Design Report

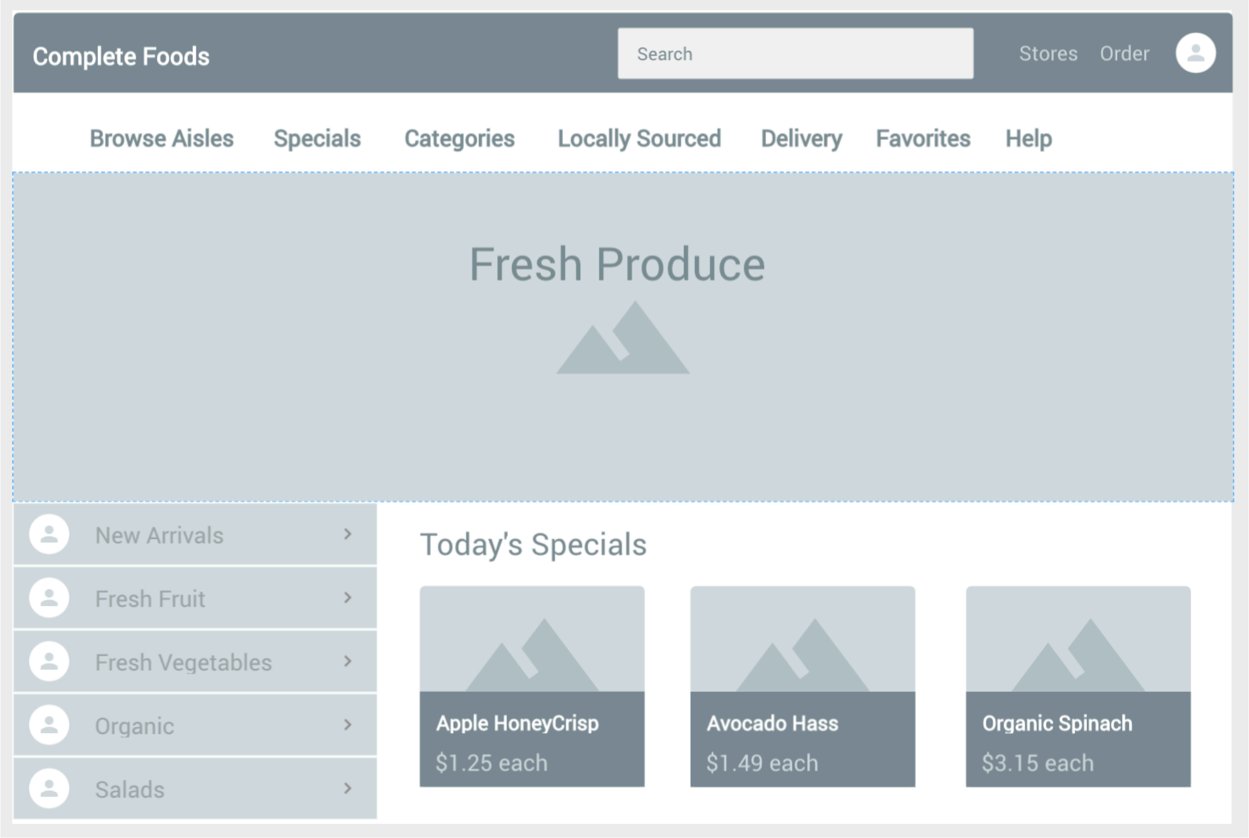
Company: Complete Foods Inc.
Project: Online Grocery Shopping Application
Team: Sergio Servantez, Michal Stafira, Weicong Hong, Zhihao Ai
Version 1.0

User Interface Prototype

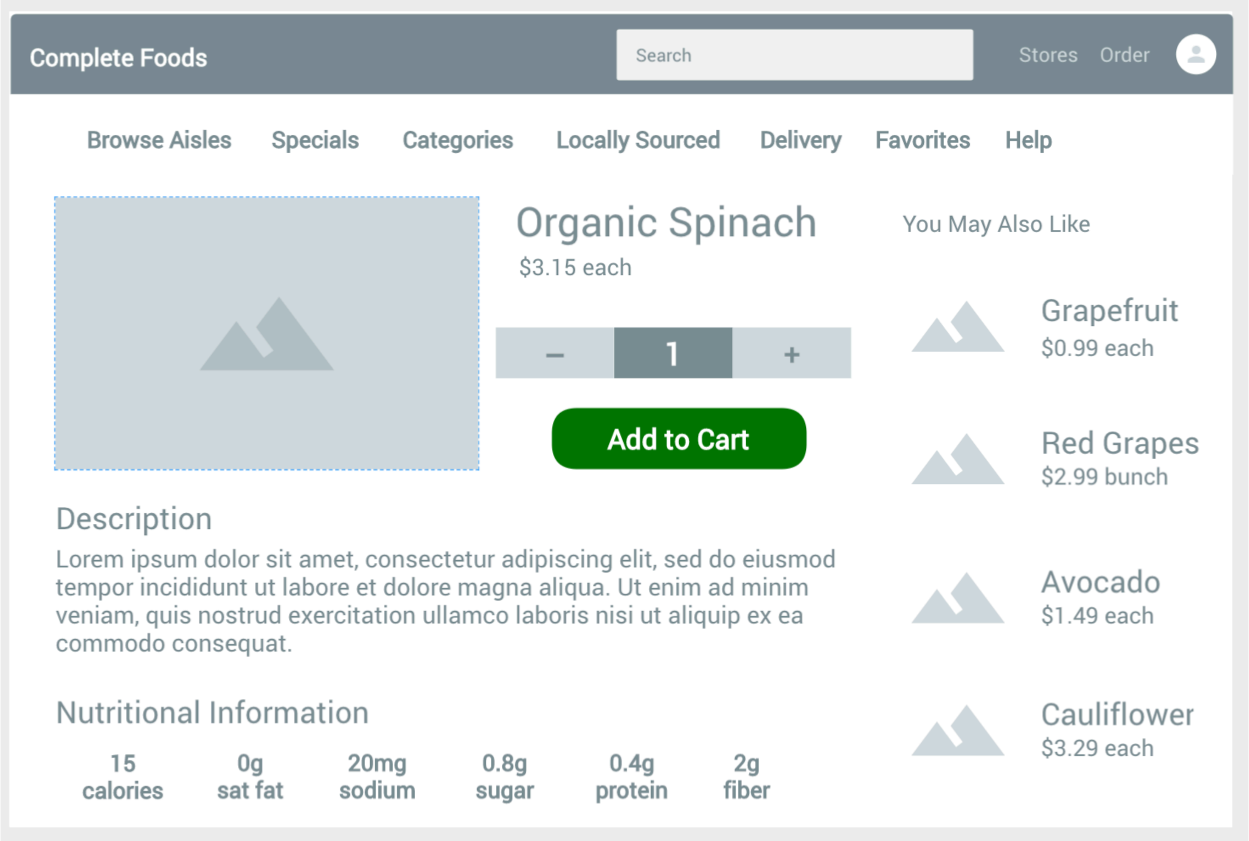
Our user interface is designed to be simple and intuitive. The page layout is constraint based which provides for automated image resizing and reactive page layout based on individual devices. Every aspect of the UI has been designed for ease of use. Users are invited to create customer accounts where they are able to store payment information, delivery addresses and favorite items for a quick and simple check-out experience. Our core customers are very health conscious so are interface is designed to provide the nutritional and quality information our customers are seeking. For a more in-depth look at the user interface, you can use the following credentials:

URL: https://www.fluidui.com
Login: sergio.servantez.esq@gmail.com
Password: Stafira487

Home Page



Product Detail Page



Organic Spinach

$3.15 each

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

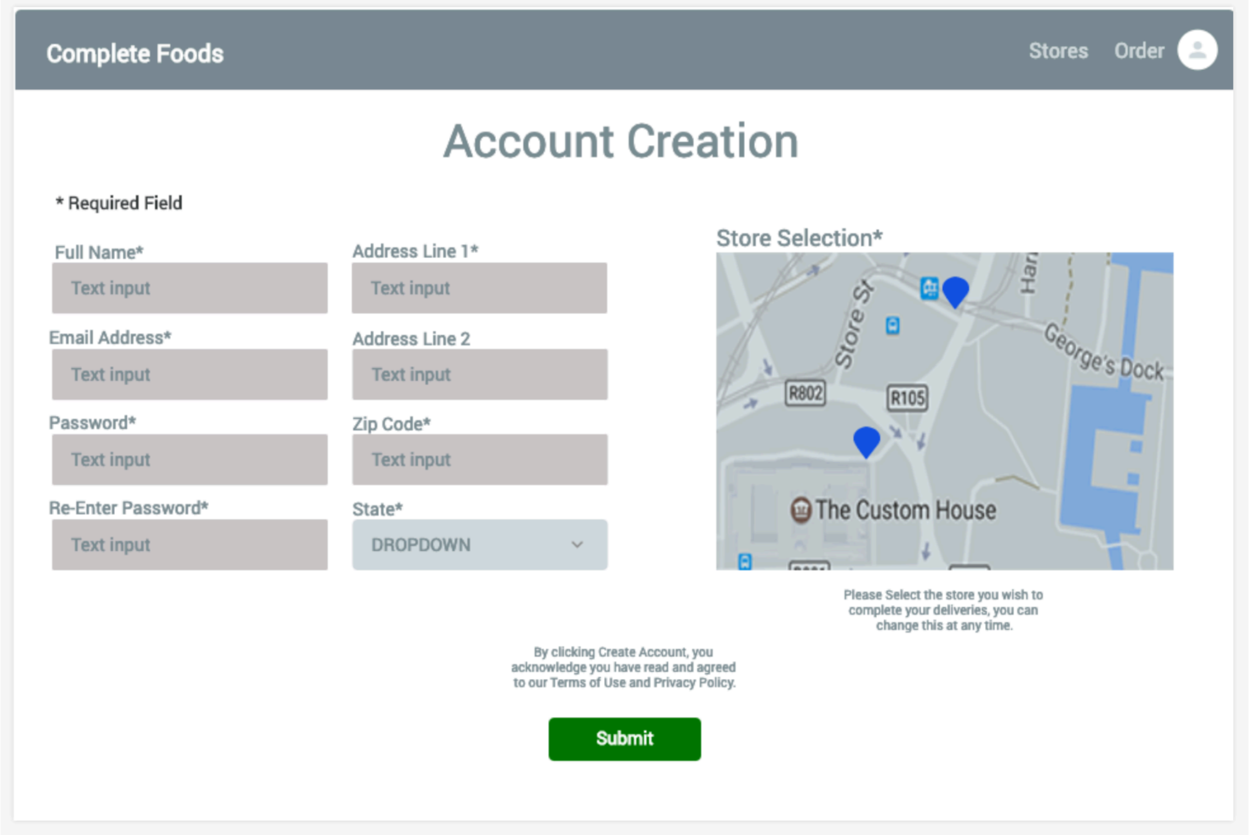
Nutritional Information

15 calories	0g sat fat	20mg sodium	0.8g sugar	0.4g protein	2g fiber

You May Also Like

Grapefruit	$0.99 each
Red Grapes	$2.99 bunch
Avocado	$1.49 each
Cauliflower	$3.29 each

