INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT AKURDI, PUNE

Documentation On

Quick Care Service

PG-DAC SEPT 2022

Submitted By:

Group No: 62

PRN no.	Name:
220941220212	Vipul Babhare
220941220034	Tushar Bhendarkar

Mr.Rohit Puranik
Centre Coordinator

Mrs.Manjiry Deshpande
Project Guide

ABSTRACT

QuickCare Services provides various services ranging from Household help to Medical help. Many a times an individual has to face difficulties in finding a proper physiotherapist if he/she is recovering from any serious fracture or struggling to maintain a proper body posture and also a proper childcare support if they want anyone to take care of his/her infant child or even a proper housemaid , Quickcare Services resolves such issues with ease.The frontend of this project is built on React and for the backend it uses Springboot whilst using MySQL for the database..

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my sincere and heartfelt thanks to our esteemed guide, **Mrs. Manjiry Deshpande** for providing me with the right guidance and advice at the crucial juncture sand for showing me the right way. I extend my sincere thanks to our respected **Centre Co-Ordinator Mr.Rohit Puranik**, for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

Vipul Babhare

Tushar Bhendarkar

Table of Contents

LIST OF FIGURES

INTRODUCTION

As technology is ruling the industrial world, manual activities and documentation are digitalizing. Our application tends to ease the process of finding caretaker in hour of need using technology. When any elderly or ailing family member is unwell and need help to do their routine work or need exceptional care at that time our application comes as helpful. In that situation we must exhaustively search for skilled people who can help within time constraint. Our application comes as handy here, we provide trained nurses, ward boy to give special attention to them. User does not have to go through the
pain of searching Caretaker. We can get skilled help on our fingertips. Our application provides a platform
that help to connect skilled nurses, ward boy to needful people for nursing services.

Quick care service/taker is a service provider system used for providing service to care receiver(customer) by
any type of service provider according to the need of care receiver(customer). This system works on two different
users:

1. Care provider (Caretaker) - Care Provider provides the service to the care receiver. Caretaker needs to register into the platform according to their hourly charges and approve the request which is being sent by customer.
2. Care receiver (Customer) - Care Receiver receives the service from care provider. Customer needs to register in platform to enter and book the caretaker according to the location and type of service he wants. Accordingly, customer will end up booking for Caretaker as per their need.

This system helps the service provider to reach to their care receiver(customer) and contact them quickly.
Customers must add their details and their need and based on their requirement the nearest caretaker can contact their customer and they can also keep track on their orders. The main objective is to help the customer by providing the service and solve their problem as soon as possible.

PROJECT OBJECTIVE AND SCOPE

QCS (Quick Care Service) allows caretaker and customer to register into the platform. Our platform provide service to customer without wasting much time searching for the caretaker. On the other side, caretaker gets a platform for getting a job and providing services. It is easy for customer and caretaker to see details of booking and the records which gets updated every time action is done in the account.

1.Customer (Caretaker/Care Receiver) registration, customer login, customer credential authentication. Customers need to fill required attributes to register.

2.Care Receiver can add patient details and view patient list. Care receiver can choose type of service and service time span. In patient list care receiver can delete and edit details of patient.

3.Care Receiver can book caretaker according to the location and type of care he wants and send request.

4.Care Receiver can view requested caretaker list where he can see status of request and

cancel request. Care Receiver can do payment if status is accepted and if rejected can view the reason for it.

5.In payment option he can choose mode of payment and pay the amount displayed on the screen. Care receiver can also give feedback for caretaker and generate report for the service received.

6. Caretaker can view all request and can accept or reject them. If he wants to reject

request,he must give reason for rejection.

7.In patient list caretaker can view patient's list, their details and whether request accepted or not. He can also view feedback received from care receiver

PROJECT OVERVIEW

QCS (Quick Care Service) allows caretaker and customer to register into the platform. Our platform provide service to customer without wasting much time searching for the caretaker. On the other side, caretaker gets a platform for getting a job and providing services. It is easy for customer and caretaker to see details of booking and the records which gets updated every time action is done in the account.

1. Care provider (Caretaker) - Care Provider provides the service to the care receiver. Caretaker needs to register into the platform according to their hourly charges and approve the request which is being sent by customer.
2. Care receiver (Customer) - Care Receiver receives the service from care provider. Customer needs to registerin platform to enter and book the caretaker according to the location and type of service he wants. Accordingly, customer will end up booking for Caretaker as per their need.

This system helps the service provider to reach to their care receiver(customer) and contact them quickly.

Modification and improvement over the existing Implementation:

As of now we have not came across no such existing project in the market. Existing services only have only

Project Plan:

No. of User's and Modules

- **Users-**

1)caretaker
2)customer

- **Modules** –

SERVICE RECEIVER MODULE (CUSTOMER):

Customer in this application who is service receiver, who will receive service after their service order being placed. Customers must register themselves if they don'thave their profile, once the registration is done, they can proceed with the login process,if they already have profile, they can directly login into their dashboard. once they reach their dashboard, they can provide placed their service request and can also see their previous records of orders.

