

# RAPID ORDERS

CLASS: 6301.014

GROUP NAME: WEBMASTERS

GROUP MEMBERS:

Apurva Inamdar (aai140130)

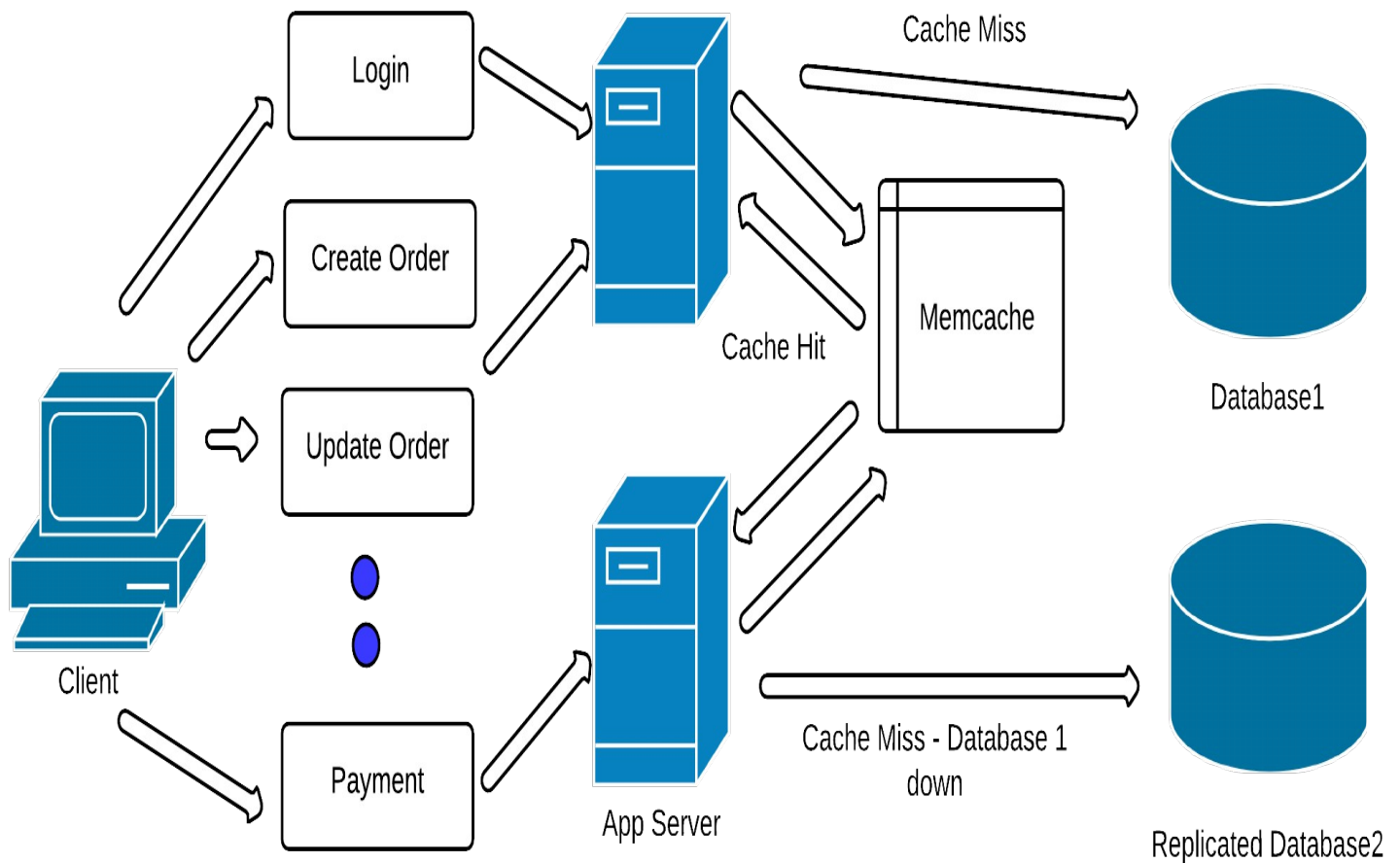
Saankya Ramdas (sxr146530)

Swapnil Hasabe (sdh140430)

## **Introduction:**

If you are an avid online shopper, or in urgent need of items, we are here to serve you. Our web application brings quick delivery service, allowing customers to browse the offerings of local shops online, order, and get the items within few hours. The customer can log in to system and enter the item name, shop name or restaurant name and select their required product. Customer will have option to select their budget for that product. The administrator or web owner can view, update the request and status of the order made by customer. The application is mainly intended towards Veterans, Disabled and Infants who will not be able to go out to buy products and need support during emergency. The pharmacy, restaurant and baby care section mainly caters to the above mentioned customers.

