Project Manual V0.1

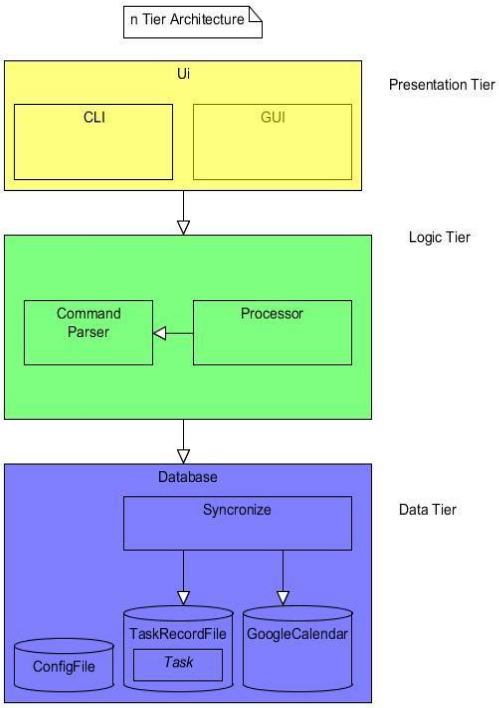
Architecture

My Hot Secretary(MHS) is a desktop client application. Main components are :

- **UI:** The user interface consists of a JTextField to get user input, a JTextArea to display program output and a JPopup to provide feedback as the user types.

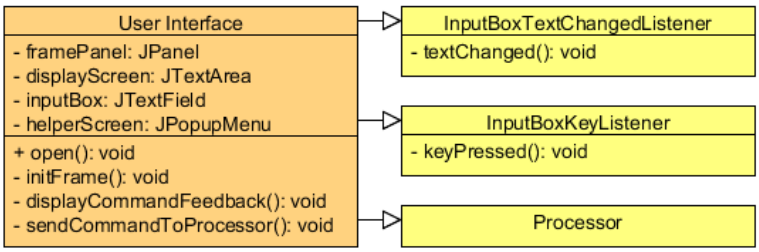
- **Processor:** The Processor handles the flow of logic between the different components of the architecture. It passes the command string over to the command parser to parse the command. It then interacts with the database to execute this command.

- **Database:** Database interfaces persistent the data storage mechanism, on local disk and remote (Google Calendar Service) and handles task queries and operations.



Important APIs

UserInterface



Processor (Logic)

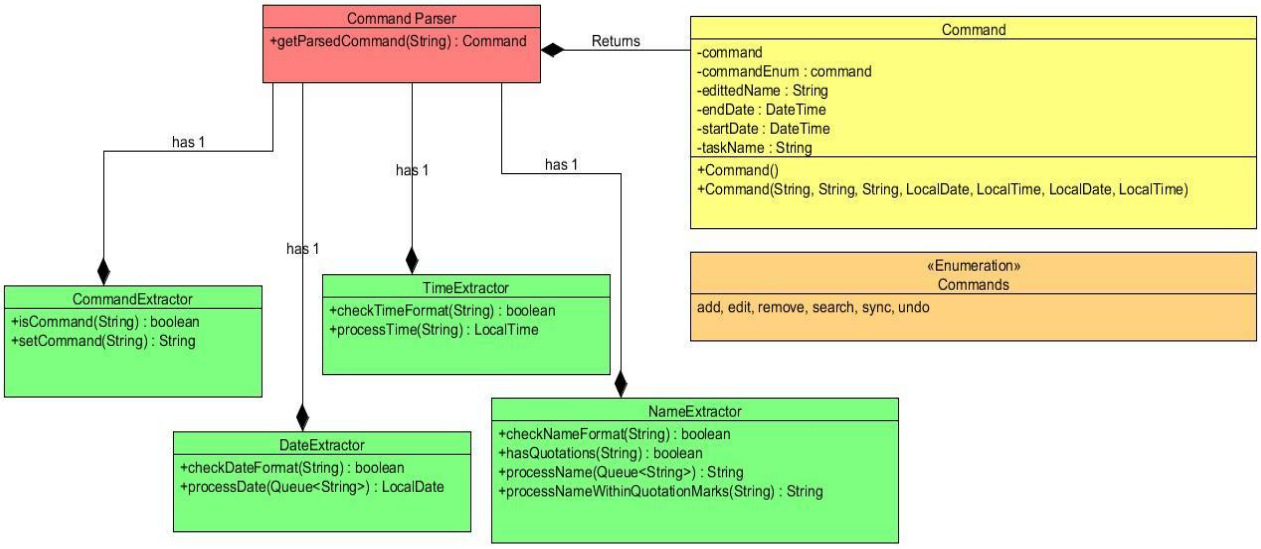
String getCommandFeedback();

String executeCommand(String userCommand){
	Command cmd = cmdProcessor(userCommand);
	process(cmd);
}

Task Processor

add(Task);
update(Task);
display();
delete();
undo();
redo();

Command Parser



To parse a string, the string is split into an array of strings which is analysed individually by each of the extractors.

DateExtractor
processDate takes in a queue filled with all the date strings in a row and returns a LocalDate

TimeExtractor
processTime takes in a time string and returns a LocalTime.

