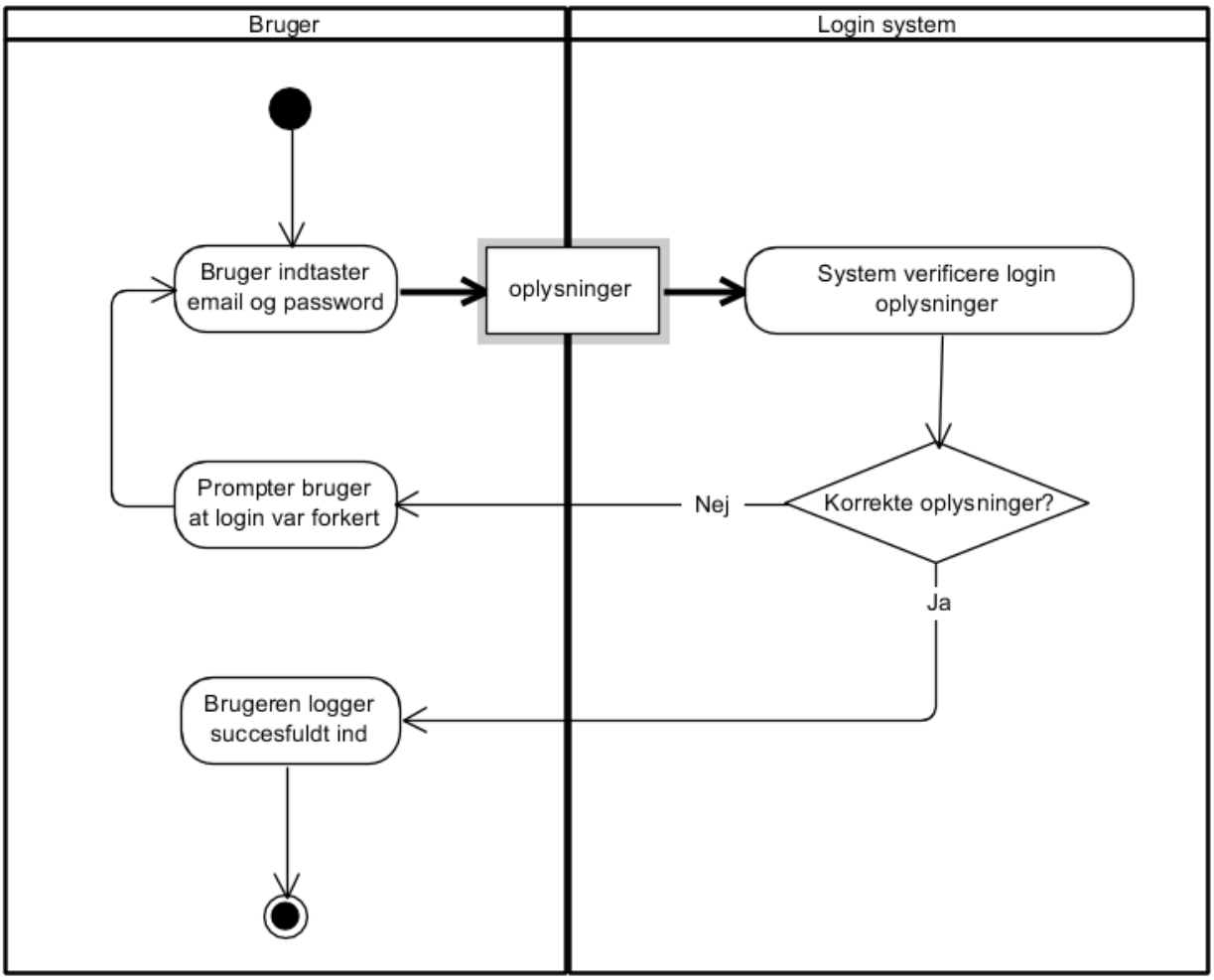
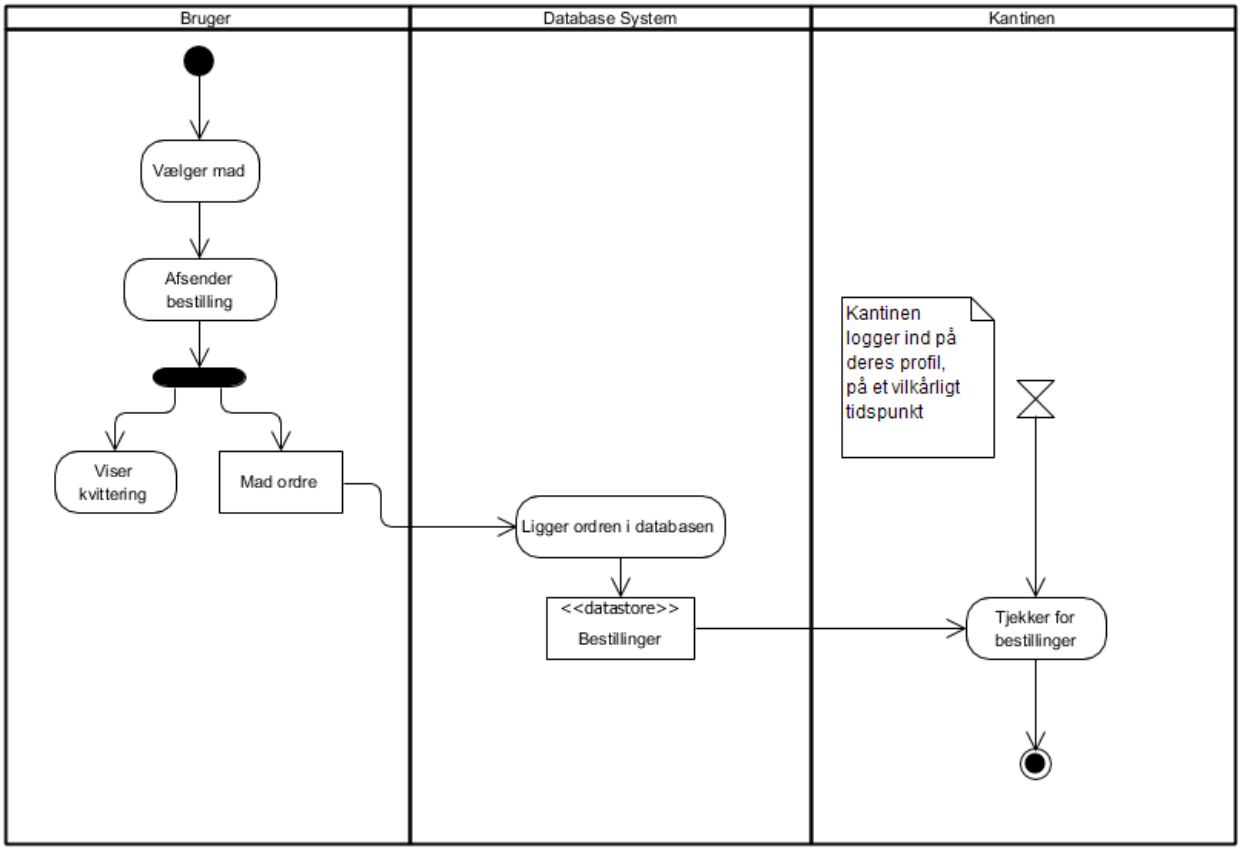
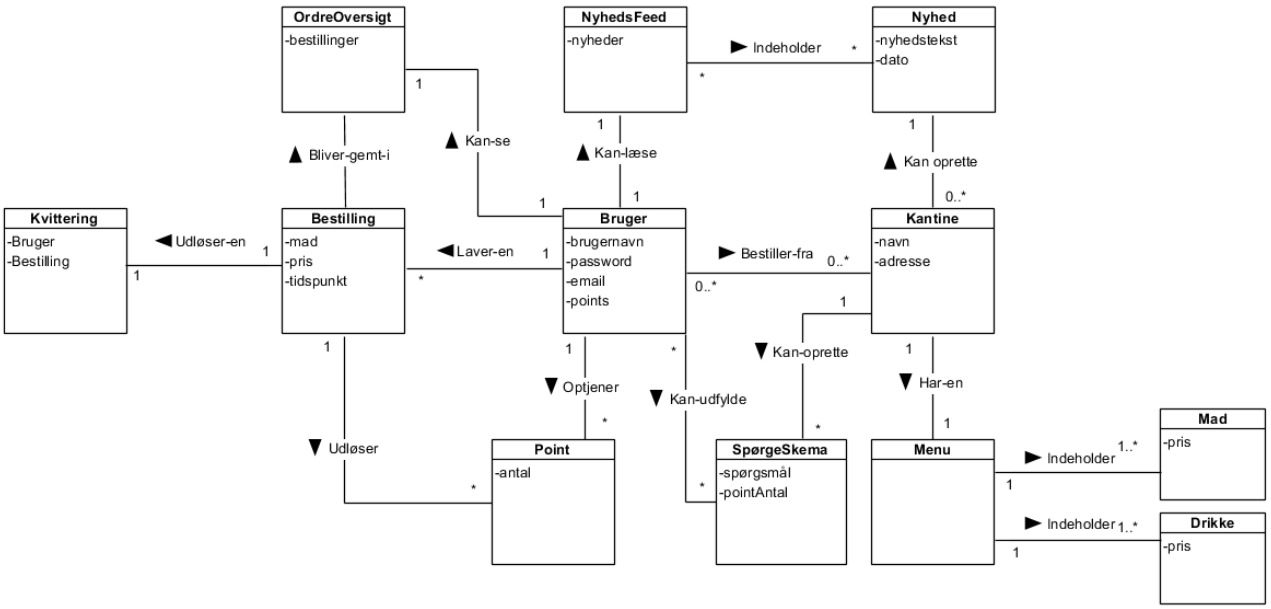
