Messaging Platform for Promoting Healthy Lifestyle Behaviours

Unit SIT764 Project Analysis and Design

Project Solution Document

Messaging Platform for Promoting Healthy Lifestyle Behaviours

Table of Contents:

Introduction

This mainly consist three elements. One Database, Website and mobile android app. As per the proposal document and recommendation that is provided we have come up with the detail technical information to how we can achieve the task.

Database

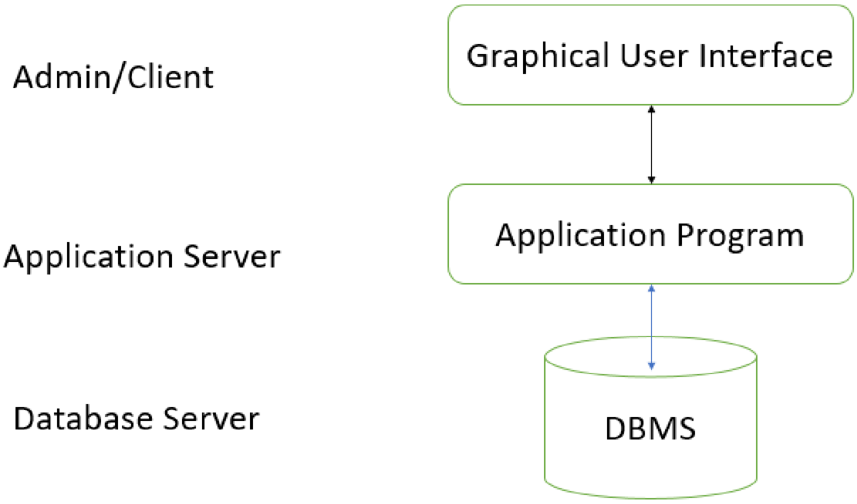


Figure 1.Database Management System

SQL database will be used to store participant's personal information which will be organized into rows, columns and tables, and it will be indexed to quickly search for relevant information regarding any participant. Database will interact with the web application and the database itself to capture and examine the data.

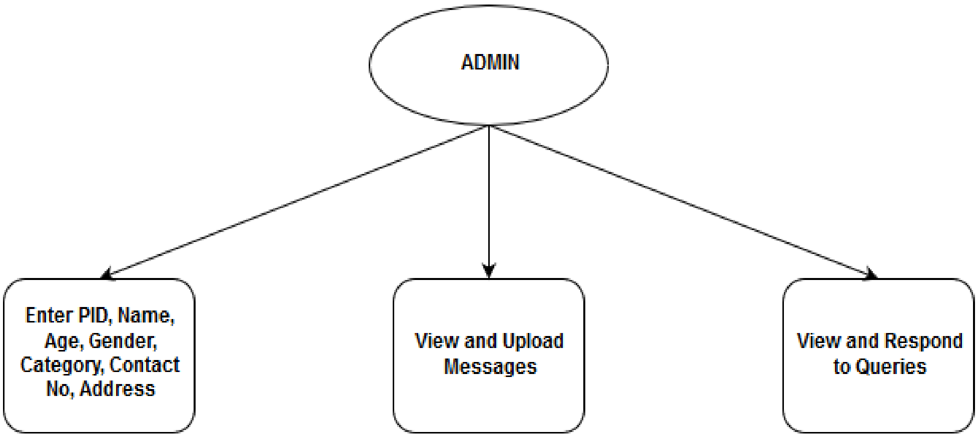

ADMIN
Enter PID, Name, Age, Gender, Category, Contact No, Address
View and Upload Messages
View and Respond to Queries

Figure 2. Admin Panel

Our project will have a table for administration and normal user login with the following field:

USERNAME	**PASSWORD**
Admin	Admin
User	User

Table where admin will be updating participant's details:

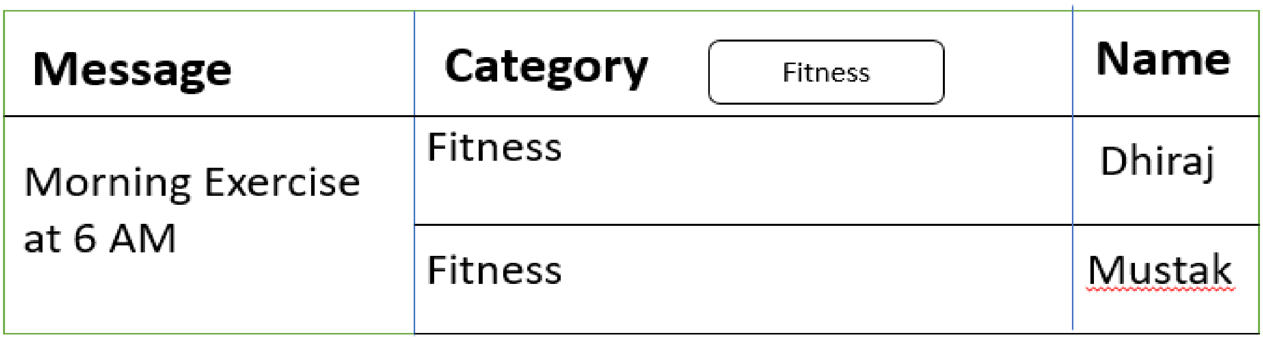
PID (primary key)	**NAME**	**AGE**	**GENDER**	**CATEGORY**	**CONTACT No.** (secondary key)	**ADDRESS**
1	Dhiraj	25	Male	Diabetic	**0426739944**	18/42-46 Burnt Street

Participant ID is the primary key for participants table and it is unique for each record in the database. The use of primary key is to implement relationship between 2 tables. Once u create a primary key u can create foreign key in the other table to relate to participant table using the primary key. Every table can have only 1 primary key.

Contact Number is the secondary key for participants table and a table can have more than 1 secondary key. Secondary key is basically used for faster searches when the database grows huge.

SQL query for creating a table with primary key:

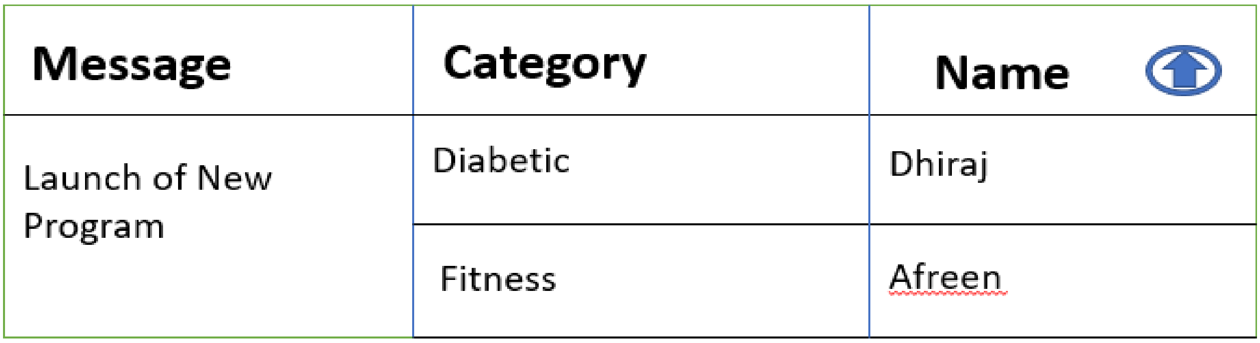
```
CREATE TABLE Participant (
 Participant ID int NOT NULL PRIMARY KEY,
 Name varchar(255),
 Age int,
Gender varchar(255),
Category varchar(255)'
Contact no int,
Address varchar(255)
);
```



Message	Category Fitness	Name
Morning Exercise at 6 AM	Fitness	Dhiraj
	Fitness	Mustak

Figure 3. Message to a group (Fitness) from Admin

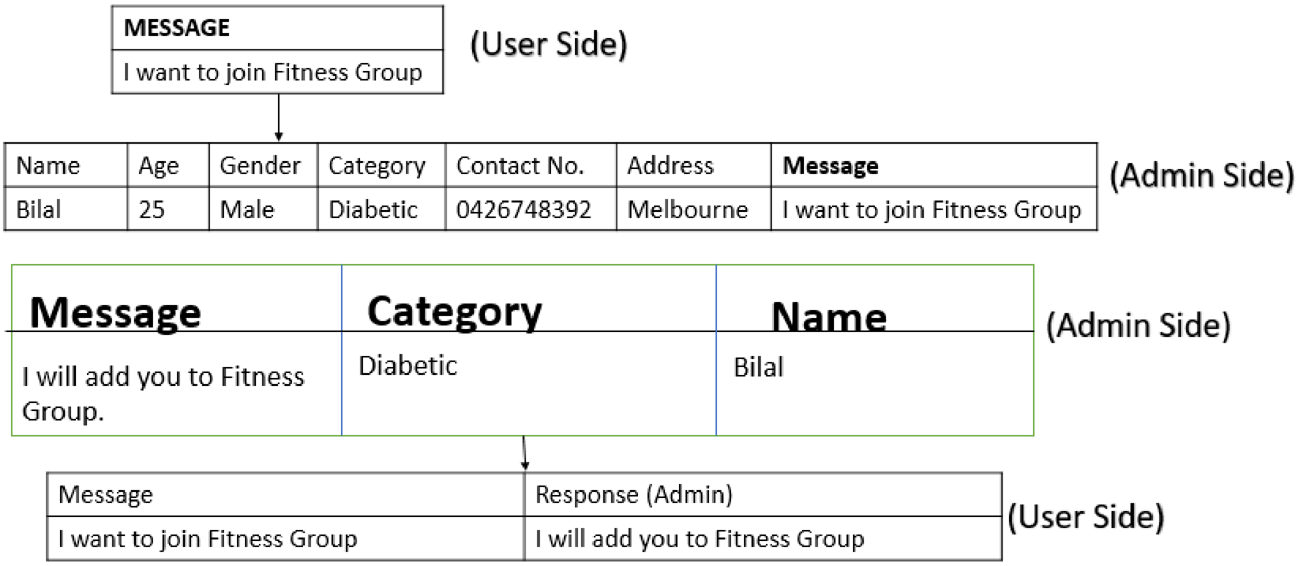
Admin will have privilege to send message to group and or an individual.
Through the use of “Category” Admin can select certain group to send selective message.
We will be using command such as: Select “Fitness” from Category to select all people who are put on Fitness group, and then the Admin can send Message to this group.



