

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).

Table of Contents

- PlayBook User Guide
 - Introduction
 - Who is PlayBook for?
 - What you should know before using PlayBook
 - Why choose PlayBook?
 - Quick start
 - Installation and setup
 - Understanding the PlayBook GUI
 - Your First Commands
 - 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: `listteam`
 - Listing all positions: `listposition`
 - Listing all injured players: `listinjured`
 - Listing all captains: `listcaptains`
 - 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
 - Generating sample data

- [Archiving data files](#) [coming in v2.0]
 - [FAQ](#)
 - [Known issues](#)
 - [Command summary](#)
-

Introduction

Who is PlayBook for?

PlayBook is designed specifically for **semi-professional youth football coaches** who:

- Manage multiple teams (e.g., U16, U18, U21 squads)
- Need to track 20-50+ players across different age groups
- Prefer keyboard-based workflows for faster data entry
- Are comfortable with basic command-line operations
- Need quick access to player availability, injuries, and team assignments

What you should know before using PlayBook

Technical Requirements:

- Basic familiarity with command-line interfaces (typing commands rather than clicking buttons)
- Basic ability to navigate folders and run programs from a terminal
- Java 17 installed on your computer

No prior coding experience needed - all commands are simple, English-based instructions like add, delete, and list.

Why choose PlayBook?

Unlike traditional contact management apps, PlayBook is **optimized for football team management**:

- **Speed:** Type commands faster than clicking through multiple menus
 - **Team-focused:** Built-in support for teams, positions, injuries, and captaincy
 - **Flexible filtering:** Quickly find available players by position, team, or injury status
 - **Always accessible:** Works offline, with instant local storage
-

Quick start

Installation and setup

1. Ensure you have Java 17 or above installed in your Computer.

Quick help: Checking your Java installation

Mac users: Ensure you have the precise JDK version prescribed [here](#).

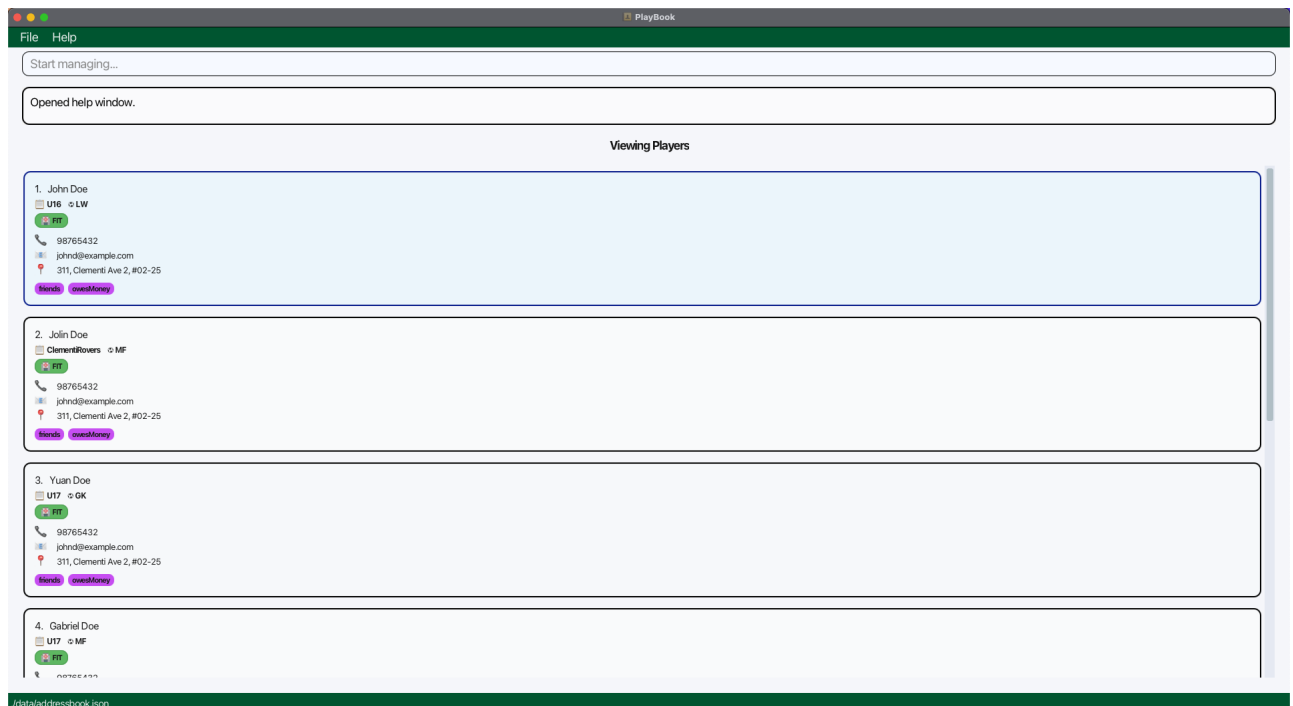
Windows users: Ensure you have the precise JDK version prescribed [here](#).

Linux users: Ensure you have the precise JDK version prescribed [here](#).

You can check your Java version by running the command `java --version` in your command terminal.

2. Download the latest `.jar` file from [here](#).
3. Copy the file to the folder you want to use as the *home folder* for your PlayBook.
4. 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.



Understanding the PlayBook GUI

The PlayBook window consists of:

- **Command Box** (top): Where you type your commands
- **Result Display**: Shows feedback messages after each command
- **Player List Panel**: Displays all players matching your current view
- **Player Card**: Shows individual player details including name, team, position, injury status, phone, email, address, and tags

Your First Commands

Type commands in the command box and press Enter to execute them. Here's a quick tutorial:

1. **Start fresh** - Type `clear` and press Enter to remove all sample data.

Expected output: "Address book has been cleared!" and the player list becomes empty.

2. **Create your first team** - Type `addteam tm/U16` and press Enter.

Expected output: "New team added: U16"

3. **Add your first player** - Type:

```
add pl/Alex Tan p/87654321 e/alex@example.com a/123 Main St tm/U16
```

Expected output: "New person added: Alex Tan; Phone: 87654321; Email: alex@example.com; Address: 123 Main St; Team: U16; Injury: FIT; Position: ; Tags: "

You should now see Alex Tan's player card in the player list.

4. **View all players** - Type `list` to see all players in your PlayBook.

More example commands to try:

- `addteam tm/U18`: Add another team named `U18` to the PlayBook.
- `add pl/John Doe p/98765432 e/johnd@example.com a/John street, block 123, #01-01 tm/U16` : Add another player to the U16 team.
- `newposition ps/LW`: Create a new position named `LW` (Left Wing) in the PlayBook.
- `assignposition pl/Alex Tan ps/LW`: Assign the position `LW` to `Alex Tan` (create the position first using `newposition`).
- `assigninjury pl/Alex Tan i/ACL` : Mark `Alex Tan` as injured with an ACL injury.
- `filter i/FIT` : View only players who are fit (not injured).
- `delete pl/John Doe` : Delete `John Doe` from the PlayBook.
- `help` : Open the help window.
- `exit` : Exit the app.

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

Features

Notes about the command format:

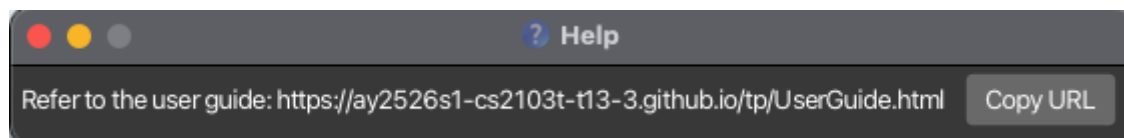
- Words in `UPPER_CASE` are the parameters to be supplied by the user.
e.g. in `add pl/PLAYER_NAME`, `PLAYER_NAME` is a parameter which can be used as `add pl/John Doe`.
- Items in square brackets are optional.
 - e.g `pl/PLAYER_NAME [t/TAG]` can be used as `pl/John Doe t/friend` or as `pl/John Doe`.

- 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`

Requirements:

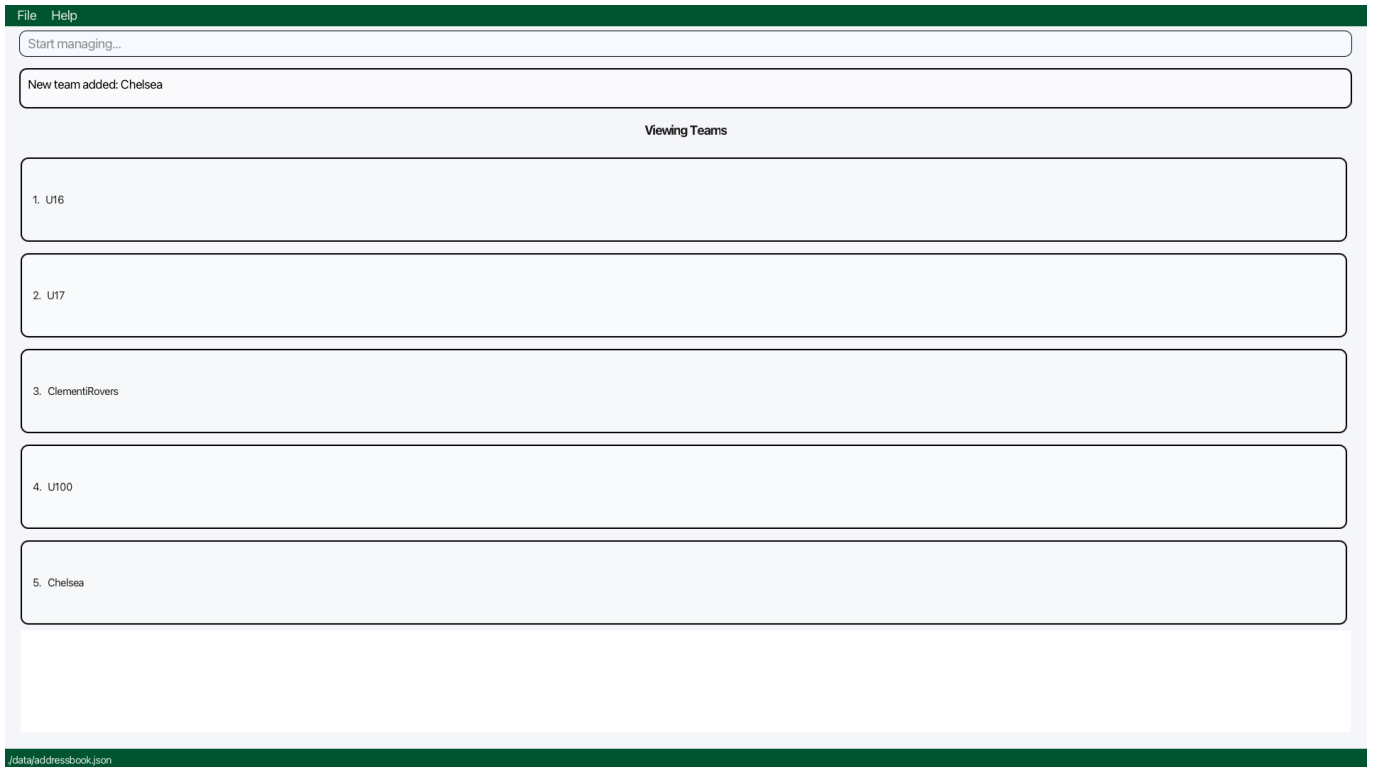
- `TEAM_NAME` must not be the same as an existing team in the PlayBook.
- `TEAM_NAME` should contain only alphanumeric characters, with no spaces.
- `TEAM_NAME` is case-insensitive, e.g. `u16` is the same as `U16`.
- `TEAM_NAME` should not be blank.

Warning: Team names cannot contain spaces. Use naming conventions like `U16`, `U18`, or `Reserves` instead of `U 16` or `U 18`.

