**A black background with white text

Description automatically generated with low confidence**

**INTRODUCTION**

**Foodify** is a cutting-edge web and mobile application that helps you plan your weekly meals while minimizing food waste. By leveraging state-of-the-art technologies such as Machine Learning and Deep Learning, Foodify recommends over 100 dishes from different nations for you to enjoy. But that's not all - with Foodify's delivery option, you can have your weekly meals planned and delivered straight to your door, without having to worry about a thing!

Whether you're a busy professional looking to streamline your meal planning process, or simply want to try out new and exciting dishes from around the world, Foodify has you covered. With a user-friendly interface and powerful back-end algorithms, Foodify takes the guesswork out of meal planning and makes it easy and fun to discover new recipes and cuisines. Give Foodify a try today and experience the future of meal planning!

**Features**

**Meal planning**: With Foodify, you can easily plan your meals for the week ahead, taking into account your dietary preferences, ingredient availability, and more. You can create personalized meal plans that fit your schedule and nutritional goals.

**Dish recommendations**: Foodify leverages advanced machine learning and deep learning algorithms to recommend over 100 dishes from different nations, so you can discover new and exciting recipes to try out. The recommendations are tailored to your preferences and can help you explore diverse cuisines.

**Delivery option**: Want to take the hassle out of meal planning? With Foodify's delivery option, you can have your weekly meals planned and delivered straight to your door. Simply select the meals you want, and our delivery partners will handle the rest, ensuring you have everything you need for a hassle-free cooking experience.

**Minimize food waste**: Foodify helps you minimize food waste by recommending recipes that make use of ingredients you already have on hand. The app suggests ingredient substitutions when necessary and provides options for utilizing leftovers effectively, reducing waste and saving money.

**User-friendly interface**: Foodify offers a sleek and intuitive interface, making it easy and enjoyable to plan your meals and discover new dishes. The user-friendly design ensures a seamless experience, whether you're using the web or mobile application.

**Personalized recommendations**: Foodify's advanced algorithms take into account your past meal selections, preferences, and dietary restrictions to provide personalized dish recommendations. This ensures that you receive tailored suggestions that align with your unique tastes and dietary needs.

**Grocery list generation**: Generate a comprehensive grocery list based on the selected meal plan, making it easy to shop for the required ingredients. This feature saves you time and ensures you have everything you need for each recipe.

**Integration with grocery delivery services**: Seamlessly integrate Foodify with local grocery delivery services, allowing you to order the required ingredients directly from the app. This streamlines the meal planning process, eliminating the need for separate shopping trips.

**Nutritional information and dietary tracking**: Access detailed nutritional information for each recipe, empowering you to make informed decisions about your meals. Foodify also offers a feature for tracking your daily calorie intake or other dietary goals, helping you maintain a healthy and balanced diet.

**Social sharing and reviews**: Share your favorite recipes or meal plans with friends and family through social media platforms directly from the app. Additionally, you can rate and leave reviews for recipes you have tried, helping others in the Foodify community make informed decisions.

**Analytics and data insights**: Foodify gathers data on user behavior, preferences, and usage patterns to provide valuable insights. This data-driven approach allows us to continually improve the app and enhance the user experience based on real-time feedback.

**TECHNOLOGIES**

**HTML**: For the website structure and layout.

**CSS**: For styling the website.

**Bootstrap**: For responsive design and additional styling.

**Python**: For Data Preprocessing and Machine Learning / Deep Learning.

**Numpy**: For Numerical computing.

**Pandas**: For Data Manipulation and Analysis.

**Matplotlib**: For Data Visualization.

**Scikit-Learn**: For Machine Learning algorithms.

**Pytorch**: For Deep Learning models.

**Flask**: for the web application framework.

**PostgreSQL**: For Database.

Abubakr Mamajonov 52493

**PROJECT FOLDER DESCRIPTION**

**For Developers**

Link for Project: <https://github.com/Abubakr1710/Foodify>

**A screenshot of a computer

Description automatically generated with medium confidence**

**Main Structure:**

**Info:** Web application information. It contains all information about how to use web application and section descriptions.

**data**: Dataset for tarining and testing machine learning/deep learning models

**models**: Machine Learning and Deep learning models saved in this folder

**version1**: First version of the application

**web**: Files for web application saved in this folder

**.gitignore**: File is used to specify files and directories that should be ignored by Git.

**README.md**: It serves as a concise and informative guide that provides an overview of the project, instructions for installation and usage, and other relevant information for developers and users.

**Requirements.txt**:File is used to specify the external dependencies and libraries required by a Python project. It provides a convenient way to manage and install the necessary packages for the project.

**Abubakr Mamajonov 52493**

**UI/UX**

**HOME**

***Introduction***

On websites and apps, the home button is a frequently utilized feature. It often takes the form of a clickable icon or piece of text that, when clicked or tapped, returns the user to the website's or application's home page.

The home button's function is to give users an easy and simple way to return to a website's home page or main page. It functions as a visual anchor that gives users a sense of orientation and familiarity and enables them to rapidly reset their location within the site hierarchy.

No of how far they have looked into a website's pages or sections, viewers can quickly return to the major content or main menu by clicking the home button. Users are spared from having to repeatedly click the browser's return button or navigating through difficult menus in order to get back.

The home button ensures that viewers can always return to the starting point, which is especially helpful for websites with considerable material or complex structures. By offering a dependable and convenient way to go back to the home page, it improves usability, streamlines navigation, and overall user experience.

***What does homepage includes?***

While the specific elements may vary depending on the website's purpose and design, here are some common components found on a home page:

**Logo:** The logo represents the brand or organization and is usually positioned prominently at the top of the home page.

**Navigation Menu:** A navigation menu helps users explore different sections or pages of the website. It may include dropdown menus or subcategories to provide easy access to various content areas.

**Hero Section:** The hero section often appears at the top of the home page and includes attention-grabbing elements like a large image, headline, and a call-to-action button. It aims to engage visitors and provide a quick overview of the website's main features or offerings.

**Featured Content**: This section highlights important or popular content, such as featured articles, products, services, or promotions. It helps direct users to relevant information or encourage them to take specific actions.

**About Us**: Many home pages include an "About Us" section that provides a brief introduction to the organization, including its mission, values, and key information. This section helps establish credibility and build trust with visitors.

**Testimonials or Reviews**: Displaying positive testimonials or reviews from satisfied customers or clients can help build trust and credibility for the website or business.

**Latest News or Blog Posts:** If the website includes a blog or regularly publishes news or articles, the home page may feature a section showcasing the most recent or popular posts. This encourages visitors to explore further and engage with the content.

**Contact Information**: Including contact details, such as a phone number, email address, or physical address, allows visitors to easily get in touch with the website owner or business.

**Social Media Links**: Providing links to social media profiles enables visitors to connect with the website or brand on various social platforms.

**Footer**: The footer section is located at the bottom of the home page and often contains additional navigation links, copyright information, privacy policy, terms of service, and other relevant links.

***Side Bar***

On a website or application interface, a sidebar is a vertical column or segment. It often includes extra or auxiliary data, features, or navigational options that support the page's primary functionality or content.

A sidebar's function is to offer rapid access to essential and practical data without distracting from the page's main subject. It functions as a convenient location for extra content or features that improve the user experience.

In a website or application, a sidebar often includes the following elements:

**Navigation:** Users can access various pages or sections of a website or application by using the menus or links that are frequently featured in sidebars. This offers a reliable and convenient way to navigate the information or access particular features.

**Modules or Widgets:** Sidebars commonly include different modules or widgets that provide extra functionality or content. Search bars, recently posted content, social media feeds, calendars, weather updates, or advertising banners are a few examples. These elements include interactive and dynamic elements that improve user experience and enable quick access to pertinent information.

**Related Content or Suggestions**: Depending on the user's current context or interests, sidebars may present related content or suggestions. This may consist of relevant articles, suggested items, comparable profiles, or tailored recommendations. User Profile and Account Information: The sidebar may have a section for showing the user's profile details, notifications, messages, or settings if the website or program demands user accounts. Users can view and manage their account-related actions in this easily accessible spot.

**Call-to-Action Buttons**: Sidebars may include links or buttons that direct visitors to take particular actions, including registering for a service, signing up for a newsletter, or making a purchase. These buttons are designed to promote user interaction and conversions.

**Contextual Information:** Sidebars may also show contextual data related to the page or task that is now being worked on. This could include of progress indicators, relevant information related to the user's actions, or contextual recommendations or hints.

Overall, the sidebar serves as a versatile and flexible component that complements the main content by offering additional functionality, navigation, or supplementary information. It enhances usability, provides quick access to relevant features, and allows for a more efficient and engaging user experience.

***What is the relation between home page and side bar?***

Here are some ways in which the home page and sidebar can interact:

**Navigation:** The sidebar often includes navigation menus or links that allow users to navigate to different sections or pages within the website. The home page can provide an overview of the main content or categories, while the sidebar offers more detailed sub-navigation options, enabling users to explore specific areas of interest.

**Content Highlights**: The home page may feature highlighted or recommended content, and the sidebar can provide related or similar content suggestions. This interaction encourages users to further explore the website by presenting additional relevant information or resources that align with their interests.

**Quick Access to Features**: The sidebar can include widgets or modules that provide quick access to specific features or tools. For example, a home page promoting an e-commerce website may have a sidebar with a "Shopping Cart" widget that allows users to easily view and manage their cart from any page on the site.

**User Account Management**: If user accounts are required, the home page can display a summary or teaser of account-related information, while the sidebar provides a more comprehensive view. Users can access their profile, notifications, messages, or settings through the sidebar, allowing them to manage their account without leaving the home page.