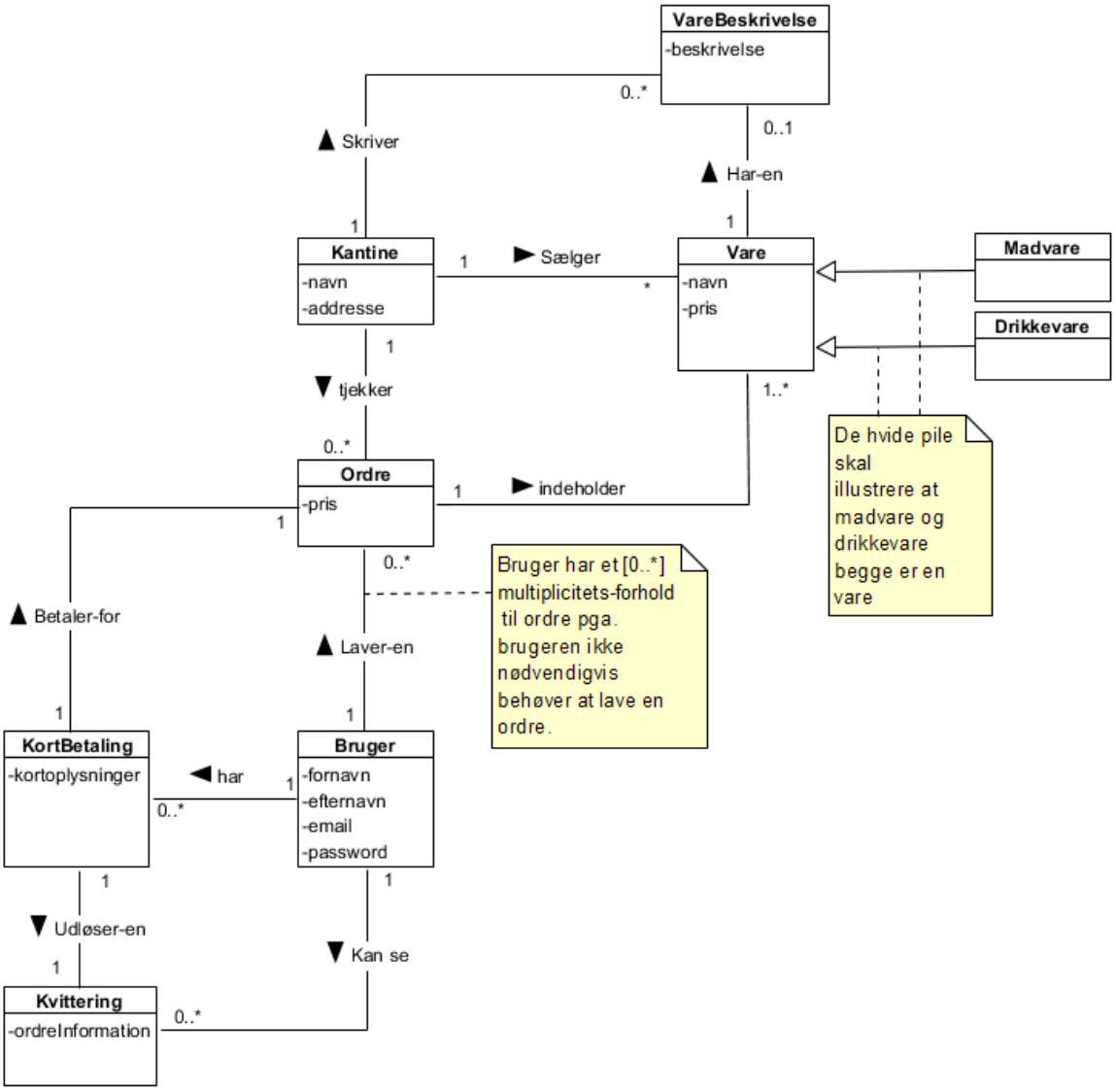
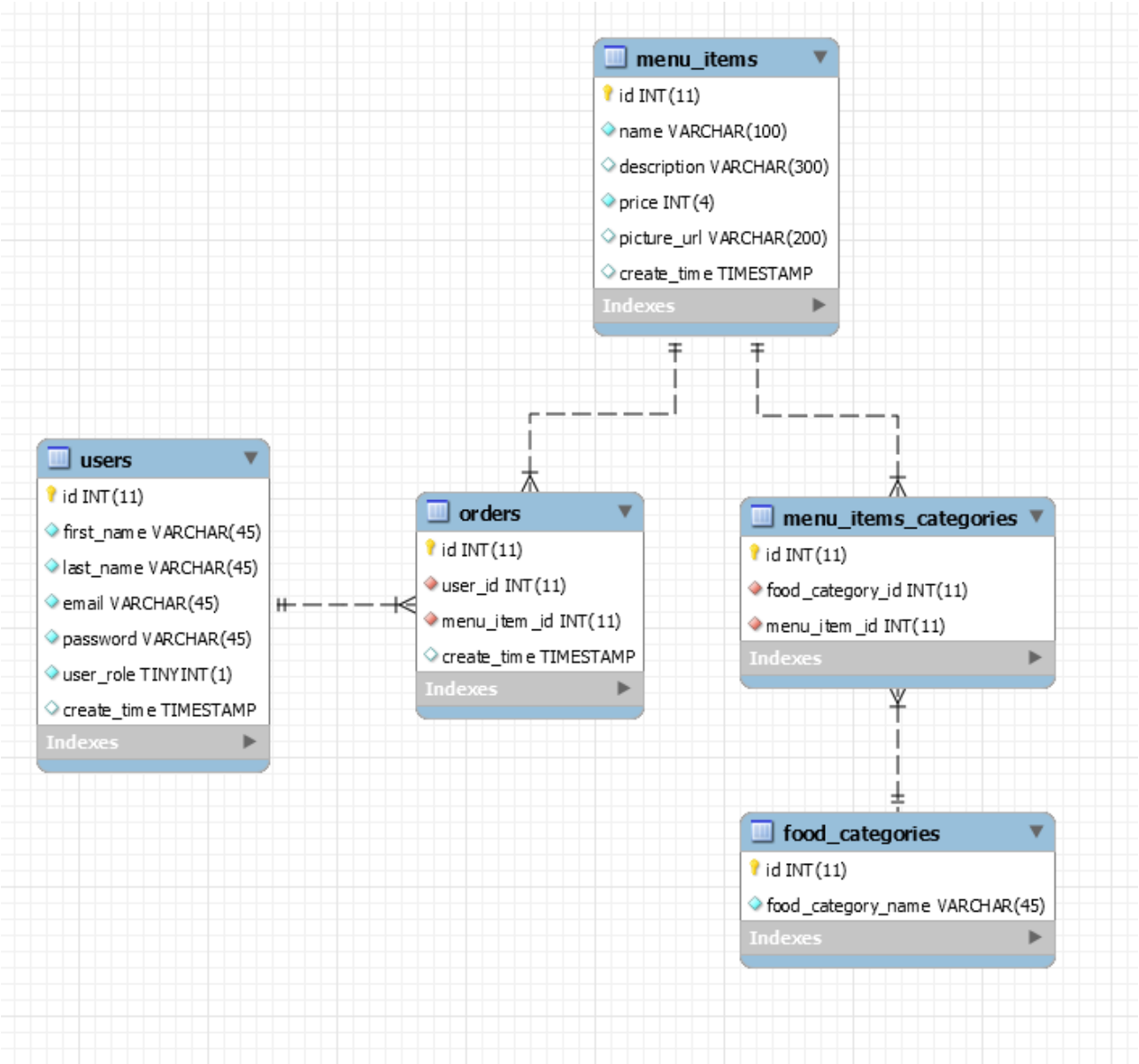
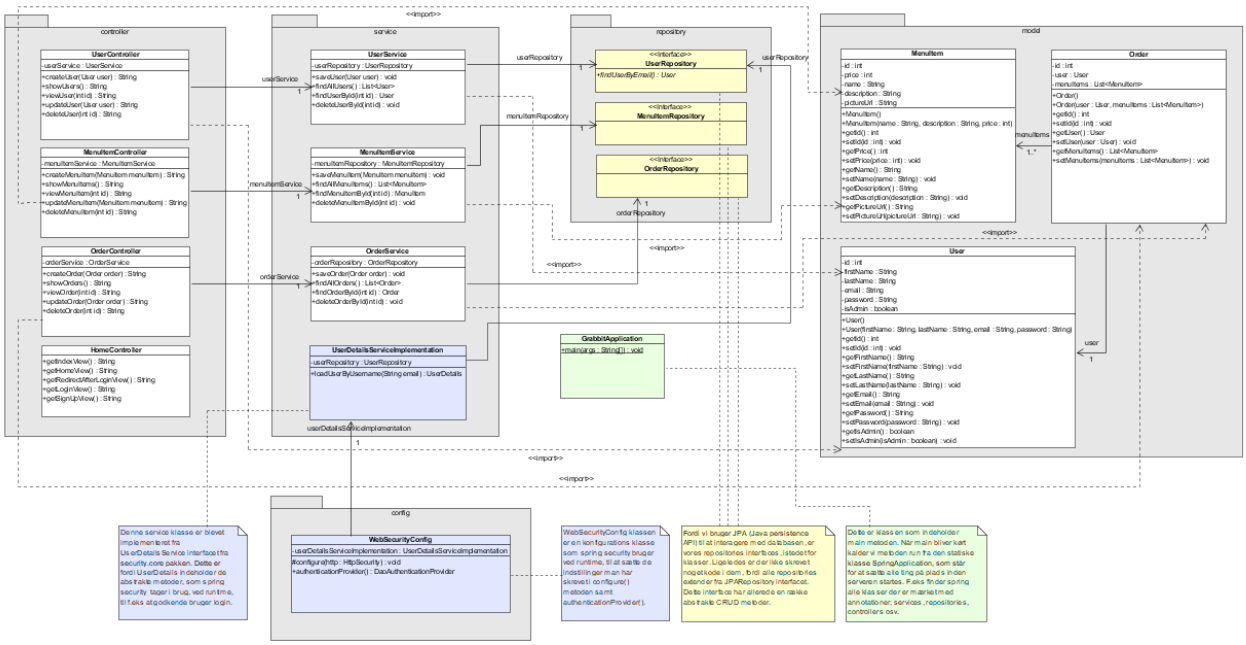