Examples:

- `addteam tm/U16` - Creates a team for under-16 players
- `addteam tm/U18` - Creates a team for under-18 players
- `addteam tm/Reserves` - Creates a reserves team

Expected output: "New team added: U16" (or the team name you specified)



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]...

Requirements:

- **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.

Warning: You must create the team first using **addteam** before adding players to it. If the team doesn't exist, you'll get an error message.

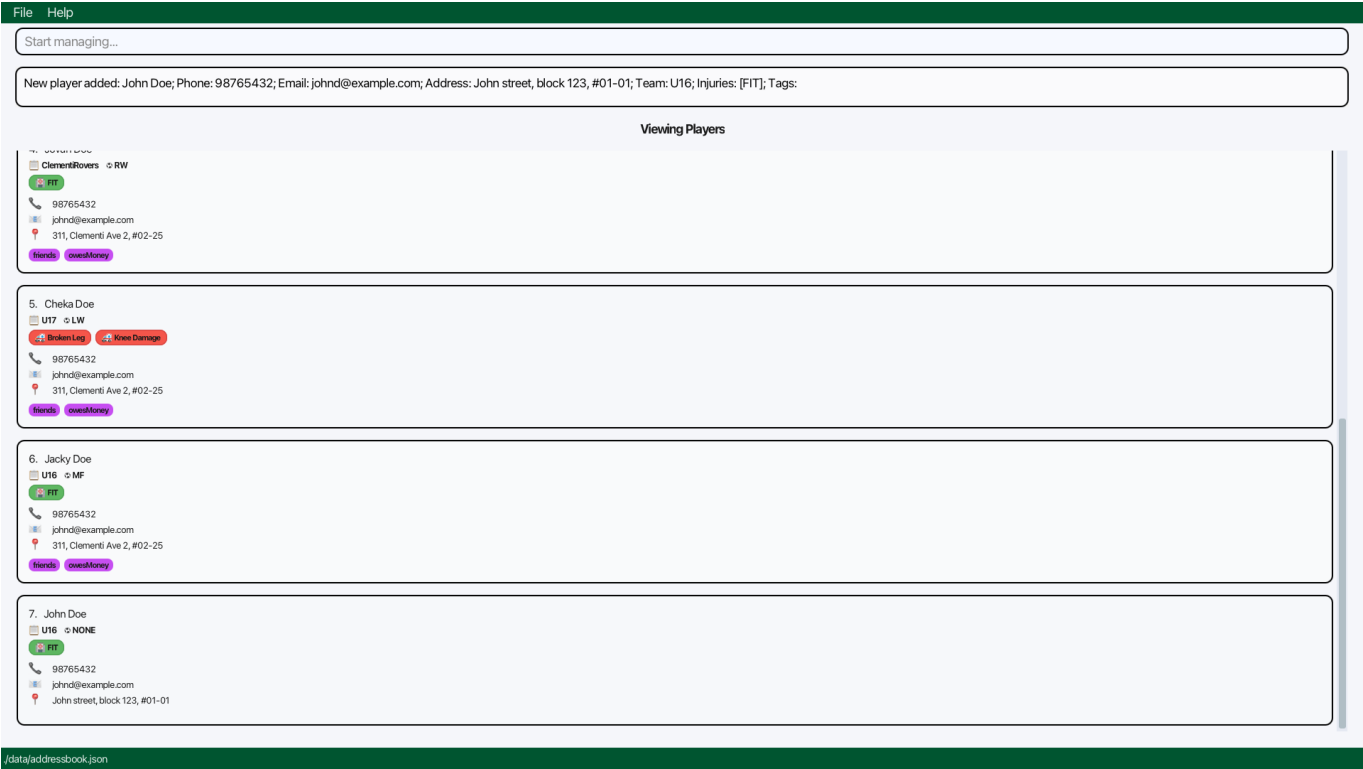
Tip: A player can have any number of tags (including 0). Use tags to mark players as "scholarship", "youth", "newcomer", etc.

Examples:

- add pl/John Doe p/98765432 e/johnd@example.com a/John street, block 123, #01-01 tm/U16
- add pl/Betsy Crowe p/1234567 e/betsycrowe@example.com a/Newgate Prison tm/U16 t/friend t/scholarship

Expected output: "New person added: John Doe; Phone: 98765432; Email: johnd@example.com; Address: John street, block 123, #01-01; Team: U16; Injury: FIT; Position: ; Tags: "

The player will appear in the player list panel with a "FIT" injury status and no position assigned by default.



Assign player to team: `assignteam`

Assigns an existing player to another existing team.

Format: `assignteam pl/PLAYER_NAME tm/TEAM_NAME`

Requirements:

- `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`.

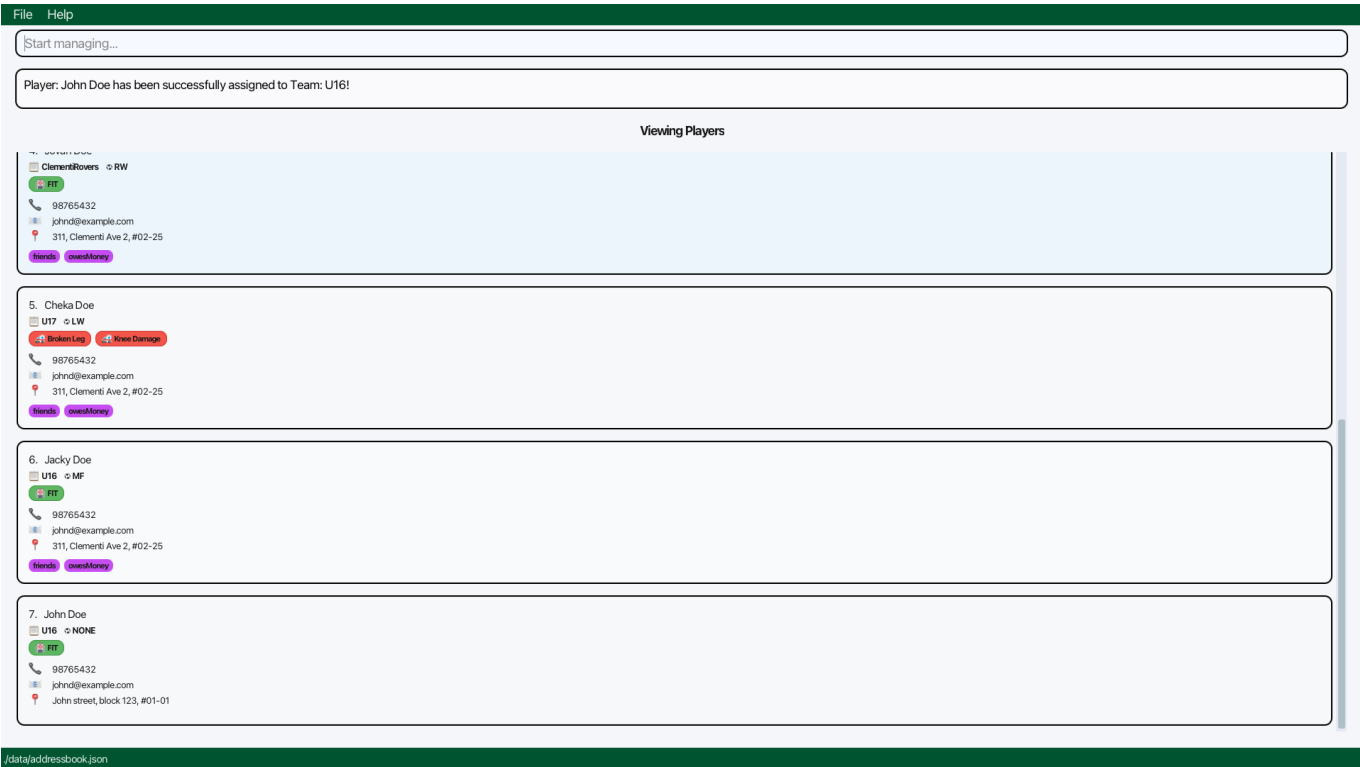
Tip: Use this command when promoting players between age groups (e.g., moving a player from U16 to U18) or reassigning players to different squads.

Examples:

- `assignteam pl/John Doe tm/U16` - Moves John Doe to the U16 team
- `assignteam pl/Betsy Crowe tm/U18` - Moves Betsy Crowe to the U18 team

Expected output: "Assigned John Doe to team: U16"

The player's team will be immediately updated in their player card.



Assign injury to player: `assigninjury`

Assigns an injury status to an existing player.

Format: `assigninjury pl/PLAYER_NAME i/INJURY`

Requirements:

- `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.

Warning: The 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`.

Tips:

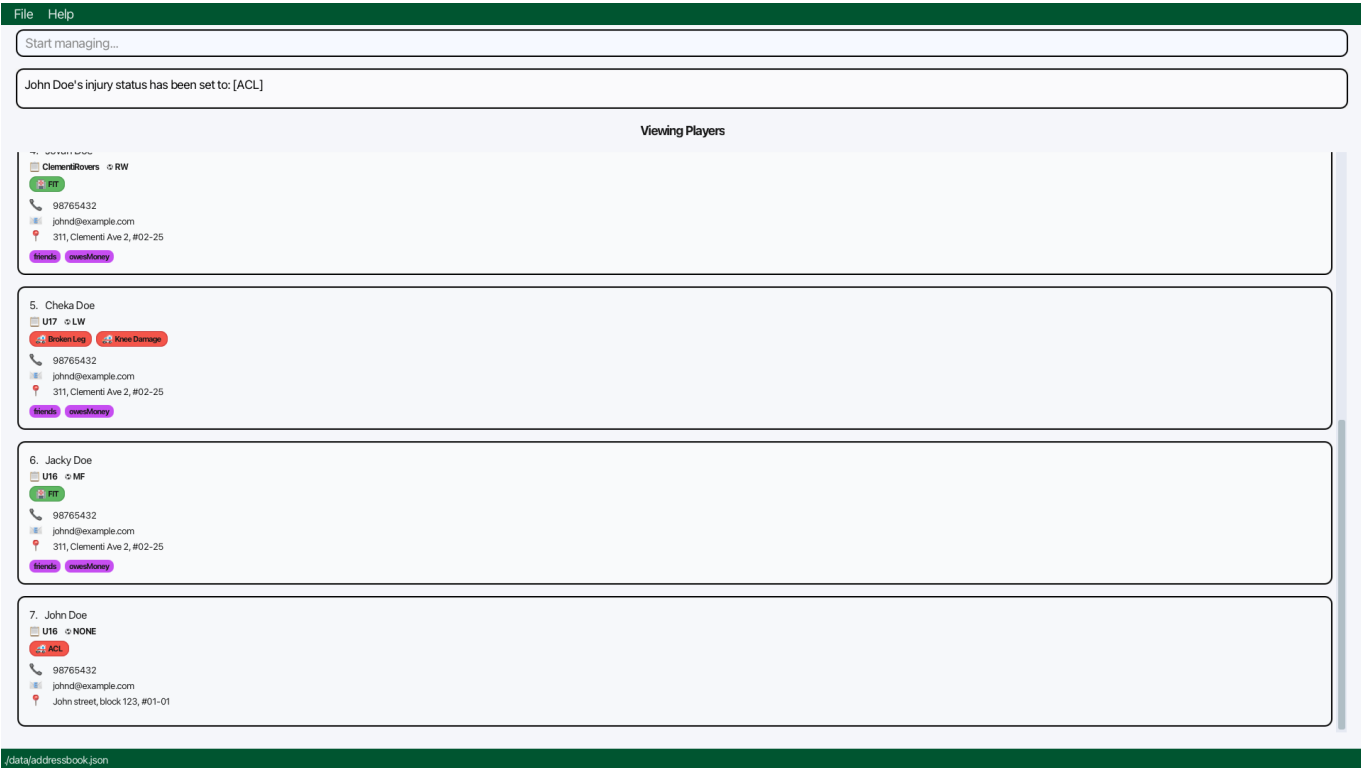
- Players can have multiple concurrent injuries. Simply use `assigninjury` multiple times with different injury names.
- Use specific injury names (e.g., "ACL", "hamstring strain", "ankle sprain") for better tracking.
- After assigning an injury, use `listinjured` to see all injured players at a glance.

