A Project Paper on Smart Gym Management System

Submitted By

Monir Ahmed
ID: 2012-2-60-048
&
Jannatun Nayeem
ID: 2012-2-60-050

Supervised By

Md. Shamsujjoha
Senior Lecturer
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
EAST WEST UNIVERSITY

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
EAST WEST UNIVERSITY

Fall 2016

Letter for Acceptance

This Project entitled "**Smart Gym Management System**" submitted by Monir Ahmed (ID:2012-2-60-048) and Jannatun Nayeem(ID:2012-2-60-050), to the Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh is accepted by the department in partial fulfillment of requirements for the Award of the Degree of Bachelor of Science in Computer Science and Engineering on Fall, 2016.

Supervisor

Md. Shamsujjoha
Senior Lecturer
Department of Computer Science and Engineering,
East West University, Dhaka, Bangladesh

Chairperson

Dr. Md. Mozammel Huq Azad Khan
Professor and Chairperson,
Department of Computer Science and Engineering,
East West University, Dhaka, Bangladesh

Declaration

We "Monir Ahmed & Jannatun Nayeem", students of Computer Science and Engineering at East West University, Dhaka, hereby declare that the project work entitled "**Smart Gym Management System**" is a record of an original work done by us under the guidance of our honorable sir Md. Shamsujjoha , Senior Lecturer , Department of Computer Science and Engineering.

Signature of Students

(Monir Ahmed)

(Jannatun Nayeem)

Acknowledgements

It gives us enormous pleasure to submit the project report on “**Smart Gym Management System**”. Our report won’t be possible without contribution of few people.

At first we desire to express our deepest sense of gratitude of almighty Allah for giving us the strength and the composure to complete this report within the scheduled time.

Then, we would like to express our gratitude & respect to our honorable teacher and project Supervisor Md. Shamsujjoha, Senior Lecturer, East West University for his constant guidance, advice, encouragement & every possible help in the overall preparation of this report.

We would also like to express our excessive thanks to our team mate for the excellent support in completing our project.

Finally, we are very much grateful to our families who always give us constant support and encouragement. We would like to thank our seniors who helped us greatly to complete this paper. In addition, we will mention our friends who also inspired and helped us to finish our work. We hope that this project paper has been prepared for the fulfillment of the course requirement. We would also like to thank our authority of East West University. They prepare this course to make us understand for near future.

Abstract

Any business that does not have a website is missing out on one of the most powerful marketing tools available to them. The main reason that it is important for businesses to have a website is how people are likely to find you. These days most people will go online and research products and companies before they make a purchase. For this reason of dependency on online, we develop a website for a gymnasium in which customer can get all information about the gymnasium and he/she can access or enroll at the gym via online. Usually, the client uses MS Excel or paper, and maintains their records, however it is not possible them to share the data from multiple system in multi user environment, there is lot of duplicate work, and chance of mistake. When the records are changed they need to update each and every excel file. The Smart Gym Management System eliminates most of the limitations of the existing software. Increasing efficiency and effectiveness, automation, accuracy, user-friendly interface, information availability, communication capacity, maintenance, cost reduction makes our system smarter than the existing system. We integrate some new and prominent features along with all the necessary features. Some of them are user login by face recognition, webcam integration, using Twilio SMS API for phone number verification.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

We all know health is a wealth. We do not need a fancy car, big apartment, a doctor degree without a health. Being healthy is a first thing we need to keep in mind. Because most of time our attitude depends on how we feel. Being healthy and fit gives us energy to do anything. Physical fitness is very necessary for a healthy and tension free life. Physical fitness includes diet, exercise and sleep. These three basic things have their own importance in each individual's life and everyone should be sensible with regard to these for a healthy life.

1.1 Overview of the Project

Our proposed "Smart Gym Management System" is for those who run a gym business. Before doing anything we did a decent research on major difficulties for gym owners. We examined carefully about how to make a huge registering system without failure as well as different functions for different kind of user depending on their privilege.

The Gym Management requires a system that will handle all the necessary and minute details easily and proper database security accordingly to the user. They requires software, which will store data about members, employees, products, payroll, receipts of members & all transactions that occur in Gym.

1.2 Objectives

The objectives of this study are summarized below:

- The main objective of the project is to design and develop a user friendly efficient computerized Gym Management System.
- An accurate system without any data redundancy.
- Secured data storage for Authority end.
- Secure the user ends data by providing each user’s own personal credentials.
- A flexible system which can maneuver the customer-staff relationship in an effective manner.
- To provide better graphical user interface.
- Computerization can be helpful as means of saving time & money.

Chapter 2

Theoretical Background

We have done a project on Gym Management and database management and transactions. This system is proposed to be an automate database management & transactions. This stores employee, member, payroll, receipts, and products information. It also provides the facility of search & advanced search for searching the records efficiently & immediately. This system provides data storing & report generation with graphical user interface (GUI).

2.1 System Study

It is always necessary to study and recognize the problems of existing system, which will help in finding out the requirements for the new system. System study helps in finding different alternatives for better solution.

The project study basically deals with different operations:

1: Data Gathering
2: Study of Existing System
3: Analyzing Problems
4: Studying various documents
5: Feasibility study for further improvements

Following are the steps taken during the initial study:

Initially, we collected all the information, which they wanted to store. Then we studied the working of the current system which is done manually. We noted the limitation of that system which motivated them to have new system. With the help of these documents we got basic ideas about the system as well as input output of the developed system.

The most important thing is to study system thoroughly. Here we are studying both existing system and proposed system so that advantages & disadvantages of both the systems can be understood. The first task was identifying how system can be computerized. Some analysis and projections was done regarding changes to be made to the existing system. The new developed system for Gym Management is simple without complexities.

2.2 Existing System

An Existing system refers to the system that is being followed till now. The gym is working manually. The current system is time consuming and also it is very costly, because it involves a lot of paperwork. To manually handle the system was very difficult task. But now-a-days computerization made easy to work.

The following are the reasons why the current system should be computerized:

- To increase efficiency with reduced cost.
- To reduce the burden of paper work.
- To save time management for recording details of each and every member and employee.
- To generate required reports easily.

2.3 Proposed System

The online gym management system is user-friendly application. This automated system makes all functionality easier for both owners and customers. It is very simple in design and to implement. The system requirements are very low. System resources and the system will work in almost all configurations.

It has the following objectives:

Enhancement:

The main objective of Smart Gym Management System is to enhance and upgrade the existing system by increasing its efficiency and effectiveness. The software improves the working methods by replacing the existing manual system with the computer-based system.

Automation:

The Smart Gym Management System automates each and every activity of the manual system and increases its throughput. Thus the response time of the system is very less and it works very fast.

Accuracy:

The Smart Gym Management System provides the uses a quick response with very accurate information regarding the users etc. Any details or system in an accurate manner, as and when required.

User-Friendly:

The software Smart Gym Management System has a very user-friendly interface. Thus the users will feel very easy to work on it. The software provides accuracy along with a pleasant interface. Make the present manual system more interactive, speedy and user friendly.

Availability:

The transaction reports of the system can be retried as and when required. Thus, there is no delay in the availability of any information, whatever needed, can be captured very quickly and easily.

Maintenance Cost:

Reduce the cost of maintenance.

Chapter 3

System Analysis & Design

The way that is followed while carrying on with the development application is as follows:

3.1 Defining a problem

Defining a problem is one of the important activities of the project. The objective is to define precisely the business problem to be solved & thereby determined the scope of the new system. This phase consist of 2 main tasks. The 1st task within this activity is to review the organization needs that originally initiated the project. The 2nd task is to identify, at an abstract or general level, the expected capabilities of the new system. Thus, it helps us to define the goal to be achieved & the boundary of the system. A clear understanding of the problem will help us in building a better system & reduce the risk of project failure. It also specifies the resources that have to be made available to the project. Three important factors project goal, project bounds & the resource limits are sometimes called the project’s term of reference.

3.2 Feasibility study

The systems objectives outlined during the feasibility study serve as the basic from which the work of system design is initiated. Much of the activities involved at this stage is of technical nature requiring a certain degree of experience in designing systems, sound knowledge of computer related technology and through understanding of computers available in the market and the various facilities provided by the vendors. Nevertheless, a system cannot be designed in isolation without the active involvement of the user. The user has a vital role to play at this stage too. As we know that data collected during feasibility study wills we utilized systematically during the system design. It should, however be kept in mind that detailed study of the existing system is not necessarily over with the completion of the feasibility study. Depending on the plan of feasibility study, the level of detailed study will vary and the system design stage will also vary in the amount of investigation that still needs to be done.

This investigation is generally an urgent activity during the system. Sometimes, but rarely, this investigation may form a separate stage between feasibility study and computer system design. Designing a new system is a creative process, which calls for logical as well as lateral thinking. The logical approach involves systematic moves towards the end product keeping in mind the capabilities of the personnel and the equipment at each decision making step. Lateral thought implies encompassing of ideas beyond the usual functions and equipment. This is to ensure that no efforts are being made to fit previous solutions into new situations.

The feasibility study proposes one or more conceptual solutions to the problem set for the project. The objective in assessing feasibility is to determine whether a development project has a reasonable chance of success. It helps us to determine the input & output of the system. The following are the criteria that are considered to confirm the project feasibility.

The following feasibility study was undertaken for the proposed system:

Technical feasibility:

At first it’s necessary to check that the proposed system is technically feasible or not & to determine the technology and skill necessary to carry out the project. If they are not available then find out the solution to obtain them. Hardware is already available in the University.

Economic feasibility:

While considering economic feasibility, it is checked in points like performance, information and outputs from the system. The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project , which will give best , return at the earliest. One of the factors , which affect the development of a new system , is the cost it would require.

Social feasibility:

Although generally there is always resistance, initially to any change in the system is aimed at reliving the work load of the users to extent the system is going to facilitate user to perform

Operations like calculating salary amounts and deductions, generating reports with less possible errors. Thus there is no reason to make system socially unfeasible.

Chapter 4

Modules & Features

4.1 Module

There are two basic modules in this system as describe briefly in below

- **Administrative module:** This user is an admin type who has full rights on the system.
- **User module:** This is a normal level of user who will be very few number of functionality of website.

4.1.1 Administrative Module

This module includes storing and retrieving the details of the data.

- Create , Update, Manage, Delete User
- Creating Offer Plan
- Manage Billing
- Manage User Enquiry through Email
- Manage Owner Info

4.1.2 User Module

Depends on the privilege user's access to features of the application is granted. Below are the some important functionality of user module.

- Applying for Package
- Account Update
- Online Payment Facility
- Login by Face Recognition
- Phone and Email Address Verification on Sign Up
- Enquiry to Authority

4.2 Features

There are many features in our system. Some salient and new features are:

- Login by Face Recognition
- Phone number verification through SMS on sign up
- Online Payment Gateway
- Webcam Integration
- Activity Log of User's

4.3 Features Description

SMS API:

We used API (application programming interface) for sending verification code through SMS.

An application programming interface (API) allows you integrate the great features of an SMS messaging platform, with your own website or application program. It is important to wisely think through your options before choosing an SMS API.

Sending SMS online is a best solution for business owners. SMS gateway API service allows business owners to reach thousands of clients within a short period of time. The major advantage of sending SMS from PC is that SMS are sent through a number of different ways in order to make sure that they are transported to the targeted recipients on time.

We have used ZARSS SMS API for sending verification code through sms.

But we also tried Twilio rest api for sending sms. Beacuse Twilio is a cloud communications Platform as a service company based in San Francisco, California. Twilio allows software developers to programmatically make and receive phone calls and send and receive text messages using its web service APIs. We used Twilio free trial account . To send a message by Twilio's rest API, just make an HTTP POST to Twilio with the body of the message and the phone number you want to send it to. SMS messages must be sent from Twilio SMS-enabled phone numbers due to the architecture of the global SMS network.

For sending sms we need some information such as account SID and auth token which we will find after sign up in twilio. If your REST request is successful, the SMS has been successfully queued for transmission. The SMS will be sent as soon as possible at a maximum rate of 1 message per second per 'From' phone number.

As we are using a Twilio Trial account, we will only be able to send SMS messages to phone numbers that we have verified with Twilio. Phone numbers can be verified via your Twilio Console's Verified Caller IDs. So this is a limitation. We can not send sms to those numbers which are not verified with Twilio and this should be done manually. Its not a dynamic process. That’s why we used ZARSS API.

