Product Backlog

Patient can:

1. Crud their PROFILE, crud their DEPENDANTS profile

// 2. has many TICKETs

3. view TEST info (assigned and their diagnosis)

4. View MED info (assigned and where available)

5. View their INVOICE so far. What has been paid? What is standing out?

6. Pay INVOICE through mob money. 3 Steps maxim.

7. View a RECEIPT at the end of each payment cycle

8. Vue their INSURANCE info (insurance\_provider)

9. Vue how many patients are ahead of them.

10. Request appointment

11. View about updates

12. Authentication & authorization.

13. Password change

The constraint of patients/dependents is a little bit challenging. One way of tackling it is to abstract the family as the primary entity.

The family representative shall register first. Then we shall ask them about

database backup. Before we remove anything or change to a new architecture.

Receptionist:

#1.Can Create a new profile

#2.Can Read existing profile

//3.Can Update a profile

//4.Can Delete a profile

#5. initiate a patients ticket.

Consultant:

#1. Can record vitals

#2. Can assign tests

3. view lab results -> select diagnosis from lab\_tests\_orders where invoice\_line\_id= id. Make a function that tells the patient cons ticket from an invoice line\_line\_id.

4. assign meds -> create invoice\_line(current\_amount, )

//5. Can view waiting patients 4 lab

//6. view waiting patients 4 meds -> status\_checker($ticket\_id, $col\_to\_check){ return status)

//7. view tickets that did not get serviced the previous day -> status\_checker($ticket\_id, $col\_to\_check){ return status)

//8. view already served tickets, people who went home -> status\_checker($ticket\_id, $col\_to\_check){ return status)

//9. admit a patient in hospital -> create a table of in\_patients (admission\_id, cons\_ticket\_id, admission\_date, discharge\_date, cons\_ticket\_id), at the end of the stay, provide the invoice. For the length of their stay, we shall add the charges to their invoice Remember that the nvoice can be paid partially, but yet still be open. We need to know that an invoice is open or closed.

//10. give a patient reference to another hospital

Lab Technician:

#1. Can view ordered tests for a ticket

#2. Can assign results for particular ticket

#3. Can constantly view waiting tickets (their total number, their IDs) -> status\_checker($ticket\_id, $col\_to\_check){ return status)

//4. Can view the number of served -> status\_checker($ticket\_id, $col\_to\_check){ return status)

Pharmacist can:

1. View assigned meds for a particular ticket ID

2. Can check that meds have been given out for a particular ticket -> status\_checker($ticket\_id, $col\_to\_check){ return status)

Finance:

Accountant:

1. view the journal for the day with total

2. The number of patients served so far

3. All other activities of a cashier

Cashier:

1. View, issue & close invoice

ITAdmin:

1. Can create a new entity (staff, med, etc... )

Owner:

1. Can view the amount of money made daily per prod category

2. How many patients where served in during the day

3. Reports (financial, Nber of patients received insights)

4. Can put suspend any users account

5.

Nurse:

1. Can view list of patients in a their ward

2.

Other functionalities: