Use Case Details

Case Use Model

Version ● Proposed

Date/Time Generated: 19/04/2018 12:48:07 p.m.
Author: TOSHIBA

Table of Contents

Case Use Model

Package in package 'Model'

Case Use diagram

Use Case diagram in package 'Case Use Model '

The Use Case model is a catalogue of system functionality described using UML Use Cases. Each Use Case represents a single, repeatable interaction that a user or "actor" experiences when using the system.

A Use Case typically includes one or more "scenarios" which describe the interactions that go on between the Actor and the System, and documents the results and exceptions that occur from the user's perspective.

Use Cases may include other Use Cases as part of a larger pattern of interaction and may also be extended by other use cases to handle exceptional conditions.

Use Case
- + Administrator
- + User
- + Sign In
- + Sign Up
- + Modify User Profile
- + Chose Type of User
- + Log Out
- + Delete Account
- + Account Overview
- + Categorized
- + Edit Expenses
- + Edit Incomes
- + Edit Queue
- + Goals
- + Historical Data
- + Generate a report
- + Payment Queue
- + Request Pool
- + Search a User
- + Send Message
- + Technical Service Request

Actors
- + Administrator
- + User
- + Free User
- + VIP User
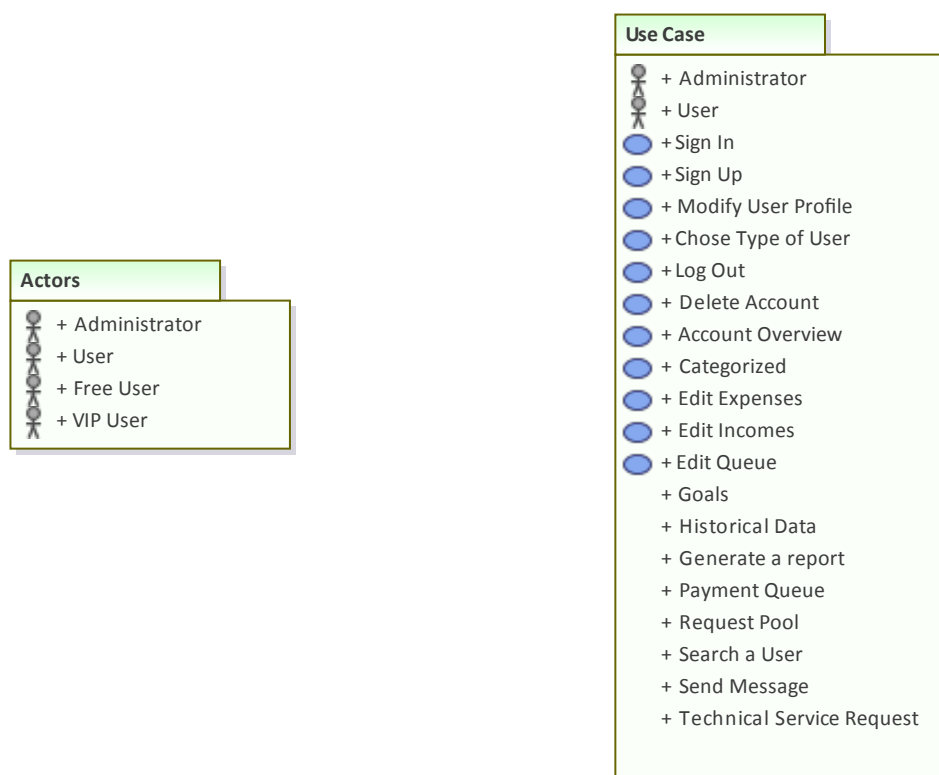
Figure 1: Case Use

Actors

Package in package 'Case Use Model '

Actors diagram

Use Case diagram in package 'Actors'

Actors are the users of the system being modeled. Each Actor will have a well-defined role, and in the context of that role have useful interactions with the system.

A person may perform the role of more than one Actor, although they will only assume one role during one use case interaction.