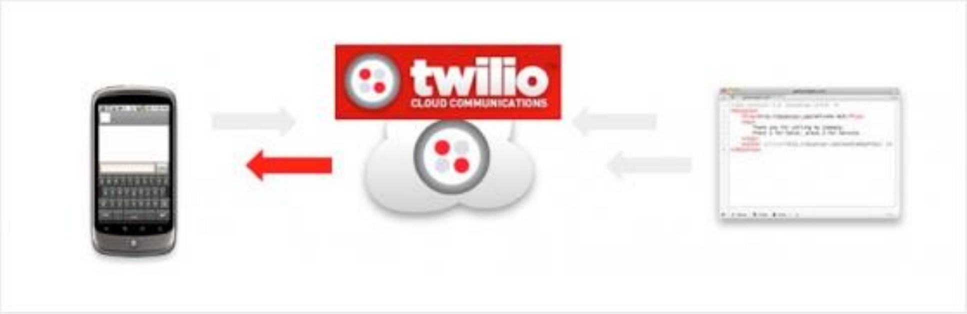
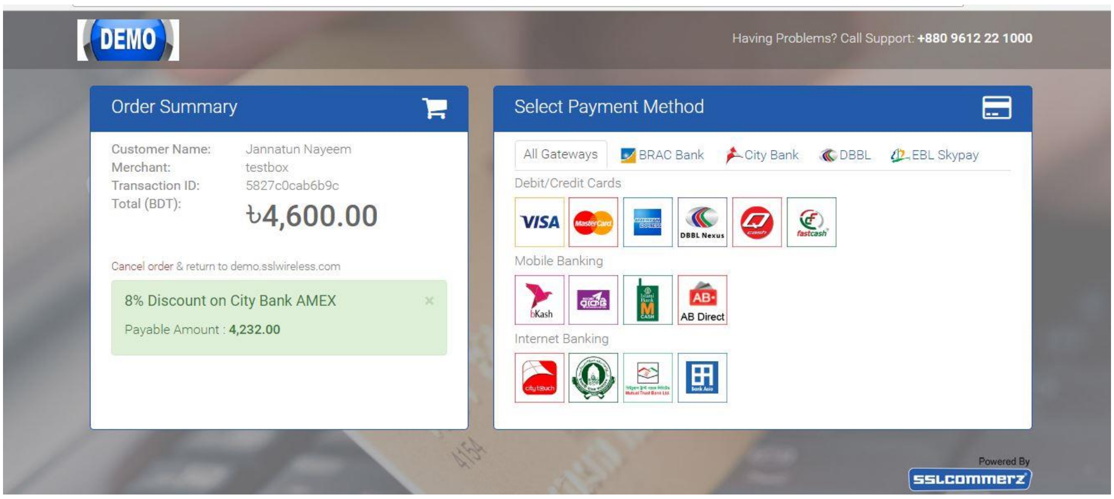


Fig. 4.1: Diagram of Twilio sms sending process

ONLINE PAYMENT GATEWAY:

To make payment easy for our customer we integrate online payment gateway in our system. We use the API of SSLCOMMERZ for online payment. SSLCOMMERZ is the first payment gateway in Bangladesh opening doors for merchants to receive payments on the internet via their online stores. SSLCOMMERZ uses industry standard Secure Sockets Layer (SSL) technology which is used worldwide for securing data encryption. Our customers will be able to pay using local credit/debit cards like VISA, MasterCard, DBBL Nexus Card and Brac Bank credit card or bank accounts right from our website.



Order Summary

Customer Name: Jannatun Nayeem
Merchant: testbox
Transaction ID: 5827c0cab6b9c
Total (BDT): ৳4,600.00

Cancel order & return to demo.sslwireless.com

8% Discount on City Bank AMEX

Payable Amount : 4,232.00

Fig. 4.2: Diagram of online payment process of SSLCOMMERZ

FACE RECOGNITION:

One of our exiting features is login by face recognition. You can capture image by webcam and if the image matches with the previous image you have, you can get access to your account. We use API of Kairos for face recognition.

Principal of face recognition:

Face recognition technology allows a program to identify an individual according to his features: distance between eyes, eyebrows, faces shape, etc. A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image. One of the way is to do this is by comparing selected facial features from the image and a facial database. With face recognition technology, the main challenge is to identify an individual despite the variety of lights, face orientations, camera angle and noise such as background, ageing, glasses or make-up. However, much progress have been achieved in recent years, and there are different types of technologies and algorithms achieving different levels of accuracy.

Kairos API:

Kairos Face Detection is a REST API for 2D and 3D face detection and recognition. In order to benefit from cut edge technologies with 3D modeling, this API would be an interesting solution. Kairos converts a face into a 3D model that will be used to match pictures that will be added in the gallery. As it is an API, it is platform independent and also provides a built SDK for IOS, making it easy to implement on mobile device and considerably lowering the complexity of implementation. A SDK for Android should be released in near future. If this API has the advantage to make advanced face recognition technology very easy to implement, it is not free. It works with a Freemium model: the first 5000 calls to the API are free, other prices are not specified on the website. However, it may still be financially more interesting than the time and cost involved to self-build a 3D model face recognition.

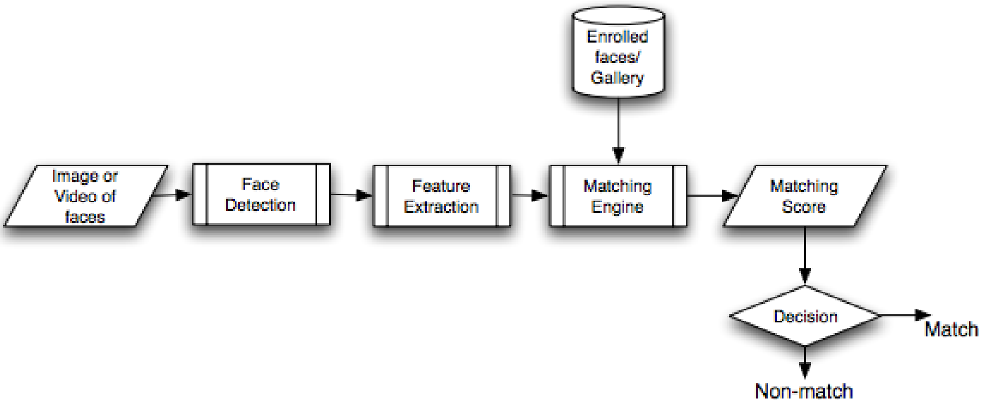


Fig. 4.3: Diagram of face recognition step of Kairos API

WEBCAM INTEGRATION:

We used jQuery Webcam Plugin to capture image. The jQuery webcam plugin is a transparent layer to communicate with a camera directly in JavaScript. This plugin provides three different modes to access a webcam through a small API directly with JavaScript - or more precisely jQuery. Thus, it is possible to bring the image on a Canvas (callback mode), to store the image on the server (save mode) and to stream the live image of the Flash element on a Canvas (stream mode).

Adobe Flash Player Settings

Camera and Microphone Access

www.xarg.org is requesting access to your camera and microphone. If you click Allow, you may be recorded.

Fig. 4.4: Diagram of webcam

ACTIVITY LOG:

We have the option of activity log of user’s. Any activity in the website will be stored in the database. Admin has the access of this log. This will help admin to track the user’s activities. If user do anything wrong, all information of this user’s may help admin.

4.4 Application Requirements

4.4.1 User Interface:

- HTML has been used for developing the user layout for the system.
- PHP and JavaScript has been used for creating all the validations and client side scripting functionality.
- CSS has been used for designing the web page of the system.

4.4.2 Application:

- Client On Internet : Web Browser , Operating System (Any)
- Web Server : Apache
- Database : MySQL
- Markup Language: HTML, CSS
- Scripting Language : PHP , Javascript, JQuery

Chapter 5

System Implementation

5.1 Implementation Methodology

We follow the MVC design pattern for developing our system. Model–view–controller (MVC) is a software design pattern for implementing user interfaces on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user.

- **Model:** The model manages the behavior and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller).
- **View:** The view manages the display of information.
- **Controller:** The controller interprets the mouse and keyboard inputs from the user, informing the model and/or the view to change as appropriate.

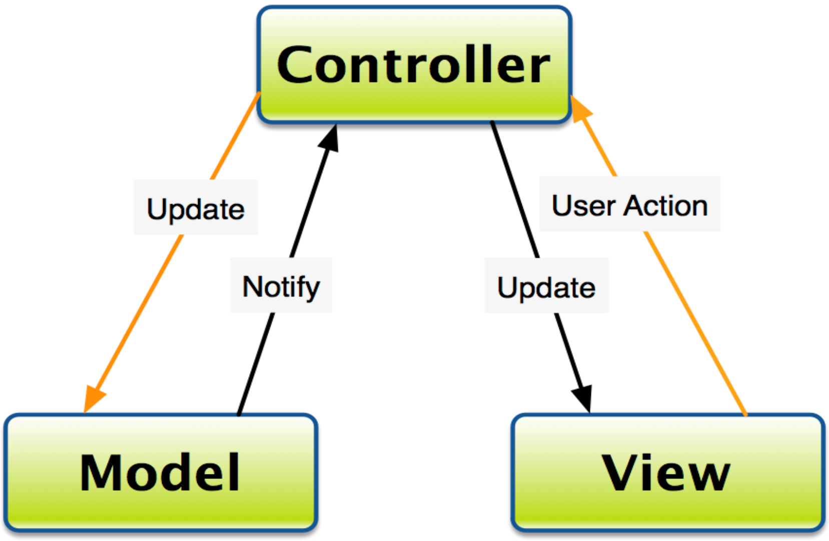


Fig. 5.1: Diagram of A typical collaboration of the MVC components.

5.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.[2] DFDs can also be used for the visualization of data processing.

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

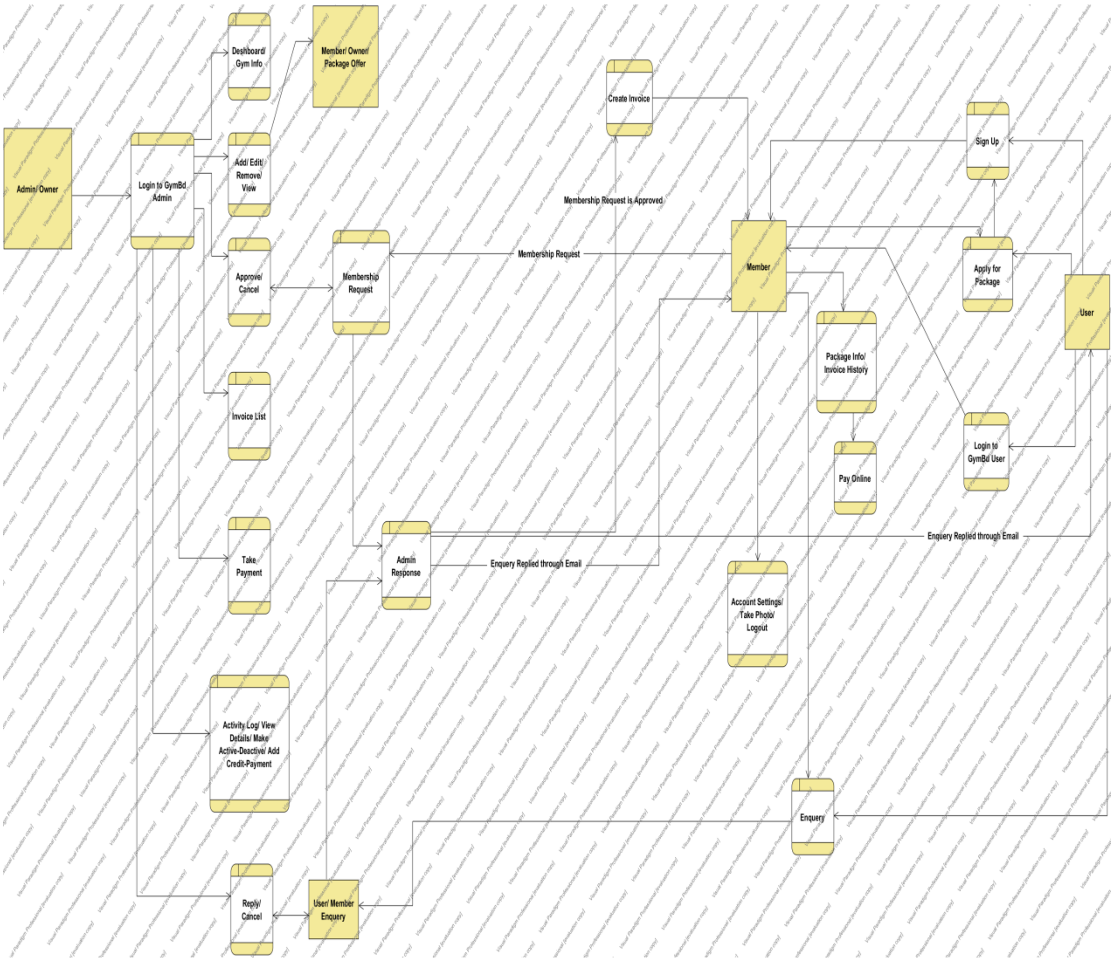


Fig. 5.2: Diagram of Level 0 DFD of our System.

