Lim Yang Sheng - Project Portfolio

Project: ABC Business Contacts

ABC Business Contacts (ABC) is a desktop Business Contact Management application. The user interacts with it using a CLI, and it has a GUI created with JavaFX. It is written in Java.

Code contributed: [Functional code] [Test code] [Unused code]

Enhancement Added: Backup/RestoreBackup

External behavior

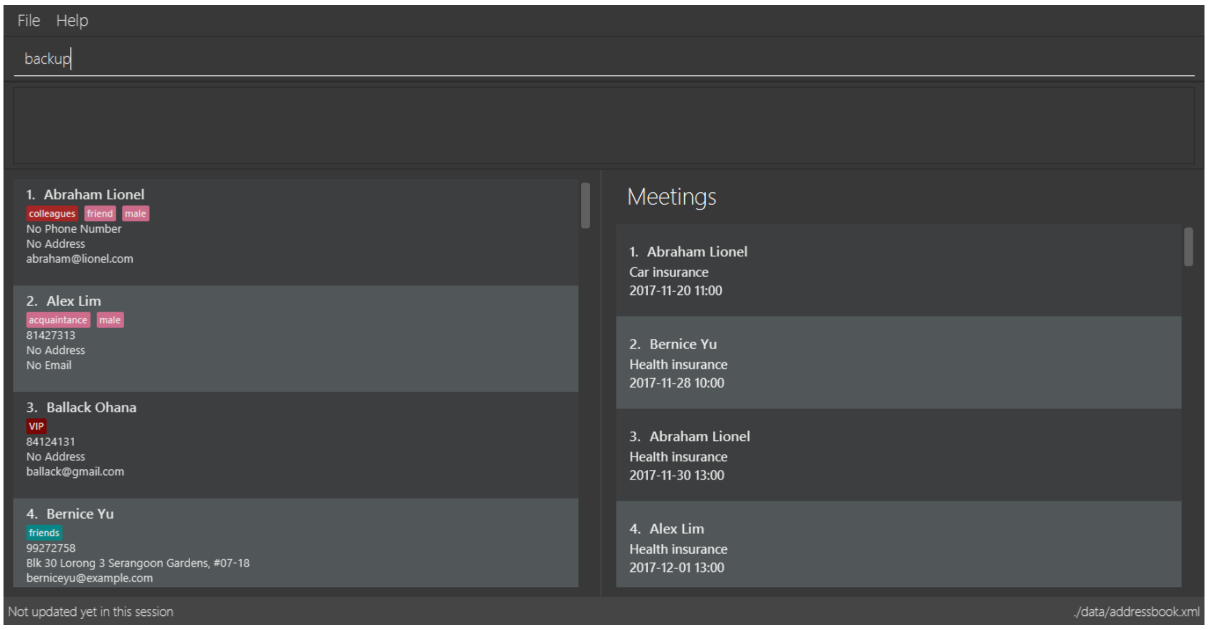
Start of Extract [from: User Guide]

Creating a backup : `backup`

Command Name: `backup`
Shorthand Alias: `b`
Function: Creates a backup file that stores the data in **ABC**
Format: `backup`

NOTE Your data is automatically backed up every time you close **ABC**

If you want to backup your data:

1. Type in
   `>> backup`
   (See Figure 4.15.1)
   

```
backup
```

1. Abraham Lionel
colleagues friend male
No Phone Number
No Address
abraham@lionel.com

2. Alex Lim
acquaintance male
81427313
No Address
No Email

3. Ballack Ohana
VIP
84124131
No Address
ballack@gmail.com

4. Bernice Yu
friends
99272758
Blk 30 Lorong 3 Serangoon Gardens, #07-18
berniceyu@example.com

1. Abraham Lionel
Car insurance
2017-11-20 11:00

2. Bernice Yu
Health insurance
2017-11-28 10:00

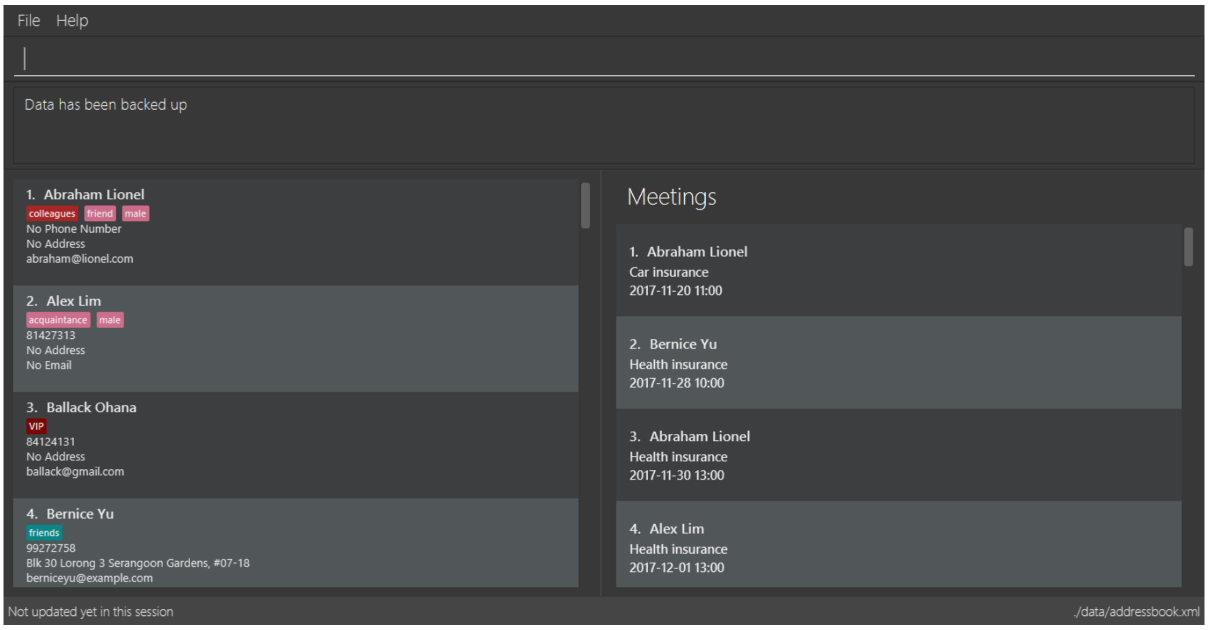
3. Abraham Lionel
Health insurance
2017-11-30 13:00