Bruger
Login system
Bruger indtaster email og password
oplysninger
System verificere login oplysninger
Korrekte oplysninger?
Nej
Ja
Prompter bruger at login var forkert
Brugeren logger succesfuldt ind

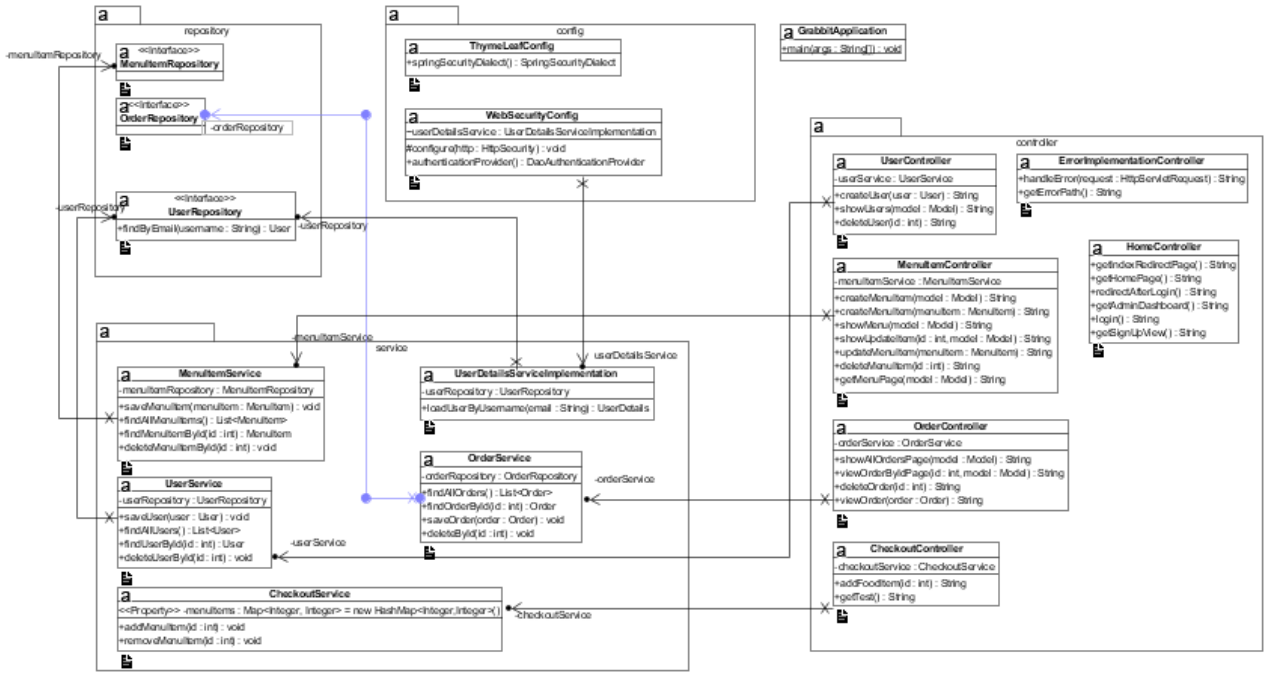

Bruger
Database System
Kantinen
Vælger mad
Afsender bestilling
Viser kvittering
Mad ordre
Ligger ordren i databasen
<<datastore>>
Bestillinger
Kantinen logger ind på deres profil, på et vilkårligt tidspunkt
Tjekker for bestillinger

