**Requirement Gathering and Analysis Phase**
**Technology Stack (Architecture & Stack)**

| | |
|---|---|
| Date | 29-June-24 |
| Team ID | SWTID1720075141 |
| Project Name | Food Mine – Order On The Go |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

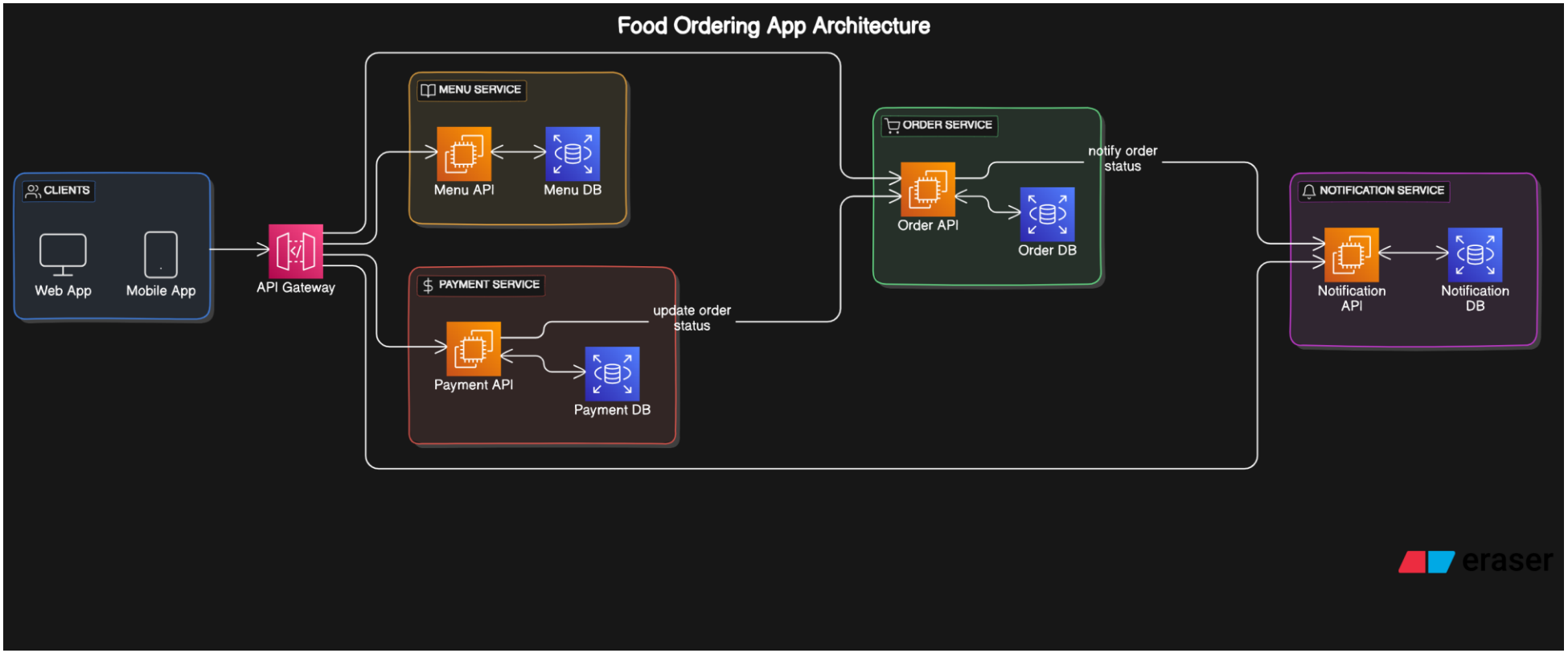

**Table-1 : Components & Technologies:**

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | Interactive web platform enabling users to explore menus, place orders, manage preferences, and track deliveries. | HTML, SCSS, JavaScript / React.js |
| 2. | API Gateway | Centralized gateway managing incoming requests from the frontend and routing them to respective backend services. | Cloud-based API Gateway service |
| 3. | Search Service | Backend system processing user queries to find nearby restaurants, filtering by cuisine, rating, and availability. | Node.js with MongoDB |

| | | | |
|---|---|---|---|
| 4. | Listing Service | Backend functionality for restaurant owners to list, update, and remove menu items with real-time synchronization. | Node.js with MongoDB |
| 5. | User Service | Service handling user account management, authentication, and personalized recommendations based on order history. | Node.js with MongoDB |
| 6. | Geolocation Service | Service leveraging GPS data to suggest local eateries, optimize delivery routes, and estimate delivery times accurately. | Google Maps Platform |
| 7. | Database | Central repository storing user profiles, restaurant data, menu items, orders, and transaction histories securely. | MongoDB |
| 8. | Messaging Service | Real-time chat solution facilitating direct communication between customers and restaurants for order inquiries. | Node.js |
| 9. | Cloud Storage | Scalable storage solution for hosting restaurant images, menu photos, and user profile pictures efficiently. | Cloudinary |
| 10. | Payment Gateway | Secure payment processing service allowing seamless transactions for food orders and subscription services. | Third-party payment gateway service |

**Table-2: Application Characteristics:**

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | Utilizes open-source frameworks to create dynamic user interfaces, robust backend support, and scalable data storage. | Prisma, Express.js, React.js |
| 2. | Security Implementations | Implements robust security measures to protect user data and manage access to food listings and user profiles. | bcrypt ,JSON web tokens(JWT), Middleware-Token Verification, Role Based Access |

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 3. | Scalable Architecture | Architecture that supports scalability for handling increasing user traffic and data volume efficiently. | MongoDB, RESTful API, Node.js |
| 4. | Availability | Ensures high availability of food ordering features, minimizing downtime and ensuring continuous service access. | Multiple Node.js instances |
| 5. | Performance | Focuses on delivering fast response times for food searches, menu details, and transactions to enhance user experience. | Prisma ORM, MongoDB, Node.js, Express |