

ClinicFlow

Test Report

Maxim Vasiliev #400043983 Susie Yu #000955758
Karl Knopf #001437217 Weilin Hu #001150873
Yunfeng Li #001335650

March 26, 2017

Contents

1	Introduction	1
1.1	Introduction	1
1.2	Definitions and Acronyms	1
2	Unit Testing	1
2.1	Overview	1
2.2	Patient Scheduling	2
2.2.1	Test Inputs	2
2.2.2	Results	2
2.2.3	Test Inputs	3
2.2.4	Results	3
2.3	Healthcare Worker Scheduling	3
2.3.1	Test Inputs	3
2.3.2	Results	4
2.3.3	Test Inputs	4
2.3.4	Results	4
2.4	Modules	4
2.4.1	Test Inputs	5
2.4.2	Results	5
2.4.3	Test Inputs	5
2.4.4	Results	6
2.5	Simulation Engine	6
2.5.1	Test Inputs	6
2.5.2	Results	6
3	System Testing	7
3.1	Login	7
3.2	Insertion and Storage of Data	8
3.3	View, Modify and Delete Data	9
3.4	Schedule Generation	11
3.5	View Schedule and Modify Schedule	12
4	Non-Functional Requirements Testing	14
4.1	Usability	14
4.2	Performance Testing	15
5	Collected Samples	17
5.1	Simulation Results	17
6	Schedule	20

List of Tables

Table #	Title
1	Definitions and Acronyms
2	Patient Schedule Input Unit Tests
4	Patient Schedule Output Unit Tests
6	Health Care Scheduling Input Unit Tests
8	Health Care Scheduling Output Unit Tests
10	Module Input Unit Tests
12	Module Output Unit Tests
14	Simulation Engine Unit Tests
17	Login System Tests
18	Database Storage System Tests
19	View, Modify and Delete Data System Tests
20	Schedule Generation System Tests
21	View Schedule and Modify Schedule System Tests
22	Usability Tests
23	Performance Tests
24	Testing Schedule

1 Introduction

1.1 Introduction

1.2 Definitions and Acronyms

Table 1: Definitions and Acronyms

Acronym	Definition
Provider	Someone who works in the clinic environment (a nurse, doctor, technician)

2 Unit Testing

2.1 Overview

As the programs will be written in the Python 3 programming language, we can use the unittest framework. We will define test cases, with inputs and expected outputs and then be able to run this for each module in the system. We have not yet specified the names of each module and the functions contained, but we do have a general idea of what modules we would want and what they should do. These tests will be refined once specific function

names and information is known.

2.2 Patient Scheduling

We intend to have a module in the program that will take in a file containing patients in the clinic, and produce a schedule for them. First we must test if scheduling system can take appropriate inputs. The initial inputs for this test would be that the program has nothing loaded into it currently.

2.2.1 Test Inputs

Table 2: Unit Tests - Patient Schedule-Input

Test #	Inputs
1	No patient dataset
2	Sample (valid) patient dataset
3	Incorrectly formatted patient dataset

2.2.2 Results

Table 3: Unit Tests - Patient Schedule-Results March 2017

Test #	Results
1	Program Crash, error: can not find patient file
2	Program loads schedule and continues to run
3	Program crash, can not read data from file

We must also check the function that will output the schedule in a useable form for the rest of the system.

2.2.3 Test Inputs

Table 4: Unit Tests - Patient Schedule-Output

Test #	Inputs
4	Sample patient dataset
5	Sample patient dataset with many repeats

2.2.4 Results

Table 5: Unit Tests - Patient Schedule-Output : Results March 2017

Test #	Results
4	Output in usable form
5	Output in usable form

2.3 Healthcare Worker Scheduling

We intend to have a module in the program that will take in a file containing the names and schedule of the providers. It will then generate a schedule object that the simulation can use.

2.3.1 Test Inputs

Table 6: Unit Tests - HealthCare Schedule-Input

Test #	Inputs
6	No health care dataset
7	Sample health care dataset
8	Incorrectly formatted health care dataset

2.3.2 Results

Table 7: Unit Tests - Health Care Schedule-Results March 2017

Test #	Results
6	Program Crash, error: can not find health care file
7	Program loads schedule and continues to run
8	Program crash, can not read data from file

We must also check the function that will output the schedule in a useable form for the rest of the system.

