

# **Bug Reports**

*a.k.a code smells*

***Include a document explaining the code smells or design issues you resolved with your refactoring. Describe what the code/design looked like before, what refactoring you applied, and how does it look like now.***

3) Postmortem analysis (5 minutes) - As a postmortem analysis, discuss some of the issues that arose during the development of your product. This will be informal; in that you can discuss any of a number of issues:

• What was the overall architecture of your system (particularly if it is different from the example system)?

- This architecture involves 5 modules:

1. One for managing accounts
2. One for chores and their utilities
3. One for storing our database setup and operations.
4. One for all the User Interfaces involving registration, login, competition, etc.
5. One for all the Integrated test cases, JUnit test cases

• How large is the project (number of methods, classes, etc.)? How much of this is (roughly) devoted to each major system component? And any other quantifiables (e.g. if you have a record of hours spent on tasks).

- We had a total of 28 classes (22 src classes and 6 Junit test classes). The code has around 340,000 lines of code.

• What went right in the development process?

- We were able to effectively implement the competition feature within the GUI itself. Originally we had started with the competition module however we realized that we weren't properly able to utilize this feature as we needed to make it so that we assign 1 chore to multiple children. In this case we need direct access to the parent and child accounts using the database. For this reason we implemented this feature in the GUI
- Chore History in the child account GUI took a surprisingly long time to implement due to the color coordination required.
- The Parent is able to complete all the tasks they need such as creating chores, assigning chores and linking their children.
- The child is able to mark their chores complete or not complete, they can use chore history to view their past chores and view the status of them, and they can also filter their chores based on various method such as payment, time, alphabetical, etc.

- Child and Parent Registrations and logins also went right in this development project
  
- What went wrong in the development process?
  - Sometimes the database would be updated but not dropped by other users which would cause issues. We would have to
  
- What would you do differently, if you had the chance to start over?
  - A more structured time frame for people to complete their assigned tasks.
  
- What took the most time? The least? Any surprises?
  - The thing that took the most time was implementing and setting up the database, the least time was the chore count feature since we had most of the information at hand for this implementation/user story. Chore history also took a considerable amount of time and effort since it required us fetching the values from the database to display the information to the user. Some of the surprises we ran into was how many times we would have to drop and rebuild the database since most of our features would be linked to databases and rely on stored information.
  
- Are there any particular design smells, or brilliant design decisions?
  -
  
- Are there any outstanding bugs?
  - One of the code smells is that the milestone feature doesn't work.
  
- Did any features work better than expected?
  - We did not expect chore history to work efficiently due to the color rendering issue of cells that we faced at the beginning however after last minute changes we were able to complete the feature by assigning red(completed after deadline), green(completed within deadline), orange(completed by a sibling), white(not completed at all). This eventually ended up working very well after we solved these rendering issues.
  - Another feature was the exporting to csv file. This feature was another feature that we found working very well. WE initially didn't know how to implement this feature so after constant trials we were able to produce this function which essentially converts all the chore details in an excel sheet.
  
- Are you using any technologies other than what was required?
  - Nothing else

- How did the project change from your initial (iteration 0) vision or stories, or did it work out as predicted?

- Initially we had our first 9 features from iteration 0 up till iteration 1, however since by iteration 3 we needed 3 features each and as there's 6 of us we needed 9 more features to implement, however as we were developing this, we came up with an additional feature making it 10. As a result, we set off by creating and implementing the new 10 features for this new iteration 3. These new features are:

1. Chore Count

2. Chore Priority

3. Parent GUI Chore Removal/Assignment

4. Chore Presets Option when creating the chore

5. Chore History

6. Chore Description

7. Chore Deadline

8. Payment Reminder for Completed chores

9. Chore Ratings

10. Chore Full Details View

We have ended up managing to execute all these additional features that we have added which allows our application to run as effectively as possible. It took a very long time however through the help of research and multiple meetings with the rest of our team mates, we were able to collaboratively complete and fix any bugs or issues with our features while implementing them.

- What did you learn about team or large project development? What will you start doing, keep doing, or stop doing next time?

- We learnt that communication is key. Communication plays a huge role in team dynamics and is a foundation of the team's success. Next time we learnt we should focus more on communication and host weekly meetings to catch up with the work every member has been doing. Something we will keep doing is planning ahead, we made plans and would plan ahead for future visions of the project within a planned time frame. Something we will stop doing next time is sticking on one task and sticking with it and trying not to make too many compromises as this also throws off the team dynamic. Perhaps spending more time on future implementations would avoid the risk of having to scratch tasks all together.

- Can you draw any conclusions from what you've done?
  - You should include both negative and positive points; they will not count against your project mark. - It is highly recommended that you include the last three items from the list above. - Also ensure that you discuss the area you can identify for improvement after your iteration 3. - Make sure your presentation has a recognizable conclusion; don't just trail off at the end.

