# Online Bioinformatics Toolkit

CYCLE 1 REPORT

Group 9

**Mason Monday, Sadaira Packer, Paul Ryu, Ben Williams, Ansleigh Yancey**

## image_guitar.jpg

**Submitted in partial fulfillment of the requirements for COMP 4710 Senior Design to the Department of Computer Science and Software Engineering, Samuel Ginn College of Engineering, Auburn University**

**Auburn, Alabama**

**March 4, 2019**

TABLE OF CONTENTS

horizontal line

1. System Metaphor...............................…………………………………………………………………………. 3
2. Cycle Intent …………………….…………………………………………………………………………………………… 3
3. User Stories…………………...…………………………………..………………………………………………….. 4-14
4. Design Documentation....…………………………...……………………..…………………………….…… 15-17
5. Testing Plan …………………………..…………………………………………………………………………..…. 18-20
6. Lessons Learned …………...…………………………………………………………………………………………… 21
7. SYSTEM METAPHOR

horizontal line

There is a lack of methods that organize genetic variants for diseases, tissues, and cell types. Advanced computational methods and software are needed to create a database and online web server that can address these issues. Our objective is to develop an interactive, online database that is searchable by disease type and/or tissue and cell line. Users will also be able to request the entry or removal of information in the database.

*By: Sadaira Packer*

horizontal line

1. CYCLE INTENT

horizontal line

Our main goal for Cycle 1 was to implement search functionality based on cell line and/or disease type. This meant intersecting two datasets based on chromosome coordinates to find how cell lines are related to certain diseases, and displaying all relevant information to the user. We also intended to make small improvements to the UI.

*By: Benjamin Williams*

3. USER STORIES

horizontal line

|  |  |
| --- | --- |
| **User Story** | Add Entry Request (Text Box) |
| **Description** | The actor sends a request to add an entry to the database using the text box. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor chooses the “send add request” option. 3. The actor enters information on a genetic mutation or genome signal into the provided text box. 4. The actor’s request to add the entry to the database is sent to the administrator. |
| **Extensions** | 1a. Failure to send entry request:   * The text box entry was empty, and no request was sent to the administrator. |
| **Post-Conditions** | 1a. The Entry is added to the database:   * The administrator reviewed the actor’s request and permitted its addition to the existing database. * The requesting actor is notified.   1b. The Entry is not added to the database:   * The administrator reviewed the actor’s request and did not permit its addition to the existing database. * The requesting actor is notified. |

|  |  |
| --- | --- |
| **User Story** | Add Entry Request (Text File) |
| **Description** | The actor sends a request to add an entry to the database by uploading a .txt file. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor chooses the “send add request” option. 3. The actor uploads a file with information on a genetic mutation or genome signal. 4. The actor’s request to add the entry to the database is sent to the administrator. |
| **Extensions** | 1a. Failure to send entry request:   * No file was uploaded and no request was sent to the administrator. * The file uploaded was not in .txt format. |
| **Post-Conditions** | 1a. The Entry is added to the database:   * The administrator reviewed the actor’s request and permitted its addition to the existing database. * The requesting actor is notified.   1b. The Entry is not added to the database:   * The administrator reviewed the actor’s request and did not permit its addition to the existing database. * The requesting actor is notified. |

|  |  |
| --- | --- |
| **User Story** | Add Entry Via Text Box (Administrator) |
| **Description** | The actor adds an entry to the database using the text box. |
| **Actor** | Administrator |
| **Precondition** | The actor is connected to the internet and has navigated to the website and is an administrator of the site. |
| **Basic Flow of Events** | 1. The actor navigates to the admin webpage. 2. The actor chooses the “add entry” option. 3. The actor enters information on a genetic mutation or genome signal into the provided text box. |
| **Extensions** | 1a. Failure to add entry:   * The text box entry was empty, and no entry was added. |
| **Post-Conditions** | The Entry is added to the database |

|  |  |
| --- | --- |
| **User Story** | Add Entry Via File (Administrator) |
| **Description** | The actor adds an entry to the database by uploading a file. |
| **Actor** | Administrator |
| **Precondition** | 1a. The actor is connected to the internet and has navigated to the website.  1b. The actor is an administrator of the site. |
| **Basic Flow of Events** | 1. The actor navigates to the admin webpage. 2. The actor chooses the “add entry” option. 3. The actor uploads a file with the information on a genetic mutation or genome signal. |
| **Extensions** | 1a. Failure to add entry:   * The file uploaded was not in the .txt format, and no entry was added. |
| **Post-Conditions** | The Entry is added to the database |