- REGISTRATION/LOGIN
- ADDING PERSONAL DETAILS
- ADDING PATIENT DETAILS
- REQUESTED CARETAKER LIST
- BOOKING SERVICE
- PAYMENT

SERVICE PROVIDER MODULE (CARETAKER):

Service provider is one of the users who will provide the services according to the customer request. Service provider also must register themselves if they do not have their profile, once the registration is done, they can proceed with the login process, if they already have profile, they can directly login into their dashboard. Once they reach their dashboard, they can see the customer request in the order list and can provide their services accordingly.

- REGISTRATION/LOGIN
- CARETAKER DASHBOARD
- ACCEPT / DECLINE OF SERVICE
- REASON FOR REJECTION
- PATIENT HISTORY

REGISTRATION MODULE:

Each user needs to get register before entering platform. The user can check the caregiver details and availability of the caregiver which is nearby them by sorting based on location and type of service. They can also check the booking status on notification section on customer dashboard.

LOGIN MODULE:

Users who have completed the registration process once they can access the application anytime in future through just login. After they have verified their usernameand password, they can directly type the username and password to login.

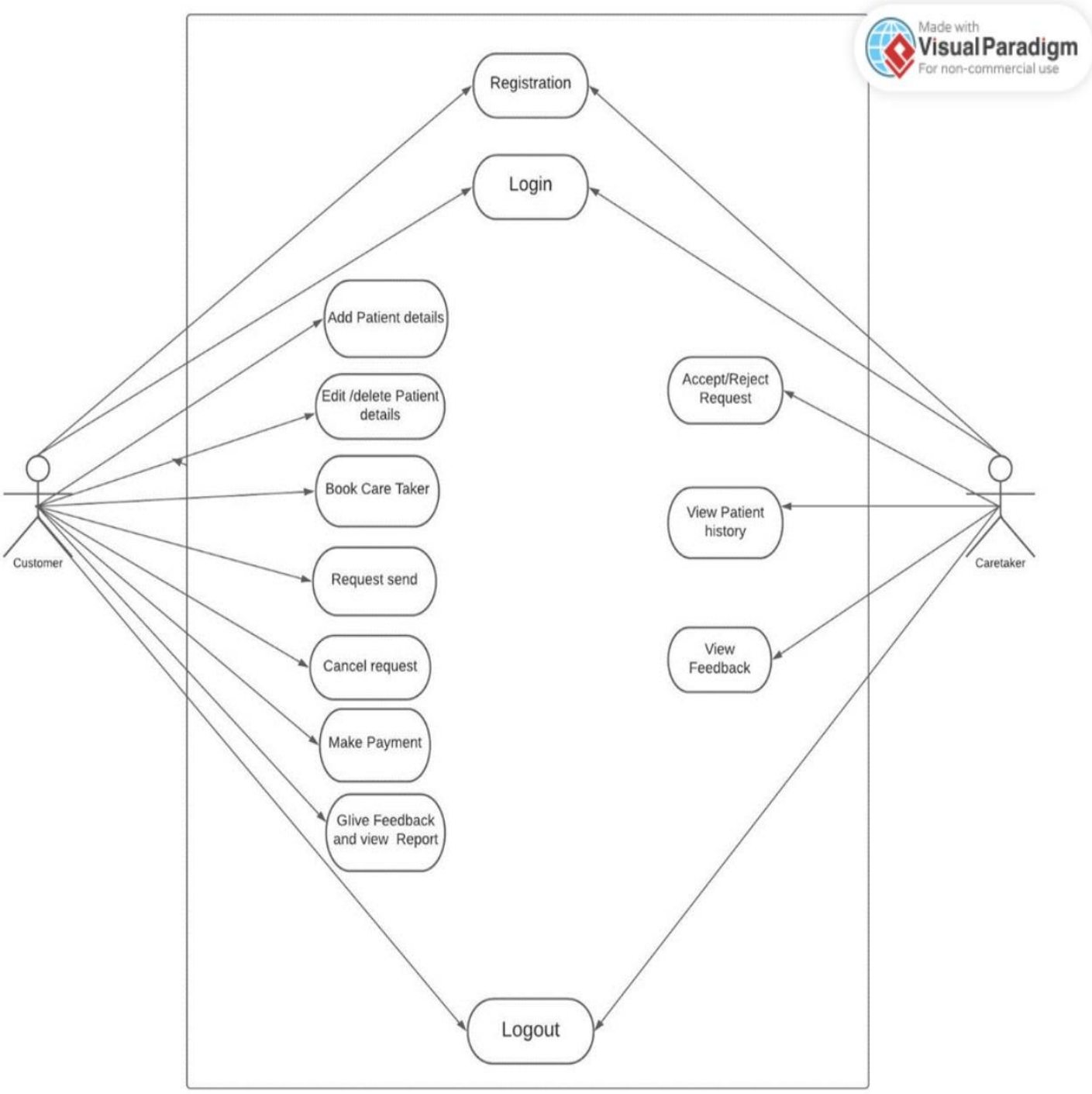
Technology:

Frontend- React
Backend- Java, Spring
Database- MySQL database
Others – Eclips IDE

Roles and Responsibilities:

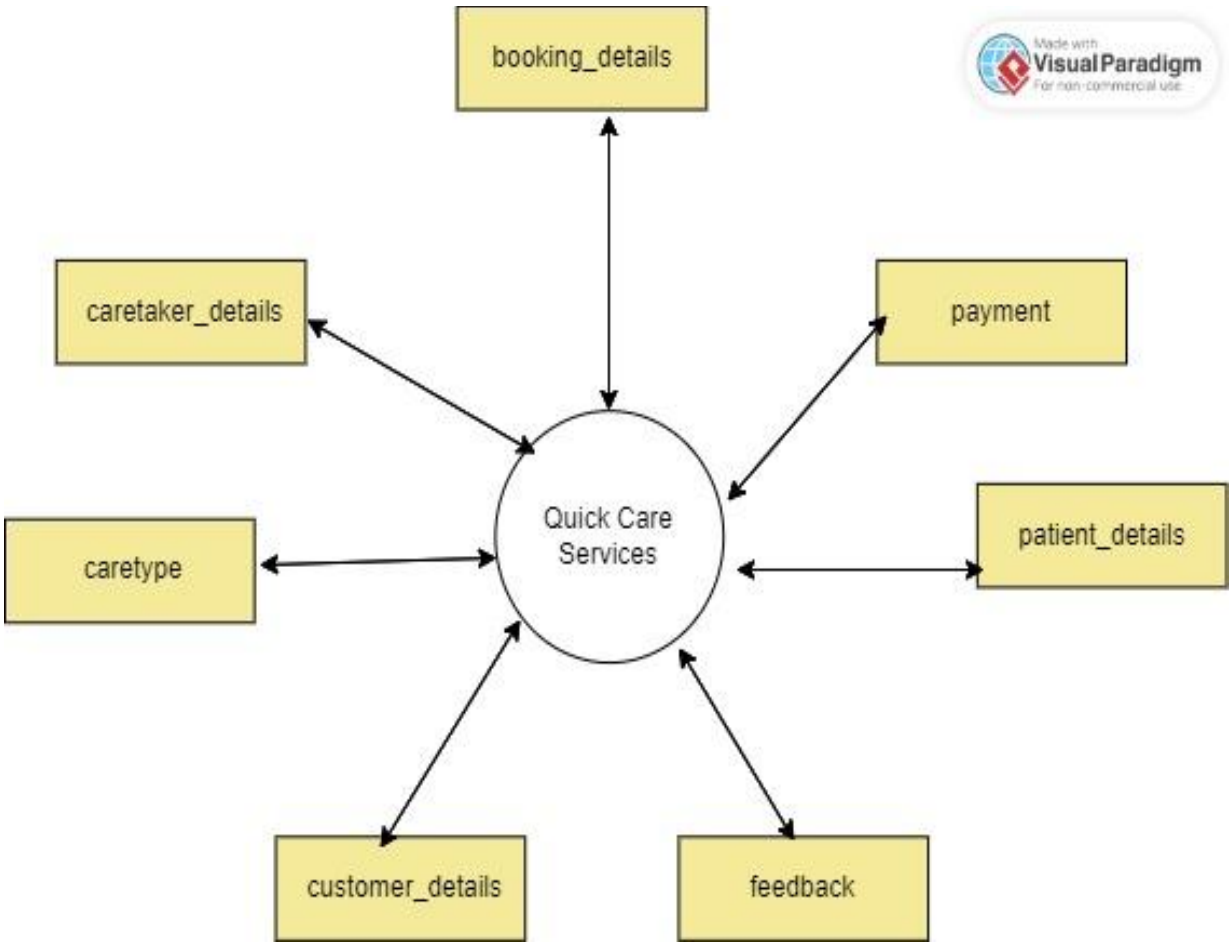
Roles And Responsibilities		
1	Role	Backend, Frontend
	Member Name	Vipul Babhare
	PRN No	220941220212
	Description	
2	Role	Backend, Frontend
	Member Name	Tushar Bhendarkar
	PRN No	220941220034
	Description	

Use Case Diagram;