Account Creation



Account Creation

* Required Field

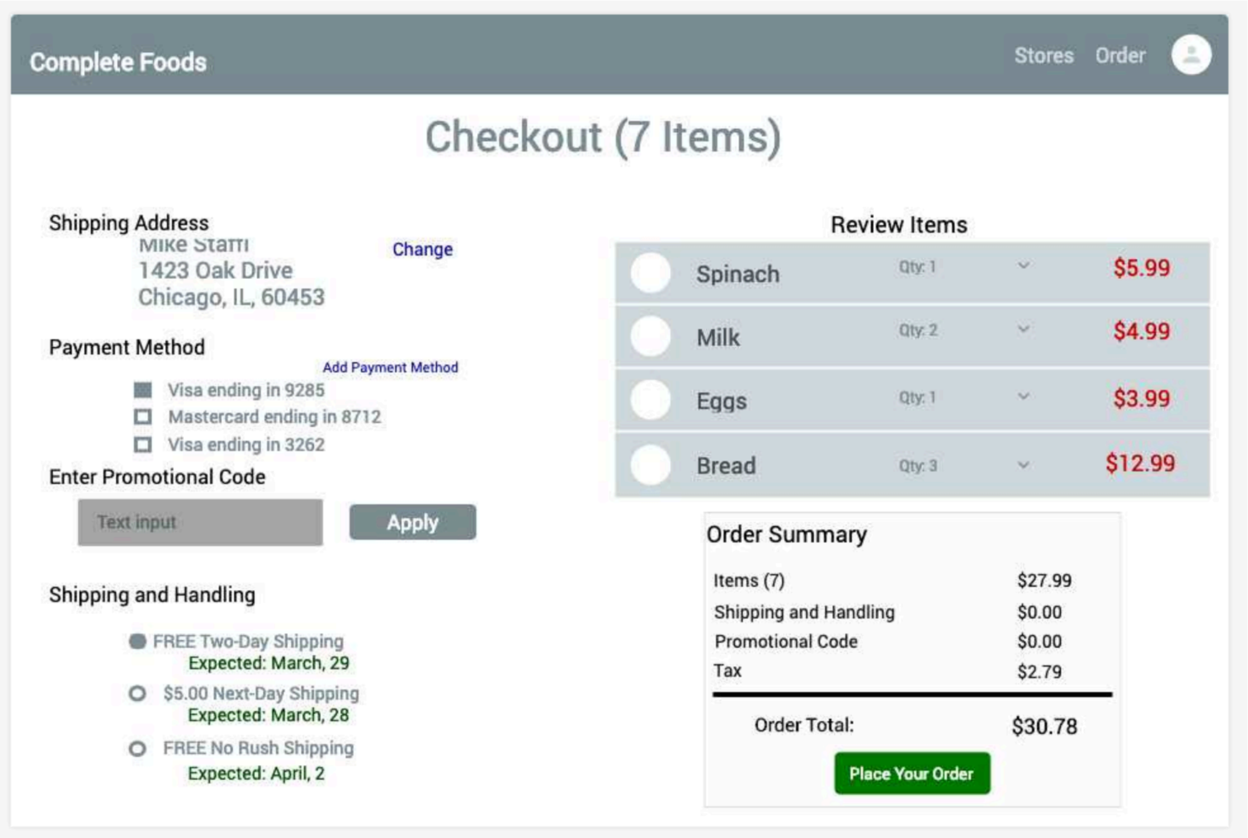
Full Name*	Address Line 1*
Text input	Text input
Email Address*	Address Line 2
Text input	Text input
Password*	Zip Code*
Text input	Text input
Re-Enter Password*	State*
Text input	DROPDOWN

Store Selection*

Please Select the store you wish to complete your deliveries, you can change this at any time.

By clicking Create Account, you acknowledge you have read and agreed to our Terms of Use and Privacy Policy.

Checkout



Checkout (7 Items)

Shipping Address

Mike Staffi
1423 Oak Drive
Chicago, IL, 60453

Payment Method

■ Visa ending in 9285
□ Mastercard ending in 8712
□ Visa ending in 3262

Enter Promotional Code

Text input

Shipping and Handling

● FREE Two-Day Shipping
Expected: March, 29
○ $5.00 Next-Day Shipping
Expected: March, 28
○ FREE No Rush Shipping
Expected: April, 2

Review Items

Spinach	Qty: 1	$5.99
Milk	Qty: 2	$4.99
Eggs	Qty: 1	$3.99
Bread	Qty: 3	$12.99

Order Summary

Items (7)	$27.99
Shipping and Handling	$0.00
Promotional Code	$0.00
Tax	$2.79
Order Total:	$30.78

System Overview

Cloud-Server Component: this component is responsible for servicing data requests from client devices. These requests will include product information, store information, user profile data, purchase transactions and delivery information. This component must receive these requests, correctly query the underlying database and return the requested information.

Cloud-Client Component: this component is responsible for making requests to the cloud-server based on user interactions and consuming the resulting response.

Data-Security Component: this component is responsible for authenticating users before any private or profile information is accessed or manipulated.