**Contextual Information**: The sidebar can dynamically update based on the user's interaction with the home page. For example, if the home page features a search bar, the sidebar can display search results or filters in real-time as the user enters their query, providing immediate feedback and options for refining their search.

**Responsive Design**: In responsive web design, the sidebar may be hidden or transformed into a collapsible menu for smaller screens. On the home page, this interaction allows users to access the sidebar's content by toggling or expanding it, providing a seamless transition between the full view and mobile view of the website.

Overall, the interaction between the home page and the sidebar aims to enhance the user experience by offering additional navigation options, supplementary information, and quick access to relevant features. It provides users with flexibility and control in exploring and interacting with the website's content and functionality.

**Bezawit Alemayehu 56386**

**LOG IN / SIGN UP**

A screenshot of a login form

Description automatically generated with medium confidence

**Introduction**

The Sign-Up Page is an essential component of any web application or service that requires user registration by providing their personal information and credentials. It enables new users to create accounts and access the application's features and functionality. The main goals of sign-up pages include:

* User Registration: Collecting user information to create an account.
* Account Security: Verifying user identities and preventing unauthorized access.
* User Engagement: Encouraging users to complete the registration process.

Sign-up pages typically include the following essential components:

Registration Form: A form where users enter their information, such as first name, last name, age, email address, password, and additional relevant details.

**Optional field**

Phone number: For users to enter their phone number if the application requires it for specific features or communication.

Address: If the application relies on location-based services, consider including an optional field for users to enter their address or location details.

**Designing Sign-Up Pages**

It is important to Keep the registration form simple and easy to understand, using clear labels and instructions. Use a prominent and visually appealing call-to-action (CTA) button to guide users through the registration process.

Provide users with clear guidelines on creating strong passwords. Include a minimum number of characters, a mix of uppercase and lowercase letters, numbers, and special characters.

To ensure a positive user experience, implement error handling mechanisms to provide meaningful error messages when users make mistakes or input invalid information. Help users understand what went wrong and how to fix it. Proper error handling will help to increase user engagement with your application.

Confirmation Message: After successful registration, display a confirmation message or redirect users to a confirmation page. Provide clear instructions on the subsequent steps, such as email verification.

Email Verification: To ensure the authenticity of user accounts, send a verification email containing a unique link for users to confirm their email addresses. Only activate the account upon successful verification.

**Security**

Encrypt and store user passwords securely using industry-standard encryption algorithms like bcrypt or Argon2. Never store passwords in plain text.

Account lockout mechanism that temporarily blocks access after a certain number of failed login attempts, protecting against brute-force attacks. Secure Communication (HTTPS), use HTTPS (HTTP Secure) to encrypt the data transmitted between the Sign-Up Page and the server. Obtain an SSL/TLS certificate to enable secure communication.

**Privacy and Data Protection**

GDPR Compliance: Adhere to the General Data Protection Regulation (GDPR) guidelines if the application handles user data from European Union (EU) residents. Obtain user consent for data processing and provide clear privacy notices.

Data Retention Policy: The policy should specify how long user data will be stored and when it will be deleted. It's recommended to delete inactive or unused accounts after a certain period, as per the policy. By having a clear data retention policy, you can ensure that your application is compliant with data privacy laws and protect user data from being stored unnecessarily.

Privacy Policy and Terms of Service: Provide links to the Privacy Policy and Terms of Service during the sign-up process. Make sure these documents are easily accessible and clearly state the application's policies and user rights.

**Accessibility**

Keyboard Accessibility: Ensure all form fields and buttons can be accessed and interacted with using only the keyboard. Implement proper focus management to aid users in navigating through the form.

Screen Reader Compatibility: Design the Sign-Up Page to be compatible with screen readers, following the Web Content Accessibility Guidelines (WCAG) to provide an inclusive experience for visually impaired users.

Color Contrast and Visual Accessibility: A well-designed Sign-Up Page is crucial for facilitating user registration while ensuring security, privacy, and accessibility. By following the guidelines and best practices outlined in this documentation, you can create a seamless and user-friendly sign-up experience for your application's users. Remember to continuously test and iterate on the sign-up process to address any user feedback and improve the overall registration flow.

A screenshot of a login page

Description automatically generated with medium confidence

The Log-In Page is a fundamental component of any web application or service that requires user authentication. It allows registered users to access their accounts and utilize the application's features and functionality.

**Design Considerations**

The main idea of the design is User-Friendly interface, we must make sure that login page has a clean interface and making it easy for users to understand the login process. The developer needs to Optimize the Log-In Page for various screen sizes and devices to provide a seamless experience for all users.

Log-In page needs to be simple and clear to avoid confusion, use clear labels and placeholders for each input field to guide users.

Incase of users forgot password, must include a password recovery mechanism, such as “Forgot Password”, to connect users to their register email account with code to recover the account.

Captcha or Anti-Bot Mechanism: To enhance the security of user accounts, it's essential to implement a captcha or anti-bot mechanism in login process. This mechanism can prevent automated login attempts by bots or malicious actors. By implementing this security measure, you can protect user accounts from unauthorized access and ensure the confidentiality of user data.

**Required Fields**

Username/Email

Prompt users to enter their registered username or email address to identify their accounts.

Password

Ask users to enter their passwords using password input fields that hide the entered characters for security purposes.

**Optional Fields**

Remember Me

Remember Me is a checkbox option to allow users to stay logged in on the device or browser for a convenient login experience.

Two-Factor Authentication (2FA)

Offer the option for users to enable two-factor authentication as an additional security measure during the login process.

**Submit and Authentication**

Submit Button

Include a prominent and visually distinguishable "Submit" or "Log In" button to initiate the login process.

Authentication Process

Verify the provided username/email and password against the stored user credentials. Grant access upon successful authentication and redirect users to the appropriate page.

Password Recovery

Provide a password recovery mechanism, such as sending a reset link via email, to assist users who have forgotten their passwords.

Sirisa Kornnawawat 56482

**DELIVERY**

A screenshot of a food delivery service

Description automatically generated with low confidence

To access the delivery section in the web application, follow these steps:

1.Open the web application in your preferred browser.

2.Locate the application bar, typically positioned at the top or left side of the screen.

3.On the application bar, look for the delivery section. It is usually indicated by an icon or text label.

4.Click on the delivery section icon or label to access the delivery functionality.

5.Once inside the delivery section, you will see a selection box where you can choose the desired delivery option.

6.Select the appropriate delivery option from the available choices.

7.After selecting the delivery option, you will be prompted to enter the address information.

8.Enter the delivery address details, including the street name, building number, city, state, and postal code, in the provided input fields.

9.If available, a location map may automatically appear or be accessible through a separate map icon.

10.To ensure accurate delivery location, use the location map to pinpoint the exact spot where the delivery is required.

11.Save or submit the entered information, depending on the application's interface, to confirm the delivery details.

By following these steps, you will be able to enter the delivery section, select a delivery option, provide the address information, and utilize the location map if necessary.

A screenshot of a delivery form

Description automatically generated with low confidence

**City:** Please select the city where you currently reside from the available options.

**Street:** Enter the name of the street where your delivery address is located within your chosen city.

**Apartment Number:** Provide the exact apartment number to help locate your specific unit within the building or complex.

**Home Number:** Enter the exact home number associated with your address for a smooth and comfortable delivery experience.

Once you have filled out all the required information, a map will be displayed, pinpointing your location based on the provided details.

Please note that this is a fictional example for illustrative purposes, and I still don't have access to real-time map data or the ability to display maps. If you have any other questions or need further assistance, feel free to ask!

A map of a city

Description automatically generated with medium confidence

After entering all information your location int map will pop up.

Arslan Shaidilla 57056

**Personal Information**

A picture containing text, screenshot, font, logo

Description automatically generated

**Introduction:**

The Personal Information section of our food service application is designed to collect and store essential details about our users. This information helps us provide personalized experiences, seamless ordering, and efficient delivery services. We understand the importance of privacy and security, and we assure you that all personal data collected is handled in accordance with our privacy policy and applicable data protection laws.

**Types of Personal Information Collected:**

The Personal Information section may include, but is not limited to, the following types of data:

Full Name: We collect your full name to address you properly and ensure accurate communication.

Contact Information: This includes your phone number and email address. We need these details to keep you informed about your orders, delivery updates, and any relevant account-related information.

Delivery Address: To ensure prompt and accurate delivery of your food orders, we require your delivery address. You can save multiple addresses for your convenience.

Payment Information: We securely store your payment information, such as credit card details or preferred payment methods, to facilitate seamless transactions. Please note that sensitive payment information is encrypted and handled with utmost care.

Dietary Preferences: We provide options to capture your dietary preferences or restrictions, such as vegetarian, vegan, gluten-free, or allergies. This helps us tailor our food recommendations and menu offerings to better suit your needs.

Profile Picture: You have the option to upload a profile picture, which adds a personal touch to your account and helps us enhance your user experience.

**Usage and Purpose of Personal Information:**

We utilize the personal information collected in the following ways:

Order Processing: Your contact information, delivery address, and payment details are essential for processing your food orders accurately and ensuring smooth delivery.

Account Management: Personal information is used to create and maintain your user account, allowing you to manage orders, track delivery status, and access personalized features within the application.

Personalized Experiences: We may analyze your dietary preferences and ordering history to offer personalized recommendations, discounts, or promotions tailored to your tastes.

Communication: We use your contact information to send order confirmations, delivery updates, and promotional offers. You have the option to unsubscribe from marketing communications if desired.

Customer Support: Your personal information enables us to provide effective customer support, address inquiries, and resolve any issues related to your orders or account.

**Data Security and Privacy:**

We prioritize the security and privacy of your personal information. We implement industry-standard security measures to protect your data from unauthorized access, loss, or disclosure. Access to personal data is strictly limited to authorized personnel and is stored on secure servers. Please refer to our privacy policy for detailed information on data protection practices.

