

---

**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

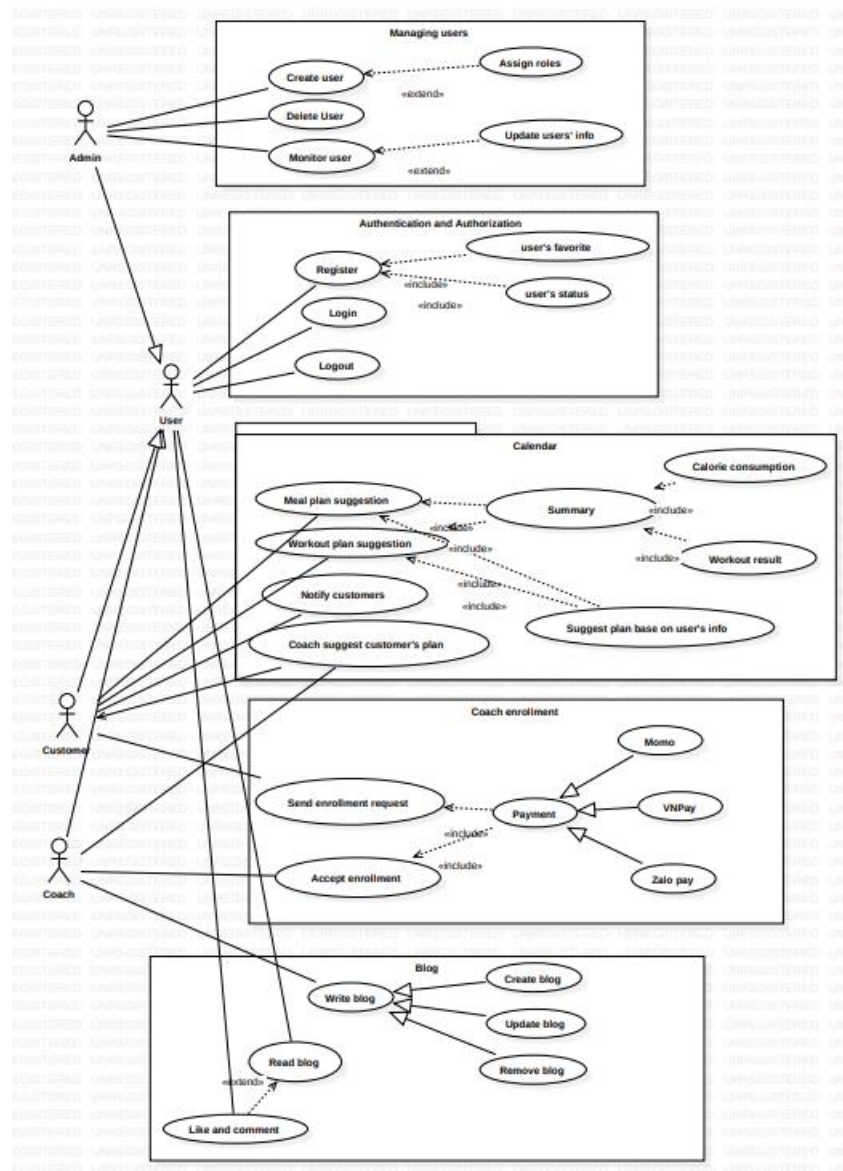
ProChicken fitness	Version: <1.0>
Use-Case Specification: <Use-Case Name>	Date: 12/11/2023>
<document identifier>	

## Table of Contents

1.	Use-case Model	4
2.	Use-case Specifications	5
2.1	Use-case: Register	5
2.2	Use-case: Login	5
2.3	Use-case: Logout	6
2.4	Use-case: Collect customer's status	6
2.5	Use-case: Collect customer's favorite	7
2.6	Use-case: Admin create a user	7
2.7	Use-case: Admin delete users	8
2.8	Use-case: Admin monitor users	8
2.9	Use-case: Admin update users' information	9
2.10	Use-case: Meal plan suggestion	9
2.11	Use-case: Workout plan suggestions	10
2.12	Use-case: Notify user	10
2.13	Use-case: Coach suggest customer plan	11
2.14	Use-case: Summary	11
2.15	Use-case: Send enrollment request	12
2.16	Use-case: Payment	12
2.17	Use-case: Accept enrollment	13
2.18	Use-case: Read blog	13
2.19	Use-case: Like and comment blog	14
2.20	Use-case: Remove blog	14
2.21	Use-case: Update blog	15
2.22	Use-case: Create blog	15

