MEMORIAL
UNIVERSITY

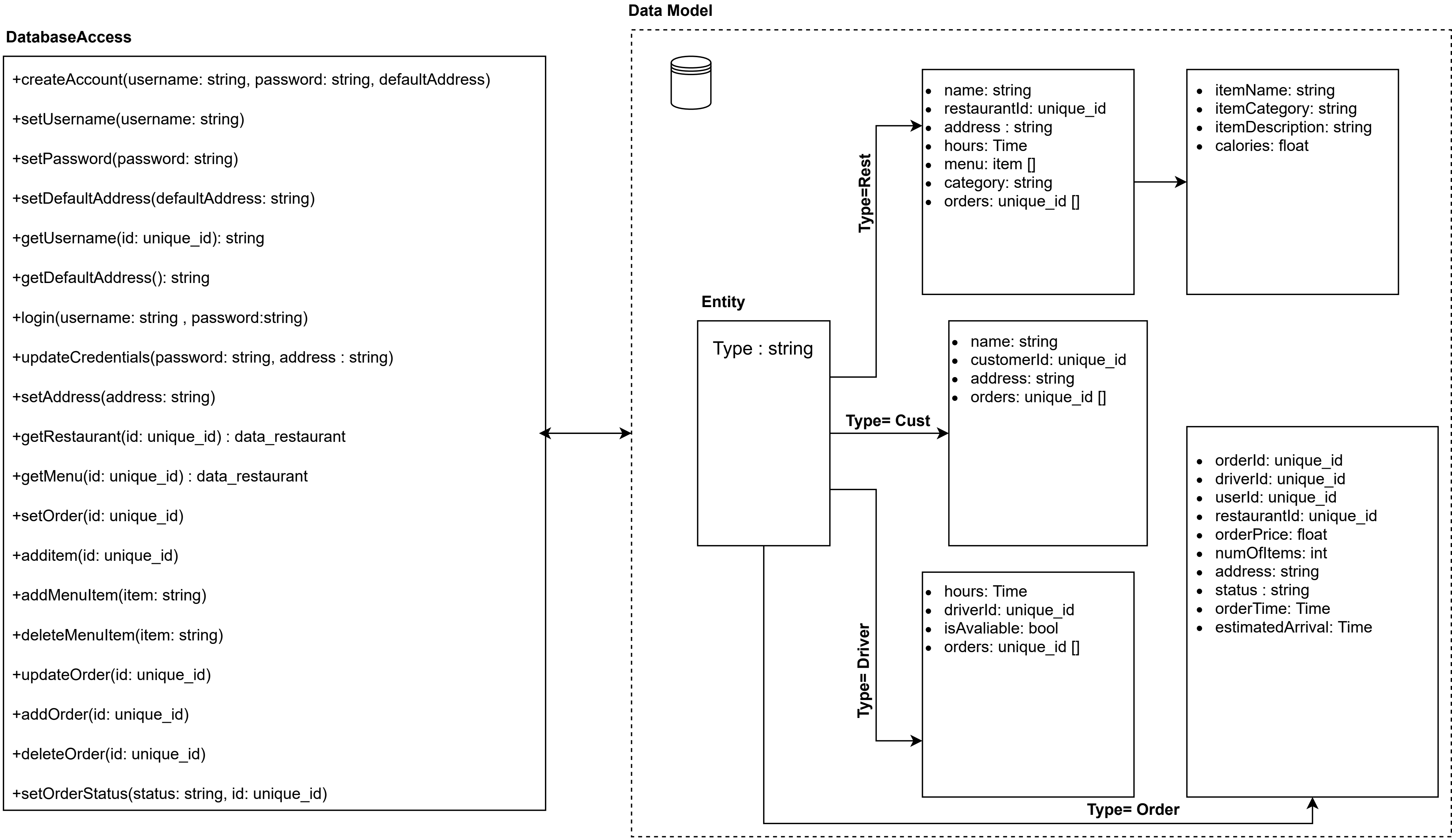
FACULTY OF ENGINEERING
AND APPLIED SCIENCE

