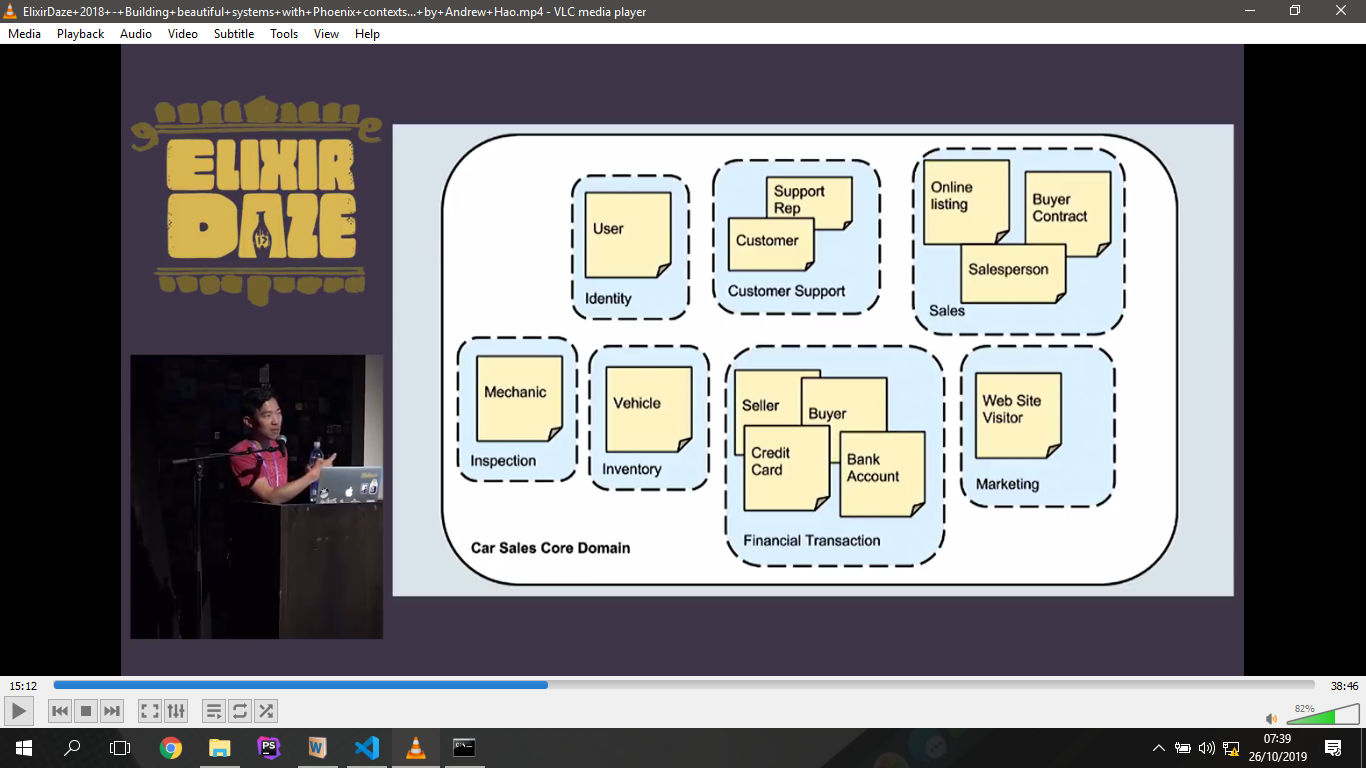
***SERVICE BOOKING ARCHITECTURE***

By: **Samuel Iso** (samdoxiso@gmail.com)

Company: ***Samsojac Concepts Services Nigeria Ltd*** | *Pandaville Labs*



|  |  |  |
| --- | --- | --- |
| **Sales**  Service\_Listing  Service\_Providers  Service\_Consumers  Service\_Orders | **Finance**  Seller/Merchant  Buyer  Wallet[Account]  Gift\_Card | **Identity**  User(Aggregrate root with Profile)  Marketing  Loyalty\_Rating |

Proposal -

Website for booking Session(Service) more details than mobile app

Payment Platform - Client can pay. Book and Pay on Website or In Saloon

Loyalty programme: booking confirmed - loyalty goes up per completion of service-

Gift Card -

Someone who pays online \_ someone can pay into virtual/wallet account

Mobile App - cost

SERVICE BOOKING ARCHITECTURE

BY: SAMUEL ISO (samdoxiso@gmail.com)

COMPANY: Samsojac Concepts Services Nigeria Ltd | Pandaville Labs

**ACTORS**

* ServiceProvider
* Consumers
* Admin

**ENTITIES**

Consumer{user\_uuid, name} <= [Service.Consumers]

Provider|ServiceProvider {user\_uuid, name} <= [Service.Providers]

Services {owner\_uuid, name, %tags{}, description, %details{description,thumbnail...}} <= [Service]

Orders {uuid, %consumer{}, created\_at, %when(from\_date, to\_date), %where{%address{}}, %order\_events[timestamp, %status]}

Invoice{uuid,order\_uuid, total, %line\_items[], %payment\_events[%payment{},timestamp, %status] }

Payments{uuid, invoice\_uuid, from\_account\_uuid, to\_account\_uuid}

Accounts{uuid, class\_account\_uuid, name, opening\_balance, current\_balance}

User {uuid, handle, %login\_details{}, created\_at} <= [Identity.Security]

Profile {uuid, %profile\_details{%profile\_photos{}, email, phone, %home\_address{}, %scoring{loyalty}}} <= [Identity.Profiles]

COMMANDS {\_transaction\_id, timestamp} in every command internally used

login{%login\_details{}, %login\_method}

register\_user {%user{}}

create\_user\_account{%user, %account} => %Account{}

profile\_user{%user, %profile{}}

create\_service{%provider, %service{}} ==> %Order{}

Update\_Service {

request\_service{%service, %requests}

accept\_service{%serviceprovider, %order\_event{ACCEPTED}}

reject\_service{%serviceprovider, %order\_event{REJECTED}}

cancel\_service{%serviceprovider, %order\_event{CANCELLED}}

confirm\_service{%serviceprovider, %order\_event{CONFIRMED}}

} ==> %Order{} #New Order state

generate\_invoice{%order, total, %line\_items[] %} ==> %Invoice{}

pay\_invoice{%invoice, amount} ==> %Invoice{%payment\_events{%payment{%from\_consumer, %to\_provider, amount}, PARTIAL/FULL}}

DOMAINS[CONTEXT] (sub-domains)

* Identity (security, profile)
* Service (service\_requests)
* Finance (billing, gift\_card, )
* Rewards (loyalty,)

**ACTIVITIES/COMMANDS**

* Registration: marks the person as a basic user (default) or service\_provider
* Authentication&Authorization
* Sales
* Payments

**EVENTS**

* order\_events
* invoice\_events