

Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

B.Tech. CE Semester – V

Subject: (CE-515) Advanced Technologies

Project Title:

Personal Health Record System

[ PillPulse ]

By:

Shah Nidhi P. (roll no: CE125 id: 17CEUOS082)

Shah Vivek H. (roll no: CE126 id: 17CEUOG041)

Guided By:

Prof. Pinkal Chavda

Prof. Niyati Buch



Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

**CERTIFICATE**

This is to certify that Advanced Technologies project entitled “Personal Health Record System” is the bonafied report of work carried out by

1. **Shah Nidhi P. ( 17CEUOS082)**

**2) Shah Vivek H. (17CEUOG041)**

Of Department of Computer Engineering, Semester V, academic year 2019-20, under our supervision and guidance.

|  |  |  |
| --- | --- | --- |
| Guide | Guide | HOD |
|  |  |  |
| **Prof. Pinkal Chavada** | **Prof. Niyati Buch** | **Dr. C. K. Bhensdadia** |
| Assistant Professor of | Assistant Professor of | Head of the Department of |
| Department of Computer | Department of Computer | Department of Computer |
| Engineering, Dharmsinh Desai University, Nadiad. | Engineering, Dharmsinh Desai University, Nadiad. | Engineering, Dharmsinh Desai University, Nadiad. |

Table of Contents

1Abstract...…………………………..…………………..……………………………………..…42 Introduction……………………………………………………………………….….................5 2.1 Project Details: Brief Introduction …………………...............................................................5 2.2 Technology and Tools Used .....................................................................................................5 3 Software Requirement Specifications ………………………...……………………………...…7 3.1 Scope ………………………………………………………...………………………………..7 3.2 System Functional Requirements ……………….…………………………………………...7 3.3 Other Non‐Functional Requirements ……………….…………………………………….....13 4Design……………….……….……………………….……………………………..………….15 4.1 XML ………………………………………..…….……………………………..…………...15 4.2 DTD ………………………….………………………...….……………………..………….17 4.3 XSD ………………………..……………………………………………….………...……..18 4.4 XSLT……………………….…………………………………………………..……..……..18 4.5 DFD ……………………………………………………………………………..………..…24 4.6 E-R Diagram ………….………………………………………………..………………..…..25 4.6 data Dictionary……..…………………………………………………………………...…...26 5 Implementation Details ……………………………….……………………………….…...…33 5.1 Modules …………………………………..…………………………………………….…...34 5.2 Functional prototypes ……………………………………………………………………….34 6 Testing ………………………………………………………………………………………...36 6.1 Testing Method….………………………………….……………………………....……….36 6.4 Test Cases …………………………………………………………………………….....….36 7 Screen‐shots of the System …………………………………………………......……..……...38 8 Conclusion ……………………………………………………………………….…...……….50 9 Limitations and Future Extensions of System ……………………………………………...…51 10 Bibliography ……………………………………………………….……………………..….51

1. Abstract

With the increasing levels of data available, there is always a shortage of solutions to store and analyze it. Healthcare is a sector where the data inflow is high. The Personal Health Record System is an electronic version of patients’ health records. The Personal Health Record System is real-time, patient-centered and accessible solution that can be used to store the health records. Storing the health record in such systems allows the user to access the data anywhere anytime. Also it can be used to filter and analyze the health of the user using the health records available. Personal Health Record System like PillPulse ensures the data availability keeping the aspect of user’s privacy in mind.

2. Introduction

**2.1 Brief Introduction**

Personal Health Record System: PillPulse allows the users to store their health records data in the form of files to the centralized server which can be accessed from anywhere via internet. There is no limit on how many files the user can upload to the system.

User can view the records uploaded on the server and download them also. The system also maintains the personal health profile of the user where user can store his long term diseases, height, weight, blood group, emergency contact and other related health information. This medical profile serves as a reference to overall health of the user.

The system also maintains a medication history of the user. Once the user has fed his medications, user will be shown today’s medication along with the time when to take that medicine. It serves as a daily medication reminder for the patient.

**2.2 Tools/Technologies Used**

**Technologies:**

HTML 5

CSS 3

Bootstrap 4

Angular cli

Typescript

Node JS

Express JS

MongoDb

**Tools**

Visual Studio Code

Git

**Platforms**

localhost/4200 for Angular

localhost/8000 for Node js

mongodb://127.0.0.1:27017

3. Software Requirement Specifications

**3.1 Product Scope**

This system is designed to enable the user to store and retrieve medical records, medications and to maintain personal health profile.

**3.2 Types of User**

1. User/Patient

**3.3 System Functional Requirements**

**R.1 User**

R.1.1 Manage Medical Records

Description: User can store and retrieve medical records.

R.1.1.1 Add Record

Description: user will upload the medical record.

Input: Medical record file

Output: Success message

R.1.1.2 View Records

Description: user can view the records existing in the database.

Input: User selection

Output: list of existing records

R.1.1.3 Delete Record

Description: user can delete the record

Input : User selection.

Output: confirmation of deletion

R.1.1.4 Download Record

Description: user can download the record

Input : User selection.

Output: record as downloadable resource

R.1.2 Manage Medical Profile

Description: User maintain his/her medical profile

R.1.2.1 Add Medical Profile

Description: user will fed the data the account creation time.

Input: medical profile info

Output: Success message

R.1.2.2 View Medical Profile

Input: User selection

Output: Medical Profile

R.1.2.3 Update Record

Description: user can update data of his/her medical profile

Input: User medical profile data

Output: confirmation of success

R.1.3 Manage Medications

Description: User can store and see his/her medications.

R.1.3.1 Add Medication

Description: user will add the details of medication

Input: user medication details

Output: Success message

R.1.3.2 View Medication

Description: user can view the medication in the database.

Input: User selection

Output: list of current medications

R.1.3.3 Delete Medication

Description: user can delete the medication

Input : User selection.

Output: confirmation of deletion

**3.4 Other Nonfunctional Requirements**

**1. Performance**

The system must be interactive and must not involve long delays. Though in case of opening the app components or loading the page the system shows the delays less than 2 seconds.

The performance of uploading and downloading the medical records is highly based on internet bandwidth and physical distance from the server.

**2. Safety**

The users’ medical data is highly personal. The system has authorization to avoid any un-authorized access to user’s private health record files and data.

**3. Reliability**

As the system has medical data, its reliability is the major factor for consideration.

