Customer Tests

Based on User-Stories worked on Iteration 1 and Iteration 2

1. ***Registration and Login:***

Test 1: Register and Login as a parent.

1. Open the app which will show up the Main Page where click on the “I’m a new parent” button to register as a new parent.
2. Enter the details, username, password and confirm password and click “Register”.
3. Once registered successfully, the app will show up the Parent Login Page or can be accessed from the Main Page. Fill up the details as registered above and click the “Login” button. If the username and password are entered correctly, it will show up the message of “Login successful!”.

Test 2: Register and Login as a parent.

1. Open the app which will show up the Main Page where click on the “I’m a new child” button to register as a new child.
2. Enter the details, username, password and confirm password and click “Register”.
3. Once registered successfully, the app will show up the Child Login Page or can be accessed from the Main Page. Fill up the details as registered above and click the “Login” button. If the username and password are entered correctly, it will show up the message of “Login successful!”

2. Chore Sorting Options:

Test 1 (Sort chores by time):

1. Register a parent account named will24 by clicking the register parent account button and the account is saved into the database for accounts.
2. Register a child account named james1 by clicking the register child account button and the account is saved into the database for accounts and childAccounts.
3. Log into the parent account by typing in the username and password and clicking the log-in button.
4. Add the new child account by clicking the add child button and type in the child’s username.
5. Create multiple chores such as dishes, folding laundry, and vacuuming by clicking the add chore button and filling out the description for the chore such as the type, category, time it takes, and payment.

6. Assign chores to a child from the dropdown menu by clicking the assign chore button.
7. Log out of the parent account by clicking the logout button.
8. Log into the child's account by clicking the child log-in button and entering account information.
9. Click on the sort by time button.
10. The list of chores is sorted by the time it takes to complete from shortest to longest.

Test 2 (Sort chores by category):

1. Follow the same steps from test 1 up to step 8.
2. Sort the list of chores by category by clicking on the filter by category button and type in the category.
3. Only chores of that category will be displayed.

Test 3 (Sort chores by payment):

1. Follow the same steps from test 1 up to step 8.
2. Sort the list of chores by their payment from lowest to highest by clicking on the sort by payment button.

Test 4 (Show only chores that are completed):

1. Follow the same steps from test 1 up to step 8.
2. Click on the chore that is completed and press the mark as completed button.
3. Click on the filter by the isCompleted button.
4. The list of chores that are completed by the child will be displayed.

Test 5 (Filter chores by the ones that have been paid by a parent)

1. Follow the same steps from test 1 up to step 8.
2. Click on the vacuuming chore that is completed and press the mark as completed button.
3. Log out of the child's account by pressing the log-out button.
4. Log into the parent account by typing in the username and password and clicking the log-in button.
5. Click on the james1 account from the dropdown menu.
6. Click on the chore that james1 completed so vacuuming.
7. Press the pay chore button.
8. Log out of the parent account by pressing the log-out button.
9. Log into the child's account by clicking the child log-in button and entering account information.
10. Press the filter isPaid button.

11. The list of chores that are paid by the parent will be displayed.

Test 6 (Sort chores by the ones that have not been completed yet):

1. Follow the same steps from test 1 up to step 8.
2. Press the filter by the isNotCompleted button.
3. The list of chores that have not been completed by the child yet will be displayed.

Test 7 (Sort chores by the ones that have not been paid yet):

1. Follow the same steps from test 1 up to step 8.
2. Press the filter by the isNotPaid button.
3. The list of chores that have not been paid for by the parent yet will be displayed.

Test 8 (Sort chores alphabetically)

1. Follow the same steps from test 1 up to step 8.
2. Press the sort by name button.
3. The list of chores that are assigned to the child will be listed in alphabetical order.

Test 9 (Sort chores by default)

1. Follow the same steps from test 1 up to step 8.
2. Press the default button.
3. The list of chores will be sorted by their id number, so the order in which they were created from oldest to newest.

3. Proof of Completion:

Test 1 (Child completes a chore):

1. Register a parent account named will24 by clicking the register parent account button and the account is saved into the database for accounts.
2. Register a child account named james1 by clicking the register child account button and the account is saved into the database for accounts and childAccounts.
3. Log into the parent account by typing in the username and password and clicking the log-in button.
4. Add the new child account by clicking the add child button and type in the child's username.
5. Create multiple chores such as dishes, folding laundry, and vacuuming by clicking the add chore button and filling out the description for the chore such as the type, category, time it takes, and payment. Chores created by that parent are stored in the database under chores.

6. Assign chores to a child from the dropdown menu by clicking the assign chore button.
7. Log out of the parent account by clicking the logout button.
8. Log into the child's account by clicking the log-in button and entering account information.
9. Click on the chore that is completed and press the mark as completed button.
10. The table will now show yes instead of no under the isCompleted column and a message will appear saying the chore is recorded as completed successfully!

Test 2 (Restrict a child to mark a chore as completed if it has already been completed by another child):

1. Register a parent account named will24 by clicking the register parent account button and the account is saved into the database for accounts.
2. Register a child account named james1 by clicking the register child account button and the account is saved into the database for accounts and childAccounts.
3. Register another child account named mike1 by clicking the register child account button and the account is saved into the database for accounts and childAccounts.
4. Log into the parent account by typing in the username and password and clicking the log-in button.
5. Add the new child account by clicking the add child button and type in the child’s username for james1.
6. Add the new child account by clicking the add child button and type in the child’s username for mike1.
7. Create multiple chores such as dishes, folding laundry, and vacuuming by clicking the add chore button and filling out the description for the chore such as the type, category, time it takes, and payment. Chores created by that parent are stored in the database under chores.
8. Assign the vacuuming and dishes chores to james1 from the dropdown menu by clicking on the assign chore button.
9. Assign the vacuuming and folding laundry to mike1 from the dropdown menu by clicking the assign chore button.
10. Log out of the parent account by clicking the logout button.
11. Log into the james1 account by clicking the child log-in button and entering the account information.
12. Click on the vacuuming chore row that is completed and press the mark as done button.
13. Log out of the james1 account by clicking the logout button.
14. Log into the mike1 account by clicking the child log-in button and entering the account information.
15. Click on the vacuuming chore row that is completed and press the mark as completed button.