Message	Category	Name ⬆
Launch of New Program	Diabetic	Dhiraj
	Fitness	Afreen

Figure 4. Message from Admin to Everyone

Admin will also be able to select “All” from the table Name.
After selecting “All” from Name table, now the admin can forward a message which will be delivered to every participant.
We will be using query like: SELECT * FROM NAME, which will select every person under that table.
Also, Admin can select individuals from the table Name if he wants to send a personal message.



MESSAGE
I want to join Fitness Group

(User Side)

Name	Age	Gender	Category	Contact No.	Address	**Message**
Bilal	25	Male	Diabetic	0426748392	Melbourne	I want to join Fitness Group

(Admin Side)

Message	Category	Name
I will add you to Fitness Group.	Diabetic	Bilal

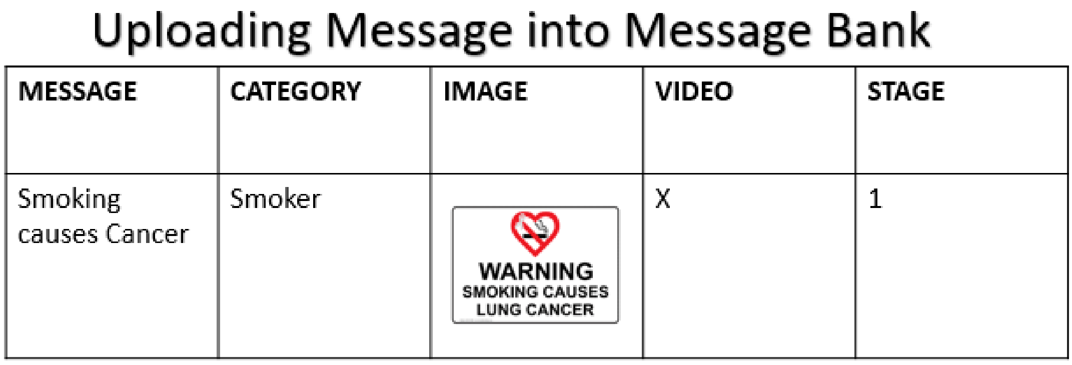
(Admin Side)

Message	Response (Admin)
I want to join Fitness Group	I will add you to Fitness Group

(User Side)

Figure 5. Message from participant to Admin and a response from admin to participant

User can also send private message to Admin as a query.
Admin will receive notification about a new message as shown in the figure above.
We will be using links to personalize message with participants.



Uploading Message into Message Bank

MESSAGE	CATEGORY	IMAGE	VIDEO	STAGE
Smoking causes Cancer	Smoker	WARNING SMOKING CAUSES LUNG CANCER	X	1

Fig 8: Message input into Message Bank

Figure 6. Message Input into message bank

Admin will have access over Message Bank where he can view and add new data. Message Bank will have different tables like Message, Category, Image, Video, Stage, etc.
We have added “Stage” table which will help admin to select different message to send at different time. This helps to send new message every time and prevents user from seeing same message every time.
We will demonstrate this algorithm in next stage.

CATEGORY	MESSAGE	IMAGE	VIDEO	STAGE
Smoker	Smoking causes Cancer		X	1
Diabetic	Consume less sugar	X	X	1
Smoker	Smoking increases chance for stroke	X	X	2
Fitness	Wake up early in the morning.	X	X	1

Figure 7. Sample of message bank

CATEGORY Smoker	MESSAGE	IMAGE	VIDEO	STAGE
Smoker	Smoking causes Cancer		X	1
Smoker	Smoking increases chance for stroke	X	X	2
Smoker	Lungs work properly after quitting smoke	X	X	3
Smoker	Our taste sense work good after 2 days of quitting smoke	X	X	4

Figure 8. Message bank view after putting "smoker" in the CATEGORY table

If admin wants to send a new message to participants in a certain category or everyone, he can create a new message and select any category as needed and send the message as shown in Fig: 5 and Fig: 6.

If the admin wants to forward a message from the MESSAGE BANK to certain category or an individual he will type the CATEGORY NAME, where he will find all message for that category.

The next step will be to choose a message by selecting the row.

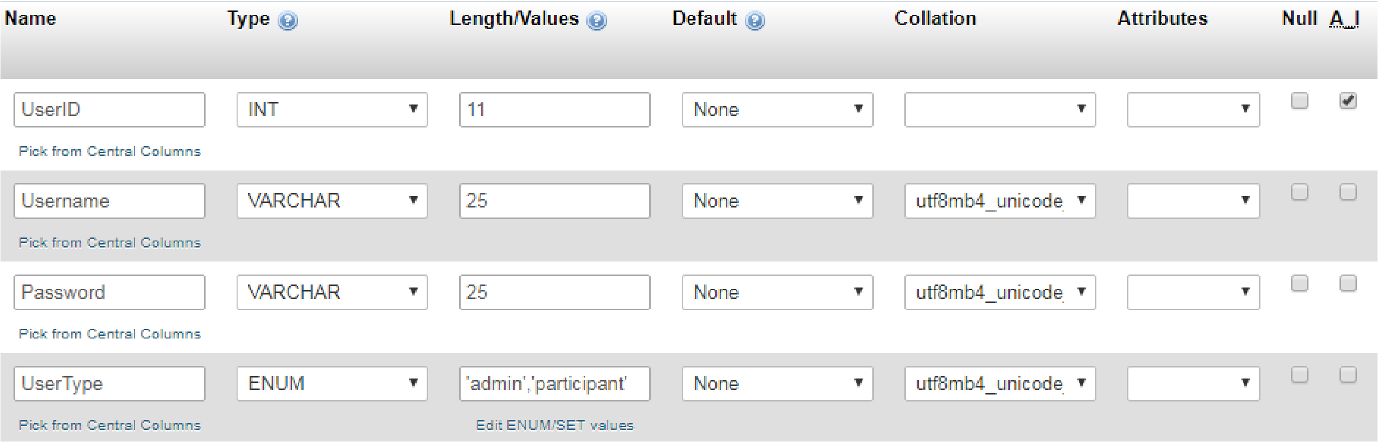
After selecting the row, the admin will be redirected to a different page where he will have option to select certain participants or all of them as shown in the figure below.

Message	Category Smoker	Name ⇧
Non-Smoking Benefits Seminar this afternoon.	Smoker	Dhiraj ⇧ Saroj ⇧

Figure 9. Admin sending selected message from MESSAGE BANK to an individual or a group

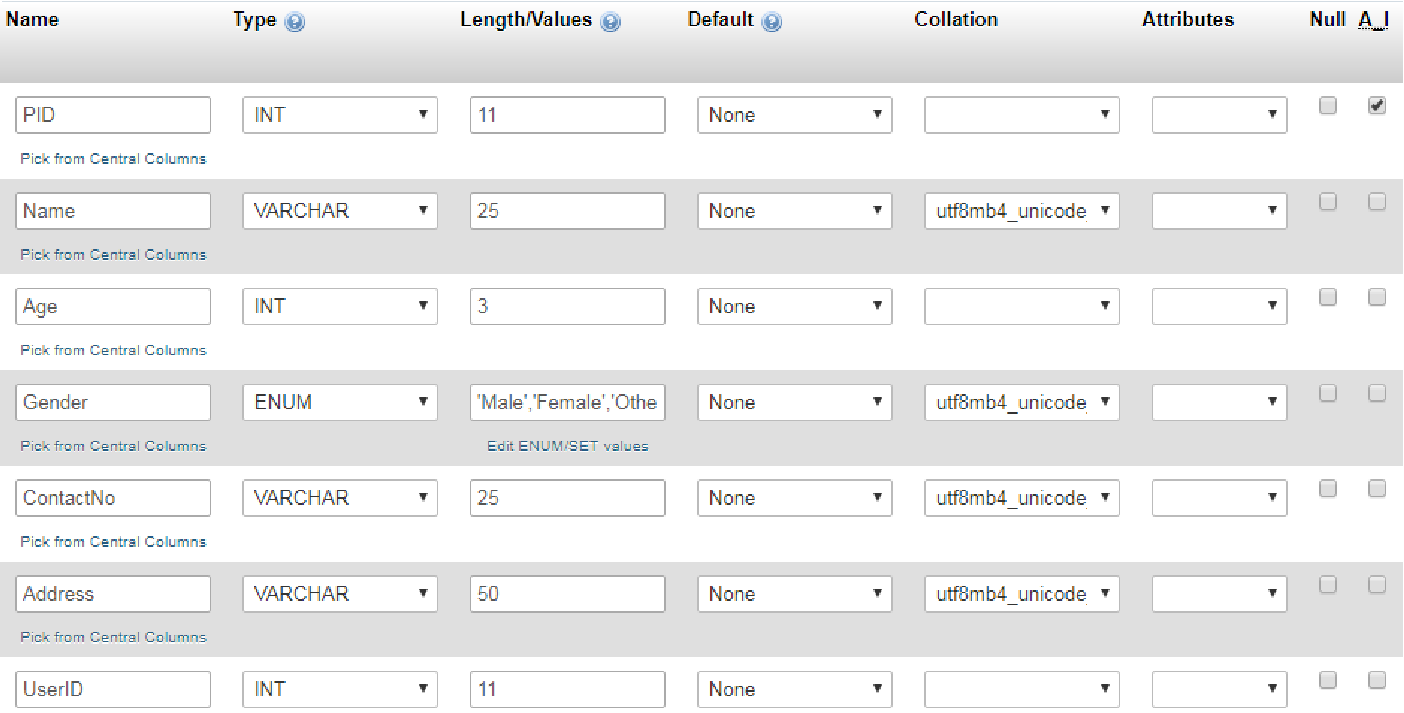
We used MySQL to code the database and the following results were produced using XAMPP.

