RAS App – Requirements & Acceptance Criteria Document

Prepared by: group 17

Sprint: 1 – Requirement Analysis & Design

Project Type: mobile app (Restaurant Aggregator System)

Development Approach: Agile (Scrum) – 3-Week Sprint Plan

SDLC Phase: Requirements Analysis

# 1. Project Scope Summary

RAS (Restaurant Aggregator System) is a cross-platform mobile and web app designed to let users search, book, and review restaurants. The app will serve different user roles: guest users, registered users, premium clients, restaurant owners, and super-admins.

# 2. Functional Requirements

## Guest Users

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Description | Priority |
| FR-1 | Restaurant Discovery | View nearby restaurants on a map or list. | High |
| FR-2 | Guest Booking | Book meals without account registration. | High |
| FR-3 | Review Access | View restaurant reviews and photos. | Medium |

## Registered Users

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Description | Priority |
| FR-4 | Filtered Table Booking | Book tables using cuisine/rating/availability filters. | High |
| FR-5 | Personalized Feed | See personalized suggestions based on preferences. | Medium |
| FR-6 | Rating System | Rate meals and restaurants after orders. | Medium |

## Premium Users

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Description | Priority |
| FR-7 | Exclusive Deals | Receive early-bird offers and pre-order options. | Medium |
| FR-8 | Favorites & Sharing | Save and share restaurant wishlists. | Low |

## Restaurant Owners

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Description | Priority |
| FR-9 | Menu Management | Enable/disable menu items live. | High |
| FR-10 | Ad Campaigns | Run targeted ad promotions. | Medium |
| FR-11 | Analytics | Access insights on feedback & customer behavior. | Medium |

## Platform Admin (Super Admin)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Description | Priority |
| FR-12 | User Management | Manage users and moderate reviews. | High |
| FR-13 | Pricing Control | Adjust pricing for ad campaigns. | Medium |
| FR-14 | System Monitoring | Monitor system uptime and error reports. | High |

## Cross-Cutting Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Description | Priority |
| FR-15 | Secure Payments | Integrate secure multi-gateway payment processing. | High |
| FR-16 | Delivery Tracking | Provide real-time order delivery tracking. | Medium |
| FR-17 | Offline Mode | Allow cached access to limited content. | Low |

# 3. Sample User Stories with Acceptance Criteria

## User Story 1.1 – Guest Restaurant Discovery

As a guest user, I want to browse nearby restaurants so that I can find nearby options close to me.

Acceptance Criteria:

* - Home screen shows a map with clickable restaurant pins.
* - List view toggles available as alternative.
* - Users can filter by cuisine, price range, and opening hours.

## User Story 2.1 – Registered Table Booking

As a registered user, I want to filter and book tables easily in other that I can secure reservations based on my preferences.

Acceptance Criteria:

* - Location-aware restaurant list with availability.
* - Filtering options for cuisine type, open slots, and user rating.
* - Successful bookings generate a confirmation notification.

## User Story 5.3 – Super Admin System Monitoring

As a platform admin, I need to monitor system uptime and errors in such a way that I can ensure high platform availability.

Acceptance Criteria:

* - Dashboard showing live API error rates and system logs.
* - Email alerts sent daily or when uptime drops below SLA.
* - Admin panel displays server health and average response times.

# 4. Tools and Formats for Documentation

|  |  |
| --- | --- |
| Tool | Purpose |
| Google Docs | Collaborative requirement drafting |
| Trello/Jira | Agile backlog & sprint management |
| Figma | UI validation against documented requirements |
| Draw.io | Entity-Relationship (ER) and Use Case diagrams |
| GitHub Wiki | Living documentation & version control |

# 5. Next Steps

* - Review requirements with stakeholders and the team.
* - Identify missing edge cases or corner scenarios.
* - Link these requirements directly to backlog issues/tickets.
* - Track requirement changes in Jira using version control.