Engineering Requirements Specification
for
Molecular Genetics Explorer (MGX)

Prepared by
the MGX Team:
Ruchi Dubey, Pradeep Kadiyala, Namita Singla and David A. Portman

Team Advisor:
Professor Ethan D. Bolker

Department of Computer Science
University of Massachusetts Boston
100 Morrissey Blvd.
Boston, MA 02125

Document Approved by Client:

Brian T. White, Ph.D.
Assistant Professor
Department of Biology
University of Massachusetts Boston
100 Morrissey Blvd.
Boston, MA 02125

Signature: _____ Date: _____

Executive Summary

The purpose of this document is to establish software development guidelines for the Molecular Genetics Explorer (MGX), a proposed computing application that will help students of General Biology at the University of Massachusetts Boston (UMB) to understand connections among genetics, molecular biology and biochemistry. MGX will bring together three virtual biological laboratories, and will enable biology students to study hypothetical creatures in an interactive computing environment. Described are the proposed project, the software development team, which is made up of four students from the Computer Science Department at UMB, the development process, risks and a project schedule. Formal use cases, which are also listed. describe the proposed functionality of MGX.

The student software developers will use the information presented in this document for product design and functional testing. Although the top-level requirements for MGX are fixed, the low-level requirement details may change during the development process.

Table of Contents

1. Introduction

The Molecular Genetics eXplorer (MGX) is a proposed computing application that will help beginning biology students to understand connections among genetics, molecular biology and biochemistry. MGX will bring together three virtual biological laboratories (VBLs): (1) a virtual genetics lab known as VGL[1], (2) a gene exploration lab known as Genex[2], and (3) a protein exploration lab known as Protex.[3]

MGX will enable students to study hypothetical creatures in an interactive computing environment. In VGL, for example, the students will be able to determine the mechanism of inheritance for one particular trait by crossing creatures and examining their progeny. In Genex, they will be able to interpret the process of transcription and translation of genetic material (*from gene to protein*). In Protex, the students will be able to visualize the unique structure and function of a protein, including protein folding and post-translation modification.

MGX will eventually be used by more than 350 students taking General Biology (course numbers Bio 111 and Bio 112) at the University of Massachusetts Boston (UMB).

2. Project Description

We propose to design and implement MGX as a software application that will run on multiple computing platforms, including Microsoft Windows, Mac OS, Linux and Sun Solaris. We anticipate two general types of MGX users—student and instructor. The instructor, who will serve as the administrator of the software, will be able to configure MGX for class assignments.

1 See http://intro.bio.umb.edu/VGL.
2 See http://intro.bio.umb.edu/GX.
3 See http://www.cs.umb.edu/~eb/folding.

The student will use MGX to complete these assignments. Both types of user will interact with MGX in a graphical computing environment. Shown in Fig. 1 is a Statechart[4] providing a top-level behavioral description of MGX. A curved arrow connector indicates the default entry point, which is VGL (an arbitrary choice). The three VBLs—VGL, Genex, and Protex are shown as separate states of the MGX system. Arrows indicate the transition paths among those states.

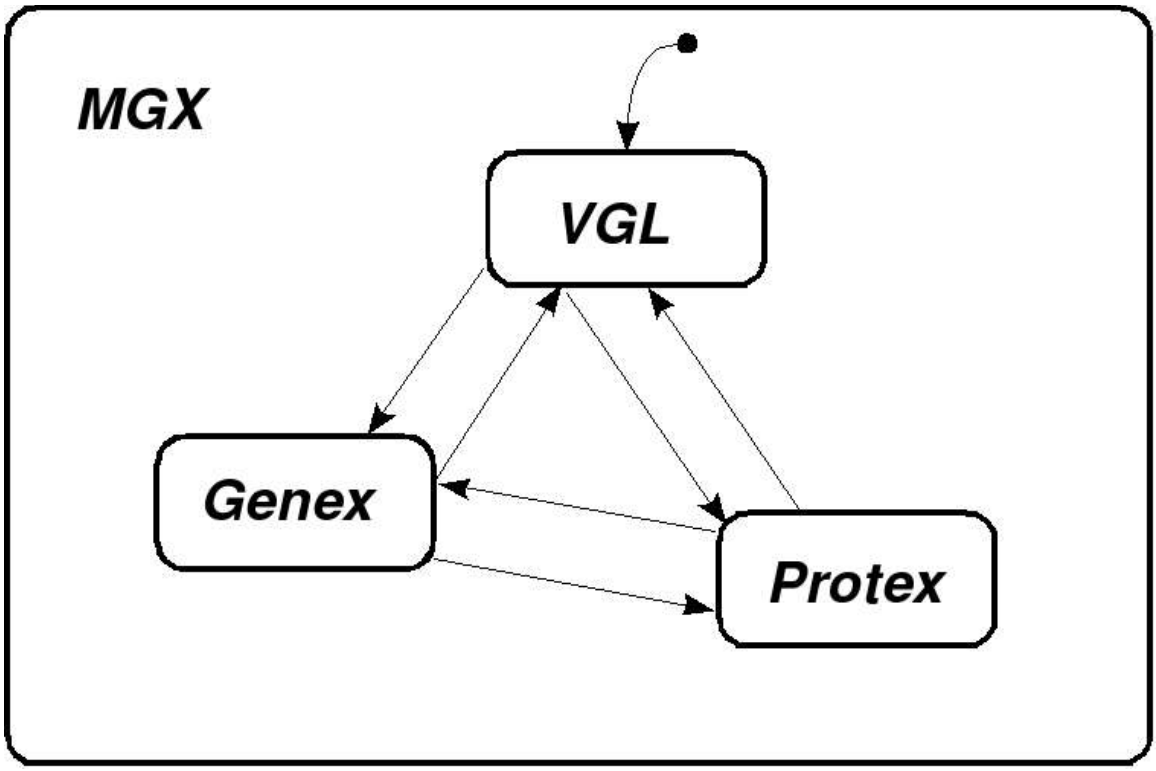

Figure 1. Statechart describing MGX.

The MGX user will be able to access each VBL on its own as a separate application—in *stand-alone mode,* and together as part of one application—in *integrated mode*. In this document we focus on the functionalities of MGX in integrated mode and of Protex in stand-alone mode.

4 The Statechart is a form of design representation devised by Harel (1987).

The stand-alone functionalities of VGL and Genex were discussed separately by VGL Team (2002) and Genex Team (2003), respectively. MGX in integrated mode will display a zoo holding hypothetical creatures. The user will be able to cross (mate) these creatures by accessing VGL in integrated mode. MGX VGL will display the resulting progeny, and will sort those progeny according to gender (male and female) and phenotype (appearance).

To interpret the information held in the DNA sequence of a parent or one of its offspring, the user will access MGX Genex, which will generate two sequences of DNA corresponding to two genes taken from a homologous pair of chromosomes. MGX Genex will scan each DNA sequence for its promoter and terminator, and will graphically display the steps of gene expression, including the transcription, splicing and translation of a gene from a nucleotide sequence into a protein. The user will also be able to modify a creature by editing its DNA sequence.

To visualize protein folding and post-transactional modifications, the user will accesss MGX Protex, which will generate two linear polypeptide sequences taken from a homologous pair of chromosomes. MGX Protex will display the folded structure and color of a protein, depending on the properties of the amino acids present in its polypeptide chains. The user will be able to alter the appearance of a creature by changing the sequence of the amino acids in these chains.

2.1 MGX Team

The MGX development team is made up of four graduate students from the UMB Computer Science (CS) Department. The team advisor is Dr. Ethan D. Bolker, Professor of

Computer Science at UMB. Discussed below is the MGX team outlook regarding organization of the team, communication among team members and the software development process.

2.1.1 Organization

We have decided that there is to be neither a leader nor a hierarchy among the student developers. Every member of the student development team is to be considered equal for the purposes of making project-related decisions and shouldering project responsibilities. The group will assign tasks to individuals on an ad-hoc basis depending on the interests of those individuals as well as on and the needs of the team. For every project task there will be at least one person who is deemed responsible.

