

## 4. UML Diagrams

### 4.6 Object Diagrams

Object diagrams show instances of classes and their relationships at a specific point in time, illustrating the system's state before and after key operations. They differ from class diagrams in that they show specific instances rather than general structures. Each object is an instance of a class with specific attribute values.

FalsoPay uses pre-condition and post-condition object diagrams to visualize the system state changes during key operations:

#### 4.6.1 User Registration

**Pre-condition:** Shows the system state before a new user registers with FalsoPay.

- The system contains only the base User type
- No user account exists for the email being registered
- The authentication system is prepared to validate the registration

**Post-condition:** Shows the system after successful user registration.

- A new User object exists with the provided information
- The system has generated a unique ID for the user
- Default account settings are established
- Authentication credentials are stored securely

#### 4.6.2 Send Money

**Pre-condition:** Depicts the system state before a money transfer.

- Sender user has sufficient balance
- Both sender and recipient accounts are active
- Transaction limits have not been exceeded

**Post-condition:** Shows the system after a successful money transfer.

- Sender balance has decreased by the transfer amount
- Recipient balance has increased by the transfer amount
- A transaction record has been created
- Notifications have been triggered for both parties

#### 4.6.3 Link Bank Account

**Pre-condition:** Illustrates the system state before linking a bank account.

- User exists in the system
- Bank account is not yet linked to the user
- Bank verification system is ready

**Post-condition:** Shows the state after successful bank account linking.

- Bank account is now associated with the user
- Bank verification status is updated
- User has access to the linked account functionality

#### 4.6.4 Request Money

**Pre-condition:** Shows the system state before a money request.

- Requester and recipient users exist
- No pending request exists between these users for this amount

**Post-condition:** Depicts the system after a money request is created.

- A new MoneyRequest object exists
- Notification has been sent to the recipient
- Request appears in both users' activity logs

#### 4.6.5 Support Ticket

**Pre-condition:** Shows the system before a support ticket is created.

- User exists in the system
- Support system is operational

**Post-condition:** Illustrates the system after a support ticket is submitted.

- A new SupportTicket object exists with the user's issue
- Ticket has been assigned a unique ID
- Initial status is set to "Open"
- Notification has been sent to support staff

#### 4.6.6 Block App Transactions

**Pre-condition:** Shows the system state before a user blocks transactions.

- User account is active
- Transaction capability is currently enabled
- User is properly authenticated

**Post-condition:** Depicts the system after transactions are blocked.

- User's TransactionStatus is set to "Blocked"
- SecurityLog entry has been created documenting the change
- Notification has been sent to the user confirming the action
- Any pending outgoing transactions are canceled

These object diagrams serve as valuable tools for:

1. Validating that operations produce the expected results
2. Understanding the complex relationships between objects at runtime
3. Documenting the concrete effects of system operations
4. Guiding implementation of business logic

The diagrams are created using PlantUML and can be viewed using the PlantUML Web Server, IDE plugins, or the included HTML viewer.