COMP201: Software Engineering I

Assignment 1.1 (2024/2025)

(100% mark for Assignment 1.1 is 15% of COMP201 grade)

Deadline for Assignment 1.1: 14th of November 2024, 17:00

OBJECTIVE

This assignment is mainly about "Requirements Engineering" and will consist of various stages to produce parts of a requirements document for a given scenario based on a "proposed drink machine system" detailed on page 2.

Assignment number	1 of 2
Weighting	15 %
Assignment Circulated date provided to class	30/9/2024
Deadline Day & Date & Time	14th of November 2024 at 17:00 (5 PM)
Submission Mode	Electronic submission on Canvas (pdf)

Learning outcome assessed	1. Realise the problems in designing and building significant computer systems 2. Understand the need to design systems that fully meet the requirements of the intended users 3. Be able to apply these principles in practice
Submission necessary in order to satisfy Module requirements	No
Purpose of assessment **Marking criteria**	To assess the students’ ability to analyse, generate and document user requirements See end of document
Late Submission Penalty	Standard UoL Policy

Instructions

- All tasks refer to the scenario outlined on page 4 so, before you begin, **read the scenario carefully**.
- You may **make some reasonable assumptions** about how the system should work (without inventing new functionality).
- **There is no "right answer" to modelling a system**, different solutions can be equally good.
- It may be helpful to refer to the course textbooks "Software Engineering", Addison-Wesley, by I. Sommerville and "Using UML", Addison-Wesley, by P. Stevens.

Task 1 (60%)

(20% for use-case diagram, 40% for use-case descriptions)

For this assignment limit your submission to 15 use-cases, and make sure each use-case provides value.
For example, "use the keypad" would not be a good use-case, "accept and validate coin" would be a good use-case.
The markers for this assignment will award marks to the first 15 use-cases.
Also you are expected to have use cases for **all** 3 domains of operation, customer, operator and engineer. You do **NOT** need to use a lot of includes or extension use cases to get good marks.

All tasks for this assignment refer to the given scenario "**proposed drink machine system**" (overleaf on page 2).

Produce a **UML use-case model** (i.e., **both** a use-case diagram **and** use-case descriptions) include use-case descriptions that are within the scope of the system, but for this coursework **ONLY** include human actors. There are **THREE** actors to identify. Your uses cases are modelling interactions with the human users of the system.

For the use-case diagram part of the model, you may use any method to draw it, including a hand-drawn diagram or ArgoUML software (available on the departmental computers (click start and then type ArgoUml into the search box) or download free via the internet). The demonstrators will be able to help you with using this program. There is also app.genmymodel.com which is easy to use and free online (for public projects), so it is convenient if you are not in the lab.

For the model diagram, if you find that using one diagram is not sufficient or becomes overly complex, feel free to produce multiple diagrams. This is encouraged if the diagram has become difficult to read. Keep all text easy to read and all fonts at least 14pt.

Please use the following template for your use case descriptions:

ID	Id if use case, example UC1
Actors	List of relevant Actors
Name	Short name for use case
Description	Description of purpose of use case
Pre-conditions	What must be true to allow use case to happen
Event flow	Line by line detailed events for the use case
Post-condition	Any changes to the systems internal state due to use case executing
Includes	Any use cases which make up this use case
Extensions	Any optional use cases that are part of initial case
Triggers	What might trigger use case

Task 2 (20%)

Identify and list 10 **non-functional verifiable** requirements of the "**proposed drink machine system**" below, using the description of the scenario (you can make some assumptions about the system not detailed in the requirement description).

Each requirement must have an appropriate criterion so it can be verified. So, it needs to be possible to objectively test each requirement in your list.

Task 3 (20%)

Devise a coding scheme for the keypad based on the information on page 8 of this document. To gain maximum marks the coding scheme, should be complete, minimal and provide the correct hamming distance.

Proposed drinks machine

Proposed drinks machine control system
Your company has been commissioned to design a drinks vending machine which will be able to produce a range of hot beverages, for example

Hot chocolate, standard coffee, flat white, cappuccino, tea and soup. For tea and standard coffee only the user can pick with milk and/or sugar.