The team intends to manage project tasks according to a *pipeline* approach whereby various tasks are partitioned among the team members. Different phases of the software development process will be carried out simultaneously and in parallel by different members of the team, thus ensuring efficiency and steady progress.

2.1.2 Communication

We will communicate publicly via the CS Forums[5] and privately via the CS Department e-mail system. We plan to hold semi-weekly team meetings on every Tuesday and Thursday (at least during the fall semester). At these meetings we will report to each other concerning individual progress and completion of assigned tasks. On Thursday we will construct a to-do list

5 See http://serlhome.cs.umb.edu/CSForums.

for the coming week, which begins on the following Sunday. It is the collective responsibility of the MGX team members to maintain all of the following:

- Web pages.
- Meeting minutes.
- Weekly to-do lists.
- Project deliverables and documentation, including
  - Risk analysis.
  - Use case scenarios.
  - Formal use cases.
  - Requirements specifications (e. g., this document).

2.2 Development Process

We intend to follow Extreme Programming (XP), which is a light-weight object-oriented iterative development process described by Beck (2000). We would like to maintain a simple system design, and to carry out pair programming when practicable. We expect to make frequent, small releases of MGX beginning as soon as possible, and we anticipate making many technical changes. Documents describing problem analysis and requirements gathering are not required for the XP process. Nevertheless, we believe that they will be necessary for MGX, which must be robust and scalable. Therefore we intend to write and maintain such documents.

2.3 Risks

We have identified the following projects risks, which are current as of December 2004.

- We have identified more than 133 formal use cases. Most of these use cases will need to be implemented and/or tested.
  - We must integrate existing applications for VGL, Genex and a protein folding algorithm, including their 68 use cases, which we may need to test.
  - Due to the integrated nature of MGX plus VGL, Genex and Protex, we have written more than 65 additional use cases, most of which we must implement and test.
- The perceived feasibility of our project may raise the expectations of our customer, Professor Brian White, and encourage him to broaden the project scope by insisting on further functionality that he originally requested.
- Professor White needs to specify the coloring and mixing algorithms of the protein structure (for Protex).
- Student developers have to work on the deliverables as well as code and integrate. An improper balance may lead to delays and slipped schedules.
- Student developers cannot estimate with confidence the amount of time required to complete the project tasks, especially those associated with design and implementation.
- We do not yet know how many bugs remain to be found in the existing codes for VGL and Genex.

2.4 Schedule

In this section we summarize completed work and ongoing work, and we discuss future work. A comprehensive schedule of activities is presented in Table 1.

Table 1. Schedule of Activities (in reverse chronological order).

Activity	Assigned To	Date Assigned	Date Due	Status	Remarks
Maintenance Release	Team	?	5/15/05	-	TBA
Poster Presentation	Team	?	5/11/05	-	Spring party with demo of MGX and poster presentation.
Production Release	Team	?	5/1/05	-	TBA
Beta Release	Team	?	4/15/05	-	TBA
Upload the project to SourceForge.net	Namita	?	2/20/05	-	Projects have a longer life at SourceForge.net. where we will have alpha, beta & maintenance releases.
Alpha Release	Team	?	2/15/05	-	TBA
Intermediate Release 0.?	Team	?	1/31/05	-	TBA
Vision, version 2	David	?	1/15/05	-	We will have a different vision when we actually start our work.
Prototype	Namita	?	1/1/05	-	TBA

Activity	Assigned To	Date Assigned	Date Due	Status	Remarks
Test Genex, VGL	Team	12/14/04	?	-	There may be unreported bugs.
Design	Namita	11/22/04	?	-	Initial on MGX integrated.
Project Progress presentation	David	11/22/04	12/9/04	-	Pradeep and David will present.
ERS	Ruchi	11/22/04	12/2/04	✓	Initial draft due.
Use Cases	Pradeep	11/22/04	11/29/04	✓	TBA
Use Case Scenarios	David	10/14/04	11/15/04	✓	Version 4 ready for Clients signature.
Folding Algorithm code	Ruchi, Namita	10/14/04	11/5/04	✓	May have delays due to presentation. Client needs a different front end and energy equation for protein folding.
Venture Capital presentation	David	10/21/04	11/4/04	✓	Presented by Namita and Ruchi.
Web site is available at www.cs.umb.edu/mgx	Pradeep	10/14/04	10/25/04	✓	Web site to be maintained by Pradeep during fall and by David during spring.
CS Forums	David	10/14/04	10/25/04	✓	Team uses forums for communication.
CVS is up and running.	Ruchi	10/14/04	10/25/04	✓	Will be tested by team before Nov 1.

Activity	Assigned To	Date Assigned	Date Due	Status	Remarks
Move Web site docs to SourceForge.net	Namita	10/14/04	10/25/04	✓	Namita will update weekly.
Vision statement	Namita	10/3/04	10/7/04	✓	Slipped by one week as we rescheduled the first client meeting.
Risk analysis	Ruchi	9/30/04	10/3/04	✓	Will be revised as needed.
Initial thoughts	Ruchi	9/30/04	10/3/04	✓	Reflect our strategy of work on the project.
Project Web pages	Pradeep	9/30/04	10/3/04	✓	Web pages reflect progress of our team.

2.4.1 Completed Work/Ongoing Work

We have not yet missed any major deadlines. The schedule posted on the MGX Web site, which we have been updating on a weekly basis, reflects this progress.[6] We have implemented a one-step build for the existing folding code (i. e., Protex) by using Ant, a Java-based tool that enables a full checkout from scratch, compiles every line of code and creates all executables, installation packages and final media.

We have delivered a folding tool to our client, Professor White, so that he can develop the coloring and mixing algorithm for Protex. The final algorithm will fold a polypeptide chain in a

6 At http://www.cs.umb.edu/MGX. We have successfully automated the procedure for updating this Web site.

particular shape, depending on the ionic interactions, hydrogen bonds and hydrophobic index of its various amino acids. We have released version VGL 1.2 on SourceForge.net. We have revised on a periodic basis the documents listing risks, scenarios, use cases and project strategies. We have established concurrent versions control (CVS) for existing codes, Web pages and all project documents.

We are currently working on a schedule for testing the functionality of legacy code and to report defects on a bug tracking system. We are modifying our one-step build script (Ant) and working on another script that will enable us automatically to transfer files, to build and to install at SourceForge.net.

2.4.2 Planned Work

We will first concentrate on the development of MGX in integrated mode. We will then prioritize the VBLs for the development of MGX in stand-alone mode. Throughout the development process, we will make use of the legacy codes written during previous years, and will remove defects (bugs) in an order depending on their severities and priorities. Beginning this winter, we will begin testing VGL and Genex, and will report all defects in our bug tracking system, which is maintained on SourceForge.net, an open-source software development Web site[7].

3. Formal Use Cases

Listed in this section are the formal use cases for MGX. Use cases such as these help

7 See https://sourceforge.net/projects/mgx.

software developers to model system requirements by establishing the nature of the interactions that will take place between the user (called the actor) and the software application (called the system). The use cases for MGX are grouped according to the type of user—i. e., student or administrator--and operating mode—i. e., integrated or stand alone.

UCID: Use case identification number.
Name: Use case name.

Actor: Human user's role.

Pre-Condition(s):
1) Condition 1.
2) Condition 2.
3) Condition 3.

Purpose: The goal of this use case.

Overview: A narrative statement describing what happens in this use case.

Typical course of events:
1) Step 1.
2) Step 2.
3) Step 3.

Post-Condition(s):
1) Condition 1.
2) Condition 2.
3) Condition 3.

Figure 2. Formal Use-Case Template.

There are four general groups of use cases listed in the remainder of this document: (1) use cases for student integrated mode—SIM; (2) use cases for administrator integrated mode—AIM; (3) use cases for student stand-alone mode—SSAM; and (4) use cases for administrator stand-alone mode—ASAM. All of these use cases (for MGX) are formatted according to the

Formal Usc-Case Template, which is shown in Fig. 2. Use cases for VGL and Genex as stand-alone applications are incorporated by reference and are listed in Appendices A and B.