Level 1 DFD (Administrative Panel) :

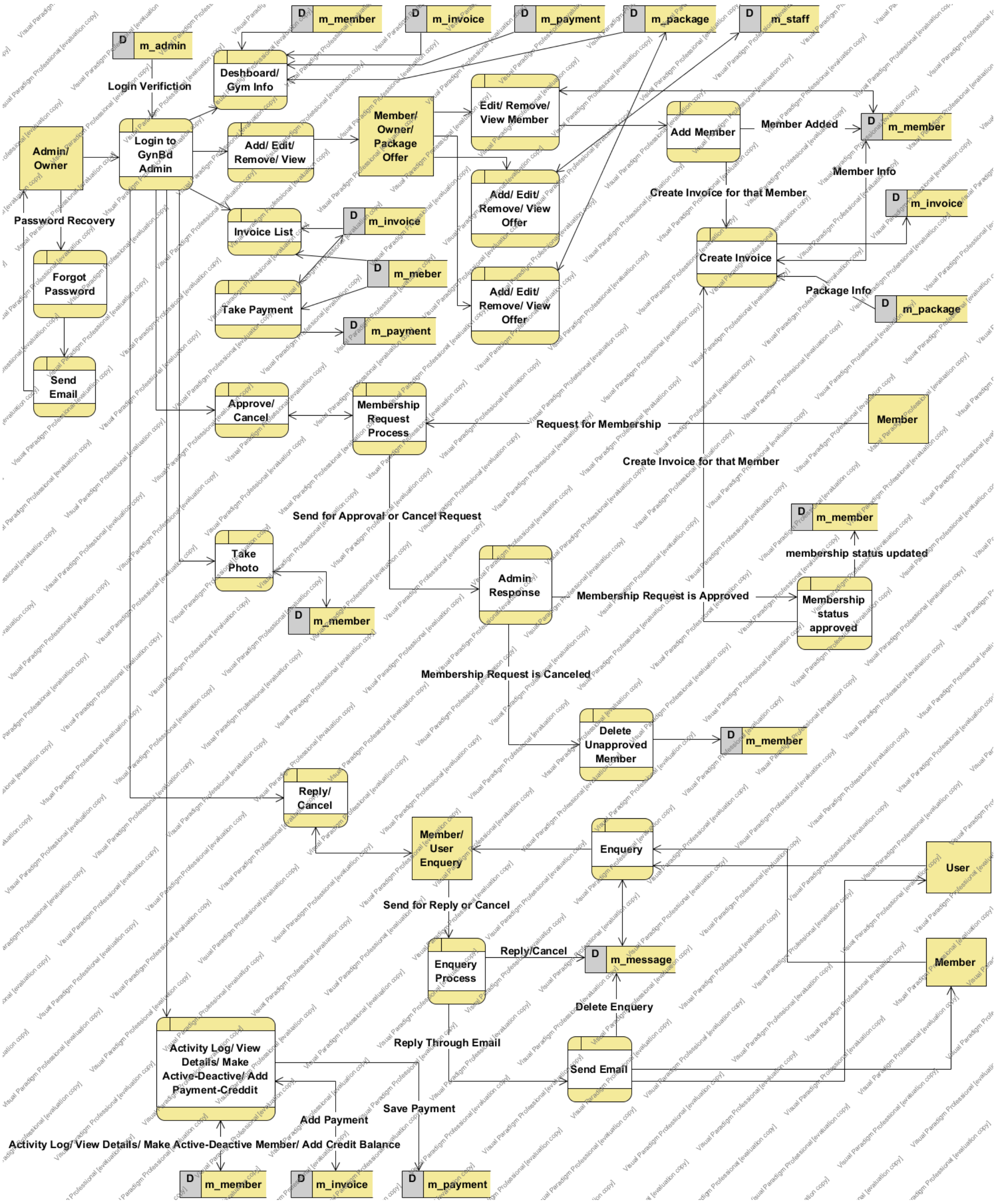


Fig. 5.3(a): Diagram of Level 1 DFD of Administrative Panel.

Level 1 DFD (User Panel) :

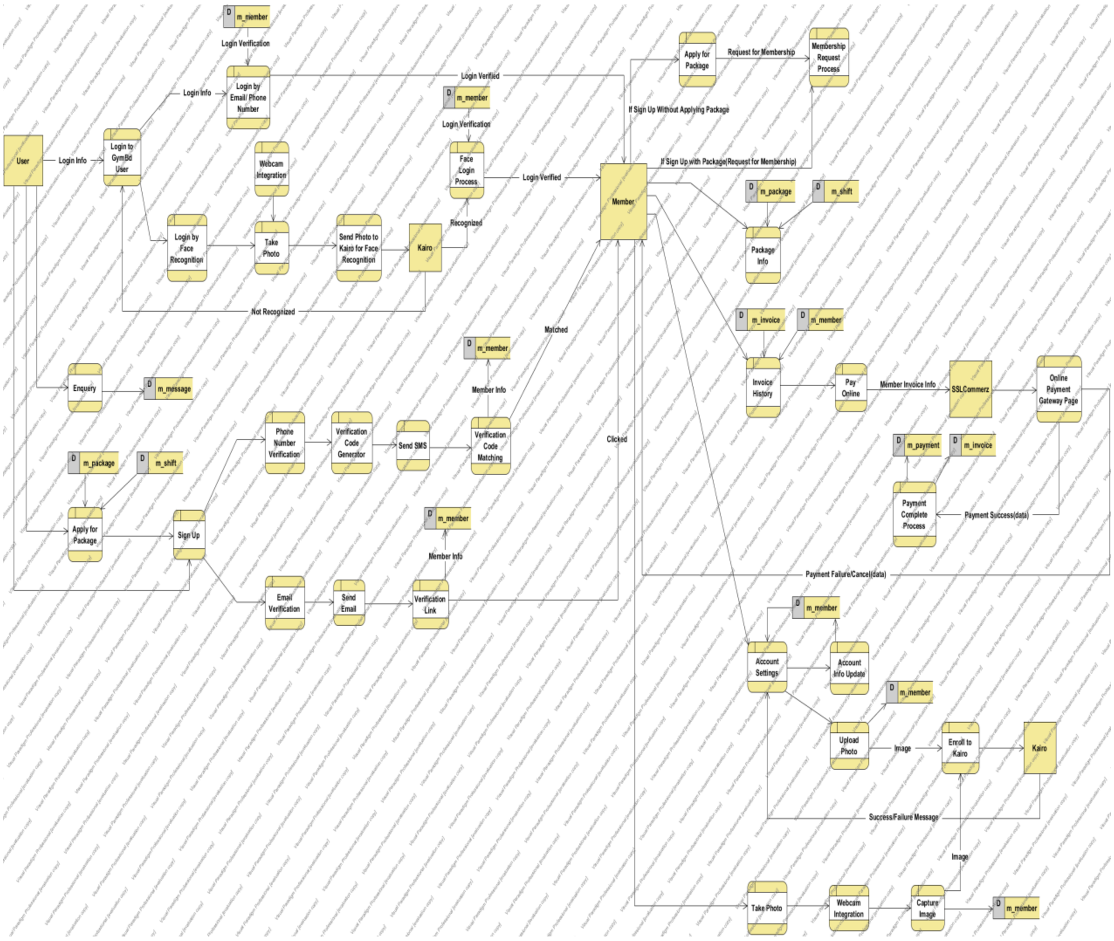


Fig. 5.3(b): Diagram of Level 1 DFD of user Panel.

5.3 Entity Relationship Diagram

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering.

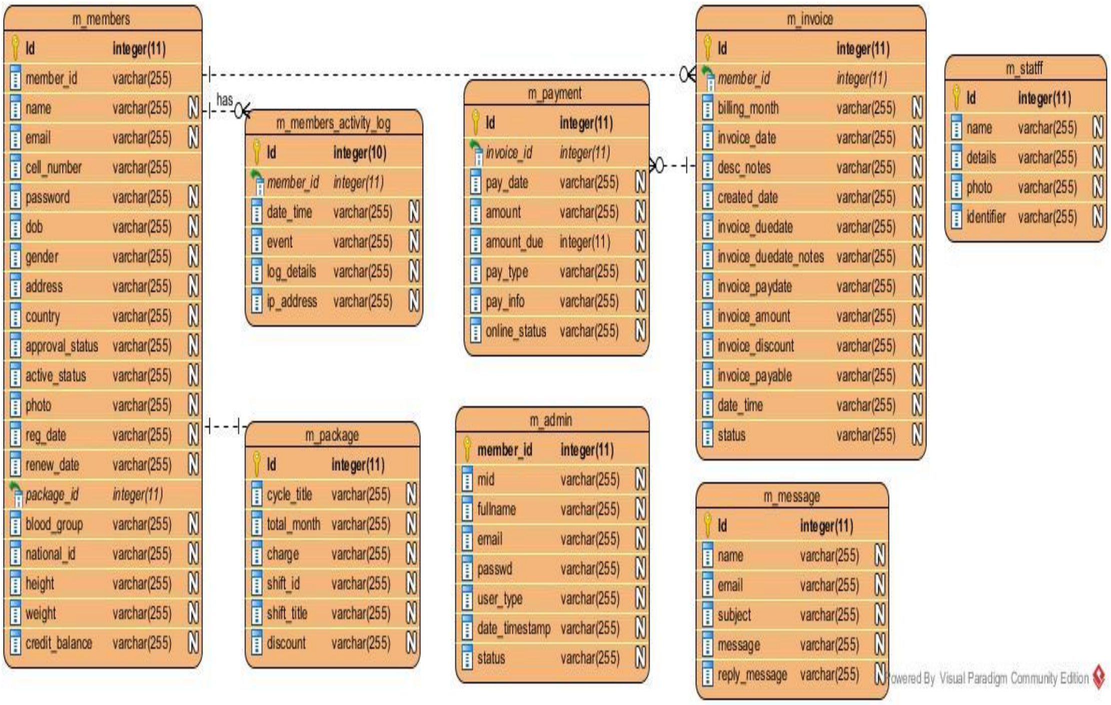


Fig. 5.4: Diagram of Entity Relationship Diagram.

5.4 Normalization

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables.

- Normalization is used for mainly two purpose,
- Eliminating redundant (useless) data.

Ensuring data dependencies make sense i.e. data is logically stored.

Without Normalization, it becomes difficult to handle and update the database, without facing data loss. Insertion, Update and Deletion Anomalies are very frequent if Database is not normalized.

Normalization rule are divided into following normal form.

- First Normal Form
- Second Normal Form
- Third Normal Form
- BCNF

As per First Normal Form, no two Rows of data must contain repeating group of information i.e each set of column must have a unique value, such that multiple columns cannot be used to fetch the same row. Each table should be organized into rows, and each row should have a primary key that distinguishes it as unique.

As per **First Normal Form**, there are no repeating or duplicate fields in our database and each cell contains only a single value. For example:

Id	date_time	event	member_id	log_details	ip_address
1	1478324932	Login	233	Login Successful	103.230.6.54
2	1478324965	Logout	233	Logout Successful	103.230.6.54

Table 5.1**:** User’s activity log table following first normal form.

As per First Normal Form, there are no repeating or duplicate fields in our database. Our system database does not have any column that has multiple repeating values. So our system database table is in first normal form.

As per Second Normal Form, Second normal form states that it should meet all the rules for 1NF and there must be no partial dependences of any of the columns on the primary key:

Let’s check with another table of our database:

Id	cycle_title	total_month	charge	shift_title	Shift_id	discount
1	Monthly	1	400	Morning	1	0%
2	Annually	12	1100	Morning	1	20%
3	Monthly	1	450	Evening	2	0%

Table 5.2**:** Package table following Second normal form

From the above table we can say it meets all the rules for 1NF and there is no column that depends on the primary key. SO our database table is also in Second Normal Form.

In Third Normal Form, it should meet all the rules for 2NF and no non-key fields can depend upon another.

But from the above table we see, there is dependencies on two non-key value shift_id and shift_title . Shift title depends on shift id. So our system database table is not in 3NF.