|  |  |
| --- | --- |
| **User Story** | Delete Entry Request |
| **Description** | The actor sends a request to delete an existing entry from the database. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor finds the entry that he/she would like to be removed. 3. The actor chooses the “send delete request” option. 4. The actor enters his/her reasoning for deletion (optional). 5. The actor’s request to delete the existing entry is sent to the administrator. |
| **Extensions** |  |
| **Post-Conditions** | 1a. The Entry is deleted from the database:   * The administrator reviewed the actor’s request and permitted its removal from the database. * The requesting actor is notified   1b. The Entry is not deleted from the database:   * The administrator reviewed the actor’s request and did not permit its removal from the database. * The requesting actor is notified. |

|  |  |
| --- | --- |
| **User Story** | Delete Entry (Administrator) |
| **Description** | The actor deletes an existing entry from the database. |
| **Actor** | Administrator |
| **Precondition** | 1a. The actor is connected to the internet and has navigated to the website.  1b. The actor is an administrator of the site. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor finds the entry that he/she would like to be removed. 3. The actor chooses the “delete entry” option. |
| **Extensions** |  |
| **Post-Conditions** | 1a. The Entry is deleted from the database:   * The administrator removed the desired entry from the database. |

|  |  |
| --- | --- |
| **User Story** | Edit Entry Request |
| **Description** | The actor sends a request to edit an existing entry from the database. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor finds the entry that he/she would like to edit. 3. The actor chooses the “send edit request” option. 4. The actor enters his/her requested changes and his/her reasoning for said changes in the provided text box. 5. The actor’s request for the existing entry to be edited is sent to the administrator. |
| **Extensions** | 1a. Failure to add entry:   * The text box entry was empty, and no edit request was sent. |
| **Post-Conditions** | 1a. The Existing Entry is Edited:   * The administrator reviewed the actor’s request and permitted the suggested changes. The entry is edited on the database. * The requesting actor is notified.   1b. The Existing Entry is Not Edited:   * The administrator reviewed the actor’s request and denied the suggested changes. The entry remains the same on the database. * The requesting actor is notified. |

|  |  |
| --- | --- |
| **User Story** | Edit Entry (Administrator) |
| **Description** | The actor edits an existing database entry. |
| **Actor** | Administrator |
| **Precondition** | 1a. The actor is connected to the internet and has navigated to the website.  1b. The actor is an administrator of the site. |
| **Basic Flow of Events** | 1. The actor navigates to the admin webpage. 2. The actor finds the entry he/she desires to edit. 3. The actor chooses the “edit entry” option. 4. The actor alters the existing entry’s information using the text box. |
| **Extensions** | 1a. Failure to add entry:   * The text box entry was empty, and no edit was submitted. |
| **Post-Conditions** | 1. The Entry is edited:  * The Administrator’s changes to the existing database entry are saved. |

|  |  |
| --- | --- |
| **User Story** | Search for Database Entry (Cell Line) |
| **Description** | The actor searches the database for the desired genetic mutation and/or human genome signal. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor completes the search entry based on predefined search options in the “cell line” drop down box. 3. The actor chooses the “search database” option. |
| **Extensions** | 1a. Failure to search for entry:   * The search options were incorrectly completed. * The search options were incomplete. |
| **Post-Conditions** | 1a. The Existing Entry is Found:   * The search options for the cell line yields one or more results. * The information is displayed for the actor.   1b. The Existing Entry is Not Found:   * The search options chosen by the actor do not yield a result. * There is no existing entry in the database. |

|  |  |
| --- | --- |
| **User Story** | Search for Database Entry (disease type) |
| **Description** | The actor searches the database for the desired genetic mutation and/or human genome signal. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor completes the search entry based on predefined search options in the “disease type” drop down box. 3. The actor chooses the “search database” option. |
| **Extensions** | 1a. Failure to search for entry:   * The search options were incorrectly completed. * The search options were incomplete. |
| **Post-Conditions** | 1a. The Existing Entry is Found:   * The search options for the disease type yields one or more results. * The information is displayed for the actor.   1b. The Existing Entry is Not Found:   * The search options chosen by the actor do not yield a result. * There is no existing entry in the database. |

|  |  |
| --- | --- |
| **User Story** | Search for Database Entry (cell line & disease type) |
| **Description** | The actor searches the database for the desired genetic mutation and/or human genome signal. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor completes the search entry based on predefined search options in the “cell line” and “disease type” drop down boxes. 3. The actor chooses the “search database” option. |
| **Extensions** | 1a. Failure to search for entry:   * The search options were incorrectly completed. * The search options were incomplete. |
| **Post-Conditions** | 1a. The Existing Entry is Found:   * The search options for the cell line and disease type yields one or more results. * The information is displayed for the actor.   1b. The Existing Entry is Not Found:   * The search options for the cell line and disease type do not yield a result. * There is no existing entry in the database. |

