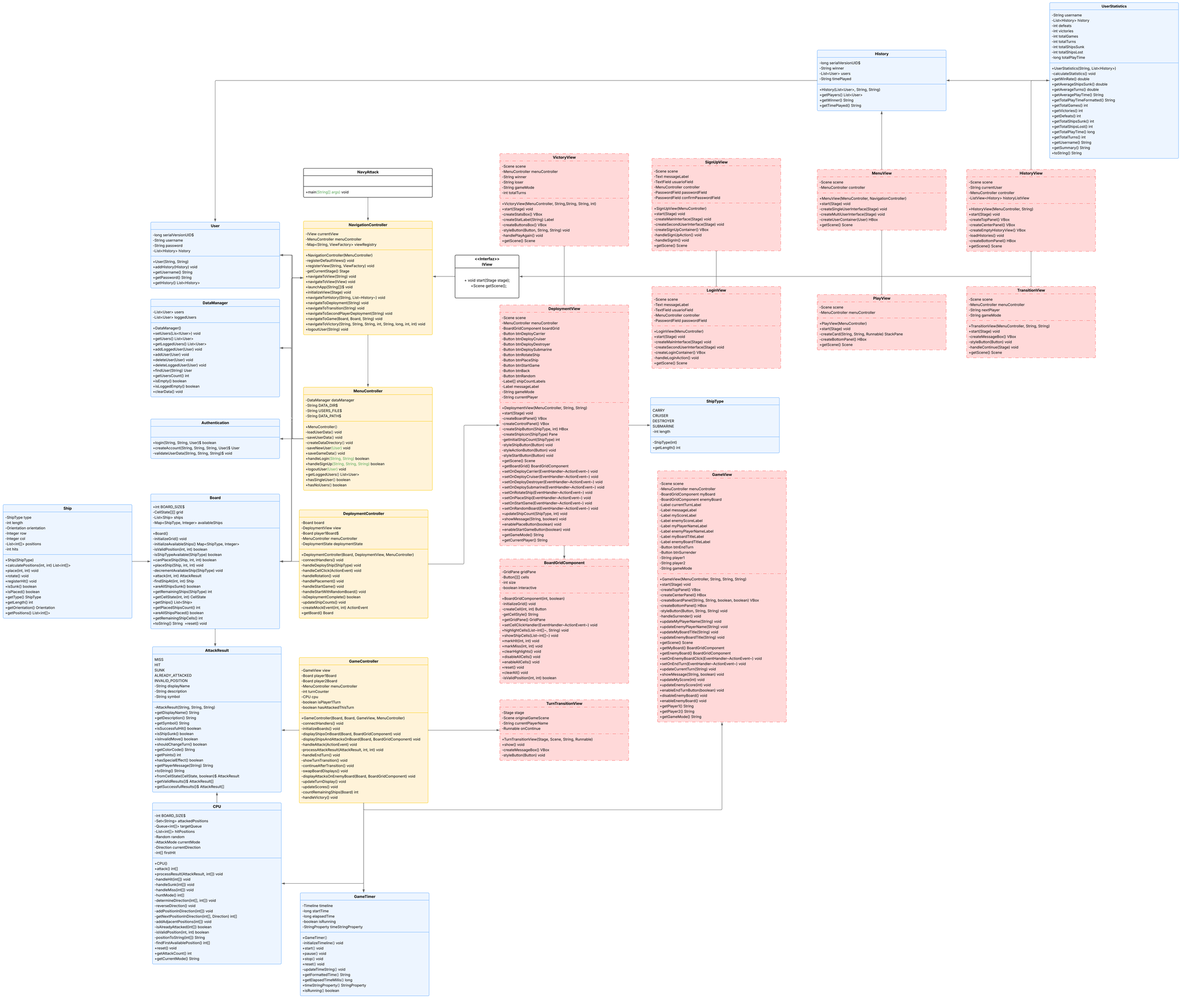
UserStatistics
-String username
-List<History> history
-int defeats
-int victories
-int totalGames
-int totalTurns
-int totalShipsSunk
-int totalShipsLost
-long totalPlayTime
+UserStatistics(String, List<History>)
-calculateStatistics() void
+getWinRate() double
+getAverageShipsSunk() double
+getAverageTurns() double
+getAveragePlayTime() String
+getTotalPlayTimeFormatted() String
+getTotalGames() int
+getVictories() int
+getDefeats() int
+getTotalShipsSunk() int
+getTotalShipsLost() int
+getTotalPlayTime() long
+getTotalTurns() int
+getUsername() String
+getSummary() String
+toString() String
History
-long serialVersionUID$
-String winner
-List<User> users
-String timePlayed
+History(List<User>, String, String)
+getPlayers() List<User>
+getWinner() String
+getTimePlayed() String
NavyAttack
+main(String[] args) void
VictoryView
-Scene scene
-MenuController menuController
-String winner
-String loser
-String gameMode
-int totalTurns
+VictoryView(MenuController, String,String, String, int)
+start(Stage) void
-createStatsBox() VBox
-createStatLabel(String) Label
-createButtonsBox() VBox
-styleButton(Button, String, String) void
-handlePlayAgain() void
+getScene() Scene
SignUpView
-Scene scene
-Text messageLabel
-TextField usuarioField
-MenuController controller
-PasswordField passwordField
-PasswordField confirmPasswordField
+SignUpView(MenuController)
+start(Stage) void
-createMainInterface(Stage) void
-createSecondUserInterface(Stage) void
-createSignUpContainer() VBox
-handleSignUpAction() void
-handleSignIn() void
+getScene() Scene
MenuView
-Scene scene
-MenuController controller
+MenuView(MenuController, NavigationController)
+start(Stage) void
-createSingleUserInterface(Stage) void
-createMultiUserInterface(Stage) void
-createUserContainer(User) HBox
+getScene() Scene
HistoryView
-Scene scene
-String currentUser
-MenuController controller
-ListView<History> historyListView
+HistoryView(MenuController, String)
+start(Stage) void
-createTopPanel() VBox
-createCenterPanel() VBox
-createEmptyHistoryView() VBox
-loadHistories() void
-createBottomPanel() HBox
+getScene() Scene
User
-long serialVersionUID$
-String username
-String password
-List<History> history
+User(String, String)
+addHistory(History) void
+getUsername() String
+getPassword() String
+getHistory() List<History>
NavigationController
-IView currentView
-MenuController menuController
-Map<String, ViewFactory> viewRegistry
+NavigationController(MenuController)
-registerDefaultViews() void
+registerView(String, ViewFactory) void
-getCurrentStage() Stage
+navigateToView(String) void
+navigateToView(IView) void
+launchApp(String[])$ void
+initializeView(Stage) void
+navigateToHistory(String, List~History~) void
+navigateToDeployment(String) void
+navigateToTransition(String) void
+navigateToSecondPlayerDeployment(String) void
+navigateToGame(Board, Board, String) void
+navigateToVictory(String, String, String, int, String, long, int, int) void
+logoutUser(String) void
<<Interfaz>>
