Systems Modelling
Fall 2014/15
Homework 2

Janno Siim, Kristjan-Julius Laak

October 4, 2014

1 Use-case diagram

Use cases diagrams are useful for illustrating how a system interacts with outside actors (Referenced from Lecture slides). The actor interacts with the system by some functionalities assigned to the actor. In the restaurant game, the only outside world actor is the player. Once the game starts, the player is the Owner of the restaurant. See Figure 1 below for the use case diagram illustrating the functionalities assigned to the Owner.

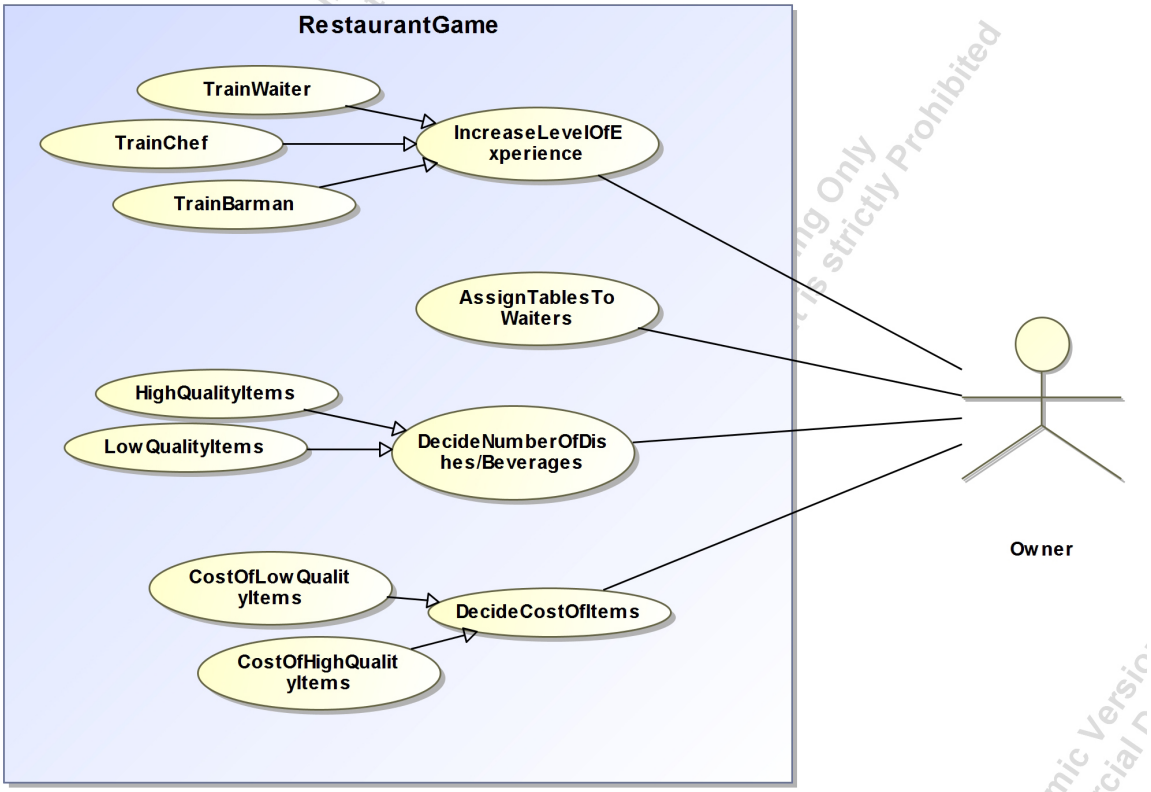

Figure 1:

2 Scenarios

2.1 Use Case : Assign tables for Waiters

Summary : The Owner decides the number of tables to be assigned to each Waiter.

Actors : Owner

Preconditions : It is the beginning of a new day and the Owner has not yet assigned tables-to-be-served for the Waiters.

Description : In the beginning of each day the restaurant is open the Owner collects the numbers of each table in the restaurant and the names of each Waiter. Next,

the Owner assignes each table to exactly one Waiter by the constraint that each Waiter can serve at most 3 tables, i.e. when already 3 tables are assigned to a particular Waiter, no more tables can be assigned to the Waiter.

