# Use-Case Scenarios for Molecular Genetics eXplorer

## Executive Summary

Software developers rely on use cases as they model the requirements for new applications. Each formal use case that the developers write is associated with a set of informal narratives. These narratives, called use-case scenarios, serve to establish the nature of the interactions that will take place between the user (the actor) and the planned application (the system). It is therefore important for the success of a software project that the customers as well as the developers review and agree on the use-case scenarios.

The Molecular Genetics eXplorer (MGX) is a computing application that will help students to understand the connections between genetics, molecular biology, and biochemistry. MGX provides an interface for three virtual biological laboratories (VBLs): a virtual genetics lab (VGL) for genetics, a gene exploration lab (Genex) for molecular biology, and a protein exploration lab (Protex) for biochemistry.

The use-case scenarios presented in this document are grouped according to type of user—student or administrator--and operating mode—integrated or stand alone. In integrated mode, MGX will display a zoo of creatures, and each VBL will make use of creatures taken from this zoo. In stand-alone mode, MGX will not display any zoo, and each laboratory will function as it currently does outside of MGX, that is, as a stand-alone application.

The following sets of of use-case scenarios are presented in this document. A customer signature area is found at the end of each set of scenarios.

1. Student Integrated Mode (SIM)
2. Administrator Integrated Mode (AIM)
3. Student Stand-Alone Mode (SSAM)
4. Administrator Stand-Alone Mode (ASAM)

MGX is based on computer science projects developed by students during previous years. The following works are incorporated here by reference.

Karki, C., P. Kejriwal, S. Adma and Z. Zhu, 2003: *GenExplorer: Software Requirements Specification, Version 1.0.* Department of Computer Science University of Massachusetts, Boston, 55 pp.

Koolar, N., N. N. Maw, C. Y. Yu, and W. Ma, 2002: *Software Requirements Specification for Virtual Genetics Lab, Version 1.0.* Department of Computer Science University of Massachusetts, Boston, 23 pp.

**Student Integrated Mode**
**SIM Scenarios**

SIM.1. The student opens MGX in student integrated mode.

The student starts the application. MGX displays a list of choices: *New Session*, *Saved Session* and *Quit.*

SIM.1.1 *New Session*

The student selects *New Session.* MGX presents a zoo holding creatures and a choice of states. Each state corresponds to a virtual biological laboratory (VBL) where the student is able to work: a virtual genetics lab (VGL), a gene exploration lab (Genex), or a protein exploration lab (Protex).

In VGL the student mutates the genotype of a single creature for one trait (color), and observes the resulting changes in its phenotype; or the student crosses two creatures, and examines their progeny. In Genex the student interprets the transcription and translation of DNA to protein. In Protex the student manipulates and views the structure and function of a protein.

SIM.1.2 *Saved Session*

If the student has already used MGX, and wants to start where she left off, she selects *Saved Session*. MGX prompts for name of file holding the last saved state. The student provides the filename. MGX loads the saved state, and displays one, two or three VBLs, as is described above (in SIM.1.1).

SIM.1.3 *Quit*

The student selects *Quit*. MGX quits.

SIM.2. In VGL.

MGX displays a large cage that can hold many creatures and two small cages that can hold just one creature each.

SIM.2.1. In VGL, the student selects one creature.

The student selects one creature from the zoo or from the large cage. MGX displays the selected creature in one of the small cages and displays the creature's genotype. The student indicates that she wants to *mutate* this genotype. MGX randomly generates a population of creatures in the large cage. Each of these creatures exhibits a randomly mutated version of the original creature's genotype. The student associates changes in the creature's genotype with changes in its phenotype (color).

SIM.2.2. In VGL, the student selects two creatures.

Student selects two creatures: both from the zoo, one from the zoo and one from the large cage, or both from the large cage. MGX displays each of the selected creatures in its own small cage. The student indicates that she wants to *cross* these two creatures. MGX clears the large cage, and displays in that cage the offspring of the crossed creatures.

SIM.2.3. In VGL, the student saves new creatures.

The student saves new creatures generated in scenario SIM.2.2 by moving them from the large cage into the zoo.

SIM.2.4. In VGL, the student empties the large cage.

The student indicates that she wants to empty (clear) the large cage. MGX asks the student to confirm that she wants to clear the large cage—*All creatures will be lost*. The student confirms that she wants to clear the large cage. MGX removes all creatures from the large cage.

SIM.2.5. In VGL, the student removes a creature from a small cage.

The student indicates that she wants to empty (clear) one of the small cages. MGX clears the creature from that cage, and returns that creature to the zoo.

SIM..3. In Genex.

MGX displays a small cage large enough to hold a single creature. There are also two views for displaying copies of the creatures DNA sequence, which is *editable*, the mRNA and protein associated with the two genes determining one trait (color) taken from a homologous pair of chromosomes.

SIM.3.1. In Genex, the student selects one creature.

The student selects a creature from the zoo. MGX displays the selected creature in the small cage and copies of DNA, mRNA and protein.

SIM.3.2. In Genex, the student modifies one creature.

The student edits the nucleotides in the DNA sequence associated with either of the genes. MGX modifies the associated messenger RNA, the protein and the appearance (color) of the creature in the cage.

SIM.3.3. In Genex, the student saves a (modified) creature.

The student indicates that she wants to save a modified creature. MGX puts a copy of the modified creature in the zoo.

SIM.3.4. In Genex, the student discards a (modified) creature.

The student indicates that she wants to discard a modified creature. MGX asks the student to confirm that she wants to discard the modified creature, which is in the small cage. The student confirms that she wants to do so. MGX removes the creature from the small cage.

SIM.4. In Protex.

MGX displays a palette holding the 20 common amino acids. There are also two view for displaying *editable* copies of the polypeptide chains (linear and folded) plus the function (color) of the protein associated with two genes (alleles) from a homologous pair of chromosomes.

SIM.4.1 In Protex, the student selects one creature.

The student selects one creature from the zoo. MGX displays the selected creature in the small cage, copies of the polypeptide chains (linear and folded) and the protein function (color).

SIM.4.2. In Protex, the student modifies one creature.

Student adds (removes) one amino acid to (from) the existing polypeptide chain of either gene. MGX folds the modified polypeptide chain, displays its altered function (color), and modifies the appearance of the selected creature in the small cage.

SIM.4.3. In Protex, the student saves a (modified) creature.

The student indicates that she wants to *save* a creature inhabiting the small cage. MGX puts a copy of the modified creature in the zoo.

SIM.4.4. In Protex, the student discards a (modified) creature.

The student indicates that she wants to discard a creature inhabiting the small cage. MGX asks the student to confirm that she wants to discard this creature. The student confirms that she wants to do so. MGX removes the creature from the cage.

SIM.5. The student prints the results of her session.

The student selects *Print*. MGX sends an image of the current (active) VBL GUI to the printer.

SIM.6. The student closes MGX in integrated mode.

The student chooses to exit the application. MGX prompts for the name of a file it will use for saving its state, including all creatures in the zoo and in cages (VGL). The student enters a filename, or indicates that she wants MGX to discard the session. If the student enters a filename, MGX saves its state. MGX quits.

**Signature**

I approve these student integrated mode (SIM) use-case scenarios.

________________________________ ________________

Professor Brian White, Customer Today's Date

## Administrator Integrated Mode
## AIM Scenarios

AIM.1. The administrator opens MGX in administrator integrated mode.

The administrator starts the application. MGX displays a list of choices: *New Session*, *Saved Session* and *Quit.*

AIM.1.1 *New Session*

The administrator selects *New Session.* MGX displays a zoo holding creatures, an options view and a choice of states corresponding to the virtual biological laboratories (VBLs) described above in scenario SIM1.1. In the options view, the administrator is able to turn on and off all VBLs, and to add and remove creatures from the zoo. A VBL that is turned off is not visible.

The appearance, features and functionality of each VBL are the same for the administrator as they are for the student (see scenarios SIM.2 through SIM.4).

