EECS4312 eHealth Project

Siraj Rauff (cse23188@cse.yorku.ca)
Skyler Layne (cse23170@cse.yorku.ca)

December 7, 2015

You may work on your own or in a team of no more than two students. **Submit only one document under one Prism account.**

Prism account used for submission: cse23188

Keep track of your revisions in the table below.

Revisions

Date	Revision	Description
date please	1.0	Initial requirements document

Requirements Document:
for Patient care eHealth System

Contents

List of Figures

List of Tables

1. System Overview

The System Under Development (SUD) is a computer system to create and manage health prescription records for Ontario.

This requirements document is specifically for prescription management. The purpose of the eHealth Patient care System is to maintain physicians, medications, patients, and patient prescriptions. The system will also control the undesirable interactions between medications, that is when two medications conflict in some way with one another. Only specialist physicians should be allowed to prescribe undesirable interactions while general physicians should be allowed to prescribe medications, as long as they do not create undesirable interactions.

2. Context Diagram

The System Under Description (SUD) is a computer *controller* to keep track of the physicians, patients, patient prescriptions, medications, and medication interactions. The monitored variables and controlled variables for this computer system can be found in Table 1 and Table 3 respectively.

The system must keep track of abstract state which isn't available to the user. For a list of the abstract states within the controller see Figure 2.

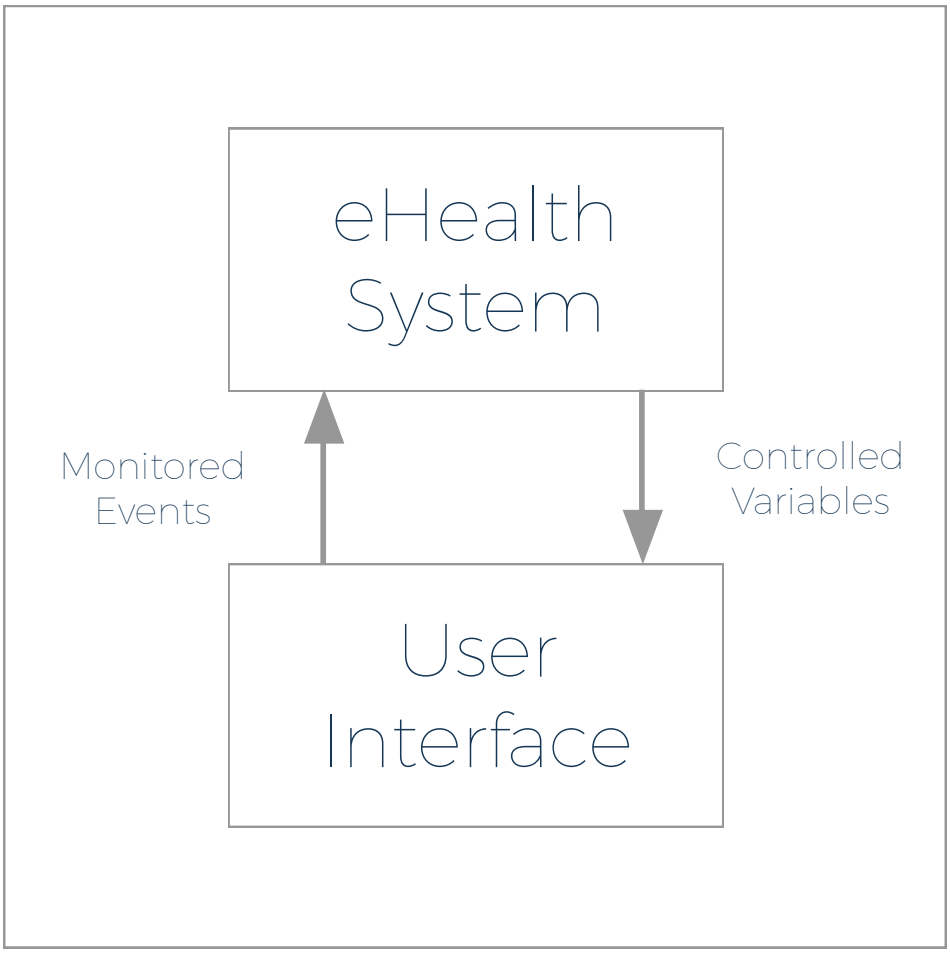

Figure 1: Context Diagram

3. Goals

The high-level goals (G) of the system are:

- G1— The user should be able to add doctors (generalists, and specialists)
- G2— The user should be able to add patients
- G3— The user should be able to add medications
- G4— The user should be able to add perscriptions
- G5— The user should be able to add interactions between medications
- G6— The user should only be able to add a prescription with a dangerous interaction if the doctor is a specialist

4. Monitored Events

The monitored events are those which come through the user interface. The following monitored events will be available to the user.

Name	Interpretation
add_physician(id: ID_MD; name: NAME; kind: PHYSICIAN_TYPE)	Add a Physician to the system
add_patient(id: ID_PT; name: NAME)	Add Patient to the system
add_medication(id: ID_MN; medicine: MEDICATION)	Add a medication to the system,
add_interaction(id1:ID_MN;id2:ID_MN)	Add an interaction between two medications
new_prescription(id: ID_RX; doctor: ID_MD; patient: ID_PT)	Add a new prescription to the system
add_medicine(id: ID_RX; medicine:ID_MN; dose: VALUE)	Add a medicine to a prescription
remove_medicine(id: ID_RX; medicine:ID_MN)	Remove a medication from a prescription
prescriptions_q(medication_id: ID_MN)	Get all the prescriptions with that medication
dpr_q	Get all the dangerous interactions prescribed

Table 1: Monitored Events

Name	Type	Interpretation
PHYSICIAN_TYPE	{gn, sp}	Constrained to either generalist or specialist
MEDICATION	[name: NAME; kind: KIND; low: VALUE; hi: VALUE]	Constrained by GUI
KIND	{pill, liquid}	Constrained to either pill or a liquid
DOSE	{mg, cc}	Constrained to either milligrams or cubic centimetres

Table 2: Monitored Types

5. Controlled Variables

The controlled variables represent what will be shown to the user.

Name	Interpretation	Abstract State
Physicians	A list of all the Physicians currently within the system	See table 2
Patients	A list of all the Patients currently in the system	See table 2
Medications	A list of all the Medications currently within the system	See table 2
Interactions	A list of all the Interactions currently within the system	See table 2
Prescriptions	A list of all the Prescriptions within the system	See table 2
Error	A message displaying the highest priority error, or ok	See table 2

Table 3: Controlled Variables

6. E/R-descriptions

6.1. Requirements Descriptions

REQ1	The system will keep track of Physicians, Patients, Medications, Interactions, and Prescriptions	See Table 3 for list of abstract states.

REQ2	A generalist cannot add a medicine to a prescription if that medicine leads to a dangerous interaction.	See Table 1 for adding medicine, and Table 3 for dangerous interactions.

REQ3	An interaction cannot be added if a specialist has not prescribed at least one of the medications.	See Table 1 for adding interaction, and Table 3.

6.2. Environmental Descriptions

ENV4	All input to the system will be constrained to the GUI grammer.	See Table 1 for the list possible monitored events.

ENV5	A constraint by the grammar will ensure PHYSICIAN_TYPE is either a generalist or a specialist.	See Table 2 for PHYSICIAN_TYPE.

Rationale: The operator interface for the nurse is designed in such a way that the increments of the input temperatures from the control interface must change by whole numbers.

ENV6	The DOSE type will be constrained by the GUI grammar to be either mg, or cc.	See Table 2 for DOSE.

ENV7	The KIND type will be constrained by the GUI grammar to be either a pill or a liquid.	See Table 2 for KIND.