16. Since the chore was already completed by james1 before mike1 finished it, in the table it says “yes” under the isCompleted column and a message will appear for mike1 that states the chore has already been completed.

Test 3 (Child accidentally marks chore as completed when it hasn’t been completed yet):

1. Follow steps 1-8 from test 1 for proof of completion.
2. Click on a chore that the child has not physically completed and press the mark as completed button.
3. Unable to mark the chore as not completed or undo the mistake.

4. Child Removal Feature for Parental Account management:

· Test 1 - Valid Removal process:
(Assuming you don’t have an existing username and password)

1) Child should create a username and password for their login details on the child account portal and login.

2) Parent should create a username and password for their login details on the parent account portal and login.

3) Parent should then add their child to the drop-down menu of the children that they have after the children have created their accounts by clicking the ‘add child’ button and entering the child’s username.

4) Next, depending on the preference of the parent they can remove a child by first selecting the child they want to take off, then they can click the ‘remove child’ button below.

5) They are then prompted by a pop up asking for reassurance of the removal of the chosen child account and is presented with options, ‘yes’ or ‘no’.

6) As they click yes as it is intended to test the delete function of our code, the child chosen is taken off the drop-down menu consisting of all their children in the parent Account GUI. In addition, the child account that has been deleted is taken off the database called ‘Accounts’ where the parent accounts and list of children under the specific parent is stored.

Test 2 – Cancel Removal process:
(Assuming that you have removed the child account you have added from above but still have the parent account)

1) Let's continue from the previous test, assuming that now you have no child added on the parent account anymore. Create a new child account on the child registration portal.

2) On the child account portal, create a username and password for the child and login.

3) Then on the parent portal, the parent should select the 'add child' button and add the username of the child created.

3) Now based on the circumstance, if the parent wants to remove the child then they can select the child from the drop-down menu towards the top of the screen and then click 'remove child'. However, if when they click to remove the child, if they by mistake, have selected the wrong child or have changed their mind of removing the child, then the pop up presented after clicking remove child asks if they are sure about their decision.

4) For the case of this test, select 'No'.

5) This now cancels the change of removing the child and thus retains the child in the parent account.

6) On the other hand, the information of the child still remains in the 'Accounts' database however there is a 'null' link to a parent.

7) (Another reason where this feature is helpful is if the parent has multiple children and if they were trying to remove one, but the unintended child was selected by mistake to be removed, the prompt asking for reassurance of deletion can save time to the parent by just plainly selecting 'no' on the prompt.)

Test 3 – Invalid Child selection:
(Assuming no child is added in the first place)

1)Let's say now that a parent, after having their account logged in, decides to remove a child without adding one in the first place or on another occasion doesn't select a child at all after being previously added.

2) Then without selecting adding any child the parent clicks remove child.

3) After clicking 'remove child' without even having a child added, an error pops up stating that the parent should select a child to be removed (as currently selectedchild == null).

4) After this alert is presented, the parent essentially knows what has gone wrong in the process and is directed to first add a child before making the decision to remove them.

5. Competition Algorithm Implementation

Test 1:

1. Follow the above steps to register a parent and register at least 2 children.
2. Let us suppose, parent is vrushank and children are bob and jim.
3. Login as parent, vrushank, using Parent Login from the Main Page.
4. Once logged in, vrushank's home page will be shown. Click "Add Child" to add both the children one by one by entering their usernames.
5. Create a chore, using the chore template given beneath the first row of buttons, by entering all the details of the chore and clicking "Create Chore" button.
6. Now, click on "Create Competition" button to create a competition between 2 children. After clicking, a form-like page will open to initialize a competition, then enter Competition Name, select the chore for competition and, select all the children that will be the participants of the competition.
7. Then, click the "Create Competition" button.
8. If competition is created without any failures, then it will pop-up the message of "Competition created successfully!".

Test 2:

1. Follow the Test 1 steps from 1 to 5.
2. Now, click on "Create Competition" button to create a competition between 2 children. After clicking, a form-like page will open to initialise a competition, then enter all the details except one.
3. Then, click the "Create Competition" button.
4. There will be a proper message of a failure stating the constraint that was not filled or selected. It will only successfully create a competition only if all the constraints, i.e., all the details required are filled or selected appropriately like select at least 2 child accounts, enter the name of competition, select only one chore.

Database Tests

Test Case 1: Register a Parent Account Successfully

1) Let's say a parent wants to register their account and they start by navigating to the main login page.
2) They then click the "Register as a Parent" button and fill in the registration information with valid parent account details (username, password).
3) They then press the register button.
    - Expected Result: Parent account is successfully registered in the database and the username and password show up properly.

Test Case 2: Register a Parent Account with Existing Username and Password

1) Let's now say that a parent already has an account registered but, they try to register an account as a parent again. So they start by navigating to the main login page.
2) They then click the "Register as a Parent" button and fill in the registration information with an existing parent username and password.
3) They try to press the register button and are then presented with an error message.
    - Expected Result: Error message indicating that the username is already taken or a message saying that the account is already registered and that you should login instead (Database should not record these entries).

Test Case 3: Register a Child Account Successfully

1) A parent wants to register their child to participate in the chore management system. Navigate to the main login page.
2) Click the "Register as a Child" button and fill in the registration information with valid child account details (username, password).
3) Press the register button.
    - Expected Result: Child account is successfully registered in the database and the username and password show up properly.

Test Case 4: Register a Child Account with Existing Username

1) A parent tries to register a child with a username that is already in use .Navigate to the main login page.
2) Click the "Register as a Child" button and fill in the registration information with an existing child username and password.
3) Press the register button.
    - Expected Result: Error message indicating that the username is already taken or a message saying that the account is already registered and that you should login instead (Database should not record these entries).

