

**RV COLLEGE OF ENGINEERING[®], BENGALURU-
560059**

(Autonomous Institution Affiliated to VTU, Belagavi)

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**



Online Medical Shop

Mini - Project Report

Submitted by

**SHAILESH KUMAR SHARMA
SHAURYA GUPTA**

**1RV18CS149
1RV18CS150**

in partial fulfillment for the requirement of 5th Semester

DBMS Laboratory Mini Project (18CS53)

Under the Guidance of

**Dr. Manjunath, A. E, Assistant Professor,
Computer Science Department, RV College of Engineering.**

Academic Year 2020- 2021

RV COLLEGE OF ENGINEERING®, BENGALURU - 560059
(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

Certified that the project work titled '**Online Medical Shop**' is carried out by **Shaliesh Kumar Sharma (1RV18CS149)** and **Shaurya Gupta (1RV18CS150)**, who are bonafide students of RV College of Engineering®, Bengaluru, in partial fulfillment of the curriculum requirement of 5th Semester Database Design Laboratory Mini Project during the academic year **2020-2021**. It is certified that all corrections/suggestions indicated for the internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in all respect laboratory mini-project work prescribed by the institution.

Signature of Faculty In-charge

Head of the Department
Dept. of CSE, RVCE

External Examination

Name of Examiners

Signature with date

1

2

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this project work. I would like to take this opportunity to thank them all.

I deeply express my sincere gratitude to my guide Dr. Manjunath,AE, Assistant Professor ,Department of CSE, RVCE, Bengaluru, for his able guidance, regular source of encouragement and assistance throughout this project

I would like to thank Dr.Ramakanth Kumar P, Head of Department, Computer Science & Engineering, R.V.C.E, Bengaluru, for his valuable suggestions and expert advice.

First and foremost I would like to thank **Dr. Subramanya. K. N**, Principal, R.V.C.E, Bengaluru, for his moral support towards completing my project work.

I thank my Parents, and all the Faculty members of Department of Computer Science & Engineering for their constant support and encouragement.

Last, but not the least, I would like to thank my peers and friends who provided me with valuable suggestions to improve my project.

Abstract

This project is based and innovated on an Online medical shop, wherein we store all the details about the customers, the stock of the medicines, orders and payments and also the project will include a page wherein the user will indicate the symptoms and will get a probable disease and the prescribed medicine. The project is aimed to modernize and support existing small business owners. In the age of technology where online medicine is dominated by ecommerce giants such as 1mg, net meds etc. We wanted to develop a solution for small business owners as well.

The existing Medicinal systems have the provision for any user to book a request for a particular medicine through e-commerce. And further the traditional methods to visit the medicinal centers for mere enquiry is time consuming and monotonous and the non-availability is disappointing. The data relevant to the processing of the request may or may not be manually stored or be captivated in a file system which is prone to manual errors, inconsistency, redundancy and difficulty in retrieval. With our system the availability can be shown so, even if a customer wants a pickup of his/her medicine. they can do so without any problems. Our model also has an integrated web scraper, which is an innovation we have come up with. This scraper can scrape medicines off the net for data warehousing.

This system maintains the storage details of all the customers, medicines that are stored in the shop. The system will keep track of the orders made and the payment details. NOSQL will be used to store future suggestions and customer reviews. The main part of the project will be a part where the customer will be able to select his/her symptoms and a medicine will be referred to them. Along with expected disease. We also would integrate Web Scraping of all the medicines related to a particular disease entered by the user to store it in our database.

Table of Contents

Acknowledgement	i
Abstract	ii
Table of Contents	iii
List of Figures	vi
List of Tables	vii

Glossary	Page No.
1. Introduction	7
2. Software Requirement Specification	8-11
2.1 Software Requirements	8
2.2 Hardware Requirements	9
2.3 Functional Requirements	10
2.4 Non Functional Requirements	11
3. ER Diagram	12
4. Detailed Design	13-16
4.1 Data Flow Diagram	
4.1.1 DFD Level 0	13
4.1.2 DFD Level 1	14
4.1.3 DFD Level 2	15,16
5. Relational Schema and Normalization	17,18
6. NoSQL	19
7. Conclusion & Future Enhancement	19
References	20
Appendix: Snapshots	20-24

List of Figures

Figure No.	Figure Name	Page.No.
3.1	ER DIAGRAM	12
4.1.1	DFD0 ADMIN	13
4.1.2	DFD0 CUSTOMER	13
4.2.1	DFD1 ADMIN	14
4.2.2	DFD1 CUSTOMER	14
4.3.1	DFD2 ADMIN	15
4.3.2	DFD2 CUSTOMER	16
5.1	RELATIONAL SCHEMA	17
Appendix 1-12	Snapshots	20-22
Appendix 12-16	Database details	22-24

GLOSSARY

DB	:	Databases
OMS	:	Online Medical Shop
SRS	:	Software Requirement Specification
SQL	:	Structured Query Language

Chapter 1

Introduction

People need medicine when they become sick and unable to enjoy their normal life. People have to go in the pharmacy for buying medicine. But sometimes they can't get the proper medicine due to the lack of easy availability. So, the patients have to search for the desired medicine in every medicine shop which wastes their time and energy. Sometimes, in emergency cases, the condition of patients become very serious without taking medicine at the right time.

Online Medicine Shop will give the opportunity to buy proper medicine sitting at home without any trouble. There are many medicine shops in every city but online medicine shop is very rare. The online medicine shop is operated over the internet. It will bring comfort to every buyer and seller of the medicine. The seller can manage the shop, store the details and relevant information of the medicine and sell the medicine in one system easily. The customer can see, search, buy medicine and give review about the service quality in this website. So, Online Medicine Shop will create a convenient way of selling and buying medicine and will make our life easier and hassle free.

The main objective is to manage the details of customer, medicine, stock, order and sell the medicine in online. In this technological era, everything becomes very faster. People love to enjoy a comfortable life. This online medicine shop will bring many facilities to our life. By using MySQL commands and its database this website Online Medical shop tends to store all the data received from the users including details and quantity remaining of the medicines. It also provides user with options add more medicine into the store and update the database accordingly. By using MYSQL server these database commands are easily initiated into the database and the ER diagram with relational schema diagrams helps us to make the structure of the database faster and it was easier to make them understand the needs of the website.