Examples:

- `assigninjury pl/John Doe i/ACL` - Marks John Doe with an ACL injury
- `assigninjury pl/Musiala i/fibula fracture` - Marks Musiala with a fibula fracture
- `assigninjury pl/John Doe i/hamstring strain` - Adds a second injury to John Doe

Expected output: "Assigned injury ACL to John Doe"

The player's injury status will be updated immediately in their player card, replacing "FIT" with the injury name.



Unassign injury from player: `unassigninjury`

Removes an injury status currently assigned to an existing player.

Format: `unassigninjury pl/PLAYER_NAME i/INJURY`

Requirements:

- `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.

- The player must not already have the default **FIT** status.

Tips:

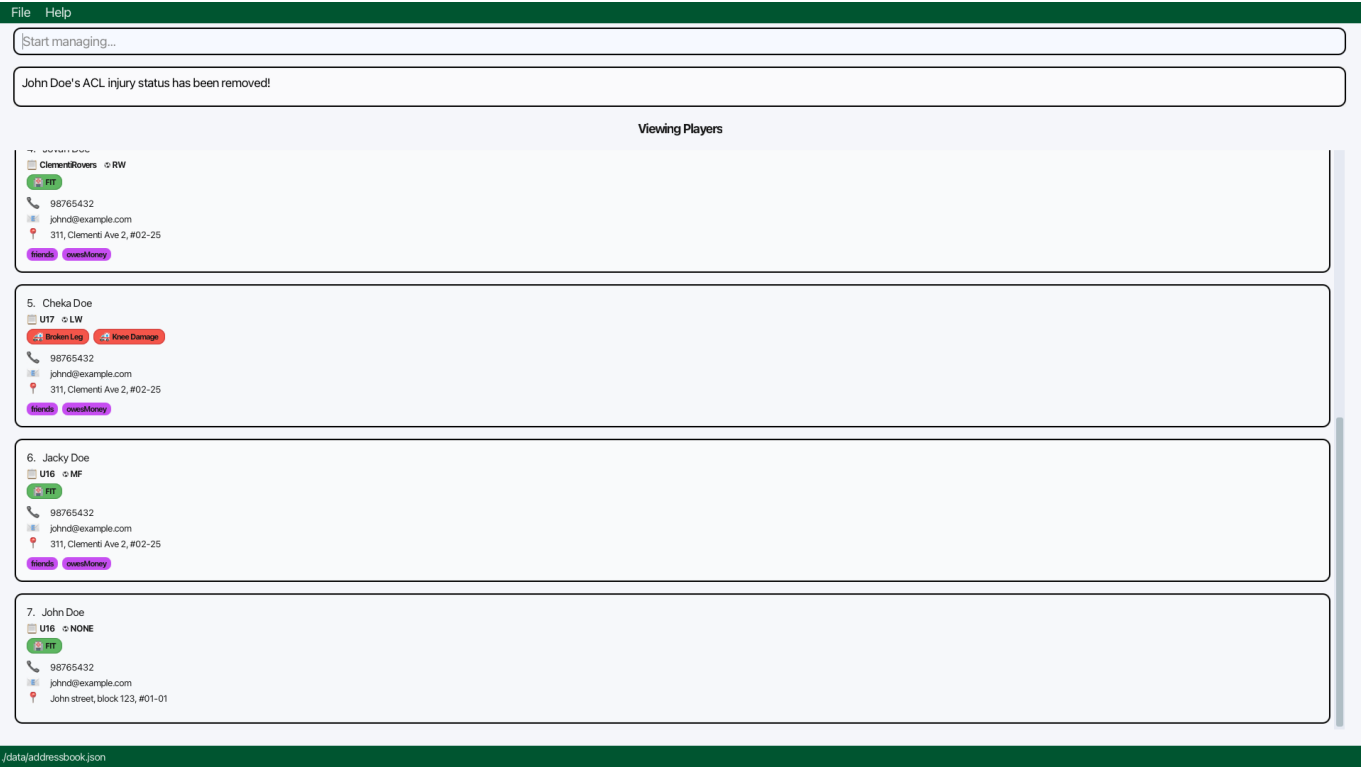
- If the player has multiple injuries, removing one injury will keep the others. Only when all injuries are removed will the status return to **FIT**.
- The injury name must match exactly (case-insensitive) with what was assigned.
- Use **listinjured** to check current injury statuses before unassigning.

Examples:

- **unassigninjury pl/John Doe i/ACL** - Removes the ACL injury from John Doe
- **unassigninjury pl/Musiala i/fibula fracture** - Removes the fibula fracture from Musiala

Expected output: "Removed injury ACL from John Doe"

If this was the player's only injury, their status will automatically return to "FIT".



Creating a new position: **newposition**

Creates a new position in the PlayBook.

Format: **newposition ps/POSITION_NAME**

Requirements:

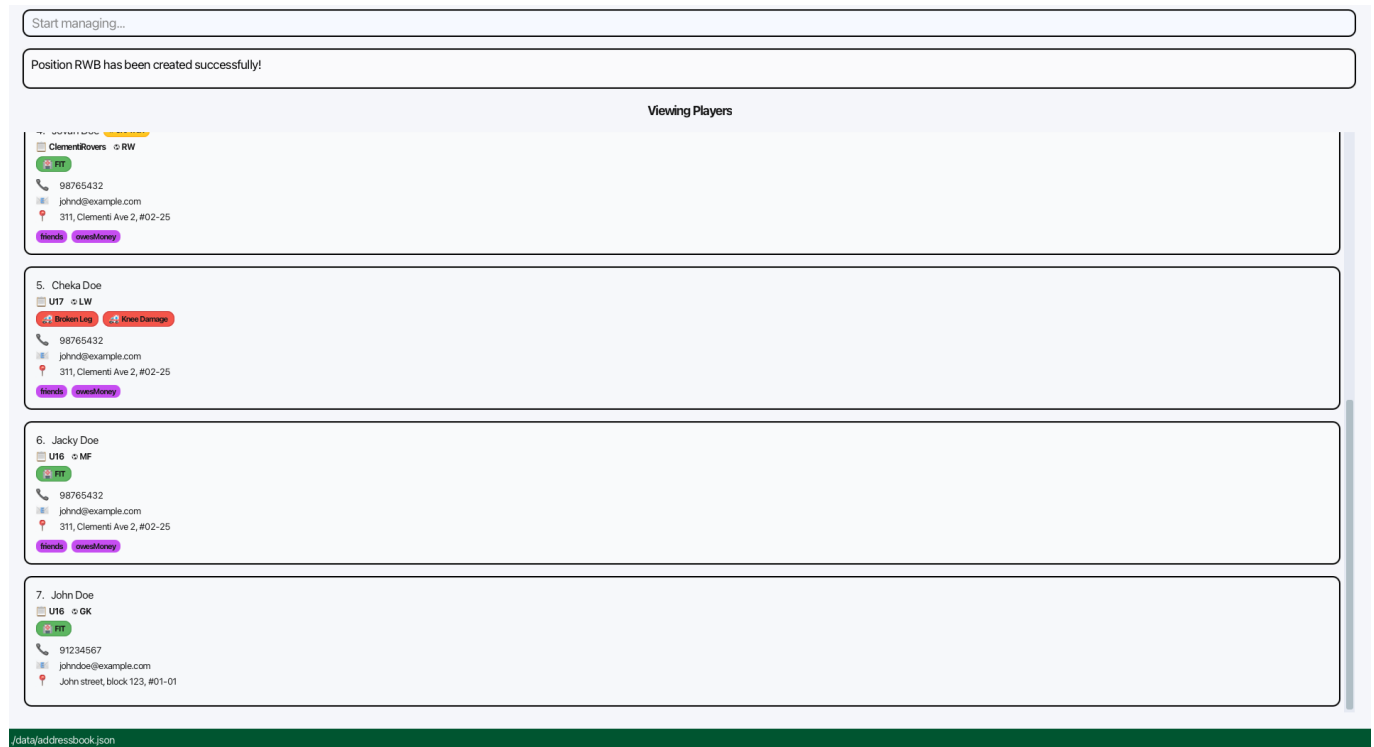
- **POSITION_NAME** must not be the same as an existing position in the PlayBook.
- **POSITION_NAME** should contain only alphanumeric characters, with no spaces.
- **POSITION_NAME** is case-insensitive, e.g. **fw** is the same as **FW**.
- **POSITION_NAME** should not be blank.

Tip: Create standard football positions like GK (Goalkeeper), CB (Center Back), LW (Left Wing), ST (Striker), etc. for easier team management.

Examples:

- `newposition ps/LW` - Creates Left Wing position
- `newposition ps/ST` - Creates Striker position
- `newposition ps/GK` - Creates Goalkeeper position

Expected output: "Position RWB has been created successfully!" (or the position name you specified)



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`

Requirements:

- `PLAYER_NAME` must be an existing player in the PlayBook.
- `PLAYER_NAME` is case-insensitive, e.g. `johndoe` 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.

Warning: You must create the position first using `newposition` before assigning it to players. If the position doesn't exist, you'll get an error message.

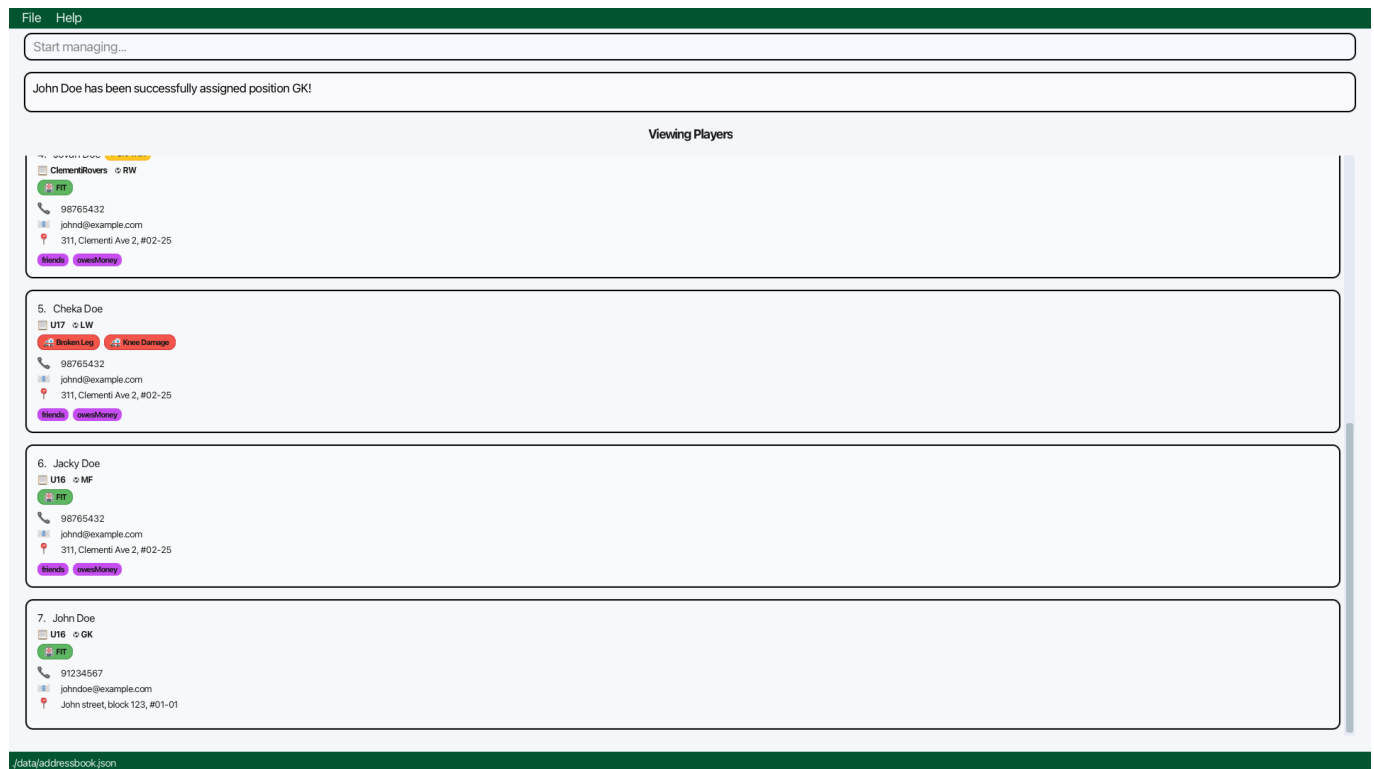
Tip: Players can be assigned multiple positions. Use this feature to track versatile players who can play in different roles.

