

FiTracks

Analysis Model

Submitted to:

Asst. Prof. Ma. Rowena C. Solamo
Faculty Member
Department of Computer Science
College of Engineering
University of the Philippines, Diliman

Submitted by:

Aguilana, Trina B.
Manguiat, Glenn Karlo D.
Villanueva, Ian N.

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2018-2019



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Unique Reference:

The documents are stored in the Project Repository Link: <https://bit.ly/2NymdWX> referenced with Group 4 - FiTracks - Analysis Model.pdf.

Purpose:

This document is provided to model the use cases and its scenarios for the health and fitness record management system through Analysis Model and Behavior Models. In the analysis model, each classes are provided with descriptions for readability for the audience.

Audience:

This document serves its purpose for Asst. Prof. Rowena Solamo and her students analyzing the use cases and its scenarios of the health and fitness record management system.

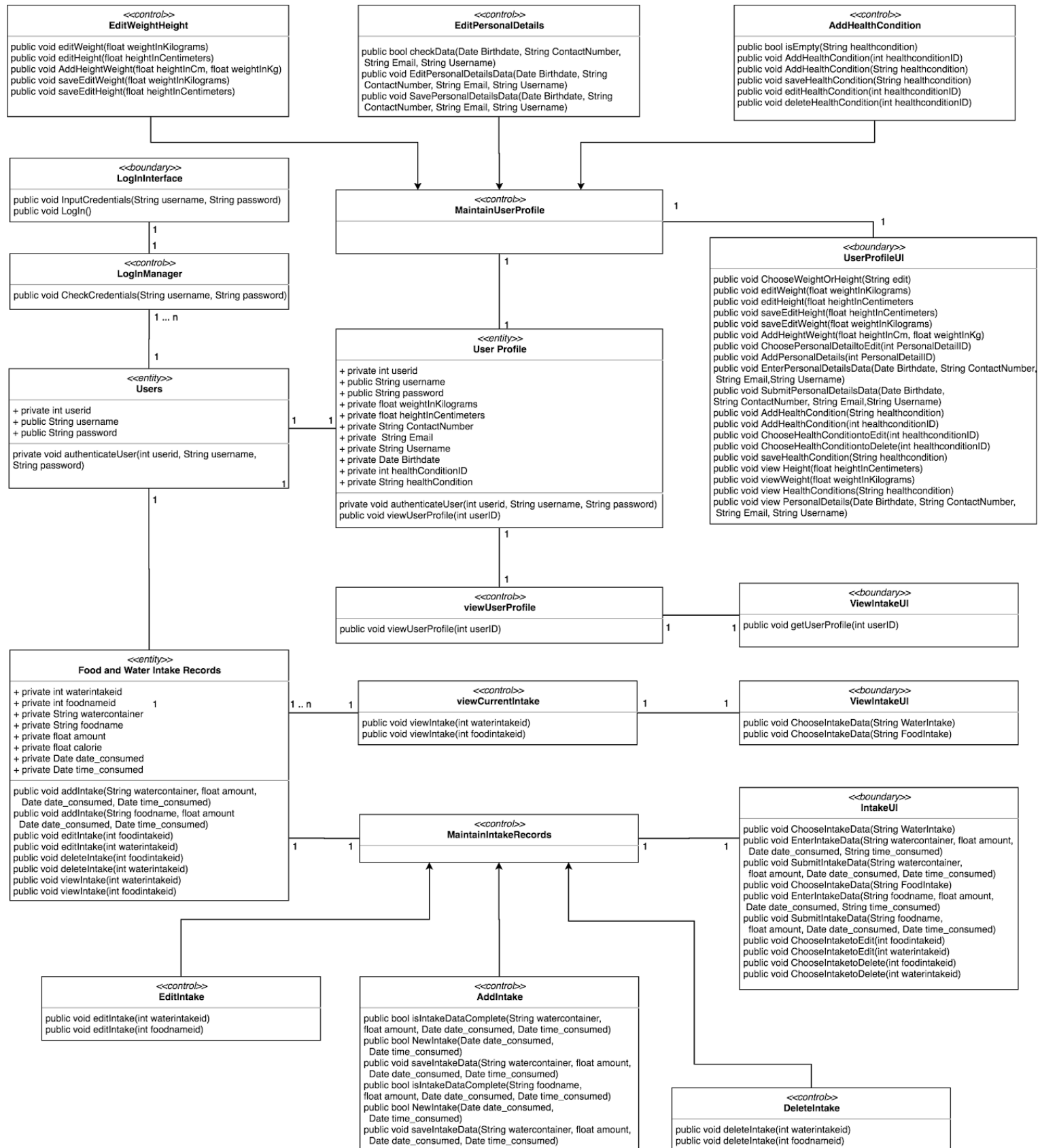
Revision Control:

<i>Revision Date</i>	<i>Person Responsible</i>	<i>Version Number</i>	<i>Modification</i>
10/01/18	Ian N. Villanueva	1.0	Initial Document; Added document purpose and audience, added system name and its description.
10/03/18	Glenn Karlo D. Manguiat	1.1	Added boundary classes for Use Cases 1.0-1.3, 7.0; Added control classes for Use Cases 1.0-1.3, 7.0; Added entity classes for Use Cases 1.0-1.3, 7.0; Added behavior models for Use Case 1.0-1.3; Edited document purpose and audience.
10/04/18	Ian N. Villanueva	1.2	Added boundary classes for Use Cases 2.0-2.3; Added control classes for Use Cases 2.0-2.3; Added entity classes for Use Cases 2.0-2.3; Added behavior models for Use Case 2.0-2.3;
10/04/18	Trina B. Aguilana	1.3	Added boundary classes for Use Cases 4.0, 6.0; Added control classes for Use Cases 4.0, 6.0; Added behavior models for Use Cases 4.0, 6.0

System Name: FiTracks - Health and Fitness Record Management System

Description: In our health and fitness record management system, a user maintains a personal food and water intake tracker upon log-in. User establishes a profile which contains his/her personal details, current height, current weight, and current health condition, if with complications or none. The establish profile may be viewed and updated at any time possible. User can input meal intake and water intake, which may be updated and deleted upon addition. Lastly, the current meal and water intake status can be viewed by the user in different time schemes.

Analysis Model:



Boundary Classes:

Class Name	Description
IntakeUI	<p>This is the interface of user whenever she or he wants to maintain the food and water intakes.</p> <p><u>Responsibilities:</u></p> <pre>public void ChooseIntakeData(String WaterIntake) public void EnterIntakeData(String watercontainer, float amount, Date date_consumed, String time_consumed) public void SubmitIntakeData(String watercontainer, float amount, Date date_consumed, Date time_consumed) public void ChooseIntakeData(String FoodIntake) public void EnterIntakeData(String foodname, float amount, Date date_consumed, String time_consumed) public void SubmitIntakeData(String foodname, float amount, Date date_consumed, Date time_consumed) public void ChooseIntaketoEdit(int foodintakeid) public void ChooseIntaketoEdit(int waterintakeid) public void ChooseIntaketoDelete(int foodintakeid) public void ChooseIntaketoDelete(int waterintakeid)</pre>
LogInInterface	<p>This is the interface for the user log-in.</p> <p><u>Responsibilities:</u></p> <pre>public void InputCredentials(String username, String password) public void LogIn()</pre>
UserProfileUI	<p>This is the interface of user whenever they want to view or edit their respective user profiles</p> <p><u>Responsibilities:</u></p> <pre>public void ChooseWeightOrHeight(String edit) public void editWeight(float weightInKilograms) public void editHeight(float heightInCentimeters) public void saveEditHeight(float heightInCentimeters) public void saveEditWeight(float weightInKilograms) public void AddHeightWeight(float heightInCm, float weightInKg) public void ChoosePersonalDetailtoEdit(int PersonalDetailID)</pre>