The main aim of the project is the management of the database of the pharmaceutical shop. This project is insight into the design and implementation of a Online Medical shop. This is done by creating a database of the available medicines in the shop. The primary aim of pharmacy management system is to improve accuracy and enhance safety and efficiency in the pharmaceutical store. The aim of this project is to develop software for the effective management of a pharmaceutical store. We have developed this software for ensuring effective policing by providing statistics of the drugs in stock.

Chapter 2

Software Requirement specification

Preamble :

The production of the requirements stage of the software development process is Software Requirements Specifications (SRS) (also called a requirements document). This report lays a foundation for software engineering activities and is constructing when entire requirements are elicited and analysed. SRS is a formal report, which acts as a representation of software that enables the customers to review whether it (SRS) is according to their requirements. Also, it comprises user requirements for a system as well as detailed specifications of the system requirements. The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment.

A software requirements specification is the basis for the entire project. It lays the framework that every team involved in development will follow.

It's used to provide critical information to multiple teams — development, quality assurance, operations, and maintenance. This keeps everyone on the same page.

Using the SRS helps to ensure requirements are fulfilled. And it can also help the designers make decisions about the product's lifecycle — for instance, when to retire a feature.

2.1 Hardware Requirements

- Processor: Minimum 1 GHz; Recommended 2GHz or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more.
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above.

2.2 Software Requirements

- Language support required: Python 3.5 or later, HTML5, JS, CSS3
- NOSQL database required: MongoDB
- Relational Database required: MySQL
- Windows 7 or 10 /Mac OS X 10.11 or higher, 64-bit /Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
- Heroku and pip are preferred for deployment and installation of packages (such as django, asgerif, mongoose, etc) specified in requirements.txt
- A web browser support needed.

If using the software through deployment, no language support in your machine is required.

2.3 Functional Requirements

We describe the functional requirements by giving various use cases.

Use case related to Creation of user and Customer account:

Use Case 1: Creation of User Account

Primary Actor: User

Pre-Condition: Internet connection available.

Main Scenario:

1: User opens portal and creates account by filling all the primary details mentioned (e.g.: Name, phone, email id, password)

Alternate Scenario:

- (a) Network and connection failure: (if deployment is used)
- (b) Account Creation aborted.

Use Case 2: Creation of Customer Account

Primary Actor: Customer

Pre-Condition: Internet connection available.

Main-Scenario:

1. User opens portal and creates account by filling all the primary details mentioned (e.g.: Name, phone, email id, password)

Alternate Scenario:

- (a) Network and connection failure: (if deployment is used)
- (b) Account Creation aborted.

Use case related to orders and medicine:

Use Case 3: stock management of medicine

Primary Actor: User

Pre-Condition: Internet connection available.

Main Scenario:

1. User open the portal and uploads the present/updated stock of any item.
2. Remove stock of any expired/damaged medicine.

Alternate Scenario:

- (a). Network and connection failure: (if deployment is used)

Use Case 4: Orders of medicine

Primary Actor: Customer

Pre-Condition: Internet connection available.

Main Scenario:

1. Customer open the portal and selects the medicine he/she wants.
2. Customer inserts the insurance details (if he/she) has
3. Customer orders the medicine
4. Customer gives feedback Alternate Scenario:
 - (a). Network and connection failure: (if deployment is used)
 - (b) Stock inventory is less/zero for a given medicine.
 - (c) Validation of feedback

Use case related Insurance:

Use Case 5: Customer has insurance

Primary Actor: Customer

Pre-Condition: Internet connection available.

Main Scenario:

1. Customer open the portal after he/she added medicines
2. Customer inserts the insurance details such as (insurance number, company, type, discount percentage) Alternate Scenario:
 - (a). Network failure
 - (b) Validation of given feedback

Use case related Payment:

Use Case 6: Customer does the payment:

Primary Actor: Customer

Pre-Condition: Internet connection available.

Main Scenario:

1. Customer open the portal after he/she ordered medicines
 2. Customer inserts the payment details on the given portal and does the payment.
- Alternate Scenario:
- (a). Network failure
 - (b) payment failure
 - (c) no favorable payment option
 - (d) order discarded
 - (e) stock mismatch

Use case related Feedback:

Use Case 7: Customer writes a feedback:

Primary Actor: Customer

Pre-Condition: Internet connection available.

Main Scenario:

1. Customer open the portal after he/she ordered medicines
2. Customer inserts order details and other details.
3. Customer uploads insurance ID and Prescription

Alternate Scenario:

- (a). Network failure
- (b) payment failure
- (c) wrong information entered

2.3 Functional Requirements

A. Performance Requirements

Some Performance requirements identified is listed below:

- The database shall be able to accommodate a thousand record to store.
- The software shall support use of multiple users at a time.
- Should run on basic machine. i.e. 500Mhz and 1Gb machine.

B: Safety Requirements:

Users with a valid ID-password can log on the system, therefore only valid users can access the functionality of the program. Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Keep specific log or history data sets

1. Assign certain functions to different module
2. Later version of the software will incorporate encryption techniques in the user/license authentication process.

C: Availability

The system while deployed should be available 24 x 7.

D: Ease of use:

The UI will be kept simple for a layman to understand and use.

