**GONZAGA UNIVERSITY**

**School of Engineering and Applied Science**

**Center for Engineering Design and Entrepreneurship**

**PROJECT PLAN**

**10/12/18**

**Interactive Graphical Tools for Learning Biochemistry**

**Prepared by:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Kaylee Moniz Kayla Larson

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Garett Palm Stephen Joyce

**Reviewed by:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Rob Bryant Dr. Jeff Watson

Faculty Project Advisor Project Sponsor/Liaison

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mike Mudge

Design Advisory Board Member

**1 Project Overview**

**1.1 Project Summary**

The overall problem this project is addressing is the lack of quality interactive tools present in the biochemistry teaching community. Specifically, the tools used to teach students about different biochemical pathways can be difficult to comprehend and are unclear. Additionally, the standard biochemical study materials are non-interactive and rarely visual, which may make learning more difficult for those who learn best visually.

We will create an interactive tool that allows the user to create models of different intracellular pathways and show the chemical reactions in a graphical interface. This product will provide a platform on which educators can build interactive simulations of various pathways to improve learning and data retention for their students. To achieve student data retention and learning, the project will have a “click-to-add” interface, that will allow a user to click on a section of their model and add a new module. Inside that module, a user will be able to input formulas and values for different variables and visualize the impact on the cell from such formulas. This ability to customize a model will allow the educator to create a simulation of pathways such as glycolysis, the process of breaking down glucose, and gluconeogenesis, the process of creating glucose, as well as other pathways that take place in a cell. Once a model has been created, the students will be able to access and modify certain modules of the model within the boundaries set by the educator. This will provide an interactive tool that the students can use to more effectively learn about various biochemical pathways.

**1.2 Project Objectives**

The objective of the website is to provide a more visual way of learning about the intracellular biochemistry pathways. We will do this in the hopes that visual learners will have an alternative way of studying these pathways that may be easier to understand and comprehend. We are hoping to revolutionize the simple arrow and box animation of biochemistry and instead create a complex but intuitive flow process of pathways. Another objective is that this website improves a student’s understanding of the structure of these pathways, as well as how the pathways work and how variables change. This will hopefully occur since a user will have the ability to see the flow of molecules in the pathway and interact with the model to test their understanding. For example, if a student was not sure if they understood what caused the creation of glucose in the cell, they could run the glycolysis model and verify whether or not they were correct. The final objective of this website is to provide professors with a customizable tool to model pathways that will decrease the difficulty of teaching these pathways to students. This will hopefully occur since a professor would have the ability to customize their models to the level of understanding their students have and by giving a professor the ability to share their model with their students.

**1.3 Project Stakeholders**

**Developers:** Stephen Joyce, Kayla Larson, Kaylee Moniz, Garett Palm

The developers will be designing the project plan, collecting requirements from the sponsor and creating, coding, and testing the final product. The developers will all directly benefit from the completion of the project and gain insight and experience that will become useful for future jobs. This is because the project will force the developers to learn new software and skills, such as MySQL, Django, Visual Studio, HTML5, JavaScript, and database management. Which are all critical for the development of the product. Additionally, the project will force the developers to learn skills on their own which will be very beneficial when entering the workforce.

**Sponsor:** Dr. Jeff Watson

The developers will be consulting Dr. Watson for all biochemistry related questions, as he is the domain expert for this project. Dr. Watson has an overall vision of the project and in working with the developers will supply feedback during the development process in compliance with the business objectives. He will directly benefit from the completion of the project as he will be able to use the application to help teach his students.

**Advisor:** Rob Bryant

Bryant is the facilitating advisor and will be monitoring the development team every week to make sure the team is on track with the project plan. He will be listening into a developer lead discussion every Monday at 3:00 pm, giving advice and going over the completed work that was done in the past week.

**DAB Member:** Mike Mudge, Dan Lenz

Mike Mudge and Dan Lenz are potential resources the developers can use to gain insight into the process of building an application for stakeholders. Dan Lenz, the Application Delivery Manager for Avista, has an understanding of what it takes to go through the entire development process successfully. Additionally has 10+ years of industry experience and is available by email or phone to the developers if they have any questions. They will both be sitting in on all the developer's presentations throughout the year to judge, ask questions and give advice.

**Target user:** Gonzaga Biochemistry Students/Faculty

Biochemistry students will be the main target user and source of feedback in the creation of the application. They are looking for a way to more efficiently study biochemistry by visualizing the flow through pathways that happens within the cell. Faculty will be interested in creating models that students can interactively use. The developers and sponsor will be working to tailor the application and user experience to best fit the needs of these students and faculty. Students will directly benefit from the creation of the biochemistry modeling website, as the website will provide a study tool that will likely be a better option for those who learn visually. Faculty will directly benefit as they will be able to teach advanced concepts to students with more ease.

**1.4 Project Deliverables**

The finalized product and primary deliverable will be a website. This website will enable users to login to their account and create biochemistry models in their native browser. Each model can be saved and then viewed at a later time. Each user will also be able to interact with each model by altering variables and altering sprites that represent the pathway controls. By manipulating these, they will see the subsequent results. This website will be deployed, managed and hosted on the local Gonzaga servers. The second deliverable will be software documentation. In the documentation, we will explain how the website was developed, how to set up the development environment, how each method in the code works, and suggest possible routes for future improvement. The document will also include a user guide so any faculty will be able to show his/her students how to use the application. This documentation will be saved in a pdf, uploaded to our GitHub repository and sent to our advisor and sponsor. The last deliverable will be test software. We will be using Jest, a software testing framework to run simple unit tests on different components of the code and display whether or not the expected result was received via print statements. This software will be uploaded/saved on our GitHub.