**4. Database**

System requires to access users data fast to maintain the performance/

4. Design

**4.1 XML**

<?xml version="1.0" encoding="utf-8"?>

<tickets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<tickets>

<ticket id="T00001">

<c\_id>C00001</c\_id>

<destId>D00004</destId>

<tkttype>Plane</tkttype>

</ticket>

<ticket id="T00002">

<c\_id>C00002</c\_id>

<destId>D00002</destId>

<tkttype>Train</tkttype>

</ticket>

<ticket id="T00003">

<c\_id>C00003</c\_id>

<destId>D00002</destId>

<tkttype>Bus</tkttype>

</ticket>

<ticket id="T00004">

<c\_id>C00004</c\_id>

<destId>D00001</destId>

<tkttype>Train</tkttype>

</ticket>

<ticket id="T00005">

<c\_id>C00005</c\_id>

<destId>D00003</destId>

<tkttype>Plane</tkttype>

</ticket>

<ticket id="T00006">

<c\_id>C00003</c\_id>

<destId>D00006</destId>

<tkttype>Train</tkttype>

</ticket>

</tickets>

<?xml version="1.0" encoding="UTF-8"?>

<packages xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<package id="1">

<pkgname>Goa</pkgname>

<pkgcost>1000</pkgcost>

<noofdays>4</enddate>

<initdest>Surat</initdest>

<pkgdetails>Enjoy the package.</pkgdetails>

</package>

<package id="2">

<pkgname>Leh-Ladakh</pkgname>

<pkgcost>10000</pkgcost>

<noofdays>6</enddate>

<initdest>Ahmedabad</initdest>

<pkgdetails>Enjoy the package.</pkgdetails>

</package>

</packages>

**4.2 DTD**

<!DOCTYPE packages[

<!ELEMENT packages (package\*)>

<!ELEMENT package (pkgname,pkgcost,stdate,enddate,initdest,pkgdetails)>

<!ELEMENT pkgname (#PCDATA)

<!ELEMENT noofdays (#PCDATA)>

<!ELEMENT initdest (#PCDATA)>

<!ELEMENT pkgdetails (#PCDATA)>

<!ATTLIST package pkgid ID #REQUIRED>

]>

**4.3 XSD**

<?xml version="1.0" encoding="utf-8"?>

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="tickets">

<xs:complexType>

<xs:sequence>

<xs:element name="ticket" type="TicketType" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:complexType name="TicketType">

<xs:all>

<xs:element name="c\_id" type="xs:string">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:pattern value="C[0-9][0-9][0-9][0-9][0-9]"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="destId" type="xs:string"/>

<xs:element name="tkttype" type="xs:string">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="plane"/>

<xs:enumeration value="train"/>

<xs:enumeration value="bus"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

</xs:all>

<xs:attribute name="id" type="xs:string" use="required"/>

</xs:complexType>

</xs:schema>

<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="employees">

<xs:complexType>

<xs:sequence>

<xs:element name="package" type="PackageType" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:complexType name="PackageType">

<xs:all>

<xs:element name="pkgname" type="NameSimpleType"/>

<xs:element name="noofdays" type="DateSimpleType"/>

<xs:element name="initdest" type="NameSimpleType"/>

<xs:element name="pkgdetails" type="DetailSimpleType"/>

</xs:all>

<xs:attribute name="pkgid" type="xs:int" use="required"/>

</xs:complexType>

<xs:simpleType name="NameSimpleType">

<xs:restriction base="xs:string">

<xs:minLength value="0"/>

<xs:maxLength value="256"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="DateSimpleType">

<xs:restriction base="xs:date">

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="DetailSimpleType">

<xs:restriction base="xs:string">

<xs:minLength value="0"/>

<xs:maxLength value="256"/>

</xs:restriction>

</xs:simpleType>

</xs:schema>

**4.4 XSLT**

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org/1999/xhtml" version="1.0">

<xsl:template match="/">

<html>

<body>

<h2>Packages</h2>

<table border="1">

<tr>

<th>Packagename</th>

<th>Cost</th>

<th>No of days</th>

</tr>

<xsl:for-each select="packages/package">

<tr>

<td><xsl:value-of select="pkgname"/></td>

<td><xsl:value-of select="pkgcost"/></td>

<td><xsl:value-of select="noofdays"/></td>

</tr>

</xsl:for-each>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

<?xml version="1.0" encoding="UTF-8" ?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

<h1>Ticket Listing</h1>

<xsl:apply-templates/>

</body>

</html>

</xsl:template>

<xsl:template match="ticket">

<div>

<h3>

Ticket ID :