User Table



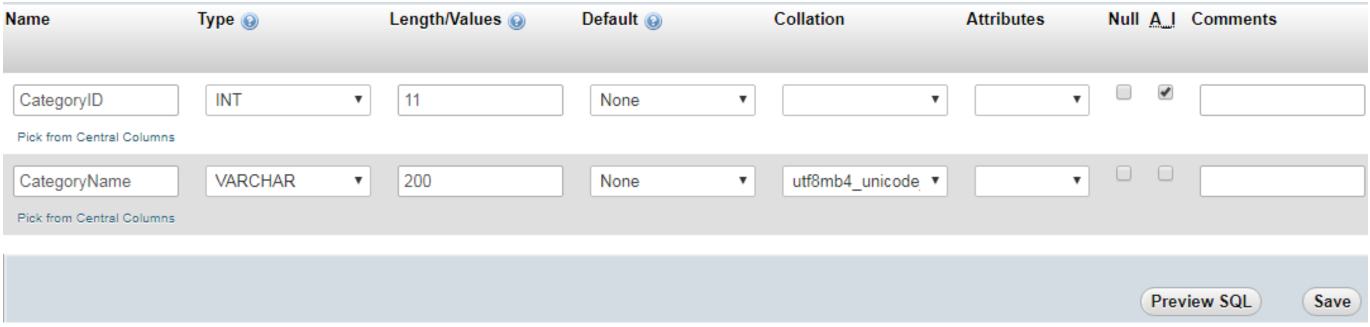
Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I
UserID Pick from Central Columns	INT	11	None			☐	☑
Username Pick from Central Columns	VARCHAR	25	None	utf8mb4_unicode		☐	☐
Password Pick from Central Columns	VARCHAR	25	None	utf8mb4_unicode		☐	☐
UserType Pick from Central Columns	ENUM	'admin','participant' Edit ENUM/SET values	None	utf8mb4_unicode		☐	☐

User info Table



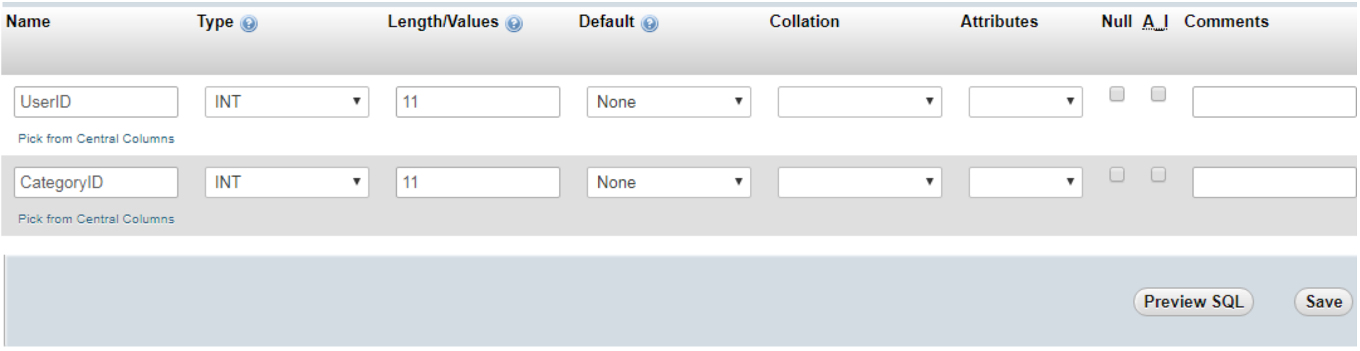
Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I
PID Pick from Central Columns	INT	11	None			☐	☑
Name Pick from Central Columns	VARCHAR	25	None	utf8mb4_unicode		☐	☐
Age Pick from Central Columns	INT	3	None			☐	☐
Gender Pick from Central Columns	ENUM	'Male','Female','Othe Edit ENUM/SET values	None	utf8mb4_unicode		☐	☐
ContactNo Pick from Central Columns	VARCHAR	25	None	utf8mb4_unicode		☐	☐
Address Pick from Central Columns	VARCHAR	50	None	utf8mb4_unicode		☐	☐
UserID	INT	11	None			☐	☐

Category Table



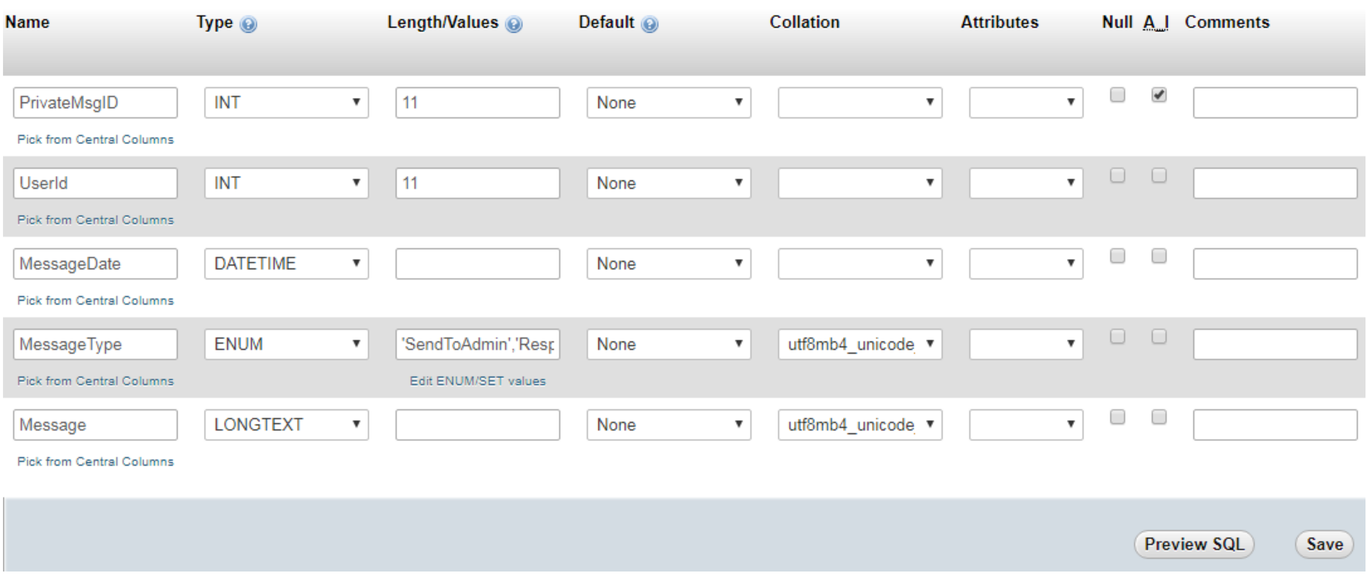
Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I	Comments
CategoryID	INT	11	None			☐	☑	
CategoryName	VARCHAR	200	None	utf8mb4_unicode		☐	☐	

User_category Table



Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I	Comments
UserID	INT	11	None			☐	☐	
CategoryID	INT	11	None			☐	☐	