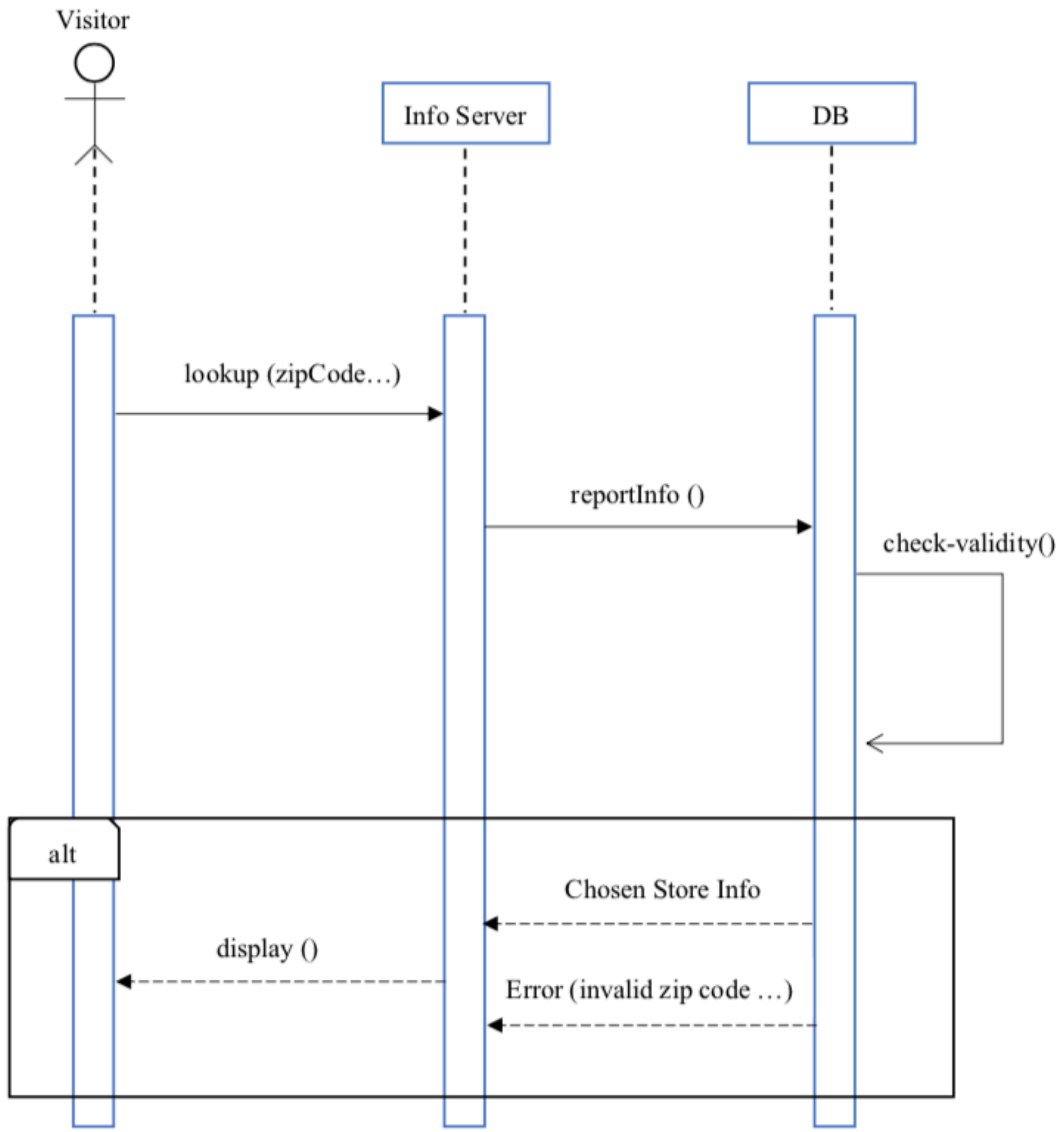
Data-Model Component: this component is responsible for converting the data received by the cloud-client into local objects and data structures.

Data-Controller Component: this component is responsible for interpreting user interactions, retrieving data requested by these interactions, and returning the requested data to the underlying view objects. In the event the requested data is not in local memory, this component is responsible for making the necessary service requests via the cloud-client component.

View Component: this component is responsible for presenting the end-user interface, receiving interaction events from the user, and updating the display based on data received from the controller.