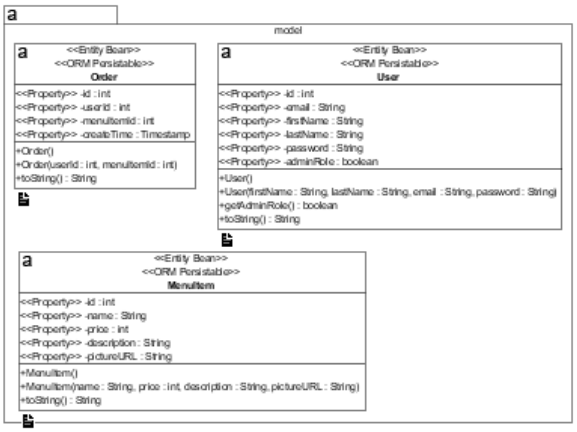

OrdreOversigt
-bestillinger
NyhedsFeed
-nyheder
Indeholder
Nyhed
-nyhedstekst
-dato
Bliver-gemt-i
Kan-se
Kan-læse
Kan oprette
Kvittering
-Bruger
-Bestilling
Udløser-en
Bestilling
-mad
-pris
-tidspunkt
Laver-en
Bruger
-brugernavn
-password
-email
-points
Bestiller-fra
Kantine
-navn
-adresse
Kan-oprette
Har-en
Optjener
Kan-udfylde
Udløser
Point
-antal
SpørgeSkema
-spørgsmål
-pointAntal
Menu
Indeholder
Mad
-pris
Indeholder
Drikke
-pris

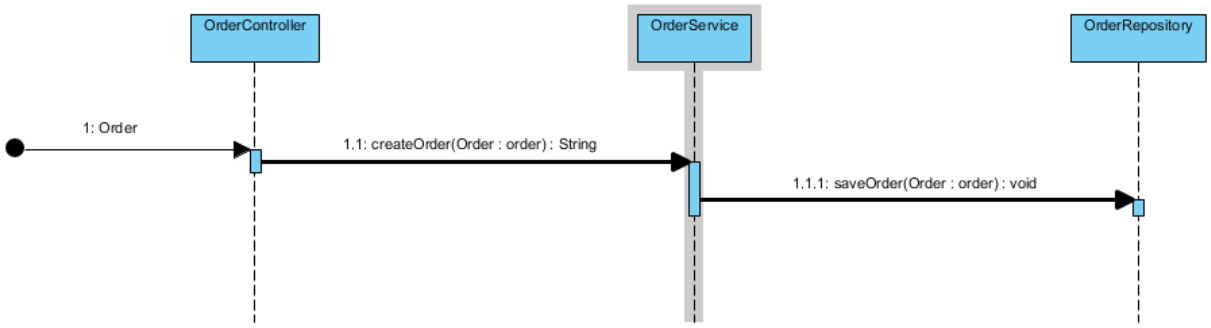

VareBeskrivelse
-beskrivelse
0..*
0..1
Skriver
Har-en
1
1
Kantine
-navn
-addresse
1
Sælger
*
Vare
-navn
-pris
Madvare
Drikkevare
1
tjekker
1..*
0..*
Ordre
-pris
1
indeholder
1
0..*
De hvide pile skal illustrere at madvare og drikkevare begge er en vare
Bruger har et [0..*] multiplicitets-forhold til ordre pga. brugeren ikke nødvendigvis behøver at lave en ordre.
Betaler-for
Laver-en
1
1
KortBetaling
-kortoplysninger
har
1
0..*
Bruger
-fornavn
-efternavn
-email
-password
1
1
Udløser-en
Kan se
1
1
Kvittering
-ordreInformation
0..*

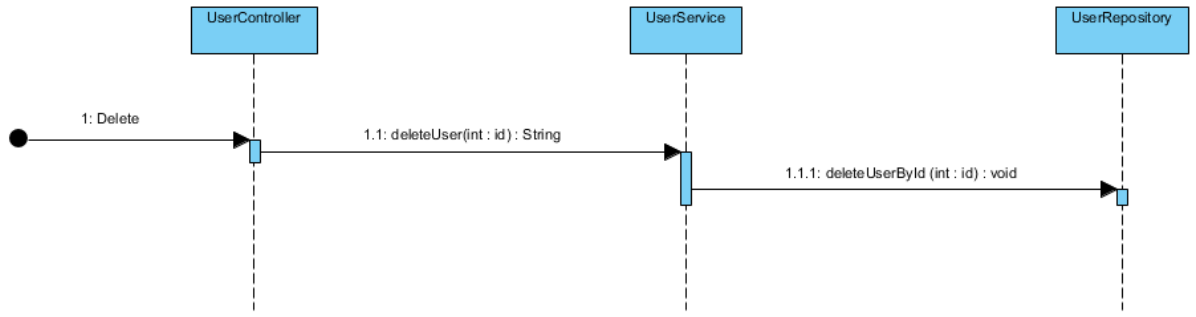

menu_items
id INT(11)
name VARCHAR(100)
description VARCHAR(300)
price INT(4)
picture_url VARCHAR(200)
create_time TIMESTAMP
Indexes
users
id INT(11)
first_name VARCHAR(45)
last_name VARCHAR(45)
email VARCHAR(45)
password VARCHAR(45)
user_role TINYINT(1)
create_time TIMESTAMP
Indexes
orders
id INT(11)
user_id INT(11)
menu_item_id INT(11)
create_time TIMESTAMP
Indexes
menu_items_categories
id INT(11)
food_category_id INT(11)
menu_item_id INT(11)
Indexes
food_categories
id INT(11)
food_category_name VARCHAR(45)
Indexes