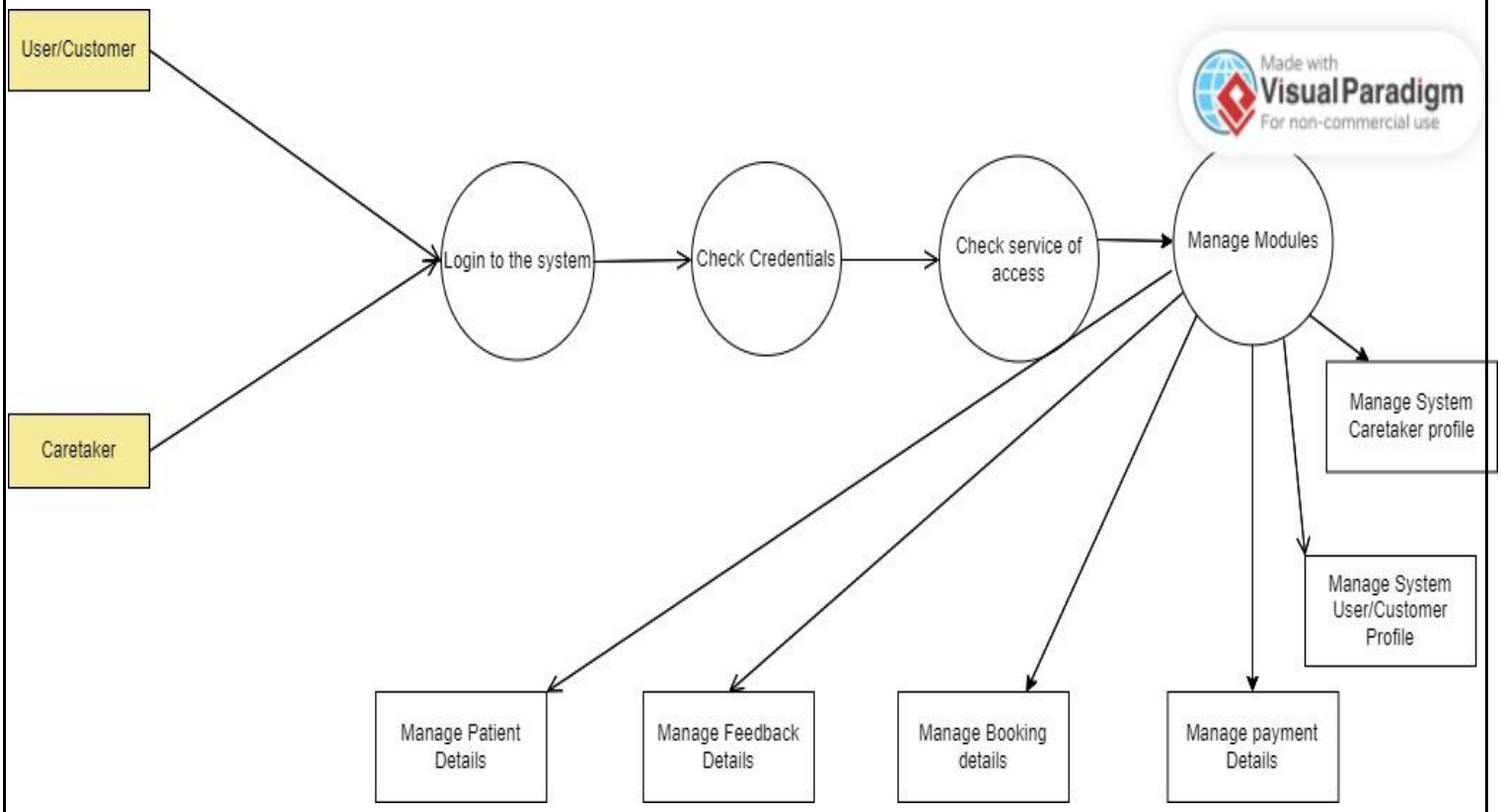


Data Flow Diagram(DFD):

