



QR Marks the Spot@MDH Design Description

Version 1.1

Project Name	Version: 1.1
Design Description	Date: 2009-10-10

Revision History

Date	Version	Description	Author
2009-10-05	1.0	Initial Draft	Sriram Sundar Rajan Soumya Kanti Chakraborty Mohd. Siblee Islam
2009-10-10	1.1	Updated UI Designs	Soumya Kanti Chakraborty

Project Name	Version: 1.1
Design Description	Date: 2009-10-10

Table of Contents

1.	Introduction	4
1.1	Purpose of this document	4
1.2	Intended Audience	4
1.3	Scope	4
1.4	Definitions and acronyms	4
1.4.1	Definitions	4
1.4.2	Acronyms and abbreviations	4
1.5	References	4
2.	External interfaces	5
3.	Software architecture	7
3.1	Conceptual design	7
3.2	System specification	8
3.3	Error handling	9
4.	Detailed software design	10
4.1	Database Design	10
4.2	Class Diagram	11
5.	Approvals	12

1. Introduction

1.1 Purpose of this document

The purpose of the document is to provide details of high level system architecture, overview of system specifications and layouts of classes and databases.

1.2 Intended Audience

The document is intended for the following audience:

Customer : To understand overall architecture and technologies involved.

Project Supervisor : To validate the system design

Project Team : To understand the system architecture , purpose of various classes and database tables and the relationship among them.

1.3 Scope

The scope of the document is to provide details of the high level architecture of the proposed system.

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
PHP	Hypertext Preprocessor
MySQL	My Structured Query Language
LAMP	Linux Apache MySQL PHP
QR	Quick Response
MVC	Model View Controller
AJAX	Asynchronous JavaScript
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator
API	Application Programming Interface

1.5 References

CakePHP Framework - <http://cakephp.org/>

QR Reader - <http://reader.kaywa.com/>

2. External interfaces

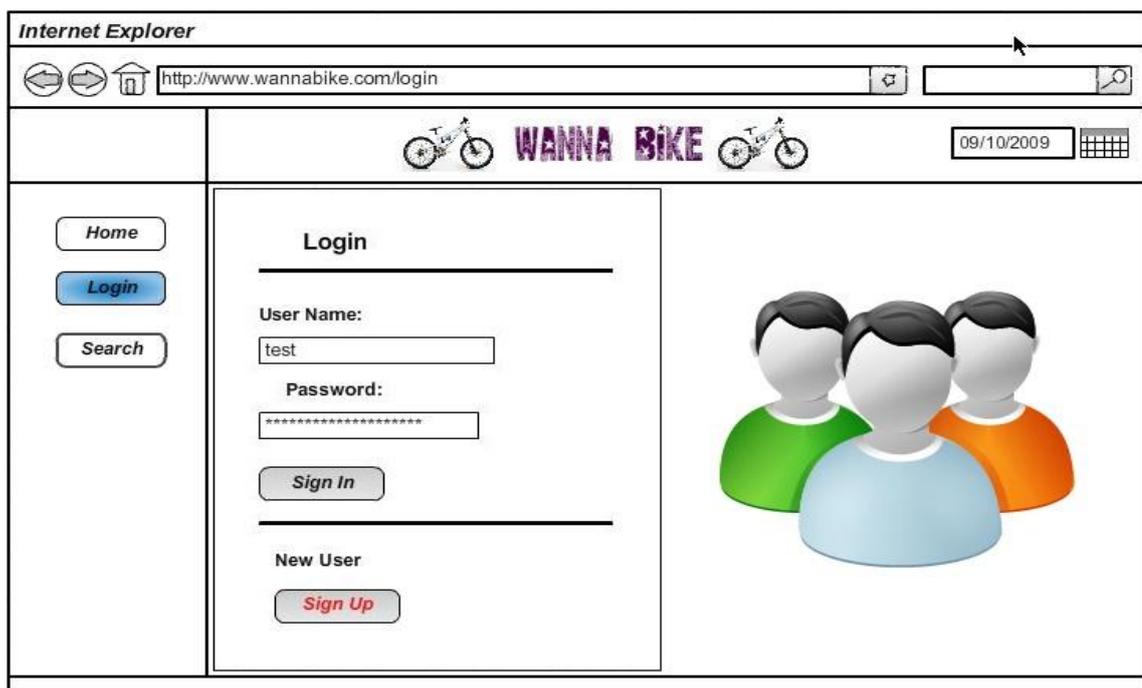
As the proposed solution is a website, HTML along with PHP and JavaScript is used to design the web pages. Adobe Dreamweaver is used to design the layout and look & feel of the web page. As the website is mostly accessed from mobile, it is ensured that the website is also WAP Enabled.

