

# PlayBook User Guide

---

PlayBook (PB) is a **desktop app for semi-professional youth football coaches to manage their players' contacts, optimized for use via a Command Line Interface (CLI)** while still having the benefits of a Graphical User Interface (GUI). If you can type fast, PB can get your contact management tasks done faster than traditional GUI apps.

- [PlayBook User Guide](#)
  - [Quick start](#)
  - [Features](#)
    - Viewing help: [help](#)
    - Adding a team: [addteam](#)
    - Adding a player: [add](#)
    - Assign player to team: [assignteam](#)
    - Assign injury to player: [assigninjury](#)
    - Unassign injury from player: [unassigninjury](#)
    - Creating a new position: [newposition](#)
    - Assigning a position to player: [assignposition](#)
    - Assigning player as captain: [makecaptain](#)
    - Unassigning player as captain: [stripcaptain](#)
    - Listing all players: [list](#)
    - Listing all teams: [listteams](#)
    - Listing all positions: [listposition](#)
    - Listing all injured: [listinjured](#)
    - Listing all captains: [listcaptain](#)
    - Editing a player: [edit](#)
    - Locating players by name: [find](#)
    - Filtering players by team, injury and/or position: [filter](#)
    - Deleting a player: [delete](#)
    - Deleting a team: [deleteteam](#)
    - Deleting a position: [deleteposition](#)
    - Clearing all entries : [clear](#)
    - Exiting the program : [exit](#)
    - Saving the data
    - Editing the data file
    - Archiving data files [[coming in v2.0](#)]
  - [FAQ](#)
  - [Known issues](#)
  - [Command summary](#)