	<pre> public void AddPersonalDetails(int PersonalDetailID) public void EnterPersonalDetailsData(Date Birthdate, String ContactNumber, String Email,String Username) public void SubmitPersonalDetailsData(Date Birthdate, String ContactNumber, String Email,String Username) public void AddHealthCondition(String healthcondition) public void AddHealthCondition(int healthconditionID) public void ChooseHealthConditiontoEdit(int healthconditionID) public void ChooseHealthConditiontoDelete(int healthconditionID) public void saveHealthCondition(String healthcondition) public void view Height(float heightInCentimeters) public void viewWeight(float weightInKilograms) public void view HealthConditions(String healthcondition) public void view PersonalDetails(Date Birthdate, String ContactNumber, String Email, String Username) </pre>
ViewUserProfileUI	<p>This is the interface for when the user wants to view his or her profile.</p> <p><u>Responsibilities:</u></p> <pre> public void getUserProfile(int userID) </pre>
viewCurrentIntake	<p>This is the interface for when the user wants to view a food or water intake.</p> <p><u>Responsibilities:</u></p> <pre> public void ChooseIntakeData(int waterintakeid) public void ChooseIntakeData(int foodintakeid) </pre>

Control Classes:

Class Name	Description
MaintainIntakeRecords	This is the control that maintain food and water intakes record. It is considered an abstract class.
AddIntake	<p>This is the control that adds a food or water intake to the system. It extends MaintainIntakeRecords.</p> <p><u>Responsibilities:</u></p> <pre>public bool isIntakeDataComplete(String watercontainer, float amount, Date date_consumed, Date time_consumed) public bool NewIntake(Date date_consumed, Date time_consumed) public void saveIntakeData(String watercontainer, float amount, Date date_consumed, Date time_consumed) public bool isIntakeDataComplete(String foodname, float amount, Date date_consumed, Date time_consumed) public bool NewIntake(Date date_consumed, Date time_consumed) public void saveIntakeData(String foodname, float amount, Date date_consumed, Date time_consumed)</pre>
EditIntake	<p>This is the control that edits a food or water intake to the system. It extends MaintainIntakeRecords.</p> <p><u>Responsibilities:</u></p> <pre>public void editIntake(int waterintakeid) public void editIntake(int foodnameid)</pre>
DeleteIntake	<p>This is the control that deletes a food or water intake to the system. It extends MaintainIntakeRecords.</p> <p><u>Responsibilities:</u></p> <pre>public void deleteIntake(int waterintakeid) public void deleteIntake(int foodnameid)</pre>
ViewIntake	<p>This is the control that lets the user view his/her current food or water intake</p> <p><u>Responsibilities:</u></p> <pre>public void viewIntake(int waterintakeid) public void viewIntake(int foodintakeid)</pre>
MaintainUserProfile	<p>This is the control that maintains user profiles. It is considered an abstract class.</p> <pre>public void CheckCredentials(String username, String password)</pre>
EditWeightHeight	<p>This is the control that edits user weight/height. It extends MaintainUserProfile</p> <p><u>Responsibilities:</u></p>

	public void editWeight(float weightInKilograms) public void editHeight(float heightInCentimeters) public void AddHeightWeight(float heightInCm, float weightInKg) public void saveEditWeight(float weightInKilograms) public void saveEditHeight(float heightInCentimeters)
EditPersonalDetails	This is the control that edits user personal details It extends MaintainUserProfile. <u>Responsibilities:</u> public bool checkData(Date Birthdate, String ContactNumber, String Email,String Username) public void EditPersonalDetailsData(Date Birthdate, String ContactNumber, String Email, String Username) public void SavePersonalDetailsData(Date Birthdate, String ContactNumber, String Email, String Username)
AddHealthCondition	This is the control that adds user health conditions. It extends MaintainUserProfile. <u>Responsibilities:</u> public bool isEmpty(String healthcondition) public void AddHealthCondition(int healthconditionID) public void AddHealthCondition(String healthcondition) public void saveHealthCondition(String healthcondition) public void editHealthCondition(int healthconditionID) public void deleteHealthCondition(int healthconditionID)
viewUserProfile	This is the control that lets the user view his/her user profile details. <u>Responsibilities:</u> public void viewUserProfile(int userID)
viewCurrentIntake	This is the control that lets the user view a food or water intake. <u>Responsibilities:</u> public void viewIntake(int waterintakeid) public void viewIntake(int foodintakeid)
LogInManager	This is the control that checks the credentials of the user log-in whether it is authenticated or not. <u>Responsibilities:</u> public void CheckCredentials(String username, String password)

Entity Classes:

Class Name	Description
Food and Water Intake Records	<p>This is the entity class Food and Water Intake Records, which contains the data about the food and water intakes added by the user.</p> <p><u>Attributes:</u></p> <p>private int waterintakeid</p> <p>private int foodnameid</p> <p>private String watercontainer</p> <p>private String foodname</p> <p>private float amount</p> <p>private float calorie</p> <p>private Date date_consumed</p> <p>private Date time_consumed</p>
Users	<p>This is the entity class Users, which contains all the information about the users in the health and fitness management record.</p> <p><u>Attributes:</u></p> <p>private int userid</p> <p>public String username</p> <p>private String password</p>
User Profile	<p>This is the entity class User Profile, which contains the user's profile information.</p> <p><u>Attributes:</u></p> <p>private int userID</p> <p>private float weightInKilograms</p> <p>private float heightInCentimeters</p> <p>private String ContactNumber</p> <p>private String Email</p> <p>private String Username</p> <p>private Date Birthdate</p> <p>private int healthConditionID</p> <p>private String healthCondition</p>

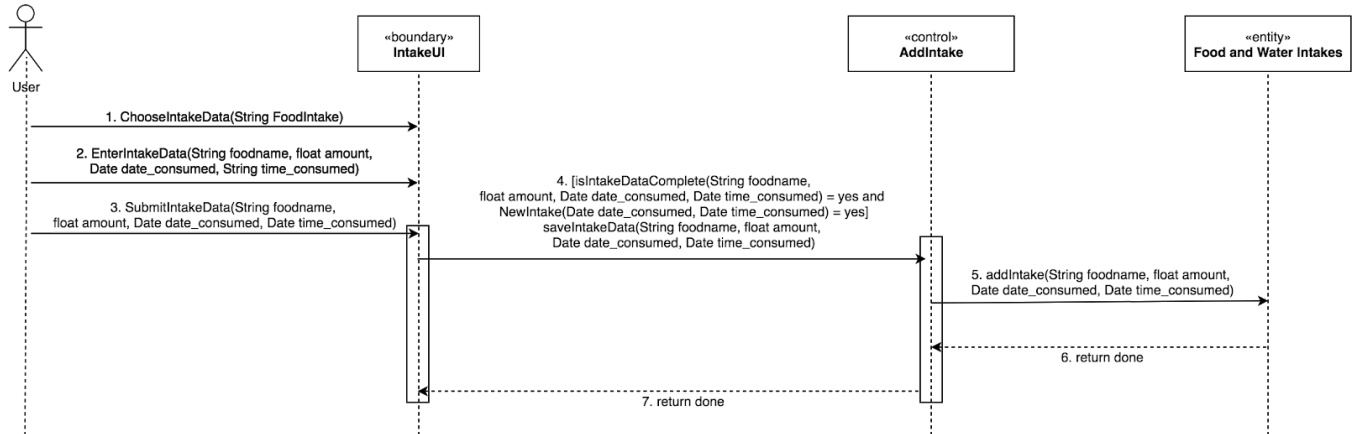
Behavioral Model:

Use-Case Name: Add Meal or Water Intake

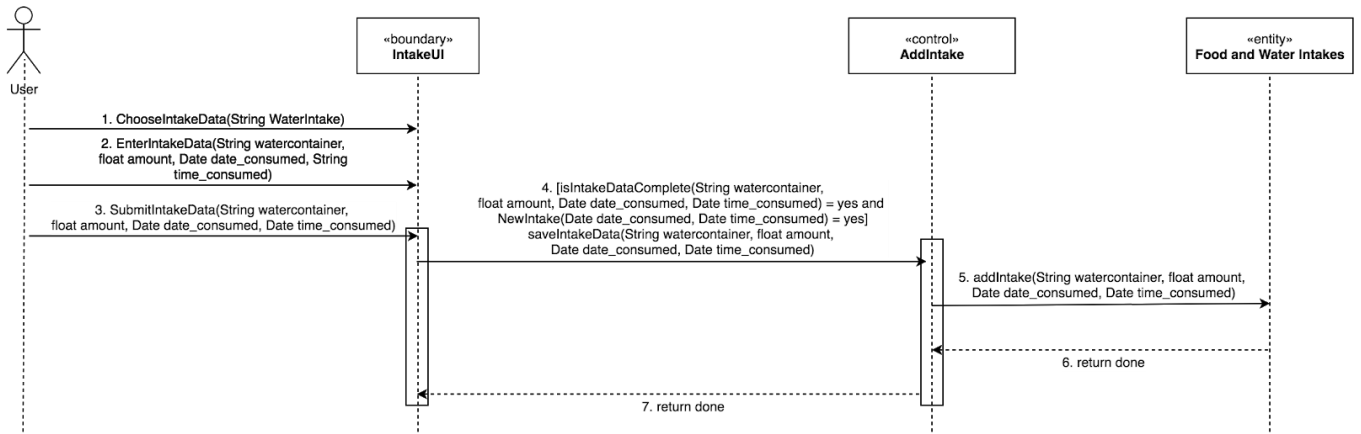
Description:

The user will be able to add a food or a water intake, and with that user describes when they ate the food, or drank the water. User must be able to add the quantity of the food/water intake.

Scenario 1: User adds a food intake.



Scenario 2: User adds a water intake.

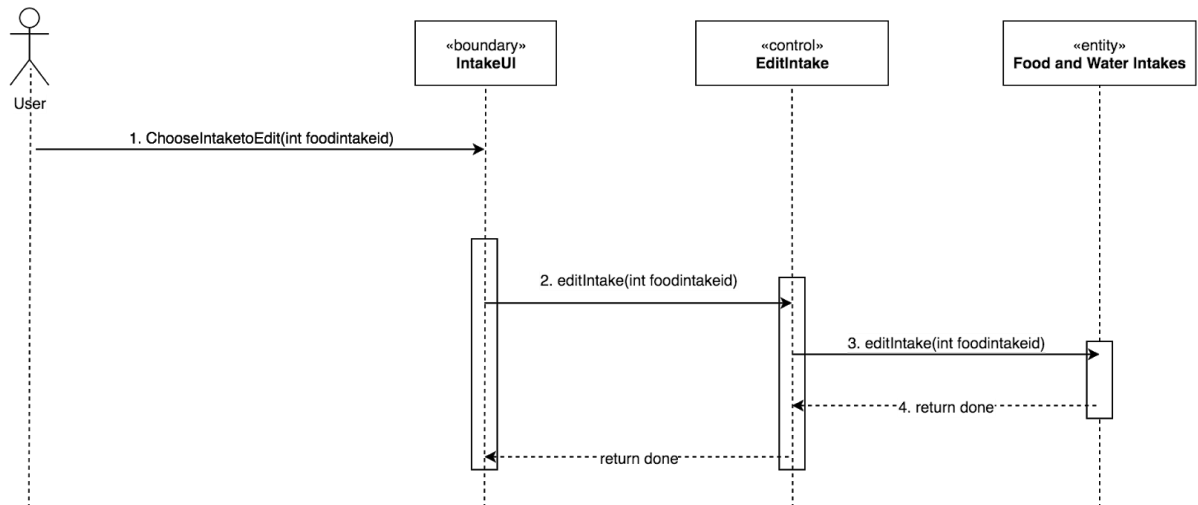


Use-Case Name: Edit Food or Water Intake

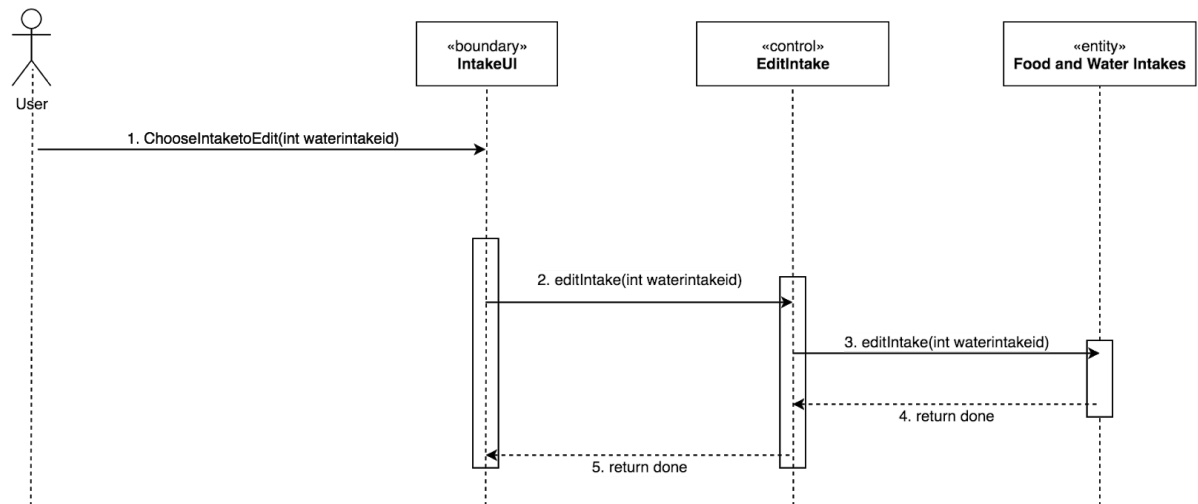
Description:

After the user is done adding a food or a water intake, the user may edit their input. User may also edit their previous intake/s in the records.

Scenario 1: User edits a food intake.



Scenario 2: User edits a water intake.