IView
+ void start(Stage stage);
+Scene getScene();
LoginView
-Scene scene
-Text messageLabel
-TextField usuarioField
-MenuController controller
-PasswordField passwordField
+LoginView(MenuController)
+start(Stage) void
-createMainInterface(Stage) void
-createSecondUserInterface(Stage) void
-createLoginContainer() VBox
-handleLoginAction() void
+getScene() Scene
PlayView
-Scene scene
-MenuController menuController
+PlayView(MenuController)
+start(Stage) void
-createCard(String, String, Runnable) StackPane
-createBottomPanel() HBox
+getScene() Scene
TransitionView
-Scene scene
-MenuController menuController
-String nextPlayer
-String gameMode
+TransitionView(MenuController, String, String)
+start(Stage) void
-createMessageBox() VBox
-styleButton(Button) void
-handleContinue(Stage) void
+getScene() Scene
DataManager
-List<User> users
-List<User> loggedUsers
+DataManager()
+setUsers(Lis<tUser>) void
+getUsers() List<User>
+getLoggedUsers() List<User>
+addLoggedUser(User) void
+addUser(User) void
+deleteUser(User) void
+deleteLoggedUser(User) void
+findUser(String) User
+getUsersCount() int
+isEmpty() boolean
+isLoggedEmpty() boolean
+clearData() void
DeploymentView
-Scene scene
-MenuController menuController
-BoardGridComponent boardGrid
-Button btnDeployCarrier
-Button btnDeployCruiser
-Button btnDeployDestroyer
-Button btnDeploySubmarine
-Button btnRotateShip
-Button btnPlaceShip
-Button btnStartGame
-Button btnBack
-Button btnRandom
-Label[] shipCountLabels
-Label messageLabel
-String gameMode
-String currentPlayer
+DeploymentView(MenuController, String, String)
+start(Stage) void
-createBoardPanel() VBox
-createControlPanel() VBox
-createShipButton(ShipType, int) HBox
-createShipIcon(ShipType) Pane
-getInitialShipCount(ShipType) int
-styleShipButton(Button) void
-styleActionButton(Button) void
-styleStartButton(Button) void
+getScene() Scene
+getBoardGrid() BoardGridComponent
+setOnDeployCarrier(EventHandler~ActionEvent~) void
+setOnDeployCruiser(EventHandler~ActionEvent~) void
+setOnDeployDestroyer(EventHandler~ActionEvent~) void
+setOnDeploySubmarine(EventHandler~ActionEvent~) void
+setOnRotateShip(EventHandler~ActionEvent~) void
+setOnPlaceShip(EventHandler~ActionEvent~) void
+setOnStartGame(EventHandler~ActionEvent~) void
+setOnRandomBoard(EventHandler~ActionEvent~) void
+updateShipCount(ShipType, int) void
+showMessage(String, boolean) void
+enablePlaceButton(boolean) void
+enableStartGameButton(boolean) void
+getGameMode() String
+getCurrentPlayer() String
ShipType
CARRY
CRUISER
DESTROYER
SUBMARINE
-int length
-ShipType(int)
+getLength() int
MenuController
-DataManager dataManager
-String DATA_DIR$
-String USERS_FILE$
-String DATA_PATH$
+MenuController()
-loadUserData() void
-saveUserData() void
-createDataDirectory() void
-saveNewUser(User) void
+saveGameData() void
+handleLogin(String, String) boolean
+handleSignUp(String, String, String) boolean
+logoutUser(User) void
+getLoggedUsers() List<User>
+hasSingleUser() boolean
+hasNoUsers() boolean
Authentication
