## Restosam Are you familiar with long waits and searching for a waiter, slow service and low quality? The system being developed should solve these issues.

It is necessary to provide the ability to book a table and order dishes for a certain time with prepayment by bank card or through a payment system. By the time you arrive at the restaurant, everything is ready.

You can book according to the hall plan, which displays the layout, location of tables, current

and average occupancy on a given day of the week. When choosing dishes for pre-order,

you can view the multimedia menu (of varying detail depending on the speed of the line), get

advice on choosing a dish.

In addition, it is necessary to provide the ability to select a restaurant by the set of dishes that you would like to have for dinner/lunch, price, distance and other parameters.

In the restaurant itself, you can call a waiter via phone, as well as pay the bill.

The system tracks and saves the client's actions in a log and, based on them, creates a preference map and suggests where to go and what to eat tomorrow/during the week.

## 

## Set of Deliverables

| **Deliverable** | **Description** | **Priority** |
| --- | --- | --- |
| Mobile Application | Native iOS/Android app for customers | Must Have |
| Web Application | Responsive web interface for customers | Must Have |
| Restaurant Admin Dashboard | Web-based management system for restaurants | Must Have |
| Payment Processing System | Integration with multiple payment providers | Must Have |
| Interactive Restaurant Maps | Visual table selection and booking system | Must Have |
| Menu Management System | Dynamic menu updates and item availability | Must Have |
| Waiter Notification System | Real-time order and service request alerts | Must Have |
| User Account Management | Registration, authentication, and profile management | Must Have |
| Order Management System | End-to-end order processing and tracking | Must Have |
| Recommendation Engine | AI-powered food suggestions based on user preferences | Should Have |
| Loyalty Program Module | Referral system and customer rewards | Should Have |
| Multi-language Support | Russian and English localization | Should Have |
| Accessibility Features | Support for color-blind and visually impaired users | Could Have |
| Analytics Dashboard | Business intelligence and reporting tools | Could Have |

## 

## Functional Requirements

| **ID** | **Requirement** | **Description** | **Priority** |
| --- | --- | --- | --- |
| FR-001 | Food Pre-ordering | Users must be able to pre-order food 1-24 hours in advance | Must Have |
| FR-002 | Table Booking | Users must be able to book specific tables using interactive restaurant maps | Must Have |
| FR-003 | Payment Processing | System must process payments via SBP, Mir, Visa, and other bank cards | Must Have |
| FR-004 | Restaurant Search | Users must be able to search restaurants by cuisine, price, location, and rating | Must Have |
| FR-005 | Interactive Restaurant Map | Visual representation of restaurant layout with table availability | Must Have |
| FR-006 | Waiter Calling | In-restaurant waiter request system with notifications | Must Have |
| FR-007 | Menu Management | Restaurant administrators must be able to update menu items and availability | Must Have |
| FR-008 | Order Reassignment | System must automatically reassign cancelled orders to prevent waste | Must Have |
| FR-009 | User Account Management | Support for email/password and social media login (Yandex, Google) | Must Have |
| FR-010 | Food Recommendations | AI-powered suggestions based on user history and ratings | Should Have |
| FR-011 | Loyalty Program | Referral system with discounts for inviting friends | Should Have |
| FR-012 | Multi-device Support | Responsive design for mobile, tablet, and desktop | Should Have |
| FR-013 | Multimedia Menu | Rich media content with images and detailed descriptions | Should Have |
| FR-014 | Bill Payment | In-restaurant bill payment through the app | Should Have |
| FR-015 | Preference Mapping | User behavior tracking to create preference profiles | Could Have |
| FR-016 | Group Reservations | Support for multiple people booking and bill splitting | Could Have |
| FR-017 | Delivery Integration | Food delivery option for future expansion | Won't Have |
| FR-018 | B2B Restaurant Orders | Restaurants ordering from each other | Won't Have |

## 

## Non-Functional Requirements

| **ID** | **Requirement** | **Description** | **Priority** |
| --- | --- | --- | --- |
| NFR-001 | Performance | Support 150 concurrent users per restaurant during peak hours | Must Have |
| NFR-002 | Scalability | Scale from 1 restaurant at launch to 10 restaurants within several months | Must Have |
| NFR-003 | Multi-language | Support for Russian and English languages | Must Have |
| NFR-004 | Cross-platform | Native mobile apps for iOS and Android | Must Have |
| NFR-005 | Accessibility | Compliance for visually impaired users | Should Have |