Exceptions : The leg of one table is broken and there are only 8 tables in the restaurant. The Owner assigns the tables to each and only one Waiter such that no Waiter serves more than 3 tables. One Waiter servers only 2 tables.

Postconditions : Each Waiter is assigned to at most 3 tables.

2.2 Use Case : Cost of high-quality items

Summary : Owner decides the cost of high quality items.

Actors : Owner

Preconditions : Owner has opened the price changing view.

Description : Owner enters a single price for all the high quality items into a high quaility price field. Number is entered in a correct format. Price of the high quality items will be changed.

Exceptions : Wrong format: Owner enters the new price for high quality items. put it is in the wrong format. Wrong format error is shown to the owner. High quality items get back its original value.

Postconditions : Game continues working with the updated price value of high quality items.

2.3 Use Case : Cost of low-quality items

Summary : Owner decides the cost of low quality items.

Actors : Owner

Preconditions : Owner has opened the price changing view.

Description : Owner enters a single price for all the low quality items into a low quaility price field. Number is entered in a correct format. Price of the low quality items will be changed.

Exceptions : Wrong format: Owner enters the new price for low quality items put it is in the wrong format. Wrong format error is shown to the owner. Low quality items get back its original value.

Postconditions : Game continues working with the updated price value of low quality items.

2.4 Use Case : Decide nr of high-quality items

Summary : Owner picks a high quality item for the menu.

Actors : Owner

Preconditions : Menu view has been opened.

Description : Owner is in the menu view and he sees the possible low and high quality items that can be picked for the menu. Also he sees the contents of the menu. Owner picks a high quality item and it is inserted to the menu.

Exceptions : Cancel: Owner chooses an high quality item but clicks cancel button on the menu and all the changes to the menu will be reversed.

Postconditions : Owner can do another operation with the menu.

2.5 Use Case : Decide nr of low-quality items

Summary : Owner picks a low quality item for the menu.

Actors : Owner

Preconditions : Menu view has been opened.

Description : Owner is in the menu view and he sees te possible low and high quality items that can be picked for the menu. Also he sees the contents of the menu. Owner picks a low quality item and it is inserted to the menu.

Exceptions : Cancel: Owner chooses an item but clicks cancel button on the menu and all the changes to the menu will be reversed.

Postconditions : Owner can do another operation with the menu.

2.6 Use Case : Train Barman

Summary : The Owner orders a training course to increase the level of experience of a Barman.

Actors : Owner

Preconditions : The Owner decides to increase the level of experience of a Barman.

Description : The Owner checks whether the budget is sufficient for ordering a training of value of 1200 euros. The Owner orders a training course that costs in total 1200 euros.

Exceptions : The Owner checks the budget and finds that there is not enough finances to order a training course for a Barman in total value of 1200 euros.

Postconditions : The level of experience of the Barman who participated in the training course increases by one level (from low to medium or from medium to high level of experience).

2.7 Use Case : Train Chef

Summary : The Owner orders a training course to increase the level of experience of a Chef.

Actors : Owner

Preconditions : he Owner decides to increase the level of experience of the Chef.

Description : The Owner checks whether the budget is sufficient for ordering a training of value of 1200 euros. The Owner orders a training course that costs in total 1200 euros.

Exceptions : The Owner checks the budget and finds that there is not enough finances to order a training course for the Chef in total value of 1200 euros.

Postconditions : The level of experience of the Chef increases by one level (from low to medium or from medium to high level of experience).

2.8 Use Case : Train Waiter

Summary : he Owner orders a training course to increase the level of experience of a waiter.

Actors : Owner

Preconditions : The Owner decides to increase the level of experience of a Waiter.

Description : The Owner checks whether the budget is sufficient for ordering a training of value 800 euros. The Owner orders a training course that costs in total 800 euros.

Exceptions : he Owner checks the budget and finds that there is not enough finances to order a training course for a Waiter in total value of 800 euros.

Postconditions : The level of experience of the Waiter who participated the training course increases by one level (from low to medium or from medium to high level of experience).