4. Alex Lim
Health insurance
2017-12-01 13:00

Not updated yet in this session ./data/addressbook.xml

Figure 4.15.1

2. Press `Enter` and you should see a message indicating the successful backup of your data (See Figure 4.15.2)



File Help

Data has been backed up

1. Abraham Lionel
colleagues friend male
No Phone Number
No Address
abraham@lionel.com

2. Alex Lim
acquaintance male
81427313
No Address
No Email

3. Ballack Ohana
VIP
84124131
No Address
ballack@gmail.com

4. Bernice Yu
friends
99272758
Blk 30 Lorong 3 Serangoon Gardens, #07-18
berniceyu@example.com

Meetings

1. Abraham Lionel
Car insurance
2017-11-20 11:00

2. Bernice Yu
Health insurance
2017-11-28 10:00

3. Abraham Lionel
Health insurance
2017-11-30 13:00

4. Alex Lim
Health insurance
2017-12-01 13:00

Not updated yet in this session
./data/addressbook.xml

Figure 4.15.2

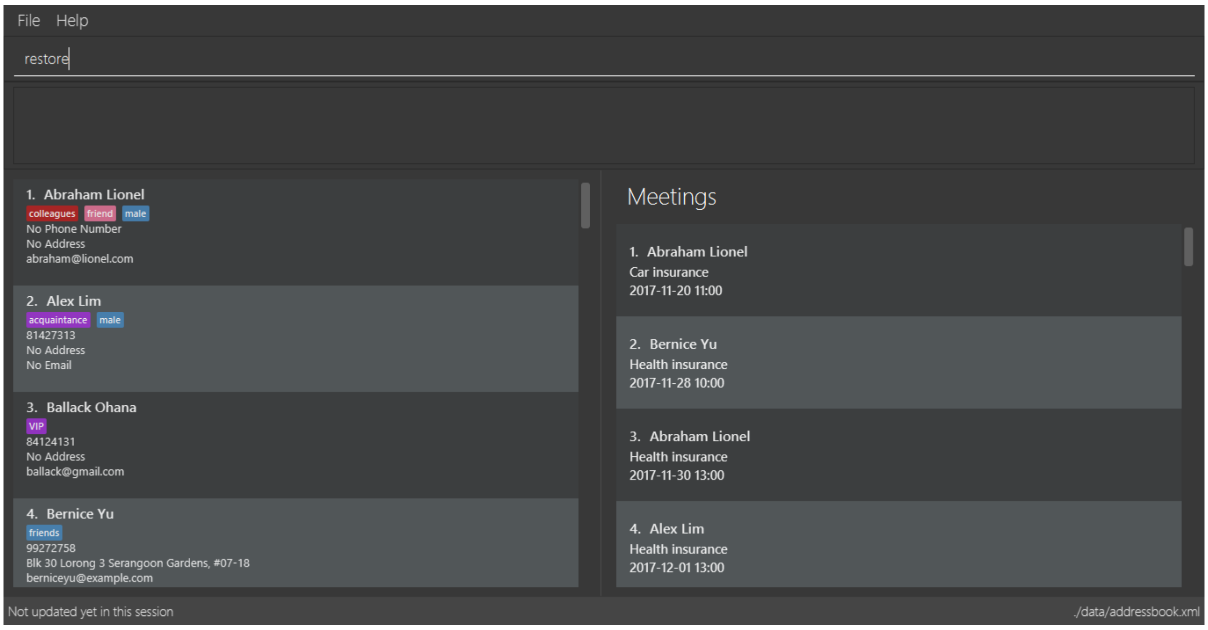
Restoring a backup : `restore`

Command Name: `restore`
Shorthand Alias: `rb`
Function: Retrieves data from a backup file and restore it in **ABC**
Format: `restore`

NOTE There must be a backup file in the default file path for `restore` command to work

If you encounter an unforeseen circumstance and want to revert to a backup:

1. Type in
   `>> restore`
   (See Figure 4.16.1)



```
restore
```

1. Abraham Lionel
colleagues friend male
No Phone Number
No Address
abraham@lionel.com

2. Alex Lim
acquaintance male
81427313
No Address
No Email

3. Ballack Ohana
VIP
84124131
No Address
ballack@gmail.com

4. Bernice Yu
friends
99272758
Blk 30 Lorong 3 Serangoon Gardens, #07-18
berniceyu@example.com

Meetings

1. Abraham Lionel
Car insurance
2017-11-20 11:00

2. Bernice Yu
Health insurance
2017-11-28 10:00

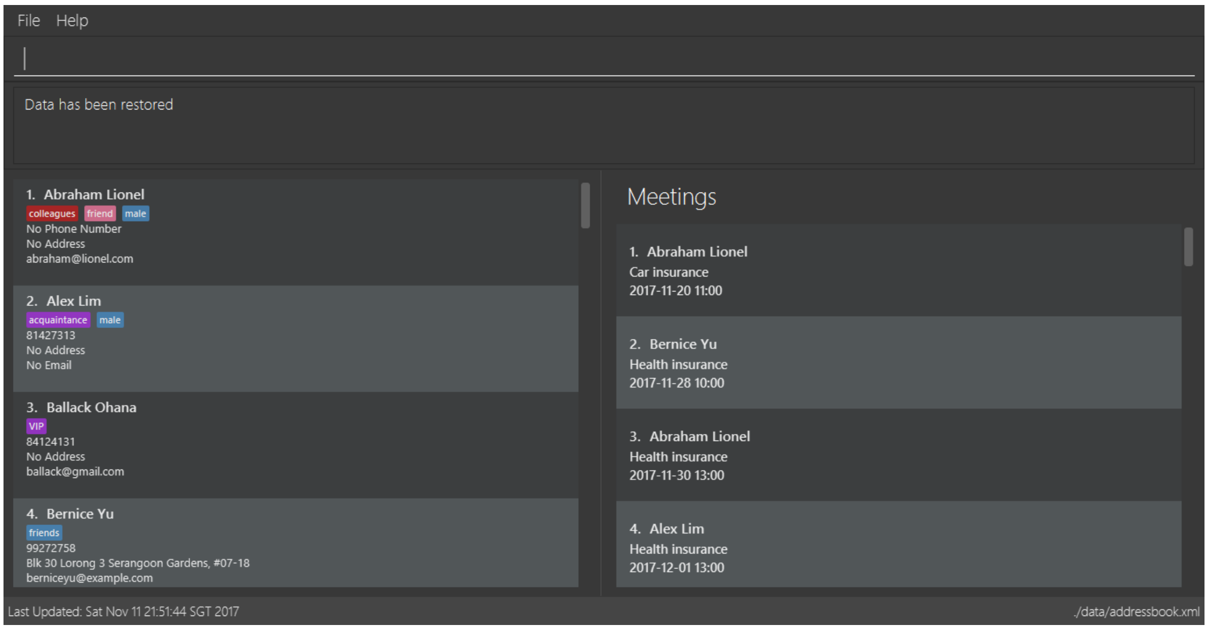
3. Abraham Lionel
Health insurance
2017-11-30 13:00