2.3.3 Test Inputs

Table 8: Unit Tests - Health Care Schedule-Output

Test #	Inputs
9	Sample (valid) health care dataset
10	Sample health care dataset with many repeats

2.3.4 Results

Table 9: Unit Tests - HealthCare-Output : Results March 2017

Test #	Results
9	Output in usable form
10	Output in usable form

2.4 Modules

Clinics can have a variety of sections (which we shall call modules) such as Blood taking or xray. Each section can be unique or have multiples. We are going to model each clinic in a data file, which will be read into the system when a simulation is run for that clinic.

2.4.1 Test Inputs

Table 10: Unit Tests - Clinic Modules-Input

Test #	Inputs
11	No module dataset
12	Sample (valid) module dataset
13	Incorrectly formatted module dataset

2.4.2 Results

Table 11: Unit Tests - ClinicModuleInput-Results March 2017

Test #	Results
11	Program Crash, error: can not find clinic file
12	Program loads clinic file and continues to run
13	Program crash, can not read data from file

We must also check that the functions in this module create a valid network of modules for the clinic.

2.4.3 Test Inputs

Table 12: Unit Tests - Clinic Module-Output

Test #	Inputs
14	Sample (valid) clinic module system with no repeats
15	Sample (valid) clinic module system with many repeats

2.4.4 Results

Table 13: Unit Tests - ClinicModule-Output : Results March 2017

Test #	Results
14	Output in usable form
15	Output in usable form

2.5 Simulation Engine

The system relies on the simulation engine to generate any results. This module must be able to take in the previous schedules and data, and use that to run a discrete event simulation using those schedules. The pass criteria for this engine should be that it is capable of generating realistic schedules. To do this, we ran data set through the engine and randomly selected patients to check to see if they followed a logical path thorough the clinic.

2.5.1 Test Inputs

Table 14: Unit Tests - Simulation Engine

Test #	Inputs
16	A dataset that is based on actual clinic data

2.5.2 Results

Table 15: Unit Tests - Simulation Engine : Results March 2017

Test #	Results
16	Clinic Output made logical sense, results were realistic.

Table 16: Unit Tests - Simulation Engine-Output : Results March 2017

Test #	Results		
	Number	Sched App Time	Arrival Time Departure Time
	24	09:00:00 AM	08:05:00 AM 10:12:00 AM
	62	8:00:00 AM	7:55:00 AM 10:07:00 AM
	82	2:00:00 PM	1:40:00 PM 3:10:00 PM
	87	8:45:00 AM	8:30:00 AM 11:35:00 AM
	123	10:15:00 AM	9:50:00 AM 11:40:00 AM
	472	9:30:00 AM	9:20:00 AM 12:15:00 PM
	588	1:30:00 PM	1:10:00 PM 2:10:00 PM

3 System Testing

3.1 Login

The purpose of user login is to ensure that only valid users allowed to in access the system. There are two types of account. One is the administrator account which has full authority to manipulate the data and control all operable functions of the system. Another one is the viewer account which only can view the information. Testing retrieves the input account and password and match with the account information in an existing database to determine whether the user is valid and what the user can do.

Table 17: System Tests: Login

#	Initial State	Inputs	Pass Criteria
17	Login Page Empty input field of account and password.	Empty input of one of the input fields or both. Click login.	Stay at same page and error message of empty input.
18	Login Page Empty input field of account and password.	Valid input of account and password of administrator account. Click login.	Redirect to the main page of the application, and give user full authority of control.
19	Login Page Empty input field of account and password.	Valid input of account and password of viewer account . Click login.	Redirect to the main page of the application, user only allowed to view the data and schedule.

20	Login Page Empty input field of account and password.	Invalid of account or invalid password. Click login.	Stay at the same page and present an error message of the invalid login.
21	Application main page	Click logout.	Redirect to the login page.

3.2 Insertion and Storage of Data

The application should allow administrator account user to insert data such as patient procedure, doctor and nurse shift hours and other necessary data into corresponding databases. The testing will compare the inserted data with the data stored in database. There exists a checking module to validate the input values.

Table 18: System Tests: Data Storage

#	Initial State	Inputs	Pass Criteria
22	Application main page, admin account.	Click Add Data.	Redirect to the Application adding data page.
23	Application adding data page.	Add the inexistent data to target database with correct data type. Click submit.	Stay in same page and all inputs are cleaned. Data appear in the corresponding database.
24	Application adding data page.	Add the data to target database with incorrect data type or incorrect pattern. Click submit.	Stay in same page and save valid data. Error messages indicate the invalid inputs.

