BANASTHALI VIDYAPITH

SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT

&

SOFTWARE DESIGN SPECIFICATION DOCUMENT

DIVY DARSHAN

(ONLINE MANDIR DARSHAN WEBSITE)

MCA/MSc. CS M08

Guided By:

Dr. Ashok Kumar

Submitted by:

Priyanshi Pal - 2142514

Laveena Tanwani – 2142879

Khushi Gupta – 2142872

Avantika Pandey - 2142855

TABLE OF CONTENT

1. Introduction

1.1 Purpose 3

1.2 Scope..... 3

1.3 Definitions, Acronyms and Abbreviations..... 4

1.4 Overview.....5.

2. General Description

2.1 Product Perspective.....6

2.1.1 Product Function 6

2.1.2 Hardware Interface.....6

2.1.3 Software Interface.....7

2.1.4 Communication Interface..... 7

2.2 User Characteristics.....7

2.3 General Constraints..... 8

2.4 Technologies used..... 8

3. Specific Requirements

3.1 Functional Requirements..... 9

3.1.1 Validity checks on inputs 9

3.1.2 Sequence of operations 10

3.1.3 Response to abnormal situation.....10

3.2 Non-Functional Requirements 11

3.2.1 AVAILABILITY..... 11

3.2.2 SECURITY..... 11

3.2.3 RELIABILITY..... 11

3.2.4 PORTABILITY..... 11

3.2.5 MAINTAINABILITY..... 11

4. System Architectural

4.1 System level Architecture 12

4.2 Client Server Architecture..... 13

4.3 Software Architecture 14

4.4. Data Flow Diagram 15

4.4. Use Case diagram..... 19

5.Data Design

5.1 Database Tables 22

5.2 Database Description 26

5.3 E-R Diagram..... 31

6. User Interface Design..... 32

7. TYPES OF TESTS (With Implementation) 34

8.References 37

1.INTRODUCTION

1.1 Purpose

The purpose of this document is to describe the software requirement in developing the website of "DIVY DARSHAN" this website provides a user-friendly interface with the users. Due to corona pandemic has dramatically changed the nature of our social interaction so that's why we create a website to provide an online darshan and provide offline darshan as well as those who want to go to the temple with the minimum capacity of the people with the social distancing and precautions .The website is an online darshan website associated with various temple and a NGO who works for underprivileged people.so user can join the prayers and arti of the temples they have registered for.

This Website will be mainly designed for online darshan as well as offline darshan in order to establish communication with the user. It also provides various facilities user can see history of temples, user can book the ticket for visiting the temple through E-pass, user can donate money to NGO or temple or combined according to his/her desire.

1.2 Scope

"DIVY DARSHAN" is web-based service which intend to connect with the users for online as well as offline darshan. It will also help to connect external users to maintain their website. "Divy Darshan" will help a user to:

- Construct a profile within a bounded system.
- Articulate a list of other users and NGO and the temple connected.
- view the list of Temples and NGO.

The scope of the project can be listed as under:

- Admin will be responsible for updating the overall database of the website.

- Users will be able to register in the website this will create the profile of the user.
- Users will be able to login after the registration.
- The user will be able to do online darshan as well as offline darshan according to his/her preference.
- Users will be able to donate to NGO or temples or combined.
- Users will be able to interact or contact with NGO as well as temples.

There are two types of users in the website. User can be Public and Admin. All the users will have different access and rights granted by this website.

1.3 Definitions, Acronyms and Abbreviations:

Title: Divy Darshan

- HTML: Hypertext Markup Language is a markup language used to design static web pages.
- HTTP: Hypertext Transfer Protocol is a transaction-oriented client server protocol between a web browser & a Web Server.
- HTTPS: Secure Hypertext Transfer Protocol is a HTTP over SSL (secure socket layer)
- www: World Wide Web
- TCP/IP: Transmission Control Protocol/Internet Protocol, the suite of communication protocols used to connect hosts on the internet TCP/IP uses several protocols, the two ones being TCP and IP.
- DB: Database
- RAM: Random Access Memory

- HDD: Hard Disk Drive
- IIS: International Information Server

1.4 Overview

users:

- First of all, users will be registered on the website. They will fill the registration form which includes some details like Name, Email, phone-INTEGER, create-password.
- Then the user will login to the website in the login form which include Email and Password.
- Users will see the temple details as well as donate money according to his/her preference. If users want to visit an NGO page, he/she will go to the NGO link.
- Users can access a link for free darshan and for paid arti by paid arti it includes payment.it can also do the E-pass registration and payment.
- Users can contact the temples and Payment.
- users can fill the feedback form of NGO and temples.

Admin:

- Admin will do login/logout.
- Admin can manage temple details. It can also manage links and manage users.
- Adim can manage donations in Danpatra /NGO.
- Admin can manage Query and feedback

2. GENERAL DESCRIPTION

2.1 Product Perspective

The main aim of our website is to provide online darshan and e-passes to the users, along with that we will also provide donations to NGO through users. This website will be reliable and user friendly.

2.1.1 Product Function

1) Users will register/login into the system.

2)Users can visit temple pages, read history of temples or view the location of temples.

3)Users can either book a ticket for paid aarti or buy an e-pass for visiting the temples. They can even get an online darshan of the temple for free after logging in into the system.

4) Users can make payments for e-pass or paid aarti, through which they can either pay in danpatra, or they can share their percentage with the NGO. They can even donate the entire amount to the NGO separately.

5) After the payment the user will get a confirmation ticket or e-pass which will be useful in entering the temple. The user will also receive a receipt after donating money to an NGO.

6) Logout from the system

2.1.2 Hardware Interface

Server side:

- RAM: 512 MB
- HDD: 20 GB or more (Free space excluding data size)
- Processor: 1-2 GHz (P4) or onwards

Client Side:

- RAM: 128 MB
- HDD: 10 GB or more (Free space excluding data size)
- Processor: 450 GHz (P2)

2.1.3 Software Interfaces

Server Side:

- OS: Windows Server 2000 or onwards
- Web Server: IIS 5.0 or onwards with ASP.Net framework

Client Side:

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

2.1.4 Communication Interface

- Clients on the Internet will be using HTTP/HTTPS protocols.
- Clients on Intranet will be using TCP/IP protocols.

2.2 User Characteristics

The users of "DivyDarshan.com" should be computer literate.

Components of this software can be categorized as following:

Users:

They are the devotees who can book their paid aarti, free online darshan and e-passes for visiting temples. They can even donate money in NGO.

They will also receive the confirmation of tickets or e-pass and receipt for donation.

Guest

They are the users who have some limited permissions on our website.

NGO

They will receive the donations made by the users and generate a receipt for the users.

Administrator

These types of users are basically the administrator of the whole website. They can change data at any time without creating any conflict or confusion for the rest of life.

2.3 General Constraints

1. Higher order language functions: The ASP.Net will be used for developing the web-pages with the help of Visual Studio 2019 and for database information MS SQL Server will be used.
2. Criticality of the application: The server applications will be available 24x7.
3. Safety and security: The password and valid email/username are the security issue.
4. External users will not be able to gain full functionality of the website.
5. Any substantial enhancement in the website will require approval of the administrator.

2.4 Technologies to be used

- Back End:

  Database server: MS SQL Server 2016 version

  Server-side scripting: ASP.NET with c# 2019 version

- Front End:

  HTML, CSS, JavaScript

- Web Application Server:

  IIS 5.0 with ASP.Net runtime.