AIM.1.2 *Saved Session*

If the administrator has already set up MGX, and wants to implement his custom configuration, he selects *Saved Session*. MGX prompts for name of file holding the last saved state. The administrator provides the filename. MGX loads the saved state, and displays the options view plus VBLs.

AIM.1.3 *Quit*

The administrator selects *Quit*. MGX quits.

AIM.2 In VGL.

MGX displays a large cage that can hold many creatures and two small cages that can hold just one creature each. See scenarios SIM.2.1 – SIM.2.5

AIM.3 In Genex.

MGX displays a small cage large enough to hold a single creature. There are also two views for displaying copies of the creatures DNA sequence, which is *editable*, the mRNA and protein associated with the two genes determining one trait (color) taken from a homologous pair of chromosomes. See scenarios SIM.3.1 – SIM.3.4.

AIM.4 In Protex.

MGX displays a palette holding the 20 common amino acids. There are also two view for displaying *editable* copies of the polypeptide chains (linear and folded) plus the function (color) of the protein associated with two genes (alleles) from a homologous pair of chromosomes. See scenario SIM.4.1 – SIM.4.4.

AIM.5. In the options view.

In the options view, MGX displays a creature supply pool (CSP) where the administrator can store creatures temporarily and a list of VBLs. If the administrator selects a VBL that VBL is made visible.

AIM.5.1. In the options view, the administrator selects VGL.

The administrator selects VGL. MGX changes the state of VGL, that is, MGX makes it visible or visible, depending on its last state.

AIM.5.2. In the options view, the administrator selects Genex.

The administrator selects Genex. MGX changes the state of Genex, that is, MGX makes it visible or invisible, depending on its last state.

AIM.5.3. In the options view, the administrator selects Protex.

The administrator selects Protex. MGX changes the state of Protex, that is, MGX makes it visible or invisible, depending on its last state.

AIM.5.4. In the options view, the administrator populates the zoo.

The administrator indicates that he would like to add creatures to the zoo. MGX prompts the administrator concerning how he will add those creatures.

AIM.5.4.1. In the options view, the administrator populates the zoo by creating new creatures.

The administrator indicates that he will generate new creatures. MGX opens Genex as in scenario SIM.3.

AIM.5.4.1.1. In Genex, the administrator selects one creature.

See scenario SIM.3.1. Replace actor *student* with administrator.

AIM.5.4.1.2. In Genex, the administrator modifies one creature.

See scenario SIM.3.2. Replace actor *student* with administrator.

AIM.5.4.1.3. In Genex, the administrator saves a (modified) creature.

See scenario SIM.3.3. Replace actor *student* with administrator.

AIM.5.4.1.4. In Genex, the administrator discards a (modified) creature.

See scenario SIM.3.4. Replace actor *student* with administrator.

AIM.5.4.2. In the options view, the administrator populates the zoo by importing existing creatures.

The administrator indicates that he will import creatures from a file. MGX prompts for the file name. The administrator enters a filename. (MGX accepts for this purpose old state files holding creatures saved by students.) MGX loads the creatures in this file into the CSP. The administrator moves creatures from the CSP into the zoo area.

AIM.5.5. In the options view, the administrator depopulates the zoo.

The administrator moves creatures from the zoo into the CSP, and indicates that he wants to discard those creatures. MGX prompts for the name of a file to hold (save) the discarded creatures. The administrator enters a filename, or he indicates that he does not want to save the creatures. If the administrator enters a filename, MGX saves in this file all of the creatures that are in the CSP. MGX clears the CSP.

AIM.6. The administrator closes MGX in integrated mode.

The administrator chooses to exit the application. MGX prompts for the name of the file it will use for saving its state and options. The administrator enters a filename, or indicates that he wants MGX to discard the session. If the administrator enters a filename, MGX saves its state, including options and creatures in the zoo and in cages (VGL). MGX quits.

**Signature**

I approve these administrator integrated mode (AIM) use-case scenarios.

_____________________________________ __________________
Professor Brian White, Customer Today's Date

**Student Stand-Alone Mode**
**SSAM Scenarios**

SSAM.1. The student opens MGX in student stand-alone mode.