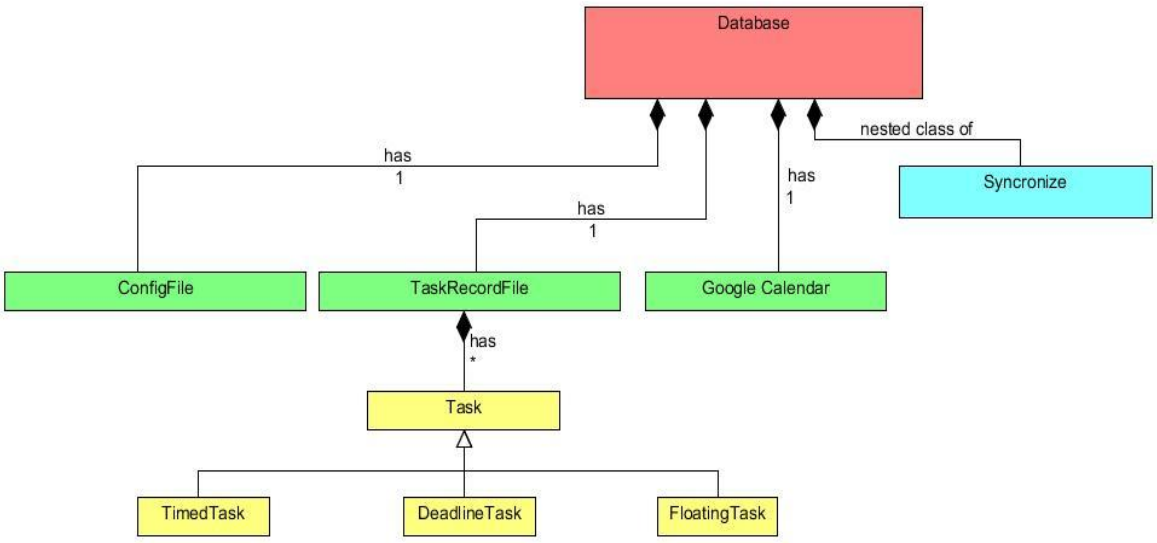
NameExtractor

processName takes a queue filled with all the name formats and appends them together and returns the name.
ProcessNameWithQuotationMarks return the name which is the string within quotation marks.

CommandExtractor
setCommand will set the first string in the string array as the command if it is of a command type, else it is defaulted as add.

Command
Has a constructor that takes in all the parameters that were extracted and sets them.
Has getters to retrieve each individual parameter.

Database



Database public methods returns clones of task(s) that are stored

Adding new task
Task newTask = new <Task Type>(…);
(<Task Type> can be Timed, Floating, Deadline)

add(newTask);

Updating Task

Task existingTask;

To update task of same type
existingTask.*setAttribute(...)* [replace attribute with member variable]

update(existingTask);

<u>To update task type</u>
Task newTaskType = new <TaskType>(existingTask.getTaskId(), ...);
update(newTaskType);

Deleting Task
delete(int taskId)

undelete(int taskId)

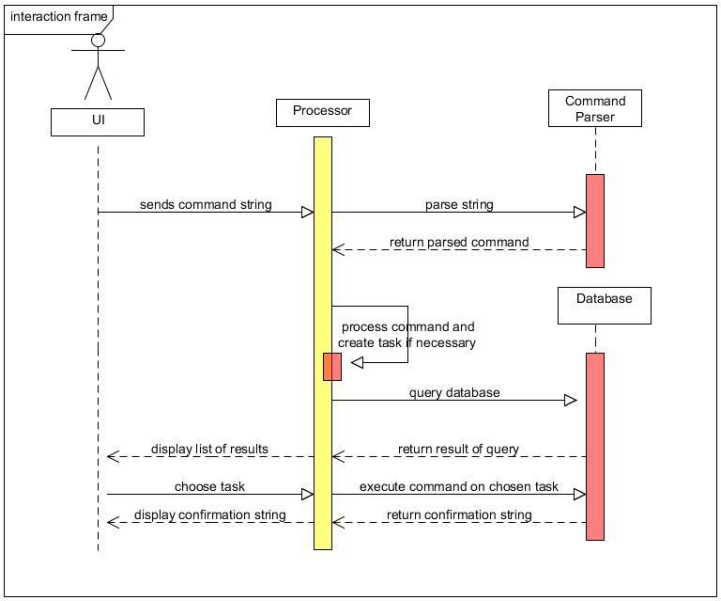
Querying Task(s)
The query method allows multiple parameter filtering of tasks

Synchronization
Synchronization between local storage and Google Calendar is done automatically, calling the syncronize method forces a sync.

Design descriptions

Sequence diagrams

Processor



Instructions for testing

Tests are contained with the Test Package - mhs.test.

1. **Manual Testing**
2. **System Testing**
3. **Unit Testing**
    - Test Suite
        - Individual jUnit tests

Note: DatabaseTest may have failed test cases if anti-virus is enabled since it is dependent on I/O operations, disable anti-virus before running tests

Appendices

MHS API JavaDoc

http://cs2103aug12-t14-2j.googlecode.com/hg/doc/index.html