

The Impractical Jongkers

Angelo Joaquin B. Alvarez

John Aidan Vincent M. Ng

Justin Carlo J. Reyes

Ateneo's Junction for Meal Assistance

Service Overview

Ateneo's Junction for Meal Assistance (AJMA) is a convenient and efficient solution designed to streamline the dining experience for Ateneo students. This service empowers students to place food orders digitally at nearby cafeterias on the Ateneo campus, eliminating the need for long queues and reducing waiting times. With AJMA, students can enjoy their favorite meals hassle-free, allowing them to make the most of their valuable time on campus.

Controllers

UserApp

Controller: OrderController			
Path: <i>/order/</i>			
Description: Customer-side of the service that controls their orders and view available items of selling vendors			
Name and Path	Description	Parameters	Returns
ViewVendors <i>/order/</i> GET	Allows the customer to view available vendors they can buy from.	Filter [String]	List of filtered vendors
ViewItems <i>/order/vendor/ID</i> GET	Allows the customer to view the items of a specified vendor	VendorID [Integer]	List of Items under VendorID
NewOrder <i>/order/new/</i>	Allows the customer to create a new order instance with empty items.	CustomerID <i>[Integer]</i> ContactNo.	OrderID <i>[Integer]</i>

POST		<i>[String].</i>	
AddItem <i>/order/add/</i> POST	<p>Adds a specified item to an order.</p> <p>If an order in the ordering status doesn't exist then it creates a new order and attaches it to the customer.</p> <p>If an order with "ordering" status exists, it appends to that order</p>	CustomerID <i>[Integer]</i> OrderID <i>[Integer]</i> ItemID <i>[Integer]</i>	List of ItemIDs under OrderID <i>[Integer]</i> TotalPrice <i>[Double]</i>
EditOrder <i>/order/edit/</i> POST (PUT if pwede)	Only available for the vendors, and allows them to change the status of an order (ordering, in progress, done, received), or cancel an order	OrderID <i>[Integer]</i> Status <i>[String]</i>	

Controller: CustomerController			
Path: <i>/customer/</i>			
Description: Admin-side and Customer-side that allows the addition of customers / users to the service			
Name and Path	Description	Parameters	Returns
ViewCustomer <i>/customer/</i> GET	Allows admin to view all users of the service		List of Customers
AddCustomer <i>/customer/register/</i>	Allows users to add themselves to the database	FirstName <i>[String]</i> LastName <i>[String]</i> ContactNo <i>[String]</i>	CustomerID <i>[Integer]</i>

POST		EmailAddress <i>[String]</i>	
RemoveCustomer <i>/customer/remove/</i> POST	Allows admin to delete users from the service	CustomerID <i>[Integer]</i>	
GetCustomerOrders <i>/[customerId]/orders/</i>	Allows admin to see customer's orders	CustomerID <i>[Integer]</i>	List of Orders

VendorApp

Controller: OrderController			
Path: <i>/order/</i>			
Description: Vendor-side controller endpoints for adding from UserApp			
Name and Path	Description	Parameters	Returns
NewOrder <i>/new/</i> POST	Creates new order	Order <i>[OrderDto]</i>	Order <i>[Order]</i>
AddToOrder <i>/add/</i> POST	Adds item to order	OrderCode <i>[String]</i> ItemID <i>[Integer]</i> Quantity <i>[Integer]</i>	Order <i>[Order]</i>
EditOrder <i>/edit/</i> OIST	Edits status of order	OrderID <i>[Integer]</i> Status <i>[String]</i>	
ViewOrder <i>/view/</i> POST	Shows order summary with total	OrderID <i>[Integer]</i>	Order Summary <i>[String]</i>

Controller: ItemController			
Path: <i>/item/</i>			
Description: Vendor-side of the service that controls the items they sell			
Name and Path	Description	Parameters	Returns
ViewItems <i>/item/</i> GET	Allows the vendors to view available items they sell.	VendorID <i>[Integer]</i>	List of items
NewItem <i>/item/new/</i> POST	Allows the vendors to create a new item instance	VendorID <i>[Integer]</i> ItemName <i>[String]</i> ItemPrice <i>[Double]</i>	
RemoveItem <i>/item/remove/</i> POST	Allows the vendors to remove an item.	ItemID <i>[Integer]</i>	
EditItem <i>/item/edit/</i> POST	Allows the vendors to edit an existing item instance	ItemID <i>[Integer]</i> ItemName <i>[String]</i> ItemPrice <i>[Double]</i> ItemDescription <i>[String]</i>	

Controller: VendorController
Path: <i>/vendor/</i>

Description: Admin-side and Vendor-side of the service that allows the creation of new vendors