Chapter 6

System Development Process

6.1 User Module

Our Smart Gym Management System first activates the sign up form. Here the user enters the username and password along with an email or a phone number. If user choose email to sign up then the authentication starts of our system. A link will be provided to the users given email address and by clicking into that link user authentication will be successful and it redirect to our main page. And if user choose phone number to sign up, a verification code will be sent to user's given phone number. Now if user enter the correct verification code, it will redirect to our system's main page otherwise it will give warning to user for invalid information. If user give both email and phone number, then default authentication will be done by sms.
Some screenshots of our sign up process:
Sign Up :

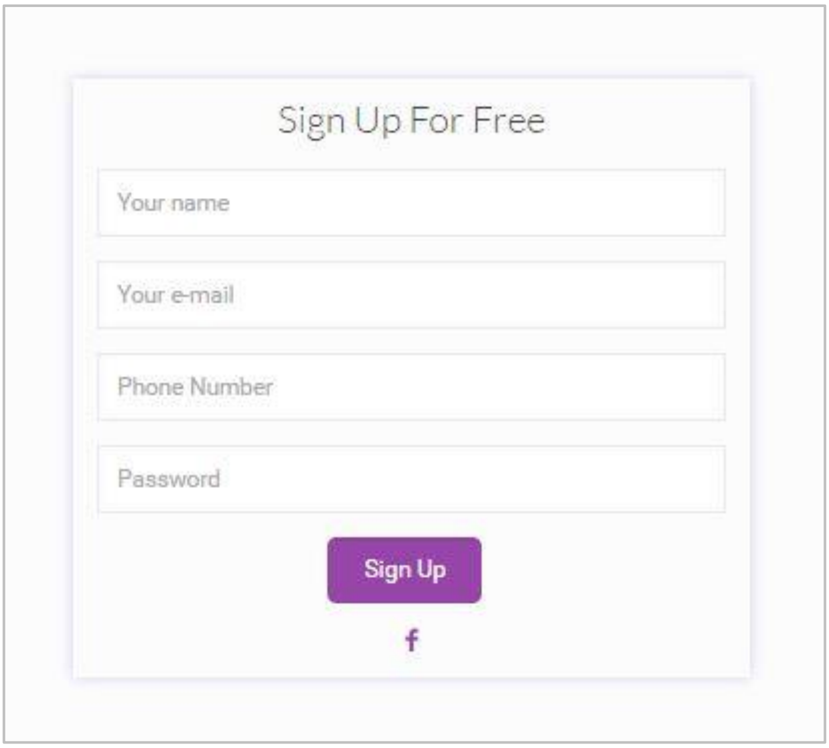


Fig. 6.1: Diagram of System's Sign Up Form.

Let, user give phone number to sign up. Then the authentication form will be like this:

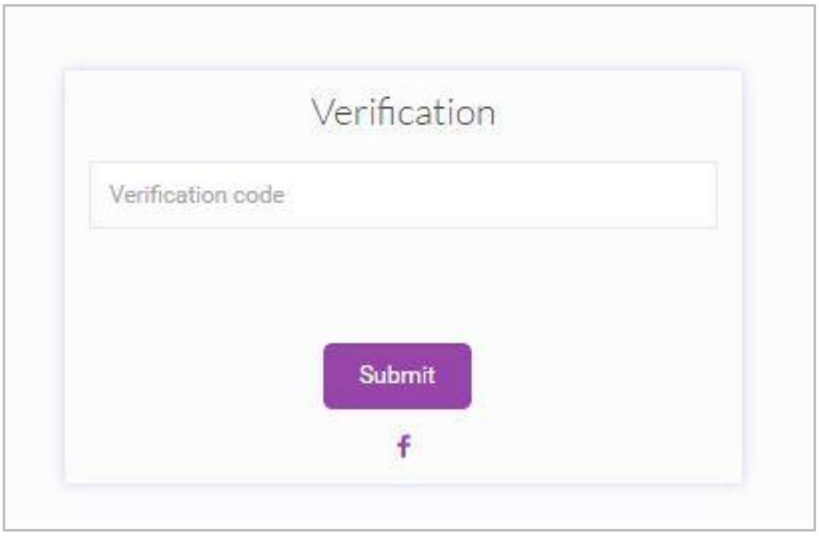
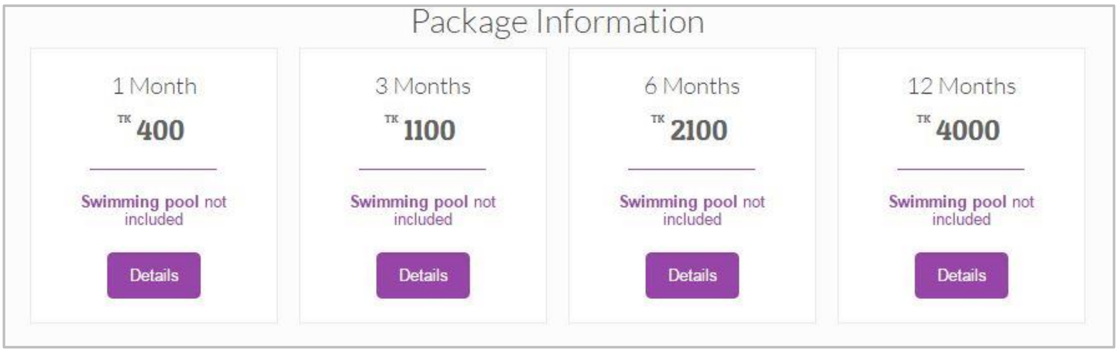


Fig. 6.2: Diagram of Phone Number Verification Form on Sign Up.

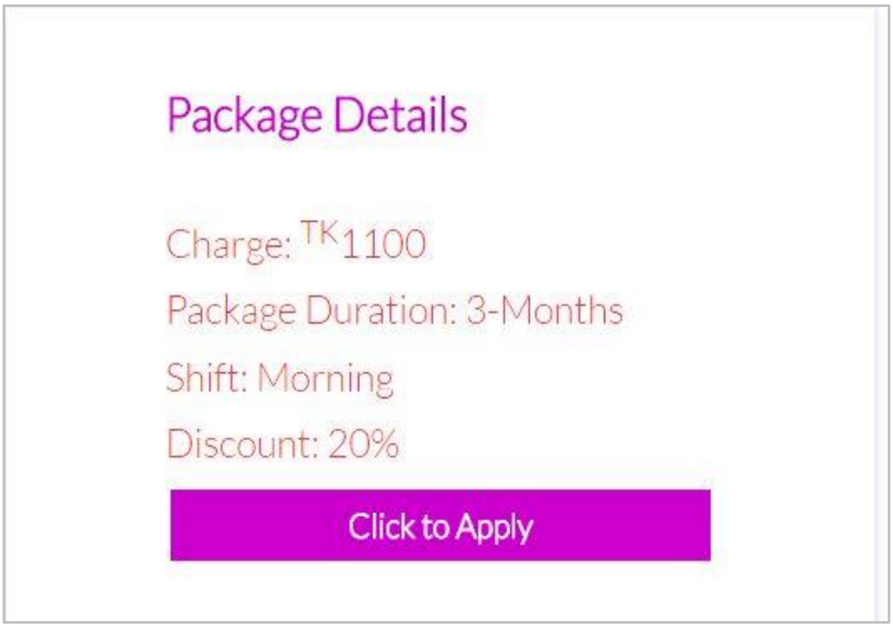
After successful authentication user will be provided with the package information gym has:


Package Information

1 Month	3 Months	6 Months	12 Months
TK 400	TK 1100	TK 2100	TK 4000
Swimming pool not included	Swimming pool not included	Swimming pool not included	Swimming pool not included
Details	Details	Details	Details

Fig. 6.3: Diagram of Package Information.

User can apply for any package:



Package Details

Charge: TK1100

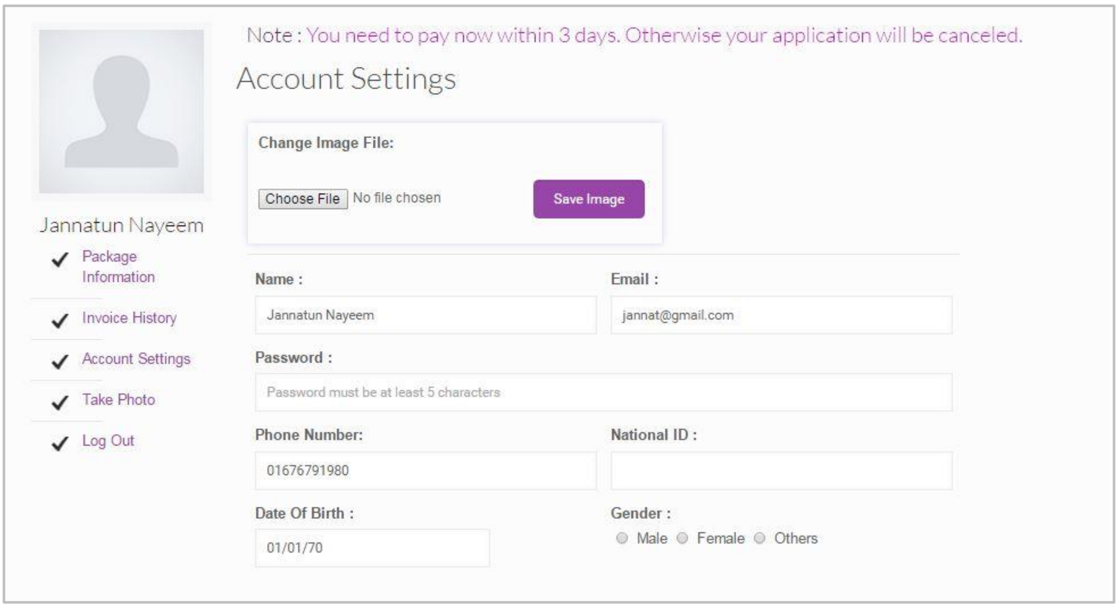
Package Duration: 3-Months

Shift: Morning

Discount: 20%

Fig. 6.4: Diagram of Package Details along with apply option.

User will have the option of his/her profile update:



Note : You need to pay now within 3 days. Otherwise your application will be canceled.

Account Settings

Change Image File:

Choose File No file chosen

Name : Jannatun Nayeem

Email : jannat@gmail.com

Password : Password must be at least 5 characters

Phone Number: 01676791980

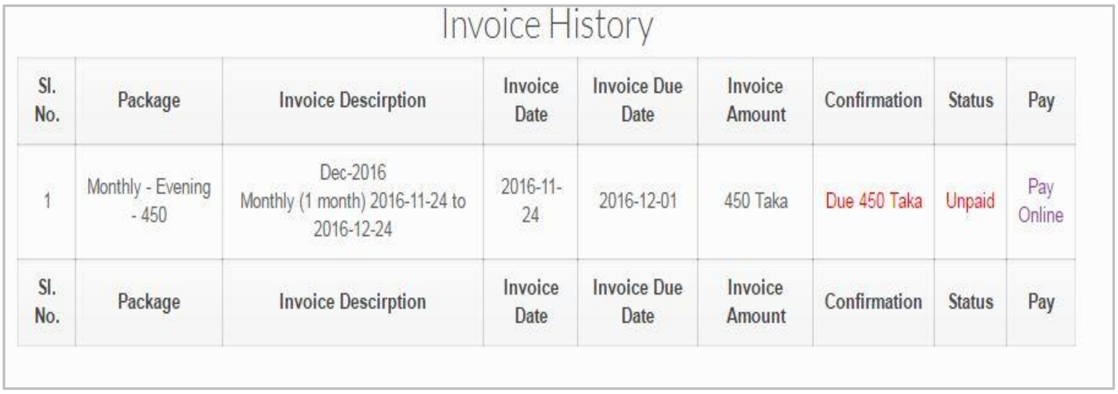
National ID :

Date Of Birth : 01/01/70

Gender : ○ Male ○ Female ○ Others

Fig. 6.5: Diagram of Account Settings Form.

User can see the invoice history they have:

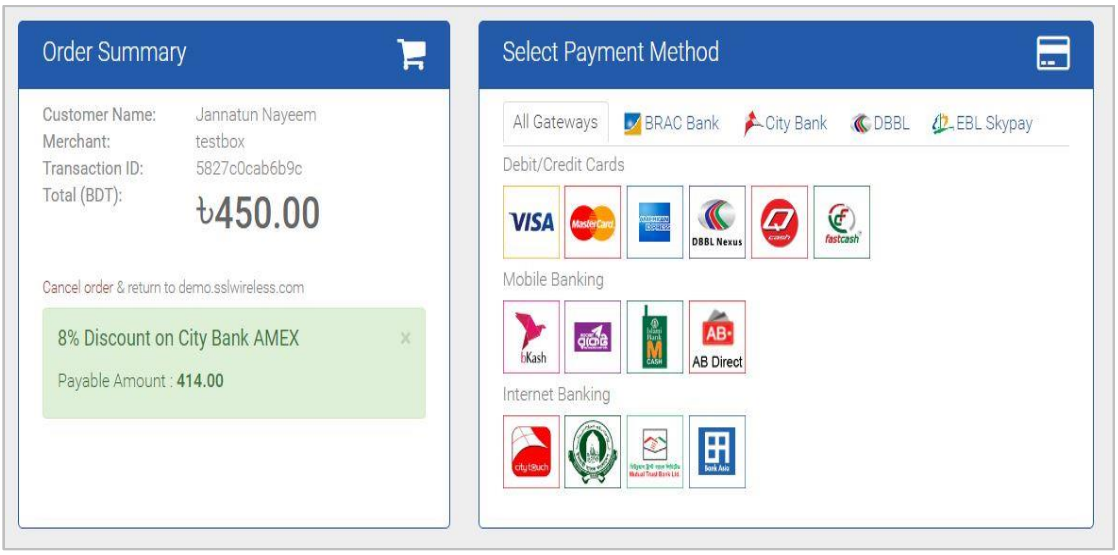


Invoice History

Sl. No.	Package	Invoice Descirption	Invoice Date	Invoice Due Date	Invoice Amount	Confirmation	Status	Pay
1	Monthly - Evening - 450	Dec-2016 Monthly (1 month) 2016-11-24 to 2016-12-24	2016-11-24	2016-12-01	450 Taka	Due 450 Taka	Unpaid	Pay Online
Sl. No.	Package	Invoice Descirption	Invoice Date	Invoice Due Date	Invoice Amount	Confirmation	Status	Pay

Fig. 6.6: Diagram of User’s Invoice History.