<xsl:value-of select="@id"/>

</h3>

<xsl:apply-templates select="c\_id"/>

<xsl:apply-templates select="destId"/>

<xsl:apply-templates select="tktType"/>

</div>

</xsl:template>

<xsl:template match="c\_id">

<b>Customer Id :</b>

<xsl:value-of select="."/>

<br />

</xsl:template>

<xsl:template match="destId">

<b>Destination Id :</b>

<xsl:value-of select="."/>

<br />

</xsl:template>

<xsl:template match="tktType">

<b>Ticket Type :</b>

<xsl:value-of select="."/>

<br />

</xsl:template>

<xsl:template match="notes">

<b>Remarks :</b>

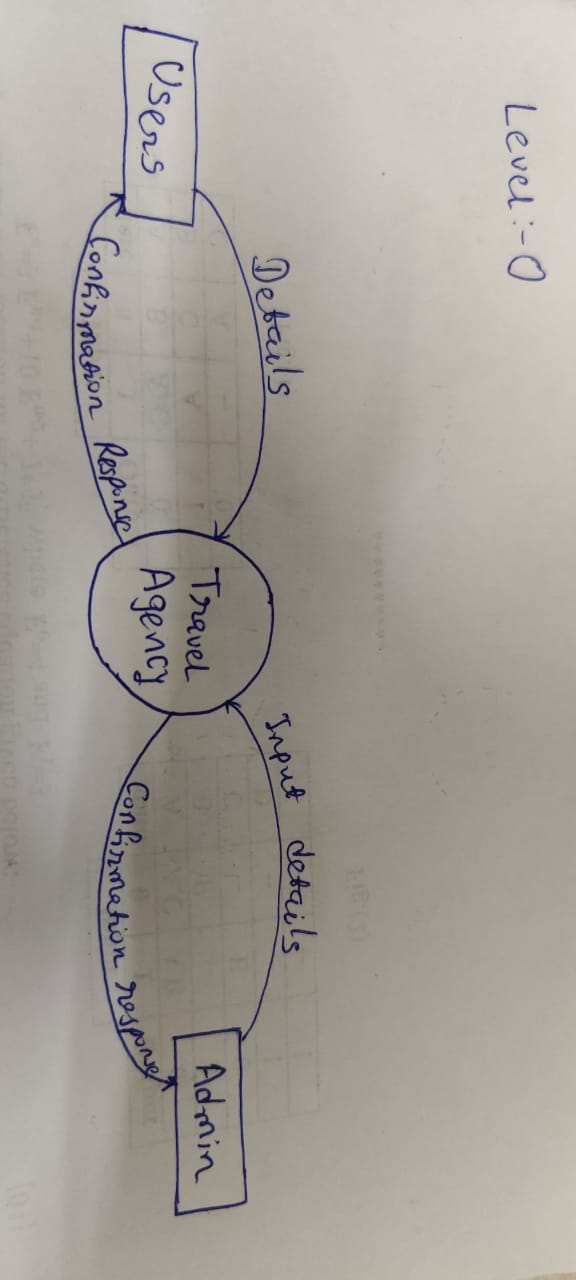
<xsl:value-of select="."/>

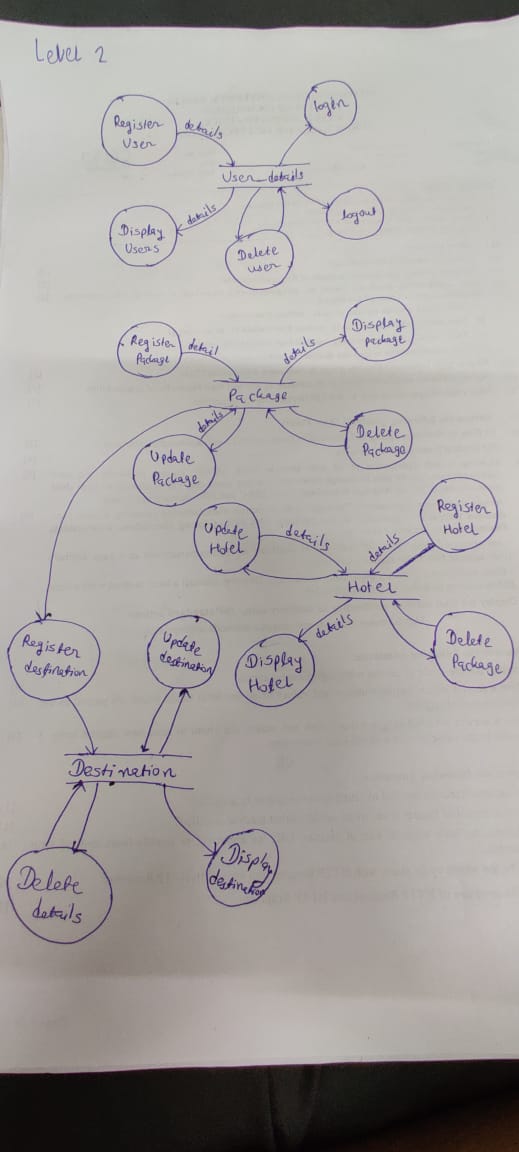
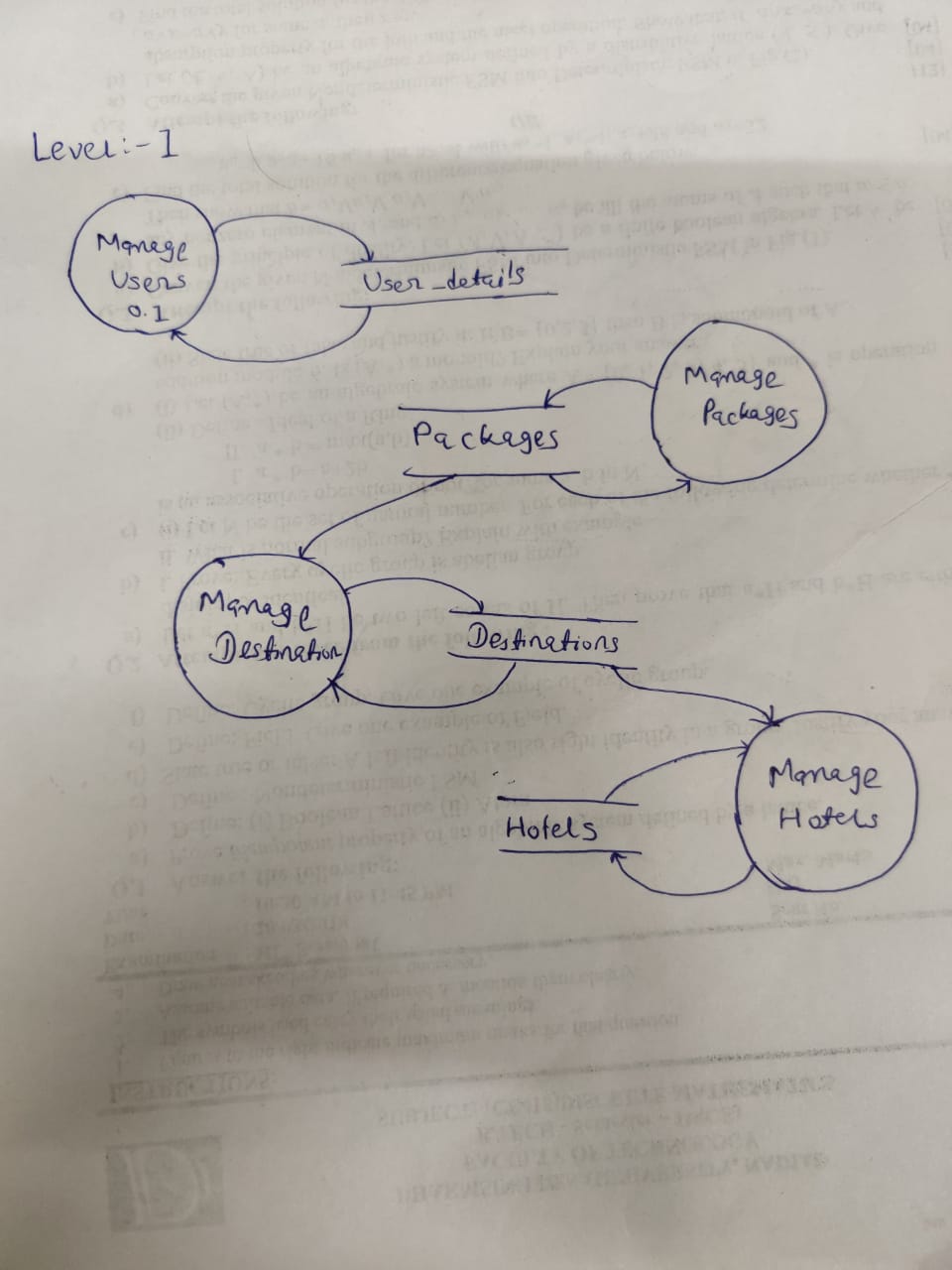
<br />