Test Case 5: Assign Parent to Child Successfully

1) A parent wants to link their account to their child's account for chore management. The parent logs in as a parent.
2) Navigate to the child management page.
3) Select a child account.
4) Assign a parent to the selected child.
    - Expected Result: Parent is successfully assigned to the child in the database.

Test Case 6: Assign Parent to Non-existent Child

1) A parent attempts to assign themselves to a child account that does not exist. The parent logs in as a parent.
2) Navigate to the child management page.
3) Attempt to assign a parent to a non-existent child.
    - Expected Result: Error message indicating that the child does not exist (Adding the non-existent child is NOT reflected on the database).

Test Case 7: Remove Parent from Child without Parent Assignment

1) A parent tries to remove themselves from a child account that has not been assigned to any parent. The parent logs in as a parent.
2) Navigate to the child management page.
3) Select a child account without any assigned parent.
4) Attempt to remove the parent from the child.
    - Expected Result: Error message indicating that no parent is assigned to the child.

Test Case 8: Add Chore Successfully

1) A parent wants to add a new chore to the system for their children to complete. The parent logs in as a parent.
2) Navigate to the chore list page.
3) Add a new chore with valid details (name, category, time, payment).
    - Expected Result: The chore is successfully added to the database.

Test Case 9: View All Chores Successfully

1) A parent wants to see all the chores listed in the system. The parent logs in as a parent.
2) Navigate to the chores list page.
3) View all chores listed on the page.
    - Expected Result: All chores stored in the database are displayed correctly on the page (This test can be repeated to accommodate the child account trying to view all the chores as well).

Test Case 10: Verify Username Existence During Account Login

1) A user attempts to log in with their registered username. The parent or child navigates to the login page of the application.
2) They enter a registered username into the username field.
3) Input a valid password into the password field.
4) The parent/child clicks on the login/submit button.

Expected Result:

- The system validates that the entered username exists in the database and verifies that the user exists and lets them log in to the application.

Test Case 11: Verify Incorrect Password and Prompt User

1) A user attempts to log in with a valid username but an incorrect password.
2) The parent/child visits the login page of the application.
3) They enter a valid username into the designated field for their respective login.
4) They then input an incorrect password associated with the entered username.
5) They click on the login button.

Expected Result:

- The system displays an error message indicating that the provided username or password is incorrect (and that it does not exist in the database).

Test Case 12: Verify Non-existent Username

1) A user attempts to log in with a username that does not exist in the system.
2) The parent/child attempts to access the login page of the application.
3) They try to enter a non-existent username into the username field.
4) They input a valid password.
5) They submit the login by clicking on the login button.

Expected Result:

- The system displays an error message informing the user that the entered username or password does not exist (in the database).

Test Case 13: Verify Successful Login

1) A user attempts to log in with valid credentials.
2) The parent/child goes to the login page of the application.
3) They enter a valid username.
4) They input the correct corresponding password.
5) They then submit the login by clicking on the login button.

Expected Result:

- The system grants access to the user's account, allowing them to proceed to their dashboard or designated landing page.

Test Case 14: Verify Empty Username

1) A user attempts to log in without providing a username/password and leaving it empty.
2) The parent/child goes to the login page of the application.
3) They leave the username/password field blank.
4) They then click on the login button to attempt to login.

Expected Result:

- The system prompts the user to enter a username or password before proceeding with the login process.

Test Case 15: Check Balance of Child Account from Parent Account

1) A parent wants to check the balance associated with their child's account from the parent's account dashboard.
2) The parent logs in as a parent and navigates to the dashboard of the ParentChoreSystem application.
3) They select the child account for which the balance needs to be checked.
4) They press the check balance button for the specified child.
5) Verify that the current balance is displayed accurately for the selected child account.

Expected Result:

- The system accurately displays the current balance associated with the selected child account from the parent's account dashboard.
- The displayed balance corresponds to the sum of rewards or payments earned by the child through completed chores.

Test Case 16: Verify Completion Status of Assigned Chore

1) A parent wants to verify whether a chore assigned to their child has been completed.
2) The parent logins in with the correct login credentials and they then view the chore list section of the dashboard of the ParentChoreSystem application.
3) They then locate the chore assigned to the child account in question.
4) Verify the completion status of the chore (If it says yes or no).

Expected Result:

- If the chore is marked as completed in the database, the test result should indicate "Yes" under the isCompleted column.
- If the chore is not marked as completed in the database, the test result should indicate "No" under the isCompleted column.
- The displayed completion status should accurately reflect the actual completion status of the assigned chore.

Test Case 17: Verify Payment Status of Assigned Chore

1) A parent wants to verify whether a chore assigned to their child has been paid.
2) The parent is logged into the ParentChoreSystem application with valid credentials. The parent has then assigned a chore to their child account.
3) The parent views the chores list on the dashboard of the ParentChoreSystem application.
4) Locate the chore assigned to the child account in question.
5) Verify the payment status of the chore.

Expected Result:

- If the chore is marked as paid in the database, the test result should indicate "Yes" under the isPaid column.
- If the chore is not marked as paid in the database, the test result should indicate "No" under the isPaid column.
- The displayed payment status should accurately reflect the actual payment status of the assigned chore.

Test Case 18: Verify Database Update on Adding Child to Parent Account

1) A parent adds a child to their parent account in the ParentChoreSystem application.
2) The parent is logged into the ParentChoreSystem application with valid credentials.
3) The parent clicks on the option to add a child to the parent account.
4) Fill in the necessary details for the new child account (e.g. username).
5) Press the add child button to add the child to the parent account.

Expected Result:

- The database should be updated to include the newly added child account under the parent's account.
- Upon successful addition, the child's details should be stored in the database, linked to the parent's account.

Test Case 19: Verify Database Update on Removing Child from Parent Account

1) A parent removes a child from their parent account in the ParentChoreSystem application.
2) The parent is logged into the ParentChoreSystem application with the correct login details.
3) The parent navigates to the list of all the added children sections of the dashboard.
4) The parent selects the option to remove a child from the parent account.
5) Choose the child account to be removed from the list of associated children.
6) The parent confirms the removal action.