3.1 Use Cases for Student Integrated Mode (SIM)

UCID: SIM.1
Name: Student opens MGX in integrated mode.

Actor: Student.

Pre-Condition: Student installed MGX application.

Purpose: Student starts MGX in integrated mode.

Overview: Student starts the MGX application, which provides a list of choices: New Session, Saved Session, and Quit.

Typical course of events:
1) Student starts the MGX in integrated mode.
2) MGX provides a list of options: New Session, Saved Session, and Quit.

Post-Condition: MGX provides a list of options: New Session, Saved Session, and Quit.

UCID: SIM.1.1
Name: Student selects New Session.

Actor: Student.

Pre-Condition: Student opens MGX in integrated mode.

Purpose: Student starts a new session to work on one, two or three virtual biological laboratories (VBLs) that has Virtual Genetics Lab (VGL), GenExplorer (Genex), Protein Explorer (Protex).

Overview: In New Session, MGX provides a zoo holding creatures and one, two or three VBLs. These VBLs offer the student a choice of labs to work in.

Typical course of events:

1) Student selects new session.
2) MGX displays a zoo holding creatures and one, two or three VBLs that have VGL, Genex, and Protex.
3) Student works on one of the virtual biological laboratories provided by MGX.

Post-Condition: MGX displays one, two or three virtual biological laboratories (VBLs) that can have Virtual Genetics Lab (VGL), GenExplorer (Genex), Protein Explorer (Protex).

UCID: SIM.1.2
Name: Student selects Saved Session.

Actor: Student.

Pre-Condition: Student opens MGX in integrated mode.

Purpose: Student opens MGX with a saved session where she could get to her previous state, which is saved most recently.

Overview: If the student already used MGX, and wants to start where she left off, she selects Saved Session. MGX loads the saved state, and displays one, two or three VBLs as described in (SIM.1.1).

Typical course of events:

1) Student selects Saved Session.
2) MGX prompts the user to open a saved session.
3) Student selects the already saved session.
4) MGX loads the saved session and displays one, two or three VBLs that have VGL, Genex, and Protex.
5) Student works on one of the virtual biological laboratory (VBL) provided by MGX.

Post-Condition: Student enters into one of the virtual biological laboratory (VBL) VGL, Genex, or Protex.

UCID: SIM.1.3
Name: Student Quits from MGX.

Actor: Student.

Pre-Condition: Student opens MGX in integrated mode.

Purpose: Student quits MGX.

Overview: Student quits MGX without starting any session.

Typical course of events:
1) Student selects Quit.
2) MGX quits.

Post-Condition: Student closes the application successfully.

UCID: SIM.2
Name: Student enters VGL.

Actor: Student.

Pre-Condition:
1) Student opens a new session or a saved session.
2) VGL is available.

Purpose: Student works on Virtual Genetics lab.

Overview: The student indicates that she wants to work in VGL. MGX displays a large cage that can hold many creatures and two small cages that can hold just one creature each.

Typical course of events:
1) Student selects VGL mode.
2) MGX displays a large cage and one or two small cages.

Post-Condition: VGL is open for work in SIM.

UCID: SIM.2.1
Name: In VGL mode, student selects one creature.

Actor: Student.

Pre-Condition: Student enters VGL mode.

Purpose: Student selects one creature and mutates to randomly generate a population of creatures. These creatures exhibit a randomly mutated version of genes taken from the original, selected creatures.

Overview: Students selects one creature from a zoo or from large cage and mutates. MGX randomly generates a population of creatures in the large cage. Each of the creatures in this large population exhibits a randomly mutated version of genes taken from the original, selected creatures. The students see the changes in the genotype of that original creature, and observe possible changes in its phenotypes (color).

Typical course of events:

1) Student selects one creature from the zoo or from the large cage.
2) MGX displays the selected creatures in a small cage and shows its genotype
3) Student mutates the selected creature.
4) MGX randomly generates a population of creatures in the large cage.

Post-Condition: MGX randomly generates a population of creatures in the large cage.

UCID: SIM.2.2
Name: In VGL mode, student selects two creatures.

Actor: Student.

Pre-Condition: Student enters VGL mode.

Purpose: Student selects two creatures and cross both of them to get the offspring of the two selected creatures.

Overview: Student selects two creatures from the large cage. When she crosses these two creatures, MGX displays all of the offspring of the two selected creatures.

Typical course of events:

1. Student selects two creatures: both from zoo, one from zoo and one from large cage, or both from the large cage.
2. MGX displays each selected creature in one of the two small cages.
3. Student crosses the two creatures.
4. MGX clears the large cage, and displays all the offspring of the two selected creatures in the large cage.

Post-Condition: MGX displays all the offspring of the two selected creatures in the large cage.

UCID: SIM.2.3
Name: In VGL mode, the student saves new creature(s).

Actor: Student.

Pre-Condition: Student mutates one creature or crosses two creatures.

Purpose: Student saves the new creature(s) for further study.

Overview: Student saves new creature(s) to keep for further study by placing them in the zoo from the large cage.

Typical course of events:
1) Students select the creature from the large cage and place them in the zoo.
2) MGX saves the creature in the zoo.

Post-Condition: MGX saves the new creature in the zoo.

UCID: SIM.2.4
Name: In VGL mode, the student empties the large cage.

Actor: Student.

Pre-Condition: Student mutates one creature or crosses two creatures.

Purpose: Student would like to start by clearing the large cage.

Overview: Student empties the large cage. MGX will prompt the user for the conformation to empty the large cage. Student confirms to remove all the creatures from the large cage. All the creatures are removed from the large cage.

Typical course of events:
1) Student indicates that she wants to empty the large cage.
2) MGX asks for the conformation.
3) Student conforms to empty the large cage.
4) MGX removes all the creatures from the large cage.

Post-Condition: All the creatures in the large cage are removed.

UCID: SIM.2.5
Name: In VGL mode, the student discards a (modified) creature from small cage.

Actor: Student.

Pre-Condition: Student selects one or two creatures.

Purpose: Student discards the selected creature(s) from the zoo.

Overview: Student discards the selected creature(s) from the zoo. MGX asks from the conformation to discard the creature(s) from the small cage.

Typical course of events:
1) Student discards the selected creature(s).
2) MGX asks for the conformation to discard the selected creature(s) in the small cage.
3) Student confirms to discard the creature(s).
4) MGX removes the creature(s) from the small cage.

Post-Condition: MGX discards the creature(s) from the small cage.

UCID: SIM.3
Name: Student enters Genex.

Actor: Student.

Pre-Condition: The Genex VBL is available.

Purpose: The student enters the Genex VBL.

Overview: The student indicates that she wants to work 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.

Typical course of events:
1) The student indicates that she wants to work in Genex.
2) MGX displays a small cage holding a single creature.
3) MGX displays two views for displaying copies of the creatures DNA sequence, the mRNA and protein.

Post-Condition: Genex is open for work in SIM.

UCID: SIM.3.1
Name: In Genex mode, the student selects one creature.

Actor: Student.

Pre-Condition: Student enters Genex.

Purpose: Student selects one creature to see the two separate DNA sequence, the (messenger) RNA and protein associated with the two genes determining one trait from a homologous pair of chromosomes (color).

Overview: Student selects a creature from the zoo. MGX displays the two separate DNA sequences, which are editable, the (messenger) RNA and the protein associated with the two genes determining one trait (color) taken from a homologous pair of chromosomes.

Typical course of events:
1) Student selects a creature from the zoo.
2) MGX displays the creature in a small cage, separate from the zoo.
3) MGX also displays two separate DNA sequences, which are editable, the (messenger) RNA, the protein associated with the two genes determining one trait (color) taken from a homologous pair of chromosomes.

Post-Condition: MGX displays two separate DNA sequences, which are editable, the (messenger) RNA, the protein associated with the two genes determining one trait (color).

UCID: SIM.3.2
Name: In Genex mode, the student modifies one creature.

Actor: Student.

Pre-Condition: Student selects one creature in Genex.

Purpose: Student edits the nucleotides in the DNA sequence to see the changes associated with the messenger RNA, the protein and the appearance of the new creature in the cage.