The student starts the application. MGX displays a list of choices: *New Session*, *Saved Session* and *Quit.*

SSAM.1.1 *New Session*

The student selects *New Session.* MGX displays a choice of states corresponding to the virtual biological laboratories (VBLs) described above in scenario SIM1.1. Each VBL displays as a stand-alone application, however. Its appearance, features and functionality therefore may differ from those while MGX is in student integrated mode (SIM). There is no zoo holding creatures.

SSAM.1.2 *Saved Session*

If the student has already used MGX and wants to start where she left off, she selects *Saved Session*. MGX prompts for name of file holding the last saved state. The student provides the filename. MGX loads the saved state, and displays a choice of VBLs, as in scenario SSAM.1.1.

SSAM.1.3 *Quit*

The student selects *Quit*. MGX quits.

SSAM.2. In VGL.

The VGL SSAM scenarios are based on and reference the use-cases of Koolar et al. (2002). We have attempted to maintain their numbering scheme. See their use cases for details.

SSAM.2.1. In VGL, the student starts a new problem.

See Koolar et al. (2002), Use Case 1: Starting a new problem.

SSAM.2.2. In VGL, the student opens an existing problem.

See Koolar et al. (2002), Use Case 2: Open

SSAM.2.3. In VGL, the student saves a problem with current name.

See Koolar et al. (2002), Use Case 3: Saving a problem with current name

SSAM.2.4. In VGL, the student saves a problem with a new name.

See Koolar et al. (2002), Use Case 4: Saving a problem with a new name

SSAM.2.5. In VGL, the student closes a problem.

See Koolar et al. (2002), Use Case 5: Closing the problem

SSAM.2.6. In VGL, the student logs off.

See Koolar et al. (2002), Use Case 6: Logging out

SSAM.2.7. In VGL, the student prints the current work to a file.

See Koolar et al. (2002), Use Case 7: Printing the Current Work to a File

SSAM.2.8. In VGL, the student prints the current work to a printer.

See Koolar et al. (2002), Use Case 8: Printing out the Current Work from a printer

SSAM.2.9. In VGL, the student starts crosses two organisms (creatures).

See Koolar et al. (2002), Use Case 9: Crossing Two Organisms

SSAM.2.10. In VGL, the student crosses two creatures.

See Koolar et al. (2002), Use Case 10: Like crossing two organisms

SSAM.2.11. In VGL, the student back crosses two creatures.

See Koolar et al. (2002), Use Case 11: Back crossing two organisms

SSAM.2.12. In VGL, the student reciprocal crosses two creatures.

See Koolar et al. (2002), Use Case 12: Reciprocal crossing two organisms

SSAM.2.13. In VGL, the student test crosses two creatures.

See Koolar et al. (2002), Use Case 13: Test crossing two organisms

SSAM.2.14. In VGL, the student views a test creature.

See Koolar et al. (2002), Use Case 14: Showing the test organism(s)

SSAM.2.15. In VGL, the student sets a test creature.

See Koolar et al. (2002), Use Case 15: Setting the test organism(s)

SSAM.2.16. In VGL, the student abbreviates names.

See Koolar et al. (2002), Use Case 16: Abbreviate Names

SSAM.2.17. In VGL, the student cleans up vials.

See Koolar et al. (2002), Use Case 17: Cleaning up vials

SSAM.2.18. In VGL,...

**Use case 18 of Koolar et al. (2002) is not available.**

SSAM.2.19. In VGL, the student destroys a vial.

See Koolar et al. (2002), Use Case 19: Destroying vial

SSAM.2.20. In VGL, the student activates balloon help.

See Koolar et al. (2002), Use Case 20: Balloon help active

SSAM.2.21. In VGL, the student chooses a help topic.

See Koolar et al. (2002), Use Case 21: Help Topic

SSAM.2.22. In VGL, the student creates a summary chart.

See Koolar et al. (2002), Use Case 22: Creating a Summary Chart

SSAM.2.23. In VGL, the student creates a chi-square analysis.