Expected Result:

- The database should be updated to remove the selected child account from the parent's account.
- Upon successful removal, the child's details should be deleted or unlinked from the parent's account in the database and the database should show that the parent linked to the child account shows up as “NULL”.

Restricted View of Competition Standings for Parental Control*

Test 1:

1. Register a new parent account named adamr by clicking the register parent account button, entering ‘adamr’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts.
2. Register a new child account named zachr by clicking the register child account button, entering ‘zachr’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts and childAccounts.
3. Register a new child account named jackr by clicking the register child account button, entering ‘jackr’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts and childAccounts.
4. Enter SELECT * FROM accounts; in terminal to see a table containing all three accounts
5. Log into the parent account by typing in the username and password and clicking the log-in button.
6. Add both new child accounts by clicking the add child button and type in each child's username one by one.
7. Create one chore such as dishes. Do this by first filling out the description for the chore such as the name, category, amount of time it takes, and payment, then clicking ‘Create Chore’.
8. Click ‘Create Competition’. Name the competition, ”First to do the Dishes!”, select both children, zachr and jackr, select the chore that was made in step 7, and finally click ‘Start Competition’.

9. Click ‘Competition Standings’. Click on the competition named ”First to do the Dishes!”. jackr and zachr will be seen in the participants list. The winner section will say “Not completed by anyone yet”. Click ‘Return’. Log out
10. Log in to the jackr account by clicking ‘Child Login’ and entering the username and password.
11. Click ‘Competition Standings’. This will look identical to the parents' view.
12. Click on the chore and click the ‘Mark as completed’ button
13. Now click ‘Competition Standings’ again. This time the winner section will say ‘jackr’
14. Logout and login into the zachr account. Click on ‘Competition Standings’, and the same page will appear, with the winner saying ‘jackr’
15. Logout and login to the parent account adamr. The chore will now be checked as completed. Click ‘Competition Standings’, and the same page will appear, with jackr as the winner. Log out
16. As we can see adamr, the parent, and zachr and jackr, both children, can see the competition details, but only the parent, adamr, can create a competition.

Test 2:

1. Register a parent account named adamr by clicking the register parent account button, entering ‘adamr’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts.
2. Register a child account named zachr by clicking the register child account button, entering ‘zachr’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts and childAccounts.
3. Register a child account named jackr by clicking the register child account button, entering ‘jackr’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts and childAccounts.
4. Register a child account named mackr by clicking the register child account button, entering ‘jackr’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts and childAccounts.
5. Enter SELECT * FROM accounts; in terminal to see a table containing all four accounts.
6. Log into the parent account by typing in the username and password and clicking the log-in button.
7. Add all 3 child accounts by clicking the add child button and type in each child's username one by one.

8. Create 3 chores such as dishes, making bed, and vacuuming. Do this by first clicking the add chore button and filling out the description for each chore such as the name, category, amount of time it takes, and payment, then clicking ‘Create Chore’.
9. Click ‘Create Competition’. Name the competition,” Zach vs Jack”, select both children, zachr and jackr, select 1 chore that was made in step 8, and finally click ‘Start Competition’.
10. Click ‘Create Competition’. Name the competition, ”Mach vs Jack”, select both children, machr and jackr, select 1 chore that was made in step 8, and finally click ‘Start Competition’.
11. Click ‘Create Competition’. Name the competition, ”All three”, select all children, machr, jackr and zachr, select 1 chore that was made in step 8, and finally click ‘Start Competition’.
12. Click ‘Competition Standings’. Click on the competition named ”Zach vs Jack”. jackr and zachr will be seen in the participants list. The winner section will say “Not completed by anyone yet”.
13. Click on the competition named ”Mach vs Jack”. jackr and machr will be seen in the participants list.
14. Click on the competition named ”All three”. jackr, machr and zachr will be seen in the participants list. Log out
15. Log in to the jackr account by clicking ‘Child Login’ and entering the username and password.
16. Click ‘Competition Standings’. The competition ”Zach vs Jack” will be visible, as well as ”Mach vs Jack”, and ”All three”.
17. Logout and login into the zachr account. Click on ‘Competition Standings’. Only ”Zach vs Jack” and “All three” will be visible
18. Logout and login into the machr account. Click on ‘Competition Standings’. Only ”Mach vs Jack” and “All three” will be visible

Test 3:

1. Register a parent account named adamr by clicking the register parent account button, entering ‘adamr’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts.

2. Register a parent account named steveh by clicking the register parent account button, entering ‘steveh’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts.
3. Register a child account named zachr by clicking the register child account button, entering ‘zachr’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password

field. The account is saved into the database for accounts and childAccounts.

4. Register a child account named jackr by clicking the register child account button, entering ‘jackr’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts and childAccounts.
5. Register a child account named alexh by clicking the register child account button, entering ‘alexh’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts and childAccounts.
6. Register a child account named bobh by clicking the register child account button, entering ‘bobh’ as the username, then any password etc.’1234’, and finally entering the same password in the confirm password field. The account is saved into the database for accounts and childAccounts.
7. Enter SELECT * FROM accounts; in terminal to see a table containing all six accounts
8. Log into the parent account adamr by typing in the username and password and clicking the log-in button.
9. Add the child accounts zachr and jackr by clicking the add child button and type in each child's username one by one.
10. Create 1 chore such as dishes. Do this by first clicking the add chore button and filling out the description for each chore such as the name, category, amount of time it takes, and payment, then clicking ‘Create Chore’.
11. Click ‘Create Competition’. Name the competition,” Zach vs Jack”, select both children, zachr and jackr, select the chore that was made in step 8, and finally click ‘Start Competition’.
12. Click ‘Competition Standings’. Click on the competition named ”Zach vs Jack”. jackr and zachr will be seen in the participants list. The winner section will say “Not completed by anyone yet”. Log out.
13. Log in to the steveh account by clicking ‘Parent Login’ and entering the username and password.
14. Add the child accounts alexh and bobh by clicking the add child button and type in each child's username one by one.
15. Create 1 chore such as dishes. Do this by first clicking the add chore button and filling out the description for each chore such as the name, category, amount of time it takes, and payment, then clicking ‘Create Chore’.
16. Click ‘Create Competition’. Name the competition,” Alex vs Bob”, select both children, alexh and bobh, select the chore that was made in step 15, and finally click ‘Start Competition’.

