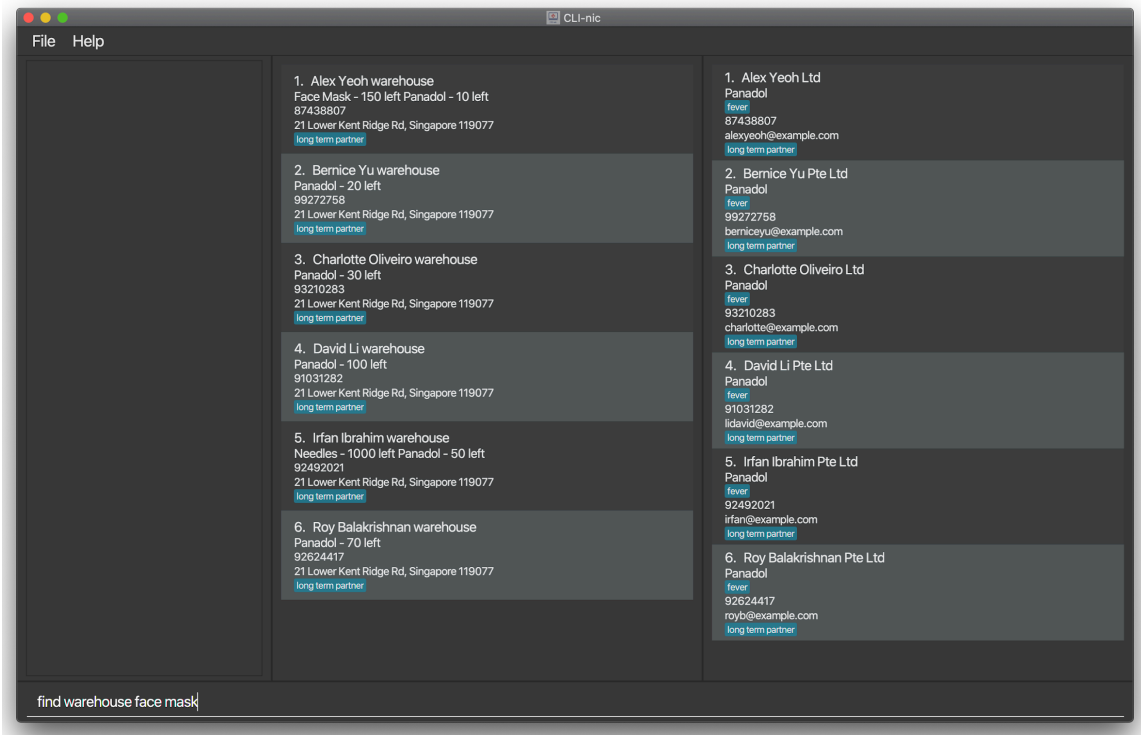



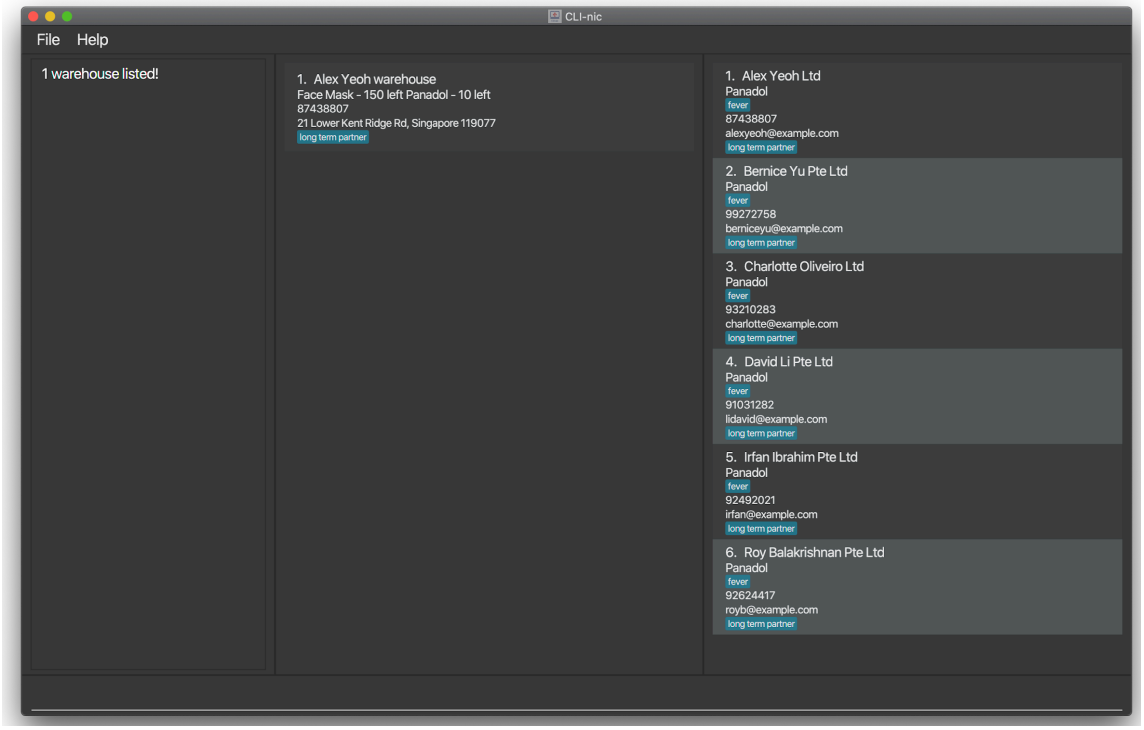
# Update command demo



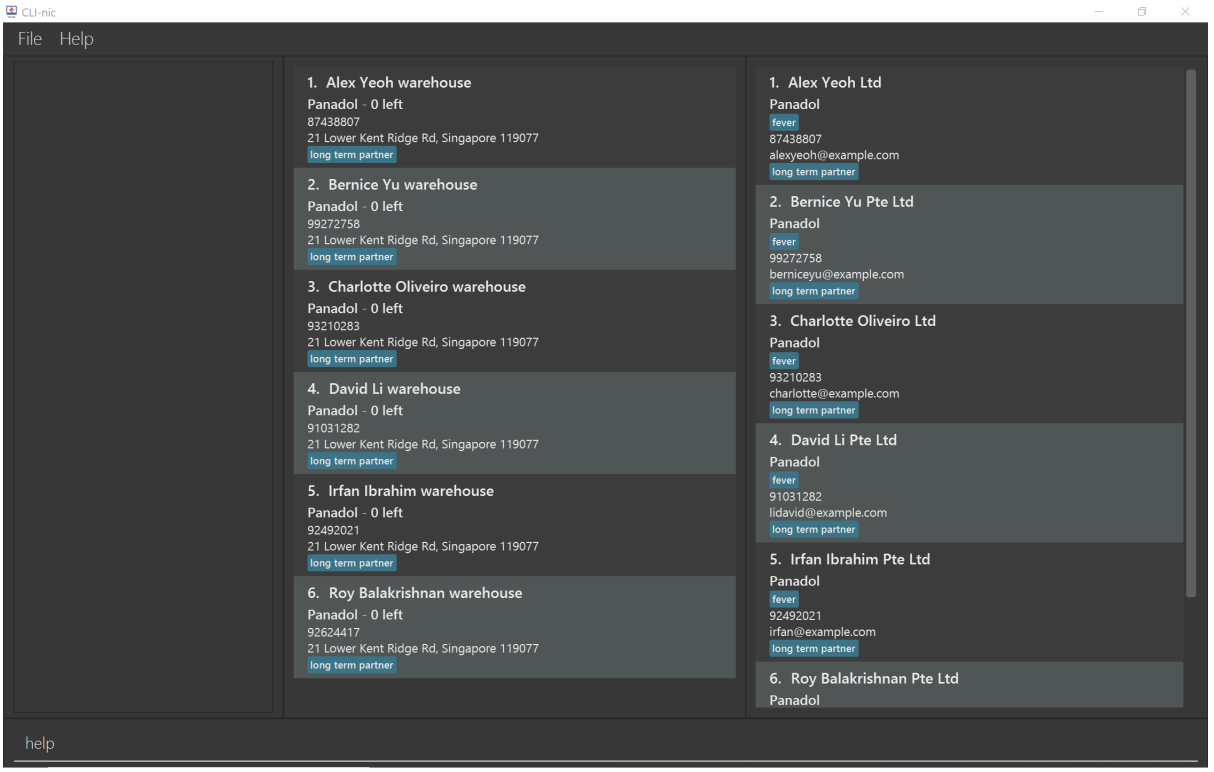



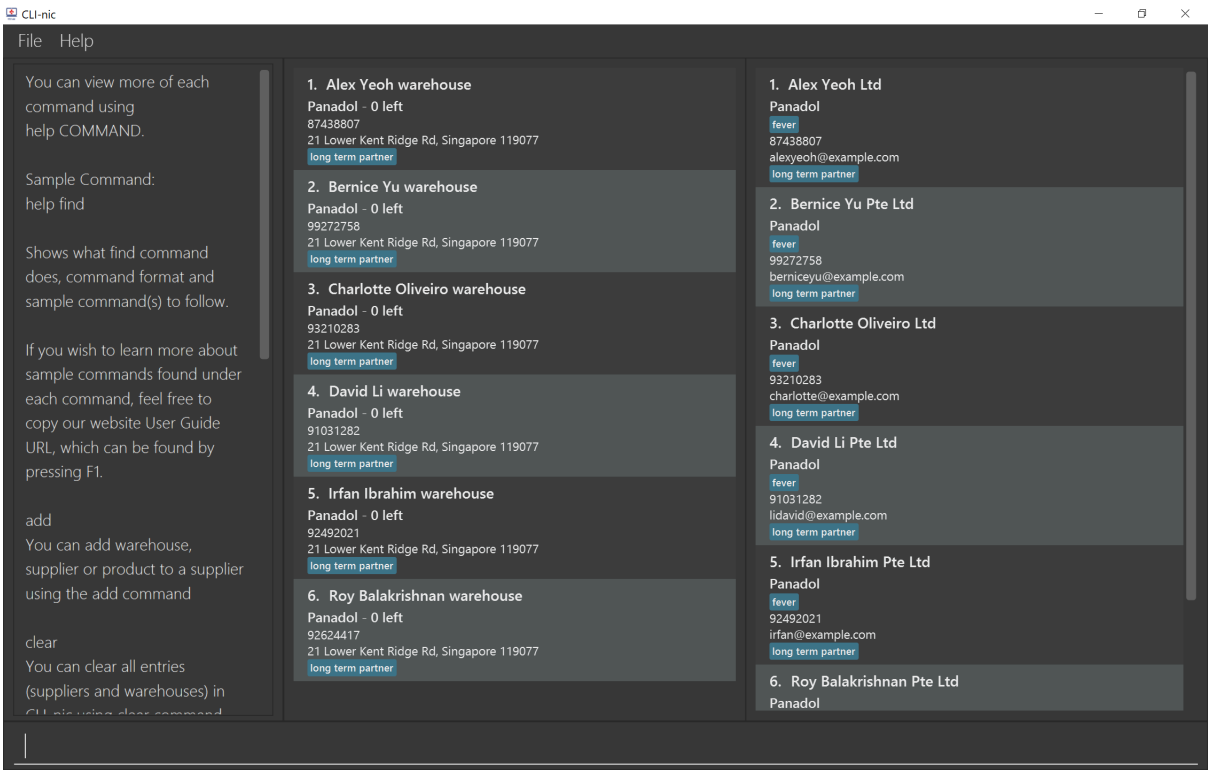
# Find command demo



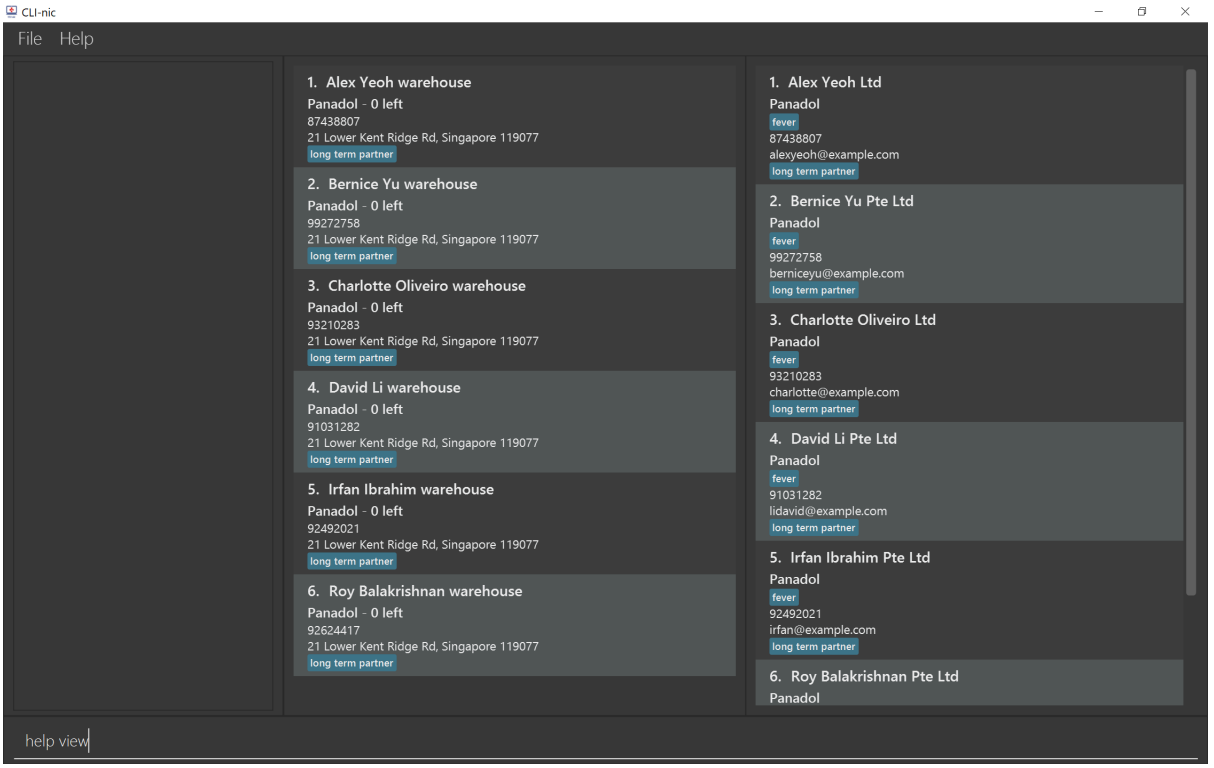



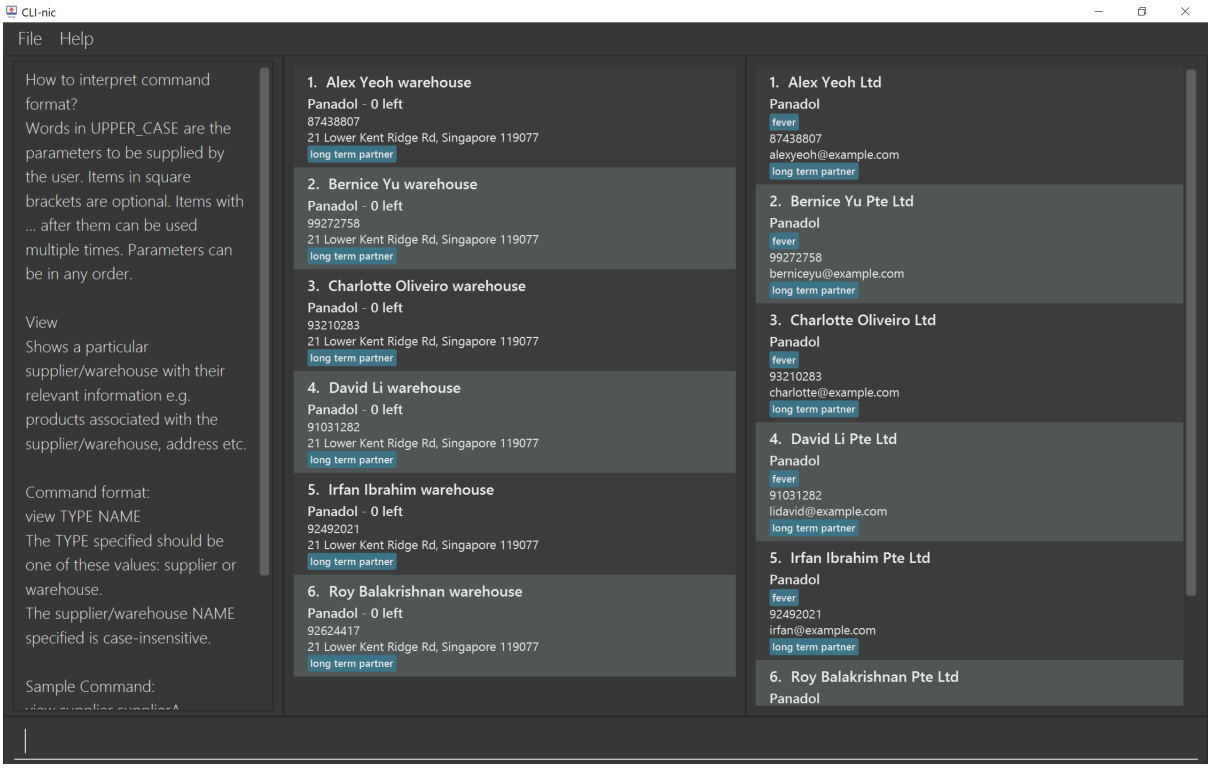
# Help command demo



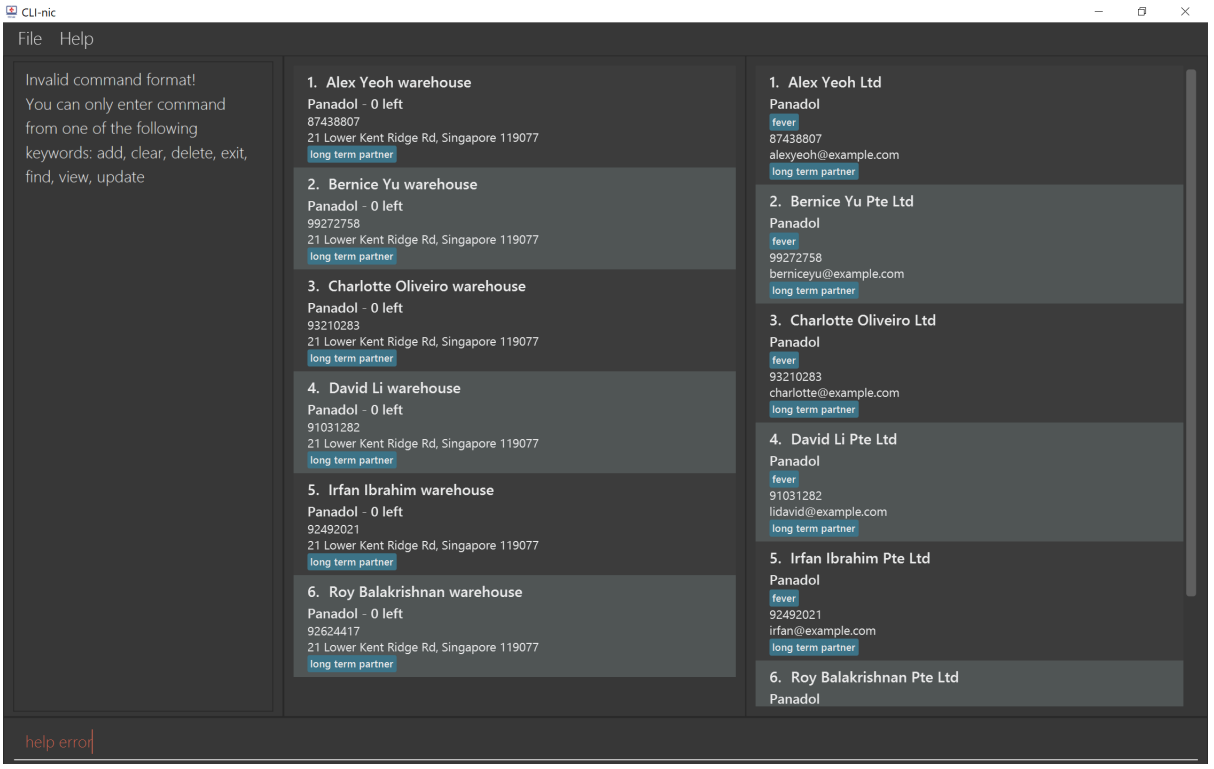



# Help view (help COMMAND used)

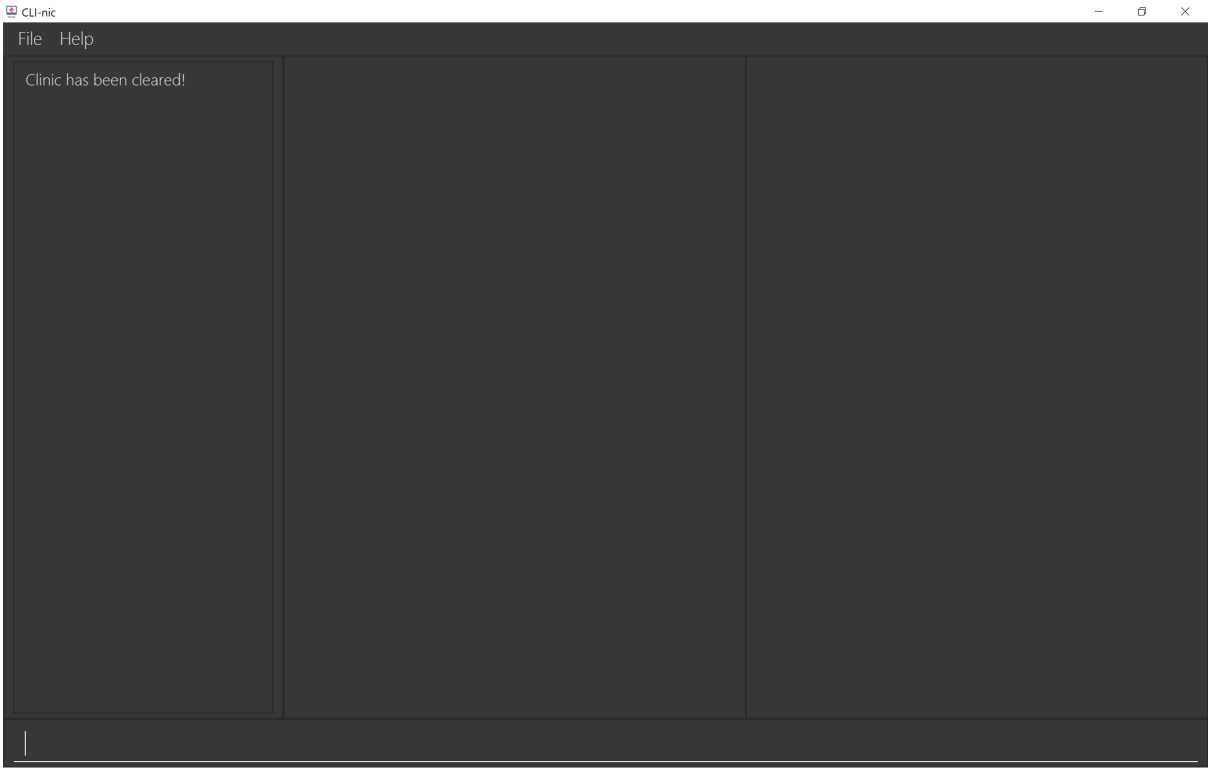


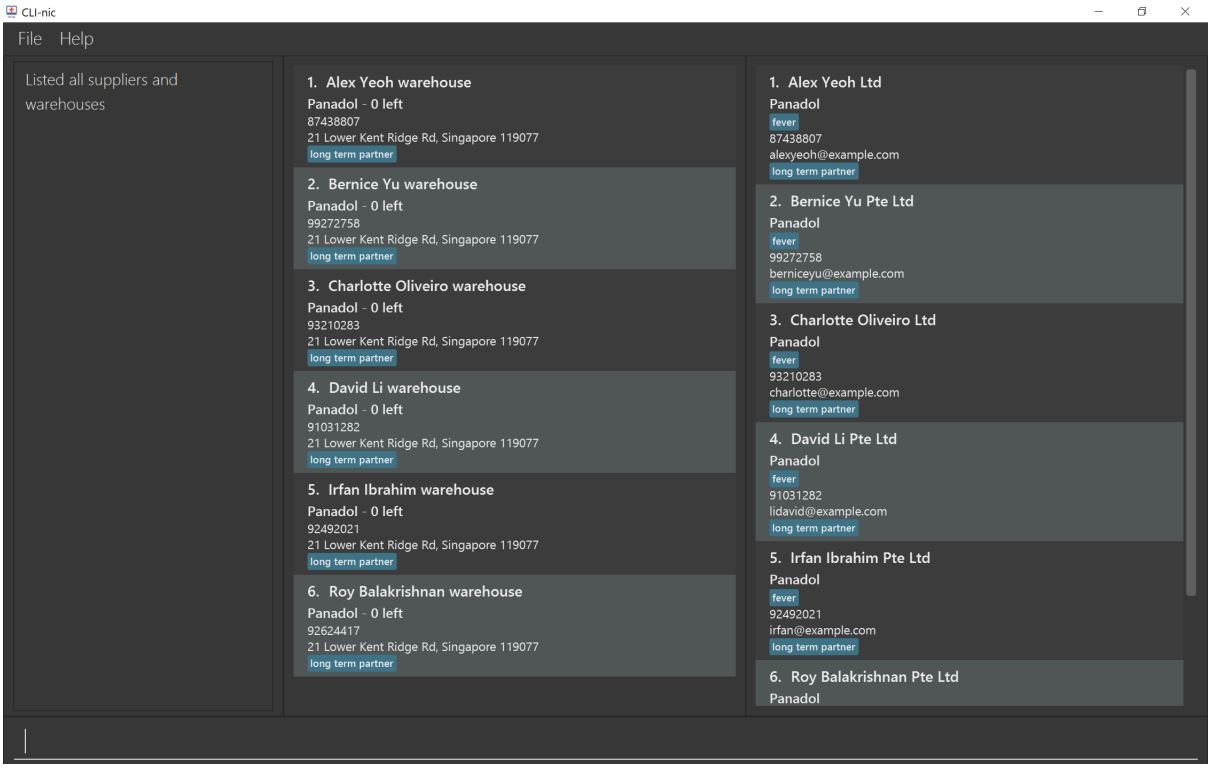
# Invalid help command



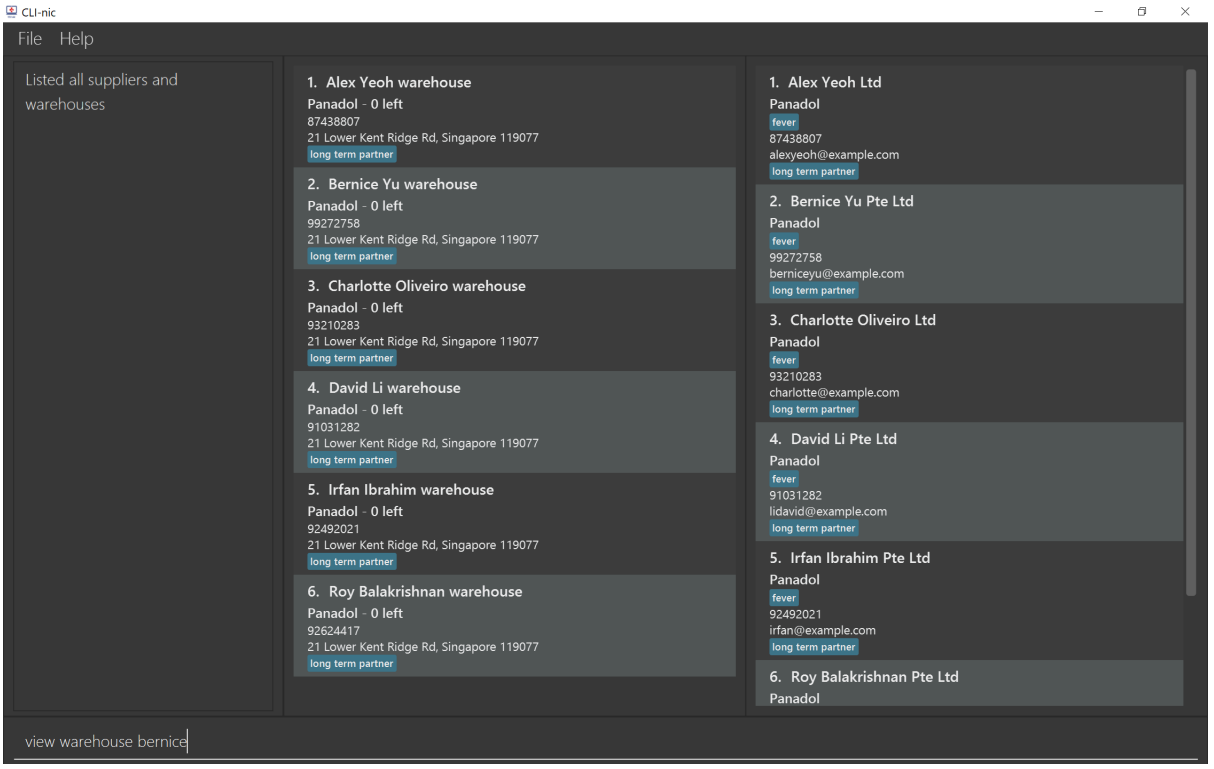

# Clear command

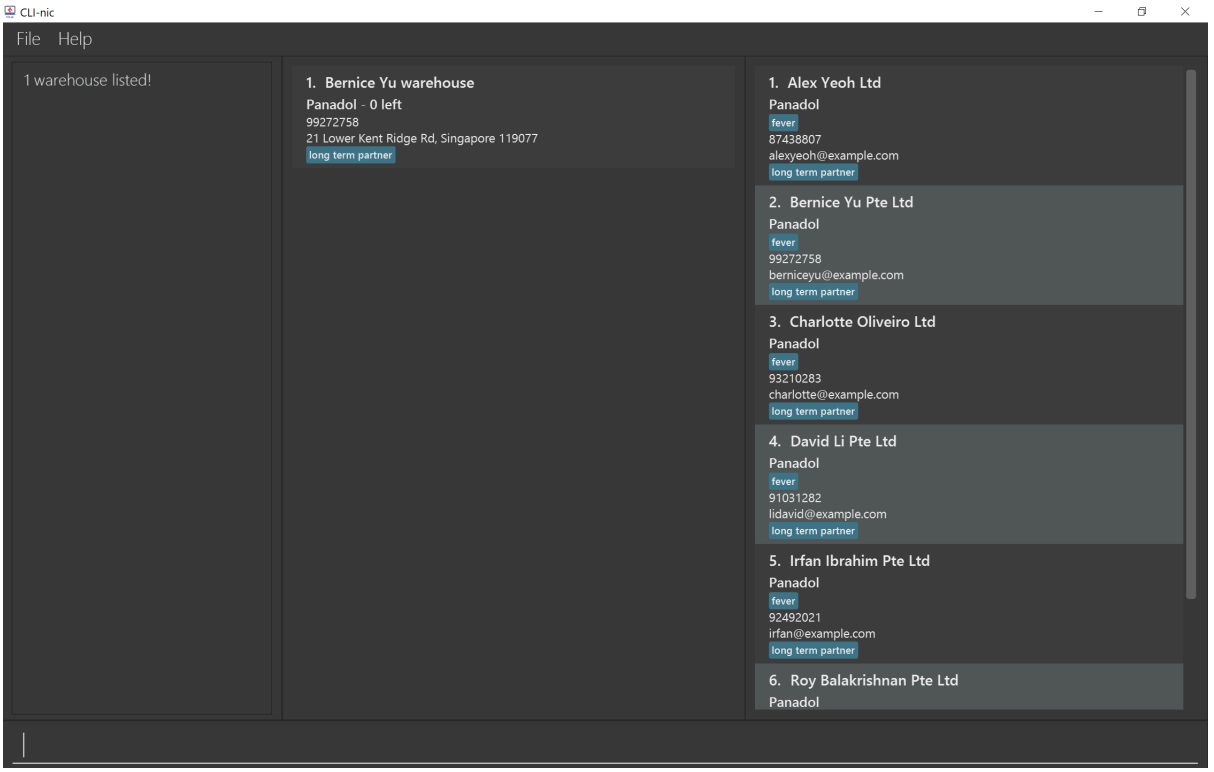


# List command

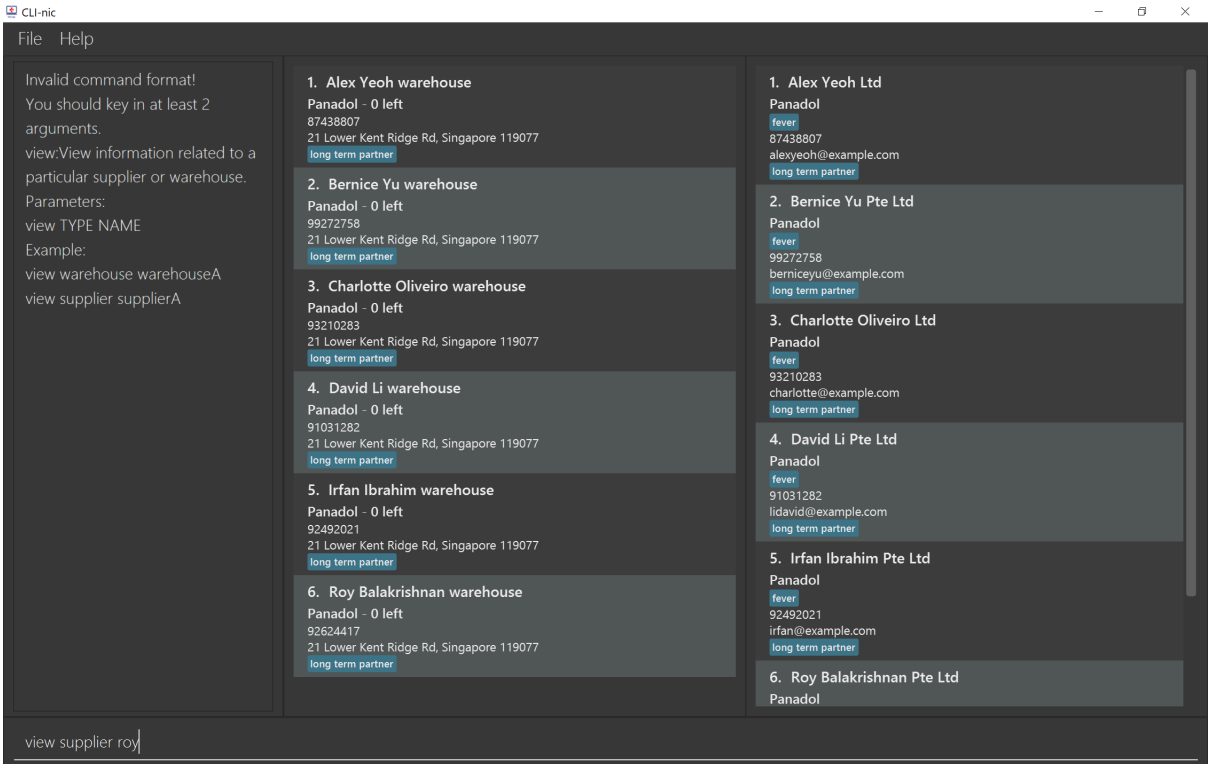


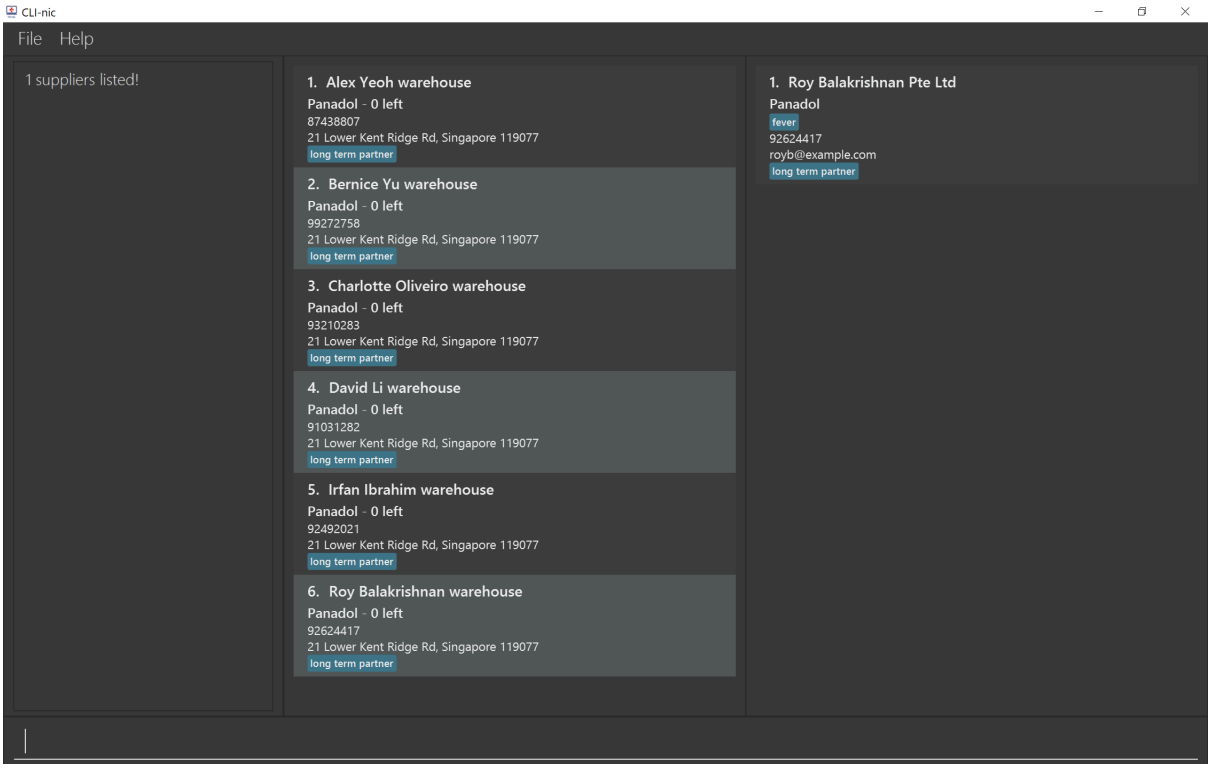

# View warehouse






# View supplier




CLI-nic
File
Help
1 suppliers listed!
1. Alex Yeoh warehouse
Panadol - 0 left
87438807
21 Lower Kent Ridge Rd, Singapore 119077
long term partner
2. Bernice Yu warehouse
Panadol - 0 left
99272758
21 Lower Kent Ridge Rd, Singapore 119077
long term partner
3. Charlotte Oliveiro warehouse
Panadol - 0 left
93210283
21 Lower Kent Ridge Rd, Singapore 119077
long term partner
4. David Li warehouse
Panadol - 0 left
91031282
21 Lower Kent Ridge Rd, Singapore 119077
long term partner
5. Irfan Ibrahim warehouse
Panadol - 0 left
92492021
21 Lower Kent Ridge Rd, Singapore 119077
long term partner
6. Roy Balakrishnan warehouse
Panadol - 0 left
92624417
21 Lower Kent Ridge Rd, Singapore 119077
long term partner
1. Roy Balakrishnan Pte Ltd
Panadol
fever
92624417
royb@example.com
long term partner

2. Iteration v1.3

   Features to be implemented:

   1. Edit Command (**Yu Ting**)

      Covers editing of details for warehouse and suppliers, excluding editing of product details (products can only be updated by their quantity via update command for warehouse)
   2. Delete products (**Zhenlin**)

      Covers deletion of products for both warehouses and suppliers.
   3. Autocomplete feature for commands keyed in by user (**Jeffrey**)

      Enhances convenience for end-user
   4. Expand find command to allow users to find suppliers and warehouses by remarks and names (**Zheng Wei**)

      Allows end user to retrieve warehouse and supplier information quickly
   5. Save a sequence of tasks into keyboard macro (**Qin Liang**)

      More convenience for end user
   6. Work on UI improvement and documentations

      All members will update the portion of developer guide they are responsible for (i.e. individual commands, user stories and use cases)

      All members will also update individual contributions under the About Us page.

