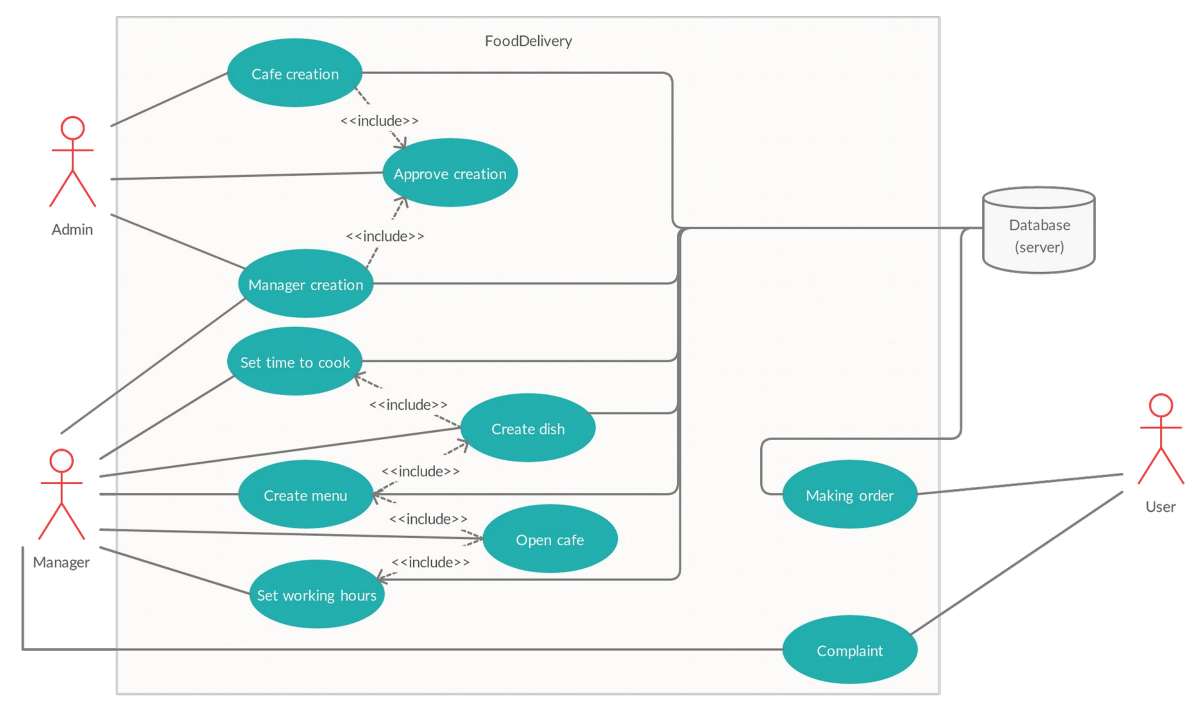
Roles:

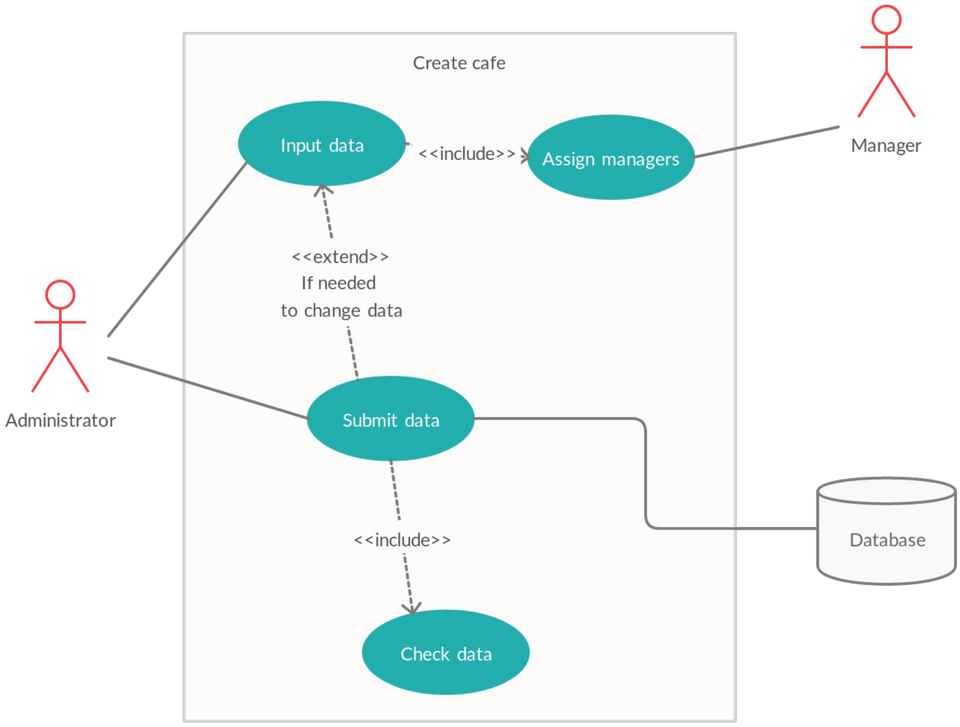
Administrator	Person that is responsible for creating and administrating cafes(asigning managers, changing contact data and info).
Manager	Person, that fills in info about canteen(menu, if canteen is open or closed, etc.) and responses to the orders.
User	Person which orders food.