**User Control and Rights:**

You have control over your personal information and can update or modify it within the application's settings or profile section. You may also exercise your rights, such as data access, correction, or deletion, as permitted by applicable laws. For any concerns or queries regarding your personal information, please reach out to our support team.

Please review this documentation to understand how your personal information is collected, used, and protected within our food service application. By using our application, you consent to the collection and processing of your personal data as described in this documentation and our privacy policy.

Jaloliddin Sultonov 56296

A screenshot of a computer

Description automatically generated with medium confidence

Within the displayed image, you will find additional buttons that allow you to specify your preferences for a more tailored experience. The first line offers the option to choose a meal plan suitable for either an individual or a family. This selection impacts the quantity of food and further customization options in subsequent steps.

Moving to the second line, users are prompted to indicate any allergies they may have. This information is crucial in ensuring their safety and well-being.

In the third line, users can select specific products, fruits, or vegetables that are known to trigger allergic reactions. Our program takes these choices into account and adjusts the displayed food options accordingly, excluding any items containing the selected allergens.

Finally, in the last line, users have the opportunity to explore various global cuisines. As shown in the example, the image showcases Polish and Uzbek cuisines, but there are likely many more options available. This feature enables users to discover and explore different culinary traditions from around the world.

Jaloliddin Sultonov 56296

A black background with white text

Description automatically generated with low confidence

**STATISTICS**

Statistics play a crucial role in understanding and managing calorie spending. Here are five reasons why statistics are important in this context:

* Tracking and monitoring: Statistics provide a quantitative measurement of calorie expenditure, allowing individuals to track and monitor their daily energy expenditure. By analyzing calorie expenditure data over time, trends and patterns can be identified, helping individuals make informed decisions about their diet and exercise routines.
* Setting goals: Statistics enable individuals to set realistic and achievable goals for calorie expenditure. By understanding average calorie burn rates during different activities, individuals can set targets for weight loss, fitness goals, or maintaining a healthy lifestyle.
* Personalized planning: Calorie expenditure statistics can be used to create personalized plans based on an individual's unique characteristics and activities. Statistical data helps in determining the optimal caloric intake and expenditure for an individual's age, weight, height, and activity level.
* Comparisons and benchmarks: Statistics allow for meaningful comparisons and benchmarks. By comparing one's calorie expenditure to statistical averages or recommended guidelines, individuals can evaluate their progress and make necessary adjustments to their lifestyle and fitness routines.
* Research and advancements: Statistics play a significant role in research and advancements related to calorie expenditure. Through population-wide studies, researchers can gather data on energy expenditure patterns, factors influencing calorie burn, and the impact of various activities on overall health. This statistical analysis contributes to the development of evidence-based guidelines and strategies for effective weight management and improved overall health.

In summary, statistics provide a quantitative framework for understanding and managing calorie spending. They help individuals track progress, set goals, plan personalized routines, compare results, and contribute to scientific research in the field of energy expenditure.

A picture containing text, screenshot, font, line

Description automatically generated

**Description: Calorie Spending Monthly Tracker**

* Welcome to our Calorie Spending Monthly Tracker! This interactive form allows you to select a specific month and instantly generate a comprehensive line chart displaying your daily calorie expenditure for the chosen period. This valuable tool provides a visual representation of your calorie spending patterns, helping you gain insights into your energy expenditure and make informed decisions about your health and fitness goals.
* To use the form, simply select the desired month from the drop-down menu. Once you submit your selection, a line chart will appear, showcasing your daily calorie expenditure throughout the entire month. Each point on the chart represents the number of calories burned on a given day, allowing you to observe trends, variations, and notable spikes or dips in your energy expenditure.
* With this detailed visualization, you can track your progress, identify consistent patterns of activity or inactivity, and determine the effectiveness of your diet and exercise routines. By understanding your calorie expenditure on a day-to-day basis, you can make adjustments to your lifestyle, such as increasing physical activity or modifying your caloric intake, to align with your desired health goals.
* Remember, this form is designed to provide you with valuable insights into your calorie spending habits, empowering you to make informed decisions about your health and wellness journey. Use this tool regularly to monitor your progress, set realistic goals, and stay motivated on your path towards a healthier lifestyle.
* Please note that the data captured in this form is confidential and only accessible to you. We prioritize your privacy and ensure that your information remains secure.

Imamidin Galdarov 57105

A graph with blue lines

Description automatically generated with low confidence

**How It Works:**

Foodify revolutionizes the way you plan your meals and explore new culinary horizons. With a user-friendly interface and powerful backend algorithms, Foodify takes the guesswork out of meal planning and makes it easy and fun to discover new recipes and cuisines.

To get started, simply create an account on the Foodify platform. Once you're logged in, you can set your dietary preferences, indicating any allergies, restrictions, or specific cuisines you prefer. This ensures that the meal recommendations are tailored to your unique tastes and dietary needs.

After setting your preferences, Foodify's advanced algorithms swing into action. Leveraging machine learning and deep learning techniques, Foodify recommends over 100 dishes from different nations for you to enjoy. These recommendations take into account your preferences, ingredient availability, and other factors, ensuring a delightful culinary experience every time.

Once you have your personalized dish recommendations, it's time to plan your meals for the week ahead. Foodify provides a simple and intuitive meal planning interface where you can create personalized meal plans that fit your schedule and nutritional goals. You can easily browse through the recommended dishes, select the ones that catch your eye, and add them to your meal plan.

But that's not all - Foodify goes the extra mile to make your meal planning experience seamless and hassle-free. With the delivery option, you can have your weekly meals planned and delivered straight to your door. Simply select the meals you want, and Foodify collaborates with trusted delivery partners to handle the rest. This takes away the stress of grocery shopping and ensures you have everything you need for a hassle-free cooking experience.

Foodify also takes sustainability seriously. It helps you minimize food waste by recommending recipes that make use of ingredients you already have on hand. The app suggests ingredient substitutions when necessary and provides options for utilizing leftovers effectively, reducing waste and saving money. With Foodify, you can feel good about your meal choices and their impact on the environment.

To ensure you have everything you need for your selected recipes, Foodify generates a comprehensive grocery list based on your meal plan. This feature saves you time and ensures you won't forget any essential ingredients when you head to the grocery store or place an order for delivery. You can even seamlessly integrate Foodify with local grocery delivery services, allowing you to order the required ingredients directly from the app. This streamlines the meal planning process, eliminating the need for separate shopping trips.

Once you have your ingredients ready, Foodify provides step-by-step recipe instructions for each dish in your meal plan. The instructions are clear and easy to follow, making it a breeze to prepare delicious meals. Whether you're a seasoned home cook or just starting your culinary journey, Foodify's recipe guidance will help you create impressive and satisfying dishes.

In addition to the practical aspects of meal planning, Foodify also offers social features to enhance your experience. You can share your favorite recipes or entire meal plans with friends and family through social media platforms directly from the app. This fosters a sense of community and allows you to inspire and be inspired by others in the Foodify community. Furthermore, you can rate and leave reviews for recipes you have tried, helping others make informed decisions when planning their meals.

Foodify also offers valuable nutritional information for each recipe, empowering you to make informed decisions about your meals. You can access detailed information about the calorie content, macronutrient composition, and other key nutritional aspects of each dish. This helps you maintain a healthy and balanced diet, ensuring that your meal choices align with your specific dietary goals.

**Contact:**

At Foodify, we are committed to providing exceptional customer support and value your feedback. If you have any questions,

suggestions, or concerns regarding Foodify, our dedicated support team is here to assist you. You can reach out to us through the contact form on our website, where you can provide details about your inquiry, and we'll get back to you promptly.

Alternatively, you can contact our customer support team directly via email or phone. Our knowledgeable and friendly representatives are ready to help address any issues or answer any queries you may have.

Stay connected with Foodify to stay up-to-date with the latest news, recipe inspiration, and community engagement. Follow us on our social media channels, where we regularly share exciting recipes, cooking tips, and more. Join our growing community of food enthusiasts and be part of the conversation.

We value your input and appreciate your contributions to making Foodify the best meal planning experience possible. Your feedback helps us continually improve the app and deliver the features and functionality that matter most to you.

Experience the future of meal planning with Foodify. Embrace the convenience, variety, and sustainability it offers, and embark on a culinary journey like no other. Sign up today and unlock a world of delicious possibilities!

**Iskhak Suranov 55399**

**DIAGRAM AND USE CASE**

Zeeshan Jilani 56855

Jeremiah Kalonga Lusajo 56288

Cuba Ogott 56978

Team Project

Diagram

Description automatically generated

**Use case:** Register Account in Foodify Application

**Preconditions and Trigger:**

The user has internet access and a device capable of accessing the food recommender application.

The user has not previously registered an account with the application.

**Basic Flow:**

The user navigates to the food recommender application's website or opens the application on their device.

The user selects the "Register" option.

The application prompts the user to enter their first name, last name, email address, and a password.

The user enters their personal information and creates a secure password.

The application verifies that the user's email address is not already registered in the system.

If the email address is new, the application sends a verification email to the user's email address.

The user confirms their email address by clicking on the verification link in the email.

The application sends a confirmation message to the user that their account has been successfully registered.

The user can now log in to the food recommender application with their email address and password.

**Extensions:**

If the user enters an email address that is already registered in the system, the application prompts the user to try again with a different email address.

If the user's password does not meet the minimum security requirements, the application prompts the user to create a stronger password.

If the user encounters any errors or issues during the registration process, the application provides appropriate error messages and suggestions to help the user complete the registration process.

**Postconditions:**

The user has a registered account with foodify application.

The user can log in to the application.