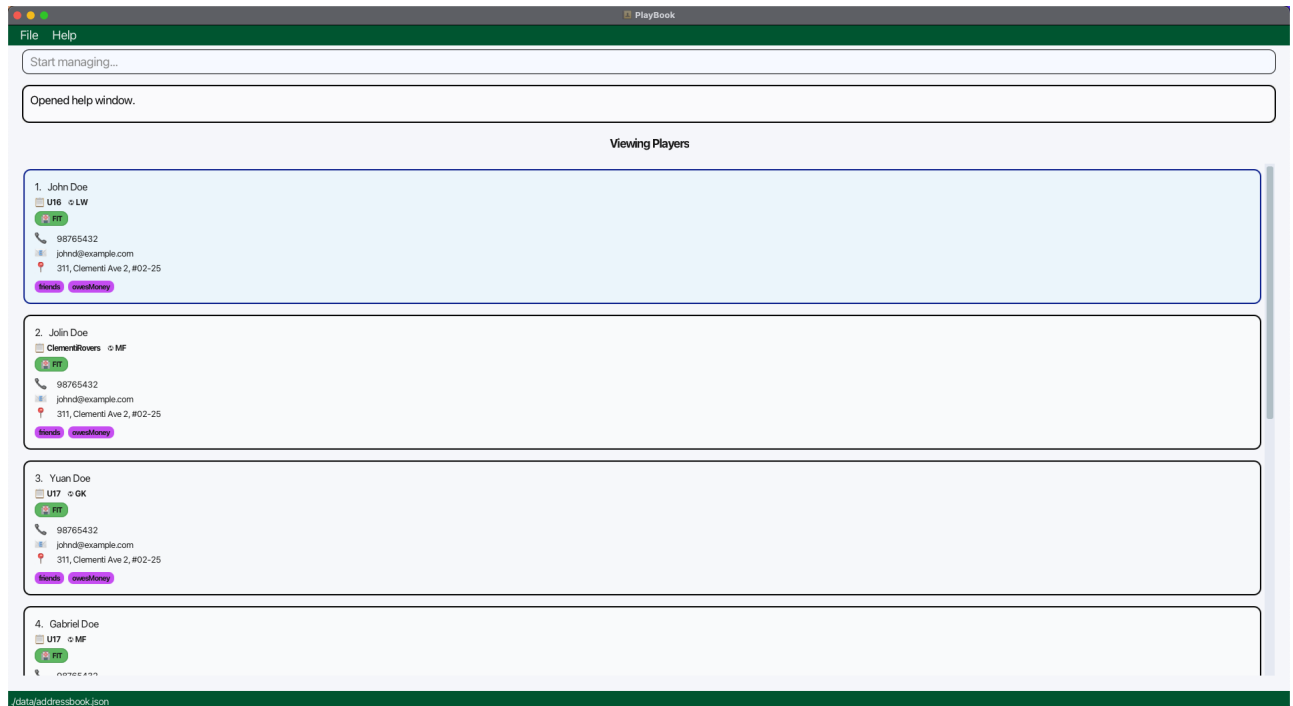
---

## Quick start

1. Ensure you have Java [17](#) or above installed in your Computer.  
**Mac users:** Ensure you have the precise JDK version prescribed [here](#).

- Download the latest `.jar` file from [here](#).
- Copy the file to the folder you want to use as the *home folder* for your PlayBook.
- Open a command terminal, `cd` into the folder you put the jar file in, and use the `java -jar playbook.jar` command to run the application.

A GUI similar to the below should appear in a few seconds. Note how the app contains some sample data.



- Type the command in the command box and press Enter to execute it. e.g. typing `help` and pressing Enter will open the help window.

Some example commands you can try:

- `list` : Lists all players.
- `addteam tm/u16` : Adds a team named `u16` to the PlayBook.
- `add pl/John Doe p/98765432 e/johnd@example.com a/John street, block 123, #01-01 tm/u16` : Adds a player named `John Doe` to the PlayBook (make sure to add the team `u16` first).
- `newposition ps/LW` : Creates a new position named `LW` (Left Wing) in the PlayBook.
- `assignposition pl/John Doe ps/LW` : Assigns the position `LW` to `John Doe` in the PlayBook.
- `assigninjury pl/John Doe i/ACL` : Assigns an injury status of `ACL` to `John Doe` from the PlayBook.
- `delete pl/John Doe` : Deletes `John Doe` from the PlayBook.
- `clear` : Deletes all players' contacts and teams from the PlayBook.
- `exit` : Exits the app.

- Refer to the [Features](#) below for details of each command.

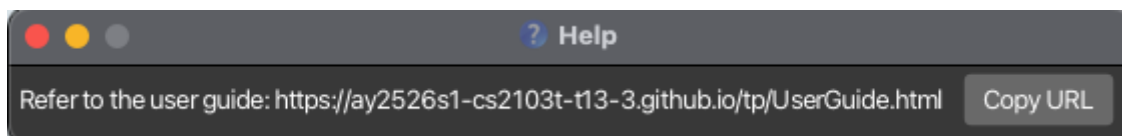
## Features

### Notes about the command format:

- Words in **UPPER\_CASE** are the parameters to be supplied by the user.  
e.g. in `add p1/PLAYER_NAME`, `PLAYER_NAME` is a parameter which can be used as `add p1/John Doe`.
- Items in square brackets are optional.
  - e.g. `p1/PLAYER_NAME [t/TAG]` can be used as `p1/John Doe t/friend` or as `p1/John Doe`.
- Items with `...` after them can be used multiple times including zero times.  
e.g. `[t/TAG]...` can be used as  (i.e. 0 times), `t/friend`, `t/friend t/family` etc.
- Parameters can be in any order.  
e.g. if the command specifies `p1/PLAYER_NAME p/PHONE_NUMBER`, `p/PHONE_NUMBER p1/PLAYER_NAME` is also acceptable.
- Extraneous parameters for commands that do not take in parameters (such as `help`, `list`, `exit` and `clear`) will be ignored.  
e.g. if the command specifies `help 123`, it will be interpreted as `help`.
- If you are using a PDF version of this document, be careful when copying and pasting commands that span multiple lines as space characters surrounding line-breaks may be omitted when copied over to the application.

### Viewing help: `help`

Shows a message explaining how to access the help page.



Format: `help`

### Adding a team: `addteam`

Adds a team to the PlayBook.

