# Chung-Yen (Philos) Tsai - Project Portfolio

## PROJECT: PrisonBook

## Overview

PrisonBook is a desktop address book application used for Prison Guards/Inmates management. The user interacts with it using a CLI, and it has a GUI created with JavaFX. It is written in Java, and has about 10 kLoC.

## Summary of contributions

- **Major enhancement**: Calendar Scheduling Feature
  - What it does: Allows user to interact with Google Calendar through PrisonBook CLI
  - Justification: Prisons need to follow a strict schedule and guards need to know where inmates are at all times.
  - Credits: Google API Guide
- **Minor enhancement**: Enhanced the find command to be able to search by Tags in addition to Name-search
- **Code contributed**: [Functional code (https://github.com/CS2103JAN2018-T11-B2/main/blob/master/collated/functional/philos22.md)] [Test code (https://github.com/CS2103JAN2018-T11-B2/main/blob/master/collated/test/philos22.md)]
- **Other contributions**:
  - Project management:
    - PR management and tagging up till V1.3
    - PR management and tagging up till v1.3
    - Verified and closed issues for v1.3 Milestone
  - Enhancements to existing features:
    - Find by tags command (Pull request #25 (https://github.com/CS2103JAN2018-T11-B2/main/pull/25))
    - View upcoming events from PrisonBook Google Calendar (Pull request #55 (https://github.com/CS2103JAN2018-T11-B2/main/pull/55))
  - Documentation:

- Converted initial documentation into PrisonBook format.
- Fix overall typos or missed headers.
- Constant updates on User Guide and Developer Guide as new features are rolled out.

# Contributions to the User Guide

*Given below are sections I contributed to the User Guide. They showcase my ability to write documentation targeting end-users.*

## Locating persons by name or tags: `find or f` [1]

Finds persons whose names/tags contain all of the given keywords.
Format: `find n/NAME_KEYWORD [MORE_KEYWORDS] t/TAG_KEYWORD [MORE_KEYWORDS]`

- The search is case insensitive. e.g `hans` will match `Hans`
- The order of the keywords does not matter. e.g. `Hans Bo` will match `Bo Hans`
- Only the name and tags are searched.
- Only full words will be matched e.g. `Han` will not match `Hans`
- Persons matching all keywords will be returned (i.e. `AND` search). e.g. `n/Hans t/family` will return `n/Hans t/family`, not `n/Hans t/enemy` or `n/Gret t/family`

Examples:

- `find n/John`
  Returns `john` and `John Doe`
- `find n/Betsy Tim John`
  Returns any person having names `Betsy`, `Tim`, or `John`
- `find n/Betsy t/Friends`
  Returns any person having names `Betsy` and are tagged `Friends`

## Various commands to interact with Google Calendar

For the first time that you open the calendar, you will be prompted to login to the PrisonBook Google account. The PrisonBook app will attempt to open the browser for you, alternatively you can copy the link and login with the following details: * Google Username: prisonbooksystem | Password: prisonbook2018

### View upcoming events in the calendar: `calendar or cal` [1] (v1.3)

Format: `calendar`

### Add event to the calendar: `calAdd or calA` [3] (v1.4)

Format: `calAdd event/EVENTNAME loc/LOCATION start/STARTDATETIME end/ENDDATETIME`

Examples: * `calAdd event/Software Meeting loc/NUS Utown Green start/2018-06-06 10:00:00 end/2018-06-06 13:00:00`
Adds an event `Software Meeting` at `NUS Utown Green` from `June 6th, 2018 10am` to `June 6th, 2018 1pm`

### Delete event from the calendar: `calDel or calD` [3] (v1.5)

Format: `calDel EVENTNUMBER`

Examples: * `calDel 11`
Deletes the 11th event on the calendar events list

## Contributions to the Developer Guide

> *Given below are sections I contributed to the Developer Guide. They showcase my ability to write technical documentation and the technical depth of my contributions to the project.*

- Chung-Yen (Philos) Tsai (philos22)

  1. Major enhancement:
     Scheduling: Keeps track of schedule of prisoners and guards.

  2. Minor enhancement:
     Find by tag: This function will allow admins to filter user list by tags. Tags may include classification of Prisoner vs. Guards, Prisoner cell blocks and cell numbers, Guards' current teams and assignments, etc…

## Find by Tag Command

### Current Implementation

When the Find command is called, FindCommandParser will look for the tag and/or name prefixes (t/ and n/ respectively). The arguments following each prefix will be made into a list. A mapping is then generated with the prefixes as the keys and the argument-lists as values. The name search portion will run like before, the tag search will follow a similar approach; creating a new predicate instance and return matching results if they exist.