Examples:

- `assignposition pl/John Doe ps/LW` - Assigns Left Wing position to John Doe
- `assignposition pl/Musiala ps/ST` - Assigns Striker position to Musiala

Expected output: "John Doe has been successfully assigned position GK!"

The position will be immediately visible in the player's card.



Assigning player as captain: `makecaptain`

Assigns an existing player in the PlayBook to be captain.

Format: `makecaptain pl/PLAYER_NAME`

Requirements:

- `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.

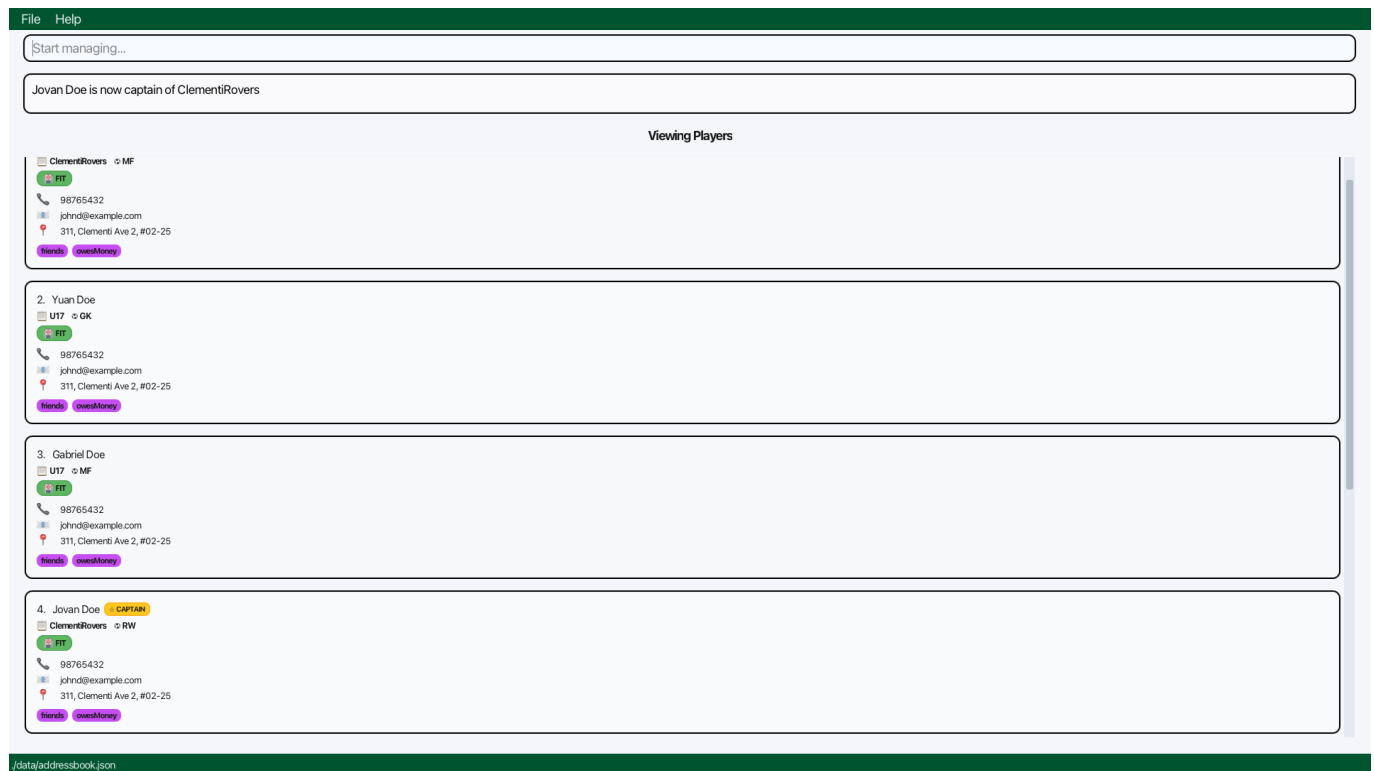
Tip: You can have multiple captains for different teams. Use `listcaptains` to see all current team captains.

Examples:

- `makecaptain pl/John Doe` - Makes John Doe a captain
- `makecaptain pl/Sergio Ramos` - Makes Sergio Ramos a captain

Expected output: "Jovan Doe is now a captain of ClementiRovers"

A captain badge or indicator will appear on the player's card.



Unassigning player as captain: **stripcaptain**

Removes captain status from an existing player in the PlayBook.

Format: **stripcaptain** pl/PLAYER_NAME

Requirements:

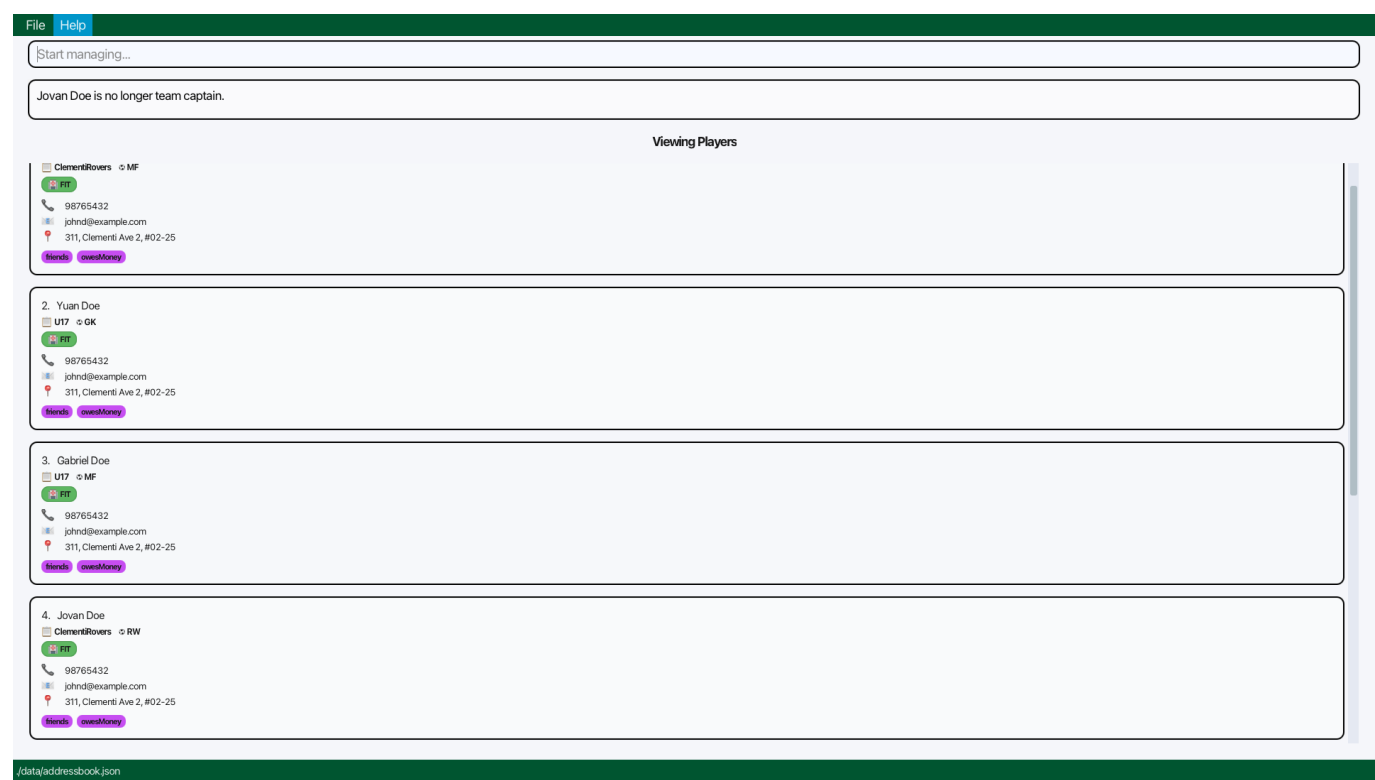
- **PLAYER_NAME** must be an existing player in the PlayBook.
- **PLAYER_NAME** is case-insensitive, e.g. **john doe** is the same as **John Doe**.
- **PLAYER_NAME** should not be blank.
- The player must already be an assigned captain.

Examples:

- **stripcaptain** pl/John Doe - Removes captain status from John Doe
- **stripcaptain** pl/Sergio Ramos - Removes captain status from Sergio Ramos

Expected output: "Jovan Doe is no longer team captain"

The captain badge/indicator will be removed from the player's card.



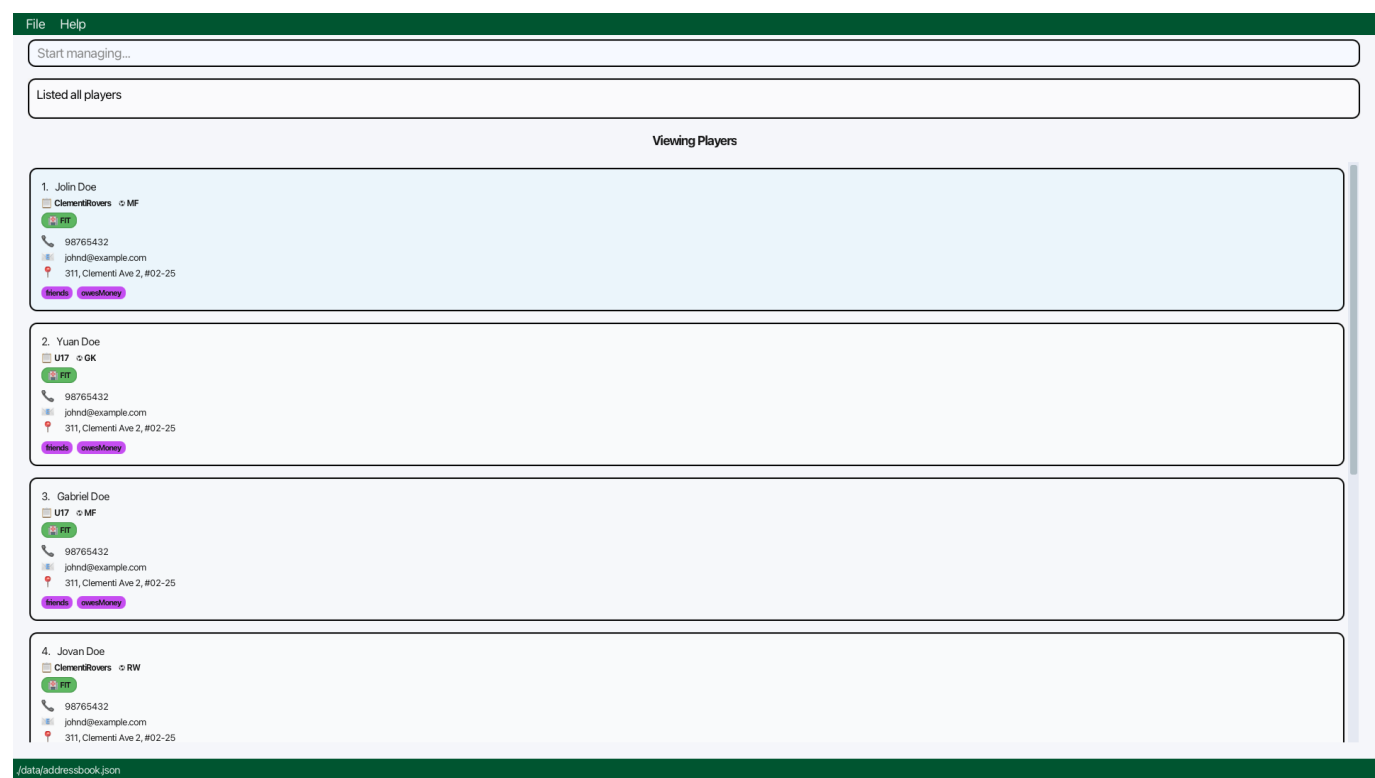
Listing all players: `list`

Shows a list of all players in the PlayBook.