**Introduction**

Welcome to the documentation for the Food Delivery Management System Database. This document serves as a comprehensive guide to understand the structure, usage, and maintenance of the database that supports our organization's deliver operations. The Food delivery Management System Database is built on a relational database management system (DBMS) and plays a vital role in storing and managing user-product-related data for our company.

The purpose of this documentation is to provide a clear understanding of the database architecture, schema definition, data access and change management processes. By following this guide, both technical and non-technical stakeholders will be able to navigate and utilize the database effectively.

The Food Delivery Management System Database follows a relational data model, allowing for efficient and structured storage of information. It comprises various tables, views, and other database objects that collectively represent the entities, their attributes, and the relationships among them. This documentation will delve into the details of each table, including the column names, data types, constraints, and relational links.

We encourage you to explore this documentation thoroughly and refer to it whenever you need to interact with or understand the Food Delivery Management System Database. Your feedback is valuable in helping us improve and maintain the database effectively.

**Database Architecture**

The Food Delivery Management System Database follows a client-server architecture, utilizing a relational database management system (DBMS) to store and manage sales-related data. The database architecture is designed to provide a scalable, secure, and efficient solution for handling our organization's sales operations.

At the core of the database architecture is the DBMS, which acts as the central hub for storing, organizing, and retrieving data. In our case, we have chosen MySQL as the preferred DBMS due to its reliability, performance, and extensive feature set.

The Sales Management System Database architecture consists of the following key components:

1. Database Server:

* The database server hosts the DBMS and manages the storage and retrieval of data.
* It runs on a dedicated machine with sufficient computational resources and memory to handle concurrent user requests.
* The server is responsible for executing SQL queries, enforcing data integrity through constraints, and managing database transactions.

2. Application Layer:

* The application layer comprises the software components that interact with the database server.
* It includes the user interface, business logic, and application server responsible for processing user requests, handling data validation, and enforcing business rules.
* The application layer communicates with the database server using SQL queries and manages the data retrieval and update processes.

3. Database Schema:

* The database schema defines the structure of the database and determines how data is organized and related.
* It consists of a set of tables, each representing a specific entity or concept in the sales domain, such as customers, products, orders, and sales representatives.
* The tables are defined with appropriate columns, data types, constraints, and relationships, ensuring data integrity and consistency.

The Food Delivery Management System Database architecture is designed to provide a scalable and reliable foundation for our sales operations. It leverages industry best practices to ensure data integrity, high performance, and security throughout the system.

**Schema Definition**

The Food Delivery Management System Database is built on a relational data model and comprises several tables that collectively represent the various entities and relationships in our sales domain. This section provides an overview of the database schema, detailing the tables, their columns, data types, constraints, and relationships.

1. Table: USER

* Description: Store information about our user.
* Columns:
  + user\_ID (integer(10), primary key): Unique identifier for each user.
  + Firstname (varchar(50)): First name of the user.
  + Lastname (varchar(50)): Last name of the user.
  + Email (varchar(50)): Email of the user.
  + Password (varchar(50)): Password of the user.
  + address\_ID (integer(10), foreign key to ADDRESS): References the user’s address.
  + PhoneNumber (varchar(20)): Phone number of the user.
  + Role (varchar(20)): Role of the user.

2. Table: PRODUCT

* Description: Contains details of the products we offer.
* Columns:
  + product\_ID (integer(10), primary key): Unique identifier for each product.
  + Name (varchar(50)): Name of the product.
  + Description (varchar(255)): Description of the product.
  + Price (decimal(10,2)): Price of the product.
  + Image (blob): Image of the product.
  + Basket\_ID (integer(10), foreign key to BASKET): References the product in the basket.
  + allergies\_ID (integer(10), foreign key to ALLERGIES): References the product’s allergies.

3. Table: BASKET

* Description: Contains details of the basket.
* Columns:
  + basket\_ID (integer(10), primary key): Unique identifier for each basket.
  + TotalAmount (integer(10)): Total number of product.
  + user\_ID (integer(10), foreign key to USER): References the basket’s user contain.
  + tracking\_ID (integer(10), foreign key to TRACKING): References the basket’s tracking.

4. Table: TRACKING

* Description: Contains details of the tracking.
* Columns:
  + tracking\_ID (integer(10), primary key): Unique identifier for each tracking.
  + Delivery\_ID (integer(10), foreign key to DELIVERY): References the tracking’s delivery.
  + OrderNumber (integer(10)): Number of the order.
  + TrackingNumber (integer(10)): Number of the tracking.
  + CourierName (varchar(50)): Name of the courier.
  + CoirierNumber (integer(10)): Number of the courier.

5. Table: DELIVERY

* Description: Contains details of the delivery.
* Columns:
  + delivery\_ID (integer(10), primary key): Unique identifier for each delivery.
  + address\_ID (integer(10), foreign key to ADDRESS): References the deliver’s address.
  + DeliveryDate (date): Date that delivery will be deliver.
  + DeliveryTimeSlot (time(7)): Slot time that the deliver will be delivery.
  + DeliveryStatus (varchar(20)): Confirmation status for the delivery.

6. Table: ADDRESS

* Description: Contains details of the address.
* Columns:
  + address\_ID (integer(10), primary key): Unique identifier for each address.
  + HouseNumber (varchar(20)): House’s number.
  + ApartmentNumber (varchar(20)): Apartment’s number.
  + Street (varchar(50)): Street.
  + City (varchar(50)): City.
  + PostCode (varchar(10)): Post code.

7. Table: PAYMENT

* Description: Contains details of the payment.
* Columns:
  + payment\_ID (integer(10), primary key): Unique identifier for each payment.
  + basket\_ID (integer(10), foreign key to BASKET): References the basket’s payment.
  + PaymentMethod (varchar(255)): Describe the payment method.
  + TransactionAmount (decimal(10,2)): Total amount of payment.
  + TransactionDate (date): Date that the payment has been made.
  + Status (varchar(20)): The confirmation of the payment.

8. Table: HISTORY\_ORDER

* Description: Contains details of the history order.
* Columns:
  + history\_ID (integer(10), primary key): Unique identifier for each history.
  + delivery\_ID (integer(10), foreign key to DELIVERY): References the history of delivery.
  + basket\_ID (integer(10), foreign key to BASKET): References the stock of the product.
  + payment\_ID (integer(10), foreign key to PAYMENT): References the payment made.

9. Table: RECIPE

* Description: Contains details of the recipe.
* Columns:
  + recipe\_ID (integer(10), primary key): Unique identifier for each recipe.
  + user\_ID (integer(10), foreign key to USER): References the user’s details.
  + payment\_ID (integer(10), foreign key to PAYMENT): References the payment’s details.
  + basket\_ID (integer(10), foreign key to BASKET): References the basket’s details.
  + RecipeImage (blob): Image of the recipe.

10. Table: PROMOTION

* Description: Contains details of the promotion.
* Columns:
  + promotion\_ID (integer(10), primary key): Unique identifier for each promotion.
  + Name (varchar(50)): Name of the promotion.
  + Description (varchar(255)): Description of the promotion.
  + DiscountType (varchar(50)): Type of the promotion.
  + DiscountAmount (decimal(10,2)): The amount of discount on the promotion.
  + StartDate (date): Starting date of the promotion.
  + EndDate (date): Ending date of the promotion.
  + product\_ID (integer(10), foreign key to PRODUCT): References the product on promotion.

11. Table: EVENT

* Description: Contains details of the event.
* Columns:
  + event\_ID (integer(10), primary key): Unique identifier for each event.
  + product\_ID (integer(10), foreign key to PRODUCT): References the product on event.
  + promotion\_ID (integer(10), foreign key to PROMOTION): References the promotion on event.
  + Name (varchar(50)): Name of the event.
  + Description (varchar(255)): Description of the event.
  + StartDate (date): Starting date of the event.
  + EndDate (date): Ending date of the event.
  + Organizer\_info (varchar(255)): Information of the organizer.

12. Table: REVIEW

* Description: Contains details of the review.
* Columns:
  + review\_ID (integer(10), primary key): Unique identifier for each review.
  + user\_ID (integer(10), foreign key to USER): References the user’s details.
  + product\_ID (integer(10), foreign key to PRODUCT): References the product mention.
  + Rating (integer(10)): Rating given by user.
  + Comment (varchar(255)): Comment given by user.
  + Date (timestamp): Date and time that the user give review.

13. Table: ALLERGIES

* Description: Contains details of the allergies.
* Columns:
  + allergies\_ID (integer(10), primary key): Unique identifier for each allergies.
  + user\_ID (integer(10), foreign key to USER): References the user’s details.
  + Allergen (varchar(50)): Name of the allergies.
  + SeverityLevel (varchar(255)): a measurement of the impact an incident.

14. Table: PREFERENCE

* Description: Contains details of the preference.
* Columns:
  + preference\_ID (integer(10), primary key): Unique identifier for each preference.
  + user\_ID (integer(10), foreign key to USER): References the user’s details.
  + PreferenceIngredients (varchar(255)): Based on the feature you notice.
  + DietaryRestriction (varchar(255)): A reduction of particular or total nutrient intake without causing malnutrition
  + NotificationSetting (varchar(255)): Allow users to make choices about the kind of push notifications they want to receive

15. Table: HELP

* Description: Contains details of the help.
* Columns:
  + help\_ID (integer(10), primary key): Unique identifier for each help.
  + user\_ID (integer(10), foreign key to USER): References the user’s details.
  + messenger\_ID (integer(10), foreign key to MESSENGER): References the messenger’s details.
  + Topic (varchar(255)): Topic which user want help form messenger.
  + Date (date): Date that the conversation happens.

