ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

SOFTWARE ENGINEERING

MOBILE APPLICATION DESIGN AND DEVELOPMENT

ASSIGNMENT

TITLE: FOOD DELIVERY APPLICATION

SECTION: 3

	GROUP MEMBER	**ID.NO**
1.	Milki Alemu	UGR/25517/14
2.	Yared Nigatu	UGR/26459/14
3.	Yitbarek tesfaye	UGR/25553/14
4.	Etsubdink aberham	UGR/26402/14
5.	Abraham yitbarek	UGR/26544/14
6.	Mohammed kasim	UGR/25433/14

1.Project Overview

Purpose

The mobile application aims to facilitate a seamless food delivery service. It allows users to browse various restaurants, view their menus, add items to a cart, and place orders. The application is designed to enhance the food ordering experience by providing an intuitive and user-friendly interface, making it easy for users to find and order their favorite meals.

Scope

The application includes several key functionalities:

1. **Restaurant Browsing**: Users can view a list of available restaurants, each with details such as name, location, delivery fee, rating, and image.
2. **Menu Viewing**: Users can select a restaurant to view its menu, including detailed information about each food item.
3. **Cart Management**: Users can add food items to a cart, view the cart contents, adjust quantities, and proceed to checkout.
4. **Favorite Items**: Users can mark food items as favorites for quick access in the future.

Problem Domain

The primary problem domain for this application is the food delivery industry. In a fast-paced world where convenience is paramount, food delivery services have become essential. However, users often face challenges such as difficulty in finding restaurants, cumbersome ordering processes, and managing their orders efficiently. This application aims to address these issues by providing a streamlined, user-friendly platform for food ordering.

Target Audience

The target audience for this application includes:

1. **General Consumers**: Individuals who frequently order food online for convenience.
2. **Busy Professionals**: People who have limited time to cook and prefer ordering food.
3. **Students**: Individuals living in dorms or apartments who might not have access to full kitchen facilities.

4. **Families**: Households looking for a convenient way to order meals, especially during busy days.
5. **Tech-Savvy Users**: Individuals who prefer using mobile apps for most of their daily tasks, including food ordering.

Conclusion

This mobile application aims to enhance the food ordering experience by offering a comprehensive and intuitive platform. By addressing common pain points in the food delivery process and catering to a wide audience, the application strives to become a go-to solution for anyone looking to order food quickly and efficiently.

3.Specific Requirements for the Mobile Application

Functional Requirements

1. **User Registration and Authentication**
    - Users should be able to create an account, log in, and log out securely.
2. **Restaurant Listing and Details**
    - Display a list of available restaurants with basic details (name, location, rating, delivery fee, and image).
    - Allow users to filter and sort restaurants by various criteria (e.g., distance, rating, delivery fee).
3. **Menu Viewing**
    - Show a detailed menu for each restaurant, including food item names, descriptions, prices, and images.
    - Allow users to view additional details about each food item.
4. **Cart Management**
    - Enable users to add items to the cart from the restaurant menu.
    - Allow users to view and edit the contents of their cart (adjust quantities, remove items).
    - Display the total cost of items in the cart.
5. **Order Placement**
    - Provide a straightforward process for users to place an order from their cart.

Non-Functional Requirements

1. **Usability**
    - Ensure the app is user-friendly with an intuitive and consistent interface.
    - Provide a responsive design that works well on various screen sizes and devices.
2. **Performance**
    - Optimize the app for fast loading times and smooth interactions.
    - Ensure the app handles a high volume of users and data efficiently.
3. **Security**
    - Implement secure user authentication and data encryption.
    - Protect sensitive user information, especially during transactions.
4. **Reliability**
    - Ensure the app is stable and minimizes crashes and bugs.
    - Provide reliable order tracking and notifications.
5. **Scalability**
    - Design the app to accommodate future growth, including more restaurants and users.

Influence on Design and Functionality

User Interface (UI) Design

- **Clean and Intuitive Layout**: A simple, clean layout ensures users can easily navigate through the app. The home page prominently features restaurant listings and search functionality.
- **Consistent Design**: Consistent use of fonts, colors, and buttons across the app to enhance user experience.
- **Visual Hierarchy**: Important information like restaurant ratings, delivery fees, and total cart amount are prominently displayed.