7. Abstract variables needed for the Function Table

Name	Interpredation	Purpose
mnid	The set of all medication ids	Keep track of all the medication ids in the system
ptid	The set of all patient ids	Keep track of all the patient ids in the system
mdid	The set of all doctor ids	Keep track of all the doctor ids in the system
rxid	The set of all prescription ids	Keep track of all the prescription ids in the system
mdpt	Relationship between Doctor and Patient	Keep track of the doctor and the patient
rx	Relation between doctor patient and medication	Keep track of all the medical prescriptions between doctor and patient in the system
prs	List of all prescriptions including dosage	Keep track of all the prescriptions in the system
di	Set of dangerous interactions between medications	Keep track of all the dangerous interactions in the system
gs	The kind of doctor	Keep track of all the type of doctor
dpi	The dangerous prescription report	Notify dangerous interactions, if they exist

Figure 2: Abstract Variables used in Function Tables

8. Function Tables

Name	Meaning
err1	physician id must be a positive integer
err2	physician id already in use
err3	name must start with a letter
err4	patient id must be a positive integer
err5	patient id already in use
err6	name must start with a letter
err7	medication id must be a positive integer
err8	medication id already in use
err9	medication name must start with a letter
err10	medication name already in use
err11	require 0 <low-dose <= hi-dose
err12	medication ids must be positive integers
err13	medication ids must be different
err14	medications with these ids must be registered
err15	interaction already exists
err16	first remove conflicting medicine prescribed by generalist
err17	prescription id must be a positive integer
err18	prescription id already in use
err19	physician with this id not registered
err20	patient with this id not registered
err21	prescription already exists for this physician and patient
err22	prescription with this id does not exist
err23	medication id must be registered
err24	medication is already prescribed
err25	specialist is required to add a dangerous interaction
err26	dose is outside allowed range
err27	medication is not in the prescription

Figure 3: Table of errors and warnings

8.1. Function Table for eHealth

Monitored Inputs		
i = 0		See Table 5
i >0	new_prescription(id, doctor, patient)	See Table 6
	add_medicine(id, medicine, dose)	See Table 7
	remove_medicine(id, medicine)	See Table 8
	add_interaction(id1, id2)	See Table 9

Figure 4: Function Table for eHealth

8.2. Function Table for init

error: $false$

Abstract State	¬ error	error
mind(i)	$\varnothing$	NC
ptid(i)	$\varnothing$	NC
mdid(i)	$\varnothing$	NC
rxid(i)	$\varnothing$	NC
mn(i)	$\varnothing$	NC
mns(i)	ε	NC
mdpt(i)	$\varnothing$	NC
rx(i)	ε	NC
prs(i)	ε	NC
di(i)	$\varnothing$	NC
gs(i)	$\varnothing \mapsto \varepsilon$	NC
dpr(i)	$\varnothing \mapsto \varepsilon$	NC
r(i)	ok	error

Figure 5: Function Table for init

8.3. Function Table for new_prescription(id, md, pt)

error: $\neg(id > 0 \wedge \neg rxid_1(id) \wedge md > 0 \wedge mdid_1(md) \wedge pt > 0 \wedge ptid_1(pt) \wedge mdpt_1(md, pt))$

Abstract State	¬ error	error
mind(i)	NC	NC
ptid(i)	NC	NC
mdid(i)	NC	NC
rxid(i)	$rxid_{-1} \cup \{id\}$	NC
mn(i)	NC	NC
mns(i)	NC	NC
mdpt(i)	NC	NC
rx(i)	$rx_{-1} \upharpoonright (id \mapsto (md, pt))$	NC
prs(i)	$prs_{-1} \upharpoonright (id \mapsto empty\_prs(mnid_1))$	NC
di(i)	NC	NC
gs(i)	NC	NC
dpr(i)	$dpr_{-1} \upharpoonright (id \mapsto \varnothing)$	NC
r(i)	ok	error

Figure 6: Function Table for new_prescription(id, doctor, patient)

8.4. Function Table for add_medicine(id, m, d)

error: $\neg(id > 0 \wedge \neg rxid_1(id) \wedge m > 0 \wedge mnid_1(m) \wedge pt > 0 has(m, prs_1(id)) \wedge sumthin \wedge isValidDose(m, d))$