16. Table: MESSENGER

* Description: Contains details of the messenger.
* Columns:
  + messenger\_ID (integer(10), primary key): Unique identifier for each messenger.
  + user\_ID (integer(10), foreign key to USER): References the user’s details.
  + MessengerContent (varchar(255)): Conversation between user and messenger.

The above tables represent a simplified view of the database schema, focusing on the key entities involved in our sales management system. In practice, additional tables and relationships may exist, depending on the specific requirements and complexity of the system.

These tables are interconnected through primary key and foreign key relationships, enabling the establishment of meaningful associations between different entities. The schema design ensures data integrity by enforcing referential integrity constraints and defining appropriate indexes to optimize query performance.

By understanding the schema and the relationships between tables, you can effectively query and manipulate the data stored in the Food Delivery Management System Database.

**Data Dictionary**

The data dictionary provides a comprehensive reference for all the entities, attributes, and relationships present in the Food Delivery Management System Database. This section presents an overview of the data dictionary, including the definitions, data types, and additional information for each item.

1. Table: USER

* user\_ID
  + Definition: Unique identifier for each user.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* Firstname
  + Definition: First name of the user.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* Lastname
  + Definition: Last name of the user.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* Email
  + Definition: Email of the user.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* Password
  + Definition: Password of the user.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* address\_ID
  + Definition: References the user’s address.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the address\_id column in the ADDRESS table.
* PhoneNumber
  + Definition: Phone number of the user.
  + Data Type: Variable-length string with a maximum length of 20 characters.
* Role
  + Definition: Role of the user.
  + Data Type: Variable-length string with a maximum length of 20 characters.

2. Table: PRODUCT

* product\_ID
  + Definition: Unique identifier for each product.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* Name
  + Definition: Name of the product.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* Description
  + Definition: Description of the product.
  + Data Type: Variable-length string with a maximum length of 255 characters.
* Price
  + Definition: Price of the product.
  + Data Type: Decimal number with a precision of 10 and a scale of 2.
* Image (blob)
  + Definition: Image of the product.
  + Data Type: blob.
* basket\_ID
  + Definition: References the product in the basket.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the basket\_id column in the BASKET table.
* allergies\_ID
  + Definition: References the product’s allergies.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the allergies\_id column in the ALLERGIES table.

3. Table: BASKET

* basket\_ID
  + Definition: Unique identifier for each basket.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* TotalAmount
  + Definition: Total number of products.
  + Data Type: Integer.
* user\_ID
  + Definition: References the basket’s user contain.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the user\_id column in the USER table.
* tracking\_ID
  + Definition: References the basket’s tracking.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the tracking\_id column in the TRACKING table.

4. Table: TRACKING

* tracking\_ID
  + Definition: Unique identifier for each tracking.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* delivery\_ID
  + Definition: References the tracking’s delivery.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the delivery\_id column in the DELIVERY table.
* OrderNumber
  + Definition: Number of the order.
  + Data Type: Integer.
* TrackingNumber
  + Definition: Number of the tracking.
  + Data Type: Integer.
* CourierName
  + Definition: Name of the courier.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* CoirierNumber
  + Definition: Number of the courier.
  + Data Type: Integer.

5. Table: DELIVERY

* delivery\_ID
  + Definition: Unique identifier for each delivery.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* address\_ID
  + Definition: References the deliver’s address.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the address\_id column in the ADDRESS table.
* DeliveryTimeSlot
  + Definition: Slot time that the deliver will be delivery
  + Data Type: time
* DeliveryStatus
  + Definition: Confirmation status for the delivery.
  + Data Type: Variable-length string with a maximum length of 20 characters.

6. Table: ADDRESS

* address\_ID
  + Definition: Unique identifier for each address.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* HouseNumber
  + Definition: House’s number.
  + Data Type: Variable-length string with a maximum length of 20 characters.
* ApartmentNumber
  + Definition: Apartment’s number.
  + Data Type: Variable-length string with a maximum length of 20 characters.
* Street
  + Definition: Street’s number.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* City
  + Definition: City’s name.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* PostCode
  + Definition: Post code number.
  + Data Type: Variable-length string with a maximum length of 10 characters.

7. Table: PAYMENT

* payment\_ID
  + Definition: Unique identifier for each payment.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* basket\_ID
  + Definition: References the basket’s payment.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the basket\_id column in the BASKET table.
* PaymentMethod
  + Definition: Describe the payment method.
  + Data Type: Variable-length string with a maximum length of 255 characters.
* TransactionAmount
  + Definition: Total amount of payment.
  + Data Type: Decimal number with a precision of 10 and a scale of 2.
* TransactionDate
  + Definition: Date that the payment has been made.
  + Data Type: date.
* Status
  + Definition: The confirmation of the payment.
  + Data Type: Variable-length string with a maximum length of 20 characters.

8. Table: HISTORY\_ORDER

* history\_ID
  + Definition: Unique identifier for each history.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* delivery\_ID
  + Definition: References the history of delivery.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the delivery\_id column in the DELIVERY table.
* basket\_ID
  + Definition: References the stock of the product.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the basket\_id column in the BASKET table.
* payment\_ID
  + Definition: References the payment made.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the payment\_id column in the PAYMENT table.

9. Table: RECIPE

* recipe\_ID
  + Definition: Unique identifier for each recipe.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* user\_ID
  + Definition: References the user’s details.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the user\_id column in the USER table.
* payment\_ID
  + Definition: References the payment’s details.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the payment\_id column in the PAYMENT table.
* basket\_ID
  + Definition: References the basket’s details
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the basket\_id column in the BASKET table.
* RecipeImage
  + Definition: Image of the recipe.
  + Data Type: blob.

10. Table: PROMOTION

* promotion\_ID
  + Definition: Unique identifier for each promotion.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* Name
  + Definition: Name of the promotion.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* Description
  + Definition: Description of the promotion.
  + Data Type: Variable-length string with a maximum length of 255 characters.
* DiscountType
  + Definition: Type of the promotion.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* DiscountAmount
  + Definition: The amount of discount on the promotion.
  + Data Type: Decimal number with a precision of 10 and a scale of 2.
* StartDate
  + Definition: Starting date of the promotion.
  + Data Type: date
* EndDate
  + Definition: Ending date of the promotion.
  + Data Type: date
* product\_ID
  + Definition: References the product on promotion.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the product \_id column in the PRODUCT table.

11. Table: EVENT

* event\_ID
  + Definition: Unique identifier for each event.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* product\_ID
  + Definition: References the product on event.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the product \_id column in the PRODUCT table.
* promotion\_ID
  + Definition: References the promotion on event.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the promotion \_id column in the PROMOTION table.
* Name
  + Definition: Name of the event.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* Description
  + Definition: Description of the event.
  + Data Type: Variable-length string with a maximum length of 255 characters.
* StartDate
  + Definition: Starting date of the event.
  + Data Type: date
* EndDate
  + Definition: Ending date of the event.
  + Data Type: date
* Organizer\_info
  + Definition: Information of the organizer.
  + Data Type: Variable-length string with a maximum length of 255 characters.

12. Table: REVIEW

* review\_ID
  + Definition: Unique identifier for each review.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* user\_ID
  + Definition: References the user’s details.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the user\_id column in the USER table.
* product\_ID
  + Definition: References the product mention.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the product \_id column in the PRODUCT table.
* Rating
  + Definition: Rating given by user.
  + Data Type: Integer.
* Comment
  + Definition: Comment given by user.
  + Data Type: Variable-length string with a maximum length of 255 characters.
* Date
  + Definition: Date and time that the user give review.
  + Data Type: timestamp

13. Table: ALLERGIES

* allergies\_ID
  + Definition: Unique identifier for each allergy.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* user\_ID
  + Definition: References the user’s details.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the user\_id column in the USER table.
* Allergen
  + Definition: Name of the allergies.
  + Data Type: Variable-length string with a maximum length of 50 characters.
* SeverityLevel
  + Definition: a measurement of the impact an incident.
  + Data Type: Variable-length string with a maximum length of 255 characters.

14. Table: PREFERENCE

* preference\_ID
  + Definition: Unique identifier for each preference.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* user\_ID
  + Definition: References the user’s details.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the user\_id column in the USER table.
* PreferenceIngredients
  + Definition: Based on the feature you notice.
  + Data Type: Variable-length string with a maximum length of 255 characters.
* DietaryRestriction
  + Definition: A reduction of particular or total nutrient intake without causing malnutrition.
  + Data Type: Variable-length string with a maximum length of 255 characters.
* NotificationSetting
  + Definition: Allow users to make choices about the kind of push notifications they want to receive.
  + Data Type: Variable-length string with a maximum length of 255 characters.

15. Table: HELP

* help\_ID
  + Definition: Unique identifier for each help.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* user\_ID
  + Definition: References the user’s details.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the user\_id column in the USER table.
* messenger\_ID
  + Definition: References the messenger’s details.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the messenger\_id column in the MESSENGER table.
* Topic
  + Definition: Topic which user want help form messenger.
  + Data Type: Variable-length string with a maximum length of 255 characters.
* Date
  + Definition: Date that the conversation happens.
  + Data Type: timestamp

16. Table: MESSENGER

* messenger\_ID
  + Definition: Unique identifier for each messenger.
  + Data Type: Integer.
  + Additional Information: Primary Key of the table.
* user\_ID
  + Definition: References the user’s details.
  + Data Type: Integer.
  + Additional Information: Foreign key referencing the user\_id column in the USER table.
* MessengerContent
  + Definition: Conversation between user and messenger.
  + Data Type: Variable-length string with a maximum length of 255 characters.

By referring to the data dictionary, users can quickly understand the purpose and structure of each table, facilitating efficient data access, querying, and manipulation. The data dictionary acts as a valuable resource for developers, administrators, and stakeholders, providing a clear overview of the database's schema and supporting a comprehensive understanding of the Food Delivery Management System Database.