Use cases:

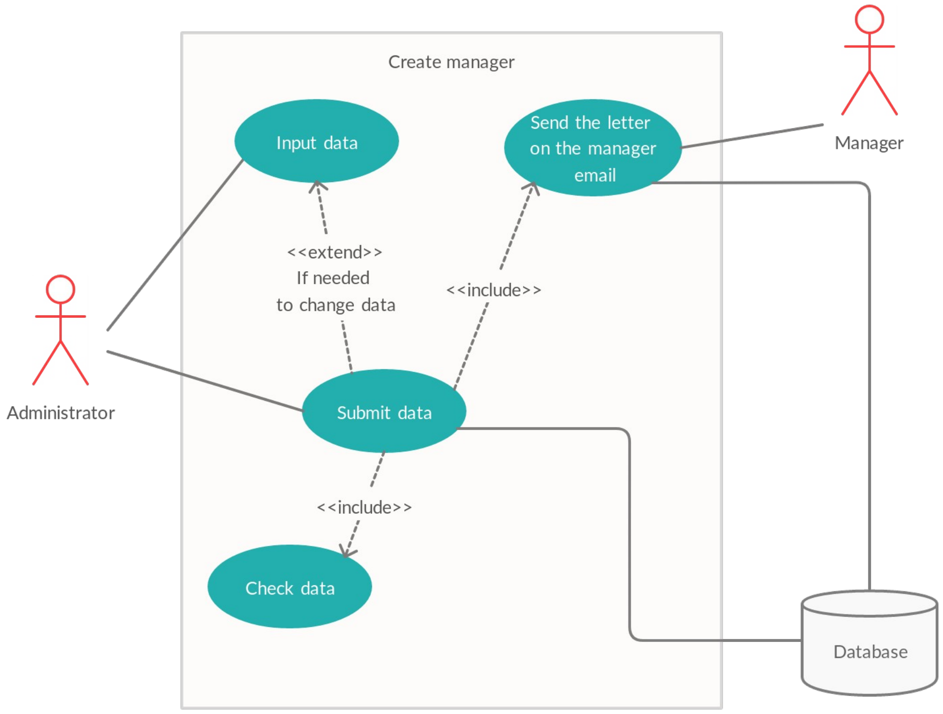


Use case name	Cafes creation use case
Unique use case ID:	CA
Primary actor(s):	Administrator of the canteens
Secondary actor(s):	Managers
Brief description:	Admin creates a new cafe/canteen, and sets up all parameters for it, he also assigns manager to that canteen.
Preconditions:	Newly built, or cafe not entered at the system, manager, earlier created by admin.
Flow of events:	First, administrator enters all the data needed for the cafe, as: Phone number, email, name, address, information about cafe,

	manager(information about him is automatically displayed when he is chosed). Then he click the create button, and get all data displayed to check, he can approve, or return back to changing, after this step he click approve button, and the response goes to server, cafe wouldn’t be seen by users, until manager would “open” it, after he fills all the needed data.
Postconditions:	New cafe added to the list of administrator, it is seen to the manager, that is responsible for this cafeterium, as well as for users, if this canteen is open. The cafe data is stationed in DB on backend.
Priority:	Top priority, because the other way there would be no canteens to serve the users.
Assumptions:	If admin would enter data in incorrect format, he would need to reenter it.
Nonbehavioral requirements:	Performance to a website response of the whole system is limited to three seconds, in this time DB should be updated and frontend must respond. Only user with the rights of admin could create cafeteriums.
Source	Such process gives possibility to make app scalable in a distant future(United Interviews, Scalability)

