Software Requirements Specification

for

Online Cake Delivery

Version 1.0

Prepared by   
Anvesh Athmakuri  
 Ashik Shaik  
Srikantha Babu Pusapati  
Sudhira Badugu

University of North Texas

10/11/16

Table of Content

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2-3

2.5 Design and Implementation Constraints 3

2.6 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 4

3.4 Communications Interfaces 4

4. System Features 4

4.1 System Feature 1 5

4.2 System Feature 2 6

4.3 SystemFeature3………………………………………………………………….…….7

4.4 System Feature4………………………………………………………………….……7

5. Other Nonfunctional Requirements 8

5.1 Performance Requirements 8

5.2 Safety Requirements 9

5.3 Security Requirements 9

5.4 Software Quality Attributes 9

6. Other Requirements 10

**7.Glossary…………………………………………………………………………….… 10**

**8.Architecture of System………………………………………………………………..10**

9.UML Design Document……………………………………………………………….12

9.1 Class Diagram**…………………………………………………………………......12**

9.2 Sequence Diagram**……………………………………………………………………….13**

9.3 UseCase Diagram**………………………………………………………………………..13**

10.Test Plan……………………………………………………………………………...15

11.Risk Management……………………………………………………………………21

12.Project Plan…………………………………………………………………………..22

13.Meeting Minutes……………………………………………………………………..24

14.Progress Report……………………………………………………………………...28

15.Member Contribution Table………………………………………………………..28

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |

1. **Introduction**

Lives are getting Hectic day by day and buying cakes in these busy schedules is bit tough. To make it simple and also to deliver the cakes through the available people that voluntarily registers online; making this a new value added feature to the customer, for a faster and user friendly approach. The main objective of this application is to deliver cakes to any corner of the city using the interactive website.

This project will help in revolutionize the way events, parties are celebrated across the city, with no permanent staff and scheduling issues, delivery on time will be taken care by using a single site with a few clicks of buttons.

**1.1Purpose**

Describing the functional requirements and software specifications for Online Cake Delivery. This document will cover each of the system’s intended features and also covers hardware requirements which are of high priority and committed for release 1.0.

**1.2. Document Conventions**

This document features some terminology in which readers may be unfamiliar with. See Appendix A (Glossary) for a list of these acronyms and their definitions. The format of the document would be Boldface and 12 fonts for headings and the rest with ‘Times New Roman’ 12 fonts.

* 1. **Intended Audience and Reading Suggestions**

This document is intended for all individuals participating in and/or supervising the Online Cake Delivery such as Developers, Customers, Administrator, Business analysts, QA team, managers (TA).

**1.4. Project Scope**

This project is to develop a web application, which helps customers to get cake from their preferred choice of store without a walk to the store. The scope so far has been the completion of the basic interfaces that will be used to build the system. The database used, has been set up and given the necessary permissions.

**1.5. References**

* <https://en.wikipedia.org/wiki/Gantt_chart>
* <https://www.youtube.com/watch?v=TjxL_hQn5w0>
* <http://economictimes.indiatimes.com/definition/risk-management>
* [www.processimpact.com](http://www.processimpact.com/)
* <http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>

1. **Overall Description**

**2.1 Product Perspective**

The ‘Online Cake Delivery’ replaces the existing system of customers manually going store and buying cakes only after visiting their preferred stores. Users simply need to login into application, enter the type of the cake they want and from the store they need it. When they submit the order from their preferred choice of from their preferred store Then they get confirmation. Depending on the type of users i.e. Customers, Deliverer and Admin they are assigned different privileges. Customers can order cakes online as they

wish from preferred store. Admin can delete/edit/add deliverer, stores etc. admin can also add cake details. Deliverer gets the notification about the order details like where to get the cake and where to deliver it. Here delivery guy buys the cake with is money and it is collected from the customer after delivery.

**2.2 Product Functions**

* Customer registration
* Login to the system for registered users
* Search the availability of cakes
* Payment is flexible with card
* Confirmation for the delivery to the customer
* Sending the confirmation and mailing the address to the deliverer
* Deliver the cake for the mailing address at given time
* Feedback for the delivery

## 2.3 User Classes and Characteristics

**User** The main functionalities of user were Registering, Login. First User has to register by giving his details like Name, Email Id, Phone number, Address. After successful registration user can login by using his Username and Password.

**Customer User** Customer user is a type of User class who is distinguished based on the functionalities offered. After Login to the system Customer can have functionalities like Search a cake, Order Request, Feedback.

**Deliverer User** Deliverer user is a type of user class who is distinguished basing on the functionalities offered**.** After Login to the system Delivere can have functionalities like View the orders, selection of an order, delivering the cake in time.

**Admin** After sign up process, admin logs in into the application. Admin can add/delete/edit information of customers, deliverers etc. who signed up into the application. Admin would send the deliverer details to customer once the order is confirmed.

