**CHAPTER 1**

**INTRODUCTION**

# 1.1 INTRODUCTION

Documents of RecipeRave is an innovative online platform crafted specifically for passionate culinary enthusiasts seeking a vibrant community to engage with and explore their love for cooking and baking. With its intuitive and user-friendly interface, RecipeRave provides a welcoming space for individuals to discover, share, and celebrate their favorite recipes from around the globe. Users are not only invited to access an extensive collection of diverse recipes but also to actively participate in the thriving community by contributing their own culinary creations and engaging with others through comments, ratings, and discussions. Whether it's a traditional family recipe passed down through generations or a modern twist on a classic dish, RecipeRave encourages users to share their culinary expertise and creativity with like-minded individuals, fostering a collaborative and inspiring environment for culinary exploration and discovery. Modules: 1. Recipe Collections, 2. Cooking Videos, 3. User Authentication, 4. Recipe Management, 5. Search and Filtering, 6. Rating and Comments, 7. Social Sharing, 8. Bookmarking, 9. User Profiles, 10. Follow User, 11. Feedback System, 12. Admin Panel.

1. **Recipe Collections**: This likely includes how recipes are organized and categorized within the platform. It may cover topics such as recipe tagging, categorization by cuisine type or dietary preferences, and organization into collections (e.g., trending recipes, seasonal favorites).

**2. Cooking Videos**: Documentation would include details on how cooking videos are integrated into the platform. This could involve video uploading, embedding, playback controls, and possibly integration with popular video hosting platforms.

**3. User Authentication**: Information on user authentication covers how users securely log in, register, and manage their accounts. It includes aspects such as password management, account recovery, and possibly integration with social media for authentication.

**4. Recipe Management**: This module involves how recipes are created, edited, and managed by users. It would include features such as recipe editing tools (text formatting, image uploading), version control, and moderation capabilities.

**5. Search and Filtering**: Documentation would outline how users can search for recipes using keywords, filters (e.g., dietary preferences, cooking time), and sorting options (e.g., by popularity, date added).

**6. Rating and Comments**: Details on how users can rate recipes and leave comments or reviews. This would cover features like star ratings, comment moderation, and possibly threaded discussions.

**7. Social Sharing**: Documentation on how users can share recipes or cooking videos on social media platforms. This could include integration with social sharing APIs and tracking of shares.

**8. Bookmarking**: Information on how users can save recipes to their personal collections or bookmark them for later viewing. This might include features like favorite lists or personalized recipe boxes.

**9. User Profiles**: Documentation on user profiles covers what information users can display (e.g., bio, profile picture), how profiles are managed, and privacy settings.

**10. Follow User**: Features allowing users to follow other users to see their activity, such as recipe uploads, comments, and ratings.

1. **Feedback System**: This involves how users can provide feedback to RecipeRave, including reporting

issues, suggesting features, and general feedback mechanisms.

1. **Admin Panel**: Documentation for administrators on managing the platform, including user

management, content moderation, analytics, and system configuration.

For each module, you would typically find detailed specifications, API documentation (if applicable), user interface mockups, and possibly implementation notes. These documents help developers, designers, and stakeholders understand how each feature should function and how they integrate with other parts of the platform. If you're developing or managing RecipeRave, having these documents ensures clarity and consistency in the platform's development and user experience.

Top of Form

Bottom of Form

**1.2 SCOPE**

The phrase "recipe rave" typically refers to the enthusiastic praise or positive reception of a particular recipe. It suggests that a recipe has garnered significant admiration or acclaim from those who have tried it. The scope of a recipe rave can include various aspects:

1. **Recipe Quality**: It reflects how well the recipe works in terms of taste, texture, and overall outcome. A rave indicates that people find the recipe delicious and enjoyable.
2. **Ease of Preparation**: A recipe that receives a rave often includes clear instructions and is relatively easy to follow, even for those who may not be experienced cooks.
3. **Ingredients**: People appreciate recipes that use accessible ingredients or combinations that result in unexpectedly good flavors.
4. **Versatility**: Some recipes are raved about because they can be easily adapted to suit different dietary preferences or ingredient availability.
5. **Cultural or Regional Appeal**: Recipes that capture the essence of a particular cuisine or region can often receive raves from those who appreciate authentic flavors.
6. **Online Reviews and social media**: In the digital age, recipe raves can spread quickly through online platforms, blogs, and social media, amplifying their popularity.

In summary, the scope of a recipe rave encompasses the overall satisfaction and positive feedback received from those who have tried and enjoyed a particular recipe. It's a testament to the recipe's appeal and its ability to resonate with a wide audience.

# 1.3 OVERVIEW OF THE PROJECT

The Recipe Rave project typically involves creating a platform or a section within a larger project dedicated to showcasing and reviewing recipes. Here’s an overview of what such a project might include:

1. **Purpose**:
   * The main goal is to provide a space where users can discover, share, and review recipes. It aims to foster a community around cooking and culinary exploration.
2. **Features**:
   * **Recipe Database**: A central repository of recipes categorized by cuisine, dietary preferences, ingredients, etc.
   * **User Profiles**: Users can create profiles to save favorite recipes, follow other cooks, and track their own cooking journeys.
   * **Recipe Reviews and Ratings**: Users can rate and review recipes they've tried, providing feedback and tips.
   * **Search and Filter**: Robust search functionality with filters (e.g., vegetarian, gluten-free, quick recipes) to help users find recipes tailored to their needs.
   * **Cooking Tips and Tricks**: Articles or sections offering culinary advice, cooking techniques, and kitchen hacks.
   * **Community Interaction**: Forums, comment sections, or social media integration where users can discuss recipes, ask questions, and share their experiences.
   * **Featured Content**: Highlighting popular recipes, chef spotlights, seasonal picks, or trending cuisines.
   * **Mobile Compatibility**: Ensuring the platform is responsive and accessible via mobile devices for convenience in the kitchen.
3. **Technical Aspects**:
   * **Backend**: Database management, user authentication, recipe storage, and API integrations for functionalities like search and social sharing.
   * **Frontend**: User interface design focusing on usability, intuitive navigation, and engaging visuals (including high-quality food photography).
   * **Security**: Ensuring data security, especially for user profiles and personal information.
   * **Scalability**: Designing the platform to handle increased traffic and content as the community grows.
4. **Monetization Strategies**:
   * **Premium Features**: Offering advanced search filters, ad-free browsing, or exclusive recipes for a subscription fee.
   * **Advertising**: Partnering with food brands or kitchen appliance companies for targeted advertisements.
   * **Affiliate Marketing**: Recommending cooking tools or ingredients with affiliate links.
   * **Merchandise**: Selling branded merchandise like aprons, kitchen gadgets, or cookbooks.
5. **Content Strategy**:
   * Regular updates with new recipes, seasonal collections, and themed content (e.g., holiday recipes, celebrity chef collaborations).
   * Engaging with the community through newsletters, social media updates, and user-generated content campaigns.
6. **Legal Considerations**:
   * Copyright and attribution for recipes and user-generated content.
   * Terms of service and privacy policies to protect user data and intellectual property.
7. **Launch and Growth**:
   * Beta testing to gather feedback from early users.
   * Marketing strategies to attract initial users, such as social media campaigns, partnerships with influencers, or SEO optimization.

By focusing on these aspects, the Recipe Rave project can create a vibrant online community centered around cooking, sharing recipes, and celebrating culinary creativity.

