ClinicFlow Software Requirments

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

April 9, 2017

Contents

List of Figures

List of Tables

Revision History

Date	Comments
October 11, 2016	first draft.
April 9, 2017	first revision

Table 1: Revision History Table

Template

This document uses Volere Template for its organization.

1 Changes made

- Changed the scope of the project
- Added Feedback from initial draft
- Added more functional requirements

2 Project Drivers

2.1 The Purpose of the Project

This project aims to produce a tool which allows parties in general clinic sector simulate and analyze various schedules for their clinic. Demand for the solution stems from the lack of relevant products, and the reliance of clinic staff on intuitive and error prone manual scheduling. With this tool, prior patient temporal data will be feed in and used to build a model of the variable involved. A simulation engine will then be used to produce feedback, giving the clinic information on how potential schedule changes could affect their overall operative times.

2.2 The Stakeholders

2.2.1 The Client

Prospective clients of this product would include any clinics providing a set of different procedures, admitting patients to undergo some subset of those procedures, and who wish to automate their scheduling processes. Other potential clients include operators of any systems which make available a set of processes of different durations, which hope to automate scheduling of multiple concurrent uses of said services under various constraints.

2.2.2 The Users

Intended Users include employees of clinics or systems described above. In both cases, users are under time/resource constraint to perform their assigned scheduling functions.

2.2.3 The Patients

This group includes the individual components to be scheduled through the system. In the case of patients, this would concern individuals whose well-being and quality of service is of importance to themselves and their satisfaction of system performance.

2.2.4 Other Stakeholders

Other Stakeholders of this project include:

- Supervisors: Dr. Wenbo He, Dr. Douglas Down, Steve Metham
- Clinic manager:

3 Project Constraints

3.1 Mandated Constraints

3.1.1 Cost Constraints

There is no immediate financial cost associated with project development. Clinic employees may have to devote some time for requirements gathering and product testing sessions. If however the system is successful and to be put in place permanently, use of a dedicated server will be required. For this, a cloud hosting and compute service such as Amazon Web Services would suffice. Financial cost for hosting and data access should be minimal, while compute costs (the run time of the program) will be determined during development, and dependant on algorithm performance.

3.1.2 Quality Constraints

One measure of project success is the quality of results. Since program output will be used in a live clinic environment, it is important that the proposed feedback be verifiable in production. This requires that changes made to schedules be tested in situ, and performance compared to manually scheduled historical data. This proof is necessary to implement use of the product in field.

3.1.3 Risks

Building on the quality considerations above, one evident risk would be failure of generated schedules to demonstrate improvements over control (previous method of scheduling). This may cause bottlenecks or backlogs in patients processing, placing additional stress on clinic employees and institution finances.

3.1.4 Scheduled Deliverable and Development Time

The final deadline for the project is mid April 2017. The detailed deliverable and their respective deadlines are listed below:

- Requirements Document - revision 0: October 12, 2016
- Proof of Concept Plan: October 26, 2016
- Test Plan - Revision 0: November 2, 2016
- Proof of Concept Demonstration: November 21, 2016
- Design Document - Revision 0: January, 11, 2016
- Demonstration - Revision 0: February 13, 2017
- Users Guide - Revision 0: March 1, 2017
- Test Plan - Revision 0: March 22, 2017
- Final Demonstration: Mid-April, 2017
- Final Documentation: April 5, 2017

3.2 Required resources

3.2.1 Available Data

Historical patient arrival and procedure duration times will be provided by the client. These will be used to build the model for the variables such as deviation between patient scheduled and actual arrival times, and average and standard deviation of performed procedures.

Clinic constraints and business rules will also be provided by the client. This includes facts such as the number and shift of clinic employees, as well as rules such and allotted break times and intended shift end times for specific employees.