User Experience (UX) Design

- **Ease of Navigation**: Clear navigation options, including a bottom navigation bar for quick access to home, cart, and favorites.

Functionality

- **Cart and Order Management**: The ability to view, edit, and manage the cart ensures users have control over their orders before checkout.
- **Favorites**: Allowing users to save favorite items enhances user retention and satisfaction.
- **Settings Management**: Customizable settings for notifications, themes, and language cater to user preferences and improve overall experience
- **Search and Filtering**: Advanced search and filtering options help users find their preferred restaurants and dishes quickly

Conclusion

These requirements and their influence on the design and functionality of the application ensure that it meets user needs while providing a seamless and enjoyable food ordering experience. By focusing on both functional and non-functional requirements, the app aims to deliver a high-quality, reliable, and secure service to its users.

3.Design Principles and Concepts Applied in the Mobile Application

1. User-Centred Design

Principle: The entire design revolves around the needs, preferences, and behaviors of the users.

Concept Applied:

- **User Research**: Conducted to understand user expectations and pain points in ordering food.
- **Persona Development**: Profiles created to represent typical users, guiding design decisions.
- **Iterative Design**: Continuous feedback loops to refine the app based on user testing and analytics.

2. Consistency and Standards

Principle: Maintain consistency in UI elements, visual design, and interaction patterns throughout the application.

Concept Applied:

- **UI Elements**: Consistent use of buttons, icons, typography, and colors across screens.
- **Navigation Flow**: Predictable navigation paths (e.g., bottom navigation bar, drawer menu) for intuitive user experience.
- **Visual Design**: Unified design language with cohesive layouts and visual hierarchy for clarity.

3. Simplicity and Minimalism

Principle: Keep the interface simple and straightforward, minimizing cognitive load for users.

Concept Applied:

- **Clean UI**: Minimalistic design approach with uncluttered screens and focused content.
- **Intuitive Interactions**: Use of familiar gestures (e.g., tapping, swiping) and logical layout for ease of use.
- **Progressive Disclosure**: Display information progressively as users navigate deeper into menus or details.

4. Feedback and Response

Principle: Provide clear and timely feedback to users for their actions or system status.

Concept Applied:

- **Visual Feedback**: Immediate visual cues (e.g., animations, color changes) when users interact with buttons or menus.
- **Toast Messages**: Informative messages (e.g., item added to cart) using toast notifications.
- **Loading Indicators**: Visual indicators (e.g., spinners, progress bars) during data loading or processing.

5. Accessibility

Principle: Ensure the app is accessible to users of all abilities and demographics.

Concept Applied:

- **Text Contrast and Size**: Legible text with adequate contrast and scalable font sizes.

7. Security and Privacy

Principle: Safeguard user data and ensure secure transactions throughout the app.

Concept Applied:

- **Secure Authentication**: Robust authentication mechanisms (e.g., OAuth, biometric authentication) for user login.

Conclusion

By applying these design principles and concepts, the mobile application not only enhances usability and functionality but also provides a delightful user experience. Each principle addresses specific aspects of UI/UX design, navigation flow, and visual elements to ensure the app meets user expectations while aligning with business goals. This holistic approach fosters user engagement, retention, and satisfaction, making the application a valuable tool in the competitive food delivery market.

4.**Development Methodology: Agile**

For the development of the food delivery mobile application, we adopted the Agile methodology. Agile is a popular approach for software development that emphasizes flexibility, collaboration, and customer feedback. Here's a detailed description of the methodology and the justification for its choice:

Key Principles of Agile:

1. **Iterative Development:**
    - The project is divided into small iterations, typically 1-2 weeks long.
    - Each iteration involves planning, design, development, testing, and review.
    - This allows for continuous improvement and adjustment based on feedback.
2. **Customer Collaboration:**
    - Regular interactions with stakeholders and users to gather feedback.
    - Ensures that the product meets user needs and expectations.
3. **Responsive to Change:**
    - Agile welcomes changing requirements, even late in development.
    - This flexibility allows the project to adapt to evolving user needs and market trends.
4. **Continuous Improvement:**
    - After each iteration, the team reviews what went well and what could be improved.
    - This leads to constant refinement of both the process and the product.