## **Architectural Diagram:**



## **Module Description:**

| Module      | Technologies Used        | Reason   |
|-------------|--------------------------|--|
| Front End   | HTML5, CSS, JQuery, AJAX | HTML5 provides improved semantics, local storage support, embed scripts, ease of displaying animations. CSS and JQuery to maintain consistency among styling of pages and better user experience. AJAX for asynchronous calls to the server, compact code, supporting dynamic content and does not require reloading of pages. |
| Back End    | MySQL                    | Easy creation, updating and querying of tables. User friendly, Open source and available for free.   |
| Server      | WAMP (PHP, Apache, SQL)  | Contains a collection of all programs available as a single package which are compatible with each other. (Do not need to download each module separately). Easy integration and hosting on the server.  |
| Cache       | Memcache                 | Increases site performance by providing distributed caching (maintains a cache in every node and checks every node before directing to the database). Memcached maintains a key and value pair to place and retrieve details from the cache which makes it easy to use. Open source and available online.                      |
| Web Service | RESTful                  | Easy to use and integrate with existing HTML, XML pages, simpler to implement compared to SOAP.  |
| Security    | OpenSSL                  | SSL/TLS for added security of the web application. Open source and available for free.   |

## **Functionalities:**

1. **Login:** User needs to log in to the application to place orders. The application is user friendly and allows the user to check out as a guest as well but returning users will have added advantage of their address and other information being stored in the database.
2. **Register:** Returning users can register and sign up for a username and password. They can later access the application and place orders with ease.
3. **Create Order:** User can shop from a variety of items provided with ease of just one click (Add to Cart). The customer has an option to order food items from Restaurants, medicines from Pharmacy, Baby Care products, Groceries from Supermarkets, Beauty and Healthcare products.
4. **Update Order:** The customer can update their order by adding/removing items from their cart before proceeding to checkout.
5. **Search items:** There is a search bar available to lookup items quickly. The search navigates the user to the specified item for quick buy.
6. **Track Order:** Once the order has been placed, the user can log in to the application and use the tracking feature to find their order with the Order ID. The application will then display the order details as well as the order status.
7. **Payment:** The application has a payment database which will be linked with the Bank's database. Once the user has chosen their items, they can proceed to payment and checkout. The application checks the payment details with the database and proceeds with the transaction on successful validation.

## **Web Services:**

1. **Register:** The user is prompted to fill out a form to register onto the application. The user details such as name, address, phone number is asked for. This form data is then converted to a JSON object and stored to the database using the POST method. For confirmation, few fields are displayed to user.
2. **Login:** The user enters the user name and password. This information is authenticated with the database and if a matching record is found, the user's first name is fetched as a JSON object using POST method and displayed on the page as - "User (Firstname) has logged in successfully".
3. **Create/Update Order:** The user adds the item on to the cart. This information is dealt using web services. The order details is stored in the session in the form of JSON data. The temporary order details are stored in the session as a JSON data. This data is retrieved in case the user wants to add/delete items from the cart. Once the changes are made, the new data is again stored in the session.
4. **Search Order:** User can search order with order ID. OrderID is matched with the database and contents corresponding to particular OrderID are fetched and displayed in JSON format. Services are enabled using RESTful.

## **Core Features:**

1. **Authentication/Authorization:**  
The application is protected by a username and password for increased security. The first time user registers in the website and is provided a username and password. This information is then used to login to the application after validation.
2. **High Performance:**  
This feature is made available in the application by using distributed caching technique. It has been implemented using memcache. The Memcache module is added as an extension in the server. This feature has been implemented in the tracking order functionality of the application. Once the user creates an order, on successful storage in the database, the order details is store in the cache memory (memcache) as a key value pair of Order ID as they key and the array of order details as the value. The user can then use the tracking functionality to track the order. During order tracking, the application first checks the memcache if the order id is available. If there is a Cache Hit (Order found in cache memory), the order details are retrieved from the cache memory which is stored on the client side and

is retrieved much faster than retrieval from database. In the case of a cache miss, the order details is retrieved from the database. This gives the user the flexibility to Update newly create orders as quickly as possible.

