Wang Yuchen - UML Diagrams

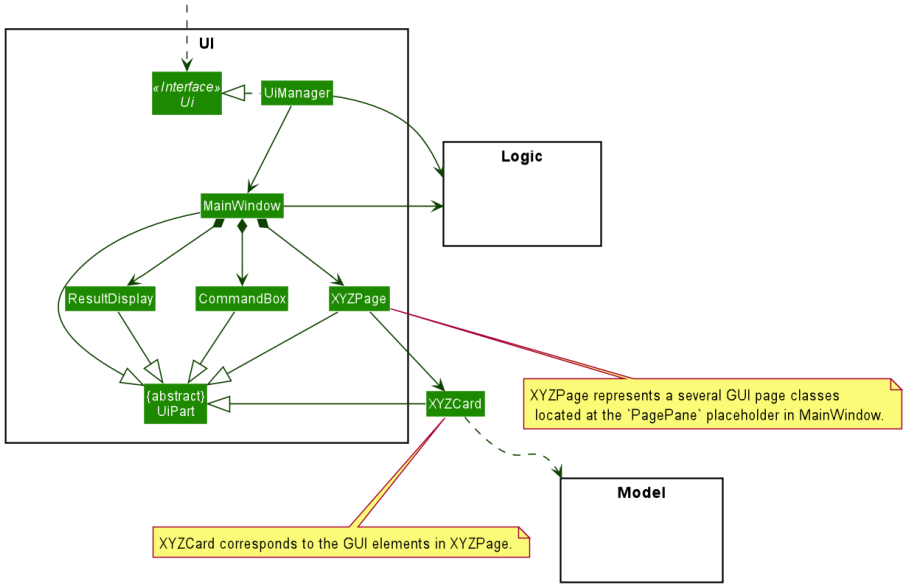

Figure 1. Structure of the UI Component

1. UiClassDiagram depicts FitHelper UI classes, their internal relationship, and the connection with Logic and Model components.
The UI consists of a `MainWindow` that is made up of parts e.g.`CommandBox`, `ResultDisplay`, `PagePane`, `ButtonList`, `CurrentPageTitle` etc. Moreover, it reserves a place for different pages to be displayed.
All these, including the `MainWindow`, inherit from the abstract `UiPart` class.