3 Sequence diagrams

To show the interaction of the restaurant game system with the Owner (i.e. the player), we constructed sequence diagrams (See Fig 2-7) A sequence diagram shows the interaction of a system with its actors to perform a use case

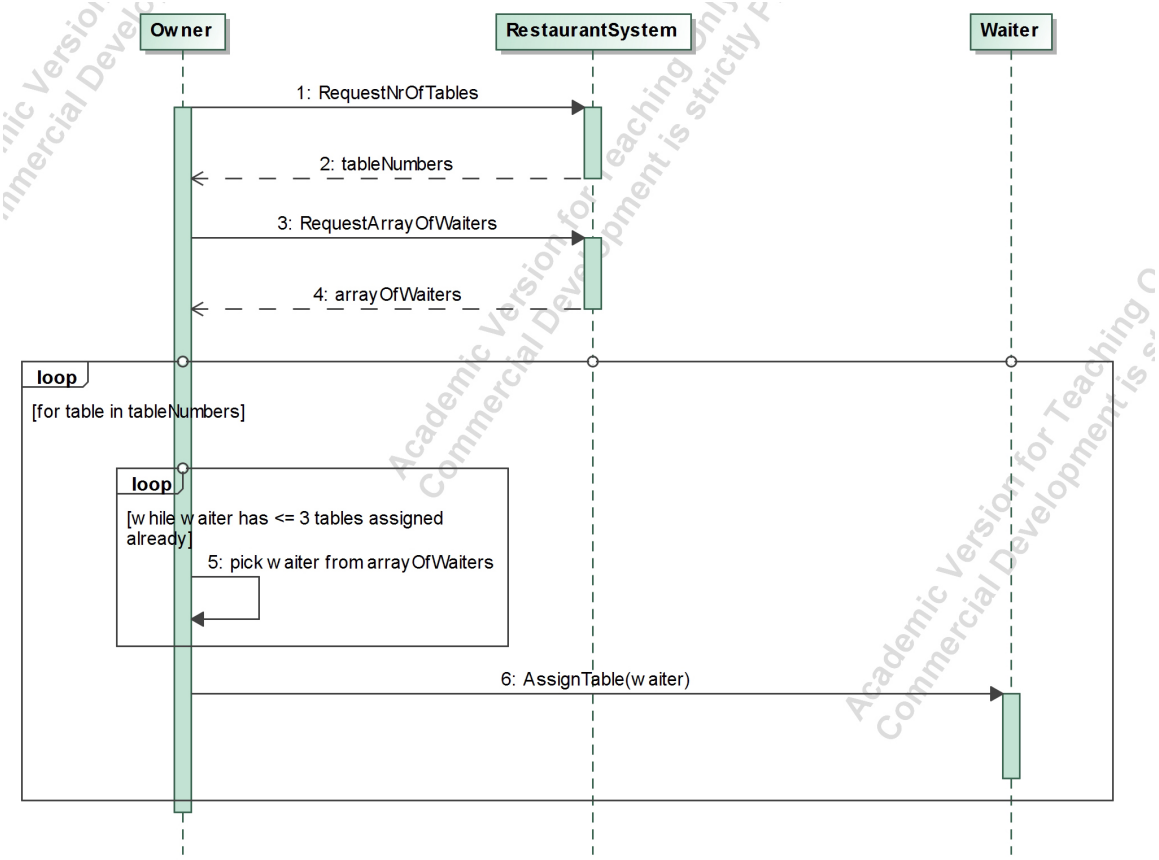

Figure 2: Sequence diagram showing interaction when Owner assigns tables to Waiters.