User can make online payment according to their generated invoice:



Order Summary

Customer Name: Jannatun Nayeem
Merchant: testbox
Transaction ID: 5827c0cab6b9c
Total (BDT): ৳450.00

Cancel order & return to demo.sslwireless.com

8% Discount on City Bank AMEX ×

Payable Amount : **414.00**

Select Payment Method

Fig. 6.7: Diagram of Online Payment Process

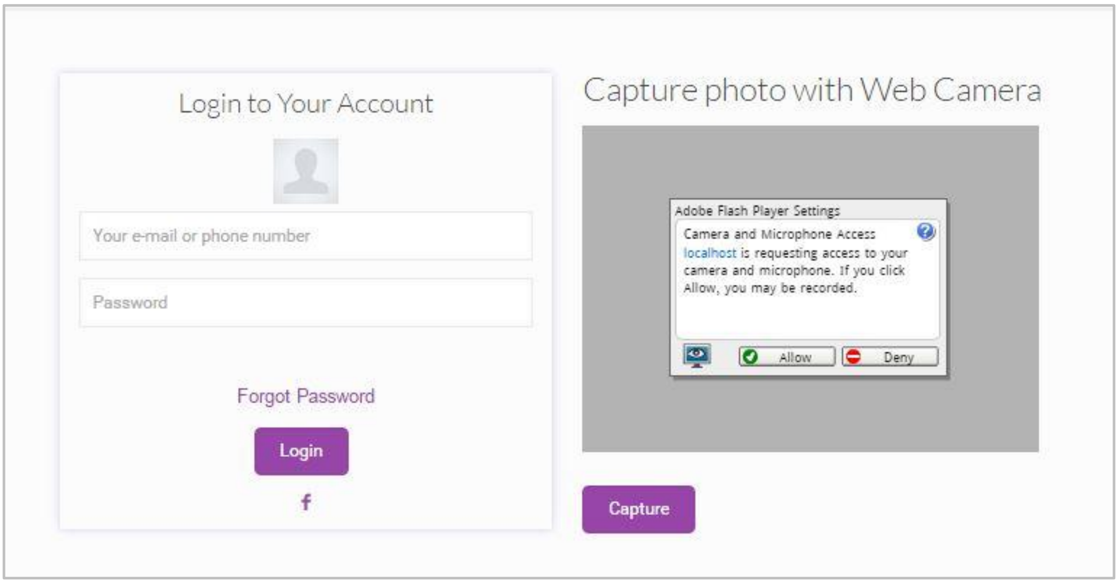
.

Our system has a new feature in login. User can login by image capturing by Webcam if he has any in the database that user uploaded previously.

User also has the option of login through email address or phone number along with the password. Our system then starts the authentication process in which the user email or phone number and password are matched with the existing information in the database. If the information matches, it is allowed to the main page else it warns the user for invalid information.

Here are screenshot of some features in user module: Login:

- Login by Face Recognition



Login to Your Account
Your e-mail or phone number
Password
Forgot Password
Login
f
Capture photo with Web Camera
Adobe Flash Player Settings
Allow
Deny
Capture

Camera and Microphone Access
localhost is requesting access to your camera and microphone. If you click Allow, you may be recorded.

Fig. 6.8: Diagram of login form along with webcam

- Login by Email and Password

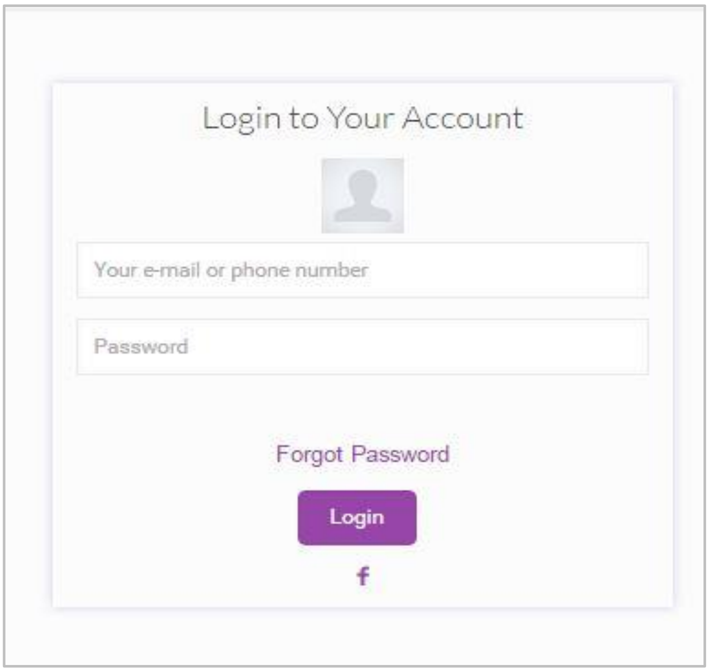


Fig. 6.9: Diagram of login form without face recognition process

User (Registered or Unregistered) can send their enquiry to admin :

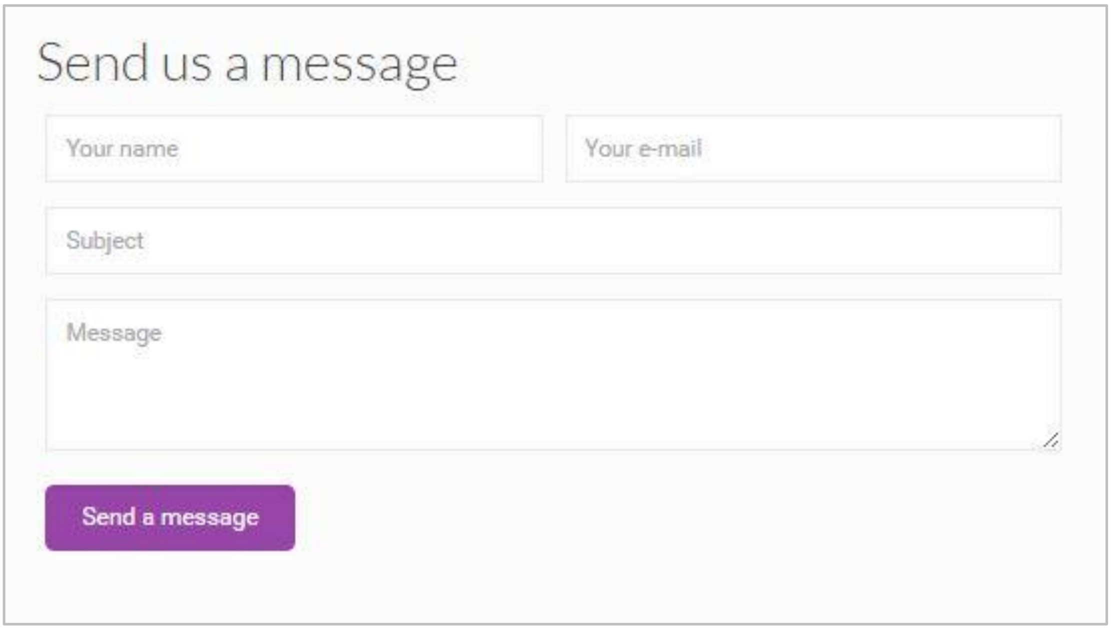
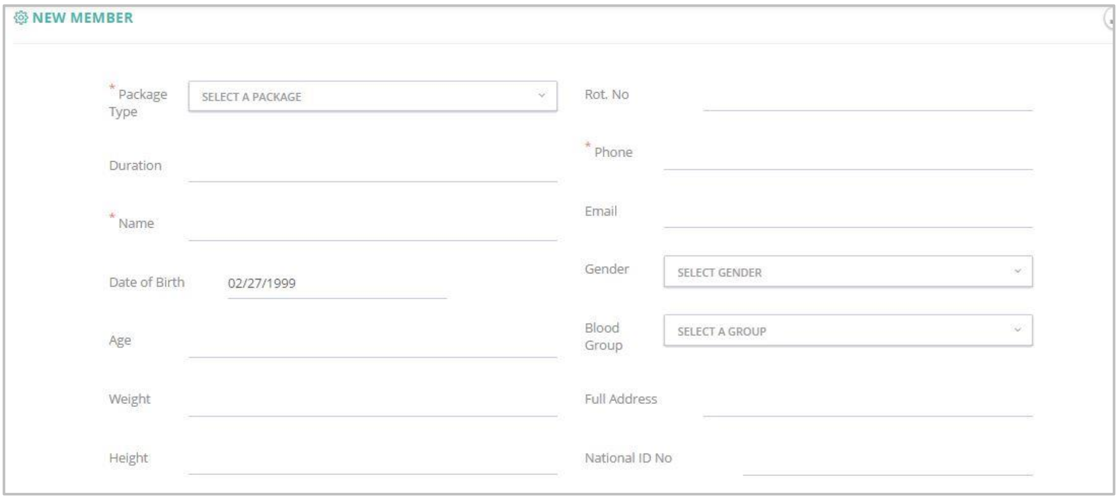


Fig. 6.10: Diagram of customer enquiry form.

6.2 Administrative Module:

Admin have control over all the features of our system. Some are :

- **Add member**

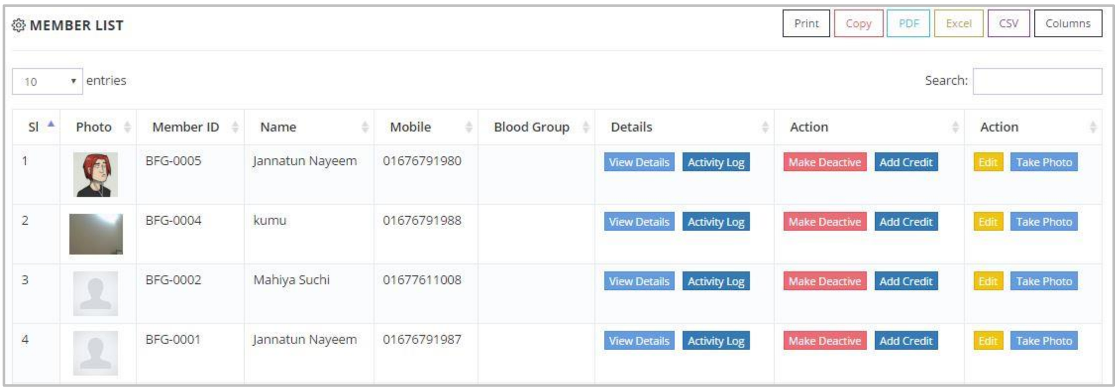


NEW MEMBER

* Package Type	SELECT A PACKAGE	Rot. No	
Duration		* Phone	
* Name		Email	
Date of Birth	02/27/1999	Gender	SELECT GENDER
Age		Blood Group	SELECT A GROUP
Weight		Full Address	
Height		National ID No	

Fig. 6.11: Diagram of member add form.