**Entity-Relationship Diagram (ERD)**

**A picture containing text, screenshot, handwriting, rectangle

Description automatically generated**

An Entity-Relationship Diagram (ERD) is a visual representation of the entities (objects), attributes, and relationships within a database. It serves as a powerful tool for designing, organizing, and understanding the structure of a database system. ERD diagrams provide a clear and concise depiction of how entities relate to each other and the attributes that define them.

ERD diagrams are commonly used in database design and development processes to create a blueprint for the database schema. They facilitate effective communication between stakeholders, including database designers, developers, and business analysts, by providing a visual representation of the database structure and its relationships.

The key components of an ERD diagram include entities, attributes, and relationships:

1. Entities: Entities represent real-world objects or concepts within the domain of the database. Each entity is depicted as a rectangle in the ERD diagram, with the entity name written inside it. Examples of entities could include "USER," "PRODUCT," or "DELIVERY."

2. Attributes: Attributes define the characteristics or properties of an entity. They provide details about the entity and are represented as ovals or ellipses connected to the respective entity rectangle in the diagram. Examples of attributes for a "USER" entity could include "user\_ID," "FirstName," "LastName," "Email," and "Phone."

3. Relationships: Relationships illustrate the associations or connections between entities in the database. They demonstrate how entities interact with each other and define the rules governing those associations. Relationships are typically represented as lines connecting the related entities, with labels describing the type of relationship (e.g., one-to-one, one-to-many, many-to-many).

ERD diagrams provide a visual roadmap for understanding the structure and flow of data within a database. They aid in identifying the relationships between entities, determining the cardinality of those relationships, and defining the attributes required to represent the data accurately.

By utilizing ERD diagrams, stakeholders can collaborate effectively, ensure data integrity, and validate the database design against the requirements of the system or application. ERD diagrams are a crucial tool in the early stages of database design and provide a foundation for creating the database schema, implementing relationships, and optimizing data retrieval and manipulation operations.

**Database Operation**

Database operations are fundamental actions performed on a database to manage and manipulate data stored within it. These operations include creating, reading, updating, and deleting data, aggregating data, and executing complex queries. Understanding and documenting these operations are essential for effectively interacting with the database and ensuring proper data management.

By documenting database operations, users gain insight into the capabilities of the database management system and its specific functionalities. They can execute queries, perform data manipulations, and retrieve relevant information to support decision-making processes and drive system functionality.

1. Create Operation:

* Description: Creating a new table or database object.
* Example:

CREATE TABLE [USER] (

user\_ID INT PRIMARY KEY,

Firstname VARCHAR(50),

Lastname VARCHAR(50),

Email VARCHAR(50),

Password VARCHAR(50),

address\_id INT FOREIGN KEY REFERENCES ADDRESS(address\_ID),

PhoneNumber INT,

Role VARCHAR(20)

);

2. Read Operation:

* Description: Retrieving data from one or more tables.
* Example:

SELECT user\_ID, Firstname, Lastname, Email, PhoneNumber

FROM [USER];

3. Update Operation:

* Description: Modifying existing data in a table.
* Example:

UPDATE [USER]

SET Email = 'newemail@example.com'

WHERE user\_ID = 1;

4. Delete Operation:

* Description: Removing a record or records from a table.
* Example:

DELETE FROM USER

WHERE user\_ID = 1;

These operations are crucial to document as they cover the fundamental actions performed on the database, including creating, reading, updating, and deleting data. Including examples and explanations of these operations in the database documentation helps users understand how to interact with the database effectively and accurately perform the required tasks.

**Data Access**

Data access is a fundamental aspect of working with databases, enabling users to retrieve, manipulate, and interact with the stored data. It involves querying the database to retrieve specific information, modifying existing data, and performing various operations to access and utilize the data effectively. Understanding the principles of data access is crucial for leveraging the full potential of a database and extracting valuable insights from the stored information.

Data access encompasses a range of operations, including retrieving records, filtering data based on specific criteria, sorting results, joining multiple tables, performing calculations, and aggregating data. These operations allow users to access the data in a structured and meaningful way, supporting various business processes, decision-making, and analysis.

1. Retrieve All Records:

* Description: Retrieving all records from a specific table.
* Example:

SELECT \* FROM [USER];

2. Filter Records with WHERE Clause:

* Description: Retrieving specific records based on specified criteria.
* Example:

SELECT \* FROM [USER] WHERE Role = 'Customer';

3. Sorting Records with ORDER BY

* Description: Retrieving records in a specific order.
* Example:

SELECT \* FROM [USER] ORDER BY Lastname ASC;

4. Joining Multiple Tables:

* Description: Combining data from multiple tables based on common columns.
* Example:

SELECT u.user\_ID, u.Firstname, a.address\_ID, a.City

FROM [USER] u

JOIN ADDRESS a ON u.user\_ID = a.address\_ID;

5. Aggregate Function and Grouping:

* Description: Performing calculations and summarizing data across multiple records.
* Example:

SELECT Name, COUNT(\*) AS Total\_Products, AVG(Price) AS Average\_Price

FROM PRODUCT

GROUP BY Name;

Data access is a fundamental aspect of working with databases, enabling users to retrieve, manipulate, and interact with stored data. It involves performing operations such as querying, filtering, sorting, joining tables, and aggregating data to retrieve meaningful information from the database.

Documentation of data access operations provides guidelines, examples, and best practices for executing queries accurately and efficiently. It ensures consistency, data integrity, and collaboration among users working with the database.

**Backup and Recovery**

1. Backup Process:

* Description: Creating a backup of the database to ensure data durability and facilitate recovery in case of system failures or data loss.
* Example:
  + Performing regular daily backups using a database backup utility or scheduled scripts.
  + Storing the backups in a secure offsite location or separate storage media to minimize the risk of data loss.
  + Employing full backups or a combination of full and incremental backups based on the organization's requirements and data volume.
  + Ensuring backups are validated and verified regularly to guarantee their integrity and completeness.

2. Recovery Process:

* Description: Restoring the database from a backup in case of system failures, data corruption, or accidental deletions.
* Example:
  + Identifying the cause of the database failure or data loss and determining the appropriate recovery strategy.
  + Restoring the most recent valid backup from the backup storage.
  + Applying incremental backups, transaction logs, or other relevant backups to recover data changes made after the last full backup.
  + Verifying the integrity and consistency of the restored data through data validation and consistency checks.
  + Testing the recovered database to ensure its functionality and the availability of critical data.
  + Communicating the recovery status to relevant stakeholders and documenting the recovery process for future reference.

Backup and recovery are critical processes to ensure the availability, durability, and integrity of the database. They protect against unforeseen events that could result in data loss or system downtime. By regularly backing up the database and following robust recovery procedures, organizations can mitigate risks and quickly restore the system to an operational state.

It is essential to customize backup and recovery strategies based on specific requirements, such as recovery point objectives (RPO) and recovery time objectives (RTO), to minimize data loss and system downtime.

Kanisorn Jinta-Achariya 55477

**Security**

**Introduction:**

The purpose of this security plan is to outline the measures that will be taken to ensure the security of the grocery shopping app. The app aims to help users budget their groceries and avoid food waste by only letting them buy items they can consume before they go bad. To ensure the confidentiality, integrity, and availability of user data and resources, a comprehensive security plan must be put in place.

**Threat and Risk Assessment:**

The following threats have been identified as potential risks to the security of Foodify:

1. Unauthorized access to user data

2. Breach of payment processing system

3. Loss or corruption of user data

4. Denial of service attacks

5. Malicious code or software

Each threat has been assessed based on its likelihood of occurrence and potential impact on the application and its users. The following risk assessment matrix was used to prioritize the identified risks:

| Likelihood / Impact | High | Medium | Low |

|---------------------------|--------|------------|-------|

| High                           | 1        | 2             | 3      |

| Medium                    | 2        | 3             | 4      |

| Low                            | 3        | 4             | 5      |

Based on this matrix, the following risks have been identified as high priority:

1. Unauthorized access to user data

2. Breach of payment processing system

These risks will be addressed through the implementation of robust security controls and access controls, which will be described in more detail in the next section of this security plan.

1.Unauthorized access to user data: This threat poses a significant risk to the security of the grocery shopping app, as it could result in the theft or misuse of sensitive user data. To mitigate this risk, the following security controls will be implemented:

* Strong password requirements: Users will be required to create strong passwords that include a mix of upper- and lower-case letters, numbers, and special characters.
* Multi-factor authentication: Two-factor authentication will be required for certain actions within the app, such as changing account settings or making a purchase.
* Data encryption: User data will be encrypted both in transit and at rest, using strong encryption algorithms and protocols.
* Role-based access control: Access to user data will be restricted based on user roles and permissions, ensuring that only authorized individuals have access to sensitive data.

2.Breach of payment processing system: Given that the grocery shopping app includes a payment processing system, the risk of a breach in this area is significant. To mitigate this risk, the following security controls will be implemented:

* PCI compliance: The app will be designed and maintained in accordance with Payment Card Industry Data Security Standards (PCI DSS) requirements, which set the standard for secure payment processing.
* Encryption of payment data: Payment data will be encrypted using strong encryption algorithms and protocols, both in transit and at rest.
* Tokenization of payment data: Payment data will be tokenized to reduce the amount of sensitive data stored within the app.
* Regular security testing: Regular security testing will be conducted on the payment processing system to identify and address vulnerabilities before exploitation.

Other risks that have been identified, but are considered lower priority, include loss or corruption of user data, denial of service attacks, and malicious code or software. These risks will also be addressed through the implementation of appropriate security controls.