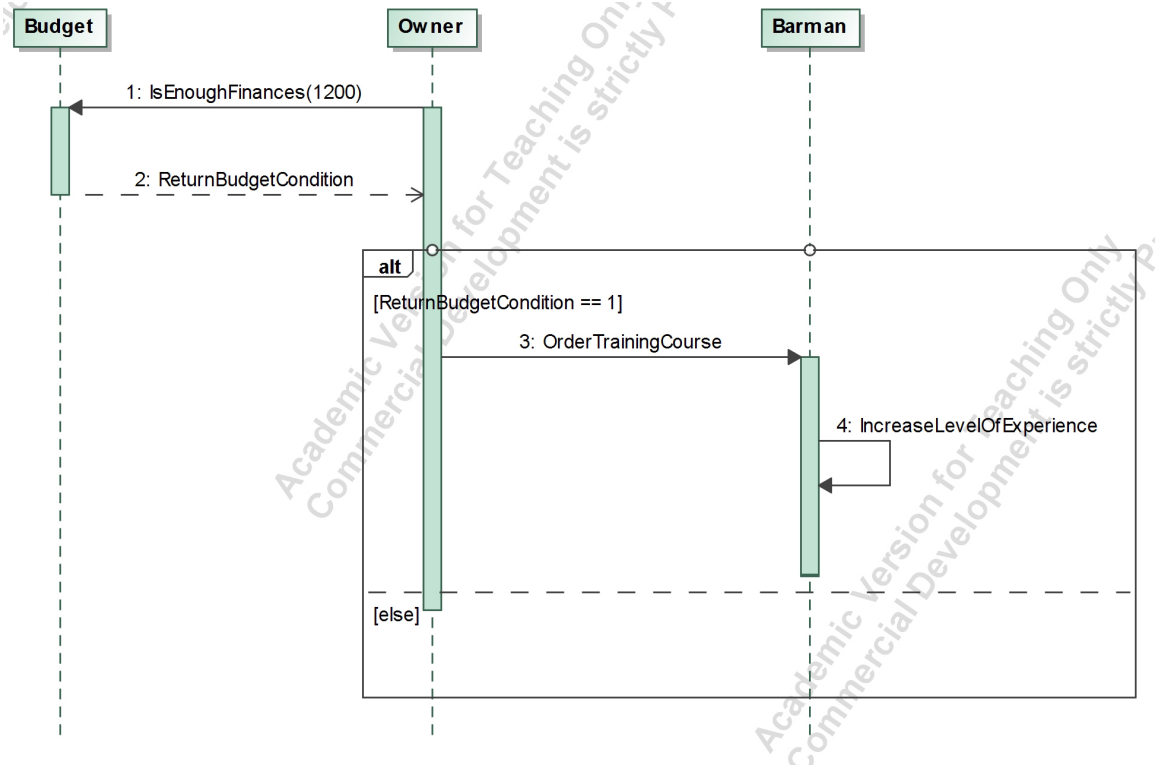

Figure 3: Sequence diagram showing activities for increasing the level of experience of the Barman

