**Software Detailed Design**

**for**

**Pharmacy Store Management System**

Version 1.0 Approved

Prepared by

Thomas Woods (TL)

Mustafa Nafia

Huda Ali

Computer Science and Engineering Department

Indiana University Southeast

Date: November4th, 2020

**Table of contents**

**Table of contents**…………………………………………………..............………….i

1. **Software Design Description**...................................................2

* Overview………………………………….....….............….2

1. **Data Design**..........................................................................2

* Attributesand Relationships.................................................2

1. **Architecture Design**............................................................3

* Characteristics......................................................................3

1. **Interface Design**..................................................................4

* Internal and External Interfaces............................................4

1. **Procedural Design**..............................................................4

* Introduction/Purpose of this Component/Entity..................4
* Input for this Component/Entity..........................................4
* Output for this Component/Entity.......................................4
* Component/Entity Process to Convert Input to Output.......4
* Design constraints and performance requirements of this Component/Entity....4
* Process (pseudo-code algorithm)........................................4

**1.Software Design Description**

**1.1 Overview**

The document demonstrates the attributes of the software and the relationship among each. When each feature of the software performs the task, this will create the functionality of the software which will form the architecture of the program. Therefore, it is important to discuss the interface design to analyze how data is being processed internally and externally.

**2. Data Design**

**2.1 Attributes and Relationships**

Installation/Maintenance module: This process is how the software initially starts and will be developed for maintenance purposes. It starts the installation process and ends with the interaction of the developer with the software to debug and perform updates.

PSMS main control module: The module will give the main screen access for the user to choose which task their wish to perform or what stored data they want to access. It will have input blanks for the info that will be used to form prescription form for current use and store the inputs in the records module.

Access patient records module: From the main control, if the user clicks on this button, it should return the previously visited patients records which include the data related to what has been entered by the user at the initial visit such as: name, age, gender, allergies, contact info, reason for visit.

Prescription form module: This module will retrieve the data related to the patient, which was used for records, with two more modules: one related to the physician info input, the other is medication use info input.

In stock module: The module will give the user the ability to check current in stock and delete add items.

**3.Architecture Design**

**3.1 Characteristics**

Access patient records input data and Prescription form modules will interact with each other to use shared data. Prescription form will access the data which was used for the records with extra information related to the physician and medication to build the prescription form. The following diagram illustrates how the modules will interact with each other to process each task.



**4.Interface Design**

* 1. Internal and External Interfaces

The interface of the program will be a GUI that will have a search bar where users can type in search queries, and then display those results. The user will require a keyboard and monitor for this interface. There will also be a button to add new entries into the various data tables, like new patients, or new prescriptions. After searching for a query, the interface will allow you to click on a displayed entry, and either modify, or delete it, depending on level of access to the system that user is assigned.

**5.Procedural Design**

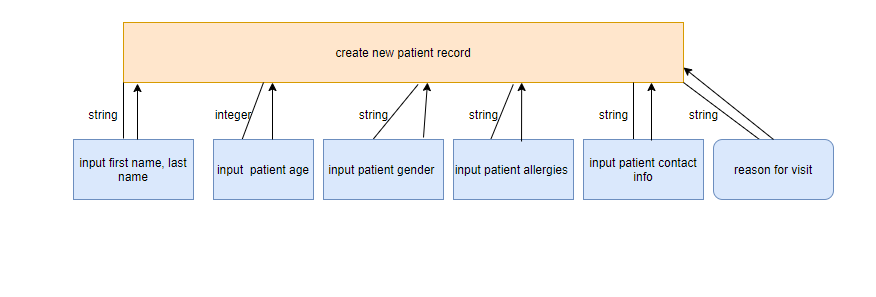
**5.1** Introduction/Purpose of this Component/Entity