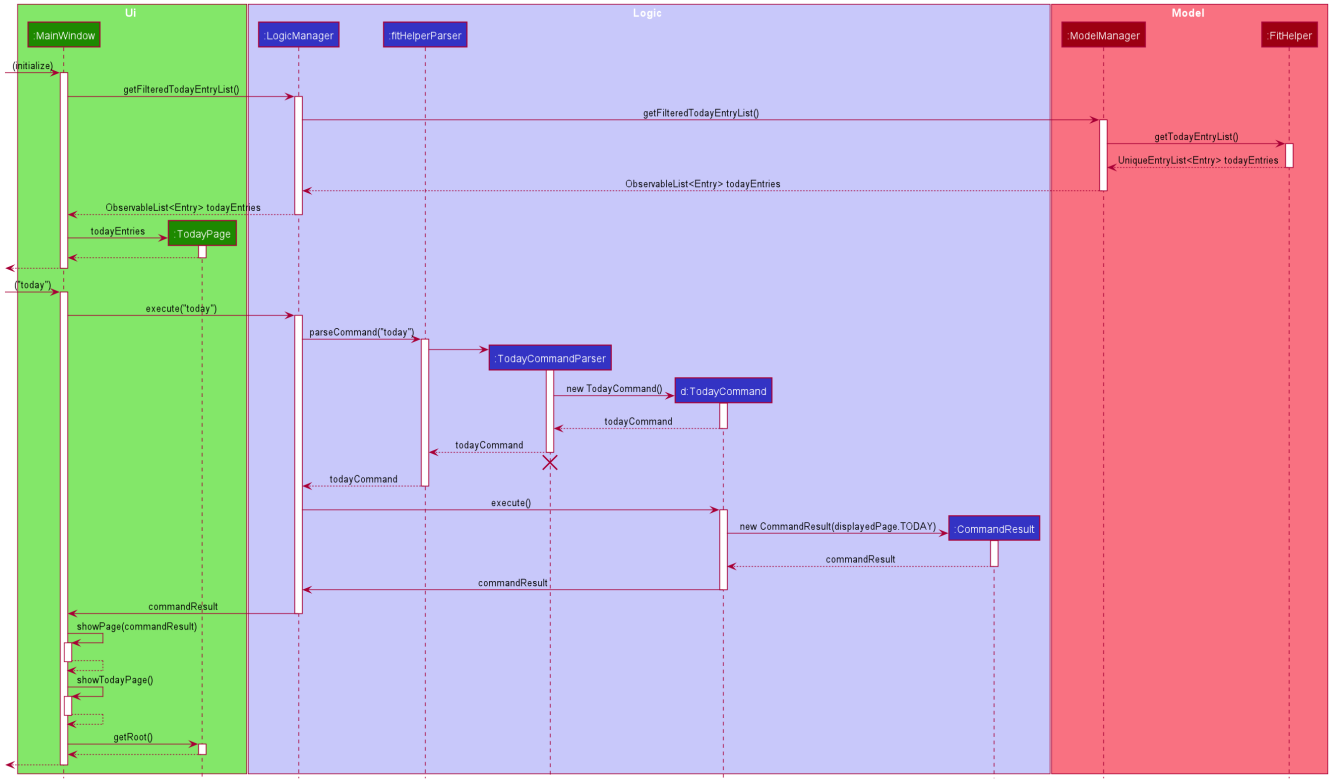

Figure 2. Sequence Diagram for Today Feature

2. InitialTodayPageSequenceDiagram illustrates how components interact with each other when `Today Page` is initialized.
The UI component passes through the Logic component to fetch `entryList` stored in FitHelper Model. `Today Page` is then fed with the data of `todayEntryList` and displays it in GUI list view.

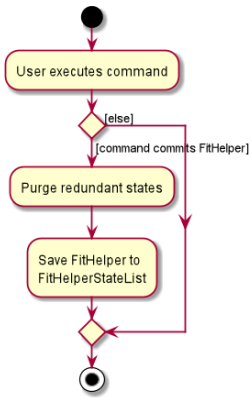

Figure 3. Activity Diagram for FitHelper Commit Feature

3. CommitActivityDiagram summarizes the activities happened when the user executes an `undo` command.
If user's intended command is "undoable", the `currentStatePointer` is moved backward in the `FitHelperStateList`, and the redundant states are purged. Otherwise the command is discarded.