25	Application adding data page.	Add the data which already existed in target database. Click submit.	Stay in same page and save data. Error message indicates the redundancy of data.
26	Application adding data page.	Add data which out of domain (such as reservation date in past days) to target database. Click submit.	Stay in same page and save data. Error message indicates that the data out of valid range.
27	Application main page, viewer account.	Click Add Data.	Error message.

3.3 View, Modify and Delete Data

Administrator account user can view, modify and delete existed data in database. Viewer account user only can read the data in the database. According to the selection of user, retrieve corresponding database and display the content on user interface. Changes and deletion created by administrator account user should be stored into database. Testing checks whether the application display right data, and whether the changes synchronize with database. Test the validation of the database when unexpected actions happen.

Table 19: System Tests: Data Manipulation

#	Initial State	Inputs	Pass Criteria
28	Application main page, admin and viewer account	Click View Data.	Redirect to the View Data page.
29	Application view data page, admin and viewer account.	Select target database, click view.	Stay at same page. Display the content of all data from corresponding database.

30	Application view data page, admin and view account.	Select target database and give specific condition, click view.	Stay at same page. Display the content of target data from corresponding database.
31	Application view data page, admin account.	Modify the displayed data and change the value by another valid value. Click save.	Stay at same page. The value in display area and database have been changed.
32	Application view data page, admin account.	Modify the displayed data and change the value by invalid value(empty for required value or out of valid range). Click save.	Stay at same page, no change happen in displayed data or database. Error messages indicate the unexpected changes.
33	Application view data page, admin account.	Delete whole one row of data by clicking deletion button at end of the row. Click save.	Stay at same page. The deleted row disappear and the data in database is deleted as well.
34	Application view data page, admin account.	Delete whole one row of data if the data is expected to use in future (Patient reservation in next few days). Click save.	Stay at same page. Pop up a deletion confirmation window. Confirming the deletion will remove the row from the list. Cancel deletion will save the data.
35	Application view data page, viewer account.	Modify the displayed data.	Stay at same page. Data is not editable.

36	Application view data page, viewer account.	Click delete button.	Stay at same page. Deletion button does not exist.
37	Application view data page, viewer account.	Click save.	Stay at same page. Save button does not exist .

3.4 Schedule Generation

The application retrieves data from databases and generates schedules based on back end algorithms. Generating schedule is a critical part in the application, because the Clinic needs accurate schedules without conflicts to maintain the efficiency and minimize the wasting of time for both parties. Thus the generated schedule should be examined carefully by the users.

Table 20: System Tests: Schedule Generation

#	Initial State	Inputs	Pass Criteria
38	Application main page, admin account	Click generate schedule	Redirect to generate page schedule
39	Application generate schedule page	Select the type of schedule, click start generating.	Create a schedule and display it to user. Saved the schedule into corresponding database.
40	Application generate schedule page	Insert several data of patients reserve same day, same time, different clinic slots. Select the type of schedule, click start generating.	Create a schedule and display it to user. The patients can be issued to same time. Saved the schedule into corresponding database.

41	Application generate schedule page	Insert several data of patients reserve same day, same time, same clinic slots. Select the type of schedule, click start generating.	Create a schedule and display it to user. The patients should be spreaded to different time periods. Saved the schedule into corresponding database.
42	Application main page, view account	Click generate schedule	Stay at same page. Generate schedule does not exist.

3.5 View Schedule and Modify Schedule

The application allows administrator account user and viewer account user to view the generated schedules. Administrator account user also has the permission to adjust the schedule. Testing ensures that the changed schedule doesn't have conflicts.

Table 21: System Tests: View Schedule and Modify Schedule

#	Initial State	Inputs	Pass Criteria
43	Application main page, admin and viewer account.	Click view schedule.	Redirect to the view schedule page.
44	Application view schedule page, admin and viewer account.	Select type and date of the schedule. Click view.	Stay at same page. The page displays the existed schedule.
45	Application view schedule page, admin account.	Click adding button and add a new reserved time into blank area of schedule. Click save.	Stay at same page. The new time period is added into schedule. The new schedule is saved into database.

46	Application view schedule, page admin account.	Click on adding button on a reserved area. Click save	If no error happens , stay at same page. No adding button on a reserved area.
47	Application view schedule page, admin account	Click on moving button on a reserved area and choose an empty area. Click save.	If no error happens stay at same page. The selected time period is set at new area. The new schedule is saved into database.
48	Application view schedule page, admin account	Click on moving button on a reserved area and choose a reserved area. Switch the reservation. Click save.	If no error happens, stay at same page. The selected time period is switched with another one. The new schedule is saved into database.
49	Application view schedule page, admin account.	Application validates the changes of schedule.	Checks on the sequences of procedures in clinic. If adding, moving, and switching violate the the sequence, show error message.