**1.5 Project Scope**

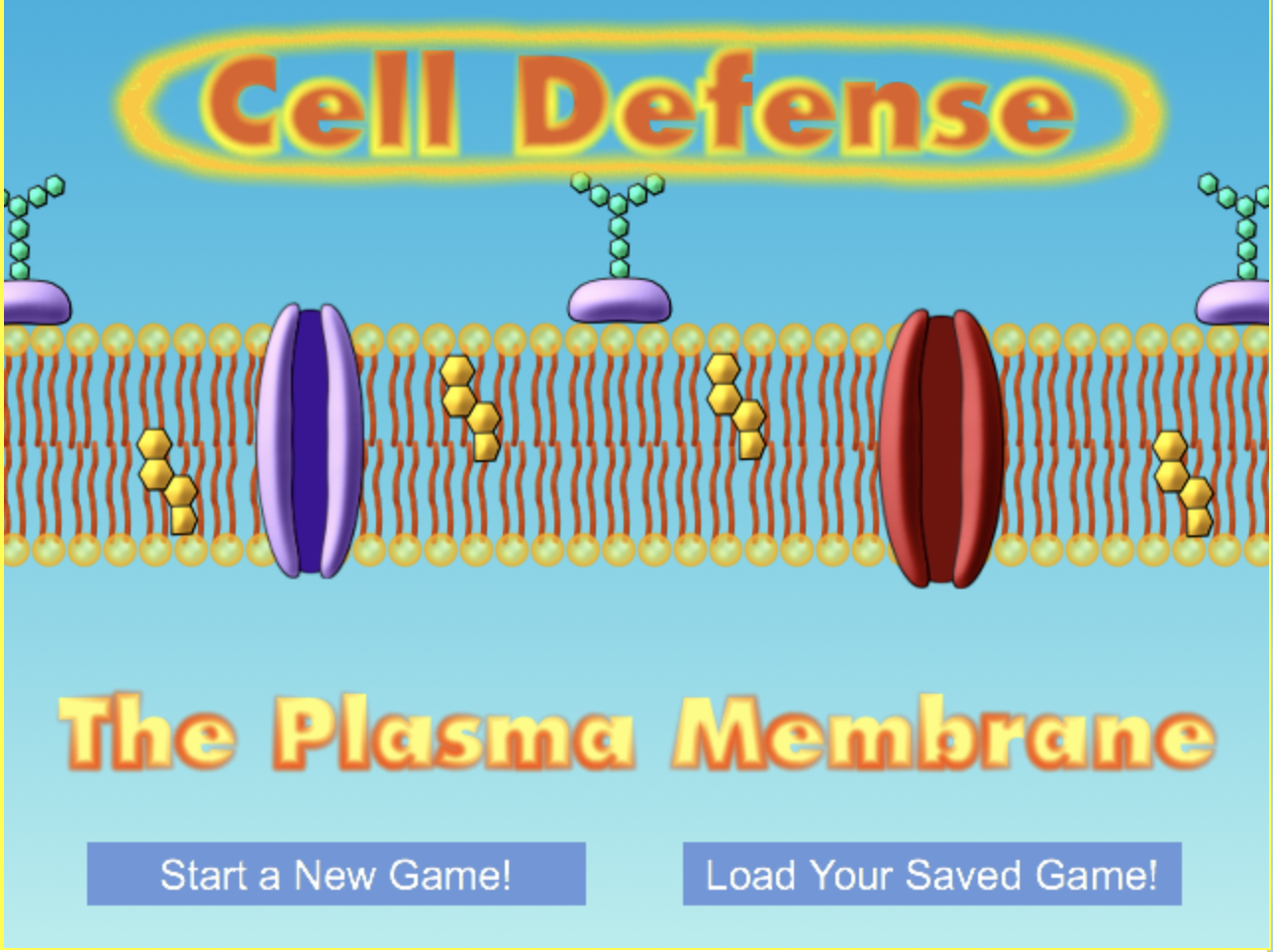
The architecture we will be using is composed of three tiers, a UI, a MySQL database and Django web framework. The Django and MySQL software is out of scope, as indicated by the dark blue color of the MySQL/Django box in figure 1. Django and MySQL are both established services that will be used to host and save the data from our website. The mathematical functions are also established, however, these functions will need to be customized and fitted to each model and are currently in scope, as indicated by the light blue color of the box labeled “Biochemistry Pathways” in figure 1. The following tasks are also currently in the scope of our project, developing a database schema, allowing for a standard user, and designing a UI to support creating a model, creating an account, logging in to an account, accessing saved models, and editing new and existing models. We will need to implement database schemas to maintain all of the necessary data, such as user information and each model created by the user. We will have to enable the creation of administrative users that will allow professors special viewing abilities of all the student’s models. We will have to create a user interface where modules can be added to the biochemical pathways as well as saved and viewed at a later time. The user interface is the most important in scope aspect of the project, we will be creating many of the visuals from scratch and the interface will be completely custom.

**Figure 1: Context diagram**

**1.6 Related Work**

The Bioman website[[1]](#footnote-0) visually represents intracellular processes in a way that is similar to what we want to achieve. The Cell Defense game will look similar to a model created in our app in that it will have a similar rendering of a cell wall which can be seen in the image below, Figure 2. Additionally, in the Cell Defense Game, a user has the ability to drag around objects which subsequently change what occurs in the system. Our app will incorporate a similar feature, allowing the user to add a module to their model and alter variables, however, they will likely not be able to drag-and-drop, instead they will click a plus button in the design grid and from there will be able to add to the pathway. Our website will also differ from the Cell Defense game by clearly indicating the changes that occur when a variable is changed. Each variable in the pathway will be dependent upon other variables and by result affect the entire pathway. Our website will incorporate a more accurate representation of the flow through intracellular pathways and include the mathematical formulas for how much of an enzyme will flow through the pathway as well as display definitions and descriptions of objects when clicked.

**Figure 2:** Cell Defense Game

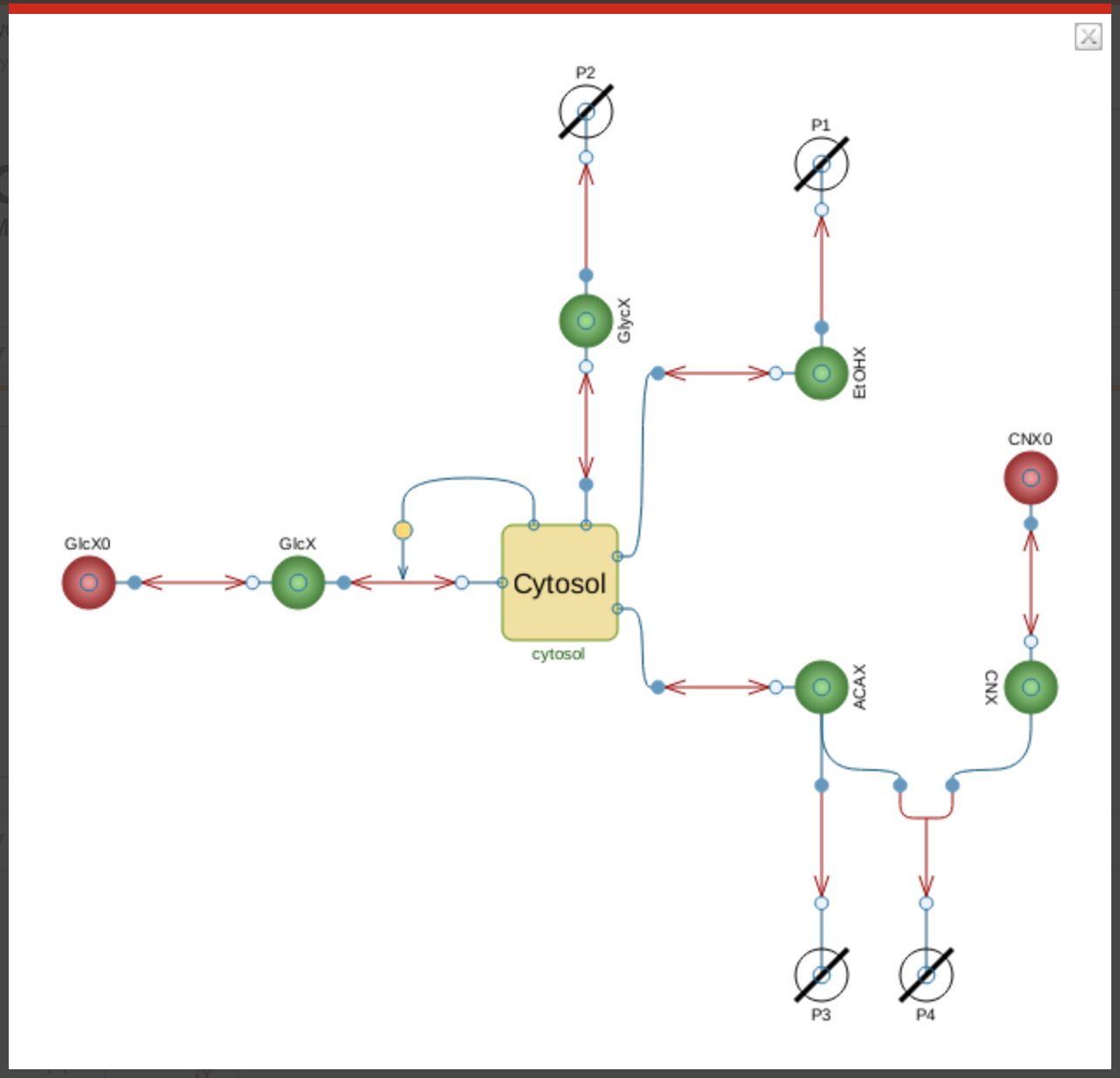


The Bioman website game also gives you an option to save the game for later. We will implement a save feature within our website in order for it to be viewed by students on different devices and to allow for models to be viewed later. It isn’t clear what database or web framework the Bioman application is using. However, as a group, we have decided to use Django as our web framework and MySQL as our database system. Django and MySQL are both popular choices for web development, are both free and have excellent support and documentation.