</xsl:template>

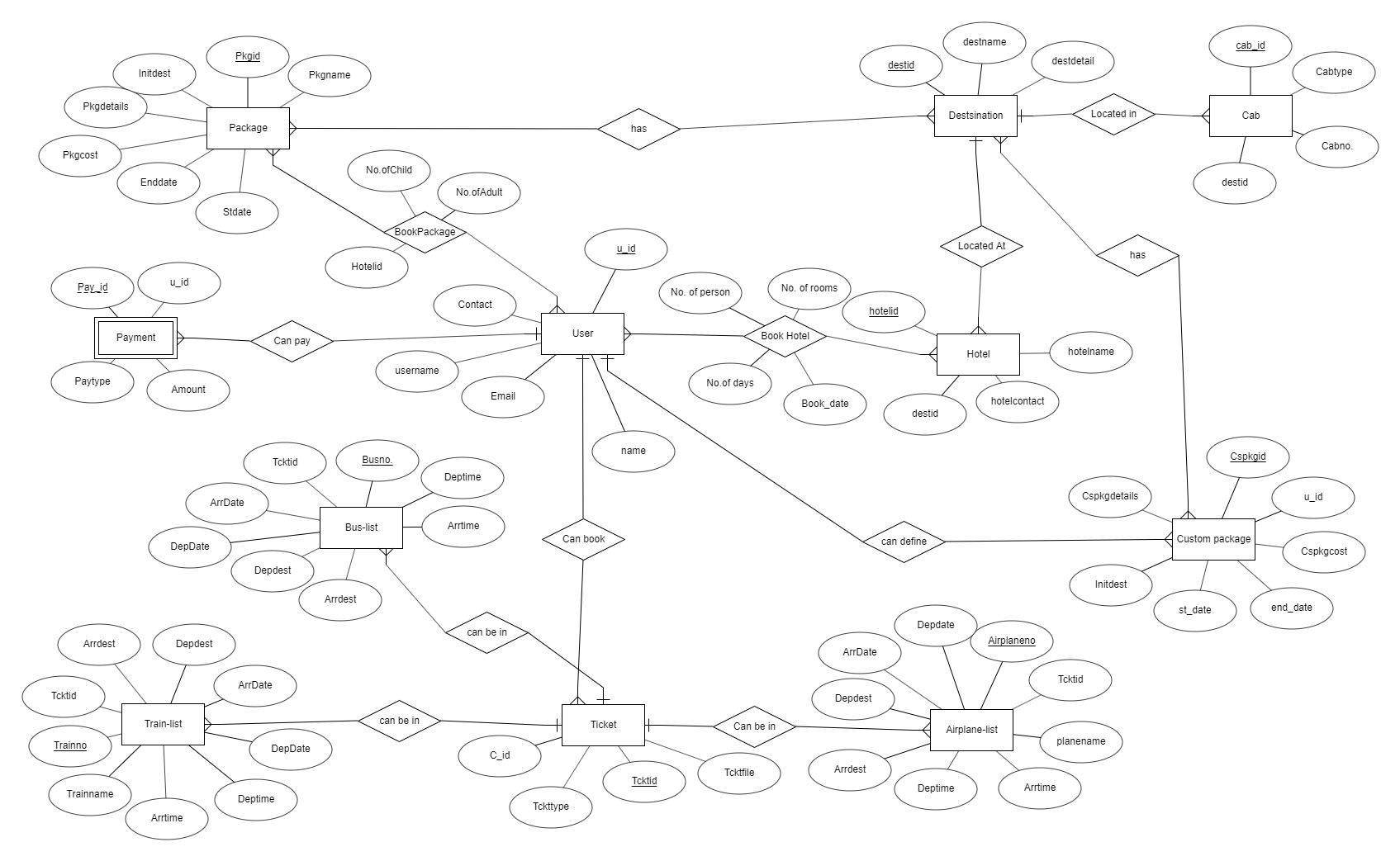
</xsl:stylesheet>

**4.5 DFD**

****

****

**4.6 E-R Diagram**

****

**4.7 Data Dictionary**

| **1.User** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | \_id | Varchar | 10 | Yes | Yes | PK |  | Auto increment key generated by database itself |
| 2. | uname | Varchar | 10 | Yes | Yes |  |  | Unique id defined by user to login |
| 3. | password | Varchar | 8 | yes | no |  |  |  |
| 4. | name | Varchar | 50 | Yes | No |  |  |  |
| 5. | uemail | Varchar | 30 | Yes | Yes |  |  |  |

| **2.Package** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Pkgid | Varchar | 10 | Yes | Yes | PK |  |  |
| 2. | Pkgname | Varchar | 20 | Yes | No |  |  |  |
| 3. | Pkgcost | Number | 8 | Yes | No |  |  |  |
| 4. | No Of Days | Number |  | Yes | No |  |  |  |
| 5. | Initdest | Varchar | 20 | Yes | No |  |  |  |
| 6. | Pkgdetail | Varchar | 100 | No | No |  |  |  |

| **3.Destination** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Destid | Varchar | 10 | Yes | Yes | PK |  |  |
| 2. | Destname | Varchar | 20 | Yes | No |  |  |  |
| 3. | Destdetail | Varchar | 100 | No | No |  |  |  |
| 4. | Pkgid | Varchar | 20 | Yes | No |  |  |  |

| **4.Hotel** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Hotelid | Varchar | 10 | Yes | Yes | PK |  |  |
| 2. | Hotelnm | Varchar | 30 | Yes | No |  |  |  |
| 3. | Hcost | Number | 10 | Yes | No |  |  |  |
| 4. | Destid | Varchar | 10 | Yes | No | FK | Destination |  |

| **5.BookPackage** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Pkgid | Varchar | 10 | Yes | Yes | FK | Package |  |
| 2. | c\_id | Varchar | 10 | Yes | Yes | FK | Customer |  |
| 4. | No of person | Number | 5 | Yes | No |  |  | Defines number of person are in the booked package. |
| **Primary key(Pkgid,c\_id)** | | | | | | | | |

| **6.Airplane-list** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Planeno. | Varchar | 10 | Yes | Yes | PK |  |  |
| 2. | PName | Varchar | 20 | Yes | No |  |  |  |
| 3. | Depdate | Date |  | Yes | No |  |  |  |
| 4. | Arrdate | Date |  | Yes | No |  |  |  |
| 5. | Deptime | Varchar | 20 | Yes | No |  |  |  |
| 6. | Arrtime | Varchar | 20 | Yes | No |  |  |  |
| 7. | Depplace | Varchar | 20 | Yes | No |  |  |  |
| 8. | Arrplace | Varchar | 20. | Yes | No |  |  |  |
| 9. | Tcktid | Varchar | 10 | Yes | No | FK | Ticket |  |