See Koolar et al. (2002), Use Case 23: Creating a Chi Squared analysis

SSAM.2.24. In VGL, the student creates a cross-matrix analysis.

See Koolar et al. (2002), Use Case 24: Creating a Cross Matrix analysis

SSAM.2.25. In VGL, the student rebuilds a chi-square analysis.

See Koolar et al. (2002), Use Case 25: Rebuilding Chi Squared analysis

SSAM.2.26. In VGL, the student recalculates a chi-square analysis.

See Koolar et al. (2002), Use Case 26: Recalculating Chi Squared analysis

SSAM.2.27. In VGL, the student rebuilds a cross-matrix analysis.

See Koolar et al. (2002), Use Case 27: Rebuilding Cross Matrix analysis

SSAM.2.28. In VGL,...

See ASAM.2.28.

SSAM.2.29. In VGL,...

See ASAM.2.29.

SSAM.2.30. In VGL,...

**Use case 30 of Koolar et al. (2002) is not available.**

SSAM.2.31. In VGL, the student logs on.

See Koolar et al. (2002), Use Case 31: Logging in

SSAM.2.32. In VGL, the student expands back the abbreviated names.

See Koolar et al. (2002), Use Case 32: Expanding back the Abbreviated Names

SSAM.2.33. In VGL, the student deactivates balloon help.

See Koolar et al. (2002), Use Case 33: Inactivating balloon help feature

SSAM.3. In Genex.

The Genex SSAM scenarios are based on and reference the use-cases of Karki et al. (2003). We have attempted to maintain their numbering scheme. See their use cases for details. For numbers ASAM3.1 – ASAM.3.4, see ASAM scenarios.

SSAM.3.4. In Genex, the student checks the input sequence.

See Karki et al. (2003), Use case 04: Check Input Sequence.

SSAM.3.5. In Genex,...*[Does not apply to student or administrator.]*

See Karki et al. (2003), Use case 05: Scanning for a Valid Promoter/Terminator Sequence.

SSAM.3.6. In Genex, the student transcribes a strand of DNA sequence into its pre-mRNA sequence.

See Karki et al. (2003), Use case 06: Transcription.

SSAM.3.7. In Genex, the student separates the introns and the exons from the pre-mRNA sequence.

See Karki et al. (2003), Use case 07: Splicing.

SSAM.3.8. In Genex, the student scans for a valid start and end intron.

See Karki et al. (2003), Use case 08: Scanning for a Valid Start and End Intron.

SSAM.3.9. In Genex, the student translates the mature mRNA into a protein sequence.

See Karki et al. (2003), Use case 09: Translation.

SSAM.3.10. In Genex,...*[Does not apply to student or administrator.]*

See Karki et al. (2003), Use case 10: Testing for Codon.

SSAM.3.11. In Genex, the student highlights the DNA sequence.

See Karki et al. (2003), Use case 11: Highlighting 1, When clicked on DNA Sequence.

SSAM.3.12. In Genex, the student highlights the pre-mRNA sequence

See Karki et al. (2003), Use case 12: Highlighting 2, When clicked on pre-mRNA Sequence.

SSAM.3.13. In Genex, the student highlights the mature mRNA sequence.

See Karki et al. (2003), Use case 13: Highlighting 3, When clicked on mature mRNA Sequence.

SSAM.3.14. In Genex, the student highlights a protein in the protein sequence.

See Karki et al. (2003), Use case 14: Highlighting 4, When clicked on a protein in Protein Sequence.

SSAM.3.15. In Genex, the student resets a DNA sequence to its default value.

See Karki et al. (2003), Use case 15: Reset DNA Sequence.

SSAM.3.16. In Genex, the student edits a DNA nucleotide base in the input sequence.

See Karki et al. (2003), Use case 16: Editing.

SSAM.3.17. In Genex, the student changes the DNA nucleotide base in the input sequence to "T"

See Karki et al. (2003), Use case 17: Changing current base to “T”.

SSAM.3.18. In Genex, the student changes the DNA nucleotide base in the input sequence to "A."

