MoveMe

MoveMe is a ride sharing app similar to Uber and Lyft.

MoveMe keeps track of two types of agents: drivers and riders.

Riders can request rides and rate drivers (1 – 5 stars).

Drivers can refuse or accept requests, complete requests, and rate riders (1 – 5 stars).

Agent properties include an associated person with a name and phone number, an average rating (1 – 5 stars), and an account. The account has a balance that can be debited or credited. In addition, driver properties includes a driver status (logged out, available, engaged) and rider properties include a rider status (waiting, riding, inactive).

Ride requests are represented by request objects. Request properties include pick-up address, drop-off address, requesting passenger, driver who accepted the request (if any), cost, ETA, and a request status (pending, awarded, completed, aborted).

When a ride is requested a request object is created and forwarded to a nearby driver. The driver may refuse or accept the request. If the driver refuses, the same request is forwarded to another driver. If no driver can be found the rider is informed. If the driver accepts the request, the rider is informed of the driver's name, phone number, and ETA. Driver and rider statuses need to be updated appropriately.

When the ride is over the driver completes the request. This debits the riders account and credits the drivers account (by the cost of the ride). It also updates the statuses of the driver and rider.

At any time drivers and riders can rate each other. This automatically recomputes the average ratings of both.

Problem 1

Create a requirements model for MoveMe.

Some details you might want to consider:

- Think of use cases as functions performed by the MoveMe server. Never mind that these functions are called by client apps on agent phones.
- Are there opportunities to use include arrows?
- Are there opportunities to use generalization arrows?
- Be wary of extends arrows. Use these to indicate exception handling.
- When are riders and drivers secondary actors?
- Elaborate the *complete request* use case.

Problem 2

Create a domain model for MoveMe.

Some details you might want to consider:

- Are there opportunities to use analysis patterns? Actor-Role, Types as Objects, etc.
- Are there opportunities to use enumerations?
- Are there opportunities to use generalization arrows?
- Which properties should be association endpoints? Which should be class attributes? (NB: At least one end of each association needs an arrowhead, name, and multiplicity.)
- Are there opportunities to include class operations? (NB: bookkeeping operations like constructors, setters, getter, and delegators are not shown.)
- What are the properties of an address?
- What are the properties of an ETA?

Problem 3

Draw an activity diagram showing the process from requesting a ride to updating rider and driver ratings.

Home (cf9c8b720f3815adecc | All Diagrams (diagrams.html) | Index (element_index.html)

- **MoveMe (cf9c8**
  - Requirements
    - Usecase M
    - Domain Mo
  - Design Model

UseCaseDiagram1

UMLUseCaseDiagram

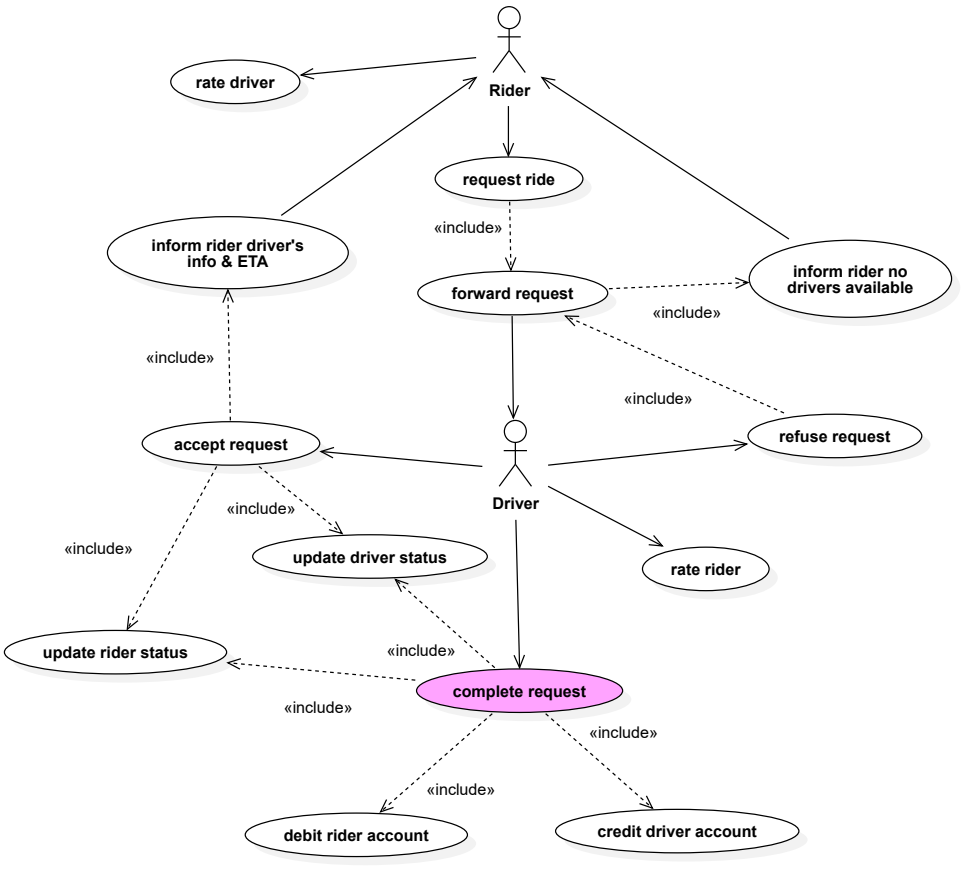
MoveMe (cf9c8b720f3815adeccaf3ef6e48c6c4.html) ::