Below are the snapshots of the proposed UI Design

1) Search Bikes



2) User Login



3) Registered User

Internet Explorer

http://www.wannabike.com/Home

09/10/2009

WANNA BIKE

Manage Password | Trace History | Maintain Personal Data

Welcome , Test

Search

Profile Management

Mobile Access

Register Bikes

Forums

Log Out

Enter Current Password

Enter New Password

Confirm New Password

Submit Cancel



4) Administrator

Internet Explorer

http://www.wannabike.com/Admin_Home

09/10/2009

WANNA BIKE

Register Bikes | Update Bike Location | Approve Bike Request

Welcome , ADMIN

Bike Management

Assign & Manage QR Codes

User Account Management

Forums Management

Log Out

Enter Bike Model

Enter the Bike Number

Enter Present Location

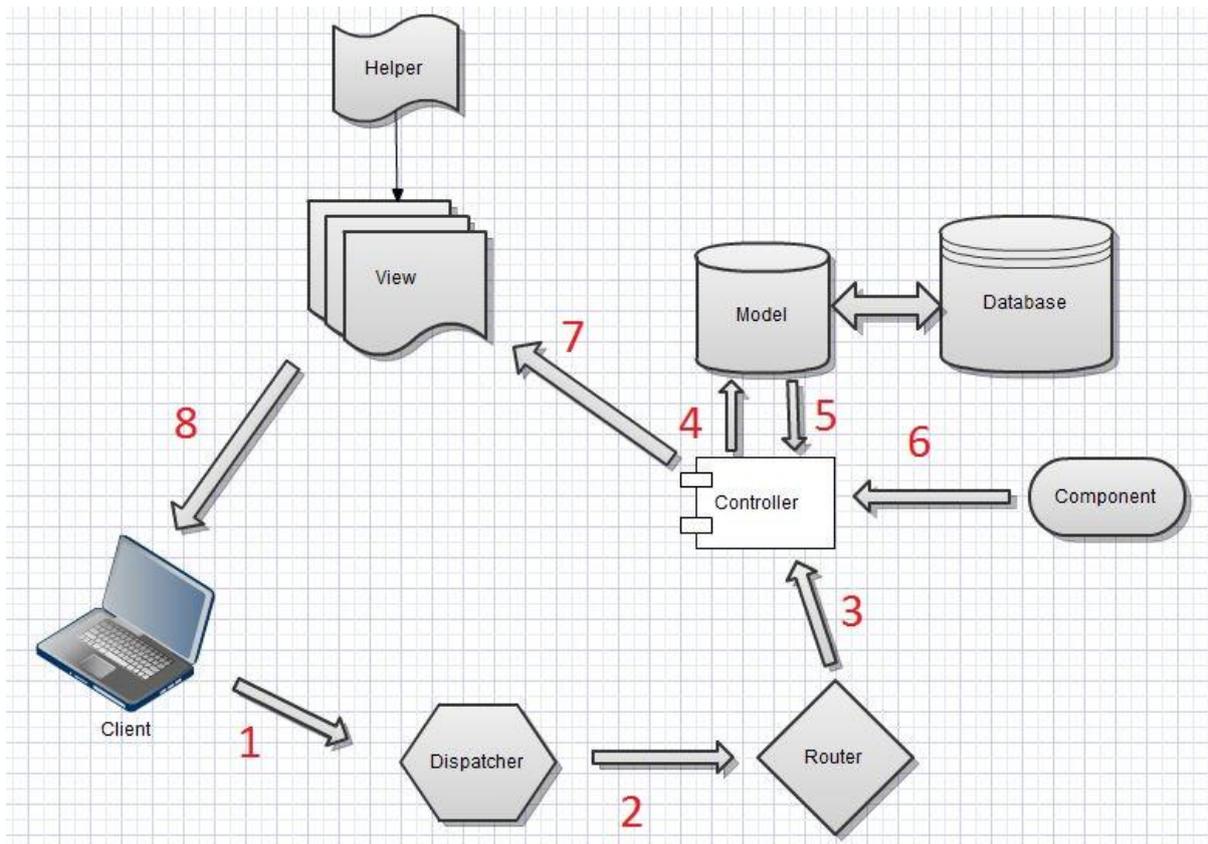
Register Cancel



3. Software architecture

3.1 Conceptual design

System Architecture



The system architecture is based on the MVC Architectural pattern and contains the following components:

Model: The Model section classes will perform different database functionalities like add, update and delete records. There are some classes that will populate data in the variables for the controller section.

View: The View renders the output provided by the controller class according to the specified layout. If it's an AJAX call then the view class takes the help from the helper section.

Controller: The Controller is used to manage the core business logic of the application. This is the heart of the design. It performs the business logic by manipulating the data according to the request by communicating with model classes. Subsequently, it will render the output by calling the corresponding class of the view section.

Helper: The Helper is a class that aids in view logic. JQuery framework is used in helper section for providing the support of AJAX.