Format: `addteam tm/TEAM_NAME`

- `TEAM_NAME` must not be the same as an existing team in the PlayBook.
- `TEAM_NAME` should contain only alphanumeric characters, with no spaces.
- `TEAM_NAME` is case-insensitive, e.g. `u16` is the same as `U16`.
- `TEAM_NAME` should not be blank.

Examples:

- `addteam tm/u16`

## Adding a player: **add**

Adds a player to the PlayBook.

Format: **add** pl/PLAYER\_NAME p/PHONE\_NUMBER e/EMAIL a/ADDRESS tm/TEAM\_NAME [t/TAG]...

- **PLAYER\_NAME** should contain alphanumeric characters and spaces only.
- **PLAYER\_NAME** should not be blank.
- **PLAYER\_NAME** is case-insensitive, e.g. **john doe** is the same as **John Doe**.
- **PLAYER\_NAME** must not be the same as an existing player in the PlayBook.
- **PHONE\_NUMBER** should only contain numbers.
- **PHONE\_NUMBER** should be at least 3 digits long.
- **PHONE\_NUMBER** should not be blank.
- **EMAIL** should not be blank.
- **ADDRESS** should not be blank.
- **TEAM\_NAME** must be an existing team in the PlayBook. Use the **addteam** command to add a team first.
- **TEAM\_NAME** should contain only alphanumeric characters, with no spaces.
- **TEAM\_NAME** is case-insensitive, e.g. **u16** is the same as **U16**.
- **TEAM\_NAME** should not be blank.
- **TAG** should contain alphanumeric characters only.

**Tip:** A player can have any number of tags (including 0).

Examples:

- **add** pl/John Doe p/98765432 e/johnd@example.com a/John street, block 123, #01-01 tm/u16
- **add** pl/Betsy Crowe t/friend e/betsycrowe@example.com tm/u16 a/Newgate Prison p/1234567 t/criminal

## Assign player to team: **assignteam**

Assigns an existing player to another existing team.

Format: **assignteam** pl/PLAYER\_NAME tm/TEAM\_NAME

- **TEAM\_NAME** must be an existing team in the PlayBook. Use the **addteam** command to add a team first.
- **TEAM\_NAME** should contain only alphanumeric characters, with no spaces.
- **TEAM\_NAME** is case-insensitive, e.g. **u16** is the same as **U16**.
- **TEAM\_NAME** should not be blank.
- **PLAYER\_NAME** must be the same as an existing player in the PlayBook.
- **PLAYER\_NAME** is case-insensitive, e.g. **john doe** is the same as **John Doe**.
- **PLAYER\_NAME** should not be blank.
- **PLAYER\_NAME** must not already be assigned to **TEAM\_NAME**.

Examples:

- **assignteam** pl/John Doe tm/u16
- **assignteam** pl/Betsy Crowe tm/u16

## Assign injury to player: **assigninjury**

Assigns an injury status to an existing player.

Format: `assigninjury pl/PLAYER_NAME i/INJURY`

- `PLAYER_NAME` must be the same as an existing player in the PlayBook.
- `PLAYER_NAME` is case-insensitive, e.g. `john doe` is the same as `John Doe`.
- `PLAYER_NAME` should not be blank.
- `INJURY` is case-insensitive, e.g. `acl` is the same as `ACL`.
- `INJURY` should not be blank.
- `INJURY` should contain alphanumeric characters and spaces only.

**Tip:** Keyword `FIT` (in any letter case) is not allowed as a valid injury. Use `unassigninjury` instead to restore the player's injury status to `FIT`.

Examples:

- `assigninjury pl/John Doe i/ACL` assigns the injury status `ACL` to `John Doe` in the PlayBook.
- `assigninjury pl/Musiala i/fibula fracture` assigns the injury status `fibula fracture` to `Musiala` in the PlayBook.

Unassign injury from player: `unassigninjury`

Removes an injury status currently assigned to an existing player.