**Order** Order has the details like Order ID, Address details, Deliverer details, Customer details. Customer initiate’s order once deliverer selects the order It would be get confirmed.

**Order List** Order List consists of all orders with their Order id, order status. Deliverer view the order list and selects the particular order based on his interest.

**Shipping Information** It consists of complete details of Delivery address. It is associated to each order.

**Payment** Payment is like Cash on delivery, once customer pays money deliverer updates delivery and payment status.

**Feedback** After completion of delivery customer can give is feedback on Cake, Deliverer, and Service.

**2.4 Operating Environment**

This application is compatible on Windows vista and above and Mac 10.1.

It requires mostly used web browsers like Internet Explorer, Mozilla Firefox, Google Chrome.

Memory: device will have 2GB internal hard drive. Software and database cannot exceed this amount

**2.5 Design and implement constraints**

Its front end should be implemented using PHP, HTML and Java-script.

At the backend we used by the MYSQL database.

The web server XAMPP is used to store the data and recover the data.

**2.6 Assumption and dependencies**

It is assumed user should have the basic knowledge with computer.

This system of application connection in the system.

It is assumed that all information given by the user for delivery is appropriate.

1. **External Interface Requirements:**

**3.1 User Interfaces:**

The main page, where user can login or register as a new user.

The registered user will be a given the option to edit their profile, search for the available cakes, order the cakes.

After the cake is ordered, it redirects into interface where payment can be made.

If the user is deliverer the interface has the available orders.

If the user is admin then it takes to a new page where admin can manage the cakes availability and order confirmation.

**3.2.   Hardware Interfaces**:

* Processor: Pentium or Higher.
* RAM: 512MB or Higher

**3.3.   Software Interfaces**:

* Operating System:  Unix, Linux, Mac, Windows etc.
* Development tool: XAMPP, PHPMyAdmin.
* Data Base: MySQL

**3.4. Communication Interface:**

* HTTP is used for send data between the server and the client and also for establishing connection between user and database.
* SMTP is used to send and receive emails and receive the confirmation of order.

1. **System Features**

**4.1 Sign Up/Register**

**4.1.1 Description and Priority**

To all the new users they have to register by clicking the register button.

Priority: 1.

**4.1.2 Stimulus/Response Sequences**

The user will be redirected to register interface. It asks for the user info and checks whether the user id and other details are previously saved in the database, if so it generates a message that the user is already registered and asks to sign in with his details.

**4.1.3 Functional Requirements**

REQ 1: - full name mandatory TYPE: **-** String

REQ 2: - emailmandatory TYPE: - Alphanumeric

REQ 3: - password mandatory TYPE: - Alphanumeric

REQ 4: - address mandatory TYPE: - Alphanumeric

REQ 5: - mobile mandatory TYPE: - Numeric

REQ 6: - Register button TYPE: - Submit Button.

**4.2 Login**

**4.2.1**  **Description and Priority**

Login is function where the user logins each time.

priority is 1.

**4.2.2 Stimulus/Response Sequences**

If the user is a customer he should enter the valid user id and the password it also has a link when the user has forgotten the password when the user clicks on that the user will send the password reset link to the email.

**4.2.3 Functional Requirements**

REQ - 1: - User Name Mandatory TYPE: -String

REQ - 2: -Password Mandatory TYPE: - Alphanumeric

REQ - 3: -Login Button Mandatory TYPE: -Submit Button

**4.3 Search for the cakes**

**4.3.1 Description and Priority**

The valid user can search for the cakes available in the website, he can see price of each cake and select the cake he likes.

Priority :1

**4.3.2 Stimulus/Response Sequences**

The user can select the any of the available cakes, if he wants any cake which is out of stack, he can request it for the cake with the delivery time then he gets the valid response, whether the cake can be delivered in the feasible time.

**4.3.3 Functional Requirements**

REQ - 1: - Cake Details mandatory TYPE: - STRING

REQ - 2: - show available cakes mandatory TYPE: - BUTTON

**4.4 Order**

**4.4.1 Description and Priority**

Only the validated/registered user can order a cake and post his mailing address and other details for the order to be completed.

Priority :2

**4.4.2 Stimulus/Response Sequences**

The user can order a cake, and he could update his appropriate mailing address which should be deliverable, if the address is not deliverable, he gets an error message that delivery is not available, and asks for another address.

