

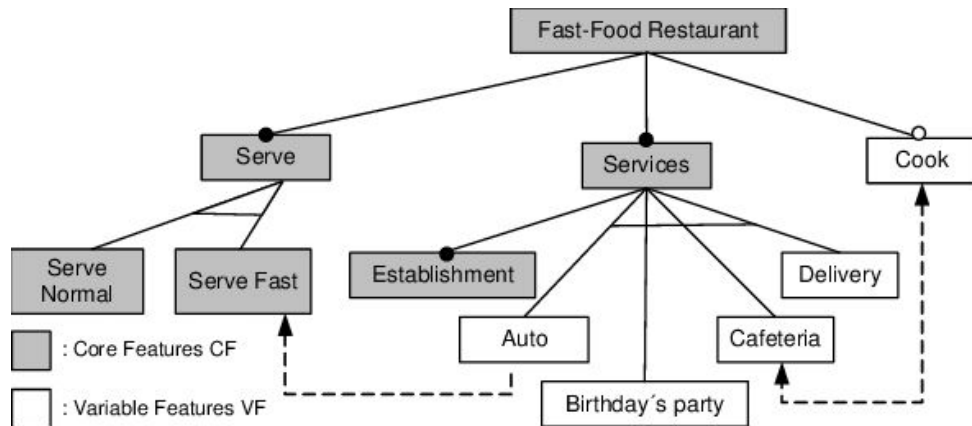
Restaurant Automation System



Vincent Stevens, Adam Oswald, Tyler Yarow, Aarya Shah, and
Amanvir Singh

Background

- Restaurants are fast-paced environments
 - Food isn't the only thing done
- Overhead
 - Payroll, ordering, scheduling, inventory...
- Restaurants like optimization



Motivation for the Idea

Goal: Use automation to reduce overhead and optimize workflow

- Centralized restaurant system
 - Payroll
 - Inventory estimations
 - Statistics/Data Collection
 - Food delivery ordering



Requirements

- Developed 20 Functional, 5 Non-Functional Requirements
- Major functionalities encompassed:
 - Training
 - Shift Scheduling
 - Payroll/ Payment Processing
 - Order Automation/ Online Ordering
 - Online Reservations



Functional				Non-Functional
System shall provide employers the ability to curate and post training modules	System shall allow managers to control priority of who gets first pick of hours	System shall allow online customers to view all menu items, add items to an order, and checkout with that order	System shall allow customers to make online reservations for tables	System should be scalable in the future, multiple restaurants should be able to utilize same system without overloading it
System shall allow employees to complete training modules	System shall allow employees to schedule working hours, pick up and trade shifts	System shall accept food orders natively and through outside applications (Grubhub, Doordash, etc)	System shall provide employees with an interface to edit reservation limitations	System should have an intuitive user interface... any information in the system should be accessible in at most two clicks/ taps
System shall allow employers to monitor employee progress through the modules	Shall allow employees to clock in and clock out when they arrive/ leave	System shall coordinate notification to employees of new online orders placed	System shall provide employees with a map of all tables, including those open, currently in use, and reserved	System should be portable... any platform should be able to run the software and communicate with users on different platforms
Shall allow managers to set shift availability for employees	Shall deduct taxes from employee paychecks before disbursing them to employee bank accounts	System will provide order updates, either directly to the customer or to the platform ordered through	System shall not let people reserve tables within 15 minutes of restaurant closing	System should process a payment and place an order in less than 3 seconds, for up to 50 simultaneous orders
Shall handle calculation of employee pay	System shall prevent employees from editing payroll information, and update managers if attempted	System shall follow required payment regulations, including proper encryption methods, protection standards, etc. (i.e. PCI-DSS)	System shall keep track of all employee-related costs and online-order generated profit for easy access by management	System should be built in such a way to allow easy future additions in case the system must grow later on

Technical Challenges

- Security
 - Cybersecurity Attacks
 - Akamai Protection (CDN Service)
 - Cloudflare Protection (CDN Service)
 - CrowdStrike Protection
 - Handling Data Securely
 - Encryption using Key Management Services (KMS) from AWS
- Scalability
 - Cloud capabilities by hosting on AWS
 - Version Control of Code

Technical Challenges - Continued

- Onshoring vs. Offshoring Data
 - Keeping data centers and servers at home or abroad
 - Privacy concerns associated with this
- Privacy
 - Concerns about how the data is handled
 - The ethics of collecting consumer data



Risks

- For cloud capabilities, we do not have direct access to data or server outage can occur
- We cannot store personal identifiable information