- Other Tools:

  IDE: Visual Studio 2019

- Image Editing: Visual paradigm online Free Editor

  Canva

  MS Word

3 SPECIFIC REQUIREMENTS

3.1 Functional Requirements

3.1.1 Validity checks on input

- Email - Must end with @gmail.com
- Password - Must contain: At least 8 characters
  : Uppercase
  : Lowercase
  : Digits
  : Special characters
- Phone INTEGER - Must be of exactly 10 digits
- Card INTEGER - Must be 16 digits
- CVV INTEGER- Must be 3 digits
- Dates - Must enter a valid and available future date

3.1.2 Sequence of operations

- Users can view the home page and browse the website.
- Users will Register on the website (if not a member) otherwise login to the website using email and password.
- User can further
  - Visit Temple Page
  - Visit NGO Page
  - Make Donation
  - Contact either Temple or NGO Management
  - Write query (if any)
- Temple page allows the user to
  - View information about that temple
  - Access link for free darshan
  - Get registration link for paid arti
  - Register for e-pass
- The NGO page allows the user to view information about different NGOs.
- User can further give feedback

3.1.3 Response to abnormal situations (error handling, overflow, recovery)

A popup message or alert message will be displayed on the screen.

3.2 Non-functional Requirements (Software System Attributes)

3.2.1 AVAILABILITY

The availability of this web-site is up to the Internet connection of the client. Since this is client-server related web-site shall be attainable all the time. User should have an account to enter the system; if user does not have an account then user can only see the information which will be displayed on the homepage of the web-site.

3.2.2 SECURITY

The authorization mechanism of the system will block the unwanted attempts to the server and also let the system decide on which privileges may the user have. The system has different types of users so, there are different levels of authorization.

3.2.3 RELIABILITY

A backup file is maintained so that in case of system crash, the data will not be affected.

3.2.4 PORTABILITY

The system is developed using ASP.Net which provides a framework for developing web-based applications.

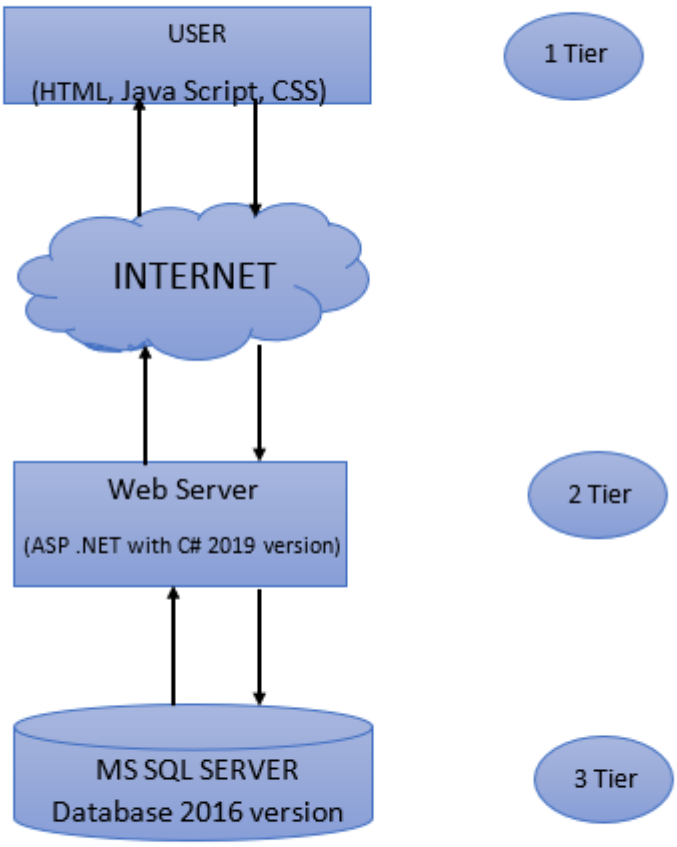
3.2.5 MAINTAINABILITY

This website will follow the modular structure so it will be easy to maintain.

4 SYSTEM ARCHITECTURE

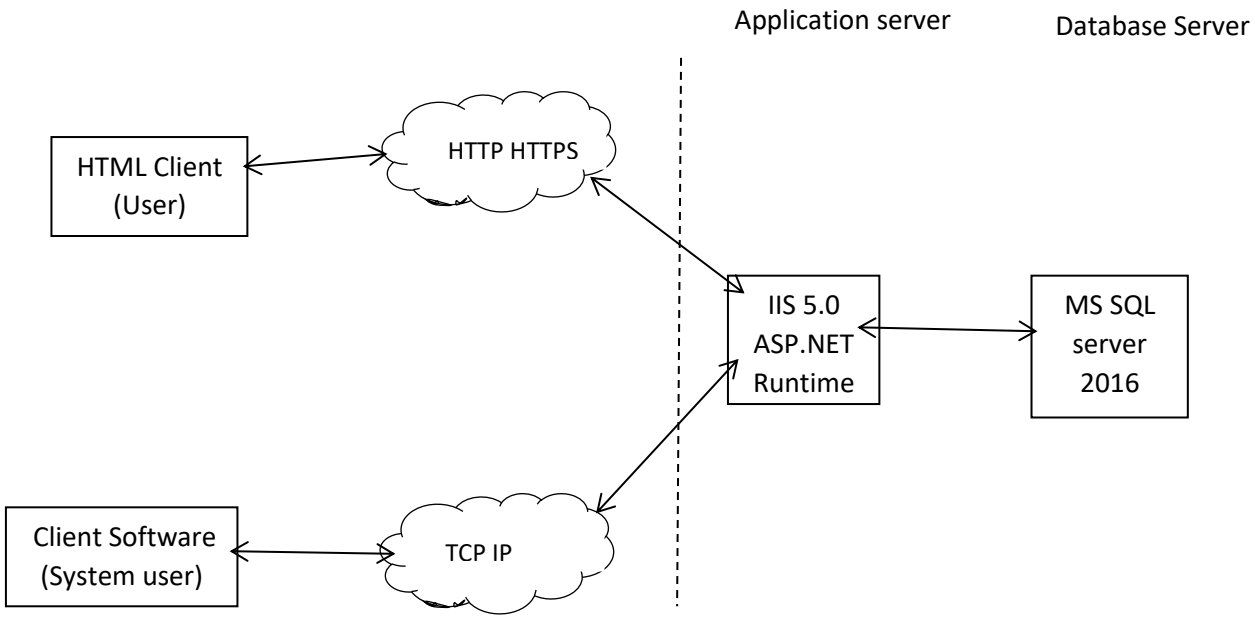
4.1 System Level Architecture

A system architecture is a conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system