Database Team

ECE 6400 - Software Development

Oct 1 - Fall 2021

Prof. Reza Shahidi



DatabaseAccess

+createAccount(username: string, password: string, defaultAddress)

+setUsername(username: string)

+setPassword(password: string)

+setDefaultAddress(defaultAddress: string)

+getUsername(id: unique_id): string

+getDefaultAddress(): string

+login(username: string , password:string)

+updateCredentials(password: string, address : string)

+setAddress(address: string)

+getRestaurant(id: unique_id) : data_restaurant

+getMenu(id: unique_id) : data_restaurant

+setOrder(id: unique_id)

+additem(id: unique_id)

+addMenuItem(item: string)

+deleteMenuItem(item: string)

+updateOrder(id: unique_id)

+addOrder(id: unique_id)

+deleteOrder(id: unique_id)

+setOrderStatus(status: string, id: unique_id)

Data Model

Entity

Type : string

Type=Rest

- name: string
- restaurantId: unique_id
- address : string
- hours: Time
- menu: item []
- category: string
- orders: unique_id []

- itemName: string
- itemCategory: string
- itemDescription: string
- calories: float

Type= Cust

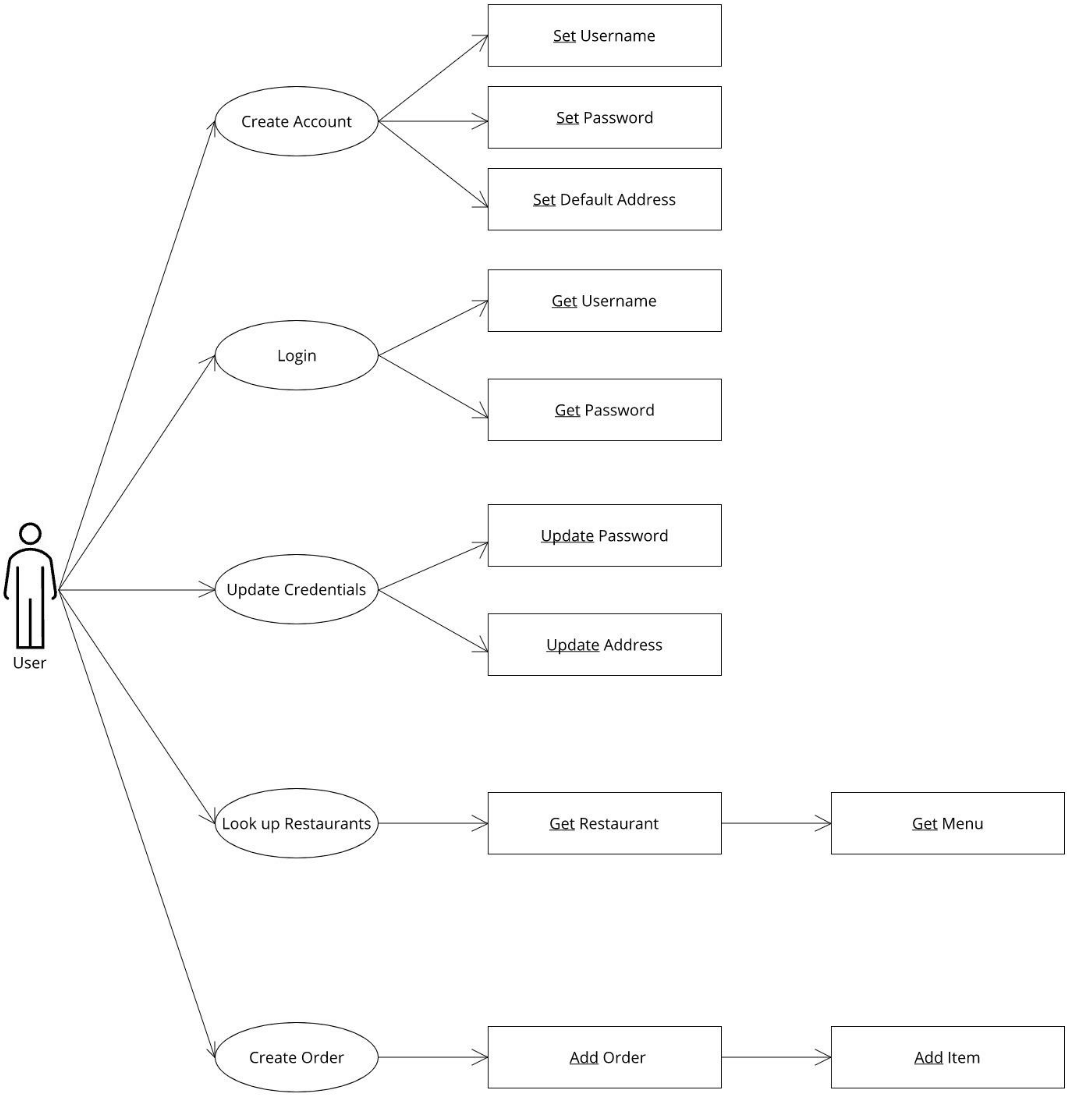
- name: string
- customerId: unique_id
- address: string
- orders: unique_id []