Overview: Student edits 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.

Typical course of events:
1) Student edits the nucleotides in the DNA sequence associated with either of the genes.
2) MGX modifies the associated messenger RNA, the protein and the appearance (color) of the creature in the cage.

Post-Condition: MGX modifies the associated messenger RNA, the protein and the appearance (color) of the creature in the cage.

UCID: SIM.3.3
Name: In Genex mode, the student saves a (modified) creature.

Actor: Student

Pre-Condition: Student modifies the selected creature.

Purpose: Student saves the modified creature in the zoo.

Overview: Student saves the modified creature that is in the small cage, to the zoo.

Typical course of events:
1) Student save the modified creature.
2) MGX places a copy of the modified creature in the zoo.

Post-Condition: MGX saves the modified creature in the zoo.

UCID: SIM.3.4
Name: In Genex mode, the student discards a (modified) creature.

Actor: Student.

Pre-Condition: Student selects a creature or modifies a creature.

Purpose: Student discards the modified creature from the zoo.

Overview: Student discards the modified creature from the zoo. MGX asks for the conformation to discard the creature from the small cage.

Typical course of events:

1) Student discards the modified creature.
2) MGX asks for the conformation to discard the modified creature in the small cage.
3) Student confirms to discard the creature.
4) MGX removes the creature from the small cage.

Post-Condition: MGX discards the creature from the small cage.

UCID: SIM.4
Name: Student enters Protex.

Actor: Student.

Pre-Condition: The Protex VBL is available.

Purpose: The student enters the Protex VBL.

Overview: The student indicates that she wants to work in Protex. MGX displays a palette holding the 20 common amino acids. There are also two views 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.

Typical course of events:

4) The student indicates that she wants to work in Protex.
5) MGX displays a palette holding the 20 common amino acids.
6) MGX displays two views holding copies of the polypeptide chains plus the functions associated with those chains.

Post-Condition: Protex VBL is open for work in SIM.

UCID: SIM.4.1
Name: In Protex mode, the student selects one creature.

Actor: Student.

Pre-Condition: Student enters Protex mode.

Purpose: Student works on Protex lab to view the two of polypeptide chain (linear and folded) plus the function (color) of the protein associated with the two genes (alleles) of the selected creature.

Overview: In Protex, MGX displays a palette holding the 20 common amino acids to add to existing polypeptide chain or to create a new polypeptide chain**.** The student selects one creature from the zoo. MGX displays two editable copies of polypeptide chain (linear and folded) plus the function (color) of the protein associated with the two genes (alleles) from the homologous pair of chromosome.

Typical course of events:

1) Student selects Protex.
2) MGX displays a palette holding 20 amino acids.
3) Student selects one creature from the zoo.
4) MGX displays the selected creatures in the small cage.
5) MGX also displays editable copies of polypeptide chain (linear and folded) plus the function (color) of the protein associated with the two genes (alleles) from the homologous pair of chromosome.

Post-Condition: MGX displays editable copies of polypeptide chain (linear and folded) plus the function (color) of the protein associated with the two genes (alleles) from the homologous pair of chromosomes.

UCID: SIM.4.2

Name: In Protex mode, the student modifies the selected creature.

Actor: Student.

Pre-Condition: Student selects a creature.

Purpose: Student modifies the selected creature to see the changes associated with the polypeptide chain, the altered function and the appearance of the creature in the small cage.

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

Typical course of events:

1) Student adds/removes one amino acid from the existing polypeptide chain of either gene.
2) MGX folds the polypeptide chain, displays its altered function (color), and modifies the appearance of the selected creature.

Post-Condition: MGX displays the folded polypeptide chain; it's altered function (color), and modifies the appearance of the selected creature.

UCID: SIM.4.3
Name: In Protex mode, the student saves the (modified) creature.

Actor: Student.

Pre-Condition: Student modifies the selected creature.

Purpose: Student saves the modified creature in the zoo.

Overview: Student saves the modified creature by having copy in the zoo.

Typical course of events:
1) Student saves the modified creature.
2) MGX makes a copy of the modified creature in the zoo.

Post-Condition: MGX saves the modified creature in the zoo.

UCID: SIM.4.4
Name: In Protex mode, the student discards the (modified) creature.

Actors: Student.

Pre-Condition: Student selects a creature or modifies a creature.

Purpose: Student discards the modified creature from the small cage.

Overview: Student discards the modified creature. MGX will ask for the conformation to discard the creature from the small cage.

Typical course of events:
1) Student discards the modified creature.
2) MGX asks for the conformation to discard the modified creature in the small cage.
3) Student confirms to discard the creature.
4) MGX removes the creature from the small cage.

Post-Condition: MGX discards the creature from the small cage.

UCID: SIM.5
Name: Student prints the results of her session.

Actors: Student.

Pre-Condition: Student has worked on VGL or Genex or Protex.

Purpose: Student prints the results of her session.

Overview: Student prints the current active VBL to the printer as an image.

Typical course of events:
1) Student selects print.
2) MGX sends an image of the current (active) VBL to the printer.

Post-Condition: The current image in MGX session is printed.

UCID: SIM.6
Name: Student closes MGX in the integrated mode.

Actor: Student.

Pre-Condition: Student opens a New Session or Saved Session.

Purpose: Student closes the MGX session, which is in the integrated mode.

Overview: Student chooses to exit the application. MGX prompts the user to save the session that includes all the creatures in the zoo and in cage. Student can save the session or close MGX without saving.

Typical course of events:
1) Student chooses to exit the application.
2) MGX prompts the user to save the session.
3) Student saves the session with a name or can close without saving.
4) MGX quits.

Post-Condition: MGX quits successfully.

3.2 Use Cases for Administrator Integrated Mode (AIM)

UCID: AIM.1
Name: Administrator opens MGX in integrated mode.

Actor: Administrator.

Pre-Condition: Administrator has installed MGX application.

Purpose: Administrator starts MGX in integrated mode.

Overview: Administrator starts the MGX application, which provides a list of choices: New Session, Saved Session, and Quit.

Typical course of events:
1) Administrator starts the MGX in integrated mode.
2) MGX provides a list of options: New Session, Saved Session, and Quit.

Post-Condition: MGX provides a list of options: New Session, Saved Session, and Quit.

UCID: AIM.1.1
Name: Administrator selects New Session.

Actor: Administrator.

Pre-Condition: Administrator opens MGX in integrated mode.

Purpose: Administrator starts a new session to configure one, two or three virtual biological laboratories (VBLs) that has Virtual Genetics Lab (VGL), GenExplorer (Genex), Protein Explorer (Protex).

Overview: In New Session, MGX displays a zoo holding creatures, an options view and a choice of states corresponding to the virtual biological laboratories (VBLs) described in use case 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.

Typical course of events:
1) Administrator selects new session.
2) MGX displays a zoo holding creatures and one, two or three VBLs that have VGL, Genex, and Protex.

Post-Condition: MGX displays one, two or three virtual biological laboratories (VBLs) that can have Virtual Genetics Lab (VGL), GenExplorer (Genex), Protein Explorer (Protex).

UCID: AIM.1.2
Name: Administrator selects Saved Session.

Actor: Administrator.

Pre-Condition: Administrator opens MGX in integrated mode.

Purpose: Administrator opens MGX with a saved session, previously configured.

Overview: If the Administrator already used MGX, and wants to implement custom configuration, she selects Saved Session. MGX loads the saved state, and displays one, two or three VBLs.

Typical course of events:
1) Administrator selects Saved Session.
2) MGX prompts the user to open a saved session.
3) Administrator selects the already saved session.
4) MGX loads the saved session and displays one, two or three VBLs that have VGL, Genex, and Protex.
5) Administrator configures one of the virtual biological laboratories (VBL) provided by MGX.

Post-Condition: MGX displays one, two or three virtual biological labs (VBL) with custom configuration.

UCID: AIM.1.3
Name: Quit.

Actor: Administrator.

Pre-Condition: Administrator opens MGX in integrated mode.

Purpose: Administrator quits MGX.