E: Design constraint:

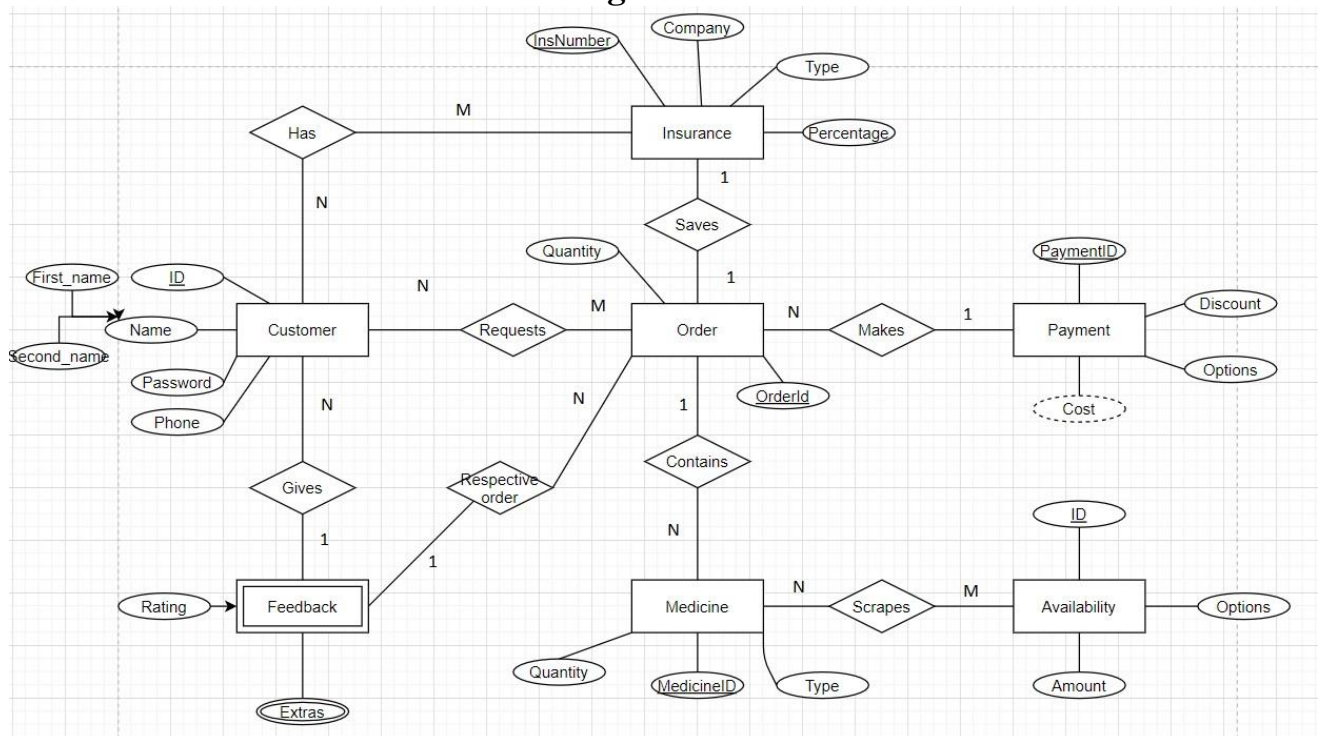
The developed system is very well accessible through any browser

Chapter 3

Preamble

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. Diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems

ER Diagram



Chapter 4 :

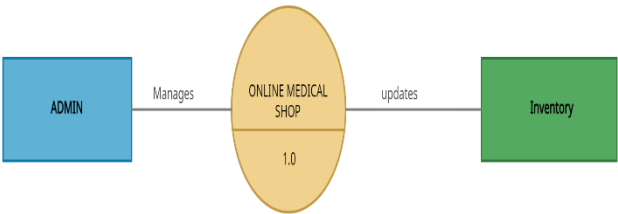
Detailed Design

Preamble

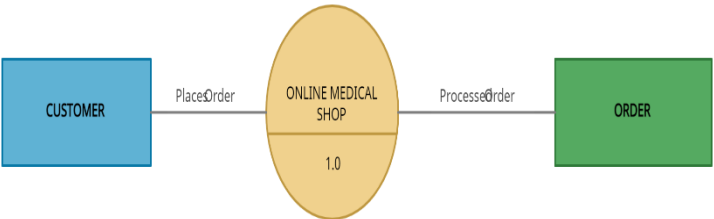
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, etc. Here, the project is realized up to a depth of level 2 DFD.