Format: `unassigninjury pl/PLAYER_NAME i/INJURY`

- `PLAYER_NAME` must be the same as an existing player in the PlayBook.
- `PLAYER_NAME` is case-insensitive, e.g. `john doe` is the same as `John Doe`.
- `PLAYER_NAME` should not be blank.
- `INJURY` is case-insensitive, e.g. `acl` is the same as `ACL`.
- `INJURY` should not be blank.
- `INJURY` should contain alphanumeric characters and spaces only.
- `INJURY` must match an injury that is already assigned to the specified player.
- If the player has no remaining injuries, the injury status of the player is reset to the default `FIT` status.
- The player must not already have the default `FIT` status.

Examples:

- `unassigninjury pl/John Doe i/ACL` removes the injury status `ACL` from `John Doe` in the Playbook.
- `unassigninjury pl/Musiala i/fibula fracture` removes the injury status `fibula fracture` from `Musiala` in the Playbook.

Creating a new position: `newposition`

Creates a new position in the PlayBook.

Format: `newposition ps/POSITION_NAME`

- `POSITION_NAME` must not be the same as an existing position in the PlayBook.
- `POSITION_NAME` should contain only alphanumeric characters, with no spaces.
- `POSITION_NAME` is case-insensitive, e.g. `fw` is the same as `FW`.

- `POSITION_NAME` should not be blank.

Examples:

- `newposition ps/LW` creates a position named `LW` (Left Wing).
- `newposition ps/ST` creates a position named `ST` (Striker).

## Assigning a position to player: `assignposition`

Assigns an existing position to an existing player in the PlayBook.

Format: `assignposition pl/PLAYER_NAME ps/POSITION_NAME`

- Assigns the position with the specified `POSITION_NAME` to the player with the specified `PLAYER_NAME`.
- `PLAYER_NAME` must be an existing player in the PlayBook.
- `PLAYER_NAME` is case-insensitive, e.g. `john doe` is the same as `John Doe`.
- `PLAYER_NAME` should not be blank.
- `POSITION_NAME` must be an existing position in the PlayBook. Use the `newposition` command to add a position first.
- `POSITION_NAME` should contain only alphanumeric characters, with no spaces.
- `POSITION_NAME` is case-insensitive, e.g. `fw` is the same as `FW`.
- `POSITION_NAME` should not be blank.
- The player must not already be assigned to the same position.

Examples:

- `assignposition pl/John Doe ps/LW` assigns the position `LW` to the player `John Doe`.
- `assignposition pl/Musiala ps/ST` assigns the position `ST` to the player `Musiala`.

## Assigning player as captain: `makecaptain`

Assigns an existing player in the PlayBook to be captain.

Format: `makecaptain pl/PLAYER_NAME`

- Assigns specified `PLAYER_NAME` to be captain.
- `PLAYER_NAME` must be an existing player in the PlayBook.
- `PLAYER_NAME` is case-insensitive, e.g. `john doe` is the same as `John Doe`.
- `PLAYER_NAME` should not be blank.
- The player must not already be an assigned captain.
- **Multiple captains per team are supported** - you can assign multiple players from the same team as captains to facilitate co-leadership structures (e.g., co-captains, senior/junior captain pairings).

Examples:

- `makecaptain pl/John Doe` assigns the player `John Doe` as a captain.
- `makecaptain pl/Sergio Ramos` assigns the player `Sergio Ramos` as a captain (even if from the same team as John Doe).
- `makecaptain pl/Jane Smith` assigns another player as captain - teams can have multiple captains.

## Unassigning player as captain: `stripcaptain`

Unassigns captain from existing player in the PlayBook.

Format: `stripcaptain pl/PLAYER_NAME`

- Unassigns specified `PLAYER_NAME` to be no longer captain.
- `PLAYER_NAME` must be an existing player in the PlayBook.
- `PLAYER_NAME` is case-insensitive, e.g. `john doe` is the same as `John Doe`.
- `PLAYER_NAME` should not be blank.
- The player must already be an assigned captain.

