

Project Report
On
Novo Connect

Submitted by
SNEHA BASER
(ID- BTBTC18144)

For the award of the degree of
B. Tech (Computer Science)

Under the supervision of

Mr. Kunal Kumar Kushwaha
SDE II
Essentia Softserv LLP



Apaji Institute of Mathematics and Applied Computer Technology
Banasthali Vidyapith
Banasthali - 304022
Session: 2020-2021

Certificate



Essentia Softserv

B-1 / H-3, Mohan Estate,
Mathura Road, New Delhi,
Delhi 110044

+91-9910589350

info@essentia.email

<https://essentiasoftserv.com>

6th December 2021

TO WHOMSOEVER IT MAY CONCERN

Certified that **Sneha Baser** has carried out the project work titled **Novo** from **20th July 2021** to **30th November 2021** for the award of the **B.Tech Computer Science** from **Essentia Softserv LLP** under my supervision. The thesis embodies result of original work and studies carried out by the student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else.

Yours sincerely

For **ESSENTIA SOFTSERV LLP**

ABSTRACT

Novo Connect is a Doctor Consultancy and Ecommerce application. It provides web based as well as mobile based application. Among the various services it provides, first is where user can seek doctor's appointment or consultant's appointment as per the problem being faced. User can have three different means to book an appointment which are:

- Virtual: where user can book an appointment for virtual consultation.
- InPerson:
 - Doctor visiting patient's location.
 - Patient can visit doctor's clinic.

For ex: User suffering from skin disease , then it can search specialist in Skin and search will generate a list of consultants who is specialized in skin problems , then user can select consultant and time slot according to availability.

User can choose among the three means to book an appointment. After selecting time slot user have to do payment. Second service that Novo Connect provide is where user can buy product as recommended by doctor or consultant. User can search for product , check detail of product and variant and order product as per the requirement. Other services includes organizing events by consultants or doctors revolving around medical awareness which can be registered by user.

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude to my Supervisor, **Mr. Kunal Kumar Kushwaha** for being ever ready to provide his valuable time, support, and constructive guidance during the entire course of my internship. His valuable suggestions and encouragement enabled me and inspired me to handle a project of this enormity with confidence. It has been a pleasure to learn and grow under his mentorship.

I am thankful to each and every person of Essentia Softserv LLP who has contributed in any of the ways in my training.

My stay in the organization has been a great learning experience and a curtain raiser to an interesting and rewarding career. This exposure has enriched me with technical knowledge and has also introduced me to the attributes of a successful professional.

I express my sincere gratitude to **Dr. C.K. Jha** (Head of Department, Computer Science, AIM & ACT, Banasthali Vidyapith, Rajasthan) and members of the staff of my university for their constant encouragement during the project work.

TABLE OF CONTENTS

- Objective (Problem Statement)
- Requirement Analysis (SRS)
 - Requirement specification
 - H/w and S/w Requirements
 - Feasibility Study
 - Product Functions
 - Use-case Diagrams
- System Design (SDS)
 - High-level Design
 - ER Diagram
 - Database Design
 - Data flow diagram
 - Sequence Diagram
- Testing
- User Interfaces
- References

OBJECTIVE

Novo Connect is an attempt to include both services which are doctor consultancy plus online pharmacy. Most often, the consumers are unaware of how they get their medicine and therefore, do not trust the e-pharmacy completely.

Novo Connect works towards keeping all of their customers informed and provides a transparent platform so as to provide the best service possible. Novo Connect also provide platform to doctors and consultants to host an event related to medical awareness.

In Market there are two different kinds of platforms. One support online consultancy with doctors or consultants and other allows to buy medicine. Example of such platform are:

Practo Search - Practo Search is a patient focused, unbiased, independent medical website with over 100,000 doctor profiles from across India and Singapore. Patients can book confirmed appointments with doctors listed on Practo's website.

PharmEasy - Pharmacy made Easy. We are one of the largest pharmacy aggregators in India. We help patients connect with local pharmacy stores and diagnostic centres in order to fulfil their extensive medical needs.

In the existing system, this was done manually. It includes the manual processes of inputting data or handing off a file to the next person who needs to work on the file, running around, keeping track, making sure all the right people signed in all the right places. Completing these tasks can be very frustrating when done manually, i.e., dealing with paper work that needs to be sent, delivered & signed each step of the flow is time consuming and inefficient.

REQUIREMENT ANALYSIS (SRS)

REQUIREMENT SPECIFICATION:

- **Functional Requirements:**

Novo Connect is web based application that covers the gap between user and tedious conservative method of appointment booking as well as ease the method of buying medicine. The user will log in to the website with its login credentials and can access and monitor the workflow according to his right. The availability of this website is up to the Internet connection of the client. It is secure, reliable and consistent.

- The user should have proper login credentials.
- Different access rights are provided according to user.

User Action:

- User can seek doctor's appointment or consultant's appointment as per the problem being faced.
- User can have three different means to book an appointment which are: Virtual and Inperson(Doctor visiting patient's location ,Patient can visit doctor's clinic).
- User can search specialist in particular problem (he/she) faced and search will generate a list of consultants who is specialized in particular problem.
- User can select consultant and book time slot according to availability.
- After selecting time slot user have to do payment.

- Second service that Novo Connect provide is where user can buy product as recommended by doctor or consultant.
- User can search for product , check detail of product and variant and order product as per the requirement
- Other services includes organizing events by consultants or doctors revolving around medical awareness which can be registered by user.

Doctor or Consultant Action :

- Doctor can register themselves according to speciality they possess and timing when they available.
- Doctor can add services and products they provide. Like different kind of therapy, massage, concussion.
- Doctor can check upcoming appointments they have.
- Doctor can suggest suppliers from which users can buy medicines which is prescribed by Doctor only.

H/w and S/w Requirement

Hardware

Server-side requirements:

Requirements	Specification
Processor	Dual Core (2 logical CPUs) or higher, 2 GHz or higher
Random Access Memory (RAM)	4 GB or higher
Hard Disk Capacity	256 GB or higher

Client-side requirements:

Requirements	Specification
Processor	Dual Core (2 logical CPUs) or higher, 2 GHz or higher
Random Access Memory (RAM)	2 GB or higher
Hard Disk Capacity	128 GB primary, or higher

Software

Client-side requirements:

- OS : Any GUI OS
- Browser : Any browser compatible with IE 5.0 or onwards

Server Side:

- OS : Ubuntu 20.04
- Framework : Visual Studio Code
- Back-end : Django
- Database Server : PostgreSQL11
- Browser : Any browser compatible with IE 5.0 or onwards
- Deployment on AWS

Development side requirements:

- OS : Windows 7 & above/ Ubuntu
- Framework : Visual Studio Code
- Front-end : HTML5, CSS3,Vuejs,Nuxt
- Back-end : Python 3.7 (Django Framework)
- Database : PostgreSQL11

Back End:

- Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much

of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Front End:

- HTML5
- CSS3
- Vuejs
- Nuxt

Vue is a **progressive framework** for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.

Nuxt.js is a framework for creating Vue.js applications. Its goal is to help Vue developers take advantage of top-notch technologies, fast, easy and in an organized way.

IDE &Others:

- Postman
- Visual Studio Code

Database:

- PostgreSQL11

PostgreSQL is an advanced, enterprise-class, and open-source relational database system. PostgreSQL supports both SQL (relational) and JSON (non-relational) querying.

PostgreSQL is a highly stable database backed by more than 20 years of development by the open-source community.

PostgreSQL is used as a primary database for many web applications as well as mobile and analytics applications.

Feasibility Study

Feasibility is the measure of how beneficial or practical the development of the system will be to the organization. It is a preliminary survey for the systems investigation. It aims to provide information to facilitate a later in-depth investigation. The report produced at the end of the feasibility study contains suggestions and reasoned arguments to help management decide whether to commit further resources to the proposed project.

- Economic feasibility: an evaluation of development cost weighted against the ultimate income or benefit derived from the developed system.
- Technical feasibility: a study of function, performance and constraints that may affect the ability to achieve an acceptable system.
- Operational feasibility: a study about the operational aspects of the system.
- Schedule feasibility: an evaluation of the time taken by the project to complete

Economic Feasibility:

It looks at the financial aspects of the project. It determines whether the management has enough resources and budget to invest in the proposed system and estimated time for the recovery of cost incurred. Economic feasibility is determined by the means of cost benefits analysis. Cost Benefit Analysis delineates costs for the project development and weighs them against tangible and intangible benefits of a system. The proposed system is economically feasible because it did not involve purchasing

of any hardware or software.

The project's economic feasibility:

- There is little to no cost incurred for the development of this project since most of the tools/technologies used are open source.
- Very less cost is required for the maintenance.
- The system is financially feasible since user pay according to consumption.

Technical Feasibility:

It is the assessment based on an outline design of system requirements, to determine whether the organization has the technical expertise to handle completion of the project. This project is being executed as per the current business requirements. We have the required technology, software, hardware, manpower, testers and debuggers to complete the project in stipulated time. Complete and proper analysis of the whole project has been done beforehand. The technology used are stable and established, readily available and easy to work with.

This web-based project has been developed using Django framework. It contains a base workspace and an extensible plug-in system for customizing the environment. It has the advantage that it is an open source platform and the developer can inspect and make changes according to the client's requirements. User management and content management is quite easy and provides better user experience. Hence, no overhead incurs while using Django as our mainstream framework in this project. We are using vue.js for frontend.

Vue.js offers simplicity, performance, documentation, faster development, easy upgrades, and much more. Vue.js is a progressive JavaScript framework that is used to build the user interface of modern applications without using a lot of resources. Vue.js mainly focus on the view layer, which enables developers to integrate it into the existing projects without any hassle. Vue.js is a perfect choice for building single-page applications (SPA). The Vue.js framework offers ready-to-use plugins that solve many critical issues in application development. Vue also provides amazing tools and efficient state management and routing options. Vue.js is backed by an active growing community of dedicated. The biggest reason behind choosing Vue.js is due to its less learning curve and support for virtual DOM.

Operational Feasibility:

The project is operationally feasible. This project is being made for the convenience of users. It will greatly reduce the manual work. The project satisfies the requirements identified in the requirements analysis phase of system development.

The project's operational feasibility:

- There was very good response from top management for development of the project.
- Besides it, no harm could be seen to developers or to other environment while developing the project.
- The website provides smooth graphical interface to the users.

Schedule Feasibility:

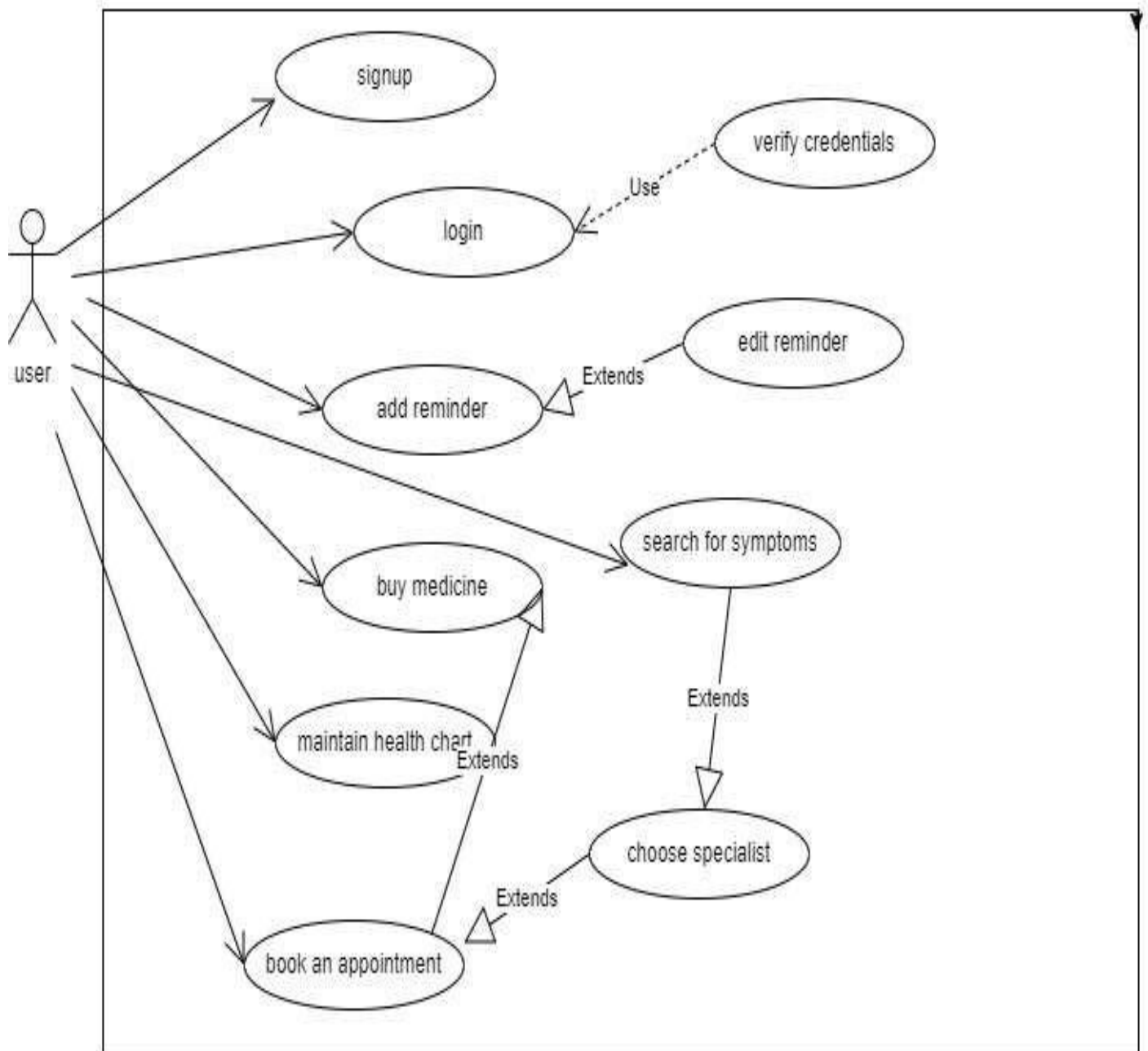
Since all the requirement analysis is done before the start of the project, the

project is scheduled timely and the tasks are achieved on or before deadlines.

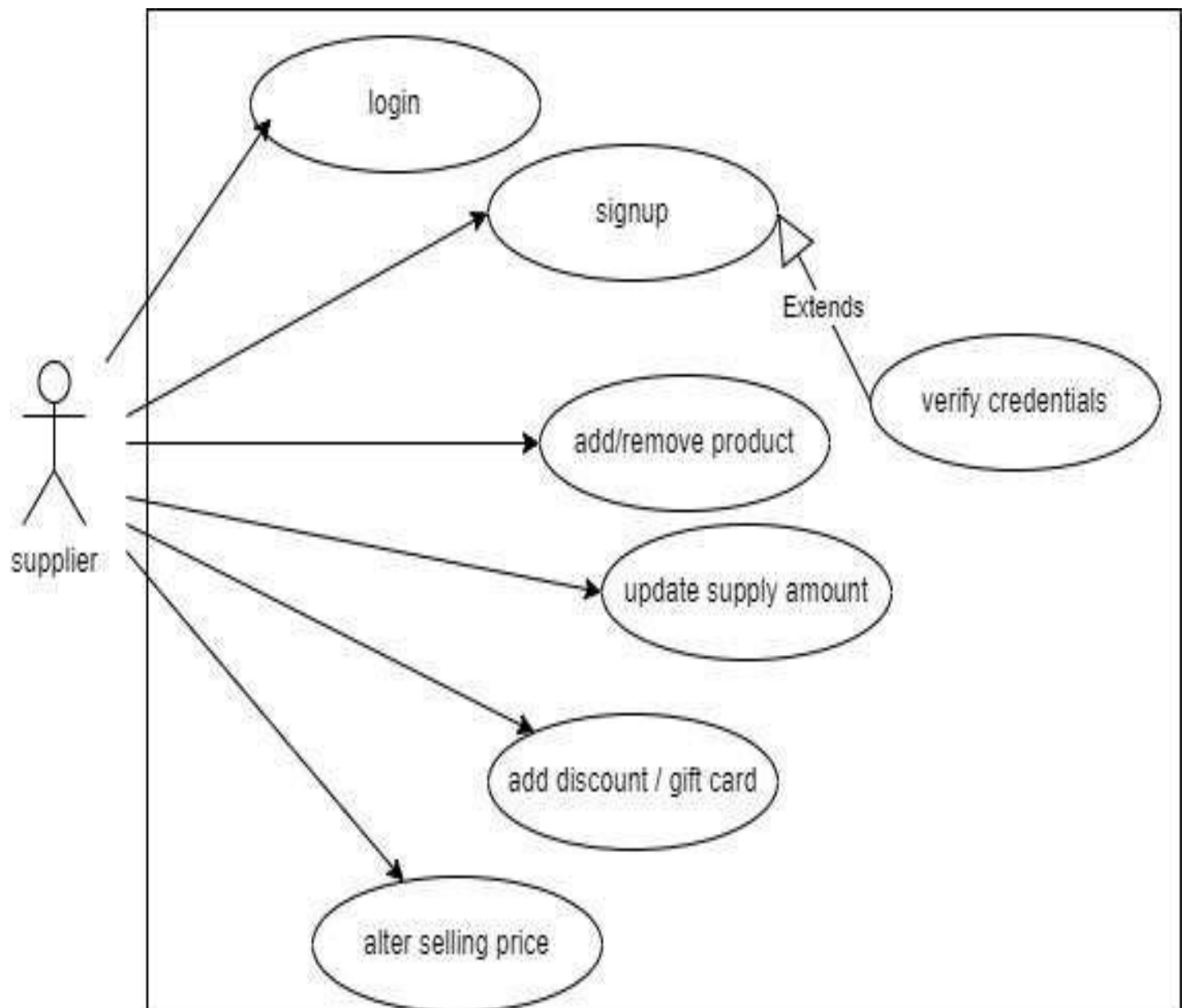
Product Functions

- Authentication: Allows only the authenticated users to login.
- User can seek doctor's appointment or consultant's appointment as per the problem being faced.
- User can have three different means to book an appointment which are: Virtual and Inperson(Doctor visiting patient's location ,Patient can visit doctor's clinic).
- User can search specialist in particular problem (he/she) faced and search will generate a list of consultants who is specialized in particular problem.
- User can select consultant and book time slot according to availability.
- After selecting time slot user have to do payment.
- Second service that Novo Connect provide is where user can buy product as recommended by doctor or consultant.
- User can search for product , check detail of product and variant and order product as per the requirement.
- Other services includes organizing events by consultants or doctors revolving around medical awareness which can be registered by user.
- Doctor can register themselves according to speciality they possess and timing when they available.
- Doctor can add services and products they provide. Like different kind of therapy, massage, concussion.
- Doctor can suggest suppliers from which users can buy medicines which is prescribed by Doctor only.
- Doctor can check upcoming appointments they have.

Use Case Diagram



Use case for user



Use case for Supplier

SYSTEM DESIGN

Software Design Descriptions

Introduction

This document is the complete product design specification for “**NOVO CONNECT**”

Purpose

The purpose of SDD is to show how application will be structured to fulfil the requirement identified in the SRS. This SDS contains a statement of the design of the “**NOVO CONNECT**”.

Scope

This SDD is a representation of a system that is used as medium for communicating software design information. This document describes the several interfaces provided to user. It provides the information about different modules. It depicts the sequence of process in the system. This document works as a blueprint of defining the functionality of the system.

Definitions and acronyms

Acronyms

SDD-Software Design Description

SRS-Software Requirement Specifications DFD-
DataFlow Diagram

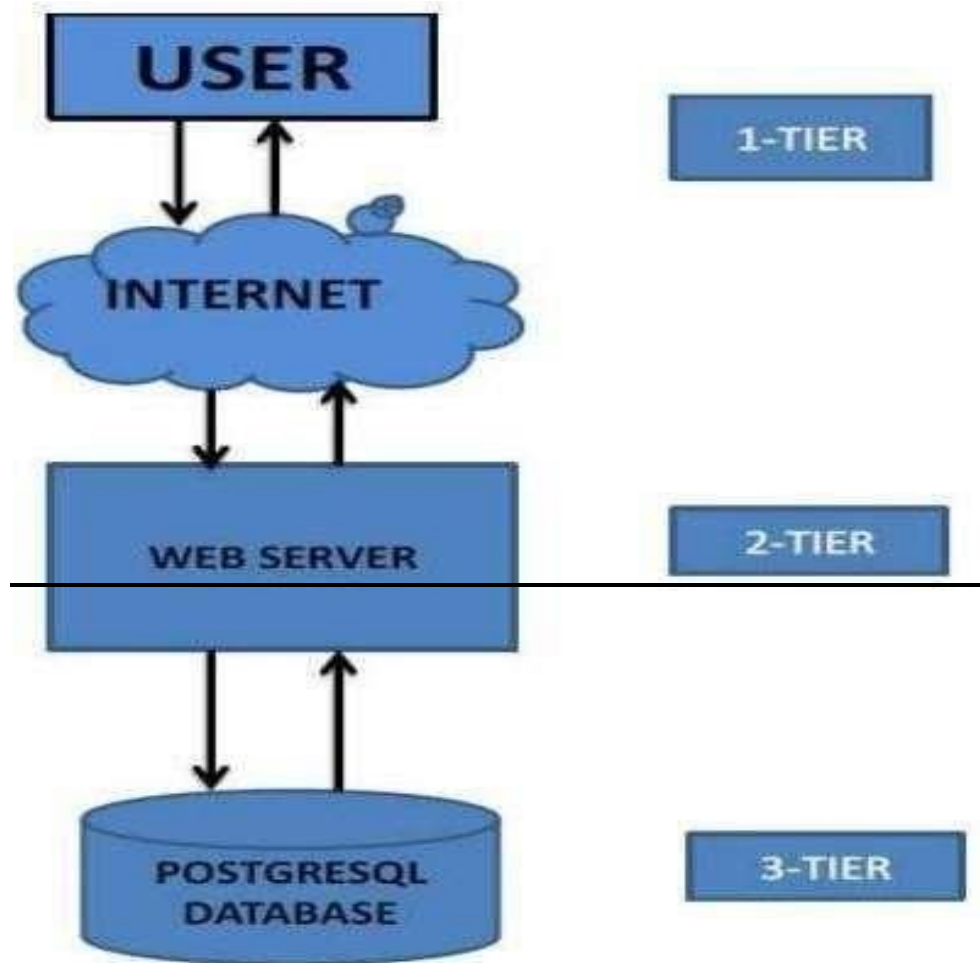
High Level Design

Web Application

The components of the application as can be viewed at a conceptual high level are: user, internet and a Database.