All of these drinks are made by heating up water, then mixing it up in a mixing vessel, the mixture is then dispensed into a disposable cup, the user also has an option to use their own cup. After each drink is dispensed the mixing vessel is cleaned out by running hot water through it for a short amount of time.

To do all this, the machine needs to be able to:

Accept and validate coins and calculate the current value of the coins inserted

Accept instructions to choose a particular drink recipe

Heat up the water to the correct temperature then mix the drinks

Dispenses and fill the disposable cups for the drinks

The system has the following sensors, 1 cup out of stock sensor, 16 drink ingredients out of stock sensors, cash box full sensor, water temperature sensor, door open sensor.

The machine has a keypad on the front with a keypad as shown in Figure 1, this is used to make the drink selection and to control the engineer’s tests. The machine also has an alphanumeric LED display.

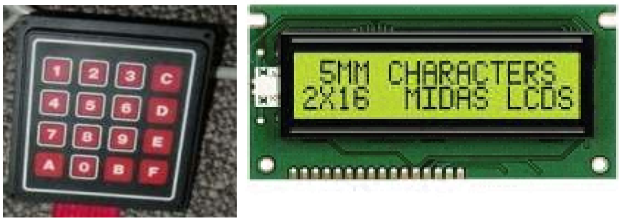

Figure 1 Keypad and alpha display

The drinks machine is fitted with a GPRS/3G/4G/5G WiFi card which allows it to be connected to the internet and be controlled by a service operator. The service operator can find out the current stock and cash levels in the machine as well as be informed via an alert if the machine's stock levels are low or the cash box is full. The operator can also make requests to download accounts information from the machine for a particular accounting period, for example 4/4/2021 to 4/5/2021. Operator access to the machine is protected using a username and password stored within the machine. There is also a master password available which can be used create new accounts in case the machines memory has been wiped due to hardware failure. Also the recipes for the drinks can be downloaded to the machine online.

When the machine requires service (such as needing more coffee or the cash to be collected or has run out of cups), a service engineer visits the machine. The back of the machine is opened, this puts the machine into service mode, the engineer will then collect the cash and does a series of tests by using the keypad. Each engineer has a unique passcode which they need to type into the keypad after opening the service door. If they don't do this within a short time interval the machine goes into alarm mode and sends an alert to the service operator.

When the cash is collected, the engineering must record the amount they collect into the keypad, this data is stored in the machine and can be retrieved by the service operator.

Keyboard coding scheme

Given that the engineer and customer both have to use the keypad, for your solution please devise a coding for at least all the drink codes. For this part of the solution, try to make it so the customers only need to press the **minimal** number of key presses possible to get their drink and also make it impossible to get the wrong drink if they only get **1 key press incorrect** (i.e. 1 key press wrong should give an invalid key press sequence). This type of code would be said to have a hamming distance of 2 key presses. Note the keyboard only has the following characters 0,1,2,3,4,5,6,7,8,9,A,B,C,D, E and F.

Scheme should be presented as follows (note this example is not correct)

Drink	Code
Coffee	1
Tea	2
etc	

Marking Criteria

Part	A++ to A 70%+	B 60-69%	C 50-59	D 40-49	E+ 35-39	E- to G <35
1	Correct notation used throughout, well-chosen set of use cases and all case descriptions present.	Good set of use cases but some descriptions missing or minor case missing or some minor notation problem	Poor set of use cases or significant problems with notation. Level of detail not sufficient for problem	Some critical use cases missing or use case descriptions missing.	Shows some correct requirements analysis of the problem.	No clear evidence that the requirements have been understood at all or no clear attempt at use case diagram or descriptions.
2	All requirements are non-functional and verifiable.	Good answers but confusion between function and non-functional requirements.	Missing one or two non-functional requirement or problem with verifiability.	Missing up to three requirement descriptions.	Only 2 or 3 correct requirements present or incorrectly defined.	Requirements don’t make sense.
3	Coding scheme is complete, minimal and robust to required hamming distance.	Up to 2 codes missing or incorrect	3-4 codes missing or incorrect	5 codes missing or incorrect	6 codes missing or incorrect	Coding scheme is not submitted or mostly incorrect