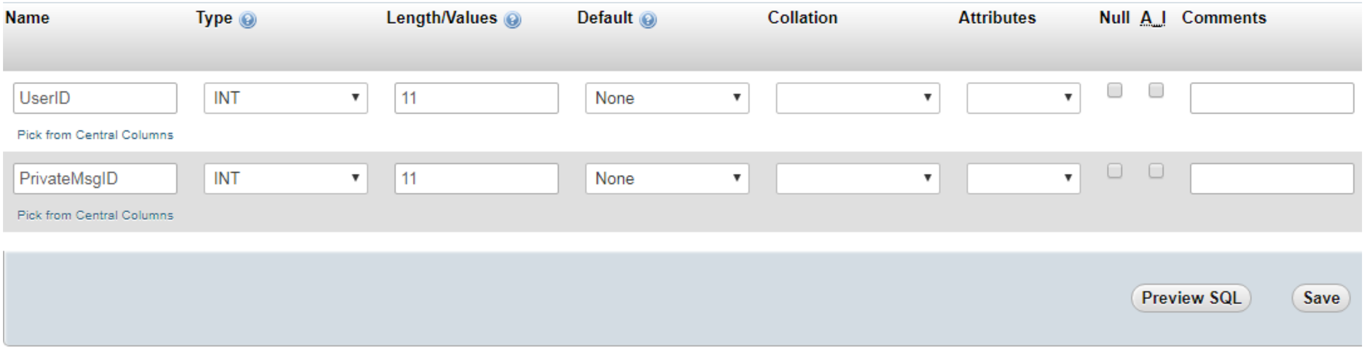
Private_message



Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I	Comments
PrivateMsgID	INT	11	None			☐	☑	
UserId	INT	11	None			☐	☐	
MessageDate	DATETIME		None			☐	☐	
MessageType	ENUM	'SendToAdmin','Resp	None	utf8mb4_unicode		☐	☐	
Message	LONGTEXT		None	utf8mb4_unicode		☐	☐	

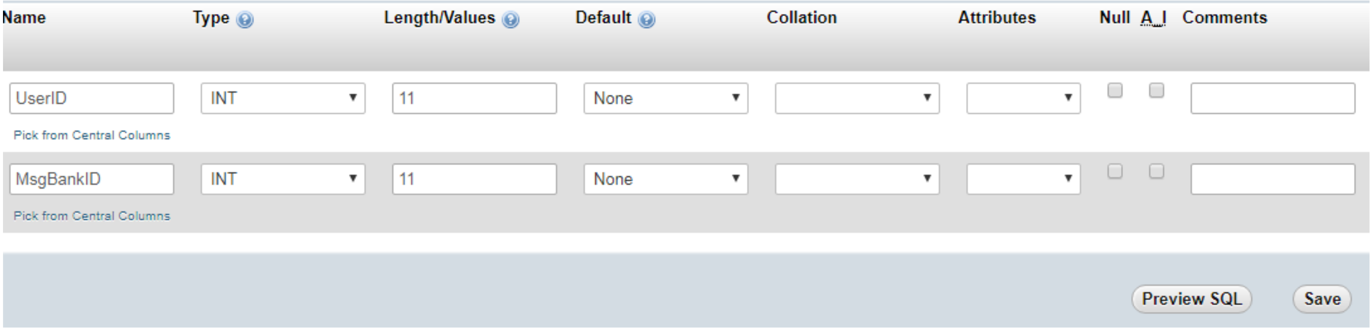
Pick from Central Columns
Pick from Central Columns
Pick from Central Columns
Pick from Central Columns
Edit ENUM/SET values
Pick from Central Columns
Preview SQL
Save

Private_message_notification



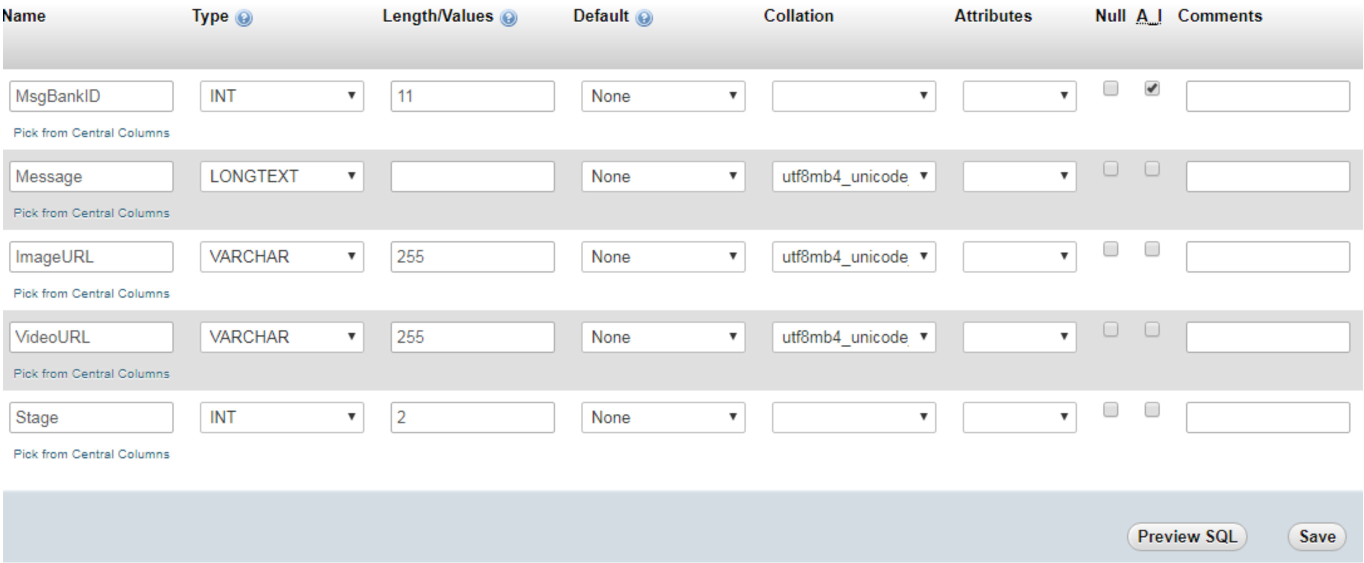
Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I	Comments
UserID Pick from Central Columns	INT	11	None			☐	☐	
PrivateMsgID Pick from Central Columns	INT	11	None			☐	☐	

Group_message_notification



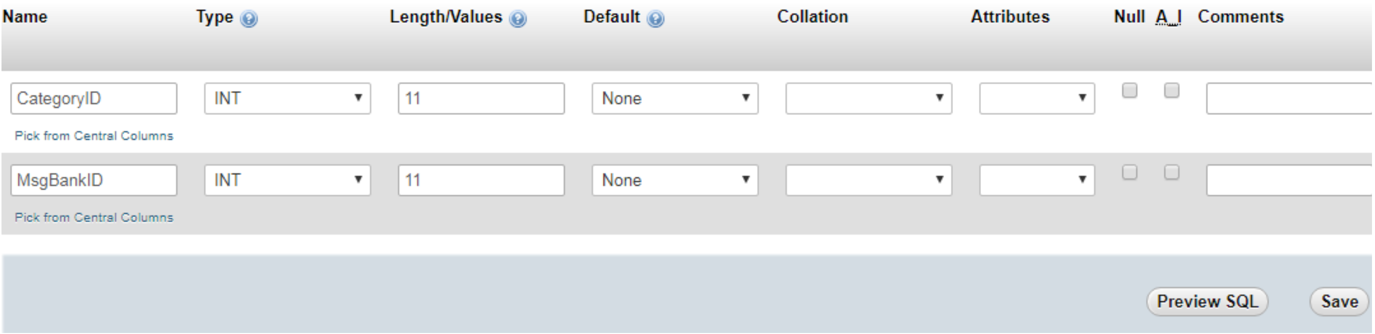
Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I	Comments
UserID Pick from Central Columns	INT	11	None			☐	☐	
MsgBankID Pick from Central Columns	INT	11	None			☐	☐	

Message_bank



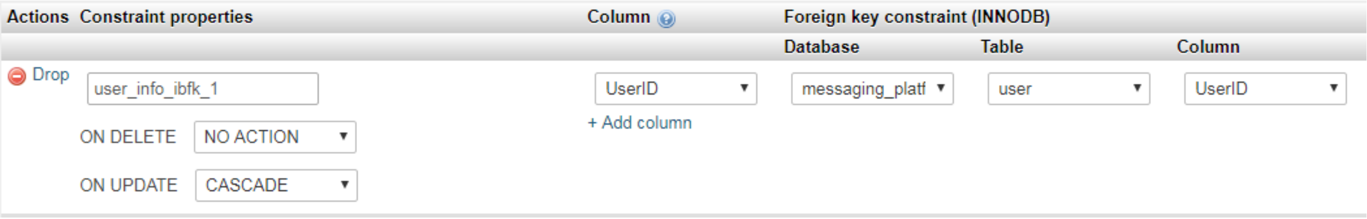
Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I	Comments
MsgBankID Pick from Central Columns	INT	11	None			☐	☑	
Message Pick from Central Columns	LONGTEXT		None	utf8mb4_unicode		☐	☐	
ImageURL Pick from Central Columns	VARCHAR	255	None	utf8mb4_unicode		☐	☐	
VideoURL Pick from Central Columns	VARCHAR	255	None	utf8mb4_unicode		☐	☐	
Stage Pick from Central Columns	INT	2	None			☐	☐	