Scratch[[2]](#footnote-1) is similar to our desired website in terms of the drag-and-drop user interface that it supports. Scratch is a tool to introduce kids to programming. This tool allows users to drag and connect commands together to build games or web applications. The similarities exist in the drag-and-drop interface and the fact that it’s a web application whose graphics can be manipulated by the user. However, Scratch is very open-ended and can be used to do many things, our website has a more specific use. Our goal is to allow the user to create a model that is deemed accurate by our sponsor as well as an in-depth simulation of the biochemistry concepts in the cell.

The Lego Mindstorms NXT[[3]](#footnote-2) software is similar to how we would like to allow users to put together models. When using the Mindstorms NXT software a user is displayed an empty grid where they can attach different command blocks together and edit each block to their specifications. Similarly, our website will display users an empty grid where they can attach modules together and customize each module to their specifications. Conversely, our website will be slightly less customizable, only allowing a user to add modules in specific locations and restricting what each module will be capable of doing according to the already existing modules.

The Wolfram System Modeler[[4]](#footnote-3) allows users to simulate different processes from areas such as mechanical engineering, electrical engineering, and computational biology. The Wolfram model of glycolysis is similar to the concept we have for our website in that it allows people to view and simulate glycolysis. A picture of the graphical representation of glycolysis from their website can be seen in the picture below, figure 3. In this model, one can see a series of lines which represent the path a molecule may take and circles representing a point where a reaction will occur. Our model website will similarly allow users to view a model like this, however, our product will differ by allowing a user to interact with and change variables as well as clearly indicate the direction of the flow of molecules. The model of computation we will be using is a continuous time model, this is because variables will be constantly changing and each constantly changing variable may or may not affect another variable or component. Similarly to how these pathways and “variables” are executed in real time in the body.



**Figure 3**

The NetLogo[[5]](#footnote-4) modeling software allows users to create interactive models for anything they would like. A user can create buttons, dialog boxes, text boxes, sliders and more that allow the other people to interact with the system they create. A user can also create their own backend math functions that will update the model when it is run. This software is similar to the website we would like to create in that it allows a user to create and customize interactive models or simulations. However, our website differs from NetLogo, in that it will be optimized specifically for the study of biochemistry pathways and only allow users to model biochemistry pathways. By doing this our website will make the modeling of these pathways much simpler.

Ptolemy II[[6]](#footnote-5) is a tool for modeling, simulating and designing concurrent real-time systems. It primarily focuses on the use of well-defined models of computation that govern the interaction between components. Ptolemy II is similar to our project in that it is based around creating a system of components that interact with each other using mathematical equations. The downside to Ptolemy II is that it is mostly code based and requires a substantial amount of time/knowhow to be able to create complex systems. It also does not have support for allowing the user to create moving components or represent the flow through the system, other than with an arrow. The idea of our system is to make creating models simple for anyone in biochemistry, while simultaneously being visually pleasing. All of these areas are currently lacking in Ptolemy II, yet many of the ideas of interconnecting components will be the same.

**2 Project Requirements**

**2.1 Major Features**

The major features in this project are the tools for creating models of biochemical pathways, the ability to save the models once they have been created, the ability to share these models, and for the platform to be lightweight enough to run the platform on the internet browsers of many different devices, including desktop and mobile browsers. The tools for creating these models is the main focus of the project and defines the reasoning behind it, making it a necessary major feature. Once the models have been built, the creator of these models will need to save them for editing and usage later. This is important since the models will likely be very complex and difficult to recreate. To make their models available to students, an educator must be able to share the model, allowing for the students to modify certain preset elements but preventing them from changing others. This is also crucial to the project since the models are being created for student learning both in the classroom and at home. To improve mobility, the website will need to be fairly lightweight. Running this application through a desktop browser is a necessary part of the project and being able to interact with a premade model on a mobile device is also important.

**Table 1: Major Features**

|  |  |
| --- | --- |
| *Feature* | *Description* |
| Tools to Create Models | A student/educator will be able to create custom models of biochemistry pathways. |
| Save/Share Models | A model creator will be able to save/share their models, allowing them to edit and reuse them in the future as well as send to other users. |
| Intuitive User Interface | The user interface will be attractive, straightforward and intuitive to students of the natural sciences. |
| Cross-platform | The model creation platform should be lightweight enough to run on any modern browser and the models that are created should be accessible from a mobile device. This application should also avoid using features that are not available on mobile devices. |

To develop this checklist of features, we discussed with our sponsor what the overall plan was for the product, what the ideal application would look like, what an unacceptable product would look like, and what key requirements were absolutely necessary for the final application. After multiple meetings, we discovered that some features that seemed very important at first were actually just features that would be nice to have. At the same time, we discovered that, although creating a model of a biochemistry pathway was important, that was not the main focus of the project. The application that the sponsor was actually looking for was a platform to build these models rather than just creating the models ourselves.

Some of the features that we thought would be nice to have were not included in our major features checklist for various reasons. For instance, we thought it would be nice to have an administrative user that could monitor the changes a student makes on the models provided to them and decide whether the student accurately simulates a situation specified by the educator. This was not included because the sponsor does not need the educators to be able to assign homework using this application. It is meant as a study tool rather than a tool for making assignments. Another potential feature that we did not include as a major feature is the ability to view every model that has been made by any user. This is not necessary because students will only need to access the pre-made models or those shared with them by their professor. However, this would definitely be one of the most important future developments for this project.

**2.2 Initial Product Backlog**

The following table, Table 2, is a complete list of the initial requirements for this project. The items in this list are all necessary features that will need to be included in the Minimum Viable Product (MVP). The estimate for each item is a generous average, in developer hours, of each group member’s estimated time for each entry.