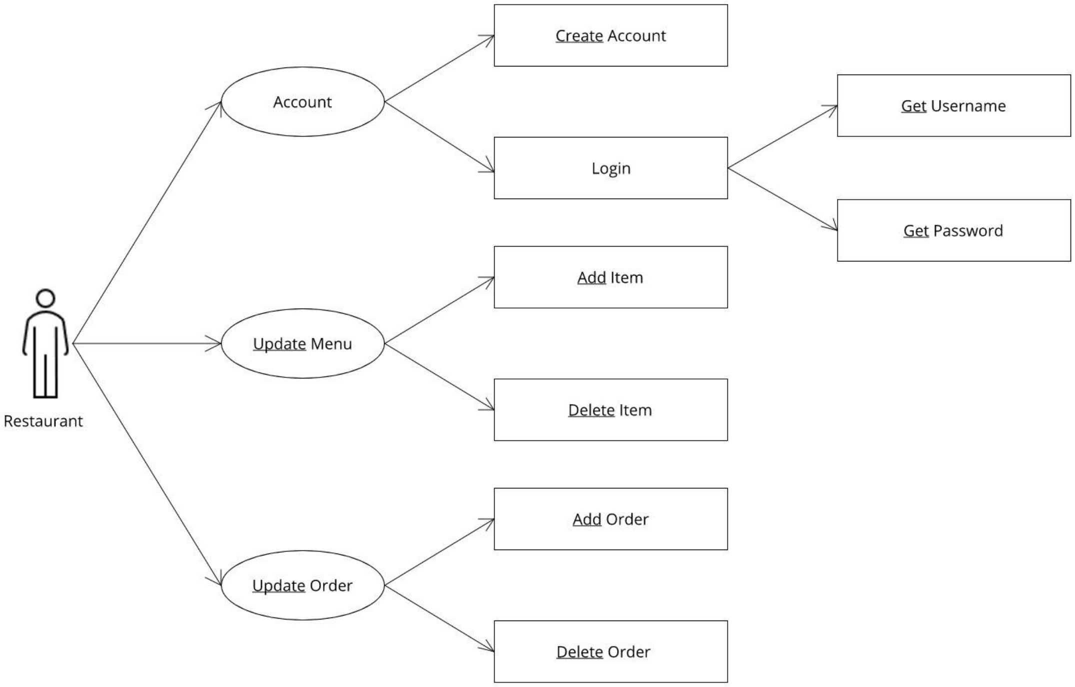
Type= Driver

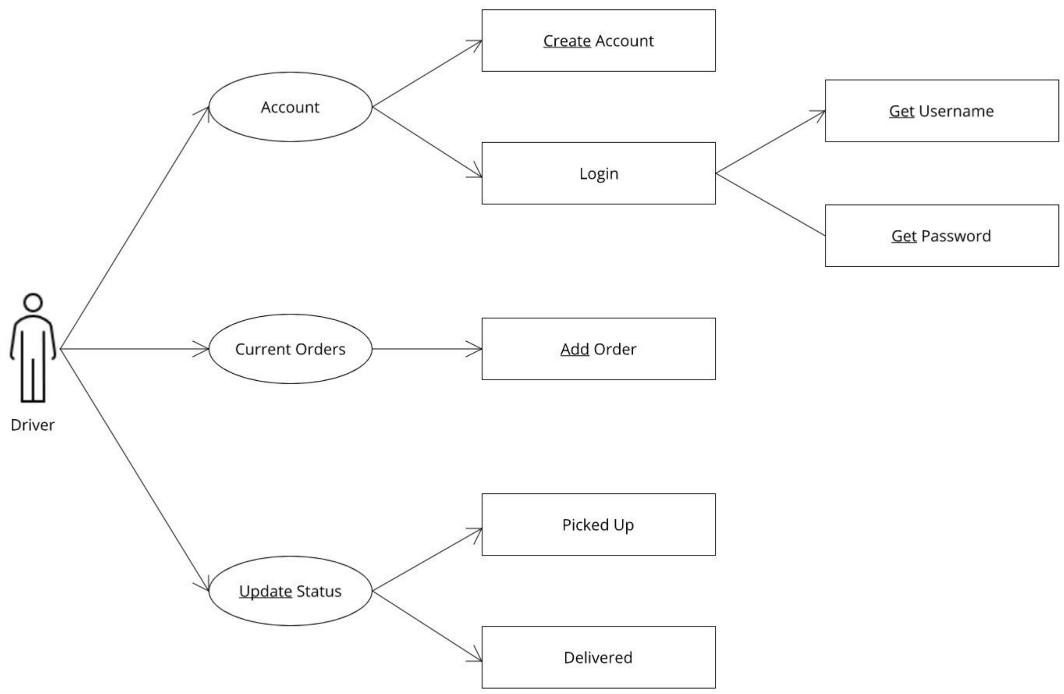
- hours: Time
- driverId: unique_id
- isAvaliable: bool
- orders: unique_id []

Type= Order

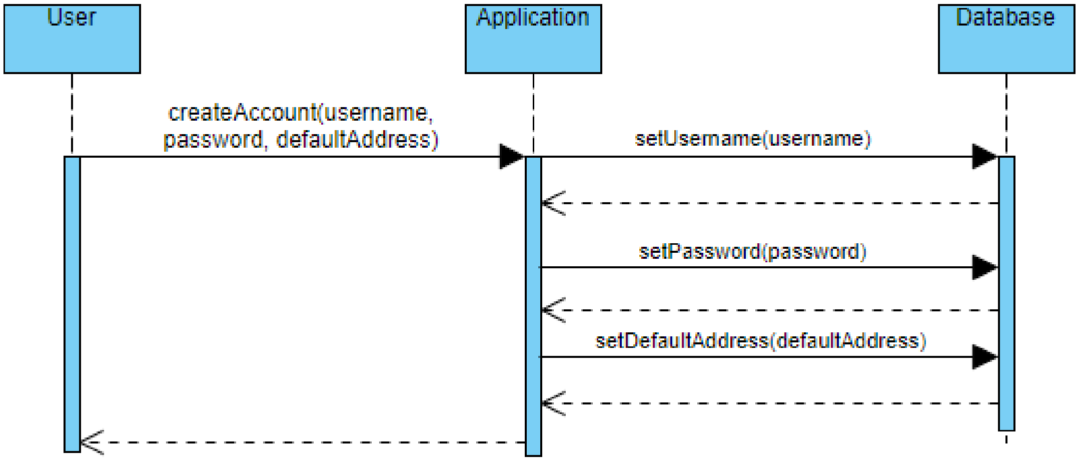
- orderId: unique_id
- driverId: unique_id
- userId: unique_id
- restaurantId: unique_id
- orderPrice: float
- numOfItems: int
- address: string
- status : string
- orderTime: Time
- estimatedArrival: Time