Component: The Component is a class that aids in controller logic .External interfaces can be integrated with the system using the component module.

1) Google Geo Class is used to call the Google Maps API to identify the co-ordinates of the bike in terms of latitude and longitude.

2) To facilitate communication among end user, a forum is integrated with the controller section.

Routes: Routes determine which controller class to be called based on the requested URL. As an example: if despatch sends a request like `www.wanabike.com/user_registrations/add`. It means that `add()` function of `user_registrations` controller class is to be called.

Despatch : Despatch gathers request from the client. Then it passes the corresponding URL to the routes section.

Client : End Users access the website with a help of a client that could be the internet such as mobile phones, laptops, iPod touch etc.

3.2 System specification

All the technologies and frameworks used in the proposed system are open source.

- Linux is the operation system with Apache HTTP Server.
- The architectural pattern used is MVC. This is because :
 - It separates/ isolates the business logic from the input and presentation layer
 - It permits independent development maintenance and testing of each module.
- CakePHP, which is an object oriented MVC framework for PHP language is used.
- JQuery, which is a JavaScript framework that has rich library support for AJAX. It enables rapid web development.
- PHP is used for server side dynamic scripting language. All aspects of business logic will be written using PHP language.
- MySQL is the relational database management system as it is very reliable and works well with PHP.
- Free and readily available QR Readers are used to read the website link that is encapsulated in the QR code.
- The system is integrated with Google Maps using Google Maps API to identify location of the bikes and to update the location of the bike in the system.