Top of Form

Bottom of Form

**Module Details**

"Recipe Rave" appears to refer to a specific module or feature related to recipes. While the exact details can vary depending on the context or platform, here are some common modules or features that such a system might include:

1. **Recipe Database**: A collection of various recipes categorized by type (e.g., appetizers, mains, desserts) or dietary preferences (e.g., vegetarian, gluten-free).
2. **Search and Filter**: Functionality that allows users to search for recipes based on keywords (e.g., ingredients, dish names) and apply filters (e.g., cooking time, difficulty level).
3. **Recipe Details**: Each recipe typically includes details such as ingredients, measurements, preparation steps, cooking time, and serving size.
4. **User Ratings and Reviews**: Enables users to rate recipes and leave reviews or comments based on their cooking experiences.
5. **Favorites and Collections**: Users can save their favorite recipes or create collections to organize recipes for specific occasions or preferences.
6. **Shopping List**: Option to generate a shopping list based on selected recipes, which lists all necessary ingredients.
7. **Nutritional Information**: Provides nutritional facts for each recipe, such as calories, protein, carbohydrates, and fat content.
8. **Meal Planning**: Tools for users to plan their meals by selecting recipes for specific days, weeks, or events.
9. **Social Sharing**: Integration with social media platforms to share recipes, cooking experiences, or photos.
10. **Cooking Tips and Videos**: Additional resources like tips, tricks, and instructional videos to assist users in preparing recipes.

The specific modules or features can vary based on the platform (website, app) and the target audience (home cooks, professional chefs, health-conscious individuals). If you have more details or a specific platform in mind, I can provide more tailored information.Top of Form

Bottom of Form

**CHAPTER 2**

**SYSTEM ANALYSIS**

System analysis is a logical process; it is an important activity that takes place when a new system is being built. The objective of this phase is not actually to solve the problem but to determine what must be done to solve the problem. It is the central intact of system development and it includes gathering and interpreting facts, diagnosing and using this information to recommend improvements to the system. System analysis gives the structure and functioning of the system and it specifies what the system should do. It is helpful to understand the problem and emphasize what is needed from the system.

Before designing an application, which will help its users, it is important that the way it currently operates should be clearly identified. The process of system investigation includes several methods of gathering the required information. It is important that the approach is appropriate to the application under consideration and the user of the system. System analysis should be creative and imaginative in producing new solutions to meet the user requirements. In short, the system is analyzed by gathering various information such as system functionality, problems encountered, requirement by the proposed system, users, their tasks and responsibilities. Here in the system analysis phase, system is viewed as a whole and inputs to the system are in defined, and also the output from the system is traced through the various data collected.

A detailed study of these processes must be made by the various techniques like interviews, questionnaires, etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now, the existing system is subject to close study and the problem areas are identified. The designer now functions on a problem solver and tires to sort out the difficulties that weighed with the existing faces. The solutions a given on a proposal. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for any endorsement by the user. The proposal is reviewed on user request and suitable changes that made. This is a loop that ends as soon as the user in satisfied with the proposal.

# 2.1 EXISTING SYSTEM

The existing system of RecipeRave is designed to cater specifically to culinary enthusiasts who are passionate about cooking and baking. Here's a detailed overview based on the description provided:

1. **Purpose and Audience**:
   * RecipeRave aims to serve as a community-driven platform where users can engage with a diverse range of recipes. It targets individuals who are not only interested in discovering new dishes but also in sharing their own culinary creations.
2. **Features**:
   * **Recipe Repository**: Offers a wide array of recipes from various cuisines and dietary preferences, allowing users to explore and find recipes that suit their tastes.
   * **User Interaction**: Encourages active participation through features like comments, ratings, and discussions. This fosters a sense of community and enables users to share feedback and cooking tips.
   * **User Contributions**: Allows users to contribute their own recipes, whether they're traditional family favorites or innovative twists on classic dishes. This promotes a collaborative environment where culinary expertise and creativity are celebrated.
   * **Community Engagement**: Provides a platform for users to connect with like-minded individuals who share their passion for cooking. This interaction enhances the overall user experience by facilitating knowledge exchange and inspiration.
3. **User Interface and Experience**:
   * **Intuitive Design**: Boasts a user-friendly interface that makes it easy for users to navigate through recipes, engage with content, and contribute their own recipes seamlessly.
   * **Visual Appeal**: Likely includes high-quality food photography and visually appealing presentation of recipes to enhance user engagement and inspire culinary exploration.
4. **Content and Diversity**:
   * **Global Cuisine Coverage**: Offers recipes from around the globe, catering to diverse culinary tastes and preferences.
   * **Recipe Variety**: Includes a broad range of recipes, from traditional dishes to contemporary creations, ensuring there's something for every user's palate and skill level.
5. **Community Building**:
   * **Collaborative Environment**: Promotes a sense of belonging and community among users who share a passion for cooking. This is facilitated through features that encourage interaction and the sharing of knowledge and experiences.
6. **Future Growth and Development**:
   * Likely focuses on expanding the recipe database, improving user engagement features, and possibly integrating advanced functionalities such as personalized recipe recommendations or cooking challenges.

In summary, RecipeRave positions itself as more than just a repository of recipes; it's a vibrant community where culinary enthusiasts can connect, discover, share, and celebrate their love for cooking and baking in a welcoming and inspiring environment.

## 2.1.1 Limitations of Existing System

1. **Limited User Verification:**
   * While RecipeRave allows users to contribute their own recipes and engage with the community, there is no robust verification system to ensure the authenticity and accuracy of user-contributed recipes and reviews. This can lead to potential issues with the quality and reliability of content.
2. **Lack of Personalized Recommendations:**
   * The current system does not offer personalized recipe recommendations based on user preferences, dietary restrictions, or past interactions. This can make it challenging for users to discover new recipes that align with their specific tastes and needs.
3. **Insufficient Advanced Search and Filtering:**
   * Although RecipeRave includes basic search and filtering options, they are not comprehensive enough to cater to all user requirements. For example, there is a need for more advanced filtering based on cooking time, ingredient availability, and skill level.
4. **Limited Community Engagement Tools:**
   * While the platform encourages community engagement through comments and ratings, it lacks more interactive features such as live cooking classes, real-time chat, and community challenges or events that could further enhance user interaction and engagement.Bottom of Form

# 2.2 PROPOSED SYSTEM

**Enhanced User Engagement Features**:

* **Live Cooking Sessions**: Introduce live streaming capabilities where chefs and users can demonstrate recipes in real-time, fostering interactive learning and engagement.
* **Cooking Challenges**: Regular themed challenges (e.g., weekly ingredients, seasonal dishes) to encourage creativity and participation within the community.
* **Virtual Cooking Classes**: Offer structured online classes hosted by professional chefs or experienced home cooks, covering various cuisines and techniques.

## 2.2.1 ADVANTAGES OF PROPOSED SYSTEM

* **Personal Recipe Collections**: Allow users to create personalized recipe collections (e.g., favorites, saved for later, meal plans) for easy access and organization.
* **Recipe Versioning**: Enable users to update or revise their posted recipes, with clear version history to track changes and improvements over time.
* **Recipe Recommendations**: Implement personalized recipe suggestions based on user preferences, past interactions, and trending recipes within the community.

**Community and Social Features**:

* **Enhanced Discussion Forums**: Create dedicated discussion boards or forums where users can ask questions, share tips, and discuss culinary topics beyond specific recipes.
* **User Profiles**: Expand user profiles to include badges or achievements based on contributions and community engagement, fostering a sense of accomplishment and recognition.
* **Social Sharing Integration**: Streamline sharing of recipes and community updates across popular social media platforms to expand RecipeRave reach and user base.

**Interactive Multimedia Content**:

* **Video Recipe Tutorials**: Enhance recipe pages with embedded video tutorials for step-by-step guidance, enhancing user experience and learning.
* **Interactive Recipe Cards**: Include interactive elements on recipe pages, such as ingredient substitution suggestions, nutritional information, and cooking tips.
* **User-Generated Content Spotlight**: Feature user-submitted photos, videos, and stories alongside recipes to showcase community creativity and inspire others.

**Mobile Optimization and Accessibility**:

* **Responsive Design**: Ensure seamless user experience across devices, optimizing for mobile access to cater to users accessing recipes while cooking in the kitchen.
* **Offline Access**: Implement offline mode or downloadable recipe collections for users to access recipes without an internet connection, ideal for cooking enthusiasts on the go.

**Monetization Strategies**:

* **Premium Memberships**: Offer premium features like ad-free browsing, exclusive content (e.g., advanced cooking classes), and early access to new features.
* **Sponsored Content and Partnerships**: Collaborate with food brands, kitchen equipment manufacturers, or culinary schools for sponsored content, contests, or featured placements.