50	Application view schedule page, admin account.	Click delete button to remove a reservation. Click save.	Pop up a window for deletion confirmation. If continue to delete, remove the reservation. Save the new schedule to database.
51	Application view schedule page, viewer account.	Click adding or click moving or click delete or click save.	Stay at same page. No adding button, no moving button, no deletion button, no save button.

4 Non-Functional Requirements Testing

4.1 Usability

Table 22: Usability Tests

#	Description	Type	Criterion	Tester(s)
52	We will list the most frequently performed tasks, and the development team will use the product to complete them. We count the number of mouse clicks.	Functional (dynamic, manual)	The average number of mouse clicks should be less than five.	Development Team
53	We will invite five doctors and five nurses to use our product and give them a three minutes demonstration on how to complete a certain task. After that, the participants will try to complete the same task.	Functional (dynamic, manual)	The participants can complete the same task correctly in three minutes.	Testing Team

54	We will invite five doctors and five nurses to use our product, but we will not give a demonstration on how to complete a certain task. Instead, the participants will try to complete it based on their previous experience and the hints provided by the application.	Functional (dynamic, manual)	The participants can complete the required tasks in five minutes and they should encounter fewer than three errors.	Testing Team
----	---	---------------------------------	---	--------------

4.2 Performance Testing

Table 23: Performance Tests

#	Description	Type	Criterion	Tester(s)
55	The development team will use the product to complete the most frequently performed tasks. We will calculate the time from the start-up of the application to the completion of the work.	Functional (dynamic, manual)	The average time should be less than five minutes.	Development Team
56	The development team will generate 100000 data points, which is ten times more than the expected data points. We will input those data points into our application. Next, we will input 100 times more data points into the application.	Functional (dynamic, manual)	The application does not crash, and there are no obvious latencies (less than 5 seconds for each task) when performing the common tasks in the first case. The application can crash in the second case.	Development Team

57	The development team will generate random data points, which contain problems such as incorrect formatting data and illegal characters. We will input those data points into our application.	Functional (dynamic, manual)	The application does not crash and prompts the users that the input data is not correct and how they could correct it.	Development Team
----	---	------------------------------	--	------------------

5 Collected Samples

Number	Scheduled Appointment Time	Arrival Time	Departure Time	Registration		RN Interview	
				Time In	Time Out	Time In	Time Out
24	9:00:00 AM	8:05:00 AM	10:12:00 AM	8:20:00 AM	8:30:00 AM	8:40:00 AM	8:57:00 AM
62	8:00:00 AM	7:55:00 AM	10:07:00 AM	8:10:00 AM	8:19:00 AM	8:35:00 AM	9:15:00 AM
82	2:00:00 PM	1:40:00 PM	3:10:00 PM	1:40:00 PM	1:46:00 PM	2:30:00 PM	2:40:00 PM
87	8:45:00 AM	8:30:00 AM	11:35:00 AM	8:32:00 AM	8:42:00 AM	9:10:00 AM	9:50:00 AM
123	10:15:00 AM	9:50:00 AM	11:40:00 AM	9:58:00 AM	10:06:00 AM	10:33:00 AM	10:50:00 AM
472	9:30:00 AM	9:20:00 AM	12:15:00 PM	9:20:00 AM	9:25:00 AM	10:17:00 AM	10:35:00 AM
588	1:30:00 PM	1:10:00 PM	2:10:00 PM	1:15:00 PM	1:20:00 PM	2:00:00 PM	2:10:00 PM

Laboratory ECG		Laboratory Bloodwork		Anesthesiologist		X-Ray in Radiology	
Time In	Time Out	Time In	Time Out	Time In	Time Out	Time In	Time Out
9:01:00 AM	9:13:00 AM			10:00:00 AM	10:10:00 AM		
9:50:00 AM			9:55:00 AM	9:15:00 AM	9:30:00 AM		
2:49:00 PM	3:03:00 PM						
9:57:00 AM	10:05:00 AM	9:57:00 AM	10:05:00 AM	10:36:00 AM	10:48:00 AM	10:52:00 AM	11:25:00 AM
10:50:00 AM	11:05:00 AM	10:50:00 AM	11:05:00 AM	11:05:00 AM	11:15:00 AM	11:20:00 AM	11:40:00 AM
10:40:00 AM	10:45:00 AM	10:40:00 AM	10:45:00 AM	12:08:00 PM	12:15:00 PM		
1:53:00 PM	2:00:00 PM	1:53:00 PM	2:00:00 PM				