An Actor role may be performed by a non-human system, such as another computer program.
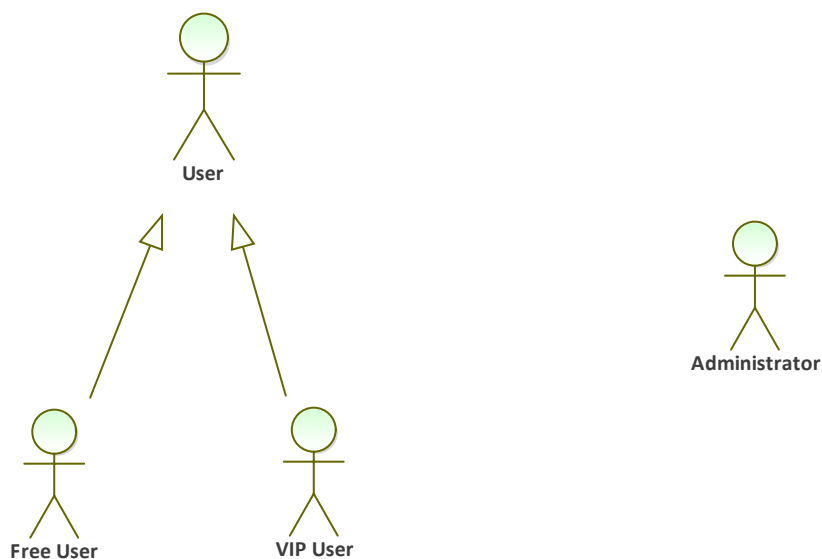
Figure 2: Actors

Use Case

Package in package 'Case Use Model '

<div align="right">

Use Case
Version 1.0 Phase 1.0 Mandatory
Dave created on 31/03/2018. Last modified 17/04/2018

</div>

SPFG diagram

Use Case diagram in package 'Use Case'

This package contains use cases which define how an Actor will interact with the proposed system.

Each interaction may be specified using scenarios, sequence diagrams, communication diagrams and other dynamic diagrams or textual descriptions which together describe how the system, when viewed as a "black-box", interacts with a user.

<div align="right">

SPFG
Version 1.0
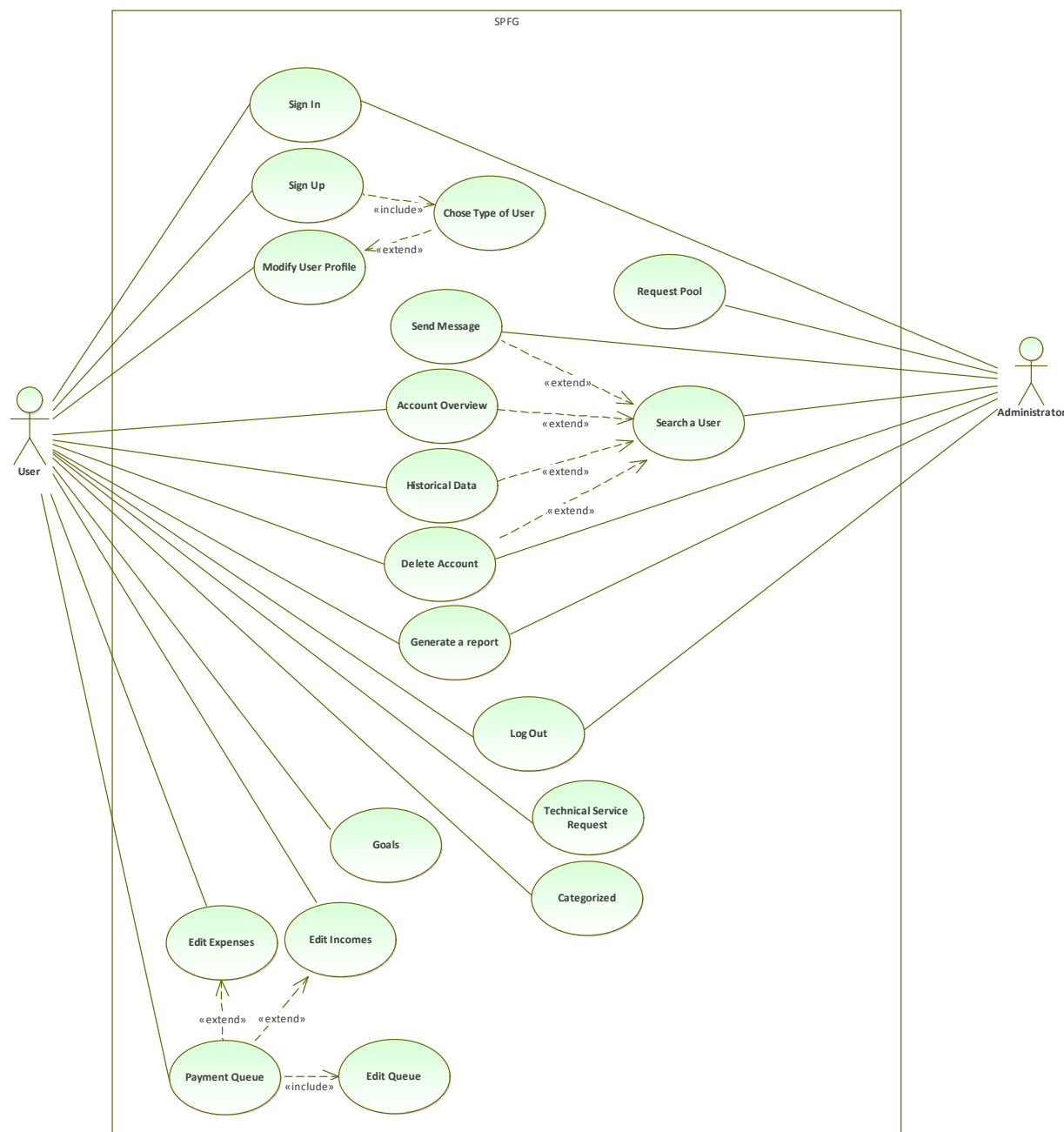TOSHIBA created on 31/03/2018. Last modified 19/04/2018