**4.4.3 Functional Requirements**

REQ - 1: - Name Mandatory TYPE: - String

REQ - 2: - Mailing Address Mandatory TYPE: - String

REQ - 3: - Phone Number Mandatory TYPE: - Numeric

REQ - 4: - Zip Code Number Mandatory TYPE: - Numeric

REQ - 5: - Time of delivery Mandatory TYPE: - String

REQ - 6: - Date of delivery Mandatory TYPE: - Time

REQ - 7: - Submit Mandatory TYPE: - Button

**4.5 Confirmation**

**4.5.1 Description and Priority**

The confirmation of the order is given by the available deliverer with in the slack time of 30 min from the delivery time specified by the user. After the order is accepted by the deliverer the confirmation email is sent to the user email and the updated in the database.

Priority :2

**4.5.2 Stimulus/Response Sequences**

If the order is confirmed by the delivery person then a confirmation mail will be sent to the customer, else if the slack time nears the delivery time and no one accepts the order then cancellation mail will be sent to the user.

**4.5.3 Functional Requirements**

REQ - 1: - Deliverer Id Mandatory TYPE: -String

REQ - 2: -Expected time of delivery mandatory TYPE: -String

REQ - 3: -Contact Number of deliverer mandatory TYPE: -Numeric

* **Priority Legends**

|  |  |
| --- | --- |
| Priority | Explanation |
| 0 | Low |
| 1 | Normal |
| 2 | High |

1. **Other Non-functional Requirements**

**5.1 Performance Requirements**

Internet connection is needed for the customers for using the application.

In case of opening forms, of popping error messages there is delay much below 2 seconds.

In case of opening databases, sorting questions and evaluation there are no delays and the operation is performed in less than 2 seconds.

**5.2 Safety Requirements**

Only authorized person can login into the system.

The data which is to be transmitted to the server without any changes.

**5.3 Security Requirements**

The administrator has the access permissions to update the database.

No user gets access to system until he gives valid user id and password.

If the user requests for the new password it should be reset by email.

**5.4 Software Quality Attributes**

**Availability:**

Checking that the system always has something to function and always pop up error messages in case of component failure. In that case the error messages appear when something goes wrong so to prevail availability problems. **Usability:**

Checking that the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

**Functionality:**

Checking that the system provides the right tools for editing question databases, creating session tests and analysing the test sessions. In that case the tools that the Database editor provide are the ones that provide that attribute.

**5.5 Business Rules**

* Once the user has ordered the cake and got the confirmation, he can only cancel the order 2 hours before the time to be delivered.
* The user cannot cancel the delivery after the deliverer reached the destination
* No refund will be provided if the cake gets damaged.

1. **Other Requirements**

• The deliverer should have his own medium of transportation.

1. **Appendix A: Glossary**

|  |  |
| --- | --- |
| SMTP | Simple mail transfer protocol |
| PHP | Hypertext preprocessor |
| XAMPP | Cross platform Apachy- Maria – PHP-Perl |
| HTTP | Hypertext transfer protocol |

1. **Architecture**

SYSTEM ARCHITECTURE DESIGN

The architecture that we are using for our project Online Cake Delivery is client-server architecture. This architecture best suits our requirements, as the project we are building is a web application. We are using this architecture because in this design we can make changes very easily to the system. This architecture supports the interactive process between the client and server, which is very important in our system because our system is mainly used as mode of communication between the customers and deliverers.

**User**

Browser

PHP Interpreter

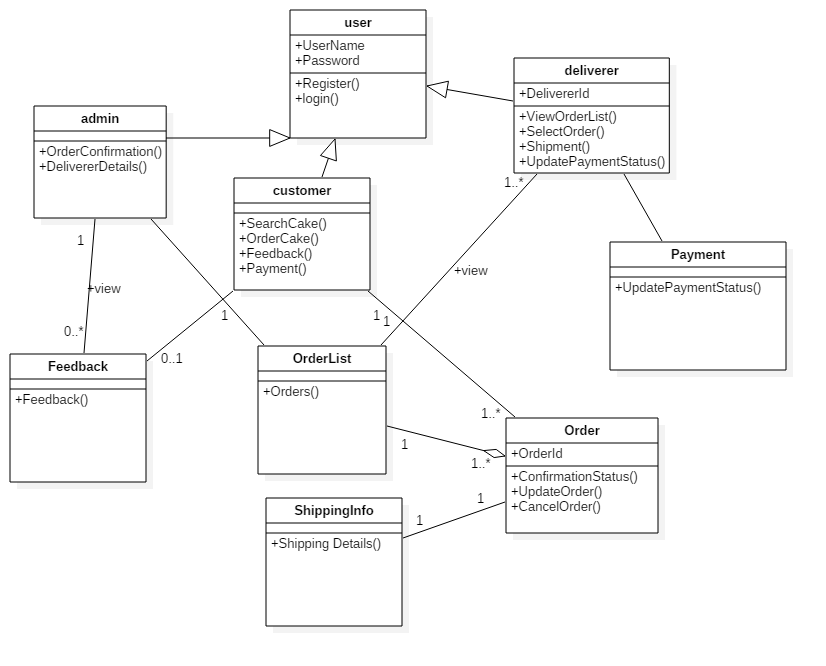
Web Server

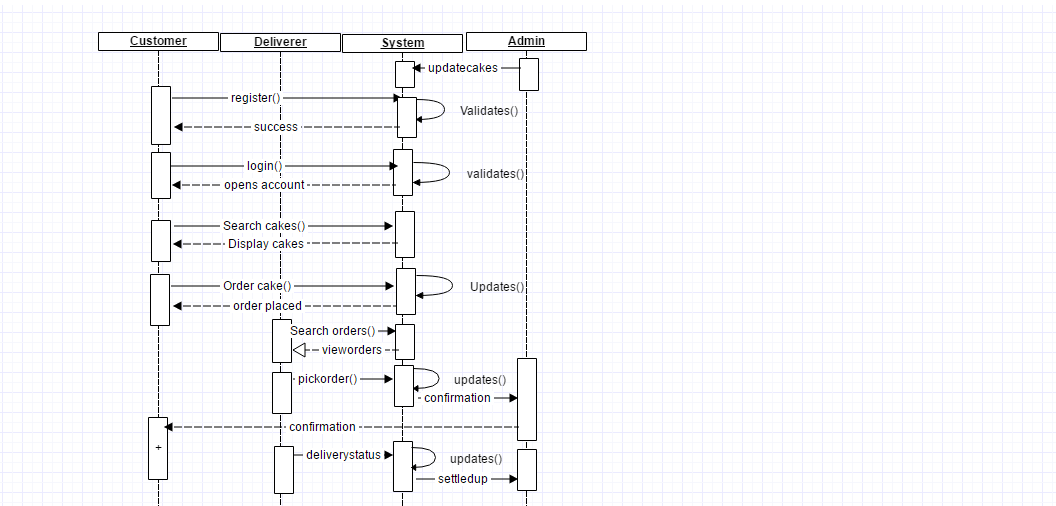
Application Server

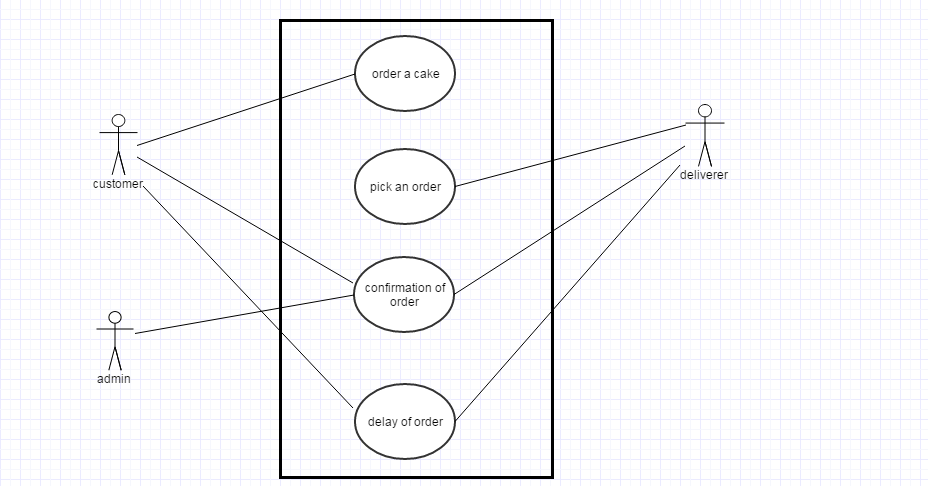
**MySQL**

**Database**

1. **UML Diagram** 
   1. **Class Diagram**



* 1. **Sequence Diagram** 
  2. **Usecase Diagram**



**Use case Text**

* Order a cake:
  + The customer logins into the system, if not registered he registers into the system.
  + After logging he browses the site for various cakes, there availability price range and selects a cake.
  + On selection of a cake he then moves onto placing the cake for delivery i.e. Order the cake. By providing details about the delivery address and other contact info.
* Pickup an order:
  + The delivery guy logs into the system, if not already registered he now registers and logs into the system
  + Views all the available orders that has to be delivered and the time available for the order.
  + Decides on a delivery he can perform on time and then picks the order for delivery.
* Confirmation of Order
  + Once the order is picked up by the deliverer a confirmation mail will be sent to the customer about the order status
  + Also the admin is notified about the deliverer order pickup.
* Delay in Order (ERROR CASE)
  + The deliverer confirms about the order delivery
  + But can’t complete the delivery on time as requested by the customer
  + Error case is when the Deliverer delays the order/ couldn’t complete the order in time.