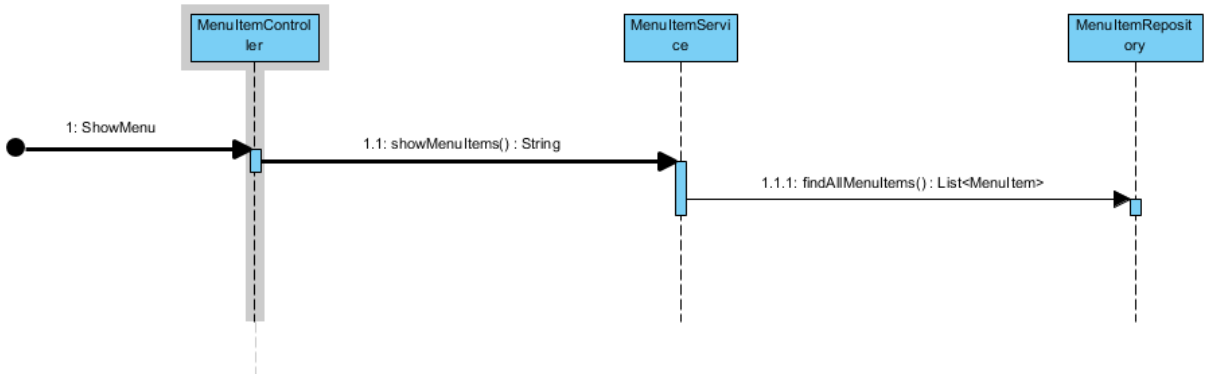

controller
UserController
-userService : UserService
+createUser(User user) : String
+showUsers() : String
+viewUser(int id) : String
+updateUser(User user) : String
+deleteUser(int id) : String
MenuItemController
-menuItemService : MenuItemService
+createMenuItem(MenuItem menuItem) : String
+showMenuItems() : String
+viewMenuItem(int id) : String
+updateMenuItem(MenuItem menuItem) : String
+deleteMenuItem(int id) : String
OrderController
-orderService : OrderService
+createOrder(Order order) : String
+showOrders() : String
+viewOrder(int id) : String
+updateOrder(Order order) : String
+deleteOrder(int id) : String
HomeController
+getIndexView() : String
+getHomeView() : String
+getRedirectAfterLoginView() : String
+getLoginView() : String
+getSignUpView() : String
<<import>>
service
UserService
-userRepository : UserRepository
+saveUser(User user) : void
+findAllUsers() : List<User>
+findUserById(int id) : User
+deleteUserById(int id) : void
MenuItemService
-menuItemRepository : MenuItemRepository
+saveMenuItem(MenuItem menuItem) : void
+findAllMenuItems() : List<MenuItem>
+findMenuItemById(int id) : MenuItem
+deleteMenuItemById(int id) : void
OrderService
-orderRepository : OrderRepository
+saveOrder(Order order) : void
+findAllOrders() : List<Order>
+findOrderById(int id) : Order
+deleteOrderById(int id) : void
UserDetailsServiceImplementation
-userRepository : UserRepository
+loadUserByUsername(String email) : UserDetails
userDetailsServiceImplementation
userService
menuItemService
orderService
repository
<<interface>>
UserRepository
+findUserByEmail() : User
<<interface>>
MenuItemRepository
<<interface>>
OrderRepository
orderRepository
usrRepository
menuItemRepository
GrabbitApplication
+main(args : String[]) : void
model
MenuItem
-id : int
-price : int
-name : String
-description : String
-pictureUrl : String
+MenuItem()
+MenuItem(name : String, description : String, price : int)
+getId() : int
+setId(id : int) : void
+getPrice() : int
+setPrice(price : int) : void
+getName() : String
+setName(name : String) : void
+getDescription() : String
+setDescription(description : String) : void
+getPictureUrl() : String
+setPictureUrl(pictureUrl : String) : void
Order
-id : int
-user : User
-menuItems : List<MenuItem>
+Order()
+Order(user : User, menuItems : List<MenuItem>)
+getId() : int
+setId(id : int) : void
+getUser() : User
+setUser(user : User) : void
+getMenuItems() : List<MenuItem>
+setMenuItems(menuItems : List<MenuItem>) : void
menuItems
1..*
<<import>>
User
-id : int
-firstName : String
-lastName : String
-email : String
-password : String
-isAdmin : boolean
+User()
+User(firstName : String, lastName : String, email : String, password : String)
+getId() : int
+setId(id : int) : void
+getFirstName() : String
+setFirstName(firstName : String) : void
+getLastName() : String
+setLastName(lastName : String) : void
+getEmail() : String
+setEmail(email : String) : void
+getPassword() : String
+setPassword(password : String) : void
+getIsAdmin() : boolean
+setIsAdmin(isAdmin : boolean) : void
user
1
<<import>>
<<import>>
<<import>>
<<import>>
config
WebSecurityConfig
-userDetailsServiceImplementation : UserDetailsServiceImplementation
#configure(HttpHttpSecurity) : void
+authenticationProvider() : DaoAuthenticationProvider
Denne service klasse er blevet implementeret fra UserDetailsService interface fra security.core pakken. Dette er fordi UserDetails indeholder de abstrakte metoder, som spring security tager i brug, ved runtime, til f.eks at godkende bruger login.
WebSecurityConfig klassen er en konfigurations klasse som spring security bruger ved runtime, til at sætte de indstillinger man har skrevet i configure() metoden samt authenticationProvider().
Fordi vi bruger JPA (Java persistance API) til at interagere med databasen, er vores repositories interfaces, i stedet for klasser. Ligeledes er der ikke skrevet noget kode i dem, fordi alle repositories extender fra JPARepository interface. Dette interface har allerede en række abstrakte CRUD metoder.
Dette er klassen som indeholder main metoden. Når main bliver kørt kalder vi metoden run fra den statiske klasse SpringApplication, som står for at sætte alle ting på plads inden serveren starter. F.eks finder spring alle klasser der er mærket med annotationer; services, repositories, controllers osv.