      **Jeffrey** will be in charge of improving the UI, **Zheng Wei** offered to help as well.

      **Qin Liang, Yu Ting** will also be helping Jeffrey with the documentations (UML diagrams, clean up of other parts of developer guide)

   All features to be done by the next meeting (19 October 2020, 4pm). 2nd week of iteration v1.3 will be for cleaning up of code (improving code quality), testing, enhancing overall quality of UG and DG.

**<u>10th Meeting Week 10 (19/10/20)</u>**

**Agenda:**

- Post mortem of the previous iteration
- Updates from members regarding features for iteration v1.3
- Finalise user input structure for commands

**Discussion Notes:**

1. Post mortem of the previous iteration
    a. Forking workflow works out well for us (minimise any merge conflicts etc), and will be keeping it for the next few iterations.
    b. All PRs were promptly reviewed
    c. Did not link certain issues to the relevant PRs. As a result, we forgot to close some of the related issues even after the PR was closed
    d. Most tests were added before the PR was merged to the master
    e. Generally, the previous iteration worked well as everything was done promptly and on time. Hence, we decide to continue with the current workflow for the next iteration, v1.3.
2. Updates from members regarding features for iteration v1.3
   Each member's responsibilities for iteration v1.3 is recorded down in the previous meeting minutes (**8th Meeting Week 9, 12/10/20, under subpoint, Iteration v1.3**)

   Every member is coping well so far and will continue to enhance our individual features and update the relevant documentations by next Sunday night.

3. Finalise user input structure for commands
   Our team has decided to make minor changes for command input of the relevant features. This standardised commands throughout the application will allow users to use the application with greater ease and familiarity over time. Some of the changes to the features commands are shown below:

   **add command**
   supplier:
   add ct/w n/warehouseA p/00000000 addr/John street, block 123, #01-01 r/First warehouse
   add ct/s n/Philips Pharmaceutical p/00000000 e/philipsPharm@gmail.com r/largest contractor

**delete command**
delete ct/w i/INDEX
delete ct/s i/INDEX
delete ct/pw i/INDEX [pd/Panadol]
delete ct/ps i/INDEX pd/Panadol

**edit command**:
edit  ct/w i/ n/ p/ addr/ r/
edit  ct/s i/ n/ p/ e/ r/

**view command:**
view ct/s i/1
view ct/w i/1

**find command:**
find ct/s(w) [r/remark1 remark2]  [pd/panadol face mask] [n/roy alice charles]

**update command:**
update ct/TYPE n/name pd/PRODUCT_NAME [q/QTY] [t/TAG]
=> qty and tags optional if the product is not in list
=> throws an error if product is in list and no qty or tag provided (or mismatch of types). If qty or tags are provided, an error will be thrown if the resulting qty or tags are unchanged

**Other housekeeping issues:**
removing / in name, address,email,remarks
create issues for bugs commented
Code standard:
Code quality:
May need to examine the accessibility of functions
Some redundant functions
The SOLID principles not followed
Test:

**11th Meeting Week 11 (25/10/20)**

**Agenda:**

- Merge in all codes for iteration v1.3
- Ensure all features are working as intended for v1.3

**Discussion Notes:**

1. Merging in of codes for iteration v1.3
   Every member was involved in code review for other members of the team to ensure code quality of the individual features.

2. Testing of features for v1.3
   Every member wrote test cases to support the features they have implemented to ensure that all test cases are passed before merging the code into the master branch.

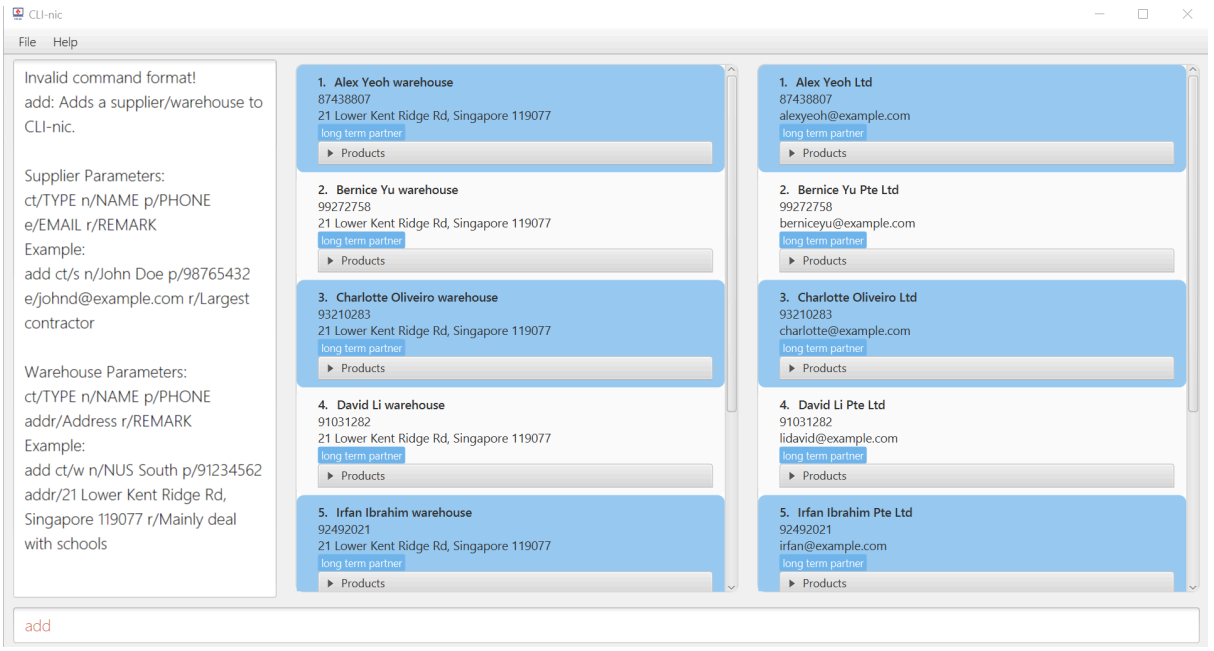
**12th Meeting Week 11 (26/10/20)**

- v1.3 Features Demo
- Merge in all UG/DG for iteration v1.3
- Do a smoke test among members and release jar file for v1.3
- Close milestone v1.3
- Other miscellaneous issues