**Table 2: Initial Product Backlog**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Requirement* | *Description* | *Acceptance Criteria* | *Major Feature* | *Priority* | *Estimate* |
| Create login screen UI | Create a screen from which the user can enter username and password and the security software (in our case Django) can confirm that they match | Each GUI component does its intended function when the user interacts with it. Correct login information is successfully verified by DB | Ability to save models | 1 | 10 |
| Create main screen UI | Create interface for a screen that displays when the user enters the application | Each GUI component does its intended function when the user interacts with it. | Tools to create models | 1 | 10 |
| Create sign-up screen UI | Create interface for a screen that allows user to sign up and add their username to the database | The screen must have working buttons and intuitive design, as decided by the user base. New user information appears in the DB | Tools to create models | 1 | 10 |
| Create mode choice screen UI | Create interface for a screen that allows the user to choose whether to view or edit/create a model | Each GUI component does its intended function when the user interacts with it | Tools to create models | 1 | 10 |
| Create model choice screen UI | Create interface for a screen that allows the user to choose a model to edit or view | The screen must have working buttons and intuitive design, as decided by the user base | Tools to create models | 1 | 10 |
| Create model view screen UI | Create interface for a screen that allows the user to view a previous model | The screen must have working buttons and intuitive design, as decided by the user base | Tools to create models | 1 | 20 |
| Create model edit/design screen UI | Create interface for a screen that allows the user to edit and design their model | The screen must have working buttons and intuitive design, as decided by the user base | Tools to create models | 1 | 25 |
| Create module edit screen UI | Create interface for a screen that allows the user to edit a small module of their model | The screen must have working buttons and intuitive design, as decided by the user base | Tools to create models | 1 | 15 |
| Create encryption for passwords | Passwords that are entered into the sign-up/login screen should be stored as encrypted data and should have the ability to be checked against the user’s login input | Passwords must be encrypted and paired with a username. Must accurately check the username and password for a match | Ability to save models | 1 | 15 |
| Create functions to connect login screen to the database | The login screen should function properly by allowing the user to log in or sign-up | Must check database for username, and if it is not there, add username and encrypted password to database | Ability to save models | 1 | 6 |
| Connect Django with MySQL | Use Django as our framework to be intermediary between our database and our frontend web app. Establish a secure connection between frontend and database | Web application must be able to display information from the database on the website | Ability to save models | 1 | 8 |
| Create functions for each type of shape to create a new one | The user should be able to create any of the possible shapes including arrows, circles, boxes, and lines | User should be able to add any one of the predetermined shapes | Tools to create models | 2 | 15 |
| Create representation for flow between reactions | The user should visually see that the flow changes depending on the changes that the user makes to the enzymes | Must change the flow based on the changes made by the user | Tools to create models | 2 | 15 |
| Create Save for later system for users | Create a system where the user can save his/her personal model data to the MySQL database for use on a different/same device at a later period | Must save the model so that it can be accessed from another device | Ability to save models | 4 | 20 |
| Create sharing ability using others' usernames | Allow a user to share any of the models that they have created using this platform and share them with other people. | Must allow the other person to access the model after sharing the model | Ability to share models | 3 | 8 |
| Design background and Music for the model creation pages | Create a background for the website and music that will be playing while the user in creating and viewing the application | Must be aesthetically pleasing as determined by the user base. | Tools to create models | 1 | 6 |
| Create a function for adding pop-up windows for the description | Create a function that will allow the user to add a pop-up window to a molecule in the pathway that will display an image, a description, the reactions it can go through, and any formulas that are behind it | Must display the pop-up window and it must be aesthetically pleasing as determined by the user base | Tools to create models | 2 | 8 |
| Connect the functions for the module edit screen | Use the previous functions, such as the create shape function and the connect shapes and lines function to allow the user to make a small module | Must allow the user to create a module that is connected with the flow as defined previously | Tools to create models | 2 | 12 |
| Create a function for the user to add an enzyme to a reaction | Allows the user to input an enzyme to an arrow and input the function that demonstrates the usefulness of the enzyme on the reaction | Must allow the user to add an enzyme which they will later be able to change | Tools to create models | 2 | 8 |
| Create an interactive slider or wheel for an enzyme | User should be able to change the flow of the molecules running through a reaction by changing the amount of the enzyme associated with it | Changes in the slider or wheel must change the flow in the model | Tools to create models | 2 | 8 |
| Create icons to represent different parts of the model | Design graphical icons to represent each object in the model. Make sure that each representation is easy to identify and distinct. | Must have a clear, intuitive representation of each portion of the pathway as determined by the user base | Tools to create models | 1 | 10 |
| Host website on a server | Have our website hosted on the Gonzaga local servers for the time being. Make sure that the website is well supported and can withstand a large number of users. | Must be constantly displayed on a website from the Gonzaga local servers and support at least 200 users | Ability to save models | 1 | 10 |
| Add functionality to browse public models | Allow the user to browse other models that have been made public by the creator. This is in addition to allowing the user to view models that were shared with them. | Must be able to browse the models that have been made by other people but have not been shared with the user | Ability to share models | 4 | 20 |
| Advanced tab for each element | Allow the user to add optional features for the elements such as other images, descriptions, and important details | Must display the information in an aesthetically pleasing way as determined by the user base | Tools to create models | 2 | 10 |
| Create a mobile version of each site | There should be a mobile version of each screen that does not break on any modern mobile devices | Must display the screen on mobile devices without unwanted overlap | Mobile - Friendly | 1 | 25 |
| Ensure this application can be run on mobile devices | This application should avoid using features such as hover states and links that are close together to enable it to be run on mobile devices | Must have all the features of the full site on a mobile device. | Mobile - Friendly | 1 | Ongoing |

**2.3 Additional Features**

The following are the features that are non-essential but would be nice to have if there was additional time available.

**Table 3: Additional Features**

|  |  |
| --- | --- |
| *Feature* | *Description* |
| Create a definition box | Allow users to click on certain objects and then have a dialog box pop up that explains what that object is and give definitions. |
| Design icons for buttons | Create graphical designs for a back button, main menu, save, create a new model, and login buttons |
| Allow for an administrator | Allow for administrative users to be created that can monitor the progress of "students" to see if they have been using the app. |
| Add functionality to browse public models | Allow the user to browse other models that have been made public by the creator. This is in addition to allowing the user to view models that were shared with them. |

**3 Design Considerations**

**3.1 Initial User Interface Design**

Our user interface is composed of the main screen, sign up screen, login screen, mode choice screen, model choice screen, model view screen, design model screen, and edit module screen. In the main screen (see figure 5) the user would choose to login or sign up. If their account is created they would then login (see figure 6) and see the mode choice screen (see figure 7). From there they could either choose to view an existing model or go into design mode. If they had chosen design mode they could either create a new model or edit an existing model. In design mode (see figure 10) the screen is divided into modules that they can edit. If they click a module or click a plus icon then they are shown the edit module screen (see figure 11) where they can change settings for that module. If the user had chosen the view model screen they would be shown the model choice screen (see figure 8) where they would need to choose what model they would like to view. From there, the selected model would be displayed in the model view screen (see figure 9) and they would be given the option to run the model.

The following are the mockups of the GUI as well as the flow between screens. We arrived at these mockups through group discussion and by combining each of the drawings we had created for the screen flow. The only area we differed in was the design and model view screens. We were somewhat unsure how our icons would look and what we would use to represent the flow through each path. After discussing with our sponsor and advisor we decided to use these mockups as they are broader and will allow us to continue to consider the best way to graphically represent the flow of molecules through a cell.

**Figure 4:** GUI Flow

This figure shows the directions a user may go to navigate from screen to screen. The numbers indicate a likely order for how a user will navigate from screen to screen. The one screen the user will always encounter is the Main Screen, as it is the first screen to be displayed when the user goes to the website.



**Figure 5**: Main Screen

This figure is a mockup of how the main screen may look. This is the first screen a user will see on the website.



**Figure 6:** Sign-up Screen

This screen is where a user would go to create their account. Both Name and Email will be eligible as usernames and any entries will be verified as unique before being deemed as acceptable. The user will also indicate if they are a student or a professor, as a professor may be allowed different capabilities than a standard user. The user will need to enter a password in the Password 1 box and then re-enter the password in the Password 2 box.



**Figure 7:** Login Screen

The login screen will allow the user to enter their existing username and password, both entries will be verified with the database.



**Figure 8:**  Mode Choice Screen

The Mode Choice screen will allow the user to choose whether or not they would like to view an existing model or Design a new model. By clicking the Design button a user will also be allowed to edit existing models.



