**Online Cake Ordering Application**

**Software Requirements Specification**

Version 1.0



**Group Id: S25PROJECT6FDAA**

**Supervisor Name : Akmal Khan**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date (dd/mm/yyyy) | Version | Description | Author |
| 10/05/2024 | 1.0 | This Software Requirements Specification (SRS) outlines the functional and non-functional requirements for the “Online Cake Ordering Application”, a web-based platform designed to streamline the process of searching, customizing, and ordering cakes. The system aims to provide an interactive and user-friendly interface for customers to personalize cakes according to their preferences, place orders, and track delivery status. It also enables administrators to manage cake listings, handle customer queries, confirm orders, and generate financial reports.  Based on the VU Process Model and adopting a hybrid approach combining the Waterfall and Spiral methodologies, this SRS includes detailed use case scenarios, system limitations, and workflow diagrams to ensure a robust, secure, and efficient online cake ordering system. The goal is to deliver a seamless experience to both customers and administrators while ensuring timely service and business efficiency. | BC220213331,BC210406286 |

**Table of Contents**

1. [Scope (of the project)](file:/C:/Users/student/Desktop/ak.as.docx#scope) Page No**: 04**
2. [Functional Requirements Non Functional requirements](file:/C:/Users/student/Desktop/ak.as.docx#FRNFR)  Page No: **05,06,07**
3. [Use Case Diagram](file:/C:/Users/student/Desktop/ak.as.docx#UCD) Page No: **08**
4. [Usage Scenarios](file:/C:/Users/student/Desktop/ak.as.docx#UCS)  Page No: **09**
5. [Adopted Methodology](file:/C:/Users/student/Desktop/ak.as.docx#Adopted)  Page No: **14**
6. [Work Plan (Use MS Project to create Schedule/Work Plan)](file:/C:/Users/student/Desktop/ak.as.docx#Gantt)\_\_ \_\_\_\_\_\_Page No: **18**

**CS619 PHASE-1**

**SRS DOCUMENT**

1. **Scope Of Project**

The primary objective of this project is to design and develop a dynamic, interactive, and user-focused web-based application titled “Online Cake Ordering Application.” This platform will serve as a comprehensive digital solution for managing cake orders, offering a seamless experience to both customers and administrators. The system aims to simplify the process of browsing, customizing, and ordering a variety of cakes while ensuring efficient order management and timely delivery services.

The proposed application will enable customers to register accounts, manage personal profiles, explore a diverse catalog of cakes organized by categories, and apply filters based on cake type, price, or other specifications. Customers can add selected cakes to a cart, customize them according to their preferences, and place orders by submitting relevant delivery and payment information. They will also have the ability to upload payment vouchers, track order status in real-time, and request special or customized cakes by specifying design, ingredients, and additional instructions. Customer interactions with the system will be guided through a responsive and intuitive user interface accessible across both desktop and mobile devices.From an administrative perspective, the application will empower the admin with full control over platform operations. Admins will be able to manage cake categories and inventory, update product details, oversee and verify customer orders and payment vouchers, and provide estimated delivery times. Additionally, the admin can review and respond to custom cake requests, handle customer queries, manage business expenses, and generate detailed financial reports, including income and profit/loss statements.

The platform will support two primary user roles: Customers and Administrators. Customers will engage with the system to explore, customize, and order cakes, while Administrators will maintain operational integrity and ensure a smooth order fulfillment process. The system will incorporate secure login mechanisms, role-based access control, efficient data handling, and a well-structured workflow to maintain order accuracy and business transparency.

This project aims to offer a scalable, secure, and user-friendly cake ordering system built on modern web technologies. By emphasizing personalization, efficient order processing, and administrative control, the Online Cake Ordering Application will address real-world bakery and customer needs while enhancing the convenience and reliability of cake ordering and delivery services.

1. **A)Functional Requirements:-**

**1: Customer Module:-**

1. The customer can register by providing all the personal details like, name, password, address, phone no etc.

2. The customer can login/logout.

3. After the login, the customer can update their profile.

4. The customer can browse and view detailed information about all the available cakes like cake id, cake name, price, picture etc.

5. The customer can search different cakes based on their category or type.

6. The customer can select one or multiple cakes from the same or different categories.

7. The customer can add the selected cakes to the cart for review before purchase.

8. The customer can update the cart by adding some new cakes or deleting the selected cakes and as a result total price/individual cake price should also be updated automatically.

9. At the time of ordering, the customer should provide all the details like his/her name, address, phone no. etc. for delivery purpose.

10. Once the order is placed, the customer can view and download the order voucher in which all the ordered items, total cost, and payment instruction must be visible to the customer.

11. The customer can upload the paid voucher as a proof of payment for order confirmation.

12. The customer can check the status of their order, including processing, confirmed, dispatched, delivered.

13. The customer can track his/her order for estimated delivery time.

14. The customer can submit a request to admin for a special or customized cake, specifying design, ingredients, and any other preferences.

15. The customer can view the admin feedback or response on their special cake request, including acceptance, suggestions, or modifications.

**2: Admin Module: -**

1. The Admin can login/logout.

2. The Admin can manage cake categories including adding, deleting, and updating different cake categories.

3. The Admin can add new cakes under specific category. At the time of addition, all the details should be provided like, cake id, cake name, price, picture, description, etc.

4. The admin can update/delete the cake details.

5. The admin can view all customers’ orders and can accept/reject the customer orders based on the availability of cakes.

6. The admin can view the paid voucher submitted/uploaded by the customers and verify the vouchers.

7. The admin can confirm the order and at the time of confirmation, the admin should add the total estimated time of the order.

8. The admin can generate a payment voucher for the customer or allow the system to automatically generate one when an order is placed.

9. The admin can view special and customized cake request submitted by the customers.

10. The admin can provide feedback/suggestions on the customer requests for special or customize cake.

11. The admin can manage business expenses related to cake production and delivery.

12. The admin can generate financial reports, including income, profit and loss reports.

1. **Non Functional Requirements:-**

**1. Security:**

This includes requirements related to the protection of the system and its data from unauthorized access, as well as the ability to detect and recover from security breaches.

**2. Usability:**

This includes requirements related to the ease of use and understandability of the system for the end-users.

**4. Maintainability:**

This includes requirements related to the ease of maintaining the system, including testing, debugging, and modifying the system.

**5. Portability:**

This includes requirements related to the ability of the system to be easily transferred to different hardware or software environments.

**6. Compliance:**

This includes requirements related to adherence to laws, regulations, industry standards, or company policies.

**7.Performance:**

The system must respond to user inputs within 2 seconds and handle a minimum of 100 concurrent users.

**8. Accessibility:**

The system must comply with accessibility standards for users with disabilities. The system must comply with accessibility standards for users with disabilities.

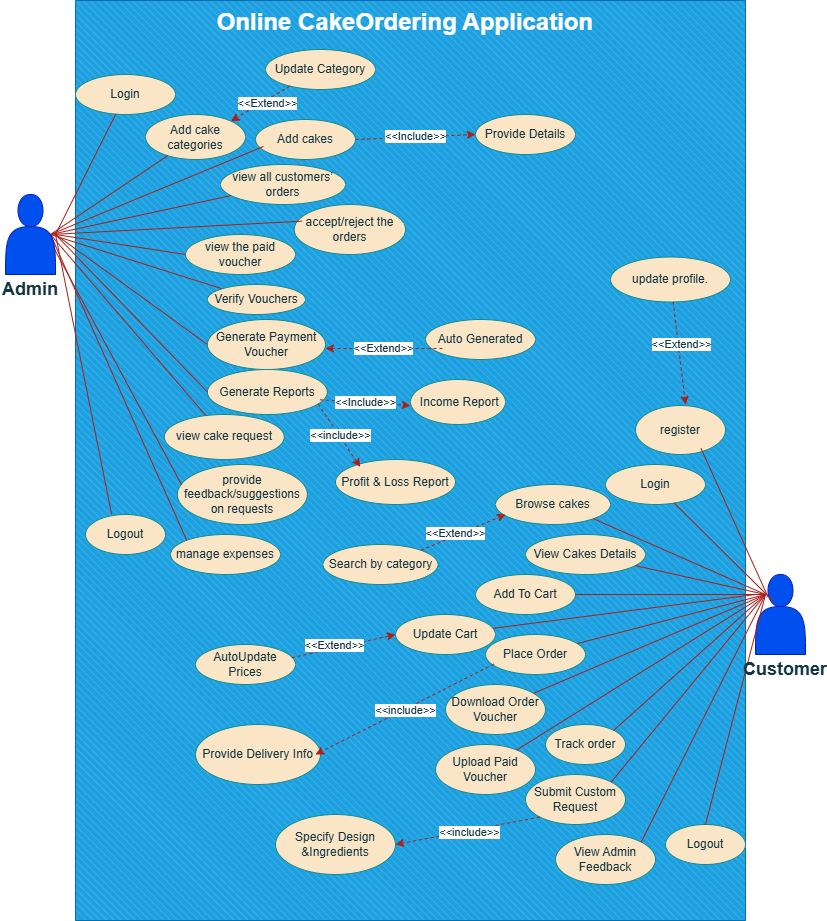
**9. Scalability:**

The system must be able to handle a 20% increase in users and transactions without a significant decrease in performance.

**10. Audit ability:**

The system must provide a clear audit trail of all transactions and updates.

1. **Use Case Diagram(s):-**



1. **Usage Scanerios:-**

Use Case Scenario Table#1: Cake Categories Management

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Actors | Description | Alternative Path | Pre Conditions | Action | Post Conditions | Author | Exception | Version |
| UC1 | Admin | Add new cake categories. Enables categorization of cakes. | - Category already exists. - System error. | Admin is logged in. Admin on category management page. | 1. Admin navigates to category section.  2. Clicks "Add Category".  3. Enters category name/details. 4. Clicks save. | New category is saved and visible. | BC220213331,BC210406286 | Invalid input. Database error. | V 1.1. |
| UC2 | Admin | Update cake categories to reflect changes. | - Invalid ID. - System unavailable. | Admin is logged in. Category exists. | 1. Admin accesses category list. 2. Selects category to edit. 3. Updates details. 4. Clicks update/save. | Category updated successfully. | BC220213331,BC210406286 | Input error. | V 1.1. |

### Use Case Scenario Table#2: Cake Management

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Actors | Description | Alternative Path | Pre Conditions | Action | Post Conditions | Author | Exception | Version |
| UC3 | Admin | Add new cakes with details. | - Image upload failure. - Duplicate entry. | Admin is logged in. At add cake page. | 1. Admin accesses "Add Cake".  2. Fills in name, category, description, image, price, etc.  3. Clicks submit. | Cake listed in catalog. | BC220213331,BC210406286 | File format error. | V 1.1. |
| UC4 | Customer | View cake details including price and ingredients. | - Data loading failure. | User on website/app. Cakes are available. | 1. User browses cakes.  2. Clicks on a cake to view full details. | Cake details displayed. | BC220213331,BC210406286 | Data retrieval error  . | V 1.1. |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Actors | Description | Alternative Path | Pre Conditions | Action | Post Conditions | Author | Exception | Version |
| UC5 | Customer | Add selected cake to cart. | - Session expired. - Stock unavailable. | Customer logged in. Cake available. | 1. User clicks "Add to cart". 2. System confirms addition. | Cake added to cart. | BC220213331,BC210406286 | Item not added. | V 1.1. |
| UC6 | Customer | Update cart (change quantity or remove item). | - System crash. | Item exists in cart. | 1. User navigates to cart.  2. Adjusts quantity or removes item. 3. System updates cart. | Updated cart state saved. | BC220213331,BC210406286 | Invalid quantity. | V 1.1. |

### Use Case Scenario Table#3: Cart Management

### Use Case Scenario Table#4: Orders & Vouchers

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Actors | Description | Alternative Path | Pre Conditions | Action | Post Conditions | Author | Exception | Version |
| UC7 | Customer | Place cake order. | - Payment failure. - Delivery info missing. | Cart has items. Customer is logged in. | 1. User reviews cart.  2. Proceeds to checkout.  3. Enters delivery info.  4. Confirms order. | Order is saved with status 'Pending'. | BC220213331,BC210406286 | Order not submitted. | V 1.1. |
| UC8 | Admin | View all orders, Accept/Reject. | - Order not found. | Admin logged in. | 1. Admin navigates to orders.  2. Reviews order. 3.Accepts/Rejects. | Order status updated. | BC220213331,BC210406286 | Update failed. | V 1.1. |
| UC9 | Customer | Download order voucher after placing order. | - Download link broken. | Order placed. | 1. Customer goes to orders.  2. Clicks download voucher. | Voucher downloaded. | BC220213331,BC210406286 | File not found. | V 1.1. |
| UC10 | Admin | Verify uploaded vouchers. | - Invalid voucher. | Voucher uploaded by customer. | 1. Admin views voucher.  2.Confirms/Rejects validity. | Voucher verified. | BC220213331,BC210406286 | Invalid file. | V 1.1. |
| UC11 | Admin | Auto-generate payment vouchers on accepted orders. | - System delay. | Order is accepted. | 1. System checks order status.  2. Generates payment voucher. | Voucher available for customer. | BC220213331,BC210406286 | Generation failed. | V 1.1. |

### Use Case Scenario Table#5: Report & Expense Management

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Actors | Description | Alternative Path | Pre Conditions | Action | Post Conditions | Author | Exception | Version |
| UC12 | Admin | Generate reports. | - Report type invalid. | Admin logged in. | 1. Admin navigates to reports.  2. Selects type (Income/ProfitLoss).  3. Enters date range. 4. Generates report. | Report displayed/downloaded. | BC220213331,BC210406286 | Export failed. | V 1.1. |
| UC13 | Admin | Manage expenses (add/edit). | - Duplicate entry. | Admin logged in. | 1. Admin opens expenses tab.  2. Adds/edits entry. 3. Saves changes. | Expenses updated in DB. | BC220213331,BC210406286 | Validation error. | V 1.1. |

### Use Case Scenario Table#6: Custom Requests & Feedback

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Actors | Description | Alternative Path | Pre Conditions | Action | Post Conditions | Author | Exception | Version |
| UC14 | Customer | Submit custom cake request with design & ingredients. | - Upload failure. | Logged in. On custom request page. | 1. User accesses form.  2. Fills design, flavor, ingredients.  3. Submits request. | Request saved. | BC220213331,BC210406286 | Invalid file format. | V 1.1. |
| UC14 | Admin | View request and give feedback/suggestions. | - No request found. | Requests available. | 1. Admin reviews request. 2. Enters suggestions or response.  3. Submits feedback. | Feedback sent to customer. | BC220213331,BC210406286 | Submission error. | V 1.1. |

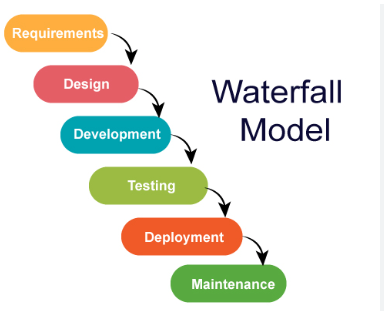
### Use Case Scenario Table#7: User Authentication/Profile

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Actors | Description | Alternative Path | Pre Conditions | Action | Post Conditions | Author | Exception | Version |
| UC15 | Customer | Register new account. | - Email already used. | Guest on registration page. | 1. Guest enters data. 2. Clicks register. | Account created. | BC220213331,BC210406286 | Email exists. | V 1.1. |
| UC16 | Customer | Login to system. | - Wrong credentials. | User account exists. | 1. User enters credentials. 2. Clicks login. | User logged in. | BC220213331,BC210406286 | Login failed. | V 1.1. |
| UC17 | Admin/Customer | Logout of system. | - Session timeout. | User logged in. | 1. User clicks logout. | Session ends. | BC220213331,BC210406286 | Logout failed. | V 1.1. |
| UC18 | Customer | Update profile information. | - Invalid input. | User logged in. | 1. User accesses profile. 2. Updates info. 3. Saves changes. | Profile updated. | BC220213331,BC210406286 | Validation failed. | V 1.1. |

1. **Adopted Methodologies:-**

**1.Waterfall Model:**

The Waterfall model is a linear, sequential software development process where each phase is completed before moving on to the next one, with no overlap or iteration. It follows a rigid structure, requiring fixed and well-defined requirements, and is suitable for projects with complexity and risk.



The six stages above are as follows:

**I. Requirement Analysis and Definition:**

What- The systems services, constraints and goals are established by consultation with system users. They are then defined in detail and serve as a system specification.

**II. System and Software Design:**

How – The system design process partitions the requirements to either hardware of software systems. It establishes and overall system architecture. Software design involves fundamental system abstractions and their relationships

**III.Implementation and Unit Testing: -**

How – During this stage the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specifications.

**IV. Integration and system testing:**

The individual program unit or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.

**V.Deployment:-**

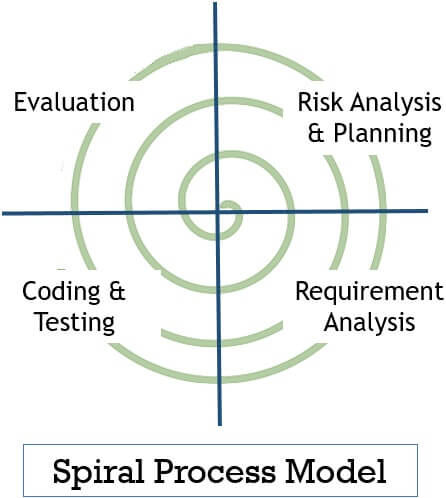
In the Deployment phase, the focus is on ensuring a smooth transition of the software from development to production, and making it available to end-users. This phase involves final testing, packaging, distribution, installation, configuration, user training, and documentation. Once the software is successfully deployed, it is handed over to the maintenance team for ongoing support and maintenance.

**VI. Operation and Maintenance:**

Normally this is the longest phase of the software life cycle. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in earlier stages of the life-cycle, improving the implementation of system units and enhancing the system’s services as new requirements are discovered

**2.Spiral Modal:**

The Spiral Model is a [**Software Development Life Cycle (SDLC)**](https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/) model that provides a systematic and iterative approach to software development. In its diagrammatic representation, looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a **Phase of the**software development.



**I. Objectives determination and identify alternative solutions:**

In this first quadrant, the main goal is to gather as much information as possible from the customers. This includes understanding their needs, expectations, and any constraints they may have. Once the objectives are clearly defined, the team starts brainstorming different solutions that could meet these objectives. These solutions are then evaluated based on their feasibility, cost, time, and alignment with the customer’s requirements. The best solutions are shortlisted for further analysis in the next quadrant.

**II. Identify and resolve Risks:**

The second quadrant is all about risk management. The team evaluates the shortlisted solutions from the first quadrant and identifies potential risks associated with each one. These risks could be technical, financial, operational, or even related to the market or customer preferences. Once the risks are identified, the team develops strategies to mitigate them. This could involve modifying the solution, developing contingency plans, or even discarding the solution if the risks are too high. The end result of this quadrant is a prototype of the chosen solution, which is then tested and refined in the next quadrant.

Develop the next ID of the Product:

**III. The third quadrant is where the actual development happens.**

Based on the prototype from the second quadrant, the team starts building the features of the product. This involves coding, testing, and debugging. The team also verifies that the product meets the objectives defined in the first quadrant and that the risks identified in the second quadrant have been effectively mitigated. By the end of this quadrant, a new ID of the product is ready for review.

**IV. Review and plan for the next Phase:**

The fourth and final quadrant involves reviewing the product with the customers. They evaluate the product to ensure it meets their needs and provides value. The team collects feedback and uses it to improve the product in the next phase. This quadrant also involves planning for the next phase of the Spiral Model, which could involve scaling the product, adding new features, or even starting a new project.

**Choosen Methodology:-**

I will choose vu process Model for my project which is the combination of waterfall and spiral model.

**3.VU Process Model:**

A structured methodology for process management and improvement, comprising four sequential stages.

**I. Vision (Definition and Goal-Setting)**

Define the desired future state of the process (Vision Statement)

Establish clear goals and objectives (SMART criteria)

Identify key performance indicators (KPIs) to measure success

Define the scope and boundaries of the process

Identify stakeholders and their requirements

**II. Understanding (Current State Assessment)**

Analyze the current state of the process (As-Is process mapping)

Identify strengths, weaknesses, opportunities, and threats (SWOT analysis)

Gather data and feedback from stakeholders (surveys, interviews, etc.)

Identify pain points and areas for improvement

Document the current process (process mapping, procedures, etc.)

**III. Upgrading (Process Improvement and Implementation)**

Design and implement process improvements (To-Be process mapping)

Develop new procedures and standards

Train and support stakeholders

Develop a change management plan

Implement the new process (transition from As-Is to To-Be)

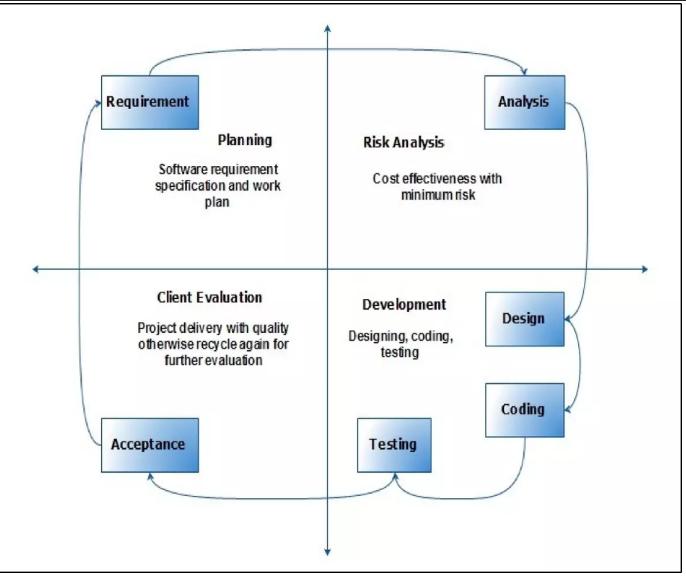
**IV. Validation (Monitoring and Evaluation)**

Monitor and evaluate the improved process

Measure KPIs and adjust as needed

Continuously review and refine the process to ensure sustainability

Celebrate successes and recognize improvements



Reasons For Choosing:-

**I.Structured Approach:**

The VU Process Model emphasizes a structured approach to software development. It ensures that you follow a systematic sequence of steps, from requirements gathering to testing and deployment.

**II. Early Specifications and Design:**

Before writing a single line of code, the VU Process Model focuses on creating detailed specifications and design documentation. This minimizes wastage of effort and time and reduces the risk of schedule slippage or unmet customer expectations1.

**III. Risk Management:**

Like the Spiral Model, the VU Process Model incorporates risk management. It allows you to identify and address potential risks early in the project. By iteratively assessing risks, you can make informed decisions and adjust your approach as needed.

**IV. Adaptability:**

The VU Process Model allows for flexibility. You can adapt it to suit your project’s specific needs. For instance, if certain requirements change during development, you can adjust the design and specifications accordingly.

**V. Clear Documentation:**

The model encourages comprehensive documentation at each stage. This documentation serves as a valuable reference for team members, stakeholders, and future maintenance.

**VI. Gantt Chart Planning:**

The VU Process Model recommends creating a Gantt chart to visualize project activities and milestones. This helps manage project timelines effectively

1. **Work Plan:-**