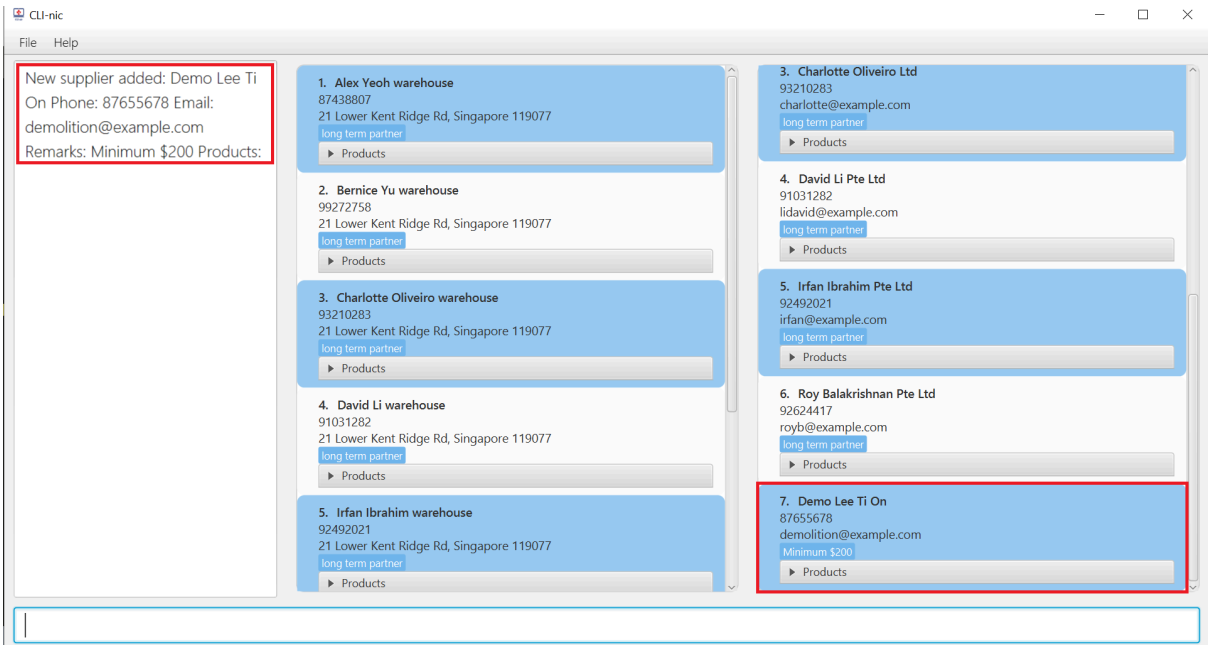
**Discussion Notes:**

1. v1.3 Features Demo

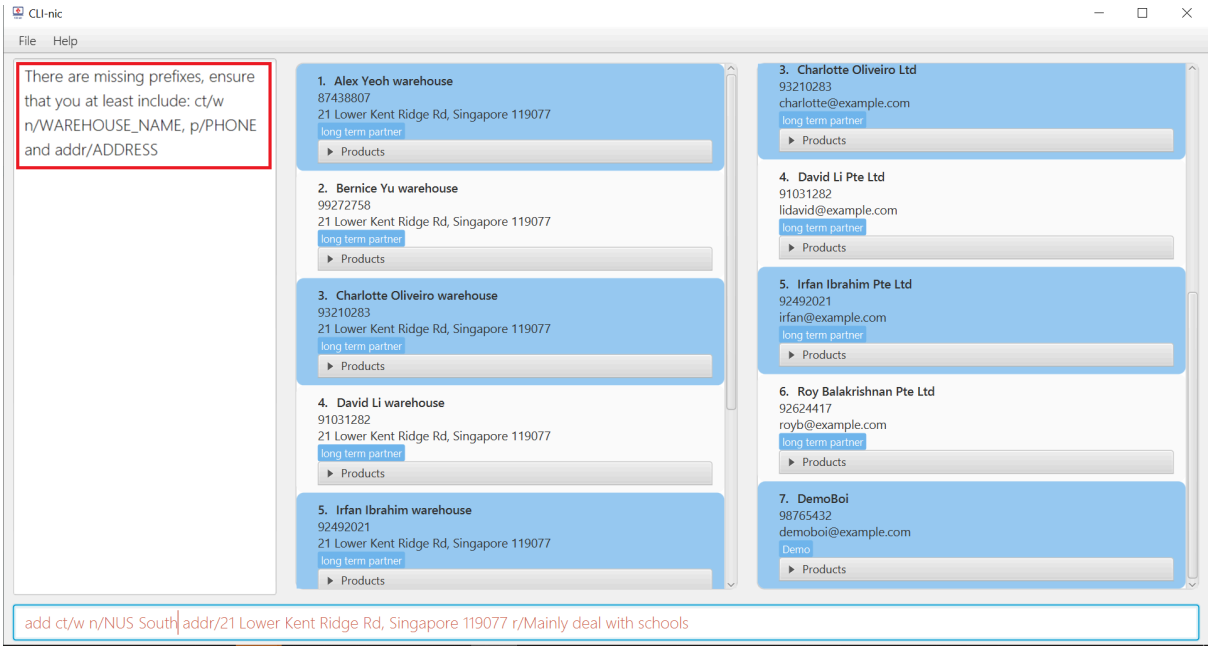
   **add**




   Guide/error message for AddCommand

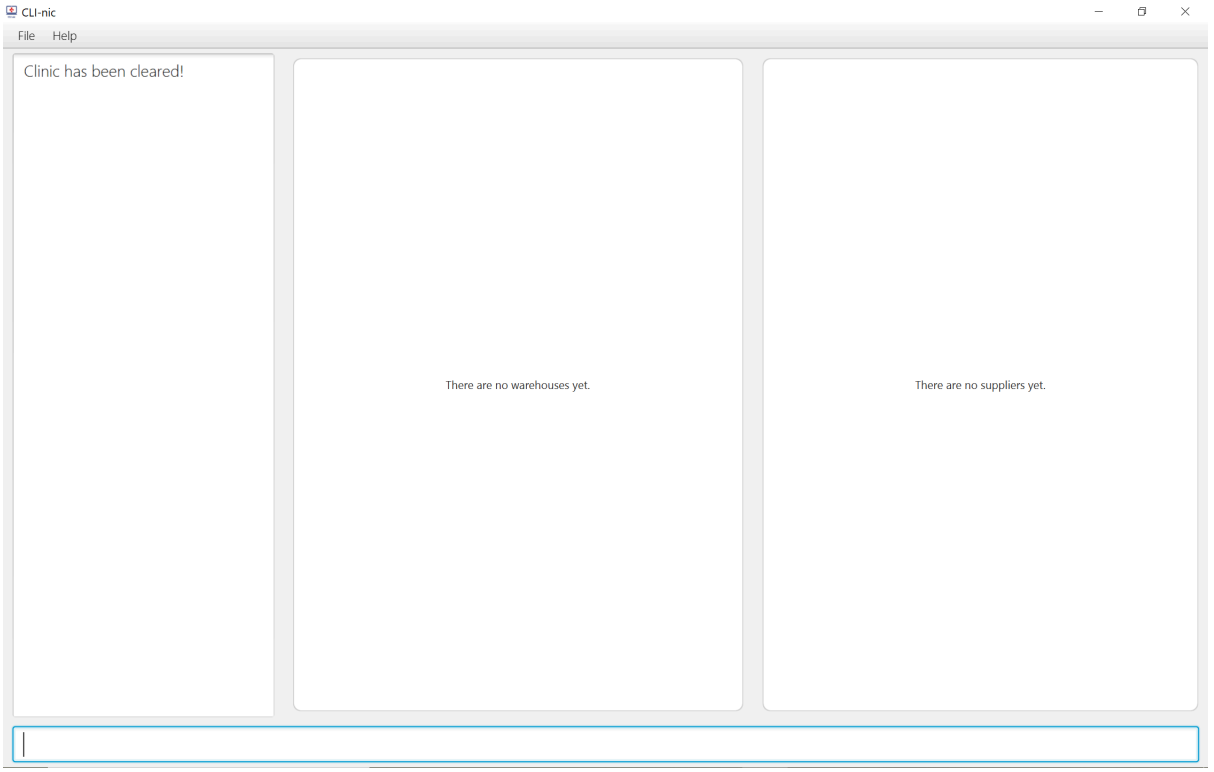



Successful addition of supplier



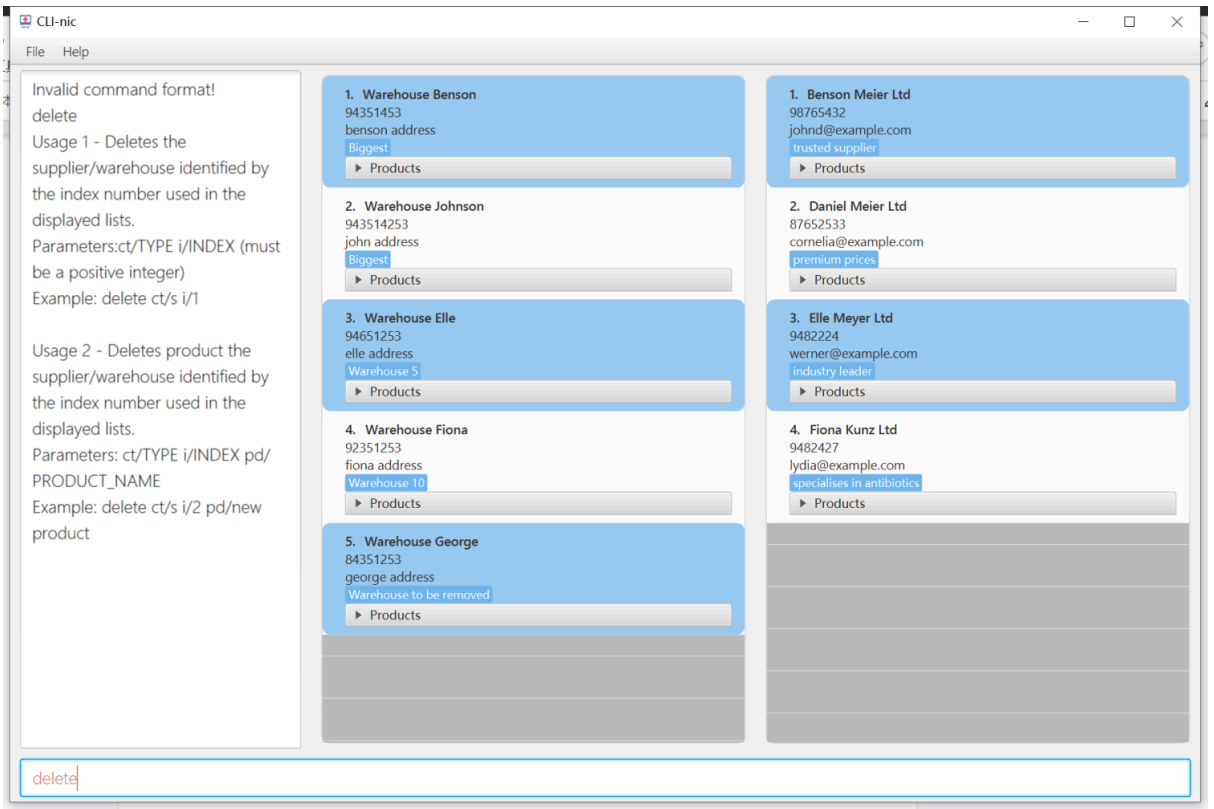

Failure to add warehouse when compulsory prefixes are missing

## clear

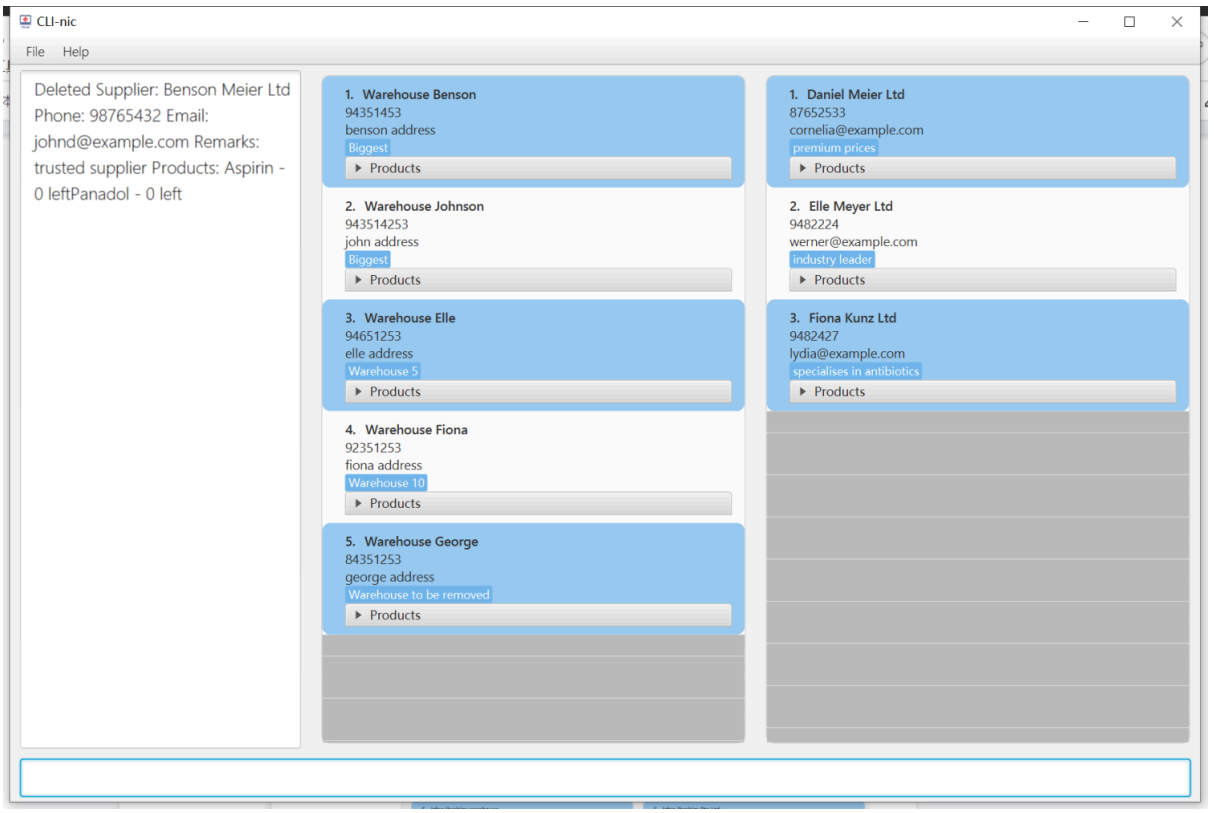



All warehouses and suppliers has been cleared

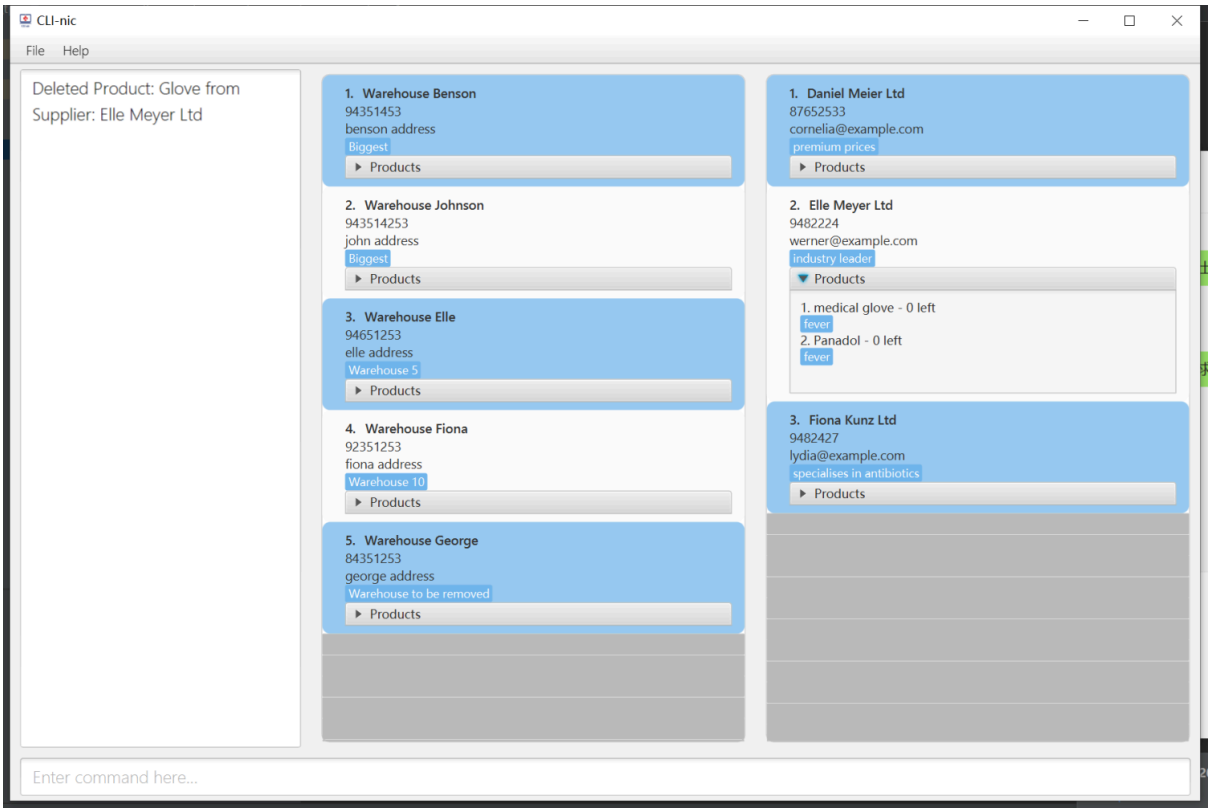
## delete




delete errors and usage

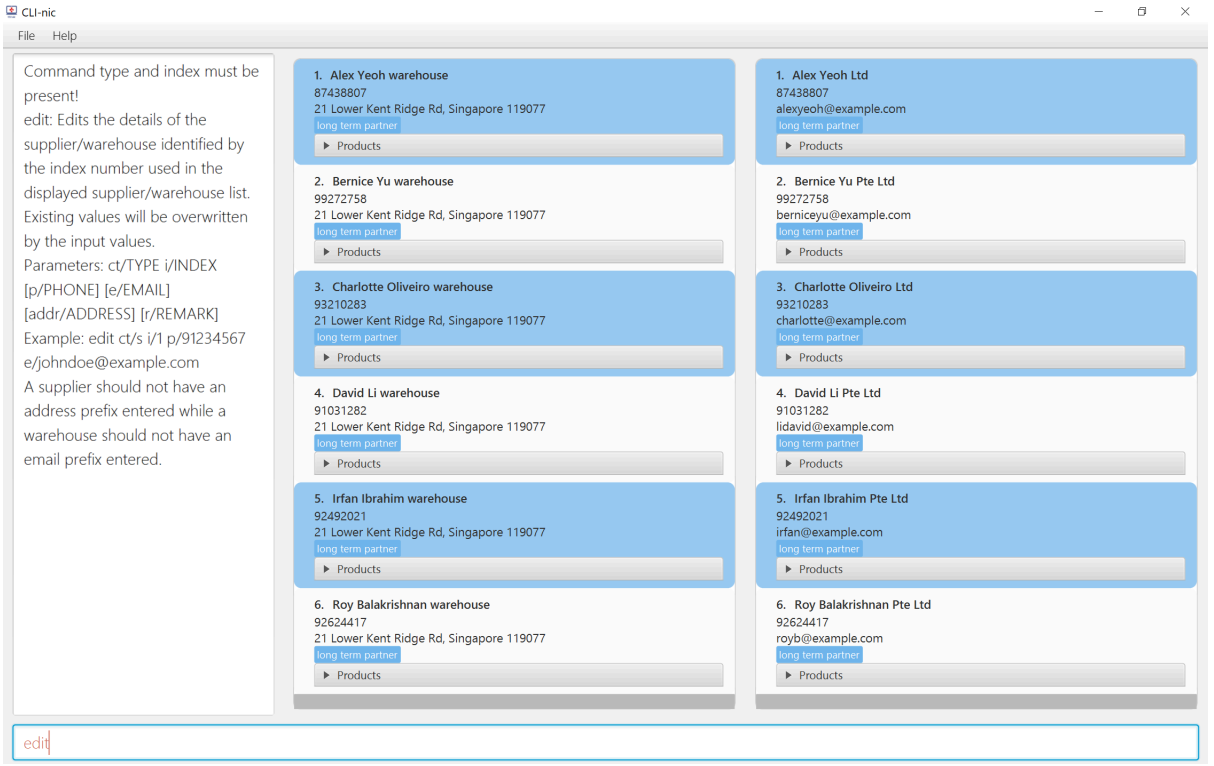



delete supplier success

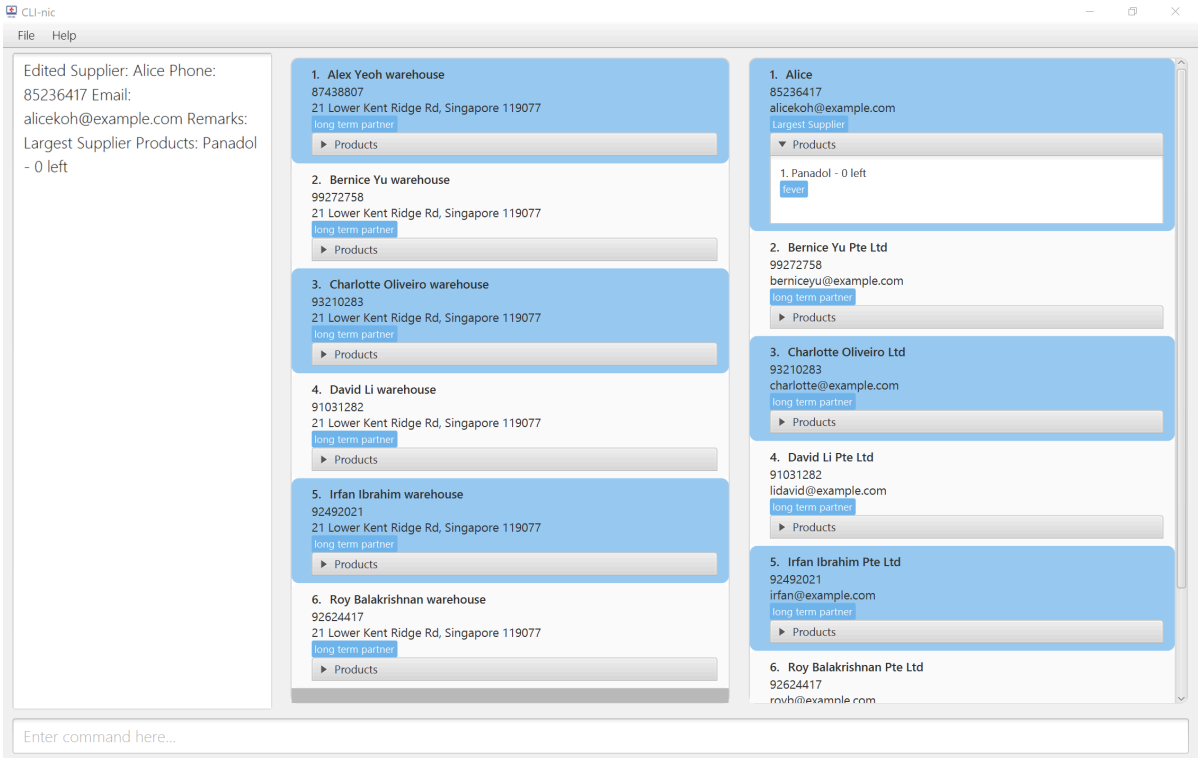



delete product success

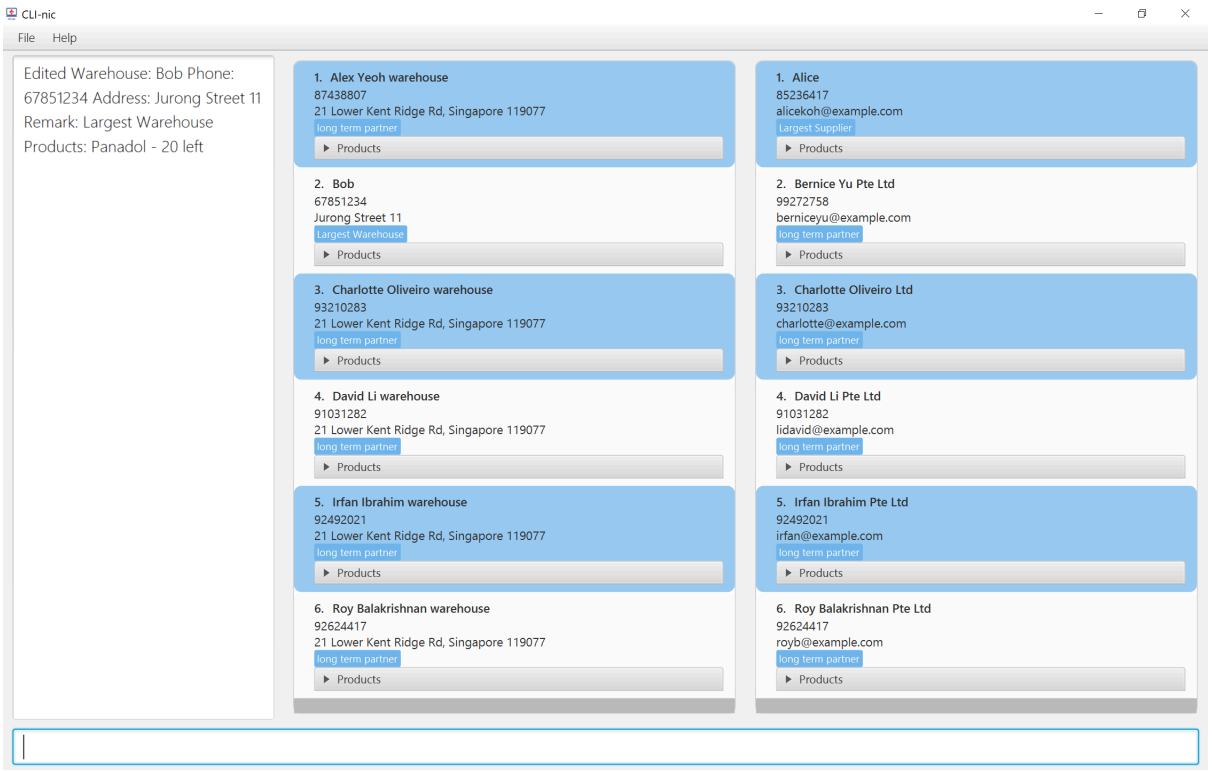