### Design Considerations

### Aspect: Find Command

- **Alternative 1 (current choice):** Add on to the current Find command
  - Pros: Allows filtering by names and/or tags
  - Cons: Requires user to include prefixes in the command to differentiate between name and tag arguments
- **Alternative 2:** Create new command for finding tags
  - Pros: More convenient if user only needs to filter by tags or filter by names separately, no need for prefixes
  - Cons: Cannot filter by names and tags in a single command

### Aspect: Argument Deliminator

- **Alternative 1 (current choice):** Only one prefix with all arguments coming after, delimited by space
  - Pros: Relatively convenient for user when they are inputting multiple arguments
  - Cons: Easier for users to mistakenly merge two arguments into one when forgetting a space
- **Alternative 2:** One prefix required for each argument
  - Pros: Allows user to order name and tag arguments in any order - the tag arguments and name arguments do not have to be grouped together after a single prefix
  - Cons: May become tedious and redundant when inputting multiple arguments, more prone to typos and human error

## Scheduling with Google Calendar

### Current Implementation

The PrisonBook calendar commands will send HTTPS requests to the REST Calendar API, whether it is calling the listEvents, addEvents or delEvents API command. Upon receiving a success confirmation from the API, the change will be reflected next time any calendar commands are called.

Calendar Commands that are supported are: * listing upcoming events * adding an event * removing an event

### Aspect: Choosing Number of Accounts to Work With

- **Alternative 1 (current choice):** Using one Google account for all prison staff
  - Pros: Very simple to administrate and fits the needs of the project scope.
  - Cons: No personalization for calendars of Warden/Guards.
- **Alternative 2:** Use one Google account for Warden and one for Guards
  - Pros: Two accounts is still relatively simple to maintain, allows for some separation of events-scheduling.
  - Cons: All guards still share the same schedule, so they cannot have personal schedules.
- **Alternative 3:** Use one Google account for Warden and one for Guards
  - Pros: Two accounts is still relatively simple to maintain, allows for some separation of events-scheduling.
  - Cons: All guards still share the same schedule, so they cannot have personal schedules.
- **Alternative 4:** Use one Google account for Warden and one for Guards
  - Pros: All prison staff have their own calendar to work with, where they can have private individual events.
  - Cons: Hard to maintain because we have to make sure all staff have the master prison schedule of the inmates' activities non-overlapping with personal events. There will also be a lot of calendars to keep track of, depending on how many guards are hired.

### Aspect: Choosing Number of Calendars per Person

- **Alternative 1 (current choice):** Only one calendar per person
  - Pros: Simple to implement, administrate and satisfies users' daily tasks
  - Cons: Events cannot overlap, limited to fit events in separate rigid time slots.
- **Alternative 2:** Multiple calendars per person
  - Pros: Users can track multiple events happening at the same time. Different calendars can also be used to organize and group logically-related events.
  - Cons: Complicated on the administration side and oversteps the scope of this enhancement. Not really looking to replace personal Google Calendars, this is just to allow prison staff to work with prison schedules.

## View Calendar Feature

### Current Implementation

The View Schedule feature can be accessed via entering the command `cal` . It allows guards/wardens to see upcoming events and their start times.

### Aspect: Event Listing UI

- **Alternative 1 (current choice):** List events as a list format
    - Pros: Simple to implement, shows more details than calendar UI
    - Cons: Not as visually pleasing, users cannot use mouse to edit calendar intuitively
- **Alternative 2:** List events in the calendar UI display
    - Pros: More intuitive with an actual interactive calendar, allows users to click and drag using mouse
    - Cons: No longer command line based, takes up more space than console

# Add Event to Calendar Feature

## Current Implementation

The Add Event feature can be accessed via entering the command `calAdd event/EVENTNAME loc/LOCATION start/STARTDATETIME end/ENDDATETIME`. It allows guards/wardens to add an event, specifying the event name, location, starting and ending times.

