

Contributions (github ID)	Code (Up to Demo 1)	Video for Demo 1
Jason Chan(jc2375)	55%	22%
David Joseph(drj63)	27%	22%
Ashirvadh Bhupathi	0%	6%
Hojun Son	0%	6%
Josh Chopra	0%	0%
Krishna Patel(ksp115)	1%	7% + 9% (editing) = 16%
Juergen Benitez (jbz8)	17%	22%
Justin Davis	0%	0%
Pavan Kunigiri	0%	6%
Ryan Van Duren	0%	0%



****Code Contributions were calculated referencing contributions page of Github Repository**

****Note all our code was created from scratch and DID NOT copy any of 2018/2019 sample code**

- **Subproblem 1: Customer**

- Ordering Interface
 - Customers should be able to order food from the menu, and customize their orders. Jason (Base Menu Complete, Cart Integrations and Orders In Progress)
 - Account login system where they can get rewards/discounts for coming frequently. (guest system also will be implemented) David (Login/Register complete, Rewards in progress)
 - Takeout/Delivery/Dine in options Jason (Implementation needed)
- Food tracking interface(Simplified Version)
 - Able to see when their order is being started as well as cook time, and estimated time of completion (Implementation needed)
- Delivery tracking interface(Simplified Version)
 - Able to track when the order has left the restaurant, and estimated time of arrival. David (in progress)
- Data Analysis(recommended food items)
 - Suggested food items based off popular menu choices (implementation needed)
- Table Occupancy interface(Availability of Table)
 - Able to see how many tables are available with X amount of seats per table. Jason (In Progress)

- **Subproblem 2: Manager**

- Delivery/food tracking interface(Based off completion)
 - Manager can see when an order has completed and when a delivery has been completed (implementation needed)
 - Has admin options if he wants to see more details that the waiter and chef see.
 - Add or remove items from the menu Jason (Implementation needed)
 - Estimated food completion time Jason (Implementation needed)
 - Whether an order has been started (implementation needed)
- Table Occupancy interface(shared)
 - Able to see how many tables are available with X amount of seats per table. Jason (in progress)
 - (Admin): can update table status to any status. Jason (completed)
- Finance tracking(Full Version)

- Keeps track of all the finances throughout the day, as well as bookkeeping Juergen (Implementation needed)
- Able to see when a table has paid and how much their order was/receipts. Juergen (Implementation needed)
- Data Analysis (Full Version)
 - Data: Food ordered by time, throughout the day/ Food ordered by day of the week. Juergen (Implementation needed)
 - A bar graph will be produced.
 - Divided by time of day and day of the week
 - Data: Number of customers at various times throughout the day/week Juergen (Implementation needed)
- Clocking hours (Full Version) Juergen (Base model completed, further integration in Progress)
 - Able to see which employee has clocked in at what time and how long they have been working, as well as normal shift hours.
- **Subproblem 3: Employee (Waiter/Chef/Driver/Busboy)**
 - Table Occupancy interface(shared)
 - Able to see which tables are occupied/not, which tables are dirty and clean. (implementation needed)
 - Waiters: Can update table status as dirty (implementation needed)
 - Host: Can update table status as occupied (implementation needed)
 - Busboy: Can update table status as Clean (implementation needed)
 - Food tracking interface(Full version)
 - Able to track all the food orders(whether they have started cooking, almost done, complete, etc.) David (in progress)
 - Delivery tracking interface(Full Version)
 - Able to see whether the food has been delivered and track delivery David (in progress)
 - Finance tracking(Whether table has paid or not)
 - Tracks the bills, and whether a table has paid or not David (in progress)

*All functionalities will need to be used in combination with another. Examples of functionalities that will be in combination with another would be: food tracking, payments, data from customer orders, clock in hours, table availability, and surveys/data.

JAVA AND XML FILES

busboy_home: Krishna - David - Jason

busboy_tablestatus: Krishna - Jason

BusboySchedule: Juergen

Chef_home: Jason - David
ChefSchedules: Juergen
customer_home: Jason - David
Customer_reservation: Jason
Delievery_adress: Jason
Driver_availailbe_orders: David
Driver_current_order: David
Driver_home: David
DriverSchedules: Juergen
EmployeePayroll: Juergen
EmployeeSchedules: Juergen
ExpandableListView Adapter: Juergen
host_home: Jason - David
Login: Jason - David
MainActivity: Jason - David
ManageEmployeeOptions: Juergen
manager_home : Jason - David
Menu_home_canorder: Jason
Modify_account: David
order_type: Jason
Payment_type: Jason
Register: Jason - David
Waiter_home: Jason - David
WaiterSchedules: Juergen

Demo Documentation Contributions:

Presentation slides- Krishna, Pavan, Ashirvadh, Hojun, Josh
User Documentation- Juergen- 75%, Krishna- 25%
Brochure - David -100%
Technical Documentation - Juergen - 100%
Code - Jason, David, Juergen, Krishna
Unit Testing - Jason - 100%
Integration Testing - Jason -70% Krishna - 30%
Data Collection - David- 100%

Jason is in charge of coordination. The integration testing will be done by Jason, Juergen, and David.