# edit




Edit errors and usage

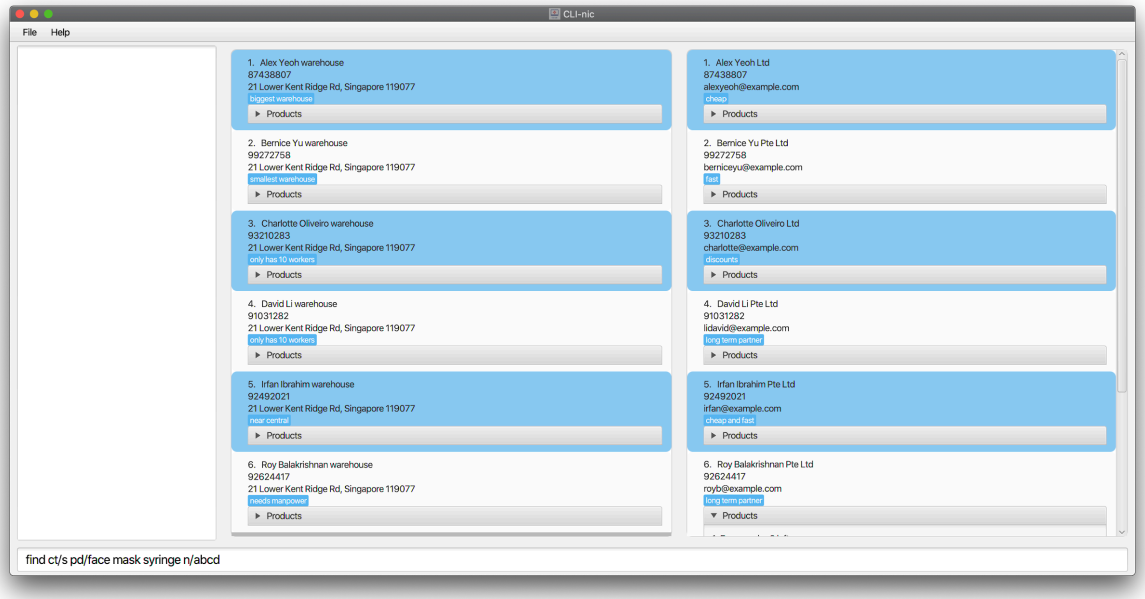


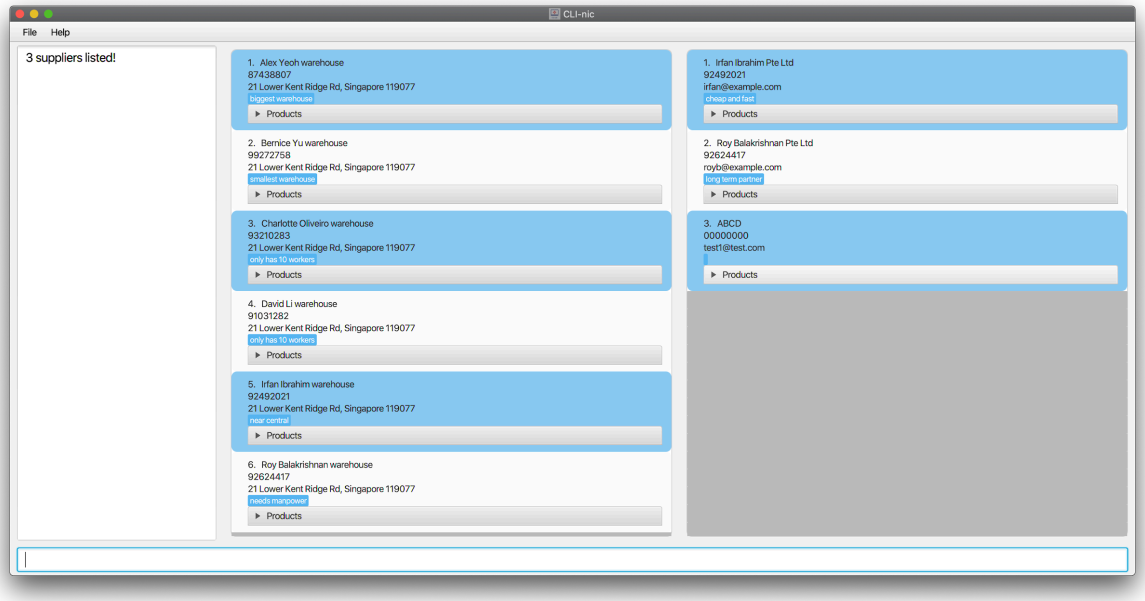

Edit supplier success



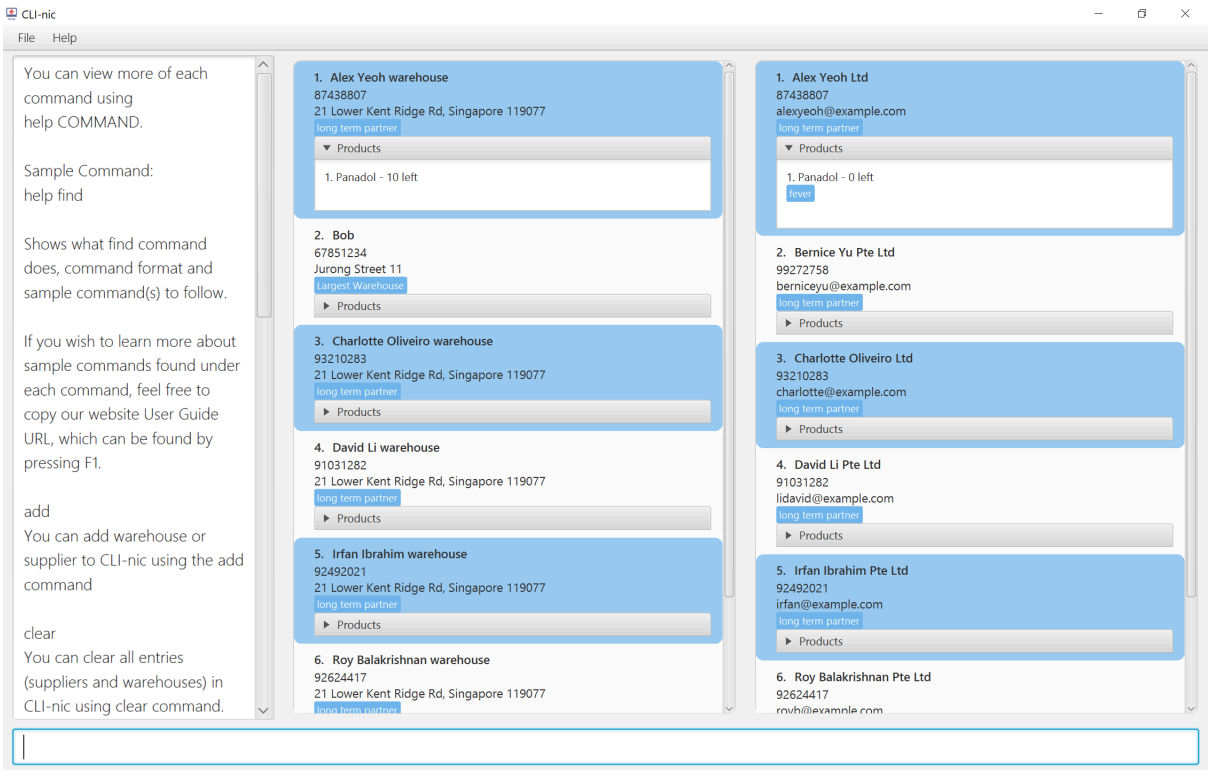

edit warehouse success

# find

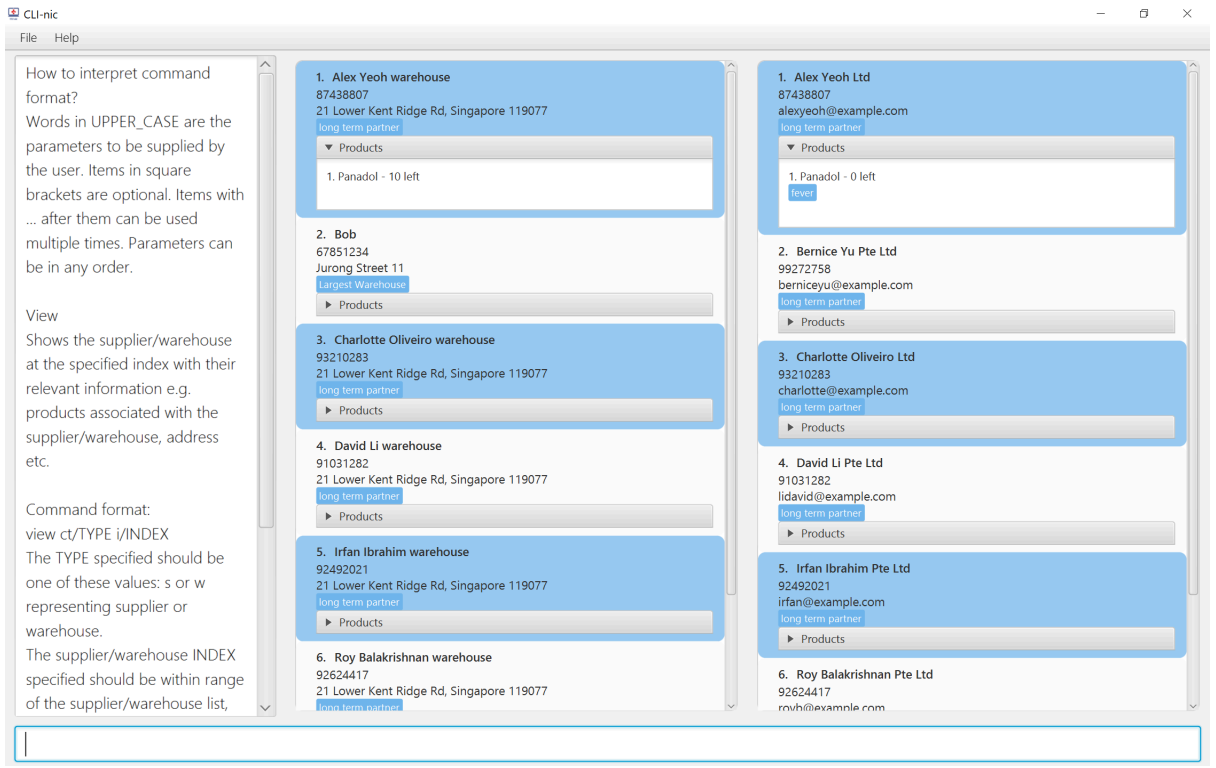





## help

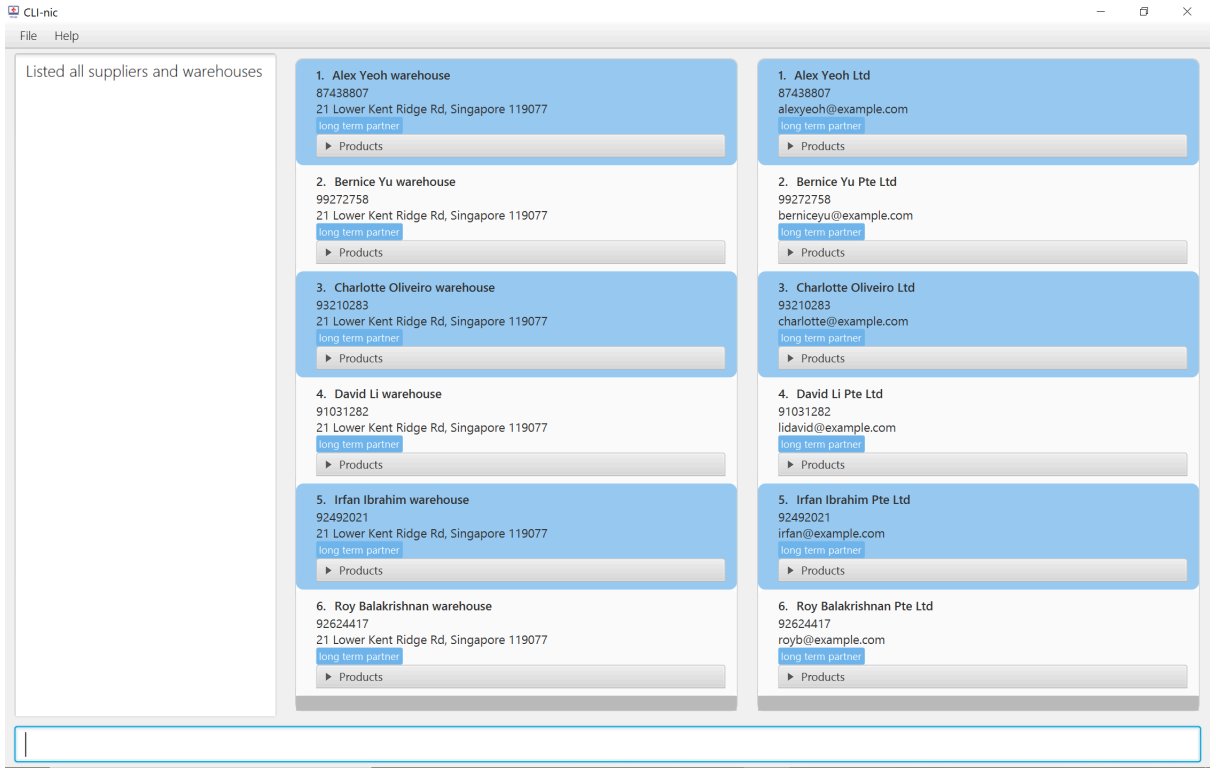



Generic help message



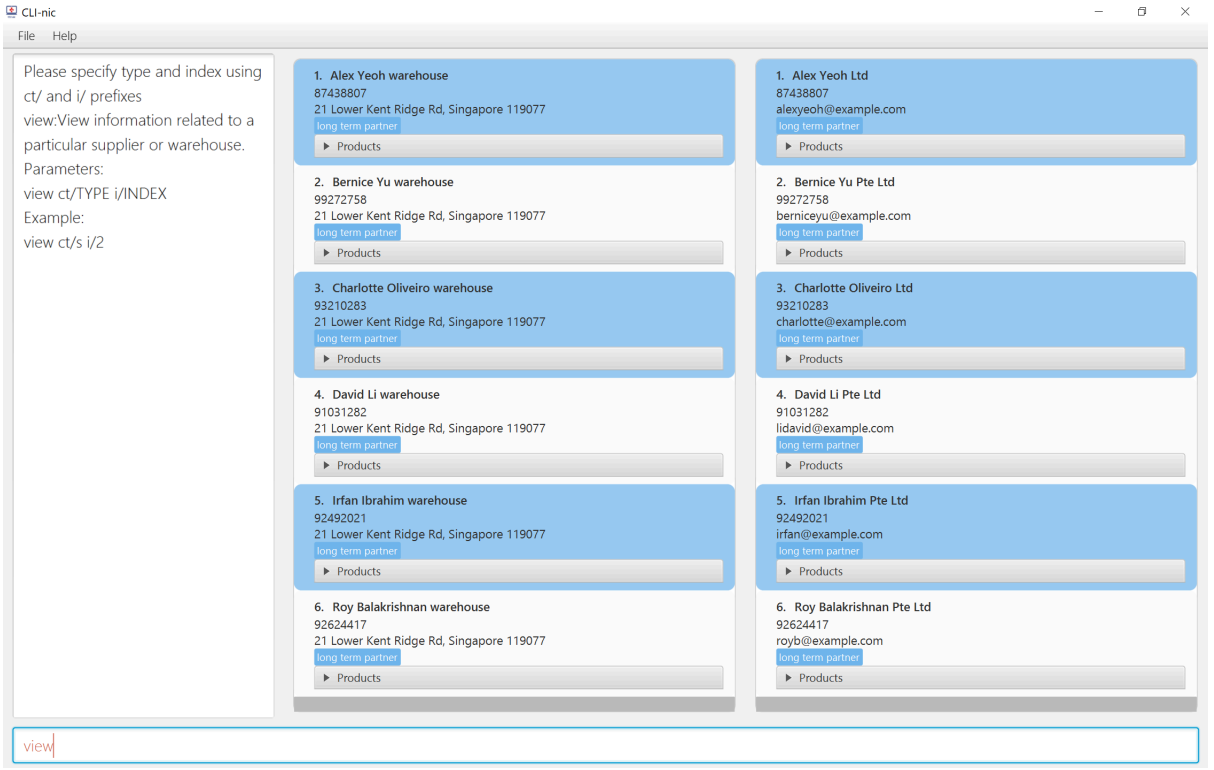

Sample help COMMAND (This case, help view) message shown

## list

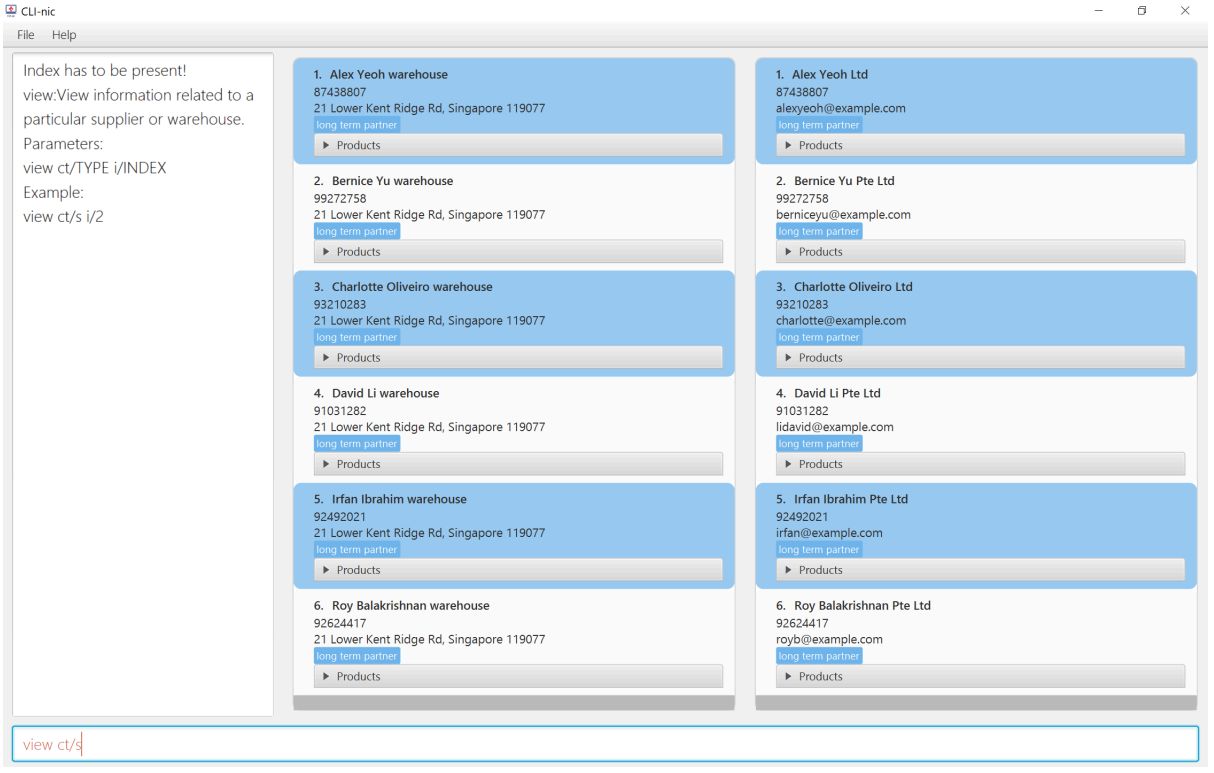



All suppliers and warehouses has been listed

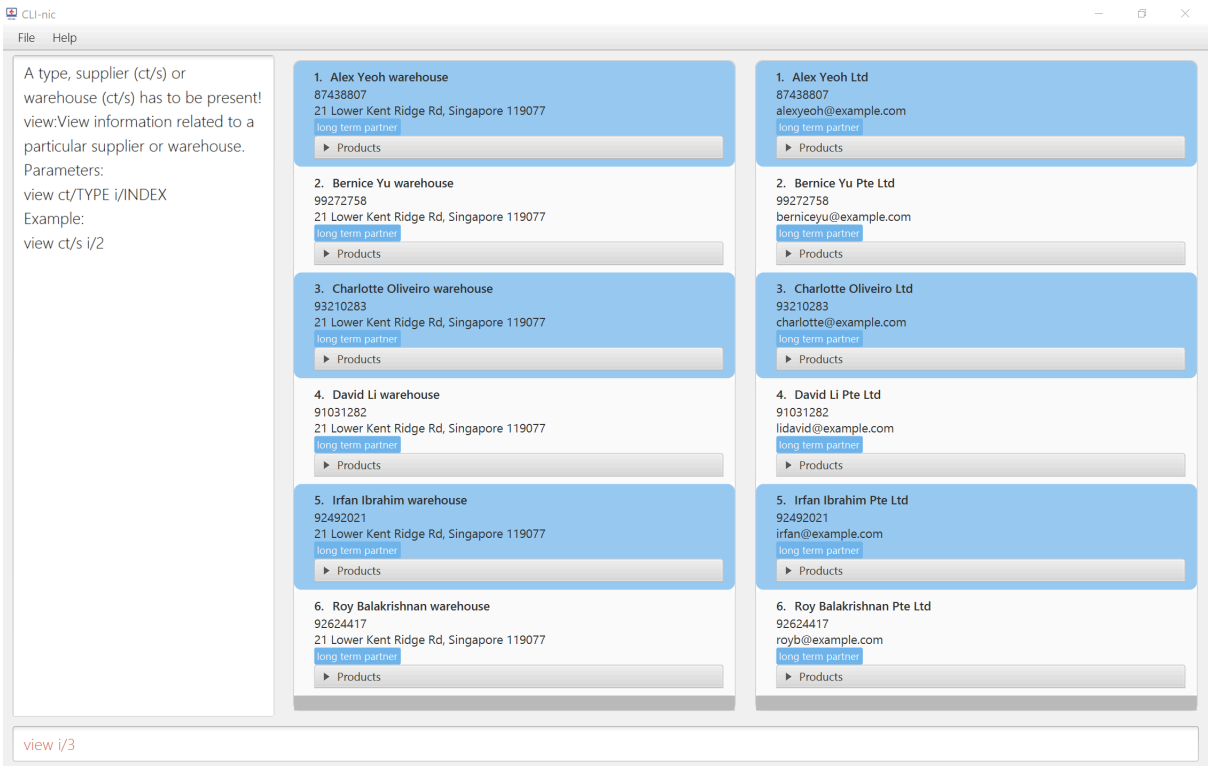
# view



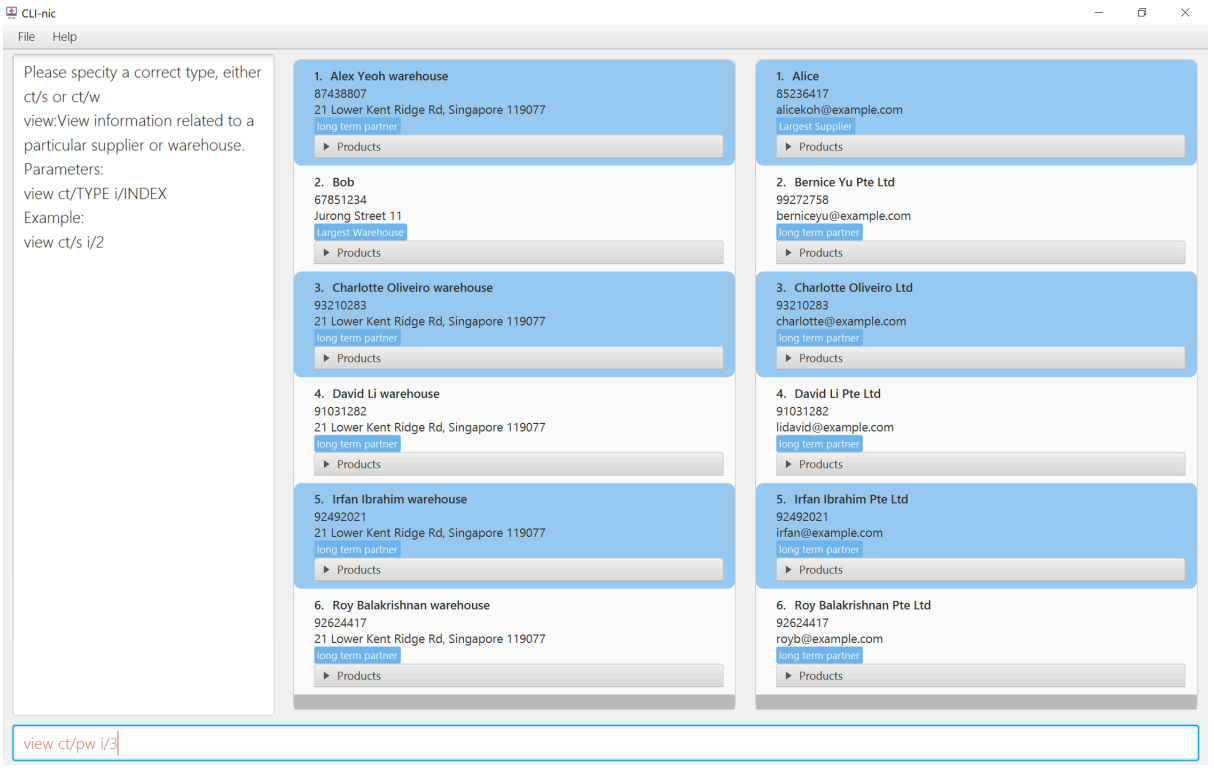

view command error + usage message (without any prefixes)



