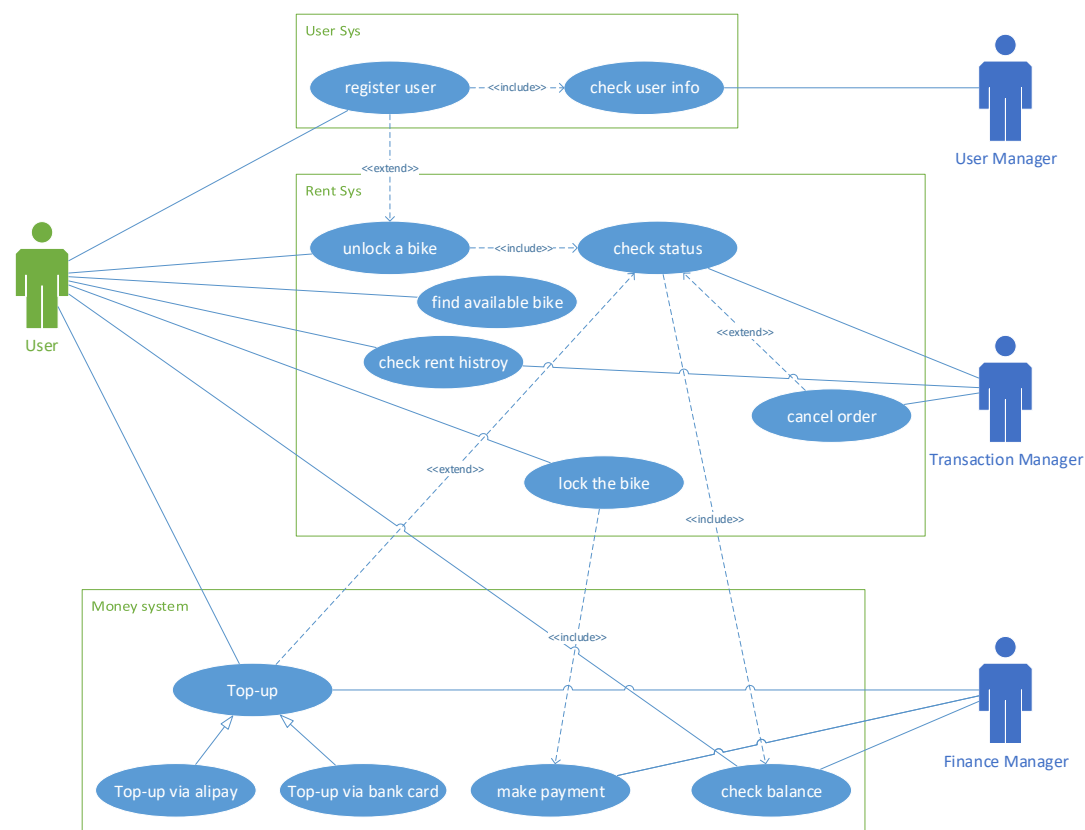


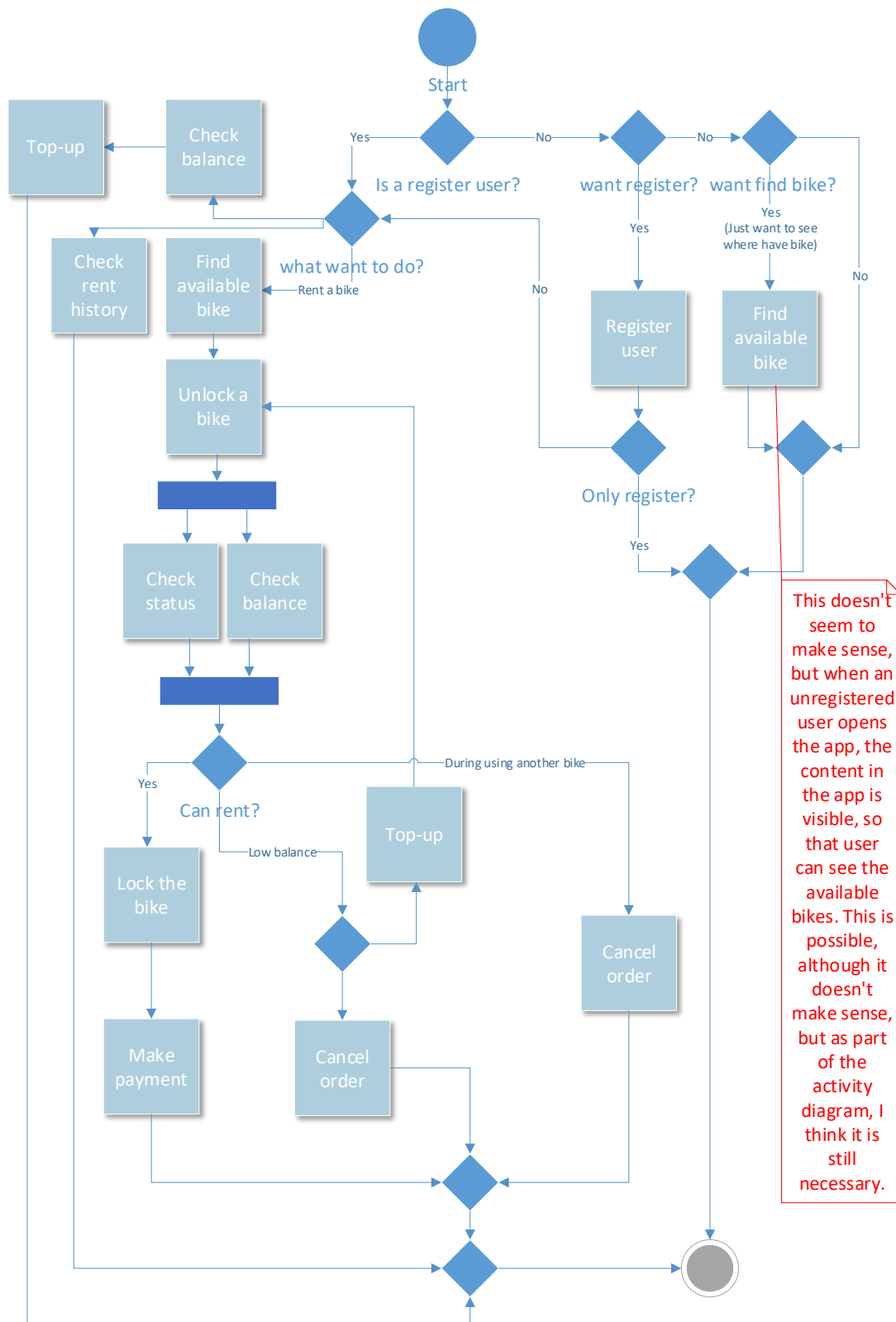
Task 1: Shared Bike Customer Management System

