
ProChicken

**ProChicken Fitness
Use-Case Specification**

Version <1.0>

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

Revision History

Date	Version	Description	Author
<12/11/2023>	<1.0>	Creating use case model and describe each of them	Duc Anh

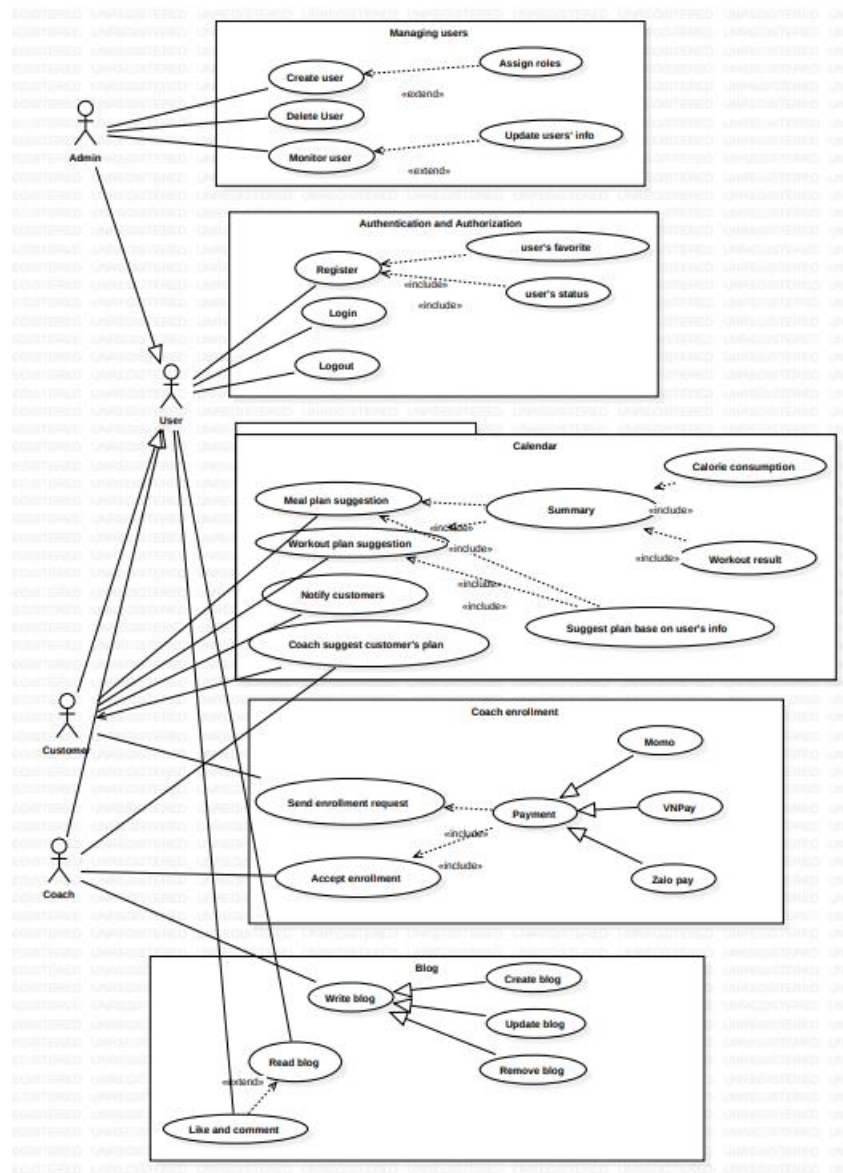
ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

Table of Contents

1.	Use-case Model	4
2.	Use-case Specifications	4
2.1	Use-case: Register	4
2.2	Use-case: Login	5
2.3	Use-case: Logout	5
2.4	Use-case: Collect customer's status	6
2.5	Use-case: Collect customer's favorite	6
2.6	Use-case: Admin create a user	6
2.7	Use-case: Admin delete users	7
2.8	Use-case: Admin monitor users	7
2.9	Use-case: Admin update users' information	7
2.10	Use-case: Meal plan suggestion	8
2.11	Use-case: Workout plan suggestions	8
2.12	Use-case: Notify user	9
2.13	Use-case: Coach suggest customer plan	9
2.14	Use-case: Summary	9
2.15	Use-case: Send enrollment request	10
2.16	Use-case: Payment	10
2.17	Use-case: Accept enrollment	10
2.18	Use-case: Read blog	11
2.19	Use-case: Like and comment blog	11
2.20	Use-case: Remove blog	12
2.21	Use-case: Update blog	12
2.22	Use-case: Create blog	13