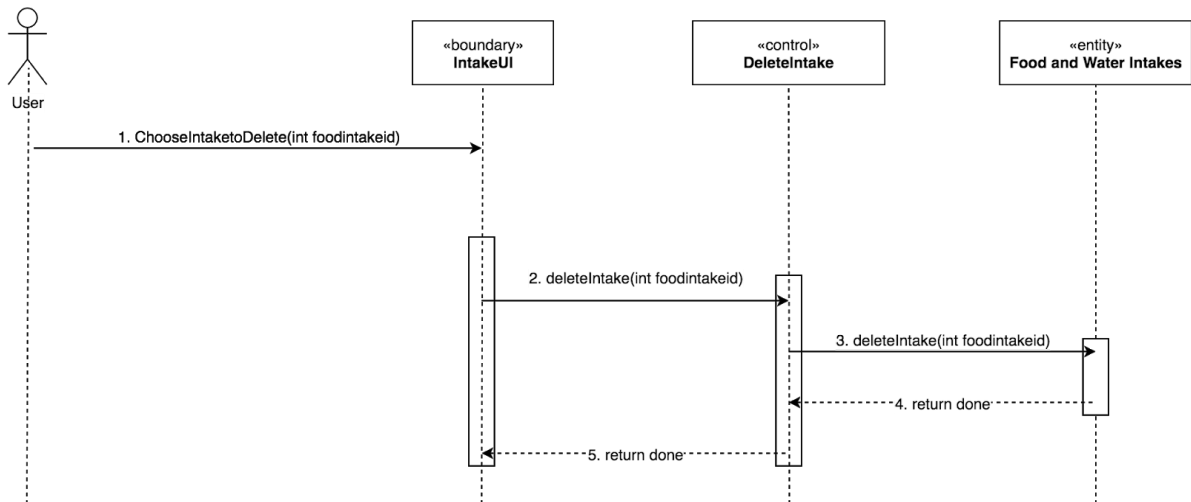


Use-Case Name: Delete Food or Water Intake

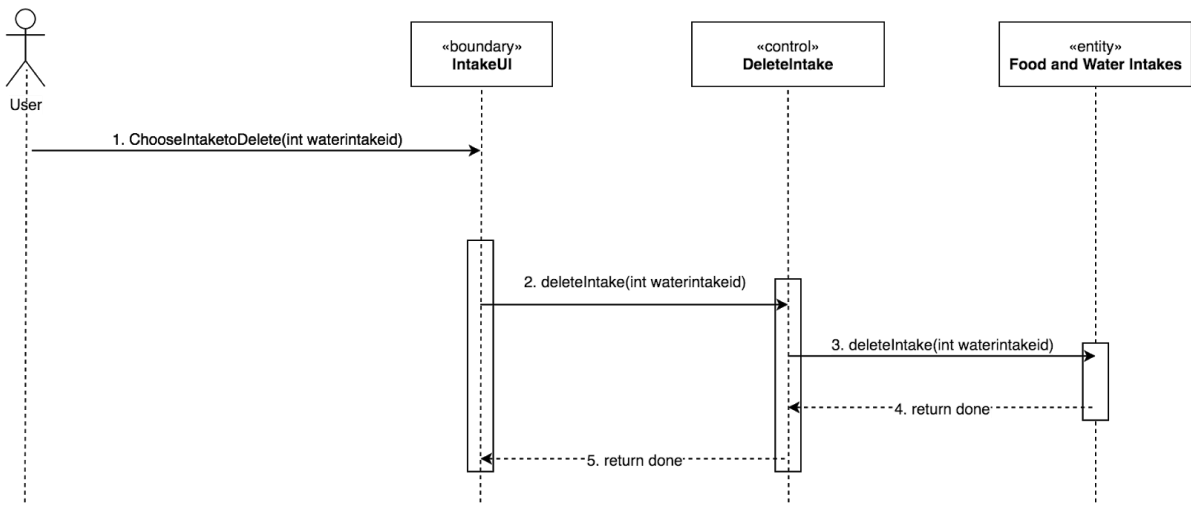
Description:

When an intake isn't needed anymore or accidentally added, all attributes associated with intake may be removed or deleted. User may clean up or declutter their records using this function.

Scenario 1: User deletes a food intake.



Scenario 2: User deletes a water intake.

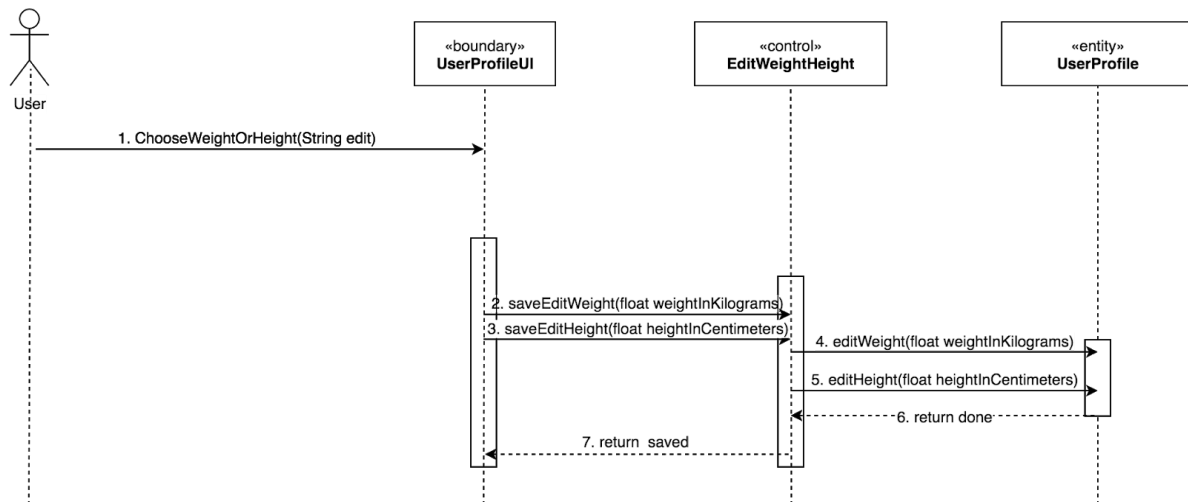


Use-Case Name: Edit Weight or Height

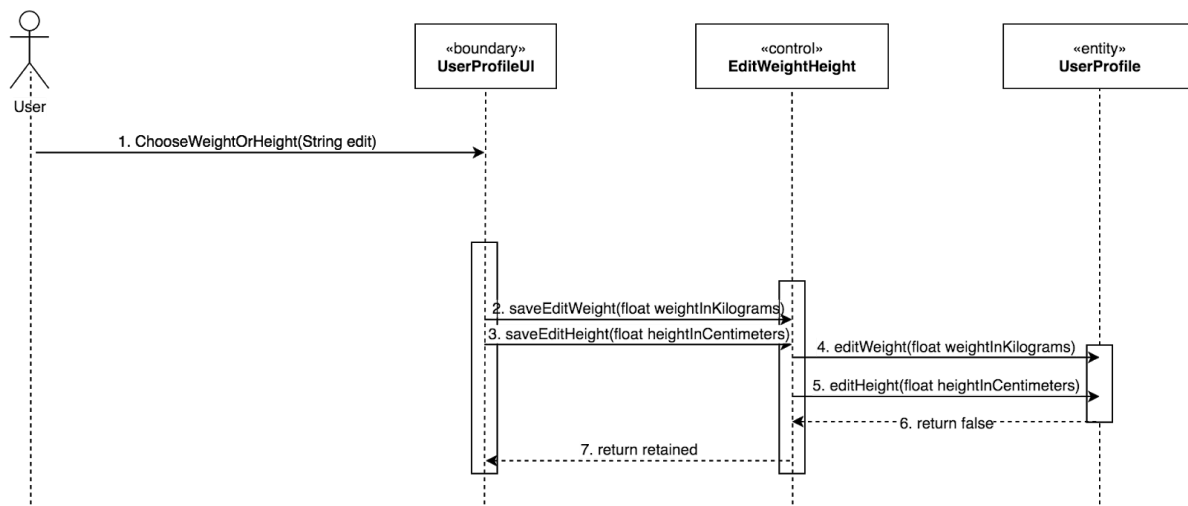
Description:

From the user's given profile, this function can edit his/her height. User can also edit his/her previous weight, because in the management of the system, user tracks his/her growth in terms of height, or gain/loss of user's weight.