+login(String, String, User)$ boolean
+createAccount(String, String, String, User)$ User
-validateUserData(String, String, String)$ void
GameView
-Scene scene
-MenuController menuController
-BoardGridComponent myBoard
-BoardGridComponent enemyBoard
-Label currentTurnLabel
-Label messageLabel
-Label myScoreLabel
-Label enemyScoreLabel
-Label myPlayerNameLabel
-Label enemyPlayerNameLabel
-Label myBoardTitleLabel
-Label enemyBoardTitleLabel
-Button btnEndTurn
-Button btnSurrender
-String player1
-String player2
-String gameMode
+GameView(MenuController, String, String, String)
+start(Stage) void
-createTopPanel() VBox
-createCenterPanel() HBox
-createBoardPanel(String, String, boolean, boolean) VBox
-createBottomPanel() HBox
-styleButton(Button, String, String) void
-handleSurrender() void
+updateMyPlayerName(String) void
+updateEnemyPlayerName(String) void
+updateMyBoardTitle(String) void
+updateEnemyBoardTitle(String) void
+getScene() Scene
+getMyBoard() BoardGridComponent
+getEnemyBoard() BoardGridComponent
+setOnEnemyBoardClick(EventHandler~ActionEvent~) void
+setOnEndTurn(EventHandler~ActionEvent~) void
+updateCurrentTurn(String) void
+showMessage(String, boolean) void
+updateMyScore(int) void
+updateEnemyScore(int) void
+enableEndTurnButton(boolean) void
+disableEnemyBoard() void
+enableEnemyBoard() void
+getPlayer1() String
+getPlayer2() String
+getGameMode() String
Board
+int BOARD_SIZE$
-CellState[][] grid
-List<Ship> ships
-Map<ShipType, Integer> availableShips
+Board()
-initializeGrid() void
-initializeAvailableShips() Map<ShipType, Integer>
-isValidPosition(int, int) boolean
-isShipTypeAvailable(ShipType) boolean
+canPlaceShip(Ship, int, int) boolean
+placeShip(Ship, int, int) void
-decrementAvailableShip(ShipType) void
+attack(int, int) AttackResult
-findShipAt(int, int) Ship
+areAllShipsSunk() boolean
+getRemainingShips(ShipType) int
+getCellState(int, int) CellState
+getShips() List<Ship>
+getPlacedShipsCount() int
+areAllShipsPlaced() boolean
+getRemainingShipCells() int
+toString() String +reset() void
Ship
-ShipType type
-int length
-Orientation orientation
-Integer row
-Integer col
-List<int[]> positions
-int hits
+Ship(ShipType)
+calculatePositions(int, int) List<int[]>
+place(int, int) void
+rotate() void
+registerHit() void
+isSunk() boolean
+isPlaced() boolean
+getType() ShipType
+getLength() int
+getOrientation() Orientation
+getPositions() List<int[]>
DeploymentController
-Board board
-DeploymentView view
-Board player1Board$
-MenuController menuController
-DeploymentState deploymentState
+DeploymentController(Board, DeploymentView, MenuController)
-connectHandlers() void
-handleDeployShip(ShipType) void
-handleCellClick(ActionEvent) void
-handleRotation() void
-handlePlacement() void
-handleStartGame() void
-handleStartWithRandomBoard() void
-isDeploymentComplete() boolean
-updateShipCounts() void
-createMockEvent(int, int) ActionEvent
+getBoard() Board
BoardGridComponent
-GridPane gridPane
-Button[][] cells
-int size
-boolean interactive
+BoardGridComponent(int, boolean)
