**Assume the following partially specified system:**

**A system manages information about patients in a hospital. There are two types of users: medical personnel and patients. The system must allow both types of users to update patient information, it must have persistent storage (information is retrieved from session to session), new users must be inserted in the system, each patient has an unique ID, first name and last name, the status of a patient must be either “in-patient” (it is checked in the hospital) or “out-patient” (not checked in the hospital). Each patient in the system has attached a treatment that can be updated at any time by the medical personnel.**

**Make any assumptions that you believe are needed to complete the questions below. Explain your answer**

1. **Write two functional requirements regarding a desired listing functionality that would be included in the category “functional suitability” of standard ISO/IEC 25010:2011**

* It can list the current patients in the system by paging and can be searched by status or first name of those patients.
* It can list the current medical personnel in the system by paging.
* It can list the details of a treatment of a patient by paging.

<https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en> (4.2.1)

Functional requirements should always start with “xxx system should …”

修改:

The system should be able to list the current patients in the system by paging and search by status or first name of those patients.

The system should be able to list the current medical personnel in the system by paging.

The system should be able to list the details of a treatment of a patient by paging.

1. **Write two non-functional requirements related to the performance of the tool that would be included in the category “performance efficiency” of the standard ISO/IEC 25010:2011**

* The response time of retrieving information of current patients from database should be no more than 3 seconds on average, in the high system utilization case, it should be no more than 5 seconds on average.
* The information page response time should be less than a second on average, in the high system utilization case, it should be no more than 2 seconds on average.
* The system framework should be able to handle maximum of 10,000 service requests on average with less than 0.1 second delay per request.

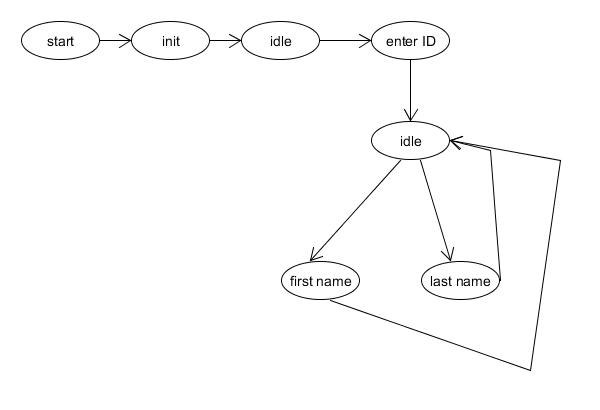
1. **Write two specific test cases you would use to test the system.**

|  |  |  |  |
| --- | --- | --- | --- |
| Title | Patient display test | Test priority | High |
| Test type | black box, system test | | |
| Purpose | Tests for retrieving the patients information | | |
| Test Description  Tests for retrieving the patients information, it lists the current patients in the database, and display their id, first name and last name at each entry of the list table | | | |
| Prerequisite  System starts up and there are some patients in the system | | | |
| Dependencies  A instance of Database is running | | | |
| Input Data  A not empty string of first name | | | |
| Expected Result  A not empty list of patients with the given first name | | | |
| Actual result  A list of patients with the given first name | | | |
| Steps   1. A not empty string of first name 2. Execute the test | | | |
| Output  A not empty list of patients with the given first name | | | |
| Exit criteria  A not empty list of patients with the given first name display on the screen | | | |
| Post-condition  Test case run successfully with no error | | | |

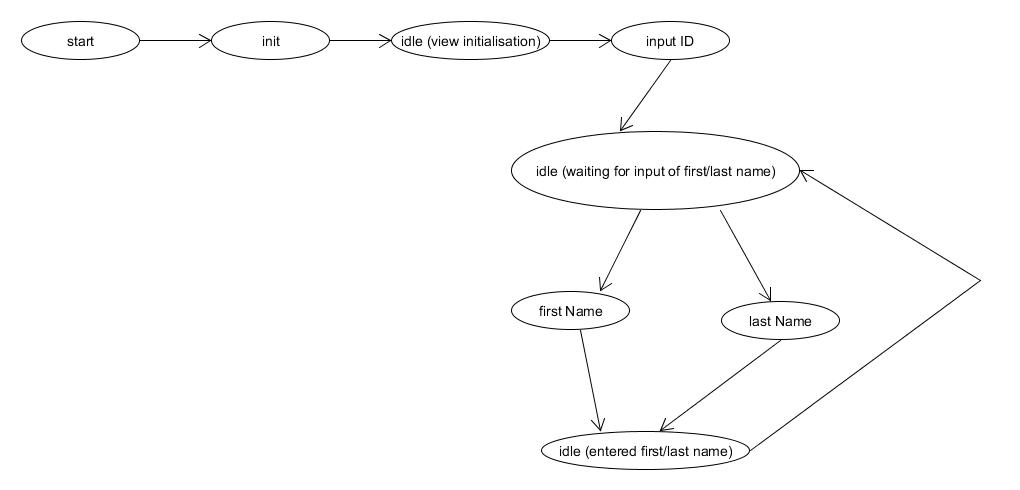
|  |  |  |  |
| --- | --- | --- | --- |
| Title | Patient information insert test | Test priority | High |
| Test type | black box, system test | | |
| Purpose | Tests for inserting the patients information | | |
| Test Description  Tests for inserting the patient’s information, it inserts a new patient information and should be a new entry in the listing page. | | | |
| Prerequisite  System starts up and listing function is working properly | | | |
| Dependencies  A instance of Database is running | | | |
| Input Data  A unique ID, the first name and last name | | | |
| Expected Result  A new patient in the database | | | |
| Actual result  A new patient in the patient listing page | | | |
| Steps   1. Enter an unique ID 2. Enter the first name 3. Enter the last name 4. Check the new patient in the listing page | | | |
| Output  The new patient is displayed in the listing page | | | |
| Exit criteria  The new patient is displayed in the listing page and its data records are in the database | | | |
| Post-condition  Test case run successfully with no error | | | |