|  |  |
| --- | --- |
| **User Story** | Download Search Results |
| **Description** | The actor downloads information about a genetic mutation and/or human genome signal. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor completes the search entry based on predefined search options in the “cell line” and/or “disease type” drop down boxes. 3. The actor chooses the “search database” option. 4. The actor chooses the download type (excel, text, word document). |
| **Extensions** |  |
| **Post-Conditions** | 1. The Existing Entry is Found:   * The actor saves the file to his/her personal device. |

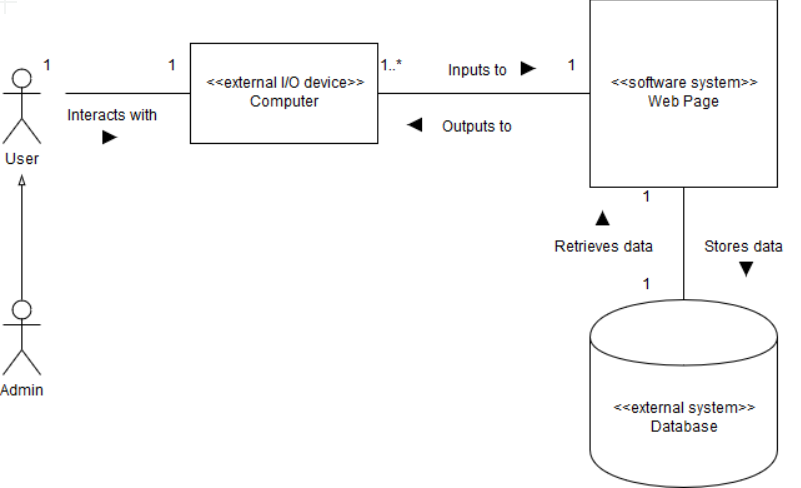
*By: Ansleigh Yancey*

4. DESIGN DOCUMENTATION

horizontal line

**Architecture:**

The human genome data is stored on a database developed in R. This database is linked to a webpage developed with R and hosted on the web server Shiny that our users will interact with. Users will input the values for the specific piece of data they wish to access into the webpage, which will then search our database for the data. Once found, the webpage retrieves the data and displays it to the user. Also, users may request changes to the database, such as adding a new entry. Upon submitting a change, the webpage holds the change until it is approved, after which the webpage will edit the desired entry.



**Structure:**

The database application will be created using the R Shiny framework, and will be hosted using shinyapps.io. The data will be contained within a table containing seven columns and a row for each entry, constructed with the following attributes:

* ID: the unique value assigned to each entry in the database
* chrom: the name of the chromosome or scaffold
* chromStart: the starting position of the chromosome or scaffold
* chromEnd: the ending position of the chromosome or scaffold
* name: the name of the genetic variant
* score: the score given to the genetic variant
* blocks: the specific block the genetic variant resides on

Users may submit requests to add and delete entries. Upon approval by an administrative party, the table will be updated. The majority of the UI and server, as well as methods to CRUD (create, read, update, or destroy entries) will be hosted in a file called “main.R” . The table will, by default, be populated with just over 4.3 million entries from a provided .bed file. As development will be done in the RStudio IDE using Git as the primary file sharing service, the system will also contain default RStudio project files and a Git repository. Documentation on the table entries can be found at <https://genome.ucsc.edu/FAQ/FAQformat.html#format1>.

*By: Benjamin Williams*

USER INTERFACE

horizontal line

* INPUTS

*Search*: The inputs are the drop down boxes for disease type and tissue and cell lines. These inputs determine the resulting genome signal information. Selecting the format type is an input for downloading results.

*Submission:* The inputs are text boxes for data submission and removal suggestions and file upload capabilities for data submission suggestions.

*Administrator*: The inputs are text boxes entries for the chromosome, start, end, name, score, and strand of the genome signal that the administrator has the authority to add or delete from the database.

* OUTPUTS

*Search*: the output is the list of genome signals that correlate with the disease type and cell line options chosen by the user.

*Submission:* There are no outputs for the delete and submission requests.

*Administrator:* For the administrator’s portal, the output for deletion and addition are direct changes to the database.