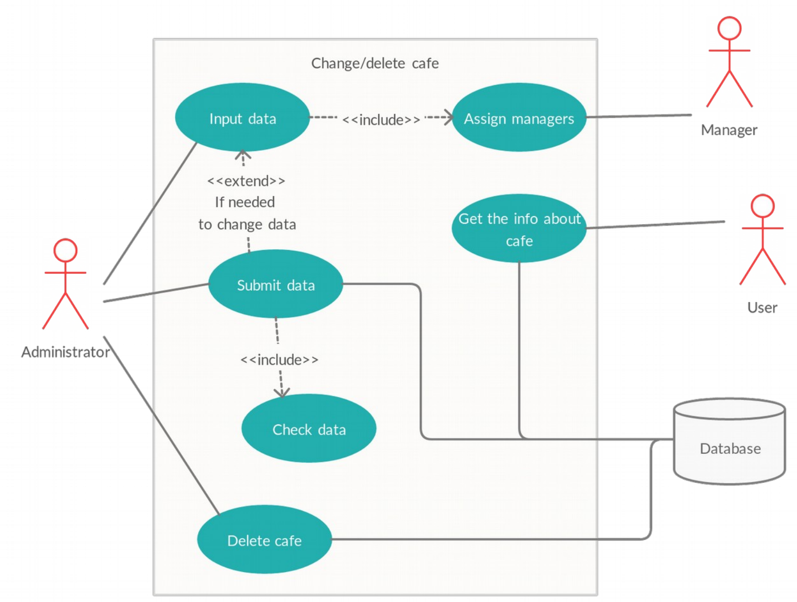


Use case name	Manager creation use case
Unique use case ID:	CA
Primary actor(s):	Administrator of the canteens
Secondary actor(s):	Managers
Brief description:	Admin creates a new manager of the canteen and sets up all parameters for him.
Preconditions:	A real human person with email and phone number, that will be used for a work.
Flow of events:	First, administrator enters all the data needed for the manager, as: Phone number, email, name,information about cafe he is responsible(if needed). Then he click the create button, and get all data displayed to check, he can approve, or return back to changing, after this step he click approve button, and the response goes to server. Parallely data about account (temporary autogenerated password,login) sent to the manager.
Postconditions:	New manager is added to the list of managers, he is now possible to assign to a canteen, if he is assign, users will see his contact data.
Priority:	Top priority, because the other way there would be no management to fill in the data about the canteens.
Assumptions:	Admin shouldn’t know managers password, for the privacy. If admin would enter data in incorrect format, he would need to reenter it. The letter would be sent automatically by the server.
Non Behavioral requirements:	Performance to a website response of the whole system is limited to three seconds, in this time DB should be updated and frontend must respond. Time of email sending is limited up to ten minutes(according to the Sofia interview). Password must be secure(created using pseudo-random generators), and should be stored in DB using hashing with salt, to prevent major breaking into the system.
Source	United Interviews, Who updates dishes