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

## 1. Use-case Model



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

## 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"> <li>1. A form register will be provided when user first access the web</li> <li>2. A user provides necessary information</li> <li>3. System validates user input</li> <li>4. Redirect user to homepage</li> </ol>
Alternative Flows	<p><b>Alternative flow 1: Invalid user input</b></p> <ol style="list-style-type: none"> <li>1. From #3 in the basic flow, error will be displayed in the form</li> <li>2. User will provide input again</li> <li>3. Continue step #3 in the basic flow</li> </ol> <p><b>Alternative flow 2: User has already registered</b></p> <ol style="list-style-type: none"> <li>1. From #1 of the basic flow, system will provide login form</li> <li>2. Continue step #2</li> </ol>
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"> <li>1. Login form will be displayed</li> <li>2. User enter their information</li> <li>3. System will validate user input</li> <li>4. Redirect user to the homepage</li> </ol>
Alternative Flows	<p><b>Alternative flow 1: Invalid user input</b></p> <ol style="list-style-type: none"> <li>1. From #3 in the basic flow, error will be displayed in the form</li> <li>2. User will provide input again</li> <li>3. Continue step #3 in the basic flow</li> </ol> <p><b>Alternative flow 2: User is not registered</b></p> <ol style="list-style-type: none"> <li>1. From #1 of the basic flow, system will provide register form</li> <li>2. Continue step #2</li> </ol>
Pre-conditions	User has already registered
Post-conditions	Save user's token on cookie

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

### 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"> <li>• A button for logging out will be displayed</li> <li>• User clicks on the button</li> <li>• Redirect user to login page</li> </ul>
Alternative Flows	
Pre-conditions	User has already logged in
Post-conditions	Redirect user to login page

### 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"> <li>1. Multiple forms will be displayed</li> <li>2. Customer fills data related to their height, weight, target goals</li> <li>3. System saves this information to the database</li> </ol>
Alternative Flows	<p><b>Alternative flow 1: User has not filled the register form</b></p> <ol style="list-style-type: none"> <li>1. From #1 in the basic flow, user is redirected to register page</li> <li>2. User will provide input again</li> <li>3. Continue step #1 in the basic flow</li> </ol> <p><b>Alternative flow 2: User has already filled the data</b></p> <ol style="list-style-type: none"> <li>1. From #1 of the basic flow, user is redirected to homepage</li> </ol>
Pre-conditions	User has already filled the register form
Post-conditions	Save user's status in the database

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

## 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"> <li>1. Provide multiple forms asking user for food's preferences</li> <li>2. This information will be saved in the database</li> </ol>
Alternative Flows	<p><b>Alternative flow 1: User has not filled the register form</b></p> <ol style="list-style-type: none"> <li>1. From #1 of the basic flow, user will be redirected to register page</li> <li>2. Continue the #1</li> </ol> <p><b>Alternative flow 2: User has already filled the data</b></p> <ol style="list-style-type: none"> <li>1. From #1 of the basic flow, redirect user to homepage</li> </ol>
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"> <li>1. Admin create a new user for current system</li> <li>2. Admin can assign the role's of user: customer, admin, coach</li> </ol>
Alternative Flows	<p><b>Alternative flow: role not assigned</b></p> <ol style="list-style-type: none"> <li>1. The user than have default role as customer</li> <li>2. User is then created normally</li> </ol>
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

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

## 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"> <li>1. Admin has an interface to delete users</li> <li>2. Users are chosen from the list by admin</li> <li>3. Admin delete selected users</li> </ol>
Alternative Flows	<b>Alternative flow: no users selected</b> <ol style="list-style-type: none"> <li>1. The app will notify the statement: “No users selected”</li> <li>2. The deletion process is canceled</li> </ol>
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"> <li>1. Admin has an interface to monitor user’s status</li> <li>2. Admin can read user’s information and current status</li> </ol>
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