Requirements Model (4108502ca9a7bbcdbbeb27e532b9b4d5.html) ::

Usecase Model (d6532bd44fbf3fd7c05fcf98d2c549c3.html) ::

UseCaseDiagram1 (40e14dde7f3432f10c59ed840ae66b10.html)

Diagram



Description

none

Properties

Name	Value
name	UseCaseDiagram1
defaultDiagram	false

Home (cf9c8b720f3815adecc | All Diagrams (diagrams.html) | Index (element_index.html)

- **MoveMe (cf9c8**
  - Requirements
    - Usecase M
    - Domain Mo
  - Design Model

ClassDiagram1

UMLClassDiagram

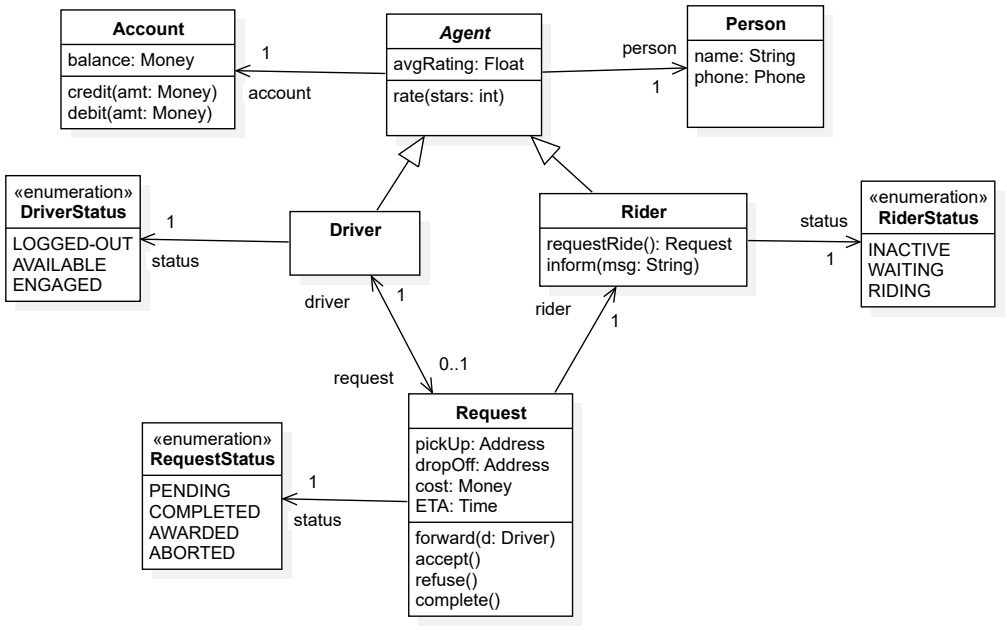
MoveMe (cf9c8b720f3815adeccaf3ef6e48c6c4.html) ::

Requirements Model (4108502ca9a7bbcdbbeb27e532b9b4d5.html) ::

Domain Model (02aeb6cb87c9703cd9ad89c4b5aed9d9.html) ::

ClassDiagram1 (44ad5c4ddff69f43098675f6c6fd5ead.html)

Diagram



Description

none

Properties

Name	Value
name	ClassDiagram1
defaultDiagram	false