3.3 Relevant Facts and Assumptions

3.3.1 Relevant Facts

- There are hundreds, if not thousands of clinics in Canada that could use this software

3.3.2 Assumptions

- That clinics all operate during the day and during regular business hours, otherwise a more dynamic time conversion would be needed
- That the clinic would not want to store more data than our servers would be able to hold.
- That a health care provider can only work at one module at once, and not conduct two or more simultaneously
- That a patient will need the modules they are scheduled to receive, not an additional amount or a lower amount
- That all providers arrive on time, and are able to work right away
- That all providers will stay until the closing of the clinic
- That all patients will stay until they finish the clinic, and not leave early
- That there will be no down time in modules due to equipment failure, or other external effects

4 Functional Requirements

4.1 The Scope of the Product

A web-interface application can allow users to insert necessary data, such historical patient procedure durations, doctor and nurse shift hours, and other constraints such as break allotments or soft constraints such as employee shift end times. Based on provided data and inputs, the system will simulate a "day" in the clinic and record important information about the simulation. This tool will be designed with the intention to be implementable in multiple health care institutions, so other clinics would able to make use of the tool.

4.1.1 Product Boundary

Users gather the information from patients, doctors and clinic departments as input into the application, and receive data sets and charts as outputs from the application. The inputs include historical patient data (arrival time, procedure time, departure time), doctor work hours, and other constraints. The application stores the inputs into a database, and extracts the data when generating simulations. The simulation results are then stored in the data base to be recalled at a later point. Users can update the database and adjust schedules at any time, rerunning the simulation engine to produce an updated prediction of daily clinic flow statistics.