4.1 DFD Level 0

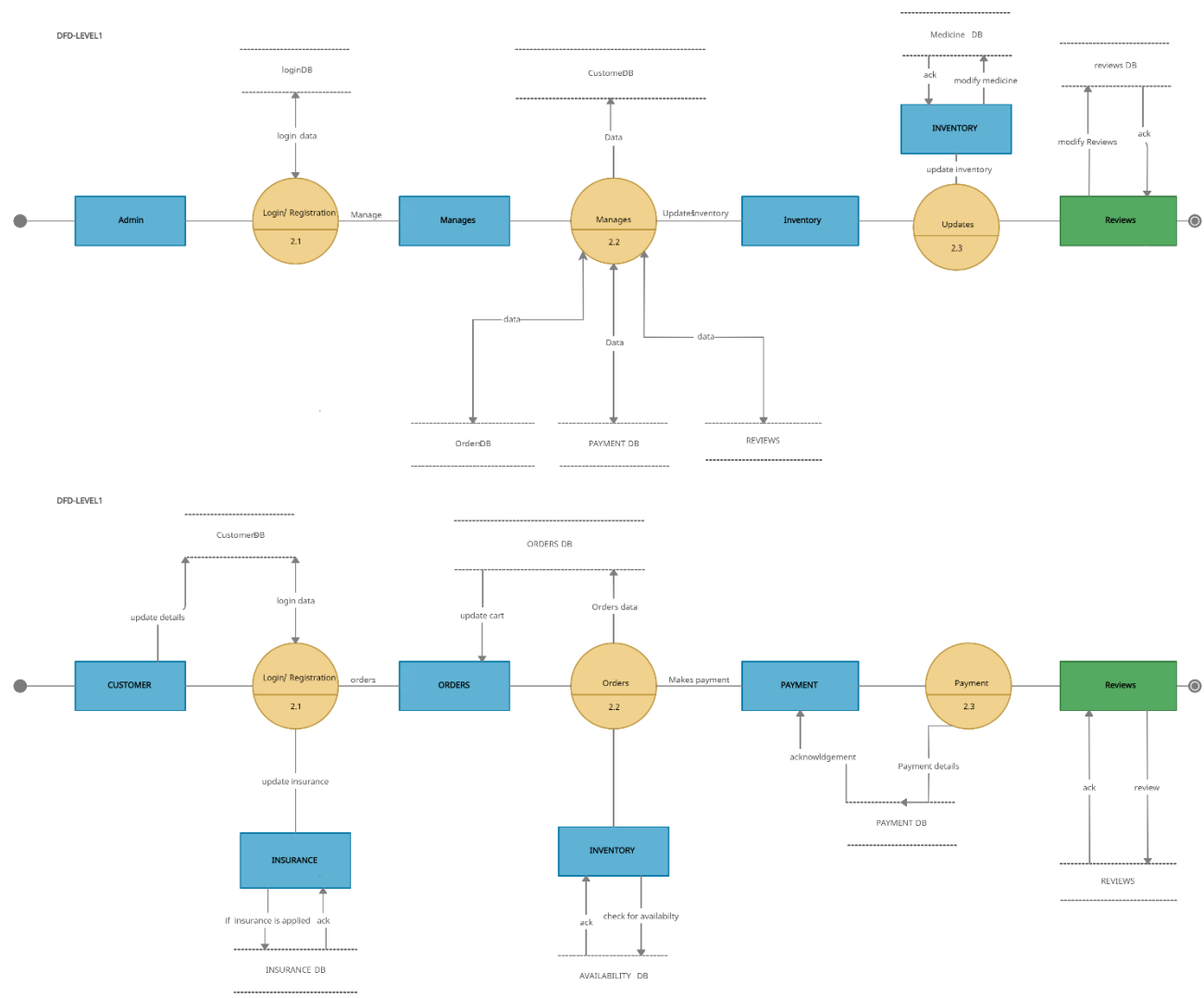
DFD-LEVEL0



DFD-LEVEL0

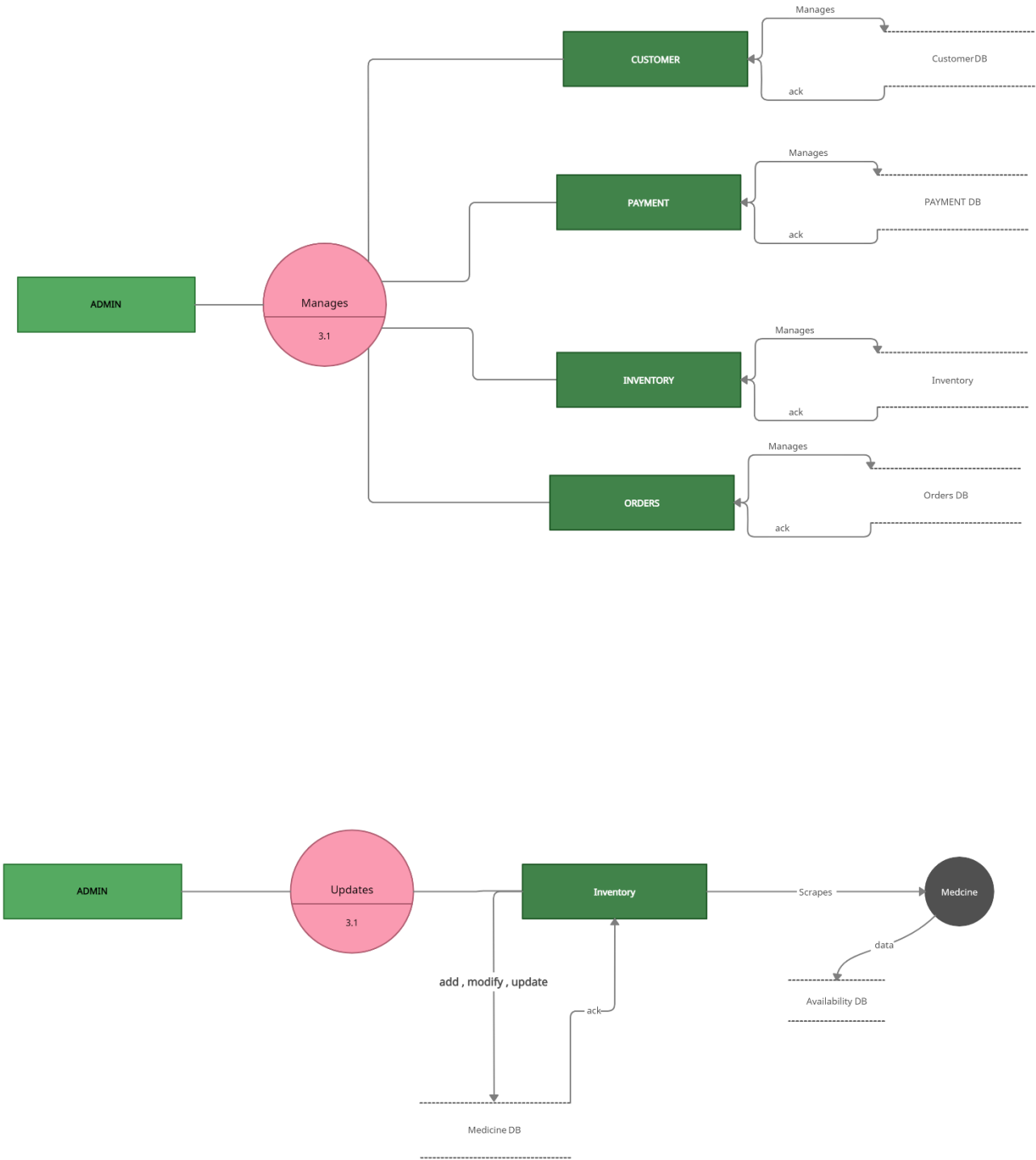