Name and Path	Description	Parameters	Returns
ViewVendors <i>/vendor/</i> GET	Allows admin to view all vendors of the service	Filter <i>[String]</i>	List of Vendors
AddVendor <i>/vendor/add/</i> POST	Allows vendors to add themselves to the database	Name <i>[String]</i> Owner <i>[String]</i> Description <i>[String]</i> Cafeteria <i>[String]</i>	VendorID <i>[Integer]</i>
RemoveVendor <i>/vendor/remove/</i> POST	Allows admin to delete vendors from the service	VendorID <i>[Integer]</i>	

Controller: CafeteriaController

Path: */cafeteria/*

Description: Admin-side of the service that allows the manipulation of cafeteria information

Name and Path	Description	Parameters	Returns
ViewCafeterias <i>/cafeteria/all</i> GET	Allows admin to view all cafeterias of the service		List of Cafeterias
AddCafeteria <i>/cafeteria/add/</i>	Allows admin to add cafeterias to the database	Name <i>[String]</i> Description	CafeteriaID <i>[Integer]</i>

POST		<i>[String]</i>	
RemoveCafeteria <i>/cafeteria/remove/</i> POST	Allows admin to delete cafeterias from the service	CafeteriaID <i>[Integer]</i>	

OrderApp

Controller: MessageController			
Path: <i>/message/</i>			
Description: Service that sends messages from one app to the other			
Name and Path	Description	Parameters	Returns
Send <i>/send/</i> POST	Creates Message entity and "sends" it.	ContactNo <i>[String]</i> Message <i>[String]</i>	List of filtered vendors

Components

UserApp

Component: DatabasePopulator			
Controller: n/a			
Path: n/a			
Notes: Upon initialization, checks if database is empty. If it is, populates database with Customers and Orders			

Component: OrderComponent			
Controller: OrderController			
Path: <i>/order/new/</i>			
NewOrder	Creates a New Order and sends it as well to VendorApp	ItemID <i>[Integer]</i>	Order <i>[OrderReceive</i>

		Quantity <i>[Integer]</i>	<i>dDto]</i>
Path: <i>/order/add/</i>			
addToOrder	Adds an item to an existing order.	OrderID <i>[Integer]</i> ItemID <i>[Integer]</i> Quantity <i>[Integer]</i>	Order <i>[OrderReceive dDto]</i>
Path: <i>/order/edit/</i>			
EditOrder	Edits order status	OrderID <i>[Integer]</i> Status <i>[String]</i>	
Path: <i>/order/</i>			
GetAllVendors	Using retrofit, communicates with VendorController in VendorApp to return result of ViewVendors	Filter <i>[String]</i>	Vendors <i>[Vendor]</i>
Path: <i>/order/{vendorId}/</i>			
ViewItems	Using retrofit, communicates with ItemController in VendorApp to return result of ViewItems	VendorId <i>[Integer]</i>	List of Items

Component: CustomerComponent			
Controller: CustomerController			
Path: <i>/customer/</i>			
Method	Description	Parameters	Returns

Path: <i>/customer/new/</i>			
AddCustomer	Allows users to add themselves to the database	FirstName <i>[String]</i> LastName <i>[String]</i> ContactNo <i>[String]</i> EmailAddress <i>[String]</i>	
Path: <i>/customer/remove/</i>			
RemoveCustomer	Allows admin to remove users from the service	CustomerID <i>[Integer]</i>	
Path: <i>/customer/</i>			
GetAllCustomers	Allows admin to view all customers		Customers <i>[Customer]</i>
Path: <i>/customer/{customerId}/orders</i>			
GetCustomerOrders	Allows admin to view list of customer's orders	CustomerID <i>[Integer]</i>	Orders <i>[Order]</i>

VendorApp

Component: DatabasePopulator
Controller: n/a
Path: n/a
Notes: Upon initialization, checks if database is empty. If it is, populates database with Items, Vendors, Cafeterias

Component: OrderComponent
Controller: OrderController
Path: <i>/order/new/</i>

NewOrder	Creates a New Order	ItemID <i>[Integer]</i> Quantity <i>[Integer]</i>	Order <i>[OrderReceivedDto]</i>
GenerateRandomDigits	For generating order code		Digits <i>[String]</i>
Path: <i>/order/add/</i>			
addToOrder	Adds an item to an existing order.	OrderID <i>[Integer]</i> ItemID <i>[Integer]</i> Quantity <i>[Integer]</i>	Order <i>[OrderReceivedDto]</i>
Path: <i>/order/edit/</i>			
EditOrder	Edits order status and sends to UserApp	OrderID <i>[Integer]</i> Status <i>[String]</i>	
Path: <i>/order/view/</i>			
ViewOrder	Returns summary of order along with total price	OrderID <i>[Integer]</i>	Order Summary <i>[String]</i>