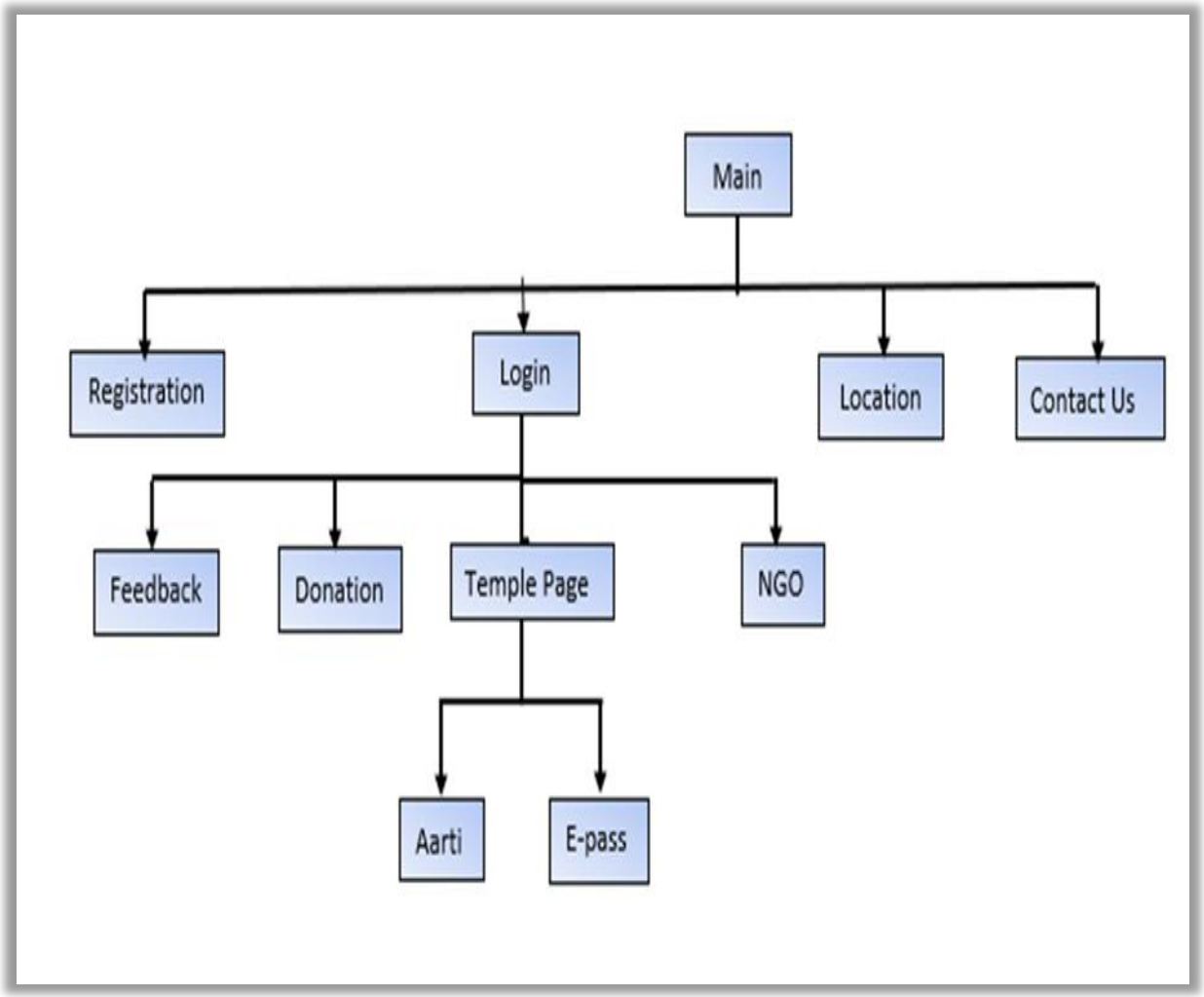
4.2 Client Server Architecture

MS SQL uses a Client/ Server architecture. This architecture has a server side and a client side. Server side includes Application server (IIS 5.0 ASP.NET Runtime) and Database server (MS SQL server 2019). They control the access, fetching, presenting and operations performed on the data. Client-side applications are the programs that communicate with the server. Client and Server establishes Communication protocols (TCP/IP) to interact with each other.



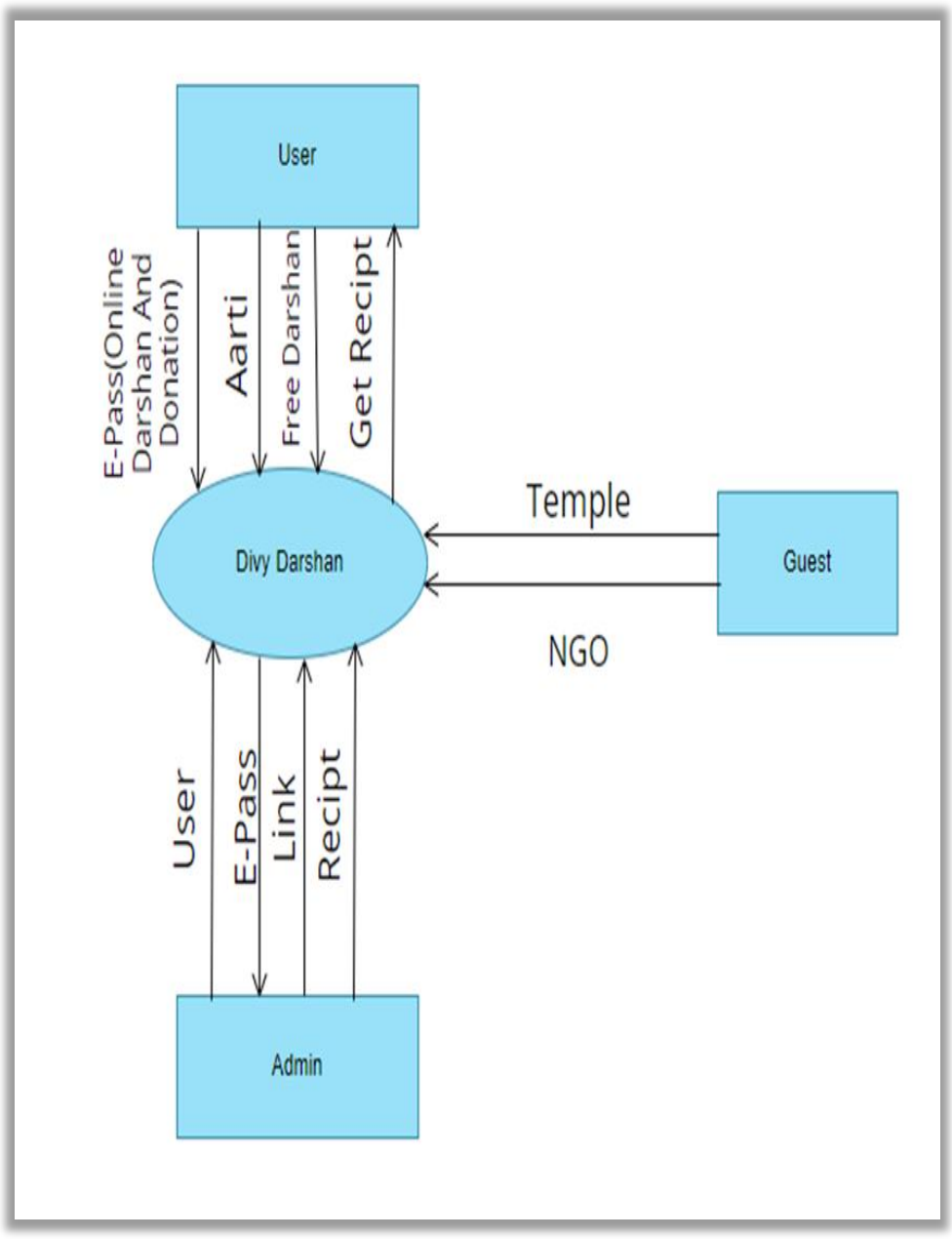
4.3 Software Architecture

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.



4.4 Data flow diagram

0 Level DFD



It's a basic overview of the whole system or process being analysed or modelled. In the above DFD we can see:

1. The admin can:

   a. Generate the link for online paid darshan as well as free darshan.

   b. Can generate the receipt for the payments done by the users.

   c. Receive the details of the users through E-Pass

   d. Add new users in the system.

2) The users can:

   a. Book their slots for visiting the temples through E-Pass.

   b. Book tickets for online paid darshan.

   c. They can also do online darshan for free by registering themselves on our website.

   d. The users can get the receipt for the payments they have made for E-pass, donations, Arti etc.

3) The guest can:

   View temples and NGOs without registering themselves on the website.