The responsibility of the system is to track the customers and transactions of the shared bicycle service provider. Shared bicycle service providers have user managers, transaction managers, and financial managers. Users can view available bicycles. User managers will check if the user information meets the specifications when the user registers. Registered users can rent a bike, top-up, view balance and rental history. The bicycle rental needs to check the user status through the a transaction manager. If the user is using another bicycle, the order is cancelled by the transaction manager. If the user has insufficient balance, the user can choose to recharge to continue or cancel the order. If the rental bicycle is successful(unlock a bike), pay back after returning the bicycle(lock the bike). All financial operations must be carried out through a financial manager.

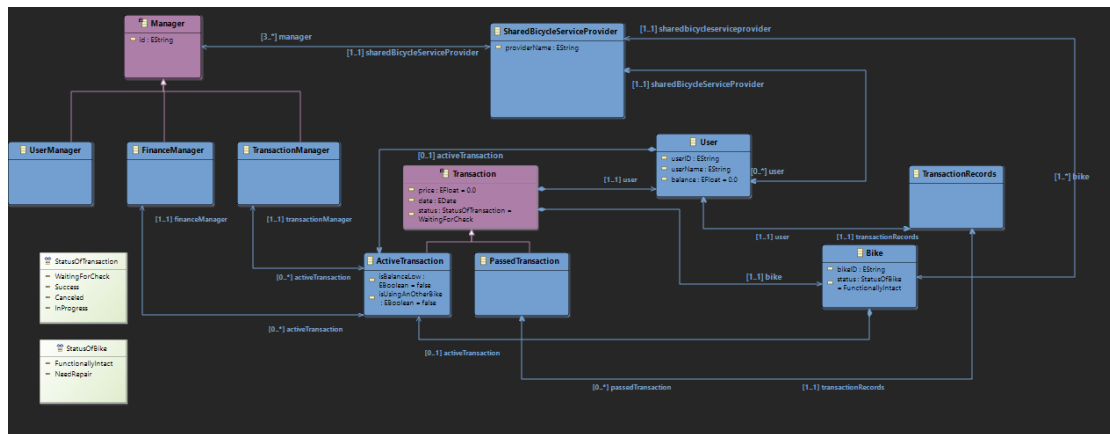
Task 2a: Use-Case Diagram



Task 2b: Activity Diagram



Task 3a: Class Diagram



Task 3b: Business Constraints

One bicycle can only be used by one person at a time (If the bike have an active transaction, then other user cannot rent that bike)

If the user does not have a transaction in progress (Active transaction), the user can start a new transaction (rent a bike).

1. The balance in user must large than the price in an active transaction (because user make payment after using the bike).
2. The status of "PassedTransaction" must be "Success" or "Canceled".
3. The status of an "ActiveTransaction" must be "InProgress"
4. The status of a bike in an "ActiveTransaction" must be "FunctionallyIntact"

context User inv:

```
self.transactionRecords.passedTransaction->forAll(status =
StatusOfTransaction::Success or status = StatusOfTransaction::Canceled)
```

and

```
self.activeTransaction->notEmpty() implies (self.activeTransaction.price <
self.balance and self.activeTransaction.status = StatusOfTransaction::InProgress
and self.activeTransaction.bike.status = StatusOfBike::FunctionallyIntact)
```

Task 3C: OD