Set Username
Create Account
Set Password
Set Default Address
Get Username
Login
Get Password
Update Password
User
Update Credentials
Update Address
Look up Restaurants
Get Restaurant
Get Menu
Create Order
Add Order
Add Item

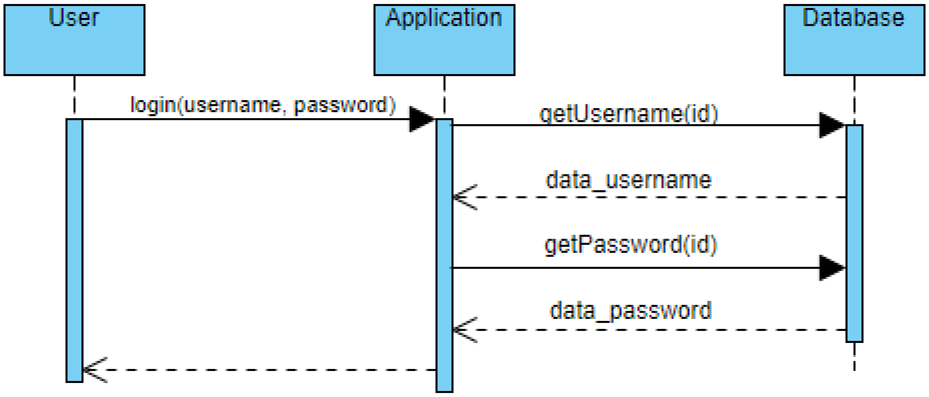

Restaurant
Account
Update Menu
Update Order
Create Account
Login
Get Username
Get Password
Add Item
Delete Item
Add Order
Delete Order


Create Account
Account
Get Username
Login
Get Password
Current Orders
Add Order
Driver
Picked Up
Update Status
Delivered

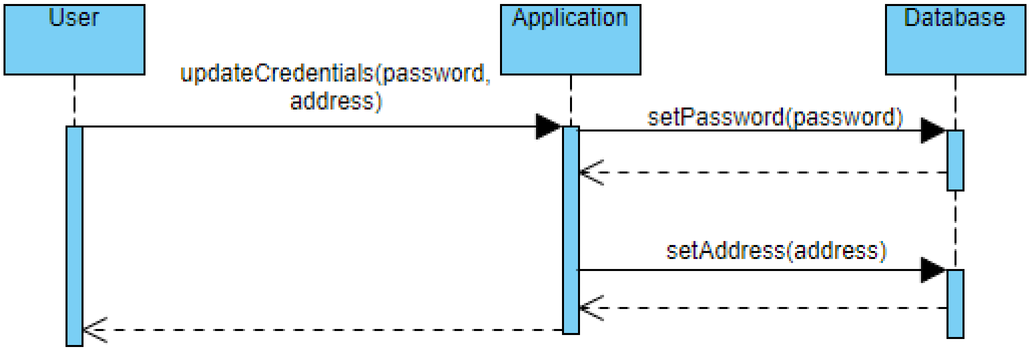
Account Creation (identical for User, Driver, and Restaurant)


User
Application
Database
createAccount(username, password, defaultAddress)
setUsername(username)
setPassword(password)
setDefaultAddress(defaultAddress)