Overview: Administrator quits MGX without starting any session.

Typical course of events:
1) Administrator selects Quit.
2) MGX quits.

Post-Condition: Administrator closes the application successfully.

UCID: AIM.2
Name: Administrator enters VGL.

Actor: Administrator.

Pre-Condition:
1) Administrator opens a new session or a saved session.
2) Administrator has enabled VGL in the options view.

Purpose: Administrator will work in VGL.

Overview: The administrator indicates that he wants to work in VGL. MGX displays a large cage that can hold many creatures and two small cages that can hold just one creature each. See scenario SIM.2.

Typical course of events:
1) Administrator indicates that he wants to work in VGL.
2) MGX displays a large cage, and one or two small cages.

Post-Condition: Administrator is able to work in VGL.

(Note: The use cases for VGL in Administrator Integrated Mode are exactly the same as those for VGL in Student Integrated Mode. See use cases SIM.2.1-SIM.2.5, and replace student *with* administrator.*)*

UCID: AIM.3
Name: Administrator enters Genex.

Actor: Administrator.

Pre-Condition: Administrator has enabled Genex in options view.

Purpose: Administrator works in Genex.

Overview: The administrator indicates that he wants to work 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 use case SIM.3.

Typical course of events:

1) Administrator indicates that he wants to work in Genex.
2) MGX displays a small cage.
3) MGX displays two views holding copies of the creatures DNA sequence, mRNA and protein.

Post-Condition: Administrator is able to work in Genex.

(Note: The use cases for Genex in Administrator Integrated Mode are exactly the same as those for Genex in Student Integrated Mode. See use cases SIM.3.1-SIM.3.4, and replace student *with* administrator.*)*

UCID: AIM.4
Name: Administrator enters Protex.

Actor: Administrator

Pre-Condition: Administrator has enabled Protex in the options view.

Purpose: Administrator works in Protex.

Overview: The administrator indicates that he wants to work in Protex. MGX displays a palette holding the 20 common amino acids. There are also two views 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.

Typical course of events:

1) Administrator indicates that he wants to work in Protex.
2) MGX displays a palette holding the 20 common amino acids.
3) MGX displays two views holding *editable* copies of the polypeptide chains (linear and folded) plus the protein's function (color).

Post-Condition: Administrator is able to work in Protex.

(Note: The use cases for Protex in Administrator Integrated Mode are exactly the same as those for Protex in Student Integrated Mode. See use cases SIM.4.1-SIM.4.4, and replace student *with* administrator.*)*

UCID: AIM.5
Name: Administrator enters options view.

Actor: Administrator.

Pre-Condition: Administrator opens a new session or a saved session.

Purpose: Administrator customizes the configuration in the options view.
Overview: 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. Otherwise it is not visible.

Typical course of events:
1) Administrator selects options view.
2) MGX displays a creature supply pool (CSP) and a list of VBLs.

Post-Condition: MGX displays a creature supply pool (CSP) and a list of VBLs.

UCID: AIM.5.1
Name: In the options view, the administrator selects VGL.

Actor: Administrator.

Pre-Condition: Administrator selects options view.

Purpose: Administrator changes the state VGL to visible or invisible mode.

Overview: Administrator selects VGL, and changes its state from visible to invisible or vice versa.

Typical course of events:
1) Administrator selects VGL.
2) MGX alters the state of VGL.

Post-Condition: The visibility of VGL is modified.

UCID: AIM.5.2
Name: In the options view, the administrator selects Genex.

Actor: Administrator.

Pre-Condition: Administrator selects options view.

Purpose: Administrator changes the state Genex to visible or invisible mode.

Overview: Administrator selects Genex, and changes its state from visible to invisible or vice versa.

Typical course of events:
1) Administrator selects Genex mode
2) MGX changes the state of Genex.

Post-Condition: The visibility of Genex is modified.

UCID: AIM.5.3
Name: In the options view, the administrator selects Protex.

Actor: Administrator.

Pre-Condition: Administrator selects options view.

Purpose: Administrator changes the state of Protex.

Overview: Administrator selects Protex, and changes its state from visible to invisible or vice versa.

Typical course of events:
1) Administrator selects Protex.
2) MGX changes the state of Protex.

Post-Condition: The visibility of Protex is modified.

UCID: AIM.5.4
Name: In the options view, the administrator populates the zoo.

Actor: Administrator.

Pre-Condition: Administrator selects the options view**.**

Purpose: Administrator adds creatures into the zoo.

Overview: Administrator populates the zoo with creatures. MGX prompts the administrator concerning how he will add those creatures.

Typical course of events:
1) Administrator indicates that he would like to add creatures to the zoo.
2) MGX prompts the administrator concerning how he will add those creatures.

Post-Condition: MGX prompts the Administrator to populate the creatures by creating new creatures or by importing existing creatures.

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

Actor: Administrator

Pre-Condition: Administrator selects add new creature to the zoo in options view.

Purpose: Administrator populates the zoo with new creatures.

Overview: The administrator indicates that he will generate new creatures. MGX opens Genex to create new creatures by providing the DNA sequence.

Typical course of events:
1) Administrator indicates that he will generate new creatures.
2) MGX opens Genex.

Post-Condition: MGX displays Genex as in use case SIM.3.

UCID: AIM.5.4.1.1
Name: In Genex, the administrator selects one creature.

Actor: Administrator.

Pre-Condition: Administrator selects add new creature to the zoo in options view.

Purpose: Administrator selects one creature to see the two separate DNA sequence, the (messenger) RNA and protein associated with the two genes determining one trait from a homologous pair of chromosomes (color).

Overview: Administrator selects a creature from the zoo. MGX displays the two separate DNA sequence that is editable, the (messenger) RNA and the protein associated with the two genes determining one trait (color) taken from a homologous pair of chromosomes.

Typical course of events:

1) Administrator selects a creature from the zoo.
2) MGX displays the creature in a small cage, separate from the zoo.
3) MGX also displays two separate DNA sequence that is editable, the (messenger) RNA, the protein associated with the two genes determining one trait (color) taken from a homologous pair of chromosomes.

Post-Condition: MGX displays two separate DNA sequences, which are editable, the (messenger) RNA, the protein associated with the two genes determining one trait (color).

UCID: AIM.5.4.1.2
Name: In Genex, the administrator modifies one creature.

Actor: Administrator.

Pre-Condition: Administrator selects one creature in Genex.

Purpose: Administrator edits the nucleotides in the DNA sequence to see the changes associated with the messenger RNA, the protein and the appearance of the new creature in the cage.

Overview: Administrator edits 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.

Typical course of events:

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

Post-Condition: MGX modifies the associated messenger RNA, the protein and the appearance (color) of the creature in the cage.

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

Actor: Administrator.

Pre-Condition: Administrator modifies the selected creature.

Purpose: Administrator saves the modified creature in the zoo.

Overview: Administrator saves the creature, modified and in the small cage, to the zoo.

Typical course of events:
1) Administrator save the modified creature.
2) MGX places a copy of the modified creature in the zoo.

Post-Condition: MGX saves the modified creature in the zoo.

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

Actor: Administrator.

Pre-Condition: Administrator selects a creature or modifies a creature.

Purpose: Administrator discards the modified creature from the zoo.

Overview: Administrator discards the modified creature from the zoo. MGX asks for the conformation to discard the creature from the small cage.

Typical course of events:
1) Administrator discards the modified creature.
2) MGX asks for the conformation to discard the modified creature in the small cage.
3) Administrator confirms to discard the creature.
4) MGX removes the creature from the small cage.

Post-Condition: MGX discards the creature from the small cage.

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

Actor: Administrator.

Pre-Condition: Administrator selects add existing creature to the zoo in the options view.

Purpose: Administrator populates the zoo with existing creatures.

Overview: Administrator imports the creatures from a file. MGX loads the creatures to the creature supply pool (CSP). The administrator moves creatures from the CSP into the zoo area.

Typical course of events:

1) Administrator selects import--creatures from a file.
2) MGX prompts for the file name.
3) Administrator enters the file name.
4) MGX loads the creatures in this file into the CSP.
5) The administrator moves creatures from the CSP into the zoo area.

