

**Faculty Of Computing**

**SCSJ 1023 - Programming Technique II**

**PT2: Project Proposal**

**Members:**

Muhammad Zulkifli Bin Zainul (A14CS0173)

**Lecturer:**

Dr. Ruhaidah Binti Samsudin

**Synopsis:**

Koperasi UTM Berhad have decided to buy shares from Pizza Hut and Domino’s companies. Then, they want to customers can easily place order to either one of the pizza shop using only a system. So they hired us as a system development company to create a system to buy from Pizza Hut and Domino.

**Class Definitions:**

**Order**

**Customer**

-CompanyName : string

-TimeBuy : string

-hour: int

-minute: int

-name: string

-address: string

-HP: string

+Order()

+Order(string)

+Order(string,string)

+SelectCompany(string) : void

setCompanyName(string): string

getTimebuy(): string

getHour(): int

getMinute(): int

+Customer(string,string,string)

+getName(string) : void

+getAddress(string) : void

+getHP(string) : void

+displayMenu() : void

+setMenu(): void

+getFullMenu(): void

+transferData(Customer): void

**Menu**

- pCrust[50]: string

-pSpecial[50]: string

- pSize[50]: string

- pAmount[50]: int

- pPrice[50]: float

- Overall: float

- time: int

- CompanyName: string

hour: int

minute: int

\*pesanan: Order

\*id: Customer

pizzac: string

pizzas: string

size: string

amount: int

+Menu(Order)

+Menu(int,int)

+DisplayMenu(): void

+PrintData(): void

+setHour(int): void

+setMinute(int): void

+setCompanyName(string): void

+setPizzaCrust(string, int ): void

+setSpecialityPizza(string, int ): void

+setSize(string , int ): void

+setAmount(int, int ): void

+getTime(): void

**ClassPizzaHut**

**ClassDomino**

+CalculatePizza(int): virtual void

+DisplayOrderPizza():void

+CalculatePizza(int): virtual void

+DisplayOrder(int): void

**The System Flow:**

This system basically designed to take orders from customers either online or offline. Customers basically can choose to buy pizza from Domino or Pizza Hut. This system can differentiate between Pizza Hut and Domino. If customer choose Pizza Hut the system will display Pizza Hut menu otherwise it will display Domino menu. This system also estimate the time for the menu to be ready and also estimate the time for the pizza to be delivery. The system also can calculate the prices of overall including the discounts if necessary. In the end the system will display all outputs.

**Customer**

* **Variable**
* name: string => for customer name
* address: string => for customer address
* HP: string =>for user hand phone number
* **Function**
* Customer(string,string,string) => for the customer constructor
* getName(string) : void => to return customer name
* getAddress(string) : void => to return customer address
* getHP(string) : void => to return customer hand phone number

**Order**

* **Variable**

- Company’s Name: Pizza Hut and Domino

- Customer’s Name: Muhammad Zulkifli

* **Function**
* setMenu: Choose our own menu
* displayMenu: Display all menu that available
* getFullMenu: get all the data about the menu that customer choose

**Menu**

* **Variable**
* pCrust[50]: string => pizza crust array
* pSpecial[50]: string => pizza speciality array
* pSize[50]: string => pizza size array
* pAmount[50]: int => pizza amount array
* -pPrice[50]: float => pizza price array
* Overall: float => The total price of pizza
* time: int => time for the pizza to finish
* CompanyName: string =>
* hour: int =>
* minute: int =>
* \*pesanan: Order =>
* \*id: Customer =>
* pizzac: string => pizza crust variable
* pizzas: string => pizza speciality variable
* size: string => pizza size variable
* amount: int => pizza amount variable
* **Function**
* Menu(Order) =>
* Menu(int,int) =>
* transferData(Customer): void =>
* DisplayMenu(): void =>
* PrintData(): void =>
* setHour(int): void =>
* setMinute(int): void =>
* setCompanyName(string): void =>
* setPizzaCrust(string, int ): void =>
* setSpecialityPizza(string, int ): void =>
* setSize(string , int ): void =>
* setAmount(int, int ): void =>
* getTime(): void =>

**DominoClass**

* **Function**
* CalculatePizza(int): virtual void =>
* DisplayOrder(int): void =>

**PizzaHutClass**

* **Function**
* CalculatePizza(int): virtual void =>
* DisplayOrderPizza():void =>

**Coding:**

**Example Print Screen Output:**

**Conclusion:**