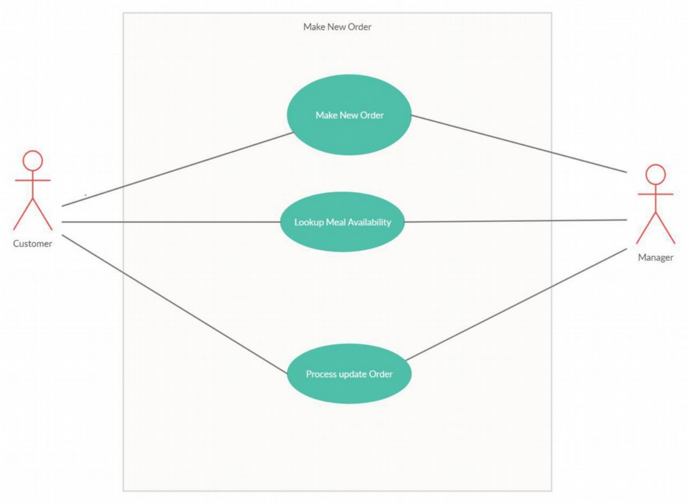


Use case name	Cafes change/delete use case
Unique use case ID:	CA
Primary actor(s):	Administrator of the canteens
Secondary actor(s):	Managers, Users
Brief description:	Admin change or delete cafe/canteen, he is free to change everything he wants, or delete cafe .
Preconditions:	Already existing cafe with manager and filled data.
Flow of events:	First, administrator is free to choose, ether he wants to delete or change cafe , if he wants to change cafe, he need to enter all needed data, and press submit, if he wants delete, he needs to press delete, ether way, he need to check his decision, and approve it, after it the changes will be written in DB, and both managers and users would see them.
Postconditions:	Ether changed cafe is submitted in DB, or cafe is deleted,
Priority:	Top priority, because the other way there admins couldn't change the errors about their cafes.

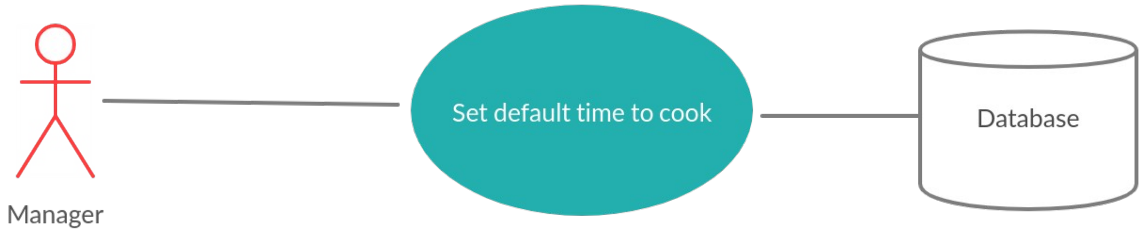
Nonbehavioral requirements:	Performance to a website response of the whole system is limited to three seconds, in this time DB should be updated and frontend must respond. Only user with the rights of admin could change/delete cafeteriums.
Source	Such process gives possibility to make app scalable in a distant future(United Interviews, Scalability)



User Case Name	Make Order
Primary Actor(s)	Customer
Brief description	Customer makes an order through the application
Preconditions	The meals/dishes to be ordered are available
Postconditions	Order is completed
Flow of events	User logs in with account email and password. Goes through the available menu, selects the desired meal and clicks make order button.
Sources	Interviews, User making order

