[00:00:00]

Which guest segments are priority for you?

[00:00:04]

That should prioritize mass auditory who want to quickly eat at restaurant and don't wait to order dishes being inside of it.

[00:00:20]

So they can pre-order the food and then just eat it at the restaurant.

[00:00:27]

What processes should be automated in MVP?

[00:00:32]

Processes of ordering food, of payment processing, of calling the waiter.

[00:00:44]

What else?

[00:00:48]

Maybe for some more functionality we can automate delivery, but I guess it is not MVP already.

[00:01:01]

So in MVP it should be waitress, payment processing and food ordering.

[00:01:09]

Also there should be an interactive map to choose the restaurant to order from and interactive map of the restaurant itself to choose a table.

[00:01:23]

What additional sources of income are being considered?

[00:01:30]

The main source of income will be ordering food from our app.

[00:01:37]

Also it will be the conventional ordering of food in my restaurant.

[00:01:46]

But in general we can consider adding delivery or some other services.

[00:01:55]

For example tips for waiters and we can take some taxes from these tips.

[00:02:06]

What is your plan to reduce return or reworks due to order changes?

[00:02:14]

If someone cancels his order we can move this order to someone else.

[00:02:23]

Because we have an automated system we can automatically decide where to reassign this order to not waste the food.

[00:02:33]

For example if someone ordered a sandwich then change it to another type of sandwich and someone at this point ordered that sandwich.

[00:02:45]

We can move this order to him without any consequences and he will get the order even faster.

[00:02:56]

Do we need to integrate some loyalty programs?

[00:03:00]

I think yes because it is a good way to get new people in our program.

[00:03:11]

For example as a more simple approach we can make a referral system where you can invite a friend for example and get a discount for the order.

[00:03:27]

Ok, that's all for business part. Now we move to functionality plans.

[00:03:34]

What details should be included in the interactive plan of the restaurant?

[00:03:38]

For example table capacity, availability, location of your window, doors or else?

[00:03:44]

I think this map should be visual.

[00:03:50]

Have you booked a place in airplane?

[00:03:53]

Yes.

[00:03:54]

You got a scheme of airplane and can pick any seat that you like.

[00:04:00]

You can do something like show where this restaurant is located.

[00:04:09]

That person can pick it and order the food to this table.

[00:04:15]

By the way I think the restaurant should be informed that this table is booked.

[00:04:20]

So the system will automatically book this table inside the restaurant to prevent any race conditions.

[00:04:30]

Ok, so should the pre-order menu adapt automatically, for example if some items will be out of stock?

[00:04:42]

I think it is hard to optimize checking if some items are out of stock.

[00:04:49]

But you can make a system that will be controlled by administrator of this restaurant where he can add or remove some dishes or ingredients.

[00:05:04]

Ok, what about recommendations for food for our users? How they must be implemented?

[00:05:13]

I think you can pick some examples from your competitors.

[00:05:32]

For example pick some recommendation system and train it for your customers.

[00:05:42]

And they will get recommendations of food based on previous dishes and for example their ratings that they have put on these dishes.

[00:05:56]

Ok, how should customers be able to search and filter restaurants? For example by price, cuisine, location or something else?

[00:06:07]

I think you should consider all the main features.

[00:06:12]

For example take Yandex map and see what information is included for restaurant here.

[00:06:18]

It will be location, average menu price, average rating etc.

[00:06:27]

So you take these features, sort them in order of relevance and pick the most relevant ones which will be included in search.

[00:06:39]

Ok, should the apps or split payments group reservations or bill sharing in the future?

[00:06:45]

I think it will be a good feature for future development of the app.

[00:06:52]

But for MVP one person order is ok.

[00:06:59]

What is the expected flow for calling a waiter or paying bill inside the restaurant?

[00:07:05]

So what actions must be done?

[00:07:09]

Do you mean from user side?

[00:07:11]

Yes, like you want to order food and you proceed with the payment and waiter accepts this order and what his actions will be.

[00:07:28]

Ok, so the user uses the app to order food he is eating inside the restaurant.

[00:07:36]

So some waiter, like a taxist, gets a notification, he can get this client or don't get it, so some other waiter will get him.

[00:07:51]

The one who gets this client will just deliver the order and maybe ask for someone else if client wants.

[00:08:03]

Ok, that's good.

[00:08:07]

Should restaurants have an admin dashboard to manage menu items, table availability and other parameters?

[00:08:13]

Yes, as I stated before, full optimization is pretty hard in this case.

[00:08:22]

So it is necessary to have an admin dashboard where administrator of the restaurant can add or remove some items, manage current orders and other stuff.

[00:08:37]

Ok.

[00:08:40]

What about tech support for individual cases like in Yandex food?

