

ClinicFlow

Test Plan

Maxim Vasiliev (400043983)

Susie Yu (000955758)

Karl Knopf (001437217)

Weilin Hu (001150873)

Yunfeng Li (001335650)

April 9, 2017

Contents

1	Introduction	1
2	Unit Testing	1
2.1	PatientSchedule.py	1
2.2	HealthcareSchedule.py	2
2.3	Clinic.py	2
2.4	SimulationEngine.py	3
2.5	Summary Output	4
2.6	User Interface	4
2.7	Data Storage	5
3	System Testing	6
3.1	Login	6
3.2	Insertion and Storage of Data	7
3.3	View, Modify and Delete Data	8
3.4	View Schedule and Modify Schedule	10
4	Non-Functional Requirements Testing	12
4.1	Usability	12
4.2	Performance Testing	13

1 Introduction

This document details our approach to testing ClinicFlow. This software will be used in a live clinic environment, so it is particularly important to ensure its robustness, reliability, and performance. We begin by testing individual components, modules and methods therein. This allows us to catch errors early in development, and help guide the design process. Next we will perform system testing, where we verify the correct operation of various user tasks. We must also perform user testing to make sure that the product is usable by clinic nurses and managers.

Definitions and Acronyms

- Module: A unit or section in a clinic
- Provider: A health care worker (A doctor, nurse, technician)

Tester

- Testing Team

2 Unit Testing

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.

2.1 PatientSchedule.py

Description

This is the 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.

Criteria

Functions create the patient schedule.

Functions output the schedule in a useable form for the rest of the system.

No.	Initial State	Input	Expected Output
1	No dataset present	No dataset	No output
2	No dataset present	formatted dataset	original dataset
3	No dataset present	unformatted dataset	No output
4	No dataset present	dataset(no repeats)	original dataset
5	No dataset present	dataset(many repeats)	original dataset

2.2 HealthcareSchedule.py

Description

This is the module in the program that will take in a file containing the names and times available of the health care workers at the clinic, and it will be able to generate a schedule for them. We should first test to see if the module is able to accept the appropriate input.

Criteria

Functions create the health care schedule.

Functions output the schedule in a useable form for the rest of the system.

No.	Initial State	Input	Expected Output
1	No dataset present	No dataset	No output
2	No dataset present	formatted dataset	original dataset
3	No dataset present	unformatted dataset	No output
4	No dataset present	dataset(no repeats)	original dataset
5	No dataset present	dataset(many repeats)	original dataset

2.3 Clinic.py

Description

Clinics can have a variety of modules such as Blood taking or x-ray. Each section can be unique or have multiples. This program module uses clinic information from a data file, which will be read into the system when a simulation is run for that clinic.

Criteria

Functions in this program module create a valid network of modules for the clinic.

No.	Initial State	Input	Expected Output
1	No dataset present	No dataset	No output
2	No dataset present	formatted dataset	original dataset
3	No dataset present	unformatted dataset	No output
4	No dataset present	dataset(no repeats)	original dataset
5	No dataset present	dataset(many repeats)	original dataset

2.4 SimulationEngine.py

Description

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. We need to test both under normal conditions and under extraordinary conditions.

Criteria

Functions in this program module can output the simulation results close to the actual sample dataset.

No.	Initial State	Input	Expected Output
1	no simulation data stored	Proper datasets under normal conditions. 25 patients in simulation.	Ouputs close to sample dataset, no more than 10 minutes
2	no simulation data stored	Proper datasets under normal conditions. 50 patients in simulation.	Ouputs close to sample dataset, no more than 10 minutes
3	no simulation data stored	Proper datasets under normal conditions. Only 2 flexible workers.	Ouputs close to sample dataset, no more than 10 minutes
4	no simulation data stored	Proper datasets, but all breaks are simultaneous.	Ouputs close to sample dataset, no more than 10 minutes
5	no simulation data stored	Changed datasets to larger numbers.	Ouputs close to sample dataset, no more than 10 minutes

2.5 Summary Output

Description

The final goal of this product is to generate output which aids clinic employees in decision making. For this, the output must be formatted in a way which conveys a quick and thorough understanding of simulation results and recommended schedule. There will be multiple formats, including visualizations of components as well as formats in text.