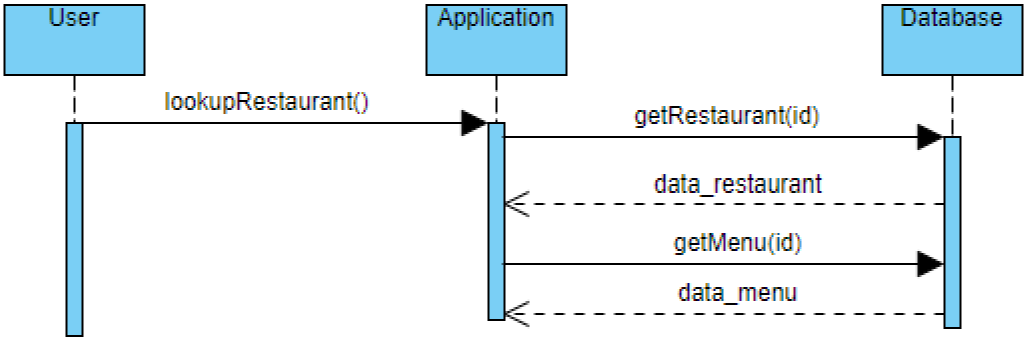
Login (identical for User, Driver, and Restaurant)



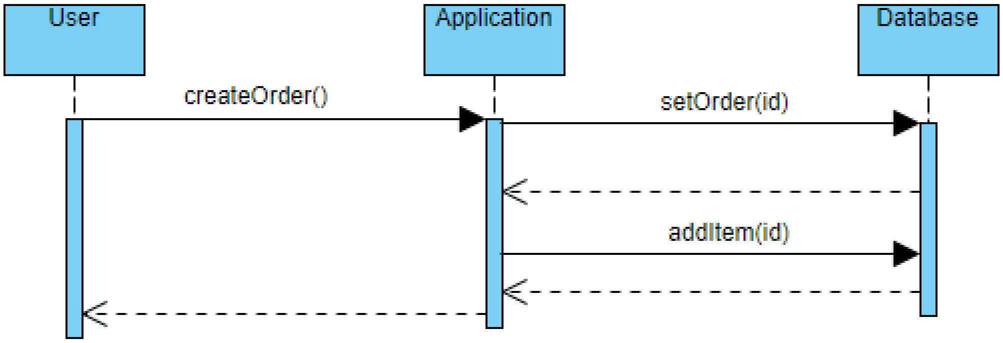
Update Credentials



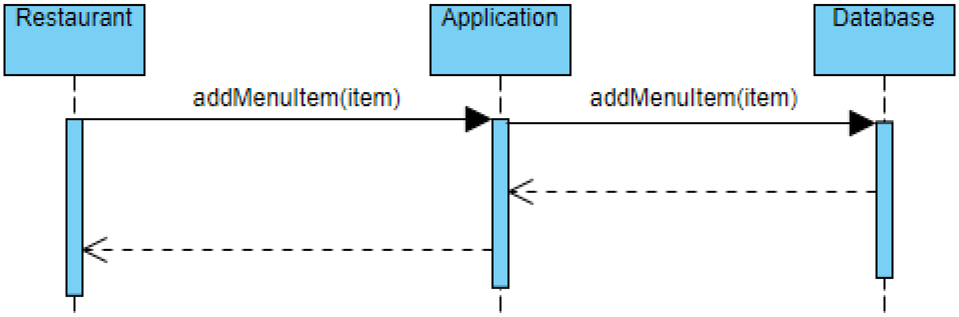
Look up restaurants