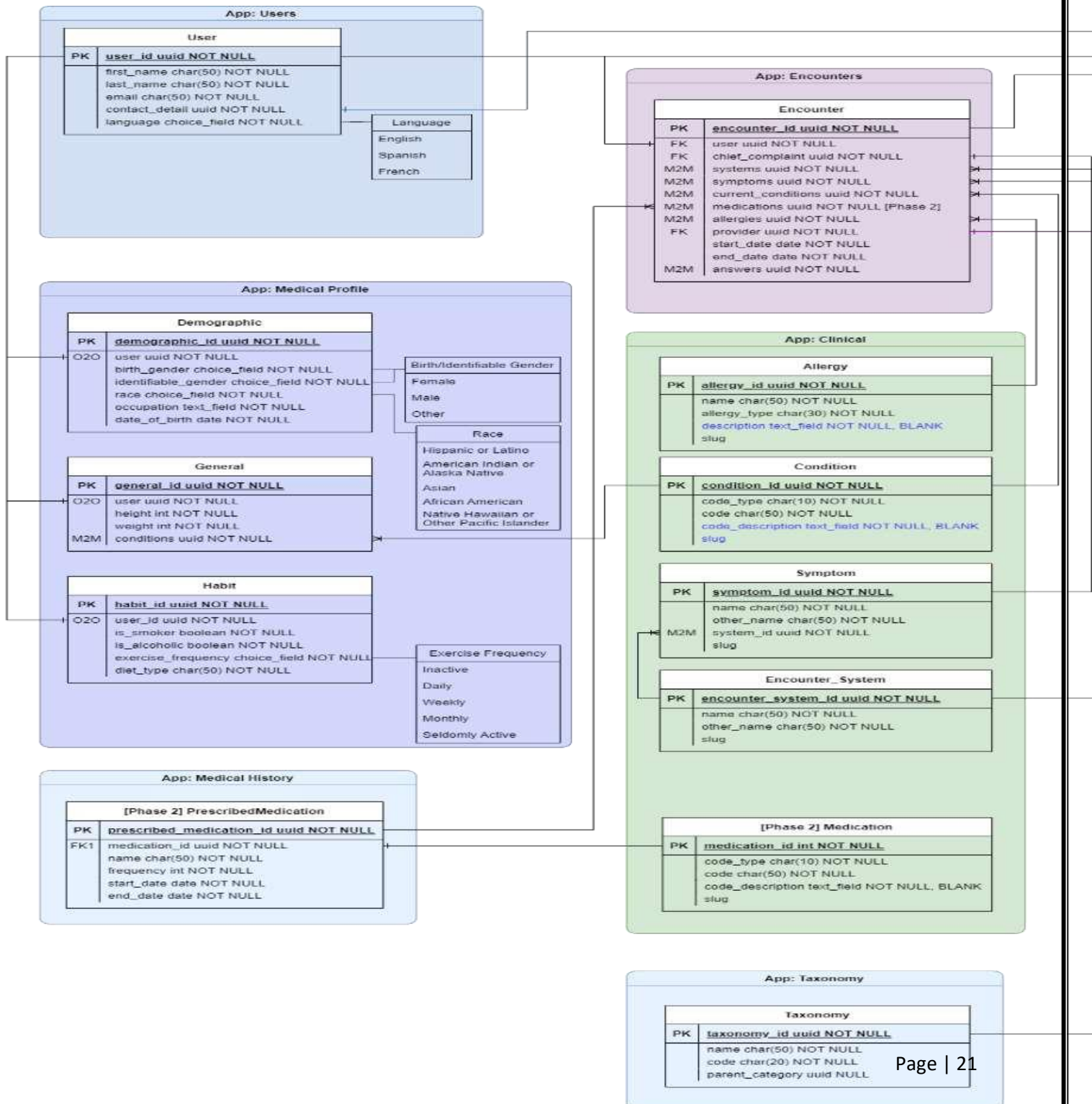
The pictorial representation of the components of our application as can be

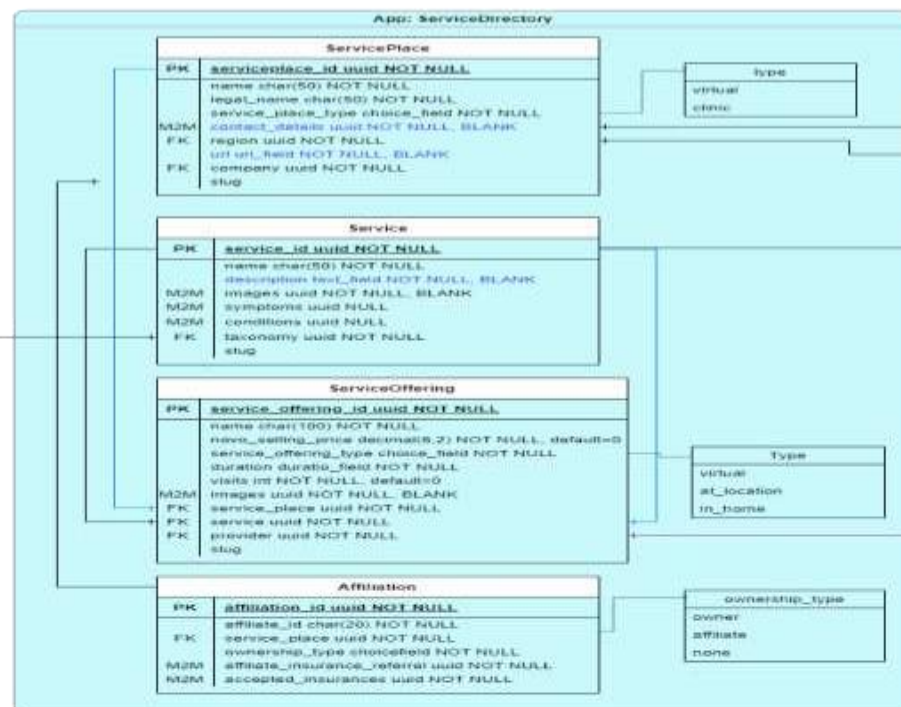
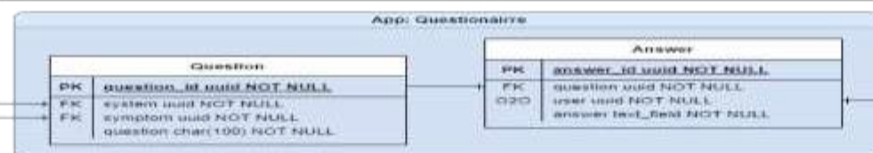
viewed at the high level is as follows:

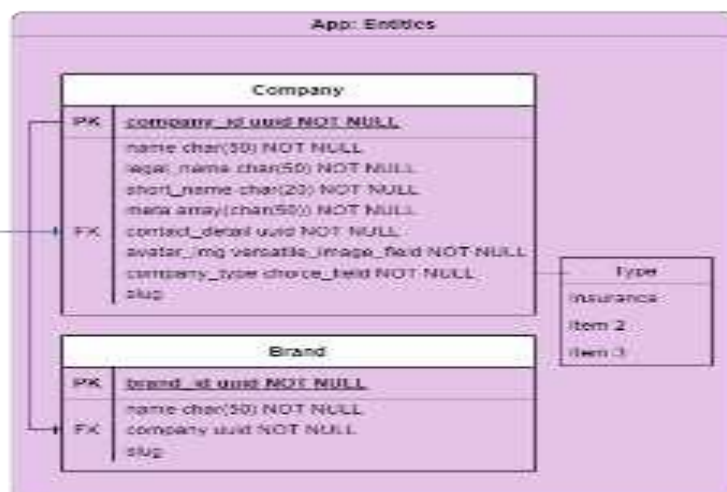
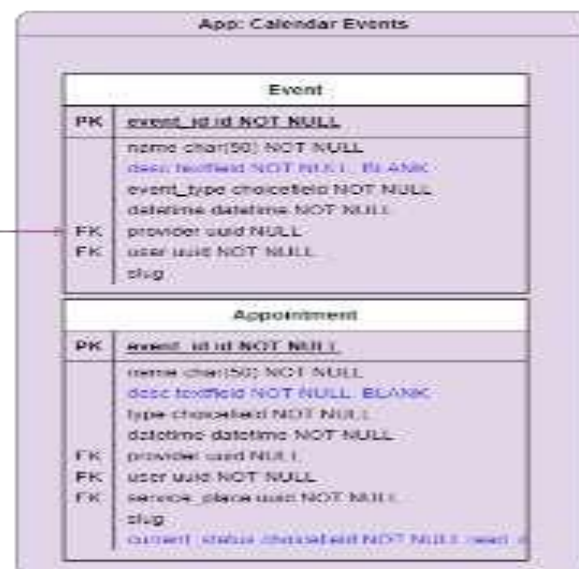
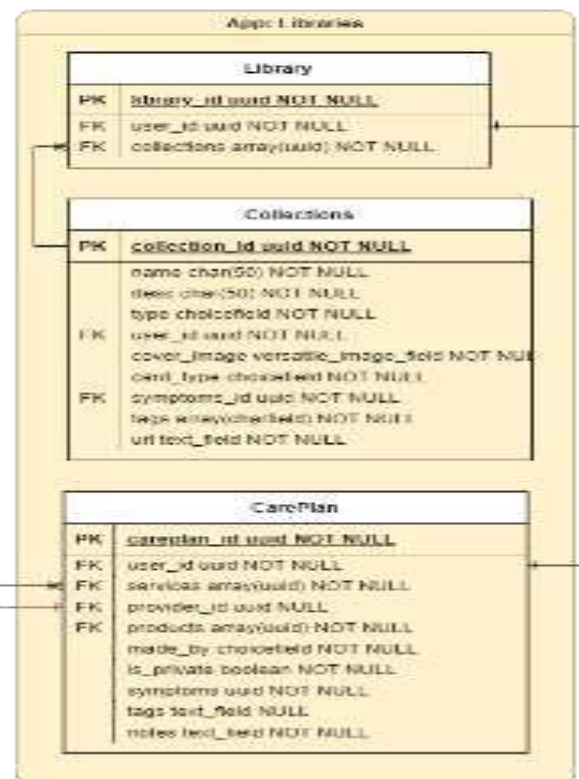
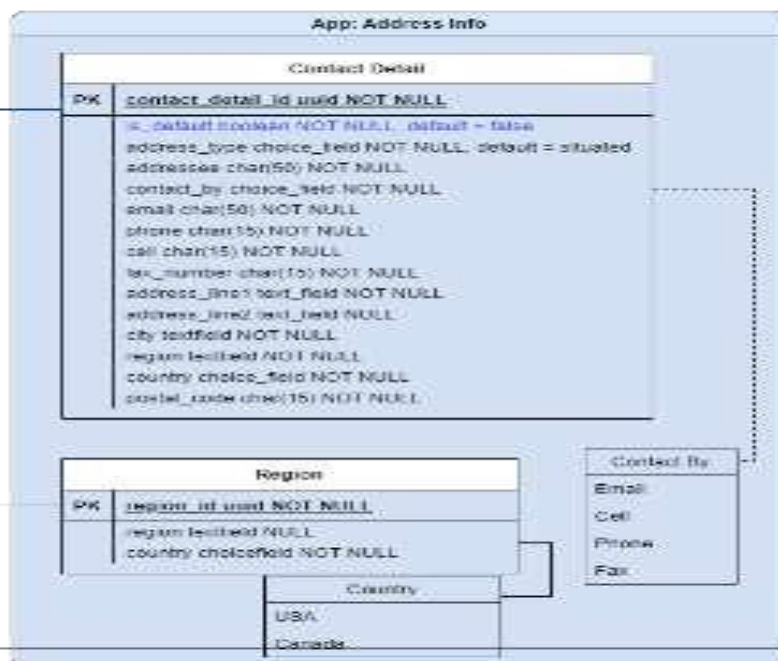


CLASS DIAGRAM:

Class Diagram is vertically split into four parts for better visibility that is placed one after another.







App: Libraries

Item	
PK	<u>item_id uuid NOT NULL</u>
	item_type choice_field NOT NULL
FK	product uuid NULL
FK	service uuid NULL
FK	brand uuid NULL
FK	company uuid NULL

CuratorAlias	
PK	<u>id uuid NOT NULL</u>
	name char(100) NOT NULL, NOT BLANK
	cover_image versatile_image_field NOT NULL
	description char(255) NOT NULL, BLANK
	slug

List	
PK	<u>list_id uuid NOT NULL</u>
	name char(50) NOT NULL
	cover_image versatile_image_field NOT NULL
	parent uuid NOT NULL, NOT BLANK
M2M	items uuid NOT NULL, BLANK
	slug

Curator	
PK	<u>id uuid NOT NULL</u>
	curator_type choice_field NOT NULL
FK	user uuid NULL
FK	provider uuid NULL
FK	company uuid NULL
FK	curator_alias uuid NULL

Channel	
PK	<u>list_id uuid NOT NULL</u>
	name char(50) NOT NULL
	cover_image versatile_image_field NOT NULL
	parent uuid NOT NULL, NOT BLANK
M2M	lists uuid NOT NULL, BLANK
	slug

Collections	
PK	<u>collection_id uuid NOT NULL</u>
	name char(50) NOT NULL
	description textfield NOT NULL, BLANK
	collection_type choicefield NOT NULL
FK	user uuid NOT NULL
	cover_image versatile_image_field NOT NULL
	card_type choicefield NOT NULL
M2M	symptoms uuid NOT NULL, BLANK
	tags array(charfield) NOT NULL, BLANK
	url url_field NOT NULL, BLANK
M2M	channels uuid NOT NULL, BLANK
	slug

CarePlan	
PK	<u>id uuid NOT NULL</u>
FK	user uuid NOT NULL
FK	provider uuid NULL
M2M	services uuid NOT NULL, BLANK
M2M	products uuid NOT NULL, BLANK
	made_by choicefield NOT NULL
	is_private boolean NOT NULL
M2M	symptoms uuid NOT NULL, BLANK
	tags array(char(10)) NOT NULL, BLANK
	notes text_field NOT NULL, BLANK
M2M	encounters uuid NOT NULL, BLANK

Library	
PK	<u>id uuid NOT NULL</u>
FK	user uuid NOT NULL
M2M	products uuid NOT NULL, BLANK
M2M	services uuid NOT NULL, BLANK
M2M	companies uuid NOT NULL, BLANK
M2M	brands uuid NOT NULL, BLANK
M2M	lists uuid NOT NULL, BLANK
M2M	care_plans uuid NOT NULL, BLANK
M2M	service_places uuid NOT NULL, BLANK
M2M	providers uuid NOT NULL, BLANK

DATABASE DESIGN:

tbRegion		
Column Name	Data type	Nullable
Name	Varchar	False
Abbr	Varchar	True
Country	Varchar	False

tbCity		
Column Name	Data type	Nullable
Name	Varchar	False
Region	Foreign key	False
Longitude	Integer	True
Latitude	Integer	true

tbContactDetail		
Column Name	Data type	Nullable
Contact_by	Varchar	False
Phone	Integer	True
Cell	Integer	True
Email	Varchar	True
Fax_number	Varchar	True

tbAddress		
Column Name	Data type	Nullable
Name	Varchar	True
Addresstype	Varchar	False
Address_line1	Varchar	False
Address_line2	Varchar	True
Cities	Varchar	False
City	Foreign key	True
Postal_code	Varchar	False

Phone	Integer	True
Is_default	Boolean	False
Longitude	Integer	True
Latitude	Integer	True

tbAvaibility		
Column Name	Data type	Nullable
Start_time	Datetime	False
End_time	Datetime	False
Slot_date	Date	False
Provider	Foreign key	False
Service_place	Foreig key	False

tbTimeslot		
Column Name	Data type	Nullable
Start_time	Datetime	False
End_time	Datetime	False
Is_available	Boolean	False
Availability	Foreign key	False

tbSubscribedEvent		
Column Name	Data type	Nullable
User	Foreign key	False
Event	Foreign key	False

tbAppointment		
Column Name	Data type	Nullable
name	Varchar	False
Description	Varchar	True
Appointment_type	Varchar	False
Appointment_url	Varchar	True
Start_time	Datetime	False

End_time	Datetime	False
Appointment_date	Date	False
User	Foreign key	False
Service_offering_variant	Foreign key	True
Provider	Foreign key	True
Slug	Varchar	False

tbAppointmentStatus		
Column Name	Data type	Nullable
Appointment	Foreign key	False
State	Varchar	False

tbReminder		
Column Name	Data type	Nullable
User	Foreign key	False
Name	Varchar	False
Datetime	Datetime	False
Note	Varchar	True

tbTax		
Column Name	Data type	Nullable
Code	Varchar	False
Name	Varchar	False
Desc	Varchar	True
Percentage	Integer	False
Region	Foreign key	False

tbProductFulfilmentLine		
Column Name	Data type	Nullable
User	Foreign key	False
Product_offering	Foreign key	False
Quantity	Integer	Default value as 0

Net_amt	Integer	Default value as 0
Discount	Foreign key	True
Tax_amt	Integer	Default value as 0
Gross_amt	Integer	Default value as 0
Current_status	Varchar	False

tbProductFulfilmentLineStatus		
Column Name	Data type	Nullable
Product_fulfilment_line	Foreign key	False
State	varchar	False

tbProductOfferingWishlist		
Column Name	Data type	Nullable
User	Foreign key	False
Product_offering	Foreign key	False

tbEncounterSystem		
Column Name	Data type	Nullable
Name	Varchar	False
Other_name	Varchar	False
Slug	Varchar	False

tbSymptom		
Column Name	Data type	Nullable
Name	Varchar	False
Other_name	Varchar	False
Encounter_systems	Foreign key	False

tbAllergy		
-----------	--	--

Column Name	Data type	Nullable
Name	Varchar	False
Allergy_type	Varchar	False
Description	Varchar	False

tbCondition		
Column Name	Data type	Nullable
Name	Varchar	False
Consumer_name	Varchar	False
User	Foreign key	False
Description	Varchar	False

tbEncounter		
Column Name	Data type	Nullable
User	Foreign key	False
Chief_complaint	Foreign key	False
Systems	Foreign key	False
Symptoms	Foreign key	False
Current_cnditions	Foreign key	False
Allergies	Foreign key	False
Provider	Foreign key	False
Start_date	Datetime	False
End_date	Datetime	False
Answers	Foreign key	False

tbCompany		
Column Name	Data type	Nullable
Name	Varchar	False
Legal_name	Varchar	False
Short_name	Varchar	False
Contacts	Foreign key	True
Addresses	Foreign key	False
Logo	Blob	True
Banner_img	Blob	True

Company_type	Varchar	False
Description	Varcar	False

tbBrand		
Column Name	Data type	Nullable
Name	Varchar	False
Company	Foreign key	False
Images	Foreign key	True

tbEvent		
Column Name	Data type	Nullable
Name	Varchar	False
Description	Varchar	True
Event_type	Varchar	False
Event_url	Varchar	True
Datetime	Datetime	False
Provider	Foreign key	Alse
User	Foreign key	False
Address	Foreign key	False

tbCuratorAlias		
Column Name	Data type	Nullable
Name	Varchar	False
Images	Blob	True
Description	Varchar	False

tbCurator		
Column Name	Data type	Nullable
Curator_type	Varchar	False
User	Foreign key	False
Provider	Foreign key	False
Company	Foreign key	False

Curatorialias	Foreign key	False
---------------	-------------	-------

tblItem		
Column Name	Data type	Nullable
Item_type	Varchar	False
Product	Foreign key	True
Service	Foreign key	True
Service_offerings	Foreign key	True
Brand	Foreign key	True
Company	Foreign key	True
Provider	Foreign key	True
List_item	Foreign key	True
Service_place	Foreign key	True
Taxonomy	Foreign key	True
Featured	Boolean	False

tblList		
Column Name	Data type	Nullable
Name	Varchar	False
Curator	Foreign key	True
Items	Foreign key	True

tbChannel		
Column Name	Data type	Nullable
Name	Varchar	False
Curator	Foreign key	True
Lists	Foreign key	False

tbCollection		
Column Name	Data type	Nullable
Name	Varchar	False
Description	Varcar	False

Collection_type	Varchar	True
User	Foreign key	False
Card_type	Varchar	False
Symptoms	Foreign key	True
Channels	Foreign key	true

tbCarePlan		
Column Name	Data type	Nullable
User	Foreign key	False
Provider	Foreign key	False
Services	Foreign key	True
Products	Foreign key	True
Is_private	Boolean	False
Notes	Varchar	True
Encounters	Foreign key	True

tbLibrary		
Column Name	Data type	Nullable
User	Foreign key	False
Product	Foreign key	True
Services	Foreign key	True
Companies	Foreign key	True
Brands	Foreign key	True
Lists	Foreign key	True
Care_plan	Foreign key	True
Service_places	Foreign kry	True
Providers	Foreign keys	True

tbLibraryOps		
Column Name	Data type	Nullable
Instance_type	Varchar	False
Instace_id	varchar	False
Library	Foreign key	false

tbDemographic		
Column Name	Data type	Nullable
User	Foreign key	False
Birth_gender	Varchar	False
Race	Varchar	False
Occupation	Varchar	False
Date_of_birth	Fate	False

tbGeneralUserDetails		
Column Name	Data type	Nullable
User	Foreign key	False
Height	Integer	False
Weight	Integer	False
Conditions	Foreign key	True

tbHabit		
Column Name	Data type	Nullable
User	Foreign key	False
Is_smoker	Boolean	False
Is_alcoholic	Boolean	False
Exercise_frequency	Varchar	False
Diet_type	Varchar	False

tbDiscount		
Column Name	Data type	Nullable
Type	Varchar	False
Code	Varchar	False
Name	Varchar	False
Desc	Varchar	True
Applicable_from	Datetime	False
Applicable_till	Datetime	False
Region	Foreign key	False
Currency	Varchar	False

Limit_usage	Integer	False
Times_used	Integer	False
Min_checkout_items_qnt	Integer	False
Min_spent_amt	Integer	False
Apply_once_per_order	Boolean	False
Apply_once_per_customer	Boolean	False
Apply_alone	Boolean	False
Is_active	Boolean	False
Max_disc_amt	Integer	True
Fixed_discount	Integer	True
Perc_discount	Integer	True

tbGiftCard		
Column Name	Data type	Nullable
User	Foreign key	False
Code	Varchar	False
Applicable_from	Datetime	False
Applicable_till	Datetime	False
Last_used_on	Datetime	False
Is_active	Boolean	False

tbPayment		
Column Name	Data type	Nullable
Payment_mode	Varchar	False
Payment_gateway	Varchar	False
Amt_to_pay	Integer	False
Amt_paiid	Integer	False
Amt_taken	Integer	False
Currency	Varchar	False
Charged_at	Datetime	False

tbProduct		
-----------	--	--

Column Name	Data type	Nullable
Name	Varchar	False
Description	Varchar	True
Alias	Varchar	True
Is_digital	Boolean	False
Is_custom	Boolean	True
Supplier	Foreign key	True
Brand	Foreign key	True
Symptoms	Foreign key	True
Conditions	Foreign key	True
Taxonomy	Foreign key	True

tbProductOffering		
Column Name	Data type	Nullable
Name	Varchar	False
Alias	Varchar	True
Supplier_sku	Varchar	False
Is_digital	Boolean	False
Is_custom	Boolean	True
Qty_per_unit	Integer	False
Units_per_shipping_pack	Integer	False
Novo_selling_price	Integer	False
Is_shippable	Boolean	False
Shipping_pack_height	Integer	False
Shipping_pack_width	Integer	False
Product	Foreign key	False
Is_taxable	Boolean	False

tbSpeciality		
Name	varchar	False
Description	varchar	false

tbProvider		
Column Name	Data type	Nullable
Credentials	Varchar	False
First_name	varchar	False
Last_name	Varchar	false
Display_name	Varchar	False
Provider_type	Varchar	False
Details	Varchar	True
Contacts	Foreign key	True
Addresses	Foreign key	True
Education	Foreign key	True
Languages	Varchar	True
Specialties	Foreign key	False
Region	Foreign key	True

tbQuestion		
Column Name	Data type	Nullable
System	Foreign key	False
Symptom	Foreign key	False
Question	Varchar	False

tbAnswer		
Column Name	Data type	Nullable
Question	Foreign key	False
Answer	varchar	False
User	Foreign key	False

tbServicePlace		
Column Name	Data type	Nullable
Name	Varchar	False
Legal_name	varchar	False
Service_place_type	varchar	False
Contacts	Foreign key	True
Addresses	Foreign key	True

Region	Foreign key	True
Company	Foreign key	True

tbAffiliation		
Column Name	Data type	Nullable
Ownership	Varchar	False
Affiliation_id	varchar	False
Service_place	Foreign key	False
Affiliated_insuarncce_referrals	Foreign key	True

tbServiceGroup		
Column Name	Data type	Nullable
Name	Varchar	False
Desc	varchar	False
Symptoms	Foreign key	True
Conditions	Foreign key	True
Taxonomy	Foreign key	True

tbService		
Column Name	Data type	Nullable
Novo_selling_price	Integer	False
Service_offering_price	varchar	False
Duration	Varchar	True
Service_place	Foreign key	True
Service	Foreign key	True
Provider	Foreign ke	True