Bug 1:

<b>Problem report number</b>	2024-03-21-001
<b>Reported by</b>	Vrushank Vaghani (tester)
<b>Date reported</b>	March 21, 2024
<b>Program (or component) name</b>	Registration of Parent and Child
<b>Release number</b>	1.0
<b>Version (build) identifier</b>	1.0.0
<b>Configuration(s)</b>	Database must be set up.
<b>Report type</b>	Functional issue
<b>Can reproduce</b>	Yes
<b>Severity</b>	High
<b>Priority</b>	High

<b>Problem summary</b>	Registration is successful even if no usernames and passwords are entered.
<b>Key words</b>	Registration, Authentication, Bug, Null, Empty, Validation
<b>Problem description and how to reproduce it</b>	<ul style="list-style-type: none"> <li>· Navigate to the registration page of either parent or child of ParentChoreSystem.</li> <li>· Leave the username and password fields blank.</li> <li>· Click on the "Register" button.</li> <li>· Notice that the registration process is completed successfully without entering any credentials. - Then login with blank fields and notice that login is also successful.</li> </ul>
<b>Suggested fix</b>	<p>Implement validation checks on the registration form to ensure that both username and password fields are mandatory before allowing registration to proceed.</p> <p>Display an error message prompting the user to fill in both fields if either or both are left empty.</p>
<b>Status</b>	Closed
<b>Resolution</b>	Authenticated username and password by cross-referencing from the database.
<b>Resolved by</b>	Vrushank Vaghani (developer)

Bug 2:

<b>Problem report number</b>	2024-03-21-002
<b>Reported by</b>	Vrushank Vaghani (tester)
<b>Date reported</b>	March 21, 2024
<b>Program (or component) name</b>	Registration of Parent and Child
<b>Release number</b>	1.0
<b>Version (build) identifier</b>	1.0.0
<b>Configuration(s)</b>	Database must be set up.
<b>Report type</b>	Functional issue
<b>Can reproduce</b>	Yes
<b>Severity</b>	Medium
<b>Priority</b>	High
<b>Problem summary</b>	Error message not displayed to the user when registering with an existing username.
<b>Key words</b>	Registration, Authentication, Bug, Username, Error, Backend

<b>Problem description and how to reproduce it</b>	<ul style="list-style-type: none"> <li>Navigate to the registration page of either parent or child of ParentChoreSystem.</li> <li>Enter a username that already exists in the system.</li> <li>Complete other registration details and click on the "Register" button.</li> <li>Notice that the registration process does not proceed, but no error message is displayed to the user.</li> </ul>
<b>Suggested fix</b>	Display an appropriate error message to the user indicating that the entered username already exists, ensuring transparency in the registration process.
<b>Status</b>	Closed
<b>Resolution</b>	Displayed error message to user
<b>Resolved by</b>	Vrushank Vaghani (developer)

Bug 3:

<b>Problem report number</b>	2024-03-21-003
<b>Reported by</b>	Nathan Binu Edappilly
<b>Date reported</b>	March 22, 2024

<b>Program (or component) name</b>	Parent Account GUI
<b>Release number</b>	1.0
<b>Version (build) identifier</b>	1.0.0
<b>Configuration(s)</b>	[Not Applicable]
<b>Report type</b>	Error not being presented
<b>Can reproduce</b>	Yes
<b>Severity</b>	Low
<b>Priority</b>	Medium
<b>Problem summary</b>	Child deletion is still successful when no child selected in the first place instead of showing error
<b>Key words</b>	Deletion, Bug, Null, Validation, Frontend



<b>Problem description and how to reproduce it</b>	<ol style="list-style-type: none"> <li>1. Open the Parent Account GUI.</li> <li>2. Navigate to the child deletion section.</li> <li>3. Attempt to delete a child without selecting any child from the dropdown menu.</li> <li>4. Notice that the deletion process proceeds without presenting any error message.</li> </ol> <p>See how child deletion takes place after selecting no child, in certain circumstances this may occur and for this reason, if this proceeds and the parent continues with the use of rest of the app, they may assume that after this, that the child has already been removed, when in reality no child was actually removed. Therefore, with this bug fix, the parent will know whether or not they made a selection of child to be removed in the first place.</p>
<b>Suggested fix</b>	Implement validation to ensure that a child is selected before proceeding with the deletion process. If no child is selected(meaning if selectedchild = null), display an error message prompting the user to choose a child from the dropdown menu.
<b>Status</b>	Closed
<b>Resolution</b>	Created if statements to check if the selected child is equal to null and to prompt the user to select a child.
<b>Resolved by</b>	Nathan Binu Edappilly (developer)

Bug 4:

<b>Problem report number</b>	2024-03-21-004
<b>Reported by</b>	William Moran
<b>Date reported</b>	2024/03/21
<b>Program (or component) name</b>	Proof of Completion
<b>Release number</b>	1.0
<b>Version (build) identifier</b>	1.0.0
<b>Configuration(s)</b>	Not Applicable
<b>Report type</b>	Design issue
<b>Can reproduce</b>	Yes
<b>Severity</b>	Low
<b>Priority</b>	Medium
<b>Problem summary</b>	Can't mark a chore as not completed in case the child pressed the wrong chore to mark as completed.
<b>Key words</b>	Uncompleted, child, button

