The use case diagram represents the components and relationships in the "Underage Saving Bank System," designed to encourage financial literacy and parental oversight for children. Below is a detailed description:

**Actors**

* **Child User:** Represents the underage user who interacts with the system to set goals, track savings, and complete financial lessons.
* **Parent User:** Represents the parent who manages and monitors the child’s financial activities.

**Functional Modules**

1. **Child User Functions:**
   * **Set Savings Goals:** Allows children to define savings targets.
   * **Track Progress:** Provides progress tracking toward savings goals.
   * **Complete Interactive Lessons:** Offers educational content on financial literacy.
   * **Earn Rewards:** Gamified feature to reward children for completing lessons and achieving goals.
   * **Request Transaction:** Enables children to request financial transactions.
   * **View Dashboard:** Displays account summaries and progress insights.
2. **Parent User Functions:**
   * **Create Child Account:** Allows parents to set up an account for the child.
   * **Set Spending Limits:** Enables parents to control the spending limits of their children.
   * **Approve Transactions:** Provides parental consent for financial transactions.
   * **Monitor Activity:** Enables parents to review and track the child’s activities.
   * **Generate Reports:** Produces detailed activity reports for better insights.
   * **Add Encouragement:** Allows parents to send motivational messages to children.
3. **System Security and Notifications:**
   * **Process Secure Transactions:** Handles safe and secure financial operations.
   * **Encrypt Sensitive Data:** Ensures data privacy through encryption.
   * **Verify Identity:** Provides secure identity verification for users.
   * **Send Notifications:** Delivers notifications for updates, approvals, and milestones.

**Relationships**

* **Primary Associations:** Direct connections between actors and functions based on user roles.
* **Include Relationships:** Functional dependencies where one component uses another (e.g., account creation includes data encryption).
* **Extend Relationships:** Optional functionalities activated under specific scenarios (e.g., earning rewards extends completing lessons).

This system design ensures a secure, educational, and interactive environment for children to develop financial habits under parental guidance.