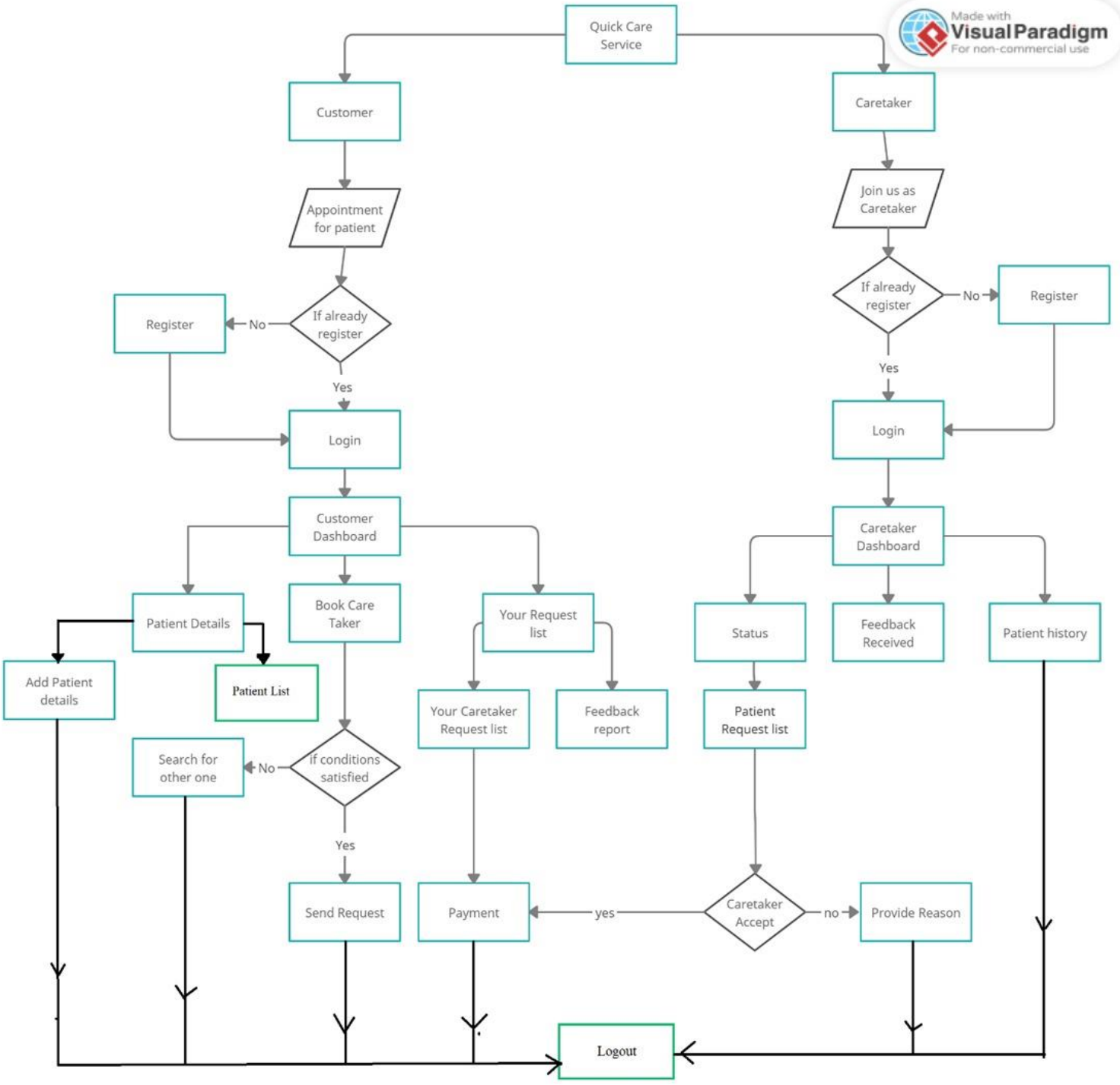
Level 0:



Level 1:


User/Customer
Caretaker
Login to the system
Check Credentials
Check service of access
Manage Modules
Manage System Caretaker profile
Manage System User/Customer Profile
Manage payment Details
Manage Booking details
Manage Feedback Details
Manage Patient Details
Made with VisualParadigm
For non-commercial use

Activity Diagram/Level 2:



Class Diagram:

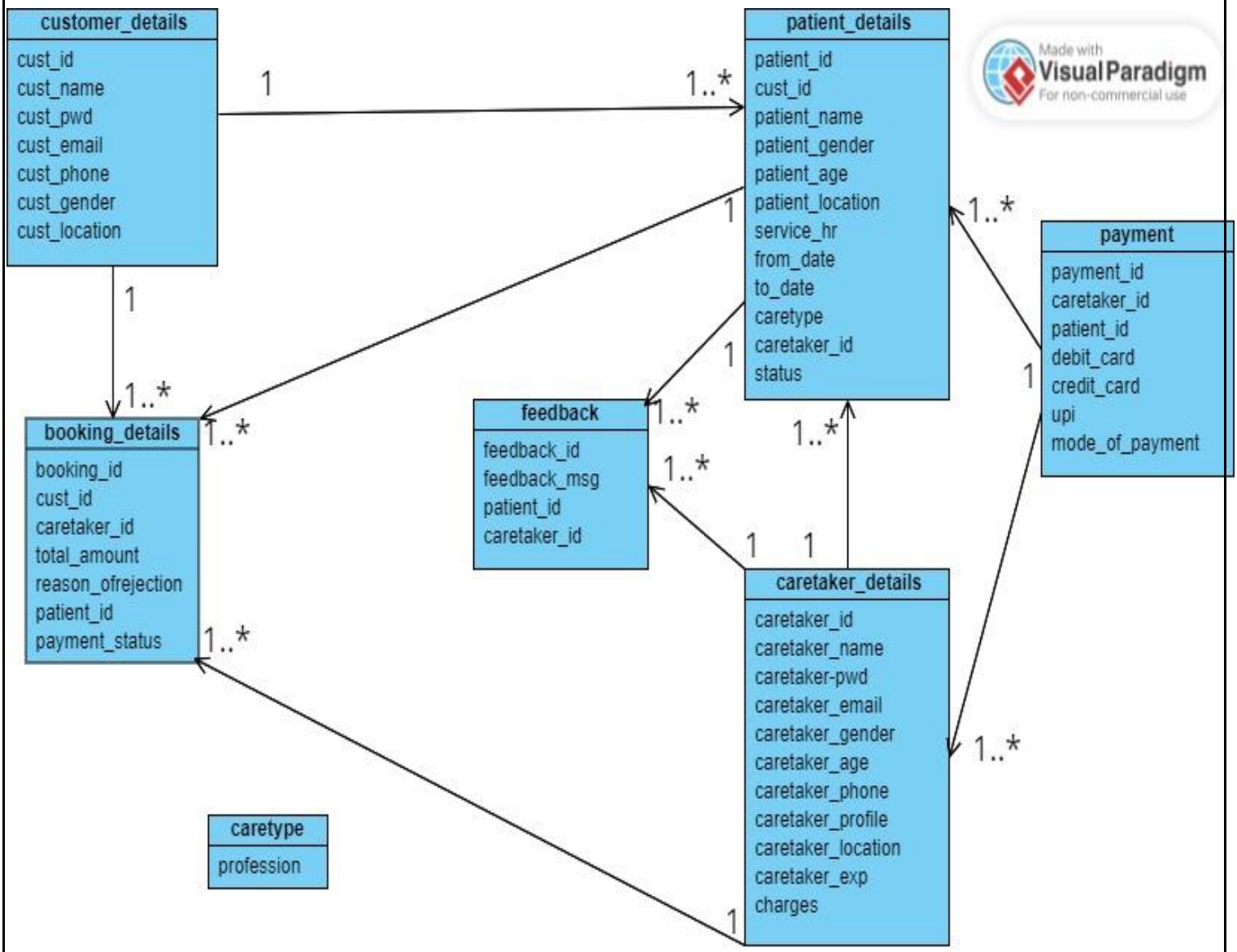


Table Structure:

1. Customer_details:

Field	Type	Null	Key	Default	Extra
cust_id	int	NO	PRI	NULL	auto_increment
cust_name	varchar(45)	YES		NULL	
cust_pwd	varchar(45)	YES		NULL	
cust_email	varchar(45)	YES		NULL	
cust_phone	varchar(45)	YES		NULL	
cust_gender	varchar(45)	YES		NULL	
cust_location	varchar(45)	YES		NULL	
usertype	varchar(45)	YES		NULL	

2. caretaker_details:

Field	Type	Null	Key	Default	Extra
caretaker_id	int	NO	PRI	NULL	auto_increment
caretaker_name	varchar(45)	YES		NULL	
caretaker_pwd	varchar(45)	YES		NULL	
caretaker_email	varchar(45)	YES		NULL	
caretaker_gender	varchar(45)	YES		NULL	
caretaker_age	int	YES		NULL	
caretaker_phone	varchar(45)	YES		NULL	
caretaker_profile	varchar(45)	YES		NULL	
caretaker_location	varchar(45)	YES		NULL	
caretaker_exp	varchar(45)	YES		NULL	
charges	double	YES		NULL	
usertype	varchar(45)	YES		NULL	

3. patient_details:

Field	Type	Null	Key	Default	Extra
patient_id	int	NO	PRI	NULL	auto_increment
cust_id	int	YES	MUL	NULL	
patient_name	varchar(45)	YES		NULL	
patient_gender	varchar(45)	YES		NULL	
patient_age	int	YES		NULL	
patient_location	varchar(45)	YES		NULL	
service_hr	int	YES		NULL	
from_date	date	YES		NULL	
to_date	date	YES		NULL	
caretype	varchar(45)	YES		NULL	
caretaker_id	int	YES	MUL	NULL	
status	varchar(45)	YES		NULL	

4. booking_details:

Field	Type	Null	Key	Default	Extra
booking_id	int	NO	PRI	NULL	auto_increment
cust_id	int	YES	MUL	NULL	
caretaker_id	int	YES	MUL	NULL	
total_amount	double	YES		NULL	
reason_ofrejection	varchar(45)	YES		NULL	
patient_id	int	YES	MUL	NULL	
payment_status	varchar(45)	YES		NULL	

5. Payment:

Field	Type	Null	Key	Default	Extra
payment_id	int	NO	PRI	NULL	auto_increment
caretaker_id	int	YES	MUL	NULL	
patient_id	int	YES	MUL	NULL	
cvv	int	YES		NULL	
expiry	date	YES		NULL	
credit_cardno	varchar(25)	YES		NULL	
nameoncard	varchar(25)	YES		NULL	

