qFlow
Project Definition Document

Document Version: 1.1
AUTHORS

This document was prepared by:

Nicholas Lasswell  
Borealis  
Phone: (618) 308-2049  
nicholas.lasswell@siu.edu

Nicholas Pennington  
Borealis  
Phone: (217) 883-8149  
npennington@siu.edu

Michael Smith  
Borealis  
Phone: (618) 571-1256  
michael.r.smith@siu.edu

VERSION HISTORY

<table>
<thead>
<tr>
<th>Date</th>
<th>Document Version</th>
<th>Document Revision History</th>
<th>Document Author/Reviser</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 10, 2017</td>
<td>1.0</td>
<td>Initial draft</td>
<td>Nicholas Pennington</td>
</tr>
<tr>
<td>November 14, 2017</td>
<td>1.1</td>
<td>Reviewed and revised during developer meeting</td>
<td>All Members</td>
</tr>
</tbody>
</table>

APPROVALS

<table>
<thead>
<tr>
<th>Date</th>
<th>Document Version</th>
<th>Approver Name and Title</th>
<th>Approver Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

1. Purpose 4
2. Problem/Opportunity 4
3. Project Goal 4
4. Project Objectives 4
5. Project Scope 4
6. Key Stakeholders 4
7. Outcomes/Success Criteria 4
8. Assumptions and Constraints 4
   8.1 Assumptions 4
   8.2 Constraints 5
9. Risks 5
10. Estimated Duration 5
11. Functional Requirements 5
12. Non-Functional Requirements 5
13. Technical Requirements 5
14. Use Cases 5
15. Data Elements 5
1. **Purpose**
This Project Definition Document provides a brief overview of qflow to promote a shared understanding of it before a more detailed plan and schedule are prepared.

2. **Problem/Opportunity**
The problem this project seeks to address is the problem of optimizing stochastic flow charts. Many systems can be modeled as networks of interconnected queues, from emergency rooms to communication in computer networks. It is, however, nontrivial to find optimal configurations for these systems, given factors such as cost and throughput. This project seeks to solve this optimization problem in a way that can be applied to many different fields.

3. **Project Goal**
The goal of this project is to develop a program, which given a stochastic flow chart and criteria to optimize, will recommend a set of parameters to maximize the specified criteria.

4. **Project Objectives**
- Develop a convenient user interface (GUI) through which users can easily specify the stochastic flow chart and property fields (such as number of servers per queue), directed connections, and desired optimization criteria.
- Simulate the flow through the model, clearly showing how it changes over time
- Create a genetic algorithm to optimize the queuing model given optimization criteria.
  - Encode the model as a chromosome
  - Determine fitness function representation

5. **Project Scope**
Deliverables:
- Clean and simple user interface for entering models of queuing systems
- Ability to simulate the queuing systems using the model
- Ability to optimize the queuing systems given optimization criteria

6. **Key Stakeholders**
The key stakeholders for this project include Dr. Rahimi and us, Team Borealis.

7. **Outcomes/Success Criteria**
- The user interface must be simple and easy enough to understand so as to be accessible to people of any discipline
- Given a model of a real-world scenario, the program must be able to measurably improve the system configuration.
- The program must return recommendations in an easy-to-understand format.

8. **Assumptions and Constraints**

8.1 **Assumptions**
- Queues and the connections between them are final
- The total number of resources is fixed.
• The queue property fields provided will, within reason, accurately reflect their real-world counterparts.

8.2 Constraints
• The project will be under time constraints, as development is restricted to one semester.

9. Risks
It is possible this project is not able to meet the requirements within the time constraints. This can be mitigated by time management and communication between group members and stakeholders.

10. Estimated Duration
The duration for the project will be about one college semester, as constrained by the senior project class.

11. Functional Requirements
• Ability to easily enter a queueing model into the program
  • Users must be able to enter general queueing models into the program with minimal effort
  • Queue model must include parameters such as arrival rate for the system, service rate for each queue, number of servers in each queue, etc.
  • Each customer that enters the model will be assigned a priority.
  • Connections between nodes are represented as vectors, with each different priority of customer having different priorities to transition to different nodes.
  • The model must be able to prevent cycles by allowing connections that can only be taken once.
  • The model must allow connections that are only open during specific hours of the day.
• Ability to simulate behavior of queueing model
  • Users must be able to run the queueing model and be able to visually see how the system changes over time
• Ability to choose optimization criteria
  • Users must be able to specify what criteria the model should be optimized by, prioritizing factors such as cost and throughput
• Ability to optimize the given queueing model
  • Program must have the capability to optimize given queueing networks by the given optimization criteria

12. Non-Functional Requirements
• Simple user-friendly interface
  • Data entry and program use must be easy to use for users of any discipline

13. Technical Requirements
• The program is to be represented as a web application.
• Backend of the web application is to be written in C++
• Objection orientation and efficiency are core requirements in this project
• Object Oriented Design:
  • Model:
    • The model itself contains a set of nodes, and a set of edges.
    • The model contains a value for arrival rate of clients.
    • The model contains a direct reference to the designated entry node.
    • The model contains probabilities for clients to come in with specified priorities.
  • Node:
    • Each node represents a state in the system.
    • The nodes contain a set of queues, and handles distribution of clients between these queues.
  • Edge:
    • Edges represent directed connections between states in the system.
- Each edge references either two nodes, or just one. (In the case of an exit node)
- Each edge contains a vector of weights, containing probabilities for transfer for each priority of patient.

  o **Queue:**
    - Each queue acts as a priority queue, processing patients in order of priority.
    - The queue contains a set of servers, and matches clients to the servers.
    - The queue contains the clients currently in the queue.

  o **Server:**
    - Each server has a distribution deciding processing rate for clients.
    - Servers have the ability to process clients.

  o **Patient:**
    - Each client contains an ID and a priority.
    - The clients travel through the model probabilistically.

  o **Resource Pool:**
    - Resource pools are pools of resources that the server needs to process clients.
    - The pools contain a fixed number of resources

14. **Use Cases**
- A hospital wants to minimize the average amount of time taken to process a patient.
- A mechanic's garage wants to maximize client throughput.
- A government office wants to minimize the amount of time a document spends without review.

15. **Data Elements**
- Number of queues
- Service Discipline: Behaviour of the queues
- Directional connections between queues
  - Each connection with an associated transition vector
- Servers per individual queue
- Arrival rate of customers
- Processing efficiency of servers (variable per class of server)
- Throughput (per queue and system-wide)

Date: Needs more work

Approved by: Shahram Rahimi

Approver Signature: [Signature]

Mentor Name: Shahram Rahimi

Mentor Signature: [Signature]