4. Alex Lim
Health insurance
2017-12-01 13:00

Not updated yet in this session ./data/addressbook.xml

Figure 4.16.1

2. Press `Enter` and you should see that the backup data is restored
(See Figure 4.16.2)



```
Data has been restored
```

1. Abraham Lionel
colleagues friend male
No Phone Number
No Address
abraham@lionel.com

2. Alex Lim
acquaintance male
81427313
No Address
No Email

3. Ballack Ohana
VIP
84124131
No Address
ballack@gmail.com

4. Bernice Yu
friends
99272758
Blk 30 Lorong 3 Serangoon Gardens, #07-18
berniceyu@example.com

Meetings

1. Abraham Lionel
Car insurance
2017-11-20 11:00

2. Bernice Yu
Health insurance
2017-11-28 10:00

3. Abraham Lionel
Health insurance
2017-11-30 13:00

4. Alex Lim
Health insurance
2017-12-01 13:00

Last Updated: Sat Nov 11 21:51:44 SGT 2017 ./data/addressbook.xml

Figure 4.16.2

End of Extract

Justification

Users may lose their original copy of saved data for many reasons. The file may be corrupted or it may have been edited accidentally. As such, the backup and restore backup command will allow users to retrieve back their loss data.

Implementation

Start of Extract [from: Developer Guide]

Backup/Restoring Backup

Mechanism

The backing up of ABC is done by BackupCommand and the restoring of data from a backup file is done by RestoreBackupCommand. BackupCommand inherits from Command as it does not support the undoing and redoing of user actions, whereas RestoreBackupCommand inherits from UndoableCommand. These commands require access to Storage from Logic and this is accomplished by posting an event to EventsCenter. Subscribers in StorageManager will handle these events and respond correspondingly. The sequence diagram below (Figure 3.2.1.1) shows how the BackupCommand is carried out.

