Usecase: Initializing Marketplace System

Actor: SystemManager

Preconditions: System Integrity rules are fully implemented

Parameters: ExternalConnector

Actions:

1. System requests for a registration of the first user meant to be the SystemManager.
2. Requests a username and password.
3. System Creates a SystemManager User.
4. The user is recorded to the database.
5. System presents method of payment service pool.
6. User must select a minimum of one option from the pool.
7. System attempts to establish connections with the selected payments services.
8. Records chosen services in the database.
9. If fails, go back to step 5 (up to 3 times, else jump to step 16).
10. System presents supplies service pool.
11. User must select a minimum of one option from the pool.
12. System attempts to establish connections with the selected supplies services.
13. Records chosen services in the database.
14. If connection succeeds, System marketplace is now open.
15. If fails, go back to step 11 (3 times).
16. Request the user to Initialize in a different time, rollback recorded data.

Successful Scenario:

System successfully creates a system manager, connects to atleast one payment service, atleast one supplies service.

Marketplace is up and running.

Erroneous Scenario:

System created a user that is not a System manager.

System could not connect to any payment service.

System could not connect to any supplies service.

System could not record the new SystemManager\Payment services\ Supplies Services in the database.

Usecase: Changing connection with external services

Actor: System Manager

Preconditions: None

Postconditions: a minimum of one of each service type is connected.

Parameters: None

Actions:

1. System presents options of: “interchanging external services connection” and “adding external services connection”.
   1. If SystemManager chooses “interchanging external services connection”
   2. System asks user to choose services to connect to and presents a pool of available services.
   3. SystemManager chooses 1 service he wants to connect to.
   4. System asks if user is done selecting services to connect to.
   5. If not done, jump to 2.2
   6. System asks user to choose services to disconnect from and presents a pool of currently connected services.
   7. SystemManager chooses 1 service he wants to disconnect from.
   8. System asks if user is done selecting services to disconnect from.
   9. If not done, jump to 2.6.

2.10. Invoke “interchanging external services connection” using parameters obtained from and 2.3 and 2.8.

* 1. If successful, inform the user.

2.12 If fails inform the user and inform that no changes were made.

* 1. If SystemManager chooses “adding external services connection”
  2. System asks user to choose services to connect to and presents a pool of available services.
  3. SystemManager chooses 1 service he wants to connect to.
  4. System asks if user is done selecting services to connect to.
  5. If not done, jump to 3.2
  6. Invoke “adding external services connection” using parameters obtained from and 3.4.
  7. If successful, inform user.
  8. If fails inform user and inform that no changes were made.

Successful Scenario:

System presents user with options:

User selects “interchanging external services connection” and selects one or more services to connect to and 1 or more services to disconnect to. Invoking interchanging external services connection usecase is successful.

User selects “adding external services connection” and selects one or more services to connect to. Invoking adding external services connection usecase is successful.

Erroneous Scenario:

System presents user with options:

User selects “interchanging external services connection”:

selects one or more services to connect to and 1 or more services to disconnect to. Invoking interchanging external services connection usecase is unsuccessful.

Pool of services to connect to has services already currently connected.

Pool of services to disconnect from has services that aren’t currently connected.

User selects “adding external services connection”:

selects one or more services to connect to. Invoking adding external services connection usecase is unsuccessful.

Pool of services to connect to has services already currently connected.

Usecase: interchanging external services connection

Actor: SystemManager, Changing connection with external services usecase.

Preconditions: Minimum of one connection of each service.

Postconditions: Minimum of one connection of each service.

Parameters: collections of services to connect to, collections of services to disconnect from.

Actions:

1. System checks that the services to connect to and services to disconnect from do not overlap.
2. System checks that the sum of **distinct** “services to connect to + service currently connected” minus the “Services to disconnect from” is at minimum 1 for each of the services.
3. If both conditions are true, System iterates through services to connect to.

for each service that isn’t already connected:

* 1. Attempt connection with the service.
  2. If successful, record the service in database.
  3. If failed, rollback the entire process.

1. If both conditions are true, System iterates through services to disconnect from.

for each service that is connected:

* 1. Attempt to disconnect from the service.
  2. If successful, delete the service in database.
  3. If failed, rollback the entire process.

Successful Scenarios:

All services to connect to are currently not connected, and all services to disconnect from are currently connected.

For each service type: number of services to connect to + number of services connected > number of services to disconnect to.

System manages to connect to all services from “services to connect to” and manages to disconnect from all services from “services to disconnect from”.

Erroneous Scenarios:

some services to connect to are currently connected, or some services to disconnect from aren’t currently connected.

One of service types: number of services to connect to + number of services connected <= number of services to disconnect to.

System can’t connect to some services from “services to connect to”.

System can’t disconnect from all services from “services to disconnect from”.

Usecase: adding external services connection

Actor: SystemManager, Changing connection with external services usecase.

Preconditions: User is logged in and is SystemManager, Minimum of one connection of each service

Postconditions: Minimum of one connection of each service

Parameters: collections of services to connect to

Actions:

1. System iterates through services to connect to.

for each service that isn’t already connected:

* 1. Attempt connection with the service.
  2. If successful, record the service in database.
  3. If failed, rollback the entire process.

Successful Scenario:

All services to connect to are currently not connected.

System manages to connect to all services from “services to connect to”.

Erroneous Scenario:

some services to connect to are currently connected.

System can’t connect to some services from “services to connect to”.

Usecase: Payment

Actor: User, Checkout usecase.

Preconditions: User has at least 1 item in cart and is attempting to place an order,

Parameters: payment amount

Actions:

1. System presents a pool of available payment methods.
2. User selects one from of them from the pool.
3. System sends a request to an external service to complete payment for the amount specified, via the payment method chosen.
4. If payment completed successfully, record the payment in the database, and return the payment Id.
5. If unsuccessful, signal back the transaction was unsuccessful.

Successful Scenarios:

System presents all available payment methods; user selects one of them and is debited his payment due. Payment is successful. A distinct transaction number is returned.

Erroneous Scenario:

Some available payment methods are not presented.

User can select more than one payment method.

Payment is unsuccessful but transaction number still returns.

System is unable to connect to the selected payment service.

A different payment method is applied than the one chosen.

The amount the user was debited is different than the amount due.