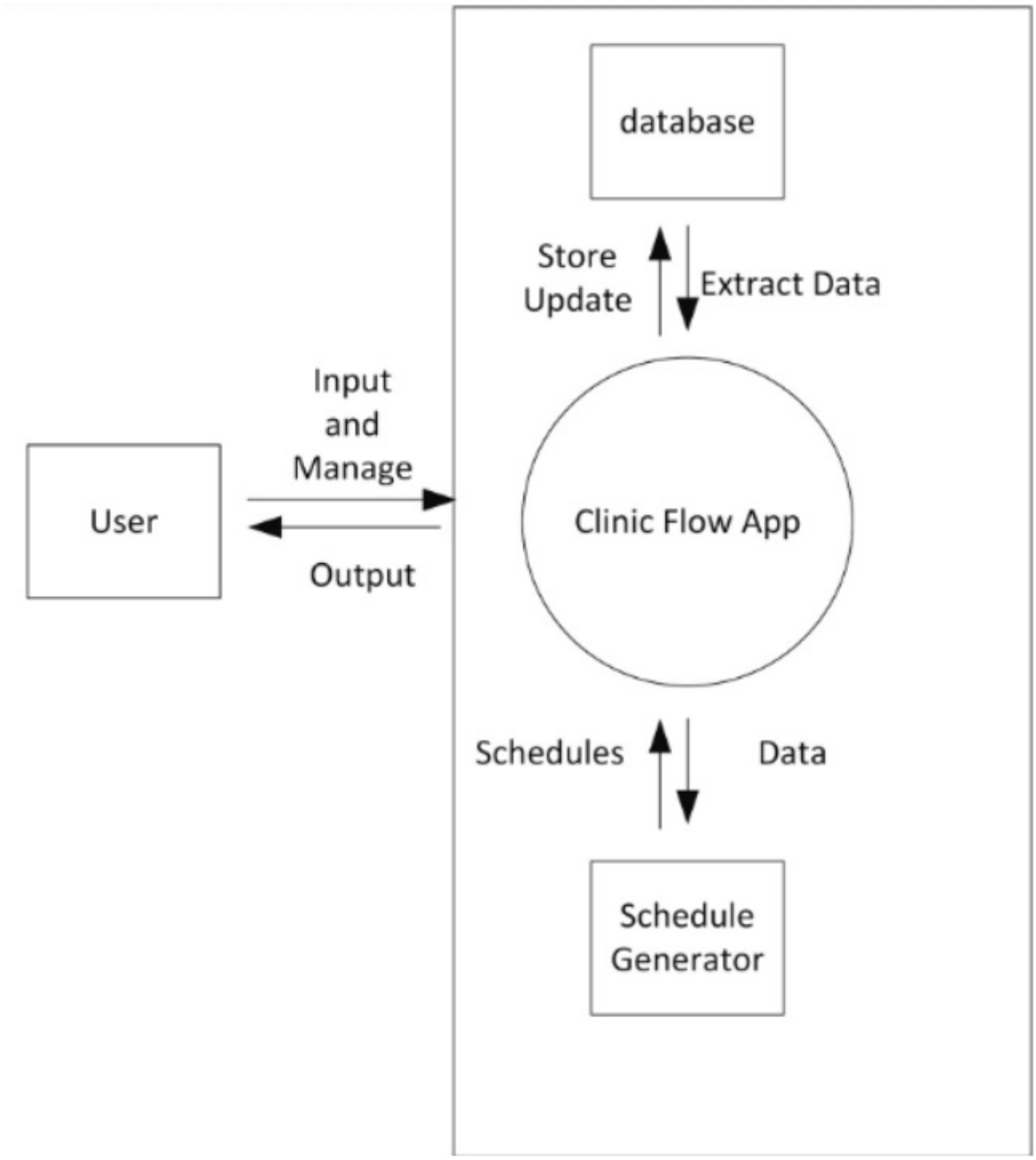

Figure 1: Product Boundary Diagram

4.1.2 Product Use Case Table

PUC#	PUC Name	Actors	Summary
1	Login	User	Access the application
2	Insert Information	User	Insert information of patients and doctors
3	Store Information	User	Save the entered data to database
4	Display Data	User	View the data
5	Update Data	User	Changer the data inside database
6	Generate Simulation	User	Assign patients to procedure slots
7	Store Simulation	User	Save generated simulations into database
8	Display Simulation Results	User	View existed simulations
10	Log out	User	Leave the system and close the authority

Table 2: Use Case Table

4.2 Functional Requirements

Requirement 1
Description:
The product shall verify the users authority to access the full system.

Fit Criterion:
The product should require user to pass one or more security guards before the user start managing the system.

Requirement 2
Description:
The product shall allow the user to enter data.

Fit Criterion:
Users can enter any type of data into the system such as strings, integers and datetimes. Then the application must send the inputted data to the database.

Requirement 3
Description:
The product shall allow the user to manage the information in the database.

Fit Criterion:
Users can update and modify the data inside database at any moment. Users can create more data types or tables for dealing with changes in the future.

Requirement 4
Description:
The product shall be able to show the data inside database to user in an informative way.

Fit Criterion:
Users can open and view details of all data inside database. The data includes the information from patients, information from doctors, and previously generated schedules.

Requirement 5
Description:
The product shall allow users to adjust generated visuals.

Fit Criterion:
The user can adjust the previously generated visualizations of the simulation results. The user could update the times displayed, or look at one section of the graph.

Requirement 6
Description:
The simulations generated by the program should be realistic. The results should accurately approximate

Fit Criterion:
The results should accurately approximate a real day in the clinic. There should not be wild deviations from reality, such as patients waiting an unresonable amount of time. This also means that things like the First Come-First Served prinicple that is implemented in the clinic is enforced by the program. The user should be able to look at the results of a single simulation and trace the flow of each patient through the clinic, seeing that they all follow a logical path

Requirement 7
Description:
The program should be able to handle any resonable set of patients for a given run.

Fit Criterion:
A reasonable amount of patients would be the amount of expected patients that the clinic would see in a single day. For the Pre-Op clinic, this would be around 25 people at most.

Requirement 8
Description:
The program should be able to handle any reasonable schedule for provider break times.

Fit Criterion:
A reasonable schedule for provider break times would involve the program beign able to handle the periodic breaks each provider must take during the day to comply with worker's regulations. The program must also be able to handle lunch breaks, and extended breaks where a provider moves to a different part of the hospital.