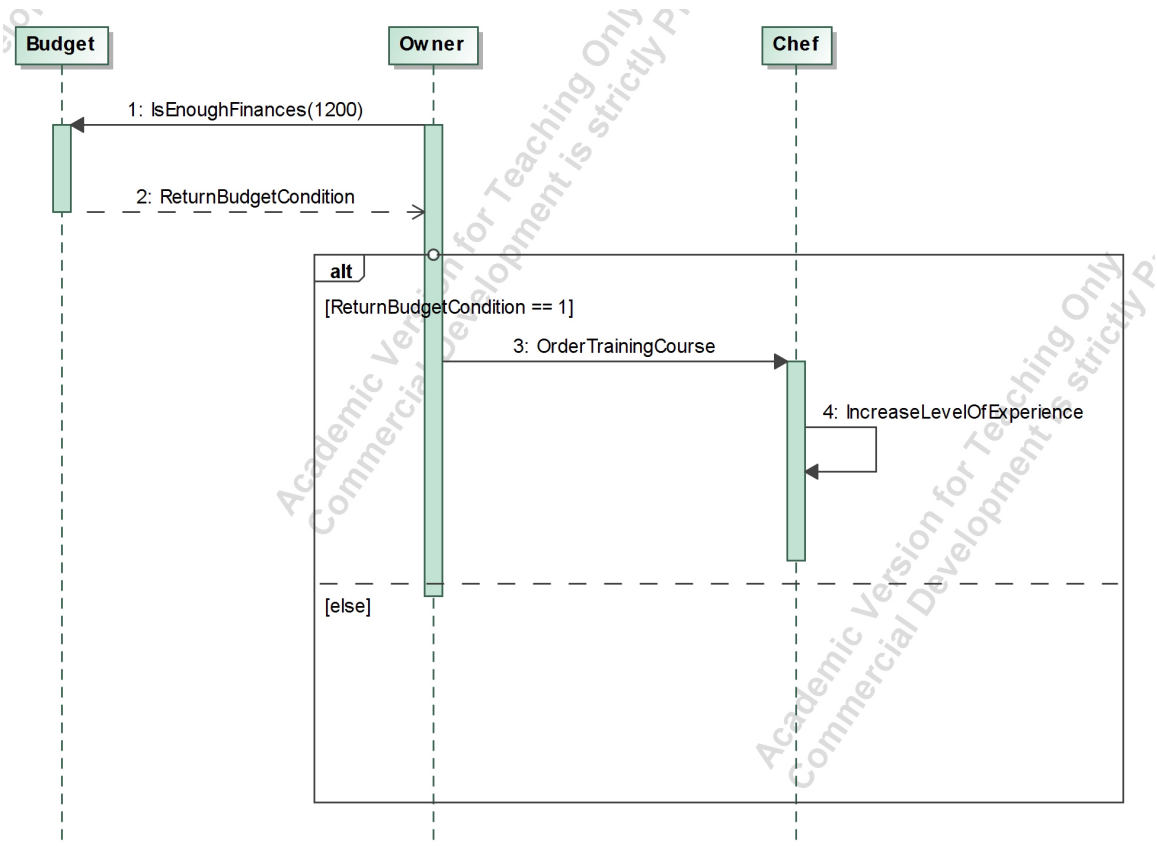

Figure 4: Sequence diagram showing activities for increasing the level of experience of the Chef

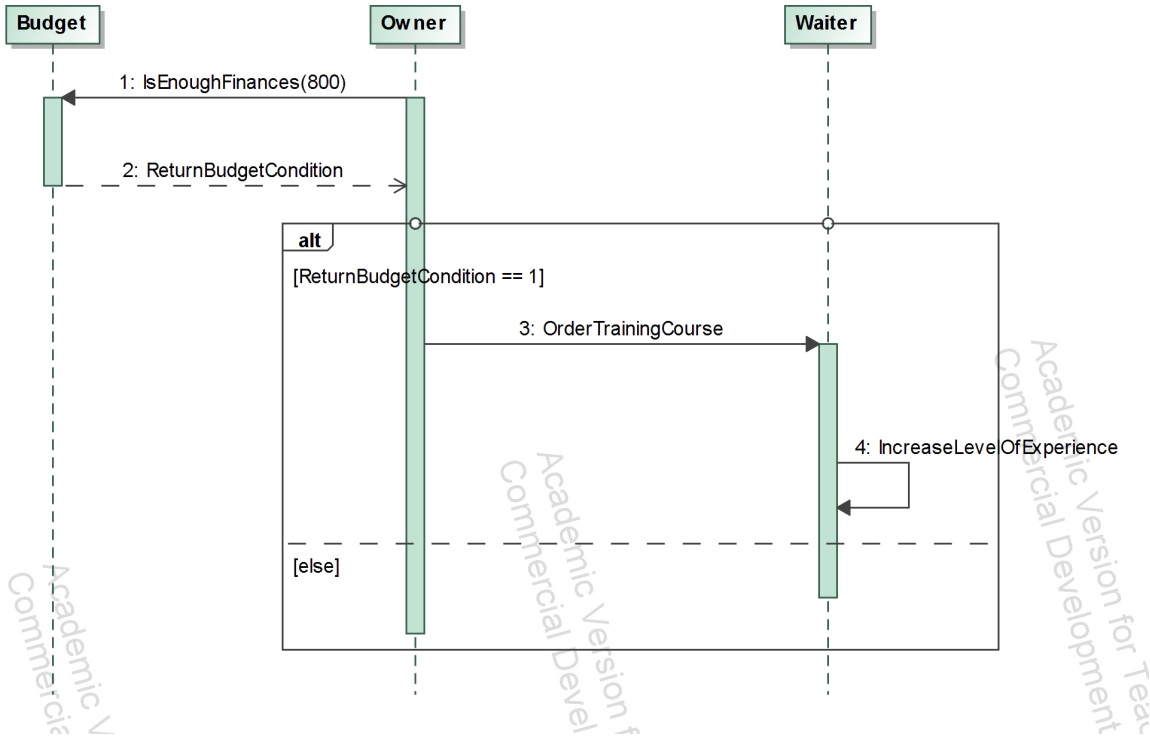

Figure 5: Sequence diagram showing activities for increasing the level of experience of a Waiter