## 

## Architectural Constraints

## Technology Constraints

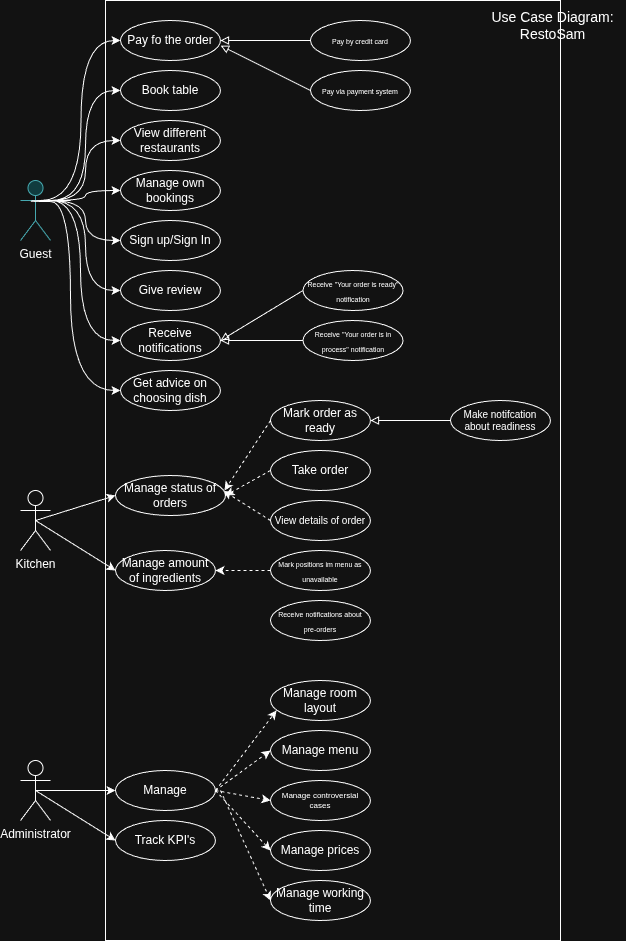
* **Payment Integration**: Must use payment systems available in Russia (SBP, Mir, Visa)
* **Third-party APIs**: No restrictions on API usage if accessible from Russia
* **Platform Support**: Web-based admin dashboard, Web and native mobile client applications
* **Real-time Processing**: Order and table booking updates must be processed in real-time
* **Concurrent Users**: Architecture must handle 150+ concurrent users per restaurant
* **Integration Requirements**: Admin dashboard for restaurant management and real-time menu updates

## Operational Constraints

* **Advance Booking Window**: 1 hour minimum to 24 hours maximum for pre-orders
* **Late Arrival Grace Period**: 15-minute buffer for customer arrivals
* **Menu Update Frequency**: Asynchronous, on-demand updates through admin panel
* **Order Cancellation**: Automatic reassignment system to minimize food waste

## User Experience Constraints

* **Accessibility Standards**: Support for color-blind users and text-to-speech functionality (optional)
* **Language Support**: Bilingual interface (Russian and English)
* **Authentication Options**: Multiple login methods including social media integration (optional)
* **Responsive Design**: Consistent experience across all device types and screen sizes



1. model of the software life cycle - Incremental
2. methodology of software development - Scrum
3. We chose incremental because it allows us to deliver versions of the product stage-by-stage with each stage being fully functional and then add more features during the next stage. We chose incremental but not waterfall since in the incremental model testing will happen after each stage, not after the fully developed project.
4. We chose Scrum because it's designed for flexible and adaptive development, which is crucial for restaurant systems where user requirements frequently change. Scrum allows fast delivery of value through short sprints (2-4 weeks), enabling the restaurant to start using basic booking functionality early. It provides a clear distribution of responsibilities between Product Owner, Scrum Master, and development team. The main idea is that Scrum focuses on delivering working software increments that provide immediate business value to restaurant operations.  
     
   XP we don’t choose because this is a very inconvenient development methodology and does not have the ability to parallelize development.

The suggested aspects of the project to cover in your argumentation:

1. Product/project type - Web & Mobile App
2. Project scale - Medium size
3. Team size 6-9 members (1 Team Lead&Scrum Master&Product Manager, 1 Web Frontend, 2 Backend, 1 Mobile, 1 DevOps, 1 SRE, 1 QA)
4. Team competence Vue (JavaScript) & Vue Native + Kotlin + Spring Boot + Golang + gRPC + PostgreSQL + MongoDB + Redis
5. Business factors (e.g. bureaucracy) - Innovative culture, low level of bureaucracy
6. Artifacts necessity - working stages of MVP, updated documentation, stable backend, stable frontend, stable mobile, stable test env
7. Technological factors - Multi-platform (iOS, Android, Web), Scalability and high availability, Security, CI/CD pipelines, cloud deployments,