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

1. Use-case Model



2. Use-case Specifications

2.1 Use-case: Register

Use case Name	Register user
Brief description	This use case describes how a user is registered
Actors	User
Basic Flow	<ol style="list-style-type: none"> 1. A form register will be provided when user first access the web 2. A user provides necessary information 3. System validates user input 4. Redirect user to homepage
Alternative Flows	Alternative flow 1: Invalid user input

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	
	<ol style="list-style-type: none"> 1. From #3 in the basic flow, error will be displayed in the form 2. User will provide input again 3. Continue step #3 in the basic flow <p>Alternative flow 2: User has already registered</p> <ol style="list-style-type: none"> 1. From #1 of the basic flow, system will provide login form 2. Continue step #2
Pre-conditions	User access the web platform for the first time
Post-conditions	Save user information in the database

2.2 Use-case: Login

Use case Name	Login user
Brief description	This use case describes how a user is logged in
Actors	User
Basic Flow	<ol style="list-style-type: none"> 1. Login form will be displayed 2. User enter their information 3. System will validate user input 4. Redirect user to the homepage
Alternative Flows	<p>Alternative flow 1: Invalid user input</p> <ol style="list-style-type: none"> 1. From #3 in the basic flow, error will be displayed in the form 2. User will provide input again 3. Continue step #3 in the basic flow <p>Alternative flow 2: User is not registered</p> <ol style="list-style-type: none"> 1. From #1 of the basic flow, system will provide register form 2. Continue step #2
Pre-conditions	User has already registered
Post-conditions	Save user's token on cookie

2.3 Use-case: Logout

Use case Name	Logout user
Brief description	This use case describes how a user is logged out
Actors	User
Basic Flow	<ul style="list-style-type: none"> • A button for logging out will be displayed • User clicks on the button • Redirect user to login page
Alternative Flows	
Pre-conditions	User has already logged in
Post-conditions	Redirect user to login page

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

2.4 Use-case: Collect customer's status

Use case Name	Collect customer's status
Brief description	This use case describes how to get a customer's status
Actors	Customer
Basic Flow	<ol style="list-style-type: none"> 1. Multiple forms will be displayed 2. Customer fills data related to their height, weight, target goals 3. System saves this information to the database
Alternative Flows	<p>Alternative flow 1: User has not filled the register form</p> <ol style="list-style-type: none"> 1. From #1 in the basic flow, user is redirected to register page 2. User will provide input again 3. Continue step #1 in the basic flow <p>Alternative flow 2: User has already filled the data</p> <ol style="list-style-type: none"> 1. From #1 of the basic flow, user is redirected to homepage
Pre-conditions	User has already filled the register form
Post-conditions	Save user's status in the database

2.5 Use-case: Collect customer's favorite

Use case Name	Collect customer's favorite
Brief description	This use case describes how customer's favorite is collected
Actors	Customer
Basic Flow	<ol style="list-style-type: none"> 1. Provide multiple forms asking user for food's preferences 2. This information will be saved in the database
Alternative Flows	<p>Alternative flow 1: User has not filled the register form</p> <ol style="list-style-type: none"> 1. From #1 of the basic flow, user will be redirected to register page 2. Continue the #1 <p>Alternative flow 2: User has already filled the data</p> <ol style="list-style-type: none"> 1. From #1 of the basic flow, redirect user to homepage
Pre-conditions	Customer has already filled in the register form
Post-conditions	Save customer's preferences in the database

2.6 Use-case: Admin create a user

Use case Name	Create user
Brief description	This use case describes how an admin create a new user for the system
Actors	Admin
Basic Flow	<ol style="list-style-type: none"> 1. Admin create a new user for current system 2. Admin can assign the role's of user: customer, admin, coach
Alternative Flows	<p>Alternative flow: role not assigned</p> <ol style="list-style-type: none"> 1. The user than have default role as customer

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

	2. User is then created normally
Pre-conditions	Admin after logging in can have the interface to create a new user
Post-conditions	A user is created and saved inside app's database with role assigned

2.7 Use-case: Admin delete users

Use case Name	Delete user
Brief description	Admin delete a user in the database
Actors	Admin
Basic Flow	<ol style="list-style-type: none"> 1. Admin has an interface to delete users 2. Users are chosen from the list by admin 3. Admin delete selected users
Alternative Flows	Alternative flow : no users selected <ol style="list-style-type: none"> 1. The app will notify the statement: "No users selected" 2. The deletion process is canceled
Pre-conditions	Admin after logging in can have the interface to delete users
Post-conditions	Users selected are deleted from the database

2.8 Use-case: Admin monitor users

Use case Name	Monitor users
Brief description	Admin can monitor and has interactions with the user's infomation
Actors	Admin
Basic Flow	<ol style="list-style-type: none"> 1. Admin has an interface to monitor user's status 2. Admin can read user's information and current status
Alternative Flows	
Pre-conditions	Admin after logging in can have a look at all users
Post-conditions	Users' progress and information are shown for the admin

