Hospital Management System

1. *Identifying Entities*
    a. Doctor
    b. Nurse
    c. Patient
    d. IT Person
    e. Pharmacist
    f. Lab Assistant
    g. Operations Manager
    h. Surgeon
    i. Receptionist
    j. Administrative Staff
    k. Cleaner
    l. Others

2. *Identifying Classes and its attributes and methods (1 & 2)*

Hospital
+ name
+ address
+ contact_number

Person
+ first_name
+ last_name
- age
- gender
- contact_number
- date_of_birth
+ get_details()

Patient *extends* ***Person***
- patient_id
- sickness
- blood_group
- allergies
- medication
+ get_appointment()
+ buy_medicines()
+ pay_bills()

Employee *extends* **Person**
+ employee_id
+ joining_date
+ work_experience
# pay_scale

+ work_timings
+ apply_for_leaves()
+ login()
+ check_patient_status()

TechnicalStaff *extends* **Employee**
+ specialization
+ provide_technical_assistance()
+ fix_computer_problems()
+ fix_hospital_server_issues()
+ repair_machinery()

OperationsStaff *extends* **Employee**
+ specialization
+ check_patient_status()

AdministrativeStaff *extends* **Employee**
+ ensure_functionality()

Doctors *extends* **OperationsStaff**
+ speciality
+ patients_assigned
+ operate()
+ prescribe_medicines()
+ prescribe_tests()
+ check_reports()
+ give_advise()
+ give_work_to_nurses()

Surgeon *extends* **doctor**
+ speciality
+ operate()

Nurses *extends* **OperationsStaff**
+ take_care_of_patients()
+ give_medicines_to_admitted_patients()
+ assist_doctors()

ClinicalAssistant *extends* **AdministrativeStaff**
+ specialisation
+ perform_tests()
+ use_machines()
+ maintain_machines()

Pharmatologist *extends* **AdministrativeStaff**
+ give_medicines()
+ check_medicines_data()

- generate_bill()

Receptionist *extends* **AdministrativeStaff**
register_patient()
- take_payments()
+ request_technical_assistance()
+ fix_appointment()
+ check_patient_status()

Medicines
+ name
+ cost
+ expiry
+ used_for
+ alternative_drug
+ quantity_available
- need_perscription
+ sell()
+ modify()
+ delete()

Test
+ type
+ cost
- need_prescrption
+ side effects
+ precautions
+ perform_test
+ return_results()

Ward
+ capacity
+ admit_ward()

Assumptions:
- The above each of the bolded words are classes.
- The + sign denotes the access modifier.
- And the values with () are functions.
- Extends assume that it inherits the class.

3. *Relations*
    a. Pharmacist Dispenses medicines to Patient
    b. Doctor prescribes medicines to Patient
    c. Doctor prescribes tests to Patient
    d. Doctor Checks reports of Patient
    e. Patient Pays Amount to receptionist