Justification for Choosing Agile:

1. **User-Centric Approach:**
    - The application aims to meet the dynamic needs of end-users, such as easy food ordering, efficient navigation, and a smooth user experience. Agile's focus on regular feedback ensures the app evolves based on actual user requirements.
2. **Flexibility:**
    - In the fast-paced tech environment, requirements can change frequently. Agile's iterative nature allows the team to quickly adapt to these changes without significant disruption.
3. **Collaboration:**
    - The project involves cross-functional teams including developers, designers, and stakeholders. Agile promotes collaboration and communication, ensuring all parties are aligned and working towards common goals.
4. **Risk Management:**

- By delivering the product in small, manageable increments, Agile helps identify and address risks early. This reduces the chances of major issues late in the project.

5. **Quality Assurance:**
    - Continuous testing and integration in Agile ensure that the product maintains high quality throughout the development process. Bugs and issues are identified and fixed promptly.
6. **Visibility and Transparency:**
    - Agile provides regular progress updates through daily stand-ups and iteration reviews. This transparency keeps stakeholders informed and engaged, leading to better decision-making.

Application of Agile in the Project:

1. **Iteration Planning:**
    - Each iteration began with a planning session where the team discussed goals, tasks, and priorities.
    - User stories were defined and added to the backlog.
2. **Daily Stand-Ups:**
    - Brief daily meetings were held to discuss progress, obstacles, and next steps.
    - These stand-ups ensured that everyone was on the same page and any blockers were addressed swiftly.
3. **Design and Development:**
    - The development team focused on implementing features such as user registration, restaurant listing, menu display, and cart management in short cycles.
    - UI/UX designers worked closely with developers to ensure a seamless and intuitive user interface.
4. **Testing:**
    - Each feature was rigorously tested within the iteration to ensure functionality and performance.
    - Bugs were tracked and resolved promptly to maintain product quality.
5. **Review and Retrospective:**

- At the end of each iteration, the team reviewed the completed work with stakeholders.
- Feedback was gathered and incorporated into the next iteration.
- Retrospectives helped the team reflect on the process and identify areas for improvement.

By adopting Agile, the project maintained a high degree of flexibility and responsiveness, ensuring that the final product was well-aligned with user needs and market demands. This methodology facilitated continuous improvement, stakeholder engagement, and efficient management of development tasks, contributing to the overall success of the project.

5.Overview of Technologies, Tools, and Frameworks Used in the Development of the Mobile Application

1. Programming Language: Dart

- **Description:** Dart is a client-optimized language for fast apps on any platform. It is developed by Google and is used to build mobile, desktop, server, and web applications.
- **Justification:** Dart is used with Flutter for its expressive and flexible syntax, which enables quick development and a smooth learning curve for developers familiar with other object-oriented languages.

2. Framework: Flutter

- **Description:** Flutter is an open-source UI software development toolkit created by Google. It is used to develop cross-platform applications for Android, iOS, Linux, Mac, Windows, and the web from a single codebase.
- **Justification:** Flutter was chosen for its ability to provide a unified experience across different platforms with a single codebase. It offers a rich set of pre-designed widgets and high performance, which are crucial for building a responsive and visually appealing mobile application.

3. Integrated Development Environment (IDE): Visual Studio Code

- **Description:** Visual Studio Code (VS Code) is a free source-code editor made by Microsoft for Windows, Linux, and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring.
- **Justification:** VS Code is lightweight yet powerful, with a rich ecosystem of plugins, including those for Dart and Flutter. It enhances productivity through its user-friendly interface and robust debugging capabilities.

4. Version Control: Git and GitHub

- **Description:** Git is a distributed version control system for tracking changes in source code. GitHub is a code hosting platform for version control and collaboration.
- **Justification:** Git provides a reliable way to manage changes and collaborate with team members. GitHub offers a centralized platform for version control, issue tracking, and continuous integration.

