# EECS 4312 F14 — Project Phase1(a)

Firstname1 Lastname1 (cse99999cse.yorku.ca)
Firstname2 Lastname2 (cse99998cse.yorku.ca)
Firstname3 Lastname3 (cse99997cse.yorku.ca)

September 12, 2014

## Electronic Submission

Submit only one printed report in the Drop Box, and submit only one electronic copy. Electronic submitted by Prism account:

cse99999

| Item | Maximum Raw Mark Score | Raw Mark Score |
|---|---|---|
| Deliverable 1 (Elicitation) | 30 | |
| Deliverable 2 (Monitored Events) | 30 | |
| Deliverable 3 (Controlled Variables) | 40 | |
| Deliverable 4 (3 Use Cases) | 60 | |
| **Total Raw Marks** | **160** | |

## Revisions

| Date | Revision | Description |
|---|---|---|
| 9 September 2014 | 1.0 | Phase 1a description of Project released. |
| 23 September 2014 | | |

# Contents

**1 The Problem** **3**

**2 What you must do** **4**

**3 Elicitation Questions and Answers** **5**

**4 Grammar specification of input commands** **6**

**5 Abstract state** **7**

**6 Three important use cases** **8**
6.1 Use Case 1: Normal scenario ??? . . . . . 8
6.2 Use Case 2: ??? . . . . . 10
6.3 Use Case 3: ??? . . . . . 11

# List of Figures

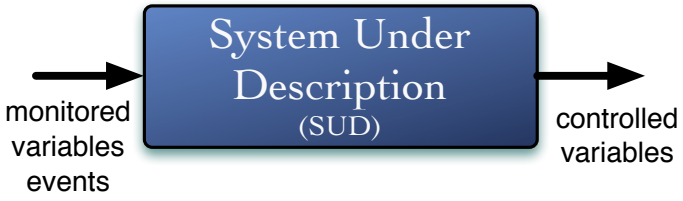
1 **invoice-definitions.txt** . . . . . 6

# List of Tables

# 1 The Problem

The problem: Our customer provided us with the following statement of their needs.

> The subject is to invoice orders. To invoice is to change the state of an order (to change it from the state pending to invoiced). On an order, we have one and one only reference to an ordered product of a certain quantity. The quantity can be different to other orders. The same reference can be ordered on several different orders. The state of the order will be changed into invoiced if the ordered quantity is either less or equal to the quantity which is in stock according to the reference of the ordered product. You have to take into account new orders, cancellations of orders, and entries of quantities in the stock.




A requirement is a separately *verifiable* contractual statement stating a need of the customer. A precise requirements document describes everything necessary to produce a safe and correct system—one that fulfills the needs of the customer—nothing more. At the same time the specification must not over-constrain developers by venturing into design and implementation detail.

Carefully read the full problem write up at: `https://wiki.eecs.yorku.ca/course_archive/2014-15/F/4312/protected:assignments:project:`

# 2 What you must do

- Carefully read the instructions for the the Project and the details for Phase 1(a) of the project.[1]
- The deliverables are:
  1. (D1) Ask and answer further questions to elicit the requirements and to clarify for yourself the real customer needs. Provide the three most important questions and their answers. Ask questions from the customer (instructor/-TAs) on the forum.
  2. (D2) Complete the invoice-definitions.txt, i.e. the grammar that specifies all possible inputs of monitored events
  3. (D3) Choose the abstract state (the controlled variables)
  4. (D4) Provide at least three complete use cases (acceptance tests) that describe the output in terms of the input (i.e. relate the controlled variables to the the monitored events). You may provide more use cases, but we will be grading the first three. So make sure that the first three use cases are a good sample of possible inputs and capture success cases as well as error cases.

Provide the above information structured as show below.

