

Parent Chore System

accountsModule

choresModule

competitionModule

UserInterface

tests

Account

-username :String
-password :String

+Account
+getUsername
+getPassword
+authenticate

DatabaseSetup

main

Chores

-name :String
-category :String
-time :Double
-payment :Double
-completed :Boolean
-paid :Boolean

+Chore
+getName
+getCategory
+setCategory
+getTime
+setTime
+getPayment
+setPayment
+isCompleted
+markCompleted
+isPaid
+markPaid
+toString

ChoresDatabase

main

ChoresUtils

+sortAlphabetically
+sortByTime
+sortByPayment
+filterCompleted
+filterNotCompleted
+filterPaid
+filterNotPaid
+filterCategory
+createExcelSheet

ChoresList

-chores :List

+ChoresList
+addChore
+removeChore
+getChores
+getNumberOfChores

Competition

-parent :parentAccount
-child1 :ChildAccount
-child2 :ChildAccount
-CHILD1_HOURS_TILL_NOW :double
-CHILD2_HOURS_TILL_NOW :double
-chore :Chore

+Competition
+getWinner

ChildAccountGUI

-chores :JTextField
-checkBalanceButton :JButton
-logoutButton :JButton
-childAccount :ChildAccount
-parentAccount :ParentAccount

+ChildAccountGUI(childAccount : ChildAccount)
-initialize(): void
-checkBalance(): void
+main(args :String[]): void

accountsAndChoresTest

-parentAccount : ParentAccount
-childAccount1 : ChildAccount
-childAccount2 : ChildAccount
-chore1 : Chore
-chore2 : Chore
-chore3 : Chore
-chore4 : Chore

+setUp(): void
+tearDown(): void
+testEditChore(): void
+testVerifyChores(): void

ChildAccount

-parent :ParentAccount
-Chores :List
#balance :Double
#hoursWorked :Double

+ChildAccount
+markChoreCompleted
+setParent
+getParent
+getChores
+getBalance
+getHoursWorked
+toString

ChildRegistration

-firstNameField :JTextField
-lastNameField :JTextField
-usernameField :JTextField
-passwordField :JPasswordField
-ageField :JTextField
-registerButton :JButton
-backButton :JButton

+ChildRegistration()
-initialize(): void
-actionPerformed(ActionEvent): void
-saveAccount(Account): void
+main(String[]): void

CompetitionStandings

- back :JButton
- goBack :JButton
- parent :ParentAccount
- child :ChildAccount
- childList :ArrayList<ChildAccount>
- table :JTable

CompetitionStandings(parent :ParentAccount, childList :ArrayList<ChildAccount>, child :ChildAccount)
-competition(): void
-actionPerformed(ActionEvent): void

choresSortTestCases

+testSortAlphabetically(): void
+testSortByTime(): void
+testSortByPayment(): void

CompetitionTest

-parent :ParentAccount
-child1 :ChildAccount
-child2 :ChildAccount
-chore :Chore
-child1HoursAtStart :double
-child2HoursAtStart :double

+setUp(): void
+tearDown(): void
+testGetWinner_Child1Wins(): void
+testGetWinner_Child2Wins(): void

ParentAccount

-childAccounts :List

+ParentAccount
+addChildAccount
+deleteChildAccount
+assignChore
+deleteChore
+getChildren
+depositChorePayment
+verifyChores
+checkChildBalance
+editChore

LoginPage

-usernameField :JTextField
-passwordField :JPasswordField
-loginButton :JButton
-parent :ParentRegistration
-backButton :JButton

+LoginPage()
-initialize(): void
-actionPerformed(ActionEvent): void
-authenticate(String, String): boolean
-openParentAccountGUI(): void
-openChildAccountGUI(): void

MainPage

-parentRegisterButton :JButton
-childRegisterButton :JButton
-loginButton :JButton

+MainPage()
-initialize(): void
-actionPerformed(ActionEvent): void
+main(String[]): void

ParentRegistration

- firstNameField :JTextField
- lastNameField :JTextField
- usernameField :JTextField
- passwordField :JPasswordField
- numberOfChildrenField :JTextField
- registerButton :JButton
- backButton :JButton

+ ParentRegistration()
+ actionPerformed(ActionEvent): void
- saveAccount(ParentAccount): void
+ main(String[]): void

ParentAccountGUI

- parentAccount :ParentAccount |
- childDropDown :JComboBox<ChildAccount>
- choreNameField :JTextField
- choreCategoryField :JTextField
- choreTimeField :JTextField
- chorePaymentField :JTextField
- createChoreButton :JButton
- checkBalanceButton :JButton
- addChildButton :JButton
- logoutButton :JButton

+ ParentAccountGUI(ParentAccount)
- initialize(): void
- addChild(): void
- createChore(): void
- checkBalance(): void
+ main(String[]): void