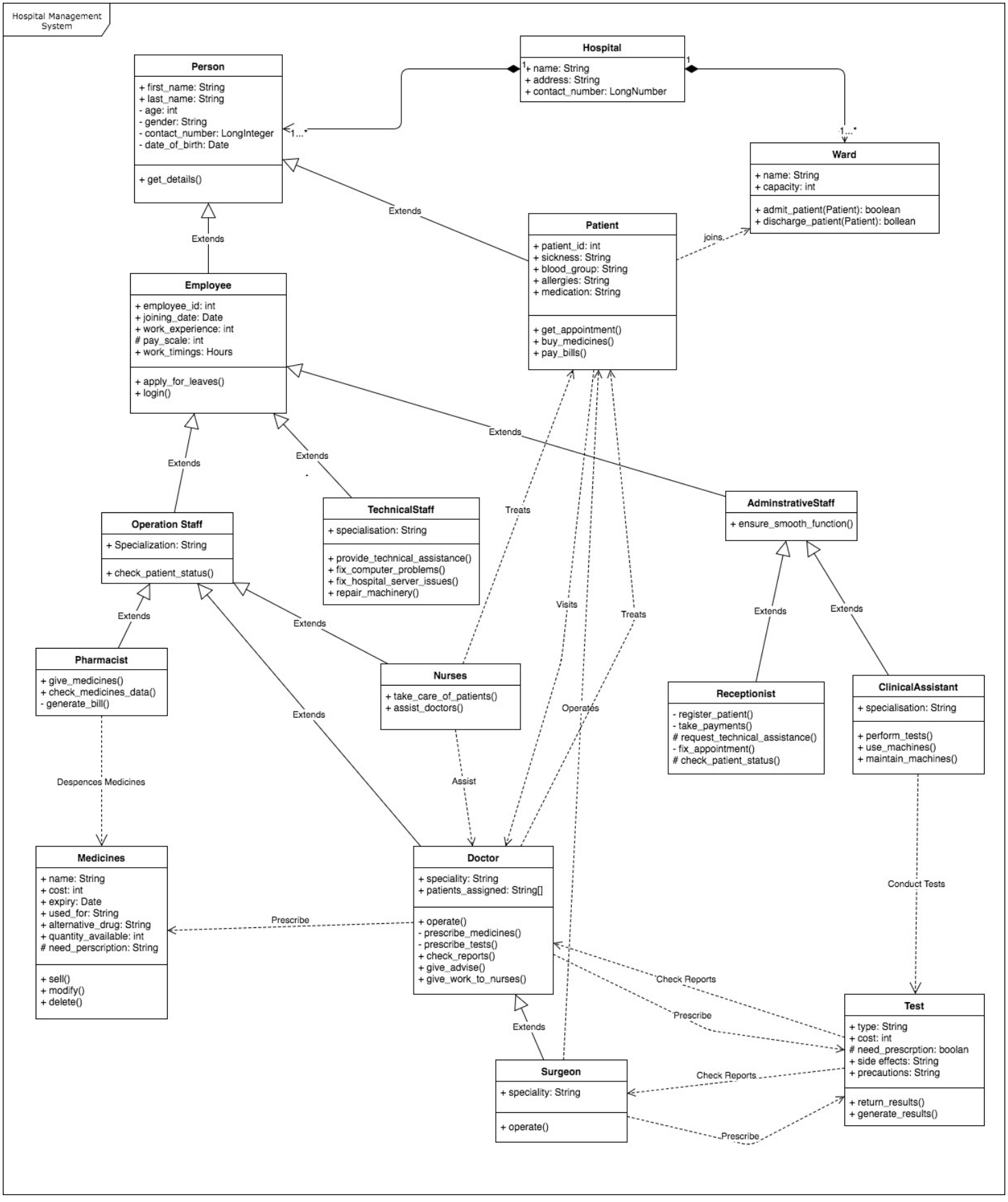
f. Patient Books appointment
g. Patient Gets admitted in hospital
h. Patient gets discharged from hospital
i. Nurse treats patients
j. Nurses assist Doctors
k. Surgeon operates on Patient
l. Surgeon checks reports of patient
m. Operations staff can check patient's status
n. Clinical staff conducts test on patients
o. Cleaner cleans the hospital

Assumptions: Many more such relations are shown in the diagram.

4. UML Diagram (4^{th} & 5^{th})

Assumptions:

- Dotted lines are relations.
- Many to one and one to many relations are also shown.
- And extends is used to show to hierarchy.


Hospital Management System
Person
+ first_name: String
+ last_name: String
- age: int
- gender: String
- contact_number: LongInteger
- date_of_birth: Date
+ get_details()
Hospital
+ name: String
+ address: String
+ contact_number: LongNumber
Ward
+ name: String
+ capacity: int
+ admit_patient(Patient): boolean
+ discharge_patient(Patient): bollean
Patient
+ patient_id: int
+ sickness: String
+ blood_group: String
+ allergies: String
+ medication: String
+ get_appointment()
+ buy_medicines()
+ pay_bills()
Employee
+ employee_id: int
+ joining_date: Date
+ work_experience: int
# pay_scale: int
+ work_timings: Hours
+ apply_for_leaves()
+ login()
Operation Staff
+ Specialization: String
+ check_patient_status()
TechnicalStaff
+ specialisation: String
+ provide_technical_assistance()
+ fix_computer_problems()
+ fix_hospital_server_issues()
+ repair_machinery()
AdminstrativeStaff
+ ensure_smooth_function()
Pharmacist
+ give_medicines()
+ check_medicines_data()
- generate_bill()
Nurses
+ take_care_of_patients()
+ assist_doctors()
Receptionist
- register_patient()
- take_payments()
request_technical_assistance()
- fix_appointment()
check_patient_status()
ClinicalAssistant
+ specialisation: String
+ perform_tests()
+ use_machines()
+ maintain_machines()
Medicines
+ name: String
+ cost: int
+ expiry: Date
+ used_for: String
+ alternative_drug: String
+ quantity_available: int
# need_perscription: String
+ sell()
+ modify()
+ delete()
Doctor
+ speciality: String
+ patients_assigned: String[]
+ operate()
- prescribe_medicines()
- prescribe_tests()
+ check_reports()
+ give_advise()
+ give_work_to_nurses()
Surgeon
+ speciality: String
+ operate()
Test
+ type: String
+ cost: int
# need_prescrption: boolan
+ side effects: String
+ precautions: String
+ return_results()
+ generate_results()
1
1...*
joins
Extends
Treats
Visits
Operates
Assist
Despences Medicines
Prescribe
Conduct Tests
Check Reports