**Data Analytics and Insights**:

* **User Engagement Metrics**: Utilize analytics to track user interactions, popular recipes, and community trends, informing content curation and platform improvements.
* **Feedback Loops**: Implement surveys, polls, and user feedback mechanisms to gather insights and continuously improve user satisfaction and platform functionality.

**Scalability and Security**:

* **Scalable Infrastructure**: Design a robust backend infrastructure capable of handling increased traffic, user-generated content, and future growth.
* **Data Security**: Ensure adherence to data protection regulations and best practices for safeguarding user information and transactions.

### Conclusion:

The proposed enhancements for RecipeRave aim to elevate the user experience by fostering deeper engagement, encouraging creativity, and expanding learning opportunities within a vibrant and supportive culinary community. By integrating advanced features and user-centric improvements, RecipeRave can solidify its position as a premier online platform for culinary enthusiasts seeking inspiration, knowledge sharing, and connection through the art of cooking and baking.

Top of Form

Bottom of Form

.

# 2.3 FEASIBILITY STUDY

To conduct a feasibility study for RecipeRave, we need to assess its technical, economic, operational, and scheduling feasibility. Here’s how each aspect could be evaluated:

### 2.3.1 Technical Feasibility:

* **Platform Development**: Assess the technical capabilities and resources required to build and maintain RecipeRave. This includes frontend development (user interface design, interactive features), backend development (database management, server infrastructure), and potential integration with third-party services (e.g., social media, analytics).
* **Scalability**: Ensure the platform can handle increasing user traffic and content uploads without compromising performance. Consider cloud-based solutions for scalability.
* **Mobile Compatibility**: Confirm compatibility across various devices (smartphones, tablets) to cater to user convenience, especially in the kitchen.

### 2.3.2 Economic Feasibility:

* **Cost Analysis**: Estimate development costs, including software, hardware, and infrastructure expenses. Consider ongoing operational costs such as server maintenance, content moderation, and customer support.
* **Revenue Generation**: Identify potential revenue streams, such as premium memberships, sponsored content, affiliate marketing, or partnerships with culinary brands. Evaluate the market size and potential profitability based on user engagement and monetization strategies.

### 2.3.3 Operational Feasibility:

* **Resource Availability**: Assess the availability of skilled personnel (developers, designers, content creators) to build and manage RecipeRave. Consider outsourcing options if necessary.
* **User Adoption**: Conduct market research or surveys to gauge user interest and validate the platform’s appeal among culinary enthusiasts. Identify potential user acquisition strategies and community-building initiatives.
* **Content Management**: Plan for efficient content moderation and curation to maintain quality and relevance of recipes and user-generated content.

### 2.3.4 Scheduling Feasibility:

* **Timeline**: Develop a realistic project timeline considering phases such as design, development, testing, launch, and post-launch iterations based on user feedback.
* **Milestones**: Define key milestones and deliverables to track progress and ensure alignment with business objectives.
* **Risk Assessment**: Identify potential risks and challenges (e.g., technical issues, market competition, regulatory compliance) and develop contingency plans to mitigate them.

Top of Form

Bottom of Form

# 2.4 MODULE DESCRIPTION

Reciperave as an innovative online platform for culinary enthusiasts, here are the module descriptions that would outline its key functionalities and components:

### 1. ****User Authentication and Management Module****

* **Description**: This module handles user registration, login, and profile management functionalities.
* **Features**:
  + User registration and login via email or social media accounts.
  + User profile management where users can update personal information and preferences.
  + Account security features like password management and account settings.

### 2. ****Recipe Management Module****

* **Description**: Manages the core functionality of adding, editing, and browsing recipes.
* **Features**:
  + **Recipe Creation**: Allows users to create new recipes by filling out fields such as ingredients, instructions, cooking time, dietary tags, and optional photos or videos.
  + **Recipe Editing**: Enables users to edit or update their own recipes post-submission.
  + **Recipe Search and Filtering**: Provides robust search and filtering options (by cuisine, dietary preferences, ingredients) for users to discover recipes.
  + **Recipe Categories**: Organizes recipes into categories and tags for easier browsing.

### 3. ****Community Interaction Module****

* **Description**: Facilitates user engagement through comments, ratings, and discussions.
* **Features**:
  + **Comments and Ratings**: Allows users to leave comments and rate recipes they have tried.
  + **Discussion Forums**: Provides forums or discussion boards where users can initiate or participate in discussions on various culinary topics beyond specific recipes.
  + **User Interaction**: Supports following other users, liking recipes, and sharing recipes on social media platforms.

### 4. ****Content Management Module****

* **Description**: Manages the overall content on the platform to ensure quality and relevance.
* **Features**:
  + **Content Moderation**: Enables moderation of user-generated content to maintain quality standards and adherence to community guidelines.
  + **Featured Content**: Highlights popular or trending recipes, chef spotlights, and seasonal picks.
  + **Content Curation**: Curates collections of recipes based on themes, occasions, or trending topics.

### 5. ****Social Integration Module****

* **Description**: Integrates social media features to enhance user interaction and platform visibility.
* **Features**:
  + **Social Sharing**: Allows users to share recipes and platform content on external social media platforms.
  + **Social Login**: Facilitates user registration and login using existing social media accounts.
  + **Social Engagement**: Tracks and displays social engagement metrics (likes, shares, comments) for recipes and platform content.

### 6. ****Analytics and Reporting Module****

* **Description**: Provides insights into platform usage and user behavior.
* **Features**:
  + **User Activity Tracking**: Monitors user interactions such as recipe views, comments, ratings, and user-generated content submissions.
  + **Performance Metrics**: Generates reports on key performance indicators (KPIs) like user engagement, popular recipes, and community growth.
  + **Analytics Dashboard**: Presents data visualizations and metrics to facilitate decision-making and optimization of platform features.

### 7. ****Admin Panel Module****

* **Description**: Administers and manages the platform’s operations and user data.
* **Features**:
  + **User Management**: Admin capabilities to manage user accounts, permissions, and access levels.
  + **Content Moderation Tools**: Tools for reviewing and moderating user-generated content, handling reported content, and enforcing community guidelines.
  + **Analytics and Reporting**: Access to platform analytics and reporting tools for monitoring and strategic planning.

### Conclusion:

Each module described above plays a crucial role in Recipe Rave's functionality, catering to both user engagement and platform management. Together, these modules create a cohesive and interactive environment where culinary enthusiasts can explore, share, and celebrate their passion for cooking and baking in a collaborative community setting.

**CHAPTER 3**

**SYSTEM SPECIFICATION**

# 3.1 SYSTEM SPECIFICATIONS

## 3.1.1 HARDWARE SPECIFICATIONS

The selection of hardware configuration is a very important task related to software development. Insufficient random-access memory may affect adversely on speed and efficiency of the entire system. The process should be powerful to handle the entire operations.

The hard disk should have sufficient capacity to store the file and application**.**

* Processor : 1.7GHz i3 Processor or above
* Hard Disk : 550 GB or above
* RAM : 2.00 GB or above
* Input Device : Standard Mouse and Keyboard
* Output Device : High Resolution Monitor

## 3.1.2 SOFTWARE SPECIFICATIONS

|  |  |  |
| --- | --- | --- |
| Operating System : | | Windows 7 or above |
| Web Server | : | Apache Tomcat |
| Web Brower | : | Microsoft Edge/ Mozilla/Google Chrome |
| Front End | : | PYTHON |
| Back End | : | MYSQL |

# 3.2 SOFTWARE TECHNOLOGY OVERVIEW

**3.2.1 Overview of PYTHON:**

**WINDOWS OS**

**Microsoft Windows**, or simply **Windows**, is a meta family of [graphical](https://en.wikipedia.org/wiki/Graphical_user_interface) [operating systems](https://en.wikipedia.org/wiki/Operating_system) developed, marketed, and sold by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It consists of several families of operating systems, each of which cater to a certain sector of the computing industry with the [OS](https://en.wikipedia.org/wiki/Operating_system) typically associated with [IBM PC compatible](https://en.wikipedia.org/wiki/IBM_PC_compatible) architecture. Active Windows families include [Windows NT](https://en.wikipedia.org/wiki/Windows_NT) and [Windows Embedded](https://en.wikipedia.org/wiki/Windows_Embedded); these may encompass subfamilies, e.g. [Windows Embedded Compact](https://en.wikipedia.org/wiki/Windows_Embedded_Compact) (Windows CE) or [Windows Server](https://en.wikipedia.org/wiki/Windows_Server). Defunct Windows families include [Windows 9x](https://en.wikipedia.org/wiki/Windows_9x), [Windows Mobile](https://en.wikipedia.org/wiki/Windows_Mobile) and [Windows Phone](https://en.wikipedia.org/wiki/Windows_Phone).