- **Manage Member**



MEMBER LIST

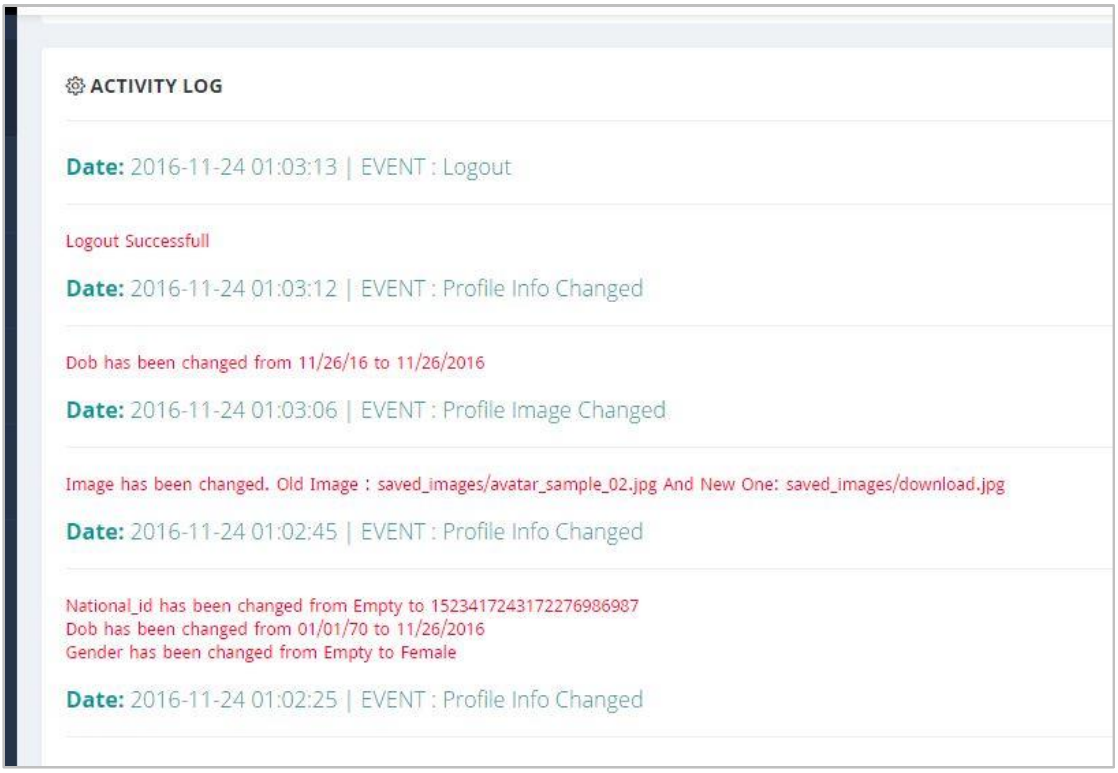
10 entries

Search:

Sl	Photo	Member ID	Name	Mobile	Blood Group	Details	Action	Action
1		BFG-0005	Jannatun Nayeem	01676791980		View Details Activity Log	Make Deactive Add Credit	Edit Take Photo
2		BFG-0004	kumu	01676791988		View Details Activity Log	Make Deactive Add Credit	Edit Take Photo
3		BFG-0002	Mahiya Suchi	01677611008		View Details Activity Log	Make Deactive Add Credit	Edit Take Photo
4		BFG-0001	Jannatun Nayeem	01676791987		View Details Activity Log	Make Deactive Add Credit	Edit Take Photo

Fig. 6.12: Diagram of manage member form.

➢ Activity log of a Member



ACTIVITY LOG

Date: 2016-11-24 01:03:13 | EVENT : Logout

Logout Successfull

Date: 2016-11-24 01:03:12 | EVENT : Profile Info Changed

Dob has been changed from 11/26/16 to 11/26/2016

Date: 2016-11-24 01:03:06 | EVENT : Profile Image Changed

Image has been changed. Old Image : saved_images/avatar_sample_02.jpg And New One: saved_images/download.jpg

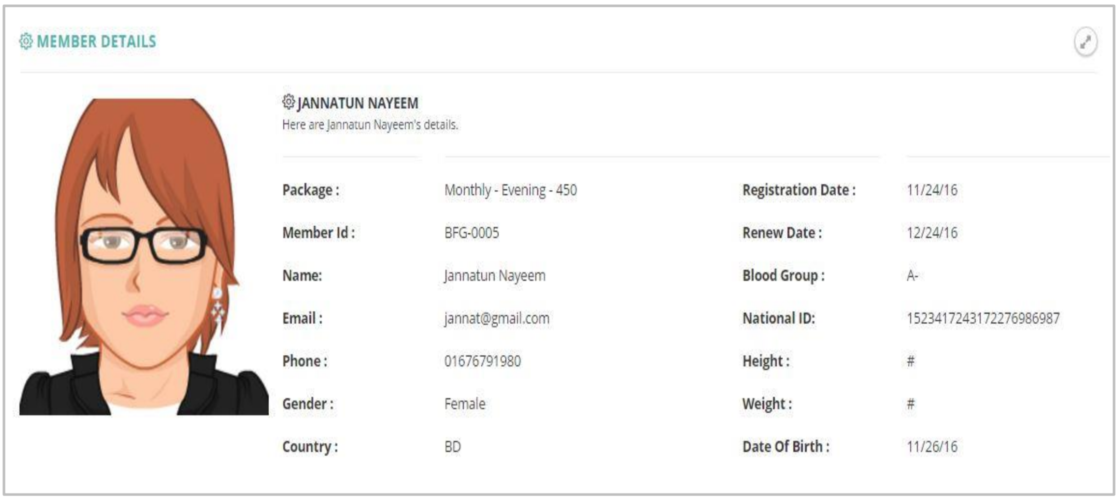
Date: 2016-11-24 01:02:45 | EVENT : Profile Info Changed

National_id has been changed from Empty to 1523417243172276986987
Dob has been changed from 01/01/70 to 11/26/2016
Gender has been changed from Empty to Female

Date: 2016-11-24 01:02:25 | EVENT : Profile Info Changed

Fig. 6.13: Diagram of activity log of a user.

➢ Member Details



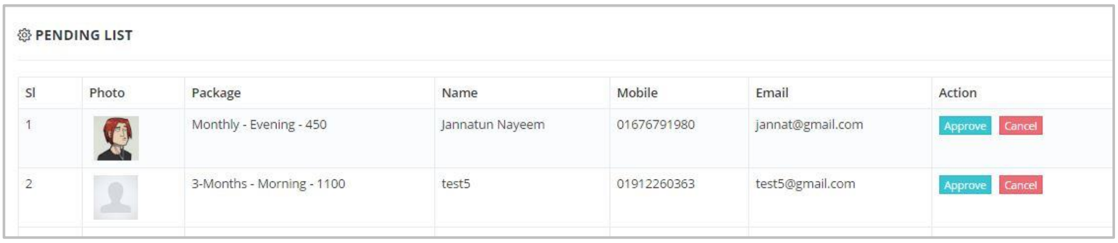
MEMBER DETAILS

JANNATUN NAYEEM
Here are Jannatun Nayeem's details.

Package :	Monthly - Evening - 450	Registration Date :	11/24/16
Member Id :	BFG-0005	Renew Date :	12/24/16
Name:	Jannatun Nayeem	Blood Group :	A-
Email :	jannat@gmail.com	National ID:	1523417243172276986987
Phone :	01676791980	Height :	#
Gender :	Female	Weight :	#
Country :	BD	Date Of Birth :	11/26/16

Fig. 6.14: Diagram of Member Details.

➢ Manage Pending Application

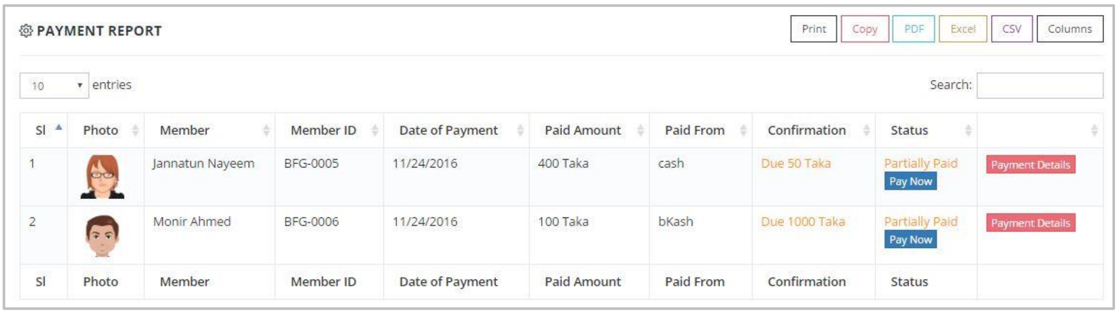


PENDING LIST

Sl	Photo	Package	Name	Mobile	Email	Action
1		Monthly - Evening - 450	Jannatun Nayeem	01676791980	jannat@gmail.com	Approve Cancel
2		3-Months - Morning - 1100	test5	01912260363	test5@gmail.com	Approve Cancel

Fig. 6.15: Diagram of pending applications.

➢ Admin can see the list of Payment Report



PAYMENT REPORT

Print Copy PDF Excel CSV Columns

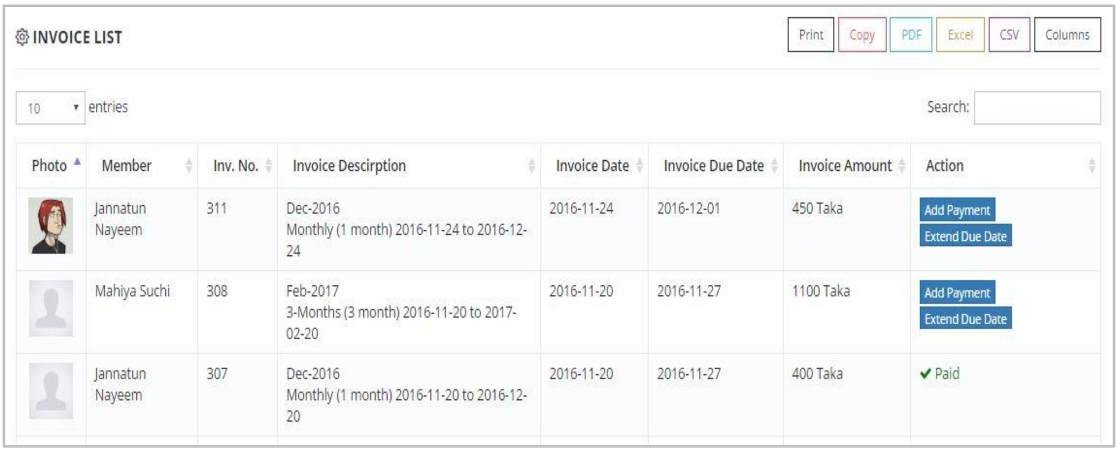
10 entries

Search:

Sl	Photo	Member	Member ID	Date of Payment	Paid Amount	Paid From	Confirmation	Status	
1		Jannatun Nayeem	BFG-0005	11/24/2016	400 Taka	cash	Due 50 Taka	Partially Paid Pay Now	Payment Details
2		Monir Ahmed	BFG-0006	11/24/2016	100 Taka	bKash	Due 1000 Taka	Partially Paid Pay Now	Payment Details
Sl	Photo	Member	Member ID	Date of Payment	Paid Amount	Paid From	Confirmation	Status	

Fig. 6.16: Diagram of member's payment report.

➢ Manage Invoice



INVOICE LIST

Print Copy PDF Excel CSV Columns

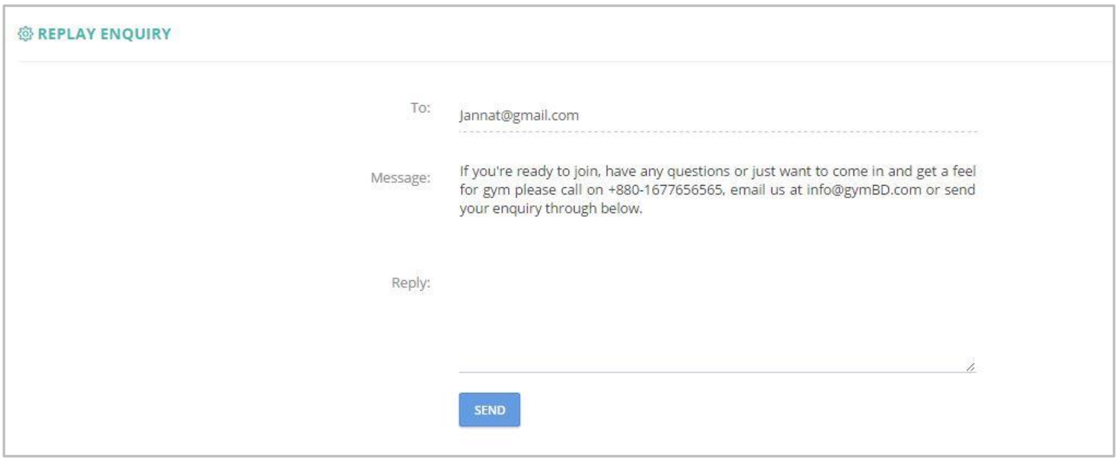
10 entries

Search:

Photo	Member	Inv. No.	Invoice Descirption	Invoice Date	Invoice Due Date	Invoice Amount	Action
	Jannatun Nayeem	311	Dec-2016 Monthly (1 month) 2016-11-24 to 2016-12-24	2016-11-24	2016-12-01	450 Taka	Add Payment Extend Due Date
	Mahiya Suchi	308	Feb-2017 3-Months (3 month) 2016-11-20 to 2017-02-20	2016-11-20	2016-11-27	1100 Taka	Add Payment Extend Due Date
	Jannatun Nayeem	307	Dec-2016 Monthly (1 month) 2016-11-20 to 2016-12-20	2016-11-20	2016-11-27	400 Taka	✓ Paid

Fig. 6.17: Diagram of member's invoice list.

