Eats Easy

1. Introduction

EatsEasy is a system that connects clients and restaurants. It helps clients to find the place to eat in, fastly, to place the order and track it from anywhere, and to pay instantly. It helps restaurant’s staff to manage the queues of orders, to keep the menu updated and to serve clients the best way they can

2. Project Description

The abilities that EatsEasy provides to the clients:

- Flexibility: Using EatsEasy the client can search where and what he wants to eat, depending on his flavor and desires, at the time that he decides, making the shortest way possible, calculated using his current location
- Up-to-date: The client is always welcomed by businesses that serve at the selected time and presented with a real time stock-based updated menu
- Remote control: The client can book a table, place an order, even remotely, and get the time estimation for his order to arrive
- Fast assistance: While seeking help or service the client can call the staff by pressing a button and a waiter will be at the client's table in no time
- Time saving: When the client is done enjoying the delicious food, he can ask for the bill using the app or even pay through the EatsEasy app and be on his way to his next adventures

The abilities that EatsEasy provides to the businesses:

- Error-proof: The waiter sees the orders through the system as they are being placed, drastically cutting the time it takes to place an order and the amount of mistakes typically made in process
- Up-to-date: The shift manager can update the menu in real time based on stock availability. No Parmesan cheese in the fridge - no Caesar salad in the menu
- Take eat easy: Different ways to decide which waiters serve which tables, based on preferences and workload
- Serviceability: The client can press a button if he has any questions, and his waiter would receive a notice. No more begging for service and fishing the waiter’s attention
- Management: Plenty of useful real time graphs and data summaries for the business staff

3. Audience

Everyone who wants to be a part of modern era, where you don't need to wait for any interraction with human beigns in order to get some tasty food

4. System Structure

- Java Spring Boot based REST API server and microservices [backend]
    - Spring MVC (Spring’s contender of JSF)
    - Tomcat Java Servlet
    - Maven Dependency Manager
    - JavaBeans functionality

    - Data adapter for relational database (Spring Data JDBC or Hibernate for Postgresql)
    - Data adapter for non-relational database (Spring Data MongoDB for User Analytics and other metrics (Big unformed scale oriented Data))
    - XML configuration
- Android app with React Native for the client [frontend]
- Android app with React Native for the staff [frontend]
- React.js web UI for all who is interested to use this product in their business [landing pages, frontend] (optional)
- React.js web UI for system administrator [dashboard, frontend] (optional)

5. Submitters

- Ron Yanku (RonYanku)
- Felix Razikov (Guzzur)

6. Links

- Issues
- System Proposal (obsolete)
- System Design Document
- API Documentation
- Repositories
    - Documentation
    - Backend (Java Spring Boot based API)
    - Clients' Android app (React Native based)
    - Staff's Android app (React Native based)
    - Dashboard (React.js based) (TBD)
    - Product landing pages (React.js based) (TBD)