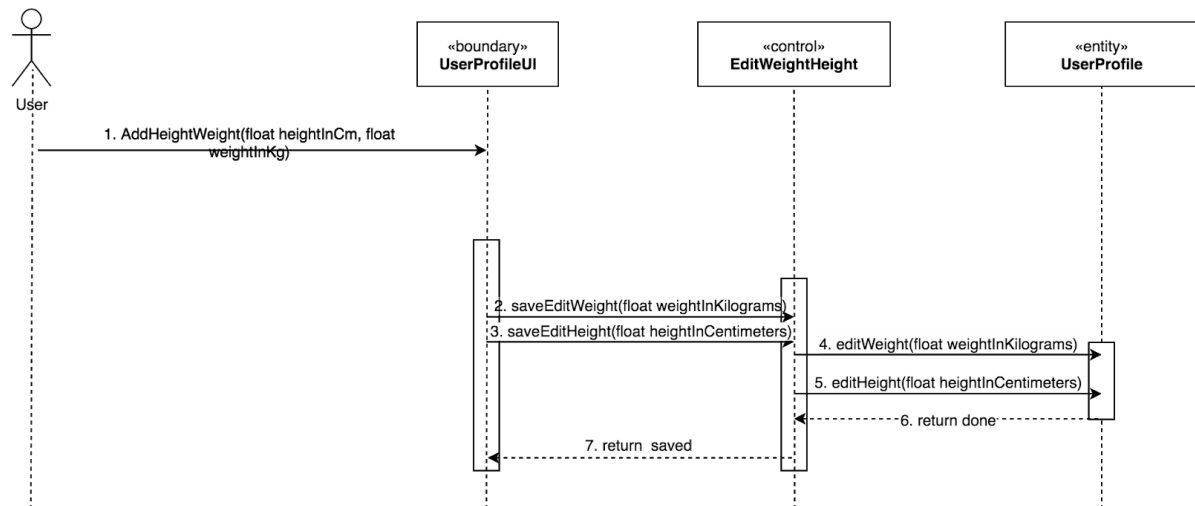
Scenario 1: User edits weight/height successfully



Scenario 2: User edits weight/height unsuccessfully



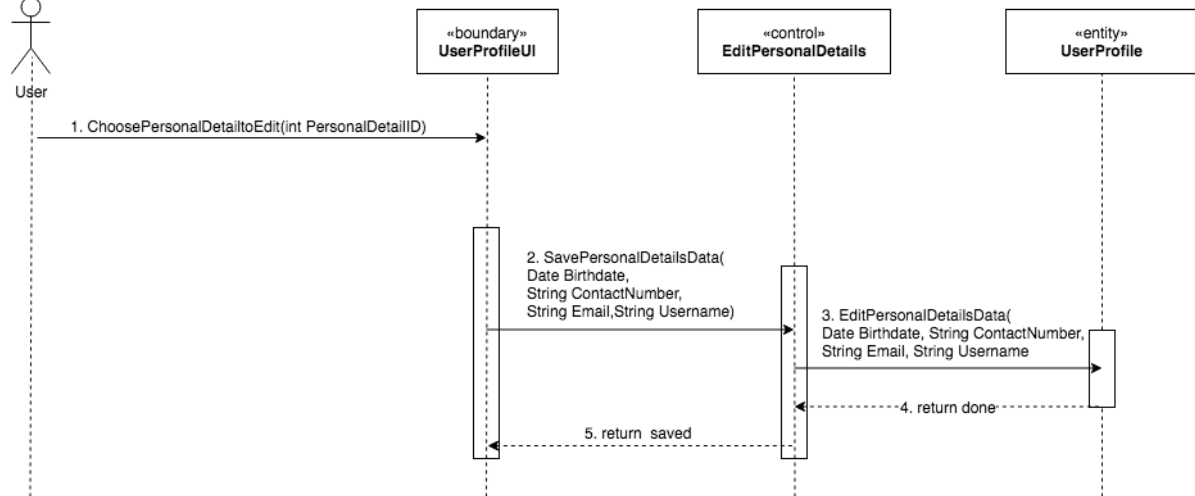
Scenario 3: First time user edits weight/height successfully



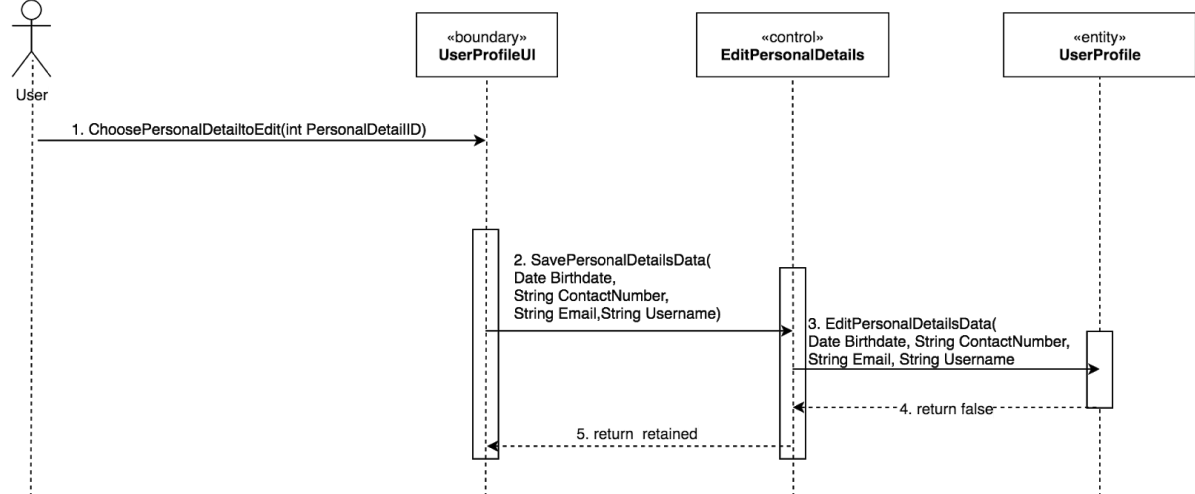
Use-Case Name: Edit Personal Details

Description: Using this function, user may edit the other personal details besides height, weight and other health conditions.

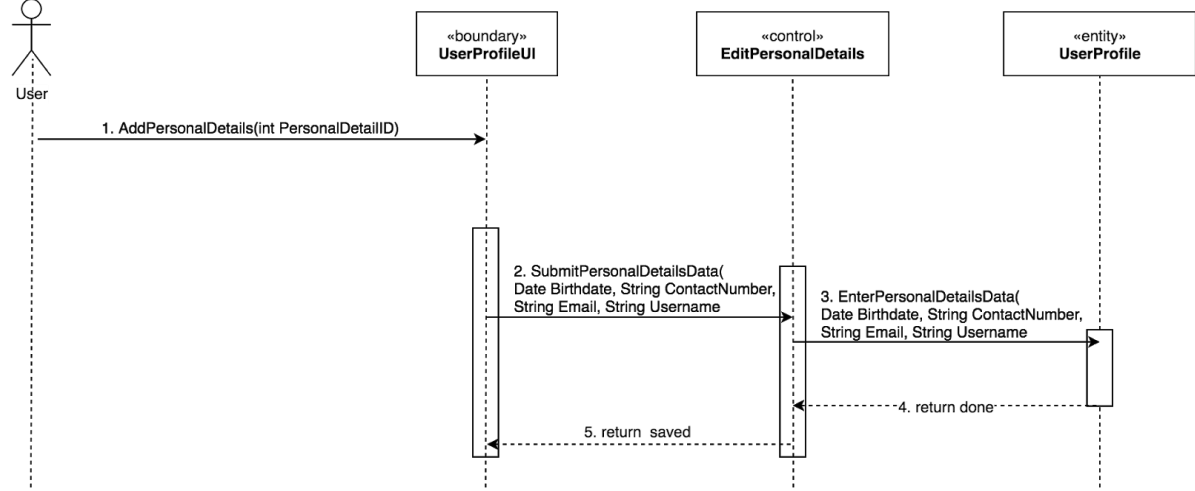
Scenario 1: User edits personal details successfully