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

## 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"> <li>1. Admin has an interface to update users data</li> <li>2. Admin than change the user's information</li> <li>3. Click update to perform the update for user</li> </ol>
Alternative Flows	<p><b>Alternative flow 1: Admin don't change any info</b></p> <ol style="list-style-type: none"> <li>1. No updates are made</li> <li>2. The user's last information is preserved</li> </ol> <p><b>Alternate flow 2: Admin change user's info but don't click the operation button</b></p> <ol style="list-style-type: none"> <li>1. No updates are made</li> <li>2. The user's last information is preserved</li> </ol>
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"> <li>1. User can click the button to auto-create a meal plan.</li> <li>2. The system then create a meal plan for user base on personal information (user's favourite, user's hates, current status)</li> <li>3. A meal plan for a week is than created and show for user.</li> <li>4. The user can tick to update their progress.</li> </ol>
Alternative Flows	<p><b>Alternate flow 1: User has no favourites or hates</b></p> <ol style="list-style-type: none"> <li>1. A default meal will be created that get along well with most other users.</li> </ol> <p><b>Alternate flow 2: The user does not tick the progress</b></p> <ol style="list-style-type: none"> <li>2. The summary will not count into the day progress</li> </ol>
Pre-conditions	User choose to create their meal plan base on current information
Post-conditions	A meal calendar is created for user to follow

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

### 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"> <li>1. The user click the workout plan create button to create a workout plan</li> <li>2. The system will collect user's status (weight, height, etc) to suggest the best fit workout exercises for user during the week</li> <li>3. The plan will show for the user</li> <li>4. The user can tick if they have done the exercises</li> </ol>
Alternative Flows	<p><b>Alternate flow 1: User has no favourites or hates</b></p> <ol style="list-style-type: none"> <li>1. A default workout plan will be created that get along well with most other users.</li> </ol> <p><b>Alternate flow 2: The user does not tick the progress</b></p> <ol style="list-style-type: none"> <li>1. The summary will not count into the day progress</li> </ol>
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"> <li>1. When the time comes, a notification will be sent to user's app</li> <li>2. The user then get the notification and prepare for it.</li> </ol>
Alternative Flows	<p><b>Alternate flow : the notification can't be sent due to internet connection.</b></p> <ol style="list-style-type: none"> <li>1. That notification is pending and will be sent later.</li> </ol>
Pre-conditions	User that already has a plan
Post-conditions	A notification is sent to user's phone

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

### 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"> <li>1. The coach has an interface to give customers suggestion.</li> <li>2. The plan will then assigned to user's calendar</li> </ol>
Alternative Flows	<b>Alternate flow : the plan does not fit user's preference</b> <ol style="list-style-type: none"> <li>1. A warning will appear notify coach.</li> <li>2. The coach still have to change the plan.</li> </ol>
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"> <li>1. After a day, a summary is calculated.</li> <li>2. The summary will contain calorie consumption, calorie defecit and calorie burnt</li> </ol>
Alternative Flows	<b>Alternate flow : The progress is not finished</b> <ol style="list-style-type: none"> <li>1. With lack of information, the users still get the summary but will get the pop-up that recommend user to follow the plan</li> </ol>
Pre-conditions	The user that has a plan in the day
Post-conditions	Give customers summary about current day

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

### 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"> <li>1. A user interface for use to choose a coach they want to follow</li> <li>2. User than guided to the description page about that coach.</li> <li>3. User can choose to follow the coach.</li> <li>4. User is switched to payment page.</li> <li>5. After the payment, the request is sent to coach. If the coach declined the request then the money will be refunded for user.</li> </ol>
Alternative Flows	<b>Alternate flow : The user cancel the request.</b> <ol style="list-style-type: none"> <li>1. The request is deleted from the coach perspective.</li> <li>2. The user can have the refund.</li> </ol>
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"> <li>1. The user has an interface to give information.</li> <li>2. The user can choose payment method (Momo, Zalopay, VNPAY).</li> <li>3. After the checkout is successful, the user will have notification about the pending request</li> </ol>
Alternative Flows	<b>Alternative flow: payment failed</b> <ol style="list-style-type: none"> <li>1. The user will be notified with the pop-up.</li> <li>2. All the transaction is canceled.</li> </ol>
Pre-conditions	A user that has chosen to follow a coach.
Post-conditions	The payment is successful and the request sent