Examples:

- `stripcaptain pl/John Doe` unassigns the player `John Doe` as a captain.
- `stripcaptain pl/Sergio Ramos` unassigns the player `Sergio Ramos` as a captain.

### Note about Multiple Captains:

PlayBook supports **multiple captains per team**, allowing you to:

- Assign co-captains who share leadership responsibilities
- Create senior/junior captain pairings for mentorship
- Designate different captains for different competitions
- Develop leadership skills in multiple players simultaneously

When you use `listcaptain`, all captains across all teams will be listed, including multiple captains from the same team. This feature gives you the flexibility to structure team leadership in the way that best suits your coaching philosophy and team dynamics.

### Example scenario:

- Assign John Doe as captain of U16 team: `makecaptain pl/John Doe`
- Assign Jane Smith as another captain of U16 team: `makecaptain pl/Jane Smith`
- View all captains: `listcaptain` (both John and Jane will appear)

Listing all players: `list`

Shows a list of all players in the PlayBook.

Format: `list`

Listing all teams: `listteam`

Shows a list of all teams in the PlayBook.

Format: `listteam`

Listing all positions: `listposition`

Shows a list of all positions in the PlayBook.

Format: `listposition`

Listing all injured players: `listinjured`

Shows a list of the injured players in the PlayBook.

Format: `listinjured`

Listing all captains: `listcaptain`

Shows a list of all captains in the PlayBook across all teams.

Format: `listcaptain`

- Lists all players marked as captains.
- Multiple captains from the same team will be displayed together.
- Useful for quickly viewing all team leaders.

Examples:

- `listcaptain` displays all captains, which may include multiple captains from the same team.

Editing a player: `edit`

Edits an existing player in the PlayBook.

Format: `edit p1/PLAYER_NAME [n/NEW_PLAYER_NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS] [tm/TEAM_NAME] [t/TAG]...`

- Edits the player with the specified `PLAYER_NAME` from the PlayBook.
- At least one of the optional fields must be provided.
- Existing values will be updated to the input values.
- When editing the `NEW_PLAYER_NAME` or `TEAM_NAME` fields, comparisons are case-insensitive, i.e., values differing only by letter case are treated as identical.
- When editing tags, the existing tags of the player will be removed i.e, adding of tags is not cumulative.
- You can remove all the player's tags by typing `t/` without specifying any tags after it.
- `PLAYER_NAME` must be the same as an existing player in the PlayBook.
- `PLAYER_NAME` is case-insensitive, e.g. `john doe` is the same as `John Doe`.
- `PLAYER_NAME` should not be blank.
- `NEW_PLAYER_NAME` should contain alphanumeric characters and spaces only.
- `NEW_PLAYER_NAME` is case-insensitive, e.g. `john doe` is the same as `John Doe`.
- `NEW_PLAYER_NAME` must not be the same as an existing player in the PlayBook.
- `PHONE_NUMBER` should only contain numbers.
- `PHONE_NUMBER` should be at least 3 digits long.
- `TEAM_NAME` must be an existing team in the PlayBook. Use the `addteam` command to add a team first.
- `TEAM_NAME` should contain only alphanumeric characters, with no spaces.
- `TEAM_NAME` is case-insensitive, e.g. `u16` is the same as `U16`.
- `TEAM_NAME` should not be blank.
- `TAG` should contain alphanumeric characters only.

Examples:

- `edit p1/John Doe p/91234567 e/johndoe@example.com` edits the phone number and email address of `John Doe` to be `91234567` and `johndoe@example.com` respectively.



- `edit pl/John Doe n/Betsy Crower t/` edits the name of `John Doe` to be `Betsy Crower` and clears all existing tags.


## Locating players by name: `find`

Finds players whose names contain any of the given keywords.

Format: `find KEYWORD [MORE_KEYWORDS]`

- The search is case-insensitive. e.g `hans` will match `Hans`.
- The order of the keywords does not matter. e.g. `Hans Bo` will match `Bo Hans`.
- Only the name is searched.
- Only full words will be matched e.g. `Han` will not match `Hans`.
- Players matching at least one keyword will be returned (i.e. `OR` search). e.g. `Hans Bo` will return `Hans Gruber`, `Bo Yang`.