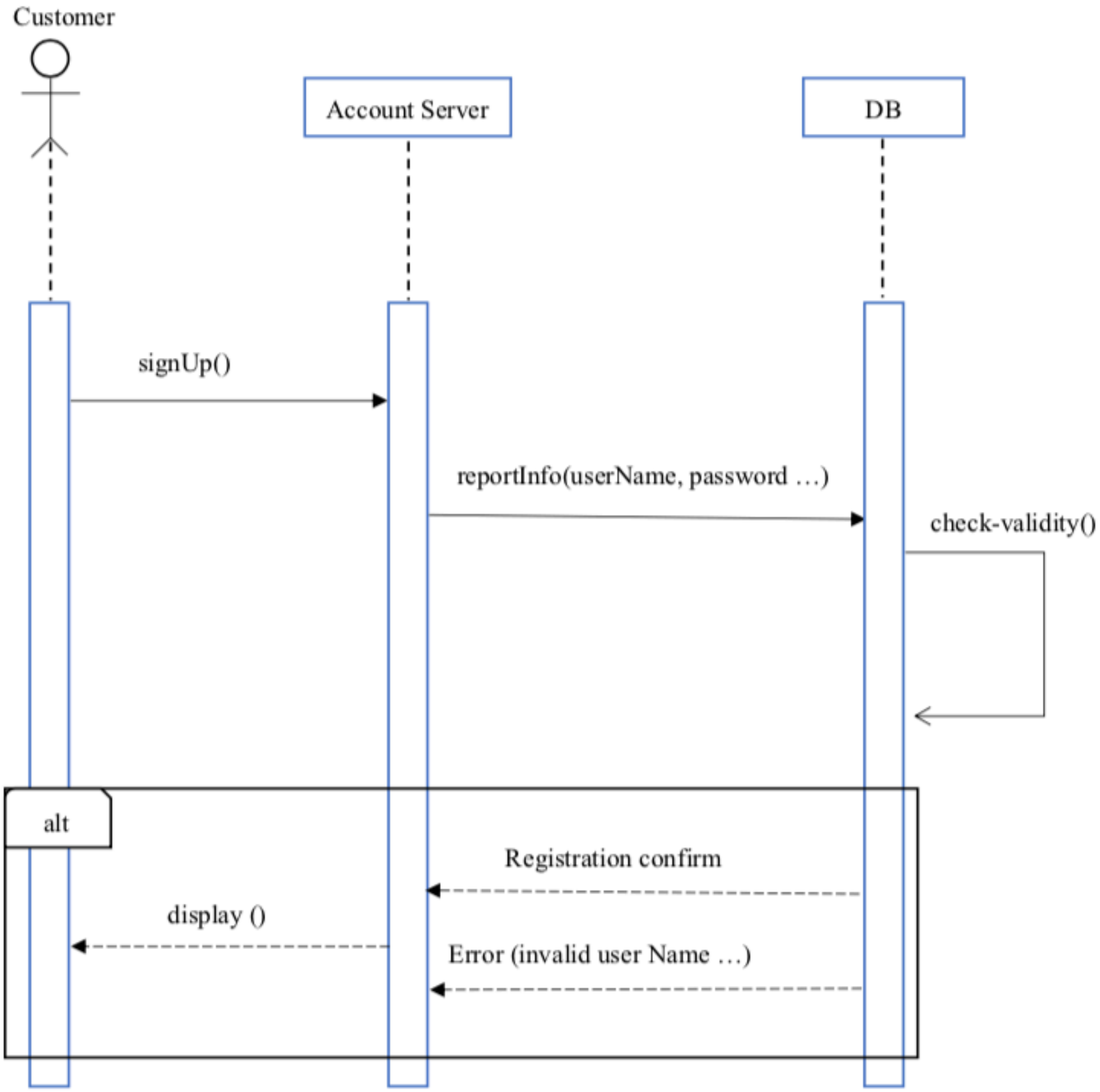
System and Data Models

Sequence diagram for View chosen store information



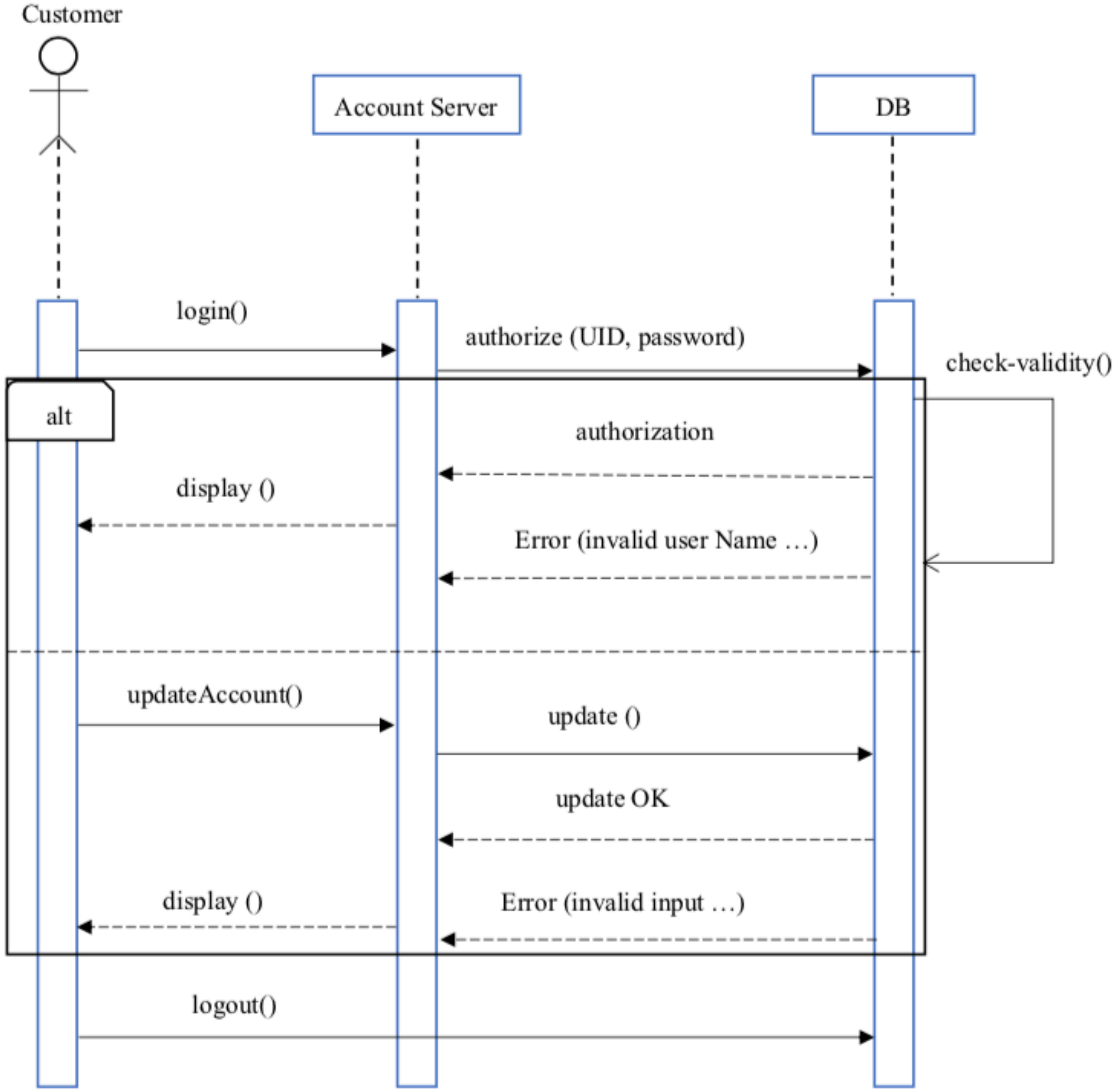
```
// a visitor allocates the nearest store
lookup (zipCode, cityName, …)
        request passes to Info Server
                request passes to DB
                        DB check validity(zipCode, cityName, …)
                                if valid    return the nearest store info to Info Server
                                else        return error msg
        return msg
```

