

# Maintenance Phase 1

## Trello bugs and description

### Bug 1

The screenshot shows a Trello card titled "bug 1" in the "To Do" list. The card has the following details:

- Notifications:** Watch
- Description:**
  - Issue:** The clone method in the Card class creates a copy of a Card object using *shallow cloning*, which means that while the Card object itself is cloned, its CardSide objects (`m_frontSide` and `m_backSide`) are not deeply cloned. This can lead to unexpected behavior if the Card object's CardSide objects are modified after cloning, as the original and cloned Card objects will share references to the same CardSide objects.
  - Example:** If you clone a Card object and then modify the front side of the cloned object using `setSides`, it will also affect the front side of the original object because they both reference the same CardSide object.
  - Solution:** To avoid this issue, instead of using the clone method, consider using a copy constructor or a copy factory method to create a deep copy of the Card object. This ensures that the Card object and its CardSide objects are fully duplicated, preventing unintended side effects from shared references.
- Activity:** Write a comment...

The right sidebar contains various collaboration and management options:

- Suggested: Join
- Add to card: Members, Labels, Checklist, Dates, Attachment, Cover, Custom Fields
- Power-Ups: + Add Power-Ups
- Automation: + Add button
- Actions: Move, Copy, Make template, Archive, Share

## Bug 2

The screenshot shows a Trello card titled "bug 2" in the "To Do" list. The card has the following details:

- Notifications:** Watch
- Description:**

**Issue:**  
A potential logical bug exists in the `isLearned` method of the Card class. This method determines if a card is considered learned based on its expiration date. However, the logic may be incorrect if the expiration date is set in the future. If `m_dateExpired` is set to a future date, the `isLearned` method will return false, indicating that the card is not learned, even though it might be considered learned in practice. This could lead to incorrect behavior, such as not displaying cards that should be considered learned.

**Example:**  
If a card's `m_dateExpired` is set to a future date, the `isLearned` method will incorrectly return false, even though the card should be considered learned.

**Solution:**  
To fix this potential bug, adjust the logic in the `isLearned` method to correctly handle cases where the expiration date is set in the future. Check if `m_dateExpired` is not null (indicating that the card has an expiration date) and if the expiration date is not before the current date (`(m_dateExpired.before(now))`). This ensures that if the expiration date is in the future or equal to the current date, the method will return true, indicating that the card is learned.
- Activity:** Write a comment...

The right sidebar contains various actions and sharing options:

- Suggested: Join, Add to card, Members, Labels, Checklist, Dates, Attachment, Cover, Custom Fields.
- Power-Ups: + Add Power-Ups.
- Automation: + Add button.
- Actions: Move, Copy, Make template, Archive, Share.

Collabs section lists users: Omar farrag, Abdelrahman Moataz, Doha Sherif, Omar Bolok, Abdelrahman Wael.

## Bug 3

The screenshot shows a Trello card titled "bug 3" in the "To Do" list. The card's description is as follows:

**Issue:**  
A potential logical bug exists in the setDateModified method of the Card class. The method checks if the new modification date is before the creation date and throws an IllegalArgumentException if it is. However, this check could be incorrect if the intention is to ensure that the modification date is always after or equal to the creation date.

**Example:**  
If the modification date is set to a date before the creation date, the setDateModified method will throw an IllegalArgumentException, even though the intention might be to allow the modification date to be equal to the creation date.

**Solution:**  
To fix this potential bug, adjust the logic in the setDateModified method to ensure that the modification date is after or equal to the creation date. Change the condition from `if (date.before(m_dateCreated))` to `if (date.before(m_dateCreated) || date.equals(m_dateCreated))`. This change allows the modification date to be equal to the creation date, ensuring that the method behaves as intended.

**Activity:** [OB] Write a comment...

**Collabs:** Omar farrag, Abdelrahman Mostaz, Doha Sherif, Omar Bolok, Abdelrahman Wael

**Actions:** Move, Copy, Make template, Archive, Share

## Bug 4

The screenshot shows a Trello card titled "bug 4" in the "To Do" list. The card has the following details:

- Description:** CardSide class
- Issue:** The core issue within the `CardSide` class is the incorrect use of `List.equals()` to perform list comparisons. This will lead to unexpected behavior in the `setImage` method because list equality in Java is based on reference equality (whether they're the same object in memory) rather than value equality (whether they have the same elements).
- Solution:**
  - Implement a helper method: Create a method to compare the contents of lists, ignoring object references:
  - like this code

```
1 private boolean imageListsEqual(List<String> list1, List<String> list2) {
```
- Activity:** There is no activity listed.
- Notifications:** Watch
- Collaborators:** Omar farrag, Abdelrahman Moataz, Doha Sherif, Omar Bolok, Abdelrahman Wael
- Power-Ups:** + Add Power-Ups
- Automation:** + Add button
- Actions:** Move, Copy, Make template, Archive, Share

# Implement jgoodies library

The screenshot shows a Trello board titled "jmemorize-maintenance Free". The board has a single column named "Phase 1" containing several cards. One card is selected, titled "Implement jgoodies library" in the "To Do" list. The card has a "Labels" section with a red label and a "Notifications" section with a "Watch" button. The "Description" section contains the following text:

JGoodies is a library that is imported but not implemented in the project  
JGoodies Forms is a Java library that simplifies the creation of sophisticated, consistent, and easy-to-use Swing user interfaces. It provides a simple API for laying out complex forms, which is a common task in desktop application development.

In the provided code, JGoodies Forms is used to create the layout of the preferences dialog. Here's how it works in the code:

1. **FormLayout**: Defines the layout of the form using a simple and flexible column and row specification.
2. **CellConstraints**: Specifies the location and span of components within the form.
3. **DefaultFormBuilder**: Helps to build the form step by step, adding components and separators.

JGoodies Forms makes it easier to create complex forms by simplifying the layout process and ensuring that the forms are aesthetically pleasing and consistent.

The card also has an "Activity" section with a comment placeholder "Write a comment..." and a "Show details" button. On the right side of the card, there is a sidebar with various options: Suggested (Join), Add to card (Members, Labels, Checklist, Dates, Attachment, Cover, Custom Fields), Power-Ups (+ Add Power-Ups), Automation (+ Add button), and Actions (Move, Copy, Make template, Archive, Share).

## Add CSVwriter and CSVReader Library