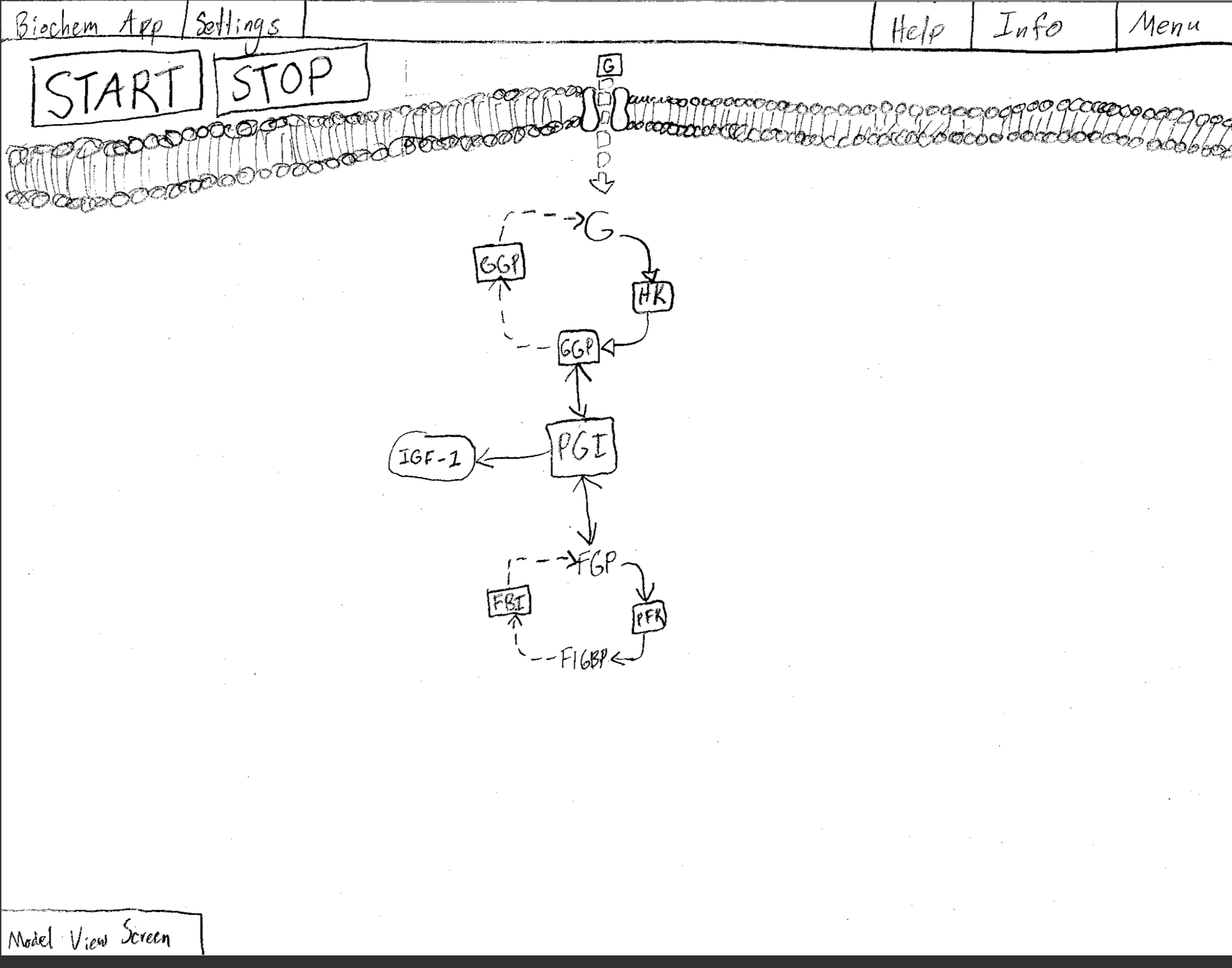
**Figure 9:** Model Choice Screen

From here the user will be able to choose to view an existing model.



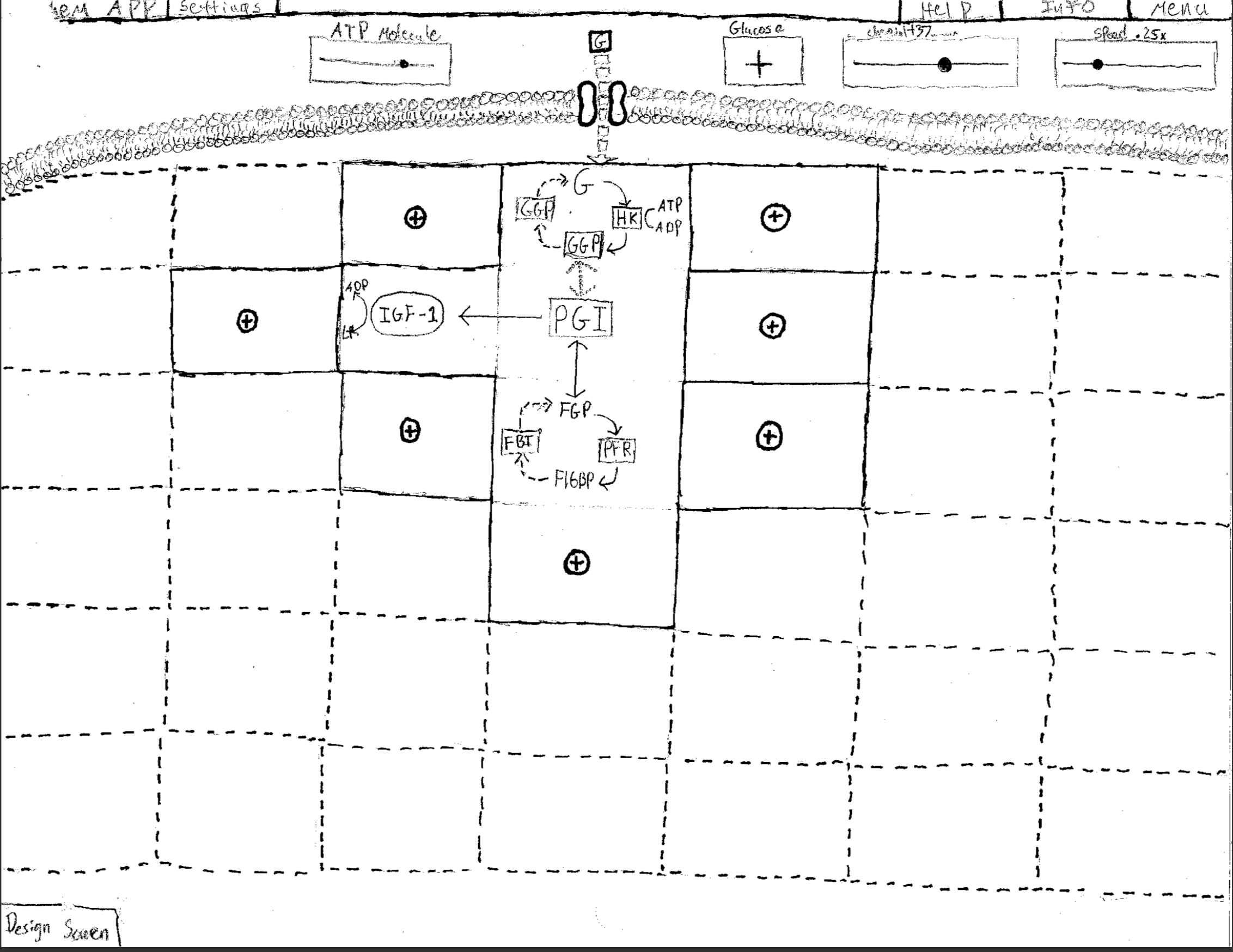
**Figure 10:** Model View Screen

In the model view screen, a user will be able to view a model and interact with it. They will have the ability to run a model to see how the variables would change and click on individual modules, allowing them to alter the quantity of the variables in the pathway. When the model is run the user will see which direction the molecules flow through the pathway and how the quantities of each of the variables, molecules, proteins, and enzymes, change during the run.