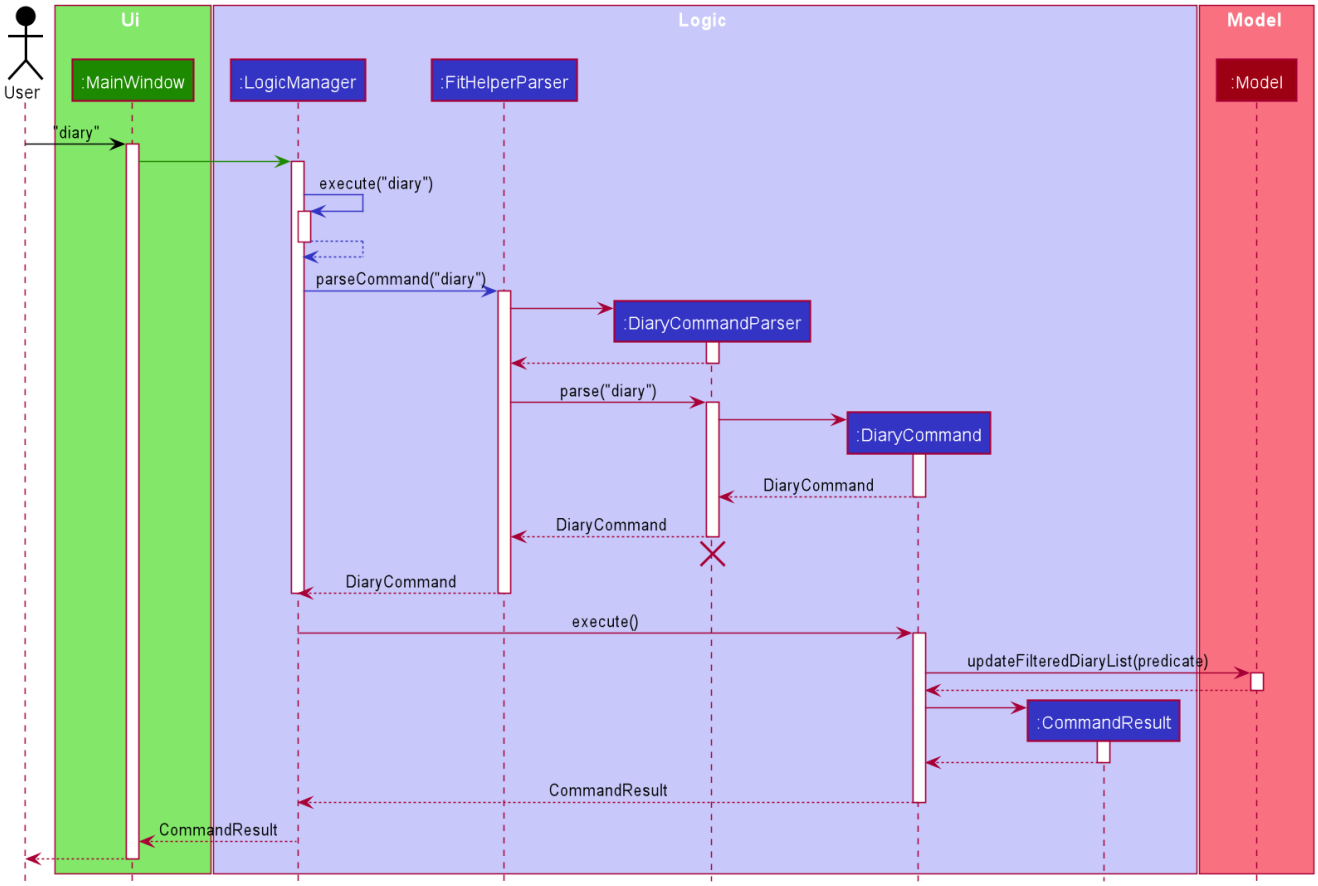

Figure 4. Sequence Diagram for Diary Feature

4. DiarySequenceDiagram illustrates how FitHelper adds a diary to the storage through the interaction between different components.
User input is taken in through UI, handled to LogicManager, converted to an `addDiary` command through the parsers, and finally added to the diary storage.

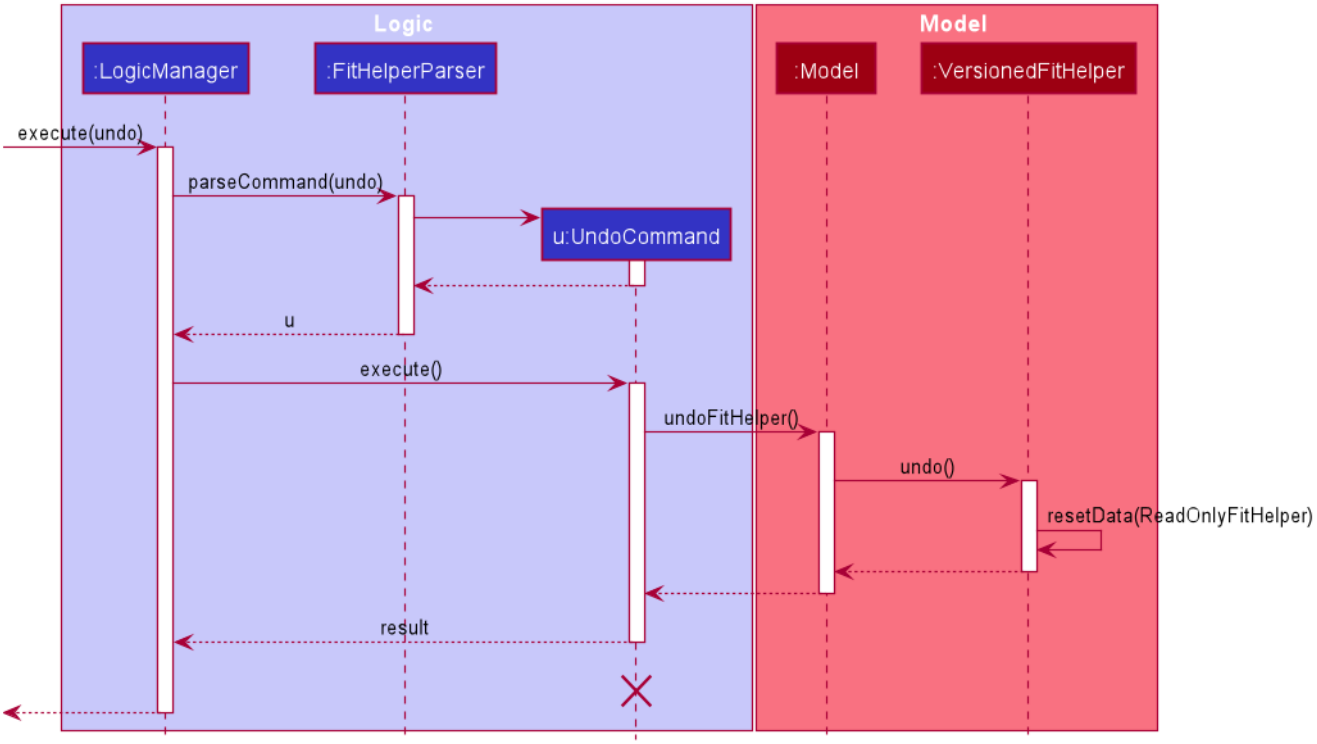

Figure 5. Sequence Diagram for Undo Feature

5. UndoSequenceDiagram depicts the interaction between different components of FitHelper, namely Logic and Model, when an `undo` command is executed. The LogicManager handles the command by converting it to an `UndoCommand` through the `UndoCommandParser`. In command execution, the Model takes in the `undoFitHelper` call from the Logic and rollbacks the current `VersionedFitHelper` to the previous state marked by the `currentStatePointer`. The command result is returned to Logic, and finally reflected and displayed by UI.