</div>

Figure 3: SPFG

Sign In

UseCase in package 'Use Case'

Allow Users and Administrators to get into their account. Also, make a differentiation between roles.

The function implemented in the model will recognize those account with a boolean value equal to false as a user. If the boolean value is equal to true, then the account will be open as a administrator account.

It should be noted that this boolean variable is an extra value added to the system users.

Sign In
Version 1.0 Phase 1.0 Mandatory
Dave created on 07/04/2018. Last modified 19/04/2018

CONNECTORS

↗ **UseCaseLink**   Source -> Destination
From: Administrator : Actor, Public
To: Sign In : UseCase, Public

↗ **UseCaseLink**   Source -> Destination
From: User : Actor, Public
To: Sign In : UseCase, Public

Sign Up

UseCase in package 'Use Case'

Register and save data from new users. Besides, it allows you to choose a pricing plan. Finally the incoming data will be save in the database as a new account.

All this stuff will be sent from a form designed to acquired necessary information. Also, this form will incorporate a function to validate incoming information in order to avoid SQL injection.

Sign Up
Version 1.0 Phase 1.0 Mandatory
Dave created on 07/04/2018. Last modified 19/04/2018

CONNECTORS

↗ **Include** «include»   Source -> Destination
From: Sign Up : UseCase, Public
To: Chose Type of User : UseCase, Public

↗ **UseCaseLink**   Source -> Destination
From: User : Actor, Public
To: Sign Up : UseCase, Public

Modify User Profile

UseCase in package 'Use Case'

Allow any kind of user to modify his/her personal data. Administrators are not allowed in this scope. This functionality include an option to choose a pricing plan to get a complete system.

A form is display to receive new information; also, data validation function is applied in this form to avoid any kind of system attack. Finally, changes will be applied in the Database if the information is sent and verified.