Requirement 9
Description:
The program should be able to generate useful and quality data for each run of the simulation. It should be able to display the results of the simulation in a readable way.

Fit Criterion:
The data gathered by the simulation for a single run should be data that the clinic can actually use to improve their schedules. It should not be data on irrelevant subjects, or things they have no ability to change. The program should also be able to convert the discrete time step used into a more readable form for a human.

Requirement 10
Description:
The program should be able to run multiple simulations of the same clinic

and generate different valid results.

Fit Criterion:
There should be sufficient variation in the output of the simulation, as no two days in the clinic are identical. If the simulations are unable to generate this reasonable variation then they are not of real worth to the clinic.

Requirement 11
Description:
The program should be able to add or subtract modules from the clinic.

Fit Criterion:
There should not be a fixed clinic structure for the simulations. Modules in the clinic are fluid in reality, with modules being added and subtracted every revision cycle. If our program can not include this feature, it would be unable to used in the long term. This requirement would also allow the program to be used for another clinic at another time.

Requirement 12
Description:
The simulation engine should use the real data to populate the variation for its simulations. It should be able to use different data to change the variation of each simulation

Fit Criterion:
The program should be able to pull data from the database, and process that data to find measures of variation (such as means and standard deviations) of the data set. It should then be able to use those measures to change the variance in the simulation

Requirement 13
Description:
The system should be fault tolerant. It should be able to handle any errors in the simulation engine.

Fit Criterion:
The program should be able to at least be able to determine if the results of the simulation are completely reasonable (i.e. the simulation shows no wait

time for patients). There should also be error handling features present in case something goes wrong in the transfer of data between sub-systems.

Requirement 14
Description:
The system should be able to store the previous simulations in the data base for future access.

Fit Criterion:
The program should have a set of methods to store and access simulation results on the data base. It needs to be able to store a reasonable amount of previous simulations (≈ 100).

Requirement 15
Description:
The system should be able to store the state of the program when a user logs out. Then when the user logs back in, they should be able to return to that same state.

Fit Criterion:
The program should have a set of methods to store the current state of system for a particular user on the database. Then when that user logs back into the system, the program should be able to return to that state.

Requirement 16
Description:
The system should be able to return to a main menu if the simulation fails in any way.

Fit Criterion:
The program should have a set of methods to make sure that if there is an error in the simulation engine, then the program will not crash and will return to the previous menu. The program should be able to abandon a simulation if it proves to difficult to compute.

5 Nonfunctional Requirements

5.1 Look and Feel Requirements

Description:
The user interface for this application will be clean in design. It will provide simple menus from which the user can select desired tasks. There will be a window to view the output, and an attached menu with options to save the results.

Fit Criterion:
The application should have no unnecessary windows and menus. The application should generate no complaints from users about being unable to find a specific option.

5.2 Usability Requirements

5.2.1 Ease of Use Requirements

Description:
Intended users include managers of preoperative clinics, or other organizations with similar scheduling goals. While these parties may not be experienced using complex software and non-graphical user interfaces, it is assumed they have some experience using common software products. As such, the program interface and output should be intuitive and readily interpretable

Fit Criterion:
The application interface should be graphical. Program output should be presented and formatted in a manner which quickly and clearly conveys pertinent information, as well as being able to be shared and distributed amongst clinic staff.

5.2.2 Language Requirements

Description:
The application should be written in and provide feedback in Canadian English. Future version may may support for other languages.

Fit Criterion:

Only Canadian English will be used in the graphic interface and outputs. Design the software so that the language may be eventually changed.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

Description:
The program should respond quickly to user input. The application should run the simulation in reasonable time, given the data size.

Fit Criterion:
Interactions between user and system shall have a maximum response time of 2 seconds. Each simulation should take a resonable amount of time depending on the data size. An approximation of this would be around a minute per 100 data values at maximum. This might not be reasonable for larger data sets, as the computation cost compound.