Format: `list`

Tip: Use this command to reset your view after using filters or searches. It displays all players regardless of team, position, or injury status.

Expected output: All players in your PlayBook will be displayed in the player list panel.



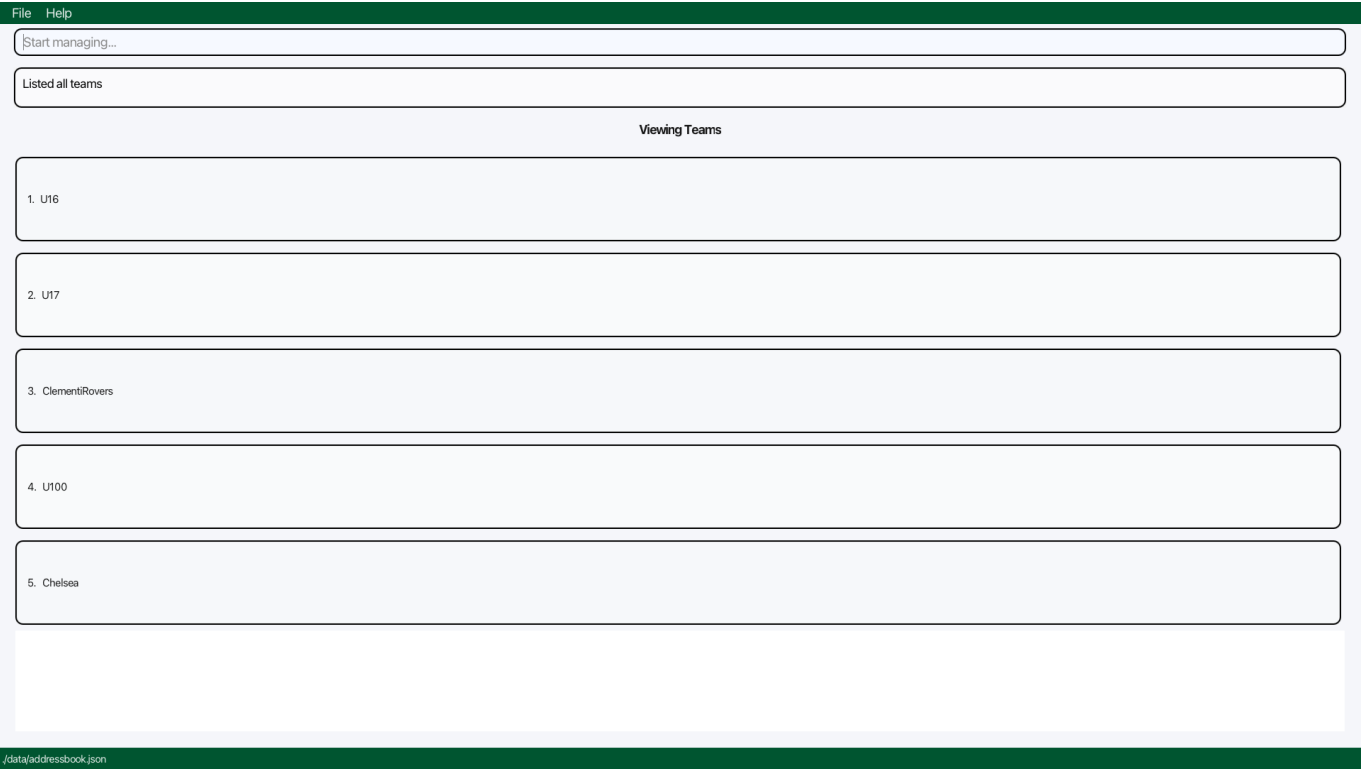
Listing all teams: `listteam`

Shows a list of all teams in the PlayBook.

Format: `listteam`

Tip: Use this to quickly see all teams you've created. Helpful for verifying team names before adding new players.

Expected output: A list of all team names will be displayed in the result box (e.g., "U16, U18, Reserves").



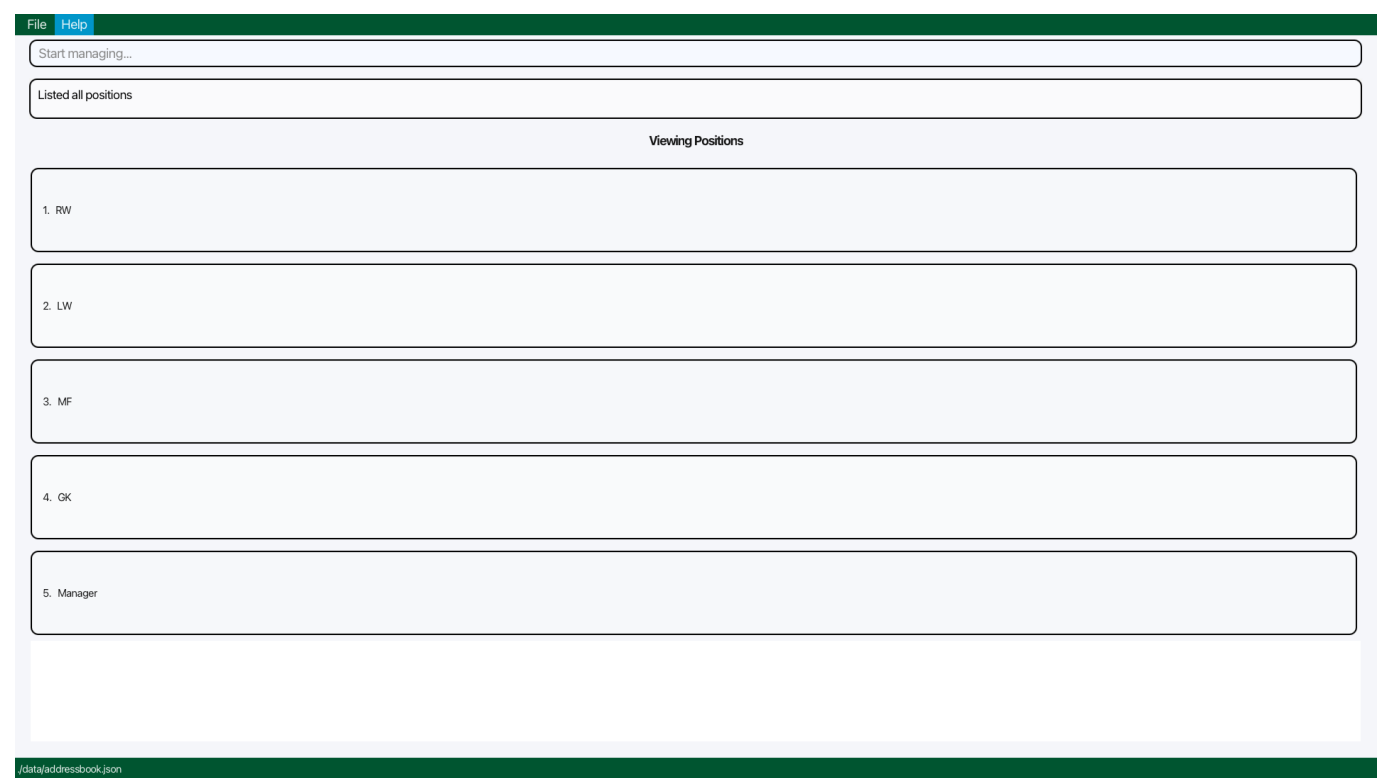
Listing all positions: `listposition`

Shows a list of all positions in the PlayBook.

Format: `listposition`

Tip: Use this to check which positions you've already created before assigning positions to players.

Expected output: A list of all position names will be displayed (e.g., "LW, ST, GK, CB").



Listing all injured players: `listinjured`

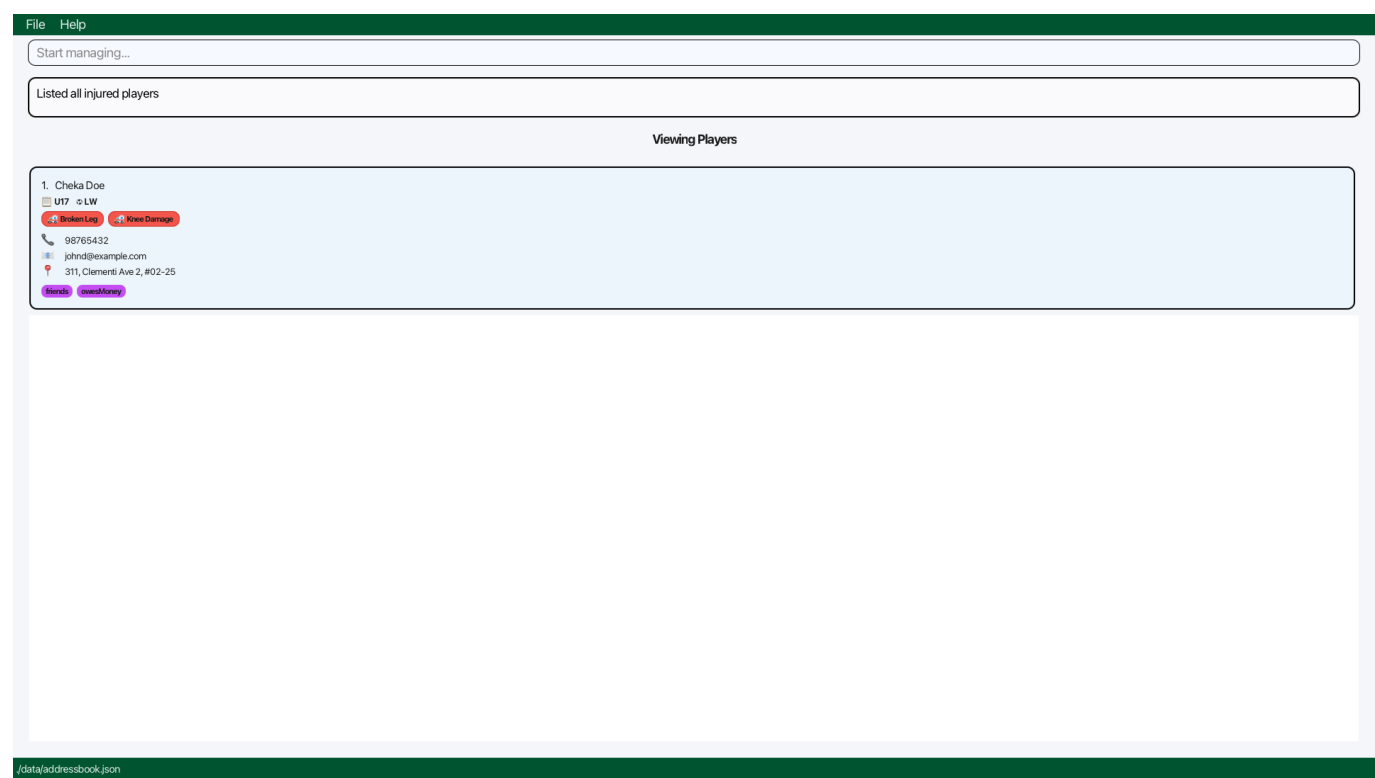
Shows a list of all injured players in the PlayBook.