4.2 DFD Level 1

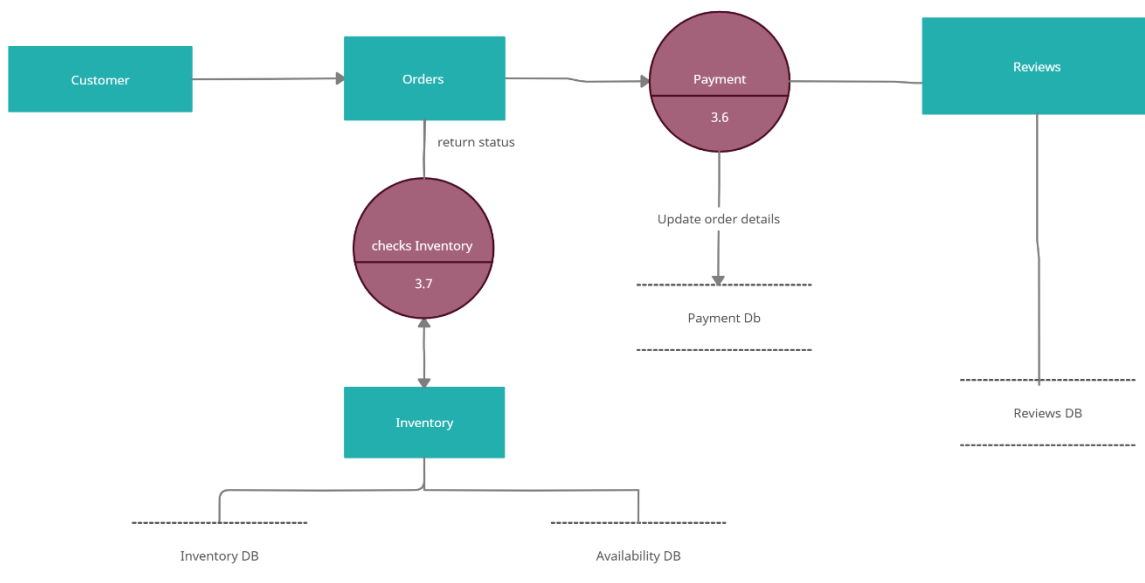
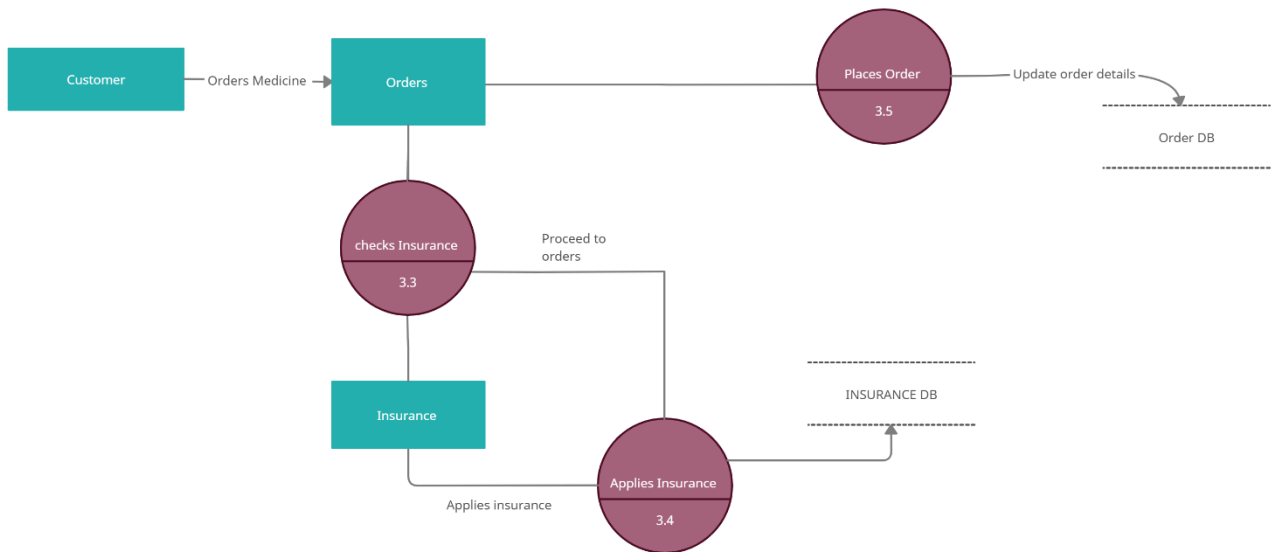
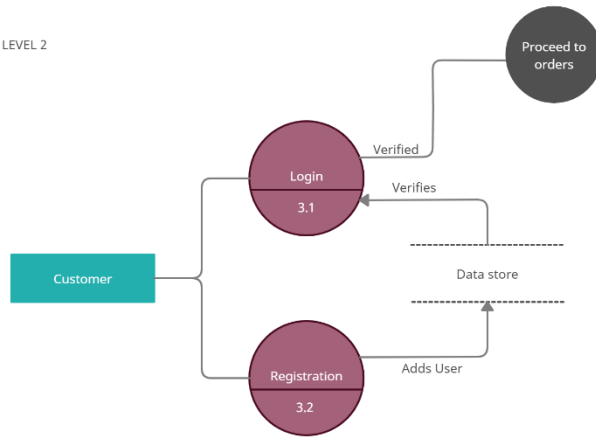