Abstract State	¬ error	error
mind(i)	NC	NC
ptid(i)	NC	NC
mdid(i)	NC	NC
rxid(i)	NC	NC
mn(i)	NC	NC
mns(i)	NC	NC
mdpt(i)	NC	NC
rx(i)	NC	NC
prs(i)	$prs_{-1}(id) \upharpoonright (m \mapsto d))$	NC
di(i)	NC	NC
gs(i)	NC	NC
dpr(i)	NC	NC
r(i)	ok	error

Figure 7: Function Table for add_medicine(id, medicine, dose)

8.5. Function Table for remove_medicine(id, med)

error: $\neg(id > 0 \wedge \neg rxid_1(id) \wedge med > 0 \wedge mnid_1(med) \wedge prs_1(id)(med)`1 > 0)$

Abstract State	¬ error	error
mind(i)	NC	NC
ptid(i)	NC	NC
mdid(i)	NC	NC
rxid(i)	NC	NC
mn(i)	NC	NC
mns(i)	NC	NC
mdpt(i)	NC	NC
rx(i)	NC	NC
prs(i)	$prs_{-1} \upharpoonright (id \mapsto \varepsilon)$	NC
di(i)	NC	NC
gs(i)	NC	NC
dpr(i)	NC	NC
r(i)	ok	error

Figure 8: Function Table for remove_medicine(id, medicine)

8.6. Function Table for add_interaction(id1, id2)

error: $\neg(id1 > 0 \wedge id2 > 0 \wedge \neg id1 = id2 \wedge mnid_1(id1) \wedge mnid_1(id2) \wedge (\exists a : (prs_1(a)(id1)`1 > 0 \wedge sumin) \vee (prs_1(id)(med)`1 > 0 \wedge sumin)))$

Abstract State	¬ error	error
mind(i)	NC	NC
ptid(i)	NC	NC
mdid(i)	NC	NC
rxid(i)	NC	NC
mn(i)	NC	NC
mns(i)	NC	NC
mdpt(i)	NC	NC
rx(i)	NC	NC
prs(i)	NC	NC
di(i)	$di_{-1} \cup (id1, id2) \wedge di_1 \cup (id2, id1)$	NC
gs(i)	NC	NC
dpr(i)	NC	NC
r(i)	ok	error

Figure 9: Function Table for add_interaction(id1, id2)

9. Validation

todo

```
***
*** top (23:34:28 11/15/2015)
*** Generated by proveit - ProofLite-6.0.9 (3/14/14)
*** Trusted Oracles
***   MetiTarski: MetiTarski Theorem Prover via PVS proof rule metit
***
 Proof summary for theory top
    Theory totals: 0 formulas, 0 attempted, 0 succeeded (0.00 s)

 Proof summary for theory Time
    r2d_TCC1.............................proved - complete   [shostak](0.23 s)
    d2r_TCC1.............................proved - complete   [shostak](0.03 s)
    held_for_TCC1........................proved - complete   [shostak](0.08 s)
    Theory totals: 3 formulas, 3 attempted, 3 succeeded (0.33 s)

 Proof summary for theory isolette
    c_md_ft_TCC1.........................proved - complete   [shostak](0.03 s)
    c_md_ft_TCC2.........................proved - complete   [shostak](0.03 s)
    c_md_ft_TCC3.........................proved - complete   [shostak](0.05 s)
    c_md_ft_TCC4.........................proved - complete   [shostak](0.10 s)
    c_md_ft_TCC5.........................proved - complete   [shostak](0.06 s)
    c_md_ft_TCC6.........................proved - complete   [shostak](0.03 s)
    c_md_ft_TCC7.........................proved - complete   [shostak](0.02 s)
    c_md_ft_TCC8.........................proved - complete   [shostak](0.02 s)
    c_md_ft_TCC9.........................proved - complete   [shostak](0.02 s)
    c_td_ft_TCC1.........................proved - complete   [shostak](0.01 s)
    c_hc_ft_TCC1.........................proved - complete   [shostak](0.07 s)
    c_hc_ft_TCC2.........................proved - complete   [shostak](0.11 s)
    c_hc_ft_TCC3.........................proved - complete   [shostak](0.07 s)
    c_hc_ft_TCC4.........................proved - complete   [shostak](0.05 s)
    c_hc_ft_TCC5.........................proved - complete   [shostak](0.03 s)
    c_al_ft_TCC1.........................proved - complete   [shostak](0.03 s)
    c_al_ft_TCC2.........................proved - complete   [shostak](0.04 s)
    c_al_ft_TCC3.........................proved - complete   [shostak](0.00 s)
    c_al_ft_TCC4.........................proved - complete   [shostak](0.07 s)
    c_al_ft_TCC5.........................proved - complete   [shostak](0.04 s)
    c_al_ft_TCC6.........................proved - complete   [shostak](0.03 s)
    inv_hc_holds.........................proved - complete   [shostak](0.34 s)
    inv_al_holds.........................proved - complete   [shostak](2.71 s)
    Theory totals: 23 formulas, 23 attempted, 23 succeeded (3.98 s)

Grand Totals: 26 proofs, 26 attempted, 26 succeeded (4.32 s)
```