Format: `listinjured`

Tips:

- Use this command before match day to quickly see who's unavailable.
- Combine with team information in the player cards to see which teams are affected by injuries.
- This shows all players with any injury status (not "FIT").

Expected output: Only players with injuries (non-FIT status) will be displayed in the player list panel. Their injury details will be visible on their player cards.



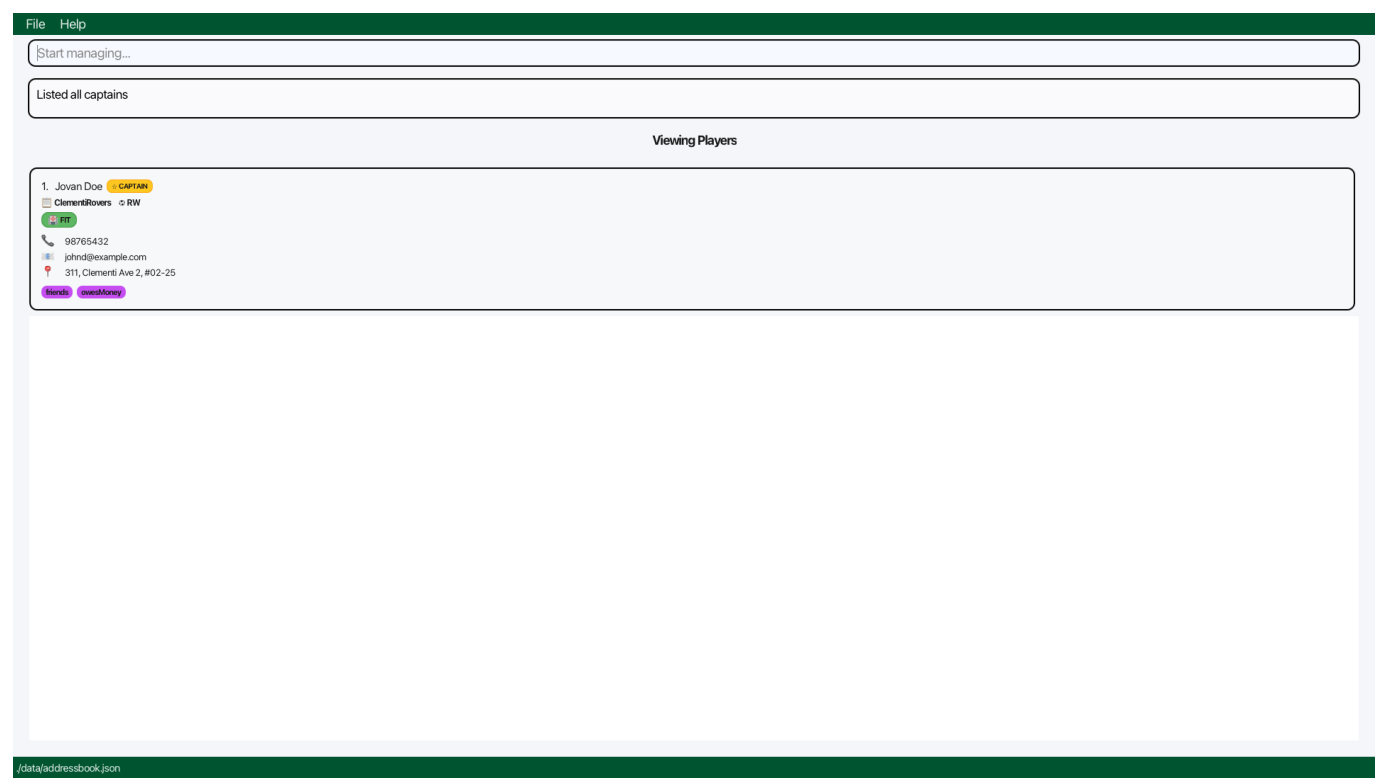
Listing all captains: `filtercaptains`

Shows a list of all captains in the PlayBook.

Format: `filtercaptains`

Tip: Useful for quickly identifying team leaders across all your squads.

Expected output: Only players designated as captains will be displayed in the player list panel.



Editing a player: `edit`

Edits an existing player in the PlayBook. Existing values will be updated to the input values.

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

Requirements:

- At least one of the optional fields must be provided.
- `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.

Warning: When editing tags, all existing tags will be replaced with the new ones. If you want to keep existing tags, you must include them in the edit command.

Tips:

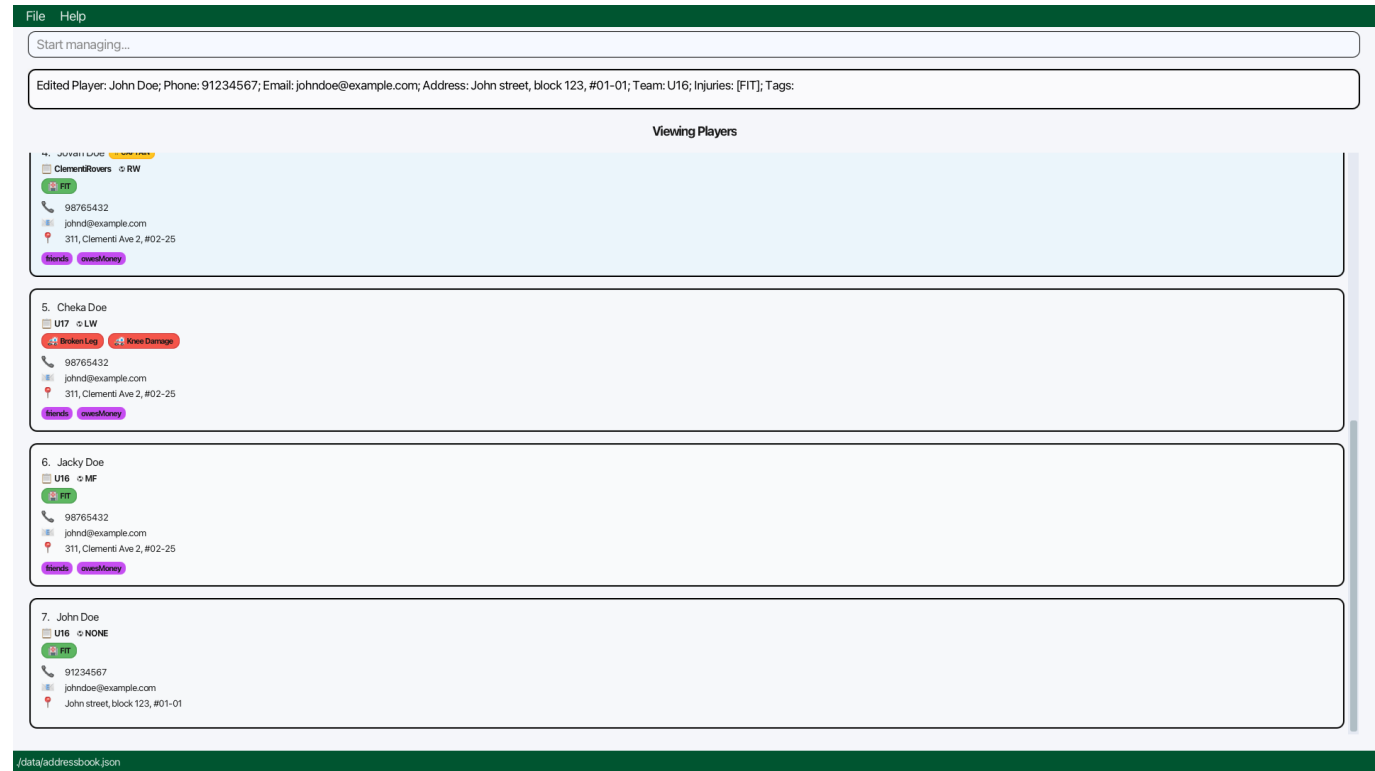
- You can edit multiple fields at once (e.g., both phone and email).
- To remove all tags, use `t/` without any tag names after it.
- You cannot edit injury status or position through this command - use dedicated commands instead.

Examples:

- `edit pl/John Doe p/91234567 e/johndoe@example.com` - Updates phone and email
- `edit pl/John Doe n/Betsy Crower t/` - Renames player and removes all tags
- `edit pl/John Doe tm/U18 t/captain t/veteran` - Moves to U18 team and adds tags

Expected output: "Edited Player: John Doe; Phone: 91234567; Email: johndoe@example.com; Address: John street, block 123, #01-01; Team: U16; Injuries: [FIT]; Tags: "

The player card will immediately reflect all changes.



Locating players by name: **find**

Finds players whose names contain any of the given keywords.

Format: **find** KEYWORD [MORE_KEYWORDS]

Requirements:

- **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** will only match full words 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**.

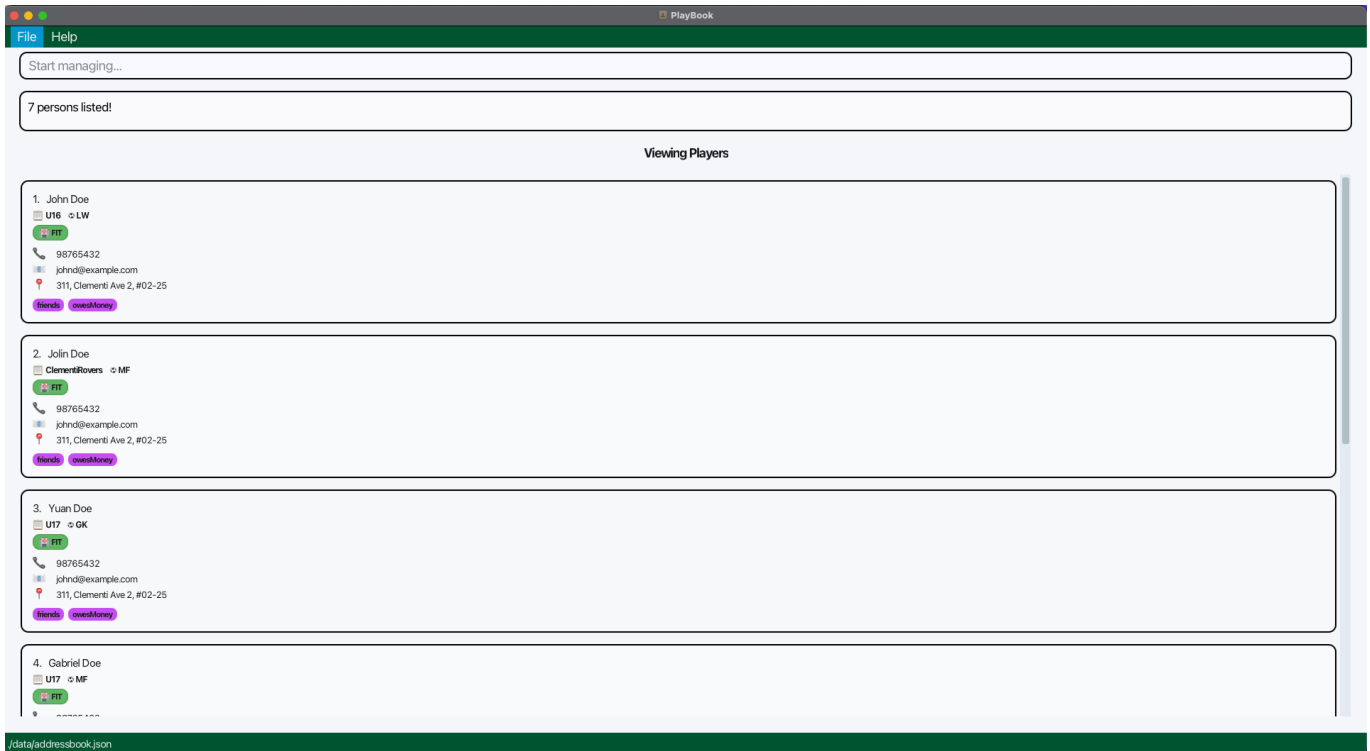
Tips:

- Use **find** when you remember part of a player's name but not the full name.
- To return to viewing all players after a search, use the **list** command.
- For more advanced filtering by team, position, or injury, use the **filter** command instead.