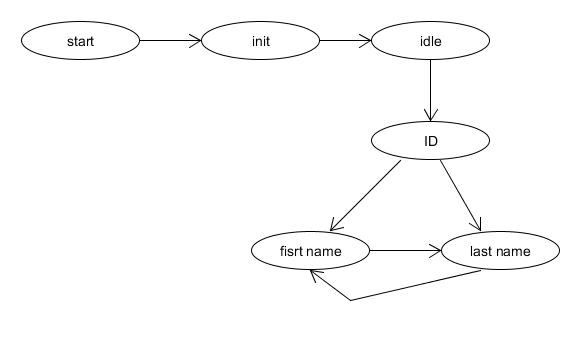
1. **The functionality for entering patient data is to be tested using a FSM. Entering a field prompts a transition in this FSM. The data required is the patient unique ID, the first name and the last name. The last two can be entered i*n any order but only after the unique ID has been entered*. Create a derived FSM and specify the transitions as a transition matrix.**

**Generally correct, but for the 2 idle states, I think you can either merge those to into one, or use “idle (view initialisation)”, “idle (waiting for input of first/last name)”, and “idle (entered first/last name)”.**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Init** | **ID** | **First name** | **Last name** |
| **Init** |  | **Field of ID** |  |  |
| **ID** | **finish** |  | **Field of first name** | **Field of last name** |
| **First name** |  | **finish** |  |  |
| **Last name** |  | **finish** |  |  |

**修改:**

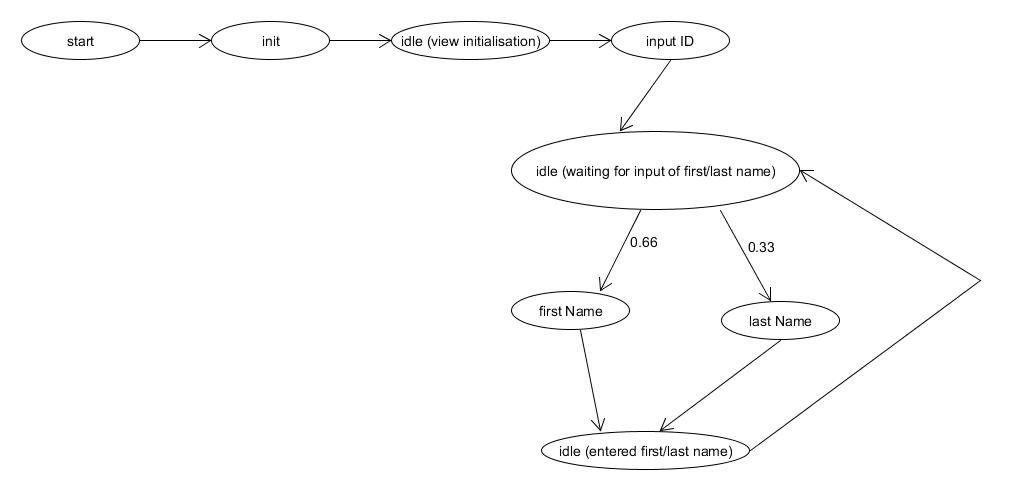
**或者:**

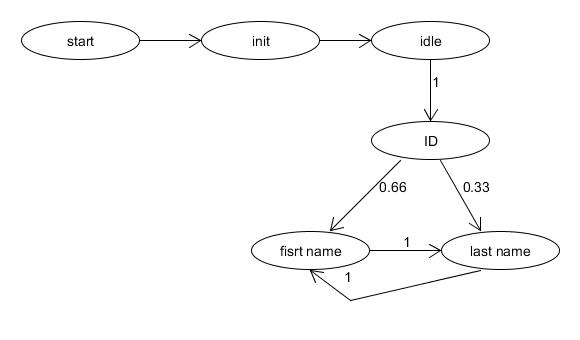
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **next** | **Init** | **ID** | **First name** | **Last name** |
| **current** |
| **Init** |  | **Field of ID** |  |  |
| **ID** |  |  | **Field of first name** | **Field of last name** |
| **First name** |  |  |  | **finish** |
| **Last name** |  |  | **finish** |  |

1. **With respect to the previous FSM, the probability of entering the first name first is twice that of entering the last name. Derive the Markov Chain from the previous FSM.**

Entering ID is a state must transit to before entering either the first name or last name, and it appears once only.

The previous sentence is a hint to this question. Check out the properties of Markov Chain. (From a single node, the sum of the probabilities on each of the out-going path should be 1).

**修改:**

**或者: **