17. Click ‘Competition Standings’. Click on the competition named ”Alex vs Bob”. alexh and bobh will be seen in the participants list. The winner section will say “Not completed by anyone yet”. Log out.
18. Enter SELECT * FROM Childaccounts; in terminal to see a table containing all four child accounts, as well as their assigned parent.
19. Login into adamr. Click ‘Competition Standings’. Only the ”Zach vs Jack” competition will be shown. Log out
20. Login into zachr. Click ‘Competition Standings’. Only the ”Zach vs Jack” competition will be shown. Log out
21. Login into jackr. Click ‘Competition Standings’. Only the ”Zach vs Jack” competition will be shown. Log out
22. Login into steveh. Click ‘Competition Standings’. Only the ”Alex vs Bob” competition will be shown. Log out
23. Login into alexh. Click ‘Competition Standings’. Only the ”Alex vs Bob” competition will be shown. Log out
24. Login into bobh. Click ‘Competition Standings’. Only the ”Alex vs Bob” competition will be shown. Log out

Customer Tests Based on User-Stories worked in Iteration 3

1) Adam - parent gui chore assignment and chore removal

Test Case 1: Chore Assignment and Deletion

1. Setup:
   - Parent registers an account using the parent registration option.
   - Parent enters username "John" and password "12345".
   - Child registers an account using the child registration option.
   - Child enters username "Adam10" and password "000".

2. Login:
   - Parent logs in to their account using the credentials provided (Username: John, Password: 12345).

3. Add Child:
   - Parent clicks on "Add Child".
   - Parent enters Adam's username ("Adam10").
   - Verification: Adam's account is successfully linked to John's parent account.

4. Create Chore:

- Parent navigates to the "Create Chore" section.
- Parent enters chore details for "washing dishes".
- Verification: The chore "washing dishes" is created and displayed in the central panel.

5. Assign Chore:
- Parent selects the "washing dishes" chore from the central panel.
- Parent selects Adam from the drop-down list.
- Parent clicks on "Assign chore" button.
- Verification: The chore "washing dishes" is assigned to Adam successfully.

6. Delete Chore:
- Parent realizes the chore assignment was incorrect.
- Parent selects the "washing dishes" chore from the central panel.
- Parent clicks on "Delete chore" button.
- Verification: The chore "washing dishes" is removed from the system.

Test Case 2: Chore Assignment and Deletion

1. Setup:
- Parent registers an account with the username "Alice" and password "alice123".
- Child registers an account with the username "Bob" and password "bob456".
- Parent logs in to their account.

2. Add Child:
- Parent navigates to the "Add Child" section.
- Parent enters "Bob" as the child's username and confirms.
- Verification: The system links Bob's account to Alice's parent account successfully.

3. Create Chore:
- Parent navigates to the "Create Chore" section.
- Parent enters chore details, such as "cleaning the room".
- Verification: The chore "cleaning the room" is created and displayed in the central panel.

4. Assign Chore:
- Parent selects the "cleaning the room" chore from the central panel.
- Parent selects Bob from the drop-down list.
- Parent clicks on "Assign chore" button.
- Verification: The chore "cleaning the room" is assigned to Bob successfully.

5. Delete Chore:
- Parent selects the "cleaning the room" chore from the central panel.

- Parent clicks on "Delete chore" button.
- Verification: The chore "cleaning the room" is removed from the system.

Test Case 3: Chore Reassignment

1. Setup:
   - Parent registers an account with the username "Eva" and password "eva789".
   - Child registers an account with the username "Liam" and password "liam2023".
   - Parent logs in to their account.

2. Add Child:
   - Parent navigates to the "Add Child" section.
   - Parent enters "Liam" as the child's username and confirms.
   - Verification: The system links Liam's account to Eva's parent account successfully.

3. Create Chore:
   - Parent navigates to the "Create Chore" section.
   - Parent enters chore details, such as "walking the dog".
   - Verification: The chore "walking the dog" is created and displayed in the central panel.

4. Assign Chore:
   - Parent selects the "walking the dog" chore from the central panel.
   - Parent selects Liam from the drop-down list.
   - Parent clicks on "Assign chore" button.
   - Verification: The chore "walking the dog" is assigned to Liam successfully.

5. Reassign Chore:
   - Parent realizes they made a mistake and wants to assign a different chore to Liam.
   - Parent selects the "walking the dog" chore from the central panel.
   - Parent clicks on "Delete chore" button.
   - Parent creates a new chore, "watering the plants".
   - Parent assigns the new chore "watering the plants" to Liam.
   - Verification: The chore "walking the dog" is removed, and the new chore "watering the plants" is assigned to Liam successfully.

---

2) Adam - Chore presets on parent account GUI

Test Case 1: Chore Assignment (Make Bed)

1. Parent Account Creation/Login:
   - Parent creates a parent account with the username "Pedro123" and password "2311".
   - Verification: Account successfully created.

2. Child Account Creation/Login:
   - Child creates an account with the username "Jimmy111" and password "100".
   - Verification: Account successfully created.

3. Chore Assignment:
   - Parent logs in with username "Pedro123" and password "2311".
   - Parent navigates to the chore assignment section.
   - Parent selects "Make Bed" from the preset chore options.
   - Verification: Chore details (Name: Make Bed, Category: Household, Time: 10 minutes, Payment: None) pre-filled.

4. Assign Chore:
   - Parent reviews the chore details.
   - Parent selects Jimmy as the recipient.
   - Parent clicks on the "Assign chore" button.
   - Verification: Chore "Make Bed" successfully assigned to Jimmy.