1 - Level DFD

It depicts basic modules in the system and the flow of data among various modules. In the above DFD we can see that:

1. The admin can:

   a. Login into the system

   b. Logout from the system.

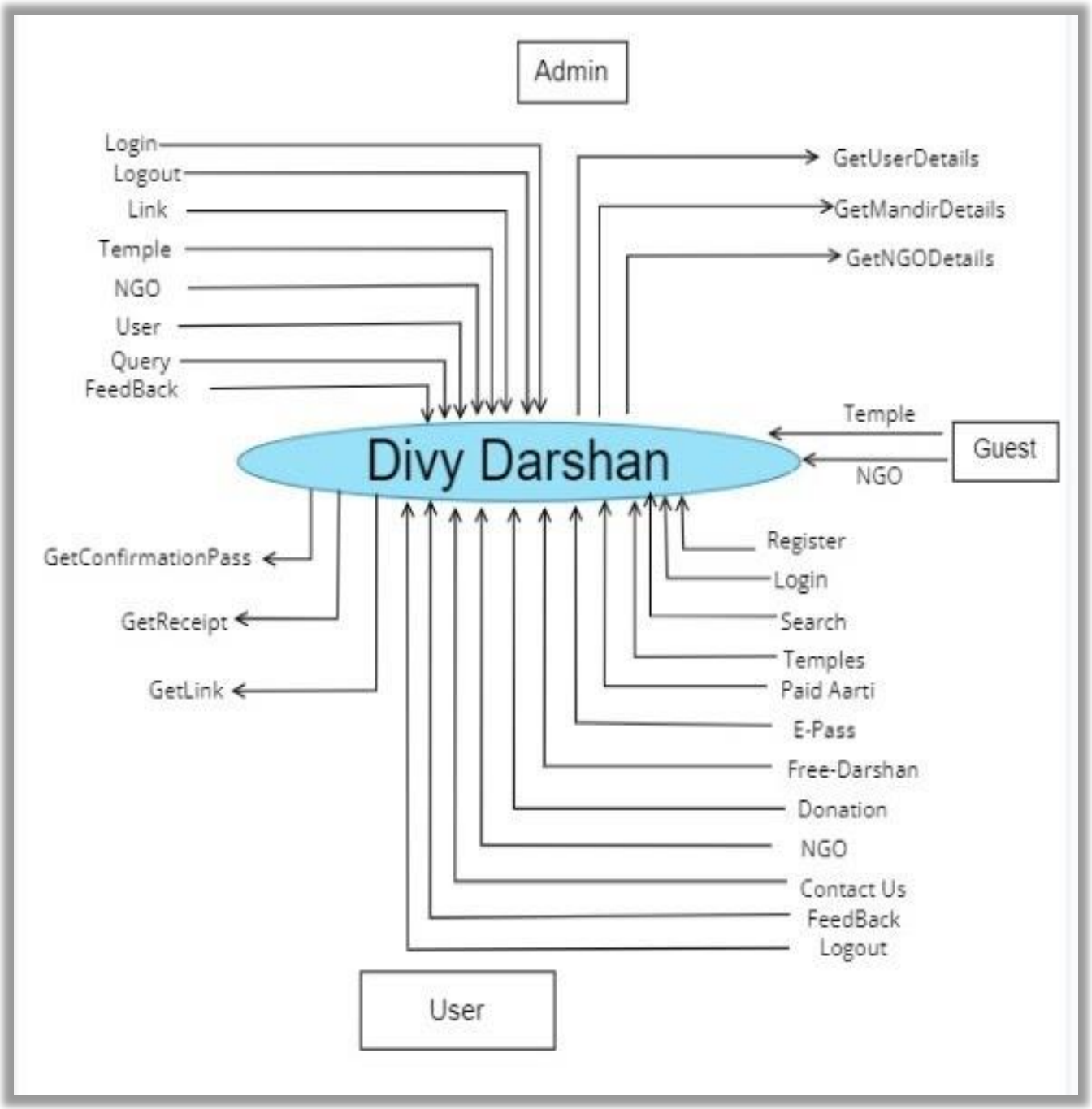
   c. Can generate links for online paid as well as free darshan.

   d. Can manage the temples, NGOs and users associated with our system.

e. Can view queries of the users.

f. Can manage feedback (any violence, or threatened comments are managed by admin)

g. Can get the details of the temples, users and NGOs who register themselves on our website.



2) The users can:

a. They can register into the system.

b. They can login into the system.

c. They can search for temples and NGOs on our website.

d. They can also view the location of temples and NGOs associated with our website.

e. They can book tickets for online paid darshan.

f. They can book e-passes for visiting the temples.

g. They can make donations to NGOs as well as temples.

h. They can contact our team via email or call.

i. They can give their feedback, or ask queries if they are facing any problems, they can also give suggestions to us so that we improve our website.

j. They can Logout from our system, and once they are registered, they don't need to register once again.

k. They can get the confirmation for the online paid darshan.

l. They can receive the receipt of donations or e-passes.

m. They can also get links for online darshan (paid as well as free darshan).

3) The guests can:

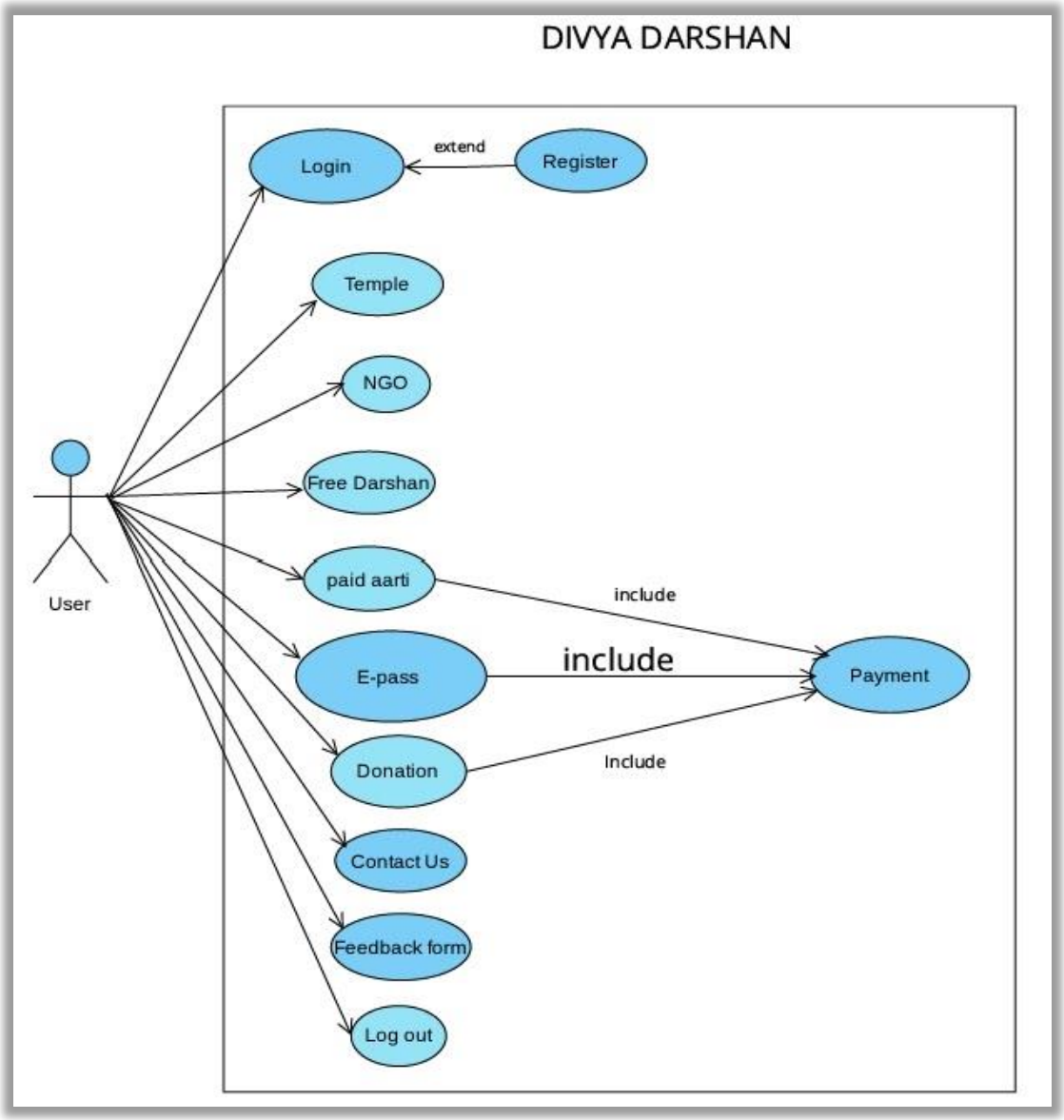
View temples and NGOs without registering themselves on our website