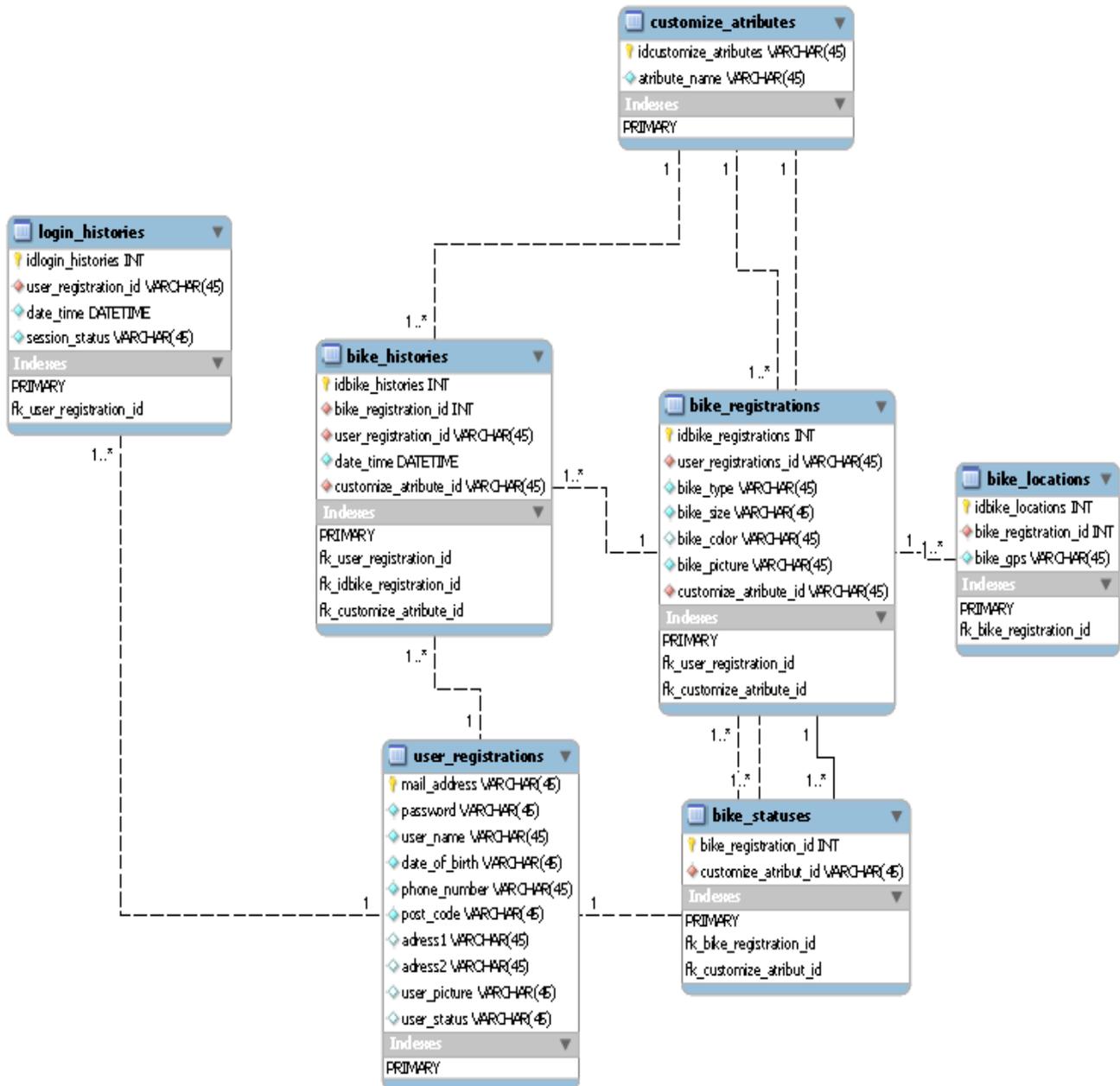
Technologies	PHP
	MySQL
	QR Reader
	Google Maps API
Framework	CakePHP
	JQuery
Environment	LAMP
Architectural Pattern	MVC

3.3 Error handling

Error	Action
User Name / Password is incorrect	The User is prompted to re-enter the username and password
Passwords don't match	User is informed to re-enter the password
Duplicate username	The user is alerted about duplicate username and asked for new username
Search yields no match	Displays an alert "No Results for the Search String"
Server is down	The Website displays a message "Site down for Maintenance"
Database Server is down	User gets a alert saying "Unable to connect to the Database"
User fails to furnish the exact location of the bike	Notifies the user with a alert saying "Enter correct details for taking the bike"
Google Maps is down	A alert is displayed saying "Maps are down"

4. Detailed software design

4.1 Database Design



The entire database is divided into three sets of tables namely master data tables, transaction data tables and customization data tables.

Master data tables : It contains the master table of the users and bikes which generally don't change very often.

user_registrations : Contains the user information such as user name , email address , postal address etc.

bike_registrations : Contains various bike information such as bike id , color etc.