Sequence diagram for Register customer account



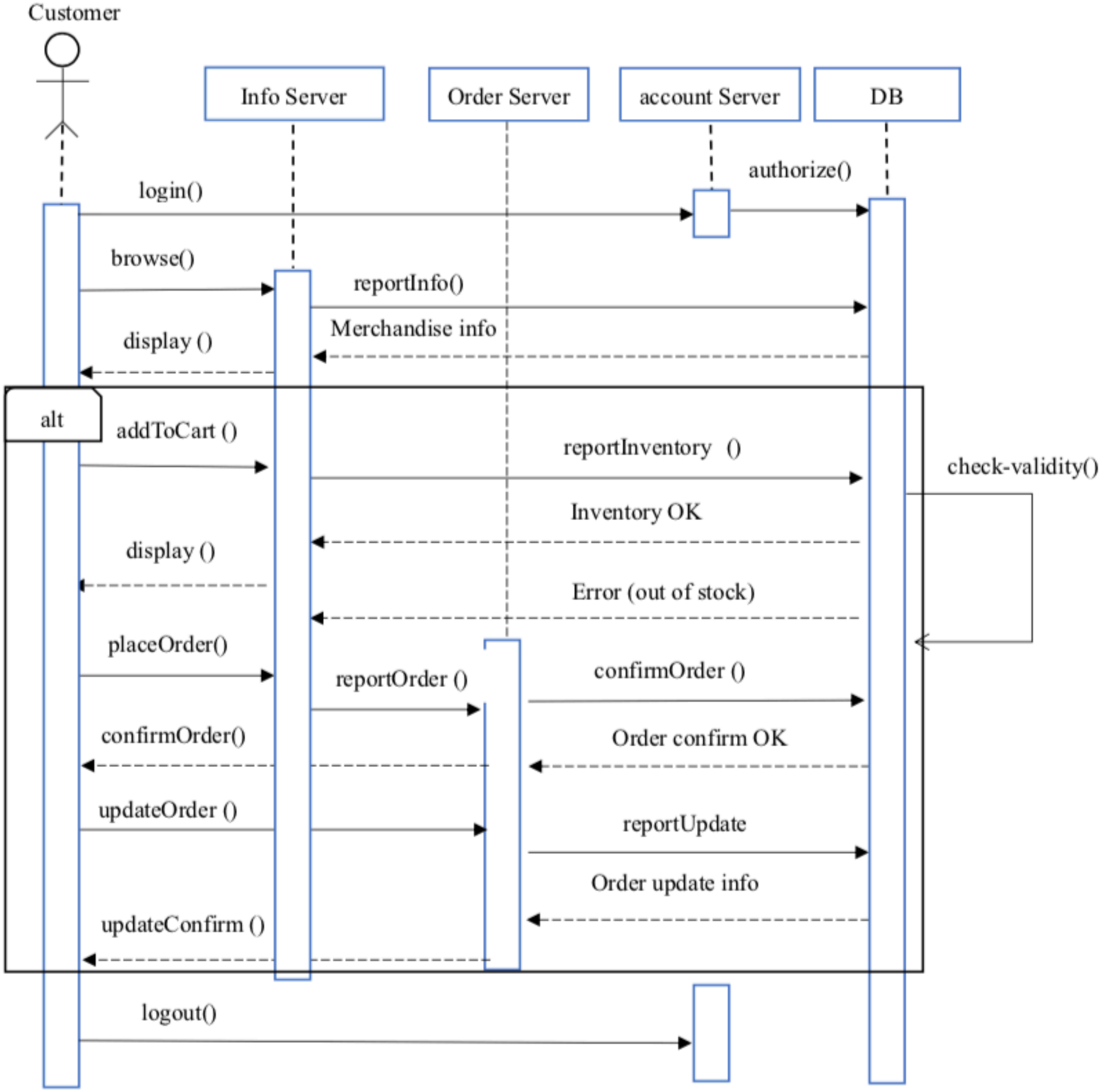
```
// a visitor signup an account
signUp()
    Account Server displays sign up web page
    validity check pass to DB
        if all input valid
            record is created in DB
            return valid msg to Account Server
        else return error msg
    return signup confirmation
```