4.5 Use case diagram

The diagram is a graphic depiction of the interaction among the elements of Divy Darshan (online darshan system). The main actors of Divya Darshan System in use case diagram are: Admin and User.

The relationship between among the actors and the use cases of Divy Darshan (Online Darshan System) are:

- User Entity: Use cases of User entity are Register, Login, Temple, NGO, Free Darshan, Paid Arti, E-Pass, Donations, Contact us, Feedback, Logout.
- Admin Entity: Use cases of Admin entity are Login, Temple, NGO, Link, User, Query, Feedback.

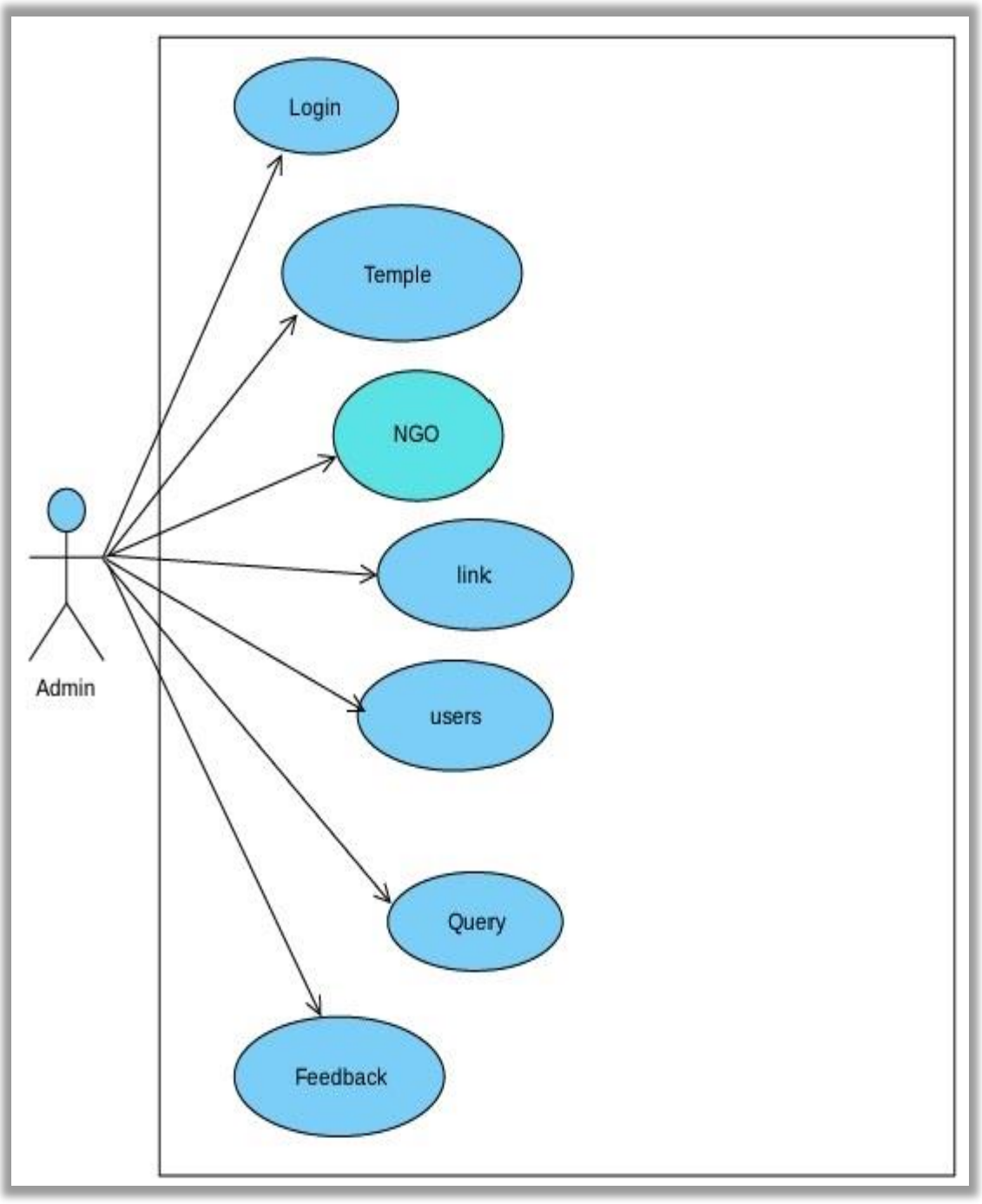


The functional requirement giving by various use cases for user:

- Register: Users can register on the website and access the benefits like free darshan, paid Arti and E-pass.
- Login: If user already have an account on the website, it has to login with email and corresponding password.
- Temple: Users can view details of various temples collaborated with the website.
- NGO: Users can see the details of various NGOs on the website.
- Free Darshan: A registered user can access free darshan of any temple on the website by accessing a link.
- Paid Arti: User can view live arti of any temple for which he/she has registered. Payment will be necessary.
- E-pass: User can get an E-pass for visiting the temples.
- Donation: Users can make donations for temples or NGOs.
- Contact Us: Users can contact Admin/Temple management/NGO management for any information.
- Feedback: Users can give feedback regarding their experience.
- Logout: Users can log out from the website.

The functional requirement giving by various use cases for admin:

- Login: Admin can login to manage the website.
- Temple: Admin can manage the temple details such as locations, scheduling hours.
- NGO: Admin can manage the NGO details.
- Link: Admin can manage links of temples for online darshan, paid aarti and E-passes.
- Users: Admin can manage users' details who access the website.
- Query: Admin can manage the queries which are asked by the users
- Feedback: Admin can manage the feedback which is given by the users.