- Admin can reply for user enquiry through email :



REPLAY ENQUIRY

To: Jannat@gmail.com

Message: If you're ready to join, have any questions or just want to come in and get a feel for gym please call on +880-1677656565, email us at info@gymBD.com or send your enquiry through below.

Reply:

Fig. 6.18: Diagram of user's enquiry reply form through email.

6.3 Testing

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Testing goes side by side with the implementation that is aimed at ensuring that the system works accurately and efficiently before the live operation is performed .The common view of testing held by the user is process of executing a program with explicit intention of handling errors. The application which has been developed has to be tested to prove its validity. Testing is considered to be the least creative phase of the whole cycle of system design. In the real sense it is the phase, which helps to bring out the creativity of the other phases, and makes it shine.

The Smart Gym Management System was tested using the following two techniques of application testing:

Unit Testing:

- In the line of strategy the entire individuals function and modules were put to test independently
- By following this strategy all the errors in coding were identified and corrected.
- This method was applied in combination with the White Box and Black Box testing

- Technique to find errors in each module.
- The effort of specific combination of data on system operation was tested.
- The following were the testes carried out for Graphical User Interface (GUI).
- It was seen that the pages opens properly based on related menu based commands.
- It was tested whether all relevant menus, buttons, icons and other controls are available and properly displayed.

System Testing

We use this testing method. System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Multi-user System Testing

Database Locking Schemes: Whenever more than one person is accessing a record/s some type of process must be used to prevent the outer users from attempting to update the same record at the same time. This process is a locking scheme. In its simplest form, a locking scheme allows only one user at a time to update information in the database.

Chapter 7

7.1 Future Work

The project has been developed in a very short period of time and all efforts have been taken so that this project is very efficient in its execution there still exists some scope of improvement in our project. The following lists some of the enhancement that can be added incorporate into the project.

Application of the project can be done more attractively. Database management and all maintenance module can be updated which helps the administrator. More security measures can be taken.

There are also few features which can be integrated with this system to make it more flexible. Below list shows the future points to be consider:

- Real-time Chat BOT option for members and trainer, so that members can directly enquiry theirs trainer on any time through the Chat BOT.
- Automated Fitness suggestion by enquiring the condition of the health.
- Real time Claim Processing Bot.
- Video conversation option for trainers and members.
- Online payment through face recognition.
- Barcode generation for membership card and using this, members can take entry to Gym.
- Finger print matching for taking entry to gym.

.

7.2 Conclusions

The “**SMART GYM MANAGEMENT SYSTEM**” is successfully designed and developed to fulfilling the necessary requirements, as identified in the requirements analysis phase, such as the system is very much user friendly, form level validation and field level validation are performing very efficiently. The old manual system was suffering from a series of drawbacks. The present project has been developed to meet the aspirations indicated in the modern age.

References

[1] www.wikipedia.com

[2] www.tutorialspoint.com

[3] https://www.apachefriends.org/download.html

[4] https://jquery.com

[5] http://www.slideshare.net/jagaarj/database-design-normalization

[6] https://generic.wordpress.soton.ac.uk/spidersocial/2016/04/02/face-recognition/

[7] https://www.twilio.com/docs/quickstart/php/sms

[8] https://www.twilio.com/docs/quickstart/php/sms/sending-via-rest

[9] https://www.kairos.com/face-recognition-api

[10] https://sslcommerz.com/

[11] https://stripe.com/docs/custom-form

Appendix: Sample Code Of ‘Smart gym Management System’.

Code for sending sms for phone number verification :

```
// START
<?php
// Require the bundled autoload file - the path may need to change
// based on where you downloaded and unzipped the SDK
require './twilio-php-master/Twilio/autoload.php';
// Use the REST API Client to make requests to the Twilio REST API
use Twilio\Rest\Client;
if(!isset($_SESSION['demo'])){
// Your Account SID and Auth Token from twilio.com/console
$sid = 'your sid';
$token = 'your auth token';
$url = "https://api.twilio.com/2010-04-01/Accounts/$sid/SMS/Messages";
$from = "+1 910-225-7400";
$to = "+8801676791987"; // twilio trial verified number
$digits =4;
$random_number = rand(pow(10, $digits-1), pow(10, $digits)-1);
$body = "Please Enter this verification code : ".$random_number;
$data = array (
   'From' => $from,
   'To' => $to,
   'Body' => $body,
);
$post = http_build_query($data);
$x = curl_init($url );
curl_setopt($x, CURLOPT_POST, true);
curl_setopt($x, CURLOPT_RETURNTRANSFER, true);
curl_setopt($x, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($x, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($x, CURLOPT_USERPWD, "$sid:$token");
curl_setopt($x, CURLOPT_POSTFIELDS, $post);
$y = curl_exec($x);
curl_close($x);
//var_dump($post);
//var_dump($y);
}
?>
```

Code For Webcam Integration :

```
<script>
  $(function () {
    //give the php file path
    webcam.set_api_url('save_login_image.php');
    webcam.set_swf_url('webcam/webcam.swf');//flash file (SWF) file path
    webcam.set_quality(100); // Image quality (1 - 100)
    webcam.set_shutter_sound(true); // play shutter click sound

    var camera = $('#camera');
    camera.html(webcam.get_html(350, 250));
    $('#capture_btn').click(function () {
      //take snap
      webcam.snap();
      //$('#show_saved_img').html('<h3>Please Wait...</h3>');
    })
```

```
       //after taking snap call show image
        webcam.set_hook('onComplete', function (img) {
            $('#user_image').attr('src', img);
            $('#user_image').css('height', '250px');
            $('#user_image').css('width', '250px');
            $('#img_src').val(img);
            $('#login_form').hide();
            $('#img_login_form').show();
            webcam.reset();
        });
 });
</script>
```

Code For image upload and Enroll image in Kairos for Face Recognition :

```
<?php
require_once('../zarsss_files/auth.php');
require_once('../zarsss_files/db.config.php');
require_once('../Kairos-SDK-PHP-master/Kairos.php');
require_once('../Kairos-SDK-PHP-master/examples/assets/helper.php');
require_once('../classes/welcome_model.php');
?>
<?php
extract($_POST);
if (is_array($_FILES)) {
 if (is_uploaded_file($_FILES['userImage']['tmp_name'])) {
        $uploadOk = 1;
        $target_file = $_FILES['userImage']['name'];
        $imageFileType = pathinfo($target_file, PATHINFO_EXTENSION);
        $check = getimagesize($_FILES["userImage"]["tmp_name"]);

        if ($check !== false) {

            //echo "File is an image - " . $check["mime"] . ".";

            $uploadOk = 1;

            if ($_FILES["userImage"]["size"] > 50000000) {

                echo "Sorry, your file is too large.";

                $uploadOk = 0;

            } else {

  if ($imageFileType != "jpg" && $imageFileType != 'JPG' && $imageFileType !=
"png" && $imageFileType != "PNG" && $imageFileType != "jpeg" &&
$imageFileType != "JPEG" && $imageFileType != "gif" && $imageFileType !=
"GIF") {

                    echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";

                    $uploadOk = 0;

                }

            }

        } else if ($check === false) {

            echo "Your File is not an image.";

            $uploadOk = 0;

        }

 if ($uploadOk !== 0) {

            $sourcePath = $_FILES['userImage']['tmp_name'];

            $targetPath ="../admin_panel/saved_images/" . $_FILES['userImage']['name'];

             if (move_uploaded_file($sourcePath, $targetPath)) {

                $targetPath = explode('../admin_panel/', $targetPath);

                $wl_mod->save_image_change_log($user_id, $targetPath[1], $old_photo);

                $wl_mod->upload_image_by_id($user_id, $targetPath[1]);
                echo $targetPath[1];

                $path = '../admin_panel/'.$targetPath[1];

//                echo $path;

//                exit();

                /* Start Enrolling Image to Kairo Gallary */


                //* * * * sample api credentials (works for example)

                $app_id = '09dcb808';

                $api_key = '70173325407b81d4b681e06fce56e875';


                //* * * * create a new instance and authenticate

                $Kairos = new Kairos($app_id, $api_key);

                /*

                In this example, we enroll a subject into a gallery using a

                path to an image file. */

                $gallery_id = 'gymBd';

                $subject_id = $user_id;

                $image_path = $path;

                $response   =   $Kairos->enrollImageWithPath($image_path,   $gallery_id,
              $subject_id);

               /* End Enrolling Image to Kairo Gallary */


              }

          }

      }

}

?>
```

Code For Face Recognition Process after capturing image :

```
require_once('./Kairos-SDK-PHP-master/Kairos.php');

require_once('./Kairos-SDK-PHP-master/examples/assets/helper.php');

extract($_GET);

extract($_POST);

//* * * * sample api credentials (works for example)

$app_id = '09dcb808';

$api_key = '70173325407b81d4b681e06fce56e875';

//* * * * create a new instance and authenticate

$Kairos = new Kairos($app_id, $api_key);

$gallery_id = 'gymBd';

$path = trim($img_src);

// Start View Subjects in Gallery.

//$response = $Kairos->listSubjectsForGallery($gallery_id);

//echo '<pre>';

//print_r(json_decode($response));

//exit();

// End View Subjects in Gallery.

//print_r($path);

//exit();

//$response = removeGallery($gallery_id);exit();

$response = $Kairos->recognizeImageWithPath($path, $gallery_id);

//echo '<pre>';

//print_r(json_decode($response));

//exit();

$result = json_decode($response);
```

```
// START No Face Found Error
$error_check = isset($result->Errors);
//$error_check = isset($result->Errors);
//if($error_check == FALSE){
//    echo "False Ca" . $error_check . "<br/>";
//}
//print_r($result);
//print_r($result->Errors[0]->Message);
//exit();
// END No Face Found Error
//print_r($result->images[0]->transaction);
$face_recognition_result = array();
if ($error_check == FALSE) {
    foreach ($result as $v_result) {
        foreach ($v_result as $face_result) {
            foreach ($face_result as $recognition_result) {
                $face_recognition_result = $recognition_result;
                break;
            }
        }
    }
// Start Face Found Result Show
//    echo '<pre>';
//    print_r($result);
//    print_r($face_recognition_result);
//    exit();
// End Face Found Result Show
```

```
$status = $face_recognition_result->status;
$uid = $face_recognition_result->subject_id;
 if ($status === 'success') {
   $sql = "SELECT * FROM m_members WHERE Id='$uid'";
   $stmt = $conn->prepare($sql);
   $stmt->execute();
   $stmt->setFetchMode(PDO::FETCH_ASSOC);
   $row = $stmt->fetch();
   $_SESSION['name'] = $row['name'];
   $_SESSION['email'] = $row['email'];
   $_SESSION['cell_number'] = $row['cell_number'];
   $_SESSION['Id'] = $row['Id'];
   /*
    * Activity Log   */
   $data = array();
   $current_date_formated = date("Y-m-d G:i:s", time());
   $data['activity_date'] = date(strtotime($current_date_formated));
   $data['event'] = "Login";
   $data['member_id'] = $_SESSION['Id'];
   $data['log_details'] = "Login Successfull";
   $data['ip_address'] = $_SERVER['REMOTE_ADDR'];
   $wl_mod->save_login_activity_log($data);
   /*
    * END
    */
```

```
        unlink($path);

        header('Location: user_profile.php');

    } else {

        unlink($path);

        $_SESSION['exception'] = 'You Gave Invalid Information. Face Not Matched.';

        header('Location: login.php');

    }

} else {

    //print_r($result->Errors[0]->Message);

    unlink($path);

    $_SESSION['exception'] = ucwords("sorry!! " . $result->Errors[0]->Message . ". try again.");

    header('Location: login.php');

}

?>
```

Classes for different functionality :