2.9 Use-case: Admin update users' information

Use case Name	Update users
Brief description	Admin can update user's information manually
Actors	Admin
Basic Flow	<ol style="list-style-type: none"> 1. Admin has an interface to update users data 2. Admin than change the user's information 3. Click update to perform the update for user
Alternative Flows	<p>Alternative flow 1: Admin don't change any info</p> <ol style="list-style-type: none"> 1. No updates are made 2. The user's last information is preserved <p>Alternate flow 2: Admin change user's info but don't click the operation button</p>

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

	<ol style="list-style-type: none"> 1. No updates are made 2. The user's last information is preserved
Pre-conditions	Admin after logging in has an interface to update the selected user
Post-conditions	User's information is updated and saved inside the database

2.10 Use-case: Meal plan suggestion

Use case Name	Meal plan suggestion
Brief description	The system can create a meal plan for user during the week
Actors	Users
Basic Flow	<ol style="list-style-type: none"> 1. User can click the button to auto-create a meal plan. 2. The system then create a meal plan for user base on personal information (user's favourite, user's hates, current status) 3. A meal plan for a week is than created and show for user. 4. The user can tick to update their progress.
Alternative Flows	<p>Alternate flow 1: User has no favourites or hates</p> <ol style="list-style-type: none"> 1. A default meal will be created that get along well with most other users. <p>Alternate flow 2: The user does not tick the progress</p> <ol style="list-style-type: none"> 2. The summary will not count into the day progress
Pre-conditions	User choose to create their meal plan base on current information
Post-conditions	A meal calendar is created for user to follow

2.11 Use-case: Workout plan suggestions

Use case Name	Workout plan suggestion
Brief description	The system can create a workout plan for user during the week
Actors	Users
Basic Flow	<ol style="list-style-type: none"> 1. The user click the workout plan create button to create a workout plan 2. The system will collect user's status (weight, height, etc) to suggest the best fit workout exercises for user during the week 3. The plan will show for the user 4. The user can tick if they have done the exercises
Alternative Flows	<p>Alternate flow 1: User has no favourites or hates</p> <ol style="list-style-type: none"> 1. A default workout plan will be created that get along well with most other users.

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

	Alternate flow 2: The user does not tick the progress 1. The summary will not count into the day progress
Pre-conditions	User choose to create their workout plan base on current information
Post-conditions	A workout calendar is created for user to follow

2.12 Use-case: Notify user

Use case Name	Notify user
Brief description	The app (mobile) can notify the user's when it's time to do the workout or meal
Actors	Users
Basic Flow	<ol style="list-style-type: none"> 1. When the time comes, a notification will be sent to user's app 2. The user then get the notification and prepare for it.
Alternative Flows	Alternate flow : the notification can't be sent due to internet connection. <ol style="list-style-type: none"> 1. That notification is pending and will be sent later.
Pre-conditions	User that already has a plan
Post-conditions	A notification is sent to user's phone

2.13 Use-case: Coach suggest customer plan

Use case Name	Coach suggest customer plan
Brief description	The coach can give suggestion for user's plan
Actors	Coach
Basic Flow	<ol style="list-style-type: none"> 1. The coach has an interface to give customers suggestion. 2. The plan will then assigned to user's calendar
Alternative Flows	Alternate flow : the plan does not fit user's preference <ol style="list-style-type: none"> 1. A warning will appear notify coach. 2. The coach still have to change the plan.
Pre-conditions	A coach is the current coach of user's
Post-conditions	A plan is created to guide customer

2.14 Use-case: Summary

Use case Name	Coach suggest customer plan
Brief description	The coach can give suggestion for user's plan
Actors	User
Basic Flow	<ol style="list-style-type: none"> 1. After a day, a summary is calculated. 2. The summary will contain calorie consumption, calorie defecit and calorie burnt
Alternative Flows	Alternate flow : The progress is not finished <ol style="list-style-type: none"> 1. With lack of information, the users still get the summary but will get the

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

	pop-up that recommend user to follow the plan
Pre-conditions	The user that has a plan in the day
Post-conditions	Give customers summary about current day

2.15 Use-case: Send enrollment request

Use case Name	Send enrollement request
Brief description	The user can send the enrollment request that follow a coach guide
Actors	User
Basic Flow	<ol style="list-style-type: none"> 1. A user interface for use to choose a coach they want to follow 2. User than guided to the description page about that coach. 3. User can choose to follow the coach. 4. User is switched to payment page. 5. After the payment, the request is sent to coach. If the coach declined the request then the money will be refunded for user.
Alternative Flows	Alternate flow : The user cancel the request. <ol style="list-style-type: none"> 1. The request is deleted from the coach perspective. 2. The user can have the refund.
Pre-conditions	A logged in user that want to follow a coach
Post-conditions	A request is sent and waited for reply.