| **7.Train-list** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Trainno. | Varchar | 10 | Yes | Yes | PK |  |  |
| 2. | TrName | Varchar | 20 | Yes | No |  |  |  |
| 3. | Depdate | Date |  | Yes | No |  |  |  |
| 4. | Arrdate | Date |  | Yes | No |  |  |  |
| 5. | Deptime | Varchar | 20 | Yes | No |  |  |  |
| 6. | Arrtime | Varchar | 20 | Yes | No |  |  |  |
| 7. | Depplace | Varchar | 20 | Yes | No |  |  |  |
| 8. | Arrplace | Varchar | 20. | Yes | No |  |  |  |
| 9. | Tcktid | Varchar | 10 | Yes | No | FK | Ticket |  |

| **8.Bus-list** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Busno. | Varchar | 10 | Yes | Yes | PK |  |  |
| 2. | BusName | Varchar | 20 | Yes | No |  |  |  |
| 3. | Depdate | Date |  | Yes | No |  |  |  |
| 4. | Arrdate | Date |  | Yes | No |  |  |  |
| 5. | Deptime | Varchar | 20 | Yes | No |  |  |  |
| 6. | Arrtime | Varchar | 20 | Yes | No |  |  |  |
| 7. | Depplace | Varchar | 20 | Yes | No |  |  |  |
| 8. | Arrplace | Varchar | 20. | Yes | No |  |  |  |
| 9. | Tcktid | Varchar | 10 | Yes | No | FK | Ticket |  |

| **9.Ticket** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Tcktid | Varchar | 10 | Yes | Yes | PK |  |  |
| 2. | c\_id | Varchar | 10 | Yes | No | FK | Customer |  |
| 3. | Tckttype | Varchar | 10 | Yes | No |  |  |  |
| 4. | Tcktfile | Varchar | 30 | Yes | No |  |  |  |
|  |  |  |  |  |  |  |  |  |

| **10.Cab** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Cabid | Varchar | 10 | Yes | Yes | PK |  |  |
| 2. | Cabtype | Varchar | 20 | Yes | No |  |  |  |
| 3. | Cabno | Varchar | 20 | Yes | Yes |  |  |  |
| 4. | Destid | Varchar | 10 | Yes | No | FK | Destination |  |

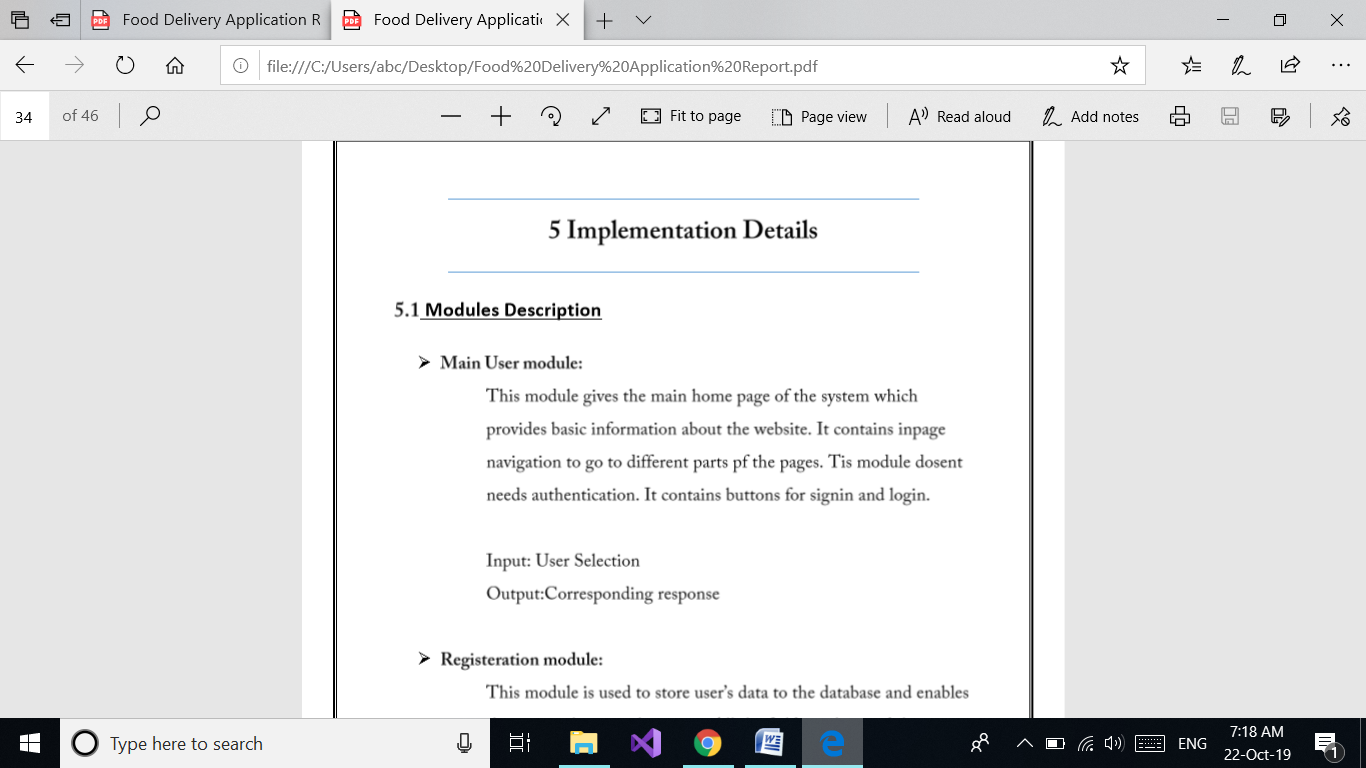
| **11.Book-hotel** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | Hbookid | Varchar | 10 | Yes | Yes | PK |  |  |
| 2. | Hotelid | Varchar | 10 | Yes | Yes | FK | Hotel |  |
| 3. | Pkgid | Varchar | 10 | No | No | FK | Package |  |
| 4. | Cspkgid | Varchar | 10 | No | No | FK | Self-Package |  |
| 4. | c\_id | Varchar | 10 | Yes | No | FK | Customer |  |
| 5. | book\_date | Date |  | Yes | No |  |  |  |
| 6. | No. of rooms | Number | 3 | Yes | No |  |  |  |
| 7. | No. of days | Number | 3 | Yes | No |  |  |  |
| 8. | Persons | Number | 3 | Yes | No |  |  |  |