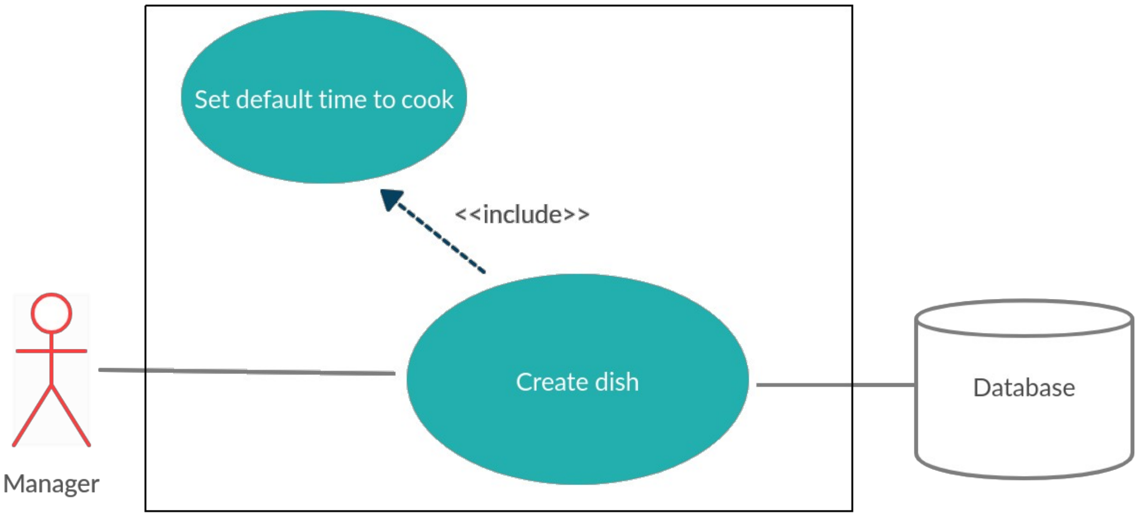


User Case Name	Set time to cook
Primary Actor(s)	Manager
Brief description	Default time to cook a standard dish
Preconditions	No default time is set. Or the default time is about to be changed.
Flow of events	Manager sets new or changes default time to cook
Postconditions	The default time is saved
Priority	High
Assumptions	Customers will see the time when their order is ready with respect to time every dish takes
Sources	United Interviews, Time of order is ready

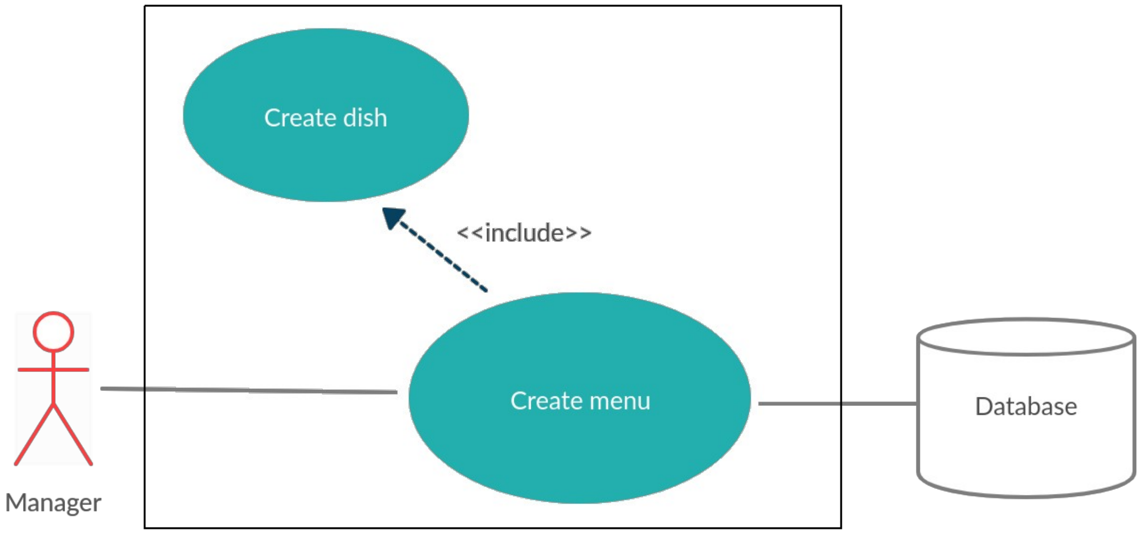


User Case Name	Create dish
Primary Actor(s)	Manager
Brief description	Manager creates a dish
Preconditions	There’s a need to add a dish to menu

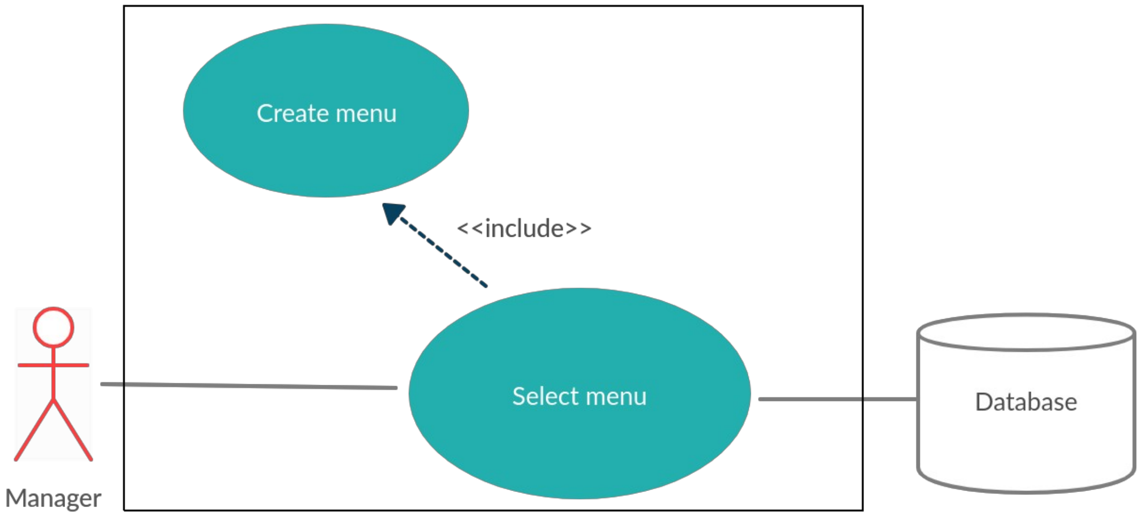
Flow of events	The form to create a new dish comes with time to cook filled, which is equal to default time. To create a dish, a manager must fill title, description and image. Also, it’s possible to set time needed to prepare the dish, if it’s different from default.
Postconditions	Time to cook, description, title and image are filled
Priority	High
Assumptions	A manager has description and the dish picture ready
Sources	United Interviews, Who updates dishes