Message_category



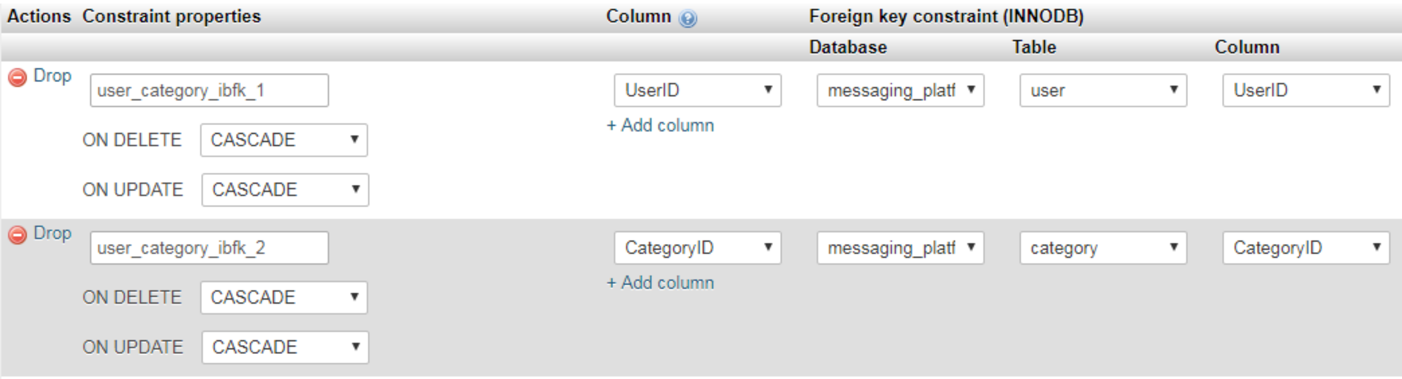
Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I	Comments
CategoryID Pick from Central Columns	INT	11	None			□	□	
MsgBankID Pick from Central Columns	INT	11	None			□	□	

Add Foreign key to **User_info** Table



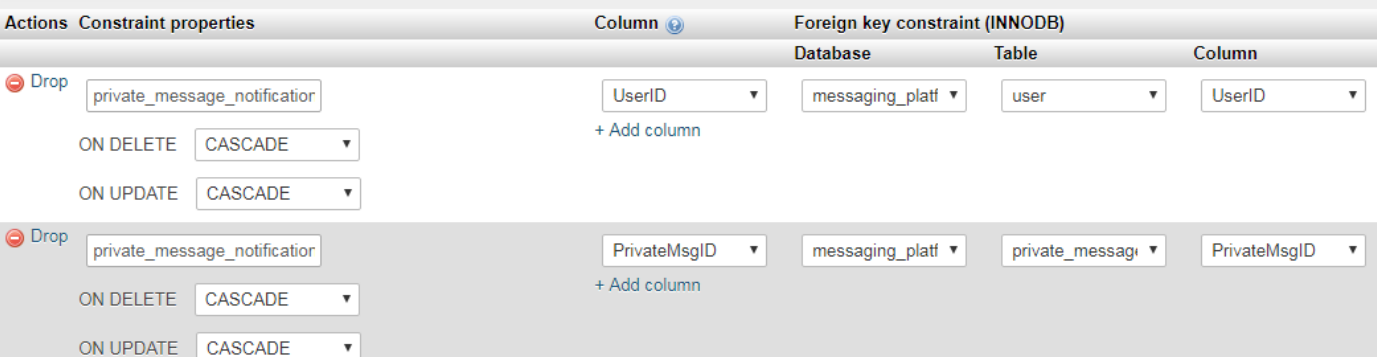
Actions	Constraint properties	Column	Foreign key constraint (INNODB)		
			Database	Table	Column
Drop	user_info_ibfk_1 ON DELETE NO ACTION ON UPDATE CASCADE	UserID + Add column	messaging_platf	user	UserID

Add Foreign key to **User_category** Table



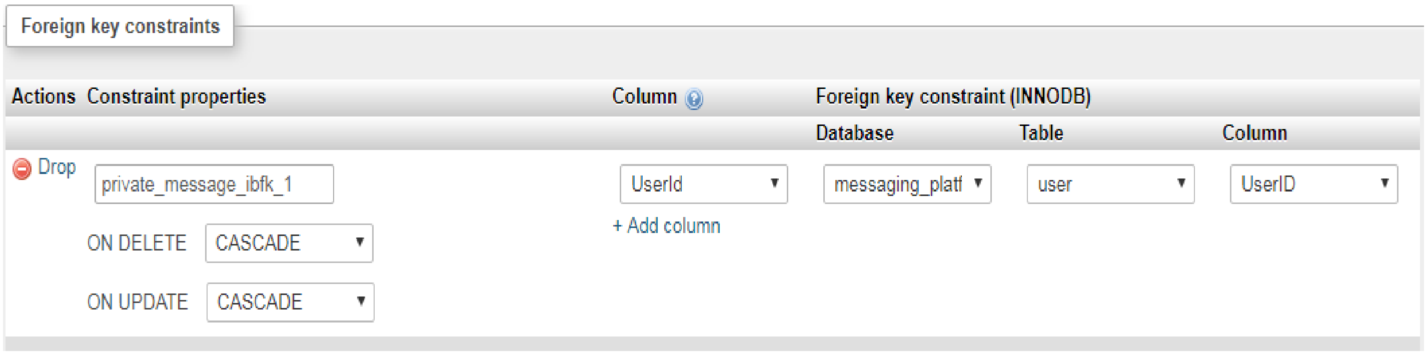
Actions	Constraint properties	Column	Foreign key constraint (INNODB)		
			Database	Table	Column
Drop	user_category_ibfk_1 ON DELETE CASCADE ON UPDATE CASCADE	UserID + Add column	messaging_platf	user	UserID
Drop	user_category_ibfk_2 ON DELETE CASCADE ON UPDATE CASCADE	CategoryID + Add column	messaging_platf	category	CategoryID

Add Foreign key to **Private_message_notification** Table



Actions	Constraint properties		Column	Foreign key constraint (INNODB)		
				Database	Table	Column
Drop	private_message_notificatior		UserID ▾	messaging_platf ▾	user ▾	UserID ▾
	ON DELETE	CASCADE ▾	+ Add column			
	ON UPDATE	CASCADE ▾				
Drop	private_message_notificatior		PrivateMsgID ▾	messaging_platf ▾	private_messag ▾	PrivateMsgID ▾
	ON DELETE	CASCADE ▾	+ Add column			
	ON UPDATE	CASCADE ▾				

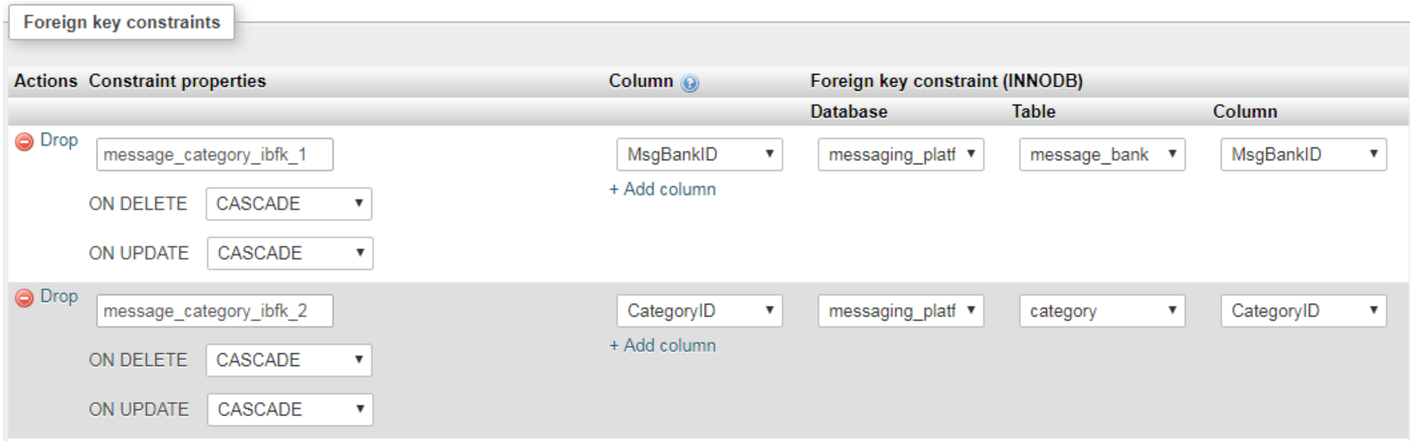
Add Foreign key to **Private_message** Table



Foreign key constraints

Actions	Constraint properties		Column	Foreign key constraint (INNODB)		
				Database	Table	Column
Drop	private_message_ibfk_1		UserId ▾	messaging_platf ▾	user ▾	UserID ▾
	ON DELETE	CASCADE ▾	+ Add column			
	ON UPDATE	CASCADE ▾				