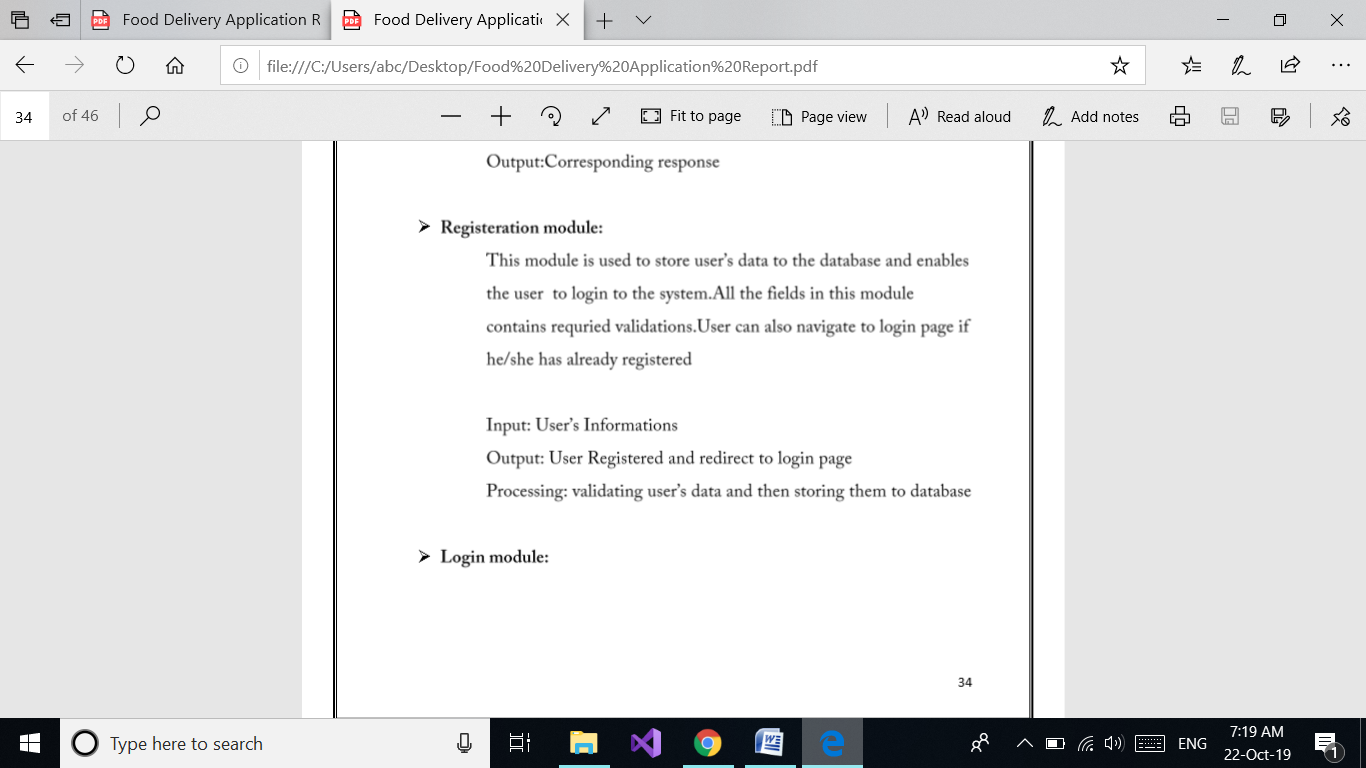
| **12.Custom package** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| If a customer needs to define his own package then he can add a package, too. This table is for the packages defined by customers. | | | | | | | | |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced table | Description |
| 1. | CsPkgid | Varchar | 10 | Yes | Yes | PK |  |  |
| 2. | C\_id | Varchar | 10 | Yes | No | FK | Customer |  |
| 3. | CsPkgcost | Number | 8 | Yes | No |  |  |  |
| 4. | Stdate | Date |  | Yes | No |  |  |  |
| 5. | Enddate | Date |  | Yes | No |  |  |  |
| 6. | Initdest | Varchar | 20 | Yes | No |  |  |  |
| 7. | CsPkgdetail | Varchar | 100 | No | No |  |  |  |

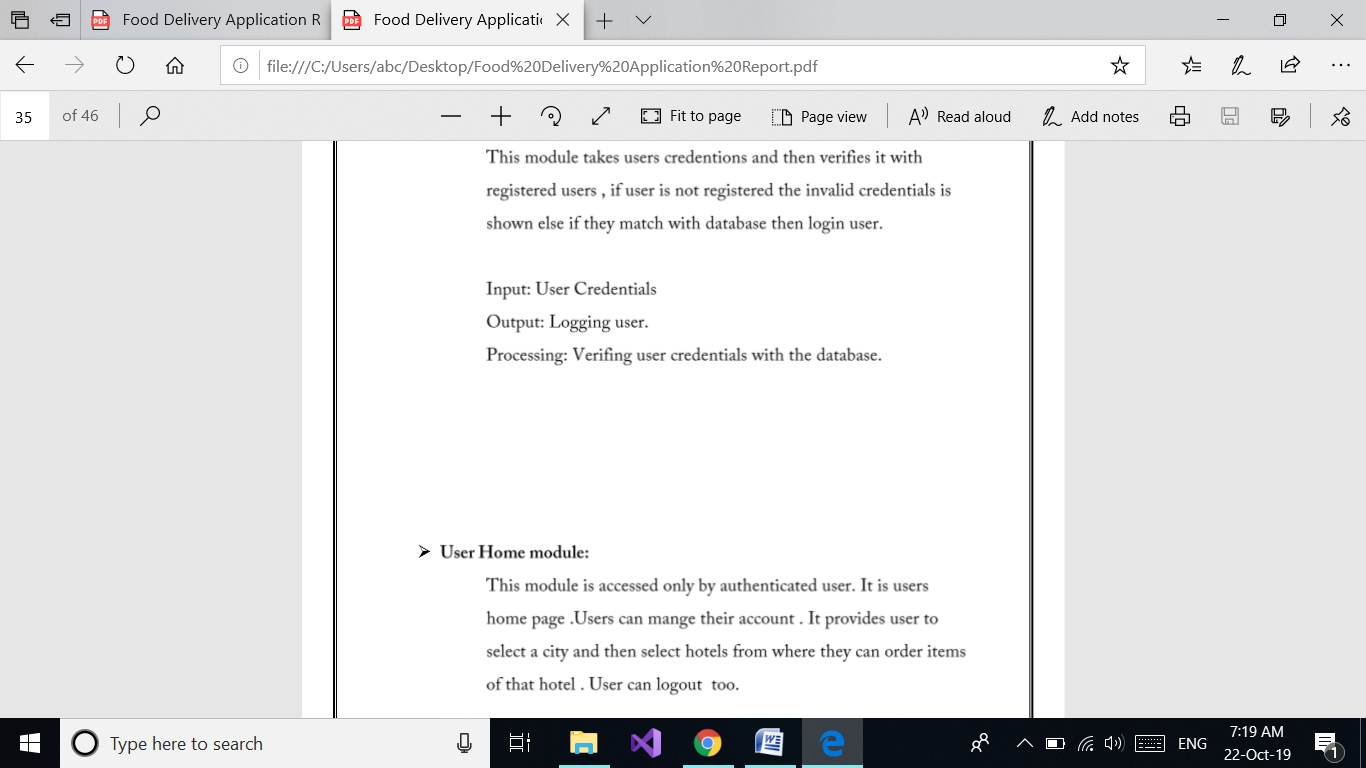
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **13.cspackage-place** | | | | | | | | |
| Sr. No. | **Field Name** | **Data Type** | Width | Required | Unique | PK/FK | **Referenced table** | Description |
| 1. | CsPkgid | Varchar | 10 | Yes | No | FK | Custom Package |  |
| 2. | Destid | Varchar | 10 | Yes | No | FK | Destination |  |
| **Primary key(CsPkgid,Destid)** | | | | | | | | |