Scenario 2: User edits personal details unsuccessfully



Scenario 3: First time user edits personal details successfully

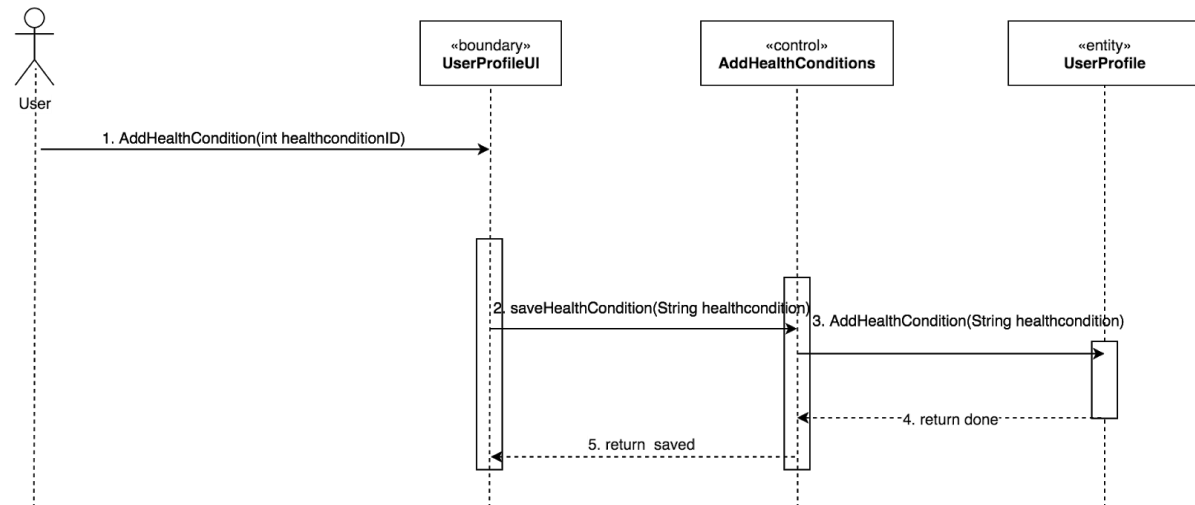


Use-Case Name: Add Health Conditions

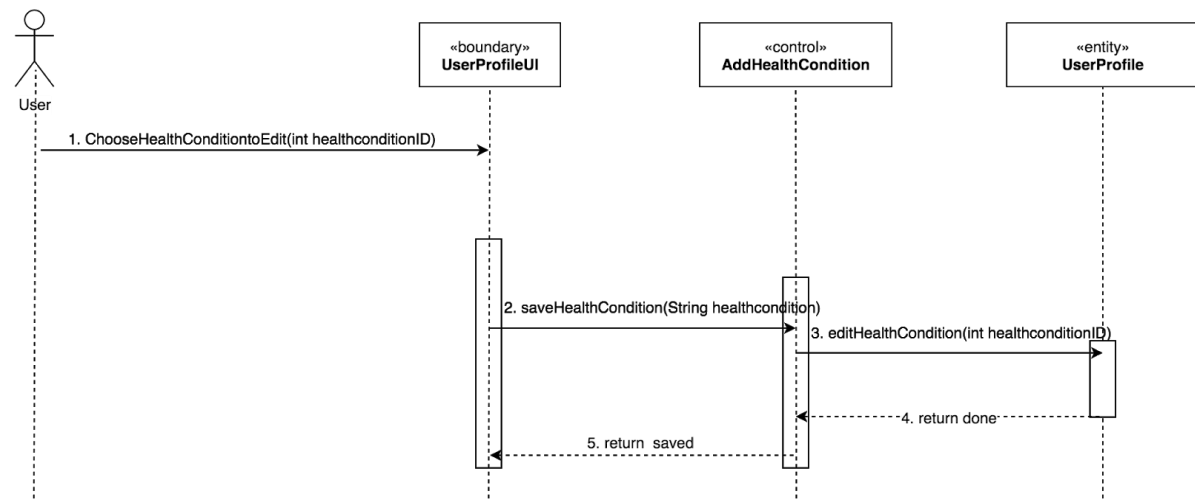
Description:

From the user's given profile, this function can edit his/her health conditions. This function helps the user established his/her health profile. Health conditions may be left blank.

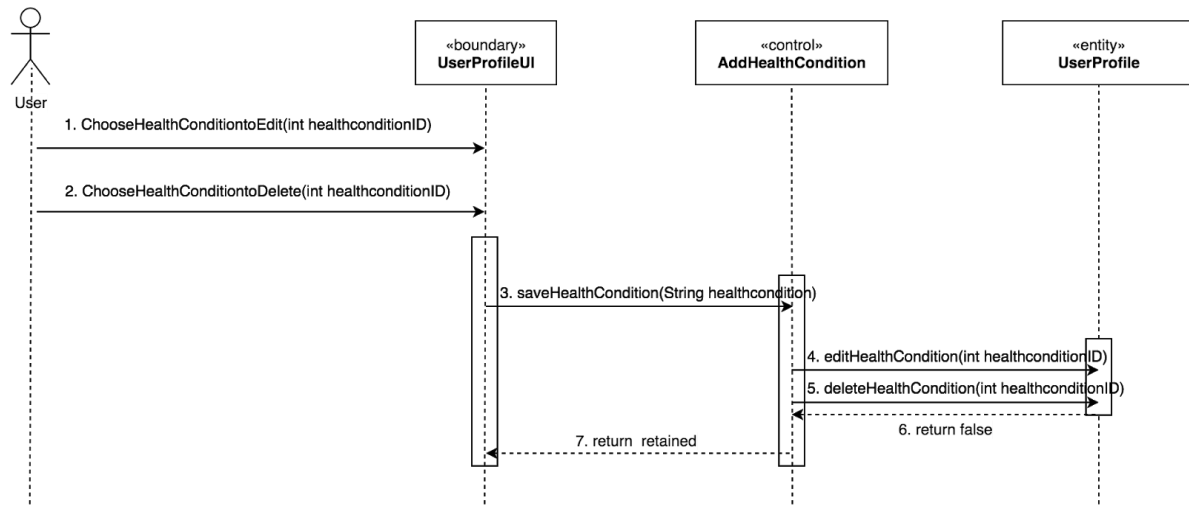
Scenario 1: User adds health condition/s successfully



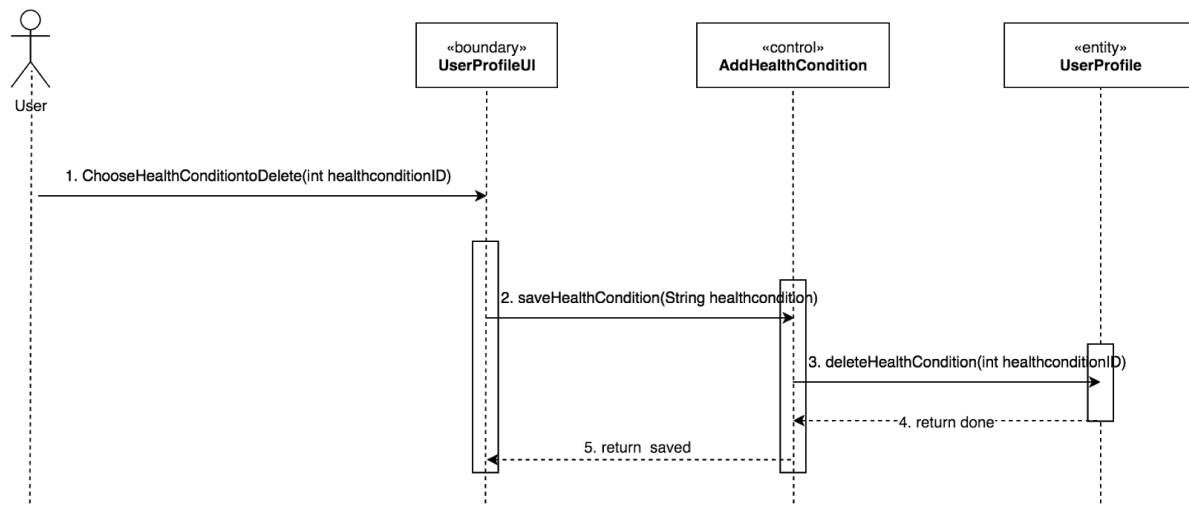
Scenario 2: User edits health condition/s successfully