* Entity 0 is the patient entity contains name component, age component, gender component, and allergies component. This entity holds all info about the patient.
* Entity 1 is the prescription form entity contains the name of the medication component, reason of taken component, time(s) of day component, physician name component, physician contact info component, and physician signature component. This entity holds all info about the prescription.
* Entity 2 is in stock entity contains current available product component, add item component, and remove item component. This entity holds all info about the current available inventory .in the store

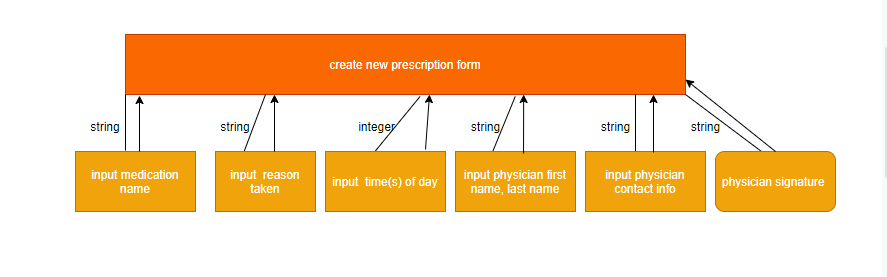


**5.2** Input for this Component/Entity

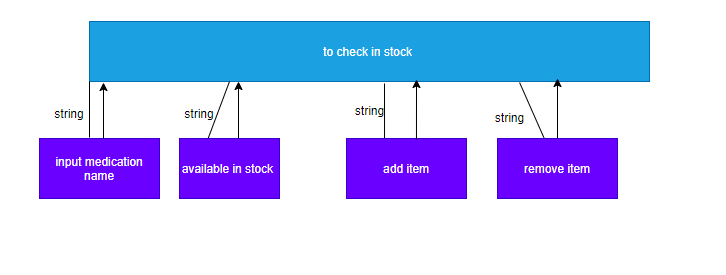
Input for entity 0 (patient)



Input for entity 1 (prescription form)



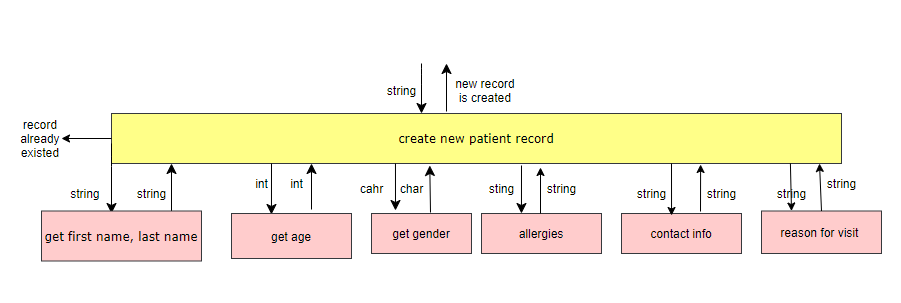
Input for entity 2(in stock)



**5.3** Output for this Component/Entity

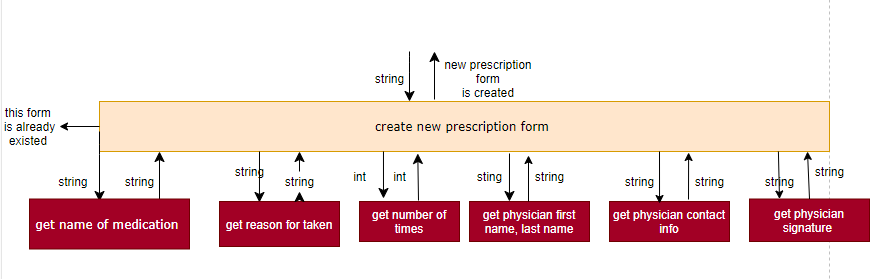
Output for entity 0 (patient)

The output will be the new record if does not already exists.