1. **Test plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case No. | Trace to Requirement No. | Test steps | Expected output | Comments | Pass/Fail |
|  | 4.1.1 | 1. Click on customer register page. | Register page should be displayed. | User should be redirected to the registration page. |  |
|  | 4.1. R1 | 1. Open customer register page. 2. Leave customer name blank. 3. Enter email Id. 4. Enter password. 5. Enter address of the customer. 6. Enter mobile number. 7. Click on Register button. | A display should prompt for enter customer name. | A ‘\*’ or popup should show that customer name is mandatory field |  |
|  | 4.1. R2 | 1. Open customer register page. 2. Enter customer name. 3. Leave email Id blank. 4. Enter password. 5. Enter address of the customer. 6. Enter mobile number. 7. Click on Register button. | A display should prompt for enter Email id. | A ‘\*’ or popup should show that Email id is mandatory field |  |
|  | 4.2. R3 | 1. Open customer register page. 2. Enter customer name. 3. Enter email Id. 4. Leave password blank. 5. Enter address of the customer. 6. Enter mobile number. 7. Click on Register button. | A display should prompt for enter password. | A ‘\*’ or popup should show that password is mandatory field |  |
|  | 4.1. R4 | 1. Open customer register page. 2. Enter customer name. 3. Enter email Id. 4. Enter password. 5. Leave address of the customer blank. 6. Enter mobile number. 7. Click on Register button. | A display should prompt for enter address of customer. | A ‘\*’ or popup should show that customer address is mandatory field |  |
|  | 4.1. R5 | 1. Open customer register page. 2. Enter customer name. 3. Enter email Id. 4. Enter password. 5. Enter address of the customer. 6. Leave mobile number blank. 7. Click on Register button. | A display should prompt for enter customer mobile number. | A ‘\*’ or popup should show that customer mobile number is mandatory field |  |
|  | 4.1. R6 | 1. Open customer register page. 2. Enter customer name. 3. Enter email Id. 4. Enter password. 5. Enter address of the customer. 6. Enter mobile number. 7. Click on Register button. | A display should prompt by suggesting the user is successfully registered. | The user should be redirected or a prompt should show that the user is successfully registered. i.e. Test the register button is working correctly. |  |
|  | 4.1. R1 | 1. Open customer register page. 2. Enter customer name with special characters. 3. Enter email Id. 4. Enter password. 5. Enter address of the customer. 6. Enter mobile number. 7. Click on Register button. | A display should prompt by suggesting the user name should be only alphabets. | A ‘\*’ or popup should show that customer name is mandatory field which accepts only alphabets |  |
|  | 4.1. R2 | 1. Open customer register page. 2. Enter customer name. 3. Enter email Id with blanks or invalid email id. 4. Enter password. 5. Enter address of the customer. 6. Enter mobile number. 7. Click on Register button. | A display should prompt by suggesting the user email id should be corrected | A ‘\*’ or popup should show that customer email is mandatory field and should be a valid mail id. |  |
|  | 4.1. R3 | 1. Open customer register page. 2. Enter customer name. 3. Enter email Id. 4. Enter password of less than 7-character length. 5. Enter address of the customer. 6. Enter mobile number. 7. Click on Register button. | A display should prompt by suggesting the user password should be of capital small and length of more than 7 characters. | A ‘\*’ or popup should show that customer password is mandatory field and should be of min length of 7 characters. |  |
|  | 4.1. R4 | 1. Open customer register page. 2. Enter customer name. 3. Enter email Id. 4. Enter password. 5. Enter address of the customer with special characters. 6. Enter mobile number. 7. Click on Register button. | A display should prompt by suggesting the user address should be correctly entered. | A ‘\*’ or popup should show that customer address is mandatory field and should be a valid alpha numeric type only. |  |
|  | 4.1. R5 | 1. Open customer register page. 2. Enter customer name. 3. Enter email Id. 4. Enter password. 5. Enter address of the customer. 6. Enter mobile number with wrong entry. 7. Click on Register button. | A display should prompt by suggesting the user number is not correct. | A ‘\*’ or popup should show that customer number is mandatory field and should be a valid with specific number format only. |  |
|  | 4.1. 1 | 1. Open customer register page. 2. Enter customer name. 3. Enter email Id. 4. Enter password. 5. Enter address of the customer. 6. Enter mobile number 7. Click on Register button. | The user should be shown successfully registered message | All the user details should be stored in database and the user should be moved to home page of the system. |  |
|  | 4.2.1 | 1. Open login page. | User should be redirected to the login page | A page should load asking the user to login using username/mail Id and password. |  |
|  | 4.2. R1 | 1. Open login page. 2. Leave User name blank 3. Enter User password 4. Click Login button. | A prompt should be displayed asking the user to enter valid/ registered user name | The username from the database is referred and validated. |  |
