OOF (Outstanding Organisation Friend) - Developer Guide

By: `Team W17-4` Since: `Aug 2019` Licence: `MIT`

1. Setting up

1.1. Prerequisites

1. **JDK** `11` or above

The `oof.jar` file is compiled using the Java version mentioned above.

2. **IntelliJ** IDE

IntelliJ by default has Gradle and JavaFx plugins installed. Do not disable them. If you have disabled them, go to `File` > `Settings` > `Plugins` to re-enable them.

1.2. Setting up the project in your computer

1. Fork this repo, and clone the fork to your computer
2. Open IntelliJ (if you are not in the welcome screen, click `File` > `Close Project` to close the existing project dialog first)
3. Set up the correct JDK version for Gradle
   a. Click `Configure` > `Project Defaults` > `Project Structure`
   b. Click `New…` and find the directory of the JDK
4. Click `Import Project`
5. Locate the `build.gradle` file and select it. Click `OK`
6. Click `Open as Project`
7. Click `OK` to accept the default settings
8. Open a console and run the command `gradlew processResources` (Mac/Linux: `./gradlew processResources`). It should finish with the `BUILD SUCCESSFUL` message.
   This will generate all resources required by the application and tests.

1.3. Verifying the setup

1. Run `oof.Oof` and try a few commands
2. Run tests to ensure they all pass. *(coming soon)*

2. Design

2.1. Architecture

2.2. UI component

2.3. Logic component

2.4. Model component

2.5. Storage component

2.6. Common classes

3. Implementation

3.1. [Proposed] Undo/Redo feature

3.2. [Proposed] Data Encryption

3.3. Logging

3.4. Configuration

4. Documentation

5. Testing

6. Dev Ops

Appendix A: Product Scope

Target User Profile:

- Has a need to manage multiple tasks at once
- Prefer desktop Command-Line-Interface (CLI) over other types
- Able to type on the keyboard really fast
- Prefers typing over mouse input
- Proficient in using CLI applications

Value proposition: manage contacts faster than a typical mouse/GUI driven app

Appendix B: User Stories

Priorities: High (must have) - * * *, Medium (nice to have) - * *, Low (unlikely to have) - *

S/N	Use Case No	Priority Level	As a …	I can …	So that I can …
01	01	* * *	Computing Student	Add a task	Won't forget the tasks I have to complete
02	02	* * *	Computing Student	Mark a task as complete	Can keep track of what is left to be completed

03	03	* * *	Computing Student	View my tasks in a calendar	Can manage my time properly
04	04	* *	Computing Student	View a summary of tomorrow's task	Will know what to expect for the next day
05	05	* * *	Computing Student	Add an event with the relevant dates, start and end times	Can keep track of my upcoming appointments and examinations
06	06	* * *	Computing Student	Get reminders of deadlines due within 24 hours	Can prioritize those tasks to be completed first
07	07	* * *	Computing Student	Sort my tasks	Can see my tasks in chronological order
08	08	*	Computing Student	Find my tasks	Do not need to scroll through the entire calendar to find certain tasks
09	09	* *	Double degree computing student	Color code the tasks	Can quickly distinguish different type of tasks
10	10	* *	Computing Student	View my tasks for the week	Can plan my time for the week
11	11	* * *	Busy Computing Student	Find free time slots	Will know which dates and times I am free to conduct project meetings
12	12	* * *	Computing Student	Cancel events	Keep my schedule updated
13	13	* * *	Computing Student	Postpone the deadline of tasks	Can properly manage my priorities
14	14	* *	Computing Student who procrastinates	View undone tasks carried forward to the next day in a bright color	Will know what assignments are lagging behind
15	15	* * *	Computing Student	Add a recurring task	Do not have to do it multiple times
16		*	Impatient Computing Student	Quickly type in one-liner commands	Can see the tasks being updated in the program quickly
17		*	Business Analytics Student	View trends for my tasks	Can see if I am lagging behind