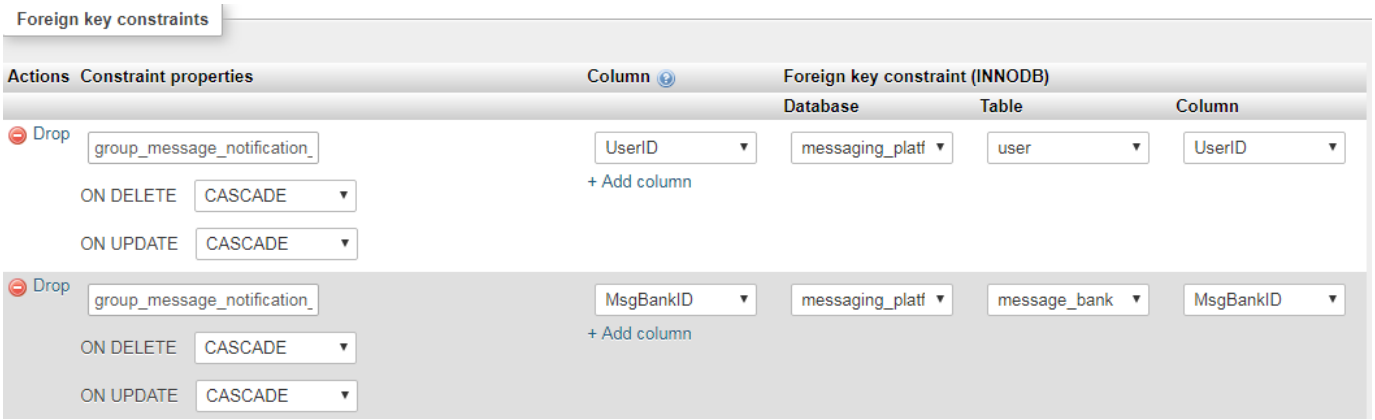
Add Foreign key to **message_category** Table



Foreign key constraints

Actions	Constraint properties		Column	Foreign key constraint (INNODB)		
				Database	Table	Column
Drop	message_category_ibfk_1		MsgBankID ▾	messaging_platf ▾	message_bank ▾	MsgBankID ▾
	ON DELETE	CASCADE ▾	+ Add column			
	ON UPDATE	CASCADE ▾				
Drop	message_category_ibfk_2		CategoryID ▾	messaging_platf ▾	category ▾	CategoryID ▾
	ON DELETE	CASCADE ▾	+ Add column			
	ON UPDATE	CASCADE ▾				

Add Foreign key to **group_messaging_notification** Table



Foreign key constraints

Actions	Constraint properties		Column	Foreign key constraint (INNODB)		
				Database	Table	Column
Drop	group_message_notification_		UserID + Add column	messaging_platf	user	UserID
	ON DELETE	CASCADE				
	ON UPDATE	CASCADE				
Drop	group_message_notification_		MsgBankID + Add column	messaging_platf	message_bank	MsgBankID
	ON DELETE	CASCADE				
	ON UPDATE	CASCADE				

Solution Data Owner

Person 1: Dhiraj Aryal

Person 2: Aafreen Shaikh

Web-Based User Interface

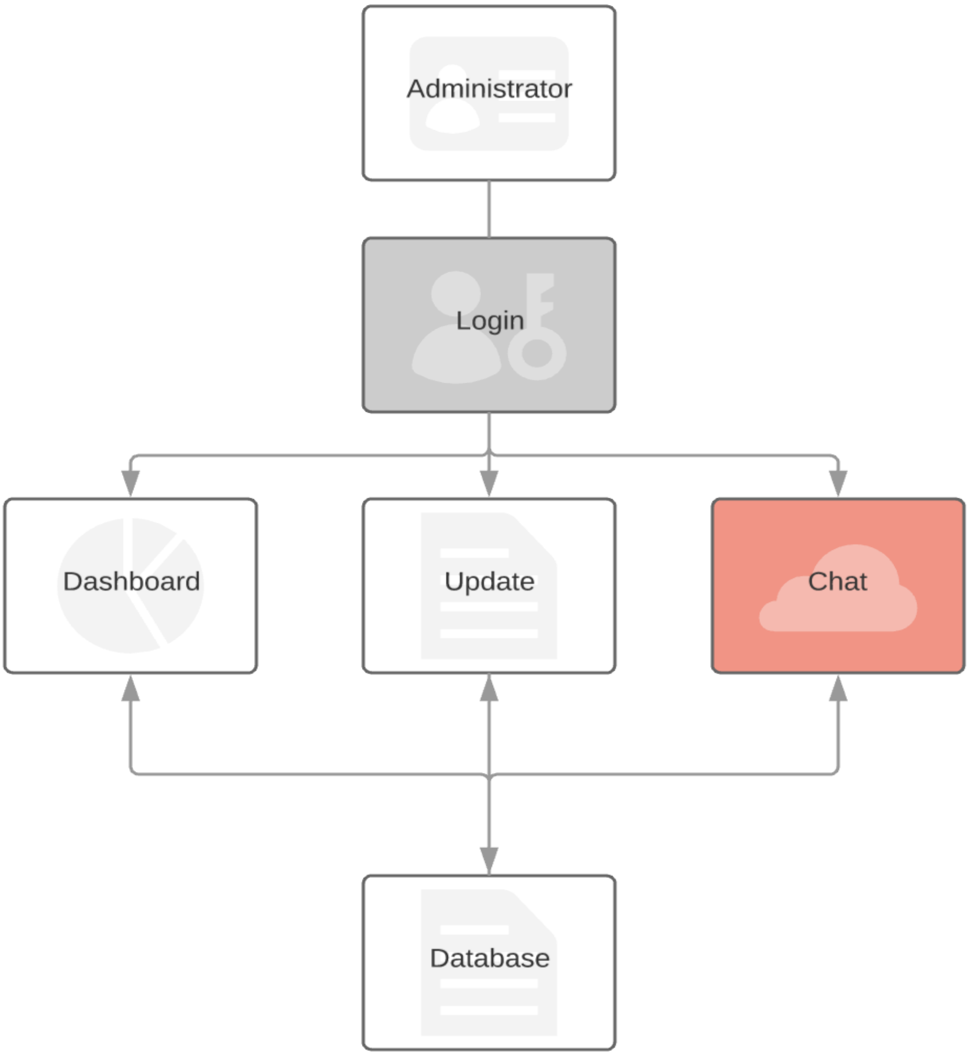


Figure 10. Web-based interface flowchart

Web-based interface will be designed for administrators using WordPress. Only the authorised personnel or administrator will be able to login to view content. As shown in fig. 1.1 (flowchart) there will be three different menus: Dashboard, Update & Chat all interacting with database to perform the actions. Dashboard as shown in fig. 1.2 will show the analytics of participants from database with the graphical representation of categorised participants based on health condition & behaviour. Update menu as shown in fig. 1.3 can be used to enter participants details & corresponding message & update the database. Lastly Chat section as shown in fig. 1.4 will be used to send new messages & view messages from participants & respond to queries from participants.

Solution Requirement:

A live website for the administrative use to update & view participants details along with the functionality to receive & send appropriate messages and the implementation of security measures.

Solution Objective:

A web-based interface to be developed as per solution requirement.

Solution Description:

i) An Update page will be created where administrator will be able to update the details of the participants.
ii) A Dashboard page will be there displaying the statistics of the participants.
iii) Administrator will be able to send & receive messages through Chat page

Solution Data Owner:

Person 1 - Bhishan Shrestha
Person 2 - Abdul

Mock Ups

Dashboard

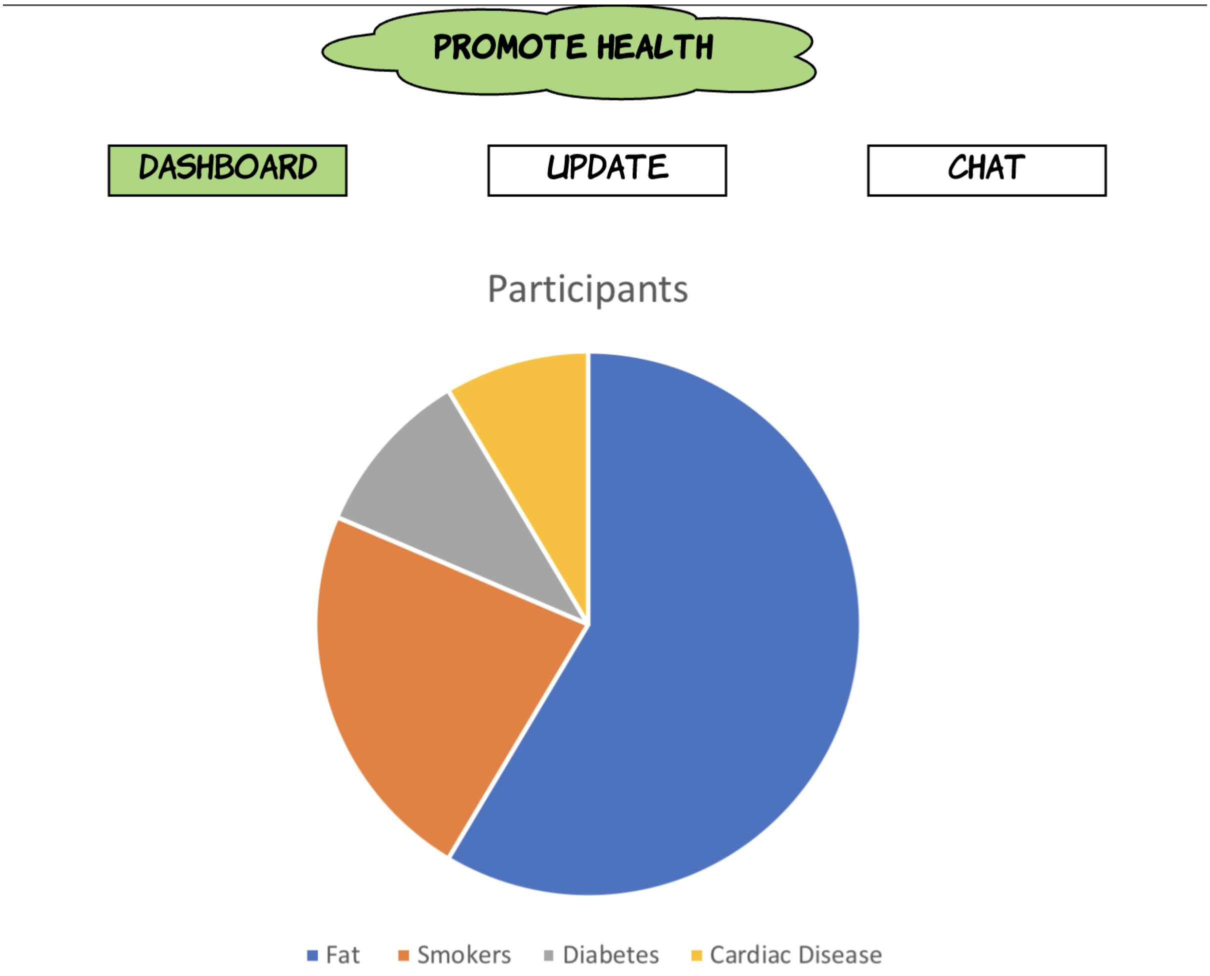


Figure 11.Dashboard

Update

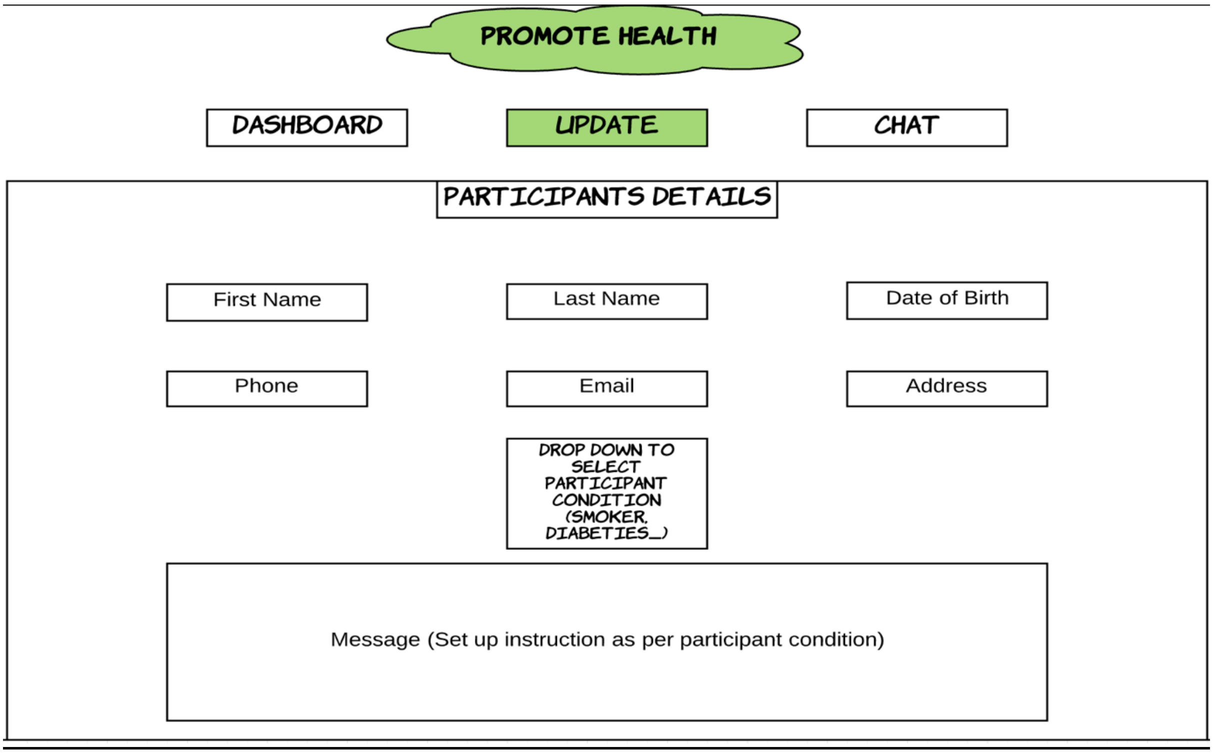

PROMOTE HEALTH
DASHBOARD
UPDATE
CHAT
PARTICIPANTS DETAILS
First Name
Last Name
Date of Birth
Phone
Email
Address
DROP DOWN TO SELECT PARTICIPANT CONDITION (SMOKER, DIABETIES_)
Message (Set up instruction as per participant condition)

Figure 12.Update Participant and message

Chat

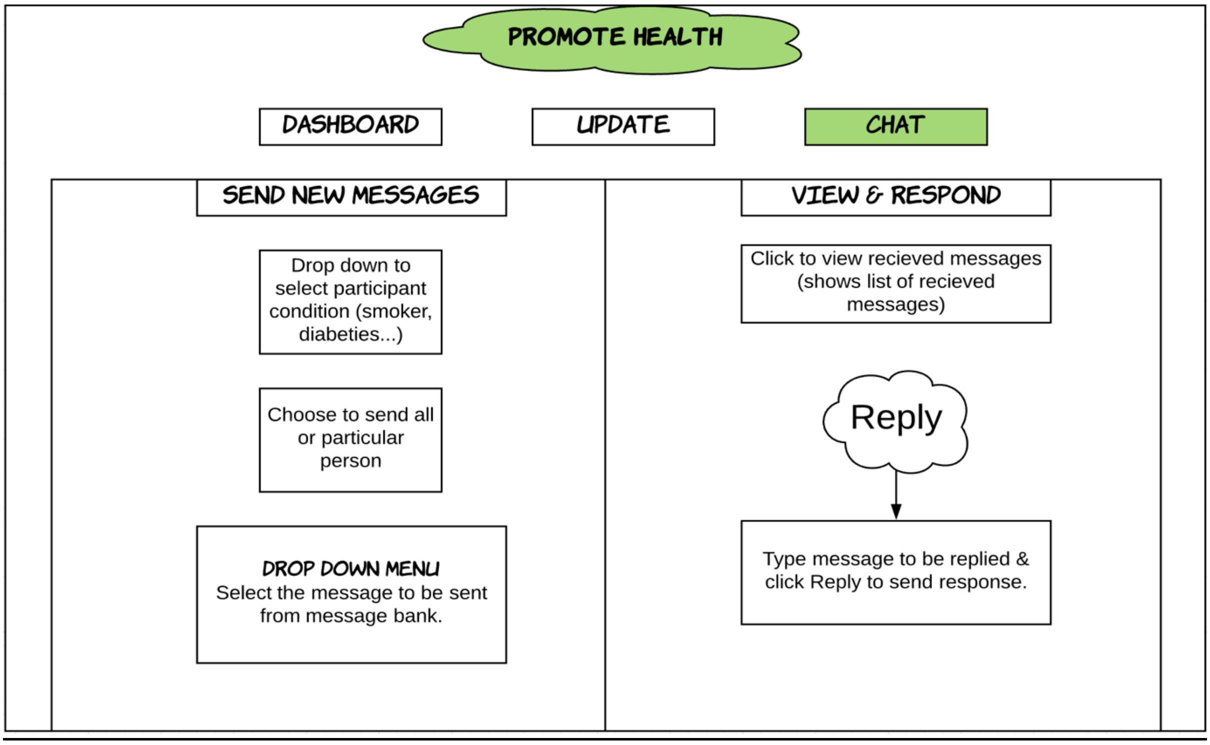

PROMOTE HEALTH
DASHBOARD
UPDATE
CHAT
SEND NEW MESSAGES
Drop down to select participant condition (smoker, diabeties...)
Choose to send all or particular person
DROP DOWN MENU
Select the message to be sent from message bank.
VIEW & RESPOND
Click to view recieved messages (shows list of recieved messages)
Reply
Type message to be replied & click Reply to send response.

Figure 13.Chat (Send and receive message)

Health Mobile messaging app: Android

In the mobile app development, we will be designing the android app which communicate between admin and the customers. This app is mainly used for the receiving health related message, and client can chat with the admin.

There are three section/form in the mobile app, that is a sign up form, message form, chat form.

Once the user downloads the app, install them the very option he gets is create a profile/signup page. Once the signup is successful he will be getting message or seeing old message. Along with that a chat option its available in case he has some queries.