In addition to the above security controls, the following access controls will be implemented:

* User authentication: Users must authenticate themselves to access the app, using a combination of email and password.
* Session management: User sessions will be managed securely, with sessions being terminated after a certain period of inactivity, and users being logged out when they close the app or navigate away from it.
* Account management: Users will be able to manage their own account settings, including changing their password and updating their personal information.

By implementing these security and access controls, the risk of unauthorized access, data loss or corruption, and other security breaches will be significantly reduced, ensuring the confidentiality, integrity, and availability of user data and resources.

Security Controls:

1. Network Security: The grocery shopping app will use secure network protocols to ensure that data is transmitted securely over the internet. This will include the use of Secure Socket Layer (SSL) encryption for all communication between the app and the servers. In addition, the app will be designed to avoid transmitting unnecessary data over the network, to reduce the risk of eavesdropping or interception.
2. System Security: The servers used to host the grocery shopping app will be hardened against attacks, by implementing the following security controls:

* Security system protection: The servers will be protected by a security system that will block unauthorized access attempts.
* Regular updates: All software running on the servers will be kept up to date with the latest security patches and updates.
* Intrusion detection and prevention: The servers will be monitored for suspicious activity, with alerts being sent to the app's administrators if any unusual activity is detected.

1. Data Security: The grocery shopping app will be designed with data security in mind to ensure user data is stored and processed securely. This will include the following security controls:

* Data backup and recovery: Regular backups of the app's data will be taken to ensure data can be recovered in the event of a disaster or system failure.
* Data validation: All user input will be validated to ensure it is safe and does not contain any malicious code or data.
* Data segregation: User data will be segregated from other data within the app, to prevent accidental or unauthorized access.

1. Personnel Security: The app's administrators and support staff will be carefully selected and trained to ensure they have the skills and knowledge required to maintain the app's security. This will include the following measures:

* Background checks: All personnel with access to sensitive data will undergo a background check to ensure they do not have a history of criminal activity or other security issues.
* Security training: All personnel who will have access to sensitive data will receive security training to ensure they understand the importance of security and are aware of the app's risks and threats.
* Access control: Personnel access to sensitive data will be limited to the minimum necessary to perform their job functions.

**Secure Communication and API Integration for Foodify**

Introduction The secure communication and API integration are critical aspects of ensuring the security and reliability of a food delivery app like Foodify. This document aims to provide an overview of the key considerations and best practices for establishing secure communication channels and implementing secure APIs within the app's architecture. By following these guidelines, you can enhance the confidentiality, integrity, and availability of user data, protect against unauthorized access, and mitigate common web vulnerabilities.

Transport Layer Security (TLS) Transport Layer Security (TLS), previously known as Secure Sockets Layer (SSL), is a crucial technology for securing communication channels over networks. Implementing TLS ensures encrypted data transmission and protects against eavesdropping, data tampering, and man-in-the-middle attacks. Key points to consider for secure communication using TLS include:

* Selection of strong TLS protocols and cipher suites
* Proper configuration of certificates and certificate authorities
* Periodic renewal and revocation of certificates
* Enforcing HTTPS for all communication between the app and backend servers
* Regular vulnerability assessments and patches for known TLS vulnerabilities

Authentication and Authorization Authentication and authorization mechanisms play a vital role in securing API endpoints and preventing unauthorized access. Key considerations in this area include:

* Implementation of strong user authentication methods (e.g., username/password, multi-factor authentication)
* Proper handling of authentication tokens and session management
* Role-based access control (RBAC) to enforce granular access rights
* Thorough input validation to prevent injection attacks (e.g., SQL injection, cross-site scripting)
* Rate limiting and throttling to prevent API abuse and protect against DoS attacks
* Secure handling of API keys and secrets, using techniques such as encryption and secure key management systems

Input Validation and Sanitization Proper input validation and sanitization are essential for protecting against common web vulnerabilities. The following practices should be adopted:

* Validating and sanitizing all user input, including query parameters, request bodies, and headers
* Implementing a robust input validation mechanism to detect and prevent malicious inputs
* Utilizing parameterized queries or prepared statements to prevent SQL injection attacks
* Applying output encoding to prevent cross-site scripting (XSS) attacks
* Regularly updating and patching the application framework and libraries to address known vulnerabilities

Logging, Monitoring, and Auditing Effective logging, monitoring, and auditing mechanisms help detect and respond to security incidents promptly. Consider the following best practices:

* Implementing comprehensive logging of security-relevant events, including API requests, authentication attempts, and error conditions
* Utilizing centralized log management and analysis tools to detect anomalies and potential security breaches
* Implementing real-time monitoring and alerting systems to promptly respond to security incidents
* Regularly reviewing and analysing logs for identifying patterns, trends, and potential vulnerabilities
* Conducting periodic security audits and penetration tests to identify and remediate vulnerabilities proactively

Secure API Design A well-designed API is essential for maintaining the security and integrity of Foodify's services. Consider the following best practices for secure API design:

* Follow the principle of least privilege
* Use proper endpoint naming conventions
* Implement versioning
* Implement secure defaults
* Implement input validation at the API layer
* Implement proper error handling and messaging

Secure Data Storage Properly securing user data in storage is crucial for maintaining user trust and complying with data protection regulations. Consider the following measures for secure data storage in Foodify:

* Encryption at rest
* Strong access controls
* Regular backups
* Secure data disposal

Secure Third-Party Integrations Foodify may integrate with third-party services, such as payment gateways or mapping services. Consider the following practices to ensure the security of these integrations:

* Evaluate third-party security
* Implement secure communication
* Restrict permissions
* Monitor and review third-party security

Incident Response and Incident Management Despite preventive measures, security incidents may still occur. Establishing an incident response and management process helps Foodify respond effectively to security breaches. Consider the following steps:

* Incident response plan
* Incident detection and reporting
* Incident containment and mitigation
* Forensic investigation
* Communication and disclosure
* Lessons learned and improvement

Staff Training and Awareness A crucial aspect of maintaining a secure environment is ensuring that the Foodify team is knowledgeable about security best practices and aware of potential risks. Consider the following measures:

* Security training
* Security policies and guidelines
* Awareness campaigns
* Ongoing education

Compliance and Data Protection Compliance with relevant data protection regulations is essential for Foodify to handle user data responsibly and maintain user trust. Consider the following aspects:

* Data protection regulations
* Privacy policy
* User consent and transparency
* Data subject rights
* Data breach notifications

Regular Security Assessments and Updates Security is an ongoing process, and regular assessments and updates are crucial for maintaining a strong security posture. Consider the following practices:

* Regular security assessments
* Patch management
* Security information sharing
* Security risk management

Secure Mobile App Development Foodify should ensure that the mobile app development process follows secure coding practices. Consider the following measures:

* Secure coding guidelines: Establish guidelines for secure coding practices, including input validation, secure storage, and secure communication.
* Code review and testing: Implement code review processes and conduct thorough testing to identify and address security vulnerabilities in the mobile app.
* Secure app distribution: Use secure channels and reputable app stores for distributing the mobile app to users, reducing the risk of tampering or distribution of malicious versions.

Secure Data Transmission In addition to implementing TLS for communication between the app and backend servers, consider other measures to secure data transmission:

* Encryption for sensitive data: Encrypt sensitive data, such as user credentials or payment information, both at rest and during transmission.
* Secure data transmission protocols: Use secure protocols like HTTPS or secure WebSocket for transmitting sensitive data between the app and backend servers.
* Certificate pinning: Implement certificate pinning to ensure the app only communicates with trusted servers and to mitigate the risk of man-in-the-middle attacks.

Security Incident Response Testing Regular testing of Foodify's security incident response plan is crucial to ensure its effectiveness in real-world scenarios. Consider the following measures:

* Tabletop exercises: Conduct simulated exercises to test the incident response plan, involving relevant stakeholders and simulating different security incidents.
* Red teaming: Engage external security experts to perform controlled simulated attacks and assess Foodify's incident response capabilities.
* Lessons learned and improvement: Continuously evaluate the effectiveness of the incident response plan and make necessary improvements based on the outcomes of testing and real incidents.

Secure DevOps Practices Implementing secure DevOps practices can help ensure that security measures are integrated throughout the development and deployment processes. Consider the following practices:

* Security automation: Integrate security testing, code analysis, and vulnerability scanning tools into the continuous integration and deployment pipelines.
* Infrastructure as code security: Apply secure configuration practices to infrastructure as code (IaC) templates to minimize security misconfigurations.
* Secure containerization: Implement container security measures, such as image vulnerability scanning and container runtime security controls, to protect against container-based attacks.

Security Incident Communication Establish clear communication protocols to ensure transparent and effective communication with users and stakeholders in the event of a security incident. Consider the following aspects:

* Incident notification: Define processes and channels for notifying affected users about security incidents, including clear and concise messaging.
* Customer support: Ensure that customer support teams are trained to handle security-related inquiries and provide appropriate guidance to affected users.
* Public relations: Coordinate with public relations teams to manage external communication and maintain Foodify's reputation during security incidents.

Vendor Risk Management Foodify may rely on third-party vendors for assorted services. Implement vendor risk management practices to ensure the security of these relationships:

* Vendor assessment: Conduct thorough assessments of potential vendors' security practices, including their security controls and incident response capabilities.
* Contractual agreements: Define security requirements in vendor contracts, including data protection measures, incident reporting, and liability.
* Ongoing monitoring: Regularly assess and monitor vendors' security posture, including their adherence to security requirements and any reported security incidents.

User Security Education Promote user security awareness and education to help users protect their personal information and use the Foodify app securely. Consider the following initiatives:

* In-app security tips: Provide users with tips and best practices for maintaining strong passwords, recognizing phishing attempts, and protecting their devices.
* Privacy settings: Empower users to manage their privacy settings within the app, such as controlling the visibility of their personal information or order history.
* Security resources: Offer accessible resources, such as FAQs or knowledge bases, that address common security concerns and provide guidance to users.