Login
Temple
NGO
link
users
Query
Feedback
Admin

5. DATA DESIGN

5.1 DATABASE TABLES:

TABLE 1: USERS

Field	Type	Description	Constraint
User_id	VARCHAR (20)	Unique ID of Registered User	Primary Key
Firstname	VARCHAR (20)	Firstname of user	Not Null
Lastname	VARCHAR (30)	Lastname of user	Not Null
Email_ID	VARCHAR (30)	Email ID of user	Unique
Password	VARCHAR (8)	Password of user	Not Null
Contact_no	VARCHAR (10)	Contact no. of user	Not Null
User_type	VARCHAR (5)	User is admin/user	Not Null

TABLE 2: TEMPLES

Field	Type	Description	Constraint
Temple_id	VARCHAR (30)	Unique ID of Temple	Primary Key
Name	VARCHAR (50)	Name of Temple	Not Null
Location	VARCHAR (40)	Location of Temple	Not Null
Email_ID	VARCHAR (30)	Email ID of Temple	Not Null, unique
Contact_no	VARCHAR (10)	Contact no. of Temple	Null
About	VARCHAR (4)	About Temple	Not Null
Image	BLOB (300Mb)	Image of Temple	Not Null
Open_time	TIME	Time when the temple gets open	Null
Close_time	TIME	Time when the temple gets close	Null

Darshan_link	VARCHAR (30)	A link for free darshan provided to user, enables only while temple opens	Unique, Not Null

TABLE 3: AARTI

Field	**Type**	**Description**	**Constraint**
Aarti_id	VARCHAR (30)	Unique ID of each aarti held in temple	Primary Key
Temple_id	VARCHAR (30)	Unique ID of temple	Foreign Key
Time	TIME	Time when the aarti occurs	Not Null, unique

TABLE 4: AARTI BOOKING

Field	**Type**	**Description**	**Constraint**
Ticket_id	VARCHAR (20)	Unique ID of Aarti Booking	Primary Key
User_id	VARCHAR (30)	Unique ID of user	Foreign Key
Aarti_id	VARCHAR (30)	Unique ID of aarti	Foreign Key
Date	DATE	Date when user wants to join aarti	Not Null
Time	TIME	Time when user wants to join aarti	Not Null
Amount	INTEGER (4)	Rs 200 for a booking	Default
Token	INTEGER (4)	Random INTEGER as token provided to user to join aarti	Unique

TABLE 5: E-PASS

Field	Type	Description	Constraint
Ticket_id	VARCHAR (30)	Unique ID of Epass ticket	Primary Key
User_id	VARCHAR (30)	Unique ID of user	Foreign Key
Temple_id	VARCHAR (30)	Unique ID of Temple	Foreign Key
Name	VARCHAR (30)	Name of person applying for epass	Not Null
Date	DATE	Date when user wants to join aarti	Not Null
No. of People	INTEGER (4)	Maximum 5 people can come from 1 EPass	Not Null
Amount	INTEGER (4)	Rs 500 for an epass	Not Null
Token	INTEGER (4)	Random INTEGER as token provided to user for epass	Not Null

TABLE 6: TRANSACTION

Field	Type	Description	Constraint
Trans_id	VARCHAR (10)	Unique ID for Transaction	Primary Key
Ticket_id	VARCHAR (10)	Unique ID of Aarti Booking/ Epass ticket	Foreign Key
Card_no	VARCHAR (16)	Card No. of card for payment	Not Null
cvv	INTEGER (4)	cvv of card for payment	Not Null
Cardholder_name	VARCHAR (30)	Name of cardholder	Not Null
Expiry_date	DATE	Expiry date of card	Not Null

TABLE 7: NGO

Field	**Type**	**Description**	**Constraint**
NGO_id	VARCHAR (30)	Unique ID of NGO	Primary Key
Name	VARCHAR (30)	Name of NGO	Not Null
Location	VARCHAR (30)	Location of NGO	Not Null
Email_ID	VARCHAR (30)	Email ID of NGO	Not Null
Contact_no	VARCHAR (30)	Contact No. of NGO	Not Null
About	VARCHAR (4)	About NGO	Not Null
Image	BLOB (300 Kb)	Image of NGO	Not Null

TABLE 10: CONTACT US

Field	**Type**	**Description**	**Constraint**
Query_id	VARCHAR (10)	Unique ID of Query	Primary Key
User_id	VARCHAR (30)	Unique ID of user	Foreign Key
Temple/NGO_id	VARCHAR (30)	Unique ID of Temple/NGO	Foreign Key
Message	VARCHAR (65535)	Messge/ Query given by user	Not Null

TABLE 8: DONATION

Field	**Type**	**Description**	**Constraint**
Trans_id	VARCHAR (30)	Unique ID of Transaction	Primary Key
User_id	VARCHAR (30)	Unique ID of user	Foreign Key
Temple/NGO_id	VARCHAR (30)	Unique ID of Temple/NGO	Foreign Key
Amount	INTEGER (4)	Amount user wants to donate	Not Null, check (<=50,000)
Show_name	TINYINT	Will be taken from checkbox and return 0 or 1	Default
Card_no	VARCHAR (16)	Card No. of card for payment	Not Null
Cvv	INTEGER (4)	cvv of card for payment	Not Null
Cardholder_name	VARCHAR (40)	Name of cardholder	Not Null
Expiry_date	DATE	Expiry date of card	Not Null

TABLE 9: FEEDBACK

Field	**Type**	**Description**	**Constraint**
Feedback_id	VARCHAR (10)	Unique ID of Feedback	Primary Key
User_id	VARCHAR (30)	Unique ID of user	Foreign Key
Temple/NGO_id	VARCHAR (30)	Unique ID of Temple /NGO	Foreign Key
Message	VARCHAR (65535)	Feedback given by user	Not Null

5.2 Database Description (ER model)

The Entity Relationship Diagram (ERD) for the 'Online Mandir Darshan System' is presented above. The purpose of the ERD is to illustrate the basic data abstraction for the system. Each entity in the above diagram is described as a Rectangular Box. Within the implementation, an instance of an entity will represent a line in a table corresponding to that entity. For example, User's entity corresponds to the list of users in the database.

An instance of an entity composed of many data items. These data items are fields or columns in the table of the entity. Each data item represents an *attribute* of the entity. Each entity contains one or more attributes which uniquely identifies an instance of the entity, these attributes are called the *key attributes* or *Primary key*. The Primary key can be used to retrieve or delete information from the entity table.

- Users Entity:

The Users entity contains the following data items:

- User_id
- Firstname
- Lastname
- Password
- Email ID
- Contact_no
- User_type

The primary key of the Users entity is the User_id.

- Temple Entity:

The Temple entity contains the following data items:

- Temple_id
- Name
- Location
- Image
- About
- Email_ID

- Contact_no
- Open time
- Close time
- Darshan link

The primary key of the Login entity is the Temple_id.

- Aarti Entity:

The Aarti entity contains the following data items:

- Aarti_id
- Temple_id
- Time

The primary key of the Aarti entity is the Aarti_id.

- Aarti Booking Entity:

The Aarti Registration entity contains the following data items:

- Ticket_id
- Aarti_id
- User_id
- Date
- Time
- Amount
- Token

The primary key of the Aarti Booking entity is the Ticket_id.

- E-pass Entity:

The E-pass entity contains the following data items:

- Ticket_id
- User_id
- Temple_id
- Name
- Date
- No. of people
- Amount

The primary key of the E-pass entity is the Ticket_id.

- Transaction Entity:

The Transaction entity contains the following data items:

- Trans_id
- Ticket_id
- Card no.
- Cardholder name
- cvv
- expiry date

The primary key of the E-pass entity is the Trans_id.

- NGO Entity:

The NGO entity contains the following data items:

- NGO_id
- Name
- Location
- About
- Image
- Email_ID
- Contact_no

The primary key of the E-pass entity is the NGO_id.

- Donation Entity:

The Donation entity contains the following data items:

- Trans_id
- User_id
- Temple/NGO_id
- Show_name/anonymous
- Amount
- Card_no
- cvv
- Cardholder name

The primary key of the Donation entity is the Trans_id.

- Contact us Entity:

The Contact us entity contains the following data items:

- Query_id
- User_id
- Temple/NGO_id
- Message

The primary key of the Contact us entity is the Query_id.