## Aspect: Relevancy of Event Attendee List

- **Alternative 1 (current choice):** Excluding Attendee List
    - Pros: Less content for user to input when creating events, does not restrict any functionality in a prison setting
    - Cons: Slightly less customization when adding events
- **Alternative 2:** Including Attendee List
    - Pros: Can invite specific individuals for smaller scale events
    - Cons: Typing each user can become tedious and extra work for the users, also not suitable for prison setting because prisoners are not "invited" for meals or to get back into their cells

# Delete Event from Calendar Feature

## Current Implementation

The Delete Event feature can be accessed via entering the command `calDel EVENTNUMBER`. It allows guards/wardens to remove events from the upcoming events list.

## Aspect: Specification of Which Event to Delete

- **Alternative 1 (current choice):** Enumerating the event list and storing it in an array
    - Pros: Easy interface for users, they just pick a simple integer from the list

- Cons: Have to map these enumerations to the actual IDs

- **Alternative 2:** Displaying the actual event IDs from the API
  - Pros: No additional parsing or eventID storing is needed
  - Cons: Users have to type the event IDs manually, which are seemingly random alphanumeric strings (prone to human error)

- **Alternative 3:** No change to list
  - Pros: No additional clutter to existing list
  - Cons: Users have to specify event name and start/end times, which is quite tedious

# Appendix A: User Stories

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

| Priority | As a … | I want to … | So that I can… |
|---|---|---|---|
| * * * | administrator | group guards together | form teams to carry out various tasks: patrol rotations, perimeter control, cell guards |
| * * * | guard | have a track record of each prisoner's past offences Type 1 – Injuring guard Type 2 – Infighting with other inmates Type 3 – Vandalism | use the appropriate level of precaution when dealing with unruly individuals |

| Priority | As a … | I want to … | So that I can… |
|---|---|---|---|
| * * * | guard | keep track of each prisoner's holding cells | ensure that there are two prisoners per cell and that certain bad prisoner combinations are avoided |
| * * * | prison warden | assign ranks to each guard | enforce access rights to view/edit prisoner and officer data |
| * * * | guard | check the length of imprisonment of prisoners I am in charge of | know when they are leaving |
| * * * | guard | add prisoners to prison cells | the prisoners have a prison cell to stay in |
| * * * | guard | check number of prison cells available | decide whether to take more prisoners in |
| * * * | guard | delete a prisoner | add more prisoners to prison |
| * * * | guard | check when I am on duty | see when I have to come to work |

| Priority | As a … | I want to … | So that I can… |
|---|---|---|---|
| * * * | guard | check who are in my team | coordinate with my team |
| * * * | prison warden | add police officers | give access to new police officers |
| * * | guard | assign prisoners to different recess blocks | separate those that start fights or cause trouble when together |
| * * | prison warden | shuffle patrol guards around teams every day | each team is made up of different members every day, prisoners will acquaint themselves with specific guards |
| * * | prison warden | divide cells into sections (A, B, C, D) and name each cell numerically (A1, A2, etc…) | refer to each cell easily when assigning prisoners to them |
| * * | guard | check whether there are visitor appointments | inform prisoners and bring them to see their visitors |

| Priority | As a … | I want to … | So that I can… |
|---|---|---|---|
| * * | guard | edit details of prisoners | update information of prisoners |
| * * | guard | transfer people to other prisons | have more empty cells for more prisoners |
| * | prisoner | know my release date | look forward to the day I can see my family/friends again |
| * | guard | see my assigned prisoners' social networks: friends, enemies, family, etc… | use this as reference when dealing with inmates |
| * | guard | assign prisoners to solitary confinement | punish those that have continuously committed serious offences |
| * | cook | check dietary requirements of all people in prison | cook sufficient food |
| * | counselor | check what prisoners' crimes | understand my patients better |

# Appendix B: Non Functional Requirements

1. Should work on any mainstream OS as long as it has Java `1.8.0_60` or higher installed.
2. Should be able to hold up to 1000 persons without a noticeable sluggishness in performance for typical usage.
3. A user with above average typing speed for regular English text (i.e. not code, not system admin commands) should be able to accomplish most of the tasks faster using commands than using the mouse.
4. Should work for 32-bit and 64-bit environments
5. System should respond within 2 seconds
6. Should have different levels of access for users of different ranks.
7. Should have a backup copy at all times in case a user accidentally deletes any information.

Last updated 2018-04-11 11:50:27 UTC