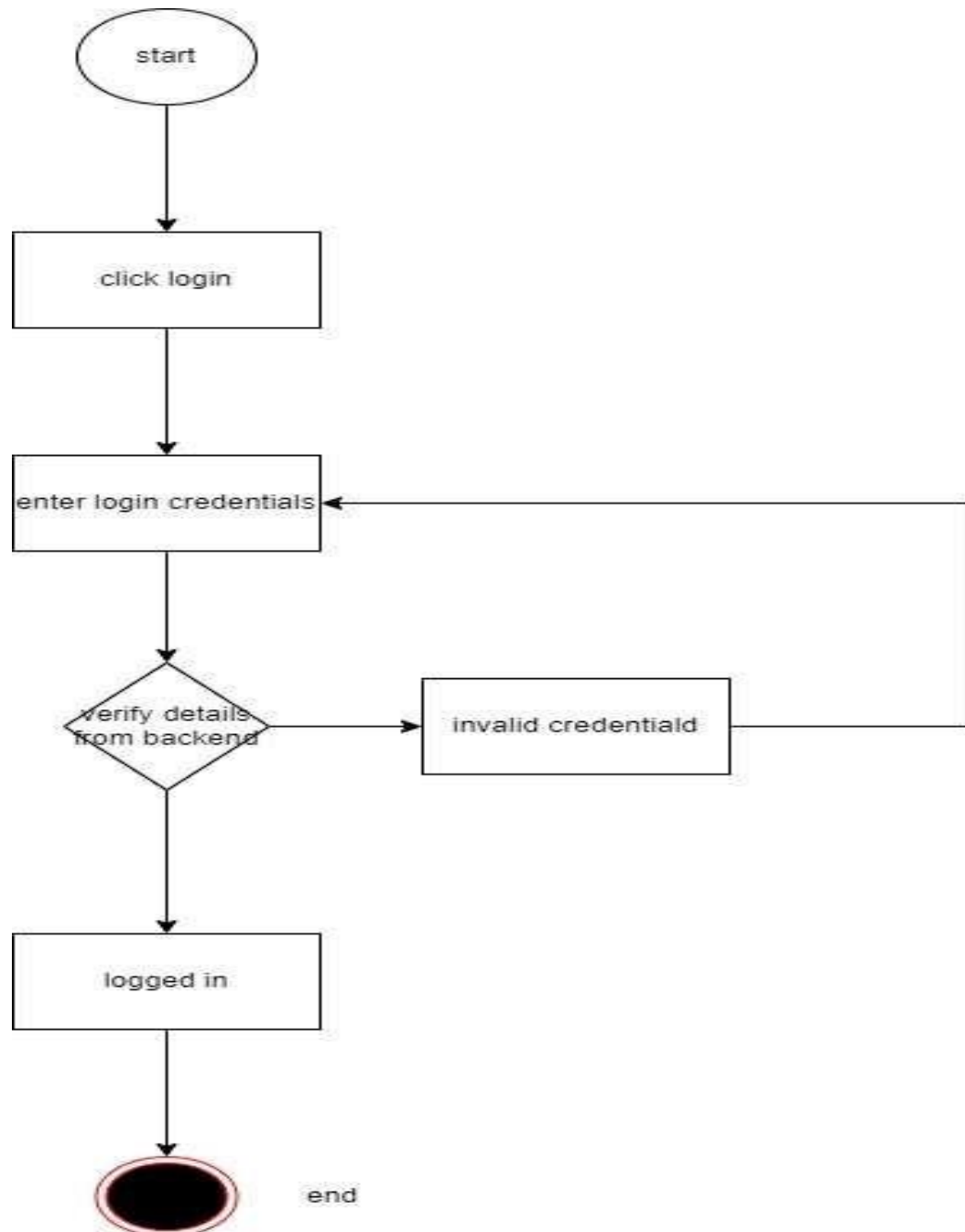
tbUser		
Column Name	Data type	Nullable
First_name	Varchar	False
Last_name	varchar	False
Email	Varchar	False

Contacts	Foreign key	True
Addresses	Foreign key	True
City	Foreign key	True
Membership	Varchar	False
Date_joined	Date	False
Email_verified	Date	False
Is_staff	Boolean	False
Is_active	Booleam	False

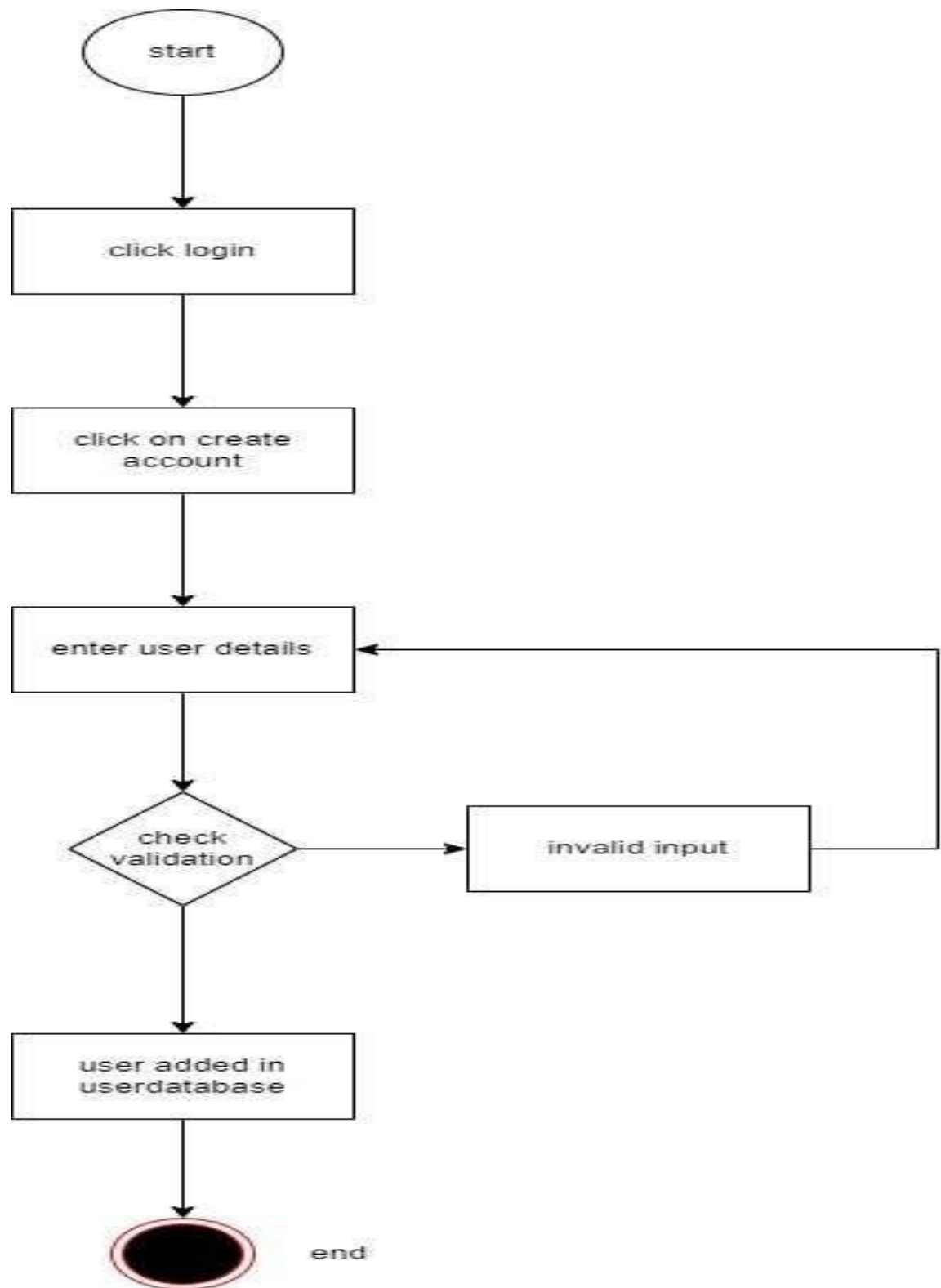
tbTaxonomy		
Column Name	Data type	Nullable
Name	Varchar	False
Code	Varchar	False
Parent_collection	Foreign key	True
Service	Foreign key	True

ACTIVITY DIAGRAM:

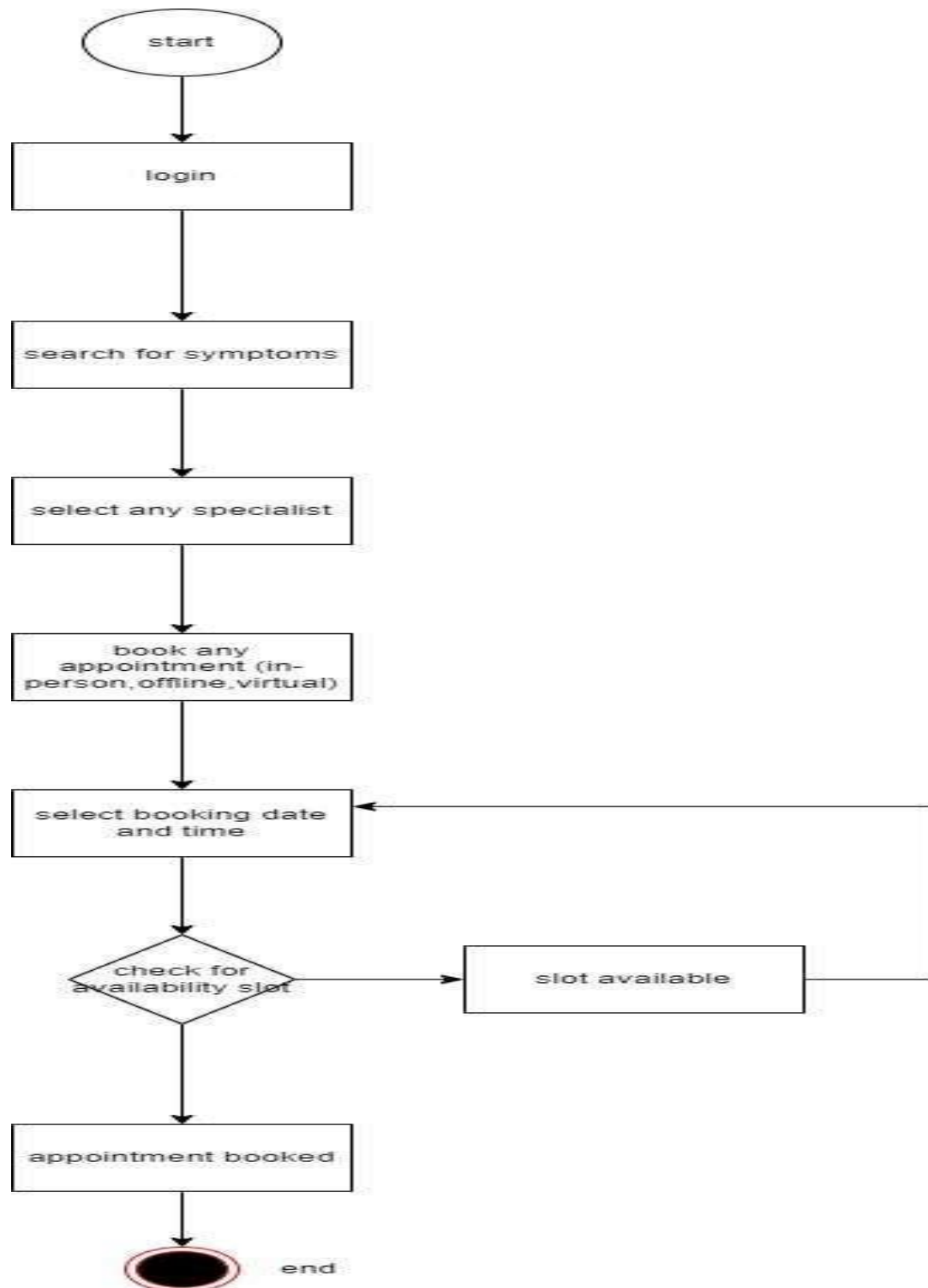
1. LOGIN



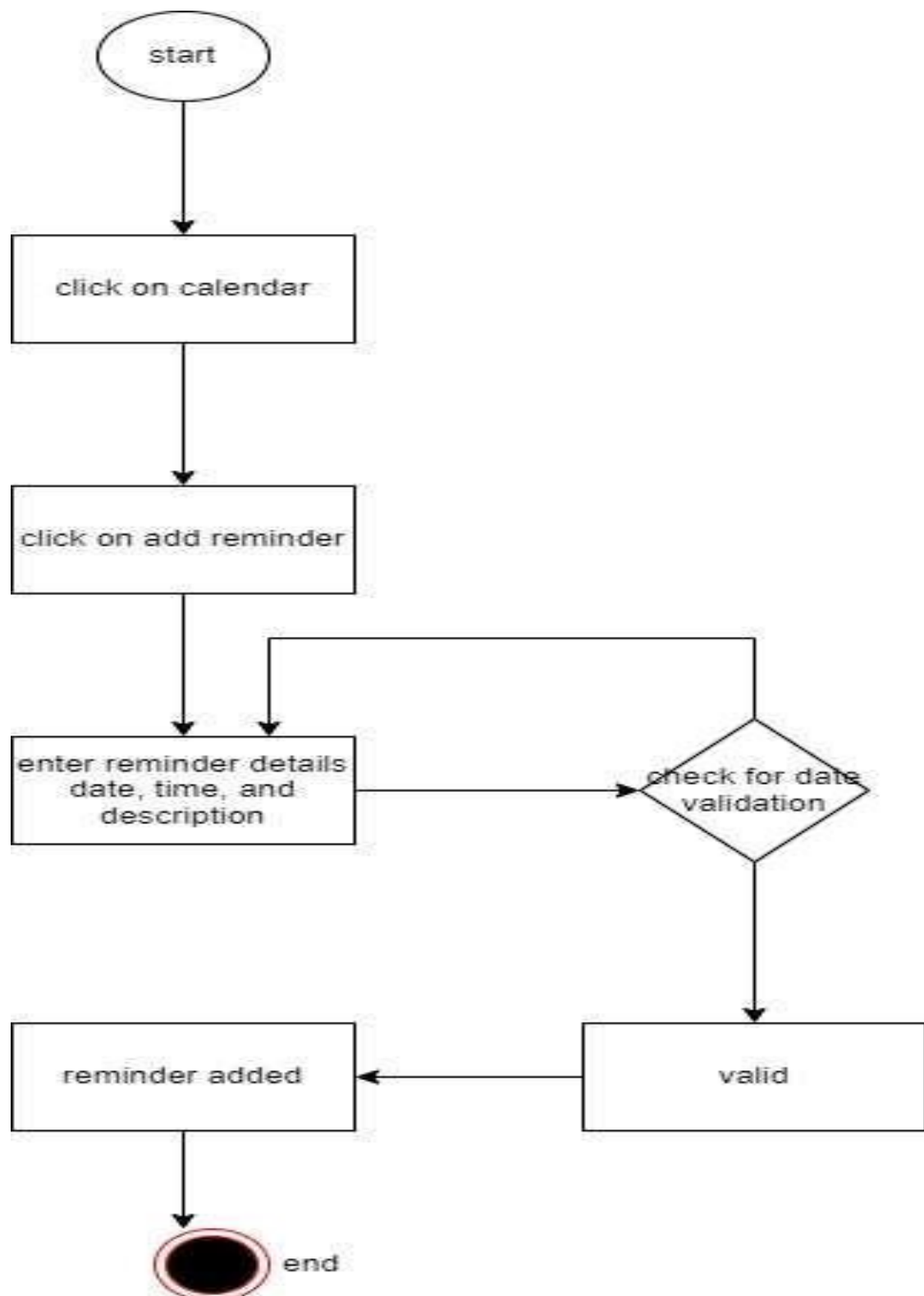
2. SIGN UP



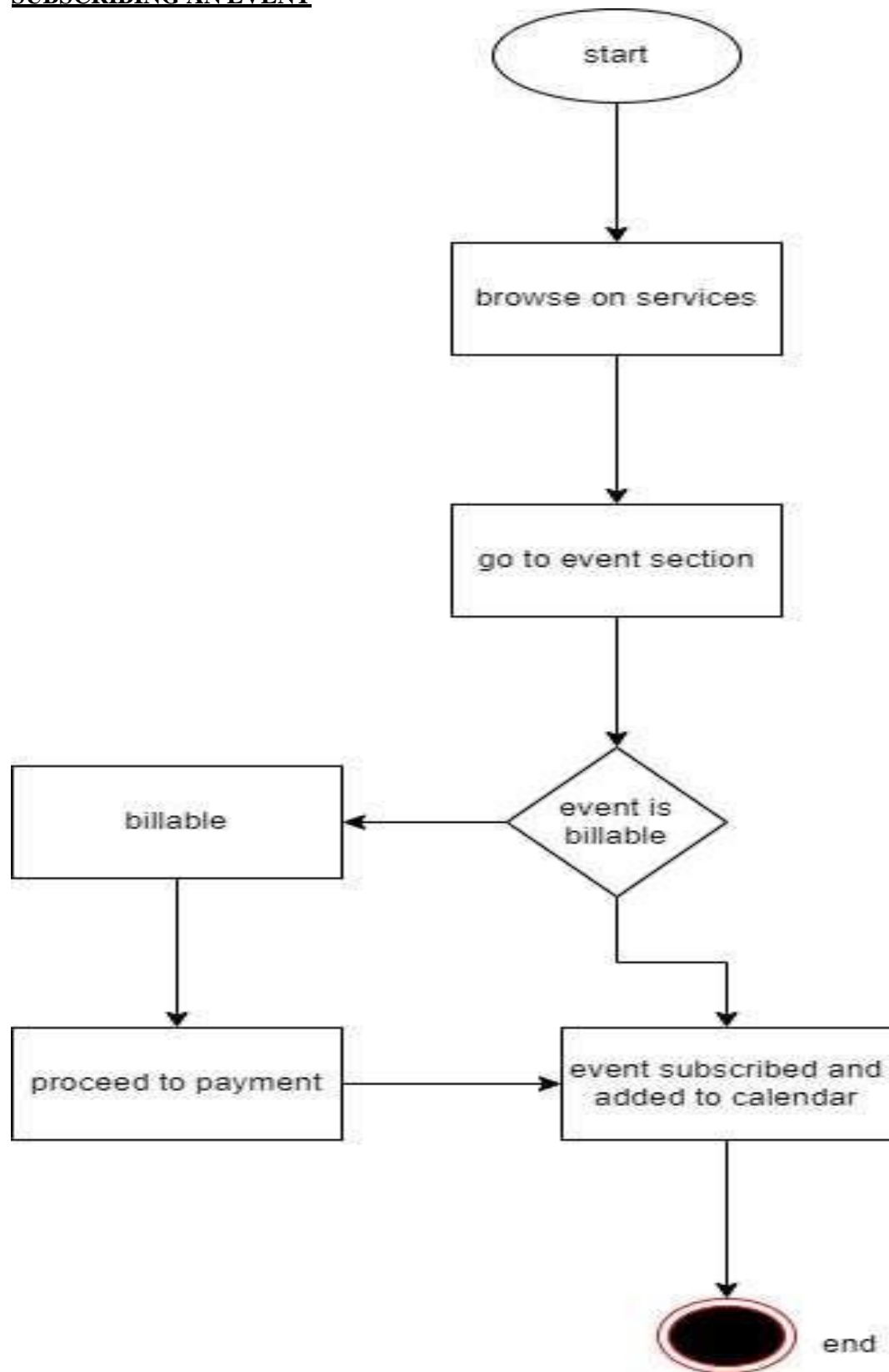
3. APPOINTMENT BOOKING



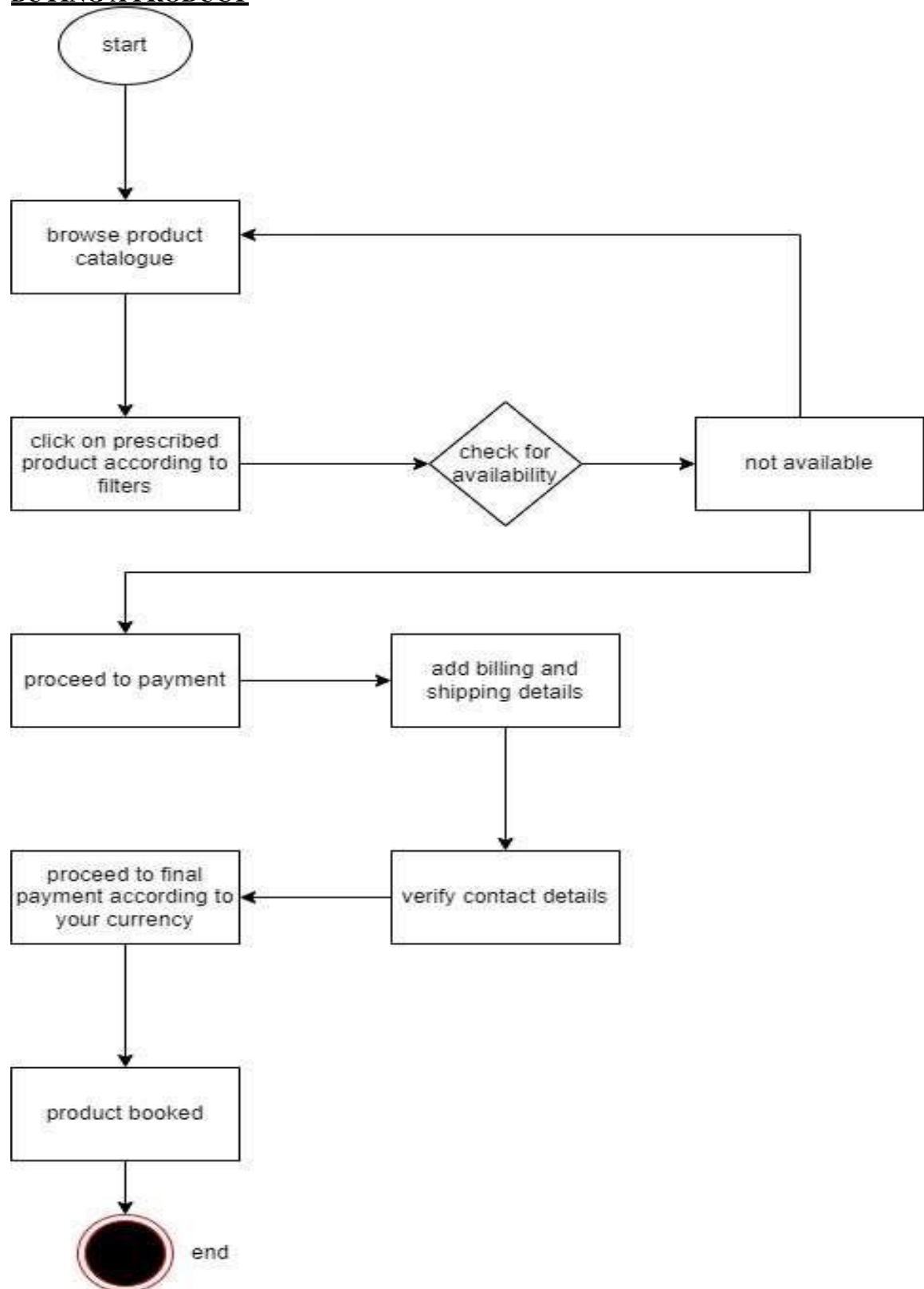
4. REMINDER SETTING



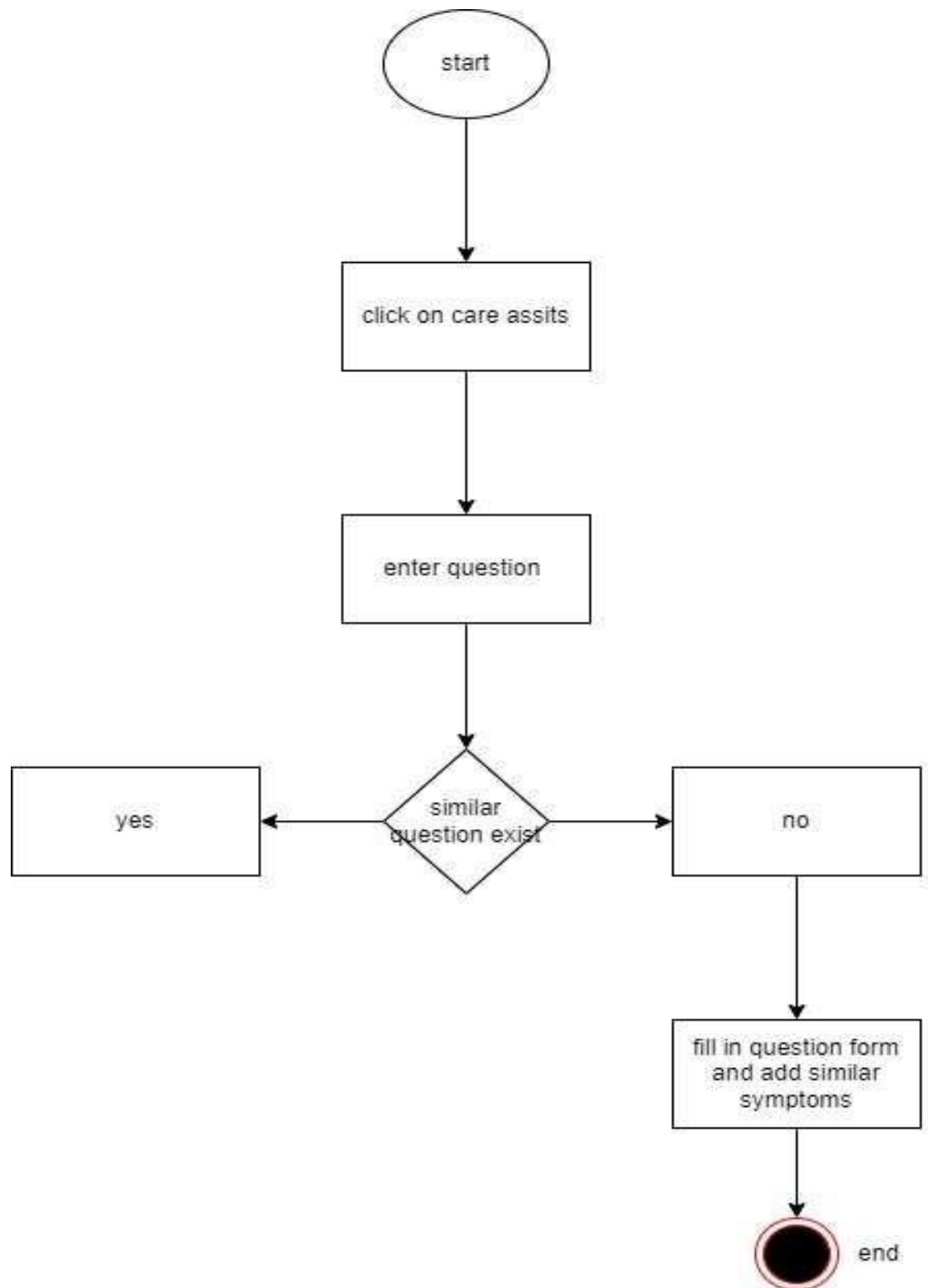
5. SUBSCRIBING AN EVENT



6. BUYING A PRODUCT

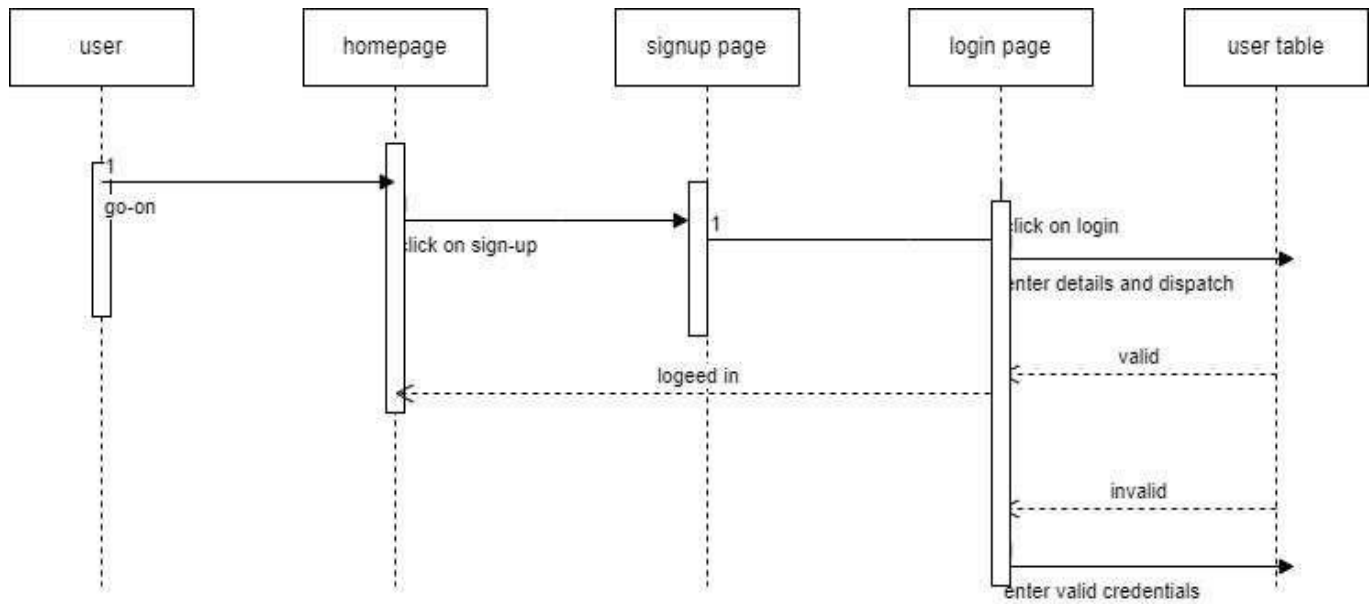


7. POSTING QUERY

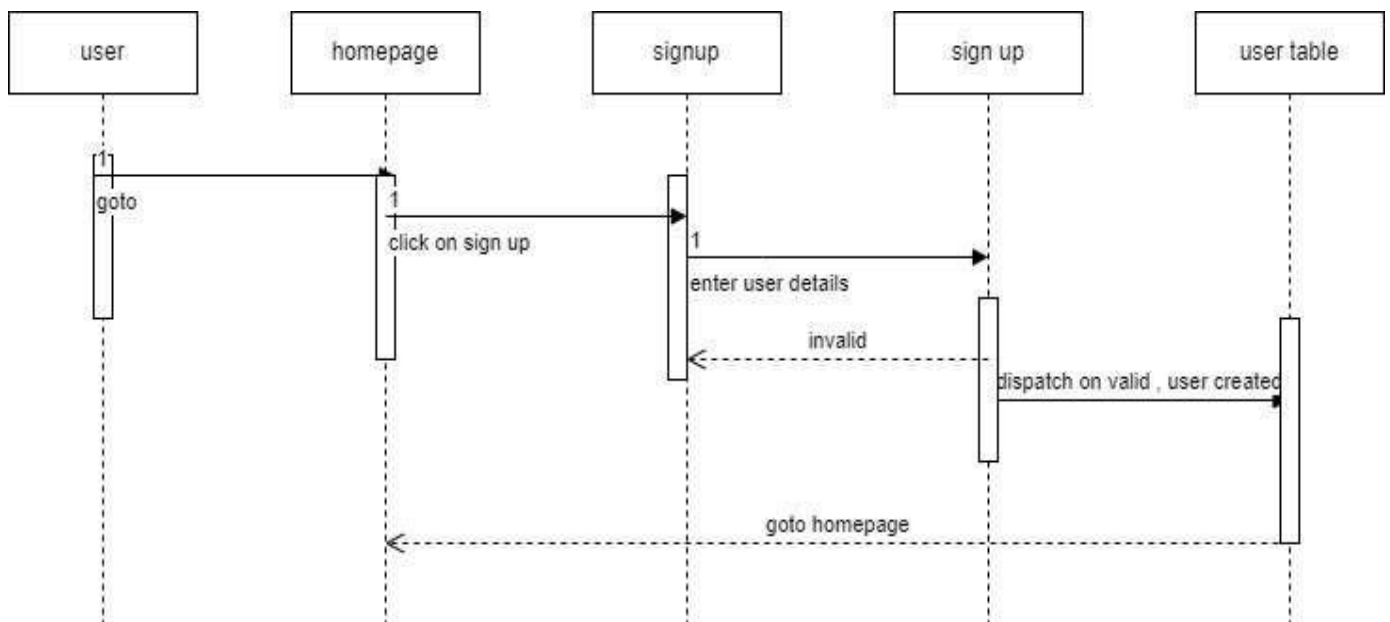


SEQUENCE DIAGRAM:

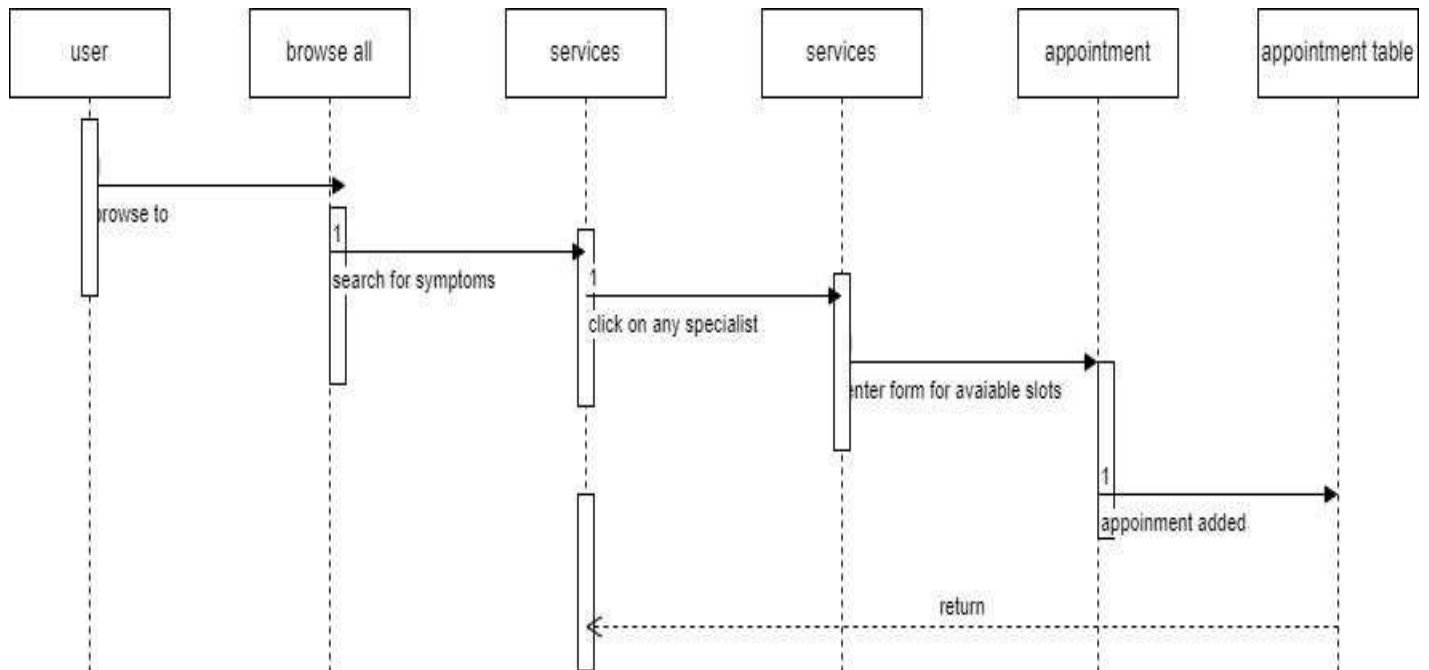
1. LOGIN



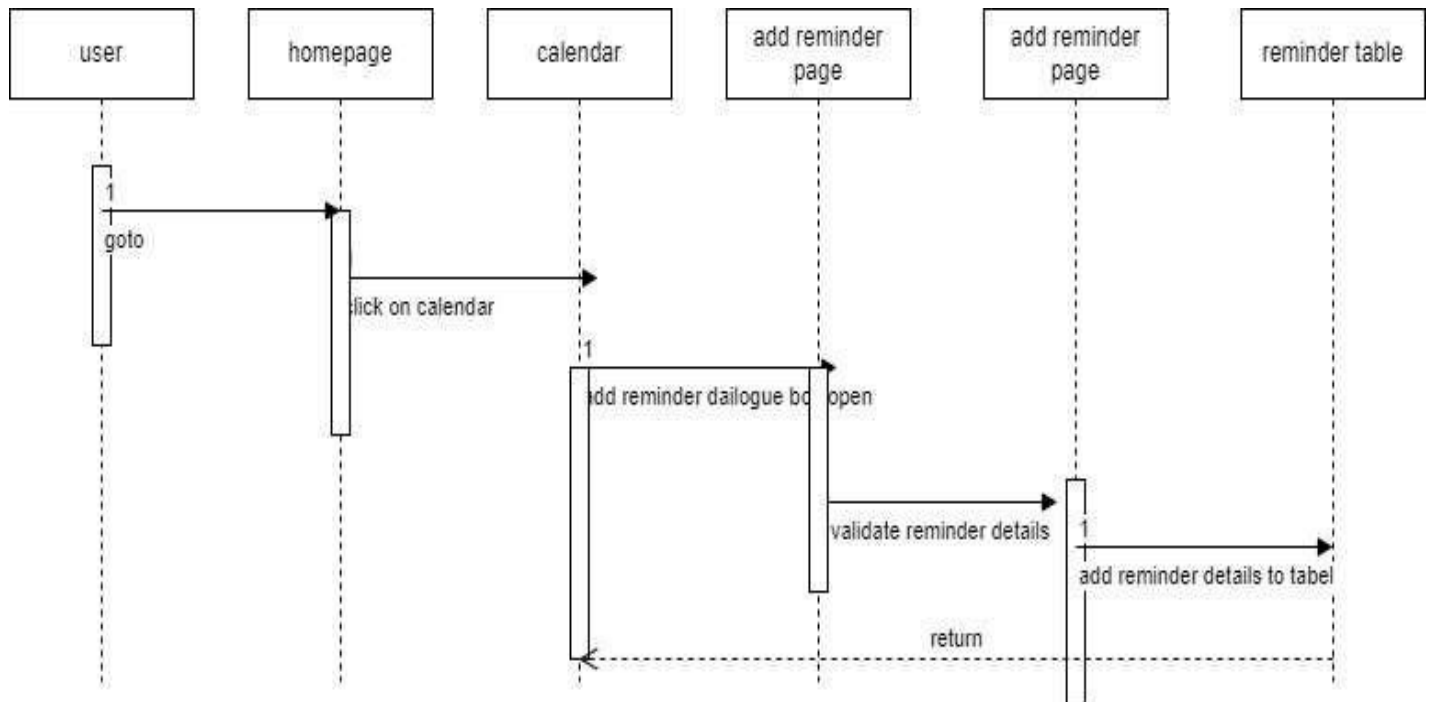
2. SIGNUP



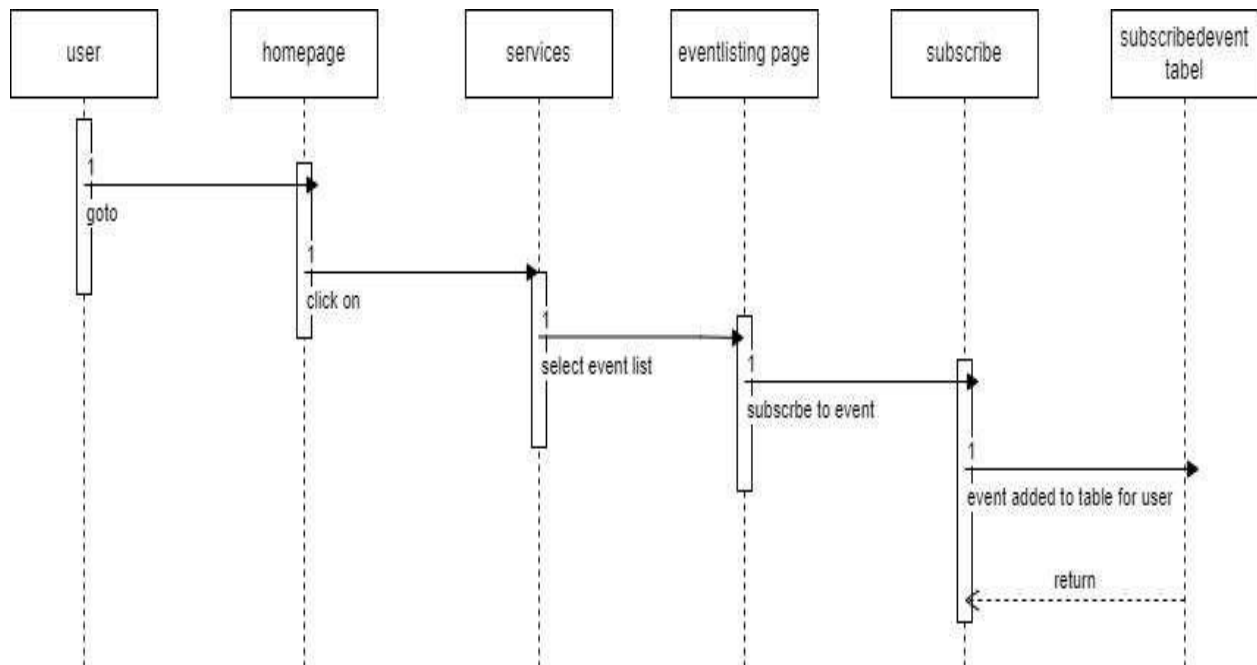
3. APPOINMENT BOOKING



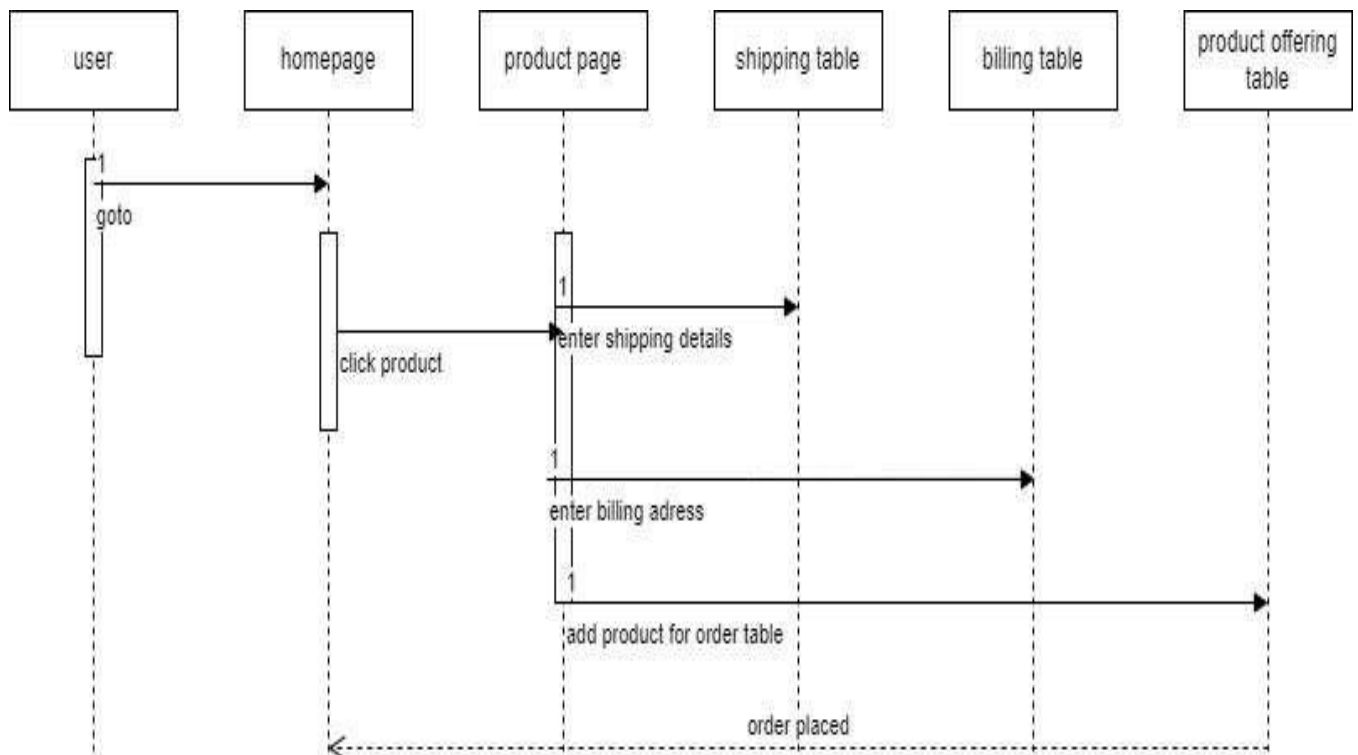
4. REMINDER SETTING



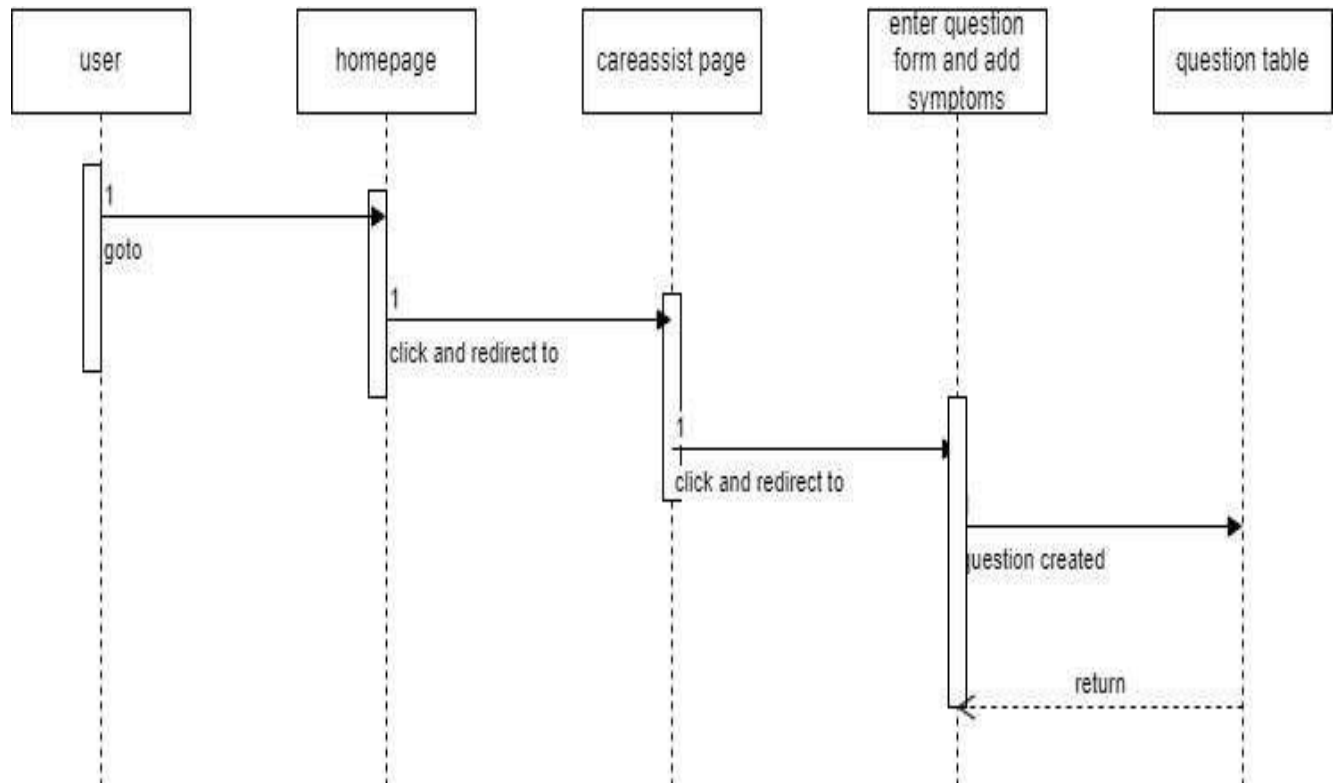
5. SUBSCRIBING TO AN EVENT



6. PURCHASNG PRODUCT



7. POSTING QUERY



TESTING

User Interface Testing: -

- Content should be correct without any spelling or grammatical errors.
- All the error messages should be correct without any spelling or grammatical errors and the error message should match with the field label.
- Enough space should be provided between field labels, columns, rows, and error messages.
- All the buttons should be in a standard format and size.
- The date format should be aligned in all the screens
- Website should behave responsively for all the screen sizes.

Functional Testing: -

- All the mandatory fields should be validated (they should get highlighted if not filled by

- the user on submitting the request).
- Check for the auto populated entries.
 - Test all the editable and non-editable fields.
 - The numeric fields should not accept the alphabets.
 - Test the maximum length of the numeric fields.
 - The system should not display the error message for optional fields.
 - Confirmation message should be displayed for any kind of update and delete operation.
 - The modules and the processes should be correctly assigned to the user as per the role.
 - Only the assigned modules and processes should be visible to the user on the menu screen.
 - Proper error messages should get displayed in case of any error in submitting/approving/rejecting the requests.
 - User can not checkout to payment gateway before adding any product to cart.
 - User should be able to choose mode of service
 - Filter should be correctly applied while listing products and services.
 - All the subscribed events should be displayed on calendar.
 - Webpage should not crash even the response from backend is null.

Database Testing: -

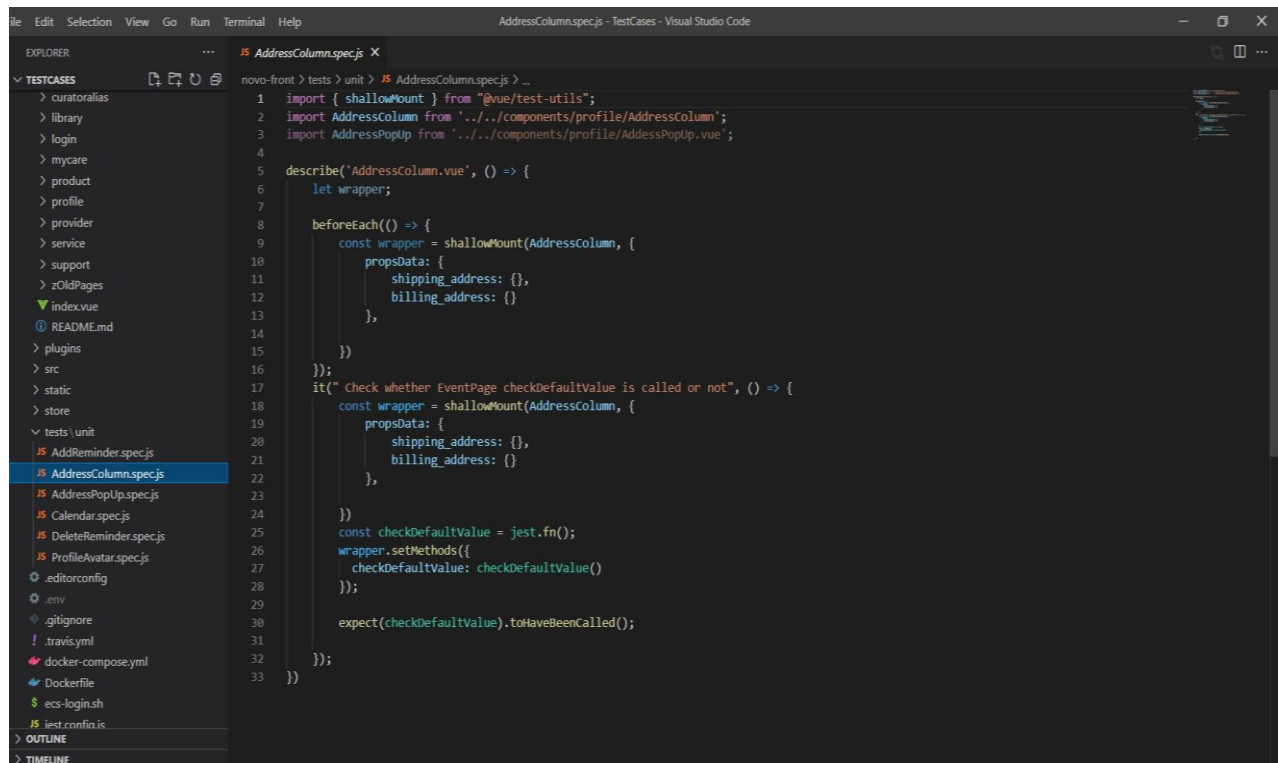
- Verify whether the column allows a null or not.
- Test the table row with duplicate/identical sample input data values.
- Verify the data gets properly saved into table(s) of the database after each page submission.
- Verify the data if the DML (Update, delete and insert) operations are performed.
- Check the length of every field
- The data displayed on the frontend should match with that in the backend.
- Verify the data validity by inserting the invalid data in the database.
- Verify the execution impact on other tables.
- Verify that the modules are correctly assigned to the respective processes.
- The roles should be correctly defined in the table.
- All the product should contain information about selling price and taxable amount.