[1] `https://wiki.eecs.yorku.ca/course_archive/2014-15/F/4312/protected:assignments:project:start`

# 3 Elicitation Questions and Answers

If our customer's needs are vague, we need to talk with them to elicit their requirements with precision and clarity. Here is an example of questions we might ask in order to clarify the role of the System Under Development (SUD):

**Question 1**: Is a reference in your your statement of needs a product (e.g. "nuts" or "bolts")?

**Answer**: Yes. Or more precisely a product ID. You may assume that the product ID is entered by the user as a string, e.g "nuts" The types of the products that customers may request must be entered into the system separately (e.g. before a customer can order 100 "bolts", the type "bolts" must already have been registered as a legitimate product type.

**Question 2**: Is it really required (as it would seem from the statement of needs) that an order be limited to a single item, e.g. 100 bolts. Might the user not want the order to contain mutiple items,e.g. 50 "nuts" and 250 "bolts"?

**Answer**: You are right. The statement of needs did seem to convey an incorrect impression that would be over-restrictive. Indeed, a user may wish to order several items in a single order.

**Question 3**: When an order of product items and their quantities is requested, must the user provide the order ID, or does the system (SUD) generate the order ID?

Answer: The system should generate order IDs. [Note: in order to test that the system generates correct outputs, the SUD must generate order IDs in sequence starting at $1, 2, 3n$, where $n$ is some positive integer depending on memory available. If an order is later cancelled, then the order ID is made available for re-use on a first-in/first-out basis. For the purposes of this project, $n = 10,000$]

> You must now complete this process by providing, below, the three most important questions and their answers that remain to be asked. You may ask and discuss this part in the forum.

# 4 Grammar specification of input commands

Basing yourself on your elicitation questions and answers, provide the grammar of the input commands (i.e. monitored events) of the SUD. Provide comments to justify your choice of commands

```
======================
invoice-definitions.txt
======================
system invoice

nothing
  -- sometimes nothing happens at the input

add_type(product_id: STRING)
  -- e.g. add_type("nuts")
  -- product types must be declared in advance of orders

add_product(a_product: STRING; quantity: INTEGER)
  -- e.g. add_product("nuts", 1000)
  -- adds 1000 nuts to stock
...
???
```

Figure 1: **invoice-definitions.txt**

# 5 Abstract state

You must complete the table below so as to describe the outputs (controlled variables) of the SUD. Provide a mathematical type, and the meaning of each variable.[2]

| **Variable** | **Type** | **Details** |
|---|---|---|
| report | STRING | report ok, otherwise report error |
| id | $1 \cdots n$ | Order ID |
| products | | |
| stock | | |
| orders | | |
| carts | | |

[2]Use `http://www.tablesgenerator.com` to produce Latex tables

# 6 Three important use cases

You must provide three important use cases. We got you started on the first one.

## 6.1 Use Case 1: Normal scenario ???

**Description**: This use case is the normal case where ....

```
========================================
Use Case 1: uc1.txt: adding product types
========================================
  report:      ok
  id:          0
  products:
  stock:
  orders:
  carts:
  order_state:
->add_type("nuts")
  report:      ok
  id:          0
  products:    nuts
  stock:
  orders:
  carts:
  order_state:
->nothing
  report:      ok
  id:          0
  products:    nuts
  stock:
  orders:
  carts:
  order_state:
->add_type("bolts")
  report:      ok
  id:          0
  products:    bolts,nuts
  stock:
  orders:
  carts:
  order_state:
```

```
->add_type("hammers")
  report:      ok
  id:          0
  products:    bolts,hammers,nuts
  stock:
  orders:
  carts:
  order_state:
->add_type("nuts")
  report:      product type already in database
  id:          0
  products:    bolts,hammers,nuts
  stock:
  orders:
  carts:
  order_state:
->add_product("bolts",100)
  report:      ok
  id:          0
  products:    bolts,hammers,nuts
  stock:       bolts->100
  orders:
  carts:
  order_state:
-> ...
```

## 6.2 Use Case 2: ???

## 6.3 Use Case 3: ???