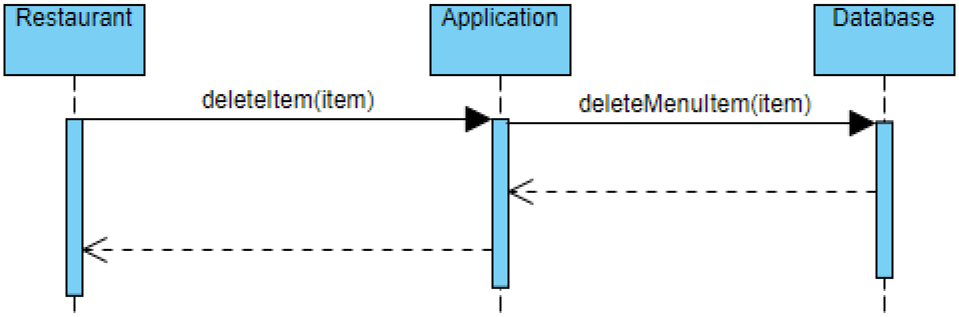
Create Order



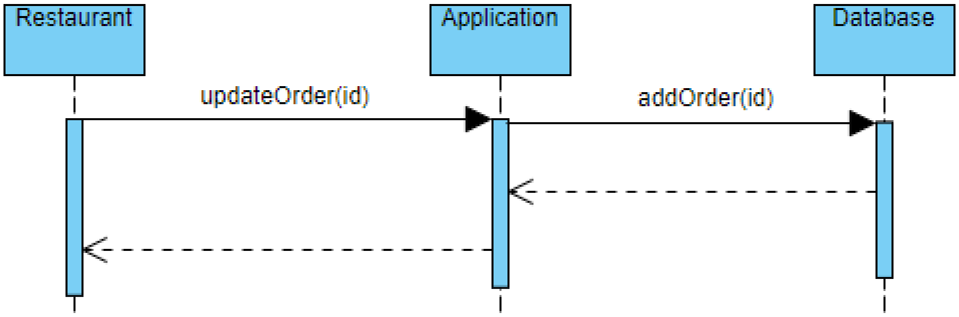
Add Menu Item



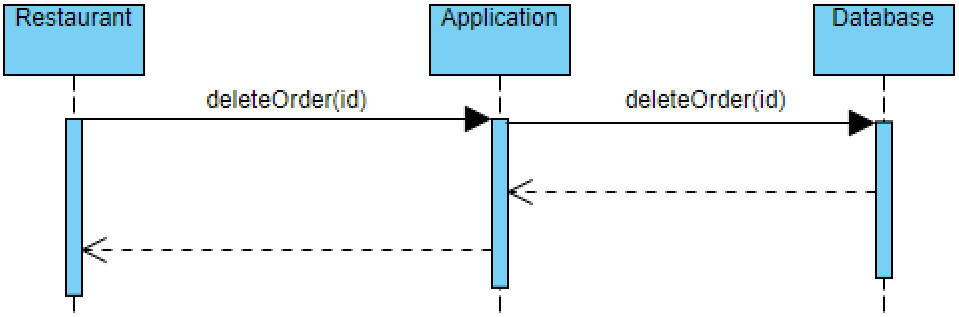
Delete Menu Item



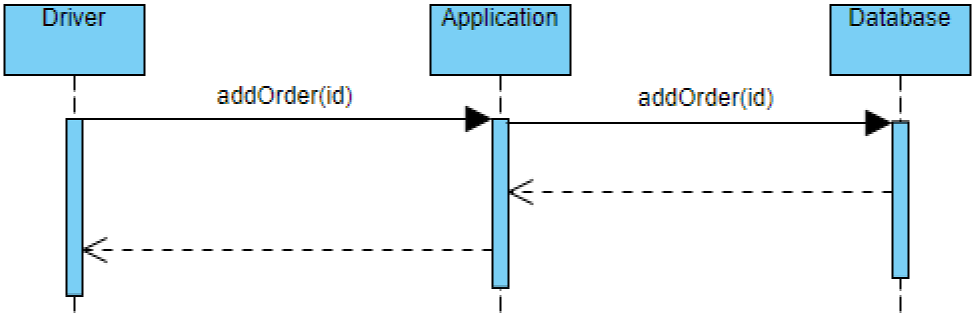
Add Order (Restaurant)