18		* *	Paranoid Computing Student	Choose the threshold before the programs sends an alert for me to complete my tasks	Can stay ahead of my schedule
19		*	Organized Computing Student	View all the tasks in a strict format	Will know what to type to enter my tasks
20		*	Computing Student in NUSSU	Export my calendar to a shareable format	Can quickly share my schedule with other people
21		* *	Computing Student	Have a do-after task	Know what tasks need to be done after completing a specific task
22		* * *	Computing Student	Have a task that needs to be done within a time period	Can better plan my schedule
23		*	Computing Student	Add my estimated time taken to complete a task	Know how much free time I would have
24		* *	Undergraduate Tutor	Have two instances of calendar	Can separate my tutor tasks and personal tasks
25		* *	Computing Student	Filter my calendar by different categories	Can view my tasks for that category easier
26		* * *	Computing Student	Add a tentative task	Can confirm it at a later date
27		* * *	Computing Student	View all commands	Do not need to memorise all the commands
28		* * *	Computing Student	Get warnings if an event I add clashes with an existing event	Will not have multiple events at the same time
29		*	Computing Student	Sync my tasks to my phone via bluetooth	Can view my tasks on the go and not just on my laptop
30		**	Computing Student	Print out my tasks stored	Can view my tasks even if my laptop runs out of battery

Appendix C: Use Cases

(MSS refers to Main Success Scenario.)

System: Outstanding Organization Friend (OOF)
Use case: UC01 - Add a task
Actor: User
MSS:

1. User wants to add a task.
2. OOF requests for description of the task.
3. User enters the description of the task.
4. OOF records the task and displays the description.

Use case ends.

Extensions:

- OOF detects empty date and time in description of task.
    - OOF requests for date and time of task.
    - User enters required data.
    - Use case resumes from step 4.
- OOF detects a clash in date and time with another task.
    - OOF warns the User of such a clah by displaying the task(s) that clash(es) and prompts for continuation or cancellation.
    - User decides for continuation or cancellation.
    - OOF requests to confirm decision.
    - User confirms decision.
    - Use case ends if the User decides to cancel the action. Use case resumes from step 4 otherwise.
- At any time, User chooses to re-enter task description.
    - OOF requests confirmation to re-enter task description.
    - User confirms to re-enter task description.
    - Use case resumes from step 3.

System: Outstanding Organization Friend (OOF)
Use case: UC02 - Mark a task as complete
Actor: User
MSS:

1. User wants to mark a task as complete.
2. OOF requests for index of task to mark as complete.
3. User enters the index of the task to mark as complete.
4. OOF records the task completion status and displays the description.

Use case ends.

Extensions:

- OOF detects non-existent index of task.
    - OOF requests for existent index and displays a range of indexes to choose from.

    - User enters required data.
    - Use case resumes from step 4.

System: Outstanding Organization Friend (OOF)
Use case: UC03 - View tasks in calendar
Actor: User
MSS:

1. User wants to view tasks in calendar format.
2. OOF requests for range of index of the tasks the user wishes to view in calendar format.
3. User enters the range of index of the task to view in calendar format.
4. OOF displays the tasks requested in calendar format.

Use case ends.

Extensions:

- OOF detects non-existent index of task in the range.
    - OOF requests for existent index and displays a range of indexes to choose from.
    - User enters required data.
    - Use case resumes from step 4.

System: Outstanding Organization Friend (OOF)
Use case: UC04 - View a summary of the next day's tasks
Actor: User
MSS:

1. User wants to view a summary of the next day's tasks.
2. OOF requests for user input.
3. User enters the summary command.
4. OOF displays the summary of the next day's tasks.

Use case ends.

Extension:

- OOF detects there are no tasks for the next day.
    - OOF prints to the console to warn User that there are no tasks for the next day.
    - Use case resumes from step 4.

System: Outstanding Organization Friend (OOF)
Use case: UC05 - Adding tasks with date and time
Actor: User
MSS:

1. User wants to add a task with date, start and end time.

2. OOF requests for description, date, start and end time of the task.
3. User enters the requested details.
4. OOF records the task and displays the task recorded.