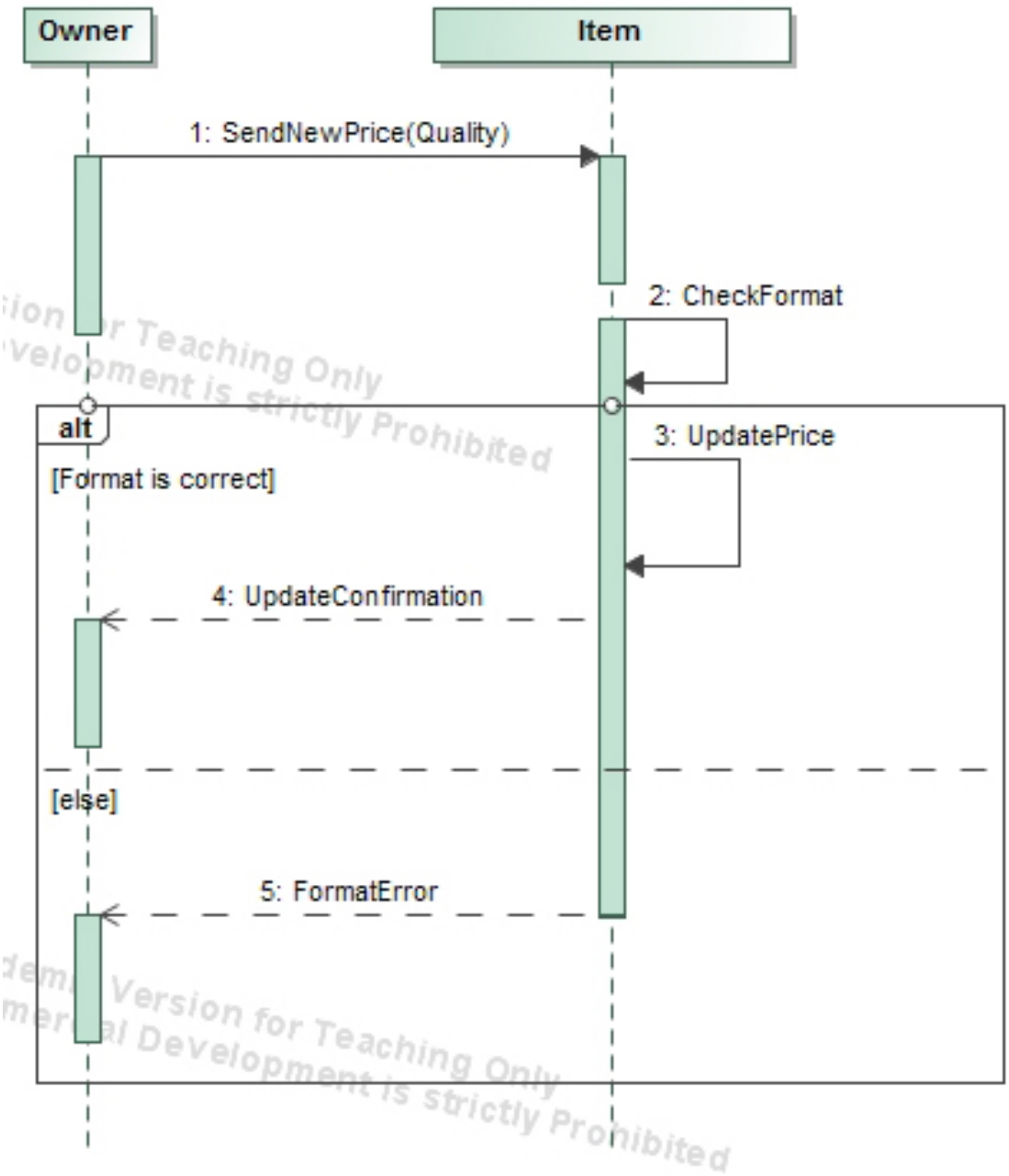

Figure 6: Sequence diagram illustrating interaction between the Owner and a menu item when updating Item price