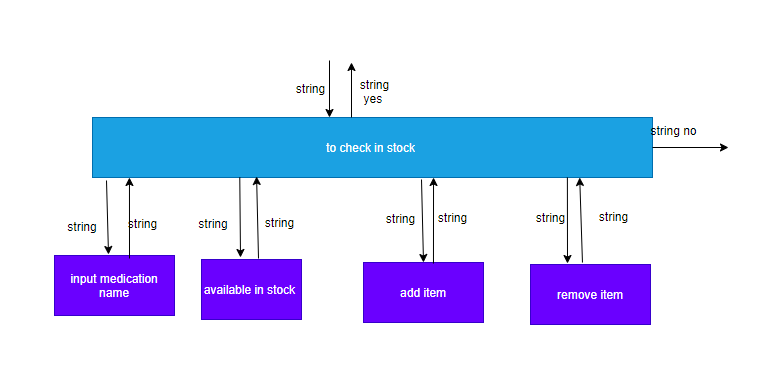
Output for entity 1 (prescription form)

The output will be the new prescription from if it is not already existed.



Output for entity 2 (in stock)

The output will be yes if available or no if it is not available in stock.



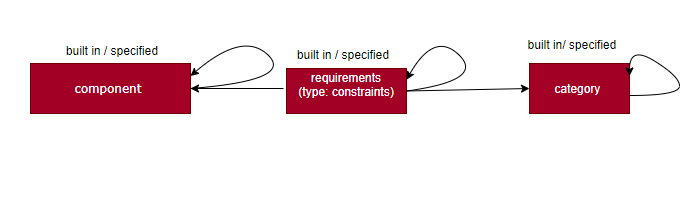
**5.4** Component/Entity Process to Convert Input to Output

For entity 0 (patient), the program will examine the input data, then it will create new patient record for new patients. In addition, the program will save the new record in the system for future use as returning or current patient.

For entity 1 (prescription form), the program will examine the input data, if it is already existed, the program will not save the from. Otherwise, it will create new prescription form for the new patients. Also, the user can update the current existed prescription form for the current patients.

Lastly, for the entity 2 (in stock), the program will examine the input data, and it will output the current available product in the stock. It will return the quantity on hand for each product item. The user will be able to add and remove product items to and from the inventory.

**5.5** Design constraints and performance requirements of this Component/Entity



|  |  |  |  |
| --- | --- | --- | --- |
| Entity modules | Attributes | Relationships | Target modules |
| Category | See 2.1 | Category | Type constraints |
| Components | See 5.1 | Components | Type constraints |
| Requirements  Type(constraints ) | Type constraints | Built in / specified | Type constraints |

**5.6** Process (pseudo-code algorithm)

For the patient module:

Public string getName (string FirstName, string LastName)

/\* it takes string input and return string output \*/

Public int getAge (int age)

/\* it takes an integer input and return integer output\*/

Public char getGender (char gender)

/\* it takes char input and return char output\*/

Public string getAllergies (string allergies)

/\* it takes string input and return string output \*/

Public string getPatientInfo (string info)

/\* it takes string input and return string output\*/

For the prescription from module:

Public string getMedicineName (string name)

/\*it takes string input and return string output \*/

Public string getReasons (string reason)

/\* it takes string input and return string output \*/

Public int getTimes (int time)

/\* it takes int input and return int output \*/

Public string getPhysicianName (string first, string last)

/\* it takes string input and return string output\*/

Public string getPhysicianContactInfo (string info)

/\* it takes string input and return string output\*/

Public string getPhysicianSignature (string signature)

/\* it takes string input and return string output\*/

For the in stock module:

Public Boolean isAvailable (string name)

/\* it takes string input for the medicine name, and it returnstrue if available and false for not available \*/

Public void add (string name, int quantity)

/\* it takes string name for the medicine and int for the quantity it will update the inventory of the store with new product items being added, and returns nothing to the user \*/

Public void remove (string name, int quantity)

/\* it takes string name for the medicine and int for the quantity; it will update the inventory of the store by subtracting the amount was taking from the inventory every time. It returns nothing to the user \*/