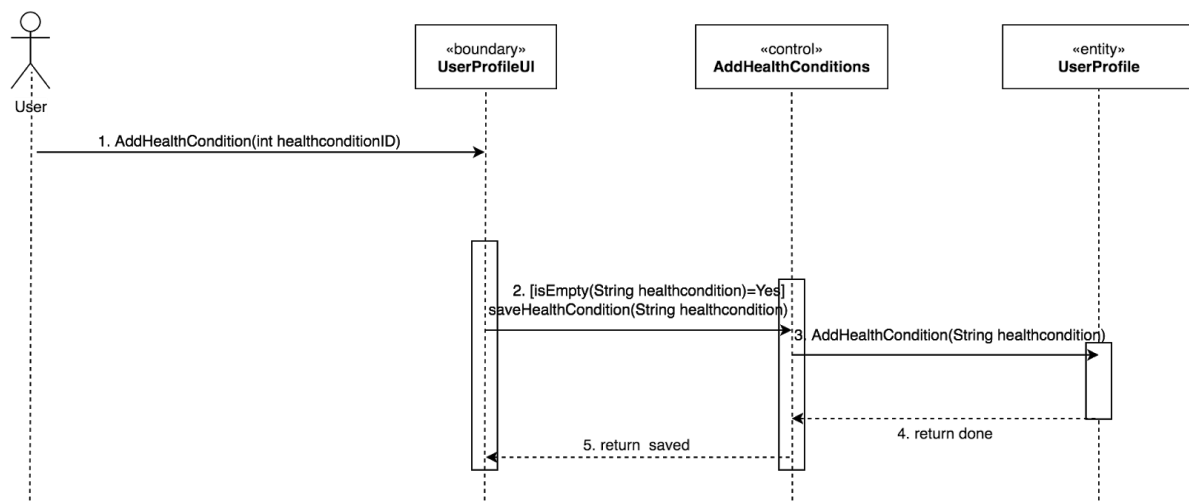
Scenario 3: User edits/deletes health condition/s unsuccessfully



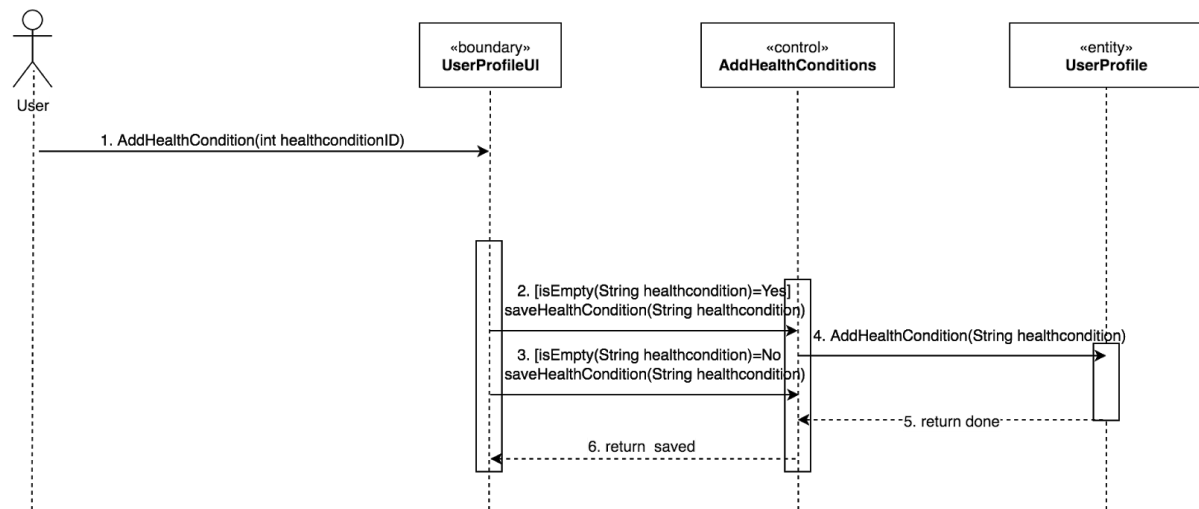
Scenario 4: User deletes health condition/s successfully



Scenario 5: First time user leaves health conditions blank



Scenario 6: First time user adds/edits health condition/s

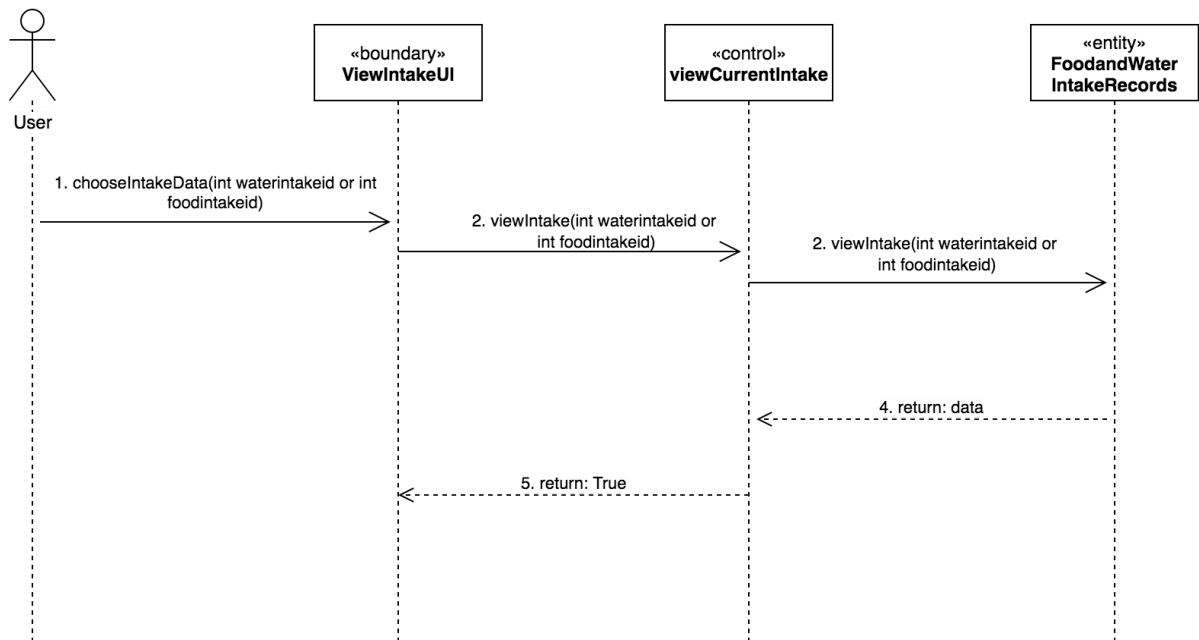


Use-Case Name: View Current Intake

Description:

The user can view his/her own user profile where he/she can see his/her current meal/water intake