**Figure 11:** Design/Model Edit

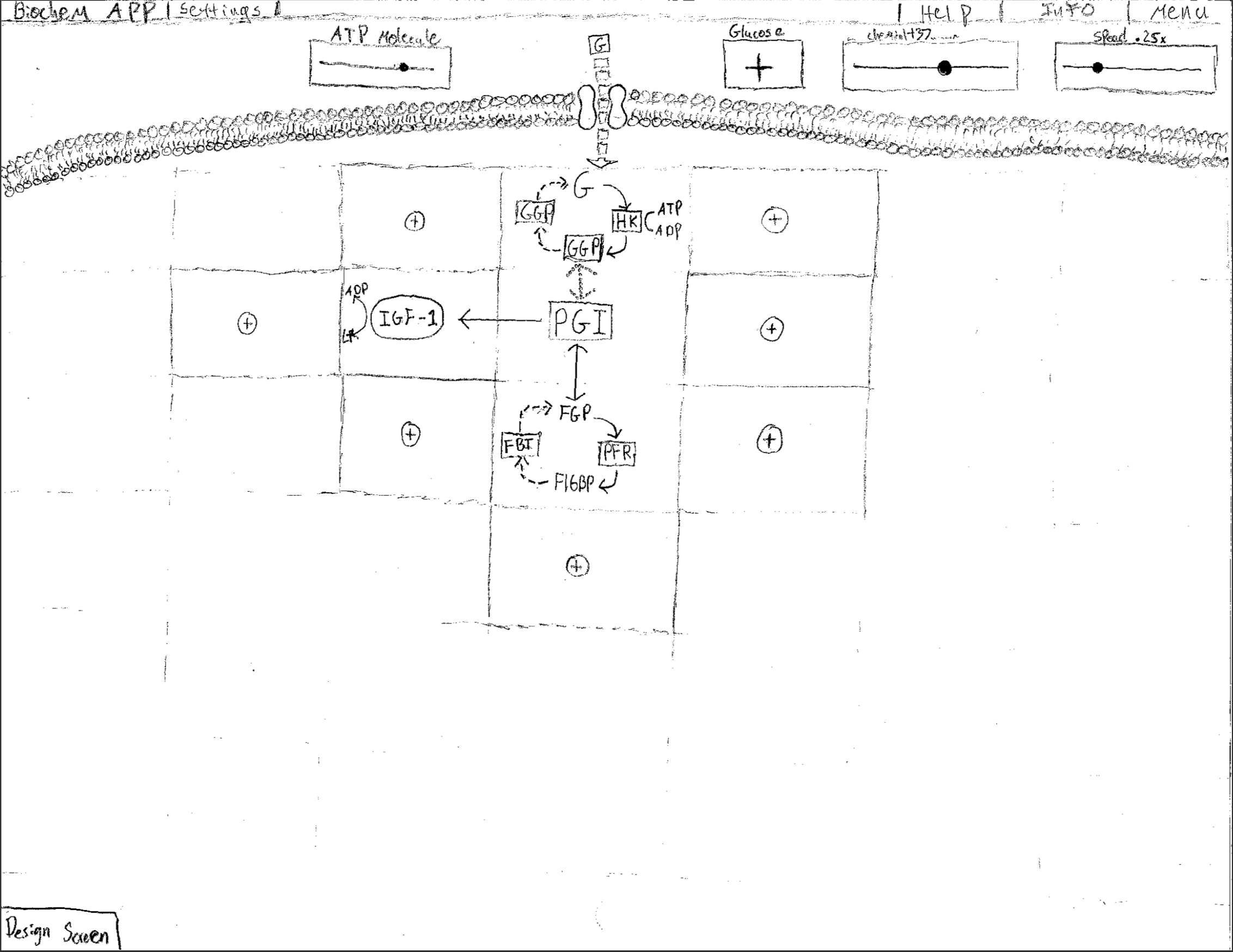
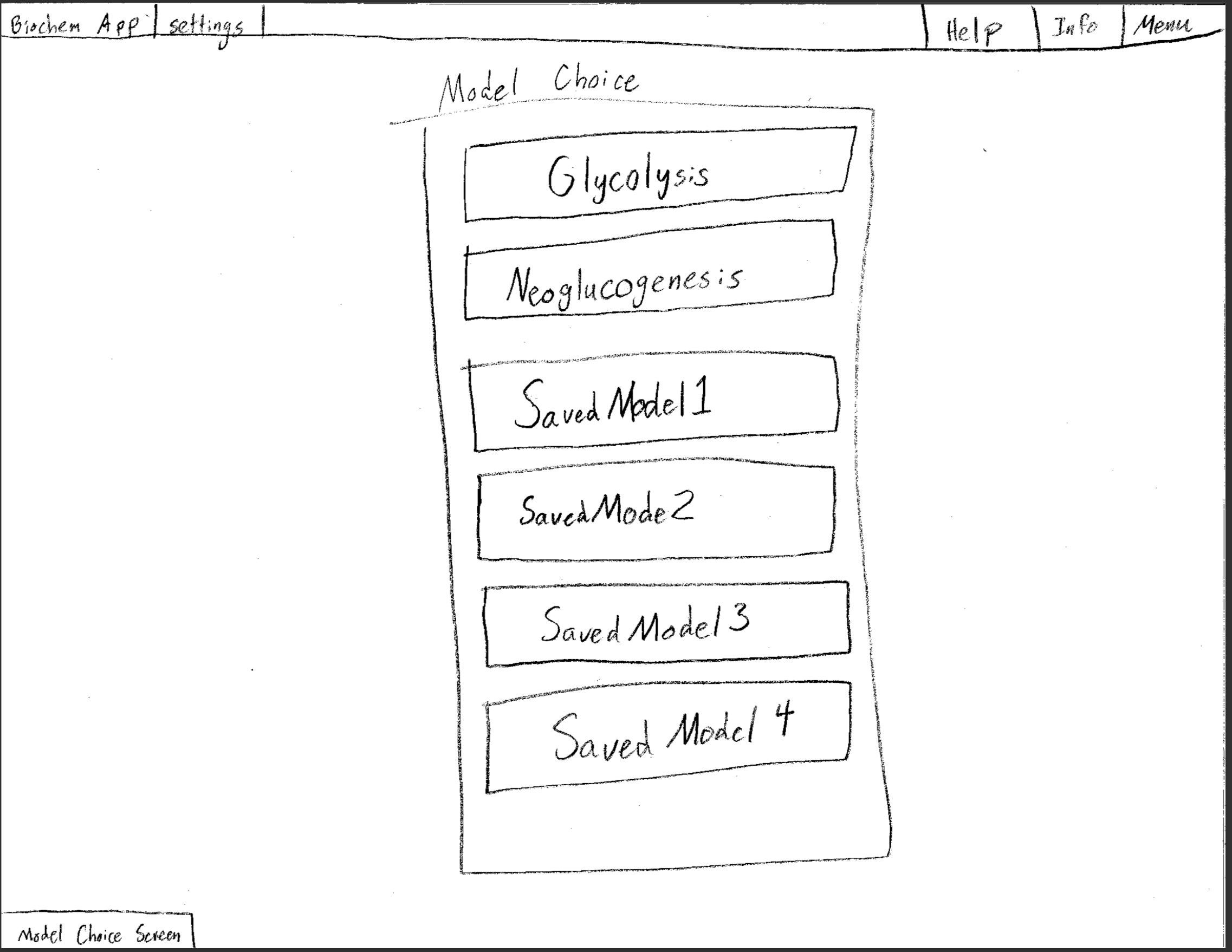
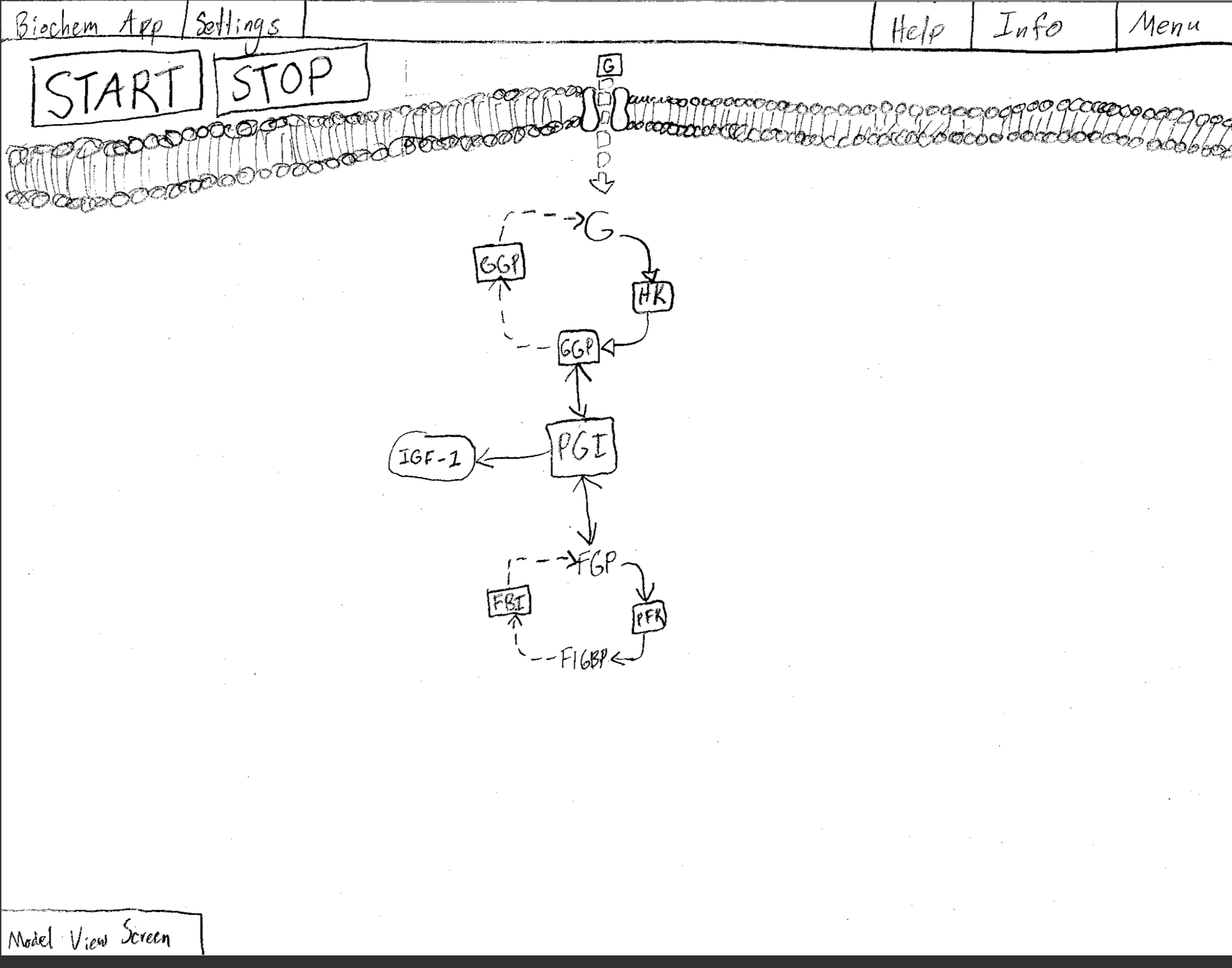
The design/model edit screen will allow the user to view the model, make changes to the model and add to the model. A user would be able to alter the initial quantity of each variable in the pathway, link this module to other modules in the pathway, and change what variables are involved in the reaction.