Examples:

- **find John** - Returns players like "John Doe" and "John Smith"
- **find alex david** - Returns "Alex Yeoh" and "David Li" (matches either keyword)

Expected output: "X persons listed!" where X is the number of matching players. The player list panel will show only the matching players.



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_NAME]

Requirements:

- At least one of the optional fields must be provided.
- **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.
- **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.
- **INJURY** will only match full words e.g. **ACL** will not match **ACLS**.
- **INJURY** matching at least one keyword will be returned (i.e. **OR** search). e.g. **Leg Arm** will return **Leg Broken, Arm Fractured**.
- **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.

Tips:

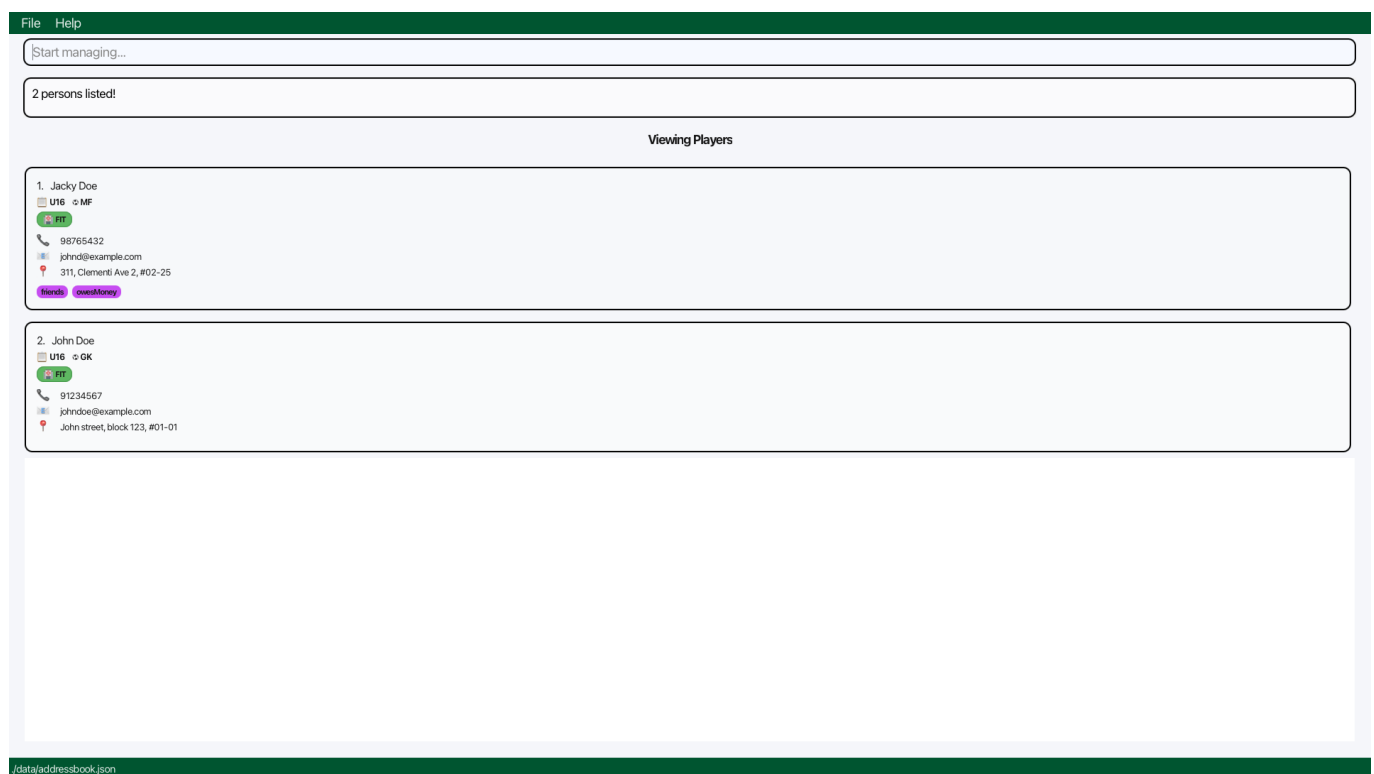
- **Pre-match planning:** Use **filter tm/U16 i/FIT** to see all available U16 players.

- **Position planning:** Use `filter ps/ST i/FIT` to find all fit strikers across teams.
- **Injury tracking:** Use `filter tm/U18 i/ACL` to see all U18 players with ACL injuries.
- Combine multiple filters for precise results. All conditions must be met (AND logic).

Examples:

- `filter tm/U16 ps/FW` - Shows U16 players who play Forward
- `filter ps/FW tm/U17 i/FIT` - Shows fit Forwards from U17 team
- `filter i/Leg Broken ps/MF` - Shows Midfielders with leg broken injury
- `filter tm/Chelsea` - Shows all Chelsea team players
- `filter tm/Manchester i/Leg Arm` - Shows all Manchester team players with an injuries with the words Leg or Arm in them

Expected output: "X persons listed!" where X is the number of players matching all filter criteria. Only matching players will be displayed.



Deleting a player: `delete`

Deletes the specified player from the PlayBook.

Format: `delete pl/PLAYER_NAME`

Requirements:

- `PLAYER_NAME` is case-insensitive, e.g `hans` will match `Hans`.
- The player to be deleted must exist in the PlayBook.

Warning: This action cannot be undone! The player and all their information (team, position, injuries, tags) will be permanently removed. Consider making a backup of your data file before bulk deletions.

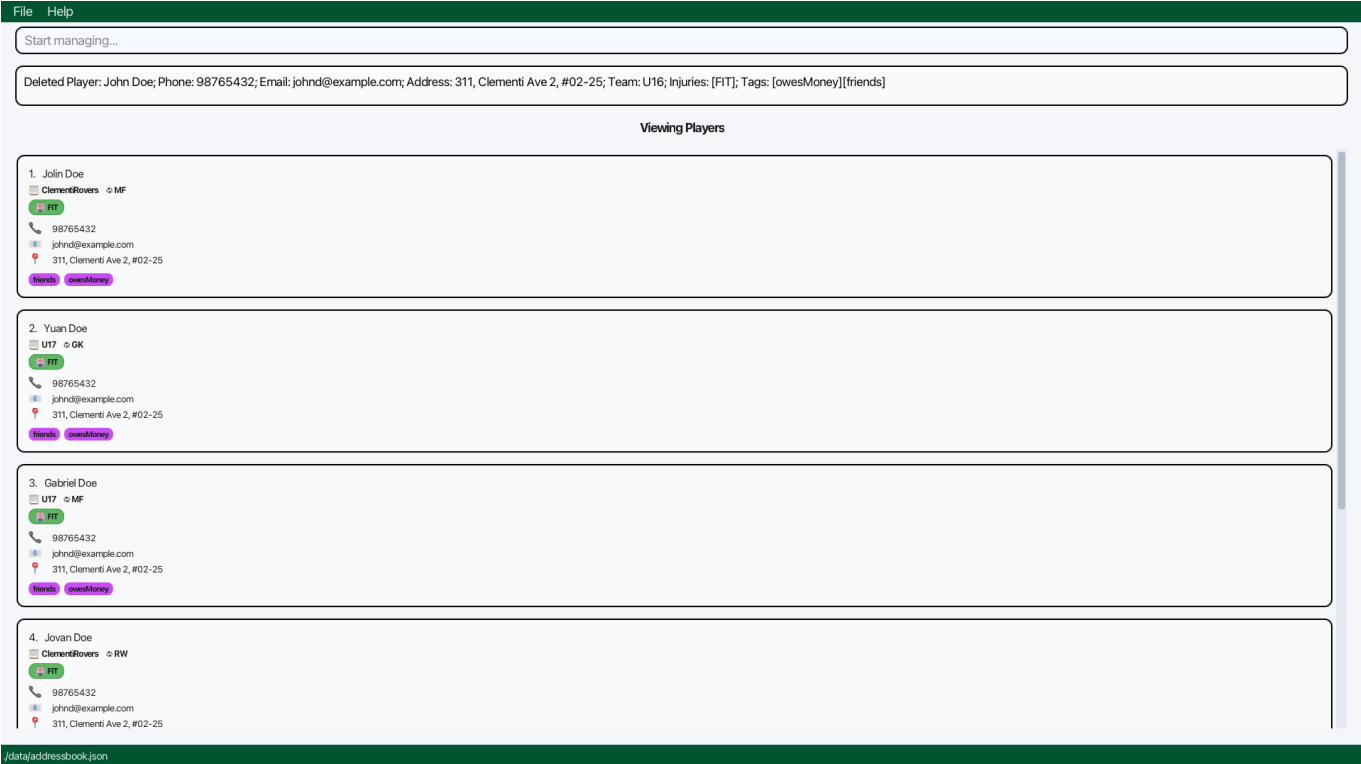
Note: The command can only delete one player at a time.

Examples:

- `delete pl/John Doe` - Permanently deletes John Doe from PlayBook
- `delete pl/Betsy Crowe` - Permanently deletes Betsy Crowe from PlayBook

Expected output: "Deleted Person: John Doe; Phone: 98765432; Email: johnd@example.com; Address: John street, block 123, #01-01; Team: U16; Injury: FIT; Position: LW; Tags: "

The player card will immediately disappear from the player list.



Deleting a team: `deleteteam`

Deletes the specified team from the PlayBook.

Format: `deleteteam tm/TEAM_NAME`

Requirements:

- `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.

Warning: You can only delete a team if there are no players assigned to it. Please remove all players from the team before attempting to delete it.

Note: The command can only delete one team at a time.