-initializeGrid() void
-createCell(int, int) Button
-getCellStyle() String
+getGridPane() GridPane
+setCellClickHandler(EventHandler~ActionEvent~) void
+highlightCells(List~int[]~, String) void
+showShipCells(List~int[]~) void
+markHit(int, int) void
+markMiss(int, int) void
+clearHighlights() void
+disableAllCells() void
+enableAllCells() void
+reset() void
+clearAll() void
-isValidPosition(int, int) boolean
AttackResult
MISS
HIT
SUNK
ALREADY_ATTACKED
INVALID_POSITION
-String displayName
-String description
-String symbol
-AttackResult(String, String, String)
+getDisplayName() String
+getDescription() String
+getSymbol() String
+isSuccessfulHit() boolean
+isShipSunk() boolean
+isInvalidMove() boolean
+shouldChangeTurn() boolean
+getColorCode() String
+getPoints() int
+hasSpecialEffect() boolean
+getPlayerMessage(String) String
+toString() String
+fromCellState(CellState, boolean)$ AttackResult
+getValidResults()$ AttackResult[]
+getSuccessfulResults()$ AttackResult[]
GameController
-GameView view
-Board player1Board
-Board player2Board
-MenuController menuController
-int turnCounter
-CPU cpu
-boolean isPlayer1Turn
-boolean hasAttackedThisTurn
+GameController(Board, Board, GameView, MenuController)
-connectHandlers() void
-initializeBoards() void
-displayShipsOnBoard(Board, BoardGridComponent) void
-displayShipsAndAttacksOnBoard(Board, BoardGridComponent) void
-handleAttack(ActionEvent) void
-processAttackResult(AttackResult, int, int) void
-handleEndTurn() void
-showTurnTransition() void
-continueAfterTransition() void
-swapBoardDisplays() void
-displayAttacksOnEnemyBoard(Board, BoardGridComponent) void
-updateTurnDisplay() void
-updateScores() void
-countRemainingShips(Board) int
-handleVictory() void
TurnTransitionView
-Stage stage
-Scene originalGameScene
-String currentPlayerName
-Runnable onContinue
+TurnTransitionView(Stage, Scene, String, Runnable)
+show() void
-createMessageBox() VBox
-styleButton(Button) void
CPU
-int BOARD_SIZE$
-Set<String> attackedPositions
-Queue<int[]> targetQueue
-List<int[]> hitPositions
-Random random
-AttackMode currentMode
-Direction currentDirection
-int[] firstHit
+CPU()
+attack() int[]
+processResult(AttackResult, int[]) void
-handleHit(int[]) void
-handleSunk(int[]) void
-handleMiss(int[]) void
-huntMode() int[]
-determineDirection(int[], int[]) void
-reverseDirection() void
-addPositionInDirection(int[]) void
-getNextPositionInDirection(int[], Direction) int[]
-addAdjacentPositions(int[]) void
-isAlreadyAttacked(int[]) boolean
-isValidPosition(int, int) boolean
-positionToString(int[]) String
-findFirstAvailablePosition() int[]
+reset() void
+getAttackCount() int
+getCurrentMode() String
GameTimer
-Timeline timeline
-long startTime
-long elapsedTime
-boolean isRunning
-StringProperty timeStringProperty
+GameTimer()
-initializeTimeline() void
+start() void
+pause() void
+stop() void
+reset() void
-updateTimeString() void
+getFormattedTime() String
+getElapsedTimeMillis() long
+timeStringProperty() StringProperty
+isRunning() boolean