Post-Condition: The creatures are added into the zoo.

UCID: AIM.5.5
Name: In the options view, the administrator depopulates the zoo.

Actor: Administrator.

Pre-Condition: The administrator populates the zoo.

Purpose: The administrator discards some of the creatures from the zoo.

Overview: The administrator moves the creatures from the zoo to the CSP and indicates that he would like to discard those creatures. MGX prompts the administrator to save the creatures in a file. The administrator can discard the creature by saving them to a file or by deleting permanently**.**

Typical course of events:

1) The administrator moves creatures from the zoo into the CSP, and indicates that he wants to discard those creatures.
2) MGX prompts for the name of a file to hold (save) the discarded creatures.
3) The administrator saves the creatures, or he indicates that he does not want to save the creatures.

Post-Condition: The creatures are discarded from the zoo.

UCID: AIM.6
Name: Administrator closes MGX in integrated mode.

Actor: Administrator.

Pre-Condition: Administrator opens a New Session or Saved Session.

Purpose: Administrator closes the MGX in integrated mode.

Overview: Administrator chooses to exit the application. MGX prompts the user to save the session, including all of the creatures in the zoo and in cages. Administrator can also save the session.

Typical course of events:
1) Administrator chooses to exit the application.
2) MGX prompts the user to save the session.
3) Administrator saves the session with a name or can close without saving.
4) MGX quits.

Post-Condition: MGX quits successfully.

3.3 Use Cases for Student Stand Alone Mode (SSAM)

UCID: SSAM.1
Name: Student opens MGX in student stand-alone mode.

Actor: Student.

Pre-Condition: Students installed MGX application.

Purpose: Student starts MGX in student stand-alone mode.

Overview: Student starts the MGX application. MGX offers a list of choices: New Session, Saved Session, and Quit.

Typical course of events:
1) Student starts the MGX in student stand-alone mode.
2) MGX provides a list of options: New Session, Saved Session, and Quit.

Post-Condition: MGX is running.

UCID: SSAM.1.1
Name: Student selects New Session.

Actor: Student.

Pre-Condition: Student opens MGX in student stand-alone mode.

Purpose: Student starts a new session to work on one, two or three virtual biological laboratories (VBLs)--Virtual Genetics Lab (VGL), GenExplorer (Genex), and/or Protein Explorer (Protex).

Overview: In New Session, MGX provides one, two or three VBLs. These VBLs offer the student a choice of labs to work in. Each VBL displays a stand-alone application. There is no zoo holding creatures.

Typical course of events:
1) Student selects new session.
2) MGX displays a choice of VBLs.
3) Student works in one of the virtual biological laboratories provided by MGX.

Post-Condition: MGX displays one, two or three virtual biological laboratories (VBLs) that have Virtual Genetics Lab (VGL), GenExplorer (Genex), Protein Explorer (Protex).

UCID: SSAM.1.2
Name: Student selects Saved Session.

Actor: Student.

Pre-Condition: Student opens MGX in student stand-alone mode.

Purpose: Student opens MGX with a previously saved session.

Overview: If the student already used MGX, and wants to start where she left off, she selects Saved Session. MGX loads the saved state, and displays one, two or three VBLs. Each VBL functions as a stand-alone application.

Typical course of events:
1) Student selects Saved Session.
2) MGX prompts the user to open a saved session.
3) Student selects the already saved session.
4) MGX loads the saved session and displays one, two or three VBLs that have VGL, Genex, and Protex.
5) Student works on one of the virtual biological laboratory (VBL) provided by MGX.

Post-Condition: Student enters into one of the virtual biological laboratory (VBL) VGL, Genex, or Protex.

UCID: SSAM.1.3
Name: Student Quits from MGX.

Actor: Student.

Pre-Condition: Student opens MGX in integrated mode.

Purpose: Student quits MGX.

Overview: Student quits MGX without starting any session.

Typical course of events:
1) Student selects Quit.
2) MGX quits.

Post-Condition: Student closes the application successfully.

UCID: SSAM.2.
Name: Student enters VGL.

Actor: Student.

Pre-Condition: Student opens a new session or a saved session.

Purpose: Student works in VGL as a stand-alone application.

Overview: The student indicates that she wants to work in VGL. MGX opens VGL as a stand-alone application.

Typical course of events:
1) Student indicates that she wants to work in VGL.
2) MGX opens VGL as a stand-alone application.

Post-Condition: VGL is running in stand-alone mode.

UCID: SSAM.2.1 – SSAM.2.33.

The VGL use cases numbered SSAM.2.1 – SSAM.2.33 are based on the use cases of VGL Team (2002). Their use cases are incorporated here by reference. The ID's and Names of those use cases are listed in Appendix A.

UCID: SSAM.3
Name: Student enters Genex.

Actor: Student.

Pre-Condition: Student opens a new session or a saved session.

Purpose: Student works in Genex as a stand-alone application.

Overview: The student indicates that she wants to work in Genex. MGX opens Genex as a stand-alone application.

Typical course of events:
1) Student indicates that she wants to work in Genex.
2) MGX opens Genex as a stand-alone application.

Post-Condition: Genex is running in stand-alone mode.

UCID: SSAM.3.1 – SSAM.3.35.

The Genex use cases numbered SSAM.3.1 – SSAM.3.35 are based on the use-cases of Genex Team (2003). Their use cases are incorporated here by reference. The ID's and Names of those use cases are listed in Appendix B.

UCID: SSAM.4
Name: Student enters Protex.

Actor: Student

Pre-Condition: Student opens a new session or a saved session.

Purpose: Student works in Protex as a stand-alone application.

Overview: The student indicates that she wants to work in Protex. MGX opens Protex as a stand-alone application. The Protex GUI includes a palette holding 20 amino acids.

Typical course of events:
1) Student indicates that she wants to work in Protex.
2) MGX opens Protex as a stand-alone application.

Post-Condition: Protex is running.

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

Actor: Student.

Pre-Condition: Student enters Protex mode.

Purpose: Student can view the folding and function of the selected polypeptide chain.

Overview: 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).

Typical course of events:
1) The student starts a new polypeptide chain by selecting one amino acid.
2) 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).

Post-Condition: The folding and the associated function of the selected polypeptide chain are displayed.

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

Actor: Student.

Pre-Condition: Student selects a polypeptide chain from the amino acid palette.

Purpose: Student modifies the existing polypeptide chain by adding one amino acid.

Overview: 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.

Typical course of events:

1) Student adds one amino acid to an existing polypeptide chain.
2) 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.

Post-Condition: A modified folded polypeptide chain with its associated color function is displayed.

UCID: SSAM.4.3

Name: In Protex, the student removes one amino acid to an existing polypeptide chain.

Actor: Student.

Pre-Condition: Student selects a polypeptide chain from the amino acid palette.

Purpose: Student modifies the existing polypeptide chain by deleting one amino acid.

Overview: The student removes one amino acid to an existing polypeptide chain. Protex removes 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.

Typical course of events:

1) Student deletes one amino acid to an existing polypeptide chain.
2) Protex removes that amino acid in the linear view and in the folded view.
3) Protex folds the modified polypeptide chain (in the folded view) and updates the associated color in the function view.

Post-Condition: A modified folded polypeptide chain with its associated color function is displayed.

UCID: SSAM.5

Name: Student prints the results of her session.

Actors: Student.

Pre-Condition: Student worked on VGL or Genex or Protex.

Purpose: Student prints the results of her session.

Overview: Student prints the current active VBL to the printer as an image.

Typical course of events:
1) Student selects print.
2) MGX sends an image of the current (active) VBL to the printer.

Post-Condition: The current image in MGX session is printed.

UCID: SSAM.6
Name: Student closes MGX in the stand-alone mode.

Actor: Student.

Pre-Condition: Student opens a New Session or Saved Session.

Purpose: Student closes the MGX session.

Overview: Student chooses to exit the application. MGX prompts the user to save the session, including all the creatures in the zoo and in cages. Student can save the session.

