I. System Interface

A. The interface shows concentrations (densities) of cells and nutrient concentrations in a chemotstat chamber over time.

i. Concentration changes are visible through animated chambers of a cartoon chemostat.

a) The chemostat is depicted with at least a nutrient reservoir, culture chamber, and conduits for inflow and outflow.

ii. The system also shows, either by default or by user prompting, concentrations in a more quantitative form (a plot or numerical values) that correspond to the current state of the depicted cell and nutrient concentrations.

B. The Interface shows parameter values.

i. Parameters:

a) Volume of the chemostat chamber.

b) Volume of the nutrient reservoir .

c) Chamber Inflow/outflow rate (volume/time).

d) Initial concentration of cells (number/volume).

e) Initial nutrient concentration in the reservoir (mass/volume).

f) Cell variety (for growth rate as a function of nutrient concentration). Growth rate is calculated using Michaelis-Menton kinetics The user can choose from a variety of cell types, with different growth rates that yield different Michaelis-Menton constants.

g) Total time the chemotstat runs.

ii. Parameter values are indicated:

a) By shape, color, or other visual cues given by the chemostat components.

b) By supplemental, maybe hidden until user-prompted, numerical values.

C. The interface gives the user control over the parameters.

i. All system parameters are set by manipulating the depicted chemostat (enlarging or shrinking the chamber, for example), or depictions of other objects, rather than using conventional GUI controls such as sliders and drop-down lists.

D. Online help is available both to instruct in the use of the simulation and also to give biological background of chemostat operation and use.

i. The interface will be intuitively usable in the default use case (see Use Cases below), and with minimal user-exploration in the other cases, that is, usable without the help documents.

a) Documented usage instructions for the simplest use case will be provided for naive computer users and to keep a complete record of the system's controls.

II. System Engineering

A. System Development Tools

i. The system will be coded in Java 6.
ii. All modules will be developed as NetBeans projects.
iii. All *.java file revisions will be tracked using the class Subversion server.

B. The system will be composed of 3 subsystems:

i. Numerics
   a) Interacts with the Model subsystem (see below).
      Given data from the Model subsystem the Numerics module returns the cell and nutrient concentrations in the chemostat chamber for the next time step.
ii. Model
   a) Interacts with the Numerics subsystem
      Provides it with the data it needs to solve chemotstat concentrations. Receives solutions from the Numerics subsystem.
   b) Interacts with the View subsystem (see below).
      Provides it with chemotstat concentrations. Receives data from the View subsystem.
iii. View
   a) Interacts with users
      Presents users with chemotstat information. Listens for and accepts user-entered data.
   b) Interacts with the model subsystem.
      Receives chemostat state data. Provides the model with data resulting from user interactions.

C. The system will provide online help documents.

i. Enumerates all intended user interactions with the system, including how to set each parameter and how to start and stop the chemotstat.
ii. Explains the significance of all color/shape changes in the chemostat representation, and how to retrieve or locate the numerical output that corresponds to the chemostat.
iii. Explains the biological and mathematical principles underlying chemotstat-behavior over time.

D. Subsystems will be separable.

i. They will communicate only through Java interfaces (method calls in objects by class or Java Interface definition).

E. The system is usable (responsive, accurately-rendered) on computers with:

i. A typical desktop or laptop system specification.
ii. A Windows, Mac, or Linux operating system.

III. System Documentation

A. In the source code, all classes and methods are documented in html files using the javadoc compiler. All java classes are commented as to their role in the system, as well as noting dependencies not obvious from the member list (such as non-standard libraries). Methods are commented for purpose as well as data type.

   i. Class summaries and method comments give the level of detail seen in documentation of Sun's Java API specification (see).

      a) Note that the NetBeans IDE has an autofill feature that automatically inserts delimiters and tags for documentation before a class or method declaration. The coder only needs to fill in comments on class, method or method arguments.

B. Coding Practices

   i. Variable names are meaningful. They reflect the role of the stored quantity in the computations that use it.

   ii. As the logic of an algorithm is implemented, it is explained at each step with brief comments.

   iii. Any unusually complicated or subtle operation is explained in full.

IV. Use Cases

A. A new user starts the program.

   i. A chemostat appears with default parameters that provide a steady state solution for a default cell type.

   ii. The user can initiate inflow and outflow in the chemostat without recourse to help documents.

   iii. The chemotstat concentrations are shown over the default time, correct for the default parameters.

B. The user changes any combination of the parameters listed above and then runs the chemotstat.

   i. The chemotstat concentrations are depicted over time, correct for the user-set parameters.

   ii. Nonsensical parameter setting is not allowed (such as negative volume, time, or concentrations).

   iii. The chemotstat does not depict negative concentrations of cells or nutrients.

   iv. The numerical representation of the concentrations and parameters agree with those depicted.

C. The user stops the chemotstat flow at any time during a simulation,

   i. With reasonable responsiveness the state of the chemotstat (parameter settings, cell and nutrient concentrations) is preserved on the interface.

   ii. Numerical representation of the concentrations corresponds to the animated state.

D. The user changes parameters while the chemotstat is running.

   i. The effect of the changes are visible in the cell and nutrient concentrations.

ii. The parameter changes are seen in the numerical representation of the parameters.

E. The user changes the cell type and initiates chemostat flow.

i. On a change of cell type the parameters other than cell type are automatically set to give a steady state solution for the cell type.

ii. The simulation correctly shows concentrations for the selected cell type, reaching a steady state given sufficient time.

F. While running the chemotstat, the user deactivates the simulation window.

i. The flow is interrupted and the state is preserved as in the case when flow is stopped by user control.

ii. Reactivation of the window initiates flow from the state in which flow was interrupted.

G. While running the chemotstat, the user iconifies the window.

i. The simulation continues as if the window were still visible.