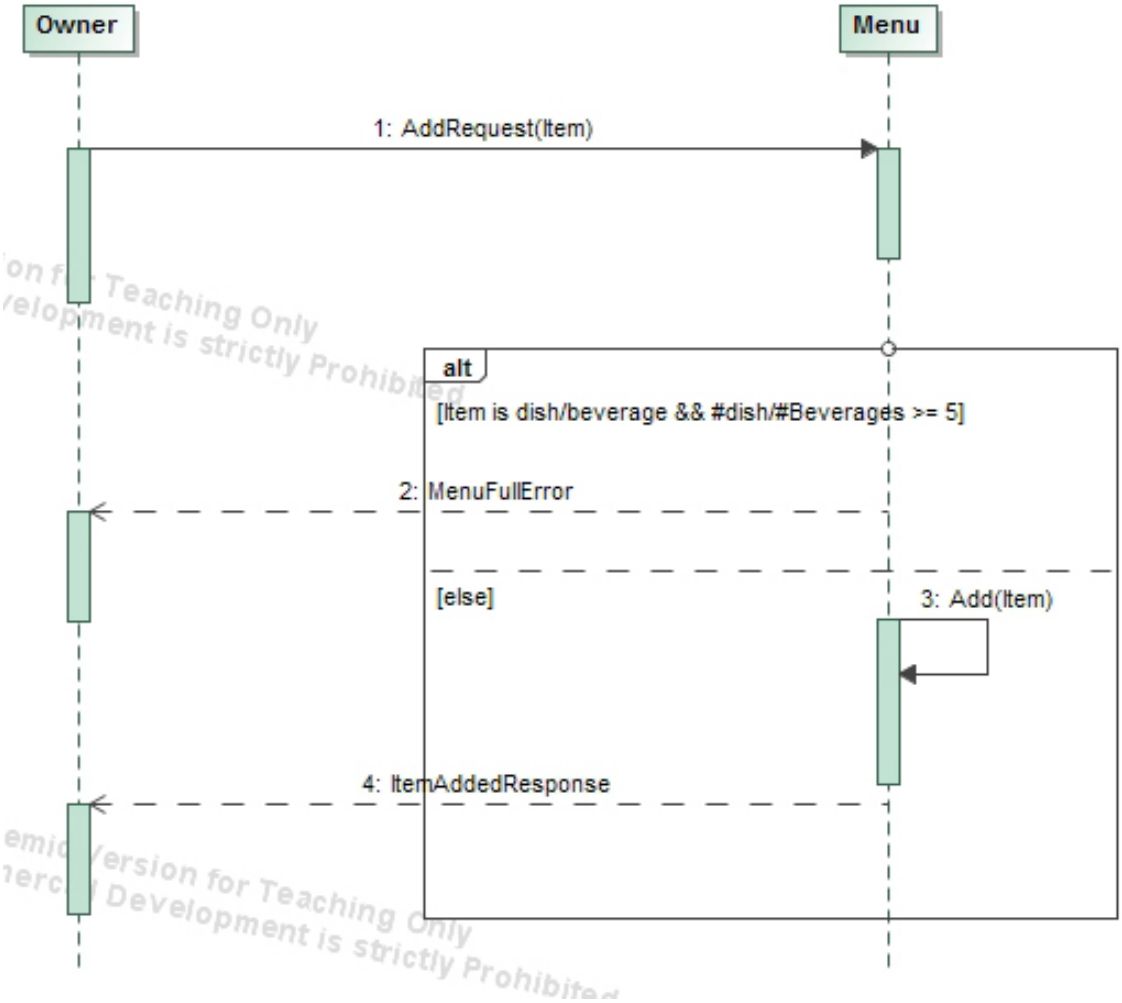

Figure 7: Sequence diagram illustrating the process of choosing dishes and beverages to the Menu

4 Application class model

An application class model is used to define more broader overview of the system defering the details. The model consist of for parts:

- Specific user interfaces

- Boundary classes
- Controllers
- Operations.

The interfaces are used by the Owner (i.e. player) to interact coherently with the game system. Boundary classes are "things" that are between the player and the system more physically (e.g. the mouse, keyboard, and screen). The controllers obviously control the flow of the game, they are what actually makes the game logic and operations work. Last, specific tasks are described by the operations. In the Figure 8 these for partitions of the application class model are shown.