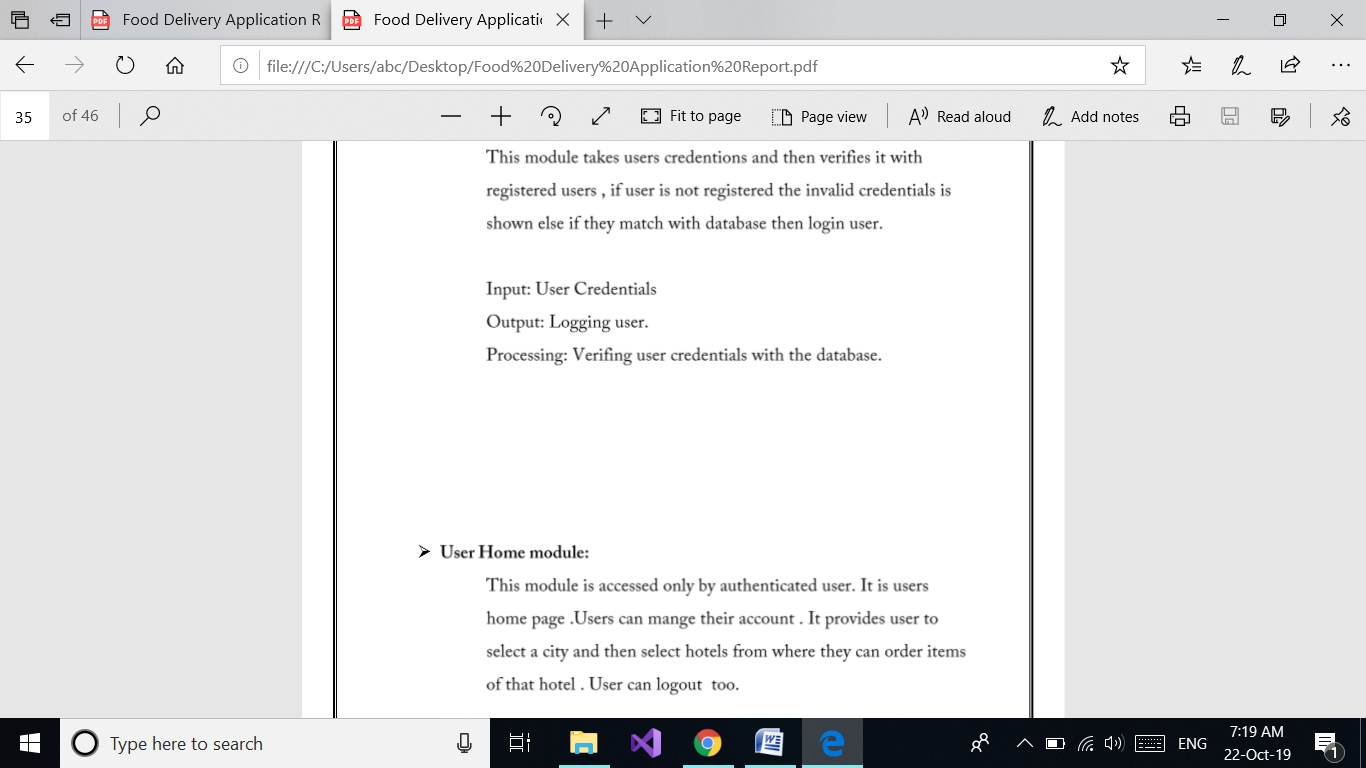
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **14.payment** | | | | | | | | |
| Sr. No. | **Field Name** | **Data Type** | **Width** | **Required** | Unique | PK/FK | **Referenced table** | Description |
| **1.** | **Pay\_id** | **Varchar** | **10** | **Yes** | **Yes** | **PK** |  |  |
| **2.** | **u\_id** | **Varchar** | **10** | **Yes** | **Yes** | **FK** | **User** |  |
| **3.** | **Amount** | **Number** | **8** | **Yes** | **No** |  |  |  |
| **4.** | **Paytype** | **Varchar** | **10** | **Yes** | **No** |  |  |  |