Criteria

Functions in this program module can output properly formatted text results.

No.	Initial State	Input	Expected Output
1	No previous output	Genuine data	Original output
2	No previous output	Skewed data	No output
3	No previous output	Invalid data	No output
4	No previous output	Missing Data	No output

2.6 User Interface

Description

The target users of this product includes clinic managers and/or other organizations with similar scheduling goals. The user interface should be clean and intuitive as some of these users might not have extensive knowledge in software products. When the user attempts to make illegal operations, the interface should respond accordingly and direct the user towards correct behaviour.

Criteria

Functions in this program module can response to the valid user inputs.

No.	Initial State	Input	Expected Output
1	no previous interactions	Base case	normal response
2	no previous interactions	Invalid interaction	No response
3	no previous interactions	Aberrant user behaviour	No response

2.7 Data Storage

Description

It is important that the data is able to be saved and recalled. Detailed simulations take vital time, and users would wish to compare with previous results to evaluate the sensitivity of the system to variables. Storage errors should also not interfere with previously stored data.

Criteria

Functions in this program module can successfully save the valid dataset.

No.	Initial State	Input	Expected Output
1	no current data stored	Valid simulation output data	data successfully stored
2	no current data stored	Missing data	data unsuccessfully stored
3	no current data stored	Incorrectly formatted data input	data unsuccessfully stored

3 System Testing

3.1 Login

Description

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.

Criteria

Meet the expected outputs.

No.	Initial State	Input	Expected Output
1	Login Page Empty input field of account and password. Click login.	Empty input of one of the input fields or both.	Stay at same page and error message of empty input.
2	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.
3	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.
4	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.
5	Application main page	Click logout.	Redirect to the login page.

3.2 Insertion and Storage of Data

Description

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.

Criteria

Meet the expected outputs.

No.	Initial State	Input	Expected Output
1	Application main page, admin account.	Click Add Data.	Redirect to the Application adding data page.
2	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.
3	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.
4	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.
5	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.
6	Application main page, viewer account.	Click Add Data.	Error message.

3.3 View, Modify and Delete Data

Description

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.

Criteria

Meet the expected outputs.

No.	Initial State	Input	Expected Output
1	Application main page, admin and viewer account	Click View Data.	Redirect to the View Data page.
2	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.
3	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.
4	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.
5	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.

No.	Initial State	Input	Expected Output
6	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.
7	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.
8	Application view data page, viewer account.	Modify the displayed data.	Stay at same page. Data is not editable.
9	Application view data page, viewer account.	Click delete button.	Stay at same page. Deletion button does not exist.
10	Application view data page, viewer account.	Click save.	Stay at same page. Save button does not exist.

3.4 View Schedule and Modify Schedule

Description

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

Criteria

Meet the expected outputs.

No.	Initial State	Input	Expected Output
1	Application main page, admin and viewer account.	Click view schedule.	Redirect to the view schedule page.
2	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 existing schedule.
3	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.
4	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.
5	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.

No.	Initial State	Input	Expected Output
6	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.
7	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.
8	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.
9	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

Description

- 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.
- 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.
- 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.

Criteria

- The average number of mouse clicks should be less than five.
- The participants can complete the same task correctly in three minutes.
- The participants can complete the required tasks in five minutes and they should encounter fewer than three errors.

No.	Initial State	Input	Expected Output
1	Application is running	Performing generate simulation schedule tasks	Less than five mouse clicks
2	Application is running	New users with instructions to complete import data task	Complete it in three minutes
3	Application is running	New users without instructions to complete import data task	Complete it in three minutes and no more than three errors

4.2 Performance Testing

Description

- 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.
- 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.
- 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.

Criteria

- The average time should be less than five minutes.
- 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.
- The application does not crash and prompts the users that the input data is not correct and how they could correct it.

No.	Initial State	Input	Expected Output
1	Application is not running	Start application and generate simulation schedule	Less than five minutes
2	Application is running	import 100 thousand data	not crash
3	Application is running	import 10000 thousand data	crash