3. **Availability:**

The application provides an added level of redundancy to support for high availability. Since the web application mainly deals with creation and delivery of orders, the database is critical part of the application. The application provides Database Replication for increased availability. The data in the main database is replicated onto a slave database. In case of a Database failure, the application connects to the redundant database and provides the required functionalities to the users without lack of information.

4. **Compression:**

The response from the server is compressed using GZIP. Compression is used to make the data flow faster from the server to client and it helps reduce resource usage, such as data storage space or transmission capacity. The compression was configured in Apache in httpd.conf file in the Wamp Server to load Mod\_filter and Mod\_deflate modules.

5. **Encryption:**

SSL was implemented using Open SSL. It is a form of security for sites that handle sensitive information such as customer names, phone numbers, addresses and credit card numbers. It ensures that all the data is handled by encrypting the information when exchanged between the client and the server. This was implemented by creating a self-signed certificate and key which is then used for further exchange in the application. The connection then changes from an http to https (secured) connection.

6. **AJAX:**

Ajax is web development technique used on the client-side to create asynchronous application. JavaScript Object Notation (JSON) is used for data interchange. AJAX allows easier and quicker interaction between user and website as pages are not reloaded for content to be displayed. XmlHttpRequest is an actual object used which increases the performance and speed by reducing response time. In our project, Ajax calls are used in the Cart Process for adding item in cart, updating cart, view cart items alerting user.

## **Additional Features:**

### **1. Single Sign-On:**

Single Sign-On was implemented using OAuth. It provides ease to the user of not having to sign up on every website and sign in through a trusted website like Facebook, google etc. In our project, Facebook login was implemented by creating a Facebook Application on the developer.facebook.com and mentioning the App ID and Secret key in the Local server. This works with a trust between the two websites which is established by the security key exchange. Communication takes place between the local server and the Facebook server once the user authenticates the credentials. User can only login with Facebook if they have created an App in their Facebook account.

## **Problems Faced:**

1. The project was started by creating the HTML pages and the database. We struggled with creating database schemas because when we actually started to implement that schemas we had to change it little so that we could extract data after completing our order. It was really challenging for us. We used JSON for data interchanging. After checkout of order, we had to assign order id to complete order and that was time taking and needed so much understanding and concentration.
2. Playing with AJAX was very time consuming. It took time to implement update order functionality. For that, we made sure that when item is added into cart, it is visible. This is done by simple if else statement.
3. While implementing Single sign on, We got error of mismatching the URL's given in Facebook app and our code. But we resolved that problem by configuring app's setting correctly.
4. While doing database replication, we tried to configure it in PHP myadmin but it did not work out as we wanted so we decided to configure it by command prompt, and it worked out successfully.
5. Finding the correct memcache dll file for the PHP took lot of time. The memcache module was getting accepted but was not able to create Memcache object in the code. There is no clear documentation available online on which dll file works for the PHP versions and on how to overcome the configuration problems.

## **References:**

### Compression

1. <http://stackoverflow.com/questions/9999654/wampserver-doesnt-go-green-stays-orange>
2. <http://stackoverflow.com/questions/21746184/enable-gzip-module-on-wampserver>

### Memcache

1. <http://stackoverflow.com/questions/3016656/how-to-enable-memcache-in-wamp>
2. <http://liuweipingblog.cn/php/how-to-install-memcache-on-windows-7-x64-with-wamp/>

### Single Sign On

1. [https://www.youtube.com/watch?v=io\\_r-0e3Qcw](https://www.youtube.com/watch?v=io_r-0e3Qcw)
2. <http://www.oauthlogin.com/home.php>

### Database Replication:

1. [http://docs.oracle.com/cd/A58617\\_01/server.804/a58227/ch\\_repli.htm](http://docs.oracle.com/cd/A58617_01/server.804/a58227/ch_repli.htm)
2. <https://www.youtube.com/watch?v=nfsmnx24gxU>

### Web Services:

1. [https://www.youtube.com/watch?v=jG1sz\\_0o22Q](https://www.youtube.com/watch?v=jG1sz_0o22Q)
2. <http://davidwalsh.name/web-service-php-mysql-xml-json>
3. <http://codecanyon.net/item/ajax-cart-for-html-websites-with-digital-products/4903762>

### Authentication:

1. <http://www.javaworld.com/article/2076292/core-java/secure-a-web-application--java-style.html>

### HTML/CSS/JQuery:

1. <http://www.w3schools.com>