Component: ItemComponent			
Controller: ItemController			
Path: <i>/item/</i>			
Method	Description	Parameters	Returns
Path: <i>/item/new/</i>			
NewItem	Creates new item entity	Item	ItemID

		<i>[ItemDto]</i>	<i>[Integer]</i>
Path: <i>/item/remove/</i>			
DeleteItem	Allows vendors to remove items from their menu	ItemID <i>[Integer]</i>	ItemID <i>[Integer]</i>
Path: <i>/item/view/</i>			
ViewItems	Shows all items for given vendor	VendorID <i>[Integer]</i>	Items <i>[List<Item>]</i>
Path: <i>/item/edit/</i>			
EditItem	Allows vendors to edit items in their menu	VendorID <i>[Integer]</i> ItemID <i>[Integer]</i> ItemName <i>[String]</i> ItemPrice <i>[Double]</i>	Item <i>[Item]</i>

Component: VendorComponent			
Controller: VendorController			
Path: <i>/vendor/</i>			
Method	Description	Parameters	Returns
ViewVendors	List down all vendors of the service		List of vendors
CafeteriaFilter	Return vendors under given cafeteria name	Filter <i>[String]</i>	List of vendors
Path: <i>/vendor/new/</i>			
NewVendor	Allows Vendors to add themselves to the database	FirstName <i>[String]</i>	

		LastName <i>[String]</i> ContactNo <i>[String]</i> EmailAddress <i>[String]</i>	
Path: <i>/vendor/remove/</i>			
RemoveVendor	Allows admin to remove vendors from the service	VendorID <i>[Integer]</i>	

Component: CafeteriaComponent			
Controller: CafeteriaController			
Path: <i>/cafeteria/</i>			
Method	Description	Parameters	Returns
ViewCafeteria	List down all cafeterias of the service		List of cafeterias
Path: <i>/cafeteria/new/</i>			
AddCafeteria	Allows cafeterias to add themselves to the database	Name <i>[String]</i> Description <i>[String]</i>	CafeteriaID <i>[Integer]</i>
Path: <i>/cafeteria/remove/</i>			
RemoveCafeteria	Allows admin to remove cafeterias from the service	CafeteriaID <i>[Integer]</i>	

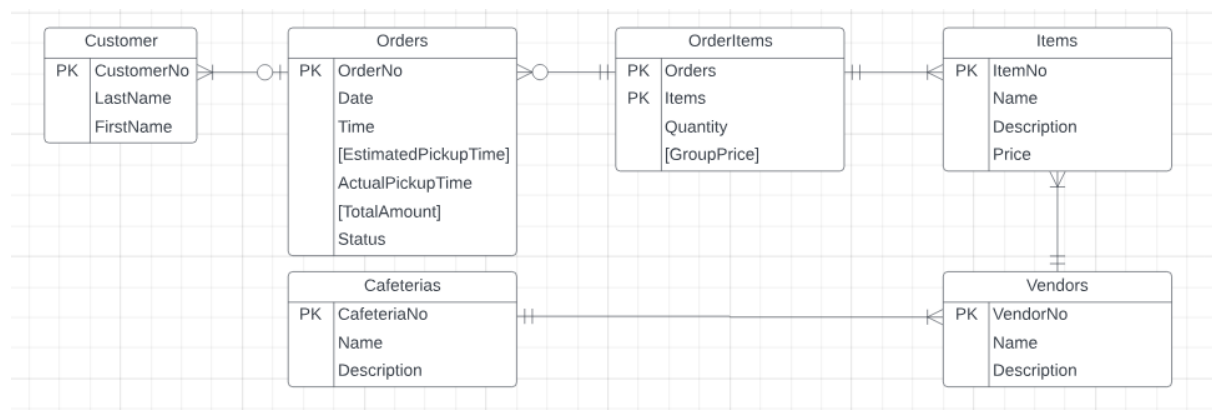
OrderApp

Component: InterComponent
Controller: MessageController

Path: /message/			
Method	Description	Parameters	Returns
Path: /message/send/			
Send	Creates Message entity	ContactNo [String] Message [String]	Confirmation message [String]

Entities and Repositories

The following entity-diagram relationship is used to determine the interactions between entities and their respective repositories. Each entity already has its own set of attributes that distinctively characterize it.