Solution Data Owner:

Person 1 – Mushthaq Ahmed Palbasha Rasheed

Person 2 - Ahmed Bilal Tariq

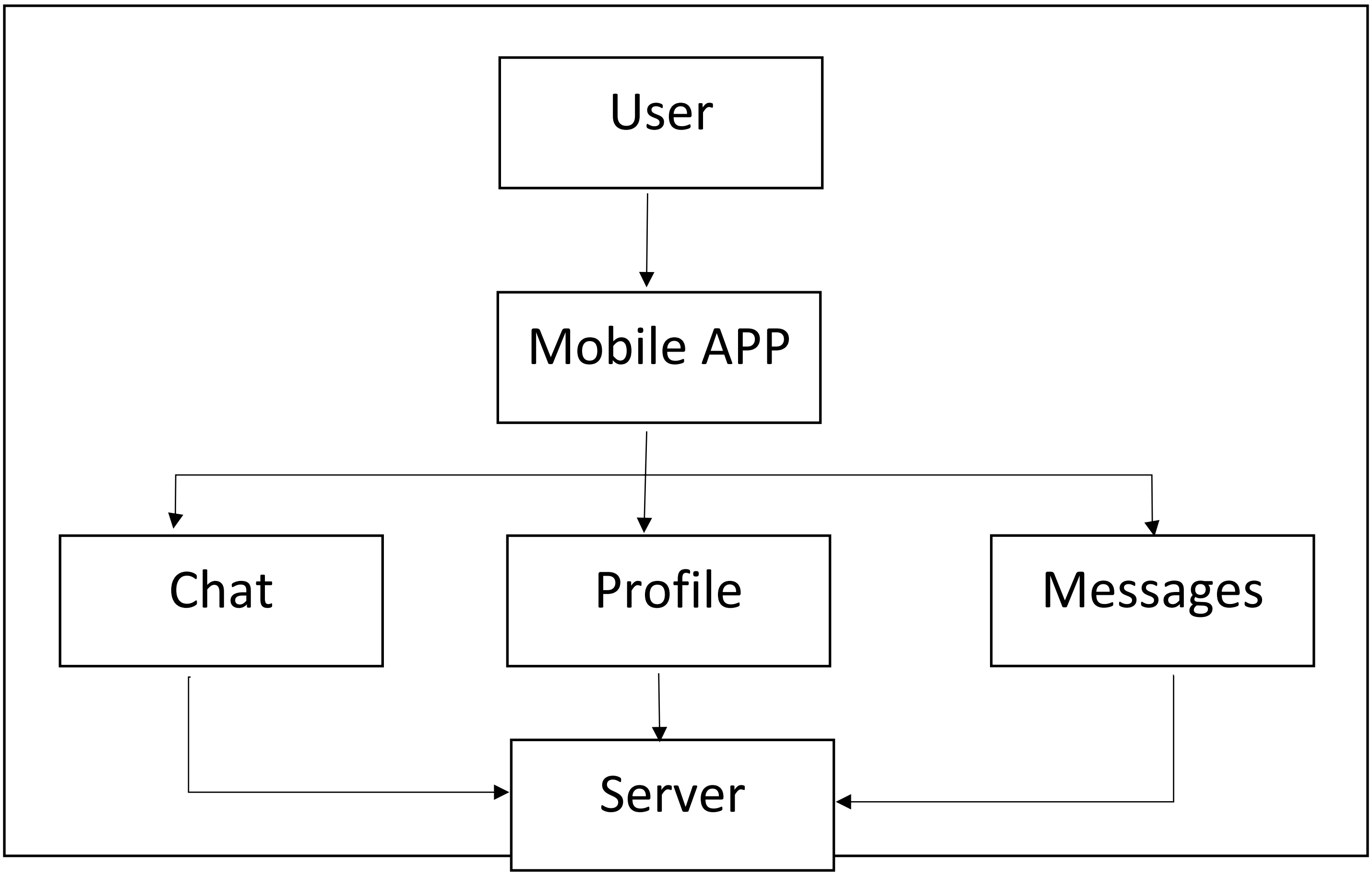


Figure 14.Mobile App Interface Flowchart

Below image are the template, in the actual application images will changes

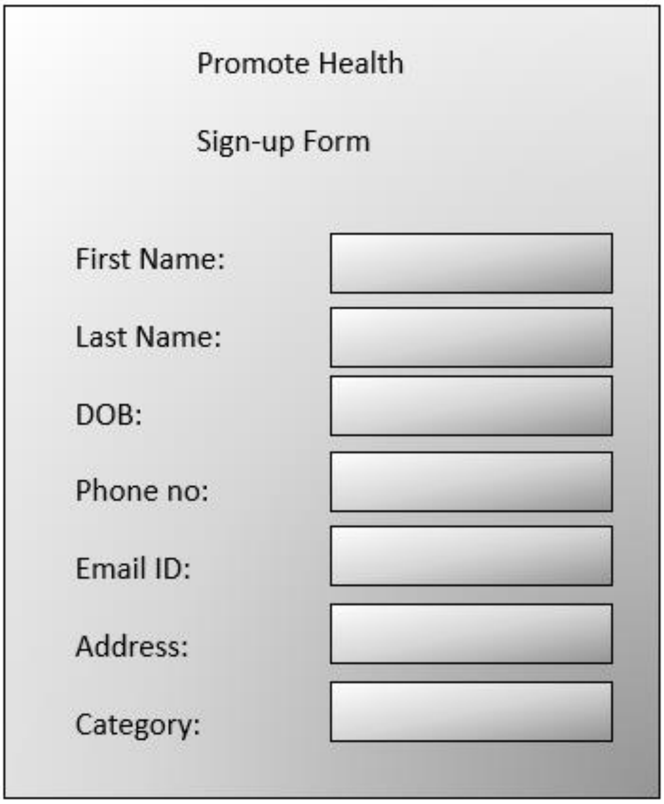


Figure 15.ign-up form

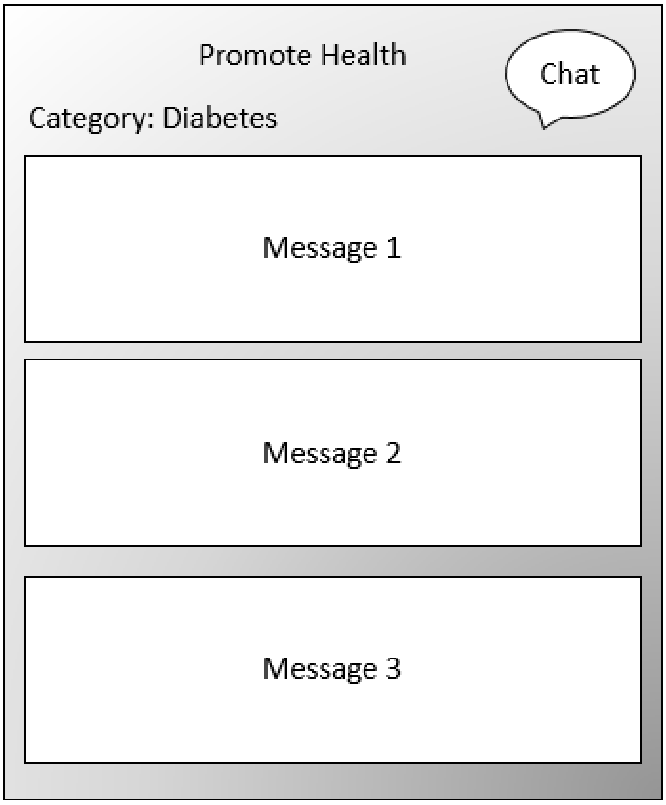


Figure 15. Messages related to the category

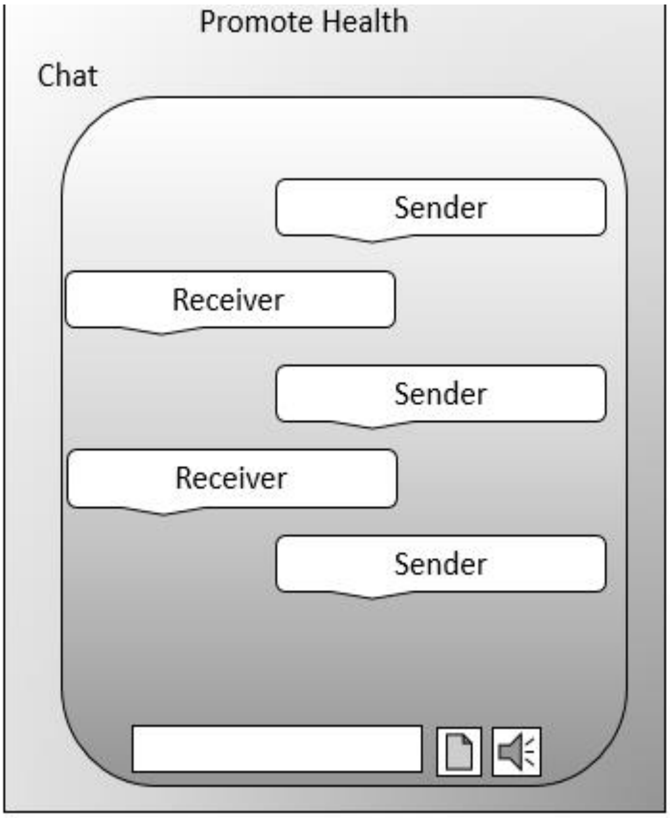


Figure 16. Char messages

Project Process

Systems Impacted:

i) Website
ii) Database

Solution Delivery Methods & Data Update Processes:

Solution will be tested & implemented using WordPress & x10hosting Server.

Methodologies:

i) Server setup with the initial configuration
ii) Installation & Configuration of WordPress
iii) Database setup to communicate with website
iv) Creation of different parts of website to meet the required functionality

Constraints:

i) Availability of resources
ii) Time frame

Solution Verification Process:

i) Team compiles each portion of site to demonstrate the checkpoint has been applied
ii) Each part of the solution & final integration will be proactively flagged by taking the feedback from supervisor & client.

Solution Documentation Inputs:

i) Requirements
ii) Project proposal document

Solution Documentation Outputs:

i) Minutes & Work log updates
ii) Updates to repository of GitHub & Trello
iii) Design documents