**This class diagram illustrates the design of a bank account system, which includes different classes such as Account, UnderageAccount, Parent, Transaction, SavingsGoal, and Notification. Below is a detailed explanation of each class and its relationships:**

### ****Classes and Attributes****

1. ****Account****
   1. ****Attributes:****

**. accountNumber: String: A unique identifier for the account.**

**. balance: double: The current balance in the account.**

**. owner: String: The name of the account holder.**

* 1. ****Methods:****

**. createAccount(): Initializes an account.**

**. deposit(amount: double): Adds the specified amount to the account balance.**

**. withdraw(amount: double): Deducts the specified amount from the balance.**

**. getAccountDetails(): Returns information about the account.**

1. ****Parent****
   1. ****Attributes:****

**. parentName: String: Name of the parent/guardian.**

**. relationship: String: Describes the parent-child relationship.**

**. contactDetails: String: Parent's contact information.**

****2 .Methods:****

**. approveTransaction(transaction: Transaction): Approves a transaction made by an underage account.**

**. receiveNotification(message: String): Receives alerts or notifications about transactions.**

1. ****UnderageAccount (inherits from Account)****
   1. ****Attributes:****

**. guardian: Parent: Reference to the associated parent or guardian.**

**. age: int: Age of the account holder.**

**. withdrawalLimit: double: Maximum allowed withdrawal limit.**

**. savingsGoal: double: Target savings goal for the account holder.**

* 1. ****Methods:****

**. setWithdrawalLimit(limit: double): Sets the withdrawal limit.**

**. approveWithdraw(): Checks if a withdrawal is approved.**

**. isUnderage(): Determines whether the account holder is underage.**

**. notifyGuardian(message: String): Sends a notification to the guardian.**

1. ****Transaction****
   1. ****Attributes:****

**. transactionId: String: Unique transaction identifier.**

**. amount: double: Amount involved in the transaction.**

**. transactionDate: Date: Date of the transaction.**

**. transactionType: String: Type of the transaction (e.g., withdrawal, deposit).**

**. status: String: Status of the transaction (e.g., completed, pending).**

* 1. ****Methods:****

**. executeTransaction(): Carries out the transaction.**

**. cancelTransaction(): Cancels a transaction if needed.**

1. ****SavingsGoal****
   1. ****Attributes:****

**. goalName: String: Name of the savings goal.**

**. targetAmount: double: Desired savings target.**

**. currentAmount: double: Current amount saved toward the target.**

* 1. ****Methods:****

**. setGoal(target: double): Sets a new savings target.**

**. updateProgress(amount: double): Updates the saved amount.**

**. checkGoalCompletion(): Checks if the goal has been reached.**

1. ****Notification****
   1. ****Attributes:****

**. notificationId: String: Unique identifier for the notification.**

**. message: String: Notification message.**

**. dateSent: Date: Date the notification was sent.**

* 1. ****Methods:****

**. sendNotification(recipient: Parent): Sends a notification to the parent.**

### ****Relationships****

1. ****Inheritance:****

**. UnderageAccount is a subclass of Account.**

1. ****Associations:****

**. UnderageAccount is associated with one Parent (guardian).**

**. UnderageAccount manages multiple Transaction instances.**

**. UnderageAccount can set one SavingsGoal.**

**. Parent receives multiple Notification objects.**

1. ****Multiplicity:****

**. Account has zero or more transactions.**

**. UnderageAccount must have one guardian.**

**. Notification can be sent to multiple parents.**

### ****Conclusion****

**This class diagram models a structured and secure bank account system for both regular and underage users. It integrates parental oversight, transaction management, and goal-oriented savings while facilitating secure communication through notifications.**