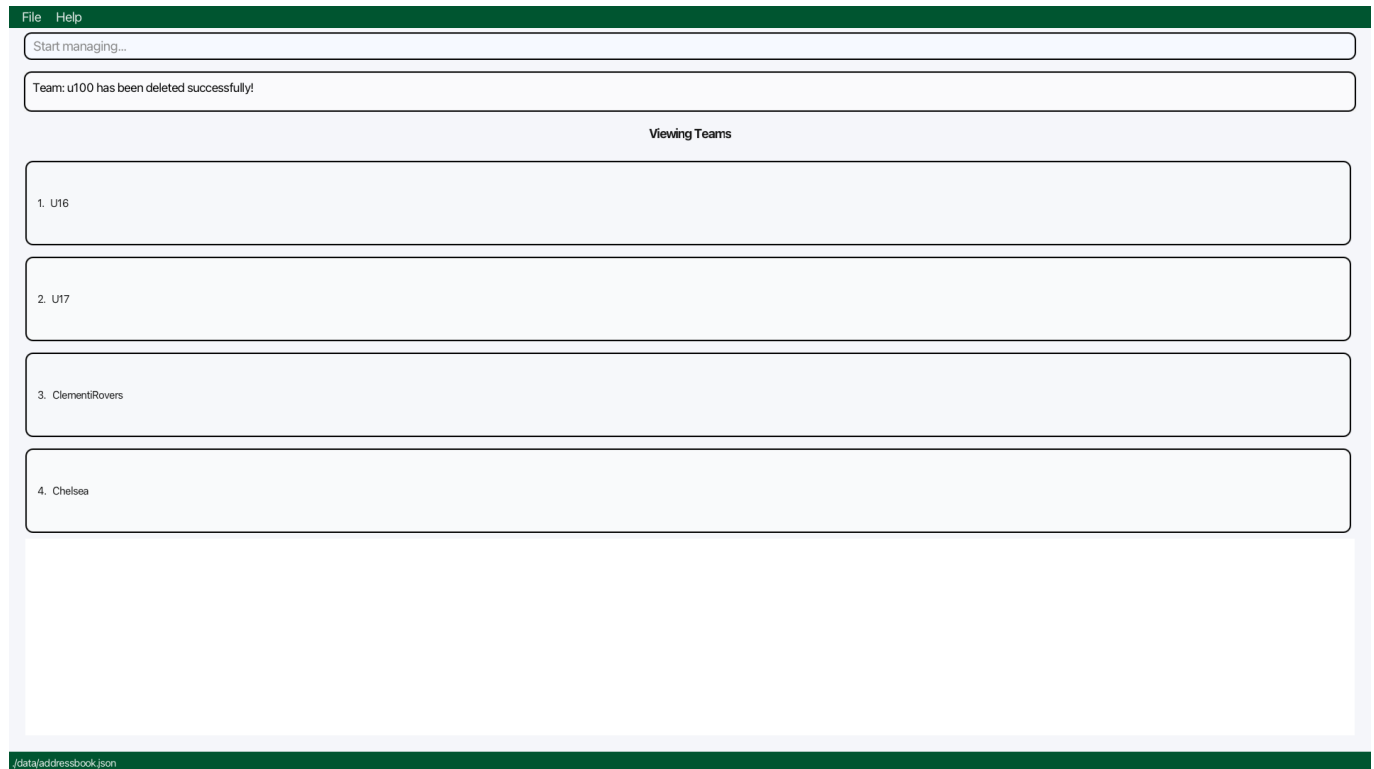
Examples:

- `deleteteam tm/U16` - Deletes the U16 team from PlayBook

- `deleteteam tm/Reserves` - Deletes the Reserves team from PlayBook

Expected output: "Team U100 has been deleted successfully!"

All players previously assigned to this team will have their team assignment removed.



Deleting a position: `deleteposition`

Deletes the specified position from the PlayBook.

Format: `deleteposition ps/POSITION_NAME`

Requirements:

- `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.

Warning: Deleting a position will remove it from all players who were assigned to that position. Make sure this is intentional before proceeding.

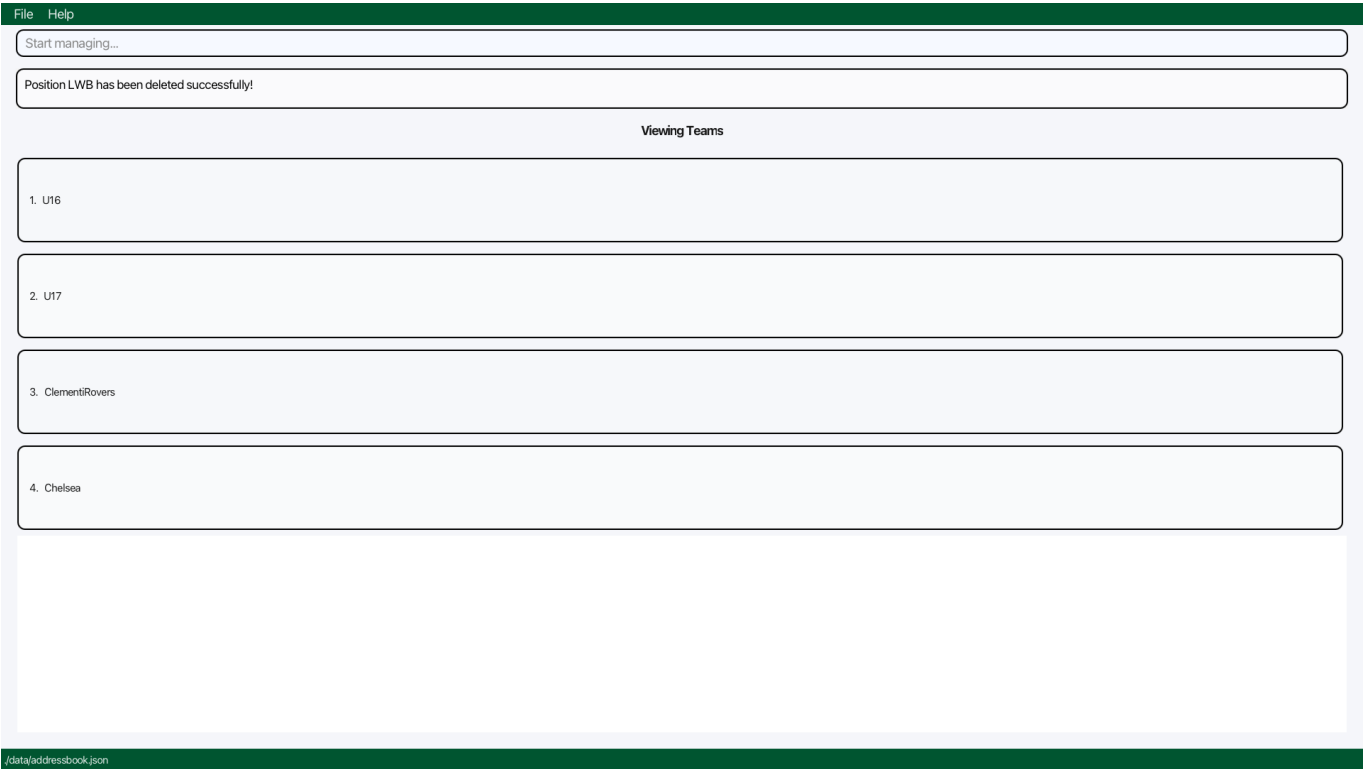
Note: The command can only delete one position at a time.

Examples:

- `deleteposition ps/LW` - Deletes Left Wing position from PlayBook
- `deleteposition ps/ST` - Deletes Striker position from PlayBook

Expected output: "Position LWB has been deleted successfully!"

All players previously assigned to this position will have it removed from their cards.



Clearing all entries: **clear**

Clears all entries from the PlayBook.

Format: **clear**

WARNING: IRREVERSIBLE ACTION!

This command will permanently delete:

- All players and their information
- All teams
- All positions
- All assignments (injuries, captaincy, etc.)

There is no undo. Make sure you have a backup of your data file (located under **/data/addressbook.json**) before using this command!

Tip: Use this command at the start of a new season to begin with a clean slate, but always backup your previous season's data first.

Expected output: "Address book has been cleared!"

The player list panel will be completely empty.

Exiting the program: **exit**

Exits the program.

Format: **exit**

Expected output: The PlayBook application window will close immediately. Your data is safely stored in `addressbook.json`.

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.

Tip: Find your data at `/data/addressbook.json`

Editing the data file

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

Caution:

- If your changes to the data file make its format invalid, PlayBook will discard all data and start with an empty data file at the next run.
- Always make a backup of the file before editing it manually.
- Certain edits can cause PlayBook to behave in unexpected ways (e.g., if a value entered is outside the acceptable range).
- Only edit the data file 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.

Q: Can I manage multiple teams at the same time?

A: Yes! PlayBook is designed for coaches managing multiple teams. Simply create different teams using `addteam` (e.g., `addteam tm/U16`, `addteam tm/U18`) and assign players accordingly.

Q: What happens if I accidentally delete a player?

A: Unfortunately, there's no undo feature. However, since PlayBook automatically saves your data, you can manually restore from a backup of the `addressbook.json` file if you made one. We recommend making regular backups of your data file.

Q: Can a player be in multiple teams?

A: No, each player can only be assigned to one team at a time. If you need to move a player to a different team, use the `assignteam` command.

Q: How do I quickly find all available players for a match?

A: Use the `filter i/FIT` command to show only players with no injuries. You can combine this with team filtering: `filter tm/U16 i/FIT` to see all fit players in the U16 team.

Q: Can I track multiple injuries for one player?

A: Yes! Players can have multiple concurrent injuries. Use `assigninjury` multiple times for different injuries, and use `unassigninjury` to remove specific injuries when they recover.

Q: What if I make a typo in a player's name?

A: Use the `edit` command to correct it. For example: `edit pl/Jon Doe n/John Doe` will rename "Jon Doe" to "John Doe".

Q: Why can't I delete a team?

A: Teams can only be deleted if they have no players assigned. First reassign all players to other teams using `assignteam`, or delete the players, then delete the team using `deleteteam`.

Q: Does PlayBook work offline?

A: Yes! PlayBook works completely offline and stores all data locally on your computer.

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
Add Player	<code>add pl/PLAYER_NAME p/PHONE_NUMBER e/EMAIL a/ADDRESS tm/TEAM_NAME [i/INJURY] [t/TAG]...</code> e.g., <code>add pl/James Ho p/22224444 e/jamesho@example.com a/123, Clementi Rd, 1234665 tm/u16 i/ACL t/friend t/colleague</code>
Add Team	<code>addteam tm/TEAM_NAME</code> e.g., <code>addteam tm/u16</code>
Assign Player to Team	<code>assignteam pl/PLAYER_NAME tm/TEAM_NAME</code> e.g., <code>assignteam pl/John Doe tm/u16</code>
Assign Injury to Player	<code>assigninjury pl/PLAYER_NAME i/INJURY</code> e.g., <code>assigninjury pl/John Doe i/ACL</code>
Unassign Injury from Player	<code>unassigninjury pl/PLAYER_NAME</code> e.g., <code>unassigninjury pl/John Doe</code>
Create New Position	<code>newposition ps/POSITION_NAME</code> e.g., <code>newposition ps/LW</code>

Action	Format, Examples
Assign Position to Player	<code>assignposition pl/PLAYER_NAME ps/POSITION_NAME</code> e.g., <code>assignposition pl/John Doe ps/LW</code>
Clear	<code>clear</code>
Delete Player	<code>delete pl/PLAYER_NAME</code> e.g., <code>delete pl/James Ho</code>
Delete Team	<code>deleteteam tm/TEAM_NAME</code> e.g., <code>deleteteam tm/u16</code>
Delete Position	<code>deleteposition ps/POSITION_NAME</code> e.g., <code>deleteposition ps/LW</code>
Edit	<code>edit pl/PLAYER_NAME [n/NEW_PLAYER_NAME] [p/PHONE] [e/EMAIL] [a/ADDRESS] [tm/TEAM_NAME] [i/INJURY] [t/TAG]...</code> e.g., <code>edit pl/John Doe n/James Lee e/jameslee@example.com</code>
Find	<code>find KEYWORD [MORE_KEYWORDS]</code> e.g., <code>find James Jake</code>
Filter Players	<code>filter [tm/TEAM_NAME] [i/INJURY] [ps/POSITION]</code> e.g., <code>filter tm/U16 i/ACL ps/FW</code>
List	<code>list</code>
Make Captain	<code>makecaptain pl/PLAYER_NAME</code> e.g., <code>makecaptain pl/John Doe</code>
Strip Captain	<code>stripcaptain pl/PLAYER_NAME</code> e.g., <code>stripcaptain pl/John Doe</code>
Filter Captains	<code>filtercaptains</code>
List Teams	<code>listteam</code>
List Positions	<code>listposition</code>
List Injured Players	<code>listinjured</code>
Help	<code>help</code>