**Application that allows our customers to create the order, do the payment and get the recurring billing**

1. REQUIREMENTS

* Customer (Art Director) to search for orders by name.
* Customer “checks out” the order → proceeds to payment.
* If payment is successful:
* Send email notification of success.
* Call Production Service to push the order into the production system.
* Create invoice in the billing system.
* If payment fails: display error.

1. ACTORS

* Art Director (User)
* System (App Server)
* Payment Service (Mock)
* Email Service (Mock)
* Production Service (Mock)
* Invoice Service / DB

1. USE CASES

* Search Order by name
* Checkout Order
* Make Payment (simulate success/failure)
* Push Order to Production Service (simulate success/failure)
* Send Email Notification (simulate success)
* Create Invoice in Invoice System

1. DOMAIN MODEL / ENTITIES

| **Entity** | **Fields** |
| --- | --- |
| User | id, name, email, password |
| Order | id, user\_id, name, status, total\_amount, created\_date |
| Payment | id, order\_id, amount, status (SUCCESS/FAILURE), timestamp |
| Invoice | id, order\_id, user\_id, amount, billing\_date, due\_date |

* Diagram

+---------+ +-------+ +---------+

| User | | Order | | Payment |

|---------| |-------| |---------|

| id (PK) |<-----+ | id(PK)|<-----+ | id (PK) |

| name | | | user\_id(FK) | | order\_id(FK) |

| email | | | name | | amount |

| password| | | status | | status |

+---------+ | | total\_amount| | timestamp|

| | created\_date| +---------+

| +-------+

|

|

v

+-----------+

| Invoice |

|------------|

| id (PK) |

| order\_id(FK)|

| user\_id(FK)|

| amount |

| billing\_date|

| due\_date |

+-----------+