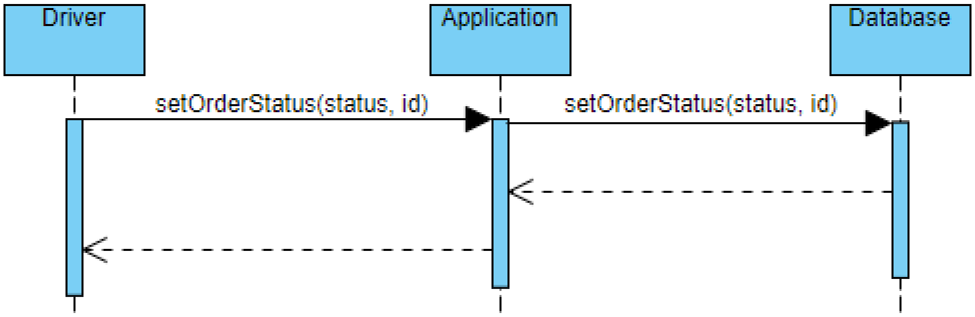
Delete Order



Add Order (Driver)



Update Status



Test case 1 for Database:

Test type: Functionality of creating and logging into a user account.

Description: User creates a user account and logs in using their user name and password.

Test Step(s):

1) Following the user selecting the create account button and providing the necessary username, password, and default address information, create a new user entity in the database.
2) Set the user’s parameters of username, password, and default address
3) Check all parameters corresponding to the new user entity (username, unique user id, etc.)

Expected Result(s):

1) User entity created in database
2) User entity has correct parameters when accessed

Actual Result: Console output of User entity attributes or visual GUI element of user account page later.

PASS/FAIL: NAN

Test case 2 for Database:

Test type: Functionality of accessing restaurant database.

Description: User selects a restaurant.

Test Step(s):

1) User selects a restaurant.
2) Restaurant database is accessed
3) Selected restaurant is returned to user

Expected Result(s):

1) Restaurant database is successfully accessed
2) User selected restaurant is successfully returned

Actual Result: Console output showing the restaurant is selected by the user.

PASS/FAIL: NAN

Test case 3 for Database:

Test type: Functionality of creating an order entity.

Description: Order entity is created in database.

Test Step(s):

1) User creates an order, order entity is created in database
2) As user adds items to order, order entity adds item attributes

Expected Result(s):

1) Order entity is successfully created
2) Items are successfully added to the order entity

Actual Result(s): Console output showing live updates as an order entity is created and modified.

PASS/FAIL: NAN

Test case 4 for Database:

Test type: Functionality of logging into Restaurant account.

Description: Restaurant account owner logs into their account.

Test Step(s):

1) Restaurant user provides username and password login information
2) Getters retrieve username and password account information

Expected Result(s):

1) Getters successfully retrieve account information

Actual Result(s): Restaurant user logs into their account.

PASS/FAIL: NAN

Test case 5 for Database:

Test type: Functionality of updating menu.

Description: Restaurant user updates menu.

Test Step(s):

1) Restaurant user adds item to menu.
2) Restaurant user deletes item from menu.

Expected Result(s):

1) Items are successfully added to menu.
2) Items are successfully deleted from menu.

Actual Result(s): Menu is modified correctly.

PASS/FAIL: NAN

Test case 6 for Database:

Test type: Functionality of updating orders.

Description: Orders are automatically updated in the restaurant database.

Test Step(s):

1) Order is added to restaurant active order database
2) Order is completed and deleted from active order database

Expected Result(s):

1) Orders are successfully added and deleted from active order database

Actual Result(s): Restaurant active order database is modified successfully.

PASS/FAIL: NAN

Test case 7 for Database:

Test type: Functionality of logging into Driver account.

Description: Driver account owner logs into their account.

Test Step(s):

1) Driver user provides username and password login information
2) Getters retrieve username and password account information

Expected Result(s):

1) Getters successfully retrieve account information

Actual Result(s): Driver user logs into their account.

PASS/FAIL: NAN

Test case 8 for Database:

Test type: Functionality of updating the status of an order.

Description: Driver updates the status of the order.

Test Step(s):

1) Driver updates order to “picked up”
2) Driver updates order to “delivered”

Expected Result(s):

1) Order status successfully changes to required states

Actual Result(s): Driver updates order status.

PASS/FAIL: NAN