*By: Ansleigh Yancey*

5. TESTING PLAN

horizontal line

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function | Required Actions | Expected Results | Comments | ✔**/X** |
| Add Entry Request  (Text Box) | Navigate to the webpage. Choose the “send add request” option. Enter information on a genetic mutation or genome signal into the provided text box. The request to add the entry to the database is sent to the administrator. | Entry request sent to administrator |  |  |
| Add Entry Request  (Text File) | Navigate to the webpage.  Chooses the “send add request” option.  Upload a file with information on a genetic mutation or genome signal.  Request to add the entry to the database is sent to the administrator. | Entry request sent to administrator |  |  |
| Add Entry via Textbox (Admin) | Navigates to the admin webpage. Chooses the “add entry” option. Enter information on a genetic mutation or genome signal into the provided text box. | Entry added to database |  |  |
| Add Entry via File (Admin) | Navigate to the admin webpage.  Choose the “add entry” option.  Upload a file with the information on a genetic mutation or genome signal. | Entry added to database |  |  |
| Delete Entry Request | Navigate to the webpage.Find the entry that you would like to be removed. Choose the “send delete request” option. Enter reasoning for deletion (optional). The request to delete the existing entry is sent to the administrator. | Entry deleted from database |  |  |
| Delete Entry (Admin) | Navigate to the webpage.Find the entry that you would like to be removed. Choose the “delete entry” option. | Entry deleted from database |  |  |
| Edit Entry Request | Navigate to the webpage. Find the entry that you would like to edit. Choose the “send edit request” option. Enter the requested changes and reasoning for said changes in the provided text box. The request for the existing entry to be edited is sent to the administrator | Request sent to administrator |  |  |
| Edit Entry (Admin) | Navigate to the admin webpage.  Find the entry that they desire to edit.  Choose the “edit entry” option.  Alter the existing entry information using the text box. | Change is made to database entry |  |  |
| Search for Database Entry (cell line) | Navigate to the webpage. Complete the search entry based on predefined search options in the “cell line” drop down box. Choose the “search database” option. | The results matching the search criteria are displayed |  |  |
| Search for Database Entry (disease type) | Navigate to the webpage. Complete the search entry based on predefined search options in the “disease type” drop down box. Choose the “search database” option. | The results matching the search criteria are displayed |  |  |
| Search for Database Entry (cell line & disease type) | Navigate to the webpage. Complete the search entry based on predefined search options in the “cell line” and “disease type” drop down boxes. Choose the “search database” option. | The results matching the search criteria are displayed |  |  |
| Download Search Results | Navigate to the webpage.  Complete the search entry based on predefined search options in the “cell line” and/or “disease type” drop down boxes. Choose the “search database” option. Choose the download type (excel, text, word document). | A file is downloaded containing the results matching the search criteria |  |  |

*By: Sadaira Packer and Paul Ryu*

6. LESSONS LEARNED

horizontal line

**Ansleigh Yancey:** With cycle 1, I learned more about Shiny and its UI capabilities. There are different fluid page types within Shiny and R Studio that allow for different layouts of the web application. I also learned more about making subsets of our current dataset based on search criteria. This cycle has been challenging, and understanding the data we are working with is not a simple task. For me, the take away from this cycle is the importance of sponsor communication and an increased understanding of user interface application development.

**Mason Monday:** I spent most of my time this cycle researching and testing our front-end search functionality. During testing, because of a lack of info from our sponsor, much of what I had to test was more theoretical in scope. This meant that I had to discover how to test certain functionality without the necessary data available to me. Also, I learned just how slowed down a project can get if useful data is not present, such as the hiccup in production we experienced on our back-end side.

**Benjamin Williams:** During the Cycle 1 phase, I learned how to filter data frames based on specific variables and their values. However, many of my efforts were not very useful since the end goal was not as clearly defined as it should have been. I learned that defining and fully understanding an objective is necessary before diving headfirst into implementation. I also learned to communicate more with team members who are more experienced in certain areas, since many hours were unnecessarily spent debugging problems that had simple solutions.

**Paul Ryu:** I learned many things about the R language and the package that came along with it - Shiny. I always thought that R was just another boring language for statistics and computing but it has turned out to be so much more and can even make graphics that look more advanced than the outputs of its modern language counterparts. During the Cycle 1 phase, I learned that there’s sometimes nothing a group can do if the client does not give any assistance.

**Sadaira Packer:** Looking at other cell line databases wasn’t very helpful in helping us combine the datasets. There being multiple cell lines for some of the dataset entries will complicate the search algorithm. If there was only a single cell line per entry we could display results by searching the cell line column for matching variables. But because some variables in the cell line column have multiple cell lines, this won’t display all the proper results.