4.3 DFD Level 2

DFD LEVEL 2 - ADMIN



DFD LEVEL 2



Chapter 5

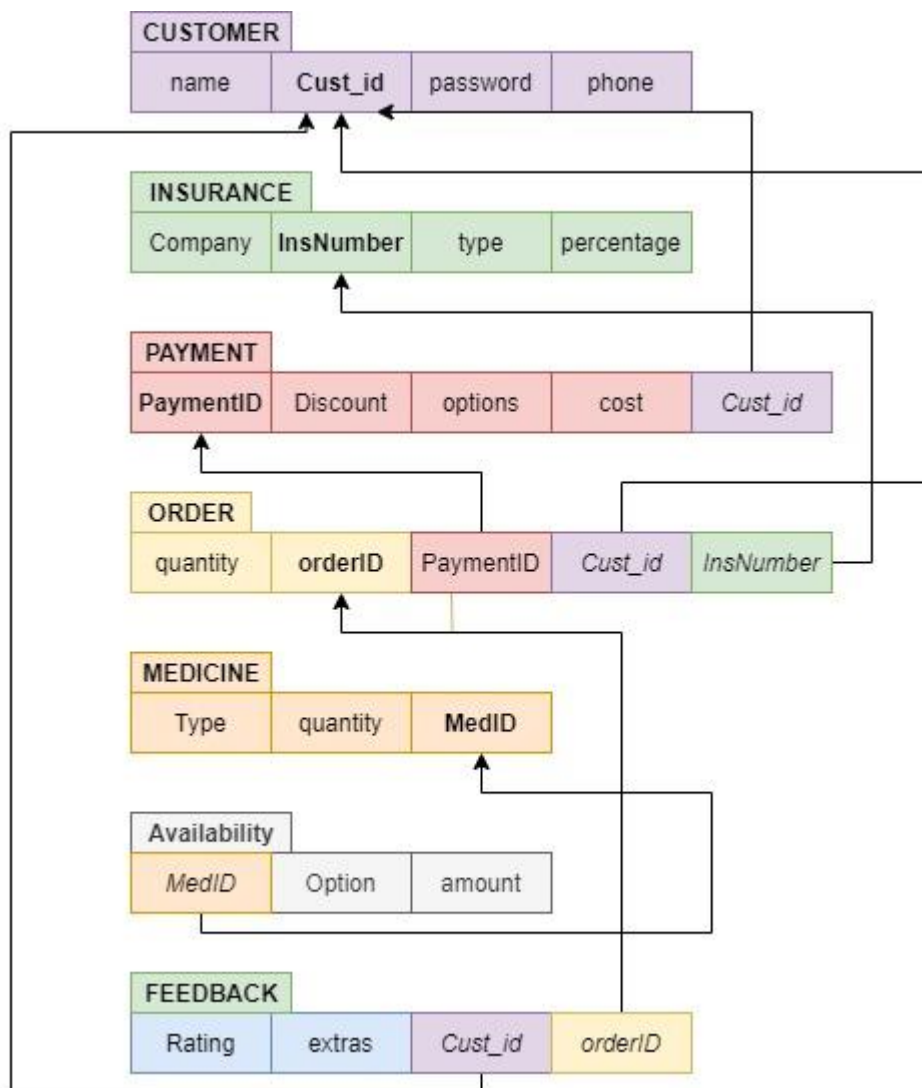
Relational schema and Normalization

Preamble

Normalization is the process in which the data is organized in the database. It is used to minimize repetition from the various relations in the database. It is also used in order to eliminate anomalies like insertion, deletion, and update in the tables. It helps in dividing a large table into multiple small normalized tables that are connected and linked using a relationship to reduce redundancy.

The need for normalization is basically for making sure that a table contains only data directly related to the primary key, that each data field contains only one item of data, and that redundant (duplicated and unnecessary) data is eliminated.

Schema diagram, Normalization and description



Normalization and Description

A relational schema usually is normalized to reduce data redundancies and to reduce insertion/update/deletion anomalies. There are various forms of normalization which may be achieved.

First Normal Form

For a table to be in the First Normal Form (1NF) it must follow the rules given below:

1. Should only have single (atomic) valued attributes.
2. Values stored in a column must adhere to the same domain
3. All columns in a table must have unique names
4. Order of data storage should not matter.

Every column of the tables in the schema above, are atomic, and have unique names, and are not sensitive to the order as stored in the database. Following this, the schema mentioned above in the section already ratifies the 1st Normal Form.

Second Normal Form

For a table to be in the Second Normal Form (2NF) it must follow the rules given below:

1. It should be in 1NF
2. It should not have any partial dependencies

There are no tables in the schema mentioned which use composite keys, and hence, the schema also ratifies the 2nd Normal Form.

Third Normal Form

For a table to be in the Second Normal Form (2NF) it must follow the rules given below:

1. It should be in 2NF
2. Every non-prime attribute of a relation should be non-transitively dependent on every key of itself.

All base entities in our schemas refer to only their primary key (which are not composite), and the relations all do not present any transitive dependencies. This implies that the schema mentioned is already in 3NF form.

Chapter 6

NOSQL

Preamble

NoSQL is a non-relational database that stores and accesses data using key-values. Instead of storing data in rows and columns like a traditional database, a NoSQL DBMS stores each item individually with a unique key. Additionally, a NoSQL database does not require a structured schema that defines each table and the related columns. This provides a much more flexible approach to storing data than a relational database. NoSQL provides an unstructured or "semi-structured" approach that is ideal for capturing and storing user generated content (UGC). This may include text, images, audio files, videos, click streams, tweets, or other data. While relational databases often become slower and more inefficient as they grow, NoSQL databases are highly scalable.

Integrating NOSQL database to SQL with complete system architecture

In any online ecommerce business or shop, customer is the end user and thus customer reviews are important for any online business. So, as NoSQL integration we added a feedback page and thankyou page for our end users, the customer can add feedback after the purchase. We have used NoSQL database to store feedback after the order from a customer, for our project purposes we have used MongoDB.

Key	Title	Form Validation	Data Type
C_id	Feedback	String	Char

Chapter 7

Conclusion & Future Enhancement

This project was successfully built and completed. The project is an online medical shop with two categories of users (admin and customer) who can update inventory and place order respectively. We have also added a web scraper as an innovation to this project. However there are a lot of changes and addition of functionalities that can be done, and which we intend to do after peer and faculty review. Some of the enhancements are :

- Listing of products linked with images to generate a more shop relevant UI
- Remove some programming language constraints
- Online Deployment
- And changes that our faculty and peers suggest.