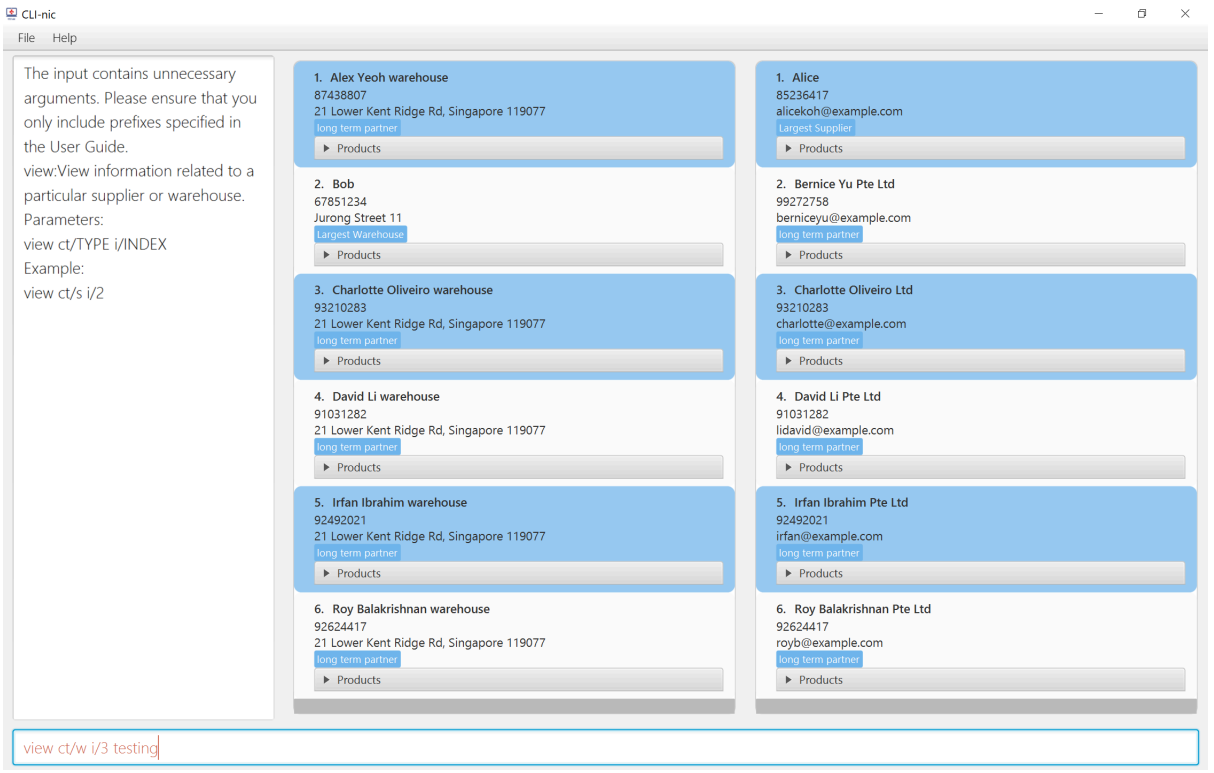

view command missing index



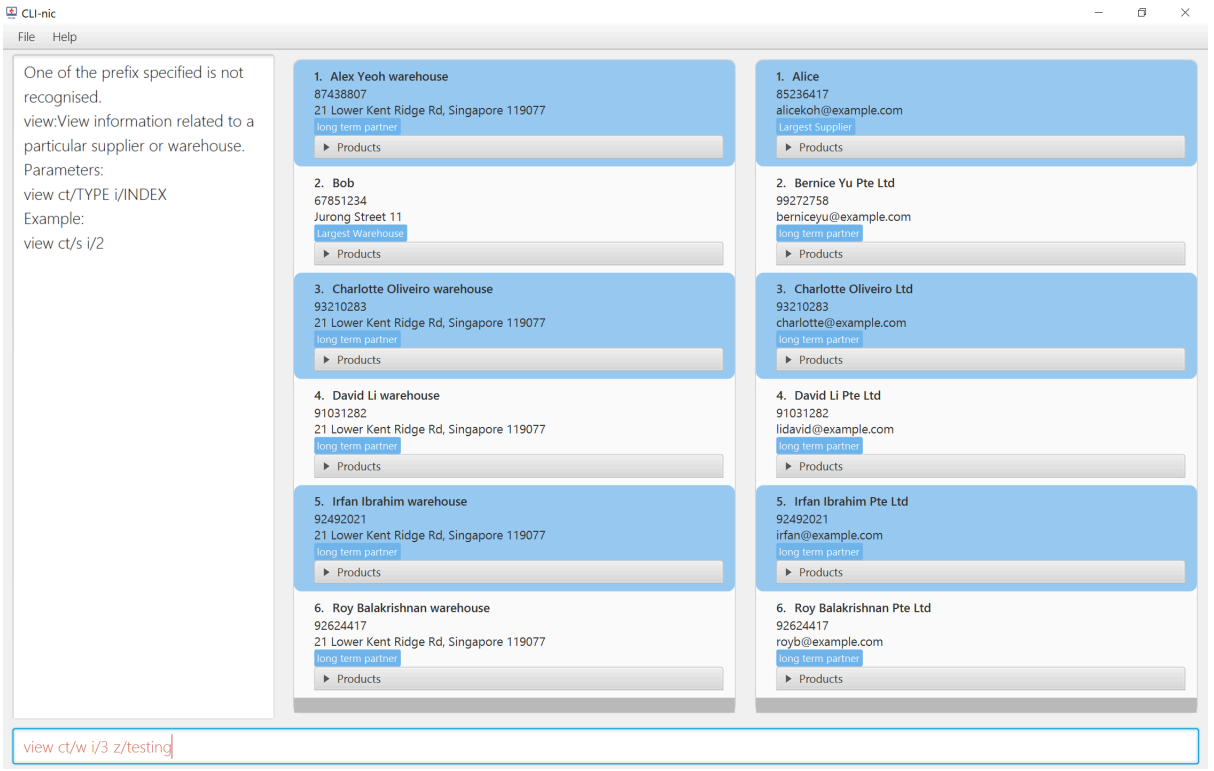

view command missing type



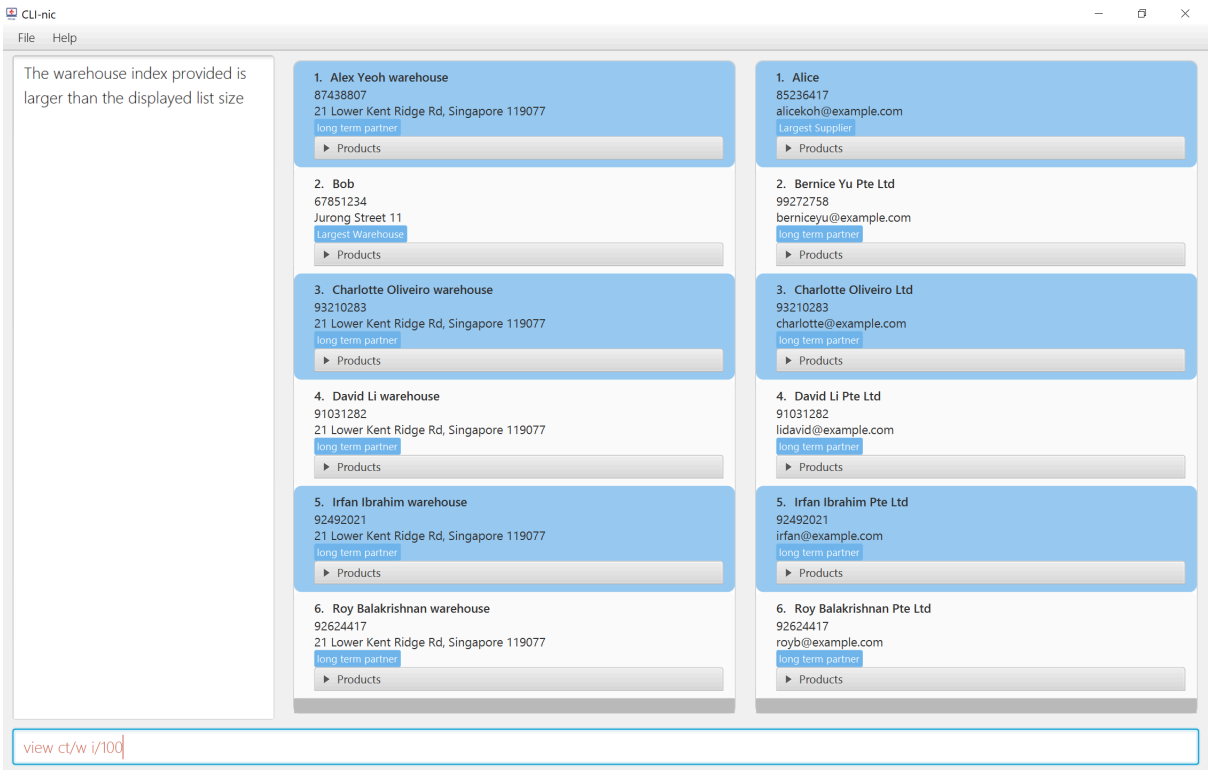

view command wrong type specified



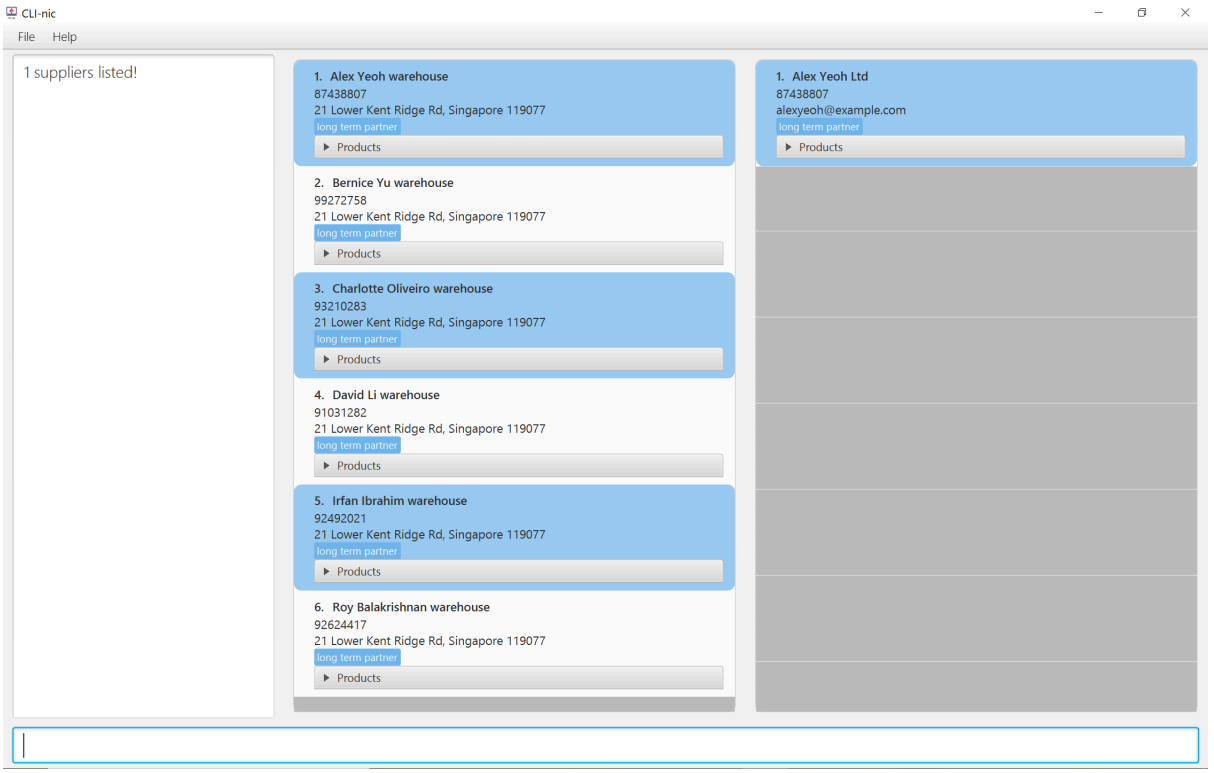

view command with extra arguments at the back



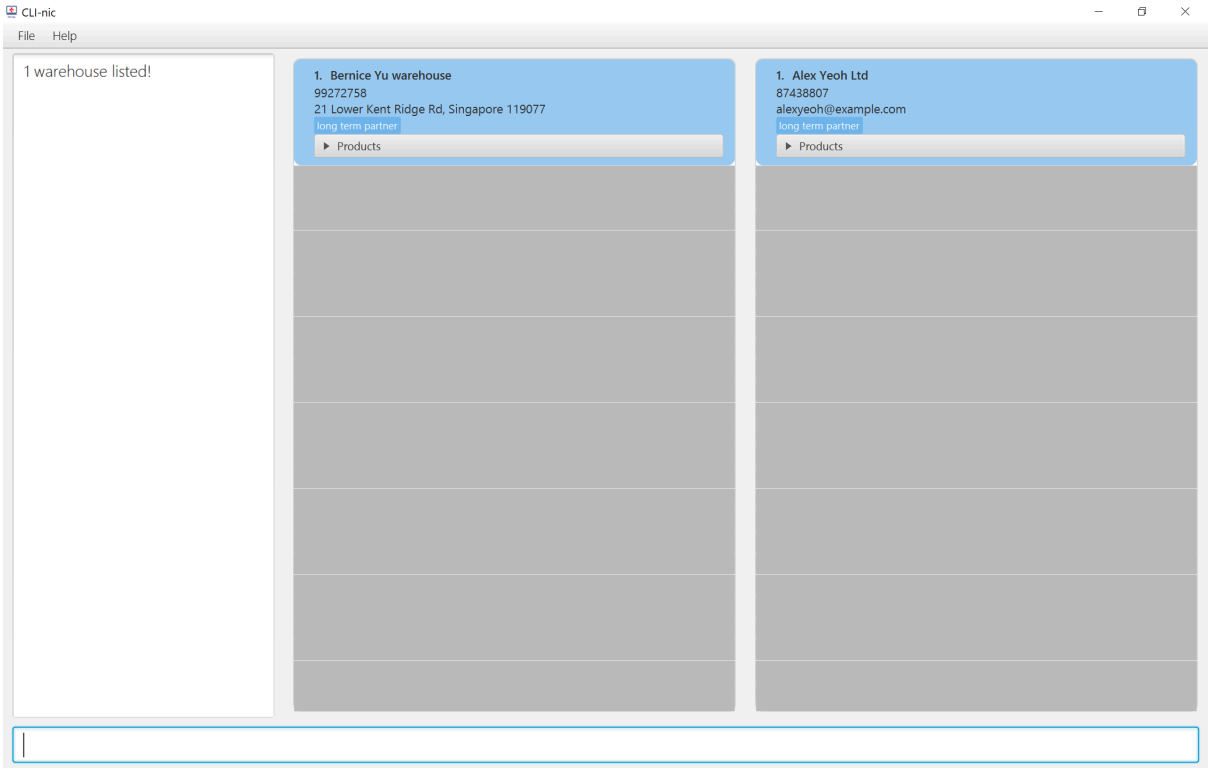

view command with wrong prefixes used




view command index specified is out of range

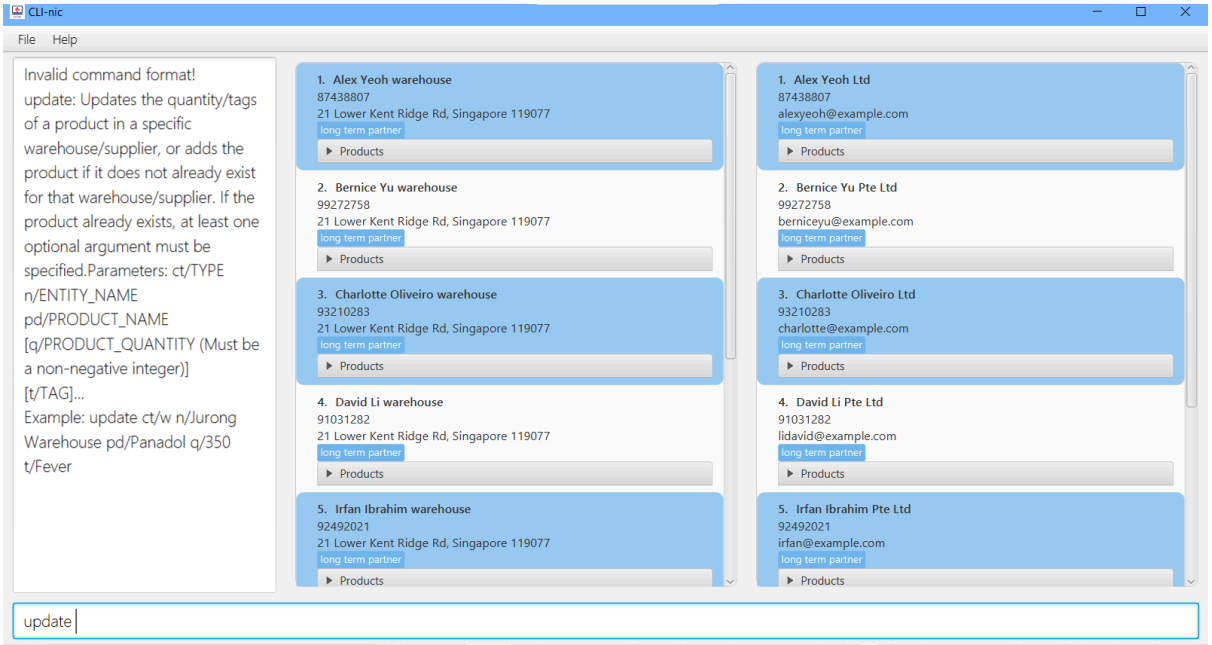



supplier view success message

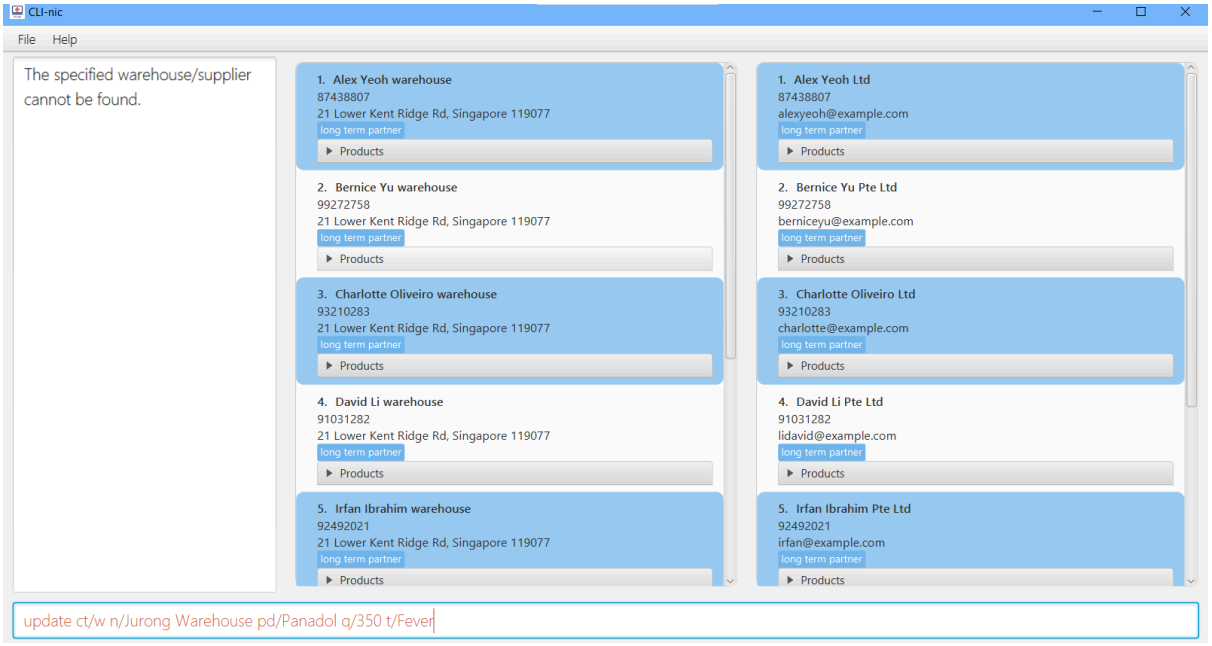