- Feedback Entity:

The Feedback entity contains the following data items:

- Feedback_id
- User_id
- Temple/NGO_id
- Message

The primary key of the Feedback entity is the Feedback_id.

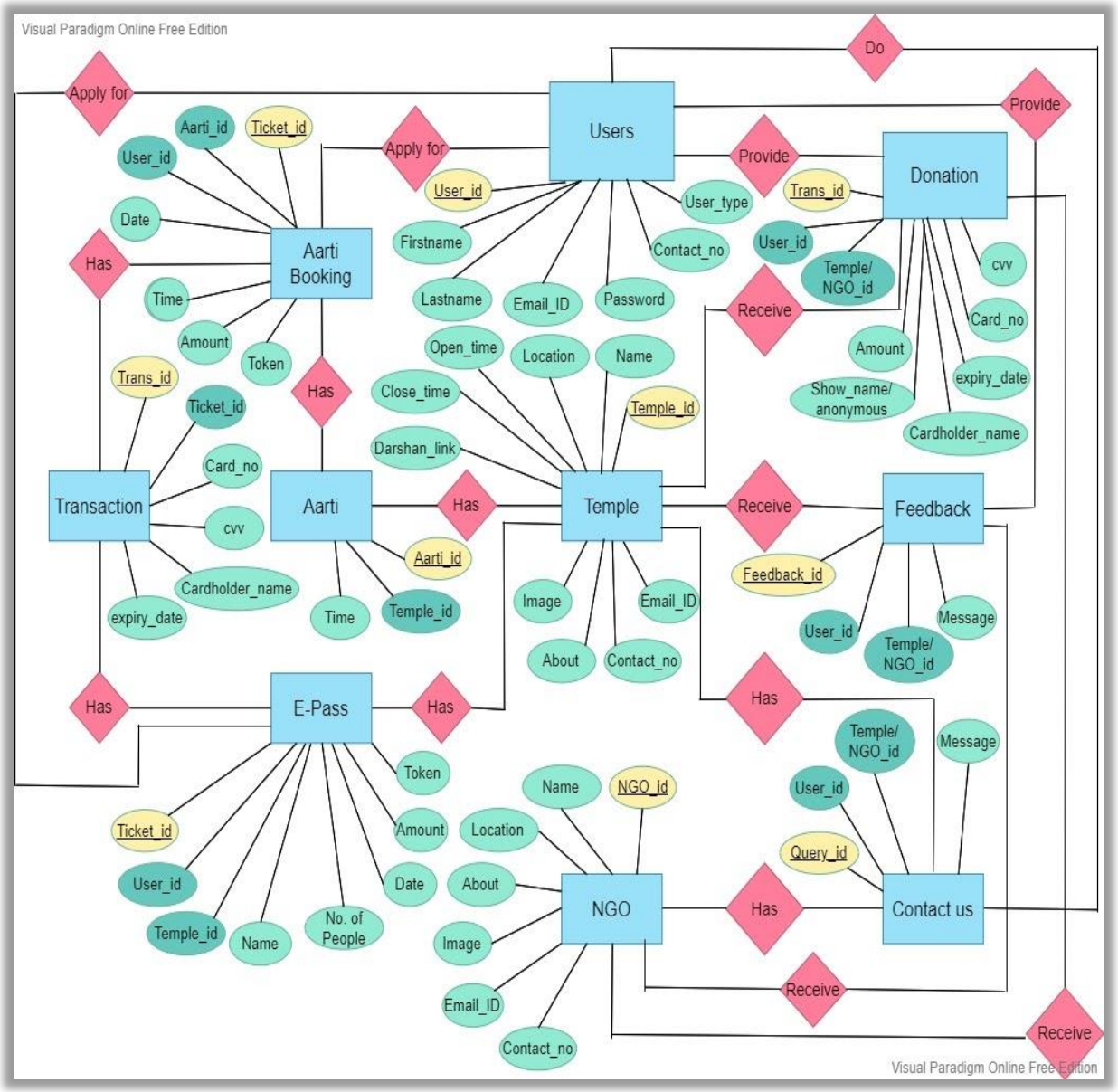
Relationships

For the 'Online Mandir Darshan System', there are following relationships between the objects. The relationships are:

1. A user can apply for online aarti booking or e-pass.
2. A user can provide feedback.
3. A user can provide donation.
4. A user can do contact for query.
5. A temple has online aarti facility.
6. An aarti has an aarti booking.
7. A temple has e-pass facility.
8. An aarti booking and a e-pass has payment transaction.
9. A temple can receive donation.
10. A temple can receive feedback.
11. A temple has contact us for query.
12. An NGO can receive donation.
13. An NGO can receive feedback.
14. An NGO has contact us for query.

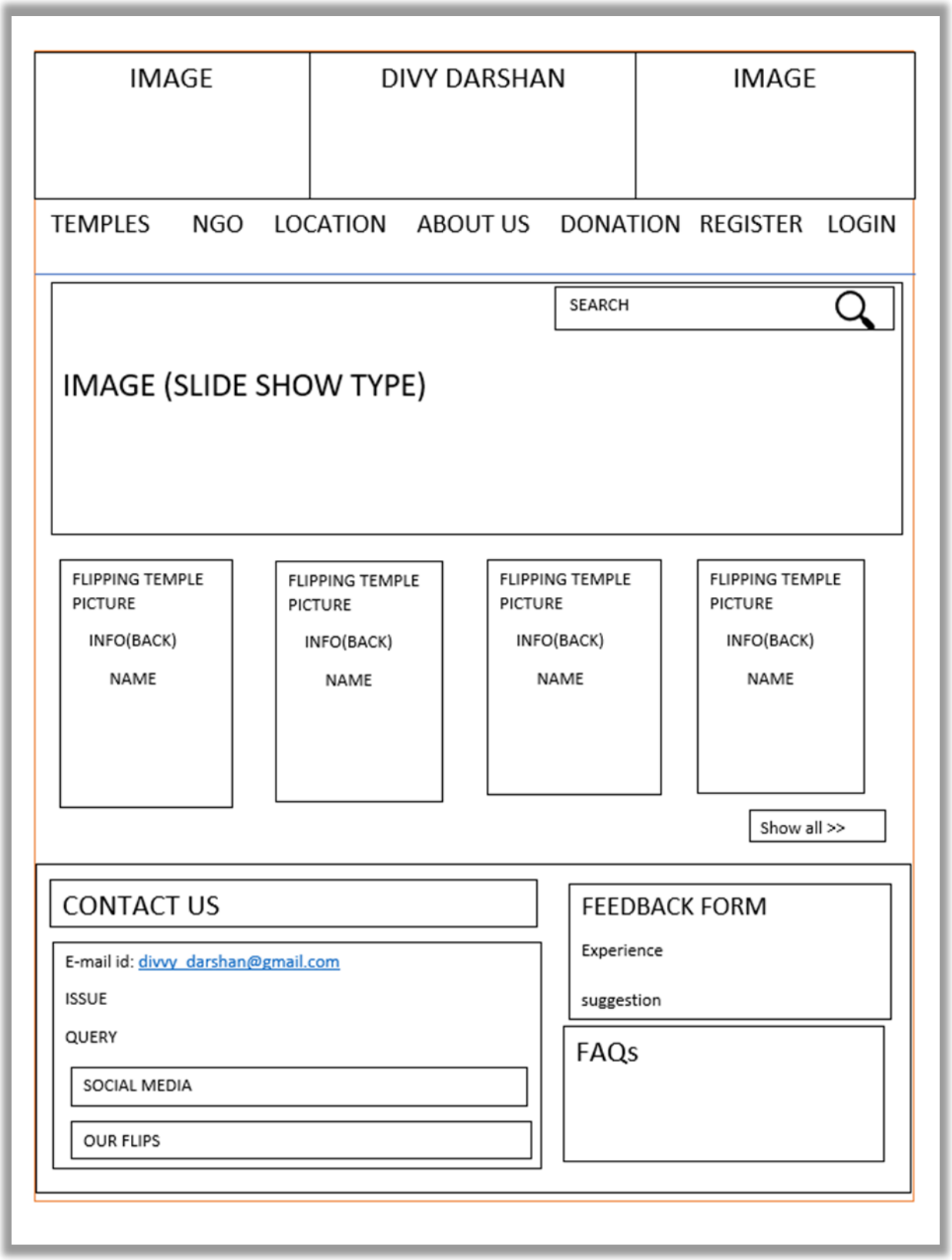
5.3 ER Diagram

In the ERD above, relationships are described in the diamonds. Each diamond corresponds to a table that relates some INTEGER of instances of one object to some INTEGER of instances of the other.