|  | 4.2. R2 | 1. Open login page. 2. Enter User name. 3. Leave User password blank. 4. Click Login button. | A prompt should be displayed asking the user to enter valid/ registered user password. | The user password from the database is referred and validated. |  |
|  | 4.2. R3 | 1. Open login page. 2. Enter User name. 3. Enter User password. 4. Click Login button. | A prompt should be displayed asking the user to enter valid/ registered user password. | The user password from the database is referred and validated. |  |
|  | 4.2. 1 | 1. Open login page. 2. Enter User name. 3. Enter User password. 4. Click Login button. | User should be logged in successfully | The user name password from the database is referred and validated and the user should be redirected to the home page of the system. |  |
|  | 4.3.1 | 1. Browse the available cakes page | User should be able to see all the available verities of cakes present. | A set of cakes with their names, prices should be visible. |  |
|  | 4.3. R1 | 1. Browse the available cakes 2. Cake details not visible 3. Display cake availability | User should be able to view all the available cake details in the browsing page | Cake details like price, ingredients should be visible. |  |
|  | 4.3 R2 | 1. Browse the available cakes 2. View cake details 3. cake availability not visible | User should be able to view which cake is available and which is not. | Cake availability should be visible to the user. |  |
|  | 4.4 | 1. Order the cake | User if logged in should be redirected to order page | On validating the user if logged in should be able to continue with the ordering of the cake. Else should be taken to login page. |  |
|  | 4.4 R1 | 1. Order the cake page 2. Leave the name on the cake. 3. Enter Delivery address. 4. Enter Phone number 5. Enter Zip code 6. Enter Time of delivery 7. Enter Date of delivery 8. Click submit button | User should be prompted to enter the message to be written on the cake. | A dialog or ‘\*’ should be shown indicating it is a mandatory field. |  |
|  | 4.4 R2 | 1. Order the cake page 2. Enter the name on the cake. 3. Leave the Delivery address blank. 4. Enter Phone number 5. Enter Zip code 6. Enter Time of delivery 7. Enter Date of delivery 8. Click submit button | User should be prompted to enter the Delivery address. | A dialog or ‘\*’ should be shown indicating it is a mandatory field to enter the user delivery address |  |
|  | 4.4 R3 | 1. Order the cake page 2. Enter the name on the cake. 3. Enter Delivery address. 4. Leave Phone number blank. 5. Enter Zip code 6. Enter Time of delivery 7. Enter Date of delivery 8. Click submit button | User should be prompted to provide phone number | A dialog or ‘\*’ should be shown indicating it is a mandatory field to enter phone number. |  |
|  | 4.4 R4 | 1. Order the cake page 2. Enter the name on the cake. 3. Enter Delivery address. 4. Enter Phone number 5. Leave Zip code 6. Enter Time of delivery 7. Enter Date of delivery 8. Click submit button | User should be prompted to enter the zip code | A dialog or ‘\*’ should be shown indicating it is a mandatory field to provide zip code for the delivery. |  |
|  | 4.4 R5 | 1. Order the cake page 2. Enter the name on the cake. 3. Enter Delivery address. 4. Enter Phone number 5. Enter Zip code 6. Leave Time of delivery 7. Enter Date of delivery 8. Click submit button | User should be prompted to enter the time of delivery. | A dialog or ‘\*’ should be shown indicating it is a mandatory field to enter time of delivery. |  |
|  | 4.4 R6 | 1. Order the cake page 2. Enter the name on the cake. 3. Enter Delivery address. 4. Enter Phone number 5. Enter Zip code 6. Enter Time of delivery 7. Leave Date of delivery 8. Click submit button | User should be prompted to enter the date of delivery | A dialog or ‘\*’ should be shown indicating it is a mandatory field to enter date of delivery. |  |
|  | 4.4 R7 | 1. Order the cake page 2. Enter the name on the cake. 3. Enter Delivery address. 4. Enter Phone number 5. Enter Zip code 6. Enter Time of delivery 7. Enter Date of delivery 8. Click submit button | User should be shown a prompt saying order placed. | Order database tables should be updated stating the new order. |  |
|  | 4.5 | 1. Confirmation page visible once the delivery person accepts the order | The user should be able to view the order status once the delivery person accepts the order. | The order and confirmation tables should be updated and the delivery person is now responsible for an on time delivery. |  |
|  | 4.5 R1 | 1. Confirmation page visible 2. Delivery ID not generated 3. Expected time of delivery updated 4. Contact number of the delivery guy is obtained. | The customer should be viewing confirmation of the order | If the is not generated, then the admin has to update/ confirm the customer of the order. |  |
|  | 4.5 R2 | 1. Confirmation page visible 2. Delivery ID generated 3. Expected time of delivery not visible correctly 4. Contact number of the delivery guy is obtained. | The customer should recheck the confirmation of the order | If the Delivery time is not generated, then the admin has to update/ confirm the customer of the order. |  |
|  | 4.5 R3 | 1. Confirmation page visible 2. Delivery ID generated 3. Expected time of delivery updated 4. Contact number of the delivery guy is not obtained. | The customer should recheck the | If the deliverer phone number is not generated, then the admin has to update/ confirm the customer of the order. |  |