warehouse view success message

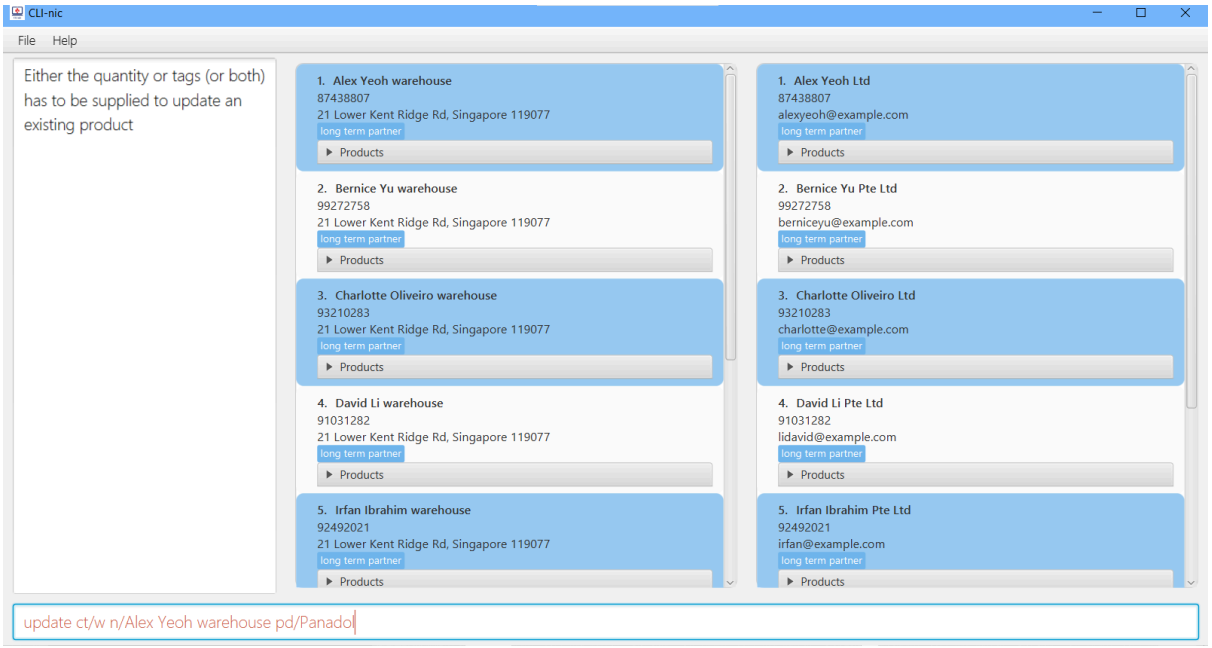
## update



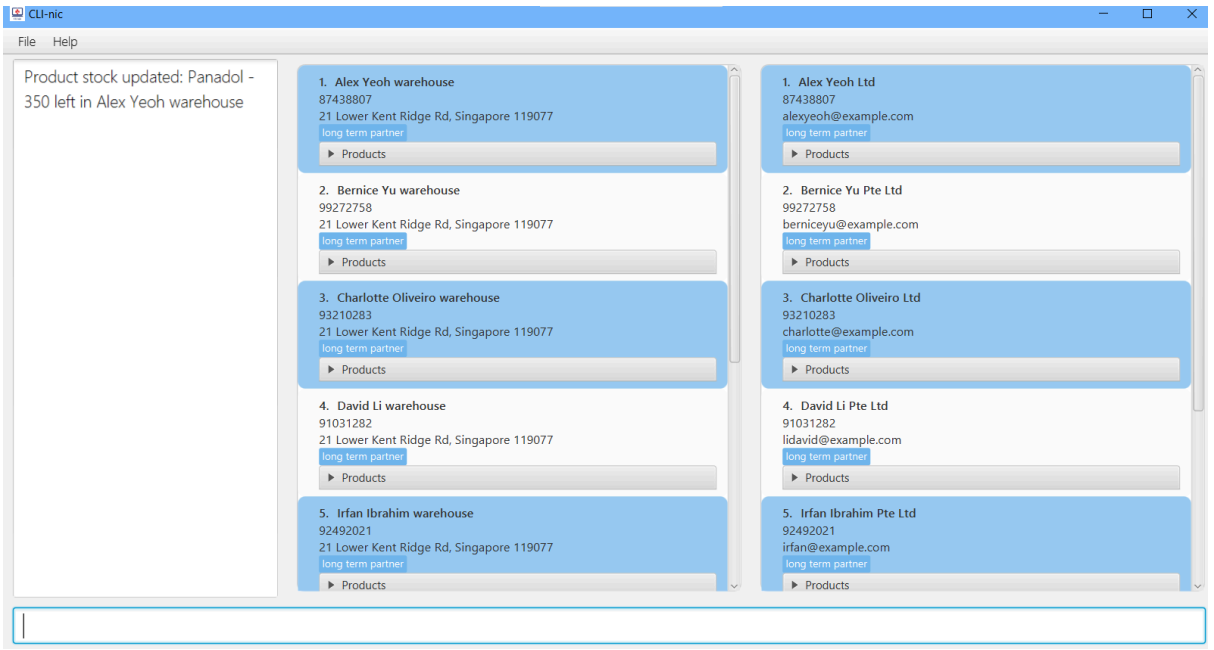

Invalid format



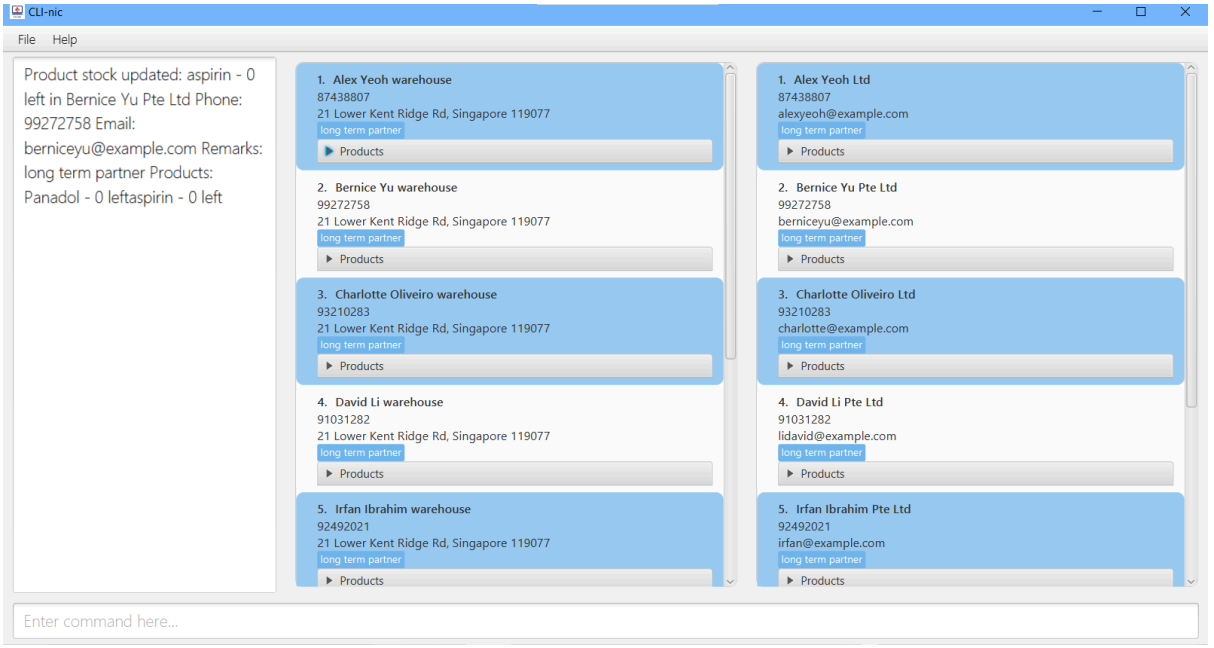

warehouse/supplier does not exist




no optional parameters supplied for a product that already exists for the warehouse/supplier

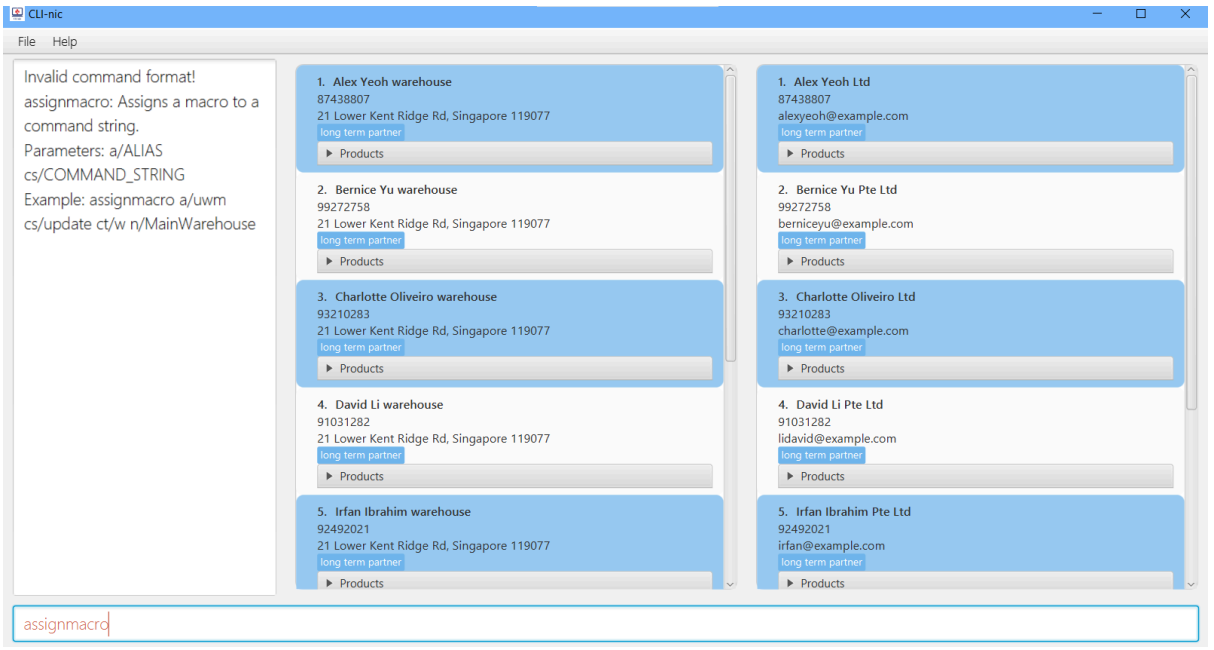



successful update of product

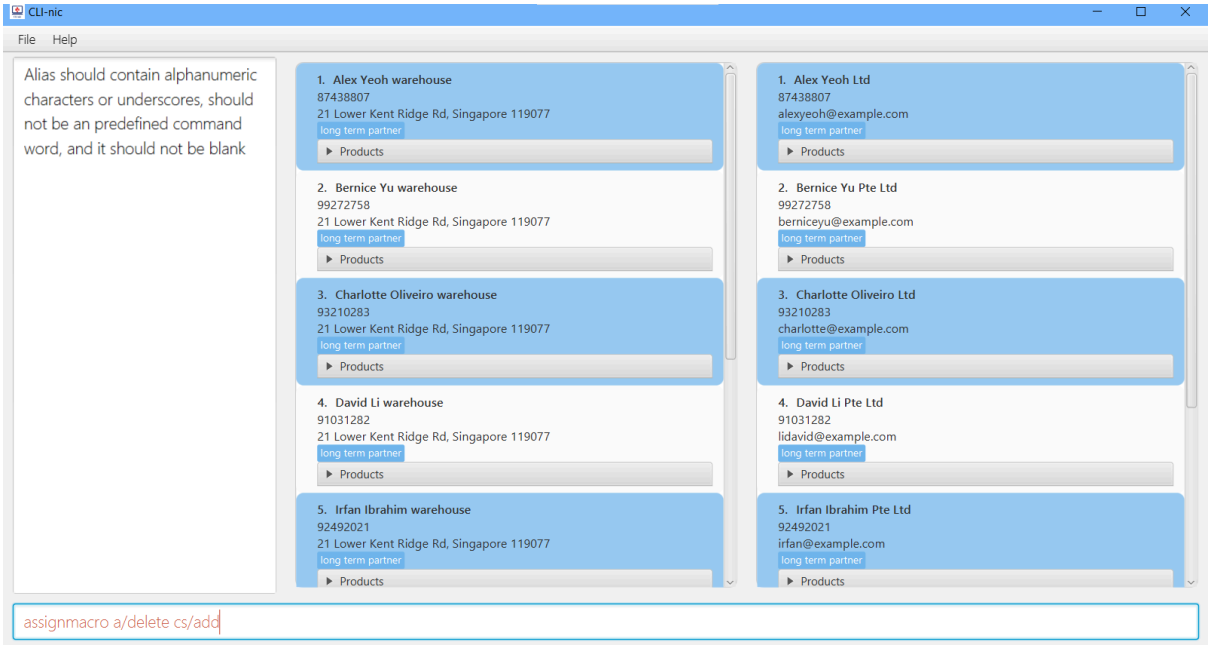



When no optional parameters supplied and that the product to be updated does not previously exist for the warehouse/supplier