Test Case 2: Chore Assignment (Shovel):

1. Parent Account Login:
   - Parent logs in with their existing parent account credentials (Username: Nathan123, Password: 2311).
   - Verification: Successfully logged in.

2. Child Account Login:
   - Child logs in with their existing child account credentials (Username: Jimmy111, Password: 100).
   - Verification: Successfully logged in.

3. Chore Assignment:
   - Parent accesses the chore assignment section.
   - Parent selects "Shovel" from the preset chore options.
   - Verification: Chore details (Name: Shovel, Category: Outdoor, Time: 1 hour, Payment: $10) pre-filled.

4. Assign Chore:
   - Parent reviews the chore details.
   - Parent selects Jimmy as the recipient.
   - Parent clicks on the "Assign chore" button.

- Verification: Chore "Shovel" successfully assigned to Jimmy.

---

3) Chore Descriptions: Include the option for parents to add detailed descriptions or instructions for each chore, ensuring clarity and consistency in task execution. - Nathan

Test 1: Chore Description (1)

1. Parent and Child Login:
   - Existing parent "Nathan123" logs in with password "2311".
   - Existing child "Jimmy111" logs in with password "100".
   - Verification: Successful login for both parent and child.

2. Chore Creation:
   - Parent navigates to the chore creation section.
   - Parent selects "Mowing the Lawn" from the preset chore options.
   - Parent fills in the chore details for 'Name', 'Category', 'Time', 'Payment', etc.
   - Verification: Chore details entered successfully.

3. Chore Description Addition:
   - Parent wants to provide specific instructions for mowing the lawn.
   - Parent uses the chore description feature to specify areas of focus or additional instructions.
   - Verification: Chore description added successfully.

4. Assign Chore:
   - Parent selects Jimmy as the recipient for the chore.
   - Parent clicks on the "Assign chore" button.
   - Verification: Chore "Mowing the Lawn" with detailed description successfully assigned to Jimmy.

Test Case 2: Chore Description (Clean Room)

1. Parent and Child Login:
   - Existing parent "Nathan123" logs in with password "2311".
   - Existing child "Jimmy111" logs in with password "100".
   - Verification: Successful login for both parent and child.

2. Chore Creation:
   - Parent navigates to the chore creation section.
   - Parent selects "Clean Room" from the preset chore options.
   - Parent fills in the chore details for 'Name', 'Category', 'Time', 'Payment', etc.
   - Verification: Chore details entered successfully.

3. Chore Description Addition:
- Parent wants to provide specific instructions for cleaning the room.
- Parent uses the chore description feature to specify areas of focus or additional instructions.
- Verification: Chore description added successfully.

4. Assign Chore:
- Parent selects Jimmy as the recipient for the chore.
- Parent clicks on the "Assign chore" button.
- Verification: Chore "Clean Room" with detailed description successfully assigned to Jimmy.

---

4) Chore History: Button on child GUI that lets children view entire history of chores they have completed [green highlight for completed at deadline, red highlight for completing after deadline, orange highlight if chore is completed by another child having the same chore, white is for not completed at all] (Nathan)

Test 1: Chore History Feature (Child Account GUI)

1. Login:
- An existing parent and child login to the child account GUI. Parent's username is 'Nathan123' with password '2311', and the child's username is 'Jimmy111' with password '100'.

2. Nathan, who's the parent can then assign 2 or more chores with a deadline as well so lets say, ‘Make bed - Bedroom - 15 min - 10 dollars’ and ‘stack shelves - kitchen - 10 min - 5 dollars' along with the previous chore in the above test case which was cleaning the room. Let's say all of these 3 chores are set with a 2 day deadline and after being set, the children complete them after the deadline, i.e: before the deadline.

3. View Chore History:
- The child navigates to the chore history section by clicking on the "History" button in the child account GUI.
- The system retrieves and displays a list of completed chores along with their details, including chore ID, name, category, time, payment, completion status, payment status, and rating if available.
- The child verifies that the displayed chore history is accurate and matches the chores they have completed.
- The history page as well highlights those 3 chores green showing that they have completed it on time.

4. Verify Chore Completion Status:
- The child checks the completion status of each chore listed in the chore history.

- For each completed chore, the child ensures that the completion status is displayed as "Yes" and confirms that they indeed completed those chores.
- For any chore marked as not completed, the child ensures that the completion status is displayed as "No" and verifies that they did not complete those chores.

Test 2: Chore History Feature (Failed to complete chores by deadline)

1. Login:
- An existing parent and child login to the child account GUI. Parent's username is 'Nathan123' with password '2311', and the child's username is 'Jimmy111' with password '100'.

2. Nathan, who's the parent can then assign 2 or more chores with a deadline as well so lets say, ‘dusting - living room - 20 min - 20 dollars’ and ‘set the table - kitchen - 10 min - 15 dollars' along with the previous chore in the above test cases which was cleaning the room. Let's say all of these 3 chores are set with a 2 day deadline and after being set, the children complete them after the deadline, i.e: before the deadline.

3. View Chore History:
- The child navigates to the chore history section by clicking on the "History" button in the child account GUI.
- The system retrieves and displays a list of completed chores along with their details, including chore ID, name, category, time, payment, completion status, payment status, and rating if available.
- The child verifies that the displayed chore history is accurate and matches the chores they have completed.
- The history page as well highlights those 3 chores red showing that they have completed it although not at the deadline.

4. Verify Chore Completion Status:
- The child checks the completion status of each chore listed in the chore history.
- For each completed chore, the child ensures that the completion status is displayed as "Yes" and confirms that they indeed completed those chores.
- For any chore marked as not completed, the child ensures that the completion status is displayed as "No" and verifies that they did not complete those chores.

Test 3: Chore History Feature (Not completed)

1. Login:
- An existing parent and child login to the child account GUI. Parent's username is 'Nathan123' with password '2311', and the child's username is 'Jimmy111' with password '100'.