Transaction data tables : It contains the transaction data of the system which is updated as and when a bike is

taken or returned.

login_histories : Contains details about the login details of the end-user.

bike_statuses : Contains the present status of the bike, i.e. whether it is available or already taken

bike_histories : Contains detailed information about when the bike was taken , when was it returned , who had taken it etc.

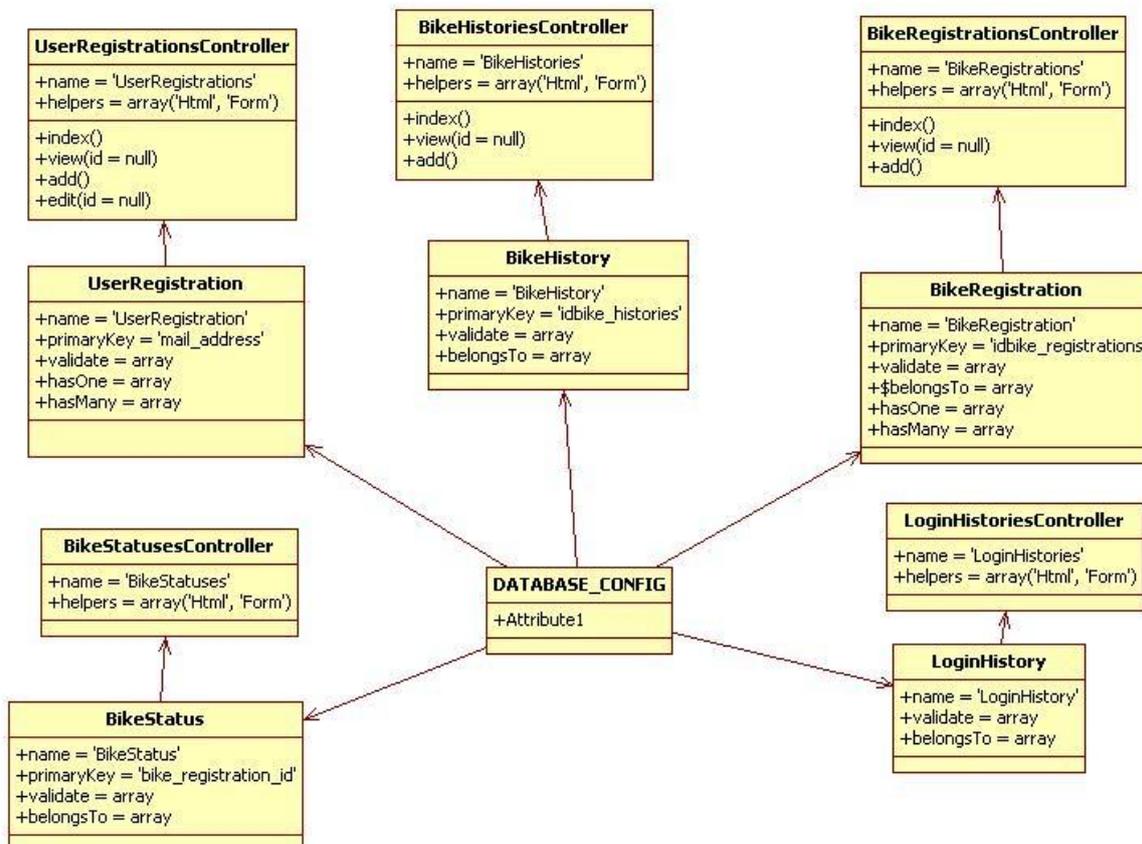
bike_locations : Contains the geographical location of each bike in terms of latitude and longitude.

Customization data tables : It contains the data that can be customized by the end user of the system.

customize_attributes : The administration can define the various statuses a bike and end user can take. This could be Active, Blocked, Pending for approval etc

4.2 Class Diagram

Based on the MVC Architecture, classes have been identified either as Controllers or models. Database_config class maintains details required for establishing database connectivity such as connection string. Every model class is mapped to its corresponding controller class.



5. Approvals

Name	Title	Date yyyy-mm-dd	Signature