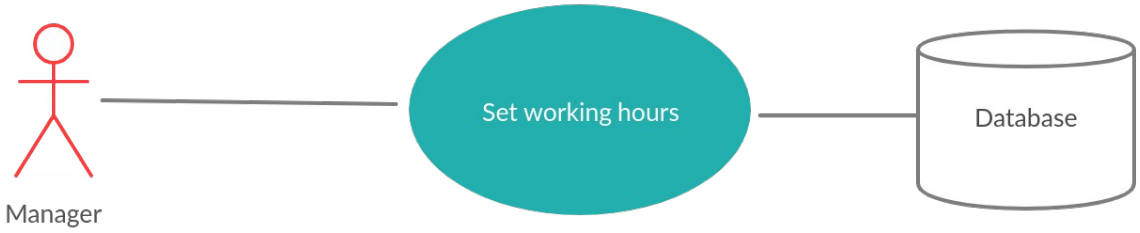
User Case Name	Create menu
Primary Actor(s)	Manager
Brief description	Creating a menu with a set of dishes
Preconditions	The dishes that is to be selected are created and saved
Flow of events	A manager selects a set of dishes for a menu that is to be available in some days (possibly weekdays). The new menu must be saved with a title.
Postconditions	New menu is created
Priority	High
Sources	United Interviews, Predefined menus for managers



User Case Name	Select menu
Primary Actor(s)	Manager
Brief description	A manager sets available menu for each weekday or for a particular date
Preconditions	The menus are created
Flow of events	It is possible for a manager to select a menu for every weekday. For example, a selected menu for Monday will be used automatically every Monday. It is possible to leave some weekdays empty. It is possible to select a menu for a particular date, for example, October 21. If a weekday menu could be used for that day too, particular date menu has higher priority.
Postconditions	The menu is selected
Priority	High
Assumptions	If all weekdays aren't filled with templates, the app won't work since 00:00 until one of the menus is selected
Sources	United Interviews, Predefined menus for managers

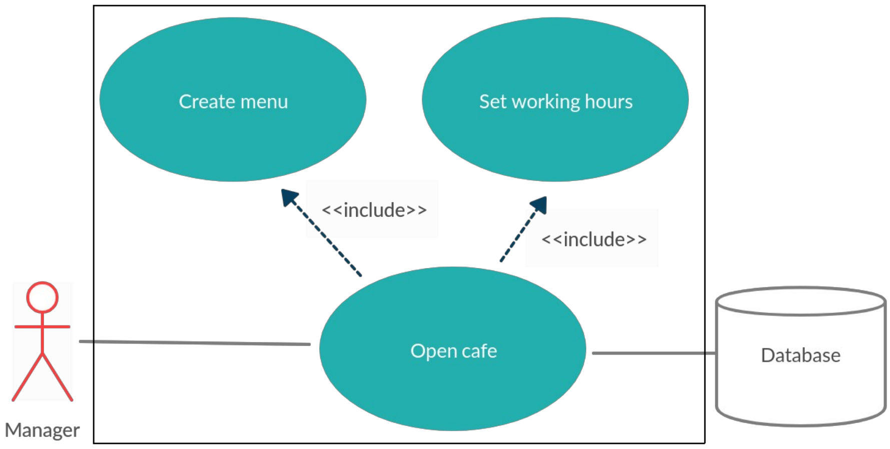


User Case Name	Set working hours
Primary Actor(s)	Manager
Brief description	A manager configures working hours, during which it's possible to order
Preconditions	Working hours are not set. Or working hours are about to be changed.
Flow of events	A manager sets working hours
Postconditions	Working hours are saved.
Priority	High
Alternative flows and exceptions	24 hours open is possible
Sources	United Interviews, Time of order is ready