Security Testing: -

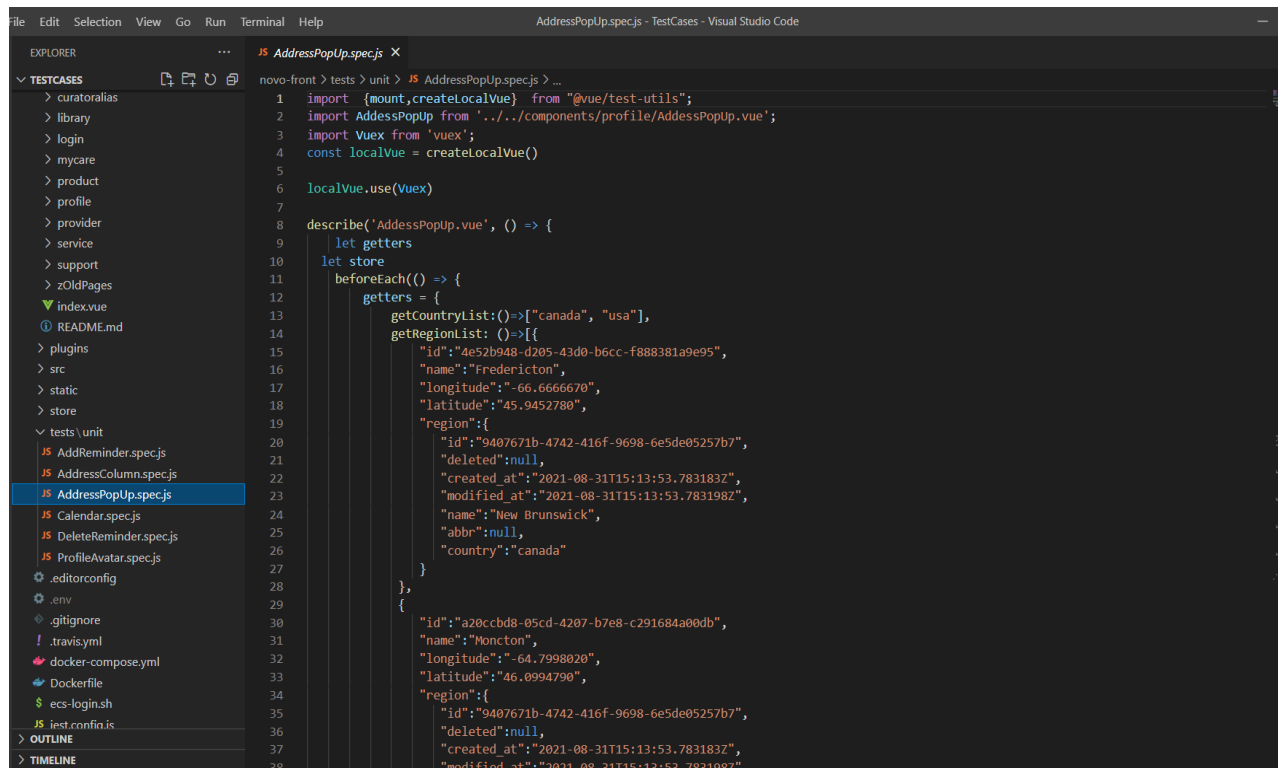
- It should be verified that only the authenticated users are allowed to login.
- Verify the users' roles and their rights.

Unit Testing: -

- To check that the components are rendering data correctly, we have done unit testing of few modules.



```
1 import { shallowMount } from '@vue/test-utils';
2 import AddressColumn from '../../components/profile/AddressColumn';
3 import AddressPopUp from '../../components/profile/AddressPopUp.vue';
4
5 describe('AddressColumn.vue', () => {
6   let wrapper;
7
8   beforeEach(() => {
9     const wrapper = shallowMount(AddressColumn, {
10       propsData: {
11         shipping_address: {},
12         billing_address: {}
13       },
14     });
15   });
16
17   it('Check whether EventPage checkDefaultValue is called or not', () => {
18     const wrapper = shallowMount(AddressColumn, {
19       propsData: {
20         shipping_address: {},
21         billing_address: {}
22       },
23     });
24
25     const checkDefaultValue = jest.fn();
26     wrapper.setMethods({
27       checkDefaultValue: checkDefaultValue()
28     });
29
30     expect(checkDefaultValue).toHaveBeenCalled();
31   });
32 });
```



```
1 import { mount, createLocalVue } from '@vue/test-utils';
2 import AddressPopUp from '../../components/profile/AddressPopUp.vue';
3 import Vuex from 'vuex';
4 const localVue = createLocalVue()
5
6 localVue.use(Vuex)
7
8 describe('AddressPopUp.vue', () => {
9   let getters
10   let store
11   beforeEach(() => {
12     getters = {
13       getCountryList: () => ["canada", "usa"],
14       getRegionList: () => [{
15         "id": "4e52b948-d205-43d0-b6cc-f888381a9e95",
16         "name": "Fredericton",
17         "longitude": "-66.6666670",
18         "latitude": "45.9452780",
19         "region": {
20           "id": "9407671b-4742-416f-9698-6e5de05257b7",
21           "deleted": null,
22           "created_at": "2021-08-31T15:13:53.783183Z",
23           "modified_at": "2021-08-31T15:13:53.783198Z",
24           "name": "New Brunswick",
25           "abbr": null,
26           "country": "canada"
27         }
28       }],
29     },
30     {
31       "id": "a20ccbd8-05cd-4207-b7e8-c291684a00db",
32       "name": "Moncton",
33       "longitude": "-64.7998020",
34       "latitude": "46.0994790",
35       "region": {
36         "id": "9407671b-4742-416f-9698-6e5de05257b7",
37         "deleted": null,
38         "created_at": "2021-08-31T15:13:53.783183Z",
39         "modified_at": "2021-08-31T15:13:53.783198Z"
```

```
File Edit Selection View Go Run Terminal Help AddressPopUp.spec.js - TestCases - Visual Studio Code

EXPLORER
TESTCASES
  curatorialias
  library
  login
  mycare
  product
  profile
  provider
  service
  support
  zOldPages
  index.vue
  README.md
  plugins
  src
  static
  store
  tests \ unit
    JS AddReminder.spec.js
    JS AddressColumn.spec.js
    JS AddressPopUp.spec.js
    JS Calendar.spec.js
    JS DeleteReminder.spec.js
    JS ProfileAvatar.spec.js
  .editorconfig
  .env
  .gitignore
  .travis.yml
  docker-compose.yml
  Dockerfile
  ecs-login.sh
  jest.config.js
  OUTLINE
  TIMELINE

novofront > tests > unit > JS AddressPopUp.spec.js > ...
283 wrapper.find('v-select').trigger('input');
284 expect(getRegions).toHaveBeenCalled();
285
286
287
288 test('check getRegions Method', () => {
289
290   const wrapper = mount(AddressPopUp, {propsData:{
291     billing:'true',
292     title:'Add Billing Address',
293     icon_name:'mdi-plus',
294     address_type:'billing',
295     is_bill_edit:'false'
296   }, store, localVue })
297   const getRegions = jest.fn();
298   wrapper.setMethods({
299     getRegions: getRegions()
300   });
301   wrapper.find('v-select').trigger('input');
302   expect(getRegions).toHaveBeenCalled();
303
304 })
305
306 test('check if address_type prop passed to component', () => {
307   const address_type='billing'
308
309   const wrapper = mount(AddressPopUp, {propsData:{
310     billing:'true',
311     title:'Add Billing Address',
312     icon_name:'mdi-plus',
313     address_type:'billing',
314     is_bill_edit:'false'
315   }, store, localVue })
316
317   expect(wrapper.props('address_type')).toBe(address_type);
318
319 })
320
```

```
File Edit Selection View Go Run Terminal Help Calendar.spec.js - TestCases - Visual Studio Code

EXPLORER
TESTCASES
  curatorialias
  library
  login
  mycare
  product
  profile
  provider
  service
  support
  zOldPages
  index.vue
  README.md
  plugins
  src
  static
  store
  tests \ unit
    JS AddReminder.spec.js
    JS AddressColumn.spec.js
    JS AddressPopUp.spec.js
    JS Calendar.spec.js
    JS DeleteReminder.spec.js
    JS ProfileAvatar.spec.js
  .editorconfig
  .env
  .gitignore
  .travis.yml
  docker-compose.yml
  Dockerfile
  ecs-login.sh
  jest.config.js
  OUTLINE
  TIMELINE

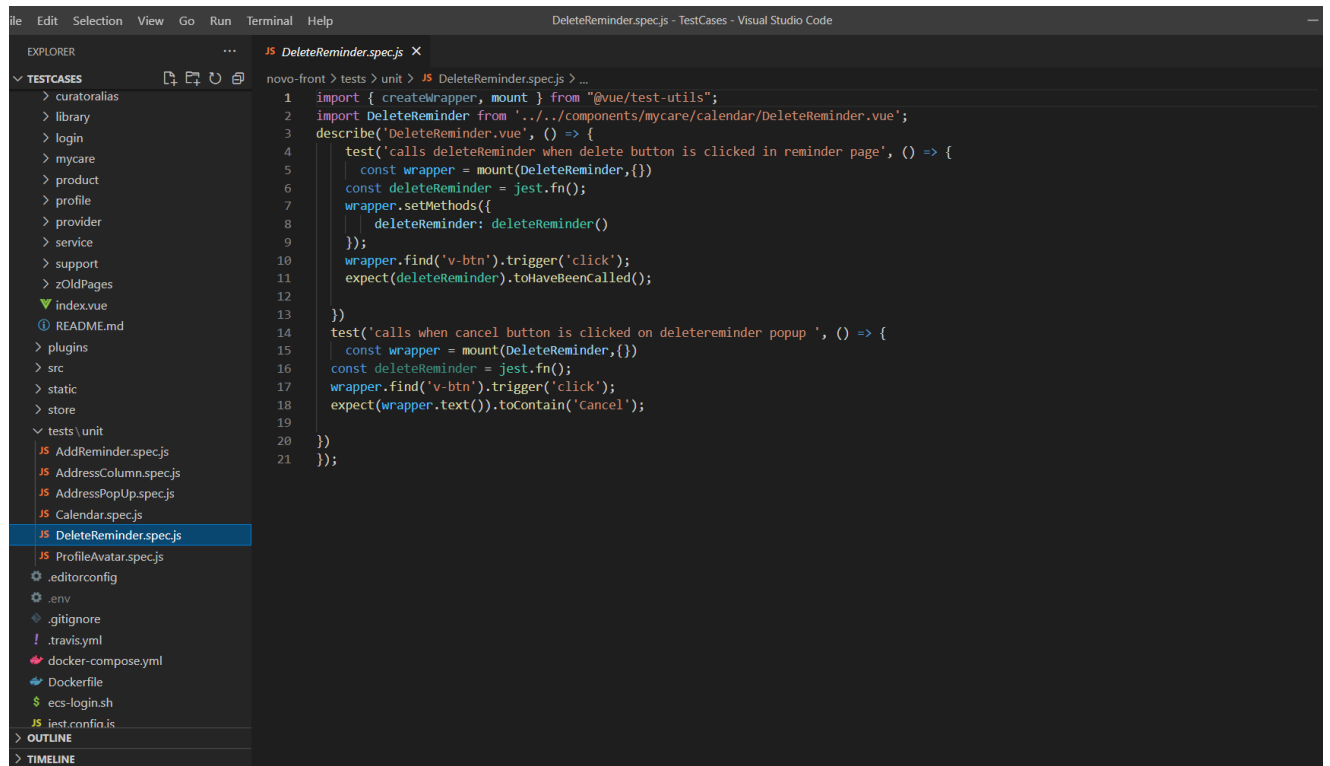
novofront > tests > unit > JS Calendar.spec.js > ...
1 import { mount } from '@vue/test-utils';
2 import Calendar from '../components/mycare/calendar/Calendar.vue'
3 describe('Calendar.vue',()=>{
4
5   test('calls when doubleclick on date in calendar component', () => {
6     const wrapper = mount(Calendar,{
7       propsData: {
8         events:[
9           {
10             "id": "ac65525c-579f-4a45-858c-bd5a4a0570b4",
11             "deleted": null,
12             "created_at": "2021-09-23T06:03:47.716423Z",
13             "modified_at": "2021-09-23T06:03:47.716452Z",
14             "name": "Reminder1",
15             "datetime": "2021-09-24T23:34:00Z",
16             "note": "this is reminder"
17           },
18           {
19             "id": "0053f53c-53c7-439d-89d0-c463acce82bb",
20             "deleted": null,
21             "created_at": "2021-08-26T06:33:37.708694Z",
22             "modified_at": "2021-09-08T09:30:58.252815Z",
23             "name": "Cancer awareness",
24             "description": "In India, the incidence of cancer is increasing rapidly; therefore it is important to step up cancer",
25             "event_type": "temporary",
26             "event_url": null,
27             "datetime": "2021-10-22T15:20:00Z",
28             "slug": "cancer-awareness",
29             "provider": {
30               "id": "3c49746e-1379-4652-a8e2-708f3980832a",
31               "deleted": null,
32               "created_at": "2021-07-28T02:04:41.804815Z",
33               "modified_at": "2021-08-31T12:10:01.859803Z",
34               "credentials": "nan",
35               "first_name": "Eva",
36               "last_name": "Tang",
37               "display_name": "Eva Tang,RPT",
38               "provider_type": "Clinician",
39               "details": "",
40               "education": ""
41             }
42           }
43         ]
44       },
45       store,
46       localVue
47     })
48     wrapper.find('v-select').trigger('input');
49     expect(getRegions).toHaveBeenCalled();
50   })
51 })
52
```

```
1 import { mount } from '@vue/test-utils';
2 import Calendar from '../components/mycare/calendar/Calendar.vue'
3 describe('Calendar.vue', () => {
4   test('calls when doubleclick on date in calendar component', () => {
5     const wrapper = mount(Calendar, {
6       propsData: {
7         events: [
8           {
9             "id": "ac65525c-579f-4a45-858c-bd5a4a0570b4",
10            "deleted": null,
11            "created_at": "2021-09-23T06:03:47.716423Z",
12            "modified_at": "2021-09-23T06:03:47.716452Z",
13            "name": "Reminder1",
14            "datetime": "2021-09-24T23:34:00Z",
15            "note": "this is reminder"
16          },
17          {
18            "id": "0053f53c-53c7-439d-89d0-c463acce82bb",
19            "deleted": null,
20            "created_at": "2021-08-26T06:33:37.708694Z",
21            "modified_at": "2021-09-08T09:30:58.252815Z",
22            "name": "cancer awareness",
23            "description": "In India, the incidence of cancer is increasing rapidly; therefore it is important to step up cancer",
24            "event_type": "temporary",
25            "event_url": null,
26            "datetime": "2021-10-22T15:20:00Z",
27            "slug": "cancer-awareness",
28            "provider": {
29              "id": "3c49746e-1379-4652-a8e2-708f3980832a",
30              "deleted": null,
31              "created_at": "2021-07-28T02:04:41.804815Z",
32              "modified_at": "2021-08-31T12:10:01.859803Z",
33              "credentials": "nan",
34              "first_name": "Eva",
35              "last_name": "Tang",
36              "display_name": "Eva Tang,RPT",
37              "provider_type": "Clinician",
38              "details": "",
39              "education": ""
40            }
41          }
42        ]
43      }
44    });
45  });
46 }
```

```
1 import { mount } from '@vue/test-utils';
2 import Calendar from '../components/mycare/calendar/Calendar.vue'
3 describe('Calendar.vue', () => {
4   test('calls when doubleclick on date in calendar component', () => {
5     const wrapper = mount(Calendar, {
6       propsData: {
7         events: [
8           {
9             "id": "ac65525c-579f-4a45-858c-bd5a4a0570b4",
10            "deleted": null,
11            "created_at": "2021-09-23T06:03:47.716423Z",
12            "modified_at": "2021-09-23T06:03:47.716452Z",
13            "name": "Reminder1",
14            "datetime": "2021-09-24T23:34:00Z",
15            "note": "this is reminder"
16          },
17          {
18            "id": "0053f53c-53c7-439d-89d0-c463acce82bb",
19            "deleted": null,
20            "created_at": "2021-08-26T06:33:37.708694Z",
21            "modified_at": "2021-09-08T09:30:58.252815Z",
22            "name": "cancer awareness",
23            "description": "In India, the incidence of cancer is increasing rapidly; therefore it is important to step up cancer",
24            "event_type": "temporary",
25            "event_url": null,
26            "datetime": "2021-10-22T15:20:00Z",
27            "slug": "cancer-awareness",
28            "provider": {
29              "id": "3c49746e-1379-4652-a8e2-708f3980832a",
30              "deleted": null,
31              "created_at": "2021-07-28T02:04:41.804815Z",
32              "modified_at": "2021-08-31T12:10:01.859803Z",
33              "credentials": "nan",
34              "first_name": "Eva",
35              "last_name": "Tang",
36              "display_name": "Eva Tang,RPT",
37              "provider_type": "Clinician",
38              "details": "",
39              "education": ""
40            }
41          }
42        ]
43      }
44    });
45  });
46 }
```

```
1 import { mount } from '@vue/test-utils';
2 import Calendar from '../components/mycare/calendar/Calendar.vue'
3 describe('Calendar.vue', () => {
4   test('calls when doubleclick on date in calendar component', () => {
5     const wrapper = mount(Calendar, {
6       propsData: {
7         events: [
8           {
9             "id": "ac65525c-579f-4a45-858c-bd5a4a0570b4",
10            "deleted": null,
11            "created_at": "2021-09-23T06:03:47.716423Z",
12            "modified_at": "2021-09-23T06:03:47.716452Z",
13            "name": "Reminder1",
14            "datetime": "2021-09-24T23:34:00Z",
15            "note": "this is reminder"
16          },
17          {
18            "id": "0053f53c-53c7-439d-89d0-c463acce82bb",
19            "deleted": null,
20            "created_at": "2021-08-26T06:33:37.708694Z",
21            "modified_at": "2021-09-08T09:30:58.252815Z",
22            "name": "cancer awareness",
23            "description": "In India, the incidence of cancer is increasing rapidly; therefore it is important to step up cancer",
24            "event_type": "temporary",
25            "event_url": null,
26            "datetime": "2021-10-22T15:20:00Z",
27            "slug": "cancer-awareness",
28            "provider": {
29              "id": "3c49746e-1379-4652-a8e2-708f3980832a",
30              "deleted": null,
31              "created_at": "2021-07-28T02:04:41.804815Z",
32              "modified_at": "2021-08-31T12:10:01.859803Z",
33              "credentials": "nan",
34              "first_name": "Eva",
35              "last_name": "Tang",
36              "display_name": "Eva Tang,RPT",
37              "provider_type": "Clinician",
38              "details": "",
39            }
40          }
41        ]
42      }
43    });
44  });
45 });
```

```
107   const date = '2021-09-20'
108   const dblClick = jest.fn()
109   wrapper.setMethods({
110     dblClick: dblClick(date)
111   })
112   // console.log(wrapper.find('v-date-picker'))
113   // wrapper.find('v-date-picker').trigger('dblclick');
114   expect(dblClick).toHaveBeenCalled()
115 });
```



Product Release Acceptance Criteria: -

The acceptance criteria is the USER Validation for move to production. They execute the tests (UAT) and after the result of testing they will provide go/no go.

USER INTERFACES

Sign up Screen:

[← Home](#)



Register for Novo Connect

Email Address

First Name

Last Name

Password

at least 8 characters

Confirm Password



Sign In screen:

[← Home](#)