1. **Updated Risk management**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Description | Comments | Contingency plans | Re-Evaluation |
|  | Learning curve | All of the team members need to be in par with the development software’s, languages being used within the project. | Need to be aware of the programing languages being used like Java, PHP and tools like xampp controller. | Ensure that each component is well understood, visualized and the team have enough knowledge to implement it on time. |
|  | Data Security | Need to confirm the Security procedure that has to be followed in the project. | Ensure Security of customer related data, by performing data encryption. | Ensure data encryption and decryption is done properly. |
|  | Ensuring Delivery on time | If no one is available to deliver the product in time, then we need to consider canceling the order made | If the order is not accepted in at least 30 min to delivery time specified, then the order shall be cancelled. | More delivery people should be willing to provide service for the project to be a success. |
|  | Customer cancelling order | If the order is canceled within 30 minutes then it can be allowed else, the user shouldn’t be allowed to cancel the order. | The customer can cancel an order within 30 minutes after confirming. | The customer can cancel the order after placing the order |
|  | Server load & performance | Need to understand the server load and number of server requests it can accept in time. | Ensure server is available for processing requests without downtime. |  |

1. **Updated Project plan**

The updated project plan during the progress of project from requirement analysis phase to development phase along with future testing and deployment phase is detailed.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Time Estimate (in Days)** | **DATE** |
| **Step1: Planning** | **8** |  |
| Activity1.1: Objective | **1** | **4th Sep** |
| Activity1.2: Functionality | **2** | **5th Sep** |
| Activity1.3: Risk Management | **2** | **7th Sep** |
| Activity1.4: Effort estimates | **1** | **9th Sep** |
| Activity1.5: Charts and documentation | **2** | **10th Sep** |
| **Step2: Requirement and Analysis** | **9** |  |
| Activity2.1: Functional & Non Functional requirements gathering. | **6** | **12th Sep** |
| Activity2.2: Finalizing the requirements | **3** | **18th Sep** |
| **Step3: Design** | **12** |  |
| Activity3.1: Architecture of the system | **2** | **21st Sep** |
| Activity3.2: UML design | **2** | **23rd Sep** |
| Activity3.2.1: Class Diagram | **3** | **25th Sep** |
| Activity3.2.2: Sequence Diagram | **3** | **28th Sep** |
| Activity3.2.3: Use Case Diagram | **2** | **1st Oct** |
| **Step5: UI design and development** | **16** |  |
| Activity4.1: Home, About us, contact us pages. | **3** | **12th Oct** |
| Activity4.2:Sign up forms for Customers, Deliverers. | **3** | **15th Oct** |
| Activity4.3: Search and Display cakes page. | **2** | **18th Oct** |
| Activity4.4: Order page. | **2** | **20th Oct** |
| Activity4.5: View Orders list. | **2** | **22nd Oct** |
| Activity4.6: Order Confirmation page | **1** | **24th Oct** |
| Activity4.7: Feedback page. | **1** | **25th Oct** |
| Activity4.8: Other UI requirements if missed any | **2** | **26th Oct** |
| **Step5: Interface Integration** | **6** |  |
| Activity5.1: Developing server controlling different actions and handling requests. | **2** | **28th Oct** |
| Activity5.2: Database Connectivity. | **4** | **30th Oct** |
| **Step6: Database Design** | **8** |  |
| Activity6.1: Database tables design and development. | **4** | **4th Nov** |
| Activity6.2: Querying DB | **4** | **8th Nov** |
| **Step7: Integration and Unit testing** | **13** |  |
| Activity7.1: Test cases writing. | **5** | **12th Nov** |
| Activity7.2: Executing those test cases. | **8** | **17th Nov** |
| **Step8: Deployment and testing** | **3** |  |
| Activity8.1: Hosting the project and testing | **3** | **25th Nov** |