Sequence diagram for Update customer account information



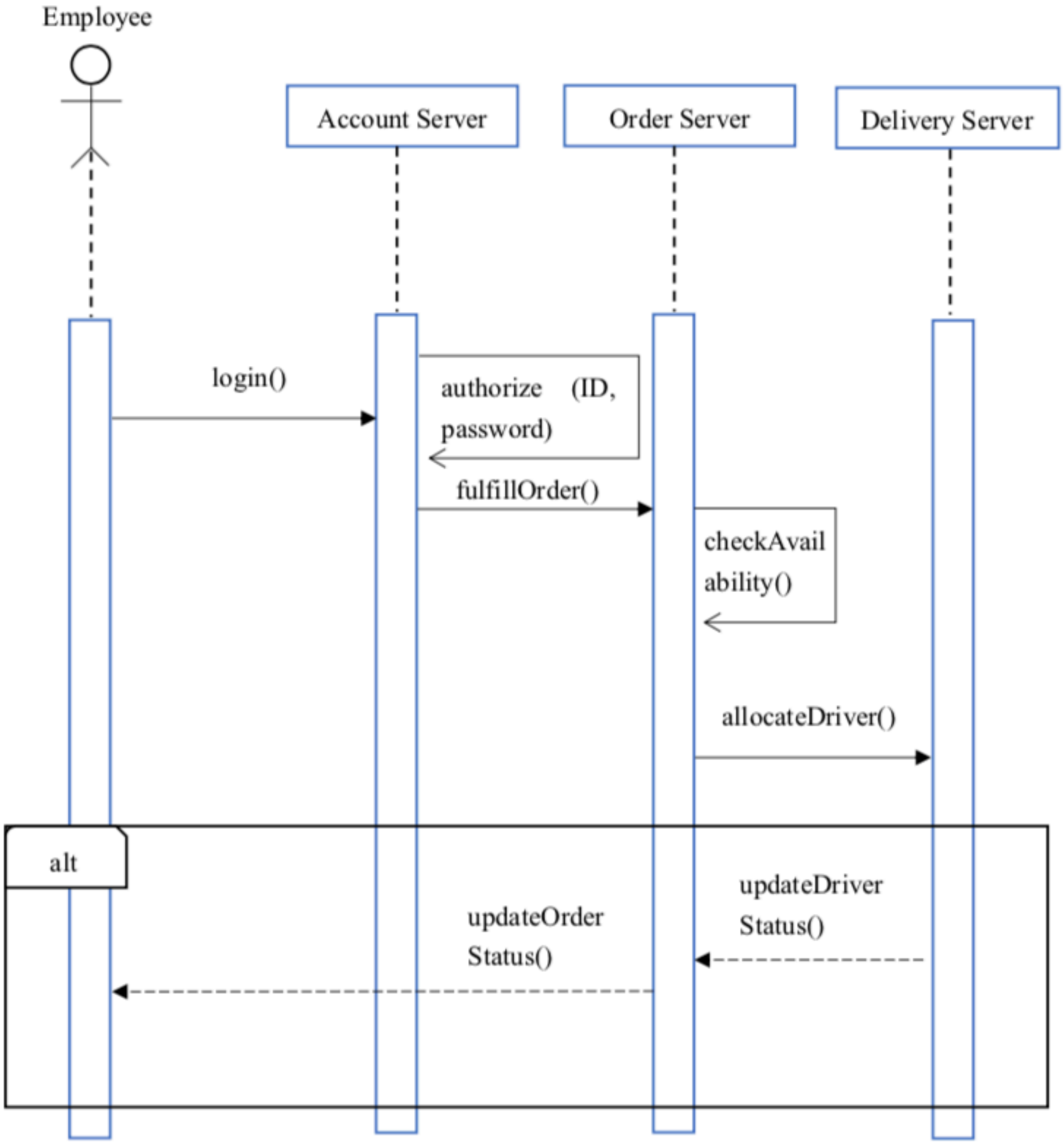
```
// a customer logins to his/her account
login()
    Account Server display the login webpage
    Account Server passes login credential to DB for authorization check
        if valid  DB returns valid msg to Account Server
        else      DB denies login service
    retum customer account web page


// a customer manages his/her account info
updateAccount()
    Account Server pass update info to DB
        DB checks validity of the info
            if valid
                    record is updated in DB
                    return update OK msg
            else    return error msg
    retum account web page with successful update confirmation/error msg


// a customer logout his/her account
logout()
    DB returns logout success to Account Server
    retum store web page with logout success msg
```

Sequence diagram for Place customer order



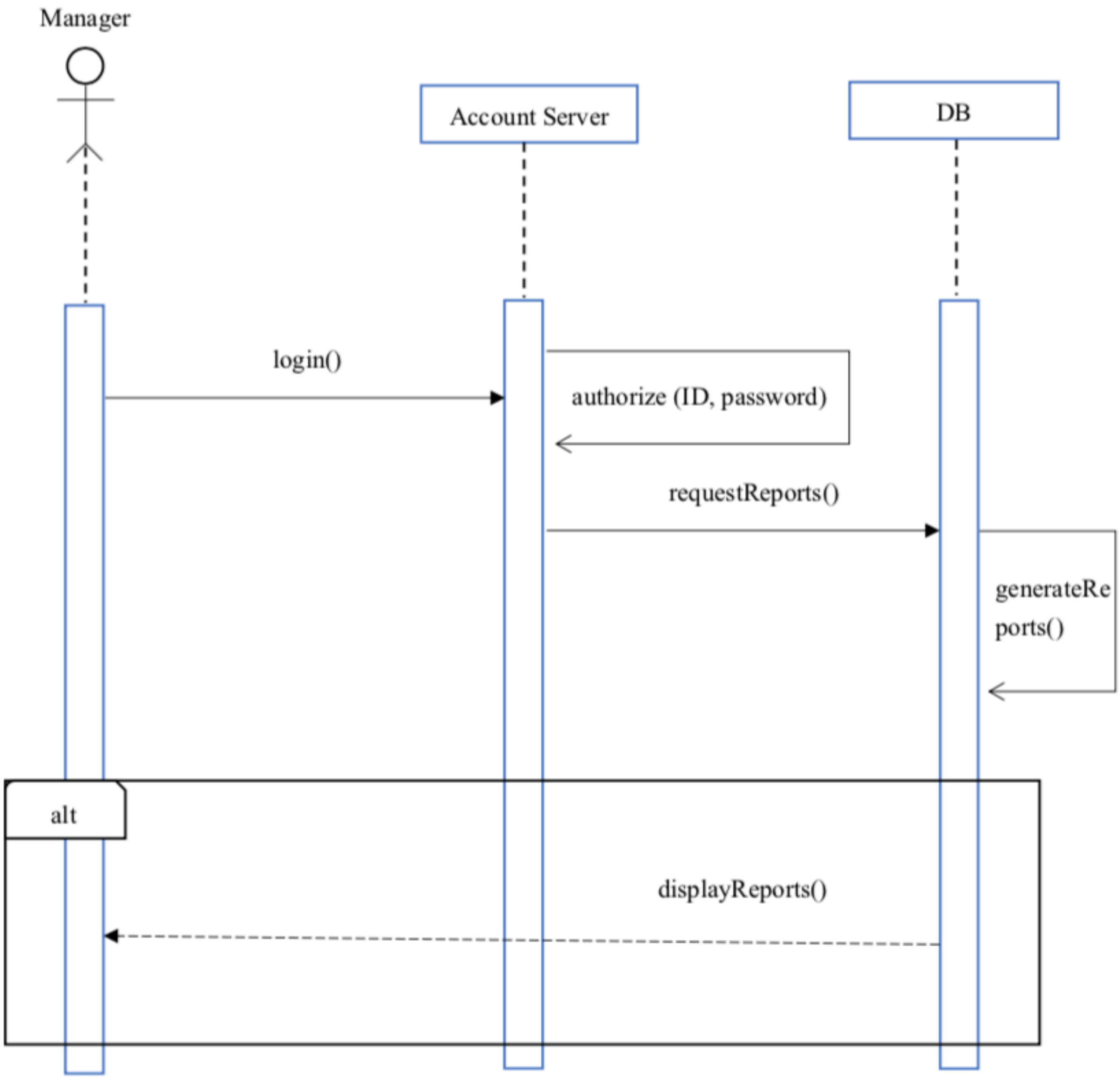
```
// a customer browse a store web page
browse()
    Infor Server pass browse request to DB
    DB returns info to Info Server
    display()
        return store web page from the Info Server


// a customer adds items to the shopping cart
addToCart()
    Info Server pass request to DB
        DB checks inventory
            if not 0    return inventory OK
            else        return error msg
    display()
        return store web page from the Info Server


// a customer place order
palceOrder()
    Order Server displays check out web page
    Order Server passes payment info to DB
        DB verifies payment info with Banking System
            if valid
                # of inventory is decremented
                return payment OK
            else return error msg
    confirmOrder(msg)
        return Order Server displays web page with successful order confirmation/error msg


// a customer updates his/her orders
updateOrder()
    Order Server passes update info to DB
        DB checks info validity
            if valid
                record is updated in DB
                return order update OK
            else return error msg
    updateConfirm(msg)
        return Order Server displays web page with order update confirmation/error msg
```