Microsoft introduced an [operating environment](https://en.wikipedia.org/wiki/Operating_environment) named *Windows* on November 20, 1985, as a graphical [operating system shell](https://en.wikipedia.org/wiki/Operating_system_shell) for [MS-DOS](https://en.wikipedia.org/wiki/MS-DOS) in response to the growing interest in [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUIs). Microsoft Windows came to [dominate](https://en.wikipedia.org/wiki/Dominance_(economics)) the world's [personal computer](https://en.wikipedia.org/wiki/Personal_computer) (PC) market with [over 90% market share](https://en.wikipedia.org/wiki/Usage_share_of_operating_systems), overtaking [Mac OS](https://en.wikipedia.org/wiki/Classic_Mac_OS), which had been introduced in 1984. [Apple](https://en.wikipedia.org/wiki/Apple_Inc) came to see Windows as an unfair encroachment on their innovation in GUI development as implemented on products such as the [Lisa](https://en.wikipedia.org/wiki/Apple_Lisa) and [Macintosh](https://en.wikipedia.org/wiki/Macintosh) (eventually settled in court in Microsoft's favor in 1993). On PCs, Windows is still the most popular operating system. However, in 2014, Microsoft admitted losing the majority of the overall operating system market to [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), because of the massive growth in sales of Android [Smartphone](https://en.wikipedia.org/wiki/Smartphone). In 2014, the number of Windows devices sold was less than 25% that of Android devices sold. This comparison however may not be fully relevant, as the two operating systems traditionally target different platforms. Still, numbers for server use of Windows (that are comparable to competitors) show one third market share, similar to for end user use.

As of September 2016, the most recent version of Windows for PCs, [tablets](https://en.wikipedia.org/wiki/Tablet_computers), [Smartphone](https://en.wikipedia.org/wiki/Smartphone) and [embedded devices](https://en.wikipedia.org/wiki/Embedded_system) is [Windows 10](https://en.wikipedia.org/wiki/Windows_10). The most recent versions for [server computers](https://en.wikipedia.org/wiki/Server_(computing)) is [Windows Server 2016](https://en.wikipedia.org/wiki/Windows_Server_2016). [A specialized version of Windows](https://en.wikipedia.org/wiki/Xbox_One_system_software) runs on the [Xbox One](https://en.wikipedia.org/wiki/Xbox_One) [video game console](https://en.wikipedia.org/wiki/Video_game_console).

**HTML**

The language used to develop webpages is called Hypertext Mark-up Language (HTML). HTML is the language interpreted by a browser. HTML is specified as TAGS in an HTML document (i.e. the webpage).

### HTML Tags

Tags are instructions that are embedded directly into the text of the document. An HTML tag is a signal to a browser that it should do something other than just throw text up on the screen. By convention all HTML tags begin with an open angle bracket (<) and end with a close angle bracket (>).

### The Structure of an HTML Program

Every HTML program has a rigid structure. The entire web page is enclosed within <HTML></HTML> tags. Within these tags two distinct sections are created using the <HEAD></HEAD> tags and the <BODY></BODY>tags.

### Cascading Style Sheets (CSS)

CSS is a style sheet language used for describing the look and formatting of a document written in a mark-up language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL, CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).CSS canal so allow the same mark-up page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different stylesheet, perhaps one on their own computer, to override the one of the authors has specified. However, if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

### JavaScript (JS)

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side Programming, game development and the creation of desktop and mobile applications. JavaScript is a prototype-based scripting language with dynamic typing and has first class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the self and scheme programming languages. It is a multi -paradigm language, supporting object-oriented, imperative, and functional programming styles. The application of JavaScript to use outside of web pages--for example, in PDF documents, site-specific browsers, and desktop widgets--is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent (post- 2018) browsers**.**

**PYTHON**

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don’t need to declare the type of variable because it is a dynamic typed language. For example, x=10, here x can be anything such as String, into etc.

**Django**

Django is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) based [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [web framework](https://en.wikipedia.org/wiki/Web_framework), which follows the model-template- view (MTV) [architectural pattern](https://en.wikipedia.org/wiki/Architectural_pattern_(computer_science)). Django's primary goal is to ease the creation of complex, database driven websites. The framework

Also included in the core framework are:

* A lightweight and standalone web server for development and testing.
* A form serialization and validation system that can translate between [HTML](https://en.wikipedia.org/wiki/HTML) forms and values suitable for storage in the database.
* A template system that utilizes the concept of [inheritance](https://en.wikipedia.org/wiki/Inheritance_(object-oriented_programming)) borrowed from object-oriented programming.
* A [caching](https://en.wikipedia.org/wiki/Web_cache) framework that can use any of several cache methods
* Support for [middleware](https://en.wikipedia.org/wiki/Middleware) classes that can intervene at various stages of request processing and carry out custom functions
* An internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signal.
* An [internationalization](https://en.wikipedia.org/wiki/Internationalization_and_localization) system, including translations of Django's own components into a variety of languages
* A [serialization](https://en.wikipedia.org/wiki/Serialization) system that can produce and read [XML](https://en.wikipedia.org/wiki/XML) and/or [JSON](https://en.wikipedia.org/wiki/JSON) representations of Django model instances
* A system for extending the capabilities of the template engine
* An interface to Python's built-in [unit test](https://en.wikipedia.org/wiki/Unit_test) framework
* Django REST framework is a powerful and flexible tool it for building Web APIs

**CLIENT-SIDE SCRIPTING: HTML, JAVASCRIPT, JQUERY**

**JQUERY**

jQuery is not a language, but it is a well written JavaScript code. As quoted on official jQuery website, "it is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development", jQuery is not a language, but it is a well written JavaScript code. As quoted on official jQuery website, "it is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development." In order to work with jQuery, you should be aware of the basics of JavaScript, HTML and CSS. It was released in January 2006 at Bar Camp NYC by John Resign. jQuery is very compact and well written JavaScript code that increases the productivity of the developer by enabling them to achieve critical UI functionality by writing very small amount of code.

* It helps to improve the performance of the application
* It helps to develop most browser compatible web page
* It helps to implement UI related critical functionality without writing hundreds of lines of codes
* It is fast
* It is extensible – jQuery can be extended to implement customized behavior

**Other advantages of jQuery are:**

* No need to learn fresh new syntaxes to use jQuery, knowing simple JavaScript syntax is enough
* Simple and cleaner code, no need to write several lines of codes to achieve complex functionality

**JAVASCRIPT**

JavaScript is a programming language that allows you to implement complex things on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, or interactive maps, or animated 2D/3D graphics, or scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which ([HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML) and [CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS)) we have covered in much more detail in other parts of the Learning Area.

* [HTML](https://developer.mozilla.org/en-US/docs/Glossary/HTML) is the markup language that we use to structure and give meaning to our web content, for example defining paragraphs, headings, and data tables, or embedding images and videos in the page.
* [CSS](https://developer.mozilla.org/en-US/docs/Glossary/CSS) is a language of style rules that we use to apply styling to our HTML content, for example setting background colors and fonts, and laying out our content in multiple columns.
* [JavaScript](https://developer.mozilla.org/en-US/docs/Glossary/JavaScript) is a programming language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else. (Okay, not everything, but it is amazing what you can achieve with a few lines of JavaScript code.)