**Figure 12:** Edit Module Screen

In the edit module screen a user will be allowed to adjust and edit variables involved in a reaction.





**3.2 Initial Software Architecture**

The architecture of the website consists of a frontend UI, a MySQL database and a Django web framework, see figure 12. The MySQL database will be accessed through functions set within Django and UI will be updated by connecting to Django through the JavaScript and HTML that makes up the web application. The use of Django will provide an extra level of separation between the UI and database in order to avoid the inclusion of connection strings within the code. Additionally, the graphics of the application will be generated using the canvas tag in HTML.



**Figure 12**

**3.3 Initial Software Test Plan**

The first test we will do in our test plan is a user interface test/usability test. The usability test will be done by students of Dr. Watson and by Dr. Watson himself to see if the look and feel of the application is up to par with other study tools. This is important because we want our application to be geared toward what will work best for students and the ability for the user to interact with the application in a fluid and simple manner. A usability test will have to be done more than once as the application becomes more complex. The first usability test will be during the first month of development and will be based on the user interface to make sure we are headed in the right direction.

The second test will be done to verify that users have the ability to login from any computer and that the pathways stored in the database work without any bugs. We want to make sure any device can access the models on any browser without complications. This will be done by the developers, the process will be to open the application and create and save pathway models and then open them on different devices and different browsers. We want this application to be conveniently usable by all students so availability on all devices will be key. This will be done early on in the project so we can accommodate any potential problems as we continue to build the application.

The third test and most important will be the functional test or in other words, a test to make sure that the biochemical pathways are completely modular. We will need to verify that a user can do whatever they want when it comes to creating the pathways without bugs. This will be tested by creating and linking nonsense to each other and making sure they still work as expected. A test like this is important because we want to make sure whether the pathway is correct or not that it will still represent itself as the user would expect. This will be done midway through the development cycle so we have ample time to fix any problems.

The fourth test and most often utilized will be unit tests, these will occur continually throughout our project typically taking place near the end of each sprint. This will be useful when we are adding code to our product, so we can make sure new additions do not negatively affect the previously written code. These unit tests will consist of small individual tests on each component of our application validating every piece is working as it should be. We will be using a unit testing system called Jest, a popular and well-documented system touted by Facebook.

**4 Project Risks**

The first risk would be that the biochemistry mathematical functions may end up being too convoluted for our purposes. The math for what occurs inside each cell is composed of many fairly complicated calculations with many variables that may be outside the scope of the parameters Dr. Watson would like us to represent. Currently, our sponsor, Dr. Watson, is working to determine the type of math necessary to support the models we would like to create and plans to consult the math department. However, if the math will likely be more complex than is necessary for our model. Thus, Dr. Watson will provide us with a real-world approximation for the pathway behavior. This could make the model less precise but the model will still provide a good representation of the occurrences inside the cell, which is the goal of the project. We will monitor the risk by ensuring we start working on the math as soon as possible so that if an issue does occur there will be enough time to remedy it. We will need to mitigate the risk if we are not able to conceptualize the math and translate it into code. If we cannot figure out how the math works we will need to alter the scope of our project and figure out the math for at least one pathway under ideal circumstances.

Another possible risk would be that our GUI becomes too inaccurate as the complexity is increased and more modules are added to the pathway. If a user can add unlimited modules to the pathway the model could become too difficult to make sense of and inaccurate. In order to prevent this, we are prepared to limit the number of modules a user can add and the number of variables that can be in the pathway. This would make the model simpler and the calculations less complex. We will monitor the risk by testing the limits of our website. We will need to mitigate the risk if the model runs too slowly or our sponsor considers it to be inaccurate. If we cannot create a reasonably sized model without it becoming too complex we will need to revisit the math behind the scenes and do everything in our power to optimize speed.

A third risk would be that our website is not supported cross-platform. Given the type of graphics, we will need to display it may become difficult to ensure that the website looks right on most browsers and this could cause us to spend too much time perfecting our website for each browser. In order to prevent this, we will continuously test our website in the most common browsers, Chrome, Safari, Internet Explorer, and Firefox to ensure that we do not need to completely reconfigure large parts of our website. We will monitor the risk with continuous testing in different browsers and on different computers. We will need to mitigate the risk if the GUI does not function properly in a particular browser. If we cannot enable the website to work properly in every browser we will recommend that only certain browsers are used and optimize the website for those browsers.