Sign in to Novo Connect

Email Address

Password

FORGOT PASSWORD

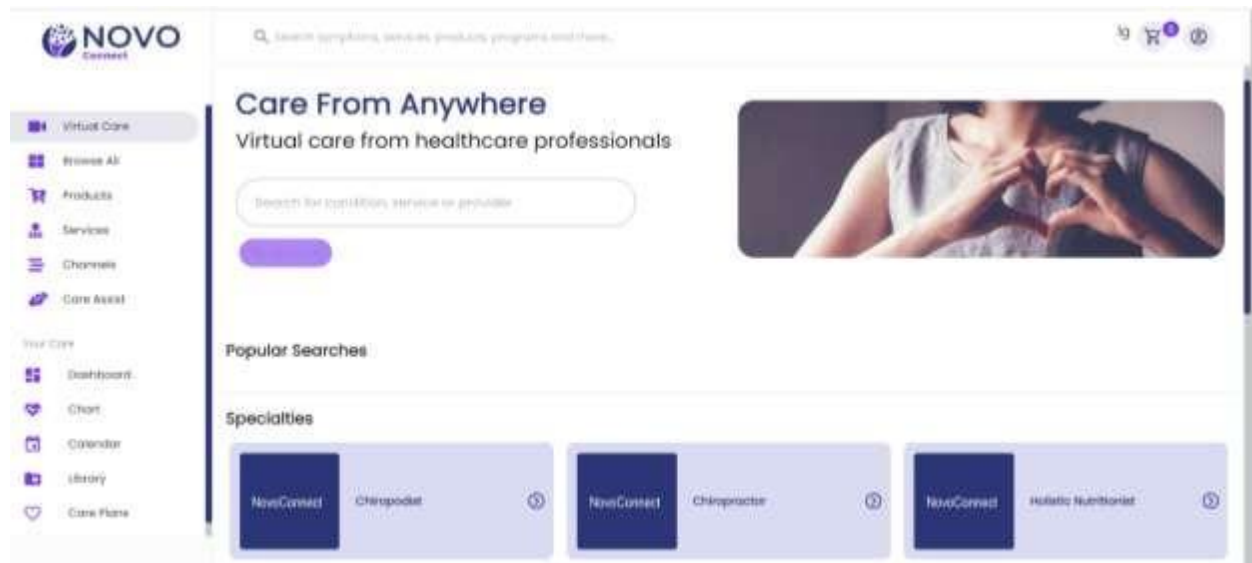
☐ Remember me

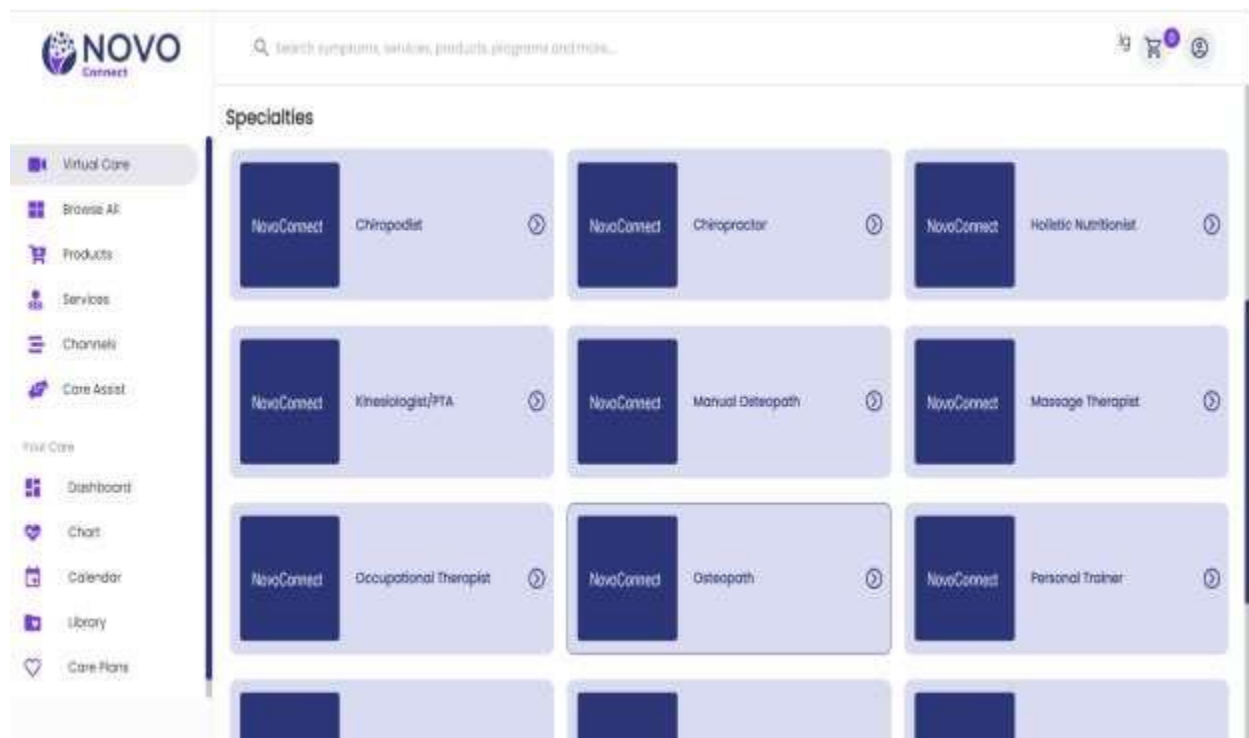
SIGN IN

NEW USER? CREATE AN ACCOUNT

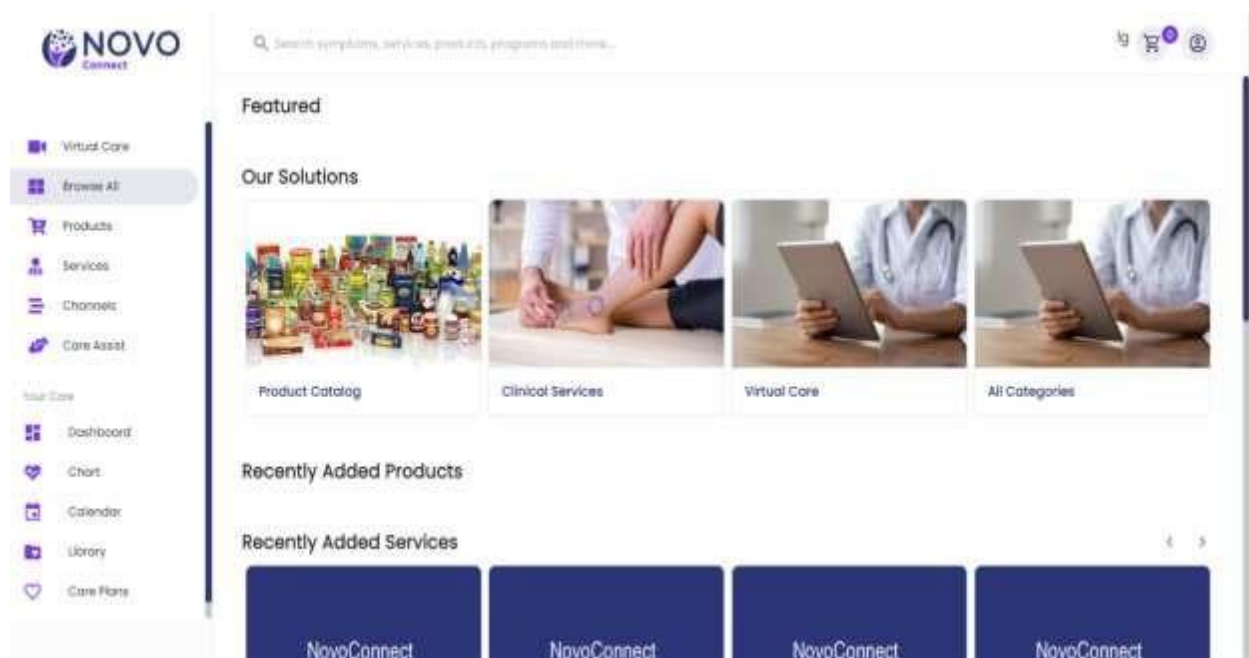


Virtual Care :

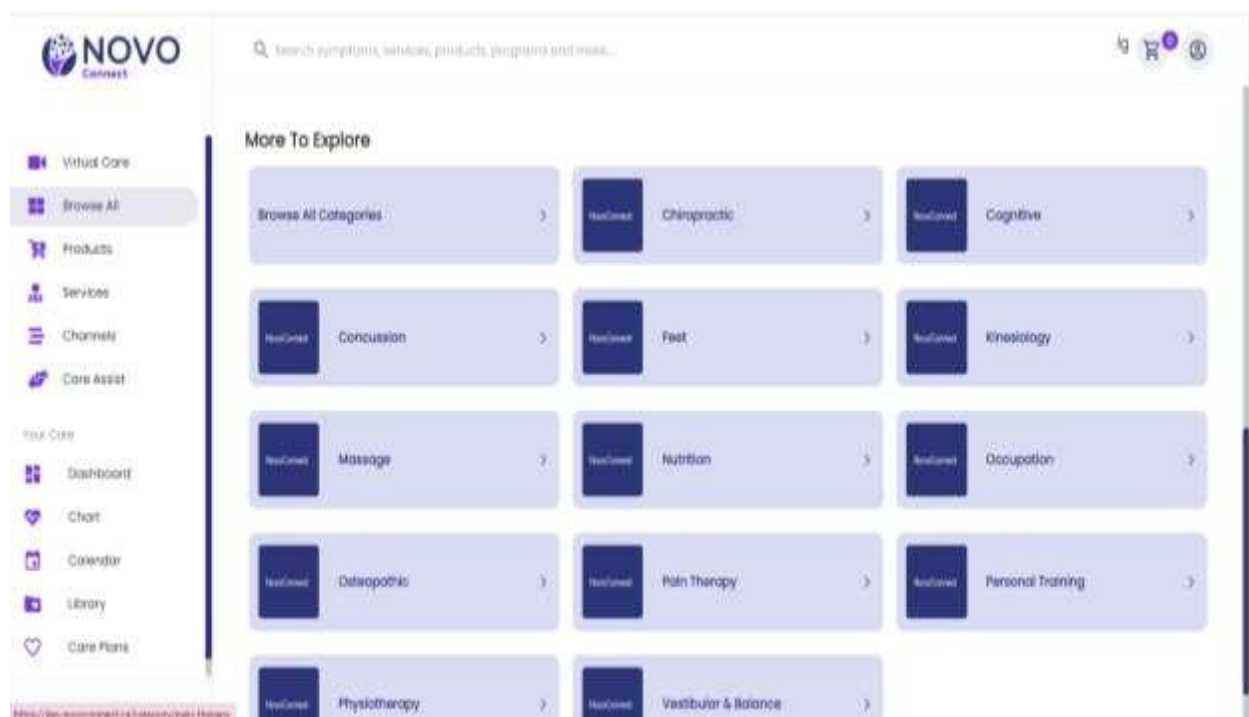
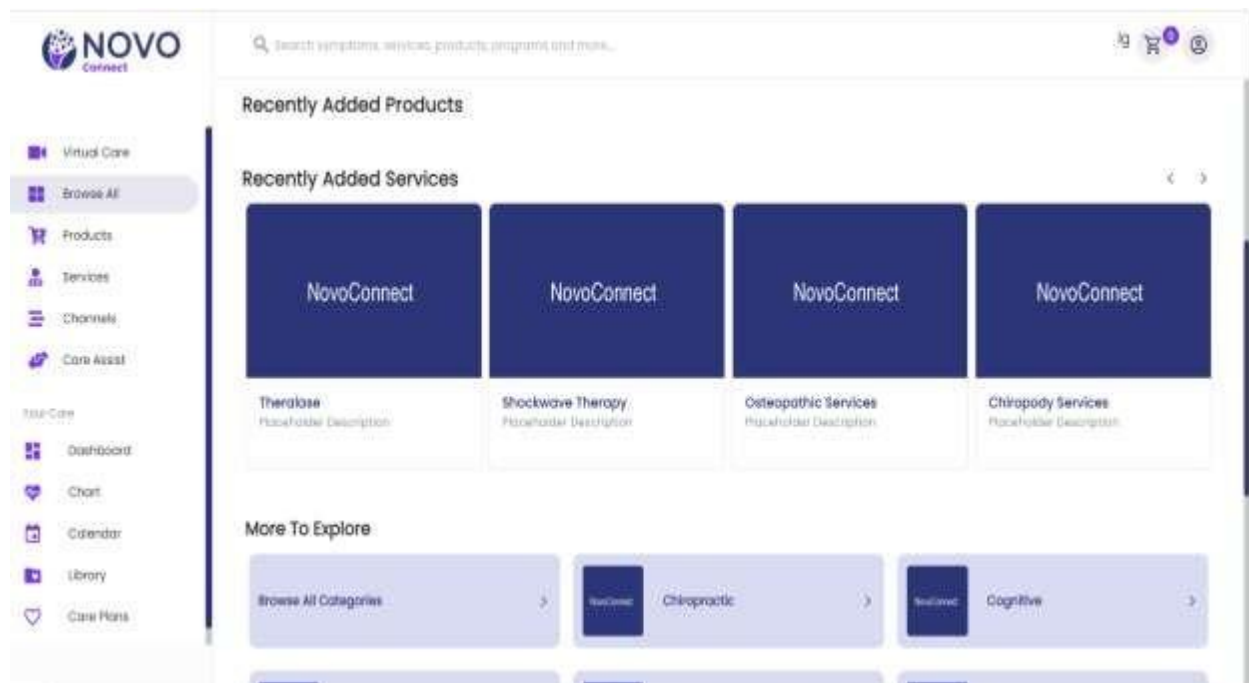




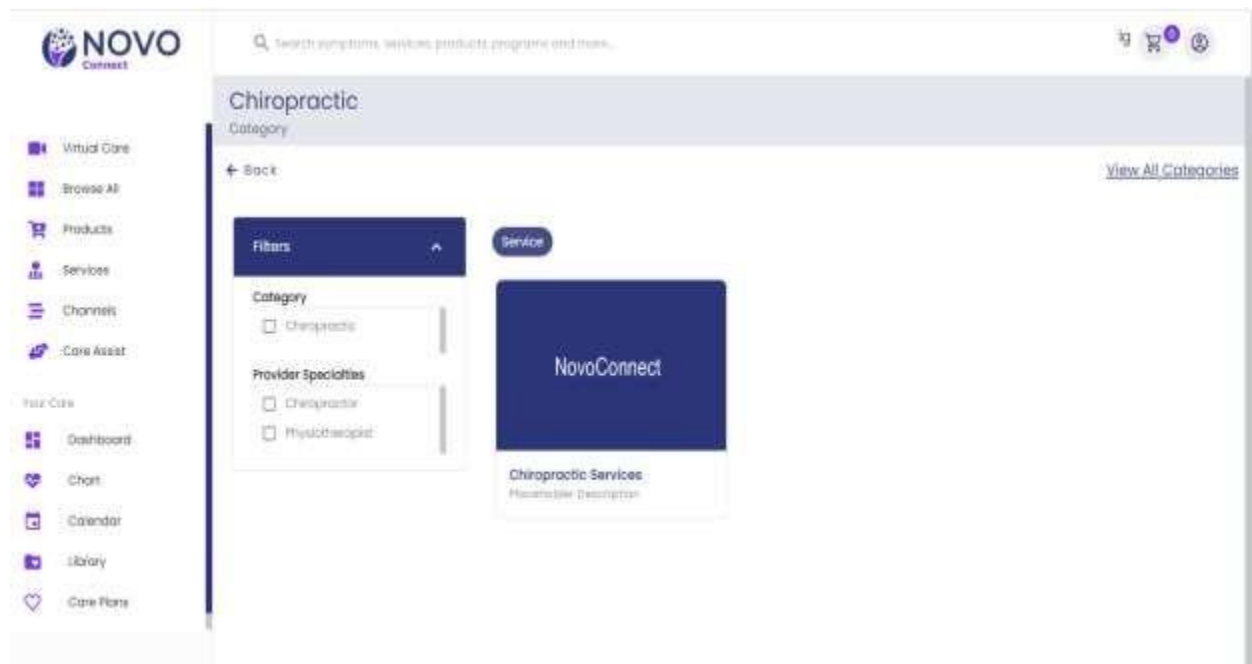
Browse All :



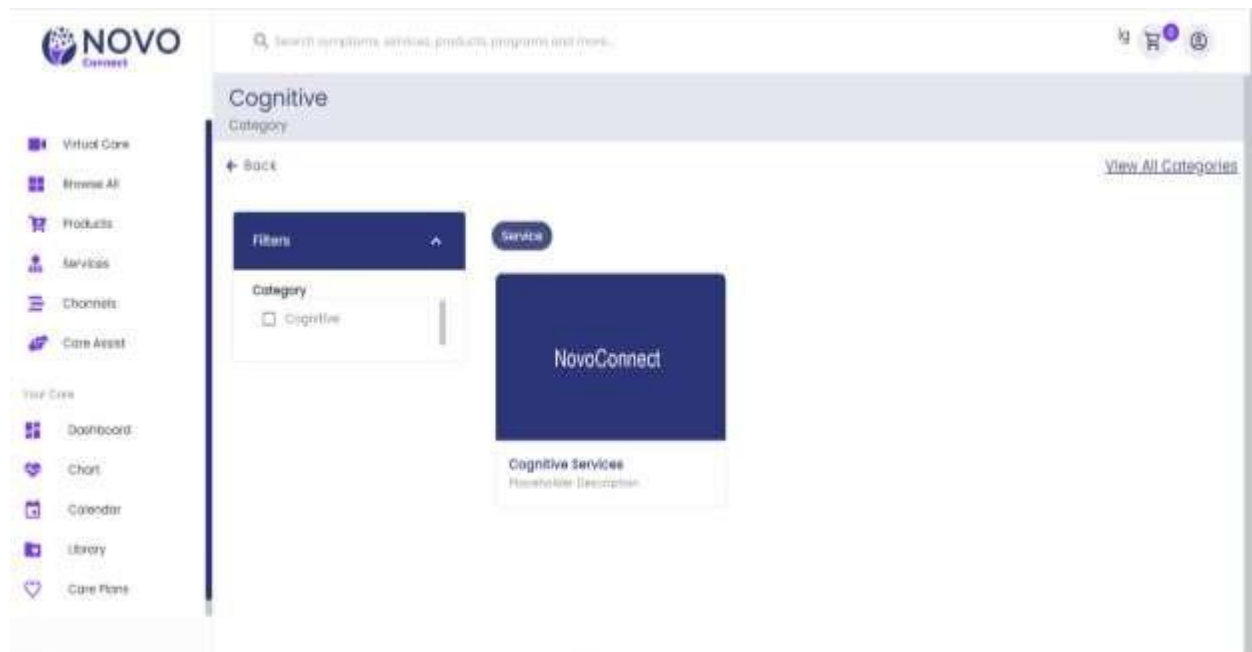
Recently Added Services:



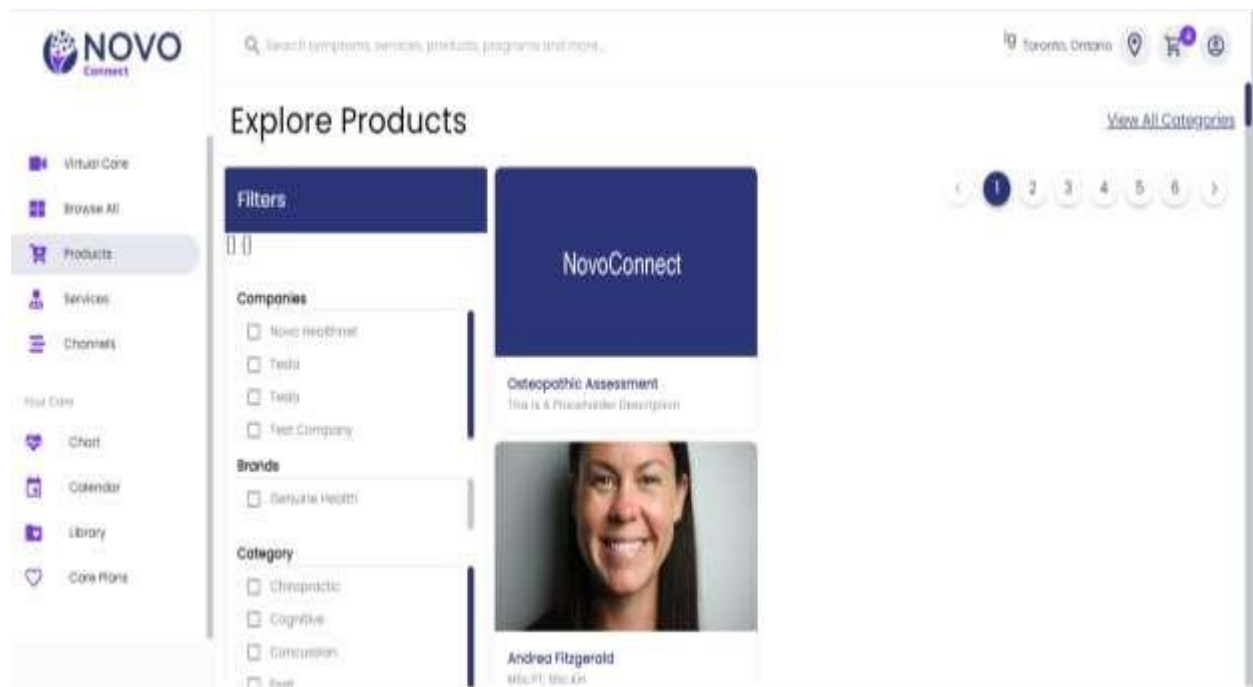
Chiropractic Category:



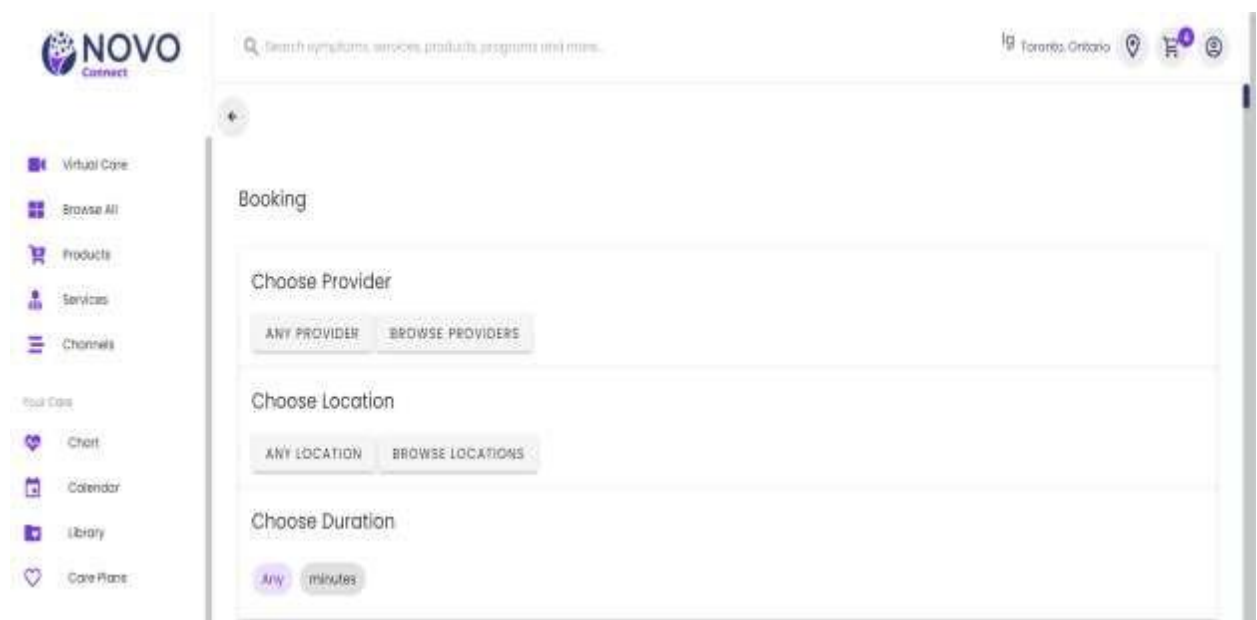
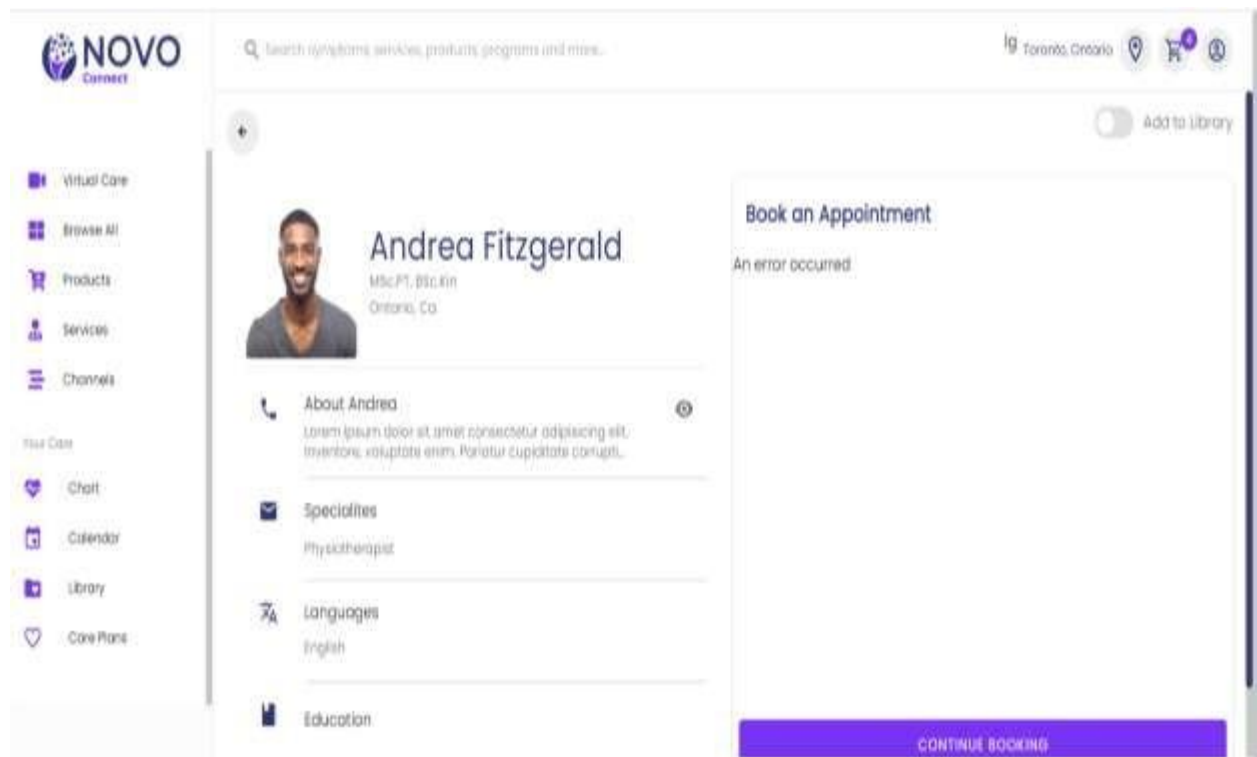
Cognitive Category:



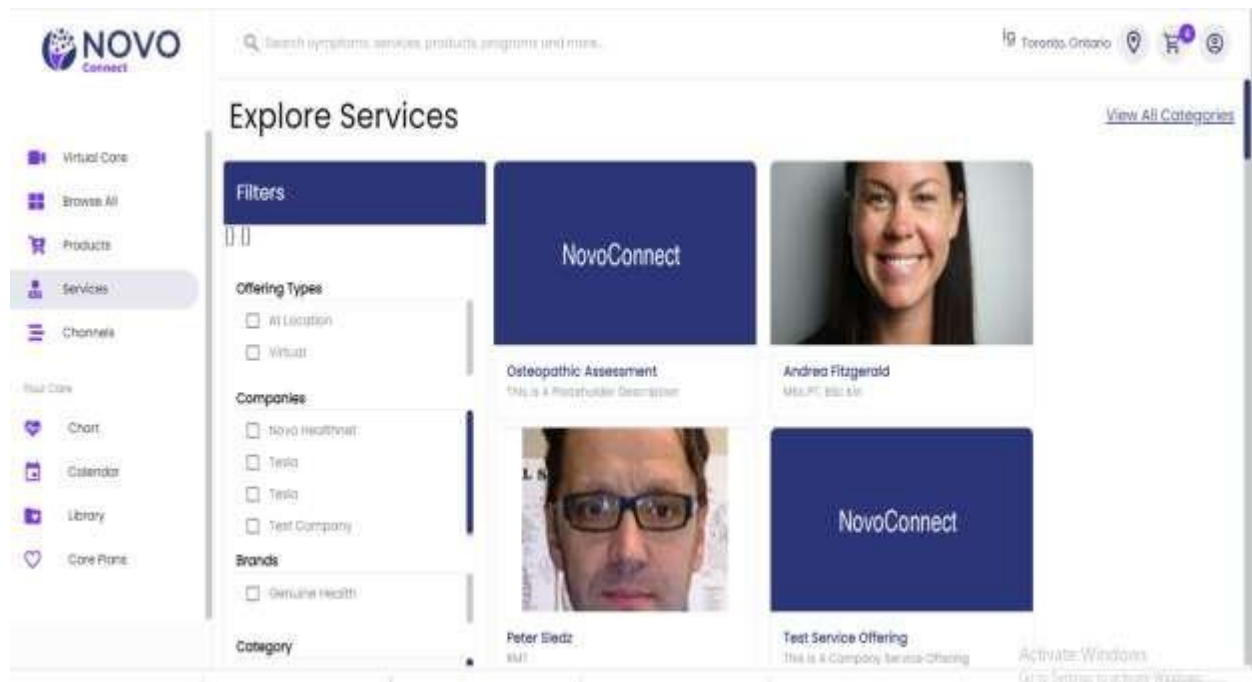
Products Page:

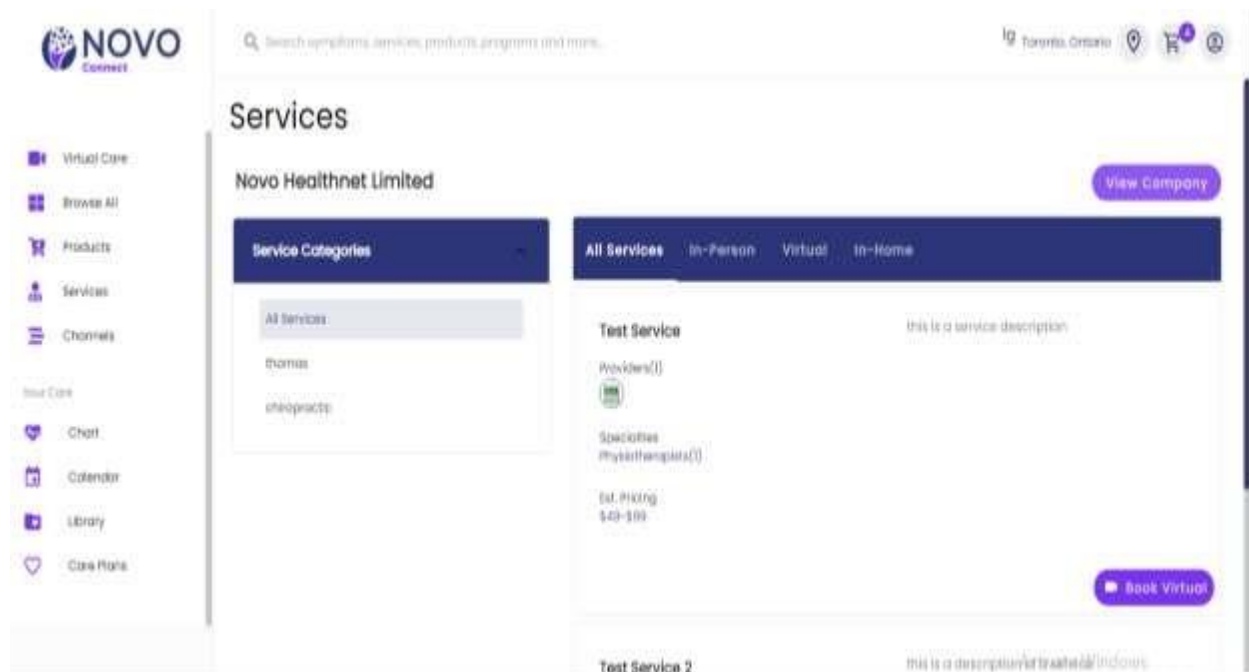


Book an Appointment Page:

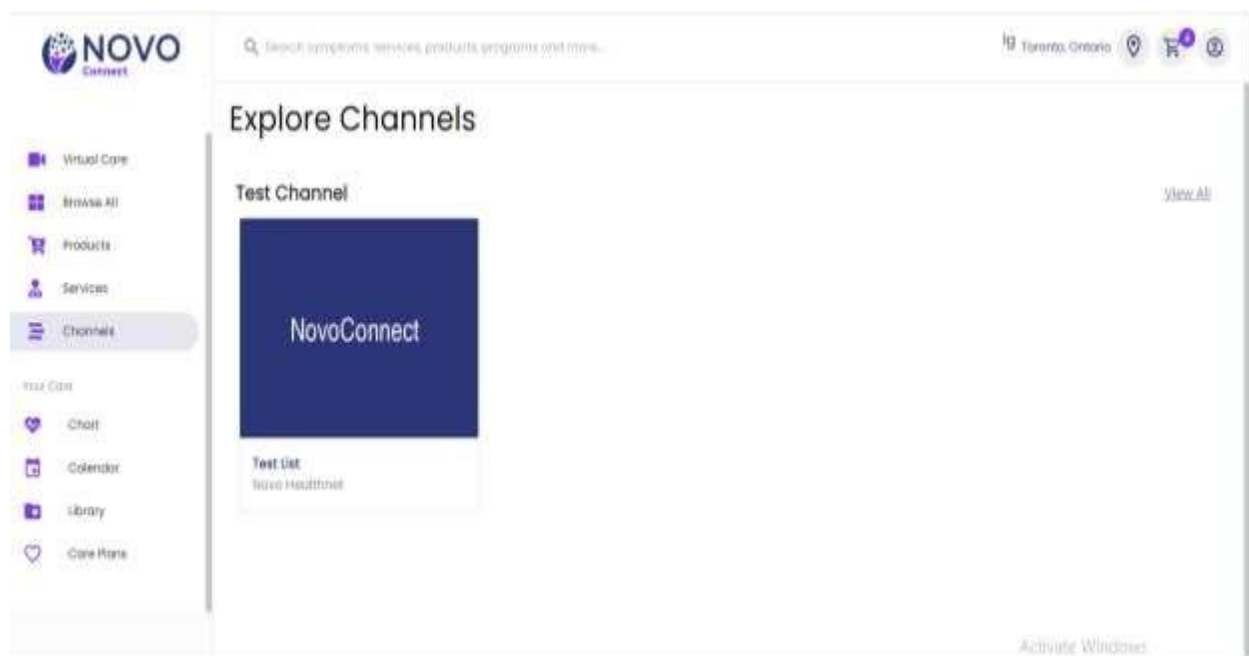


Services Page:

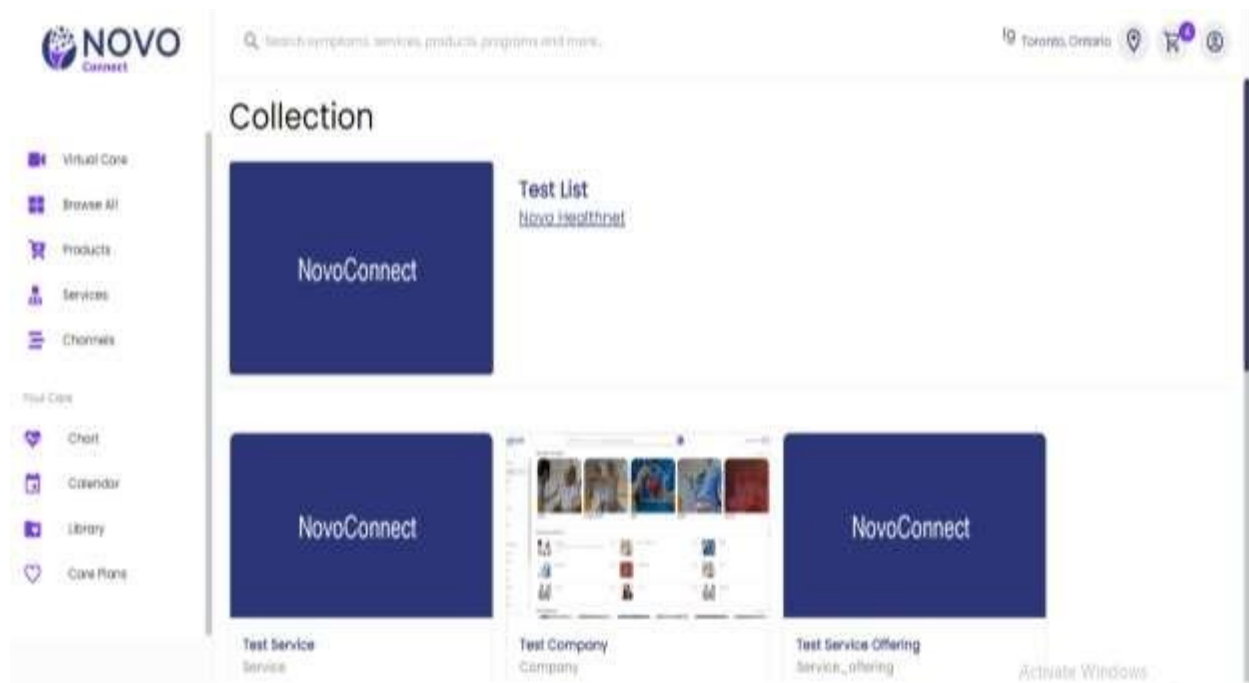




Channels Page:



Collections:



Company Page:



Chart Page:

The screenshot shows the 'My Chart' page in the NOVO Connect application. The left sidebar contains navigation options: Virtual Care, Browse All, Products, Services, Channels, and Your Care (with sub-options: Chart, Calendar, Library, and Care Plans). The main content area is titled 'My Chart' and features a search bar at the top. Below the search bar, there are four tabs: Demographics, General, Conditions, and Habits. The Demographics tab is active, displaying a list of patient information items, each with a purple icon and a right arrow for editing:

- Date Of Birth: Feb 11, 1998
- Gender: Female
- Gender Identity: Female
- Race: Hispanic, or, latino
- Occupation: Manager/ Dunder MIF.

The General tab displays:

- Height: 75 inches
- Weight: 236 lbs

The Conditions tab displays:

- ICD10: E11.9 Diabetes Mellitus
- ICD10: P70.1 Infant Of Diabetic M.

The Habits tab displays:

- Diet: Vegan
- Exercise: Daily
- Smoker: No
- Alcohol: No

The bottom right corner of the page shows the date '2021-12-23 12:00:00'.

Calendar Page:

The screenshot shows the 'My Calendar' page in the NOVO Connect application. The left sidebar is identical to the 'My Chart' page. The main content area is titled 'My Calendar' and displays 'No upcoming appointments'. On the right side, there is a calendar for December 2021. The current date, 'Thu, Dec 23', is highlighted in a dark blue box. Below the calendar, there is a 'Previous' section with three buttons labeled 'Dec 2nd', 'Dec 23rd', and 'Dec 23rd'. A 'View All' link is located at the bottom right of the calendar area. The bottom right corner of the page shows the date '2021-12-23 12:00:00'.

Previous Appointments:

NOVO Connect

Search symptoms, services, products, programs and more...

Toronto, Ontario

12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

Previous [View All](#)

Date	Appointment	Time	Provider
Dec 2nd	temp	Thursday, Dec 02, 2021 03:20 - 03:20	Ashut Mishra
Dec 15th	this is a test	Wednesday, Dec 15, 2021 5:31 PM	
Dec 16th	this is a test 2	Wednesday, Dec 16, 2021 5:36 PM	

Activate Windows

NOVO Connect

Search 4

Toronto, Ontario

Appointment

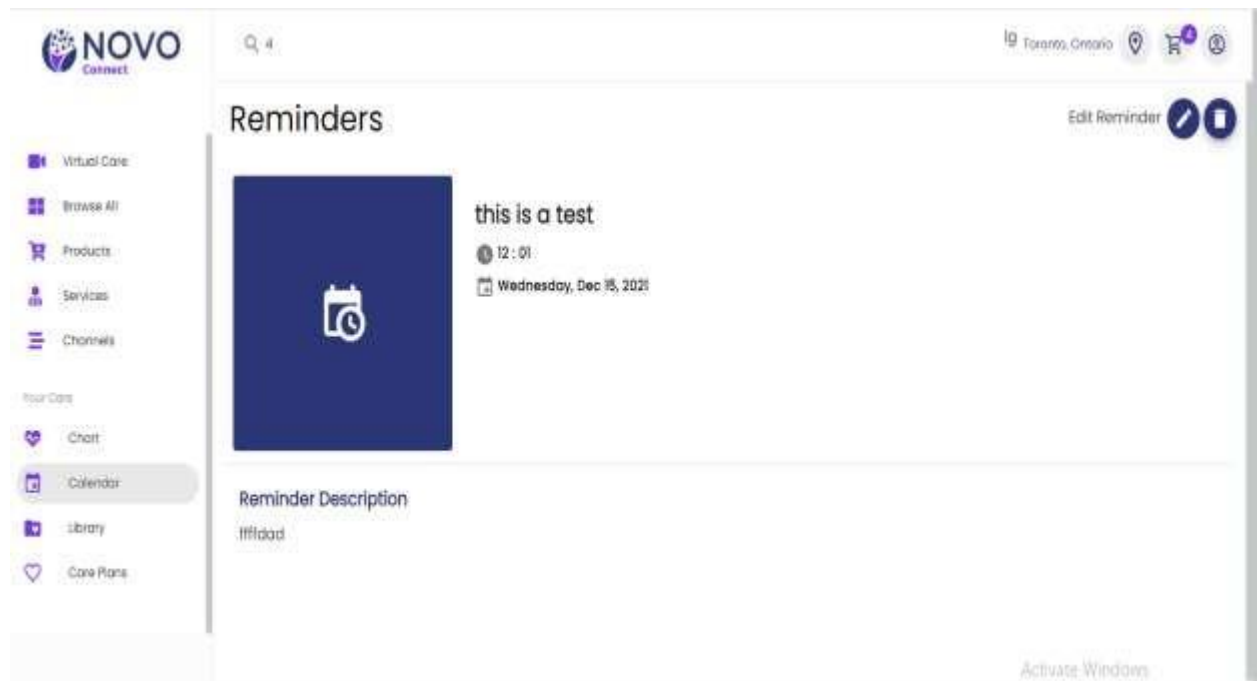
temp

December 2, 2021
10:20

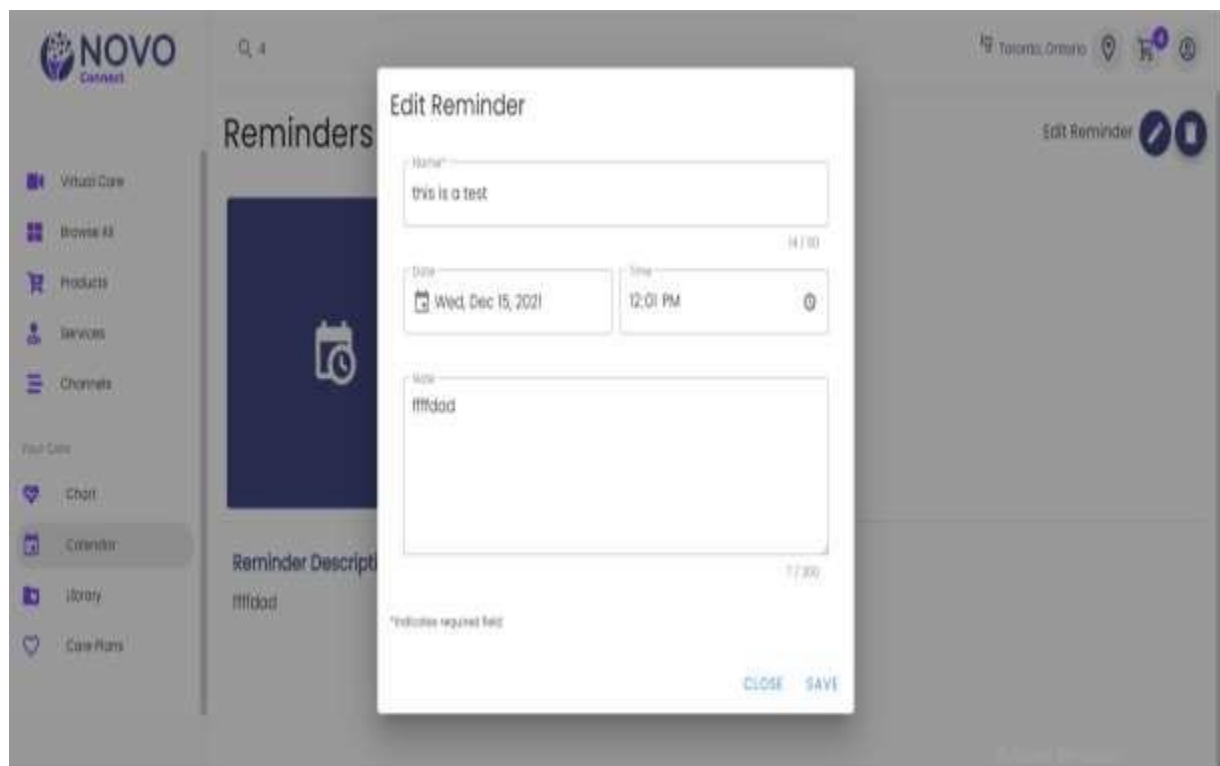
Appointment Details	Provider Details	Company Details
<p>Physical</p> <ul style="list-style-type: none"> Address Phone Email 	<ul style="list-style-type: none"> Name: Ashut Mishra Role: doctor 	<ul style="list-style-type: none"> Name Type

Activate Windows

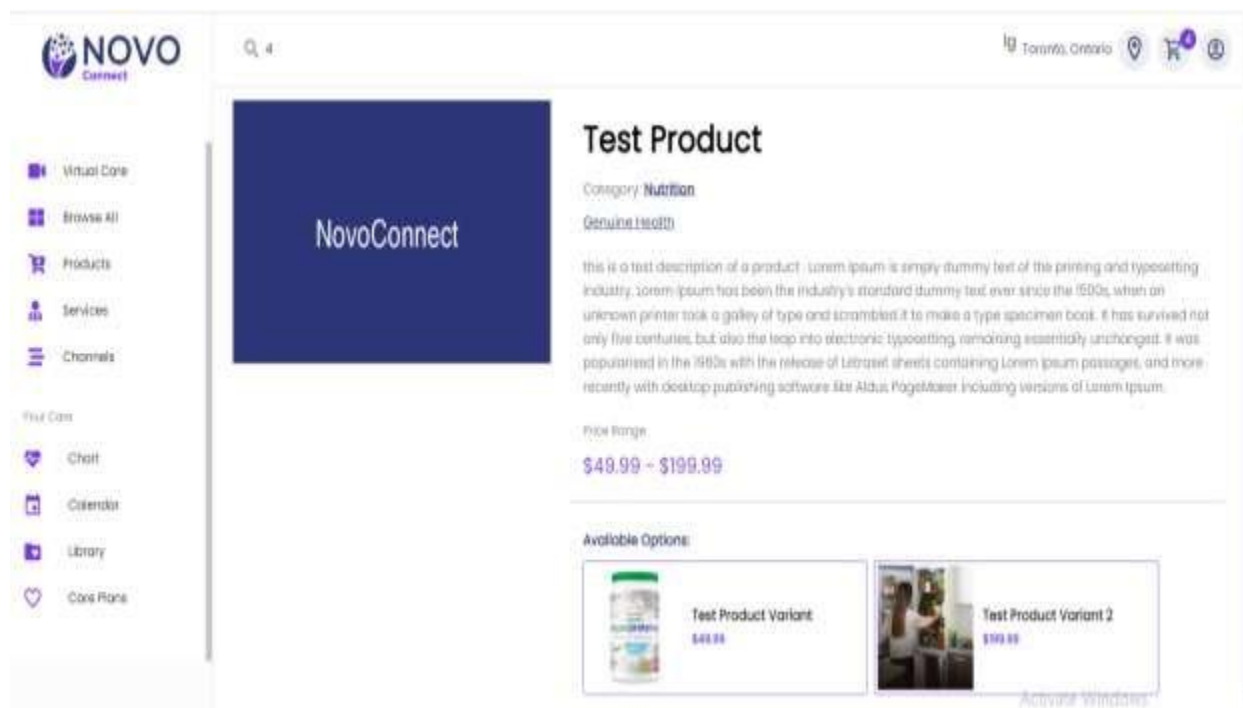
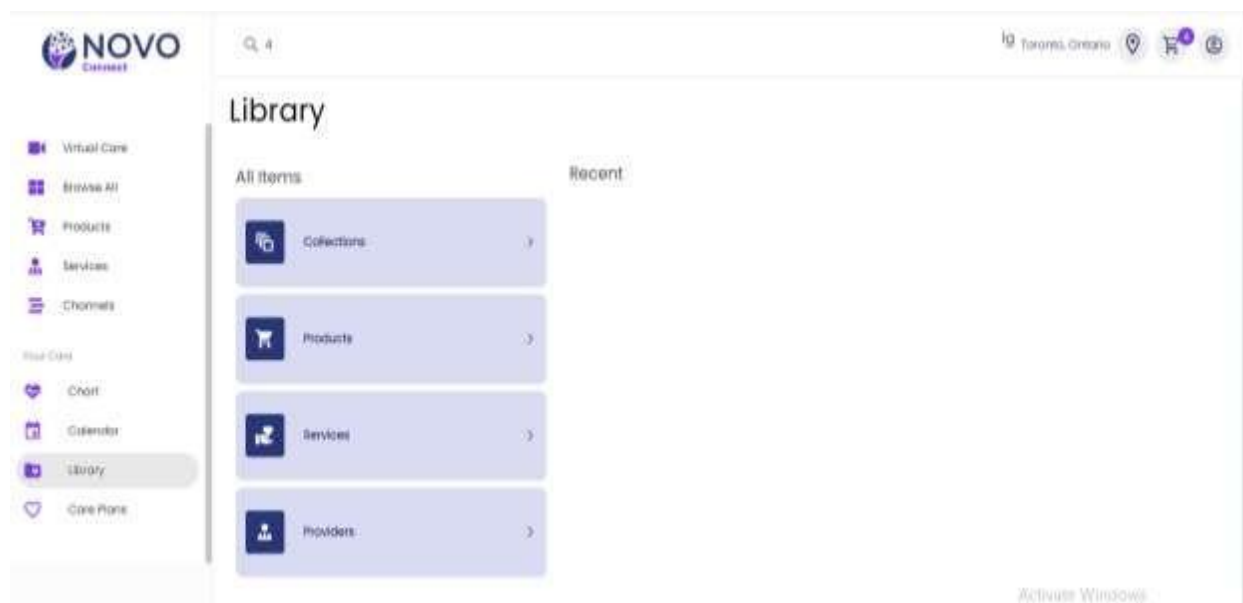
Reminder Page:



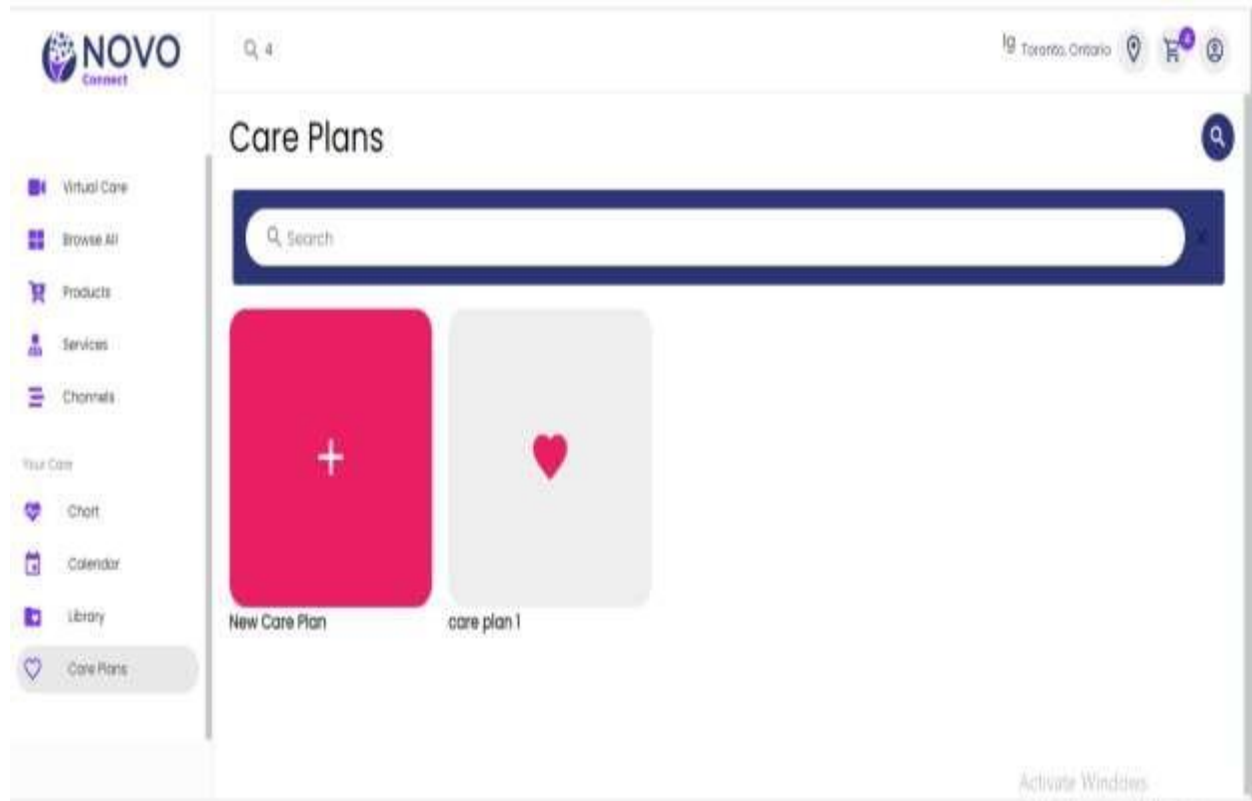
Edit Remainder Page:



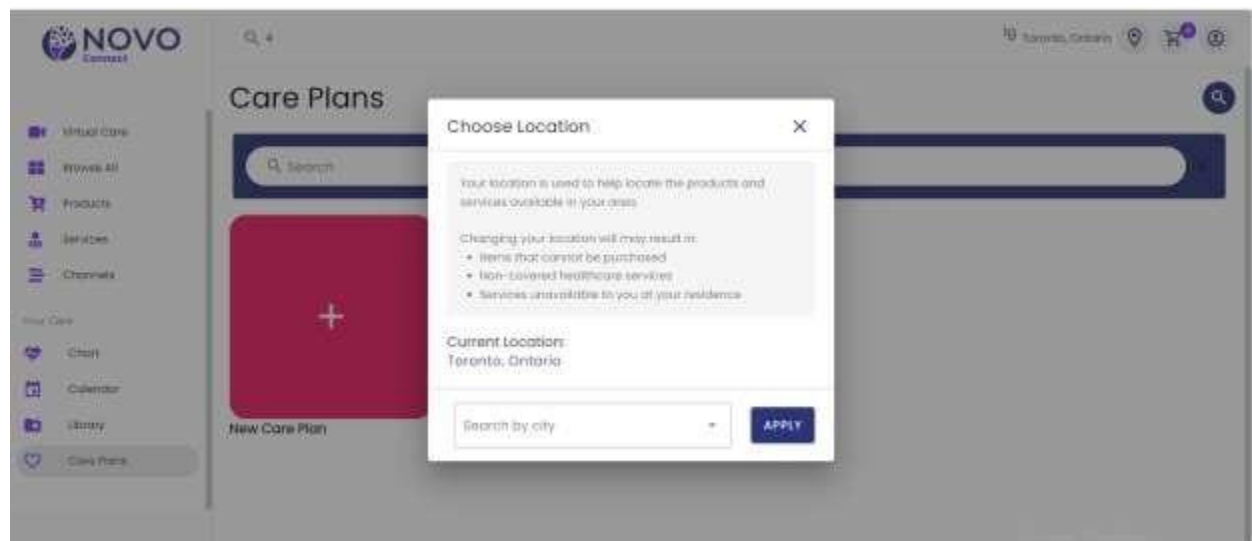
Library Page:



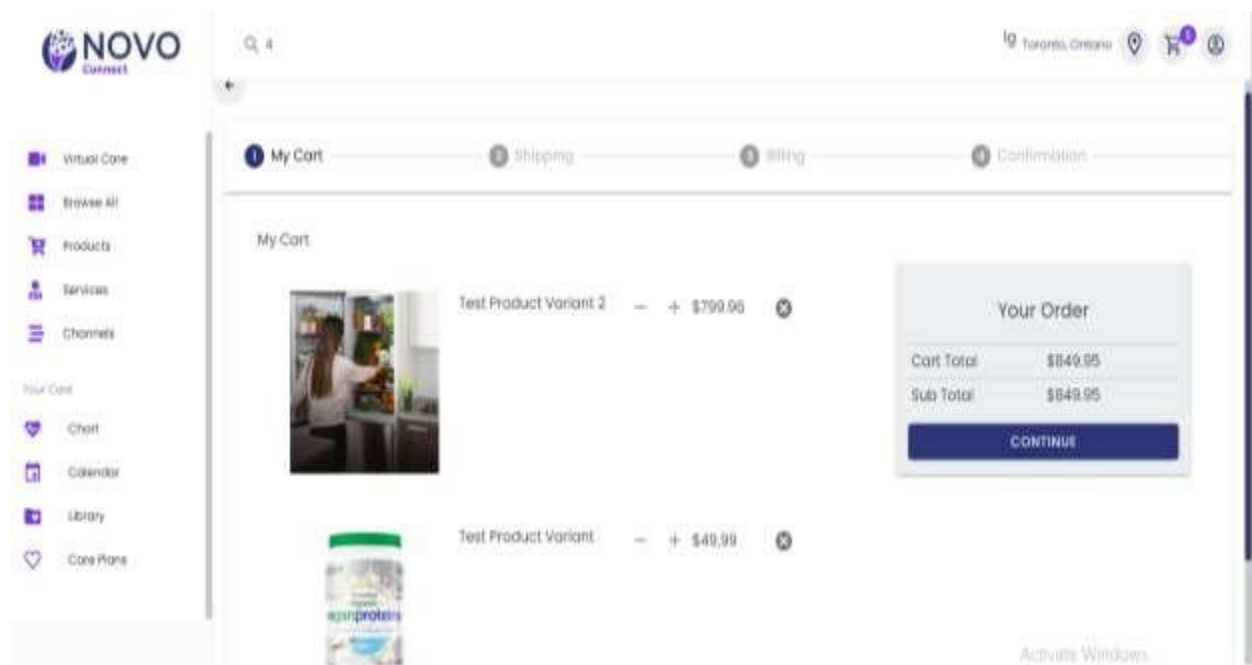
Available Options:



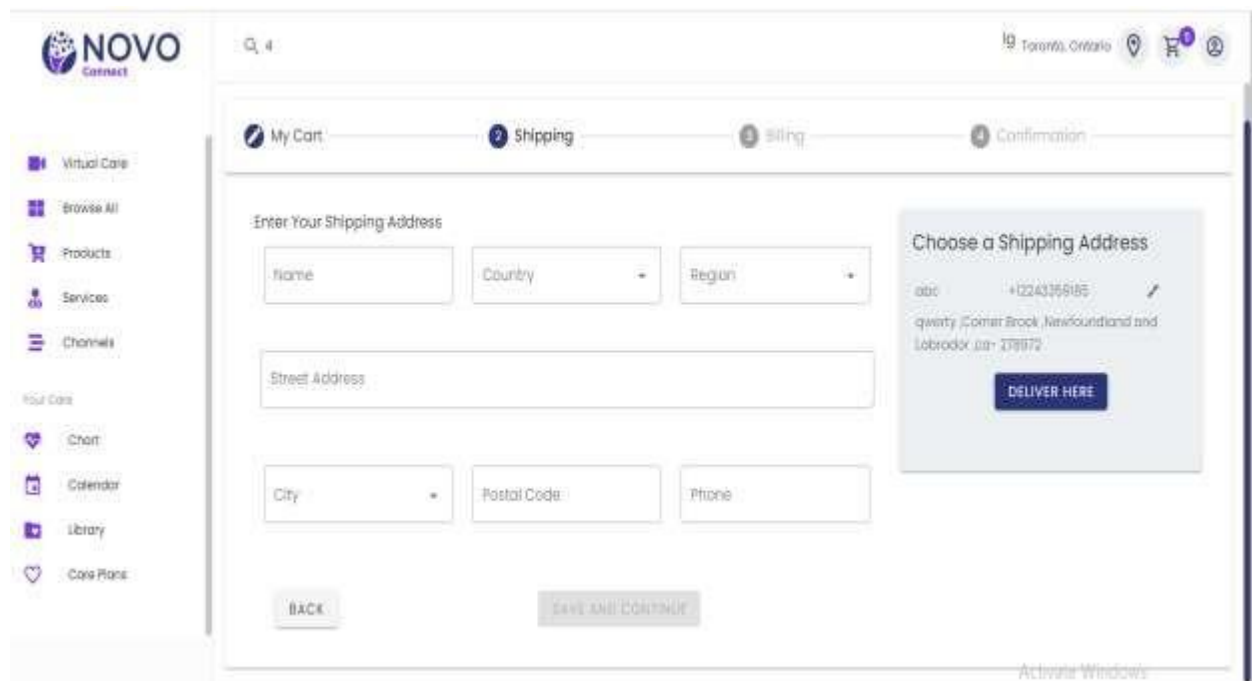
Choose Location :



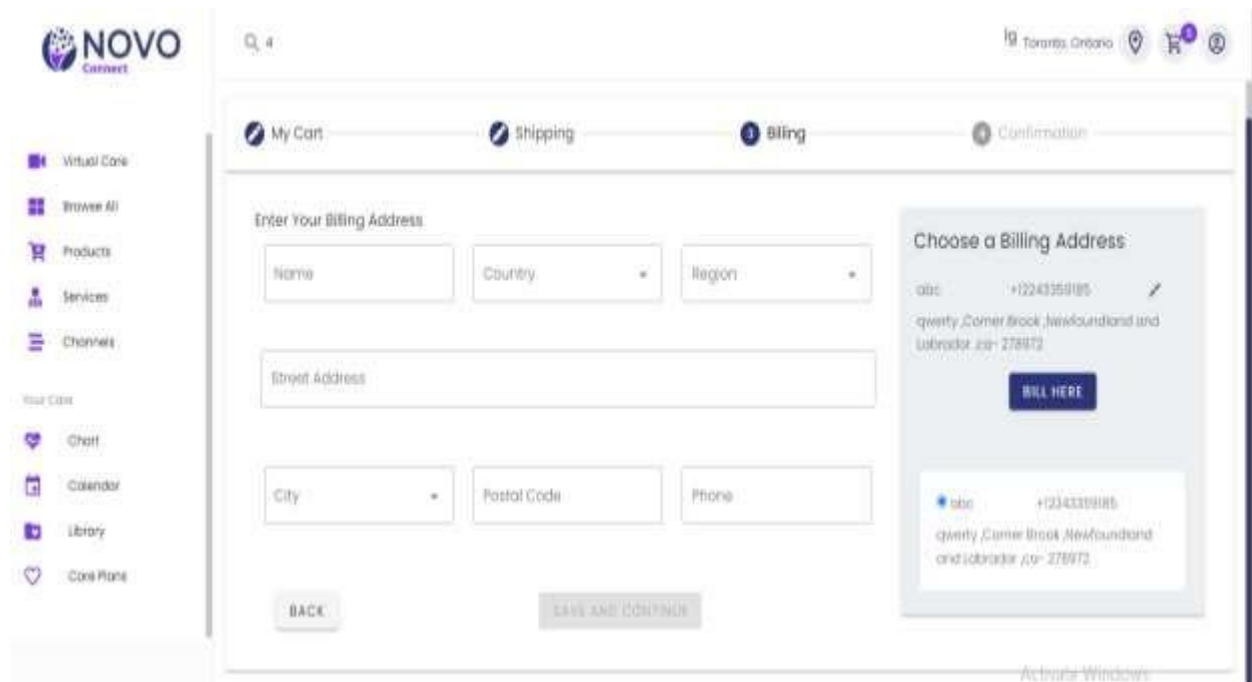
My Cart:



Shipping Page:



Billing Page:



The screenshot shows the 'Billing' step of a checkout process on the NOVO Connect website. The left sidebar contains navigation links: Virtual Care, Browse All, Products, Services, Channels, Your Cart, Chat, Calendar, Library, and Care Plans. The main content area has a progress bar with four steps: My Cart, Shipping, Billing (active), and Confirmation. Below the progress bar, there are two sections: 'Enter Your Billing Address' and 'Choose a Billing Address'.

Enter Your Billing Address:

Name Country Region

Street Address

City Postal Code Phone

[BACK](#) [SAVE AND CONTINUE](#)

Choose a Billing Address:

abc +12343358185 [✎](#)

qwerty, Corner Brook, Newfoundland and Labrador, A1A 2B2

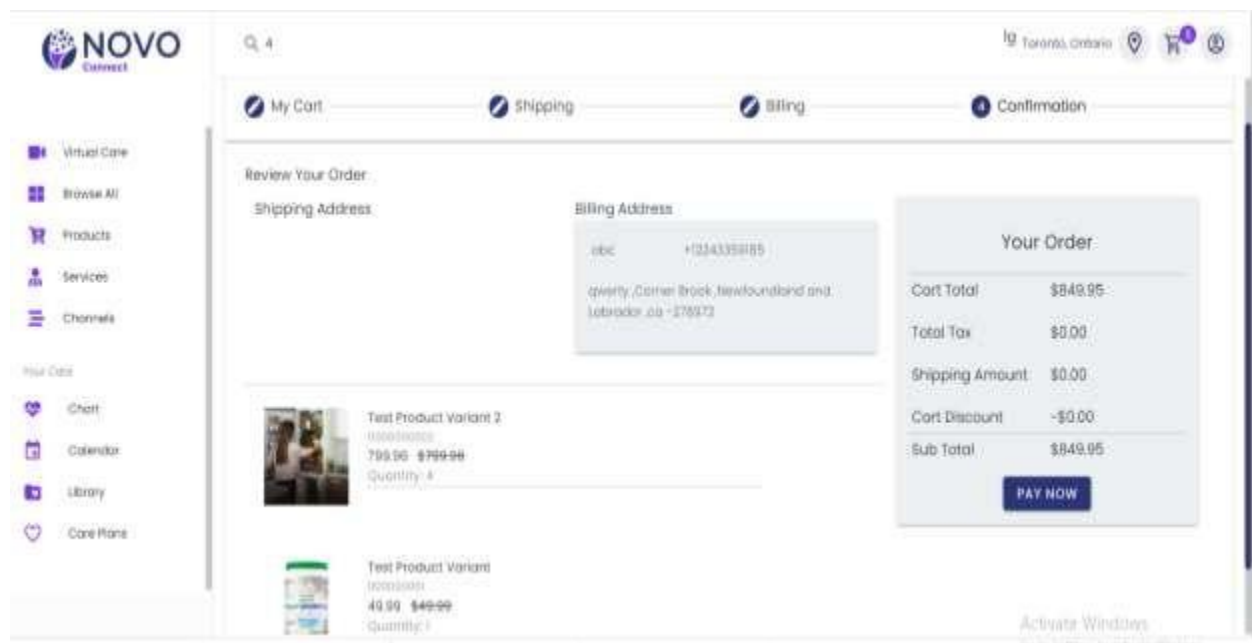
[BILL HERE](#)

abc +12343358185

qwerty, Corner Brook, Newfoundland and Labrador, A1A 2B2

Activate Windows

Confirmation Page:



The screenshot shows the 'Confirmation' step of a checkout process on the NOVO Connect website. The left sidebar is the same as the Billing page. The main content area has a progress bar with four steps: My Cart, Shipping, Billing, and Confirmation (active). Below the progress bar, there are three sections: 'Review Your Order', 'Shipping Address', and 'Billing Address'.

Review Your Order:

Shipping Address

Billing Address

abc +12343358185

qwerty, Corner Brook, Newfoundland and Labrador, A1A 2B2

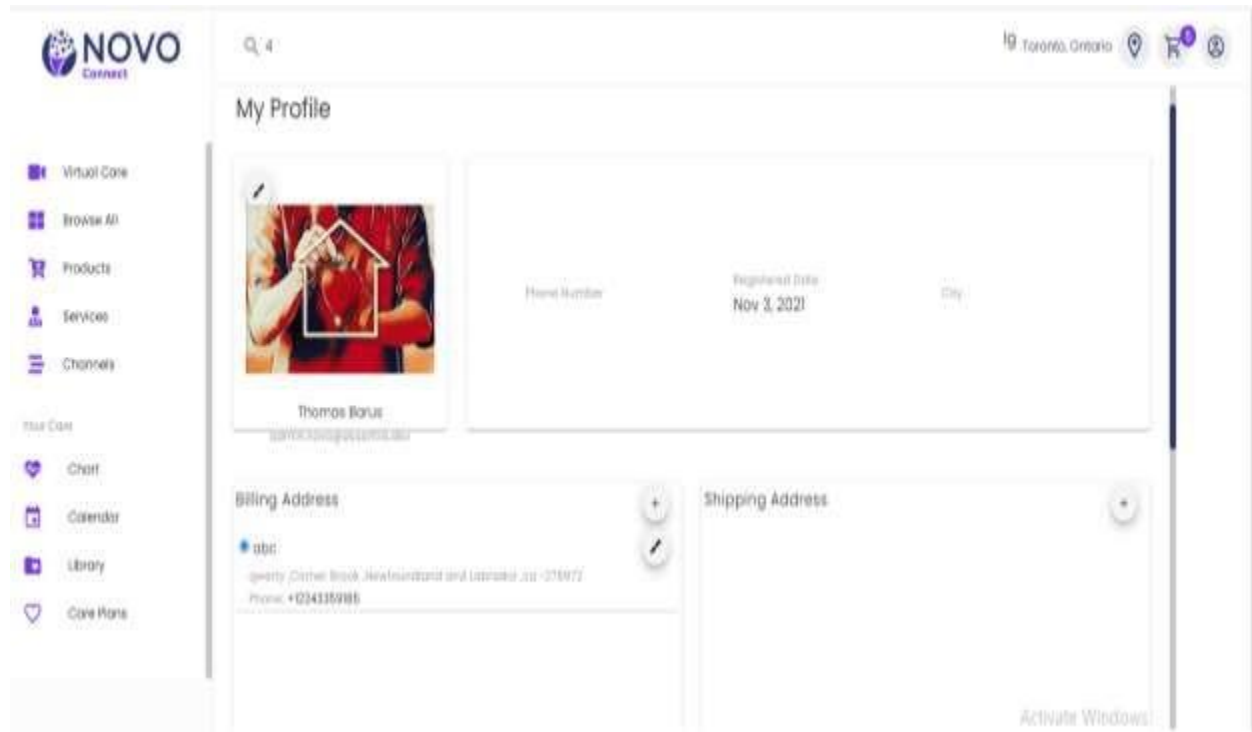
Your Order:

Your Order:	
Cart Total	\$849.95
Total Tax	\$0.00
Shipping Amount	\$0.00
Cart Discount	-\$0.00
Sub Total	\$849.95

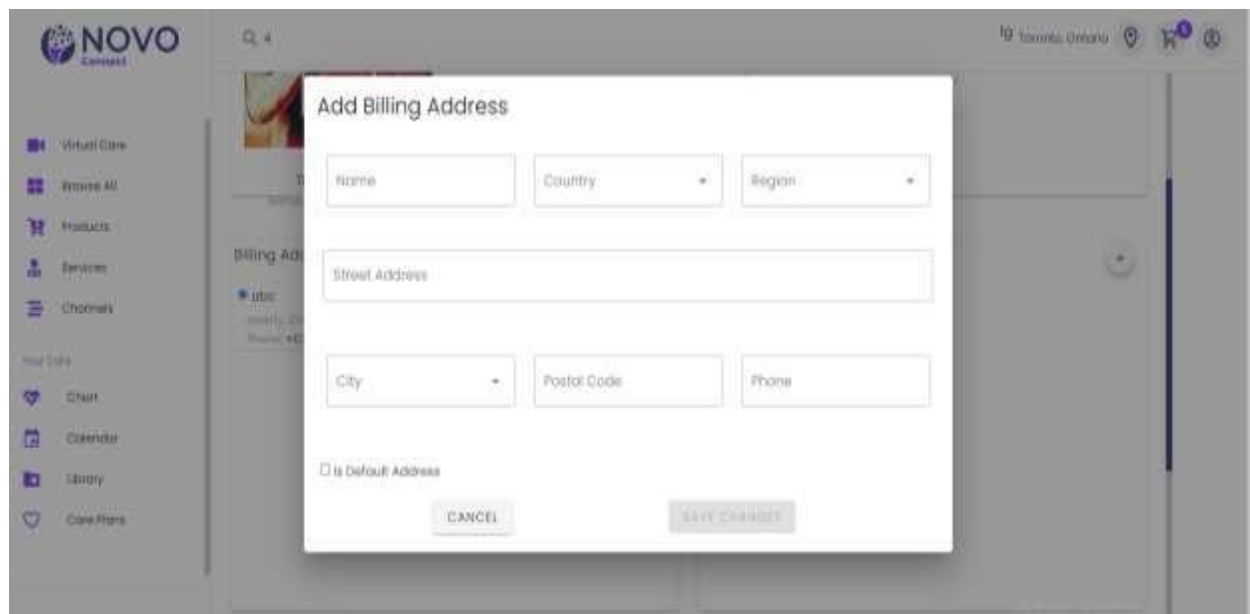
[PAY NOW](#)

Activate Windows

My Profile Page:



Add Billing Address Page:



Edit Billing Address:

The screenshot shows the 'Edit Billing Address' modal form overlaid on the NOVO Connect user interface. The form contains the following fields and options:

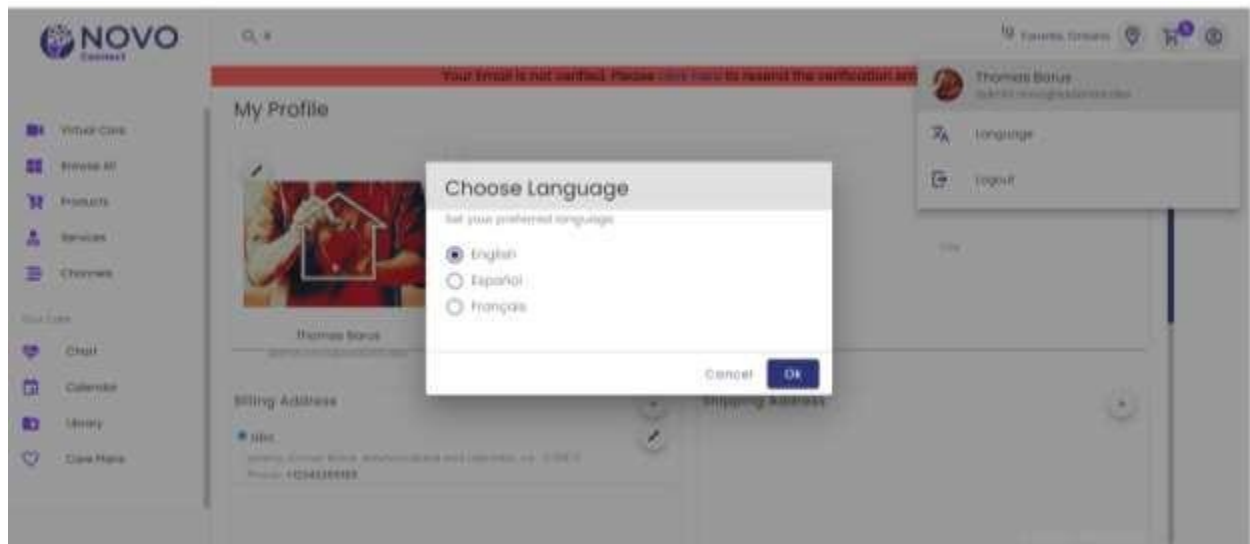
- Name:** A text input field containing 'abc'.
- Country:** A dropdown menu.
- Region:** A dropdown menu.
- Street Address:** A text input field containing 'qwerty'.
- City:** A dropdown menu.
- Postal Code:** A text input field containing '278972'.
- Phone:** A text input field containing '+12243356185'.
- Is Default Address:** An unchecked checkbox.
- Buttons:** 'CANCEL' and 'SAVE CHANGES' buttons at the bottom.

Add Shipping Address:

The screenshot shows the 'Add Shipping Address' modal form overlaid on the NOVO Connect user interface. The form contains the following fields and options:

- Name:** A text input field.
- Country:** A dropdown menu.
- Region:** A dropdown menu.
- Street Address:** A text input field.
- City:** A dropdown menu.
- Postal Code:** A text input field.
- Phone:** A text input field.
- Is Default Address:** An unchecked checkbox.
- Buttons:** 'CANCEL' and 'SAVE CHANGES' buttons at the bottom.

Choose Language:



Please Note:

Code Snippets of the project are not mentioned in this report in accordance with the Company's Policy.

REFERENCES:

- <https://vuejs.org/>
- <https://www.djangoproject.com/>
- <https://nuxtjs.org/>
- <https://www.geeksforgeeks.org/types-of-feasibility-study-in-software-project-development/>