Scenario 1: User is able to view all his food or water intake

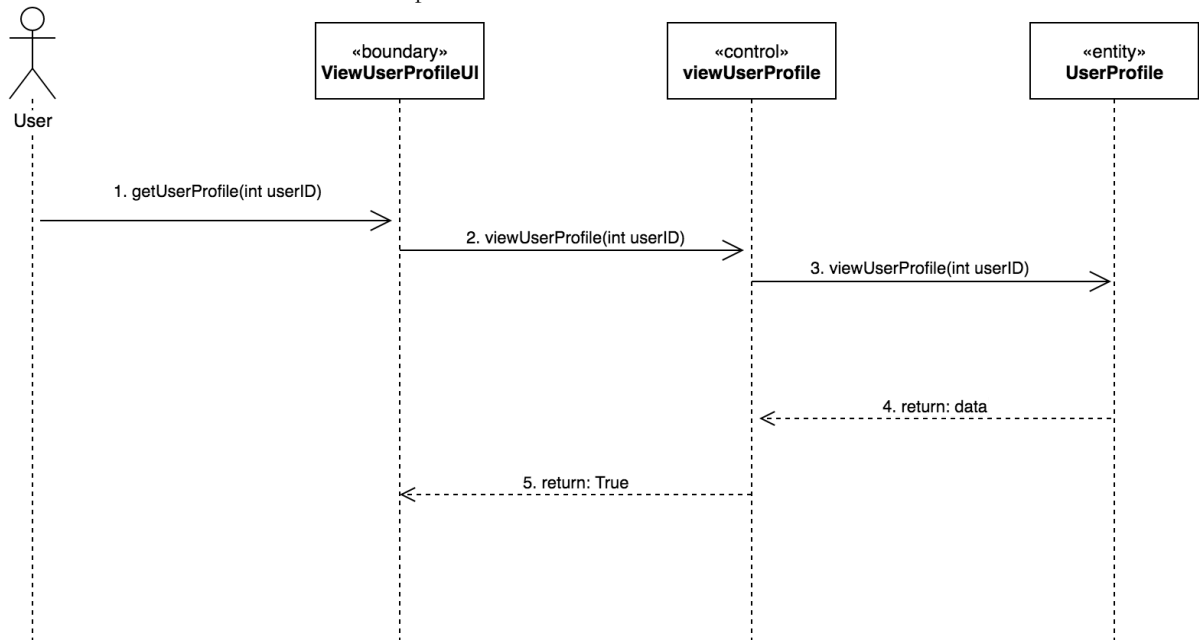


Use-Case Name: View User Profile

Description:

The user can view his/her own user profile where he/she can see his/her current height, weight, health conditions, and other personal details.

Scenario 1: User is able to view his/her user profile.

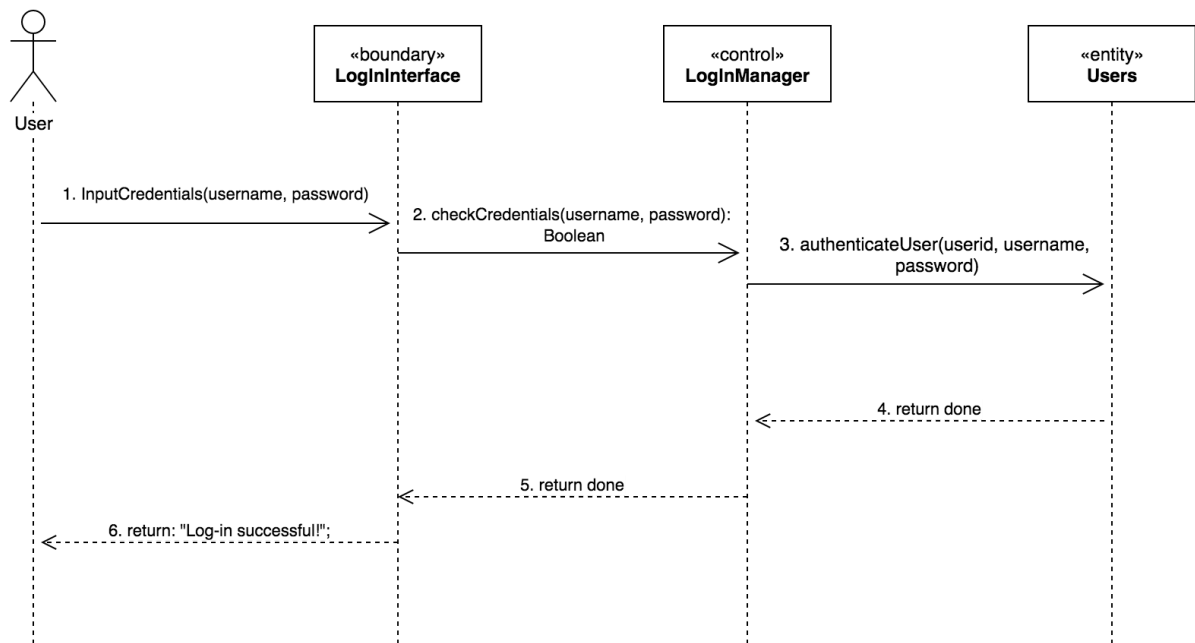


Use-Case Name: Log-in

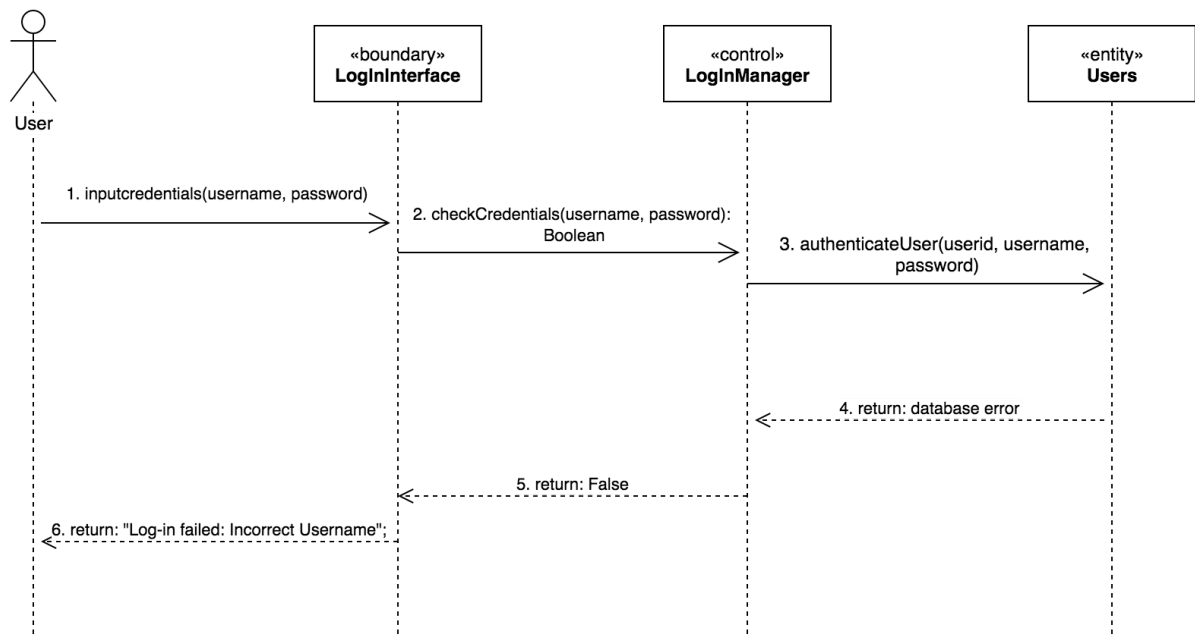
Description:

The user must create his/her own account and login using his/her own account in order to start viewing his health and fitness records and maintain it accordingly.

Scenario 1: User logs his or her username and password successfully.



Scenario 2: User logs username and password, but the entered username is invalid.



Scenario 3: User logs username and password, but the entered password is invalid.