[00:08:43]

Uh I think main support should be uh uh related to the uh restaurant itself but in case of technical uh problems uh there should be some contact uh to fix it.

[00:09:10]

Okay, and the last one about tech support. Do you prefer chatbot or do you prefer chatbot with AI or some real manager behind the tech support?

[00:09:25]

Uh, I think it's okay to have a chatbot if you want. Uh it will uh simplify the work of tech support because often chatbot is enough.

[00:00:00]

How frequently will menu updates need to be pushed to the system?

[00:00:07]

I don't think this will be scheduled, but it will be controlled via admin panel by administrators of the restaurants.

[00:00:18]

So it should work asynchronically and concurrently.

[00:00:26]

What about identification methods? What methods will be used for customer accounts?

[00:00:34]

I think it will depend on the way you will develop the application.

[00:00:43]

For example, if it will be a web browser, then it will be just an account with email or password, or integrating with other services.

[00:00:55]

For example, Yandex account login, Google account login, etc.

[00:01:02]

Which payment systems and banks must be used by the system to integrate with?

[00:01:08]

The system should work with SBP, it should work with major bank cards, for example, Mir, Visa, etc.

[00:01:25]

Maybe some bank payment systems, for example, Tepay, XBRPay, etc.

[00:01:33]

You talked about the devices. Should the interface adapt to different devices, for example, mobile, tablet or desktop?

[00:01:43]

Yes, because users will use our app in different circumstances.

[00:01:50]

They can pre-order the food from home, for example, from desktop, or order it being inside the restaurant using mobile phone or tablet.

[00:02:04]

So the interface should be adaptable.

[00:02:08]

What languages should the system support?

[00:02:12]

I think it will be enough to support Russian and English languages.

[00:02:18]

And what accessibility standards must the system meet?

[00:02:27]

I can't easily say, but you need to provide some brief research on this topic.

[00:02:36]

For example, you can use the style of interface that will be accessible by color-blind people.

[00:02:50]

And what about text-to-speech?

[00:02:54]

You can also integrate text-to-speech for blind people as well.

[00:03:01]

Okay.

[00:03:06]

So, what is the maximum number of concurrent users the system must handle during peak hours?

[00:03:15]

Because we will have several restaurants in our system, it is hard to predict exactly how many users it should handle.

[00:03:27]

But if we pick, for example, 150 users per restaurant at peak hours, it should scale like this.

[00:03:40]

And, for example, for 10 restaurants, it will be at least 1.5 thousand people.

[00:03:53]

Okay.

[00:03:54]

So, what about the amount of restaurants should the system support at launch?

[00:03:58]

And what is the goal to scale?

[00:04:03]

At launch, it should support at least 1 restaurant and scale up to 10 in several months.

[00:04:12]

Okay.

[00:04:13]

Are there any restrictions on third-party APIs?

[00:04:16]

No, you can use any APIs if they are available to pay from Russia.

[00:04:27]

Okay.

[00:04:28]

So, let's talk about timings for every stage of the order, including ordering in advance, cooking time, late coming of a client.

[00:04:42]

So, the client should have ability to order in advance.

[00:04:47]

For example, let us pick a boundary for one hour in advance.

[00:04:57]

So, minimum one hour.

[00:04:58]

Maximum one hour.

[00:05:02]

Or minimum.

[00:05:03]

So, can user have order two hours before his arrival?

[00:05:11]

Yes, it will be minimum one hour before.

[00:05:15]

So, the order will be properly scheduled and cooked at the restaurant.

[00:05:23]

And he can order the food, for example, a day before.

[00:05:33]

So, the maximum boundary is a day, 24 hours, but it should consider the changes in menu.

[00:05:43]

So, if some ingredients aren't available at some point, the user will be notified in advance to change his order.

[00:05:55]

Like, if you have used delivery in Pyaterochka, they notify you if they don't have this product.

[00:06:07]

So, yeah.

[00:06:09]

For cooking time, let us take one hour.

[00:06:14]

So, it will be at least one hour before.

[00:06:16]

And late coming of a client will be up to 15 minutes.

[00:06:24]

I think that's enough and it will be okay.

[00:06:28]

Okay, the last one.

[00:06:30]

Is it allowed for restaurants to order food from each other using the app?

[00:06:37]

For example, I think it's if some products are not available or changes happen.

[00:06:45]

So, it will be B2B.

[00:06:54]

Like, with transporting food from one restaurant to another.

[00:06:59]

I think it's a cool idea, but for scale, not for the MVP.

[00:07:06]

For the MVP, let's use B2C system with only client order.

[00:07:17]

Okay, thank you.

[00:07:18]

Thank you, goodbye.

## 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