5.1 Simulation Results

u24

u24,8:09,arrive,P

u24,Registration,8:09,start service,P

u24,Registration,8:15,leave service,P

u24,RNInterview,8:15,start service,P

u24,RNInterview,8:26,leave service,P

u24,LaboratoryECG,8:26,start service,P

u24,LaboratoryECG,8:35,leave service,P

u24,Anesthesiologist,8:50,start service,P

u24,Anesthesiologist,9:19,leave service,P

u62

u62,7:01,arrive,P

u62,Registration,7:01,start service,P

u62,Registration,7:07,leave service,P

u62,RNInterview,7:07,start service,P

u62,RNInterview,7:17,leave service,P

u62,LaboratoryECG,7:17,start service,P

u62,LaboratoryECG,7:26,leave service,P

u62,LaboratoryBloodwork,7:26,start service,P

u62,LaboratoryBloodwork,7:35,leave service,P

u62,Anesthesiologist,7:35,start servce,P
u62,Anesthesiologist,7:58,leave service,P

u82

u82,13:06,arrive,P
u82,Registration,13:06,start servce,P
u82,Registration,13:11,leave service,P
u82,RNInterview,13:11,start servce,P
u82,RNInterview,13:22,leave service,P
u82,LaboratoryECG,13:22,start servce,P
u82,LaboratoryECG,13:30,leave service,P

u87

u87,7:53,arrive,P
u87,Registration,7:53,start servce,P
u87,Registration,7:59,leave service,P
u87,RNInterview,7:59,start servce,P
u87,RNInterview,8:10,leave service,P
u87,LaboratoryECG,8:10,start servce,P
u87,LaboratoryECG,8:16,leave service,P
u87,LaboratoryBloodwork,8:16,start servce,P
u87,LaboratoryBloodwork,8:23,leave service,P
u87,Anesthesiologist,8:23,start servce,P
u87,Anesthesiologist,8:50,leave service,P
u87,X-Ray,8:50,start servce,P
u87,X-Ray,9:10,leave service,P

u123

u123,9:19,arrive,P
u123,Registration,9:19,start servce,P
u123,Registration,9:25,leave service,P
u123,RNInterview,9:25,start servce,P
u123,RNInterview,9:44,leave service,P
u123,LaboratoryBloodwork,9:44,start servce,P
u123,LaboratoryBloodwork,9:50,leave service,P
u123,Anesthesiologist,9:50,start servce,P
u123,Anesthesiologist,10:15,leave service,P

(*missing X-Ray part)

u472

u472,8:36,arrive,P
u472,Registration,8:36,start service,P
u472,Registration,8:42,leave service,P
u472,RNInterview,8:42,start service,P
u472,RNInterview,8:59,leave service,P
u472,LaboratoryECG,8:59,start service,P
u472,LaboratoryECG,9:08,leave service,P
u472,LaboratoryBloodwork,9:08,start service,P
u472,LaboratoryBloodwork,9:17,leave service,P
u472,Anesthesiologist,9:19,start service,P
u472,Anesthesiologist,9:43,leave service,P

u588

u588,12:32,arrive,P
u588,Registration,12:32,start service,P
u588,Registration,12:39,leave service,P
u588,RNInterview,12:39,start service,P
u588,RNInterview,12:56,leave service,P
u588,LaboratoryECG,12:56,start service,P
u588,LaboratoryECG,13:03,leave service,P
u588,LaboratoryBloodwork,13:03,start service,P
u588,LaboratoryBloodwork,13:11,leave service,P

6 Schedule

Table 24: Testing Schedule

Date Fin- ished	Test Type	Event	Testers
2016-11-20	Initial Tests	Proof of Concept Demonstration	Development Team
2017-02-12	Unit Tests	Demonstration Revision 0	Developement Team
2017-02-12	System Tests	Demonstration Revision 0	Developement Team
2017-03-22	Unit Tests	Test Report Revision 0	Developement Team
2017-03-22	System Tests	Test Report Revision 0	Developement Team
2017-03-22	Non-Functional Tests	Test Report Revision 0	Developement Team
2017-04-04	All Tests	Final Documentation	Developement Team