Figure 10: Validated Isolette

You must also provide and prove in PVS one important safety invariant for the heat control *c_hc* and one important safety invariant for the alarm control *c_al*.

Include the PVS sources in the appendix to this document but summarize the proofs here (top.summary).

10. Use Cases

See Section A2 of [**?**] for some use cases. The use cases need to be adapted to the revised descriptions of the previous sections of this document.

11. Acceptance Tests

```
  at1.txt
add_physician      (1, "Mayo", specialist)
add_patient        (3, "Dora")
add_patient        (1, "Drew")
add_medication     (1, ["Wafarin", pill, 1.0, 6.0])
add_medication     (3, ["caffeine", liquid, 1.0, 16.0])
add_medication     (2, ["acetaminophen", liquid, 1.0, 25.5])
add_interaction    (2,3)
add_interaction    (1,2)
new_prescription   (2, 1, 3)
new_prescription   (1, 1, 1)
dpr_q
add_medicine       (1, 1, 5.5)
add_medicine       (1, 2, 5.5)
add_medicine       (1, 3, 5.5)
add_medicine       (2, 2, 5.5)
add_medicine       (2, 3, 5.5)
add_medicine       (2, 1, 5.5)
prescriptions_q(1)
```

Figure 11: First Acceptance Test

```
-- at2.txt
add_physician      (1, "Mayo", specialist)
add_patient        (3, "Drew")
add_patient        (1, "Helen")
add_medication     (1, ["Wafarin", pill, 1.0, 6.0])
add_medication     (3, ["caffeine", liquid, 1.0, 16.0])
add_medication     (2, ["acetaminophen", liquid, 1.0, 25.5])
add_interaction    (1,2)
add_interaction    (1,3)
add_interaction    (2,3)
new_prescription   (2, 1, 3)
new_prescription   (1, 1, 1)
add_medicine       (1, 1, 5.5)
add_medicine       (1, 2, 5.5)
add_medicine       (1, 3, 5.5)
add_medicine       (2, 2, 5.5)
add_medicine       (2, 3, 5.5)
add_medicine       (2, 1, 5.5)
dpr_q
remove_medicine(2,1)
remove_medicine(2,2)
remove_medicine(2,3)
add_medicine       (1, 1, 5.5)
add_medicine       (1, 2, 5.5)
add_medicine       (1, 3, 5.5)
dpr_q
```

Figure 12: Second Acceptance Test

12. Traceability

Matrix to show which acceptance tests passed, and which R-descriptions they checked.

13. Glossary

The definition of important terms is placed in this section. You are not required to complete this.

A. Additional Requirements

REQ8	In *fail* mode, the controller shall only return to *normal* mode if • The sensor is working and • The operator controls are working	In *fail* mode the controller shall return to *normal* mode when *m_st* returns "valid"

REQ9	In any mode, the controller will transition to the *off* mode if the nurse turns the switch off.	The controller will transition to *off* mode from any mode if *m_sw* becomes *off*