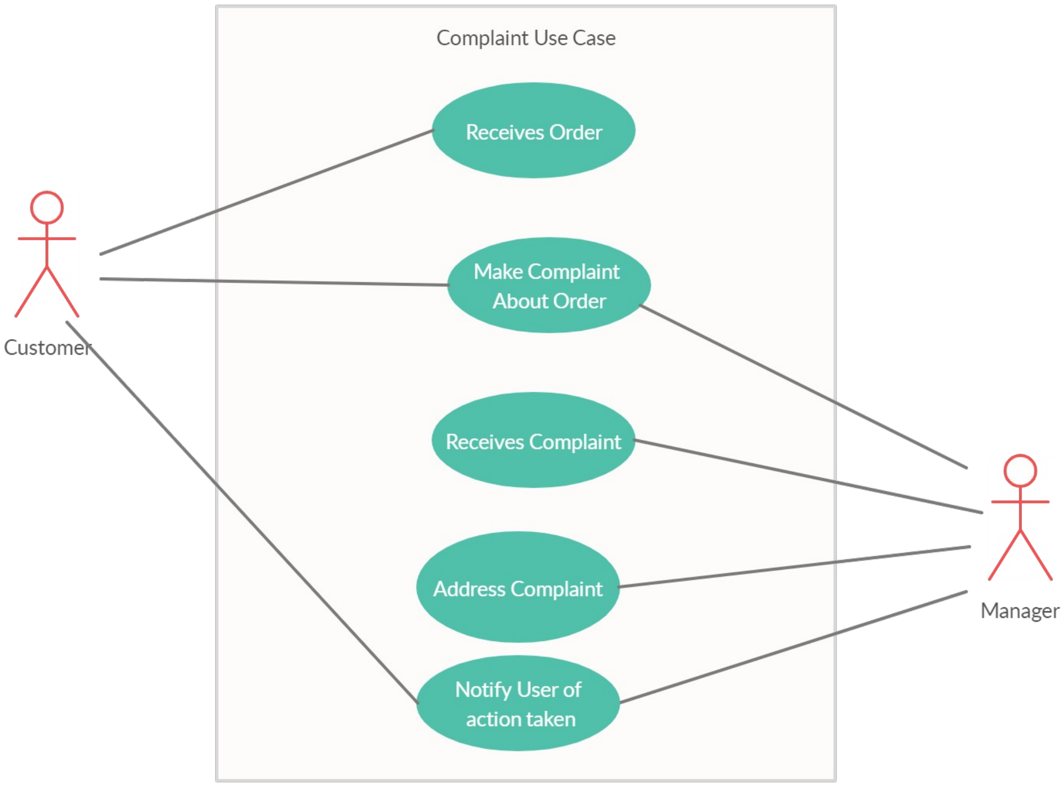


User Case Name	Open cafe
Primary Actor(s)	Manager
Brief description	Open cafe when you are ready
Preconditions	Cafe is not opened

Flow of events	When all required information is filled (dishes, menus, working hours), a manager opens the cafe
Postconditions	Cafe is opened
Priority	High
Alternative flows and exceptions	24 hours open is possible
Sources	United Interviews, Who updates dishes



User Case Name	Make Complaint
Primary Actor(s)	Customer
Brief description	When a customer is not satisfied or happy with the order/delivery.
Preconditions	Customer made an order of meal (Dissatisfied)
Postconditions	Complaint sent to manager
Flow of Events	Users receives an order he/she is not satisfied with. User goes to the list (history) of his/her orders made, then choices the order in question and clicks the complaint button. A textbox is filled and submitted, the message is sent to the manager.
Sources	Interviews, User making complaint


Complaint Use Case
Receives Order
Make Complaint About Order
Receives Complaint
Address Complaint
Notify User of action taken
Customer
Manager