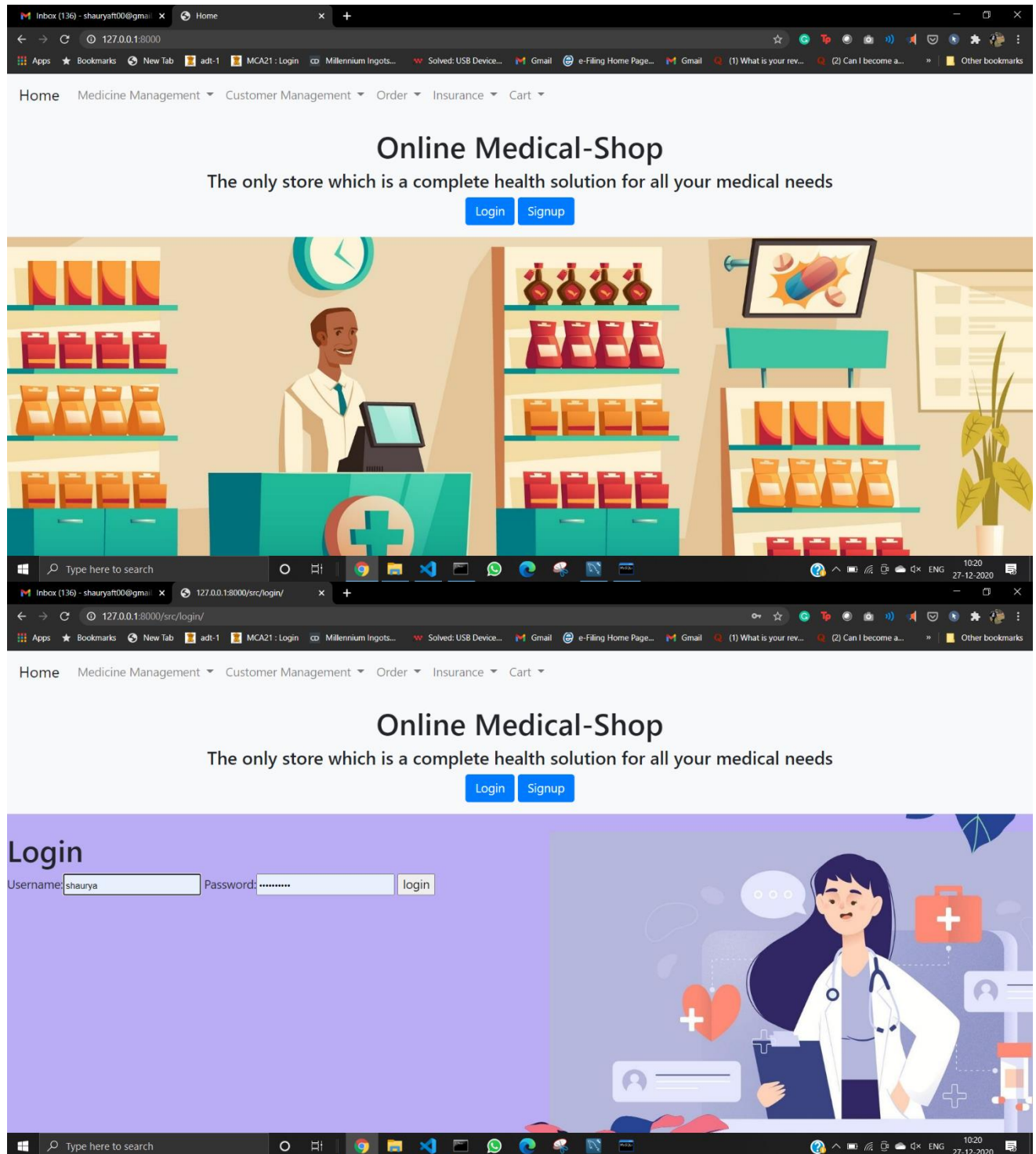
In the end , we would once again thank our college, examiners, faculty guides and teachers to help us finish the project within the speculated timeline.

References

- F. Anjum, A. S. M. Shoaib, A. I. Hossain and M. M. Khan, "Online health care," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2018, pp. 580-583, doi: 10.1109/CCWC.2018.8301617.
- Q. Yu and W. Yang, "The Analysis and Design of System of Experimental Consumables Based on Django and QR code," *2019 2nd International Conference on Safety Produce Informatization (IICSPI)*, Chongqing, China, 2019, pp. 137-141, doi: 10.1109/IICSPI48186.2019.9095914.
- K. I. Satoto, R. R. Isnanto, R. Kridalukmana and K. T. Martono, "Optimizing MySQL database system on information systems research, publications and community service," *2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, Semarang, 2016, pp. 1-5, doi: 10.1109/ICITACEE.2016.7892476.
- K. I. Satoto, R. R. Isnanto, R. Kridalukmana and K. T. Martono, "Optimizing MySQL database system on information systems research, publications and community service," *2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, Semarang, 2016, pp. 1-5, doi: 10.1109/ICITACEE.2016.7892476.
- "Documentation," Django. [Online]. Available: <https://docs.djangoproject.com/en/3.1/>. [Accessed: 12-Oct-2020].
- "MySQL Documentation," MySQL. [Online]. Available: <https://dev.mysql.com/doc/>. [Accessed: 17-Sep-2020].
- "Where Developers Learn, Share, & Build Careers," Stack Overflow. [Online]. Available: <https://stackoverflow.com/>. [Accessed: Nov-2020].

Appendix - Screenshots

The following are the pages for our application.



Inbox (136) - shauryaft00@gmail.com x All Customer Details x +

127.0.0.1:8000/src/custtable/

Home Medicine Management Customer Management Order Insurance Cart

Online Medical-Shop


The only store which is a complete health solution for all your medical needs

logout

CUSTOMER DETAILS

Customer ID	First Name	Last Name	Address	Phone Number	Email
1RV18CS150	SHAURYA	GUPTA	5/57A VIPUL KHAND	7390875221	Shauryagupta.cs18@rvce.edu.in

Click on a Customer ID to Update or Delete the record.



Inbox (136) - shauryaft00@gmail.com x ORDER Details x +

127.0.0.1:8000/src/ordform/

Home Medicine Management Customer Management Order Insurance Cart

Online Medical-Shop

The only store which is a complete health solution for all your medical needs

logout

Add ORDER


Order no:

Quantity:

Customer ID:

Medicine ID:

[Cancel](#) [Add Record](#)



Online Medical-Shop

The only store which is a complete health solution for all your medical needs

[logout](#)

INSURANCE DETAILS

Insurance Number	Customer ID	Company	Percent
522	011	SBI	20

Click on a Medicine ID to Update or Delete the record.