6 USER INTERFACE DESIGN (PROTOTYPE)

Home Page:



Temple Page:

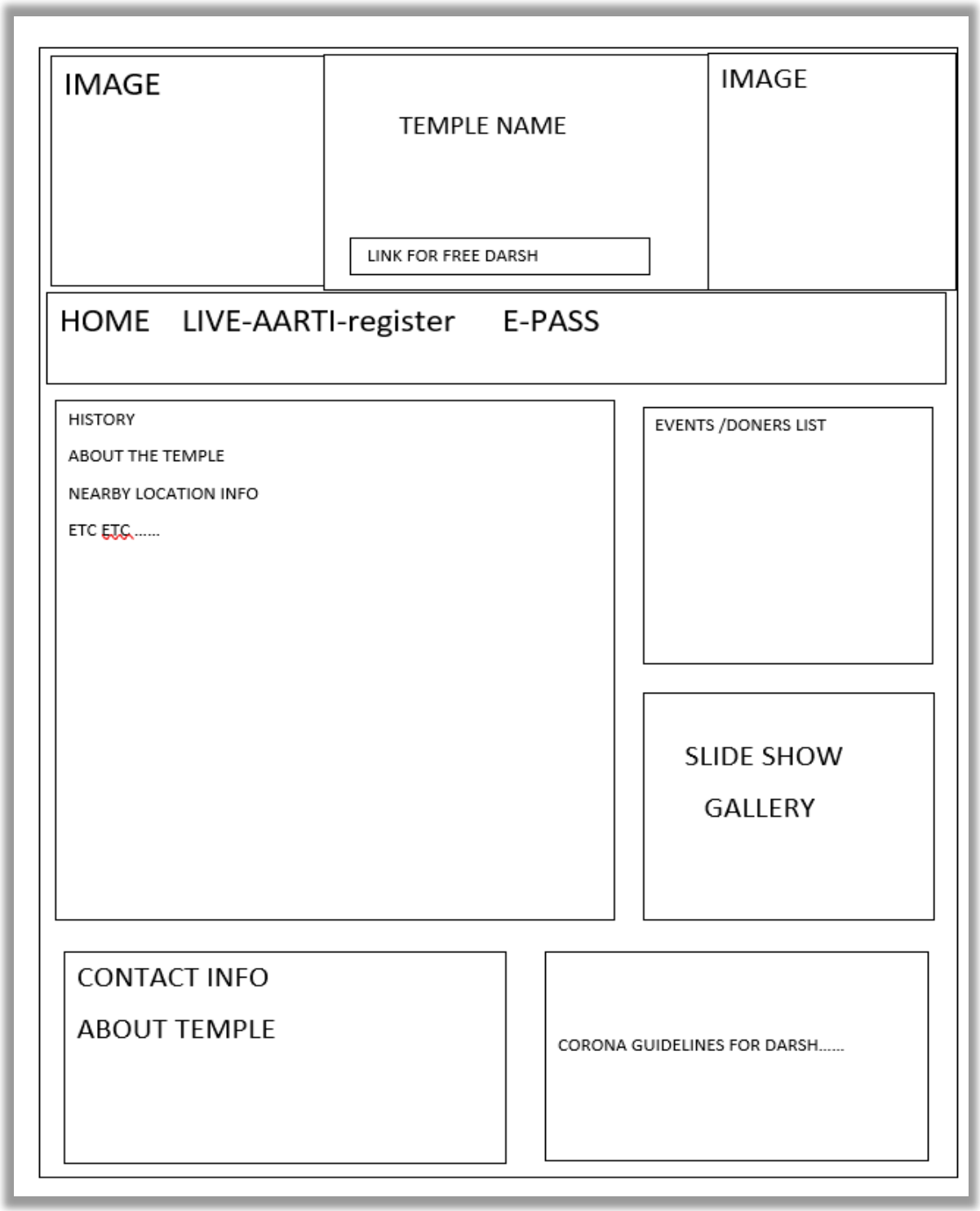


IMAGE
TEMPLE NAME
LINK FOR FREE DARSH
IMAGE
HOME LIVE-AARTI-register E-PASS
HISTORY
ABOUT THE TEMPLE
NEARBY LOCATION INFO
ETC ETC.....
EVENTS /DONERS LIST
SLIDE SHOW
GALLERY
CONTACT INFO
ABOUT TEMPLE
CORONA GUIDELINES FOR DARSH.....

7. TYPES OF TESTS

Unit Tests

Unit tests are most commonly done by developers on their own machines or on a common server that is very volatile. It is not necessary that the unit test machines be the same platform and operating system as the target deployment environment, but the movement from the unit test environment to other testing environments should not require material code changes by developers. A plan for one machine per developer plus one small server should be included in the overall system architecture.

System Tests

The system test environment allows multiple modules to be connected together and executed as in a typical use-case scenario. The choice as to whether this is done on a separate machine from unit testing is up to the implementation and test team. If the target deployment environment is different from the unit test environment, the system test environment should contain a machine that matches the target environment. Although the system test machine need not match the size of the deployment box, it should have the same platform and operating system. A good rule of thumb is to prepare to add one more box for system tests of a smaller size, but the same operating system as the target environment. Again, this will be a relatively volatile environment, so it should not be viewed as a place to do industrial-strength testing by a large team.

Regression Tests

To perform integration and regression tests, it is advisable to have a separate environment that is similar to the target environment.

Generally, one server will be enough at this point. However, the contents of this server should be strictly controlled. Either the test coordinator or his or her designate should make all software changes to this environment. Stability and auditability are essential to ensuring the accuracy of test results. Plan for at least one more server at this stage in testing.

Stress Tests

Stress tests should be done in an environment identical to the target hosting environment. In the first development cycle, this can be done in the production site, before the cutover to production. For subsequent development cycles, a separate environment will have to be maintained for stress testing. Plan to replicate the deployment environment as part of the test bed for at least the second development cycle of the site. It has also been observed that the most common problem after performance in a high stress environment is database deadlock due to improper programming. Deadlocks are typically difficult to detect and fix and may not show up until the site is highly stressed in production. So it is important that these conditions be stringently tested during the stress test phase.

Acceptance Tests and Staging

Acceptance tests are generally performed in the same environment as the stress tests, so additional hardware is not needed to support this phase of testing. Again, during the initial development cycle, the production environment can be used to perform both acceptance testing and staging, but a new environment should be created for subsequent development cycles.

8 REFERENCES

- Geeksforgeeks.org
- Toutorialspoint.com
- Ehindistudy.com
- Codebun.com
- Slideshare.net
- Perforce.com
- Toptol.com
- Youtu.be
- Guru99.com