Typical course of events:
1) Student chooses to exit the application.
2) MGX prompts the user to save the session.
3) Student saves the session with a name or can close without saving.
4) MGX quits.

Post-Condition: MGX quits successfully.

3.4 Use Cases for Administrator Stand-Alone Mode (ASAM)

UCID: ASAM.1
Name: Administrator opens MGX in stand-alone mode.

Actor: Administrator.

Pre-Condition: Administrator installed the MGX application.

Purpose: Administrator starts MGX.

Overview: The administrator starts MGX. MGX offers a list of choices: New Session, Saved Session, and Quit.

Typical course of events:
1) Administrator starts MGX.
2) MGX provides a list of options: New Session, Saved Session, and Quit.

Post-Condition: MGX provides a list of options: New Session, Saved Session, and Quit

UCID: ASAM.1.1
Name: Administrator selects New Session

Actor: Administrator

Pre-Condition: Administrator opens MGX in administrator stand-alone mode

Purpose: Administrator starts a new session to configure one, two or three virtual biological laboratories (VBLs) that has Virtual Genetics Lab (VGL), GenExplorer (Genex), Protein Explorer (Protex).

Overview: In New Session, MGX displays an options view and one, two or three VBLs. These VBLs offer the student a choice of labs to work in as described in SSAM.1.1. 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.

Typical course of events:
1) Administrator starts a New Session.
2) MGX displays an options view and one, two or three VBLs.

Post-Condition: MGX displays an options view and one, two or three VBLs.

UCID: ASAM.1.2
Name: Administrator selects Saved Session.

Actor: Administrator.

Pre-Condition: Administrator opens MGX in stand-alone mode.

Purpose: Administrator opens a saved session in MGX.

Overview: If the Administrator already used MGX, and wants to implement custom configuration, she selects Saved Session. MGX loads the saved state, and displays one, two or three VBLs.

Typical course of events:
1) Administrator selects Saved Session.
2) MGX prompts the user to open a saved session.
3) Administrator selects the already saved session.
4) MGX loads the saved session and displays one, two or three VBLs that have VGL, Genex, and Protex.
5) Administrator configures one of the virtual biological laboratories (VBL) provided by MGX.

Post-Condition: MGX displays one, two or three virtual biological labs (VBL) with custom configuration.

UCID: ASAM.1.3
Name: Administrator selects Quit.

Actor: Administrator.

Pre-Condition: Administrator opens MGX in administrator stand-alone mode.

Purpose: Administrator quits MGX.

Overview: Administrator quits MGX without starting any session

Typical course of events:
1) Administrator selects Quit.
2) MGX quits.

Post-Condition: Administrator closes the application successfully.

UCID: ASAM.2
Name: Administrator enters VGL.

Actor: Administrator.

Pre-Condition: Administrator opens a new session or a saved session

Purpose: Administrator opens VGL as a stand-alone application.

Overview: Administrator indicates that he wants to open VGL as a stand-alone application. MGX starts VGL as a stand-alone application.

Typical course of events:
3) Administrator indicates that he wants to open VGL.
4) MGX starts VGL as a stand-alone application.

Post-Condition: VGL is running.

UCID: ASAM.2.1 – ASAM.2.27, ASAM.2.30 – ASAM.2.33.

The use cases for VGL in Administrator Stand-Alone Mode are similar to use cases of Student Stand-Alone Mode. See use cases SSAM.2.1 - SSAM.2.27 and SSAM.2.30 - SSAM.2.33.

UCID: ASAM.2.28 – SSAM.2.29.

The VGL use cases numbered ASAM.2.28 – ASAM.2.29 are based on the use cases of VGL Team (2002). Their use cases are incorporated here by reference. The ID's and Names of those use cases are listed in Appendix A.

UCID: ASAM.3
Name: Administrator enters Genex.

Actor: Administrator.

Pre-Condition: Administrator opens a new session or a saved session.

Purpose: Administrator opens in Genex as a stand-alone application.

Overview: Administrator indicates that he wants to open Genex. MGX starts Genex as a stand-alone application.

Typical course of events:
1) Administrator selects Genex.
2) MGX starts Genex as a stand-alone application.

Post-Condition: Genex is running.

UCID: ASAM.3.1 – SSAM.3.3.

The Genex use cases numbered ASAM.3.1 – ASAM.3.3 are based on the use cases of Genex Team (2002). Their use cases are incorporated here by reference. The ID's and Names of those use cases are listed in Appendix B.

UCID: ASAM.3.4 – ASAM.3.35.

The use cases for Genex in Administrator Stand-Alone Mode are similar to use cases of Student Stand-Alone Mode. See use cases SSAM.3.4 - SSAM.3.35.

UCID: ASAM.4
Name: Administrator enters Protex.

Actor: Administrator.

Pre-Condition: Administrator opens a new session or a saved session.

Purpose: Administrator works in Protex as a stand-alone application.

Overview: The administrator indicates that she wants to work in Protex. MGX opens Protex as a stand-alone application. The Protex GUI includes a palette holding 20 amino acids.

Typical course of events:
1) Administrator indicates that he wants to work in Protex.
2) MGX opens Protex as a stand-alone application.

Post-Condition: Protex is running.

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

Actor: Administrator.

Pre-Condition: Administrator enters Protex.

Purpose: Administrator can view the folding and function of the selected polypeptide chain.

Overview: The administrator 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).

Typical course of events:

1) The administrator starts a new polypeptide chain by selecting one amino acid.
2) 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).

Post-Condition: The folding and the associated function of the selected polypeptide chain are displayed.

UCID: ASAM.4.2

Name: In Protex, the Administrator adds one amino acid to an existing polypeptide chain.

Actor: Administrator.

Pre-Condition: Administrator selects a polypeptide chain from the amino acid palette.

Purpose: Administrator modifies the existing polypeptide chain by adding one amino acid.

Overview: The Administrator 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.

Typical course of events:

1) Administrator adds one amino acid to an existing polypeptide chain.
2) 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.

Post-Condition: A modified folded polypeptide chain with its associated color function is displayed.

UCID: ASAM.4.3
Name: In Protex, the Administrator removes one amino acid to an existing polypeptide chain.

Actor: Administrator.

Pre-Condition: Administrator selects a polypeptide chain from the amino acid palette.

Purpose: Administrator modifies the existing polypeptide chain by deleting one amino acid.

Overview: The Administrator removes one amino acid to an existing polypeptide chain. Protex removes 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.

Typical course of events:
1) Administrator deletes one amino acid to an existing polypeptide chain.
2) Protex removes 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.

Post-Condition: A modified folded polypeptide chain with its associated color function is displayed.

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

Actor: Administrator.

Pre-Condition: Administrator enters Protex.

Purpose: Administrator sets the weight of X-Y-Z of the polypeptide chain-folding algorithm.

Overview: 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.

Typical course of events:

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

Post-Condition: Protex sets the X-Y-Z weights to the administrator specified weights and displays a folded polypeptide chain in the folded view.

UCID: ASAM.4.5

Name: In Protex, the administrator specifies the function (color) for a polypeptide chain.

Actor: Administrator.

Pre-Condition: Administrator enters Protex mode.

Purpose: The administrator specifies the function (color) for an existing polypeptide chain.

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

Typical course of events:

1) Administrator specifies the function (color) for an existing polypeptide chain.
2) Protex displays that function (color) in the function view.

Post-Condition: Protex displays that function (color) in the function view.

UCID: AIM.6

Name: Administrator closes MGX in the administrator stand-alone mode.

Actor: Administrator.

Pre-Condition: Administrator opens a New Session or Saved Session.

Purpose: Administrator closes the MGX session that is in the administrator stand-alone mode.

Overview: Administrator chooses to exit the application. MGX prompts the user to save the session that includes all the creatures in the zoo and in cage. Administrator can save the session or close MGX without saving.

Typical course of events:

1) Administrator chooses to exit the application.
2) MGX prompts the user to save the session.
3) Administrator saves the session with a name or can close without saving.
4) MGX quits.

Post-Condition: MGX quits successfully.

4. Bibliography

Beck, K., 2000. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Boston, 190 pp.