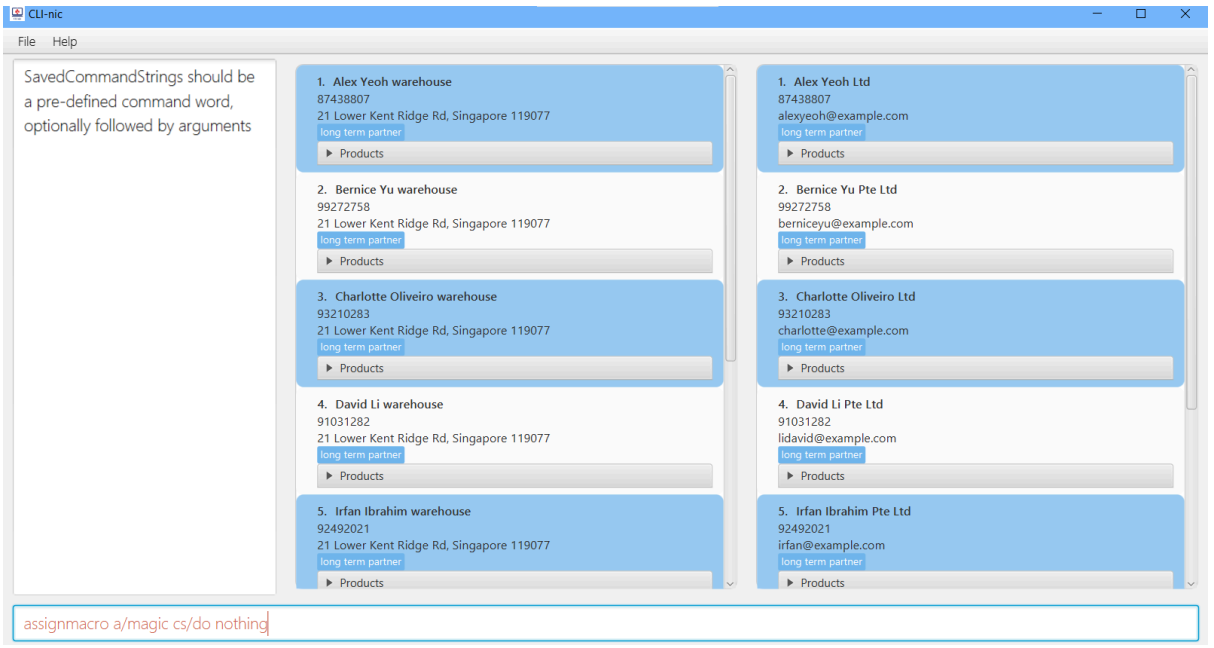
## assignmacro



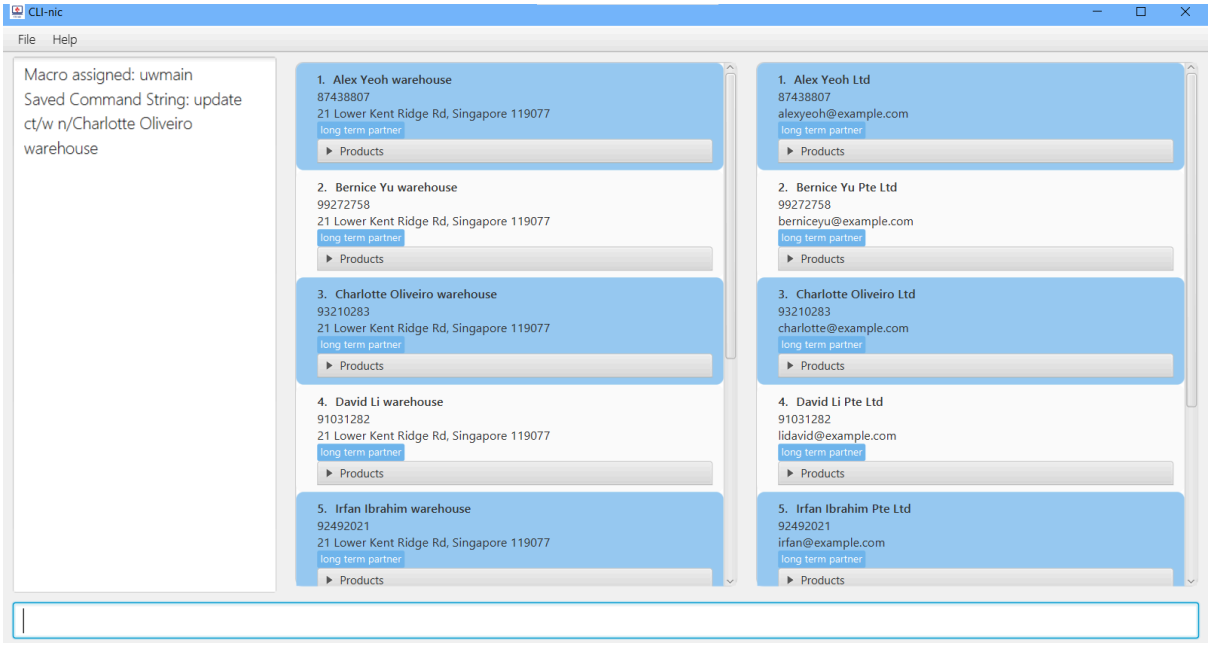

invalid command format




Using a pre-defined command word as the alias

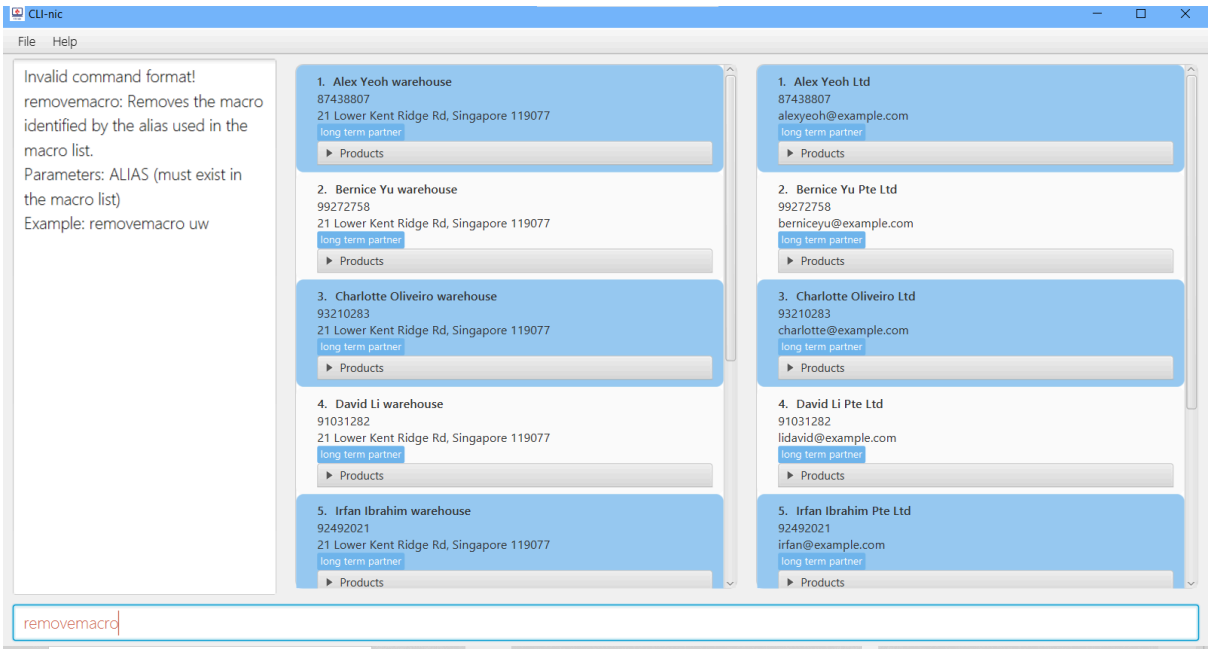



Saving a command string that does not start with a pre-defined command word

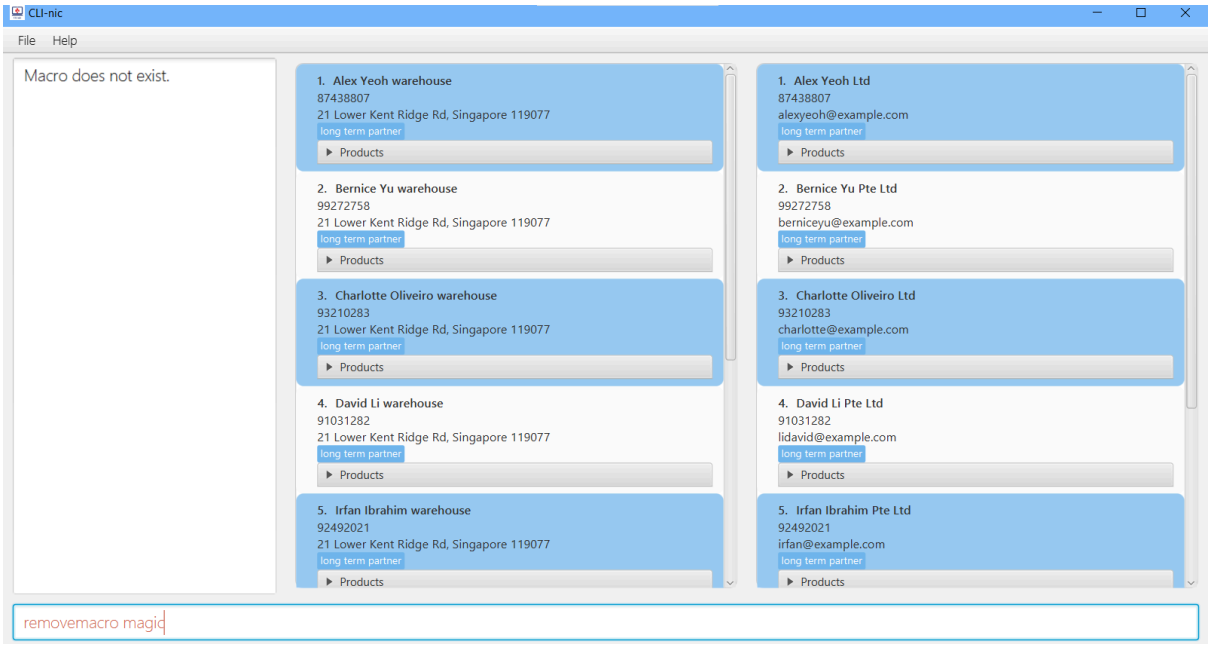



successful assignment of macro

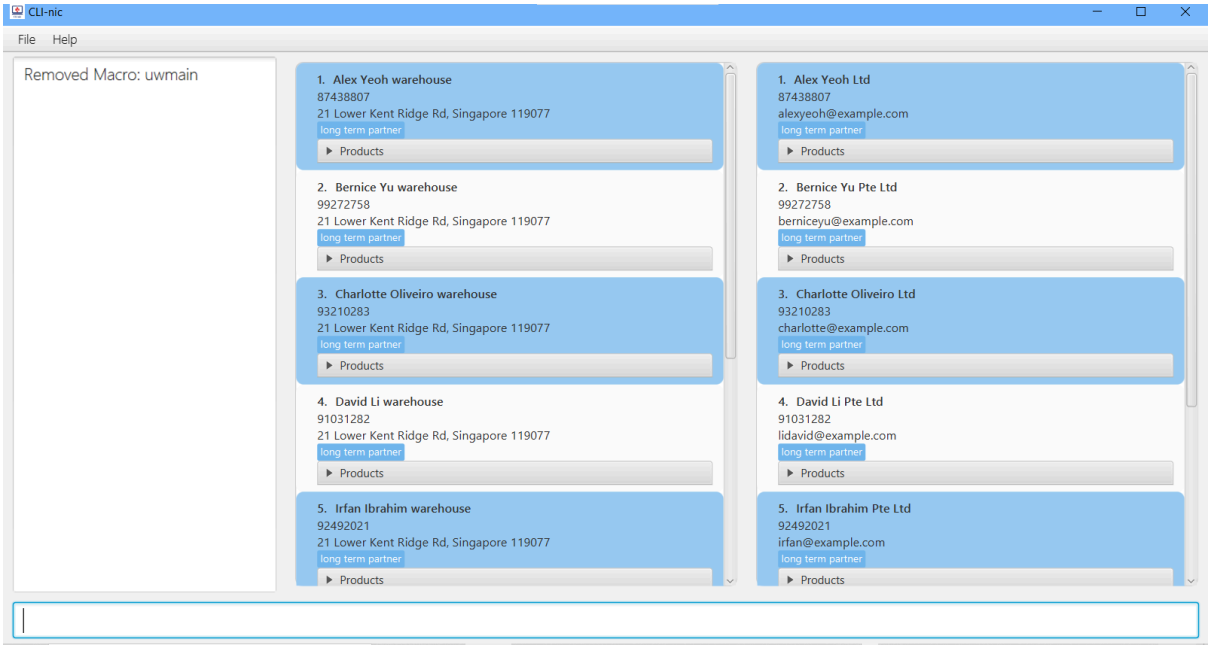
**removemacro**




Invalid format




macro does not exist




success

**exit**

No screenshots available as app terminates upon exit command

2. Merge in all UG/DG for iteration v1.3
   All UG/DG contributions from every member to update the new features has been merged into the master branch and reflected on the team's website.

3. Smoke test among members and release jar file for v1.3 & closing of milestone v1.3
   A jar file was generated for smoke tests (Members using different OSes) to ensure that the jar file works and the functions for the various features are working correctly as well. A jar file for v1.3 has been released on Github after the meeting. Milestone v1.3 and issues associated with it have been closed following the release of the jar file as well.

4. Other miscellaneous issues
   Changes to be made for manual adjustment of UML diagram
   skinparam defaultFontSize 17
   skinparam ArrowFontSize 17
   skinparam ArrowThickness 3
   skinparam ParticipantPadding 1

**13th Meeting Week 12 (1/11/20)**

**Agenda:**

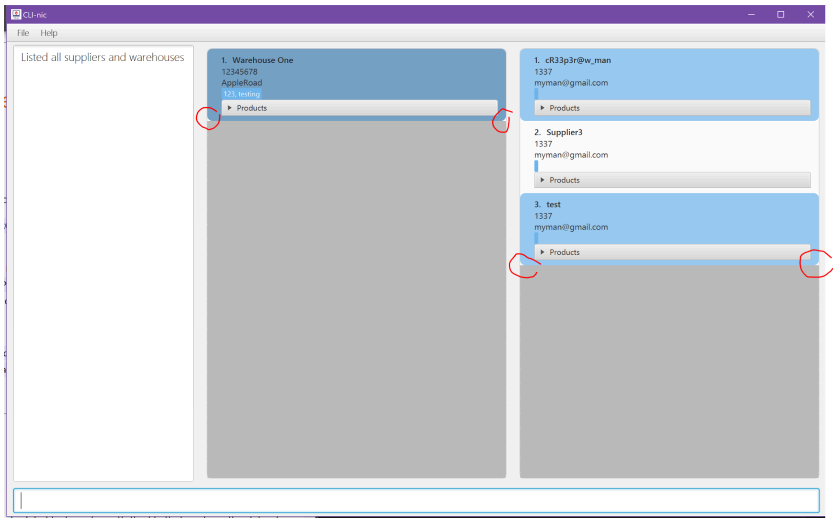
- Discussion of bug reports from PE dry run
- Discussion of any additional features we want to implement, internal deadlines for documentations

**Discussion Notes:**

1. Discussion of bug reports from PE dry run
   Functionality bug
   a. Add command, “wrong error message when typed add ct/ , where s/w/pw/ps was stated as accepted type”. (Tested true)
      i. If a user uses ct/pw or ct/ps then a warehouse will be added (if other fields of warehouse are present) => check conditional (Jeffrey, fixed)
   b. Add command displays wrong error message, where non-existent prefix used, but the error shows type invalid (when invalid prefix used after a ct/s etc)
   e.g. ct/s testing
   => throw generic error like now, or we go into the specifics?
   =>will commit a change to show specific error and see how it looks (Yu Ting)
   => won't work on name, address

   Team Notes: name and address spaces are allowed, slashes not allowed => error message (now)

   c. assignmacro accepts improper capitalisation, but won’t be usable (Tested true)
      i. reported ss: myAlias cs/add, but not able to use myAlias
      =>case insensitive issue
   d. Rounded rectangle for GUI




https://github.com/AY2021S1-CS2103-W14-4/tp/issues/181

=> possible fix: change background colour to grey (column between sections(?)) (Jeffrey)

e. Names cannot take in a single character => cannot add, cannot find by 1 character as well (Might want to check RegEx) => (Name fixed)
f. Edit command accepts name with slashes (Tested true)
    i. Just did a test for add, find command, this happens as well, might want to check RegEx for name
g. Error message does not match the error (e.g. add command, with extra param, instead of showing extra param, says type is invalid) => I don't quite agree with this one, since user did not even have a proper full input in the first place
   => Make sure to complete format first before ensuring inputs are valid or not?
   => standardise invalid command format thrown (will change edit to throw a generic error) (Yu Ting)

h. Assignmacro to an assignmacro does not work (This should not happen though? else it will just be a cyclic?) => I guess the tester was implying that the error message is problematic
   https://github.com/AY2021S1-CS2103-W14-4/tp/issues/194
   =>Update UG to state that cannot assignmacro to an assignmacro command [fixed in UG]
   => check if assignmacro is present in cs => yes throw an error

i. Address should not be present in add command for supplier, however if address prefix was used with an input given, supplier was still created (Tested true), also the same for warehouse (Tested as well)
   https://github.com/AY2021S1-CS2103-W14-4/tp/issues/196
   => throw an error if address/email (Check right at the start => after ensuring type is supplier/warehouse) (Jeffrey)

j. Not able to validate the address, tester specifies that blahblah should not be allowed. => I disagree with this, no way to check what determines a valid address, since some addresses may not contain Street/Avenue etc in them, could be just a building name. Perhaps we can be clearer in our UG saying what valid address is.
   =>edit UG to make it clearer [removed this point from UG]

Features bug

k. A feature to list added macros (will implement)
   listmacros
l. Inconsistency regarding use of i/ for edit/view/delete but n/ for update

i. Another issue also mentioned by using name, not suitable for CLI since the name can be rather long => I think this is debatable since a fast typist should be able to type a long name fast as well.

Switch to index for update [fixed in UG]

m. Redundancy of view command, since user will scroll to see the index which they can just click to expand the products
   i. Show the s or w with product pane expanded
n. Product dropdown list is not CLI friendly (have to click it), tester suggested for view or finding a product, it can drop down automatically => kind of agree with this, if we cannot find a way for it to drop down automatically, at least for view (with another tester mentioning the redundancy of view) maybe we can just expand by default?
   =>try to look into changing UI (?) would be interesting if pressing a key like ("F2 KEY") to expand all the products
~~o. Purpose of assignmacro, not sure how to utilise in other commands, and no way to list all the macros assigned. => perhaps tester is saying no point (implies) in assignmacro other than updating of stocks, since you cannot add more than 1 warehouse with same name etc~~
   ~~https://github.com/AY2021S1-CS2103-W14-4/tp/issues/202~~
p. Fast typist can have typo, serious issue since delete if it is misspelt as s instead of ps, it will delete an entire supplier even if pd/ is present
   i. I think we can check if pd is present, if it is and type is s/w we will not let it through. However, fast typist making typo is contentious since that is an assumption.

(fixed)

Documentation bug

q. No clear guide on how to add a product (user only know when they read update portion in UG) => direct them to update section under add command so users know how to add products [fixed]
r. Clarity of UG for index, is it the index of original list, or the list currently displayed to user => should specify this in UG [fixed]
s. Clarity of UG for assign macro: What does it mean by pairing macro and commands? What does pair mean in this situation? (Improve phrasing) [fixed]
t. No Footnote/caption to explain which sample command does the screenshot refer to [not needed now since we only have 1 example now]
u. Typo for help and edit command
   (https://github.com/AY2021S1-CS2103-W14-4/tp/issues/190) [fixed]
   (https://github.com/AY2021S1-CS2103-W14-4/tp/issues/189) [fixed]
v. Typo in command features
   (https://github.com/AY2021S1-CS2103-W14-4/tp/issues/199) [fixed]

2. Discussion of additional features, internal deadlines for documentations
   list assignmacros
   autocomplete => testing: either works or not works (Jeffrey try)
   Stats => how many warehouses/suppliers in total
   summary of products each supplier or warehouse have => just number
   sorting (specify by what?) specify name of product, sort them by qty
   filtering
   up command history
   Any implementation thurs/fri testing by saturday
   Clear up bugs (features and UG ) => 4th Nov
   Deadlines for documentations (DG, PPP (personal contribution))
   => (except new features (if applicable)) Friday night**
   => new features (Saturday)**

   Standardisation of DG:
   all classes and command words: just use ``
   Write up for implementation:
   1. Parsing
   2. Execution
   3. Result Display

   If want diagrams => detailed enough, then less write up
   else, more write up to explain

   Sizing of diagram

   "optimised for Command Line Interface (CLI) users while still having the benefits of a Graphical User Interface (GUI)"