Final Report – Team D

I. Introduction

Our project is a food ordering/delivery system that is based on a real-life application scenario. Customers can sign in; browse/search dishes; start a discussion; deposit money, etc. Employees in our system are also able to login and check orders. Managers have special privileges when signed on. A manager is taken to their own page where they can view current employees, see their current comments (whether negative or positive), and view their ranking. If a manager is to decide to create a new employee, they will be prompted to enter their information. Once entered, the main manager page will be updated with the new employee information.

This project was coded and hosted on Glitch (allows for multiple people to code at the same time and see changes live) by frontend: HTML/CSS/JS; backend: back4app (parse)(can make it work with spring boot) DB: back4app(MongoDB, PostgreSQL)

II. 1-1 Correspondence

1. Provide a GUI, not necessarily web-based, with pictures to show the descriptions of each dish and price; each registered customer/VIP has a password to login, when they log in, based on the history of their prior choices, different registered customer/VIP will have different top 3 listing dishes. For new customers or visitors, the top 3 most popular dishes and top 3 highest rated dishes are listed on the first page. DONE partially
2. A customer can choose to 1) pick up the dishes in person, or 2) by restaurant delivery. For case 1) s/he can only complain/compliment the chef. DONE
3. A customer can file complaints/compliments to chef of the food s/he purchased and deliver person who delivered the dish or other customers who didn't behave in the discussion forums. Delivery person can complain/compliment customers s/he delivered dishes; all complaints/compliments are handled by the manager. The complained person has the right to dispute the complaint, the manager made the final call to dismiss the complaint or let the warning stay and inform the impacted parties. Customers/delivery people whose complaints are decided to be without merit by the manager will receive one additional warning. DONE

4. Registered customers having 3 warnings are de-registered. VIPs having 2 warnings are put back to registered customers (with warnings cleared). The warnings should be displayed in the page when the customer logs in. DONE
5. Every customer should deposit some money to the system. If the price of the order is more expensive than the deposited money in the account, the order is rejected and the customer receives one warning automatically for being reckless. DONE
6. Customers who are kicked out of the system or choose to quit the system will be handled by the manager: clear the deposit and close the account. And kicked-out customer is on the blacklist of the restaurant: cannot register any more. DONE
7. The chef whose dishes received consistently low ratings (<2) or 3 complaints, will be demoted (less salary), a chef demoted twice is fired. Conversely, a chef whose dishes receive high ratings (>4) or 3 compliments, will receive a bonus. One compliment can be used to cancel one complaint. The delivery people are handled the same way. DONE partially
8. The delivery people will compete to deliver the order by bidding, the manager assigns the order from bidding results: the one with lowest delivery price is generally chosen; if the one with higher asking price is chosen, the manager should write a memo in the system as justifications. The delivery person who didn't deliver any in the past 5 orders will automatically receive one warning. DONE partially
9. Each team comes up with a creativity feature of the system to make it more exciting, e.g., smart-phone based system, voice-based features, or efficient route planning for delivery, which is worth 10% of overall score of the final project. Beautiful homepage

III. Contributions

- Matheus Figueroa Contribution: Worked on the manager page and implemented CSS styles for an interactive user experience when a mouse hovers over employee "tile" list. Employees automatically get generated from Read operations from the DB and displayed for manager to see. If a manager decides to create a new employee, details are prompted, once entered, the main manager page will create a new employee object and read again from the DB and populate the main page. If a employee has more than 5 warnings, the manager can fire that employee. Once fired, the object ID of that employee will be destroyed in the DB.
- Radmir Sataev Contribution: Built foundation for the website. Backend, DB, sign in page, amongus.js utilization of session storage across pages, asynchronous

functions, functions responsible for checkout, assigning drivers/orders, creating orders/users, becoming vip etc. (CRUD).

- Jacob Onbreyt Contribution: Made the Homepage, Chef page, navigation bar, most of the frontend – designed frontend framework, user interactivity and reactivity features, css, jquery (script.js). Implemented animations for user experience, designed logo
- Yevheniya Solomyana Contribution: Order page (order.html, order.js), add address, create menu (fooditem.js), search dishes in the menu, add dishes to cart items, calculate total item and amount, worked on the web design.
- Mohamed Kharma Contribution: Made the account page, made the bedding page for the delivery drivers, worked on css and js for these pages. Helped designing how the website looked before/after the user logged in to his account. Worked on the front end of the discussion page

IV. Git Repository

https://github.com/Zabegalovka/Zabegalovka

V. Remarks/Suggestions (Optional)