The list of the following entities, their attributes, and their respective repositories are as follows:

UserApp

Entity: Customer			
Repository: CustomerRepository			
Description: The customers that order from vendors			
Attribute	Description	Data Type	Restrictions
CustomerID	Unique identifier of a customer	Integer	PRIMARY KEY Auto Increments
LastName	Last name of the customer	String	NOT NULL

FirstName	First name of the customer	String	NOT NULL
ContactNo	Contact number of the customer	String	Size (11-12 characters)
EmailAddress	Email address of the customer	String	NOT NULL Email

Entity: Order			
Repository: OrderRepository			
Description: Particular orders of customers consisting of items (User-side version)			
Attribute	Description	Data Type	Restrictions
OrderID	Unique identifier of an order	Integer	PRIMARY KEY Auto Increments
OrderCode	Unique identifier for customers to identify order	String	NOT NULL
Customer	Customer who made the order	Customer	FOREIGN KEY ManyToOne (CustomerID of Customer)
DateOrdered	The date the customer created the order	Date	NOT NULL
TotalPrice	Calculated field that adds up the cost of groups of items in an order.	Double	NOT NULL
Status	Status of the order (Ordering, In Progress, Done, and Received)	String	NOT NULL
Repository Methods			
Name	Description	Purpose	
FindByCustomer_Id(customerId)	Returns list of orders given associated customer	For getCustomerOrders in	

		CustomerController
FindByOrderCode(orderCode)	Returns order based on given code	Used for adding items

VendorApp

Entity: Order			
Repository: OrderRepository			
Description: Particular orders of customers consisting of items (Vendor-side version)			
Attribute	Description	Data Type	Restrictions
OrderID	Unique identifier of an order	Integer	PRIMARY KEY Auto Increments
OrderCode	Unique identifier for customers to identify order	String	
OrderItems	Associated OrderItem entities	List of OrderItems	FOREIGN KEY (OrderItemID of OrderItems) OneToMany
DateOrdered	The date the customer created the order	Date	
TotalPrice	Calculated field that adds up the cost of groups of items in an order.	Double	
Status	Status of the order (Ordering, In Progress, Done, and Received)	String	
Repository Methods			
Name	Description	Purpose	
FindByOrderCode(orderCode)	Returns order based on given code	Used for adding items	

Entity: Item			
Repository: ItemRepository			
Description: Particular items or food vendors are selling			
Attribute	Description	Data Type	Restrictions
ItemID	Unique identifier of an item	Integer	PRIMARY KEY Auto Increments
Name	Name of the item	String	NOT NULL
Price	Price of a single item	Double	NOT NULL
Vendor	Associated Vendor	Vendor	FOREIGN KEY (VendorID of Vendor) ManyToOne
Description	Description of the Item	String	
Repository Methods			
Name	Description	Purpose	
FindAllByVendor	Returns list of items given associated vendor	For getting items in UserApp OrderController	

Entity: OrderItem			
Repository: OrderItemRepository			
Description: Connects the relationship between order and item entities, such that each order can have multiple items.			
Attribute	Description	Data Type	Restrictions
OrderItemID	Unique identifier for entity	Integer	PRIMARY KEY Auto increments
ItemID	ID of associated item	Integer	FOREIGN KEY (ItemID of Item)

OrderID	ID of associated order	Integer	FOREIGN KEY (OrderID of Order)
Quantity	Quantity of the item ordered	Integer	NOT NULL
GroupPrice	Calculated field from item price multiplied by the quantity of the item ordered.	Double	

Entity: Vendor			
Repository: VendorRepository			
Description: Vendors that sell a list of items to customers			
Attribute	Description	Data Type	Restrictions
VendorID	Unique identifier of a vendor	Integer	PRIMARY KEY Auto Increments
Name	Name of the vendor	String	NOT NULL
Owner	Name of the owner	String	NOT NULL
Description	Description of the vendor	String	NOT NULL
Items	List of Items sold	List of Items	FOREIGN KEY (ItemID of Item)
Cafeteria	Associated Cafeteria	Cafeteria	FOREIGN KEY (CafeteriaID of Cafeteria)
Repository Methods			
Name		Description	Purpose
FindAllByCafeteria		Returns list of vendors given cafeteria	For filtering vendors

Entity: Cafeteria
Repository: CafeteriaRepository

Description: Cafeterias that contain a set of vendors			
Attribute	Description	Data Type	Restrictions
CafeteriaID	Unique identifier of a cafeteria	Integer	PRIMARY KEY Auto Increments
Name	Name of the cafeteria	String	NOT NULL
Description	Description of the cafeteria	String	
Vendors	List of associated Vendors	List of Vendors	FOREIGN KEY (VendorID of vendors) OneToMany

OrderApp

Entity: Message			
Repository: MessageRepository			
Description: Message to be sent to vendor or user			
Attribute	Description	Data Type	Restrictions
MessageID	Unique identifier of a message	Integer	PRIMARY KEY Auto Increments
ContactNo	Contact Number of Recipient	String	NOT NULL Size (11-12 characters)
Message	Content of message	String	NOT NULL