

Software Engineering

Formal Specifications

3 Ba INF 2018-2019

Stephen Pauwels

13/11/2018

1 Practical

- Deadline: **November 26, 23u55**
- Submission: **Both your report and Yakindu model files**
- Assignment to be made **individually**
- This assignment will be graded on the quality of the presented work and its correctness

2 Context

In order to get a more detailed design of our software system Formal Specifications can be used. We can use them to correctly specify what the (unambiguous) behavior has to be rather than how the exact behavior must be achieved. Since these techniques are formal, we are also able to use the resulting models to check for possible inconsistencies in our design.

3 Assignment

3.1 Z

1. Model the next entities in Z, read the descriptions carefully and choose appropriate data structures to store the information in. Do **not** use quantifiers in your model.
 - Item: Every item has a name and a unique ID which can be used to uniquely describe the item. An item also has a price (positive float) and a category.

- Cart: A cart keeps track of the item IDs of items present in the order together with their quantities.
 - Order: An order keeps track of the item IDs of items present in the order together with their quantities and the client. Also, the state of the order is kept (Open, Payed and Delivered). It also keeps track of the date the order has been placed.
 - Client: A client keeps track of the address information, the order history of the client.
 - System: Keeps a list of all items, orders and clients. A cart is supposed to be kept at the client and thus is not part of System.
2. Make sure you can perform the following queries:
 - Get the ID from an Item given the name.
 - Get the price of an Item given the ID of an Item.
 - Get the price of an Item given the name of an Item.
 - Get the total cost of an Order.
 - Get all Orders who are not payed yet.
 - Get the total income from al payed Orders who have been placed between two dates.
 3. Ensure you can perform the following actions:
 - Add an Item to the System.
 - Add an Item (+ quantity) to a Cart, given the item's ID.
 - Transform a Cart to an Order
 - Change the Order's status from Open to Payed and from Payed to Delivered. Use a single schema to represent both transitions.
 - Remove a certain amount of an Item from a Cart.
 4. Convert your schemas to a basic UML diagram and indicate the different invariants, pre-and postconditions (cfr. slide 29).
 5. Explain your choices in your report, also indicate how you used your schemes to create UML diagrams.

3.2 Statecharts

Model the following behavior using the Yakindu tool, incorporate all the mentioned assumptions in your solution:

1. Model the following pages and possible transitions between pages:
 - Home: standard page that shows when a user enters the site

- Catalog: the Catalog containing all the available Items (you do not have to model the filtering of Items)
 - Item: a generic page showing an Item.
 - Cart: an overview page of the items currently in your Cart
 - Order: the page that transform your Cart into an Order. This page exists out of the following sub-pages:
 - Cart-overview: give an overview of what is in the Cart is is about the be ordered
 - Information: asks the basic information needed from a Client
 - Payment: lets the user choose how to pay for the Order
 - Confirmation: confirm that the order has been placed and payment has been received
 - Log-in: page that allows a user to enter his/her username and password
 - Profile: page containing all information about a registered user
 - Admin: page that has all the administrative tools
2. A user can be logged in or not. When a user is not logged in she/he can choose to go to the Log-in page at any time. After a successful login attempt she/he gets redirected back to the previous page she/he visited. If a user is not showing any activity for 5 minutes, she/he automatically gets logged out.
 3. There exist two different kind of registered users:
 - Only a registered user can visit the Profile page
 - Only an administrator can visit the Admin page
 4. Add Operations that show the functionality of each page.
 5. Make sure no inconsistent states can occur.
 6. Use the appropriate statechart constructions to model the different aspects of the system.
 7. Use your model to determine different test cases in order to test all transitions present in your model. Clearly indicate in your report why a particular test case is important to add and to what part of your model it corresponds (cfr. slide 39).
 8. Explain the different design choices you made in your report.