**HTML**

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the [» W3C](http://www.w3.org/), the organization charged with designing and maintaining the language.

The definition of HTML is Hypertext Markup Language.

* Hypertext is the method by which you move around on the web — by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.
* Markup is what HTML tags do to the text inside them. They mark it as a certain type of text (italicized text, for example).
* HTML is a Language, as it has code-words and syntax like any other language.

HTML consists of a series of short codes typed into a text-file by the site author — these are the tags. The text is then saved as a html file, and viewed through a [browser](http://www.yourhtmlsource.com/starthere/glossary.html#browser), like Internet Explorer or Netscape Navigator. This browser reads the file and translates the text into a visible form, hopefully rendering the page as the author had intended. Writing your own HTML entails using tags correctly to create your vision. You can use anything from a rudimentary text-editor to a powerful graphical editor to create HTML pages.

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, macOS, Microsofte Windows NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, OracleSolaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

The MySQL server software itself and the client libraries use [dual-licensing](https://en.wikipedia.org/wiki/Dual_license) distribution. They are offered under GPL version 2, beginning from 28 June 2000 (which in 2009 has been extended with a [FLOSS](https://en.wikipedia.org/wiki/Alternative_terms_for_free_software) License Exception) or to use a proprietary license.

Support can be obtained from the official manual. Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organizations exist to provide support and services, including [MariaDB](https://en.wikipedia.org/wiki/MariaDB" \o "MariaDB) and [Percona](https://en.wikipedia.org/wiki/Percona" \o "Percona).

MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case" and that the "developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good". It has also been tested to be a "fast, stable and true multi-user, multi-threaded SQL database server".

**FEATURES**

* Speed: Of course, the speed at which a server-side program runs depends primarily on the server hardware .Given that the server hardware is optimal, MySQL runs very fast. It supports clustered servers for demanding applications.
* Ease of use: MySQL is a high-performance, relatively simple database system. From the beginning, MySQL has been configured, monitored, and managed from the command line. However, several MySQL graphical interfaces are available as described below:
* MySQL Administrator: This tool makes it possible for administrators to set up, evaluate and tune their MySQL database server. This is intended as a replacement for MySQL admin.
* MySQL Query Browser: Provides database developers and operators with a graphical database operation interface. It is especially useful for seeing multiple query plans and result sets in a single user interface.
* Configuration Wizard: Administrators can choose from a predefined list of optional settings, or create their own.
* MySQL System Tray: Provides Windows based administrator a single view of their MySQL, instance, including the ability to start and stop their database server.
* Cost: MySQL is available free of cost. MySQL is a “open-source database”. MySQL is a part of LAMP (Linux, Apache, MySQL, PHP/Perl/Python) environment, a fast-growing open-source enterprises software stacks because of its lower cost, reliability, and documentation.

**CHAPTER 4**

**SYSTEM DESIGN & DEVELOPMENT**

# 4.1 INFRASTRUCTURE DESIGN

System design provides an understanding of the procedure details, necessary for implementing the system recommended in the feasibility study. Basically, it is all about the creation of a new system. This is critical phase since it decides the quality of the system and has a major impact on the testing and implementation phases. Design is the second phase in the system development life cycle Software design is the first of the three technical activities in the software development process such as design, code writing and testing.

During this phase, the analyst schedules design activities, works with the user to determine the various data inputs to the system, plans how data willow though the system, designs required outputs and writes program specifications. Again, the analyst's activities focus on solving a user's problem in logical terms.

During this second step, analysts employ s variety of tools such as data flow diagrams, entity- relationship diagrams, data dictionaries and Gantt chart. The system's design converts the theoretical solution introduced by the feasibility study into a logical reality.

During design the analyst:

* Draws a model of the new system, using data flow and entity-relationship diagrams
* Develop methods for collecting and inputting data
* Defines the detailed data requirements with a data dictionary
* Writes program specifications
* Specifies control techniques for the system's outputs, databases and inputs.
* Identifies and orders any hardware or software that the system will need.

In the physical design phase, necessary software is developed to accept input from the user, to perform necessary calculations through the manipulation of data stored in the databases to produce the appropriate result.

## 4.2 INPUT DESIGN

Input Design is the process of converting a user-oriented description of the inputs to a computer-based business system into a programmer-oriented specification. The aim of making input design is to make the data entry as easy, logical and free from errors as possible. An input format should be logical and easy to understand. In the design, the user-oriented inputs are converted into computer recognizable format.

The collection of data is most expensive part of the system in terms of the equipment used, time and no. of clients involved etc. in the input design data is accepted and it can be readily used for data processing or can be stored in a database for further use. The activities used for inputs are very user friendly. Different names are associated with each data entry activity item makes data entry an easy job. Each data entry contains separate buttons for submitting the form and proper validation checking is carried out and necessary message will be presented to the user in case of improper data entry. The proposed system satisfies the following input design objectives

* A cost-effective method of input
* The highest possible level of accuracy
* The input is acceptable to and understood by the user.

Input objectives are

1. **Controlling the amount of input:** Wherever user input is required, the number of key strokes is reduced by giving possible input values as default in that area. The viewer can select the answer in single click. The amount of information entered by the viewer using the virtual keypad is reduced to the maximum and the software is made very user friendly.
2. **Avoiding Delay:** A processing delay resulting from data preparation or data entry operations is called a bottleneck. Such bottlenecks are avoided to the maximum. The only time the viewer has to wait is when the file is uploaded or downloaded. Progress bar or progress meters are displayed to keep the user waiting and also to show the speed and amount of download.
3. **Avoiding Errors in Data:** The rate at which errors occur depends on the quantity of data. Here the quantity of data is reduced to the lowest, and a text file is easily manageable.
4. **Avoiding Extra steps:** The viewer can quit at any point of time. Even at the time of upload or download the viewer can quit. The viewer need not wait for any specific event to happen for quitting the process.
5. **Keeping the process Simple:** This implies that the system has all the measure to keep the errors out even if the user is giving wrong data. It handles the situation with grace and doesn’t create much hype about the situation to the user.

## 4.3 OUTPUT DESIGN

It has been an ongoing activity from the beginning of the project. It includes the process of finalizing of the medium format and exact contents of each output to be produced by the proposed system. The primary objective in creating an output is accuracy and neatness.

Efficient output design should improve the system's relationship with the user and help in decision making. The system has got the capability to display standard screen layouts. These layouts should be designed around the output requirements and they must be designed with utmost care and details in the screen layouts must be simple, descriptive and clear to the user. While designing a system's output, the analyst must make several independent decisions. Every system produces some kind of reports. No matter what is the content of the report. The output design of this project is made with these objectives in mind.

#### Output Types

* External Outputs, whose destination is outside the organization and is the main image of the organization,
* Internal Outputs, whose destination is within the organization and which require careful design because it is user’s main interface with the Android mobile.
* Interactive Outputs, which involve the user in communicating directly with the Android mobile.

### 4.3.1 FORM DESIGN

**LOGIN FORM**

|  |  |
| --- | --- |
| **Username** |  |
| **Password** |  |
| LOGIN | |

**REGISTER FORM**

|  |  |
| --- | --- |
| **First Name** |  |
| **Last Name** |  |
| **Username** |  |
| **Email** |  |
| **Phone** |  |
| **Password** |  |
| REGISTER | |

**PROFILE UPDATE FORM**

|  |  |
| --- | --- |
| **Username** |  |
| **Email** |  |
| **Phone** |  |
| UPDATE | |

**FEEDBACK FORM**

|  |  |
| --- | --- |
| **Message** |  |
| SUBMIT | |

**RECIPE UPLOAD FORM**

|  |  |
| --- | --- |
| **Recipe Name** |  |
| **Total Time** |  |
| **Dish Type** |  |
| **Description** |  |
| **Photo/Video** |  |
| **Instructions** |  |
| UPLOAD | |

## 4.4 DATA FLOW DIAGRAM

A DFD, also known as a "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flow in the system.