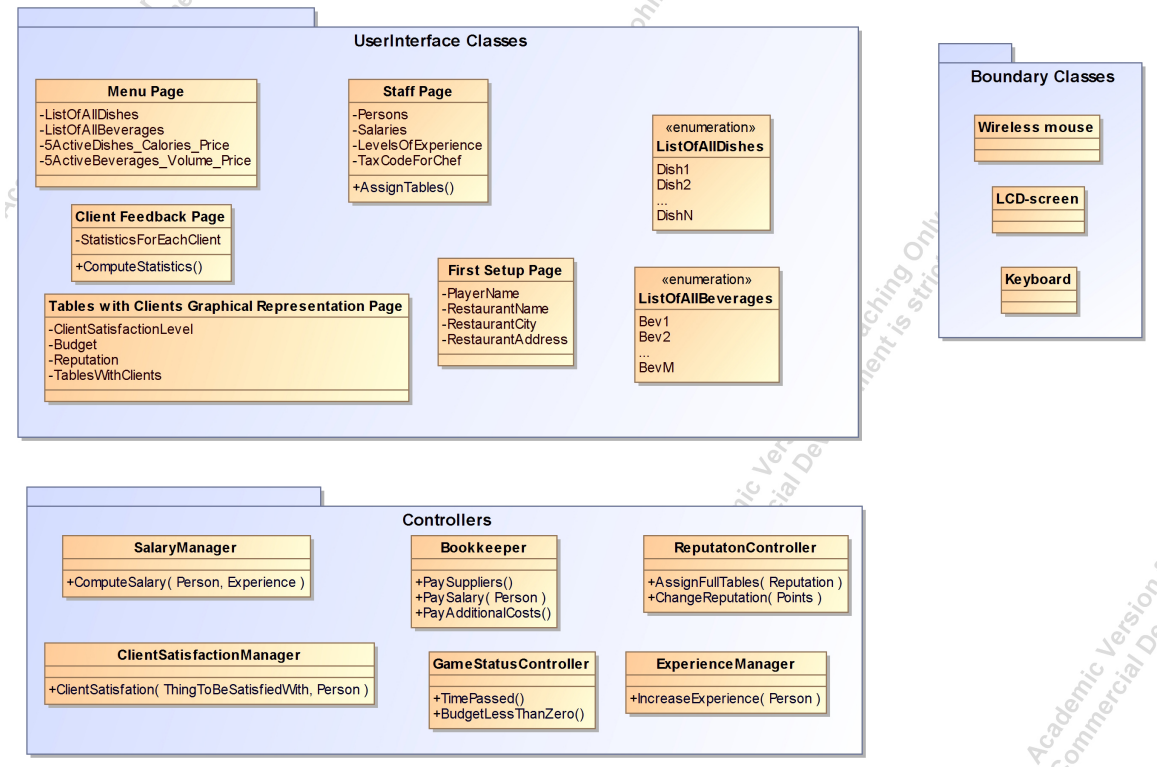

Figure 8:

To begin with, when the game starts, the player is promted with the First Setup Page where the player's name, restaurant name, city, and address are entered. The

player can then move on to the Staff Page to manage personnel of the restaurant and assign tables to the waiters. Next, throughout the game, the player can change the contents of the menu by picking items from the list of all dishes and beverages. The main game area is constantly available a view showing tables with clients (with satisfaction levels), the budget and reputation of the restaurant, and navigation menu to other sections of the user interface. Last, when the game ends, the user sees general statistics about each client's behaviour (consumed calories, number of dishes, etc.).

The controllers represent the functional units of the game system. Salary Manager is used to compute salaries accordingly to the experience level of each person (chef, barman, and 3 waiters). Bookkeepr is a controller that manages the budget: Pays salaries, sends money for ingredient suppliers, and manages additional costs (rent, cleaning service, taxes, etc.). The status of the budget and how long the game has lasted is monitored by the Game Status Controller. If the time period exceeds 30 days or the budget goes empty, the game is over. The Owner of the restaurant (player) can order trainings for the personnel. In these cases, the Experience Manager is used to increase the level of experience of the restorant staff. The clients of the restaurant are really the marker of reputation of the restaurant. The Client Satisfaction Manager computes all the changes in the clients' satisfaction levels according to the exprience of the staff serving the exact clients. The reputation of the restaurant implies how many tables are full of clients and changes when accordingly how satisfied the clients are - this is all done by the Reputation Controller.