See Karki et al. (2003), Use case 18: Changing current base to “A”.

SSAM.3.19. In Genex, the student changes the DNA nucleotide base in the input sequence to "C."

See Karki et al. (2003), Use case 19: : Changing current base to “C”.

SSAM.3.20. In Genex, the student changes the DNA nucleotide base in the input sequence to "G."

See Karki et al. (2003), Use case 20: : Changing current base to “G”.

SSAM.3.21. In Genex, the student inserts a nucleotide base after the current base in the DNA sequence.

See Karki et al. (2003), Use case 21: Inserting a Nucleotide base after the current base.

SSAM.3.22. In Genex, the student deletes the current nucleotide base in the DNA sequence.

See Karki et al. (2003), Use case 22: Deleting the current Nucleotide base.

SSAM.3.23. In Genex, the student undoes his last action.

See Karki et al. (2003), Use case 23: Undo.

SSAM.3.24. In Genex, the student clears an existing DNA sequence.

See Karki et al. (2003), Use case 24: Clearing a DNA Sequence.

SSAM.3.25. In Genex, the student enters a new DNA sequence.

See Karki et al. (2003), Use case 25: Entering a new DNA Sequence.

SSAM.3.26. In Genex, the student requests help from the Lab Manual.

See Karki et al. (2003), Use case 26: Help - Lab Manual.

SSAM.3.27. In Genex, the student requests help form the User Manual.

See Karki et al. (2003), Use case 27: Help – User Manual.

SSAM.3.28. In Genex, the student requests help concerning an object on screen.

See Karki et al. (2003), Use case 28: Help – Tool Tip.

SSAM.3.29. In Genex, the student requests help by using a question mark (?).

See Karki et al. (2003), Use case 29: Question Mark.

SSAM.3.30. In Genex, the student requests a printer-friendly version of all sequences.

See Karki et al. (2003), Use case 30: Printer Friendly Version.

SSAM.3.31. In Genex, the student folds a protein chain based on the hydrophylic-hydrophobic nature of the amino acids making it up.

See Karki et al. (2003), Use case 31: Folding.

SSAM.3.32. In Genex, the student zooms in and out.

See Karki et al. (2003), Use case 32: Zoom In and Out.

SSAM.3.33. In Genex, the student turns the animation on or off.

See Karki et al. (2003), Use case 33: Animation On/Off.

SSAM.3.34. In Genex, the student sets the default parameters by using Java Swing.

See Karki et al. (2003), Use case 34: Setting the Default parameters using Swings application.

SSAM.3.35. In Genex, the student sets the Default parameters by using a text file.

See Karki et al. (2003), Use case 35: Setting the Default parameters using Text File.

SSAM.4. In Protex.

The student selects Protex. MGX displays the *Protex* VBL interface. which holds a palette of 20 amino acids.

SSAM.4.1. In Protex, the student starts a new polypeptide chain.

The student starts a new polypeptide chain by selecting one amino acid. Protex displays the selected amino acid in two views: a linear view and a folded view. Protex also displays a third view indicating the associated function (color).

SSAM.4.2. In Protex, the student adds one amino acid to an existing polypeptide chain.

The student adds one amino acid to an existing polypeptide chain. Protex adds that amino acid in the linear view and in the folded view. Protex folds the modified polypeptide chain (in the folded view) and updates the associated color in the function view.

SSAM.4.3. In Protex, the student removes one amino acid from an existing polypeptide chain.

The student removes one amino acid from a polypeptide chain. Protex removes that amino acid from the linear view and from the folded view. Protex folds the modified polypeptide chain (in the folded view) and updates the associated color in the function view.

SSAM.5. The student prints the results of her session.

The student selects *Print*. MGX sends an image of the current (active) VBL GUI to the printer.

SSAM.6. The student closes MGX in stand-alone mode.

The student chooses to exit the application. MGX prompts for the name of a file it will use for saving its state. The student enters a filename, or indicates that she wants MGX to discard the session. If she enters a filename, MGX saves its state. MGX quits.

**Signature**

I approve these student stand-alone mode (SSAM) use-case scenarios.