Finally, as we will need to learn new technologies, concepts and software to complete this project, our level of expertise is a potential risk. Lack of experience with Django and MySQL, coupled with a lack of understanding of biochemistry pathways and experience in web development means that we will need to invest a lot of time in learning. This could cause us to take longer than anticipated to create the MVP. In order to prevent this, we will be sure to use our time wisely and start developing as early as we can. Additionally, we will be sure to use all resources available and reach out to our advisor, sponsor, professors and peers when needed. We will monitor the risk by continually updating our backlog and task sheet so that we can keep track of our progress and verify that we are on track. We will need to mitigate the risk if we need to spend more time learning than developing and get behind schedule. If we cannot stay within our time constraints we are prepared to narrow our scope so that we can get the MVP completed.

**5 Initial Product Release Plan**

**5.1 Major Milestones**

The major milestones will be the first prototype, the platform with limited capabilities, the platform with saving, login, and sharing capabilities, the full platform, and the fully tested and documented platform. The first prototype is due first since it must be created for the sponsor to review since the layout of the application is the portion of the application that will likely need the most input from the sponsor. The platform should be created next since that is the basis for the rest of the application. The saving, login, and sharing options can be most effectively implemented after the base platform is created since there will be an accurate approximation of the size of the models that will be saved. The next step is to finish the rest of the project so that it is ready for alpha or beta testing. Finally, the full tested product and documentation must be completed and ready for deployment.

**Table 3: Major Milestones**

|  |  |  |
| --- | --- | --- |
| *Milestone* | *Description* | *Target Completion Date* |
| *Prototype* | There must be a prototype that is able to display some information from the database and have a full web layout | First week of December |
| *Working platform for modeling a simple pathway* | There must be a platform where a pathway can be modeled. This does not need to include all the options, just what is needed for a simple pathway | Third week of January |
| *Working platform with saving, login, and sharing capabilities* | Must have all the features of the previous versions as well as the options to save, login, and share. | Second week of February |
| *Full product* | This should be the full product ready for alpha or beta testing. | Third week of March |
| *Full product and documents* | All of the documents should be completed as well as the full product which should have gone through alpha testing. | Last week of April |

**5.2 Initial Sprint Releases**

The following table is our anticipated sprint release plan. By the end of each sprint, we plan to have each goal met and the corresponding backlog entries completed. We plan to adjust when necessary, however, the sprints up to January 16th should all be completed in the time allotted.

**Table 4: Sprint Release Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| *Sprint Date* | *Sprint Goal* | *Backlog* | *What we will demo* |
| *10/24 - 11/6* | Develop login screen and main screen | Create login screen  Create Main Screen | Show that the buttons on the main screen allow a user to navigate to a sign up or login page. Show that a user is able to enter input into the sign up and login page. Show that there are functions created that will later hook in to the database and send the user input for validation. |
| *11/7 - 11/20* | Develop the sign up screen and create a user database where the users data can be stored | Create Sign up Screen  Create user database | We will show that a person can create an account by entering a unique username and password in the UI. Show that the entered username and password are stored in the database table. Show that the new user can now login and their account information is validated. |
| *11/21 - 12/4* | Develop the graphical representations of the interactive pieces of the model | Create icons to represent different parts of the model  Create representation for flow between reactions | Show each icon and explain what the icon represents. Show the icons representing molecules flowing along a line. |
| *1/16 - 1/29* | Develop the UI for the model creation page  **Usability Testing** | Design background and Music for the model creation pages  Create module edit screen UI | Show a screen that displays the chosen background and when the screen appears have music begin to play. |
| *1/30 - 2/12* | Develop model flow creation page I | Create representation for flow between reactions  Create functions for each type of shape to create a new one | Show a screen that displays a screen with a rendering of a cell wall and an empty grid below it with one + icon on it. Show that when the plus icon is clicked a user is sent to another page that they will later be able to edit their module on. |
| *2/13 - 2/26* | Develop model flow creation page II  **Functional Test** | Create a function for the user to add an enzyme to a reaction  Create an interactive slider or wheel for an enzyme  Advanced tab for each element | Show that when the + icon from the previous model flow creation page is clicked that the user is displayed another screen giving them the option to customize their module by adding connection ports and altering variables. |
| *2/27 - 3/12* | Create a database system which allows the user to be able to save  **Database/User account Testing** | Create Save for later system for users | Show that you can log on to an account, save something, and then access it on a different computer. |
| *3/13 - 3/26* | Develop a share system between users and make sure the application can be run on any popular device | Add functionality to browse public models  Ensure this application can be run on mobile devices | Show that you can search for other people’s models in the application. Showcase the application working on multiple devices. |
| *3/27 - 4/9* | **Usability Testing**  Misc. Error Fixing |  |  |
| *4/10 - 4/23* | Misc. Error Fixing  Deployment |  |  |
| *4/24 - 5/10* | TBD |  |  |

**6 Maintenance Considerations**

Since this is a modular application (the user will be able to model any number of pathways they would like) if the project is completed and passes all of our unit tests then the only maintenance there should be is dealing with the storage capacity of the database. This will be done by whoever is in control of the application since this is a new website isn’t certain whose hands the project will end up in once completed. The database will need to be scaled based on how many users we have, currently, we are looking to have anywhere between 50-100 users just at Gonzaga. With the amount of storage we currently have in our database this should be fine, but after our senior design is over we will lose the school hosted database and will have to find a storage host somewhere else.

The maintenance for the project could end up in our hands or in other students graduating in the coming years or even end up in the hands of some biochemistry book publisher. We are using popular up to date development applications for a website and will be thoroughly documenting our code in order to make any maintenance as easy as possible. The developer will need to know how to talk to a MySQL database through Django and be able to write code proficiently in HTML, JavaScript, and Canvas.

Once the Minimum Viable Product has been created, this product could be expanded upon in a number of ways. One way would be to expand the number of pathway models that are premade. Another potential expansion would be to allow our website to model other processes in biochemistry, rather than just the intracellular pathways. Such as cell replication, DNA replication, neuron communication, nutritional science as well as hundreds of others potential models in the natural sciences.

**7 Project Management Considerations**

The team will meet weekly on Mondays in Herak at 5 pm and Wednesdays in Jepson at 4:25 pm to discuss product tasks and what needs to be completed before the next time we meet. We meet with our faculty advisor, Rob Bryant every Monday at 3 pm in his office, Herak 226 and our sponsor, Dr. Watson right after every other Monday at 4 pm in the Humanities building. Additionally, we plan to get in touch with our DAB members, Mike Mudge, and Dan Lentz through email and meet with him once a month if necessary. We will update our Sponsor, Advisor, and DAB member through email or in person. Throughout the year we will evenly break up the work by assigning each team member a weekly task during our Monday meetings. Lastly, we are planning on communicating our progress to our team members and sponsors through an Excel weekly task sheet and email.

1. <https://biomanbio.com/HTML5GamesandLabs/Cellgames/celldefensehtml5page.html> [↑](#footnote-ref-0)
2. <https://scratch.mit.edu/> [↑](#footnote-ref-1)
3. <https://www.youtube.com/watch?v=0A6SA3h0hbc> [↑](#footnote-ref-2)
4. <https://reference.wolfram.com/system-modeler/libraries/BioChem/BioChem.Examples.YeastGlycolysis.Glycolysis.html> [↑](#footnote-ref-3)
5. <https://ccl.northwestern.edu/netlogo/> [↑](#footnote-ref-4)
6. <https://ptolemy.berkeley.edu/ptolemyII/index.htm> [↑](#footnote-ref-5)