Modify User Profile
Version 1.0 Phase 1.0 Mandatory
Dave created on 09/04/2018. Last modified 16/04/2018

CONNECTORS

    ↗ **Extend** «extend»   Source -> Destination
    From: Chose Type of User : UseCase, Public
    To: Modify User Profile : UseCase, Public

    ↗ **UseCaseLink**   Source -> Destination
    From: User : Actor, Public
    To: Modify User Profile : UseCase, Public

Chose Type of User

UseCase in package 'Use Case'

Free users can use this function to upgrade their account as a VIP user. Some pre-requisites are needed to do it: An extra email to ensure a new layer of protection, a phone number to get the notifications function and a reset of your password. Once done, the new VIP user will be set as it. (It requires to restart or reload the application)

VIP user will use this function in a different way. If you want to stop your payment plan, you must deselect the VIP user option. This will deactivate your extra functions and return your account as a free user.

Finally, this option will be unable for a week once you use it in whatever of the explained situation.

<div align="right">

Chose Type of User
Version 1.0 Phase 1.0 Mandatory
Dave created on 09/04/2018. Last modified 16/04/2018

</div>

CONNECTORS

    ↗ **Extend** «extend»   Source -> Destination
    From: Chose Type of User : UseCase, Public
    To: Modify User Profile : UseCase, Public

    ↗ **Include** «include»   Source -> Destination
    From: Sign Up : UseCase, Public
    To: Chose Type of User : UseCase, Public

Log Out

UseCase in package 'Use Case'

Close the actual session. There's not difference between users and administrators. Once done the system user will be redirected to the login page.

<div align="right">

Log Out
Version 1.0 Phase 1.0 Mandatory
TOSHIBA created on 09/04/2018. Last modified 16/04/2018

</div>

CONNECTORS

⬈ **UseCaseLink**   Source -> Destination
From: User : Actor, Public
To: Log Out : UseCase, Public

⬈ **UseCaseLink**   Source -> Destination
From: Administrator : Actor, Public
To: Log Out : UseCase, Public

Delete Account

UseCase in package 'Use Case'

Change the visibility of the account. Also, this function save the date when an account was closed. An erased account can be seen from an administrator account but it won´t count for a new user who's registering with the same data (Specially in duplicated Emails). The purpose of this function is to create a table to see the application performance.

An extra plus from this function is allow users to recover a deleted account as long as a new account won't have been created with the same data. (Could happen in particular situations). In that case, the deleted account will be irrecoverable.

<div align="right">

Delete Account
Version 1.0 Phase 1.0 Proposed
Dave created on 09/04/2018. Last modified 19/04/2018

</div>

CONNECTORS

⬈ **Extend** «extend»   Source -> Destination
From: Delete Account : UseCase, Public
To: Search a User : UseCase, Public

⬈ **UseCaseLink**   Source -> Destination
From: Administrator : Actor, Public
To: Delete Account : UseCase, Public

⬈ **UseCaseLink**   Source -> Destination
From: User : Actor, Public
To: Delete Account : UseCase, Public

Account Overview

UseCase in package 'Use Case'

Make an overview of an account. Latest incomes and expenses will be summarized next to a special diagram* with an option to get more detailed information. The goals segment is included in this section. Furthermore, the actual user amount is shown in a format color that shown the balance between incomes and expenses.

This function is better summarized as a organizational tool rather than a functional tool. The active part could be found in goals management, but we will do it in a new use case.

Special Diagram: It's called special because it could be manage with different date ranges. A daily range will show differences in the same day from two different weeks. A weekly range will compare two followed weeks and a monthly range will make a comparison from two continuous months.

<div align="right">

Account Overview
Version 1.0 Phase 1.0 Mandatory
Dave created on 09/04/2018. Last modified 19/04/2018

</div>

CONNECTORS

   ↗ **Extend** «extend»   Source -> Destination
      From: Account Overview : UseCase, Public
      To: Search a User : UseCase, Public

   ↗ **UseCaseLink**   Source -> Destination
      From: User : Actor, Public
      To: Account Overview : UseCase, Public

Categorized

UseCase in package 'Use Case'

Allow users to organize any kind of incomes and expenses in a category. A category is defined under an ID that represent a list from different Incomes/Expenses ID's. A category can be added, modified or deleted. Also, the system won't allow categories with the same name.