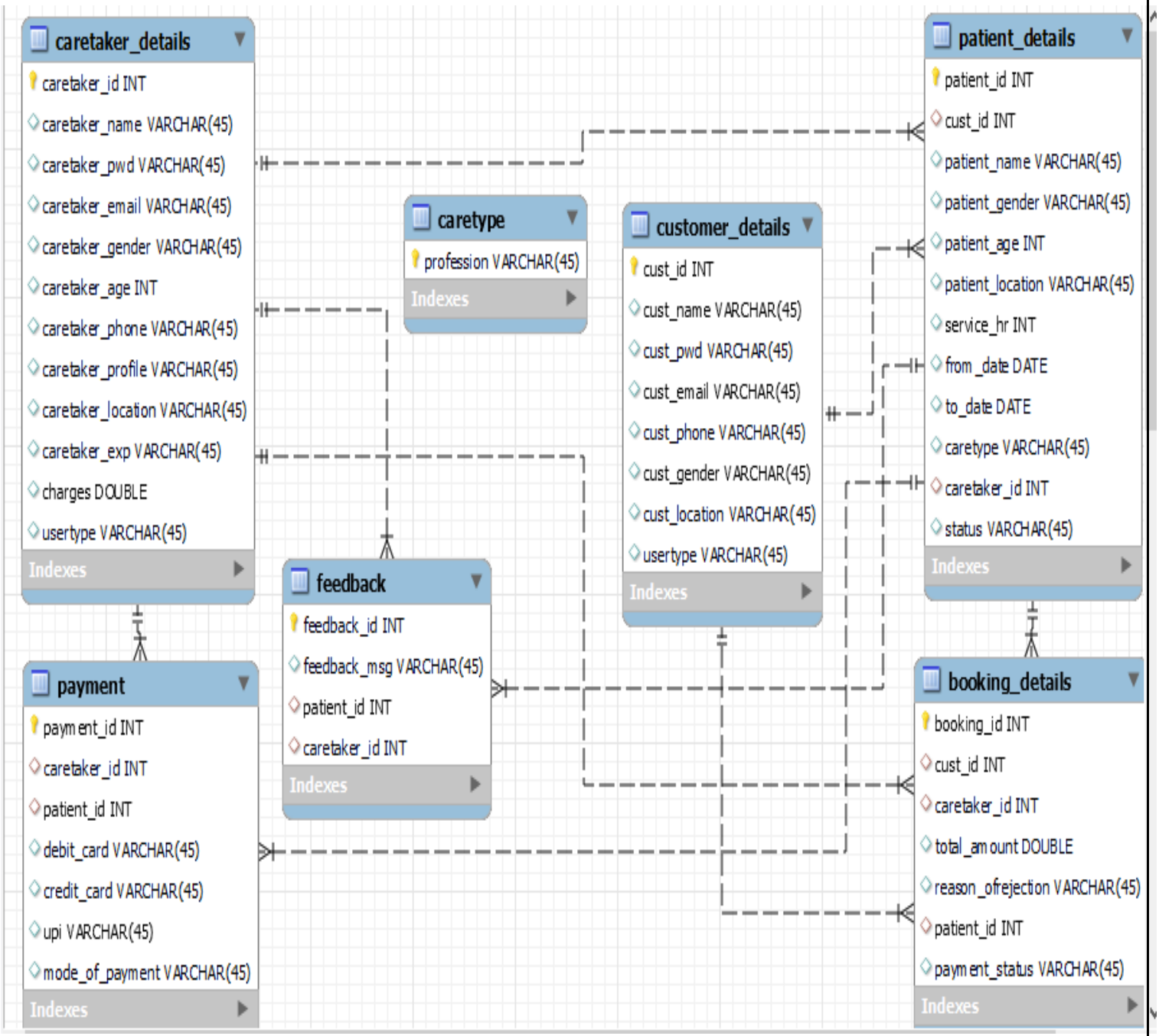
6. Feedback:

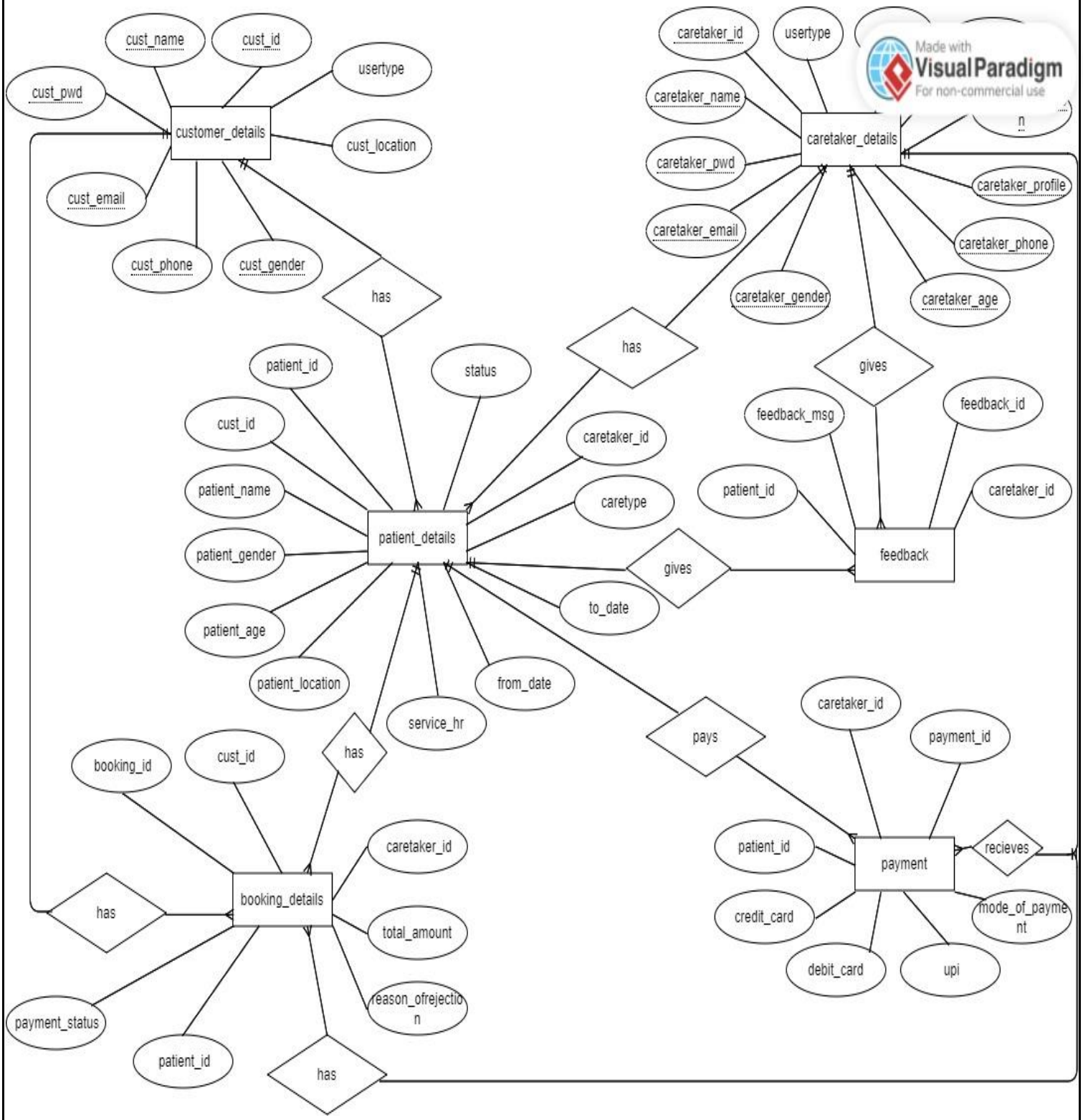
Field	Type	Null	Key	Default	Extra
feedback_id	int	NO	PRI	NULL	auto_increment
feedback_msg	varchar(45)	YES		NULL	
patient_id	int	YES	MUL	NULL	
caretaker_id	int	YES	MUL	NULL	

7. Caretype:

Field	Type	Null	Key	Default	Extra
profession	varchar(45)	NO	PRI	NULL	

ERD / Database Tables:




cust_name
cust_id
usertype
cust_pwd
customer_details
cust_location
cust_email
cust_phone
cust_gender
has
caretaker_id
usertype
caretaker_name
caretaker_details
caretaker_pwd
caretaker_profile
caretaker_email
caretaker_phone
caretaker_gender
caretaker_age
has
gives
patient_id
status
cust_id
caretaker_id
patient_name
caretype
patient_details
patient_gender
gives
feedback
to_date
patient_age
patient_location
from_date
service_hr
has
pays
feedback_msg
feedback_id
patient_id
caretaker_id
booking_id
cust_id
caretaker_id
payment_id
patient_id
payment
recieves
has
booking_details
credit_card
mode_of_payment
total_amount
debit_card
upi
payment_status
reason_ofrejection
patient_id
has
Made with
VisualParadigm
For non-commercial use

DATABASE DESIGN:

Databases are the storehouses of data used in the software systems. The data is stored in Tables inside the database. Several tables are created for the manipulation of the data forthe system. Two essential settings for a database are

- Primary key - the field that is unique for all the record occurrences
- Foreign key - the field used to set relation between tables

Normalization is a technique to avoid redundancy in the tables.

SYSTEM TOOLS:

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

FRONT END

React is a library which is developed by Facebook are utilized to implement the frontend. React(also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries

for routing, as well as certain client-side functionality.

BACKEND:

The back end is implemented using MySQL which is used to design databases.

MySQL:

MySQL is the world's second most widely used open-source relational database Management system (RDBMS). The SQL phrase stands for Structured Query Language.
An application software called Navicert was used to design the tables in MySQL.

Spring-Boot:

This is used to connect MYSQL and fetch data from database and store the data in database. The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there Are extensions for building web applications on top of the Java EE (Enterprise Edition) platform.Although the framework does not impose any specific programming model, it has become popularin the Java community as an addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is Open-source Framework.

Future Scope:

We can latter integrate this project in centralized manner with chain of hospitals, therapy centres and such healthcare service providers etc.

Reference:

- Mrs. Kishori Khadilkar for Database.
- Mrs. Kishori Khadilkar for REACT JS.
- Mrs. Madhura Anturkar for Springboot,java.
- https://www.slideshare.net
- https://www.projectideas.co.in

A special thanks to IACSD Management who arranged extra lab time for us.

ONLINE REFERENCE

[1] www.Google.com
[2] www.w3school.com
[3] www.javatpoint.com