__________________________________ __________________

Professor Brian White, Customer Today's Date

**Administrator Stand-Alone Mode**
**ASAM Scenarios**

ASAM.1. The administrator opens MGX in administrator stand-alone mode.

The administrator starts the application. MGX displays a list of choices: *New Session*, *Saved Session* and *Quit.*

ASAM.1.1 *New Session*

The administrator selects *New Session.*

MGX displays an options view and a choice of states corresponding to the virtual biological laboratories (VBLs) described above in scenario SSAM1.1. There is no zoo holding creatures. In the options view, the administrator is able to turn on and off all VBLs, and to add and remove creatures from the zoo. A VBL that is turned off is not visible.

ASAM.1.2 *Saved Session*

If the administrator has already set up MGX, and wants to implement his custom configuration, he selects *Saved Session*. MGX prompts for name of file holding the last saved state. The administrator provides the filename. MGX loads the saved state, and displays the options view plus VBLs, as in ASAM.1.1.

ASAM.1.3 *Quit*

The administrator selects *Quit*. MGX quits.

ASAM.2. In VGL.

The VGL ASAM scenarios are based on and reference the use-cases of Koolar et al. (2002). We have attempted to maintain their numbering scheme. See their use cases for details. For scenarios numbered ASAM2.1 – ASAM.2.27 and ASAM.2.30 – ASAM.2.32, see SSIM scenarios and replace *student* with administrator.

ASAM.2.38. In VGL, the administrator edits an existing problem.

See Koolar et al. (2002), Use Case 28: Editing An Existing Problem

ASAM.2.29. In VGL, the administrator creates a new problem.

See Koolar et al. (2002), Use Case 29: Creating A New Problem

ASAM.3. In Genex.

The Genex ASAM scenarios are based on and reference the use-cases of Karki et al. (2003). We have attempted to maintain their numbering scheme. See their use cases for details. For numbers ASAM3.4 – ASAM.3.35, see SSAM scenarios and replace *student* with administrator.

ASAM.3.1. In Genex, the administrator specifies the default DNA sequence.

See Karki et al. (2003), Use case 01: Specifying default DNA sequence by Instructor.

ASAM.3.2. In Genex, the administrator specifies the default promoter/terminator sequence.

See Karki et al. (2003), Use case 02: Specifying default Promoter/Terminator sequence by Instructor.

ASAM.3.3. In Genex, the administrator specifies the default start and end intron sequence.

See Karki et al. (2003), Use case 03: Specifying default Start and End Intron sequence by Instructor.

ASAM.4. In Protex.

See scenario SSAM.4. Replace actor *student* with administrator.

ASAM.4.1. In Protex, the administrator starts a new polypeptide chain.

See scenario SSAM.4.1. Replace actor *student* with administrator.

ASAM.4.2. In Protex, the administrator adds one amino acid to an existing polypeptide chain.

See scenario SSAM.4.2. Replace actor *student* with administrator.

ASAM.4.3. In Protex, the administrator removes one amino acid from an existing polypeptide chain.

See scenario SSAM.4.3. Replace actor *student* with administrator.

ASAM.4.4. In Protex, the administrator specifies the weights X-Y-Z of the polypeptide chain folding algorithm.

The administrator specifies the weights X-Y-Z of the polypeptide chain folding algorithm. Protex folds the polypeptide chain, and displays the results (a folded polypeptide chain) in the folded view.

ASAM.4.5. In Protex, the administrator specifies the function (color) for a polypeptide chain.

The administrator specifies the function (color) for an existing polypeptide chain. Protex displays that function (color) in the function view.

ASAM.5. The administrator closes MGX in stand-alone mode.

The administrator chooses to exit the application. MGX prompts for the name of a file it will use for saving its state. The administrator enters a filename, or indicates that he wants MGX to discard the session. If he enters a filename, MGX saves its state. MGX quits.

## Signature

I approve the administrator stand-alone mode (ASAM) use-case scenarios.

___________________________________ ____________________

Professor Brian White, Customer Today's Date