By default the system will provide you with a Category called "Others"*. This category can´t be deleted since it's the pool for uncategorized incomes/expense. Also, categories could have the same name if they're not in the same environment**.

* Obviously, there would be a default category for Expenses and a default category for Incomes.

**Environments: Expenses and Incomes.

<div align="right">

Categorized
Version 1.0 Phase 1.0 Mandatory
Dave created on 16/04/2018. Last modified 16/04/2018

</div>

CONNECTORS

   ↗ **UseCaseLink**   Source -> Destination
      From: User : Actor, Public
      To: Categorized : UseCase, Public

Edit Expenses

UseCase in package 'Use Case'

Complex functionality to add or delete expenses. An expense can be added or deleted but not modify. Each expense is registered in a master table with the user's ID and Expense's ID. Deleted expenses are stored during two months to verify application performance; also, deleted expenses will be differenced from normal expenses through a boolean state that indicate income visibility.

An expense is describe as follows: ID (Automatic), Name, amount, categorization (Not necessary, by default it will be added to "Others"), payment plan (Not necessary, by default it will be unique) and a date (Automatic).

<div align="right">

Edit Expenses
Version 1.0 Phase 1.0 Mandatory
Dave created on 09/04/2018. Last modified 16/04/2018

</div>

CONNECTORS

↗ **Extend** «extend»   Source -> Destination
From: Payment Queue : UseCase, Public
To: Edit Expenses : UseCase, Public

↗ **UseCaseLink**   Source -> Destination
From: User : Actor, Public
To: Edit Expenses : UseCase, Public

Edit Incomes

UseCase in package 'Use Case'

Complex functionality to add or delete incomes. An income can be added or deleted but not modify. Each income is registered in a master table with the user's ID and Income's ID. Deleted incomes are stored during two months to verify application performance; also, deleted incomes will be differenced from normal incomes through a boolean state that indicate income visibility.

An income is describe as follows: ID (Automatic), Name, amount, categorization (Not necessary, by default it will be added to "Others"), payment plan (Not necessary, by default it will be unique) and a date (Automatic).

<div align="right">

Edit Incomes
Version 1.0 Phase 1.0 Mandatory
Dave created on 09/04/2018. Last modified 16/04/2018

</div>

CONNECTORS

↗ **UseCaseLink**   Source -> Destination
From: User : Actor, Public
To: Edit Incomes : UseCase, Public

↗ **Extend** «extend»   Source -> Destination
From: Payment Queue : UseCase, Public
To: Edit Incomes : UseCase, Public

Edit Queue

UseCase in package 'Use Case'

Allow users to add or delete a payment plan. This function work with incomes categorized by users. An income have an Income ID so it could added in the queue with a timer. Incomes with the same name won't be problem since Income ID is the identifier.

Edit Queue
Version 1.0 Phase 1.0 Mandatory
Dave created on 16/04/2018. Last modified 16/04/2018

CONNECTORS
Include «include»   Source -> Destination From: Payment Queue : UseCase, Public To: Edit Queue : UseCase, Public

Goals

UseCase in package 'Use Case'

Users can created a goal as a future objective. A goal acts as a kind of timer that shown your progress to buy different things. A goal is reached when the system send you a notification. Also, you can delete or modify a goal.

The structure of a goal is as follow: Goal ID (Automatic), name, amount, state*,date of creation (Automatic) and "final day" (Closed when the objective is reached).

*State: In Progress, Completed.

Goals
Version 1.0 Phase 1.0 Mandatory
Dave created on 16/04/2018. Last modified 16/04/2018

CONNECTORS
UseCaseLink   Source -> Destination From: User : Actor, Public To: Goals : UseCase, Public

Historical Data

UseCase in package 'Use Case'

Make an exhaustive analysis from the last weeks, months and even years. The balance from each day will be shown in different colors. The Historical Data will be organized by months with a balance defined by day. When a day is clicked, a modal will appear with the specific information from that day*.

*The information from an empty day won't be found so it will appear with gray color.