5. State Management: Provider

- **Description:** Provider is a Flutter library for state management. It is a wrapper around InheritedWidget to make it easier to use and more reusable.
- **Justification:** Provider was chosen for its simplicity and efficiency in managing state across the application. It enables easy sharing of state and logic between different parts of the app, ensuring a reactive and responsive UI.

6. Package Manager: Pub

- **Description:** Pub is the package manager for the Dart programming language, used for managing dependencies and packages.
- **Justification:** Pub makes it easy to include external libraries and packages in the project, enhancing functionality and reducing development time by leveraging pre-built solutions.

8. User Interface Components: Material Design

- **Description:** Material Design is a design language developed by Google, used in Flutter through the Material widgets.

- **Justification:** Material Design provides a comprehensive set of design guidelines and components, ensuring a consistent and intuitive user experience. Its integration in Flutter allows for the creation of visually appealing and responsive interfaces.

9. Backend and Cloud Services: Firebase

- **Description:** Firebase is a platform developed by Google for creating mobile and web applications. It provides services such as authentication, real-time database, cloud storage, and cloud functions.
- **Justification:** Firebase was chosen for its comprehensive suite of backend services, which simplifies the development of scalable and secure applications. Its real-time database and cloud storage facilitate efficient data management and synchronization across devices.

10. Design and Prototyping: Figma

- **Description:** Figma is a cloud-based design tool used for interface design and prototyping. It allows real-time collaboration among designers and stakeholders.
- **Justification:** Figma was used to create high-fidelity UI/UX designs and prototypes, ensuring a consistent and user-friendly interface. Its collaborative features allow for seamless communication and iteration between designers and developers.

6.Key Features and Functionality Implemented in the Food Delivery Mobile Application

The food delivery mobile application has been designed to provide a seamless and user-friendly experience for users looking to order food from their favorite restaurants. Below are the key features and functionalities implemented in the application:

1. User Authentication

- **Feature:** Users can sign up, log in, and log out of the application.
- **Functionality:**
    - Firebase Authentication is used to manage user authentication.

    - Secure handling of user credentials.
    - Password recovery functionality.

2. Restaurant Listing

- **Feature:** Users can view a list of available restaurants.
- **Functionality:**
    - Display of restaurant names, images, and basic details.
    - Data fetched from a backend service or a local database.

3. Restaurant Details Page

- **Feature:** Users can view detailed information about a selected restaurant.
- **Functionality:**
    - Display of restaurant's menu items with images, names, and prices.
    - Dynamic fetching and displaying of restaurant-specific food items.
    - Integration of an "Add to Cart" button for each food item.

4. Menu and Food Items

- **Feature:** Users can browse the menu of a selected restaurant.
- **Functionality:**
    - Display of food items with details like name, description, price, and image.
    - User interface for adding items to the cart.

5. Shopping Cart

- **Feature:** Users can add food items to their cart and view their cart.
- **Functionality:**
    - Management of cart state using the Provider for state management.
    - Ability to add and remove items from the cart.
    - Display of cart summary with total cost calculation.

6. Order Placement

- **Feature:** Users can place an order for the items in their cart.
- **Functionality:**
    - Form to enter delivery details.

    - Integration with a backend service to handle order processing.
    - Confirmation message upon successful order placement.

Testing Strategies for Ensuring Functionality and Reliability

Ensuring the functionality and reliability of the food delivery mobile application requires a comprehensive testing strategy that covers various aspects of the application. Below are the key testing strategies employed:

1. Unit Testing

- **Purpose:** To test individual units or components of the application to ensure that each part functions correctly in isolation.
- **Approach:**
    - Write unit tests for each function, method, and class using the Flutter test package.
    - Focus on business logic, data models, and state management.
    - Mock dependencies to isolate the unit under test.
- **Tools Used:** Flutter Test,

2. Widget Testing

- **Purpose:** To test the UI components (widgets) to ensure they behave as expected.
- **Approach:**
    - Write widget tests to verify the behavior and appearance of widgets.
    - Simulate user interactions and validate the widget's response.
    - Test layout, rendering, and navigation between widgets.
- **Tools Used:** Flutter Test, Flutter Driver.