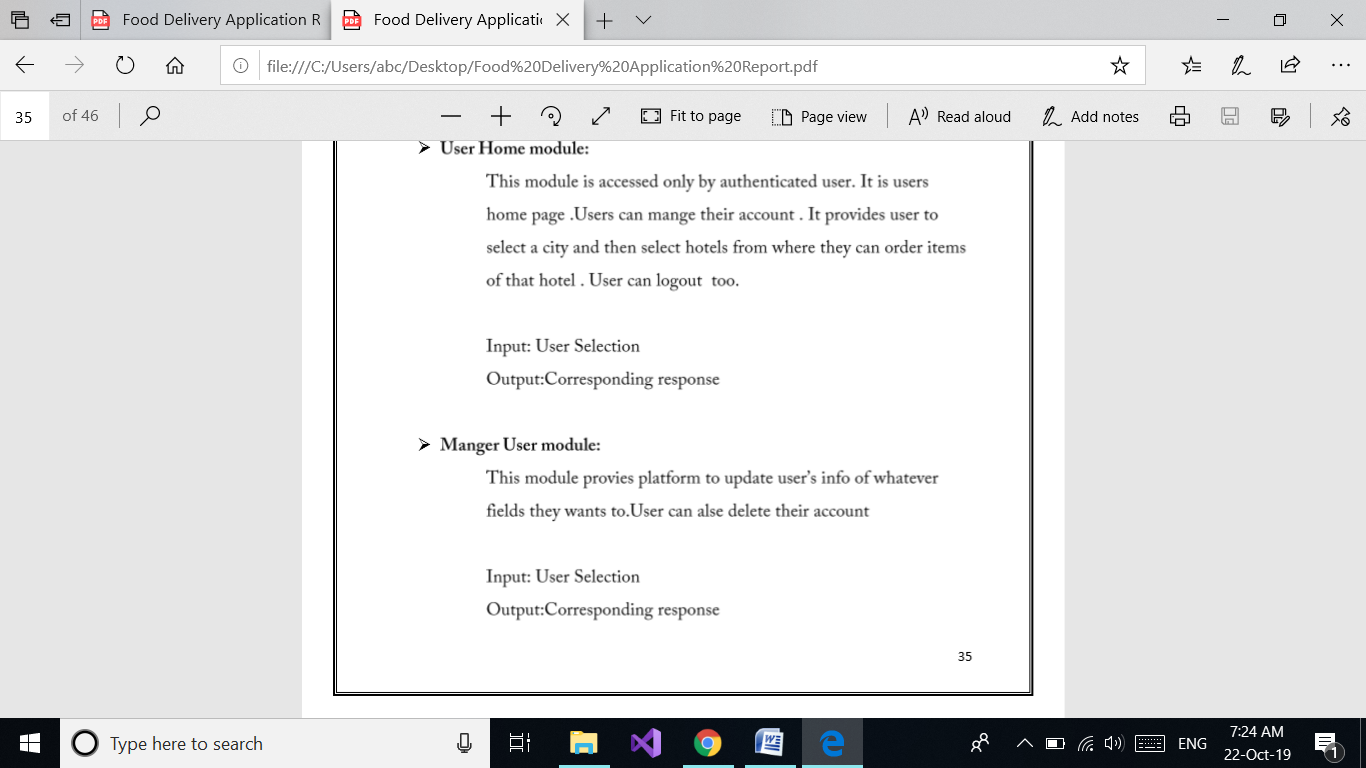
5. Implementation Details





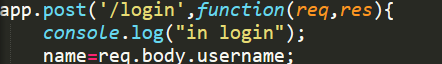






**5.2 Function prototypes**

Login



Delete Package

fn2.PNG

Register

fn3.PNG

View package

fn4.PNG

Book Package

fn5.PNG

View destination by package

fn6.PNG

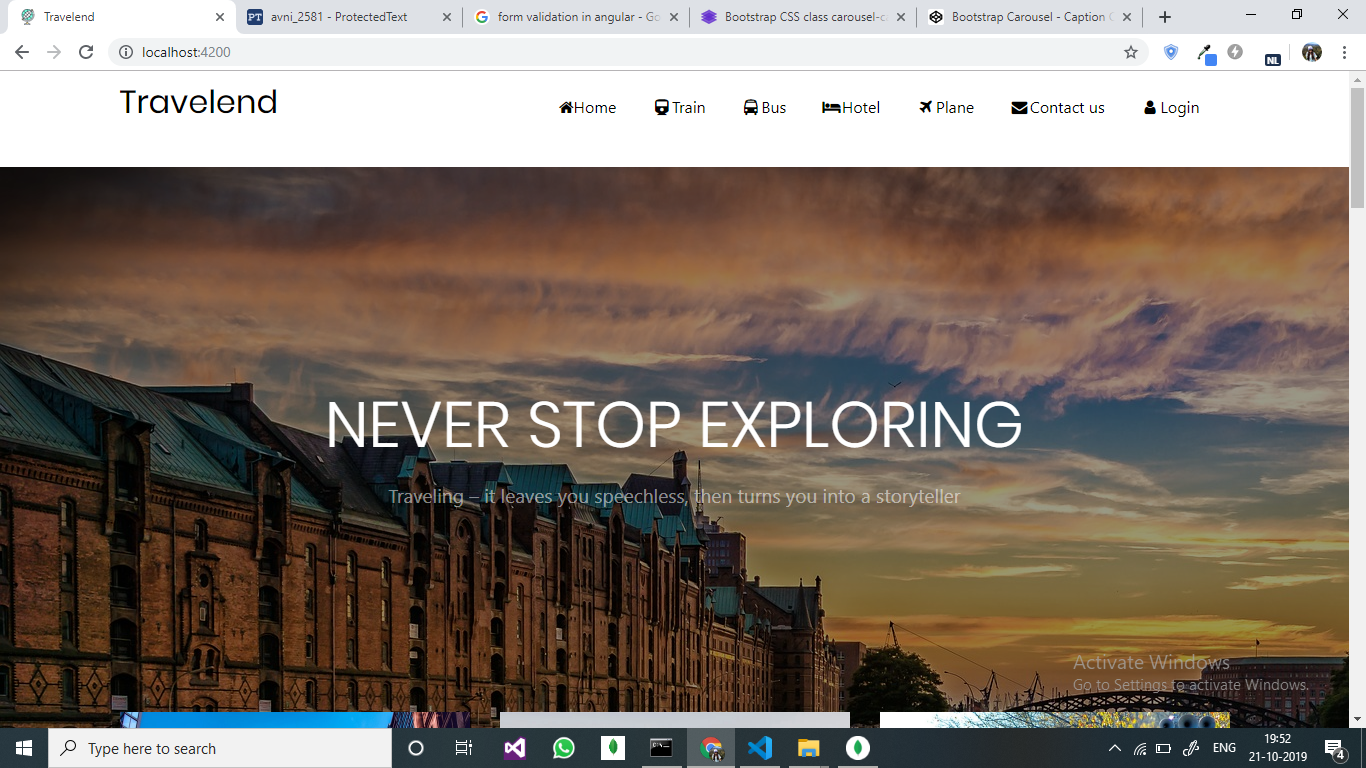
Update package

fn7.PNG

6. Testing

Manual testing was done in order to find and fix the bugs in development process.

7. ScreenShots



8. Conclusion

The functionalities implemented in system after understanding all the system modules according to the requirements. Functionalities That are successfully implemented in the system are:

* User registration containing all the necessary validation on field
* login
* user authentication
* logout
* record storage
* record retrieval
* medical profile storage
* medical profile update
* medications daily list

After the implementation and coding of system comprehensive testing was performed on the system to determine the errors and possible flaws in the system.

9. Limitations and Future Enhancements

The System has adequate scope for modification in future if it is necessary.

The current record storage module can be extended to filter and sort specific records based on parameters. System can also be integrated with Artificial Intelligent agents to generate insights of user’s health.

The current medical profile module can be extended to profile sharing function so that user can share their medical profiles and records with their doctors. Medications modules can be extended so that user can mark the medicines he/she has taken. Also an auto-buy function can also be provided.

10. Reference / Bibliography

Following links and websites were referred during the development of this project.

Google.com

Stackoverflow.com

Github.com

Angular.io

Docs.mongoDb.com

Getbootstrap.com

Codepen.io