5.3.2 Reliability and Availability Requirements

Description:
The product shall be available for use as frequently as possible.

Fit Criterion:
The product should not need any downtime to function. If the system is web based, then it should try to have 99% up time. Ideally it would always be able to be run outside of server and program maintenance time. It should at a minimum be available during the working hours of the hospital adminstrative staff (8:00am - 5:00pm Mon/Fri).

5.3.3 Scalability or Extensibility Requirements

Description:
The product should be able to handle the initial data set, around 900 data points but it should be able to scale to at least 10000 data points as the data set grows.

Fit Criterion:
The simulation engine of the program should not be bound by the size of

the data. The application should be able to save the necessary data in an accessible format.

5.3.4 Robustness or Fault Tolerance Requirements

Description:
The program should prompt the user if it is unable to run the simulation on the first try. As well if the user tries to run an operation and it fails, the user should be notified.

Fit Criterion:
A warning message will be provided to the user if the simulation is unable to be run. A warning message will be provided to the user if their selected operation can be be done.

5.4 Maintainability and Support Requirements

5.4.1 Maintenance Requirements

Description:
Changes to system will be done within one week of being presented to the maintenance team. Changes will be applied to the application as soon as they are available.

Fit Criterion:
There is only a delay of a week between the report of an error in the software, and a fix being applied. There should be minimal down time between the creation of a fix and its application to the program.

5.4.2 Supportability Requirements

Description:
The user of the software should have direct access to the support team, and be able to make a request directly to the team.

Fit Criterion:
A system to allow the client to report directly to the team shall be implemented.

Description:
There should be a member of the support team assigned to the project for at least one year after the completion of the project

Fit Criterion:
A member of the support team (taken from the development team) shall continue to support the product for a year after project completion or until the product is no longer in use, which ever comes first.

5.4.3 Adaptability Requirements

Description:
As the application will be web-based, it should be accessible from any operating system that has internet browsing capability. Furthermore, if needed the application can be made to run locally on some machine by installing all server components on said machine.

Fit Criterion:
The application runs on Amazon Web Services. Adaptability to other systems should be considered during system design.

5.5 Security Requirements

5.5.1 Access Environment

Description:
Managers of the clinic posses full administrative access to the program. Doctors and nurses may be able to see historical results of simulations but may not run them.

Fit Criterion:
The program should have two sets of credentials granting access to it. Credentials consist of a username and a password. One set grants access to full administrative level program functions, while the other allows only observational access.

5.5.2 Privacy Requirements

Description:
The system should not store any personal information, and the data should be clean of any personal information before being entered. The output should use generic names (Nurse 1) to create a schedule, the assignment of the generic names to actual people is left up to the manager.

Fit Criterion:
Historical patient data used to build the model will have all personally identifiable information removed prior to being handed to the development team. The application is unable to save personal information. The application does not use specific names when creating a schedule.

6 Project Issues

6.1 Design Decisions

The implementation and working environment dependencies:

- HTML, CSS and JavaScript will be used to write the graphical user interface.
- PHP will be used as an interface to the database due to its simplicity
- MongoDB will also be used due to its simplicity and non-essentiality of a relational model for data storage.
- The simulation and optimization engine will be written in Python 3. Specifically, the SimPy package will be used for discrete event simulation components.
- The application can be installed locally on a computer, or it can be accessed from a remote server.

6.2 Off the Shelf Solutions

6.2.1 Ready Made Products

There are many commercial simulation software existing in the market, however, they do not offer the level of customization and patient flow simulation

functions.

6.2.2 Reusable Components

Because the source code of commercial software is not available, there is no reusable component.

6.3 Costs

The product will be open-source, but if it is installed on a server, maintenance fees and Internet fee may apply. There is currently a small fund set up by the development team to pay for expected server access and maintenance costs associated with the project. If the costs exceed the expected amounts, additional funds must be accquired or the project will have to finish early.