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

## 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
Basic Flow	<ol style="list-style-type: none"> <li>1. Coach has a button for viewing enrollment requests from customer</li> <li>2. Coach clicks on the button</li> <li>3. Redirect coach to enrollment requests page</li> <li>4. System shows coach a list of enrollments requests</li> <li>5. Each enrollment request has accept and deny button</li> <li>6. Coach clicks on button</li> <li>7. System handles the business logic</li> </ol>
Alternative Flows	<p><b>Alternative flow 1: Coach clicks on the accept button</b></p> <ol style="list-style-type: none"> <li>1. From the #7 of the basic flow, system transfer money from customer's bank to coach's bank</li> </ol> <p><b>Alternative flow 2: Coach clicks on the deny button</b></p> <ol style="list-style-type: none"> <li>1. From the #7 of the basic flow, system will delete customer's request enrollment from coach's enrollment list</li> <li>2. The money transaction is not continued</li> </ol>
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"> <li>1. A list of blogs is displayed to the user blog section</li> <li>2. Each blog has a button for viewing more</li> <li>3. User clicks on that button</li> <li>4. System redirects user to blog detail page</li> <li>5. User reads the full blog</li> </ol>
Alternative Flows	<p><b>Alternative flow 1: User has not logged in</b></p> <ol style="list-style-type: none"> <li>1. From the #1 of the basic flow, user is directed to login page</li> </ol>
Pre-conditions	User has already logged in
Post-conditions	Display a full blog

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

## 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"> <li>1. A list of blogs is displayed to the user blog section</li> <li>2. Each blog has a button for viewing more</li> <li>3. User clicks on that button</li> <li>4. System redirects user to blog detail page</li> <li>5. Blog detail page has comment and like section at the bottom of the page</li> <li>6. User writes their comment or clicks a like button</li> <li>7. Update the blog user have reacted</li> </ol>
Alternative Flows	<p><b>Alternative flow 1: Customer click a like button</b></p> <ol style="list-style-type: none"> <li>1. From the #6 of the basic flow, increase the number of likes of that blog</li> </ol> <p><b>Alternative flow 2: Customer comment a blog</b></p> <ol style="list-style-type: none"> <li>2. From the #6 of the basic flow, add a new comment to that blog in the database</li> </ol>
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	<ol style="list-style-type: none"> <li>1. A list of coach's blogs is displayed</li> <li>2. Each blog has a remove button</li> <li>3. Coach clicks on that button</li> <li>4. System removes this blog from the database</li> </ol>
Alternative Flows	<p><b>Alternative flow 1: User is not a coach</b></p> <ol style="list-style-type: none"> <li>1. From the #1 of the basic flow, a list of blogs is not displayed</li> </ol>
Pre-conditions	User is a coach
Post-conditions	Remove a coach's blog from the database

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

## 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	<ol style="list-style-type: none"> <li>1. A list of coach's blogs will be displayed</li> <li>2. Each blog has a button for update</li> <li>3. Coach clicks on that button</li> <li>4. System redirect coach to edit blog page</li> <li>5. Coach updates their blog</li> </ol>
Alternative Flows	<b>Alternative flow 1: User is not a coach</b> <ol style="list-style-type: none"> <li>1. From the #1 of the basic flow, a list of blogs is not displayed</li> </ol>
Pre-conditions	User is a coach
Post-conditions	Save blog's updated blog to the database

## 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	<ol style="list-style-type: none"> <li>1. A button allowing coach to write a blog</li> <li>2. Coach clicks on that button</li> <li>3. Coach is redirected to write blog page</li> <li>4. Coach writes their blog</li> <li>5. System saves this blog to the database</li> </ol>
Alternative Flows	<b>Alternative flow 1: User is not a coach</b> <ol style="list-style-type: none"> <li>1. From the #1 of the basic flow, the button to write a blog is not displayed</li> </ol>
Pre-conditions	User is a coach
Post-conditions	Save coach's blog to the database