Conclusion By implementing the recommended practices for secure communication, API integration, and overall security measures, Foodify can establish a robust and trustworthy food delivery app that prioritizes the confidentiality, integrity, and availability of user data. Through the adoption of TLS, strong authentication and authorization mechanisms, input validation and sanitization, logging and monitoring practices, and regular security assessments, Foodify can mitigate security risks, protect against attacks, and ensure a secure user experience. Ongoing vigilance, education, and compliance with data protection regulations will help Foodify stay ahead of emerging threats and maintain the trust of its users.

**SECURITY PAYMENT PLAN**

A security payment plan in a food app refers to the comprehensive strategy and measures put in place to ensure the secure processing and handling of payment transactions within the application. It outlines the specific steps and protocols that the app follows to protect sensitive financial information and prevent unauthorized access or fraudulent activities.

Most time the security payment plan includes some of the most important tools/parts that must be considered for the success of the app security payment plan. They include the following important things for our foody app outline below;

* A secure payment gateway is a service or platform that facilitates the secure processing of payment transactions between a customer and a merchant (business) during an online purchase or transaction. It acts as an intermediary between the customer, the merchant, and the financial institutions involved in the payment process (such as banks and credit card networks). The primary function of a secure payment gateway is to securely transmit and process payment data while protecting sensitive information.

* Encryption: Utilize strong encryption techniques to protect sensitive data during transmission. This involves implementing SSL/TLS protocols to establish a secure connection between the user's device and the server, ensuring that the data exchanged is encrypted and cannot be intercepted by malicious parties.

* Tokenization: Implement tokenization to replace sensitive payment data, such as credit card numbers, with randomly generated tokens. This ensures that even if the data is accessed, it holds no meaningful information and cannot be used for fraudulent purposes.

* Compliance with Payment Card Industry Data Security Standard (PCI DSS): Ensure that your app adheres to the requirements set forth by the PCI DSS. This includes maintaining a secure network infrastructure, regularly updating security systems, and undergoing periodic audits to validate compliance.

* Two-Factor Authentication (2FA): Implement an additional layer of security by requiring users to provide a second form of authentication, such as a unique verification code sent via SMS or email, in addition to their password. This helps prevent unauthorized access to user accounts.

* Fraud Detection and Prevention: Incorporate fraud detection mechanisms within the payment system to identify and prevent fraudulent activities. This can involve real-time monitoring of transactions, analysis of user behaviour patterns, and integration with fraud detection services or algorithms.

* Regular Security Audits: Conduct regular security audits and assessments to identify vulnerabilities and address them promptly. This includes performing penetration testing to simulate potential attacks and uncover any weaknesses in the payment system.

* User Education: Educate users about best practices for secure payment transactions, such as avoiding public Wi-Fi networks for financial transactions, regularly updating their app, and being cautious of phishing attempts. Provide clear instructions and resources to help users protect their payment information.

* Data Protection and Privacy: Establish policies and procedures to safeguard user payment data. This includes implementing access controls, encryption of data at rest, and secure storage practices. Comply with relevant data protection and privacy regulations, such as the GDPR or local data protection laws.

* Incident Response Plan: Develop a comprehensive incident response plan to address potential security breaches or data compromises. Define the steps to be taken in the event of an incident, including communication with affected users, containment of the breach, investigation, and remediation.

**SECURITY TESTING AND INCIDENT RESPONSE**

Security testing and incident response are critical components of any IT project to ensure the protection of sensitive information, maintain the integrity of systems, and promptly address security incidents. In this comprehensive explanation, we will explore the significance of security testing and incident response in IT projects, their key methodologies, and best practices, and how they contribute to overall project success.

* Security Testing in IT Projects:
* Security testing involves assessing the security posture of systems, networks, and applications within an IT project to identify vulnerabilities, weaknesses, and potential risks. It aims to proactively evaluate and enhance the security measures in place. Here are some essential aspects of security testing in IT projects:

* Vulnerability Assessment: Conducting vulnerability assessments helps identify and prioritize potential vulnerabilities within the project's infrastructure. This assessment involves using automated tools and manual techniques to scan for known vulnerabilities in software, networks, and systems. The results of a vulnerability assessment guide the implementation of necessary patches and security updates.

* Penetration Testing: Penetration testing, often referred to as ethical hacking, involves simulating real-world attacks to identify vulnerabilities that could be exploited by malicious actors. Skilled penetration testers attempt to gain unauthorized access to systems and applications, exposing weaknesses that need to be addressed. Penetration testing can be performed at various stages of the project, such as during development, prior to deployment, or periodically as part of ongoing security measures.

* Security Code Review: Security code review involves reviewing the source code of an application to identify security flaws. This process helps identify insecure coding practices, such as input validation vulnerabilities, insecure storage of sensitive data, or potential backdoors. Conducting security code reviews during the development phase enables early detection and remediation of security issues.

* Security Auditing: Security auditing evaluates the project's adherence to security policies, standards, and best practices. It involves assessing the project's compliance with relevant regulations, industry standards, and internal security guidelines. Security auditing encompasses reviewing access controls, encryption practices, logging mechanisms, and physical security measures. Regular security audits help identify gaps in security controls and ensure continuous improvement in security measures.

* Incident Response in IT Projects:
* Incident response is a systematic approach to address and manage security incidents or breaches within an IT project. It focuses on minimizing the impact of security incidents, containing the situation, and restoring normal operations. Here are the key aspects of incident response in IT projects:

* Incident Response Planning: Developing a comprehensive incident response plan specific to the IT project is crucial. This plan outlines the roles, responsibilities, and procedures to be followed in the event of a security incident. It includes defining the incident response team, establishing communication channels, defining incident severity levels, and specifying escalation procedures. The incident response plan should be regularly reviewed, updated, and communicated to all relevant stakeholders.

* Incident Identification and Reporting: Prompt identification and reporting of security incidents are vital for effective incident response. This involves implementing robust monitoring systems and security controls to detect and alert on potential security breaches. Incident identification includes analysing system logs, intrusion detection systems, and security information and event management (SIEM) solutions. Once identified, incidents should be reported following predefined procedures, enabling timely and appropriate responses.

* Incident Containment and Mitigation: Once a security incident is detected, immediate containment measures should be implemented to prevent further damage. This may involve isolating affected systems, shutting down compromised accounts, or blocking network access. Mitigation strategies should be applied to minimize the impact of the incident. These strategies may include patching vulnerabilities, updating access controls, or implementing temporary workarounds until a permanent solution is in place.

* Incident Investigation and Analysis: After containing the incident, a thorough investigation and analysis are conducted to determine the root

**Data Encryption and Storage**

Data encryption and secure storage are essential components of a comprehensive security plan in IT. They help protect sensitive information from unauthorized access, data breaches, and ensure confidentiality, integrity, and availability of the data.

In the context of Foodify, data encryption and secure storage play a critical role in protecting sensitive information related to customers, orders, payment details, and other confidential data. Here's how data encryption and secure storage can be incorporated into the security plan for our own Foodify:

**Customer Data Protection:**

* Encrypt personal identifiable information (PII) such as names, addresses, contact numbers, and email addresses of customers. This ensures that even if the data is accessed without authorization, it remains unreadable.
* Utilize strong encryption algorithms and secure key management practices to protect customer data both at rest and in transit.

**Payment Data Security:**

* Encrypt payment information, including credit card details, bank account numbers, and other sensitive financial data.
* Comply with industry-standard security protocols such as Payment Card Industry Data Security Standard (PCI DSS) to ensure secure handling of payment data.
* Implement tokenization or secure third-party payment gateways to minimize the storage of sensitive payment data within the application.

**Secure Order Information:**

* Encrypt order details, including items, quantities, delivery addresses, and any special instructions provided by customers.
* Apply encryption to ensure the confidentiality of order information, especially if it contains proprietary recipes, unique food formulations, or trade secrets.

**Access Controls and User Authentication:**

* Implement robust access controls and user authentication mechanisms to restrict access to sensitive data within the food application.
* Utilize strong passwords, multi-factor authentication (MFA), and role-based access controls to ensure authorized access to data.
* Regularly review and update user access permissions to align with the principle of least privilege.

**Secure Storage Practices:**

* Employ secure storage mechanisms such as encryption at the file system or database level.
* Implement appropriate access controls to limit access to stored data only to authorized personnel.
* Regularly backup data and securely store backups in offsite locations to ensure data availability and disaster recovery.

**Compliance and Regulations:**

* Familiarize yourself with relevant data protection regulations specific to the food industry, such as General Data Protection Regulation (GDPR) or data privacy laws in your region.
* Ensure your security plan adheres to the specific requirements outlined in these regulations.
* Regular Security Audits and Updates:
* Conduct regular security audits to identify vulnerabilities and ensure the effectiveness of encryption and secure storage measures.
* Stay up to date with emerging security threats, industry best practices, and advancements in encryption technologies.
* Promptly apply security patches and updates to maintain the security of the food application.

Remember that data encryption and secure storage are just a part of a comprehensive security plan. Other security measures, such as access controls, intrusion detection systems, and security audits, should be implemented in conjunction with encryption and secure storage to create a robust security framework.

Lilian kamanzi ID-57094   
Security payment plan  
  
  
Tanatswa Alexander Dzumbunu ID-56874  
Secure Communication and API Integration for Foodify  
  
  
Tapiwa Masoka ID-56335  
Data Encryption and Storage  
  
  
Arnold Mukucha ID-55322  
Security Testing and Incident Response  
  
Joan Nansamba Sebalamu   
ID-57153  
Risk and threat assessment