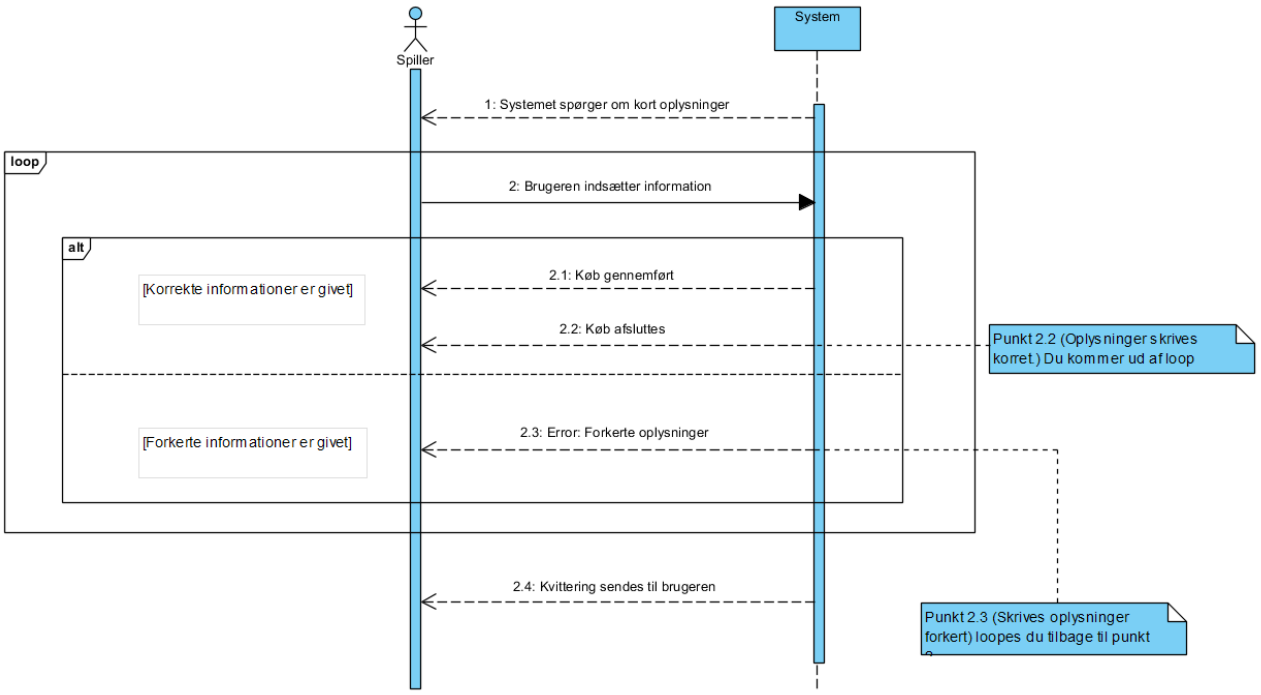

repository
<<Interface>>
MenuItemRepository
<<Interface>>
OrderRepository
-orderRepository
<<Interface>>
UserRepository
+findByEmail(username : String) : User
-menuItemRepository
-userRepository
-userRepository
config
ThymeLeafConfig
+springSecurityDialect() : SpringSecurityDialect
WebSecurityConfig
-userDetailsService : UserDetailsServiceImplementation
#configure(http : HttpSecurity) : void
+authenticationProvider() : DaoAuthenticationProvider
GrabbitApplication
+main(args : String[]) : void
controller
UserController
-userService : UserService
+createUser(user : User) : String
+showUsers(model : Model) : String
+deleteUser(id : int) : String
ErrorImplementationController
+handleError(request : HttpServletRequest) : String
+getErrorPath() : String
MenuItemController
-menuItemService : MenuItemService
+createMenuItem(model : Model) : String
+createMenuItem(menuItem : MenuItem) : String
+showMenu(model : Model) : String
+showUpdateItem(id : int, model : Model) : String
+updateMenuItem(menuItem : MenuItem) : String
+deleteMenuItem(id : int) : String
+getMenuPage(model : Model) : String
HomeController
+getIndexRedirectPage() : String
+getHomePage() : String
+redirectAfterLogin() : String
+getAdminDashboard() : String
+login() : String
+getSignUpView() : String
OrderController
-orderService : OrderService
+showAllOrdersPage(model : Model) : String
+viewOrderByIdPage(id : int, model : Model) : String
+deleteOrder(id : int) : String
+viewOrder(order : Order) : String
CheckoutController
-checkoutService : CheckoutService
+addFoodItem(id : int) : String
+getTest() : String
-menuItemService
service
-userDetailsService
MenuItemService
-menuItemRepository : MenuItemRepository
+saveMenuItem(menuItem : MenuItem) : void
+findAllMenuItems() : List<MenuItem>
+findMenuItemById(id : int) : MenuItem
+deleteMenuItemById(id : int) : void
UserDetailsServiceImplementation
-userRepository : UserRepository
+loadUserByUsername(email : String) : UserDetails
OrderService
-orderRepository : OrderRepository
+findAllOrders() : List<Order>
+findOrderById(id : int) : Order
+saveOrder(order : Order) : void
+deleteById(id : int) : void
-orderService
UserService
-userRepository : UserRepository
+saveUser(user : User) : void
+findAllUsers() : List<User>
+findUserById(id : int) : User
+deleteUserById(id : int) : void
-userService
CheckoutService
<<Property>> -menuItems : Map<Integer, Integer> = new HashMap<Integer,Integer>()
+addMenuItem(id : int) : void
+removeMenuItem(id : int) : void
-checkoutService

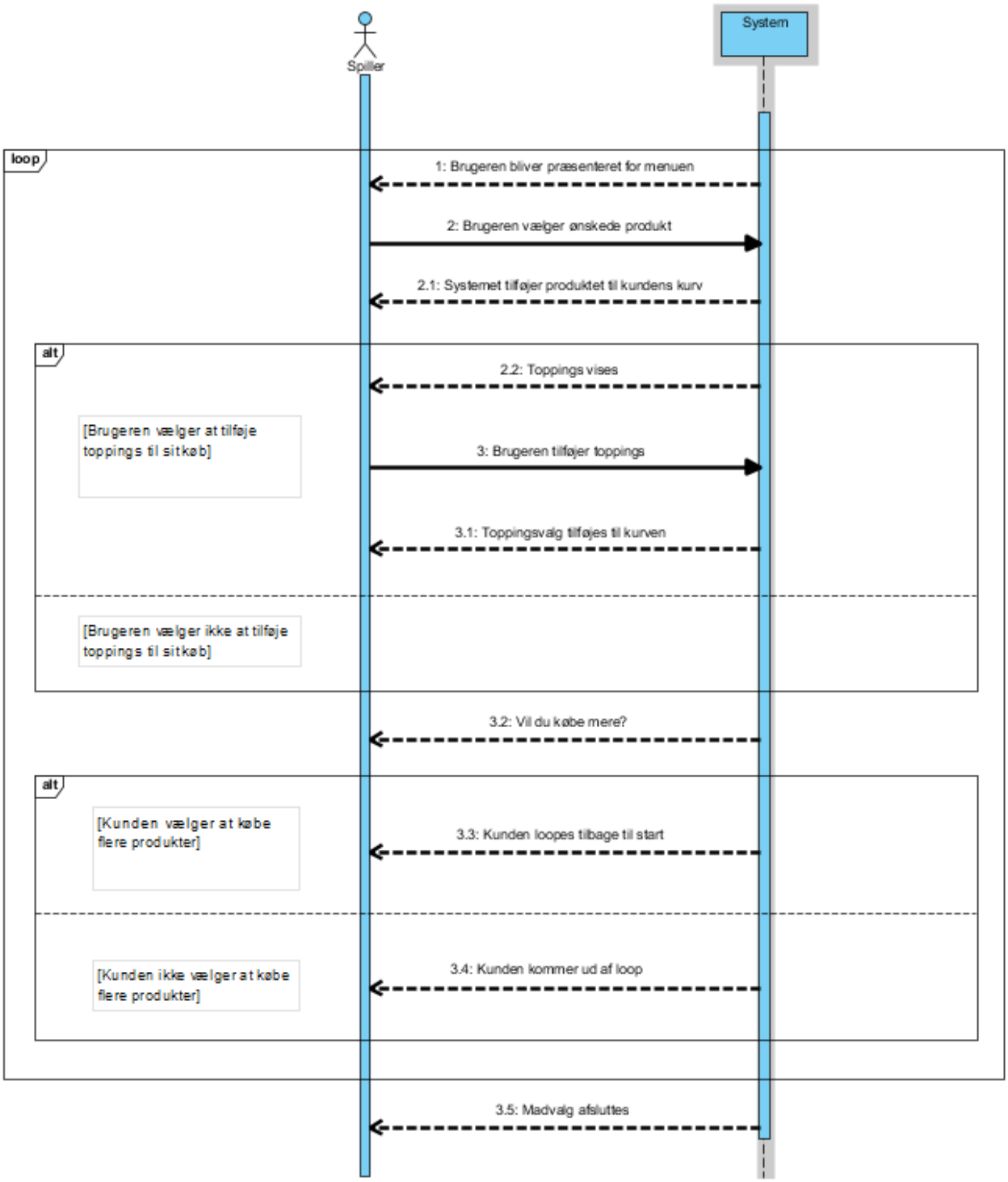

model
<<Entity Bean>>
<<ORM Persistable>>
Order
<<Property>> -id : int
<<Property>> -userId : int
<<Property>> -menuItemId : int
<<Property>> -createTime : Timestamp
+Order()
+Order(userId : int, menuItemId : int)
+toString() : String
<<Entity Bean>>
<<ORM Persistable>>
User
<<Property>> -id : int
<<Property>> -email : String
<<Property>> -firstName : String
<<Property>> -lastName : String
<<Property>> -password : String
<<Property>> -adminRole : boolean
+User()
+User(firstName : String, lastName : String, email : String, password : String)
+getAdminRole() : boolean
+toString() : String
<<Entity Bean>>
<<ORM Persistable>>
MenuItem
<<Property>> -id : int
<<Property>> -name : String
<<Property>> -price : int
<<Property>> -description : String
<<Property>> -pictureURL : String
+MenuItem()
+MenuItem(name : String, price : int, description : String, pictureURL : String)
+toString() : String