A data flow diagram may be used to represent a system or software at any level of abstraction DFD's can be partitioned into levels that represent increasing information flow and functional details. A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change or transform data throughout a system. It is a structured analysis and design tool that can be used for flowcharting in place of or in association with, information oriented and process-oriented system flowcharts.

When analysts prepare the DFD, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply a physical implementation. The DFD reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

Basic data flow diagram symbols are:

* “Rectangle” defines a source or destination.
* An “Arrow” identifies data flow. It is a pipeline

through which information flows.

* A “Circle” represents a process that transforms

incoming data flow(s) into outgoing data flow(s).

* An “Open Rectangle” is a data store

Steps to Construct Data Flow Diagrams:Three steps are commonly used to construct a DFD.

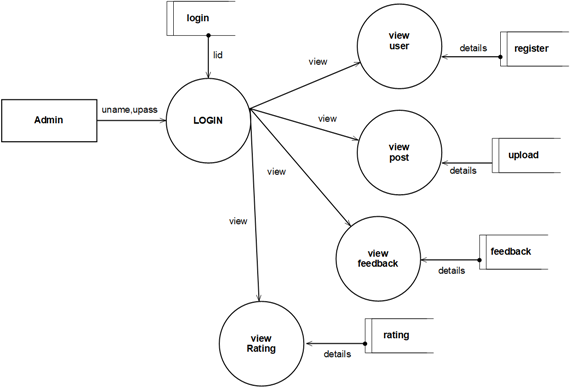
* Process should be named and numbered for easy reference each name should be representative of the process.
* The direction of flow is from top to bottom and from left to right.
* When a process is exploded in to lower-level details they are numbered.

### LEVEL 0 DFD (CONTEXT DIAGRAM)



**Fig: 4.4.1 Context Diagram**

### Level 1 Admin



**Fig: 4.4.2 Level 1 Admin**

### Level 1 User



**Fig: 4.4.2 Level 1 User**

## 4.5 ANALYSIS TOOLS

System analysis is the process of collecting and interpreting facts, understanding problems and using this information to suggest improvements in the system. This will help to understand the existing system and determine how computers make their operations more effective.

In software engineering, a use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-Case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The use case diagram shows the position or context of the use case among other use cases. As an organizing mechanism, a set of consistent, coherent use cases promotes a useful picture of system behavior, a common understanding between the customer/ owner/ user and the development team.

Use Case is a technique for capturing functional requirements of systems and systems-of-systems. Each use case provides one or more scenarios that convey how the system should interact with the users called actors to achieve a specific business goal or function. Use case actors may be end users or other systems. Use cases typically avoid technical jargon, preferring instead the language of the end user.

The UML standard describes graphical notation for relationships. Use cases are represented by ovals and the actors are represented by stick figures. The use case diagram of this application is given below.

## 4.6 Database Design

After designing the input and output activities, the designer moves to concentrate on database design. How data are organized depends on the data and response requirements that determine hardware configurations. The database is organized to ensure that the system resources are not wasted. The objective of the database design is to provide the effective auxiliary storage and contribute to the overall efficiency to the computer program components of the proposed system. The database design translates the data models that were developed for the system users during the definition phase into the data structures supported by the chosen database technology. The goals of database design are as follows:

1. A database should provide for the efficient storage, update and retrieval of data.
2. A database should be reliable; the stored data should have high priority to promote user trust in that data.
3. A database should be adaptable and scalable to new and unforeseen requirements and applications.

The techniques used to improve a data model in preparation for database design is called data analysis. Data analysis is a process that prepares a data model for implementation as a simple, non-redundant and adaptable database. The specific technique is called Normalization. Normalization is a technique that organizes the data attributes such that they are grouped to form stable, flexible and adaptive entities. The table involved in inspection process along with attributes, data types, constraints and brief description about the fields are stated in the below mentioned table.

### 4.5.1 Normalization

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this application for the following reasons.

* To structure the data so that there is no operation of data, this helps in saving space.
* To permit simple retrieval of data in response to query and report request.
* To simplify the maintenance of the data through updates, insertions and deletions.
* To reduce the need to restructures or recognize data which new application requirements arise.

Primary key is assigned for this purpose. The primary key fields in the tables help to ease the search and improve efficiency. The proposed system is using second normal form as it is found most suitable. In second normal form each row must contain associated field that describes an attribute of the entry that the table describes.

### Table 4.6.1 Login

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Uid | INT | PRIMARY KEY | User id |
| Uname | VARCHAR(50) | NOT NULL | Username |
| Upass | VARCHAR(50) | NOT NULL | User password |
| Utype | VARCHAR(50) | NOT NULL | User type |

### Table 4.6.2 Registration

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Uid | INT | PRIMARY KEY | User Id |
| Fname | VARCHAR(50) | NOT NULL | First Name |
| Lname | VARCHAR(50) | NOT NULL | Last Name |
| Username | VARCHAR(50) | NOT NULL | Username |
| Email | VARCHAR(50) | NOT NULL | E-mail |
| Phone | VARCHAR(50) | NOT NULL | Phone |

### Table 4.6.3 Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Fid | INT | PRIMARY KEY | branch id |
| Uid | VARCHAR(50) | NOT NULL | Branch name |
| Message | VARCHAR(1000) | NOT NULL | address |

### Table 4.6.4 Rcap

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Rid | INT | PRIMARY KEY | Recipe id |
| Uid | INT | FOREIGN KEY | User id |
| recipe\_name | VARCHAR(50) | NOT NULL | Recipe name |
| total\_time | VARCHAR(50) | NOT NULL | time |
| dishtype | VARCHAR(50) | NOT NULL | Dish type |
| description | VARCHAR(50) | NOT NULL | description |
| photo\_vedio | VARCHAR(50) | NOT NULL | Images |
| instruction | VARCHAR(50) | NOT NULL | Instruction |

### Table 4.6.5 Rating

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Ratid | INT | PRIMARY KEY | rating id |
| Rid | INT | FOREIGN KEY | recipe id |
| Uid | INT | FOREIGN KEY | user id |

### Table 4.6.6 Follow

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| fid | INT | PRIMARY KEY | follow id |
| uid | INT | FOREIGN KEY | user id |
| Userid | INT | FOREIGN KEY | User id |

### Table 4.6.7 Category

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| cid | INT | PRIMARY KEY | Category ID |
| Dishtype | VARCHAR(50) | NOT NULL | Dish type |

### 4.7 E-R DIAGRAM

An entity-relationship (ER) diagram is specialized graphic that illustrates the interrelationship between entities in a database Boxes are commonly used to represent entity Diamonds are normally used to represent relationships and ovals are used to represent attributes.

An entity is piece of data is shared between entities.

### Classifying Relationships

Relationships are classifieds by their degree, connectivity, cardinality, direction, type and existence.

### Degree of Relationships

The degree of a relationship is the number of entities associated with the relationship. The n- array relationship is there general from for degree n. Special cases are binary, ternary where the degree is 2 and 3 respectively.

### Connectivity and Cardinality

The connectivity of a relationship describes the mapping of associated entity instances in the relationship. The values of connectivity are "one" or "many" The cardinality of a relationship is the actual number of related occurrences for each of the two entities. The basic types of connectivity of relations are: One-to-one, one-to- many, many-to-many.

**A one-to-one (1:1) is** when at most one instance of an entity A is associated with one instance of entity B.

**0-to-many (1: N) is** when for an instance of an entity A, there are zero, one or many instances of entity B, but for instance of the entity B, there is only one instance of the entity A.

**A many-to-many (M: N) relationship**, sometimes called non-specific, is when for one instance of entity A, there are zero, one or more instances of entity B and for one instance of entity B there are zero, one or many instances of entity A.

The symbols used ER Diagram is

Entity

Attributes

Relationships

Lines



**FIG: 4.7 ER Diagram**

**CHAPTER 5**

**SYSTEM TESTING**

## 5.1 INTRODUCTION

System testing is actually a series of different testes whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work together to verify that the system elements have been properly integrated and perform all functions clearly.