Online Medical-Shop

The only store which is a complete health solution for all your medical needs

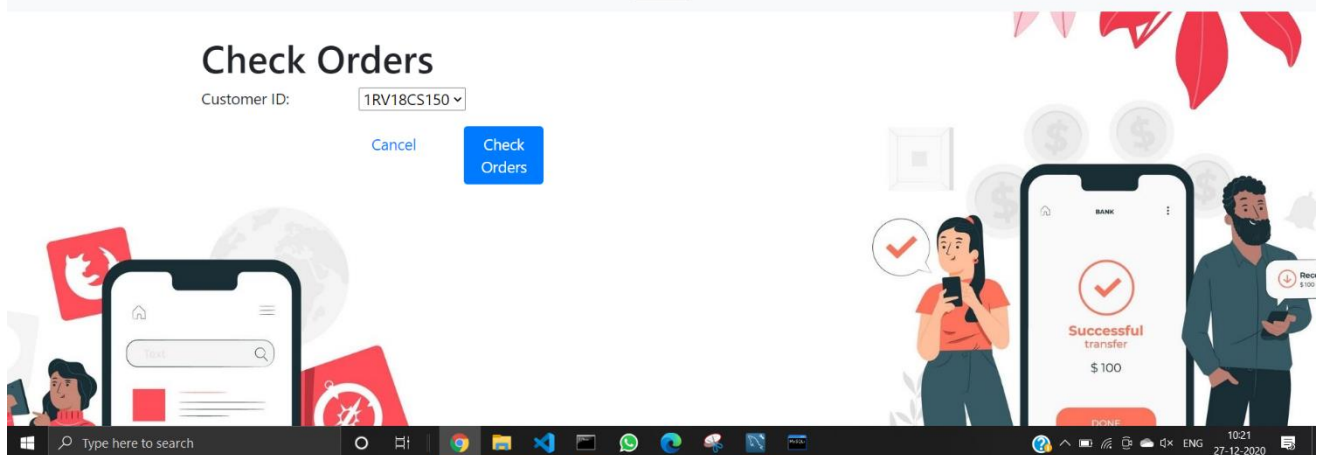
[logout](#)

Check Orders

Customer ID: 1RV18CS150

[Cancel](#)

[Check Orders](#)



Some of the MySQL screenshots

```
MySQL 8.0 Command Line Client - Unicode
3 rows in set (0.32 sec)

mysql> desc src_availability;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | int  | NO   | PRI | NULL    | auto_increment |
| m_id   | varchar(30) | NO   | UNI | NULL    |
| options | int  | NO   |     | NULL    |
| number | int  | NO   |     | NULL    |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)

mysql> desc src_customer;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | int  | NO   | PRI | NULL    | auto_increment |
| c_id   | varchar(30) | NO   | UNI | NULL    |
| fname  | varchar(30) | NO   |     | NULL    |
| lname  | varchar(30) | NO   |     | NULL    |
| address | varchar(100) | NO   |     | NULL    |
| email  | varchar(50) | NO   |     | NULL    |
| phn_no | bigint | NO   | UNI | NULL    |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.02 sec)

mysql> desc src_medicine;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | int  | NO   | PRI | NULL    | auto_increment |
| m_id   | varchar(30) | NO   | UNI | NULL    |
| mname  | varchar(30) | NO   |     | NULL    |
| quantity | int  | NO   |     | NULL    |
| m_type | varchar(30) | NO   |     | NULL    |
| price  | int  | NO   |     | NULL    |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)

mysql> desc src_payment;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | int  | NO   | PRI | NULL    | auto_increment |
| p_id   | varchar(30) | NO   | UNI | NULL    |
| s_id   | varchar(30) | NO   |     | NULL    |
| cost   | int  | NO   |     | NULL    |
| disc   | int  | NO   |     | NULL    |
+-----+-----+-----+-----+-----+-----+

MySQL 8.0 Command Line Client - Unicode
src_medicine
src_orderlist
src_payment
+-----+-----+-----+
16 rows in set (0.04 sec)

mysql> show tables;
+-----+
| Tables_in_medicalshop |
+-----+
| auth_group             |
| auth_group_permissions |
| auth_permission        |
| auth_user              |
| auth_user_groups       |
| auth_user_user_permissions |
| django_admin_log       |
| django_content_type    |
| django_migrations      |
| django_session         |
| src_availability       |
| src_customer           |
| src_insurance          |
| src_medicine           |
| src_orderlist          |
| src_payment            |
+-----+
16 rows in set (0.11 sec)

mysql>
```

MySQL 8.0 Command Line Client - Unicode

16 rows in set (0.11 sec)

mysql> desc django_migrations

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
app	varchar(255)	NO		NULL	
name	varchar(255)	NO		NULL	
applied	datetime(6)	NO		NULL	

4 rows in set (0.45 sec)

mysql> desc auth_permission;

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(255)	NO		NULL	
content_type_id	int	NO	MUL	NULL	
codename	varchar(100)	NO		NULL	

4 rows in set (0.05 sec)

mysql> desc auth_group

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(150)	NO	UNI	NULL	

2 rows in set (0.03 sec)

mysql> desc auth_group_permissions;

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
group_id	int	NO	MUL	NULL	
permission_id	int	NO	MUL	NULL	

3 rows in set (0.32 sec)

mysql>

MySQL 8.0 Command Line Client - Unicode

mysql> desc src_payment;

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
p_id	varchar(30)	NO	UNI	NULL	
c_id	varchar(30)	NO		NULL	
cost	int	NO		NULL	
disc	int	NO		NULL	

5 rows in set (0.03 sec)

mysql> desc src_insurance;

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
c_id	varchar(30)	NO		NULL	
ins_no	int	NO	UNI	NULL	
company	varchar(30)	NO		NULL	
percent	int	NO		NULL	

5 rows in set (0.00 sec)

mysql> desc src_availability;

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
m_id	varchar(30)	NO	UNI	NULL	
options	int	NO		NULL	
number	int	NO		NULL	

4 rows in set (0.00 sec)

mysql> desc src_orderlist;

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
o_id	varchar(30)	NO	UNI	NULL	
c_id	varchar(30)	NO		NULL	
m_id	varchar(30)	NO		NULL	
quantity	int	NO		NULL	
cost	int	NO		NULL	

6 rows in set (0.00 sec)

mysql>