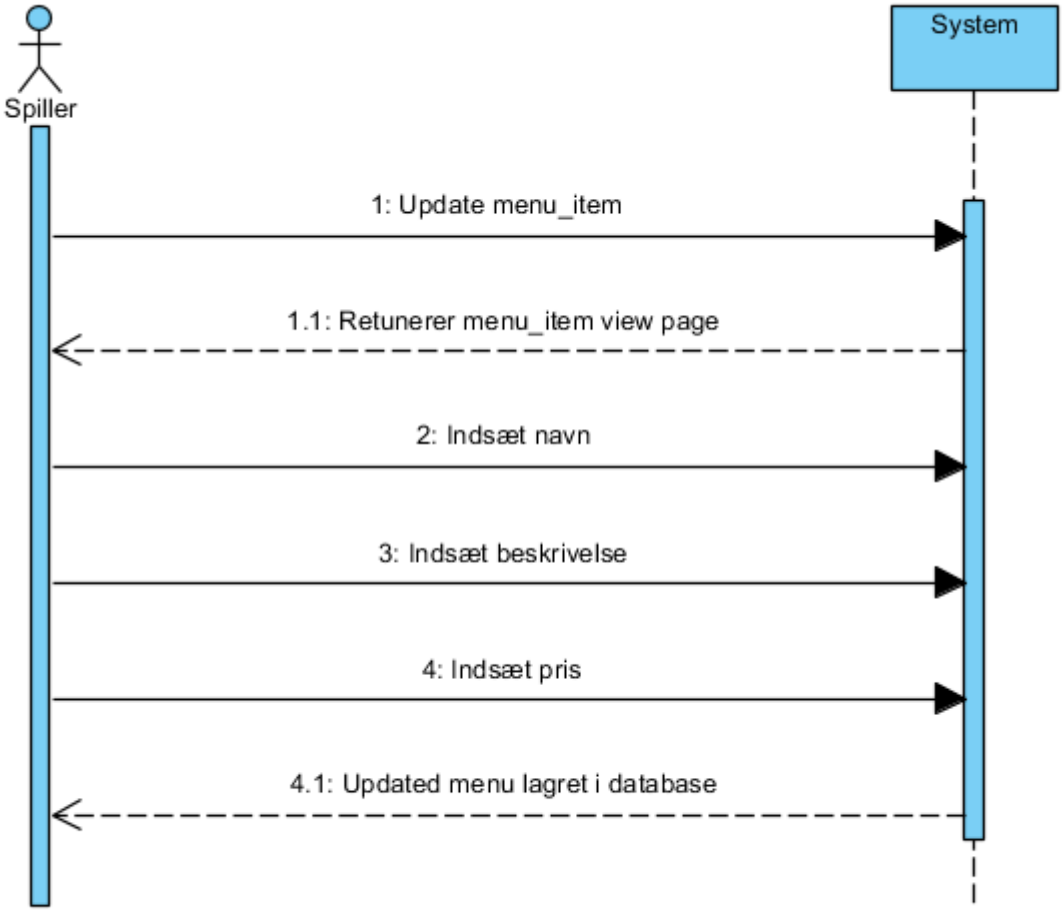

OrderController
OrderService
OrderRepository
1: Order
1.1: createOrder(Order : order) : String
1.1.1: saveOrder(Order : order) : void

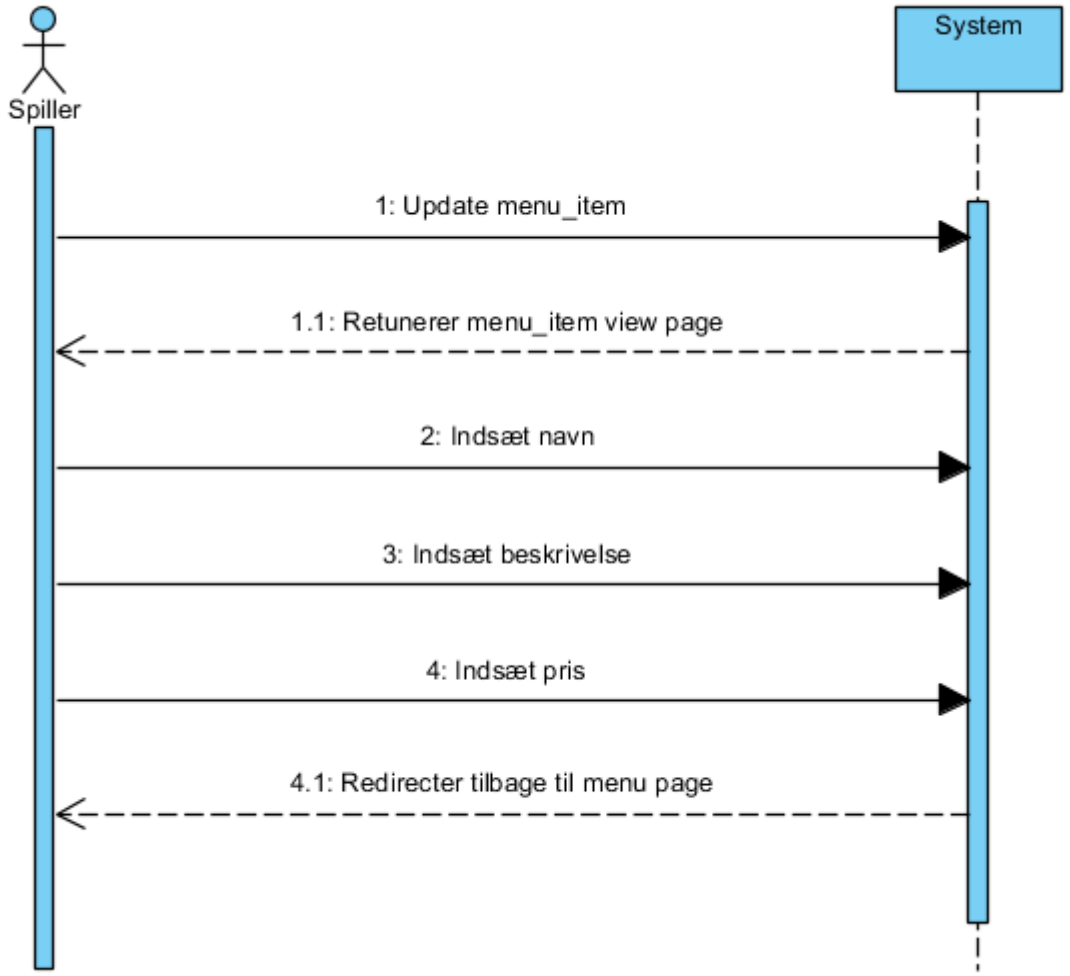

UserController
UserService
UserRepository
1: Delete
1.1: deleteUser(int : id) : String
1.1.1: deleteUserById (int : id) : void