CONNECTORS

↗ **Extend** «extend»   Source -> Destination
From: Historical Data : UseCase, Public
To: Search a User : UseCase, Public

↗ **UseCaseLink**   Source -> Destination
From: User : Actor, Public
To: Historical Data : UseCase, Public

Generate a report

UseCase in package 'Use Case'

Users can generate a report from the last n-months. It will included an option to add diagrams and graphics related to the last incomes and expenses.

Administrator report function will be a little more simple. The generated document will just include a simple tracing from the last n-activities done in the system.

CONNECTORS

↗ **UseCaseLink**   Source -> Destination
From: Administrator : Actor, Public
To: Generate a report : UseCase, Public

↗ **UseCaseLink**   Source -> Destination
From: User : Actor, Public
To: Generate a report : UseCase, Public

Payment Queue

UseCase in package 'Use Case'

Implement a queue to establish a payment plan. For example, if you have to pay the internet service each month, it will add this expense to the queue; so, when the time is right, the expense will be automatic registered. This function can be used with your notification system.

CONNECTORS

↗ **Extend** «extend»   Source -> Destination
From: Payment Queue : UseCase, Public
To: Edit Expenses : UseCase, Public

↗ **Include** «include»   Source -> Destination
From: Payment Queue : UseCase, Public
To: Edit Queue : UseCase, Public

↗ **Extend** «extend»   Source -> Destination
From: Payment Queue : UseCase, Public
To: Edit Incomes : UseCase, Public

↗ **UseCaseLink**   Source -> Destination
From: User : Actor, Public
To: Payment Queue : UseCase, Public

Request Pool

UseCase in package 'Use Case'

The request pool is a place where all technical request are stored.
There is just one request pool. Only administrators will have the access to this functionality. If a technical request is solved, it will be marked as "Solved". Subsequently the system will record it and add the administrator's id next to the date of the solution.

Request Pool
Version 1.0 Phase 1.0 Proposed
Dave created on 09/04/2018. Last modified 12/04/2018

CONNECTORS

↗ **UseCaseLink**   Source -> Destination
From: Administrator : Actor, Public
To: Request Pool : UseCase, Public

Search a User

UseCase in package 'Use Case'

An exclusive administrator function. Search for a user based in ID account, username or email. The function is implemented in the model part as a Query request to the database.

Search a User
Version 1.0 Phase 1.0 Mandatory
Dave created on 09/04/2018. Last modified 16/04/2018

CONNECTORS

   ↗ **UseCaseLink**   Source -> Destination
       From: Administrator : Actor, Public
       To: Search a User : UseCase, Public

   ↗ **Extend** «extend»   Source -> Destination
       From: Delete Account : UseCase, Public
       To: Search a User : UseCase, Public

   ↗ **Extend** «extend»   Source -> Destination
       From: Send Message : UseCase, Public
       To: Search a User : UseCase, Public

   ↗ **Extend** «extend»   Source -> Destination
       From: Historical Data : UseCase, Public
       To: Search a User : UseCase, Public

   ↗ **Extend** «extend»   Source -> Destination
       From: Account Overview : UseCase, Public
       To: Search a User : UseCase, Public

Send Message

UseCase in package 'Use Case'

The administrator can send a message to different user accounts. The incorporation of this function is suggested to contact users in case of problems with their accounts. (Such us Identity thief, Application bugs, etc...)

<div align="right">

Send Message
Version 1.0 Phase 1.0 Proposed
Dave created on 09/04/2018. Last modified 16/04/2018

</div>

CONNECTORS

   ↗ **Extend** «extend»   Source -> Destination
       From: Send Message : UseCase, Public
       To: Search a User : UseCase, Public

   ↗ **UseCaseLink**   Source -> Destination
       From: Administrator : Actor, Public
       To: Send Message : UseCase, Public

Technical Service Request

UseCase in package 'Use Case'

Send a request to the administrator in case of technical problem or bugs. It's known as the "unique contact point".

<div align="right">

Technical Service Request
Version 1.0 Phase 1.0 Proposed
Dae created on 09/04/2018. Last modified 17/04/2018

</div>

CONNECTORS

UseCaseLink   Source -> Destination
From: User : Actor, Public
To: Technical Service Request : UseCase, Public