2. Nathan, who's the parent can then assign 2 or more chores with a deadline as well so lets say, ‘dusting - living room - 20 min - 20 dollars’ and ‘set the table - kitchen - 10 min - 15 dollars' along with the previous chore in the above test cases which was cleaning the room. Let's say all of these 3 chores are set with a 2 day deadline and after being set, the children didn't complete it at all.

3. View Chore History:
  - The child navigates to the chore history section by clicking on the "History" button in the child account GUI.
  - The system retrieves and displays a list of completed chores along with their details, including chore ID, name, category, time, payment, completion status, payment status, and rating if available.
  - The child verifies that the displayed chore history is accurate and matches the chores they have completed.
- The history page as well highlights those 3 chores white showing that they have not completed it at all.

4. Verify Chore Completion Status:
  - The child checks the completion status of each chore listed in the chore history.
  - For each completed chore, the child ensures that the completion status is displayed as "Yes" and confirms that they indeed completed those chores.
  - For any chore marked as not completed, the child ensures that the completion status is displayed as "No" and verifies that they did not complete those chores.

Test 4: Chore History Feature (Completed by another child of the parent within deadline)

1. Login:
  - An existing parent and child login to the child account GUI. Parent's username is 'Nathan123' with password '2311', and the child's username is 'Jimmy111' with password '100'. Now, let's say Nathan wants to add his second child to his account. For this, his second child Harman has to go through the child registration process and he creates ‘Username: Harman567’ and ‘Password: 123456’

2. Nathan, who's the parent can then assign 2 or more chores with a deadline as well so lets say, ‘dusting - Living room - 20 min - 20 dollars’ and ‘set the table - kitchen - 10 min - 15 dollars' to Jimmy and ‘dusting - Living Room - 20 min - 20 dollars’ and ‘clean up desk - Study room - 10 min - 15 dollars' was assigned to Harman . Let's say all of these chores are set with a 2 day deadline and after being set, Jimmy finishes all his chores within the deadline except the dusting of the living room, however, Harman finishes all his chores including the dusting of the living room.

3. View Chore History:

  - The child navigates to the chore history section by clicking on the "History" button in the child account GUI.

- The system retrieves and displays a list of completed chores along with their details, including chore ID, name, category, time, payment, completion status, payment status, and rating if available.
- The child verifies that the displayed chore history is accurate and matches the chores they have completed.
- Now when Jimmy opens up his chore history on his account it will show up as 1 green and 1 orange as his sibling has completed the same chore that he is assigned. On the other side, when Harman clicks chore history then he sees 2 green highlighted chores as he has completed both of his chores within the deadline.

4. Verify Chore Completion Status:
- The child checks the completion status of each chore listed in the chore history.
- For each completed chore, the child ensures that the completion status is displayed as "Yes" and confirms that they indeed completed those chores.
- For any chore marked as not completed, the child ensures that the completion status is displayed as "No" and verifies that they did not complete those chores.

Deadline Feature for Chores Customer Tests:

Test Case 1: Setting Deadline for Chores (Parent Perspective)

Scenario:

As a parent, I want to set deadlines for chores and see when tasks are approaching or have passed their due date.

Preconditions:

- The parent has an active user account.
- The parent has assigned chores to the child.
- The parent has access to the chore management section.

Test Steps:

1. Parent logs into their account.
2. Parent navigates to the chore creation section.
3. Parents create a chore either by selecting a preset or creating a unique one and sets a deadline for it.
4. Parent confirms the deadline and creates the chore.
5. Parent verifies that the deadline is successfully set for the chore..

Expected Results:

- The parent successfully sets a deadline for the chore.
- The deadline is not on a date before the current date.
- The database displays the deadline for the chore accurately.

- As the chore deadline approaches or passes, the parent and child both can view the deadline under chore additional details.

Test Case 2: Chore Deadline Notification (Child Perspective)

Scenario:

As a child, I want to be able to see when the deadline for a chore is approaching or has passed.

Preconditions:

- The child has an active user account.
- The child has been assigned chores by the parent with deadlines set.
- The child has access to the chore list on the dashboard.

Test Steps:

1. Child logs into their account.
2. Child navigates to the chore list on the dashboard.
3. Child checks the list of assigned chores and their respective deadlines.
4. Child verifies the deadlines set for each chore under additional details.
5. Child completes the chore before or on the deadline.
6. Child waits for confirmation from the parent regarding the completion of the chore.

Expected Results:

- The child sees the correct date set and is able to mark it as complete either on or after the deadline.
- If the chore is completed by the deadline, it will be coloured green when the child has completed the chore on time.
- Database accurately reflects the deadline as well.

Payment Reminder Feature Customer Tests:

Test Case 1: Payment Reminder for Completed Chores (Parent Perspective)

Scenario: As a parent, I want to receive a reminder to pay my child when they complete their assigned chores.

Preconditions for the test case:

- The parent and child both have active user accounts.
- The parent has assigned chores to the child.
- The child has completed one or more chores.

Test Steps:

1. Parent logs into their account.
2. Parent navigates to the chores management section.
3. Parent checks the list of completed chores.
4. If there are completed chores that require payment, the parent receives a notification or message reminding them to pay the child.
5. Parent confirms the payment and makes the necessary transaction.

Expected Results:

- The parent receives a payment reminder message for each completed chore that requires payment.
- After confirming the payment, the parent successfully makes the transaction to pay the child for the completed chores.

Test Case 2: Payment Reminder for Completed Chores (Child Perspective)

Scenario:

As a child, I want my parents to receive a reminder to pay me when I complete my assigned chores.

Preconditions:

- The parent and child both have active user accounts.
- The parent has assigned chores to the child.
- The child has completed one or more chores.

Test Steps:

1. Child logs into their account.
2. Child navigates to the completed chores section or dashboard.
3. Child verifies that the completed chores are listed accurately.
4. Child checks if there is a notification indicating that the payment reminder has been sent to the parent.
5. Child waits for confirmation from the parent regarding the payment.

Expected Results:

- The child sees a notification indicating that the payment reminder has been sent to the parent for the completed chores.
- The child receives confirmation from the parent that the payment has been made for the completed chores.