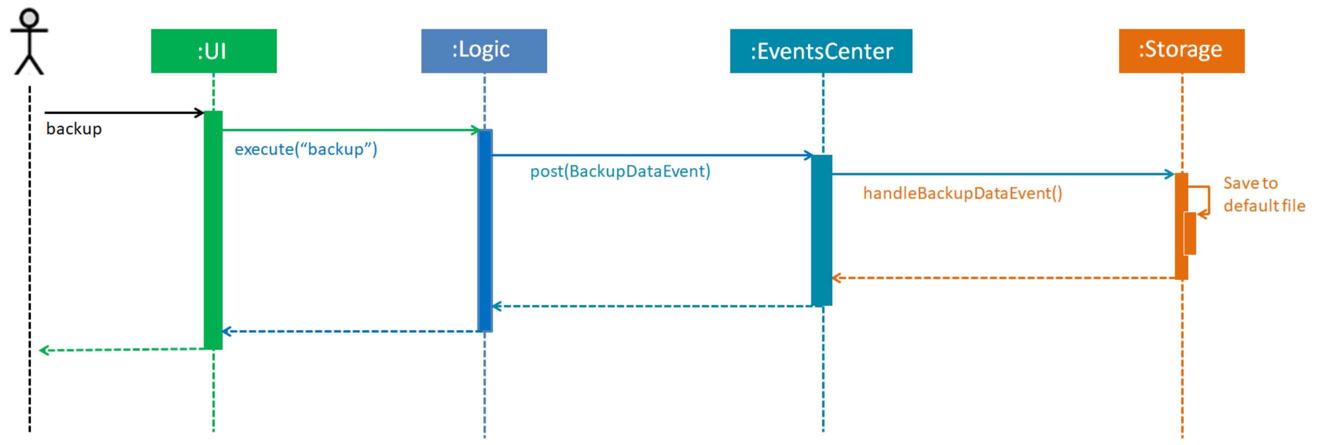


Figure 3.2.1.1 : Backup Command Sequence Diagram

NOTE RestoreBackupCommand shares a similar flow for its sequence diagram.

The BackupCommand is executed when the command backup is entered. The data that is in Model or the active address book is first passed as a parameter to BackupDataEvent. The event will be handled by StorageManager and is saved into the default file path "data/addressbook-backup.xml". The following is the implementation of BackupCommand:

```
public class BackupCommand extends Command {
  //... variables, constructor, other methods...

  @Override
  public CommandResult execute() throws CommandException {
      // reading data from model
      ReadOnlyAddressBook backupAddressBookData = model.getAddressBook();

      // posting event to backup data
      EventsCenter.getInstance().post(new BackupDataEvent(backupAddressBookData));
      return new CommandResult(String.format(MESSAGE_SUCCESS));
  }
}
```

The `RestoreBackupCommand` is executed when the command `restore` is entered. `RestoreBackupDataEvent` is posted and `StorageManager` handles it. The data from default file path "data/addressbook-backup.xml" will be retrieved and it will replace the active address book. The following is the implementation of `RestoreBackupCommand`:

```
public class RestoreBackupCommand extends UndoableCommand {
  //... variables, constructor, other methods...

  @Override
  public CommandResult execute() throws CommandException {
      //... other codes and checks...

      RestoreBackupDataEvent event = new RestoreBackupDataEvent();

      // posting event to help with restoring backup data
      EventsCenter.getInstance().post(event);

      // overwriting the data in active address book
      ReadOnlyAddressBook backupAddressBookData = event.getAddressBookData();
      model.resetData(backupAddressBookData);
      return new CommandResult(String.format(MESSAGE_SUCCESS));

      //... other codes and checks...
  }
}
```

If the backup file does not exist in the default file path, an error message will be shown to the user. This check is done before `RestoreBackupDataEvent` is posted. Once again, this requires `Logic` to access `Storage`. Therefore, a `BackupFilePresentEvent` will be posted and the `Subscriber` in `StorageManager` would handle this event to check if the backup file exists.

NOTE A backup of the data is automatically created when **ABC** is closed.

Design Considerations

Aspect: Accessing `Storage` from `Logic`
Alternative 1 (current choice): Make use of `EventBus` to post events and have `StorageManager` handle the backing up or retrieval of data
Pros: Follow the architecture closely without introducing dependencies between components.
Cons: New `Event` classes have to be created every time a command requires access to data in the storage.
Alternative 2: Allow `Logic` to access `Storage` and its functions
Pros: Easier implementation for current and future functions or commands related to `Storage`.
Cons: Increases coupling between the components.

End of Extract

Enhancement Proposed: Multiple backup files with specified path

User should be able to save different copies of backup data files as they may want to restore data from different sessions. They should be given the choice to specify the file path to save to and the file path to restore data from.

Other contributions

- Replace browser UI with meeting list UI (Pull requests #88, Pull requests #122)
- Updated various commands to work with new meeting list (Pull requests #116, Pull requests #127, Pull requests #132,)
- Discovered bugs during acceptance testing (Issues #59, #60)