Use case ends.

Extension:

- OOF detects an error with the entered data.
    - OOF requests for the correct data.
    - User enters new data.
    - Steps 3a1-3a2 are repeated until the data entered are correct.
    - Use case resumes from step 4.
- At any time, User choose to stop adding a task.
    - OOF requests to confirm the cancellation.
    - User confirms the cancellation.
    - Use case ends.

System: Outstanding Organization Friend (OOF)
Use case: UC06 - Reminder for expiring tasks (within 24hrs)
Actor: User
MSS:

1. User chooses to activate the reminder for expiring tasks.
2. OOF requests for confirmation of this action.
3. User confirms the action.
4. OOF displays the expiring tasks everytime OOF is started.

Use case ends.

Extensions:

- At any time, User chooses to cancel the activation.
    - OOF requests to confirm the cancellation.
    - User confirms the cancellation.
    - Use case ends.

System: Outstanding Organization Friend (OOF)
Use case: UC07 - Sort tasks in chronological order
Actor: User
MSS:

1. User requests to sort current tasks in chronological order.

2. OOF requests for confirmation of this action.
3. User confirms this request.
4. OOF sorts and displays the tasks in chronological order.

Use case ends.

Extensions:

- OOF detects that there are no tasks to be sorted.
    - OOF warns User that there are no tasks to be sorted
    - Use case ends.
- At any time, User chooses to cancel the request.
    - OOF requests to confirm the cancellation.
    - User confirms the cancellation.
    - Use case ends.

System: Outstanding Organization Friend (OOF)
Use case: UC08 - Find tasks
Actor: User
MSS:

1. User requests to find certain tasks.
2. OOF requests for the description of the tasks.
3. User enters a description of the tasks.
4. OOF displays the tasks that match the description.

Use case ends.

Extensions:

- OOF detects that there are no tasks that match the description given.
    - OOF requests for the User to enter a new description.
    - User enters a new description.
    - Steps 3a1-3a2 are repeated until at least one task matches the description.
    - Use case resumes from step 4.
- At any time, User chooses the stop finding tasks.
    - OOF requests to confirm the request.
    - User confirms the requests.
    - Use case ends.

System: Outstanding Organization Friend (OOF)
Use case: UC09 - Colour code tasks
Actor: User

MSS:

1. User requests to colour code tasks.
2. OOF displays the current tasks present in the program and prompts for the tasks to be colour coded and their respective colours to be coded.
3. User enters the required information.
4. OOF displays the current tasks present after colour coding the selected tasks.

Use case ends.

Extensions:

- OOF detects that there are no tasks to be colour coded.
    - OOF displays the warning that no tasks are available to be colour coded.
    - Use case ends.
- OOF detects an error in the information entered.
    - OOF prompts for User to enter the correct information.
    - User enters the correct information.
    - Steps 3a1-3a2 are repeated until the User enters in the correct information.
    - Use case resumes from step 4.
- At any time, User requests to cancel this action.
    - OOF requests to confirm the cancellation.
    - User confirms the cancellation.
    - Use case ends.

System: Outstanding Organization Friend (OOF)
Use case: UC10 - View tasks for the week
Actor: User
MSS:

1. User requests to view tasks for the week.
2. OOF requests to confirm the request.
3. User confirms the request.
4. OOF displays the tasks for the week.

Use case ends.

Extensions:

- OOF detects that there are no tasks for the week.
    - OOF warns the User that there are no tasks for the week.
    - Use case ends.

- At any time, User chooses to cancel this action.
    - OOF requests for confirmation.
    - User confirms the requests.
    - Use case ends.

System: Outstanding Organization Friend (OOF)
Use case: UC11 - Find free time slots
Actor: User
MSS:

1. User requests to find free time slots.
2. OOF requests for the time period from the User.
3. User enters in the time period of interest.
4. OOF displays the free time slots within the time period.

Use case ends.