Priority Feature Customer Tests:

Test Case 1: Setting Priority for Chores (Parent Perspective)
Scenario: As a parent, I want to assign priority levels to chores to communicate their importance to my child effectively.
Preconditions:

- The parent has an active user account.
- The parent has access to the chore management section or dashboard.
- The parent has assigned chores to their child.

Test Steps:

1. Parent logs into their account.
2. Parent navigates to the chores list section on the dashboard.
3. Parent selects a chore and sets its priority level as High from the options: (high, medium, or low).
4. Parents assign the chore to their child with the respective priority level.
5. Parent verifies that the priority level is correctly displayed next to the assigned chore.

Expected Results:

- The parent successfully sets the priority level for the chore.
- The assigned chore displays the correct priority level next to it.
- The priority level shows up on the database as well as in the chore full details.

Test Case 2: Viewing Chores by Priority (Child Perspective)
Scenario: As a child, I want to see chores based on their priority level to understand their importance and complete them accordingly.
Preconditions:

- The child has an active user account.
- The child has been assigned chores by the parent with different priority levels.

Test Steps:

1. Child logs into their account.
2. Child navigates to the chores list section on the dashboard.
3. Child can see all the chores based on the priority level (high, medium, or low) that each chore has under additional details.
4. Child verifies that the chores are displayed according to their priority levels.
5. Child completes the chores based on their priority levels.

Expected Results:

- The child successfully filters or sorts the chores by priority level.
- The chores are displayed in the correct order of priority (high to low).
- The child understands the importance of each chore and completes them accordingly.

Chore Count Feature Customer Test:

Test Case 1: Chore Count Feature (Parent Perspective)
Scenario: As a parent, I want to keep track of the number of chores assigned to my child and their completion status.
Preconditions:

- The parent has an active user account.
- The parent has access to the chores list on the dashboard.
- Chores have been assigned to the child.

Test Steps:

1. Parent logs into their account.
2. Parent navigates to the chore management section on the dashboard.
3. Parent views the total number of chores assigned to the child.
4. Parent verifies the breakdown of chores by status (completed, pending).
5. Parent confirms the accuracy of the chore count and status.

Expected Results:

- The parent can view the total number of chores assigned to the child.
- The breakdown of chores by status (completed, pending) is displayed accurately.
- The chore count matches the actual number of assigned chores, and the status reflects their current completion state.
- Database reflects this as well

Test Case 2: Chore Count Feature (Child Perspective)
Scenario: As a child, I want to see the total number of chores assigned to me and track their completion status.
Preconditions:

- The child has an active user account.
- The child has been assigned chores by the parent.

Test Steps:

1. Child logs into their account.
2. Child navigates to the chores list section on the dashboard.
3. Child views the total number of assigned chores.
4. Child checks the breakdown of chores by status (completed, pending).
5. Child confirms the accuracy of the chore count on the GUI and status.

Expected Results:

- The child can view the total number of chores assigned to them.
- The breakdown of chores by status (completed, pending) is displayed accurately.
- The chore count matches the actual number of assigned chores, and the status reflects their current completion state.
- Database reflects this as well.

9) Parent can rate their child’s work once the chore is completed. A box can appear asking to rate their child’s work on a scale of 1-5 after each chore is payed for.(William)

Test 1:

1. Login:
   - Nathan creates a new parent account with the username 'Will1001' and password 'willaim001'. Then, Jonathan, Nathan's child, creates a new child account with the username 'Jonathan111' and password 'Jonny010'.

2. Assign Chores:
   - Nathan, the parent, assigns a chore to Jonathan. He creates a chore named "clean dishes" with details: category - kitchen, duration - 20 min, payment - $12, and sets a deadline of 1 day from the current moment.
   - Nathan adds Jonathan to his parent account and selects the chore "clean dishes" from the central panel displaying all chores. He then clicks on the "Assign Chore" button to assign it to Jonathan.

3. Chore Completion:
   - Jonathan logs into his child account and sees the assigned chore "clean dishes" in his chore list.
   - Jonathan efficiently completes the chore within the same day it is assigned.

4. Chore Rating:
   - After Jonathan completes the chore, Nathan, the parent, logs into his parent account and views the completed chore.
   - Nathan evaluates the quality of Jonathan's performance and rates the completed chore on a scale of 1 to 5, where 1 indicates poor performance and 5 indicates excellent performance.
   - Nathan selects an appropriate rating reflecting how well Jonathan completed the "clean dishes" chore, considering factors like thoroughness, timeliness, and effort. He submits the rating to provide feedback to Jonathan on his performance.
   - Jonathan can view the rating provided by Nathan, which serves as feedback on his completed chore.

Test 2:

1. Login:
   - Parent 'Emily99' creates a new parent account with the username 'Emily99' and password 'emilypass123'. Then, her child 'Jacob111' creates a new child account with the username 'Jacob111' and password 'jacobpass456'.

2. Assign Chores:

- Emily, the parent, assigns a chore to Jacob. She creates a chore named "take out the trash" with details: category - household, duration - 15 min, payment - $8, and sets a deadline of 2 days from the current date.
- Emily adds Jacob to her parent account and selects the chore "take out the trash" from the list of available chores. She then clicks on the "Assign Chore" button to assign it to Jacob.

3. Chore Completion:
- Jacob logs into his child account and finds the assigned chore "take out the trash" in his chore list.
- Jacob promptly completes the chore within the assigned deadline.

4. Chore Rating:
- After Jacob completes the chore, Emily, the parent, logs into her parent account and checks the completed chore details.
- Emily evaluates Jacob's performance and provides a rating for the completed chore on a scale of 1 to 5, where 1 indicates poor performance and 5 indicates excellent performance.
- Emily carefully selects a rating that reflects Jacob's effort and effectiveness in completing the "take out the trash" chore. She submits the rating to provide constructive feedback to Jacob.
- Jacob views the rating assigned by Emily, which helps him understand how well he performed the chore and allows him to improve in the future.