Harel, D., 1987. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, **8**, 231-274.

Genex Team, 2003. *GenExplorer: Software Requirements Specification, Version 1.0.* Department of Computer Science, University of Massachusetts, Boston, 55 pp.

VGL Team, 2002. *Software Requirements Specification for Virtual Genetics Lab, Version 1.0.* Department of Computer Science, University of Massachusetts, Boston, 23 pp.

Appendix A
Use Cases SSAM.2.x and ASAM.2.y
Taken from VGL Team (2002)

UCID: SSAM.2.1
Name: In VGL, the student starts new problem.

See VGL Team (2002), Use Case 1: Starting a new problem.

UCID: SSAM.2.2
Name: In VGL, the student opens existing problem.

See VGL Team (2002), Use Case 2: Open.

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

See VGL Team (2002), Use Case 3: Saving a problem with current name.

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

See VGL Team (2002), Use Case 4: Saving a problem with new name.

UCID: SSAM.2.5
Name: In VGL, the student closes a problem

See VGL Team (2002), Use Case 5: Closing the problem

UCID: SSAM.2.6
Name: In VGL, the student logs off

See VGL Team (2002), Use Case 6: Logging out.

UCID: SSAM.2.7
Name: In VGL, the student prints the current work to a File

See VGL Team (2002), Use Case 7: Printing the current work to a File.

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

See VGL Team (2002), Use Case 8: Printing out the Current Work from a printer.

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

See VGL Team (2002), Use Case 9: Crossing Two Organisms

UCID: SSAM.2.10
Name: In VGL, the student crosses two creatures

See VGL Team (2002), Use Case 10: Like crossing two organisms

UCID: SSAM.2.11
Name: In VGL, the student backcrosses two creatures.

See VGL Team (2002), Use Case 11: Back crossing two organisms.

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

See VGL Team (2002), Use Case 12: Reciprocal crossing two organisms.

UCID: SSAM.2.13
Name: In VGL, the student testcrosses two creatures.

See VGL Team (2002), Use Case 13: Test crossing two organisms.

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

See VGL Team (2002), Use Case 14: Showing the test organism(s).

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

See VGL Team (2002), Use Case 15: Setting the test organism(s).

UCID: SSAM.2.16
Name: In VGL, the student abbreviates names.

See VGL Team (2002), Use Case 16: Abbreviate Names.

UCID: SSAM.2.17
Name: In VGL, the student cleans up vials.

See VGL Team (2002), Use Case 17: Cleaning up vials.

UCID: SSAM.2.18
Name: In VGL, UNKNOWN CASE I.

Use case 18 of VGL Team (2002) is not available.

UCID: SSAM.2.19
Name: In VGL, the student destroys a vial.

See VGL Team (2002), Use Case 19: Destroying vial.

UCID: SSAM.2.20
Name: In VGL, the student activates balloon help.

See VGL Team (2002), Use Case 20: Balloon help active.

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

See VGL Team (2002), Use Case 21: Help Topic.

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

See VGL Team (2002), Use Case 22: Creating a Summary Chart.

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

See VGL Team (2002), Use Case 23: Creating a Chi Squared analysis.

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

See VGL Team (2002), Use Case 24: Creating a Cross Matrix analysis.

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

See VGL Team (2002), Use Case 25: Rebuilding Chi Squared analysis.

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

See VGL Team (2002), Use Case 26: Recalculating Chi Squared analysis.

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

See VGL Team (2002), Use Case 27: Rebuilding Cross Matrix analysis.

UCID: ASAM.2.28
Name: In VGL, the administrator edits an existing problem.

See VGL Team (2002), Use Case 28: Editing An Existing Problem.

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

See VGL Team (2002), Use Case 29: Creating A New Problem.

UCID: SSAM.2.30
Name: In VGL, UNKNOWN USE CASE II.

Use case 30 of VGL Team (2002) is not available.

UCID: SSAM.2.31
Name: In VGL, the student logs on.

See VGL Team (2002), Use Case 31: Logging in.

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

See VGL Team (2002), Use Case 32: Expanding backs the Abbreviated Names.

UCID: SSAM.2.33
Name: In VGL, the student deactivates balloon help.

See VGL Team (2002), Use Case 33: Inactivating balloon help feature.

Appendix B
Use Cases SSAM.3.x and ASAM.3.y
Taken from Genex Team (2003)

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

See Genex Team (2003), Use case 01: Specifying default DNA sequence by Instructor.

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

See Genex Team (2003), Use case 02: Specifying default Promoter/Terminator sequence by Instructor.

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

See Genex Team (2003), Use case 03: Specifying default Start and End Intron sequence by instructor.

UCID: SSAM.3.4
Name: In Genex, the student checks the input sequence.

See Genex Team (2003), Use case 04: Check Input Sequence.

UCID: SSAM.3.5
Name: In Genex,NOT APPLICABLE.

See Genex Team (2003), Use case 05: Scanning for a Valid Promoter/Terminator Sequence.

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

See Genex Team (2003), Use Case 06: Transcription.

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

See Genex Team (2003), Use Case 07: Splicing.

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

See Genex Team (2003), Use case 08: Scanning for a Valid Start and End Intron.

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

See Genex Team (2003), Use case 09: Translation.

UCID: SSAM.3.10
Name: In Genex, NOT APPLICABLE.

See Genex Team (2003), Use case 10: Testing for Codon.

UCID: SSAM.3.11
Name: In Genex, the student highlights the DNA sequence.

See Genex Team (2003), Use case 11: Highlighting 1, when clicked on DNA Sequence.

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

See Genex Team (2003), Use case 12: Highlighting 2, when clicked on pre-mRNA Sequence.

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

See Genex Team (2003), Use case 13: Highlighting 3, when clicked on mature mRNA Sequence.

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

See Genex Team (2003), Use case 14: Highlighting 4, when clicked on a protein in Protein Sequence.

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

See Genex Team (2003), Use case 15: Reset DNA Sequence.

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

See Genex Team (2003), Use case 16: Editing.

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

See Genex Team (2003), Use case 17: Changing current base to “T”.

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

See Genex Team (2003), Use case 18: Changing current base to “A”.

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

See Genex Team (2003), Use case 19: Changing current base to “C”.

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

See Genex Team (2003), Use case 20: Changing current base to “G”.

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

See Genex Team (2003), Use case 21: Inserting a Nucleotide base after the current base.

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

See Genex Team (2003), Use case 22: Deleting the current Nucleotide base.

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

See Genex Team (2003), Use case 23: Undo.

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

See Genex Team (2003), Use case 24: Clearing a DNA Sequence.

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

See Genex Team (2003), Use case 25: Entering a new DNA Sequence.

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

See Genex Team (2003), Use case 26: Help - Lab Manual.

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

See Genex Team (2003), Use case 27: Help – User Manual.

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

See Genex Team (2003), Use case 28: Help – Tool Tip.

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

See Genex Team (2003), Use case 29: Question Mark.

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

See Genex Team (2003), Use case 30: Printer Friendly Version.

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

See Genex Team (2003), Use case 31: Folding.

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

See Genex Team (2003), Use case 32: Zoom In and Out.

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

See Genex Team (2003), Use case 33: Animation On/Off.

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

See Genex Team (2003), Use case 34: Setting the Default parameters using Swings application.

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

See Genex Team (2003), Use case 35: Setting the Default parameters using Text File.

Appendix C
Document Revision Histories

Table C-1: Revision History of ERS for MGX (This Document).

Document Description	Version	Completion Date
ERS draft	1.0	12/02/04
ERS (added table of contents)	1.1	12/05/04
ERS (formatted and added narrative of work done so far)	1.2	12/07/04
ERS (modified table of contents)	1.3	12/08/04

Table C-2. Version Histories of Ancillary and Supporting Documents

Document Description	Version	Completion Date
Project Description	1.0	10/03/04
Vision	1.4	10/07/04
Scenarios/ functional specifications	1.4	11/22/04
Use cases	1.2	11/30/04
Risks	1.3	12/06/04
Schedule	1.3	12/06/04
Project Strategy	1.3	12/06/04