Examples:

- `find John` returns `john` and `John Doe`
  - `find alex david` returns `Alex Yeoh`, `David Li`
- result for 'find alex david'

## Filtering players by team, injury and/or position: `filter`

Filter players by team, injury and/or position.

Format: `filter [tm/TEAM_NAME] [i/INJURY] [ps/POSITION]`

- Shows all the players that matches the specified `TEAM_NAME`, `POSITION` and/or `INJURY` if they are provided.
- At least one of the optional fields must be provided.
- Duplicate flags are not allowed.
- The fields are all case-insensitive. e.g `acl` will match `ACL`.
- The order of the keywords for the injury field does not matter. e.g. `Leg Broken` will match `Broken Leg`.
- Only full words will be matched by the injury field. `ACL` will not match `ACLs`.

Examples:

- `filter tm/U16 ps/FW` shows players that are from the team `U16` and play the position `FW`.
- `filter ps/FW tm/U17 i/FIT` shows players that are from the team `U17`, play the position `FW` and are `FIT`.
- `filter i/Leg Broken ps/MF` shows players that play the position `MF` and have their `Leg Broken`.
- `filter tm/Chelsea` shows players that are from the team `Chelsea`.

## Deleting a player: `delete`

Deletes the specified player from the PlayBook.

Format: `delete pl/PLAYER_NAME`

- Deletes the player with the specified `PLAYER_NAME` from the PlayBook.

- `PLAYER_NAME` is case-insensitive, e.g `hans` will match `Hans`.
- The player to be deleted must exist in the PlayBook.
- The command can only delete one player at a time.

Examples:

- `delete pl/John Doe` deletes the player named `John Doe` from the PlayBook.
- `delete pl/Betsy Crowe` deletes the player named `Betsy Crowe` from the PlayBook.

### Deleting a team: `deleteteam`

Deletes the specified team from the PlayBook.

Format: `deleteteam tm/TEAM_NAME`

- Deletes the team with the specified `TEAM_NAME` from the PlayBook.
- Team deletion is only allowed if there are no players assigned to the team, i.e. all players assigned to the team must be reassigned to other teams or deleted first.
- `TEAM_NAME` must be an existing team in the PlayBook.
- `TEAM_NAME` should contain only alphanumeric characters, with no spaces.
- `TEAM_NAME` is case-insensitive, e.g. `u16` is the same as `U16`.
- `TEAM_NAME` should not be blank.
- The team to be deleted must exist in the PlayBook.
- The command can only delete one team at a time.

Examples:

- `deleteteam tm/u16` deletes the team named `u16` from the PlayBook, assuming it exists.

### Deleting a position: `deleteposition`

Deletes the specified position from the PlayBook.

Format: `deleteposition ps/POSITION_NAME`

- Deletes the position with the specified `POSITION_NAME` from the PlayBook.
- `POSITION_NAME` must be an existing position in the PlayBook.
- `POSITION_NAME` should contain only alphanumeric characters, with no spaces.
- `POSITION_NAME` is case-insensitive, e.g. `fw` is the same as `FW`.
- `POSITION_NAME` should not be blank.
- The position to be deleted must exist in the PlayBook.
- The command can only delete one position at a time.

Examples:

- `deleteposition ps/LW` deletes the position named `LW` from the PlayBook.
- `deleteposition ps/ST` deletes the position named `ST` from the PlayBook, assuming it exists.

### Clearing all entries : `clear`

Clears all entries from the PlayBook.

Format: `clear`

Exiting the program : `exit`

Exits the program.

Format: `exit`

## Saving the data

PlayBook data are saved in the hard disk automatically after any command that changes the data. There is no need to save manually.

## Editing the data file

PlayBook data are saved automatically as a JSON file `[JAR file location]/data/addressbook.json`. Advanced users are welcome to update data directly by editing that data file.