2.16 Use-case: Payment

Use case Name	Payment progress
Brief description	The user will choose payment method and pay for the coach
Actors	User
Basic Flow	<ol style="list-style-type: none"> 1. The user has an interface to give information. 2. The user can choose payment method (Momo, Zalopay, VNPay). 3. After the checkout is successful, the user will have notification about the pending request
Alternative Flows	Alternative flow: payment failed <ol style="list-style-type: none"> 1. The user will be notified with the pop-up. 2. All the transaction is canceled.
Pre-conditions	A user that has chosen to follow a coach.
Post-conditions	The payment is successful and the request sent

2.17 Use-case: Accept enrollment

Use case Name	Accept enrollment
Brief description	This use case describes how a coach accept enrollment request from customer
Actors	Coach

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

Basic Flow	<ol style="list-style-type: none"> 1. Coach has a button for viewing enrollment requests from customer 2. Coach clicks on the button 3. Redirect coach to enrollment requests page 4. System shows coach a list of enrollments requests 5. Each enrollment request has accept and deny button 6. Coach clicks on button 7. System handles the business logic
Alternative Flows	<p>Alternative flow 1: Coach clicks on the accept button</p> <ol style="list-style-type: none"> 1. From the #7 of the basic flow, system transfer money from customer's bank to coach's bank <p>Alternative flow 2: Coach clicks on the deny button</p> <ol style="list-style-type: none"> 1. From the #7 of the basic flow, system will delete customer's request enrollment from coach's enrollment list 2. The money transaction is not continued
Pre-conditions	User is coach
Post-conditions	Money transaction between coach and customer is processed or delete customer's request from coach's enrollment list

2.18 Use-case: Read blog

Use case Name	Read Blog
Brief description	This use case describes how a user read a blog
Actors	User
Basic Flow	<ol style="list-style-type: none"> 1. A list of blogs is displayed to the user blog section 2. Each blog has a button for viewing more 3. User clicks on that button 4. System redirects user to blog detail page 5. User reads the full blog
Alternative Flows	<p>Alternative flow 1: User has not logged in</p> <ol style="list-style-type: none"> 1. From the #1 of the basic flow, user is directed to login page
Pre-conditions	User has already logged in
Post-conditions	Display a full blog

2.19 Use-case: Like and comment blog

Use case Name	Like and comment blog
Brief description	This use case describes how a user like and comment a blog
Actors	User
Basic Flow	<ol style="list-style-type: none"> 6. A list of blogs is displayed to the user blog section 7. Each blog has a button for viewing more 8. User clicks on that button

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

	9. System redirects user to blog detail page 10. Blog detail page has comment and like section at the bottom of the page 11. User writes their comment or clicks a like button 12. Update the blog user have reacted
Alternative Flows	Alternative flow 1: Customer click a like button 2. From the #6 of the basic flow, increase the number of likes of that blog Alternative flow 2: Customer comment a blog 1. From the #6 of the basic flow, add a new comment to that blog in the database
Pre-conditions	User has already logged in
Post-conditions	Add a blog's comment or update number of likes

2.20 Use-case: Remove blog

Use case Name	Remove blog
Brief description	This use case describes how a coach remove a blog
Actors	Coach
Basic Flow	1. A list of coach's blogs is displayed 2. Each blog has a remove button 3. Coach clicks on that button 4. System removes this blog from the database
Alternative Flows	Alternative flow 1: User is not a coach 8. From the #1 of the basic flow, a list of blogs is not displayed
Pre-conditions	User is a coach
Post-conditions	Remove a coach's blog from the database

2.21 Use-case: Update blog

Use case Name	Update blog
Brief description	This use case describes how a coach update a blog
Actors	Coach
Basic Flow	5. A list of coach's blogs will be displayed 6. Each blog has a button for update 7. Coach clicks on that button 8. System redirect coach to edit blog page 9. Coach updates their blog
Alternative Flows	Alternative flow 1: User is not a coach 1. From the #1 of the basic flow, a list of blogs is not displayed
Pre-conditions	User is a coach
Post-conditions	Save blog's updated blog to the database

ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

2.22 Use-case: Create blog

Use case Name	Create blog
Brief description	This use case describes how a coach write a blog
Actors	Coach
Basic Flow	10. A button allowing coach to write a blog 11. Coach clicks on that button 12. Coach is redirected to write blog page 13. Coach writes their blog 14. System saves this blog to the database
Alternative Flows	Alternative flow 1: User is not a coach 2. From the #1 of the basic flow, the button to write a blog is not displayed
Pre-conditions	User is a coach
Post-conditions	Save coach's blog to the database