System testing makes logical assumptions that if all parts of the system are correct, the goal will be successfully achieved. Testing is the process of executing the program with the intent of finding errors. Testing cannot show the absence of defects, it can only show that software errors are present.

Testing on this project can be done in many ways, such as module testing, where every single program module is examined thoroughly, this project is also divided into many modules such service, user interface, viewing schedules and deleting schedules etc. Also, the whole unit will be tested as every data entered and searched will also be tested; all the runtime errors can be detected and corrected accordingly. There are also other types of testing such as integration testing; validation testing etc. the whole project is integrated so it has to be tested at each and every point.

Testing is a process of checking whether the developed system is working according to the original objectives and requirements. A test case is one that has a possibility of finding as yet undiscovered error. A successful test is one that uncovers as yet undiscovered error. The developed system is tested whether it works efficiently and whether it satisfies all the user requirements by taking a series of test cases.

### 5.1.1. Types of Testing

The software, which has been developed, has to be tested to prove its validity. Testing is considered to be least creative phase of the whole cycle of the system design. In the real sense it is the phase, which helps to bring out the creativity of the phases. No system design is ever perfect. Errors occur due to communication problems, programmer’s negligence or time constraints. All these must be eliminated before the system is ready for user acceptance testing.

Different levels of testing are employed during different stages of the system development.

#### 5.1.1.1 White Box Testing

By using this technique, it was tested that all the individual logical paths were executed at least once as logical decisions were tested on both their true and false sides. All the loops were tested with data in between the ranges and especially at the boundary values.

#### 5.1.1.2 Black Box Testing

By the use of this technique, the missing functions were identified and placed in their positions. The errors in the interfaces were identified and corrected. This technique was used to identify the initialization and termination errors and correct them.

#### 5.1.1.3 Unit Testing

In the lines of this strategy, all the individual functions and modules were put to the 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 techniques to find the errors in each module.

#### 5.1.1.4 Integration Testing

This testing strategy has two different approaches namely the top-down approach, in which the integrations are carried out from the top-level module to the bottom and bottom-up approach in which the integration is carried out from the low-level module to the top. The modules were tested using the bottom-up approaches by introducing stubs for the top-level functions. This test was used to identify the errors in the interfaces, the errors in passing the parameters between the functions and to correct them.

#### 5.1.1.5 Validation Testing

Validation testing can be defined in many ways, but a simple definition that validation succeeds when the software in a manner that is reasonably expected by the customer. Software validation is achieved through a series of black box test that demonstrate conformability with requirements. After validation test have conducted, one of the two possible conditions exists.

* The function or performance characterized confirm the specification and are accepted.
* Deviation from specification is found and a deficiency list is created.

#### 5.1.1.6 Output Testing

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in a specific format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration.

The output format of a screen is found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, output comes out as the specified requirements by the user. Hence output testing does not result in any correction in the system.

## 5.2 TEST CASES

A specific set of steps and data along with expected results for a particular test objective. A test case should only test one limited subset of a feature of functionality. Test case documents for each functionality/ testing areas will be written, reviewed and maintained separately in Excel sheets.

In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same. Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error message s and logs. Realistically, if error test cases are not yet written, it is OK for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any, is expected to trigger errors.

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

## 6.1 INTRODUCTION

Implementation is the stage of project when the theoretical design is turned into a working system. At this stage, the main workload, the greatest upheaval and the major impact on existing practices shift to the user department. If the implementation stage is not carefully planned and controlled, it can cause chaos. The implementation stage is a system project in its own right. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, training of staff in the changeover procedure and evaluation of changeover methods.

The implementation is the final and important phase. It involves user training system testing in order to ensure successful running of the proposed system. Once the system design phase is over, the next stage is to implement and monitors the operation of the system to ensure that it continues the work effectively and efficiently.

The three main phases in implementation take place in series. These are the initial installation, the test of the system as a whole and evaluation maintenance and control of the system. The implementation plan and action to implement should be bound closely together. The implementation plan is a function of line management at least as far as key decision or alternative plans are concerned. The implementation plan was to convert the existing clerical files to the computer. The implementation plan listed all sub tasks so that individuals in the organization may be assigned specific responsibilities.

The installation of the new system that is bound to replace the current one may require a major revision of computer facilities as well as completely new after space. Space planning took into account the space occupied by the people, space by equipment and the movement of people and equipment in the working investment. After conduction the initial testing the system is loaded on the client office's computer. Some of the user employees in this case are selected. These users are trained first and they run the system. A detailed documentation is prepared to this set of employees. There may be slight modifications to meet the organization.

After all modifications specified by the users in the documentation are made, the computer system is run along with manual system. Even though this kind of parallel run make extra burden to the employees and management, the system is run in parallel for the sake of checking reliability and efficiency.

After this document which compares the result of the manual system with those of the computerized is prepared. In the case of management many of whom participated in the development of the system short seminars were given. Particular attention was paid to the training of end users. The training sessions were aimed at giving the user staff the specific skills required in their new jobs. Education involved creating the right atmosphere and motivation of user staff. It explained the need for changes and helped to overcome the resentment caused by the feeling that computers took away the responsibility from individual departments.

Various measures have been taken by department officials in order to find suitable solutions by the following issues

* About the skill to be acquired.
* Reduction of man power in department.
* About the new form having all required option.

## 6.2 IMPLEMENTATION PROCEDURE

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage, the main work load, the greatest upheaval and the major impact on the existing system shifts to the department. If the implementation is not carefully planned and controlled, it can cause confusion. Implementation includes all those activities that take place to convert from the old system to the new one. Proper implementation is essential to provide a reliable system to meet the organizational requirements.

Successful implementation may guarantee improvement in the organization using the new system, but improper installation will prevent it. The process of putting the developed system into the actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specification of the system.

The most crucial stage is achieving a new successful system and giving confidence on the new system for the user that it will work efficiently. It involves careful planning, investigation of the current system and is constraints on implementation, design of methods to achieve the changeover. The more the complex system being implemented is, the more involved will be the system analysis and the design effort required for its implementation.

### 6.2.1 EQUIPMENT ACQUISITION

Here, the necessary equipment is acquired to implement the system. Major steps involved in the implementation are installation of software. In case of this system there is no special hardware requirement for the working of the software. In addition to basic hardware requirements one thing that is essential is the software requirements with an operating system that has server support. The proposed system can run on any PC, which works on android and any versions of it. Our software is platform dependent and will run only on the windows platform after installing the software, it is essential to ensure that the software is working accordingly with the existing software.

## 6.2.2 TRAINING

For this system, it was explained to the users how to use the system, what details are to be given while creating a new profile, how to use it so that we get the maximum output out of it. The proposed system may be entirely new, replacing an existing one or it may be modifications to the existing system. In either case, proper implementation is necessary to provide a reliable system to meet organizational requirements.

**6.2.3 EVALUATION**

Evaluation of the system is performed to identify its strength and weaknesses.

### 6.2.3.1OPERATIONAL EVALUATION

Assessment of the manner in which the system functions, including ease of use, response time, overall reliability and level of utilization.

#### 6.2.4 ORGANIZATIONAL IMPACT

Identification and measurement of benefits to the organization in such areas like financial concerns, operational efficiency and competitive impact.

#### 6.2.5 USER MANAGEMENT ASSESSMENT

Evaluation of attitude of senior and user managers within the organization, as well as end users.

#### 6.2.6 DEVELOPMENT PERFORMANCE

Evaluation of the development process based on overall development time and effort, conformance to budgets and standards, and other project management criteria includes assessment of development methods and tools.

#### 6.2.7 DOCUMENTATION

After the testing and implementation was completed, the whole system was presented and documented in a readable manner. This was done to ensure that any corrections, manipulations or updating are performed in future, the users would face no problem in performing those changes. Documentations include the source code, the tables that were used to construct the base for the system, the framework which is bound to the programs etc.

#### CHAPTER 7

**SOFTWARE MAINTENANCE**

Software maintenance denoted any changes made to a software product after it has been delivered to the customer. Maintenance is inevitable for almost any kind of product. Most products need maintenance due to wear and tear by use. Although software does not wear out like a piece of hardware its “ages” and eventually fails to perform. So, maintenance becomes necessity. Types of software maintenance:

### 7.1 CORRECTIVE MAINTENANCE

Corrective maintenance of a software product is necessary either to rectify the bugs observed while the system is in use.