3. Integration Testing

- **Purpose:** To test the interactions between different parts of the application and ensure they work together as expected.
- **Approach:**
    - Write integration tests to cover end-to-end scenarios.
    - Test the flow of the application from launching to order placement.

    - Validate the integration of different modules like user authentication, cart management, and order processing.
- **Tools Used:** Flutter Driver.

4. Automated UI Testing

- **Purpose:** To automate the testing of the user interface to detect regressions and ensure a consistent user experience.
- **Approach:**
    - Use automated UI testing tools to simulate user actions and verify the UI response.
    - Test critical user flows such as login, browsing menus, adding items to the cart, and placing orders.
- **Tools Used:**Figma

5. Manual Testing

- **Purpose:** To perform exploratory testing and validate the user experience.
- **Approach:**
    - Conduct manual testing to identify usability issues and edge cases not covered by automated tests.
    - Perform device testing on different screen sizes and orientations to ensure responsive design.
    - Validate the overall look and feel of the application.
- **Tools Used:** Physical and virtual devices.

6. Performance Testing

- **Purpose:** To ensure the application performs well under various conditions.
- **Approach:**
    - Measure the application's startup time, memory usage, and CPU usage.
    - Test the application's responsiveness under load, such as handling multiple simultaneous users.
    - Identify and optimize performance bottlenecks.
- **Tools Used:** Flutter DevTools, Firebase Performance Monitoring.

7. Security Testing

- **Purpose:** To identify and mitigate security vulnerabilities in the application.
- **Approach:**
    - Perform penetration testing to identify common security issues like SQL injection, XSS, and authentication flaws.
    - Validate the secure storage and transmission of user data.
    - Ensure compliance with data protection regulations.
- **Tools Used:** OWASP ZAP, Firebase Security Rules.

8. Continuous Integration and Continuous Deployment (CI/CD)

- **Purpose:** To automate the build, test, and deployment process to ensure code quality and expedite release cycles.
- **Approach:**
    - Set up CI/CD pipelines to run automated tests on every code commit.
    - Ensure code quality checks and enforce coding standards.
    - Automate the deployment process to staging and production environments.
- **Tools Used:** GitHub Actions, CircleCI, Firebase App Distribution.

Potential Future Enhancements and Features

To keep the food delivery mobile application competitive and user-friendly, several enhancements and features could be integrated. These potential improvements aim to provide a better user experience, increase customer engagement, and streamline operations for restaurant partners.

1. Personalized Recommendations

- **Feature:** Implement a recommendation engine that suggests dishes based on users' past orders, preferences, and browsing history.
- **Benefit:** Enhances user experience by offering tailored suggestions, potentially increasing order frequency and customer satisfaction.

2. Real-Time Order Tracking

- **Feature:** Add real-time GPS tracking for deliveries, allowing customers to see the exact location of their orders in real-time.
- **Benefit:** Increases transparency and reduces customer anxiety by providing accurate delivery time estimates.

3. Loyalty and Rewards Program

- **Feature:** Implement a loyalty program where customers earn points for each order, which can be redeemed for discounts or free items.
- **Benefit:** Encourages repeat business and rewards loyal customers, fostering long-term customer relationships.

4. Advanced Search and Filters

- **Feature:** Enhance the search functionality with advanced filters (e.g., cuisine type, price range, dietary preferences, and distance).
- **Benefit:** Helps users quickly find the food that meets their specific needs and preferences.

5. Multiple Payment Options

- **Feature:** Integrate additional payment gateways, including Apple Pay, Google Wallet, and cryptocurrency options.
- **Benefit:** Provides users with flexible payment options, catering to a broader audience.

6. Scheduled Orders

- **Feature:** Allow customers to schedule orders in advance for delivery at a later time or date.
- **Benefit:** Offers convenience for users who want to plan their meals ahead, increasing customer satisfaction.