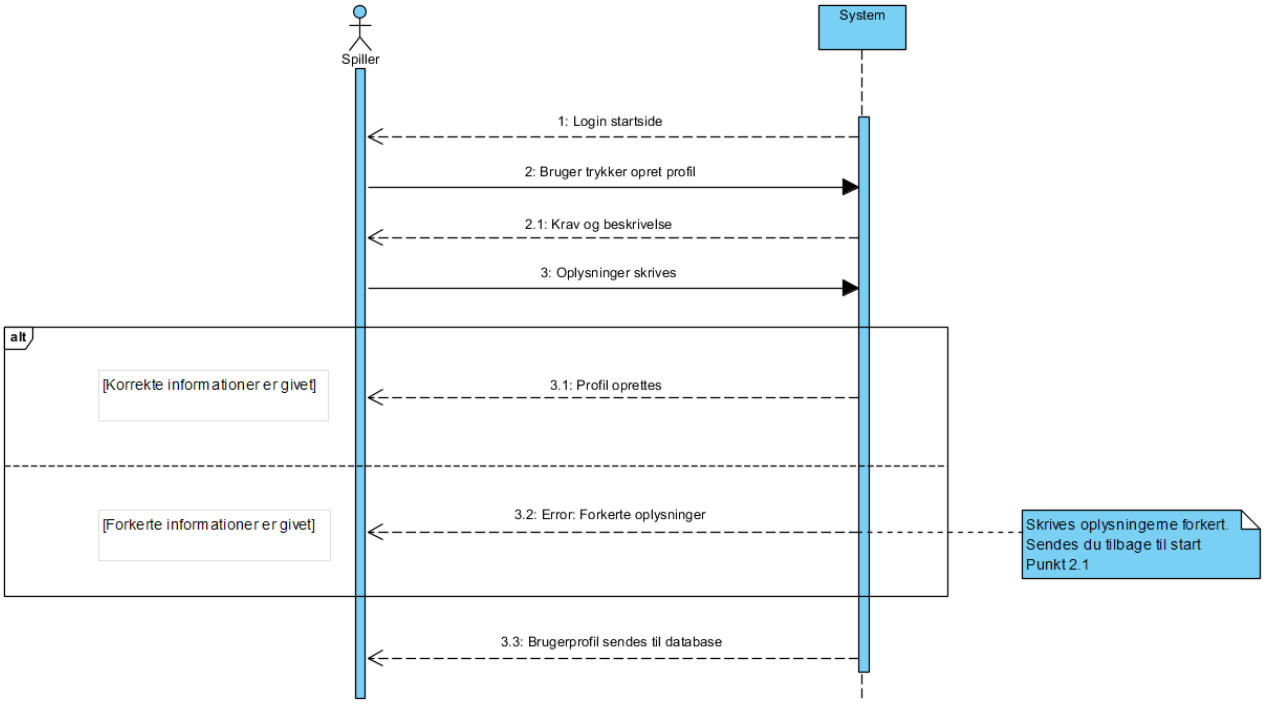

MenuItemControl
ler
MenuItemServi
ce
MenuItemReposit
ory
1: ShowMenu
1.1: showMenuItems() : String
1.1.1: findAllMenuItems() : List<MenuItem>

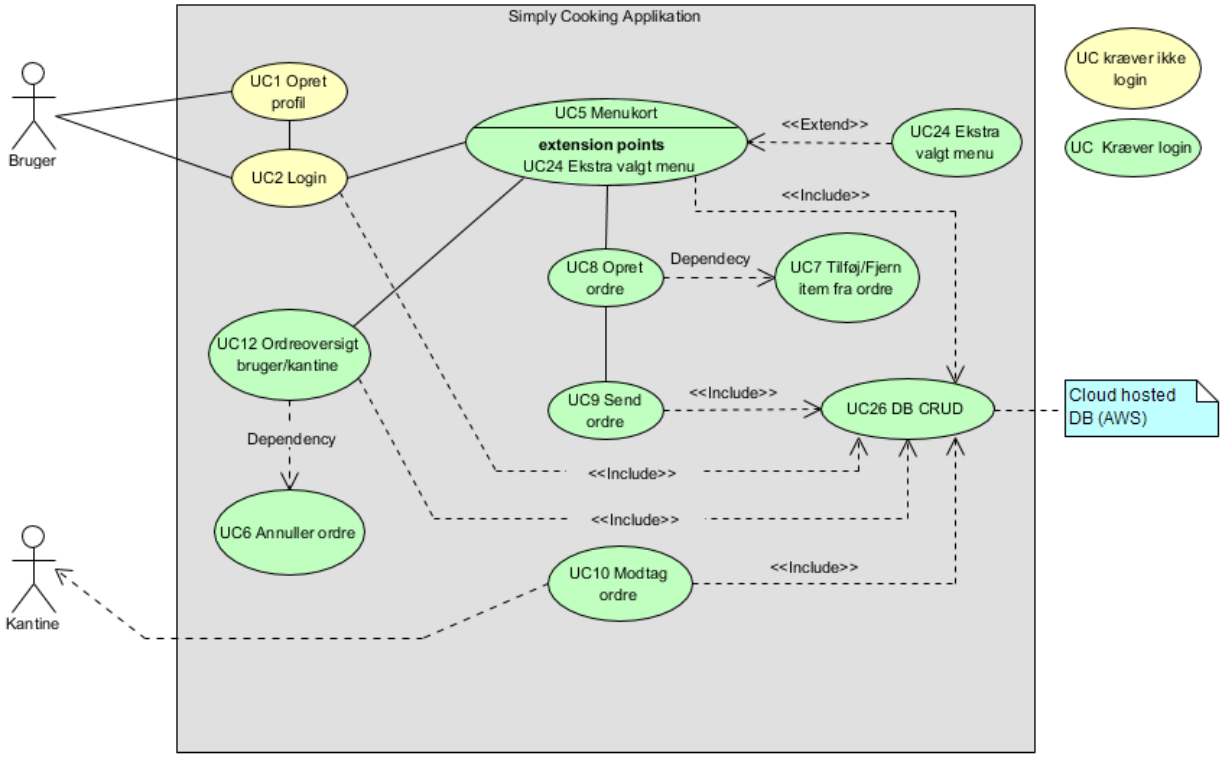

Spiller
System
1: Systemet spørger om kort oplysninger
loop
2: Brugeren indsætter information
alt
[Korrekte informationer er givet]
2.1: Køb gennemført
2.2: Køb afsluttes
Punkt 2.2 (Oplysninger skrives korret.) Du kommer ud af loop
[Forkerte informationer er givet]
2.3: Error: Forkerte oplysninger
2.4: Kvittering sendes til brugeren
Punkt 2.3 (Skrives oplysninger forkert) loopes du tilbage til punkt


Spiller
System
loop
1: Brugeren bliver præsenteret for menuen
2: Brugeren vælger ønskede produkt
2.1: Systemet tilføjer produktet til kundens kurv
alt
2.2: Toppings vises
[Brugeren vælger at tilføje toppings til sit køb]
3: Brugeren tilføjer toppings
3.1: Toppingsvalg tilføjes til kurven
[Brugeren vælger ikke at tilføje toppings til sit køb]
3.2: Vil du købe mere?
alt
[Kunden vælger at købe flere produkter]
3.3: Kunden loopes tilbage til start
[Kunden ikke vælger at købe flere produkter]
3.4: Kunden kommer ud af loop
3.5: Madvalg afsluttes


Spiller
System
1: Update menu_item
1.1: Retunerer menu_item view page
2: Indsæt navn
3: Indsæt beskrivelse
4: Indsæt pris
4.1: Updated menu lagret i database


Spiller
System
1: Update menu_item
1.1: Retunerer menu_item view page
2: Indsæt navn
3: Indsæt beskrivelse
4: Indsæt pris
4.1: Redirecter tilbage til menu page


Spiller
System
1: Login startside
2: Bruger trykker opret profil
2.1: Krav og beskrivelse
3: Oplysninger skrives
alt
[Korrekte informationer er givet]
3.1: Profil oprettes
[Forkerte informationer er givet]
3.2: Error: Forkerte oplysninger
Skrives oplysningerne forkert. Sendes du tilbage til start Punkt 2.1
3.3: Brugerprofil sendes til database


Simply Cooking Applikation
Bruger
UC1 Opret profil
UC2 Login
UC5 Menukort
extension points
UC24 Ekstra valgt menu
<<Extend>>
UC24 Ekstra valgt menu
<<Include>>
UC8 Opret ordre
Dependecy
UC7 Tilføj/Fjern item fra ordre
UC12 Ordreoversigt bruger/kantine
UC9 Send ordre
<<Include>>
UC26 DB CRUD
Cloud hosted DB (AWS)
Dependency
<<Include>>
UC6 Annuller ordre
<<Include>>
UC10 Modtag ordre
<<Include>>
Kantine
UC kræver ikke login
UC Kræver login