<b>Problem description and how to reproduce it</b>	<p>-Navigate to the child account</p> <p>-A child clicks on a chore to complete and they accidentally click on one that they have not yet completed physically and press the mark as a completed button.</p> <p>-The child has no way of changing the status back so that it says it is not completed so they know to complete it in the future.</p>
<b>Suggested fix</b>	Have a button that says mark as not completed that allows it to change the status of the chore to not completed.
<b>Status</b>	Open
<b>Resolution</b>	[To be filled by the project manager]
<b>Resolved by</b>	Nathan Binu Edappily (Developer)

Bug 5:

<b>Problem report number</b>	2024-03-23-001
<b>Reported by</b>	Adam Raihan (tester)
<b>Date reported</b>	March 23, 2024
<b>Program (or component) name</b>	Parent Account GUI

<b>Release number</b>	1.0
<b>Version (build) identifier</b>	1.0.0
<b>Configuration(s)</b>	Database must be set up.
<b>Report type</b>	Design issue
<b>Can reproduce</b>	Yes
<b>Severity</b>	Low
<b>Priority</b>	[assigned by devs/proj. manager]
<b>Problem summary</b>	If user clicks 'add child' button, but then changes their mind and cancels it, an error message "Please enter a valid child username!" is displayed.
<b>Key words</b>	Parent Account, Bug, Validation
<b>Problem description and how to reproduce it</b>	<ul style="list-style-type: none"> <li>· Login to parent account</li> <li>· Click the "Add Child" button</li> <li>· Click on the "Cancel" button or the red x in the corner.</li> <li>· Notice that the message "Please enter a valid child username!" is displayed.</li> </ul>

<b>Suggested fix</b>	Implement validation checks in the Parent Account GUI class where the message “Please enter a valid child username!” is only displayed when the user clicks the ‘Ok’ button with an invalid child account in the text field.
<b>Status</b>	Open
<b>Resolution</b>	[To be filled by the project manager]
<b>Resolved by</b>	[To be filled by the project manager]

Bug 7:

<b>Problem report number</b>	2024-03-23-002
<b>Reported by</b>	Adam Raihan (tester)
<b>Date reported</b>	March 23, 2024
<b>Program (or component) name</b>	Parent Account GUI
<b>Release number</b>	1.0
<b>Version (build) identifier</b>	1.0.0
<b>Configuration(s)</b>	Database must be set up.
<b>Report type</b>	Design issue

<b>Can reproduce</b>	Yes
<b>Severity</b>	Low
<b>Priority</b>	Low
<b>Problem summary</b>	Chores fields do not reset to blank when parent creates new chore
<b>Key words</b>	Chore Creation, Bug, Frontend
<b>Problem description and how to reproduce it</b>	<ul style="list-style-type: none"> <li>· Login to a parent account</li> <li>· Create a chore by first filling in name, category, time, and payment fields, then clicking 'Create Chore' button</li> <li>· Notice that the fields in every category remain filled in with previous entries, even after that chore was created.</li> </ul>
<b>Suggested fix</b>	After a chore is completed, reset all fields in the Chore Creation section so that they are blank every time a new chore is created
<b>Status</b>	Closed
<b>Resolution</b>	Reset Chore Fields
<b>Resolved by</b>	Vrushank Vaghani (developer)

Bug 9:

<b>Problem report number</b>	2024-03-24-001
<b>Reported by</b>	Harmandeep Arneja (Tester)
<b>Date reported</b>	March 24th, 2024
<b>Program (or component) name</b>	Database For Storing User Information
<b>Release number</b>	1.0
<b>Version (build) identifier</b>	1.0.0
<b>Configuration(s)</b>	Database must be set up
<b>Report type</b>	Functional Issue
<b>Can reproduce</b>	Yes
<b>Severity</b>	Low
<b>Priority</b>	Medium
<b>Problem summary</b>	When a parent account attempts to check the balance without having any child accounts associated with it, the system generates an error, as it fails to locate any child usernames in the database.

<b>Key words</b>	Check Balance, Child Account, Error, Database, Error Message
<b>Problem description and how to reproduce it</b>	<p>Scenario: Checking Balance with No Child Accounts</p> <ol style="list-style-type: none"> <li>1) A parent attempts to check the balance associated with their child accounts.</li> </ol> <p>Steps to Reproduce:</p> <ol style="list-style-type: none"> <li>2) Log in to the ParentChoreSystem application as a parent with valid parent credentials.</li> <li>3) Click on the check balance button on the main parent page when no children have been added.</li> </ol> <ul style="list-style-type: none"> <li>• Expected Result: The system should display the balance associated with the parent's child accounts or should tell the user to add a child to view his/her balance.</li> <li>• Actual Result: The system generates an error, indicating that it could not locate any child usernames in the database.</li> </ul>
<b>Suggested fix</b>	<p>Modify the check balance functionality to handle cases where no child accounts are associated with the parent account. Instead of querying the database for child usernames directly, implement a check to verify if any child accounts exist before attempting to retrieve their balances. If no child accounts are found, display a user-friendly message indicating that no balances are available.</p>
<b>Status</b>	Open
<b>Resolution</b>	[To be filled by the project manager]
<b>Resolved by</b>	[To be filled by the project manager]