**Gantt Chart**

1. **Meeting Minutes**

Minutes of meeting document is listed in document ‘Meeting\_minutes.docx’.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DATE | DISCUSSION HIGHLIGHTS | TIME | MEMEMBERS ATTENDED | ATTENDEES |
| 30th AUG | Project idea brain storming | 1hr 30min | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 1st SEP | Defining Functionalities precisely | 2 hours | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 4th SEP | Research on platforms and languages | 1hr | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 6th SEP | Identifying the risks in the project | 1hr. 20 min | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 8th SEP | Gathering the requirements and installations | 1hr. | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 9th SEP | Work Sharing of Delivarble -1 | 3hrs | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 10th SEP | Presentation of Delivearble -1 | 1hr | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 11th SEP | Gantt Chart making | 1hr | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 13th SEP | Study and quick review of concepts | 50 mins | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 15th SEP | Presentation for the deliverable 1 | 30 min | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 20th SEP | Conducted workshop, talked with a group of clients and decided to implement feedback system. | 40 min | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 22nd SEP | Feedback of the deliverable 1 obtained and analyzed. | 20 min | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 27th SEP | Reanalyzed and modified the deliverable 1 document on discussing. | 1hr | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 29th SEP | Use case diagram for the project discussed. | 1hr | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 4th OCT | Functional and non-functional requirements discussed. | 30 min | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 6th OCT | System architecture discussed. | 45 min | 4 | Ashik, Anvesh, Srikanth, Sudhira |
| 11th OCT | Deliverable 2 documents and related matter discussed. | 2 hr | 4 | Ashik, Anvesh, Srikanth, Sudhira |

**Meeting 1**

**Date:** 30th August

## Points Discussed

1. The major purpose of the project.
2. Defining the major actors and interaction between them.
3. Visualizing the project.

**Meeting 2**

**Date:** 1st September

## Points Discussed

1. Requirement gathering.
2. Project functionalities defined.
3. Researched on the project required languages, frameworks that can be leveraged.

**Meeting 3**

**Date:** 4th September

## Points Discussed

1. Discussed on the various available platforms to work on project.
2. Interaction between team members they were comfortable to code.
3. Finding strengths of each member of team.
4. Finalized languages to be used

**Meeting 4**

**Date:** 6th September

## Points Discussed

1. Identifying the project deadlines for completion.
2. Making a vague idea of the risks may occur.
3. Classifying and listing the risks and making the risk management table.

**Meeting 5**

**Date:** 8th September

## Points Discussed

1. Installing and getting versed with the push, pull functions.
2. Getting the required software installed like XAMPP server.
3. Making sure everyone is flexible with actions can be performed.

**Meeting 6**

**Date:** 9th September

## Points Discussed

1. Sharing the work mentioned in Deliverable-I
2. Quick view of the functionalities and problems may occur during implementation.

**Meeting 7**

**Date:** 10th September

## Points Discussed

1. Dividing the modules to be covered in the presentation.
2. Preparing the presentation.

**Meeting 8**

**Date**: 11th September

## Points Discussed

1. Project planning.
2. Preparing gantt chart for the project

**Meeting 9**

**Date**: 13th September

## Points Discussed

1. Making a review on the concepts required.
2. Discussion of implementing methods.

**Meeting 10**

**Date**: 15th September

## Points Discussed

1. Discussing points for the presentation.
2. Discussed milestones for the project that are completed.

**Meeting 11**

**Date**: 20th September

## Points Discussed

1. Discussed with other clients in workshop.
2. Observed an update in our requirements.

**Meeting 12**

**Date**: 22nd September

## Points Discussed

1. Confirmed to implement feedback in our system.
2. Deliverable 1 feedback obtained.

**Meeting 13**

**Date**: 27th September

## Points Discussed

1. modified and updated the report of deliverable1.

**Meeting 14**

**Date**: 29th September

## Points Discussed

1. Actors and the main functionalities of the project are discussed.
2. Use case diagram for the project is drawn.

**Meeting 15**

**Date**: 04th October

## Points Discussed

1. Use case diagram is further analyzed and sequence diagram is discussed.
2. The functional and non-functional requirements are discussed.

**Meeting 16**

**Date**: 06th October

## Points Discussed

1. System architecture is discussed.

**Meeting 17**

**Date**: 11th October

## Points Discussed

1. Preparation of deliverable 2 documentations.
2. **Progress Report**

Online Cake Delivery System provides the user deliver of the cake at the given place.

**Things Done:**

Planning, Requirement Analysis and design are completed and documented successfully from September th,2015 to October 4th,2015.

Currently development of various GUI's in our project like Login page, cakes available information

**Things to Do:**

Coming up are establishing database connectivity, developing servlets, database tables design, unit testing, integration testing and deployment.

Our Online Cake Delivery is expected to be working with full potential by November 26th,2016.

1. **Member Contribution Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Member name | Contribution description | Overall Contribution (%) | Note  (if applicable) |
| Anvesh | User classes characteristics  Class Diagram  Project planning  Use case Diagram  System Architecture | 25% |  |
| Ashik | Sequence diagram  Class Diagram  Use case Diagram  Deliverable-II document  Gantt Chart | 25% |  |
| Srikanth | Class diagram  Test planning  Meeting of Minutes  SRS(functional requirements)  System Architecture | 25% |  |
| Sudhira | Sequence Diagram  Project summary report  Functional and non-functional requirements  SRS  Member contribution table | 25% |  |