Extensions:

- OOF detects that the time period entered is invalid.
    - OOF requests for the User to input a valid time period.
    - User enters a valid time period.
    - Steps 3a1-3a2 are repeated until a valid time period is entered.
    - Use case resumes from step 4.
- At any time, User chooses to cancel the action.
    - OOF requests for confirmation.
    - User confirms the request.
    - Use case ends.

System: Outstanding Organization Friend (OOF)
Use case: UC12 - Delete tasks
Actor: User
MSS:

1. User requests to delete tasks.
2. OOF lists the current tasks saved in the program and prompts User to select the task to be deleted.
3. User chooses the task to be deleted.
4. OOF deletes and display the task that was deleted and the number of tasks saved in the program.

Use case ends.

Extensions:

- OOF detects that there are no tasks saved in the program.
    - OOF warns the User that there are no tasks to be deleted.
    - Use case ends.
- OOF detects an error in the task that was selected by the User.
    - OOF prompts the user to enter a valid input.
    - User enters a valid input.
    - Steps 3a1-3a2 are repeated until the User enters a valid input.
    - Use case resumes from step 4.
- At any time, User chooses to cancel the action.
    - OOF requests for confirmation from the User.
    - User confirms the cancellation.
    - Use case ends.

System: Outstanding Organization Friend (OOF)
Use case: UC13 - Postpone tasks
Actor: User
MSS:

1. User requests to postpone a task.
2. OOF displays the current tasks saved in the program and prompts the User the indicate the task to be postponed and its postponed date.
3. User enters the task and the postponed date.
4. OOF displays the task that was postponed with its new deadline.

Use case ends.

Extensions:

- OOF detects that there are no tasks saved in the program.
    - OOF warns the User that there are no tasks to be postponed.
    - Use case ends.
- OOF detects an error in the task that was selected by the User.
    - OOF prompts the user to enter a valid input.
    - User enters a valid input.
    - Steps 3a1-3a2 are repeated until the User enters a valid input.
    - Use case resumes from step 4.
- At any time, User chooses to cancel the action.
    - OOF requests for confirmation from the User.

    - User confirms the cancellation.
    - Use case ends.

System: Outstanding Organization Friend (OOF)
Use case: UC14 - Overdue tasks
Actor: User
MSS:

1. User requests to highlight tasks that are overdue.
2. OOF requests to confirm the request.
3. User confirms the request.
4. OOF displays the overdue tasks

Use case ends.

Extensions:

- OOF detects that there are no overdue tasks.
    - OOF warns the User that there are no overdue tasks.
    - Use case ends.
- At any time, User chooses to cancel the activation.
    - OOF requests to confirm the cancellation.
    - User confirms the cancellation.
    - Use case ends.

System: Outstanding Organization Friend (OOF)
Use case: UC15 - Recurring tasks
Actor: User
MSS:

1. User chooses to add recurring tasks.
2. OOF displays the current tasks saved in the program and prompts the User to input the task that is recurring and its respective frequency.
3. User enters the task and recurring frequency.
4. OOF displays the task selected and automatically adds the recurring task at relevant time intervals.

Use case ends.

Extensions:

- OOF detects that there are no tasks saved in the program.
    - OOF warns the User that there are no tasks to be marked as recurring.
    - Use case ends.

- OOF detects an error in the task that was selected by the User.
    - OOF prompts the user to enter a valid input.
    - User enters a valid input.
    - Steps 3a1-3a2 are repeated until the User enters a valid input.
    - Use case resumes from step 4.
- At any time, User chooses to cancel the action.
    - OOF requests for confirmation from the User.
    - User confirms the cancellation.
    - Use case ends.

Appendix D: Non Functional Requirements

1. Should work on any mainstream OS as long as it has Java 11 or above installed
2. Should be able to hold up to 200 tasks/events without performance deterioration
3. A user with above average typing speed for regular English Text should be able to store their tasks faster using commands than using the mouse

Appendix E: Glossary

Mainstream OS

Windows, Linux, Unix, OS-X

Appendix F: Instructions for Manual Testing