### 7.2 ADAPTIVE MAINTENANCE

A software product might need maintenance when the customers need the product to run on new platforms, on new operating systems, or when they need the product to interface with new hardware or software.

### 7.3 PERFECTIVE MAINTENANCE

A software product needs maintenance to support the new features that users want it to support, to change different functionalities of the system according to customer demands, or to enhance the performance of the system.

### 7.4 PREVENTIVE MAINTENANCE

Modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults.

**CHAPTER 8**

**CONCLUSION**

In conclusion, Reciperave stands as a beacon for culinary enthusiasts, offering a dynamic platform where creativity and community converge. By seamlessly integrating diverse recipe collections, engaging cooking videos, and robust user interaction features like ratings, comments, and social sharing, Reciperave not only facilitates the exploration of global cuisines but also cultivates a supportive environment for sharing personal culinary journeys. With its user-friendly interface and comprehensive modules designed for both novices and seasoned chefs alike, Reciperave empowers individuals to showcase their expertise, discover new flavors, and forge meaningful connections within a vibrant community dedicated to the art of cooking and baking. Whether you're seeking inspiration or eager to share your own culinary innovations, Reciperave invites you to join a passionate community where every recipe tells a story and every dish is an opportunity for creativity and connection.

**CHAPTER 9**

**FUTURE ENHANCEMENT**

Looking ahead, RecipeRave envisions several exciting enhancements to further enrich the user experience and expand its culinary community. One key focus area involves leveraging artificial intelligence and machine learning algorithms to personalize recipe recommendations based on user preferences and cooking habits. This enhancement will not only streamline the recipe discovery process but also ensure that users continually encounter recipes tailored to their tastes and dietary preferences.

Additionally, RecipeRave plans to introduce interactive cooking classes led by renowned chefs and expert cooks through live streaming or pre-recorded sessions. These cooking classes will provide users with hands-on learning opportunities, enabling them to refine their culinary skills while directly interacting with instructors and fellow participants in real-time.

Furthermore, in response to user feedback, RecipeRave intends to enhance its social features by introducing thematic cooking challenges and contests. These initiatives will encourage users to showcase their creativity and culinary prowess by submitting entries based on specific themes or ingredients, fostering a spirit of friendly competition and community engagement.

Lastly, RecipeRave aims to expand its international presence by collaborating with chefs and culinary experts from diverse cultural backgrounds to curate specialized recipe collections that celebrate global cuisines. This initiative will not only broaden the platform's recipe offerings but also promote cultural exchange and appreciation among users worldwide.

By embracing these future enhancements, RecipeRave is committed to solidifying its position as the premier online destination for culinary enthusiasts, empowering them to discover, learn, and share their passion for cooking and baking in an innovative and inclusive community-driven environment.

Top of Form

Bottom of Form

**CHAPTER 10**

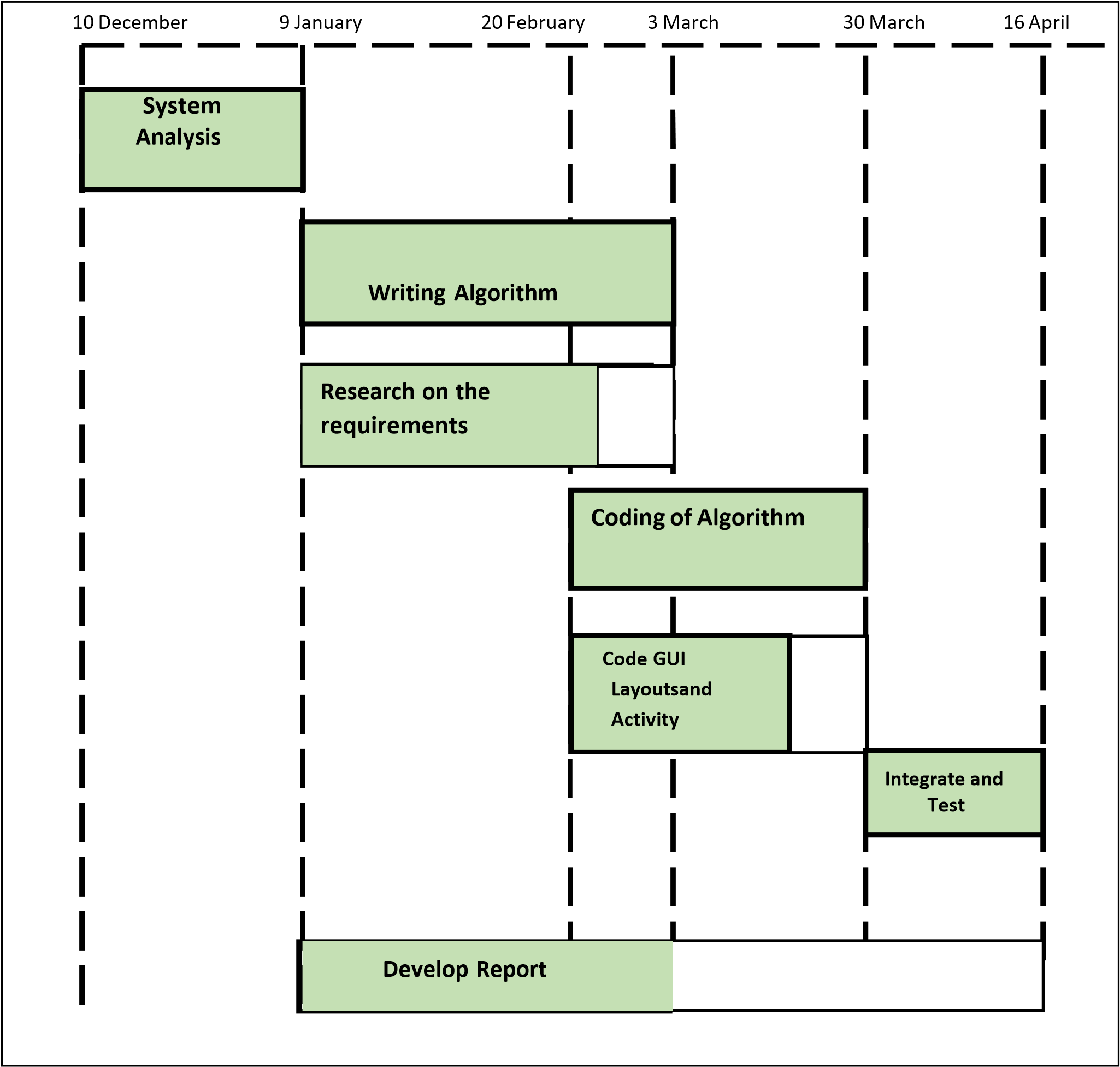
**APPENDIX**

**Fig: 10.1.7 Contact Form**

### 10.2 SAMPLE SOURCE

### 10.3 GANTT CHART

The Gantt chart was developed by Henry Gantt. Those are used in software project management and enhanced version of standard Gantt chart. These are mainly allocating resources to activities. It is a special type of bar chart. Each bar represents an activity. Bars are drawn alone a time line. Length of each bar is proportional to duration of time planned for corresponding activity.

march 1 april14may25june18july 30 august 29

**CHAPTER 11**

**REFERENCE**

**Text Books:**

#### Fundamentals of Software Engineering, Fifth Edition, Rajib Mall- PHI 2018

* **Software Engineering: A Practitioner's Approach (IRWIN COMPUTER**

**SCIENCE) Hardcover – 16 March 2014**

**Websites:**

* https://www.tutorialspoint.com/uml/uml\_class\_diagram.htm
* <https://www.w3schools.com/php/>
* [https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/](https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/)
* [https://myiee.org/documentation-requirementshttps://myiee.org/documentation-requirements](https://myiee.org/documentation-requirements)
* <https://www.w3schools.com/mysql/default.asp>