Billing_model Class :

```
<?php


class Billing_Model {

    public $conn;

    public function get_due_balance_by_member_id($uid) {

        $sql_fst = "SELECT MAX(Id) FROM m_invoice WHERE member_id='$uid'";

        $result_fst = $this->conn->query($sql_fst);

        $row_fst = $result_fst->fetch(PDO::FETCH_ASSOC);

        $max_inv_id = $row_fst['MAX(Id)'];
    if ($max_inv_id) {
      $sql = "SELECT MAX(Id) FROM m_payment WHERE invoice_id = '$max_inv_id'";
      $stmt = $this->conn->prepare($sql);
      $stmt->execute();
      $stmt->setFetchMode(PDO::FETCH_ASSOC);
      $row = $stmt->fetch();
      $payment_id = $row['MAX(Id)'];
      //echo "payment ID :".$payment_id."<br/>ENDALL";
      if ($payment_id) {
        //echo "Payment 2 :".$payment_id;
        $sql = "SELECT * FROM m_payment WHERE invoice_id = '$max_inv_id' AND Id = '$payment_id'";
       $stmt = $this->conn->prepare($sql);
        $stmt->execute();
        $stmt->setFetchMode(PDO::FETCH_ASSOC);
        $row = $stmt->fetch();
        return $row;
      }
    }  else {
      return FALSE;
    }
  }
public function create_invoice() {
    // Start Set TimeZone
    date_default_timezone_set('Asia/Dhaka');
    $current_date_formated = date("Y-m-d G:i:s", time());
    $timestamp = date(strtotime($current_date_formated));
    // End Set Timezone
```

```
$sql = "SELECT * FROM m_members WHERE active_status=1";
$stmt = $this->conn->prepare($sql);
$stmt->execute();
$stmt->setFetchMode(PDO::FETCH_ASSOC);
$member_info = $stmt->fetchAll();

foreach ($member_info as $member_info_data) {
    $uid = $member_info_data['Id'];
    $member_id = $member_info_data['member_id'];
    $reg_date = $member_info_data['reg_date'];
    $renew_date = $member_info_data['renew_date'];
    $package_id = $member_info_data['package_id'];
    $sql = "SELECT * FROM m_package WHERE Id = '$package_id'";
    $stmt = $this->conn->prepare($sql);
    $stmt->execute();
    $stmt->setFetchMode(PDO::FETCH_ASSOC);
    $package_info = $stmt->fetch();
    $cycle_title = $package_info['cycle_title'];
    $total_month = $package_info['total_month'];
     $status = 0;
    $charge = $package_info['charge'];
   // Start GET Credit and Due Balance of Member
    $due_balance = $this->get_due_balance_by_member_id($uid);
    // End GET Credit and Due Balance of Member
```

```
 if ($due_balance) {
            $charge = $charge + $credit_n_due_balance['amount_due'];
        }
        $credit_balance = $member_info_data['credit_balance'];
         if($credit_balance < $charge){
            $charge = $charge - $credit_balance;
            $credit_balance = 0;
            $sql = "UPDATE m_members SET credit_balance = '$credit_balance'
WHERE Id = '$uid'";
            $this->conn->exec($sql);
        }else if($credit_balance >= $charge){
            $credit_balance = $credit_balance - $charge;
            $charge = 0;
            $status = 4;
            $sql = "UPDATE m_members SET credit_balance = '$credit_balance'
WHERE Id = '$uid'";
            $this->conn->exec($sql);
        }

        $shift_id = $package_info['shift_id'];

         //echo $cycle_title;
         // Start Create Invoice

         if ($reg_date > $renew_date) {
            $reg_date_title = date("Y-m-d", $reg_date);
            $renew_until = "$total_month month";
            $renew_date_new=date('Y-m-d',strtotime($renew_until,
strtotime($reg_date_title)));
```

```
            $new_renew_date_set = strtotime($renew_date_new);

            // Start Renew Date update of member with Id=uid

            $sql = "UPDATE m_members SET renew_date = '$new_renew_date_set'
WHERE Id = '$uid'";

            $this->conn->exec($sql);

           // End Renew Date update of member with Id=uid

            // Start Insert Invoice

            $invoice_due_date = $reg_date + 604800;

            $billing_month = date("M-Y", $new_renew_date_set);

            $desc_notes = "$cycle_title ($renew_until) $reg_date_title to
$renew_date_new ";

            $created_date = date("m/d/y", time());

            $created_timestamp = date(strtotime($created_date));

            $new_invoice = "INSERT INTO
m_invoice(member_id,billing_month,desc_notes,invoice_date,"
"created_date,invoice_duedate,invoice_amount,invoice_payable,date_time," . "status)
VALUES('$uid','$billing_month','$desc_notes','$reg_date',"

                  .
"'$created_timestamp','$invoice_due_date','$charge','$charge','$timestamp','$status')";

            $this->conn->exec($new_invoice);

          // End Insert Invoice

         }

         if ($timestamp > $renew_date && $renew_date != 0) {

            $current_renew_date_title = date("Y-m-d", $renew_date);

            $renew_until_1 = "$total_month month";

            $renew_date_new_1 = date('Y-m-d', strtotime($renew_until_1,
strtotime($current_renew_date_title)));

            $new_renew_date_set_1 = strtotime($renew_date_new_1);
            // Start Renew Date update of member with Id=uid

         $sql = "UPDATE m_members SET renew_date = '$new_renew_date_set_1'
WHERE Id = '$uid'";

            $this->conn->exec($sql);

         // End Renew Date update of member with Id=uid

            $invoice_due_date = $renew_date + 604800;

            $billing_month = date("M-Y", $new_renew_date_set_1);

            $desc_notes = "$cycle_title ($renew_until_1) $current_renew_date_title to
$renew_date_new_1 ";

            $created_date = date("m/d/y", time());

            $created_timestamp = date(strtotime($created_date));

            $new_invoice = "INSERT INTO
m_invoice(member_id,billing_month,desc_notes,invoice_date,"

."created_date,invoice_duedate,invoice_amount,invoice_payable,date_time,"
."status)VALUES('$uid','$billing_month','$desc_notes','$renew_date',".

"'$created_timestamp','$invoice_due_date','$charge','$charge','$timestamp','$status')";

            $this->conn->exec($new_invoice);

  // End Insert Invoice

         }


       }

    }


}


$bl_mod = new Billing_Model();

$bl_mod->conn = $conn;
```

Admin_model Class :

```
<?php
class Admin_Model {
 public $conn;
 public function check_unique_email_address($email) {
        $sql = "SELECT * FROM m_members WHERE email = '$email' ";
        $stmt = $this->conn->prepare($sql);
        $stmt->execute();
        $stmt->setFetchMode(PDO::FETCH_ASSOC);
        $row = $stmt->fetch();
        return $row;
    }
public function check_unique_phone_number($phone) {
        $sql = "SELECT * FROM m_members WHERE cell_number = '$phone' ";
        $stmt = $this->conn->prepare($sql);
        $stmt->execute();
        $stmt->setFetchMode(PDO::FETCH_ASSOC);
        $row = $stmt->fetch();
        return $row;
    }
public function deactivate_user($user_id) {
        $sql = "UPDATE m_members SET active_status=0 WHERE Id='$user_id'";
        $this->conn->query($sql);
    }
public function activate_user($user_id) {
        $sql = "UPDATE m_members SET active_status=1 WHERE Id='$user_id'";
        $this->conn->query($sql);
    }
```

```
public function get_total_member() {
        $sql = "SELECT COUNT(*) FROM m_members WHERE approval_status=1";
        $result = $this->conn->query($sql);
        $row = $result->fetch(PDO::FETCH_ASSOC);
        return $row;
    }
public function get_total_invoice_collection() {
        $sql = "SELECT SUM(amount) FROM m_payment";
        $result = $this->conn->query($sql);
        $row = $result->fetch(PDO::FETCH_ASSOC);
        return $row;
    }
public function get_daily_invoice_collection() {
        $current_date_formated = date("m/d/y", time());
        $timestamp = date(strtotime($current_date_formated));
        $sql = "SELECT SUM(amount) FROM m_payment WHERE
pay_date='$timestamp'";
        $result = $this->conn->query($sql);
        $row = $result->fetch(PDO::FETCH_ASSOC);
        return $row;
    }
public function get_total_due_payment() {
        $total_amount_due = 0;
        $sql = "SELECT Id FROM m_invoice";
        $stmt = $this->conn->prepare($sql);
        $stmt->execute();
        $stmt->setFetchMode(PDO::FETCH_ASSOC);
        $row = $stmt->fetchAll();
        foreach ($row as $inv_id) {

            $sql = "SELECT MAX(Id) FROM m_payment WHERE invoice_id =
'$inv_id[Id]'";

            $stmt = $this->conn->prepare($sql);

            $stmt->execute();

            $stmt->setFetchMode(PDO::FETCH_ASSOC);

            $py_row = $stmt->fetch();

            $payment_id = $py_row['MAX(Id)'];


            $sql = "SELECT amount_due FROM m_payment WHERE Id =
'$payment_id'";

            $stmt = $this->conn->prepare($sql);

            $stmt->execute();

            $stmt->setFetchMode(PDO::FETCH_ASSOC);

            $am_du_row = $stmt->fetch();

            $total_amount_due += $am_du_row['amount_due'];

        }

   $sql = "SELECT SUM(invoice_amount) FROM m_invoice WHERE status = 0";

        $stmt = $this->conn->prepare($sql);

        $stmt->execute();

        $stmt->setFetchMode(PDO::FETCH_ASSOC);

        $unpaid = $stmt->fetch();

        $total_amount_due += $unpaid['SUM(invoice_amount)'];

        return $total_amount_due;

    }

public function get_total_active_members() {

        $sql = "SELECT COUNT(*) FROM m_members WHERE active_status=1";

        $result = $this->conn->query($sql);

        $row = $result->fetch(PDO::FETCH_ASSOC);
        return $row;
    }
public function check_email_address($email) {
        $sql = "SELECT * FROM m_admin WHERE email = '$email' ";
        $stmt = $this->conn->prepare($sql);
        $stmt->execute();
        $stmt->setFetchMode(PDO::FETCH_ASSOC);
        $row = $stmt->fetch();
        return $row;
    }
public function get_todays_generated_invoice() {
        $current_date_formated = date("m/d/y", time());
        $timestamp = date(strtotime($current_date_formated));
        $sql = "SELECT SUM(invoice_amount) FROM m_invoice WHERE created_date='$timestamp'";
        $result = $this->conn->query($sql);
        $row = $result->fetch(PDO::FETCH_ASSOC);
        return $row;
    }
public function get_new_admitted_members_today() {
        $current_date_formated = date("m/d/y", time());
        $timestamp = date(strtotime($current_date_formated));
        $sql = "SELECT COUNT(*) FROM m_members WHERE reg_date='$timestamp'";
        $result = $this->conn->query($sql);
        $row = $result->fetch(PDO::FETCH_ASSOC);
        return $row;
    }
```

```
 public function get_total_pending_application() {
       $sql = "SELECT COUNT(*) FROM m_members WHERE approval_status=0";
        $result = $this->conn->query($sql);
        $row = $result->fetch(PDO::FETCH_ASSOC);
        return $row;
    }
 public function send_email($row) {
       $name = $row['fullname'];
        //echo $name;exit();
        $to = $row['email'];
        $subject = "Reset Password";
        $encoded_name = urlencode($name);
        //echo $encoded_name;exit();

$path="http://zarss.online/bodyflex/admin_2/mailer_script/password_reset_email.php
?reset_name=$encoded_name&reset_email=$to";
      $message = file_get_contents($path);
    // Always set content-type when sending HTML email
        $headers = "MIME-Version: 1.0" . "\r\n";
        $headers .= "Content-type:text/html;charset=UTF-8" . "\r\n";
        // More headers
        $headers .= 'From: <kumu@zarss.online>' . "\r\n";
        $headers .= 'Cc: monir.ahmed0993@gmail.com' . "\r\n";
        mail($to, $subject, $message, $headers);
    }
```

```
public function send_replay_email() {

        extract($_POST);

        // update m_message

        $sql = "UPDATE m_message SET replay_message = :replay_message WHERE
Id = :Id";

        $prepare_sql = $this->conn->prepare($sql);

        $prepare_sql->execute(array(

            ':replay_message' => $replay_message,

            ':Id' => $Id

        ));

        $to = $email;

        $subject = "GymBd Replay";

        $encoded_name = urlencode($name);

        $encoded_replay = urlencode($replay_message);


$path="http://localhost/gymBD/admin_panel/mailer_script/replay_email.php?reset_n
ame=$encoded_name&replay_message=$encoded_replay";

        $message = file_get_contents($path);

        // Always set content-type when sending HTML email

        $headers = "MIME-Version: 1.0" . "\r\n";

        $headers .= "Content-type:text/html;charset=UTF-8" . "\r\n";

       // More headers

        $headers .= 'From: <kumu@zarss.online>' . "\r\n";

        $headers .= 'Cc: monir.ahmed0993@gmail.com' . "\r\n";

        mail($to, $subject, $message, $headers);

        $_SESSION['complete_message'] = "Your Replay has been Sent.";

        header("Location: reply_message.php");

    }
    public function delete_message_by_id($Id) {
      $sql = "DELETE FROM m_message WHERE Id='$Id'";
      $this->conn->exec($sql);
   }
public function update_password_by_email() {
      extract($_POST);
      $Encrypted_Password = password_hash($password, PASSWORD_DEFAULT);
      $sql = "UPDATE m_admin SET passwd = '$Encrypted_Password' WHERE email = '$email' ";
      $this->conn->exec($sql);
   }
}
$ad_mod = new Admin_Model();
$ad_mod->conn = $conn;
```

A Project Paper on Smart Gym Management System

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
EAST WEST UNIVERSITY

Fall 2016