Behavioral model of employees fulfilling orders



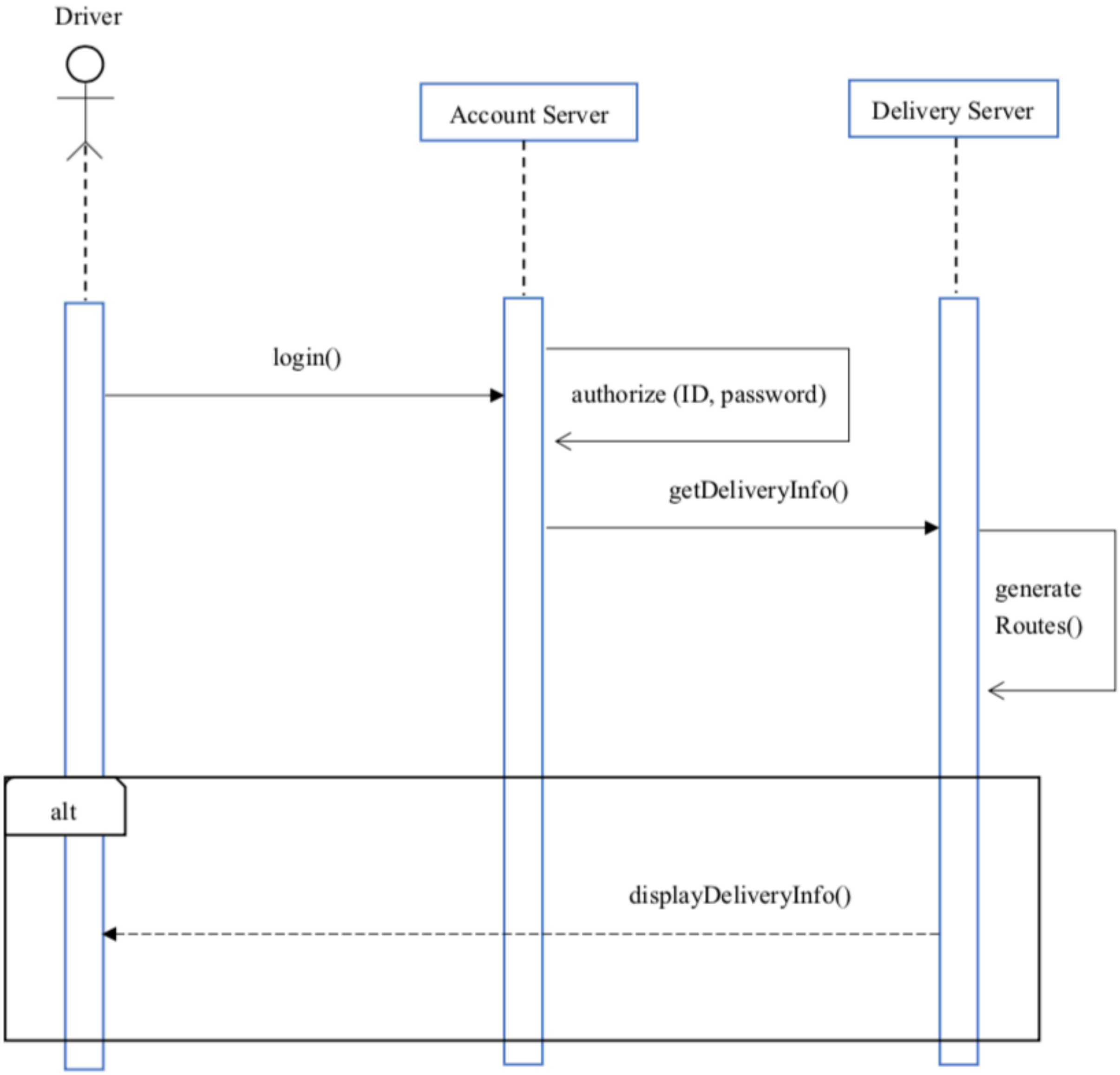
```
// An employee fulfills orders
login()
    authorize(ID, password)
    fulfillOrder()
    Order Server checkAvailability()
        if valid    employee picks up the goods on orders
        else        return error
        Delivery Server allocateDriver()
```

Behavioral model of managers checking reports



```
// A manager requests reports
login()
        authorize(ID, password)
        requestReports() from database
                Database generateReports()
        return reports
        displayReports()
```

Behavioral model of drivers making delivery



```
// A driver makes delivery
login()
        authorize(ID, password)
        getDeliveryInfo() from Delivery Server
                Delivery Server generateRoutes()
        return routes
        displayDeliveryInfo()
```