**Caution:** If your changes to the data file makes its format invalid, PlayBook will discard all data and start with an empty data file at the next run. Hence, it is recommended to take a backup of the file before editing it. Furthermore, certain edits can cause the PlayBook to behave in unexpected ways (e.g., if a value entered is outside the acceptable range). Therefore, edit the data file only if you are confident that you can update it correctly.

## Archiving data files `[coming in v2.0]`

*Details coming soon ...*

---

## FAQ

**Q:** How do I transfer my data to another Computer?

**A:** Install the app in the other computer and overwrite the empty data file it creates with the file that contains the data of your previous PlayBook home folder.

---

## Known issues

1. **When using multiple screens**, if you move the application to a secondary screen, and later switch to using only the primary screen, the GUI will open off-screen. The remedy is to delete the `preferences.json` file created by the application before running the application again.
2. **If you minimize the Help Window** and then run the `help` command (or use the `Help` menu, or the keyboard shortcut `F1`) again, the original Help Window will remain minimized, and no new Help Window will appear. The remedy is to manually restore the minimized Help Window.

---

## Command summary

Action	Format, Examples
--------	------------------

---

Action	Format, Examples
<b>Add Player</b>	<p>add pl/PLAYER_NAME p/PHONE_NUMBER e/EMAIL a/ADDRESS tm/TEAM_NAME [i/INJURY] [t/TAG]...</p> <p>e.g., add pl/James Ho p/22224444 e/jamesho@example.com a/123, Clementi Rd, 1234665 tm/u16 i/ACL t/friend t/colleague</p>
<b>Add Team</b>	<p>addteam tm/TEAM_NAME</p> <p>e.g., addteam tm/u16</p>
<b>Assign Player to Team</b>	<p>assignteam pl/PLAYER_NAME tm/TEAM_NAME</p> <p>e.g., assignteam pl/John Doe tm/u16</p>
<b>Assign Injury to Player</b>	<p>assigninjury pl/PLAYER_NAME i/INJURY</p> <p>e.g., assigninjury pl/John Doe i/ACL</p>
<b>Unassign Injury from Player</b>	<p>unassigninjury pl/PLAYER_NAME</p> <p>e.g., unassigninjury pl/John Doe</p>
<b>Create New Position</b>	<p>newposition ps/POSITION_NAME</p> <p>e.g., newposition ps/LW</p>
<b>Assign Position to Player</b>	<p>assignposition pl/PLAYER_NAME ps/POSITION_NAME</p> <p>e.g., assignposition pl/John Doe ps/LW</p>
<b>Clear</b>	clear
<b>Delete Player</b>	<p>delete pl/PLAYER_NAME</p> <p>e.g., delete pl/James Ho</p>
<b>Delete Team</b>	<p>deleteteam tm/TEAM_NAME</p> <p>e.g., deleteteam tm/u16</p>
<b>Delete Position</b>	<p>deleteposition ps/POSITION_NAME</p> <p>e.g., deleteposition ps/LW</p>
<b>Edit</b>	<p>edit pl/PLAYER_NAME [n/NEW_PLAYER_NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS] [tm/TEAM_NAME] [i/INJURY] [t/TAG]...</p> <p>e.g., edit pl/John Doe n/James Lee e/jameslee@example.com</p>
<b>Find</b>	<p>find KEYWORD [MORE_KEYWORDS]</p> <p>e.g., find James Jake</p>
<b>Filter Players</b>	<p>filter [tm/TEAM_NAME] [i/INJURY] [ps/POSITION]</p> <p>e.g., filter tm/U16 i/ACL ps/FW</p>
<b>List</b>	list
<b>List Teams</b>	listteams
<b>Make Captain</b>	<p>makecaptain pl/PLAYER_NAME</p> <p>e.g., makecaptain pl/John Doe</p>
<b>Strip Captain</b>	<p>stripcaptain pl/PLAYER_NAME</p> <p>e.g., stripcaptain pl/John Doe</p>

Action	Format, Examples
List Captains	listcaptain
List Teams	listteam
List Positions	listposition
List Injured Players	listinjured
Help	help